PyCozmo

Release 0.8.0

Nov 12, 2020

Documentation

1 Overview **1**
- 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

2 PyCozmo Architecture **5**
- 2.1 Overview . . . 5
- 2.2 Connection Layer . . . 7
- 2.3 Client Layer (SDK) . . . 7
- 2.4 Application Layer . . . 7

3 Cozmo Protocol **9**
- 3.1 Overview . . . 9
- 3.2 Network Setup . . . 9
- 3.3 Frames . . . 10
- 3.4 Packets . . . 10
- 3.5 Connection Establishment . . . 12

4 Capturing Cozmo Communication **13**
- 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

5 Cozmo Functions **19**
- 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

6 Cozmo Off-Board Functions **27**
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

7 Cozmo Firmware Versions **31**
7.1 Production Versions . . . 31
7.2 Factory Versions . . . 33

8 Cozmo Hardware Versions **35**
8.1 Hardware Version 4 . . . 35
8.2 Hardware Version 5 . . . 35
8.3 Hardware Version 6 . . . 35
8.4 Hardware Version 7 . . . 36

9 ESP8266 **37**
9.1 SPI Flash Memory Map . . . 37

10 pycozmo package **39**
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 . . . 70
10.16 pycozmo.emotions . . . 75
10.17 pycozmo.event . . . 76
10.18 pycozmo.exception . . . 79
10.19 pycozmo.filter . . . 80
10.20 pycozmo.frame . . . 80
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 . . . 85
10.26 pycozmo.protocol_base . . . 90
10.27 pycozmo.protocol_declaration . . . 92
10.28 pycozmo.protocol_encoder . . . 92
10.29 pycozmo.protocol_generator . . . 136
10.30 pycozmo.protocol_utils . . . 137
10.31 pycozmo.robot . . . 139
10.32 pycozmo.robot_debug . . . 141
10.33 pycozmo.run . . . 141
10.34 pycozmo.util . . . 142
10.35 pycozmo.window . . . 146

11 Indices and tables **149**

Python Module Index **151**

Index **153**

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.

```
+------------------+                    +------------------+                    +-------
↪-----------+
|     SDK app      |     Cozmo SDK      |    Cozmo app     |       Cozmo        |      ⌊
↪Cozmo       |
|      "game"      |     cozmoclad      |     "engine"     |      protocol      |
↪"robot"      |
|                  | -----------------> |   Wi-Fi client   | -----------------> |      ⌊
↪Wi-Fi AP     |
|                  |        USB         |    UDP client    |     UDP/Wi-Fi      |    ⌊
↪UDP Server    |
+------------------+                    +------------------+                    +-------
↪-----------+
```

1.3 Requirements

- Python 3.6.0 or newer
- Pillow 6.0.0 - Python image library
- FlatBuffers - serialization library
- dpkt - TCP/IP packet parsing 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.

CHAPTER 2

PyCozmo Architecture

2.1 Overview

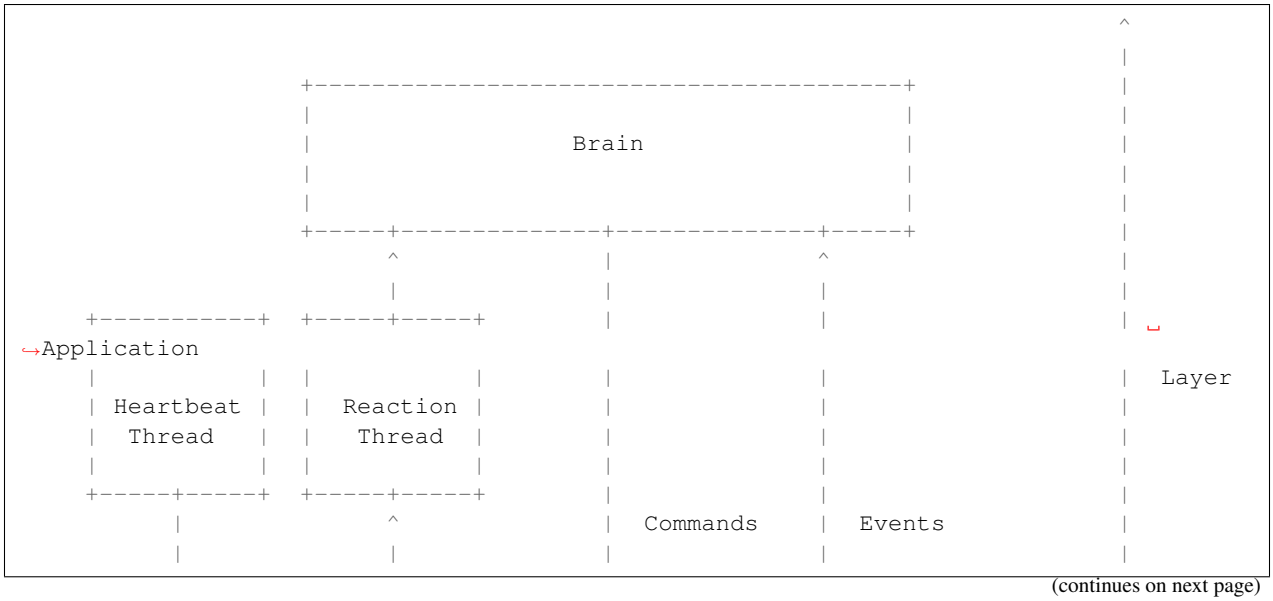
PyCozmo is designed as a multithreaded library.

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

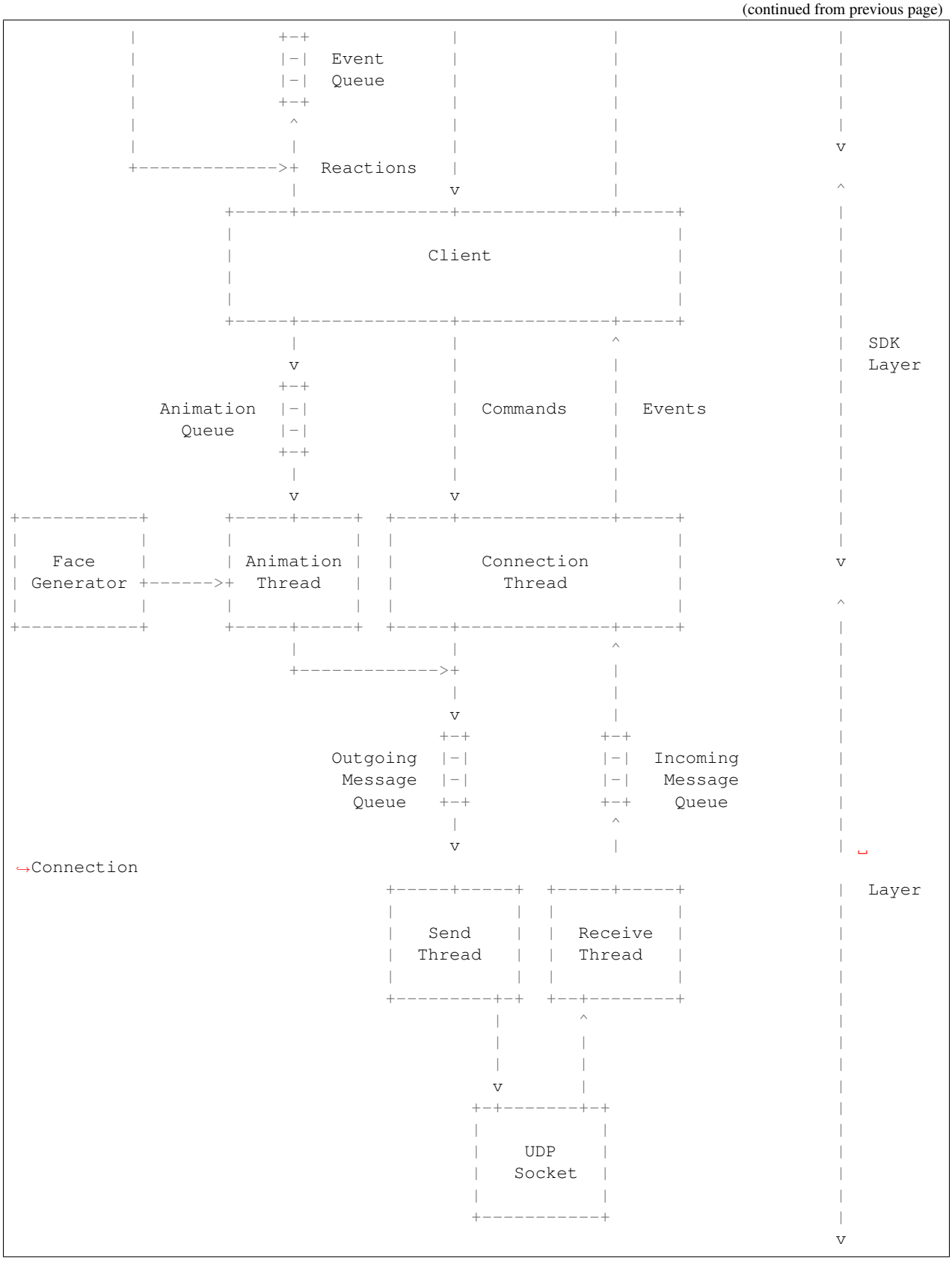
Each layer provides it's own API and can be used independently.

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.

CHAPTER 3

Cozmo Protocol

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.

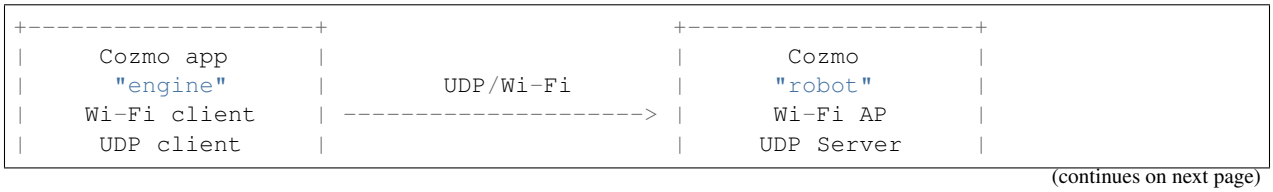
3.2 Network Setup

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.

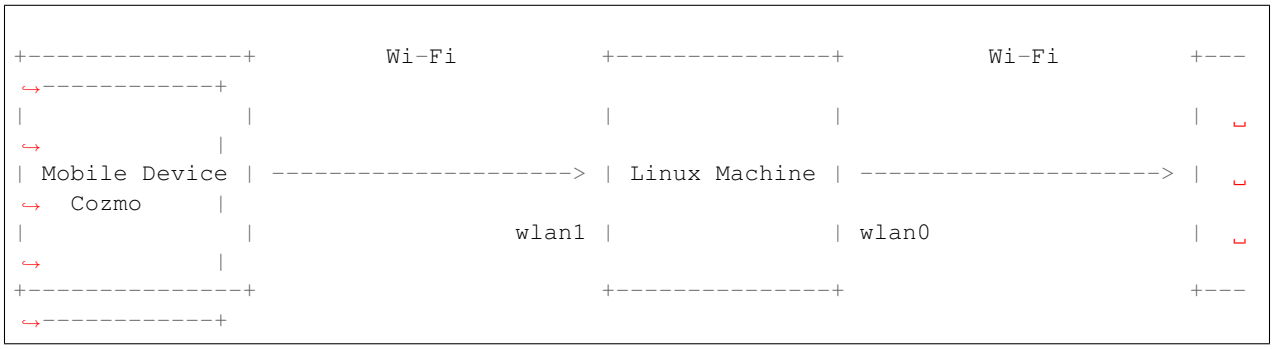
CHAPTER 4

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="XXXXXXXXXXXX"
}
```

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 dhcpdcd's configuration file:

```
$ sudo vi /etc/dhcpcd.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`.

CHAPTER 5

Cozmo Functions

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_mmps` and `rwheel_speed_mmps` fields of the `RobotState` message.

In addition, the and `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 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` filed of the `RobotState` message has a `robot_pathing` flag that indicates whether the robot is currently traversing a path. The `curr_path_segment` filed 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 send 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 Cozmos 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 e `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 ata processing and provides basic events with the following messages:

- `ObjectMoved`
- `ObjectStoppedMoving`
- `ObjectUpAxisChnaged`
- `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.

CHAPTER 6

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

CHAPTER 7

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": "90d8f3273055624b8444fbcbef555ee8",
    "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": "5914fda0b97c7aadaf0e4d97fc72610f",
    "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
}
```

CHAPTER 8

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.
```

CHAPTER 9

ESP8266

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

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

Continued on next page

Table 1 – continued from previous page

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

10.1 pycozmo.audiokinetic.exception

AudioKinetic WWise exceptions.

Exceptions

AudioKineticBaseError	AudioKinetic WWise base error.
AudioKineticFormatError	Invalid file format error.
AudioKineticIOError	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)
- https://github.com/rickvg/Wwise-BNKExtract

Classes

Event(soundbank_id, event_id, name, action_ids)	AudioKinetic WWise Event.
EventAction(soundbank_id, ea_id, scope, ...)	AudioKinetic WWise Event Action.
File(soundbank_id, file_id, offset, length)	AudioKinetic WWise WEM File.
SFX(soundbank_id, sfx_id, name, location, ...)	AudioKinetic WWise sound effect/voice.
SoundBank()	AudioKinetic WWise SoundBank (.bnk) file representation class.
SoundBankReader(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*, *location: 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

load_soundbanksinfo(fspec, TextIO])	Load SoundbanksInfo.xml and return a dictionary of parsed Info objects.

Classes

EventInfo(soundbank_id, event_id, name, ...)	Event representation in SoundbanksInfo.xml .
FileInfo(soundbank_id, file_id, name, path, ...)	File representation in SoundbanksInfo.xml .
SoundBankInfo(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

Amazement(params, width, height)	
Anger(params, width, height)	
Annoyance(params, width, height)	
Asleep(params, width, height)	
Boredom(params, width, height)	
Confusion(params, width, height)	
Despair(params, width, height)	
Disappointment(params, width, height)	
Disgust(params, width, height)	
Embarrassment(params, width, height)	
Excitement(params, width, height)	
Fear(params, width, height)	
Fury(params, width, height)	aka "enragement".
Guilt(params, width, height)	
Happiness(params, width, height)	
Horror(params, width, height)	
Neutral(params, width, height)	
Pleading(params, width, height)	
Rejection(params, width, height)	
Sadness(params, width, height)	
Skepticism(params, width, height)	
Surprise(params, width, height)	
Suspicion(params, width, height)	
Tiredness(params, width, height)	
Vulnerability(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/stable/lib/python3.
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/stable/lib/python3.
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/stable/lib/python3.
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/stable/lib/python3.
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/stable/lib/python3.
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/stable/lib/python3.
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/stable/lib/python3.
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/stable/lib/python3.
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/stable/lib/python3.
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/stable/lib/python3.
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/stable/lib/python3.
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/stable/lib/python3.
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/stable/lib/python3.
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/stable/lib/python3.
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/stable/lib/python3.
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/stable/lib/python3.
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/stable/lib/python3.
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/stable/lib/python3.
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/stable/lib/python3.
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/stable/lib/python3.
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/stable/lib/python3.
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/stable/lib/python3.
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/stable/lib/python3.
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/stable/lib/python3.
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/stable/lib/python3.
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

from_dict(info, VT])	
load_activities(resource_dir)	Load activity map from cozmo resources.

Classes

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

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*) → pycozmo.anim.PreprocessedClip

classmethod keyframe_to_im(*keyframe*) → <module 'PIL.Image' from '/home/docs/checkouts/readthedocs.org/user_builds/pycozmo/envs/stable/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, pycozmo.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, pycozmo.anim.BackpackAnimation]

10.8 pycozmo.anim_controller

Animation controller for audio, image, and animation playback.

Classes

AnimationController(cli)	Animation controller class.
AnimationQueue()	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], image_pkt: Optional[pycozmo.protocol_encoder.DisplayImage], pkts: Optional[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], image_pkt: Optional[pycozmo.protocol_encoder.DisplayImage], pkts: Optional[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

get_clip_metadata(dspec)	Retrieve clip metadata from animation FlatBuffers .bin files.

Classes

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

Continued on next page

Table 13 – continued from previous page

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

to_dict() → dict

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

class pycozmo.anim_encoder.**AnimTurnToRecordedHeading**(*trigger_time_ms: int = 0*, *duration_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*) → pycozmo.anim_encoder.AnimBodyMotion

to_dict() → dict

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

class pycozmo.anim_encoder.**AnimBackpackLights**(*trigger_time_ms: int = 0*, *duration_ms: int = 0*, *left: pycozmo.anim_encoder.AnimLight = <pycozmo.anim_encoder.AnimLight object>*, *front: pycozmo.anim_encoder.AnimLight = <pycozmo.anim_encoder.AnimLight object>*, *middle: pycozmo.anim_encoder.AnimLight = <pycozmo.anim_encoder.AnimLight object>*, *back: pycozmo.anim_encoder.AnimLight = <pycozmo.anim_encoder.AnimLight object>*, *right: pycozmo.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*) → pycozmo.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*) → pycozmo.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*) → pycozmo.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: Iterable[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*) → pycozmo.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*) → pycozmo.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, pycozmo.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
- http://dystopiancode.blogspot.com/2012/02/pcm-law-and-u-law-companding-algorithms.html

Functions

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

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

Continued on next page

Table 15 – continued from previous page

load_reaction_trigger_behavior_map(resource_dir)	

Classes

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

pycozmo.behavior.**load_reaction_trigger_behavior_map**(*resource_dir: str*) → Dict[str, pycozmo.behavior.ReactionTrigger]

10.12 pycozmo.brain

Brain class - high level behavior and emotion engine.

Classes

Brain(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: pycozmo.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_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

mini_to_jpeg_helper(mini, width, height, header)	Low-level mini*ToJpeg format to normal JPEG format conversion.
minicolor_to_jpeg(minicolor, width, height)	Converts miniColorToJpeg format to normal JPEG format.

Continued on next page

Table 18 – continued from previous page

minigray_to_jpeg(minigray, width, height)	Converts miniGrayToJpeg format to normal JPEG format.

pycozmo.camera.**RESOLUTIONS = {<ImageResolution.VerificationSnapshot: 0>: (16, 16), <Image**
Camera resolutions.

pycozmo.camera.**minigray_to_jpeg**(*minigray*, *width*, *height*)
Converts miniGrayToJpeg format to normal JPEG format.

pycozmo.camera.**minicolor_to_jpeg**(*minicolor*, *width*, *height*)
Converts miniColorToJpeg format to normal JPEG format.

10.14 pycozmo.client

Cozmo protocol client and high-level API.

Classes

Client(robot_addr, int]] = None, ...)	Cozmo protocol client and high-level API class.

class pycozmo.client.**Client**(*robot_addr: Optional[Tuple[str, int]] = None*, *protocol_log_messages: Optional[list] = None*, *auto_initialize: bool = True*, *enable_animations: bool = True*, *enable_procedural_face: bool = True*)
Bases: *pycozmo.event.Dispatcher*

Cozmo protocol client and high-level API class.

activate_behavior(*behavior*)

add_child_dispatcher(*child*)

add_handler(*event*, *f*, *one_shot=False*)

anim_names

cancel_anim() → None

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/stable/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*)

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

10.15 pycozmo.conn

Cozmo protocol low-level client and server connection.

Classes

Connection(robot_addr, int]] = None, ...)	Cozmo protocol low-level connection implementing bot client and server sides.
ReceiveThread(sock, send_thread, ...)	Cozmo protocol connection receive thread.

Continued on next page

Table 20 – continued from previous page

SendThread(sock, receiver_address, int]])	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, protocol_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

load_emotion_events(resource_dir)	
load_emotion_types(resource_dir)	

Classes

DecayGraph(nodes)	

Continued on next page

Table 22 – continued from previous page

EmotionEvent(name, affectors, float])	EmotionEvent representation class.
EmotionType(name, decay_graph, repetition_penaly)	Emotion type class.
Node(x, y)	

class pycozmo.emotions.**EmotionType**(*name: str*, *decay_graph: pycozmo.emotions.DecayGraph*, *repetition_penaly: pycozmo.emotions.DecayGraph*)

Bases: object

Emotion type class.

decay_graph

name

repetition_penalty

update()

Update from decay function.

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, pycozmo.emotions.EmotionType]

pycozmo.emotions.**load_emotion_events**(*resource_dir: str*) → Dict[str, pycozmo.emotions.EmotionEvent]

10.17 pycozmo.event

Event declaration and dispatching.

