INTERNET OF THINGS LABORATORY

LAB MANUAL

Year	:	2019-2020
Course Code	:	BPSB10
Regulations	:	R18
Branch	:	EEE (EPS)
Class / Semester	:	M. Tech – I Sem

Prepared by
B. Navothna
Assistant Professor

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

INSTITUTE OF AERONAUTICAL ENGINEERING
(AUTONOMOUS)

Dundigal – 500 043, Hyderabad

INSTITUTE OF AERONAUTICALENGINEERING

(Autonomous)

Dundigal, Hyderabad – 500043

Department of Electrical and Electronics Engineering

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

The students of M.Tech Electrical Power Systems are prepared to:

PEOI:	Impart engineering knowledge in specific and re-equip with latest technologies to analyze, synthesize the problems in power system and multidisciplinary sectors.
PEOII:	Design, develop innovative products and services in the field of electrical power systems with the latest technology and toolset.
PEOIII:	Inculcate research attitude and life-long learning for a successful career.
PEOIV:	Attain intellectual leadership skills to cater the needs of power industry, academia, society and environment.

PROGRAMME OUTCOMES (POs)

Upon completion of M.Tech Electrical Power Systems, the students will be able to:

PO1:	Identify, formulate and solve power system related problems using advanced level computing techniques.
PO2:	Explore ideas to carry out research / investigation independently to solve practical problems through continuing education.
PO3:	Demonstrate knowledge and execute projects on contemporary issues in multidisciplinary environment.
PO4:	Ability to write and present a substantial technical report / document.
PO5:	Inculcate ethics, professionalism, multidisciplinary approach, entrepreneurial thinking and effective communication skills.
PO6:	Function effectively as an individual or a leader in a team to propagate ideas and promote teamwork.
PO 7:	Develop confidence for self-study and to engage in lifelong learning.

INDEX

ATTAINMENT OF PROGRAM OUTCOMES

Exp. No	Experiment	Program Outcomes Attained
1	Design of digital dc voltmeter and ammeter	PO1, PO2
2	Design of digital ac voltmeter and ammeter	PO1, PO2
3	Direction control of three phase induction motor	PO1, PO2, PO3
4	Design of digital frequency meter	PO1, PO2
5	Measurement of power and energy	PO1, PO2, PO3
6	Measurement of phase shift and power factor	PO1, PO2, PO3
7	Implementation of over current relay	PO1, PO2, PO3, PO4
8	Over/under voltage protection of home appliances	PO2, PO3
9	Protection of three phase induction motor	PO1, PO2, PO3
10	Traffic signal control	PO2, PO3
11	Railway gate control by stepper motors	PO2, PO3
12	Direction and Speed control of DC motor	PO2, PO3, PO4

INTERNET OF THINGS LABORATORY

OBJECTIVE

The main objective IOT applications in Electrical engineering laboratory is to know the different real time sensors used to measure the different electrical parameters and to control the different electrical devices from anywhere through IOT.

OUTCOMES

Upon the completion of **INTERNET OF THINGS LABORATORY**, the student will be able to attain the followings:

1. List out the different IOT applications and importance of IOT in present scenario.
2. List the application of Arduino and Node MCU
3. Know the different sensors available to measure the current and voltage
4. Design the digital voltmeter and ammeter for both AC and DC circuits
5. Design a digital frequency meter to measure the frequency in an AC circuit.
6. Measure the power and energy consumption in a home using Arduino
7. Measure the power factor and phase angle in an AC circuit using Arduino/Node MCU.
8. Design a system to control the traffic signals through IOT
9. Develop a system to control the direction of three phase induction motor
10. Model a system to control the railway gate using stepper motors.
11. Know the functioning of relay module and a 3-phase contactor.
12. Design a system to protect the three phase induction motor from abnormal fault conditions
13. Design a system to control the direction and speed of DC motor
14. Design a relay to protect the home appliances from over currents, under voltages and over voltages.

EXPERIMENT – 1

DESIGN OF DIGITAL DC VOLTMETER AND AMMETER

1.1 AIM:

To design a Digital DC Voltmeter and Ammeter to measure the voltage and current in DC electrical circuits using Arduino and display the values in LCD display

1.2 APPARATUS:

S. No	Name of the equipment	Quantity
1	Arduino Uno	1
2	16 x 2 LCD display	1
3	ACS712 Voltage sensor	1
4	9/12V Battery	1
5	Computer with Arduino IDE software	1
6	Center tapped transformer (230/6-0-6V)	1
7	jumper cables	As required

1.3 ELECTRICAL CIRCUIT DIAGRAM:

A) DC VOLTMETER:

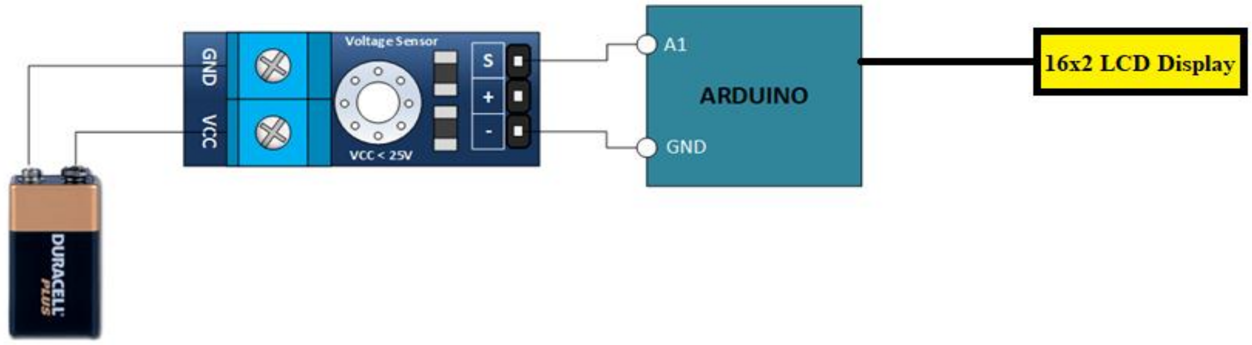

Figure.1.1 Electrical Circuit for DC Volt meter

B) DC AMMETER:

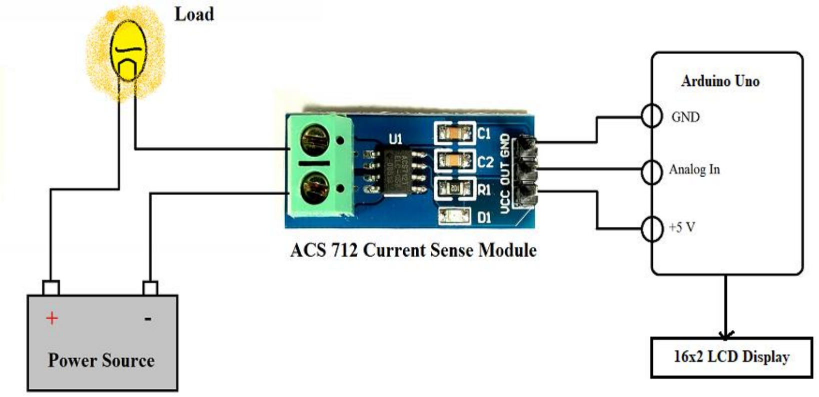

Figure: 1.2 Circuit Diagram for Digital DC ammeter

1.4 ARDUINO CODE:

A) Code for DC Voltmeter:

```
void setup()
{
Serial.begin(9600);
}
void loop() {
  int volt = analogRead(A0);// read the input
  double voltage = map(volt,0,1023, 0, 2500) + offset;// map 0-1023 to 0-2500 and add correction offset
    voltage /=100;// divide by 100 to get the decimal values
Serial.print("Voltage: ");
Serial.print(voltage);//print the voltge
Serial.println("V");
delay(500);
 }
```

B) Code for DC Voltmeter:

```
void setup() {
Serial.begin(9600);
}
void loop() {
float average = 0;
for(int i = 0; i< 1000; i++) {
average = average + (.0264 * analogRead(A0) -13.51) / 1000;
//5A mode, if 20A or 30A mode, need to modify this formula to
//(.19 * analogRead(A0) -25) for 20A mode and
//(.044 * analogRead(A0) -3.78) for 30A mode
delay(1);
  }
Serial.println(average);
}
```

1.5 PROCEDURE

A) Procedure for DC Voltmeter:

1. Make the connection as per the electrical circuit diagram
2. Open the Arduino IDE in computer and write the program.
3. Compile the program for any errors and upload it to the Arduino.
4. Observe the voltage on LCD display.
5. If a potentiometer is used between the battery and sensor to change the voltage value.
6. Observe the changed voltage is updated on LCD display.

