



MICROCONTROLADORES

Práctica No. 20. Convertidor Analógico - Digital.

1. Objetivo

- Uso del convertidor analógico-digital del STM32F103.

2. Material y Equipo.

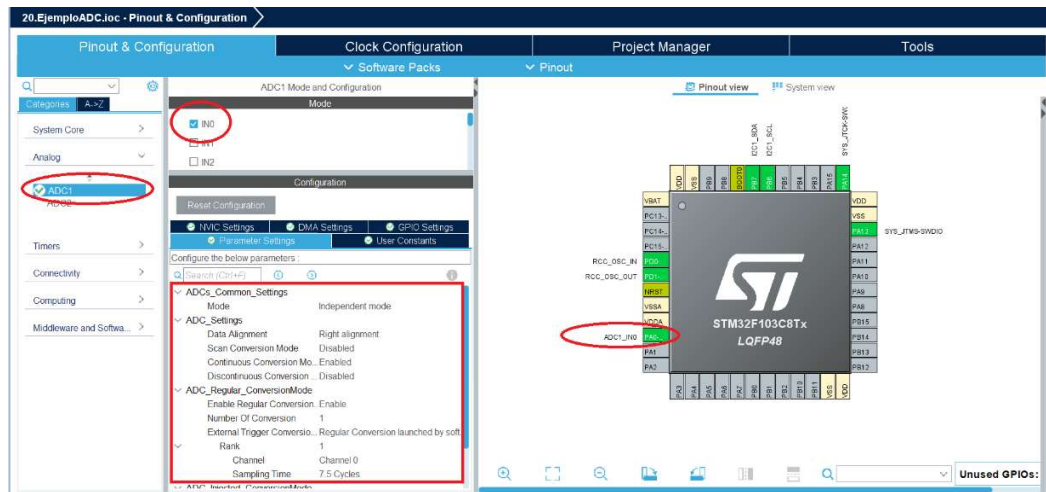
- Computador o laptop con STM32CubeIDE.
- Un Sensor de temperatura LM35.
- Una modulo PCF8574 con LCD.

3. Marco de Referencia.

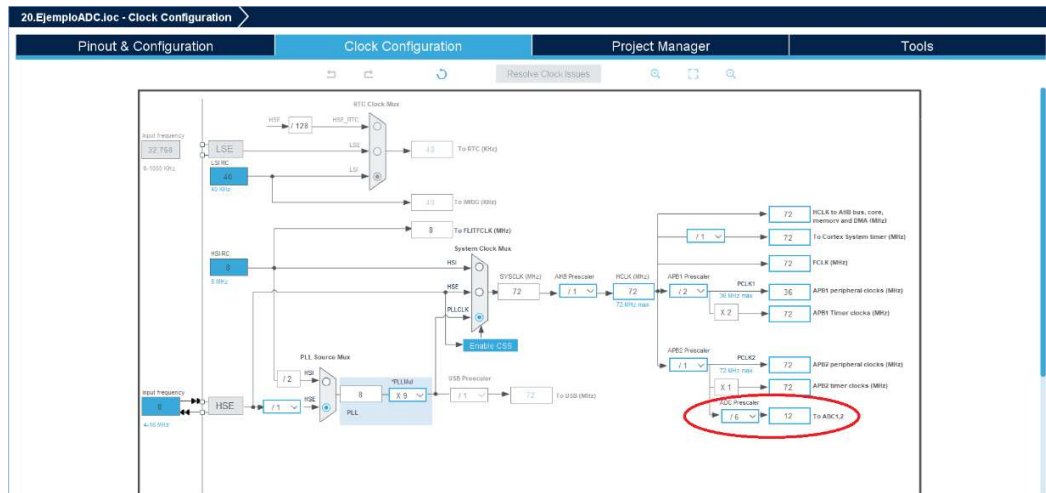
El convertidor analógico-digital o ADC es un módulo interno del STM32F103 que nos permite tomar señales analógicas externas y obtener su equivalente digital. La resolución del ADC del microcontrolador es de 12 bits con lo cual podremos expresar una conversión en el rango de 0 a 4095. El sensor LM35 es un sensor analógico que nos entrega un valor de 10mV/°C con lo que podemos hacer un termómetro de 0 a 100 °C. Para tomar la señal del sensor tenemos que configurar el ADC lo primero es configurar el modo de trabajo "Independent mode" en el STM32CubeMX, después configuramos el preescaler del ADC a seis en la pestaña "Clock Configuration" que nos permite darle tiempo al ADC para realizar la conversión, después seleccionamos el canal por el cual se tomara la lectura del sensor que es el canal 0 del ADC1 marcado como "IN0" en la opción "Analog/ADC1" justificamos a la derecha el valor resultante.

