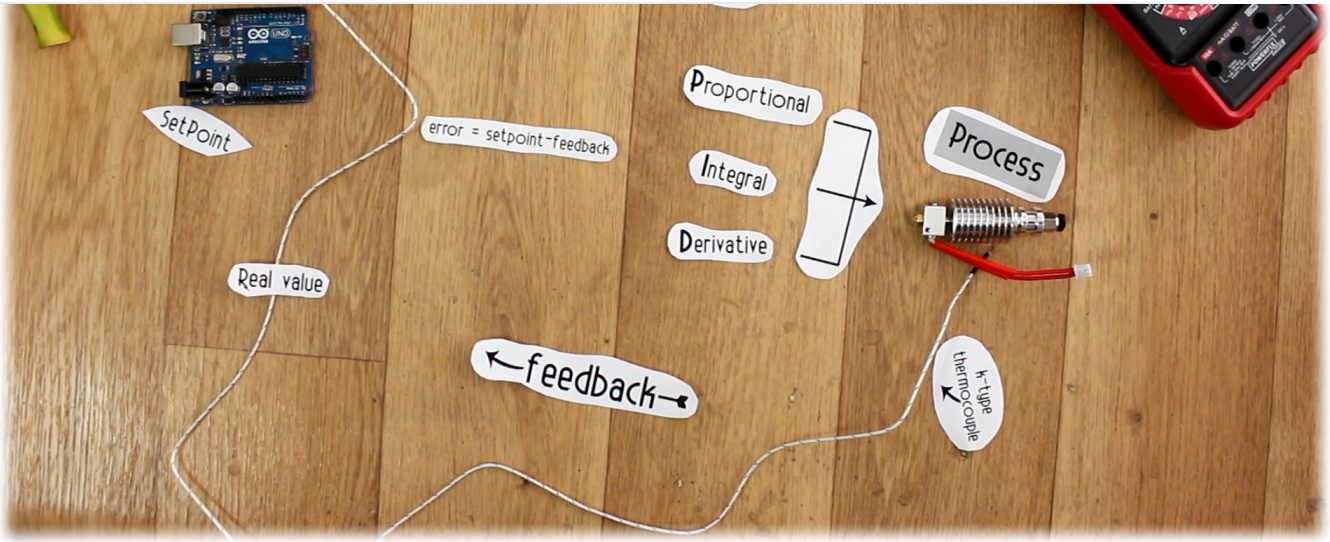
(https://www.youtube.com/user/ELECTRONOOBS) (https://www.instagram.com/electronoobs/) (https://www.facebook.com/ELECTRONOOBS/) (https://twitter.com/ELECTRONOOBS)

Temperature PID controller - Arduino

What's up my friends, welcome back. What I want, is the aluminum block below to have let's say, exactly 100 degrees. I'll control the real temperature using a K type thermocouple. To read the data I'll use the MAX66 75 breakout module and control the PID algorithm with and Arduino. Finally, to apply power we will make a small circuit using a MOSFET or maybe a TRIAC in case of high AC voltages. This will be a close loop. The thermocouple measures the real values, the Arduino creates the signal applied to the MOSFET and this transistor will control the power of a heating element inside of the aluminum block and once again the thermocouple will measure the value, that's why it's a close loop.



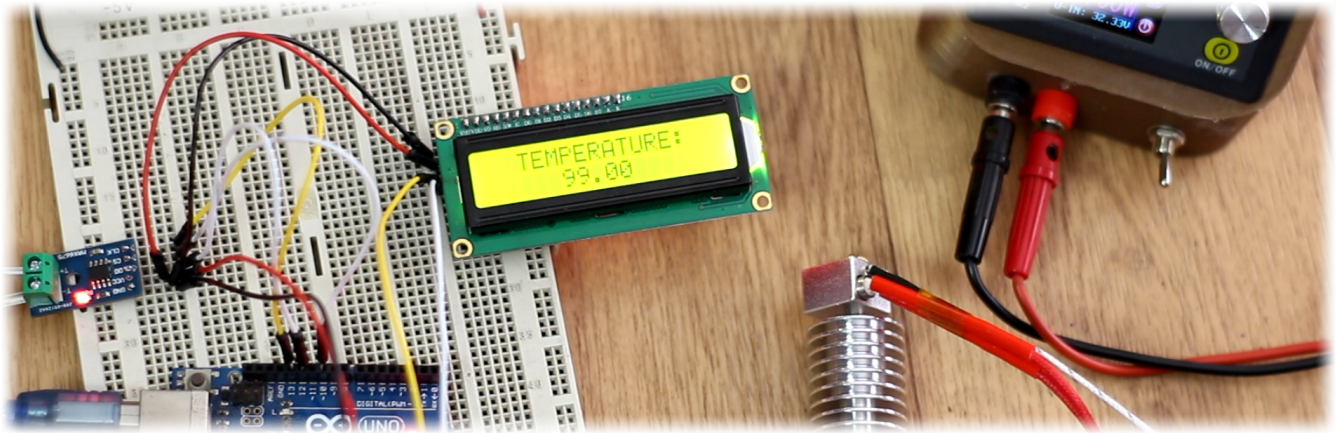
See the full part list here: GO TO IT (eng_arduino_tut24_parts1.php)

