

---

# PyCozmo

*Release 0.8.0*

Nov 15, 2020



<b>1</b>	<b>Overview</b>	<b>1</b>
1.1	Usage . . . . .	1
1.2	PyCozmo vs. the Cozmo SDK . . . . .	2
1.3	Requirements . . . . .	2
1.4	Installation . . . . .	2
1.5	Support . . . . .	3
1.6	Disclaimer . . . . .	3
<b>2</b>	<b>PyCozmo Architecture</b>	<b>5</b>
2.1	Overview . . . . .	5
2.2	Connection Layer . . . . .	7
2.3	Client Layer (SDK) . . . . .	7
2.4	Application Layer . . . . .	7
<b>3</b>	<b>Cozmo Protocol</b>	<b>9</b>
3.1	Overview . . . . .	9
3.2	Network Setup . . . . .	9
3.3	Frames . . . . .	10
3.4	Packets . . . . .	10
3.5	Connection Establishment . . . . .	12
<b>4</b>	<b>Capturing Cozmo Communication</b>	<b>13</b>
4.1	Overview . . . . .	13
4.2	Prerequisites . . . . .	13
4.3	Connecting to Cozmo . . . . .	14
4.4	Masquerading as a Cozmo . . . . .	15
4.5	Capturing Communication . . . . .	16
<b>5</b>	<b>Cozmo Functions</b>	<b>19</b>
5.1	Overview . . . . .	19
5.2	Wi-Fi . . . . .	20
5.3	Backpack LEDs . . . . .	20
5.4	Backpack Button . . . . .	21
5.5	Wheels . . . . .	21
5.6	Localization . . . . .	21
5.7	Path Tracking . . . . .	21
5.8	Head . . . . .	22

5.9	Lift	22
5.10	OLED display	22
5.11	Speaker	22
5.12	Camera	23
5.13	IR LED	23
5.14	Accelerometers	23
5.15	Gyro	23
5.16	Cliff Sensor	23
5.17	Battery voltage	24
5.18	NV RAM Storage	24
5.19	Firmware Updates	24
5.20	Bluetooth LE	25
5.21	Cube LEDs	25
5.22	Cube Battery Voltage	25
5.23	Cube Accelerometers	26
5.24	Animations	26
<b>6</b>	<b>Cozmo Off-Board Functions</b>	<b>27</b>
6.1	Directory structure	27
6.2	Audio files	28
6.3	Animations	28
6.4	Animation groups	28
6.5	Behaviors	29
6.6	Reactions	29
6.7	Emotions	29
6.8	Activities	29
<b>7</b>	<b>Cozmo Firmware Versions</b>	<b>31</b>
7.1	Production Versions	31
7.2	Factory Versions	33
<b>8</b>	<b>Cozmo Hardware Versions</b>	<b>35</b>
8.1	Hardware Version 4	35
8.2	Hardware Version 5	35
8.3	Hardware Version 6	35
8.4	Hardware Version 7	36
<b>9</b>	<b>ESP8266</b>	<b>37</b>
9.1	SPI Flash Memory Map	37
<b>10</b>	<b>pycozmo package</b>	<b>39</b>
10.1	pycozmo.audiokinetic.exception	40
10.2	pycozmo.audiokinetic.soundbank	41
10.3	pycozmo.audiokinetic.soundbanksinfo	42
10.4	pycozmo.audiokinetic.wem	44
10.5	pycozmo.expressions.expressions	44
10.6	pycozmo.activity	58
10.7	pycozmo.anim	59
10.8	pycozmo.anim_controller	60
10.9	pycozmo.anim_encoder	61
10.10	pycozmo.audio	66
10.11	pycozmo.behavior	66
10.12	pycozmo.brain	67
10.13	pycozmo.camera	68
10.14	pycozmo.client	69

10.15	pycozmo.conn	71
10.16	pycozmo.emotions	76
10.17	pycozmo.event	77
10.18	pycozmo.exception	79
10.19	pycozmo.filter	80
10.20	pycozmo.frame	81
10.21	pycozmo.image_encoder	81
10.22	pycozmo.lights	82
10.23	pycozmo.object	83
10.24	pycozmo.procedural_face	83
10.25	pycozmo.protocol_ast	86
10.26	pycozmo.protocol_base	91
10.27	pycozmo.protocol_declaration	93
10.28	pycozmo.protocol_encoder	93
10.29	pycozmo.protocol_generator	136
10.30	pycozmo.protocol_utils	137
10.31	pycozmo.robot	140
10.32	pycozmo.robot_debug	141
10.33	pycozmo.run	142
10.34	pycozmo.util	142
10.35	pycozmo.window	147
<b>11</b>	<b>Indices and tables</b>	<b>149</b>
	<b>Python Module Index</b>	<b>151</b>
	<b>Index</b>	<b>153</b>



# CHAPTER 1

---

## Overview

---

<https://github.com/zayfod/pycozmo>

PyCozmo is a pure-Python communication library, alternative SDK, and application for the [Cozmo robot](#). It allows controlling a Cozmo robot directly, without having to go through a mobile device, running the Cozmo app.

The library is loosely based on the [Anki Cozmo Python SDK](#) and the [cozmoclad](#) (“C-Like Abstract Data”) library.

This project is a tool for exploring the hardware and software of the Digital Dream Labs (originally Anki) Cozmo robot. It is unstable and heavily under development.

## 1.1 Usage

Basic:

```
import time
import pycozmo

with pycozmo.connect() as cli:
    cli.set_head_angle(angle=0.6)
    time.sleep(1)
```

Advanced:

```
import pycozmo

cli = pycozmo.Client()
cli.start()
cli.connect()
cli.wait_for_robot()

cli.drive_wheels(lwheel_speed=50.0, rwheel_speed=50.0, duration=2.0)
```

(continues on next page)

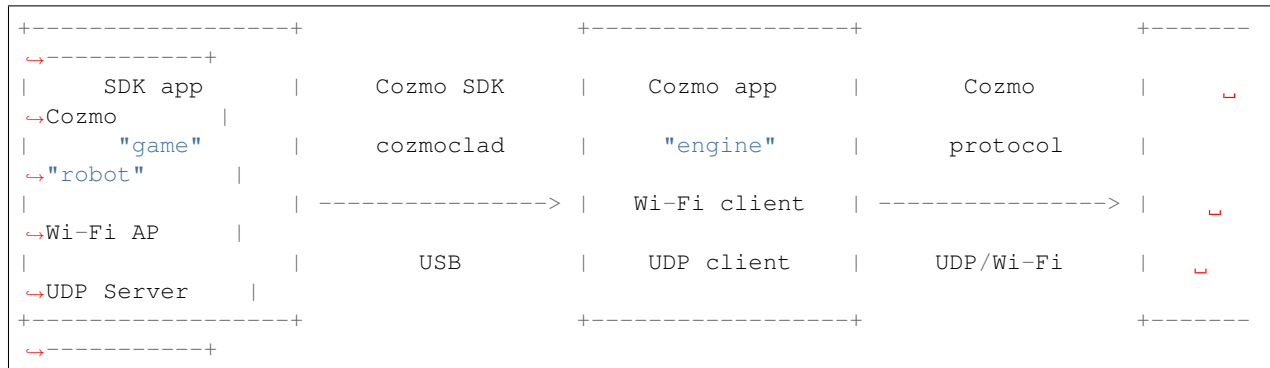
(continued from previous page)

```
cli.disconnect()
cli.stop()
```

## 1.2 PyCozmo vs. the Cozmo SDK

A Cozmo SDK application (aka “game”) acts as a client to the Cozmo app (aka “engine”) that runs on a mobile device. The low-level communication happens over USB and is handled by the `cozmoclad` library.

In contrast, an application using PyCozmo basically replaces the Cozmo app and acts as the “engine”. PyCozmo handles the low-level UDP communication with Cozmo.



## 1.3 Requirements

- Python 3.6.0 or newer
- Pillow 6.0.0 or newer - Python image library
- FlatBuffers - serialization library
- dpkt - TCP/IP packet parsing library
- OpenCV 4.0.0 or newer - computer vision library

## 1.4 Installation

Using pip:

```
pip install --user pycozmo
pycozmo_resources.py download
```

From source:

```
git clone https://github.com/zayfod/pycozmo.git
cd pycozmo
python setup.py install --user
pycozmo_resources.py download
```



From source, for development:

```
git clone git@github.com:zayfod/pycozmo.git
cd pycozmo
python setup.py develop --user
pip install --user -r requirements-dev.txt

pycozmo_resources.py download
```

## 1.5 Support

Bug reports and changes should be sent via GitHub:

<https://github.com/zayfod/pycozmo>

DDL Robot Discord server, channel #development-cozmo:

<https://discord.gg/ew92haS>

## 1.6 Disclaimer

This project is not affiliated with [Digital Dream Labs](#) or [Anki](#).



## PyCozmo Architecture

### 2.1 Overview

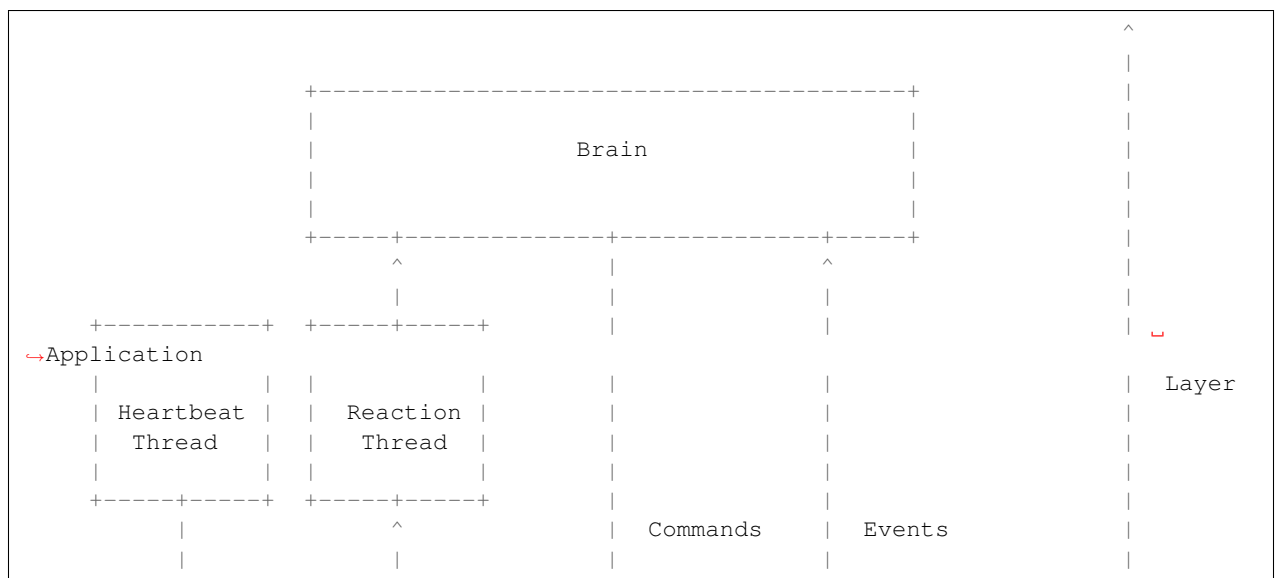
PyCozmo is designed as a multithreaded library.

It is organized in three layers with each higher layer building on the ones below it:

- low-level connection layer
- client or SDK layer
- application layer

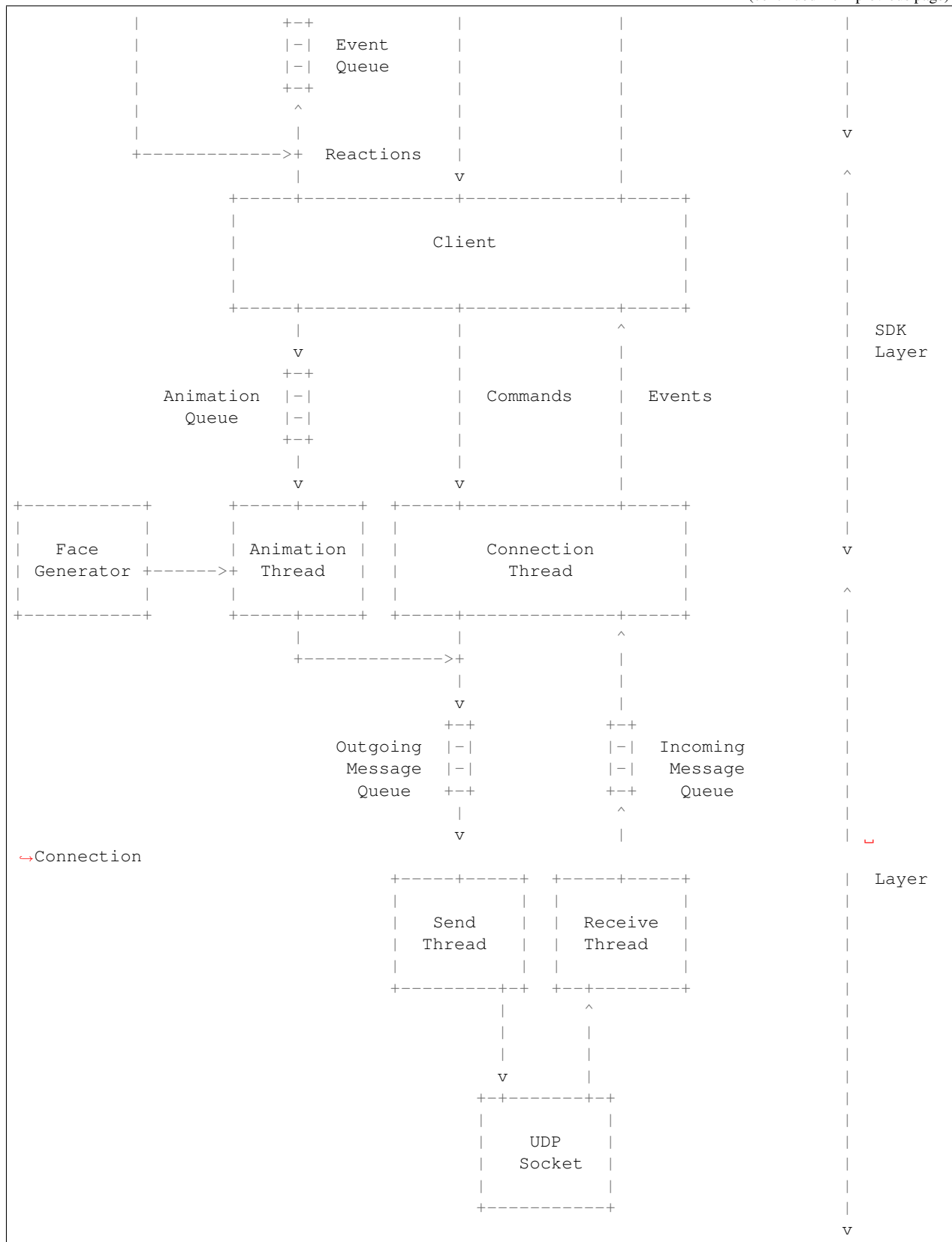
Each layer provides it's own API and can be used independently.

The following diagram illustrates the library architecture.



(continues on next page)

(continued from previous page)



## 2.2 Connection Layer

The connection layer implements the Cozmo communication protocol.

The receive thread reads Cozmo protocol frames, encapsulated in UDP datagrams, from the UDP socket. It maintains a receive window for incoming packets and sends a stream of incoming packets in the correct order over the incoming message queue to the connection thread.

The send thread reads a stream of outgoing packets from the outgoing message queue, builds Cozmo protocol frames and sends them over the UDP socket. It maintains a send window and resends packets that are not acknowledged in time.

The connection thread reads a stream of incoming packets from the incoming message queue and dispatches them to registered handler functions. It sends ping packets on a regular basis to maintain connection with the robot.

## 2.3 Client Layer (SDK)

The client layer provides access to robot on-board functions.

It allows sending commands and registering handler function for incoming packets and events.

It performs:

- camera image reconstruction
- display image encoding
- audio encoding
- animation and audio playback
- procedural face generation

The animation controller synchronizes animations, audio playback, and image display. It works as a separate thread that aims to send images and audio to the robot at 30 frames per second. All on-board function of the robot are synchronized to this framerate, including images, audio playback, backpack and cube LED animations.

## 2.4 Application Layer

The application layer implements high-level off-board functions:

- reactions and behaviors
- personality engine
- computer vision (CV) camera image processing

Events from the client layer are converted to reactions. The reaction thread reads events from its incoming event queue and handles them appropriately. Reactions normally trigger behaviors.

The heartbeat thread drives the personality engine and timers for activities and behaviors.



3.1 Overview

The Cozmo protocol is a UDP-based variant of the selective repeat automatic-repeat request (ARQ) protocol.

The Cozmo app (aka “engine”) acts as a client and Cozmo (aka “robot”) acts as a server.

The two exchange frames, encapsulated in UDP packets.

Each frame can contain 0, 1, or more packets.

See `protocol_declaration.py` for packet details.

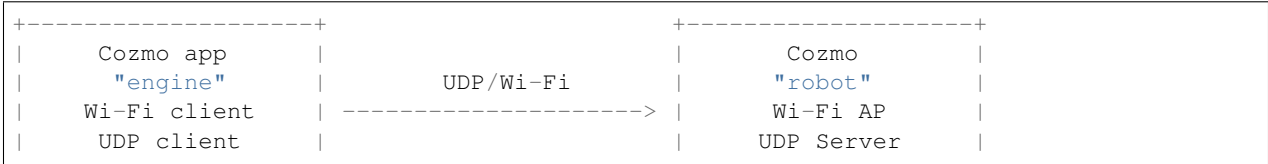
3.2 Network Setup

The robot acts as a Wi-Fi access point. It always uses an SSID that follows the form “Cozmo\_XXXXXX”, where XXXXXX are upper-case hexadecimal digits. It acts as a DHCP server and assigns Wi-Fi clients an IP address in the range 172.31.1.0/24 .

The app searches for robot APs. If it finds only one, it will associate with it automatically. If it finds more than one, it will allow the user to select one manually.

The robot acts as a server. It always uses the IP address 172.31.1.1 and will expect UDP packets on port 5551. It will only accept packets originating from an IP address in the range 172.31.1.0/24 .

The app acts as a client and initiates connections. It will only accept packets originating from the IP address 172.31.1.1 .



(continues on next page)

(continued from previous page)

+-----+	+-----+
172.31.1.0/24	172.31.1.1:5551

### 3.3 Frames

Each frame has the following structure:

Field	Length	Description
id	7	Always "COZ\x03RE\x01"
type	1	Frame type
first_seq	2	First packet sequence number in the frame or 0
seq	2	Last packet sequence number in the frame or 0
ack	2	Peer packet sequence number acknowledgement
packets	-	0 or more encapsulated packets

Frame types:

Type	Source	Description
0x01	engine	Reset
0x02	robot	Reset ACK
0x03	engine	Disconnect
0x04	engine	Engine packet - single
0x07	engine	Engine packets - zero or more
0x09	robot	Robot packets - zero or more
0x0b	engine	Out-of-band engine ping

### 3.4 Packets

Packet types:

Type	OOB	Source	Description
0x02	n	robot	Connect
0x03	n	engine	Disconnect
0x04	n	both	Command
0x05	y	robot	Event
0x0a	y	engine	Keyframe
0x0b	y	engine	Ping

Out of band packets do not get assigned sequence IDs.

Packet content is Cozmo firmware version specific.

Commands and events are identified by an 8-bit ID. IDs in the range 0-0xaf are sent by the engine. IDs in the range 0xb0-0xff are sent by the robot.

IDs in the range 0xf0-0xff are used for out-of-band updates. These are packets that are not tracked by a sequence ID and thus not retransmitted. Only their latest received value is considered important.



ID	Min	Max	Name	
0x03		31	31	LightStateCenter
0x04		40	40	CubeLights
0x05		5	5	ObjectConnect
0x0b		1	1	SetHeadLight
0x0c		1	1	
0x10		5	5	CubeId
0x11		21	21	LightStateSide
0x25		0	0	Enable
0x32		16	16	DriveWheels
0x33		10	10	TurnInPlaceAtSpeed
0x34		4	4	DriveLift
0x35		4	4	DriveHead
0x36		17	17	SetLiftHeight
0x37		17	17	SetHeadAngle
0x39		20	20	TurnInPlace
0x3b		0	0	StopAllMotors
0x3d				DriveStraight
0x45		24	24	
0x4b		8	8	EnableBodyACC
0x4c		2	2	EnableCamera
0x50		2	2	
0x54		2	2	
0x57		7	7	SetCameraParams
0x60		1	1	EnableStopOnCliff
0x64		2	2	SetRobotVolume
0x66		1	1	EnableColorImages
0x80		4	4	
0x81		12	144	* NvStorageOp
0x8d		0	0	
0x8e		744	744	OutputAudio
0x8f		0	0	OutputSilence
0x93		3	3	
0x94		3	3	
0x97		4	188	* DisplayImage
0x98		10	10	
0x99		4	4	
0x9a		0	0	
0x9b		1	1	
0x9d		1	1	
0x9e		1	1	
0x9f		0	0	EnableAnimationState
0xa0		16	16	
0xaf		1026	1026	FirmwareUpdate
0xb0		8	40	* UnknownB0
0xb2		16	16	
0xb4		21	21	ObjectMoved
0xb5		8	8	ObjectStoppedMoving
0xb6		12	12	ObjectTapped
0xb9		10	10	ObjectTapFiltered
0xc2		0	0	RobotDelocalized
0xc3		0	0	RobotPoked
0xc4		1	1	AcknowledgeAction
0xc8		29	29	
0xc9		6	6	HardwareInfo
0xca		1	1	

(continues on next page)

(continued from previous page)

0xcb	1	1		
0xcd	12	1036	*	NvStorageOpResult
0xce	9	9		ObjectPowerLevel
0xcf	8	8		
0xd0	13	13		ObjectConnectionState
0xd1	3	3		
0xd2	44	44		
0xd7	9	9		ObjectUpAxisChanged
0xec	4	4		
0xed	12	12		BodyInfo
0xee	449	449		FirmwareSignature
0xef	7	7		FirmwareUpdateResult
0xf0	91	91		RobotState
0xf1	15	15		AnimationState
0xf2	24	1172	*	ImageChunk
0xf3	9	9		ObjectAvailable
0xf4	17	17		ImageImuData

### 3.5 Connection Establishment

The engine sends a reset frame (0x01) to the robot with first\_seq and seq set to 1 and ack set to 0.

The robot responds with a robot packets frame (0x09) with first\_seq and seq set to 1 and ack set to 1, containing a connect packet (0x02). This establishes the connections.

The engine maintains the connection by periodically sending ping frames (0x0b). The robot responds with robot packet frames (0x09), containing a copy of the engine's ping in a ping packet (0x0b). The pings have a sequence ID and a time stamp and allow the engine to measure round-trip time.

If the robot stops receiving ping frames for more than 5 s it will disconnect and display the message “COZMO 01”.

The engine can gracefully close the connection in one of two ways:

- by sending a disconnect frame (0x03)
- by sending an engine packets frame (0x07), containing a disconnect packet (0x03).

As long as a connection is established, the engine and the robot can exchange packets.

The engine sends packets in frames of types 0x04 and 0x07.

The robot sends packets in frames of type 0x09.

---

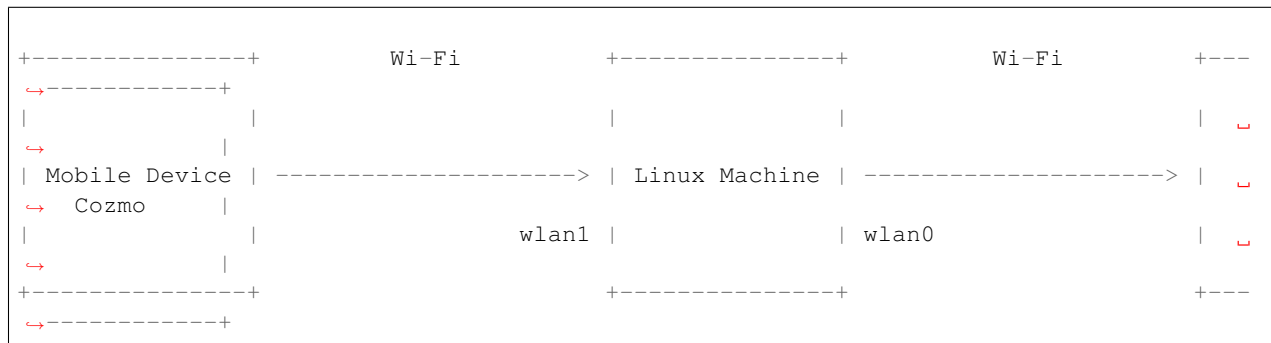
## Capturing Cozmo Communication

---

### 4.1 Overview

Capturing the communication between the Cozmo app and Cozmo is very valuable for understanding how Cozmo works.

One way to achieve this is by placing a Linux machine between the two as shown on the following diagram.



The Linux machine acts as a Wi-Fi client on one interface (wlan0) and associates with Cozmo. It acts as a Wi-Fi access point (AP) on the other interface and allows a mobile device, running the Cozmo app to associate with it.

With appropriate network configuration such a setup allows capturing Cozmo communication in [pcap files](#) using tcpdump.

### 4.2 Prerequisites

- [Cozmo robot](#)
- Mobile device with the [Cozmo app](#)
- (Ubuntu) Linux machine with 2 Wi-Fi interfaces (e.g. a Raspberry Pi)

- The following tools installed:
  - wireless-tools
  - wpa\_supplicant
  - hostapd
  - dnsmasq
  - tcpdump

## 4.3 Connecting to Cozmo

Ensure that wireless tools and wpa\_supplicant are installed.

```
$ sudo apt-get install wireless-tools wpasupplicant
```

Wake up Cozmo but placing it on the charging platform.

Make Cozmo display it's Wi-Fi PSK key by rising and lowering its lift.

Get Cozmo's Wi-Fi SSID by scanning for Wi-Fi devices:

```
$ sudo iwlist wlan0 scan
wlan0      Scan completed :
           Cell 01 - Address: 5E:CF:7F:XX:XX:XX
                        ESSID:"Cozmo_XXXXXX"
                        Protocol:IEEE 802.11bg
                        Mode:Master
                        Frequency:2.412 GHz (Channel 1)
                        Encryption key:on
                        Bit Rates:54 Mb/s
                        Extra:rsn_ie=30180100000fac020200000fac04000fac020100000fac020000
                        IE: IEEE 802.11i/WPA2 Version 1
                               Group Cipher : TKIP
                               Pairwise Ciphers (2) : CCMP TKIP
                               Authentication Suites (1) : PSK
                        Quality=100/100  Signal level=100/100
```

Open wpa\_supplicant's [configuration file](#):

```
$ sudo vi /etc/wpa_supplicant/wpa_supplicant.conf
```

Configure wpa\_supplicant to automatically connect to Cozmo by adding the following:

```
network={
    ssid="Cozmo_XXXXXX"
    psk="XXXXXXXXXXXXX"
}
```

Load the new configuration (or reboot):

```
$ sudo wpa_cli -i wlan0 reconfigure
OK
```

At this point the Linux machine should be associated with Cozmo:

```
$ ip addr
...
3: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default_
  qlen 1000
    link/ether 80:1f:02:XX:XX:XX brd ff:ff:ff:ff:ff:ff
    inet 172.31.1.172/24 brd 172.31.1.255 scope global wlan0
        valid_lft forever preferred_lft forever
    inet6 fe80::1d4b:9d3b:c6f0:f5b1/64 scope link
        valid_lft forever preferred_lft forever
```

Cozmo should respond to ping:

```
$ ping 172.31.1.1
PING 172.31.1.1 (172.31.1.1) 56(84) bytes of data.
64 bytes from 172.31.1.1: icmp_seq=1 ttl=128 time=1.94 ms
64 bytes from 172.31.1.1: icmp_seq=2 ttl=128 time=2.28 ms
...
```

## 4.4 Masquerading as a Cozmo

Install hostapd and dnsmasq:

```
$ sudo apt-get install hostapd dnsmasq
```

Edit dhcpd's configuration file:

```
$ sudo vi /etc/dhcpd.conf
```

Disable wpa\_supplicant on wlan1 and configure a static IP address by adding the following:

```
interface wlan1
nohook wpa_supplicant
static ip_address=192.168.50.1/24
```

Edit dnsmasq's configuration file:

```
$ sudo vi /etc/dnsmasq.conf
```

Configure DHCP on wlan1 by adding the following:

```
interface=wlan1
dhcp-range=192.168.50.50,192.168.50.100,255.255.255.0,24h
```

Restart dnsmasq

```
$ sudo systemctl start dnsmasq
```

Create a configuration file for hostapd:

```
$ sudo vi /etc/hostapd/hostapd.conf
```

Configure a Wi-Fi AP with WPA2 PSK on wlan1 by adding the following:

```
interface=wlan1
hw_mode=g
channel=1
wmm_enabled=0
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
ssid=Cozmo_111111
wpa_passphrase=XXXXXXXXXXXX
```

The SSID should be different from Cozmo's SSID and should follow the form `Cozmo_XXXXXX`, where `XXXXXX` are upper-case hexadecimal digits as this is what the Cozmo app looks for.

The passphrase should consist of exactly 12 upper-case hexadecimal digits as this is what the Cozmo app expects.

Edit `/etc/default/hostapd`:

```
$ sudo vi /etc/default/hostapd
```

Configure the newly created configuration file:

```
DAEMON_CONF="/etc/hostapd/hostapd.conf"
```

Enable and start hostapd:

```
$ sudo systemctl unmask hostapd
$ sudo systemctl enable hostapd
$ sudo systemctl start hostapd
```

Ensure that IP forwarding is enabled on boot:

```
$ sudo vi /etc/sysctl.conf
```

The following line should be uncommented:

```
net.ipv4.ip_forward=1
```

Ensure that IP forwarding is enabled:

```
$ sudo sysctl net.ipv4.ip_forward=1
```

The Cozmo app always tries to communicate with Cozmo using the IP address `172.31.1.1`.

Configure masquerading on `wlan0` so that packets, coming from the Cozmo app, with source IP in the range `192.168.50.0/24`, reach Cozmo with the `wlan0` IP address of the Linux machine.

```
$ sudo iptables -t nat -A POSTROUTING -o wlan0 -j MASQUERADE
```

This is necessary, because Cozmo only responds to UDP packets with source IP address in the range `172.31.1.0/24`.

## 4.5 Capturing Communication

Ensure that `tcpdump` is installed:

```
$ sudo apt-get install tcpdump
```

At this point, it should be possible to capture Cozmo communication using tcpdump:

```
$ sudo tcpdump -i wlan0 -w cozmo.pcap
```

Connect to cozmo from the app. The app should find at least 2 Cozmos (one being the masqueraded Linux machine) and a selection screen should show up.

The captured PCAP file can be analyzed with [Wireshark](#) or with `pycozmo_dump.py`.





### 5.1 Overview

Cozmo is a complex distributed embedded system with the following main parts:

- robot
- cubes
- charging platform

The robot can be subdivided into:

- head
  - Wi-Fi communication controller ([Espressif ESP8266](#))
  - Real-time and Image Processing (RTIP) controller ([NXP Kinetis K02](#))
- body
  - Body controller ([Nordic nRF51822](#))

The Wi-Fi communication controller is responsible for the following functions:

- Wi-Fi communication
- over-the-air (OTA) firmware updates
- NV RAM storage

Once Cozmo is powered on, the communications controller remains always powered on to maintain Wi-Fi communication.

On connection, the robot transmits its serial number with the `HardwareInfo` message and firmware version with the `FirmwareSignature` message.

The RTIP controller is responsible for:

- OLED display image decoding

- speaker audio decoding
- camera image encoding
- accelerometers
- gyro

The body controller is in charge of:

- left and right tread motors and encoders encoders
- head motor and encoder
- lift motor and encoder
- backpack LEDs
- backpack button (on newer models only)
- Bluetooth LE communication (to cubes and charging platform)
- IR LED
- cliff sensor
- batter charging

The body is powered on with the `Enable` message. The `BodyInfo` message communicates the body hardware version, serial number, and color.

Cubes use Nordic nRF31512 MCU. They are communicated with over Bluetooth LE and provide access to:

- LEDs
- Accelerometers
- Battery voltage

Some charging platforms (aka “pads”) can be communicated with over Bluetooth LE. They contains 3 RGB LEDs that can be controlled, similar to cube LEDs.

The following sections provide more details on the use of each function.

## 5.2 Wi-Fi

Wi-Fi is activated automatically when the head board is powered on. The robot operates in access point (AP) mode.

`cozmoclad` defines a `SetBodyRadioMode` message that seems to allow changing the Wi-Fi channel but it is unclear how it can be used with the Cozmo protocol.

`WifiOff Shutdown`

## 5.3 Backpack LEDs

The 5 Backpack LEDs can be set controlled with 2 messages:

- `lightStateCenter` - controls the top, middle, and bottom RGB LEDs.
- `LightStateSide` - controls the left and right red-only LEDs.

Each color is defined by a 5-bit value for a total of 32768 colors.

See `examples/backpack_lights.py` for example usage.

## 5.4 Backpack Button

v1.5 and newer Cozmo models have a backpack button.

Button press and release events are communicated by the `ButtonPressed` message. It is immediately available on connection and does not require `Enable` to be used.

The `RobotState` message has a `backpack_touch_sensor_raw` field but it seems that it's value does not change as a result of button presses.

See `examples/events.py` for example usage.

## 5.5 Wheels

The left and the right motor speeds can be controlled directly using the `DriveWheels` and `TurnInPlaceAtSpeed` messages. The motors can be stopped using the `StopAllMotors` message.

The actual speed of wheels is measured with Hall magnetic sensors. The values for each wheel can be read through the `lwheel_speed_mmpps` and `rwheel_speed_mmpps` fields of the `RobotState` message.

In addition, the `TurnInPlace` message can be used to turn to a specific angle.

## 5.6 Localization

The robot maintains a world frame internally. It's position and orientation with respect to it are transmitted every 30 ms or about 33 times per second with the `RobotState` message.

If the robot is unable to maintain correct position and orientation, for example because it is picked up or pushed, it will communicate this with a `RobotDelocalized` message.

The origin (0,0,0) of the world frame as well as “pose ID” can be set with the `SetOrigin` message. This is usually done on initial connection and on receiving a `RobotDelocalized` message.

The timestamp in `RobotState` messages can be synchronized using the `SyncTime` message.

## 5.7 Path Tracking

The robot can traverse paths, composed of lines, arcs, and turns in place, described in world frame coordinates. The `AppendPathSegLine`, `AppendPathSegArc`, and `AppendPathSegPointTurn` messages can be used to build paths.

The last composed path can be executed using the `ExecutePath` message. One of it's arguments can be used to request the reception of `PathFollowingEvent` message when path traversing finishes.

The `status` field of the `RobotState` message has a `robot_pathing` flag that indicates whether the robot is currently traversing a path. The `curr_path_segment` field indicates which segment is being traversed.

The `ClearPath` message can be used to destroy an already composed path. The `TrimPath` message can be used to delete path segments from the beginning or the end of a composed path.

See `examples/path.py` and `examples/go_to_pose.py` for example usage.

## 5.8 Head

The head motor can be controlled directly, using the `DriveHead` and `SetHeadAngle` messages. `SetHeadAngle` is always followed by an `AcknowledgeAction` message before the head starts moving.

The actual head angle can be read through the `head_angle_rad` field of the `RobotState` message. The `head_in_pos` flag of the `status` field indicates whether the head is in position or in motion.

The motor can be stopped using the `StopAllMotors` message.

The robot measures the angle of the head, relative to its lowest possible position. This measurement is automatically triggered on connection. The head can be forced to an unknown angle for example as a result of a fall. In such situations, the robot recalibrates the head motor automatically. Calibration can also be triggered on request, using the `StartMotorCalibration` message. The `MotorCalibration` message indicates whether calibration is in progress.

See `examples/extremes.py` for example usage.

## 5.9 Lift

The head motor can be controlled directly, using the `DriveLift` and `SetLiftHeight` messages. `SetLiftHeight` is always followed by an `AcknowledgeAction` message before the lift start moving.

The actual lift height can be read through the `lift_height_mm` field of the `RobotState` message. The `lift_inpos` flag of the `status` field indicates whether the lift is in position or in motion.

The motor can be stopped using the `StopAllMotors` message.

The robot measures the angle of the lift, relative to its lowest possible position. It is calibrated similar to the head motor.

See `examples/extremes.py` for example usage.

## 5.10 OLED display

Images can be displayed on the robot's OLED 128x64 display using the `DisplayImage` message. To reduce display burn-in, consecutive images are interleaved and only half of the display's rows can be used at a time and the effective display resolution is 128x32.

The Cozmo protocol uses a special run-length encoding to compress images.

Display and audio are synchronized by audio messages (`OutputAudio` and `OutputSilence`).

`AnimationState` message which can be enabled using the `EnableAnimationState` message provide statistics on display usage.

See `examples/display_image.py` and `examples/display_lines.py` for example usage.

## 5.11 Speaker

The `OutputAudio` message can be used to transmit 744 audio samples at a time. The samples are 8-bit and u-law encoded.

Speaker volume can be adjusted with the `SetRobotVolume` message.

`AnimationState` message which can be enabled using the `EnableAnimationState` message provide statistics on audio usage.

See `examples/audio.py` for example usage.

## 5.12 Camera

Cozmo can send a stream of camera images in 320x240 (QVGA) resolution at a rate of ~15 frames per second.

The `EnableCamera` message enables camera image reception and the `EnableColorImages` message allows switching between grayscale and color images.

The camera gain, exposure time, and auto exposure can be controlled with the `SetCameraParams` message.

Images are encoded in JPEG format and transmitted as a series of `ImageChunk` messages. The header of the JPEG files is not transmitted to save bandwidth.

The `ImageImuData` message provides accelerometer readings at the time of capturing every image to allow for motion blur compensation.

See `examples/camera.py` for example usage.

## 5.13 IR LED

The IR LED (aka head light) can improve the camera performance in dark environments.

The IR LED can be turned on and off using the `SetHeadLight` message.

## 5.14 Accelerometers

The `RobotState` message communicates accelerometer readings which represent acceleration along the x, y, and z axes.

In addition, the robot automatically detects and communicates 2 types of events. The `RobotPoked` message is sent if the robot has been moved rapidly by an external force along the x or y axes. The `FallingStarted` and `FallingStopped` messages are sent if the robot is moving rapidly along the z axis.

See `examples/events.py` for example usage.

## 5.15 Gyro

The `RobotState` message communicates gyro readings which represent angular velocity around the x, y, and z axes.

See `examples/events.py` for example usage.

## 5.16 Cliff Sensor

The robot has a “cliff sensor” that measures the distance to ground below the robot. This allows detecting cliffs and detecting when the robot is being picked up or put down.

The `RobotState` message communicates the raw cliff sensor readings.

In addition, the robot can be made to automatically stop when a cliff is detected with the `EnableStopOnCliff` message.

See `examples/events.py` for example usage.

## 5.17 Battery voltage

The `RobotState` message communicates raw battery voltage readings.

## 5.18 NV RAM Storage

The robot provides access to some amount of non-volatile memory (aka NV RAM) intended to store two main types of data:

- unit-specific parameters (ex. camera calibration data and cube IDs)
- mobile app data (ex. sparks and unlocked games and tricks)

The NV RAM storage is backed by the head's ESP8266 controller external SPI flash. It is a NOR flash which drives the following specifics for its use:

- an erase operation is needed before a write operation
- data is erased in pages

The `NvStorageOp` message allows performing read, erase, and write operations. Data is addressed by the `tag` field and only the values enumerated by `NvEntryTag` can be used. Using any other address results in a `NV_BAD_ARGS`. Tags smaller than `0x80000000` are direct NOT flash memory addresses. Tags larger than `0x80000000` are virtual addresses that seem to be stored in the `NVEntry_FactoryBaseTagWithBCOffset` area.

`NvStorageOpResult` messages communicate results of `NvStorageOp` operations.

A backup through the mobile app, preserves the data behind the following keys:

- `NVEntry_GameSkillLevels`
- `NVEntry_Onboarding`
- `NVEntry_GameUnlocks`
- `NVEntry_FaceEnrollData`
- `NVEntry_FaceAlbumData`
- `NVEntry_NurtureGameData`
- `NVEntry_InventoryData`
- `NVEntry_LabAssignments`

See `examples/nvram.py` for example usage.

## 5.19 Firmware Updates

Cozmo firmware updates are distributed in “cozmo.safe” files that seem to contain firmware images for all three of Cozmo's controllers - the Wi-Fi communication controller, the RTIP controller, and the body controller.

The “cozmo.safe” files start with a firmware signature in JSON format:

```
{
  "version": 2381,
  "git-rev": "408d28a7f6e68cbb5b29c1dcd8c8db2b38f9c8ce",
  "date": "Tue Jan 8 10:27:05 2019",
  "time": 1546972025,
  "messageEngineToRobotHash": "9e4a965ace4e09d86997b87ba14235d5",
  "messageRobotToEngineHash": "a259247f16231db440957215baba12ab",
  "build": "DEVELOPMENT",
  "wifiSig": "69ca03352e42143d340f0f7fac02ed8ff96ef10b",
  "rtipSig": "36574986d76144a70e9252ab633be4617a4bc661",
  "bodySig": "695b59eff43664acd1a5a956d08c682b3f8bd2c8"
}
```

This is the same signature, delivered with the `FirmwareSignature` message on initial connection establishment.

See `docs/versions.md` for more examples.

There seem to be individual signatures for each controller but the structure of the `cozmo.safe` files is not known.

The firmware image is transferred as-is from the engine to the robot, using `FirmwareUpdate` messages. It is divided into 1024 B chunks that are numbered consecutively, starting with 0. Each chunk is confirmed by the robot with a `FirmwareUpdateResult` message with `status` field set to 0.

Firmware transfer completion is indicated by the engine with a `FirmwareUpdate` message with chunk ID set to 0xFFFF and data set to all-zeros. The robot confirms firmware update completion by sending a `FirmwareUpdateResult` message that repeats the last chunk ID and has a `status` field set to 10.

## 5.20 Bluetooth LE

“Objects”, that can be connected to over Bluetooth LE announce their availability with an `ObjectAvailable` message periodically. The `ObjectAvailable` message contains the object type (e.g. light cube 1, 2, 3 or charging pad) and the object factory ID which identifies it uniquely.

The `ObjectConnect` message is used to initiate or terminate a connection to objects, using their factory ID.

Connection establishment and termination is announced with the `ObjectConnectionState` message. It contains a temporary “object ID” that is used to identify the object for the duration of the connection with it.

## 5.21 Cube LEDs

Cubes have 4 RGB LEDs that can be controlled individually.

A cube has to be “selected” first, using the `CubeId` message. A subsequent `CubeLights` message sets the state of all 4 cube LEDs.

Cubes can be programmed to perform simple LED light animations autonomously using the `LightState` structure and the `CubeId.rotation_period_frames` field.

See `examples/cube_lights.py` and `examples/cube_light_animation.py` for example usage.

## 5.22 Cube Battery Voltage

Cube battery voltage is communicated periodically with `ObjectPowerLevel` messages.

## 5.23 Cube Accelerometers

Cube accelerometer value reception can be enabled with the `StreamObjectAccel` message and are communicated every 30 ms with the `ObjectAccel` message.

In addition, the robot performs basic cube accelerometer data processing and provides basic events with the following messages:

- `ObjectMoved`
- `ObjectStoppedMoving`
- `ObjectUpAxisChanged`
- `ObjectTapped`
- `ObjectTapFiltered`

## 5.24 Animations

To play animations, `AnimationState` message have to be enabled first using the `EnableAnimationState` message.

Animations are controlled with the `StartAnimation`, `EndAnimation`, and `AbortAnimation` messages.

Keyframes are transferred with the `AnimHead`, `AnimLift`, `AnimBody`, `AnimBackpackLights`, `RecordHeading`, `TurnToRecordedHeading`, and `OutputAudio` messages.

See `examples/anim.py` for example usage.



---

## Cozmo Off-Board Functions

---

Cozmo mobile application resources consist of:

- audio files
- animations
- animation group descriptions
- behaviors
- reaction triggers
- emotions
- activities
- text-to-speech models

Robot firmware images are also distributed as part of the app resources.

### 6.1 Directory structure

```
cozmo_resources/  
  assets/  
    animationGroupMaps/  
    animationGroups/  
    animations/  
    cubeAnimationGroupMaps/  
    faceAnimations/  
    RewardedActions/  
  config/  
    engine/  
      animations/  
      behaviorSystem/  
      activities/
```

(continues on next page)

(continued from previous page)

```
        behaviors/  
        emotionevents/  
        firmware/  
        lights/  
            backpackLights/  
            cubeLights/  
sound/  
    English (US)  
tts/
```

## 6.2 Audio files

### 6.2.1 WEM files

### 6.2.2 BNK files

## 6.3 Animations

Cozmo “animations” allow animating the following aspects of the robot:

- body movement
- lift movement
- head movement
- face images
- backpack LED animations
- audio

Cozmo animations are series of keyframes, stored in binary files in [FlatBuffers](#) format. Animation data structures are declared in FlatBuffers format in `files/cozmo/cozmo_resources/config/cozmo_anim.fbs`. The animation files are available in the following directory of the Android mobile application:

```
files/cozmo/cozmo_resources/assets/animations
```

Face images are generated procedurally. They are described by 43 parameters - 5 for the face and 19 for each eye. The face as a whole can be translated, scaled, and rotated. Each individual eye can be translated, scaled, and rotated. The 4 corners of each eye can be controlled and each eye has a lower and upper lid.

The following presentation from Anki provides some background information on Cozmo animations:

[Cozmo: Animation pipeline for a physical robot](#)

## 6.4 Animation groups

Animation groups are sets of animations with the same purpose.

## 6.5 Behaviors

Behaviors can be thought of as small applications that perform a specific function using the robot client API.

## 6.6 Reactions

Reactions map robot events to behaviors.

## 6.7 Emotions

Emotions are modeled as value functions that change in one of the following ways:

- over time, driven by a decay function
- as a result of reactions
- as a result of behaviors

## 6.8 Activities

Activities are sets of behaviors with a rule how to choose



---

## Cozmo Firmware Versions

---

Cozmo firmware images can be found under `com.anki.cozmo/files/cozmo/cozmo_resources/config/engine/firmware` in the Cozmo app.

### 7.1 Production Versions

```
{
  "version": 2381,
  "git-rev": "408d28a7f6e68cbb5b29c1dcd8c8db2b38f9c8ce",
  "date": "Tue Jan  8 10:27:05 2019",
  "time": 1546972025,
  "messageEngineToRobotHash": "9e4a965ace4e09d86997b87ba14235d5",
  "messageRobotToEngineHash": "a259247f16231db440957215baba12ab",
  "build": "DEVELOPMENT",
  "wifiSig": "69ca03352e42143d340f0f7fac02ed8ff96ef10b",
  "rtipSig": "36574986d76144a70e9252ab633be4617a4bc661",
  "bodySig": "695b59eff43664acd1a5a956d08c682b3f8bd2c8"
}
```

```
{
  "version": 2380,
  "git-rev": "6ef227df0d64427f95cb943e01d8ac3956646e4d",
  "date": "Thu Dec 20 17:33:45 2018",
  "time": 1545356025,
  "messageEngineToRobotHash": "3aed3b94dbf19e11b2775ff980874213",
  "messageRobotToEngineHash": "c5a95cb6f44c1b89a42784d0c637fda8",
  "build": "DEVELOPMENT",
  "wifiSig": "8694122d7de45ee085c488274d28b69b7b1f2f44",
  "rtipSig": "8acba259c7b440dc0a3467ae73f262a224f036db",
  "bodySig": "14d4420c42432211ae4cda4f78a41841b03a6b40"
}
```

```
{
  "version": 2315,
  "git-rev": "d96caf034da1c4a33d70d2c1e3bc5732ec68594a",
  "date": "Thu Nov 9 15:37:45 2017",
  "time": 1510270665,
  "messageEngineToRobotHash": "5d963ecd52d4ae18af796f14f02a3f60",
  "messageRobotToEngineHash": "d07d1f4dea884725adefd33de221a49f",
  "build": "DEVELOPMENT",
  "wifiSig": "2749d9fb97a138aa7b56429c3a587baf6dadfb6f",
  "rtipSig": "0605ff5cd1f37cf75573caac3678ecba12b9bebe",
  "bodySig": "76dc76aa624fac230603101206d3a4e2e50e76cb"
}
```

```
{
  "version": 2313,
  "git-rev": "7381fe56705992ffd03bef1bb1a7b2e7258e9bc2",
  "date": "Tue Nov 7 21:13:04 2017",
  "time": 1510117984,
  "messageEngineToRobotHash": "838bbe94628fd10783e40f6b6b9874df",
  "messageRobotToEngineHash": "6ae9b7733e469f4fef89479d63e214ba",
  "build": "DEVELOPMENT",
  "wifiSig": "5bfbabc73e0ec5e20a072b6ab87b60da8a51310a",
  "rtipSig": "349d2224cc00e56ee50a5b4ecb905a5ba64c791d",
  "bodySig": "5ac6821655294e88b5fb852427bd99120af16fb5"
}
```

```
{
  "version": 2214,
  "git-rev": "c363ccc897bc3748d234f80c21e4c8a33757d063",
  "date": "Wed Aug 9 11:01:32 2017",
  "time": 1502301692,
  "messageEngineToRobotHash": "861bbc71828456c0f073c4464bdcb21e",
  "messageRobotToEngineHash": "2dc8419f768f6f3fd4843cbb0a86f7f7",
  "build": "DEVELOPMENT",
  "wifiSig": "da7eb444c13475eb67b0c13336b24021b8cf540f",
  "rtipSig": "4cba42517073e77967ce8c7340376713001b4d0a",
  "bodySig": "74a1776d1c6a4213ccfbb0ad2c4099eafdf7ad0c"
}
```

```
{
  "version": 2158,
  "git-rev": "44c8d8a1d3a2b09b54da0ff4b6ceee75ec66e267",
  "date": "Thu Jun 15 10:00:23 2017",
  "time": 1497546023,
  "messageEngineToRobotHash": "71beec8d11144f3a3718d2cc5ea602f3",
  "messageRobotToEngineHash": "4018b2e764ec08f5fcacdb6358847cb0",
  "build": "DEVELOPMENT",
  "wifiSig": "e3f4a7e29b76321e3563f50e0b09c89378b5dc97",
  "rtipSig": "64efe94218e8eaac3576f2405bc5f01f020b0b7a",
  "bodySig": "d0c34ed006c71abe45ac735e4bb68bf1153b082b"
}
```

```
{
  "version": 1889,
  "git-rev": "e541e4247376d7945fd42a82a826b443effbeff2",
  "date": "Thu Mar 23 17:15:50 2017",

```

(continues on next page)

(continued from previous page)

```

    "time": 1490314550,
    "messageEngineToRobotHash": "7098b4a266c0ccc2102a61fda53b8999",
    "messageRobotToEngineHash": "9b83f21da9120fdeebfeabe84af81c32",
    "build": "DEVELOPMENT",
    "wifiSig": "266d1d4f91c5ee069e628550a0331e8b0eb90f2b",
    "rtipSig": "bc90e2949be66851fb7ac035f5de9b52ff69fd14",
    "bodySig": "ccbb209db374f21ef233945f1515a70b8fe43114"
}

```

```

{
    "version": 1859,
    "git-rev": "11a52d6a4f2c5d89cef7085b836e8d0f2525808b",
    "date": "Mon Mar 20 23:29:56 2017",
    "time": 1490077796,
    "messageEngineToRobotHash": "54195812be0de998a4ebde795364d62b",
    "messageRobotToEngineHash": "90d8f3273055624b8444fbcbe555ee8",
    "build": "DEVELOPMENT",
    "wifiSig": "79dca08e85f21311e5551e38ecf0d3dab6ce006f",
    "rtipSig": "72519cd2bfb11bc799915dd8506a67b0ae5186da",
    "bodySig": "8746362ebc89e6235e3da103b9e9c0133cc3d1c1"
}

```

```

{
    "version": 1299,
    "git-rev": "6ced81297ac14067662acbed79cecac7f5eacd28",
    "date": "Mon Nov 21 15:25:58 2016",
    "time": 1479770758,
    "messageEngineToRobotHash": "61879d8808f0308cd8ae6340ddfe06e6",
    "messageRobotToEngineHash": "5914fda0b97c7aada0e4d97fc72610f",
    "wifiSig": "6cd4d9263e7a5b5da9eedc33e32c8baeb04a0ea6",
    "rtipSig": "24591dd715955eef0c1c7f0d89b4b41c122cbb26",
    "bodySig": "412ce6fc22f7407cb2e87eaacee3e9c4d7ca47ea"
}

```

## 7.2 Factory Versions

Cozmo factory firmwares identify with large version numbers.

Seen on Cozmo with HW v1.5:

```

{
    "build": "FACTORY",
    "version": 10501,
    "date": "Fri Apr 14 20:28:21 2017",
    "time": 1492201868
}

```

```

{
    "build": "FACTORY",
    "version": 10502,
    "date": "Mon Aug 7 09:21:24 2017",
    "time": 1502122884
}

```

Seen on development Cozmo with HW v1.7:

```
{  
  "build": "FACTORY",  
  "version": 10700,  
  "date": "Thu Mar 28 14:18:13 2019",  
  "time": 1553807893  
}
```



## Cozmo Hardware Versions

### 8.1 Hardware Version 4

- fall 2016
- does not have a button
- come with platforms with LEDs?

```
2020-09-23 19:12:56.567 pycozmo.general INFO Firmware version 2381.
2020-09-23 19:12:56.568 pycozmo.robot INFO hardware.revision: Hardware 1.0
2020-09-23 19:12:56.598 pycozmo.general INFO Body S/N 0x088xxxxx, HW version_
↪4, color 0.
```

### 8.2 Hardware Version 5

- fall 2017
- has an off button (EU Certification)
- observed to have factory firmware v10501
- teardown - <https://www.microcontrollertips.com/teardown-anki-cozmo-vector/>

```
2020-09-26 12:31:32.421 pycozmo.general INFO Firmware version 2381.
2020-09-26 12:31:32.422 pycozmo.robot INFO hardware.revision: Hardware 1.5
2020-09-26 12:31:32.453 pycozmo.general INFO Body S/N 0x088xxxxx, HW version_
↪5, color 3.
```

### 8.3 Hardware Version 6

- fall 2018

- has an off button (Japan certification)

## 8.4 Hardware Version 7

- fall 2019
- has an on/off button
- observed with development units
- observed to have factory firmware v10700
- observed to report undocumented color “5”

```
2020-09-24 20:04:35.823 pycozmo.general INFO Firmware version 10700.
2020-09-24 20:04:35.831 pycozmo.robot INFO hardware.revision: Hardware 1.7
2020-09-24 20:04:35.856 pycozmo.general INFO Body S/N 0x088xxxxx, HW version_
↪7, color 5.
```

The ESP8266 is the main Cozmo controller, responsible for Wi-Fi communication.

## 9.1 SPI Flash Memory Map

The SPI flash size is 2 MB.

The below memory map has been reconstructed based on a SPI flash memory dump and NvEntryTag values.

Offset	Length	Type	Description
0x00000000	0x00001000	Code	Bootloader.
0x00001000	0x00001000	Data	Unknown. The first 4 bytes are the head serial number.
0x00002000	0x00001000	Data	Unknown.
0x00003000	0x0007b800	Code	Application image 1.
0x0007e800	0x00001800	Data	Application image 1 signature. See versions.md .
0x00080000	0x0005e000	Code	Recovery image / factory firmware.
0x000de000	0x00000030	Data	Birth certificate.
0x000de030	0x00021fd0	Data	Factory data.
0x00100000	0x00003000	Data	Unknown.
0x00103000	0x0007b800	Code	Application image 2
0x0017e800	0x00001800	Data	Application image 2 signature. See versions.md .
0x00180000	0x00018000	Data	Application data.
0x00198000	0x00028000	Data	Empty.
0x001c0000	0x0001e000	Data	Factory reserved 1.

(continues on next page)

(continued from previous page)

0x001de000	0x0001e000	Data	Factory reserved 2. Empty?
0x001fc000	0x00001000	Data	Unknown.
0x001fd000	0x00001000	Data	Wi-Fi configuration 1.
0x001fe000	0x00001000	Data	Wi-Fi configuration 2.
0x001ff000	0x00001000	Data	Unknown.

## CHAPTER 10

---

### pycozmo package

---

<code>pycozmo.audiokinetic.exception</code>	AudioKinetic Wwise exceptions.
<code>pycozmo.audiokinetic.soundbank</code>	AudioKinetic Wwise SoundBank representation and reading.
<code>pycozmo.audiokinetic.soundbanksinfo</code>	AudioKinetic Wwise SoundbanksInfo.xml representation and reading.
<code>pycozmo.audiokinetic.wem</code>	AudioKinetic Wwise WEM file representation and reading.
<code>pycozmo.expressions.expressions</code>	Facial expression definitions.
<code>pycozmo.activity</code>	Activity representation and reading.
<code>pycozmo.anim</code>	Animation clip representation, reading, and preprocessing.
<code>pycozmo.anim_controller</code>	Animation controller for audio, image, and animation playback.
<code>pycozmo.anim_encoder</code>	Reading and writing of Cozmo animations in FlatBuffers (.bin) and JSON format.
<code>pycozmo.audio</code>	Cozmo audio encoding.
<code>pycozmo.behavior</code>	Behavior representation and reading.
<code>pycozmo.brain</code>	Brain class - high level behavior and emotion engine.
<code>pycozmo.camera</code>	Camera image decoding.
<code>pycozmo.client</code>	Cozmo protocol client and high-level API.
<code>pycozmo.conn</code>	Cozmo protocol low-level client and server connection.
<code>pycozmo.emotions</code>	Emotion representation and reading.
<code>pycozmo.event</code>	Event declaration and dispatching.
<code>pycozmo.exception</code>	Exception declarations.
<code>pycozmo.filter</code>	ID filtering for logging.
<code>pycozmo.frame</code>	Cozmo protocol frame representation and encoding and decoding.
<code>pycozmo.image_encoder</code>	Cozmo image run-length encoding and decoding.
<code>pycozmo.lights</code>	Helper routines for working with colors and lights.
<code>pycozmo.logging</code>	

Continued on next page

Table 1 – continued from previous page

<code>pycozmo.object</code>	Cozmo objects (cubes, platforms, etc.).
<code>pycozmo.procedural_face</code>	Cozmo procedural face rendering.
<code>pycozmo.protocol_ast</code>	Cozmo protocol abstract syntax tree (AST) types.
<code>pycozmo.protocol_base</code>	Cozmo protocol implementation base.
<code>pycozmo.protocol_declaration</code>	Cozmo protocol abstract syntax tree (AST) declaration.
<code>pycozmo.protocol_encoder</code>	Cozmo protocol packet encoder classes, based on protocol version 2381.
<code>pycozmo.protocol_generator</code>	Cozmo protocol packet encoder code generator.
<code>pycozmo.protocol_utils</code>	Cozmo protocol encoding helper classes and functions.
<code>pycozmo.robot</code>	Robot constants and helper code.
<code>pycozmo.robot_debug</code>	Cozmo firmware debug message decoding.
<code>pycozmo.run</code>	Helper functions for running PyCozmo applications.
<code>pycozmo.util</code>	Utility classes and functions.
<code>pycozmo.window</code>	Cozmo protocol sliding window implementation.

