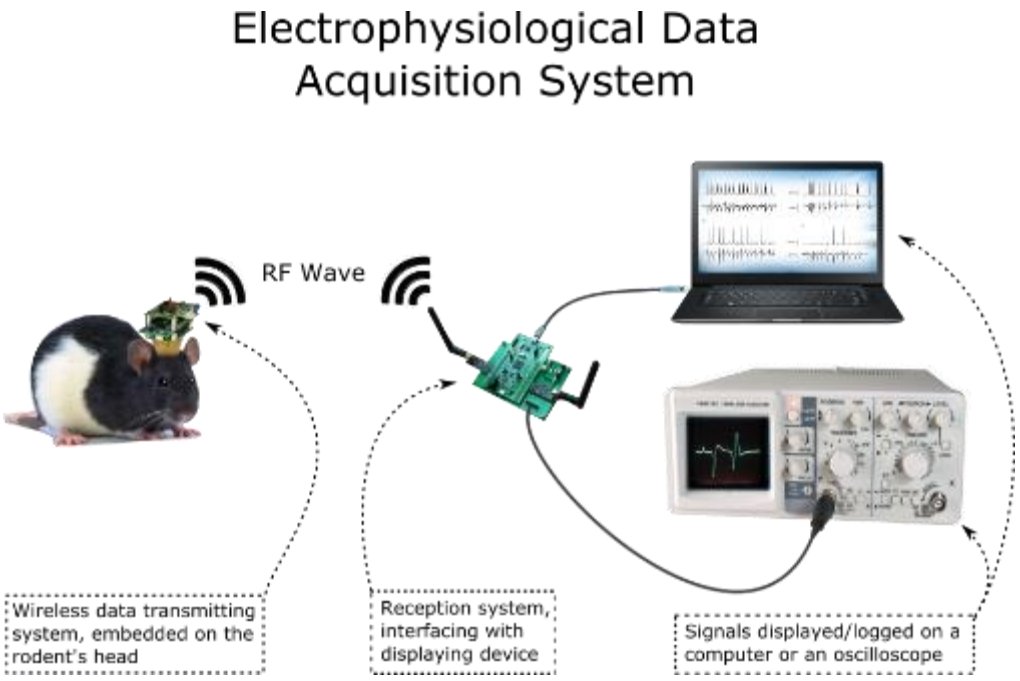
User manual step by step and explanations

This file will reunite all the instructions to build the hardware part of the rat embedded system and of the interfacing system. We will describe as well how to download the codes in them.

Quick description of the system

The embedded emitting system connected to the implant of the rat, will digitalize the neural signals with an analog to digital converter, compress data, before sending them wirelessly to the second system. This latter, after reception, will decompress data, converted them into USB packets and finally transmit them to the computer for display and recording.



Summary

I) Building the neural recording System
- How to create, the rat head stage PCB
  - Display it on Eagle
  - Order it from a PCB creating company on internet
  - Solder the components
- How to create the Interfacing system
  - Solder components
- How to program the system
  - Setup of Keil (software)
  - Setup of ST link V2
  - Program the two microcontrollers
- How to test the system

II) Explanation about the System
- General explanation about the system
- Explanations about the Code

Glossary:

MCU: microcontroller
RHD: Rhd2132 the analog to digital module that will convert the neural signal of the rat
NRF: Nrf24L01+ the transceiver module used to communicate wirelessly
PCB: Printed circuit board
Embedded system: the system that will be carried by the rat on its head
Interfacing system: the system that will do the interface between the embedded system and the computer

Building the neural recording System

How to create, the rat head stage PCB

Display on Eagle

First, we have to download Eagle, a software from Cadsoft from their website. The free version, is enough. We take the last version, corresponding with our OS (windows 7, 64 bits in our case)

http://www.cadsoftusa.com/download-eagle

Now, it's time to install it.

Now download the file: [lien]
Since the Embedded system is composed of 3 sub-systems, we have 3 different eagle files/folders. We open one of the, for instance [blab la bla.egl].

