



Pseudocódigo PIC

/*////////////////////////////////////

 $\ast/$

/*////////////////////////////////////

*/

```
char read_value;
```

/*////////////////////////////////////

 $\ast/$

}

```

if(PIR1bits.TXIF == HIGH){

    //Envio los valores del acelerometro cada vez que se enciende la bandera
    //de la interrupcion

    TXREG = sensor_value[cont];

    if (cont == 5){

        cont = 0;

    }

    else{

        cont++;

    }

}

}

```

```

/*//////////////////////////////////////

```

```

* CICLO PRINCIPAL

```

```

*/

```

```

void main(void) {

```

```

    //Inicio el PIC

```

```

    SYSTEM_Initialize();

```

```

    while(1){

```

```

        if(read_value == 1){

```

```

            PORTD0 = 1;

```

```

        }

```

```

        else if(read_value == 2){

```

```

            PORTD0 = 0;

```

```

        }

```

```

        else if(read_value == 3){

```

```

            PORTD1 = 1;

```

```

    }
    else if(read_value == 4){
        PORTD1 = 0;
    }

    //Obtengos los valores del acelerometro y los guardo en un array
    sensor_value[0] = leer_adxl(posición_x0);
    sensor_value[1] = leer_adxl (posición_x1);

    sensor_value[2] = leer_adxl (posición_y0);
    sensor_value[3] = leer_adxl (posición_y1);

    sensor_value[4] = leer_adxl (posición_z0);
    sensor_value[5] = leer_adxl(posición_z1);

    }
}

----- Fin del Código -----

```

Pseudocódigo ESP32

```

----- Inicio del Código -----

#define IO_LOOP_DELAY 15000

unsigned long lastUpdate;

char data_sen0;

int count;

char data_sen[6];

```

```
char *val1;
```

```
char *val2;
```

```
void setup() {
```

```
    // Note the format for setting a serial port is as follows: Serial2.begin(baud-rate, protocol, RX pin,  
    TX pin);
```

```
    Serial.begin(9600);
```

```
    Funcion_Para_Conectarse_Adafruit()
```

```
}
```

```
}
```

```
void loop() {
```

```
    //Leo los valores enviados por el PIC
```

```
    for(int i = 0; i <= 5; i++){
```

```
        if(Serial.available()){
```

```
            data_sen[i] = Serial.read();
```

```
        }
```

```
    }
```

```
    //Reviso si ya paso el tiempo mminimo para actualizar los datos en adafruit
```

```
    if (millis() > (lastUpdate + IO_LOOP_DELAY)) {
```

```
        //Envio los datos del sensor al servidor en adafruit
```

```
        xaxis->save(data_sen[0]);
```

```
        yaxis->save(data_sen[2]);
```

```
        zaxis->save(data_sen[4]);
```

```
        //Obtengo los datos de los botones del servidor en adafruit
```

```
    boton1->get();  
    boton2->get();  
  
    //Le envio los nuevos datos al PIC  
    Serial.write(boton1)  
    Serial.write(boton2)  
    // update timer  
    lastUpdate = millis();  
}  
}
```

----- Fin del Código -----