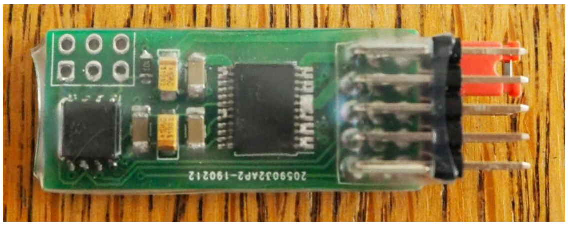
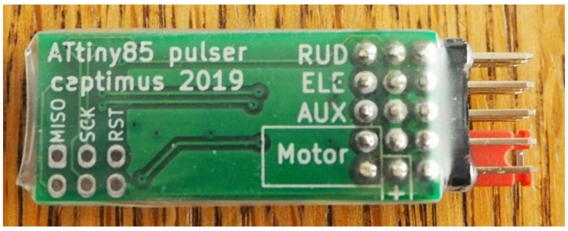
ATtiny85 pulser

The ATtiny85 pulser is a small device intended to be installed in radio control models: it allows a standard modern receiver to control ‘Pulse Proportional’ waggling (flapping, oscillating) control surfaces of the ‘Galloping Ghost’ or ‘Adams Actuator’ type.





The pulser can operate motors / actuators that work at voltages up to 15V, and at currents up to 2.4A (average) or 6.4A (peak).

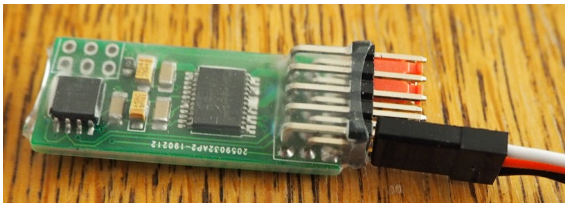
The main ‘logic’ of the board operates at a voltage of 3.3V to 6V. This means you connect it to your receiver just like a standard servo. You must be careful not to exceed 6V and not to wrongly connect the + and Gnd ‘servo’ wires, as this may destroy the pulser.

If your motor / actuator is suitable for being powered by your receiver battery or ESC, then you may fit a small link between two pins (the red link shown in the photos) and the only connections to the pulser are then the ‘servo’ connection(s) from your receiver and the two wires to your motor / actuator.

When you wish to use a separate battery to power your motor / actuator, the link is removed and the separate battery is connected to the pulser’s ‘+’ pin and Gnd.

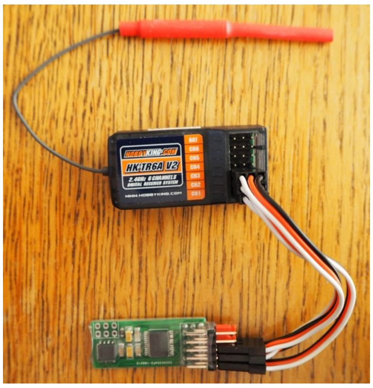
Connections to receiver

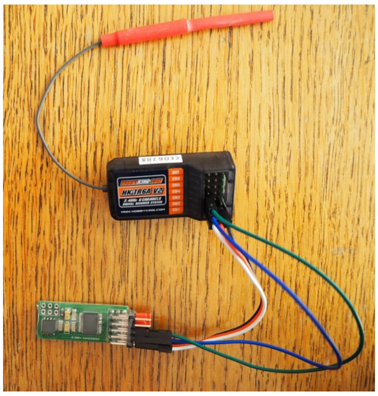
In the simplest installation, only one three-wire cable (like a normal servo cable) need be connected between the pulser and your receiver. I recommend using the so-called ‘Male to Male Servo Extension Lead Cables.’ These can be found on eBay in various lengths from 100mm to 300mm and in pack sizes of 1, 2, 5, 10, … costing about £0.50 each at the time of writing (March 2019).

