SNS COLLEGE OF TECHNOLOGY

Coimbatore-35

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with
'A++' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University,
Chennai

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

19ECT213- IoT SYSTEM ARCHITECTURE

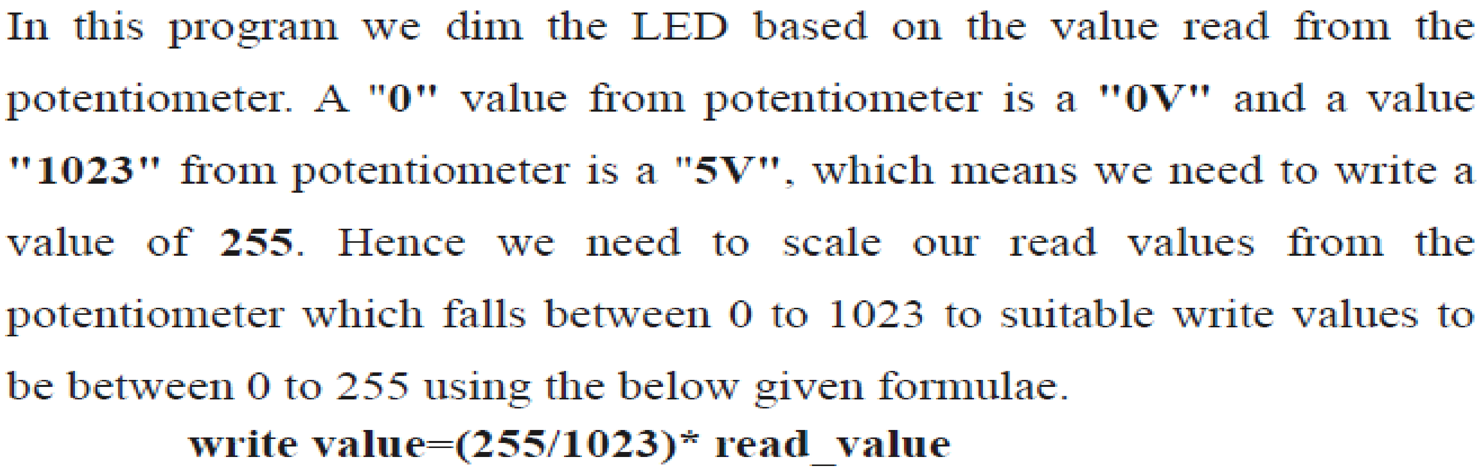
II ECE / IV SEMESTER

UNIT 2 – MICROCONTROLLER AND INTERFACING TECHNIQUES FOR IoT DEVICES

Analog Sensor Interfacing

Creating a Dimmable LED using Potentiometer

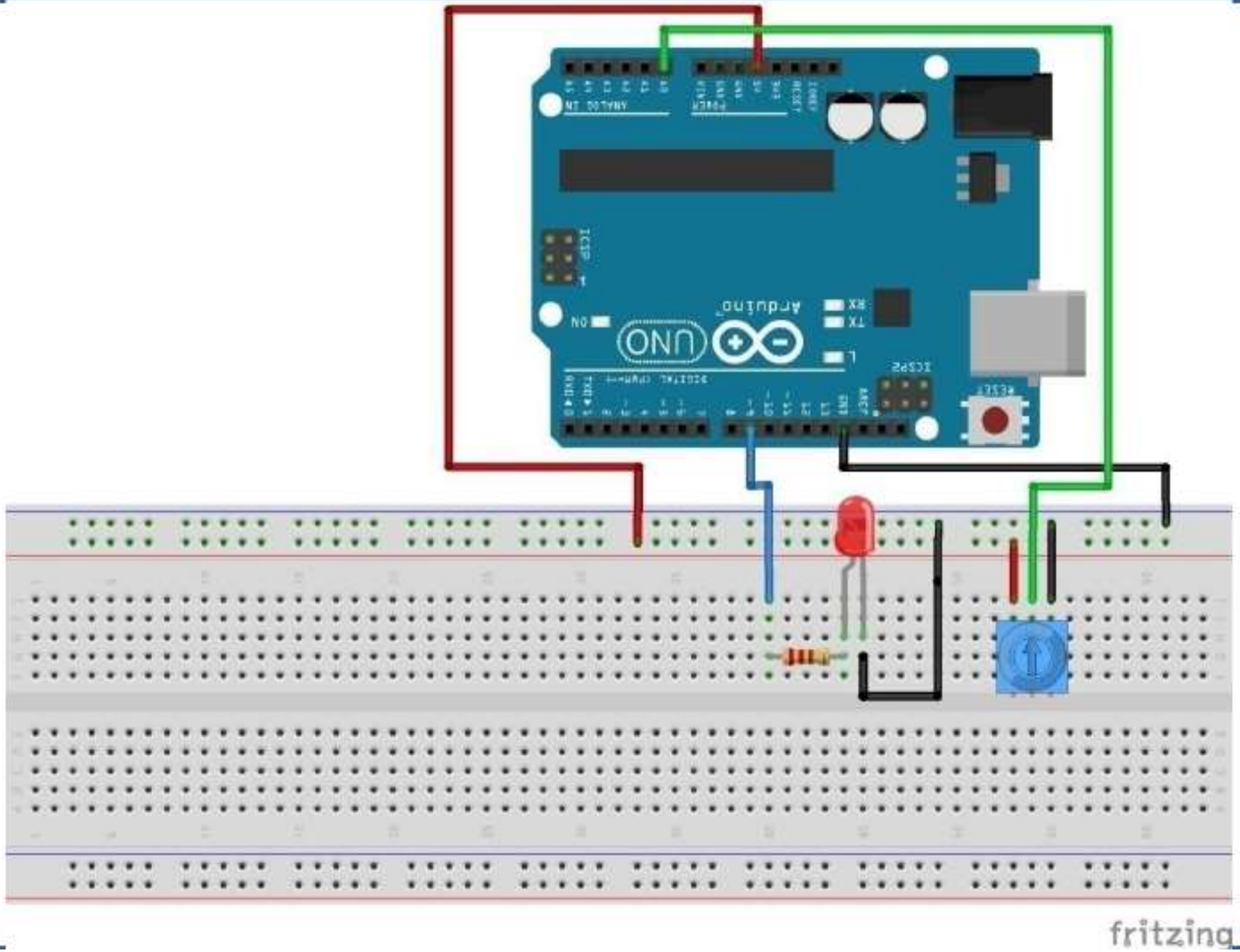
Components Required	1-LED, 220Ω resistor, 1-Potentiometer, Jumper wires, Breadboard



In this program we dim the LED based on the value read from the potentiometer. A **"0"** value from potentiometer is a **"0V"** and a value **"1023"** from potentiometer is a **"5V"**, which means we need to write a value of **255**. Hence we need to scale our read values from the potentiometer which falls between 0 to 1023 to suitable write values to be between 0 to 255 using the below given formulae.

