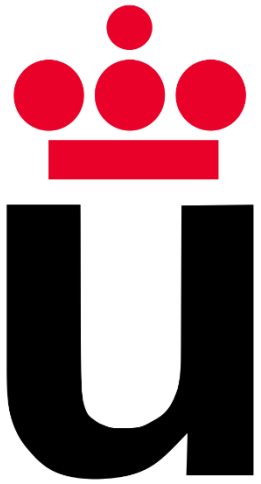


Digital Systems and Microprocessors

Project 3: Analog to Digital Converter – ADC, and sensors

Enrique Almazán Sánchez
Víctor Miguel Álvarez Camarero

Table of Contents

Objetive

The general objective of this project is to demonstrate the use and skill using the Arduino System, and the handling of different types of sensors.

Description

Connect the LDR and LM35 temperature sensor to the A0 and A1 analog inputs of the Arduino development board, respectively.

Requirements

For this setup, we will require the following components:

- Arduino Mega 2560
- 7-segment display (used as an output device)
- Cables for connections
- 7 resistors with a resistance of 220Ω each
- 1 resistor with a resistance of 1KΩ.
- Protoboard for circuit assembly
- **LDR (Light Dependent Resistor)**: photoresistor-based light sensor, a resistance that varies its value depending on the amount of light that falls on its surface. The greater the amount of light that falls on the surface of the LDR or photoresistor, the lower its resistance value, and the less light that falls on its surface, the higher its resistance value. It has 2 pins, one connected to the analog input A0 and the other to GND.
- **LM 35**: a temperature sensor which pins are shown below. The temperature sensor has 3 terminals labelled Vcc (connected to +5V), OUT (connected to the analog input A0) and GND (connected to GND).

Exercise 1

Write a code in which light incident on the LDR is measured and displayed on an 7-segment display the intensity of the measured amount of light. The measured incident light will be divided into 6 ranges from 0 to 5, and each range will be divided into 171 units (ADC conversion intervals).

When the sensor output voltage is between 0 and 171 (code generated by the ADC), range 0 is displayed, displaying the number 0, and so on. The minimum value displayed, range 0, is when the incident light on the LDR is maximum, while the maximum value displayed, range 5, is when the incident light is minimal, i.e no light.

The range of the light detected by the LDR should also be displayed on the serial monitor. Therefore, the following message "Measured light range:" should be displayed, after this message the numerical value of the range should be displayed.

The code presented for this exercise reads the output of a LDR sensor and displays the intensity of the measured light on a 7-segment display, being 0 the highest amount of light received and 5 the largest one .

First of all, some variables are initialized, as well as the baud rate, the speed at which data bits are going to be transmitted, at 9600.

- “analogpin” (constant integer): set to A0, which is the analog input pin connected to the LDR sensor.
- “sensorvalue” (float): initialized to 0, used to store the analog reading from the LDR sensor.
- “cont” (integer): initialized to 0, used to represent the range of light intensity detected by the LDR sensor.
- “displaypin”: defines de pins connected to the 7-segment display.
- “displaycode” (array): defines the values to be displayed on the 7-segment display.

```
const int analogpin = A0;
const int baudrate = 9600;
float sensorvalue = 0;
int cont = 0;
int displaypin[] = {2,3,4,5,6,7,8};

int displaycode[10][7]=
  {
  {1,1,1,1,1,1,0}, //0
  {0,1,1,0,0,0,0}, //1
  {1,1,0,1,1,0,1}, //2
  {1,1,1,1,0,0,1}, //3
  {0,1,1,0,0,1,1}, //4
  {1,0,1,1,0,1,1}, //5
  {1,0,1,1,1,1,1}, //6
  {1,1,1,0,0,0,0}, //7
  {1,1,1,1,1,1,1}, //8
  {1,1,1,1,0,1,1}, //9
  };
```