Classes

Dispatcher()	Event dispatcher class.
Event	Base class for events.
EvtAnimationCompleted	
EvtAudioCompleted	
EvtBehaviorDone	
EvtChargerOOSChange	
EvtCliffDetectedChange	
EvtNewRawCameraImage	Triggered when a new raw image is received from the robot's camera.
EvtPacketReceived	Triggered when a new packet has been received from the robot.
EvtReactionTrigger	

Continued on next page

Table 23 – continued from previous page

EvtRobotAnimBufferFullChange	
EvtRobotAnimatingChange	
EvtRobotAnimatingIdleChange	
EvtRobotBodyAccModeChange	
EvtRobotCarryingBlockChange	
EvtRobotChargingChange	
EvtRobotFallingChange	
EvtRobotFound	Triggered when the robot has been first connected.
EvtRobotHeadInPositionChange	
EvtRobotLiftInPositionChange	
EvtRobotMovingChange	
EvtRobotOnChargerChange	
EvtRobotOrientationChange	Triggered when the robot orientation changes.
EvtRobotPathingChange	
EvtRobotPickedUpChange	
EvtRobotPickingOrPlacingChange	
EvtRobotReady	Triggered when the robot has been initialized and is ready for commands.
EvtRobotStateUpdated	Triggered when a new robot state is received.
EvtRobotWheelsMovingChange	
Handler(f, one_shot)	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.
NoSpace	Out of space.
PyCozmoConnectionError	Base class for all PyCozmo connection exceptions.
PyCozmoException	Base class for all PyCozmo exceptions.
ResourcesNotFound	Cozmo resources not found.
Timeout	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

Frame(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*) → pycozmo.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/stable/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/stable/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_frame**
Green light.

pycozmo.lights.**red_light = LightState(on_color=31744, off_color=31744, on_frames=0, off_fra**
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_**
White light.

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

10.23 pycozmo.object

Cozmo objects (cubes, platforms, etc.).

Classes

Object(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

interpolate(from_face, to_face, steps)	Given two ProceduralFace objects, generate interpolated ProceduralFace objects in a number of steps.

Classes

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

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/stable/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/stable/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/stable/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

Argument(name, description, default)	Base class for packet arguments.
BoolArgument(name, description, default)	8-bit boolean.
Command(packet_id, name, group, description, ...)	Command packet.
Connect(description)	Connect packet.

Continued on next page

Table 33 – continued from previous page

Disconnect(description)	Disconnect packet.
DoubleArgument(name, description, default)	64-bit floating point number.
Enum(name, description, members, base)	Enumeration.
EnumArgument(name, enum_type, description, ...)	Enumeration argument.
EnumMember(name, value, description)	Enumeration member.
Event(packet_id, name, group, description, ...)	Event packet.
FArrayArgument(name, description, data_type, ...)	Fixed-length array.
FloatArgument(name, description, default)	32-bit floating point number.
FrameType	Frame type enumeration.
Int16Argument(name, description, default)	16-bit signed integer.
Int32Argument(name, description, default)	32-bit signed integer.
Int8Argument(name, description, default)	8-bit signed integer.
IntArgument(name, description, default)	Base class for signed integers.
Keyframe(description)	Keyframe packet.
Packet(packet_type, name, packet_id, group, ...)	Base class for packets.
PacketType	Packet type enumeration.
Ping(description)	Ping packet.
Protocol(enums, structs, packets)	Protocol declaration.
StringArgument(name, description, ...)	String.
Struct(name, description, arguments)	Structure.
UInt16Argument(name, description, default)	16-bit unsigned integer.
UInt32Argument(name, description, default)	32-bit unsigned integer.
UInt8Argument(name, description, default)	8-bit unsigned integer.
UIntArgument(name, description, default)	Base class for unsigned integers.
VArrayArgument(name, description, data_type, ...)	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*, *description: Optional[str] = None*, *arguments: Optional[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: Optional[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: Optional[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: Optional[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

Packet(packet_type, packet_id)	
Struct	
UnknownCommand(packet_id, data)	
UnknownEvent(packet_id, data)	
UnknownPacket(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*) → pycozmo.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*) → pycozmo.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

AbortAnimation()	
AcknowledgeAction([action_id])	
AnimBackpackLights([colors])	
AnimBody([speed, unknown])	
AnimHead([duration_ms, variability_deg, ...])	
AnimLift([duration_ms, variability_mm, ...])	
AnimationEnded([anim_id])	
AnimationStarted([anim_id])	
AnimationState([timestamp, ...])	
AppendPathSegArc([center_x, center_y, ...])	
AppendPathSegLine([from_x, from_y, to_x, ...])	
AppendPathSegPointTurn([x, y, angle_rad, ...])	
BodyColor	An enumeration.
BodyInfo([serial_number, body_hw_version, ...])	
ButtonPressed([pressed])	
ClearPath([unknown])	
Connect()	
CubeId([object_id, rotation_period_frames])	
CubeLights([states])	
DebugData([format_id, unused, name_id, ...])	
Disconnect()	
DisplayImage([image])	
DriveWheels([lwheel_speed_mmps, ...])	
Enable()	
EnableAnimationState()	
EnableCamera([image_send_mode, image_resolution])	
EnableColorImages([enable])	
EnableStopOnCliff([enable])	
EndAnimation()	
ExecutePath([event_id, unknown])	
FallingStarted([unknown])	
FallingStopped([unknown, duration_ms, ...])	
FirmwareSignature([unknown, signature])	
FirmwareUpdate([chunk_id, data])	
FirmwareUpdateResult([byte_count, chunk_id, ...])	
HardwareInfo([serial_number_head, unknown1, ...])	
ImageChunk([frame_timestamp, image_id, ...])	
ImageEncoding	An enumeration.
ImageImuData([image_id, rate_x, rate_y, ...])	
ImageResolution	An enumeration.
ImageSendMode	An enumeration.
Keyframe()	
LightState([on_color, off_color, on_frames, ...])	
LightStateCenter([states, unknown])	
LightStateSide([states, unknown])	
MotorCalibration([motor_id, calib_started, ...])	
MotorID	An enumeration.
MoveHead([speed_rad_per_sec])	

Continued on next page

Table 35 – continued from previous page

MoveLift([speed_rad_per_sec])	
NvEntryTag	An enumeration.
NvOperation	An enumeration.
NvResult	An enumeration.
NvStorageOp([tag, length, op, unknown, data])	
NvStorageOpResult([tag, length, op, result, ...])	
ObjectAccel([timestamp, object_id, accel_x, ...])	
ObjectAvailable([factory_id, object_type, rssi])	
ObjectConnect([factory_id, connect])	
ObjectConnectionState([object_id, ...])	
ObjectMoved([timestamp, object_id, ...])	
ObjectPowerLevel([object_id, ...])	
ObjectStoppedMoving([timestamp, object_id])	
ObjectTapFiltered([timestamp, object_id, ...])	
ObjectTapped([timestamp, object_id, ...])	
ObjectType	An enumeration.
ObjectUpAxisChanged([timestamp, object_id, axis])	
OutputAudio([samples])	
OutputSilence()	
PathEventType	An enumeration.
PathFollowingEvent([event_id, event_type])	
PathSegmentSpeed([speed_mmps, accel_mmps2, ...])	
Ping([time_sent_ms, counter, last, unknown])	
RecordHeading()	
RobotDelocalized()	
RobotPoked()	
RobotState([timestamp, pose_frame_id, ...])	
SetAccessoryDiscovery([enable])	
SetCameraParams([gain, exposure_ms, ...])	
SetHeadAngle([angle_rad, ...])	
SetHeadLight([enable])	
SetLiftHeight([height_mm, ...])	
SetOrigin([unknown0, pose_frame_id, ...])	
SetRobotVolume([level])	
ShutdownRobot()	
StartAnimation([anim_id])	
StartMotorCalibration([head, lift])	
StopAllMotors()	
StreamObjectAccel([object_id, enable])	
SyncTime([timestamp, unknown])	
TrimPath([head, tail])	
TurnInPlace([angle_rad, speed_rad_per_sec, ...])	
TurnInPlaceAtSpeed([wheel_speed_mmps, ...])	
TurnToRecordedHeading()	
UpAxis	An enumeration.
WifiOff([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.**AnimLift**(*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*, *tag=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_mmps=0.0*, *accel_mmps2=0.0*, *decel_mmps2=0.0*)

Bases: *pycozmo.protocol_base.Packet*

accel_mmps2

ack

center_x

center_y

decel_mmps2

classmethod from_bytes(*buffer*)

classmethod from_reader(*reader*)

id

is_from_engine() → bool

is_from_robot() → bool

is_oob() → bool

radius_mm

seq

speed_mmps

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_mmps=0.0*, *accel_mmps2=0.0*, *decel_mmps2=0.0*)

Bases: *pycozmo.protocol_base.Packet*

accel_mmps2

ack

decel_mmps2

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*, *accel_mmps2=0.0*, *decel_mmps2=0.0*, *unknown=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_mmps2=0.0*, *rwheel_accel_mmps2=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, impact_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*, *status=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

NVEntry_CliffValOnGround = 2147483656

NVEntry_FACTORY_RESERVED1 = 1835008

NVEntry_FACTORY_RESERVED2 = 1957888

NVEntry_FaceAlbumData = 1589248

NVEntry_FaceEnrollData = 1585152

NVEntry_FactoryBaseTag = 909312

NVEntry_FactoryBaseTagWithBCOffset = 909360

NVEntry_FactoryLock = 2147483665

NVEntry_GameSkillLevels = 1572864

NVEntry_GameUnlocks = 1581056

NVEntry_IMUAverages = 3221225476

NVEntry_IMUInfo = 2147483654

NVEntry_Invalid = 4294967295

NVEntry_InventoryData = 1658880

NVEntry_LabAssignments = 1662976

NVEntry_NEXT_SLOT = 1671168

NVEntry_NurtureGameData = 1654784

NVEntry_ObservedCubePose = 2147483653

NVEntry_OnboardingData = 1576960

NVEntry_PlaypenTestResults = 2147483664

NVEntry_PrePlaypenCentroids = 3221225473

NVEntry_PrePlaypenResults = 3221225472

NVEntry_SavedCubeIDs = 1667072

NVEntry_ToolCodeImageLeft = 2148532224

NVEntry_ToolCodeImageRight = 2148597760

NVEntry_ToolCodeInfo = 2147483650

NVEntry_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, unknown=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*, *connected=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*, *active_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*, *battery_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_mmps=0.0*, *accel_mmps2=0.0*, *decel_mmps2=0.0*)

Bases: *pycozmo.protocol_base.Struct*

accel_mmps2

decel_mmps2

classmethod from_bytes(*buffer*)

classmethod from_reader(*reader*)

speed_mmps

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_mmps=0.0*, *rwheel_speed_mmps=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*, *battery_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*, *accel_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*, *accel_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*, *direction=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

get_enum_fmt(argument)	
get_farray_fmt(argument)	
get_fmt_by_type(t)	
get_string_fmt(argument)	
get_varray_fmts(argument)	
int_to_str(value, base)	

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

get_farray_size(fmt, length)	Figures out the size of a fixed array with given format.
get_object_farray_size(value, length)	Figures out the size of a given fixed-length object sequence.
get_object_size(value)	Figures out the size of a given object.
get_size(fmt)	Figures out the size of a value with the given format.
get_string_size(value, length_format)	Figures out the size of a string with given length format.
get_varray_size(value, length_format, ...)	Figures out the size of a variable-length array with given format.
validate_bool(name, value)	
validate_farray(name, value, length, ...)	
validate_float(name, value)	
validate_integer(name, value, minimum, maximum)	
validate_object(name, value, expected_type)	
validate_string(name, value, maximum_length)	
validate_varray(name, value, maximum_length, ...)	

Classes

BinaryReader(buffer, offset)	Used to read in a stream of binary data, keeping track of the current position.
BinaryWriter()	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

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

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 mmps>**
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_PI**
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

get_debug_message(name_id, format_id, args)	Generate a log message from robot debug name and format IDs.
get_log_level(robot_level)	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

connect(log_level, protocol_log_level, ...)	
setup_basic_logging(log_level, ... [, target])	

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

angle_z_to_quaternion(angle_z)	Converts an angle in the z axis (Euler angle z component) to a quaternion.
check_assets()	Check whether Cozmo assets are available.
frange(start, stop, step)	
get_cozmo_anim_dir()	Get Cozmo animation asset directory.
get_cozmo_asset_dir()	Get Cozmo asset directory.
get_pycozmo_dir()	Get PyCozmo directory.
hex_dump(data)	
hex_load(data)	

Classes

Angle(radians, degrees)	Angle representation.
Distance(mm, inches)	Represents a distance.
FPSTimer(fps)	A timer that maintains frame rate by sleeping for a variable amount of time.
Matrix44(m00, m10, m20, m30, m01, m11, m21, ...)	A 4x4 Matrix for representing the rotation and/or position of an object in the world.
Pose(x, y, z, q0, q1, q2, q3, angle_z, ...)	A combination of position (vector) and rotation (quaternion).
Quaternion(q0, q1, q2, q3, angle_z)	Represents rotation.
Speed(mmps)	Speed representation.
Vector3(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.**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*) → str

pycozmo.util.**hex_load**(*data: str*) → bytes

pycozmo.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

BaseWindow(seq_bits, size, max_seq)	Base communication window class.
ReceiveWindow(seq_bits, size, max_seq)	Receive communication window class.
SendWindow(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

Python Module Index

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, 70
pycozmo.emotions, 75
pycozmo.event, 76
pycozmo.exception, 79
pycozmo.expressions.expressions, 44
pycozmo.filter, 80
pycozmo.frame, 80
pycozmo.image_encoder, 81
pycozmo.lights, 82
pycozmo.object, 83
pycozmo.procedural_face, 83
pycozmo.protocol_ast, 85
pycozmo.protocol_base, 90
pycozmo.protocol_declaration, 92
pycozmo.protocol_encoder, 92
pycozmo.protocol_generator, 136
pycozmo.protocol_utils, 137
pycozmo.robot, 139
pycozmo.robot_debug, 141
pycozmo.run, 141
pycozmo.util, 142
pycozmo.window, 146

Index

A

AbortAnimation (*class in pycozmo.protocol_encoder*), 94
abs_value (*pycozmo.util.Angle attribute*), 142
accel_mmps2 (*pycozmo.protocol_encoder.AppendPathSegArc attribute*), 98
accel_mmps2 (*pycozmo.protocol_encoder.AppendPathSegLine attribute*), 99
accel_mmps2 (*pycozmo.protocol_encoder.AppendPathSegPointTurn attribute*), 100
accel_mmps2 (*pycozmo.protocol_encoder.PathSegmentSpeed attribute*), 125
accel_rad_per_sec2 (*pycozmo.protocol_encoder.SetHeadAngle attribute*), 129
accel_rad_per_sec2 (*pycozmo.protocol_encoder.SetLiftHeight attribute*), 130
accel_rad_per_sec2 (*pycozmo.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*), 80
ack (*pycozmo.protocol_base.Packet attribute*), 91
ack (*pycozmo.protocol_base.UnknownCommand attribute*), 91
ack (*pycozmo.protocol_base.UnknownEvent attribute*), 92
ack (*pycozmo.protocol_base.UnknownPacket attribute*), 91
ack (*pycozmo.protocol_encoder.AbortAnimation attribute*), 94
ack (*pycozmo.protocol_encoder.AcknowledgeAction attribute*), 95
ack (*pycozmo.protocol_encoder.AnimationEnded attribute*), 97
ack (*pycozmo.protocol_encoder.AnimationStarted attribute*), 97
ack (*pycozmo.protocol_encoder.AnimationState attribute*), 98
ack (*pycozmo.protocol_encoder.AnimBackpackLights attribute*), 95
ack (*pycozmo.protocol_encoder.AnimBody attribute*), 96
ack (*pycozmo.protocol_encoder.AnimHead attribute*), 96
ack (*pycozmo.protocol_encoder.AnimLift attribute*), 96
ack (*pycozmo.protocol_encoder.AppendPathSegArc attribute*), 98
ack (*pycozmo.protocol_encoder.AppendPathSegLine attribute*), 99
ack (*pycozmo.protocol_encoder.AppendPathSegPointTurn attribute*), 100
ack (*pycozmo.protocol_encoder.BodyInfo attribute*), 100
ack (*pycozmo.protocol_encoder.ButtonPressed attribute*), 101
ack (*pycozmo.protocol_encoder.ClearPath attribute*), 101
ack (*pycozmo.protocol_encoder.Connect attribute*), 102
ack (*pycozmo.protocol_encoder.CubeId attribute*), 102
ack (*pycozmo.protocol_encoder.CubeLights attribute*), 102
ack (*pycozmo.protocol_encoder.DebugData attribute*), 103
ack (*pycozmo.protocol_encoder.Disconnect attribute*), 103
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*), 105
ack (*pycozmo.protocol_encoder.EnableColorImages attribute*), 106
ack (*pycozmo.protocol_encoder.EnableStopOnCliff attribute*), 106
ack (*pycozmo.protocol_encoder.EndAnimation attribute*), 106
ack (*pycozmo.protocol_encoder.ExecutePath attribute*), 107
ack (*pycozmo.protocol_encoder.FallingStarted attribute*), 107
ack (*pycozmo.protocol_encoder.FallingStopped attribute*), 108
ack (*pycozmo.protocol_encoder.FirmwareSignature attribute*), 108
ack (*pycozmo.protocol_encoder.FirmwareUpdate attribute*), 108
ack (*pycozmo.protocol_encoder.FirmwareUpdateResult attribute*), 109
ack (*pycozmo.protocol_encoder.HardwareInfo attribute*), 109
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*), 113
ack (*pycozmo.protocol_encoder.MoveHead attribute*), 114
ack (*pycozmo.protocol_encoder.MoveLift attribute*), 114
ack (*pycozmo.protocol_encoder.NvStorageOp attribute*), 117
ack (*pycozmo.protocol_encoder.NvStorageOpResult attribute*), 117
ack (*pycozmo.protocol_encoder.ObjectAccel attribute*), 118
ack (*pycozmo.protocol_encoder.ObjectAvailable attribute*), 118
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*), 121
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*), 124
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*), 126
ack (*pycozmo.protocol_encoder.RobotState attribute*), 127
ack (*pycozmo.protocol_encoder.SetAccessoryDiscovery attribute*), 128
ack (*pycozmo.protocol_encoder.SetCameraParams attribute*), 128
ack (*pycozmo.protocol_encoder.SetHeadAngle attribute*), 129
ack (*pycozmo.protocol_encoder.SetHeadLight attribute*), 129
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*), 131
ack (*pycozmo.protocol_encoder.StartMotorCalibration attribute*), 132
ack (*pycozmo.protocol_encoder.StopAllMotors attribute*), 132
ack (*pycozmo.protocol_encoder.StreamObjectAccel attribute*), 133
ack (*pycozmo.protocol_encoder.SyncTime attribute*), 133
ack (*pycozmo.protocol_encoder.TrimPath attribute*), 133
ack (*pycozmo.protocol_encoder.TurnInPlace attribute*), 134
ack (*pycozmo.protocol_encoder.TurnInPlaceAtSpeed at-*

tribute), 134
ack (*pycozmo.protocol_encoder.TurnToRecordedHeading attribute*), 135
ack (*pycozmo.protocol_encoder.WifiOff attribute*), 135
ack() (*pycozmo.conn.SendThread method*), 72
ACK_TIMEOUT (*pycozmo.conn.SendThread attribute*), 72
acknowledge() (*pycozmo.window.SendWindow method*), 147
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_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*), 80
Amazement (*class in pycozmo.expressions.expressions*), 57
Anger (*class in pycozmo.expressions.expressions*), 45
Angle (*class in pycozmo.util*), 142
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*), 83
angle (*pycozmo.robot.LiftPosition attribute*), 141
angle_deg (*pycozmo.protocol_encoder.AnimHead attribute*), 96
angle_offset (*pycozmo.procedural_face.ProceduralLid attribute*), 83
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*), 145
angle_z_to_quaternion() (*in module pycozmo.util*), 144
anim_id (*pycozmo.protocol_encoder.AnimationEnded attribute*), 97
anim_id (*pycozmo.protocol_encoder.AnimationStarted attribute*), 97
anim_id (*pycozmo.protocol_encoder.StartAnimation attribute*), 131
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*), 97
AnimationState (*class in pycozmo.protocol_encoder*), 98
AnimBackpackLights (*class in pycozmo.anim_encoder*), 64
AnimBackpackLights (*class in pycozmo.protocol_encoder*), 95
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*), 96
AnimLiftHeight (*class in pycozmo.anim_encoder*), 63
AnimLiftHeight (*class in pycozmo.anim_encoder*), 63
AnimProceduralFace (*class in pycozmo.anim_encoder*), 65
AnimRecordHeading (*class in pycozmo.anim_encoder*), 63
AnimRobotAudio (*class in pycozmo.anim_encoder*), 65
AnimTurnToRecordedHeading (*class in pycozmo.anim_encoder*), 64
Annoyance (*class in pycozmo.expressions.expressions*), 52
AppendPathSegArc (*class in pycozmo.protocol_encoder*), 98
AppendPathSegLine (*class in pycozmo.protocol_encoder*), 99
AppendPathSegPointTurn (*class in pycozmo.protocol_encoder*), 99
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*), 79
args (*pycozmo.exception.NoSpace attribute*), 80
args (*pycozmo.exception.PyCozmoConnectionError attribute*), 79
args (*pycozmo.exception.PyCozmoException attribute*), 79
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 (*pycozmo.protocol_encoder.SetCameraParams*