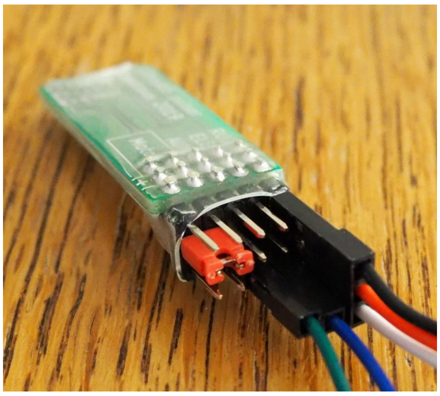


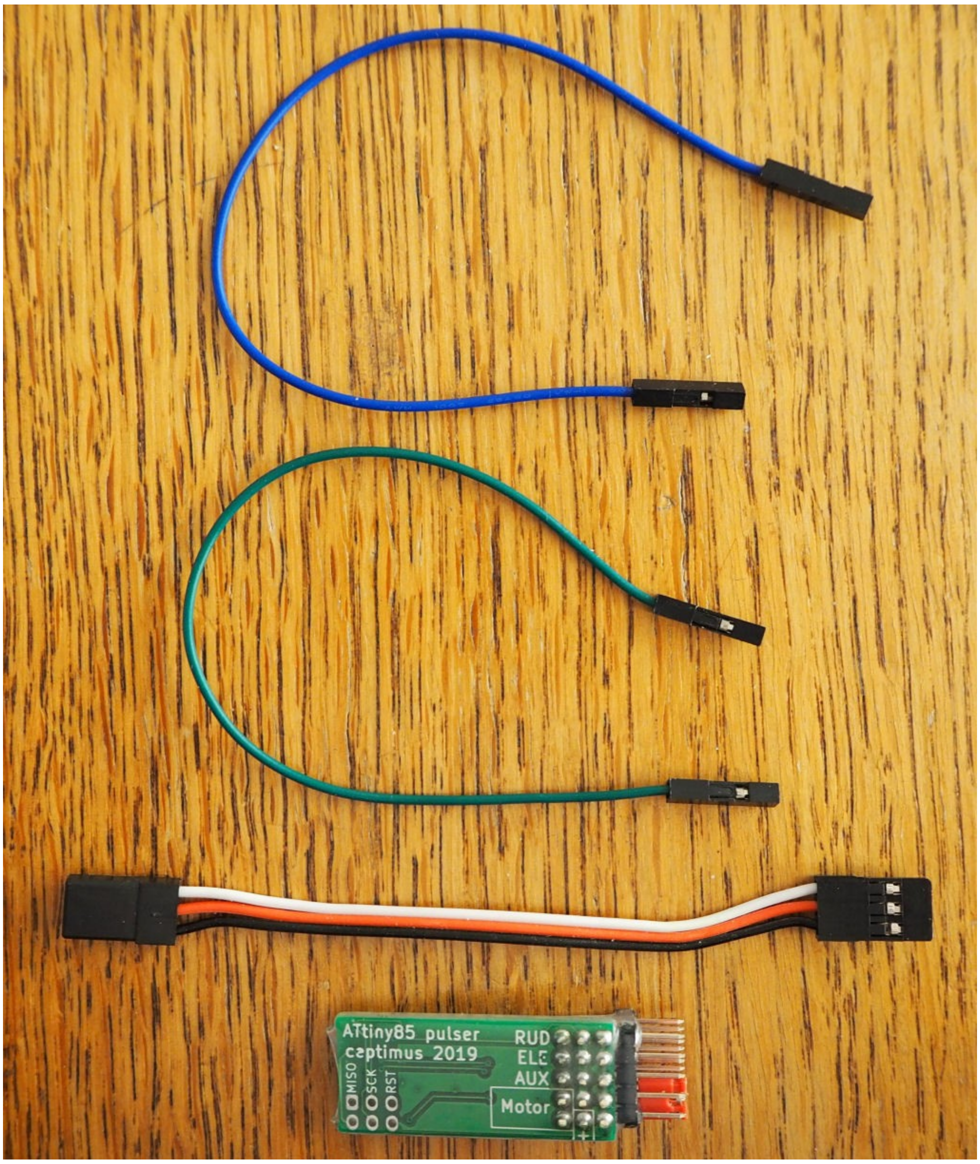
You may, of course prefer to make your own leads, possibly by soldering together two leads cut from old broken servos. If you are intending to power your motor / actuator via this cable (and the pulser’s link) make sure that the cable has thick enough wires to carry the necessary current.

To provide greater control over the pulsing rate and power you can connect one or two additional channels from your receiver to the pulser. To do this you may use additional three-way cables like the first one. The ground wires of all the cables (usually the darkest colour, black or brown) are all connected to each other at both the receiver and pulser, and the power wires (usually red) are likewise connected to each other, so for any additional channels you may choose just to connect the ‘signal wire’ which on standard servos is the lightest colour – white, orange, or yellow.





