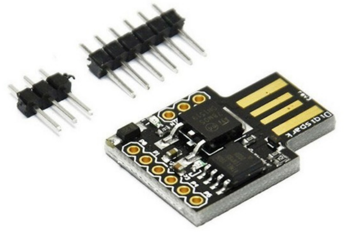
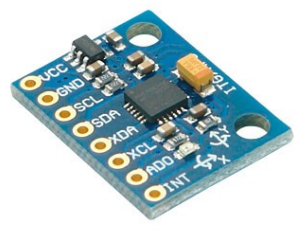
DIY gyro project

The completed gyro is a single-axis stabilization unit for model helicopters, or other similar models. Gain and operating mode can (optionally) be controlled from the transmitter. The gyro can work in normal “rate mode” or “heading hold mode”. Throughout this manual, I will assume you’re using it for the tail rotor control of a model helicopter. When talking about radio channels, I’ll often use the term ‘rudder’ rather than ‘tail rotor’.

The gyro has two plugs that connect to your radio receiver – plugging in like servos normally would. These connectors provide both power and control signals to the gyro. They also provide the power that is fed out to the socket where you plug in your tail rotor servo. There is another ‘servo-style’ plug with unusual wire colours, but this should not be plugged into your receiver – it is for configuring or reprogramming the gyro, and should be left disconnected in normal use.

The components used to build the gyro are a Digispark ATtiny85 board, an MPU6050 gyro module, a 3D-printed case, and some servo-style connectors and wires.





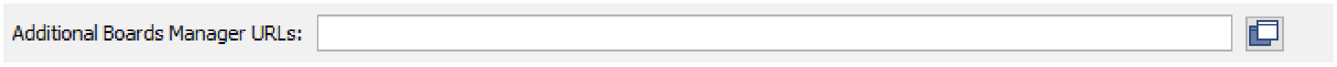
At the time of writing (July 2020), the Digispark boards are available new on ebay for £1.19 each, and the MPU6050 boards for £1.51 each.

You’ll also need a programming device to write the program to the ATtiny85, or if you wish to edit the settings and the recommended one is a USBASP. You don’t actually need the ribbon cable and adapter that converts from 10-pin to 6-pin for this project, but they are worth having for other Arduino-type projects. You can get the complete kit of all three parts new on ebay for £1.71 at the time of writing.

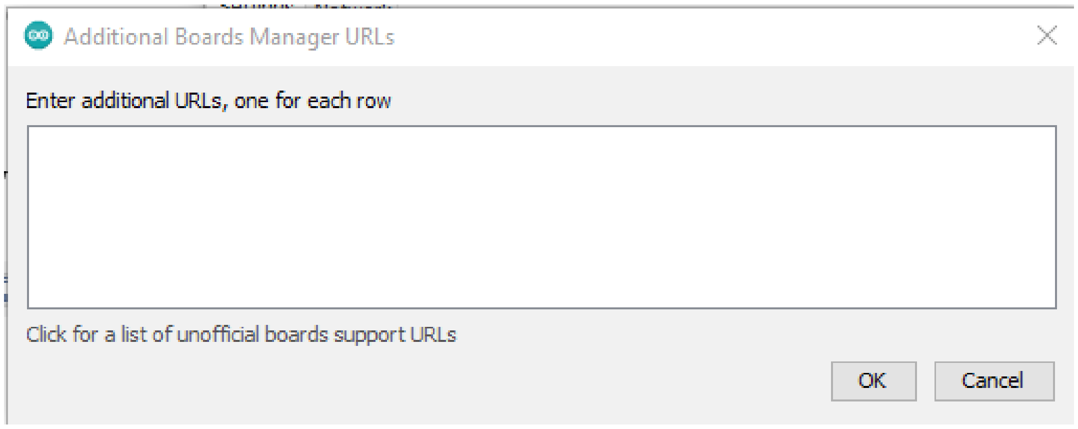
Configuring the Arduino IDE to support the ATtiny85

By default the IDE doesn't support the ATtiny85. These are the steps to take to add that support (your IDE will still work with all the existing board types it supports already).

