

MICROCONTROLADORES

Práctica No. 22. BUS I²C.

1. Objetivo

- Uso del módulo I²C del STM32F103.

2. Material y Equipo.

- Computador o laptop con STM32CubeIDE.
- Un sensor de presión digital BMP280.
- Una Pantalla OLED 0.96" 128x64 (SSD1306).

3. Marco de Referencia.

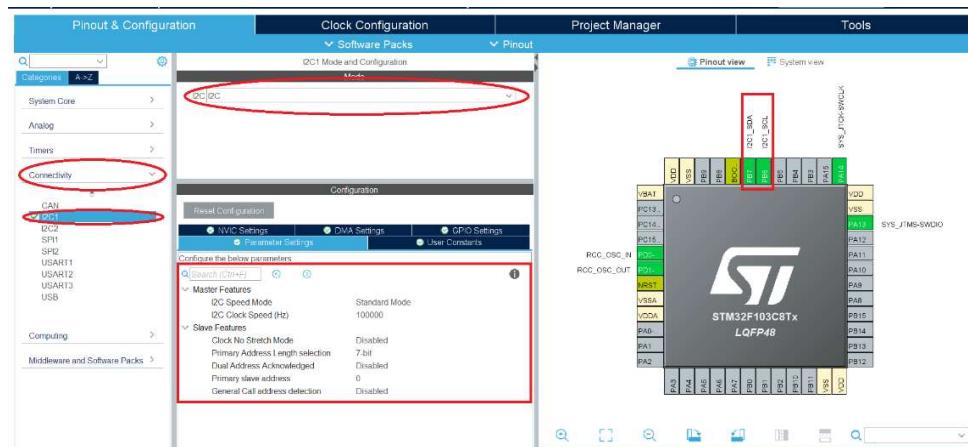
El bus I²C es un estándar muy utilizado para comunicar periféricos como sensores, memorias, relojes de tiempo real, ADC's, DAC's, pantallas tipo OLED, etc. Son muchos los dispositivos hechos con este estándar pudiendo ampliar las capacidades de los microcontroladores con su uso. El bus I²C está basado en comunicación maestro – esclavo, donde el maestro controla y genera la línea de reloj para poder comunicarse con el esclavo ya que el esclavo es incapaz de iniciar o enviar datos sin que el maestro se lo solicite. El bus I²C permite conectar hasta 128 dispositivos en el mismo bus esto se logra debido a la AND cableada que se hace al conectar tanto los dispositivos esclavos y el maestro en las líneas SDA (datos) y SCL (reloj) estas dos líneas tienen una salida de drenador abierto y dependiendo de la velocidad (hasta 1mbps) se elige el valor de las resistencias para una velocidad estándar de 100Khz se conectan dos resistencias (en SDA y SCL) de 4.7KΩ. Cuando el maestro realiza una

operación de lectura y escritura debe enviar esta petición a un dispositivo a la vez y cada esclavo tiene una dirección única (de 7 bits de ancho y un octavo bit para establecer si es lectura o escritura) con la cual el maestro puede solicitarle o enviarle información, la dirección se asigna a los dispositivos en el momento de su fabricación pero si queremos conectar un microcontrolador como dispositivo esclavo debemos asignarle una dirección diferente a la de los esclavos que tenemos en el bus y también diferente de cero ya que la dirección cero es una llamada a todos los dispositivos conectados al bus el maestro envía una dirección cero y todos los esclavos conectados al bus deben contestar. Para iniciar una comunicación el maestro debe enviar una secuencia de inicio con la cual el maestro toma el control del bus, para finalizar debe enviar una secuencia de paro y liberar el control del bus, si el maestro desea hacer otra transmisión sin dejar el control del bus se hace un inicio repetido. Después de enviar la secuencia de inicio el maestro enviar los 7 bits de la dirección del esclavo mas un octavo bit para lectura o escritura (1 para lectura y 0 para escritura) si se hace una solicitud de escritura después de la dirección se envía la dirección de memoria donde se desea almacenar el dato y por ultimo se envía el dato, si es una secuencia de lectura después de la dirección se enviar la dirección que se desea leer y el esclavo devolverá el dato almacenado en esa dirección por la línea SDA al maestro. Después de que el maestro envía un byte al esclavo este debe contestar con un bit llamado ACK (acknowledge) el esclavo responde con un cero avisando al

