QuadQuad PCB Datasheet

4ch Quadrature Decoder

Description

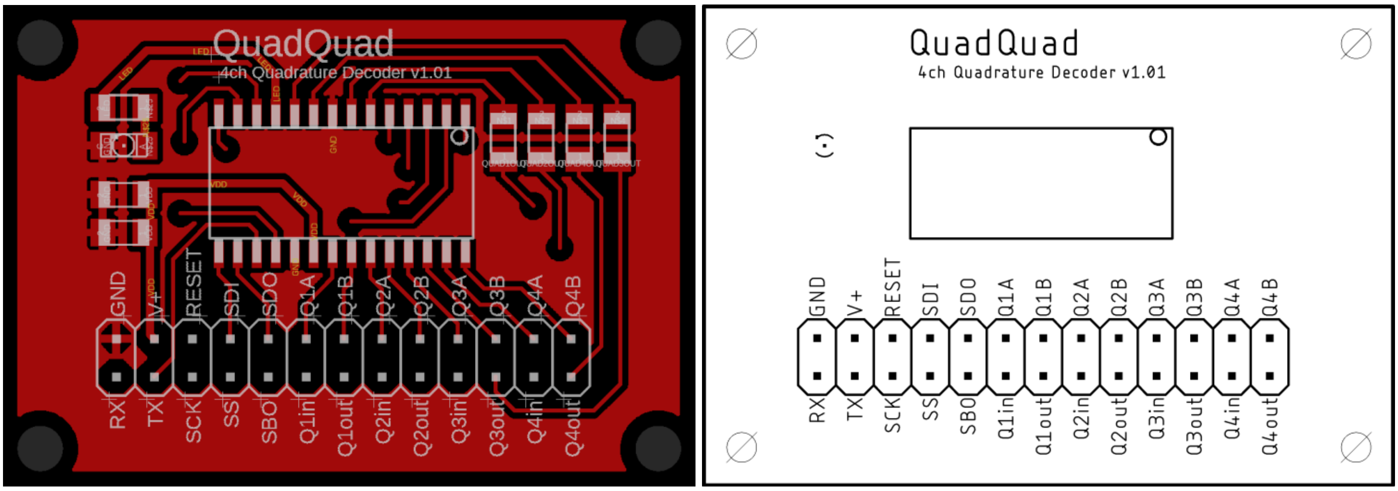
The QuadQuad microprocessor decodes up to four incremental quadrature encoders simultaneously and provides position, velocity and metadata. The device communicates as SPI slave and data can be either polled or streamed. Home/index inputs are provided.

Features

- Four quadrature channels (sampled 4x per indent)
- Data provided:
  - Position (8, 16 or 32-bit)
  - Velocity (12-bit resolution)
  - Metadata: status, stream timing
- Velocity up to 10,000 transitions/s
- Velocity precision of 0.1% at rated velocity
- Home/index input for each channel
- SPI slave interface up to 2MHz
- Streamed data packets up to at least 50Hz with 4x channels at maximum rated velocity
- Operating voltage 3.3V to 5.5V
- Supports 4x Pololu HP motors at their maximum rated shaft speed of 30,000 RPM
- Arduino library and demo available

Contents

Physical Connections



The QuadQuad PCB features one 26-pin connector with the following pins:

Name	Function
Q1A, Q1B…	Quadrature inputs A and B for channels 1-4
Q1in, Q2in…	Home/index inputs for channels 1-4 (Optional. Polarity is software configurable)
Q1out, Q2out…	Outputs for channels 1-4 (Not yet implemented)
GND, V+	Power connections (GND is 0V ground, V+ is 3.3V to 5.5V)
RESET	Reset pin (Active low, can be left unconnected)
SS	SPI Slave Select (Active low)
SCK	SPI Serial Clock
SDI	SPI Serial Data In (Connect to MOSI on master)
SDO	SPI Serial Data Out (Connect to MISO on master)
SBO	SPI Serial Back Off (Handshake line, see section “SPI Interface”)
RX, TX	UART Receive and Transmit (Not yet implemented)

Electrical Characteristics

- Operating power (*V+*) should be 3.3V to 5.5V. Absolute maximum power range is -0.3V to 6.5V.
- Input pins are Schmitt triggers with low/high thresholds of 0.2*V+* and 0.8*V+*.
- All input pins have internal pull-ups.
- Output pin low/high voltages are max 0.6V and min *V+* - 0.7.