Moreover, in the setup function, begins serial communication with the specified baud rate ("*Serial.begin*"). It continues setting both, the analog reference pin to the power supply of the board ("*analogReference()*") and the pins connected to the 7-segment display as outputs ("*pinMode()*"). Then, the 7-segment display is initialized ("*segmentdisplay()*") with the initial value of "cont", printing its initial value to the serial monitor ("*Serial.print()*" and "*Serial.println()*"). Delays of 100ms and 500ms are presented for the displaying and printing respectively.

```
void setup() {
  Serial.begin(baudrate);
  analogReference(DEFAULT);
    for(int i=0;i<=6;i++){
    pinMode(displaypin[i], OUTPUT);
  }
  Serial.begin(baudrate);
  delay(100);
  segmentdisplay(cont);
  Serial.print("Number = ");
  Serial.println(cont);
  delay(500);
}
```

Furthermore, the loop function reads the analog input from pin "A0" from the LDR sensor ("*analogRead()*") and stores it in a variable ("*sensorvalue*"). As the sensor reading value varies from 0 to 5 volts, the microcontroller internally represents the analog voltage with a 10-bit number, ranging from 0 to 1023. To obtain the real-world voltage, the "sensorvalue" is scaled linearly by performing the following operation:

$$sensorvalue * 5 / 1023$$

This calculated voltage value is then printed to the serial monitor ("*Serial.print*"). Hence, based on the "sensorvalue", the code determines the range of light intensity detected by the LDR sensor. This range is used to update the "cont" variable, reflecting the detected intensity range. The corresponding range is displayed on the 7-segment display, and the detected range is printed to the serial monitor. After each iteration, the code pauses for 200ms before proceeding to the next loop iteration. In addition, 6 different ranges divided into 171 units are obtained, which can be seen in the following code:

```
void loop() {
  sensorvalue = analogRead(A0);

  Serial.print("ADC code = ");
  Serial.println(sensorvalue);
  Serial.print("Inpt voltage source = ");
  Serial.println(sensorvalue*5/1023);
  delay(200);

    if ((sensorvalue >= 0) && (sensorvalue<=171)){
    cont=0;
  }
    else if ((sensorvalue >=171) && (sensorvalue <= 342)){
```

```
    cont=1;
  }
    else if ((sensorvalue >=342) && (sensorvalue <= 513)){
    cont=2;
  }
    else if ((sensorvalue >=513) && (sensorvalue <= 684)){
    cont=3;
  }
    else if ((sensorvalue >=684) && (sensorvalue <= 855)){
    cont=4;
  }
    else if ((sensorvalue >=855) && (sensorvalue <= 1023)){
    cont=5;
  }
  segmentdisplay(cont);
  Serial.print("MEASURED LIGHT RANGE = ");
  Serial.println(cont);

  delay(200);
}
```

Finally, the segmentdisplay function, which accepts an integer argument, representing the digit to be displayed, writes the binary codes for the specified digit to the output port pins connected to the 7-segment display ("*digitalWrite()*").

```
void segmentdisplay(int x){
  for(int i=0;i<=6;i++){
    digitalWrite(displaypin[i], displaycode[x][i]);
  }
}
```

The assembly circuit can be seen in the below figure, while the entire implementation in the video 'Exercise_1'.

