ELEN30013 Workshop 8: Sensors

You should get the marks during the workshop. You can only be marked in your workshop in Week 8

Total points: 100 points

Attendance: 20 points

1. Thermistor and Simple Temperature Sensor

1.1 Basic Thermistor Circuit

Step 1: Thermistors are variable resistors that change their resistance with temperature. In Negative Temperature Coefficient (NTC) thermistors the resistance decreases as temperature increases. In this tutorial, we will use NTC thermistors, which are the most commonly used type of thermistor. The relationship of NTC thermistor resistance vs temperature is:

$$R_T = R_0 * e^{B*\left(\frac{1}{273+T}-\frac{1}{273+T0}\right)}$$

Where B is the function shape parameter (usually we use B at 25°C, you could find B in the datasheet).
T_0 is the reference temperature which is 25°C.
R_0 is the reference resistance which should be 10K.

With these equations, you can calculate the temperature by reading the thermistor's resistance value. Please answer Question 1.1 using this equation.

Step 2: Construct a circuit on a breadboard using this simple voltage divider (see figure 1) where the Arduino Analog input pin directly reads the Thermistor's voltage level:

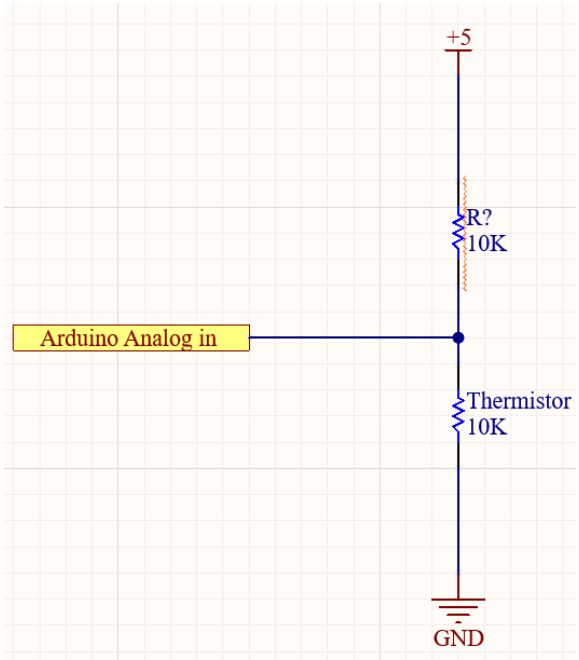


Figure 1. Simple thermistor voltage divider

Step 3: Copy and paste the following code (code 1) into your Arduino IDE and run the serial monitor, observe the readings.

```
int ThermistorPin = 0; //Set value to match which analog input you are using
float Vo;
void setup() {
Serial.begin(9600);
}
void loop() {
  Vo = analogRead(ThermistorPin);
  Serial.print("Voltage reading: ");
  Serial.println(Vo);
  delay(1000);
}
```

Code 1. Simple analog read with serial print

Step 4: Now what you can see from the readings should be a number from 0 – 1023, not the actual voltage reading value, that is because Arduino's ADC convertor convert the analog voltage 0 – 5 Volts to 1024 different digital level value. Please modify the code to the following (see code 2):

```
int ThermistorPin = 0; //Set value to match which analog input you are using
float Vo;

void setup() {
Serial.begin(9600);
}

void loop() {
  Vo = 5.0*analogRead(ThermistorPin)/1024.0;
  Serial.print("Voltage reading: ");
  Serial.println(Vo);
  delay(1000);
}
```

Code 2. Simple analog voltage read with serial print

Step 5: Modify the code (code 2), according to the requirement of Question 1.2.

1.2 Arduino LCD

The LCD Keypad Shield is Arduino compactible boards, to provide a user-friendly interface that allows users to display what they want to and make selections etc. it consists of a 1602 white character blue backlight LCD. The keypad consists of 5 keys — select, up, right, down and left. To save the digital IO pins, the keypad interface uses only one ADC channel (AD0). The key value is read through a 5-stage voltage divider. In this workshop, we will only use the display part. If you have interest in how the buttons work, you can refer to the link: https://www.elecrow.com/wiki/index.php?title=LCD_Keypad_Shield

Step 1: Connect the following pins on the LCD screen to the pins on your Arduino board:

LCD Pin	Arduino Pin
RS	Digital 8
Enable	Digital 9
D4	Digital 4
D5	Digital 5
D6	Digital 6
D7	Digital 7
VSS	GND
VCC	5V

Table 1. LCD wiring

Step 2: Copy and paste the code into your Arduino IDE and run the serial monitor, observe the results.

```
#include <LiquidCrystal.h>
// include the library https://www.arduino.cc/en/Reference/LiquidCrystal

const int rs = 8, en = 9, d4 = 4, d5 = 5, d6 = 6, d7 = 7;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
```

```
void setup() {
  lcd.begin(16, 2);   // set up the LCD's number of columns and rows:
  Serial.begin(9600); // start the library
}

void loop() {
  lcd.setCursor(0,0);  //set cursor to the start of the first line
  lcd.print("Number:"); // print a simple message
  lcd.setCursor(0, 1); //move cursor to the start of the second line
  // print the number 64 since reset:
  lcd.print(64);
}
```

Code 3.

1.3 Questions

Question 1.1 (10 points):

Please calculate the temperature (assuming a reference temperature of 25°C) based on the readings of voltage value. (Hint: First find the B and R0 under 25°C in the thermistor's specifications: https://components101.com/resistors/ntc-thermistor-10k). Write down the results and give an explanation.

