PSYONIC

ABILITY HAND™ API I²C Reference

Introduction

The hand has 2 different packet structures and 4 different control modes. The first covers the ‘industry standard’ *Grip Control Mode* interface, consisting of a 3-byte transmission and 7-byte reception, which allows the master device to select a grip index and speed, and read back some basic status information. The second packet structure covers the *Finger Control Mode* interface, which allows for individual control of the fingers and direct pressure sensor readings. This has a 25-byte transmission packet and a reception packet ranging from 24-64 bytes in length.

Wiring:

The 6-wire interface to the hand is provided through a Molex 0874380643 connector. The pinout is as follows:

Pin 1: **Power**
Pin 2: **Ground**
Pin 3: **Analog Input 1**
Pin 4: **Analog Input 2**
Pin 5: **SDA**
Pin 6: **SCL**

1 Basic I2C Specification:

The hand supports both standard (100kHz) and fast (400kHz) I2C modes. When operating in ‘fast mode’, the cable length of the hand should be limited to preserve packet integrity. In general, it is recommended to keep the I2C cable length as short as possible between the master and slave devices. The slave address of the hand is always 0x50. The hand has built in 10k pullup resistors, so external I2C pullup resistors are not necessary for a functional interface.

Grip Control Mode

Packet Structure:
Transmissions: (From Master to Hand)

S	Addr (0x50) - R/W bit low	A	command type	command	speed	P

1. *Hand slave address:*
   0x50
2. *Command type:*

Value	Command	Description
0x1D	HAND_SET_GRASP	Normal control mode. Hand will parse data bytes for grip control
0x48	HAND_STATUS	Handshaking. Hand will reply with 0xDEADBEEF

3. *Grip Commands: (Valid for packets starting with 0x1D)*

Value	Command	Description
0x00	Open	Opens the hand. Open configuration will vary depending on the last grip.
0x01	Power	All digits flexed with the thumb rotated internally.
0x02	Key	Also known as lateral grasp. When key grasp is opened, only the thumb flexor extends and all other digits remain flexed.
0x03	Pinch	Precision pinch with index finger and thumb.
0x04	Tripod Opened	3-jaw chuck where the pinky and ring fingers are extended.
0x05	Sign of the Horns	\m/
0x06	Cylinder	Thumb will touch the middle and ring fingers
0x07	Mouse Grasp	Grip which allows the use of a computer mouse
0x08	Power/Key Switch	When the thumb is rotated internally, the hand will close into a power grasp. When the thumb is rotated externally, the hand will close into a key grasp
0x09	Point	Index finger point with the thumb rotated internally and flexed.
0x0A	Rude Point	Middle finger point

0x0B	Hook Grasp	Static finger hooked configuration
0x0C	Relax Grip	Puts the hand in a ‘relaxed’ configuration.
0x0D	Sleeve Grip	Puts the thumb out of the way so the hand can be guided through the sleeve of a shirt or jacket
0x0E	Peace Grip	Makes an approximation of a ‘peace’ hand sign
0x0F	Tripod Closed	3-jaw chuck where the ring and pinky are flexed
0x10	Hangloose Grip	Makes the ‘hangloose’ hand gesture
0x11	Handshake Grip	Halfway between a key grip and a power grip
0x12	Fixed Pinch Grip	Pinch where the thumb doesn’t move while opening
0x13	User Grip 7	User defined grip
0x14	User Grip 8	User defined grip
0x15	User Grip 9	User defined grip
0x16	User Grip 10	User defined grip
0x17	User Grip 11	User defined grip
0x18	Trigger Grip	For grabbing/pulling trigger on spray bottles
0x19	User Grip 12	User defined grip
0x1A	User Grip 13	User defined grip
0x1B	User Grip 14	User defined grip
0x1C	User Grip 15	User defined grip
0x1D	User Grip 16	User defined grip
0x1E	User Grip 17	User defined grip
0x1F	User Grip 18	User defined grip
0x20	Finger Wave	If enabled (enabled by default) this grip will make the fingers move periodically and out of phase in a ‘waggle’ pattern. If disabled, this is a slot for another user defined grip.

