Q-Pixie Datasheet

N20 Gearmotor Motor Controller with Quadrature Feedback

Description

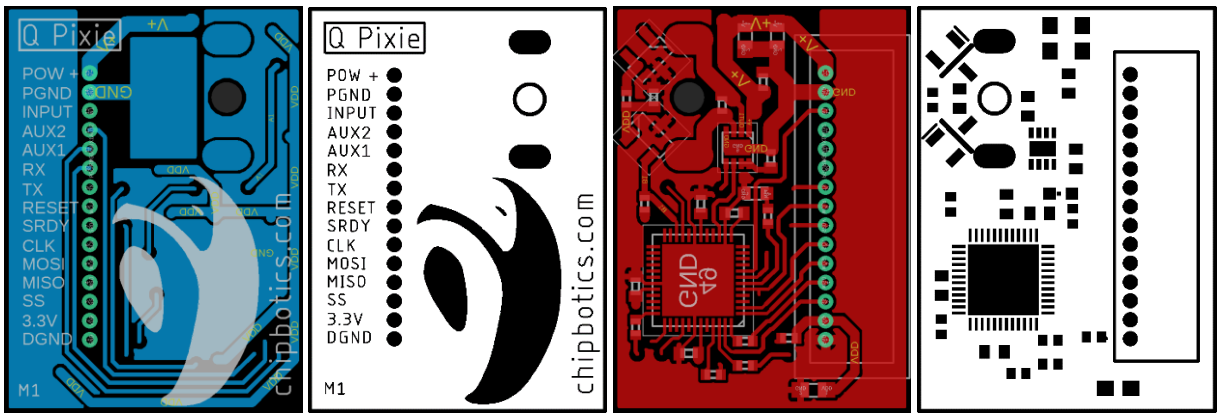
The Chipbotics Q-Pixie is a tiny PCB that solders onto the back of the popular N20 gearmotor (with extended rear shaft) and provides closed loop control and position, velocity and metadata feedback. The device communicates as SPI slave and data can be either polled or streamed. A home/index/edge input is also provided.

Features

- Based around the QuadQuad2Drv microprocessor and drivers
- Includes magnet wheel to press onto rear motor shaft
- Motor power up to 11V, 1.8A (peak)
- Closed loop PID control:
  - Four modes: Off, Power, Position PID, Velocity PID
  - Position and velocity ramp commands
  - Power, velocity and acceleration limiting
  - Loop rate of 500Hz
- Feedback data provided:
  - Position (8, 16 or 32-bit)
  - Velocity
  - Metadata: status, stream timing
- Rated up to 50,000 quadrature transitions/s with velocity resolution of 4 transitions/s
- Home/index/edge input for each encoder
- Operating voltage 3.3V with 5V tolerant SPI inputs
- Molex Picoblade 15-way connector
- Dimensions: 27mm x 20mm
- Bootloader for firmware updates via UART serial
- Arduino library and demo code downloadable
- PIC library available on request

Contents

Physical Connections



The Q-Pixie PCB features a single Molex Picoblade 15-way connector:

Digital Communications Block

Name	Interface	Function
DGND (VSS)	Digital Power	Digital Ground
3.3V (VDD)	Digital Power	Digital Power 3.3V
SS	Input (5V tolerant)	SPI Slave Select Input (Active low, has pull-up)
MISO	Output	SPI Data Out (Connect to Data In / MISO on master)
MOSI	Input (5V tolerant)	SPI Data In (Connect to Data Out / MOSI on master)
CLK	Input (5V tolerant)	SPI Serial Clock Input
SRDY	Output	SPI Data Ready Output (See section *SPI Interface*)
RESET	Input	Reset Pin Input (Active low, has pull-up)
TX	Output	UART Receive and Transmit at 115,200bps. (Used by bootloader)
RX	Input	
AUX1	Output	Used to access bootloader (Short AUX1 and AUX2 and reset/power up to start bootloader)
AUX2	Input (5V tolerant)	
INPUT	Input	Home/index/edge Inputs (Optional, polarity is software configurable)
PGND	Motor Power	Motor Power Ground (Must be floating or connected to DGND)
POW +	Motor Power	Motor Power up to 11V

Electrical Characteristics

- Digital operating voltage 3.3V. Absolute maximum 3.0V – 4.0V.
- SPI inputs and AUX2 are 5V tolerant (MOSI, SS, SCK, AUX2). Absolute maximum 3.0V – 5.5V.
- Motor power voltage maximum 11V.
- Motor power maximum 1.8A.
- Digital input pins are Schmitt triggers with low/high thresholds of 0.2 *VDD* and 0.8 *VDD*.
- Digital output pin low/high voltages are max 0.42V and min 2.4V.

Bootloader & Firmware Updates

The Q-Pixie firmware can be updated via UART serial using the bootloader. A common USB-to-serial cable can be used. Connect the serial cable as follows:

Cable Wire	Board Pin
DGND	DGND
TX	RX
RX	TX

Take care to use only a serial cable with 3.3V interface and power the board only with 3.3V. A serial cable with 3.3V power output is convenient as they can also be used to power the board (such as the FTDI TTL-232RG-VSW3V3-WE or TTL-232RG-VREG3V3-WE) by connecting the cable 3.3V VCC to the 3.3V board pin. The common FTDI TTL-232R-3V3 can be used, but since this cable has a 5V output, the board must be powered some other way.

