



Facultad Regional Tucumán
Departamento Electrónica

Técnicas Digitales II

Actividad de Formación Práctica 1

**Tema: Entorno de Desarrollo Integrado STM32CubeIDE,
programación de micro controladores.**

Profesores:

Ing. Rubén Darío Mansilla

ATTP:

Ing. Lucas Abdala

Grupo 5

Integrante:

- Angelillo Alessandro
- Plaate Iván
- Torres Juan
- Hualampa Leonel Javier

- 1) Elabore un tutorial en soporte digital, que explique cómo se aborda el uso del **STM32CubeIDE**, que contenga lo que se indica a continuación:

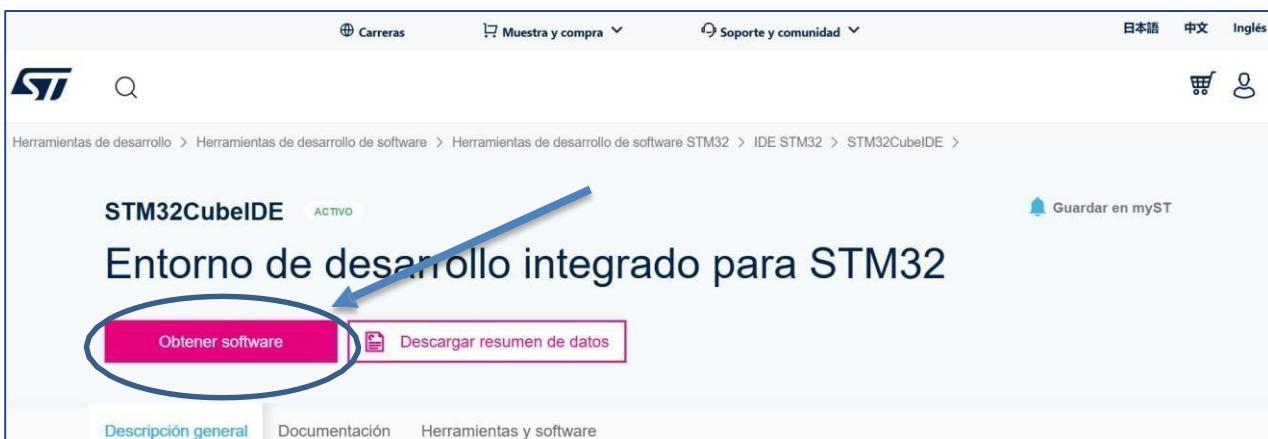
Instalacion

- **COMO DESCARGAR**

Primer paso descargamos el software **STM32CubeIDE** haciendo click en el siguiente link que le dejamos a continuación:

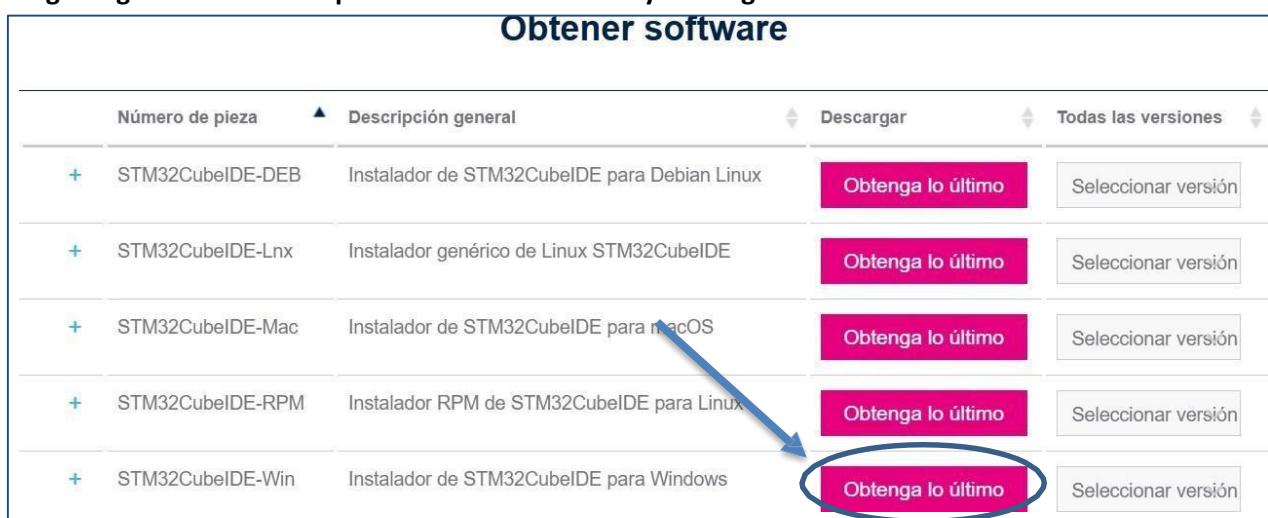
[Descargar stm32cubeide](#)

