

FACULTAD DE CIENCIAS
EXACTAS E INGENIERÍA

SISTEMAS EMBEBIDOS 1

PROYECTO FINAL



EMILIO AMIR OROS SALAZAR
FRANCESC JULIO AQUIZE FLORES



UNIVERSIDAD
CATÓLICA
BOLIVIANA

SENSOR DE TEMPERATURA Y HUMEDAD CONTROLADO POR TARJETA DE DESARROLLO STM32F103C8T6



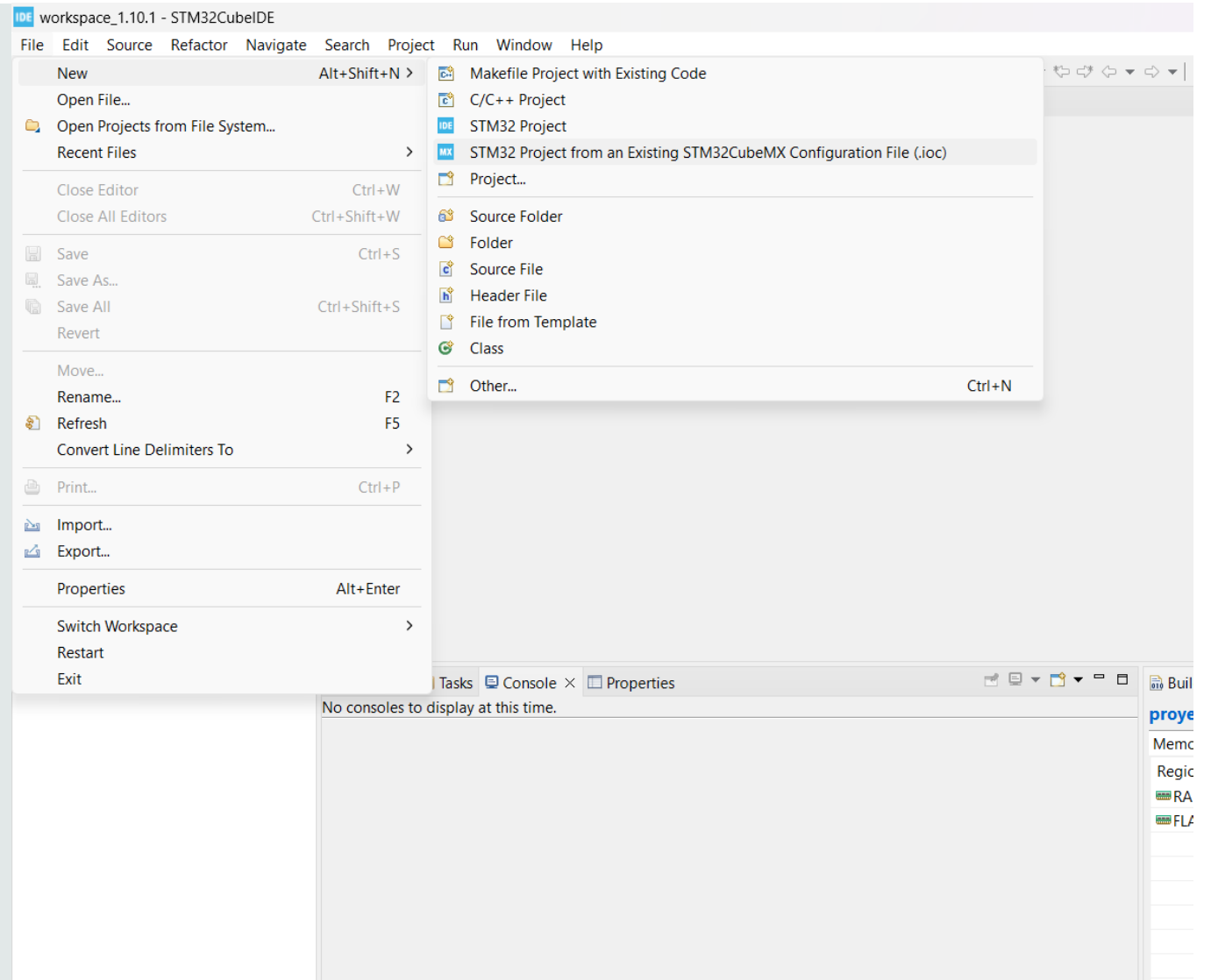
La presente diapositiva mostrará las configuraciones iniciales para la elaboración de un pequeño proyecto que consiste en la medición de temperatura y humedad a través de un sensor DTH11, estos datos pasan a través de una tarjeta programable stm32f103c8t6 y son mostrados mediante una pantalla oled I2C.

PASOS PARA REALIZAR EL PROYECTO



PASO 1.

Se debe abrir el IDE de programación STM32CubeIDE y seguidamente se debe crear un nuevo proyecto.



PASOS PARA REALIZAR EL PROYECTO



PASO 2.

Se selecciona la tarjeta que se utilizará para el proyecto, en este caso una tarjeta stm32f103c8t6

STM32 Project

Target Selection

⚠ STM32 target or STM32Cube example selection is required

MCU/MPU Selector | Board Selector | Example Selector | Cross Selector

MCU/MPU Filters

Commercial Part Number: STM32F103C8T6

PRODUCT INFO

- Segment
- Series
- Line
- Marketing Status
- Price
- Package
- Core
- Coprocessor

MEMORY

Flash = 64 (kBytes)

64

Featu... Block Diagr... Docs & Resourc... CAD Resour... Datasht...

STM32Cube

STM32U5 ultra-low-power MCU series with comprehensive STM32Cube ecosystem

MCUs/MPUs List: 2 items

Comme...	Part No	Reference	Market...	Unit Price ...	B...	Package	Flash	RAM	Fre...
☆ STM32F10...	STM32F...	STM32F1...	Active	2.7946		LQFP 48 7x7x1.4 ...	64 kB...	20 kB...	37 72 MHz
☆ STM32F10...	STM32F...	STM32F1...	Active	2.7946		LQFP 48 7x7x1.4 ...	64 kB...	20 kB...	37 72 MHz

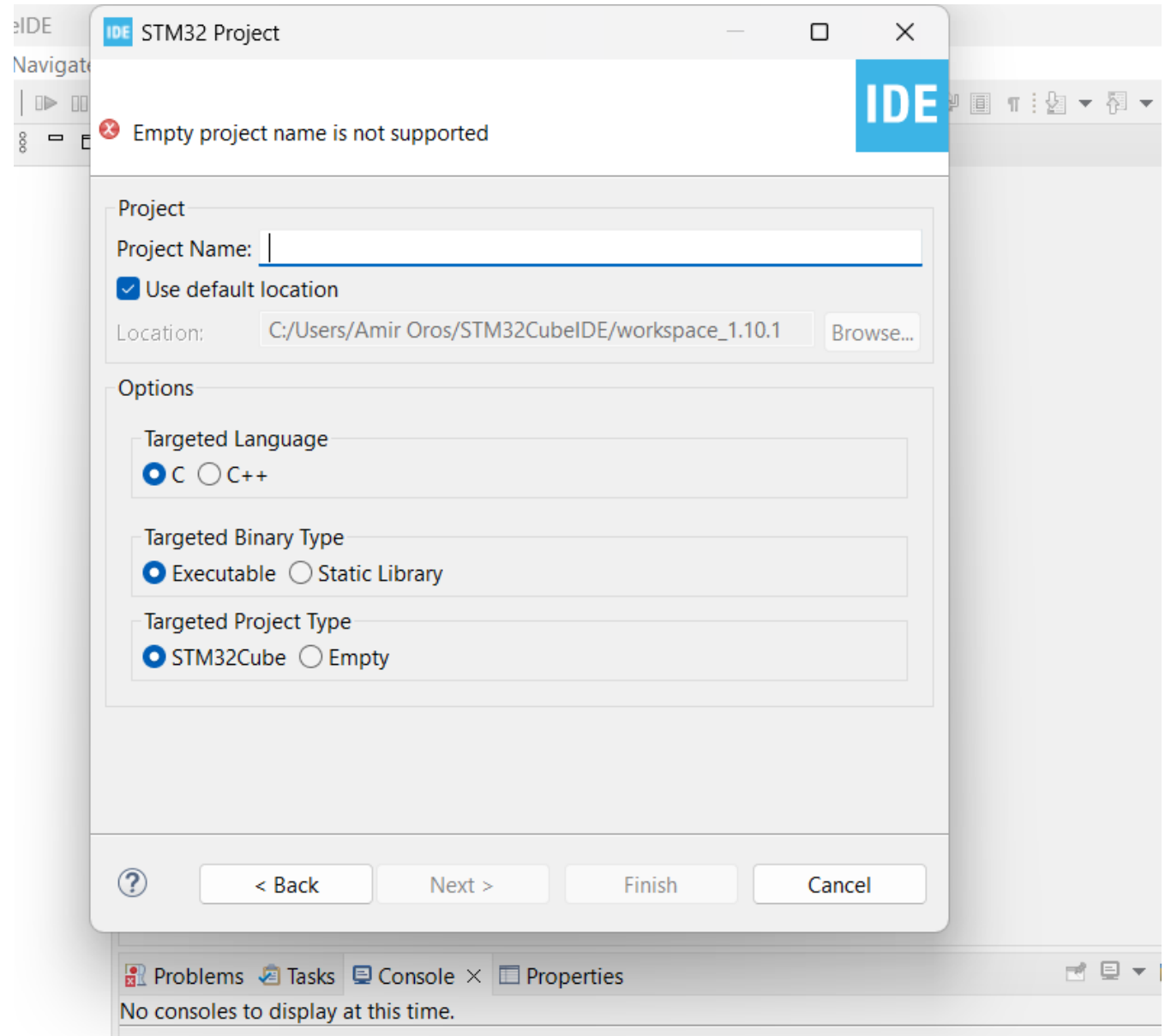
< Back Next > Finish Cancel

PASOS PARA REALIZAR EL PROYECTO



PASO 3.

Se asigna un nombre al proyecto.
(el nombre no debe contener
espacios)



Se guardan los cambios y el IDE generará el código base de forma automática.

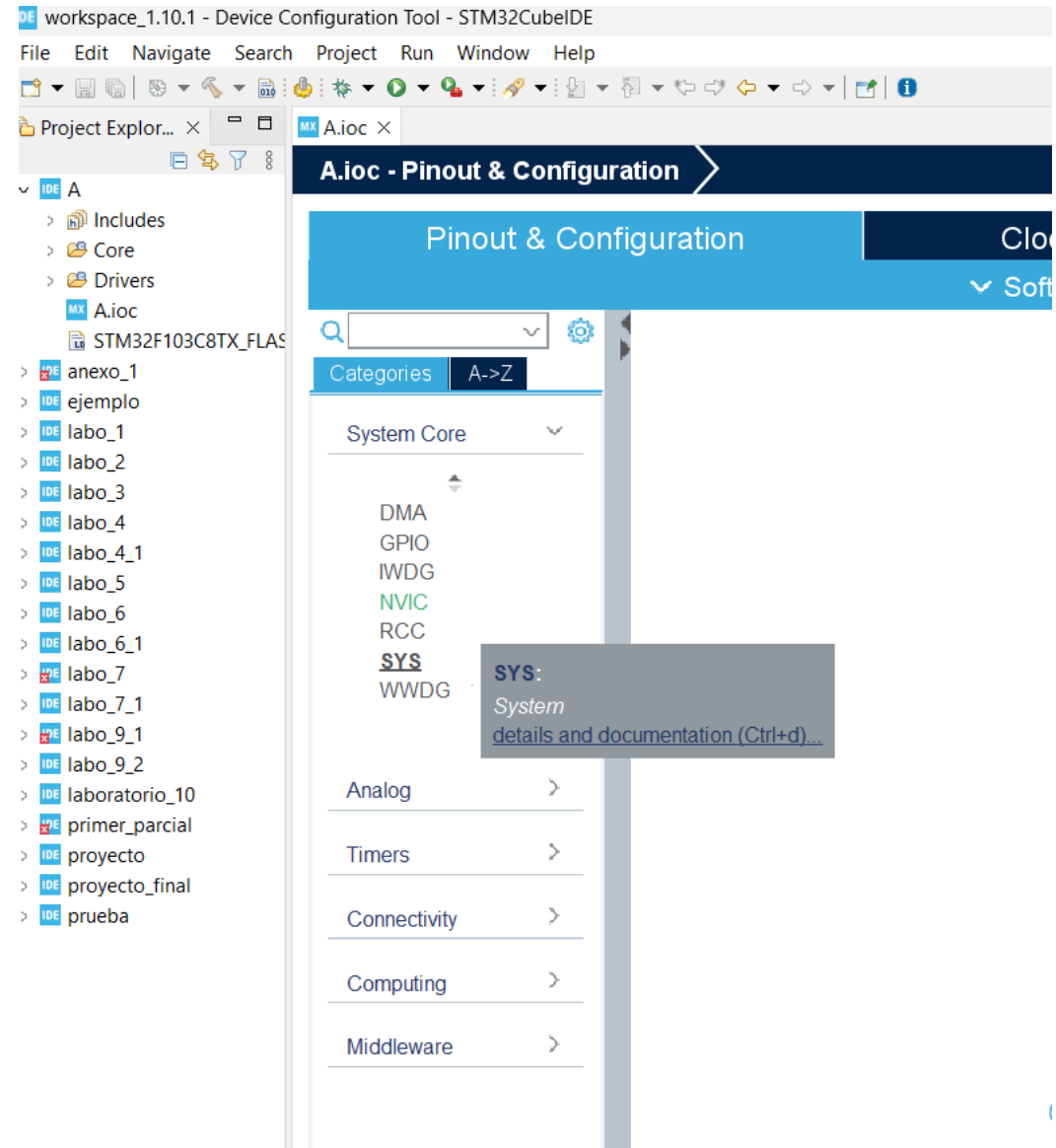


PASOS PARA REALIZAR EL PROYECTO



PASO 5.

Se debe ir hacia la pestaña SYS,
hacer un click.

