Experiment No. 02 – Interfacing Simple Input Devices

Part 1 – LDR Data IN Serial Monitor OUT

Equipment List:

1. 1x Arduino Nano
2. 1x USB 2.0 to USB type A connecting cable
3. 1x Breadboard
4. 1x 10kΩ Resistor
5. 1x LDR (Light Dependent Resistor)

Schematic Diagram:

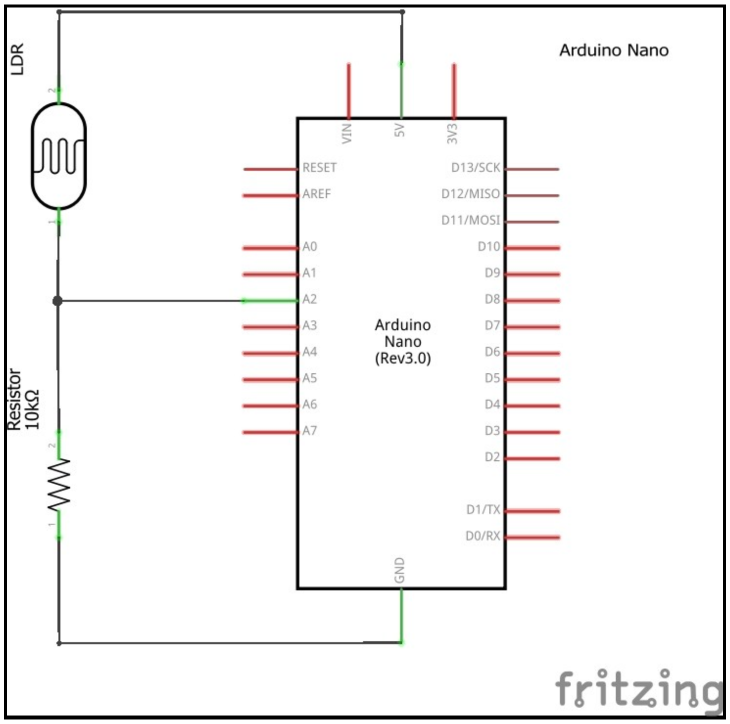

Figure 1 - Exp. 02 - Part 1 – LDR interfacing

Experiment Setup:

- Connect one leg of the LDR (Light Dependent Resistor) with the 5v pin of Arduino.
- Connect the other leg with a 10kΩ resistor.
- Connect the other leg of the 10kΩ resistor with the GROUND pin of the Arduino.
- Connect the leg of the LDR to any ANALOG (in our case it is 2) pin of ARDUINO.
- Compile and upload the given code.
- Open the SERIAL MONITOR in Arduino IDE.
- Analyze the output.

Code:

```
int sensorPin = 2;   // select the input pin for LDR
int sensorValue = 0; // variable to store the value coming from the sensor
void setup() {
      Serial.begin(9600);     //sets serial port for communication
}
void loop() {
      sensorValue = analogRead(sensorPin); //read the value from the sensor
      Serial.println(sensorValue);  //prints the values coming from the
                                      sensor on the screen
      delay(100);
}
```

Code Listing 1 - Code for LDR interfacing

Expected Output:

- The Serial Monitor should list the quantitative brightness that falls on the LDR.
- If the room is dark then it should show a small value and if the room is bright it should show a large value.
- The values range from 0 to 1024.

Appendix:

An LDR is a component that has a (variable) resistance that changes with the light intensity that falls upon it. This allows them to be used in light sensing circuits.

The most common type of LDR has a resistance that falls with an increase in the light intensity falling upon the device (as shown in the image above). The resistance of an LDR may typically have the following resistances:

Daylight = 5000Ω

Dark = 20000000Ω

You can therefore see that there is a large variation between these figures. If you plotted this variation on a graph you would get something similar to that shown by the graph shown above.

Applications of LDRs:

There are many applications for Light Dependent Resistors. These include:

Lighting switch: The most obvious application for an LDR is to automatically turn on a light at a certain light level. An example of this could be a street light or a garden light.

Camera shutter control: LDRs can be used to control the shutter speed on a camera. The LDR would be used to measure the light intensity which then adjusts the camera shutter speed to the appropriate level.

Reference: "How an LDR (Light Dependent Resistor) Works", Kitronik, 2019. [Online]. Available: https://www.kitronik.co.uk/blog/how-an-ldr-light-dependent-resistor-works/.

