TP1

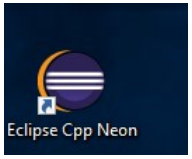
bosio@lirmm.fr

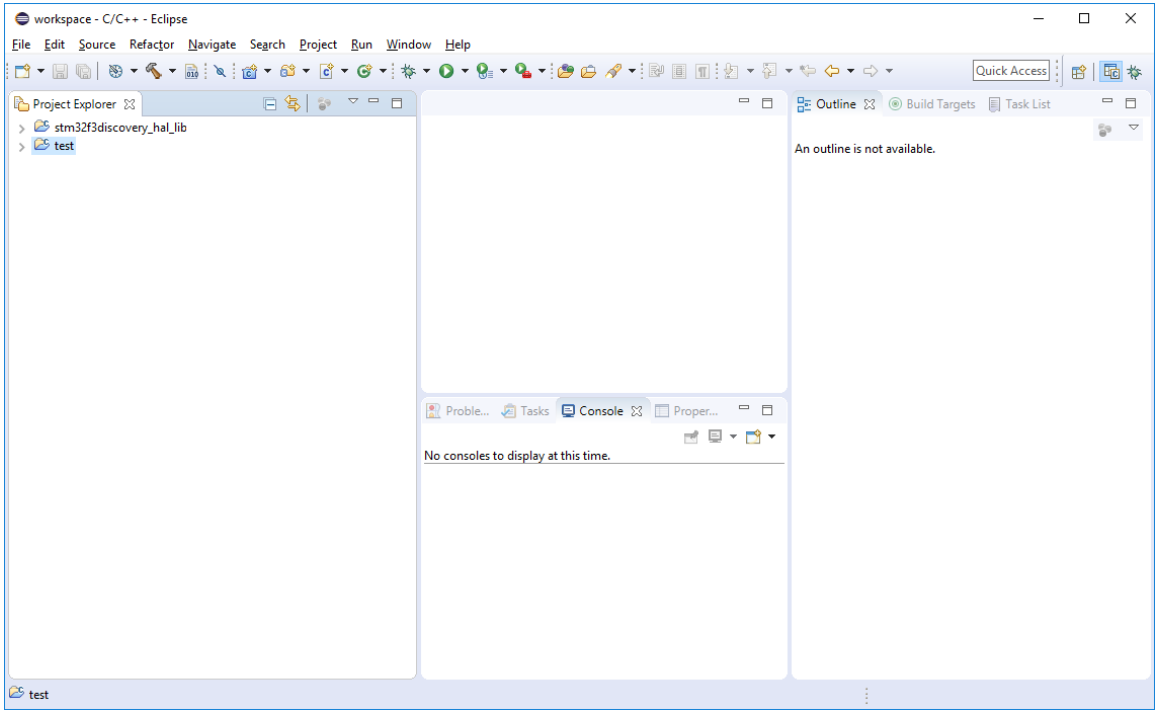
PC configuration

- Eclipse https://www.eclipse.org/home/index.php
- Board STM32F3 Discovery
- System workbench (www.openstm32.org)