The bootloader can be accessed by shorting AUX1 and AUX2 and resetting or powering up the board. Note that the RX line must be high before reset / power up. This is normally done by a USB-serial interface, but requires the USB to be plugged into a computer before reset / power up.

Use a serial terminal that supports line delays such as the popular Tera Term. For Tera Term, go to *Setup->Serial port* and set the transmit delay to 50 msec/line. Set the baud rate to 115,200 bps.

When the terminal is set up, the serial cable is connected, the AUX pins are shorted and the board is powered up or reset, the bootloader will print a header line specifically including the word “bootloader”. For Tera Term, the firmware file can be dragged and dropped on the terminal window and the bootloader will confirm each line. If the upload is successful, AUX1/AUX2 can be un-shorted and the board reset or power cycled or type “reset” in the terminal window.

Protocol

SPI Interface

The SPI master interface must be configured for SPI mode 1. We recommend clocking SPI up to 500kHz.

The Q-Pixie SPI interface includes a handshake line from the slave to master, called *Serial Ready* (SRDY), which is set high by the slave to indicate that a new data packet is available.

The following sequence should be followed for a typical packet query:

1. Master waits until slave asserts *Serial Ready* (SRDY) low.
2. Master asserts *Slave Select* (SS) low.
3. Master reads out one entire packet.
4. Master may leave *Slave Select* (SS) low.

Stream packets are sent at a steady rate without requiring a query packet. When a stream packet becomes available, the slave asserts SRDY and the stream packet can be read.

Packet Format

Applicable interface protocol version: v1.

Size	U8	U8	U8	Variable	U8
Field	STX	Packet Size	Packet ID	Payload	Checksum

STX is ASCII character 2.
Packet Size is the size of the entire packet, in bytes.
Checksum is the 8-bit checksum of the entire packet.

Q-Pixie Specific Notes

The Q-Pixie protocol is identical to the QuadQuad2Drv protocol. The single motor is attached to the first channel. The other channels are not physically connected.

Packets

The following section contains details of the packets that can be communicated, specifically, the *Payload* field. The documented fields below are read left to right first, then row by row.

<table>
<tr><th>Packet ID</th><th colspan="2">Description / Payload</th></tr>
<tr><td rowspan="3">1 - Get Version</td><td colspan="2">Read firmware and protocol version numbers.</td></tr>
<tr><td>Send</td><td>None</td></tr>
<tr><td>Reply</td><td>
<table>
<tr><td>Size</td><td>U8</td><td>U8</td></tr>
<tr><td>Field</td><td>Return Code</td><td>Firmware Version Major</td></tr>
<tr><td>Size</td><td>U8</td><td>U8</td></tr>
<tr><td>Field</td><td>Firmware Version Minor</td><td>Protocol Version</td></tr>
</table>
</td></tr>
</table>

2 - Get Binary Motion Data

Read motion data. Each quadrature channel and field in the reply payload, except for *Return Code*, is optional and configurable using the *6 - Set Data Mask* and *8 - Set Stream Config* packets. Channels and fields that are deselected will simply be omitted (zero bytes). Alternatively, this data can be streamed using packet *4 - Set Stream Period*.

Send

None

Reply

Size	U8	Variable	Variable
Field	Return Code	[Channel 1 Data]	[Channel 2 Data]
Size	Variable	Variable	
Field	[Channel 3 Data]	[Channel 4 Data]	

Channel Data:

Size	I8/I16/I32	I32	U8
Field	[Position]	[Velocity]	[Status]

Position: Number of quadrature transitions forward/backward. There are four transitions per detent. The number of bits used to represent *Position* can be configured using packet *6 - Set Data Mask* and defaults to 32-bits. When less than 32-bit position is specified, the lower 8- or 16 bits will simply be retrieved and wrapping will occur on overflow. When relative position mode is enabled using packet *6 - Set Data Mask*, *Position* will contain the change in position since the last position read.

Velocity: Rate at which *Position* is changing, measured as:
Transitions Per Second *= Velocity*.
There are four transitions per detent. Velocity is unaffected by home/index inputs.

Status:

Bit	7	6	5..4
Field	Glitch	Overspeed	Reserved
Bit	3	2	1..0
Field	Input Active	Input Active Accumulator	Input Trigger Accumulator

Glitch: This bit is set if the two quadrature lines A and B have made an invalid transition. This may be caused by noise on the lines or if quadrature velocity exceeds the rated maximum velocity. *Position* and *Velocity* data may be inaccurate if this bit is set.

Overspeed: Quadrature velocity has exceeded rated maximum velocity and *Position* and *Velocity* data may be inaccurate.

Input Active: The input is currently active (evaluated after polarity setting applied).

Input Active Accumulator: The input has been active at least once since the last time status was transmitted (evaluated after polarity setting applied).

Input Trigger Accumulator: If non-zero, the input has been triggered at least once since the last time status was transmitted. The Position counter has been set as configured using packet *14 – Set Input Mode*, or the index/home/edge position has been recorded. This differs from the *Input Active* flags in that a trigger requires additional conditions. For example, an index trigger also requires the quadrature A and B lines both to be 0 and an edge trigger is only set once when the input transitions from inactive to active.