Select the Preferences item from the IDE's File menu. Towards the bottom of the resulting window you'll see:


Additional Boards Manager URLs:

Click on the little 'window' icon at the right of that line and another window will open up


Additional Boards Manager URLs

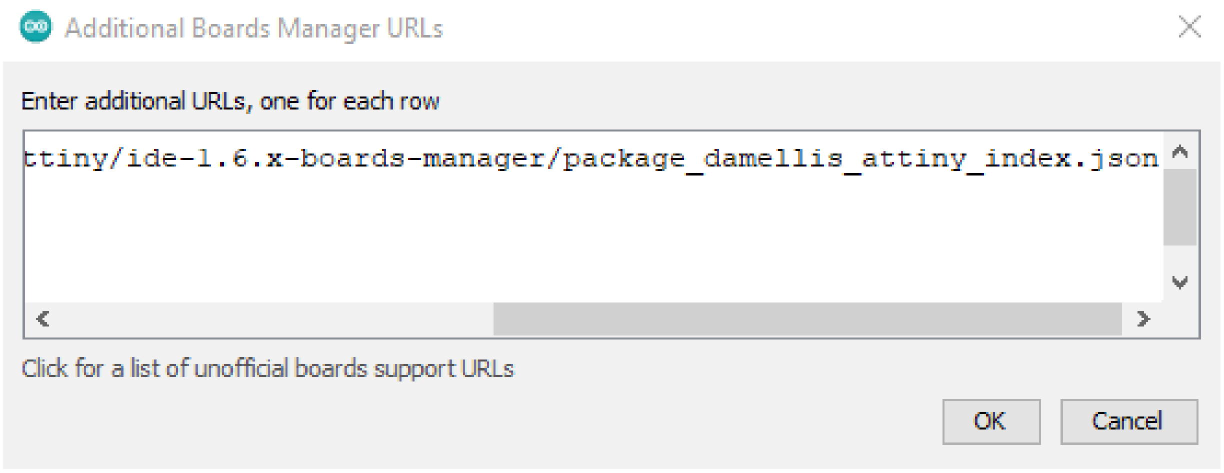
Enter additional URLs, one for each row

Click for a list of unofficial boards support URLs

Your text box may be blank, as shown above, or already contain one or more lines if you've previously added additional board(s). The line you need to add for the ATtiny85 is:

https://raw.githubusercontent.com/damellis/attiny/ide-1.6.x-boards-manager/package_damellis_attiny_index.json

You can either copy that from here, or you'll find it if you click on the text in the window, 'Click for a list of unofficial boards support URLs'. Type or paste that line into the window, on its own line, so that it looks like this:


Additional Boards Manager URLs

Enter additional URLs, one for each row

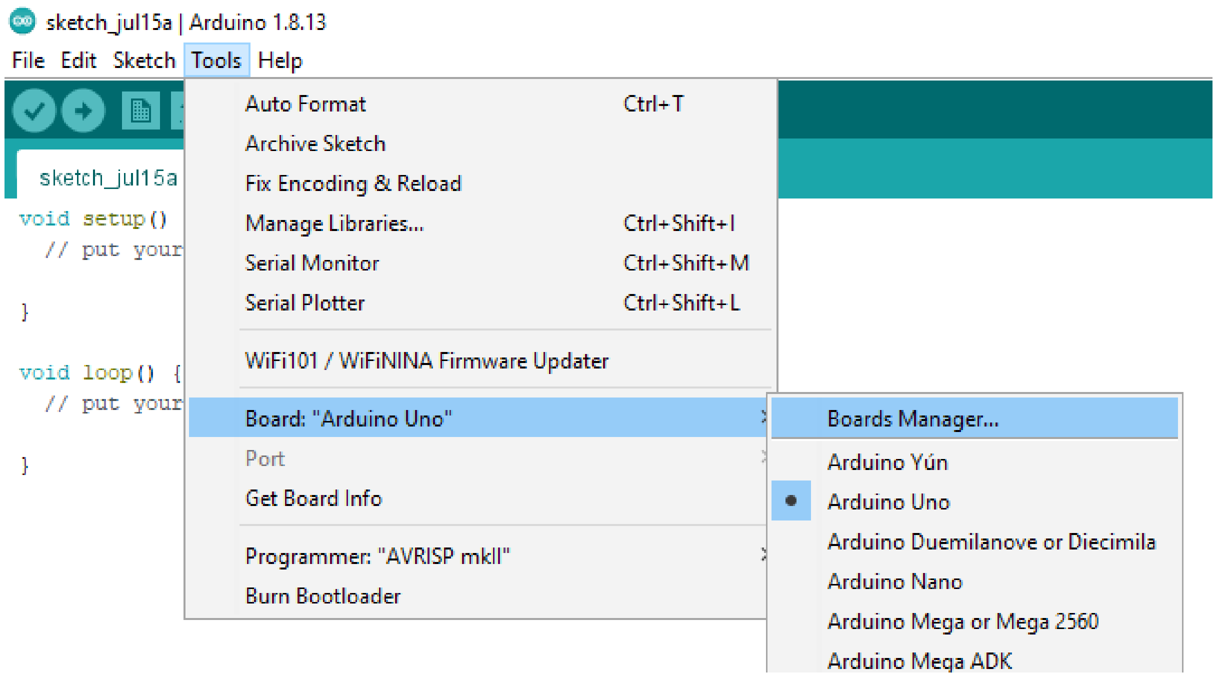
```
ttiny/ide-1.6.x-boards-manager/package_damellis_attiny_index.json
```

