

“Quick Start Guide” MUC STM32F4 Discovery

M.Sc. Ing. José Luis Ramirez Arias Ph.D.



Universidad Cooperativa
de Colombia

Descarga en instalación de software

- Descargar ST-Link

- http://www.st.com/content/st_com/en/products/embedded-software/development-tool-software/stsw-link009.html



- Descargar Keil ARM μ Vision 5

- <https://www.keil.com/demo/eval/arm.htm#/DOWNLLOAD>



MDK-ARM

Version 5.23 (February 2017)

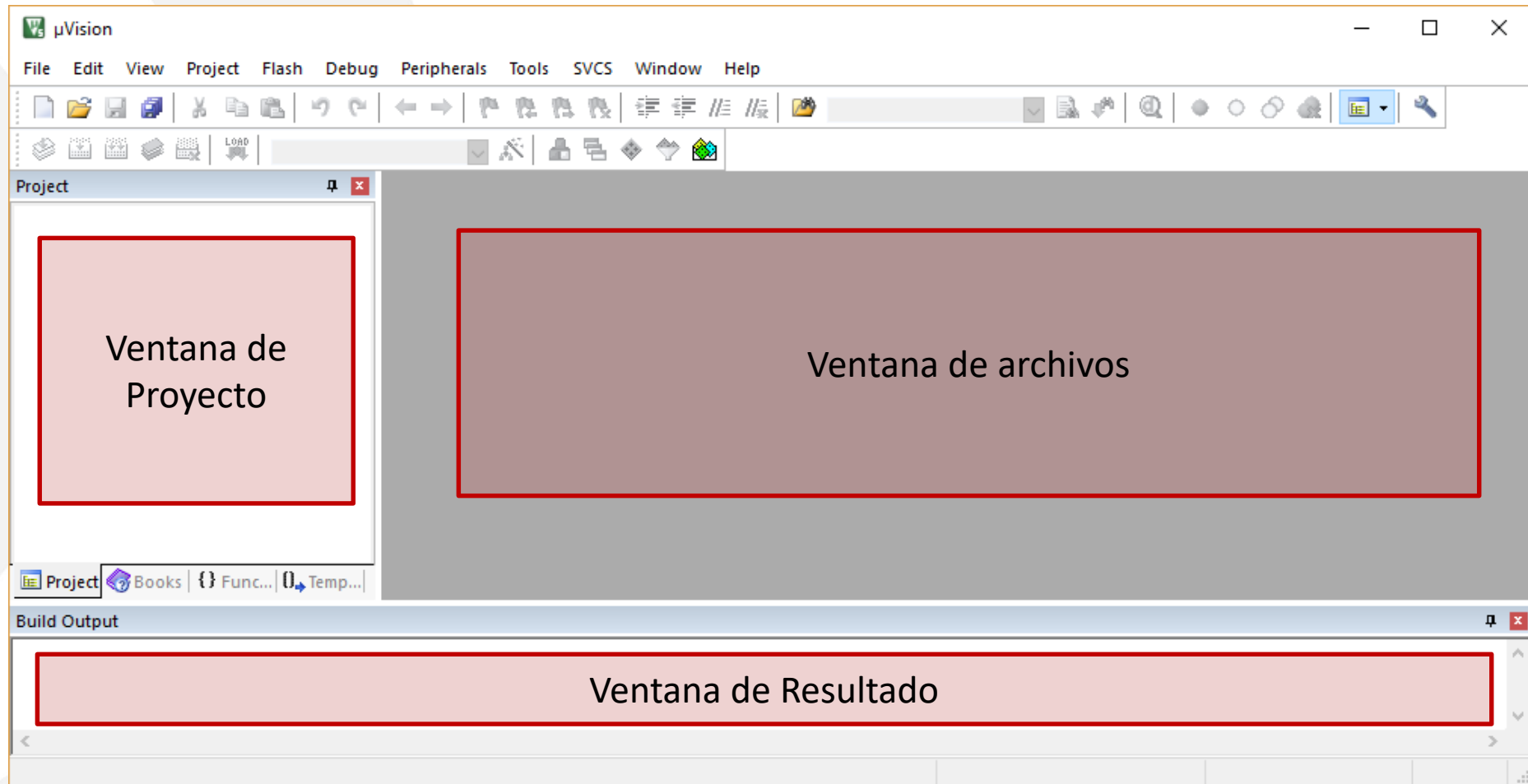
Development environment for Cortex and ARM devices.