PART 1 How PID works

So, the first thing is to understand how a PID temperature controller will work. As in any PID system we need to define a **final process**, in our case will be the final temperature that we will achieve. In order to control this temperature, we will need a **feedback**. So,

temperature. The system will make the difference between the desired value and the feedback from the output (**error**), and using 3 constants, the **proportional**, the **integral** and **derivative**, we can change the output according to the feedback.

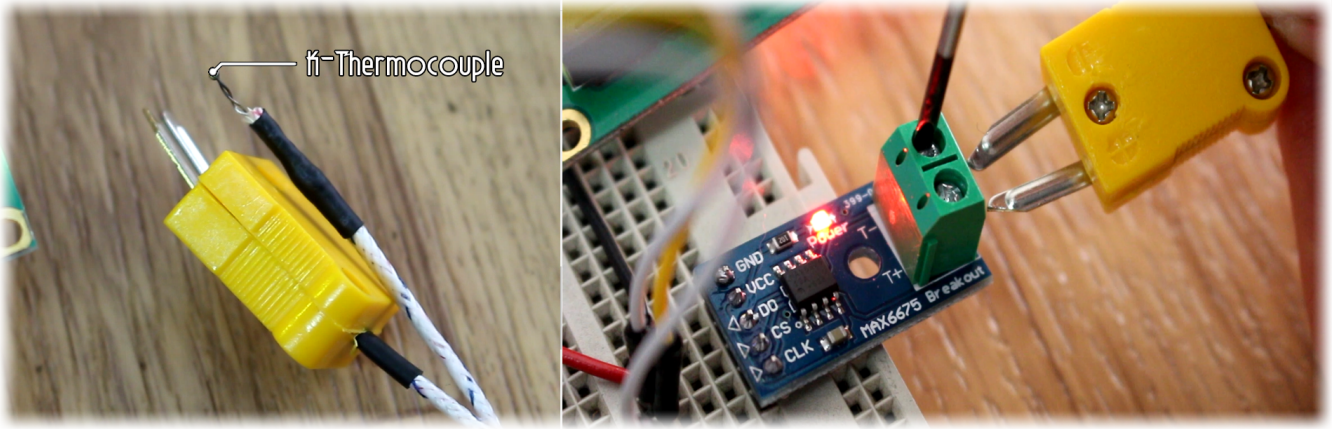
So, if we want the heating block to have exactly 100 degrees what we will do first is apply power to it. This will start heating up. By the time it reaches the setpoint value, which in this case is 100 degrees, the feedback will inform that to the PID control and this will start lowering the power applied to the heating element, and in our example that will be made using a PWM signal applied to a MOSFET that will control the voltage that goes to the heating element inside the block.



If we do just that, that is called P control, or **proportional** control, and will end up in a temperature oscillation between certain values and it will very difficult or never be stable. For that we add the D control, or **derivative**. This kind of control will react to the speed of temperature change. So, if, for example, if we blow air on the aluminum block, the derivative will fast push the power to maximum in order to keep the temperature. Finally, we have the I, or **integral**. This will sum the error on each loop getting bigger and bigger with each loop, or, in case of negative error, getting lower and lower. **The sum of all these parts, the P, the I and D, makes a PID control**. It's our job to find the correct constants for each of this PID elements.

PART 2 K-thermocouple temperature read

So, let's start building this project. The first thing we will do, is see how to read the real temperature. Below we can see a K type thermocouple, and the MAX 66 75 breakout module. This will amplify and compensate the voltage created by the thermocouple. It has an SPI communication so we'll have to connect these pins to the Arduino SPI port.