The two bits indicate which edge has been triggered in HOME and EDGE input modes. When input mode is set to INDEX, the Positive Trigger will always be used. Also see packet *14 – Set Input Mode*.

Bit	1	0
Field	Negative Edge	Positive Edge

<table>
<tr><td>3 - Binary Stream Data</td><td colspan="2">Stream packet with motion data. Each quadrature channel and field in the reply payload, except for Return Code, is optional and configurable using the 6 - Set Data Mask and 8 - Set Stream Config packets. Fields that are deselected will simply be omitted (zero bytes). Use packet 4 - Set Stream Period to set up streaming. Also see section “SPI Interface” for how to use the Serial Ready (SRDY) handshake line.</td></tr>
<tr><td></td><td>Send</td><td>This packet must not be sent by the master device.</td></tr>
<tr><td></td><td>Reply</td><td>
<table>
<tr><td>Size</td><td>U16</td><td>U8</td><td>Variable</td></tr>
<tr><td>Field</td><td>[Stream Period Timing]</td><td>[Stream Periods Elapsed]</td><td>[Channel 1 Data]</td></tr>
<tr><td>Size</td><td>Variable</td><td>Variable</td><td>Variable</td></tr>
<tr><td>Field</td><td>[Channel 2 Data]</td><td>[Channel 3 Data]</td><td>[Channel 4 Data]</td></tr>
</table>

Stream Period Timing: Time elapsed since start of calculation of previous stream packet until the current stream packet became available for transmission, measured as: Stream Period Timing in Microseconds = Stream Period Timing x 40.96.
In short, this number measures how old the stream data is and is typically the configured Stream Period plus a small processing overhead. Example: If you set Stream Period = 100 and measure Stream Period Timing = 108, the processing overhead since capturing the quadrature data for the current packet and making it available for transmission is 8 x 40.96ms. If the master fails to read an entire stream packet before the next period, a new stream packet will not be provided until the master reads out the packet. Stream Period Timing excludes time for any missed stream periods. Stream Period Timing will clip at 0xFFFF.

Stream Periods Elapsed: Number of Stream Periods elapsed since the last successful stream packet transmission. This will normally be 1, unless Stream Period is set too small such that the master cannot read the packet in time. Stream Periods Elapsed will clip at 0xFF.

Channel Data: See packet 2 - Get Binary Motion Data.</td></tr>
<tr><td>4 - Set Stream Period</td><td colspan="2">Set period at which motion data is streamed to the master. Alternatively, motion data can be polled using packet 2 - Get Binary Motion Data.</td></tr>
<tr><td></td><td>Send</td><td>
<table>
<tr><td>Size</td><td>U16</td></tr>
<tr><td>Field</td><td>Stream Period</td></tr>
</table>

Stream Period: Time between stream packets, measured as:
Stream Period in Microseconds = Stream Period x 40.96.
Set Stream Period = 0 to disable the stream. Set Stream Period = 1 to receive stream packets at maximum rate. In reality, stream packets cannot be sent at very high rates such as period = 1. The actual rate is affected by the amount of processing required, such as number of fields in the stream, SPI clock speed and the master response time.
Also see Stream Period Timing in packet 3 - Binary Stream Data.
Default at start-up: 0 (Disabled)</td></tr>
<tr><td></td><td>Reply</td><td>
<table>
<tr><td>Size</td><td>U8</td></tr>
<tr><td>Field</td><td>Return Code</td></tr>
</table>
</td></tr>
<tr><td>5 - Get Stream Period</td><td colspan="2">Read period at which motion data is streamed to the master.</td></tr>
<tr><td></td><td>Send</td><td>None</td></tr>
<tr><td></td><td>Reply</td><td>
<table>
<tr><td>Size</td><td>U8</td><td>U16</td></tr>
<tr><td>Field</td><td>Return Code</td><td>Stream Period</td></tr>
</table>

Stream Period: See packet 4 - Set Stream Period.</td></tr>
</table>

6 - Set Data Mask

Set which channels and fields are to be included in motion data (see packet *2 - Get Binary Motion Data* and *3 - Binary Stream Data*).

Send

Size	U8	U8	U8	U8
Field	Chan Mask 1	Data Mask 1	[Chan Mask…]	[Data Mask…]

Any number of *Channel Mask* and *Data Mask* pairs can be concatenated to set multiple masks to multiple channels. If any mask conflicts occur, any latter bit value will override the former.

Channel Mask: Bit mask selecting which channels *Data Mask* will be applied to. Multiple channels can be selected by adding masks.

Bit	7..4	3	2	1	0
Field	Unused	Channel 4	Channel 3	Channel 2	Channel 1

Data Mask: Bit mask selecting what data to include for channels selected by *Channel Mask*.

Bit	7..6	5	4	3	2	1..0
Field	Unused	Status	Reserved	Velocity	Position Relative	Position Size

Position Size: Select how many bits will be used to represent position value.