B) Procedure for DC Ammeter:

1. Make the connection as per the electrical circuit diagram
2. Open the Arduino IDE in computer and write the program.
3. Compile the program for any errors and upload it to the Arduino.
4. Observe the current on LCD display.
5. If the load is increased by adding a series resistor to LED then current increases and is updated in LCD display.

1.6 PRE LAB VIVA QUESTIONS

1. Define voltage.
2. Define ohms law.
3. Define KVL and KCL
4. What is a current controlled device?
5. Define Resistance.

1. 7 POST LAB VIVA QUESTIONS

1. What is the sensor used to measure the voltage?
2. What is meant by ACS?
3. What is electrical circuit in ACS712 sensor?
4. How many lines can be printed on LCD display?
5. How many letters will be printed on LCD display?

EXPERIMENT – 2

DESIGN OF DIGITAL AC VOLTMETER AND AMMETER

2.1 AIM:

To design a Digital AC Voltmeter and Ammeter to measure the voltage and current in AC electrical circuits using Arduino and display the values in LCD display.

2.2 APPARATUS:

S No	Name of the equipment	Quantity
1	Arduino Uno	1
2	16 x 2 LCD display	1
3	ACS712 Voltage sensor	1
4	Center tapped transformer (230/6-0-6V)	1
5	Computer with Arduino IDE software	1
6	Jumper cables	As required

2.3 ELECTRICAL CIRCUIT DIAGRAM:

A) AC VOLTMETER:

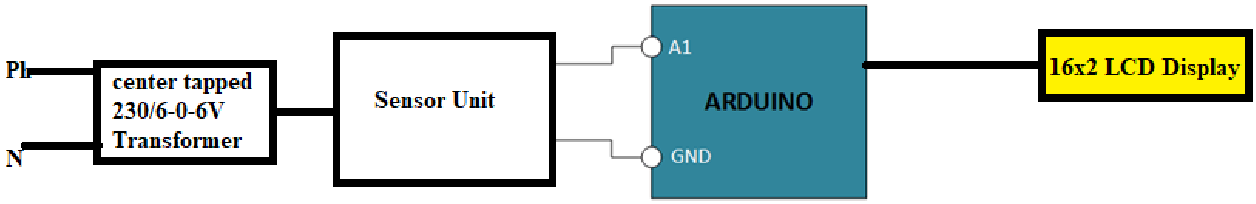

Figure: 2.1Circuit Diagram for Digital AC voltmeter

B) AC AMMETER:

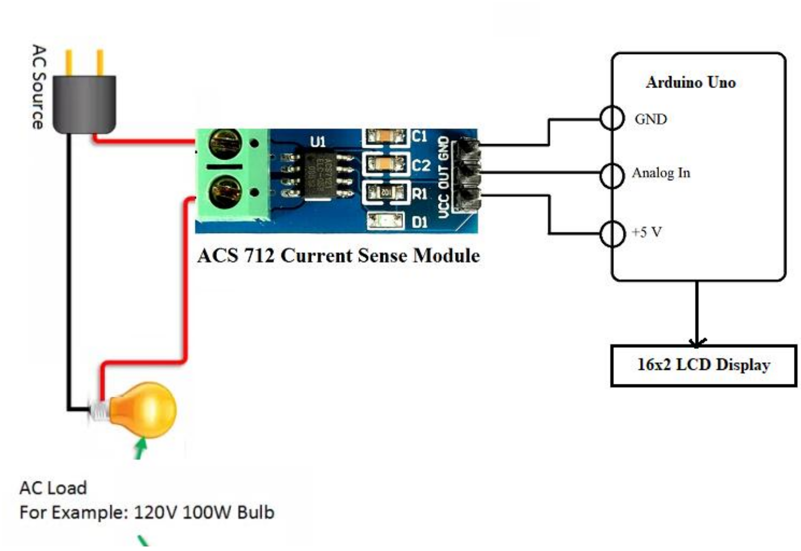

Figure: 2.2 Circuit Diagram for Digital AC ammeter

2.4 Arduino Code

A) Code for AC Voltmeter:

```
int m;// initialise variable m
float n;//initialise variable n
void setup()
{
  pinMode(A0,INPUT); // set pin a0 as input pin
  Serial.begin(9600);// begin serial communication between arduino and pc
}
void loop()
{
  m=analogRead(A0);// read analog values from pin A0 across capacitor
   n=(m* .304177);// converts analog value(x) into input ac supply value
   Serial.print("   analaog input  " ) ; // specify name to the corresponding value to be printed
   Serial.print(m) ; // print input analog value on serial monitor
   Serial.print(" ac voltage  ") ; // specify name to the corresponding value to be printed
   Serial.print(n) ; // prints the ac value on Serial monitor
   Serial.println();
}
```

B) Code for AC Ammeter:

```
#define CURRENT_SENSOR A0  // Define Analog input pin that sensor is attached
float amplitude_current;      // Float amplitude current
float effective_value;       // Float effective current
void setup()
{
    Serial.begin(9600);
    pins_init();
}
void loop()
{
    int sensor_max;
    sensor_max = getMaxValue();
    Serial.print("sensor_max = ");
    Serial.println(sensor_max);
    //the VCC on the Arduino interface of the sensor is 5v
    amplitude_current=(float)(sensor_max-512)/1024*5/185*1000000; // for 5A mode,you need to
modify this with 20 A and 30A mode;
    effective_value=amplitude_current/1.414;
    //for minimum current=1/1024*5/185*1000000/1.414=18.7(mA)
    //Only sinusoidal alternating current
    Serial.println("The amplitude of the current is(in mA)");
    Serial.println(amplitude_current,1);
    //Only one number after the decimal point
    Serial.println("The effective value of the current is(in mA)");
    Serial.println(effective_value,1);
}
void pins_init()
{
    pinMode(CURRENT_SENSOR, INPUT);
}
/*Function: Sample for 1000ms and get the maximum value from the S pin*/
```

```
int getMaxValue()
{
   int sensorValue;    //value read from the sensor
   int sensorMax = 0;
   uint32_t start_time = millis();
   while((millis()-start_time) < 1000) //sample for 1000ms
   {
   sensorValue = analogRead(CURRENT_SENSOR);
   if (sensorValue>sensorMax)
       {
  /*record the maximum sensor value*/
  sensorMax = sensorValue;
       }
   }
   return sensorMax;
}
```

2.5 PROCEDURE

A) Procedure for AC Voltmeter:

1. Make the connection as per the electrical circuit diagram
2. Open the Arduino IDE in computer and write the program.
3. Compile the program for any errors and upload it to the Arduino.
4. Observe the voltage on LCD display.
5. If a potentiometer is used between the battery and sensor to change the voltage value.
6. Observe the changed voltage is updated on LCD display.

B) Procedure for AC Ammeter:

1. Make the connection as per the electrical circuit diagram
2. Open the Arduino IDE in computer and write the program.
3. Compile the program for any errors and upload it to the Arduino.
4. Observe the current on LCD display.
5. If the load is increased by adding a series resistor to LED then current increases and is updated in LCD display.

2.6 RESULT:

2.7 PRE LAB VIVA QUESTIONS:

1. What is the principle of transformer?
2. Does Arduino take the analog values directly? Why?
3. What is the processor used in Arduino?
4. What is the name of the cable used to upload the Arduino code?
5. What is meant by LCD? Differentiate LCD from LED display.

2.8 POST LAB VIVA QUESTIONS:

1. What is the need of sensor?
2. What is the function of center tapped transformer in this experiment?
3. What is the difference between an AC Digital voltmeter and DC digital voltmeter?
4. What is the term “anlogRead” in the code?
5. How to control the brightness of LCD display?

EXPERIMENT - 3

CONTROL THE DIRECTION OF THREE PHASE INDUCTION MOTOR

3.1 AIM:

To design a system to control the direction of three phase induction motor through IOT.