attribute), 128
auto_started (*pycozmo.protocol_encoder.MotorCalibration attribute*), 113
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*), 146
battery_level (*pycozmo.protocol_encoder.ObjectPowerLevel attribute*), 120
battery_voltage (*pycozmo.protocol_encoder.RobotState attribute*), 127
BAYER (*pycozmo.protocol_encoder.ImageEncoding attribute*), 110
Behavior (*class in pycozmo.behavior*), 67
behavior_id (*pycozmo.behavior.ReactionTrigger attribute*), 67
bend (*pycozmo.procedural_face.ProceduralLid attribute*), 83
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*), 82
blue_light (*in module pycozmo.lights*), 82
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*), 100
BoolArgument (*class in pycozmo.protocol_ast*), 87
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*), 113
cancel_anim() (*pycozmo.anim_controller.AnimationController method*), 60
cancel_anim() (*pycozmo.client.Client method*), 69
CE_LM_v15 (*pycozmo.protocol_encoder.BodyColor attribute*), 100
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*), 98
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*), 98
Charger_Basic (*pycozmo.protocol_encoder.ObjectType attribute*), 122
check_assets() (*in module pycozmo.util*), 146
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*), 108
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*), 101
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*), 122
ClipMetadata (*class in pycozmo.anim_encoder*), 66
COLLECT_INTERVAL (*pycozmo.conn.SendThread attribute*), 72
CollisionObstacle (*pycozmo.protocol_encoder.ObjectType attribute*), 122
Color (*class in pycozmo.lights*), 82
colors (*pycozmo.protocol_encoder.AnimBackpackLights attribute*), 95
Command (*class in pycozmo.protocol_ast*), 89
COMMAND (*pycozmo.protocol_ast.PacketType attribute*), 86
Confusion (*class in pycozmo.expressions.expressions*), 56
Connect (*class in pycozmo.protocol_ast*), 89
Connect (*class in pycozmo.protocol_encoder*), 102
CONNECT (*pycozmo.protocol_ast.PacketType attribute*), 86
connect (*pycozmo.protocol_encoder.ObjectConnect attribute*), 119
connect() (*in module pycozmo.run*), 142
connect() (*pycozmo.client.Client method*), 69
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*), 73
ConnectionTimeout, 79
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*), 102
curr_path_segment (*pycozmo.protocol_encoder.RobotState attribute*), 127
CustomFixedObstacle (*pycozmo.protocol_encoder.ObjectType attribute*), 122
CustomType00 (*pycozmo.protocol_encoder.ObjectType attribute*), 122
CustomType01 (*pycozmo.protocol_encoder.ObjectType attribute*), 122
CustomType02 (*pycozmo.protocol_encoder.ObjectType attribute*), 122
CustomType03 (*pycozmo.protocol_encoder.ObjectType attribute*), 122
CustomType04 (*pycozmo.protocol_encoder.ObjectType attribute*), 122
CustomType05 (*pycozmo.protocol_encoder.ObjectType attribute*), 122
CustomType06 (*pycozmo.protocol_encoder.ObjectType attribute*), 122
CustomType07 (*pycozmo.protocol_encoder.ObjectType attribute*), 122
CustomType08 (*pycozmo.protocol_encoder.ObjectType attribute*), 122
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*), 111

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*), 91
data (*pycozmo.protocol_base.UnknownEvent attribute*), 92
data (*pycozmo.protocol_base.UnknownPacket attribute*), 91
data (*pycozmo.protocol_encoder.FirmwareUpdate attribute*), 108
data (*pycozmo.protocol_encoder.ImageChunk attribute*), 110

data (*pycozmo.protocol_encoder.NvStorageOp attribute*), 117
data (*pycozmo.protocol_encoder.NvStorageOpResult attribute*), 117
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*), 69
DebugData (*class in pycozmo.protocol_encoder*), 103
decay_graph (*pycozmo.emotions.EmotionType attribute*), 76
decel_mmps2 (*pycozmo.protocol_encoder.AppendPathSegArc attribute*), 98
decel_mmps2 (*pycozmo.protocol_encoder.AppendPathSegLine attribute*), 99
decel_mmps2 (*pycozmo.protocol_encoder.AppendPathSegPointTurn attribute*), 100
decel_mmps2 (*pycozmo.protocol_encoder.PathSegmentSpeed attribute*), 125
decode() (*pycozmo.image_encoder.ImageDecoder method*), 81
define_pose_relative_this() (*pycozmo.util.Pose method*), 145
degrees (*pycozmo.util.Angle attribute*), 143
del_all_handlers() (*pycozmo.behavior.Behavior method*), 67
del_all_handlers() (*pycozmo.client.Client method*), 69
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*), 69
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*), 69
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*), 80
Despair (*class in pycozmo.expressions.expressions*), 49
DEV (*pycozmo.protocol_encoder.BodyColor attribute*), 100
direction (*pycozmo.protocol_encoder.TurnInPlaceAtSpeed attribute*), 134
Disappointment (*class in pycozmo.expressions.expressions*), 50
Disconnect (*class in pycozmo.protocol_ast*), 89
Disconnect (*class in pycozmo.protocol_encoder*), 103
DISCONNECT (*pycozmo.protocol_ast.PacketType attribute*), 86
disconnect() (*pycozmo.client.Client method*), 69
disconnect() (*pycozmo.conn.Connection method*), 74
disconnect() (*pycozmo.conn.ReceiveThread method*), 71
Disgust (*class in pycozmo.expressions.expressions*), 47
dispatch() (*pycozmo.behavior.Behavior method*), 67
dispatch() (*pycozmo.client.Client method*), 69
dispatch() (*pycozmo.conn.Connection method*), 74
dispatch() (*pycozmo.event.Dispatcher method*), 79
Dispatcher (*class in pycozmo.event*), 78
display_image() (*pycozmo.anim_controller.AnimationController method*), 60
display_image() (*pycozmo.client.Client method*), 69
DisplayImage (*class in pycozmo.protocol_encoder*), 104
Distance (*class in pycozmo.util*), 143
DoubleArgument (*class in pycozmo.protocol_ast*), 87
drive_wheels() (*pycozmo.client.Client method*), 69
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*), 96
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.SetAccessoryDiscovery attribute*), 128
enable (*pycozmo.protocol_encoder.SetHeadLight attribute*), 129
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*), 69
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*), 105
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*), 81
EndAnimation (*class in pycozmo.protocol_encoder*), 106
ENGINE (*pycozmo.protocol_ast.FrameType attribute*), 86
ENGINE_ACT (*pycozmo.protocol_ast.FrameType attribute*), 86
Enum (*class in pycozmo.protocol_ast*), 87
EnumArgument (*class in pycozmo.protocol_ast*), 88
EnumMember (*class in pycozmo.protocol_ast*), 87
euler_angles (*pycozmo.util.Quaternion attribute*), 145
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*), 86
event_id (*pycozmo.protocol_encoder.ExecutePath attribute*), 107
event_id (*pycozmo.protocol_encoder.PathFollowingEvent attribute*), 124
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*), 78
EvtAudioCompleted (*class in pycozmo.event*), 78
EvtBehaviorDone (*class in pycozmo.event*), 78
EvtChargerOOSChange (*class in pycozmo.event*), 78
EvtCliffDetectedChange (*class in pycozmo.event*), 78
EvtNewRawCameraImage (*class in pycozmo.event*), 77
EvtPacketReceived (*class in pycozmo.event*), 77
EvtReactionTrigger (*class in pycozmo.event*), 78
EvtRobotAnimatingChange (*class in pycozmo.event*), 78
EvtRobotAnimatingIdleChange (*class in pycozmo.event*), 78
EvtRobotAnimBufferFullChange (*class in pycozmo.event*), 78
EvtRobotBodyAccModeChange (*class in pycozmo.event*), 77
EvtRobotCarryingBlockChange (*class in pycozmo.event*), 77
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*), 77
EvtRobotOnChargerChange (*class in pycozmo.event*), 78
EvtRobotOrientationChange (*class in pycozmo.event*), 78
EvtRobotPathingChange (*class in pycozmo.event*), 78
EvtRobotPickedUpChange (*class in pycozmo.event*), 77
EvtRobotPickingOrPlacingChange (*class in pycozmo.event*), 77
EvtRobotReady (*class in pycozmo.event*), 77
EvtRobotStateUpdated (*class in pycozmo.event*),

78
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*), 147
exposure_ms (*pycozmo.protocol_encoder.SetCameraParams attribute*), 128
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*), 83
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*), 46
eye_width (*pycozmo.expressions.expressions.Skepticism*

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*), 83
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*), 118
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*), 80
filter() (*pycozmo.filter.Filter method*), 80
FIN (*pycozmo.protocol_ast.FrameType attribute*), 86
FirmwareSignature (*class in pycozmo.protocol_encoder*), 108
FirmwareUpdate (*class in pycozmo.protocol_encoder*), 108
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*), 144
FPSTimer (*class in pycozmo.util*), 146
Frame (*class in pycozmo.frame*), 80
FRAME_RATE (*in module pycozmo.robot*), 140
frame_timestamp (*pycozmo.protocol_encoder.ImageChunk attribute*), 110
FrameType (*class in pycozmo.protocol_ast*), 86
frange() (*in module pycozmo.util*), 146
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*), 90
from_bytes() (*pycozmo.protocol_base.UnknownCommand class method*), 91
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*), 97
from_bytes() (*pycozmo.protocol_encoder.AnimationState class method*), 98
from_bytes() (*pycozmo.protocol_encoder.AnimBackpackLights class method*), 95
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.AppendPathSegArc class method*), 98
from_bytes() (*pycozmo.protocol_encoder.AppendPathSegLine class method*), 99
from_bytes() (*pycozmo.protocol_encoder.AppendPathSegPoint 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*), 101
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*), 102
from_bytes() (*pycozmo.protocol_encoder.DebugData class method*), 103
from_bytes() (*pycozmo.protocol_encoder.Disconnect class method*), 103
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*), 105
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*), 106
from_bytes() (*pycozmo.protocol_encoder.ExecutePath class method*), 107
from_bytes() (*pycozmo.protocol_encoder.FallingStarted class method*), 107
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*), 108
from_bytes() (*pycozmo.protocol_encoder.FirmwareUpdateResult class method*), 109
from_bytes() (*pycozmo.protocol_encoder.HardwareInfo class method*), 109
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*), 112
from_bytes() (*pycozmo.protocol_encoder.LightStateCenter class method*), 113
from_bytes() (*pycozmo.protocol_encoder.LightStateSide*

class method), 113
from_bytes() (pycozmo.protocol_encoder.MotorCalibration class method), 113
from_bytes() (pycozmo.protocol_encoder.MoveHead class method), 114
from_bytes() (pycozmo.protocol_encoder.MoveLift class method), 114
from_bytes() (pycozmo.protocol_encoder.NvStorageOp class method), 117
from_bytes() (pycozmo.protocol_encoder.NvStorageOpResult class method), 117
from_bytes() (pycozmo.protocol_encoder.ObjectAccel class method), 118
from_bytes() (pycozmo.protocol_encoder.ObjectAvailable class method), 118
from_bytes() (pycozmo.protocol_encoder.ObjectConnected class method), 119
from_bytes() (pycozmo.protocol_encoder.ObjectConnectionState 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.ObjectTapFiltered class method), 121
from_bytes() (pycozmo.protocol_encoder.ObjectTapped class method), 121
from_bytes() (pycozmo.protocol_encoder.ObjectUpAxisChanged class method), 123
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), 126
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), 128
from_bytes() (pycozmo.protocol_encoder.SetHeadAngle class method), 129
from_bytes() (pycozmo.protocol_encoder.SetHeadLight class method), 129
from_bytes() (pycozmo.protocol_encoder.SetLiftHeight class method), 130
from_bytes() (pycozmo.protocol_encoder.SetOrigin class method), 130
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), 131
from_bytes() (pycozmo.protocol_encoder.StartMotorCalibration class method), 132
from_bytes() (pycozmo.protocol_encoder.StopAllMotors class method), 132
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), 133
from_bytes() (pycozmo.protocol_encoder.TurnInPlace class method), 134
from_bytes() (pycozmo.protocol_encoder.TurnInPlaceAtSpeed class method), 134
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*), 90
from_reader() (*pycozmo.protocol_base.UnknownCommand class method*), 91
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*), 97
from_reader() (*pycozmo.protocol_encoder.AnimationState class method*), 98
from_reader() (*pycozmo.protocol_encoder.AnimBackpackLights class method*), 95
from_reader() (*pycozmo.protocol_encoder.AnimBody class method*), 96
from_reader() (*pycozmo.protocol_encoder.AnimHead class method*), 96
from_reader() (*pycozmo.protocol_encoder.AnimLift class method*), 97
from_reader() (*pycozmo.protocol_encoder.AppendPathSegArc class method*), 98
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

from_reader() (pycozmo.protocol_encoder.ButtonPressed class method), 101
from_reader() (pycozmo.protocol_encoder.ClearPath class method), 101
from_reader() (pycozmo.protocol_encoder.Connect class method), 102
from_reader() (pycozmo.protocol_encoder.CubeId class method), 102
from_reader() (pycozmo.protocol_encoder.CubeLights class method), 102
from_reader() (pycozmo.protocol_encoder.DebugData class method), 103
from_reader() (pycozmo.protocol_encoder.Disconnect class method), 103
from_reader() (pycozmo.protocol_encoder.DisplayImage class method), 104
from_reader() (pycozmo.protocol_encoder.DriveWheels class method), 104
from_reader() (pycozmo.protocol_encoder.Enable class method), 105
from_reader() (pycozmo.protocol_encoder.EnableAnimationState class method), 105
from_reader() (pycozmo.protocol_encoder.EnableCamera class method), 105
from_reader() (pycozmo.protocol_encoder.EnableColorImages class method), 106
from_reader() (pycozmo.protocol_encoder.EnableStopOnCliff class method), 106
from_reader() (pycozmo.protocol_encoder.EndAnimation class method), 106
from_reader() (pycozmo.protocol_encoder.ExecutePath class method), 107
from_reader() (pycozmo.protocol_encoder.FallingStarted class method), 107
from_reader() (pycozmo.protocol_encoder.FallingStopped class method), 108
from_reader() (pycozmo.protocol_encoder.FirmwareSignature class method), 108
from_reader() (pycozmo.protocol_encoder.FirmwareUpdate class method), 109
from_reader() (pycozmo.protocol_encoder.FirmwareUpdateResult class method), 109
from_reader() (pycozmo.protocol_encoder.HardwareInfo class method), 109
from_reader() (pycozmo.protocol_encoder.ImageChunk class method), 110
from_reader() (pycozmo.protocol_encoder.ImageImuData class method), 111
from_reader() (pycozmo.protocol_encoder.Keyframe class method), 112
from_reader() (pycozmo.protocol_encoder.LightState class method), 112
from_reader() (pycozmo.protocol_encoder.LightStateCenter class method), 113
from_reader() (pycozmo.protocol_encoder.LightStateSide class method), 113
from_reader() (pycozmo.protocol_encoder.MotorCalibration class method), 114
from_reader() (pycozmo.protocol_encoder.MoveHead class method), 114
from_reader() (pycozmo.protocol_encoder.MoveLift class method), 114
from_reader() (pycozmo.protocol_encoder.NvStorageOp class method), 117
from_reader() (pycozmo.protocol_encoder.NvStorageOpResult class method), 117
from_reader() (pycozmo.protocol_encoder.ObjectAccel class method), 118
from_reader() (pycozmo.protocol_encoder.ObjectAvailable class method), 118
from_reader() (pycozmo.protocol_encoder.ObjectConnect class method), 119
from_reader() (pycozmo.protocol_encoder.ObjectConnectionState class method), 119
from_reader() (py-

cozmo.protocol_encoder.ObjectMoved class method), 120
from_reader() (*pycozmo.protocol_encoder.ObjectPowerLevel class method*), 120
from_reader() (*pycozmo.protocol_encoder.ObjectStoppedMoving class method*), 121
from_reader() (*pycozmo.protocol_encoder.ObjectTapFiltered class method*), 121
from_reader() (*pycozmo.protocol_encoder.ObjectTapped class method*), 121
from_reader() (*pycozmo.protocol_encoder.ObjectUpAxisChanged class method*), 123
from_reader() (*pycozmo.protocol_encoder.OutputAudio class method*), 124
from_reader() (*pycozmo.protocol_encoder.OutputSilence class method*), 124
from_reader() (*pycozmo.protocol_encoder.PathFollowingEvent class method*), 125
from_reader() (*pycozmo.protocol_encoder.PathSegmentSpeed class method*), 125
from_reader() (*pycozmo.protocol_encoder.Ping class method*), 125
from_reader() (*pycozmo.protocol_encoder.RecordHeading class method*), 126
from_reader() (*pycozmo.protocol_encoder.RobotDelocalized class method*), 126
from_reader() (*pycozmo.protocol_encoder.RobotPoked class method*), 126
from_reader() (*pycozmo.protocol_encoder.RobotState class method*), 127
from_reader() (*pycozmo.protocol_encoder.SetAccessoryDiscovery class method*), 128
from_reader() (*pycozmo.protocol_encoder.SetCameraParams class method*), 128
from_reader() (*pycozmo.protocol_encoder.SetHeadAngle class method*), 129
from_reader() (*pycozmo.protocol_encoder.SetHeadLight class method*), 129
from_reader() (*pycozmo.protocol_encoder.SetLiftHeight class method*), 130
from_reader() (*pycozmo.protocol_encoder.SetOrigin class method*), 130
from_reader() (*pycozmo.protocol_encoder.SetRobotVolume class method*), 131
from_reader() (*pycozmo.protocol_encoder.ShutdownRobot class method*), 131
from_reader() (*pycozmo.protocol_encoder.StartAnimation class method*), 131
from_reader() (*pycozmo.protocol_encoder.StartMotorCalibration class method*), 132
from_reader() (*pycozmo.protocol_encoder.StopAllMotors class method*), 132
from_reader() (*pycozmo.protocol_encoder.StreamObjectAccel class method*), 133
from_reader() (*pycozmo.protocol_encoder.SyncTime class method*), 133
from_reader() (*pycozmo.protocol_encoder.TrimPath class method*), 133
from_reader() (*pycozmo.protocol_encoder.TurnInPlace class method*), 134
from_reader() (*pycozmo.protocol_encoder.TurnInPlaceAtSpeed class method*), 134
from_reader() (*pycozmo.protocol_encoder.TurnToRecordedHeading class method*), 135
from_reader() (*pycozmo.protocol_encoder.WifiOff class method*), 136
from_x (*pycozmo.protocol_encoder.AppendPathSegLine attribute*), 99
from_y (*pycozmo.protocol_encoder.AppendPathSegLine attribute*), 99
fspec (*pycozmo.audiokinetic.soundbank.SoundBank attribute*), 42
Fury (*class in pycozmo.expressions.expressions*), 53

