

Subject: “Internet of Things (IoT)” LAB

Subject Code: 18CSL65

Program-1

Write a program using different components,

a) LEDs for configuring and controlling different on board peripherals

b) Switches to configure an LED as an output and a switch as an input Pull up mode
on TM4C123 Launchpad using Energia software

a) LED Blink

```
void setup()
{
  // put your setup code here, to run once:
  pinMode(RED_LED, OUTPUT);
}

void loop()
{
  // put your main code here, to run repeatedly:
  digitalWrite(RED_LED, HIGH);
  delay(1000);
  digitalWrite(RED_LED, LOW);
  delay(1000);
}
```



b) Switch 1 & 2

```
void setup()
{
  // put your setup code here, to run once:
  pinMode(RED_LED, OUTPUT);

  pinMode(PUSH1, INPUT_PULLUP);
}

void loop()
{
  // put your main code here, to run repeatedly:
  int SW1 = digitalRead(PUSH1);

  if(SW1 == LOW){
    digitalWrite(RED_LED, HIGH);
  }
  else{
    digitalWrite(RED_LED, LOW);
  }
}
```



SWITCH 1



SWITCH 2

On pressing SWITCH 1 OR SWITCH 2, LED glows

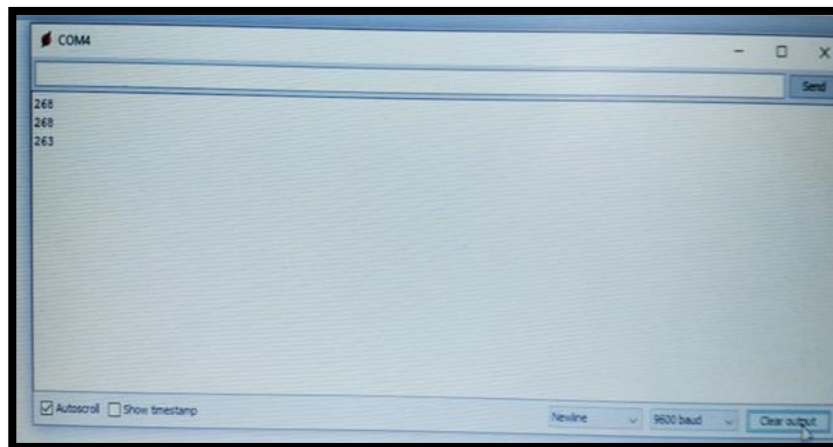
Program-2

Write a program to show

- a) Analog to Digital Conversion and its output being monitored on the serial monitor
 - b) Pulse Width Modulation(PWM) for changing the intensity of the LED
 - c) UART for serial communication
- using TM4C123 Launchpad and Energia software

a) ADC

```
void setup()
{
    // put your setup code here, to run once:
    pinMode(A0, INPUT);
    Serial.begin(9600);
}
void loop()
{
    // put your main code here, to run repeatedly:
    int a = analogRead(A0);
    Serial.println(a);
    delay(500);
}
```



b) PWM

CASE 1:

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

```
pinMode(RED_LED, OUTPUT);
}
void loop()
{
  // put your main code here, to run repeatedly:
  int value;
  for(value = 0; value < 256; value+=10){
    analogWrite(RED_LED, value);
    delay(300);
  }
}
```

CASE 2:

```
void setup()
{
  // put your setup code here, to run once:
  pinMode(RED_LED, OUTPUT);
}
void loop()
{
  // put your main code here, to run repeatedly:
  int value;
  for(value = 255; value >= 0; value-=10){
    analogWrite(RED_LED, value);
    delay(300);
  }
}
```

CASE 1

CASE 2



c) Serial Communication

```
void setup()
{
  // put your setup code here, to run once:
  Serial.begin(9600);
}

void loop()
{
  // put your main code here, to run repeatedly:
  if(Serial.available()){
    char a = Serial.read();
    Serial.print(a);
    delay(500);
  }
}
```



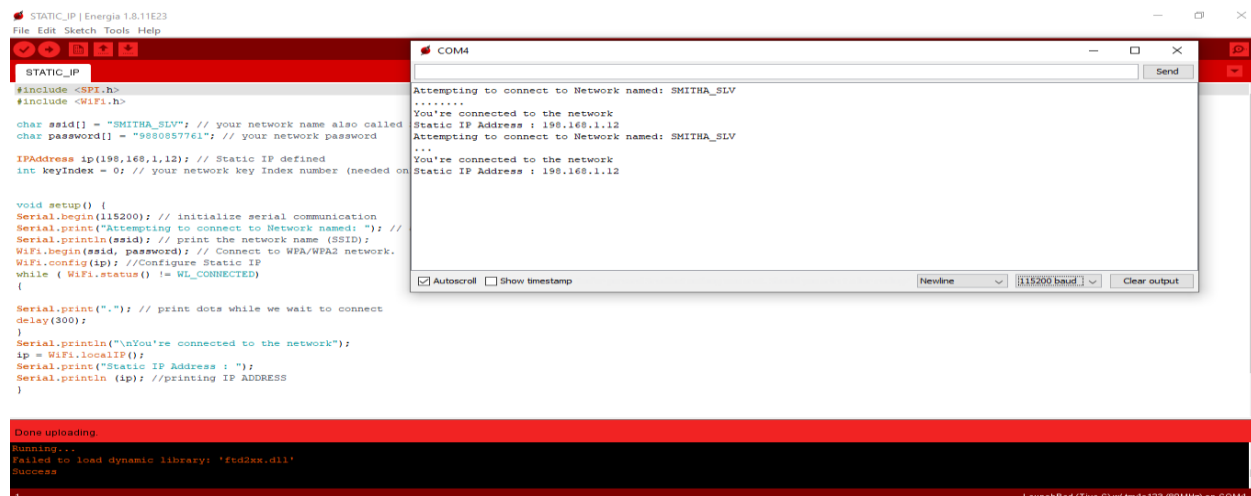
Program-3

Write a program to connect the Launchpad with Wi-Fi network & print the static IP Addresses on the Serial Monitor

```
#include <SPI.h>
#include <WiFi.h>
char ssid[] = ""; // your network name also called SSID
char password[] = ""; // your network password
IPAddress ip(198,168,1,12); // Static IP defined
int keyIndex = 0; // your network key Index number (needed only for WEP)

void setup()
{
  Serial.begin(115200); // initialize serial communication
  Serial.print("Attempting to connect to Network named: "); // attempt to connect to Wifi network:
  Serial.println(ssid); // print the network name (SSID);
  WiFi.begin(ssid, password); // Connect to WPA/WPA2 network.
  WiFi.config(ip); //Configure Static IP
  while ( WiFi.status() != WL_CONNECTED)
  {
    Serial.print("."); // print dots while we wait to connect
    delay(300);
  }
  Serial.println("\nYou're connected to the network");
  ip = WiFi.localIP();
  Serial.print("Static IP Address : ");
  Serial.println(ip); //printing IP ADDRESS
}

void loop()
{
}
```