We can follow this [link tutorial] tutorial to increase our understanding and be able to navigate between files. As we open it, three windows will be displayed one for the project manager, one for the schematics of the circuit and one for the actual representation and design of the circuit.

Order PCB's

Since we only want to order printed circuit and not design it, we won't say more about eagle. Many tutorial are available on internet.

To order the printed circuits, we need to pick on PCB manufacturer available on internet and which accept PCB layout file under eagle file format. For the following of this tutorial, we will use PCB-Pool printed circuit manufacturer: [pcb pool]

Again, since the embedded system is composed of 3 PCB, we have to do 3 orders. On the PCB pol website (the English version), we go to "Order" ➔ "PCB's". Then we fill the formulary. To reduce the price you can unselect some option, but we advise to use the option as shown below. Also you can select another manufacturer to reduce the costs. The solution we give is just a working example.

1. Technology

Rigid printed circuit board [?]	○ 1 layer ◉ 2 layers ○ 4 layers ○ 6 layers
Flexible printed circuit board [?]	○ 1 layer ○ 2 layers
Aluminium core [?]	○ 1 layer
Other technology [?]	PCB-Overseas price query

2. Quantity 10

We will calculate the best price for you: The area of your chosen project is smaller than 1.00 dm². You will receive a total of **25** printed circuit boards

3. Dimensions [?] Length in mm: 20 Width in mm: 20

3a. Panel
- ◉ Best possible use of the smallest area.
- ○ Online panel configuration

4. Base material [?]
- ◉ FR4, 35 µm Cu, 1,6 mm
- ○ FR4, 35 µm Cu, 1,0 mm

5. Soldermask [?] ◉ yes ○ no

6. Silkscreen [?] ○ yes, only top ○ yes, only bottom ○ yes, top and bottom ○ no

7. Surface [?] ENIG (Electroless Nickel Gold) for ultra-flat pads

8. Layout specifications [?]

Min. track / gap size:
- ◉ >= 0.125 mm (5 mil)
- ○ >= 0.150 mm (6 mil)

Min. drill-end diameter:
- ◉ >= 0.2 mm (8 mil)
- ○ >= 0.3 mm (12 mil)

9. Overdelivery [?] ◉ yes, if available ○ no ○ yes, if available as Magic-PCB® - with free embedded RFID chip (Important information)

10. Magic-PCB [?] ◉ no ○ yes, no connection ○ yes, electrical connection

11. E Test [?] ◉ yes ○ no

12. Delivery time in WD [?] ○ 1 ○ 2 ○ 3 ○ 4 ○ **5** ○ 7 ◉ **8**

13. File format [?] EAGLE ▼

14. Project name Eagle headstage RHD

15. Free Stencil [?] ◉ Yes I want one. ○ no

16. Assembly [?] ○ yes ◉ no

17. Series production? Are you also interested in a quotation for a larger number of units?
- ☐ Yes, please offer

Your PCB

Required number: 10

	Net EUR	Gross EUR
Price per panel	8.30	10.21
Total order value	**83.03**	**102.12**

... more for less!

Components list
Be sure to order every components needed, which are listed [here] in their correct package.

Soldering

Soldering is the most delicate part of building the system. Many technics are usable such as using a soldering iron, a hot air soldering station, or a reflow oven. In our case we soldered every component except the RHD2132 and the NRF24L01+ whit a soldering iron. For the later component, we used a hot air soldering station. About soldering, many tutorials exists as well (youtube.com).
The advices we can give are the following:

- Use a microscope or a good lens
- Set first the components intended to be soldered by hot air
- Then set first the most complicated to solder components (microcontroller, Led, connectors)

How to create the interface system

The interface system is centered on a STMicroelectronics development board, the STM32F411 development board:

http://www.st.com/web/catalog/tools/FM116/SC959/SS1532/LN1848/PF260946

Follow this list of components to create the interfacing system

[List]

And solder them following that schematic.

[Schematic]
There is no particular difficulties for soldering these components.

