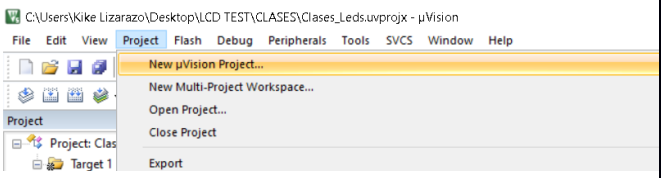
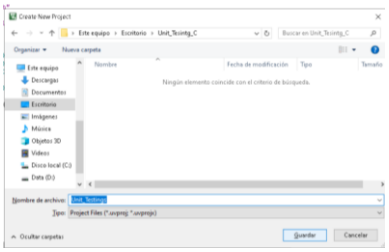
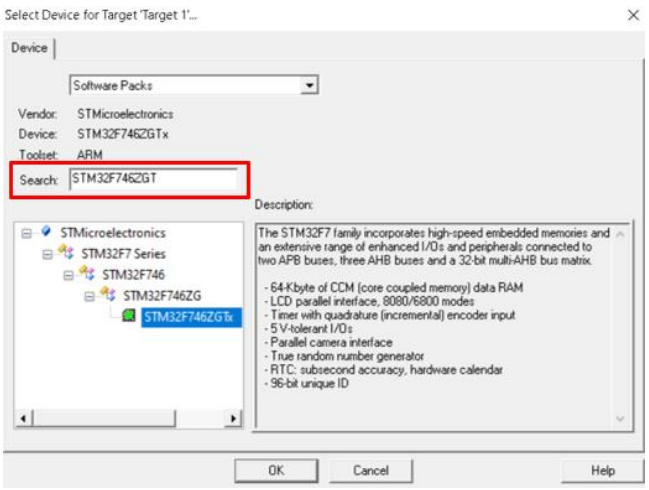
Unit Test Procedure and Report (Creation Project & Classes)

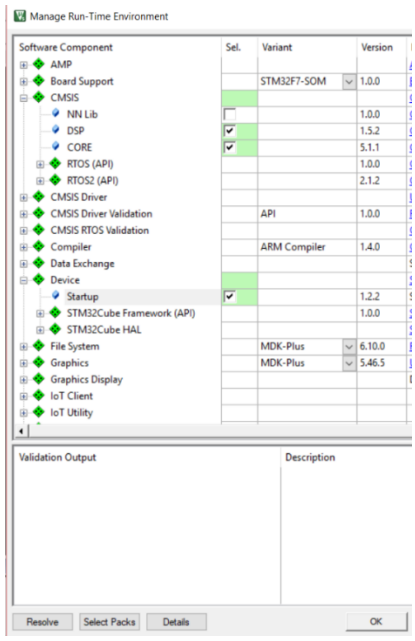
Hardware Configuration	Microcontroller test (Flat configuration)
Software Configuration	Local Main
Test Input	Creation and developing of classes and functions in a new project.
Requirements	Keil uVision 5 , STM32F07VG Board
Instructions	- Connect the STM32F7 to a computer using the micro USB port. - For testing make sure that all variables had been initialized as you want. - Clean and check the variables you want to follow using the Debugger. - Build and program.
Special Notes	The test result for the creation project & classes unit test is to be able to create form 0 a new project with the necessary configuration to work with a STM32F07VG Board. Also it is to create the project using classes distribution in order to achieve the correct results for operations not specified in the root code or basic C libraries by creating them with new functions.
Expected test results	Make the program compile an external class created to provide order to the code.
Test Engineer	David Enrique Lizarazo Vesga
QA	Robinson Jiménez Moreno

Instructions	Notes	Log	Test	QA
Power on and connect			X	
Run unit test	Classes		X	
Previous Instructions		Build time is: Feb 24 2019 **Create a New Keil project as shown below:** Go to the rut: Project -> New uVision Project…  Select a folder and give it a name.  Now you would have to choose the Device, it is	X	

better to use the search tool. As known the microcontroller is the: “STM32F746ZGT”.



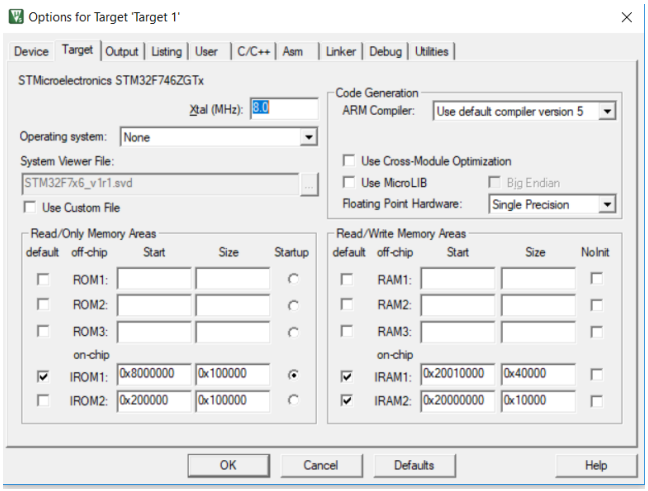
To use the debug system and other properties, you should set the next configuration in the Pop-up window