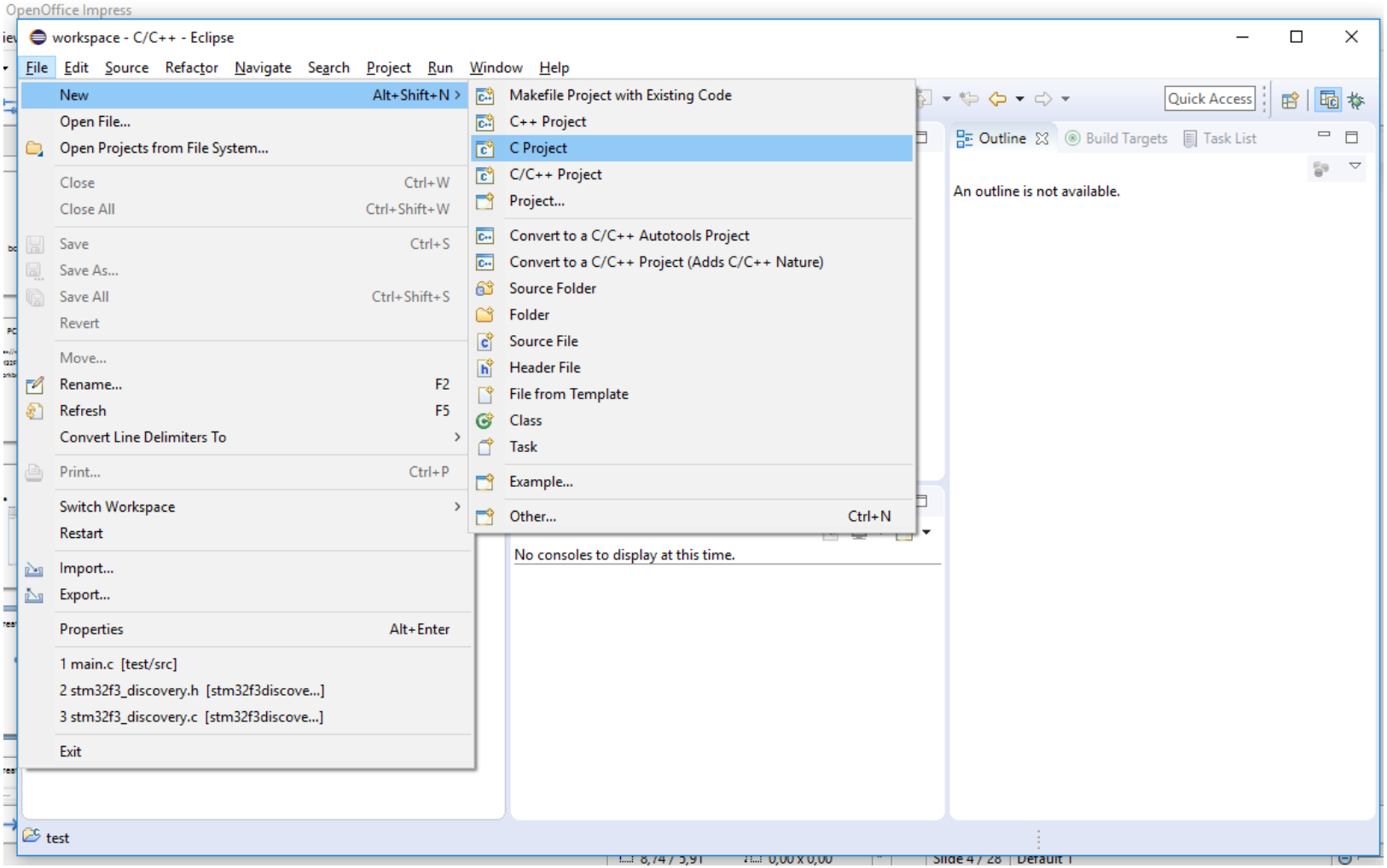
Eclipse

- Run eclipse

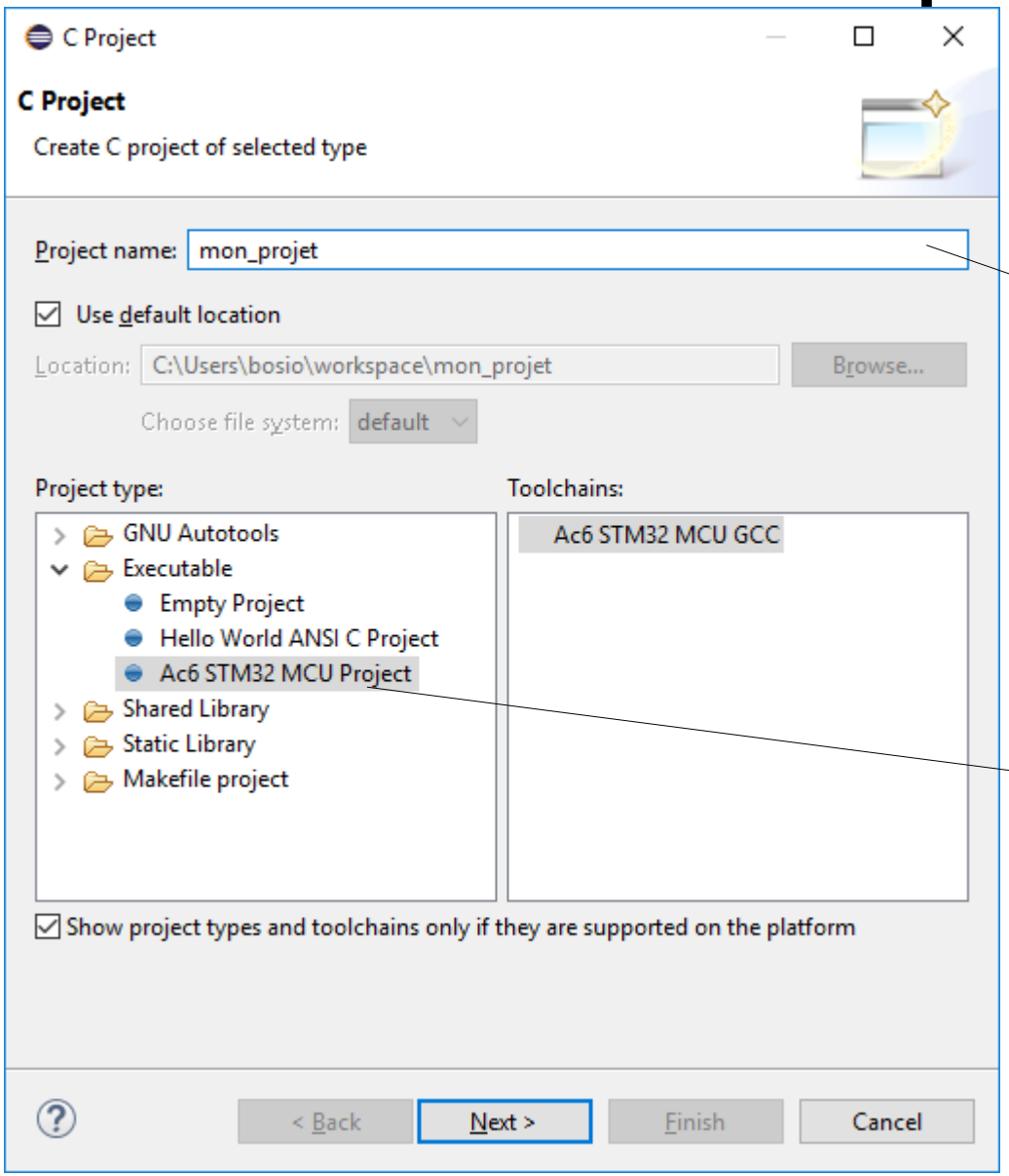




Create a new project