How to program the system

This section will speak about how to set up the system. We won't approach the understanding of the code and the utilization of Keil the development tool used to develop the code and debug the system. We will see how to compile the code and program both MCU's.

First we need to install the last version of Keil:

- Let's go to: http://www.keil.com/

- Then: "product download" ➔ "MDK-ARM"

- Then: we complete the formulary and download the software compatible with our Operating system, in our case, windows 7, and 64 bits.

- Then we install it.

- Doing this quick tutorial **will help you a lot** about working with Keil and the microcontroller we want to program:
  http://www.keil.com/appnotes/files/apnt_261.pdf

Now it's time to download the code from internet, open it in Keil and compile it.
Since we have two systems: the rat embedded one and the interfacing one, there is two projects under Keil. One project to create the code for each microcontroller. We want to compile the code of each project one at time, compile it, and download it to the corresponding microcontroller.

For that:
We go to: https://github.com/pseudoincorrect/Electrophy_Base_System
And click on "download zip".

Then we go to: https://github.com/pseudoincorrect/Electrophy_Embedded_System
And click on "download zip" as well.

Keil project setting up

On each folder (after unzipping), we navigate to: Base_System or Embedded_System ➔ Projects ➔ MDK-ARM ➔ and open the .uvprojx file (either Embedded_System.uvprojx or Base_System.uvprojx)

This will open the corresponding Keil project. Since we presume it's the first we open Keil, some packages are needed to be added in order to compile properly and to download the code to the MCU.

After open, Keil will ask to choose either:
"Migrate to Device Pack" or "Install Legacy Support", we choose the first one, "Migrate to Device Pack".

Then depending which project we opened we will follow the appropriate one

For the INTERFACING SYSTEM

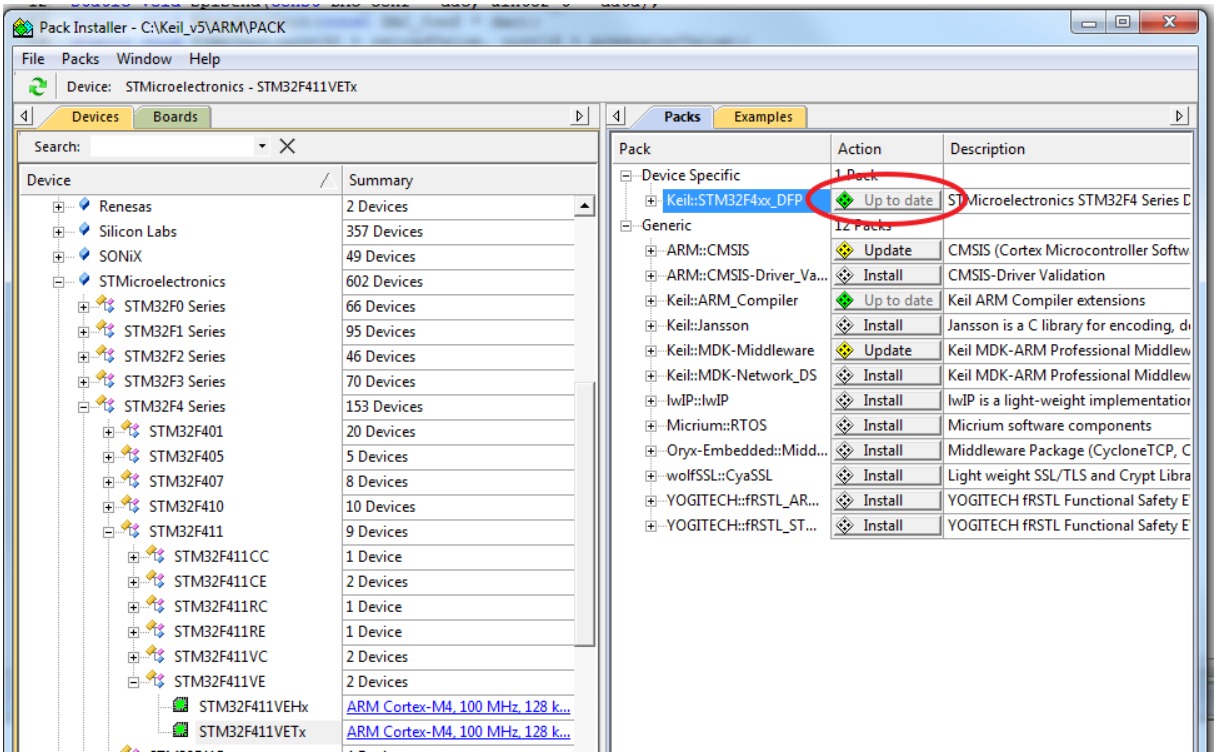
If the "pack installer" is not opened, we open it by clicking on "Project" ➔ "Manage" ➔ "Pack installer"