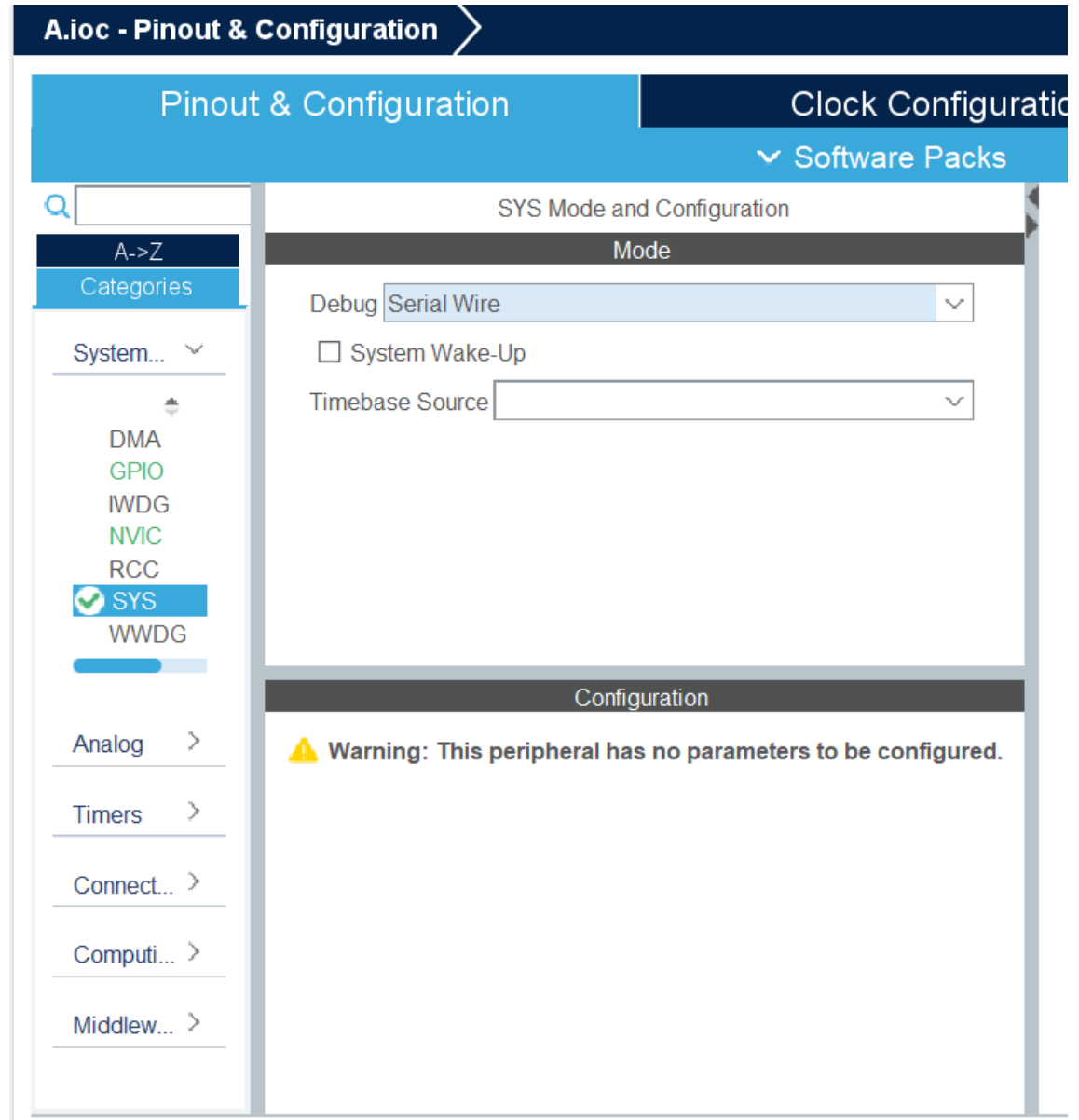


PASOS PARA REALIZAR EL PROYECTO



PASO 6.

Se debe configurar el debug como
«serial wire»



PASOS PARA REALIZAR EL PROYECTO



PASO 7.

Se debe dirigir a la pestaña RCC.

Habilitar el High Speed clock y
seleccionar cristal ceramic resonator

A.ioc - Pinout & Configuration

Pinout & Configuration | **Clock Configurati**

▼ Software Packs

Search

RCC Mode and Configuration

Mode

High Speed Clock (HSE) Crystal/Ceramic Resonator ▼

Low Speed Clock (LSE) Disable ▼

☐ Master Clock Output

Configuration

Reset Configuration

✓ NVIC Settings | ✓ GPIO Settings

✓ Parameter Settings | ✓ User Constants

Configure the below parameters :

Search (Ctrl+F)

▼ System Parameters

VDD voltage (V)	3.3 V
Prefetch Buffer	Enabled
Flash Latency(WS)	0 WS (1 CPU cycle)

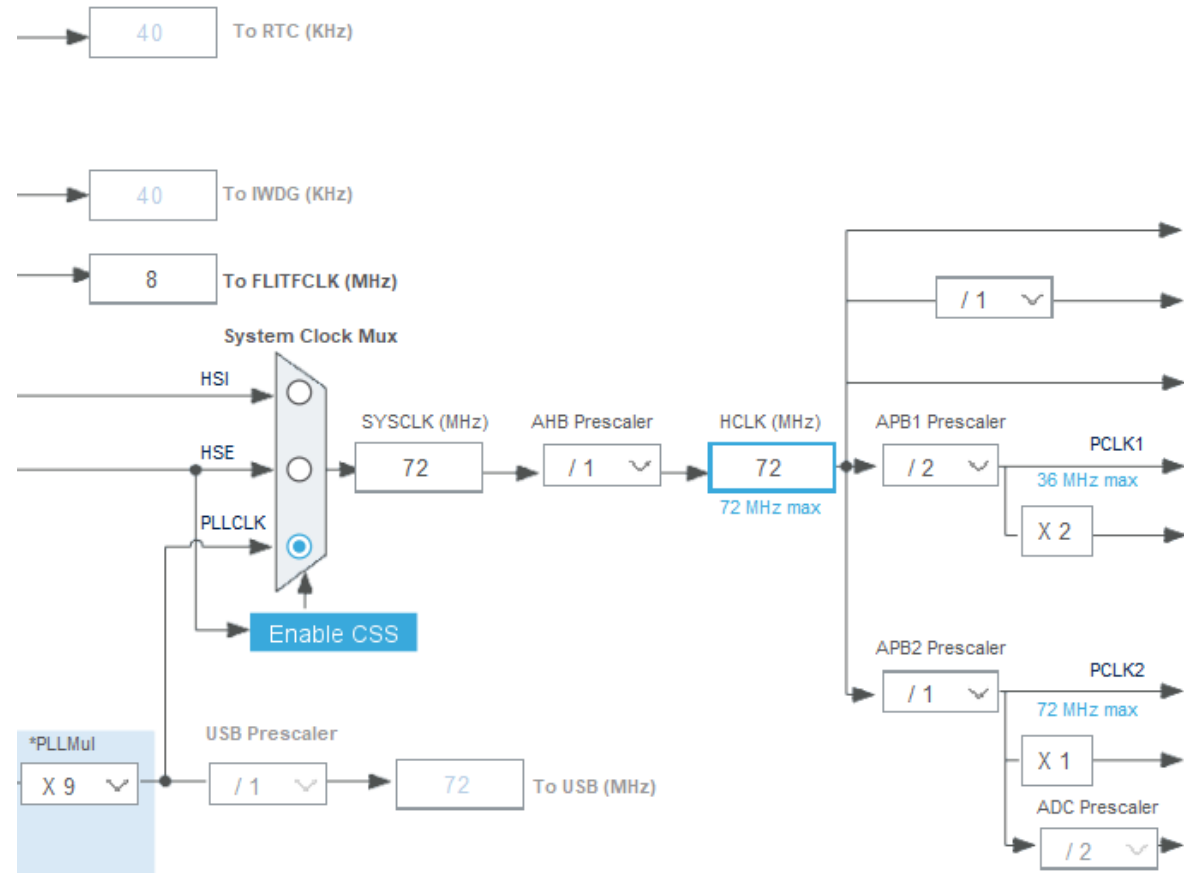
PASOS PARA REALIZAR EL PROYECTO



PASO 8.

Se debe dirigir a la pestaña clock configuration, consecuentemente deberá cambiar el valor de SYSCLK de 8 Mhz (por defecto) por una frecuencia de 72 Mhz.

Se presioná enter/intro y el software se tomará un tiempo para calcular los demás parámetros que se ajusten a nuestra frecuencia de forma automática.

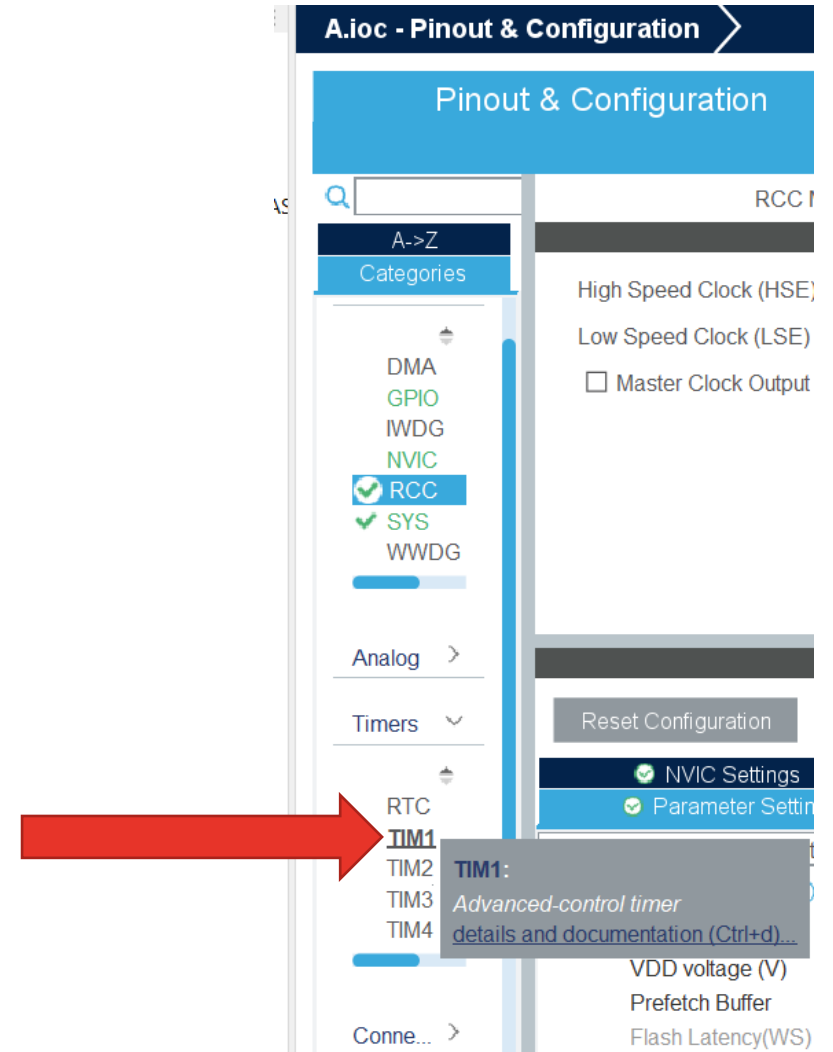


PASOS PARA REALIZAR EL PROYECTO



PASO 9.

Se debe presionar la pestaña Pinout
& configuration y dirigirse a la
pestaña TIM1



PASOS PARA REALIZAR EL PROYECTO



PASO 10.

Se debe configurar clock source
como internal clock y en la
frecuencia del pre-escalador se debe
colocar 71

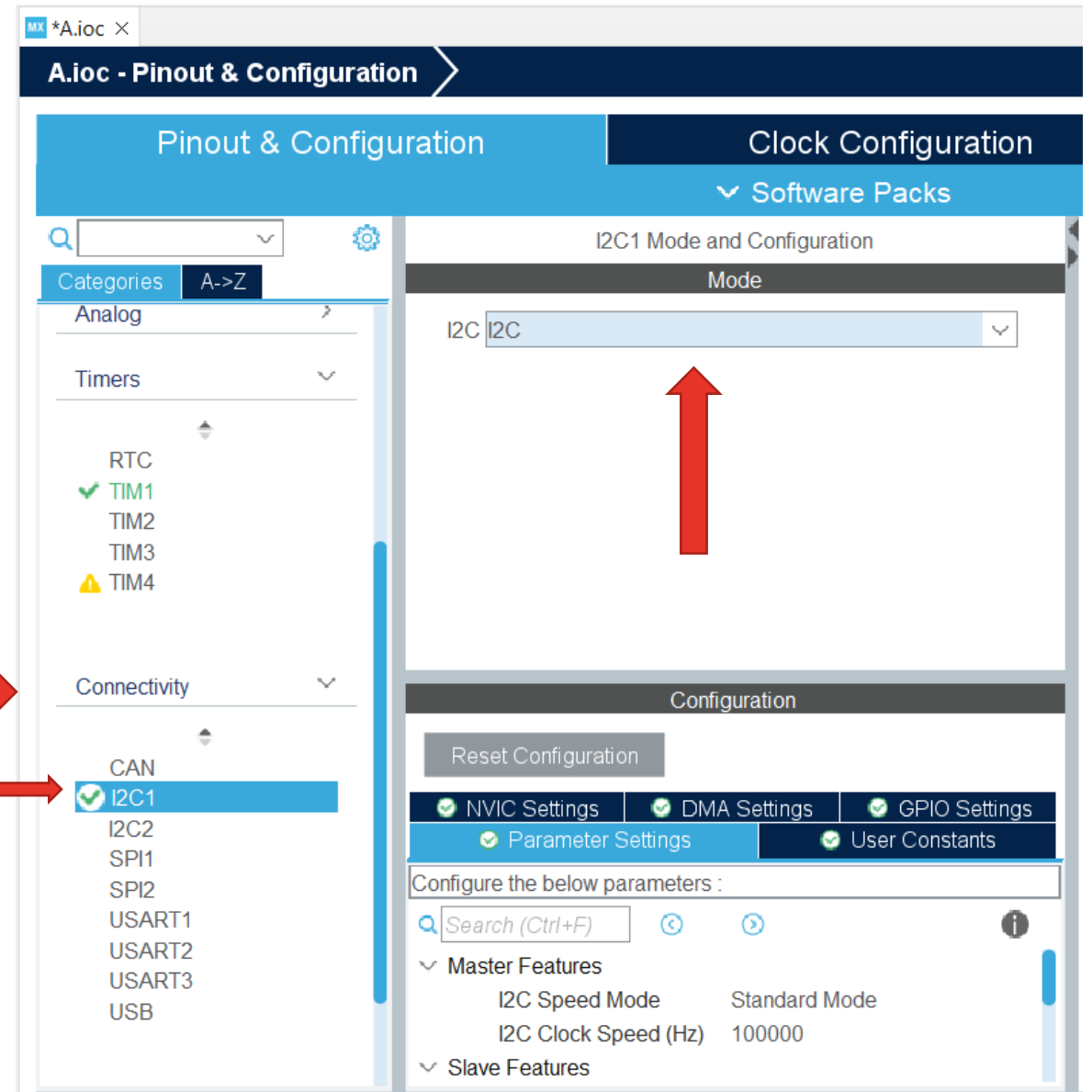
The screenshot displays the STM32CubeMX Pinout & Configuration window. The left sidebar shows a list of components: DMA, GPIO, IWDG, NVIC, RCC (checked), SYS (checked), WWDG, Analog, Timers, RTC, TIM1 (checked), TIM2, TIM3, and TIM4. The main panel is titled 'TIM1 Mode and Configuration'. It has two sections: 'Mode' and 'Configuration'. The 'Mode' section includes dropdown menus for Slave Mode (Disable), Trigger Source (Disable), Clock Source (Internal Clock), Channel1 (Disable), Channel2 (Disable), Channel3 (Disable), Channel4 (Disable), and Combined Channels (Disable). The 'Configuration' section includes a 'Reset Configuration' button and four checked settings: NVIC Settings, DMA Settings, Parameter Settings, and User Constants. Below these, a search bar is present. The 'Counter Settings' section is expanded, showing 'Prescaler (PSC - 16 bits v. 71)' set to 71, 'Counter Mode' set to Up, and 'Counter Period (AutoRelo.. 65535)'.

