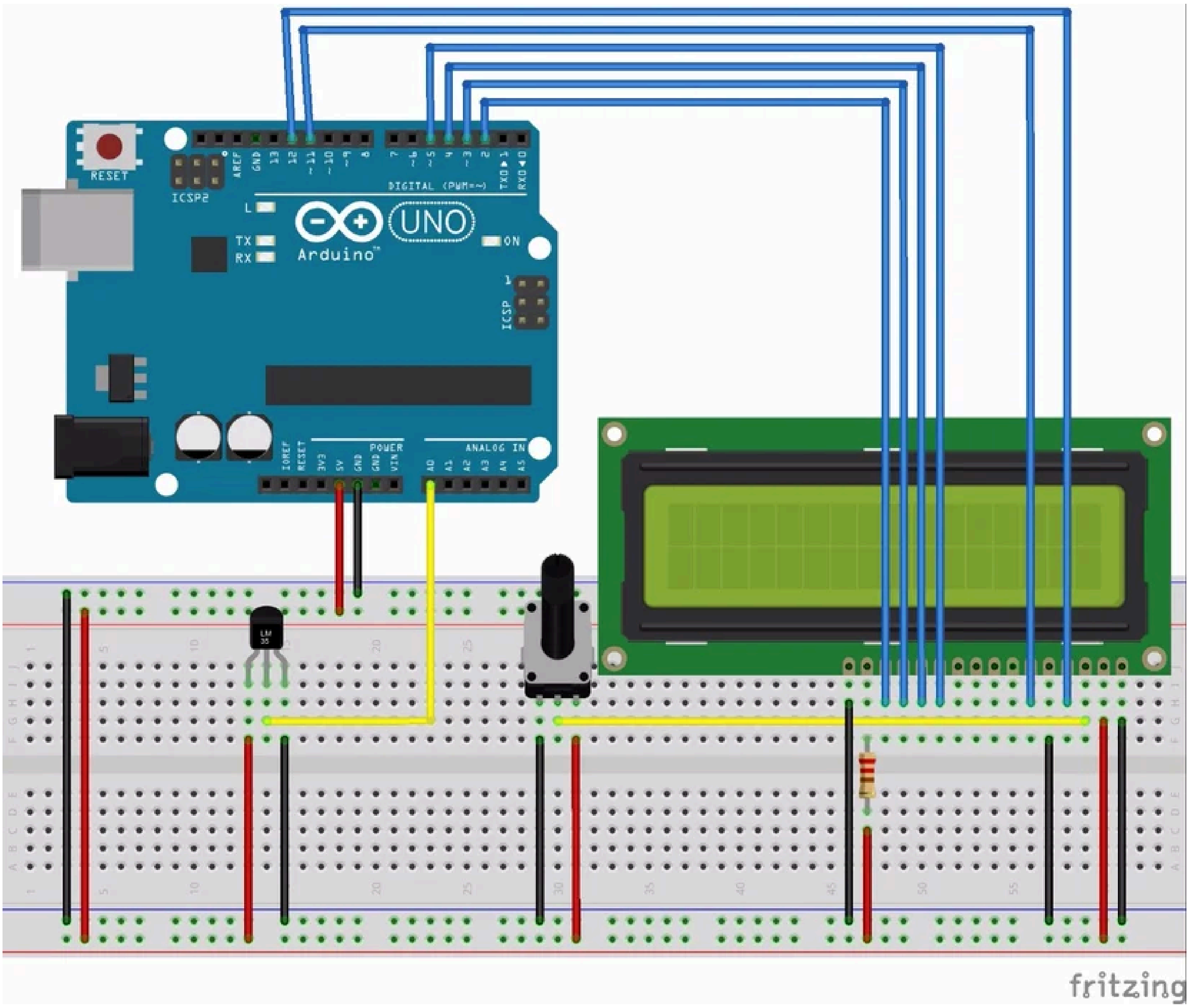
Arduino Uno: Temperature Sensor With Display

