Electrónica Digital 2

Trabajo Práctico Final

# 

# Sensado de temperatura, control de encendido/apagado de aires acondicionados y comunicación serie con PC para muestra de información

# 

# Integrantes:

## Pasolli, Nestor Jeremias

## Weihmüller, Facundo

# Profesor:

## Petrashin, Pablo

# ¿De qué trata nuestro proyecto?

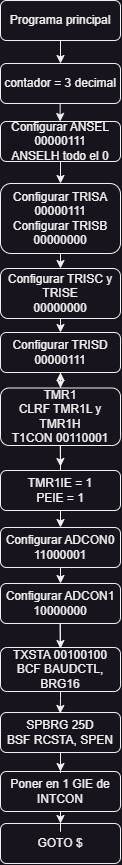
Nuestro proyecto implementa el microcontrolador PIC16F887 y consiste en un sistema que simula la automatización de encendido de 3 aires acondicionados en 3 habitaciones distintas de una casa o departamento, y a su vez nos permite visualizar la temperatura en el momento de cada una de ellas a través de un programa en Python en una computadora.

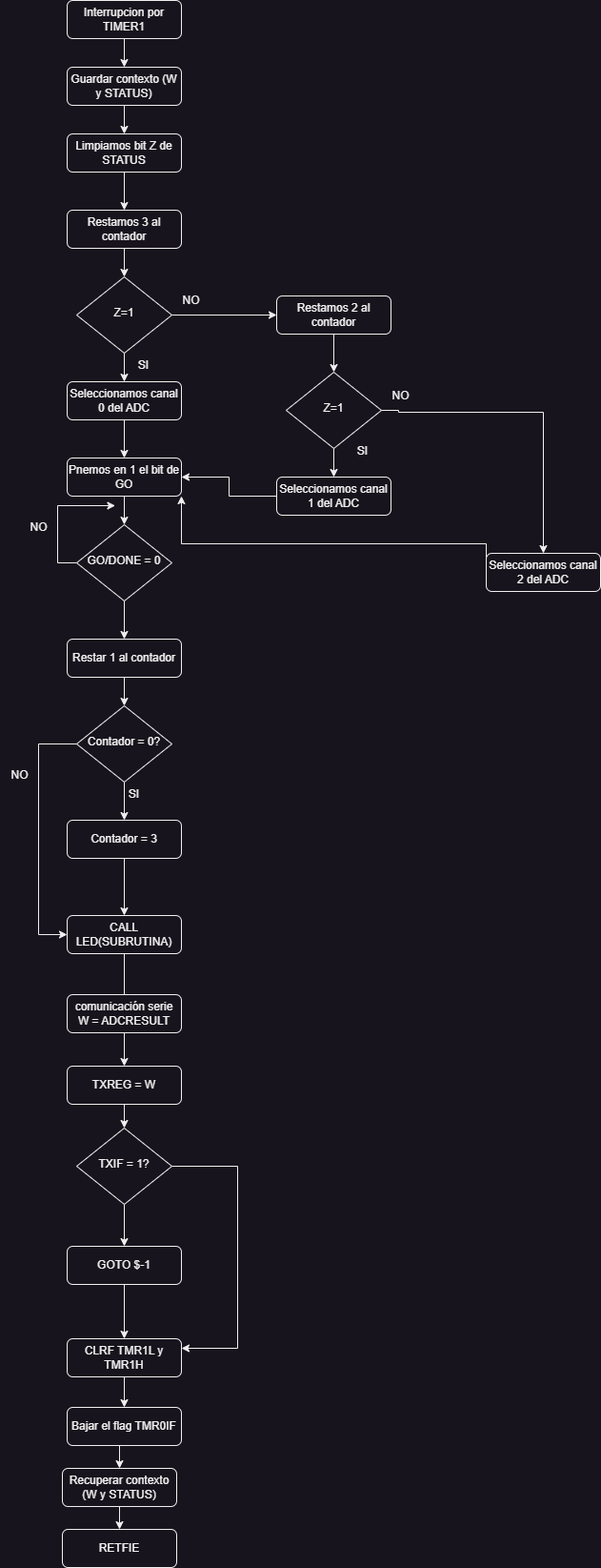
Para lograr nuestro objetivo, utilizamos 3 sensores de temperatura LM35, y con ayuda del ADC del microcontrolador, recibimos la información para trabajarla en forma digital. Para ser más específicos, cada sensor está conectado a un canal del conversor, el cual funciona en loops de 15 segundos por TIMER 1, seleccionando cada 5 segundos el canal del sensor siguiente. De esta manera se actualiza el valor de temperatura de cada habitación cada 15 segundos.

