King Fahd University of Petroleum and Minerals

College of Computer Science and Engineering

COE306: Introduction to Embedded Systems

COE306 Project

Automatic A/C Controller

By:

Osama Bujwaied

201661700

For:

Dr. Hazim Selmi

Dr. Abdulaziz Tabakh

2 May 2020

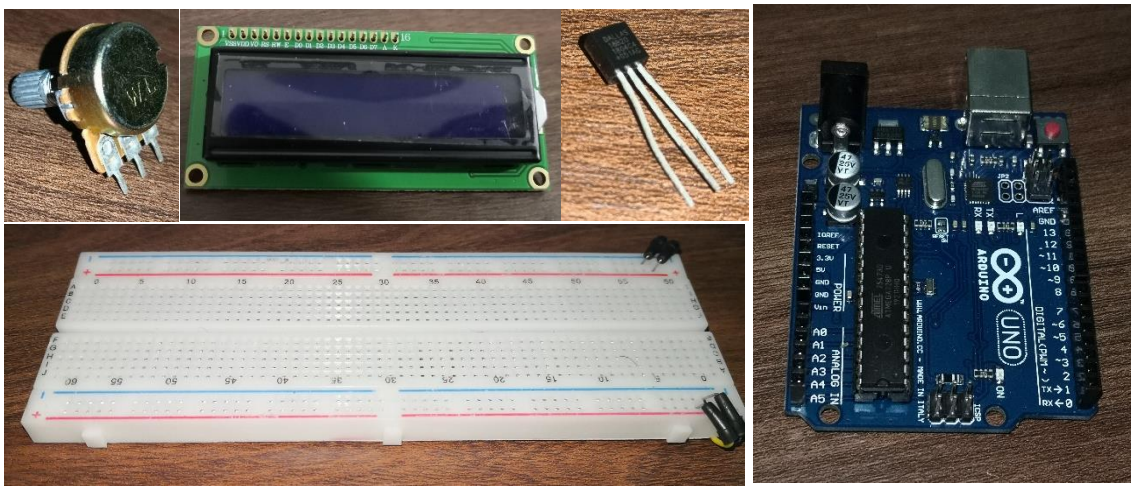
Table of Contents

1. OBJECTIVE

The objective of this report is to exhibit automatic A/C controller unit that maintain requested temperature using Arduino equipment.

2. PART LIST

- Arduino Uno R3 Device
- USB cable
- Jumper wires
- Breadboard 3
- LEDs (red, blue, yellow)
- Adjustable Potentiometer
- DS18B20 Temperature Sensor
- I2C LCD Display



3. BACKGROUND

3.1. Arduino Uno

- In this project, I used Arduino Uno R3.
- It has 2 power source pins 3.3v and 5v in addition to 3 grounds.
- Also, it has 6 pins for analogy signal.
- In addition, there is 14 digital pins with 5 of them are PWM pins.

3.2. Temperature Sensor

- I used DS18B20 Temperature Sensor model.
- It has 3 wires. Left connected to the ground, Right to the voltage source. The middle connected to both power source using resistor and to digital to read it input.
- It can read operating temperature range: -55ºC to +125ºC, then it converts it to digital value.

3.3. I2C LCD Display

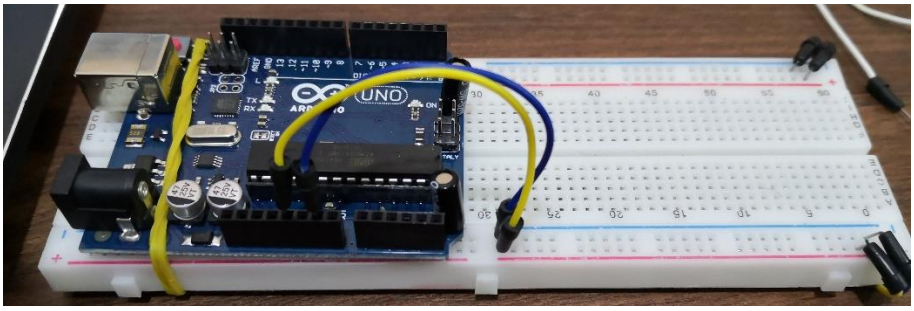
3.5 Programing (code)

- Function
  - Serial.begin(X); // Start serial communication with X speed
  - Serial.print(); // Print text or value to the console
  - analogRead (AX) // Read analogy value from analogy pin X.
  - analogWrite(X, Y)// Write Y value to PWM pin X.
  - delay(X) // Add some delay to code my X ms
  - sensors.begin(); // Start up the library for the sensors
- Library
  - OneWire
    - Used to connect temperature sensor.
  - DallasTemperature
    - Used to transform voltage temperature sensor value taken from the pin to Celsius value.
  - LiquidCrystal