Base_System.uvprojx

Go to device TAB
Go to:
"STMincroelectronics" ➔
"STM32F4 Series" ➔
"STM32F411VE" ➔
Click on "STM32F411VETx"

Go to Packs TAB
Install:
"Keil::STM32F4xx_DFP"

As shown bellow



After the packs installation, close the packs manager window, and go to Option for targets:

On device tab, select:
“STMincroelectronics” ➔
“STM32F4 Series” ➔
“STM32F411” ➔
“STM32F411VE”
“**STM32F411VETx**”

On C/C++ Tab, in the box “define”, write: “USE_HAL_DRIVER,STM32F411xx”

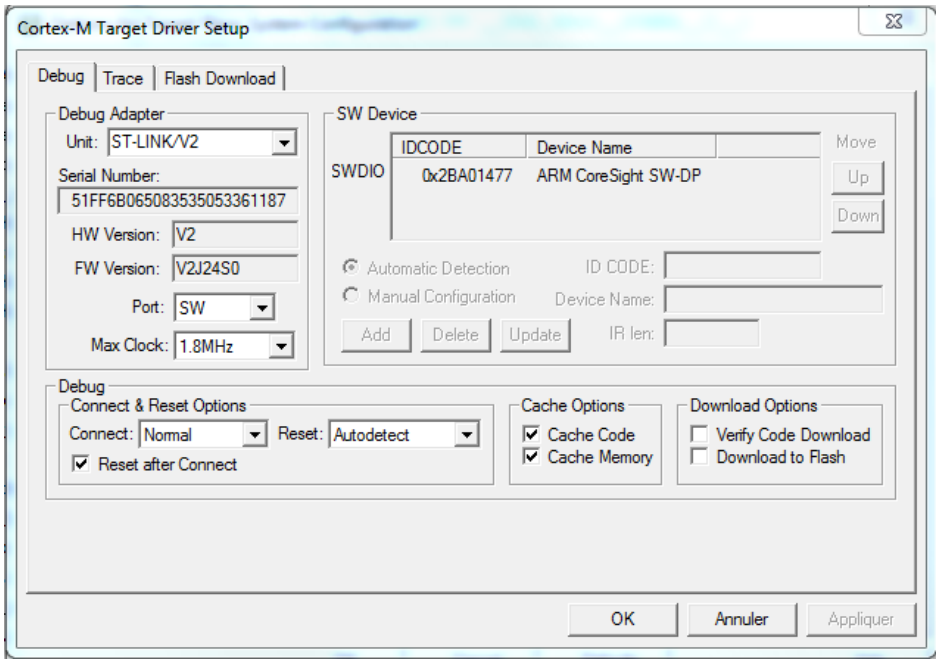
Select” Optimization level 3 (-O3)”

Select “Optimize for time”