Position Size	Number of Bits
0x00	Position omitted
0x01	8-bit signed integer
0x02	16-bit signed integer
0x03	32-bit signed integer

Default at start-up: 0x03 (32-bit signed integer)

Position Relative: Set this bit to enable relative position mode. This mode will cause position values to represent the change in position since the last read position value (via either packet *2 - Get Binary Motion Data* or *3 - Binary Stream Data*) instead of absolute position value. Absolute position value can still be read using packet *11 - Get Position*.
Default at start-up: 0 (Disabled)

Velocity: Set this bit to include velocity data.
Default at start-up: 1 (Enabled)

Status: Set this bit to include status data. See packet *2 - Get Binary Motion Data*.
Default at start-up: 0 (Disabled)

Reply

Size	U8
Field	Return Code

7 - Get Data Mask

Read which channels and fields are to be included in motion data (packets *2 - Get Binary Motion Data* and *3 - Binary Stream Data*).

Send

None

Reply

Size	U8	U8	U8
Field	Return Code	Chan 1 Data Mask	Chan 2 Data Mask
Size	U8	U8	
Field	Chan 3 Data Mask	Chan 4 Data Mask	

The *Data Mask* for each channel is returned.

<table>
<tr><td></td><td></td><td>Data Mask: See packet 6 - Set Data Mask.</td></tr>
<tr><td>8 - Set Stream Config</td><td colspan="2">Set which additional fields to include in stream data.</td></tr>
<tr><td></td><td>Send</td><td>
<table>
<tr><td>Size</td><td>U8</td></tr>
<tr><td>Field</td><td>Stream Config Mask</td></tr>
</table>

Stream Config Mask:
<table>
<tr><td>Bit</td><td>7..2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td>Reserved</td><td>Stream Periods Elapsed</td><td>Stream Period Timing</td></tr>
</table>

Stream Period Timing: See packets 3 - Binary Stream Data and 4 - Set Stream Period.
Default at start-up: 0 (Disabled)

Stream Periods Elapsed: See packets 3 - Binary Stream Data and 4 - Set Stream Period.
Default at start-up: 0 (Disabled)</td></tr>
<tr><td></td><td>Reply</td><td>
<table>
<tr><td>Size</td><td>U8</td></tr>
<tr><td>Field</td><td>Return Code</td></tr>
</table>
</td></tr>
<tr><td>9 - Get Stream Config</td><td colspan="2">Read which additional fields to include in stream data.</td></tr>
<tr><td></td><td>Send</td><td>None</td></tr>
<tr><td></td><td>Reply</td><td>
<table>
<tr><td>Size</td><td>U8</td><td>U8</td></tr>
<tr><td>Field</td><td>Return Code</td><td>Stream Config Mask</td></tr>
</table>

Stream Config Mask: See packet 8 - Set Stream Config.</td></tr>
<tr><td>10 - Set Position</td><td colspan="2">Set absolute position value.</td></tr>
<tr><td></td><td>Send</td><td>
<table>
<tr><td>Size</td><td>U8</td><td>I8/I16/I32</td></tr>
<tr><td>Field</td><td>Channel & Size Mask 1</td><td>Position 1</td></tr>
<tr><td>Size</td><td>U8</td><td>I8/I16/I32</td></tr>
<tr><td>Field</td><td>[Channel & Size Mask…]</td><td>[Position…]</td></tr>
</table>

Any number of Channel & Size Mask and Position pairs can be concatenated to set different position values for multiple channels.

Channel & Size Mask: Select which channels to write Position value to and the format of the Position value. Multiple channels can be selected simultaneously.
<table>
<tr><td>Bit</td><td>7..6</td><td>5..4</td><td>3..0</td></tr>
<tr><td>Field</td><td>Unused</td><td>Position Size</td><td>Channel Mask</td></tr>
</table>

Channel Mask: Bit mask selecting which channels Position will be applied to. Multiple channels can be selected by adding masks.
<table>
<tr><td>Bit</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td>Channel 4</td><td>Channel 3</td><td>Channel 2</td><td>Channel 1</td></tr>
</table>

Position Size: Select the size of the Position value following. The whole 32-bit position value will be overwritten even if an 8-bit or 16-bit value is specified.
<table>
<tr><td>Position Size</td><td>Number of Bits</td></tr>
<tr><td>0x00</td><td>Position omitted and assumed zero</td></tr>
<tr><td>0x01</td><td>8-bit signed integer</td></tr>
<tr><td>0x02</td><td>16-bit signed integer</td></tr>
<tr><td>0x03</td><td>32-bit signed integer</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td></td><td>Position: The Position value to be written. The size must match the size specified in Position Size. If Position Size = 0x00, Position must be omitted and is assumed Position = 0.</td></tr>
<tr><td></td><td>Reply</td><td><table><tr><td>Size</td><td>U8</td></tr><tr><td>Field</td><td>Return Code</td></tr></table></td></tr>
<tr><td>11 - Get Position</td><td colspan="2">Read absolute position value.</td></tr>
<tr><td></td><td>Send</td><td>Three different parameter formats are possible and will determine the contents and format of the reply. The format is identified by the size of the payload.