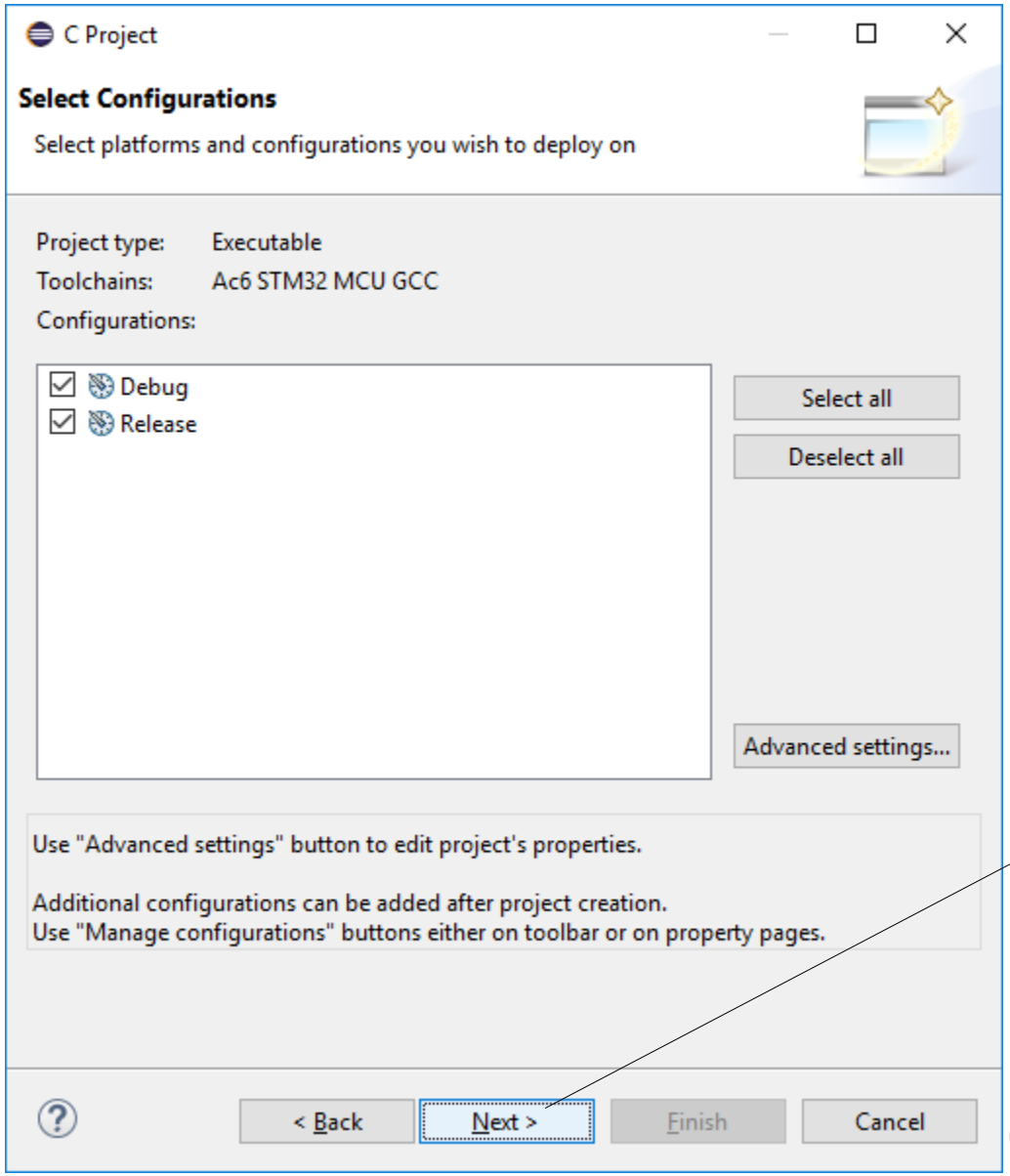
Create a new project



Type the name of your project

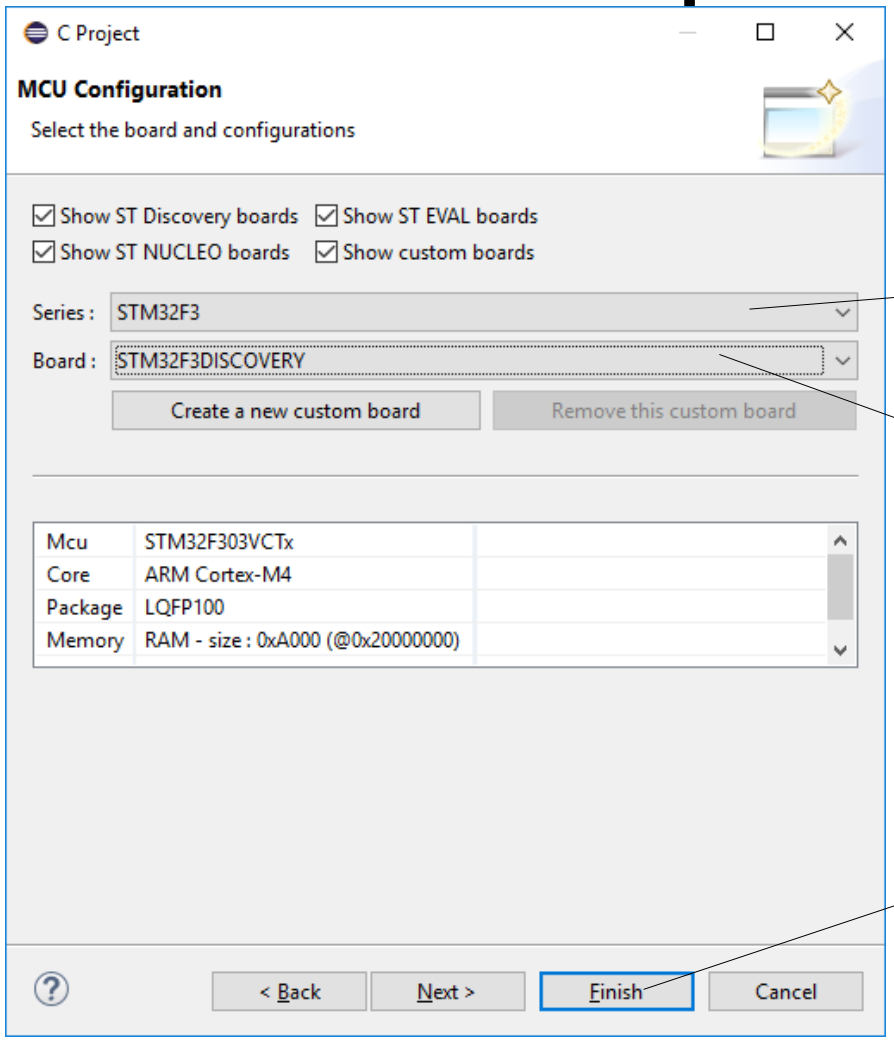