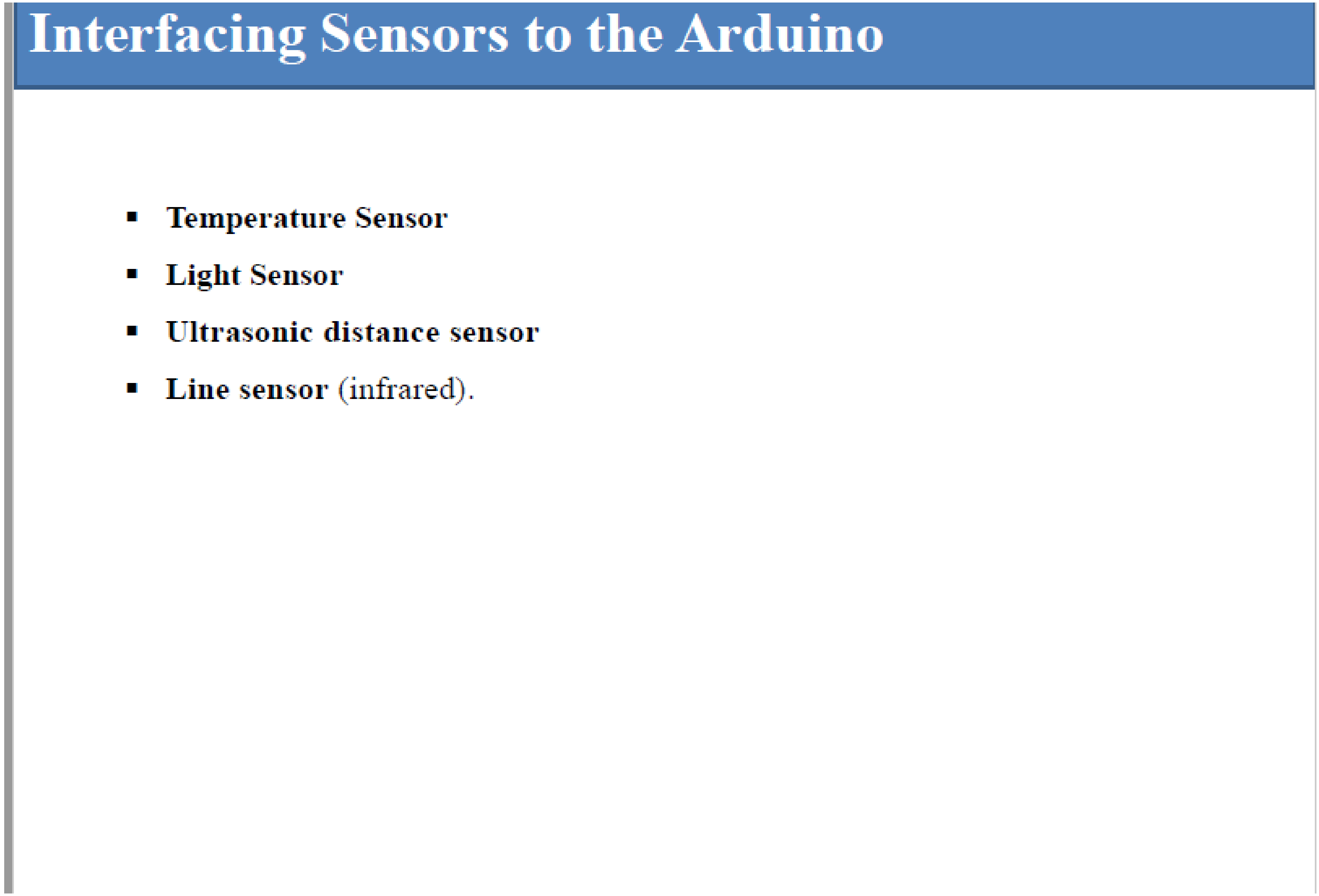
write value=(255/1023)* read_value

Creating a Dimmable LED using Potentiometer



Creating a Dimmable LED using Potentiometer

```
//Declaring the pins corresponds to an LED-to pin 9 and a Potentiometer- to
//pinA0
int pot_Pin= A0;
int LED_Pin= 9;
int read_Value;  // To store the value read by potentiometer
int write_Value; // To write the value to LED
void setup( )
{ pinMode(pot_Pin, INPUT);
  pinMode(LED_Pin, OUTPUT);
  Serial.begin(9600);      }
void loop( )
{ read_Value = analogRead(pot_Pin);  //Potentiometer reading
 write_Value = (255./1023.) * readValue; //Write value for LED is calculated
 analogWrite(LEDPin, writeValue);      //Write to the LED
 Serial.print("The writing vlues to the LED is ");  //Debugging purpose
 Serial.println(write_Value); }
```



Interfacing Sensors to the Arduino

- **Temperature Sensor**
- **Light Sensor**
- **Ultrasonic distance sensor**
- **Line sensor** (infrared).