Size 0: An empty payload will request the position values in the sizes previously configured using Position Size in packet 6 - Set Data Mask, or omitted if the size was set to 0.<table><tr><td>Size</td><td>Zero</td></tr><tr><td>Field</td><td>None</td></tr></table>
Size 1: A single byte will specify a channel mask. The position values of the channels selected in Channel Mask will be returned in the reply.<table><tr><td>Size</td><td>U8</td></tr><tr><td>Field</td><td>Channel Mask</td></tr></table>
Size 4: Four bytes will specify the size/format of the position value, for each channel, to return in the reply.<table><tr><td>Size</td><td>U8</td><td>U8</td></tr><tr><td>Field</td><td>Channel 1 Position Size</td><td>Channel 2 Position Size</td></tr><tr><td>Size</td><td>U8</td><td>U8</td></tr><tr><td>Field</td><td>Channel 3 Position Size</td><td>Channel 4 Position Size</td></tr></table>
Channel Mask:<table><tr><td>Bit</td><td>7..4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>Field</td><td>Unused</td><td>Channel 4</td><td>Channel 3</td><td>Channel 2</td><td>Channel 1</td></tr></table>
Channel Position Size:<table><tr><td>Position Size</td><td>Number of Bits</td></tr><tr><td>0x00</td><td>Position omitted</td></tr><tr><td>0x01</td><td>8-bit signed integer</td></tr><tr><td>0x02</td><td>16-bit signed integer</td></tr><tr><td>0x03</td><td>32-bit signed integer</td></tr></table></td></tr>
<tr><td></td><td>Reply</td><td><table><tr><td>Size</td><td>U8</td><td>I8/I16/I32</td><td>I8/I16/I32</td><td>I8/I16/I32</td><td>I8/I16/I32</td></tr><tr><td>Field</td><td>Return Code</td><td>[Position 1]</td><td>[Position 2]</td><td>[Position 3]</td><td>[Position 4]</td></tr></table>
The presence and size/format of each Position value is determined by the specified parameters.</td></tr>
<tr><td>12 - Set History Dimensions</td><td colspan="2">Set maximum history length and averaging time for velocity calculation purposes.</td></tr>
<tr><td></td><td>Send</td><td><table><tr><td>Size</td><td>U8</td><td>U8</td></tr><tr><td>Field</td><td>History Length</td><td>Maximum Averaging Time in Bits</td></tr></table>
History Length: The maximum number of quadrature capture events that will be averaged. A capture event occurs on every fourth quadrature transition, i.e. once per detent. Larger values will produce more accurate velocity averages, but will be slower to respond to changes in velocity.
Range: 2 - 127</td></tr>
</table>

Default at start-up: 31

Maximum Averaging Time in Bits: The maximum time over which quadrature capture events are averaged, specified in bits, where:
Maximum Averaging Time = $2^{Maximum\ Averaging\ Time\ in\ Bits}$ and
Maximum Averaging Time in Microseconds = *Maximum Averaging Time* x 0.64

Maximum Averaging Time in Bits	Maximum Averaging Time	Maximum Averaging Time in Milliseconds
14	16384	10.5
15	32768	21.0
16	65536	41.9
17	131072	83.9
18	262144	167.8
19	524288	335.5
20	1048576	671.1
21	2097152	1,342.2
22	4194304	2,684.4
23	8388608	5,368.7
24	16777216	10,737.4
25	33554432	21,474.8
26	67108864	42,949.7
27	134217728	85,899.3
28	268435456	171,798.7
29	536870912	343,597.4
30	1073741824	687,194.8
31	2147483648	1,374,389.5
32	4294967296	2,748,779.1

At lower velocities, the total duration of *History Length* quadrature capture events increases and thus the lower the minimum detectable velocity will be, but the slower average velocity will wind down when motion is significantly slowed down or stopped, since there will be physically less transitions to detect. Larger values will allow lower velocities to be detected.
Range: 14 - 32
Default at start-up: 20

At very low velocities, when *History Length* quadrature capture events exceeds *Maximum Averaging Time*, less than *History Length* capture events will be used for averaging, but allowing for faster responses and longer total averaging times instead.

Reply

Size	U8
Field	Return Code

13 - Get History Dimensions

Read maximum history length and averaging time for velocity calculation purposes.

Send

None

Reply

Size	U8	U8	U8
Field	Return Code	History Length	Maximum Averaging Time in Bits

14 – Set Input Mode

Set the function of quadrature input pins as home / index / edge / disabled. The status of the input can be read in the <Status> field using packet *2 - Get Binary Motion Data* or packet *3 - Binary Stream* Data.

Send

Size	U8	U8	U16	I32
Field	Channel Mask 1	Input Configuration 1	Spacing 1	Position 1
Size	U8	U8	U16	I32
Field	[Channel Mask…]	[Input Configuration…]	[Spacing…]	[Position…]

Any number of *Channel Mask*, *Input Configuration* and *Position* pairs can be concatenated to set different input modes for multiple channels.

Channel Mask: Bit mask selecting which channels I*nput Mode* and *Position* will be applied to. Multiple channels can be selected by adding masks.

Bit	7..4	3	2	1	0
Field	Unused	Channel 4	Channel 3	Channel 2	Channel 1

Input Configuration:

Bit	7..6	5..4	3	2	1..0
Field	Unused	Edge	Input Polarity	Reserved	Input Mode

Edge: Select how to behave when an input trigger occurs. Either the position counter is set to *<Position>* specified in this packet, or the current position counter is just recorded. The module records the extreme positions reached before an input is activated. In HOME and EDGE modes, the position will only be set on the specified positive/negative end. If <Spacing> is set to 0, a single end-stop is assumed and <Edge> will specify on which end the end-stop is. The direction is irrelevant in INDEX mode.

