**Program-1**

**Write a program using different components,**

**a)LEDs for configuring and controlling different on board peripherals**

void setup()

{

pinMode(RED\_LED, OUTPUT);

}

void loop()

{

digitalWrite(RED\_LED, HIGH);

delay(1000);

digitalWrite(RED\_LED, LOW);

delay(1000);

}

**b)Switches to configure an LED as an output and a switch as an input Pull up mode**

**on TM4C123 Launchpad using Energia software**

void setup()

{

pinMode(RED\_LED, OUTPUT);

pinMode(BLUE\_LED, OUTPUT);

pinMode(PUSH1, INPUT\_PULLUP);

pinMode(PUSH2, INPUT\_PULLUP);

}

void loop()

{

int SW1 = digitalRead(PUSH1);

int SW2 = digitalRead(PUSH2);

if(SW1 == LOW)

digitalWrite(RED\_LED, HIGH);

else

digitalWrite(RED\_LED, LOW);

if(SW2 == LOW)

digitalWrite(BLUE\_LED, HIGH);

else

digitalWrite(BLUE\_LED, LOW);

}

**Program-2**

**Write a program to show**

**a) Analog to Digital Conversion and its output being monitored on the serial monitor**

void setup()

{

pinMode(A0, INPUT);

Serial.begin(9600);

}

void loop()

{

int a = analogRead(A0);

Serial.println(a);

delay(500);

}

**b) Pulse Width Modulation(PWM) for changing the intensity of the LED**

**Case 1**

void setup()

{

pinMode(RED\_LED, OUTPUT);

}

void loop()

{

int value;

for(value = 0; value < 256; value+=10)

{

analogWrite(RED\_LED, value);

delay(300);

}

}

**Case 2**

void setup()

{

pinMode(RED\_LED, OUTPUT);

}

void loop()

{

int value;

for(value = 255; value >= 0; value-=10)

{

analogWrite(RED\_LED, value);

delay(300);

}

}

**c)UART for serial communication using TM4C123 Launchpad and Energia software**

void setup()

{

Serial.begin(9600);

}

void loop()

{

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[] = "Realme C1";

char password[] = "donotconnect";

IPAddress ip(198,168,1,12);

int keyIndex = 0;

void setup()

{

Serial.begin(115200);

Serial.print("Attempting to connect to Network named: ");

Serial.println(ssid);

WiFi.begin(ssid, password);

WiFi.config(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);

}

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[]="Realme C1";

char password[]="donotconnect";

IPAddress ip;

void setup()

{

Serial.begin(9600);

Serial.print("Attempting to connect to Network named: ");

Serial.println(ssid);

WiFi.begin(ssid, password);

while ( WiFi.status() != WL\_CONNECTED)

{

Serial.print(".");

delay(300);

}

Serial.println("\nYou're connected to the network");

while (WiFi.localIP() == INADDR\_NONE) {

Serial.print(".");

delay(300);

}

ip=WiFi.localIP();

Serial.print(" Dynamic IP Address :" );

Serial.println(ip);

}

void loop()

{

}

**Program-5**

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

#include <SPI.h>

#include <WiFi.h>

char ssid[]="Realme C1";

char password[]="donotconnect";

IPAddress ip,subnet,gateway;

void setup()

{

Serial.begin(9600);

Serial.print("Attempting to connect to Network named: ");

Serial.println(ssid);

WiFi.begin(ssid, password);

while ( WiFi.status() != WL\_CONNECTED)

{

Serial.print(".");

delay(300);

}

Serial.println("\nYou're connected to the network");

while (WiFi.localIP() == INADDR\_NONE)

{

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()

{

}

**Program-6**

**Illustrate TCP based Client Server Communication Model.**

**Client**

#include <SPI.h>

#include <WiFi.h>

char ssid[]= "Realme C1";

char password[]= "donotconnect";

uint16\_t port =9999;

IPAddress server(192,168,1,122);

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);

}

}

**Server**

#include <SPI.h>

#include <WiFi.h>

IPAddress ip(192, 168, 1, 122);

char ssid[] = "Realme C1";

char password[] ="donotconnect";

unsigned char data=0;

WiFiServer server(9999);

void setup()

{

Serial.begin(9600);

Serial.print("Attempting to connect to Network named: ");

Serial.println(ssid);

WiFi.config(ip);

WiFi.begin(ssid, password);

while ( WiFi.status() != WL\_CONNECTED)

{

Serial.print(".");

delay(300);

}

Serial.println("\nYou're connected to the network");

Serial.println("Waiting for an ip address");

while (WiFi.localIP() == INADDR\_NONE)

{

Serial.print(".");

delay(300);

}

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

printWifiStatus();

server.begin();

}

void loop()

{

WiFiClient client = server.available();

if(client==true)

{

server.print("HELLO\n");

delay(500);

}

}

void printWifiStatus()

{

Serial.print("SSID: ");

Serial.println(WiFi.SSID());

IPAddress IP = WiFi.localIP();

