**PSEUDOCODIGOS**

**PARA PIC16F887**

uint8\_t s, h, m, s\_u, s\_d, m\_u, m\_d, h\_u, h\_d, EstadoPiloto;

char time[];

void main(void) {

Puertos como salidas

Activar interrupciones

EUSART\_conf();

I2C\_Master\_Init(100000);

RTC\_conf();

SetClock() con dato inicial del reloj

EstadoPiloto = 0;

while (1) {

I2C\_Master\_Start();

I2C\_Master\_Write(S\_Add\_W); //Le mando el byte de direccion del RTC en modo Write

I2C\_Master\_Write(0x00); /\* Le envio la direccion a la cual deseo apuntar, que

en este caso es la 0 ya que ahi se guarda el dato de

los segundos segun la datasheet\*/

I2C\_Master\_RepeatedStart();

I2C\_Master\_Write(S\_Add\_R); //Le mando el byte de direccion del RTC en modo Read para leer los datos

s = I2C\_Master\_Read(1);

s &= 0b01111111;

m = I2C\_Master\_Read(1);

m &= 0b01111111;

h = I2C\_Master\_Read(0);

h &= 0b00111111;

I2C\_Master\_Stop();

\_\_delay\_ms(200);

// SendString("Reloj "+Decena(h)+Unidad(h)+":");

// if (EstadoPiloto) {

SendChar(10);

SendChar(Decena(h));

SendChar(Unidad(h));

SendString(":");

SendChar(Decena(m));

SendChar(Unidad(m));

SendString(":");

SendChar(Decena(s));

SendChar(Unidad(s));

\_\_delay\_ms(200);

// }

if (EstadoPiloto == 'A') {

LedR = 1;

LedV = 1;

} else if (EstadoPiloto == 'B') {

LedR = 1;

LedV = 0;

} else if (EstadoPiloto == 'C') {

LedR = 0;

LedV = 1;

} else if (EstadoPiloto == 'D') {

LedR = 0;

LedV = 0;

} else {

LedR = LedR;

LedV = LedV;

}

// PORTA = EstadoPiloto;

uint8\_t Decena(uint8\_t valor) {

return (valor >> 4) + 48;

}

uint8\_t Unidad(uint8\_t valor) {

return (valor & 0x0F) + 48;

}

void SetClock(void) {

I2C\_Master\_Start();

I2C\_Master\_Write(S\_Add\_W);

I2C\_Master\_Write(0x00);

// I2C\_Master\_RepeatedStart();

I2C\_Master\_Write(0x00);

I2C\_Master\_Write(0x30);

I2C\_Master\_Write(0b00011001);

I2C\_Master\_Stop();

}

void \_\_interrupt() isr(void) {

if (PIR1bits.RCIF == 1) {

EstadoPiloto = Receive(); //Aqui recibimos el dato de la recepcion

PIR1bits.RCIF = 0;

}

}

**PARA ESP32**

Crear variables

uint8\_t Hora;

char datos[8];

String Time;

String ValorR;

String ValorV;

Conectarse con cada feed

AdafruitIO\_Feed \*HoraFeed = io.feed("HoraPrueba");

AdafruitIO\_Feed \*RojaFeed = io.feed("LedR");

AdafruitIO\_Feed \*VerdeFeed = io.feed("LedV");

void setup() {

// start the serial connection

Serial.begin(115200);

Serial2.begin(9600,SERIAL\_8N1,16,17);

Esperar conexion serial

RojaFeed->onMessage(handleMessage);

VerdeFeed->onMessage(handleMessage1);

Obtener datos de la nube

RojaFeed->get();

VerdeFeed->get()

void loop()

io.run();

//while(Serial2.read()!=22);

if(Serial2.available()>0){

Time = "Reloj ";

Serial2.readBytesUntil(10, datos, 8);

// Serial2.write(0);

Time = Time + datos;

}

//Codigo para enviar estados de piloto

if ((ValorR=="ON") & (ValorV=="ON")){

Serial2.print('A');

Serial.println("A");

}

else if ((ValorR=="ON") & (ValorV=="OFF")){

Serial2.print('B');

Serial.println("B");

}

else if ((ValorR=="OFF") & (ValorV=="ON")){

Serial2.print('C');

Serial.println("C");

}

else if ((ValorR=="OFF") & (ValorV=="OFF")){

Serial2.print('D');

Serial.println("D");

}

// save count to the 'counter' feed on Adafruit IO

// Serial.print("sending -> ");

HoraFeed->save(Time);

// increment the count by 1

Hora++;

}

void handleMessage(AdafruitIO\_Data \*data) {

//Serial.print("received <- ");

Serial.println(data->value());

ValorR = data->value();

}

void handleMessage1(AdafruitIO\_Data \*data) {

//Serial.print("received <- ");

Serial.println(data->value());

ValorV = data->value();

}