LO MEJOR
DE TI
HACIENDO
LO MEJOR
POR LOS
DEMÁS



Universidad Cooperativa
de Colombia

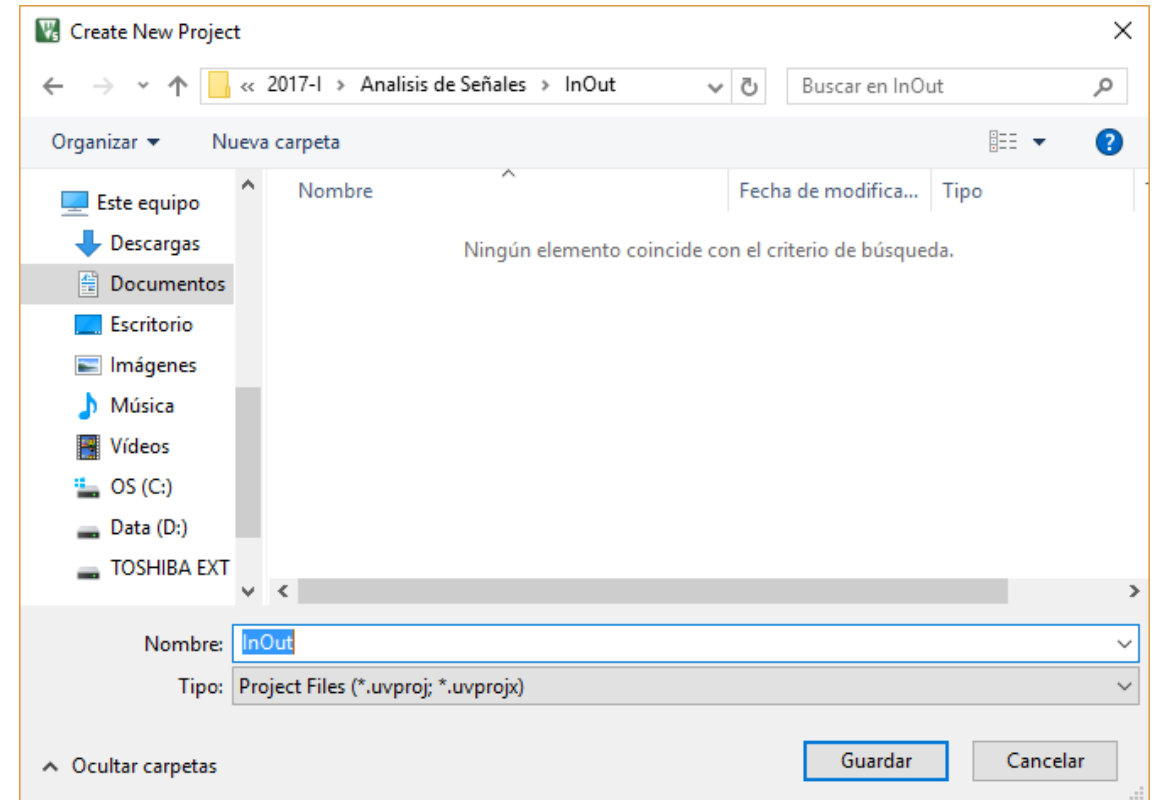
Interface Grafica Keil μ Vision 5



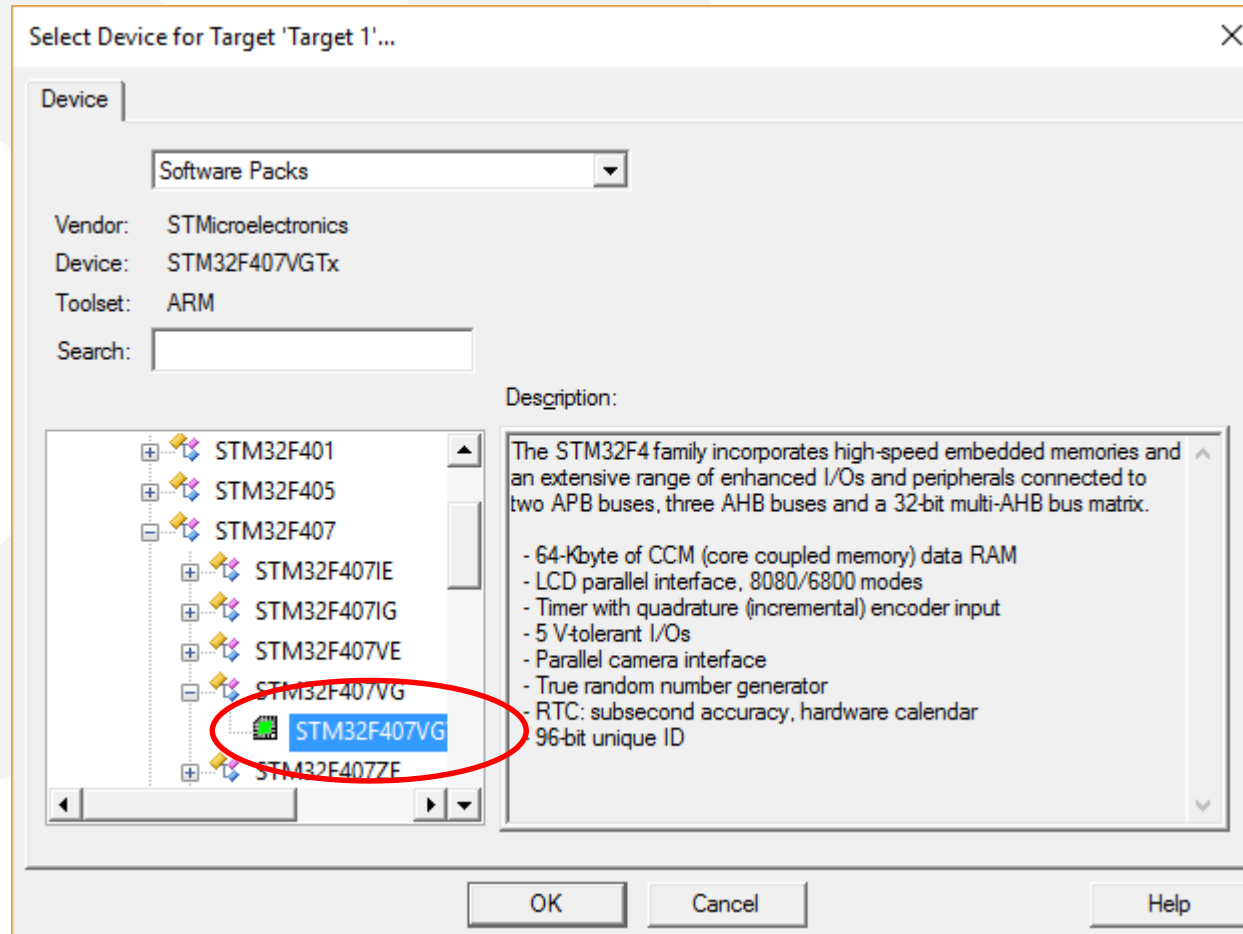
Creación de un nuevo proyecto

- Project->New μ Vision Project
- Seleccionar una carpeta o crearla
- Asignarle nombre al proyecto
- Guardar