Input Polarity	Meaning
0x00	Record position only. If <spacing>=0, this is a positive end-stop.
0x01	Record position only. If <spacing>=0, this is a negative end-stop.
0x02	Set position counter on positive end trigger
0x03	Set position counter on negative end trigger

Input Polarity: Select whether the input pin will be active when low or high.

Input Polarity	Meaning
0x00	Input is active when low
0x01	Input is active when high

Input Mode:

Input Mode	Meaning	Description
0x00	Disabled	Signal on input has no effect.
0x01	Home mode	While input is active (see *Input Polarity*), *Position* will be written into relevant channel position value if so configured. This is typically used on linear actuators where a button or infrared sensor acts as input to indicate that the end stop has been reached.
0x02	Index mode	While input is active (see *Input Polarity*) and quadrature input A = B = 0, *Position* will be written into relevant channel position value if so configured. This is typically used with an index signal output from a quadrature encoder.
0x03	Edge mode	When an input transition from inactive to active occurs, *Position* will be written into relevant channel position value if so configured. This is ideal for end-stop sensors.

Spacing: Specifies a hysteresis threshold when arranged such that the input is triggered on either positive/negative end. This can be when using two separate sensors on each end, where either can set the input, or when dealing with circular motion where a single sensor can be set in either direction. The hysteresis is typically

a small value used to prevent triggering on the wrong end by requiring the positive end to be this distance away from the negative end and vice versa.

Further, this is used to specify whether there is one or two end-stops. A value of 0 indicates one end-stop and any other values indicates two.

Position: The *Position* value to be written. If *Input Mode* is set to *Disabled*, *Position* must be omitted.

Reply

Size	U8
Field	Return Code

16 – Set Control Mode

Set the motor control mode.

Send

Size	U8	U8
Field	Channel Mask 1	Control Mode 1
Size	U8	U8
Field	[Channel Mask…]	[Control Mode…]

Any number of *Channel Mask* and *Control Mode* pairs can be concatenated to set different motor control modes for multiple channels.

Channel Mask: Bit mask selecting which channels *Control Mode* will be applied to. Multiple channels can be selected by adding masks.

Bit	7..4	3	2	1	0
Field	Unused	Channel 4	Channel 3	Channel 2	Channel 1

Control Mode:

Input Mode	Meaning	Description
0x00	Off	Motor output will be set to zero / off.
0x01	Power	Motor output will be set to the power value specified using packet *18 – Set Motor* Power.
0x02	Position PID Control	Motor output will be controlled using closed loop PID to maintain a position target. See position control related commands.
0x03	Velocity PID Control	Motor output will be controlled using closed loop PID to maintain a velocity target. See velocity control related commands.

Reply

Size	U8
Field	Return Code

18 – Set Motor Power

Set the motor output power. Control Mode must be set to *Power* using packet *16 – Set Control Mode*.

Send

Size	U8	U8
Field	Channel Mask 1	Power 1
Size	U8	I16
Field	[Channel Mask…]	[Power…]

Any number of *Channel Mask* and *Power* pairs can be concatenated to set different motor output powers for multiple channels.

Channel Mask: Bit mask selecting which channels *Power* will be applied to. Multiple channels can be selected by adding masks.

Bit	7..4	3	2	1	0
Field	Unused	Channel 4	Channel 3	Channel 2	Channel 1

<table>
<tr><td></td><td></td><td>Power: Power to apply to the motor output. The sign determines the direction.
Range: -8,191 - 8,191
Default at start-up: 0</td></tr>
<tr><td></td><td>Reply</td><td><table><tr><td>Size</td><td>U8</td></tr><tr><td>Field</td><td>Return Code</td></tr></table></td></tr>
<tr><td>20 – Set Position Control Target</td><td colspan="2">Set the motor position control target. Control Mode must be set to Position PID Control using packet 16 – Set Control Mode.</td></tr>
<tr><td></td><td>Send</td><td><table><tr><td>Size</td><td>U8</td><td>I32</td></tr><tr><td>Field</td><td>Channel Mask 1</td><td>Position Target 1</td></tr><tr><td>Size</td><td>U8</td><td>I32</td></tr><tr><td>Field</td><td>[Channel Mask…]</td><td>[Position Target…]</td></tr></table>
Any number of Channel Mask and Control Mode pairs can be concatenated to set different position targets for multiple channels.

Channel Mask: Bit mask selecting which channels Position Target will be applied to. Multiple channels can be selected by adding masks.<table><tr><td>Bit</td><td>7..4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>Field</td><td>Unused</td><td>Channel 4</td><td>Channel 3</td><td>Channel 2</td><td>Channel 1</td></tr></table>
Position Target: Position target the closed loop PID will aim to achieve.</td></tr>
<tr><td></td><td>Reply</td><td><table><tr><td>Size</td><td>U8</td></tr><tr><td>Field</td><td>Return Code</td></tr></table></td></tr>
<tr><td>22 – Set Velocity Control Target</td><td colspan="2">Set the motor velocity control target. Control Mode must be set to Velocity PID Control using packet 16 – Set Control Mode.</td></tr>
<tr><td></td><td>Send</td><td><table><tr><td>Size</td><td>U8</td><td>I32</td></tr><tr><td>Field</td><td>Channel Mask 1</td><td>Velocity Target 1</td></tr><tr><td>Size</td><td>U8</td><td>I32</td></tr><tr><td>Field</td><td>[Channel Mask…]</td><td>[Velocity Target…]</td></tr></table>
Any number of Channel Mask and Control Mode pairs can be concatenated to set different velocity targets for multiple channels.

