```
// Core library for code-sense - IDE-based
#include "Energia.h"
#include <WiFi.h>
#include <BMA222.h>
#include "Adafruit_TMP006.h"

BMA222 bma222;
Adafruit_TMP006 tmp006(0x41);

const float acc_resolution = 15.6/1000;

#define LED RED_LED

// Define variables and constants
char wifi_name[] = "\"microverse\"";
char wifi_password[] = "fortytwo";

// Define structures and classes
WiFiServer myServer(80); // Port 80
uint8_t oldCountClients = 0;
uint8_t countClients = 0;

boolean state = false;
const int leftButton = PUSH1;
const int rightButton = PUSH2;

bool red_led = false;

//initialize variables to store sensor readings
int button1 = 0;
int button2 = 0;


void setup()
{
// Initialize Serial
  Serial.begin(115200);
  delay(500);
  Serial.println("HERE we GO");
  pinMode(RED_LED,OUTPUT);
  red_led = false;

// Initialize Sensors
  bma222.begin();
```

```
  if(! tmp006.begin()) {
    Serial.println("Couldn't find valid TMP006 sensor.");
    while (1);
  }


//Sensor initializations
  pinMode(leftButton, INPUT_PULLUP);  //leftButton is configured as an input with
a pullup resistor
  pinMode(rightButton, INPUT_PULLUP); //rightButton is configured the same as
leftButton

  delay(500);

  Serial.println("*** LaunchPad CC3200 WiFi Web-Server in AP Mode");

  // Start WiFi and create a network with wifi_name as the network name
  // with wifi_password as the password.
  Serial.print("Starting AP... ");
  WiFi.disconnect();
  WiFi.beginNetwork(wifi_name, wifi_password);
  while (WiFi.localIP() == INADDR_NONE)
  {
    // print dots while we wait for the AP config to complete
    Serial.print('.');
    delay(300);
  }

  Serial.println("DONE");

  Serial.print("LAN name = ");
  Serial.println(wifi_name);
  Serial.print("WPA password = ");
  Serial.println(wifi_password);

  IPAddress ip = WiFi.localIP();
  Serial.print("Webserver IP address = ");
  Serial.println(ip);

  Serial.print("Web-server port = ");
  myServer.begin();                           // start the web server on port 80
  Serial.println("80");
  Serial.println();
}
```

```
// Add loop code
void loop()
{

  WiFiClient myClient = myServer.available();

//  check_if_anything_is_connected();

  if (myClient)
  { // if you get a client,
    Serial.println("Client Connected to server!");
    Serial.println("Collecting data! (and blinknig RED)");
    Serial.println("");

// collect data as long as a client is connected to the server
    while (myClient.connected())
    {
      digitalWrite(RED_LED,red_led);
      myClient.print( tmp006.readObjTempC()); myClient.print(","); //Prints temp

      myClient.print( bma222.readXData()*acc_resolution); myClient.println();

      myClient.print(millis());               //Returns the number of
milliseconds since the Microcontroller was powered up
      myClient.print(",");
  myClient.print(digitalRead(leftButton));  //digitalRead returns a 1 if the
button is pressed, otherwise it returns a 0
  myClient.print(",");
  myClient.print(digitalRead(rightButton));
  myClient.print(",");
  myClient.print(bma222.readXData()*acc_resolution);       //Returns current
reading from the accelerometer's x-axis
  myClient.print(",");
  myClient.print(bma222.readYData()*acc_resolution);       //Returns current
reading from the accelerometer's y-axis
  myClient.print(",");
  myClient.print(bma222.readZData()*acc_resolution);       //Returns current
reading from the accelerometer's z-axis
  myClient.print(",");
  // This is the tempurature reading for what is in front of the sensor
  float objt = tmp006.readObjTempC();     //Stores output from Object temperature
sensor in degrees Celsius;
  myClient.print(objt);                     //Prints value stored in the previous
line to the serial port
  myClient.print(",");
```

```
  // This is the Ambient tempurature of the tempurature sensor
  float diet = tmp006.readDieTempC();     //Stores output from Ambient
temperature sensor in degrees Celsius;
  myClient.println(diet);

      delay(20);                            //This is used to delay time between
taking sensor readings
      //Check to see if we are still connected
      myClient = myServer.available();
      red_led = !red_led;
    }
// client disconnected. Close server.
    myClient.stop();
    Serial.println("That's it! (and not blinknig RED");
    Serial.println("Client disconnected from server");
    Serial.println();
  }
}
```