## 10.1 pycozmo.audiokinetic.exception

AudioKinetic Wwise exceptions.

### Exceptions

<code>AudioKineticBaseError</code>	AudioKinetic Wwise base error.
<code>AudioKineticFormatError</code>	Invalid file format error.
<code>AudioKineticIOError</code>	File I/O error.

**exception** `pycozmo.audiokinetic.exception.AudioKineticBaseError`

Bases: `pycozmo.exception.PyCozmoException`

AudioKinetic Wwise base error.

**args**

**with\_traceback()**

Exception.with\_traceback(tb) – set self.\_\_traceback\_\_ to tb and return self.

**exception** `pycozmo.audiokinetic.exception.AudioKineticFormatError`

Bases: `pycozmo.audiokinetic.exception.AudioKineticBaseError`

Invalid file format error.

**args**

**with\_traceback()**

Exception.with\_traceback(tb) – set self.\_\_traceback\_\_ to tb and return self.

**exception** `pycozmo.audiokinetic.exception.AudioKineticIOError`

Bases: `pycozmo.audiokinetic.exception.AudioKineticBaseError`

File I/O error.

**args**

**with\_traceback()**

Exception.with\_traceback(tb) – set self.\_\_traceback\_\_ to tb and return self.

## 10.2 pycozmo.audiokinetic.soundbank

AudioKinetic Wwise SoundBank representation and reading.

### References:

- [http://wiki.xentax.com/index.php/Wwise\\_SoundBank\\_\(\\*.bnk\)](http://wiki.xentax.com/index.php/Wwise_SoundBank_(*.bnk))
- <https://github.com/rickvg/Wwise-BNKExtract>

### Classes

<i>Event</i> (soundbank_id, event_id, name, action_ids)	AudioKinetic Wwise Event.
<i>EventAction</i> (soundbank_id, ea_id, scope, ...)	AudioKinetic Wwise Event Action.
<i>File</i> (soundbank_id, file_id, offset, length)	AudioKinetic Wwise WEM File.
<i>SFX</i> (soundbank_id, sfx_id, name, location, ...)	AudioKinetic Wwise sound effect/voice.
<i>SoundBank</i> ()	AudioKinetic Wwise SoundBank (.bnk) file representation class.
<i>SoundBankReader</i> (soundbankinfo, Any[])	

```
class pycozmo.audiokinetic.soundbank.Event (soundbank_id: int, event_id: int, name: str,
                                             action_ids: Iterable[int])
```

Bases: `object`

AudioKinetic Wwise Event.

**action\_ids**

**id**

**name**

**soundbank\_id**

```
class pycozmo.audiokinetic.soundbank.EventAction (soundbank_id: int, ea_id: int, scope:
                                                    int, ea_type: int, reference_id: int)
```

Bases: `object`

AudioKinetic Wwise Event Action.

**id**

**reference\_id**

**scope**

**soundbank\_id**

**type**

```
class pycozmo.audiokinetic.soundbank.File (soundbank_id: int, file_id: int, offset: int,
                                             length: int)
```

Bases: `object`

AudioKinetic Wwise WEM File.

**id**

**length**

**offset**

```
    soundbank_id

class pycozmo.audiokinetic.soundbank.SFX(soundbank_id: int, sfx_id: int, name: str, loca-
                                           tion: int, file_id: int, length: int, sfx_type: int)
    Bases: object
    AudioKinetic Wwise sound effect/voice.

    file_id
    id
    length
    location
    name
    soundbank_id
    type

class pycozmo.audiokinetic.soundbank.SoundBank
    Bases: object
    AudioKinetic Wwise SoundBank (.bnk) file representation class.

    data_offset
    fspec
    id
    name
    objs
    version

class pycozmo.audiokinetic.soundbank.SoundBankReader(soundbankinfo: Dict[int, Any])
    Bases: object

    load(fspec: str) → pycozmo.audiokinetic.soundbank.SoundBank
        Load a SoundBank .bnk file and return a SoundBank object.

    load_file(f: BinaryIO, fspec: str) → pycozmo.audiokinetic.soundbank.SoundBank
        Load a SoundBank .bnk file object and return a SoundBank object.
```

## 10.3 pycozmo.audiokinetic.soundbanksinfo

AudioKinetic Wwise SoundbanksInfo.xml representation and reading.

See `assets/cozmo_resources/sound/SoundbanksInfo.xml`

### Functions

---

<code>load_soundbanksinfo(fspect, TextIO)</code>	Load SoundbanksInfo.xml and return a dictionary of parsed Info objects.
--	--

---



## Classes

<i>EventInfo</i> (soundbank_id, event_id, name, ...)	Event representation in SoundbanksInfo.xml .
<i>FileInfo</i> (soundbank_id, file_id, name, path, ...)	File representation in SoundbanksInfo.xml .
<i>SoundBankInfo</i> (soundbank_id, name, path, ...)	SoundBank representation in SoundbanksInfo.xml .

```
class pycozmo.audiokinetic.soundbanksinfo.EventInfo(soundbank_id: int, event_id: int,
                                                    name: str, object_path: str)
```

Bases: `object`

Event representation in SoundbanksInfo.xml .

**id**

**name**

**object\_path**

**soundbank\_id**

```
class pycozmo.audiokinetic.soundbanksinfo.FileInfo(soundbank_id: int, file_id: int,
                                                    name: str, path: str, embedded:
                                                    bool, prefetch_size: int)
```

Bases: `object`

File representation in SoundbanksInfo.xml .

**embedded**

**id**

**name**

**path**

**prefetch\_size**

**soundbank\_id**

```
class pycozmo.audiokinetic.soundbanksinfo.SoundBankInfo(soundbank_id: int, name:
                                                         str, path: str, language:
                                                         str, object_path: str)
```

Bases: `object`

SoundBank representation in SoundbanksInfo.xml .

**id**

**language**

**name**

**object\_path**

**path**

```
pycozmo.audiokinetic.soundbanksinfo.load_soundbanksinfo(fspec: Union[str, TextIO])
                                                         → Dict[int, Any]
```

Load SoundbanksInfo.xml and return a dictionary of parsed Info objects.

## 10.4 pycozmo.audiokinetic.wem

AudioKinetic Wwise WEM file representation and reading.

## 10.5 pycozmo.expressions.expressions

Facial expression definitions.

Based on the “Expressive Eyes” project by Catherine Chambers: <https://git.brl.ac.uk/ca2-chambers/expressive-eyes>

### Classes

<i>Amazement</i> (params, width, height)	
<i>Anger</i> (params, width, height)	
<i>Annoyance</i> (params, width, height)	
<i>Asleep</i> (params, width, height)	
<i>Boredom</i> (params, width, height)	
<i>Confusion</i> (params, width, height)	
<i>Despair</i> (params, width, height)	
<i>Disappointment</i> (params, width, height)	
<i>Disgust</i> (params, width, height)	
<i>Embarrassment</i> (params, width, height)	
<i>Excitement</i> (params, width, height)	
<i>Fear</i> (params, width, height)	
<i>Fury</i> (params, width, height)	aka “enragement”.
<i>Guilt</i> (params, width, height)	
<i>Happiness</i> (params, width, height)	
<i>Horror</i> (params, width, height)	
<i>Neutral</i> (params, width, height)	
<i>Pleading</i> (params, width, height)	
<i>Rejection</i> (params, width, height)	
<i>Sadness</i> (params, width, height)	
<i>Skepticism</i> (params, width, height)	
<i>Surprise</i> (params, width, height)	
<i>Suspicion</i> (params, width, height)	
<i>Tiredness</i> (params, width, height)	
<i>Vulnerability</i> (params, width, height)	

```
class pycozmo.expressions.expressions.Neutral (params: Optional[List[float]] = None,  
                                              width: int = 128, height: int = 64)
```

```
    Bases: pycozmo.procedural_face.ProceduralFace
```

```
    angle
```

```
    center_x
```

```
    center_y
```

```
    eye_height
```

```
    eye_width
```

```
    eyes
```

```

    half_eye_height
    half_eye_width
    height
    offset
    params
    render() → <module 'PIL.Image' from '/home/docs/checkouts/readthedocs.org/user_builds/pycozmo/envs/latest/lib/python3.7
        packages/PIL/Image.py'>
    scale_factor_lid_bend
    scale_factor_lid_height
    scale_x
    scale_y
    width

class pycozmo.expressions.expressions.Anger (params: Optional[List[float]] = None,
                                              width: int = 128, height: int = 64)
    Bases: pycozmo.procedural_face.ProceduralFace
    angle
    center_x
    center_y
    eye_height
    eye_width
    eyes
    half_eye_height
    half_eye_width
    height
    offset
    params
    render() → <module 'PIL.Image' from '/home/docs/checkouts/readthedocs.org/user_builds/pycozmo/envs/latest/lib/python3.7
        packages/PIL/Image.py'>
    scale_factor_lid_bend
    scale_factor_lid_height
    scale_x
    scale_y
    width

class pycozmo.expressions.expressions.Sadness (params: Optional[List[float]] = None,
                                              width: int = 128, height: int = 64)
    Bases: pycozmo.procedural_face.ProceduralFace
    angle
    center_x
    center_y

```

```
    eye_height
    eye_width
    eyes
    half_eye_height
    half_eye_width
    height
    offset
    params
    render() → <module 'PIL.Image' from '/home/docs/checkouts/readthedocs.org/user_builds/pycozmo/envs/latest/lib/python3.7
        packages/PIL/Image.py'>
    scale_factor_lid_bend
    scale_factor_lid_height
    scale_x
    scale_y
    width

class pycozmo.expressions.expressions.Happiness (params: Optional[List[float]] = None,
                                                width: int = 128, height: int = 64)
    Bases: pycozmo.procedural_face.ProceduralFace
    angle
    center_x
    center_y
    eye_height
    eye_width
    eyes
    half_eye_height
    half_eye_width
    height
    offset
    params
    render() → <module 'PIL.Image' from '/home/docs/checkouts/readthedocs.org/user_builds/pycozmo/envs/latest/lib/python3.7
        packages/PIL/Image.py'>
    scale_factor_lid_bend
    scale_factor_lid_height
    scale_x
    scale_y
    width

class pycozmo.expressions.expressions.Surprise (params: Optional[List[float]] = None,
                                                width: int = 128, height: int = 64)
    Bases: pycozmo.procedural_face.ProceduralFace
```

```

    angle
    center_x
    center_y
    eye_height
    eye_width
    eyes
    half_eye_height
    half_eye_width
    height
    offset
    params
    render() → <module 'PIL.Image' from '/home/docs/checkouts/readthedocs.org/user_builds/pycozmo/envs/latest/lib/python3.7
        packages/PIL/Image.py'>
    scale_factor_lid_bend
    scale_factor_lid_height
    scale_x
    scale_y
    width

class pycozmo.expressions.expressions.Disgust (params: Optional[List[float]] = None,
        width: int = 128, height: int = 64)
    Bases: pycozmo.procedural_face.ProceduralFace
    angle
    center_x
    center_y
    eye_height
    eye_width
    eyes
    half_eye_height
    half_eye_width
    height
    offset
    params
    render() → <module 'PIL.Image' from '/home/docs/checkouts/readthedocs.org/user_builds/pycozmo/envs/latest/lib/python3.7
        packages/PIL/Image.py'>
    scale_factor_lid_bend
    scale_factor_lid_height
    scale_x
    scale_y

```

```
width

class pycozmo.expressions.expressions.Fear (params: Optional[List[float]] = None, width:
                                         int = 128, height: int = 64)
    Bases: pycozmo.procedural_face.ProceduralFace
    angle
    center_x
    center_y
    eye_height
    eye_width
    eyes
    half_eye_height
    half_eye_width
    height
    offset
    params
    render () → <module 'PIL.Image' from '/home/docs/checkouts/readthedocs.org/user_builds/pycozmo/envs/latest/lib/python3.7
        packages/PIL/Image.py'>
    scale_factor_lid_bend
    scale_factor_lid_height
    scale_x
    scale_y
    width

class pycozmo.expressions.expressions.Pleading (params: Optional[List[float]] = None,
                                         width: int = 128, height: int = 64)
    Bases: pycozmo.procedural_face.ProceduralFace
    angle
    center_x
    center_y
    eye_height
    eye_width
    eyes
    half_eye_height
    half_eye_width
    height
    offset
    params
    render () → <module 'PIL.Image' from '/home/docs/checkouts/readthedocs.org/user_builds/pycozmo/envs/latest/lib/python3.7
        packages/PIL/Image.py'>
    scale_factor_lid_bend
```

```

    scale_factor_lid_height
    scale_x
    scale_y
    width
class pycozmo.expressions.expressions.Vulnerability(params: Optional[List[float]] =
    None, width: int = 128, height:
    int = 64)
    Bases: pycozmo.procedural_face.ProceduralFace
    angle
    center_x
    center_y
    eye_height
    eye_width
    eyes
    half_eye_height
    half_eye_width
    height
    offset
    params
    render() → <module 'PIL.Image' from '/home/docs/checkouts/readthedocs.org/user_builds/pycozmo/envs/latest/lib/python3.7
        packages/PIL/Image.py'>
    scale_factor_lid_bend
    scale_factor_lid_height
    scale_x
    scale_y
    width
class pycozmo.expressions.expressions.Despair(params: Optional[List[float]] = None,
    width: int = 128, height: int = 64)
    Bases: pycozmo.procedural_face.ProceduralFace
    angle
    center_x
    center_y
    eye_height
    eye_width
    eyes
    half_eye_height
    half_eye_width
    height
    offset

```

```
    params
    render() → <module 'PIL.Image' from '/home/docs/checkouts/readthedocs.org/user_builds/pycozmo/envs/latest/lib/python3.7
        packages/PIL/Image.py'>
    scale_factor_lid_bend
    scale_factor_lid_height
    scale_x
    scale_y
    width

class pycozmo.expressions.expressions.Guilt (params: Optional[List[float]] = None,
                                             width: int = 128, height: int = 64)
    Bases: pycozmo.procedural_face.ProceduralFace
    angle
    center_x
    center_y
    eye_height
    eye_width
    eyes
    half_eye_height
    half_eye_width
    height
    offset
    params
    render() → <module 'PIL.Image' from '/home/docs/checkouts/readthedocs.org/user_builds/pycozmo/envs/latest/lib/python3.7
        packages/PIL/Image.py'>
    scale_factor_lid_bend
    scale_factor_lid_height
    scale_x
    scale_y
    width

class pycozmo.expressions.expressions.Disappointment (params: Optional[List[float]]
                                                       = None, width: int = 128,
                                                       height: int = 64)
    Bases: pycozmo.procedural_face.ProceduralFace
    angle
    center_x
    center_y
    eye_height
    eye_width
    eyes
```



```

    half_eye_height
    half_eye_width
    height
    offset
    params
    render() → <module 'PIL.Image' from '/home/docs/checkouts/readthedocs.org/user_builds/pycozmo/envs/latest/lib/python3.7
        packages/PIL/Image.py'>
    scale_factor_lid_bend
    scale_factor_lid_height
    scale_x
    scale_y
    width

class pycozmo.expressions.expressions.Embarrassment (params: Optional[List[float]] =
    None, width: int = 128, height:
    int = 64)

    Bases: pycozmo.procedural_face.ProceduralFace
    angle
    center_x
    center_y
    eye_height
    eye_width
    eyes
    half_eye_height
    half_eye_width
    height
    offset
    params
    render() → <module 'PIL.Image' from '/home/docs/checkouts/readthedocs.org/user_builds/pycozmo/envs/latest/lib/python3.7
        packages/PIL/Image.py'>
    scale_factor_lid_bend
    scale_factor_lid_height
    scale_x
    scale_y
    width

class pycozmo.expressions.expressions.Horror (params: Optional[List[float]] = None,
    width: int = 128, height: int = 64)
    Bases: pycozmo.procedural_face.ProceduralFace
    angle
    center_x

```

```
center_y
eye_height
eye_width
eyes
half_eye_height
half_eye_width
height
offset
params
render() → <module 'PIL.Image' from '/home/docs/checkouts/readthedocs.org/user_builds/pycozmo/envs/latest/lib/python3.7
packages/PIL/Image.py'>
scale_factor_lid_bend
scale_factor_lid_height
scale_x
scale_y
width

class pycozmo.expressions.expressions.Skepticism(params: Optional[List[float]] =
None, width: int = 128, height: int =
64)

Bases: pycozmo.procedural_face.ProceduralFace
angle
center_x
center_y
eye_height
eye_width
eyes
half_eye_height
half_eye_width
height
offset
params
render() → <module 'PIL.Image' from '/home/docs/checkouts/readthedocs.org/user_builds/pycozmo/envs/latest/lib/python3.7
packages/PIL/Image.py'>
scale_factor_lid_bend
scale_factor_lid_height
scale_x
scale_y
width
```

```

class pycozmo.expressions.expressions.Annoyance (params: Optional[List[float]] = None,
                                                    width: int = 128, height: int = 64)
    Bases: pycozmo.procedural\_face.ProceduralFace
    angle
    center_x
    center_y
    eye_height
    eye_width
    eyes
    half_eye_height
    half_eye_width
    height
    offset
    params
    render () → <module 'PIL.Image' from '/home/docs/checkouts/readthedocs.org/user_builds/pycozmo/envs/latest/lib/python3.7
        packages/PIL/Image.py'>
    scale_factor_lid_bend
    scale_factor_lid_height
    scale_x
    scale_y
    width

class pycozmo.expressions.expressions.Fury (params: Optional[List[float]] = None, width:
                                                    int = 128, height: int = 64)
    Bases: pycozmo.procedural\_face.ProceduralFace
    aka "enragement".
    angle
    center_x
    center_y
    eye_height
    eye_width
    eyes
    half_eye_height
    half_eye_width
    height
    offset
    params
    render () → <module 'PIL.Image' from '/home/docs/checkouts/readthedocs.org/user_builds/pycozmo/envs/latest/lib/python3.7
        packages/PIL/Image.py'>
    scale_factor_lid_bend

```

```
    scale_factor_lid_height
    scale_x
    scale_y
    width

class pycozmo.expressions.expressions.Suspicion (params: Optional[List[float]] = None,
                                                  width: int = 128, height: int = 64)
    Bases: pycozmo.procedural_face.ProceduralFace
    angle
    center_x
    center_y
    eye_height
    eye_width
    eyes
    half_eye_height
    half_eye_width
    height
    offset
    params
    render () → <module 'PIL.Image' from '/home/docs/checkouts/readthedocs.org/user_builds/pycozmo/envs/latest/lib/python3.7
        packages/PIL/Image.py'>
    scale_factor_lid_bend
    scale_factor_lid_height
    scale_x
    scale_y
    width

class pycozmo.expressions.expressions.Rejection (params: Optional[List[float]] = None,
                                                  width: int = 128, height: int = 64)
    Bases: pycozmo.procedural_face.ProceduralFace
    angle
    center_x
    center_y
    eye_height
    eye_width
    eyes
    half_eye_height
    half_eye_width
    height
    offset
```

```

    params
    render() → <module 'PIL.Image' from '/home/docs/checkouts/readthedocs.org/user_builds/pycozmo/envs/latest/lib/python3.7
        packages/PIL/Image.py'>
    scale_factor_lid_bend
    scale_factor_lid_height
    scale_x
    scale_y
    width

class pycozmo.expressions.expressions.Boredom(params: Optional[List[float]] = None,
                                              width: int = 128, height: int = 64)
    Bases: pycozmo.procedural_face.ProceduralFace
    angle
    center_x
    center_y
    eye_height
    eye_width
    eyes
    half_eye_height
    half_eye_width
    height
    offset
    params
    render() → <module 'PIL.Image' from '/home/docs/checkouts/readthedocs.org/user_builds/pycozmo/envs/latest/lib/python3.7
        packages/PIL/Image.py'>
    scale_factor_lid_bend
    scale_factor_lid_height
    scale_x
    scale_y
    width

class pycozmo.expressions.expressions.Tiredness(params: Optional[List[float]] = None,
                                              width: int = 128, height: int = 64)
    Bases: pycozmo.procedural_face.ProceduralFace
    angle
    center_x
    center_y
    eye_height
    eye_width
    eyes
    half_eye_height

```

```
half_eye_width
height
offset
params
render() → <module 'PIL.Image' from '/home/docs/checkouts/readthedocs.org/user_builds/pycozmo/envs/latest/lib/python3.7
packages/PIL/Image.py'>
scale_factor_lid_bend
scale_factor_lid_height
scale_x
scale_y
width

class pycozmo.expressions.expressions.Asleep(params: Optional[List[float]] = None,
                                              width: int = 128, height: int = 64)
    Bases: pycozmo.procedural_face.ProceduralFace
    angle
    center_x
    center_y
    eye_height
    eye_width
    eyes
    half_eye_height
    half_eye_width
    height
    offset
    params
    render() → <module 'PIL.Image' from '/home/docs/checkouts/readthedocs.org/user_builds/pycozmo/envs/latest/lib/python3.7
packages/PIL/Image.py'>
    scale_factor_lid_bend
    scale_factor_lid_height
    scale_x
    scale_y
    width

class pycozmo.expressions.expressions.Confusion(params: Optional[List[float]] = None,
                                                  width: int = 128, height: int = 64)
    Bases: pycozmo.procedural_face.ProceduralFace
    angle
    center_x
    center_y
    eye_height
```

`eye_width``eyes``half_eye_height``half_eye_width``height``offset``params``render()` → <module 'PIL.Image' from '/home/docs/checkouts/readthedocs.org/user\_builds/pycozmo/envs/latest/lib/python3.7/packages/PIL/Image.py'>`scale_factor_lid_bend``scale_factor_lid_height``scale_x``scale_y``width`

```
class pycozmo.expressions.expressions.Amazement (params: Optional[List[float]] = None,
                                                  width: int = 128, height: int = 64)
```

Bases: `pycozmo.procedural_face.ProceduralFace`

`angle``center_x``center_y``eye_height``eye_width``eyes``half_eye_height``half_eye_width``height``offset``params``render()` → <module 'PIL.Image' from '/home/docs/checkouts/readthedocs.org/user\_builds/pycozmo/envs/latest/lib/python3.7/packages/PIL/Image.py'>`scale_factor_lid_bend``scale_factor_lid_height``scale_x``scale_y``width`

```
class pycozmo.expressions.expressions.Excitement (params: Optional[List[float]] =
                                                  None, width: int = 128, height: int =
                                                  64)
```

Bases: `pycozmo.procedural_face.ProceduralFace`

```
angle
center_x
center_y
eye_height
eye_width
eyes
half_eye_height
half_eye_width
height
offset
params
render() → <module 'PIL.Image' from '/home/docs/checkouts/readthedocs.org/user_builds/pycozmo/envs/latest/lib/python3.7
packages/PIL/Image.py'>
scale_factor_lid_bend
scale_factor_lid_height
scale_x
scale_y
width
```

## 10.6 pycozmo.activity

Activity representation and reading.

### Functions

---

<code>from_dict(info, VT)</code>	
<code>load_activities(resource_dir)</code>	Load activity map from cozmo resources.

---

### Classes

---

<code>Activity(activity_id, activity_type, strategy)</code>	Activity representation class.
<code>BehaviorChooser(choice_type, behaviors)</code>	
<code>BehaviorsActivity(behavior_chooser, *args, ...)</code>	
<code>FeedingActivity(universal_chooser, *args, ...)</code>	
<code>FreeplayActivity(cube_only_activity, ...)</code>	
<code>NeedsActivity(behavior_chooser, *args, **kwargs)</code>	
<code>Objective(objective, behavior_id, ...)</code>	
<code>PyramidActivity(setup_chooser, ...)</code>	
<code>SocializeActivity(behavior_chooser, ...)</code>	
<code>SparkedActivity(require_spark, ...)</code>	

---

Continued on next page



Table 8 – continued from previous page

---

VoiceCommandActivity(\*args, \*\*kwargs)

---

**class** pycozmo.activity.**Activity** (activity\_id: str, activity\_type: str, strategy: str)

Bases: object

Activity representation class.

**choose** ()

**id**
**strategy**
**type**

pycozmo.activity.**load\_activities** (resource\_dir: str) → Dict[str, pycozmo.activity.Activity]

Load activity map from cozmo resources.

## 10.7 pycozmo.anim

Animation clip representation, reading, and preprocessing.

### Functions

---

load\_animation\_groups(resource\_dir)

---

load\_backpack\_light\_patterns(resource\_dir)

---

load\_cube\_animation\_groups(resource\_dir)

---

load\_trigger\_map(resource\_dir,  
map\_relative\_path)

---

### Classes

---

AnimationGroup(members)

---

AnimationGroupMember(name, weight, ...)

---

BackpackAnimation(\*args, \*\*kwargs)

---

CubeAnimation(duration, rotation\_period, ...)

---

LightAnimation(on\_colors, off\_colors, ...)

---

PreprocessedClip(keyframes, ...) Preprocessed animation clip that can be played back.

---

**class** pycozmo.anim.**PreprocessedClip** (keyframes: Optional[Dict[int,  
List[pycozmo.protocol\_base.Packet]]] = None)

Bases: object

Preprocessed animation clip that can be played back.

**classmethod** **from\_anim\_clip** (clip: pycozmo.anim\_encoder.AnimClip) → py-  
cozmo.anim.PreprocessedClip

**classmethod** **keyframe\_to\_im** (keyframe) → <module 'PIL.Image' from  
'/home/docs/checkouts/readthedocs.org/user\_builds/pycozmo/envs/latest/lib/python3.7/site-  
packages/PIL/Image.py'>

```
class pycozmo.anim.AnimationGroupMember(name: str, weight: float, cooldown_time: float,  
                                         mood: str, use_head_angle: Optional[bool] =  
                                         False, head_angle_min: Optional[float] = 0.0,  
                                         head_angle_max: Optional[float] = 0.0)  
  
    Bases: object  
  
    cooldown_time  
  
    classmethod from_json(data: Dict[KT, VT])  
  
    head_angle_max  
  
    head_angle_min  
  
    mood  
  
    name  
  
    use_head_angle  
  
    weight  
  
class pycozmo.anim.AnimationGroup(members: Iterable[pycozmo.anim.AnimationGroupMember])  
    Bases: object  
  
    choose_member()  
        Choose member by weight.  
  
    classmethod from_json(data: Dict[KT, VT])  
  
    member_probabilities  
  
    members  
  
pycozmo.anim.load_animation_groups(resource_dir: str) → Dict[str, py-  
                                cozmo.anim.AnimationGroup]  
  
pycozmo.anim.load_cube_animation_groups(resource_dir: str) → Dict[str,  
                                List[pycozmo.anim.CubeAnimation]]  
  
pycozmo.anim.load_backpack_light_patterns(resource_dir: str) → Dict[str, py-  
                                cozmo.anim.BackpackAnimation]
```

## 10.8 pycozmo.anim\_controller

Animation controller for audio, image, and animation playback.

### Classes

---

<a href="#"><i>AnimationController</i></a> (cli)	Animation controller class.
<a href="#"><i>AnimationQueue</i></a> ()	Synchronized animation queue class.

---

```
class pycozmo.anim_controller.AnimationController(cli)  
    Bases: object  
  
    Animation controller class.  
  
    cancel_anim()  
  
    display_image(pkt: pycozmo.protocol_encoder.DisplayImage) → None
```

```

enable_animations (enabled: bool = True) → None
enable_procedural_face (enabled: bool = True) → None
play_anim_frame (audio_pkt: Optional[pycozmo.protocol_encoder.OutputAudio],    im-
                  age_pkt: Optional[pycozmo.protocol_encoder.DisplayImage],    pks: Op-
                  tional[Iterable[pycozmo.protocol_base.Packet]]) → None
play_audio (pkts: List[pycozmo.protocol_encoder.OutputAudio]) → None
start ()
stop ()
class pycozmo.anim_controller.AnimationQueue
    Bases: object
    Synchronized animation queue class.
    MAXLEN = 4500
    clear ()
    get () → Tuple[bytes, bytes, Tuple[Any]]
    is_empty ()
    put_anim_frame (audio_pkt: Optional[pycozmo.protocol_encoder.OutputAudio],    im-
                   age_pkt: Optional[pycozmo.protocol_encoder.DisplayImage],    pks: Op-
                   tional[Iterable[pycozmo.protocol_base.Packet]]) → None
    put_audio (pkts: List[pycozmo.protocol_encoder.OutputAudio]) → None
    put_image (pkt: pycozmo.protocol_encoder.DisplayImage) → None

```

## 10.9 pycozmo.anim\_encoder

Reading and writing of Cozmo animations in FlatBuffers (.bin) and JSON format.

Cozmo animations are stored in files/cozmo/cozmo\_resources/assets/animations inside the Cozmo mobile application.

Animation data structures are declared in FlatBuffers format in files/cozmo/cozmo\_resources/config/cozmo\_anim.fbs .

### Functions

<code>get_clip_metadata(dspeg)</code>	Retrieve clip metadata from animation FlatBuffers .bin files.
---------------------------------------	---

### Classes

<code>AnimBackpackLights(trigger_time_ms, ...)</code>	Backpack lights keyframe class.
<code>AnimBase()</code>	Animation element base class.
<code>AnimBodyMotion(trigger_time_ms, duration_ms, ...)</code>	Body motion keyframe class.
<code>AnimClip(name, keyframes)</code>	Animation clip class.
<code>AnimClips(clips)</code>	Animation clips class.

Continued on next page

Table 13 – continued from previous page

<code>AnimEvent(trigger_time_ms, event_id)</code>	Event keyframe class.
<code>AnimFaceAnimation(trigger_time_ms, anim_name)</code>	Face animation keyframe class.
<code>AnimHeadAngle(trigger_time_ms, duration_ms, ...)</code>	Head angle keyframe class.
<code>AnimKeyframe()</code>	Animation keyframe base class.
<code>AnimLiftHeight(trigger_time_ms, duration_ms, ...)</code>	Lift height keyframe class.
<code>AnimLight(red, green, blue, ir)</code>	Light color class.
<code>AnimProceduralFace(trigger_time_ms, angle, ...)</code>	Procedural face keyframe class.
<code>AnimRecordHeading(trigger_time_ms)</code>	Record heading keyframe class.
<code>AnimRobotAudio(trigger_time_ms, ...)</code>	Robot audio keyframe class.
<code>AnimTurnToRecordedHeading(trigger_time_ms, ...)</code>	Turn-to-recorded-heading keyframe class.
<code>ClipMetadata(fspect, index, name, ...)</code>	Animation clip metadata class.

**class** pycozmo.anim\_encoder.**AnimBase**

Bases: `abc.ABC`

Animation element base class.

**classmethod** `from_dict` (*data*)

**classmethod** `from_fb` (*buf*)

`to_dict` () → dict

`to_fb` (*builder*: `flatbuffers.builder.Builder`)

**class** pycozmo.anim\_encoder.**AnimClip** (*name*: *str*, *keyframes*: *Iterable*[`pycozmo.anim_encoder.AnimKeyframe`] = ())

Bases: `pycozmo.anim_encoder.AnimBase`

Animation clip class.

**classmethod** `from_dict` (*data*: dict) → `pycozmo.anim_encoder.AnimClip`

**classmethod** `from_fb` (*fbclip*: `pycozmo.CozmoAnim.AnimClip.AnimClip`) → `pycozmo.anim_encoder.AnimClip`

`to_dict` () → dict

`to_fb` (*builder*: `flatbuffers.builder.Builder`)

**class** pycozmo.anim\_encoder.**AnimClips** (*clips*: *Iterable*[`pycozmo.anim_encoder.AnimClip`] = ())

Bases: `pycozmo.anim_encoder.AnimBase`

Animation clips class.

**classmethod** `from_dict` (*data*: dict) → `pycozmo.anim_encoder.AnimClips`

**classmethod** `from_fb` (*fbclips*: `pycozmo.CozmoAnim.AnimClips.AnimClips`) → `pycozmo.anim_encoder.AnimClips`

**classmethod** `from_fb_file` (*fspec*: *str*) → `pycozmo.anim_encoder.AnimClips`

**classmethod** `from_fb_stream` (*f*: *BinaryIO*) → `pycozmo.anim_encoder.AnimClips`

**classmethod** `from_json_file` (*fspec*: *str*) → `pycozmo.anim_encoder.AnimClips`

**classmethod** `from_json_stream` (*f*: *TextIO*) → `pycozmo.anim_encoder.AnimClips`

`to_dict` () → dict

```

    to_fb (builder: flatbuffers.builder.Builder)
    to_fb_file (fspec: str)
    to_fb_stream (f: BinaryIO)
    to_json_file (fspec: str) → None
    to_json_stream (f: TextIO) → None

class pycozmo.anim_encoder.AnimLight (red: int = 0, green: int = 0, blue: int = 0, ir: int = 0)
    Bases: object
    Light color class.

    classmethod from_dict (data) → pycozmo.anim_encoder.AnimLight
    to_dict () → list

class pycozmo.anim_encoder.AnimKeyframe
    Bases: pycozmo.anim_encoder.AnimBase, abc.ABC
    Animation keyframe base class.

    classmethod from_dict (data)
    classmethod from_fb (buf)
    to_dict () → dict
    to_fb (builder: flatbuffers.builder.Builder)

class pycozmo.anim_encoder.AnimHeadAngle (trigger_time_ms: int = 0, duration_ms: int = 0,
                                           angle_deg: int = 0, variability_deg: int = 0)
    Bases: pycozmo.anim_encoder.AnimKeyframe
    Head angle keyframe class.

    classmethod from_dict (data) → pycozmo.anim_encoder.AnimHeadAngle
    classmethod from_fb (fbkf: pycozmo.CozmoAnim.HeadAngle.HeadAngle) → py-
                                cozmo.anim_encoder.AnimHeadAngle
    to_dict () → dict
    to_fb (builder: flatbuffers.builder.Builder)

class pycozmo.anim_encoder.AnimLiftHeight (trigger_time_ms: int = 0, duration_ms: int = 0,
                                           height_mm: int = 0, variability_mm: int = 0)
    Bases: pycozmo.anim_encoder.AnimKeyframe
    Lift height keyframe class.

    classmethod from_dict (data) → pycozmo.anim_encoder.AnimLiftHeight
    classmethod from_fb (fbkf: pycozmo.CozmoAnim.LiftHeight.LiftHeight) → py-
                                cozmo.anim_encoder.AnimLiftHeight
    to_dict () → dict
    to_fb (builder: flatbuffers.builder.Builder)

class pycozmo.anim_encoder.AnimRecordHeading (trigger_time_ms: int = 0)
    Bases: pycozmo.anim_encoder.AnimKeyframe
    Record heading keyframe class.

    classmethod from_dict (data) → pycozmo.anim_encoder.AnimRecordHeading

```

```
classmethod from_fb (fbkf: pycozmo.CozmoAnim.RecordHeading.RecordHeading) → py-
    cozmo.anim_encoder.AnimRecordHeading

to_dict () → dict

to_fb (builder: flatbuffers.builder.Builder)

class pycozmo.anim_encoder.AnimTurnToRecordedHeading (trigger_time_ms: int = 0, du-
    ration_ms: int = 0, offset_deg:
    int = 0, speed_deg_per_sec: int
    = 0, accel_deg_per_sec_2: int
    = 1000, decel_deg_per_sec_2:
    int = 1000, tolerance_deg: int
    = 2, num_half_revs: int =
    0, use_shortest_dir: bool =
    False)

Bases: pycozmo.anim_encoder.AnimKeyframe
Turn-to-recorded-heading keyframe class.

classmethod from_dict (data) → pycozmo.anim_encoder.AnimTurnToRecordedHeading

classmethod from_fb (fbkf: pycozmo.CozmoAnim.TurnToRecordedHeading.TurnToRecordedHeading)
    → pycozmo.anim_encoder.AnimTurnToRecordedHeading

to_dict () → dict

to_fb (builder: flatbuffers.builder.Builder)

class pycozmo.anim_encoder.AnimBodyMotion (trigger_time_ms: int = 0, duration_ms: int =
    0, radius_mm: Union[float, str] = 'STRAIGHT',
    speed: int = 0)

Bases: pycozmo.anim_encoder.AnimKeyframe
Body motion keyframe class.

classmethod from_dict (data) → pycozmo.anim_encoder.AnimBodyMotion

classmethod from_fb (fbkf: pycozmo.CozmoAnim.BodyMotion.BodyMotion) → py-
    cozmo.anim_encoder.AnimBodyMotion

to_dict () → dict

to_fb (builder: flatbuffers.builder.Builder)

class pycozmo.anim_encoder.AnimBackpackLights (trigger_time_ms: int = 0, du-
    ration_ms: int = 0, left: py-
    cozmo.anim_encoder.AnimLight = <py-
    cozmo.anim_encoder.AnimLight object>,
    front: pycozmo.anim_encoder.AnimLight
    = <pycozmo.anim_encoder.AnimLight
    object>, middle: py-
    cozmo.anim_encoder.AnimLight = <py-
    cozmo.anim_encoder.AnimLight object>,
    back: pycozmo.anim_encoder.AnimLight
    = <pycozmo.anim_encoder.AnimLight
    object>, right: py-
    cozmo.anim_encoder.AnimLight =
    <pycozmo.anim_encoder.AnimLight
    object>)
```

Bases: `pycozmo.anim_encoder.AnimKeyframe`

Backpack lights keyframe class.

```

classmethod from_dict (data) → pycozmo.anim_encoder.AnimBackpackLights
classmethod from_fb (fbkf: pycozmo.CozmoAnim.BackpackLights.BackpackLights) → py-
    cozmo.anim_encoder.AnimBackpackLights
to_dict () → dict
to_fb (builder: flatbuffers.builder.Builder)

class pycozmo.anim_encoder.AnimFaceAnimation (trigger_time_ms: int = 0, anim_name: str
    = "")
    Bases: pycozmo.anim_encoder.AnimKeyframe
    Face animation keyframe class.
classmethod from_dict (data) → pycozmo.anim_encoder.AnimFaceAnimation
classmethod from_fb (fbkf: pycozmo.CozmoAnim.FaceAnimation.FaceAnimation) → py-
    cozmo.anim_encoder.AnimFaceAnimation
to_dict () → dict
to_fb (builder: flatbuffers.builder.Builder)

class pycozmo.anim_encoder.AnimProceduralFace (trigger_time_ms: int = 0, angle: float
    = 0.0, center_x: float = 0.0, center_y:
    float = 0.0, scale_x: float = 1.0, scale_y:
    float = 1.0, left_eye: Iterable[float] = (),
    right_eye: Iterable[float] = ())
    Bases: pycozmo.anim_encoder.AnimKeyframe
    Procedural face keyframe class.
classmethod from_dict (data) → pycozmo.anim_encoder.AnimProceduralFace
classmethod from_fb (fbkf: pycozmo.CozmoAnim.ProceduralFace.ProceduralFace) → py-
    cozmo.anim_encoder.AnimProceduralFace
to_dict () → dict
to_fb (builder: flatbuffers.builder.Builder)

class pycozmo.anim_encoder.AnimRobotAudio (trigger_time_ms: int = 0, audio_event_ids: Iter-
    able[int] = (), volume: float = 1.0, probabilities:
    Iterable[float] = (), has_alts: bool = True)
    Bases: pycozmo.anim_encoder.AnimKeyframe
    Robot audio keyframe class.
classmethod from_dict (data) → pycozmo.anim_encoder.AnimRobotAudio
classmethod from_fb (fbkf: pycozmo.CozmoAnim.RobotAudio.RobotAudio) → py-
    cozmo.anim_encoder.AnimRobotAudio
to_dict () → dict
to_fb (builder: flatbuffers.builder.Builder)

class pycozmo.anim_encoder.AnimEvent (trigger_time_ms: int = 0, event_id: str = "")
    Bases: pycozmo.anim_encoder.AnimKeyframe
    Event keyframe class.
classmethod from_dict (data) → pycozmo.anim_encoder.AnimEvent
classmethod from_fb (fbkf: pycozmo.CozmoAnim.Event.Event) → py-
    cozmo.anim_encoder.AnimEvent
to_dict () → dict

```

`to_fb` (*builder: flatbuffers.builder.Builder*)

```
class pycozmo.anim_encoder.ClipMetadata (fspec:      str,      index:      int,      name:
                                         str,      has_head_angle_track:      bool,
                                         has_lift_height_track:      bool,
                                         has_record_heading_track:      bool,
                                         has_turn_to_recorded_heading_track:
                                         bool,      has_body_motion_track:      bool,
                                         has_backpack_lights_track:      bool,
                                         has_face_animation_track:      bool,
                                         has_procedural_face_track:      bool,
                                         has_robot_audio_track:      bool, has_event_track:
                                         bool)
```

Bases: `object`

Animation clip metadata class.

```
pycozmo.anim_encoder.get_clip_metadata (dspec:      str)      →      Dict[str,      py-
                                         cozmo.anim_encoder.ClipMetadata]
```

Retrieve clip metadata from animation FlatBuffers .bin files.

## 10.10 pycozmo.audio

Cozmo audio encoding.

### References:

- [https://en.wikipedia.org/wiki/%CE%9C-law\\_algorithm](https://en.wikipedia.org/wiki/%CE%9C-law_algorithm)
- <http://dystopiancode.blogspot.com/2012/02/pcm-law-and-u-law-companding-algorithms.html>

### Functions

<code>bytes_to_cozmo(byte_string, rate_correction, ...)</code>	Convert a 744 sample, 16-bit audio frame into a U-law encoded frame.
<code>load_wav(filename)</code>	Load a WAVE file into a list of OutputAudio packets.
<code>u_law_encoding(sample)</code>	U-law encode a 16-bit PCM sample.

```
pycozmo.audio.load_wav (filename: str) → List[pycozmo.protocol_encoder.OutputAudio]
```

Load a WAVE file into a list of OutputAudio packets.

## 10.11 pycozmo.behavior

Behavior representation and reading.

### Functions

<code>get_behavior_class_from_dict(data)</code>	Choose a behavior class, based on the behaviorClass JSON attribute.
<code>load_behaviors(resource_dir, cli)</code>	

Continued on next page



Table 15 – continued from previous page

---

`load_reaction_trigger_behavior_map(resource_dir)`


---

**Classes**

<code>Behavior(cli, conf)</code>	Behavior representation class.
<code>BehaviorDriveOffCharger(cli, conf)</code>	
<code>BehaviorPlayAnim(cli, conf)</code>	Play a sequence of animation triggers.
<code>BehaviorPlayArbitraryAnim(cli, conf)</code>	Play a random animation trigger.
<code>BehaviorReactToCliff(cli, conf)</code>	ReactToCliff behavior - currently, just plays animation.
<code>ReactionTrigger(name, behavior_id, ...)</code>	Reaction trigger representation class.

---

```
class pycozmo.behavior.ReactionTrigger(name: str, behavior_id: str, should_resume_last:
                                     Optional[bool] = False)
```

Bases: `object`

Reaction trigger representation class.

**behavior\_id**

**classmethod** `from_json(data: Dict[KT, VT])`

**name**

**should\_resume\_last**

```
class pycozmo.behavior.Behavior(cli: pycozmo.client.Client, conf: Any)
```

Bases: `pycozmo.event.Dispatcher`

Behavior representation class.

**activate()** → None

**add\_child\_dispatcher(child)**

**add\_handler(event, f, one\_shot=False)**

**deactivate()** → None

**del\_all\_handlers()**

**del\_child\_dispatcher(child)**

**del\_handler(event, handler)**

**dispatch(event, \*args, \*\*kwargs)**

**get\_id()** → str

**wait\_for(evt, timeout: Optional[float] = None)** → None

```
pycozmo.behavior.load_behaviors(resource_dir: str, cli: pycozmo.client.Client) → Dict[str, py-
                                     cozmo.behavior.Behavior]
```

```
pycozmo.behavior.load_reaction_trigger_behavior_map(resource_dir: str)
→ Dict[str, py-
   cozmo.behavior.ReactionTrigger]
```

## 10.12 pycozmo.brain

Brain class - high level behavior and emotion engine.

## Classes

---

<i>Brain</i> (cli)	Cozmo robot brain class.
--------------------	--------------------------

---

```
class pycozmo.brain.Brain (cli: pycozmo.client.Client)
    Bases: object
    Cozmo robot brain class.

    activate_behavior (behavior_id: str) → None
    deactivate_behavior () → None
    heartbeat_thread_run () → None
        Heartbeat thread loop.
    on_behavior_done (cli: pycozmo.client.Client) → None
    on_camera_image (cli: pycozmo.client.Client, new_im) → None
        Process images, coming from the robot camera.
    on_cliff_detected (cli: pycozmo.client.Client, state: bool) → None
    on_robot_falling_change (cli: pycozmo.client.Client, state: bool)
    on_robot_on_charger_change (cli: pycozmo.client.Client, state: bool) → None
    on_robot_orientation_change (cli: pycozmo.client.Client, orientation: py-
        cozmo.robot.RobotOrientation) → None
    on_robot_picked_up_change (cli: pycozmo.client.Client, state: bool) → None
    post_reaction (reaction_trigger: str) → None
        Post a reaction trigger to the reaction trigger queue.
    process_emotion_event (reaction_trigger: str) → None
        Process emotion events by applying affectors to emotion types.
    process_reaction (reaction_trigger: str) → None
    reaction_thread_run () → None
        Reaction thread loop. Reaction trigger queue processing.
    start ()
    stop ()
    update_emotion_types () → None
        Update emotion types from their decay functions.
```

## 10.13 pycozmo.camera

Camera image decoding.

### Functions

---

<i>mini_to_jpeg_helper</i> (mini, width, height, header)	Low-level mini*ToJpeg format to normal JPEG format conversion.
--	--

---

Continued on next page

Table 18 – continued from previous page

<code>minicolor_to_jpeg(minicolor, width, height)</code>	Converts miniColorToJpeg format to normal JPEG format.
<code>minigray_to_jpeg(minigray, width, height)</code>	Converts miniGrayToJpeg format to normal JPEG format.

## Classes

<code>CameraConfig(focal_length_x, focal_length_y, ...)</code>	Robot camera fixed property representation.
--	---

`pycozmo.camera.RESOLUTIONS = {<ImageResolution.VerificationSnapshot: 0>: (16, 16), <ImageResolution.VerificationSnapshot: 1>: (32, 32), <ImageResolution.VerificationSnapshot: 2>: (64, 64), <ImageResolution.VerificationSnapshot: 3>: (128, 128), <ImageResolution.VerificationSnapshot: 4>: (256, 256), <ImageResolution.VerificationSnapshot: 5>: (512, 512), <ImageResolution.VerificationSnapshot: 6>: (1024, 1024), <ImageResolution.VerificationSnapshot: 7>: (2048, 2048), <ImageResolution.VerificationSnapshot: 8>: (4096, 4096), <ImageResolution.VerificationSnapshot: 9>: (8192, 8192), <ImageResolution.VerificationSnapshot: 10>: (16384, 16384), <ImageResolution.VerificationSnapshot: 11>: (32768, 32768), <ImageResolution.VerificationSnapshot: 12>: (65536, 65536), <ImageResolution.VerificationSnapshot: 13>: (131072, 131072), <ImageResolution.VerificationSnapshot: 14>: (262144, 262144), <ImageResolution.VerificationSnapshot: 15>: (524288, 524288), <ImageResolution.VerificationSnapshot: 16>: (1048576, 1048576), <ImageResolution.VerificationSnapshot: 17>: (2097152, 2097152), <ImageResolution.VerificationSnapshot: 18>: (4194304, 4194304), <ImageResolution.VerificationSnapshot: 19>: (8388608, 8388608), <ImageResolution.VerificationSnapshot: 20>: (16777216, 16777216), <ImageResolution.VerificationSnapshot: 21>: (33554432, 33554432), <ImageResolution.VerificationSnapshot: 22>: (67108864, 67108864), <ImageResolution.VerificationSnapshot: 23>: (134217728, 134217728), <ImageResolution.VerificationSnapshot: 24>: (268435456, 268435456), <ImageResolution.VerificationSnapshot: 25>: (536870912, 536870912), <ImageResolution.VerificationSnapshot: 26>: (1073741824, 1073741824), <ImageResolution.VerificationSnapshot: 27>: (2147483648, 2147483648), <ImageResolution.VerificationSnapshot: 28>: (4294967296, 4294967296), <ImageResolution.VerificationSnapshot: 29>: (8589934592, 8589934592), <ImageResolution.VerificationSnapshot: 30>: (17179869184, 17179869184), <ImageResolution.VerificationSnapshot: 31>: (34359738368, 34359738368), <ImageResolution.VerificationSnapshot: 32>: (68719476736, 68719476736), <ImageResolution.VerificationSnapshot: 33>: (137438953472, 137438953472), <ImageResolution.VerificationSnapshot: 34>: (274877906944, 274877906944), <ImageResolution.VerificationSnapshot: 35>: (549755813888, 549755813888), <ImageResolution.VerificationSnapshot: 36>: (1099511627776, 1099511627776), <ImageResolution.VerificationSnapshot: 37>: (2199023255552, 2199023255552), <ImageResolution.VerificationSnapshot: 38>: (4398046511104, 4398046511104), <ImageResolution.VerificationSnapshot: 39>: (8796093022208, 8796093022208), <ImageResolution.VerificationSnapshot: 40>: (17592186044416, 17592186044416), <ImageResolution.VerificationSnapshot: 41>: (35184372088832, 35184372088832), <ImageResolution.VerificationSnapshot: 42>: (70368744177664, 70368744177664), <ImageResolution.VerificationSnapshot: 43>: (140737488355328, 140737488355328), <ImageResolution.VerificationSnapshot: 44>: (281474976710656, 281474976710656), <ImageResolution.VerificationSnapshot: 45>: (562949953421312, 562949953421312), <ImageResolution.VerificationSnapshot: 46>: (1125899906842624, 1125899906842624), <ImageResolution.VerificationSnapshot: 47>: (2251799813685248, 2251799813685248), <ImageResolution.VerificationSnapshot: 48>: (4503599627370496, 4503599627370496), <ImageResolution.VerificationSnapshot: 49>: (9007199254740992, 9007199254740992), <ImageResolution.VerificationSnapshot: 50>: (18014398509481984, 18014398509481984), <ImageResolution.VerificationSnapshot: 51>: (36028797018963968, 36028797018963968), <ImageResolution.VerificationSnapshot: 52>: (72057594037927936, 72057594037927936), <ImageResolution.VerificationSnapshot: 53>: (144115188075855872, 144115188075855872), <ImageResolution.VerificationSnapshot: 54>: (288230376151711744, 288230376151711744), <ImageResolution.VerificationSnapshot: 55>: (576460752303423488, 576460752303423488), <ImageResolution.VerificationSnapshot: 56>: (1152921504606846976, 1152921504606846976), <ImageResolution.VerificationSnapshot: 57>: (2305843009213693952, 2305843009213693952), <ImageResolution.VerificationSnapshot: 58>: (4611686018427387904, 4611686018427387904), <ImageResolution.VerificationSnapshot: 59>: (9223372036854775808, 9223372036854775808), <ImageResolution.VerificationSnapshot: 60>: (18446744073709551616, 18446744073709551616), <ImageResolution.VerificationSnapshot: 61>: (36893488147419103232, 36893488147419103232), <ImageResolution.VerificationSnapshot: 62>: (73786976294838206464, 73786976294838206464), <ImageResolution.VerificationSnapshot: 63>: (147573952589676412928, 147573952589676412928), <ImageResolution.VerificationSnapshot: 64>: (295147905179352825856, 295147905179352825856), <ImageResolution.VerificationSnapshot: 65>: (590295810358705651712, 590295810358705651712), <ImageResolution.VerificationSnapshot: 66>: (1180591620717411303424, 1180591620717411303424), <ImageResolution.VerificationSnapshot: 67>: (2361183241434822606848, 2361183241434822606848), <ImageResolution.VerificationSnapshot: 68>: (4722366482869645213696, 4722366482869645213696), <ImageResolution.VerificationSnapshot: 69>: (9444732965739290427392, 9444732965739290427392), <ImageResolution.VerificationSnapshot: 70>: (18889465931478580854784, 18889465931478580854784), <ImageResolution.VerificationSnapshot: 71>: (37778931862957161709568, 37778931862957161709568), <ImageResolution.VerificationSnapshot: 72>: (75557863725914323419136, 75557863725914323419136), <ImageResolution.VerificationSnapshot: 73>: (151115727451828646838272, 151115727451828646838272), <ImageResolution.VerificationSnapshot: 74>: (302231454903657293676544, 302231454903657293676544), <ImageResolution.VerificationSnapshot: 75>: (604462909807314587353088, 604462909807314587353088), <ImageResolution.VerificationSnapshot: 76>: (1208925819614629174706176, 1208925819614629174706176), <ImageResolution.VerificationSnapshot: 77>: (2417851639229258349412352, 2417851639229258349412352), <ImageResolution.VerificationSnapshot: 78>: (4835703278458516698824704, 4835703278458516698824704), <ImageResolution.VerificationSnapshot: 79>: (9671406556917033397649408, 9671406556917033397649408), <ImageResolution.VerificationSnapshot: 80>: (19342813113834066795298816, 19342813113834066795298816), <ImageResolution.VerificationSnapshot: 81>: (38685626227668133590597632, 38685626227668133590597632), <ImageResolution.VerificationSnapshot: 82>: (77371252455336267181195264, 77371252455336267181195264), <ImageResolution.VerificationSnapshot: 83>: (154742504910672534362390528, 154742504910672534362390528), <ImageResolution.VerificationSnapshot: 84>: (309485009821345068724781056, 309485009821345068724781056), <ImageResolution.VerificationSnapshot: 85>: (618970019642690137449562112, 618970019642690137449562112), <ImageResolution.VerificationSnapshot: 86>: (1237940039285380274899124224, 1237940039285380274899124224), <ImageResolution.VerificationSnapshot: 87>: (2475880078570760549798248448, 2475880078570760549798248448), <ImageResolution.VerificationSnapshot: 88>: (4951760157141521099596496896, 4951760157141521099596496896), <ImageResolution.VerificationSnapshot: 89>: (9903520314283042199192993792, 9903520314283042199192993792), <ImageResolution.VerificationSnapshot: 90>: (19807040628566084398385987584, 19807040628566084398385987584), <ImageResolution.VerificationSnapshot: 91>: (39614081257132168796771975168, 39614081257132168796771975168), <ImageResolution.VerificationSnapshot: 92>: (79228162514264337593543950336, 79228162514264337593543950336), <ImageResolution.VerificationSnapshot: 93>: (158456325028528675187087900672, 158456325028528675187087900672), <ImageResolution.VerificationSnapshot: 94>: (316912650057057350374175801344, 316912650057057350374175801344), <ImageResolution.VerificationSnapshot: 95>: (633825300114114700748351602688, 633825300114114700748351602688), <ImageResolution.VerificationSnapshot: 96>: (1267650600228229401496703205376, 1267650600228229401496703205376), <ImageResolution.VerificationSnapshot: 97>: (2535301200456458802993406410752, 2535301200456458802993406410752), <ImageResolution.VerificationSnapshot: 98>: (5070602400912917605986812821504, 5070602400912917605986812821504), <ImageResolution.VerificationSnapshot: 99>: (10141204801825835211973625643008, 10141204801825835211973625643008), <ImageResolution.VerificationSnapshot: 100>: (20282409603651670423947251286016, 20282409603651670423947251286016), <ImageResolution.VerificationSnapshot: 101>: (40564819207303340847894502572032, 40564819207303340847894502572032), <ImageResolution.VerificationSnapshot: 102>: (81129638414606681695789005144064, 81129638414606681695789005144064), <ImageResolution.VerificationSnapshot: 103>: (162259276829213363391578010288128, 162259276829213363391578010288128), <ImageResolution.VerificationSnapshot: 104>: (324518553658426726783156020576256, 324518553658426726783156020576256), <ImageResolution.VerificationSnapshot: 105>: (649037107316853453566312041152512, 649037107316853453566312041152512), <ImageResolution.VerificationSnapshot: 106>: (1298074214633706907132624082305024, 1298074214633706907132624082305024), <ImageResolution.VerificationSnapshot: 107>: (2596148429267413814265248164610048, 2596148429267413814265248164610048), <ImageResolution.VerificationSnapshot: 108>: (5192296858534827628530496329220096, 5192296858534827628530496329220096), <ImageResolution.VerificationSnapshot: 109>: (10384593717069655257060992658440192, 10384593717069655257060992658440192), <ImageResolution.VerificationSnapshot: 110>: (20769187434139310514121985316880384, 20769187434139310514121985316880384), <ImageResolution.VerificationSnapshot: 111>: (41538374868278621028243970633760768, 41538374868278621028243970633760768), <ImageResolution.VerificationSnapshot: 112>: (83076749736557242056487941267521536, 83076749736557242056487941267521536), <ImageResolution.VerificationSnapshot: 113>: (166153499473114484112975882535043072, 166153499473114484112975882535043072), <ImageResolution.VerificationSnapshot: 114>: (332306998946228968225951765070086144, 332306998946228968225951765070086144), <ImageResolution.VerificationSnapshot: 115>: (664613997892457936451903530140172288, 664613997892457936451903530140172288), <ImageResolution.VerificationSnapshot: 116>: (1329227995784915872903807060280344576, 1329227995784915872903807060280344576), <ImageResolution.VerificationSnapshot: 117>: (2658455991569831745807614120560689152, 2658455991569831745807614120560689152), <ImageResolution.VerificationSnapshot: 118>: (5316911983139663491615228241121378304, 5316911983139663491615228241121378304), <ImageResolution.VerificationSnapshot: 119>: (10633823966279326983230456482242756608, 10633823966279326983230456482242756608), <ImageResolution.VerificationSnapshot: 120>: (21267647932558653966460912964485513216, 21267647932558653966460912964485513216), <ImageResolution.VerificationSnapshot: 121>: (42535295865117307932921825928971026432, 42535295865117307932921825928971026432), <ImageResolution.VerificationSnapshot: 122>: (85070591730234615865843651857942052864, 85070591730234615865843651857942052864), <ImageResolution.VerificationSnapshot: 123>: (170141183460469231731687303715884105728, 170141183460469231731687303715884105728), <ImageResolution.VerificationSnapshot: 124>: (340282366920938463463374607431768211456, 340282366920938463463374607431768211456), <ImageResolution.VerificationSnapshot: 125>: (680564733841876926926749214863536422912, 680564733841876926926749214863536422912), <ImageResolution.VerificationSnapshot: 126>: (1361129467683753853853498429727072845824, 1361129467683753853853498429727072845824), <ImageResolution.VerificationSnapshot: 127>: (2722258935367507707706996859454145691648, 2722258935367507707706996859454145691648), <ImageResolution.VerificationSnapshot: 128>: (5444517870735015415413993718908291383296, 5444517870735015415413993718908291383296), <ImageResolution.VerificationSnapshot: 129>: (10889035741470030830827987437816582766592, 10889035741470030830827987437816582766592), <ImageResolution.VerificationSnapshot: 130>: (21778071482940061661655974875633165533184, 21778071482940061661655974875633165533184), <ImageResolution.VerificationSnapshot: 131>: (43556142965880123323311949751266331066368, 43556142965880123323311949751266331066368), <ImageResolution.VerificationSnapshot: 132>: (87112285931760246646623899502532662132736, 87112285931760246646623899502532662132736), <ImageResolution.VerificationSnapshot: 133>: (174224571863520493293247799005065324265472, 174224571863520493293247799005065324265472), <ImageResolution.VerificationSnapshot: 134>: (348449143727040986586495598010130648530944, 348449143727040986586495598010130648530944), <ImageResolution.VerificationSnapshot: 135>: (696898287454081973172991196020261297061888, 696898287454081973172991196020261297061888), <ImageResolution.VerificationSnapshot: 136>: (1393796574908163946345982392040522594123776, 1393796574908163946345982392040522594123776), <ImageResolution.VerificationSnapshot: 137>: (2787593149816327892691964784081045188247552, 2787593149816327892691964784081045188247552), <ImageResolution.VerificationSnapshot: 138>: (5575186299632655785383929568162090376495104, 5575186299632655785383929568162090376495104), <ImageResolution.VerificationSnapshot: 139>: (11150372599265311570767859136324180752990208, 11150372599265311570767859136324180752990208), <ImageResolution.VerificationSnapshot: 140>: (22300745198530623141535718272648361505980416, 22300745198530623141535718272648361505980416), <ImageResolution.VerificationSnapshot: 141>: (44601490397061246283071436545296723011960832, 44601490397061246283071436545296723011960832), <ImageResolution.VerificationSnapshot: 142>: (89202980794122492566142873090593446023921664, 89202980794122492566142873090593446023921664), <ImageResolution.VerificationSnapshot: 143>: (178405961588244985132285746181186892047843328, 178405961588244985132285746181186892047843328), <ImageResolution.VerificationSnapshot: 144>: (356811923176489970264571492362373784095686656, 356811923176489970264571492362373784095686656), <ImageResolution.VerificationSnapshot: 145>: (713623846352979940529142984724747568191373312, 713623846352979940529142984724747568191373312), <ImageResolution.VerificationSnapshot: 146>: (1427247692705959881058285969449495136382746624, 1427247692705959881058285969449495136382746624), <ImageResolution.VerificationSnapshot: 147>: (2854495385411919762116571938898990272765493248, 2854495385411919762116571938898990272765493248), <ImageResolution.VerificationSnapshot: 148>: (5708990770823839524233143877797980545530986496, 5708990770823839524233143877797980545530986496), <ImageResolution.VerificationSnapshot: 149>: (11417981541647679048466287755595961091061972992, 11417981541647679048466287755595961091061972992), <ImageResolution.VerificationSnapshot: 150>: (22835963083295358096932575511191922182123945984, 22835963083295358096932575511191922182123945984), <ImageResolution.VerificationSnapshot: 151>: (45671926166590716193865151022383844364247891968, 45671926166590716193865151022383844364247891968), <ImageResolution.VerificationSnapshot: 152>: (91343852333181432387730302044767688728495783936, 91343852333181432387730302044767688728495783936), <ImageResolution.VerificationSnapshot: 153>: (182687704666362864775460604089535377456991567872, 182687704666362864775460604089535377456991567872), <ImageResolution.VerificationSnapshot: 154>: (365375409332725729550921208179070754913983135744, 365375409332725729550921208179070754913983135744), <ImageResolution.VerificationSnapshot: 155>: (730750818665451459101842416358141509827966271488, 730750818665451459101842416358141509827966271488), <ImageResolution.VerificationSnapshot: 156>: (1461501637330902918203684832716283019655932542976, 1461501637330902918203684832716283019655932542976), <ImageResolution.VerificationSnapshot: 157>: (2923003274661805836407369665432566039311865085952, 2923003274661805836407369665432566039311865085952), <ImageResolution.VerificationSnapshot: 158>: (5846006549323611672814739330865132078623730171904, 5846006549323611672814739330865132078623730171904), <ImageResolution.VerificationSnapshot: 159>: (11692013098647223345629478661730264157247460343808, 11692013098647223345629478661730264157247460343808), <ImageResolution.VerificationSnapshot: 160>: (23384026197294446691258957323460528314494920687616, 23384026197294446691258957323460528314494920687616), <ImageResolution.VerificationSnapshot: 161>: (46768052394588893382517914646921056628989841375232, 46768052394588893382517914646921056628989841375232), <ImageResolution.VerificationSnapshot: 162>: (93536104789177786765035829293842113257979682750464, 93536104789177786765035829293842113257979682750464), <ImageResolution.VerificationSnapshot: 163>: (187072209578355573530071658587684226515959365500928, 187072209578355573530071658587684226515959365500928), <ImageResolution.VerificationSnapshot: 164>: (374144419156711147060143317175368453031918731001856, 374144419156711147060143317175368453031918731001856), <ImageResolution.VerificationSnapshot: 165>: (748288838313422294120286634350736906063837462003712, 7482888383134222941202866343507369060638374620037`

**clear\_screen** () → None

**connect** () → None

**deactivate\_behavior** (*behavior*)

**del\_all\_handlers** ()

**del\_child\_dispatcher** (*child*)

**del\_handler** (*event*, *handler*)

**disconnect** () → None

**dispatch** (*event*, *\*args*, *\*\*kwargs*)

**display\_image** (*im*: <module 'PIL.Image' from '/home/docs/checkouts/readthedocs.org/user\_builds/pycozmo/envs/latest/lib/packages/PIL/Image.py'>, *duration*: *Optional*[float] = None) → None

**drive\_wheels** (*lwheel\_speed*: float, *rwheel\_speed*: float, *lwheel\_acc*: *Optional*[float] = 0.0, *rwheel\_acc*: *Optional*[float] = 0.0, *duration*: *Optional*[float] = None) → None

**enable\_animations** (*enabled*: bool = True) → None

**enable\_camera** (*enable*: bool = True, *color*: bool = False) → None  
Enable or disable camera image streaming in color or grayscale.