Channel Mask: Bit mask selecting which channels Velocity Target will be applied to. Multiple channels can be selected by adding masks.<table><tr><td>Bit</td><td>7..4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>Field</td><td>Unused</td><td>Channel 4</td><td>Channel 3</td><td>Channel 2</td><td>Channel 1</td></tr></table>
Velocity Target: Velocity target the closed loop PID will aim to achieve, specified as: Velocity Target = <transitions/s>.
Range: -100,000 shl 12 - 100,000 shl 12</td></tr>
<tr><td></td><td>Reply</td><td><table><tr><td>Size</td><td>U8</td></tr><tr><td>Field</td><td>Return Code</td></tr></table></td></tr>
<tr><td>24 – Set Power Limit</td><td colspan="2">Set the motor power limit. This limit will be applied regardless of control mode.</td></tr>
<tr><td></td><td>Send</td><td><table><tr><td>Size</td><td>U8</td><td>U16</td></tr><tr><td>Field</td><td>Channel Mask 1</td><td>Power Limit 1</td></tr><tr><td>Size</td><td>U8</td><td>U16</td></tr><tr><td>Field</td><td>[Channel Mask…]</td><td>[Power Limit…]</td></tr></table></td></tr>
</table>

<table>
<tr><td></td><td></td><td>Any number of Channel Mask and Power Limit pairs can be concatenated to set different power limits for multiple channels.

Channel Mask: Bit mask selecting which channels Power Limit will be applied to. Multiple channels can be selected by adding masks.
<table>
<tr><td>Bit</td><td>7..4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td>Unused</td><td>Channel 4</td><td>Channel 3</td><td>Channel 2</td><td>Channel 1</td></tr>
</table>
Power Limit: Value that output power will be limited to.
Range: 0 - 8,191</td></tr>
<tr><td></td><td>Reply</td><td>
<table>
<tr><td>Size</td><td>U8</td></tr>
<tr><td>Field</td><td>Return Code</td></tr>
</table>
</td></tr>
<tr><td>26 – Set Velocity Limit</td><td colspan="2">Set the motor velocity limit. This limit will be applied in both Position PID Control and Velocity PID Control modes.

Note: This limit is not guaranteed as it depends on PID configuration and real-world interaction.</td></tr>
<tr><td></td><td>Send</td><td>
<table>
<tr><td>Size</td><td>U8</td><td>U32</td></tr>
<tr><td>Field</td><td>Channel Mask 1</td><td>Velocity Limit 1</td></tr>
<tr><td>Size</td><td>U8</td><td>U32</td></tr>
<tr><td>Field</td><td>[Channel Mask…]</td><td>[Velocity Limit…]</td></tr>
</table>
Any number of Channel Mask and Velocity Limit pairs can be concatenated to set different velocity limits for multiple channels.

Channel Mask: Bit mask selecting which channels Velocity Limit will be applied to. Multiple channels can be selected by adding masks.
<table>
<tr><td>Bit</td><td>7..4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Field</td><td>Unused</td><td>Channel 4</td><td>Channel 3</td><td>Channel 2</td><td>Channel 1</td></tr>
</table>
Velocity Limit: Value that control velocity will be limited to, specified as:
Velocity Limit = <transitions/s> shl 12.
The extra bits can be used to specify a fractional velocity. Such low velocities are not normally achievable in speed control mode and primarily apply to Position PID Control mode.
Range: 0 - 100,000 shl 12</td></tr>
<tr><td></td><td>Reply</td><td>
<table>
<tr><td>Size</td><td>U8</td></tr>
<tr><td>Field</td><td>Return Code</td></tr>
</table>
</td></tr>
<tr><td>28 – Set Acceleration Limit</td><td colspan="2">Set the motor acceleration limit. This limit will be applied in both Position PID Control and Velocity PID Control modes. Additionally, in Position PID Control mode, the motor will decelerate at this rate in advance, as it approaches the position target, to prevent it from overshooting.

Note: This limit is not guaranteed as it depends on PID configuration and real-world interaction.</td></tr>
<tr><td></td><td>Send</td><td>
<table>
<tr><td>Size</td><td>U8</td><td>U32</td></tr>
<tr><td>Field</td><td>Channel Mask 1</td><td>Acceleration Limit 1</td></tr>
<tr><td>Size</td><td>U8</td><td>U32</td></tr>
<tr><td>Field</td><td>[Channel Mask…]</td><td>[Acceleration Limit…]</td></tr>
</table>
</td></tr>
</table>

Any number of *Channel Mask* and *Acceleration Limit* pairs can be concatenated to set different acceleration limits for multiple channels.

Channel Mask: Bit mask selecting which channels *Acceleration Limit* will be applied to. Multiple channels can be selected by adding masks.

Bit	7..4	3	2	1	0
Field	Unused	Channel 4	Channel 3	Channel 2	Channel 1

Acceleration Limit: Value that control acceleration will be limited to, specified as:
Acceleration Limit = <transitions/s/s> shl 12.
The extra bits can be used to specify a fractional acceleration. A value of zero disables acceleration limiting.
Range: 0 - 1000,000 shl 12

Reply

Size	U8
Field	Return Code

30 – Set Position Control PID Constants

Set the PID constants for *Position PID Control* mode.