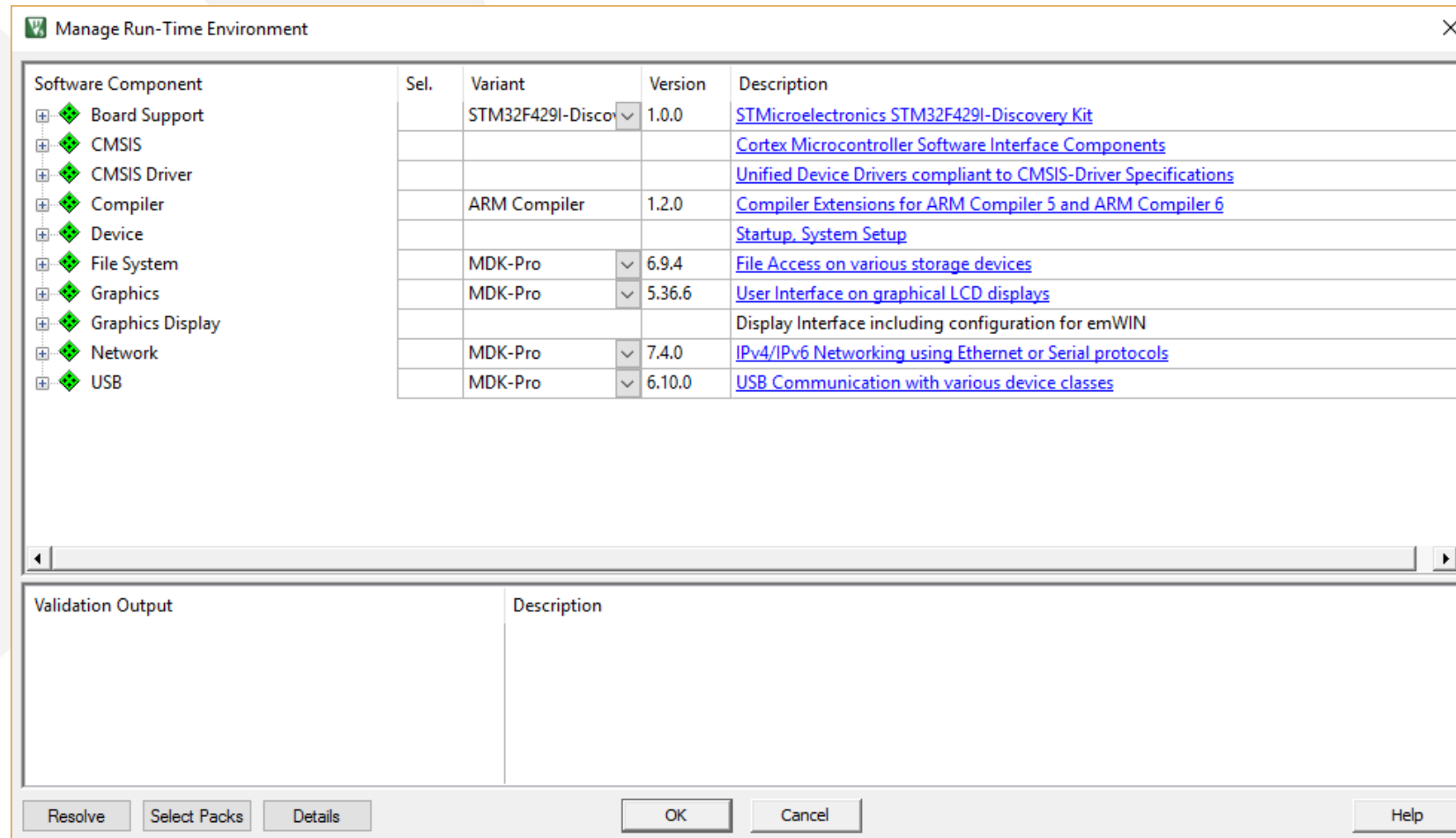
**Aparece cuadro de dialogo
“Select Device fot Target 1”**



Seleccionar Dispositivo



Administración del run-time environment

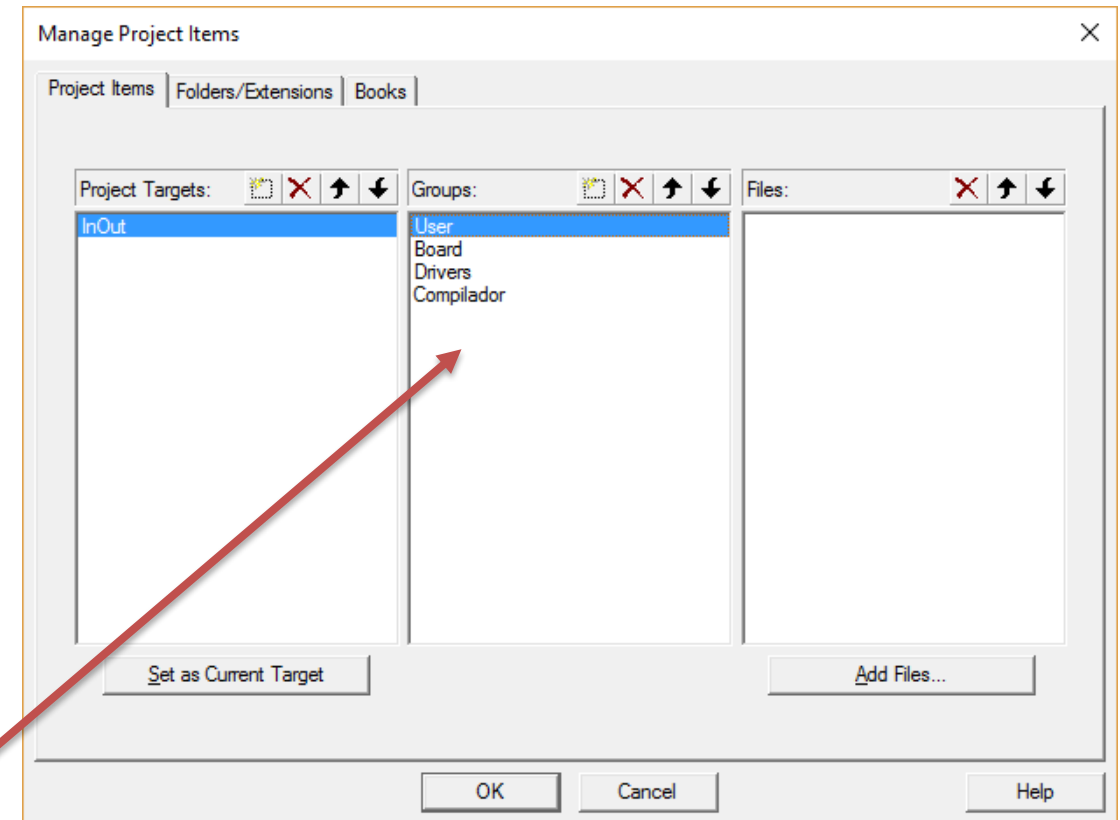


Crear Grupos de Código

- Click en el botón “file extensions”