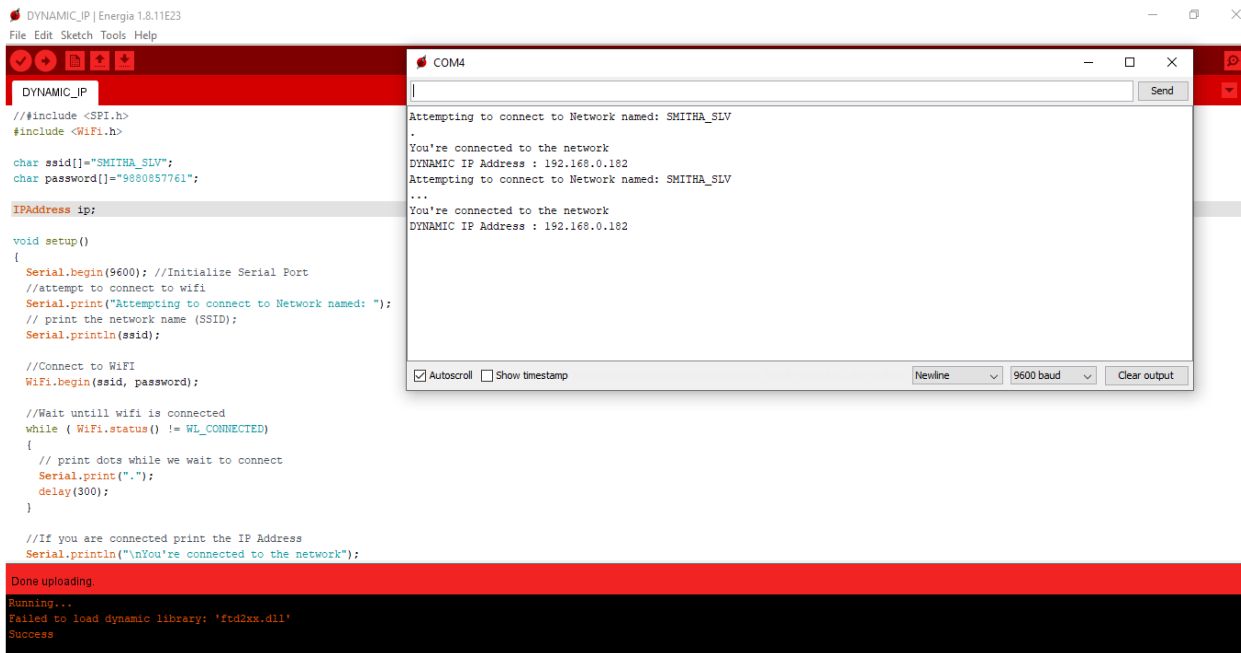


Program-4

Write a program to connect the Launchpad with Wi-Fi network & print the dynamic IP Addresses on the Serial Monitor

```
#include <SPI.h>
#include <WiFi.h>
char ssid[]=" ";
char password[]=" ";
IPAddress ip;
void setup()
{
  //Initialize Serial Port
  Serial.begin(9600);
  //attempt to connect to wifi
  Serial.print("Attempting to connect to Network named: ");
  // print the network name (SSID);
  Serial.println(ssid);
  //Connect to WiFi
  WiFi.begin(ssid, password);
  //Wait untill wifi is connected
  while ( WiFi.status() != WL_CONNECTED)
  {
    // print dots while we wait to connect
    Serial.print(".");
    delay(300);
  }
  //If you are connected print the IP Address
  Serial.println("\nYou're connected to the network");

  //wait untill you get an IP address
  while (WiFi.localIP() == INADDR_NONE) {
    // print dots while we wait for an ip addresss
    Serial.print(".");
    delay(300);
  }
  ip=WiFi.localIP();
  Serial.print(" Dynamic IP Address : ");
  Serial.println(ip);
}
void loop()
{
  // put your main code here, to run repeatedly:
}
```

Program-5

Write a program to connect the Launchpad with Wi-Fi and print the local IP, Subnet Mask, Gateway IP on the Serial Monitor

```
#include <SPI.h>
#include <WiFi.h>
char ssid[]=" ";
char password[]=" ";
IPAddress ip,subnet,gateway;
void setup()
{
  Serial.begin(9600); //Initialize Serial Port
  //attempt to connect to wifi
  Serial.print("Attempting to connect to Network named: ");
  // print the network name (SSID);
  Serial.println(ssid);
  //Connect to WiFi
  WiFi.begin(ssid, password);
  //Wait untill wifi is connected
  while ( WiFi.status() != WL_CONNECTED)
  {
    // print dots while we wait to connect
    Serial.print(".");
    delay(300);
  }
}
```



```

//If you are connected print the IP Address
Serial.println("\nYou're connected to the network");
//wait untill you get an IP address
while (WiFi.localIP() == INADDR_NONE) {
// print dots while we wait for an ip addresss
Serial.print(".");
delay(300);
}
ip=WiFi.localIP();
Serial.print("IP Address : ");
Serial.println(ip);
subnet=WiFi.subnetMask();
Serial.print("SubnetMask Address : ");
Serial.println(subnet);
gateway=WiFi.gatewayIP();
Serial.print("Gateway IP is: ");
Serial.println(gateway);
}
void loop()
{
// put your main code here, to run repeatedly:
}