4. STEPS OF PROJECT

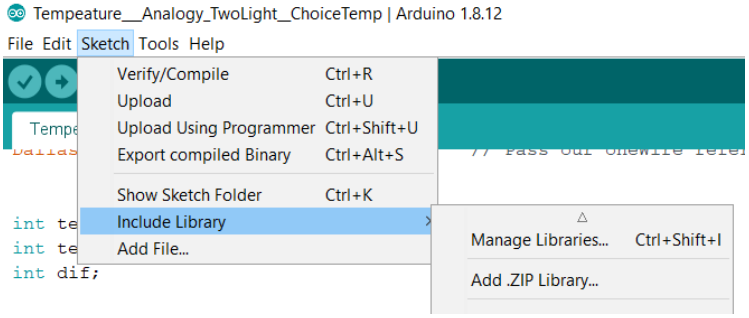
4.0. Orginize project component

- Installing Arduino uno device above breadboard using rubber.
- Connect red(+) sign to 5v source in both direction, and Connect blue(-) sign to ground source in both direction.



4.1. Connection and Programing Temperature Sensor

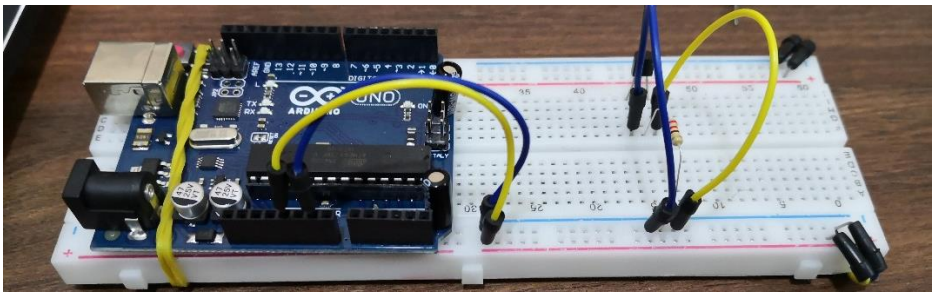
- **Code**
  - Downloading OneWire and DallasTemperature libraries from Arduino program (Sketch > Include Library > Manage Library > "Search for OneWire and DallasTemperature libraries")



  - Assigning the needed code to read the temperature from sensor

```
#include <DallasTemperature.h>
#include <OneWire.h>

#define ONE_WIRE_BUS 13                // Data wire is conntec to the Arduino digital pin 13
OneWire oneWire(ONE_WIRE_BUS);         // Setup a oneWire instance to communicate with any OneWire devices
DallasTemperature sensors(&oneWire);   // Pass our oneWire reference to Dallas Temperature sensor

void setup() {
  // put your setup code here, to run once:

  Serial.begin(9600);                  // Start serial communication for debugging purposes
  sensors.begin();                     // Start up the library
}

void loop() {
  sensors.requestTemperatures();       // Call sensors.requestTemperatures() to issue a global temperature and Requests to all devices on the bus

  Serial.print("Current celsius temperature: ");
  Serial.println(sensors.getTempCByIndex(0));
}
```

- **Device**

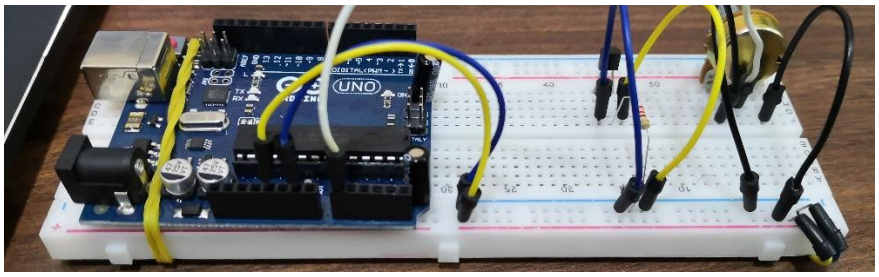


  - It has 3 wires connected as follow.
    - Left connected to the ground.
    - Right to the voltage source.
    - The middle connected to both power source using resistor and to digital pin 13.

4.2. Reading User Temperature Request by Potentiometer

- **Code**
  - Making A0 as analogy input from Potentiometer
  - After Reading the analog value (0-1023) do this equation to it get a range of value between 18-33
    - Requested temperature = (A0/64) + 18

```
temp_Choice = 18 +(analogRead(A0)/64);
Serial.print("Celsius temperature chosen = ");
Serial.println(temp_Choice);
```