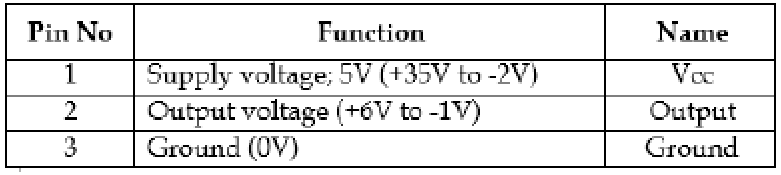
Interfacing Temperature Sensor

Components Required	Buzzer, LM35 Temperature Sensor, Jumper wires, Breadboard

LM35 Temperature Sensor:

The LM35 series are the gadgets with precision integrated circuit temperature whose yield voltage falls directly corresponding to the Centigrade temperature.

- □ Calibrated Directly in Celsius (Centigrade)
- □ Operates from 4 V to 30 V
- □ Ranges are evaluated from Full −55°C to 150°C.
- □ Suitable for Remote Applications
- □ Used in Battery Management

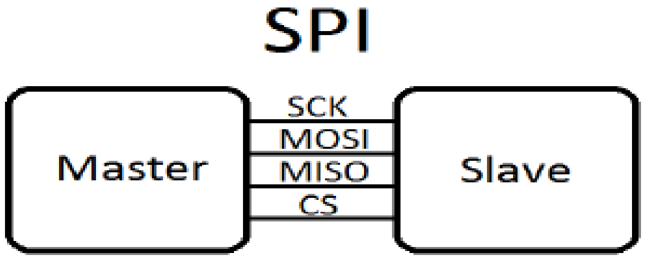


Pin No	Function	Name
1	Supply voltage; 5V (+35V to -2V)	Vcc
2	Output voltage (+6V to -1V)	Output
3	Ground (0V)	Ground