```

The screenshot shows the Arduino IDE interface. The top menu bar includes File, Edit, Sketch, Tools, and Help. The main editor displays the 'SUBMASKGATEWAY' sketch with the following code:

```

#include <SPI.h>
#include <WiFi.h>
char ssid[]="SMITHA_SLV";
char password[]="98080857761";
IPAddress ip,subnet,gateway;
void setup()
{
  Serial.begin(9600); //Initialize Serial Port
  //attempt to connect to wifi
  Serial.print("Attempting to connect to wifi");
  // print the network name (SSID):
  Serial.println(ssid);
  //Connect to WiFi
  WiFi.begin(ssid, password);
  //Wait untill wifi is connected
  while ( WiFi.status() != WL_CONNECTED)
  {
    // print dots while we wait to connect
    Serial.print(".");
    delay(300);
  }
  //If you are connected print the IP Address
  Serial.println("\nYou're connected to the network");
  //wait untill you get an IP address
  while (WiFi.localIP() == INADDR_NONE) {
    // print dots while we wait for an ip address
    Serial.print(".");
    delay(300);
  }
}

```

The serial monitor window (COM4) shows the following output:

```

Attempting to connect to Network named: SMITHA_SLV
.
You're connected to the network
...IP Address : 192.168.0.182
SubnetMask Address : 255.255.255.0
Gateway IP is: 192.168.0.1

```

At the bottom of the IDE, a status bar indicates 'Done uploading', 'Running...', and 'Failed to load dynamic library: 'ftd2xx.dll''. The bottom right corner shows 'LaunchPad (Tiva C) w/ tm4c123 (80MHz) on COM4'.

Program-6

Illustrate TCP based Client Server Communication Model.

SERVER SIDE

```
#include <SPI.h>
#include <WiFi.h>
IPAddress ip(192, 168, 1, 122);
// your network name also called SSID
char ssid[] = " ";
// your network password
char password[] = " ";
unsigned char data=0;
WiFiServer server(9999);
void setup() {
  //Initialize serial and wait for port to open:
  Serial.begin(9600);
  // attempt to connect to Wifi network:
  Serial.print("Attempting to connect to Network named: ");
  // print the network name (SSID);
  Serial.println(ssid);
  // Connect to WPA/WPA2 network. Change this line if using open or WEP network:

  WiFi.config(ip);
  WiFi.begin(ssid, password);
  while ( WiFi.status() != WL_CONNECTED) {
    // print dots while we wait to connect
    Serial.print(".");
    delay(300);
  }
  Serial.println("\nYou're connected to the network");
  Serial.println("Waiting for an ip address");
  while (WiFi.localIP() == INADDR_NONE) {
    // print dots while we wait for an ip addresss
    Serial.print(".");
    delay(300);
  }
  Serial.println("\nIP Address obtained");
  // you're connected now, so print out the status:
printWifiStatus();
  // start the server:
  server.begin();
}
void loop()
{
  // wait for a new client:
  WiFiClient client = server.available();
```

```

if(client==true)
{
  server.print("HELLO\n");
  delay(500);
}
}

void printWifiStatus() {
  // print the SSID of the network you're attached to:
  Serial.print("SSID: ");
  Serial.println(WiFi.SSID());
  // print your WiFi shield's IP address:
  IPAddress IP = WiFi.localIP();
  Serial.print("IP Address: ");
  Serial.println(IP);
}

```

CLIENT SIDE

```

#include <SPI.h>
#include <WiFi.h>
char ssid[] = " ";
char password[] = " ";
uint16_t port = 9999; // port number of the server
IPAddress server(192,168,1,122); // IP Address of the server
WiFiClient client;
void setup()
{
  Serial.begin(9600);
  WiFi.begin(ssid, password);

  while(WiFi.status() != WL_CONNECTED)
  {
    Serial.println("Waiting for Wi-Fi Connection");
    Serial.print(".");
    delay(300);
  }

  Serial.println("WiFi Connected");
  Serial.println("Attempting to connect to Server");

  while(!client.connect(server,port));
  Serial.println("Connected to Server");
}

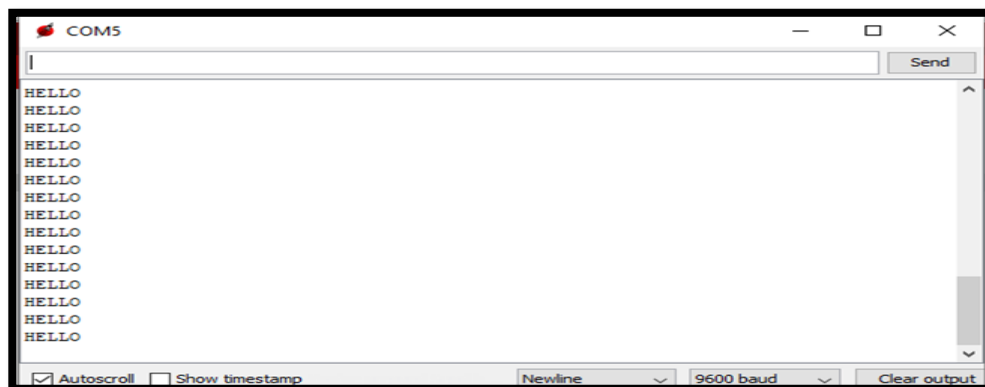
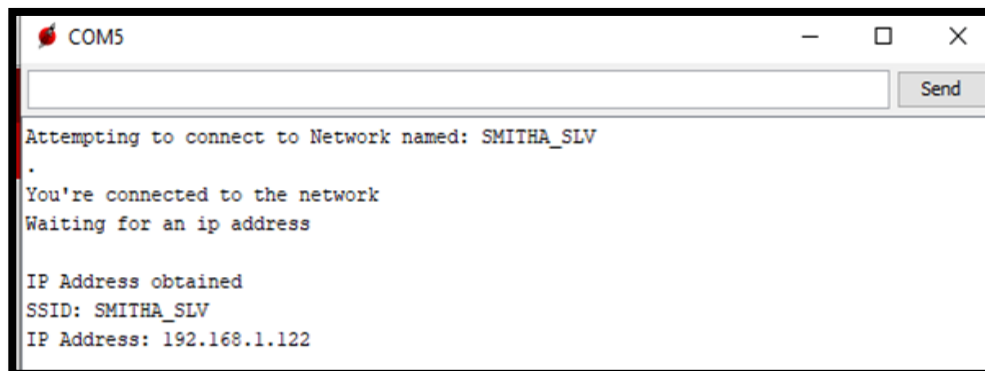
void loop()
{
  if(client.available())

```

```

{
  char c=client.read();
  Serial.print(c);
}
}

```



Program-7

Illustrate UDP based Client Server Communication Model.

