

Tema 5: Entorno de Desarrollo

Sistemas Digitales Basados en Microprocesadores

Universidad Carlos III de Madrid

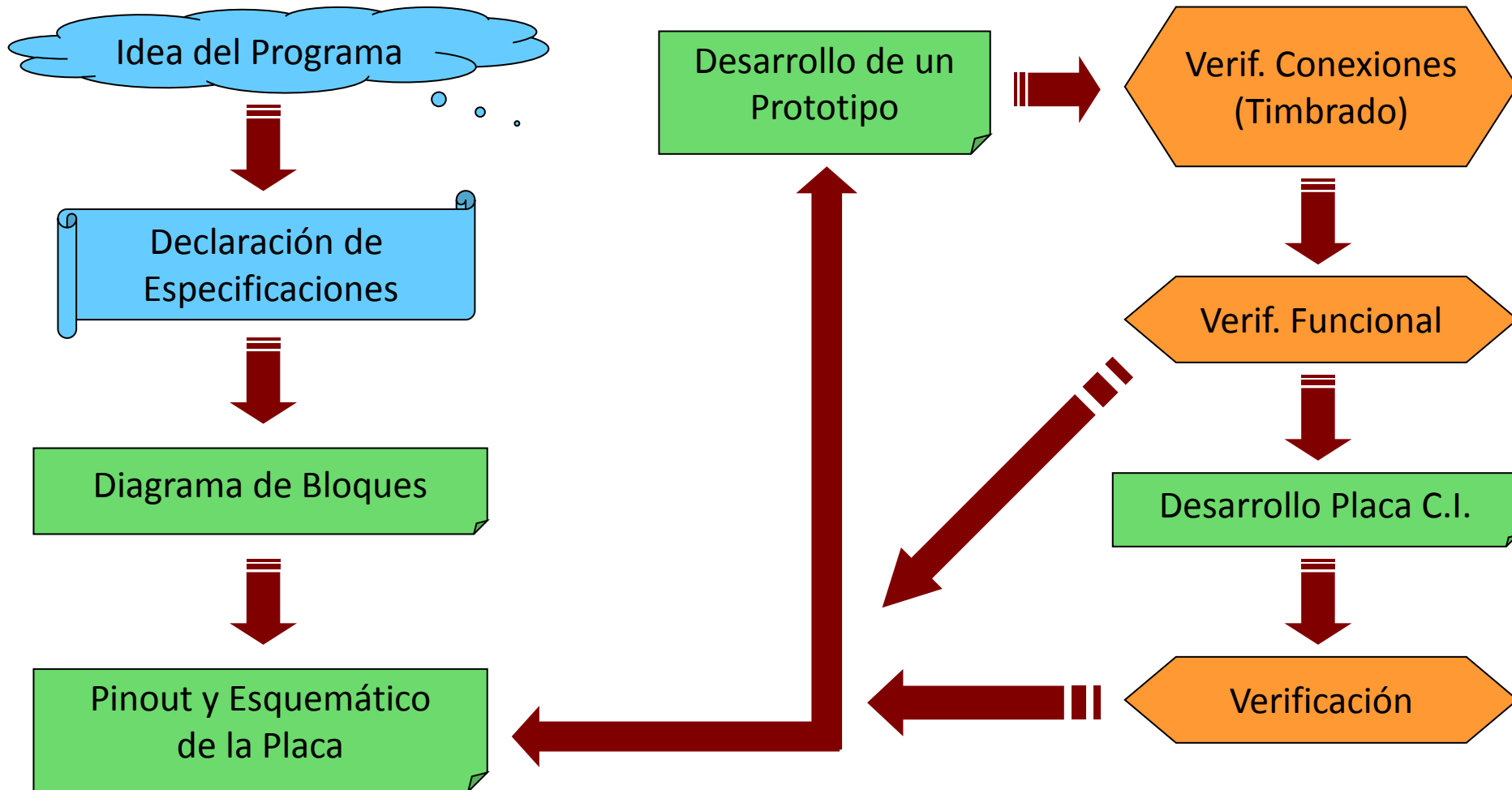
Dpto. Tecnología Electrónica

Índice

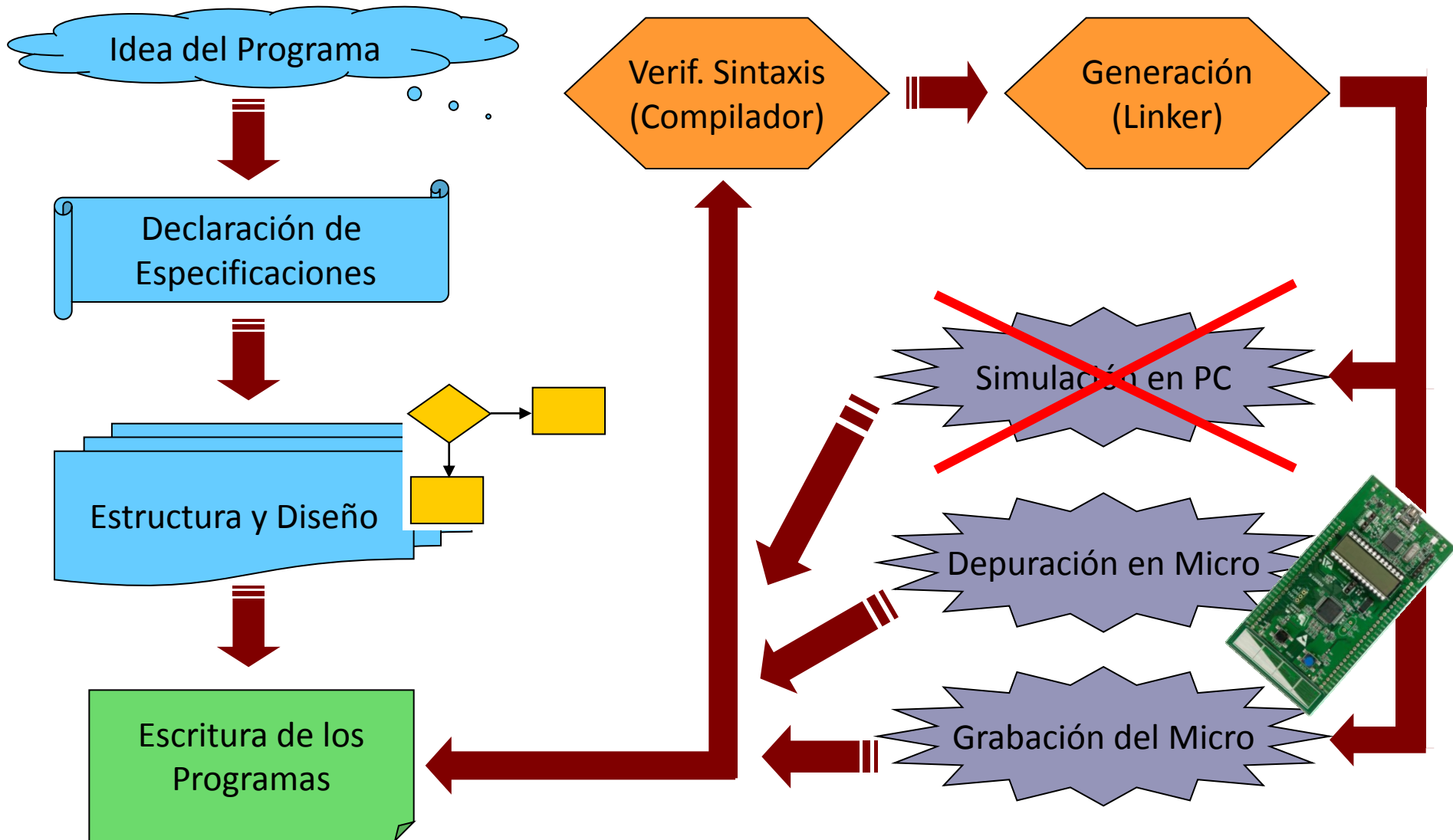
- 1 - Ciclo de Desarrollo
- 2 - Diagramas de Flujo
- 3 - La placa de Desarrollo STM32L-DISCOVERY
- 4 - Instalación de μ Vision5
- 5 - Instalación de STM32 Cube MX
- 6 - Primera ejecución del STM32 Cube MX
- 7 - Creación de la estructura básica de trabajo
- 8 - Pasos para la creación de un proyecto
- 9 - Pasos para la depuración de un proyecto
- 10 - Peculiaridades de la Programación en C en Microcontroladores
- 11 - Recomendaciones de Uso de la Placa de Desarrollo
- 12 - Ejercicios

1 - Ciclo de Desarrollo

Ciclo de Desarrollo Hardware



Ciclo de Desarrollo Software



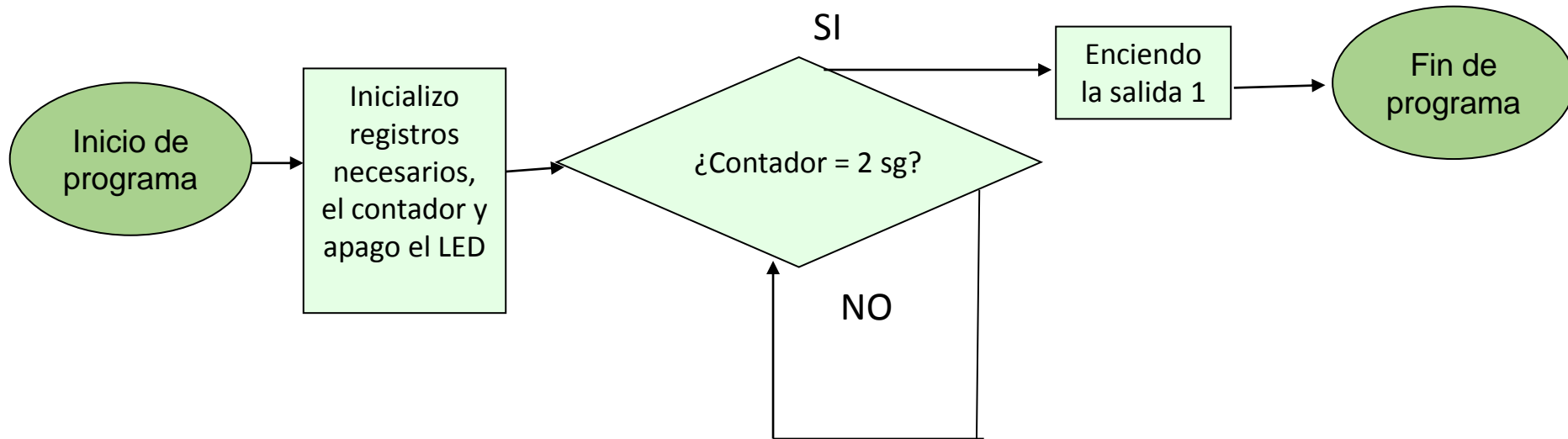
2 - Diagramas de Flujo

Diagramas de Flujo

- Son representaciones del funcionamiento de un programa
 - De forma genérica – independiente de la arquitectura
 - Nunca pueden contener referencias a registros de la arquitectura utilizada, ni a instrucciones de la CPU
 - Que muestran la solución al problema planteado
 - Que tienen que servir de guía, tanto al programador como a los posibles programadores que tengan que tocar ese programa
- Se pueden escribir a distintos niveles de detalle/abstracción
 - El nivel de detalle que debe ser utilizado dependerá de la situación
- Tradicionalmente se utilizarán sólo los símbolos sencillos:
 - Elipse o círculo, para indicar una etiqueta
 - Rectángulo, para indicar un proceso
 - Rombo, para indicar una decisión

Ejemplo

Programa que espera 2 segundos para encender un LED en la salida digital PA1 del micro y luego la deja encendida para siempre



3 - La placa de Desarrollo STM32L-DISCOVERY

STM32L-Discovery



STM32L-DISCOVERY

STM32L ultralow power discovery board

Data brief

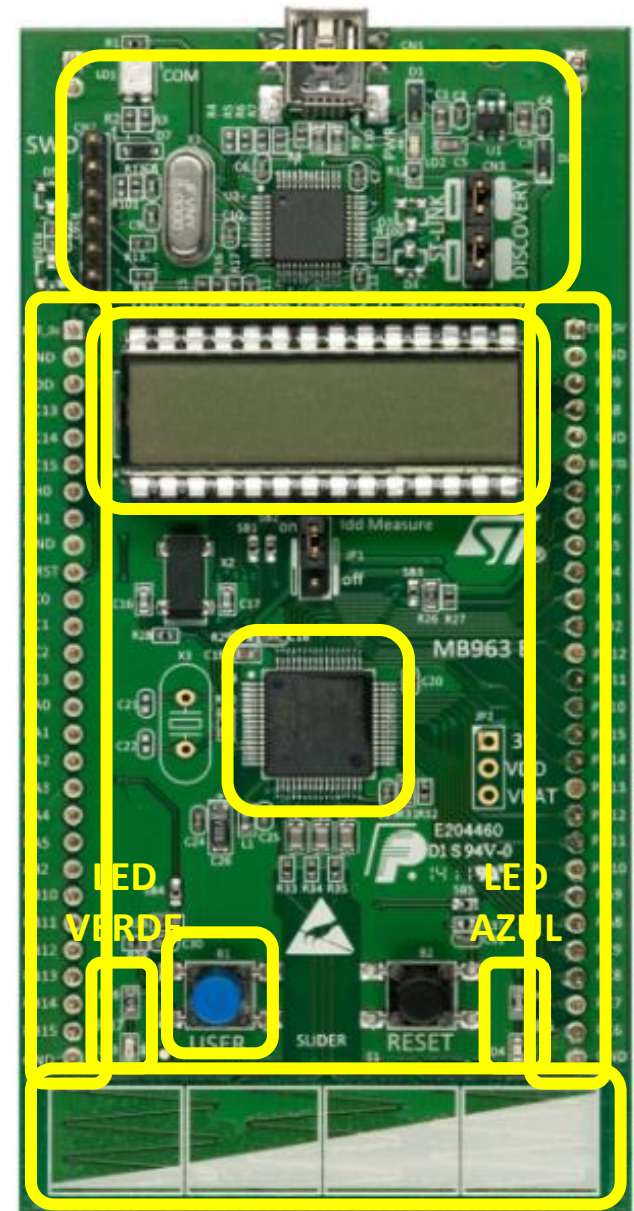
Features

- STM32L152RBT6 microcontroller featuring 128 KB Flash, 16 KB RAM, 4 KB EEPROM, in an LQFP64 package
- On-board ST-Link/V2 with selection mode switch to use the kit as a standalone ST-Link/V2 (with SWD connector for programming and debugging)
- Board power supply: through USB bus or from an external 3.3 or 5 V supply voltage
- External application power supply: 3 V and 5 V
- I_{DD} current measurement
- LCD
 - DIP28 package
 - 24 segments, 4 commons
- Four LEDs:
 - LD1 (red/green) for USB communication
 - LD2 (red) for 3.3 V power on
 - Two user LEDs, LD3 (green) and LD4 (blue)
- Two pushbuttons (user and reset)
- One linear touch sensor or four touchkeys
- Extension header for LQFP64 I/Os for quick connection to prototyping board and easy probing

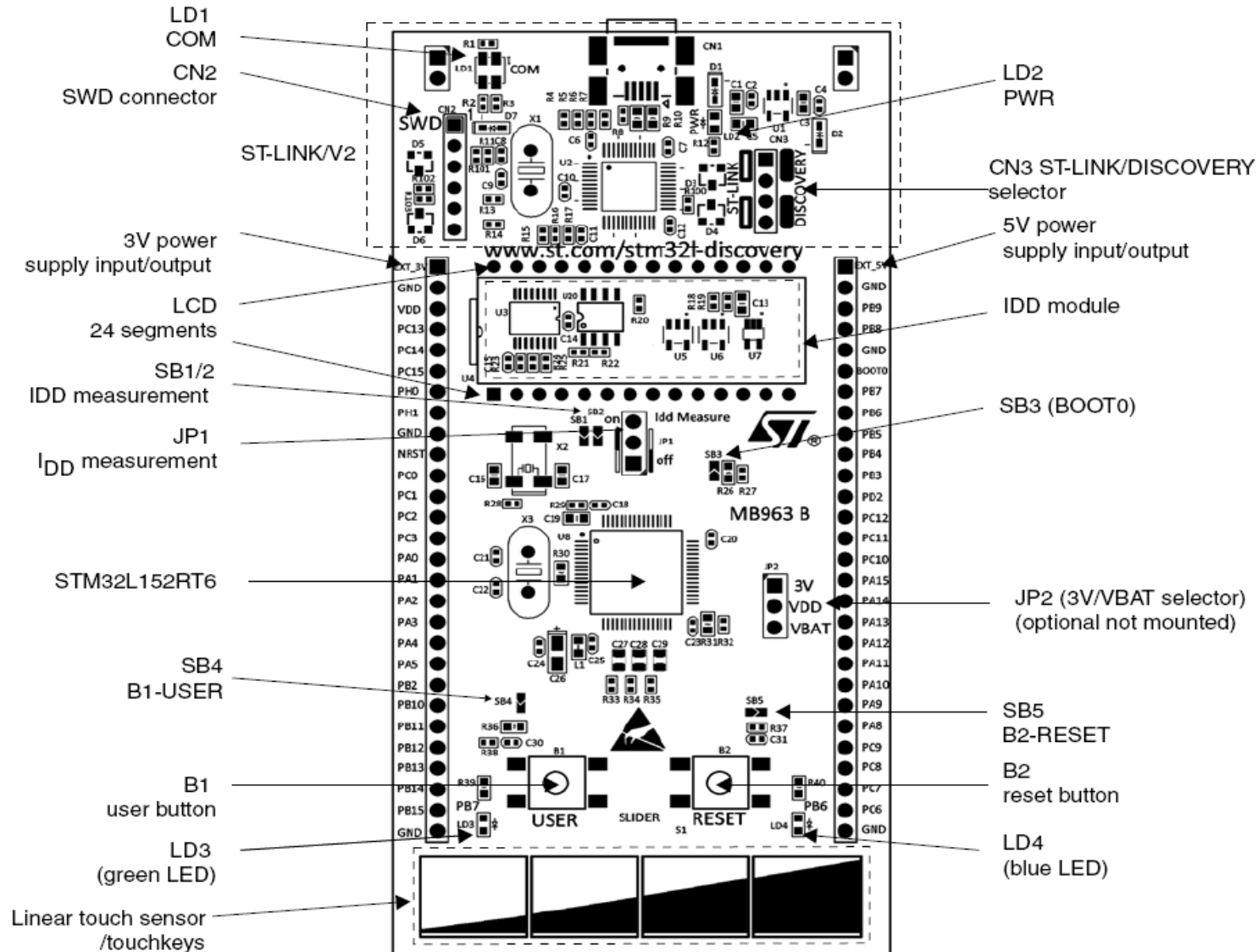


STM32L-Discovery

- La placa de desarrollo tiene las siguientes funcionalidades:
 - Microcontrolador STM32L152RB
 - Interfaz de depuración ST-LINK/V2 incluido (conectado al ordenador a través de Mini-USB)
 - Una pantalla LCD de 24 segmentos y 4 comunes, o sea, 6 dígitos
 - 4 LEDs
 - 2 de ellos programables por el usuario (LED_VERDE, LED AZUL)
 - Un sensor táctil lineal, con posibilidad de ser utilizado como 4 teclas individuales
 - Botón programable por el usuario (USER)
 - 2 Puertos de expansión a placa adicional (P1 y P2)

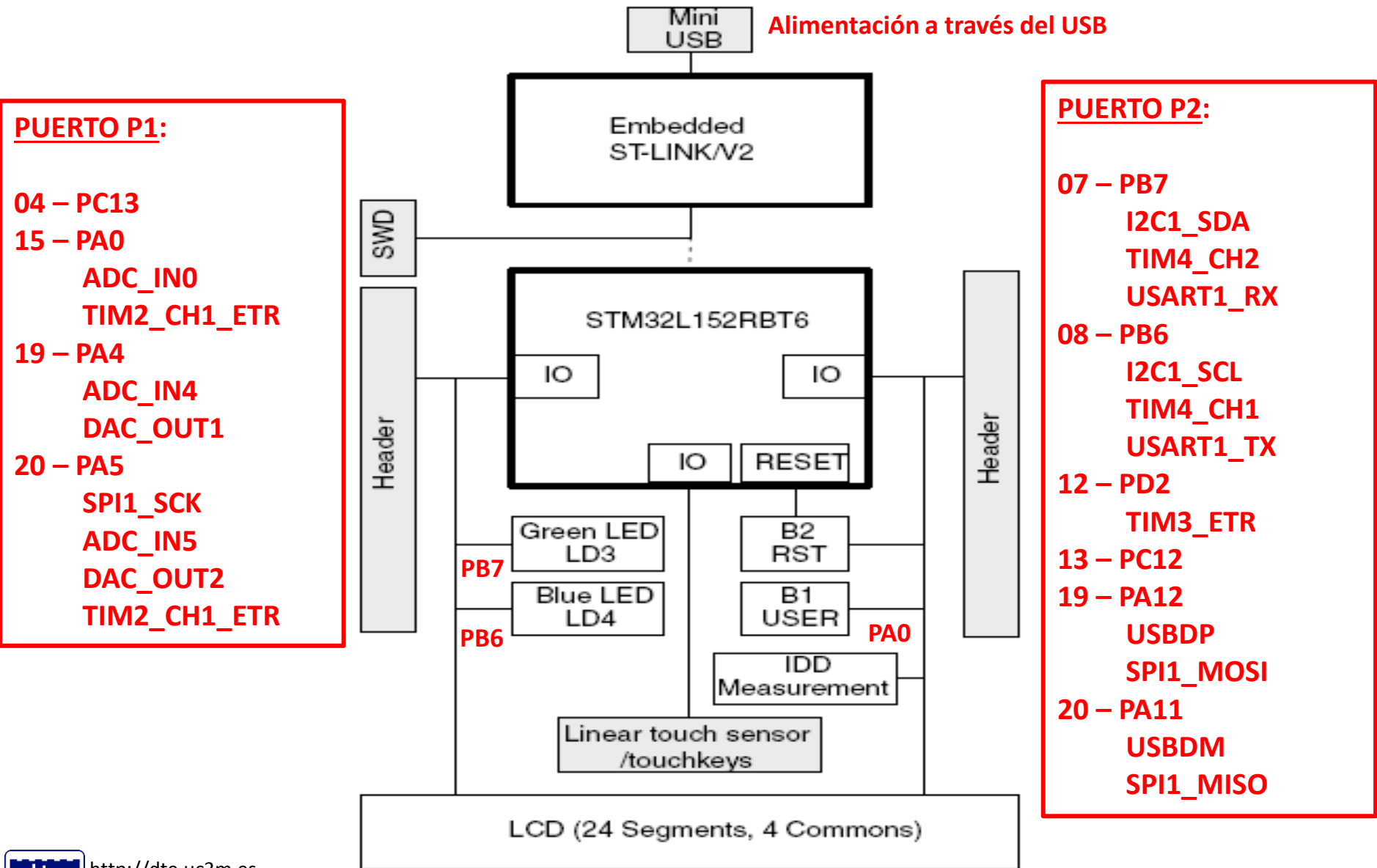


D. de Bloques y Layout



MS19049V1

Diagrama de Bloques y Layout



Préstamo de la Placa de Desarrollo

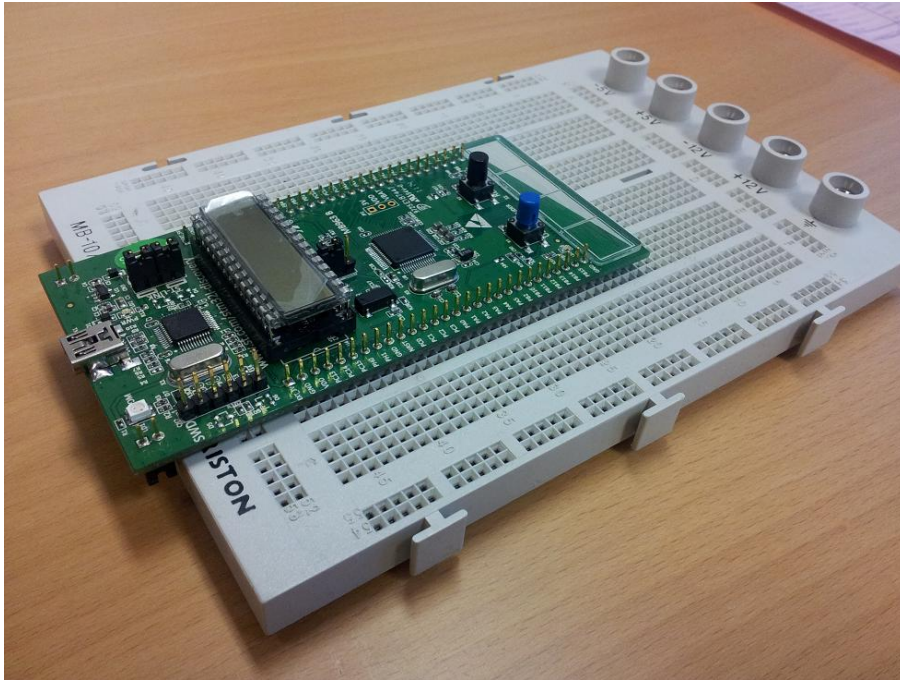
- Durante el curso, se va a permitir el préstamo personal de:
 - La Placa de Desarrollo STM32L-Discovery
- El préstamo se hace con la condición de que el material se devuelva antes del examen en perfecto estado
 - De no ser así, los alumnos implicados no serán evaluados en la asignatura
- Para obtener el material:
 - Imprimir, rellenar y firmar **DOS** copias del formulario de préstamo del material. El formulario se encuentra en Aula Global.
 - Entregar al coordinador de la asignatura el formulario, para que lo firme y se quede con una copia.
 - Acercarse al despacho 1.1.I.04 o 1.1.I.03 para entregar la autorización firmada por el profesor, y recoger el material.
- El material se devolverá en los mismos despachos, entregándolo con el formulario de devolución **ANTES DEL EXAMEN FINAL**



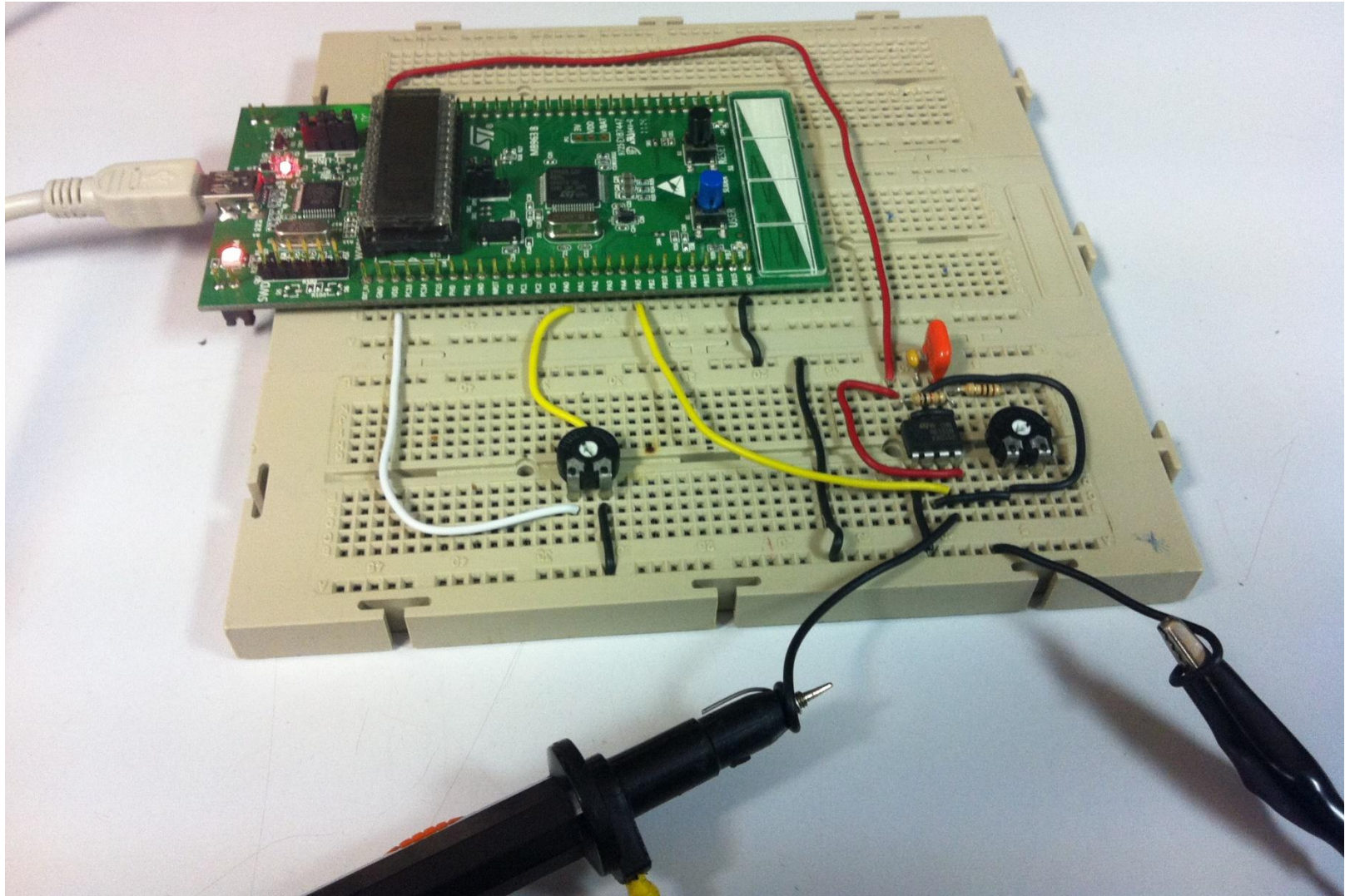
Recomendaciones para el uso de la placa

- Para poder utilizar mucho mejor la placa STM32L1-Discovery, es aconsejable:
 - Pincharla en una protoboard (o en un conjunto de ellas), de forma que los pines no se cortocircuiten y además dejen huecos para conectar cables
 - Meter el conjunto de la protoboard, la placa, así como las conexiones realizadas, en una caja, para su transporte sin que se suelten las conexiones
- En la siguiente transparencia se puede ver el detalle de conexión, así como un ejemplo de uso

Inserción de la placa en una protoboard



Ejemplo de Uso



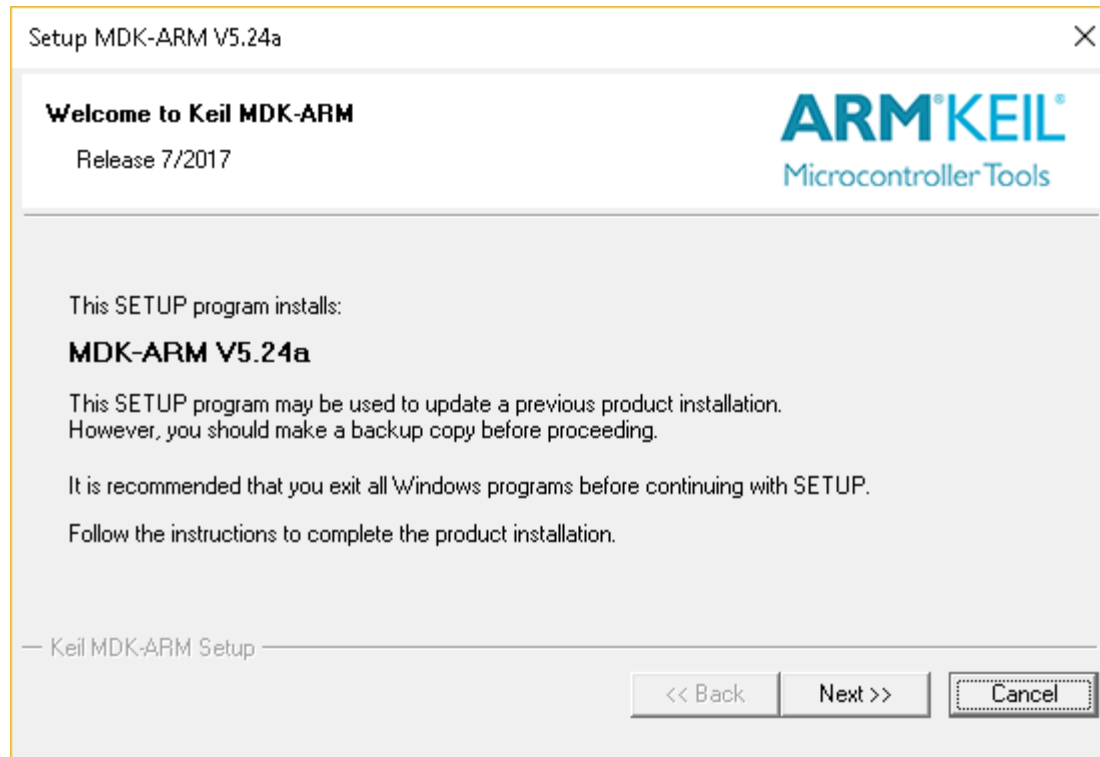
4 - Instalación de Keil uVision 5

Instalación

- El programa de instalación se puede descargar:
 - Directamente de la página de Keil (<https://www.keil.com/arm/demo/eval/arm.htm>) rellenando el formulario y descargando en MDK
 - Desde Aula Global , donde se encuentra el programa comprimido en varios ficheros .rar, que una vez descomprimidos, generan un fichero .exe que es el programa de instalación.
- Ejecutando el mdk526.exe (o equivalente, ya que los números pueden cambiar, al cambiar la versión) se arranca el instalador del μ Vision :
 - Es preferible instalar el programa con la placa ya conectada, para que así se instale el driver de la placa
 - Si no se hace así, se puede instalar posteriormente el driver
 - De a todas las pantallas sobre Next, introduciendo la información solicitada
 - Si no sabe que poner en alguna opción, déjela en su valor por defecto.
 - Al finalizar la instalación del μ Vision, se solicitará la instalación de los drivers tipo ULINK, ST-Link, etc.
 - Aceptar todo el proceso
- En algunos ordenadores o cuando se haya instalado el μ Vision sin tener la placa conectada, hay que instalar a mano el driver. Para ello:
 - Vaya a la página <http://www.st.com/internet/evalboard/product/251168.jsp> y pulse en la pestaña “Design Support”. (el fichero está también disponible en Aula Global)
 - Seleccione para descargar el “ST-LINK/V2 USB driver for Windows”
 - Descárguelo, descomprímalo y ejecútelo aceptando todas las opciones (incluso las preguntas de seguridad que impone el Windows)
 - Si sigue dando problemas, desenchufe el cable USB, espere unos segundos, y vuelva a enchufarlo.
 - Pruebe en distintos puertos USB
 - Si tenía una instalación anterior que no funcionaba, tendrá que ir, con la placa conectada, al Administrador de Dispositivos, y borrar el controlador que da fallos (el que salga con un símbolo de exclamación).