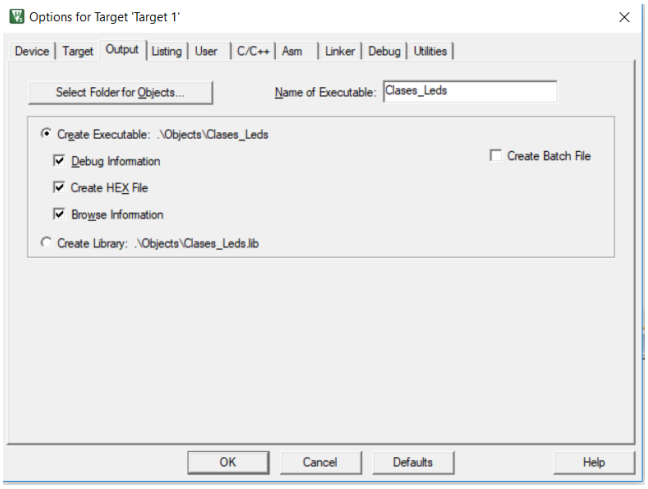


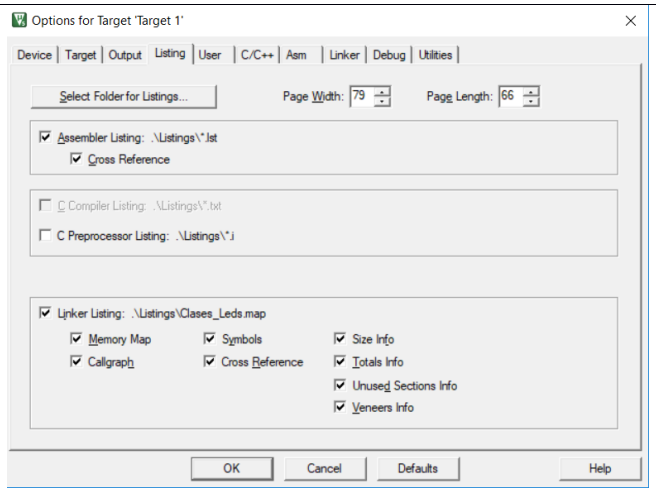
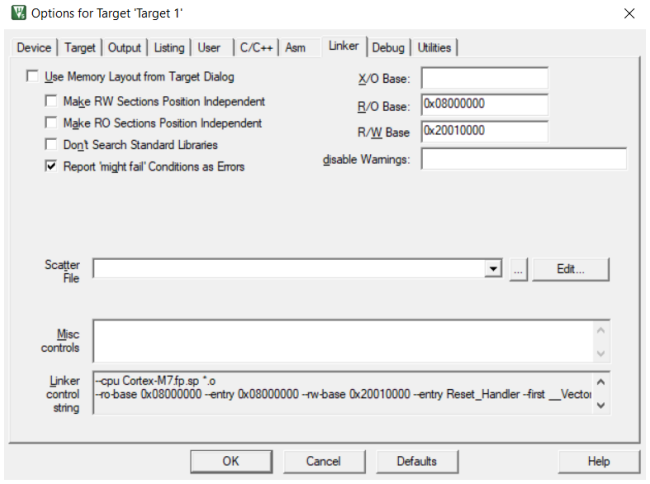
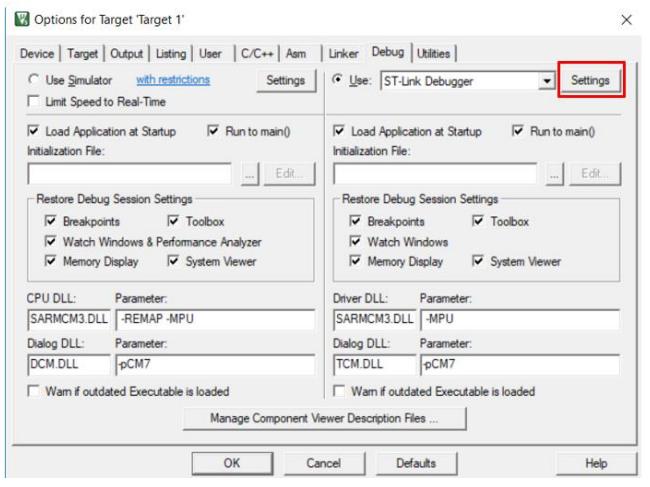
After that you have created the project. Then you would need to set the parameters for the microcontroller to work.