PASOS PARA REALIZAR EL PROYECTO



PASO 11.

Se debe dirigir a la pestaña
Conectivity y seleccionar I2C



PASOS PARA REALIZAR EL PROYECTO



PASO 12.

Deberá quedar una configuración
idéntica a esta.

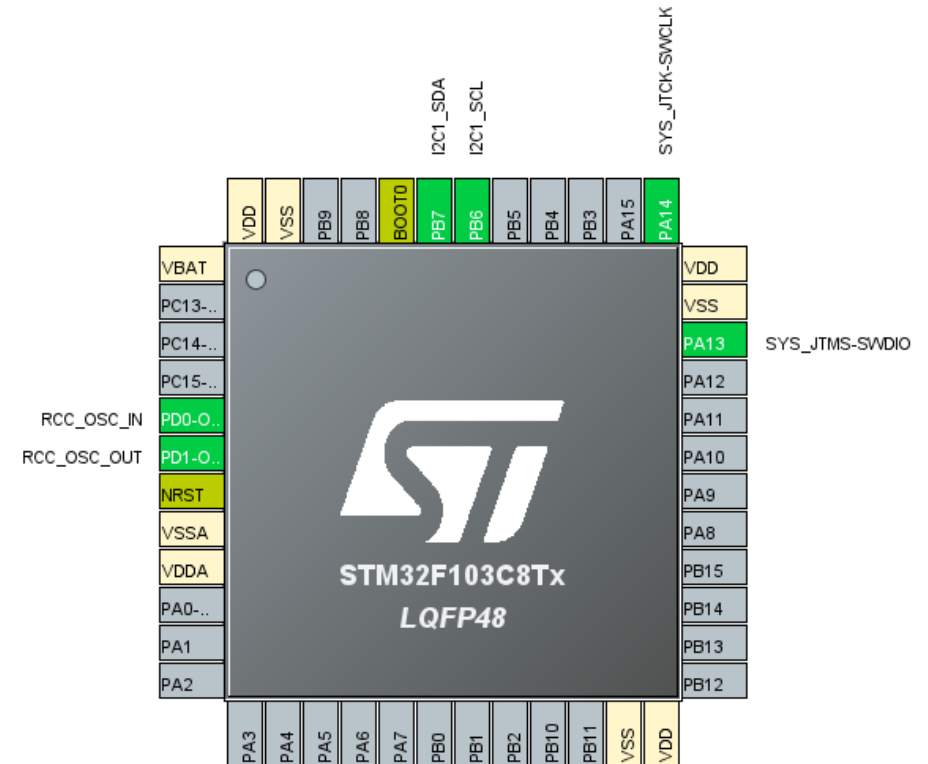
Pinout



Pinout view



System view

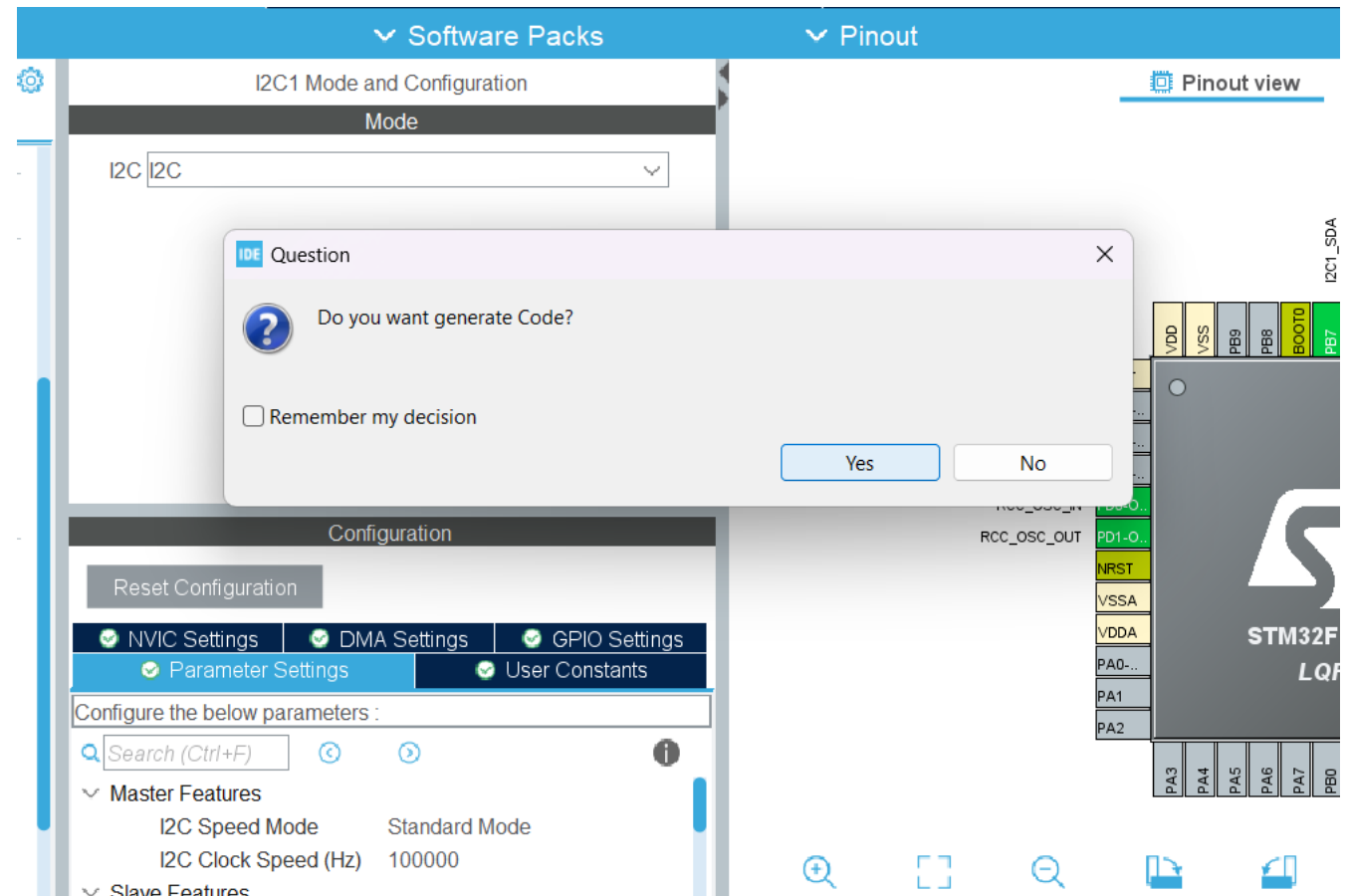


PASOS PARA REALIZAR EL PROYECTO



PASO 13.

Guardar el proyecto para generar el código base.

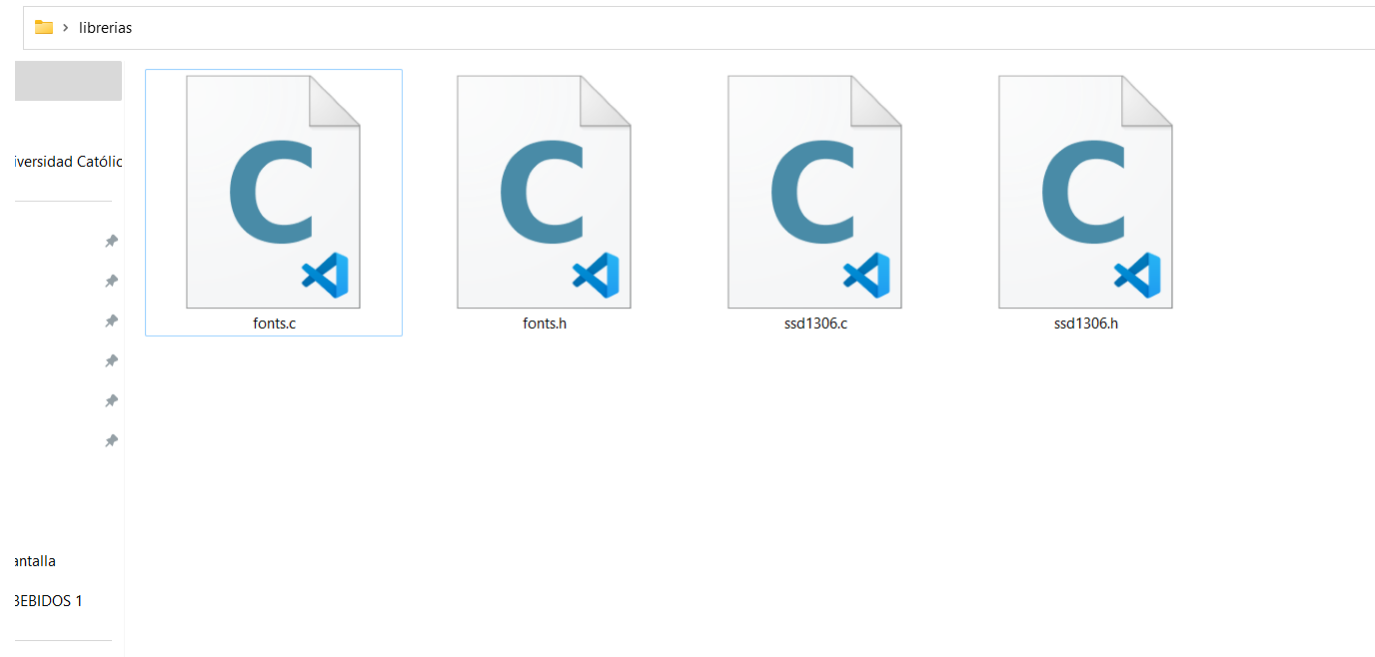


PASOS PARA REALIZAR EL PROYECTO



PASO 14.

Una vez generado el código base,
debemos adjuntar las librerías de
nuestros componentes a las carpetas
Source e INC

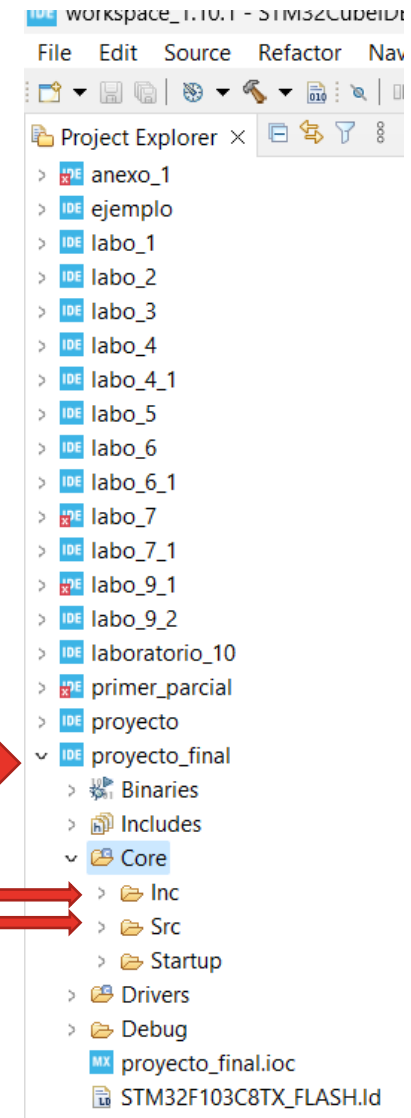


PASOS PARA REALIZAR EL PROYECTO



PASO 14.

Una vez generado el código base,
debemos adjuntar las librerías de
nuestros componentes a las carpetas
Source e INC

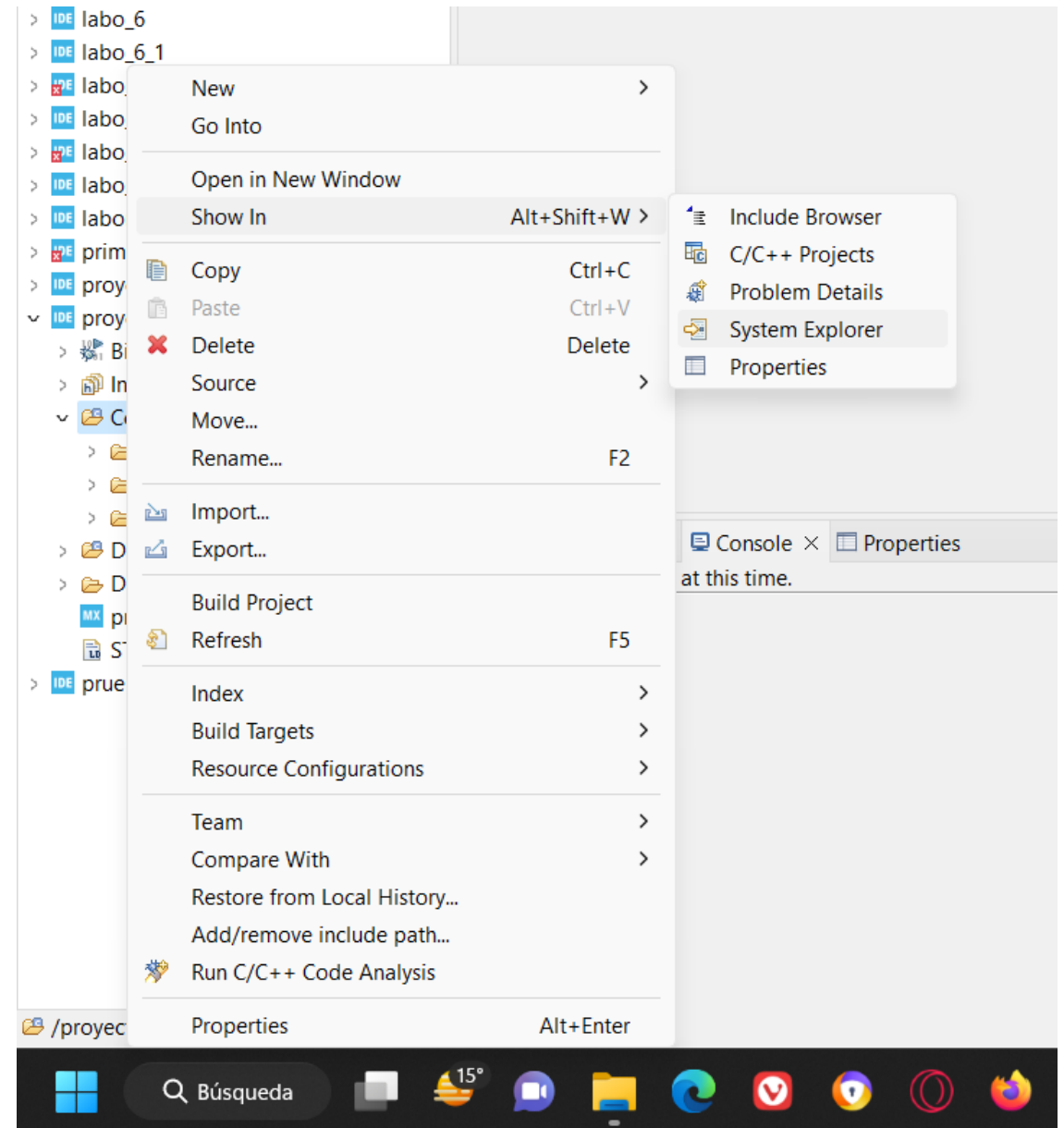


PASOS PARA REALIZAR EL PROYECTO



PASO 15.