Nos dirigimos a la siguiente casilla y hacemos click



The screenshot shows the STM32CubeIDE download page. At the top, there are navigation links for Carreras, Muestra y compra, Soporte y comunidad, and language options (日本語, 中文, Inglés). Below the header, the page title is "STM32CubeIDE ACTIVO". A large heading says "Entorno de desarrollo integrado para STM32". Below it are two buttons: "Obtener software" (highlighted with a blue oval and arrow) and "Descargar resumen de datos". At the bottom of the main content area, there are tabs for Descripción general, Documentación, and Herramientas y software.

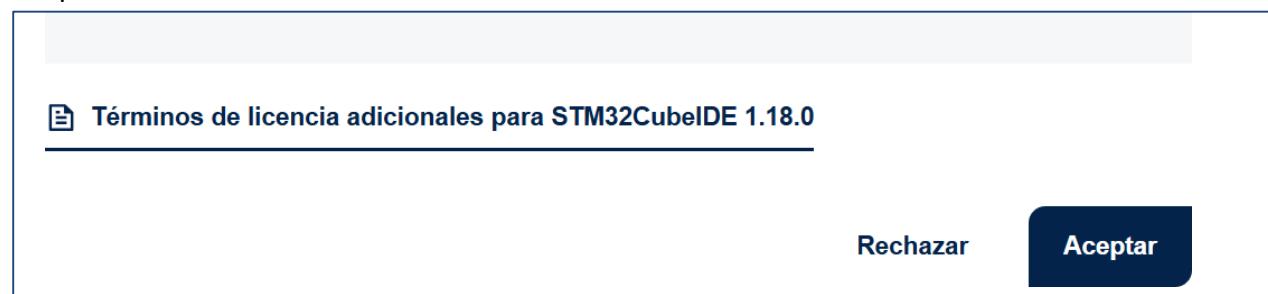
Luego elegimos el sistema operativo estamos usando y descargamos la última versión.



The screenshot shows the "Obtener software" page. It lists five download options for different operating systems: STM32CubeIDE-DEB (Debian Linux), STM32CubeIDE-Lnx (generic Linux), STM32CubeIDE-Mac (macOS), STM32CubeIDE-RPM (Linux RPM), and STM32CubeIDE-Win (Windows). Each row has a "Descargar" button. The "STM32CubeIDE-Win" row is circled with a blue oval and has an arrow pointing to its "Descargar" button, which is also highlighted with a pink oval.