*Note: all ‘User defined grips’ will result in no motion if selected, until modified using the PSYONIC mobile app or otherwise through the BLE interface

4. *Speed:*

4.1 When this byte is 0, the hand will stop.

4.2 When this byte is between 1 and 254, the **finger period** will vary linearly between 2 seconds and .29 seconds. A value of 1 will set the finger period to 2 seconds, and a value of 254 will set the finger period to .29 seconds. Finger period is defined as the allotted time for a finger to move through its motion path, and multiple periods can elapse over the course of a given grip.

4.3 When this byte is 255, the finger period will be set to .2 seconds.

Receptions: (From Hand to Master)

1. HAND_SET_GRASP mode (byte 0 of last transmission was 0x1D):

S	Addr (0x50) - R/W bit low	A	S	Addr (0x50) - R/W bit high	A	Bytes 0-1: vibration motor intensity	Bytes 2-3: 0xBEEF (for handshaking)	Bytes 4-6 LED color (for PSYONIC IO board)	Byte 7: grip status	Byte 8: grip percentage	P

Bytes 0-1 comprise a 16-bit integer that maps to the maximum pressure sensor value, for use in controlling our vibration motor.

Byte 7 will return 0xFF when the grip is in progress, and for closed grips, the grip command of the last requested grip from the master when the grip is completed. For opened grips, due to legacy support of an old COAPT protocol, Byte 7 may sometimes be zero, and sometimes be a non-zero value not equal to the current grasp command. A value not equal to 0xFF will indicate that the hand has completed its motion.

Tap Notification:

Packet Structure:
Transmissions: (From Master to Hand)

Tap messages are to be sent by a valid PSYONIC IO board. The full 4 byte message is sent once per tap, and each 4 byte word corresponds to a different tap type.

Single Tap:

S	Addr (0x50) - R/W bit low	A	Header Word: 0x5A	0x55	0x59	0xA6	P

Default action: no action

Double Tap:

S	Addr (0x50) - R/W bit low	A	Header Word: 0x5A	0xA5	0x7B	0x36	P

Default action: the hand will switch to the next grip in the direct control grip set (4 grips, circular buffer).

Triple Tap:

S	Addr (0x50) - R/W bit low	A	Header Word: 0x5A	0x55	0xDA	0x55	P

Default action: the hand will enter ‘freeze mode’. Direct control inputs are temporarily ignored while in freeze mode, and the hand will indicate freeze mode by blinking the requested RGB led color.

2 Extended I2C Interface

Transmission Packet Format (Master to Hand)

-			Byte 0	Bytes 1 - 24	Byte 25	-
S	Addr (0x50) - R/W bit low	A	Control Mode	Tracking values (4-byte floating point format, little endian)	Checksum	P

Each transmission packet has a single header/prefix byte which determines the control mode that will be entered. The remaining 24 bytes will contain floating 6 groups of 4 bytes, where each set of 4 bytes encodes a floating-point number (IEEE 754 single precision) encoded in little endian (as it is stored in ARM/x86 program memory).

Bytes 1-4	Bytes 5-8	Bytes 9-12	Bytes 13-16	Bytes 17-20	Bytes 21-24
Index setpoint	Middle setpoint	Ring setpoint	Pinky setpoint	Thumb Flexor setpoint	Thumb Rotator setpoint

Control Mode

Value	Function	Description
0xAD	Position Control Mode	Floating point numbers sent in the rest of the packet interpreted as finger position
0xAB	Torque Control Mode	Floating point numbers sent in the rest of the packet interpreted as finger torque
0xAC	Velocity Control Mode	Floating point numbers sent in the rest of the packet interpreted as finger velocity

*Note: packets with these headers ignore all bytes that follow. Packet can be terminated after sending just this byte and the setting will be applied. PoR: Power-on Reset
**Note: this header is only parsed after the hand has entered one of the 3 motion control modes

Mode Descriptions:

1. Position control mode:

In this mode, setpoints are interpreted as finger tracking positions in degrees, zero-referenced to the 'open hand' stall configuration which is measured during the calibration routine on startup. For the index, middle, ring, pinky and thumb flexor motors, only positive values should be used (negative values are out of range/will stall open), whereas for the thumb rotator only negative values should be used (positive values are out of range/will stall open).

