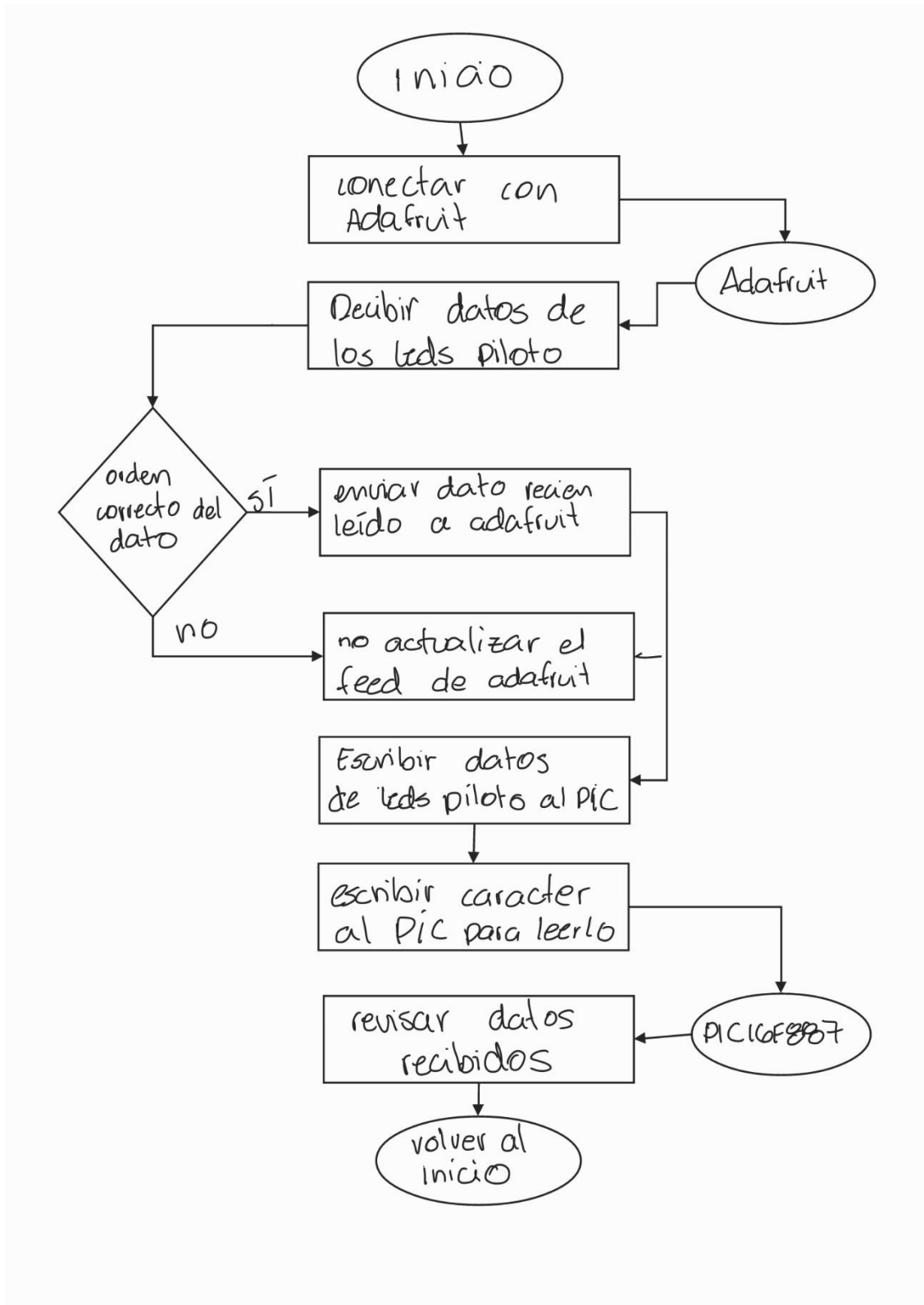
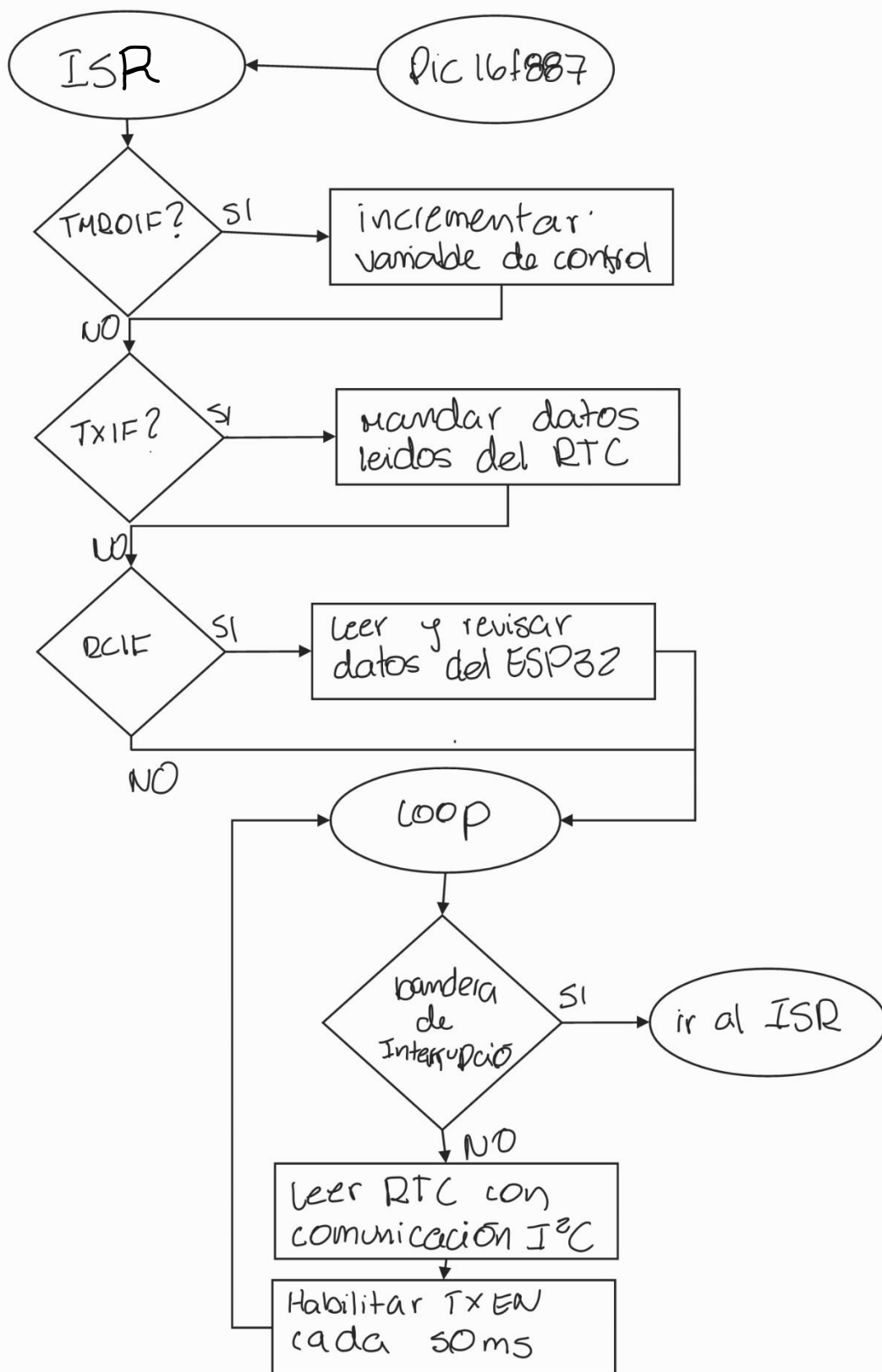