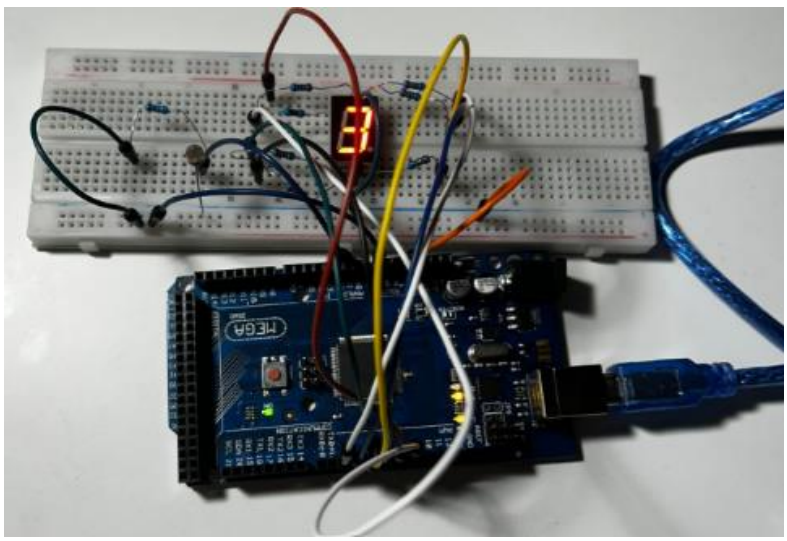


Figure 1: Assembly circuit for a LDR with a 7-Segement display.

Exercise 2

Write a code where the temperature detected by the LM35 is measured and that temperature is displayed on the serial monitor. Two temperature ranges are selected, 25ºC and 30ºC, to establish a minimum and maximum working limit of the system.

When the temperature sensor detects a temperature below the minimum limit, the following message, "Temperature below minimum limit" appears on the serial monitor. When the temperature sensor detects a temperature above the maximum limit, the following message, "Temperature above maximum limit" will be displayed on the serial monitor.

When the temperature sensor detects a temperature between the minimum and maximum limit, the following message, "Temperature normal" appears on the serial monitor.

The code presented for this exercise reads the output of a temperature detector (LM35 sensor). Depending on the temperature recorded by this sensor an output message will be displayed in the Serial Monitor.

First of all, some variables are initialized, as well as the baud rate at 9600. As in the previous exercise the variables “analogpin” (set to `A0’), and “sensorvalue” (float, initialized to 0), but for the LM 35 temperature sensor, while the rest are suppressed as the 7-segment display was not added.

```
const int analogpin = A0;
const int baudrate = 9600;
float sensorvalue = 0;
```

The setup function also has similarities with exercise 1, starting serial communication with the specified baud rate (“*Serial.begin*”) and setting the analog reference pin to the power supply of the board (“*analogReference()*”). A delay of 500ms is presented.

```
void setup() {
  Serial.begin(baudrate);
  analogReference(DEFAULT);
  delay(500);
}
```

In the loop function the analog input from pin A0 is read, representing the LM 35 temperature sensor, and stores it in the variable sensorvalue (“*analogRead()*”). It then prints the raw ADC value read from the sensor to the serial monitor (“*Serial.print()*” or “*Serial.println()*”).

```
void loop() {
  sensorvalue = analogRead(A0);

  Serial.print("ADC code = ");
  Serial.println(sensorvalue);
  Serial.print("Inpt voltage source = ");

  float temp = sensorvalue*5/1023; // temperature in voltage
  Serial.println(temp);
  float t = temp*100; // temperature in volts

  delay(1000);
```

```
  // The following if-else body for the different values that the temp
variable will get depending on the sensor value
  (if-else body seen below)

  delay(1000);
}
```

Moreover, the temperature value in volts (temp) is computed by scaling the sensor value to a voltage between 0 and 5 volts, as well as the temperature in degrees by scaling with the relationship given in class (100 times the temperature in volts). The computed voltage value (temp) is printed to the serial monitor ("*Serial.print()*" or "*Serial.println()*"), as well as the degrees. Also, a delay of 1 second is added. Subsequently, the code checks the value of temp to determine the temperature range:

```
  if (temp<0.25){
    Serial.print("Temperature is below 25 degrees. Exactly");
    Serial.println(t);
  }
  else if ((temp>=0.25)&&(temp<=0.3)){
    Serial.print("Temperature is normal. Exactly");
    Serial.println(t);
  }
  else{
    Serial.print("Temperature is above 30 degrees. Exactly");
    Serial.println(t);
  }
```

Finally, a delay of 1 second is included between each loop iteration for stability using delay(1000).

The assembly circuit can be seen in the below figure, while the entire implementation in the video 'Exercise_2'.