**enable\_procedural\_face** (*enabled*: bool = True) → None

**get\_anim\_names** () → set

**go\_to\_pose** (*pose*: pycozmo.util.Pose, *relative\_to\_robot*: bool = False) → None  
Move to a specific pose (position and orientation).

**load\_anims** () → None

**move\_head** (*speed*: float) → None

**move\_lift** (*speed*: float) → None

**play\_anim** (*name*: str) → None

**play\_anim\_group** (*anim\_group\_name*: str) → None

**play\_anim\_ppclip** (*ppclip*: pycozmo.anim.PreprocessedClip) → None

**play\_audio** (*fspec*: str) → None

**set\_all\_backpack\_lights** (*light*) → None

**set\_backpack\_lights** (*left\_light*, *front\_light*, *center\_light*, *rear\_light*, *right\_light*) → None

**set\_backpack\_lights\_off** () → None

**set\_center\_backpack\_lights** (*light*) → None

**set\_head\_angle** (*angle*: float, *accel*: float = 10.0, *max\_speed*: float = 10.0, *duration*: float = 0.0)

**set\_head\_light** (*enable*: bool) → None

**set\_lift\_height** (*height*: float, *accel*: float = 10.0, *max\_speed*: float = 10.0, *duration*: float = 0.0)

**set\_volume** (*level*: int) → None  
Set audio output volume to a level in the range 0-65535.

**start** () → None

**stop** () → None

**stop\_all\_motors** () → None

`wait_for (evt, timeout: Optional[float] = None) → None`

`wait_for_robot (timeout: float = 5.0) → None`

## 10.15 pycozmo.conn

Cozmo protocol low-level client and server connection.

### Classes

<code>Connection(robot_addr, int]] = None, ...)</code>	Cozmo protocol low-level connection implementing bot client and server sides.
<code>ReceiveThread(sock, send_thread, ...)</code>	Cozmo protocol connection receive thread.
<code>SendThread(sock, receiver_address, int]])</code>	Cozmo protocol connection send thread.

`pycozmo.conn.ROBOT_ADDR = ('172.31.1.1', 5551)`

Default robot address (IP, port).

**class** `pycozmo.conn.ReceiveThread` (*sock: socket.socket, send\_thread: pycozmo.conn.SendThread, sender\_address: Optional[Tuple[str, int]], delivery\_handler, buffer\_size: int = 2048*)

Bases: `threading.Thread`

Cozmo protocol connection receive thread.

#### **daemon**

A boolean value indicating whether this thread is a daemon thread.

This must be set before `start()` is called, otherwise `RuntimeError` is raised. Its initial value is inherited from the creating thread; the main thread is not a daemon thread and therefore all threads created in the main thread default to `daemon = False`.

The entire Python program exits when only daemon threads are left.

**deliver** (*pkt: pycozmo.protocol\_base.Packet*)

**deliver\_sequence** () → None

**disconnect** ()

**getName** ()

**handle\_fin** ()

**handle\_frame** (*frame: pycozmo.frame.Frame*) → None

**handle\_pkt** (*pkt: pycozmo.protocol\_base.Packet*) → None

**handle\_reset** (*address*)

#### **ident**

Thread identifier of this thread or None if it has not been started.

This is a nonzero integer. See the `get_ident()` function. Thread identifiers may be recycled when a thread exits and another thread is created. The identifier is available even after the thread has exited.

**isAlive** ()

Return whether the thread is alive.

This method is deprecated, use `is_alive()` instead.

**isDaemon()**

**is\_alive()**

Return whether the thread is alive.

This method returns True just before the run() method starts until just after the run() method terminates. The module function enumerate() returns a list of all alive threads.

**join(timeout=None)**

Wait until the thread terminates.

This blocks the calling thread until the thread whose join() method is called terminates – either normally or through an unhandled exception or until the optional timeout occurs.

When the timeout argument is present and not None, it should be a floating point number specifying a timeout for the operation in seconds (or fractions thereof). As join() always returns None, you must call is\_alive() after join() to decide whether a timeout happened – if the thread is still alive, the join() call timed out.

When the timeout argument is not present or None, the operation will block until the thread terminates.

A thread can be join()ed many times.

join() raises a RuntimeError if an attempt is made to join the current thread as that would cause a deadlock. It is also an error to join() a thread before it has been started and attempts to do so raises the same exception.

**name**

A string used for identification purposes only.

It has no semantics. Multiple threads may be given the same name. The initial name is set by the constructor.

**reset()**

**run()** → None

Method representing the thread's activity.

You may override this method in a subclass. The standard run() method invokes the callable object passed to the object's constructor as the target argument, if any, with sequential and keyword arguments taken from the args and kwargs arguments, respectively.

**setDaemon(daemonic)**

**setName(name)**

**start()**

Start the thread's activity.

It must be called at most once per thread object. It arranges for the object's run() method to be invoked in a separate thread of control.

This method will raise a RuntimeError if called more than once on the same thread object.

**stop()** → None

**class** pycozmo.conn.**SendThread**(sock: socket.socket, receiver\_address: Optional[Tuple[str, int]])

Bases: threading.Thread

Cozmo protocol connection send thread.

**ACK\_TIMEOUT** = 0.1

**COLLECT\_INTERVAL** = 0.011111111111111112

**ack**(seq: int, last\_ack: int) → None

**daemon**

A boolean value indicating whether this thread is a daemon thread.

This must be set before `start()` is called, otherwise `RuntimeError` is raised. Its initial value is inherited from the creating thread; the main thread is not a daemon thread and therefore all threads created in the main thread default to `daemon = False`.

The entire Python program exits when only daemon threads are left.

**getName()****ident**

Thread identifier of this thread or `None` if it has not been started.

This is a nonzero integer. See the `get_ident()` function. Thread identifiers may be recycled when a thread exits and another thread is created. The identifier is available even after the thread has exited.

**isAlive()**

Return whether the thread is alive.

This method is deprecated, use `is_alive()` instead.

**isDaemon()****is\_alive()**

Return whether the thread is alive.

This method returns `True` just before the `run()` method starts until just after the `run()` method terminates. The module function `enumerate()` returns a list of all alive threads.

**join(timeout=None)**

Wait until the thread terminates.

This blocks the calling thread until the thread whose `join()` method is called terminates – either normally or through an unhandled exception or until the optional timeout occurs.

When the timeout argument is present and not `None`, it should be a floating point number specifying a timeout for the operation in seconds (or fractions thereof). As `join()` always returns `None`, you must call `is_alive()` after `join()` to decide whether a timeout happened – if the thread is still alive, the `join()` call timed out.

When the timeout argument is not present or `None`, the operation will block until the thread terminates.

A thread can be `join()`ed many times.

`join()` raises a `RuntimeError` if an attempt is made to join the current thread as that would cause a deadlock. It is also an error to `join()` a thread before it has been started and attempts to do so raises the same exception.

**name**

A string used for identification purposes only.

It has no semantics. Multiple threads may be given the same name. The initial name is set by the constructor.

**reset()** → `None`**run()** → `None`

Method representing the thread's activity.

You may override this method in a subclass. The standard `run()` method invokes the callable object passed to the object's constructor as the target argument, if any, with sequential and keyword arguments taken from the `args` and `kwargs` arguments, respectively.

**send(data: Any)** → `None`

**setDaemon** (*daemonic*)

**setName** (*name*)

**start** ()

Start the thread's activity.

It must be called at most once per thread object. It arranges for the object's `run()` method to be invoked in a separate thread of control.

This method will raise a `RuntimeError` if called more than once on the same thread object.

**stop** () → None

```
class pycozmo.conn.Connection (robot_addr: Optional[Tuple[str, int]] = None, proto-  
col_log_messages: Optional[list] = None, server: bool =  
False)
```

Bases: `threading.Thread`, `pycozmo.event.Dispatcher`

Cozmo protocol low-level connection implementing bot client and server sides.

**CONNECTED** = 3

**CONNECTING** = 2

**IDLE** = 1

**PING\_INTERVAL** = 0.5

**RUN\_INTERVAL** = 0.01

**STATS\_INTERVAL** = 60.0

**add\_child\_dispatcher** (*child*)

**add\_handler** (*event, f, one\_shot=False*)

**connect** () → None

**daemon**

A boolean value indicating whether this thread is a daemon thread.

This must be set before `start()` is called, otherwise `RuntimeError` is raised. Its initial value is inherited from the creating thread; the main thread is not a daemon thread and therefore all threads created in the main thread default to `daemon = False`.

The entire Python program exits when only daemon threads are left.

**del\_all\_handlers** ()

**del\_child\_dispatcher** (*child*)

**del\_handler** (*event, handler*)

**disconnect** () → None

**dispatch** (*event, \*args, \*\*kwargs*)

**getName** ()

**ident**

Thread identifier of this thread or None if it has not been started.

This is a nonzero integer. See the `get_ident()` function. Thread identifiers may be recycled when a thread exits and another thread is created. The identifier is available even after the thread has exited.



**isAlive()**

Return whether the thread is alive.

This method is deprecated, use `is_alive()` instead.

**isDaemon()****is\_alive()**

Return whether the thread is alive.

This method returns `True` just before the `run()` method starts until just after the `run()` method terminates. The module function `enumerate()` returns a list of all alive threads.

**join(timeout=None)**

Wait until the thread terminates.

This blocks the calling thread until the thread whose `join()` method is called terminates – either normally or through an unhandled exception or until the optional timeout occurs.

When the timeout argument is present and not `None`, it should be a floating point number specifying a timeout for the operation in seconds (or fractions thereof). As `join()` always returns `None`, you must call `is_alive()` after `join()` to decide whether a timeout happened – if the thread is still alive, the `join()` call timed out.

When the timeout argument is not present or `None`, the operation will block until the thread terminates.

A thread can be `join()`ed many times.

`join()` raises a `RuntimeError` if an attempt is made to join the current thread as that would cause a deadlock. It is also an error to `join()` a thread before it has been started and attempts to do so raises the same exception.

**log\_stats()****name**

A string used for identification purposes only.

It has no semantics. Multiple threads may be given the same name. The initial name is set by the constructor.

**post\_event(evt, \*args, \*\*kwargs) → None****run() → None**

Method representing the thread's activity.

You may override this method in a subclass. The standard `run()` method invokes the callable object passed to the object's constructor as the target argument, if any, with sequential and keyword arguments taken from the `args` and `kwargs` arguments, respectively.

**send(pkt: pycozmo.protocol\_base.Packet) → None****setDaemon(daemonic)****setName(name)****start() → None**

Start the thread's activity.

It must be called at most once per thread object. It arranges for the object's `run()` method to be invoked in a separate thread of control.

This method will raise a `RuntimeError` if called more than once on the same thread object.

**stop() → None****wait\_for(evt, timeout: Optional[float] = None) → None**

## 10.16 pycozmo.emotions

Emotion representation and reading.

### Functions

---

<code>load_emotion_events(resource_dir)</code>
<code>load_emotion_types(resource_dir)</code>

---

### Classes

---

<code>DecayGraph(nodes)</code>	
<code>EmotionEvent(name, affectors, float)</code>	EmotionEvent representation class.
<code>EmotionType(name, decay_graph, ...)</code>	Emotion type class.
<code>Node(x, y)</code>	

---

```
class pycozmo.emotions.EmotionType (name: str, decay_graph: pycozmo.emotions.DecayGraph,
                                     repetition_penalty: pycozmo.emotions.DecayGraph, de-
                                     fault_value: float = 0.0, min_value: float = -1.0,
                                     max_value: float = 1.0)
```

Bases: `object`

Emotion type class.

**add** (delta: float) → None

**decay\_graph**

**last\_change**

**last\_value**

**max**

**min**

**name**

**repetition\_penalty**

**set** (value: float) → None

**update** () → None

Update from decay function.

**value**

```
class pycozmo.emotions.EmotionEvent (name: str, affectors: Dict[str, float])
```

Bases: `object`

EmotionEvent representation class.

**affectors**

**classmethod from\_json** (data: Dict[KT, VT])

**name**

```
pycozmo.emotions.load_emotion_types(resource_dir: str) → Dict[str, py-
                                cozmo.emotions.EmotionType]
pycozmo.emotions.load_emotion_events(resource_dir: str) → Dict[str, py-
                                cozmo.emotions.EmotionEvent]
```

## 10.17 pycozmo.event

Event declaration and dispatching.

### Classes

<i>Dispatcher()</i>	Event dispatcher class.
<i>Event</i>	Base class for events.
<i>EvtAnimationCompleted</i>	
<i>EvtAudioCompleted</i>	
<i>EvtBehaviorDone</i>	
<i>EvtChargerOOSChange</i>	
<i>EvtCliffDetectedChange</i>	
<i>EvtNewRawCameraImage</i>	Triggered when a new raw image is received from the robot's camera.
<i>EvtPacketReceived</i>	Triggered when a new packet has been received from the robot.
<i>EvtReactionTrigger</i>	
<i>EvtRobotAnimBufferFullChange</i>	
<i>EvtRobotAnimatingChange</i>	
<i>EvtRobotAnimatingIdleChange</i>	
<i>EvtRobotBodyAccModeChange</i>	
<i>EvtRobotCarryingBlockChange</i>	
<i>EvtRobotChargingChange</i>	
<i>EvtRobotFallingChange</i>	
<i>EvtRobotFound</i>	Triggered when the robot has been first connected.
<i>EvtRobotHeadInPositionChange</i>	
<i>EvtRobotLiftInPositionChange</i>	
<i>EvtRobotMovingChange</i>	
<i>EvtRobotOnChargerChange</i>	
<i>EvtRobotOrientationChange</i>	Triggered when the robot orientation changes.
<i>EvtRobotPathingChange</i>	
<i>EvtRobotPickedUpChange</i>	
<i>EvtRobotPickingOrPlacingChange</i>	
<i>EvtRobotReady</i>	Triggered when the robot has been initialized and is ready for commands.
<i>EvtRobotStateUpdated</i>	Triggered when a new robot state is received.
<i>EvtRobotWheelsMovingChange</i>	
<i>Handler(f, one_shot)</i>	Event handler class.

```
class pycozmo.event.Event
```

Bases: `object`

Base class for events.

```
class pycozmo.event.EvtRobotFound
    Bases: pycozmo.event.Event

    Triggered when the robot has been first connected.

class pycozmo.event.EvtRobotReady
    Bases: pycozmo.event.Event

    Triggered when the robot has been initialized and is ready for commands.

class pycozmo.event.EvtPacketReceived
    Bases: pycozmo.event.Event

    Triggered when a new packet has been received from the robot.

class pycozmo.event.EvtNewRawCameraImage
    Bases: pycozmo.event.Event

    Triggered when a new raw image is received from the robot's camera.

class pycozmo.event.EvtRobotMovingChange
    Bases: pycozmo.event.Event

class pycozmo.event.EvtRobotCarryingBlockChange
    Bases: pycozmo.event.Event

class pycozmo.event.EvtRobotPickingOrPlacingChange
    Bases: pycozmo.event.Event

class pycozmo.event.EvtRobotPickedUpChange
    Bases: pycozmo.event.Event

class pycozmo.event.EvtRobotBodyAccModeChange
    Bases: pycozmo.event.Event

class pycozmo.event.EvtRobotFallingChange
    Bases: pycozmo.event.Event

class pycozmo.event.EvtRobotAnimatingChange
    Bases: pycozmo.event.Event

class pycozmo.event.EvtRobotPathingChange
    Bases: pycozmo.event.Event

class pycozmo.event.EvtRobotLiftInPositionChange
    Bases: pycozmo.event.Event

class pycozmo.event.EvtRobotHeadInPositionChange
    Bases: pycozmo.event.Event

class pycozmo.event.EvtRobotAnimBufferFullChange
    Bases: pycozmo.event.Event

class pycozmo.event.EvtRobotAnimatingIdleChange
    Bases: pycozmo.event.Event

class pycozmo.event.EvtRobotOnChargerChange
    Bases: pycozmo.event.Event

class pycozmo.event.EvtRobotChargingChange
    Bases: pycozmo.event.Event

class pycozmo.event.EvtCliffDetectedChange
    Bases: pycozmo.event.Event
```

```
class pycozmo.event.EvtRobotWheelsMovingChange
    Bases: pycozmo.event.Event

class pycozmo.event.EvtChargerOOSChange
    Bases: pycozmo.event.Event

class pycozmo.event.EvtRobotStateUpdated
    Bases: pycozmo.event.Event
    Triggered when a new robot state is received.

class pycozmo.event.EvtRobotOrientationChange
    Bases: pycozmo.event.Event
    Triggered when the robot orientation changes.

class pycozmo.event.EvtAudioCompleted
    Bases: pycozmo.event.Event

class pycozmo.event.EvtAnimationCompleted
    Bases: pycozmo.event.Event

class pycozmo.event.EvtReactionTrigger
    Bases: pycozmo.event.Event

class pycozmo.event.EvtBehaviorDone
    Bases: pycozmo.event.Event

class pycozmo.event.Handler (f: Callable, one_shot: bool)
    Bases: object
    Event handler class.

class pycozmo.event.Dispatcher
    Bases: object
    Event dispatcher class.
    add_child_dispatcher (child)
    add_handler (event, f, one_shot=False)
    del_all_handlers ()
    del_child_dispatcher (child)
    del_handler (event, handler)
    dispatch (event, *args, **kwargs)
    wait_for (evt, timeout: Optional[float] = None) → None
```

10.18 pycozmo.exception

Exception declarations.

Exceptions

ConnectionTimeout	Connection timeout.
InvalidOperation	Invalid operation.

Continued on next page

Table 25 – continued from previous page

<i>NoSpace</i>	Out of space.
<i>PyCozmoConnectionError</i>	Base class for all PyCozmo connection exceptions.
<i>PyCozmoException</i>	Base class for all PyCozmo exceptions.
<i>ResourcesNotFound</i>	Cozmo resources not found.
<i>Timeout</i>	Operation timed out.

**exception** `pycozmo.exception.PyCozmoException`

Bases: `Exception`

Base class for all PyCozmo exceptions.

**args**

**with\_traceback()**

Exception.with\_traceback(tb) – set self.\_\_traceback\_\_ to tb and return self.

**exception** `pycozmo.exception.PyCozmoConnectionError`

Bases: `pycozmo.exception.PyCozmoException`

Base class for all PyCozmo connection exceptions.

**args**

**with\_traceback()**

Exception.with\_traceback(tb) – set self.\_\_traceback\_\_ to tb and return self.

**exception** `pycozmo.exception.ConnectionTimeout`

Bases: `pycozmo.exception.PyCozmoConnectionError`

Connection timeout.

**args**

**with\_traceback()**

Exception.with\_traceback(tb) – set self.\_\_traceback\_\_ to tb and return self.

**exception** `pycozmo.exception.Timeout`

Bases: `pycozmo.exception.PyCozmoException`

Operation timed out.

**args**

**with\_traceback()**

Exception.with\_traceback(tb) – set self.\_\_traceback\_\_ to tb and return self.

**exception** `pycozmo.exception.NoSpace`

Bases: `pycozmo.exception.PyCozmoException`

Out of space.

**args**

**with\_traceback()**

Exception.with\_traceback(tb) – set self.\_\_traceback\_\_ to tb and return self.

## 10.19 pycozmo.filter

ID filtering for logging.

## Classes

---

*Filter()*

---

```
class pycozmo.filter.Filter
    Bases: object

    allow_ids (ids: Set[int]) → None
    deny_ids (ids: Set[int]) → None
    filter (target_id: int) → bool
```

## 10.20 pycozmo.frame

Cozmo protocol frame representation and encoding and decoding.

### Classes

---

<i>Frame</i> (type_id, first_seq, seq, ack, pkts)	Cozmo protocol frame.
---	-----------------------

---

```
class pycozmo.frame.Frame (type_id: pycozmo.protocol_ast.FrameType, first_seq: int, seq: int, ack:
                           int, pkts: List[pycozmo.protocol_base.Packet])
    Bases: object
    Cozmo protocol frame.
    ack
    first_seq
    classmethod from_bytes (buffer: bytes) → pycozmo.frame.Frame
    classmethod from_reader (reader: pycozmo.protocol_utils.BinaryReader) → py-
                           cozmo.frame.Frame
    pkts
    seq
    to_bytes () → bytes
    to_writer (writer: pycozmo.protocol_utils.BinaryWriter) → None
    type
```

## 10.21 pycozmo.image\_encoder

Cozmo image run-length encoding and decoding.

### Functions

---

```
image_to_str(image)
render(image)
str_to_image(sim)
```

---

## Classes

---

```
ImageDecoder(buffer)
ImageEncoder(im)
```

---

```
pycozmo.image_encoder.render(image: bytes) → None
```

```
pycozmo.image_encoder.image_to_str(image)
```

```
pycozmo.image_encoder.str_to_image(sim: str) → <module 'PIL.Image' from
'/home/docs/checkouts/readthedocs.org/user_builds/pycozmo/envs/latest/lib/python
packages/PIL/Image.py'>
```

```
class pycozmo.image_encoder.ImageDecoder(buffer: bytes)
```

```
    Bases: object
```

```
    decode() → bytes
```

```
class pycozmo.image_encoder.ImageEncoder(im: <module 'PIL.Image' from
'/home/docs/checkouts/readthedocs.org/user_builds/pycozmo/envs/latest/l
packages/PIL/Image.py'>)
```

```
    Bases: object
```

```
    encode() → bytearray
```

## 10.22 pycozmo.lights

Helper routines for working with colors and lights.

### Classes

---

```
Color(int_color, rgb, int, int] = None, name)          A Color to be used with a Light.
```

---

```
class pycozmo.lights.Color(int_color: Optional[int] = None, rgb: Optional[Tuple[int, int, int]] =
None, name: str = <class 'NoneType'>)
```

```
    Bases: object
```

```
    A Color to be used with a Light.
```

```
    Either int_color or rgb may be used to specify the actual color. Any alpha components (from int_color) are
    ignored - all colors are fully opaque.
```

```
    Args: int_color (int): A 32 bit value holding the binary RGBA value. rgb (tuple): A tuple holding the integer
    values from 0-255 for (red, green, blue) name (str): A name to assign to this color
```

```
    classmethod from_int16(value: int) → pycozmo.lights.Color
```

```
    int_color
```

```
    to_int16() → int
```



```
pycozmo.lights.green = Color(name=green, int_color=0x00ff00ff)
    Green color.
```

```
pycozmo.lights.red = Color(name=red, int_color=0xff0000ff)
    Red color.
```

```
pycozmo.lights.blue = Color(name=blue, int_color=0x0000ffff)
    Blue color.
```

```
pycozmo.lights.white = Color(name=white, int_color=0xffffffff)
    White color.
```

```
pycozmo.lights.off = Color(name=off, int_color=0x00000000)
    Off/no color.
```

```
pycozmo.lights.green_light = LightState(on_color=992, off_color=992, on_frames=0, off_frames=0)
    Green light.
```

```
pycozmo.lights.red_light = LightState(on_color=31744, off_color=31744, on_frames=0, off_frames=0)
    Red light.
```

```
pycozmo.lights.blue_light = LightState(on_color=31, off_color=31, on_frames=0, off_frames=0)
    Blue light.
```

```
pycozmo.lights.white_light = LightState(on_color=32767, off_color=32767, on_frames=0, off_frames=0)
    White light.
```

```
pycozmo.lights.off_light = LightState(on_color=0, off_color=0, on_frames=0, off_frames=0, t=0)
    Off/no light.
```

## 10.23 pycozmo.object

Cozmo objects (cubes, platforms, etc.).

### Classes

<code>Object</code> (factory_id, object_type)	Object representation.
---	------------------------

```
class pycozmo.object.Object (factory_id: int, object_type: pycozmo.protocol_encoder.ObjectType)
```

Bases: `object`

Object representation.

## 10.24 pycozmo.procedural\_face

Cozmo procedural face rendering.

### Functions

<code>interpolate</code> (from_face, to_face, steps)	Given two ProceduralFace objects, generate interpolated ProceduralFace objects in a number of steps.
--	--

## Classes

<code>ProceduralBase(params, offset, width, height)</code>	
<code><i>ProceduralEye</i>(params, offset, x_offset, ...)</code>	
<code><i>ProceduralFace</i>(params, width, height)</code>	
<code>ProceduralFaceGenerator()</code>	A generator class to produce eye animation.
<code><i>ProceduralLid</i>(params, offset, y_offset, ...)</code>	

```
class pycozmo.procedural_face.ProceduralLid(params: List[float], offset: int, y_offset:
                                             float, angle_offset: float, width: int, height:
                                             int)
```

```
    Bases: pycozmo.procedural_face.ProceduralBase
```

```
    angle
```

```
    angle_offset
```

```
    bend
```

```
    eye_height
```

```
    eye_width
```

```
    classmethod get_black(width, height)
```

```
    half_eye_height
```

```
    half_eye_width
```

```
    height
```

```
    offset
```

```
    params
```

```
    render(im: <module 'PIL.Image' from '/home/docs/checkouts/readthedocs.org/user_builds/pycozmo/envs/latest/lib/python3.7/
              packages/PIL/Image.py'>) → None
```

```
    scale_factor_lid_bend
```

```
    scale_factor_lid_height
```

```
    width
```

```
    y
```

```
    y_offset
```

```
class pycozmo.procedural_face.ProceduralEye(params: List[float], offset: int, x_offset: float
                                             = 0.0, width: int = 128, height: int = 64)
```

```
    Bases: pycozmo.procedural_face.ProceduralBase
```

```
    angle
```

```
    center_x
```

```
    center_y
```

```
    corner_radius
```

```
    eye_height
```

```
    eye_width
```

```
    half_eye_height
```

```

half_eye_width
height
lids
lower_inner_radius_x
lower_inner_radius_y
lower_outer_radius_x
lower_outer_radius_y
offset
params
render(im: <module 'PIL.Image' from '/home/docs/checkouts/readthedocs.org/user_builds/pycozmo/envs/latest/lib/python3.7/
packages/PIL/Image.py'>) → None
scale_factor_lid_bend
scale_factor_lid_height
scale_x
scale_y
upper_inner_radius_x
upper_inner_radius_y
upper_outer_radius_x
upper_outer_radius_y
width
x_offset

class pycozmo.procedural_face.ProceduralFace(params: Optional[List[float]] = None,
                                              width: int = 128, height: int = 64)
    Bases: pycozmo.procedural_face.ProceduralBase
    angle
    center_x
    center_y
    eye_height
    eye_width
    eyes
    half_eye_height
    half_eye_width
    height
    offset
    params
    render() → <module 'PIL.Image' from '/home/docs/checkouts/readthedocs.org/user_builds/pycozmo/envs/latest/lib/python3.7/
packages/PIL/Image.py'>
    scale_factor_lid_bend

```

**scale\_factor\_lid\_height**

**scale\_x**

**scale\_y**

**width**

`pycozmo.procedural_face.interpolate` (*from\_face*: `pycozmo.procedural_face.ProceduralFace`,  
*to\_face*: `pycozmo.procedural_face.ProceduralFace`,  
*steps*: `int`) → `Generator[pycozmo.procedural_face.ProceduralFace, None, None]`

Given two `ProceduralFace` objects, generate interpolated `ProceduralFace` objects in a number of steps.

## 10.25 pycozmo.protocol\_ast

Cozmo protocol abstract syntax tree (AST) types.

### Classes

<code>Argument(name, description, default)</code>	Base class for packet arguments.
<code>BoolArgument(name, description, default)</code>	8-bit boolean.
<code>Command(packet_id, name, group, description, ...)</code>	Command packet.
<code>Connect(description)</code>	Connect packet.
<code>Disconnect(description)</code>	Disconnect packet.
<code>DoubleArgument(name, description, default)</code>	64-bit floating point number.
<code>Enum(name, description, members, base)</code>	Enumeration.
<code>EnumArgument(name, enum_type, description, ...)</code>	Enumeration argument.
<code>EnumMember(name, value, description)</code>	Enumeration member.
<code>Event(packet_id, name, group, description, ...)</code>	Event packet.
<code>FArrayArgument(name, description, data_type, ...)</code>	Fixed-length array.
<code>FloatArgument(name, description, default)</code>	32-bit floating point number.
<code>FrameType</code>	Frame type enumeration.
<code>Int16Argument(name, description, default)</code>	16-bit signed integer.
<code>Int32Argument(name, description, default)</code>	32-bit signed integer.
<code>Int8Argument(name, description, default)</code>	8-bit signed integer.
<code>IntArgument(name, description, default)</code>	Base class for signed integers.
<code>Keyframe(description)</code>	Keyframe packet.
<code>Packet(packet_type, name, packet_id, group, ...)</code>	Base class for packets.
<code>PacketType</code>	Packet type enumeration.
<code>Ping(description)</code>	Ping packet.
<code>Protocol(enums, structs, packets)</code>	Protocol declaration.
<code>StringArgument(name, description, ...)</code>	String.
<code>Struct(name, description, arguments)</code>	Structure.
<code>UInt16Argument(name, description, default)</code>	16-bit unsigned integer.
<code>UInt32Argument(name, description, default)</code>	32-bit unsigned integer.
<code>UInt8Argument(name, description, default)</code>	8-bit unsigned integer.
<code>UIntArgument(name, description, default)</code>	Base class for unsigned integers.
<code>VArrayArgument(name, description, data_type, ...)</code>	Variable-length array.

```

class pycozmo.protocol_ast.FrameType
    Bases: enum.Enum

    Frame type enumeration.

    ENGINE = 7
    ENGINE_ACT = 4
    FIN = 3
    PING = 11
    RESET = 1
    RESET_ACK = 2
    ROBOT = 9

class pycozmo.protocol_ast.PacketType
    Bases: enum.Enum

    Packet type enumeration.

    COMMAND = 4
    CONNECT = 2
    DISCONNECT = 3
    EVENT = 5
    KEYFRAME = 10
    PING = 11
    UNKNOWN = -1

class pycozmo.protocol_ast.EnumMember(name: str, value: int, description: Optional[str] =
                                     None)
    Bases: object

    Enumeration member.

class pycozmo.protocol_ast.Enum(name: str, description: Optional[str] = None, members:
                                Optional[List[pycozmo.protocol_ast.EnumMember]] = None,
                                base: int = 10)
    Bases: object

    Enumeration.

class pycozmo.protocol_ast.Struct(name: Optional[str] = None, descrip-
                                tion: Optional[str] = None, arguments: Op-
                                tional[List[pycozmo.protocol_ast.Argument]] = None)
    Bases: pycozmo.protocol_ast.Argument

    Structure.

    type_hint()

class pycozmo.protocol_ast.Argument(name: Optional[str] = None, description: Optional[str]
                                = None, default: Any = None)
    Bases: abc.ABC

    Base class for packet arguments.

    type_hint() → Optional[str]

```

```
class pycozmo.protocol_ast.FloatArgument (name: Optional[str] = None, description: Optional[str] = None, default: float = 0.0)
    Bases: pycozmo.protocol\_ast.Argument
    32-bit floating point number.
    type_hint()

class pycozmo.protocol_ast.DoubleArgument (name: Optional[str] = None, description: Optional[str] = None, default: float = 0.0)
    Bases: pycozmo.protocol\_ast.Argument
    64-bit floating point number.
    type_hint()

class pycozmo.protocol_ast.BoolArgument (name: Optional[str] = None, description: Optional[str] = None, default: bool = False)
    Bases: pycozmo.protocol\_ast.Argument
    8-bit boolean.
    type_hint()

class pycozmo.protocol_ast.UIntArgument (name: Optional[str] = None, description: Optional[str] = None, default: Any = None)
    Bases: pycozmo.protocol\_ast.Argument, abc.ABC
    Base class for unsigned integers.
    type_hint() → Optional[str]

class pycozmo.protocol_ast.UInt8Argument (name: Optional[str] = None, description: Optional[str] = None, default: int = 0)
    Bases: pycozmo.protocol\_ast.UIntArgument
    8-bit unsigned integer.
    type_hint()

class pycozmo.protocol_ast.UInt16Argument (name: Optional[str] = None, description: Optional[str] = None, default: int = 0)
    Bases: pycozmo.protocol\_ast.UIntArgument
    16-bit unsigned integer.
    type_hint()

class pycozmo.protocol_ast.UInt32Argument (name: Optional[str] = None, description: Optional[str] = None, default: int = 0)
    Bases: pycozmo.protocol\_ast.UIntArgument
    32-bit unsigned integer.
    type_hint()

class pycozmo.protocol_ast.IntArgument (name: Optional[str] = None, description: Optional[str] = None, default: Any = None)
    Bases: pycozmo.protocol\_ast.Argument, abc.ABC
    Base class for signed integers.
    type_hint() → Optional[str]

class pycozmo.protocol_ast.Int8Argument (name: Optional[str] = None, description: Optional[str] = None, default: int = 0)
    Bases: pycozmo.protocol\_ast.IntArgument
```

8-bit signed integer.

**type\_hint()**

**class** pycozmo.protocol\_ast.**Int16Argument** (*name: Optional[str] = None, description: Optional[str] = None, default: int = 0*)

Bases: [pycozmo.protocol\\_ast.IntArgument](#)

16-bit signed integer.

**type\_hint()**

**class** pycozmo.protocol\_ast.**Int32Argument** (*name: Optional[str] = None, description: Optional[str] = None, default: int = 0*)

Bases: [pycozmo.protocol\\_ast.IntArgument](#)

32-bit signed integer.

**type\_hint()**

**class** pycozmo.protocol\_ast.**EnumArgument** (*name: str, enum\_type: pycozmo.protocol\_ast.Enum, description: Optional[str] = None, data\_type: Union[pycozmo.protocol\_ast.IntArgument, pycozmo.protocol\_ast.UIntArgument] = <pycozmo.protocol\_ast.Int8Argument object>, default: int = 0*)

Bases: [pycozmo.protocol\\_ast.Argument](#)

Enumeration argument.

**type\_hint()**

**class** pycozmo.protocol\_ast.**FArrayArgument** (*name: Optional[str] = None, description: Optional[str] = None, data\_type: pycozmo.protocol\_ast.Argument = <pycozmo.protocol\_ast.UInt8Argument object>, length: int = 0, default: Tuple = ()*)

Bases: [pycozmo.protocol\\_ast.Argument](#)

Fixed-length array.

**type\_hint()**

**class** pycozmo.protocol\_ast.**VArrayArgument** (*name: Optional[str] = None, description: Optional[str] = None, data\_type: pycozmo.protocol\_ast.Argument = <pycozmo.protocol\_ast.UInt8Argument object>, length\_type: pycozmo.protocol\_ast.Argument = <pycozmo.protocol\_ast.UInt16Argument object>, default: Tuple = ()*)

Bases: [pycozmo.protocol\\_ast.Argument](#)

Variable-length array.

**type\_hint()**

**class** pycozmo.protocol\_ast.**StringArgument** (*name: Optional[str] = None, description: Optional[str] = None, length\_type: pycozmo.protocol\_ast.Argument = <pycozmo.protocol\_ast.UInt16Argument object>, default: str = ""*)

Bases: [pycozmo.protocol\\_ast.Argument](#)

String.

**type\_hint()**

```
class pycozmo.protocol_ast.Packet (packet_type: pycozmo.protocol_ast.PacketType, name: str,  
                                   packet_id: Optional[int] = None, group: Optional[str] =  
                                   None, description: Optional[str] = None, arguments: Op-  
                                   tional[List[pycozmo.protocol_ast.Argument]] = None)
```

Bases: `pycozmo.protocol_ast.Struct`, `abc.ABC`

Base class for packets.

**type\_hint()**

```
class pycozmo.protocol_ast.Connect (description: Optional[str] = None)
```

Bases: `pycozmo.protocol_ast.Packet`

Connect packet.

**type\_hint()**

```
class pycozmo.protocol_ast.Disconnect (description: Optional[str] = None)
```

Bases: `pycozmo.protocol_ast.Packet`

Disconnect packet.

**type\_hint()**

```
class pycozmo.protocol_ast.Command (packet_id: int, name: str, group: Optional[str] = None,  
                                   description: Optional[str] = None, arguments: Op-  
                                   tional[List[pycozmo.protocol_ast.Argument]] = None)
```

Bases: `pycozmo.protocol_ast.Packet`

Command packet.

**type\_hint()**

```
class pycozmo.protocol_ast.Event (packet_id: int, name: str, group: Optional[str] =  
                                   None, description: Optional[str] = None, arguments: Op-  
                                   tional[List[pycozmo.protocol_ast.Argument]] = None)
```

Bases: `pycozmo.protocol_ast.Packet`

Event packet.

**type\_hint()**

```
class pycozmo.protocol_ast.Ping (description: Optional[str] = None)
```

Bases: `pycozmo.protocol_ast.Packet`

Ping packet.

**type\_hint()**

```
class pycozmo.protocol_ast.Keyframe (description: Optional[str] = None)
```

Bases: `pycozmo.protocol_ast.Packet`

Keyframe packet.

**type\_hint()**

```
class pycozmo.protocol_ast.Protocol (enums: List[pycozmo.protocol_ast.Enum], structs:  
                                   List[pycozmo.protocol_ast.Struct], packets:  
                                   List[pycozmo.protocol_ast.Packet])
```

Bases: `object`

Protocol declaration.



## 10.26 pycozmo.protocol\_base

Cozmo protocol implementation base.

### Classes

---

<i>Packet</i> (packet_type, packet_id)
<i>Struct</i>
<i>UnknownCommand</i> (packet_id, data)
<i>UnknownEvent</i> (packet_id, data)
<i>UnknownPacket</i> (packet_type, data, packet_id)

---

```

class pycozmo.protocol_base.Struct
    Bases: abc.ABC

    classmethod from_bytes (buffer: bytes) → pycozmo.protocol_base.Struct
    classmethod from_reader (reader: pycozmo.protocol_utils.BinaryReader) → py-
        cozmo.protocol_base.Struct

    to_bytes () → bytes
    to_writer (writer: pycozmo.protocol_utils.BinaryWriter) → None

class pycozmo.protocol_base.Packet (packet_type: pycozmo.protocol_ast.PacketType,
    packet_id: Optional[int] = None)
    Bases: pycozmo.protocol_base.Struct, abc.ABC

    ack

    classmethod from_bytes (buffer: bytes) → pycozmo.protocol_base.Struct
    classmethod from_reader (reader: pycozmo.protocol_utils.BinaryReader) → py-
        cozmo.protocol_base.Struct

    id

    is_from_engine () → bool
    is_from_robot () → bool
    is_oob () → bool

    seq

    to_bytes () → bytes
    to_writer (writer: pycozmo.protocol_utils.BinaryWriter) → None

    type

class pycozmo.protocol_base.UnknownPacket (packet_type: pycozmo.protocol_ast.PacketType,
    data: bytes, packet_id: Optional[int] = None)
    Bases: pycozmo.protocol_base.Packet

    ack

    data

    classmethod from_bytes (buffer)
    classmethod from_reader (reader)

```

```
    id
    is_from_engine() → bool
    is_from_robot() → bool
    is_oob() → bool
    seq
    to_bytes()
    to_writer(writer)
    type
class pycozmo.protocol_base.UnknownCommand(packet_id: int, data: bytes = b")
    Bases: pycozmo.protocol_base.UnknownPacket
    ack
    data
    classmethod from_bytes(buffer)
    classmethod from_reader(reader)
    id
    is_from_engine() → bool
    is_from_robot() → bool
    is_oob() → bool
    seq
    to_bytes()
    to_writer(writer)
    type
class pycozmo.protocol_base.UnknownEvent(packet_id: int, data: bytes = b")
    Bases: pycozmo.protocol_base.UnknownPacket
    ack
    data
    classmethod from_bytes(buffer)
    classmethod from_reader(reader)
    id
    is_from_engine() → bool
    is_from_robot() → bool
    is_oob() → bool
    seq
    to_bytes()
    to_writer(writer)
    type
```

## 10.27 pycozmo.protocol\_declaration

Cozmo protocol abstract syntax tree (AST) declaration.

```
pycozmo.protocol_declaration.PROTOCOL = <pycozmo.protocol_ast.Protocol object>
    Cozmo protocol declaration.
```

## 10.28 pycozmo.protocol\_encoder

Cozmo protocol packet encoder classes, based on protocol version 2381.

Generated from protocol\_declaration.py by protocol\_generator.py .

Do not modify.

### Classes

<i>AbortAnimation()</i>	
<i>AcknowledgeAction</i> ([action_id])	
<i>AnimBackpackLights</i> ([colors])	
<i>AnimBody</i> ([speed, unknown])	
<i>AnimHead</i> ([duration_ms, variability_deg, ...])	
<i>AnimLift</i> ([duration_ms, variability_mm, ...])	
<i>AnimationEnded</i> ([anim_id])	
<i>AnimationStarted</i> ([anim_id])	
<i>AnimationState</i> ([timestamp, ...])	
<i>AppendPathSegArc</i> ([center_x, center_y, ...])	
<i>AppendPathSegLine</i> ([from_x, from_y, to_x, ...])	
<i>AppendPathSegPointTurn</i> ([x, y, angle_rad, ...])	
<i>BodyColor</i>	An enumeration.
<i>BodyInfo</i> ([serial_number, body_hw_version, ...])	
<i>ButtonPressed</i> ([pressed])	
<i>ClearPath</i> ([unknown])	
<i>Connect</i> ()	
<i>CubeId</i> ([object_id, rotation_period_frames])	
<i>CubeLights</i> ([states])	
<i>DebugData</i> ([format_id, unused, name_id, ...])	
<i>Disconnect</i> ()	
<i>DisplayImage</i> ([image])	
<i>DriveWheels</i> ([wheel_speed_mmpps, ...])	
<i>Enable</i> ()	
<i>EnableAnimationState</i> ()	
<i>EnableCamera</i> ([image_send_mode, im- age_resolution])	
<i>EnableColorImages</i> ([enable])	
<i>EnableStopOnCliff</i> ([enable])	
<i>EndAnimation</i> ()	
<i>ExecutePath</i> ([event_id, unknown])	
<i>FallingStarted</i> ([unknown])	
<i>FallingStopped</i> ([unknown, duration_ms, ...])	

Continued on next page

Table 36 – continued from previous page

<i>FirmwareSignature</i> ([unknown, signature])	
<i>FirmwareUpdate</i> ([chunk_id, data])	
<i>FirmwareUpdateResult</i> ([byte_count, chunk_id, ...])	
<i>HardwareInfo</i> ([serial_number_head, unknown1, ...])	
<i>ImageChunk</i> ([frame_timestamp, image_id, ...])	
<i>ImageEncoding</i>	An enumeration.
<i>ImageImuData</i> ([image_id, rate_x, rate_y, ...])	
<i>ImageResolution</i>	An enumeration.
<i>ImageSendMode</i>	An enumeration.
<i>Keyframe</i> ()	
<i>LightState</i> ([on_color, off_color, on_frames, ...])	
<i>LightStateCenter</i> ([states, unknown])	
<i>LightStateSide</i> ([states, unknown])	
<i>MotorCalibration</i> ([motor_id, calib_started, ...])	
<i>MotorID</i>	An enumeration.
<i>MoveHead</i> ([speed_rad_per_sec])	
<i>MoveLift</i> ([speed_rad_per_sec])	
<i>NvEntryTag</i>	An enumeration.
<i>NvOperation</i>	An enumeration.
<i>NvResult</i>	An enumeration.
<i>NvStorageOp</i> ([tag, length, op, unknown, data])	
<i>NvStorageOpResult</i> ([tag, length, op, result, ...])	
<i>ObjectAccel</i> ([timestamp, object_id, accel_x, ...])	
<i>ObjectAvailable</i> ([factory_id, object_type, rssi])	
<i>ObjectConnect</i> ([factory_id, connect])	
<i>ObjectConnectionState</i> ([object_id, ...])	
<i>ObjectMoved</i> ([timestamp, object_id, ...])	
<i>ObjectPowerLevel</i> ([object_id, ...])	
<i>ObjectStoppedMoving</i> ([timestamp, object_id])	
<i>ObjectTapFiltered</i> ([timestamp, object_id, ...])	
<i>ObjectTapped</i> ([timestamp, object_id, ...])	
<i>ObjectType</i>	An enumeration.
<i>ObjectUpAxisChanged</i> ([timestamp, object_id, axis])	
<i>OutputAudio</i> ([samples])	
<i>OutputSilence</i> ()	
<i>PathEventType</i>	An enumeration.
<i>PathFollowingEvent</i> ([event_id, event_type])	
<i>PathSegmentSpeed</i> ([speed_mmpps, accel_mmpps2, ...])	
<i>Ping</i> ([time_sent_ms, counter, last, unknown])	
<i>RecordHeading</i> ()	
<i>RobotDelocalized</i> ()	
<i>RobotPoked</i> ()	
<i>RobotState</i> ([timestamp, pose_frame_id, ...])	
<i>SetAccessoryDiscovery</i> ([enable])	
<i>SetCameraParams</i> ([gain, exposure_ms, ...])	
<i>SetHeadAngle</i> ([angle_rad, ...])	