UDP Client

```

#include <SPI.h>
#include <WiFi.h>

```

```

char net[] = "Realme C1";
char pass[] = " ";
char a[15];

```

```

unsigned int LP = 5555;
IPAddress ip(192,168,43,155);
WiFiUDP Udp;
void setup()
{
  Serial.begin(9600);
  WiFi.begin(net,pass);
  while(WiFi.status() != WL_CONNECTED)
  {
    Serial.print(">");
    delay(500);
  }
  Serial.print(net);
  Serial.println(":conneted");

  Serial.println("Acquiring IP address");
  while(WiFi.localIP() == INADDR_NONE)
  {
    Serial.print(">");
    delay(500);
  }
  Serial.print("Acquired IP address is :");
  Serial.println(WiFi.localIP());
  Udp.begin(LP);
}

void loop()
{
  char i =0;
  // put your main code here, to run repeatedly:
  if(Serial.available())
  {
    Serial.readBytes(a,15);
    Udp.beginPacket(ip,2600);//Starts a connection to write UDP data
    Udp.write(a); //writes UDP data to remote connection
    Udp.endPacket(); //It finishes off the packet and sends it.
    for(i = 0;i<10;i++)
      a[i] = ' ';
  }
  int packsize = Udp.parsePacket();
  if(packsize)
  {
    memset(a,0,10);
    Udp.read(a,10);
    /*ip address of the

```

```

host who sent the current incoming packet*/
  Serial.println(Udp.remoteIP());
/*the port of the host
who sent the current incoming packet*/
  Serial.println(Udp.remotePort());
  Serial.println(a);
}
}

```

UDP SERVER

```

#include <SPI.h>
#include <WiFi.h>

char net[] = "Realme C1";
char pass[] = " ";
char a[15];
unsigned int LP = 2600;
IPAddress ip(192,168,43,155);
IPAddress ipr(192,168,43,161);

WiFiUDP Udp;
void setup()
{
  Serial.begin(9600);
  WiFi.config(ip);
  WiFi.begin(net,pass);
  while(WiFi.status() != WL_CONNECTED)
  {
    Serial.print(">");
    delay(500);
  }
  Serial.print(net);
  Serial.println(":connected");
  Serial.println("Acquiring IP address");
  while(WiFi.localIP() == INADDR_NONE)
  {
    Serial.print(">");
    delay(500);
  }
  Serial.print("Acquired IP address is :");
  Serial.println(WiFi.localIP());
  Udp.begin(LP);
}

void loop()
{

```

```

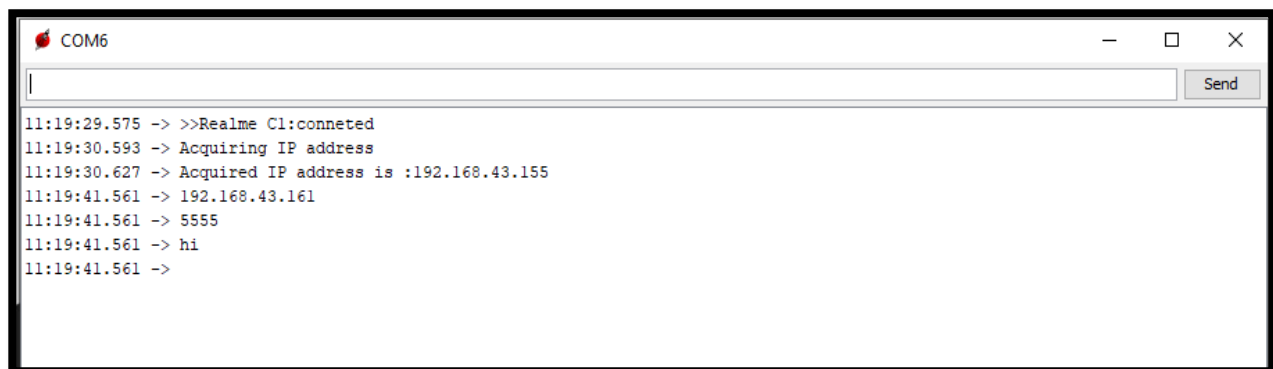
//It starts processing the next available
//incoming packet and returns the size in
//bytes
int packsize = Udp.parsePacket();
if(packsize)
{
    memset(a,0,10);
    Udp.read(a,10);
    /*ip address of the
    host who sent the current incoming packet*/
    Serial.println(Udp.remoteIP());
    /*the port of the host
    who sent the current incoming packet*/
    Serial.println(Udp.remotePort());
    Serial.println(a);
}

char i =0;
// put your main code here, to run repeatedly:
if(Serial.available())
{
    Serial.readBytes(a,15);
    Udp.beginPacket(ipr,5555); //Starts a connection to write UDP data
    Udp.write(a); //writes UDP data to remote connection
    Udp.endPacket(); //It finishes off the packet and sends it.
    for(i = 0;i<10;i++)
    a[i] = ' ';
}

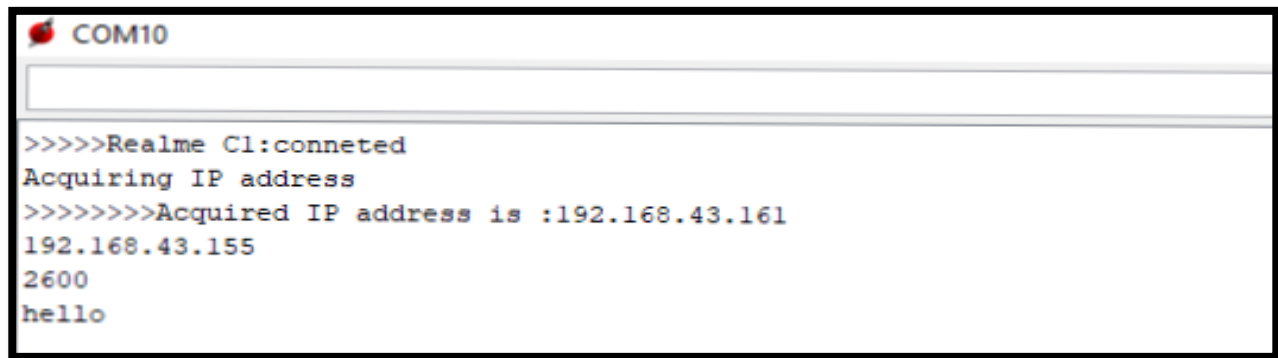
delay(100);
}