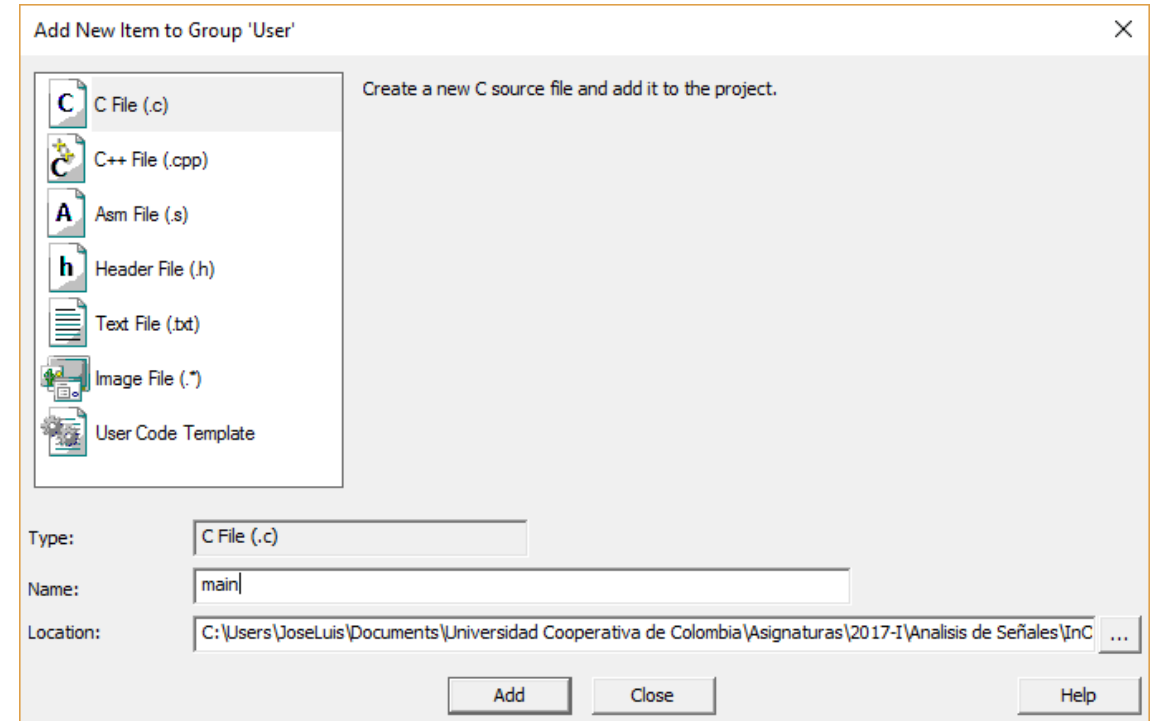


- Modificar el Nombre del targer
- Crear cuatro grupos: User, Board, Drivers y Compilador

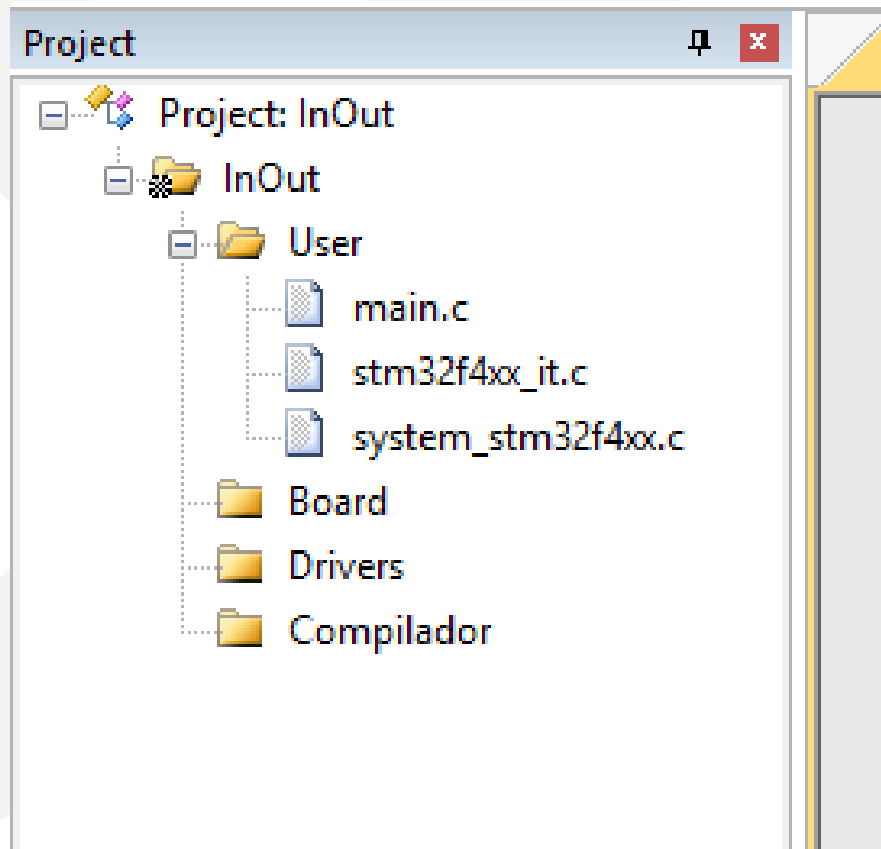


Archivo main.c

- Hacer click derecho sobre el grupo user
- Seleccionar Add new ítem
- Seleccionar C File (.c)
- En el campo Name escribir main
- Click en Add



Archivos requeridos

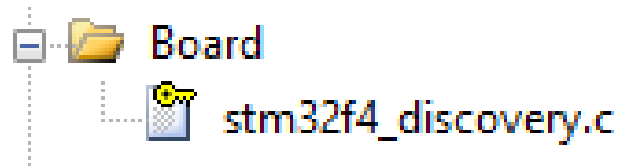


Nombre del archivo	Función
stm32f4xx_it.c	Manejo de excepciones e interrupciones
stm32f4xx_it.h	
system_stm32f4xx.c	Configuración del reloj
stm32f4xx_conf.h	Librería de configuración

Los archivos deben ser pegados en la misma carpeta donde se creo el archivo main.c

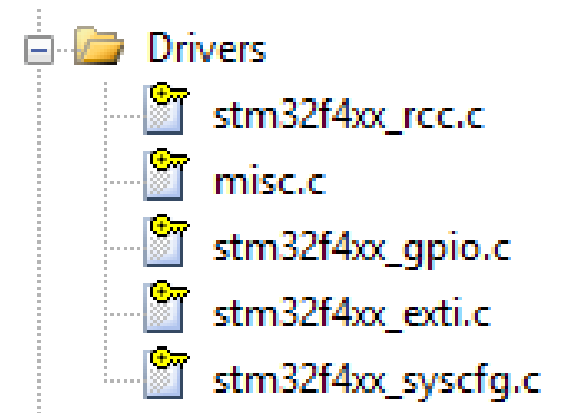
Grupo board

- Contiene un solo archivo stm32f4_discovery.c que contiene la información para manejar los leds y los botones incluidos
- El archivo esta ubicado en:
..\Drivers STM32F4 Discovery\Utilities\STM32F4-Discovery