3.2 APPARATUS:

S No	Name of the equipment	Quantity
1	Node MCU	1
2	16 x 2 LCD display	1
3	Contactors	3
4	Three phase induction motor	1
5	Relay module	2
6	Computer with Arduino IDE software	1
7	jumper cables	As required

3.3 ELECTRICAL CIRCUIT DIAGRAM:

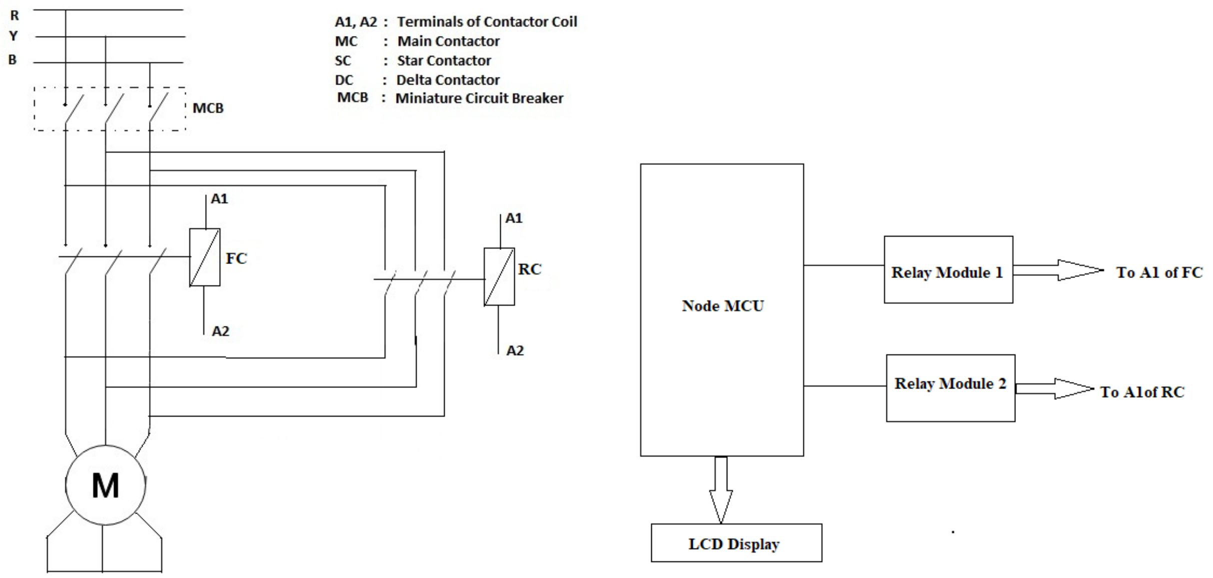

Figure: 3.1Circuit Diagram for Digital AC ammeter.

3.4 Arduino Code

```
/*
 * DDS Sine Generator mit ATMEGS 168
 * Timer2 generates the  31250 KHz Clock Interrupt
 *
 * KHM 2009 /  MartinNawrath
 * KunsthochschulefuerMedienKoeln
 * Academy of Media Arts Cologne
 */
#include "avr/pgmspace.h"
#define cbi(sfr, bit) (_SFR_BYTE(sfr) &= ~_BV(bit))
#define sbi(sfr, bit) (_SFR_BYTE(sfr) |= _BV(bit))
int ledPin = 13;                 // LED pin 7
int testPin = 7;
int t2Pin = 6;
byte bb;
double dfreq;
// const double refclk=31372.549;  // =16MHz / 510
const double refclk=31376.6;      // measured
// variables used inside interrupt service declared as voilatile
volatile byte icnt;              // var inside interrupt
volatile byte icnt1;             // var inside interrupt
volatile byte c4ms;              // counter incremented all 4ms
volatile unsigned long phaccu;   // pahse accumulator
volatile unsigned long tword_m;  // dds tuning word m
void setup()
{
pinMode(ledPin, OUTPUT);      // sets the digital pin as output
Serial.begin(115200);        // connect to the serial port
Serial.println("DDS Test");
pinMode(6, OUTPUT);      // sets the digital pin as output
pinMode(7, OUTPUT);      // sets the digital pin as output
pinMode(11, OUTPUT);     // pin11= PWM  output / frequency output
  Setup_timer2();
  // disable interrupts to avoid timing distortion
cbi (TIMSK0,TOIE0);              // disable Timer0 !!! delay() is now not available
sbi (TIMSK2,TOIE2);              // enable Timer2 Interrupt
dfreq=1000.0;                    // initial output frequency = 1000.o Hz
tword_m=pow(2,32)*dfreq/refclk;  // calulate DDS new tuning word
}
void loop()
{
while(1) {
     if (c4ms > 250) {                 // timer / wait fou a full second
      c4ms=0;
dfreq=analogRead(0);             // read Poti on analog pin 0 to adjust output frequency from 0..1023 Hz
cbi (TIMSK2,TOIE2);              // disble Timer2 Interrupt
tword_m=pow(2,32)*dfreq/refclk;  // calulate DDS new tuning word
sbi (TIMSK2,TOIE2);              // enable Timer2 Interrupt
Serial.print(dfreq);
Serial.print("  ");
Serial.println(tword_m);
    }
sbi(PORTD,6); // Test / set PORTD,7 high to observe timing with a scope
cbi(PORTD,6); // Test /reset PORTD,7 high to observe timing with a scope
  }
 }
// set prscaler to 1, PWM mode to phase correct PWM,  16000000/510 = 31372.55 Hz clock
void Setup_timer2() {
// Timer2 Clock Prescalerto : 1
sbi (TCCR2B, CS20);
cbi (TCCR2B, CS21);
cbi (TCCR2B, CS22);
  // Timer2 PWM Mode set to Phase Correct PWM
cbi (TCCR2A, COM2A0);  // clear Compare Match
sbi (TCCR2A, COM2A1);
sbi (TCCR2A, WGM20);  // Mode 1  / Phase Correct PWM
cbi (TCCR2A, WGM21);
cbi (TCCR2B, WGM22);
}
// Timer2 Interrupt Service at 31372,550 KHz = 32uSec
// this is the timebase REFCLOCK for the DDS generator
// FOUT = (M (REFCLK)) / (2 exp 32)
// runtime : 8 microseconds ( inclusive push and pop)
ISR(TIMER2_OVF_vect) {
sbi(PORTD,7);          // Test / set PORTD,7 high to observe timing with a oscope
phaccu=phaccu+tword_m; // soft DDS, phase accu with 32 bits
icnt=phaccu>>24;     // use upper 8 bits for phase accu as frequency information
                       // read value fron ROM sine table and send to PWM DAC
  OCR2A=pgm_read_byte_near(sine256 + icnt);
if(icnt1++ == 125) {  // increment variable c4ms all 4 milliseconds
    c4ms++;
    icnt1=0;
   }
cbi(PORTD,7);           // reset PORTD,7
}
```

3.5 PROCEDURE

1. Make the connection as per the electrical circuit diagram
2. Open the Arduino IDE in computer and write the program.
3. Compile the program for any errors and upload it to the Arduino.
4. Observe the direction of motor on LCD display.
5. The status of the motor whether is rotating in forward or reverse direction an also be displayed on a web link through IOT.

3.6 RESULT:

3.7 PRE LAB VIVA QUESTIONS:

1. What is a 3 pole contactor?
2. What is the function of relay module?
3. How the direction of three phase induction motor can be reversed?
4. What is the principle of induction motor?
5. What are the types of induction motors?

3.8 POST LAB VIVA QUESTIONS:

1. Why node MCU in this experiment? Why not Arduino?
2. What is the difference between node MCU and Arduino?
3. Which IDE is used for writing the node MCU program?
4. Can we connect the phase voltage directly to node MCU? Why?
5. Which induction motor is having high starting torque?

EXPERIMENT – 4

DESIGN OF DIGITAL FREQUENCY METER

4.1 AIM:

To design a Digital frequency meter to measure the frequency in any AC electrical circuit using Arduino and display the values in LCD display.

4.2 APPARATUS:

S. No	Name of the equipment	Quantity
1	Arduino Uno	1
2	16 x 2 LCD display	1
3	Diode (IN4007)	1
4	Resistor (47K/1W)	1
5	Opto-coupler	1
6	Computer with Arduino IDE software	1
7	jumper cables	As required

4.3 ELECTRICAL CIRCUIT DIAGRAM:

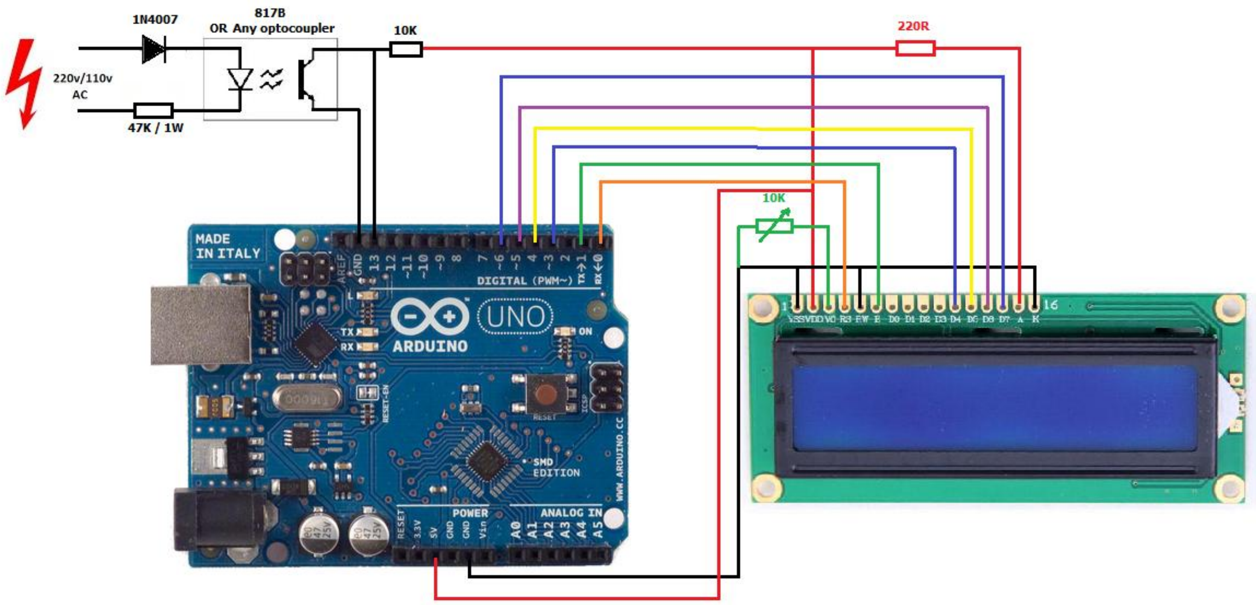

Figure: 4.1 Circuit Diagram for Digital frequency meter.

