The readme file of the Footwarmer

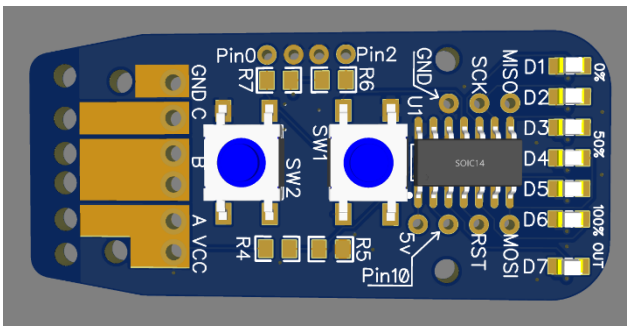
Safety first

Do not use the footwarmer unless you are able to swiftly disconnect the power or unplug the soles in case they get too warm.

General description

The footwarmer is a contraption for controlling the power in electrically heated soles. The power to the soles is pulse modulated and the longer pulse, the more power to the soles.

The power can be set in 5 steps of 20% in each step and 6 LED´s are indicating the current power level. To step up and down one of the buttons needs to be pressed for a certain time (default 600ms). Pressing the two buttons simultaneously will give a rough indication of the voltage of the power source.



In normal mode:
Led 1, green: indicates power to the board.
Led 2 and 3, green: indicates 20 and 40% power
Led 4, yellow: indicates 60% power
Led 5 and 6, red: indicates 80% and 100% power
Led 7, blue: indicated when power is delivered to the soles.

In voltage check mode:
LED 1, green: ≥ 90%
LED 2, green: ≥ 75 and < 90%
LED 3, green: ≥ 60 and < 75%
LED 4, yellow: ≥ 45 and < 60%
LED 5, red: ≥ 30 and < 45%
LED 6, red: < 30%
LED 7, blue: not used

The voltage thresholds are variable set in the “void setup()”-section. Example of 12v lead acid

```
if (BatteryType == 6) {    // 12v Lead acid
  ShuntFactor = 38;        // Reducing the max power to aprox 10W
  VoltThr1 = 11.75;        // 30%
  VoltThr2 = 11.98;        // 45%
  VoltThr3 = 12.2;         // 60%
  VoltThr4 = 12.37;        // 75%
  VoltThr5 = 12.50;        // 90%
```

The “ShuntFactor” is a power adjustment used if soles are manufactured for an operating voltage of approx. 3 to 5 volts shall be fed with higher voltage. The shut factor will set the maximum length of the pulse width modulation. Experience shows that 10 W combined is a reasonable power setting.

Examples:

The soles combined resistance	Max power @8 volts (2S LiPo)	Max power @12 volts (3S LiPo or 12 lead/acid)	Max power @12 volts ShuntF. = 42% (3S LiPo or 12 lead acid)	Max power @13 volts (3S LiPo or 12 lead/acid)	Max power @12 volts ShuntF. = 35% (3S LiPo or 12 lead acid)
6	10.6 W	24.0 W	10.0 W	28.2 W	9.8 W
8	8.0 W	18.0 W	7.6 W	21.1 W	7.4 W
12	5.3 W	12.0 W	5.0 W	14.1 W	4.9 W

Watch dog

In the worst-case scenario, the microprocessor enters the loop with the output set high which is powering the soles continuously with power.

This is prevented in two ways, firstly the code is written without any loops or while statements to enter in the first place. Secondly, a watch dog is implemented in the code and if the watch dog is not set to zero within 500ms the watchdog timer will reset the microprocessor and the output will be set to zero.

Inhibited start up

If a button is pressed during boot the board will enter an inhibited state and indicate it with flashing red LED´s. This is to ensure that the power is set to 100% by mistake.

The board will be reset by the watchdog automatically so just make sure the buttons are unpressed.

Low battery interlock

In case the incoming voltage drops below 30% the board will shut down and enter an interlock and shut down the output.

The interlock can be reset by pressing the both buttons at the same time and check the voltage.

It´s important to set the correct type of battery in order for the board to shut down at the correct voltage level.

Before first use

1. Set the correct type of battery

Download the latest source code from GitHub. Configure the following parameters according to your application:

```
// Confugurable parameters for adjustment for each application
int BatteryType = 6;     // 1 = 2S LiPo/LiIon, 2 = 3S LiPo/LiIon, 3 = 2S LiFePo4, 4 = 3S LiFePo4, 5 = 4S LiFePo4, 6 = Lead 12v
int ButtonTime = 600;    // The time in ms for a button to be pressed before notching up/down
```

2. Upload the source code

Upload the source code to the microprocessor on the board according to the instructions in the following chapters.