Número de pieza	Descripción general	Descargar	Todas las versiones
STM32CubeIDE-DEB	Instalador de STM32CubeIDE para Debian Linux	Obtenga lo último	Seleccionar versión
STM32CubeIDE-Lnx	Instalador genérico de Linux STM32CubeIDE	Obtenga lo último	Seleccionar versión
STM32CubeIDE-Mac	Instalador de STM32CubeIDE para macOS	Obtenga lo último	Seleccionar versión
STM32CubeIDE-RPM	Instalador RPM de STM32CubeIDE para Linux	Obtenga lo último	Seleccionar versión
STM32CubeIDE-Win	Instalador de STM32CubeIDE para Windows	Obtenga lo último	Seleccionar versión

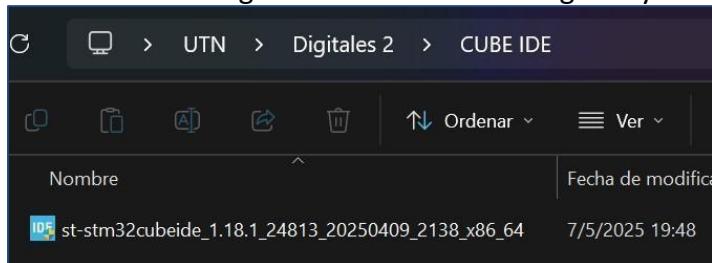
Aceptamos el acuerdo de licencia



The screenshot shows a dialog box asking for license acceptance. It contains the text "Términos de licencia adicionales para STM32CubeIDE 1.18.0" and two buttons at the bottom: "Rechazar" and "Aceptar". The "Aceptar" button is highlighted with a pink oval.

Una vez ahí nos pedirá si queremos (**Descargar como invitado**) o (**Crear una cuenta en MyST**)

Cuando se descarga el archivo veremos algo así y hacemos doble click sobre el

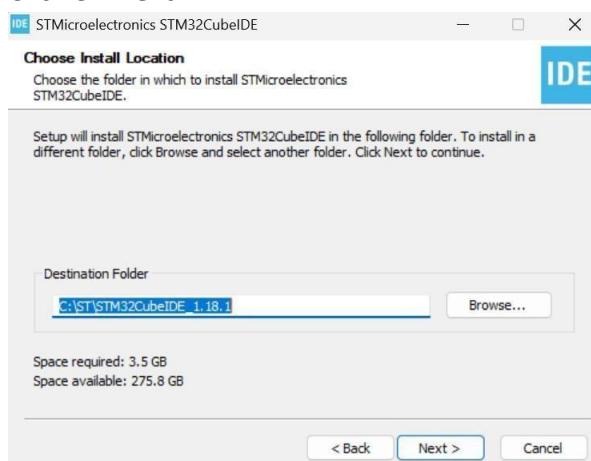


- **COMO INSTALAR**

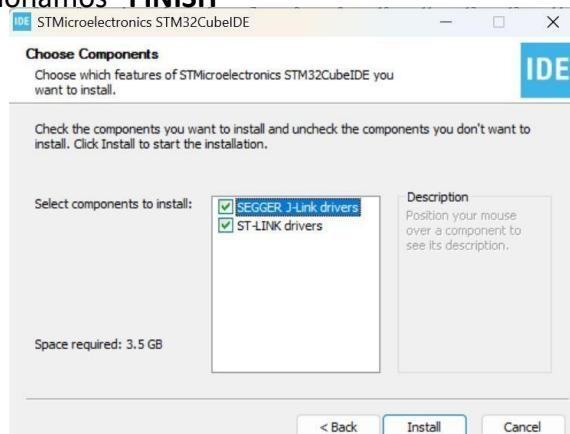
Se nos abre la siguiente ventana y hacemos click en “**Next**”



