

```

1  //
2  //
3  //  Desarrolladores de este código:
4  //
5  //    - Xavier Vilanova Barrio - 2022
6  //
7  //    - Marc Mora Riera - 2022
8  //
9  //
10 //  Este programa utiliza de base el Programa Sensores  de: JUAN PALOMARES MOYANO -
    2019
11 //
12 //  No se permite la distribución, la utilización o la venta de este código sin el
    permiso de los desarrolladores
13 //
14 //
15 /*~~~~~*/
16 /*~~~~~LIBRERÍAS~~~~~*/
17 /*~~~~~*/
18
19 //Conexionados UART e I2C
20 #include <Wire.h>
21 #include "SoftwareSerial.h"
22 #include "WiFi.h"
23 #include "ESPAsyncWebServer.h"
24 //IMU
25 #include "MPU6050_6Axis_MotionApps20.h" //Mediante esta biblioteca se pueden obtener
    datos del DMP (Digital Motion Processor) de la IMU
26 #include "I2Cdev.h" //Mediante esta biblioteca se pueden obtener datos del DMP de la
    IMU
27 #include "helper_3dmath.h" //Mediante esta biblioteca se pueden realizar operaciones
    de cuaterniones
28 #define MPU9250_ADDRESS 0x68 //Es la MPU6050 Pero definida amb un altre nom
29
30 /*~~~~~*/
31 /*~~~~DEFINICIÓN DE VARIABLES~~~~*/
32 /*~~~~~Y CONSTANTES~~~~~*/
33 /*~~~~~*/
34 const char* ssid = "ESP32-Access-Point";
35 const char* password = "123456789";
36 const float pi = 3.141592653;
37
38 AsyncWebServer server(80);
39
40 //IMU
41 const int mpuAddress = 0x68; // Se define la dirección de
    la IMU, puede ser 0x68 o 0x69
42 MPU6050 mpu(mpuAddress); //Se crea un objeto MPU6050
    para poder extraer datos de la DMP
43 int fifoCount = 0, packetSize; //Se crea un contador de
    valores en el FIFO de la MPU y el tamaño del paquete que obtendremos
44 byte fifoBuffer[42]; //Se crea el buffer que se
    utilizará para obtener los datos roll, pitch y yaw.
45 float roll, pitch, yaw, sqx, sqy, sqz, sqw, test; //Se crea tanto las
    variables de roll, pitch y yaw como las variables "float"
46 bool inici = false; //Variable booleana que
    decidira quan enviem les dades
47 int16_t ax, ay, az; //intermedias necesarias
    para obtenerlas
48 long espera = 50; //Tiempo Espera de
    enviamientos de datos en ms
49 Quaternion q; //Se crea el cuaternión
    necesario para obtener los valores de pitch, yaw y roll
50 int mostra = 0;
51 int boto = 15; //Variable que sirve para
    definir el pin del boton
52 int mostra;
53 float ax_m_s2, ay_m_s2, az_m_s2;
54 int16_t gx, gy, gz;
55
56
57
58

```

```

59  /* ~~~~~*/
60  /*~~~~~SETUP~~~~~*/
61  /*~~~~~*/
62  //Función que se devuelve lo que queremos enviar
63  String readAcc() {
64
65      //Serial.println("enviament");
66      return String((String(yaw) + "/" + String(pitch) + "/" + String(roll) + "/" +
        String(q.x) + "/" + String(q.y) + "/" + String(q.z) + "/" + String(q.w) + "/" +
        String(ax) + "/" + String(ay) + "/" + String(az) + "/" + String(gx) + "/" +
        String(gy) + "/" + String(gz) + "/" + String(mostra)));
67  }
68
69  void setup()
70  {
71      //PUERTO SERIE
72      Serial.begin(115200); //abrimos el puerto serie
73      pinMode (boto, INPUT_PULLDOWN); //definimos el boton
74      Wire.begin(); //Se inicializa la librería Wire.h
75
76      //IMU
77      mpu.initialize(); //Inicialización de la imu
78      mpu.dmpInitialize(); //Inicialización del DMP
79      mpu.setDMPEnabled(true); //Habilitación del DMP
80      packetSize = mpu.dmpGetFIFOPacketSize(); //Se obtiene el tamaño del paquete que
        obtendremos del DMP (42)
81
82      //Se introducen los offsets calculados mediante MPU6050_calibration
83      mpu.setXAccelOffset(5444);
84      mpu.setYAccelOffset(6328);
85      mpu.setZAccelOffset(9715);
86      mpu.setXGyroOffset(5);
87      mpu.setYGyroOffset(-13);
88      mpu.setZGyroOffset(37);
89
90      //Se define el cuaternión inicial de actitud
91      q.w = 1;
92      q.x = 0;
93      q.y = 0;
94      q.z = 0;
95      Serial.println("Debug");
96
97      // Setting the ESP as an access point
98      Serial.print("Setting AP (Access Point)...");
99      // Remove the password parameter, if you want the AP (Access Point) to be open
100      WiFi.softAP(ssid, password);
101
102      IPAddress IP = WiFi.softAPIP();
103
104      Serial.print("AP IP address: ");
105      Serial.println(IP);
106
107      server.on("/Acceleracio", HTTP_GET, [] (AsyncWebServerRequest * request) {
108          request->send_P(200, "text/plain", readAcc().c_str());
109      });
110
111      bool status;
112      // Start server
113      server.begin();
114  }
115
116  void loop()
117  {
118      IMU();
119  }
120

```