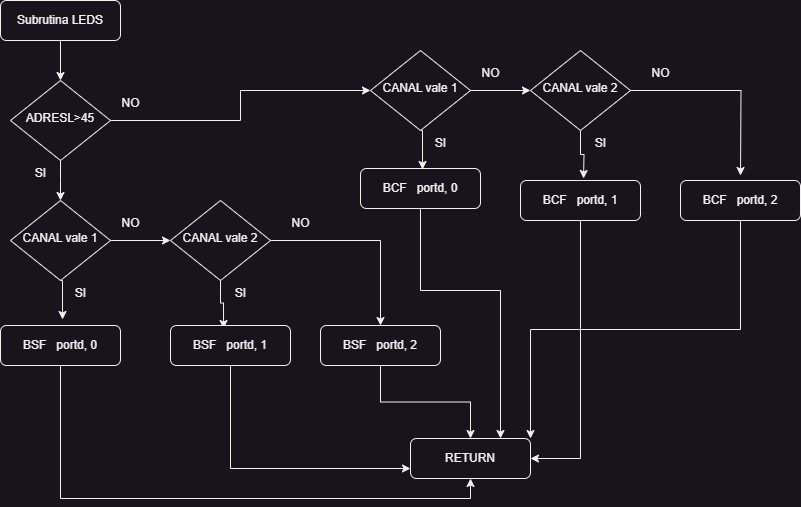
Posteriormente, se desarrolló un algoritmo en lenguaje assembler para que los aires acondicionados (simbolizados con 3 LEDs color rojo) se prendan individualmente cuando la temperatura en su respectiva habitación sea mayor a 30°C, o se apaguen en caso contrario si es que estaban encendidos. Además, como se mencionó anteriormente, podemos visualizar la temperatura de cada habitación en todo momento a través de nuestra PC. Para esto se utilizó un módulo de comunicación serie, que permite recibir los valores en binario (de a 1 Byte por vez), mediante una transmisión serie de tipo asíncrona. Como un detalle adicional, se instalaron 3 LEDs más pequeños, uno al lado de cada sensor, para indicar que canal (sensor) es el que está enviando los datos al ADC en cada momento.

# Desarrollo

El programa en assembler se corresponde con el siguiente diagrama de flujo (programa principal y subrutinas):



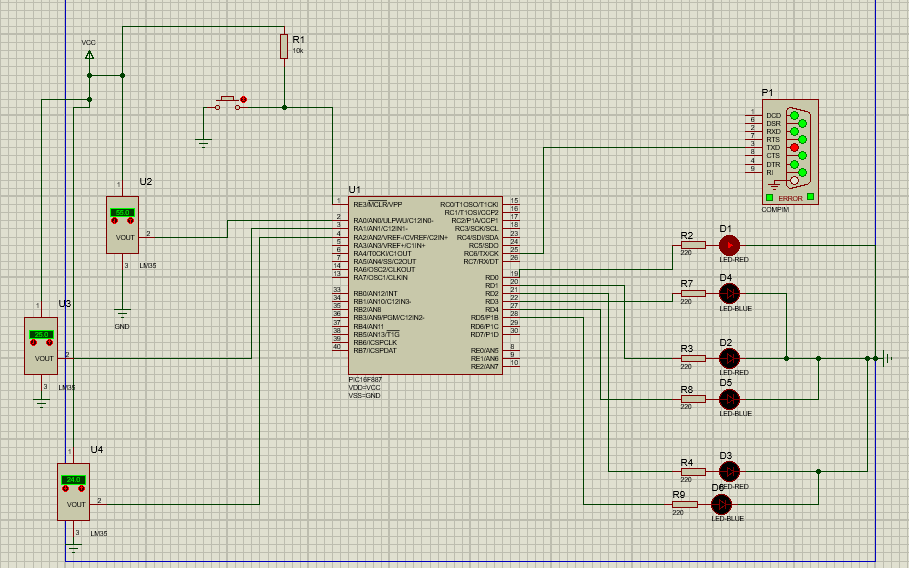




Guiados por estos diagramas implementamos el código en assembler para controlar el PIC.

Para modelar el circuito en una primera instancia utilizamos el software Proteus. Este modelo contiene los mismos componentes que el circuito físico realizado posteriormente, y nos permitió testear y debuggear el código a cargar en el microcontrolador. Además, simulando puertos virtuales RS232 comunicamos el circuito (simulado) con la PC, para la siguiente etapa de desarrollo del script en Python para recibir, procesar y mostrar los datos obtenidos de los sensores.