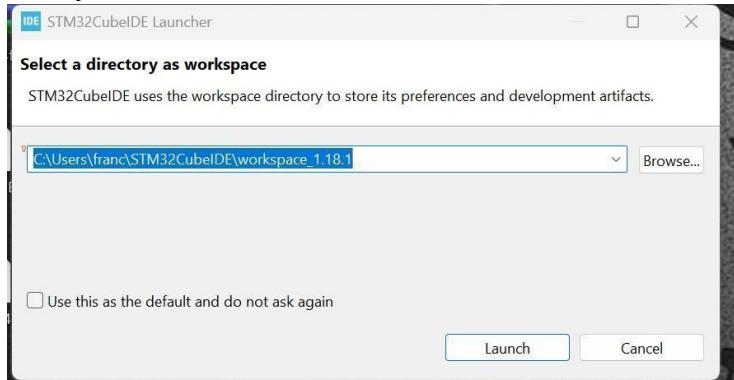
Click en **Next**



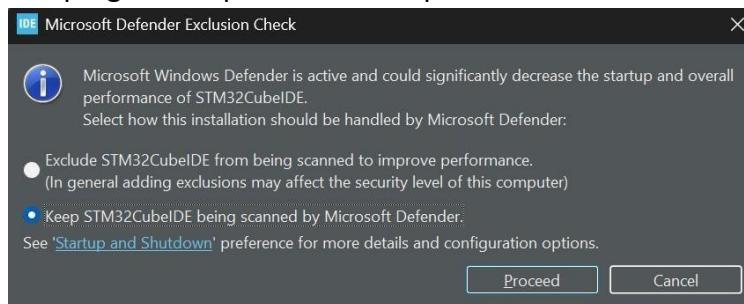
Dejamos marcados los componentes que deseamos que se instalen. Esperamos y presionamos “**FINISH**”



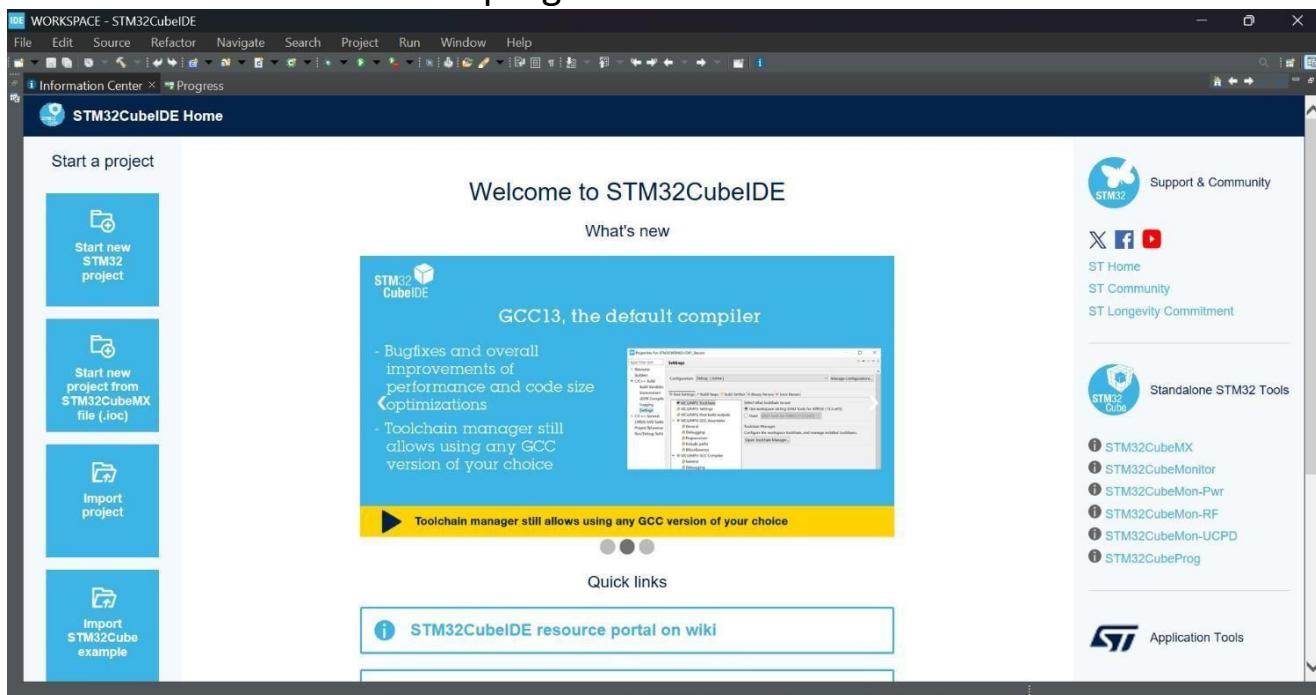
- La Primera vez que ejecutamos el programa nos pedirá una Dirección para el espacio de trabajo Presionamos en “Launch”