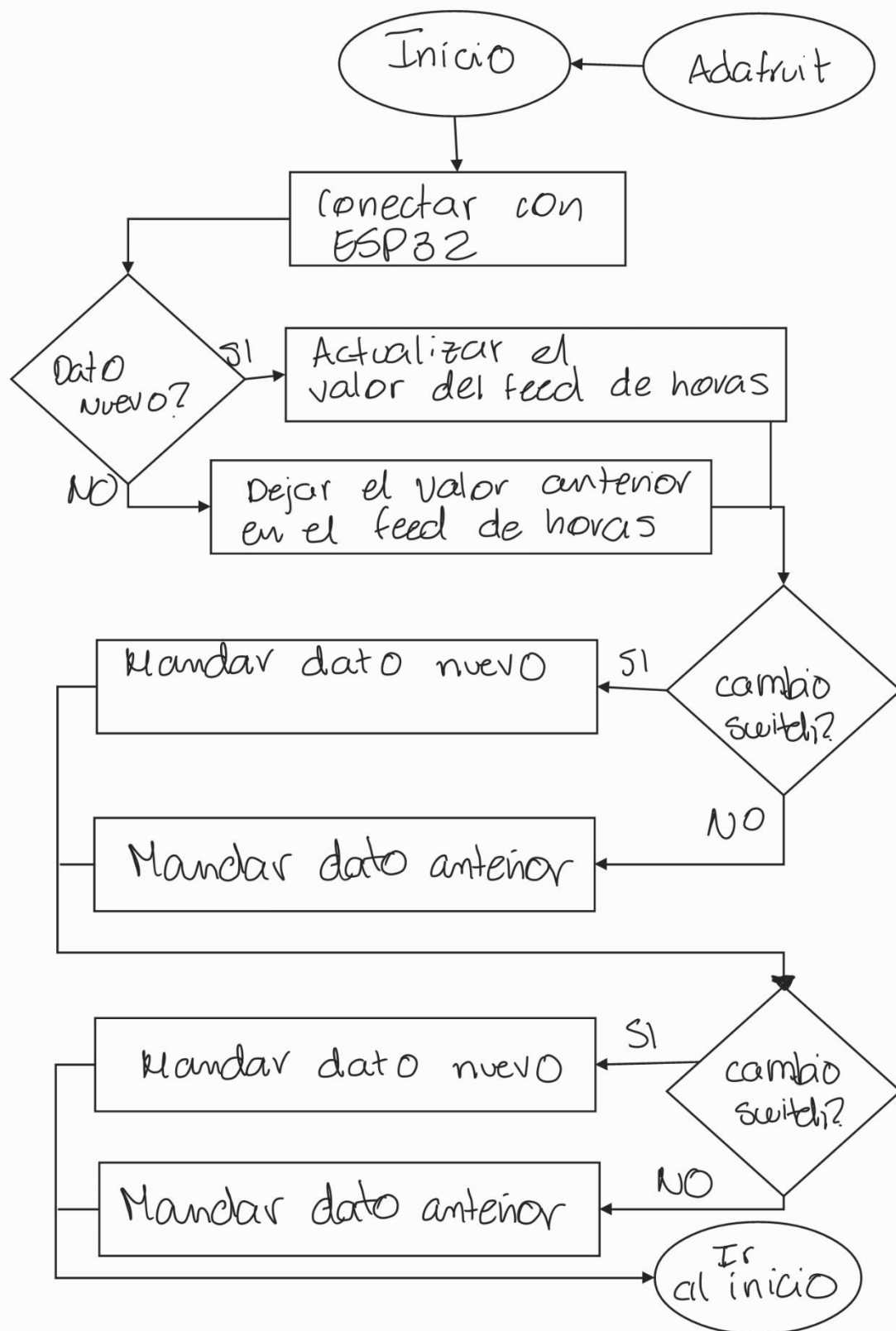