Continued on next page

Table 36 – continued from previous page

<i>SetHeadLight</i> ([enable])	
<i>SetLiftHeight</i> ([height_mm, ...])	
<i>SetOrigin</i> ([unknown0, pose_frame_id, ...])	
<i>SetRobotVolume</i> ([level])	
<i>ShutdownRobot</i> ()	
<i>StartAnimation</i> ([anim_id])	
<i>StartMotorCalibration</i> ([head, lift])	
<i>StopAllMotors</i> ()	
<i>StreamObjectAccel</i> ([object_id, enable])	
<i>SyncTime</i> ([timestamp, unknown])	
<i>TrimPath</i> ([head, tail])	
<i>TurnInPlace</i> ([angle_rad, speed_rad_per_sec, ...])	
<i>TurnInPlaceAtSpeed</i> ([wheel_speed_mmpps, ...])	
<i>TurnToRecordedHeading</i> ()	
<i>UpAxis</i>	An enumeration.
<i>WifiOff</i> ([enable])	

```

class pycozmo.protocol_encoder.AbortAnimation
    Bases: pycozmo.protocol_base.Packet
    ack
    classmethod from_bytes (buffer)
    classmethod from_reader (reader)
    id
    is_from_engine () → bool
    is_from_robot () → bool
    is_oob () → bool
    seq
    to_bytes ()
    to_writer (writer)
    type

class pycozmo.protocol_encoder.AcknowledgeAction (action_id=0)
    Bases: pycozmo.protocol_base.Packet
    ack
    action_id
    classmethod from_bytes (buffer)
    classmethod from_reader (reader)
    id
    is_from_engine () → bool
    is_from_robot () → bool
    is_oob () → bool
    seq

```

```
    to_bytes()
    to_writer(writer)
    type
class pycozmo.protocol_encoder.AnimBackpackLights(colors=())
    Bases: pycozmo.protocol_base.Packet
    ack
    colors
    classmethod from_bytes(buffer)
    classmethod from_reader(reader)
    id
    is_from_engine() → bool
    is_from_robot() → bool
    is_oob() → bool
    seq
    to_bytes()
    to_writer(writer)
    type
class pycozmo.protocol_encoder.AnimBody(speed=0, unknown=0)
    Bases: pycozmo.protocol_base.Packet
    ack
    classmethod from_bytes(buffer)
    classmethod from_reader(reader)
    id
    is_from_engine() → bool
    is_from_robot() → bool
    is_oob() → bool
    seq
    speed
    to_bytes()
    to_writer(writer)
    type
    unknown
class pycozmo.protocol_encoder.AnimHead(duration_ms=0, variability_deg=0, angle_deg=0)
    Bases: pycozmo.protocol_base.Packet
    ack
    angle_deg
    duration_ms
```

```

    classmethod from_bytes (buffer)
    classmethod from_reader (reader)
    id
    is_from_engine () → bool
    is_from_robot () → bool
    is_oob () → bool
    seq
    to_bytes ()
    to_writer (writer)
    type
    variability_deg
class pycozmo.protocol_encoder.AnimationLift (duration_ms=0,          variability_mm=0,
                                              height_mm=0)
    Bases: pycozmo.protocol_base.Packet
    ack
    duration_ms
    classmethod from_bytes (buffer)
    classmethod from_reader (reader)
    height_mm
    id
    is_from_engine () → bool
    is_from_robot () → bool
    is_oob () → bool
    seq
    to_bytes ()
    to_writer (writer)
    type
    variability_mm
class pycozmo.protocol_encoder.AnimationEnded (anim_id=0)
    Bases: pycozmo.protocol_base.Packet
    ack
    anim_id
    classmethod from_bytes (buffer)
    classmethod from_reader (reader)
    id
    is_from_engine () → bool
    is_from_robot () → bool

```

```
    is_oob() → bool
    seq
    to_bytes()
    to_writer(writer)
    type
class pycozmo.protocol_encoder.AnimationStarted(anim_id=0)
    Bases: pycozmo.protocol_base.Packet
    ack
    anim_id
    classmethod from_bytes(buffer)
    classmethod from_reader(reader)
    id
    is_from_engine() → bool
    is_from_robot() → bool
    is_oob() → bool
    seq
    to_bytes()
    to_writer(writer)
    type
class pycozmo.protocol_encoder.AnimationState(timestamp=0,
                                              num_anim_bytes_played=0,
                                              num_audio_frames_played=0,
                                              enabled_anim_tracks=0,
                                              client_drop_count=0)
    Bases: pycozmo.protocol_base.Packet
    ack
    client_drop_count
    enabled_anim_tracks
    classmethod from_bytes(buffer)
    classmethod from_reader(reader)
    id
    is_from_engine() → bool
    is_from_robot() → bool
    is_oob() → bool
    num_anim_bytes_played
    num_audio_frames_played
    seq
    tag
```



```

    timestamp
    to_bytes()
    to_writer(writer)
    type
class pycozmo.protocol_encoder.AppendPathSegArc(center_x=0.0, center_y=0.0, radius_mm=0.0, start_angle_rad=0.0, sweep_rad=0.0, speed_mmmps=0.0, accel_mmmps2=0.0, decel_mmmps2=0.0)

    Bases: pycozmo.protocol_base.Packet
    accel_mmmps2
    ack
    center_x
    center_y
    decel_mmmps2
    classmethod from_bytes(buffer)
    classmethod from_reader(reader)
    id
    is_from_engine() → bool
    is_from_robot() → bool
    is_oob() → bool
    radius_mm
    seq
    speed_mmmps
    start_angle_rad
    sweep_rad
    to_bytes()
    to_writer(writer)
    type
class pycozmo.protocol_encoder.AppendPathSegLine(from_x=0.0, from_y=0.0, to_x=0.0, to_y=0.0, speed_mmmps=0.0, accel_mmmps2=0.0, decel_mmmps2=0.0)

    Bases: pycozmo.protocol_base.Packet
    accel_mmmps2
    ack
    decel_mmmps2
    classmethod from_bytes(buffer)
    classmethod from_reader(reader)
    from_x
    from_y

```

```
id
is_from_engine() → bool
is_from_robot() → bool
is_oob() → bool
seq
speed_mmps
to_bytes()
to_writer(writer)
to_x
to_y
type
class pycozmo.protocol_encoder.AppendPathSegPointTurn(x=0.0, y=0.0, angle_rad=0.0,
angle_tolerance_rad=0.0,
speed_mmps=0.0,          ac-
cel_mmps2=0.0,           de-
cel_mmps2=0.0,           un-
known=False)

Bases: pycozmo.protocol_base.Packet
accel_mmps2
ack
angle_rad
angle_tolerance_rad
decel_mmps2
classmethod from_bytes(buffer)
classmethod from_reader(reader)
id
is_from_engine() → bool
is_from_robot() → bool
is_oob() → bool
seq
speed_mmps
to_bytes()
to_writer(writer)
type
unknown
x
y
```

```

class pycozmo.protocol_encoder.BodyColor
    Bases: enum.Enum

    An enumeration.

    CE_LM_v15 = 3

    DEV = 5

    LE_BL_v16 = 4

    RESERVED = 1

    UNKNOWN = -1

    WHITE_v10 = 0

    WHITE_v15 = 2

class pycozmo.protocol_encoder.BodyInfo(serial_number=0, body_hw_version=0,
                                         body_color=-1)
    Bases: pycozmo.protocol_base.Packet

    ack

    body_color

    body_hw_version

    classmethod from_bytes(buffer)

    classmethod from_reader(reader)

    id

    is_from_engine() → bool

    is_from_robot() → bool

    is_oob() → bool

    seq

    serial_number

    to_bytes()

    to_writer(writer)

    type

class pycozmo.protocol_encoder.ButtonPressed(pressed=False)
    Bases: pycozmo.protocol_base.Packet

    ack

    classmethod from_bytes(buffer)

    classmethod from_reader(reader)

    id

    is_from_engine() → bool

    is_from_robot() → bool

    is_oob() → bool

    pressed

    seq

```

```
    to_bytes()
    to_writer(writer)
    type

class pycozmo.protocol_encoder.ClearPath(unknown=0)
    Bases: pycozmo.protocol_base.Packet
    ack
    classmethod from_bytes(buffer)
    classmethod from_reader(reader)
    id
    is_from_engine() → bool
    is_from_robot() → bool
    is_oob() → bool
    seq
    to_bytes()
    to_writer(writer)
    type
    unknown

class pycozmo.protocol_encoder.Connect
    Bases: pycozmo.protocol_base.Packet
    ack
    classmethod from_bytes(buffer)
    classmethod from_reader(reader)
    id
    is_from_engine() → bool
    is_from_robot() → bool
    is_oob() → bool
    seq
    to_bytes()
    to_writer(writer)
    type

class pycozmo.protocol_encoder.CubeId(object_id=0, rotation_period_frames=0)
    Bases: pycozmo.protocol_base.Packet
    ack
    classmethod from_bytes(buffer)
    classmethod from_reader(reader)
    id
    is_from_engine() → bool
```

```

    is_from_robot() → bool
    is_oob() → bool
    object_id
    rotation_period_frames
    seq
    to_bytes()
    to_writer(writer)
    type
class pycozmo.protocol_encoder.CubeLights(states=())
    Bases: pycozmo.protocol_base.Packet
    ack
    classmethod from_bytes(buffer)
    classmethod from_reader(reader)
    id
    is_from_engine() → bool
    is_from_robot() → bool
    is_oob() → bool
    seq
    states
    to_bytes()
    to_writer(writer)
    type
class pycozmo.protocol_encoder.DebugData(format_id=0, unused=0, name_id=0, level=0,
                                           args=())
    Bases: pycozmo.protocol_base.Packet
    ack
    args
    format_id
    classmethod from_bytes(buffer)
    classmethod from_reader(reader)
    id
    is_from_engine() → bool
    is_from_robot() → bool
    is_oob() → bool
    level
    name_id
    seq

```

```
    to_bytes()
    to_writer(writer)
    type
    unused

class pycozmo.protocol_encoder.Disconnect
    Bases: pycozmo.protocol_base.Packet

    ack

    classmethod from_bytes(buffer)
    classmethod from_reader(reader)

    id

    is_from_engine() → bool
    is_from_robot() → bool
    is_oob() → bool

    seq

    to_bytes()
    to_writer(writer)
    type

class pycozmo.protocol_encoder.DisplayImage(image=())
    Bases: pycozmo.protocol_base.Packet

    ack

    classmethod from_bytes(buffer)
    classmethod from_reader(reader)

    id

    image

    is_from_engine() → bool
    is_from_robot() → bool
    is_oob() → bool

    seq

    to_bytes()
    to_writer(writer)
    type

class pycozmo.protocol_encoder.DriveWheels(lwheel_speed_mmps=0.0,
                                             rwheel_speed_mmps=0.0,
                                             lwheel_accel_mmmps2=0.0,
                                             rwheel_accel_mmmps2=0.0)
    Bases: pycozmo.protocol_base.Packet

    ack

    classmethod from_bytes(buffer)
```

```

    classmethod from_reader(reader)
    id
    is_from_engine() → bool
    is_from_robot() → bool
    is_oob() → bool
    lwheel_accel_mmps2
    lwheel_speed_mmps
    rwheel_accel_mmps2
    rwheel_speed_mmps
    seq
    to_bytes()
    to_writer(writer)
    type
class pycozmo.protocol_encoder.Enable
    Bases: pycozmo.protocol_base.Packet
    ack
    classmethod from_bytes(buffer)
    classmethod from_reader(reader)
    id
    is_from_engine() → bool
    is_from_robot() → bool
    is_oob() → bool
    seq
    to_bytes()
    to_writer(writer)
    type
class pycozmo.protocol_encoder.EnableAnimationState
    Bases: pycozmo.protocol_base.Packet
    ack
    classmethod from_bytes(buffer)
    classmethod from_reader(reader)
    id
    is_from_engine() → bool
    is_from_robot() → bool
    is_oob() → bool
    seq
    to_bytes()

```

```
    to_writer(writer)

    type

class pycozmo.protocol_encoder.EnableCamera (image_send_mode=1, image_resolution=4)
    Bases: pycozmo.protocol_base.Packet

    ack

    classmethod from_bytes (buffer)

    classmethod from_reader (reader)

    id

    image_resolution

    image_send_mode

    is_from_engine () → bool

    is_from_robot () → bool

    is_oob () → bool

    seq

    to_bytes ()

    to_writer(writer)

    type

class pycozmo.protocol_encoder.EnableColorImages (enable=False)
    Bases: pycozmo.protocol_base.Packet

    ack

    enable

    classmethod from_bytes (buffer)

    classmethod from_reader (reader)

    id

    is_from_engine () → bool

    is_from_robot () → bool

    is_oob () → bool

    seq

    to_bytes ()

    to_writer(writer)

    type

class pycozmo.protocol_encoder.EnableStopOnCliff (enable=False)
    Bases: pycozmo.protocol_base.Packet

    ack

    enable

    classmethod from_bytes (buffer)

    classmethod from_reader (reader)
```



```

    id
    is_from_engine () → bool
    is_from_robot () → bool
    is_oob () → bool
    seq
    to_bytes ()
    to_writer (writer)
    type
class pycozmo.protocol_encoder.EndAnimation
    Bases: pycozmo.protocol_base.Packet
    ack
    classmethod from_bytes (buffer)
    classmethod from_reader (reader)
    id
    is_from_engine () → bool
    is_from_robot () → bool
    is_oob () → bool
    seq
    to_bytes ()
    to_writer (writer)
    type
class pycozmo.protocol_encoder.ExecutePath (event_id=0, unknown=False)
    Bases: pycozmo.protocol_base.Packet
    ack
    event_id
    classmethod from_bytes (buffer)
    classmethod from_reader (reader)
    id
    is_from_engine () → bool
    is_from_robot () → bool
    is_oob () → bool
    seq
    to_bytes ()
    to_writer (writer)
    type
    unknown

```

```
class pycozmo.protocol_encoder.FallingStarted(unknown=0)
    Bases: pycozmo.protocol_base.Packet
    ack
    classmethod from_bytes(buffer)
    classmethod from_reader(reader)
    id
    is_from_engine() → bool
    is_from_robot() → bool
    is_oob() → bool
    seq
    to_bytes()
    to_writer(writer)
    type
    unknown

class pycozmo.protocol_encoder.FallingStopped(unknown=0, duration_ms=0, im-
    pact_intensity=0.0)
    Bases: pycozmo.protocol_base.Packet
    ack
    duration_ms
    classmethod from_bytes(buffer)
    classmethod from_reader(reader)
    id
    impact_intensity
    is_from_engine() → bool
    is_from_robot() → bool
    is_oob() → bool
    seq
    to_bytes()
    to_writer(writer)
    type
    unknown

class pycozmo.protocol_encoder.FirmwareSignature(unknown=0, signature="")
    Bases: pycozmo.protocol_base.Packet
    ack
    classmethod from_bytes(buffer)
    classmethod from_reader(reader)
    id
    is_from_engine() → bool
```

```

    is_from_robot() → bool
    is_oob() → bool
    seq
    signature
    to_bytes()
    to_writer(writer)
    type
    unknown

class pycozmo.protocol_encoder.FirmwareUpdate(chunk_id=0, data=())
    Bases: pycozmo.protocol_base.Packet
    ack
    chunk_id
    data
    classmethod from_bytes(buffer)
    classmethod from_reader(reader)
    id
    is_from_engine() → bool
    is_from_robot() → bool
    is_oob() → bool
    seq
    to_bytes()
    to_writer(writer)
    type

class pycozmo.protocol_encoder.FirmwareUpdateResult(byte_count=0, chunk_id=0, sta-
                                                    tus=0)
    Bases: pycozmo.protocol_base.Packet
    ack
    byte_count
    chunk_id
    classmethod from_bytes(buffer)
    classmethod from_reader(reader)
    id
    is_from_engine() → bool
    is_from_robot() → bool
    is_oob() → bool
    seq
    status

```

```
    to_bytes()
    to_writer(writer)
    type
class pycozmo.protocol_encoder.HardwareInfo(serial_number_head=0, unknown1=0, unknown2=0)
    Bases: pycozmo.protocol_base.Packet
    ack
    classmethod from_bytes(buffer)
    classmethod from_reader(reader)
    id
    is_from_engine() → bool
    is_from_robot() → bool
    is_oob() → bool
    seq
    serial_number_head
    to_bytes()
    to_writer(writer)
    type
    unknown1
    unknown2
class pycozmo.protocol_encoder.ImageChunk(frame_timestamp=0, image_id=0,
chunk_debug=0, image_encoding=0, image_resolution=0, image_chunk_count=0,
chunk_id=0, status=0, data=())
    Bases: pycozmo.protocol_base.Packet
    ack
    chunk_debug
    chunk_id
    data
    frame_timestamp
    classmethod from_bytes(buffer)
    classmethod from_reader(reader)
    id
    image_chunk_count
    image_encoding
    image_id
    image_resolution
    is_from_engine() → bool
```

```

    is_from_robot() → bool
    is_oob() → bool
    seq
    status
    to_bytes()
    to_writer(writer)
    type
class pycozmo.protocol_encoder.ImageEncoding
    Bases: enum.Enum
    An enumeration.
    BAYER = 4
    JPEGColor = 6
    JPEGColorHalfWidth = 7
    JPEGGray = 5
    JPEGMinimizedColor = 9
    JPEGMinimizedGray = 8
    NoneImageEncoding = 0
    RawGray = 1
    RawRGB = 2
    YUYV = 3
class pycozmo.protocol_encoder.ImageImuData(image_id=0, rate_x=0.0, rate_y=0.0,
                                             rate_z=0.0, line_2_number=0)
    Bases: pycozmo.protocol_base.Packet
    ack
    classmethod from_bytes(buffer)
    classmethod from_reader(reader)
    id
    image_id
    is_from_engine() → bool
    is_from_robot() → bool
    is_oob() → bool
    line_2_number
    rate_x
    rate_y
    rate_z
    seq
    to_bytes()

```

```
    to_writer(writer)

    type

class pycozmo.protocol_encoder.ImageResolution
    Bases: enum.Enum

    An enumeration.

    CVGA = 5

    ImageResolutionCount = 13

    ImageResolutionNone = 14

    QQQQVGA = 1

    QQQVGA = 2

    QQVGA = 3

    QUXGA = 12

    QVGA = 4

    QXGA = 11

    SVGA = 7

    SXGA = 9

    UXGA = 10

    VGA = 6

    VerificationSnapshot = 0

    XGA = 8

class pycozmo.protocol_encoder.ImageSendMode
    Bases: enum.Enum

    An enumeration.

    Off = 0

    SingleShot = 2

    Stream = 1

class pycozmo.protocol_encoder.Keyframe
    Bases: pycozmo.protocol_base.Packet

    ack

    classmethod from_bytes(buffer)

    classmethod from_reader(reader)

    id

    is_from_engine() → bool

    is_from_robot() → bool

    is_oob() → bool

    seq

    to_bytes()
```

```

    to_writer(writer)

    type

class pycozmo.protocol_encoder.LightState(on_color=0, off_color=0, on_frames=0,
                                           off_frames=0, transition_on_frames=0, transition_off_frames=0, offset=0)

    Bases: pycozmo.protocol_base.Struct

    classmethod from_bytes(buffer)

    classmethod from_reader(reader)

    off_color

    off_frames

    offset

    on_color

    on_frames

    to_bytes()

    to_writer(writer)

    transition_off_frames

    transition_on_frames

class pycozmo.protocol_encoder.LightStateCenter(states=(), unknown=0)

    Bases: pycozmo.protocol_base.Packet

    ack

    classmethod from_bytes(buffer)

    classmethod from_reader(reader)

    id

    is_from_engine() → bool

    is_from_robot() → bool

    is_oob() → bool

    seq

    states

    to_bytes()

    to_writer(writer)

    type

    unknown

class pycozmo.protocol_encoder.LightStateSide(states=(), unknown=0)

    Bases: pycozmo.protocol_base.Packet

    ack

    classmethod from_bytes(buffer)

    classmethod from_reader(reader)

    id

```

```
is_from_engine() → bool
is_from_robot() → bool
is_oob() → bool
seq
states
to_bytes()
to_writer(writer)
type
unknown

class pycozmo.protocol_encoder.MotorCalibration(motor_id=0,      calib_started=False,
                                                auto_started=False)
    Bases: pycozmo.protocol_base.Packet
    ack
    auto_started
    calib_started
    classmethod from_bytes(buffer)
    classmethod from_reader(reader)
    id
    is_from_engine() → bool
    is_from_robot() → bool
    is_oob() → bool
    motor_id
    seq
    to_bytes()
    to_writer(writer)
    type

class pycozmo.protocol_encoder.MotorID
    Bases: enum.Enum

    An enumeration.

    MOTOR_HEAD = 3
    MOTOR_LEFT_WHEEL = 0
    MOTOR_LIFT = 2
    MOTOR_RIGHT_WHEEL = 1

class pycozmo.protocol_encoder.MoveHead(speed_rad_per_sec=0.0)
    Bases: pycozmo.protocol_base.Packet
    ack
    classmethod from_bytes(buffer)
    classmethod from_reader(reader)
```



```

    id
    is_from_engine () → bool
    is_from_robot () → bool
    is_oob () → bool
    seq
    speed_rad_per_sec
    to_bytes ()
    to_writer (writer)
    type
class pycozmo.protocol_encoder.MoveLift (speed_rad_per_sec=0.0)
    Bases: pycozmo.protocol_base.Packet
    ack
    classmethod from_bytes (buffer)
    classmethod from_reader (reader)
    id
    is_from_engine () → bool
    is_from_robot () → bool
    is_oob () → bool
    seq
    speed_rad_per_sec
    to_bytes ()
    to_writer (writer)
    type
class pycozmo.protocol_encoder.NvEntryTag
    Bases: enum.Enum
    An enumeration.
    NvEntry_BirthCertificate = 2147483648
    NvEntry_CalibImage1 = 2147549184
    NvEntry_CalibImage2 = 2147614720
    NvEntry_CalibImage3 = 2147680256
    NvEntry_CalibImage4 = 2147745792
    NvEntry_CalibImage5 = 2147811328
    NvEntry_CalibImage6 = 2147876864
    NvEntry_CalibMetaInfo = 2147483652
    NvEntry_CalibPose = 2147483651
    NvEntry_CameraCalib = 2147483649
    NvEntry_CliffValOnDrop = 2147483655

```

```
NEntry_CliffValOnGround = 2147483656
NEntry_FACTORY_RESERVED1 = 1835008
NEntry_FACTORY_RESERVED2 = 1957888
NEntry_FaceAlbumData = 1589248
NEntry_FaceEnrollData = 1585152
NEntry_FactoryBaseTag = 909312
NEntry_FactoryBaseTagWithBCOffset = 909360
NEntry_FactoryLock = 2147483665
NEntry_GameSkillLevels = 1572864
NEntry_GameUnlocks = 1581056
NEntry_IMUAverages = 3221225476
NEntry_IMUIInfo = 2147483654
NEntry_Invalid = 4294967295
NEntry_InventoryData = 1658880
NEntry_LabAssignments = 1662976
NEntry_NEXT_SLOT = 1671168
NEntry_NurtureGameData = 1654784
NEntry_ObservedCubePose = 2147483653
NEntry_OnboardingData = 1576960
NEntry_PlaypenTestResults = 2147483664
NEntry_PrePlaypenCentroids = 3221225473
NEntry_PrePlaypenResults = 3221225472
NEntry_SavedCubeIDs = 1667072
NEntry_ToolCodeImageLeft = 2148532224
NEntry_ToolCodeImageRight = 2148597760
NEntry_ToolCodeInfo = 2147483650
NEntry_VersionMagic = 2147483666

class pycozmo.protocol_encoder.NvOperation
    Bases: enum.Enum
    An enumeration.

    NVOP_ERASE = 2
    NVOP_READ = 0
    NVOP_WIPEALL = 3
    NVOP_WRITE = 1

class pycozmo.protocol_encoder.NvResult
    Bases: enum.Enum
    An enumeration.
```

```

NV_BAD_ARGS = -6
NV_BUSY = -5
NV_CORRUPT = -9
NV_ERROR = -3
NV_LOOP = -8
NV_MORE = 3
NV_NOT_FOUND = -1
NV_NO_DO = 2
NV_NO_MEM = -7
NV_NO_ROOM = -2
NV_OKAY = 0
NV_SCHEDULED = 1
NV_TIMEOUT = -4
NV_UNKNOWN_4 = 4
NV_UNKNOWN_5 = 5
NV_UNKNOWN_6 = 6
NV_UNKNOWN_7 = 7
NV_UNKNOWN_8 = 8

class pycozmo.protocol_encoder.NvStorageOp (tag=4294967295, length=0, op=0, un-
                                             known=0, data=())
    Bases: pycozmo.protocol_base.Packet
    ack
    data
    classmethod from_bytes (buffer)
    classmethod from_reader (reader)
    id
    is_from_engine () → bool
    is_from_robot () → bool
    is_oob () → bool
    length
    op
    seq
    tag
    to_bytes ()
    to_writer (writer)
    type
    unknown

```

```
class pycozmo.protocol_encoder.NvStorageOpResult (tag=4294967295, length=0, op=0,
                                                result=0, data=())
    Bases: pycozmo.protocol_base.Packet
    ack
    data
    classmethod from_bytes (buffer)
    classmethod from_reader (reader)
    id
    is_from_engine () → bool
    is_from_robot () → bool
    is_oob () → bool
    length
    op
    result
    seq
    tag
    to_bytes ()
    to_writer (writer)
    type

class pycozmo.protocol_encoder.ObjectAccel (timestamp=0, object_id=0, accel_x=0.0, accel_y=0.0, accel_z=0.0)
    Bases: pycozmo.protocol_base.Packet
    accel_x
    accel_y
    accel_z
    ack
    classmethod from_bytes (buffer)
    classmethod from_reader (reader)
    id
    is_from_engine () → bool
    is_from_robot () → bool
    is_oob () → bool
    object_id
    seq
    timestamp
    to_bytes ()
    to_writer (writer)
    type
```

```

class pycozmo.protocol_encoder.ObjectAvailable (factory_id=0, object_type=-1, rssi=0)
    Bases: pycozmo.protocol_base.Packet

    ack
    factory_id
    classmethod from_bytes (buffer)
    classmethod from_reader (reader)
    id
    is_from_engine () → bool
    is_from_robot () → bool
    is_oob () → bool
    object_type
    rssi
    seq
    to_bytes ()
    to_writer (writer)
    type

class pycozmo.protocol_encoder.ObjectConnect (factory_id=0, connect=False)
    Bases: pycozmo.protocol_base.Packet

    ack
    connect
    factory_id
    classmethod from_bytes (buffer)
    classmethod from_reader (reader)
    id
    is_from_engine () → bool
    is_from_robot () → bool
    is_oob () → bool
    seq
    to_bytes ()
    to_writer (writer)
    type

class pycozmo.protocol_encoder.ObjectConnectionState (object_id=0,    factory_id=0,
                                                         object_type=-1,    con-
                                                         nected=False)
    Bases: pycozmo.protocol_base.Packet

    ack
    connected
    factory_id

```

```
classmethod from_bytes (buffer)
classmethod from_reader (reader)
id
is_from_engine () → bool
is_from_robot () → bool
is_oob () → bool
object_id
object_type
seq
to_bytes ()
to_writer (writer)
type
class pycozmo.protocol_encoder.ObjectMoved (timestamp=0,      object_id=0,      ac-
                                             tive_accel_x=0.0,      active_accel_y=0.0,
                                             active_accel_z=0.0, axis_of_accel=7)
Bases: pycozmo.protocol_base.Packet
ack
active_accel_x
active_accel_y
active_accel_z
axis_of_accel
classmethod from_bytes (buffer)
classmethod from_reader (reader)
id
is_from_engine () → bool
is_from_robot () → bool
is_oob () → bool
object_id
seq
timestamp
to_bytes ()
to_writer (writer)
type
class pycozmo.protocol_encoder.ObjectPowerLevel (object_id=0, missed_packets=0, bat-
                                                  tery_level=0)
Bases: pycozmo.protocol_base.Packet
ack
battery_level
```

```

    classmethod from_bytes (buffer)
    classmethod from_reader (reader)
    id
    is_from_engine () → bool
    is_from_robot () → bool
    is_oob () → bool
    missed_packets
    object_id
    seq
    to_bytes ()
    to_writer (writer)
    type
class pycozmo.protocol_encoder.ObjectStoppedMoving (timestamp=0, object_id=0)
    Bases: pycozmo.protocol_base.Packet
    ack
    classmethod from_bytes (buffer)
    classmethod from_reader (reader)
    id
    is_from_engine () → bool
    is_from_robot () → bool
    is_oob () → bool
    object_id
    seq
    timestamp
    to_bytes ()
    to_writer (writer)
    type
class pycozmo.protocol_encoder.ObjectTapFiltered (timestamp=0, object_id=0, time=0,
                                                    intensity=0)
    Bases: pycozmo.protocol_base.Packet
    ack
    classmethod from_bytes (buffer)
    classmethod from_reader (reader)
    id
    intensity
    is_from_engine () → bool
    is_from_robot () → bool

```

```
is_oob() → bool
object_id
seq
time
timestamp
to_bytes()
to_writer(writer)
type

class pycozmo.protocol_encoder.ObjectTapped(timestamp=0, object_id=0, num_taps=0,
                                             tap_time=0, tap_neg=0, tap_pos=0)
    Bases: pycozmo.protocol_base.Packet
    ack
    classmethod from_bytes(buffer)
    classmethod from_reader(reader)
    id
    is_from_engine() → bool
    is_from_robot() → bool
    is_oob() → bool
    num_taps
    object_id
    seq
    tap_neg
    tap_pos
    tap_time
    timestamp
    to_bytes()
    to_writer(writer)
    type

class pycozmo.protocol_encoder.ObjectType
    Bases: enum.Enum
    An enumeration.
    Block_LIGHTCUBE1 = 1
    Block_LIGHTCUBE2 = 2
    Block_LIGHTCUBE3 = 3
    Block_LIGHTCUBE_GHOST = 4
    Bridge_LONG = 10
    Bridge_SHORT = 11
```



```

Charger_Basic = 13
CliffDetection = 15
CollisionObstacle = 16
CustomFixedObstacle = 37
CustomType00 = 17
CustomType01 = 18
CustomType02 = 19
CustomType03 = 20
CustomType04 = 21
CustomType05 = 22
CustomType06 = 23
CustomType07 = 24
CustomType08 = 25
CustomType09 = 26
CustomType10 = 27
CustomType11 = 28
CustomType12 = 29
CustomType13 = 30
CustomType14 = 31
CustomType15 = 32
CustomType16 = 33
CustomType17 = 34
CustomType18 = 35
CustomType19 = 36
FlatMat_ANKI_LOGO_8BIT = 7
FlatMat_GEARS_4x4 = 5
FlatMat_LAVA_PLAYTEST = 8
FlatMat_LETTERS_4x4 = 6
InvalidObject = -1
Platform_LARGE = 9
ProxObstacle = 14
Ramp_Basic = 12
UnknownObject = 0

class pycozmo.protocol_encoder.ObjectUpAxisChanged(timestamp=0, object_id=0,
                                                    axis=7)
    Bases: pycozmo.protocol_base.Packet
    ack

```

```
axis
classmethod from_bytes (buffer)
classmethod from_reader (reader)
id
is_from_engine () → bool
is_from_robot () → bool
is_oob () → bool
object_id
seq
timestamp
to_bytes ()
to_writer (writer)
type
class pycozmo.protocol_encoder.OutputAudio (samples=())
  Bases: pycozmo.protocol_base.Packet
  ack
  classmethod from_bytes (buffer)
  classmethod from_reader (reader)
  id
  is_from_engine () → bool
  is_from_robot () → bool
  is_oob () → bool
  samples
  seq
  to_bytes ()
  to_writer (writer)
  type
class pycozmo.protocol_encoder.OutputSilence
  Bases: pycozmo.protocol_base.Packet
  ack
  classmethod from_bytes (buffer)
  classmethod from_reader (reader)
  id
  is_from_engine () → bool
  is_from_robot () → bool
  is_oob () → bool
  seq
```

```

    to_bytes()
    to_writer(writer)
    type

class pycozmo.protocol_encoder.PathEventType
    Bases: enum.Enum

    An enumeration.

    PATH_COMPLETED = 2
    PATH_INTERRUPTED = 1
    PATH_STARTED = 0

class pycozmo.protocol_encoder.PathFollowingEvent(event_id=0, event_type=0)
    Bases: pycozmo.protocol_base.Packet

    ack
    event_id
    event_type
    classmethod from_bytes(buffer)
    classmethod from_reader(reader)
    id
    is_from_engine() → bool
    is_from_robot() → bool
    is_oob() → bool
    seq
    to_bytes()
    to_writer(writer)
    type

class pycozmo.protocol_encoder.PathSegmentSpeed(speed_mmmps=0.0, accel_mmmps2=0.0,
                                                  decel_mmmps2=0.0)
    Bases: pycozmo.protocol_base.Struct

    accel_mmmps2
    decel_mmmps2
    classmethod from_bytes(buffer)
    classmethod from_reader(reader)
    speed_mmmps
    to_bytes()
    to_writer(writer)

class pycozmo.protocol_encoder.Ping(time_sent_ms=0.0, counter=0, last=0, unknown=0)
    Bases: pycozmo.protocol_base.Packet

    ack
    counter

```

```
    classmethod from_bytes (buffer)
    classmethod from_reader (reader)
    id
    is_from_engine () → bool
    is_from_robot () → bool
    is_oob () → bool
    last
    seq
    time_sent_ms
    to_bytes ()
    to_writer (writer)
    type
    unknown

class pycozmo.protocol_encoder.RecordHeading
    Bases: pycozmo.protocol_base.Packet
    ack
    classmethod from_bytes (buffer)
    classmethod from_reader (reader)
    id
    is_from_engine () → bool
    is_from_robot () → bool
    is_oob () → bool
    seq
    to_bytes ()
    to_writer (writer)
    type

class pycozmo.protocol_encoder.RobotDelocalized
    Bases: pycozmo.protocol_base.Packet
    ack
    classmethod from_bytes (buffer)
    classmethod from_reader (reader)
    id
    is_from_engine () → bool
    is_from_robot () → bool
    is_oob () → bool
    seq
    to_bytes ()
```

```

    to_writer(writer)

    type

class pycozmo.protocol_encoder.RobotPoked
    Bases: pycozmo.protocol_base.Packet

    ack

    classmethod from_bytes(buffer)

    classmethod from_reader(reader)

    id

    is_from_engine() → bool

    is_from_robot() → bool

    is_oob() → bool

    seq

    to_bytes()

    to_writer(writer)

    type

class pycozmo.protocol_encoder.RobotState(timestamp=0, pose_frame_id=0,
                                           pose_origin_id=0, pose_x=0.0, pose_y=0.0,
                                           pose_z=0.0, pose_angle_rad=0.0,
                                           pose_pitch_rad=0.0, lwheel_speed_mmmps=0.0,
                                           rwheel_speed_mmmps=0.0,
                                           head_angle_rad=0.0, lift_height_mm=0.0,
                                           accel_x=0.0, accel_y=0.0, accel_z=0.0,
                                           gyro_x=0.0, gyro_y=0.0, gyro_z=0.0, bat-
                                           tery_voltage=0.0, status=0, cliff_data_raw=(),
                                           backpack_touch_sensor_raw=0,
                                           curr_path_segment=0)

    Bases: pycozmo.protocol_base.Packet

    accel_x

    accel_y

    accel_z

    ack

    backpack_touch_sensor_raw

    battery_voltage

    cliff_data_raw

    curr_path_segment

    classmethod from_bytes(buffer)

    classmethod from_reader(reader)

    gyro_x

    gyro_y

    gyro_z

```

```
    head_angle_rad
    id
    is_from_engine() → bool
    is_from_robot() → bool
    is_oob() → bool
    lift_height_mm
    lwheel_speed_mmps
    pose_angle_rad
    pose_frame_id
    pose_origin_id
    pose_pitch_rad
    pose_x
    pose_y
    pose_z
    rwheel_speed_mmps
    seq
    status
    timestamp
    to_bytes()
    to_writer(writer)
    type

class pycozmo.protocol_encoder.SetAccessoryDiscovery(enable=False)
    Bases: pycozmo.protocol\_base.Packet
    ack
    enable
    classmethod from_bytes(buffer)
    classmethod from_reader(reader)
    id
    is_from_engine() → bool
    is_from_robot() → bool
    is_oob() → bool
    seq
    to_bytes()
    to_writer(writer)
    type
```

```

class pycozmo.protocol_encoder.SetCameraParams (gain=0.0,          exposure_ms=0,
                                                auto_exposure_enabled=False)

    Bases: pycozmo.protocol_base.Packet

    ack
    auto_exposure_enabled
    exposure_ms
    classmethod from_bytes (buffer)
    classmethod from_reader (reader)
    gain
    id
    is_from_engine () → bool
    is_from_robot () → bool
    is_oob () → bool
    seq
    to_bytes ()
    to_writer (writer)
    type

class pycozmo.protocol_encoder.SetHeadAngle (angle_rad=0.0,
                                              max_speed_rad_per_sec=15.0,          ac-
                                              cel_rad_per_sec2=20.0,  duration_sec=0.0,
                                              action_id=0)

    Bases: pycozmo.protocol_base.Packet

    accel_rad_per_sec2
    ack
    action_id
    angle_rad
    duration_sec
    classmethod from_bytes (buffer)
    classmethod from_reader (reader)
    id
    is_from_engine () → bool
    is_from_robot () → bool
    is_oob () → bool
    max_speed_rad_per_sec
    seq
    to_bytes ()
    to_writer (writer)
    type

```

```
class pycozmo.protocol_encoder.SetHeadLight (enable=False)
    Bases: pycozmo.protocol_base.Packet

    ack

    enable

    classmethod from_bytes (buffer)

    classmethod from_reader (reader)

    id

    is_from_engine () → bool

    is_from_robot () → bool

    is_oob () → bool

    seq

    to_bytes ()

    to_writer (writer)

    type

class pycozmo.protocol_encoder.SetLiftHeight (height_mm=0.0,
                                              max_speed_rad_per_sec=3.0,      ac-
                                              cel_rad_per_sec2=20.0, duration_sec=0.0,
                                              action_id=0)

    Bases: pycozmo.protocol_base.Packet

    accel_rad_per_sec2

    ack

    action_id

    duration_sec

    classmethod from_bytes (buffer)

    classmethod from_reader (reader)

    height_mm

    id

    is_from_engine () → bool

    is_from_robot () → bool

    is_oob () → bool

    max_speed_rad_per_sec

    seq

    to_bytes ()

    to_writer (writer)

    type

class pycozmo.protocol_encoder.SetOrigin (unknown0=0,                      pose_frame_id=0,
                                           pose_origin_id=1, pose_x=0.0, pose_y=0.0,
                                           unknown5=2147483648)

    Bases: pycozmo.protocol_base.Packet
```



```

    ack
    classmethod from_bytes (buffer)
    classmethod from_reader (reader)
    id
    is_from_engine () → bool
    is_from_robot () → bool
    is_oob () → bool
    pose_frame_id
    pose_origin_id
    pose_x
    pose_y
    seq
    to_bytes ()
    to_writer (writer)
    type
    unknown0
    unknown5

class pycozmo.protocol_encoder.SetRobotVolume (level=0)
    Bases: pycozmo.protocol_base.Packet
    ack
    classmethod from_bytes (buffer)
    classmethod from_reader (reader)
    id
    is_from_engine () → bool
    is_from_robot () → bool
    is_oob () → bool
    level
    seq
    to_bytes ()
    to_writer (writer)
    type

class pycozmo.protocol_encoder.ShutdownRobot
    Bases: pycozmo.protocol_base.Packet
    ack
    classmethod from_bytes (buffer)
    classmethod from_reader (reader)
    id

```

```
    is_from_engine() → bool
    is_from_robot() → bool
    is_oob() → bool
    seq
    to_bytes()
    to_writer(writer)
    type
class pycozmo.protocol_encoder.StartAnimation(anim_id=0)
    Bases: pycozmo.protocol\_base.Packet
    ack
    anim_id
    classmethod from_bytes(buffer)
    classmethod from_reader(reader)
    id
    is_from_engine() → bool
    is_from_robot() → bool
    is_oob() → bool
    seq
    to_bytes()
    to_writer(writer)
    type
class pycozmo.protocol_encoder.StartMotorCalibration(head=False, lift=False)
    Bases: pycozmo.protocol\_base.Packet
    ack
    classmethod from_bytes(buffer)
    classmethod from_reader(reader)
    head
    id
    is_from_engine() → bool
    is_from_robot() → bool
    is_oob() → bool
    lift
    seq
    to_bytes()
    to_writer(writer)
    type
```

---

```

class pycozmo.protocol_encoder.StopAllMotors
    Bases: pycozmo.protocol_base.Packet

    ack

    classmethod from_bytes (buffer)
    classmethod from_reader (reader)

    id

    is_from_engine () → bool
    is_from_robot () → bool
    is_oob () → bool

    seq

    to_bytes ()
    to_writer (writer)

    type

class pycozmo.protocol_encoder.StreamObjectAccel (object_id=0, enable=False)
    Bases: pycozmo.protocol_base.Packet

    ack

    enable

    classmethod from_bytes (buffer)
    classmethod from_reader (reader)

    id

    is_from_engine () → bool
    is_from_robot () → bool
    is_oob () → bool

    object_id

    seq

    to_bytes ()
    to_writer (writer)

    type

class pycozmo.protocol_encoder.SyncTime (timestamp=0, unknown=0)
    Bases: pycozmo.protocol_base.Packet

    ack

    classmethod from_bytes (buffer)
    classmethod from_reader (reader)

    id

    is_from_engine () → bool
    is_from_robot () → bool
    is_oob () → bool

```

```
seq
timestamp
to_bytes()
to_writer(writer)
type
unknown

class pycozmo.protocol_encoder.TrimPath(head=0, tail=0)
    Bases: pycozmo.protocol_base.Packet
    ack
    classmethod from_bytes(buffer)
    classmethod from_reader(reader)
    head
    id
    is_from_engine() → bool
    is_from_robot() → bool
    is_oob() → bool
    seq
    tail
    to_bytes()
    to_writer(writer)
    type

class pycozmo.protocol_encoder.TurnInPlace(angle_rad=0.0, speed_rad_per_sec=0.0,
                                           accel_rad_per_sec2=0.0, angle_tolerance_rad=0.0,
                                           unknown4=0, unknown5=0, is_absolute=False, action_id=0)
    Bases: pycozmo.protocol_base.Packet
    accel_rad_per_sec2
    ack
    action_id
    angle_rad
    angle_tolerance_rad
    classmethod from_bytes(buffer)
    classmethod from_reader(reader)
    id
    is_absolute
    is_from_engine() → bool
    is_from_robot() → bool
    is_oob() → bool
```

```

    seq
    speed_rad_per_sec
    to_bytes()
    to_writer(writer)
    type
    unknown4
    unknown5
class pycozmo.protocol_encoder.TurnInPlaceAtSpeed(wheel_speed_mmps=0.0,
                                                    wheel_accel_mmps2=0.0,    di-
                                                    rection=0)

    Bases: pycozmo.protocol_base.Packet
    ack
    direction
    classmethod from_bytes(buffer)
    classmethod from_reader(reader)
    id
    is_from_engine() → bool
    is_from_robot() → bool
    is_oob() → bool
    seq
    to_bytes()
    to_writer(writer)
    type
    wheel_accel_mmps2
    wheel_speed_mmps
class pycozmo.protocol_encoder.TurnToRecordedHeading
    Bases: pycozmo.protocol_base.Packet
    ack
    classmethod from_bytes(buffer)
    classmethod from_reader(reader)
    id
    is_from_engine() → bool
    is_from_robot() → bool
    is_oob() → bool
    seq
    to_bytes()
    to_writer(writer)
    type

```

```
class pycozmo.protocol_encoder.UpAxis
    Bases: enum.Enum

    An enumeration.

    NumAxes = 6
    UnknownAxis = 7
    XNegative = 0
    XPositive = 1
    YNegative = 2
    YPositive = 3
    ZNegative = 4
    ZPositive = 5

class pycozmo.protocol_encoder.WifiOff(enable=False)
    Bases: pycozmo.protocol_base.Packet

    ack
    enable

    classmethod from_bytes(buffer)
    classmethod from_reader(reader)

    id
    is_from_engine() → bool
    is_from_robot() → bool
    is_oob() → bool
    seq
    to_bytes()
    to_writer(writer)
    type
```

## 10.29 pycozmo.protocol\_generator

Cozmo protocol packet encoder code generator.

### Functions

---

<code>get_enum_fmt(argument)</code>
<code>get_farray_fmt(argument)</code>
<code>get_fmt_by_type(t)</code>
<code>get_string_fmt(argument)</code>
<code>get_varray_fmts(argument)</code>
<code>int_to_str(value, base)</code>

---

## Classes

---

*ProtocolGenerator(f)*

---

```
class pycozmo.protocol_generator.ProtocolGenerator(f)
    Bases: object

    generate() → None
    generate_argument_assignments(struct: pycozmo.protocol_ast.Struct) → None
    generate_argument_defaults(struct: pycozmo.protocol_ast.Struct) → None
    generate_argument_methods(struct: pycozmo.protocol_ast.Struct) → None
    generate_enum(enum_type: pycozmo.protocol_ast.Enum) → None
    generate_enum_validation(argument: pycozmo.protocol_ast.EnumArgument) → None
    generate_farray_validation(argument: pycozmo.protocol_ast.FArrayArgument) → None
    generate_group_map() → None
    generate_id_map() → None
    generate_len_method(struct: pycozmo.protocol_ast.Struct) → None
    generate_packet(packet: pycozmo.protocol_ast.Packet) → None
    generate_packet_argument_assignments(packet: pycozmo.protocol_ast.Packet) → None
    generate_packet_decoding(struct: pycozmo.protocol_ast.Struct) → None
    generate_packet_encoding(struct: pycozmo.protocol_ast.Struct) → None
    generate_packet_slots(struct: pycozmo.protocol_ast.Struct) → None
    generate_repr_method(struct: pycozmo.protocol_ast.Struct) → None
    generate_string_validation(argument: pycozmo.protocol_ast.StringArgument) → None
    generate_struct(struct: pycozmo.protocol_ast.Struct) → None
    generate_varray_validation(argument: pycozmo.protocol_ast.VArrayArgument) → None
```

## 10.30 pycozmo.protocol\_utils

Cozmo protocol encoding helper classes and functions.

### Functions

<i>get_farray_size</i> (fmt, length)	Figures out the size of a fixed array with given format.
<i>get_object_farray_size</i> (value, length)	Figures out the size of a given fixed-length object sequence.
<i>get_object_size</i> (value)	Figures out the size of a given object.
<i>get_size</i> (fmt)	Figures out the size of a value with the given format.
<i>get_string_size</i> (value, length_format)	Figures out the size of a string with given length format.

Continued on next page

Table 39 – continued from previous page

<code>get_varray_size(value, length_format, ...)</code>	Figures out the size of a variable-length array with given format.
<code>validate_bool(name, value)</code>	
<code>validate_farray(name, value, length, ...)</code>	
<code>validate_float(name, value)</code>	
<code>validate_integer(name, value, minimum, maximum)</code>	
<code>validate_object(name, value, expected_type)</code>	
<code>validate_string(name, value, maximum_length)</code>	
<code>validate_varray(name, value, maximum_length, ...)</code>	

## Classes

<code>BinaryReader(buffer, offset)</code>	Used to read in a stream of binary data, keeping track of the current position.
<code>BinaryWriter()</code>	Used to write out a stream of binary data.

`pycozmo.protocol_utils.validate_float(name, value)`

`pycozmo.protocol_utils.validate_bool(name, value)`

`pycozmo.protocol_utils.validate_integer(name, value, minimum, maximum)`

`pycozmo.protocol_utils.validate_object(name, value, expected_type)`

`pycozmo.protocol_utils.validate_farray(name, value, length, element_validation)`

`pycozmo.protocol_utils.validate_varray(name, value, maximum_length, element_validation)`

`pycozmo.protocol_utils.validate_string(name, value, maximum_length)`

`pycozmo.protocol_utils.get_size(fmt)`

Figures out the size of a value with the given format.

`pycozmo.protocol_utils.get_farray_size(fmt, length)`

Figures out the size of a fixed array with given format.

`pycozmo.protocol_utils.get_varray_size(value, length_format, data_format)`

Figures out the size of a variable-length array with given format.

`pycozmo.protocol_utils.get_string_size(value, length_format)`

Figures out the size of a string with given length format.

`pycozmo.protocol_utils.get_object_size(value)`

Figures out the size of a given object.

`pycozmo.protocol_utils.get_object_farray_size(value, length)`

Figures out the size of a given fixed-length object sequence.

**class** `pycozmo.protocol_utils.BinaryReader(buffer: bytes, offset: int = 0)`

Bases: `object`

Used to read in a stream of binary data, keeping track of the current position.

**buffer**

**read** (*fmt*)

Reads in a single value of the given format.



**read\_farray** (*fmt, length*)  
 Reads in a fixed-length array of the given format and length.

**read\_object** (*from\_\_reader\_method*)  
 Reads in an object according to the given method.

**read\_object\_farray** (*from\_\_reader\_method, length*)  
 Reads in a fixed-length object sequence according to the given method.

**read\_object\_varray** (*from\_\_reader\_method, length\_format*)  
 Reads in a variable-length object sequence according to the given method.

**read\_string** (*length\_format*)  
 Reads in a variable-length string with the given length format.

**read\_string\_farray** (*string\_length\_format, array\_length*)  
 Reads in a fixed-length array of variable-length strings with the given length format.

**read\_string\_varray** (*string\_length\_format, array\_length\_format*)  
 Reads in a variable-length array of variable-length strings with the given length format.

**read\_varray** (*data\_format, length\_format*)  
 Reads in a variable-length array with the given length format and data format.

**seek\_cur** (*offset: int*) → None

**seek\_set** (*offset: int*) → None

**tell** ()  
 Returns the current stream position as an offset within the buffer.

**class** pycozmo.protocol\_utils.BinaryWriter

Bases: `object`

Used to write out a stream of binary data.

**clear** ()

**dumps** () → bytes

**write** (*value, fmt*)  
 Writes out a single value of the given format.

**write\_bytes** (*value: bytes*) → None  
 Writes out a byte sequence.

**write\_farray** (*value, fmt, length*)  
 Writes out a fixed-length array of the given format and length.

**write\_object** (*value*)  
 Writes out an object that supports a `to_writer()` method.

**write\_object\_farray** (*value, length*)  
 Writes out a fixed-length object sequence that supports a `to_writer()` method.

**write\_object\_varray** (*value, length\_format*)  
 Writes out a variable-length object sequence that supports a `to_writer()` method.

**write\_string** (*value, length\_format*)  
 Writes out a variable-length string with the given length format.

**write\_string\_farray** (*value, string\_length\_format, array\_length*)  
 Writes out a fixed-length array of variable-length strings with the given length format.

**write\_string\_varray** (*value, string\_length\_format, array\_length\_format*)

Writes out a variable-length array of variable-length strings with the given length format.

**write\_varray** (*value, data\_format, length\_format*)

Writes out a variable-length array with the given length format and data format.

## 10.31 pycozmo.robot

Robot constants and helper code.

### Classes

---

<i>LiftPosition</i> ([height, ratio, angle])	Represents the position of Cozmo's lift.
RobotOrientation	Robot orientation enumeration.
<i>RobotStatusFlag</i>	

---

`pycozmo.robot.MIN_HEAD_ANGLE = <Angle -0.44 radians (-25.00 degrees)>`

Minimum head angle.

`pycozmo.robot.MAX_HEAD_ANGLE = <Angle 0.78 radians (44.50 degrees)>`

Maximum head angle.

`pycozmo.robot.MIN_LIFT_HEIGHT = <Distance 32.00 mm (1.26 inches)>`

Minimum lift height.

`pycozmo.robot.MAX_LIFT_HEIGHT = <Distance 92.00 mm (3.62 inches)>`

Maximum lift height.

`pycozmo.robot.LIFT_ARM_LENGTH = <Distance 66.00 mm (2.60 inches)>`

Lift arm length.

`pycozmo.robot.LIFT_PIVOT_HEIGHT = <Distance 45.00 mm (1.77 inches)>`

Lift arm pivot point height.

`pycozmo.robot.MIN_LIFT_ANGLE = <Angle -0.20 radians (-11.36 degrees)>`

Minimum lift arm angle.

`pycozmo.robot.MAX_LIFT_ANGLE = <Angle 0.79 radians (45.41 degrees)>`

Maximum lift arm angle.

`pycozmo.robot.MAX_WHEEL_SPEED = <Speed 200.00 mm/s>`

Maximum wheel speed.

`pycozmo.robot.TRACK_WIDTH = <Distance 45.00 mm (1.77 inches)>`

Track width.

`pycozmo.robot.FRAME_RATE = 30`

Number of frames per second for animations.

**class** `pycozmo.robot.RobotStatusFlag`

Bases: `object`

`ARE_WHEELS_MOVING = 32768`

`CLIFF_DETECTED = 16384`

`HEAD_IN_POS = 512`

```

IS_ANIMATING = 64
IS_ANIMATING_IDLE = 2048
IS_ANIM_BUFFER_FULL = 1024
IS_BODY_ACC_MODE = 16
IS_CARRYING_BLOCK = 2
IS_CHARGER_OOS = 65536
IS_CHARGING = 8192
IS_FALLING = 32
IS_MOVING = 1
IS_ON_CHARGER = 4096
IS_PATHING = 128
IS_PICKED_UP = 8
IS_PICKING_OR_PLACING = 4
LIFT_IN_POS = 256

```

```
pycozmo.robot.RobotStatusFlagNames = {1: 'IS_MOVING', 2: 'IS_CARRYING_BLOCK', 4: 'IS_PICKED_UP', 8: 'IS_PICKING_OR_PLACING', 16: 'IS_BODY_ACC_MODE', 32: 'IS_FALLING', 64: 'IS_ANIMATING', 128: 'IS_PATHING', 256: 'IS_LIFT_IN_POS', 512: 'IS_CHARGING', 1024: 'IS_ANIM_BUFFER_FULL', 2048: 'IS_ANIMATING_IDLE', 4096: 'IS_ON_CHARGER', 65536: 'IS_CHARGER_OOS'}
```

Robot status flag names.

```
class pycozmo.robot.LiftPosition (height=None, ratio=None, angle=None)
    Bases: object
```

Represents the position of Cozmo's lift.

The class allows the position to be referred to as either absolute height above the ground, as a ratio from 0.0 to 1.0, or as the angle of the lift arm relative to the ground.

**Args:** height (cozmo.util.Distance): The height of the lift above the ground. ratio (float): The ratio from 0.0 to 1.0 that the lift is raised from the ground. angle (cozmo.util.Angle): The angle of the lift arm relative to the ground.

**angle**  
The angle of the lift arm relative to the ground.

**height**  
Height above the ground.

**ratio**  
The ratio from 0 to 1 that the lift is raised, 0 at the bottom, 1 at the top.

## 10.32 pycozmo.robot\_debug

Cozmo firmware debug message decoding.

Based on AnkiLogStringTables.json .

### Functions

<code>get_debug_message(name_id, format_id, args)</code>	Generate a log message from robot debug name and format IDs.
<code>get_log_level(robot_level)</code>	Translate robot log level to Python log level.

`pycozmo.robot_debug.get_debug_message` (*name\_id: int, format\_id: int, args: List[Any]*) → *Optional[str]*  
 Generate a log message from robot debug name and format IDs.

`pycozmo.robot_debug.get_log_level` (*robot\_level: int*) → *int*  
 Translate robot log level to Python log level.

## 10.33 pycozmo.run

Helper functions for running PyCozmo applications.

### Functions

<code>connect(log_level, protocol_log_level, ...)</code>
<code>setup_basic_logging(log_level, ..., target)</code>

`pycozmo.run.setup_basic_logging` (*log\_level: Optional[str] = None, protocol\_log\_level: Optional[str] = None, robot\_log\_level: Optional[str] = None, target=<\_io.TextIOWrapper name='<stderr>' mode='w' encoding='UTF-8'>*) → *None*

`pycozmo.run.connect` (*log\_level: Optional[str] = None, protocol\_log\_level: Optional[str] = None, protocol\_log\_messages: Optional[list] = None, robot\_log\_level: Optional[str] = None, auto\_initialize: bool = True, enable\_animations: bool = True, enable\_procedural\_face: bool = True*) → *pycozmo.client.Client*

## 10.34 pycozmo.util

Utility classes and functions.

### Functions

<code>angle_z_to_quaternion(angle_z)</code>	Converts an angle in the z axis (Euler angle z component) to a quaternion.
<code>check_assets()</code>	Check whether Cozmo assets are available.
<code>frange(start, stop, step)</code>	
<code>get_cozmo_anim_dir()</code>	Get Cozmo animation asset directory.
<code>get_cozmo_asset_dir()</code>	Get Cozmo asset directory.
<code>get_pycozmo_dir()</code>	Get PyCozmo directory.
<code>hex_dump(data)</code>	
<code>hex_load(data)</code>	

## Classes

<i>Angle</i> (radians, degrees)	Angle representation.
<i>Distance</i> (mm, inches)	Represents a distance.
<i>FPSTimer</i> (fps)	A timer that maintains frame rate by sleeping for a variable amount of time.
<i>Matrix44</i> (m00, m10, m20, m30, m01, m11, m21, ...)	A 4x4 Matrix for representing the rotation and/or position of an object in the world.
<i>Pose</i> (x, y, z, q0, q1, q2, q3, angle_z, ...)	A combination of position (vector) and rotation (quaternion).
<i>Quaternion</i> (q0, q1, q2, q3, angle_z)	Represents rotation.
<i>Speed</i> (mmps)	Speed representation.
<i>Vector2</i> (x, y)	Represents a 2D Vector (type/units aren't specified)
<i>Vector3</i> (x, y, z)	Represents a 3D Vector (type/units aren't specified).

**class** pycozmo.util.**Angle** (*radians: Optional[float] = None, degrees: Optional[float] = None*)

Bases: `object`

Angle representation.

### Args:

**radians (float):** The number of radians the angle should represent (cannot be combined with degrees)

**degrees (float):** The number of degrees the angle should represent (cannot be combined with radians)

### abs\_value

cozmo.util.Angle: The absolute value of the angle.

If the Angle is positive then it returns a copy of this Angle, otherwise it returns -Angle.

### degrees

Returns the angle in degrees.

### radians

Returns the angle in radians.

**class** pycozmo.util.**Distance** (*mm: Optional[float] = None, inches: Optional[float] = None*)

Bases: `object`

Represents a distance.

The class allows distances to be returned in either millimeters or inches.

### Args:

**mm (float):** The number of millimeters the distance should represent (cannot be combined with distance\_inches).

**inches (float):** The number of inches the distance should represent (cannot be combined with distance\_mm).

### inches

The distance in inches.

### mm

The distance in millimeters.

**class** `pycozmo.util.Speed` (*mmps: float*)

Bases: `object`

Speed representation.

**Args:** `mmps` (float): The number of millimeters per second the speed should represent.

**mmps**

Returns the speed in millimeters per second (mmps).

**class** `pycozmo.util.Vector2` (*x: float, y: float*)

Bases: `object`

Represents a 2D Vector (type/units aren't specified)

**Args:** `x` (float): X component `y` (float): Y component

**set\_to** (*rhs*) → None

Copy the x and y components of the given vector.

**Args:**

**rhs** (`Vector2`): The right-hand-side of this assignment - the source vector to copy into this vector.

**x**

The x component.

**x\_y**

The X, Y elements of the Vector2 (x,y).

**y**

The y component.

**class** `pycozmo.util.Vector3` (*x: float, y: float, z: float*)

Bases: `object`

Represents a 3D Vector (type/units aren't specified).

**Args:** `x` (float): X component `y` (float): Y component `z` (float): Z component

**set\_to** (*rhs*)

Copy the x, y and z components of the given vector.

**Args:**

**rhs** (`Vector3`): The right-hand-side of this assignment - the source vector to copy into this vector.

**x**

The x component.

**x\_y\_z**

The X, Y, Z elements of the Vector3 (x,y,z).

**y**

The y component.

**z**

The z component.

`pycozmo.util.angle_z_to_quaternion` (*angle\_z: pycozmo.util.Angle*) → Tuple[float, float, float, float]

Converts an angle in the z axis (Euler angle z component) to a quaternion.

---

```
class pycozmo.util.Matrix44 (m00, m10, m20, m30, m01, m11, m21, m31, m02, m12, m22, m32,  
                             m03, m13, m23, m33)
```

Bases: `object`

A 4x4 Matrix for representing the rotation and/or position of an object in the world.

Can be generated from a Quaternion for a pure rotation matrix, or combined with a position for a full translation matrix, as done by `Pose.to_matrix()`.

**forward\_xyz**

Returns the x,y,z components representing the matrix's forward vector.

**in\_column\_order**

Returns the contents of the matrix in column order.

**in\_row\_order**

Returns the contents of the matrix in row order.

**left\_xyz**

Returns the x,y,z components representing the matrix's left vector.

**m00**

**m01**

**m02**

**m03**

**m10**

**m11**

**m12**

**m13**

**m20**

**m21**

**m22**

**m23**

**m30**

**m31**

**m32**

**m33**

**pos\_xyz**

Returns the x,y,z components representing the matrix's position vector.

**set\_forward** (*x: float, y: float, z: float*) → None

Set the x,y,z components representing the matrix's forward vector.

**set\_left** (*x: float, y: float, z: float*) → None

Set the x,y,z components representing the matrix's left vector.

**set\_pos** (*x: float, y: float, z: float*) → None

Set the x,y,z components representing the matrix's position vector.

**set\_up** (*x: float, y: float, z: float*) → None

Set the x,y,z components representing the matrix's up vector.

**tabulated\_string**

str: A multi-line string formatted with tabs to show the matrix contents.

**up\_xyz**

Returns the x,y,z components representing the matrix's up vector.

```
class pycozmo.util.Quaternion(q0: Optional[float] = None, q1: Optional[float] = None, q2: Optional[float] = None, q3: Optional[float] = None, angle_z: Optional[pycozmo.util.Angle] = None)
```

Bases: `object`

Represents rotation.

**angle\_z****euler\_angles**

Returns the pitch, yaw, roll Euler components of the object's rotation defined as rotations in the x, y, and z axis respectively.

**Returns**

**q0**

**q0\_q1\_q2\_q3**

**q1**

**q2**

**q3**

**to\_matrix** (*pos\_x: float = 0.0, pos\_y: float = 0.0, pos\_z: float = 0.0*) → `pycozmo.util.Matrix44`

Convert the Quaternion to a 4x4 matrix representing this rotation.

A position can also be provided to generate a full translation matrix.

```
class pycozmo.util.Pose(x: float, y: float, z: float, q0: Optional[float] = None, q1: Optional[float] = None, q2: Optional[float] = None, q3: Optional[float] = None, angle_z: Optional[pycozmo.util.Angle] = None, origin_id: int = -1, is_accurate: bool = True)
```

Bases: `object`

A combination of position (vector) and rotation (quaternion).

**define\_pose\_relative\_this** (*new\_pose*)

Creates a new pose such that new\_pose's origin is now at the location of this pose.

**invalidate** () → `None`

Mark this pose as being invalid (unusable).

**is\_accurate**

Returns True if this pose is valid and accurate.

Poses are marked as inaccurate if we detect movement via accelerometer, or if they were observed from far enough away that we're less certain of the exact pose.

**is\_comparable** (*other\_pose: pycozmo.util.Pose*) → `bool`

Are these two poses comparable.

Poses are comparable if they're valid and having matching origin IDs.

**is\_valid**

Checks whether a pose is valid (usable).

**origin\_id**

Returns an ID maintained by the robot (engine) which represents which coordinate frame this pose is in.



**position**

Returns the position component of this pose.