Click derecho, luego show in y
finalmente system explorer.

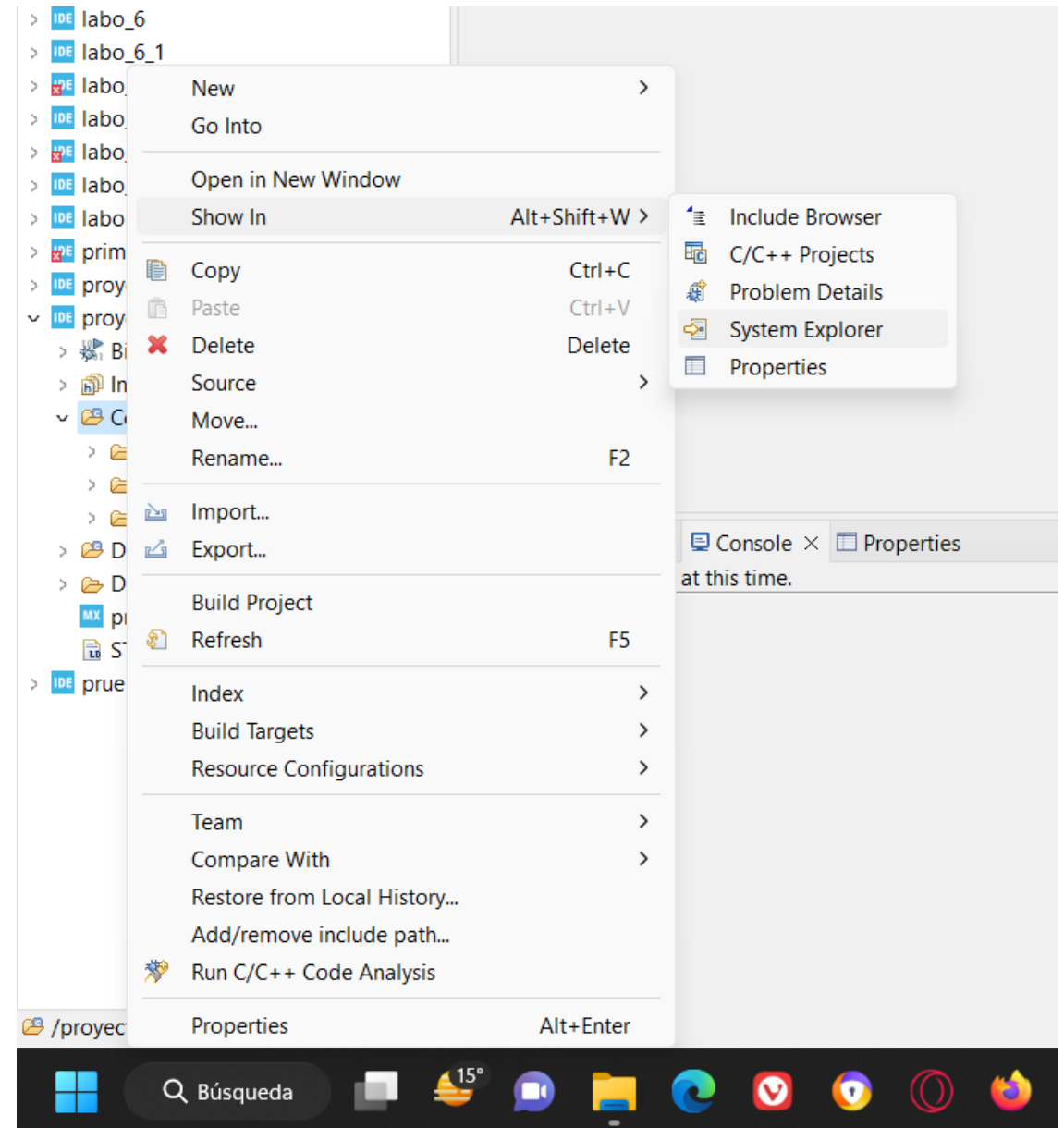


PASOS PARA REALIZAR EL PROYECTO



PASO 15.

Click derecho, luego show in y
finalmente system explorer.



PASOS PARA REALIZAR EL PROYECTO



PASO 16.

Colocar los archivos .h en INC.

Colocar los archivos .c en SRC

PROYECTO FINAL SISTEMAS EMBEBIDOS 1 > proyecto_final > Core > Inc					
Herramientas	Nombre	Fecha de modificación	Tipo	Tamaño	
	fonts.h	28/11/2022 23:43	Archivo de origen ...	3 KB	
	main.h	28/11/2022 23:21	Archivo de origen ...	2 KB	
	ssd1306.h	28/11/2022 23:43	Archivo de origen ...	11 KB	
	stm32f1xx_hal_conf.h	28/11/2022 23:27	Archivo de origen ...	16 KB	
	stm32f1xx_it.h	28/11/2022 23:21	Archivo de origen ...	2 KB	

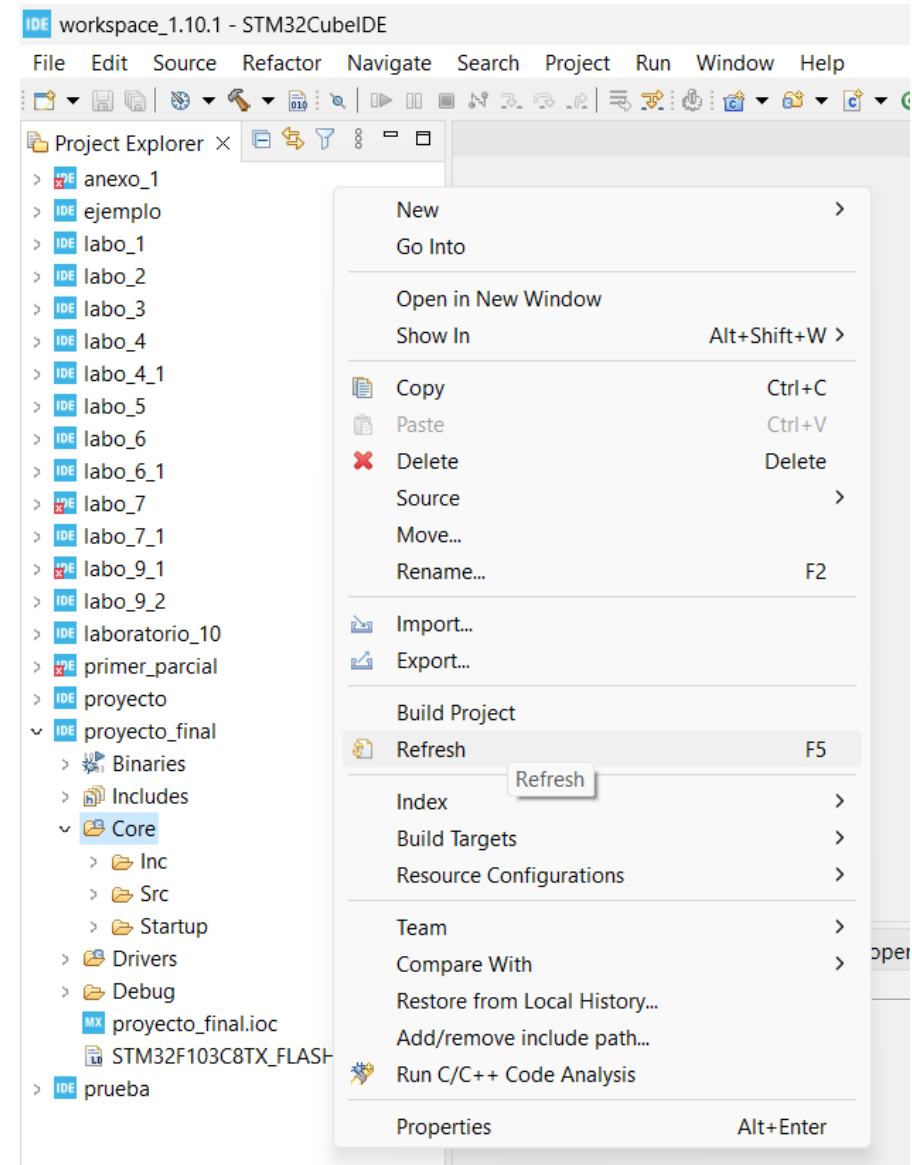
PROYECTO FINAL SISTEMAS EMBEBIDOS 1 > proyecto_final > Core > Src					
Herramientas	Nombre	Fecha de modificación	Tipo	Tamaño	
	fonts.c	28/11/2022 23:43	Archivo de origen C	43 KB	
	main.c	28/11/2022 23:39	Archivo de origen C	12 KB	
	ssd1306.c	28/11/2022 23:44	Archivo de origen C	17 KB	
	stm32f1xx_hal_msp.c	28/11/2022 23:27	Archivo de origen C	5 KB	
	stm32f1xx_it.c	28/11/2022 23:21	Archivo de origen C	6 KB	
	syscalls.c	28/11/2022 23:21	Archivo de origen C	3 KB	
	systemem.c	28/11/2022 23:21	Archivo de origen C	3 KB	
	system_stm32f1xx.c	28/11/2022 16:15	Archivo de origen C	15 KB	

PASOS PARA REALIZAR EL PROYECTO

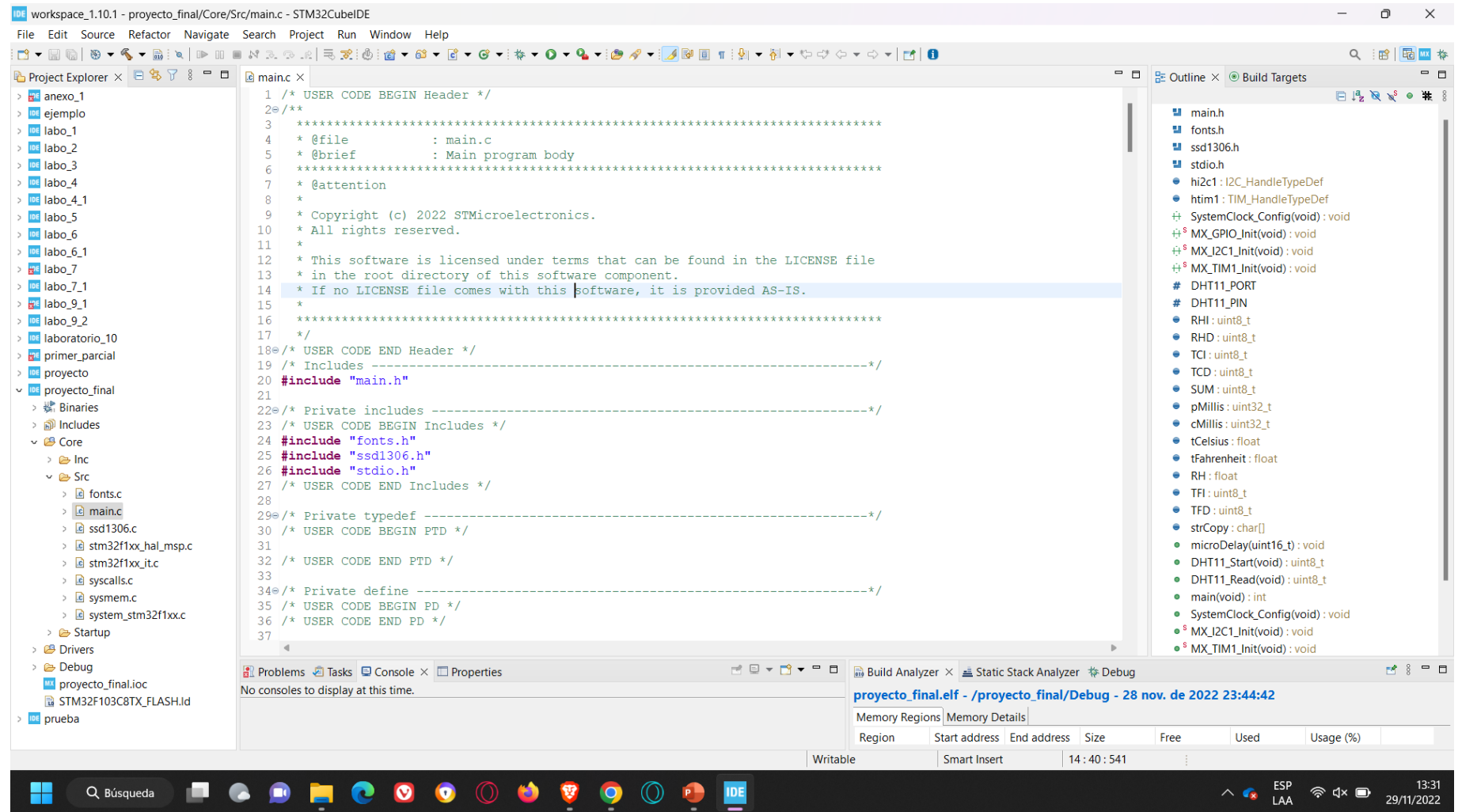


PASO 17.