3. Test without load

Connect the board to the soles and a power source. A fuse of 3 ampere is recommended. Vcc = positive power, GND = Ground.

Check if the buttons are working and the different power settings are able to set. Do also check the voltage indication and verify that the indicated voltage is reasonable.

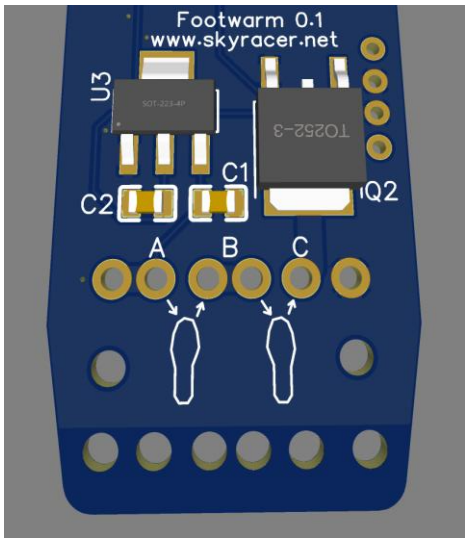
Note: the board can get damaged if the polarity is reversed.

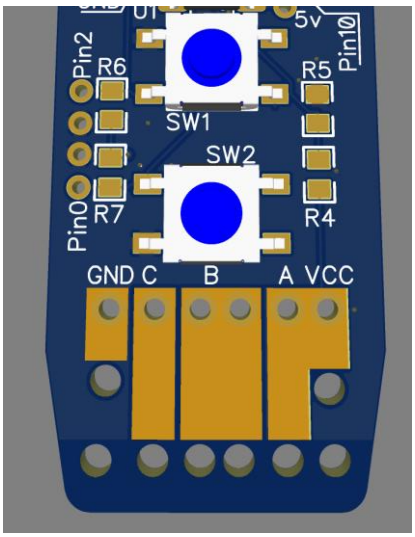
4. Connecting the soles

Depending on the resistance and the operating voltage of the soles and the voltage level of the power source, the soles can either be connected in serial or in parallel. If in doubt, please start by connecting them in serial.

Serial connection: Connect one sole between A and B and the other between B and C.

Parallel connection: Connect both soles between A and C.





Under side *Top side*

5. Start with a low power setting and check the amount of heat the soles are generating. Gradually Test the grade of heat in the soles

Start with a low power setting and check the amount of heat the soles are generating. Gradually increase the power level and verify that the soles don't get burning warm at the maximum level. I recommend to sit on chair and use the soles as a seat pad.

If the power is too high or too low at maximum level, please adjust the "ShuntFactor" accordingly and upload the source code again. Please note that the "ShuntFactor" is set for each type of battery type.

Sources

An old video referring to an obsolete version of ATTinyCore, but the current version works so it is applicable anyway:

https://www.youtube.com/watch?v=TyJQtaTvj3Q

ATTinyCore on GitHub:

https://github.com/SpenceKonde/ATTinyCore

Programming the board using Sketch

In order to program the board an Arduino Mega 2650 was used as used as a programmer gateway. Other Arduino boards can be used as well, but then the pin numbers in ArduinoISP.ino needs to be adjusted.

Note: you can get the ArduinoISP.ino from the examples within Sketch, but some adjustments were made hence the file presence of the file on GitHud.