Serial.print("IP Address: ");

Serial.println(IP);

}

**Program-7**

**Illustrate UDP based Client Server Communication Model**

**UDP Client**

#include <SPI.h>

#include <WiFi.h>

char net[] = "Realme C1";

char pass[] = "donotconnect";

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;

if(Serial.available())

{

Serial.readBytes(a,15);

Udp.beginPacket(ip,2600);

Udp.write(a);

Udp.endPacket();

for(i = 0;i<10;i++)

a[i] = ' ';

}

int packsize = Udp.parsePacket();

if(packsize)

{

memset(a,0,10);

Udp.read(a,10);

Serial.println(Udp.remoteIP());

Serial.println(Udp.remotePort());

Serial.println(a);

}

}

**UDP Server**

#include <SPI.h>

#include <WiFi.h>

char net[] = "Realme C1";

char pass[] = "donotconnect";

char a[15];

unsigned int LP = 2600;

IPAddress ip(192,168,43,155);

IPAddress ipr(192,168,43,10);

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(":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()

{

int packsize = Udp.parsePacket();

if(packsize)

{

memset(a,0,10);

Udp.read(a,10);

Serial.println(Udp.remoteIP());

Serial.println(Udp.remotePort());

Serial.println(a);

}

char i =0;

if(Serial.available())

{

Serial.readBytes(a,15);

Udp.beginPacket(ipr,5555);

Udp.write(a);

Udp.endPacket();

for(i = 0;i<10;i++)

a[i] = ' ';

}

delay(100);

}

**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\_\_ //2 unerscore, space after indef

#include <SPI.h>

#endif

#include <WiFi.h>

int m;

char ssid[] = "DRAIT Wi-Fi";

char password[] = "admin@1982";

WiFiServer server(80);

void setup() {

Serial.begin(9600);

pinMode(RED\_LED, OUTPUT);

//pinMode(40, OUTPUT);

//pinMode(39, OUTPUT);

Serial.print("Attempting to connect to Network named: ");

Serial.println(ssid);

WiFi.begin(ssid, password);

while ( WiFi.status() != WL\_CONNECTED)

{

Serial.print(".");

delay(300);

}

Serial.println("\nYou're connected to the network");

Serial.println("Waiting for an ip address");

while (WiFi.localIP() == INADDR\_NONE)

{

Serial.print(".");

delay(300);

}

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

printWifiStatus();

Serial.println("Starting webserver on port 80");

server.begin();

Serial.println("Webserver started!");

}

void loop() {

int i = 0;

WiFiClient client = server.available();

if (client)

{

Serial.println("new client");

char buffer[150] = {0};

while (client.connected())

{

if (client.available())

{

char c = client.read();

Serial.write(c);

if (c == '\n')

{

if (strlen(buffer) == 0)

{

server.println("HTTP/1.1 200 OK");

server.println("Content-type:text/html");

server.println();

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.println();

break;

}

else

{

memset(buffer, 0, 150);

i = 0;

}

}

else if (c != '\r')

{

buffer[i++] = c;

}

if (endsWith(buffer, "GET /H"))

{

digitalWrite(30,HIGH);

}

if (endsWith(buffer, "GET /L"))

{

digitalWrite(30, LOW);

}

}

}

client.stop();

Serial.println("client disonnected");

}

}

boolean endsWith(char\* inString, char\* compString)

{

int compLength = strlen(compString);

int strLength = strlen(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()

{

Serial.print("SSID: ");

Serial.println(WiFi.SSID());

IPAddress ip = WiFi.localIP();

Serial.print("IP Address: ");

Serial.println(ip);

long rssi = WiFi.RSSI();

Serial.print("signal strength (RSSI):");

Serial.print(rssi);

Serial.println(" dBm");

Serial.print("To see this page in action, open a browser to http://");

Serial.println(ip);

}

**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[] = "Realme C1";

char password[] = "donotconnect";

long sensorValue = 0;

WiFiClient client;

WiFiServer server(80);

void setup()

{

Serial.begin(9600);

pinMode(RED\_LED, OUTPUT);

Serial.print("Attempting to connect to Network named: ");

Serial.println(ssid);

WiFi.begin(ssid, password);

while ( WiFi.status() != WL\_CONNECTED)

{

Serial.print(".");

delay(300);

}

Serial.println("\nYou're connected to the network");

Serial.println("Waiting for an ip address");

while (WiFi.localIP() == INADDR\_NONE)

{

Serial.print(".");

delay(300);

}

PrintWifiStatus();

Serial.println("Starting webserver on port 80");

IPAddress ip = WiFi.localIP();

Serial.println(ip);

server.begin();

Serial.println("Webserver started!");

}

void loop()

{

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");

client.println("Refresh: 2");

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

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()

{

Serial.print("Network Name: ");

Serial.println(WiFi.SSID());

IPAddress ip = WiFi.localIP();

Serial.print("IP Address: ");

Serial.println(ip);

}