Grupo Drivers

Nombre del archivo	Función	Ubicacion
stm32f4xx_rcc.c	Manejo de reloj e interrupciones	..\Drivers STM32F4 Discovery\Libraries\STM32F4xx_StdPeriph_Driver\src
misc.c	Manejo de interrupciones y prioridades	
stm32f4xx_gpio.c	Manejo de periféricos de entrada y salida, configuración de funciones alternativas	
stm32f4xx_exti.c	Manejo de interrupciones externas	
stm32f4xx_syscfg.c	Mapeo de memoria, conexión de interrupciones externas y configuración eth	

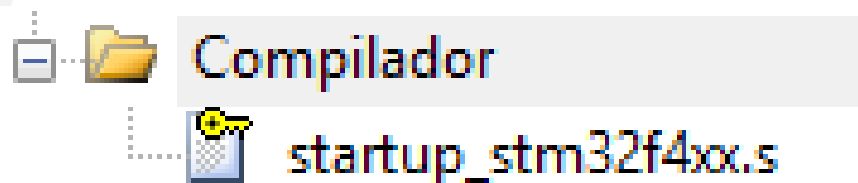


Grupo Compilador

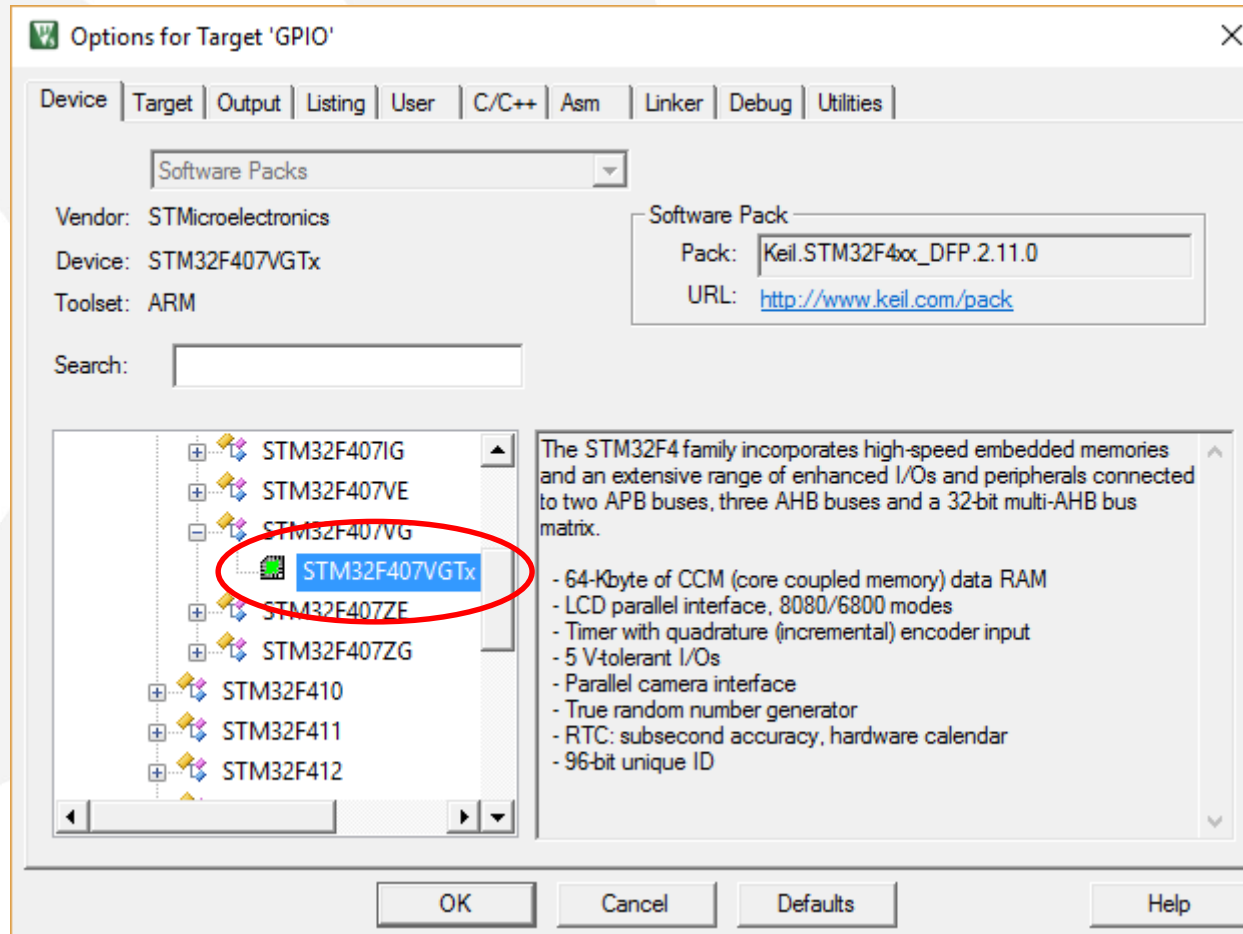
- Contiene un solo archivo startup_stm32f4xx.s que contiene la información para configuración inicial del dispositivo
- El archivo esta ubicado en:

..\Drivers STM32F4

Discovery\Libraries\CMSIS\ST\STM32F4xx\Source\Templates\arm



Options for target (Device)



LO MEJOR
DE TI
HACIENDO
LO MEJOR
POR LOS
DEMÁS

UC
Universidad Cooperativa
de Colombia

Options for target (Target)

Options for Target 'GPIO'

