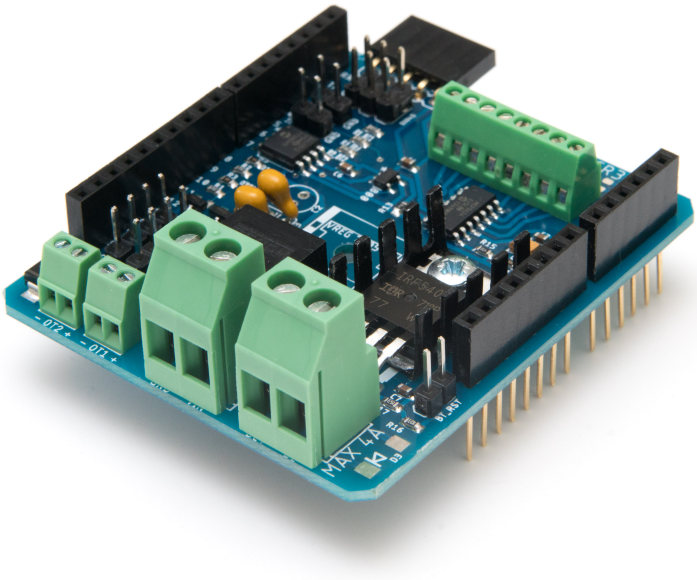
TC4+ Arduino Shield

June 25, 2022



1 Introduction

1.1 What?

The TC4+ is an Arduino shield providing a 4-channel thermocouple interface and driver logic for AC and DC loads. It is fully compatible to the established TC4-shield, which has been used extensively for home-made coffee roasters. The TC4+ has been designed specifically with small popcorn machine based roasters in mind: Integration of a voltage regulator and DC PWM driver enable control of a popcorn machine roaster with only one controller assembly (Arduino & TC4+ shield), a single DC power supply, and one solid state relay. A Bluetooth

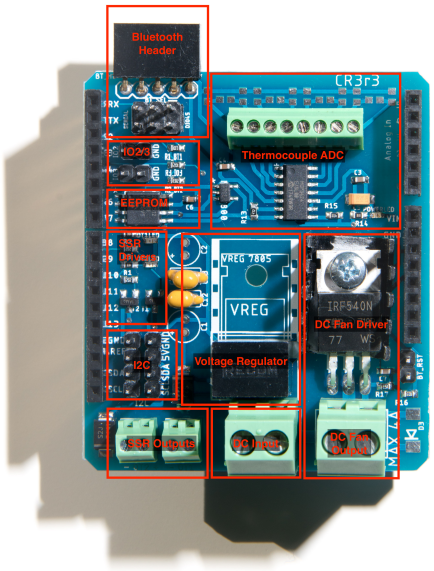


Figure 1: The TC4+ board and its on-board components.

header allows for wireless communication via an optional Bluetooth module (or even Wi-Fi! - coming soon). Full compatibility with TC4-based Arduino sketches gives access to a well-established ecosystem of home-roasting software.

1.2 Features

- 4-channel thermocouple interface using high-resolution ADC with cold-junction compensation for accurate roast monitoring.
- Full hardware-level compatibility with TC4-based Arduino sketches and established software ecosystem.
- Highly simplified control of popcorn machine based roasters: On-board voltage regulator and DC PWM driver allow popcorn roaster setup with a single DC power supply and one solid state relay.
- Designed with a high degree of flexibility in mind. Wide range of voltage regulators supported; extra I2C headers for displays and other peripherals; retains all TC4 features and full Arduino pinout.
- Wireless control with optional Bluetooth module. Even wireless uploading of Arduino sketches is possible.

1.3 Differences Compared to TC4

The TC4+ provides the same hardware as a TC4, plus additional extras: PWM DC driver, voltage controller and Bluetooth header. The main advantage of the first two of these is that DC fan setup is greatly simplified: A TC4-based DC

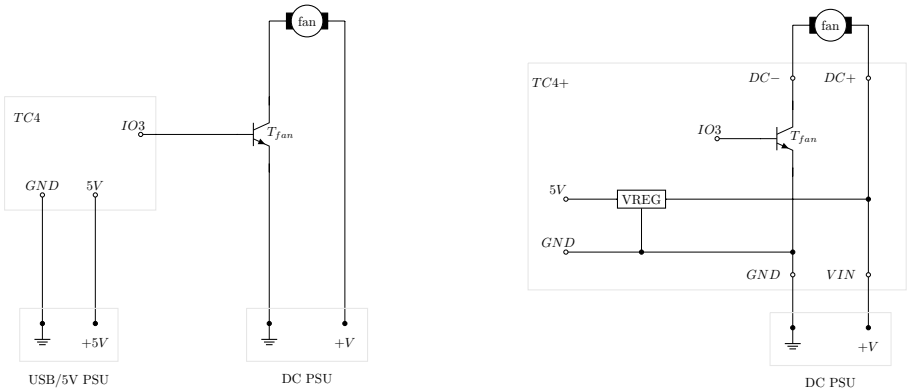


Figure 2: Left: Controlling a fan through a TC4 requires a separate MOSFET transistor (T_{fan}) board, as well as two power supplies - one for the fan, one of the Arduino & TC4. Right: The TC4+ has a MOSFET as well as a voltage regulator on-board, eliminating the need for a separate MOSFET and the separate PSU for the Arduino.

fan control requires a separate power MOSFET (likely mounted on a dedicated PCB or prototyping board), and usually two separate DC power supplies or additional voltage regulation hardware (supplying 5V for the Arduino, and a higher DC voltage for the fan). The TC4+ already includes a power MOSFET as well as a voltage regulator on board. Figure 2 illustrates simplified sample connection layouts for TC4 and TC4+ DC fan setups.