G

gain (*pycozmo.protocol_encoder.SetCameraParams attribute*), 128
generate() (*pycozmo.protocol_generator.ProtocolGenerator method*), 136

generate_argument_assignments() (*pycozmo.protocol_generator.ProtocolGenerator method*), 136
generate_argument_defaults() (*pycozmo.protocol_generator.ProtocolGenerator method*), 136
generate_argument_methods() (*pycozmo.protocol_generator.ProtocolGenerator method*), 136
generate_enum() (*pycozmo.protocol_generator.ProtocolGenerator method*), 136
generate_enum_validation() (*pycozmo.protocol_generator.ProtocolGenerator method*), 136
generate_farray_validation() (*pycozmo.protocol_generator.ProtocolGenerator method*), 136
generate_group_map() (*pycozmo.protocol_generator.ProtocolGenerator method*), 136
generate_id_map() (*pycozmo.protocol_generator.ProtocolGenerator method*), 136
generate_len_method() (*pycozmo.protocol_generator.ProtocolGenerator method*), 137
generate_packet() (*pycozmo.protocol_generator.ProtocolGenerator method*), 137
generate_packet_argument_assignments() (*pycozmo.protocol_generator.ProtocolGenerator method*), 137
generate_packet_decoding() (*pycozmo.protocol_generator.ProtocolGenerator method*), 137
generate_packet_encoding() (*pycozmo.protocol_generator.ProtocolGenerator method*), 137
generate_packet_slots() (*pycozmo.protocol_generator.ProtocolGenerator method*), 137
generate_repr_method() (*pycozmo.protocol_generator.ProtocolGenerator method*), 137
generate_string_validation() (*pycozmo.protocol_generator.ProtocolGenerator method*), 137
generate_struct() (*pycozmo.protocol_generator.ProtocolGenerator method*), 137
generate_varray_validation() (*pycozmo.protocol_generator.ProtocolGenerator method*), 137
get() (*pycozmo.anim_controller.AnimationQueue method*), 61
get() (*pycozmo.window.ReceiveWindow method*), 147
get() (*pycozmo.window.SendWindow method*), 147
get_anim_names() (*pycozmo.client.Client method*), 70
get_black() (*pycozmo.procedural_face.ProceduralLid class method*), 83
get_clip_metadata() (*in module pycozmo.anim_encoder*), 66
get_cozmo_anim_dir() (*in module pycozmo.util*), 146
get_cozmo_asset_dir() (*in module pycozmo.util*), 146
get_debug_message() (*in module pycozmo.robot_debug*), 141
get_farray_size() (*in module pycozmo.protocol_utils*), 138
get_id() (*pycozmo.behavior.Behavior method*), 67
get_log_level() (*in module pycozmo.robot_debug*), 141
get_object_farray_size() (*in module pycozmo.protocol_utils*), 138
get_object_size() (*in module pycozmo.protocol_utils*), 138
get_pycozmo_dir() (*in module pycozmo.util*), 146
get_size() (*in module pycozmo.protocol_utils*), 138
get_string_size() (*in module pycozmo.protocol_utils*), 138
get_varray_size() (*in module pycozmo.protocol_utils*), 138
getName() (*pycozmo.conn.Connection method*), 74
getName() (*pycozmo.conn.ReceiveThread method*), 71
getName() (*pycozmo.conn.SendThread method*), 72
go_to_pose() (*pycozmo.client.Client method*), 70
green (*in module pycozmo.lights*), 82
green_light (*in module pycozmo.lights*), 82
Guilt (*class in pycozmo.expressions.expressions*), 50
gyro_x (*pycozmo.protocol_encoder.RobotState attribute*), 127
gyro_y (*pycozmo.protocol_encoder.RobotState attribute*), 127
gyro_z (*pycozmo.protocol_encoder.RobotState attribute*), 127

H

half_eye_height (*pycozmo.expressions.expressions.Amazement attribute*), 57
half_eye_height (*pycozmo.expressions.expressions.Anger attribute*), 45
half_eye_height (*pycozmo.expressions.expressions.Annoyance*

attribute), 53
half_eye_height (*pycozmo.expressions.expressions.Asleep attribute*), 56
half_eye_height (*pycozmo.expressions.expressions.Boredom attribute*), 55
half_eye_height (*pycozmo.expressions.expressions.Confusion attribute*), 57
half_eye_height (*pycozmo.expressions.expressions.Despair attribute*), 49
half_eye_height (*pycozmo.expressions.expressions.Disappointment attribute*), 50
half_eye_height (*pycozmo.expressions.expressions.Disgust attribute*), 47
half_eye_height (*pycozmo.expressions.expressions.Embarrassment attribute*), 51
half_eye_height (*pycozmo.expressions.expressions.Excitement attribute*), 58
half_eye_height (*pycozmo.expressions.expressions.Fear attribute*), 48
half_eye_height (*pycozmo.expressions.expressions.Fury attribute*), 53
half_eye_height (*pycozmo.expressions.expressions.Guilt attribute*), 50
half_eye_height (*pycozmo.expressions.expressions.Happiness attribute*), 46
half_eye_height (*pycozmo.expressions.expressions.Horror attribute*), 52
half_eye_height (*pycozmo.expressions.expressions.Neutral attribute*), 44
half_eye_height (*pycozmo.expressions.expressions.Pleading attribute*), 48
half_eye_height (*pycozmo.expressions.expressions.Rejection attribute*), 54
half_eye_height (*pycozmo.expressions.expressions.Sadness attribute*), 46
half_eye_height (*pycozmo.expressions.expressions.Skepticism attribute*), 52
half_eye_height (*pycozmo.expressions.expressions.Surprise attribute*), 47
half_eye_height (*pycozmo.expressions.expressions.Suspicion attribute*), 54
half_eye_height (*pycozmo.expressions.expressions.Tiredness attribute*), 55
half_eye_height (*pycozmo.expressions.expressions.Vulnerability attribute*), 49
half_eye_height (*pycozmo.procedural_face.ProceduralEye attribute*), 84
half_eye_height (*pycozmo.procedural_face.ProceduralFace attribute*), 85
half_eye_height (*pycozmo.procedural_face.ProceduralLid attribute*), 83
half_eye_width (*pycozmo.expressions.expressions.Amazement attribute*), 57
half_eye_width (*pycozmo.expressions.expressions.Anger attribute*), 45
half_eye_width (*pycozmo.expressions.expressions.Annoyance attribute*), 53
half_eye_width (*pycozmo.expressions.expressions.Asleep attribute*), 56
half_eye_width (*pycozmo.expressions.expressions.Boredom attribute*), 55
half_eye_width (*pycozmo.expressions.expressions.Confusion attribute*), 57
half_eye_width (*pycozmo.expressions.expressions.Despair attribute*), 49
half_eye_width (*pycozmo.expressions.expressions.Disappointment attribute*), 51
half_eye_width (*pycozmo.expressions.expressions.Disgust attribute*), 47
half_eye_width (*pycozmo.expressions.expressions.Embarrassment attribute*), 51
half_eye_width (*pycozmo.expressions.expressions.Excitement*

attribute), 58
half_eye_width (*pycozmo.expressions.expressions.Fear attribute*), 48
half_eye_width (*pycozmo.expressions.expressions.Fury attribute*), 53
half_eye_width (*pycozmo.expressions.expressions.Guilt attribute*), 50
half_eye_width (*pycozmo.expressions.expressions.Happiness attribute*), 46
half_eye_width (*pycozmo.expressions.expressions.Horror attribute*), 52
half_eye_width (*pycozmo.expressions.expressions.Neutral attribute*), 45
half_eye_width (*pycozmo.expressions.expressions.Pleading attribute*), 48
half_eye_width (*pycozmo.expressions.expressions.Rejection attribute*), 54
half_eye_width (*pycozmo.expressions.expressions.Sadness attribute*), 46
half_eye_width (*pycozmo.expressions.expressions.Skepticism attribute*), 52
half_eye_width (*pycozmo.expressions.expressions.Surprise attribute*), 47
half_eye_width (*pycozmo.expressions.expressions.Suspicion attribute*), 54
half_eye_width (*pycozmo.expressions.expressions.Tiredness attribute*), 55
half_eye_width (*pycozmo.expressions.expressions.Vulnerability attribute*), 49
half_eye_width (*pycozmo.procedural_face.ProceduralEye attribute*), 84
half_eye_width (*pycozmo.procedural_face.ProceduralFace attribute*), 85
half_eye_width (*pycozmo.procedural_face.ProceduralLid attribute*), 83
handle_fin() (*pycozmo.conn.ReceiveThread method*), 71
handle_frame() (*pycozmo.conn.ReceiveThread method*), 71
handle_pkt() (*pycozmo.conn.ReceiveThread method*), 71
handle_reset() (*pycozmo.conn.ReceiveThread method*), 71
Handler (*class in pycozmo.event*), 78
Happiness (*class in pycozmo.expressions.expressions*), 46
HardwareInfo (*class in pycozmo.protocol_encoder*), 109
head (*pycozmo.protocol_encoder.StartMotorCalibration attribute*), 132
head (*pycozmo.protocol_encoder.TrimPath attribute*), 133
head_angle_max (*pycozmo.anim.AnimationGroupMember attribute*), 60
head_angle_min (*pycozmo.anim.AnimationGroupMember attribute*), 60
head_angle_rad (*pycozmo.protocol_encoder.RobotState attribute*), 127
HEAD_IN_POS (*pycozmo.robot.RobotStatusFlag attribute*), 140
heartbeat_thread_run() (*pycozmo.brain.Brain method*), 68
height (*pycozmo.expressions.expressions.Amazement attribute*), 57
height (*pycozmo.expressions.expressions.Anger attribute*), 45
height (*pycozmo.expressions.expressions.Annoyance attribute*), 53
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*), 84
height (*pycozmo.procedural_face.ProceduralFace attribute*), 85
height (*pycozmo.procedural_face.ProceduralLid attribute*), 83
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*), 146
hex_load() (*in module pycozmo.util*), 146
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*), 91
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*), 97
id (*pycozmo.protocol_encoder.AnimationState attribute*), 98
id (*pycozmo.protocol_encoder.AnimBackpackLights attribute*), 95
id (*pycozmo.protocol_encoder.AnimBody attribute*), 96
id (*pycozmo.protocol_encoder.AnimHead attribute*), 96
id (*pycozmo.protocol_encoder.AnimLift attribute*), 97
id (*pycozmo.protocol_encoder.AppendPathSegArc attribute*), 98
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*), 101
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*), 103
id (*pycozmo.protocol_encoder.DisplayImage attribute*), 104
id (*pycozmo.protocol_encoder.DriveWheels attribute*), 104
id (*pycozmo.protocol_encoder.Enable attribute*), 105
id (*pycozmo.protocol_encoder.EnableAnimationState attribute*), 105
id (*pycozmo.protocol_encoder.EnableCamera attribute*), 105
id (*pycozmo.protocol_encoder.EnableColorImages attribute*), 106
id (*pycozmo.protocol_encoder.EnableStopOnCliff attribute*), 106
id (*pycozmo.protocol_encoder.EndAnimation attribute*), 107

id (*pycozmo.protocol_encoder.ExecutePath attribute*), 107
id (*pycozmo.protocol_encoder.FallingStarted attribute*), 107
id (*pycozmo.protocol_encoder.FallingStopped attribute*), 108
id (*pycozmo.protocol_encoder.FirmwareSignature attribute*), 108
id (*pycozmo.protocol_encoder.FirmwareUpdate attribute*), 109
id (*pycozmo.protocol_encoder.FirmwareUpdateResult attribute*), 109
id (*pycozmo.protocol_encoder.HardwareInfo attribute*), 109
id (*pycozmo.protocol_encoder.ImageChunk attribute*), 110
id (*pycozmo.protocol_encoder.ImageImuData attribute*), 111
id (*pycozmo.protocol_encoder.Keyframe attribute*), 112
id (*pycozmo.protocol_encoder.LightStateCenter attribute*), 113
id (*pycozmo.protocol_encoder.LightStateSide attribute*), 113
id (*pycozmo.protocol_encoder.MotorCalibration attribute*), 114
id (*pycozmo.protocol_encoder.MoveHead attribute*), 114
id (*pycozmo.protocol_encoder.MoveLift attribute*), 114
id (*pycozmo.protocol_encoder.NvStorageOp attribute*), 117
id (*pycozmo.protocol_encoder.NvStorageOpResult attribute*), 117
id (*pycozmo.protocol_encoder.ObjectAccel attribute*), 118
id (*pycozmo.protocol_encoder.ObjectAvailable attribute*), 118
id (*pycozmo.protocol_encoder.ObjectConnect attribute*), 119
id (*pycozmo.protocol_encoder.ObjectConnectionState attribute*), 119
id (*pycozmo.protocol_encoder.ObjectMoved attribute*), 120
id (*pycozmo.protocol_encoder.ObjectPowerLevel attribute*), 120
id (*pycozmo.protocol_encoder.ObjectStoppedMoving attribute*), 121
id (*pycozmo.protocol_encoder.ObjectTapFiltered attribute*), 121
id (*pycozmo.protocol_encoder.ObjectTapped attribute*), 122
id (*pycozmo.protocol_encoder.ObjectUpAxisChanged attribute*), 123
id (*pycozmo.protocol_encoder.OutputAudio attribute*), 124
id (*pycozmo.protocol_encoder.OutputSilence attribute*), 124
id (*pycozmo.protocol_encoder.PathFollowingEvent attribute*), 125
id (*pycozmo.protocol_encoder.Ping attribute*), 125
id (*pycozmo.protocol_encoder.RecordHeading attribute*), 126
id (*pycozmo.protocol_encoder.RobotDelocalized attribute*), 126
id (*pycozmo.protocol_encoder.RobotPoked attribute*), 126
id (*pycozmo.protocol_encoder.RobotState attribute*), 127
id (*pycozmo.protocol_encoder.SetAccessoryDiscovery attribute*), 128
id (*pycozmo.protocol_encoder.SetCameraParams attribute*), 128
id (*pycozmo.protocol_encoder.SetHeadAngle attribute*), 129
id (*pycozmo.protocol_encoder.SetHeadLight attribute*), 129
id (*pycozmo.protocol_encoder.SetLiftHeight attribute*), 130
id (*pycozmo.protocol_encoder.SetOrigin attribute*), 130
id (*pycozmo.protocol_encoder.SetRobotVolume attribute*), 131
id (*pycozmo.protocol_encoder.ShutdownRobot attribute*), 131
id (*pycozmo.protocol_encoder.StartAnimation attribute*), 131
id (*pycozmo.protocol_encoder.StartMotorCalibration attribute*), 132
id (*pycozmo.protocol_encoder.StopAllMotors attribute*), 132
id (*pycozmo.protocol_encoder.StreamObjectAccel attribute*), 133
id (*pycozmo.protocol_encoder.SyncTime attribute*), 133
id (*pycozmo.protocol_encoder.TrimPath attribute*), 133
id (*pycozmo.protocol_encoder.TurnInPlace attribute*), 134
id (*pycozmo.protocol_encoder.TurnInPlaceAtSpeed attribute*), 135
id (*pycozmo.protocol_encoder.TurnToRecordedHeading attribute*), 135
id (*pycozmo.protocol_encoder.WifiOff attribute*), 136
ident (*pycozmo.conn.Connection attribute*), 74
ident (*pycozmo.conn.ReceiveThread attribute*), 71
ident (*pycozmo.conn.SendThread attribute*), 72
IDLE (*pycozmo.conn.Connection attribute*), 74
image (*pycozmo.protocol_encoder.DisplayImage attribute*), 104
image_chunk_count (*pycozmo.protocol_encoder.ImageChunk attribute*), 110
image_encoding (*pycozmo.protocol_encoder.ImageChunk at-*

tribute), 110
image_id (*pycozmo.protocol_encoder.ImageChunk attribute*), 110
image_id (*pycozmo.protocol_encoder.ImageImuData attribute*), 111
image_resolution (*pycozmo.protocol_encoder.EnableCamera attribute*), 105
image_resolution (*pycozmo.protocol_encoder.ImageChunk attribute*), 110
image_send_mode (*pycozmo.protocol_encoder.EnableCamera attribute*), 105
image_to_str() (*in module pycozmo.image_encoder*), 81
ImageChunk (*class in pycozmo.protocol_encoder*), 110
ImageDecoder (*class in pycozmo.image_encoder*), 81
ImageEncoder (*class in pycozmo.image_encoder*), 81
ImageEncoding (*class in pycozmo.protocol_encoder*), 110
ImageImuData (*class in pycozmo.protocol_encoder*), 111
ImageResolution (*class in pycozmo.protocol_encoder*), 111
ImageResolutionCount (*pycozmo.protocol_encoder.ImageResolution attribute*), 111
ImageResolutionNone (*pycozmo.protocol_encoder.ImageResolution attribute*), 111
ImageSendMode (*class in pycozmo.protocol_encoder*), 112
impact_intensity (*pycozmo.protocol_encoder.FallingStopped attribute*), 108
in_column_order (*pycozmo.util.Matrix44 attribute*), 144
in_row_order (*pycozmo.util.Matrix44 attribute*), 144
inches (*pycozmo.util.Distance attribute*), 143
Int16Argument (*class in pycozmo.protocol_ast*), 88
Int32Argument (*class in pycozmo.protocol_ast*), 88
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*), 85
invalidate() (*pycozmo.util.Pose method*), 145
InvalidObject (*pycozmo.protocol_encoder.ObjectType attribute*), 123
is_absolute (*pycozmo.protocol_encoder.TurnInPlace attribute*), 134
is_accurate (*pycozmo.util.Pose attribute*), 145
is_alive() (*pycozmo.conn.Connection method*), 74
is_alive() (*pycozmo.conn.ReceiveThread method*), 71
is_alive() (*pycozmo.conn.SendThread method*), 73
IS_ANIM_BUFFER_FULL (*pycozmo.robot.RobotStatusFlag attribute*), 140
IS_ANIMATING (*pycozmo.robot.RobotStatusFlag attribute*), 140
IS_ANIMATING_IDLE (*pycozmo.robot.RobotStatusFlag attribute*), 140
IS_BODY_ACC_MODE (*pycozmo.robot.RobotStatusFlag attribute*), 140
IS_CARRYING_BLOCK (*pycozmo.robot.RobotStatusFlag attribute*), 140
IS_CHARGER_OOS (*pycozmo.robot.RobotStatusFlag attribute*), 140
IS_CHARGING (*pycozmo.robot.RobotStatusFlag attribute*), 140
is_comparable() (*pycozmo.util.Pose method*), 145
is_empty() (*pycozmo.anim_controller.AnimationQueue method*), 61
IS_FALLING (*pycozmo.robot.RobotStatusFlag attribute*), 140
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*), 91
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*), 97
is_from_engine() (*pycozmo.protocol_encoder.AnimationState method*), 98

is_from_engine() (pycozmo.protocol_encoder.AnimBackpackLights method), 95
is_from_engine() (pycozmo.protocol_encoder.AnimBody method), 96
is_from_engine() (pycozmo.protocol_encoder.AnimHead method), 96
is_from_engine() (pycozmo.protocol_encoder.AnimLift method), 97
is_from_engine() (pycozmo.protocol_encoder.AppendPathSegArc method), 98
is_from_engine() (pycozmo.protocol_encoder.AppendPathSegLine method), 99
is_from_engine() (pycozmo.protocol_encoder.AppendPathSegPointTurn method), 100
is_from_engine() (pycozmo.protocol_encoder.BodyInfo method), 101
is_from_engine() (pycozmo.protocol_encoder.ButtonPressed method), 101
is_from_engine() (pycozmo.protocol_encoder.ClearPath method), 101
is_from_engine() (pycozmo.protocol_encoder.Connect method), 102
is_from_engine() (pycozmo.protocol_encoder.CubeId method), 102
is_from_engine() (pycozmo.protocol_encoder.CubeLights method), 103
is_from_engine() (pycozmo.protocol_encoder.DebugData method), 103
is_from_engine() (pycozmo.protocol_encoder.Disconnect method), 103
is_from_engine() (pycozmo.protocol_encoder.DisplayImage method), 104
is_from_engine() (pycozmo.protocol_encoder.DriveWheels method), 104
is_from_engine() (pycozmo.protocol_encoder.Enable method), 105
is_from_engine() (pycozmo.protocol_encoder.EnableAnimationState method), 105
is_from_engine() (pycozmo.protocol_encoder.EnableCamera method), 105
is_from_engine() (pycozmo.protocol_encoder.EnableColorImages method), 106
is_from_engine() (pycozmo.protocol_encoder.EnableStopOnCliff method), 106
is_from_engine() (pycozmo.protocol_encoder.EndAnimation method), 107
is_from_engine() (pycozmo.protocol_encoder.ExecutePath method), 107
is_from_engine() (pycozmo.protocol_encoder.FallingStarted method), 107
is_from_engine() (pycozmo.protocol_encoder.FallingStopped method), 108
is_from_engine() (pycozmo.protocol_encoder.FirmwareSignature method), 108
is_from_engine() (pycozmo.protocol_encoder.FirmwareUpdate method), 109
is_from_engine() (pycozmo.protocol_encoder.FirmwareUpdateResult method), 109
is_from_engine() (pycozmo.protocol_encoder.HardwareInfo method), 109
is_from_engine() (pycozmo.protocol_encoder.ImageChunk method), 110
is_from_engine() (pycozmo.protocol_encoder.ImageImuData method), 111
is_from_engine() (pycozmo.protocol_encoder.Keyframe method), 112
is_from_engine() (pycozmo.protocol_encoder.LightStateCenter method), 113
is_from_engine() (pycozmo.protocol_encoder.LightStateSide method), 113
is_from_engine() (pycozmo.protocol_encoder.MotorCalibration method), 114