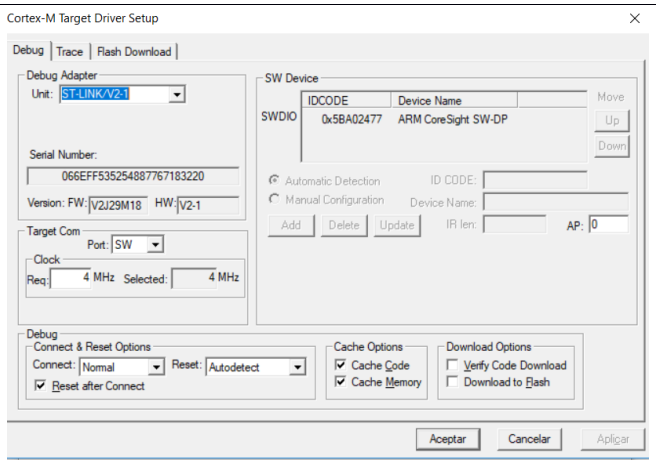
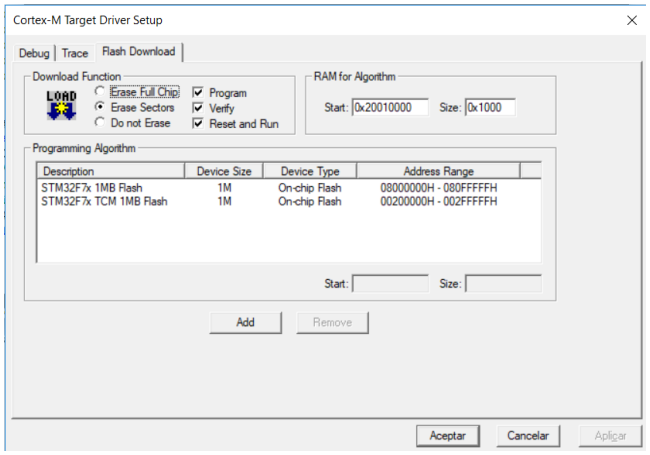
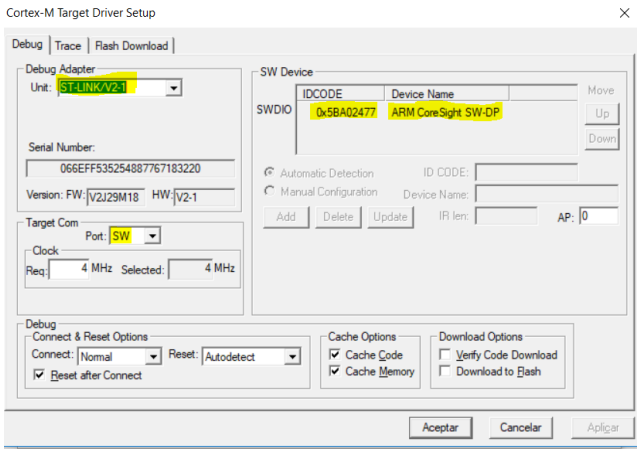
Open the **Options for target.** You could do this by clicking the icon shown in the picture bellow, or just by right clicking the folder “Target 1” created by default and clicking in “Options for target”.

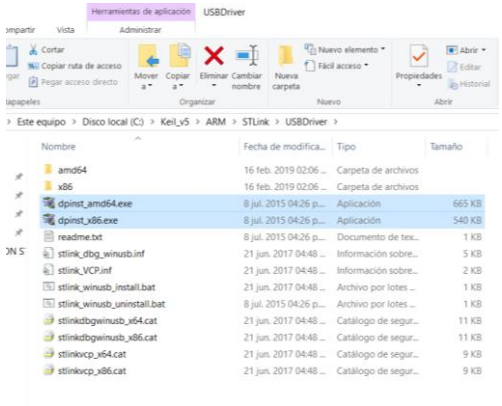
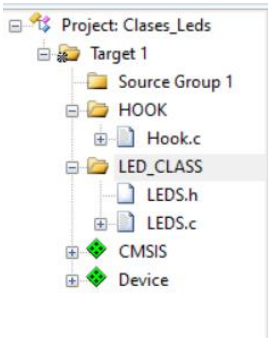
Once in it, you should do the next changes shown in the pictures bellow.