Click for a list of unofficial boards support URLs

Note that if you look on the ‘unofficial boards support URLs’ link page, you’ll find several options that support the ATtiny85. These probably all work fine, but I’ve not tried them – the one I’ve used is the one shown above the ‘damellis’ one, made by David Mellis.

Once you’ve added the line in the window as shown above, click on the OK button to close the window, and then close the Preferences window in the usual way (click on the little X at the top right corner).

Now go to the “Board:” item of the Tools menu, and click on Boards Manager...


sketch_jul15a | Arduino 1.8.13
File Edit Sketch Tools Help
sketch_jul15a
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino Uno"
Port
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Boards Manager...
Arduino Yún
Arduino Uno
Arduino Duemilanove or Diecimila
Arduino Nano
Arduino Mega or Mega 2560
Arduino Mega ADK

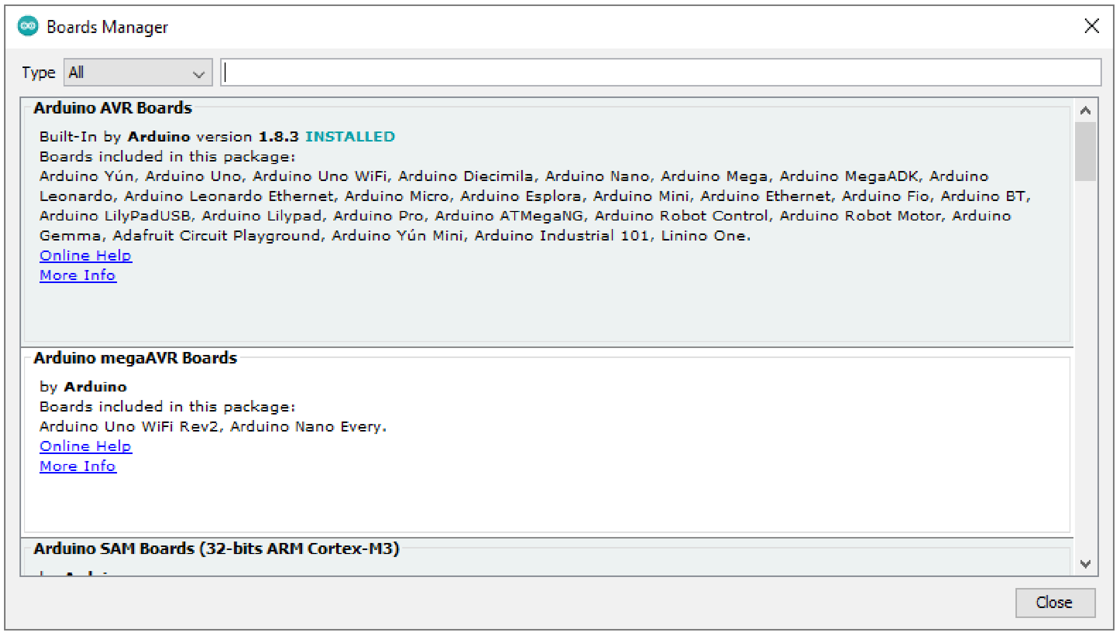
```
void setup()
  // put your

}

void loop() {
  // put your

}
```

A new window will open.