is_from_engine() (*pycozmo.protocol_encoder.MoveHead method*), 114
is_from_engine() (*pycozmo.protocol_encoder.MoveLift method*), 115
is_from_engine() (*pycozmo.protocol_encoder.NvStorageOp method*), 117
is_from_engine() (*pycozmo.protocol_encoder.NvStorageOpResult method*), 117
is_from_engine() (*pycozmo.protocol_encoder.ObjectAccel method*), 118
is_from_engine() (*pycozmo.protocol_encoder.ObjectAvailable method*), 118
is_from_engine() (*pycozmo.protocol_encoder.ObjectConnect method*), 119
is_from_engine() (*pycozmo.protocol_encoder.ObjectConnectionState method*), 119
is_from_engine() (*pycozmo.protocol_encoder.ObjectMoved method*), 120
is_from_engine() (*pycozmo.protocol_encoder.ObjectPowerLevel method*), 120
is_from_engine() (*pycozmo.protocol_encoder.ObjectStoppedMoving method*), 121
is_from_engine() (*pycozmo.protocol_encoder.ObjectTapFiltered method*), 121
is_from_engine() (*pycozmo.protocol_encoder.ObjectTapped method*), 122
is_from_engine() (*pycozmo.protocol_encoder.ObjectUpAxisChanged method*), 123
is_from_engine() (*pycozmo.protocol_encoder.OutputAudio method*), 124
is_from_engine() (*pycozmo.protocol_encoder.OutputSilence method*), 124
is_from_engine() (*pycozmo.protocol_encoder.PathFollowingEvent method*), 125
is_from_engine() (*pycozmo.protocol_encoder.Ping method*), 125
is_from_engine() (*pycozmo.protocol_encoder.RecordHeading method*), 126
is_from_engine() (*pycozmo.protocol_encoder.RobotDelocalized method*), 126
is_from_engine() (*pycozmo.protocol_encoder.RobotPoked method*), 126
is_from_engine() (*pycozmo.protocol_encoder.RobotState method*), 127
is_from_engine() (*pycozmo.protocol_encoder.SetAccessoryDiscovery method*), 128
is_from_engine() (*pycozmo.protocol_encoder.SetCameraParams method*), 128
is_from_engine() (*pycozmo.protocol_encoder.SetHeadAngle method*), 129
is_from_engine() (*pycozmo.protocol_encoder.SetHeadLight method*), 129
is_from_engine() (*pycozmo.protocol_encoder.SetLiftHeight method*), 130
is_from_engine() (*pycozmo.protocol_encoder.SetOrigin method*), 130
is_from_engine() (*pycozmo.protocol_encoder.SetRobotVolume method*), 131
is_from_engine() (*pycozmo.protocol_encoder.ShutdownRobot method*), 131
is_from_engine() (*pycozmo.protocol_encoder.StartAnimation method*), 132
is_from_engine() (*pycozmo.protocol_encoder.StartMotorCalibration method*), 132
is_from_engine() (*pycozmo.protocol_encoder.StopAllMotors method*), 132
is_from_engine() (*pycozmo.protocol_encoder.StreamObjectAccel method*), 133
is_from_engine() (*pycozmo.protocol_encoder.SyncTime method*), 133
is_from_engine() (*pycozmo.protocol_encoder.TrimPath method*), 133
is_from_engine() (*py-*

cozmo.protocol_encoder.TurnInPlace method), 134
is_from_engine() (*pycozmo.protocol_encoder.TurnInPlaceAtSpeed method*), 135
is_from_engine() (*pycozmo.protocol_encoder.TurnToRecordedHeading method*), 135
is_from_engine() (*pycozmo.protocol_encoder.WifiOff method*), 136
is_from_robot() (*pycozmo.protocol_base.Packet method*), 91
is_from_robot() (*pycozmo.protocol_base.UnknownCommand method*), 92
is_from_robot() (*pycozmo.protocol_base.UnknownEvent method*), 92
is_from_robot() (*pycozmo.protocol_base.UnknownPacket method*), 91
is_from_robot() (*pycozmo.protocol_encoder.AbortAnimation method*), 95
is_from_robot() (*pycozmo.protocol_encoder.AcknowledgeAction method*), 95
is_from_robot() (*pycozmo.protocol_encoder.AnimationEnded method*), 97
is_from_robot() (*pycozmo.protocol_encoder.AnimationStarted method*), 97
is_from_robot() (*pycozmo.protocol_encoder.AnimationState method*), 98
is_from_robot() (*pycozmo.protocol_encoder.AnimBackpackLights method*), 95
is_from_robot() (*pycozmo.protocol_encoder.AnimBody method*), 96
is_from_robot() (*pycozmo.protocol_encoder.AnimHead method*), 96
is_from_robot() (*pycozmo.protocol_encoder.AnimLift method*), 97
is_from_robot() (*pycozmo.protocol_encoder.AppendPathSegArc method*), 99
is_from_robot() (*pycozmo.protocol_encoder.AppendPathSegLine method*), 99
is_from_robot() (*pycozmo.protocol_encoder.AppendPathSegPointTurn method*), 100
is_from_robot() (*pycozmo.protocol_encoder.BodyInfo method*), 101
is_from_robot() (*pycozmo.protocol_encoder.ButtonPressed method*), 101
is_from_robot() (*pycozmo.protocol_encoder.ClearPath method*), 101
is_from_robot() (*pycozmo.protocol_encoder.Connect method*), 102
is_from_robot() (*pycozmo.protocol_encoder.CubeId method*), 102
is_from_robot() (*pycozmo.protocol_encoder.CubeLights method*), 103
is_from_robot() (*pycozmo.protocol_encoder.DebugData method*), 103
is_from_robot() (*pycozmo.protocol_encoder.Disconnect method*), 103
is_from_robot() (*pycozmo.protocol_encoder.DisplayImage method*), 104
is_from_robot() (*pycozmo.protocol_encoder.DriveWheels method*), 104
is_from_robot() (*pycozmo.protocol_encoder.Enable method*), 105
is_from_robot() (*pycozmo.protocol_encoder.EnableAnimationState method*), 105
is_from_robot() (*pycozmo.protocol_encoder.EnableCamera method*), 105
is_from_robot() (*pycozmo.protocol_encoder.EnableColorImages method*), 106
is_from_robot() (*pycozmo.protocol_encoder.EnableStopOnCliff method*), 106
is_from_robot() (*pycozmo.protocol_encoder.EndAnimation method*), 107
is_from_robot() (*pycozmo.protocol_encoder.ExecutePath method*),

107
is_from_robot() (pycozmo.protocol_encoder.FallingStarted method), 107
is_from_robot() (pycozmo.protocol_encoder.FallingStopped method), 108
is_from_robot() (pycozmo.protocol_encoder.FirmwareSignature method), 108
is_from_robot() (pycozmo.protocol_encoder.FirmwareUpdate method), 109
is_from_robot() (pycozmo.protocol_encoder.FirmwareUpdateResult method), 109
is_from_robot() (pycozmo.protocol_encoder.HardwareInfo method), 109
is_from_robot() (pycozmo.protocol_encoder.ImageChunk method), 110
is_from_robot() (pycozmo.protocol_encoder.ImageImuData method), 111
is_from_robot() (pycozmo.protocol_encoder.Keyframe method), 112
is_from_robot() (pycozmo.protocol_encoder.LightStateCenter method), 113
is_from_robot() (pycozmo.protocol_encoder.LightStateSide method), 113
is_from_robot() (pycozmo.protocol_encoder.MotorCalibration method), 114
is_from_robot() (pycozmo.protocol_encoder.MoveHead method), 114
is_from_robot() (pycozmo.protocol_encoder.MoveLift method), 115
is_from_robot() (pycozmo.protocol_encoder.NvStorageOp method), 117
is_from_robot() (pycozmo.protocol_encoder.NvStorageOpResult method), 117
is_from_robot() (pycozmo.protocol_encoder.ObjectAccel method), 118
is_from_robot() (pycozmo.protocol_encoder.ObjectAvailable method), 118
is_from_robot() (pycozmo.protocol_encoder.ObjectConnect method), 119
is_from_robot() (pycozmo.protocol_encoder.ObjectConnectionState method), 119
is_from_robot() (pycozmo.protocol_encoder.ObjectMoved method), 120
is_from_robot() (pycozmo.protocol_encoder.ObjectPowerLevel method), 120
is_from_robot() (pycozmo.protocol_encoder.ObjectStoppedMoving method), 121
is_from_robot() (pycozmo.protocol_encoder.ObjectTapFiltered method), 121
is_from_robot() (pycozmo.protocol_encoder.ObjectTapped method), 122
is_from_robot() (pycozmo.protocol_encoder.ObjectUpAxisChanged method), 123
is_from_robot() (pycozmo.protocol_encoder.OutputAudio method), 124
is_from_robot() (pycozmo.protocol_encoder.OutputSilence method), 124
is_from_robot() (pycozmo.protocol_encoder.PathFollowingEvent method), 125
is_from_robot() (pycozmo.protocol_encoder.Ping method), 125
is_from_robot() (pycozmo.protocol_encoder.RecordHeading method), 126
is_from_robot() (pycozmo.protocol_encoder.RobotDelocalized method), 126
is_from_robot() (pycozmo.protocol_encoder.RobotPoked method), 126
is_from_robot() (pycozmo.protocol_encoder.RobotState method), 127
is_from_robot() (pycozmo.protocol_encoder.SetAccessoryDiscovery method), 128
is_from_robot() (pycozmo.protocol_encoder.SetCameraParams method), 128

is_from_robot() (*pycozmo.protocol_encoder.SetHeadAngle method*), 129
is_from_robot() (*pycozmo.protocol_encoder.SetHeadLight method*), 129
is_from_robot() (*pycozmo.protocol_encoder.SetLiftHeight method*), 130
is_from_robot() (*pycozmo.protocol_encoder.SetOrigin method*), 130
is_from_robot() (*pycozmo.protocol_encoder.SetRobotVolume method*), 131
is_from_robot() (*pycozmo.protocol_encoder.ShutdownRobot method*), 131
is_from_robot() (*pycozmo.protocol_encoder.StartAnimation method*), 132
is_from_robot() (*pycozmo.protocol_encoder.StartMotorCalibration method*), 132
is_from_robot() (*pycozmo.protocol_encoder.StopAllMotors method*), 132
is_from_robot() (*pycozmo.protocol_encoder.StreamObjectAccel method*), 133
is_from_robot() (*pycozmo.protocol_encoder.SyncTime method*), 133
is_from_robot() (*pycozmo.protocol_encoder.TrimPath method*), 134
is_from_robot() (*pycozmo.protocol_encoder.TurnInPlace method*), 134
is_from_robot() (*pycozmo.protocol_encoder.TurnInPlaceAtSpeed method*), 135
is_from_robot() (*pycozmo.protocol_encoder.TurnToRecordedHeading method*), 135
is_from_robot() (*pycozmo.protocol_encoder.WifiOff method*), 136
is_full() (*pycozmo.window.SendWindow method*), 147
IS_MOVING (*pycozmo.robot.RobotStatusFlag attribute*), 140
IS_ON_CHARGER (*pycozmo.robot.RobotStatusFlag attribute*), 140
is_oob() (*pycozmo.protocol_base.Packet method*), 91
is_oob() (*pycozmo.protocol_base.UnknownCommand method*), 92
is_oob() (*pycozmo.protocol_base.UnknownEvent method*), 92
is_oob() (*pycozmo.protocol_base.UnknownPacket method*), 91
is_oob() (*pycozmo.protocol_encoder.AbortAnimation method*), 95
is_oob() (*pycozmo.protocol_encoder.AcknowledgeAction method*), 95
is_oob() (*pycozmo.protocol_encoder.AnimationEnded method*), 97
is_oob() (*pycozmo.protocol_encoder.AnimationStarted method*), 97
is_oob() (*pycozmo.protocol_encoder.AnimationState method*), 98
is_oob() (*pycozmo.protocol_encoder.AnimBackpackLights method*), 95
is_oob() (*pycozmo.protocol_encoder.AnimBody method*), 96
is_oob() (*pycozmo.protocol_encoder.AnimHead method*), 96
is_oob() (*pycozmo.protocol_encoder.AnimLift method*), 97
is_oob() (*pycozmo.protocol_encoder.AppendPathSegArc method*), 99
is_oob() (*pycozmo.protocol_encoder.AppendPathSegLine method*), 99
is_oob() (*pycozmo.protocol_encoder.AppendPathSegPointTurn method*), 100
is_oob() (*pycozmo.protocol_encoder.BodyInfo method*), 101
is_oob() (*pycozmo.protocol_encoder.ButtonPressed method*), 101
is_oob() (*pycozmo.protocol_encoder.ClearPath method*), 101
is_oob() (*pycozmo.protocol_encoder.Connect method*), 102
is_oob() (*pycozmo.protocol_encoder.CubeId method*), 102
is_oob() (*pycozmo.protocol_encoder.CubeLights method*), 103
is_oob() (*pycozmo.protocol_encoder.DebugData method*), 103
is_oob() (*pycozmo.protocol_encoder.Disconnect method*), 103
is_oob() (*pycozmo.protocol_encoder.DisplayImage method*), 104
is_oob() (*pycozmo.protocol_encoder.DriveWheels method*), 104
is_oob() (*pycozmo.protocol_encoder.Enable method*), 105
is_oob() (*pycozmo.protocol_encoder.EnableAnimationState*

method), 105
is_oob() (*pycozmo.protocol_encoder.EnableCamera method*), 105
is_oob() (*pycozmo.protocol_encoder.EnableColorImages method*), 106
is_oob() (*pycozmo.protocol_encoder.EnableStopOnCliff method*), 106
is_oob() (*pycozmo.protocol_encoder.EndAnimation method*), 107
is_oob() (*pycozmo.protocol_encoder.ExecutePath method*), 107
is_oob() (*pycozmo.protocol_encoder.FallingStarted method*), 107
is_oob() (*pycozmo.protocol_encoder.FallingStopped method*), 108
is_oob() (*pycozmo.protocol_encoder.FirmwareSignature method*), 108
is_oob() (*pycozmo.protocol_encoder.FirmwareUpdate method*), 109
is_oob() (*pycozmo.protocol_encoder.FirmwareUpdateResult method*), 109
is_oob() (*pycozmo.protocol_encoder.HardwareInfo method*), 109
is_oob() (*pycozmo.protocol_encoder.ImageChunk method*), 110
is_oob() (*pycozmo.protocol_encoder.ImageImuData method*), 111
is_oob() (*pycozmo.protocol_encoder.Keyframe method*), 112
is_oob() (*pycozmo.protocol_encoder.LightStateCenter method*), 113
is_oob() (*pycozmo.protocol_encoder.LightStateSide method*), 113
is_oob() (*pycozmo.protocol_encoder.MotorCalibration method*), 114
is_oob() (*pycozmo.protocol_encoder.MoveHead method*), 114
is_oob() (*pycozmo.protocol_encoder.MoveLift method*), 115
is_oob() (*pycozmo.protocol_encoder.NvStorageOp method*), 117
is_oob() (*pycozmo.protocol_encoder.NvStorageOpResult method*), 117
is_oob() (*pycozmo.protocol_encoder.ObjectAccel method*), 118
is_oob() (*pycozmo.protocol_encoder.ObjectAvailable method*), 118
is_oob() (*pycozmo.protocol_encoder.ObjectConnect method*), 119
is_oob() (*pycozmo.protocol_encoder.ObjectConnectionState method*), 119
is_oob() (*pycozmo.protocol_encoder.ObjectMoved method*), 120
is_oob() (*pycozmo.protocol_encoder.ObjectPowerLevel method*), 120
is_oob() (*pycozmo.protocol_encoder.ObjectStoppedMoving method*), 121
is_oob() (*pycozmo.protocol_encoder.ObjectTapFiltered method*), 121
is_oob() (*pycozmo.protocol_encoder.ObjectTapped method*), 122
is_oob() (*pycozmo.protocol_encoder.ObjectUpAxisChanged method*), 123
is_oob() (*pycozmo.protocol_encoder.OutputAudio method*), 124
is_oob() (*pycozmo.protocol_encoder.OutputSilence method*), 124
is_oob() (*pycozmo.protocol_encoder.PathFollowingEvent method*), 125
is_oob() (*pycozmo.protocol_encoder.Ping method*), 125
is_oob() (*pycozmo.protocol_encoder.RecordHeading method*), 126
is_oob() (*pycozmo.protocol_encoder.RobotDelocalized method*), 126
is_oob() (*pycozmo.protocol_encoder.RobotPoked method*), 126
is_oob() (*pycozmo.protocol_encoder.RobotState method*), 127
is_oob() (*pycozmo.protocol_encoder.SetAccessoryDiscovery method*), 128
is_oob() (*pycozmo.protocol_encoder.SetCameraParams method*), 128
is_oob() (*pycozmo.protocol_encoder.SetHeadAngle method*), 129
is_oob() (*pycozmo.protocol_encoder.SetHeadLight method*), 129
is_oob() (*pycozmo.protocol_encoder.SetLiftHeight method*), 130
is_oob() (*pycozmo.protocol_encoder.SetOrigin method*), 130
is_oob() (*pycozmo.protocol_encoder.SetRobotVolume method*), 131
is_oob() (*pycozmo.protocol_encoder.ShutdownRobot method*), 131
is_oob() (*pycozmo.protocol_encoder.StartAnimation method*), 132
is_oob() (*pycozmo.protocol_encoder.StartMotorCalibration method*), 132
is_oob() (*pycozmo.protocol_encoder.StopAllMotors method*), 132
is_oob() (*pycozmo.protocol_encoder.StreamObjectAccel method*), 133
is_oob() (*pycozmo.protocol_encoder.SyncTime method*), 133
is_oob() (*pycozmo.protocol_encoder.TrimPath 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*), 147
is_out_of_order() (*pycozmo.window.SendWindow method*), 147
IS_PATHING (*pycozmo.robot.RobotStatusFlag attribute*), 140
IS_PICKED_UP (*pycozmo.robot.RobotStatusFlag attribute*), 140
IS_PICKING_OR_PLACING (*pycozmo.robot.RobotStatusFlag attribute*), 140
is_valid (*pycozmo.util.Pose attribute*), 145
is_valid_seq() (*pycozmo.window.BaseWindow method*), 146
is_valid_seq() (*pycozmo.window.ReceiveWindow method*), 147
is_valid_seq() (*pycozmo.window.SendWindow method*), 147
isAlive() (*pycozmo.conn.Connection method*), 74
isAlive() (*pycozmo.conn.ReceiveThread method*), 71
isAlive() (*pycozmo.conn.SendThread method*), 72
isDaemon() (*pycozmo.conn.Connection method*), 74
isDaemon() (*pycozmo.conn.ReceiveThread method*), 71
isDaemon() (*pycozmo.conn.SendThread method*), 73

J

join() (*pycozmo.conn.Connection method*), 74
join() (*pycozmo.conn.ReceiveThread method*), 71
join() (*pycozmo.conn.SendThread method*), 73
JPEGColor (*pycozmo.protocol_encoder.ImageEncoding attribute*), 110
JPEGColorHalfWidth (*pycozmo.protocol_encoder.ImageEncoding attribute*), 110
JPEGGray (*pycozmo.protocol_encoder.ImageEncoding attribute*), 110
JPEGMinimizedColor (*pycozmo.protocol_encoder.ImageEncoding attribute*), 110
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*), 86
keyframe_to_im() (*pycozmo.anim.PreprocessedClip class method*), 59

L

language (*pycozmo.audiokinetic.soundbanksinfo.SoundBankInfo attribute*), 43
last (*pycozmo.protocol_encoder.Ping attribute*), 125
LE_BL_v16 (*pycozmo.protocol_encoder.BodyColor attribute*), 100
left_xyz (*pycozmo.util.Matrix44 attribute*), 144
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*), 117
level (*pycozmo.protocol_encoder.DebugData attribute*), 103
level (*pycozmo.protocol_encoder.SetRobotVolume attribute*), 131
lids (*pycozmo.procedural_face.ProceduralEye attribute*), 84
lift (*pycozmo.protocol_encoder.StartMotorCalibration attribute*), 132
LIFT_ARM_LENGTH (*in module pycozmo.robot*), 140
lift_height_mm (*pycozmo.protocol_encoder.RobotState attribute*), 127
LIFT_IN_POS (*pycozmo.robot.RobotStatusFlag attribute*), 140
LIFT_PIVOT_HEIGHT (*in module pycozmo.robot*), 140
LiftPosition (*class in pycozmo.robot*), 140
LightState (*class in pycozmo.protocol_encoder*), 112
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*), 76
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*), 84
lower_inner_radius_y (*pycozmo.procedural_face.ProceduralEye attribute*), 84
lower_outer_radius_x (*pycozmo.procedural_face.ProceduralEye attribute*), 84
lower_outer_radius_y (*pycozmo.procedural_face.ProceduralEye attribute*), 84
lwheel_accel_mmps2 (*pycozmo.protocol_encoder.DriveWheels attribute*), 104
lwheel_speed_mmps (*pycozmo.protocol_encoder.DriveWheels attribute*), 104
lwheel_speed_mmps (*pycozmo.protocol_encoder.RobotState attribute*), 127