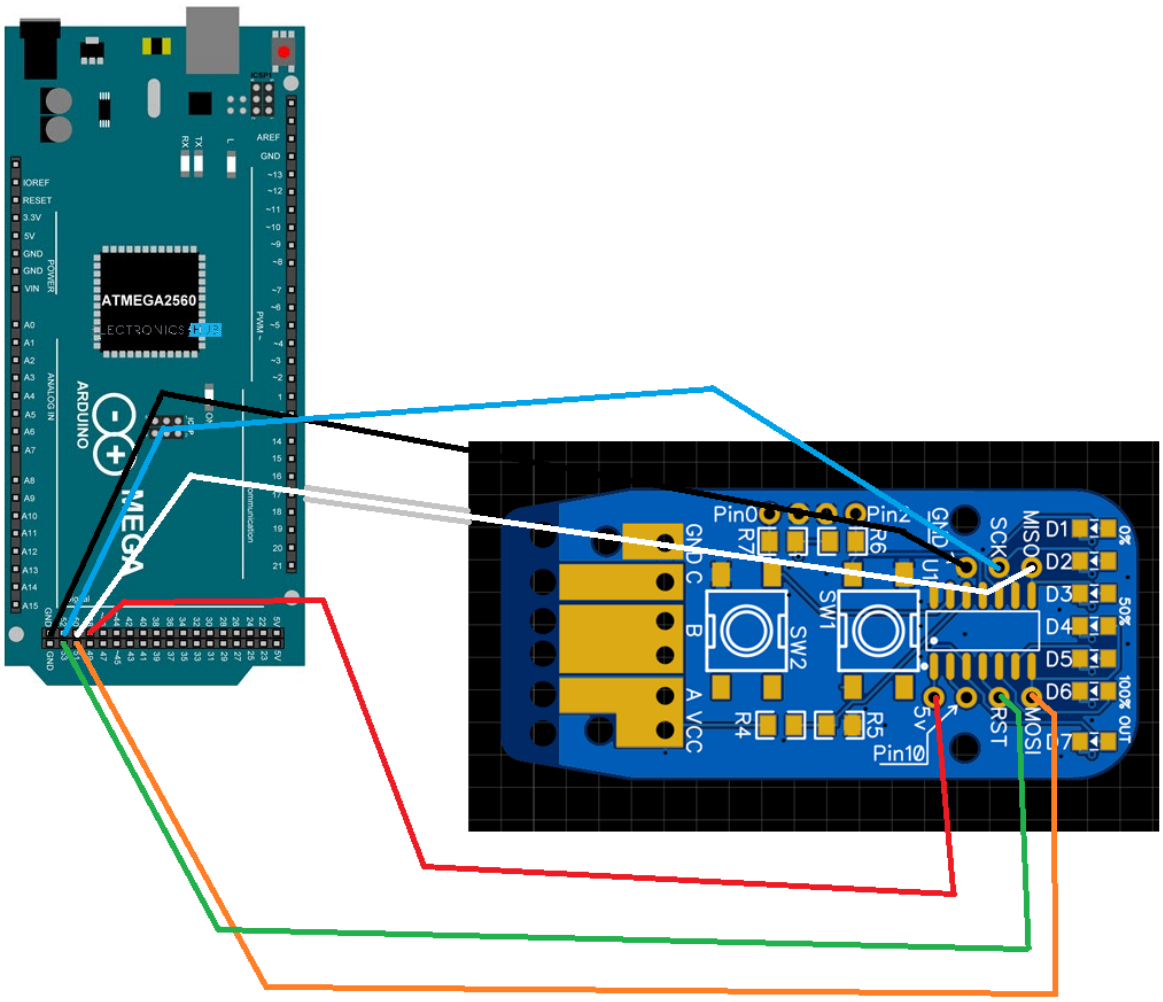
First, upload the ArduinoISP.ino (https://github.com/speedbird620/footwarm/blob/main/ArduinoISP.ino) to the Arduino Mega 2650 board. Use these settings:

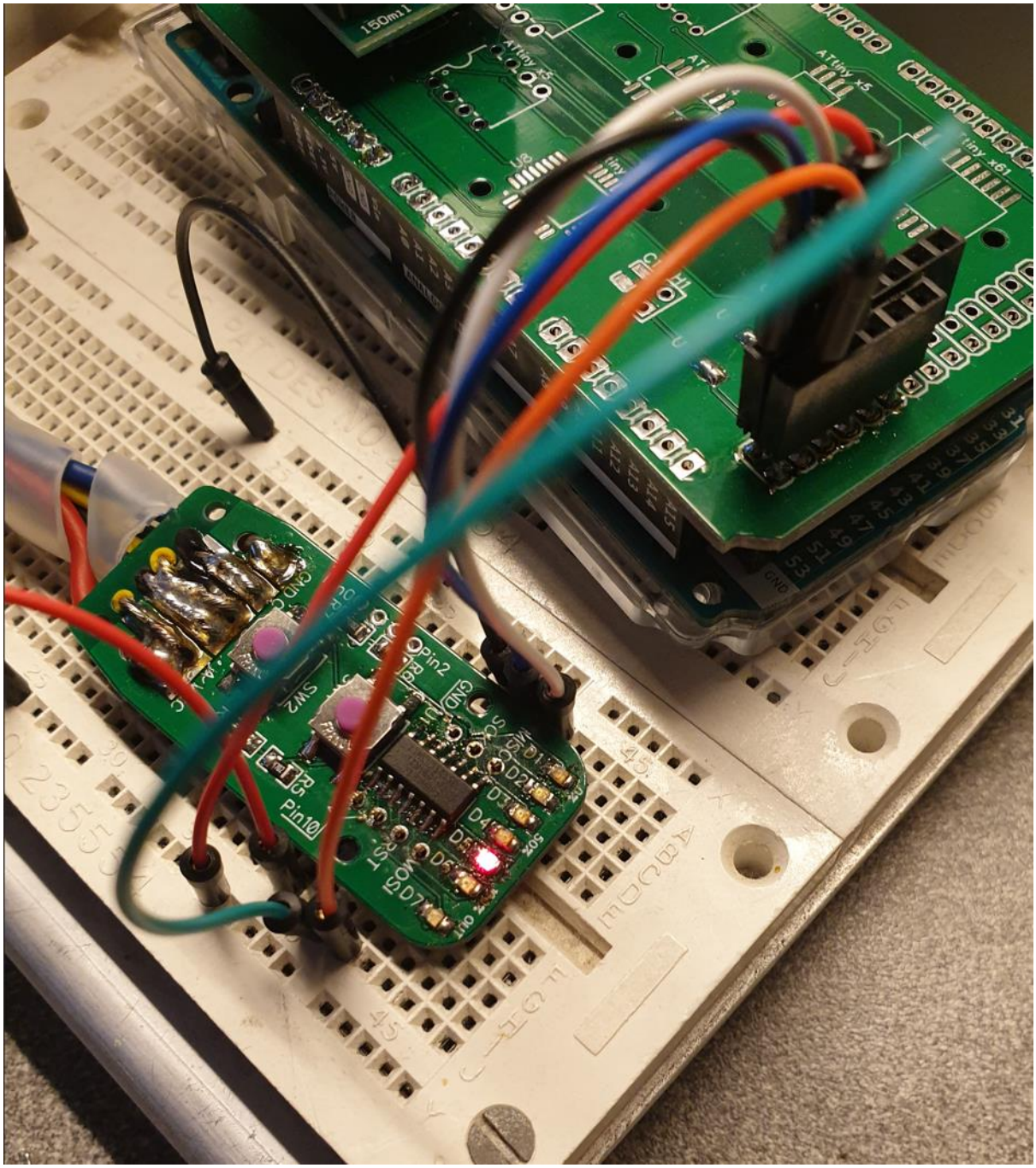
Please note that the com port the Arduino is mounted to is specific for each computer. Look in the device manager if unsure.

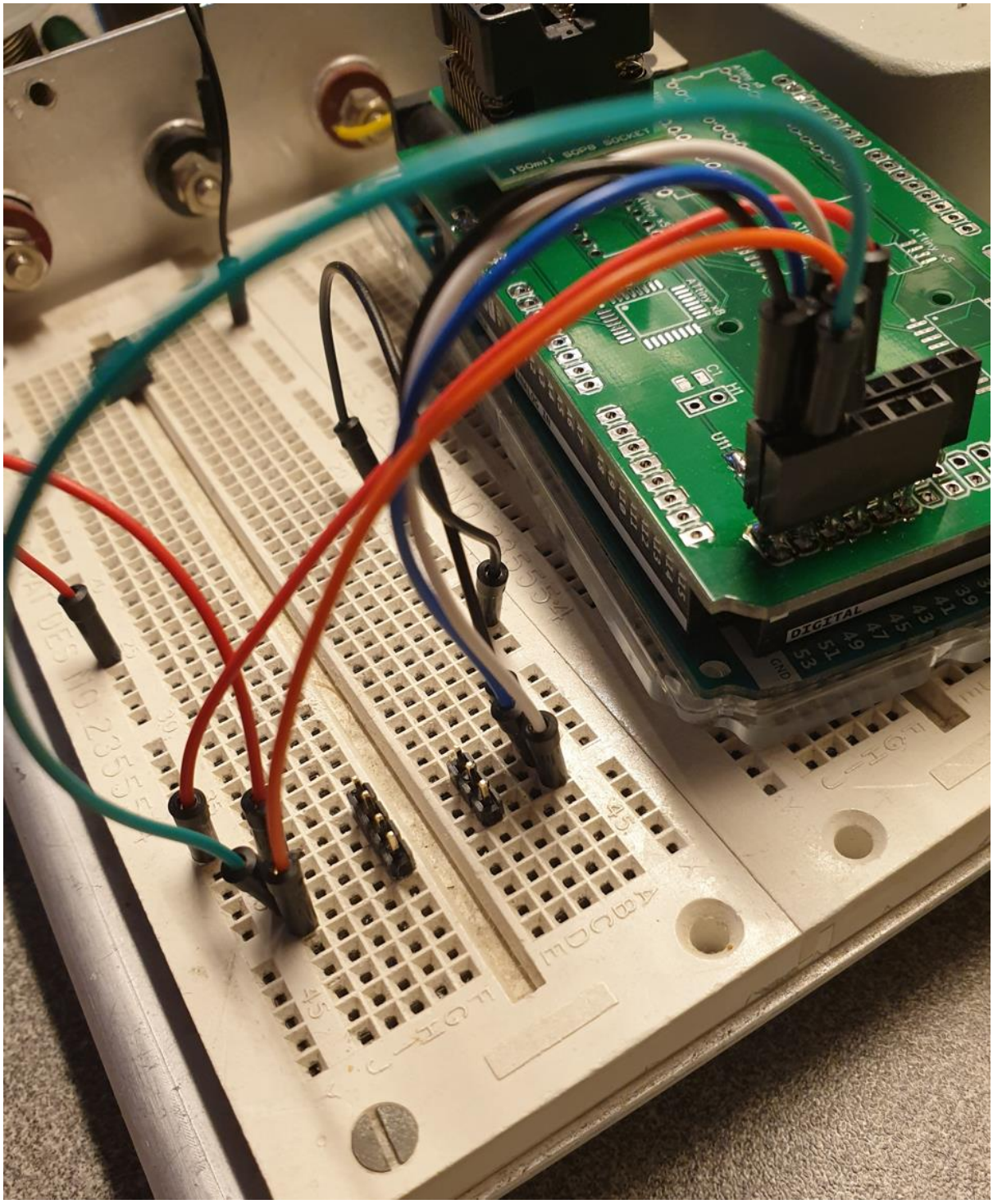
Then connect the board to the Arduino Mega 2650:

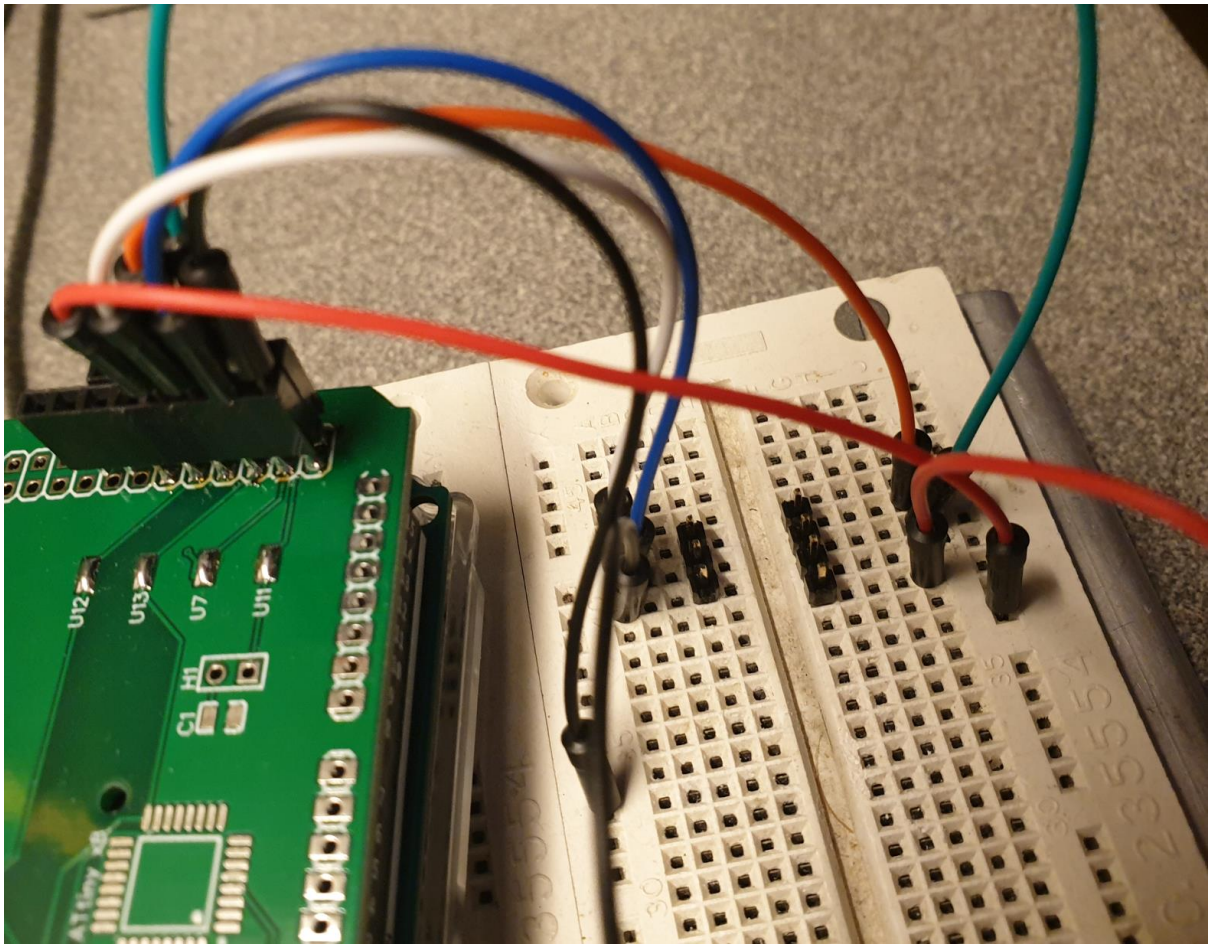
Arduino Mega 2650 pin number	Footwarm pin name	Function
48	5v	Power to the Footwarmer
50	MISO	Master In Slave Out
51	MOSI	Master Out Slave In
52	SCK	Sync
53	RST	Reset
GND	GND	Ground