Introduction to Communications

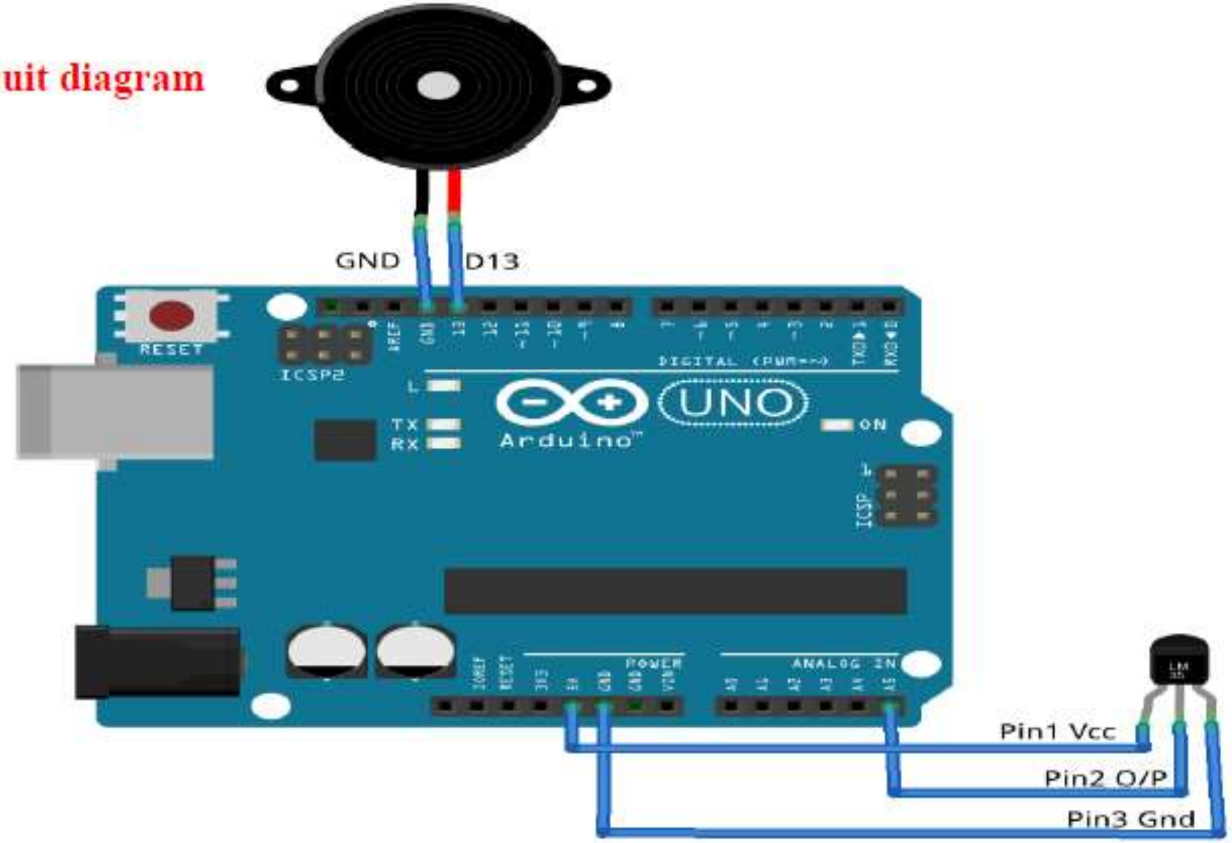
SPI communications

- When device SS pin is low, communication happens with the master, if SS pin is high device ignores the maser. This allows multiple SPI devices sharing the the same MISO, MOSI and CLK lines.
- To program a new SPI device some key points to be noted which are
  - Maximum SPI speed of the device used?
  - How data is shifted like MSB/LSB?
  - Data clock is idle when high/low.

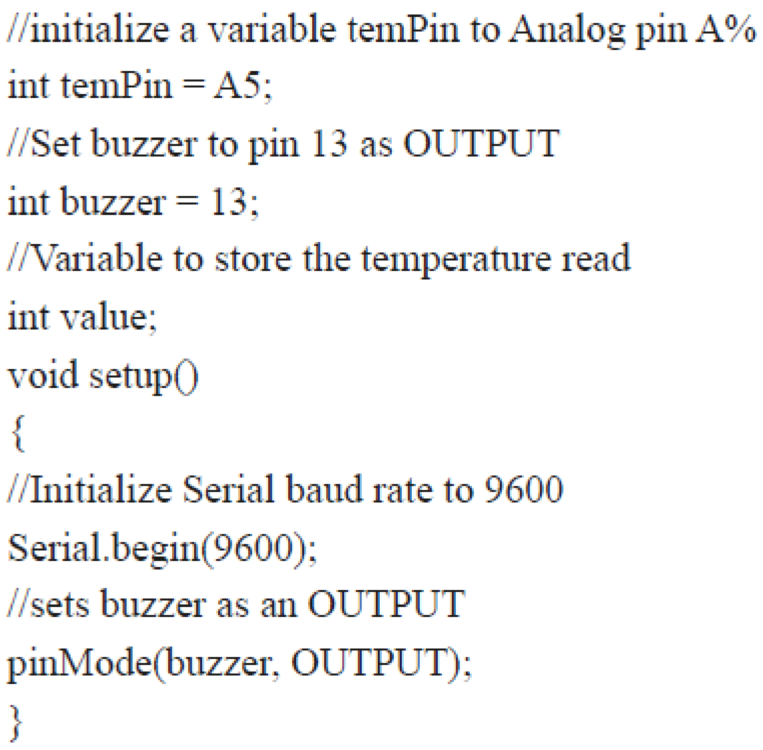


Interfacing Temperature Sensor

Circuit diagram



Interfacing Temperature Sensor



```
//initialize a variable temPin to Analog pin A%
int temPin = A5;
//Set buzzer to pin 13 as OUTPUT
int buzzer = 13;
//Variable to store the temperature read
int value;
void setup()
{
//Initialize Serial baud rate to 9600
Serial.begin(9600);
//sets buzzer as an OUTPUT
pinMode(buzzer, OUTPUT);
}
```