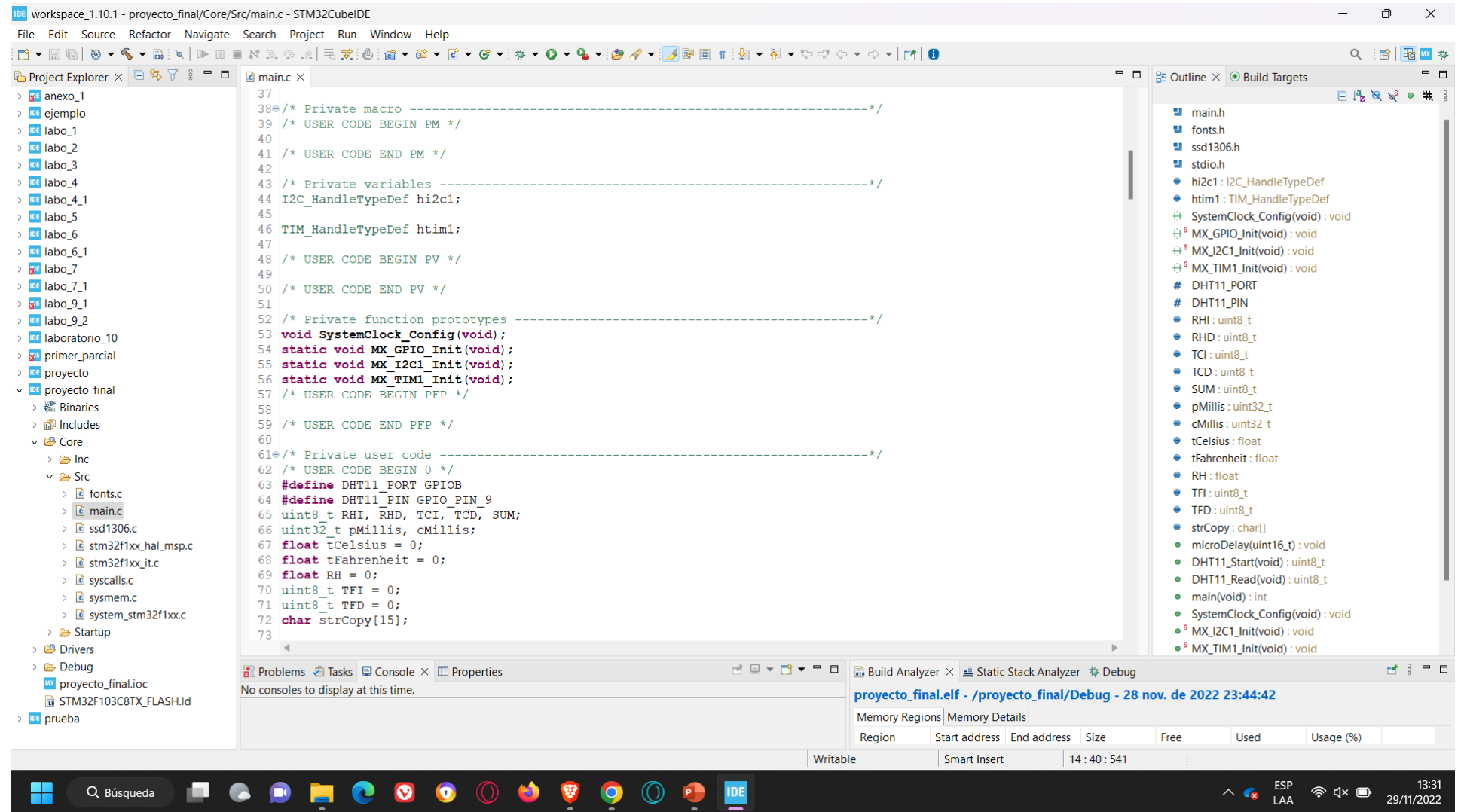
Volver al IDE y hacer click derecho en cualquier parte vacía del lateral izquierdo y colocar refresh.



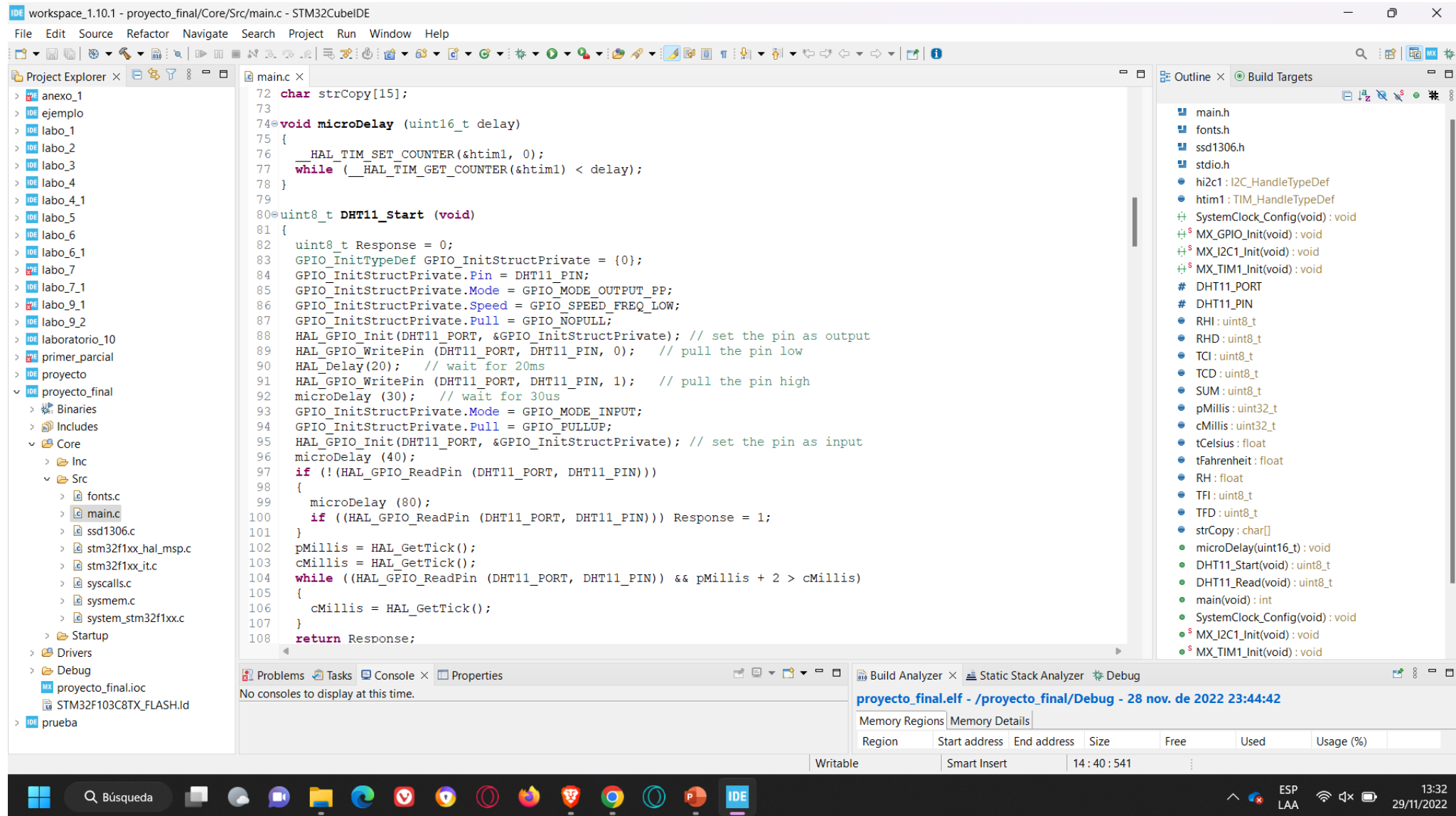
Vista del código
utilizado con las
modificaciones
sobre el template
de base (código de
base)



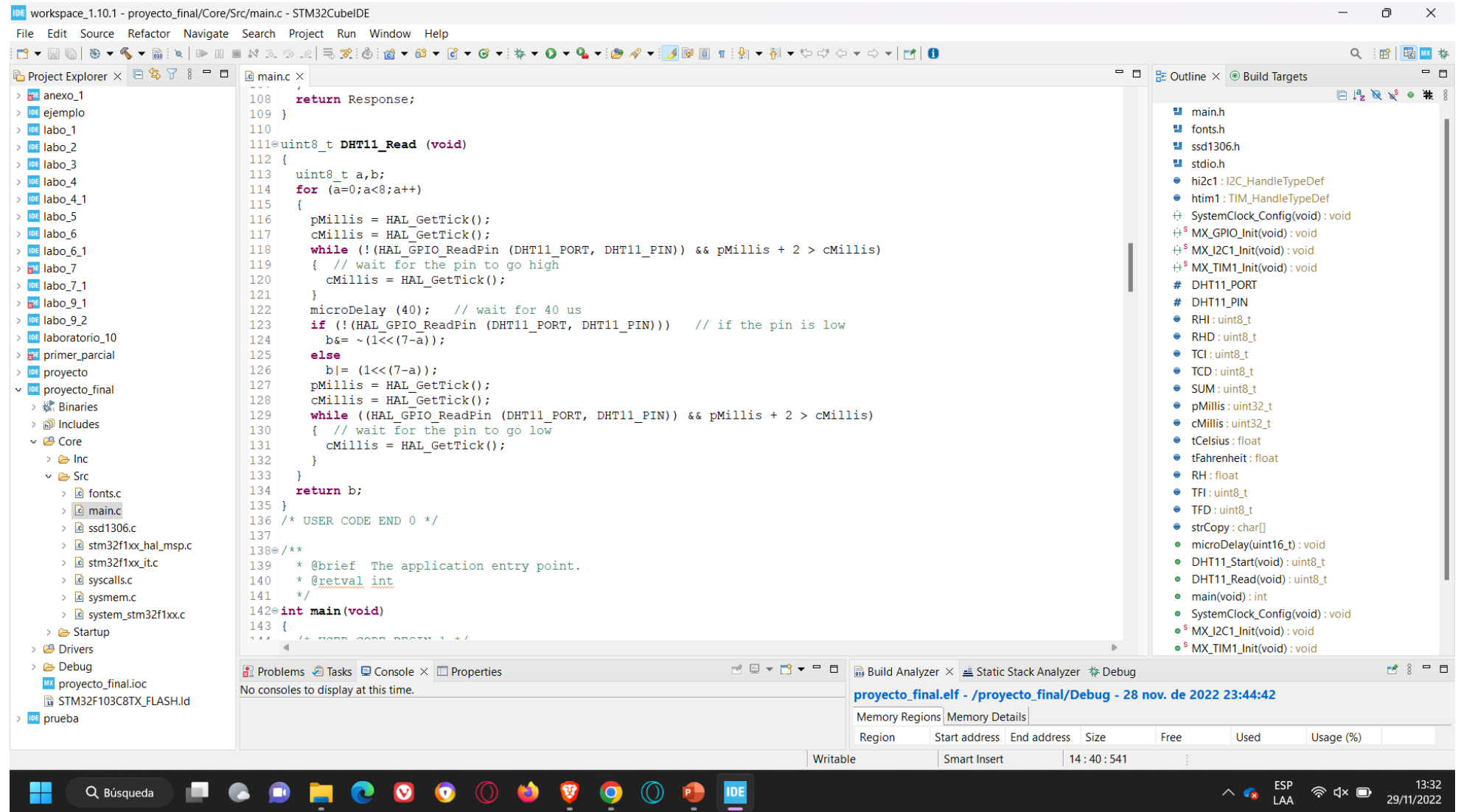
Vista del código
utilizado con las
modificaciones
sobre el template
de base (código de
base)



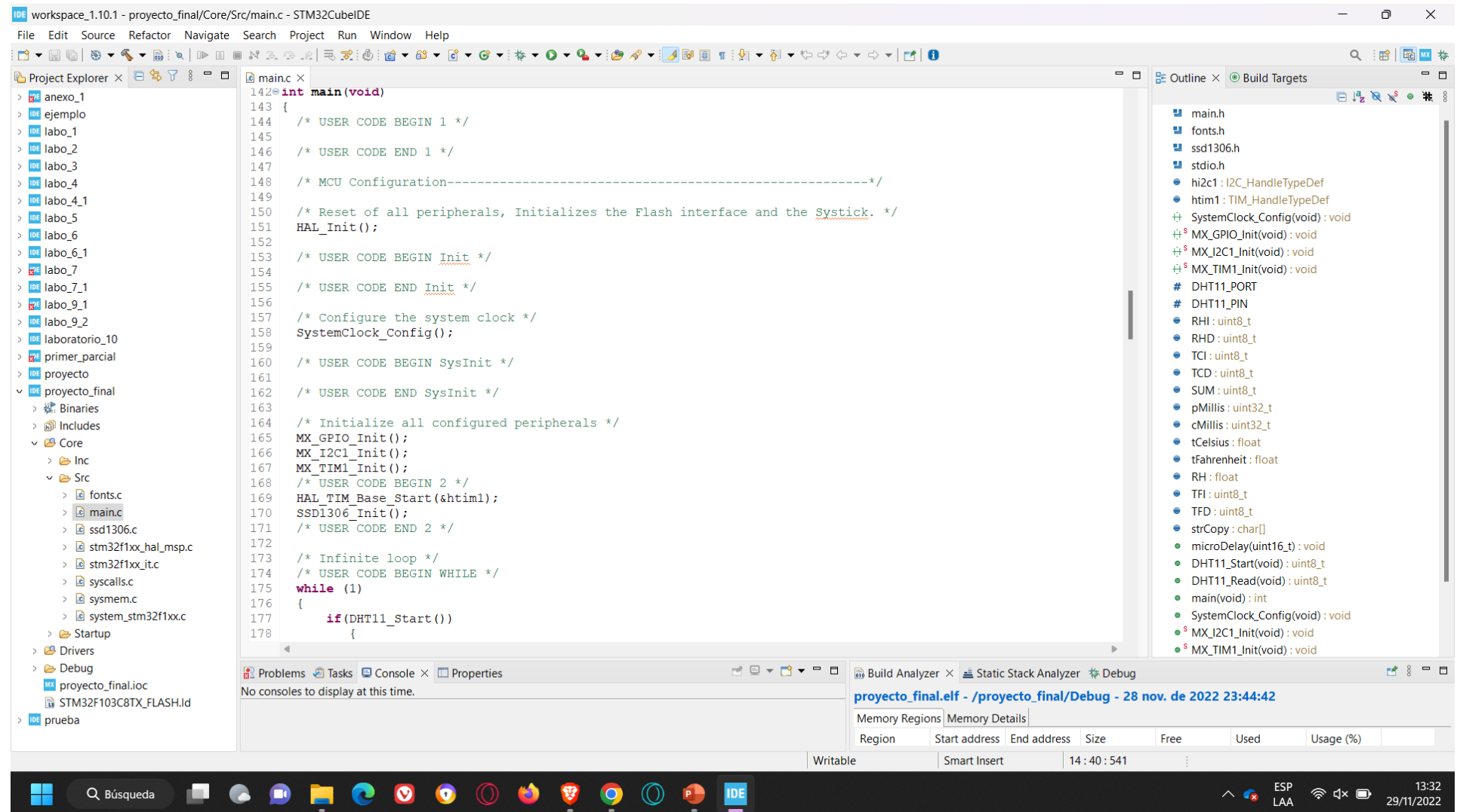
Vista del código
utilizado con las
modificaciones
sobre el template
de base (código de
base)



