Savitribai Phule Pune University Second Year of Computer Engineering (2015 Course) 210256: Advanced Data Structures Lab		
Lab Scheme: **PR: 04 Hours/Week**	**Credit** **02**	**Examination Scheme:** **TW: 25 Marks** **PR: 25 Marks**

==========

Course Objectives:-

- Hardware platforms and operating systems commonly used in IoT systems.
- Help the students in providing a good learning environment and also work with real time problems faced in day to day life.

==========

Course Outcomes:-

- Understand IOT Application Development using Raspberry Pi/ Beagle board/ Arduino board
- Develop and modify the code for various sensor based applications using wireless sensor modules and working with a variety of modules like environmental modules.
- Make use of Cloud platform to upload and analyze any sensor data

==========

Guidelines for Student Journal

==========

- Student should submit term work in the form of journal with write-ups based on specified list of assignments.
- Practical and Oral Examination will be based on all the assignments in the lab manual
- Candidate is expected to know the theory involved in the experiment.
- The practical examination should be conducted if and only if the journal of the candidate is complete in all respects.

==========

List of Experiments

Sr. No	Experiment Title	Page No.
1	Study of Raspberry-Pi/ Beagle board/ Arduino and other microcontroller (History & Elevation)	
2	Study of different operating systems for Raspberry-Pi /Beagle board/Arduino. Understanding the process of OS installation	
3	Write an application to read temperature from the environment. If temperature crosses threshold value then it notifies with buzzer	
4	Write a program using Arduino to control LED (One or more ON/OFF). Or Blinking.	
5	Create a program so that when the user enters ‘b’ the green light blinks, ‘g’ the green light is illuminated ‘y’ the yellow light is illuminated and ‘r’ the red light is illuminated.	
6	Write a program that asks the user for a number and outputs the number squared that is entered.	
7	Write a program read the temperature sensor and send the values to the serial monitor on the computer.	
8	Write a program to control the color of the LED by turning 3 different potentiometers. One will be read for the value of Red, one for the value of Green, and one for the value of Blue	
9	Write a program so that it displays temperature in Fahrenheit as well maximum and minimum temperature it has seen.	

10	Write an application to control the operation of hardware simulated traffic signals.	

Assignment No. 9

Aim: Write a program so that it displays temperature in Fahrenheit as well maximum and minimum temperature it has seen.

● Outcome: Understanding working principle of DHT11, LM35 temperature sensor.

● Hardware Requirement: Arduino, LED, LM35

● Software Requirement: Arduino IDE

● Theory:

LM35 Temperature Sensor:

Pin Number	Pin Name	Description
1	Vcc	Input voltage is +5V for typical applications
2	Analog Out	There will be increase in 10mV for raise of every 1°C. Can range from -1V(-55°C) to 6V(150°C)
3	Ground	Ground Connected to ground of circuit

● LM35 Sensor Features:

- Minimum and Maximum Input Voltage is 35V and -2V respectively typically 5V.
- Can measure temperature ranging from -55°C to 150°C
- Output voltage is directly proportional (Linear) to temperature (i.e.) there will be a rise of
- 10mV (0.01V) for every 1°C rise in temperature.
- Drain current is less than 60uA
- Low cost temperature sensor
- Small and hence suitable for remote applications

- How to use LM35 Temperature Sensor:

LM35 is a precession Integrated circuit Temperature sensor, whose output voltage varies, based on the temperature around it. It is a small and cheap IC which can be used to measure temperature anywhere between -55°C to 150°C. It can easily be interfaced with any Microcontroller that has ADC function or any development platform like Arduino.
Power the IC by applying a regulated voltage like +5V (VS) to the input pin and connected the ground pin to the ground of the circuit. Now, you can measure the temperature in form of voltage .

If the temperature is 0°C, then the output voltage will also be 0V. There will be rise of 0.01V (10mV) for every degree Celsius rise in temperature. The voltage can converted into temperature using the below formulae.

Vout=10mv/°C *T

- LM35 Temperature Sensor Applications:
- Measuring temperature of a particular environment
- Providing thermal shut down for a circuit/component
- Monitoring Battery Temperature
- Measuring Temperatures for HVAC applications.

LM35 can also be directly connected to Arduino. The output of LM35 temperature can also be given to comparator circuit and can be used for over temperature indication or by using a simple relay can be used as a temperature controller.
Following will read the data from analog pin. Here we set pin A1.
int temp = analogRead(pinTemp);
Using float F = temp*(9/5)+32 this formula we convert temperature to Fahrenheit.

```
int Tmax = max(temp);
int Tmin=min(temp)
```

This will diplay maximum and minimum temperature.