Arduino AVR Boards

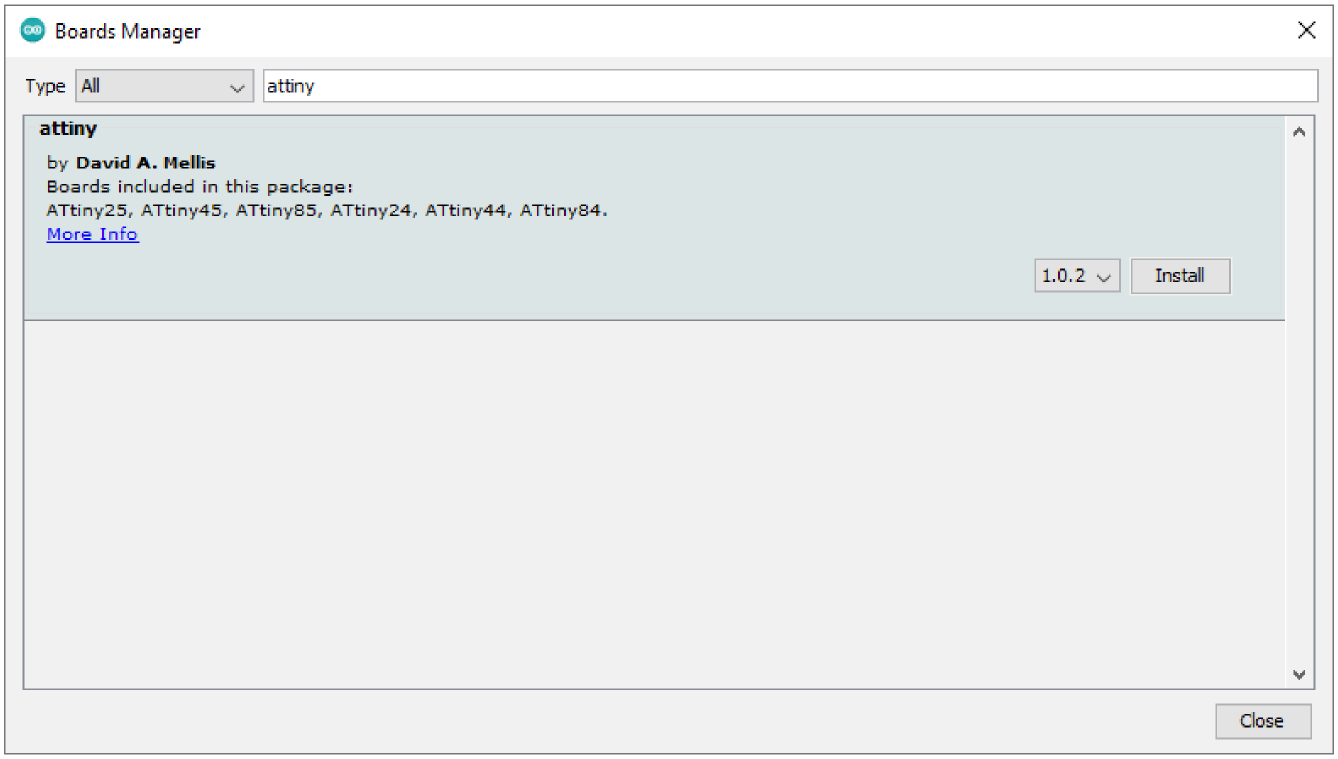
Built-In by **Arduino** version **1.8.3** **INSTALLED**
Boards included in this package:
Arduino Yún, Arduino Uno, Arduino Uno WiFi, Arduino Diecimila, Arduino Nano, Arduino Mega, Arduino MegaADK, Arduino Leonardo, Arduino Leonardo Ethernet, Arduino Micro, Arduino Esplora, Arduino Mini, Arduino Ethernet, Arduino Fio, Arduino BT, Arduino LilyPadUSB, Arduino Lilypad, Arduino Pro, Arduino ATMegaNG, Arduino Robot Control, Arduino Robot Motor, Arduino Gemma, Adafruit Circuit Playground, Arduino Yún Mini, Arduino Industrial 101, Linino One.
Online Help
More Info

Arduino megaAVR Boards

by **Arduino**
Boards included in this package:
Arduino Uno WiFi Rev2, Arduino Nano Every.
Online Help
More Info

Arduino SAM Boards (32-bits ARM Cortex-M3)

Scroll down, or type attiny in the search box at the top, and you should see the attiny option.