Select the AC6 options

Create a new project



Click on next

Create a new project: select the development board

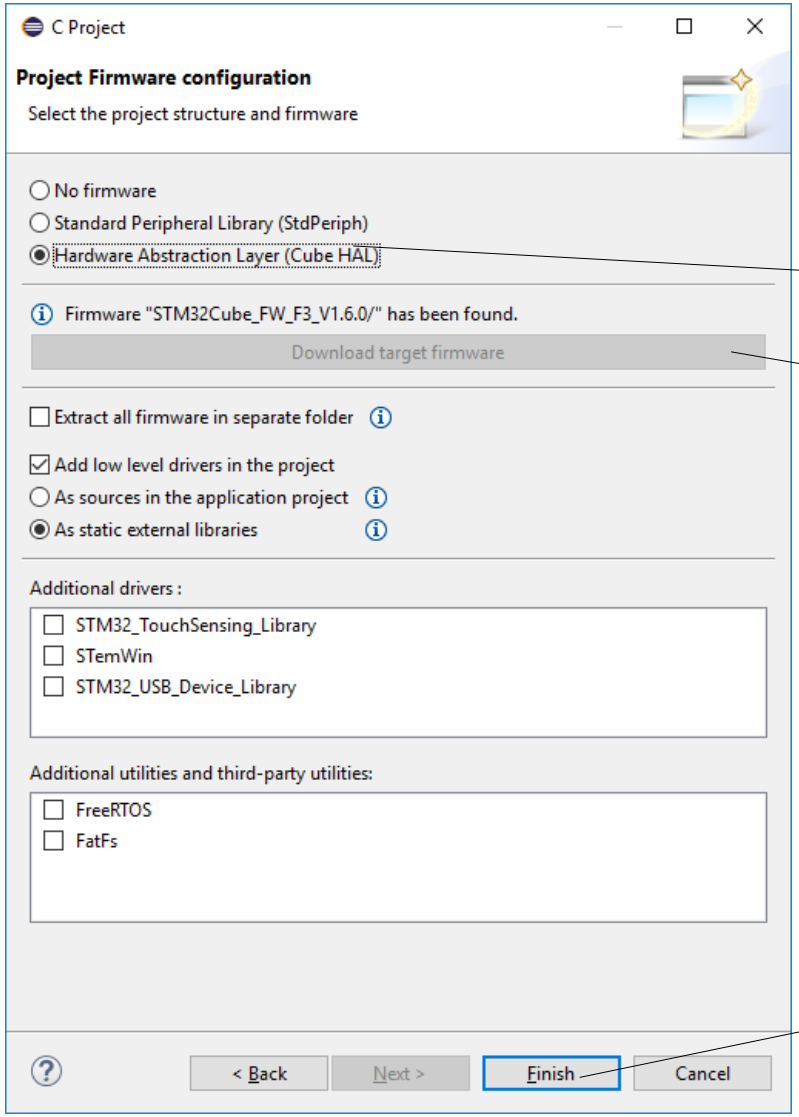