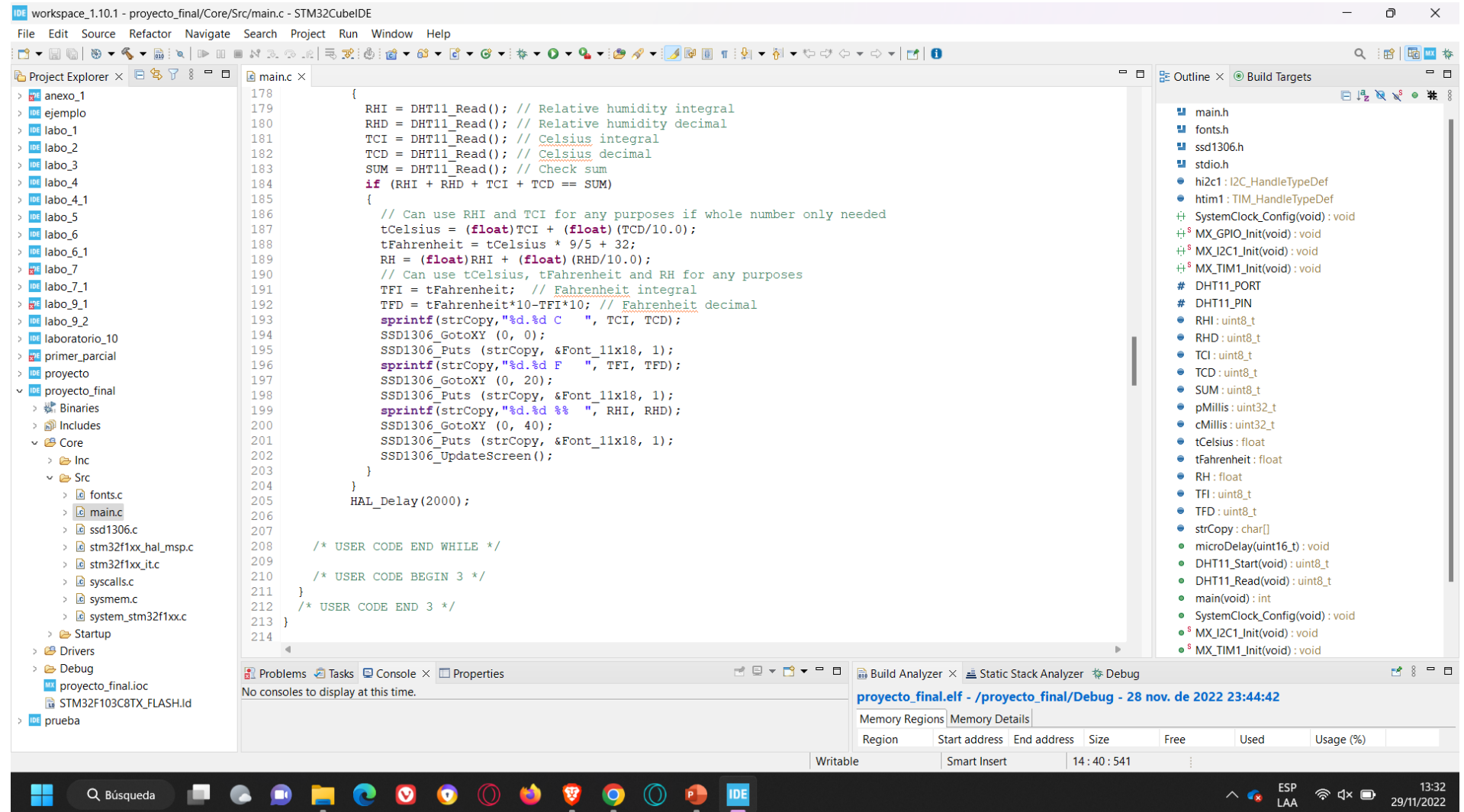
Vista del código
utilizado con las
modificaciones
sobre el template
de base (código de
base)



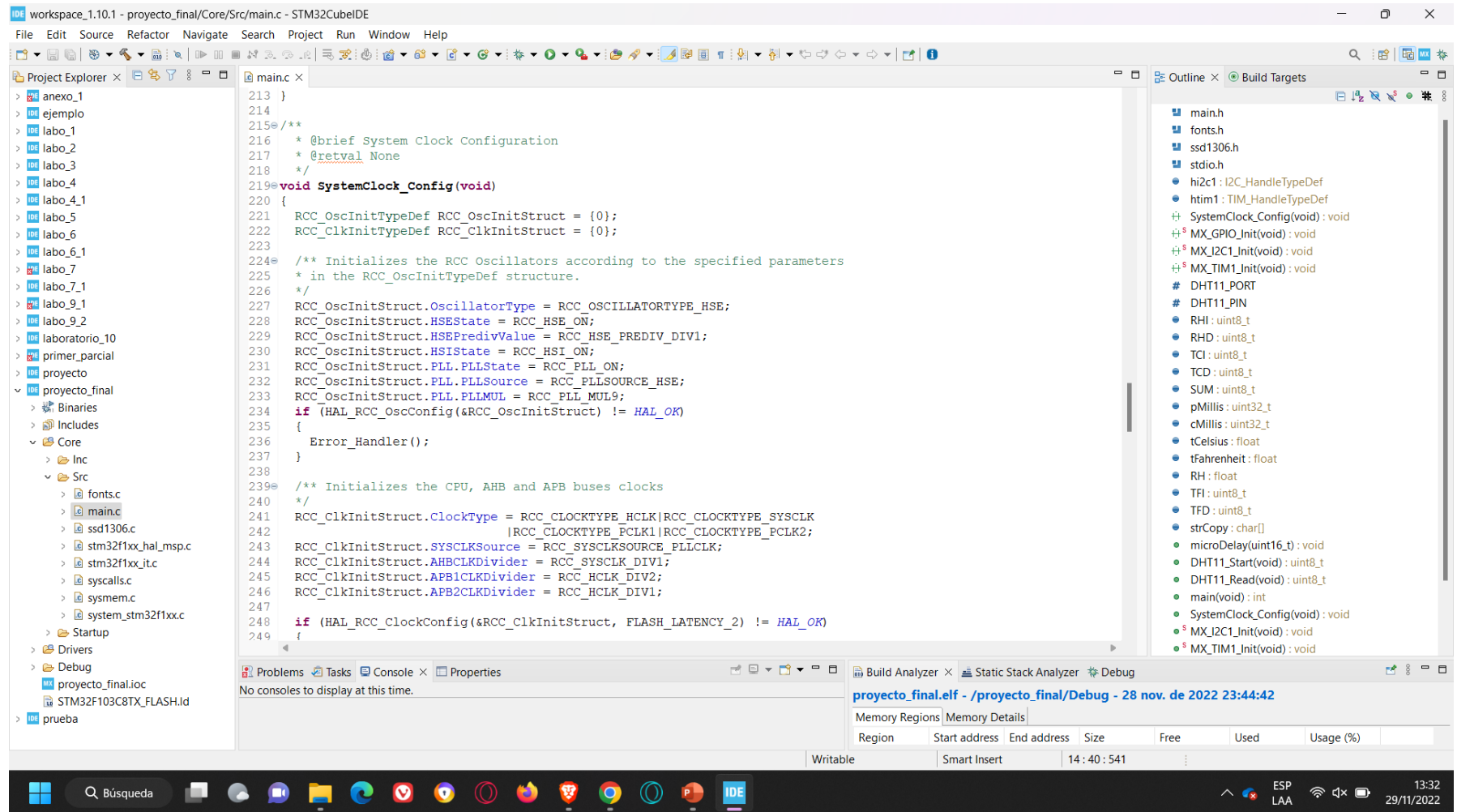
Vista del código
utilizado con las
modificaciones
sobre el template
de base (código de
base)



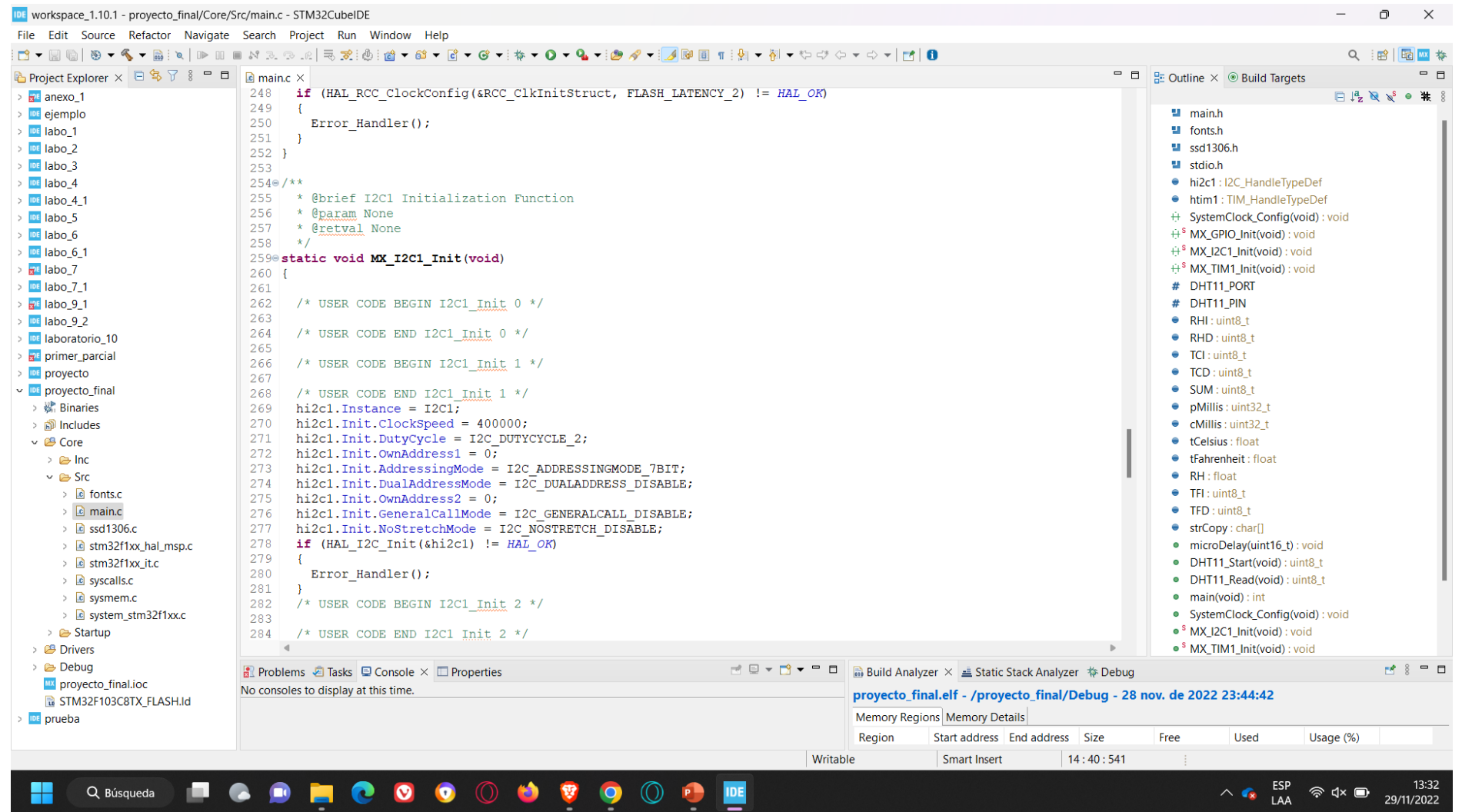
Vista del código
utilizado con las
modificaciones
sobre el template
de base (código de
base)



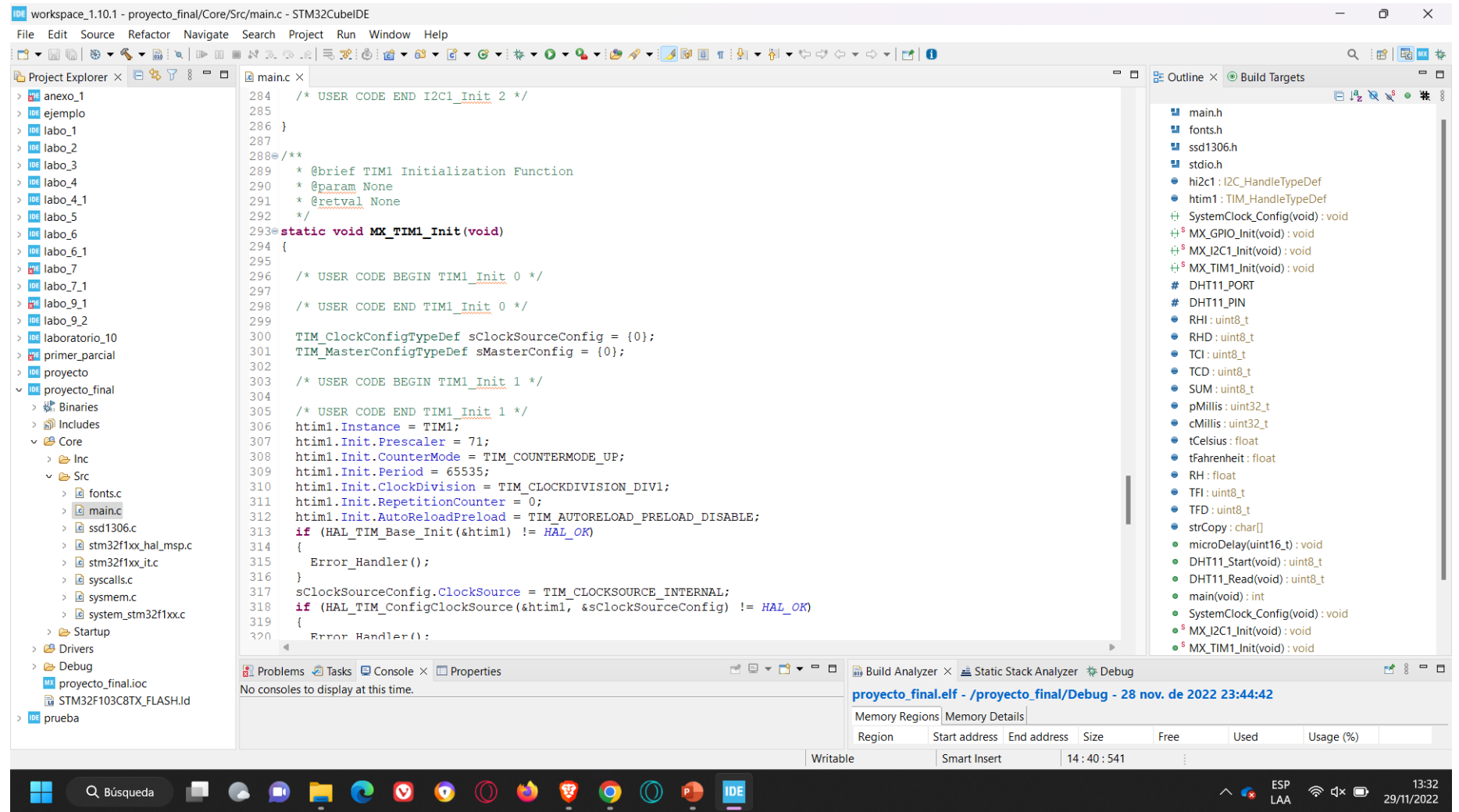
Vista del código
utilizado con las
modificaciones
sobre el template
de base (código de
base)