Protocol

SPI Interface

The SPI master interface must be configured for SPI mode 1 and can be clocked up to 2Mhz.

The QuadQuad SPI interface includes a handshake line from the slave to master, called *Serial Back Off* (SBO), used to prevent the master from overrunning the slave's buffers. While SBO is asserted low, SCK must not be clocked. The SBO line is also used by the slave to signal when data is available, while SS is de-asserted high.

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

1. Master asserts *Slave Select* (SS) low.
2. Master waits until slave asserts *Serial Back Off* (SBO) low and then master clocks out one byte. Repeat for entire packet.
3. Master de-asserts *Slave Select* (SS) high. Slave de-asserts *Serial Back Off* (SBO).
4. Master waits until slave asserts *Serial Back Off* (SBO) low. This indicates slave has packet data to transmit.
5. Master asserts *Slave Select* (SS) low.
6. Master waits until slave asserts *Serial Back Off* (SBO) low and then master reads one byte. Repeat for entire packet. The master may also transmit data at the same time or after reading a packet from the slave, in which case repeat step 3.
7. Master de-asserts *Slave Select* (SS) high. Slave de-asserts *Serial Back Off* (SBO).

Stream packets are a special case, since they are sent at a steady rate without requiring a query packet. When a stream packet becomes available, the slave asserts SBO and the stream packet can be read by proceeding from step 6.

Packet Format

Applicable interface protocol version: v1.00.

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.

Packets

The following section contains details of the packets that can be communicated, specifically, the *Payload* field.

<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	I16	U8
Field	[Position]	[Velocity]	[Status]

Position: Number of quadrature transitions forward/backward. There are four transitions per indent. 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* * 15,624 / 4,096.
There are four transitions per indent. Velocity is unaffected by home/index inputs.

Status:

Bit	7	6..1	0
Field	Glitch	Reserved	Overspeed

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.

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

3 - Binary Stream Data

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 Back Off* (SBO) handshake line.

Send: This packet must not be sent by the master device.

Reply:

Size	U16	U8	Variable
Field	[Stream Period Timing]	[Stream Periods Elapsed]	[Channel 1 Data]
Size	Variable	Variable	Variable
Field	[Channel 2 Data]	[Channel 3 Data]	[Channel 4 Data]

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 Seconds = *Stream Period Timing* x 4/15625.
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 4/15625. 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*.

4 - Set Stream Period

Set period at which motion data is streamed to the master. Alternatively, this data can be polled using packet *2 - Get Binary Motion* Data.

Send

Size	U16
Field	Stream Period

Stream Period: Time between stream packets, measured as:
Stream Period in Seconds = *Stream Period* x 4/15625.
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)

Reply

Size	U8
Field	Return Code

5 - Get Stream Period

Read period at which motion data is streamed to the master.

Send

None

Reply

Size	U8	U16
Field	Return Code	Stream Period

Stream Period: See packet *4 - Set Stream Period*.

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.

Data Mask: See packet *6 - Set Data Mask*.

8 - Set Stream Config

Set which additional fields to include in stream data.

Send

Size	U8
Field	Stream Config Mask

Stream Config Mask:

Bit	7..2	1	0
Field	Reserved	Stream Periods Elapsed	Stream Period Timing

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)