STM32F3

STM32F3DISCOVERY

Click on finish

Download the Firmware



Cube hal

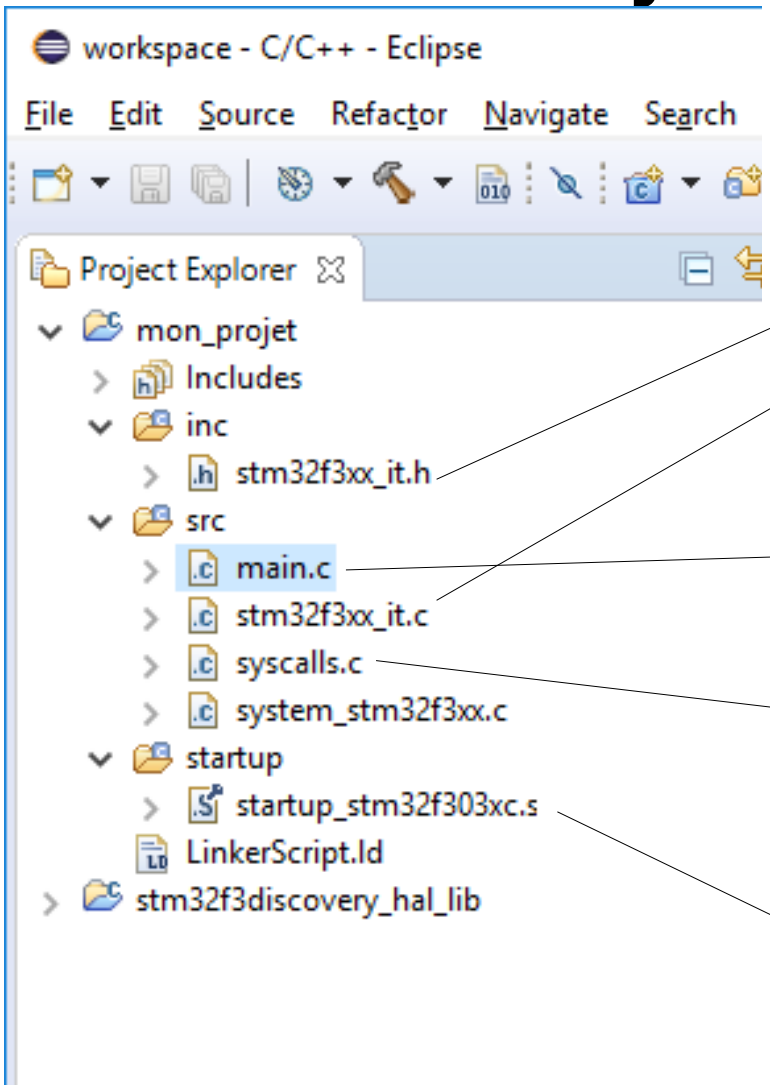
Download the firmware (required only the first time)

Click on finish

Project Structure

- Your project is composed of several files
- Let’s have a look...