4.4 ARDUINO CODE

```
#include <LiquidCrystal.h>
int input=13;
int high_time;
int low_time;
float time_period;
float freq;
float frequency;
LiquidCrystallcd(0, 1, 3, 4, 5, 6);
void setup()
{
pinMode(input,INPUT);
lcd.begin(16, 2);
}
void loop()
{
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Frequency Meter");
high_time=pulseIn(input,HIGH);
low_time=pulseIn(input,LOW);
time_period=high_time+low_time;
time_period=time_period/1000;
frequency=1000/time_period;
lcd.setCursor(0,1);
lcd.print(frequency);
lcd.print(" Hz");
delay(500);
}
```

4.5 PROCEDURE

1. Make the connection as per the electrical circuit diagram
2. Open the Arduino IDE in computer and write the program.
3. Compile the program for any errors and upload it to the Arduino.
4. Observe the frequency on LCD display.

4.6 RESULT:

4.7 PRE LAB VIVA QUESTIONS:

1. Define frequency.
2. What is time period?
3. What is the frequency in DC circuits?
4. What is the frequency of India and USA?
5. How many cycles are there in 50Hz supply?

4.8 POST LAB VIVA QUESTIONS:

1. What is the type of Arduino used in this experiment?
2. What is the microcontroller used in Arduino uno?
3. What is an opto-coupler?
4. What is the function of diode in this experiment?
5. What is the power rating of resistor used in this experiment?

EXPERIMENT – 5

MEASUREMENT OF POWER AND ENERGY

5.1 AIM:

To measure the power and energy in electrical circuit using Arduino and display the values in LCD display.

5.2 APPARATUS:

S No	Name of the Equipment	Quantity
1	Arduino Uno	1
2	16 x 2 LCD display	1
3	Diode (IN4007)	1
4	Resistor (47K/1W)	1
5	Opto-coupler	1
6	Computer with Arduino IDE software	1
7	jumper cables	As required

5.3 ELECTRICAL CIRCUIT DIAGRAM:

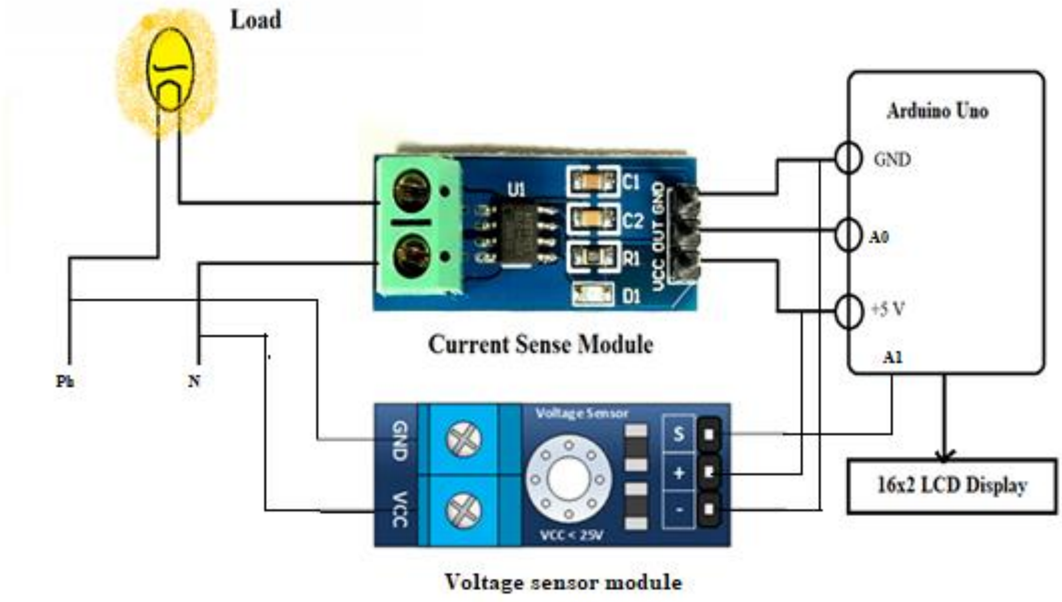

Figure: 5.1 Circuit Diagram for measurement of power and energy.

5.4 ARDUINO CODE

```
#include <LiquidCrystal.h>
int currentPin = 1; double kilos = 0;
int peakPower = 0;
LiquidCrystallcd(8, 9, 4, 5, 6, 7);
void setup()
{
```

```
lcd.begin(16,2);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Running");
}
void loop()
{
  int current = 0;
  int maxCurrent = 0;
  int minCurrent = 1000;
  for (int i=0 ;i<=200 ; i++)  //Monitors and logs the current input for 200 cycles to determine max and
min current
  {
current = analogRead(currentPin);  //Reads current input and records maximum and minimum current
if(current >= maxCurrent)
maxCurrent = current;
else if(current <= minCurrent)
minCurrent = current;
  }
  if (maxCurrent<= 517)
  {
maxCurrent = 516;
  }
  double RMSCurrent = ((maxCurrent - 516)*0.707)/11.8337;     //Calculates RMS current based on
maximum value
  int RMSPower = 220*RMSCurrent;    //Calculates RMS Power Assuming Voltage 220VAC, change
to 110VAC accordingly
  if (RMSPower>peakPower)
  {
peakPower = RMSPower;
  }
  kilos = kilos + (RMSPower * (2.05/60/60/1000));    //Calculate kilowatt hours used
  delay (2000);
lcd.clear();
lcd.setCursor(0,0);           // Displays all current data
lcd.print(RMSCurrent);
lcd.print("A");
lcd.setCursor(10,0);
lcd.print(RMSPower);
lcd.print("W");
lcd.setCursor(0,1);
lcd.print(kilos);
lcd.print("kWh");
lcd.setCursor(10,1);
lcd.print(peakPower);
lcd.print("W");
}
```

5.5 PROCEDURE

1. Make the connection as per the electrical circuit diagram
2. Open the Arduino IDE in computer and write the program.
3. Compile the program for any errors and upload it to the Arduino.
4. Observe the Power, Energy consumed by the load on LCD display.

5.6 RESULT:

5.7 PRE LAB VIVA QUESTIONS:

1. Define power.
2. Define Energy?
3. What is apparent power?
4. What is reactive power?
5. Differentiate active, reactive powers.

5.8 POST LAB VIVA QUESTIONS:

1. What is the type of Arduino used in this experiment?
2. What is the microcontroller used in Arduino uno?
3. Can we measure the DC power with this experiment?
4. What are the units of energy?
5. How the time can be calculated for measuring energy in Arduino?

EXPERIMENT -6

MEASUREMENT OF PHASE SHIFT AND POWER FACTOR

6.1 AIM:

To measure the phase shift and power factor in an electrical circuit for different loads using Arduino and display the value in LCD display.

6.2 APPARATUS:

S No	Name of the equipment	Quantity
1	Arduino Uno	1
2	16 x 2 LCD display	1
3	Voltage sensor	1
4	Current sensor	1
5	Opto-coupler (4N35)	2
6	Computer with Arduino IDE software	1
7	LM358 comparator	2
8	7845 XOR gate	1
9	Jumper cables	As required

6.3 ELECTRICAL CIRCUIT DIAGRAM:

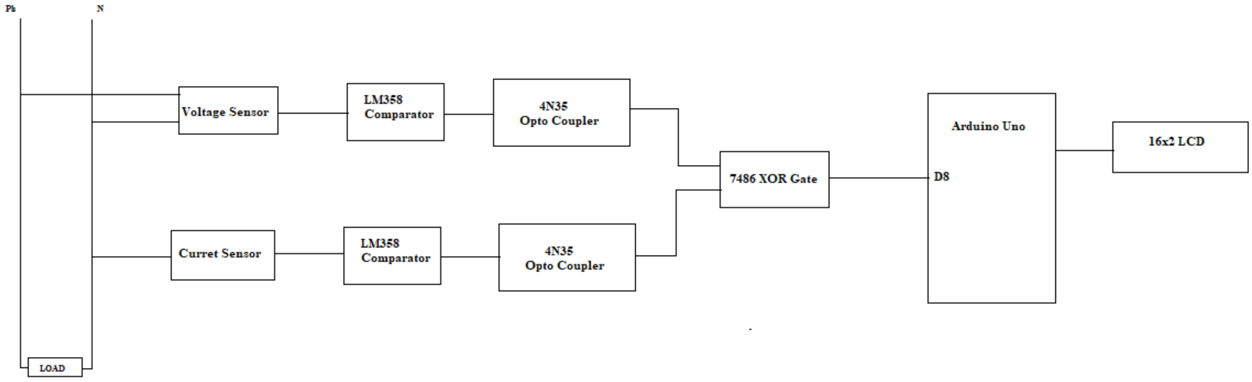

Figure: 6.1Circuit Diagram for measurement of power and energy.