By AlexanderP23 in CircuitsArduino

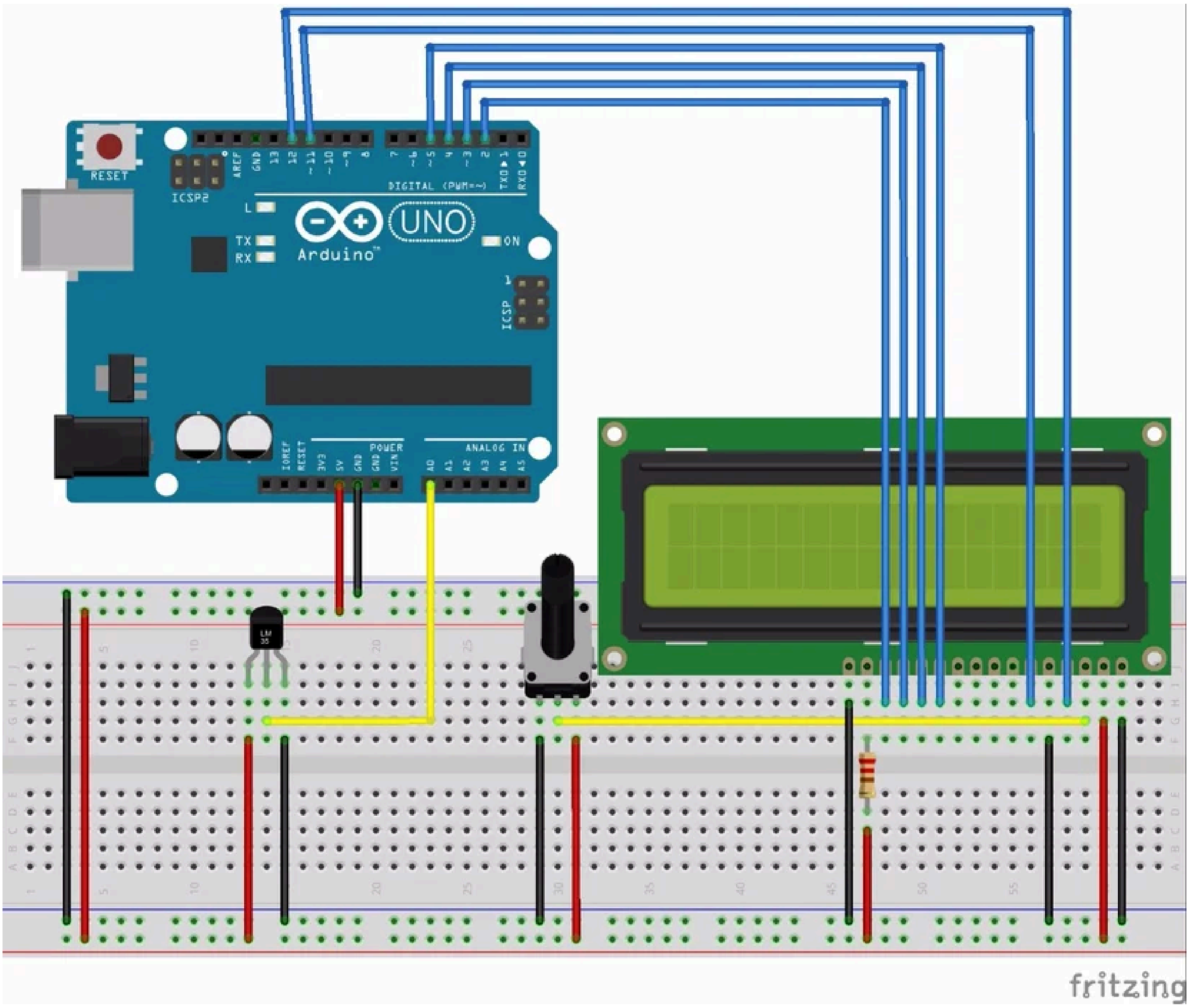
Introduction: Arduino Uno: Temperature Sensor With Display



I wrote a code that displays the temperature on an LCD screen. I used these components:

- Arduino Uno board (rev 3)
- 2x16 character LCD display
- breadboard
- TMP36 temperature sensor
- Potentiometer