Nos pregunta si queremos darle permiso Windows defender

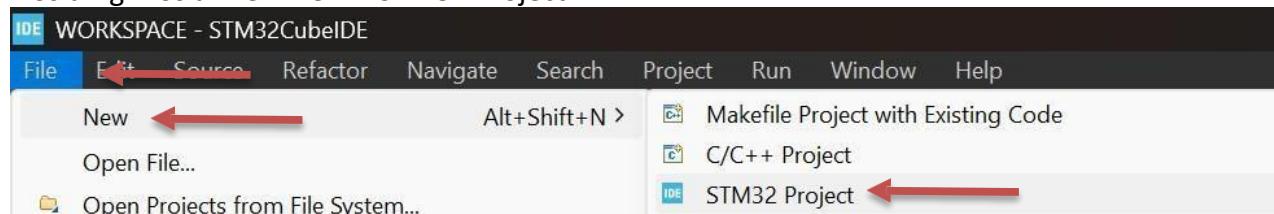


- Primera vista del programa

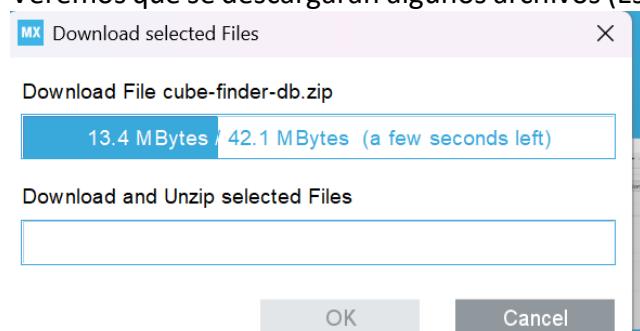


Crear Un Nuevo Proyecto

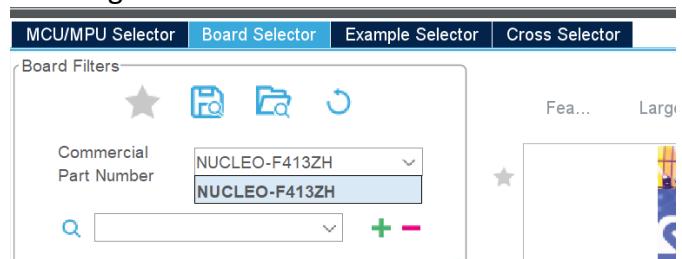
Nos dirigimos a File – New – STM32 Project



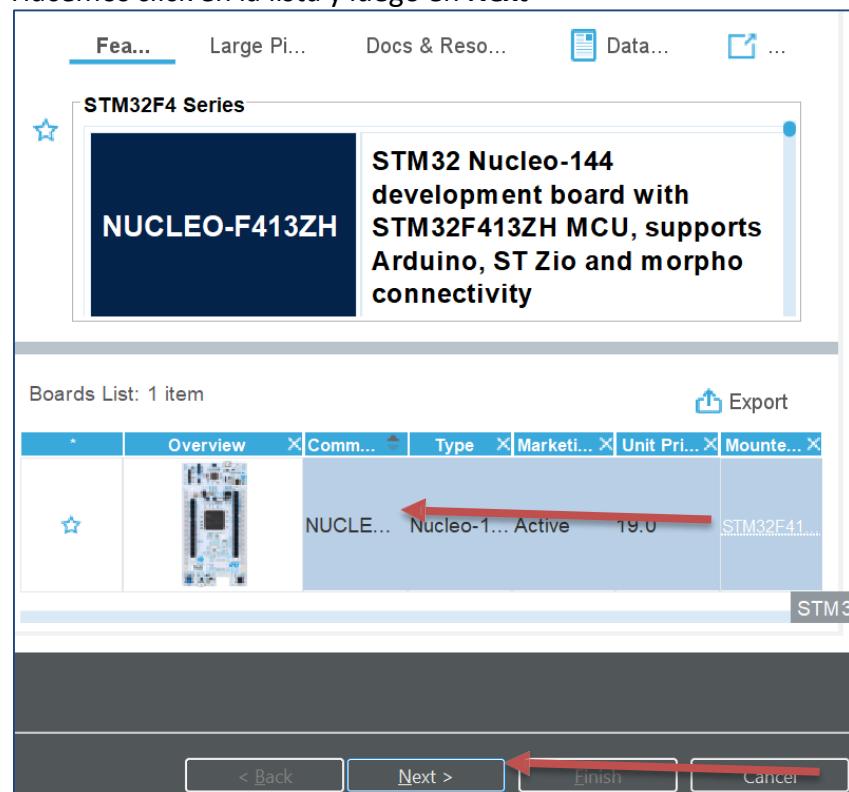
Veremos que se descargaran algunos archivos (Esperamos)