```

SERVER



CLIENT



```
>>>>>Realme C1:conneted
Acquiring IP address
>>>>>>>Acquired IP address is :192.168.43.161
192.168.43.155
2600
hello
```

Program-8

Write a program for HTTP based webserver to manipulate the GPIO's of WiFi Module CC3100 and blink a LED(HIGH/LOW)..

```
#ifndef __CC3200R1M1RGC__
// Do not include SPI for CC3200 LaunchPad
#include <SPI.h>
#endif
#include <WiFi.h>
int m;
// your network name also called SSID
char ssid[] = "SMITHA_SLV";
// your network password
char password[] = "";
WiFiServer server(80); //HTTP server port number

void setup() {
  Serial.begin(9600); // initialize serial communication
  pinMode(RED_LED, OUTPUT); // set the LED pin mode
  //pinMode(40, OUTPUT);
  //pinMode(39, OUTPUT);
  // attempt to connect to Wifi network:
  Serial.print("Attempting to connect to Network named: ");
  // print the network name (SSID);
  Serial.println(ssid);
  // Connect to WPA/WPA2 network. Change this line if using open or WEP network:
  WiFi.begin(ssid, password);
  while ( WiFi.status() != WL_CONNECTED) {
    // print dots while we wait to connect
    Serial.print(".");
    delay(300);
  }

  Serial.println("\nYou're connected to the network");
  Serial.println("Waiting for an ip address");

  while (WiFi.localIP() == INADDR_NONE) {
    // print dots while we wait for an ip addresss
    Serial.print(".");
    delay(300);
  }

  Serial.println("\nIP Address obtained");

  // you're connected now, so print out the status
```

```

printWifiStatus();

Serial.println("Starting webserver on port 80");
server.begin();           // start the web server on port 80
Serial.println("Webserver started!");
}

void loop() {
  int i = 0;
  WiFiClient client = server.available(); // listen for incoming clients

  if (client) {            // if you get a client,
    Serial.println("new client"); // print a message out the serial port
    char buffer[150] = {0}; // make a buffer to hold incoming data
    while (client.connected()) { // loop while the client's connected
      if (client.available()) { // if there's bytes to read from the client,
        char c = client.read(); // read a byte, then
        Serial.write(c);        // print it out the serial monitor
        if (c == '\n') {        // if the byte is a newline character

          // if the current line is blank, you got two newline characters in a row.
          // that's the end of the client HTTP request, so send a response:
          if (strlen(buffer) == 0) {
            // HTTP headers always start with a response code (e.g. HTTP/1.1 200 OK)
            // and a content-type so the client knows what's coming, then a blank line:
            server.println("HTTP/1.1 200 OK"); /*Means that the server is responding
            using the HTTP protocol version 1.1. 200 is the code used when everything is ok. */
            server.println("Content-type:text/html");
            server.println();

            // the content of the HTTP response follows the header:
            server.println("<html><head><title>Energia CC3100 WiFi Web Server</title></head><body
align=center>");
            server.println("<h1 align=center><font color=\"red\">Welcome to the CC3100 WiFi Web
Server</font></h1>");
            server.print("<p align=center><font color=\"green\">Check the Status of RED LED</font></p>");
            server.print("<button onclick=\"location.href='/H'\">HIGH</button>");
            server.println("<button onclick=\"location.href='/L'\">LOW</button><br>");
            //server.print("<button onclick=\"location.href='/A'\">HIGH</button>");
            //server.println("<button onclick=\"location.href='/B'\">LOW</button><br>");
            //server.print("<button onclick=\"location.href='/C'\">HIGH</button>");
            //server.println("<button onclick=\"location.href='/D'\">LOW</button><br>");
            // The HTTP response ends with another blank line:
            server.println();
            // break out of the while loop:
            break;
          }
        }
      }
    }
  }
}

```

```

    else { // if you got a newline, then clear the buffer:
        memset(buffer, 0, 150);
        i = 0;
    }
}
else if (c != '\r') { // if you got anything else but a carriage return character,
    buffer[i++] = c; // add it to the end of the currentLine
}

// Check to see if the client request was "GET /H" or "GET /L":
if (endsWith(buffer, "GET /H")) {
    digitalWrite(30,HIGH); // GET /H turns the LED on
}
if (endsWith(buffer, "GET /L")) {
    digitalWrite(30, LOW); // GET /L turns the LED off
}

/* if (endsWith(buffer, "GET /A")) {
    digitalWrite(40, HIGH); // GET /H turns the LED on
}
if (endsWith(buffer, "GET /B")) {
    digitalWrite(40, LOW); // GET /L turns the LED off
}
if (endsWith(buffer, "GET /C")) {
    digitalWrite(39, HIGH); // GET /H turns the LED on
}
if (endsWith(buffer, "GET /D")) {
    digitalWrite(39, LOW); // GET /L turns the LED off
}*/
}
}
// close the connection:
client.stop();
Serial.println("client disconnected");
}
}

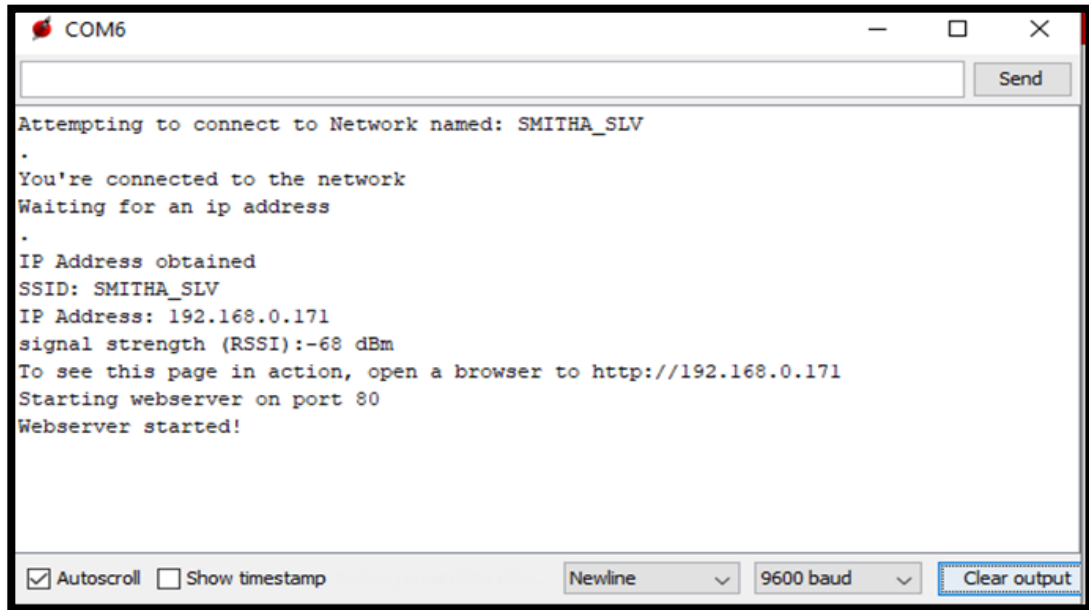
//
//a way to check if one array ends with another array
//
boolean endsWith(char* inString, char* compString) {
    int compLength = strlen(compString);
    int strLength = strlen(inString);

    //compare the last "compLength" values of the inString
    int i;