Device Target Output Listing User C/C++ Asm Linker Debug Utilities

STMicroelectronics STM32F407VGTx

Xtal (MHz): 12.0

Operating system: None

System Viewer File: STM32F40x.svd

☐ Use Custom File

Code Generation

ARM Compiler: Use default compiler version

☐ Use Cross-Module Optimization

☒ Use MicroLIB

☐ Big Endian

Floating Point Hardware: Single Precision

Read/Only Memory Areas

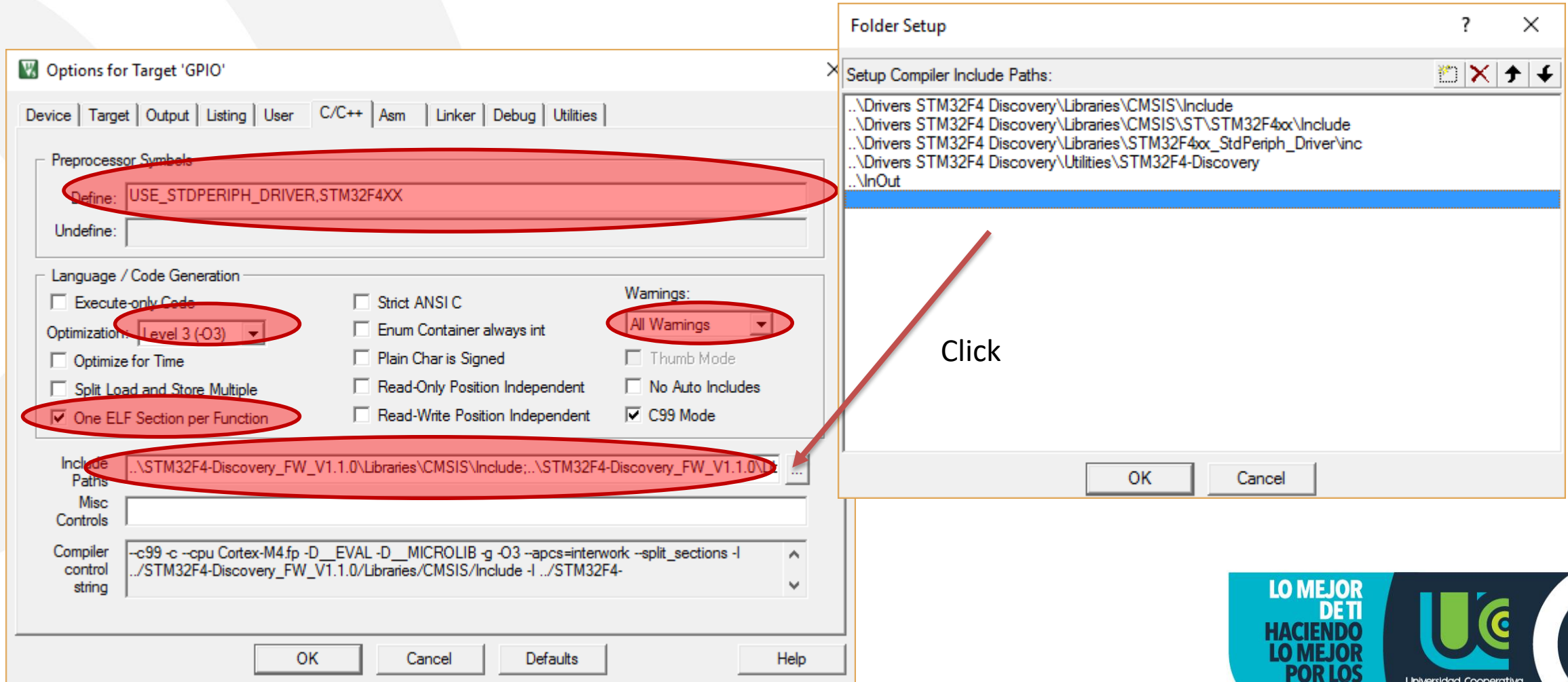
	default	off-chip	Start	Size	Startup
<input type="checkbox"/> ROM1:					<input type="radio"/>
<input type="checkbox"/> ROM2:					<input type="radio"/>
<input type="checkbox"/> ROM3:					<input type="radio"/>
on-chip					
<input checked="" type="checkbox"/> IROM1:			0x8000000	0x100000	<input checked="" type="radio"/>
<input type="checkbox"/> IROM2:					<input type="radio"/>