There is only a few minor points where the TC4+ is not a strict superset of the TC4: The TC4+ does not include a separate JeePort header, Analog In headers, or a reset button. The functionality of all these is still accessible through the standard Arduino pins if required. Everything else that is on the TC4 is also present on the TC4+.

1.4 Cost, Availability and Shipping

Both kits (board with SMD components already soldered, and loose through-hole parts) as well as fully assembled boards are available. Custom orders on request. Please send an email to coffee@gerstgrasser.net for up-to-date pricing and shipping information.

1.5 Questions

Can I use any TC4 software with this?

Yes. I've tested it with aArtisan, but any other TC4-based Arduino sketch should work just as well.

I don't need any of the additional features, can I use this just as I would use a TC4?

Yes. The ADC, temperature sensor, and EEPROM are the same chips as on a TC4, and all the additions are entirely optional.

Does this work with an AC fan? Do you also make a zero-cross detector?

Yes (with a ZCD), and no. I have designed this with DC fans in mind, so the board does not include a ZCD or any other hardware specifically for AC fans. It is still possible to attach an external ZCD board and us a random-fire SSR to control an AC fan, IF you can find or make a ZCD. At this time I am not making or selling a ZCD myself.

2 Detailed Information

2.1 Important Safety Information

The TC4+ board itself provides no safety features whatsoever. It is the user's responsibility to ensure they are designing and operating their equipment in a safe manner. This document will list several key steps that are necessary or strongly recommended for safe operation (see sections 2.3, 2.5, 3.2.1), but these are starting points only, not definite guidance; nor is the list of issues mentioned exhaustive by any means. Use this board and any contraptions made with it at your own risk. **Never leave unattended** while powered on, and have contingency plans ready.

2.2 Thermocouple Interface

The TC4+ provides a 4-channel thermocouple interface with cold-junction compensation through its MCP 3424 ADC and MCP 9800 on-board temperature sensor. The ADC channels are provided through an 8-pin 2.54mm-spaced screw terminal near the north edge of the board. Due to board layout constraints, the order of the pins is not ascending, but in order CH2-, CH2+, CH1-, CH1+, CH4-, CH4+, CH3-, CH3+. Figure 3 illustrates this layout.

Some common software (i.e. Artisan) by default expects bean temperature on channel 2. It is recommended to follow this for simplicity. By default "environmental temperature", i.e. temperature of the incoming air, is expected on channel 1. This is less relevant for hot-air (popcorn machine) roasters, and could be considered optional altogether.

By default, the thermocouple connections are attached directly and only to the respective pins on the MCP 3424 ADC chip. If it is found that additional filtering or buffering is needed, the board provides SMD soldering pads for resistors or capacitors between each of the pins and GND, and between each channel's positive and negative pins. These soldering pads are 0603 size, and located between the screw terminal and the north edge of the board. They are arranged as 4 H-shapes, with the four vertically arranged pads going between each pin and ground, and the two horizontally aligned pads going between

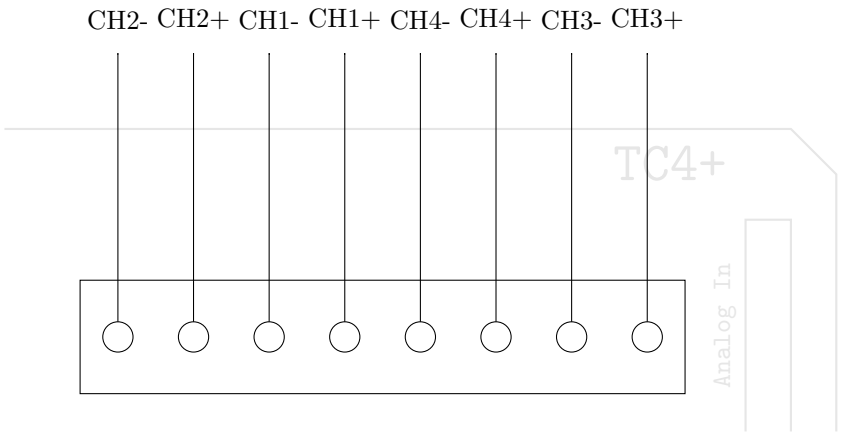


Figure 3: Layout of the thermocouple channels on the TC3 board.

positive and negative pins. This is entirely optional, and most users will have no need for this.

2.3 SSR Drivers

The board features two SSR drivers implemented with MMBT3904 transistors, and feature an activity LED for each. These are broken out into OT1 and OT2 headers near the south west corner of the board. Each of these channels can provide up to 200mA of current, which should be plenty for solid state relays. In a typical application, one channel would be connected to a SSR controlling a heating element, for instance.

Important: Please note that the TC4+ board by itself cannot provide protection against overheating of a load controlled via SSR. It is strongly recommended to implement appropriate safety measures. If a popcorn machine is to be modified, it might already contain a thermal fuse on its heater assembly. Do not remove / short this fuse. If your popcorn machine does not feature a thermal fuse, it is strongly recommended to add an appropriate thermal fuse in series with and close proximity to the heating element.

Important: In popcorn machines, powering the heating element but not the fan will lead to near-instantaneous overheating. See section 3.2.1 for further information.

2.4 Voltage Regulator

The TC4+ board has space for an on-board voltage regulator and smoothing capacitors. By default, the kit and fully assembled boards come with a R-78E5.0 switching regulator, and two 106 (10μF) ceramic capacitors. If for any reason a different voltage regulator is preferred, the board has space for any TO-220 packaged regulator or suitable drop-replacements. Additionally, the board has space for two capacitors on the supply side (C1 and C12) and two on the output side of the regulator (C2 and C13). When using a linear regulator with an input voltage much higher than the 5V output, a heatsink is strongly recommended.