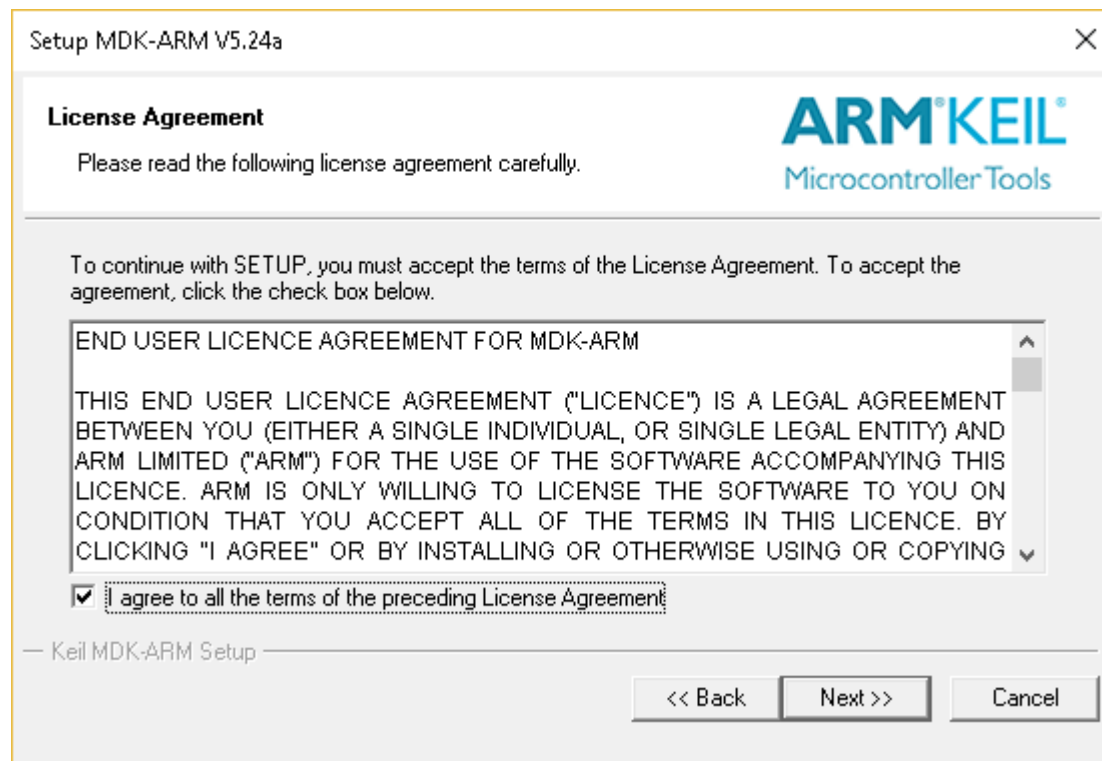
Instalación del Keil uVision 5

- Ejecuta el programa de instalación (mdk526.exe o similar).



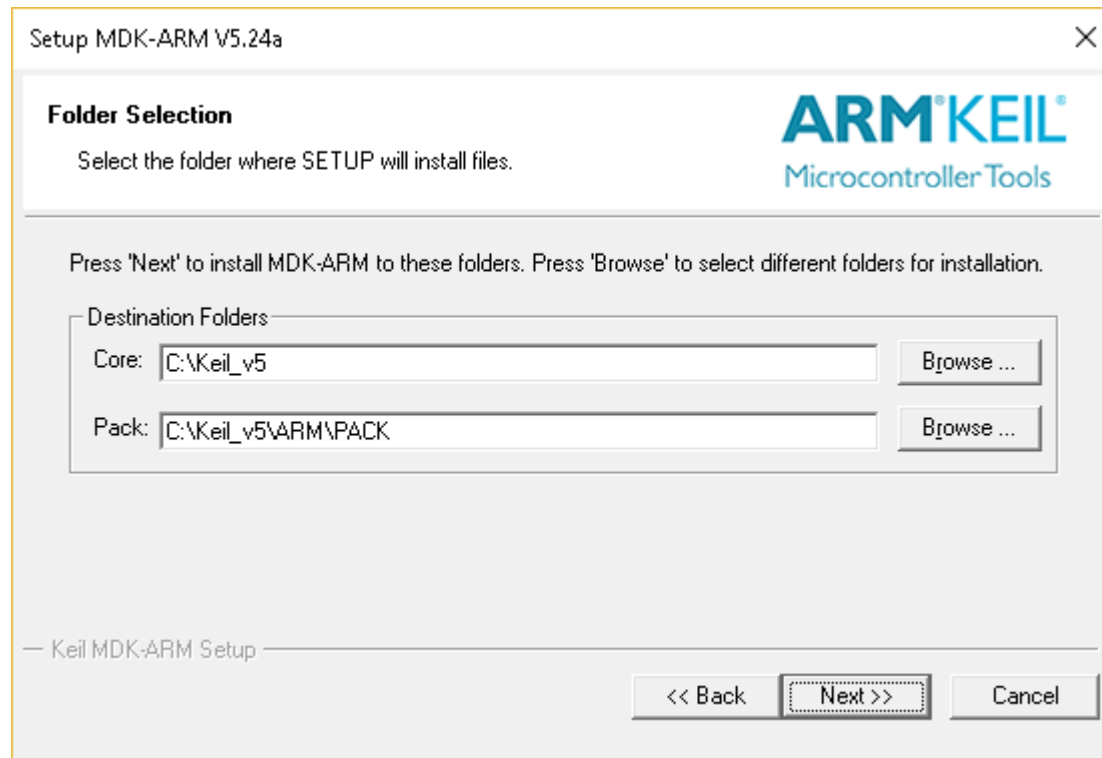
Instalación del Keil uVision 5

- Acepta las condiciones de la licencia.



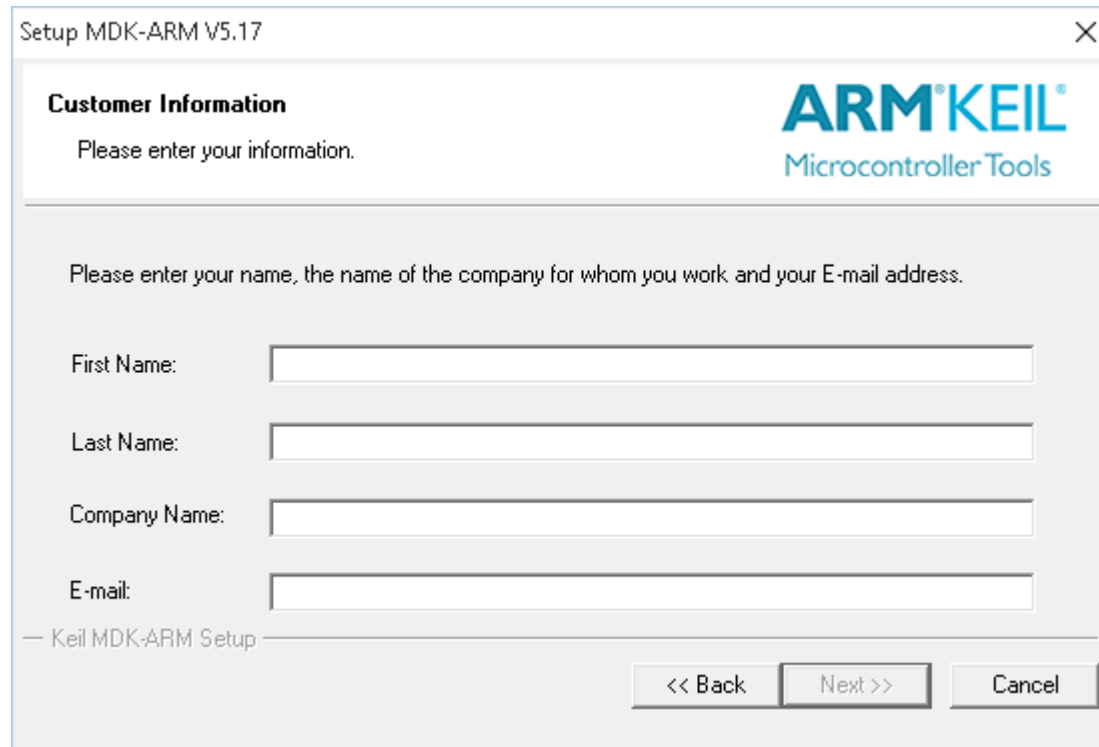
Instalación del Keil uVision 5

- Selecciona el lugar para la instalación o acepta la ubicación por defecto.



Instalación del Keil uVision 5

- Introduce tus datos (da igual lo que pongas, sobre todo en el e-mail).



Setup MDK-ARM V5.17

Customer Information

Please enter your information.

ARM KEIL
Microcontroller Tools

Please enter your name, the name of the company for whom you work and your E-mail address.

First Name:

Last Name:

Company Name:

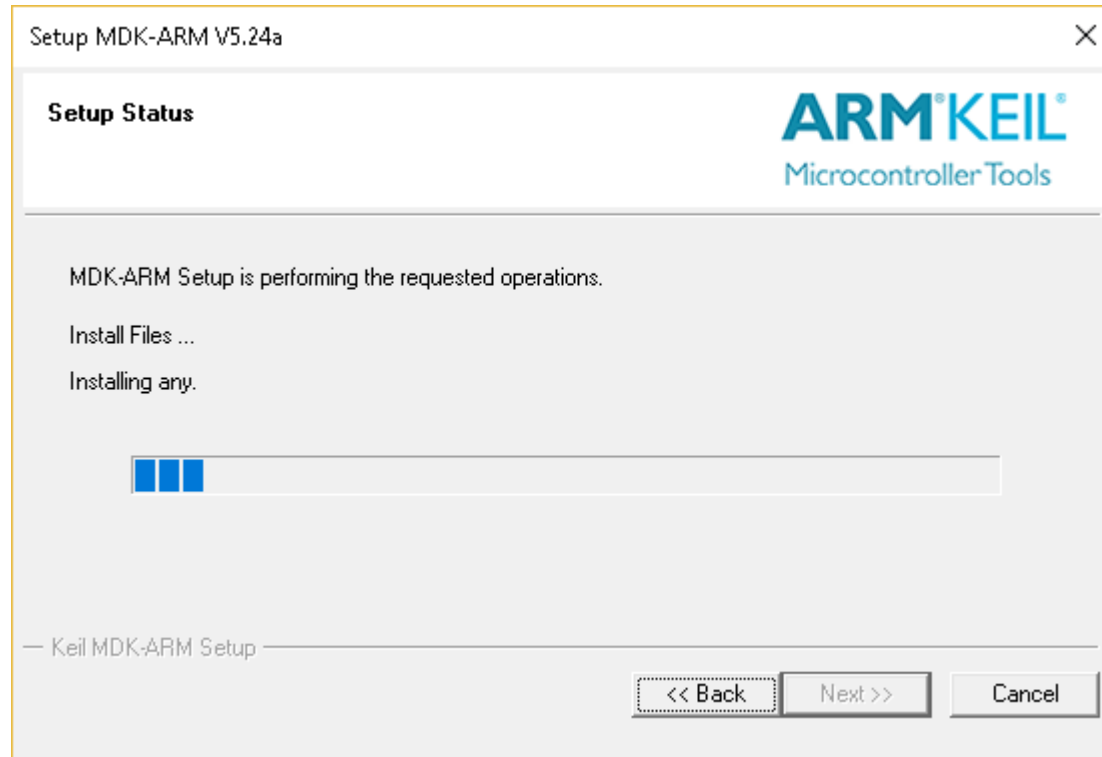
E-mail:

— Keil MDK-ARM Setup —

<< Back Next >> Cancel

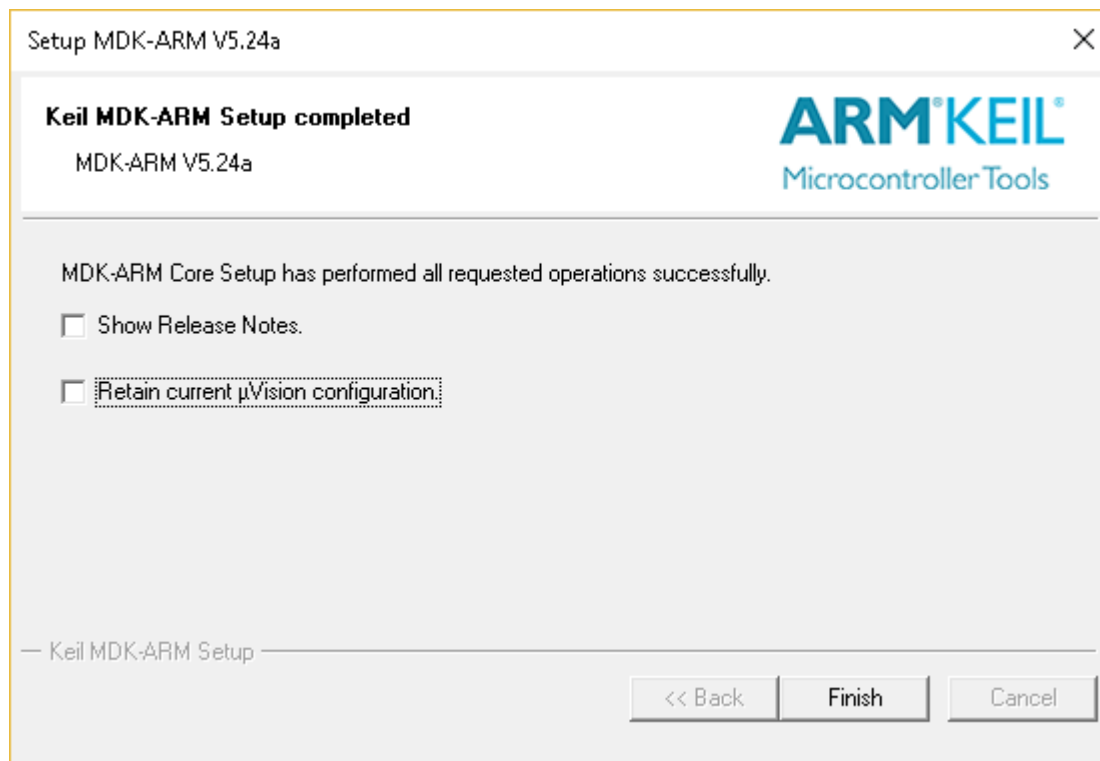
Instalación del Keil uVision 5

- Espera a que se instale la aplicación.



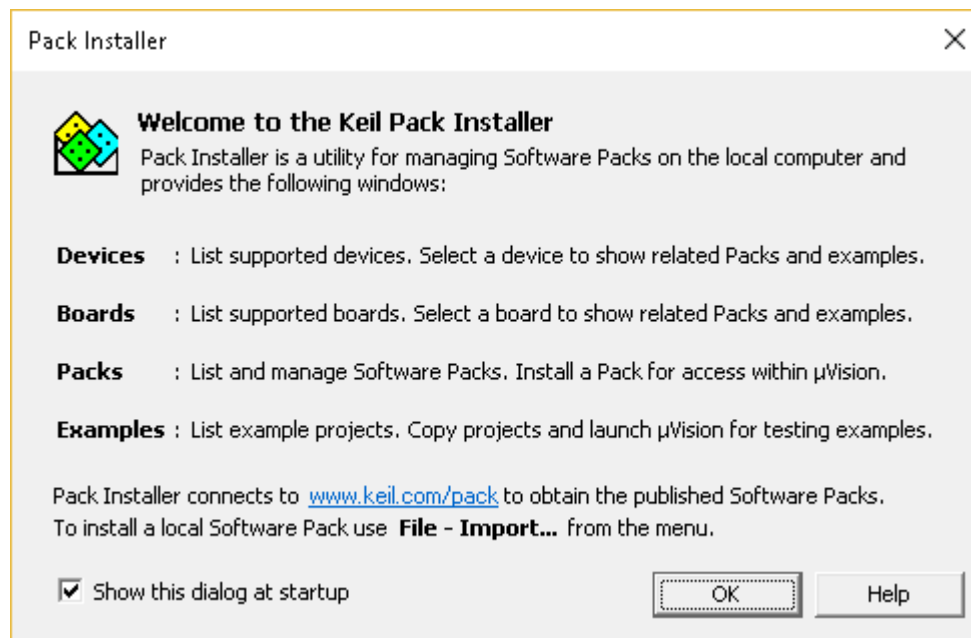
Instalación del Keil uVision 5

- Una vez finalizada la instalación puede solicitar ver las Release Notes y usar (o no) la configuración anterior (si se tenía ya una versión anterior instalada). Déjalo normalmente como se muestra en la imagen.



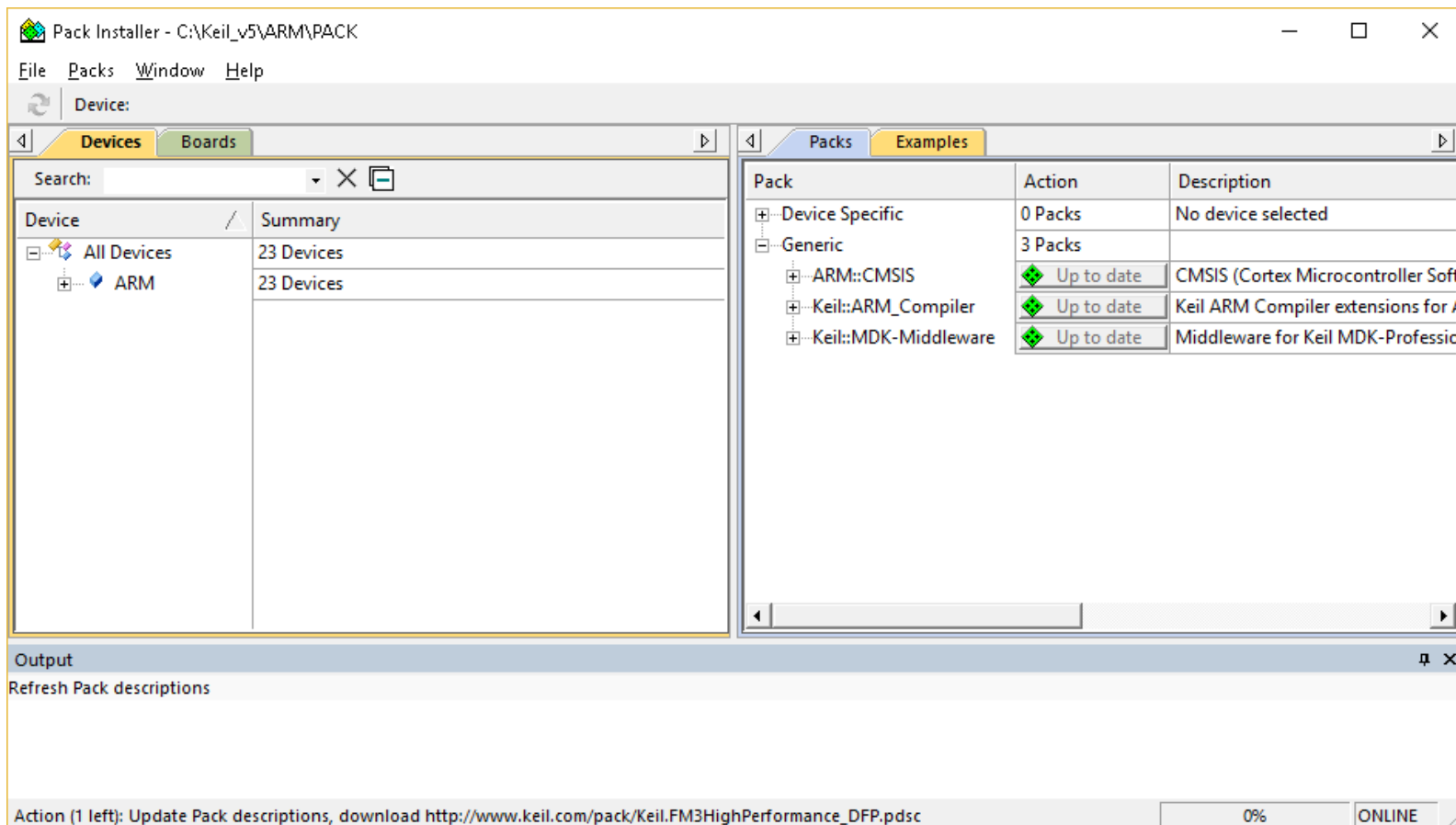
Instalación del Keil uVision 5

- Una vez instalado el uVision, se solicitará que instales los paquetes necesarios.
 - uVision es una plataforma de desarrollo que contempla multitud de procesadores, por lo que instalar todo puede ser excesivo.



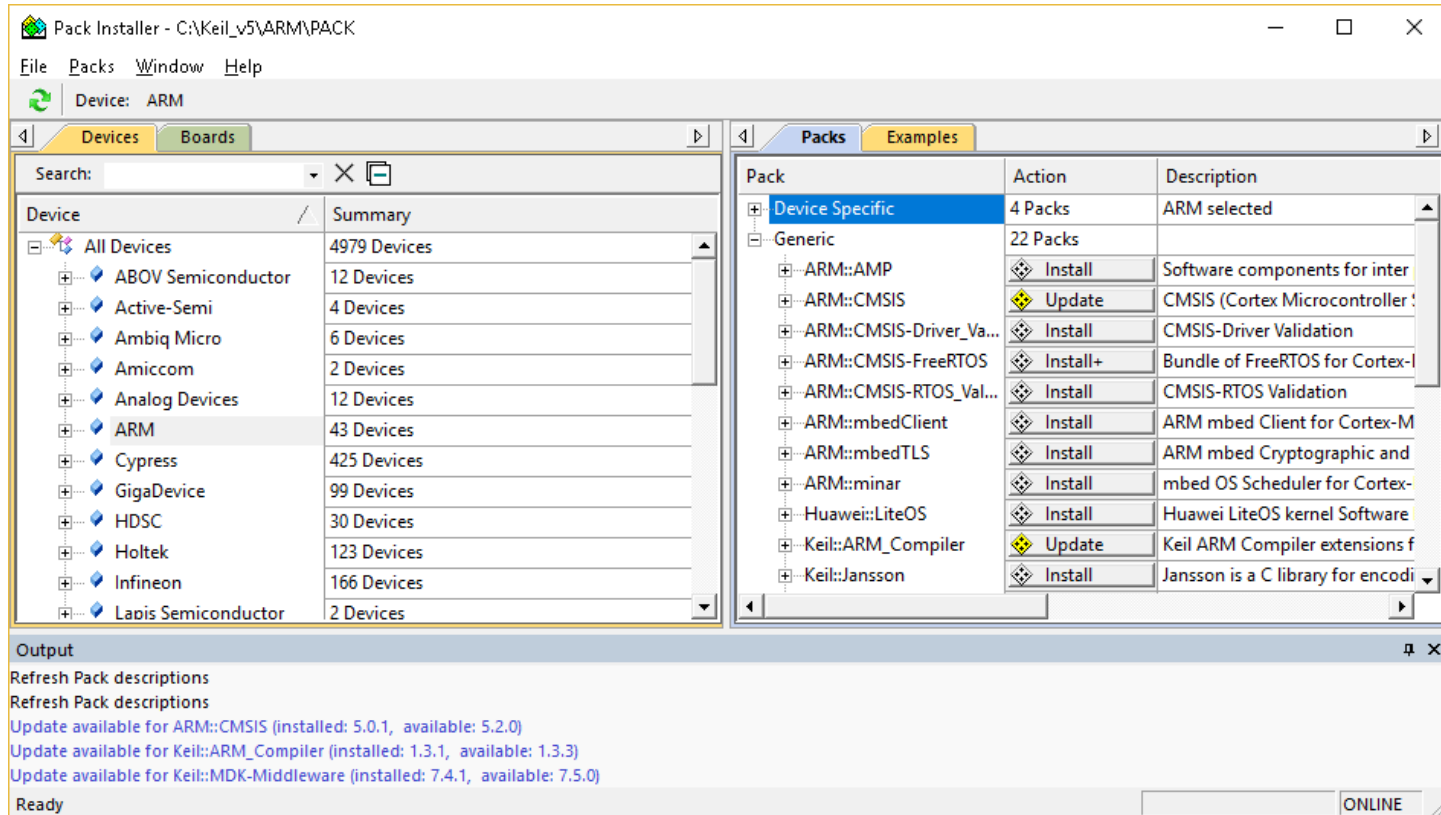
Instalación del Keil uVision 5

- Permite (bastante) tiempo para que la herramienta localice todos los módulos disponibles a través de la red.



Instalación del Keil uVision 5

- Espera a que todos los “software pack descriptions” se hayan instalado (tarda algo de tiempo).



Instalación del Keil uVision 5

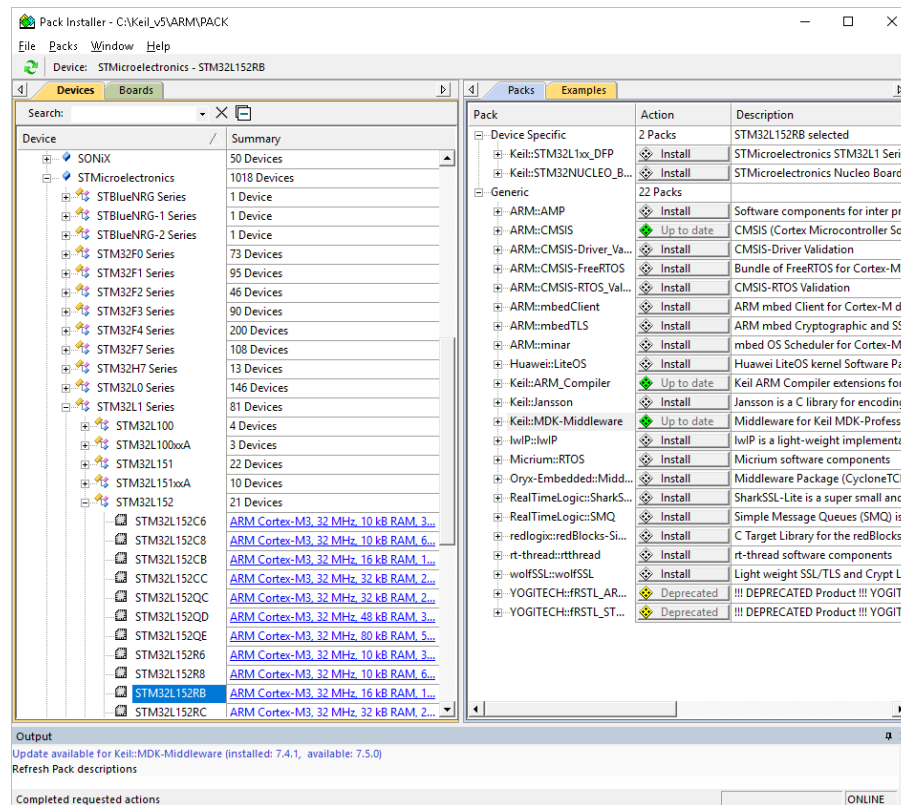
- En el árbol de la izquierda pulsa en la carpeta “ARM” y ahora en la derecha sobre “Update”, en donde te permita esa opción (símbolos en amarillo), hasta que se ponga en verde.