M

m00 (*pycozmo.util.Matrix44 attribute*), 144
m01 (*pycozmo.util.Matrix44 attribute*), 144
m02 (*pycozmo.util.Matrix44 attribute*), 144
m03 (*pycozmo.util.Matrix44 attribute*), 144
m10 (*pycozmo.util.Matrix44 attribute*), 144
m11 (*pycozmo.util.Matrix44 attribute*), 144
m12 (*pycozmo.util.Matrix44 attribute*), 144
m13 (*pycozmo.util.Matrix44 attribute*), 144
m20 (*pycozmo.util.Matrix44 attribute*), 144
m21 (*pycozmo.util.Matrix44 attribute*), 144
m22 (*pycozmo.util.Matrix44 attribute*), 144
m23 (*pycozmo.util.Matrix44 attribute*), 144
m30 (*pycozmo.util.Matrix44 attribute*), 144
m31 (*pycozmo.util.Matrix44 attribute*), 144
m32 (*pycozmo.util.Matrix44 attribute*), 144
m33 (*pycozmo.util.Matrix44 attribute*), 144
Matrix44 (*class in pycozmo.util*), 144
MAX_HEAD_ANGLE (*in module pycozmo.robot*), 139
MAX_LIFT_ANGLE (*in module pycozmo.robot*), 140
MAX_LIFT_HEIGHT (*in module pycozmo.robot*), 140
max_speed_rad_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_HEAD_ANGLE (*in module pycozmo.robot*), 139
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*), 120
mm (*pycozmo.util.Distance attribute*), 143
mmps (*pycozmo.util.Speed attribute*), 143
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
MotorCalibration (*class in pycozmo.protocol_encoder*), 113
MotorID (*class in pycozmo.protocol_encoder*), 114
move_head() (*pycozmo.client.Client method*), 70
move_lift() (*pycozmo.client.Client method*), 70
MoveHead (*class in pycozmo.protocol_encoder*), 114

MoveLift (*class in pycozmo.protocol_encoder*), 114

N

name (*pycozmo.anim.AnimationGroupMember attribute*), 60
name (*pycozmo.audiokinetic.soundbank.Event attribute*), 41
name (*pycozmo.audiokinetic.soundbank.SFX attribute*), 42
name (*pycozmo.audiokinetic.soundbank.SoundBank attribute*), 42
name (*pycozmo.audiokinetic.soundbanksinfo.EventInfo attribute*), 43
name (*pycozmo.audiokinetic.soundbanksinfo.FileInfo attribute*), 43
name (*pycozmo.audiokinetic.soundbanksinfo.SoundBankInfo attribute*), 43
name (*pycozmo.behavior.ReactionTrigger attribute*), 67
name (*pycozmo.conn.Connection attribute*), 75
name (*pycozmo.conn.ReceiveThread attribute*), 72
name (*pycozmo.conn.SendThread attribute*), 73
name (*pycozmo.emotions.EmotionEvent attribute*), 76
name (*pycozmo.emotions.EmotionType attribute*), 76
name_id (*pycozmo.protocol_encoder.DebugData attribute*), 103
Neutral (*class in pycozmo.expressions.expressions*), 44
NoneImageEncoding (*pycozmo.protocol_encoder.ImageEncoding attribute*), 111
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*), 135
NV_BAD_ARGS (*pycozmo.protocol_encoder.NvResult attribute*), 116
NV_BUSY (*pycozmo.protocol_encoder.NvResult attribute*), 116
NV_CORRUPT (*pycozmo.protocol_encoder.NvResult attribute*), 116
NV_ERROR (*pycozmo.protocol_encoder.NvResult attribute*), 116
NV_LOOP (*pycozmo.protocol_encoder.NvResult attribute*), 116
NV_MORE (*pycozmo.protocol_encoder.NvResult attribute*), 116
NV_NO_DO (*pycozmo.protocol_encoder.NvResult attribute*), 116
NV_NO_MEM (*pycozmo.protocol_encoder.NvResult attribute*), 116
NV_NO_ROOM (*pycozmo.protocol_encoder.NvResult attribute*), 116
NV_NOT_FOUND (*pycozmo.protocol_encoder.NvResult attribute*), 116
NV_OKAY (*pycozmo.protocol_encoder.NvResult attribute*), 116
NV_SCHEDULED (*pycozmo.protocol_encoder.NvResult attribute*), 116
NV_TIMEOUT (*pycozmo.protocol_encoder.NvResult attribute*), 116
NV_UNKNOWN_4 (*pycozmo.protocol_encoder.NvResult attribute*), 116
NV_UNKNOWN_5 (*pycozmo.protocol_encoder.NvResult attribute*), 117
NV_UNKNOWN_6 (*pycozmo.protocol_encoder.NvResult attribute*), 117
NV_UNKNOWN_7 (*pycozmo.protocol_encoder.NvResult attribute*), 117
NV_UNKNOWN_8 (*pycozmo.protocol_encoder.NvResult attribute*), 117
NVEntry_BirthCertificate (*pycozmo.protocol_encoder.NvEntryTag attribute*), 115
NVEntry_CalibImage1 (*pycozmo.protocol_encoder.NvEntryTag attribute*), 115
NVEntry_CalibImage2 (*pycozmo.protocol_encoder.NvEntryTag attribute*), 115
NVEntry_CalibImage3 (*pycozmo.protocol_encoder.NvEntryTag attribute*), 115
NVEntry_CalibImage4 (*pycozmo.protocol_encoder.NvEntryTag attribute*), 115
NVEntry_CalibImage5 (*pycozmo.protocol_encoder.NvEntryTag attribute*), 115
NVEntry_CalibImage6 (*pycozmo.protocol_encoder.NvEntryTag attribute*), 115
NVEntry_CalibMetaInfo (*pycozmo.protocol_encoder.NvEntryTag attribute*), 115
NVEntry_CalibPose (*pycozmo.protocol_encoder.NvEntryTag attribute*), 115
NVEntry_CameraCalib (*pycozmo.protocol_encoder.NvEntryTag attribute*), 115
NVEntry_CliffValOnDrop (*pycozmo.protocol_encoder.NvEntryTag at-*

tribute), 115
NVEntry_CliffValOnGround (*pycozmo.protocol_encoder.NvEntryTag attribute*), 115
NVEntry_FaceAlbumData (*pycozmo.protocol_encoder.NvEntryTag attribute*), 115
NVEntry_FaceEnrollData (*pycozmo.protocol_encoder.NvEntryTag attribute*), 115
NVEntry_FACTORY_RESERVED1 (*pycozmo.protocol_encoder.NvEntryTag attribute*), 115
NVEntry_FACTORY_RESERVED2 (*pycozmo.protocol_encoder.NvEntryTag attribute*), 115
NVEntry_FactoryBaseTag (*pycozmo.protocol_encoder.NvEntryTag attribute*), 115
NVEntry_FactoryBaseTagWithBCOffset (*pycozmo.protocol_encoder.NvEntryTag attribute*), 115
NVEntry_FactoryLock (*pycozmo.protocol_encoder.NvEntryTag attribute*), 115
NVEntry_GameSkillLevels (*pycozmo.protocol_encoder.NvEntryTag attribute*), 115
NVEntry_GameUnlocks (*pycozmo.protocol_encoder.NvEntryTag attribute*), 115
NVEntry_IMUAverages (*pycozmo.protocol_encoder.NvEntryTag attribute*), 115
NVEntry_IMUInfo (*pycozmo.protocol_encoder.NvEntryTag attribute*), 115
NVEntry_Invalid (*pycozmo.protocol_encoder.NvEntryTag attribute*), 115
NVEntry_InventoryData (*pycozmo.protocol_encoder.NvEntryTag attribute*), 115
NVEntry_LabAssignments (*pycozmo.protocol_encoder.NvEntryTag attribute*), 116
NVEntry_NEXT_SLOT (*pycozmo.protocol_encoder.NvEntryTag attribute*), 116
NVEntry_NurtureGameData (*pycozmo.protocol_encoder.NvEntryTag attribute*), 116
NVEntry_ObservedCubePose (*pycozmo.protocol_encoder.NvEntryTag attribute*), 116
NVEntry_OnboardingData (*pycozmo.protocol_encoder.NvEntryTag attribute*), 116
NVEntry_PlaypenTestResults (*pycozmo.protocol_encoder.NvEntryTag attribute*), 116
NVEntry_PrePlaypenCentroids (*pycozmo.protocol_encoder.NvEntryTag attribute*), 116
NVEntry_PrePlaypenResults (*pycozmo.protocol_encoder.NvEntryTag attribute*), 116
NVEntry_SavedCubeIDs (*pycozmo.protocol_encoder.NvEntryTag attribute*), 116
NVEntry_ToolCodeImageLeft (*pycozmo.protocol_encoder.NvEntryTag attribute*), 116
NVEntry_ToolCodeImageRight (*pycozmo.protocol_encoder.NvEntryTag attribute*), 116
NVEntry_ToolCodeInfo (*pycozmo.protocol_encoder.NvEntryTag attribute*), 116
NVEntry_VersionMagic (*pycozmo.protocol_encoder.NvEntryTag attribute*), 116
NvEntryTag (*class in pycozmo.protocol_encoder*), 115
NVOP_ERASE (*pycozmo.protocol_encoder.NvOperation attribute*), 116
NVOP_READ (*pycozmo.protocol_encoder.NvOperation attribute*), 116
NVOP_WIPEALL (*pycozmo.protocol_encoder.NvOperation attribute*), 116
NVOP_WRITE (*pycozmo.protocol_encoder.NvOperation attribute*), 116
NvOperation (*class in pycozmo.protocol_encoder*), 116
NvResult (*class in pycozmo.protocol_encoder*), 116
NvStorageOp (*class in pycozmo.protocol_encoder*), 117
NvStorageOpResult (*class in pycozmo.protocol_encoder*), 117

O

Object (*class in pycozmo.object*), 83
object_id (*pycozmo.protocol_encoder.CubeId attribute*), 102
object_id (*pycozmo.protocol_encoder.ObjectAccel attribute*), 118
object_id (*pycozmo.protocol_encoder.ObjectConnectionState attribute*), 119

object_id (*pycozmo.protocol_encoder.ObjectMoved attribute*), 120
object_id (*pycozmo.protocol_encoder.ObjectPowerLevel attribute*), 120
object_id (*pycozmo.protocol_encoder.ObjectStoppedMoving attribute*), 121
object_id (*pycozmo.protocol_encoder.ObjectTapFiltered attribute*), 121
object_id (*pycozmo.protocol_encoder.ObjectTapped attribute*), 122
object_id (*pycozmo.protocol_encoder.ObjectUpAxisChanged attribute*), 123
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*), 118
object_type (*pycozmo.protocol_encoder.ObjectConnectionState attribute*), 119
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*), 119
ObjectPowerLevel (*class in pycozmo.protocol_encoder*), 120
ObjectStoppedMoving (*class in pycozmo.protocol_encoder*), 120
ObjectTapFiltered (*class in pycozmo.protocol_encoder*), 121
ObjectTapped (*class in pycozmo.protocol_encoder*), 121
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*), 82
Off (*pycozmo.protocol_encoder.ImageSendMode attribute*), 112
off_color (*pycozmo.protocol_encoder.LightState attribute*), 112
off_frames (*pycozmo.protocol_encoder.LightState attribute*), 112
off_light (*in module pycozmo.lights*), 82
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*), 54
offset (*pycozmo.expressions.expressions.Tiredness attribute*), 56
offset (*pycozmo.expressions.expressions.Vulnerability attribute*), 49
offset (*pycozmo.procedural_face.ProceduralEye attribute*), 84
offset (*pycozmo.procedural_face.ProceduralFace at-*

tribute), 85
offset (*pycozmo.procedural_face.ProceduralLid attribute*), 83
offset (*pycozmo.protocol_encoder.LightState attribute*), 112
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*), 112
on_frames (*pycozmo.protocol_encoder.LightState attribute*), 112
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*), 117
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*), 89
Packet (*class in pycozmo.protocol_base*), 91
PacketType (*class in pycozmo.protocol_ast*), 86
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*), 84
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*), 124
PATH_INTERRUPTED (*pycozmo.protocol_encoder.PathEventType attribute*), 124
PATH_STARTED (*pycozmo.protocol_encoder.PathEventType attribute*), 124
PathEventType (*class in pycozmo.protocol_encoder*), 124

PathFollowingEvent (*class in pycozmo.protocol_encoder*), 124
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*), 86
PING (*pycozmo.protocol_ast.PacketType attribute*), 86
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*), 144
Pose (*class in pycozmo.util*), 145
pose_angle_rad (*pycozmo.protocol_encoder.RobotState attribute*), 127
pose_frame_id (*pycozmo.protocol_encoder.RobotState attribute*), 127
pose_frame_id (*pycozmo.protocol_encoder.SetOrigin attribute*), 130
pose_origin_id (*pycozmo.protocol_encoder.RobotState attribute*), 127
pose_origin_id (*pycozmo.protocol_encoder.SetOrigin attribute*), 130
pose_pitch_rad (*pycozmo.protocol_encoder.RobotState attribute*), 127
pose_x (*pycozmo.protocol_encoder.RobotState attribute*), 127
pose_x (*pycozmo.protocol_encoder.SetOrigin attribute*), 130
pose_y (*pycozmo.protocol_encoder.RobotState attribute*), 128
pose_y (*pycozmo.protocol_encoder.SetOrigin attribute*), 130
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*), 83
process_reaction() (*pycozmo.brain.Brain method*), 68
Protocol (*class in pycozmo.protocol_ast*), 90
PROTOCOL (*in module pycozmo.protocol_declaration*), 92
ProtocolGenerator (*class in pycozmo.protocol_generator*), 136
ProxObstacle (*pycozmo.protocol_encoder.ObjectType attribute*), 123
put() (*pycozmo.window.ReceiveWindow method*), 147
put() (*pycozmo.window.SendWindow method*), 147
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*), 70

pycozmo.emotions (*module*), 75
pycozmo.event (*module*), 76
pycozmo.exception (*module*), 79
pycozmo.expressions.expressions (*module*), 44
pycozmo.filter (*module*), 80
pycozmo.frame (*module*), 80
pycozmo.image_encoder (*module*), 81
pycozmo.lights (*module*), 82
pycozmo.object (*module*), 83
pycozmo.procedural_face (*module*), 83
pycozmo.protocol_ast (*module*), 85
pycozmo.protocol_base (*module*), 90
pycozmo.protocol_declaration (*module*), 92
pycozmo.protocol_encoder (*module*), 92
pycozmo.protocol_generator (*module*), 136
pycozmo.protocol_utils (*module*), 137
pycozmo.robot (*module*), 139
pycozmo.robot_debug (*module*), 141
pycozmo.run (*module*), 141
pycozmo.util (*module*), 142
pycozmo.window (*module*), 146
PyCozmoConnectionError, 79
PyCozmoException, 79

Q

q0 (*pycozmo.util.Quaternion attribute*), 145
q0_q1_q2_q3 (*pycozmo.util.Quaternion attribute*), 145
q1 (*pycozmo.util.Quaternion attribute*), 145
q2 (*pycozmo.util.Quaternion attribute*), 145
q3 (*pycozmo.util.Quaternion attribute*), 145
QQQQVGA (*pycozmo.protocol_encoder.ImageResolution attribute*), 111
QQQVGA (*pycozmo.protocol_encoder.ImageResolution attribute*), 111
QQVGA (*pycozmo.protocol_encoder.ImageResolution attribute*), 111
Quaternion (*class in pycozmo.util*), 145
QUXGA (*pycozmo.protocol_encoder.ImageResolution attribute*), 111
QVGA (*pycozmo.protocol_encoder.ImageResolution attribute*), 111
QXGA (*pycozmo.protocol_encoder.ImageResolution attribute*), 111

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*), 138
read_object_farray() (*pycozmo.protocol_utils.BinaryReader method*), 138
read_object_varray() (*pycozmo.protocol_utils.BinaryReader method*), 138
read_string() (*pycozmo.protocol_utils.BinaryReader method*), 138
read_string_farray() (*pycozmo.protocol_utils.BinaryReader method*), 138
read_string_varray() (*pycozmo.protocol_utils.BinaryReader method*), 138
read_varray() (*pycozmo.protocol_utils.BinaryReader method*), 138
ReceiveThread (*class in pycozmo.conn*), 71
ReceiveWindow (*class in pycozmo.window*), 147
RecordHeading (*class in pycozmo.protocol_encoder*), 126
red (*in module pycozmo.lights*), 82
red_light (*in module pycozmo.lights*), 82
reference_id (*pycozmo.audiokinetic.soundbank.EventAction attribute*), 41
Rejection (*class in pycozmo.expressions.expressions*), 54
render() (*in module pycozmo.image_encoder*), 81
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*), 84
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*), 100
RESET (*pycozmo.protocol_ast.FrameType attribute*), 86
reset() (*pycozmo.conn.ReceiveThread method*), 72
reset() (*pycozmo.conn.SendThread method*), 73
reset() (*pycozmo.window.BaseWindow method*), 146
reset() (*pycozmo.window.ReceiveWindow method*), 147
reset() (*pycozmo.window.SendWindow method*), 147
RESET_ACK (*pycozmo.protocol_ast.FrameType attribute*), 86
RESOLUTIONS (*in module pycozmo.camera*), 69
result (*pycozmo.protocol_encoder.NvStorageOpResult attribute*), 117
ROBOT (*pycozmo.protocol_ast.FrameType attribute*), 86
ROBOT_ADDR (*in module pycozmo.conn*), 71
RobotDelocalized (*class in pycozmo.protocol_encoder*), 126
RobotPoked (*class in pycozmo.protocol_encoder*), 126
RobotState (*class in pycozmo.protocol_encoder*), 127
RobotStatusFlag (*class in pycozmo.robot*), 140
RobotStatusFlagNames (*in module pycozmo.robot*), 140
rotation (*pycozmo.util.Pose attribute*), 146
rotation_period_frames (*pycozmo.protocol_encoder.CubeId attribute*), 102
rssi (*pycozmo.protocol_encoder.ObjectAvailable attribute*), 118
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_mmps2 (*pycozmo.protocol_encoder.DriveWheels attribute*), 104
rwheel_speed_mmps (*pycozmo.protocol_encoder.DriveWheels attribute*), 104
rwheel_speed_mmps (*pycozmo.protocol_encoder.RobotState attribute*), 128