The voltage regulator is fed from the same power supply as the on-board DC PWM driver (see next section). You must therefore check that the voltage regulator is appropriate for the DC fan's operating voltage; and vice versa, that the DC power supply can provide enough current for the DC load.

When using the DC IN header, it is recommended to connect to the board using Bluetooth, or if using USB, to use a USB optoisolator.

On v5 and higher of the board, a jumper header is provided next to the voltage regulator (2x1 header near the center of the board) labelled VREG_EN. This can be used to connect or disconnect the voltage regulator. When using USB, it is recommended to disconnect this jumper. When using Bluetooth and thus powering the board from the DC IN header, this jumper must be shorted. On v5 and higher, the onboard voltage regulator feeds 9V into the Arduino's VIN rail rather than 5V into the 5V rail.

2.5 DC PWM driver

The board features an on-board power MOSFET transistor to drive small DC loads, with PWM control. This is controlled via the Arduino's IO3 pin. By default, the kit and assembled boards come with an IRF540N transistor. This is rated for up to 100V and 33A. Note however that the copper traces on the board are only rated for around 4A. A current much higher than 4A is strongly discouraged.

The MOSFET acts as a low-side switch for the load attached to the DC+ and DC- terminals. In other words, it sits between DC- and GND. DC+ is connected directly to VIN. Therefore, with a suitable power supply connected to VIN and GND, you can connect a DC load to DC+ and DC-, and control it from the Arduino. The circuit was designed with DC fans in a popcorn roaster in mind, but the driver is by no means limited to this type of load. Make sure that the load is PWM-compatible.

Important: The DC PWM driver does not include a flyback diode or similar protections. When using a brushed DC motor, you must take additional steps; otherwise current induced when the fan motor is stopped can damage the MOSFET. There is two options:

- Simplest approach: A flyback diode is a diode put across the motor's terminals, allowing current to flow from the negative to the positive lead. This gives the motor a path to discharge. Typically, a 1N4001 diode would be suitable for this purpose. This could be soldered directly onto the motor's leads; or screwed into the DC+- header. This is shown in figure 4 (a).

- Alternatively, a suitable Zener diode could be soldered across the MOSFET. Some revisions of the TC4+ board provide DO214AA-sized SMD solder pads for this purpose labelled D3, at the south east corner of the PCB. Alternatively a through-hole Zener diode could be screwed into the DC- and GND terminals. This is shown in figure 4 (b). Note that the Zener diode must be tuned to the parameters of the particular motor. A full discussion of this is beyond the scope of this document, however.

Important: In popcorn machines, powering the heating element but not the fan will lead to near-instantaneous overheating. See also section 3.2.1.

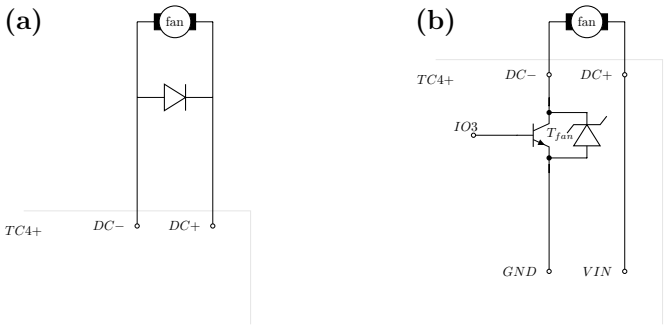


Figure 4: Two potential protection circuits for brushed DC motors. Left: A flyback diode across the motor. Right: A Zener diode across the MOSFET.

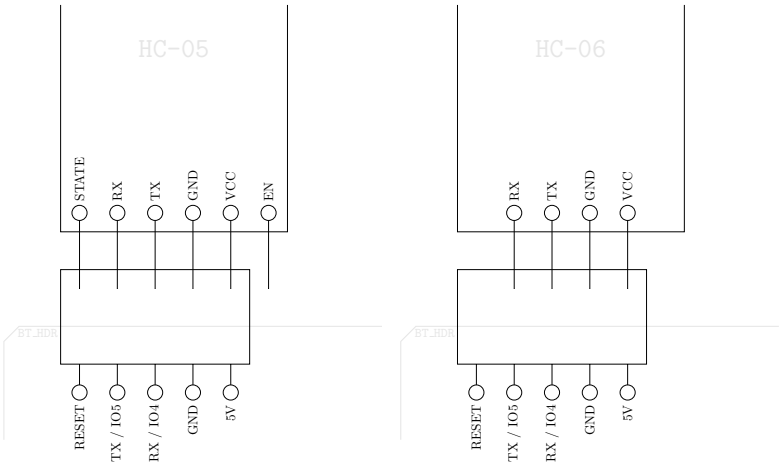


Figure 5: Connection of the HC-05 (left) and HC-06 (right) Bluetooth modules.

2.6 Bluetooth Header

The board provides a 5-pin header breaking out a reset circuit, TX voltage-divided to 3.3V, RX, GND and 5V. This allows for easy connection of a HC-05 or HC-06 Bluetooth module. With a HC-05 (but not HC-06), the module can optionally trigger the Arduino's reset circuit whenever a new Bluetooth connection is made. This effectively allows the Arduino IDE to communicate with the boards bootloader, enabling wireless uploading of sketches. For configuration purposes, the module can optionally be connected to to IO4 and IO5 pins instead of RX and TX, selectable via two jumpers. This allows the Arduino and TC4+ shield to act as a bridge between a host computer and the module's firmware. See the configuration section for details.