**rotation**

Returns the rotation component of this pose.

**to\_matrix()**

Convert the Pose to a Matrix44.

**class** pycozmo.util.FPSTimer(*fps: int*)Bases: `object`

A timer that maintains frame rate by sleeping for a variable amount of time.

**sleep()**

Sleep to maintain the framerate. Should be called at the end of a frame.

pycozmo.util.hex\_dump(*data: bytes*) → strpycozmo.util.hex\_load(*data: str*) → bytespycozmo.util.frange(*start, stop, step*)

pycozmo.util.get\_pycozmo\_dir() → pathlib.Path

Get PyCozmo directory.

pycozmo.util.get\_cozmo\_asset\_dir() → pathlib.Path

Get Cozmo asset directory.

pycozmo.util.check\_assets() → None

Check whether Cozmo assets are available.

pycozmo.util.get\_cozmo\_anim\_dir() → pathlib.Path

Get Cozmo animation asset directory.

## 10.35 pycozmo.window

Cozmo protocol sliding window implementation.

### Classes

<i>BaseWindow</i> (seq_bits, size, max_seq)	Base communication window class.
<i>ReceiveWindow</i> (seq_bits, size, max_seq)	Receive communication window class.
<i>SendWindow</i> (seq_bits, size, max_seq)	Send communication window class.

**class** pycozmo.window.BaseWindow(*seq\_bits: int, size: Optional[int] = None, max\_seq: Optional[int] = None*)Bases: `object`

Base communication window class.

**is\_valid\_seq**(*seq: int*) → bool

Check whether a sequence number is valid for the window.

**reset**() → None

Reset the window.

**class** pycozmo.window.ReceiveWindow(*seq\_bits: int, size: Optional[int] = None, max\_seq: Optional[int] = None*)

Bases: `pycozmo.window.BaseWindow`

Receive communication window class.

When packets are received (in whatever order), they are put in the window using the `put()` method.

Packets are extracted from the window in the expected order using the `get()` method.

**exists** (*seq: int*) → bool

Check whether a sequence number has already been received (assuming it is valid).

**get** () → Any

If data is available, return it and move the window forward. Return None otherwise.

**is\_out\_of\_order** (*seq: int*) → bool

Check whether a sequence number is outside the current window (assuming it is valid).

**is\_valid\_seq** (*seq: int*) → bool

Check whether a sequence number is valid for the window.

**put** (*seq: int, data: Any*) → None

Add the data, associated with a particular sequence number to the window.

**reset** () → None

Reset the window.

**class** `pycozmo.window.SendWindow` (*seq\_bits: int, size: Optional[int] = None, max\_seq: Optional[int] = None*)

Bases: `pycozmo.window.BaseWindow`

Send communication window class.

When packets are sent, they are put in the window using the `put()` method which returns a sequence number.

Packets are removed from the window when they are acknowledged with the `acknowledge()` method.

**acknowledge** (*seq: int*) → None

Acknowledge a sequence number and remove any associated data from the window.

**get** () → List[Tuple[int, Any]]

Get the contents of the window as a list of tuples (sequence number, data).

**is\_full** () → bool

Check whether the window is full.

**is\_out\_of\_order** (*seq: int*) → bool

Check whether a sequence number is outside the current window (assuming it is valid).

**is\_valid\_seq** (*seq: int*) → bool

Check whether a sequence number is valid for the window.

**put** (*data: Any*) → int

Add data to the window. Raises `NoSpace` exception if the window is full.

**reset** ()

Reset the window.

# CHAPTER 11

---

## Indices and tables

---

- `genindex`
- `modindex`
- `search`



### p

- `pycozmo.activity`, 58
- `pycozmo.anim`, 59
- `pycozmo.anim_controller`, 60
- `pycozmo.anim_encoder`, 61
- `pycozmo.audio`, 66
- `pycozmo.audiokinetic.exception`, 40
- `pycozmo.audiokinetic.soundbank`, 41
- `pycozmo.audiokinetic.soundbanksinfo`, 42
- `pycozmo.audiokinetic.wem`, 44
- `pycozmo.behavior`, 66
- `pycozmo.brain`, 67
- `pycozmo.camera`, 68
- `pycozmo.client`, 69
- `pycozmo.conn`, 71
- `pycozmo.emotions`, 76
- `pycozmo.event`, 77
- `pycozmo.exception`, 79
- `pycozmo.expressions.expressions`, 44
- `pycozmo.filter`, 80
- `pycozmo.frame`, 81
- `pycozmo.image_encoder`, 81
- `pycozmo.lights`, 82
- `pycozmo.object`, 83
- `pycozmo.procedural_face`, 83
- `pycozmo.protocol_ast`, 86
- `pycozmo.protocol_base`, 91
- `pycozmo.protocol_declaration`, 93
- `pycozmo.protocol_encoder`, 93
- `pycozmo.protocol_generator`, 136
- `pycozmo.protocol_utils`, 137
- `pycozmo.robot`, 140
- `pycozmo.robot_debug`, 141
- `pycozmo.run`, 142
- `pycozmo.util`, 142
- `pycozmo.window`, 147



## A

- AbortAnimation (class in py-cozmo.protocol\_encoder), 95
- abs\_value (pycozmo.util.Angle attribute), 143
- accel\_mmmps2 (pycozmo.protocol\_encoder.AppendPathSegArc attribute), 99
- accel\_mmmps2 (pycozmo.protocol\_encoder.AppendPathSegLine attribute), 99
- accel\_mmmps2 (pycozmo.protocol\_encoder.AppendPathSegPointTurn attribute), 100
- accel\_mmmps2 (pycozmo.protocol\_encoder.PathSegmentSpeed attribute), 125
- accel\_rad\_per\_sec2 (py-cozmo.protocol\_encoder.SetHeadAngle attribute), 129
- accel\_rad\_per\_sec2 (py-cozmo.protocol\_encoder.SetLiftHeight attribute), 130
- accel\_rad\_per\_sec2 (py-cozmo.protocol\_encoder.TurnInPlace attribute), 134
- accel\_x (pycozmo.protocol\_encoder.ObjectAccel attribute), 118
- accel\_x (pycozmo.protocol\_encoder.RobotState attribute), 127
- accel\_y (pycozmo.protocol\_encoder.ObjectAccel attribute), 118
- accel\_y (pycozmo.protocol\_encoder.RobotState attribute), 127
- accel\_z (pycozmo.protocol\_encoder.ObjectAccel attribute), 118
- accel\_z (pycozmo.protocol\_encoder.RobotState attribute), 127
- ack (pycozmo.frame.Frame attribute), 81
- ack (pycozmo.protocol\_base.Packet attribute), 91
- ack (pycozmo.protocol\_base.UnknownCommand attribute), 92
- ack (pycozmo.protocol\_base.UnknownEvent attribute), 92
- ack (pycozmo.protocol\_base.UnknownPacket attribute), 91
- ack (pycozmo.protocol\_encoder.AbortAnimation attribute), 95
- ack (pycozmo.protocol\_encoder.AcknowledgeAction attribute), 95
- ack (pycozmo.protocol\_encoder.AnimationEnded attribute), 97
- ack (pycozmo.protocol\_encoder.AnimationStarted attribute), 98
- ack (pycozmo.protocol\_encoder.AnimationState attribute), 98
- ack (pycozmo.protocol\_encoder.AnimBackpackLights attribute), 96
- ack (pycozmo.protocol\_encoder.AnimBody attribute), 96
- ack (pycozmo.protocol\_encoder.AnimHead attribute), 96
- ack (pycozmo.protocol\_encoder.AnimLift attribute), 97
- ack (pycozmo.protocol\_encoder.AppendPathSegArc attribute), 99
- ack (pycozmo.protocol\_encoder.AppendPathSegLine attribute), 99
- ack (pycozmo.protocol\_encoder.AppendPathSegPointTurn attribute), 100
- ack (pycozmo.protocol\_encoder.BodyInfo attribute), 101
- ack (pycozmo.protocol\_encoder.ButtonPressed attribute), 101
- ack (pycozmo.protocol\_encoder.ClearPath attribute), 102
- ack (pycozmo.protocol\_encoder.Connect attribute), 102
- ack (pycozmo.protocol\_encoder.CubeId attribute), 102
- ack (pycozmo.protocol\_encoder.CubeLights attribute), 103
- ack (pycozmo.protocol\_encoder.DebugData attribute), 103
- ack (pycozmo.protocol\_encoder.Disconnect attribute), 104
- ack (pycozmo.protocol\_encoder.DisplayImage attribute), 104
- ack (pycozmo.protocol\_encoder.DriveWheels attribute), 104

`ack` (`pycozmo.protocol_encoder.Enable` attribute), 105

`ack` (`pycozmo.protocol_encoder.EnableAnimationState` attribute), 105

`ack` (`pycozmo.protocol_encoder.EnableCamera` attribute), 106

`ack` (`pycozmo.protocol_encoder.EnableColorImages` attribute), 106

`ack` (`pycozmo.protocol_encoder.EnableStopOnCliff` attribute), 106

`ack` (`pycozmo.protocol_encoder.EndAnimation` attribute), 107

`ack` (`pycozmo.protocol_encoder.ExecutePath` attribute), 107

`ack` (`pycozmo.protocol_encoder.FallingStarted` attribute), 108

`ack` (`pycozmo.protocol_encoder.FallingStopped` attribute), 108

`ack` (`pycozmo.protocol_encoder.FirmwareSignature` attribute), 108

`ack` (`pycozmo.protocol_encoder.FirmwareUpdate` attribute), 109

`ack` (`pycozmo.protocol_encoder.FirmwareUpdateResult` attribute), 109

`ack` (`pycozmo.protocol_encoder.HardwareInfo` attribute), 110

`ack` (`pycozmo.protocol_encoder.ImageChunk` attribute), 110

`ack` (`pycozmo.protocol_encoder.ImageImuData` attribute), 111

`ack` (`pycozmo.protocol_encoder.Keyframe` attribute), 112

`ack` (`pycozmo.protocol_encoder.LightStateCenter` attribute), 113

`ack` (`pycozmo.protocol_encoder.LightStateSide` attribute), 113

`ack` (`pycozmo.protocol_encoder.MotorCalibration` attribute), 114

`ack` (`pycozmo.protocol_encoder.MoveHead` attribute), 114

`ack` (`pycozmo.protocol_encoder.MoveLift` attribute), 115

`ack` (`pycozmo.protocol_encoder.NvStorageOp` attribute), 117

`ack` (`pycozmo.protocol_encoder.NvStorageOpResult` attribute), 118

`ack` (`pycozmo.protocol_encoder.ObjectAccel` attribute), 118

`ack` (`pycozmo.protocol_encoder.ObjectAvailable` attribute), 119

`ack` (`pycozmo.protocol_encoder.ObjectConnect` attribute), 119

`ack` (`pycozmo.protocol_encoder.ObjectConnectionState` attribute), 119

`ack` (`pycozmo.protocol_encoder.ObjectMoved` attribute), 120

`ack` (`pycozmo.protocol_encoder.ObjectPowerLevel` attribute), 120

`ack` (`pycozmo.protocol_encoder.ObjectStoppedMoving` attribute), 121

`ack` (`pycozmo.protocol_encoder.ObjectTapFiltered` attribute), 121

`ack` (`pycozmo.protocol_encoder.ObjectTapped` attribute), 122

`ack` (`pycozmo.protocol_encoder.ObjectUpAxisChanged` attribute), 123

`ack` (`pycozmo.protocol_encoder.OutputAudio` attribute), 124

`ack` (`pycozmo.protocol_encoder.OutputSilence` attribute), 124

`ack` (`pycozmo.protocol_encoder.PathFollowingEvent` attribute), 125

`ack` (`pycozmo.protocol_encoder.Ping` attribute), 125

`ack` (`pycozmo.protocol_encoder.RecordHeading` attribute), 126

`ack` (`pycozmo.protocol_encoder.RobotDelocalized` attribute), 126

`ack` (`pycozmo.protocol_encoder.RobotPoked` attribute), 127

`ack` (`pycozmo.protocol_encoder.RobotState` attribute), 127

`ack` (`pycozmo.protocol_encoder.SetAccessoryDiscovery` attribute), 128

`ack` (`pycozmo.protocol_encoder.SetCameraParams` attribute), 129

`ack` (`pycozmo.protocol_encoder.SetHeadAngle` attribute), 129

`ack` (`pycozmo.protocol_encoder.SetHeadLight` attribute), 130

`ack` (`pycozmo.protocol_encoder.SetLiftHeight` attribute), 130

`ack` (`pycozmo.protocol_encoder.SetOrigin` attribute), 130

`ack` (`pycozmo.protocol_encoder.SetRobotVolume` attribute), 131

`ack` (`pycozmo.protocol_encoder.ShutdownRobot` attribute), 131

`ack` (`pycozmo.protocol_encoder.StartAnimation` attribute), 132

`ack` (`pycozmo.protocol_encoder.StartMotorCalibration` attribute), 132

`ack` (`pycozmo.protocol_encoder.StopAllMotors` attribute), 133

`ack` (`pycozmo.protocol_encoder.StreamObjectAccel` attribute), 133

`ack` (`pycozmo.protocol_encoder.SyncTime` attribute), 133

`ack` (`pycozmo.protocol_encoder.TrimPath` attribute), 134

`ack` (`pycozmo.protocol_encoder.TurnInPlace` attribute), 134

`ack` (`pycozmo.protocol_encoder.TurnInPlaceAtSpeed` attribute), 134



- tribute*), 135
- `ack` (*pycozmo.protocol\_encoder.TurnToRecordedHeading attribute*), 135
- `ack` (*pycozmo.protocol\_encoder.WifiOff attribute*), 136
- `ack` () (*pycozmo.conn.SendThread method*), 72
- `ACK_TIMEOUT` (*pycozmo.conn.SendThread attribute*), 72
- `acknowledge` () (*pycozmo.window.SendWindow method*), 148
- `AcknowledgeAction` (class in *pycozmo.protocol\_encoder*), 95
- `action_id` (*pycozmo.protocol\_encoder.AcknowledgeAction attribute*), 95
- `action_id` (*pycozmo.protocol\_encoder.SetHeadAngle attribute*), 129
- `action_id` (*pycozmo.protocol\_encoder.SetLiftHeight attribute*), 130
- `action_id` (*pycozmo.protocol\_encoder.TurnInPlace attribute*), 134
- `action_ids` (*pycozmo.audiokinetic.soundbank.Event attribute*), 41
- `activate` () (*pycozmo.behavior.Behavior method*), 67
- `activate_behavior` () (*pycozmo.brain.Brain method*), 68
- `activate_behavior` () (*pycozmo.client.Client method*), 69
- `active_accel_x` (*pycozmo.protocol\_encoder.ObjectMoved attribute*), 120
- `active_accel_y` (*pycozmo.protocol\_encoder.ObjectMoved attribute*), 120
- `active_accel_z` (*pycozmo.protocol\_encoder.ObjectMoved attribute*), 120
- `Activity` (class in *pycozmo.activity*), 59
- `add` () (*pycozmo.emotions.EmotionType method*), 76
- `add_child_dispatcher` () (*pycozmo.behavior.Behavior method*), 67
- `add_child_dispatcher` () (*pycozmo.client.Client method*), 69
- `add_child_dispatcher` () (*pycozmo.conn.Connection method*), 74
- `add_child_dispatcher` () (*pycozmo.event.Dispatcher method*), 79
- `add_handler` () (*pycozmo.behavior.Behavior method*), 67
- `add_handler` () (*pycozmo.client.Client method*), 69
- `add_handler` () (*pycozmo.conn.Connection method*), 74
- `add_handler` () (*pycozmo.event.Dispatcher method*), 79
- `affectors` (*pycozmo.emotions.EmotionEvent attribute*), 76
- `allow_ids` () (*pycozmo.filter.Filter method*), 81
- `Amazement` (class in *pycozmo.expressions.expressions*), 57
- `Anger` (class in *pycozmo.expressions.expressions*), 45
- `Angle` (class in *pycozmo.util*), 143
- `angle` (*pycozmo.expressions.expressions.Amazement attribute*), 57
- `angle` (*pycozmo.expressions.expressions.Anger attribute*), 45
- `angle` (*pycozmo.expressions.expressions.Annoyance attribute*), 53
- `angle` (*pycozmo.expressions.expressions.Asleep attribute*), 56
- `angle` (*pycozmo.expressions.expressions.Boredom attribute*), 55
- `angle` (*pycozmo.expressions.expressions.Confusion attribute*), 56
- `angle` (*pycozmo.expressions.expressions.Despair attribute*), 49
- `angle` (*pycozmo.expressions.expressions.Disappointment attribute*), 50
- `angle` (*pycozmo.expressions.expressions.Disgust attribute*), 47
- `angle` (*pycozmo.expressions.expressions.Embarrassment attribute*), 51
- `angle` (*pycozmo.expressions.expressions.Excitement attribute*), 57
- `angle` (*pycozmo.expressions.expressions.Fear attribute*), 48
- `angle` (*pycozmo.expressions.expressions.Fury attribute*), 53
- `angle` (*pycozmo.expressions.expressions.Guilt attribute*), 50
- `angle` (*pycozmo.expressions.expressions.Happiness attribute*), 46
- `angle` (*pycozmo.expressions.expressions.Horror attribute*), 51
- `angle` (*pycozmo.expressions.expressions.Neutral attribute*), 44
- `angle` (*pycozmo.expressions.expressions.Pleading attribute*), 48
- `angle` (*pycozmo.expressions.expressions.Rejection attribute*), 54
- `angle` (*pycozmo.expressions.expressions.Sadness attribute*), 45
- `angle` (*pycozmo.expressions.expressions.Skepticism attribute*), 52
- `angle` (*pycozmo.expressions.expressions.Surprise attribute*), 46
- `angle` (*pycozmo.expressions.expressions.Suspicion attribute*), 54
- `angle` (*pycozmo.expressions.expressions.Tiredness attribute*), 55
- `angle` (*pycozmo.expressions.expressions.Vulnerability*

attribute), 49  
 angle (pycozmo.procedural\_face.ProceduralEye attribute), 84  
 angle (pycozmo.procedural\_face.ProceduralFace attribute), 85  
 angle (pycozmo.procedural\_face.ProceduralLid attribute), 84  
 angle (pycozmo.robot.LiftPosition attribute), 141  
 angle\_deg (pycozmo.protocol\_encoder.AnimHead attribute), 96  
 angle\_offset (pycozmo.procedural\_face.ProceduralLid attribute), 84  
 angle\_rad (pycozmo.protocol\_encoder.AppendPathSegPointTurn attribute), 100  
 angle\_rad (pycozmo.protocol\_encoder.SetHeadAngle attribute), 129  
 angle\_rad (pycozmo.protocol\_encoder.TurnInPlace attribute), 134  
 angle\_tolerance\_rad (pycozmo.protocol\_encoder.AppendPathSegPointTurn attribute), 100  
 angle\_tolerance\_rad (pycozmo.protocol\_encoder.TurnInPlace attribute), 134  
 angle\_z (pycozmo.util.Quaternion attribute), 146  
 angle\_z\_to\_quaternion() (in module pycozmo.util), 144  
 anim\_id (pycozmo.protocol\_encoder.AnimationEnded attribute), 97  
 anim\_id (pycozmo.protocol\_encoder.AnimationStarted attribute), 98  
 anim\_id (pycozmo.protocol\_encoder.StartAnimation attribute), 132  
 anim\_names (pycozmo.client.Client attribute), 69  
 AnimationController (class in pycozmo.anim\_controller), 60  
 AnimationEnded (class in pycozmo.protocol\_encoder), 97  
 AnimationGroup (class in pycozmo.anim), 60  
 AnimationGroupMember (class in pycozmo.anim), 59  
 AnimationQueue (class in pycozmo.anim\_controller), 61  
 AnimationStarted (class in pycozmo.protocol\_encoder), 98  
 AnimationState (class in pycozmo.protocol\_encoder), 98  
 AnimBackpackLights (class in pycozmo.anim\_encoder), 64  
 AnimBackpackLights (class in pycozmo.protocol\_encoder), 96  
 AnimBase (class in pycozmo.anim\_encoder), 62  
 AnimBody (class in pycozmo.protocol\_encoder), 96  
 AnimBodyMotion (class in pycozmo.anim\_encoder), 64  
 AnimClip (class in pycozmo.anim\_encoder), 62  
 AnimClips (class in pycozmo.anim\_encoder), 62  
 AnimEvent (class in pycozmo.anim\_encoder), 65  
 AnimFaceAnimation (class in pycozmo.anim\_encoder), 65  
 AnimHead (class in pycozmo.protocol\_encoder), 96  
 AnimHeadAngle (class in pycozmo.anim\_encoder), 63  
 AnimKeyframe (class in pycozmo.anim\_encoder), 63  
 AnimLift (class in pycozmo.protocol\_encoder), 97  
 AnimLiftHeight (class in pycozmo.anim\_encoder), 63  
 AnimTurnToRecordedHeading (class in pycozmo.anim\_encoder), 64  
 Annoyance (class in pycozmo.expressions.expressions), 52  
 AppendPathSegArc (class in pycozmo.protocol\_encoder), 99  
 AppendPathSegLine (class in pycozmo.protocol\_encoder), 99  
 AppendPathSegPointTurn (class in pycozmo.protocol\_encoder), 100  
 ARE\_WHEELS\_MOVING (pycozmo.robot.RobotStatusFlag attribute), 140  
 args (pycozmo.audiokinetic.exception.AudioKineticBaseError attribute), 40  
 args (pycozmo.audiokinetic.exception.AudioKineticFormatError attribute), 40  
 args (pycozmo.audiokinetic.exception.AudioKineticIOError attribute), 40  
 args (pycozmo.exception.ConnectionTimeout attribute), 80  
 args (pycozmo.exception.NoSpace attribute), 80  
 args (pycozmo.exception.PyCozmoConnectionError attribute), 80  
 args (pycozmo.exception.PyCozmoException attribute), 80  
 args (pycozmo.exception.Timeout attribute), 80  
 args (pycozmo.protocol\_encoder.DebugData attribute), 103  
 Argument (class in pycozmo.protocol\_ast), 87  
 Asleep (class in pycozmo.expressions.expressions), 56  
 AudioKineticBaseError, 40  
 AudioKineticFormatError, 40  
 AudioKineticIOError, 40  
 auto\_exposure\_enabled (py-

- cozmo.protocol\_encoder.SetCameraParams* (attribute), 129
- auto\_started* (*pycozmo.protocol\_encoder.MotorCalibration* attribute), 114
- axis* (*pycozmo.protocol\_encoder.ObjectUpAxisChanged* attribute), 123
- axis\_of\_accel* (*pycozmo.protocol\_encoder.ObjectMoved* attribute), 120
- ## B
- backpack\_touch\_sensor\_raw* (*pycozmo.protocol\_encoder.RobotState* attribute), 127
- BaseWindow* (class in *pycozmo.window*), 147
- battery\_level* (*pycozmo.protocol\_encoder.ObjectPowerLevel* attribute), 120
- battery\_voltage* (*pycozmo.protocol\_encoder.RobotState* attribute), 127
- BAYER* (*pycozmo.protocol\_encoder.ImageEncoding* attribute), 111
- Behavior* (class in *pycozmo.behavior*), 67
- behavior\_id* (*pycozmo.behavior.ReactionTrigger* attribute), 67
- bend* (*pycozmo.procedural\_face.ProceduralLid* attribute), 84
- BinaryReader* (class in *pycozmo.protocol\_utils*), 138
- BinaryWriter* (class in *pycozmo.protocol\_utils*), 139
- Block\_LIGHTCUBE1* (*pycozmo.protocol\_encoder.ObjectType* attribute), 122
- Block\_LIGHTCUBE2* (*pycozmo.protocol\_encoder.ObjectType* attribute), 122
- Block\_LIGHTCUBE3* (*pycozmo.protocol\_encoder.ObjectType* attribute), 122
- Block\_LIGHTCUBE\_GHOST* (*pycozmo.protocol\_encoder.ObjectType* attribute), 122
- blue* (in module *pycozmo.lights*), 83
- blue\_light* (in module *pycozmo.lights*), 83
- body\_color* (*pycozmo.protocol\_encoder.BodyInfo* attribute), 101
- body\_hw\_version* (*pycozmo.protocol\_encoder.BodyInfo* attribute), 101
- BodyColor* (class in *pycozmo.protocol\_encoder*), 100
- BodyInfo* (class in *pycozmo.protocol\_encoder*), 101
- BoolArgument* (class in *pycozmo.protocol\_ast*), 88
- Boredom* (class in *pycozmo.expressions.expressions*), 55
- Brain* (class in *pycozmo.brain*), 68
- Bridge\_LONG* (*pycozmo.protocol\_encoder.ObjectType* attribute), 122
- Bridge\_SHORT* (*pycozmo.protocol\_encoder.ObjectType* attribute), 122
- buffer* (*pycozmo.protocol\_utils.BinaryReader* attribute), 138
- ButtonPressed* (class in *pycozmo.protocol\_encoder*), 101
- byte\_count* (*pycozmo.protocol\_encoder.FirmwareUpdateResult* attribute), 109
- ## C
- calib\_started* (*pycozmo.protocol\_encoder.MotorCalibration* attribute), 114
- CameraConfig* (class in *pycozmo.camera*), 69
- cancel\_anim()* (*pycozmo.anim\_controller.AnimationController* method), 60
- cancel\_anim()* (*pycozmo.client.Client* method), 69
- CE\_LM\_v15* (*pycozmo.protocol\_encoder.BodyColor* attribute), 101
- center\_x* (*pycozmo.expressions.expressions.Amazement* attribute), 57
- center\_x* (*pycozmo.expressions.expressions.Anger* attribute), 45
- center\_x* (*pycozmo.expressions.expressions.Annoyance* attribute), 53
- center\_x* (*pycozmo.expressions.expressions.Asleep* attribute), 56
- center\_x* (*pycozmo.expressions.expressions.Boredom* attribute), 55
- center\_x* (*pycozmo.expressions.expressions.Confusion* attribute), 56
- center\_x* (*pycozmo.expressions.expressions.Despair* attribute), 49
- center\_x* (*pycozmo.expressions.expressions.Disappointment* attribute), 50
- center\_x* (*pycozmo.expressions.expressions.Disgust* attribute), 47
- center\_x* (*pycozmo.expressions.expressions.Embarrassment* attribute), 51
- center\_x* (*pycozmo.expressions.expressions.Excitement* attribute), 58
- center\_x* (*pycozmo.expressions.expressions.Fear* attribute), 48
- center\_x* (*pycozmo.expressions.expressions.Fury* attribute), 53
- center\_x* (*pycozmo.expressions.expressions.Guilt* attribute), 50
- center\_x* (*pycozmo.expressions.expressions.Happiness* attribute), 46
- center\_x* (*pycozmo.expressions.expressions.Horror* attribute), 51

`center_x` (`pycozmo.expressions.expressions.Neutral` attribute), 44  
`center_x` (`pycozmo.expressions.expressions.Pleading` attribute), 48  
`center_x` (`pycozmo.expressions.expressions.Rejection` attribute), 54  
`center_x` (`pycozmo.expressions.expressions.Sadness` attribute), 45  
`center_x` (`pycozmo.expressions.expressions.Skepticism` attribute), 52  
`center_x` (`pycozmo.expressions.expressions.Surprise` attribute), 47  
`center_x` (`pycozmo.expressions.expressions.Suspicion` attribute), 54  
`center_x` (`pycozmo.expressions.expressions.Tiredness` attribute), 55  
`center_x` (`pycozmo.expressions.expressions.Vulnerability` attribute), 49  
`center_x` (`pycozmo.procedural_face.ProceduralEye` attribute), 84  
`center_x` (`pycozmo.procedural_face.ProceduralFace` attribute), 85  
`center_x` (`pycozmo.protocol_encoder.AppendPathSegArc` attribute), 99  
`center_y` (`pycozmo.expressions.expressions.Amazement` attribute), 57  
`center_y` (`pycozmo.expressions.expressions.Anger` attribute), 45  
`center_y` (`pycozmo.expressions.expressions.Annoyance` attribute), 53  
`center_y` (`pycozmo.expressions.expressions.Asleep` attribute), 56  
`center_y` (`pycozmo.expressions.expressions.Boredom` attribute), 55  
`center_y` (`pycozmo.expressions.expressions.Confusion` attribute), 56  
`center_y` (`pycozmo.expressions.expressions.Despair` attribute), 49  
`center_y` (`pycozmo.expressions.expressions.Disappointment` attribute), 50  
`center_y` (`pycozmo.expressions.expressions.Disgust` attribute), 47  
`center_y` (`pycozmo.expressions.expressions.Embarrassment` attribute), 51  
`center_y` (`pycozmo.expressions.expressions.Excitement` attribute), 58  
`center_y` (`pycozmo.expressions.expressions.Fear` attribute), 48  
`center_y` (`pycozmo.expressions.expressions.Fury` attribute), 53  
`center_y` (`pycozmo.expressions.expressions.Guilt` attribute), 50  
`center_y` (`pycozmo.expressions.expressions.Happiness` attribute), 46  
`center_y` (`pycozmo.expressions.expressions.Horror` attribute), 51  
`center_y` (`pycozmo.expressions.expressions.Neutral` attribute), 44  
`center_y` (`pycozmo.expressions.expressions.Pleading` attribute), 48  
`center_y` (`pycozmo.expressions.expressions.Rejection` attribute), 54  
`center_y` (`pycozmo.expressions.expressions.Sadness` attribute), 45  
`center_y` (`pycozmo.expressions.expressions.Skepticism` attribute), 52  
`center_y` (`pycozmo.expressions.expressions.Surprise` attribute), 47  
`center_y` (`pycozmo.expressions.expressions.Suspicion` attribute), 54  
`center_y` (`pycozmo.expressions.expressions.Tiredness` attribute), 55  
`center_y` (`pycozmo.expressions.expressions.Vulnerability` attribute), 49  
`center_y` (`pycozmo.procedural_face.ProceduralEye` attribute), 84  
`center_y` (`pycozmo.procedural_face.ProceduralFace` attribute), 85  
`center_y` (`pycozmo.protocol_encoder.AppendPathSegArc` attribute), 99  
`Charger_Basic` (`pycozmo.protocol_encoder.ObjectType` attribute), 122  
`check_assets()` (in module `pycozmo.util`), 147  
`choose()` (`pycozmo.activity.Activity` method), 59  
`choose_member()` (`pycozmo.anim.AnimationGroup` method), 60  
`chunk_debug` (`pycozmo.protocol_encoder.ImageChunk` attribute), 110  
`chunk_id` (`pycozmo.protocol_encoder.FirmwareUpdate` attribute), 109  
`chunk_id` (`pycozmo.protocol_encoder.FirmwareUpdateResult` attribute), 109  
`chunk_id` (`pycozmo.protocol_encoder.ImageChunk` attribute), 110  
`clear()` (`pycozmo.anim_controller.AnimationQueue` method), 61  
`clear()` (`pycozmo.protocol_utils.BinaryWriter` method), 139  
`clear_screen()` (`pycozmo.client.Client` method), 69  
`ClearPath` (class in `pycozmo.protocol_encoder`), 102  
`Client` (class in `pycozmo.client`), 69  
`client_drop_count` (`pycozmo.protocol_encoder.AnimationState` attribute), 98  
`cliff_data_raw` (`pycozmo.protocol_encoder.RobotState` attribute), 127

CLIFF\_DETECTED (*pycozmo.robot.RobotStatusFlag attribute*), 140

CliffDetection (*pycozmo.protocol\_encoder.ObjectType attribute*), 123

ClipMetadata (*class in pycozmo.anim\_encoder*), 66

COLLECT\_INTERVAL (*pycozmo.conn.SendThread attribute*), 72

CollisionObstacle (*pycozmo.protocol\_encoder.ObjectType attribute*), 123

Color (*class in pycozmo.lights*), 82

colors (*pycozmo.protocol\_encoder.AnimBackpackLights attribute*), 96

Command (*class in pycozmo.protocol\_ast*), 90

COMMAND (*pycozmo.protocol\_ast.PacketType attribute*), 87

Confusion (*class in pycozmo.expressions.expressions*), 56

Connect (*class in pycozmo.protocol\_ast*), 90

Connect (*class in pycozmo.protocol\_encoder*), 102

CONNECT (*pycozmo.protocol\_ast.PacketType attribute*), 87

connect (*pycozmo.protocol\_encoder.ObjectConnect attribute*), 119

connect () (*in module pycozmo.run*), 142

connect () (*pycozmo.client.Client method*), 70

connect () (*pycozmo.conn.Connection method*), 74

CONNECTED (*pycozmo.conn.Connection attribute*), 74

connected (*pycozmo.protocol\_encoder.ObjectConnectionState attribute*), 119

CONNECTING (*pycozmo.conn.Connection attribute*), 74

Connection (*class in pycozmo.conn*), 74

ConnectionTimeout, 80

cooldown\_time (*pycozmo.anim.AnimationGroupMember attribute*), 60

corner\_radius (*pycozmo.procedural\_face.ProceduralEye attribute*), 84

counter (*pycozmo.protocol\_encoder.Ping attribute*), 125

CubeId (*class in pycozmo.protocol\_encoder*), 102

CubeLights (*class in pycozmo.protocol\_encoder*), 103

curr\_path\_segment (*pycozmo.protocol\_encoder.RobotState attribute*), 127

CustomFixedObstacle (*pycozmo.protocol\_encoder.ObjectType attribute*), 123

CustomType00 (*pycozmo.protocol\_encoder.ObjectType attribute*), 123

CustomType01 (*pycozmo.protocol\_encoder.ObjectType attribute*), 123

CustomType02 (*pycozmo.protocol\_encoder.ObjectType attribute*), 123

CustomType03 (*pycozmo.protocol\_encoder.ObjectType attribute*), 123

CustomType04 (*pycozmo.protocol\_encoder.ObjectType attribute*), 123

CustomType05 (*pycozmo.protocol\_encoder.ObjectType attribute*), 123

CustomType06 (*pycozmo.protocol\_encoder.ObjectType attribute*), 123

CustomType07 (*pycozmo.protocol\_encoder.ObjectType attribute*), 123

CustomType08 (*pycozmo.protocol\_encoder.ObjectType attribute*), 123

CustomType09 (*pycozmo.protocol\_encoder.ObjectType attribute*), 123

CustomType10 (*pycozmo.protocol\_encoder.ObjectType attribute*), 123

CustomType11 (*pycozmo.protocol\_encoder.ObjectType attribute*), 123

CustomType12 (*pycozmo.protocol\_encoder.ObjectType attribute*), 123

CustomType13 (*pycozmo.protocol\_encoder.ObjectType attribute*), 123

CustomType14 (*pycozmo.protocol\_encoder.ObjectType attribute*), 123

CustomType15 (*pycozmo.protocol\_encoder.ObjectType attribute*), 123

CustomType16 (*pycozmo.protocol\_encoder.ObjectType attribute*), 123

CustomType17 (*pycozmo.protocol\_encoder.ObjectType attribute*), 123

CustomType18 (*pycozmo.protocol\_encoder.ObjectType attribute*), 123

CustomType19 (*pycozmo.protocol\_encoder.ObjectType attribute*), 123

CVGA (*pycozmo.protocol\_encoder.ImageResolution attribute*), 112

## D

daemon (*pycozmo.conn.Connection attribute*), 74

daemon (*pycozmo.conn.ReceiveThread attribute*), 71

daemon (*pycozmo.conn.SendThread attribute*), 72

data (*pycozmo.protocol\_base.UnknownCommand attribute*), 92

data (*pycozmo.protocol\_base.UnknownEvent attribute*), 92

data (*pycozmo.protocol\_base.UnknownPacket attribute*), 91

data (*pycozmo.protocol\_encoder.FirmwareUpdate attribute*), 109

data (*pycozmo.protocol\_encoder.ImageChunk attribute*), 110



data (*pycozmo.protocol\_encoder.NvStorageOp attribute*), 117  
 data (*pycozmo.protocol\_encoder.NvStorageOpResult attribute*), 118  
 data\_offset (*pycozmo.audiokinetic.soundbank.SoundBank attribute*), 42  
 deactivate() (*pycozmo.behavior.Behavior method*), 67  
 deactivate\_behavior() (*pycozmo.brain.Brain method*), 68  
 deactivate\_behavior() (*pycozmo.client.Client method*), 70  
 DebugData (*class in pycozmo.protocol\_encoder*), 103  
 decay\_graph (*pycozmo.emotions.EmotionType attribute*), 76  
 decel\_mmmps2 (*pycozmo.protocol\_encoder.AppendPathSegment attribute*), 99  
 decel\_mmmps2 (*pycozmo.protocol\_encoder.AppendPathSegment attribute*), 99  
 decel\_mmmps2 (*pycozmo.protocol\_encoder.AppendPathSegment attribute*), 100  
 decel\_mmmps2 (*pycozmo.protocol\_encoder.PathSegment attribute*), 125  
 decode() (*pycozmo.image\_encoder.ImageDecoder method*), 82  
 define\_pose\_relative\_this() (*pycozmo.util.Pose method*), 146  
 degrees (*pycozmo.util.Angle attribute*), 143  
 del\_all\_handlers() (*pycozmo.behavior.Behavior method*), 67  
 del\_all\_handlers() (*pycozmo.client.Client method*), 70  
 del\_all\_handlers() (*pycozmo.conn.Connection method*), 74  
 del\_all\_handlers() (*pycozmo.event.Dispatcher method*), 79  
 del\_child\_dispatcher() (*pycozmo.behavior.Behavior method*), 67  
 del\_child\_dispatcher() (*pycozmo.client.Client method*), 70  
 del\_child\_dispatcher() (*pycozmo.conn.Connection method*), 74  
 del\_child\_dispatcher() (*pycozmo.event.Dispatcher method*), 79  
 del\_handler() (*pycozmo.behavior.Behavior method*), 67  
 del\_handler() (*pycozmo.client.Client method*), 70  
 del\_handler() (*pycozmo.conn.Connection method*), 74  
 del\_handler() (*pycozmo.event.Dispatcher method*), 79  
 deliver() (*pycozmo.conn.ReceiveThread method*), 71  
 deliver\_sequence() (*pycozmo.conn.ReceiveThread method*), 71  
 deny\_ids() (*pycozmo.filter.Filter method*), 81  
 Despair (*class in pycozmo.expressions.expressions*), 49  
 DEV (*pycozmo.protocol\_encoder.BodyColor attribute*), 101  
 direction (*pycozmo.protocol\_encoder.TurnInPlaceAtSpeed attribute*), 135  
 Disappointment (*class in pycozmo.expressions.expressions*), 50  
 Disconnect (*class in pycozmo.protocol\_ast*), 90  
 Disconnect (*class in pycozmo.protocol\_encoder*), 104  
 DISCONNECT (*pycozmo.protocol\_ast.PacketType attribute*), 87  
 disconnect() (*pycozmo.client.Client method*), 70  
 disconnect() (*pycozmo.conn.Connection method*), 74  
 disconnect() (*pycozmo.conn.ReceiveThread method*), 71  
 Dislike (*class in pycozmo.expressions.expressions*), 47  
 dispatch() (*pycozmo.behavior.Behavior method*), 67  
 dispatch() (*pycozmo.client.Client method*), 70  
 dispatch() (*pycozmo.conn.Connection method*), 74  
 dispatch() (*pycozmo.event.Dispatcher method*), 79  
 Dispatcher (*class in pycozmo.event*), 79  
 display\_image() (*pycozmo.anim\_controller.AnimationController method*), 60  
 display\_image() (*pycozmo.client.Client method*), 70  
 DisplayImage (*class in pycozmo.protocol\_encoder*), 104  
 Distance (*class in pycozmo.util*), 143  
 DoubleArgument (*class in pycozmo.protocol\_ast*), 88  
 drive\_wheels() (*pycozmo.client.Client method*), 70  
 DriveWheels (*class in pycozmo.protocol\_encoder*), 104  
 dumps() (*pycozmo.protocol\_utils.BinaryWriter method*), 139  
 duration\_ms (*pycozmo.protocol\_encoder.AnimHead attribute*), 96  
 duration\_ms (*pycozmo.protocol\_encoder.AnimLift attribute*), 97  
 duration\_ms (*pycozmo.protocol\_encoder.FallingStopped attribute*), 108  
 duration\_sec (*pycozmo.protocol\_encoder.SetHeadAngle attribute*), 129  
 duration\_sec (*pycozmo.protocol\_encoder.SetLiftHeight attribute*), 130

## E

Embarrassment (*class in pycozmo.expressions.expressions*), 51  
 embedded (*pycozmo.audiokinetic.soundbanksinfo.FileInfo attribute*), 43  
 EmotionEvent (*class in pycozmo.emotions*), 76

EmotionType (class in pycozmo.emotions), 76  
 Enable (class in pycozmo.protocol\_encoder), 105  
 enable (pycozmo.protocol\_encoder.EnableColorImages attribute), 106  
 enable (pycozmo.protocol\_encoder.EnableStopOnCliff attribute), 106  
 enable (pycozmo.protocol\_encoder.SetAccessoryDiscover attribute), 128  
 enable (pycozmo.protocol\_encoder.SetHeadLight attribute), 130  
 enable (pycozmo.protocol\_encoder.StreamObjectAccel attribute), 133  
 enable (pycozmo.protocol\_encoder.WifiOff attribute), 136  
 enable\_animations() (pycozmo.anim\_controller.AnimationController method), 60  
 enable\_animations() (pycozmo.client.Client method), 70  
 enable\_camera() (pycozmo.client.Client method), 70  
 enable\_procedural\_face() (pycozmo.anim\_controller.AnimationController method), 61  
 enable\_procedural\_face() (pycozmo.client.Client method), 70  
 EnableAnimationState (class in pycozmo.protocol\_encoder), 105  
 EnableCamera (class in pycozmo.protocol\_encoder), 106  
 EnableColorImages (class in pycozmo.protocol\_encoder), 106  
 enabled\_anim\_tracks (pycozmo.protocol\_encoder.AnimationState attribute), 98  
 EnableStopOnCliff (class in pycozmo.protocol\_encoder), 106  
 encode() (pycozmo.image\_encoder.ImageEncoder method), 82  
 EndAnimation (class in pycozmo.protocol\_encoder), 107  
 ENGINE (pycozmo.protocol\_ast.FrameType attribute), 87  
 ENGINE\_ACT (pycozmo.protocol\_ast.FrameType attribute), 87  
 Enum (class in pycozmo.protocol\_ast), 87  
 EnumArgument (class in pycozmo.protocol\_ast), 89  
 EnumMember (class in pycozmo.protocol\_ast), 87  
 euler\_angles (pycozmo.util.Quaternion attribute), 146  
 Event (class in pycozmo.audiokinetic.soundbank), 41  
 Event (class in pycozmo.event), 77  
 Event (class in pycozmo.protocol\_ast), 90  
 EVENT (pycozmo.protocol\_ast.PacketType attribute), 87  
 event\_id (pycozmo.protocol\_encoder.ExecutePath attribute), 107  
 event\_id (pycozmo.protocol\_encoder.PathFollowingEvent attribute), 125  
 event\_type (pycozmo.protocol\_encoder.PathFollowingEvent attribute), 125  
 EventAction (class in pycozmo.audiokinetic.soundbank), 41  
 EventInfo (class in pycozmo.audiokinetic.soundbanksinfo), 43  
 EvtAnimationCompleted (class in pycozmo.event), 79  
 EvtAudioCompleted (class in pycozmo.event), 79  
 EvtBehaviorDone (class in pycozmo.event), 79  
 EvtChargerOOSChange (class in pycozmo.event), 79  
 EvtCliffDetectedChange (class in pycozmo.event), 78  
 EvtNewRawCameraImage (class in pycozmo.event), 78  
 EvtPacketReceived (class in pycozmo.event), 78  
 EvtReactionTrigger (class in pycozmo.event), 79  
 EvtRobotAnimatingChange (class in pycozmo.event), 78  
 EvtRobotAnimatingIdleChange (class in pycozmo.event), 78  
 EvtRobotAnimBufferFullChange (class in pycozmo.event), 78  
 EvtRobotBodyAccModeChange (class in pycozmo.event), 78  
 EvtRobotCarryingBlockChange (class in pycozmo.event), 78  
 EvtRobotChargingChange (class in pycozmo.event), 78  
 EvtRobotFallingChange (class in pycozmo.event), 78  
 EvtRobotFound (class in pycozmo.event), 77  
 EvtRobotHeadInPositionChange (class in pycozmo.event), 78  
 EvtRobotLiftInPositionChange (class in pycozmo.event), 78  
 EvtRobotMovingChange (class in pycozmo.event), 78  
 EvtRobotOnChargerChange (class in pycozmo.event), 78  
 EvtRobotOrientationChange (class in pycozmo.event), 79  
 EvtRobotPathingChange (class in pycozmo.event), 78  
 EvtRobotPickedUpChange (class in pycozmo.event), 78  
 EvtRobotPickingOrPlacingChange (class in pycozmo.event), 78  
 EvtRobotReady (class in pycozmo.event), 78  
 EvtRobotStateUpdated (class in pycozmo.event),

79

EvtRobotWheelsMovingChange (class in pycozmo.event), 78

Excitement (class in pycozmo.expressions.expressions), 57

ExecutePath (class in pycozmo.protocol\_encoder), 107

exists() (pycozmo.window.ReceiveWindow method), 148

exposure\_ms (pycozmo.protocol\_encoder.SetCameraParameters attribute), 129

eye\_height (pycozmo.expressions.expressions.Amazement attribute), 57

eye\_height (pycozmo.expressions.expressions.Anger attribute), 45

eye\_height (pycozmo.expressions.expressions.Annoyance attribute), 53

eye\_height (pycozmo.expressions.expressions.Asleep attribute), 56

eye\_height (pycozmo.expressions.expressions.Boredom attribute), 55

eye\_height (pycozmo.expressions.expressions.Confusion attribute), 56

eye\_height (pycozmo.expressions.expressions.Despair attribute), 49

eye\_height (pycozmo.expressions.expressions.Disappointment attribute), 50

eye\_height (pycozmo.expressions.expressions.Disgust attribute), 47

eye\_height (pycozmo.expressions.expressions.Embarrassment attribute), 51

eye\_height (pycozmo.expressions.expressions.Excitement attribute), 58

eye\_height (pycozmo.expressions.expressions.Fear attribute), 48

eye\_height (pycozmo.expressions.expressions.Fury attribute), 53

eye\_height (pycozmo.expressions.expressions.Guilt attribute), 50

eye\_height (pycozmo.expressions.expressions.Happiness attribute), 46

eye\_height (pycozmo.expressions.expressions.Horror attribute), 52

eye\_height (pycozmo.expressions.expressions.Neutral attribute), 44

eye\_height (pycozmo.expressions.expressions.Pleading attribute), 48

eye\_height (pycozmo.expressions.expressions.Rejection attribute), 54

eye\_height (pycozmo.expressions.expressions.Sadness attribute), 45

eye\_height (pycozmo.expressions.expressions.Skepticism attribute), 52

eye\_height (pycozmo.expressions.expressions.Surprise attribute), 47

eye\_height (pycozmo.expressions.expressions.Suspicion attribute), 54

eye\_height (pycozmo.expressions.expressions.Tiredness attribute), 55

eye\_height (pycozmo.expressions.expressions.Vulnerability attribute), 49

eye\_height (pycozmo.procedural\_face.ProceduralEye attribute), 84

eye\_height (pycozmo.procedural\_face.ProceduralFace attribute), 85

eye\_height (pycozmo.procedural\_face.ProceduralLid attribute), 84

eye\_width (pycozmo.expressions.expressions.Amazement attribute), 57

eye\_width (pycozmo.expressions.expressions.Anger attribute), 45

eye\_width (pycozmo.expressions.expressions.Annoyance attribute), 53

eye\_width (pycozmo.expressions.expressions.Asleep attribute), 56

eye\_width (pycozmo.expressions.expressions.Boredom attribute), 55

eye\_width (pycozmo.expressions.expressions.Confusion attribute), 56

eye\_width (pycozmo.expressions.expressions.Despair attribute), 49

eye\_width (pycozmo.expressions.expressions.Disappointment attribute), 50

eye\_width (pycozmo.expressions.expressions.Disgust attribute), 47

eye\_width (pycozmo.expressions.expressions.Embarrassment attribute), 51

eye\_width (pycozmo.expressions.expressions.Excitement attribute), 58

eye\_width (pycozmo.expressions.expressions.Fear attribute), 48

eye\_width (pycozmo.expressions.expressions.Fury attribute), 53

eye\_width (pycozmo.expressions.expressions.Guilt attribute), 50

eye\_width (pycozmo.expressions.expressions.Happiness attribute), 46

eye\_width (pycozmo.expressions.expressions.Horror attribute), 52

eye\_width (pycozmo.expressions.expressions.Neutral attribute), 44

eye\_width (pycozmo.expressions.expressions.Pleading attribute), 48

eye\_width (pycozmo.expressions.expressions.Rejection attribute), 54

eye\_width (pycozmo.expressions.expressions.Sadness attribute), 45

eye\_width (pycozmo.expressions.expressions.Skepticism attribute), 46



- attribute), 52  
 eye\_width (pycozmo.expressions.expressions.Surprise attribute), 47  
 eye\_width (pycozmo.expressions.expressions.Suspicion attribute), 54  
 eye\_width (pycozmo.expressions.expressions.Tiredness attribute), 55  
 eye\_width (pycozmo.expressions.expressions.Vulnerability attribute), 49  
 eye\_width (pycozmo.procedural\_face.ProceduralEye attribute), 84  
 eye\_width (pycozmo.procedural\_face.ProceduralFace attribute), 85  
 eye\_width (pycozmo.procedural\_face.ProceduralLid attribute), 84  
 eyes (pycozmo.expressions.expressions.Amazement attribute), 57  
 eyes (pycozmo.expressions.expressions.Anger attribute), 45  
 eyes (pycozmo.expressions.expressions.Annoyance attribute), 53  
 eyes (pycozmo.expressions.expressions.Asleep attribute), 56  
 eyes (pycozmo.expressions.expressions.Boredom attribute), 55  
 eyes (pycozmo.expressions.expressions.Confusion attribute), 57  
 eyes (pycozmo.expressions.expressions.Despair attribute), 49  
 eyes (pycozmo.expressions.expressions.Disappointment attribute), 50  
 eyes (pycozmo.expressions.expressions.Disgust attribute), 47  
 eyes (pycozmo.expressions.expressions.Embarrassment attribute), 51  
 eyes (pycozmo.expressions.expressions.Excitement attribute), 58  
 eyes (pycozmo.expressions.expressions.Fear attribute), 48  
 eyes (pycozmo.expressions.expressions.Fury attribute), 53  
 eyes (pycozmo.expressions.expressions.Guilt attribute), 50  
 eyes (pycozmo.expressions.expressions.Happiness attribute), 46  
 eyes (pycozmo.expressions.expressions.Horror attribute), 52  
 eyes (pycozmo.expressions.expressions.Neutral attribute), 44  
 eyes (pycozmo.expressions.expressions.Pleading attribute), 48  
 eyes (pycozmo.expressions.expressions.Rejection attribute), 54  
 eyes (pycozmo.expressions.expressions.Sadness attribute), 46  
 eyes (pycozmo.expressions.expressions.Skepticism attribute), 52  
 eyes (pycozmo.expressions.expressions.Surprise attribute), 47  
 eyes (pycozmo.expressions.expressions.Suspicion attribute), 54  
 eyes (pycozmo.expressions.expressions.Tiredness attribute), 55  
 eyes (pycozmo.expressions.expressions.Vulnerability attribute), 49  
 eyes (pycozmo.procedural\_face.ProceduralFace attribute), 85
- ## F
- factory\_id (pycozmo.protocol\_encoder.ObjectAvailable attribute), 119  
 factory\_id (pycozmo.protocol\_encoder.ObjectConnect attribute), 119  
 factory\_id (pycozmo.protocol\_encoder.ObjectConnectionState attribute), 119  
 FallingStarted (class in pycozmo.protocol\_encoder), 107  
 FallingStopped (class in pycozmo.protocol\_encoder), 108  
 FArrayArgument (class in pycozmo.protocol\_ast), 89  
 Fear (class in pycozmo.expressions.expressions), 48  
 File (class in pycozmo.audiokinetic.soundbank), 41  
 file\_id (pycozmo.audiokinetic.soundbank.SFX attribute), 42  
 FileInfo (class in pycozmo.audiokinetic.soundbanksinfo), 43  
 Filter (class in pycozmo.filter), 81  
 filter() (pycozmo.filter.Filter method), 81  
 FIN (pycozmo.protocol\_ast.FrameType attribute), 87  
 FirmwareSignature (class in pycozmo.protocol\_encoder), 108  
 FirmwareUpdate (class in pycozmo.protocol\_encoder), 109  
 FirmwareUpdateResult (class in pycozmo.protocol\_encoder), 109  
 first\_seq (pycozmo.frame.Frame attribute), 81  
 FlatMat\_ANKI\_LOGO\_8BIT (pycozmo.protocol\_encoder.ObjectType attribute), 123  
 FlatMat\_GEARS\_4x4 (pycozmo.protocol\_encoder.ObjectType attribute), 123  
 FlatMat\_LAVA\_PLAYTEST (pycozmo.protocol\_encoder.ObjectType attribute), 123  
 FlatMat\_LETTERS\_4x4 (pycozmo.protocol\_encoder.ObjectType attribute), 123