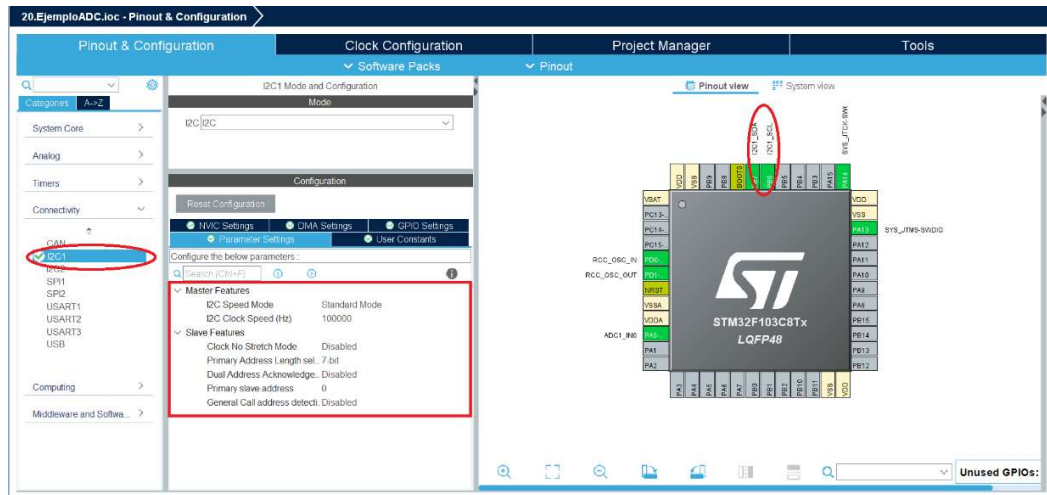
4. Desarrollo y Procedimiento.

Se creará un proyecto en el STM32CubeIDE como se indicó anteriormente.



Configuramos el prescaler del ADC como se muestra a continuación.



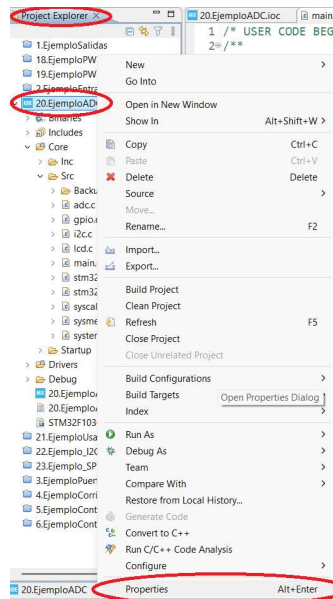


El código de la práctica es el siguiente en el archivo main.c.

