

## Mini Proyecto 2

- Pseudocódigo

### PIC16F887

//LIBRERIAS

xc.h

stdint.h

LCD.h (pruebas proteus)

I2C.h

USART.h

//CONFIG IO

Configurar PINES como entradas o o salidas

- RD0 y RD1 salidas digitales (luces piloto)
- ANSEL y ANSELH = 0 (no hay pines analógicos)
- Ponemos todos los puertos en 0

//CONFIG USART

- Configuramos interrupciones (Para USART)
- Baud rate de 9600 a 8MHz
- TXEN enable
- Comunicación asíncrona

//INTERRUPCIONES

// MAIN (void y loop)

- **Void**
- Inicializamos I2C (TRISC3 y TRISC4 = 1) con una frec de 100000
- En main cargamos datos iniciales (hora y fecha)
- Inicializamos LCD (pruebas)
- **Loop**
- Leemos nuevos datos
- Realizamos conversión BCD a Decimal (la respuesta del RTC es en BCD)
- Separamos los dígitos de cada dato (la respuesta del RTC es de 2 dígitos por dato)
- Generamos char para envía por USART (y a la LCD para las pruebas)
- Actualizamos nuevos datos
- Enviamos datos I2C por USART

## ESP32

//LIBRERIAS  
config.h

//config.h

- Agregar Usuario y Key de Adafruit
- Agregar Info WFI

//DEFINICIONES

- Variables para enviar y recibir datos por USART
- Luces piloto en PINES 2 y 5
- RX2 y TX2 en pines 16 y 17
- Delay de 5000
- Agregar los Feeds de Adafruit (piloto1, piloto2, sensor)

//SETUP

- PIN 2 y PIN 5 outputs
- Comunicación Serial a 115200
- Nos conectamos a adafruit
- Definimos cuales datos vamos a recibir
- Verificamos que se haya hecho la conexión
- Al estar conectados recibimos datos (recibirFeed -> get();)

//LOOP

- Escribir en el puerto Serial2 lo que se lea en el puerto Serial (Enviar datos del ESP32 al PIC)
- Escribir en el puerto Serial lo que se lea en el puerto Serial2 (del ESP32 a Adafruit)
- Imprimir los datos que se leyeron y guardarlos en una variable
- Iniciamos proceso para enviar y recibir datos de Adafruit
- Enviamos los datos del ESP32 a Adafruit
- Revisar el estado de los datos recibidos para encender o apagar los pines de los LEDs

