Antonio Jimenez

Love – o – meter

3/8/20-3/22/20

Lab

**Discover:** analog input using serial monitor

**Time:** 45 minutes

**Level:** beginner 1

**Builds on project:** 1, 2

**Introduction:**

In this project we will be exploring more than on and off, other than from the digital tools the Arduino provides, we can also get analog information from analog sensors to measure light, sound, and temperature. To do this you use the Arduinos built in Analog-to-Digital converter (ADC). The pins would be A0 – A5 which report back numbers 0-1023, which is a range from 0 to 5 volts. We will be learning analog inputs and use a serial monitor.

**Materials:**

* 3 – red LED’s
* 3 – 220Ω resistors
* 1- temperature sensor TMP36
* Jumper wires

**Observations:**

Its been a while since I’ve gotten back into Arduino engineering, so I have started on the next project of this kit which is called the love-o-meter. Basically, what it entails is using a temperature sensor to measure certain degrees in Celsius then with that information it can display that on the serial monitor and outputs up to 3 red lights to turn on.

Alright, I hooked up the Arduino uno to the computer and uploaded the code and it should be working. Well, I expected a more cooperative behavior from the projects but instead when I first uploaded the code and got firsthand look at the results only one of the lights had lit up. When I covered my hands on the temperature sensor, only for a short while, two of the three lights turned on. My goal is to try to make all three lights turn on I tried rubbing my hands together to make more heat, but the result stayed the same. I investigated what could be the cause, so I looked at the serial monitor. The serial monitor is where all the information is being recorded and measured. The serial monitor says that my highest heat that my body could generate is 30.0 degrees Celsius for room temperature the amount is 20.0 degrees Celsius. Since these facts are present, I had to change the code to fit my situation. I changed the base temperature to 21 degrees Celsius and I even added a block of code where all three lights are turned on.

So far, I’m not a big fan of this project its slow and inefficient and I don’t think it would work every time. What you need for this experiment is a room with a controlled temperature and you need to configure this project with the room temperature as the base temperature. And you must also measure how much your body can output. What you need is a controlled subject to where everyone can agree that this person can output the maximum amount of heat and everyone can measure comparing to the controlled subject.

**Notes:**

We will be using a **temperature sensor** to measure the heat of your skin. The sensor has three pins: one that connects to ground, another that connects to power, and a third that outputs a variable voltage to the Arduino (which is the middle pin). There are different models for a temperature sensor, we will be using a **TMP36** because the output voltage is directly proportional to the temperature in degrees Celsius. Basically how this process works is that the temperature senses a change in temperature then it takes that information to the Arduino, the Arduino takes the next step in converting values then executing the command by sending out signals from the pins.

The Arduino IDE is equipped with a **serial monitor** which reports back results from the microcontroller. You can use the serial monitor to get information from the analog outputs and inputs.

Warning: in this project you need to check the ambient temperature of the room before proceeding. Usually you would do this manually by looking up the temperature, or you could do this through calibrations. Have the Arduino take a sample before starting the loop function. Project 6 get into more detail, but you can look for the calibration tool using the IDE

When assembling, you assemble how you would assemble in the past project. Place the TMP36 on the breadboard have the rounded part facing away from the Arduino (order of the pins is important) facing the flat side, the left pin is the power pin, the middle pin is the pin where we connect to the Arduino analog in, and finally the right pin where we have connected to ground.

**Code:**

Again, the code is divided into three sections initialization, setup, and loop. According to the code the initialization is where we define the variable and constants are variables that do not change in this program. We name the sensor input for reference and the temperature as a reference point. The project has made it a point that for every increase of 2 degrees from the base temperature one LED turns on. We have already seen which is a whole number, the temperature is a decimal, but the data type is called a .

In the setup we start with this makes a connection between the Arduino and computer, so now you can be able to use your serial monitor on your computer. The argument 9600 is the rate at which the computer communicates with you, 9600 bits per second. A for loop is in the setup, it is used to initialize the output pins. Instead of having to write out every time, you could just use a for loop to reduce the redundancies.

In the loop section, the first line of code you get is:

The function reads the value that being place inside the A0 pin which we named , that value could be between 0 and 1023 which is a representation of the voltage on the pin. We then place that value in . The next piece of code you encounter is:

is a function that sends information from the Arduino to the serial monitor. can print out strings with quotation marks.

We now move on to converting values by using a little math:

The voltage will be a value between 0 and 5 therefore we create a ratio of the sensor values and multiply it by 5 to get an exact voltage, we then store it inside a float data type. We then print out the voltage value using .

You can review the data sheet of the starter kit components at arduino.cc/kitdatasheets you can use that data sheet to look at the range of output voltage for a component. The data sheet explains that every change in 10 millivolts from the sensor is equivalent to a temperature change in 1 degree Celsius. We can also see that the sensor also measures negative temperatures. Because of this, you need to offset for values below freezing (0 degrees). If we subtract the voltage by 0.5 and multiply by 100, we can get the accurate temperature in Celsius. We then store the value in

With the real temperature you can set up using an if else statement to turn on the LED’s in certain situations. Using the baseline of the temperature as a starting point, you will turn on one LED for every two degrees the base line temperature increases.

If you focus on line 39 you can see that there is a && symbol in the conditional statement, this means “AND” in the code effectively saying “do this block of code if this statement and this statement are both true” you can see the rest of the code is a bunch of conditional statements.

Once you have uploaded your code, you can see that the serial monitor is already collecting data.

**Possible uses:**

One of the uses I can think of right away with this project is anything that deals with analog inputs. With the temperature sensor I believe we can use this to measure temperature and, in any case, when a certain temperature is reached the Arduino can sense it and do something. Possibly I could make a system if a room is too cold or too hot the sensor can pick it up and adjust the room temperature for me. I could also probably use this on a robot to pick up high heat or cold weather.

**Conclusion:**

Expanding the types of inputs, you can read, you’ve used and the serial monitor to track changes inside your Arduino. Now it is possible to read a large number of analog inputs.