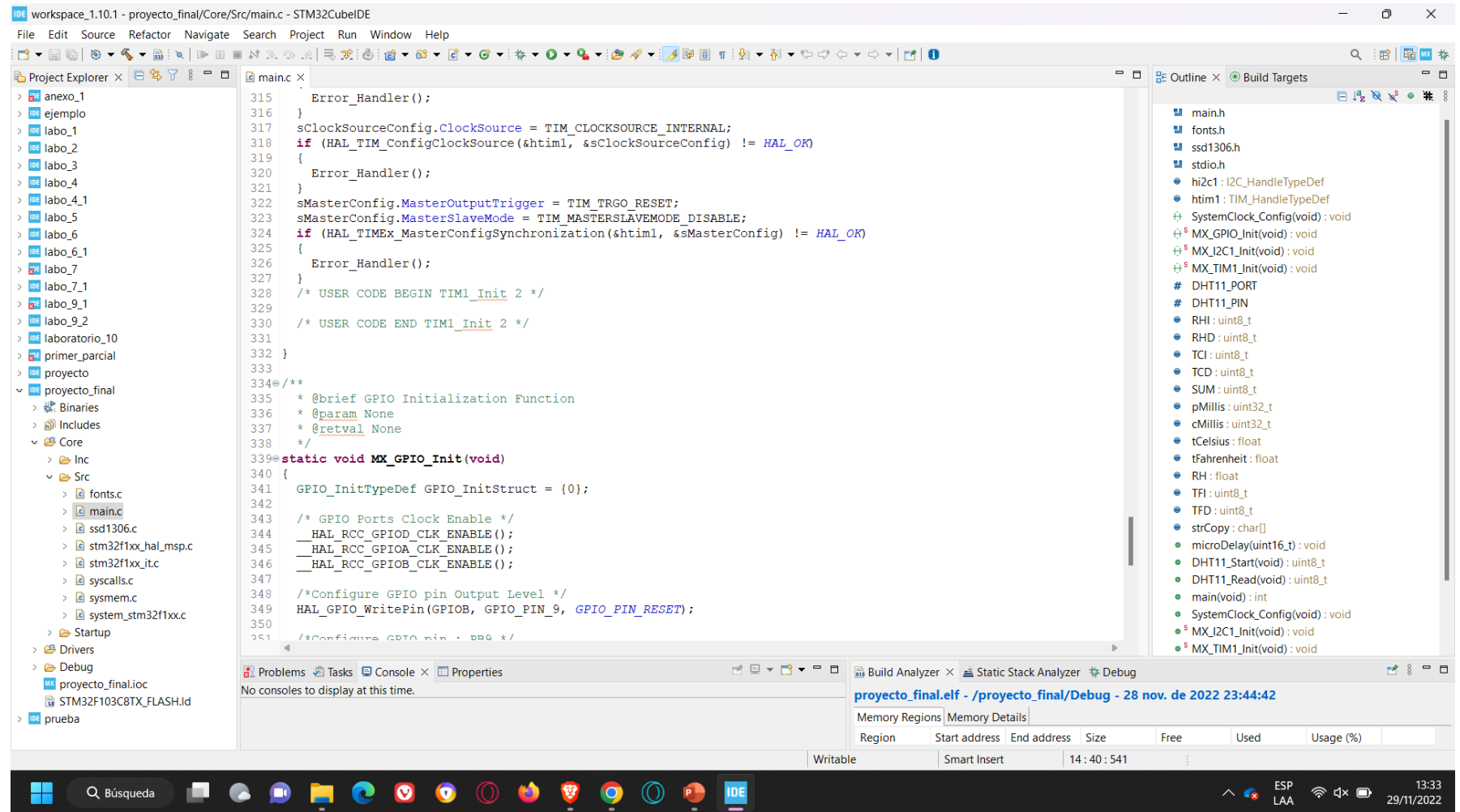
Vista del código
utilizado con las
modificaciones
sobre el template
de base (código de
base)



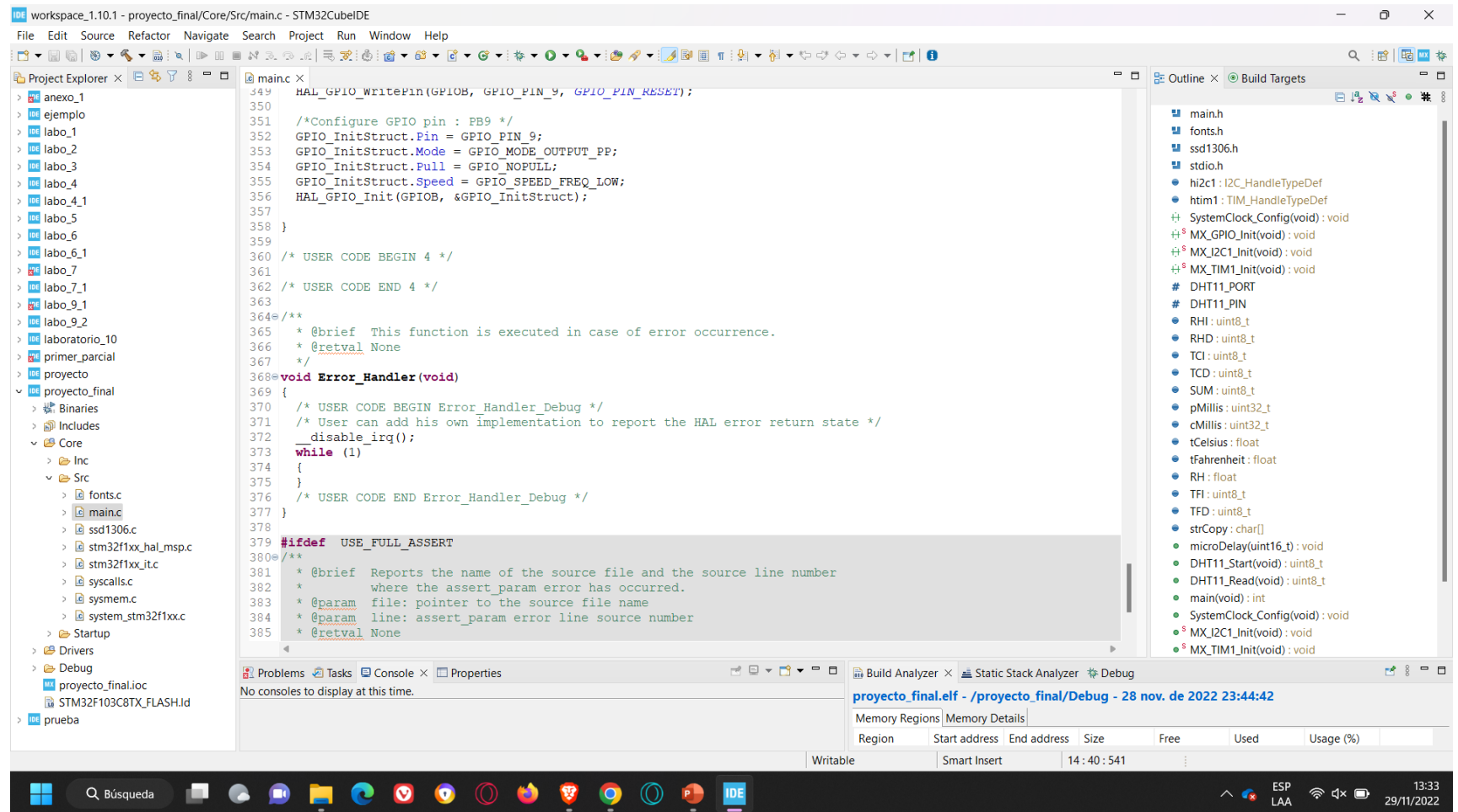
Vista del código
utilizado con las
modificaciones
sobre el template
de base (código de
base)



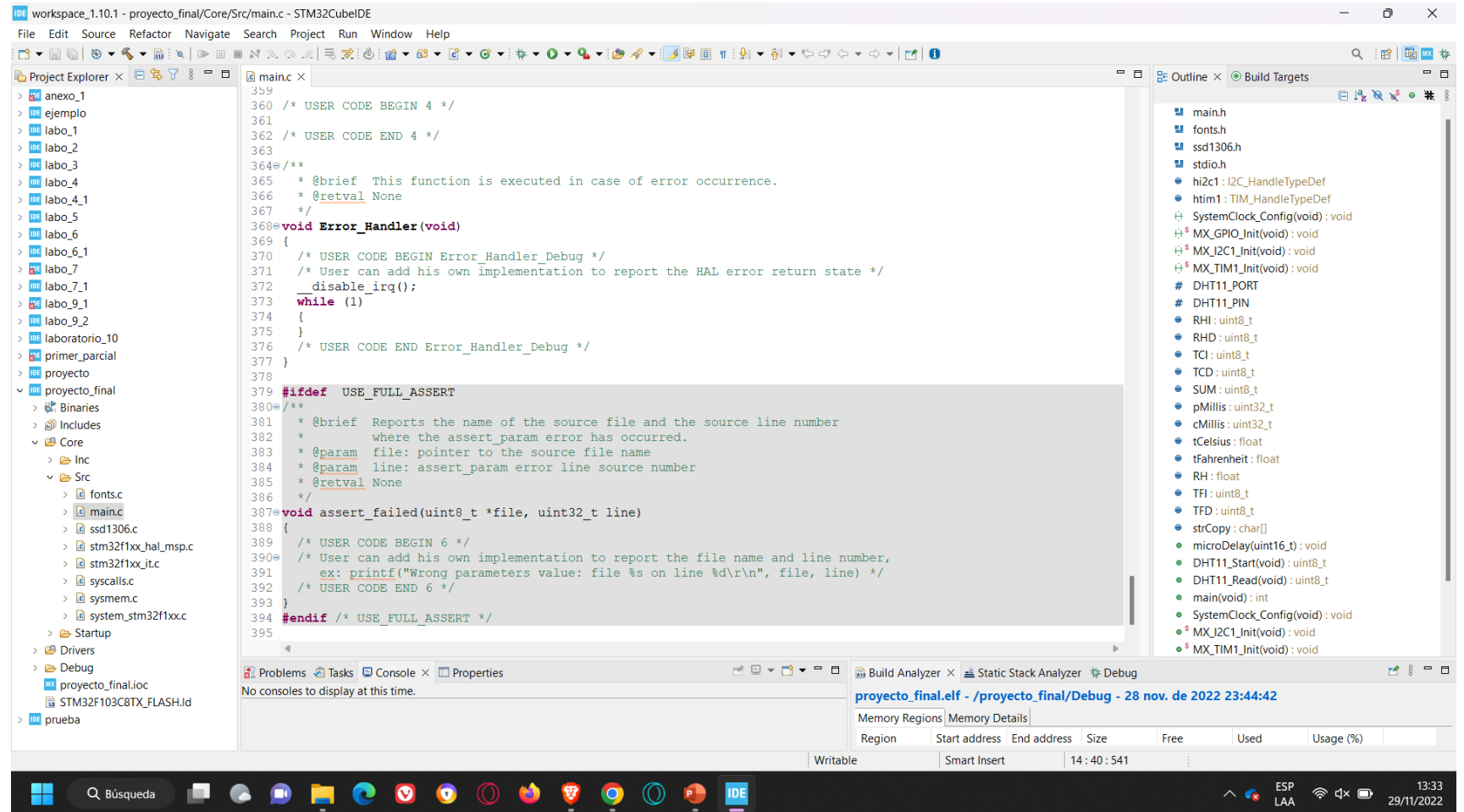
Vista del código
utilizado con las
modificaciones
sobre el template
de base (código de
base)



Vista del código
utilizado con las
modificaciones
sobre el template
de base (código de
base)



Vista del código
utilizado con las
modificaciones
sobre el template
de base (código de
base)



Explicación de partes puntuales del código

Corresponde a nuestras librerías y drivers del sensor de temperatura y de la pantalla oled.

```
C /* USER CODE BEGIN Includes */ Untitled-1 ●
1  /* USER CODE BEGIN Includes */
2  #include "fonts.h"
3  #include "ssd1306.h"
4  #include "stdio.h"
5  /* USER CODE END Includes */
~
```

Explicación de partes puntuales del código

Instanciamos los datos que nos permitirán realizar las conversiones de temperatura y humedad.

```
6
7  /* USER CODE BEGIN 0 */
8  #define DHT11_PORT GPIOB
9  #define DHT11_PIN GPIO_PIN_9
10 uint8_t RHI, RHD, TCI, TCD, SUM;
11 uint32_t pMillis, cMillis;
12 float tCelsius = 0;
13 float tFahrenheit = 0;
14 float RH = 0;
15 uint8_t TFI = 0;
16 uint8_t TFD = 0;
17 char strCopy[15];
```