```

```
for (i = 0; i < compLength; i++) {  
    char a = inString[(strLength - 1) - i];  
    char b = compString[(compLength - 1) - i];  
    if (a != b) {  
        return false;  
    }  
}  
return true;  
}
```

```
void printWifiStatus() {  
    // print the SSID of the network you're attached to:  
    Serial.print("SSID: ");  
    Serial.println(WiFi.SSID());  
  
    // print your WiFi IP address:  
    IPAddress ip = WiFi.localIP();  
    Serial.print("IP Address: ");  
    Serial.println(ip);  
  
    // print the received signal strength:  
    long rssi = WiFi.RSSI();  
    Serial.print("signal strength (RSSI):");  
    Serial.print(rssi);  
    Serial.println(" dBm");  
    // print where to go in a browser:  
    Serial.print("To see this page in action, open a browser to http://");  
    Serial.println(ip);  
}
```



After observing the output in serial monitor ,copy the IPAddress(192.168.0.168) in browser and observe the web Page .



Click on HIGH/LOW button to observe the transition of LED on TIVA board.

18CSL65 - IOT LAB PROGRAMS

Program-9

Write a program for sensor based webserver to manipulate the GPIO's of WiFi Module CC3100 and monitor the sensor data connected with it.

```
#include <SPI.h>
#include <WiFi.h>

char ssid[] = "Nokia 5";
char password[] = " ";

long sensorValue = 0;
WiFiClient client;
WiFiServer server(80);
void setup()
{
    Serial.begin(9600);    // initialize serial communication
    pinMode(RED_LED, OUTPUT);    // set the LED pin mode
    Serial.print("Attempting to connect to Network named: ");
    Serial.println(ssid);
    WiFi.begin(ssid, password);
    while ( WiFi.status() != WL_CONNECTED)
    {
        // print dots while we wait to connect
        Serial.print(".");
        delay(300);
    }
    Serial.println("\nYou're connected to the network");
    Serial.println("Waiting for an ip address");

    while (WiFi.localIP() == INADDR_NONE)
    {
        // print dots while we wait for an ip addresss
        Serial.print(".");
        delay(300);
    }

    PrintWifiStatus();
    Serial.println("Starting webserver on port 80");
    server.begin();           // start the web server on port 80
    Serial.println("Webserver started!");
}

void loop()
{
```


18CSL65 - IOT LAB PROGRAMS

```
    client = server.available();
    if(client)
    {
        Serial.println("new client");
        boolean currentLineBlank=true;
        while(client.connected())
        {
            if(client.available())
            {
                char c=client.read();
                Serial.write(c);
                if(c=='\n' && currentLineBlank)
                {
                    SendResponseServer();
                    break;
                }
                if(c=='\n')
                {
                    currentLineBlank=true;
                }
                else if(c!='\r')
                {
                    currentLineBlank=false;
                }
            }
        }
        delay(1);
        client.stop();
        Serial.println("client disconnected");
    }
}

void SendResponseServer()
{
    // send a standard http response header
    client.println("HTTP/1.1 200 OK");
    client.println("Content-Type: text/html");
    client.println("Connection: close"); // the connection will be closed after completion of the
response
    client.println("Refresh: 2"); // refresh the page automatically every 2 secs
    client.println();
    client.println("<!DOCTYPE HTML>");
    client.println("<html>");
    client.println("<head><title>WiFi Web Server</title></head><body align=center>");
    client.println("<h1 align=center><font color=\"red\">Welcome to Sensor Server</font></h1>");
    // output the value of each analog input pin
```

18CSL65 - IOT LAB PROGRAMS

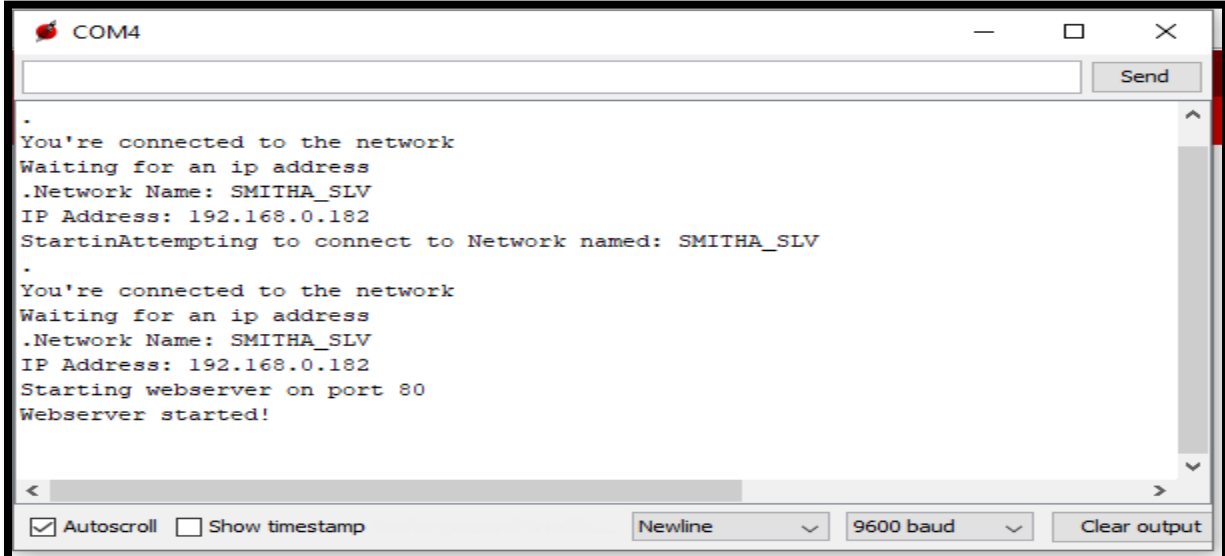
```
    sensorValue = analogRead(A0);
    client.print("Potentiometer");
    client.print(" is ");
    client.println("<mark>");
    client.println( "<h1>");
    client.println("<mark>");
    client.println(sensorValue);
    client.println("</marked>");
    client.print("</h1>");
    client.println("<br />");
    // client.println("<meta http-equiv=refresh content=1;URL='//192.168.43.173/'>");
    client.println("</html>");
}
```

```
void PrintWifiStatus() {
    // print the SSID of the network you're attached to:
    Serial.print("Network Name: ");
    Serial.println(WiFi.SSID());

    // print your WiFi shield's IP address:
    IPAddress ip = WiFi.localIP();
    Serial.print("IP Address: ");
    Serial.println(ip);
}
```