The screenshot shows the 'Pack Installer' window for the device 'STM32L152RB'. The left pane shows a tree view of device categories, with 'ARM' selected. The right pane shows a list of packs with their actions. The 'ARM::CMSIS' pack is highlighted, and its action is 'Update' (yellow icon). The bottom pane shows the output log with update notifications for 'ARM::CMSIS', 'Keil::ARM_Compiler', and 'Keil::MDK-Middleware'. The status bar at the bottom indicates the current action: 'Action (1 left): Install "ARM.CMSIS.5.2.0", download http://www.keil.com/pack/ARM.CMSIS.5.2.0.pack'.

Pack	Action	Description
Device Specific	2 Packs	STM32L152RB selected
Keil::STM32L1xx_DFP	Install	STMicroelectronics STM32L1 Se
Keil::STM32NUCLEO_B...	Install	STMicroelectronics Nucleo Boa
Generic	22 Packs	
ARM::AMP	Install	Software components for inter
ARM::CMSIS	Update	CMSIS (Cortex Microcontroller :
ARM::CMSIS-Driver_Va...	Install	CMSIS-Driver Validation
ARM::CMSIS-FreeRTOS	Install+	Bundle of FreeRTOS for Cortex-I
ARM::CMSIS-RTOS_Val...	Install	CMSIS-RTOS Validation
ARM::mbedClient	Install	ARM mbed Client for Cortex-M
ARM::mbedTLS	Install	ARM mbed Cryptographic and
ARM::minar	Install	mbed OS Scheduler for Cortex-
Huawei::LiteOS	Install	Huawei LiteOS kernel Software
Keil::ARM_Compiler	Update	Keil ARM Compiler extensions f
Keil::Jansson	Install	Jansson is a C library for encodi
Keil::MDK-Middleware	Update	Middleware for Keil MDK-Profe

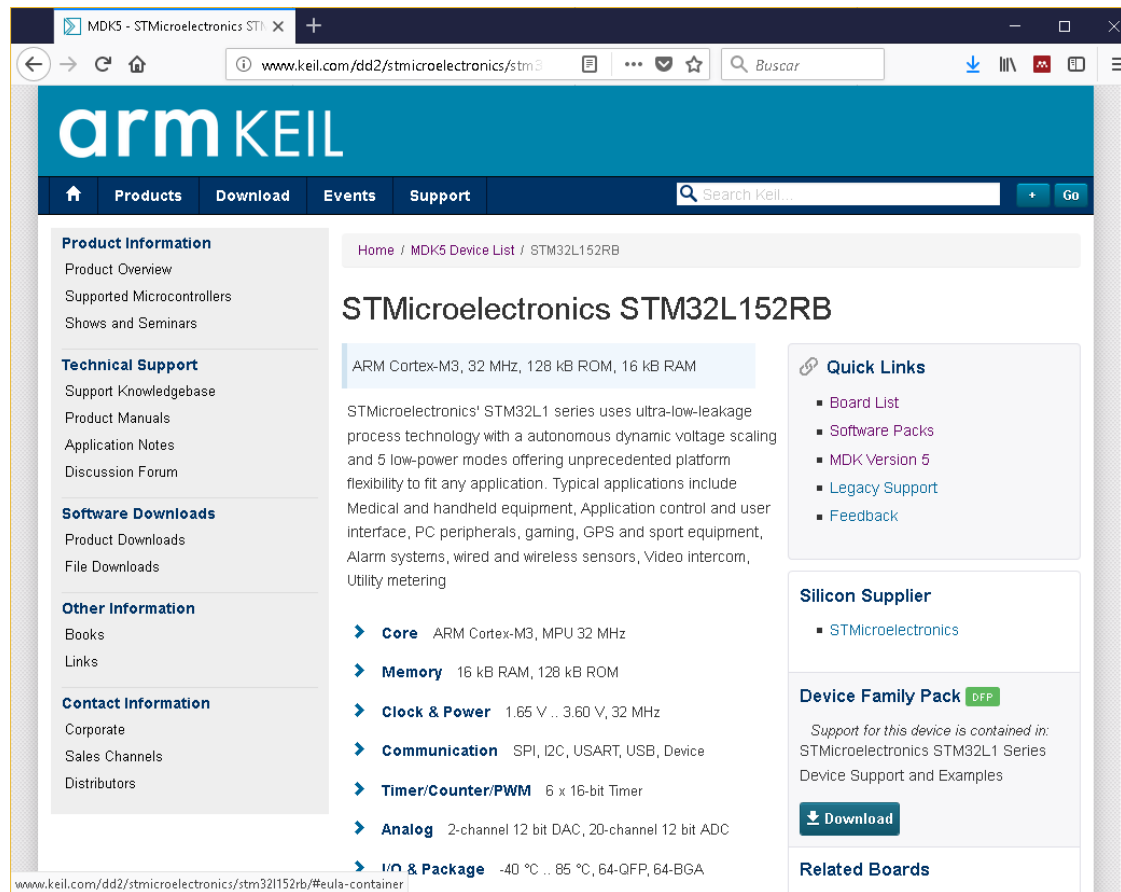
Instalación del Keil uVision 5

- Una vez finalizado, en el árbol de la izquierda, busca el “STM32L152RB” dentro de la carpeta “STMicroelectronics” y pincha en el enlace web que hay al lado.



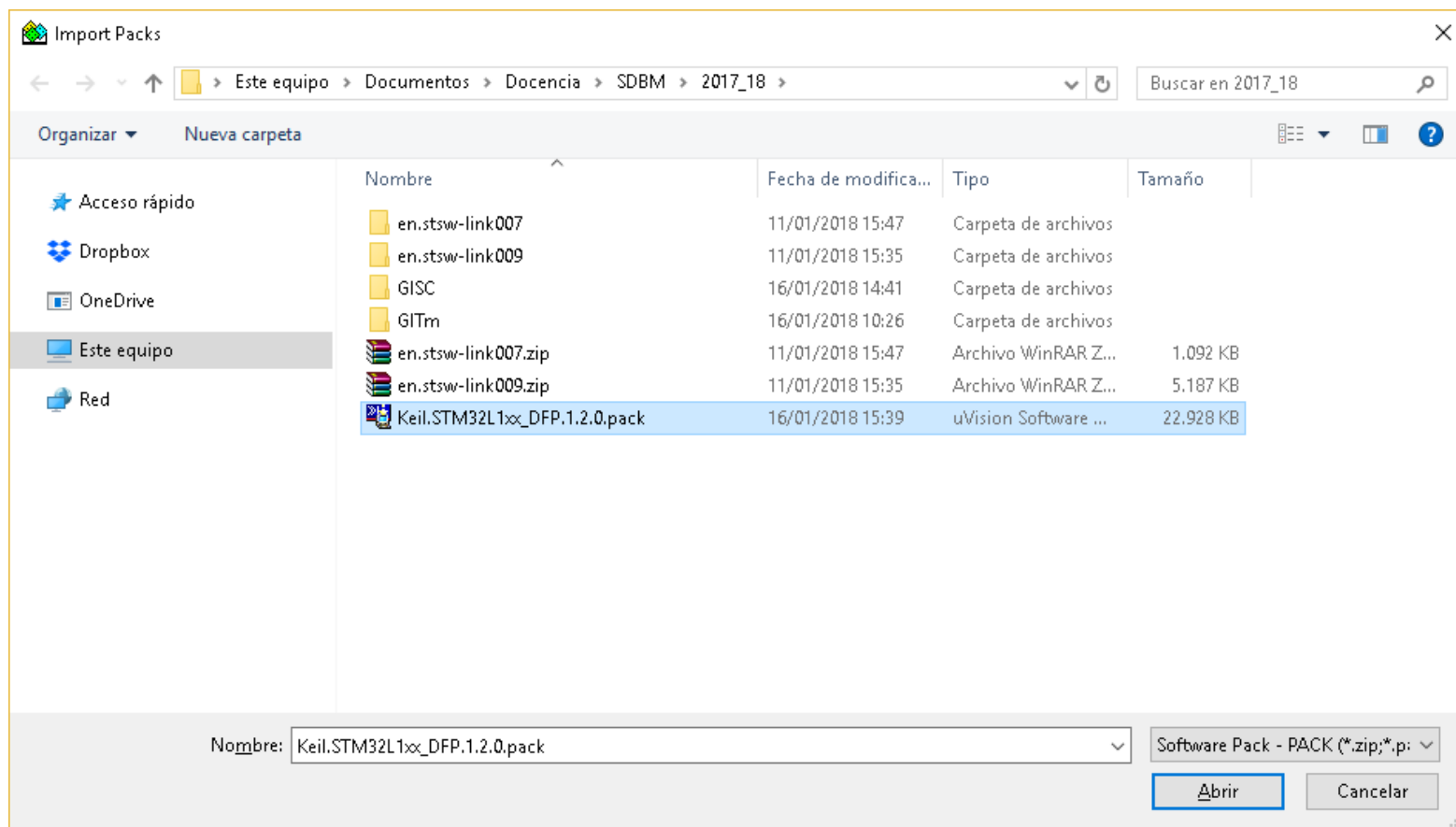
Instalación del Keil uVision 5

- En la página web, pincha en “**Download**” del Device Family Pack (o coge el fichero del cargado en Aula Global).



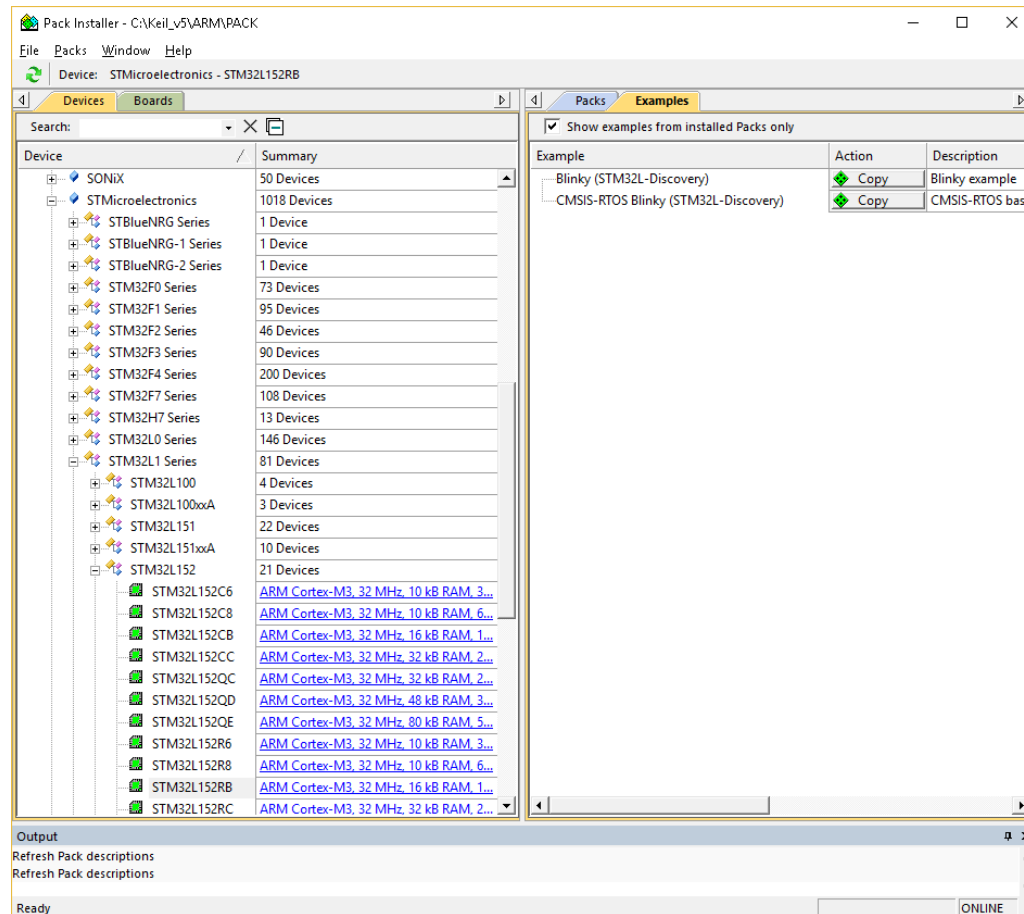
Instalación del Keil uVision 5

- Una vez descargado, importa el Paquete en el Pack Installer (opción “File -> Import”).



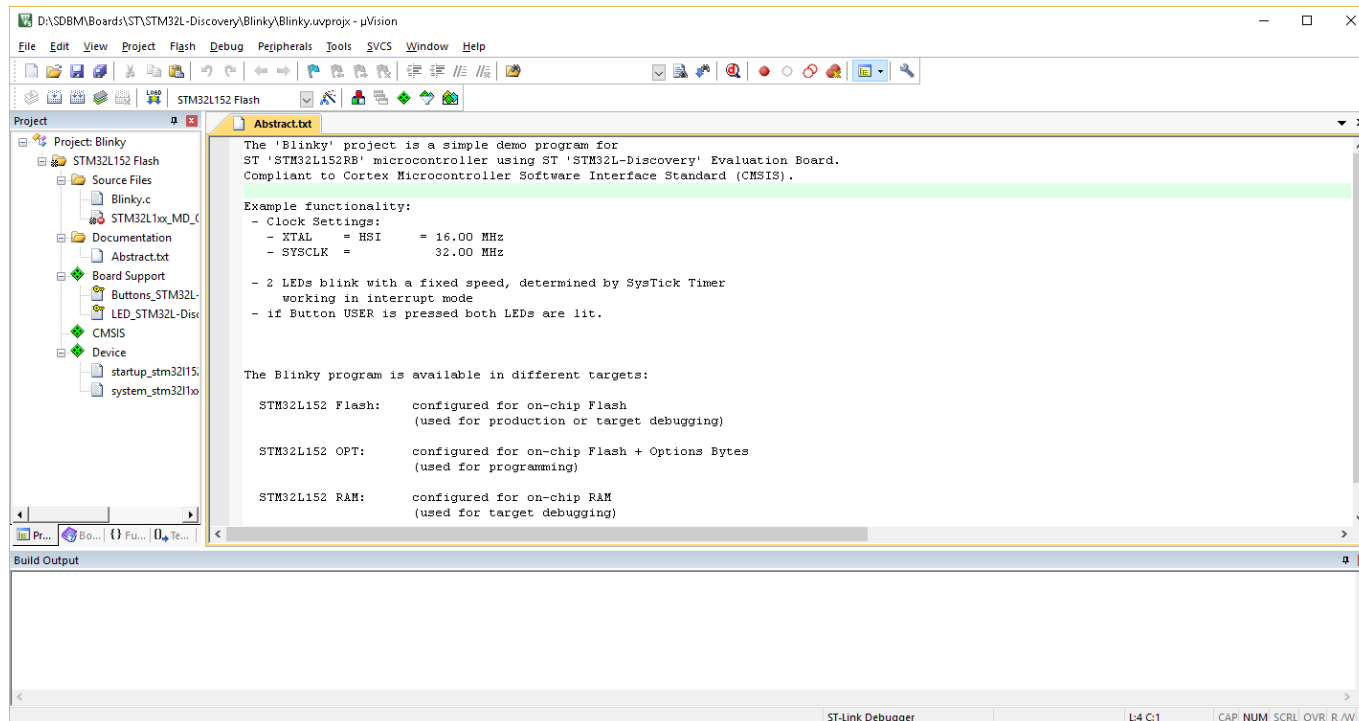
Instalación del Keil uVision 5

- Es recomendable copiar a una carpeta local el ejemplo Blinky (STM32L-Discovery), pulsando en el botón “Copy” en la pestaña de “Examples”.



Instalación del Keil uVision 5

- Una vez copiado, puedes probar el proyecto ejemplo en el entorno para comprobar que todo ha ido correctamente.
 - Se arranca el ejemplo Blinky en uVision, se compila y carga en la placa. Si todo va bien, al simular parpadean los LED azul y verde integrados en la placa. La forma de hacer todo esto se explica en las siguientes transparencias con la explicación del trabajo con el entorno.



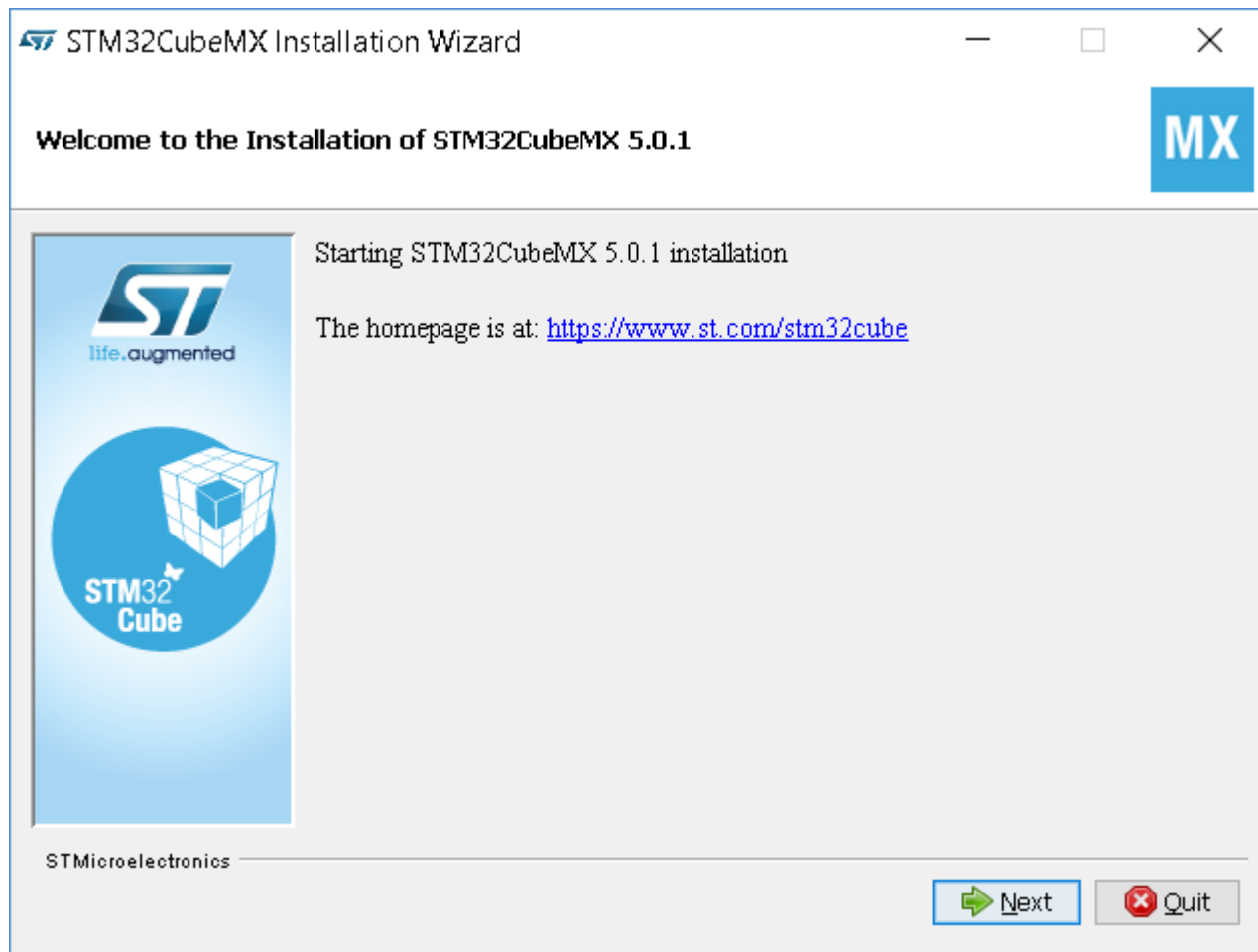
5 - Instalación de STM32 CubeMX

Instalación

- El programa de instalación se puede descargar:
 - Directamente de la página de ST (<https://www.st.com/en/development-tools/stm32cubemx.html>)
 - Desde Aula Global , donde se encuentra el programa comprimido en ficheros .rar, que una vez descomprimidos, generan un fichero .exe que es el programa de instalación.
- Ejecutando el fichero SetupSTM32CubeMX-5.0.1.exe (o equivalente, ya que los números pueden cambiar, al cambiar la versión) se arranca el instalador:
 - En algunas versiones de Windows, no reconoce el autor de este programa e intenta impedir su instalación
 - Hay que darle a Más Información y luego elegir la opción “Ejecutar de todas formas”

Instalación del STM32 CubeMX

- Ejecuta el programa de instalación (SetupSTM32CubeMX-5.0.1.exe o similar).



Instalación del STM32 CubeMX

- Acepta las condiciones.



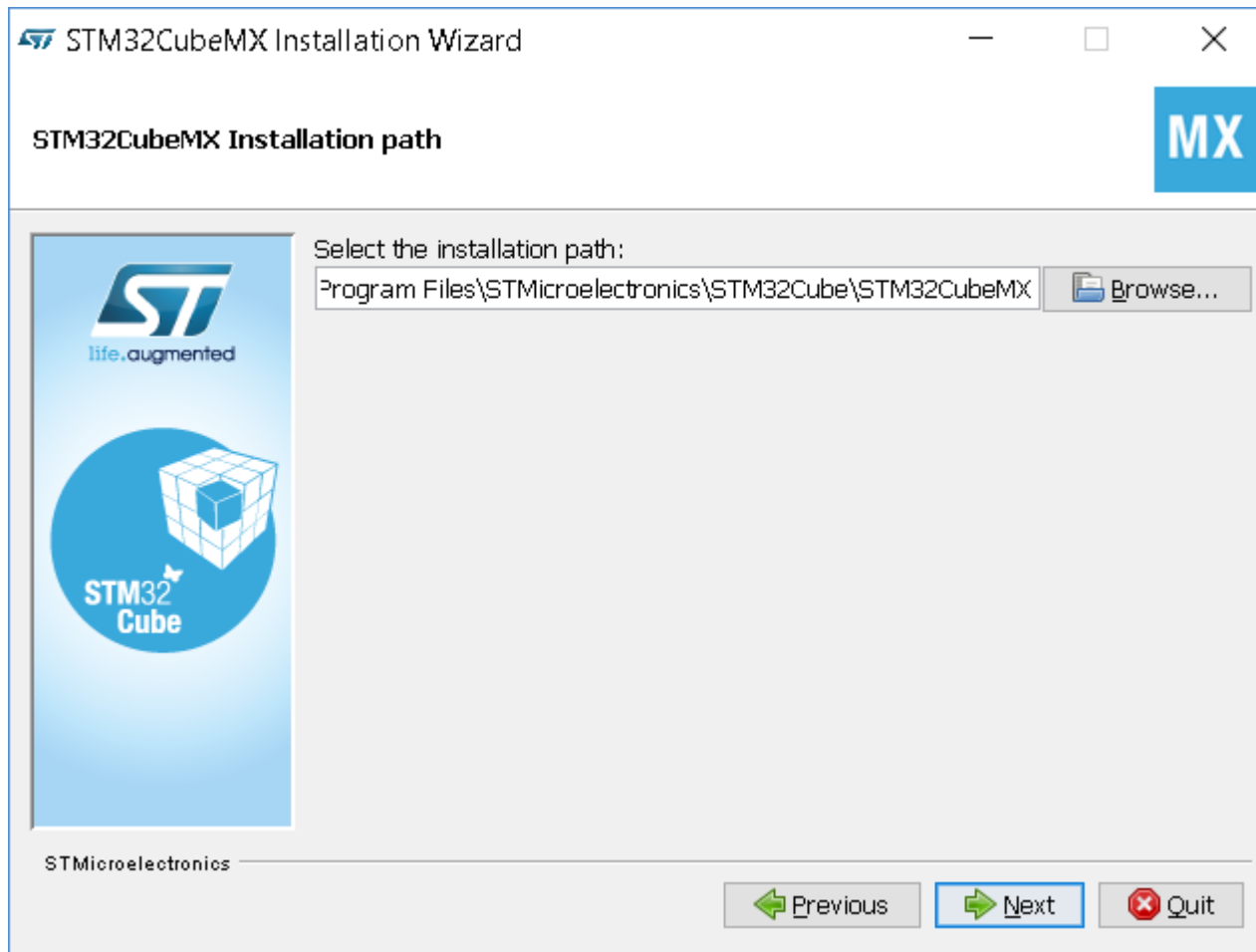
Instalación del STM32 CubeMX

- Acepta la política de privacidad.
- Si lo deseas, puede aceptar que ST pueda usar información sobre cómo se usa la herramienta, para su mejora.



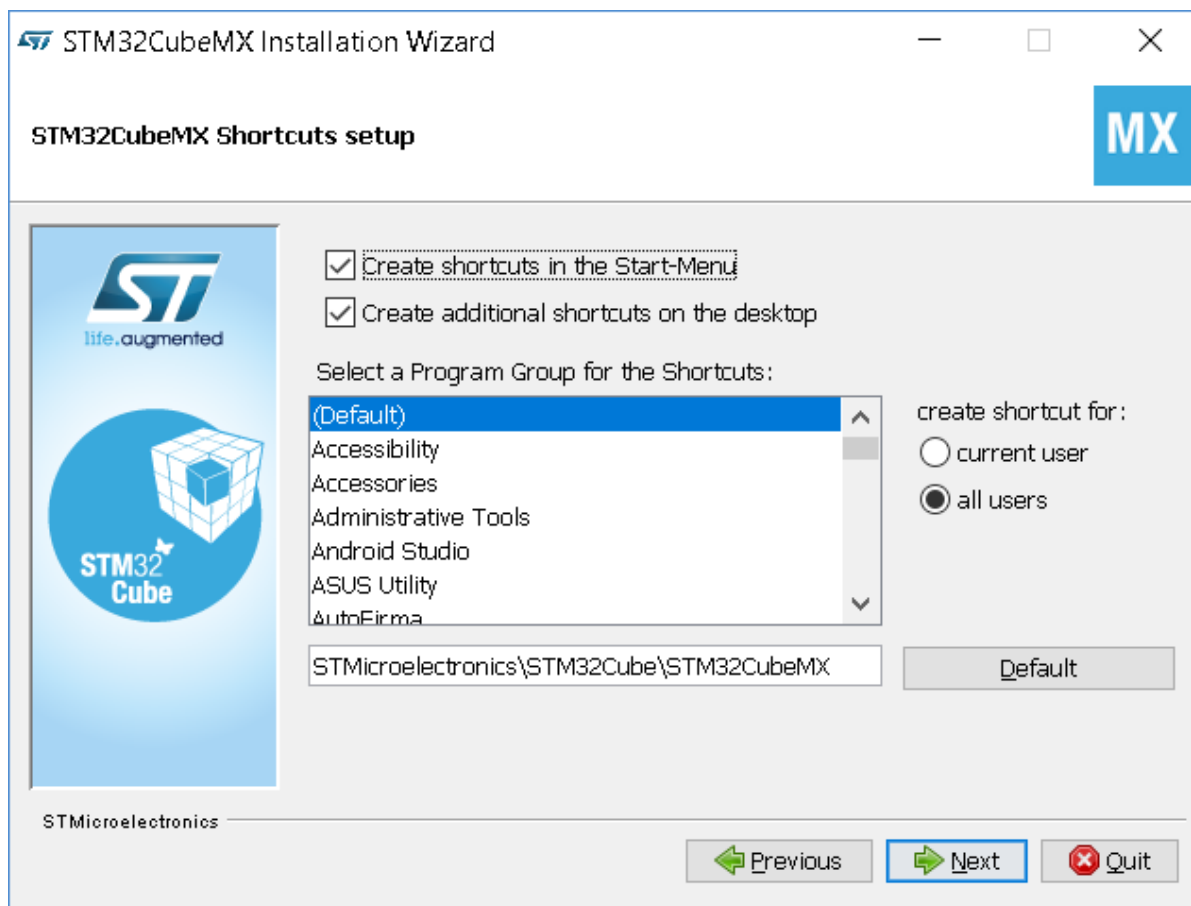
Instalación del STM32 CubeMX

- Elije la ruta de instalación o deja la que hay por defecto.