Read/Write Memory Areas

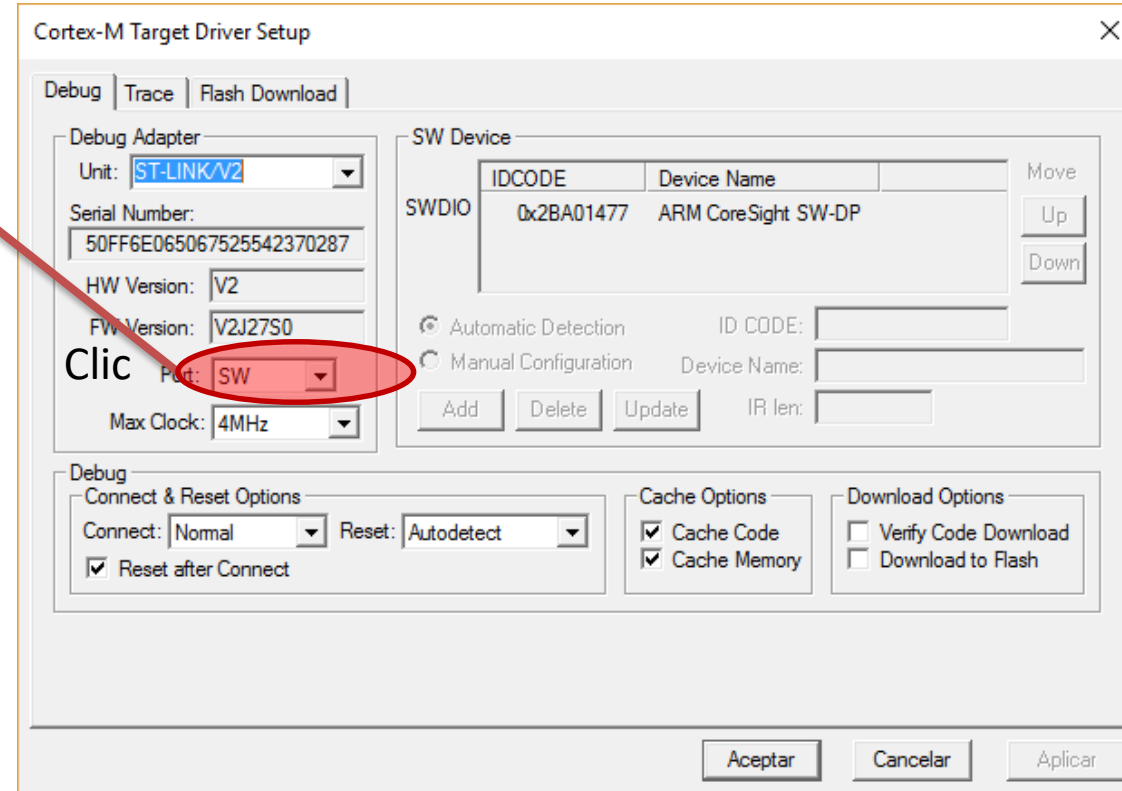
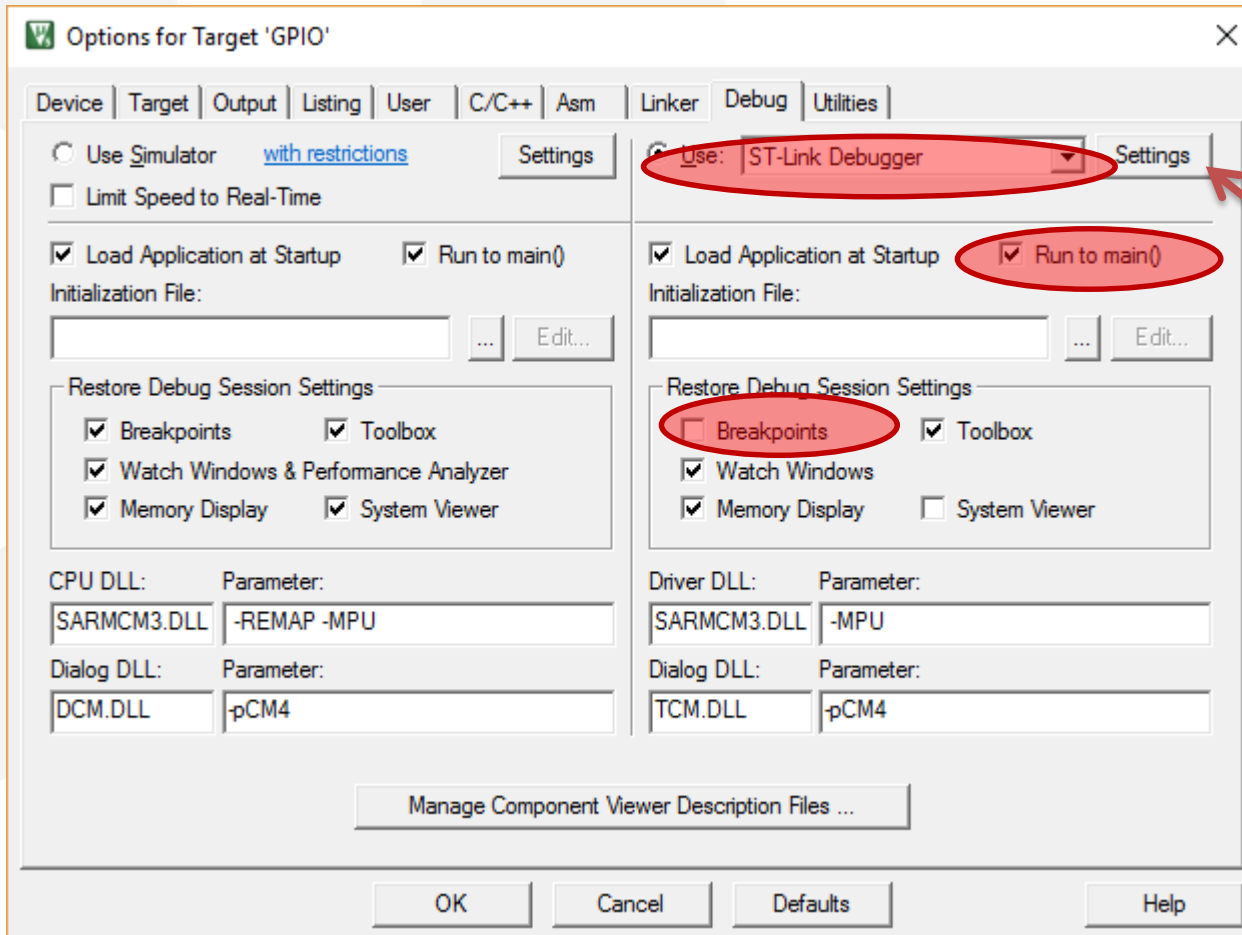
	default	off-chip	Start	Size	NoInit
<input type="checkbox"/> RAM1:					<input type="checkbox"/>
<input type="checkbox"/> RAM2:					<input type="checkbox"/>
<input type="checkbox"/> RAM3:					<input type="checkbox"/>
on-chip					
<input checked="" type="checkbox"/> IRAM1:			0x20000000	0x20000	<input type="checkbox"/>
<input type="checkbox"/> IRAM2:			0x10000000	0x10000	<input type="checkbox"/>