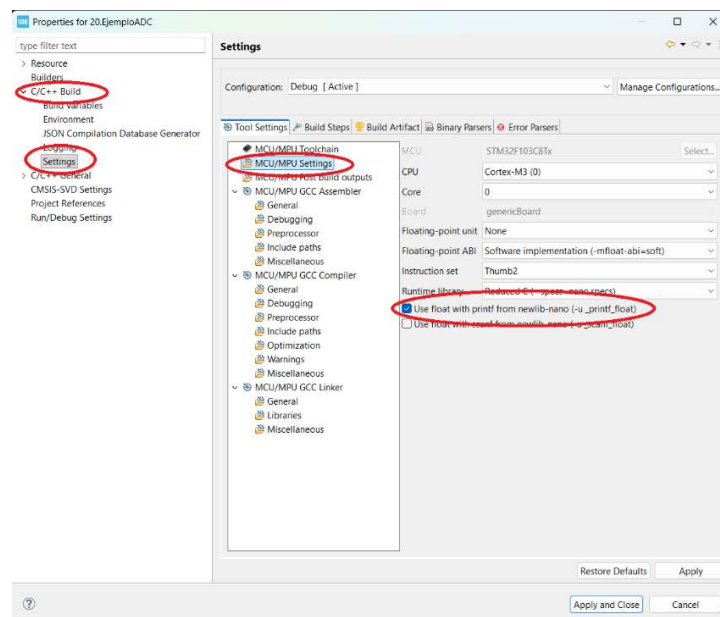
```
1 #include "main.h"
2 #include "adc.h"
3 #include "i2c.h"
4 #include "gpio.h"
5 #include "lcd.h"
6 #include <stdio.h>
7
8 uint32_t adcResult;
9 char strTemp[20];
10 float voltaje, temperatura;
11
12 void SystemClock_Config(void);
13
14 int main(void)
15 {
16     HAL_Init();
17     SystemClock_Config();
18     MX_GPIO_Init();
19     MX_ADC1_Init();
20     MX_I2C1_Init();
21     Lcd_Init();
22     Lcd_Gotoxy(1, 1);
23     Lcd_Print("Termometro");
24     HAL_ADC_Start(&hadc1);
25
26     while (1)
27     {
28         HAL_ADC_PollForConversion(&hadc1, 100);
29         adcResult = HAL_ADC_GetValue(&hadc1);
30         voltaje = (((float)adcResult) * 3.3) / 4095.0;
31         temperatura = voltaje * 100.0;
32         sprintf(strTemp, "Temp: %0.2f C", temperatura);
33         Lcd_Gotoxy(1, 2);
34         Lcd_Print(strTemp);
35         HAL_Delay(200);
36     }
37 }
```

Para poder compilar variables de punto flotante debemos habilitar una bandera de compilación de la siguiente forma. En el "Project

Explorer” seleccionamos el proyecto y damos click derecho y seleccionamos “Propiedades”

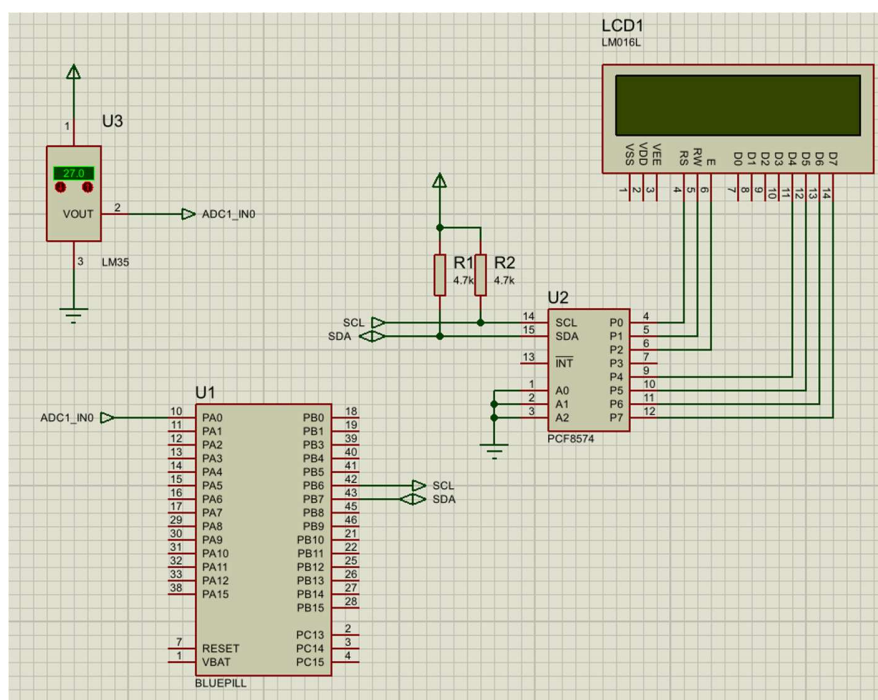


Después habilitamos la bandera de compilación para punto flotante de la siguiente manera.



5. Esquemático del circuito.

El esquemático de la práctica se muestra a continuación.



6. Observaciones.

Esta sección es para que el alumno anote sus observaciones.

7. Conclusiones.

Esta sección es para que el alumno anote sus conclusiones.

8. Importante.

La práctica deberá ser validada en el salón de clases antes de anexar el reporte al manual de prácticas. Una vez validada realizar el reporte de práctica como se anteriormente y anexar al manual de prácticas que se entregará a final del curso.