S

Sadness (*class in pycozmo.expressions.expressions*), 45
samples (*pycozmo.protocol_encoder.OutputAudio attribute*), 124
scale_factor_lid_bend (*pycozmo.expressions.expressions.Amazement attribute*), 57
scale_factor_lid_bend (*pycozmo.expressions.expressions.Anger attribute*), 45

scale_factor_lid_bend (*pycozmo.expressions.expressions.Annoyance attribute*), 53
scale_factor_lid_bend (*pycozmo.expressions.expressions.Asleep attribute*), 56
scale_factor_lid_bend (*pycozmo.expressions.expressions.Boredom attribute*), 55
scale_factor_lid_bend (*pycozmo.expressions.expressions.Confusion attribute*), 57
scale_factor_lid_bend (*pycozmo.expressions.expressions.Despair attribute*), 50
scale_factor_lid_bend (*pycozmo.expressions.expressions.Disappointment attribute*), 51
scale_factor_lid_bend (*pycozmo.expressions.expressions.Disgust attribute*), 47
scale_factor_lid_bend (*pycozmo.expressions.expressions.Embarrassment attribute*), 51
scale_factor_lid_bend (*pycozmo.expressions.expressions.Excitement attribute*), 58
scale_factor_lid_bend (*pycozmo.expressions.expressions.Fear attribute*), 48
scale_factor_lid_bend (*pycozmo.expressions.expressions.Fury attribute*), 53
scale_factor_lid_bend (*pycozmo.expressions.expressions.Guilt attribute*), 50
scale_factor_lid_bend (*pycozmo.expressions.expressions.Happiness attribute*), 46
scale_factor_lid_bend (*pycozmo.expressions.expressions.Horror attribute*), 52
scale_factor_lid_bend (*pycozmo.expressions.expressions.Neutral attribute*), 45
scale_factor_lid_bend (*pycozmo.expressions.expressions.Pleading attribute*), 48
scale_factor_lid_bend (*pycozmo.expressions.expressions.Rejection attribute*), 55
scale_factor_lid_bend (*pycozmo.expressions.expressions.Sadness attribute*), 46
scale_factor_lid_bend (*pycozmo.expressions.expressions.Skepticism attribute*), 52
scale_factor_lid_bend (*pycozmo.expressions.expressions.Surprise attribute*), 47
scale_factor_lid_bend (*pycozmo.expressions.expressions.Suspicion attribute*), 54
scale_factor_lid_bend (*pycozmo.expressions.expressions.Tiredness attribute*), 56
scale_factor_lid_bend (*pycozmo.expressions.expressions.Vulnerability attribute*), 49
scale_factor_lid_bend (*pycozmo.procedural_face.ProceduralEye attribute*), 84
scale_factor_lid_bend (*pycozmo.procedural_face.ProceduralFace attribute*), 85
scale_factor_lid_bend (*pycozmo.procedural_face.ProceduralLid attribute*), 84
scale_factor_lid_height (*pycozmo.expressions.expressions.Amazement attribute*), 57
scale_factor_lid_height (*pycozmo.expressions.expressions.Anger attribute*), 45
scale_factor_lid_height (*pycozmo.expressions.expressions.Annoyance attribute*), 53
scale_factor_lid_height (*pycozmo.expressions.expressions.Asleep attribute*), 56
scale_factor_lid_height (*pycozmo.expressions.expressions.Boredom attribute*), 55
scale_factor_lid_height (*pycozmo.expressions.expressions.Confusion attribute*), 57
scale_factor_lid_height (*pycozmo.expressions.expressions.Despair attribute*), 50
scale_factor_lid_height (*pycozmo.expressions.expressions.Disappointment attribute*), 51
scale_factor_lid_height (*pycozmo.expressions.expressions.Disgust attribute*), 47
scale_factor_lid_height (*pycozmo.expressions.expressions.Embarrassment attribute*), 51

scale_factor_lid_height (*pycozmo.expressions.expressions.Excitement attribute*), 58
scale_factor_lid_height (*pycozmo.expressions.expressions.Fear attribute*), 48
scale_factor_lid_height (*pycozmo.expressions.expressions.Fury attribute*), 53
scale_factor_lid_height (*pycozmo.expressions.expressions.Guilt attribute*), 50
scale_factor_lid_height (*pycozmo.expressions.expressions.Happiness attribute*), 46
scale_factor_lid_height (*pycozmo.expressions.expressions.Horror attribute*), 52
scale_factor_lid_height (*pycozmo.expressions.expressions.Neutral attribute*), 45
scale_factor_lid_height (*pycozmo.expressions.expressions.Pleading attribute*), 48
scale_factor_lid_height (*pycozmo.expressions.expressions.Rejection attribute*), 55
scale_factor_lid_height (*pycozmo.expressions.expressions.Sadness attribute*), 46
scale_factor_lid_height (*pycozmo.expressions.expressions.Skepticism attribute*), 52
scale_factor_lid_height (*pycozmo.expressions.expressions.Surprise attribute*), 47
scale_factor_lid_height (*pycozmo.expressions.expressions.Suspicion attribute*), 54
scale_factor_lid_height (*pycozmo.expressions.expressions.Tiredness attribute*), 56
scale_factor_lid_height (*pycozmo.expressions.expressions.Vulnerability attribute*), 49
scale_factor_lid_height (*pycozmo.procedural_face.ProceduralEye attribute*), 84
scale_factor_lid_height (*pycozmo.procedural_face.ProceduralFace attribute*), 85
scale_factor_lid_height (*pycozmo.procedural_face.ProceduralLid attribute*), 84
scale_x (*pycozmo.expressions.expressions.Amazement attribute*), 57
scale_x (*pycozmo.expressions.expressions.Anger attribute*), 45
scale_x (*pycozmo.expressions.expressions.Annoyance attribute*), 53
scale_x (*pycozmo.expressions.expressions.Asleep attribute*), 56
scale_x (*pycozmo.expressions.expressions.Boredom attribute*), 55
scale_x (*pycozmo.expressions.expressions.Confusion attribute*), 57
scale_x (*pycozmo.expressions.expressions.Despair attribute*), 50
scale_x (*pycozmo.expressions.expressions.Disappointment attribute*), 51
scale_x (*pycozmo.expressions.expressions.Disgust attribute*), 47
scale_x (*pycozmo.expressions.expressions.Embarrassment attribute*), 51
scale_x (*pycozmo.expressions.expressions.Excitement attribute*), 58
scale_x (*pycozmo.expressions.expressions.Fear attribute*), 48
scale_x (*pycozmo.expressions.expressions.Fury attribute*), 54
scale_x (*pycozmo.expressions.expressions.Guilt attribute*), 50
scale_x (*pycozmo.expressions.expressions.Happiness attribute*), 46
scale_x (*pycozmo.expressions.expressions.Horror attribute*), 52
scale_x (*pycozmo.expressions.expressions.Neutral attribute*), 45
scale_x (*pycozmo.expressions.expressions.Pleading attribute*), 49
scale_x (*pycozmo.expressions.expressions.Rejection attribute*), 55
scale_x (*pycozmo.expressions.expressions.Sadness attribute*), 46
scale_x (*pycozmo.expressions.expressions.Skepticism attribute*), 52
scale_x (*pycozmo.expressions.expressions.Surprise attribute*), 47
scale_x (*pycozmo.expressions.expressions.Suspicion attribute*), 54
scale_x (*pycozmo.expressions.expressions.Tiredness attribute*), 56
scale_x (*pycozmo.expressions.expressions.Vulnerability attribute*), 49
scale_x (*pycozmo.procedural_face.ProceduralEye attribute*), 84
scale_x (*pycozmo.procedural_face.ProceduralFace attribute*), 85

scale_y (*pycozmo.expressions.expressions.Amazement attribute*), 57
scale_y (*pycozmo.expressions.expressions.Anger attribute*), 45
scale_y (*pycozmo.expressions.expressions.Annoyance attribute*), 53
scale_y (*pycozmo.expressions.expressions.Asleep attribute*), 56
scale_y (*pycozmo.expressions.expressions.Boredom attribute*), 55
scale_y (*pycozmo.expressions.expressions.Confusion attribute*), 57
scale_y (*pycozmo.expressions.expressions.Despair attribute*), 50
scale_y (*pycozmo.expressions.expressions.Disappointment attribute*), 51
scale_y (*pycozmo.expressions.expressions.Disgust attribute*), 47
scale_y (*pycozmo.expressions.expressions.Embarrassment attribute*), 51
scale_y (*pycozmo.expressions.expressions.Excitement attribute*), 58
scale_y (*pycozmo.expressions.expressions.Fear attribute*), 48
scale_y (*pycozmo.expressions.expressions.Fury attribute*), 54
scale_y (*pycozmo.expressions.expressions.Guilt attribute*), 50
scale_y (*pycozmo.expressions.expressions.Happiness attribute*), 46
scale_y (*pycozmo.expressions.expressions.Horror attribute*), 52
scale_y (*pycozmo.expressions.expressions.Neutral attribute*), 45
scale_y (*pycozmo.expressions.expressions.Pleading attribute*), 49
scale_y (*pycozmo.expressions.expressions.Rejection attribute*), 55
scale_y (*pycozmo.expressions.expressions.Sadness attribute*), 46
scale_y (*pycozmo.expressions.expressions.Skepticism attribute*), 52
scale_y (*pycozmo.expressions.expressions.Surprise attribute*), 47
scale_y (*pycozmo.expressions.expressions.Suspicion attribute*), 54
scale_y (*pycozmo.expressions.expressions.Tiredness attribute*), 56
scale_y (*pycozmo.expressions.expressions.Vulnerability attribute*), 49
scale_y (*pycozmo.procedural_face.ProceduralEye attribute*), 84
scale_y (*pycozmo.procedural_face.ProceduralFace attribute*), 85
scope (*pycozmo.audiokinetic.soundbank.EventAction attribute*), 41
seek_cur() (*pycozmo.protocol_utils.BinaryReader method*), 138
seek_set() (*pycozmo.protocol_utils.BinaryReader method*), 138
send() (*pycozmo.conn.Connection method*), 75
send() (*pycozmo.conn.SendThread method*), 73
SendThread (*class in pycozmo.conn*), 72
SendWindow (*class in pycozmo.window*), 147
seq (*pycozmo.frame.Frame attribute*), 81
seq (*pycozmo.protocol_base.Packet attribute*), 91
seq (*pycozmo.protocol_base.UnknownCommand attribute*), 92
seq (*pycozmo.protocol_base.UnknownEvent attribute*), 92
seq (*pycozmo.protocol_base.UnknownPacket attribute*), 91
seq (*pycozmo.protocol_encoder.AbortAnimation attribute*), 95
seq (*pycozmo.protocol_encoder.AcknowledgeAction attribute*), 95
seq (*pycozmo.protocol_encoder.AnimationEnded attribute*), 97
seq (*pycozmo.protocol_encoder.AnimationStarted attribute*), 97
seq (*pycozmo.protocol_encoder.AnimationState attribute*), 98
seq (*pycozmo.protocol_encoder.AnimBackpackLights attribute*), 95
seq (*pycozmo.protocol_encoder.AnimBody attribute*), 96
seq (*pycozmo.protocol_encoder.AnimHead attribute*), 96
seq (*pycozmo.protocol_encoder.AnimLift attribute*), 97
seq (*pycozmo.protocol_encoder.AppendPathSegArc attribute*), 99
seq (*pycozmo.protocol_encoder.AppendPathSegLine attribute*), 99
seq (*pycozmo.protocol_encoder.AppendPathSegPointTurn attribute*), 100
seq (*pycozmo.protocol_encoder.BodyInfo attribute*), 101
seq (*pycozmo.protocol_encoder.ButtonPressed attribute*), 101
seq (*pycozmo.protocol_encoder.ClearPath attribute*), 101
seq (*pycozmo.protocol_encoder.Connect attribute*), 102
seq (*pycozmo.protocol_encoder.CubeId attribute*), 102
seq (*pycozmo.protocol_encoder.CubeLights attribute*), 103
seq (*pycozmo.protocol_encoder.DebugData attribute*), 103
seq (*pycozmo.protocol_encoder.Disconnect attribute*), 104
seq (*pycozmo.protocol_encoder.DisplayImage attribute*), 104

seq (*pycozmo.protocol_encoder.DriveWheels attribute*), 104
seq (*pycozmo.protocol_encoder.Enable attribute*), 105
seq (*pycozmo.protocol_encoder.EnableAnimationState attribute*), 105
seq (*pycozmo.protocol_encoder.EnableCamera attribute*), 106
seq (*pycozmo.protocol_encoder.EnableColorImages attribute*), 106
seq (*pycozmo.protocol_encoder.EnableStopOnCliff attribute*), 106
seq (*pycozmo.protocol_encoder.EndAnimation attribute*), 107
seq (*pycozmo.protocol_encoder.ExecutePath attribute*), 107
seq (*pycozmo.protocol_encoder.FallingStarted attribute*), 107
seq (*pycozmo.protocol_encoder.FallingStopped attribute*), 108
seq (*pycozmo.protocol_encoder.FirmwareSignature attribute*), 108
seq (*pycozmo.protocol_encoder.FirmwareUpdate attribute*), 109
seq (*pycozmo.protocol_encoder.FirmwareUpdateResult attribute*), 109
seq (*pycozmo.protocol_encoder.HardwareInfo attribute*), 109
seq (*pycozmo.protocol_encoder.ImageChunk attribute*), 110
seq (*pycozmo.protocol_encoder.ImageImuData attribute*), 111
seq (*pycozmo.protocol_encoder.Keyframe attribute*), 112
seq (*pycozmo.protocol_encoder.LightStateCenter attribute*), 113
seq (*pycozmo.protocol_encoder.LightStateSide attribute*), 113
seq (*pycozmo.protocol_encoder.MotorCalibration attribute*), 114
seq (*pycozmo.protocol_encoder.MoveHead attribute*), 114
seq (*pycozmo.protocol_encoder.MoveLift attribute*), 115
seq (*pycozmo.protocol_encoder.NvStorageOp attribute*), 117
seq (*pycozmo.protocol_encoder.NvStorageOpResult attribute*), 117
seq (*pycozmo.protocol_encoder.ObjectAccel attribute*), 118
seq (*pycozmo.protocol_encoder.ObjectAvailable attribute*), 118
seq (*pycozmo.protocol_encoder.ObjectConnect attribute*), 119
seq (*pycozmo.protocol_encoder.ObjectConnectionState attribute*), 119
seq (*pycozmo.protocol_encoder.ObjectMoved attribute*), 120
seq (*pycozmo.protocol_encoder.ObjectPowerLevel attribute*), 120
seq (*pycozmo.protocol_encoder.ObjectStoppedMoving attribute*), 121
seq (*pycozmo.protocol_encoder.ObjectTapFiltered attribute*), 121
seq (*pycozmo.protocol_encoder.ObjectTapped attribute*), 122
seq (*pycozmo.protocol_encoder.ObjectUpAxisChanged attribute*), 123
seq (*pycozmo.protocol_encoder.OutputAudio attribute*), 124
seq (*pycozmo.protocol_encoder.OutputSilence attribute*), 124
seq (*pycozmo.protocol_encoder.PathFollowingEvent attribute*), 125
seq (*pycozmo.protocol_encoder.Ping attribute*), 125
seq (*pycozmo.protocol_encoder.RecordHeading attribute*), 126
seq (*pycozmo.protocol_encoder.RobotDelocalized attribute*), 126
seq (*pycozmo.protocol_encoder.RobotPoked attribute*), 126
seq (*pycozmo.protocol_encoder.RobotState attribute*), 128
seq (*pycozmo.protocol_encoder.SetAccessoryDiscovery attribute*), 128
seq (*pycozmo.protocol_encoder.SetCameraParams attribute*), 128
seq (*pycozmo.protocol_encoder.SetHeadAngle attribute*), 129
seq (*pycozmo.protocol_encoder.SetHeadLight attribute*), 129
seq (*pycozmo.protocol_encoder.SetLiftHeight attribute*), 130
seq (*pycozmo.protocol_encoder.SetOrigin attribute*), 130
seq (*pycozmo.protocol_encoder.SetRobotVolume attribute*), 131
seq (*pycozmo.protocol_encoder.ShutdownRobot attribute*), 131
seq (*pycozmo.protocol_encoder.StartAnimation attribute*), 132
seq (*pycozmo.protocol_encoder.StartMotorCalibration attribute*), 132
seq (*pycozmo.protocol_encoder.StopAllMotors attribute*), 132
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*), 109
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*), 144
set_head_angle() (*pycozmo.client.Client method*), 70
set_head_light() (*pycozmo.client.Client method*), 70
set_left() (*pycozmo.util.Matrix44 method*), 144
set_lift_height() (*pycozmo.client.Client method*), 70
set_pos() (*pycozmo.util.Matrix44 method*), 144
set_to() (*pycozmo.util.Vector3 method*), 143
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*), 73
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*), 108
SingleShot (*pycozmo.protocol_encoder.ImageSendMode attribute*), 112
Skepticism (*class in pycozmo.expressions.expressions*), 52
sleep() (*pycozmo.util.FPSTimer method*), 146
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_mmps (*pycozmo.protocol_encoder.AppendPathSegArc attribute*), 99
speed_mmps (*pycozmo.protocol_encoder.AppendPathSegLine attribute*), 99
speed_mmps (*pycozmo.protocol_encoder.AppendPathSegPointTurn attribute*), 100
speed_mmps (*pycozmo.protocol_encoder.PathSegmentSpeed attribute*), 125
speed_rad_per_sec (*pycozmo.protocol_encoder.MoveHead attribute*), 114
speed_rad_per_sec (*pycozmo.protocol_encoder.MoveLift attribute*), 115
speed_rad_per_sec (*pycozmo.protocol_encoder.TurnInPlace attribute*), 134
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*), 73
start_angle_rad (*pycozmo.protocol_encoder.AppendPathSegArc attribute*), 99
StartAnimation (*class in pycozmo.protocol_encoder*), 131
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*), 113
STATS_INTERVAL (*pycozmo.conn.Connection attribute*), 74
status (*pycozmo.protocol_encoder.FirmwareUpdateResult attribute*), 109
status (*pycozmo.protocol_encoder.ImageChunk attribute*), 110
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*), 73
stop_all_motors() (*pycozmo.client.Client method*), 70
StopAllMotors (*class in pycozmo.protocol_encoder*), 132
str_to_image() (*in module pycozmo.image_encoder*), 81
strategy (*pycozmo.activity.Activity attribute*), 59
Stream (*pycozmo.protocol_encoder.ImageSendMode attribute*), 112
StreamObjectAccel (*class in pycozmo.protocol_encoder*), 132
StringArgument (*class in pycozmo.protocol_ast*), 89
Struct (*class in pycozmo.protocol_ast*), 87
Struct (*class in pycozmo.protocol_base*), 90
Surprise (*class in pycozmo.expressions.expressions*), 46
Suspicion (*class in pycozmo.expressions.expressions*), 54
SVGA (*pycozmo.protocol_encoder.ImageResolution attribute*), 111
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*), 121
time_sent_ms (*pycozmo.protocol_encoder.Ping attribute*), 125
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*), 121
timestamp (*pycozmo.protocol_encoder.ObjectTapped attribute*), 122
timestamp (*pycozmo.protocol_encoder.ObjectUpAxisChanged attribute*), 123
timestamp (*pycozmo.protocol_encoder.RobotState attribute*), 128
timestamp (*pycozmo.protocol_encoder.SyncTime attribute*), 133
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*), 90
to_bytes() (*pycozmo.protocol_base.UnknownCommand method*), 92

to_bytes() (*pycozmo.protocol_base.UnknownEvent method*), 92
to_bytes() (*pycozmo.protocol_base.UnknownPacket method*), 91
to_bytes() (*pycozmo.protocol_encoder.AbortAnimation method*), 95
to_bytes() (*pycozmo.protocol_encoder.AcknowledgeAction method*), 95
to_bytes() (*pycozmo.protocol_encoder.AnimationEnded method*), 97
to_bytes() (*pycozmo.protocol_encoder.AnimationStarted method*), 97
to_bytes() (*pycozmo.protocol_encoder.AnimationState method*), 98
to_bytes() (*pycozmo.protocol_encoder.AnimBackpackLights method*), 95
to_bytes() (*pycozmo.protocol_encoder.AnimBody method*), 96
to_bytes() (*pycozmo.protocol_encoder.AnimHead method*), 96
to_bytes() (*pycozmo.protocol_encoder.AnimLift method*), 97
to_bytes() (*pycozmo.protocol_encoder.AppendPathSegArc method*), 99
to_bytes() (*pycozmo.protocol_encoder.AppendPathSegLine method*), 99
to_bytes() (*pycozmo.protocol_encoder.AppendPathSegPointTurn method*), 100
to_bytes() (*pycozmo.protocol_encoder.BodyInfo method*), 101
to_bytes() (*pycozmo.protocol_encoder.ButtonPressed method*), 101
to_bytes() (*pycozmo.protocol_encoder.ClearPath method*), 102
to_bytes() (*pycozmo.protocol_encoder.Connect method*), 102
to_bytes() (*pycozmo.protocol_encoder.CubeId method*), 102
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*), 104
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*), 106
to_bytes() (*pycozmo.protocol_encoder.EndAnimation method*), 107
to_bytes() (*pycozmo.protocol_encoder.ExecutePath method*), 107
to_bytes() (*pycozmo.protocol_encoder.FallingStarted method*), 107
to_bytes() (*pycozmo.protocol_encoder.FallingStopped method*), 108
to_bytes() (*pycozmo.protocol_encoder.FirmwareSignature method*), 108
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*), 110
to_bytes() (*pycozmo.protocol_encoder.ImageImuData method*), 111
to_bytes() (*pycozmo.protocol_encoder.Keyframe method*), 112
to_bytes() (*pycozmo.protocol_encoder.LightState method*), 112
to_bytes() (*pycozmo.protocol_encoder.LightStateCenter method*), 113
to_bytes() (*pycozmo.protocol_encoder.LightStateSide method*), 113
to_bytes() (*pycozmo.protocol_encoder.MotorCalibration method*), 114
to_bytes() (*pycozmo.protocol_encoder.MoveHead method*), 114
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*), 118
to_bytes() (*pycozmo.protocol_encoder.ObjectConnect method*), 119
to_bytes() (*pycozmo.protocol_encoder.ObjectConnectionState method*), 119
to_bytes() (*pycozmo.protocol_encoder.ObjectMoved method*), 120
to_bytes() (*pycozmo.protocol_encoder.ObjectPowerLevel method*), 120