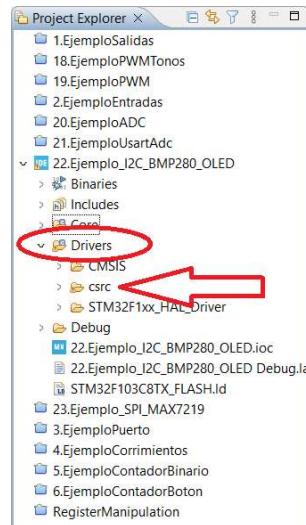
maestro que el dato se recibió correctamente si el esclavo no recibe el ACK entonces el maestro estaría recibiendo un noACK con esto el maestro sabe que el dato no se envió correctamente y el maestro debe reenviar el dato nuevamente, por otro lado cuando el esclavo envía un dato el maestro debe avisarle al esclavo que recibió ese dato con un ACK si lo recibió correctamente o un noACK si no fue así.

4. Desarrollo y Procedimiento.

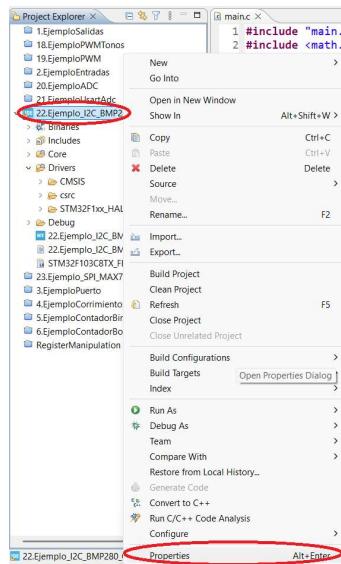
Se creará un proyecto en el STM32CubeIDE como se indicó anteriormente.



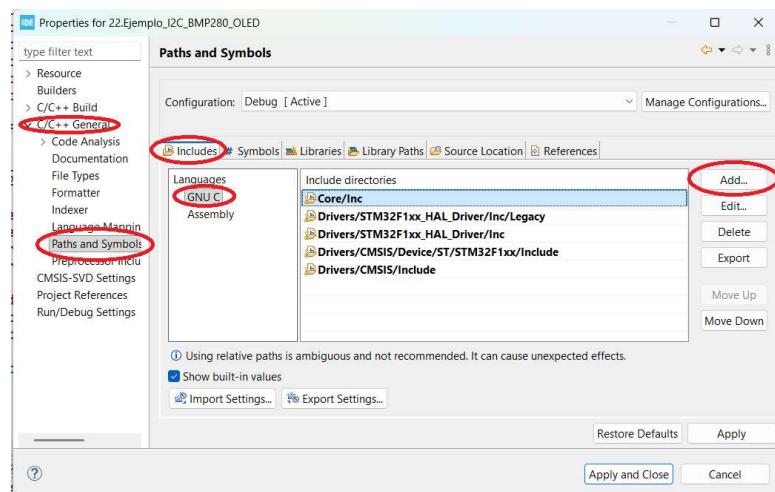
La librería para el manejo de la pantalla OLED (librería u8g2) debe guardarse en la siguiente ubicación en el proyecto.



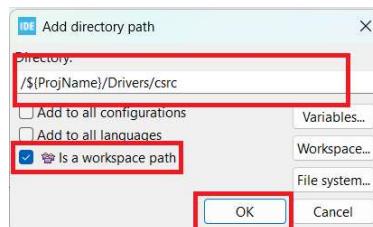
Después tenemos que indicarle al compilador la ruta de la carpeta de la librería dando click derecho sobre el proyecto y seleccionando la opción “Propiedades”.