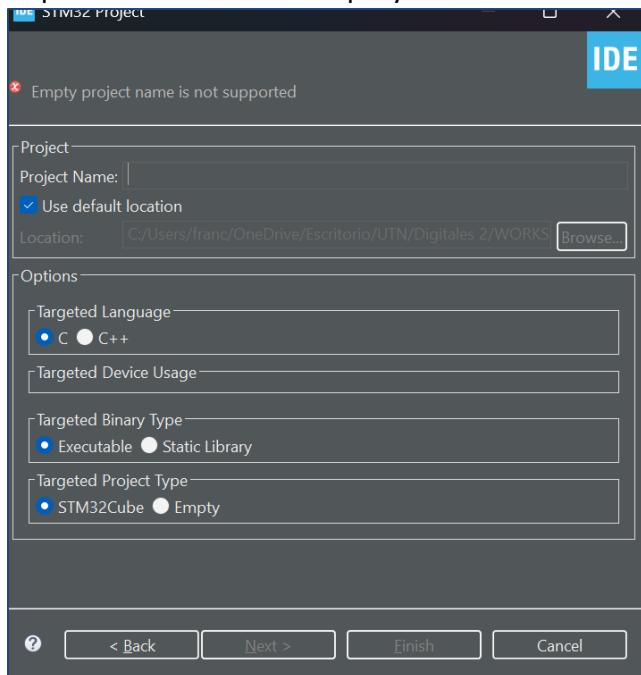
Nos dirigimos a Board Selector – Comercial Part Number – NUCLEO – NUMERO DE LA PLACA



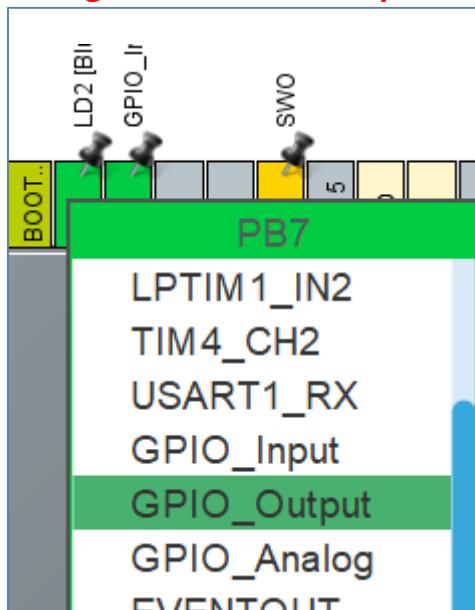
Hacemos click en la lista y luego en Next



