HUB-ee 6020 BMDS

Basic Motor Driver with Sensors User Guide

Document Version 1.0

Contents

DISCLAIMER OF LIABILITY

While we always try to provide accurate information, there may be times when factual, technical or typographical inaccuracies appear in our documentation. We apologise if this happens and we reserve the right to make changes and corrections at any time, without notice. Creative Robotics expressly disclaims liability for errors or omissions in the content of this document and makes no commitment to update the information contained in these documents.

Creative Robotics expressly disclaims all liability for the interpretation and use by others of any information contained in this document. All products are subject to change without prior notice. Creative Robotics disclaims any responsibility for errors, omissions or inaccuracies in product documentation or any other data relating to our products. Creative Robotics disclaims any responsibility for loss, damage or harm caused directly or indirectly by the use of our products.

(Basically: be careful, have fun but if you mess up then don't blame us)

Copyright © Creative Robotics Ltd 2013

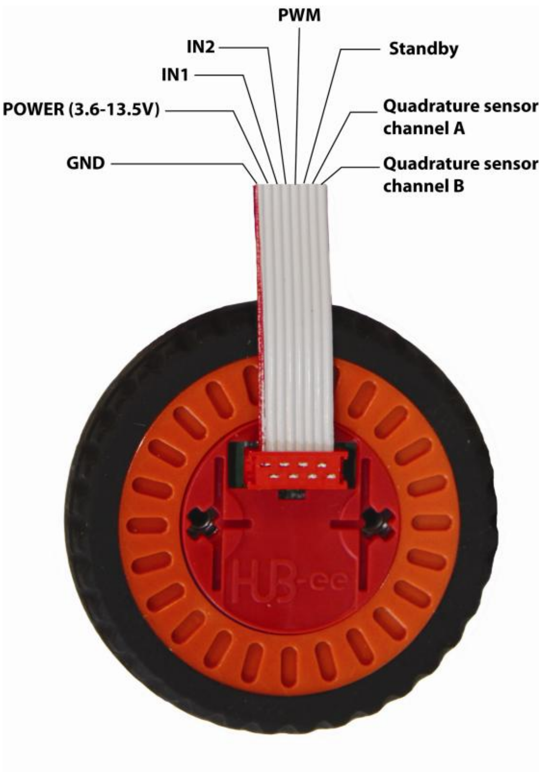
Electronics:

The BMD-S PCB uses a Toshiba TB6593FNG motor driver IC with and on-board 3.3v regulator, logic input buffers providing 5V tolerant inputs, and a pair of reflective optical sensors that provide a quadrature sensor output.

Motor Driver:	Sensors:	DC Motor:
Operating voltage range: 3.6 – 13.5V **Max continuous motor current:** 1.2A **Peak Motor Current:** 3.2A **Control input range:** 2.6 – 5.5V.	**Sensor resolution:** 128 pulse /revolution (32 stripe reflective sensor) **Sensor outputs:** Quadrature phase A and B (3.3v logic output via 1k resistors)	**Motor:** 0.9A Stall current @ 7V.

Connector Pins:

The table on the right illustrates what signals you need to send in order to turn the wheel – as a basic rule, send a PWM signal to the PWM pin (kinda obvious) then use the IN1 and IN2 pins to control direction – pull one high and the other low to move in one direction, reverse them for the other. The brake mode (when both IN1 and IN2 are high) will short circuit the motor coils, forcing the wheel to stop quickly and resist movement. The Stop –No Brake mode disconnects the motor and allows it to freewheel a little.



IN1	IN2	PWM	Standby*	Result:
H	H	H/L	H	**Stop - Brake**
L	H	H	H	**Turn Forwards**
L	H	L	H	**Stop - Brake**
H	L	H	H	**Turn Backwards**
H	L	L	H	**Stop - Brake**
L	L	H/L	H	**Stop – No Brake**
H/L	H/L	H/L	L	**Standby**

**Standby is active LOW and has a pull up resistor - All other inputs have pull down resistors. Standby can be left unconnected for normal operation.*

Copyright © Creative Robotics Ltd 2013

Cables:

The PCB uses an 8 way MicroMaTch connector socket manufactured by TE Electronics. Suitable cables can be bought pre made or made by crimping a plug onto ribbon cable using a vice. These connectors have a polarising tab so they cannot be plugged in the wrong way round.