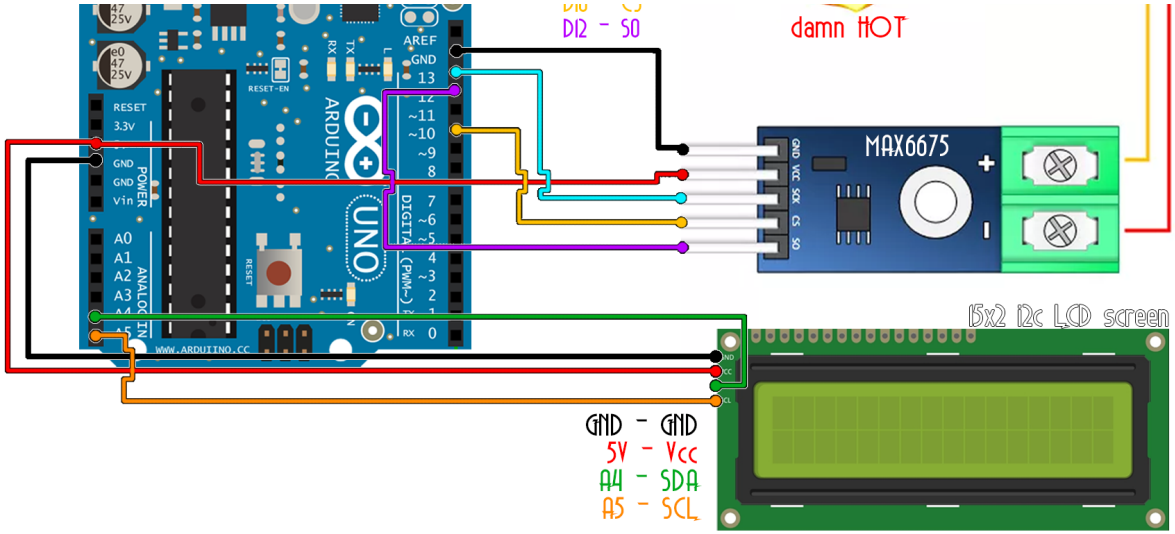


Use the connections below and let's test it out. On a breadboard I connect the MAX 66 75 and the thermocouple. Be careful, the thermocouple has polarity so connect positive to positive and negative to negative. Connect the SPI pins to the Arduino and also supply 5V and GND.

(https://bit.ly/2ZNWgqy)

Brushed moto

DIY Inverter tu



(eng_arduino_tut24_sch1.php)

Download schematic here: DOWNLOAD (eng_arduino_tut24_sch1.php)

Thermocouple read code

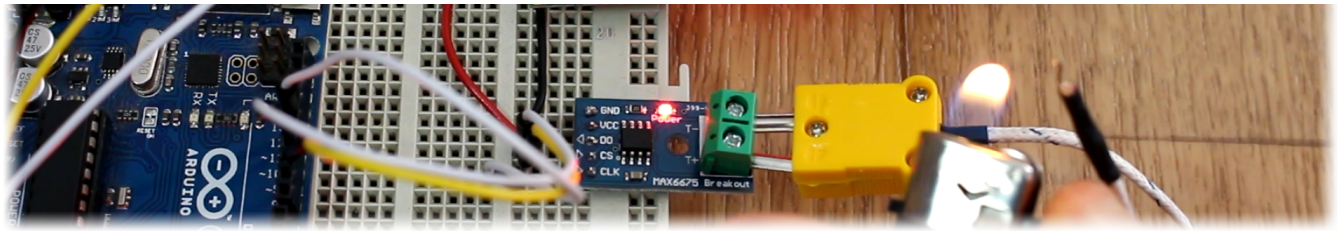
IDE. I heat the thermocouple with a lighter and there you go, I have the real value on the LCD screen.

Download full code here: 

(eng_arduino_tut24_code1.php)

```
/*    Max6675 Module  ==>   Arduino
 *    CS              ==>     D10
 *    SO              ==>     D12
 *    SCK             ==>     D13
 *    Vcc             ==>     Vcc (5v)
 *    Gnd             ==>     Gnd      */

//LCD config
```



NEXT (eng_arduino_tut24_2.php)

Alejandro Herrera Malque

woao me encanto tu projecto solo que me gustaria mas contennido en español no el ingles no lo manejo muy bien pero entiendo . sigue asi amigo buen proyecto

Like · Reply · 4y

Paulo Delos Santos

Can this circuit withstand even 200ºC?

Like · Reply · 4y

Jean-François Moquin

its in kelvin

Like · Reply · 34w

Arsl An

why do you need a transistor to drive the gate of MOSFET? Can this not be done directly with D3 pin? Thanks

Like · Reply · 4y

Viktor Kunz

The Gate of a MOSFET is charge driven. It means the gate doesn't draw current, but needs only a voltage to charge up the gate, so the channel becomes conductive. After charging up the gate remains conductive. The output current of the Arduino isn't that much to charge up the gate quickly, as it needs in this case (PWM). There for the BJT is used to drive more current. A way better is to use complementary BJTs to quickly charge and discharge the gate.

Like · Reply · 2y

Maleeha Babar

Why PID controller is specifically designed here? Why can't we use PI or PD alone ?

Like · Reply · 4y

Safdar Asif

Without the I part, there will be a steady state error, e.g the heater will stop at 99 degrees instead of 100

Like · Reply · 3y

Santosh Babu

Load 5 more comments

ELECTRONOOBS 2022
Keep up you guys!