Project Structure



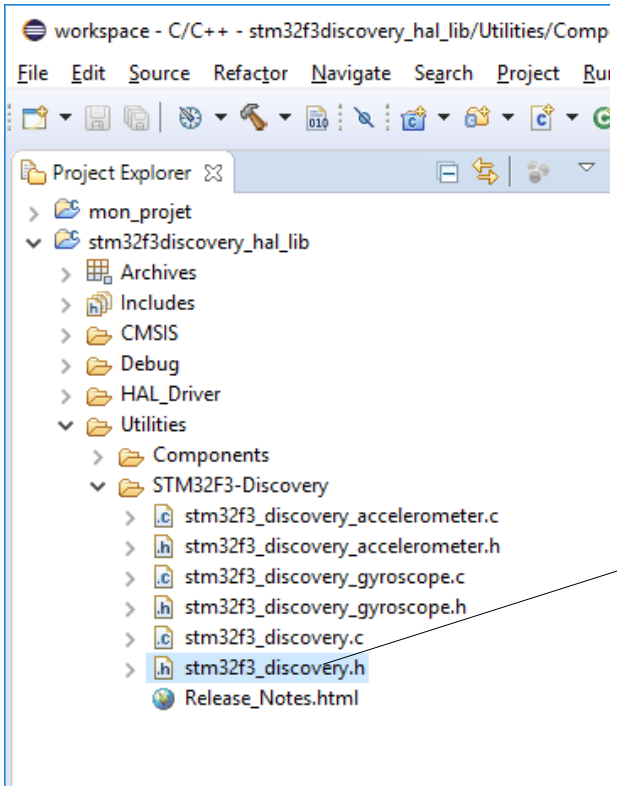
Interrupt definition header and c file

Main with your code

System calls (redefinition of stdio.h calls)

Assembly startup file, the very first code executed by MCU and interrupt prototypes

HAL Library



- HAL Driver: developed by ST, deliver high level function to access peripherals (one file each periph.)

Some usueful functions, we will use in the first Labs

First Project

- Download the file
  - www.lirmm.fr/~bosio/L3/lab1.c
- Open the lab1.c with a text editor and copy the code into the main.c of your eclipse project

Look at the code

```
18 int find_maximum(char* , int);
19
20 int main() {
21   int size, location, maximum;  //Int variables (32 bit)
22   //Array of Unsigned 8 bit
23   char array[]  = {3, 6, 3, 7, 12, 15, 1, 8, 256, 2, 763, 85};
24
25   //Compute size of the array
26   size = sizeof(array)/sizeof(array[0]);
27   location = find_maximum(array, size-1);
28   maximum  = array[location];
29
30   BSP_LED_Init(LED4);
31   BSP_LED_On(LED4);
32
33   while(1);
34 }
35
36 int find_maximum(char *a, int n) {
37   int c, max, index;
38   max = a[0];
39   index = 0;
40
41   for (c = 1; c < n; c++) {
42     if (a[c] > max) {
43        index = c;
44        max = a[c];
45     }
46   }
```