<table>
<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 rowspan="3">9 - Get Stream Config</td><td colspan="2">Read which additional fields to include in stream data.</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>Stream Config Mask</td></tr></table>
<p><b>Stream Config Mask</b>: See packet <i>8 - Set Stream Config</i>.</p></td></tr>
<tr><td rowspan="3">10 - Set Position</td><td colspan="2">Set absolute position value.</td></tr>
<tr><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>
<p>Any number of <i>Channel & Size Mask</i> and <i>Position</i> pairs can be concatenated to set different position values for multiple channels.</p>
<p><b><i>Channel & Size Mask</i></b>: Select which channels to write <i>Position</i> value to and the format of the <i>Position</i> value. Multiple channels can be selected simultaneously.</p>
<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>
<p><b><i>Channel Mask</i></b>: Bit mask selecting which channels <i>Position</i> will be applied to. Multiple channels can be selected by adding masks.</p>
<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>
<p><b><i>Position Size</i></b>: Select the size of the <i>Position</i> value following. The whole 32-bit position value will be overwritten even if an 8-bit or 16-bit value is specified.</p>
<table><tr><td>Position Size</td><td>Number of Bits</td></tr><tr><td>0x00</td><td><i>Position</i> 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>
<p><b><i>Position</i></b>: The <i>Position</i> value to be written. The size must match the size specified in <i>Position Size</i>. If <i>Position Size</i> = 0x00, <i>Position</i> must be omitted and is assumed <i>Position</i> = 0.</p></td></tr>
<tr><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 rowspan="2">11 - Get Position</td><td colspan="2">Read absolute position value.</td></tr>
<tr><td>Send</td><td><p>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.</p>
<p><b><i>Size 0</i></b>: An empty payload will request the position values in the sizes previously configured using <i>Position Size</i> in packet <i>6 - Set Data Mask</i>, or omitted if the size was set to 0.</p>
<table><tr><td>Size</td><td>Zero</td></tr><tr><td>Field</td><td>None</td></tr></table></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.

Size	U8
Field	Channel Mask

Size 4: Four bytes will specify the size/format of the position value, for each channel, to return in the reply.

Size	U8	U8
Field	Channel 1 Position Size	Channel 2 Position Size
Size	U8	U8
Field	Channel 3 Position Size	Channel 4 Position Size

Channel Mask:

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

Channel Position Size:

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

Reply

Size	U8	I8/I16/I32	I8/I16/I32	I8/I16/I32	I8/I16/I32
Field	Return Code	[Position 1]	[Position 2]	[Position 3]	[Position 4]

The presence and size/format of each *Position* value is determined by the specified parameters.

12 - Set History Dimensions

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

Send

Size	U8	U8
Field	History Length	Maximum Averaging Time in Bits

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 indent. Larger values will produce more accurate velocity averages, but will be slower to respond to changes in velocity.
Range: 2-31
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 Seconds = *Maximum Averaging Time* / 62,500

Maximum Averaging Time in Bits	Maximum Averaging Time	Maximum Averaging Time in Milliseconds
10	1024	16.4
11	2048	32.8
12	4096	65.5
13	8192	131.1
14	16384	262.1
15	32768	524.3
16	65536	1,048.6

At lower velocities, the total duration of *History Length* quadrature capture events increases and thus the lower the minimum detectable speed 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: 10-16
Default at start-up: 16

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 / disabled.

Send

Size	U8	U8	I8/I16/I32
Field	Channel & Size Mask 1	Input Configuration 1	Position 1
Size	U8	U8	I8/I16/I32
Field	Channel & Size Mask…	Input Configuration…	Position…

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

Channel & Size Mask: Select for which channels to configure input pints and the format of the *Position* value. Multiple channels can be selected simultaneously.

Bit	7..6	5..4	3..0
Field	Unused	Position Size	Channel Mask

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

Bit	3	2	1	0
Field	Channel 4	Channel 3	Channel 2	Channel 1

Position Size: Select the size of the *Position* value following. Selecting Zero will result in a *Position* of zero being assumed without having to provide a *Position* value.

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

Input Configuration:

Bit	7..4	3	2	1..0
Field	Unused	Input Polarity	Unused	Input Mode

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. 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. This is typically used with an index signal output from a quadrature encoder.

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 it is assumed *Position* = 0. Similarly, if *Input Mode* is set to *Disabled*, *Position* must be omitted.

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 QuadQuad device, send the following packet (in decimal bytes):

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

You should receive the following reply (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