UP Competitive Robotics Club (CRC)
Research and Design Committee
Circuit prepared by: Alyssa Leila Bugas & Francis Mikael Magueflor
Documentation prepared by: Jan Lendl Uy

TMP36 (Temperature Sensor) Documentation

I. Overview of the Circuit

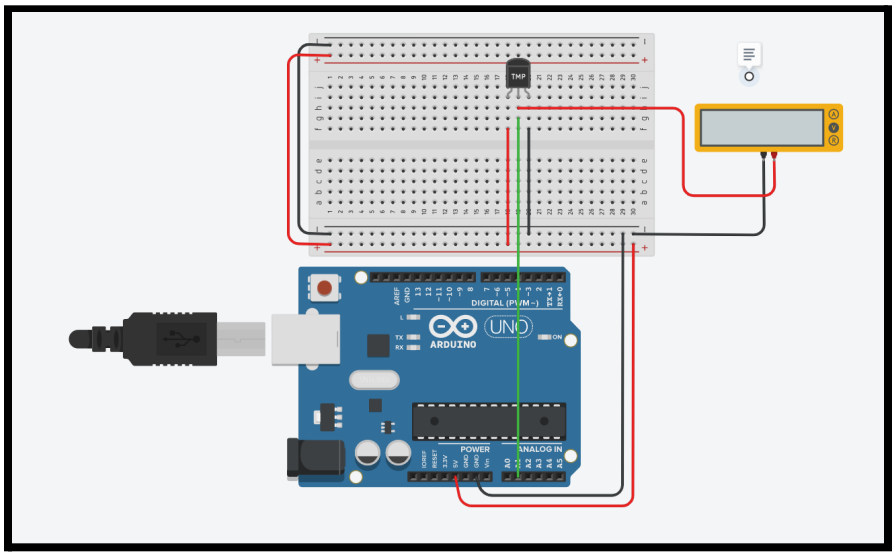

Fig. 1. Schematic implementation of the temperature sensor circuit in Tinkercad

The temperature sensor circuit comprises the following components: (i) Arduino Uno and a (ii) TMP36 sensor. The circuit essentially functions to detect the ambient temperature.

A. Breadboard

A breadboard, otherwise known as a protoboard, serves as a platform for electronic components. It is the preferred base for constructing circuits due to the ease of connecting devices. Placing devices on a breadboard only involves attaching the two ends of a component (can be more than two for

some devices) on the contact holes of the board. The breadboard consists of two areas for electric connection, the bus strips and the terminal strips. For the former, these are usually located at the ends of the board (i.e. when oriented in landscape, they can be found at the top and at the bottom). In the schematic above, this can be located by looking for the rows with the gray-colored and red-colored strip, denoted by a negative and a positive sign respectively. These strips are usually utilized for connecting a supply (e.g. voltage source) to the circuit. The electrical connection for the bus strips is horizontal, making them useful for adding multiple connections to the supply. On the other hand, the terminal strips are the areas where the majority of the components are placed and connected with each other. These are located in the central region of the board, which can be identified by looking at the area with more contact holes. Unlike with the bus strips, the electrical connections for this area of the board is vertical.

B. Arduino Uno

Arduino Uno is a programmable microcontroller which consists of fourteen (14) digital pins and six (6) analog pins. It can be operated through an external voltage source (e.g. battery) rated between seven (7) and twenty (20) volts or through a USB connection. For the circuit implementation, only a single analog pin, i.e. *pin A1*, is used.

C. TMP36 Temperature Sensor

The TMP36 is an analog sensor which measures temperature. Unlike other temperature measurement devices, the TMP36 operates by varying the voltage drop across the component depending on the ambient temperature. This voltage can be mathematically converted into a unit for temperature (e.g. Celsius) based on a predetermined linear relationship between the two aforementioned quantities.

II. Breakdown of the Code

A. Setting up Arduino Uno

```
void setup()
```

```
{
  Serial.begin(9600);
  pinMode(A1, INPUT);

}
```

Fig. 2. Initializing Arduino Uno

To test the functionality of the light sensor circuit, the Arduino Uno must first be initialized in order to be able to collect current readings from the photodiode. In the code above, the polling rate is set to 9600 bits per second. This is a standard line of code for programs involving collection of data from a circuit device, in this case, the photodiode. The second line of code essentially initializes *pin A1* for the collection of voltage readings, which will be displayed in the multimeter.

B. Obtaining the voltage reading and converting to Celsius

```
void loop()
{
  //Local variables
  float current = 0;
  float temp = 0;
  float mV = 0;

  //Notes:
  //read raw value at pin A1
  float raw_val = analogRead(A1);

  //Notes:
  //raw_val is a value from 0 - 1023 (1023 representing 5V)
  //to get the actual voltage in mV it we need to scale it
  //we can do this using the map function
  //you can read about the function here:
  //https://www.arduino.cc/reference/en/language/functions/math/map/
  mV = map(raw_val ,0,1023,0,5000);
  Serial.print("V: ");
  Serial.print(mV);
  Serial.print("mV");


  //Notes:
  //after getting the voltage we can then calculate the
  //temperature
  //500 mV offset is from the datasheet:

//https://cdn-learn.adafruit.com/assets/assets/000/010/131/original/TMP
```

```
35_36_37.pdf
  temp = ( mV - 500 ) / 10;
  Serial.print(" Temp: ");
  Serial.print(temp);
  Serial.print("C\n");
}
```

Fig. 3. Conditional logic for displaying current readings

The purpose of the code above is essentially to obtain the voltage drop of the TMP36 sensor and to convert the voltage reading into Celsius. Upon the initialization of important variables, the raw voltage reading of the TMP36 is obtained. The possible values that can be achieved from the sensor varies between 0 and 1023, the latter representing a $5V$ reading. This raw value is then converted into the actual voltage using the *map* function. To obtain the temperature measured by the sensor, the following formula is used:

$$T = \frac{V_{actual} - 500}{10}$$

The mathematical equation above is presented in the datasheet of the TMP36 sensor. Note that the actual voltage is in the millivolt range. The calculated temperature in Celsius is then displayed in the *Serial Monitor* tab of the Tinkercad Web Application, which can be found in the lower right portion of the interface.