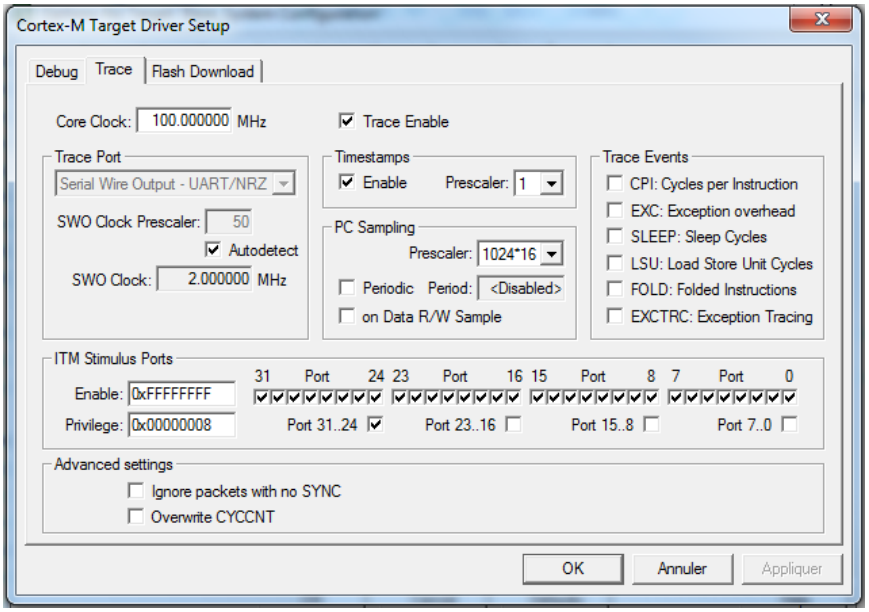
On Debug TAB in the use combo box, select “ST-Link Debugger” plug your Interface system by USB, wait for the top led to be steady red and click on Settings.

Then modify the options as shown bellow:

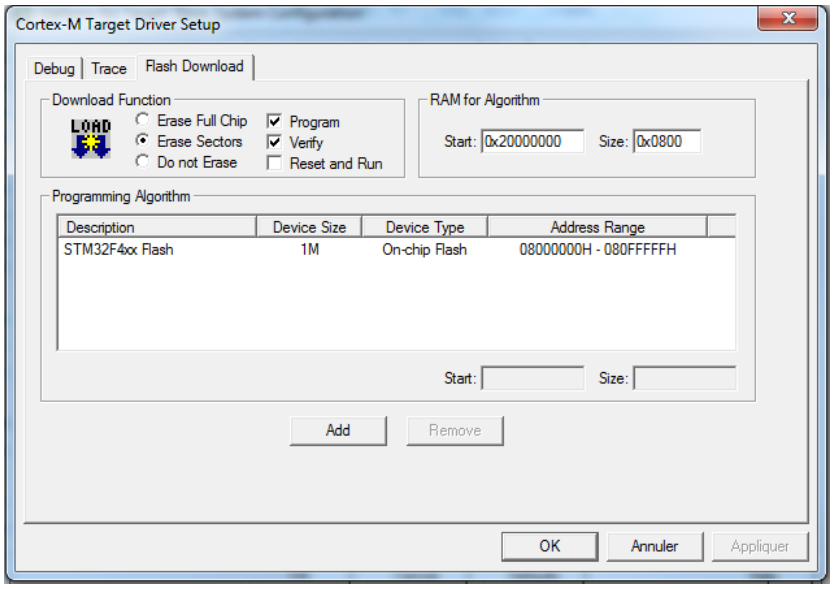
Debug TAB :



Trace TAB :



Flash Download TAB:



If there is nothing in Program Algorithm, select the one above “STM324xx Flash” by clicking on “Add”.

If this algorithm isn’t proposed, it’s because the proper pack isn’t installed, in pack manager, as saw above.