Step 1: Step 1: Wiring



Wiring the LCD

Here's some explanation for the wiring of the LCD: pin 7 to 10 is not used in this case. Pin 11 to 14 are used to sent the characters to the display. Pin 14 is the power for the LED backlight and needs to be connected tot +5V with a 220 Ohm resistor in series. Pin 16 is the ground for the LED backlight and needs to be connected to the ground of the Arduino. Pin 1 (Vss) needs to be connected to the ground. Pin 2 needs to be connected to +5V. Both pin 1 and 2 power the LCD display. Pin 3 needs to be connected to the middel connector of the potentiometer and is used tot control the contrast of the display. Pin 4 is the register select connector and controls where the characters will appear on the display. This pin is connected to pin 12 of the Arduino Uno. Pin 5 is the read/write pin and because we will use the display to show things, we'll write to the display. To use the write mode, we need to connect pin 5 to the ground. Pin 6 is the enabler pin and tells the display it wil be receiving a command. This one is connected tot pin 11 of the Arduino Uno.

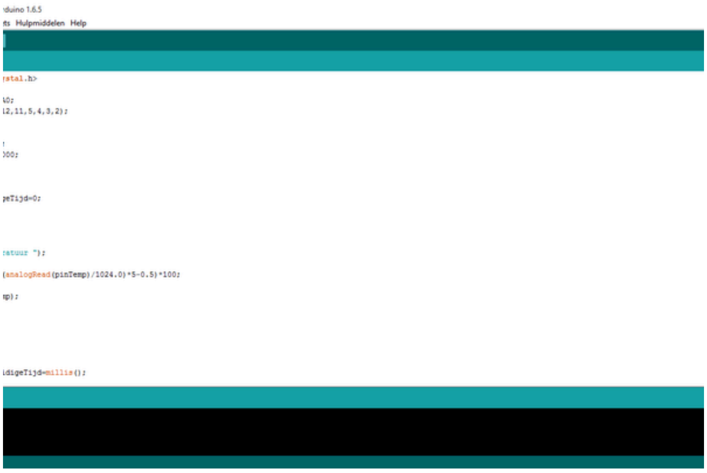
Wiring the TMP36 temperature sensor

The sensor has 3 pins. When facing the flat side of the sensor, this is the pin layout:

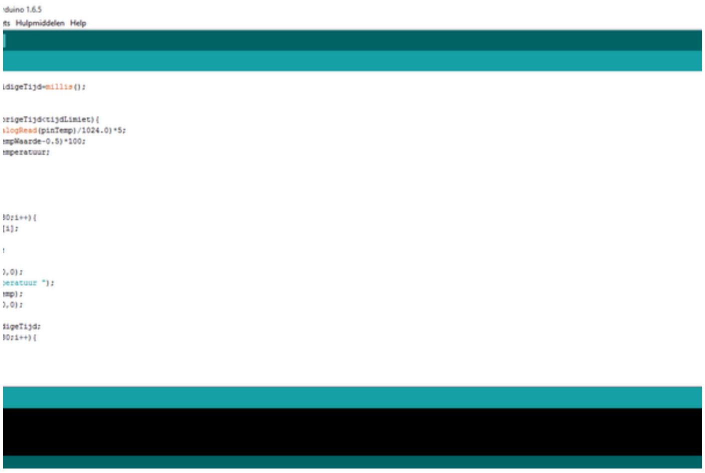
- Left pin: +5V
- Middle pin: analog output (0-5V)
- Right pin: GND

DON'T switch the left and right pin!

Step 2: Step 2: Write the Code



```
rstal.h>

A0;
12,11,5,4,3,2);


:
000;



geTijd=0;





ratuur ");

(analogRead(pinTemp)/1024.0)*5-0.5)*100;

mp);






idigeTijd=millis();
```



```
idigeTijd=millis();


origeTijd<tijdLimiet){
alogRead(pinTemp)/1024.0)*5;
empWaarde-0.5)*100;
emperatuur;






30;i++){
[i];

:

),0);
peratuur ");
emp);
),0);

digeTijd;
30;i++){
```

The code is not that difficult to understand. The key of the code is this: when you read the sensor value in a loop and write it everytime to the display, the value changes constantly. To avoid this, I read the sensorvalue every second and I store it in an array. After 30 seconds, I'm calculating the mean of the values and this value, I write to the display. The value still varies but it's much more stable.

The code: Temperature_LCD.ino (also attached as images)