Le ponemos un nombre al proyecto

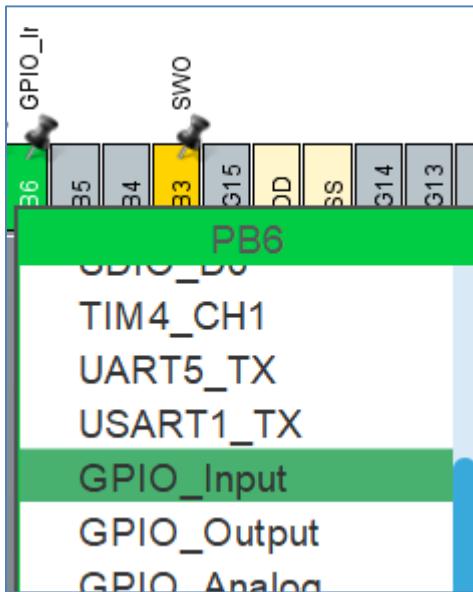


Configuraciones básicas para el uso del modulo GPIO



Hacemos click sobre el PIN **PB7** y definimos al LED2 como salida **GPIO_OUTPUT**

Hacemos click sobre el PIN **PB6** Definimos como pulsador de usuario **GPIO_Input**



Estructura del arbol de archivos

Al crear un proyecto podemos ver:

- **Core**: Código fuente
- **Inc**: Archivo cabecera (contiene **main.h**)
- **Src**: Archivo fuente (contiene **main.c**)
- **Drivers**: Bibliotecas por defecto
- **.ioc**: Configuración del proyecto

The screenshot shows a development environment interface with a menu bar: File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help. Below the menu is a toolbar with various icons. The main area has two panes: 'Project Explorer' on the left and a code editor on the right. The Project Explorer shows a project structure with several folders: 'App 1 1 Grupo 2 2025' (containing 'Binaries', 'Includes', 'Core' (with 'Inc' and 'Src' subfolders), 'Startup', 'Drivers', 'Debug', and files like 'main.c', 'stm32f4xx_hal_msp.c', etc.), and 'STM32F413ZHTX_FLASH.Id' and 'STM32F413ZHTX_RAM.Id'. The code editor shows the 'main.c' file with the following content:

```

1  /* USER CODE BEGIN Header */
2 /**
3  ****
4  * @file           : main.c
5  * @brief          : Main program body
6  ****
7  * @attention
8  *
9  * Copyright (c) 2025 STMicroelectronics.
10 * All rights reserved.
11 *
12 * This software is licensed under terms tha
13 * in the root directory of this software co
14 * If no LICENSE file comes with this softwa
15 *
16 ****
17 */
18 /* USER CODE END Header */
19 /* Includes -----
20 #include "main.h"
21 /* Private includes -----
22 /* USER CODE BEGIN Includes */

```

Desarrollo de un proyecto que utilice uno de los leds integrados en la placa y lo haga parpadear cada 250 ms.

The screenshot shows the STM32CubeIDE interface. In the code editor, the main.c file contains C code for initializing USART3, USB OTG FS PCD, and a GPIO pin (LD1_Pin) which toggles every 250ms. A callout box highlights the code segment from line 102 to 108. Below the code editor is a terminal window showing a successful build message: "01:41:55 Build Finished. 0 errors, 0 warnings. (took 1s.696ms)".

```

MX_USART3_UART_Init();
MX_USB_OTG_FS_PCD_Init();
/* USER CODE BEGIN 2 */

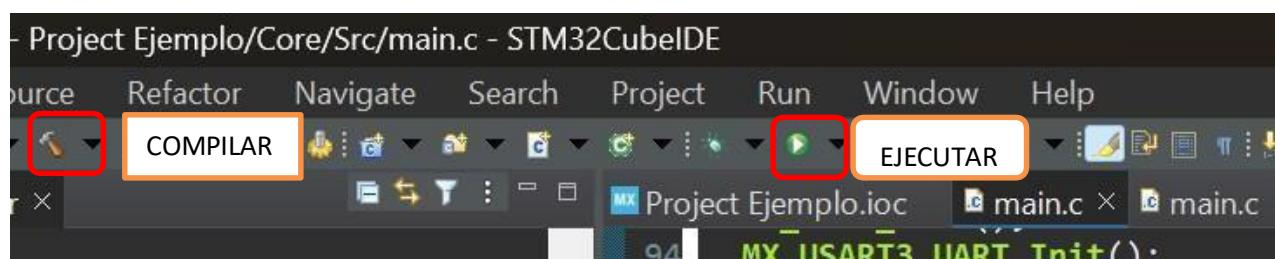
/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
    HAL_GPIO_WritePin(GPIOB, LD1_Pin, !HAL_GPIO_ReadPin(GPIOB, LD1_Pin));
    HAL_Delay(250);
}
/* USER CODE END 3 */

/**
 * @brief System Clock Configuration
 * @retval None
 */
void SystemClock_Config(void)
{
    RCC_OscInitTypeDef RCC_OscInitStruct = {0};