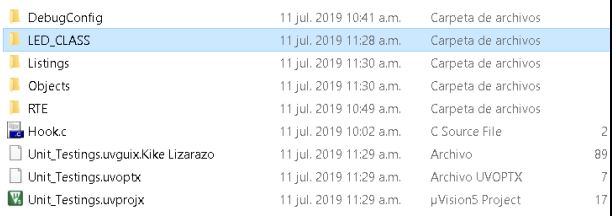
		   Open the settings for Debug and check the configuration		

		 		
Possible problems		If after you create the project the program does not recognize the microcontroller, please try following the follow steps. First, you will need to know if it is a compiler problem. To do so go to the “Options for target” as show previously. Then go to the Debug Settings, in the Debug tag you could see something like this:  If you don’t, then follow the next steps		

		Go to the next folder in your computer: C:\Keil_v5\ARM\STLink\USBDriver.  Then execute the drivers: dpinst_amd64.exe dpinst_x86.exe If problem, persist then try restarting the computer to update the new drivers installed.		
Previous instructions		Build time is: Feb 24 2019 **Add the project files:** The next step is to add the main file and the headers for the classes. You can add the files by clicking in the target and add an existing file. Or create a new one. As an advice, and in view of the future number of classes to maintain the order and co-working in the project, iits recommended to add folders to contain the header and source files of every class. Just as shown in the example:  • If you add an existing file, you should be sure that the file is in the project file, this for the compiler to find it. Another recommendation in case you do so, is to add the path to the		

project in case you want to order your folder. Go to “Option for target” , and in the C/C++ file , go to the “Include Path” option. Using it, you could specify where to find the headers and source files that are not contain in the same folder with the project.

For the example, we will have the next distribution in the project folder:



As you can see, we have created a folder to contain the header and source files. You would have to do this in the folder and into the program, this because the Groups you crate in Keil does not affect the folders.

To add your classes files, just need to include the header in the beginning of the main code as shown below.

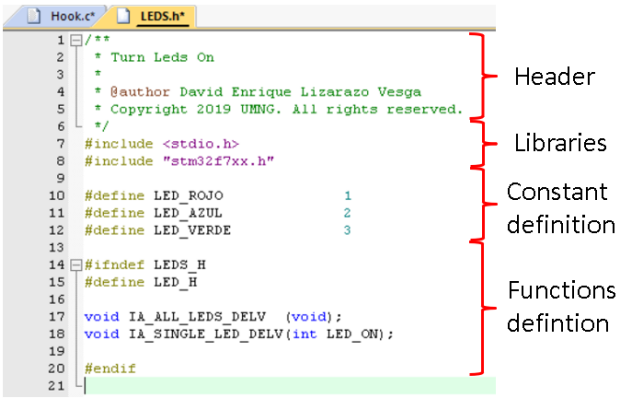
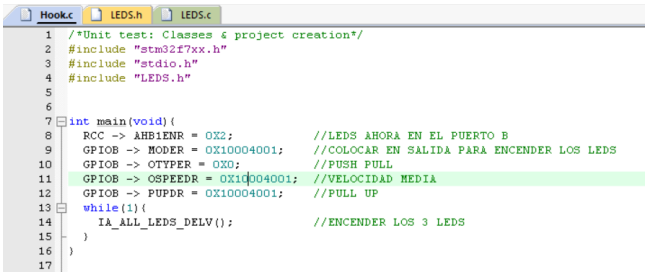
Hook.c*

```
1  /*Unit test: Classes & project creation*/
2  #include "stm32f7xx.h"
3  #include "stdio.h"
4  #include "LED_CLASS/LEDS.h"
```

Note that in case you add the path using the Options for target, you could skip the location of the header file.

Hook.c*

```
1  /*Unit test: Classes & project creation*/
2  #include "stm32f7xx.h"
3  #include "stdio.h"
4  #include "LEDS.h"
```

Previous instructions		Build time is: Feb 24 2019 **Files Structure:** After creating the files and including them into the main code, we have decided to use an structure to name and distribute the functions. In this example would be shown how:  Functions: `void IA_ALL_LEDS_DELV (void);` The functions have been created following some rules. Start the name with IA (from Artificial intelligence), next will be the name or description of the function, and ends with the acronym of the programmer who did the function.			
Record test results		Build time is: Oct 24, 2018 11:10 The test consists in build and download the program into the microcontroller, using the classes to turn the leds On. The final code to do so, would look like this:  As you can see, inside the While loop would be the calling for the function that turn the leds on.	X	

		So, after building and downloading the code in the microcontroller, it give the results shown below. 		
Possible Errors		One possible error that could happen in the moment of building the project, is the moment when that function is called. **The error : L6218E Undefined Symbol.** This Linker error could happen for multiple reasons. If you get this error, make sure that all the files (Headers and source files) are contained in the project structure, and that all the paths had been added or that the correct folders had been chosen. If the error persist and the source file where the function that gives the mistake is a C file (extension .c) change it to C++ (extension .cpp).		