FloatArgument (class in *pycozmo.protocol\_ast*), 87  
 format\_id (*pycozmo.protocol\_encoder.DebugData* attribute), 103  
 forward\_xyz (*pycozmo.util.Matrix44* attribute), 145  
 FPSTimer (class in *pycozmo.util*), 147  
 Frame (class in *pycozmo.frame*), 81  
 FRAME\_RATE (in module *pycozmo.robot*), 140  
 frame\_timestamp (*pycozmo.protocol\_encoder.ImageChunk* attribute), 110  
 FrameType (class in *pycozmo.protocol\_ast*), 87  
 frange() (in module *pycozmo.util*), 147  
 from\_anim\_clip() (*pycozmo.anim.PreprocessedClip* class method), 59  
 from\_bytes() (*pycozmo.frame.Frame* class method), 81  
 from\_bytes() (*pycozmo.protocol\_base.Packet* class method), 91  
 from\_bytes() (*pycozmo.protocol\_base.Struct* class method), 91  
 from\_bytes() (*pycozmo.protocol\_base.UnknownCommand* class method), 92  
 from\_bytes() (*pycozmo.protocol\_base.UnknownEvent* class method), 92  
 from\_bytes() (*pycozmo.protocol\_base.UnknownPacket* class method), 91  
 from\_bytes() (*pycozmo.protocol\_encoder.AbortAnimation* class method), 95  
 from\_bytes() (*pycozmo.protocol\_encoder.AcknowledgeAction* class method), 95  
 from\_bytes() (*pycozmo.protocol\_encoder.AnimationEnded* class method), 97  
 from\_bytes() (*pycozmo.protocol\_encoder.AnimationStarted* class method), 98  
 from\_bytes() (*pycozmo.protocol\_encoder.AnimationState* class method), 98  
 from\_bytes() (*pycozmo.protocol\_encoder.AnimBackpackLights* class method), 96  
 from\_bytes() (*pycozmo.protocol\_encoder.AnimBody* class method), 96  
 from\_bytes() (*pycozmo.protocol\_encoder.AnimHead* class method), 96  
 from\_bytes() (*pycozmo.protocol\_encoder.AnimLift* class method), 97  
 from\_bytes() (*pycozmo.protocol\_encoder.AppendPathSegment* class method), 99  
 from\_bytes() (*pycozmo.protocol\_encoder.AppendPathSegmentLine* class method), 99  
 from\_bytes() (*pycozmo.protocol\_encoder.AppendPathSegmentPoint* class method), 100  
 from\_bytes() (*pycozmo.protocol\_encoder.BodyInfo* class method), 101  
 from\_bytes() (*pycozmo.protocol\_encoder.ButtonPressed* class method), 101  
 from\_bytes() (*pycozmo.protocol\_encoder.ClearPath* class method), 102  
 from\_bytes() (*pycozmo.protocol\_encoder.Connect* class method), 102  
 from\_bytes() (*pycozmo.protocol\_encoder.CubeId* class method), 102  
 from\_bytes() (*pycozmo.protocol\_encoder.CubeLights* class method), 103  
 from\_bytes() (*pycozmo.protocol\_encoder.DebugData* class method), 103  
 from\_bytes() (*pycozmo.protocol\_encoder.Disconnect* class method), 104  
 from\_bytes() (*pycozmo.protocol\_encoder.DisplayImage* class method), 104  
 from\_bytes() (*pycozmo.protocol\_encoder.DriveWheels* class method), 104  
 from\_bytes() (*pycozmo.protocol\_encoder.Enable* class method), 105  
 from\_bytes() (*pycozmo.protocol\_encoder.EnableAnimationState* class method), 105  
 from\_bytes() (*pycozmo.protocol\_encoder.EnableCamera* class method), 106  
 from\_bytes() (*pycozmo.protocol\_encoder.EnableColorImages* class method), 106  
 from\_bytes() (*pycozmo.protocol\_encoder.EnableStopOnCliff* class method), 106  
 from\_bytes() (*pycozmo.protocol\_encoder.EndAnimation* class method), 107  
 from\_bytes() (*pycozmo.protocol\_encoder.ExecutePath* class method), 107  
 from\_bytes() (*pycozmo.protocol\_encoder.FallingStarted* class method), 108  
 from\_bytes() (*pycozmo.protocol\_encoder.FallingStopped* class method), 108  
 from\_bytes() (*pycozmo.protocol\_encoder.FirmwareSignature* class method), 108  
 from\_bytes() (*pycozmo.protocol\_encoder.FirmwareUpdate* class method), 109  
 from\_bytes() (*pycozmo.protocol\_encoder.FirmwareUpdateResult* class method), 109  
 from\_bytes() (*pycozmo.protocol\_encoder.HardwareInfo* class method), 110  
 from\_bytes() (*pycozmo.protocol\_encoder.ImageChunk* class method), 110  
 from\_bytes() (*pycozmo.protocol\_encoder.ImageImuData* class method), 111  
 from\_bytes() (*pycozmo.protocol\_encoder.Keyframe* class method), 112  
 from\_bytes() (*pycozmo.protocol\_encoder.LightState* class method), 113  
 from\_bytes() (*pycozmo.protocol\_encoder.LightStateCenter* class method), 113  
 from\_bytes() (*pycozmo.protocol\_encoder.LightStateSide* class method), 113

class method), 113  
 from\_bytes() (pycozmo.protocol\_encoder.MotorCalibration class method), 114  
 from\_bytes() (pycozmo.protocol\_encoder.MoveHead class method), 114  
 from\_bytes() (pycozmo.protocol\_encoder.MoveLift class method), 115  
 from\_bytes() (pycozmo.protocol\_encoder.NvStorageOp class method), 117  
 from\_bytes() (pycozmo.protocol\_encoder.NvStorageOpResult class method), 118  
 from\_bytes() (pycozmo.protocol\_encoder.ObjectAccel class method), 118  
 from\_bytes() (pycozmo.protocol\_encoder.ObjectAvailable class method), 119  
 from\_bytes() (pycozmo.protocol\_encoder.ObjectConnected class method), 119  
 from\_bytes() (pycozmo.protocol\_encoder.ObjectConnectedState class method), 119  
 from\_bytes() (pycozmo.protocol\_encoder.ObjectMoved class method), 120  
 from\_bytes() (pycozmo.protocol\_encoder.ObjectPowerLevel class method), 120  
 from\_bytes() (pycozmo.protocol\_encoder.ObjectStoppedMoving class method), 121  
 from\_bytes() (pycozmo.protocol\_encoder.ObjectTapFilter class method), 121  
 from\_bytes() (pycozmo.protocol\_encoder.ObjectTapped class method), 122  
 from\_bytes() (pycozmo.protocol\_encoder.ObjectUpAxisChanged class method), 124  
 from\_bytes() (pycozmo.protocol\_encoder.OutputAudio class method), 124  
 from\_bytes() (pycozmo.protocol\_encoder.OutputSilence class method), 124  
 from\_bytes() (pycozmo.protocol\_encoder.PathFollowingEvent class method), 125  
 from\_bytes() (pycozmo.protocol\_encoder.PathSegmentSpeed class method), 125  
 from\_bytes() (pycozmo.protocol\_encoder.Ping class method), 125  
 from\_bytes() (pycozmo.protocol\_encoder.RecordHeading class method), 126  
 from\_bytes() (pycozmo.protocol\_encoder.RobotDelocalized class method), 126  
 from\_bytes() (pycozmo.protocol\_encoder.RobotPoked class method), 127  
 from\_bytes() (pycozmo.protocol\_encoder.RobotState class method), 127  
 from\_bytes() (pycozmo.protocol\_encoder.SetAccessoryDiscovery class method), 128  
 from\_bytes() (pycozmo.protocol\_encoder.SetCameraParams class method), 129  
 from\_bytes() (pycozmo.protocol\_encoder.SetHeadAngle class method), 129  
 from\_bytes() (pycozmo.protocol\_encoder.SetHeadLight class method), 130  
 from\_bytes() (pycozmo.protocol\_encoder.SetLiftHeight class method), 130  
 from\_bytes() (pycozmo.protocol\_encoder.SetOrigin class method), 131  
 from\_bytes() (pycozmo.protocol\_encoder.SetRobotVolume class method), 131  
 from\_bytes() (pycozmo.protocol\_encoder.ShutdownRobot class method), 131  
 from\_bytes() (pycozmo.protocol\_encoder.StartAnimation class method), 132  
 from\_bytes() (pycozmo.protocol\_encoder.StartMotorCalibration class method), 132  
 from\_bytes() (pycozmo.protocol\_encoder.StopAllMotors class method), 133  
 from\_bytes() (pycozmo.protocol\_encoder.StreamObjectAccel class method), 133  
 from\_bytes() (pycozmo.protocol\_encoder.SyncTime class method), 133  
 from\_bytes() (pycozmo.protocol\_encoder.TrimPath class method), 134  
 from\_bytes() (pycozmo.protocol\_encoder.TurnInPlace class method), 134  
 from\_bytes() (pycozmo.protocol\_encoder.TurnInPlaceAtSpeed class method), 135  
 from\_bytes() (pycozmo.protocol\_encoder.TurnToRecordedHeading class method), 135  
 from\_bytes() (pycozmo.protocol\_encoder.WifiOff class method), 136  
 from\_dict() (pycozmo.anim\_encoder.AnimBackpackLights class method), 64  
 from\_dict() (pycozmo.anim\_encoder.AnimBase class method), 62  
 from\_dict() (pycozmo.anim\_encoder.AnimBodyMotion class method), 64  
 from\_dict() (pycozmo.anim\_encoder.AnimClip class method), 62  
 from\_dict() (pycozmo.anim\_encoder.AnimClips class method), 62  
 from\_dict() (pycozmo.anim\_encoder.AnimEvent class method), 65  
 from\_dict() (pycozmo.anim\_encoder.AnimFaceAnimation class method), 65  
 from\_dict() (pycozmo.anim\_encoder.AnimHeadAngle class method), 63  
 from\_dict() (pycozmo.anim\_encoder.AnimKeyframe class method), 63  
 from\_dict() (pycozmo.anim\_encoder.AnimLiftHeight class method), 63  
 from\_dict() (pycozmo.anim\_encoder.AnimLight class method), 63  
 from\_dict() (pycozmo.anim\_encoder.AnimProceduralFace

class method), 65  
 from\_dict() (pycozmo.anim\_encoder.AnimRecordHeading class method), 63  
 from\_dict() (pycozmo.anim\_encoder.AnimRobotAudio class method), 65  
 from\_dict() (pycozmo.anim\_encoder.AnimTurnToRecordedHeading class method), 64  
 from\_fb() (pycozmo.anim\_encoder.AnimBackpackLights class method), 65  
 from\_fb() (pycozmo.anim\_encoder.AnimBase class method), 62  
 from\_fb() (pycozmo.anim\_encoder.AnimBodyMotion class method), 64  
 from\_fb() (pycozmo.anim\_encoder.AnimClip class method), 62  
 from\_fb() (pycozmo.anim\_encoder.AnimClips class method), 62  
 from\_fb() (pycozmo.anim\_encoder.AnimEvent class method), 65  
 from\_fb() (pycozmo.anim\_encoder.AnimFaceAnimation class method), 65  
 from\_fb() (pycozmo.anim\_encoder.AnimHeadAngle class method), 63  
 from\_fb() (pycozmo.anim\_encoder.AnimKeyframe class method), 63  
 from\_fb() (pycozmo.anim\_encoder.AnimLiftHeight class method), 63  
 from\_fb() (pycozmo.anim\_encoder.AnimProceduralFace class method), 65  
 from\_fb() (pycozmo.anim\_encoder.AnimRecordHeading class method), 63  
 from\_fb() (pycozmo.anim\_encoder.AnimRobotAudio class method), 65  
 from\_fb() (pycozmo.anim\_encoder.AnimTurnToRecordedHeading class method), 64  
 from\_fb\_file() (pycozmo.anim\_encoder.AnimClips class method), 62  
 from\_fb\_stream() (pycozmo.anim\_encoder.AnimClips class method), 62  
 from\_int16() (pycozmo.lights.Color class method), 82  
 from\_json() (pycozmo.anim.AnimationGroup class method), 60  
 from\_json() (pycozmo.anim.AnimationGroupMember class method), 60  
 from\_json() (pycozmo.behavior.ReactionTrigger class method), 67  
 from\_json() (pycozmo.emotions.EmotionEvent class method), 76  
 from\_json\_file() (pycozmo.anim\_encoder.AnimClips class method), 62  
 from\_json\_stream() (pycozmo.anim\_encoder.AnimClips class method), 62  
 from\_reader() (pycozmo.frame.Frame class method), 81  
 from\_reader() (pycozmo.protocol\_base.Packet class method), 91  
 from\_reader() (pycozmo.protocol\_base.Struct class method), 91  
 from\_reader() (pycozmo.protocol\_base.UnknownCommand class method), 92  
 from\_reader() (pycozmo.protocol\_base.UnknownEvent class method), 92  
 from\_reader() (pycozmo.protocol\_base.UnknownPacket class method), 91  
 from\_reader() (pycozmo.protocol\_encoder.AbortAnimation class method), 95  
 from\_reader() (pycozmo.protocol\_encoder.AcknowledgeAction class method), 95  
 from\_reader() (pycozmo.protocol\_encoder.AnimationEnded class method), 97  
 from\_reader() (pycozmo.protocol\_encoder.AnimationStarted class method), 98  
 from\_reader() (pycozmo.protocol\_encoder.AnimationState class method), 98  
 from\_reader() (pycozmo.protocol\_encoder.AnimBackpackLights class method), 96  
 from\_reader() (pycozmo.protocol\_encoder.AnimBody class method), 96  
 from\_reader() (pycozmo.protocol\_encoder.AnimHead class method), 97  
 from\_reader() (pycozmo.protocol\_encoder.AnimLift class method), 97  
 from\_reader() (pycozmo.protocol\_encoder.AppendPathSegArc class method), 99  
 from\_reader() (pycozmo.protocol\_encoder.AppendPathSegLine class method), 99  
 from\_reader() (pycozmo.protocol\_encoder.AppendPathSegPointTurn class method), 100  
 from\_reader() (pycozmo.protocol\_encoder.BodyInfo class method), 101

<code>from_reader()</code> <i>cozmo.protocol_encoder.ButtonPressed</i> method), 101	(py- class	<code>from_reader()</code> <i>cozmo.protocol_encoder.FirmwareUpdate</i> class method), 109	(py-
<code>from_reader()</code> <i>cozmo.protocol_encoder.ClearPath</i> method), 102	(py- class	<code>from_reader()</code> <i>cozmo.protocol_encoder.FirmwareUpdateResult</i> class method), 109	(py-
<code>from_reader()</code> ( <i>pycozmo.protocol_encoder.Connect</i> class method), 102		<code>from_reader()</code> <i>cozmo.protocol_encoder.HardwareInfo</i> method), 110	(py- class
<code>from_reader()</code> ( <i>pycozmo.protocol_encoder.CubeId</i> class method), 102		<code>from_reader()</code> <i>cozmo.protocol_encoder.ImageChunk</i> method), 110	(py- class
<code>from_reader()</code> <i>cozmo.protocol_encoder.CubeLights</i> method), 103	(py- class	<code>from_reader()</code> <i>cozmo.protocol_encoder.ImageImuData</i> method), 111	(py- class
<code>from_reader()</code> <i>cozmo.protocol_encoder.DebugData</i> method), 103	(py- class	<code>from_reader()</code> <i>cozmo.protocol_encoder.Keyframe</i> method), 112	(py- class
<code>from_reader()</code> <i>cozmo.protocol_encoder.Disconnect</i> method), 104	(py- class	<code>from_reader()</code> <i>cozmo.protocol_encoder.LightState</i> method), 113	(py- class
<code>from_reader()</code> <i>cozmo.protocol_encoder.DisplayImage</i> method), 104	(py- class	<code>from_reader()</code> <i>cozmo.protocol_encoder.LightStateCenter</i> class method), 113	(py-
<code>from_reader()</code> <i>cozmo.protocol_encoder.DriveWheels</i> method), 104	(py- class	<code>from_reader()</code> <i>cozmo.protocol_encoder.LightStateSide</i> method), 113	(py- class
<code>from_reader()</code> ( <i>pycozmo.protocol_encoder.Enable</i> class method), 105		<code>from_reader()</code> <i>cozmo.protocol_encoder.MotorCalibration</i> class method), 114	(py-
<code>from_reader()</code> <i>cozmo.protocol_encoder.EnableAnimationState</i> class method), 105	(py-	<code>from_reader()</code> <i>cozmo.protocol_encoder.MoveHead</i> method), 114	(py- class
<code>from_reader()</code> <i>cozmo.protocol_encoder.EnableCamera</i> method), 106	(py- class	<code>from_reader()</code> ( <i>pycozmo.protocol_encoder.MoveLift</i> class method), 115	
<code>from_reader()</code> <i>cozmo.protocol_encoder.EnableColorImages</i> class method), 106	(py-	<code>from_reader()</code> <i>cozmo.protocol_encoder.NvStorageOp</i> method), 117	(py- class
<code>from_reader()</code> <i>cozmo.protocol_encoder.EnableStopOnCliff</i> class method), 106	(py-	<code>from_reader()</code> <i>cozmo.protocol_encoder.NvStorageOpResult</i> class method), 118	(py-
<code>from_reader()</code> <i>cozmo.protocol_encoder.EndAnimation</i> method), 107	(py- class	<code>from_reader()</code> <i>cozmo.protocol_encoder.ObjectAccel</i> method), 118	(py- class
<code>from_reader()</code> <i>cozmo.protocol_encoder.ExecutePath</i> method), 107	(py- class	<code>from_reader()</code> <i>cozmo.protocol_encoder.ObjectAvailable</i> class method), 119	(py-
<code>from_reader()</code> <i>cozmo.protocol_encoder.FallingStarted</i> method), 108	(py- class	<code>from_reader()</code> <i>cozmo.protocol_encoder.ObjectConnect</i> method), 119	(py- class
<code>from_reader()</code> <i>cozmo.protocol_encoder.FallingStopped</i> class method), 108	(py-	<code>from_reader()</code> <i>cozmo.protocol_encoder.ObjectConnectionState</i> class method), 120	(py-
<code>from_reader()</code> <i>cozmo.protocol_encoder.FirmwareSignature</i> class method), 108	(py-	<code>from_reader()</code>	(py-



<code>cozmo.protocol_encoder.ObjectMoved</code>	class	<code>method</code> ), 130
<code>from_reader()</code>	(py- <code>cozmo.protocol_encoder.ObjectPowerLevel</code> class method), 121	<code>from_reader()</code> (py- <code>cozmo.protocol_encoder.SetLiftHeight</code> method), 130
<code>from_reader()</code>	(py- <code>cozmo.protocol_encoder.ObjectStoppedMoving</code> class method), 121	<code>from_reader()</code> (py- <code>cozmo.protocol_encoder.SetOrigin</code> method), 131
<code>from_reader()</code>	(py- <code>cozmo.protocol_encoder.ObjectTapFiltered</code> class method), 121	<code>from_reader()</code> (py- <code>cozmo.protocol_encoder.SetRobotVolume</code> class method), 131
<code>from_reader()</code>	(py- <code>cozmo.protocol_encoder.ObjectTapped</code> class method), 122	<code>from_reader()</code> (py- <code>cozmo.protocol_encoder.ShutdownRobot</code> class method), 131
<code>from_reader()</code>	(py- <code>cozmo.protocol_encoder.ObjectUpAxisChanged</code> class method), 124	<code>from_reader()</code> (py- <code>cozmo.protocol_encoder.StartAnimation</code> class method), 132
<code>from_reader()</code>	(py- <code>cozmo.protocol_encoder.OutputAudio</code> class method), 124	<code>from_reader()</code> (py- <code>cozmo.protocol_encoder.StartMotorCalibration</code> class method), 132
<code>from_reader()</code>	(py- <code>cozmo.protocol_encoder.OutputSilence</code> class method), 124	<code>from_reader()</code> (py- <code>cozmo.protocol_encoder.StopAllMotors</code> class method), 133
<code>from_reader()</code>	(py- <code>cozmo.protocol_encoder.PathFollowingEvent</code> class method), 125	<code>from_reader()</code> (py- <code>cozmo.protocol_encoder.StreamObjectAccel</code> class method), 133
<code>from_reader()</code>	(py- <code>cozmo.protocol_encoder.PathSegmentSpeed</code> class method), 125	<code>from_reader()</code> (py- <code>cozmo.protocol_encoder.SyncTime</code> class method), 133
<code>from_reader()</code>	(pycozmo.protocol_encoder.Ping class method), 126	<code>from_reader()</code> (pycozmo.protocol_encoder.TrimPath class method), 134
<code>from_reader()</code>	(py- <code>cozmo.protocol_encoder.RecordHeading</code> class method), 126	<code>from_reader()</code> (py- <code>cozmo.protocol_encoder.TurnInPlace</code> class method), 134
<code>from_reader()</code>	(py- <code>cozmo.protocol_encoder.RobotDelocalized</code> class method), 126	<code>from_reader()</code> (py- <code>cozmo.protocol_encoder.TurnInPlaceAtSpeed</code> class method), 135
<code>from_reader()</code>	(py- <code>cozmo.protocol_encoder.RobotPoked</code> class method), 127	<code>from_reader()</code> (py- <code>cozmo.protocol_encoder.TurnToRecordedHeading</code> class method), 135
<code>from_reader()</code>	(py- <code>cozmo.protocol_encoder.RobotState</code> class method), 127	<code>from_reader()</code> (pycozmo.protocol_encoder.WifiOff class method), 136
<code>from_reader()</code>	(py- <code>cozmo.protocol_encoder.SetAccessoryDiscovery</code> class method), 128	<code>from_x</code> (pycozmo.protocol_encoder.AppendPathSegLine attribute), 99
<code>from_reader()</code>	(py- <code>cozmo.protocol_encoder.SetCameraParams</code> class method), 129	<code>from_y</code> (pycozmo.protocol_encoder.AppendPathSegLine attribute), 99
<code>from_reader()</code>	(py- <code>cozmo.protocol_encoder.SetHeadAngle</code> class method), 129	<code>fspec</code> (pycozmo.audiokinetic.soundbank.SoundBank at- tribute), 42
<code>from_reader()</code>	(py- <code>cozmo.protocol_encoder.SetHeadLight</code> class	<code>Fury</code> (class in pycozmo.expressions.expressions), 53

<code>generate_argument_assignments()</code>	( <code>pycozmo.protocol_generator.ProtocolGenerator</code> method), 137	<code>get()</code>	( <code>pycozmo.anim_controller.AnimationQueue</code> method), 61
<code>generate_argument_defaults()</code>	( <code>pycozmo.protocol_generator.ProtocolGenerator</code> method), 137	<code>get()</code>	( <code>pycozmo.window.ReceiveWindow</code> method), 148
<code>generate_argument_methods()</code>	( <code>pycozmo.protocol_generator.ProtocolGenerator</code> method), 137	<code>get()</code>	( <code>pycozmo.window.SendWindow</code> method), 148
<code>generate_enum()</code>	( <code>pycozmo.protocol_generator.ProtocolGenerator</code> method), 137	<code>get_anim_names()</code>	( <code>pycozmo.client.Client</code> method), 70
<code>generate_enum_validation()</code>	( <code>pycozmo.protocol_generator.ProtocolGenerator</code> method), 137	<code>get_black()</code>	( <code>pycozmo.procedural_face.ProceduralLid</code> class method), 84
<code>generate_farray_validation()</code>	( <code>pycozmo.protocol_generator.ProtocolGenerator</code> method), 137	<code>get_camera_matrix()</code>	( <code>pycozmo.camera.CameraConfig</code> method), 69
<code>generate_group_map()</code>	( <code>pycozmo.protocol_generator.ProtocolGenerator</code> method), 137	<code>get_clip_metadata()</code>	(in module <code>pycozmo.anim_encoder</code> ), 66
<code>generate_id_map()</code>	( <code>pycozmo.protocol_generator.ProtocolGenerator</code> method), 137	<code>get_cozmo_anim_dir()</code>	(in module <code>pycozmo.util</code> ), 147
<code>generate_len_method()</code>	( <code>pycozmo.protocol_generator.ProtocolGenerator</code> method), 137	<code>get_cozmo_asset_dir()</code>	(in module <code>pycozmo.util</code> ), 147
<code>generate_packet()</code>	( <code>pycozmo.protocol_generator.ProtocolGenerator</code> method), 137	<code>get_debug_message()</code>	(in module <code>pycozmo.robot_debug</code> ), 142
<code>generate_packet_argument_assignments()</code>	( <code>pycozmo.protocol_generator.ProtocolGenerator</code> method), 137	<code>get_farray_size()</code>	(in module <code>pycozmo.protocol_utils</code> ), 138
<code>generate_packet_decoding()</code>	( <code>pycozmo.protocol_generator.ProtocolGenerator</code> method), 137	<code>get_id()</code>	( <code>pycozmo.behavior.Behavior</code> method), 67
<code>generate_packet_encoding()</code>	( <code>pycozmo.protocol_generator.ProtocolGenerator</code> method), 137	<code>get_log_level()</code>	(in module <code>pycozmo.robot_debug</code> ), 142
<code>generate_packet_slots()</code>	( <code>pycozmo.protocol_generator.ProtocolGenerator</code> method), 137	<code>get_object_farray_size()</code>	(in module <code>pycozmo.protocol_utils</code> ), 138
<code>generate_repr_method()</code>	( <code>pycozmo.protocol_generator.ProtocolGenerator</code> method), 137	<code>get_object_size()</code>	(in module <code>pycozmo.protocol_utils</code> ), 138
<code>generate_string_validation()</code>	( <code>pycozmo.protocol_generator.ProtocolGenerator</code> method), 137	<code>get_pycozmo_dir()</code>	(in module <code>pycozmo.util</code> ), 147
<code>generate_struct()</code>	( <code>pycozmo.protocol_generator.ProtocolGenerator</code> method), 137	<code>get_size()</code>	(in module <code>pycozmo.protocol_utils</code> ), 138
<code>generate_varray_validation()</code>	( <code>pycozmo.protocol_generator.ProtocolGenerator</code> method), 137	<code>get_string_size()</code>	(in module <code>pycozmo.protocol_utils</code> ), 138
		<code>get_varray_size()</code>	(in module <code>pycozmo.protocol_utils</code> ), 138
		<code>getName()</code>	( <code>pycozmo.conn.Connection</code> method), 74
		<code>getName()</code>	( <code>pycozmo.conn.ReceiveThread</code> method), 71
		<code>getName()</code>	( <code>pycozmo.conn.SendThread</code> method), 73
		<code>go_to_pose()</code>	( <code>pycozmo.client.Client</code> method), 70
		<code>green</code>	(in module <code>pycozmo.lights</code> ), 82
		<code>green_light</code>	(in module <code>pycozmo.lights</code> ), 83
		<code>Guilt</code>	(class in <code>pycozmo.expressions.expressions</code> ), 50
		<code>gyro_x</code>	( <code>pycozmo.protocol_encoder.RobotState</code> attribute), 127
		<code>gyro_y</code>	( <code>pycozmo.protocol_encoder.RobotState</code> attribute), 127
		<code>gyro_z</code>	( <code>pycozmo.protocol_encoder.RobotState</code> attribute), 127
		<b>H</b>	
		<code>half_eye_height</code>	( <code>pycozmo.expressions.expressions.Amazement</code> attribute), 57
		<code>half_eye_height</code>	( <code>pycozmo.expressions.expressions.Anger</code> attribute), 45

half_eye_height cozmo.expressions.expressions.Annoyance attribute), 53	(py-	half_eye_height cozmo.expressions.expressions.Skepticism attribute), 52	(py-
half_eye_height cozmo.expressions.expressions.Asleep tribute), 56	(py- at-	half_eye_height cozmo.expressions.expressions.Surprise tribute), 47	(py- at-
half_eye_height cozmo.expressions.expressions.Boredom attribute), 55	(py-	half_eye_height cozmo.expressions.expressions.Suspicion attribute), 54	(py-
half_eye_height cozmo.expressions.expressions.Confusion attribute), 57	(py-	half_eye_height cozmo.expressions.expressions.Tiredness attribute), 55	(py-
half_eye_height cozmo.expressions.expressions.Despair tribute), 49	(py- at-	half_eye_height cozmo.expressions.expressions.Vulnerability tribute), 49	(py- at-
half_eye_height cozmo.expressions.expressions.Disappointment attribute), 50	(py-	half_eye_height cozmo.procedural_face.ProceduralEye tribute), 84	(py- at-
half_eye_height cozmo.expressions.expressions.Disgust tribute), 47	(py- at-	half_eye_height cozmo.procedural_face.ProceduralFace tribute), 85	(py- at-
half_eye_height cozmo.expressions.expressions.Embarrassment attribute), 51	(py-	half_eye_height cozmo.procedural_face.ProceduralLid tribute), 84	(py- at-
half_eye_height cozmo.expressions.expressions.Excitement attribute), 58	(py-	half_eye_width cozmo.expressions.expressions.Amazement attribute), 57	(py-
half_eye_height cozmo.expressions.expressions.Fear attribute), 48	(py- attribute),	half_eye_width cozmo.expressions.expressions.Anger tribute), 45	(py- at-
half_eye_height cozmo.expressions.expressions.Fury attribute), 53	(py- attribute),	half_eye_width cozmo.expressions.expressions.Annoyance tribute), 53	(py- at-
half_eye_height cozmo.expressions.expressions.Guilt attribute), 50	(py- attribute),	half_eye_width cozmo.expressions.expressions.Asleep tribute), 56	(py- at-
half_eye_height cozmo.expressions.expressions.Happiness attribute), 46	(py-	half_eye_width cozmo.expressions.expressions.Boredom tribute), 55	(py-
half_eye_height cozmo.expressions.expressions.Horror tribute), 52	(py- at-	half_eye_width cozmo.expressions.expressions.Confusion tribute), 57	(py- at-
half_eye_height cozmo.expressions.expressions.Neutral tribute), 44	(py- at-	half_eye_width cozmo.expressions.expressions.Despair tribute), 49	(py- at-
half_eye_height cozmo.expressions.expressions.Pleading tribute), 48	(py-	half_eye_width cozmo.expressions.expressions.Disappointment tribute), 51	(py-
half_eye_height cozmo.expressions.expressions.Rejection tribute), 54	(py- at-	half_eye_width cozmo.expressions.expressions.Disgust tribute), 47	(py- at-
half_eye_height cozmo.expressions.expressions.Sadness tribute), 46	(py- at-	half_eye_width cozmo.expressions.expressions.Embarrassment tribute), 51	(py- at-



half_eye_width	(py-cozmo.expressions.expressions.Excitement attribute), 58	handle_fin()	(pycozmo.conn.ReceiveThread method), 71
half_eye_width	(py-cozmo.expressions.expressions.Fear attribute), 48	handle_frame()	(pycozmo.conn.ReceiveThread method), 71
half_eye_width	(py-cozmo.expressions.expressions.Fury attribute), 53	handle_pkt()	(pycozmo.conn.ReceiveThread method), 71
half_eye_width	(py-cozmo.expressions.expressions.Guilt attribute), 50	handle_reset()	(pycozmo.conn.ReceiveThread method), 71
half_eye_width	(py-cozmo.expressions.expressions.Happiness attribute), 46	Handler	(class in pycozmo.event), 79
half_eye_width	(py-cozmo.expressions.expressions.Horror attribute), 52	Happiness	(class in pycozmo.expressions.expressions), 46
half_eye_width	(py-cozmo.expressions.expressions.Neutral attribute), 45	HardwareInfo	(class in pycozmo.protocol_encoder), 110
half_eye_width	(py-cozmo.expressions.expressions.Pleading attribute), 48	head	(pycozmo.protocol_encoder.StartMotorCalibration attribute), 132
half_eye_width	(py-cozmo.expressions.expressions.Rejection attribute), 54	head	(pycozmo.protocol_encoder.TrimPath attribute), 134
half_eye_width	(py-cozmo.expressions.expressions.Sadness attribute), 46	head_angle_max	(py-cozmo.anim.AnimationGroupMember attribute), 60
half_eye_width	(py-cozmo.expressions.expressions.Skepticism attribute), 52	head_angle_min	(py-cozmo.anim.AnimationGroupMember attribute), 60
half_eye_width	(py-cozmo.expressions.expressions.Surprise attribute), 47	head_angle_rad	(py-cozmo.protocol_encoder.RobotState attribute), 127
half_eye_width	(py-cozmo.expressions.expressions.Suspicion attribute), 54	HEAD_IN_POS	(pycozmo.robot.RobotStatusFlag attribute), 140
half_eye_width	(py-cozmo.expressions.expressions.Tiredness attribute), 55	heartbeat_thread_run()	(pycozmo.brain.Brain method), 68
half_eye_width	(py-cozmo.expressions.expressions.Vulnerability attribute), 49	height	(pycozmo.expressions.expressions.Amazement attribute), 57
half_eye_width	(py-cozmo.procedural_face.ProceduralEye attribute), 84	height	(pycozmo.expressions.expressions.Anger attribute), 45
half_eye_width	(py-cozmo.procedural_face.ProceduralFace attribute), 85	height	(pycozmo.expressions.expressions.Annoyance attribute), 53
half_eye_width	(py-cozmo.procedural_face.ProceduralLid attribute), 84	height	(pycozmo.expressions.expressions.Asleep attribute), 56
		height	(pycozmo.expressions.expressions.Boredom attribute), 55
		height	(pycozmo.expressions.expressions.Confusion attribute), 57
		height	(pycozmo.expressions.expressions.Despair attribute), 49
		height	(pycozmo.expressions.expressions.Disappointment attribute), 51
		height	(pycozmo.expressions.expressions.Disgust attribute), 47
		height	(pycozmo.expressions.expressions.Embarrassment attribute), 51
		height	(pycozmo.expressions.expressions.Excitement attribute), 58
		height	(pycozmo.expressions.expressions.Fear attribute), 48

height (*pycozmo.expressions.expressions.Fury attribute*), 53  
 height (*pycozmo.expressions.expressions.Guilt attribute*), 50  
 height (*pycozmo.expressions.expressions.Happiness attribute*), 46  
 height (*pycozmo.expressions.expressions.Horror attribute*), 52  
 height (*pycozmo.expressions.expressions.Neutral attribute*), 45  
 height (*pycozmo.expressions.expressions.Pleading attribute*), 48  
 height (*pycozmo.expressions.expressions.Rejection attribute*), 54  
 height (*pycozmo.expressions.expressions.Sadness attribute*), 46  
 height (*pycozmo.expressions.expressions.Skepticism attribute*), 52  
 height (*pycozmo.expressions.expressions.Surprise attribute*), 47  
 height (*pycozmo.expressions.expressions.Suspicion attribute*), 54  
 height (*pycozmo.expressions.expressions.Tiredness attribute*), 56  
 height (*pycozmo.expressions.expressions.Vulnerability attribute*), 49  
 height (*pycozmo.procedural\_face.ProceduralEye attribute*), 85  
 height (*pycozmo.procedural\_face.ProceduralFace attribute*), 85  
 height (*pycozmo.procedural\_face.ProceduralLid attribute*), 84  
 height (*pycozmo.robot.LiftPosition attribute*), 141  
 height\_mm (*pycozmo.protocol\_encoder.AnimLift attribute*), 97  
 height\_mm (*pycozmo.protocol\_encoder.SetLiftHeight attribute*), 130  
 hex\_dump() (*in module pycozmo.util*), 147  
 hex\_load() (*in module pycozmo.util*), 147  
 Horror (*class in pycozmo.expressions.expressions*), 51  
 I  
 id (*pycozmo.activity.Activity attribute*), 59  
 id (*pycozmo.audiokinetic.soundbank.Event attribute*), 41  
 id (*pycozmo.audiokinetic.soundbank.EventAction attribute*), 41  
 id (*pycozmo.audiokinetic.soundbank.File attribute*), 41  
 id (*pycozmo.audiokinetic.soundbank.SFX attribute*), 42  
 id (*pycozmo.audiokinetic.soundbank.SoundBank attribute*), 42  
 id (*pycozmo.audiokinetic.soundbanksinfo.EventInfo attribute*), 43  
 id (*pycozmo.audiokinetic.soundbanksinfo.FileInfo attribute*), 43  
 id (*pycozmo.audiokinetic.soundbanksinfo.SoundBankInfo attribute*), 43  
 id (*pycozmo.protocol\_base.Packet attribute*), 91  
 id (*pycozmo.protocol\_base.UnknownCommand attribute*), 92  
 id (*pycozmo.protocol\_base.UnknownEvent attribute*), 92  
 id (*pycozmo.protocol\_base.UnknownPacket attribute*), 91  
 id (*pycozmo.protocol\_encoder.AbortAnimation attribute*), 95  
 id (*pycozmo.protocol\_encoder.AcknowledgeAction attribute*), 95  
 id (*pycozmo.protocol\_encoder.AnimationEnded attribute*), 97  
 id (*pycozmo.protocol\_encoder.AnimationStarted attribute*), 98  
 id (*pycozmo.protocol\_encoder.AnimationState attribute*), 98  
 id (*pycozmo.protocol\_encoder.AnimBackpackLights attribute*), 96  
 id (*pycozmo.protocol\_encoder.AnimBody attribute*), 96  
 id (*pycozmo.protocol\_encoder.AnimHead attribute*), 97  
 id (*pycozmo.protocol\_encoder.AnimLift attribute*), 97  
 id (*pycozmo.protocol\_encoder.AppendPathSegArc attribute*), 99  
 id (*pycozmo.protocol\_encoder.AppendPathSegLine attribute*), 99  
 id (*pycozmo.protocol\_encoder.AppendPathSegPointTurn attribute*), 100  
 id (*pycozmo.protocol\_encoder.BodyInfo attribute*), 101  
 id (*pycozmo.protocol\_encoder.ButtonPressed attribute*), 101  
 id (*pycozmo.protocol\_encoder.ClearPath attribute*), 102  
 id (*pycozmo.protocol\_encoder.Connect attribute*), 102  
 id (*pycozmo.protocol\_encoder.CubeId attribute*), 102  
 id (*pycozmo.protocol\_encoder.CubeLights attribute*), 103  
 id (*pycozmo.protocol\_encoder.DebugData attribute*), 103  
 id (*pycozmo.protocol\_encoder.Disconnect attribute*), 104  
 id (*pycozmo.protocol\_encoder.DisplayImage attribute*), 104  
 id (*pycozmo.protocol\_encoder.DriveWheels attribute*), 105  
 id (*pycozmo.protocol\_encoder.Enable attribute*), 105  
 id (*pycozmo.protocol\_encoder.EnableAnimationState attribute*), 105  
 id (*pycozmo.protocol\_encoder.EnableCamera attribute*), 106  
 id (*pycozmo.protocol\_encoder.EnableColorImages attribute*), 106  
 id (*pycozmo.protocol\_encoder.EnableStopOnCliff attribute*), 106

<code>id (pycozmo.protocol_encoder.EndAnimation attribute),</code>	<code>124</code>
<code>107</code>	<code>id (pycozmo.protocol_encoder.OutputSilence attribute),</code>
<code>id (pycozmo.protocol_encoder.ExecutePath attribute),</code>	<code>124</code>
<code>107</code>	<code>id (pycozmo.protocol_encoder.PathFollowingEvent attribute),</code>
<code>id (pycozmo.protocol_encoder.FallingStarted attribute),</code>	<code>125</code>
<code>108</code>	<code>id (pycozmo.protocol_encoder.Ping attribute),</code>
<code>id (pycozmo.protocol_encoder.FallingStopped attribute),</code>	<code>126</code>
<code>108</code>	<code>id (pycozmo.protocol_encoder.RecordHeading attribute),</code>
<code>id (pycozmo.protocol_encoder.FirmwareSignature attribute),</code>	<code>126</code>
<code>108</code>	<code>id (pycozmo.protocol_encoder.RobotDelocalized attribute),</code>
<code>id (pycozmo.protocol_encoder.FirmwareUpdate attribute),</code>	<code>126</code>
<code>109</code>	<code>id (pycozmo.protocol_encoder.RobotPoked attribute),</code>
<code>id (pycozmo.protocol_encoder.FirmwareUpdateResult attribute),</code>	<code>127</code>
<code>109</code>	<code>id (pycozmo.protocol_encoder.RobotState attribute),</code>
<code>id (pycozmo.protocol_encoder.HardwareInfo attribute),</code>	<code>128</code>
<code>110</code>	<code>id (pycozmo.protocol_encoder.SetAccessoryDiscovery attribute),</code>
<code>id (pycozmo.protocol_encoder.ImageChunk attribute),</code>	<code>128</code>
<code>110</code>	<code>id (pycozmo.protocol_encoder.SetCameraParams attribute),</code>
<code>id (pycozmo.protocol_encoder.ImageImuData attribute),</code>	<code>129</code>
<code>111</code>	<code>id (pycozmo.protocol_encoder.SetHeadAngle attribute),</code>
<code>id (pycozmo.protocol_encoder.Keyframe attribute),</code>	<code>129</code>
<code>112</code>	<code>id (pycozmo.protocol_encoder.SetHeadLight attribute),</code>
<code>id (pycozmo.protocol_encoder.LightStateCenter attribute),</code>	<code>130</code>
<code>113</code>	<code>id (pycozmo.protocol_encoder.SetLiftHeight attribute),</code>
<code>id (pycozmo.protocol_encoder.LightStateSide attribute),</code>	<code>130</code>
<code>113</code>	<code>id (pycozmo.protocol_encoder.SetOrigin attribute),</code>
<code>id (pycozmo.protocol_encoder.MotorCalibration attribute),</code>	<code>131</code>
<code>114</code>	<code>id (pycozmo.protocol_encoder.SetRobotVolume attribute),</code>
<code>id (pycozmo.protocol_encoder.MoveHead attribute),</code>	<code>131</code>
<code>114</code>	<code>id (pycozmo.protocol_encoder.ShutdownRobot attribute),</code>
<code>id (pycozmo.protocol_encoder.MoveLift attribute),</code>	<code>131</code>
<code>115</code>	<code>id (pycozmo.protocol_encoder.StartAnimation attribute),</code>
<code>id (pycozmo.protocol_encoder.NvStorageOp attribute),</code>	<code>132</code>
<code>117</code>	<code>id (pycozmo.protocol_encoder.StartMotorCalibration attribute),</code>
<code>id (pycozmo.protocol_encoder.NvStorageOpResult attribute),</code>	<code>132</code>
<code>118</code>	<code>id (pycozmo.protocol_encoder.StopAllMotors attribute),</code>
<code>id (pycozmo.protocol_encoder.ObjectAccel attribute),</code>	<code>133</code>
<code>118</code>	<code>id (pycozmo.protocol_encoder.StreamObjectAccel attribute),</code>
<code>id (pycozmo.protocol_encoder.ObjectAvailable attribute),</code>	<code>133</code>
<code>119</code>	<code>id (pycozmo.protocol_encoder.SyncTime attribute),</code>
<code>id (pycozmo.protocol_encoder.ObjectConnect attribute),</code>	<code>133</code>
<code>119</code>	<code>id (pycozmo.protocol_encoder.TrimPath attribute),</code>
<code>id (pycozmo.protocol_encoder.ObjectConnectionState attribute),</code>	<code>134</code>
<code>120</code>	<code>id (pycozmo.protocol_encoder.TurnInPlace attribute),</code>
<code>id (pycozmo.protocol_encoder.ObjectMoved attribute),</code>	<code>134</code>
<code>120</code>	<code>id (pycozmo.protocol_encoder.TurnInPlaceAtSpeed attribute),</code>
<code>id (pycozmo.protocol_encoder.ObjectPowerLevel attribute),</code>	<code>135</code>
<code>121</code>	<code>id (pycozmo.protocol_encoder.TurnToRecordedHeading attribute),</code>
<code>id (pycozmo.protocol_encoder.ObjectStoppedMoving attribute),</code>	<code>135</code>
<code>121</code>	<code>id (pycozmo.protocol_encoder.WifiOff attribute),</code>
<code>id (pycozmo.protocol_encoder.ObjectTapFiltered attribute),</code>	<code>136</code>
<code>121</code>	<code>ident (pycozmo.conn.Connection attribute),</code>
<code>id (pycozmo.protocol_encoder.ObjectTapped attribute),</code>	<code>74</code>
<code>122</code>	<code>ident (pycozmo.conn.ReceiveThread attribute),</code>
<code>id (pycozmo.protocol_encoder.ObjectUpAxisChanged attribute),</code>	<code>71</code>
<code>124</code>	<code>ident (pycozmo.conn.SendThread attribute),</code>
<code>id (pycozmo.protocol_encoder.OutputAudio attribute),</code>	<code>73</code>
	<code>IDLE (pycozmo.conn.Connection attribute),</code>
	<code>74</code>
	<code>image (pycozmo.protocol_encoder.DisplayImage attribute),</code>
	<code>104</code>
	<code>image_chunk_count (pycozmo.protocol_encoder.ImageChunk attribute),</code>
	<code>110</code>

image\_encoding (pycozmo.protocol\_encoder.ImageChunk attribute), 110  
 image\_id (pycozmo.protocol\_encoder.ImageChunk attribute), 110  
 image\_id (pycozmo.protocol\_encoder.ImageImuData attribute), 111  
 image\_resolution (pycozmo.protocol\_encoder.EnableCamera attribute), 106  
 image\_resolution (pycozmo.protocol\_encoder.ImageChunk attribute), 110  
 image\_send\_mode (pycozmo.protocol\_encoder.EnableCamera attribute), 106  
 image\_to\_str() (in module pycozmo.image\_encoder), 82  
 ImageChunk (class in pycozmo.protocol\_encoder), 110  
 ImageDecoder (class in pycozmo.image\_encoder), 82  
 ImageEncoder (class in pycozmo.image\_encoder), 82  
 ImageEncoding (class in pycozmo.protocol\_encoder), 111  
 ImageImuData (class in pycozmo.protocol\_encoder), 111  
 ImageResolution (class in pycozmo.protocol\_encoder), 112  
 ImageResolutionCount (pycozmo.protocol\_encoder.ImageResolution attribute), 112  
 ImageResolutionNone (pycozmo.protocol\_encoder.ImageResolution attribute), 112  
 ImageSendMode (class in pycozmo.protocol\_encoder), 112  
 impact\_intensity (pycozmo.protocol\_encoder.FallingStopped attribute), 108  
 in\_column\_order (pycozmo.util.Matrix44 attribute), 145  
 in\_row\_order (pycozmo.util.Matrix44 attribute), 145  
 inches (pycozmo.util.Distance attribute), 143  
 Int16Argument (class in pycozmo.protocol\_ast), 89  
 Int32Argument (class in pycozmo.protocol\_ast), 89  
 Int8Argument (class in pycozmo.protocol\_ast), 88  
 int\_color (pycozmo.lights.Color attribute), 82  
 IntArgument (class in pycozmo.protocol\_ast), 88  
 intensity (pycozmo.protocol\_encoder.ObjectTapFiltered attribute), 121  
 interpolate() (in module pycozmo.procedural\_face), 86  
 invalidate() (pycozmo.util.Pose method), 146  
 InvalidObject (pycozmo.protocol\_encoder.ObjectType attribute), 123  
 is\_absolute (pycozmo.protocol\_encoder.TurnInPlace attribute), 134  
 is\_accurate (pycozmo.util.Pose attribute), 146  
 is\_alive() (pycozmo.conn.Connection method), 75  
 is\_alive() (pycozmo.conn.ReceiveThread method), 72  
 is\_alive() (pycozmo.conn.SendThread method), 73  
 IS\_ANIM\_BUFFER\_FULL (pycozmo.robot.RobotStatusFlag attribute), 141  
 IS\_ANIMATING (pycozmo.robot.RobotStatusFlag attribute), 140  
 IS\_ANIMATING\_IDLE (pycozmo.robot.RobotStatusFlag attribute), 141  
 IS\_BODY\_ACC\_MODE (pycozmo.robot.RobotStatusFlag attribute), 141  
 IS\_CARRYING\_BLOCK (pycozmo.robot.RobotStatusFlag attribute), 141  
 IS\_CHARGER\_OOS (pycozmo.robot.RobotStatusFlag attribute), 141  
 IS\_CHARGING (pycozmo.robot.RobotStatusFlag attribute), 141  
 is\_comparable() (pycozmo.util.Pose method), 146  
 is\_empty() (pycozmo.anim\_controller.AnimationQueue method), 61  
 IS\_FALLING (pycozmo.robot.RobotStatusFlag attribute), 141  
 is\_from\_engine() (pycozmo.protocol\_base.Packet method), 91  
 is\_from\_engine() (pycozmo.protocol\_base.UnknownCommand method), 92  
 is\_from\_engine() (pycozmo.protocol\_base.UnknownEvent method), 92  
 is\_from\_engine() (pycozmo.protocol\_base.UnknownPacket method), 92  
 is\_from\_engine() (pycozmo.protocol\_encoder.AbortAnimation method), 95  
 is\_from\_engine() (pycozmo.protocol\_encoder.AcknowledgeAction method), 95  
 is\_from\_engine() (pycozmo.protocol\_encoder.AnimationEnded method), 97  
 is\_from\_engine() (pycozmo.protocol\_encoder.AnimationStarted method), 98  
 is\_from\_engine() (pycozmo.protocol\_encoder.AnimationStarted method), 98

<i>cozmo.protocol_encoder.AnimationState</i> <i>method</i> ), 98	<i>cozmo.protocol_encoder.Enable</i> 105	<i>method</i> ),
<i>is_from_engine()</i> (py- <i>cozmo.protocol_encoder.AnimBackpackLights</i> <i>method</i> ), 96	<i>is_from_engine()</i> (py- <i>cozmo.protocol_encoder.EnableAnimationState</i> <i>method</i> ), 105	
<i>is_from_engine()</i> (py- <i>cozmo.protocol_encoder.AnimBody</i> <i>method</i> ), 96	<i>is_from_engine()</i> (py- <i>cozmo.protocol_encoder.EnableCamera</i> <i>method</i> ), 106	
<i>is_from_engine()</i> (py- <i>cozmo.protocol_encoder.AnimHead</i> <i>method</i> ), 97	<i>is_from_engine()</i> (py- <i>cozmo.protocol_encoder.EnableColorImages</i> <i>method</i> ), 106	
<i>is_from_engine()</i> (py- <i>cozmo.protocol_encoder.AnimLift</i> <i>method</i> ), 97	<i>is_from_engine()</i> (py- <i>cozmo.protocol_encoder.EnableStopOnCliff</i> <i>method</i> ), 107	
<i>is_from_engine()</i> (py- <i>cozmo.protocol_encoder.AppendPathSegArc</i> <i>method</i> ), 99	<i>is_from_engine()</i> (py- <i>cozmo.protocol_encoder.EndAnimation</i> <i>method</i> ), 107	
<i>is_from_engine()</i> (py- <i>cozmo.protocol_encoder.AppendPathSegLine</i> <i>method</i> ), 100	<i>is_from_engine()</i> (py- <i>cozmo.protocol_encoder.ExecutePath</i> <i>method</i> ), 107	
<i>is_from_engine()</i> (py- <i>cozmo.protocol_encoder.AppendPathSegPointTurn</i> <i>method</i> ), 100	<i>is_from_engine()</i> (py- <i>cozmo.protocol_encoder.FallingStarted</i> <i>method</i> ), 108	
<i>is_from_engine()</i> (py- <i>cozmo.protocol_encoder.BodyInfo</i> <i>method</i> ), 101	<i>is_from_engine()</i> (py- <i>cozmo.protocol_encoder.FallingStopped</i> <i>method</i> ), 108	
<i>is_from_engine()</i> (py- <i>cozmo.protocol_encoder.ButtonPressed</i> <i>method</i> ), 101	<i>is_from_engine()</i> (py- <i>cozmo.protocol_encoder.FirmwareSignature</i> <i>method</i> ), 108	
<i>is_from_engine()</i> (py- <i>cozmo.protocol_encoder.ClearPath</i> <i>method</i> ), 102	<i>is_from_engine()</i> (py- <i>cozmo.protocol_encoder.FirmwareUpdate</i> <i>method</i> ), 109	
<i>is_from_engine()</i> (py- <i>cozmo.protocol_encoder.Connect</i> <i>method</i> ), 102	<i>is_from_engine()</i> (py- <i>cozmo.protocol_encoder.FirmwareUpdateResult</i> <i>method</i> ), 109	
<i>is_from_engine()</i> (py- <i>cozmo.protocol_encoder.CubeId</i> <i>method</i> ), 102	<i>is_from_engine()</i> (py- <i>cozmo.protocol_encoder.HardwareInfo</i> <i>method</i> ), 110	
<i>is_from_engine()</i> (py- <i>cozmo.protocol_encoder.CubeLights</i> <i>method</i> ), 103	<i>is_from_engine()</i> (py- <i>cozmo.protocol_encoder.ImageChunk</i> <i>method</i> ), 110	
<i>is_from_engine()</i> (py- <i>cozmo.protocol_encoder.DebugData</i> <i>method</i> ), 103	<i>is_from_engine()</i> (py- <i>cozmo.protocol_encoder.ImageImuData</i> <i>method</i> ), 111	
<i>is_from_engine()</i> (py- <i>cozmo.protocol_encoder.Disconnect</i> <i>method</i> ), 104	<i>is_from_engine()</i> (py- <i>cozmo.protocol_encoder.Keyframe</i> <i>method</i> ), 112	
<i>is_from_engine()</i> (py- <i>cozmo.protocol_encoder.DisplayImage</i> <i>method</i> ), 104	<i>is_from_engine()</i> (py- <i>cozmo.protocol_encoder.LightStateCenter</i> <i>method</i> ), 113	
<i>is_from_engine()</i> (py- <i>cozmo.protocol_encoder.DriveWheels</i> <i>method</i> ), 105	<i>is_from_engine()</i> (py- <i>cozmo.protocol_encoder.LightStateSide</i> <i>method</i> ), 113	
<i>is_from_engine()</i> (py-	<i>is_from_engine()</i> (py-	



```

        cozmo.protocol_encoder.MotorCalibration
        method), 114
is_from_engine() (py-
    cozmo.protocol_encoder.MoveHead method),
    115
is_from_engine() (py-
    cozmo.protocol_encoder.MoveLift method),
    115
is_from_engine() (py-
    cozmo.protocol_encoder.NvStorageOp
    method), 117
is_from_engine() (py-
    cozmo.protocol_encoder.NvStorageOpResult
    method), 118
is_from_engine() (py-
    cozmo.protocol_encoder.ObjectAccel method),
    118
is_from_engine() (py-
    cozmo.protocol_encoder.ObjectAvailable
    method), 119
is_from_engine() (py-
    cozmo.protocol_encoder.ObjectConnect
    method), 119
is_from_engine() (py-
    cozmo.protocol_encoder.ObjectConnectionState
    method), 120
is_from_engine() (py-
    cozmo.protocol_encoder.ObjectMoved
    method), 120
is_from_engine() (py-
    cozmo.protocol_encoder.ObjectPowerLevel
    method), 121
is_from_engine() (py-
    cozmo.protocol_encoder.ObjectStoppedMoving
    method), 121
is_from_engine() (py-
    cozmo.protocol_encoder.ObjectTapFiltered
    method), 121
is_from_engine() (py-
    cozmo.protocol_encoder.ObjectTapped
    method), 122
is_from_engine() (py-
    cozmo.protocol_encoder.ObjectUpAxisChanged
    method), 124
is_from_engine() (py-
    cozmo.protocol_encoder.OutputAudio method),
    124
is_from_engine() (py-
    cozmo.protocol_encoder.OutputSilence
    method), 124
is_from_engine() (py-
    cozmo.protocol_encoder.PathFollowingEvent
    method), 125
is_from_engine() (pycozmo.protocol_encoder.Ping
    method), 126
is_from_engine() (py-
    cozmo.protocol_encoder.RecordHeading
    method), 126
is_from_engine() (py-
    cozmo.protocol_encoder.RobotDelocalized
    method), 126
is_from_engine() (py-
    cozmo.protocol_encoder.RobotPoked method),
    127
is_from_engine() (py-
    cozmo.protocol_encoder.RobotState method),
    128
is_from_engine() (py-
    cozmo.protocol_encoder.SetAccessoryDiscovery
    method), 128
is_from_engine() (py-
    cozmo.protocol_encoder.SetCameraParams
    method), 129
is_from_engine() (py-
    cozmo.protocol_encoder.SetHeadAngle
    method), 129
is_from_engine() (py-
    cozmo.protocol_encoder.SetHeadLight
    method), 130
is_from_engine() (py-
    cozmo.protocol_encoder.SetLiftHeight
    method), 130
is_from_engine() (py-
    cozmo.protocol_encoder.SetOrigin method),
    131
is_from_engine() (py-
    cozmo.protocol_encoder.SetRobotVolume
    method), 131
is_from_engine() (py-
    cozmo.protocol_encoder.ShutdownRobot
    method), 131
is_from_engine() (py-
    cozmo.protocol_encoder.StartAnimation
    method), 132
is_from_engine() (py-
    cozmo.protocol_encoder.StartMotorCalibration
    method), 132
is_from_engine() (py-
    cozmo.protocol_encoder.StopAllMotors
    method), 133
is_from_engine() (py-
    cozmo.protocol_encoder.StreamObjectAccel
    method), 133
is_from_engine() (py-
    cozmo.protocol_encoder.SyncTime method),
    133
is_from_engine() (py-
    cozmo.protocol_encoder.TrimPath method),

```

134			
is_from_engine()	(py-	is_from_robot()	(py-
cozmo.protocol_encoder.TurnInPlace method),		cozmo.protocol_encoder.AppendPathSegLine	method), 100
134		is_from_robot()	(py-
is_from_engine()	(py-	cozmo.protocol_encoder.AppendPathSegPointTurn	method), 100
cozmo.protocol_encoder.TurnInPlaceAtSpeed		is_from_robot()	(py-
method), 135		cozmo.protocol_encoder.BodyInfo	method),
is_from_engine()	(py-	101	
cozmo.protocol_encoder.TurnToRecordedHeading		is_from_robot()	(py-
method), 135		cozmo.protocol_encoder.ButtonPressed	method), 101
is_from_engine()	(py-	is_from_robot()	(py-
cozmo.protocol_encoder.WifiOff method),		cozmo.protocol_encoder.ClearPath	method),
136		102	
is_from_robot()	(pycozmo.protocol_base.Packet	is_from_robot()	(py-
method), 91		cozmo.protocol_encoder.Connect	method),
is_from_robot()	(py-	102	
cozmo.protocol_base.UnknownCommand		is_from_robot()	(py-
method), 92		cozmo.protocol_encoder.CubeId	method),
is_from_robot()	(py-	102	
cozmo.protocol_base.UnknownEvent method),		is_from_robot()	(py-
92		cozmo.protocol_encoder.CubeLights	method),
is_from_robot()	(py-	103	
cozmo.protocol_base.UnknownPacket method),		is_from_robot()	(py-
92		cozmo.protocol_encoder.DebugData	method),
is_from_robot()	(py-	103	
cozmo.protocol_encoder.AbortAnimation		is_from_robot()	(py-
method), 95		cozmo.protocol_encoder.Disconnect	method),
is_from_robot()	(py-	104	
cozmo.protocol_encoder.AcknowledgeAction		is_from_robot()	(py-
method), 95		cozmo.protocol_encoder.DisplayImage	method), 104
is_from_robot()	(py-	is_from_robot()	(py-
cozmo.protocol_encoder.AnimationEnded		cozmo.protocol_encoder.DriveWheels	method),
method), 97		105	
is_from_robot()	(py-	is_from_robot()	(py-
cozmo.protocol_encoder.AnimationStarted		cozmo.protocol_encoder.Enable	method),
method), 98		105	
is_from_robot()	(py-	is_from_robot()	(py-
cozmo.protocol_encoder.AnimationState		cozmo.protocol_encoder.EnableAnimationState	method), 105
method), 98		is_from_robot()	(py-
is_from_robot()	(py-	cozmo.protocol_encoder.EnableCamera	method), 106
cozmo.protocol_encoder.AnimBackpackLights		is_from_robot()	(py-
method), 96		cozmo.protocol_encoder.EnableColorImages	method), 106
is_from_robot()	(py-	is_from_robot()	(py-
cozmo.protocol_encoder.AnimBody	method),	cozmo.protocol_encoder.EnableStopOnCliff	method), 107
96		is_from_robot()	(py-
is_from_robot()	(py-	cozmo.protocol_encoder.EndAnimation	method), 107
cozmo.protocol_encoder.AnimHead	method),		
97			
is_from_robot()	(py-		
cozmo.protocol_encoder.AnimLift	method),		
97			
is_from_robot()	(py-		
cozmo.protocol_encoder.AppendPathSegArc	method), 99		

<code>is_from_robot()</code> <i>(py-cozmo.protocol_encoder.ExecutePath method), 107</i>	<code>is_from_robot()</code> <i>(py-cozmo.protocol_encoder.ObjectAvailable method), 119</i>
<code>is_from_robot()</code> <i>(py-cozmo.protocol_encoder.FallingStarted method), 108</i>	<code>is_from_robot()</code> <i>(py-cozmo.protocol_encoder.ObjectConnect method), 119</i>
<code>is_from_robot()</code> <i>(py-cozmo.protocol_encoder.FallingStopped method), 108</i>	<code>is_from_robot()</code> <i>(py-cozmo.protocol_encoder.ObjectConnectionState method), 120</i>
<code>is_from_robot()</code> <i>(py-cozmo.protocol_encoder.FirmwareSignature method), 109</i>	<code>is_from_robot()</code> <i>(py-cozmo.protocol_encoder.ObjectMoved method), 120</i>
<code>is_from_robot()</code> <i>(py-cozmo.protocol_encoder.FirmwareUpdate method), 109</i>	<code>is_from_robot()</code> <i>(py-cozmo.protocol_encoder.ObjectPowerLevel method), 121</i>
<code>is_from_robot()</code> <i>(py-cozmo.protocol_encoder.FirmwareUpdateResult method), 109</i>	<code>is_from_robot()</code> <i>(py-cozmo.protocol_encoder.ObjectStoppedMoving method), 121</i>
<code>is_from_robot()</code> <i>(py-cozmo.protocol_encoder.HardwareInfo method), 110</i>	<code>is_from_robot()</code> <i>(py-cozmo.protocol_encoder.ObjectTapFiltered method), 121</i>
<code>is_from_robot()</code> <i>(py-cozmo.protocol_encoder.ImageChunk method), 110</i>	<code>is_from_robot()</code> <i>(py-cozmo.protocol_encoder.ObjectTapped method), 122</i>
<code>is_from_robot()</code> <i>(py-cozmo.protocol_encoder.ImageImuData method), 111</i>	<code>is_from_robot()</code> <i>(py-cozmo.protocol_encoder.ObjectUpAxisChanged method), 124</i>
<code>is_from_robot()</code> <i>(py-cozmo.protocol_encoder.Keyframe method), 112</i>	<code>is_from_robot()</code> <i>(py-cozmo.protocol_encoder.OutputAudio method), 124</i>
<code>is_from_robot()</code> <i>(py-cozmo.protocol_encoder.LightStateCenter method), 113</i>	<code>is_from_robot()</code> <i>(py-cozmo.protocol_encoder.OutputSilence method), 124</i>
<code>is_from_robot()</code> <i>(py-cozmo.protocol_encoder.LightStateSide method), 114</i>	<code>is_from_robot()</code> <i>(py-cozmo.protocol_encoder.PathFollowingEvent method), 125</i>
<code>is_from_robot()</code> <i>(py-cozmo.protocol_encoder.MotorCalibration method), 114</i>	<code>is_from_robot()</code> <i>(pycozmo.protocol_encoder.Ping method), 126</i>
<code>is_from_robot()</code> <i>(py-cozmo.protocol_encoder.MoveHead method), 115</i>	<code>is_from_robot()</code> <i>(py-cozmo.protocol_encoder.RecordHeading method), 126</i>
<code>is_from_robot()</code> <i>(py-cozmo.protocol_encoder.MoveLift method), 115</i>	<code>is_from_robot()</code> <i>(py-cozmo.protocol_encoder.RobotDelocalized method), 126</i>
<code>is_from_robot()</code> <i>(py-cozmo.protocol_encoder.NvStorageOp method), 117</i>	<code>is_from_robot()</code> <i>(py-cozmo.protocol_encoder.RobotPoked method), 127</i>
<code>is_from_robot()</code> <i>(py-cozmo.protocol_encoder.NvStorageOpResult method), 118</i>	<code>is_from_robot()</code> <i>(py-cozmo.protocol_encoder.RobotState method), 128</i>
<code>is_from_robot()</code> <i>(py-cozmo.protocol_encoder.ObjectAccel method), 118</i>	<code>is_from_robot()</code> <i>(py-cozmo.protocol_encoder.SetAccessoryDiscovery method), 128</i>
	<code>is_from_robot()</code> <i>(py-</i>



