TC4+ Arduino Shield
Bluetooth configuration

April 17, 2019

Bluetooth Module Configuration

To communicate with the board using Bluetooth, you can use either a HC-05 or a HC-06 module. With the HC-05 (but not with the HC-06), you also get the ability to wirelessly upload sketches to the Arduino directly from the Arduino IDE.

In either case, there is a small amount of configuration to be done:

1. First, you need to connect directly to the Bluetooth module's serial console. If you have a USB-to-TTL-serial adapter around, you could connect that directly to the BT module. Otherwise, we will use the Arduino as a serial bridge:

   (a) Attach the BT module to the TC4+ board. For a HC-05, pins STATE to VCC connect to the 5-pin header. EN pin is left unattached. For HC-06, the 4 pins of the module go into the four header pins that are away from the board edge, and closer to the thermocouple header. The leftmost pin on the header is left unconnected. (The HC-06 might have an unsoldered connecter in the same place labelled STATE.)

   (b) Using the jumpers right next to the BT header, connect the BT module to IO4 and IO5 pins. (Put the jumpers on the right position.) We will use the Arduino to relay commands between the native serial interface (connected to the host PC via USB) and a software serial interface on these two pins.

   (c) Copy and paste the bluetoothATmode code below (figure 1) into an empty Arduino sketch, and flash it to the Arduino via USB. You may have to adjust the baud rate of the software serial. Some modules default to 38400 baud, some 9600, and possibly others too.

   (d) Open Serial Monitor in the Arduino IDE, and type "AT+NAME". If you get a response, good. If not, see the preceding step.

2. You should now be able to configure the BT module using AT commands. There is various resources on that online, and the exact command set differs greatly between modules and firmware versions. For our purposes, the following should do the trick:

(a) AT+NAME should return the default name of the module, just to check that you are connected. AT+VERSION might give you a firmware version. Might be AT+NAME? and AT+VERSION? for some modules. Just AT should get an OK as a response.

(b) AT+NAMEcoffeeroaster or possibly AT+NAME=coffeeroaster should rename the board to “coffeeroaster”. Feel free to choose any name, obviously. (Optional.)

(c) AT+PIN1234 (or possibly AT+PSWD1234, or possibly either with a =) should allow you to set a custom PIN for pairing. (Optional.)

(d) AT+BAUD8 sets the baud rate to 115200. Other options include AT+BAUD1 (1200) or AT+BAUD4 (9600). Set this to whatever you have set in the sketch you intend to run on the Arduino. aArtisan defaults to 115200 I believe. (Strictly required! Arduino and BT module baud rates must match.)

(e) For wireless uploading of sketches with HC-05 only: Set AT+POLAR=1,0 to make the STATE pin go low whenever a new BT connection is made. (This will trigger the Arduino reset circuit whenever a new serial connection is made, thus enabling the Arduino IDE to talk to the board’s bootloader.)

3. That’s it! Return the BT jumpers to the left / SERIAL position, and short the BT_RESET jumper if desired. Flash whatever Arduino sketch you want to use with the board. Scan for Bluetooth devices on your computer, and connect to the BT module. (Note that it might still show up as HC-05 respectively HC-06 before you connect to it, even if you set a custom name earlier.) The module should now show up as a new serial port on your computer. You may select in in the Arduino IDE and in Artisan, and proceed as normal. Note that there is sometimes a small delay between when a program first accesses the serial ports, and when the BT connection is made. As a result, you might sometimes get an error in Artisan at first. Wait for a few seconds – temperature readings should eventually start coming in.

4. Troubleshooting: If it’s not working, check that you are in range. If it’s not that, then in Windows you could try turning off and on the BT serial port profile in device manager. If that doesn’t help, unpair the module and pair it again.

bluetoothATmode.ino

```
#include <SoftwareSerial.h>
SoftwareSerial BTSerial(4, 5); // RX, TX
String response = "";

void setup() {
// Serial connection to host PC:
Serial.begin(9600);
Serial.println("Type AT commands!");

// Serial connection to BT module:
BTSerial.begin(115200);
// You might have to adjust the baud rate!
}

void loop() {
// Check if BT module has sent data, and pass it to host PC:
if (BTSerial.available()) {
delay(10);
while(BTSerial.available()) {
response += (char)BTSerial.read();
}
Serial.println(response);
response = "";
}

// Vice versa: Check if host PC has sent data, and pass it to BT module:
if (Serial.available()){
delay(10);
BTSerial.write(Serial.read());
}
}
```

Figure 1: The sketch used to configure the Bluetooth module.

Jumper Configuration for Bluetooth

The BT_SEL connectors attach the Bluetooth module to serial (put jumper on middle and left side, labelled SERIAL) or IO4 and IO5 pins (put jumper on middle and right, labelled DIO45). Note that when the BT_SEL jumper is in the SERIAL position and a Bluetooth module is attached, the USB serial will not work. To use the USB serial, remove the Bluetooth module and place jumpers in DIO45 position.

A jumper across BT_RST enables Bluetooth reset via BT module STATE pin, for uploading sketches via Bluetooth.

Figure 2 shows these options.

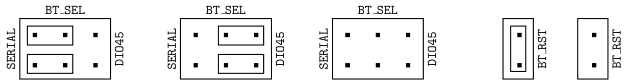

Figure 2: Jumper configuration for Bluetooth serial. From left to right:

- Bluetooth module attached to hardware serial interface
- BT module attached to IO4 & IO5 for software serial
- BT module unattached
- Bluetooth reset feature enabled
- BT reset disabled