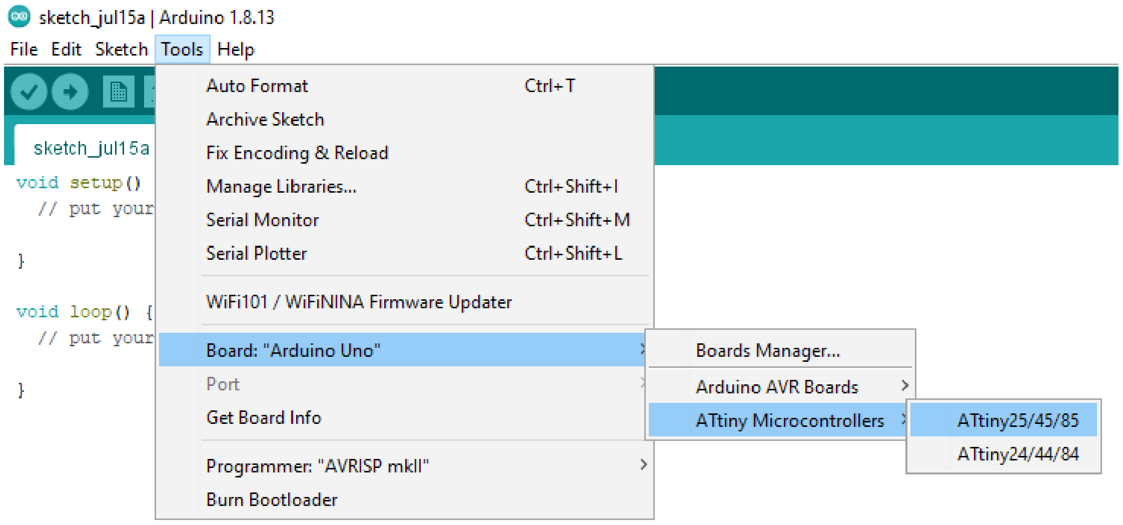

Boards Manager

Type All attiny

attiny
by **David A. Mellis**
Boards included in this package:
ATtiny25, ATtiny45, ATtiny85, ATtiny24, ATtiny44, ATtiny84.
More Info

Click on the Install button. Then close the Boards Manager window.

Now you can select Attiny25/45/85 from the Board item of the Tools menu.

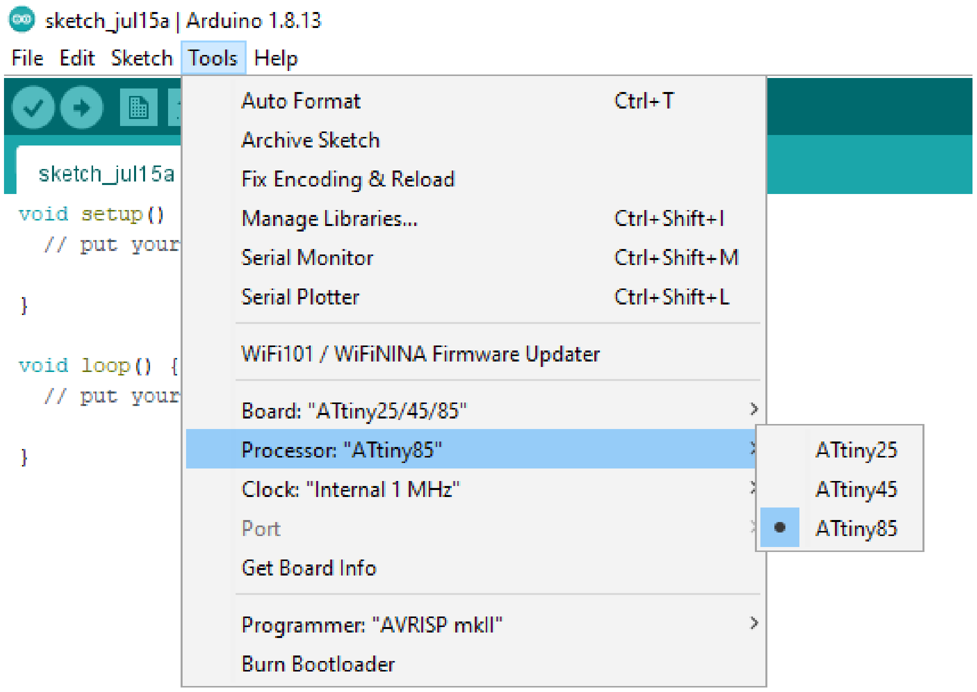

sketch_jul15a | Arduino 1.8.13

```
void setup()
  // put your
}

void loop() {
  // put your
}
```

Note that your boards may not be laid out as shown in the image above. Depending on which version of the Arduino IDE you have, the ATtiny Microcontrollers item may not be in its own sub-menu, and may be part of a much longer list of boards you need to scroll through. As long as you select ATtiny25/45/85 as the Board: it makes no difference.