<code>cozmo.protocol_encoder.SetCameraParams</code> (method), 129	<code>IS_ON_CHARGER</code> ( <code>pycozmo.robot.RobotStatusFlag</code> attribute), 141
<code>is_from_robot()</code> ( <code>pycozmo.protocol_encoder.SetHeadAngle</code> method), 129	<code>is_oob()</code> ( <code>pycozmo.protocol_base.Packet</code> method), 91
<code>is_from_robot()</code> ( <code>pycozmo.protocol_encoder.SetHeadLight</code> method), 130	<code>is_oob()</code> ( <code>pycozmo.protocol_base.UnknownCommand</code> method), 92
<code>is_from_robot()</code> ( <code>pycozmo.protocol_encoder.SetLiftHeight</code> method), 130	<code>is_oob()</code> ( <code>pycozmo.protocol_base.UnknownEvent</code> method), 92
<code>is_from_robot()</code> ( <code>pycozmo.protocol_encoder.SetOrigin</code> method), 131	<code>is_oob()</code> ( <code>pycozmo.protocol_base.UnknownPacket</code> method), 92
<code>is_from_robot()</code> ( <code>pycozmo.protocol_encoder.SetRobotVolume</code> method), 131	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.AbortAnimation</code> method), 95
<code>is_from_robot()</code> ( <code>pycozmo.protocol_encoder.ShutdownRobot</code> method), 132	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.AcknowledgeAction</code> method), 95
<code>is_from_robot()</code> ( <code>pycozmo.protocol_encoder.StartAnimation</code> method), 132	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.AnimationEnded</code> method), 97
<code>is_from_robot()</code> ( <code>pycozmo.protocol_encoder.StartMotorCalibration</code> method), 132	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.AnimationStarted</code> method), 98
<code>is_from_robot()</code> ( <code>pycozmo.protocol_encoder.StopAllMotors</code> method), 133	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.AnimationState</code> method), 98
<code>is_from_robot()</code> ( <code>pycozmo.protocol_encoder.StreamObjectAccel</code> method), 133	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.AnimBackpackLights</code> method), 96
<code>is_from_robot()</code> ( <code>pycozmo.protocol_encoder.SyncTime</code> method), 133	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.AnimBody</code> method), 96
<code>is_from_robot()</code> ( <code>pycozmo.protocol_encoder.TrimPath</code> method), 134	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.AnimHead</code> method), 97
<code>is_from_robot()</code> ( <code>pycozmo.protocol_encoder.TurnInPlace</code> method), 134	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.AnimLift</code> method), 97
<code>is_from_robot()</code> ( <code>pycozmo.protocol_encoder.TurnInPlaceAtSpeed</code> method), 135	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.AppendPathSegArc</code> method), 99
<code>is_from_robot()</code> ( <code>pycozmo.protocol_encoder.TurnToRecordedHeading</code> method), 135	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.AppendPathSegLine</code> method), 100
<code>is_from_robot()</code> ( <code>pycozmo.protocol_encoder.WifiOff</code> method), 136	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.AppendPathSegPointTurn</code> method), 100
<code>is_full()</code> ( <code>pycozmo.window.SendWindow</code> method), 148	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.BodyInfo</code> method), 101
<code>IS_MOVING</code> ( <code>pycozmo.robot.RobotStatusFlag</code> attribute), 141	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.ButtonPressed</code> method), 101
	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.ClearPath</code> method), 102
	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.Connect</code> method), 102
	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.CubeId</code> method), 103
	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.CubeLights</code> method), 103
	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.DebugData</code> method), 103
	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.Disconnect</code> method), 104
	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.DisplayImage</code> method), 104
	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.DriveWheels</code> method), 105
	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.Enable</code> method),

105	method), 120
<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.EnableAnimationState</code> method), 105	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.ObjectPowerLevel</code> method), 121
<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.EnableCamera</code> method), 106	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.ObjectStoppedMoving</code> method), 121
<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.EnableColorImages</code> method), 106	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.ObjectTapFiltered</code> method), 121
<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.EnableStopOnCliff</code> method), 107	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.ObjectTapped</code> method), 122
<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.EndAnimation</code> method), 107	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.ObjectUpAxisChanged</code> method), 124
<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.ExecutePath</code> method), 107	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.OutputAudio</code> method), 124
<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.FallingStarted</code> method), 108	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.OutputSilence</code> method), 124
<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.FallingStopped</code> method), 108	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.PathFollowingEvent</code> method), 125
<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.FirmwareSignature</code> method), 109	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.Ping</code> method), 126
<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.FirmwareUpdate</code> method), 109	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.RecordHeading</code> method), 126
<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.FirmwareUpdateResult</code> method), 109	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.RobotDelocalized</code> method), 126
<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.HardwareInfo</code> method), 110	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.RobotPoked</code> method), 127
<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.ImageChunk</code> method), 111	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.RobotState</code> method), 128
<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.ImageImuData</code> method), 111	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.SetAccessoryDiscovery</code> method), 128
<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.Keyframe</code> method), 112	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.SetCameraParams</code> method), 129
<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.LightStateCenter</code> method), 113	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.SetHeadAngle</code> method), 129
<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.LightStateSide</code> method), 114	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.SetHeadLight</code> method), 130
<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.MotorCalibration</code> method), 114	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.SetLiftHeight</code> method), 130
<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.MoveHead</code> method), 115	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.SetOrigin</code> method), 131
<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.MoveLift</code> method), 115	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.SetRobotVolume</code> method), 131
<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.NvStorageOp</code> method), 117	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.ShutdownRobot</code> method), 132
<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.NvStorageOpResult</code> method), 118	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.StartAnimation</code> method), 132
<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.ObjectAccel</code> method), 118	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.StartMotorCalibration</code> method), 132
<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.ObjectAvailable</code> method), 119	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.StopAllMotors</code> method), 133
<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.ObjectConnect</code> method), 119	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.StreamObjectAccel</code> method), 133
<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.ObjectConnectionState</code> method), 120	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.SyncTime</code> method), 133
<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.ObjectMoved</code> method), 120	<code>is_oob()</code> ( <code>pycozmo.protocol_encoder.TrimPath</code> method), 133

method), 134  
 is\_oob() (pycozmo.protocol\_encoder.TurnInPlace method), 134  
 is\_oob() (pycozmo.protocol\_encoder.TurnInPlaceAtSpeed method), 135  
 is\_oob() (pycozmo.protocol\_encoder.TurnToRecordedHeading method), 135  
 is\_oob() (pycozmo.protocol\_encoder.WifiOff method), 136  
 is\_out\_of\_order() (pycozmo.window.ReceiveWindow method), 148  
 is\_out\_of\_order() (pycozmo.window.SendWindow method), 148  
 IS\_PATHING (pycozmo.robot.RobotStatusFlag attribute), 141  
 IS\_PICKED\_UP (pycozmo.robot.RobotStatusFlag attribute), 141  
 IS\_PICKING\_OR\_PLACING (pycozmo.robot.RobotStatusFlag attribute), 141  
 is\_valid (pycozmo.util.Pose attribute), 146  
 is\_valid\_seq() (pycozmo.window.BaseWindow method), 147  
 is\_valid\_seq() (pycozmo.window.ReceiveWindow method), 148  
 is\_valid\_seq() (pycozmo.window.SendWindow method), 148  
 isAlive() (pycozmo.conn.Connection method), 74  
 isAlive() (pycozmo.conn.ReceiveThread method), 71  
 isAlive() (pycozmo.conn.SendThread method), 73  
 isDaemon() (pycozmo.conn.Connection method), 75  
 isDaemon() (pycozmo.conn.ReceiveThread method), 71  
 isDaemon() (pycozmo.conn.SendThread method), 73

## J

join() (pycozmo.conn.Connection method), 75  
 join() (pycozmo.conn.ReceiveThread method), 72  
 join() (pycozmo.conn.SendThread method), 73  
 JPEGColor (pycozmo.protocol\_encoder.ImageEncoding attribute), 111  
 JPEGColorHalfWidth (pycozmo.protocol\_encoder.ImageEncoding attribute), 111  
 JPEGGray (pycozmo.protocol\_encoder.ImageEncoding attribute), 111  
 JPEGMinimizedColor (pycozmo.protocol\_encoder.ImageEncoding attribute), 111  
 JPEGMinimizedGray (pycozmo.protocol\_encoder.ImageEncoding attribute), 111

## K

Keyframe (class in pycozmo.protocol\_ast), 90  
 Keyframe (class in pycozmo.protocol\_encoder), 112  
 KEYFRAME (pycozmo.protocol\_ast.PacketType attribute), 87  
 keyframe\_to\_im() (pycozmo.anim.PreprocessedClip class method), 59

## L

language (pycozmo.audiokinetic.soundbanksinfo.SoundBankInfo attribute), 43  
 last (pycozmo.protocol\_encoder.Ping attribute), 126  
 last\_change (pycozmo.emotions.EmotionType attribute), 76  
 last\_value (pycozmo.emotions.EmotionType attribute), 76  
 LE\_BL\_v16 (pycozmo.protocol\_encoder.BodyColor attribute), 101  
 left\_xyz (pycozmo.util.Matrix44 attribute), 145  
 length (pycozmo.audiokinetic.soundbank.File attribute), 41  
 length (pycozmo.audiokinetic.soundbank.SFX attribute), 42  
 length (pycozmo.protocol\_encoder.NvStorageOp attribute), 117  
 length (pycozmo.protocol\_encoder.NvStorageOpResult attribute), 118  
 level (pycozmo.protocol\_encoder.DebugData attribute), 103  
 level (pycozmo.protocol\_encoder.SetRobotVolume attribute), 131  
 lids (pycozmo.procedural\_face.ProceduralEye attribute), 85  
 lift (pycozmo.protocol\_encoder.StartMotorCalibration attribute), 132  
 LIFT\_ARM\_LENGTH (in module pycozmo.robot), 140  
 lift\_height\_mm (pycozmo.protocol\_encoder.RobotState attribute), 128  
 LIFT\_IN\_POS (pycozmo.robot.RobotStatusFlag attribute), 141  
 LIFT\_PIVOT\_HEIGHT (in module pycozmo.robot), 140  
 LiftPosition (class in pycozmo.robot), 141  
 LightState (class in pycozmo.protocol\_encoder), 113  
 LightStateCenter (class in pycozmo.protocol\_encoder), 113  
 LightStateSide (class in pycozmo.protocol\_encoder), 113  
 line\_2\_number (pycozmo.protocol\_encoder.ImageImuData attribute), 111

`load()` (*pycozmo.audiokinetic.soundbank.SoundBankReader* method), 42  
`load_activities()` (in module *pycozmo.activity*), 59  
`load_animation_groups()` (in module *pycozmo.anim*), 60  
`load_anims()` (*pycozmo.client.Client* method), 70  
`load_backpack_light_patterns()` (in module *pycozmo.anim*), 60  
`load_behaviors()` (in module *pycozmo.behavior*), 67  
`load_cube_animation_groups()` (in module *pycozmo.anim*), 60  
`load_emotion_events()` (in module *pycozmo.emotions*), 77  
`load_emotion_types()` (in module *pycozmo.emotions*), 76  
`load_file()` (*pycozmo.audiokinetic.soundbank.SoundBankReader* method), 42  
`load_reaction_trigger_behavior_map()` (in module *pycozmo.behavior*), 67  
`load_soundbanksinfo()` (in module *pycozmo.audiokinetic.soundbanksinfo*), 43  
`load_wav()` (in module *pycozmo.audio*), 66  
`location` (*pycozmo.audiokinetic.soundbank.SFX* attribute), 42  
`log_stats()` (*pycozmo.conn.Connection* method), 75  
`lower_inner_radius_x` (*pycozmo.procedural\_face.ProceduralEye* attribute), 85  
`lower_inner_radius_y` (*pycozmo.procedural\_face.ProceduralEye* attribute), 85  
`lower_outer_radius_x` (*pycozmo.procedural\_face.ProceduralEye* attribute), 85  
`lower_outer_radius_y` (*pycozmo.procedural\_face.ProceduralEye* attribute), 85  
`lwheel_accel_mmmps2` (*pycozmo.protocol\_encoder.DriveWheels* attribute), 105  
`lwheel_speed_mmmps` (*pycozmo.protocol\_encoder.DriveWheels* attribute), 105  
`lwheel_speed_mmmps` (*pycozmo.protocol\_encoder.RobotState* attribute), 128  
  
**M**  
`m00` (*pycozmo.util.Matrix44* attribute), 145  
`m01` (*pycozmo.util.Matrix44* attribute), 145  
`m02` (*pycozmo.util.Matrix44* attribute), 145  
`m03` (*pycozmo.util.Matrix44* attribute), 145  
`m10` (*pycozmo.util.Matrix44* attribute), 145  
`m11` (*pycozmo.util.Matrix44* attribute), 145  
`m12` (*pycozmo.util.Matrix44* attribute), 145  
`m13` (*pycozmo.util.Matrix44* attribute), 145  
`m20` (*pycozmo.util.Matrix44* attribute), 145  
`m21` (*pycozmo.util.Matrix44* attribute), 145  
`m22` (*pycozmo.util.Matrix44* attribute), 145  
`m23` (*pycozmo.util.Matrix44* attribute), 145  
`m30` (*pycozmo.util.Matrix44* attribute), 145  
`m31` (*pycozmo.util.Matrix44* attribute), 145  
`m32` (*pycozmo.util.Matrix44* attribute), 145  
`m33` (*pycozmo.util.Matrix44* attribute), 145  
`Matrix44` (class in *pycozmo.util*), 144  
`max` (*pycozmo.emotions.EmotionType* attribute), 76  
`MAX_HEAD_ANGLE` (in module *pycozmo.robot*), 140  
`MAX_LIFT_ANGLE` (in module *pycozmo.robot*), 140  
`MAX_LIFT_HEIGHT` (in module *pycozmo.robot*), 140  
`max_revolutions_per_sec` (*pycozmo.protocol\_encoder.SetHeadAngle* attribute), 129  
`max_speed_rad_per_sec` (*pycozmo.protocol\_encoder.SetLiftHeight* attribute), 130  
`MAX_WHEEL_SPEED` (in module *pycozmo.robot*), 140  
`MAXLEN` (*pycozmo.anim\_controller.AnimationQueue* attribute), 61  
`member_probabilities` (*pycozmo.anim.AnimationGroup* attribute), 60  
`members` (*pycozmo.anim.AnimationGroup* attribute), 60  
`min` (*pycozmo.emotions.EmotionType* attribute), 76  
`MIN_HEAD_ANGLE` (in module *pycozmo.robot*), 140  
`MIN_LIFT_ANGLE` (in module *pycozmo.robot*), 140  
`MIN_LIFT_HEIGHT` (in module *pycozmo.robot*), 140  
`minicolor_to_jpeg()` (in module *pycozmo.camera*), 69  
`minigray_to_jpeg()` (in module *pycozmo.camera*), 69  
`missed_packets` (*pycozmo.protocol\_encoder.ObjectPowerLevel* attribute), 121  
`mm` (*pycozmo.util.Distance* attribute), 143  
`mmmps` (*pycozmo.util.Speed* attribute), 144  
`mood` (*pycozmo.anim.AnimationGroupMember* attribute), 60  
`MOTOR_HEAD` (*pycozmo.protocol\_encoder.MotorID* attribute), 114  
`motor_id` (*pycozmo.protocol\_encoder.MotorCalibration* attribute), 114  
`MOTOR_LEFT_WHEEL` (*pycozmo.protocol\_encoder.MotorID* attribute), 114  
`MOTOR_LIFT` (*pycozmo.protocol\_encoder.MotorID* attribute), 114

MOTOR_RIGHT_WHEEL (pycozmo.protocol_encoder.MotorID attribute), 114	NV_CORRUPT (pycozmo.protocol_encoder.NvResult attribute), 117
MotorCalibration (class in pycozmo.protocol_encoder), 114	NV_ERROR (pycozmo.protocol_encoder.NvResult attribute), 117
MotorID (class in pycozmo.protocol_encoder), 114	NV_LOOP (pycozmo.protocol_encoder.NvResult attribute), 117
move_head() (pycozmo.client.Client method), 70	NV_MORE (pycozmo.protocol_encoder.NvResult attribute), 117
move_lift() (pycozmo.client.Client method), 70	NV_NO_DO (pycozmo.protocol_encoder.NvResult attribute), 117
MoveHead (class in pycozmo.protocol_encoder), 114	NV_NO_MEM (pycozmo.protocol_encoder.NvResult attribute), 117
MoveLift (class in pycozmo.protocol_encoder), 115	NV_NO_ROOM (pycozmo.protocol_encoder.NvResult attribute), 117
<b>N</b>	NV_NOT_FOUND (pycozmo.protocol_encoder.NvResult attribute), 117
name (pycozmo.anim.AnimationGroupMember attribute), 60	NV_OKAY (pycozmo.protocol_encoder.NvResult attribute), 117
name (pycozmo.audiokinetic.soundbank.Event attribute), 41	NV_SCHEDULED (pycozmo.protocol_encoder.NvResult attribute), 117
name (pycozmo.audiokinetic.soundbank.SFX attribute), 42	NV_TIMEOUT (pycozmo.protocol_encoder.NvResult attribute), 117
name (pycozmo.audiokinetic.soundbank.SoundBank attribute), 42	NV_UNKNOWN_4 (pycozmo.protocol_encoder.NvResult attribute), 117
name (pycozmo.audiokinetic.soundbanksinfo.EventInfo attribute), 43	NV_UNKNOWN_5 (pycozmo.protocol_encoder.NvResult attribute), 117
name (pycozmo.audiokinetic.soundbanksinfo.FileInfo attribute), 43	NV_UNKNOWN_6 (pycozmo.protocol_encoder.NvResult attribute), 117
name (pycozmo.audiokinetic.soundbanksinfo.SoundBankInfo attribute), 43	NV_UNKNOWN_7 (pycozmo.protocol_encoder.NvResult attribute), 117
name (pycozmo.behavior.ReactionTrigger attribute), 67	NV_UNKNOWN_8 (pycozmo.protocol_encoder.NvResult attribute), 117
name (pycozmo.conn.Connection attribute), 75	NVEntry_BirthCertificate (pycozmo.protocol_encoder.NvEntryTag attribute), 115
name (pycozmo.conn.ReceiveThread attribute), 72	NVEntry_CalibImage1 (pycozmo.protocol_encoder.NvEntryTag attribute), 115
name (pycozmo.conn.SendThread attribute), 73	NVEntry_CalibImage2 (pycozmo.protocol_encoder.NvEntryTag attribute), 115
name (pycozmo.emotions.EmotionEvent attribute), 76	NVEntry_CalibImage3 (pycozmo.protocol_encoder.NvEntryTag attribute), 115
name (pycozmo.emotions.EmotionType attribute), 76	NVEntry_CalibImage4 (pycozmo.protocol_encoder.NvEntryTag attribute), 115
name_id (pycozmo.protocol_encoder.DebugData attribute), 103	NVEntry_CalibImage5 (pycozmo.protocol_encoder.NvEntryTag attribute), 115
Neutral (class in pycozmo.expressions.expressions), 44	NVEntry_CalibImage6 (pycozmo.protocol_encoder.NvEntryTag attribute), 115
NoneImageEncoding (pycozmo.protocol_encoder.ImageEncoding attribute), 111	NVEntry_CalibMetaInfo (pycozmo.protocol_encoder.NvEntryTag attribute), 115
NoSpace, 80	
num_anim_bytes_played (pycozmo.protocol_encoder.AnimationState attribute), 98	
num_audio_frames_played (pycozmo.protocol_encoder.AnimationState attribute), 98	
num_taps (pycozmo.protocol_encoder.ObjectTapped attribute), 122	
NumAxes (pycozmo.protocol_encoder.UpAxis attribute), 136	
NV_BAD_ARGS (pycozmo.protocol_encoder.NvResult attribute), 116	
NV_BUSY (pycozmo.protocol_encoder.NvResult attribute), 117	



<i>cozmo.protocol_encoder.NvEntryTag</i> <i>tribute</i> ), 115	at-	<i>cozmo.protocol_encoder.NvEntryTag</i> <i>tribute</i> ), 116	at-
NVEntry_CalibPose <i>cozmo.protocol_encoder.NvEntryTag</i> <i>tribute</i> ), 115	(py-at-	NVEntry_NEXT_SLOT <i>cozmo.protocol_encoder.NvEntryTag</i> <i>tribute</i> ), 116	(py-at-
NVEntry_CameraCalib <i>cozmo.protocol_encoder.NvEntryTag</i> <i>tribute</i> ), 115	(py-at-	NVEntry_NurtureGameData <i>cozmo.protocol_encoder.NvEntryTag</i> <i>tribute</i> ), 116	(py-at-
NVEntry_CliffValOnDrop <i>cozmo.protocol_encoder.NvEntryTag</i> <i>tribute</i> ), 115	(py-at-	NVEntry_ObservedCubePose <i>cozmo.protocol_encoder.NvEntryTag</i> <i>tribute</i> ), 116	(py-at-
NVEntry_CliffValOnGround <i>cozmo.protocol_encoder.NvEntryTag</i> <i>tribute</i> ), 115	(py-at-	NVEntry_OnboardingData <i>cozmo.protocol_encoder.NvEntryTag</i> <i>tribute</i> ), 116	(py-at-
NVEntry_FaceAlbumData <i>cozmo.protocol_encoder.NvEntryTag</i> <i>tribute</i> ), 116	(py-at-	NVEntry_PlaypenTestResults <i>cozmo.protocol_encoder.NvEntryTag</i> <i>tribute</i> ), 116	(py-at-
NVEntry_FaceEnrollData <i>cozmo.protocol_encoder.NvEntryTag</i> <i>tribute</i> ), 116	(py-at-	NVEntry_PrePlaypenCentroids <i>cozmo.protocol_encoder.NvEntryTag</i> <i>tribute</i> ), 116	(py-at-
NVEntry_FACTORY_RESERVED1 <i>cozmo.protocol_encoder.NvEntryTag</i> <i>tribute</i> ), 116	(py-at-	NVEntry_PrePlaypenResults <i>cozmo.protocol_encoder.NvEntryTag</i> <i>tribute</i> ), 116	(py-at-
NVEntry_FACTORY_RESERVED2 <i>cozmo.protocol_encoder.NvEntryTag</i> <i>tribute</i> ), 116	(py-at-	NVEntry_SavedCubeIDs <i>cozmo.protocol_encoder.NvEntryTag</i> <i>tribute</i> ), 116	(py-at-
NVEntry_FactoryBaseTag <i>cozmo.protocol_encoder.NvEntryTag</i> <i>tribute</i> ), 116	(py-at-	NVEntry_ToolCodeImageLeft <i>cozmo.protocol_encoder.NvEntryTag</i> <i>tribute</i> ), 116	(py-at-
NVEntry_FactoryBaseTagWithBCOffset ( <i>pycozmo.protocol_encoder.NvEntryTag</i> <i>tribute</i> ), 116	at-	NVEntry_ToolCodeImageRight <i>cozmo.protocol_encoder.NvEntryTag</i> <i>tribute</i> ), 116	(py-at-
NVEntry_FactoryLock <i>cozmo.protocol_encoder.NvEntryTag</i> <i>tribute</i> ), 116	(py-at-	NVEntry_ToolCodeInfo <i>cozmo.protocol_encoder.NvEntryTag</i> <i>tribute</i> ), 116	(py-at-
NVEntry_GameSkillLevels <i>cozmo.protocol_encoder.NvEntryTag</i> <i>tribute</i> ), 116	(py-at-	NVEntry_VersionMagic <i>cozmo.protocol_encoder.NvEntryTag</i> <i>tribute</i> ), 116	(py-at-
NVEntry_GameUnlocks <i>cozmo.protocol_encoder.NvEntryTag</i> <i>tribute</i> ), 116	(py-at-	NvEntryTag (class in <i>pycozmo.protocol_encoder</i> ), 115	
NVEntry_IMUAverages <i>cozmo.protocol_encoder.NvEntryTag</i> <i>tribute</i> ), 116	(py-at-	NVOP_ERASE ( <i>pycozmo.protocol_encoder.NvOperation</i> <i>tribute</i> ), 116	
NVEntry_IMUInfo <i>cozmo.protocol_encoder.NvEntryTag</i> <i>tribute</i> ), 116	(py-at-	NVOP_READ ( <i>pycozmo.protocol_encoder.NvOperation</i> <i>tribute</i> ), 116	
NVEntry_Invalid <i>cozmo.protocol_encoder.NvEntryTag</i> <i>tribute</i> ), 116	(py-at-	NVOP_WIPEALL ( <i>pycozmo.protocol_encoder.NvOperation</i> <i>tribute</i> ), 116	
NVEntry_InventoryData <i>cozmo.protocol_encoder.NvEntryTag</i> <i>tribute</i> ), 116	(py-at-	NVOP_WRITE ( <i>pycozmo.protocol_encoder.NvOperation</i> <i>tribute</i> ), 116	
NVEntry_LabAssignments	(py-	NvOperation (class in <i>pycozmo.protocol_encoder</i> ), 116	
		NvResult (class in <i>pycozmo.protocol_encoder</i> ), 116	
		NvStorageOp (class in <i>pycozmo.protocol_encoder</i> ), 117	
		NvStorageOpResult (class in <i>pycozmo.protocol_encoder</i> ), 117	

## O

- Object (class in `pycozmo.object`), 83
- object\_id (`pycozmo.protocol_encoder.CubeId` attribute), 103
- object\_id (`pycozmo.protocol_encoder.ObjectAccel` attribute), 118
- object\_id (`pycozmo.protocol_encoder.ObjectConnectionState` attribute), 120
- object\_id (`pycozmo.protocol_encoder.ObjectMoved` attribute), 120
- object\_id (`pycozmo.protocol_encoder.ObjectPowerLevel` attribute), 121
- object\_id (`pycozmo.protocol_encoder.ObjectStoppedMoving` attribute), 121
- object\_id (`pycozmo.protocol_encoder.ObjectTapFiltered` attribute), 122
- object\_id (`pycozmo.protocol_encoder.ObjectTapped` attribute), 122
- object\_id (`pycozmo.protocol_encoder.ObjectUpAxisChanged` attribute), 124
- object\_id (`pycozmo.protocol_encoder.StreamObjectAccel` attribute), 133
- object\_path (`pycozmo.audiokinetic.soundbanksinfo.EventInfo` attribute), 43
- object\_path (`pycozmo.audiokinetic.soundbanksinfo.SoundBankInfo` attribute), 43
- object\_type (`pycozmo.protocol_encoder.ObjectAvailable` attribute), 119
- object\_type (`pycozmo.protocol_encoder.ObjectConnectionState` attribute), 120
- ObjectAccel (class in `pycozmo.protocol_encoder`), 118
- ObjectAvailable (class in `pycozmo.protocol_encoder`), 118
- ObjectConnect (class in `pycozmo.protocol_encoder`), 119
- ObjectConnectionState (class in `pycozmo.protocol_encoder`), 119
- ObjectMoved (class in `pycozmo.protocol_encoder`), 120
- ObjectPowerLevel (class in `pycozmo.protocol_encoder`), 120
- ObjectStoppedMoving (class in `pycozmo.protocol_encoder`), 121
- ObjectTapFiltered (class in `pycozmo.protocol_encoder`), 121
- ObjectTapped (class in `pycozmo.protocol_encoder`), 122
- ObjectType (class in `pycozmo.protocol_encoder`), 122
- ObjectUpAxisChanged (class in `pycozmo.protocol_encoder`), 123
- objs (`pycozmo.audiokinetic.soundbank.SoundBank` attribute), 42
- off (in module `pycozmo.lights`), 83
- Off (`pycozmo.protocol_encoder.ImageSendMode` attribute), 112
- off\_color (`pycozmo.protocol_encoder.LightState` attribute), 113
- off\_frames (`pycozmo.protocol_encoder.LightState` attribute), 113
- off\_light (in module `pycozmo.lights`), 83
- offset (`pycozmo.audiokinetic.soundbank.File` attribute), 41
- offset (`pycozmo.expressions.expressions.Amazement` attribute), 57
- offset (`pycozmo.expressions.expressions.Anger` attribute), 45
- offset (`pycozmo.expressions.expressions.Annoyance` attribute), 53
- offset (`pycozmo.expressions.expressions.Asleep` attribute), 56
- offset (`pycozmo.expressions.expressions.Boredom` attribute), 55
- offset (`pycozmo.expressions.expressions.Confusion` attribute), 57
- offset (`pycozmo.expressions.expressions.Despair` attribute), 49
- offset (`pycozmo.expressions.expressions.Disappointment` attribute), 51
- offset (`pycozmo.expressions.expressions.Disgust` attribute), 47
- offset (`pycozmo.expressions.expressions.Embarrassment` attribute), 51
- offset (`pycozmo.expressions.expressions.Excitement` attribute), 58
- offset (`pycozmo.expressions.expressions.Fear` attribute), 48
- offset (`pycozmo.expressions.expressions.Fury` attribute), 53
- offset (`pycozmo.expressions.expressions.Guilt` attribute), 50
- offset (`pycozmo.expressions.expressions.Happiness` attribute), 46
- offset (`pycozmo.expressions.expressions.Horror` attribute), 52
- offset (`pycozmo.expressions.expressions.Neutral` attribute), 45
- offset (`pycozmo.expressions.expressions.Pleading` attribute), 48
- offset (`pycozmo.expressions.expressions.Rejection` attribute), 54
- offset (`pycozmo.expressions.expressions.Sadness` attribute), 46
- offset (`pycozmo.expressions.expressions.Skepticism` attribute), 52
- offset (`pycozmo.expressions.expressions.Surprise` attribute), 47
- offset (`pycozmo.expressions.expressions.Suspicion` attribute), 47

tribute), 54

offset (*pycozmo.expressions.expressions.Tiredness attribute*), 56

offset (*pycozmo.expressions.expressions.Vulnerability attribute*), 49

offset (*pycozmo.procedural\_face.ProceduralEye attribute*), 85

offset (*pycozmo.procedural\_face.ProceduralFace attribute*), 85

offset (*pycozmo.procedural\_face.ProceduralLid attribute*), 84

offset (*pycozmo.protocol\_encoder.LightState attribute*), 113

on\_behavior\_done() (*pycozmo.brain.Brain method*), 68

on\_camera\_image() (*pycozmo.brain.Brain method*), 68

on\_cliff\_detected() (*pycozmo.brain.Brain method*), 68

on\_color (*pycozmo.protocol\_encoder.LightState attribute*), 113

on\_frames (*pycozmo.protocol\_encoder.LightState attribute*), 113

on\_robot\_falling\_change() (*pycozmo.brain.Brain method*), 68

on\_robot\_on\_charger\_change() (*pycozmo.brain.Brain method*), 68

on\_robot\_orientation\_change() (*pycozmo.brain.Brain method*), 68

on\_robot\_picked\_up\_change() (*pycozmo.brain.Brain method*), 68

op (*pycozmo.protocol\_encoder.NvStorageOp attribute*), 117

op (*pycozmo.protocol\_encoder.NvStorageOpResult attribute*), 118

origin\_id (*pycozmo.util.Pose attribute*), 146

OutputAudio (*class in pycozmo.protocol\_encoder*), 124

OutputSilence (*class in pycozmo.protocol\_encoder*), 124

## P

Packet (*class in pycozmo.protocol\_ast*), 90

Packet (*class in pycozmo.protocol\_base*), 91

PacketType (*class in pycozmo.protocol\_ast*), 87

params (*pycozmo.expressions.expressions.Amazement attribute*), 57

params (*pycozmo.expressions.expressions.Anger attribute*), 45

params (*pycozmo.expressions.expressions.Annoyance attribute*), 53

params (*pycozmo.expressions.expressions.Asleep attribute*), 56

params (*pycozmo.expressions.expressions.Boredom attribute*), 55

params (*pycozmo.expressions.expressions.Confusion attribute*), 57

params (*pycozmo.expressions.expressions.Despair attribute*), 49

params (*pycozmo.expressions.expressions.Disappointment attribute*), 51

params (*pycozmo.expressions.expressions.Disgust attribute*), 47

params (*pycozmo.expressions.expressions.Embarrassment attribute*), 51

params (*pycozmo.expressions.expressions.Excitement attribute*), 58

params (*pycozmo.expressions.expressions.Fear attribute*), 48

params (*pycozmo.expressions.expressions.Fury attribute*), 53

params (*pycozmo.expressions.expressions.Guilt attribute*), 50

params (*pycozmo.expressions.expressions.Happiness attribute*), 46

params (*pycozmo.expressions.expressions.Horror attribute*), 52

params (*pycozmo.expressions.expressions.Neutral attribute*), 45

params (*pycozmo.expressions.expressions.Pleading attribute*), 48

params (*pycozmo.expressions.expressions.Rejection attribute*), 54

params (*pycozmo.expressions.expressions.Sadness attribute*), 46

params (*pycozmo.expressions.expressions.Skepticism attribute*), 52

params (*pycozmo.expressions.expressions.Surprise attribute*), 47

params (*pycozmo.expressions.expressions.Suspicion attribute*), 54

params (*pycozmo.expressions.expressions.Tiredness attribute*), 56

params (*pycozmo.expressions.expressions.Vulnerability attribute*), 49

params (*pycozmo.procedural\_face.ProceduralEye attribute*), 85

params (*pycozmo.procedural\_face.ProceduralFace attribute*), 85

params (*pycozmo.procedural\_face.ProceduralLid attribute*), 84

path (*pycozmo.audiokinetic.soundbanksinfo.FileInfo attribute*), 43

path (*pycozmo.audiokinetic.soundbanksinfo.SoundBankInfo attribute*), 43

PATH\_COMPLETED (*pycozmo.protocol\_encoder.PathEventType attribute*), 56



tribute), 125

PATH\_INTERRUPTED (pycozmo.protocol\_encoder.PathEventType attribute), 125

PATH\_STARTED (pycozmo.protocol\_encoder.PathEventType attribute), 125

PathEventType (class in pycozmo.protocol\_encoder), 125

PathFollowingEvent (class in pycozmo.protocol\_encoder), 125

PathSegmentSpeed (class in pycozmo.protocol\_encoder), 125

Ping (class in pycozmo.protocol\_ast), 90

Ping (class in pycozmo.protocol\_encoder), 125

PING (pycozmo.protocol\_ast.FrameType attribute), 87

PING (pycozmo.protocol\_ast.PacketType attribute), 87

PING\_INTERVAL (pycozmo.conn.Connection attribute), 74

pkts (pycozmo.frame.Frame attribute), 81

Platform\_LARGE (pycozmo.protocol\_encoder.ObjectType attribute), 123

play\_anim() (pycozmo.client.Client method), 70

play\_anim\_frame() (pycozmo.anim\_controller.AnimationController method), 61

play\_anim\_group() (pycozmo.client.Client method), 70

play\_anim\_ppclip() (pycozmo.client.Client method), 70

play\_audio() (pycozmo.anim\_controller.AnimationController method), 61

play\_audio() (pycozmo.client.Client method), 70

Pleading (class in pycozmo.expressions.expressions), 48

pos\_xyz (pycozmo.util.Matrix44 attribute), 145

Pose (class in pycozmo.util), 146

pose\_angle\_rad (pycozmo.protocol\_encoder.RobotState attribute), 128

pose\_frame\_id (pycozmo.protocol\_encoder.RobotState attribute), 128

pose\_frame\_id (pycozmo.protocol\_encoder.SetOrigin attribute), 131

pose\_origin\_id (pycozmo.protocol\_encoder.RobotState attribute), 128

pose\_origin\_id (pycozmo.protocol\_encoder.SetOrigin attribute), 131

pose\_pitch\_rad (pycozmo.protocol\_encoder.RobotState attribute), 128

pose\_x (pycozmo.protocol\_encoder.RobotState attribute), 128

pose\_x (pycozmo.protocol\_encoder.SetOrigin attribute), 131

pose\_y (pycozmo.protocol\_encoder.RobotState attribute), 128

pose\_y (pycozmo.protocol\_encoder.SetOrigin attribute), 131

pose\_z (pycozmo.protocol\_encoder.RobotState attribute), 128

position (pycozmo.util.Pose attribute), 146

post\_event() (pycozmo.conn.Connection method), 75

post\_reaction() (pycozmo.brain.Brain method), 68

prefetch\_size (pycozmo.audiokinetic.soundbanksinfo.FileInfo attribute), 43

PreprocessedClip (class in pycozmo.anim), 59

pressed (pycozmo.protocol\_encoder.ButtonPressed attribute), 101

ProceduralEye (class in pycozmo.procedural\_face), 84

ProceduralFace (class in pycozmo.procedural\_face), 85

ProceduralLid (class in pycozmo.procedural\_face), 84

process\_emotion\_event() (pycozmo.brain.Brain method), 68

process\_reaction() (pycozmo.brain.Brain method), 68

Protocol (class in pycozmo.protocol\_ast), 90

PROTOCOL (in module pycozmo.protocol\_declaration), 93

ProtocolGenerator (class in pycozmo.protocol\_generator), 137

ProxObstacle (pycozmo.protocol\_encoder.ObjectType attribute), 123

put() (pycozmo.window.ReceiveWindow method), 148

put() (pycozmo.window.SendWindow method), 148

put\_anim\_frame() (pycozmo.anim\_controller.AnimationQueue method), 61

put\_audio() (pycozmo.anim\_controller.AnimationQueue method), 61

put\_image() (pycozmo.anim\_controller.AnimationQueue method), 61

pycozmo.activity (module), 58

pycozmo.anim (module), 59

pycozmo.anim\_controller (module), 60

pycozmo.anim\_encoder (module), 61

pycozmo.audio (module), 66

pycozmo.audiokinetic.exception (module), 40

pycozmo.audiokinetic.soundbank (module), 41

pycozmo.audiokinetic.soundbanksinfo (module), 42

pycozmo.audiokinetic.wem (module), 44

pycozmo.behavior (module), 66

pycozmo.brain (module), 67

pycozmo.camera (module), 68

pycozmo.client (module), 69

pycozmo.conn (module), 71

pycozmo.emotions (module), 76

pycozmo.event (module), 77

pycozmo.exception (module), 79

pycozmo.expressions.expressions (module), 44

pycozmo.filter (module), 80

pycozmo.frame (module), 81

pycozmo.image\_encoder (module), 81

pycozmo.lights (module), 82

pycozmo.object (module), 83

pycozmo.procedural\_face (module), 83

pycozmo.protocol\_ast (module), 86

pycozmo.protocol\_base (module), 91

pycozmo.protocol\_declaration (module), 93

pycozmo.protocol\_encoder (module), 93

pycozmo.protocol\_generator (module), 136

pycozmo.protocol\_utils (module), 137

pycozmo.robot (module), 140

pycozmo.robot\_debug (module), 141

pycozmo.run (module), 142

pycozmo.util (module), 142

pycozmo.window (module), 147

PyCozmoConnectionError, 80

PyCozmoException, 80

## Q

q0 (pycozmo.util.Quaternion attribute), 146

q0\_q1\_q2\_q3 (pycozmo.util.Quaternion attribute), 146

q1 (pycozmo.util.Quaternion attribute), 146

q2 (pycozmo.util.Quaternion attribute), 146

q3 (pycozmo.util.Quaternion attribute), 146

QQQVGA (pycozmo.protocol\_encoder.ImageResolution attribute), 112

QQVGA (pycozmo.protocol\_encoder.ImageResolution attribute), 112

QVGA (pycozmo.protocol\_encoder.ImageResolution attribute), 112

Quaternion (class in pycozmo.util), 146

QUXGA (pycozmo.protocol\_encoder.ImageResolution attribute), 112

QVGA (pycozmo.protocol\_encoder.ImageResolution attribute), 112

QXGA (pycozmo.protocol\_encoder.ImageResolution attribute), 112

## R

radians (pycozmo.util.Angle attribute), 143

radius\_mm (pycozmo.protocol\_encoder.AppendPathSegArc attribute), 99

Ramp\_Basic (pycozmo.protocol\_encoder.ObjectType attribute), 123

rate\_x (pycozmo.protocol\_encoder.ImageImuData attribute), 111

rate\_y (pycozmo.protocol\_encoder.ImageImuData attribute), 111

rate\_z (pycozmo.protocol\_encoder.ImageImuData attribute), 111

ratio (pycozmo.robot.LiftPosition attribute), 141

RawGray (pycozmo.protocol\_encoder.ImageEncoding attribute), 111

RawRGB (pycozmo.protocol\_encoder.ImageEncoding attribute), 111

reaction\_thread\_run() (pycozmo.brain.Brain method), 68

ReactionTrigger (class in pycozmo.behavior), 67

read() (pycozmo.protocol\_utils.BinaryReader method), 138

read\_farray() (pycozmo.protocol\_utils.BinaryReader method), 138

read\_object() (pycozmo.protocol\_utils.BinaryReader method), 139

read\_object\_farray() (pycozmo.protocol\_utils.BinaryReader method), 139

read\_object\_varray() (pycozmo.protocol\_utils.BinaryReader method), 139

read\_string() (pycozmo.protocol\_utils.BinaryReader method), 139

read\_string\_farray() (pycozmo.protocol\_utils.BinaryReader method), 139

read\_string\_varray() (pycozmo.protocol\_utils.BinaryReader method), 139

read\_varray() (pycozmo.protocol\_utils.BinaryReader method), 139

ReceiveThread (class in pycozmo.conn), 71

ReceiveWindow (class in pycozmo.window), 147

RecordHeading (class in pycozmo.protocol\_encoder), 126

red (in module pycozmo.lights), 83

red\_light (in module pycozmo.lights), 83

reference\_id (pycozmo.audiokinetic.soundbank.EventAction attribute), 41

Rejection (class in `pycozmo.expressions.expressions`), 54  
 render () (in module `pycozmo.image_encoder`), 82  
 render () (`pycozmo.expressions.expressions.Amazement` method), 57  
 render () (`pycozmo.expressions.expressions.Anger` method), 45  
 render () (`pycozmo.expressions.expressions.Annoyance` method), 53  
 render () (`pycozmo.expressions.expressions.Asleep` method), 56  
 render () (`pycozmo.expressions.expressions.Boredom` method), 55  
 render () (`pycozmo.expressions.expressions.Confusion` method), 57  
 render () (`pycozmo.expressions.expressions.Despair` method), 50  
 render () (`pycozmo.expressions.expressions.Disappointment` method), 51  
 render () (`pycozmo.expressions.expressions.Disgust` method), 47  
 render () (`pycozmo.expressions.expressions.Embarrassment` method), 51  
 render () (`pycozmo.expressions.expressions.Excitement` method), 58  
 render () (`pycozmo.expressions.expressions.Fear` method), 48  
 render () (`pycozmo.expressions.expressions.Fury` method), 53  
 render () (`pycozmo.expressions.expressions.Guilt` method), 50  
 render () (`pycozmo.expressions.expressions.Happiness` method), 46  
 render () (`pycozmo.expressions.expressions.Horror` method), 52  
 render () (`pycozmo.expressions.expressions.Neutral` method), 45  
 render () (`pycozmo.expressions.expressions.Pleading` method), 48  
 render () (`pycozmo.expressions.expressions.Rejection` method), 55  
 render () (`pycozmo.expressions.expressions.Sadness` method), 46  
 render () (`pycozmo.expressions.expressions.Skepticism` method), 52  
 render () (`pycozmo.expressions.expressions.Surprise` method), 47  
 render () (`pycozmo.expressions.expressions.Suspicion` method), 54  
 render () (`pycozmo.expressions.expressions.Tiredness` method), 56  
 render () (`pycozmo.expressions.expressions.Vulnerability` method), 49  
 render () (`pycozmo.procedural_face.ProceduralEye` method), 85  
 render () (`pycozmo.procedural_face.ProceduralFace` method), 85  
 render () (`pycozmo.procedural_face.ProceduralLid` method), 84  
 repetition\_penalty (`pycozmo.emotions.EmotionType` attribute), 76  
 RESERVED (`pycozmo.protocol_encoder.BodyColor` attribute), 101  
 RESET (`pycozmo.protocol_ast.FrameType` attribute), 87  
 reset () (`pycozmo.conn.ReceiveThread` method), 72  
 reset () (`pycozmo.conn.SendThread` method), 73  
 reset () (`pycozmo.window.BaseWindow` method), 147  
 reset () (`pycozmo.window.ReceiveWindow` method), 148  
 reset () (`pycozmo.window.SendWindow` method), 148  
 RESET\_ACK (`pycozmo.protocol_ast.FrameType` attribute), 87  
 RESOLUTIONS (in module `pycozmo.camera`), 69  
 result (`pycozmo.protocol_encoder.NvStorageOpResult` attribute), 118  
 ROBOT (`pycozmo.protocol_ast.FrameType` attribute), 87  
 ROBOT\_ADDR (in module `pycozmo.conn`), 71  
 RobotDelocalized (class in `pycozmo.protocol_encoder`), 126  
 RobotPoked (class in `pycozmo.protocol_encoder`), 127  
 RobotState (class in `pycozmo.protocol_encoder`), 127  
 RobotStatusFlag (class in `pycozmo.robot`), 140  
 RobotStatusFlagNames (in module `pycozmo.robot`), 141  
 rotation (`pycozmo.util.Pose` attribute), 147  
 rotation\_period\_frames (`pycozmo.protocol_encoder.CubeId` attribute), 103  
 rssi (`pycozmo.protocol_encoder.ObjectAvailable` attribute), 119  
 run () (`pycozmo.conn.Connection` method), 75  
 run () (`pycozmo.conn.ReceiveThread` method), 72  
 run () (`pycozmo.conn.SendThread` method), 73  
 RUN\_INTERVAL (`pycozmo.conn.Connection` attribute), 74  
 rwheel\_accel\_mmmps2 (`pycozmo.protocol_encoder.DriveWheels` attribute), 105  
 rwheel\_speed\_mmmps (`pycozmo.protocol_encoder.DriveWheels` attribute), 105  
 rwheel\_speed\_mmmps (`pycozmo.protocol_encoder.RobotState` attribute), 128  
 S  
 Sadness (class in `pycozmo.expressions.expressions`), 45

<code>samples</code> ( <code>pycozmo.protocol_encoder.OutputAudio</code> attribute), 124		<code>cozmo.expressions.expressions.Pleading</code> attribute), 48	
<code>scale_factor_lid_bend</code> ( <code>pycozmo.expressions.expressions.Amazement</code> attribute), 57		<code>scale_factor_lid_bend</code> ( <code>pycozmo.expressions.expressions.Rejection</code> attribute), 55	
<code>scale_factor_lid_bend</code> ( <code>pycozmo.expressions.expressions.Anger</code> attribute), 45		<code>scale_factor_lid_bend</code> ( <code>pycozmo.expressions.expressions.Sadness</code> attribute), 46	
<code>scale_factor_lid_bend</code> ( <code>pycozmo.expressions.expressions.Annoyance</code> attribute), 53		<code>scale_factor_lid_bend</code> ( <code>pycozmo.expressions.expressions.Skepticism</code> attribute), 52	
<code>scale_factor_lid_bend</code> ( <code>pycozmo.expressions.expressions.Asleep</code> attribute), 56		<code>scale_factor_lid_bend</code> ( <code>pycozmo.expressions.expressions.Surprise</code> attribute), 47	
<code>scale_factor_lid_bend</code> ( <code>pycozmo.expressions.expressions.Boredom</code> attribute), 55		<code>scale_factor_lid_bend</code> ( <code>pycozmo.expressions.expressions.Suspicion</code> attribute), 54	
<code>scale_factor_lid_bend</code> ( <code>pycozmo.expressions.expressions.Confusion</code> attribute), 57		<code>scale_factor_lid_bend</code> ( <code>pycozmo.expressions.expressions.Tiredness</code> attribute), 56	
<code>scale_factor_lid_bend</code> ( <code>pycozmo.expressions.expressions.Despair</code> attribute), 50		<code>scale_factor_lid_bend</code> ( <code>pycozmo.expressions.expressions.Vulnerability</code> attribute), 49	
<code>scale_factor_lid_bend</code> ( <code>pycozmo.expressions.expressions.Disappointment</code> attribute), 51		<code>scale_factor_lid_bend</code> ( <code>pycozmo.procedural_face.ProceduralEye</code> attribute), 85	
<code>scale_factor_lid_bend</code> ( <code>pycozmo.expressions.expressions.Disgust</code> attribute), 47		<code>scale_factor_lid_bend</code> ( <code>pycozmo.procedural_face.ProceduralFace</code> attribute), 85	
<code>scale_factor_lid_bend</code> ( <code>pycozmo.expressions.expressions.Embarrassment</code> attribute), 51		<code>scale_factor_lid_bend</code> ( <code>pycozmo.procedural_face.ProceduralLid</code> attribute), 84	
<code>scale_factor_lid_bend</code> ( <code>pycozmo.expressions.expressions.Excitement</code> attribute), 58		<code>scale_factor_lid_height</code> ( <code>pycozmo.expressions.expressions.Amazement</code> attribute), 57	
<code>scale_factor_lid_bend</code> ( <code>pycozmo.expressions.expressions.Fear</code> attribute), 48		<code>scale_factor_lid_height</code> ( <code>pycozmo.expressions.expressions.Anger</code> attribute), 45	
<code>scale_factor_lid_bend</code> ( <code>pycozmo.expressions.expressions.Fury</code> attribute), 53		<code>scale_factor_lid_height</code> ( <code>pycozmo.expressions.expressions.Annoyance</code> attribute), 53	
<code>scale_factor_lid_bend</code> ( <code>pycozmo.expressions.expressions.Guilt</code> attribute), 50		<code>scale_factor_lid_height</code> ( <code>pycozmo.expressions.expressions.Asleep</code> attribute), 56	
<code>scale_factor_lid_bend</code> ( <code>pycozmo.expressions.expressions.Happiness</code> attribute), 46		<code>scale_factor_lid_height</code> ( <code>pycozmo.expressions.expressions.Boredom</code> attribute), 55	
<code>scale_factor_lid_bend</code> ( <code>pycozmo.expressions.expressions.Horror</code> attribute), 52		<code>scale_factor_lid_height</code> ( <code>pycozmo.expressions.expressions.Confusion</code> attribute), 57	
<code>scale_factor_lid_bend</code> ( <code>pycozmo.expressions.expressions.Neutral</code> attribute), 45		<code>scale_factor_lid_height</code> ( <code>pycozmo.expressions.expressions.Despair</code> attribute), 50	
<code>scale_factor_lid_bend</code> ( <code>pycozmo.expressions.expressions.Pleading</code> attribute), 48		<code>scale_factor_lid_height</code> ( <code>pycozmo.expressions.expressions.Rejection</code> attribute), 55	

<i>cozmo.expressions.expressions.Disappointment</i> attribute), 51	<i>cozmo.procedural_face.ProceduralEye</i> attribute), 85
scale_factor_lid_height (py- <i>cozmo.expressions.expressions.Disgust</i> at- tribute), 47	scale_factor_lid_height (py- <i>cozmo.procedural_face.ProceduralFace</i> attribute), 85
scale_factor_lid_height (py- <i>cozmo.expressions.expressions.Embarrassment</i> attribute), 51	scale_factor_lid_height (py- <i>cozmo.procedural_face.ProceduralLid</i> at- tribute), 84
scale_factor_lid_height (py- <i>cozmo.expressions.expressions.Excitement</i> attribute), 58	scale_x ( <i>pycozmo.expressions.expressions.Amazement</i> attribute), 57
scale_factor_lid_height (py- <i>cozmo.expressions.expressions.Fear</i> attribute), 48	scale_x ( <i>pycozmo.expressions.expressions.Anger</i> at- tribute), 45
scale_factor_lid_height (py- <i>cozmo.expressions.expressions.Fury</i> attribute), 53	scale_x ( <i>pycozmo.expressions.expressions.Annoyance</i> attribute), 53
scale_factor_lid_height (py- <i>cozmo.expressions.expressions.Guilt</i> attribute), 50	scale_x ( <i>pycozmo.expressions.expressions.Asleep</i> at- tribute), 56
scale_factor_lid_height (py- <i>cozmo.expressions.expressions.Happiness</i> attribute), 46	scale_x ( <i>pycozmo.expressions.expressions.Boredom</i> attribute), 55
scale_factor_lid_height (py- <i>cozmo.expressions.expressions.Horror</i> at- tribute), 52	scale_x ( <i>pycozmo.expressions.expressions.Confusion</i> attribute), 57
scale_factor_lid_height (py- <i>cozmo.expressions.expressions.Neutral</i> tribute), 45	scale_x ( <i>pycozmo.expressions.expressions.Despair</i> at- tribute), 50
scale_factor_lid_height (py- <i>cozmo.expressions.expressions.Pleading</i> attribute), 48	scale_x ( <i>pycozmo.expressions.expressions.Disappointment</i> attribute), 51
scale_factor_lid_height (py- <i>cozmo.expressions.expressions.Rejection</i> attribute), 55	scale_x ( <i>pycozmo.expressions.expressions.Disgust</i> at- tribute), 47
scale_factor_lid_height (py- <i>cozmo.expressions.expressions.Sadness</i> tribute), 46	scale_x ( <i>pycozmo.expressions.expressions.Embarrassment</i> attribute), 51
scale_factor_lid_height (py- <i>cozmo.expressions.expressions.Skepticism</i> attribute), 52	scale_x ( <i>pycozmo.expressions.expressions.Excitement</i> attribute), 58
scale_factor_lid_height (py- <i>cozmo.expressions.expressions.Surprise</i> tribute), 47	scale_x ( <i>pycozmo.expressions.expressions.Fear</i> attribute), 48
scale_factor_lid_height (py- <i>cozmo.expressions.expressions.Suspicion</i> attribute), 54	scale_x ( <i>pycozmo.expressions.expressions.Fury</i> attribute), 54
scale_factor_lid_height (py- <i>cozmo.expressions.expressions.Tiredness</i> attribute), 56	scale_x ( <i>pycozmo.expressions.expressions.Guilt</i> attribute), 50
scale_factor_lid_height (py- <i>cozmo.expressions.expressions.Vulnerability</i> attribute), 49	scale_x ( <i>pycozmo.expressions.expressions.Happiness</i> attribute), 46
scale_factor_lid_height (py-	scale_x ( <i>pycozmo.expressions.expressions.Horror</i> at- tribute), 52
	scale_x ( <i>pycozmo.expressions.expressions.Neutral</i> at- tribute), 45
	scale_x ( <i>pycozmo.expressions.expressions.Pleading</i> attribute), 49
	scale_x ( <i>pycozmo.expressions.expressions.Rejection</i> attribute), 55
	scale_x ( <i>pycozmo.expressions.expressions.Sadness</i> at- tribute), 46
	scale_x ( <i>pycozmo.expressions.expressions.Skepticism</i> attribute), 52
	scale_x ( <i>pycozmo.expressions.expressions.Surprise</i> at- tribute), 47
	scale_x ( <i>pycozmo.expressions.expressions.Suspicion</i> attribute), 54