Question 1.2: (10 points)

Please modify the code in step 1.1.4 (code 2), let Arduino directly print *via* serial the temperature value in degrees Celsius.

Question 1.3: (10 points)

Measure the voltage value of thermistor using the Analog Discovery 2 and compare the readings from Arduino. Give some comments on any difference or similarity.

Question 1.4: (10 points)

Combine section 1.1 (Basic Thermistor Circuit) and section 1.2 (Arduino LCD), design a circuit and create Arduino code which can directly show the temperature value on the LCD screen.

2. DHT11 Sensor (Temperature Sensor)

2.1 Temperature Sensor Reading

Step 1: Wire up the circuit as described in the following table (table 2):

DHT11 Pin	Arduino Pin
+	5V
-	GND

Table 2. DHT11 wiring

Step 2: Before you can use the DHT11 with the Arduino, you need to install the DHTLib library. It's easy to install, just download the DHTLib.zip file below and open the Arduino IDE.

Then go to Sketch>Include Library>Add .ZIP Library and select the DHTLib.zip file. Copy and paste the code into your Arduino IDE and run the serial monitor, observe the results.

```
#include <dht.h>
// include the library from LMS
// http://arduino.cc/playground/Main/DHTLib
dht DHT;

#define DHT11_PIN 3

void setup(){
  Serial.begin(9600);
}

void loop(){
  int chk = DHT.read11(DHT11_PIN);
  Serial.print("Temperature = ");
  Serial.println(DHT.temperature);
  delay(100);
}
```

Code 4.

2.2 Questions

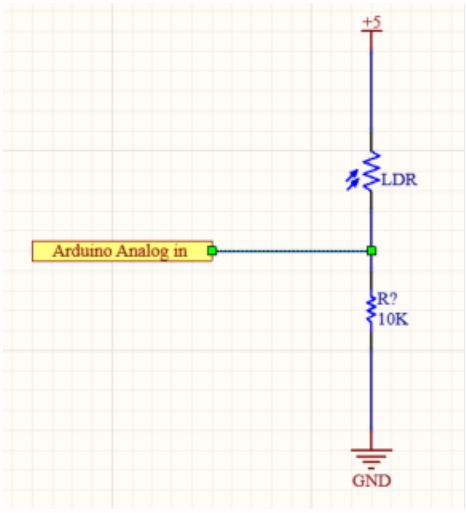
Question 2.1: (20 points)

Calibrate the thermistor circuit with the DHT11 Sensor, list detail steps and give reasonable explanations.

3. Photoresistor and simple light compensation circuit

3.1 Photoresistor reading

Step 1: A photoresistor (also known as a light-dependent resistor, LDR, or photo-conductive cell) is a passive component that decreases resistance with respect to receiving luminosity (light) on the component's sensitive surface. The resistance of a photoresistor decreases with an increase in incident light intensity. At the start, we will construct a simple reading circuit. Connect circuit in bread board as follows:



Step 2: Copy and paste the code below into Arduino IDE. Observe the result and answer Question 3.1.

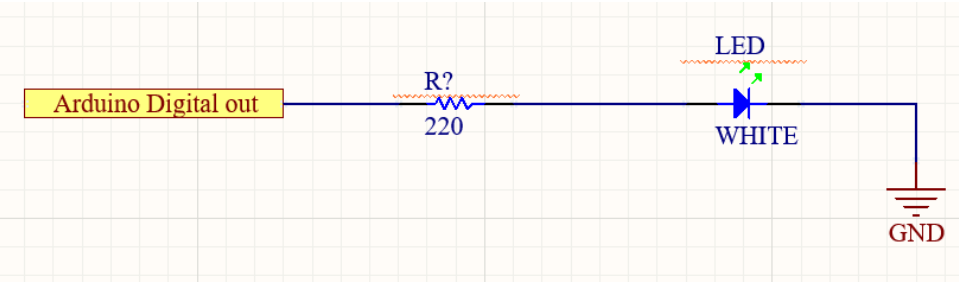
```
const int ldrPin = A0;  //The analog input pin connect to LDR

void setup() {
Serial.begin(9600);
pinMode(ldrPin, INPUT);
}

void loop() {
int ldrReading = analogRead(ldrPin);
Serial.println(ldrReading);
}
```

Code 5.

Step 3: Next, connect another LED to the Arduino digital output pin as follows.



Step 4: Copy and paste the code below into Arduino IDE. Give a reasonable threshold value according to the answer of Question 3.2. Observe the result.

```
const int ledPin = 13;
const int ldrPin = A0;
const int threshold;

void setup() {
Serial.begin(9600);
pinMode(ledPin, OUTPUT);
pinMode(ldrPin, INPUT);
}

void loop() {
int ldrReading = analogRead(ldrPin);
if (ldrReading <= threshold) {
digitalWrite(ledPin, HIGH);
Serial.print("Its DARK, Turn on the LED : ");
Serial.println(ldrReading);
} else {
digitalWrite(ledPin, LOW);
Serial.print("Its BRIGHT, Turn off the LED : ");
Serial.println(ldrReading);
}
}
```

Code 6.

3.2 Questions

Question 3.1: (15 points)

Note down the Arduino monitor reading under these three cases: 1. Light a flashlight directly on the LDR. 2. Block the LDR from light source. 3. Room light.

Question 3.2: (5 points)

Modify the code in step 4. Let the active buzzer change by PWM control according to the LDR reading value. (hint: Using tone() https://www.arduino.cc/reference/en/language/functions/advanced-io/tone/)