2.7 Application Example

The TC4+ board was designed to simplify control of popcorn machine based home coffee roasters. In such a scenario, the board would be connected as

follows.

- The INPUT (GND / VIN) screw terminal would connect to a DC power supply unit.
- the FAN (DC+ / DC-) would connect to the fan motor. A suitable flyback diode would be soldered directly to the motor terminals.
- Thermocouple channel 1 would connect to a thermocouple measuring the temperature of the incoming air (optioal).
- Thermocouple channel 2 would connect to a thermocouple measuring the temperature inside the roast chamber.
- The Bluetooth header would optionally connect to a Bluetooth module.
- The OT1 screw terminal would connect to a solid state relay (SSR), which in turn would control the AC live wire going to the heating element. A thermal fuse in series with the heater would be strongly recommended.

Figure 6 gives a diagram with all these connections.

2.8 Further Connectors, Jumpers & Status LEDs

Further to the connectors discussed in the application example, the TC4+ features the headers, jumpers and status LEDs listed below. Figure **??** shows the location of all headers, jumpers and LEDs on the TC4+ board.

- IO2, IO3 break out the Arduino's IO2 and IO3 ports and GND. Note that IO3 is also connected to the MOSFET. On newer versions of the board, a third pin carrying 5V has been added.
- I2C provides two 4-pin I2C headers for displays etc.
- Arduino stackable headers provide all Arduino pins. Note that IO2, IO3, IO9 and IO10 are in use for IO2, IO3, OT1, OT2, respectively. IO4 and IO5 may be in use for the Bluetooth module.
- BT_SEL and BT_RST jumpers for configuring Bluetooth connection and optional Bluetooth reset circuit.
- Status LEDs for Power On, OT1 and OT2.

2.9 Compatibility with TC4 Arduino sketches

The thermocouple ADC and on-board temperature sensor are implemented using the same components (MCP 3424 and 9800) as are used on the TC4. Additionally, a 24LC512 512kbit EEPROM is included on the TC4+ board for compatibility with some TC4 sketches that rely on this. For instance, aArtisanQ_PID in "standalone" mode uses the EEPROM to store roast profiles. SSR drivers are connected to pins IO9 and IO10; and the DC fan driver to pin IO3. This is also the same pinout as on TC4 boards. Any TC4-based Arduino sketches will therefore work with the TC4+ board.

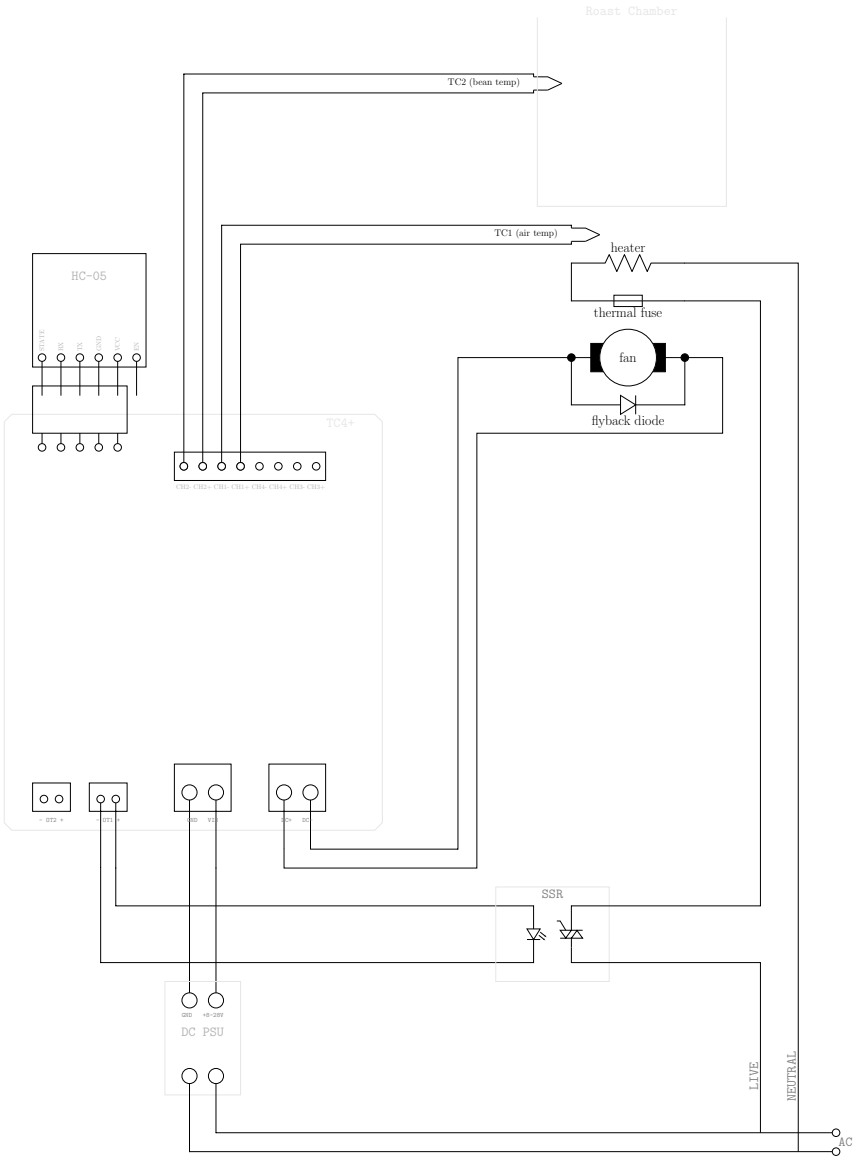


Figure 6: A typical application example of the TC4+ board. This shows the TC4+ board connected to a DC PSU; a fan with flyback diode; a SSR controlling a heater (with thermal fuse); a HC-05 Bluetooth module; and two thermocouples measuring bean and air temperature.