6.4 ARDUINO CODE

```
#include<LiquidCrystal.h>
#include<math.h>
float pw=0;
float pf=0;
float ph=0;
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystallcd(rs, en, d4, d5, d6, d7);
void setup()
```

```
{
lcd.begin(16,2);
Serial.begin(9600);
}
void loop()
{
pw=pulseIn(8,HIGH);
ph=360*50*(pw/1000000);
pf=cos(ph*3.1415/180);
lcd.setCursor(0,0);
//lcd.print('Phase angle=');
lcd.print(ph);
lcd.setCursor(0,1);
//lcd.print('power factor=');
lcd.print(pf);
delay(100);
}
```

6.5 PROCEDURE

1. Make the connection as per the electrical circuit diagram
2. Open the Arduino IDE in computer and write the program.
3. Compile the program for any errors and upload it to the Arduino.
4. Observe the Phase angle and power factor on LCD display.

6.6 RESULT:

6.7 PRE LAB VIVA QUESTIONS:

1. What is the phase shift?
2. What is power factor?
3. How to improve the power factor?
4. What are the effects of decreasing power factor?
5. What should be done to improve the power factor?

6.8 POST LAB VIVA QUESTIONS:

1. What is the function of L358?
2. Why opto-coupler is needed in this experiment?
3. What is the function of XOR gate?
4. What is the function of “pulseIn” command?
5. What is a zero crossing detector?

EXPERIMENT – 7

IMPLEMENTATION OF OVER CURRENT RELAY

7.1 AIM:

To design an over current relay for distribution system and displaying the tripping status of the relay in substation through IOT

7.2 APPARATUS:

S. No	Name of the equipment	Quantity
1	Node MCU	1
2	16 x 2 LCD display	1
3	Current sensor/CT	1
4	Resistive load (1KW)	1
5	Relay module	1
6	Computer with Arduino IDE software	1
7	jumper cables	As required

7.3 ELECTRICAL CIRCUIT DIAGRAM:

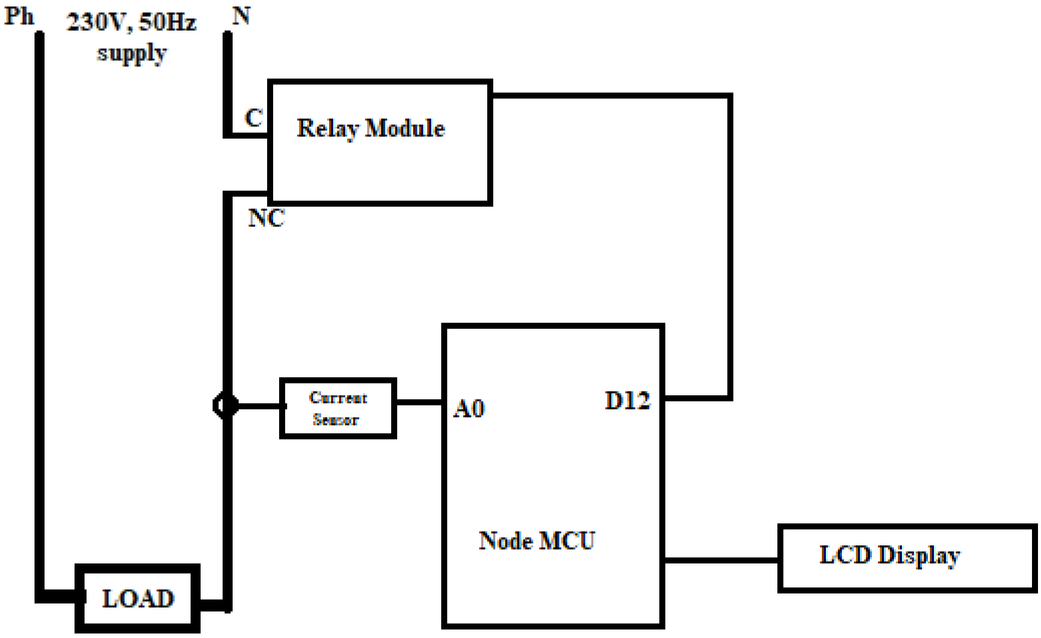

Figure: 7.1Circuit Diagram for over current protection.

7.4 ARDUINO CODE

```
#define CURRENT_SENSOR A0  // Define Analog input pin that sensor is attached
float amplitude_current;      // Float amplitude current
float effective_value;       // Float effective current
void setup()
{
    Serial.begin(9600);
    pins_init();
}
void loop()
{
    int current =0, currentTh=5;
    current = anloagRead(0);
    Current =current*(5/1023)-2.5;
    if Current >= currentTh
{
lcd.print('Over current detection');
digitalWrite(12,HIGH);
lcd.print('Relay Tripped');
}
}
```

7.5 PROCEDURE

1. Make the connection as per the electrical circuit diagram
2. Open the Arduino IDE in computer and write the program.
3. Compile the program for any errors and upload it to the Arduino.
4. Observe the status of the relay on the LCD display and also on the web-link through IOT.

7.6 RESULT:

7.7 PRE LAB VIVA QUESTIONS:

1. What is over current?
2. What are the reasons for over current?
3. What are the effects of over current?
4. What is a current transformer?
5. What is a relay?

7.8 POST LAB VIVA QUESTIONS:

15. What is the difference between a relay and circuit breaker?
16. Draw the internal circuit of relay module.
17. What is the type of sensor used in this experiment?
18. What is the output of a current sensor?
19. What is the firmware used in node MCU?

EXPERIMENT – 8

OVER/UNDER VOLTAGE PROTECTION OF HOME APPLIANCES

8.1 AIM:

Design a system to protect home appliances from over and under voltages using Arduino.

8.2 APPARATUS:

S. No	Name of the equipment	Quantity
1	Node MCU	1
2	16 x 2 LCD display	1
3	Voltage sensors	1
4	Relay module	1
5	Computer with Arduino IDE software	1
6	jumper cables	As required

8.3 ELECTRICAL CIRCUIT DIAGRAM:

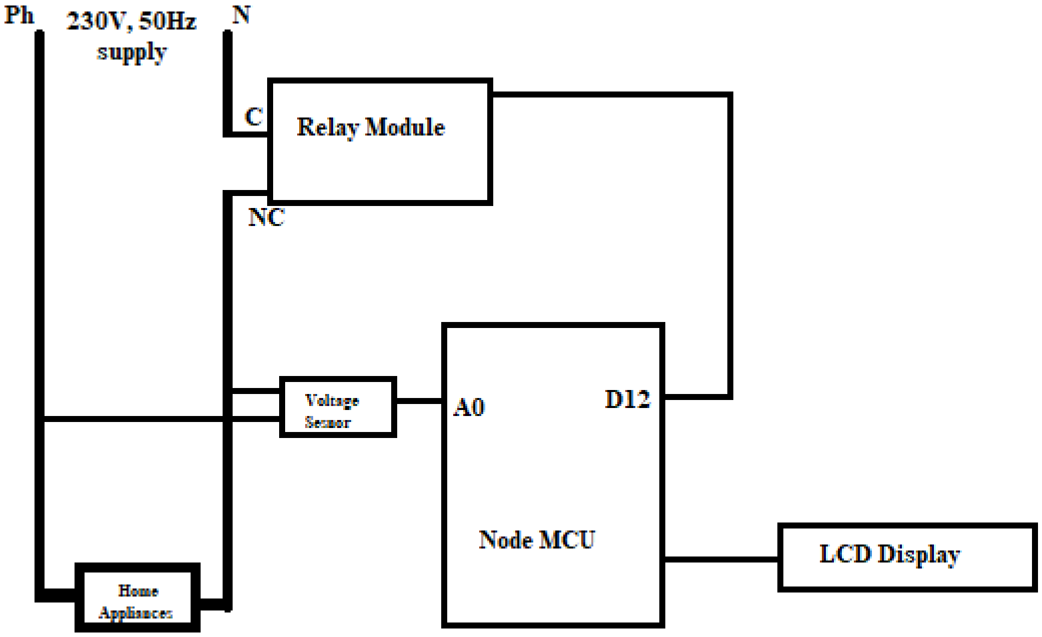

Figure: 8.1Circuit Diagram for over/under voltage protection.