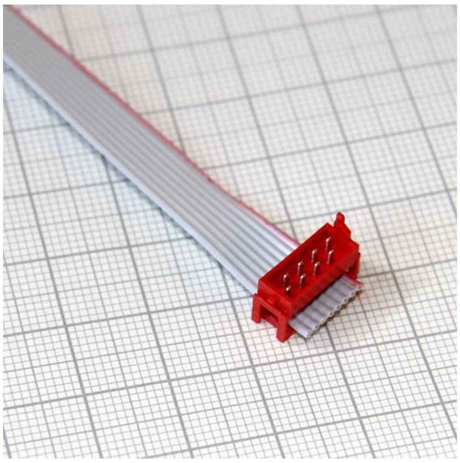
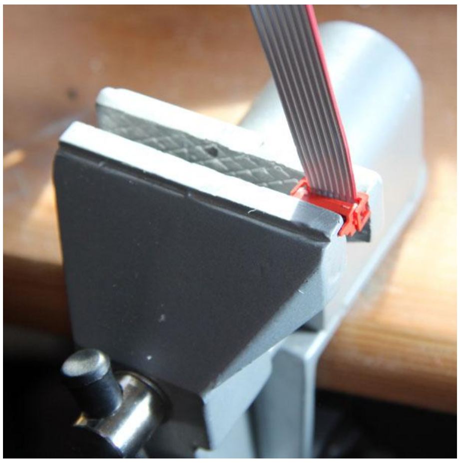
The manufacturer's part number for the 8 way IDC plug is: **7-215083-8**
You can get them from here:

Digikey code: A99462CT-ND
Farnell code: 149070

Connector guide:

Crimping connectors to the ribbon cable without an expensive tool:

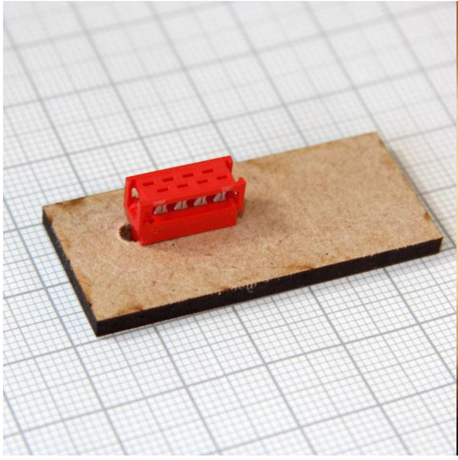
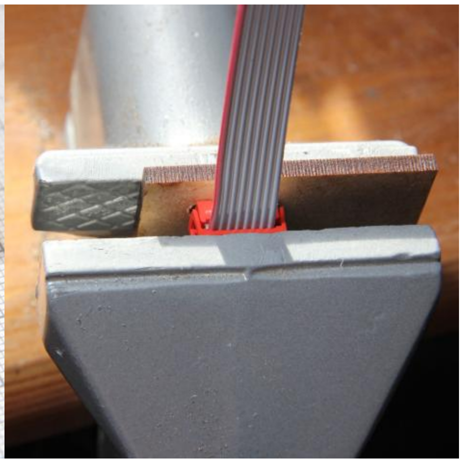
With a bit of care you can crimp the connectors onto a ribbon cable using just a small bench vice. The most basic way is to place the connector in the side of the vice as shown, so the polarising pin is on one side, and then close the vice.

The connectors have a polarising tab on one side – the contact closest to this tab connects to the ground pin when it is plugged into the HUB-ee. Ribbon cable usually has one of its conductors marked in a different colour so you can use this to make sure that all the connectors on a cable are the right way around – the polarising tabs should always be on the same side – beware though, a lot of ribbon cables use red (like in the photo) but don't assume that the red wire is positive!

A slightly better method is to get a small block of wood and drill a 2mm hole in it, the connector polarising pin can then poke through this hole and allow you to get the connector neatly in the centre of the vice, which results in a slightly better crimp.

Copyright © Creative Robotics Ltd 2013

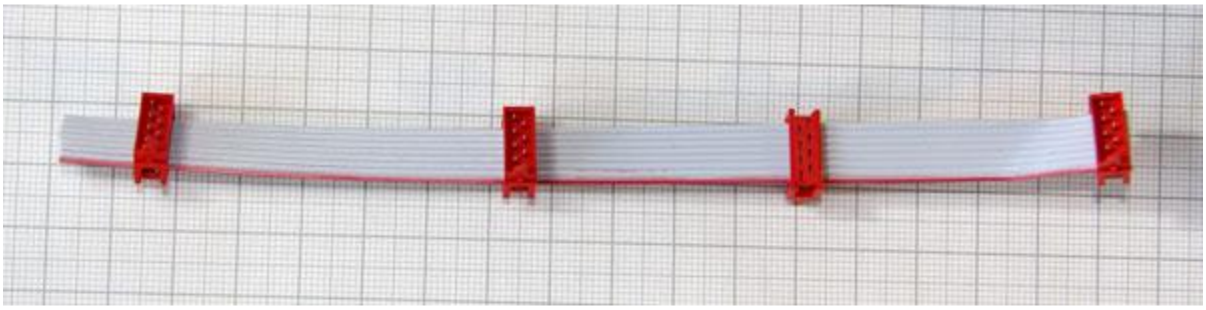
The best method is to buy specialised crimp tooling, at a cost of several hundred dollars – we prefer the vice methods!