OK Cancel Defaults Help

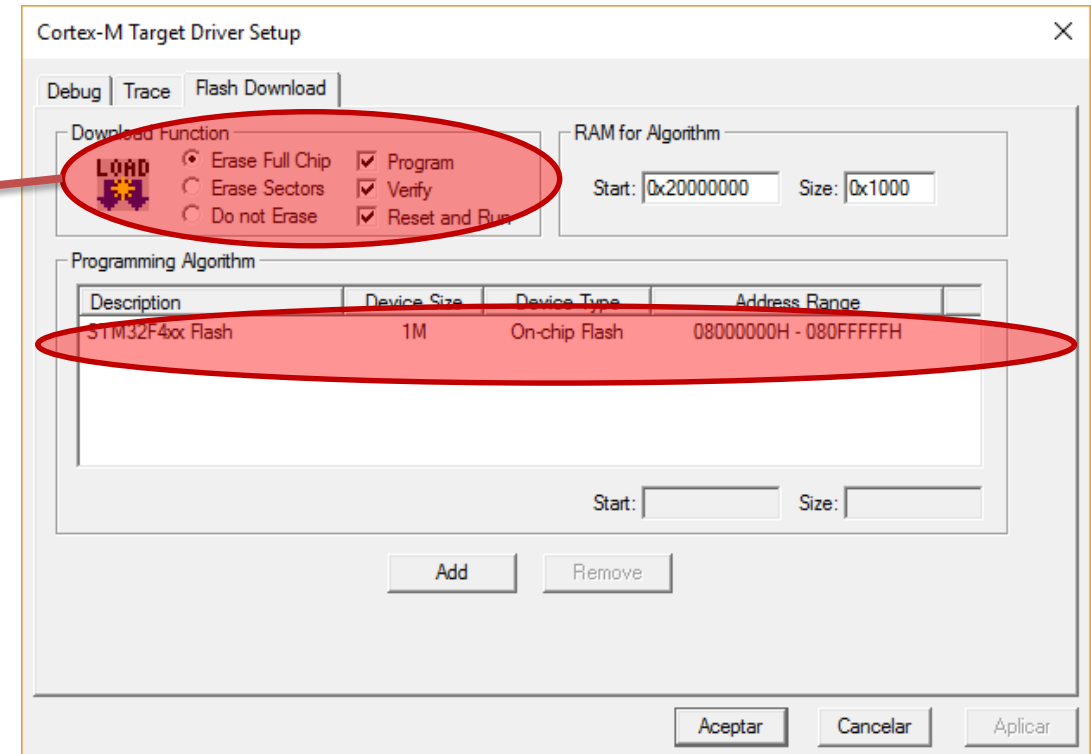
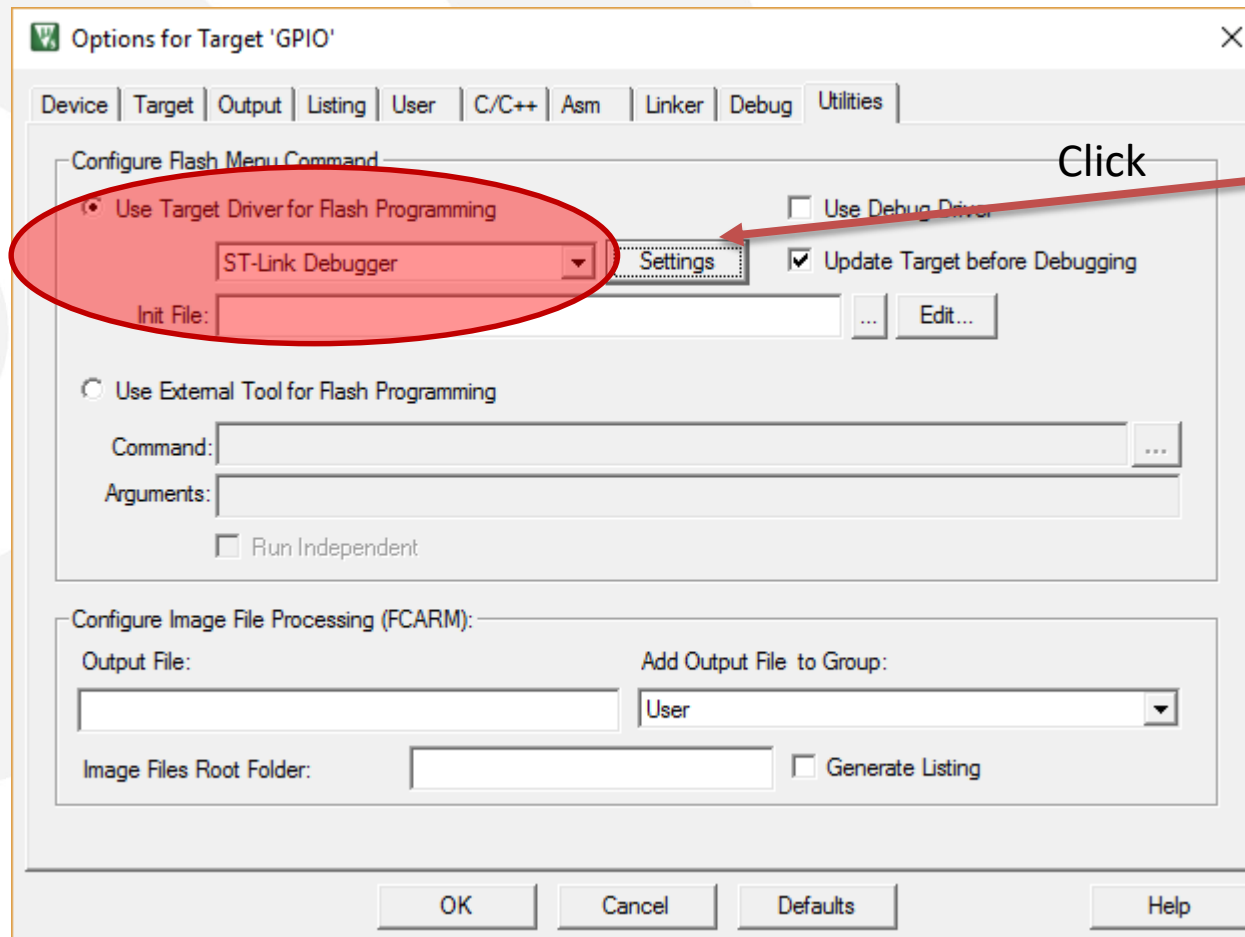
Options for target (C/C++)



Options for target (Debug)



Options for target (Utilities)



Código Fuente

```
#include "stm32f4_discovery.h"

GPIO_InitTypeDef GPIO_InitStructure;

void Delay(__IO uint32_t nCount);

int main(void)
{
    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOID, ENABLE);

    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_12 | GPIO_Pin_13 | GPIO_Pin_14 | GPIO_Pin_15;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
    GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_100MHz;
    GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_NOPULL;
    GPIO_Init(GPIOID, &GPIO_InitStructure);

    while (1)
    {
        /* PD12 to be toggled */
        GPIO_ResetBits(GPIOID, GPIO_Pin_15);
        GPIO_SetBits(GPIOID, GPIO_Pin_12);

        /* Insert delay */
        Delay(0x3FFFFFF);
    }
```

```
GPIO_ResetBits(GPIOID, GPIO_Pin_12);
GPIO_SetBits(GPIOID, GPIO_Pin_13);
```

```
Delay(0x3FFFFFF);
```

```
GPIO_ResetBits(GPIOID, GPIO_Pin_13);
GPIO_SetBits(GPIOID, GPIO_Pin_14);
```

```
Delay(0x3FFFFFF);
```

```
GPIO_ResetBits(GPIOID, GPIO_Pin_14);
GPIO_SetBits(GPIOID, GPIO_Pin_15);
```

```
Delay(0x7FFFFFF);
```

```
GPIO_ResetBits(GPIOID, GPIO_Pin_12 | GPIO_Pin_13 | GPIO_Pin_14 | GPIO_Pin_15);
```

```
Delay(0xFFFFF);
}
}
```

```
void Delay(__IO uint32_t nCount)
{
    while(nCount--)
    {
    }
}
```