8.4 ARDUINO CODE

```
#define CURRENT_SENSOR A0  // Define Analog input pin that sensor is attached
float amplitude_current;      // Float amplitude current
float effective_value;       // Float effective current
void setup()
{
   Serial.begin(9600);
   pins_init();
}
void loop()
{
   int current =0, currentTh=5;
   current = anloagRead(0);
   Current =current*(5/1023)-2.5;
   if Current >= currentTh
{
lcd.print(‘Over current detection’);
digitalWrite(12,HIGH);
lcd.print(‘Relay Tripped’);
}
}
```

8.5 PROCEDURE

1. Make the connection as per the electrical circuit diagram
2. Open the Arduino IDE in computer and write the program.
3. Compile the program for any errors and upload it to the Arduino.
4. Observe the status of the relay on the LCD display and also on the web-link through IOT.

8.6 RESULT:

8.7 PRE LAB VIVA QUESTIONS:

1. What do you mean by over voltage?
2. What do you mean by under voltages?
3. What are the effects of over voltages?
4. What are the effects of under voltages?
5. What is the node MCU?

8.8 POST LAB VIVA QUESTIONS:

1. What is the wi-fi module used in node MCU?
2. Which sensor is used to measure AC voltages?
3. What is the rated voltage of home appliances?
4. Draw the internal circuit diagram of a relay module.
5. When over/under voltages will occur?

EXPERIMENT – 9

PROTECTION OF THREE PHASE INDUCTION MOTOR

9.1 AIM:

Design a system for protecting the three phase induction motor from over voltages, over currents, temperature and displaying the status of the motor at remote location using IOT.

9.2 APPARATUS:

S. No	Name of the equipment	Quantity
1	Arduino and ESP8266 wi fi module	1
2	16 x 2 LCD display	1
3	Voltage sensor	1
4	Current sensor	1
5	Temperature sensor	1
6	Relay module	1
7	3 pole contactor	1
8	3 phase induction motor	1
9	Computer with Arduino IDE software	1
10	Connecting wires	As required

9.3 ELECTRICAL CIRCUIT DIAGRAM:

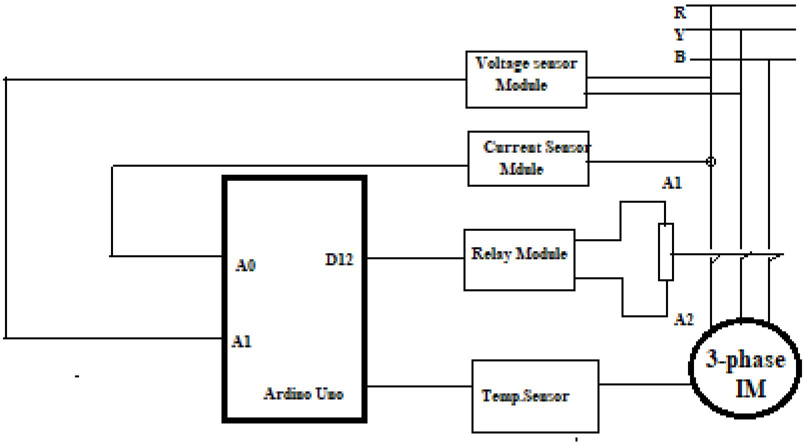

Figure: 9.1 Circuit Diagram for protection of three phase induction motor

9.4 CODE

```
#define CURRENT_SENSOR A0  // Define Analog input pin that sensor is attached
float amplitude_current;      // Float amplitude current
float effective_value;       // Float effective current
void setup()
{
   Serial.begin(9600);
   pins_init();
}
void loop()
{
 int current =0, currentTh=5;
 current = anloagRead(0);
 Current =current*(5/1023)-2.5;
 if Current >= currentTh
{
lcd.print('Over current detection');
digitalWrite(12,HIGH);
lcd.print('Relay Tripped');
}
}
```

9.5 PROCEDURE

1. Make the connection as per the electrical circuit diagram
2. Open the Arduino IDE in computer and write the program.
3. Compile the program for any errors and upload it to the Arduino.
4. Observe the parameters of an induction motor on the web-link through IOT.

9.6 RESULT:

9.7 PRE LAB VIVA QUESTIONS:

1. What is temperature?
2. What are the starting methods of three phase induction motor?
3. What is the effect of change in voltage on induction motor?
4. What is the effect of change in current on induction motor?
5. What is the effect of change in temperature on induction motor?

9.8 POST LAB VIVA QUESTIONS:

1. What is a 3 pole contactor?
2. Why 3 polecontactor is used in this experiment?
3. Which sensor is used to measure temperature?
4. How to measure the 3 phase voltage?
5. Which sensor is used to measure AC current?

EXPERIMENT – 10

TRAFFIC SIGNAL CONTROL

10.1 AIM:

To design a traffic control system using IOT

10.2 APPARATUS:

S No	Name of the equipment	Quantity
1	Arduino and ESP8266 wi fi module	1
2	16 x 2 LCD display	1
3	Proximity sensor	1
4	LEDs (R,Y,G)	6
5	Computer with Arduino IDE software	1
6	Connecting wires	As required

10.3 CIRCUIT DIAGRAM:

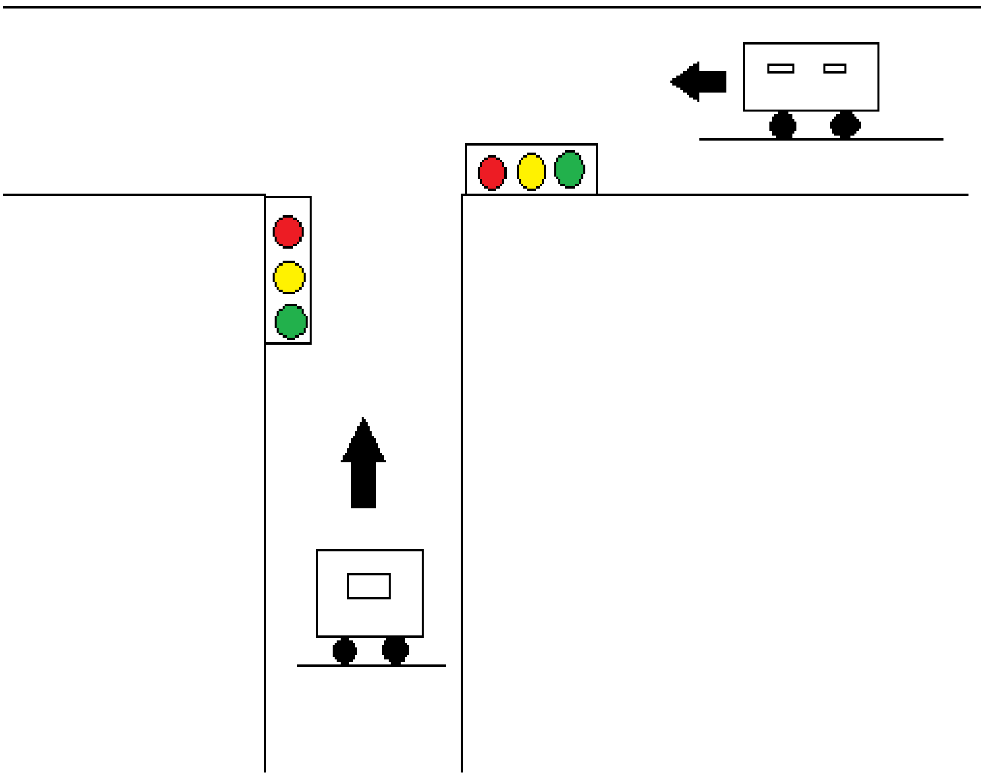

Figure: 10.1Circuit Diagram for protection of three phase induction motor

10.4 CODE

```
// light one
int red1 = 10;
int yellow1 = 9;
int green1 = 8;
// light two
int red2 = 13;
int yellow2 = 12;
int green2 = 11;
void setup()
{
   // light one
pinMode(red1, OUTPUT);
pinMode(yellow1, OUTPUT);
pinMode(green1, OUTPUT);
   // light two
pinMode(red2, OUTPUT);
pinMode(yellow2, OUTPUT);
pinMode(green2, OUTPUT);
}
void loop()
{
changeLights();
delay(15000);
}
void changeLights()
{
   // turn both yellows on
digitalWrite(green1, LOW);
digitalWrite(yellow1, HIGH);
digitalWrite(yellow2, HIGH);
delay(5000);
   // turn both yellows off, and opposite green and red
digitalWrite(yellow1, LOW);
digitalWrite(red1, HIGH);
digitalWrite(yellow2, LOW);
digitalWrite(red2, LOW);
digitalWrite(green2, HIGH);
delay(5000);
   // both yellows on again
digitalWrite(yellow1, HIGH);
digitalWrite(yellow2, HIGH);
digitalWrite(green2, LOW);
delay(3000);
   // turn both yellows off, and opposite green and red
digitalWrite(green1, HIGH);
digitalWrite(yellow1, LOW);
digitalWrite(red1, LOW);
digitalWrite(yellow2, LOW);
digitalWrite(red2, HIGH);
delay(5000);
}
```

10.5 PROCEDURE

1. Make the connection as per the electrical circuit diagram
2. Open the Arduino IDE in computer and write the program.
3. Compile the program for any errors and upload it to the Arduino.
4. Observe the status of traffic light on junctions at traffic control room through IOT.