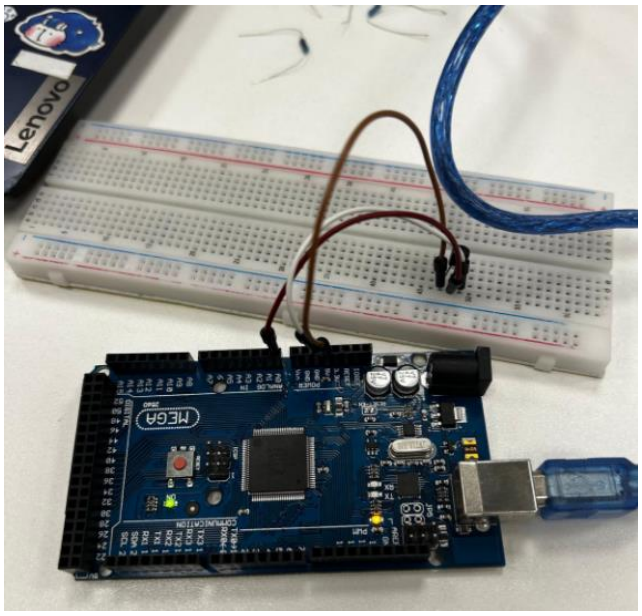
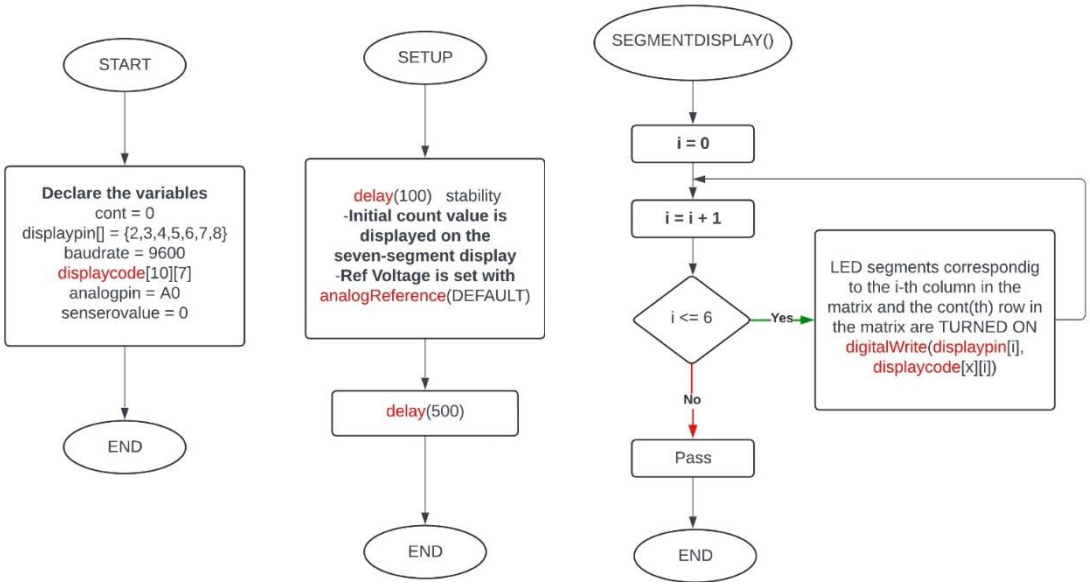