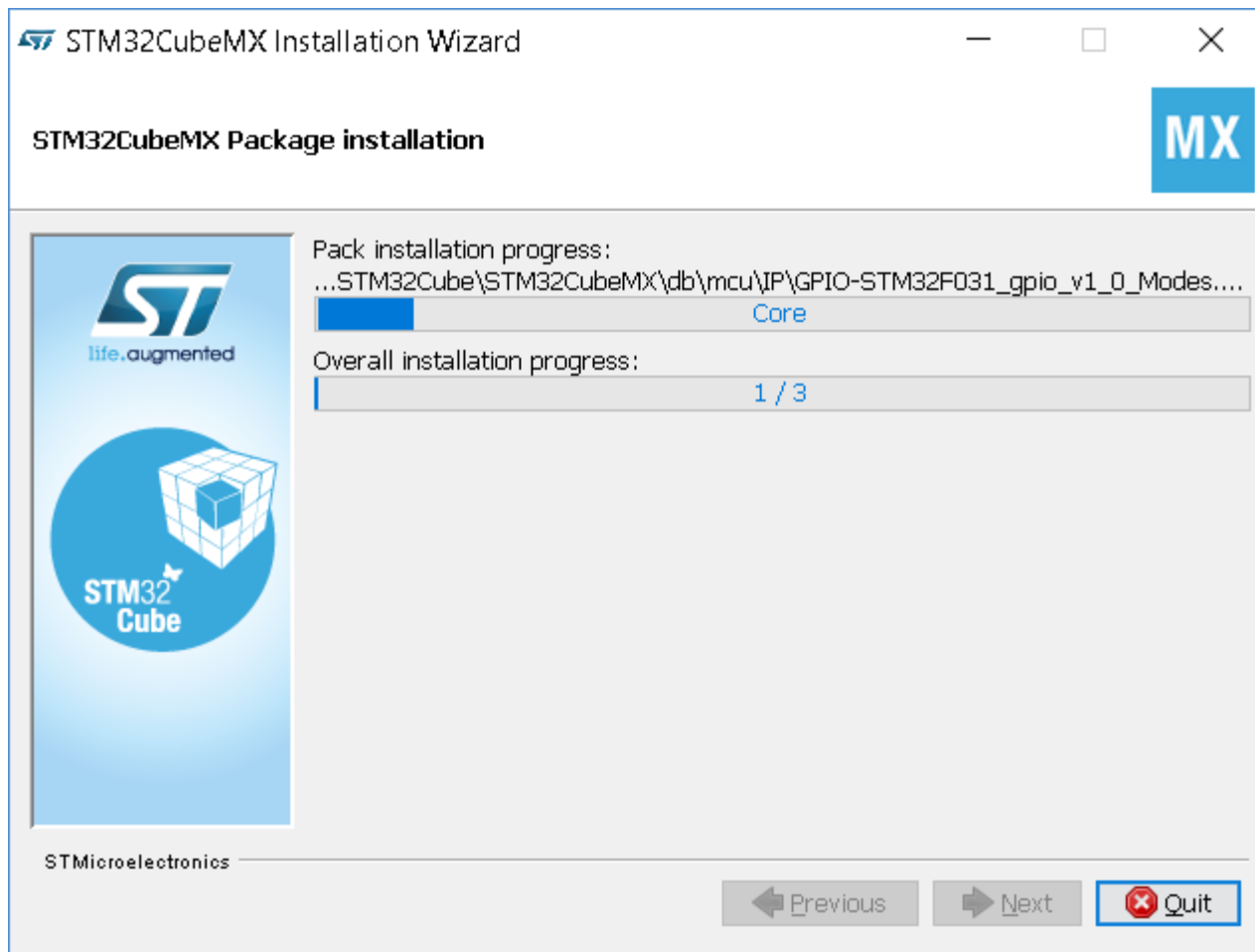
Instalación del STM32 CubeMX

- Configura los detalles de cómo se integrará en el entorno Windows. Deja normalmente lo que ves en la imagen.



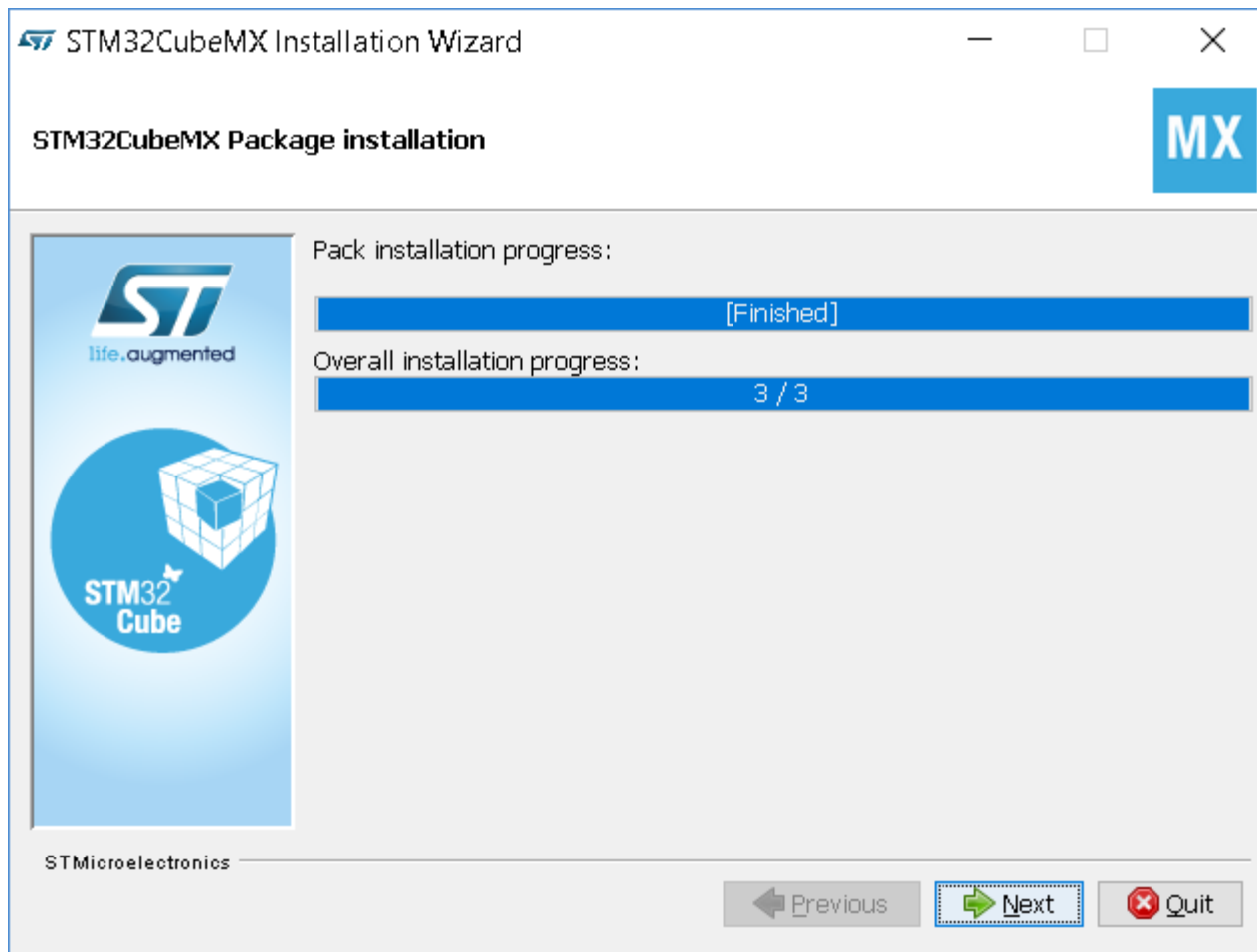
Instalación del STM32 CubeMX

- Espera a que acabe la instalación.



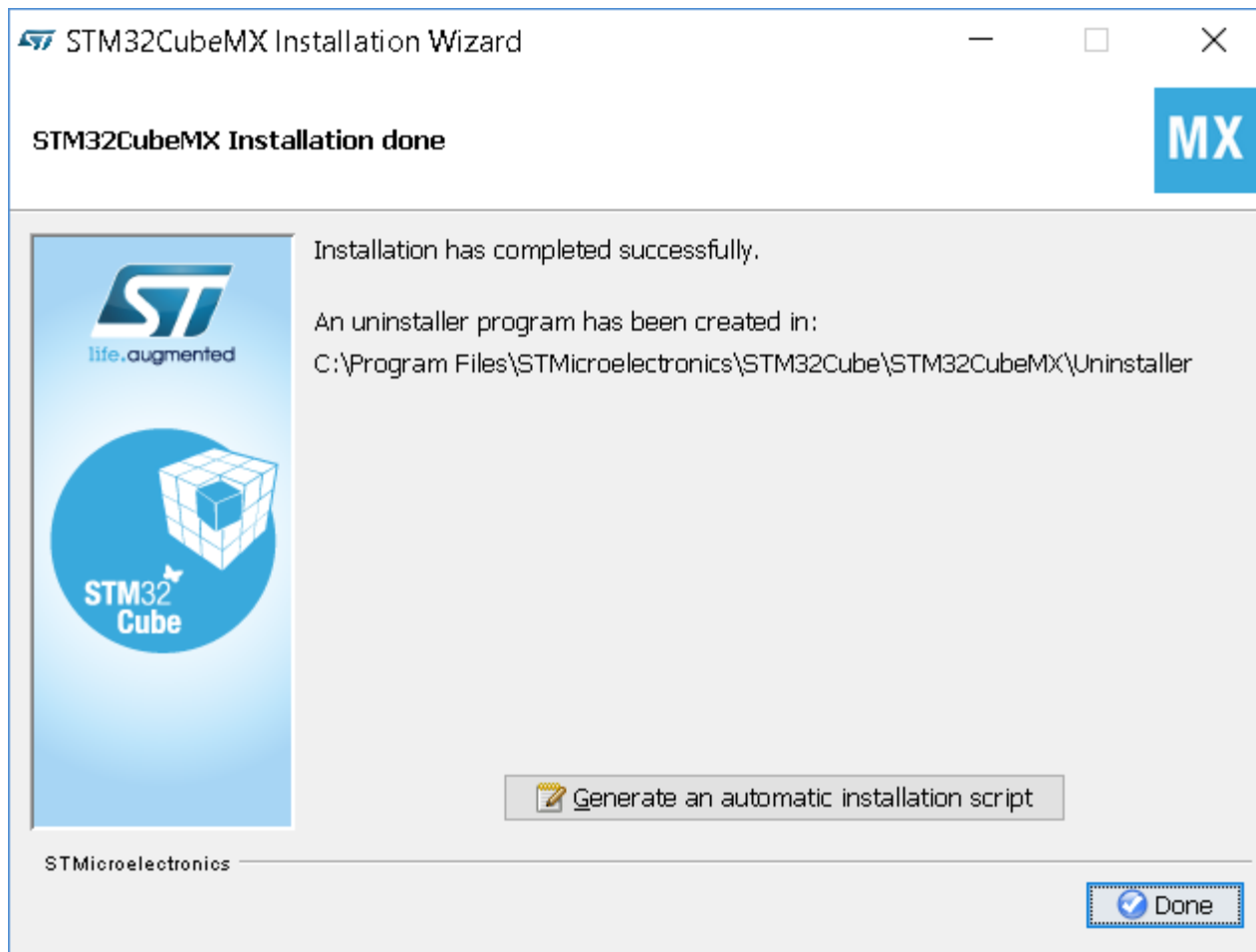
Instalación del STM32 CubeMX

- Una vez finalizada, pulsa en “Next”.



Instalación del STM32 CubeMX

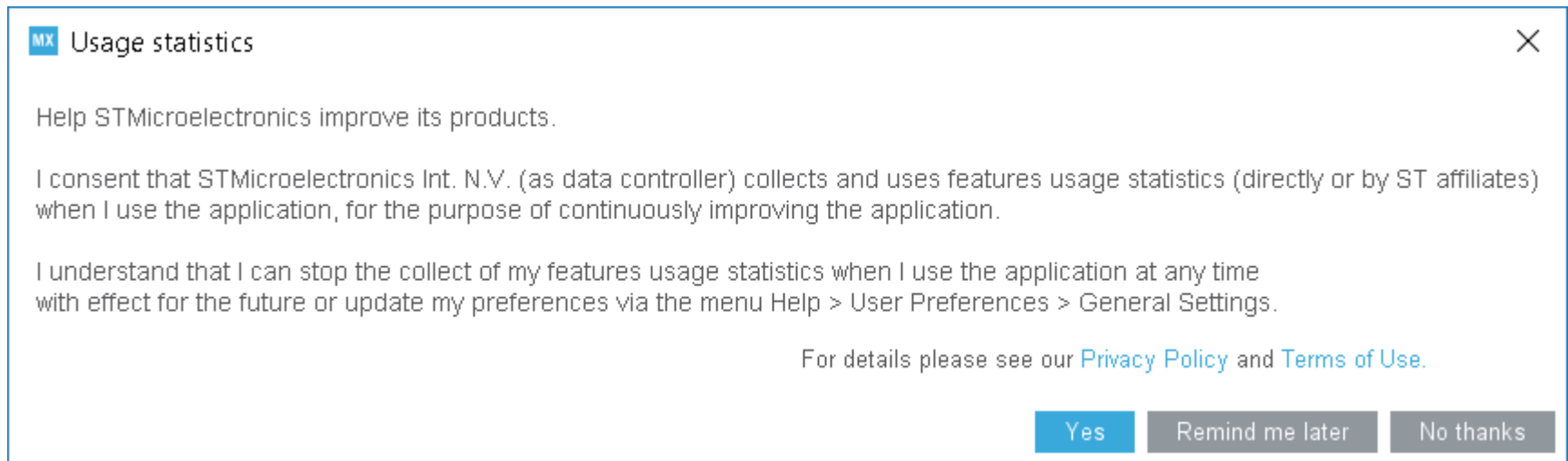
- Finalmente pulsa en “Done”.



6 - Primera Ejecución del STM32 CubeMX

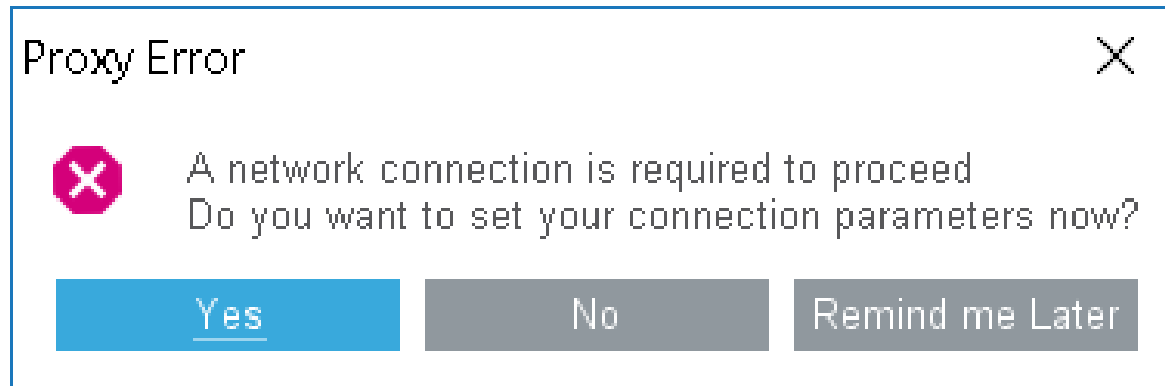
Primera Ejecución del STM32 CubeMX

- Para finalizar la instalación, es necesario que se inicie la aplicación una primera vez para que se actualicen sus componentes y sus bases de datos.
- Al arrancar la primera vez, preguntará para que decidas si quieres o no que ST use estadísticas de uso de la aplicación
 - Elige la opción que deseas.



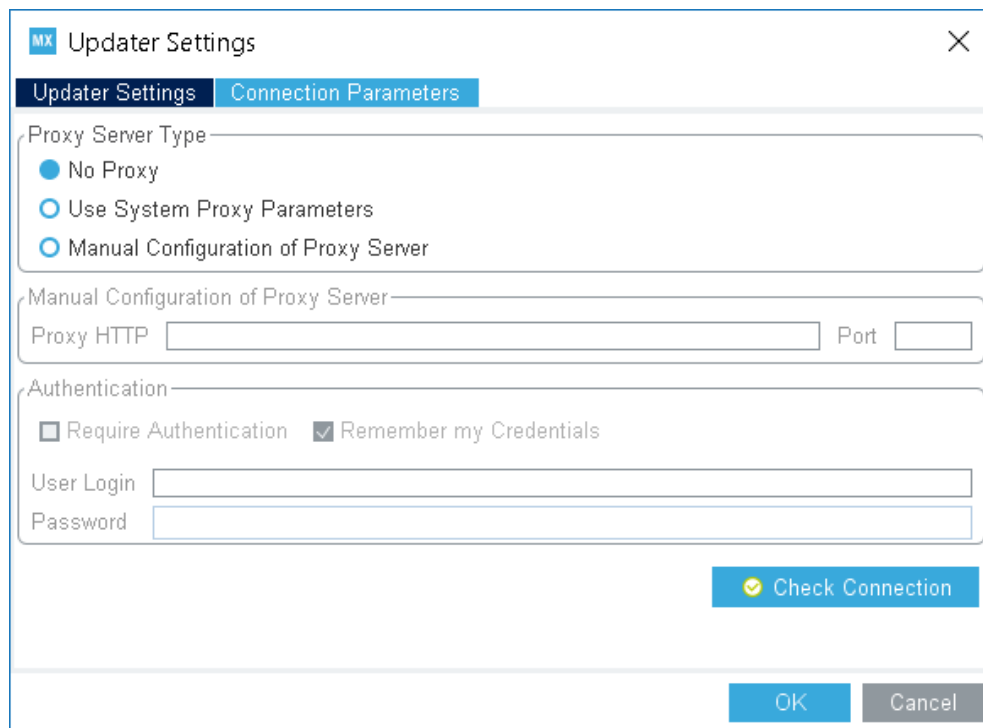
Primera Ejecución del STM32 CubeMX

- Permite la conexión a red para, por lo menos, cargar actualizaciones.



Primera Ejecución del STM32 CubeMX

- Acepta la configuración de red o modifícala atendiendo a tus necesidades. Normalmente hay que dejar lo que ves en la imagen.
- Puedes pulsar sobre el botón “Check” para comprobar que tienes conexión de Internet con la configuración elegida.



Primera Ejecución del STM32 CubeMX

- En la pantalla principal, selecciona “Check for Updates”.

The screenshot shows the STM32CubeMX software interface. The title bar reads "STM32CubeMX Untitled". The menu bar includes "File", "Window", and "Help". The main content area is divided into three sections:

- Existing Projects:** Contains an "Open Existing Projects" button with a folder icon.
- New Project:** A dark blue box with the heading "I need to :". It contains two options:
 - "Start My project from MCU ..." with an "ACCESS TO MCU SELECTOR" button.
 - "Start My project from STBoar..." with an "ACCESS TO BOARD SELECTOR" button.
- Manage software installations:** Contains two sections:
 - "Check for STM32CubeMX and embedded software packag..." with a "CHECK FOR UPDATES" button.
 - "Install or remove embedded software packages" with an "INSTALL / REMOVE" button.

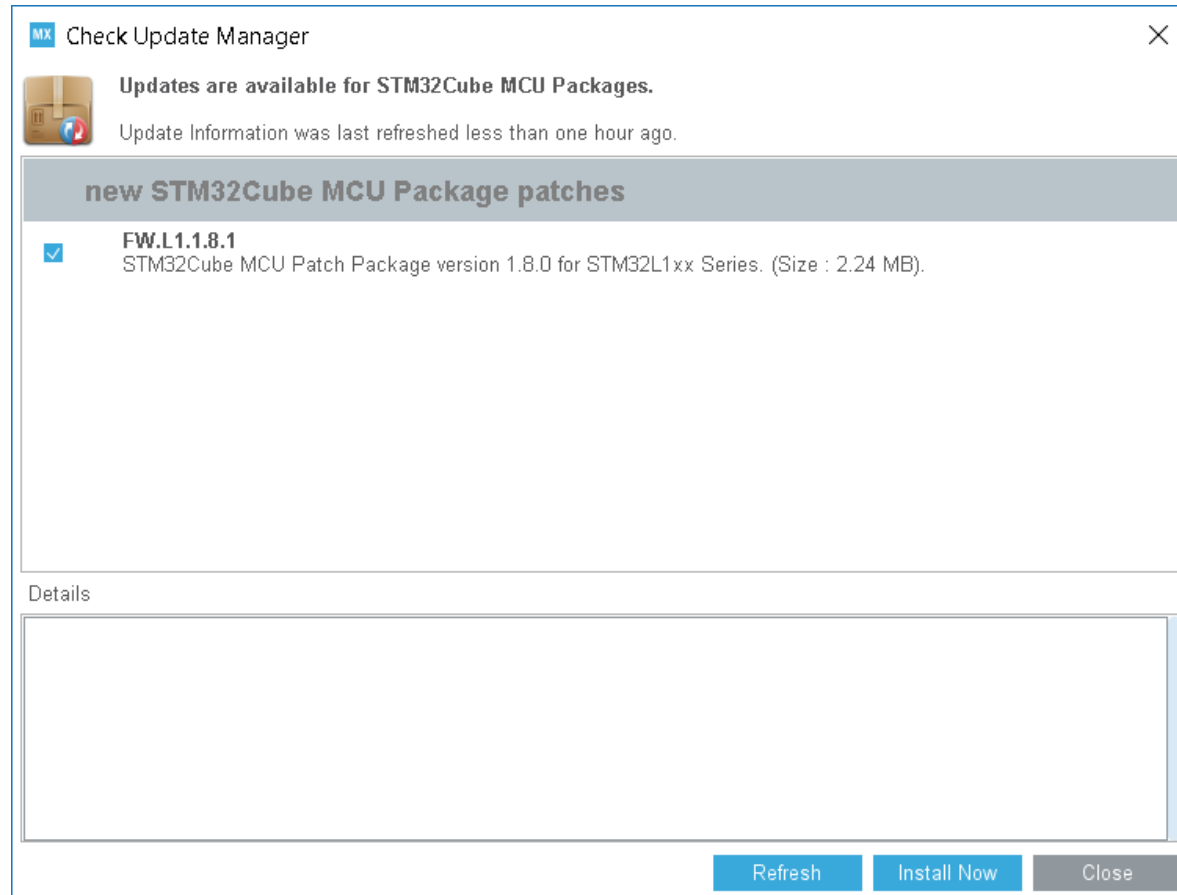
At the bottom right, there is a promotional banner for "New STM32G0 series MCUs" featuring an image of the chip and the following characteristics:

- Efficient
- Robust
- Simple

The ST logo is visible in the bottom right corner of the banner.

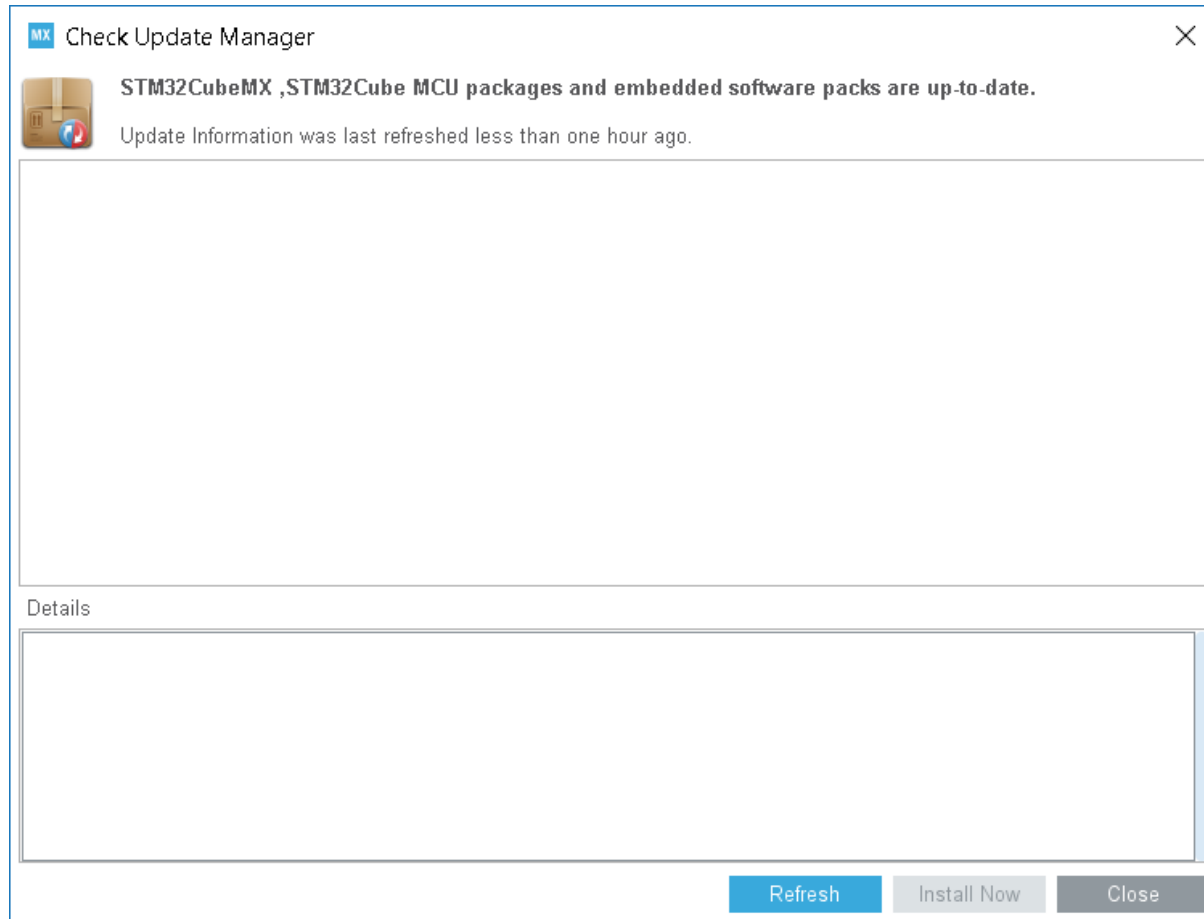
Primera Ejecución del STM32 CubeMX

- Pulsa sobre “Refresh”. Si tiene actualizaciones pendientes se mostrarán en la ventana. Si es así, pulsa sobre “Install Now” para instalarlas.



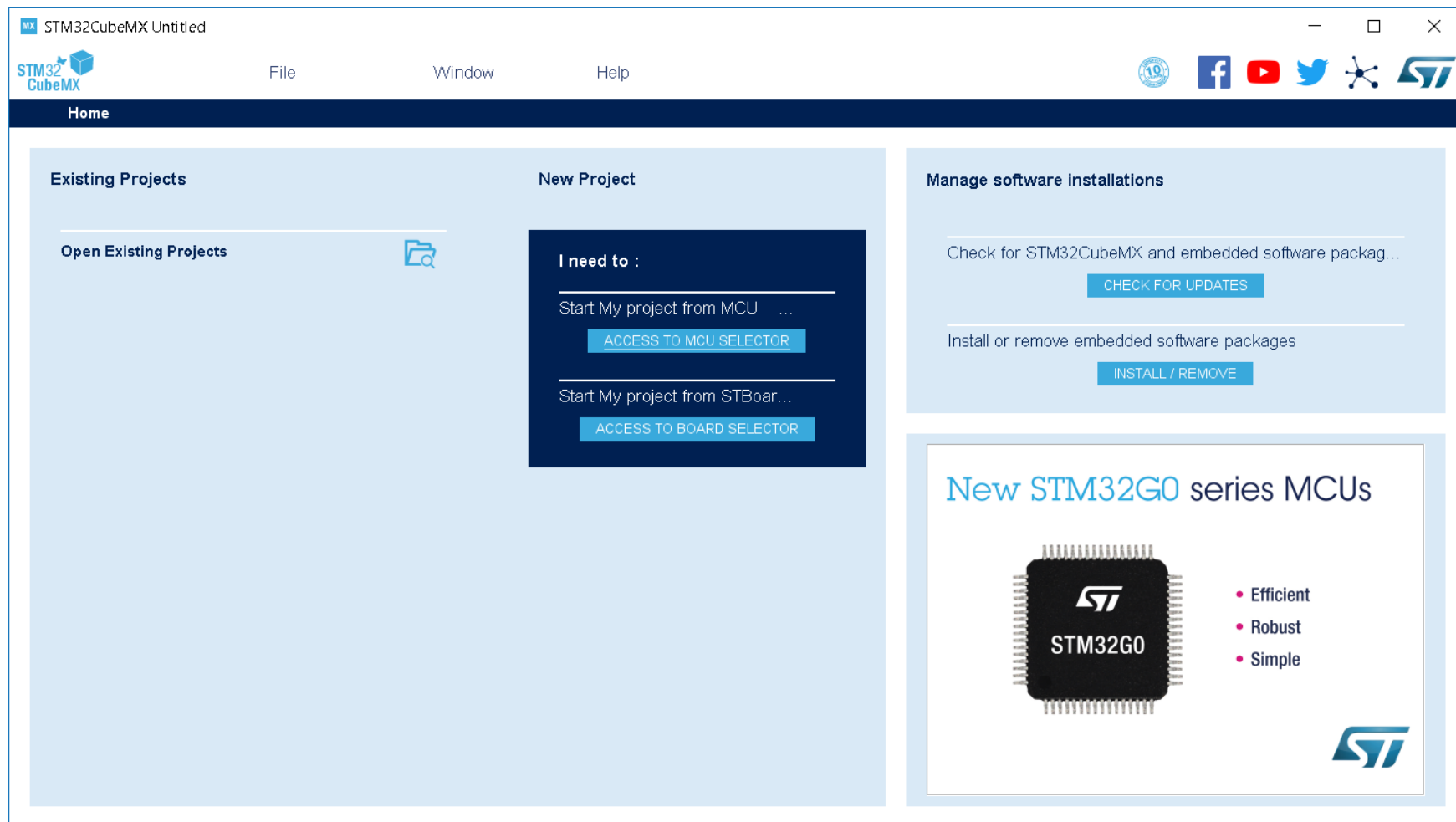
Primera Ejecución del STM32 CubeMX

- Cuando ya no queden actualizaciones por cargar y actualizar, o no haya salido ninguna desde el principio, pulsa sobre “Close”.



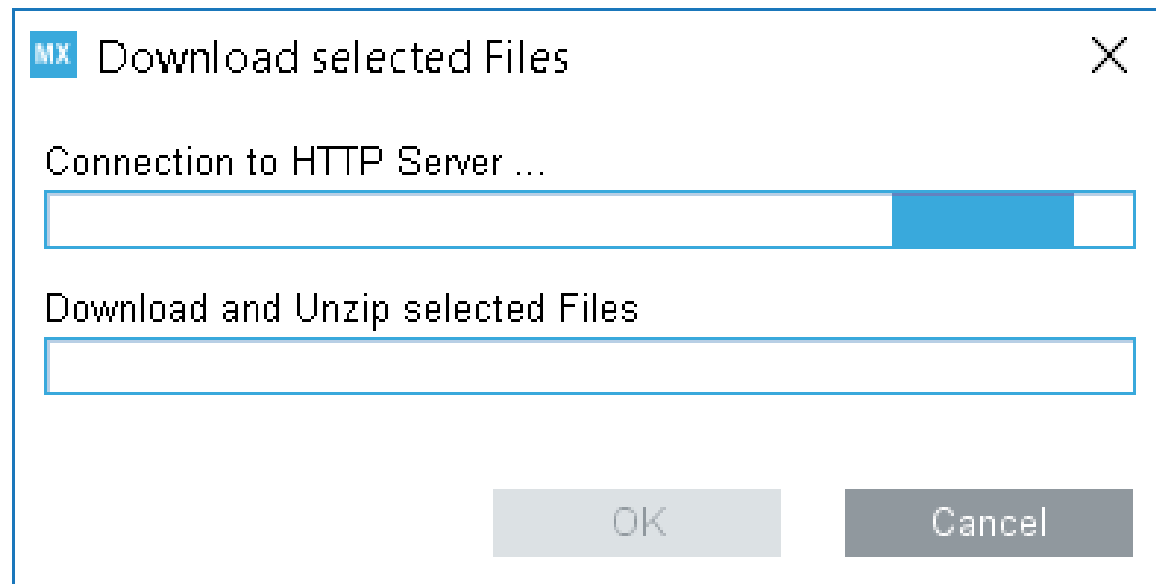
Primera Ejecución del STM32 CubeMX

- Selecciona ahora “Start my Project from STBoards” o “...from MCU”.