<code>scale_x</code> ( <code>pycozmo.expressions.expressions.Tiredness</code> attribute), 56	<code>scale_y</code> ( <code>pycozmo.expressions.expressions.Tiredness</code> attribute), 56
<code>scale_x</code> ( <code>pycozmo.expressions.expressions.Vulnerability</code> attribute), 49	<code>scale_y</code> ( <code>pycozmo.expressions.expressions.Vulnerability</code> attribute), 49
<code>scale_x</code> ( <code>pycozmo.procedural_face.ProceduralEye</code> attribute), 85	<code>scale_y</code> ( <code>pycozmo.procedural_face.ProceduralEye</code> attribute), 85
<code>scale_x</code> ( <code>pycozmo.procedural_face.ProceduralFace</code> attribute), 86	<code>scale_y</code> ( <code>pycozmo.procedural_face.ProceduralFace</code> attribute), 86
<code>scale_y</code> ( <code>pycozmo.expressions.expressions.Amazement</code> attribute), 57	<code>scope</code> ( <code>pycozmo.audiokinetic.soundbank.EventAction</code> attribute), 41
<code>scale_y</code> ( <code>pycozmo.expressions.expressions.Anger</code> attribute), 45	<code>seek_cur()</code> ( <code>pycozmo.protocol_utils.BinaryReader</code> method), 139
<code>scale_y</code> ( <code>pycozmo.expressions.expressions.Annoyance</code> attribute), 53	<code>seek_set()</code> ( <code>pycozmo.protocol_utils.BinaryReader</code> method), 139
<code>scale_y</code> ( <code>pycozmo.expressions.expressions.Asleep</code> attribute), 56	<code>send()</code> ( <code>pycozmo.conn.Connection</code> method), 75
<code>scale_y</code> ( <code>pycozmo.expressions.expressions.Boredom</code> attribute), 55	<code>send()</code> ( <code>pycozmo.conn.SendThread</code> method), 73
<code>scale_y</code> ( <code>pycozmo.expressions.expressions.Confusion</code> attribute), 57	<code>SendThread</code> (class in <code>pycozmo.conn</code> ), 72
<code>scale_y</code> ( <code>pycozmo.expressions.expressions.Despair</code> attribute), 50	<code>SendWindow</code> (class in <code>pycozmo.window</code> ), 148
<code>scale_y</code> ( <code>pycozmo.expressions.expressions.Disappointment</code> attribute), 51	<code>seq</code> ( <code>pycozmo.frame.Frame</code> attribute), 81
<code>scale_y</code> ( <code>pycozmo.expressions.expressions.Disgust</code> attribute), 47	<code>seq</code> ( <code>pycozmo.protocol_base.Packet</code> attribute), 91
<code>scale_y</code> ( <code>pycozmo.expressions.expressions.Embarrassment</code> attribute), 51	<code>seq</code> ( <code>pycozmo.protocol_base.UnknownCommand</code> attribute), 92
<code>scale_y</code> ( <code>pycozmo.expressions.expressions.Excitement</code> attribute), 58	<code>seq</code> ( <code>pycozmo.protocol_base.UnknownEvent</code> attribute), 92
<code>scale_y</code> ( <code>pycozmo.expressions.expressions.Fear</code> attribute), 48	<code>seq</code> ( <code>pycozmo.protocol_base.UnknownPacket</code> attribute), 92
<code>scale_y</code> ( <code>pycozmo.expressions.expressions.Fury</code> attribute), 54	<code>seq</code> ( <code>pycozmo.protocol_encoder.AbortAnimation</code> attribute), 95
<code>scale_y</code> ( <code>pycozmo.expressions.expressions.Guilt</code> attribute), 50	<code>seq</code> ( <code>pycozmo.protocol_encoder.AcknowledgeAction</code> attribute), 95
<code>scale_y</code> ( <code>pycozmo.expressions.expressions.Happiness</code> attribute), 46	<code>seq</code> ( <code>pycozmo.protocol_encoder.AnimationEnded</code> attribute), 98
<code>scale_y</code> ( <code>pycozmo.expressions.expressions.Horror</code> attribute), 52	<code>seq</code> ( <code>pycozmo.protocol_encoder.AnimationStarted</code> attribute), 98
<code>scale_y</code> ( <code>pycozmo.expressions.expressions.Neutral</code> attribute), 45	<code>seq</code> ( <code>pycozmo.protocol_encoder.AnimationState</code> attribute), 98
<code>scale_y</code> ( <code>pycozmo.expressions.expressions.Pleading</code> attribute), 49	<code>seq</code> ( <code>pycozmo.protocol_encoder.AnimBackpackLights</code> attribute), 96
<code>scale_y</code> ( <code>pycozmo.expressions.expressions.Rejection</code> attribute), 55	<code>seq</code> ( <code>pycozmo.protocol_encoder.AnimBody</code> attribute), 96
<code>scale_y</code> ( <code>pycozmo.expressions.expressions.Sadness</code> attribute), 46	<code>seq</code> ( <code>pycozmo.protocol_encoder.AnimHead</code> attribute), 97
<code>scale_y</code> ( <code>pycozmo.expressions.expressions.Skepticism</code> attribute), 52	<code>seq</code> ( <code>pycozmo.protocol_encoder.AnimLift</code> attribute), 97
<code>scale_y</code> ( <code>pycozmo.expressions.expressions.Surprise</code> attribute), 47	<code>seq</code> ( <code>pycozmo.protocol_encoder.AppendPathSegArc</code> attribute), 99
<code>scale_y</code> ( <code>pycozmo.expressions.expressions.Suspicion</code> attribute), 54	<code>seq</code> ( <code>pycozmo.protocol_encoder.AppendPathSegLine</code> attribute), 100
	<code>seq</code> ( <code>pycozmo.protocol_encoder.AppendPathSegPointTurn</code> attribute), 100
	<code>seq</code> ( <code>pycozmo.protocol_encoder.BodyInfo</code> attribute), 101
	<code>seq</code> ( <code>pycozmo.protocol_encoder.ButtonPressed</code> attribute), 101
	<code>seq</code> ( <code>pycozmo.protocol_encoder.ClearPath</code> attribute), 102
	<code>seq</code> ( <code>pycozmo.protocol_encoder.Connect</code> attribute), 102
	<code>seq</code> ( <code>pycozmo.protocol_encoder.CubeId</code> attribute), 103

seq (pycozmo.protocol_encoder.CubeLights attribute), 103	118
seq (pycozmo.protocol_encoder.DebugData attribute), 103	seq (pycozmo.protocol_encoder.ObjectAvailable attribute), 119
seq (pycozmo.protocol_encoder.Disconnect attribute), 104	seq (pycozmo.protocol_encoder.ObjectConnect attribute), 119
seq (pycozmo.protocol_encoder.DisplayImage attribute), 104	seq (pycozmo.protocol_encoder.ObjectConnectionState attribute), 120
seq (pycozmo.protocol_encoder.DriveWheels attribute), 105	seq (pycozmo.protocol_encoder.ObjectMoved attribute), 120
seq (pycozmo.protocol_encoder.Enable attribute), 105	seq (pycozmo.protocol_encoder.ObjectPowerLevel attribute), 121
seq (pycozmo.protocol_encoder.EnableAnimationState attribute), 105	seq (pycozmo.protocol_encoder.ObjectStoppedMoving attribute), 121
seq (pycozmo.protocol_encoder.EnableCamera attribute), 106	seq (pycozmo.protocol_encoder.ObjectTapFiltered attribute), 122
seq (pycozmo.protocol_encoder.EnableColorImages attribute), 106	seq (pycozmo.protocol_encoder.ObjectTapped attribute), 122
seq (pycozmo.protocol_encoder.EnableStopOnCliff attribute), 107	seq (pycozmo.protocol_encoder.ObjectUpAxisChanged attribute), 124
seq (pycozmo.protocol_encoder.EndAnimation attribute), 107	seq (pycozmo.protocol_encoder.OutputAudio attribute), 124
seq (pycozmo.protocol_encoder.ExecutePath attribute), 107	seq (pycozmo.protocol_encoder.OutputSilence attribute), 124
seq (pycozmo.protocol_encoder.FallingStarted attribute), 108	seq (pycozmo.protocol_encoder.PathFollowingEvent attribute), 125
seq (pycozmo.protocol_encoder.FallingStopped attribute), 108	seq (pycozmo.protocol_encoder.Ping attribute), 126
seq (pycozmo.protocol_encoder.FirmwareSignature attribute), 109	seq (pycozmo.protocol_encoder.RecordHeading attribute), 126
seq (pycozmo.protocol_encoder.FirmwareUpdate attribute), 109	seq (pycozmo.protocol_encoder.RobotDelocalized attribute), 126
seq (pycozmo.protocol_encoder.FirmwareUpdateResult attribute), 109	seq (pycozmo.protocol_encoder.RobotPoked attribute), 127
seq (pycozmo.protocol_encoder.HardwareInfo attribute), 110	seq (pycozmo.protocol_encoder.RobotState attribute), 128
seq (pycozmo.protocol_encoder.ImageChunk attribute), 111	seq (pycozmo.protocol_encoder.SetAccessoryDiscovery attribute), 128
seq (pycozmo.protocol_encoder.ImageImuData attribute), 111	seq (pycozmo.protocol_encoder.SetCameraParams attribute), 129
seq (pycozmo.protocol_encoder.Keyframe attribute), 112	seq (pycozmo.protocol_encoder.SetHeadAngle attribute), 129
seq (pycozmo.protocol_encoder.LightStateCenter attribute), 113	seq (pycozmo.protocol_encoder.SetHeadLight attribute), 130
seq (pycozmo.protocol_encoder.LightStateSide attribute), 114	seq (pycozmo.protocol_encoder.SetLiftHeight attribute), 130
seq (pycozmo.protocol_encoder.MotorCalibration attribute), 114	seq (pycozmo.protocol_encoder.SetOrigin attribute), 131
seq (pycozmo.protocol_encoder.MoveHead attribute), 115	seq (pycozmo.protocol_encoder.SetRobotVolume attribute), 131
seq (pycozmo.protocol_encoder.MoveLift attribute), 115	seq (pycozmo.protocol_encoder.ShutdownRobot attribute), 132
seq (pycozmo.protocol_encoder.NvStorageOp attribute), 117	seq (pycozmo.protocol_encoder.StartAnimation attribute), 132
seq (pycozmo.protocol_encoder.NvStorageOpResult attribute), 118	seq (pycozmo.protocol_encoder.StartMotorCalibration attribute), 132
seq (pycozmo.protocol_encoder.ObjectAccel attribute),	

- `seq` (`pycozmo.protocol_encoder.StopAllMotors` attribute), 133
- `seq` (`pycozmo.protocol_encoder.StreamObjectAccel` attribute), 133
- `seq` (`pycozmo.protocol_encoder.SyncTime` attribute), 133
- `seq` (`pycozmo.protocol_encoder.TrimPath` attribute), 134
- `seq` (`pycozmo.protocol_encoder.TurnInPlace` attribute), 134
- `seq` (`pycozmo.protocol_encoder.TurnInPlaceAtSpeed` attribute), 135
- `seq` (`pycozmo.protocol_encoder.TurnToRecordedHeading` attribute), 135
- `seq` (`pycozmo.protocol_encoder.WifiOff` attribute), 136
- `serial_number` (`pycozmo.protocol_encoder.BodyInfo` attribute), 101
- `serial_number_head` (`pycozmo.protocol_encoder.HardwareInfo` attribute), 110
- `set()` (`pycozmo.emotions.EmotionType` method), 76
- `set_all_backpack_lights()` (`pycozmo.client.Client` method), 70
- `set_backpack_lights()` (`pycozmo.client.Client` method), 70
- `set_backpack_lights_off()` (`pycozmo.client.Client` method), 70
- `set_center_backpack_lights()` (`pycozmo.client.Client` method), 70
- `set_forward()` (`pycozmo.util.Matrix44` method), 145
- `set_head_angle()` (`pycozmo.client.Client` method), 70
- `set_head_light()` (`pycozmo.client.Client` method), 70
- `set_left()` (`pycozmo.util.Matrix44` method), 145
- `set_lift_height()` (`pycozmo.client.Client` method), 70
- `set_pos()` (`pycozmo.util.Matrix44` method), 145
- `set_to()` (`pycozmo.util.Vector2` method), 144
- `set_to()` (`pycozmo.util.Vector3` method), 144
- `set_up()` (`pycozmo.util.Matrix44` method), 145
- `set_volume()` (`pycozmo.client.Client` method), 70
- `SetAccessoryDiscovery` (class in `pycozmo.protocol_encoder`), 128
- `SetCameraParams` (class in `pycozmo.protocol_encoder`), 128
- `setDaemon()` (`pycozmo.conn.Connection` method), 75
- `setDaemon()` (`pycozmo.conn.ReceiveThread` method), 72
- `setDaemon()` (`pycozmo.conn.SendThread` method), 73
- `SetHeadAngle` (class in `pycozmo.protocol_encoder`), 129
- `SetHeadLight` (class in `pycozmo.protocol_encoder`), 129
- `SetLiftHeight` (class in `pycozmo.protocol_encoder`), 130
- `setName()` (`pycozmo.conn.Connection` method), 75
- `setName()` (`pycozmo.conn.ReceiveThread` method), 72
- `setName()` (`pycozmo.conn.SendThread` method), 74
- `SetOrigin` (class in `pycozmo.protocol_encoder`), 130
- `SetRobotVolume` (class in `pycozmo.protocol_encoder`), 131
- `setup_basic_logging()` (in module `pycozmo.run`), 142
- `SFX` (class in `pycozmo.audiokinetic.soundbank`), 42
- `should_resume_last` (`pycozmo.behavior.ReactionTrigger` attribute), 67
- `ShutdownRobot` (class in `pycozmo.protocol_encoder`), 131
- `signature` (`pycozmo.protocol_encoder.FirmwareSignature` attribute), 109
- `SingleShot` (`pycozmo.protocol_encoder.ImageSendMode` attribute), 112
- `Skepticism` (class in `pycozmo.expressions.expressions`), 52
- `sleep()` (`pycozmo.util.FPSTimer` method), 147
- `SoundBank` (class in `pycozmo.audiokinetic.soundbank`), 42
- `soundbank_id` (`pycozmo.audiokinetic.soundbank.Event` attribute), 41
- `soundbank_id` (`pycozmo.audiokinetic.soundbank.EventAction` attribute), 41
- `soundbank_id` (`pycozmo.audiokinetic.soundbank.File` attribute), 41
- `soundbank_id` (`pycozmo.audiokinetic.soundbank.SFX` attribute), 42
- `soundbank_id` (`pycozmo.audiokinetic.soundbanksinfo.EventInfo` attribute), 43
- `soundbank_id` (`pycozmo.audiokinetic.soundbanksinfo.FileInfo` attribute), 43
- `SoundBankInfo` (class in `pycozmo.audiokinetic.soundbanksinfo`), 43
- `SoundBankReader` (class in `pycozmo.audiokinetic.soundbank`), 42
- `Speed` (class in `pycozmo.util`), 143
- `speed` (`pycozmo.protocol_encoder.AnimBody` attribute), 96
- `speed_mmmps` (`pycozmo.protocol_encoder.AppendPathSegArc` attribute), 99
- `speed_mmmps` (`pycozmo.protocol_encoder.AppendPathSegLine` attribute), 100
- `speed_mmmps` (`pycozmo.protocol_encoder.AppendPathSegPointTurn` attribute), 100
- `speed_mmmps` (`pycozmo.protocol_encoder.PathSegmentSpeed` attribute), 125
- `speed_rad_per_sec` (`pycozmo.protocol_encoder.MoveHead` attribute), 115



speed\_rad\_per\_sec (pycozmo.protocol\_encoder.MoveLift attribute), 115  
 speed\_rad\_per\_sec (pycozmo.protocol\_encoder.TurnInPlace attribute), 135  
 start() (pycozmo.anim\_controller.AnimationController method), 61  
 start() (pycozmo.brain.Brain method), 68  
 start() (pycozmo.client.Client method), 70  
 start() (pycozmo.conn.Connection method), 75  
 start() (pycozmo.conn.ReceiveThread method), 72  
 start() (pycozmo.conn.SendThread method), 74  
 start\_angle\_rad (pycozmo.protocol\_encoder.AppendPathSegArc attribute), 99  
 StartAnimation (class in pycozmo.protocol\_encoder), 132  
 StartMotorCalibration (class in pycozmo.protocol\_encoder), 132  
 states (pycozmo.protocol\_encoder.CubeLights attribute), 103  
 states (pycozmo.protocol\_encoder.LightStateCenter attribute), 113  
 states (pycozmo.protocol\_encoder.LightStateSide attribute), 114  
 STATS\_INTERVAL (pycozmo.conn.Connection attribute), 74  
 status (pycozmo.protocol\_encoder.FirmwareUpdateResult attribute), 109  
 status (pycozmo.protocol\_encoder.ImageChunk attribute), 111  
 status (pycozmo.protocol\_encoder.RobotState attribute), 128  
 stop() (pycozmo.anim\_controller.AnimationController method), 61  
 stop() (pycozmo.brain.Brain method), 68  
 stop() (pycozmo.client.Client method), 70  
 stop() (pycozmo.conn.Connection method), 75  
 stop() (pycozmo.conn.ReceiveThread method), 72  
 stop() (pycozmo.conn.SendThread method), 74  
 stop\_all\_motors() (pycozmo.client.Client method), 70  
 StopAllMotors (class in pycozmo.protocol\_encoder), 132  
 str\_to\_image() (in module pycozmo.image\_encoder), 82  
 strategy (pycozmo.activity.Activity attribute), 59  
 Stream (pycozmo.protocol\_encoder.ImageSendMode attribute), 112  
 StreamObjectAccel (class in pycozmo.protocol\_encoder), 133  
 StringArgument (class in pycozmo.protocol\_ast), 89  
 Struct (class in pycozmo.protocol\_ast), 87  
 Struct (class in pycozmo.protocol\_base), 91  
 Surprise (class in pycozmo.expressions.expressions), 46  
 Suspicion (class in pycozmo.expressions.expressions), 54  
 SVGA (pycozmo.protocol\_encoder.ImageResolution attribute), 112  
 sweep\_rad (pycozmo.protocol\_encoder.AppendPathSegArc attribute), 99  
 SXGA (pycozmo.protocol\_encoder.ImageResolution attribute), 112  
 SyncTime (class in pycozmo.protocol\_encoder), 133

## T

tabulated\_string (pycozmo.util.Matrix44 attribute), 145  
 tag (pycozmo.protocol\_encoder.AnimationState attribute), 98  
 tag (pycozmo.protocol\_encoder.NvStorageOp attribute), 117  
 tag (pycozmo.protocol\_encoder.NvStorageOpResult attribute), 118  
 tail (pycozmo.protocol\_encoder.TrimPath attribute), 134  
 tap\_neg (pycozmo.protocol\_encoder.ObjectTapped attribute), 122  
 tap\_pos (pycozmo.protocol\_encoder.ObjectTapped attribute), 122  
 tap\_time (pycozmo.protocol\_encoder.ObjectTapped attribute), 122  
 tell() (pycozmo.protocol\_utils.BinaryReader method), 139  
 time (pycozmo.protocol\_encoder.ObjectTapFiltered attribute), 122  
 time\_sent\_ms (pycozmo.protocol\_encoder.Ping attribute), 126  
 Timeout, 80  
 timestamp (pycozmo.protocol\_encoder.AnimationState attribute), 98  
 timestamp (pycozmo.protocol\_encoder.ObjectAccel attribute), 118  
 timestamp (pycozmo.protocol\_encoder.ObjectMoved attribute), 120  
 timestamp (pycozmo.protocol\_encoder.ObjectStoppedMoving attribute), 121  
 timestamp (pycozmo.protocol\_encoder.ObjectTapFiltered attribute), 122  
 timestamp (pycozmo.protocol\_encoder.ObjectTapped attribute), 122  
 timestamp (pycozmo.protocol\_encoder.ObjectUpAxisChanged attribute), 124  
 timestamp (pycozmo.protocol\_encoder.RobotState attribute), 128

`timestamp` (`pycozmo.protocol_encoder.SyncTime` attribute), 134

`Tiredness` (class in `pycozmo.expressions.expressions`), 55

`to_bytes()` (`pycozmo.frame.Frame` method), 81

`to_bytes()` (`pycozmo.protocol_base.Packet` method), 91

`to_bytes()` (`pycozmo.protocol_base.Struct` method), 91

`to_bytes()` (`pycozmo.protocol_base.UnknownCommand` method), 92

`to_bytes()` (`pycozmo.protocol_base.UnknownEvent` method), 92

`to_bytes()` (`pycozmo.protocol_base.UnknownPacket` method), 92

`to_bytes()` (`pycozmo.protocol_encoder.AbortAnimation` method), 95

`to_bytes()` (`pycozmo.protocol_encoder.AcknowledgeAction` method), 95

`to_bytes()` (`pycozmo.protocol_encoder.AnimationEnded` method), 98

`to_bytes()` (`pycozmo.protocol_encoder.AnimationStarted` method), 98

`to_bytes()` (`pycozmo.protocol_encoder.AnimationState` method), 99

`to_bytes()` (`pycozmo.protocol_encoder.AnimBackpackLights` method), 96

`to_bytes()` (`pycozmo.protocol_encoder.AnimBody` method), 96

`to_bytes()` (`pycozmo.protocol_encoder.AnimHead` method), 97

`to_bytes()` (`pycozmo.protocol_encoder.AnimLift` method), 97

`to_bytes()` (`pycozmo.protocol_encoder.AppendPathSegmentArc` method), 99

`to_bytes()` (`pycozmo.protocol_encoder.AppendPathSegmentLine` method), 100

`to_bytes()` (`pycozmo.protocol_encoder.AppendPathSegmentPointToLine` method), 100

`to_bytes()` (`pycozmo.protocol_encoder.BodyInfo` method), 101

`to_bytes()` (`pycozmo.protocol_encoder.ButtonPressed` method), 102

`to_bytes()` (`pycozmo.protocol_encoder.ClearPath` method), 102

`to_bytes()` (`pycozmo.protocol_encoder.Connect` method), 102

`to_bytes()` (`pycozmo.protocol_encoder.CubeId` method), 103

`to_bytes()` (`pycozmo.protocol_encoder.CubeLights` method), 103

`to_bytes()` (`pycozmo.protocol_encoder.DebugData` method), 103

`to_bytes()` (`pycozmo.protocol_encoder.Disconnect` method), 104

`to_bytes()` (`pycozmo.protocol_encoder.DisplayImage` method), 104

`to_bytes()` (`pycozmo.protocol_encoder.DriveWheels` method), 105

`to_bytes()` (`pycozmo.protocol_encoder.Enable` method), 105

`to_bytes()` (`pycozmo.protocol_encoder.EnableAnimationState` method), 105

`to_bytes()` (`pycozmo.protocol_encoder.EnableCamera` method), 106

`to_bytes()` (`pycozmo.protocol_encoder.EnableColorImages` method), 106

`to_bytes()` (`pycozmo.protocol_encoder.EnableStopOnCliff` method), 107

`to_bytes()` (`pycozmo.protocol_encoder.EndAnimation` method), 107

`to_bytes()` (`pycozmo.protocol_encoder.ExecutePath` method), 107

`to_bytes()` (`pycozmo.protocol_encoder.FallingStarted` method), 108

`to_bytes()` (`pycozmo.protocol_encoder.FallingStopped` method), 108

`to_bytes()` (`pycozmo.protocol_encoder.FirmwareSignature` method), 109

`to_bytes()` (`pycozmo.protocol_encoder.FirmwareUpdate` method), 109

`to_bytes()` (`pycozmo.protocol_encoder.FirmwareUpdateResult` method), 109

`to_bytes()` (`pycozmo.protocol_encoder.HardwareInfo` method), 110

`to_bytes()` (`pycozmo.protocol_encoder.ImageChunk` method), 111

`to_bytes()` (`pycozmo.protocol_encoder.ImageImuData` method), 111

`to_bytes()` (`pycozmo.protocol_encoder.Keyframe` method), 112

`to_bytes()` (`pycozmo.protocol_encoder.LightState` method), 113

`to_bytes()` (`pycozmo.protocol_encoder.LightStateCenter` method), 113

`to_bytes()` (`pycozmo.protocol_encoder.LightStateSide` method), 114

`to_bytes()` (`pycozmo.protocol_encoder.MotorCalibration` method), 114

`to_bytes()` (`pycozmo.protocol_encoder.MoveHead` method), 115

`to_bytes()` (`pycozmo.protocol_encoder.MoveLift` method), 115

`to_bytes()` (`pycozmo.protocol_encoder.NvStorageOp` method), 117

`to_bytes()` (`pycozmo.protocol_encoder.NvStorageOpResult` method), 118

`to_bytes()` (`pycozmo.protocol_encoder.ObjectAccel`

`method)`, 118  
`to_bytes()` (`pycozmo.protocol_encoder.ObjectAvailable` `method)`, 119  
`to_bytes()` (`pycozmo.protocol_encoder.ObjectConnect` `method)`, 119  
`to_bytes()` (`pycozmo.protocol_encoder.ObjectConnectionState` `method)`, 120  
`to_bytes()` (`pycozmo.protocol_encoder.ObjectMoved` `method)`, 120  
`to_bytes()` (`pycozmo.protocol_encoder.ObjectPowerLevel` `method)`, 121  
`to_bytes()` (`pycozmo.protocol_encoder.ObjectStoppedMoving` `method)`, 121  
`to_bytes()` (`pycozmo.protocol_encoder.ObjectTapFiltered` `method)`, 122  
`to_bytes()` (`pycozmo.protocol_encoder.ObjectTapped` `method)`, 122  
`to_bytes()` (`pycozmo.protocol_encoder.ObjectUpAxisChanged` `method)`, 124  
`to_bytes()` (`pycozmo.protocol_encoder.OutputAudio` `method)`, 124  
`to_bytes()` (`pycozmo.protocol_encoder.OutputSilence` `method)`, 124  
`to_bytes()` (`pycozmo.protocol_encoder.PathFollowingEvent` `method)`, 125  
`to_bytes()` (`pycozmo.protocol_encoder.PathSegmentSpeed` `method)`, 125  
`to_bytes()` (`pycozmo.protocol_encoder.Ping` `method)`, 126  
`to_bytes()` (`pycozmo.protocol_encoder.RecordHeading` `method)`, 126  
`to_bytes()` (`pycozmo.protocol_encoder.RobotDelocalized` `method)`, 126  
`to_bytes()` (`pycozmo.protocol_encoder.RobotPoked` `method)`, 127  
`to_bytes()` (`pycozmo.protocol_encoder.RobotState` `method)`, 128  
`to_bytes()` (`pycozmo.protocol_encoder.SetAccessoryDiscovery` `method)`, 128  
`to_bytes()` (`pycozmo.protocol_encoder.SetCameraParameters` `method)`, 129  
`to_bytes()` (`pycozmo.protocol_encoder.SetHeadAngle` `method)`, 129  
`to_bytes()` (`pycozmo.protocol_encoder.SetHeadLight` `method)`, 130  
`to_bytes()` (`pycozmo.protocol_encoder.SetLiftHeight` `method)`, 130  
`to_bytes()` (`pycozmo.protocol_encoder.SetOrigin` `method)`, 131  
`to_bytes()` (`pycozmo.protocol_encoder.SetRobotVolume` `method)`, 131  
`to_bytes()` (`pycozmo.protocol_encoder.ShutdownRobot` `method)`, 132  
`to_bytes()` (`pycozmo.protocol_encoder.StartAnimation` `method)`, 132  
`to_bytes()` (`pycozmo.protocol_encoder.StartMotorCalibration` `method)`, 132  
`to_bytes()` (`pycozmo.protocol_encoder.StopAllMotors` `method)`, 133  
`to_bytes()` (`pycozmo.protocol_encoder.StreamObjectAccelerometer` `method)`, 133  
`to_bytes()` (`pycozmo.protocol_encoder.SyncTime` `method)`, 134  
`to_bytes()` (`pycozmo.protocol_encoder.TrimPath` `method)`, 134  
`to_bytes()` (`pycozmo.protocol_encoder.TurnInPlace` `method)`, 135  
`to_bytes()` (`pycozmo.protocol_encoder.TurnInPlaceAtSpeed` `method)`, 135  
`to_bytes()` (`pycozmo.protocol_encoder.TurnToRecordedHeading` `method)`, 135  
`to_bytes()` (`pycozmo.protocol_encoder.WifiOff` `method)`, 136  
`to_dict()` (`pycozmo.anim_encoder.AnimBackpackLights` `method)`, 65  
`to_dict()` (`pycozmo.anim_encoder.AnimBase` `method)`, 62  
`to_dict()` (`pycozmo.anim_encoder.AnimBodyMotion` `method)`, 64  
`to_dict()` (`pycozmo.anim_encoder.AnimClip` `method)`, 62  
`to_dict()` (`pycozmo.anim_encoder.AnimClips` `method)`, 62  
`to_dict()` (`pycozmo.anim_encoder.AnimEvent` `method)`, 65  
`to_dict()` (`pycozmo.anim_encoder.AnimFaceAnimation` `method)`, 65  
`to_dict()` (`pycozmo.anim_encoder.AnimHeadAngle` `method)`, 63  
`to_dict()` (`pycozmo.anim_encoder.AnimKeyframe` `method)`, 63  
`to_dict()` (`pycozmo.anim_encoder.AnimLiftHeight` `method)`, 63  
`to_dict()` (`pycozmo.anim_encoder.AnimLight` `method)`, 63  
`to_dict()` (`pycozmo.anim_encoder.AnimProceduralFace` `method)`, 65  
`to_dict()` (`pycozmo.anim_encoder.AnimRecordHeading` `method)`, 64  
`to_dict()` (`pycozmo.anim_encoder.AnimRobotAudio` `method)`, 65  
`to_dict()` (`pycozmo.anim_encoder.AnimTurnToRecordedHeading` `method)`, 64  
`to_fb()` (`pycozmo.anim_encoder.AnimBackpackLights` `method)`, 65  
`to_fb()` (`pycozmo.anim_encoder.AnimBase` `method)`, 62  
`to_fb()` (`pycozmo.anim_encoder.AnimBodyMotion` `method)`, 62

`method)`, 64  
`to_fb()` (`pycozmo.anim_encoder.AnimClip` `method`), 62  
`to_fb()` (`pycozmo.anim_encoder.AnimClips` `method`), 62  
`to_fb()` (`pycozmo.anim_encoder.AnimEvent` `method`), 65  
`to_fb()` (`pycozmo.anim_encoder.AnimFaceAnimation` `method`), 65  
`to_fb()` (`pycozmo.anim_encoder.AnimHeadAngle` `method`), 63  
`to_fb()` (`pycozmo.anim_encoder.AnimKeyframe` `method`), 63  
`to_fb()` (`pycozmo.anim_encoder.AnimLiftHeight` `method`), 63  
`to_fb()` (`pycozmo.anim_encoder.AnimProceduralFace` `method`), 65  
`to_fb()` (`pycozmo.anim_encoder.AnimRecordHeading` `method`), 64  
`to_fb()` (`pycozmo.anim_encoder.AnimRobotAudio` `method`), 65  
`to_fb()` (`pycozmo.anim_encoder.AnimTurnToRecordedHeading` `method`), 64  
`to_fb_file()` (`pycozmo.anim_encoder.AnimClips` `method`), 63  
`to_fb_stream()` (`pycozmo.anim_encoder.AnimClips` `method`), 63  
`to_int16()` (`pycozmo.lights.Color` `method`), 82  
`to_json_file()` (`pycozmo.anim_encoder.AnimClips` `method`), 63  
`to_json_stream()` (`pycozmo.anim_encoder.AnimClips` `method`), 63  
`to_matrix()` (`pycozmo.util.Pose` `method`), 147  
`to_matrix()` (`pycozmo.util.Quaternion` `method`), 146  
`to_writer()` (`pycozmo.frame.Frame` `method`), 81  
`to_writer()` (`pycozmo.protocol_base.Packet` `method`), 91  
`to_writer()` (`pycozmo.protocol_base.Struct` `method`), 91  
`to_writer()` (`pycozmo.protocol_base.UnknownCommand` `method`), 92  
`to_writer()` (`pycozmo.protocol_base.UnknownEvent` `method`), 92  
`to_writer()` (`pycozmo.protocol_base.UnknownPacket` `method`), 92  
`to_writer()` (`pycozmo.protocol_encoder.AbortAnimation` `method`), 95  
`to_writer()` (`pycozmo.protocol_encoder.AcknowledgeAction` `method`), 96  
`to_writer()` (`pycozmo.protocol_encoder.AnimationEnded` `method`), 98  
`to_writer()` (`pycozmo.protocol_encoder.AnimationStarted` `method`), 98  
`to_writer()` (`pycozmo.protocol_encoder.AnimationState` `method`), 99  
`to_writer()` (`pycozmo.protocol_encoder.AnimBackpackLights` `method`), 96  
`to_writer()` (`pycozmo.protocol_encoder.AnimBody` `method`), 96  
`to_writer()` (`pycozmo.protocol_encoder.AnimHead` `method`), 97  
`to_writer()` (`pycozmo.protocol_encoder.AnimLift` `method`), 97  
`to_writer()` (`pycozmo.protocol_encoder.AppendPathSegArc` `method`), 99  
`to_writer()` (`pycozmo.protocol_encoder.AppendPathSegLine` `method`), 100  
`to_writer()` (`pycozmo.protocol_encoder.AppendPathSegPointTurn` `method`), 100  
`to_writer()` (`pycozmo.protocol_encoder.BodyInfo` `method`), 101  
`to_writer()` (`pycozmo.protocol_encoder.ButtonPressed` `method`), 102  
`to_writer()` (`pycozmo.protocol_encoder.ClearPath` `method`), 102  
`to_writer()` (`pycozmo.protocol_encoder.Connect` `method`), 102  
`to_writer()` (`pycozmo.protocol_encoder.CubeId` `method`), 103  
`to_writer()` (`pycozmo.protocol_encoder.CubeLights` `method`), 103  
`to_writer()` (`pycozmo.protocol_encoder.DebugData` `method`), 104  
`to_writer()` (`pycozmo.protocol_encoder.Disconnect` `method`), 104  
`to_writer()` (`pycozmo.protocol_encoder.DisplayImage` `method`), 104  
`to_writer()` (`pycozmo.protocol_encoder.DriveWheels` `method`), 105  
`to_writer()` (`pycozmo.protocol_encoder.Enable` `method`), 105  
`to_writer()` (`pycozmo.protocol_encoder.EnableAnimationState` `method`), 105  
`to_writer()` (`pycozmo.protocol_encoder.EnableCamera` `method`), 106  
`to_writer()` (`pycozmo.protocol_encoder.EnableColorImages` `method`), 106  
`to_writer()` (`pycozmo.protocol_encoder.EnableStopOnCliff` `method`), 107  
`to_writer()` (`pycozmo.protocol_encoder.EndAnimation` `method`), 107  
`to_writer()` (`pycozmo.protocol_encoder.ExecutePath` `method`), 107  
`to_writer()` (`pycozmo.protocol_encoder.FallingStarted` `method`), 108  
`to_writer()` (`pycozmo.protocol_encoder.FallingStopped` `method`), 108

---

<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.FirmwareSignature</code> method), 109	<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.PathFollowingEvent</code> method), 125
<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.FirmwareUpdate</code> method), 109	<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.PathSegmentSpeed</code> method), 125
<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.FirmwareUpdateResult</code> method), 110	<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.Ping</code> method), 126
<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.HardwareInfo</code> method), 110	<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.RecordHeading</code> method), 126
<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.ImageChunk</code> method), 111	<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.RobotDelocalized</code> method), 126
<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.ImageImuData</code> method), 111	<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.RobotPoked</code> method), 127
<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.Keyframe</code> method), 112	<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.RobotState</code> method), 128
<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.LightState</code> method), 113	<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.SetAccessoryDiscovery</code> method), 128
<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.LightStateCenter</code> method), 113	<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.SetCameraParams</code> method), 129
<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.LightStateSide</code> method), 114	<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.SetHeadAngle</code> method), 129
<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.MotorCalibration</code> method), 114	<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.SetHeadLight</code> method), 130
<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.MoveHead</code> method), 115	<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.SetLiftHeight</code> method), 130
<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.MoveLift</code> method), 115	<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.SetOrigin</code> method), 131
<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.NvStorageOp</code> method), 117	<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.SetRobotVolume</code> method), 131
<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.NvStorageOpResult</code> method), 118	<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.ShutdownRobot</code> method), 132
<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.ObjectAccel</code> method), 118	<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.StartAnimation</code> method), 132
<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.ObjectAvailable</code> method), 119	<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.StartMotorCalibration</code> method), 132
<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.ObjectConnect</code> method), 119	<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.StopAllMotors</code> method), 133
<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.ObjectConnectionState</code> method), 120	<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.StreamObjectAccel</code> method), 133
<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.ObjectMoved</code> method), 120	<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.SyncTime</code> method), 134
<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.ObjectPowerLevel</code> method), 121	<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.TrimPath</code> method), 134
<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.ObjectStoppedMoving</code> method), 121	<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.TurnInPlace</code> method), 135
<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.ObjectTapFiltered</code> method), 122	<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.TurnInPlaceAtSpeed</code> method), 135
<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.ObjectTapped</code> method), 122	<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.TurnToRecordedHeading</code> method), 135
<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.ObjectUpAxisChanged</code> method), 124	<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.WifiOff</code> method), 136
<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.OutputAudio</code> method), 124	<code>to_x</code> ( <code>pycozmo.protocol_encoder.AppendPathSegLine</code> attribute), 100
<code>to_writer()</code> ( <code>pycozmo.protocol_encoder.OutputSilence</code> method), 125	<code>to_y</code> ( <code>pycozmo.protocol_encoder.AppendPathSegLine</code> attribute), 100



TRACK_WIDTH (in module <i>pycozmo.robot</i> ), 140	type ( <i>pycozmo.protocol_encoder.ClearPath</i> attribute), 102
transition_off_frames (pycozmo.protocol_encoder.LightState attribute), 113	type ( <i>pycozmo.protocol_encoder.Connect</i> attribute), 102
transition_on_frames (pycozmo.protocol_encoder.LightState attribute), 113	type ( <i>pycozmo.protocol_encoder.CubeId</i> attribute), 103
TrimPath (class in <i>pycozmo.protocol_encoder</i> ), 134	type ( <i>pycozmo.protocol_encoder.CubeLights</i> attribute), 103
TurnInPlace (class in <i>pycozmo.protocol_encoder</i> ), 134	type ( <i>pycozmo.protocol_encoder.DebugData</i> attribute), 104
TurnInPlaceAtSpeed (class in <i>pycozmo.protocol_encoder</i> ), 135	type ( <i>pycozmo.protocol_encoder.Disconnect</i> attribute), 104
TurnToRecordedHeading (class in <i>pycozmo.protocol_encoder</i> ), 135	type ( <i>pycozmo.protocol_encoder.DisplayImage</i> attribute), 104
type ( <i>pycozmo.activity.Activity</i> attribute), 59	type ( <i>pycozmo.protocol_encoder.DriveWheels</i> attribute), 105
type ( <i>pycozmo.audiokinetic.soundbank.EventAction</i> attribute), 41	type ( <i>pycozmo.protocol_encoder.Enable</i> attribute), 105
type ( <i>pycozmo.audiokinetic.soundbank.SFX</i> attribute), 42	type ( <i>pycozmo.protocol_encoder.EnableAnimationState</i> attribute), 106
type ( <i>pycozmo.frame.Frame</i> attribute), 81	type ( <i>pycozmo.protocol_encoder.EnableCamera</i> attribute), 106
type ( <i>pycozmo.protocol_base.Packet</i> attribute), 91	type ( <i>pycozmo.protocol_encoder.EnableColorImages</i> attribute), 106
type ( <i>pycozmo.protocol_base.UnknownCommand</i> attribute), 92	type ( <i>pycozmo.protocol_encoder.EnableStopOnCliff</i> attribute), 107
type ( <i>pycozmo.protocol_base.UnknownEvent</i> attribute), 92	type ( <i>pycozmo.protocol_encoder.EndAnimation</i> attribute), 107
type ( <i>pycozmo.protocol_base.UnknownPacket</i> attribute), 92	type ( <i>pycozmo.protocol_encoder.ExecutePath</i> attribute), 107
type ( <i>pycozmo.protocol_encoder.AbortAnimation</i> attribute), 95	type ( <i>pycozmo.protocol_encoder.FallingStarted</i> attribute), 108
type ( <i>pycozmo.protocol_encoder.AcknowledgeAction</i> attribute), 96	type ( <i>pycozmo.protocol_encoder.FallingStopped</i> attribute), 108
type ( <i>pycozmo.protocol_encoder.AnimationEnded</i> attribute), 98	type ( <i>pycozmo.protocol_encoder.FirmwareSignature</i> attribute), 109
type ( <i>pycozmo.protocol_encoder.AnimationStarted</i> attribute), 98	type ( <i>pycozmo.protocol_encoder.FirmwareUpdate</i> attribute), 109
type ( <i>pycozmo.protocol_encoder.AnimationState</i> attribute), 99	type ( <i>pycozmo.protocol_encoder.FirmwareUpdateResult</i> attribute), 110
type ( <i>pycozmo.protocol_encoder.AnimBackpackLights</i> attribute), 96	type ( <i>pycozmo.protocol_encoder.HardwareInfo</i> attribute), 110
type ( <i>pycozmo.protocol_encoder.AnimBody</i> attribute), 96	type ( <i>pycozmo.protocol_encoder.ImageChunk</i> attribute), 111
type ( <i>pycozmo.protocol_encoder.AnimHead</i> attribute), 97	type ( <i>pycozmo.protocol_encoder.ImageImuData</i> attribute), 112
type ( <i>pycozmo.protocol_encoder.AnimLift</i> attribute), 97	type ( <i>pycozmo.protocol_encoder.Keyframe</i> attribute), 113
type ( <i>pycozmo.protocol_encoder.AppendPathSegArc</i> attribute), 99	type ( <i>pycozmo.protocol_encoder.LightStateCenter</i> attribute), 113
type ( <i>pycozmo.protocol_encoder.AppendPathSegLine</i> attribute), 100	type ( <i>pycozmo.protocol_encoder.LightStateSide</i> attribute), 114
type ( <i>pycozmo.protocol_encoder.AppendPathSegPointTurn</i> attribute), 100	type ( <i>pycozmo.protocol_encoder.MotorCalibration</i> attribute), 114
type ( <i>pycozmo.protocol_encoder.BodyInfo</i> attribute), 101	type ( <i>pycozmo.protocol_encoder.MoveHead</i> attribute), 115
type ( <i>pycozmo.protocol_encoder.ButtonPressed</i> attribute), 102	

type (`pycozmo.protocol_encoder.MoveLift attribute`), 115  
 type (`pycozmo.protocol_encoder.NvStorageOp attribute`), 117  
 type (`pycozmo.protocol_encoder.NvStorageOpResult attribute`), 118  
 type (`pycozmo.protocol_encoder.ObjectAccel attribute`), 118  
 type (`pycozmo.protocol_encoder.ObjectAvailable attribute`), 119  
 type (`pycozmo.protocol_encoder.ObjectConnect attribute`), 119  
 type (`pycozmo.protocol_encoder.ObjectConnectionState attribute`), 120  
 type (`pycozmo.protocol_encoder.ObjectMoved attribute`), 120  
 type (`pycozmo.protocol_encoder.ObjectPowerLevel attribute`), 121  
 type (`pycozmo.protocol_encoder.ObjectStoppedMoving attribute`), 121  
 type (`pycozmo.protocol_encoder.ObjectTapFiltered attribute`), 122  
 type (`pycozmo.protocol_encoder.ObjectTapped attribute`), 122  
 type (`pycozmo.protocol_encoder.ObjectUpAxisChanged attribute`), 124  
 type (`pycozmo.protocol_encoder.OutputAudio attribute`), 124  
 type (`pycozmo.protocol_encoder.OutputSilence attribute`), 125  
 type (`pycozmo.protocol_encoder.PathFollowingEvent attribute`), 125  
 type (`pycozmo.protocol_encoder.Ping attribute`), 126  
 type (`pycozmo.protocol_encoder.RecordHeading attribute`), 126  
 type (`pycozmo.protocol_encoder.RobotDelocalized attribute`), 127  
 type (`pycozmo.protocol_encoder.RobotPoked attribute`), 127  
 type (`pycozmo.protocol_encoder.RobotState attribute`), 128  
 type (`pycozmo.protocol_encoder.SetAccessoryDiscovery attribute`), 128  
 type (`pycozmo.protocol_encoder.SetCameraParams attribute`), 129  
 type (`pycozmo.protocol_encoder.SetHeadAngle attribute`), 129  
 type (`pycozmo.protocol_encoder.SetHeadLight attribute`), 130  
 type (`pycozmo.protocol_encoder.SetLiftHeight attribute`), 130  
 type (`pycozmo.protocol_encoder.SetOrigin attribute`), 131  
 type (`pycozmo.protocol_encoder.SetRobotVolume attribute`), 131  
 type (`pycozmo.protocol_encoder.ShutdownRobot attribute`), 132  
 type (`pycozmo.protocol_encoder.StartAnimation attribute`), 132  
 type (`pycozmo.protocol_encoder.StartMotorCalibration attribute`), 132  
 type (`pycozmo.protocol_encoder.StopAllMotors attribute`), 133  
 type (`pycozmo.protocol_encoder.StreamObjectAccel attribute`), 133  
 type (`pycozmo.protocol_encoder.SyncTime attribute`), 134  
 type (`pycozmo.protocol_encoder.TrimPath attribute`), 134  
 type (`pycozmo.protocol_encoder.TurnInPlace attribute`), 135  
 type (`pycozmo.protocol_encoder.TurnInPlaceAtSpeed attribute`), 135  
 type (`pycozmo.protocol_encoder.TurnToRecordedHeading attribute`), 135  
 type (`pycozmo.protocol_encoder.WifiOff attribute`), 136  
 type\_hint() (`pycozmo.protocol_ast.Argument method`), 87  
 type\_hint() (`pycozmo.protocol_ast.BoolArgument method`), 88  
 type\_hint() (`pycozmo.protocol_ast.Command method`), 90  
 type\_hint() (`pycozmo.protocol_ast.Connect method`), 90  
 type\_hint() (`pycozmo.protocol_ast.Disconnect method`), 90  
 type\_hint() (`pycozmo.protocol_ast.DoubleArgument method`), 88  
 type\_hint() (`pycozmo.protocol_ast.EnumArgument method`), 89  
 type\_hint() (`pycozmo.protocol_ast.Event method`), 90  
 type\_hint() (`pycozmo.protocol_ast.FArrayArgument method`), 89  
 type\_hint() (`pycozmo.protocol_ast.FloatArgument method`), 88  
 type\_hint() (`pycozmo.protocol_ast.Int16Argument method`), 89  
 type\_hint() (`pycozmo.protocol_ast.Int32Argument method`), 89  
 type\_hint() (`pycozmo.protocol_ast.Int8Argument method`), 89  
 type\_hint() (`pycozmo.protocol_ast.IntArgument method`), 88  
 type\_hint() (`pycozmo.protocol_ast.Keyframe method`), 90  
 type\_hint() (`pycozmo.protocol_ast.Packet method`), 90

[type\\_hint\(\)](#) (*pycozmo.protocol\_ast.Ping method*), 90  
[type\\_hint\(\)](#) (*pycozmo.protocol\_ast.StringArgument method*), 90  
[type\\_hint\(\)](#) (*pycozmo.protocol\_ast.Struct method*), 87  
[type\\_hint\(\)](#) (*pycozmo.protocol\_ast.UInt16Argument method*), 88  
[type\\_hint\(\)](#) (*pycozmo.protocol\_ast.UInt32Argument method*), 88  
[type\\_hint\(\)](#) (*pycozmo.protocol\_ast.UInt8Argument method*), 88  
[type\\_hint\(\)](#) (*pycozmo.protocol\_ast.UIntArgument method*), 88  
[type\\_hint\(\)](#) (*pycozmo.protocol\_ast.VArrayArgument method*), 89

## U

[UInt16Argument](#) (*class in pycozmo.protocol\_ast*), 88  
[UInt32Argument](#) (*class in pycozmo.protocol\_ast*), 88  
[UInt8Argument](#) (*class in pycozmo.protocol\_ast*), 88  
[UIntArgument](#) (*class in pycozmo.protocol\_ast*), 88  
[UNKNOWN](#) (*pycozmo.protocol\_ast.PacketType attribute*), 87  
[unknown](#) (*pycozmo.protocol\_encoder.AnimBody attribute*), 96  
[unknown](#) (*pycozmo.protocol\_encoder.AppendPathSegPoint attribute*), 100  
[UNKNOWN](#) (*pycozmo.protocol\_encoder.BodyColor attribute*), 101  
[unknown](#) (*pycozmo.protocol\_encoder.ClearPath attribute*), 102  
[unknown](#) (*pycozmo.protocol\_encoder.ExecutePath attribute*), 107  
[unknown](#) (*pycozmo.protocol\_encoder.FallingStarted attribute*), 108  
[unknown](#) (*pycozmo.protocol\_encoder.FallingStopped attribute*), 108  
[unknown](#) (*pycozmo.protocol\_encoder.FirmwareSignature attribute*), 109  
[unknown](#) (*pycozmo.protocol\_encoder.LightStateCenter attribute*), 113  
[unknown](#) (*pycozmo.protocol\_encoder.LightStateSide attribute*), 114  
[unknown](#) (*pycozmo.protocol\_encoder.NvStorageOp attribute*), 117  
[unknown](#) (*pycozmo.protocol\_encoder.Ping attribute*), 126  
[unknown](#) (*pycozmo.protocol\_encoder.SyncTime attribute*), 134  
[unknown0](#) (*pycozmo.protocol\_encoder.SetOrigin attribute*), 131  
[unknown1](#) (*pycozmo.protocol\_encoder.HardwareInfo attribute*), 110

[unknown2](#) (*pycozmo.protocol\_encoder.HardwareInfo attribute*), 110  
[unknown4](#) (*pycozmo.protocol\_encoder.TurnInPlace attribute*), 135  
[unknown5](#) (*pycozmo.protocol\_encoder.SetOrigin attribute*), 131  
[unknown5](#) (*pycozmo.protocol\_encoder.TurnInPlace attribute*), 135  
[UnknownAxis](#) (*pycozmo.protocol\_encoder.UpAxis attribute*), 136  
[UnknownCommand](#) (*class in pycozmo.protocol\_base*), 92  
[UnknownEvent](#) (*class in pycozmo.protocol\_base*), 92  
[UnknownObject](#) (*pycozmo.protocol\_encoder.ObjectType attribute*), 123  
[UnknownPacket](#) (*class in pycozmo.protocol\_base*), 91  
[unused](#) (*pycozmo.protocol\_encoder.DebugData attribute*), 104  
[up\\_xyz](#) (*pycozmo.util.Matrix44 attribute*), 146  
[UpAxis](#) (*class in pycozmo.protocol\_encoder*), 135  
[update\(\)](#) (*pycozmo.emotions.EmotionType method*), 76  
[update\\_emotion\\_types\(\)](#) (*pycozmo.brain.Brain method*), 68  
[upper\\_inner\\_radius\\_x](#) (*pycozmo.procedural\_face.ProceduralEye attribute*), 85  
[upper\\_inner\\_radius\\_y](#) (*pycozmo.procedural\_face.ProceduralEye attribute*), 85  
[upper\\_outer\\_radius\\_x](#) (*pycozmo.procedural\_face.ProceduralEye attribute*), 85  
[upper\\_outer\\_radius\\_y](#) (*pycozmo.procedural\_face.ProceduralEye attribute*), 85  
[use\\_head\\_angle](#) (*pycozmo.anim.AnimationGroupMember attribute*), 60  
[UXGA](#) (*pycozmo.protocol\_encoder.ImageResolution attribute*), 112

## V

[validate\\_bool\(\)](#) (*in module pycozmo.protocol\_utils*), 138  
[validate\\_farray\(\)](#) (*in module pycozmo.protocol\_utils*), 138  
[validate\\_float\(\)](#) (*in module pycozmo.protocol\_utils*), 138  
[validate\\_integer\(\)](#) (*in module pycozmo.protocol\_utils*), 138  
[validate\\_object\(\)](#) (*in module pycozmo.protocol\_utils*), 138



`validate_string()` (in module `pycozmo.protocol_utils`), 138  
`validate_varray()` (in module `pycozmo.protocol_utils`), 138  
`value` (`pycozmo.emotions.EmotionType` attribute), 76  
`variability_deg` (`pycozmo.protocol_encoder.AnimHead` attribute), 97  
`variability_mm` (`pycozmo.protocol_encoder.AnimLift` attribute), 97  
`VArrayArgument` (class in `pycozmo.protocol_ast`), 89  
`Vector2` (class in `pycozmo.util`), 144  
`Vector3` (class in `pycozmo.util`), 144  
`VerificationSnapshot` (`pycozmo.protocol_encoder.ImageResolution` attribute), 112  
`version` (`pycozmo.audiokinetic.soundbank.SoundBank` attribute), 42  
`VGA` (`pycozmo.protocol_encoder.ImageResolution` attribute), 112  
`Vulnerability` (class in `pycozmo.expressions.expressions`), 49

**W**  
`wait_for()` (`pycozmo.behavior.Behavior` method), 67  
`wait_for()` (`pycozmo.client.Client` method), 70  
`wait_for()` (`pycozmo.conn.Connection` method), 75  
`wait_for()` (`pycozmo.event.Dispatcher` method), 79  
`wait_for_robot()` (`pycozmo.client.Client` method), 71  
`weight` (`pycozmo.anim.AnimationGroupMember` attribute), 60  
`wheel_accel_mmpps2` (`pycozmo.protocol_encoder.TurnInPlaceAtSpeed` attribute), 135  
`wheel_speed_mmpps` (`pycozmo.protocol_encoder.TurnInPlaceAtSpeed` attribute), 135  
`white` (in module `pycozmo.lights`), 83  
`white_light` (in module `pycozmo.lights`), 83  
`WHITE_v10` (`pycozmo.protocol_encoder.BodyColor` attribute), 101  
`WHITE_v15` (`pycozmo.protocol_encoder.BodyColor` attribute), 101  
`width` (`pycozmo.expressions.expressions.Amazement` attribute), 57  
`width` (`pycozmo.expressions.expressions.Anger` attribute), 45  
`width` (`pycozmo.expressions.expressions.Annoyance` attribute), 53  
`width` (`pycozmo.expressions.expressions.Asleep` attribute), 56  
`width` (`pycozmo.expressions.expressions.Boredom` attribute), 55  
`width` (`pycozmo.expressions.expressions.Confusion` attribute), 57  
`width` (`pycozmo.expressions.expressions.Despair` attribute), 50  
`width` (`pycozmo.expressions.expressions.Disappointment` attribute), 51  
`width` (`pycozmo.expressions.expressions.Disgust` attribute), 47  
`width` (`pycozmo.expressions.expressions.Embarrassment` attribute), 51  
`width` (`pycozmo.expressions.expressions.Excitement` attribute), 58  
`width` (`pycozmo.expressions.expressions.Fear` attribute), 48  
`width` (`pycozmo.expressions.expressions.Fury` attribute), 54  
`width` (`pycozmo.expressions.expressions.Guilt` attribute), 50  
`width` (`pycozmo.expressions.expressions.Happiness` attribute), 46  
`width` (`pycozmo.expressions.expressions.Horror` attribute), 52  
`width` (`pycozmo.expressions.expressions.Neutral` attribute), 45  
`width` (`pycozmo.expressions.expressions.Pleading` attribute), 49  
`width` (`pycozmo.expressions.expressions.Rejection` attribute), 55  
`width` (`pycozmo.expressions.expressions.Sadness` attribute), 46  
`width` (`pycozmo.expressions.expressions.Skepticism` attribute), 52  
`width` (`pycozmo.expressions.expressions.Surprise` attribute), 47  
`width` (`pycozmo.expressions.expressions.Suspicion` attribute), 54  
`width` (`pycozmo.expressions.expressions.Tiredness` attribute), 56  
`width` (`pycozmo.expressions.expressions.Vulnerability` attribute), 49  
`width` (`pycozmo.procedural_face.ProceduralEye` attribute), 85  
`width` (`pycozmo.procedural_face.ProceduralFace` attribute), 86  
`width` (`pycozmo.procedural_face.ProceduralLid` attribute), 84  
`WifiOff` (class in `pycozmo.protocol_encoder`), 136  
`with_traceback()` (`pycozmo.audiokinetic.exception.AudioKineticBaseError` method), 40  
`with_traceback()` (`pycozmo.audiokinetic.exception.AudioKineticFormatError` method), 40

`method`), 40  
`with_traceback()` (`pycozmo.audiokinetic.exception.AudioKineticIOError` `method`), 40  
`with_traceback()` (`pycozmo.exception.ConnectionTimeout` `method`), 80  
`with_traceback()` (`pycozmo.exception.NoSpace` `method`), 80  
`with_traceback()` (`pycozmo.exception.PyCozmoConnectionError` `method`), 80  
`with_traceback()` (`pycozmo.exception.PyCozmoException` `method`), 80  
`with_traceback()` (`pycozmo.exception.Timeout` `method`), 80  
`write()` (`pycozmo.protocol_utils.BinaryWriter` `method`), 139  
`write_bytes()` (`pycozmo.protocol_utils.BinaryWriter` `method`), 139  
`write_farray()` (`pycozmo.protocol_utils.BinaryWriter` `method`), 139  
`write_object()` (`pycozmo.protocol_utils.BinaryWriter` `method`), 139  
`write_object_farray()` (`pycozmo.protocol_utils.BinaryWriter` `method`), 139  
`write_object_varray()` (`pycozmo.protocol_utils.BinaryWriter` `method`), 139  
`write_string()` (`pycozmo.protocol_utils.BinaryWriter` `method`), 139  
`write_string_farray()` (`pycozmo.protocol_utils.BinaryWriter` `method`), 139  
`write_string_varray()` (`pycozmo.protocol_utils.BinaryWriter` `method`), 139  
`write_varray()` (`pycozmo.protocol_utils.BinaryWriter` `method`), 140

## X

`x` (`pycozmo.protocol_encoder.AppendPathSegPointTurn` `attribute`), 100  
`x` (`pycozmo.util.Vector2` `attribute`), 144  
`x` (`pycozmo.util.Vector3` `attribute`), 144  
`x_offset` (`pycozmo.procedural_face.ProceduralEye` `attribute`), 85

`x_y` (`pycozmo.util.Vector2` `attribute`), 144  
`x_y_z` (`pycozmo.util.Vector3` `attribute`), 144  
`XGA` (`pycozmo.protocol_encoder.ImageResolution` `attribute`), 112  
`XNegative` (`pycozmo.protocol_encoder.UpAxis` `attribute`), 136  
`XPositive` (`pycozmo.protocol_encoder.UpAxis` `attribute`), 136

## Y

`y` (`pycozmo.procedural_face.ProceduralLid` `attribute`), 84  
`y` (`pycozmo.protocol_encoder.AppendPathSegPointTurn` `attribute`), 100  
`y` (`pycozmo.util.Vector2` `attribute`), 144  
`y` (`pycozmo.util.Vector3` `attribute`), 144  
`y_offset` (`pycozmo.procedural_face.ProceduralLid` `attribute`), 84  
`YNegative` (`pycozmo.protocol_encoder.UpAxis` `attribute`), 136  
`YPositive` (`pycozmo.protocol_encoder.UpAxis` `attribute`), 136  
`YUYV` (`pycozmo.protocol_encoder.ImageEncoding` `attribute`), 111

## Z

`z` (`pycozmo.util.Vector3` `attribute`), 144  
`ZNegative` (`pycozmo.protocol_encoder.UpAxis` `attribute`), 136  
`ZPositive` (`pycozmo.protocol_encoder.UpAxis` `attribute`), 136