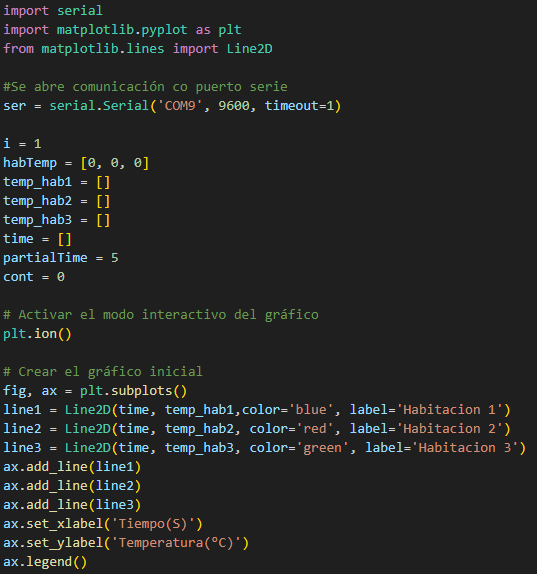
El circuito simulado en proteus fue el siguiente:

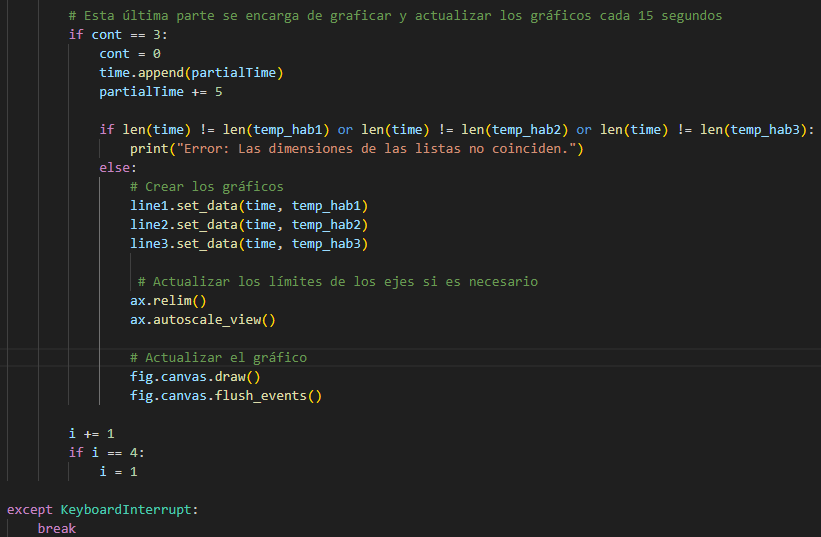


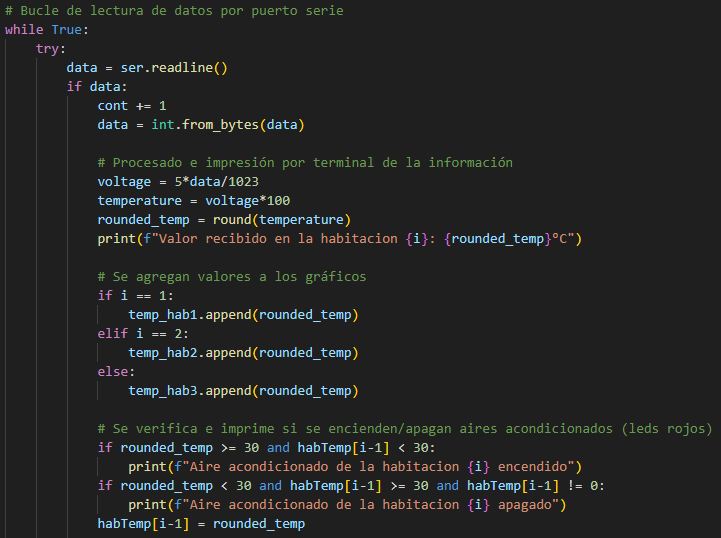
Donde podemos ver los componentes utilizados y su conexión con los distintos pines del PIC. Los leds D1, D2 y D3 se corresponden a los leds rojos que indican si se encienden o apagan los aires acondicionados de las habitaciones. Los otros tres leds son simplemente indicadores que nos permiten saber con qué canal está trabajando el ADC en cada momento (en el caso de la simulación son azules). Los tres sensores de la izquierda son los lm35, y el módulo de arriba a la derecha simula un adaptador a RS232 para la comunicación serie con la PC.

Hecho el circuito en la simulación y habiendo corroborado su correcto funcionamiento, procedimos a la implementación de un código en lenguaje Python para procesar la información transmitida por nuestro PIC y poder mostrarla en pantalla.