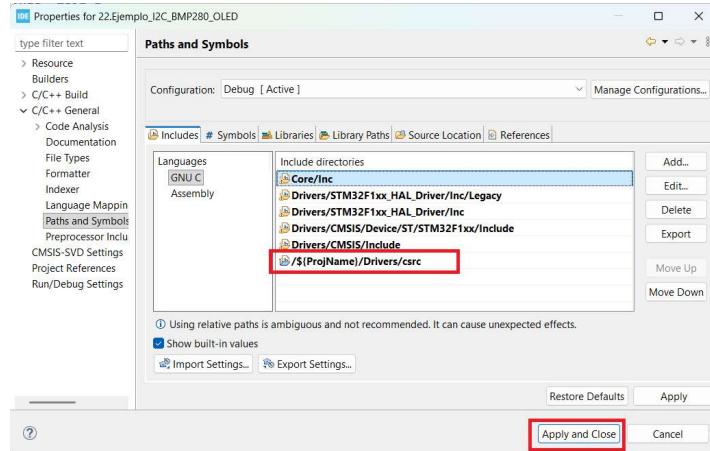
Después dentro de las propiedades del proyecto agregamos la ruta de la siguiente forma.



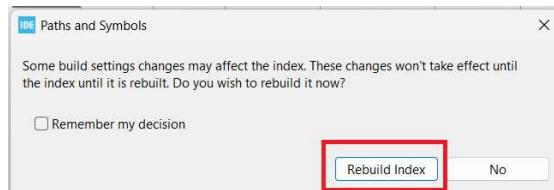
Damos click en el botón “Add...” y agregamos la ruta de la siguiente manera.



La ruta de la librería debería quedar agregada quedandon de la siguiente forma y ya solo queda dar click en el botón “Apply and Close”.



Por ultimo nos aparecerá una ventana donde nos pide reconstruir unos índices del proyecto solo damos click en “Rebuild Index”.



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

```
1 #include "main.h"
2 #include <math.h>
3 #include <string.h>
4 #include <stdio.h>
5 #include "BMP280_STM32.h"
6 #include "u8g2.h"
7
8 #define DELAY      5000
9
10 I2C_HandleTypeDef hi2c1;
11 u8g2_t myDisplay;
12
13 extern float temperature, pressure, altitude;
14 extern float init_height;
15 char string1[50];
16 char string2[50];
17
18 void SystemClock_Config(void);
19 static void MX_GPIO_Init(void);
20 static void MX_I2C1_Init(void);
21
22 uint8_t u8x8_gpio_and_delay(u8x8_t *u8x8, uint8_t msg, uint8_t arg_int, void *arg_ptr){
23     switch(msg){
24         case U8X8_MSG_DELAY_MILLI:
25             HAL_Delay(arg_int);
26             break;
27     }
28     return 1;
29 }
30
31 uint8_t u8x8_i2c(u8x8_t *u8x8, uint8_t msg, uint8_t arg_int, void *arg_ptr){
32     static uint8_t buffer[32];
33     static uint8_t buf_idx;
34     uint8_t *data;
35
36     switch(msg){
37         case U8X8_MSG_BYTE_SEND:
38             data = (uint8_t *)arg_ptr;
39             while(arg_int > 0){
40                 buffer[buf_idx++] = *data;
41                 data++;
42                 arg_int--;
43             }
44             break;
45         case U8X8_MSG_BYTE_START_TRANSFER:
46             buf_idx = 0;
47             break;
48         case U8X8_MSG_BYTE_END_TRANSFER:
49             HAL_I2C_Master_Transmit(&hi2c1, 0x78, buffer, buf_idx, 1000);
50             break;
51         default:
52             return 0;
53     }
54     return 1;
55 }
56
57 int main(void)
58 {
59     uint32_t tick = 0;
60
61     HAL_Init();
62     SystemClock_Config();
63     MX_GPIO_Init();
64     MX_I2C1_Init();
65     BMP280_init();
66
67     u8g2_Setup(ssd1306_i2c_128x64_noname_f(&myDisplay, U8G2_R0, u8x8_i2c, u8x8_gpio_and_delay);
68     u8g2_InitDisplay(&myDisplay);
69     u8g2_SetPowerSave(&myDisplay, 0);
70
71     u8g2_ClearDisplay(&myDisplay);
72     u8g2_SetFont(&myDisplay, u8g2_font_helvB10_tr);
73     u8g2_DrawStr(&myDisplay, 0, 15, "Temperatura:");
74     u8g2_DrawStr(&myDisplay, 0, 30, "° C");
75     u8g2_DrawStr(&myDisplay, 0, 45, "Presión:");
76     u8g2_DrawStr(&myDisplay, 0, 60, "° Pa");
77     u8g2_DrawStr(&myDisplay, 0, 75, "Altitud");
78     u8g2_DrawStr(&myDisplay, 0, 90, "° m");
79     u8g2_SendBuffer(&myDisplay);
```