Explicación de partes puntuales del código

Aquí se define la forma en la que tratará los pines el microcontrolador.

```
24
25 uint8_t DHT11_Start (void)
26 {
27     uint8_t Response = 0;
28     GPIO_InitTypeDef GPIO_InitStructure = {0};
29     GPIO_InitStructure.Pin = DHT11_PIN;
30     GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;
31     GPIO_InitStructure.Speed = GPIO_SPEED_FREQ_LOW;
32     GPIO_InitStructure.Pull = GPIO_NOPULL;
33     HAL_GPIO_Init(DHT11_PORT, &GPIO_InitStructure); // Asignar el pin como salida
34     HAL_GPIO_WritePin (DHT11_PORT, DHT11_PIN, 0); // Colocar el pin en valor bajo
35     HAL_Delay(20); // espera 20 microsegundos
36     HAL_GPIO_WritePin (DHT11_PORT, DHT11_PIN, 1); // Colocar el pin en valor alto
37     microDelay (30); // espera 30 microsegundos
38     GPIO_InitStructure.Mode = GPIO_MODE_INPUT;
39     GPIO_InitStructure.Pull = GPIO_PULLUP;
40     HAL_GPIO_Init(DHT11_PORT, &GPIO_InitStructure); // Asignar el pin como entrada
41     microDelay (40);
42     if (!(HAL_GPIO_ReadPin (DHT11_PORT, DHT11_PIN)))
43     {
44         microDelay (80);
45         if ((HAL_GPIO_ReadPin (DHT11_PORT, DHT11_PIN))) Response = 1;
46     }
47     pMillis = HAL_GetTick();
48     cMillis = HAL_GetTick();
49     while ((HAL_GPIO_ReadPin (DHT11_PORT, DHT11_PIN)) && pMillis + 2 > cMillis)
50     {
51         cMillis = HAL_GetTick();
52     }
53     return Response;
54 }
55
56 uint8_t DHT11_Read (void)
57 {
```

Explicación de partes puntuales del código

Aquí se configuran los parámetros iniciales.

```
56 uint8_t DHT11_Read (void)
57 {
58     uint8_t a,b;
59     for (a=0;a<8;a++)
60     {
61         pMillis = HAL_GetTick();
62         cMillis = HAL_GetTick();
63         while (!(HAL_GPIO_ReadPin (DHT11_PORT, DHT11_PIN)) && pMillis + 2 > cMillis)
64         { // esperar a que el pin vuelva a la condicion alta
65             cMillis = HAL_GetTick();
66         }
67         microDelay (40); //espera 40 microsegundos
68         if (!(HAL_GPIO_ReadPin (DHT11_PORT, DHT11_PIN))) // solo si el pin esta en la condicion baja
69             b&= ~(1<<(7-a));
70         else
71             b|= (1<<(7-a));
72         pMillis = HAL_GetTick();
73         cMillis = HAL_GetTick();
74         while ((HAL_GPIO_ReadPin (DHT11_PORT, DHT11_PIN)) && pMillis + 2 > cMillis)
75         { // espera a que el pin vuelva a la posicion baja
76             cMillis = HAL_GetTick();
77         }
78     }
79     return b;
80 }
```

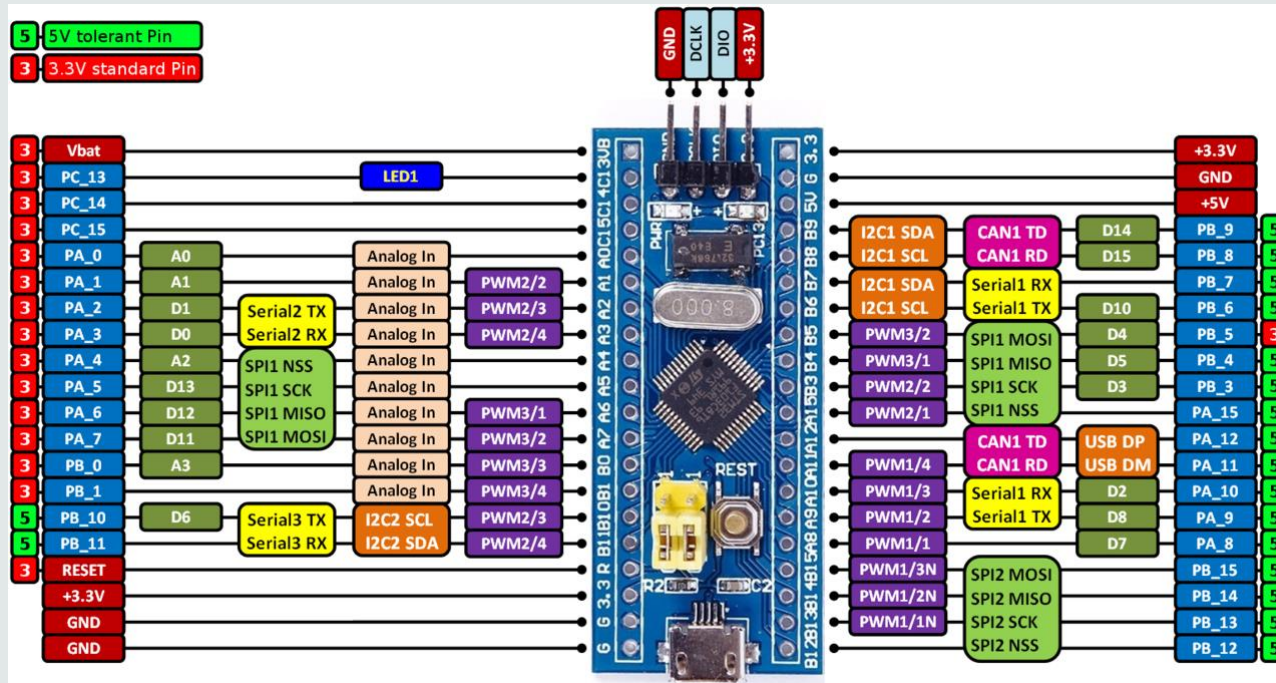
Explicación de partes puntuales del código

Aquí se realiza la lectura y conversión interna de unidades.

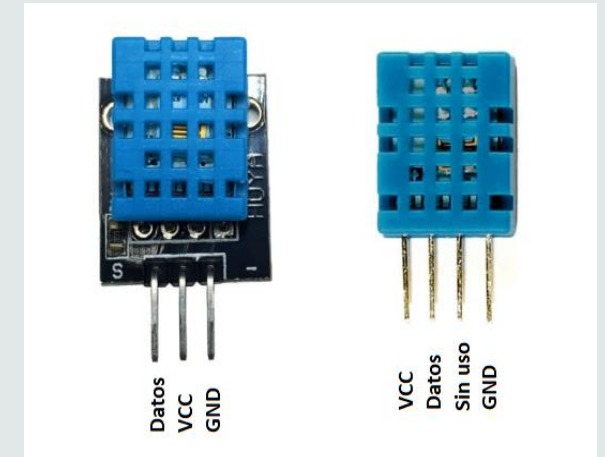
```
{
    if(DHT11_Start())
    {
        RHI = DHT11_Read(); // integral relativa de humedad
        RHD = DHT11_Read(); // humedad relativa decimal
        TCI = DHT11_Read(); // integral de grados celcius
        TCD = DHT11_Read(); // decimal de grados celcius
        SUM = DHT11_Read(); // verifica la suma
        if (RHI + RHD + TCI + TCD == SUM)
        {
            tCelsius = (float)TCI + (float)(TCD/10.0);
            tFahrenheit = tCelsius * 9/5 + 32;
            RH = (float)RHI + (float)(RHD/10.0);

            TFI = tFahrenheit;
            TFD = tFahrenheit*10-TFI*10;
            sprintf(strCopy,"%d.%d C  ", TCI, TCD);
            SSD1306_GotoXY (0, 0);
            SSD1306_Puts (strCopy, &Font_11x18, 1);
            sprintf(strCopy,"%d.%d F  ", TFI, TFD);
            SSD1306_GotoXY (0, 20);
            SSD1306_Puts (strCopy, &Font_11x18, 1);
            sprintf(strCopy,"%d.%d %% ", RHI, RHD);
            SSD1306_GotoXY (0, 40);
            SSD1306_Puts (strCopy, &Font_11x18, 1);
            SSD1306_UpdateScreen();
        }
    }
}
```

MATERIALES DEL PROYECTO



TARJETA DE DESARROLLO STM32F103C8T6

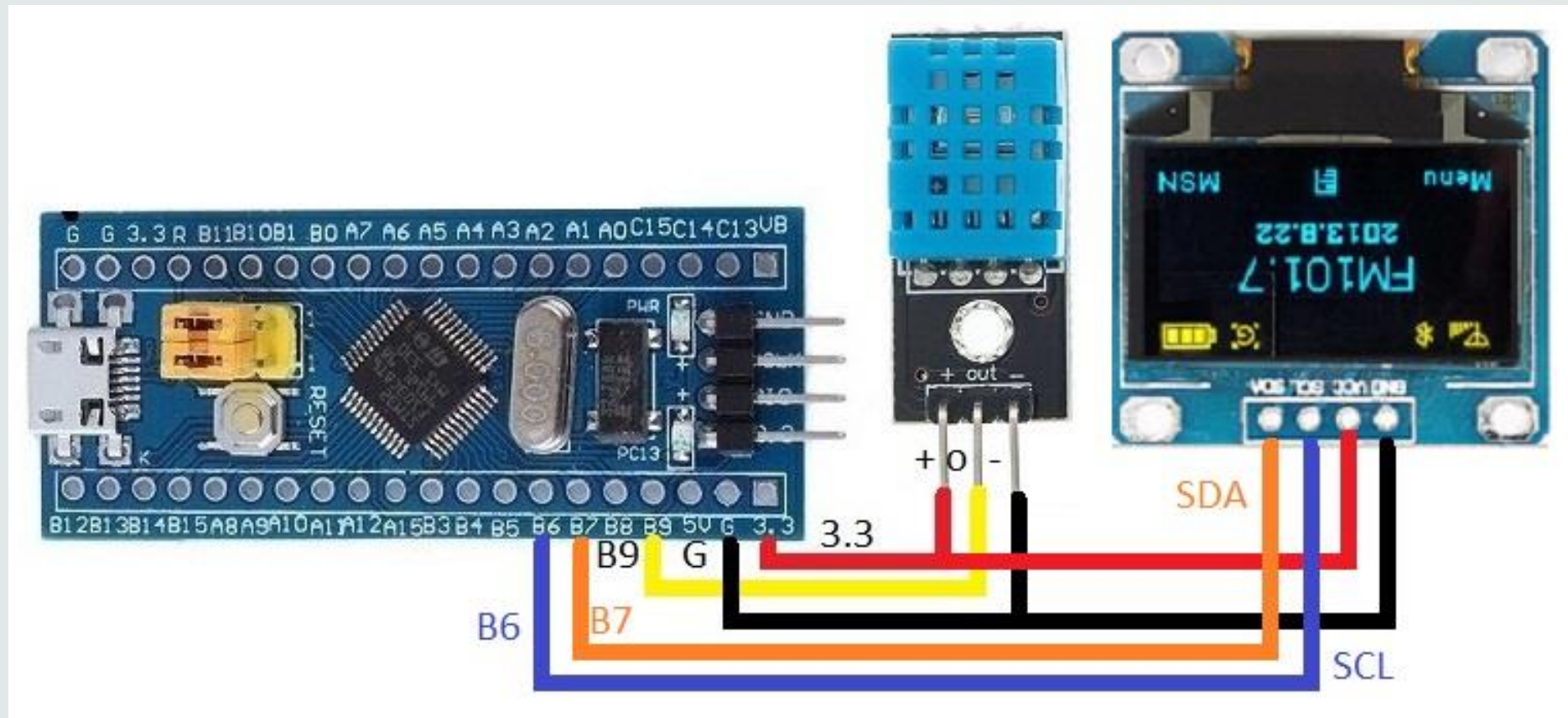


SENSOR DE TEMPERATURA Y HUMEDAD DHT11

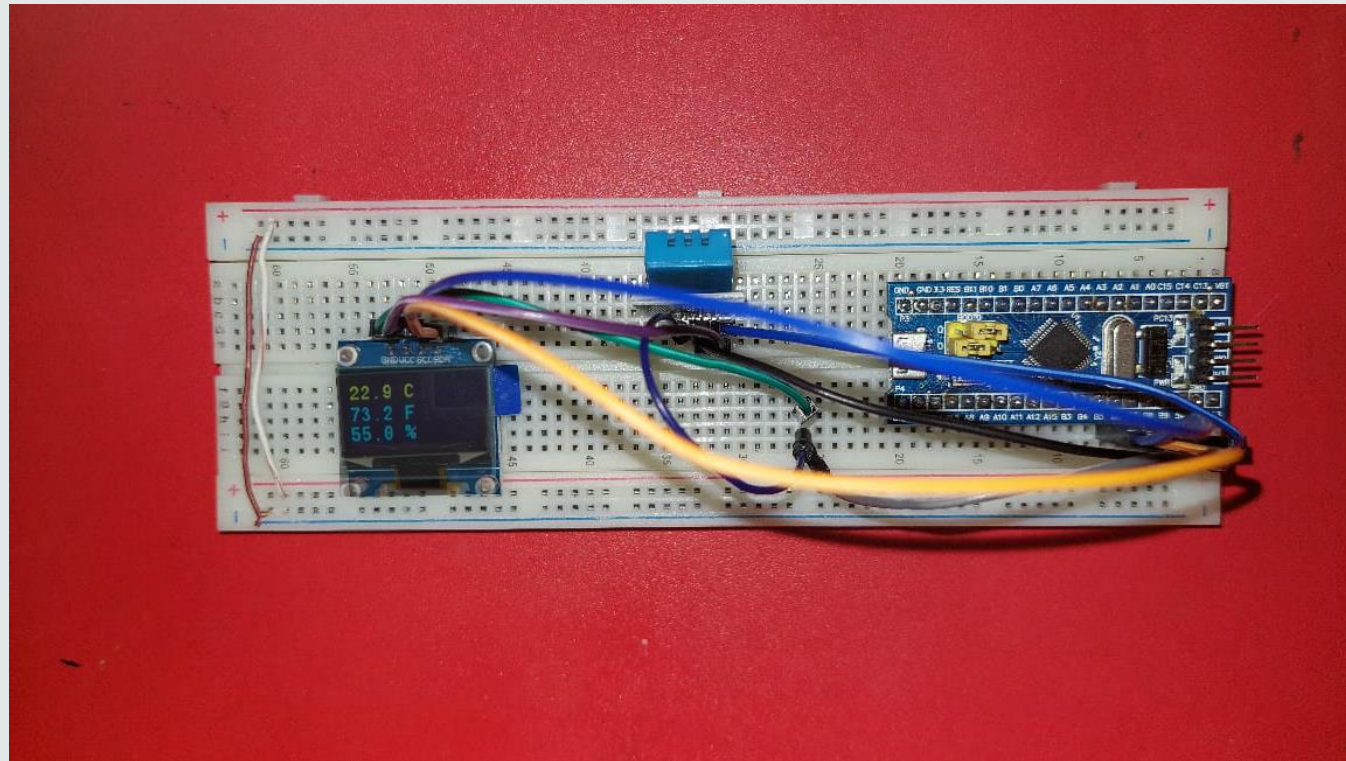


PANTALLA OLED I2C

DIAGRAMA ESQUEMÁTICO DEL CIRCUITO



CIRCUITO FUNCIONANDO



ENCUENTRA EL PROYECTO EN GITHUB

https://github.com/AmirOros/proyecto_final.git



FIN