Part 2 – LM35 Temperature Sensor IN Serial Monitor OUT

Equipment List:

1. 1x Arduino Nano
2. 1x USB 2.0 to USB type A connecting cable
3. 1x Breadboard
4. 1x LM35 Temperature Sensor

Schematic Diagram:

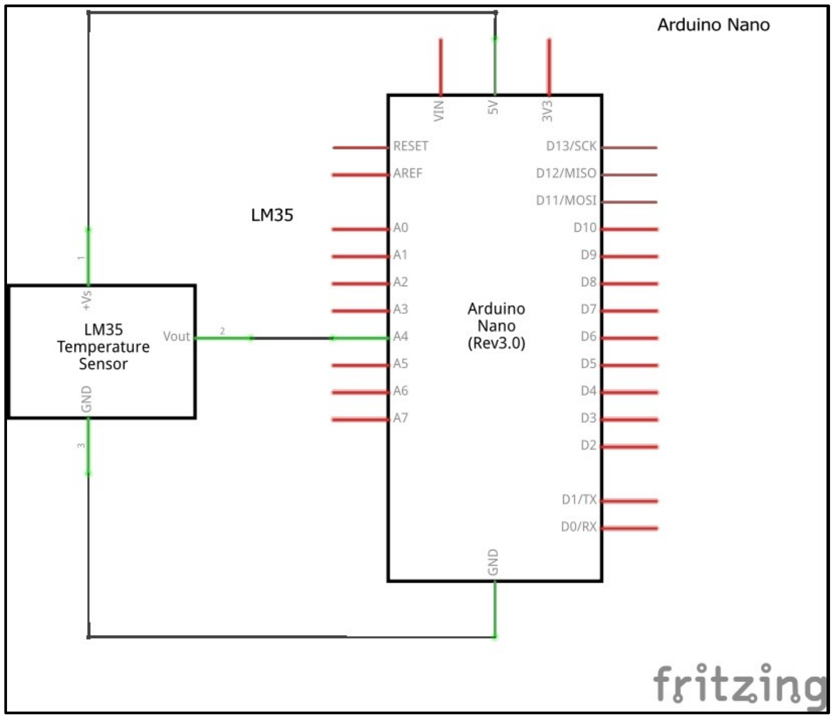

Figure 2 - Exp. 02 - Part 2 – LM35 Temperature Sensor Monitoring

Experiment Setup:

- Connect PIN 1 (VCC) of LM35 with 5v of Arduino.
- Connect PIN 2 (VOUT) of LM35 with any ANALOG (in our case it is 4) pin of Arduino.
- Connect PIN 3 (GND) of LM35 with GROUND pin of Arduino.
- Compile and upload given code to the development board.
- Open Serial Monitor of Arduino.
- Analyze the output.

Code:

```
int val;
int tempPin = 4;
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  val = analogRead(tempPin);
  float mv = ( val/1024.0)*5000;
  float cel = mv/10;
  float farh = (cel*9)/5 + 32;
  Serial.print("TEMPRATURE = ");
  Serial.print(cel);
  Serial.print("*C");
  Serial.println();
  delay(1000);
  /* uncomment this to get temperature in farenhite
  Serial.print("TEMPRATURE = ");
  Serial.print(farh);
  Serial.print("*F");
  Serial.println();
  */
}
```

Code Listing 2 – Code for LM35 interfacing

Expected Output:

- The Serial Monitor will output the current temperature of the room.
- If the LM35 is heated from body temperature then you will find a change in the readings.

Appendix:

The LM35 series are precision integrated-circuit temperature devices with an output voltage linearly-proportional to the Centigrade temperature. The LM35 device has an advantage over linear temperature sensors calibrated in Kelvin, as the user is not required to subtract a large constant voltage from the output to obtain convenient Centigrade scaling. The LM35 device does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±¾°C, over a full −55°C to 150°C temperature range. Lower cost is assured by trimming and calibration at the wafer level. The low-output impedance, linear output, and precise inherent calibration of the LM35 device makes interfacing to readout or control circuitry especially easy. The device is used with single power supplies, or with plus and minus supplies. As the LM35 device draws only 60 µA from the supply, it has very low self-heating of less than 0.1°C in still air. The LM35 device is rated to operate over a −55°C to 150°C temperature range.

Reference: "LM35 (ACTIVE) ±0.5°C Temperature Sensor with Analog Output and 30V Capability", *http://www.ti.com/*, 2019. [Online]. Available: http://www.ti.com/product/LM35.