Now from the Processor: item of the tools menu, select ATtiny85



sketch_jul15a | Arduino 1.8.13

File Edit Sketch Tools Help

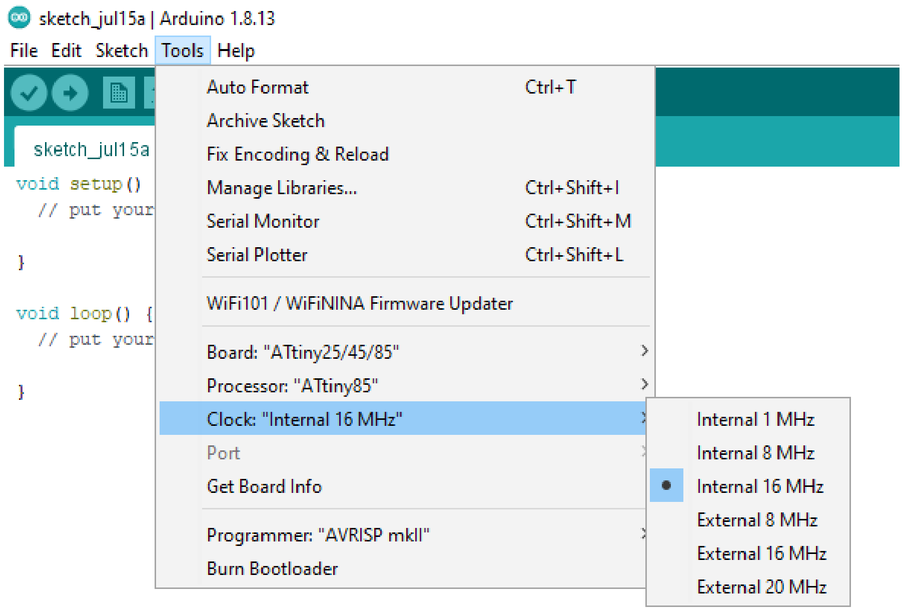
```
void setup()
  // put your

}

void loop() {
  // put your

}
```

...and from the Clock: item of the Tools menu select Internal 16MHz



sketch_jul15a | Arduino 1.8.13

File Edit Sketch Tools Help

```
void setup()
  // put your

}

void loop() {
  // put your

}
```

That’s the process completed, and you’re now ready to compile the sketch.

Connecting the USBasp programmer to the gyro

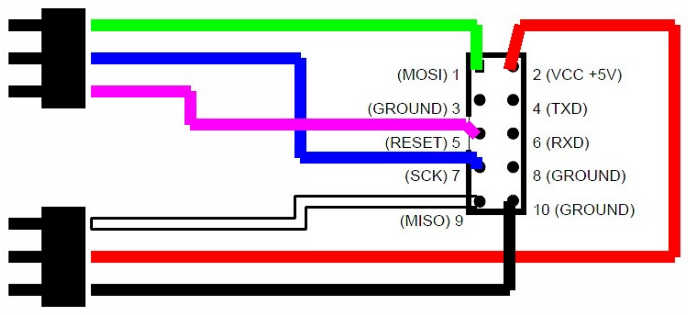
The colours mentioned in this section assume that you've used the same colours I did when building the prototype. Obviously if you've used different colours then you need to change the connections accordingly.

The easiest way to make the connection is to use some single wire male-to-female jumper wires such as the ones shown here.

If you're going to use the programming lead repeatedly, you can glue the connectors together at each end to form plugs and sockets suitable for connection to the gyro and your USBasp. Use a tiny amount of superglue, and be careful not to accidentally permanently glue the leads into your USBasp socket!

One of the leads from the gyro has green/blue/violet wires and these connect to MOSI, SCK, and RST respectively. The other 'programming' wire is MISO which is normally used in non-programming mode as the signal pin of the AUX channel. On my prototype this was a white wire. The red and black wires of the same 'AUX' plug can also be conveniently used to supply power to the gyro when programming it – your USBasp can provide sufficient power, though you should unplug the remaining gyro connectors from your receiver and servo during programming. An alternative way of providing power is to plug a normal 4-cell nickle battery (or other 5V source) into the gyro's servo socket while programming.

If you're using the 10-pin connector of your USBasp, then, looking at the pins of the connector, the connections are shown below.


(MOSI) 1
2 (VCC +5V)
(GROUND) 3
4 (TXD)
(RESET) 5
6 (RXD)
(SCK) 7
8 (GROUND)
(MISO) 9
10 (GROUND)