Diagrama de flujo del proyecto:







Seudocódigos de los microcontroladores involucrados:

- **PIC16F887**

1. Configurar las interrupciones que se utilizaran:
 - Timer0: Configurar el desborde cada 5ms para que incremente una variable de control.
 - TXREG: Interrupción para enviar el ESP32 los datos leídos del RTC por comunicación I2C.
 - RCREG: Recibir datos del ESP32. Se reciben 4 datos, de los cuales 2 corresponden a el estado de los LEDS piloto, la confirmación para mandar los datos y un enter para verificar que la cadena se haya recibido correctamente.
2. Llamar la configuración inicial donde se configura todos los registros que se van a utilizar, llamar a las librerías de configuración del UART, la librería de configuración del oscilador, la librería de configuración del I2C y las entradas y salidas respectivas.
3. Darle las condiciones iniciales al sensor (únicamente debe realizarse una vez).
4. Configurar el ciclo principal
 - Habilitar la interrupción del TX cada vez que la variable de control llegue a 10.
 - Leer los valores del sensor a través de comunicación I2C.

- **ESP32**

1. Configurar el serial0 para comunicar al ESP32 con la computadora y el serial2 para comunicar al ESP32 con el PIC16F887.
2. Establecer comunicación con Adafruit
3. Recibir los valores de los feeds de LEDs piloto en Adafruit
4. Enviar a Adafruit los valores recibidos por el puerto serial que vienen del PIC
5. Mandar 4 caracteres al PIC donde los primeros 2 corresponden al estado de los LEDs, el tercero corresponde al enable para que el PIC envíe datos y un enter como verificación.
6. Leer los datos que el PIC envía
7. Verificar que los datos recibidos estén en el orden correcto

Enlace de repositorio en Github:

- https://github.com/gil19443/Digital_2.git

Enlace del video en youtube:

- https://youtu.be/XCv_0o83ifQ