Primera Ejecución del STM32 CubeMX

- Espera a que cargue el listado entero de dispositivos y/o placas de Desarrollo.



Primera Ejecución del STM32 CubeMX

- Una vez descargado el listado aparecerá la siguiente pantalla:

The screenshot shows the STM32CubeMX interface for selecting a board. The main display area features the text "New STM32G0 series MCUs" and an image of the STM32G0 chip. To the right of the chip image, the following features are listed:

- Efficient
- Robust
- Simple

Below the main display, there is a "Boards List: 101 items" section with a table of boards. The table has the following columns: Overview, Part No., Type, Marketing Status, Unit Price (US\$), and Mounted Device.

Overview	Part No.	Type	Marketing Status	Unit Price (US\$)	Mounted Device
	32F0308DISCOVERY	Discovery	Active	8.9	STM32F0308T:x
	32F072BDISCOVERY	Discovery	Active	10.4	STM32F072RBT:x
	32F3348DISCOVERY	Discovery	Active	17.0	STM32F334C8T:x

Primera Ejecución del STM32 CubeMX

- Selecciona en el menú de filtrado de la izquierda lo siguiente: Vendor = STMicroelectronics, Type = Discovery, MCU Series = STM32L1.

The screenshot shows the 'New Project from a Board' interface in STM32CubeMX. The left sidebar contains filters for Board Filters, Part Number Search, Vendor (STMicroelectronics), Type (Discovery), and MCU Series (STM32L1). The main area displays 'New STM32G0 series MCUs' with a chip image and features: Efficient, Robust, Simple. Below is a 'Boards List' table with 3 items.

	Overview	Part No.	Type	Marketing Status	Unit Price (US\$)	Mounted Device
☆		32L100CDISCOVERY	Discovery	Active	7.9	STM32L100RCTx
☆		32L152CDISCOVERY	Discovery	Active	12.4	STM32L152RCTx
☆		STM32L-DISCOVERY	Discovery		0.0	STM32L152RBTx

Primera Ejecución del STM32 CubeMX

- Del listado de placas resultante, pulsa sobre la última “STM32L-Discovery”, y a continuación pulsa sobre el botón “Start Project”.

The screenshot shows the 'New Project from a Board' wizard in STM32CubeMX. The 'Board Selector' tab is active, displaying a list of boards on the left and a detailed view of the selected 'STM32L-Discovery' board on the right.

Board Selector (Left Panel):

- Type: Check/Uncheck All
 - Discovery
 - Evaluation Board
 - Nucleo64
- MCU Series: Check/Uncheck All
 - STM32F0
 - STM32F1
 - STM32F3
 - STM32F4
 - STM32F7
 - STM32G0
 - STM32L0
 - STM32L1
 - STM32L4
 - STM32L4+
- Other: Price (0.0 to 12.4), Oscillator Freq. = 8 (MHz)
- Peripheral: Accelerometer (0/0), Analog I/O (0/0), Arduino Form Factor (0/0), Audio Line In (0/0), Audio Line Out (0/0), Button (0/2), CAN (0/0)

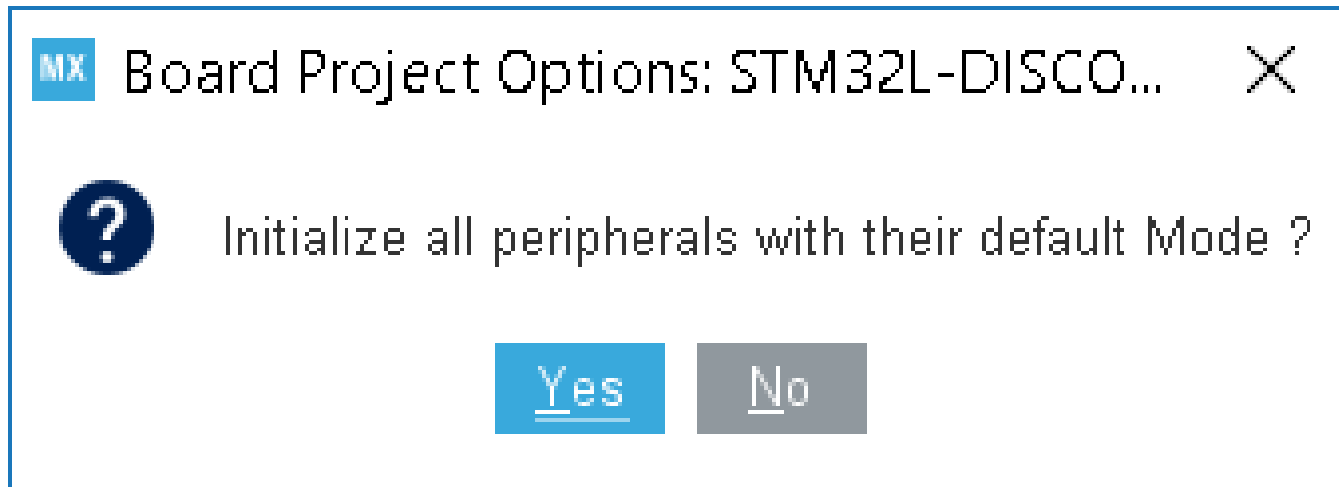
Board Details (Right Panel):

- Features: Large Picture, Docs & Resources, Datasheet, Buy, Start Project
- STM32L-Discovery: STM32 L1
- STMMicroelectronics STM32L152C Discovery Board Support and Examples
- Unit Price (US\$): 0.0
- Mounted device: [STM32L152RBTx](#)
- Features:
 - On-board ST-LINKV2
 - Supply through ST-Link USB
 - External Supply: 3V and 5V
 - JP1 (Idd) for current measurement
 - One LCD 24 segments, 4 commons on DIP28 package
 - One Linear Touch Sensor or four Touch Keys
 - Two Push-buttons: User and Reset
 - Four LEDs: USB COM, 3.3 V Power, user (Green/Blue)
- Boards List: 3 items

	Overview	Part No.	Type	Marketing Status	Unit Price (US\$)	Mounted Device
☆		32L100DISCOVERY	Discovery	Active	7.9	STM32L100RCTX
☆		32L152DISCOVERY	Discovery	Active	12.4	STM32L152RCTX
☆		STM32L-DISCOVERY	Discovery		0.0	STM32L152RBTx

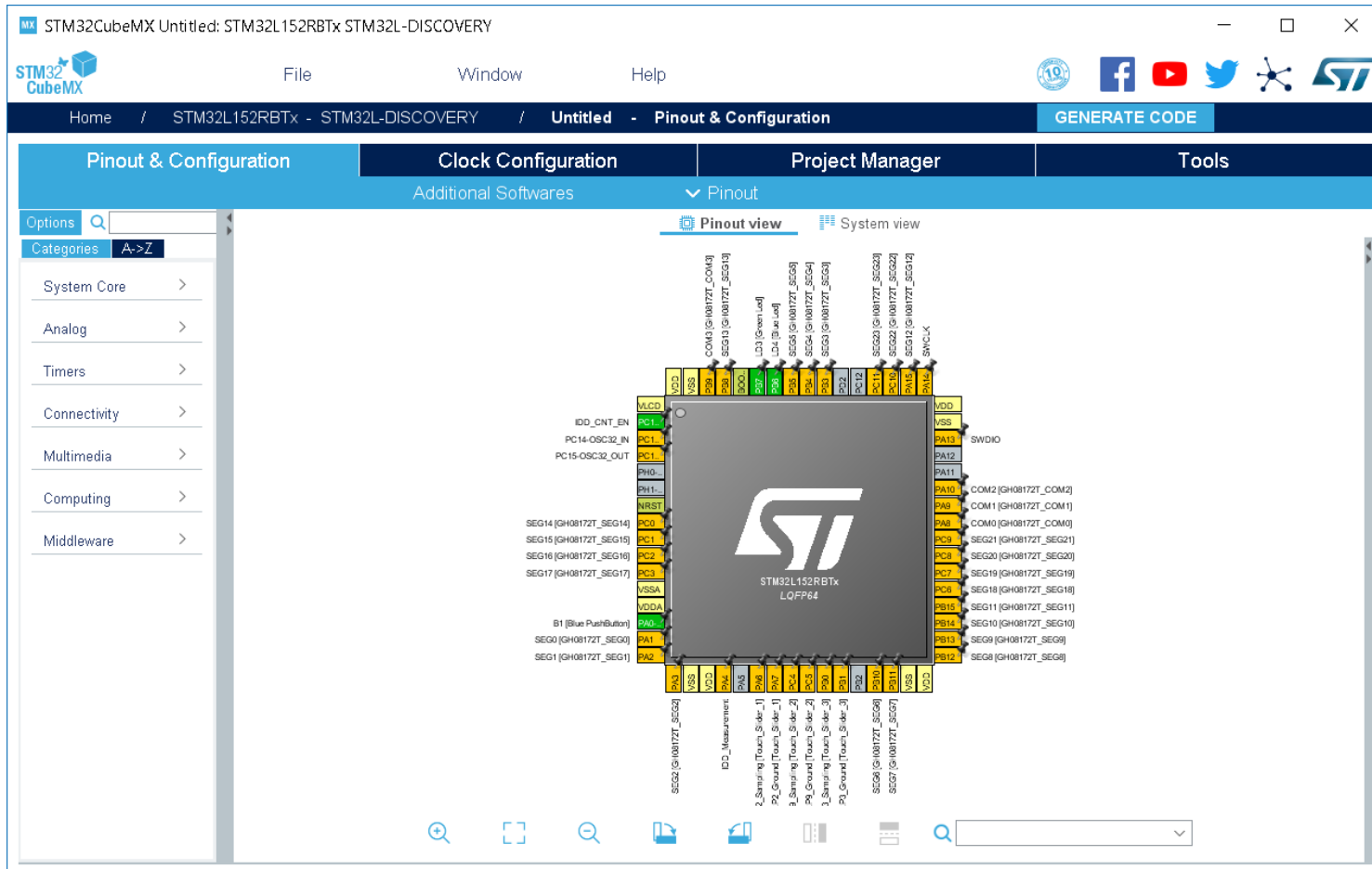
Primera Ejecución del STM32 CubeMX

- Por motivos docentes, selecciona “Yes” a la pregunta sobre “Inicializar los periféricos en el Modo por defecto”



Primera Ejecución del STM32 CubeMX

- Finalmente aparecerá la siguiente pantalla, en la cual ya se puede empezar a trabajar. De momento lo dejamos aquí.



Trabajando con el STM32 CubeMX

- Una vez que hayas llegado a ver la pantalla anterior, ya se puede empezar a configurar las funcionalidades del microcontrolador y de la placa que le interesen para el proyecto.
- Antes de hacer esto, tendrás que haber determinado previamente:
 - Qué periféricos vas a utilizar
 - Qué pines vas a usar y con qué funcionalidad
 - Qué configuración vas a querer utilizar para cada uno de esos pines y periféricos
- Los pasos a trabajar con el Cube MX son los siguientes:
 1. Configurar los pines y periféricos
 2. Seleccionar el reloj del microcontrolador
 3. Gestionar el proyecto
 4. Pulsar “GENERATE CODE” y una vez finalizado, pulsar “Open Project” para abrir el Keil uVision con el proyecto.
- Todo esto se verá en las siguientes transparencias mediante un ejemplo
 - Pero primero echemos un vistazo a las dos pantallas principales del CubeMX.

Configuración de pines y periféricos

- La pestaña de configuración de pines y periféricos se puede dividir en 3 zonas. Más adelante en el curso se explicará en más detalle.

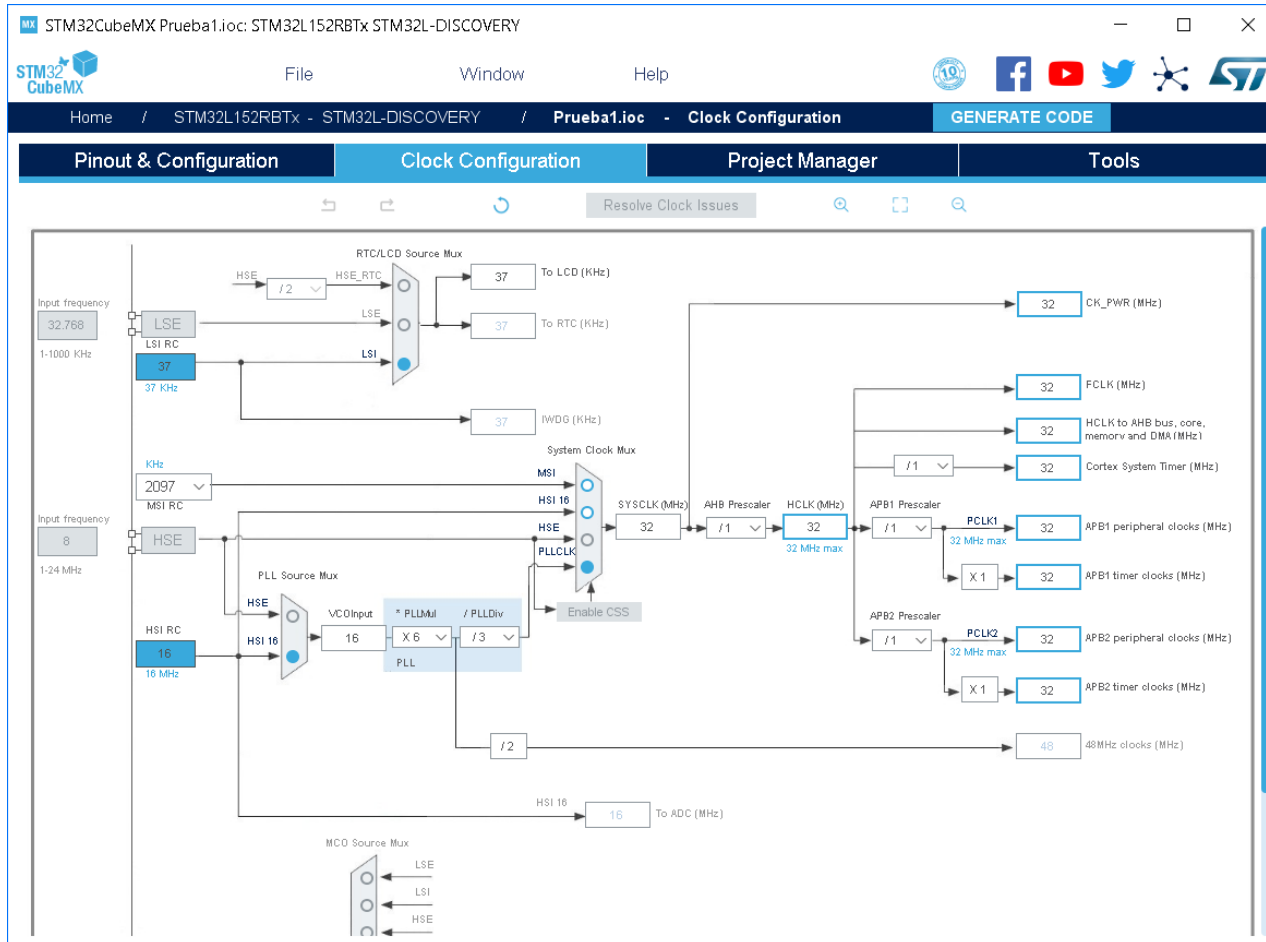
The screenshot shows the STM32CubeMX software interface for configuring the STM32L152RBTx. The 'Pinout & Configuration' tab is active, displaying the 'GPIO Mode and Configuration' section. The interface is divided into three main areas:

- Vista en árbol de periféricos:** A tree view on the left side showing the system core and various peripherals like DMA, GPIO, IWDG, NVIC, RCC, SYS, TS, and WWDG. A green arrow points to this area with the text 'Vista en árbol de periféricos'.
- Vista de los detalles del periférico seleccionado en el árbol:** A table in the center showing the configuration for selected peripherals. A green arrow points to this table with the text 'Vista de los detalles del periférico seleccionado en el árbol'. The table contains the following data:

Pin Name	Signal on Pin	GPIO output	GPIO mode	GPIO Pull-up/	Maximum out.	User Label	Modified
PA0-WKUP1	n/a	n/a	External Even.	No pull-up an...	n/a	B1 [Blue Pus...	?
PB6	n/a	Low	Output Push...	No pull-up an...	Very Low	LD4 [Blue Led]	?
PB7	n/a	Low	Output Push...	No pull-up an...	Very Low	LD3 [Green L...	?
PC13-WKUP2	n/a	Low	Output Push...	No pull-up an...	Very Low	ID0_CNT_EN	?
- Vista del chip para configurar funcionalidad de los pines:** A pinout diagram on the right side showing the physical layout of the STM32L152RBTx chip with various pins labeled. A green arrow points to this diagram with the text 'Vista del chip para configurar funcionalidad de los pines'.

Configuración de reloj

- La pestaña de configuración hay que dejarla, de momento, como en la figura, para tener reloj de 32MHz partiendo del HSI.



7 - Creación de la Estructura Básica de Trabajo

Estructura Básica de Trabajo

- A lo largo del curso se van a hacer múltiples ejercicios, tanto de clase como de Laboratorio.
- Cada uno de esos ejercicios será un Proyecto distinto.
- Por tanto, es preferible crear una estructura básica de trabajo, para así tenerlo todo ordenado.
- Crea en el disco duro de tu ordenador una carpeta para la asignatura (por ejemplo, D:\SDBM\).
 - Se recomienda que NO se utilice el escritorio, ya que las rutas son mucho más largas y muchas veces con caracteres complejos (por ejemplo, espacios).
- Descarga de Aula Global el fichero “STM32L152C-Discovery.rar” (es decir, la biblioteca de la placa Discovery) y descomprímelo dentro de la carpeta de la asignatura.
 - Debería crear en la carpeta de la asignatura una subcarpeta con nombre STM32L152C-Discovery, y dentro de esa subcarpeta debe haber 6 ficheros (entre ellos 2 .c y 2 .h).
 - Esa es la librería de alto nivel de uso del LCD.
- A partir de aquí, sigue con el proyecto que has dejado en la transparencia 55 y ejecuta los pasos de generación de proyecto que se indican a continuación.

8 - Pasos para la creación de un proyecto

Pasos para crear un Proyecto

- Hagámoslo de forma práctica, a través de un ejemplo (aprovechando todo lo que ya has creado hasta la transparencia 55):
 - El programa irá sacando mensajes en el LCD secuencialmente, con un tiempo de espera apreciable (por ejemplo, 1 segundo), entre paso y paso:
 - Paso 1:
 - LCD = “UNO”
 - Paso 2:
 - LCD = “DOS”
 - Paso 3:
 - LCD = “TRES”
 - Paso 4:
 - LCD = “MAMBO”
- Antes de ponerse a trabajar, es necesario tener claro qué periféricos y qué características de los mismos se van a necesitar
 - En este caso sólo va a ser el LCD

Pasos para crear un Proyecto

1. Cuando ya tenemos claro qué recursos se van a necesitar en el Proyecto, se arranca el STM32CubeMX.
2. Se selecciona ACCESS TO BOARD SELECTOR.
3. Se selecciona la STM32L-DISCOVERY.
4. Se pulsa Start Project.
5. Y se indica “Yes” para inicializar los periféricos en modo por defecto (hasta aquí ya los has hecho hasta la transparencia 55).
6. Esto arranca la pantalla de configuración de pines y periféricos
 1. En el árbol se secciona Multimedia
 2. Se pincha en LCD
 3. En la vista de detalles, en la parte superior, se indica que el LCD se usará en Modo 1/4 Duty Cycle
 4. Y ahora se modifican los parámetros de configuración según se indica en la transparencia siguiente

Configuración del LCD

Options

LCD Mode and Configuration

Mode

Mode 1/4 Duty Cycle

Multiplex Mode

SEG0

SEG1

SEG2

SEG3

SEG4

SEG5

SEG6

SEG7

SEG8

Reset Configuration

Configuration

Reset Configuration

Parameter Settings User Constants NVIC Settings GF

Configure the below parameters :

Search (Ctrl+F)

▼ Clock Parameters

Clock Prescaler 1

Clock Divider 31

▼ Basic Parameters

Duty Selection 1/4

Bias Selector 1/3

Multiplex mode Disable

▼ Advanced Parameters

Voltage Source Selection Internal

Contrast Control 2.60V

Dead Time Duration No dead Time

High Drive Disable

Pulse ON Duration 4/CK_PS

Blink Mode Disabled

Blink Frequency fLCD/32

- Clock Divider = 31
- Bias Selector = 1/3
- Pulse ON Duration = $4/CK_PS$
- Blink Frequency = $f_{LCD}/32$

Siguientes Pasos

7. Selecciona ahora la pestaña “Clock Configuration” y se asegúrate que está configurado como se ha comentado anteriormente
 - HSI seleccionado, y el resto de parámetros necesarios para tener reloj de 32MHz
8. Selecciona la pestaña “Project Manager”:
 - En la sub-pestaña “Project” (transparencia 66):
 1. Se le da un nombre al proyecto (por ejemplo, Tema5_Ej1)
 2. Se selecciona la carpeta de la asignatura (por ejemplo, D:\SDBM\)
 3. “Application Structure” se deja en “Basic”
 4. “Toolchain / IDE” se selecciona como “MDK-ARM V5”
 - En la sub-pestaña “Code Generator” (transparencia 67):
 1. Se activa la opción “Copy only the necessary library files”
9. Ahora se pulsa “GENERATE CODE”
10. Para finalizar, se pulsa “Open Project” (transparencia 68)

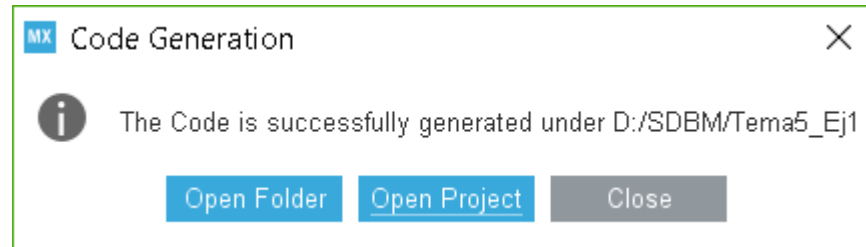
Configuración del Proyecto

Project	<p>Project Settings</p> <p>Project Name Tema5_Ej1</p> <p>Project Location D:\SDBM Browse</p>
Code Generator	<p>Application Structure Basic ▼ <input type="checkbox"/> Do not generate the main()</p> <p>Toolchain Folder Location D:\SDBM\Tema5_Ej1\</p> <p>Toolchain / IDE MDK-ARM V5 ▼ <input type="checkbox"/> Generate Under Root</p>
Advanced Settings	<p>Linker Settings</p> <p>Minimum Heap Size <input type="text" value="0x200"/></p> <p>Minimum Stack Size <input type="text" value="0x400"/></p>
	<p>Mcu and Firmware Package</p> <p>Mcu Reference STM32L152RBTx</p> <p>Firmware Package Name and Version STM32Cube_FW_L1_V1.8.1 ▼ <input checked="" type="checkbox"/> Use latest available version</p> <p><input checked="" type="checkbox"/> Use Default Firmware Location <input type="text" value="C:/Users/Raul/STM32Cube/Repository/STM32Cube_FW_L1_V1.8.1"/> Browse</p>

Configuración del Proyecto

Project	<p>STM32Cube Firmware Library Package</p> <ul style="list-style-type: none"><input type="radio"/> Copy all used libraries into the project folder<input checked="" type="radio"/> Copy only the necessary library files<input type="radio"/> Add necessary library files as reference in the toolchain project configuration file
Code Generator	<p>Generated files</p> <ul style="list-style-type: none"><input type="checkbox"/> Generate peripheral initialization as a pair of '.c/.h' files per peripheral<input type="checkbox"/> Backup previously generated files when re-generating<input checked="" type="checkbox"/> Keep User Code when re-generating<input checked="" type="checkbox"/> Delete previously generated files when not re-generated
Advanced Settings	<p>HAL Settings</p> <ul style="list-style-type: none"><input type="checkbox"/> Set all free pins as analog (to optimize the power consumption)<input type="checkbox"/> Enable Full Assert
	<p>Template Settings</p> <p>Select a template to generate customized code Settings...</p>

Finalizando con el STM32 CubeMX

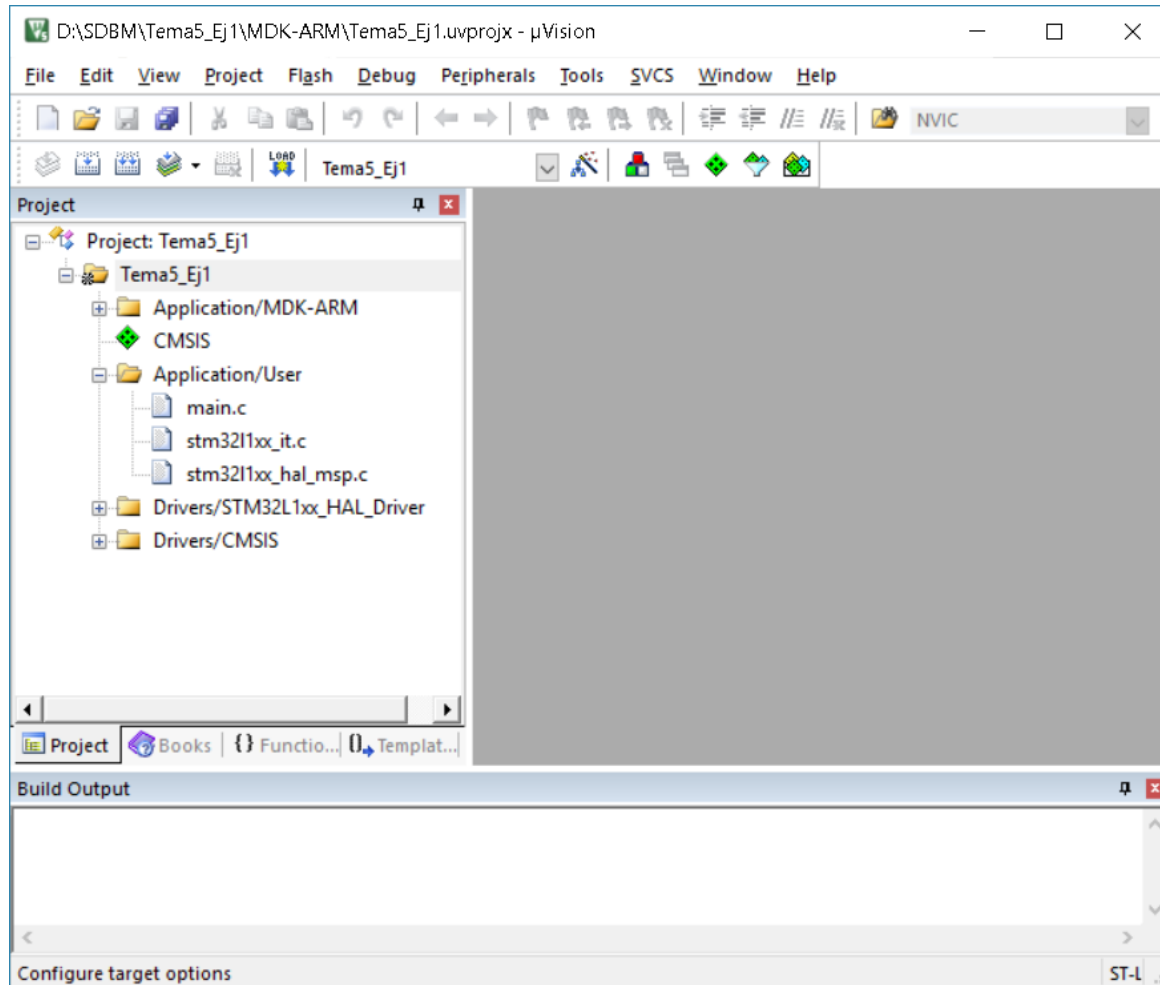


Trabajando con el Keil uVision5

- Con esto se crea y se abre el uVision, los siguientes pasos a realizar son los siguientes (todos estos pasos se comentan en las siguientes transparencias):
 1. Abrir las Opciones del Target.
 2. Añadir a la ruta de compilación de C/C++ la ubicación de la biblioteca STM32L152C-Discovery (la que se copió en el raíz de la carpeta de la asignatura en la transparencia 60).
 3. Configurar las herramientas de depuración.
 4. Configurar las utilidades del entorno.
 5. Añadir la Biblioteca de la Discovery.
 6. Escribir el Programa de la Solución.
 7. Compilar y Probar la solución.
- Posteriormente se explicará como depurar la solución.