10.6 RESULT:

10.7 PRE LAB VIVA QUESTIONS:

1. What is red light indicates in traffic signals?
2. What is yellow light indicates in traffic signals?
3. What is green light indicates in traffic signals?
4. What do you mean by free left at signal junctions?
5. What are the precautions to be taken while driving?

10.8 POST LAB VIVA QUESTIONS:

1. What is meant by LED?
2. How the colour for an LED will be changes?
3. What are the operating voltages of red, yellow and green LEDs?
4. What are the parameters that can be controlled through IOT in this experiment?
5. Where this traffic signal controlling system with IOT is preferable?

EXPERIMENT – 11

RAILWAY GATE CONTROL BY STEPPER MOTOR

11.1 AIM:

To design a railway gate control using stepper motor using IOT

11.2 APPARATUS:

S No	Name of the equipment	Quantity
1	Arduino and ESP8266 wi-fi module	1
2	16 x 2 LCD display	1
3	IR sensors	2
4	Stepper motors	2
5	Railway track kit	1
6	Computer with Arduino IDE software	1
7	Connecting wires	As required

11.3 ELECTRICAL CIRCUIT DIAGRAM:

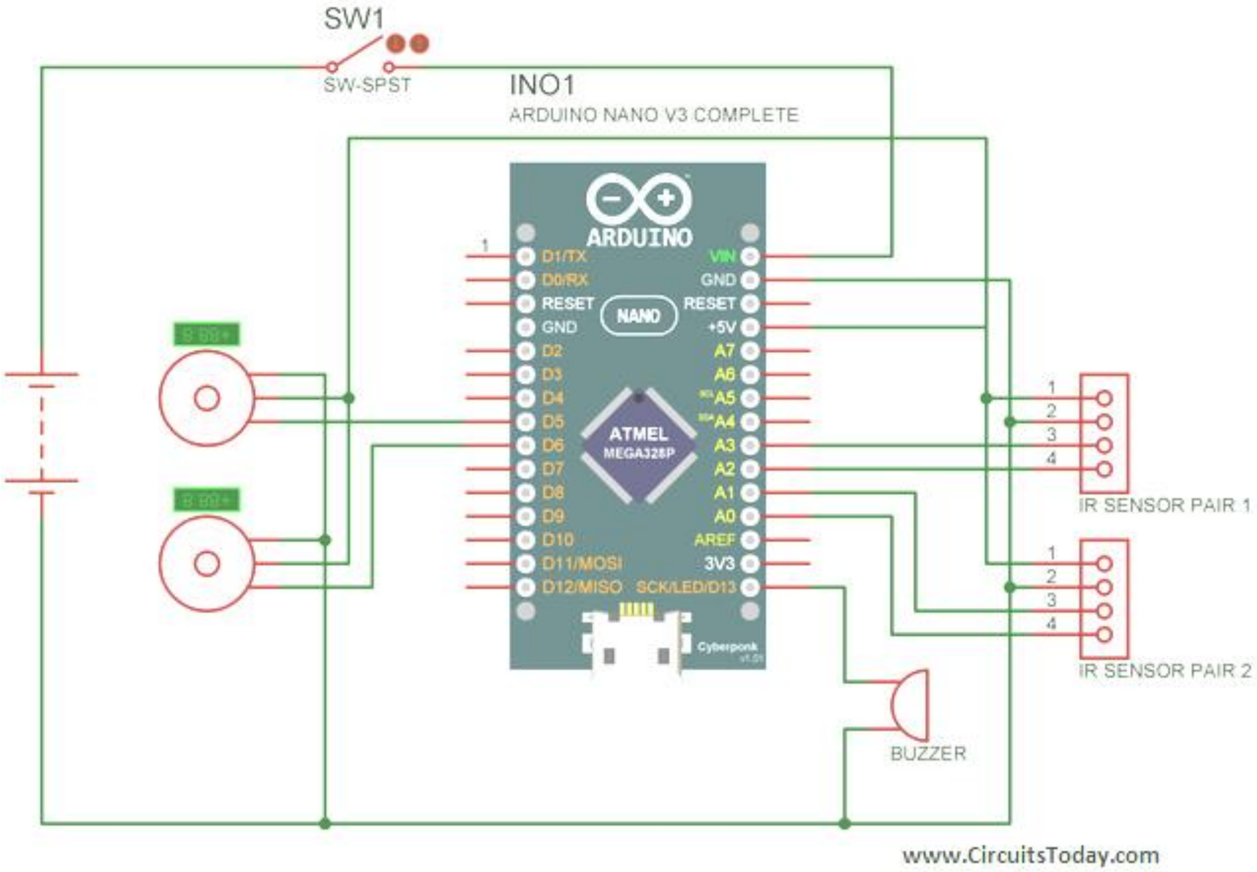

Figure: 11.1Circuit Diagram for automatic railway gate control

11.4 CODE

```
#include <EEPROM.h>
```

```
#include <Servo.h>
Servo myservo1;
Servo myservo2;
int pos = 0;
int sensor1 = A0;
int sensor2 = A1;
int sensor3 = A2;
int sensor4 = A3;
int RLED1 = 8;
int GLED1 = 9;
int RLED2 = A4;
int GLED2 = A5;
int buzzer = 13;
void setup() {
  //Serial.begin(9600);
  myservo1.attach(5);
  myservo2.attach(6);
pinMode(sensor1,INPUT);
pinMode(sensor2,INPUT);
pinMode(sensor3,INPUT);
pinMode(sensor4,INPUT);
pinMode(buzzer,OUTPUT);
EEPROM.write(0, 0);
EEPROM.write(1, 0);
}
void loop() {
    if (EEPROM.read(1)==0 || EEPROM.read(0)==0){
    while (analogRead(sensor1)>500){
      if (analogRead(sensor2)>500){
        if (EEPROM.read(1)!=1){ EEPROM.write(1, 1);}
        if (EEPROM.read(0)!=1){ EEPROM.write(0, 1);}
              // Serial.println("X");
delay(100);
        }}
    while (analogRead(sensor3)>500){
      if (analogRead(sensor4)>500){
        if (EEPROM.read(1)!=1){ EEPROM.write(1, 1);}
        if (EEPROM.read(0)!=1){ EEPROM.write(0, 1);}
       // Serial.println("Y");
delay(100);
        }}
  }
  if (EEPROM.read(1)==1 || EEPROM.read(0)==0){
    while (analogRead(sensor2)>500){
      if (analogRead(sensor1)>500){
        if (EEPROM.read(1)!=0){ EEPROM.write(1, 0);}
        if (EEPROM.read(0)!=1){ EEPROM.write(0, 1);}
                //Serial.println("Z");
delay(100);
        }}
    while (analogRead(sensor4)>500){
      if (analogRead(sensor3)>500){
        if (EEPROM.read(1)!=0){ EEPROM.write(1, 0);}
        if (EEPROM.read(0)!=1){ EEPROM.write(0, 1);}
        //Serial.println("A");
delay(100);
        }}
  }
  if (EEPROM.read(1)==1){     //Gate open
    if (pos != 90){
      for (pos = 0; pos < 90; pos += 1){
      myservo1.write(pos);
      myservo2.write(pos);
     // Serial.println(pos);
digitalWrite(buzzer,HIGH);
delay(10);
digitalWrite(buzzer,LOW);
delay(10);
    }}
        //Serial.println("Gate Opend");
digitalWrite(buzzer,LOW);
    if (EEPROM.read(0)!=0)  {EEPROM.write(0, 0);  /*Serial.println("OK");*/}
  }
  if (EEPROM.read(1)==0){     //Gate close
    if (pos != 0){
     for (pos = 90; pos > 0; pos -= 1){
         myservo1.write(pos);
         myservo2.write(pos);
digitalWrite(buzzer,HIGH);
delay(10);
digitalWrite(buzzer,LOW);
delay(10);
    }}
        //Serial.println("Gate Closed");
    if (EEPROM.read(0)!=0)  {EEPROM.write(0, 0); Serial.println("OK");}
  }
}
```

11.5 PROCEDURE

1. Make the connection as per the electrical circuit diagram
2. Open the Arduino IDE in computer and write the program.
3. Compile the program for any errors and upload it to the Arduino.
4. Observe the control of railway get when train comes near to the gate and closes when train passes away the gate and display the status of railway gate at nearby station through IOT.