Conclusion: Here we write a program to display temperature in Fahrenheit as well maximum and minimum temperature it has seen.

```
int pinTemp = A1;   //This is where our Output data goes
int Tmax = max(temp);
int Tmin=min(temp);

void setup() {
  Serial.begin(9600);
}
void loop() {
  int temp = analogRead(pinTemp);    //Read the analog pin
  temp = temp * 0.48828125;   // convert output (mv) to readable celcius
  Tmax=max(temp);
  Tmin=min(temp);
  Serial.print("Max Temperature is : ",Tmax);
  Serial.print("Min Temperature is : ",Tmin);
  Serial.print("Temperature: ");
  Serial.print(temp);
  Serial.println("*C");  //print the temperature status
  float F = temp*(9/5)+32
  Serial.println(F)
  delay(1000);
}
```

Assignment No. 10

Aim: Write an application to control the operation of hardware simulated traffic signals.

Components

- Arduino UNO
- Red LED
- Yellow LED
- Green LED
- Connecting wires
- Prototyping board
- Power adapter

Theory:

Introduction

In this project we will demonstrate how to make a very simple and very cool traffic light Using Arduino.

This project will involve using a little a bit of code and a very simple circuit that's great for beginners.

This simple little project uses an Arduino and some LEDs to replicate a traffic light. It uses code as an internal timer and continues to run until you cut the Arduino's power supply.

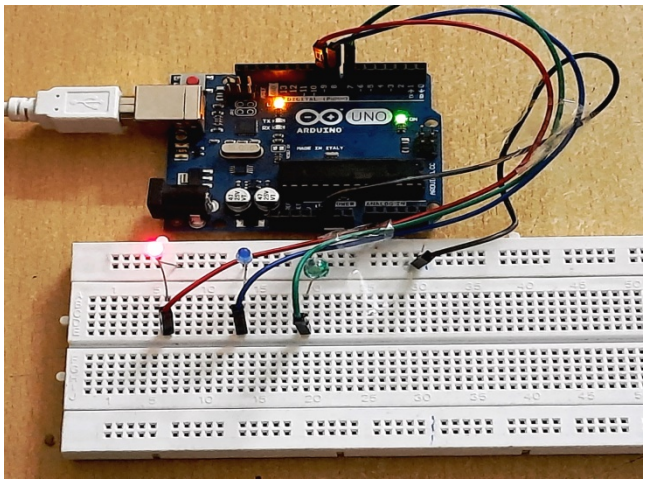
Working on Basics

The LED have been powered by Arduino UNO (Board). It contains a code which uploaded to the board. And once it simulated LED Start's blinking like a traffic light. In this 15 Second will for Red Light 6 Second for Yellow Light and 20 Second for Green Light.

Usage

There is multiple usage of the project:-

1. If you are beginner so it is a intro to Arduino
1. You can Make a real but small Traffic Light using it
1. You can also add more LED and Can change as per your usage



- Hook the **GND pin (Negative Pin)** of all led to **Pin GND** of Arduino.
- Connect Red LED **VCC Pin (Positive Pin)** to **Pin 9** of Arduino.
- Connect Yellow LED **VCC Pin (Positive Pin)** to **Pin 8** of Arduino.
- Connect Green LED **VCC Pin (Positive Pin)** to **Pin 7** of Arduino.

Conclusion: Here we are writing an application to control the operation of hardware simulated traffic signals.

```
int red = 9;
int yellow = 8;
int green = 7;

void setup(){
```

```
  pinMode(red, OUTPUT);
  pinMode(yellow, OUTPUT);
  pinMode(green,  OUTPUT);

}
void loop(){
digitalWrite(red, HIGH);
 delay(15000);
digitalWrite(red,  LOW);

  digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow,  LOW);
delay(500);

  digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow,  LOW);
delay(500);

  digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow,  LOW);
delay(500);

  digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow, LOW);
delay(500);
```

```
  digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow, LOW);
delay(500);

digitalWrite(green, HIGH);
delay(20000);
digitalWrite(green,  LOW);
//
digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow,  LOW);
delay(500);

  digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow,  LOW);
delay(500);

  digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow, LOW);
delay(500);

  digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow, LOW);
delay(500);
```

```
  digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow, LOW);
delay(500);



}
```