Send

Size	U8	U16	U16	U16	U16
Field	Channel Mask 1	P1	I1	D1	Windup Limit 1
Size	U8	U16	U16	U16	U16
Field	[Channel Mask…]	[P…]	[I…]	[D…]	[Windup Limit …]

Any number of *Channel Mask* and *PID* pairs can be concatenated to set different PID constants for multiple channels.

Channel Mask: Bit mask selecting which channels *PID* constants will be applied to. Multiple channels can be selected by adding masks.

Bit	7..4	3	2	1	0
Field	Unused	Channel 4	Channel 3	Channel 2	Channel 1

P: Proportional Constant.
Default: 100

I: Integration Constant.
Default: 500

D: Differential Constant.
Default: 500

Windup Limit: Integration windup limit.
Range: 0 - 8,191
Default: 600

Reply

Size	U8
Field	Return Code

32 – Set Velocity Control PID Constants

Set the PID constants for *Velocity PID Control* mode.

Send

Size	U8	U16	U16	U16
Field	Channel Mask 1	P1	I1	D1
Size	U8	U16	U16	U16
Field	[Channel Mask…]	[P…]	[I…]	[D…]

Any number of *Channel Mask* and *PID* pairs can be concatenated to set different PID constants for multiple channels.

		Channel Mask: Bit mask selecting which channels *PID* constants will be applied to. Multiple channels can be selected by adding masks. Bit 7..4 | 3 | 2 | 1 | 0 Field Unused | Channel 4 | Channel 3 | Channel 2 | Channel 1 ***P***: Proportional Constant. Default: 100 ***I***: Integration Constant. Default: 200 ***D***: Differential Constant. Default: 500
	Reply	Size: U8 Field: Return Code
34 – Position Ramp	Ramp to the specified position target at the specified velocity and optional acceleration rates. Control Mode must be set to *Position PID Control* using packet *16 – Set Control Mode*.	
	Send	Size: U8 | I32 | U32 | U32 Field: Channel Mask 1 | Position 1 | Velocity 1 | Acceleration 1 Size: U8 | I32 | U32 | U32 Field: [Channel Mask…] | [Position…] | [Velocity…] | [Acceleration…] Any number of *Channel Mask* and ramp pairs can be concatenated to initiate different ramps on multiple channels. ***Channel Mask***: Bit mask selecting on which channels to initiate position ramps. Multiple channels can be selected by adding masks. Bit 7..4 | 3 | 2 | 1 | 0 Field Unused | Channel 4 | Channel 3 | Channel 2 | Channel 1 ***Position***: Position target. ***Velocity***: Velocity to ramp at, specified as: *Velocity* = <transitions/s> shl 12. The internal position target will be ramped at this rate and may be exceeded or not achieved in reality due to the acceleration limit or real-world interactions (such as re-accelerating after physically stopping the motor momentarily). The velocity limit set using packet *26 – Set Velocity Limit* 28 – Set Acceleration Limit still applies. Range: 0 – 100,000 shl 12 ***Acceleration***: Rate at which to accelerate towards the specified velocity, specified as: *Acceleration* = <transitions/s/s> shl 12. A value of zero disables acceleration limiting. The acceleration limit set using packet *28 – Set Acceleration Limit* still applies. Range: 0 – 1,000,000 shl 12
	Reply	Size: U8 Field: Return Code

36 – Velocity Ramp

Ramp to the specified velocity target at the specified acceleration rate. Control Mode must be set to *Speed PID Control* using packet *16 – Set Control Mode*.

Send

Size	U8	I32	U32
Field	Channel Mask 1	Velocity 1	Acceleration 1
Size	U8	I32	U32
Field	[Channel Mask…]	[Velocity…]	[Acceleration…]

Any number of *Channel Mask* and ramp pairs can be concatenated to initiate different ramps on multiple channels.

Channel Mask: Bit mask selecting on which channels to initiate velocity ramps. Multiple channels can be selected by adding masks.

Bit	7..4	3	2	1	0
Field	Unused	Channel 4	Channel 3	Channel 2	Channel 1

Velocity: Velocity target, specified as:
Velocity = <transitions/s>.
Range: 0 - 100,000

Acceleration: Rate at which to accelerate towards the specified velocity, specified as:
Acceleration = <transitions/s/s> shl 12.
The internal velocity target will be ramped at this rate and may be exceeded or not achieved in reality due to real-world interactions (such as re-accelerating after physically stopping the motor momentarily). Set to 0 to ignore this parameter. The acceleration limit set using packet *28 – Set Acceleration Limit* still applies.
Range: 0 – 1,000,000 shl 12

Reply

Size	U8
Field	Return Code

Worked Examples

Read Version Numbers

Refer to sections *Physical Connections* and *SPI Interface* and packet *1 - Get Version* for more detail.

To query the version numbers from the Q-Pixie device, send the following packet (in decimal bytes):

Meaning	STX	Packet Size	Packet ID	Checksum
Value	2	4	1	249

You should receive a reply similar to this (in decimal bytes):

Meaning	STX	Packet Size	Packet ID	Return Code	Firmware Version Major	Firmware Version Minor	Protocol Version	Checksum
Value	2	8	1	0	0	1	0	244