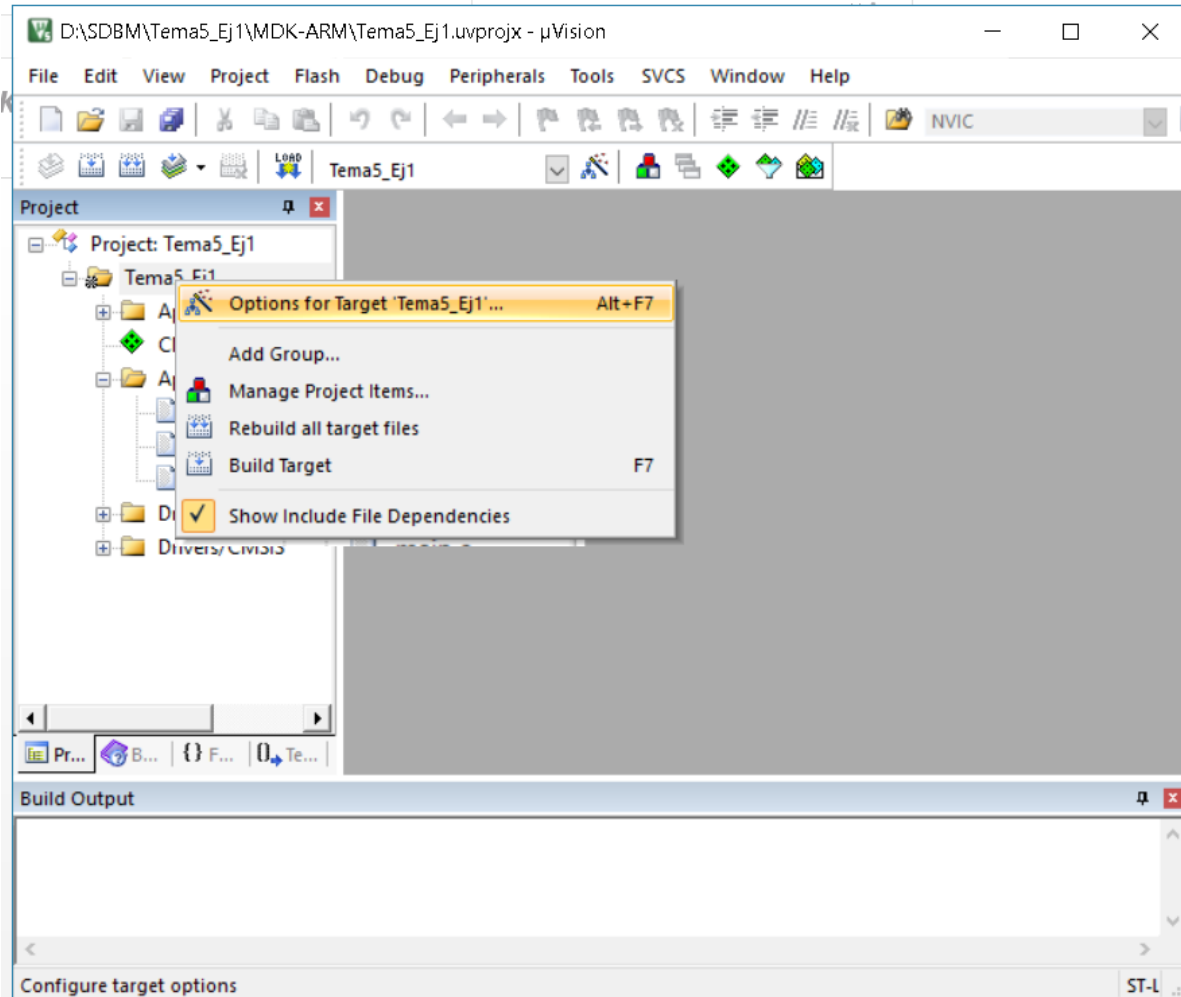
Configurando el Proyecto en Keil uVision5

- Al abrirse el uVision, verás la siguiente estructura ya creada.



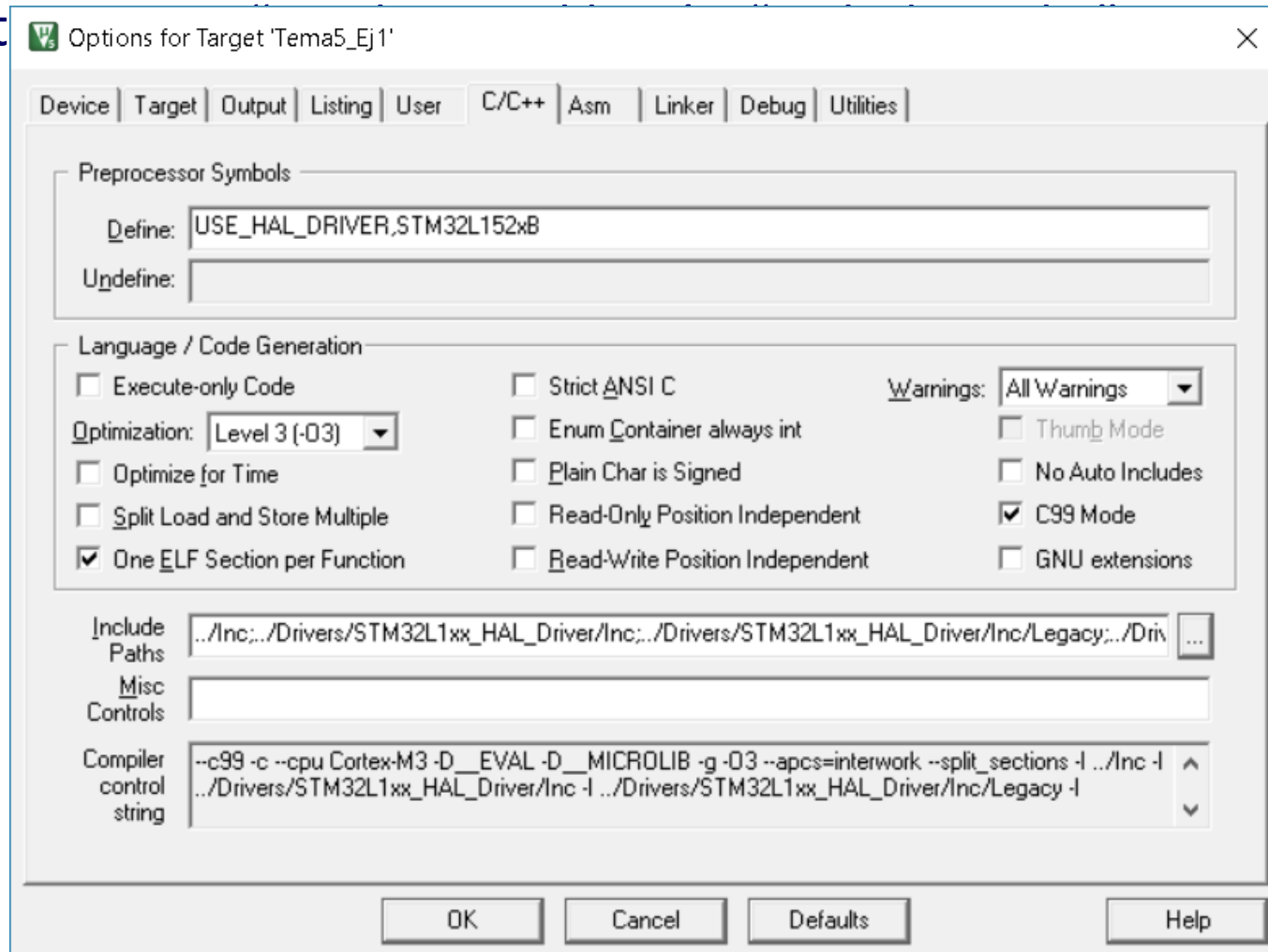
Configurando el Proyecto en Keil uVision5

- Abre el menú emergente del proyecto y elije “Options for Target...”.



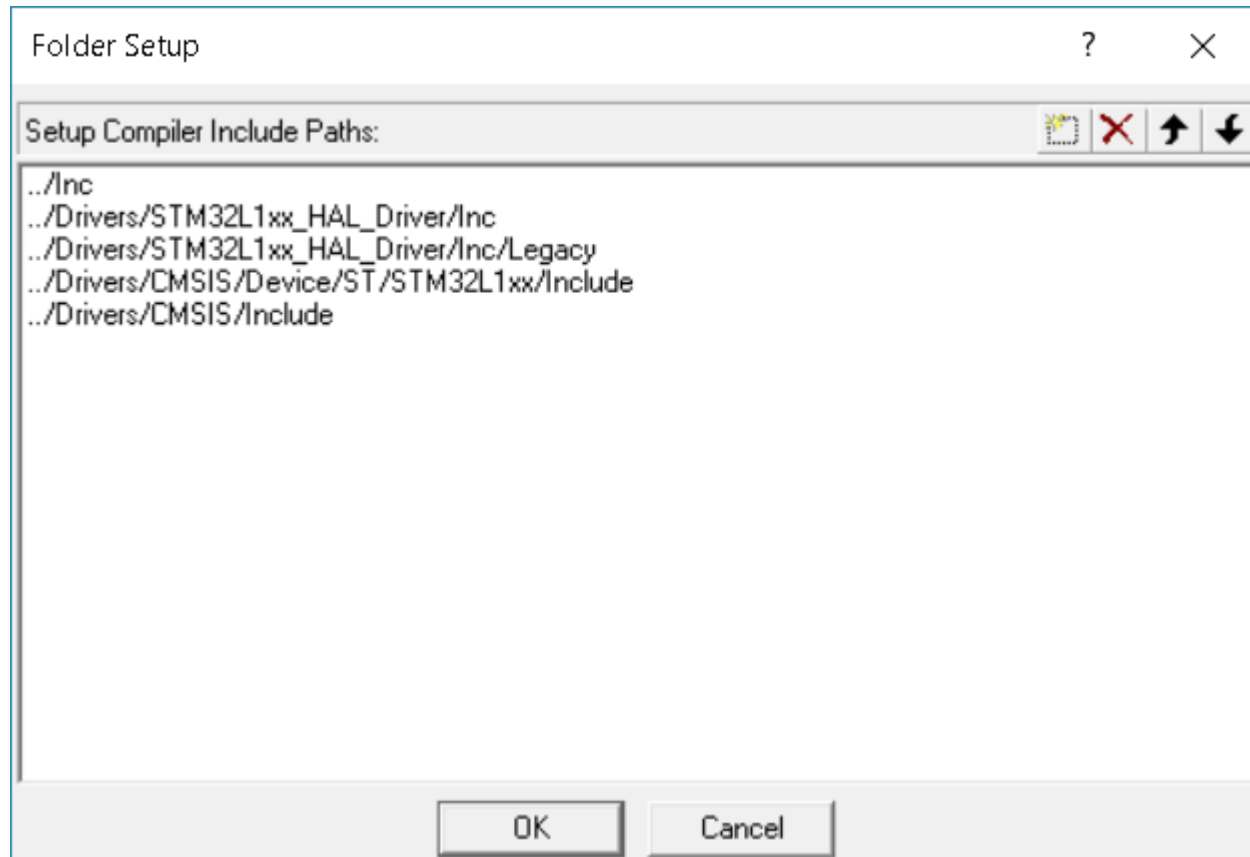
Configurando el Proyecto en Keil uVision5

- Selecciona la pestaña “C/C++”, ajusta la opción “Level 0” en “Opt



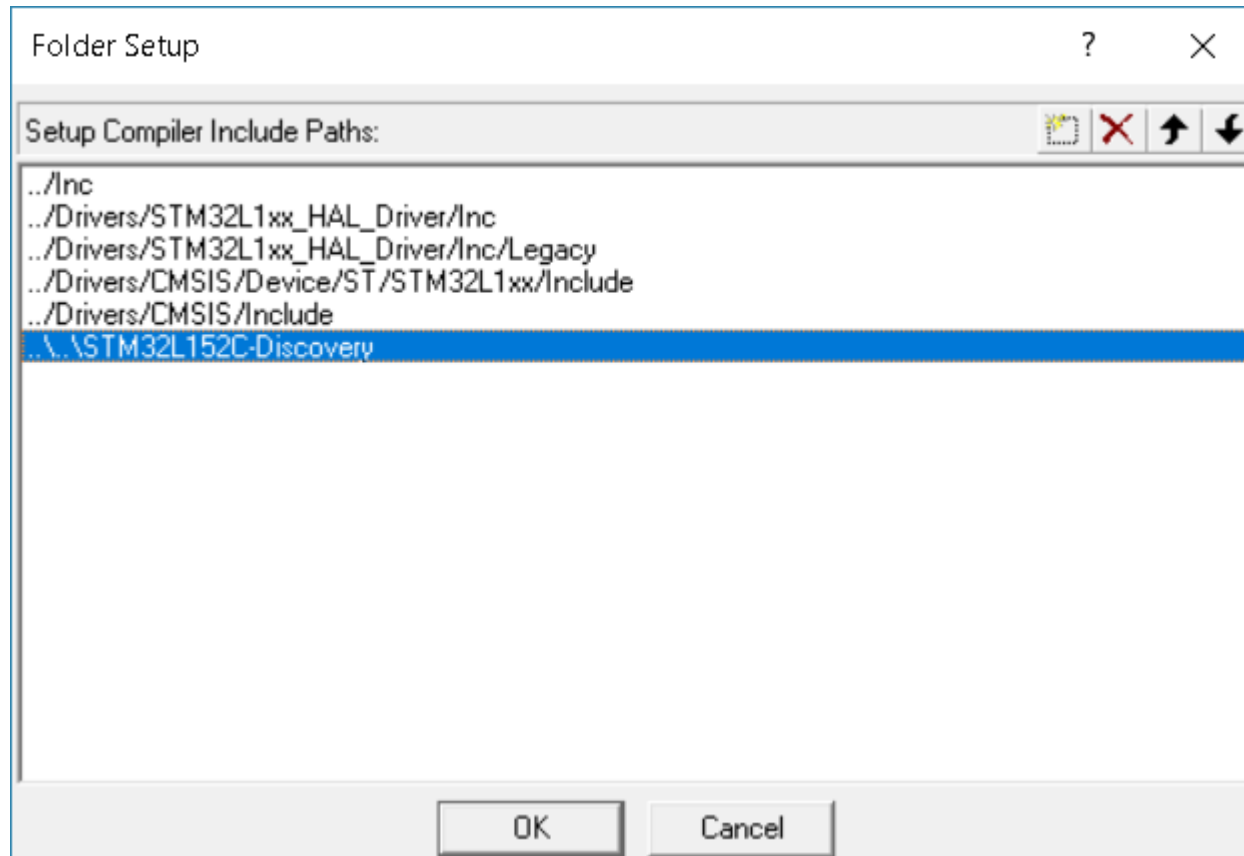
Configurando el Proyecto en Keil uVision5

- Pulsa en el botón de “Add New” (al lado de la cruz roja).



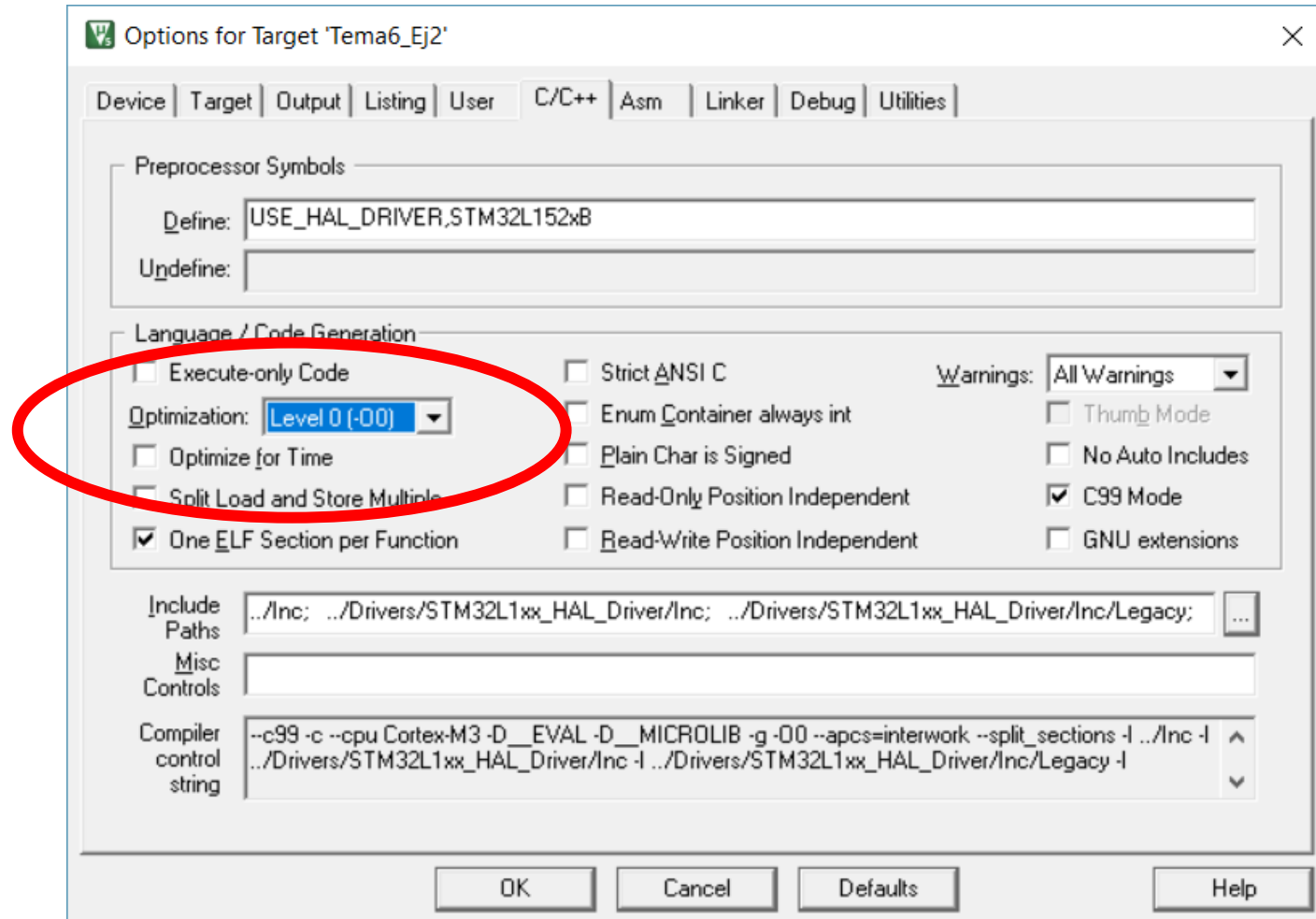
Configurando el Proyecto en Keil uVision5

- Selecciona la carpeta “STM32L152C-Discovery” que se creó en el raíz de la carpeta de la asignatura, la de la transparencia 60).



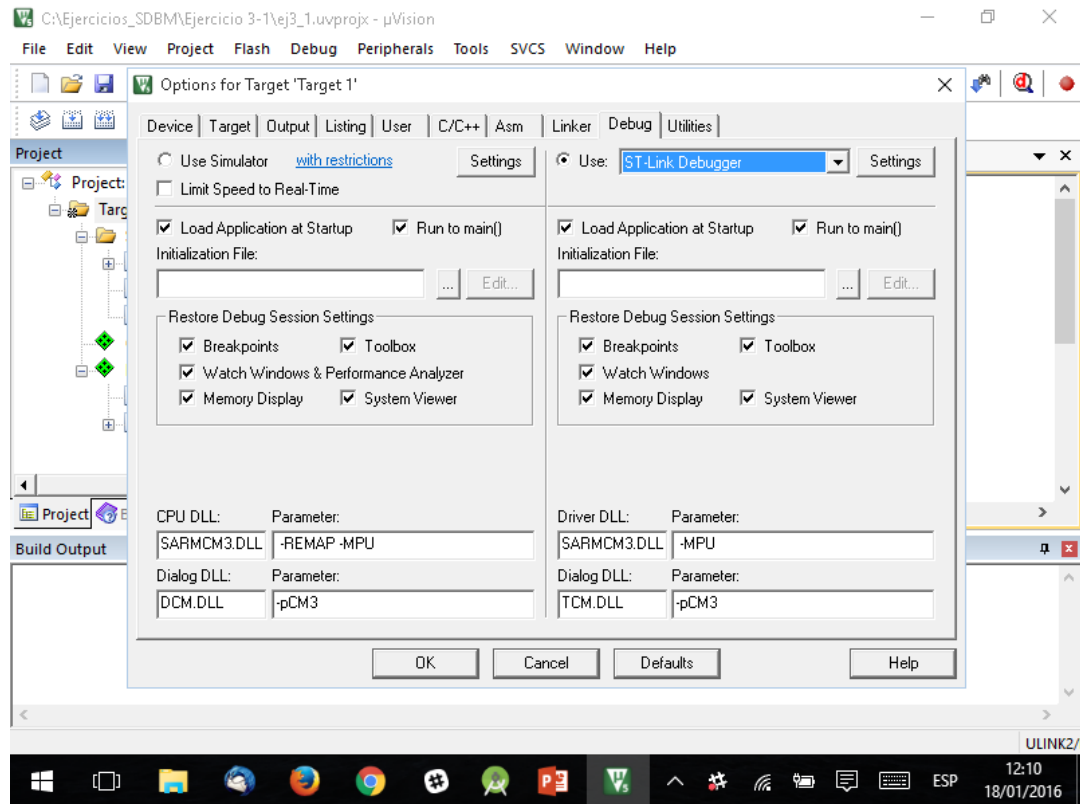
Configurando el Proyecto en Keil uVision5

- **IMPORTANTE:** Selecciona el nivel de optimización a 0, y no actives la optimización por tiempo



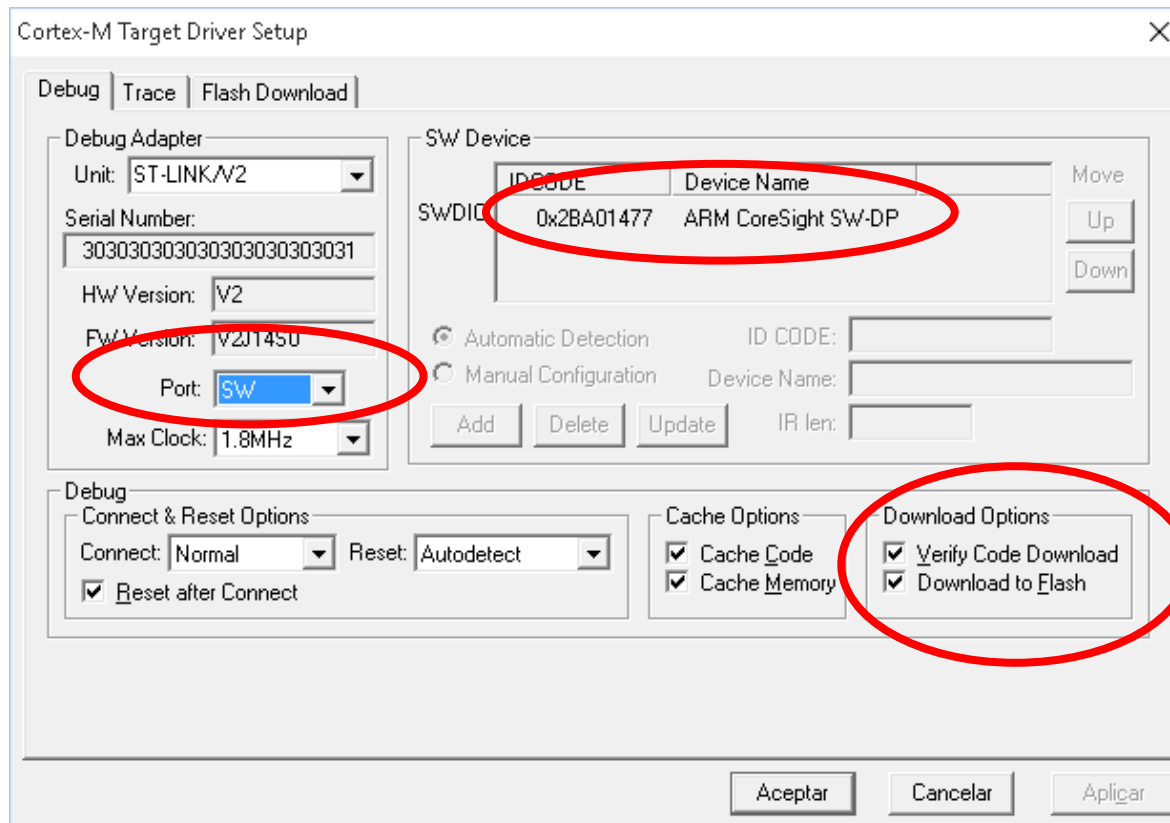
Configurando el Proyecto en Keil uVision5

- Selecciona la pestaña “Debug” y selecciona la opción de utilizar un depurador (la columna de la derecha), y selecciona “ST-Link Debugger” dentro de las opciones de depuradores.
- Fíjate en las casillas que están activadas en la figura, puesto que es la configuración recomendada.

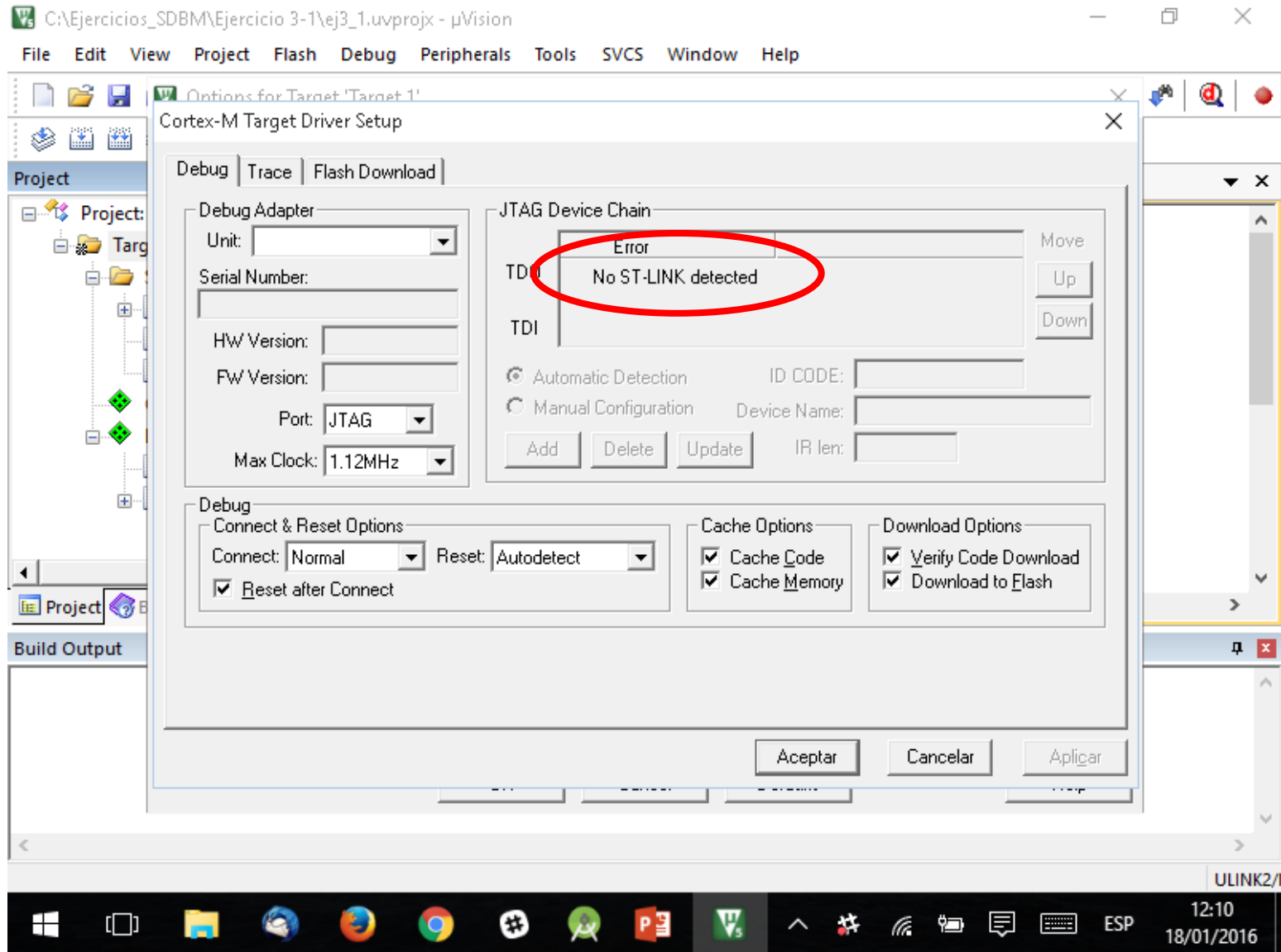


Configurando el Proyecto en Keil uVision5

- Después de seleccionar el depurador (ST-Link Debugger), pulsa el botón “Settings”, para ver la configuración del mismo.
- Selecciona “SW” como puerto.
- Si se tiene conectada la placa (recomendado), entonces verás a la derecha los datos de la placa conectada (ver esta imagen). **Si no, se verá una pantalla como la de la figura de la siguiente transparencia.**
- En cualquier caso, fíjate en las casillas seleccionadas y usa esta configuración

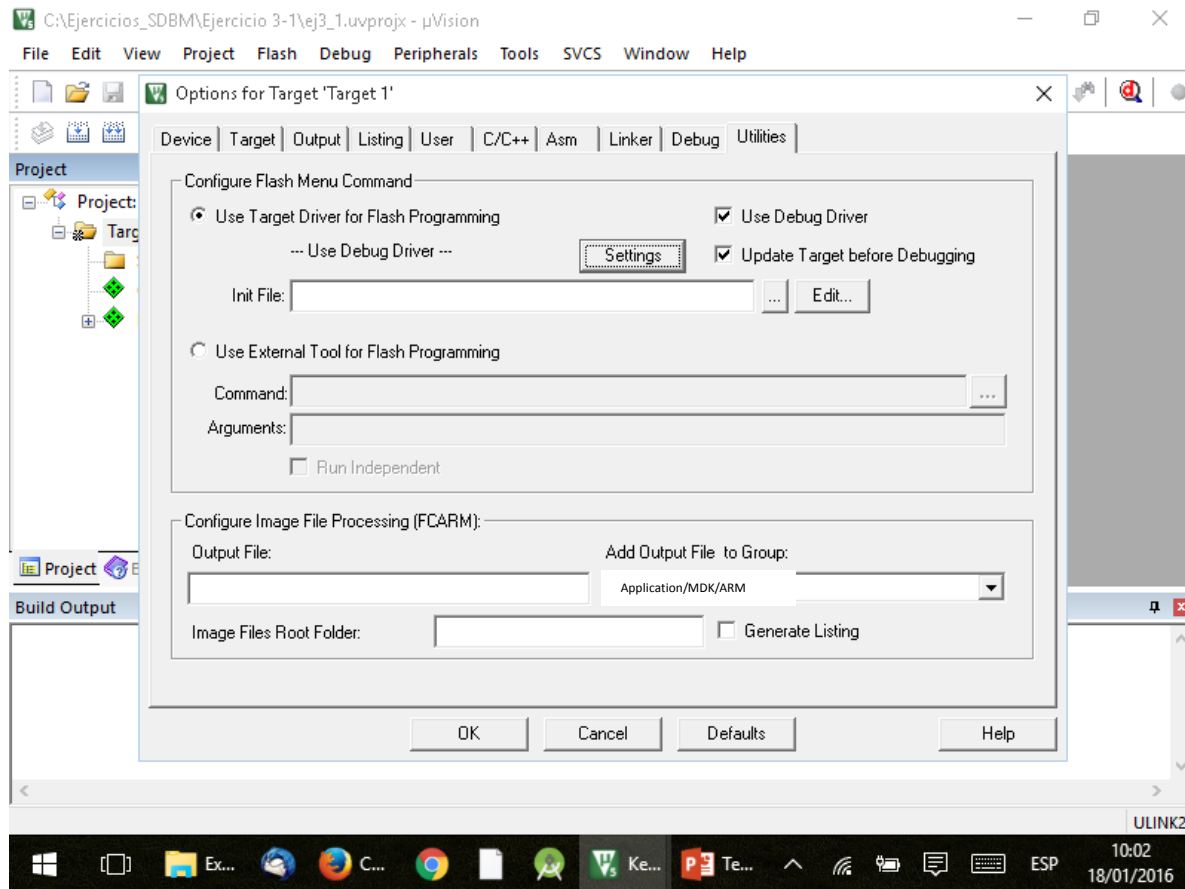


Configurando el Proyecto en Keil uVision5



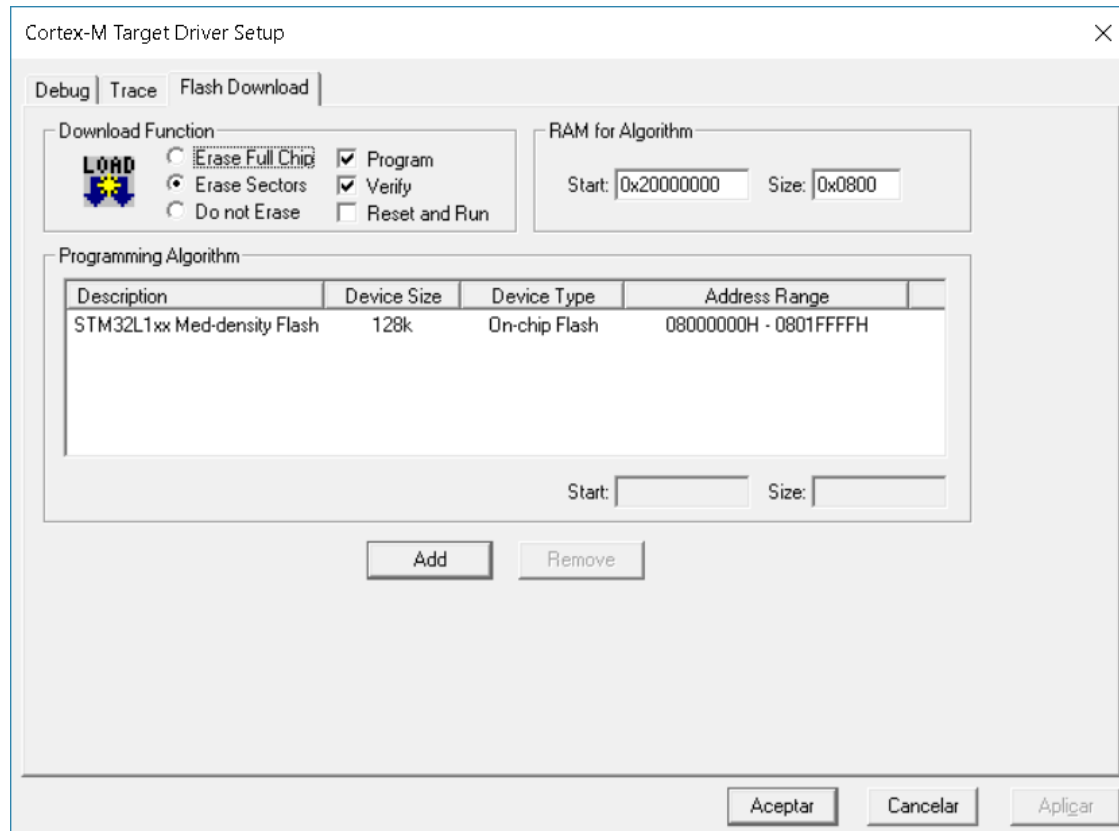
Configurando el Proyecto en Keil uVision5

- Después de aceptar las opciones del depurador, selecciona la pestaña “Utilities” y copia la configuración que se muestra en la pantalla (deja todo por defecto según sale).