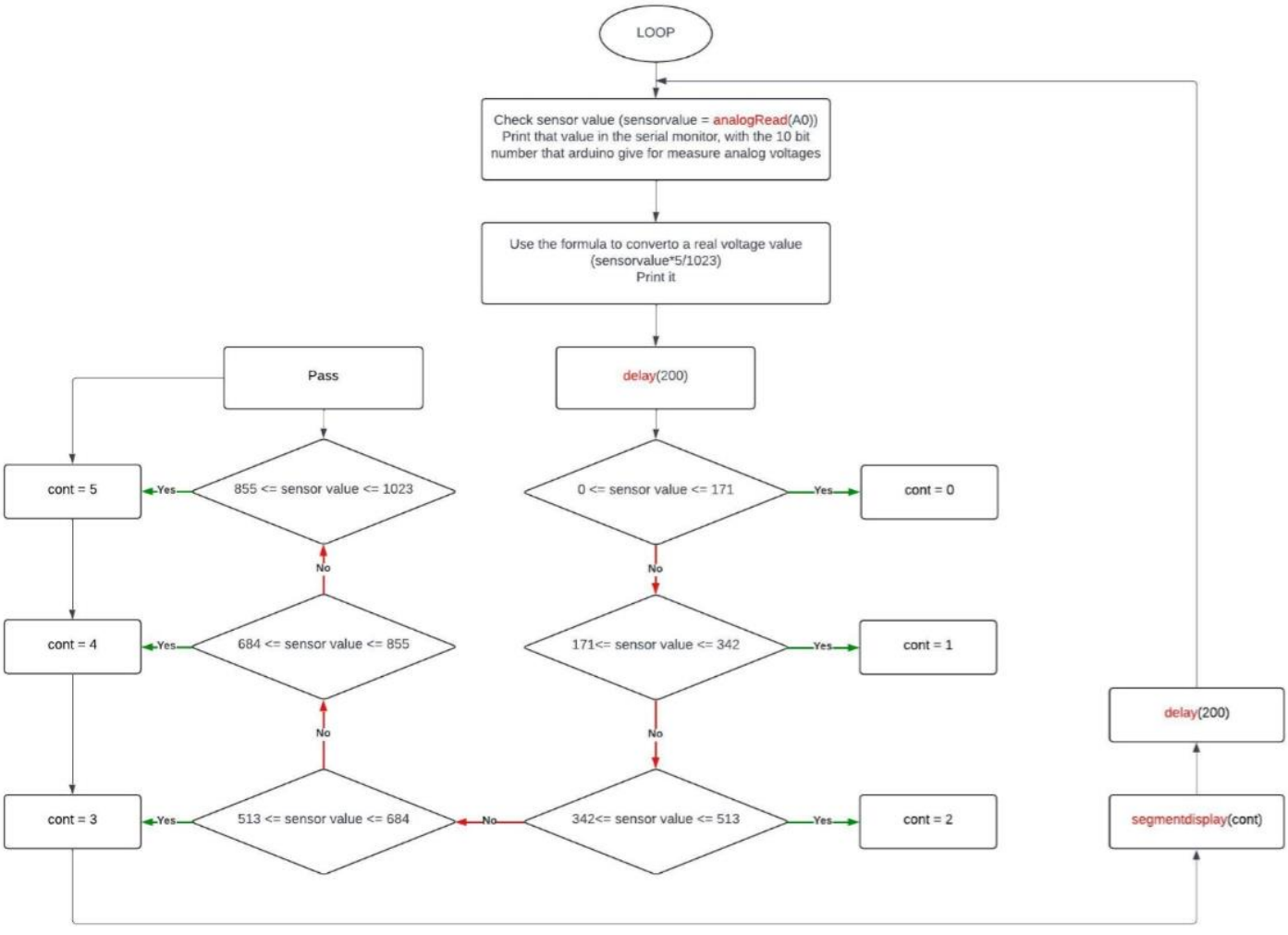