Some distributors also supply pre-made cables.

Connecting several wheels to one cable:

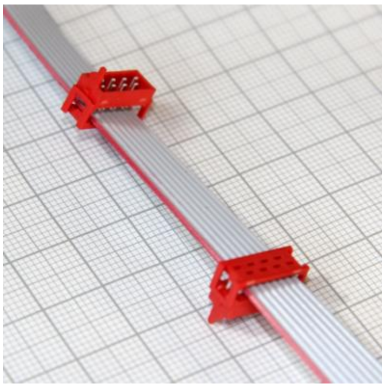
OK, so you want to make a tank bot with multiple powered wheels. You can easily crimp several connectors onto one cable so that all the wheels get the same control signals but there are a few things to beware of.

Firstly, each wheel can draw up to 1A of current when stalled, which is about the same as the current rating for ribbon cable, so if you plug ten wheels onto one cable and stall them all you are going to put 10 amps through a cable rated for only 1A. We are not suggesting that you shouldn't do this, just that you shouldn't be surprised if your cable starts to melt, or catches fire.



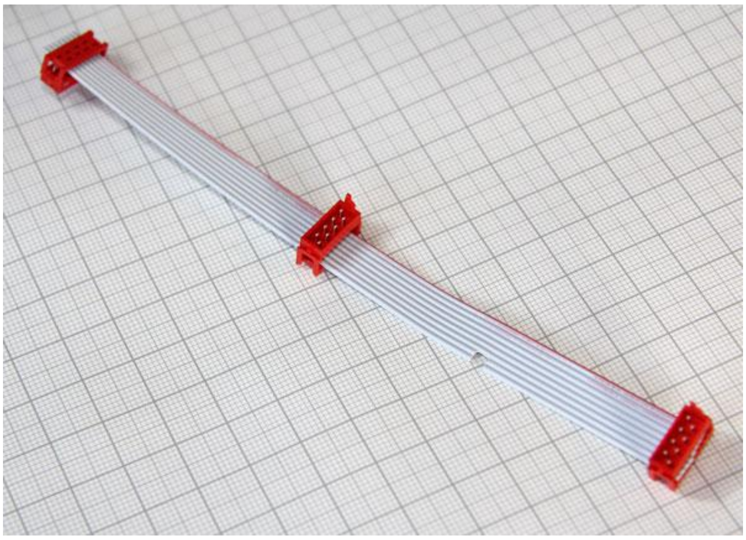
Connectors can go either way up, just so long as the polarising tabs are all on the same side.

Copyright © Creative Robotics Ltd 2013



When you put two wheels on the same cable you end up connecting their quadrature sensor outputs together. These outputs are protected by 1k resistors so although you won't damage them, you won't get any useful quadrature signals from the wheels.

A way around this is to cut the two sensor wires connecting the first wheel in the chain to the rest of the chain, now you sill get nice clean sensor signals from that wheel, but none from the others. The photo below shows a cable with the sensor wires cut – The connector on the far left would plug into the controller and the middle connector would plug into the wheel from which the sensor signals would be read.



Copyright © Creative Robotics Ltd 2013

What the sensors do:

The wheel has a pair of reflective optical sensors that shine on a code disk in the outer rim of the wheel. Called a Quadrature Encoder, these sensors pick up the black and white alternating pattern of the code disk as the wheel turns and convert it into electrical pulses. Because of the positions of the sensors relative to the black and white stripes the two electrical outputs will produce pulses in a particular pattern that depends on the direction of the wheel – To cut a long story short, you can use a microcontroller to read the direction and speed of the wheel, and with a bit of clever control theory (A PID control loop) you can exercise some fairly precise control how fast the wheel is moving along with keeping track of how far the wheel has moved.