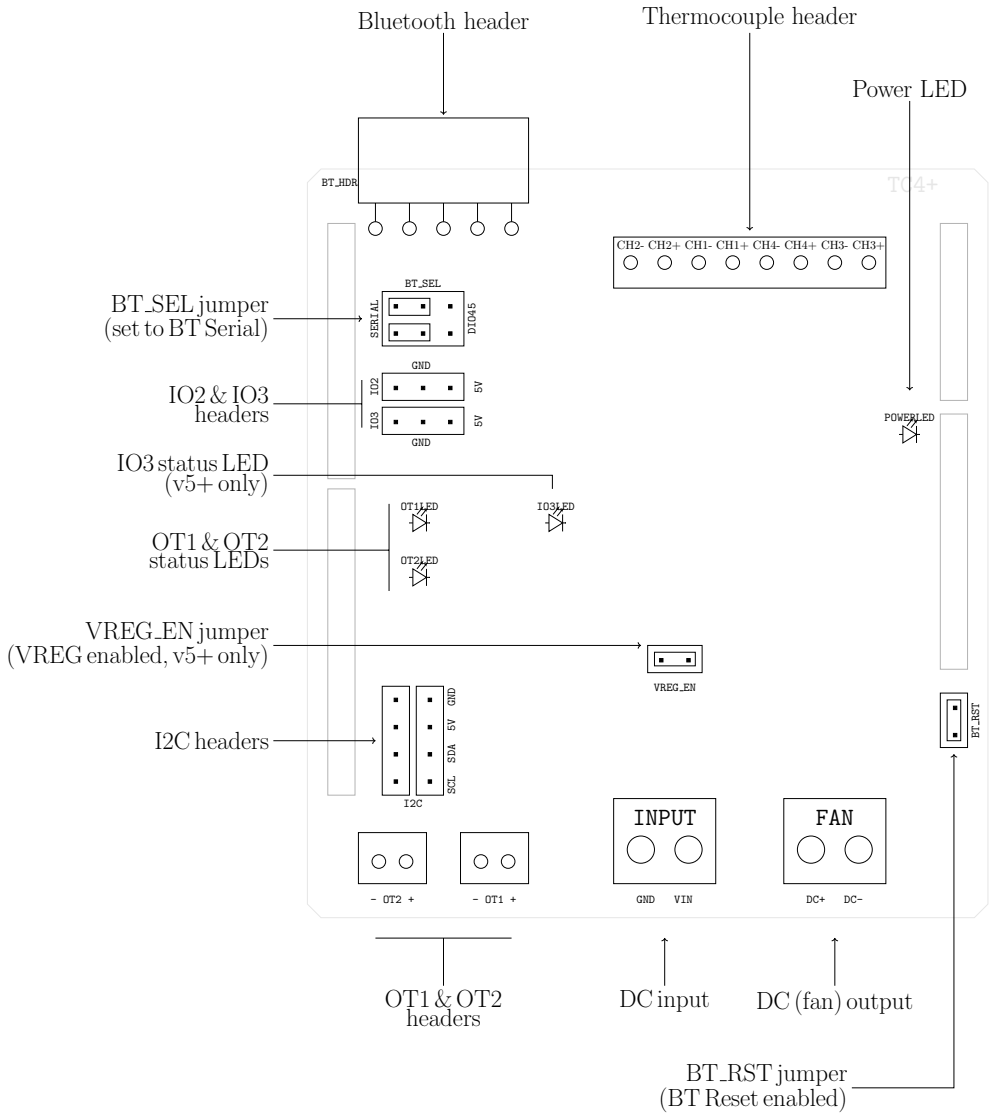


Figure 7: Layout of all connectors, jumpers and status LEDs on the TC4+ board.

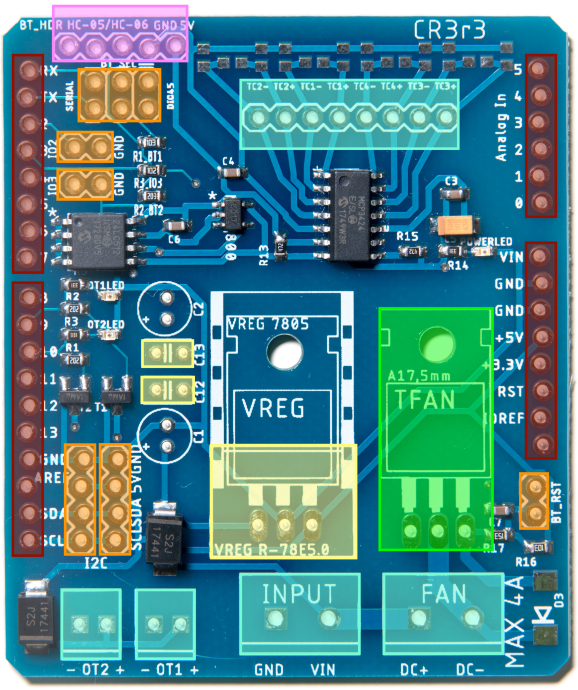


Figure 8: Layout of TC4+ through-hole components (v4 and earlier).