to_bytes() (*pycozmo.protocol_encoder.ObjectStoppedMoving method*), 121
to_bytes() (*pycozmo.protocol_encoder.ObjectTapFiltered method*), 121
to_bytes() (*pycozmo.protocol_encoder.ObjectTapped method*), 122
to_bytes() (*pycozmo.protocol_encoder.ObjectUpAxisChanged method*), 123
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*), 125
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.SetCameraParams method*), 129
to_bytes() (*pycozmo.protocol_encoder.SetHeadAngle method*), 129
to_bytes() (*pycozmo.protocol_encoder.SetHeadLight method*), 129
to_bytes() (*pycozmo.protocol_encoder.SetLiftHeight method*), 130
to_bytes() (*pycozmo.protocol_encoder.SetOrigin method*), 130
to_bytes() (*pycozmo.protocol_encoder.SetRobotVolume method*), 131
to_bytes() (*pycozmo.protocol_encoder.ShutdownRobot method*), 131
to_bytes() (*pycozmo.protocol_encoder.StartAnimation method*), 132
to_bytes() (*pycozmo.protocol_encoder.StartMotorCalibration method*), 132
to_bytes() (*pycozmo.protocol_encoder.StopAllMotors method*), 132
to_bytes() (*pycozmo.protocol_encoder.StreamObjectAcked method*), 133
to_bytes() (*pycozmo.protocol_encoder.SyncTime method*), 133
to_bytes() (*pycozmo.protocol_encoder.TrimPath method*), 134
to_bytes() (*pycozmo.protocol_encoder.TurnInPlace method*), 134
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*), 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*), 146
to_matrix() (*pycozmo.util.Quaternion method*), 145
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*), 91
to_writer() (*pycozmo.protocol_encoder.AbortAnimation method*), 95
to_writer() (*pycozmo.protocol_encoder.AcknowledgeAction method*), 95
to_writer() (*pycozmo.protocol_encoder.AnimationEnded method*), 97
to_writer() (*pycozmo.protocol_encoder.AnimationStarted method*), 98
to_writer() (*pycozmo.protocol_encoder.AnimationState method*), 98
to_writer() (*pycozmo.protocol_encoder.AnimBackpackLights method*), 95
to_writer() (*pycozmo.protocol_encoder.AnimBody method*), 96
to_writer() (*pycozmo.protocol_encoder.AnimHead method*), 96
to_writer() (*pycozmo.protocol_encoder.AnimLift method*), 97
to_writer() (*pycozmo.protocol_encoder.AppendPathSegArc method*), 99
to_writer() (*pycozmo.protocol_encoder.AppendPathSegLine method*), 99
to_writer() (*pycozmo.protocol_encoder.AppendPathSegPointTurn method*), 100
to_writer() (*pycozmo.protocol_encoder.BodyInfo method*), 101
to_writer() (*pycozmo.protocol_encoder.ButtonPressed method*), 101
to_writer() (*pycozmo.protocol_encoder.ClearPath method*), 102
to_writer() (*pycozmo.protocol_encoder.Connect method*), 102
to_writer() (*pycozmo.protocol_encoder.CubeId method*), 102
to_writer() (*pycozmo.protocol_encoder.CubeLights method*), 103
to_writer() (*pycozmo.protocol_encoder.DebugData method*), 103
to_writer() (*pycozmo.protocol_encoder.Disconnect method*), 104
to_writer() (*pycozmo.protocol_encoder.DisplayImage method*), 104
to_writer() (*pycozmo.protocol_encoder.DriveWheels method*), 104
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*), 106
to_writer() (*pycozmo.protocol_encoder.EndAnimation method*), 107
to_writer() (*pycozmo.protocol_encoder.ExecutePath method*), 107
to_writer() (*pycozmo.protocol_encoder.FallingStarted method*), 107
to_writer() (*pycozmo.protocol_encoder.FallingStopped method*), 108
to_writer() (*pycozmo.protocol_encoder.FirmwareSignature method*), 108
to_writer() (*pycozmo.protocol_encoder.FirmwareUpdate method*), 109
to_writer() (*pycozmo.protocol_encoder.FirmwareUpdateResult method*), 109
to_writer() (*pycozmo.protocol_encoder.HardwareInfo method*), 110
to_writer() (*pycozmo.protocol_encoder.ImageChunk method*), 110
to_writer() (*pycozmo.protocol_encoder.ImageImuData*

method), 111
to_writer() (*pycozmo.protocol_encoder.Keyframe method*), 112
to_writer() (*pycozmo.protocol_encoder.LightState method*), 112
to_writer() (*pycozmo.protocol_encoder.LightStateCenter method*), 113
to_writer() (*pycozmo.protocol_encoder.LightStateSide method*), 113
to_writer() (*pycozmo.protocol_encoder.MotorCalibration method*), 114
to_writer() (*pycozmo.protocol_encoder.MoveHead method*), 114
to_writer() (*pycozmo.protocol_encoder.MoveLift method*), 115
to_writer() (*pycozmo.protocol_encoder.NvStorageOp method*), 117
to_writer() (*pycozmo.protocol_encoder.NvStorageOpResult method*), 118
to_writer() (*pycozmo.protocol_encoder.ObjectAccel method*), 118
to_writer() (*pycozmo.protocol_encoder.ObjectAvailable method*), 119
to_writer() (*pycozmo.protocol_encoder.ObjectConnect method*), 119
to_writer() (*pycozmo.protocol_encoder.ObjectConnectionState method*), 119
to_writer() (*pycozmo.protocol_encoder.ObjectMoved method*), 120
to_writer() (*pycozmo.protocol_encoder.ObjectPowerLevel method*), 120
to_writer() (*pycozmo.protocol_encoder.ObjectStoppedMoving method*), 121
to_writer() (*pycozmo.protocol_encoder.ObjectTapFiltered method*), 121
to_writer() (*pycozmo.protocol_encoder.ObjectTapped method*), 122
to_writer() (*pycozmo.protocol_encoder.ObjectUpAxisChanged method*), 123
to_writer() (*pycozmo.protocol_encoder.OutputAudio method*), 124
to_writer() (*pycozmo.protocol_encoder.OutputSilence method*), 124
to_writer() (*pycozmo.protocol_encoder.PathFollowingEvent method*), 125
to_writer() (*pycozmo.protocol_encoder.PathSegmentSpeed method*), 125
to_writer() (*pycozmo.protocol_encoder.Ping method*), 125
to_writer() (*pycozmo.protocol_encoder.RecordHeading method*), 126
to_writer() (*pycozmo.protocol_encoder.RobotDelocalized method*), 126
to_writer() (*pycozmo.protocol_encoder.RobotPoked method*), 127
to_writer() (*pycozmo.protocol_encoder.RobotState method*), 128
to_writer() (*pycozmo.protocol_encoder.SetAccessoryDiscovery method*), 128
to_writer() (*pycozmo.protocol_encoder.SetCameraParams method*), 129
to_writer() (*pycozmo.protocol_encoder.SetHeadAngle method*), 129
to_writer() (*pycozmo.protocol_encoder.SetHeadLight method*), 129
to_writer() (*pycozmo.protocol_encoder.SetLiftHeight method*), 130
to_writer() (*pycozmo.protocol_encoder.SetOrigin method*), 130
to_writer() (*pycozmo.protocol_encoder.SetRobotVolume method*), 131
to_writer() (*pycozmo.protocol_encoder.ShutdownRobot method*), 131
to_writer() (*pycozmo.protocol_encoder.StartAnimation method*), 132
to_writer() (*pycozmo.protocol_encoder.StartMotorCalibration method*), 132
to_writer() (*pycozmo.protocol_encoder.StopAllMotors method*), 132
to_writer() (*pycozmo.protocol_encoder.StreamObjectAccel method*), 133
to_writer() (*pycozmo.protocol_encoder.SyncTime method*), 133
to_writer() (*pycozmo.protocol_encoder.TrimPath method*), 134
to_writer() (*pycozmo.protocol_encoder.TurnInPlace method*), 134
to_writer() (*pycozmo.protocol_encoder.TurnInPlaceAtSpeed method*), 135
to_writer() (*pycozmo.protocol_encoder.TurnToRecordedHeading method*), 135
to_writer() (*pycozmo.protocol_encoder.WifiOff method*), 136
to_x (*pycozmo.protocol_encoder.AppendPathSegLine attribute*), 99
to_y (*pycozmo.protocol_encoder.AppendPathSegLine attribute*), 99
TRACK_WIDTH (*in module pycozmo.robot*), 140
transition_off_frames (*pycozmo.protocol_encoder.LightState attribute*), 112
transition_on_frames (*pycozmo.protocol_encoder.LightState attribute*), 113
TrimPath (*class in pycozmo.protocol_encoder*), 133
TurnInPlace (*class in pycozmo.protocol_encoder*), 134
TurnInPlaceAtSpeed (*class in py-*

cozmo.protocol_encoder), 134
TurnToRecordedHeading (*class in pycozmo.protocol_encoder*), 135
type (*pycozmo.activity.Activity attribute*), 59
type (*pycozmo.audiokinetic.soundbank.EventAction attribute*), 41
type (*pycozmo.audiokinetic.soundbank.SFX attribute*), 42
type (*pycozmo.frame.Frame attribute*), 81
type (*pycozmo.protocol_base.Packet attribute*), 91
type (*pycozmo.protocol_base.UnknownCommand attribute*), 92
type (*pycozmo.protocol_base.UnknownEvent attribute*), 92
type (*pycozmo.protocol_base.UnknownPacket attribute*), 91
type (*pycozmo.protocol_encoder.AbortAnimation attribute*), 95
type (*pycozmo.protocol_encoder.AcknowledgeAction attribute*), 95
type (*pycozmo.protocol_encoder.AnimationEnded attribute*), 97
type (*pycozmo.protocol_encoder.AnimationStarted attribute*), 98
type (*pycozmo.protocol_encoder.AnimationState attribute*), 98
type (*pycozmo.protocol_encoder.AnimBackpackLights attribute*), 96
type (*pycozmo.protocol_encoder.AnimBody attribute*), 96
type (*pycozmo.protocol_encoder.AnimHead attribute*), 96
type (*pycozmo.protocol_encoder.AnimLift attribute*), 97
type (*pycozmo.protocol_encoder.AppendPathSegArc attribute*), 99
type (*pycozmo.protocol_encoder.AppendPathSegLine attribute*), 99
type (*pycozmo.protocol_encoder.AppendPathSegPointTurn attribute*), 100
type (*pycozmo.protocol_encoder.BodyInfo attribute*), 101
type (*pycozmo.protocol_encoder.ButtonPressed attribute*), 101
type (*pycozmo.protocol_encoder.ClearPath attribute*), 102
type (*pycozmo.protocol_encoder.Connect attribute*), 102
type (*pycozmo.protocol_encoder.CubeId attribute*), 102
type (*pycozmo.protocol_encoder.CubeLights attribute*), 103
type (*pycozmo.protocol_encoder.DebugData attribute*), 103
type (*pycozmo.protocol_encoder.Disconnect attribute*), 104
type (*pycozmo.protocol_encoder.DisplayImage attribute*), 104
type (*pycozmo.protocol_encoder.DriveWheels attribute*), 104
type (*pycozmo.protocol_encoder.Enable attribute*), 105
type (*pycozmo.protocol_encoder.EnableAnimationState attribute*), 105
type (*pycozmo.protocol_encoder.EnableCamera attribute*), 106
type (*pycozmo.protocol_encoder.EnableColorImages attribute*), 106
type (*pycozmo.protocol_encoder.EnableStopOnCliff attribute*), 106
type (*pycozmo.protocol_encoder.EndAnimation attribute*), 107
type (*pycozmo.protocol_encoder.ExecutePath attribute*), 107
type (*pycozmo.protocol_encoder.FallingStarted attribute*), 107
type (*pycozmo.protocol_encoder.FallingStopped attribute*), 108
type (*pycozmo.protocol_encoder.FirmwareSignature attribute*), 108
type (*pycozmo.protocol_encoder.FirmwareUpdate attribute*), 109
type (*pycozmo.protocol_encoder.FirmwareUpdateResult attribute*), 109
type (*pycozmo.protocol_encoder.HardwareInfo attribute*), 110
type (*pycozmo.protocol_encoder.ImageChunk attribute*), 110
type (*pycozmo.protocol_encoder.ImageImuData attribute*), 111
type (*pycozmo.protocol_encoder.Keyframe attribute*), 112
type (*pycozmo.protocol_encoder.LightStateCenter attribute*), 113
type (*pycozmo.protocol_encoder.LightStateSide attribute*), 113
type (*pycozmo.protocol_encoder.MotorCalibration attribute*), 114
type (*pycozmo.protocol_encoder.MoveHead attribute*), 114
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 at-*

tribute), 119
type (*pycozmo.protocol_encoder.ObjectConnectionState attribute*), 119
type (*pycozmo.protocol_encoder.ObjectMoved attribute*), 120
type (*pycozmo.protocol_encoder.ObjectPowerLevel attribute*), 120
type (*pycozmo.protocol_encoder.ObjectStoppedMoving attribute*), 121
type (*pycozmo.protocol_encoder.ObjectTapFiltered attribute*), 121
type (*pycozmo.protocol_encoder.ObjectTapped attribute*), 122
type (*pycozmo.protocol_encoder.ObjectUpAxisChanged attribute*), 123
type (*pycozmo.protocol_encoder.OutputAudio attribute*), 124
type (*pycozmo.protocol_encoder.OutputSilence attribute*), 124
type (*pycozmo.protocol_encoder.PathFollowingEvent attribute*), 125
type (*pycozmo.protocol_encoder.Ping attribute*), 125
type (*pycozmo.protocol_encoder.RecordHeading attribute*), 126
type (*pycozmo.protocol_encoder.RobotDelocalized attribute*), 126
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*), 129
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*), 131
type (*pycozmo.protocol_encoder.StartAnimation attribute*), 132
type (*pycozmo.protocol_encoder.StartMotorCalibration attribute*), 132
type (*pycozmo.protocol_encoder.StopAllMotors attribute*), 132
type (*pycozmo.protocol_encoder.StreamObjectAccel attribute*), 133
type (*pycozmo.protocol_encoder.SyncTime attribute*), 133
type (*pycozmo.protocol_encoder.TrimPath attribute*), 134
type (*pycozmo.protocol_encoder.TurnInPlace attribute*), 134
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*), 87
type_hint() (*pycozmo.protocol_ast.Command method*), 90
type_hint() (*pycozmo.protocol_ast.Connect method*), 89
type_hint() (*pycozmo.protocol_ast.Disconnect method*), 89
type_hint() (*pycozmo.protocol_ast.DoubleArgument method*), 87
type_hint() (*pycozmo.protocol_ast.EnumArgument method*), 88
type_hint() (*pycozmo.protocol_ast.Event method*), 90
type_hint() (*pycozmo.protocol_ast.FArrayArgument method*), 89
type_hint() (*pycozmo.protocol_ast.FloatArgument method*), 87
type_hint() (*pycozmo.protocol_ast.Int16Argument method*), 88
type_hint() (*pycozmo.protocol_ast.Int32Argument method*), 88
type_hint() (*pycozmo.protocol_ast.Int8Argument method*), 88
type_hint() (*pycozmo.protocol_ast.IntArgument method*), 88
type_hint() (*pycozmo.protocol_ast.Keyframe method*), 90
type_hint() (*pycozmo.protocol_ast.Packet method*), 89
type_hint() (*pycozmo.protocol_ast.Ping method*), 90
type_hint() (*pycozmo.protocol_ast.StringArgument method*), 89
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*), 87
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*), 87
UIntArgument (*class in pycozmo.protocol_ast*), 87
UNKNOWN (*pycozmo.protocol_ast.PacketType attribute*), 87
unknown (*pycozmo.protocol_encoder.AnimBody attribute*), 96
unknown (*pycozmo.protocol_encoder.AppendPathSegPointTurn attribute*), 100
UNKNOWN (*pycozmo.protocol_encoder.BodyColor attribute*), 100
unknown (*pycozmo.protocol_encoder.ClearPath attribute*), 102
unknown (*pycozmo.protocol_encoder.ExecutePath attribute*), 107
unknown (*pycozmo.protocol_encoder.FallingStarted attribute*), 107
unknown (*pycozmo.protocol_encoder.FallingStopped attribute*), 108
unknown (*pycozmo.protocol_encoder.FirmwareSignature attribute*), 108
unknown (*pycozmo.protocol_encoder.LightStateCenter attribute*), 113
unknown (*pycozmo.protocol_encoder.LightStateSide attribute*), 113
unknown (*pycozmo.protocol_encoder.NvStorageOp attribute*), 117
unknown (*pycozmo.protocol_encoder.Ping attribute*), 126
unknown (*pycozmo.protocol_encoder.SyncTime attribute*), 133
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*), 134
unknown5 (*pycozmo.protocol_encoder.SetOrigin attribute*), 131
unknown5 (*pycozmo.protocol_encoder.TurnInPlace attribute*), 134
UnknownAxis (*pycozmo.protocol_encoder.UpAxis attribute*), 135
UnknownCommand (*class in pycozmo.protocol_base*), 91
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*), 103
up_xyz (*pycozmo.util.Matrix44 attribute*), 145
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*), 84
upper_inner_radius_y (*pycozmo.procedural_face.ProceduralEye attribute*), 84
upper_outer_radius_x (*pycozmo.procedural_face.ProceduralEye attribute*), 84
upper_outer_radius_y (*pycozmo.procedural_face.ProceduralEye attribute*), 84
use_head_angle (*pycozmo.anim.AnimationGroupMember attribute*), 60
UXGA (*pycozmo.protocol_encoder.ImageResolution attribute*), 112

V

validate_bool() (*in module pycozmo.protocol_utils*), 137
validate_farray() (*in module pycozmo.protocol_utils*), 138
validate_float() (*in module pycozmo.protocol_utils*), 137
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
variability_deg (*pycozmo.protocol_encoder.AnimHead attribute*), 96
variability_mm (*pycozmo.protocol_encoder.AnimLift attribute*), 97
VArrayArgument (*class in pycozmo.protocol_ast*), 89
Vector3 (*class in pycozmo.util*), 143

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*), 70
weight (*pycozmo.anim.AnimationGroupMember attribute*), 60
wheel_accel_mmps2 (*pycozmo.protocol_encoder.TurnInPlaceAtSpeed attribute*), 135
wheel_speed_mmps (*pycozmo.protocol_encoder.TurnInPlaceAtSpeed attribute*), 135
white (*in module pycozmo.lights*), 82
white_light (*in module pycozmo.lights*), 82
WHITE_v10 (*pycozmo.protocol_encoder.BodyColor attribute*), 100
WHITE_v15 (*pycozmo.protocol_encoder.BodyColor attribute*), 100
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*), 84
width (*pycozmo.procedural_face.ProceduralFace attribute*), 85
width (*pycozmo.procedural_face.ProceduralLid attribute*), 84
WifiOff (*class in pycozmo.protocol_encoder*), 135
with_traceback() (*pycozmo.audiokinetic.exception.AudioKineticBaseError method*), 40
with_traceback() (*pycozmo.audiokinetic.exception.AudioKineticFormatError method*), 40
with_traceback() (*pycozmo.audiokinetic.exception.AudioKineticIOError method*), 40
with_traceback() (*pycozmo.exception.ConnectionTimeout method*), 79
with_traceback() (*pycozmo.exception.NoSpace method*), 80
with_traceback() (*pycozmo.exception.PyCozmoConnectionError method*), 79
with_traceback() (*pycozmo.exception.PyCozmoException method*),

79
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), 139

X

x (pycozmo.protocol_encoder.AppendPathSegPointTurn attribute), 100
x (pycozmo.util.Vector3 attribute), 143
x_offset (pycozmo.procedural_face.ProceduralEye attribute), 85
x_y_z (pycozmo.util.Vector3 attribute), 143
XGA (pycozmo.protocol_encoder.ImageResolution attribute), 112
XNegative (pycozmo.protocol_encoder.UpAxis attribute), 135
XPositive (pycozmo.protocol_encoder.UpAxis attribute), 135

Y

y (pycozmo.procedural_face.ProceduralLid attribute), 84
y (pycozmo.protocol_encoder.AppendPathSegPointTurn attribute), 100
y (pycozmo.util.Vector3 attribute), 143
y_offset (pycozmo.procedural_face.ProceduralLid attribute), 84
YNegative (pycozmo.protocol_encoder.UpAxis attribute), 135
YPositive (pycozmo.protocol_encoder.UpAxis attribute), 135
YUYV (pycozmo.protocol_encoder.ImageEncoding attribute), 111

Z

z (pycozmo.util.Vector3 attribute), 143
ZNegative (pycozmo.protocol_encoder.UpAxis attribute), 135
ZPositive (pycozmo.protocol_encoder.UpAxis attribute), 135