Figure 2: Assembly circuit with a LM 35 temperature sensor.

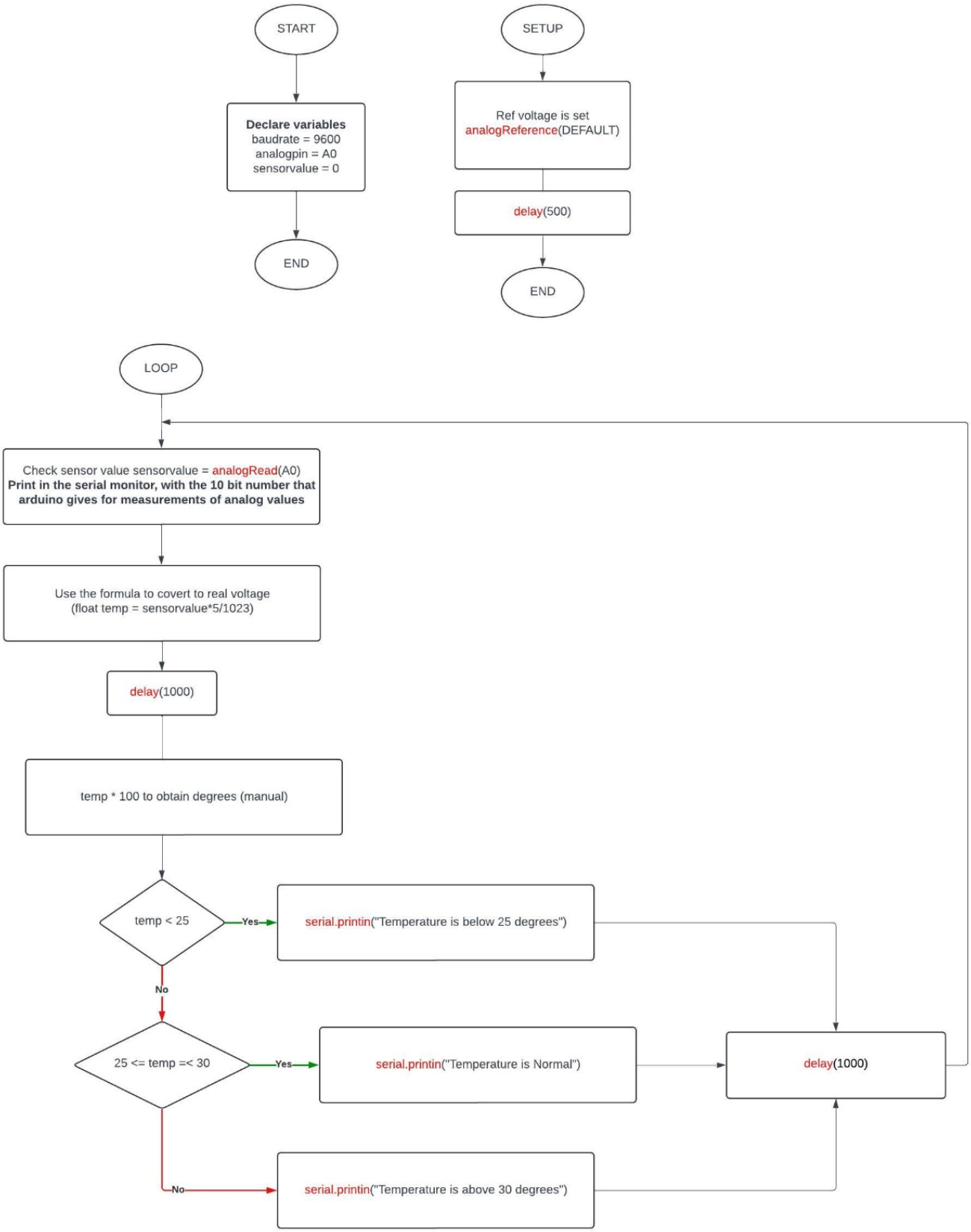
Flowcharts

Exercise 1





Exercise 2


START
Declare variables
baudrate = 9600
analogpin = A0
sensorvalue = 0
END
SETUP
Ref voltage is set
analogReference(DEFAULT)
delay(500)
END
LOOP
Check sensor value sensorvalue = analogRead(A0)
Print in the serial monitor, with the 10 bit number that arduino gives for measurements of analog values
Use the formula to covert to real voltage
(float temp = sensorvalue*5/1023)
delay(1000)
temp * 100 to obtain degrees (manual)
temp < 25
Yes
serial.printin("Temperature is below 25 degrees")
No
25 <= temp =< 30
Yes
serial.printin("Temperature is Normal")
delay(1000)
No
serial.printin("Temperature is above 30 degrees")