2.10 Known Issues

On the current PCB design the labelling of the thermocouple channels is hidden by the screw terminal when the board is fully assembled. The labelling will be moved further afar in a future revision. Note that from left to right the pins are Ch2- Ch2+ Ch1- Ch1+ Ch4- Ch4+ Ch3- Ch3+.

3 Assembly & Configuration

3.1 Assembly

If you purchased a TC4+ KIT, it will come with all surface-mount devices soldered, and loose through-hole components which you will have to solder yourself. Figure 8 and Figure 9 show the location of these components in different colours:

- Dark Red: Stackable headers.
- Orange: Male headers; break apart as necessary. [1]
- Orange: Male header 2x1, VREG_EN, near centre of board, not shown on photograph. On v5 and higher only.
- Pink: Angled female header.[2]

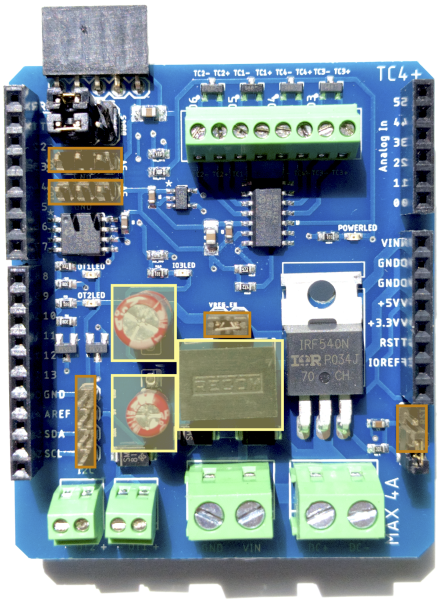


Figure 9: Layout of TC4+ v5 through-hole components. Pin headers and jumpers (marked in orange) and capacitors for voltage regulator (marked yellow) are slightly different than in earlier versions.

- Blue/Turquoise: Screw terminals. [2]
- Yellow, labelled VREG: Voltage regulator. The kit comes with a R-78E5.0 switching regulator.
- Yellow, C12/C13:
  - On version 4 and earlier: 10uf ceramic capacitors. C1/C2 are not used with the R-78E5.0 switching regulator. On newer versions of the board, C12/C13 have an additional third soldering hole to more easily accommodate capacitors with wide leg spacing. You can use the outer two holes.
  - On version 5: Electrolytic capacitors. C12 is 10nF, C13 is 22nF. **These are polarized:** The positive legs need to go to the two outermost holes. The negative legs should be closer together, and can be either the middler or innermost holes, whichever are easier.
- Green, labelled TFAN: IRF540 transistor. Optionally with heatsink, and fixed in place with M3 bolt.

Notes: [1] BT_SEL (2x3, top left) and BT_RST (1x2, bottom right) are for jumpers, so male headers are required. For the I2C, IO2 and IO3 headers, a different connector could be used if preferred.

[2] Included connectors are suggestions – any other 2.54mm / 5.08mm spaced header could be used. For the Bluetooth board, an angled female 4/5 pin header will have the Bluetooth module horizontal next to the TC4+ shield. For vertical BT module, use a straight female 4/5 pin header.

3.2 Software

The TC4+ board is fully hardware-compatible with the TC4 board, and any Arduino sketch meant for the TC4 board will work with the TC4+. These include aArtisan and aArtisanQ_PID. A variety of application sketches can be found on `https://github.com/greencardigan/TC4-shield`.

3.2.1 Overheating Safety Measures

For popcorn machine roasters: It is recommended to make sure that the heating element cannot be turned on without a minimal fan level. (This would lead to near-instantaneous overheating, and at best a blown thermal fuse.) Some Arduino sketches might provide this as a feature out of the box. In others, a check can easily be added. One such option is listed in figure 10, where a brief `if`-statement is added at the end of the Arduino sketch's main loop. This has been verified to work with aArtisan 3.10. Make sure that you use a sufficient fan level in this check.

Note that this protects only against one very specific issue, not overheating in general. Additional checks could be performed based on thermocouple readings in a similar manner, shutting off heater output above a threshold temperature. Do not in any case rely on software measures for safety.

aArtisan.ino

```
[...]
void loop()
{
   [...]
   if (levelOT1 !=0 && levelIO3<=50){
      levelIO3 = 50;
      float pow = 2.55 * levelIO3;
      analogWrite( IO3, round( pow ) );
   }
}
```

Figure 10: A brief check to ensure the heating element is never turned on without a minimum 50% fan duty cycle. If heater output (OT1) is non-zero and fan level (IO3) is less than 50, set the fan to 50.

3.3 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. (For HC-05 only. See below for configuration.)

Figure 11 shows these options.

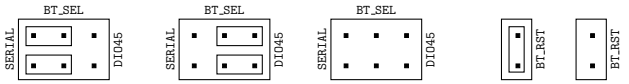


Figure 11: Jumper configuration for Bluetooth serial. From left to right:

- 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

3.4 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 12) 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.)