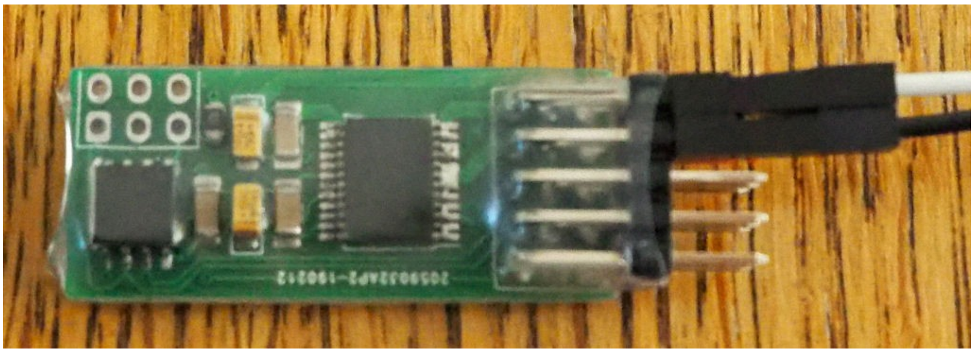




For details on how the extra (ELE and AUX) channels work, see the ‘Control channels’ section below.

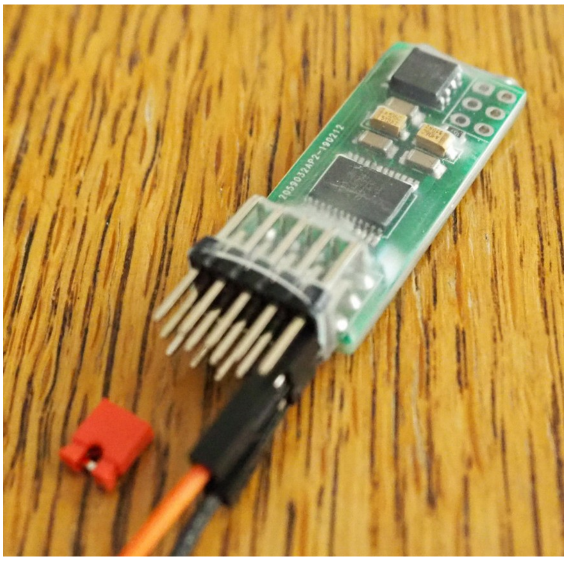
Connections to motor / actuator

The motor or actuator is connected to the two pins marked ‘Motor’. Swapping the wires will reverse the left/right operation of the motor / actuator.



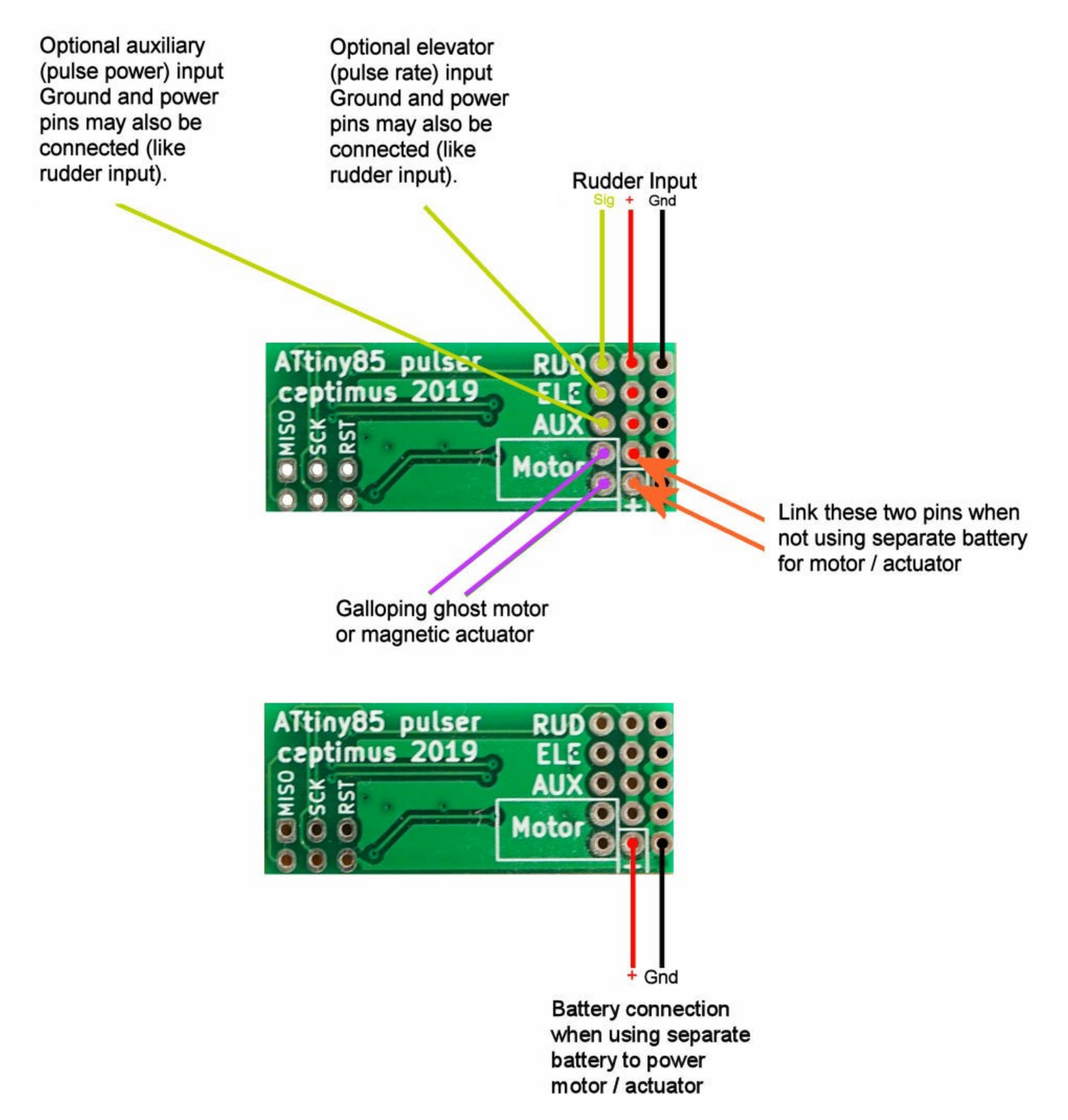
Connection of a separate motor / actuator battery