For the EMBEDDED SYSTEM

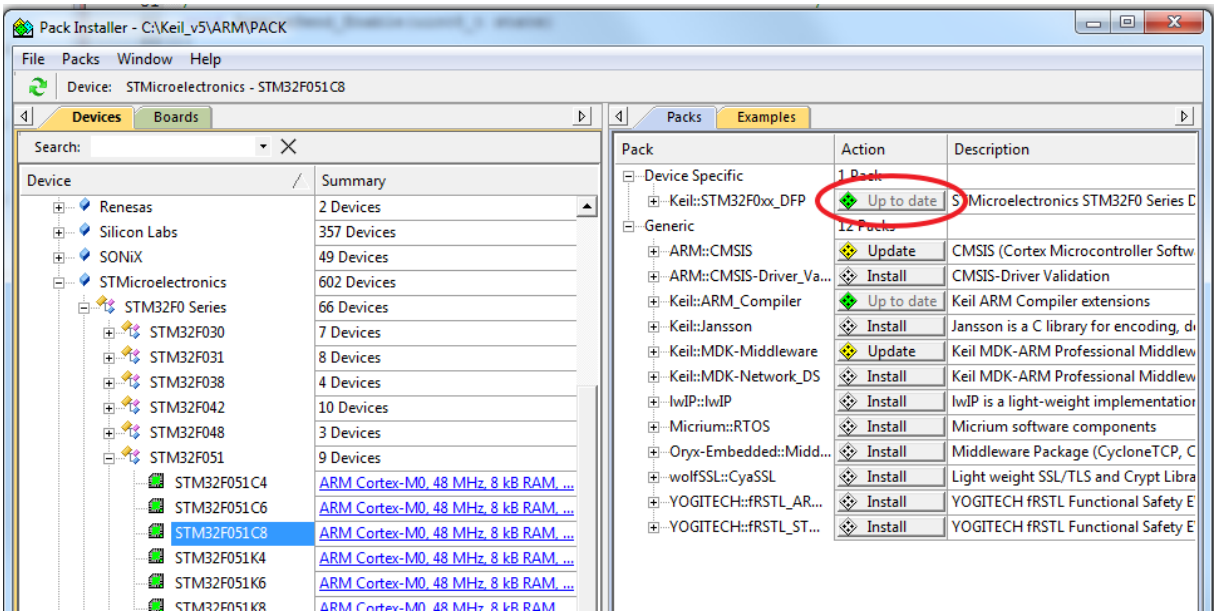
If the “pack installer” is not opened, we open it by clicking on “Project” ➔ “Manage” ➔ “Pack installer”

Base_System.uvprojx

Go to device TAB
Go to:
“STMincroelectronics” ➔
“STM32F0 Series” ➔
“STM32F051” ➔
Click on “STM32F051C8”

Go to Packs TAB
Install:
“Keil::STM32F0xx_DFP”

As shown bellow



After the packs installation, close the packs manager window, and go to Option for targets:

On device tab, select:
“STMincroelectronics” ➔
“STM32F0 Series” ➔
“STM32F051” ➔
“STM32F051C8”

On C/C++ Tab, in the box “define”, write: “USE_HAL_DRIVER,STM32F051x8”

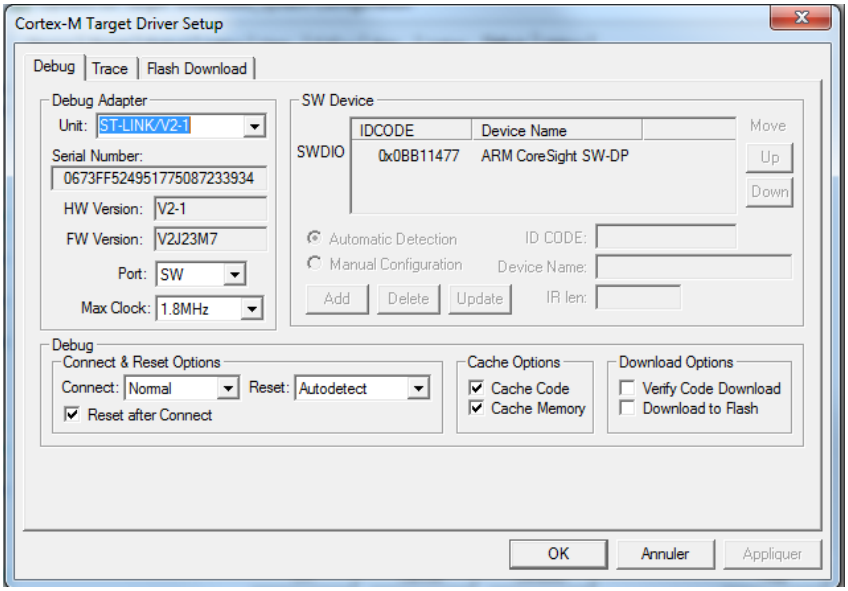
Select” Optimization level 3 (-O3)”