For more information on Quadrature encoders and PID control, start with these Wikipedia pages:

http://en.wikipedia.org/wiki/Rotary_encoder#Incremental_rotary_encoder

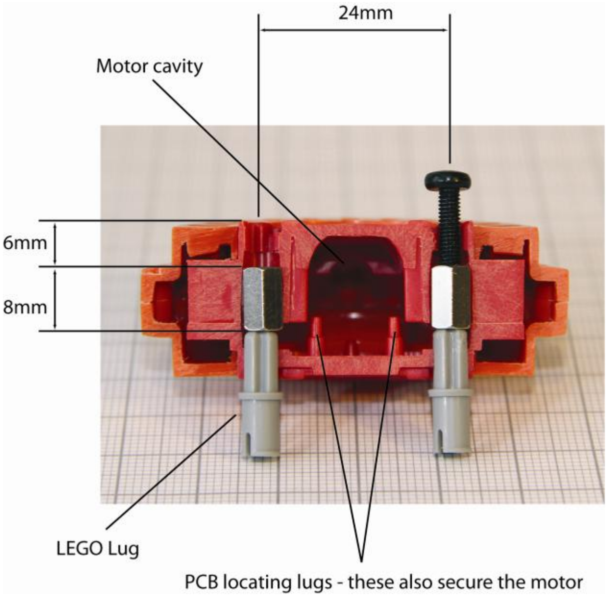
http://en.wikipedia.org/wiki/PID_controller

Copyright © Creative Robotics Ltd 2013

Mechanical stuff:

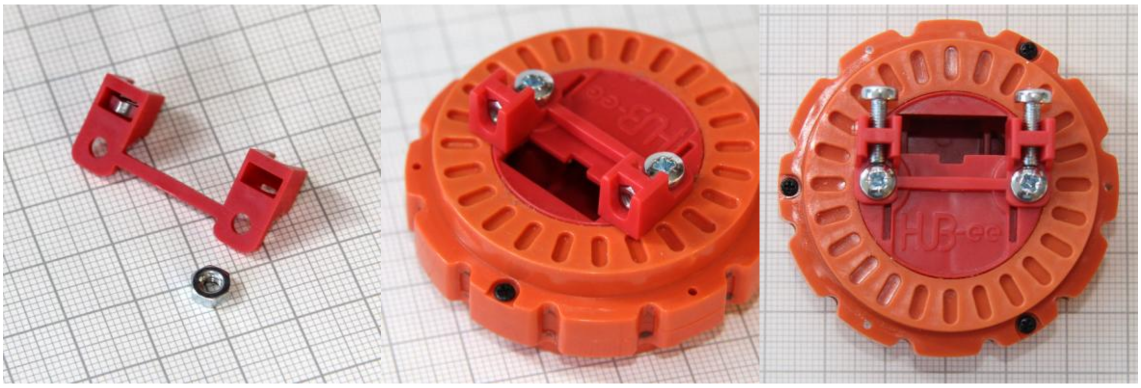
Each wheel has a pair of metal threaded inserts inside so you can bolt it onto things securely; these will either have a metric M3 thread, or an imperial 4/40 thread, depending on the one you bought. These can be removed and replaced if you want. The bolt holes are also designed to accept the tip of a LEGO axle so you can incorporate it into your favourite LEGO creations.

The threaded inserts are 8mm long and sit in the centre of the wheel, that means that any screw you use has to go in 6mm deep before it will engage with the thread – for example, if you want to bolt the wheel to a 3mm thick piece of plastic then you are going to need a screw that is around 12mm long.



Take care when bolting the wheel – try not to over tighten the screws – the current design is not quite as strong as we wanted where the inserts are so if you over tighten the screws you can cause some minor damage to the insides – fortunately not enough to stop the wheel working!

Each wheel also comes with a handy right angle bracket designed to make it easy to attach to a flat chassis – ok, we lied a bit, the bracket is *almost* right angle (actually 89 degrees) thanks to the requirement for it to come out of the mould easily. It has a pair of slots so you can put bolts inside and have something to screw a screw into, these slots will take a metric M3 bolt, or a small 4/40 bolt (Some 4/40 bolts are too large)



The screws that go in at the top, through the nuts (see pic above), will press up against the screws that bolt the bracket to the wheel if they go in too far.

Copyright © Creative Robotics Ltd 2013

As an alternative mounting solution for people who prefer imperial screws you can always try this metal bracket from Sparkfun:
https://www.sparkfun.com/products/10228

Which actually comes from here:
http://www.keyelco.com/products/specs/spec37.asp

Copyright © Creative Robotics Ltd 2013