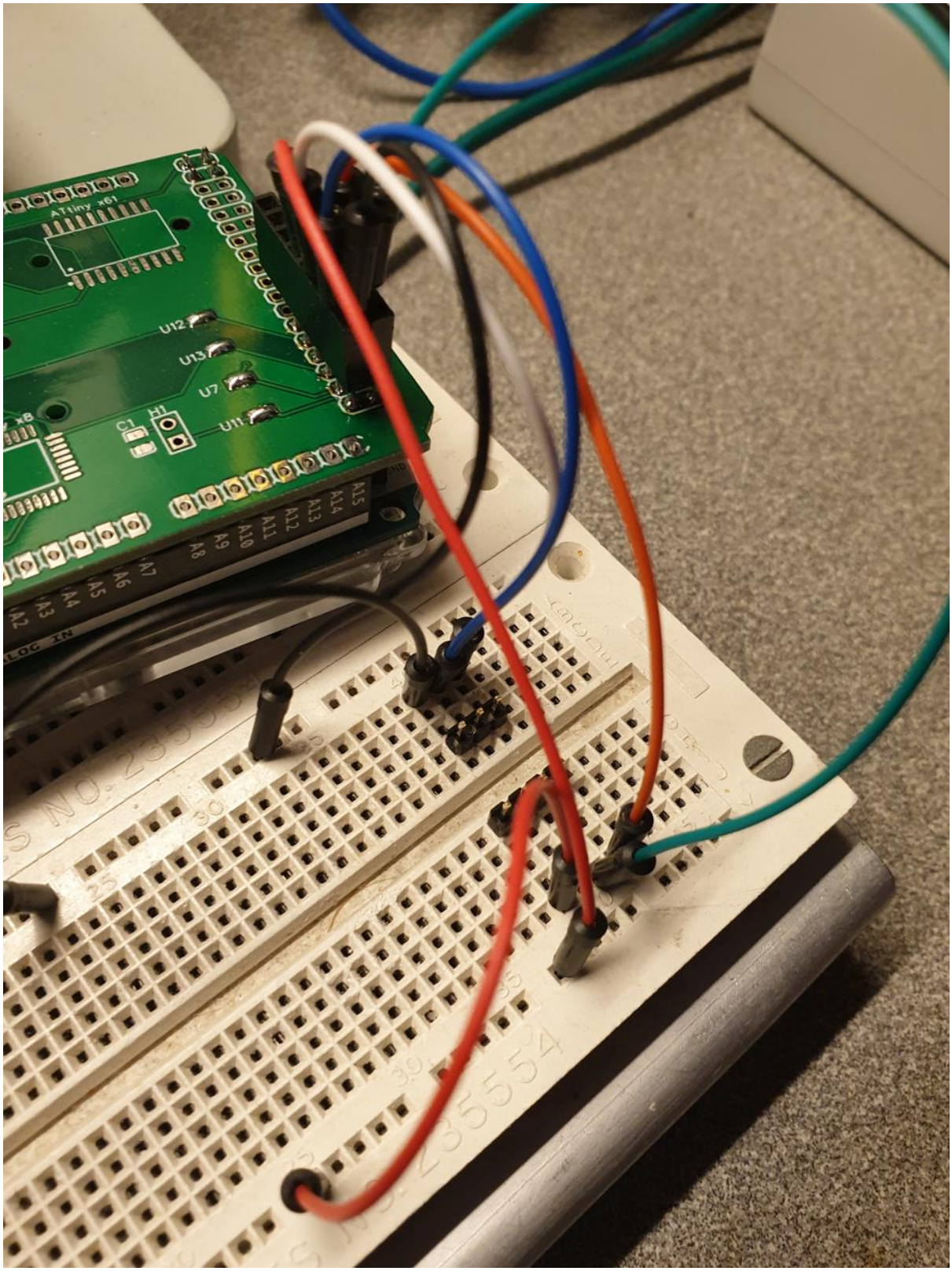
In some tutorials they are mentioning a capacitor on the reset pin, but I never got it to work so I just left it out.