If you wish to use a separate battery to power the motor, remove the red link and join the positive battery wire to the ‘+’ pin and the negative (ground) battery wire to the Gnd pin below.



Connections – summary

The image below summarizes all the connections.



Control channels

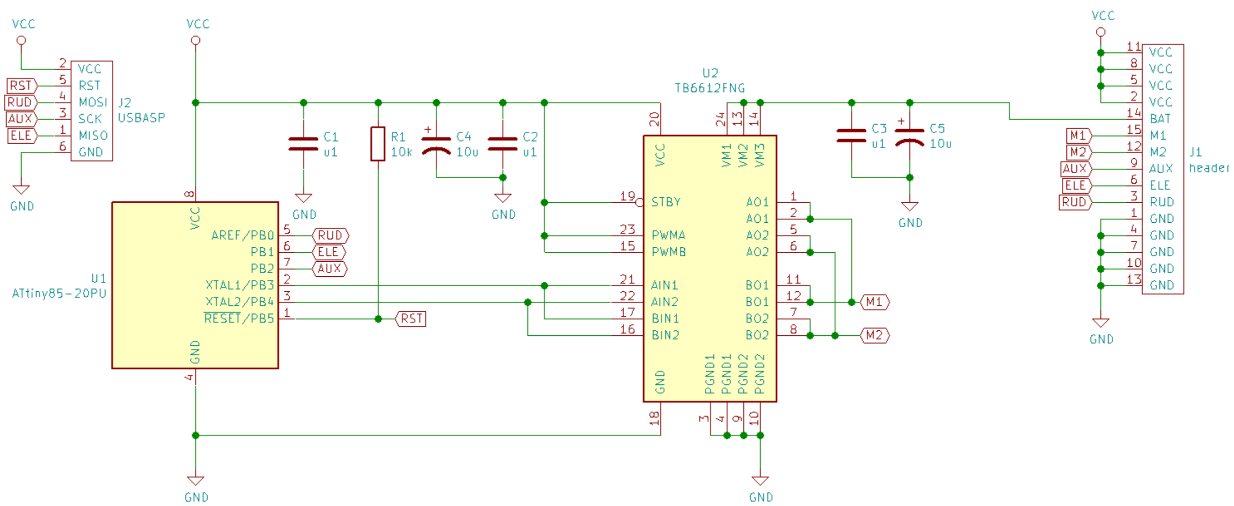
RUD: This is the primary control channel and controls the 'bias' of the motor / actuator drive. When your transmitter stick for this channel is centred, the motor/actuator is driven alternately left / right and the amounts of 'left' and 'right' time are both the same. When correctly trimmed this should result in you model flying straight. As you move your transmitter stick left and right, the proportion of the time the motor drives left / right changes. The cycle time (pulse rate) is kept constant.