El código es el siguiente:



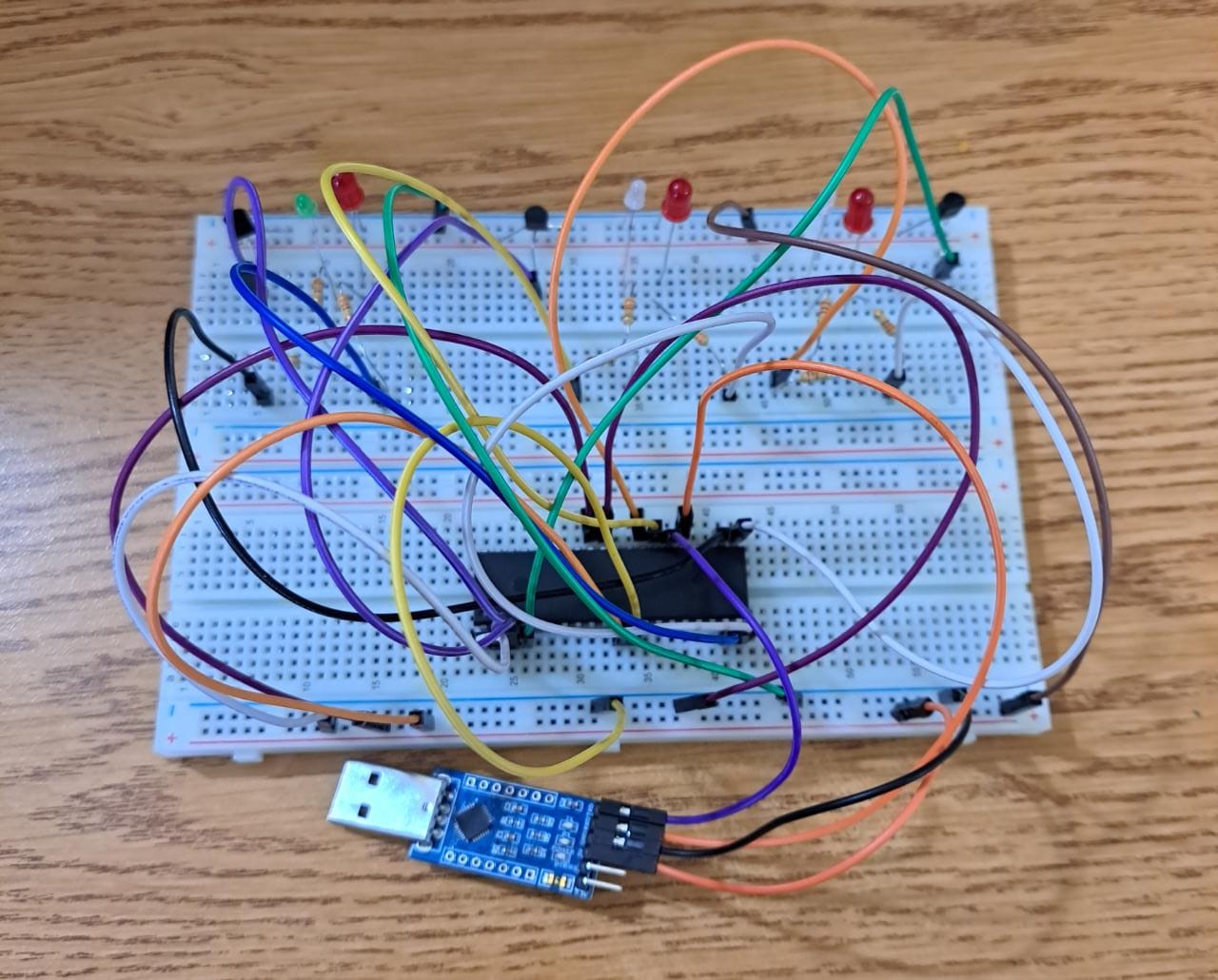




Lo que hace este script es recibir la información del PIC, que en nuestro caso serán los 8 bits menos significativos del resultado de conversión del ADC, procesarla para transformarla en un valor de temperatura, imprimir por consola ese valor y si corresponde encender o apagar el aire, y finalmente graficarlo.

Por lo tanto, tendremos una gráfica de las temperaturas de las 3 habitaciones y su evolución a lo largo del tiempo, permitiéndonos visualizar en tiempo real la evolución de las temperaturas.

Finalmente, una vez que todo esto funcionó correctamente en la simulación, procedimos a armar el circuito físico en protoboard. El circuito es el siguiente:



Hubo algunos problemas durante el armado del circuito, principalmente por la comunicación serie y el adaptador USB que utilizamos, ya que en un primer momento la PC no reconocía correctamente el adaptador USB. Esto se solucionó mediante la instalación de drivers específicos del fabricante del adaptador. Testeamos el correcto funcionamiento del circuito calentando y enfriando los sensores de temperatura, y observando cómo reaccionaba ante esto el sistema (variaciones de temperatura en las gráficas y encendido/apagado de los leds rojos).