```

Para compilar



Para cargar el programa en la placa debemos

- Conectar la placa por **USB**
- Presionar **Ejecutar o Debug**
- Veremos que se carga el programa (En la placa veremos una luz que parpadea)
- Luego se encenderá el **led** cada 250ms.

Desarrollo de Aplicaciones: (Hacer click en el hipervínculo para ver el código)

App 1_1: Desarrolle una aplicación que encienda y apague de manera secuencial los tres leds onboard de la placa de desarrollo. La secuencia debe encender 200 ms y apagar 200 ms cada led comenzando por el LED1 (Green), continuando con el LED2 (Blue) y luego el LED3 (Red) para volver a iniciar con el LED1. La aplicación debe ser de carácter general, de manera que pueda extenderse a una cantidad mayor de leds con mínimas modificaciones, por este motivo se sugiere que use un vector para el manejo de los Leds

[App 1_1](#)

App 1_2: Desarrolle una aplicación que utilice el pulsador onboard de la placa de desarrollo para alternar entre dos secuencias diferentes. La aplicación inicia con la secuencia de la App 1.1 y, cuando se presione el pulsador, dicha secuencia debe invertirse y continuar, de manera que, cada vez que se presione el pulsador la secuencia actual se invierta. La aplicación debe ser de carácter general, por lo que aplica la misma recomendación para el punto anterior.

[APP1_2](#)

App 1_3: Desarrolle una aplicación que utilice el pulsador onboard de la placa de desarrollo para alternar entre cuatro secuencias diferentes. La app iniciara con la secuencia 1 hasta que se presione el pulsador y pase a la secuencia 2, luego de presionar de nuevo el pulsador pasara a la secuencia 3 y así, sucesivamente, hasta la secuencia 4 para volver a comenzar con la secuencia 1. Descripción de las secuencias:

- Secuencia 1: idem App 1.1 con una alternancia de 150 ms.
- Secuencia 2: hace parpadear los tres leds simultáneamente con una alternancia de 300 ms.
- Secuencia 3: hace parpadear el LED1 con una alternancia de 100 ms, el LED2 con una alternancia de 300 ms y el led3 con una alternancia de 600 ms.
- Secuencia 4: Hace parpadear simultáneamente LED1 y LED3, mientras que LED2 lo hará de manera inversa, con una alternancia de 150 ms. Mientras LED1 y LED3 estan encendidos, LED2 estará apagado y luego a la inversa.

[App1_3](#)

App 1_4: Desarrolle una aplicación que haga parpadear simultáneamente los 3 leds onboard de la placa y que use el pulsador onboard para cambiar la frecuencia de parpadeo de manera secuencial entre 4 frecuencias predefinidas por los siguientes tiempos de alternancia:

- Tiempo 1: 100 ms.
- Tiempo 2: 250 ms.
- Tiempo 3: 500 ms.
- Tiempo 4: 1000 ms.

La aplicación inicia con la alternancia determinada por el tiempo 1, al presionar el pulsador pasa al tiempo 2 y así sucesivamente hasta llegar al tiempo 4, para reiniciar con el tiempo 1.

[App1_4](#)