11.6 RESULT:

11.7 PRE LAB VIVA QUESTIONS:

1. What is a stepper motor?
2. What is the principle of operation of stepper motor?
3. What is the step angle of stepper motor?
4. What is the voltage level of Indian railway system?
5. What is the symbol X which is back side of the last coach?

11.8 POST LAB VIVA QUESTIONS:

1. What is the microcontroller used in this experiment?
2. What is an IR sensor?
3. What is the range of IR sensor?
4. What are the libraries needed to run the stepper motor?
5. What are the applications of stepper motor?

EXPERIMENT – 12

DIRECTION AND SPEED CONTROL OF DC MOTOR

12.1 AIM:

To control the speed and direction of a DC motor using Arduino and display the status of the motor at the remote location using IOT.

12.2 APPARATUS:

S No	Name of the equipment	Quantity
1	Arduino and ESP8266 wi-fi module	1
2	16 x 2 LCD display	1
3	DC motor (5V)	1
4	L296D motor driver module	1
5	Computer with Arduino IDE software	1
6	Connecting wires	As required

12.3 ELECTRICAL CIRCUIT DIAGRAM:

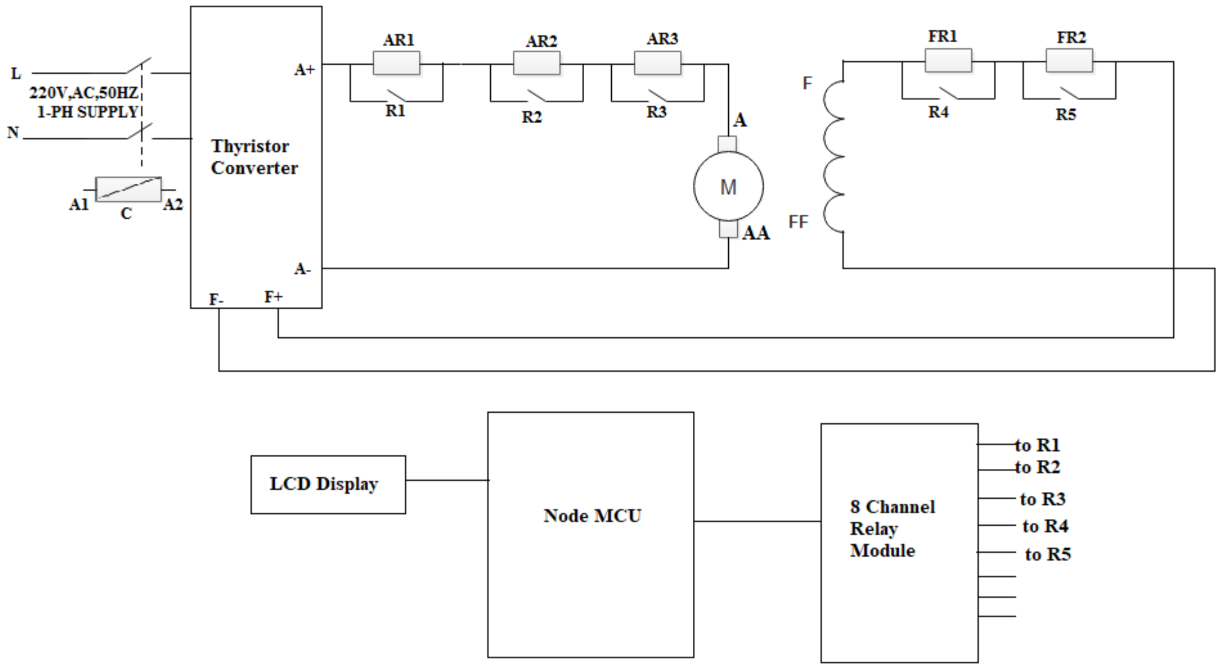

Figure: 12.1Circuit Diagram for speed control of DC motor

12.4 CODE

```
#include <LiquidCrystal.h>
//Keyboard Controls:
// C - Clockwise
// S - Stop
// A - Anti-clockwise
// Declare L298N Controller pins
// Motor 1
int count=255;
int dir1PinA = 2;
int dir2PinA = 5;
int speedPinA = 6; // PWM control
LiquidCrystallcd(8, 9, 10, 11, 12, 13);
void setup() {
Serial.begin(9600); // baud rate
lcd.begin(20, 4);
lcd.setCursor(5,0);
lcd.print("DC Motor");
lcd.setCursor(5,1);
lcd.print("Direction");
lcd.setCursor(5,2);
lcd.print("Control");
lcd.setCursor(2,3);
lcd.print("via Arduino UNO");
delay(3000);
lcd.clear ();
lcd.setCursor(0,2);
lcd.print("www.TheEngineering");
lcd.setCursor(4,3);
lcd.print("Projects.com");
//Define L298N Dual H-Bridge Motor Controller Pins
pinMode(dir1PinA,OUTPUT);
pinMode(dir2PinA,OUTPUT);
pinMode(speedPinA,OUTPUT);
analogWrite(speedPinA, 255);//Sets speed variable via PWM
}
void loop() {
// Initialize the Serial interface:
if (Serial.available() > 0) {
int inByte = Serial.read();
int speed; // Local variable
switch (inByte) {
case 'C': // Clockwise rotation
//analogWrite(speedPinA, 255);//Sets speed variable via PWM
digitalWrite(dir1PinA, LOW);
digitalWrite(dir2PinA, HIGH);
Serial.println("Clockwise rotation"); // Prints out “Motor 1 Forward” on the serial monitor
Serial.println("   "); // Creates a blank line printed on the serial monitor
//lcd.clear();
lcd.setCursor(0,0);
lcd.print("Clockwise rotation");
break;
case 'S': // No rotation
//analogWrite(speedPinA, 0); // 0 PWM (Speed)
digitalWrite(dir1PinA, LOW);
digitalWrite(dir2PinA, LOW);
Serial.println("No rotation");
Serial.println("   ");
//lcd.clear();
lcd.setCursor(0,0);
lcd.print("No rotation");
break;
case 'H': //Accelrating motor
count=count+20;
  if (count>255)
  {
  count =255;
  }
analogWrite(speedPinA,count);
delay(50);
//digitalWrite(dir1PinA, LOW);
//digitalWrite(dir2PinA, HIGH);
Serial.println("Motor is accelrating slowly");
Serial.println("   ");
Serial.println(count);
lcd.setCursor(0,0);
lcd.print("Motor is accelrating");
break;
case 'L': //Deaccelrating motor
count=count-20;
  if (count<20)
  {
  count=20;
  }
analogWrite(speedPinA,count);
delay(50);
//digitalWrite(dir1PinA, LOW);
//digitalWrite(dir2PinA, HIGH);
Serial.println("Motor is deaccelrating slowly");
Serial.println("   ");
Serial.println(count);
lcd.setCursor(0,0);
lcd.print("Motor Deaccelrates");
break;
case 'A': // Anti-clockwise rotation
//analogWrite(speedPinA, 255); // Maximum PWM (speed)
digitalWrite(dir1PinA, HIGH);
digitalWrite(dir2PinA, LOW);
Serial.println("Anti-clockwise rotation");
Serial.println("   ");
//lcd.clear();
lcd.setCursor(0,0);
lcd.print("Anti-clockwise");
break;
default:
// Turn off the motor if any other key is being pressed
for (int thisPin = 2; thisPin< 11; thisPin++) {
digitalWrite(thisPin, LOW);
}
Serial.println("Wrong key is pressed");
//lcd.clear();
lcd.setCursor(0,0);
lcd.print("Wrong key is pressed");
  }
    }
      }
```

12.5 PROCEDURE

1. Make the connection as per the electrical circuit diagram
2. Open the Arduino IDE in computer and write the program.
3. Compile the program for any errors and upload it to the Arduino.
4. Observe the speed and direction of the motor at remote location through IOT.

12.6 RESULT:

12.7 PRE LAB VIVA QUESTIONS:

1. What is principle of DC motor?
2. What is armature reaction?
3. What is a commutator?
4. What is the type of the EMF produced in the armature conductors?
5. What is the material used for brushes of a DC motor?

12.8 POST LAB VIVA QUESTIONS:

1. What is the microcontroller used in this experiment?
2. How to control the direction of DC motor?
3. What is the motor driver used in the experiment?
4. How many motors can be controlled by the L296D driver?
5. How to control the speed of the dc motor?