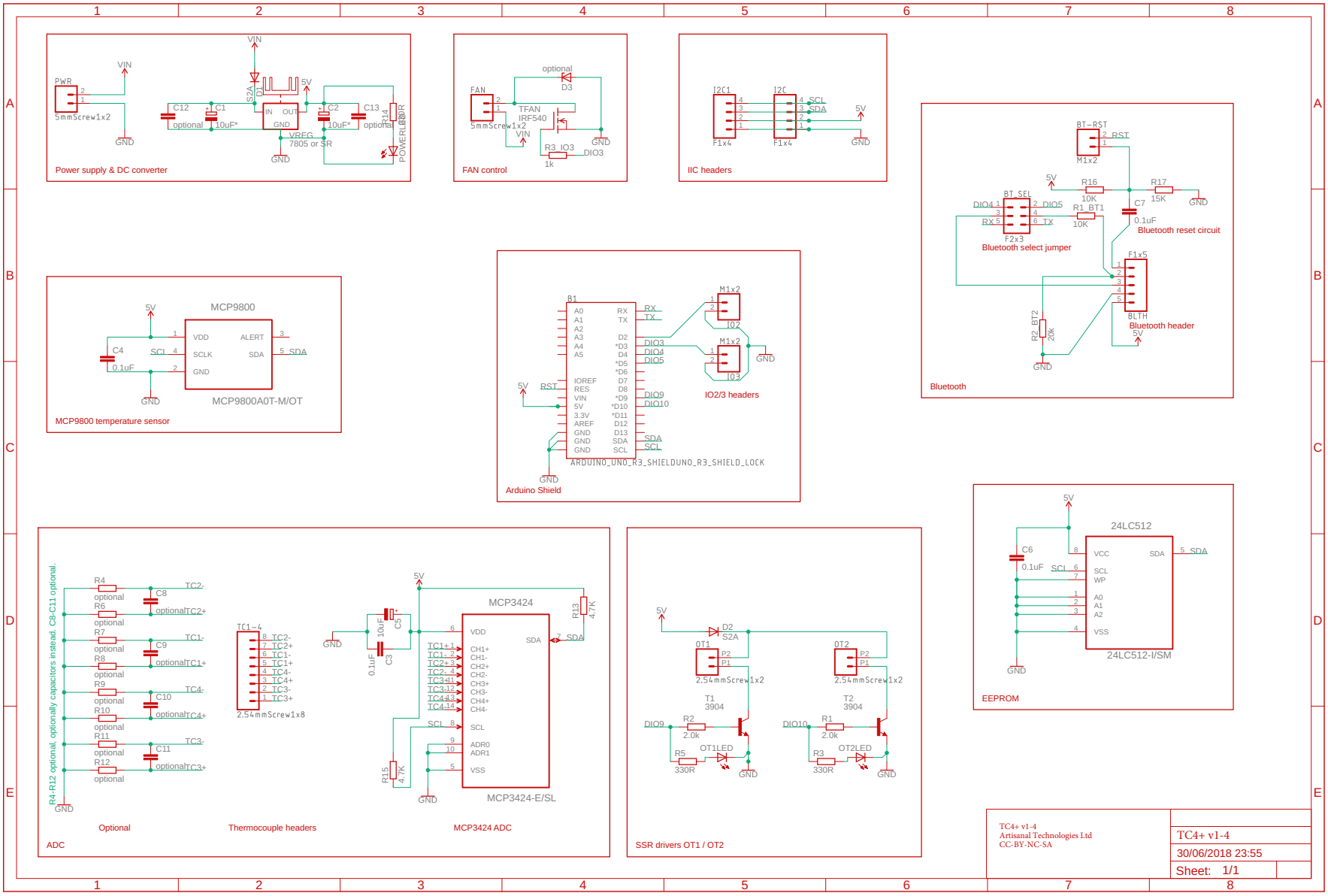
3. That's it! Return the BT jumpers to the left / SERIAL position, and short the BT_RESET jumper if desired. Flash whatever Arduino sketch you want to use with the board. Scan for Bluetooth devices on your computer, and connect to the BT module. (Note that it might still show up as HC-05 respectively 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.

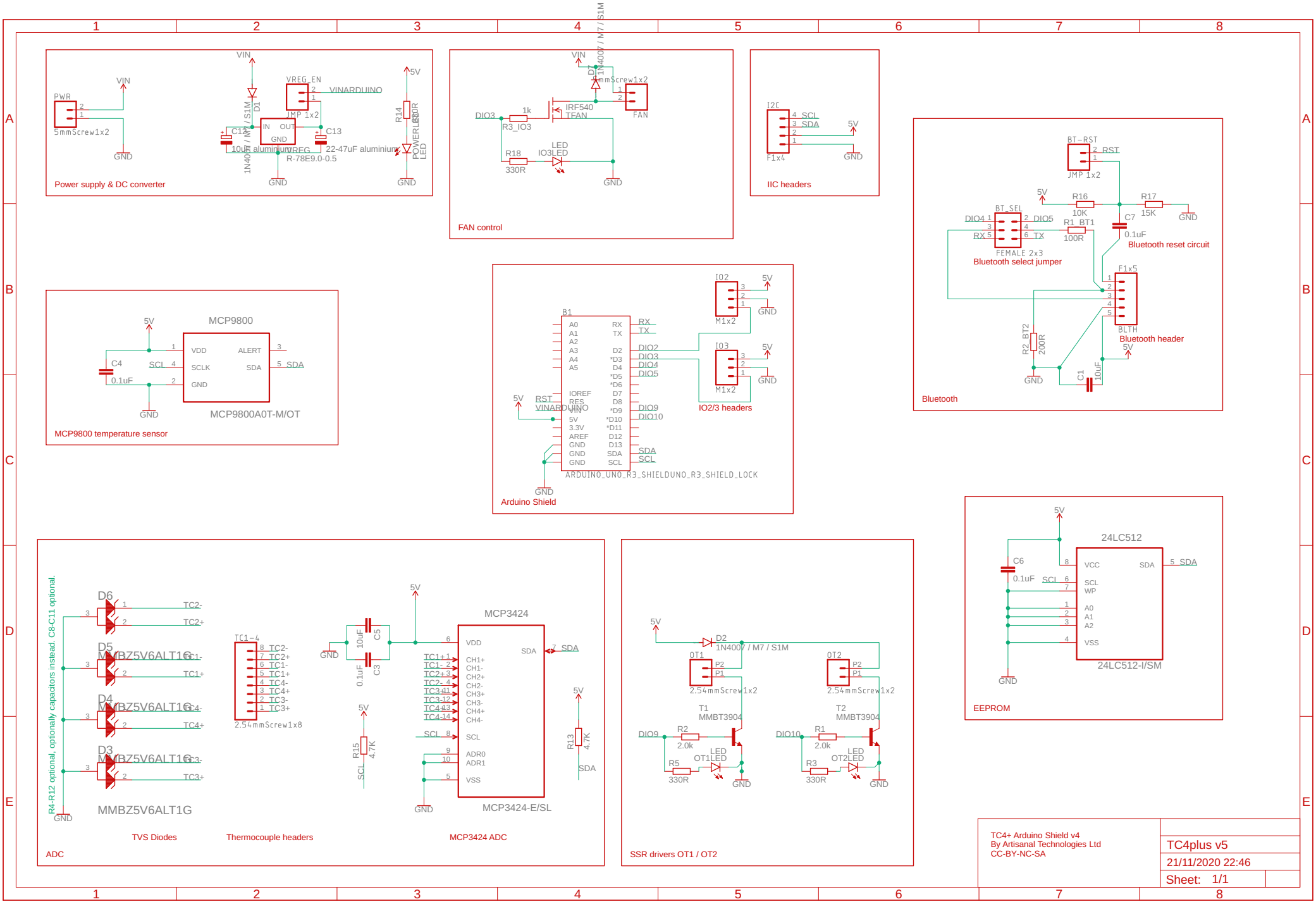
4. Troubleshooting: If it's not working, check that you are in range. If it's not that, then in Windows you could try turning off and on the BT serial port profile in device manager. If that doesn't help, unpair the module and pair it again.