Select “Optimize for time”

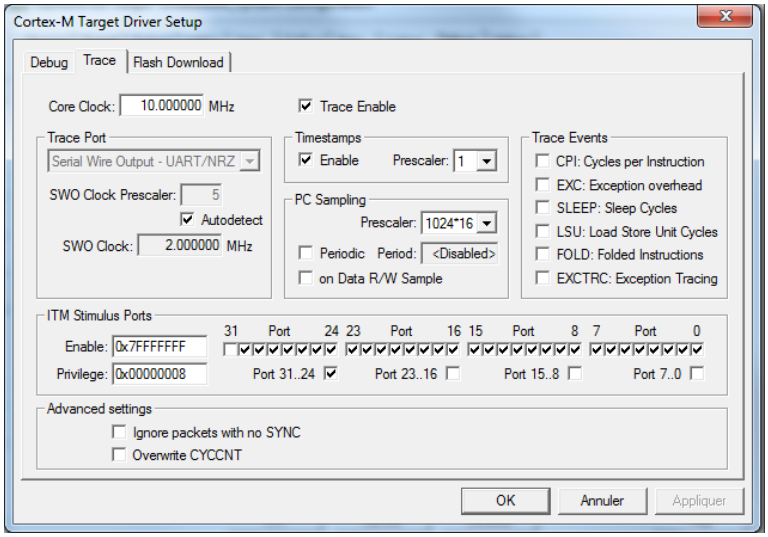
On Debug TAB in the use combo box, select “ST-Link Debugger” plug your Interface system by USB, wait for the top led to be steady red and click on Settings.

Then modify the options as shown below

Debug TAB:



Trace TAB:



Flash Download TAB:

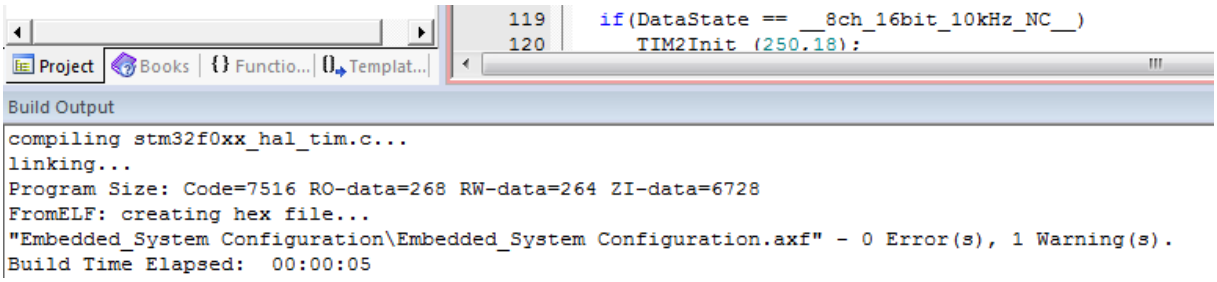
If there is nothing in Program Algorithm, select the one above “STM324xx Flash” by clicking on “Add”.

If this algorithm isn’t proposed, it’s because the proper pack isn’t installed, in pack manager, as saw above.

Compilation of the Code and download to the microcontroller

To compile the code click on the Rebuild icon

If the compilation succeeded, we will get this message in the build Output console:



Otherwise, if we get error, a “Target not created” message will be displayed in the console. Which in that case a debugging is needed.

Now that the compilation was successful, we check that the target (the microcontroller) is linked and detected by the software by clicking on “option for target” ➔ “Debug” tab ➔ “Settings” button. And see if there is “**ARM CoreSight SW-DP**” in the SW Device block.

Then close the windows and click on “Download”