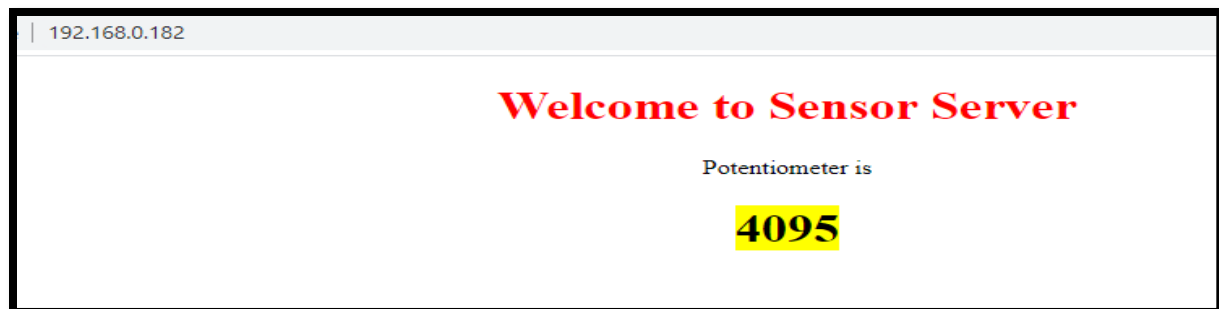
After running the above Program , the Ip address of server will be printed on serial terminal. Copying that Ip address into browser window, a Html page will be displayed with all the analog channel values.

18CSL65 - IOT LAB PROGRAMS



```
COM4
.
You're connected to the network
Waiting for an ip address
.Network Name: SMITHA_SLV
IP Address: 192.168.0.182
StartinAttempting to connect to Network named: SMITHA_SLV
.
You're connected to the network
Waiting for an ip address
.Network Name: SMITHA_SLV
IP Address: 192.168.0.182
Starting webserver on port 80
Webserver started!
```

Autoscroll ☒ Show timestamp ☐ Newline 9600 baud Clear output



18CSL65 - IOT LAB PROGRAMS

Program-10

Write a program for HTTP based webserver to control the movement of the RSLK.

/*

WiFi Web Server LED Blink

A simple web server that lets you blink an LED via the web.

This sketch will print the IP address of your WiFi (once connected)

to the Serial monitor. From there, you can open that address in a web browser to turn on and off the LED on pin 9.

If the IP address of your WiFi is yourAddress:

<http://yourAddress/H> turns the LED on

<http://yourAddress/L> turns it off

This example is written for a network using WPA encryption. For WEP or WPA, change the `Wifi.begin()` call accordingly.

Circuit:

* CC3200 WiFi LaunchPad or CC3100 WiFi BoosterPack
with TM4C or MSP430 LaunchPad

created 25 Nov 2012

by Tom Igoe

modified 6 July 2014

by Noah Luskey

*/

18CSL65 - IOT LAB PROGRAMS

```
#ifndef __CC3200R1M1RGC_  
// Do not include SPI for CC3200 LaunchPad  
#define LS 40  
#define RS 39  
#define LD 29  
#define RD 30  
  
#include <SPI.h>  
#endif  
#include <WiFi.h>  
  
// your network name also called SSID  
char ssid[] = "OPPO K1";  
// your network password  
char password[] = "";  
// your network key Index number (needed only for WEP)  
int keyIndex = 0;  
  
WiFiServer server(80);  
  
void setup() {  
  Serial.begin(115200);  // initialize serial communication  
  pinMode(RED_LED, OUTPUT);  // set the LED pin mode  
  
  // attempt to connect to Wifi network:  
  Serial.print("Attempting to connect to Network named: ");  
  // print the network name (SSID);  
  Serial.println(ssid);  
  // Connect to WPA/WPA2 network. Change this line if using open or WEP network:  
  WiFi.begin(ssid, password);  
  while ( WiFi.status() != WL_CONNECTED) {  
    // print dots while we wait to connect  
    Serial.print(".");  
    delay(300);  
  }  
  
  Serial.println("\nYou're connected to the network");  
  Serial.println("Waiting for an ip address");  
  
  while (WiFi.localIP() == INADDR_NONE) {  
    // print dots while we wait for an ip address  
    Serial.print(".");  
    delay(300);  
  }  
  
  Serial.println("\nIP Address obtained");
```

18CSL65 - IOT LAB PROGRAMS

```
// you're connected now, so print out the status
printWifiStatus();

Serial.println("Starting webserver on port 80");
server.begin();           // start the web server on port 80
Serial.println("Webserver started!");

pinMode(LD,OUTPUT);
pinMode(RD,OUTPUT);

}

void loop() {
  int i = 0;
  WiFiClient client = server.available(); // listen for incoming clients

  if (client) {             // if you get a client,
    Serial.println("new client"); // print a message out the serial port
    char buffer[150] = {0}; // make a buffer to hold incoming data
    while (client.connected()) { // loop while the client's connected
      if (client.available()) { // if there's bytes to read from the client,
        char c = client.read(); // read a byte, then
        Serial.write(c);        // print it out the serial monitor
        if (c == '\n') {        // if the byte is a newline character

          // if the current line is blank, you got two newline characters in a row.
          // that's the end of the client HTTP request, so send a response:
          if (strlen(buffer) == 0) {
            // HTTP headers always start with a response code (e.g. HTTP/1.1 200 OK)
            // and a content-type so the client knows what's coming, then a blank line:
            client.println("HTTP/1.1 200 OK");
            client.println("Content-type:text/html");
            client.println();

            // the content of the HTTP response follows the header:
            client.println("<html><head><title>Energia RSLK based WiFi Web
Server</title></head><body align=center>");
            client.println("<h1 align=center><font color='\"red\"'>Welcome to the RSLK WiFi Web
Server</font></h1>");
            client.print("ROBOT movement <button
onclick='\"location.href='/A'\">FORWARD</button>");
            client.println(" <button onclick='\"location.href='/B'\">BACKWARD</button><br>");
            client.print("Robot Movement <button onclick='\"location.href='/C'\">RIGHT</button>");
            client.println(" <button onclick='\"location.href='/D'\">LEFT</button><br>");
            client.print("Robot Movement <button onclick='\"location.href='/E'\">STOP</button>");
```