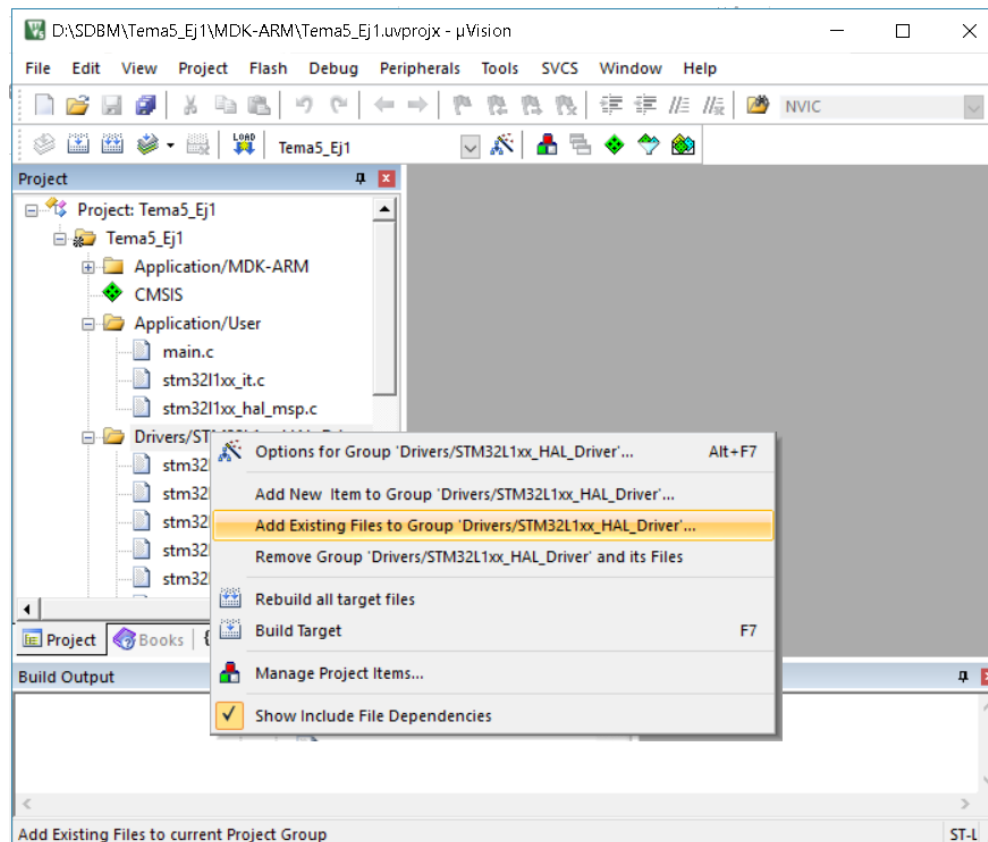
Configurando el Proyecto en Keil uVision5

- Pulsa en el botón “Settings” (dentro de “Utilities”) y verifica que hay un mapa de memoria instalado (por ejemplo el expuesto en la figura).
- También se recomienda que se seleccione “Erase Sectors”.



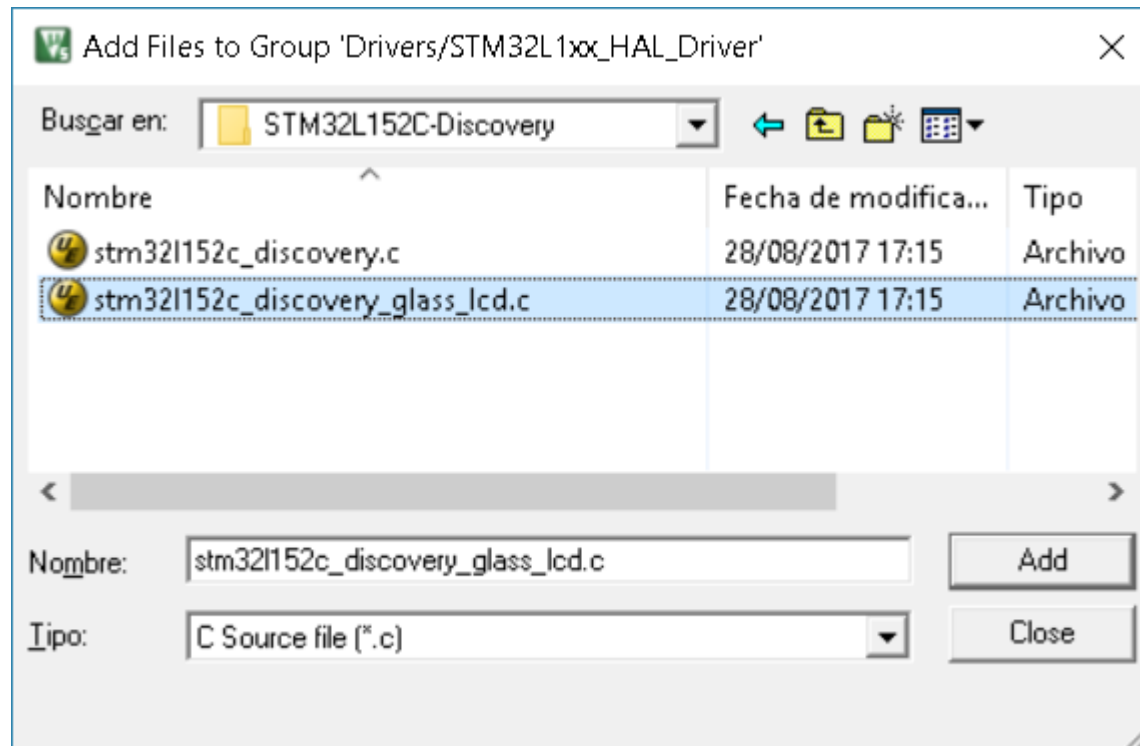
Añadiendo la Biblioteca de la Discovery

- El siguiente paso es añadir al proyecto la biblioteca de la Placa Discovery, para poder acceder a las funciones del LCD.
- Para hacer eso, selecciona la opción “Add Existing Files to ...” en la carpeta “Drivers/STML32”.



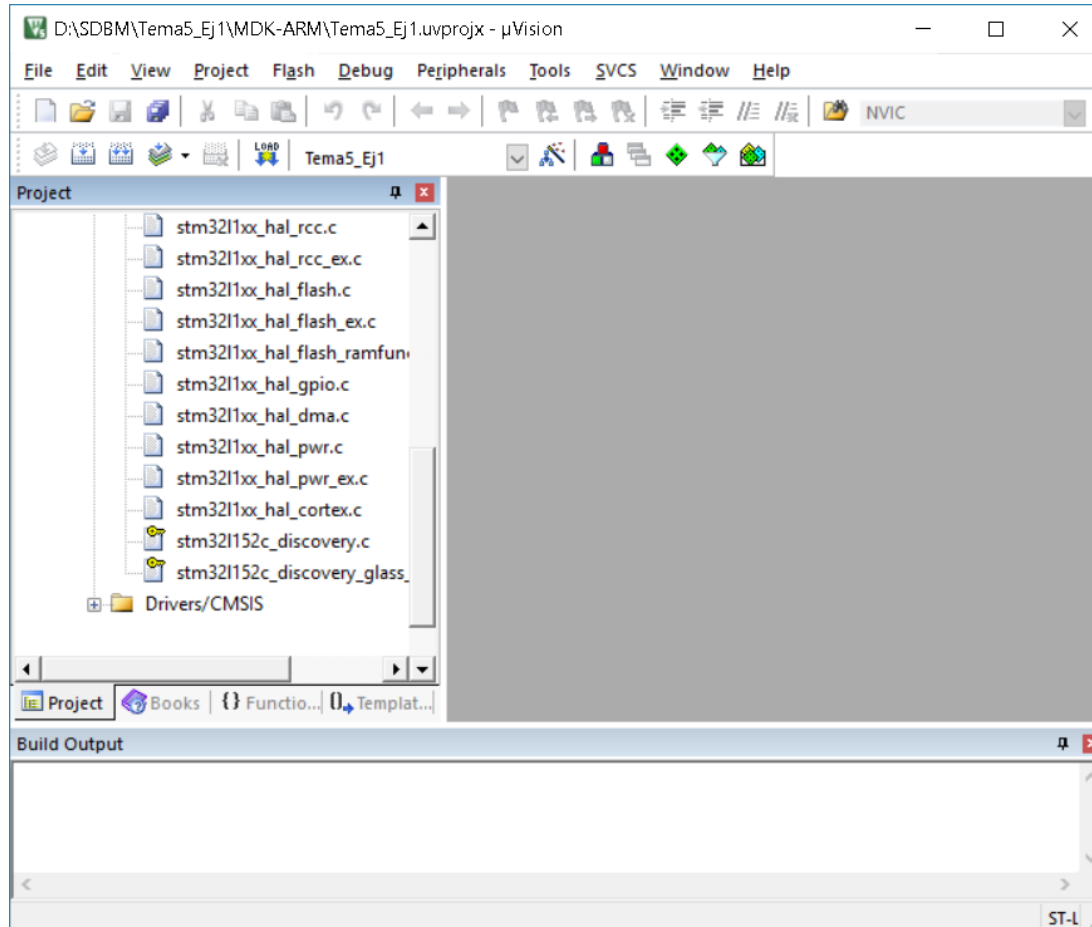
Añadiendo la Biblioteca de la Discovery

- Añade los dos ficheros .c que contiene la carpeta “STM32L152C-Discovery” que se instaló en la carpeta de la asignatura en la transparencia 60.



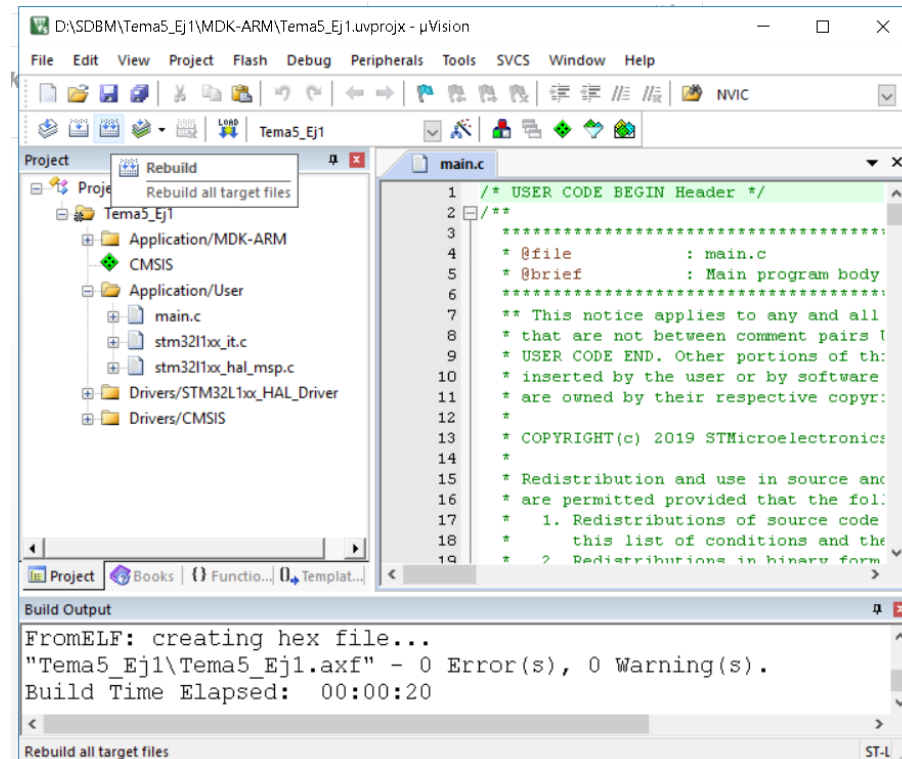
Añadiendo la Biblioteca de la Discovery

- El resultado será que se añadirán los dos ficheros a la lista de ficheros de esa carpeta del árbol del proyecto.



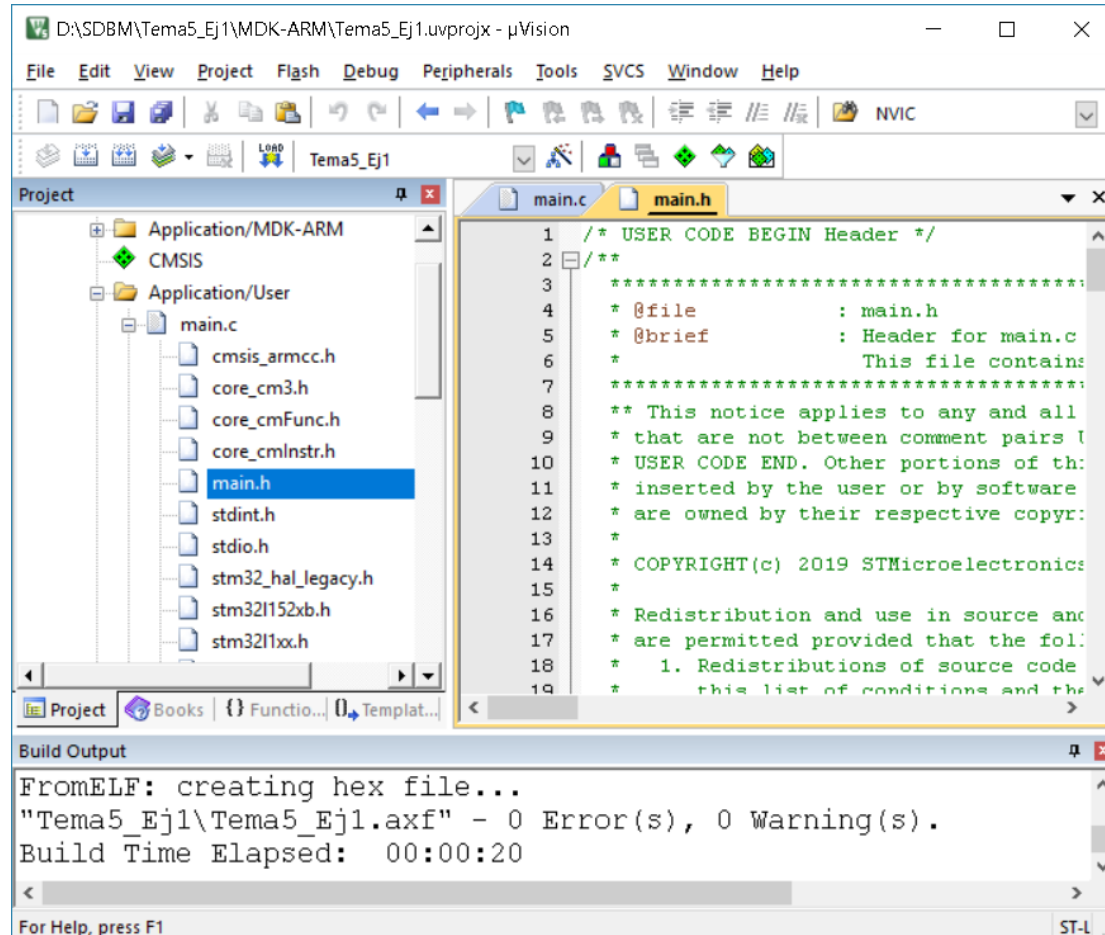
Añadiendo la Biblioteca de la Discovery

- Para facilitar el siguiente paso, compila el proyecto para crear en el árbol las relaciones entre los ficheros .c y los .h.
- Para hacer eso, selecciona la opción “Project -> Rebuild all target files”. Si todo está correcto deberá salir abajo 0 errores.



Añadiendo la Biblioteca de la Discovery

- Selecciona el fichero “main.h” (que está dentro del fichero “mail.c”) para editarlo.



Añadiendo la Biblioteca de la Discovery

- En el fichero “main.h” localiza la línea:
 - /* Private includes -----*/
 - /* USER CODE BEGIN Includes */
- Y escribe después de esa línea las dos siguientes:
 - #include "stm32l152c_discovery.h"
 - #include "stm32l152c_discovery_glass_lcd.h"
- Graba y cierra el fichero “main.h”.
- Vuelve al fichero “main.c” y localiza la función:
 - int main(void)
- Esta será la única función que vas a modificar en el fichero “main.c”, ya que el resto del código se hará en librerías, tal y como se le enseñará posteriormente.

Programando el Microcontrolador

- Todo programa de microcontrolador se basa en 2 partes fundamentales de la función main():
 - Una parte de inicialización de los periféricos, las variables de la función “main”, las inicialización de las funciones del LCD, etc.
 - Se ejecutará una única vez al principio del programa.
 - Una parte de código metida dentro de un bucle infinito, que representará la funcionalidad continua deseada.
- Por lo tanto tu trabajo constará en escribir dos partes de código:
 - Una de inicialización:
 - Se escribirá en la función main(), entre las líneas
 - `/* USER CODE BEGIN 2 */`
 - `/* USER CODE END 2 */`
 - Una de funcionamiento cíclico del programa:
 - Se escribirá en la función main(), entre las líneas
 - `/* USER CODE BEGIN WHILE */`
 - `while(1)`
 - `{`
 - `}`
 - `/* USER CODE END WHILE */`

El Código del Ejemplo

- La inicialización para el LCD en este ejemplo es la siguiente (escríbela en el archivo):

```
/* USER CODE BEGIN 2 */  
BSP_LCD_GLASS_Init();  
BSP_LCD_GLASS_BarLevelConfig(0);  
BSP_LCD_GLASS_Clear();  
/* USER CODE END 2 */
```

El Código del Ejemplo

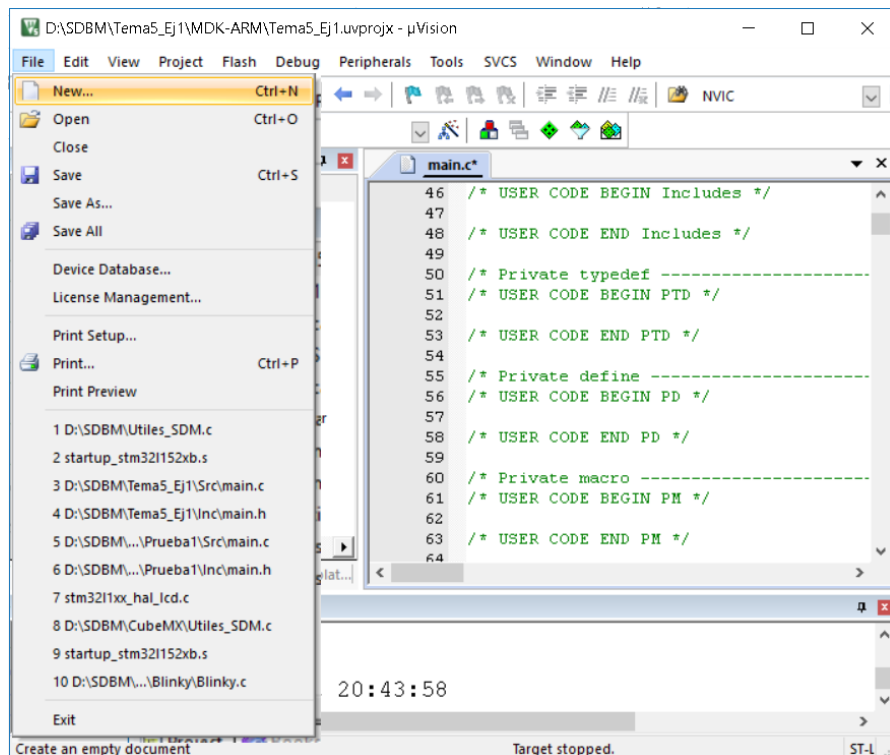
- La ejecución cíclica para este ejemplo es la siguiente (escríbela en el archivo):

```
/* USER CODE BEGIN WHILE */
while (1)
{
    BSP_LCD_GLASS_Clear();
    BSP_LCD_GLASS_DisplayString((uint8_t *) "UNO");
    espera(5000000);
    BSP_LCD_GLASS_Clear();
    BSP_LCD_GLASS_DisplayString((uint8_t *) "DOS");
    espera(5000000);
    BSP_LCD_GLASS_Clear();
    BSP_LCD_GLASS_DisplayString((uint8_t *) "TRES");
    espera(5000000);
    BSP_LCD_GLASS_Clear();
    BSP_LCD_GLASS_DisplayString((uint8_t *) "MAMBO");
    espera(5000000);
/* USER CODE END WHILE */

/* USER CODE BEGIN 3 */
}
/* USER CODE END 3 */
```

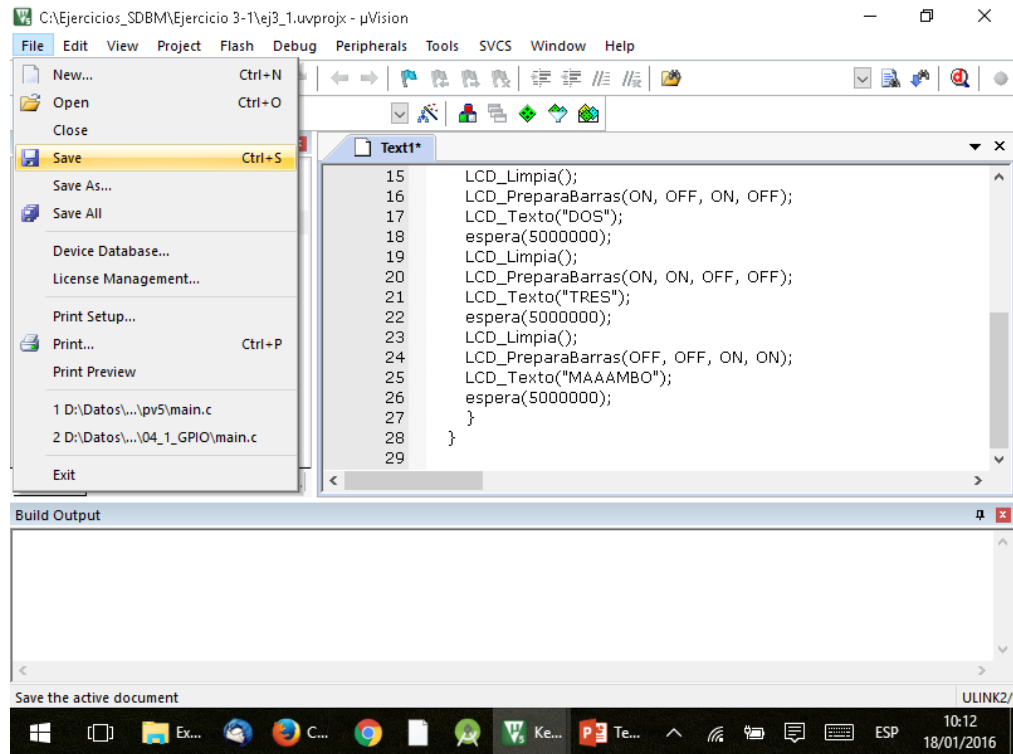
Creación de una biblioteca propia

- Como se puede ver en el código, hay una función que no está definida: espera()
 - Esa función será de utilidad en varios proyectos futuros, por lo que se va a crear una biblioteca de funciones propias.
- Los siguientes pasos muestran cómo hacerlo, pero no se mostrará el código puesto que es un ejercicio propuesto.
- Empezamos creando un fichero fuente, para lo que seleccionamos “File -> New...”.



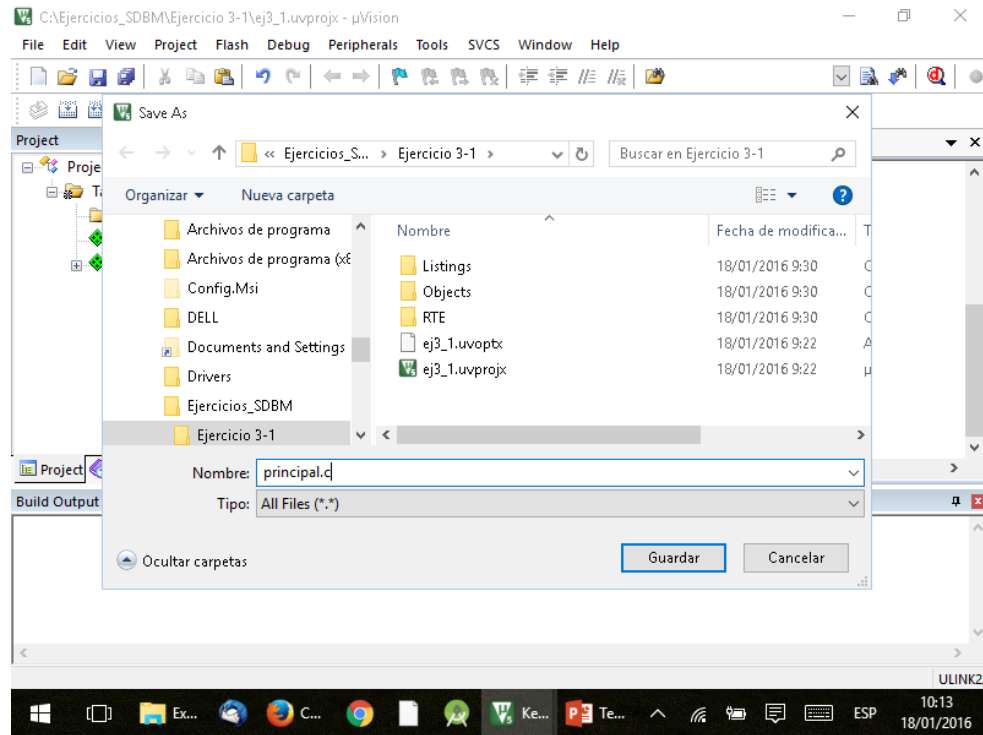
Creación de una biblioteca propia

- Una vez escrito el código (en el que no está resaltada la sintaxis por no saber el entorno que tipo de fichero es), hay que guardar el fichero con la opción “File -> Save”.
- Cuidado: lo que ves en la imagen no es el código se la librería espera(), ese lo tienes que pensar tú.