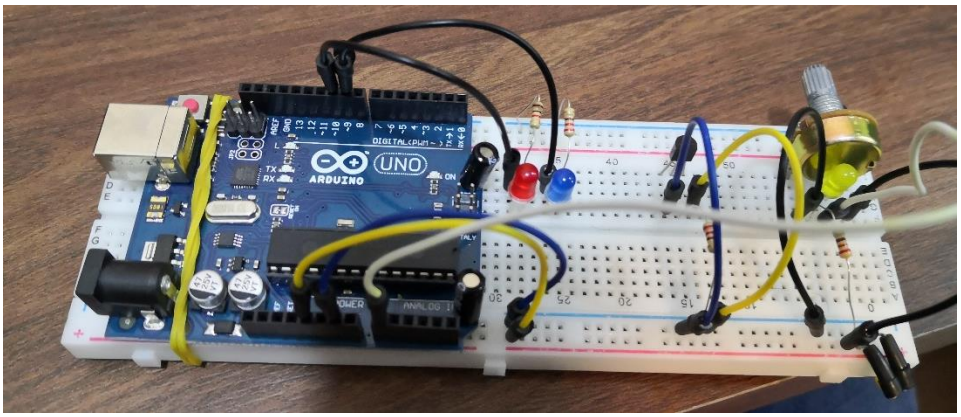
- **Device**
  - 
  - It has 3 wires connected as follow.
    - Left connected to the ground.
    - Right to the voltage source.
    - The middle connected to Analogy pin A0.

4.3. Assigning Light to Show A/C Condition

- **Code**
  - Making PWM digital pins 9 and 10 as output for the blue and the red light.
  - If reading temperature are greater than the requested temperature turn on the (Cold A/C) blue light.
  - If reading temperature are lower than the requested temperature turn on the (Hot A/C) Red light.
  - Increase the (fan speed) LED light intensity comparing to the different between requested and current room temperature

```
temp_C = sensors.getTempCByIndex(0);
if (temp_C > temp_Choice) {dif = (temp_C-temp_Choice)*20; analogWrite(9,dif);}
else {analogWrite(9,0);}

if (temp_C < temp_Choice) {dif = (temp_Choice-temp_C)*20; analogWrite(10,dif);}
else {analogWrite(10,0);}
```

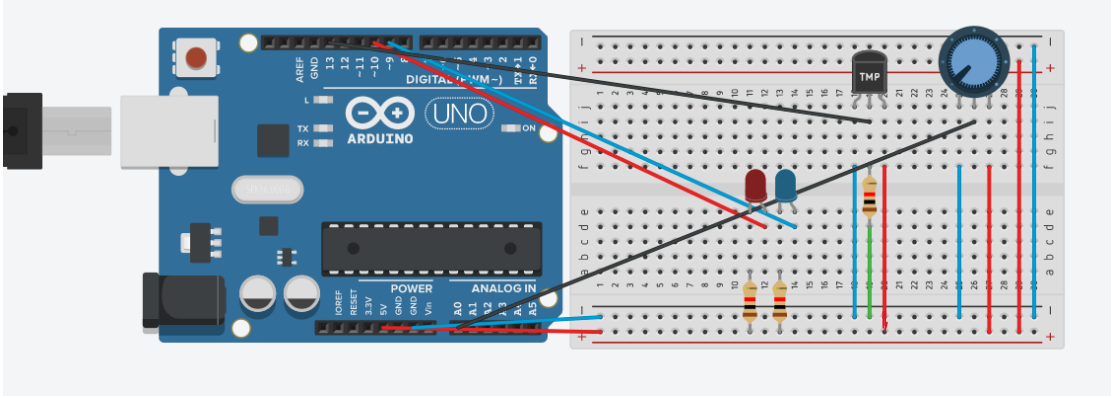
- **Device**



  - For both LED light connect (+) long wire to the digital pins 9 and 10, and the (-) short wire to the ground using a resistor.

4.4. Configure I2C LCD Display to Show Temperature Condition

4.5. Overall Design



Tempeature___Analogy_TwoLight__ChoiceTemp §

```
#include <LiquidCrystal.h>
#include <DallasTemperature.h>
#include <OneWire.h>

#define ONE_WIRE_BUS 13                 // Data wire is conntec to the Arduino digital pin 13
OneWire oneWire(ONE_WIRE_BUS);          // Setup a oneWire instance to communicate with any OneWire devices
DallasTemperature sensors(&oneWire);    // Pass our oneWire reference to Dallas Temperature sensor

int temp_C;
int temp_Choice;
int dif;

void setup(void)
{
  Serial.begin(9600);                    // Start serial communication for debugging purposes
  sensors.begin();                       // Start up the library
}

void loop(void){
  sensors.requestTemperatures();          // Call sensors.requestTemperatures() to issue a global temperature

  Serial.print("Current celsius temperature: ");
  Serial.println(sensors.getTempCByIndex(0));

  temp_Choice = 18 +(analogRead(A0)/64);
  Serial.print("Celsius temperature chosen = ");
  Serial.println(temp_Choice);

  temp_C = sensors.getTempCByIndex(0);
  if (temp_C > temp_Choice) {dif = (temp_C-temp_Choice)*20; analogWrite(9,dif);}
  else {analogWrite(9,0);}
  if (temp_C < temp_Choice) {dif = (temp_Choice-temp_C)*20; analogWrite(10,dif);}
  else {analogWrite(10,0);}

  Serial.println();
  delay(1000);
}
```