18CSL65 – IOT LAB PROGRAMS

```
// The HTTP response ends with another blank line:
client.println();
// break out of the while loop:
break;
}
else { // if you got a newline, then clear the buffer:
  memset(buffer, 0, 150);
  i = 0;
}
}
else if (c != '\r') { // if you got anything else but a carriage return character,
  buffer[i++] = c; // add it to the end of the currentLine
}

if (endsWith(buffer, "GET /A")) {
  digitalWrite(LD , LOW);
  digitalWrite(RD ,LOW);
  analogWrite(LS ,20);
  analogWrite(RS ,20);
  delay(1000);// GET /H turns the LED on
}
if (endsWith(buffer, "GET /B")) {

  digitalWrite(LD , HIGH);
  digitalWrite(RD ,HIGH);
  analogWrite(LS ,20);
  analogWrite(RS ,20);
  delay(1000);// GET /L turns the LED off
}
if (endsWith(buffer, "GET /C")) {
  digitalWrite(LD , LOW);
  digitalWrite(RD ,LOW);
  analogWrite(LS ,20);
  analogWrite(RS ,0);
  delay(1000); // GET /H turns the LED on
}
if (endsWith(buffer, "GET /D")) {
  digitalWrite(LD , LOW);
  digitalWrite(RD ,LOW);
  analogWrite(LS ,0);
  analogWrite(RS ,20);
  delay(1000); // GET /L turns the LED off
}
if (endsWith(buffer, "GET /E")) {
  analogWrite(LS ,0);
  analogWrite(RS ,0);
```


18CSL65 - IOT LAB PROGRAMS

```
        delay(1000);    // GET /L turns the LED off
    }
}
}
// close the connection:
client.stop();
Serial.println("client disconnected");
}
}

//
//a way to check if one array ends with another array
//
boolean endsWith(char* inString, const char* compString) {
    int compLength = strlen(compString);
    int strLength = strlen(inString);

    //compare the last "compLength" values of the inString
    int i;
    for (i = 0; i < compLength; i++) {
        char a = inString[(strLength - 1) - i];
        char b = compString[(compLength - 1) - i];
        if (a != b) {
            return false;
        }
    }
    return true;
}

void printWifiStatus() {
    // print the SSID of the network you're attached to:
    Serial.print("SSID: ");
    Serial.println(WiFi.SSID());

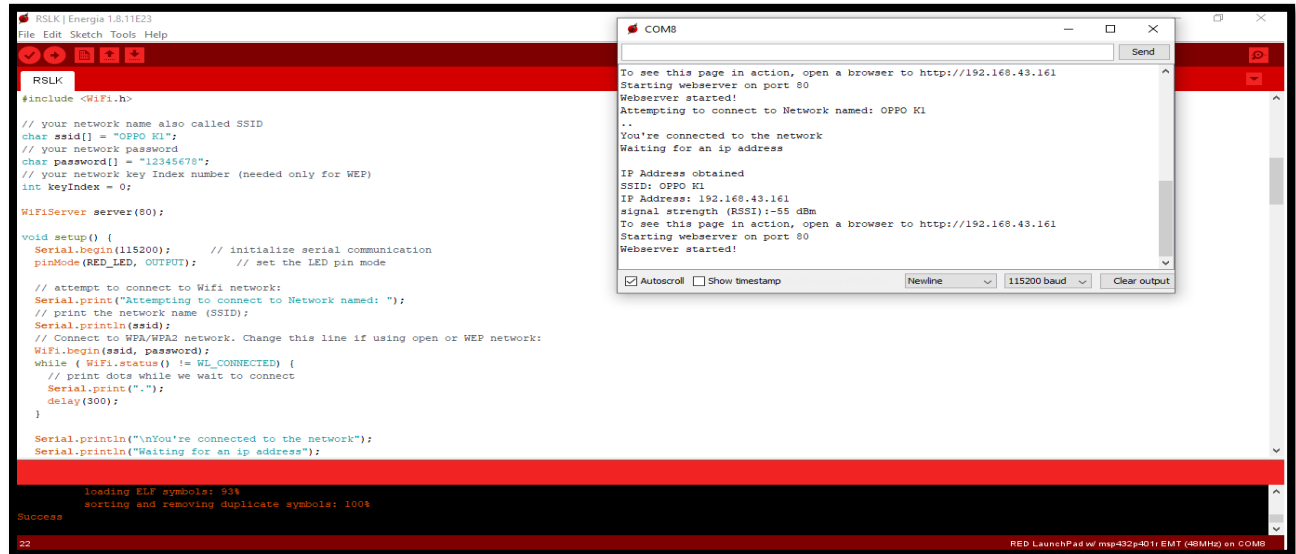
    // print your WiFi IP address:
    IPAddress ip = WiFi.localIP();
    Serial.print("IP Address: ");
    Serial.println(ip);

    // print the received signal strength:
    long rssi = WiFi.RSSI();
    Serial.print("signal strength (RSSI):");
    Serial.print(rssi);
    Serial.println(" dBm");
    // print where to go in a browser:
    Serial.print("To see this page in action, open a browser to http://");
```

18CSL65 - IOT LAB PROGRAMS

```
Serial.println(ip);  
}
```

On the Serial Monitor



Copy the IP address displayed on the serial monitor and paste it on the web browser on any other device(Computer/Phone), acting as a client and check the following



Now , click on the buttons (FORWARD, BACKWARD,RIGHT,LEFT,STOP) to control ROBOT movement.

Dr. Smitha Shekar B and Prof. Lavanya Santhosh, Dept. Of C.S.E, Dr.A.I.T.

18CSL65 – IOT LAB PROGRAMS