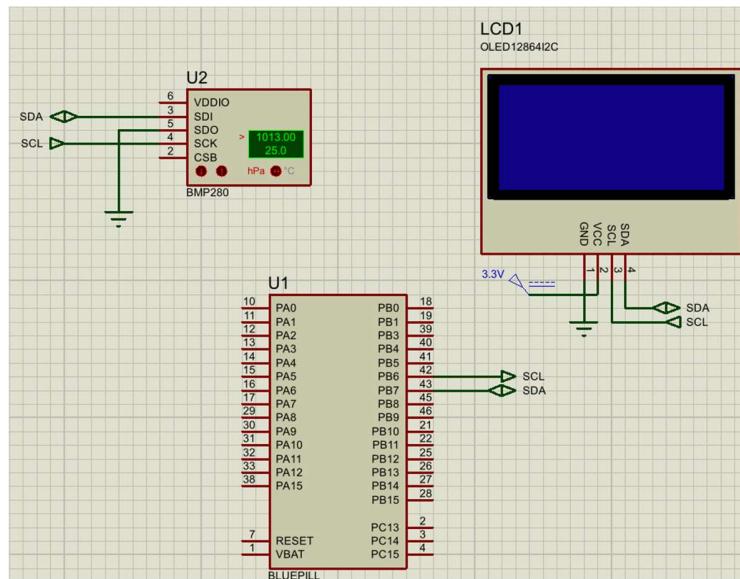
```

81     while (1)
82     {
83         memset(&string1, 0, strlen(string1));
84         memset(&string2, 0, strlen(string2));
85
86         u8g2_ClearDisplay(&myDisplay);
87         u8g2_DrawStr(&myDisplay, 0, 15, "Temperatura:");
88         u8g2_DrawStr(&myDisplay, 0, 45, "Presión:");
89         u8g2_SendBuffer(&myDisplay);
90
91         tick = HAL_GetTick();
92         while((HAL_GetTick() - tick) < DELAY){
93             BMP280_calc_values();
94             sprintf(string1, "%0.2f C", temperature);
95             sprintf(string2, "%0.2f Pa", pressure);
96             u8g2_DrawStr(&myDisplay, 0, 30, string1);
97             u8g2_DrawStr(&myDisplay, 0, 60, string2);
98             u8g2_SendBuffer(&myDisplay);
99             HAL_Delay(100);
100        }
101
102        memset(&string1, 0, strlen(string1));
103        u8g2_ClearDisplay(&myDisplay);
104        u8g2_DrawStr(&myDisplay, 0, 15, "Altitud:");
105        u8g2_SendBuffer(&myDisplay);
106
107        tick = HAL_GetTick();
108        while((HAL_GetTick() - tick) < DELAY){
109            BMP280_calc_values();
110            sprintf(string1, "%0.2f m", altitude);
111            u8g2_DrawStr(&myDisplay, 0, 30, string1);
112            u8g2_SendBuffer(&myDisplay);
113            HAL_Delay(100);
114        }
115    }
116 }

```

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 entregara a final del curso.