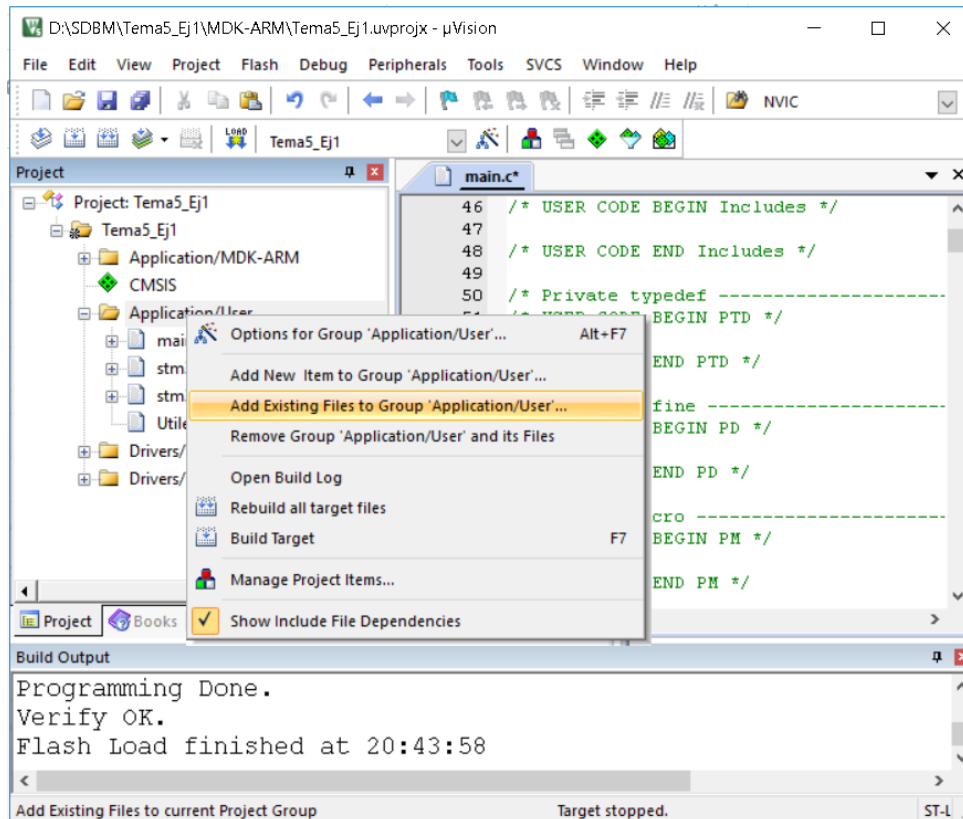
Creación de una biblioteca propia

- Pónle un nombre con la extensión adecuada al formato (por ejemplo, si es un fichero en C, pónle un nombre acabado con .c; si es un fichero “header” de C, pónle un nombre acabado con .h)
- Al almacenarlo con un nombre y extensión, ya será posible que el entorno resalte la sintaxis con colores.
- El nombre del fichero puede ser cualquiera, pero se aconseja que sea un nombre representativo (por ejemplo, Utils_SDBM.c y Utils_SDBM.h).



Creación de una biblioteca propia

- Sin embargo, todavía este fichero no está en el proyecto, por lo que habrá que decirle al proyecto la necesidad de incorporarlo al mismo.
- Para ello pulsa el botón derecho en “Application/User”, selecciona la opción “Add Existing Files to Group ...” y mete sólo el archivo .C.

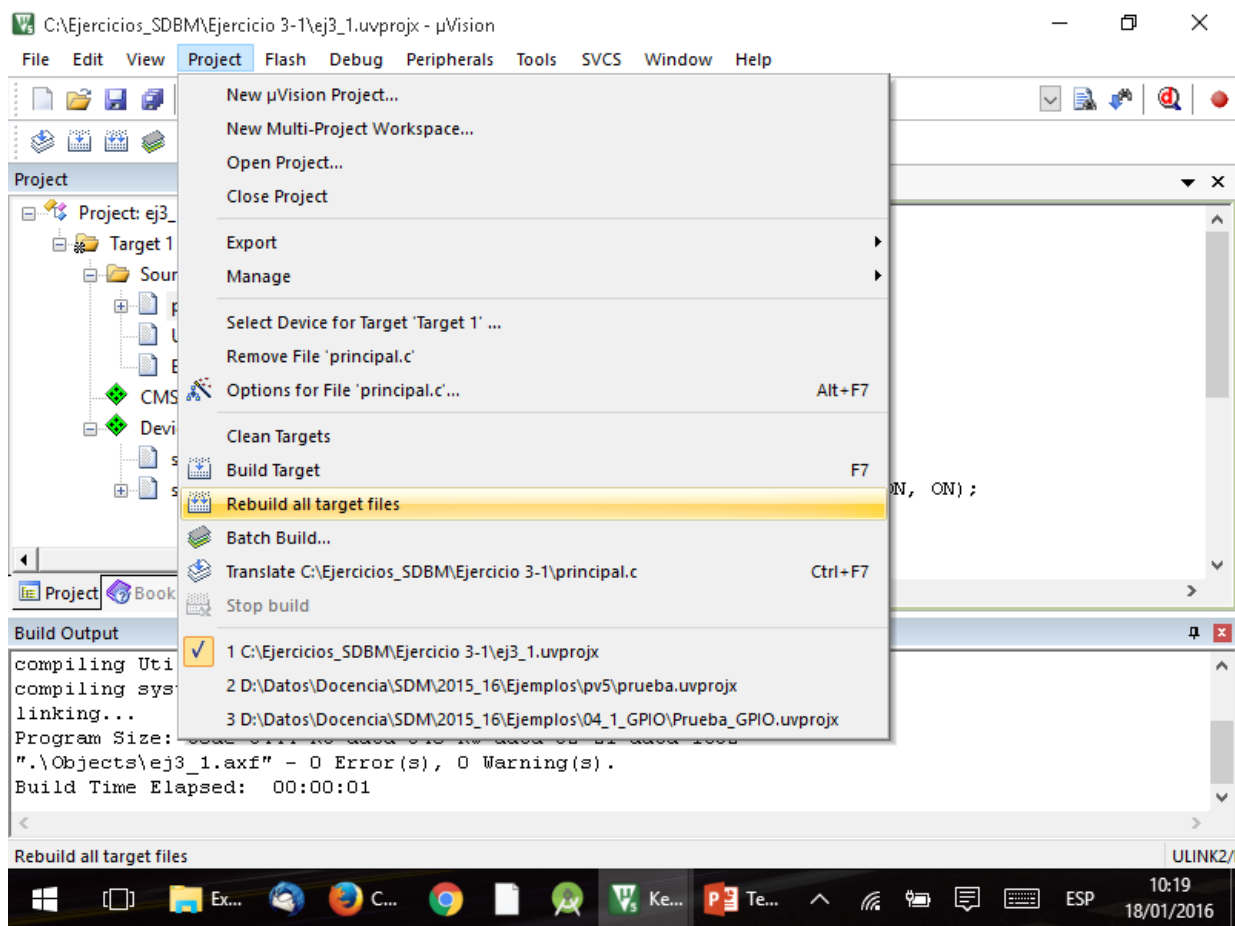


Creación de una biblioteca propia

- Se puede ver el fichero integrado ahora en el proyecto, pero hay que añadir el .h en el código del proyecto.
- Añade el .h en un #include dentro del código del fichro “main.c”.
 - Localiza la línea #include main.h
 - Después de esa línea incluye la siguiente (cuidado: el nombre del fichero debe ser el que hayas dado al tuyo, y ten cuidado con la localización relativa al proyecto):
 - #include "..\..\Utils_SDM.h“

Compilación

- El proyecto ya está listo para compilar.
- Selecciona la opción “Project -> Rebuild all target files”.



Compilación

- Durante la compilación verá aparecer mensajes en la ventana inferior del programa, y finalmente un mensaje diciendo el número de errores y avisos (warnings) resultantes.
- Una vez conseguido 0 errores, está listo para ejecutar el proyecto.


```
1 /** Ejemplo de uso del LCD **/  
2  
3 #include "Biblioteca_SDM.h"  
4 #include "Utiles_SDM.h"  
5  
6 int main(void) {  
7     Init_SDM();  
8     Init_LCD();  
9  
10    while (1) {  
11        LCD_Limpia();  
12        LCD_PreparaBarras(ON, ON, ON, ON);  
13        LCD_Texto ("UNO");  
14        espera (5000000);  
15        LCD_Limpia();  
16    }
```

Build Output


```
assembling startup_stm3211xx_md.s...  
compiling system_stm3211xx.c...  
linking...  
Program Size: Code=6444 RO-data=348 RW-data=32 ZI-data=1632  
".\Objects\ej3_1.axf" - 0 Error(s), 0 Warning(s).  
Build Time Elapsed: 00:00:00
```

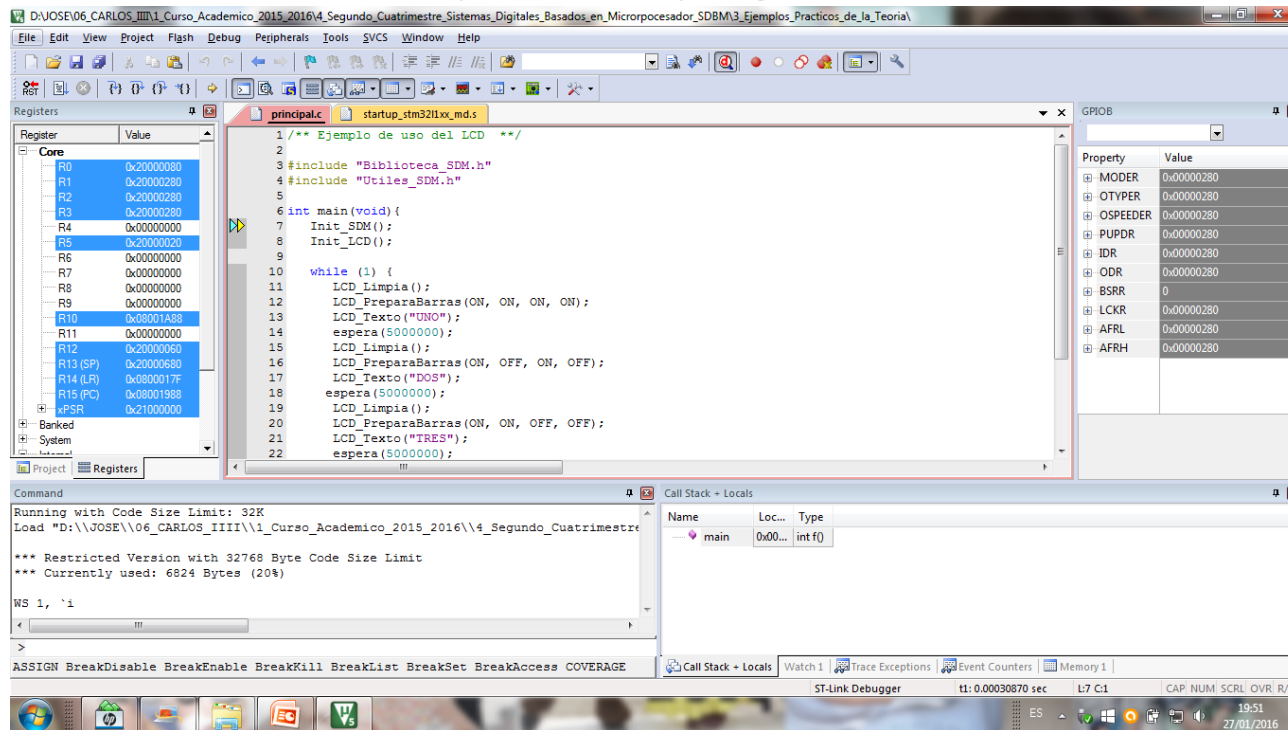
9 - Pasos para la depuración de un proyecto

Depuración de un Proyecto

- Una vez compilado correctamente, conecta la placa al puerto USB (si no estaba ya) y ejecuta el depurador con el botón  :






 - Tardará bastante hasta que cargue y luego aparecerá un aviso del modo de evaluación. Pulsa “Aceptar”.

- Si se abre una ventana con ensamblador, pulsa  para cerrarla.
- Como se puede ver, el programa se ha ejecutado hasta llegar a la función main(). La flecha amarilla indica el punto del programa donde se encuentra la ejecución.



The screenshot shows the ST-Link Debugger interface. The main window displays the source code of 'principal.c' with a yellow arrow pointing to line 6, which is the start of the 'main' function. The left sidebar shows the 'Registers' window with the 'Core' registers listed. The bottom status bar indicates the program is running with a code size limit of 32K and is currently using 6824 Bytes (20%). The system tray shows the date as 27/01/2016 and the time as 19:51.

Depuración de un Proyecto

- Para depurar, se pueden utilizar las siguientes opciones:
 - 1) Ejecutar de corrido todo el programa desde el punto actual. Si hay puntos de ruptura, se para en ese punto: 
 - 2) Se depura la función actual, pero sin entrar en ella: 
 - 3) Se ejecuta todo el programa hasta el punto donde actualmente se encuentra el cursor: 
 - 4) Se depura la función actual, entrando en ella, si es necesario: 
 - 5) Se sale de la función actual (se ejecuta lo que queda de función y se pausa la ejecución al finalizarla). Solo funciona cuando se ha activado la anterior previamente: 
- Además se puede poner un punto de ruptura (breakpoint), es decir pausar la ejecución cuando se llega a ese punto), haciendo doble clic en cualquier punto a la izquierda del número de línea.
 - Aparecerá un círculo en rojo.
 - Si se vuelve a hacer doble clic, se quita el breakpoint.

Depuración de un Proyecto

- Para examinar el valor de una variable, abra una Watch Window (View -> Watch Windows-> Watch 1)
 - En la parte inferior derecha, junto con “Call Stack” y con “Memory 1” se abre una nueva pestaña denominada “Watch 1”.
 - En dicha pestaña, escribe el nombre de la variable, registro o conjunto de registros.
 - Por ejemplo, escriba RCC y verá una estructura completa con los registros del reloj.
 - En cada variable se puede ver su valor (sólo si se entra en la función donde se ejecuta) o incluso modificarlo para que sea tenido en cuenta en el siguiente paso de ejecución.
- Para ver los registros asociados a los periféricos o modificar su valor, selecciona la opción “Peripherals ->System Viewer” y ahora haz visible el registro deseado (por ejemplo: GPIO -> GPIOB)
 - Por ejemplo, activa el LED verde y rojo en PB6 y PB7, activando los bits BS6 y BS7 del registro BSRR en GPIOB, o desactívelos activando los bits BR6 y BR7 de dicho registro (cuando el depurador esté dentro de la función WHILE(1)).

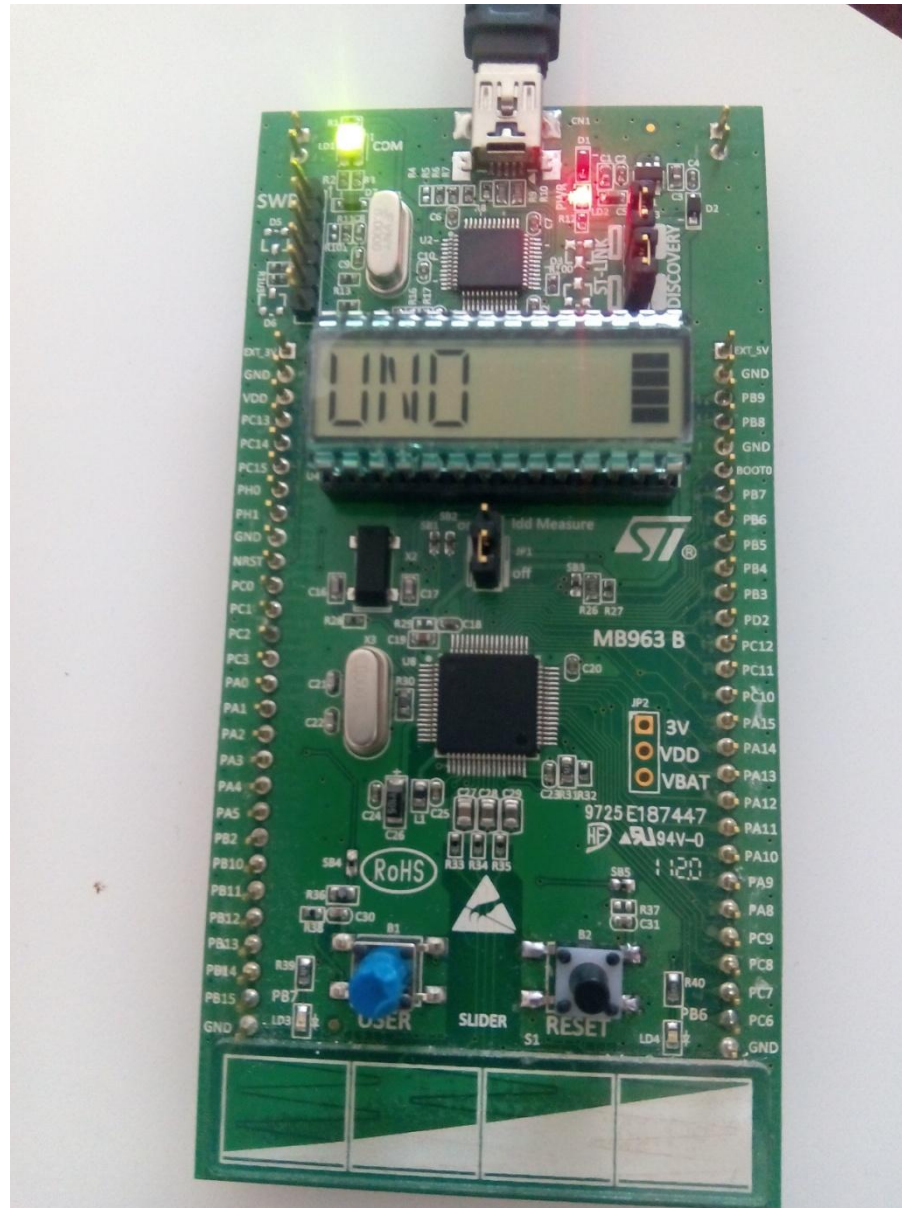
Depuración de un Proyecto

The screenshot displays the ST-Link Debugger interface with the following components:

- Registers Window:** Lists registers for the Core and System. Core registers include R0 through R15 (SP, LR, PC) and xPSR. System registers include Banked and System.
- Code Editor:** Shows the source code for `principal.c`. The current execution point is at line 10, which is the start of a `while (1) {` loop. The code includes `Utiles_SDM.h` and performs LCD operations and delays.
- GPIOB Window:** Shows the configuration for the GPIOB peripheral, including bits BS0 through BS11 and the LCKR register.
- Command Window:** Displays the execution command: `Running with Code Size Limit: 32K` and `Load "D:\JOSE\06_CARLOS_IIII\1_Curso_Academico_2015_2016\4_Segundo_Cuatrimestre..."`. It also shows the code size limit: `*** Restricted Version with 32768 Byte Code Size Limit` and `*** Currently used: 6824 Bytes (20%)`.
- Watch Window:** Shows the values of variables in memory. The variable `i` is of type `uchar` and its value is `<cannot evaluate>`. Other variables include `RCC` (pointer, 0x40023800), `CR` (unsigned int, 0x03030303), `ICSCR` (unsigned int, 0x005DD06B), `CFGR` (unsigned int, 0x0091000F), `CIR` (unsigned int, 0x00000000), `AHBRSTR` (unsigned int, 0x00000000), and `APB2RSTR` (unsigned int, 0x00000000).

The status bar at the bottom indicates the debugger is running on the ST-Link Debugger, with a time of 0.40413890 sec, L10 C1, and CAP NUM SCRL OVR R/W.

Prueba del Proyecto Explicado



10- Peculiaridades de la Programación en C en Microcontroladores

Programación de Microcontroladores en C

- El Lenguaje C utilizado es ANSI-C, es decir, no tiene de por sí, ninguna variación a cualquier otra plataforma.
- Sin embargo sí que es necesario mantener unas determinadas prácticas adicionales, para hacer una programación satisfactoria.
- La razón es que NO estamos programando una aplicación en un PC, donde los recursos se podrían considerar limitados.
 - Aquí se plantea una arquitectura con UNOS RECURSOS MUY LIMITADOS, tanto en cantidad de memoria, como en potencia de cálculo.
 - Además se trata de una arquitectura que no tiene por qué tener un teclado, una pantalla o una conexión a internet, por lo que el desarrollador tiene que tener muy presente la arquitectura con la que trabaja.
- En las siguientes transparencias se ilustran algunas recomendaciones.

Programación de Microcontroladores en C

- Usa el tamaño de variable que más se ajuste a tus necesidades.
 - Si una variable va a tener solo valores entre el 0 y el 5, usa un **unsigned char**, en lugar de un **int** (se pasa de usar 1 byte a 4 bytes).
 - Si la variable no va a tener valores negativos, usa **unsigned**, para limitar los valores (y evitar problemas de cálculo posteriores).
- Salvo que sea estrictamente necesario, no utilice asignación dinámica de memoria, sino estática.
 - La asignación dinámica de memoria implica:
 - Utilización de métodos adicionales como **malloc()** que consumen tiempo y recursos.
 - Que cada trozo de memoria asignado, debe tener también espacio de reserva para los punteros de asignación (cada puntero son 4 bytes).
 - Que la programación tenga que ser más cuidadosa para no desbordar la memoria o acceder a partes reservadas de la memoria.
 - En concreto, un control bastante elevado del uso de punteros.
 - Por ejemplo, si se va a usar una variable para el mensaje del LCD, y se sabe que dicho mensaje es como mucho de 6 caracteres, usa:
 - **unsigned char message[6];**
 - Y accede de forma estática al tercer carácter con **message[2];**

Programación de Microcontroladores en C

- No utilices funciones externas de las que desconozcas su verdadero funcionamiento, o que estén sobredimensionadas a sus necesidades.
 - Por ejemplo, se suele cometer el error de pensar que **printf (message)** va a mostrar el contenido de **message** por pantalla pero:
 - ¿Seguro que se va a mandar por pantalla?
 - No, se va a mandar por un puerto serie.
 - ¿Cuanta memoria ocupa utilizar **printf ()** ?
 - Muchísima más que la necesaria, ya que no sólo es el envío de caracteres, sino que también contiene la forma de formatear la cadena **message** (por ejemplo los parámetros %).
- Pasa los parámetros complejos por referencia, en lugar de por valor.
 - Al pasar los parámetros por referencia, sólo se copia el puntero al dato, pero el dato no se duplica en la función.
 - Sin embargo, al pasarlo por valor, el contenido del parámetro se copia como una nueva variable en la ejecución de la función.
 - Por defecto, el lenguaje C pasa todos los parámetros complejos por referencia, pero una mala programación puede forzar a pasarlos por valor.

Programación de Microcontroladores en C

- Utiliza siempre variables locales.
 - De esta forma, cuando se salga de la función, se destruye la variable y se libera la memoria.
 - Si crea una variable compleja dentro de una función, acuérdate de eliminarla coherentemente antes de salir de la función.
- No utilices cálculos de excesivo coste, cuando no es necesario. En concreto, intenta evitar cálculos con decimales.
 - La aritmética del microcontrolador es una aritmética entera, por lo que cualquier cálculo decimal supondría el tener que incluir en el código las rutinas correspondientes para hacer el cálculo en punto fijo o en punto flotante a partir de la ALU entera.
 - Esto conlleva un aumento enorme del tamaño del programa, así como de su tiempo de computación.
 - Por ejemplo, si en un problema va a trabajar con temperaturas, con precisión de un decimal, entre -20 y +50 grados, no trabajes con decimales, sino considera trabajar con enteros entre -200 y +500, que en este caso cabrían en una variable de tipo **short**.
- También recuerda que si usa aritmética entera $x/y = 0$ siempre que x sea menor que y , por lo que debería multiplicar x por algún valor antes de dividir.

Otras Recomendaciones

- Si hay parámetros que pueden llegar a cambiar de proyecto a proyecto, utiliza #define:
 - Los #define se ubican normalmente en la parte inicial de un fichero .c, o en los ficheros .h.
 - Al cambiar el valor, automáticamente se cambia en todos los sitios donde se haya utilizado el identificador utilizado en el #define.
 - Además el cambio se hace en momento de compilación, por lo que no implica un coste computacional.
 - Por ejemplo:

```
#define MAXIMO 500;  
  
...  
If (valor > MAXIMO) Error();  
Else Procesa(valor);  
  
...  
Porcentaje = valor*100/MAXIMO;
```

Particularidades del Keil para ARM

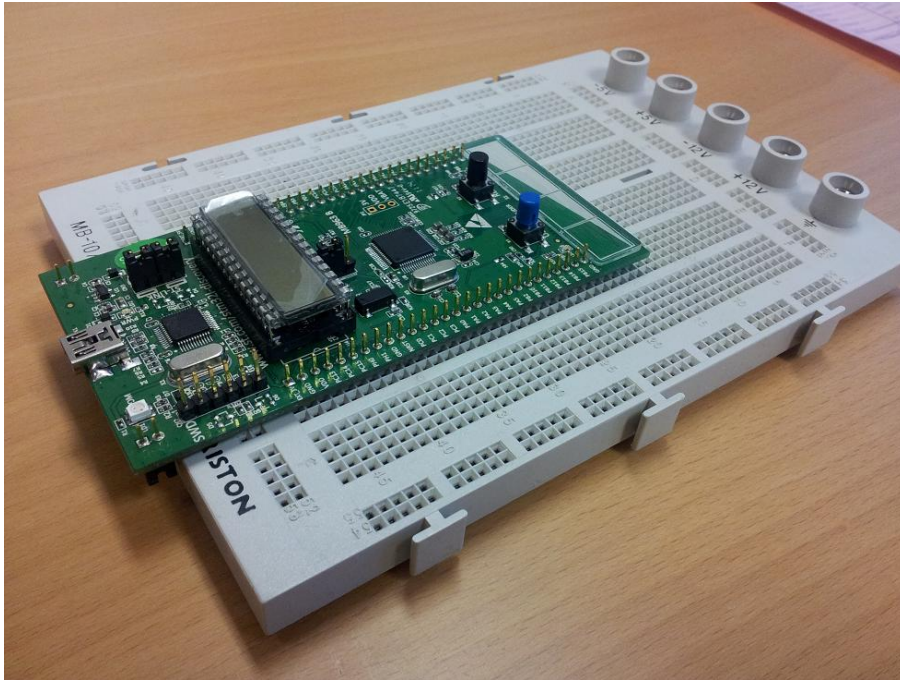
- Los registros ya se encuentran definidos en:
 - `stm32l152xb.h`
- Esta biblioteca ya se encuentra incluida en los proyectos, por lo que no hay que hacer nada adicional para usarla.

11 - Recomendaciones de Uso de la Placa de Desarrollo

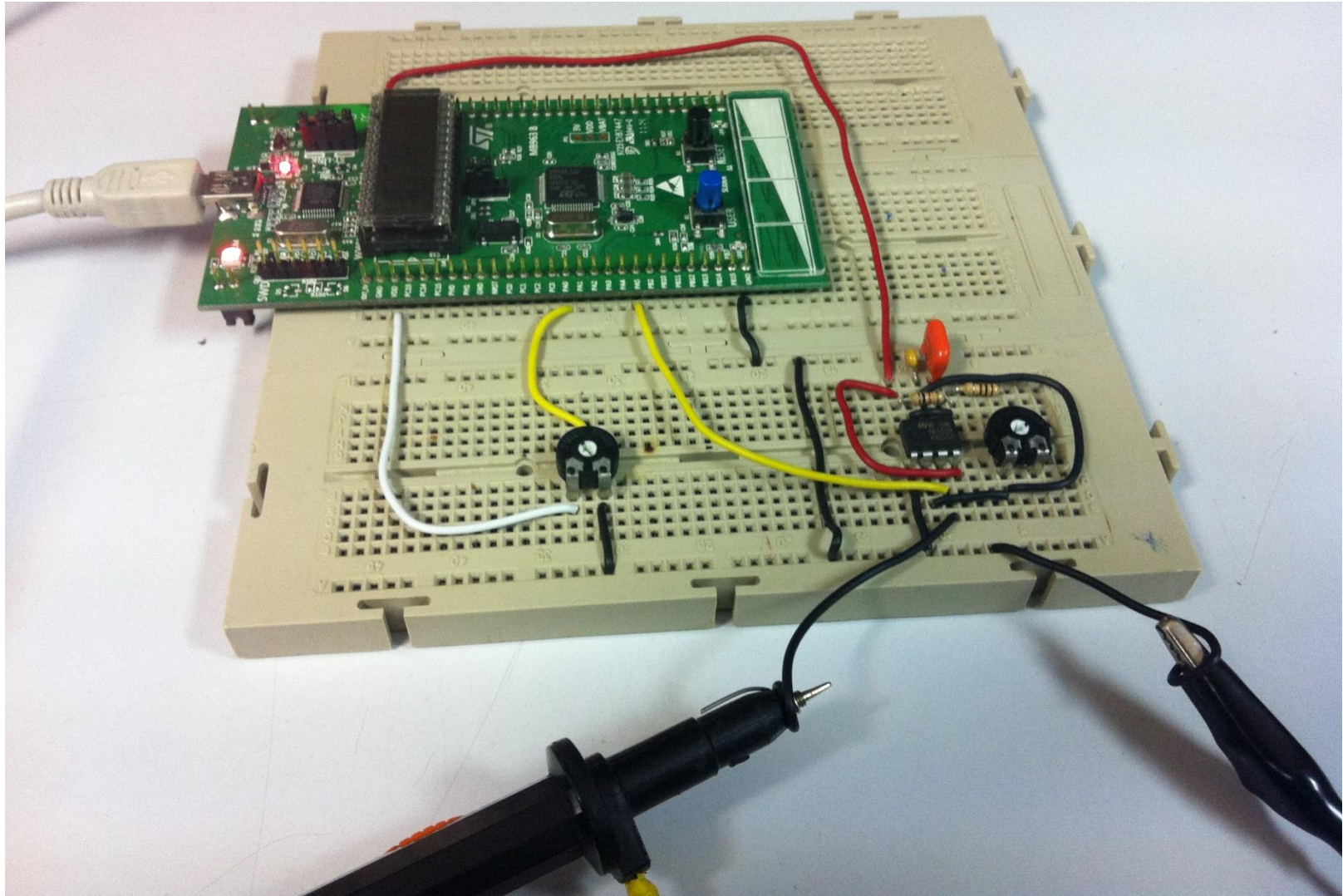
Recomendaciones para el uso de la placa

- Para poder utilizar mucho mejor la placa STM32L1-Discovery, es aconsejable:
 - Pincharla en una protoboard (o en un conjunto de ellas), de forma que los pines no se cortocircuiten y además dejen huecos para conectar los cables.
 - Meter el conjunto de la protoboard (la placa y las conexiones realizadas) en una caja para su transporte y así evitar que se suelten las conexiones.
- En la siguiente transparencia se puede ver el detalle de conexión, así como un ejemplo de uso.

Inserción de la placa en una protoboard



Ejemplo de Uso



12 - Ejercicios

Ejercicios Propuestos

1. Crea la función `espera(int ciclos)` que permita, mediante el parámetro `ciclos`, esperar un determinado tiempo correspondiente al número de ciclos. Utiliza para ello un bucle **for** (intenta ajustar el número de ciclos del bucle **for** para que cada espera sea de 1 segundo). Para ello, ten en cuenta:
 1. Que cada instrucción en ensamblador se ejecuta en un único ciclo de reloj.
 2. Que cada instrucción en C son varias instrucciones en ensamblador.
 3. Que el reloj está funcionando a 32MHz.
2. Crea el proyecto ejemplo del LCD explicado a lo largo de todo este tema, compílalo, depúralo y comprueba que funciona como se ha explicado en la transparencia 62.