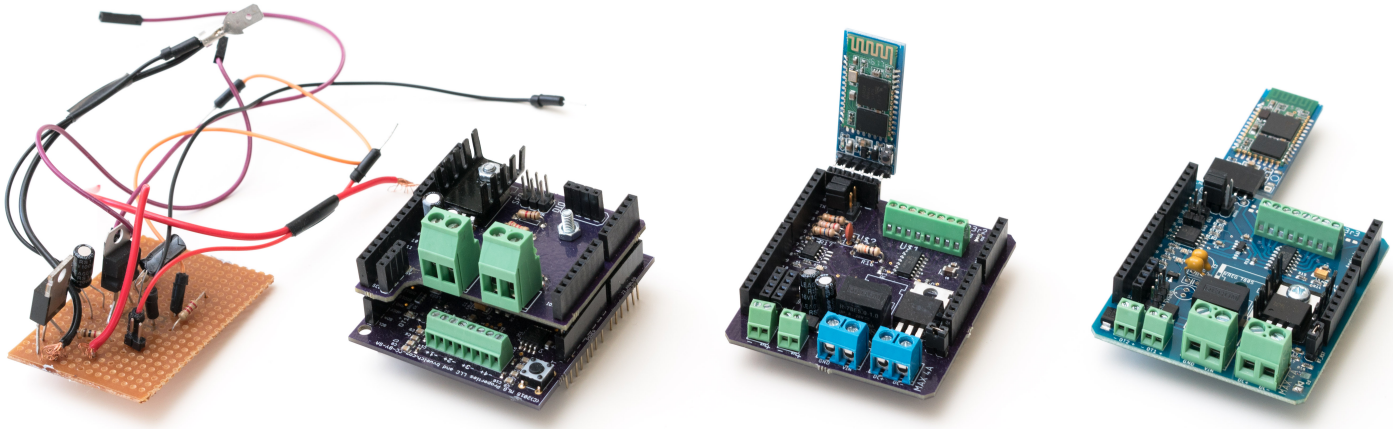
bluetoothATmode.ino

```
#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 12: The sketch used to configure the Bluetooth module.


Power supply & DC converter
FAN control
IIC headers
Bluetooth
Bluetooth select jumper
Bluetooth reset circuit
Bluetooth header
MCP9800 temperature sensor
MCP9800A0T-M/OT
Arduino Shield
ARDUINO_UNO_R3_SHIELDUNO_R3_SHIELD_LOCK
IO2/3 headers
EEPROM
24LC512
24LC512-I/SM
ADC
Optional
Thermocouple headers
MCP3424 ADC
MCP3424-E/SL
SSR drivers OT1 / OT2
R4-R12 optional, optionally capacitors instead. C8-C11 optional.
TC4+ v1-4
Artisanal Technologies Ltd
CC-BY-NC-SA
TC4+ v1-4
30/06/2018 23:55
Sheet: 1/1


Power supply & DC converter
FAN control
IIC headers
MCP9800 temperature sensor
Arduino Shield
IO2/3 headers
Bluetooth
Bluetooth select jumper
Bluetooth reset circuit
Bluetooth header
EEPROM
ADC
TVS Diodes
Thermocouple headers
MCP3424 ADC
SSR drivers OT1 / OT2
TC4+ Arduino Shield v4
By Artisanal Technologies Ltd
CC-BY-NC-SA
TC4plus v5
21/11/2020 22:46
Sheet: 1/1



Bonus picture: The evolution of the TC4+ board. From left: Remnants of my very first setup with a stripboard holding a MOSFET and a 7805 voltage regulator; A printed circuit board stacked on top of the TC4, with MOSFET, voltage regulator and bluetooth header, all through-hole components; An early prototype version of the TC4+; The current revision of the TC4+ with further tweaks and improvements.