2. *Torque control mode:*

In this mode, setpoints are interpreted as motor currents, which are tracked using field oriented control. Input range is -6 to 6.

3. *Velocity control mode:*

In this mode, setpoints are interpreted as motor velocities, in units of degrees/second. Velocity tracking is internally performed with position control, where the internally tracked position is increased or decreased at a rate specified by the velocity setpoint.

Note: All control modes have a safety procedure in place which prevents fingers from getting too close to their mechanical open/close limits.

4. *Safety:*

The API has built in motor overtemperature protection that limits the torque available to the motors. Motor movement is never fully disabled, but the maximum available motor torque is limited as a linear function of the heuristically estimated motor temperature. The higher the temperature, the lower the available torque. The lowest possible motor threshold is 22% of maximum torque. This rule is enforced in the hand firmware.

Reception Packet Format (Hand to Master)

-						Bytes 0-23	Bytes 24-68	Byte 69	Byte 70	-
S	Addr (0x50) - R/W bit low	A	S	Addr (0x50) - R/W bit high	A	Finger Positions (4 byte floating point format, little endian	Pressure sensors (30 values, 12bit, packed)	Bitmask indicating finger enable/disable status	Checksum	P

Finger position (bytes 0-23) format:
Each reception packet will begin by returning finger positions in degrees, zero-referenced from the open stall configuration obtained on startup. The hand will always return data in this format when in position, velocity, or torque control modes. For the index, middle, ring, pinky, and thumb flexor only positive angles are achievable, and for the thumb rotator only negative angles are achievable (if the open stall routine on startup completed successfully). Positions are encoded in 6 groups of 4 bytes, little endian, in IEEE 754 single precision floating point format. The byte order is specified in the table below:

Bytes 0-3	Bytes 4-7	Bytes 8-11	Bytes 12-15	Bytes 16-29	Bytes 20-23
Index position	Middle position	Ring position	Pinky position	Thumb flexor position	Thumb rotator position

Pressure data (bytes 24-68) format:
Pressure sensor data is encoded in the remaining 45 bytes. Pressure data is packed into these bytes as 12 bit unsigned integers. Note that each word is broken up and packed with little-endianness. The endian-ness of the receiving system should be considered when unpacking this range of bytes.

Byte 0	Byte 1	Byte 2	Byte 3	...
bits 11-4 of index site 1	bits 3-0 of index site 1, and bits 11-8 of the index site 2	bits 7-0 of index site 2	bits 11-4 of index site 3	...

The following C code will decode this range of bytes when run on a little-endian system.

```c
/*
* Load packed 12 bit values located in an 8bit array into
* an unpacked (zero padded) 16 bit array. FSR utility function
*/
void unpack_8bit_into_12bit(uint8_t* arr, uint16_t* vals, int valsize)
{
        for(int i = 0; i < valsize; i++)
                vals[i] = 0;    //clear the buffer before loading it with |=
    for (int bidx = valsize * 12 - 4; bidx >= 0; bidx -= 4)
    {
        int validx = bidx / 12;
        int arridx = bidx / 8;
        int shift_val = (bidx % 8);
        vals[validx] |= ((arr[arridx] >> shift_val) & 0x0F) << (bidx % 12);
    }
}
```

Safety protocol status (byte 69) format:
The status of the torque limiting safety function is exposed through the api in the form of a bitmask, which indicates whether a motor is ‘hot’ or ‘cold’. If the temperature has exceeded an internally defined threshold, the finger is deemed ‘hot’, and if it dips below a different internally defined threshold, it is deemed ‘cold’. The bitmask format is detailed in the table below:

Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
Index	Middle	Ring	Pinky	Thumb Flexor	Thumb Rotator	(Ignored)	(Ignored)

A value of ‘1’ indicates hot and a value of ‘0’ indicates cold.