Part 3 – DHT11 (RHT01) Temperature and Humidity Sensor IN Serial Monitor OUT

Equipment List:

1. 1x Arduino Nano
2. 1x USB 2.0 to USB type A connecting cable
3. 1x Breadboard
4. 1x DHT11 (RHT01) Temperature and Humidity Sensor
5. Optional: 1x 10kΩ Resistor for pull-up (connect as like the LDR)

Schematic Diagram:

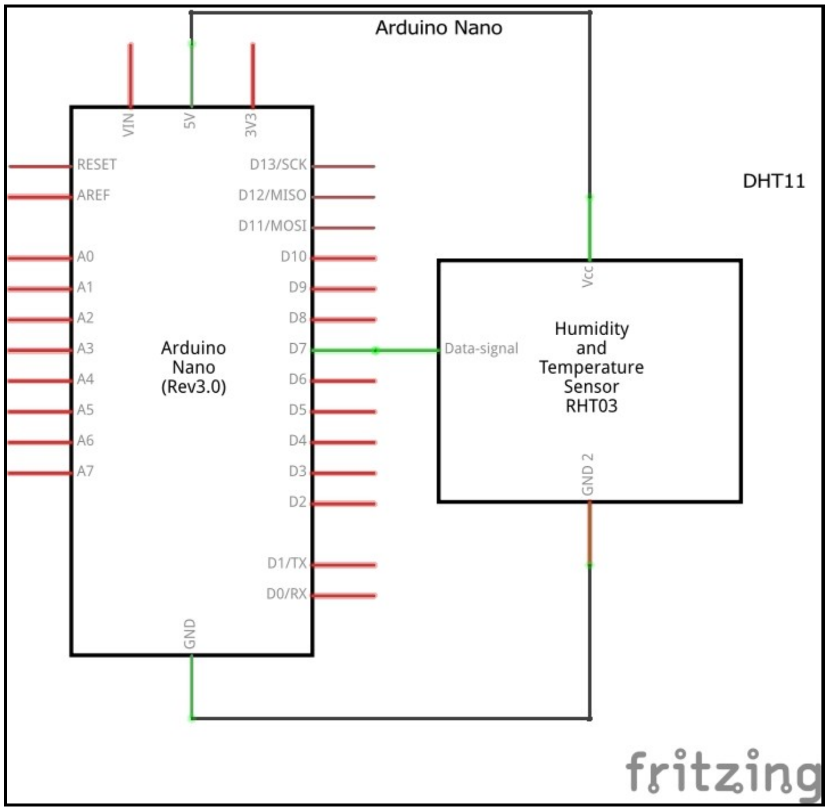

Figure 3 - Exp. 02 - Part 3 – DHT11 (RHT01) Temperature Sensor Monitoring

Experiment Setup:

- Connect PIN 1 (VCC) of DHT11 with 5v of Arduino.
- Connect PIN 2 (VOUT) of DHT11 with any ANALOG (in our case it is 3) pin of Arduino.
- Connect PIN 3/4 (GND) of DHT11 with GROUND pin of Arduino.
- Import the .zip DHT library to the Arduino IDE first.
- Compile and upload given code to the development board.
- Open Serial Monitor of Arduino.
- Analyze the output.

Code:

```
#include<dht.h>
dht DHT;
#define DHT11_PIN 3
void setup() {
  Serial.begin(9600);
}
void loop() { // READ DATA
  int chk = DHT.read11(DHT11_PIN);
  Serial.println(" Humidity " );
  Serial.println(DHT.humidity, 1);
  Serial.println(" Temparature ");
  Serial.println(DHT.temperature, 1);
  delay(2000);
}
```

Code Listing 3 – Code for DHT11 interfacing

Expected Output:

- The Serial Monitor will output the current temperature and humidity of the room.
- If the DHT11 is heated from body temperature then you will find a change in the readings.
- DHT11 is comparatively accurate than LM35.

Appendix:

The DHT11 is a basic, ultra-low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin (no analog input pins needed). It's fairly simple to use, but requires careful timing to grab data.

Refernce: "A. Industries, "DHT11 basic temperature-humidity sensor + extras", *Adafruit.com*, 2019. [Online]. Available: https://www.adafruit.com/product/386.

Tasks:

1. Connect a LDR and an LED with an Arduino and do the following:
    a. Turn the LED on if the brightness of the room decreases less than 512.
    b. Sync the brightness value of the LED in reverse with the brightness value of the room; i.e., if the room goes dark then the LED goes bright and vice versa.
2. Connect a LM35 and an LED with your Arduino and do the following:
    a. The LED should light up if the temperature increases than 28° C.
    b. The LED will change brightness according to temperature increase.
3. Connect an LDR, LM35 and DHT11 simultaneously with an Arduino and do the following:
    a. Show all values separately in the Serial Monitor.