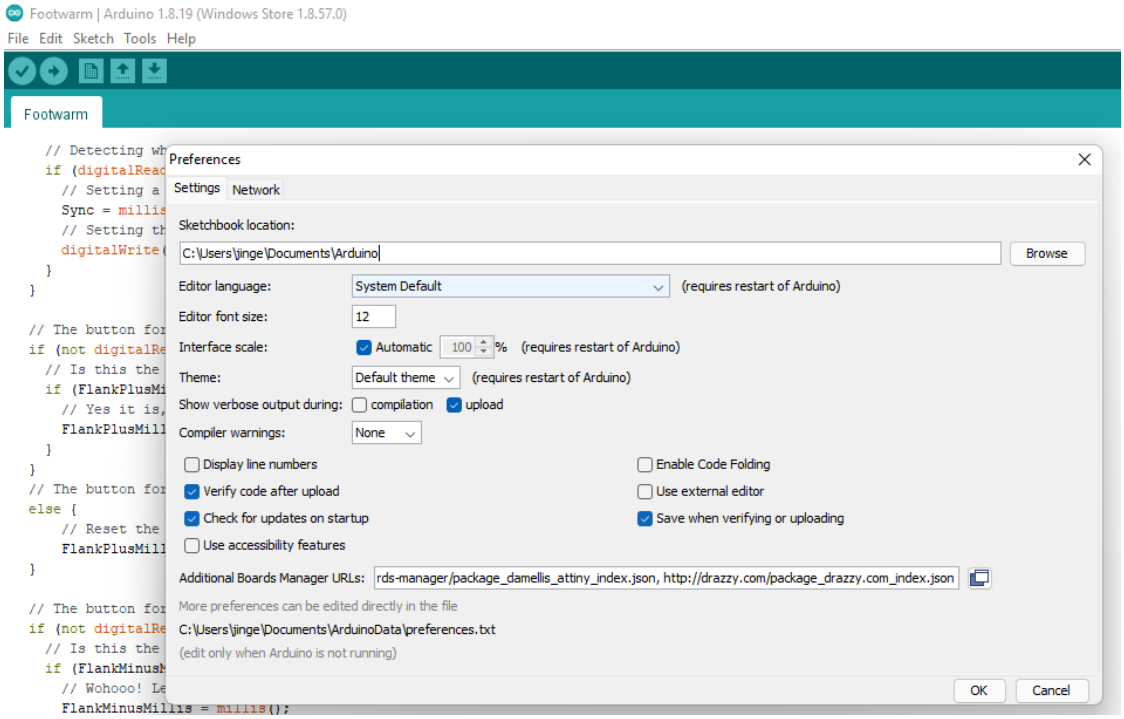

150ml
ATtiny x5
ATtiny x5
tiny x61
GND
Pin10
D1
D2
D3
D4
D5
D6
D7

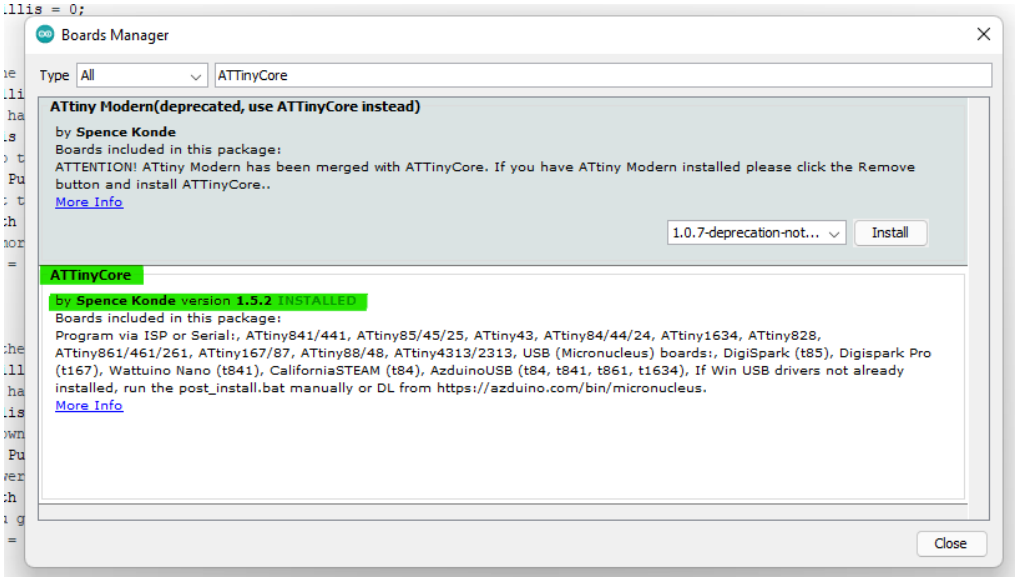

DIGITAL

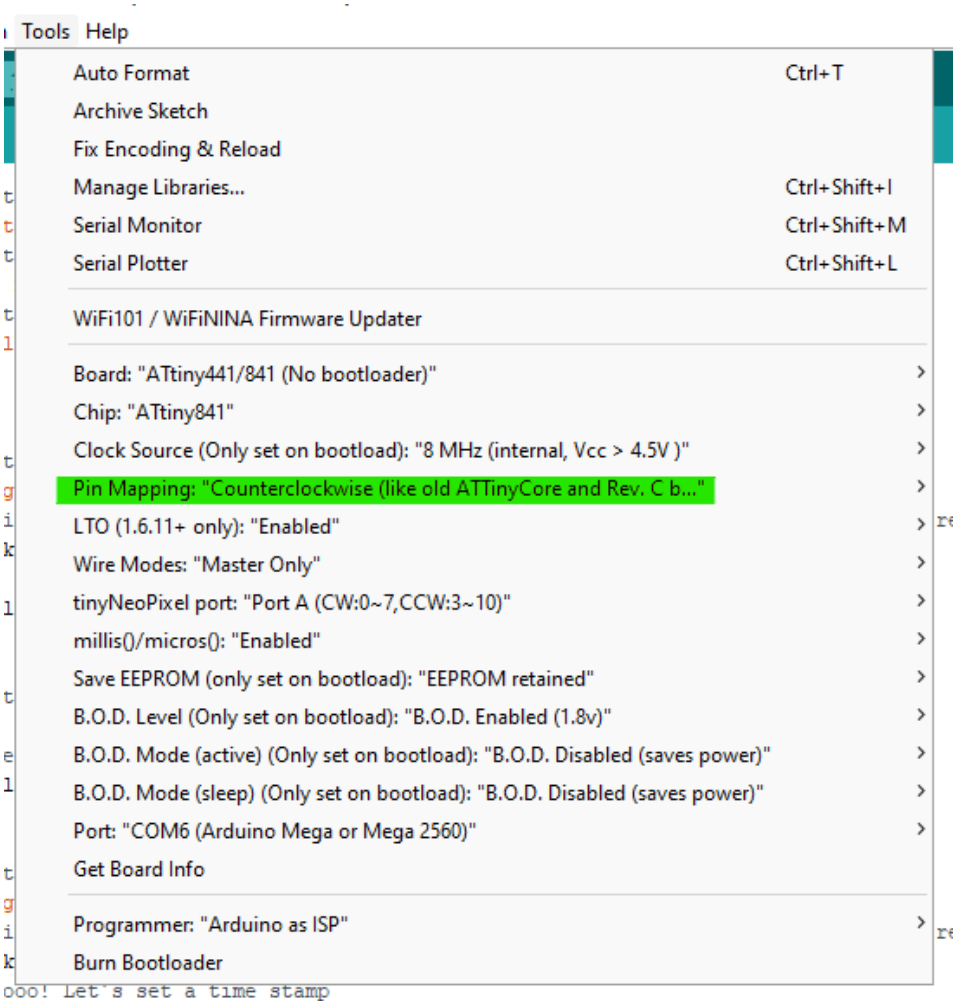

U12
U13
U7
U11
H1
C1


ATtiny x61
U12
U13
U7
U11
C1
H1
A15
A14
A13
A12
A11
A10
A9
A8
A7
A6
A5
A4
A3
A2

Proved settings used in sketch:







All default choices except pin mapping marked in green.

First, choose the “Burn bootloader” in order to get the settings right in the ATTiny. Then compile  and upload  and it should work.

If not, check settings and connections. Still not lucky? Use Google, it usually helps. The most common fault will be added in the Q&A below.

Q&A:

Q: What´s the meaning of life?
A: 42