ELE: This optional channel controls the pulsing rate from fast to slow. This channel is necessary if you wish to implement elevator control via a Galloping Ghost or Rand-type actuator, but even for rudder-only motor or magnetic actuator models, you may choose to connect it to a ‘spare’ channel of your receiver. On your transmitter you can then drive this channel from a preset value (or a pot) so that you can adjust the pulsing rate to your preference. If you choose not to connect to the ELE channel, the pulser defaults to operating at a constant, fairly fast, pulse rate that suits the majority of motors / actuators.

AUX: This optional channel controls the proportion of time the motor / actuator is powered. You may choose to connect this to a ‘spare’ channel on your receiver, and use a preset, pot, or switched channel on your transmitter to control how much power is provided to the motor actuator. I like to have a two-position switch on my transmitter operating this channel, and adjust the two switch control positions on the transmitter so that one position is 0% power (off) and the other is tuned to give the optimum power needed by the model’s motor / actuator / battery combination. I find it useful to be able to switch off the flapping while setting the model up in the pits, and after landing before disconnecting the battery. If you choose not to connect the AUX channel, the pulser defaults to operating at close to 100% power (which is the typical behaviour of most vintage Pulse Proportional systems and so suits standard rand-type and Adams-type actuators).

Technical

The pulser uses an Atmel (now Microchip) ATtiny85 microcontroller and a Toshiba TB6612FNG driver chip.



The program (“sketch”) for the microcontroller was written for the Arduino IDE using the Arduino system’s standard language (a dialect of C++). If you wish to inspect the sketch, or modify it to provide different functionality, you can download the sketch from http://ceptimus.co.uk/pulserATtiny85.zip Simple modifications would be to alter the default pulse rate or power to suit your actuator when you don’t wish to use the ‘ELE’ and ‘AUX’ channels for that purpose.

To upload a modified sketch to the pulser you need a programming device – an ICSP such as the popular USBasp. This connects using the standard 6-pin connector to the pulser’s ICSP header pads. Three of the header connections are labelled on board: MISO, SCK, RST, the other three pins: VCC, MOSI, GND, are not labelled, but are laid out in the default ICSP format. If your pulser board has the default heat-shrink covering you will have to remove or puncture it to gain access to the pins. If you are conducting extensive tests you may wish to solder a 6-pin header to the board, but for occasional program changes it's sufficient to just push a header into the holes and lean/twist it to one side while programming so that it makes good temporary connections to the holes. All the pins with the exception of RST are also available on the normal 'servo connection' pins, so you may just choose to make up a connection harness and use those, pushing a single pin through the heat-shrink to contact the RST pin.

Your Arduino IDE will need to be configured to work with ATtiny Microcontrollers (this is simple – you can find many tutorials on the Web explaining how). Then under tools choose:

Board: ATtiny25/45/85

Processor: ATtiny85

Clock: Internal 16 MHz

Programmer: USBasp (or whatever ICSP you use)

… and just compile and upload the sketch in the usual way.

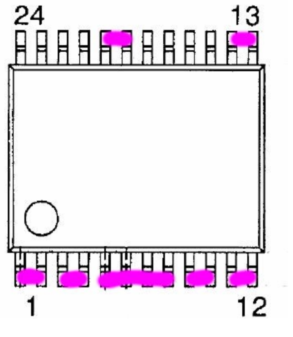
Videos

You can watch videos demonstrating/explaining the operation of the pulser at:

https://youtu.be/rZu2foaUOLs and https://youtu.be/MqPz3oUb888

Further videos showing models flying using this pulser may be added in due course. Watch for them on the same, ceptimus, YouTube channel.

Soldering



Don't worry if you notice apparent 'solder shorts' on the TB6612FNG chip (the larger of the two chips on the board). Some pins are intentionally connected via the tracks on the PCB: when soldering, the solder tends to bridge across adjacent connected pins. This is nothing to worry about and, if anything, probably helps a little by lowering the resistance between pins that are already linked anyway. The image at the left shows which pins are linked.

I give each board a thorough inspection and test before shipping. Let me know if you have any problems though, and I'll send a free replacement or refund. ceptimus@ceptimus.co.uk

Forum

You probably already know about it if you're reading this, but if not, I recommend you visit the Mode Zero forum http://mode-zero.uk/ It's free to register and post, and you will find lots of details about this pulser project and similar 'retro electronics' posts there – by myself and others. Also lots of interesting posts about retro modelling and associated topics.