Function prototypes

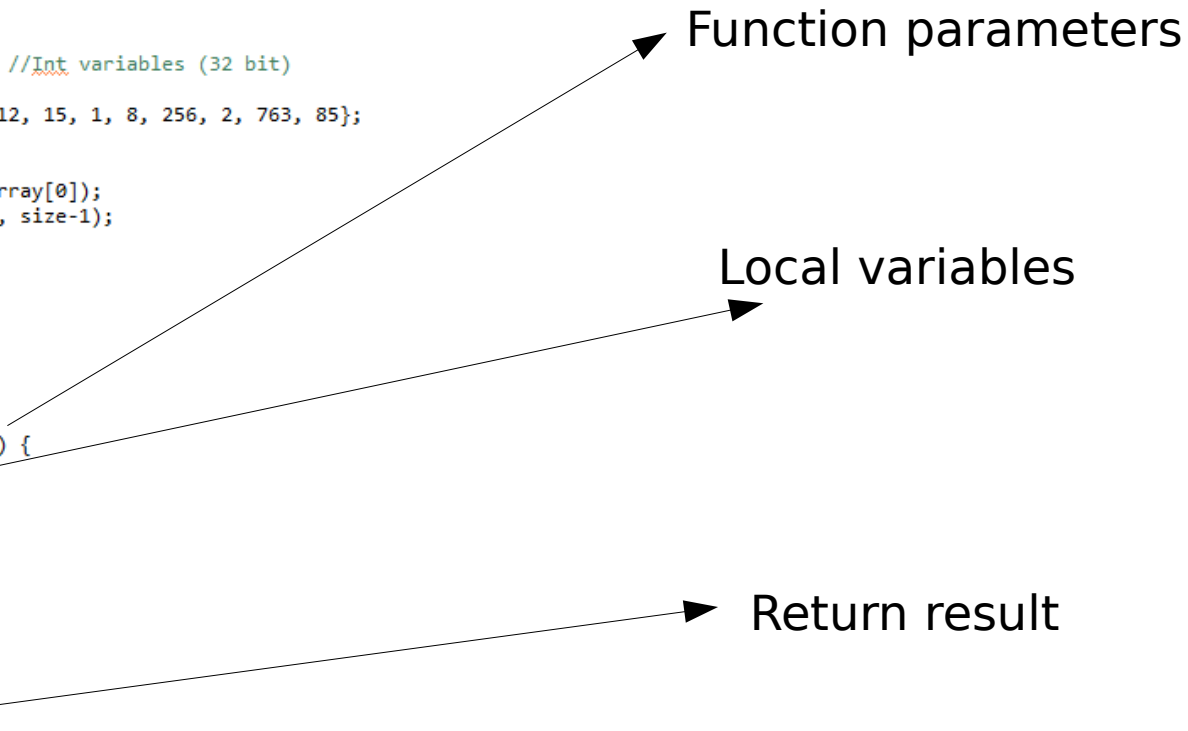
Local Variables

Switch on LED4 on board (where the function is defined? Look into stm32f3_discovery.h and stm32f3_discovery.c)

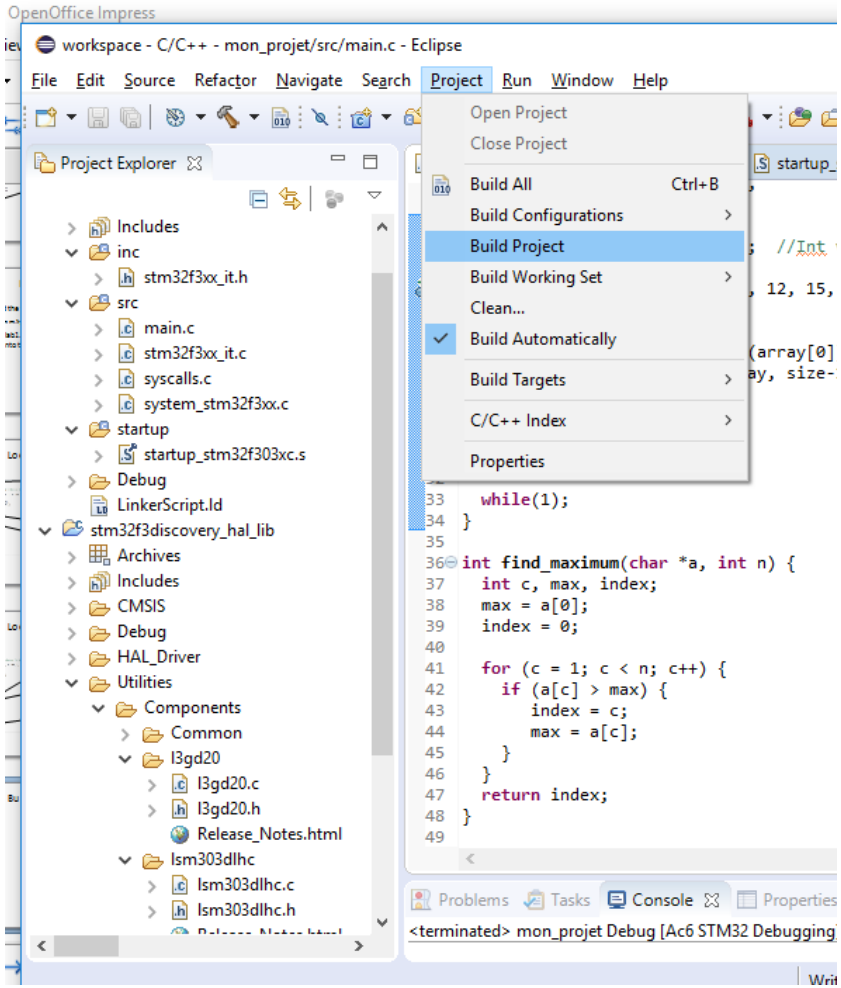
Never exit from main

Look at the code

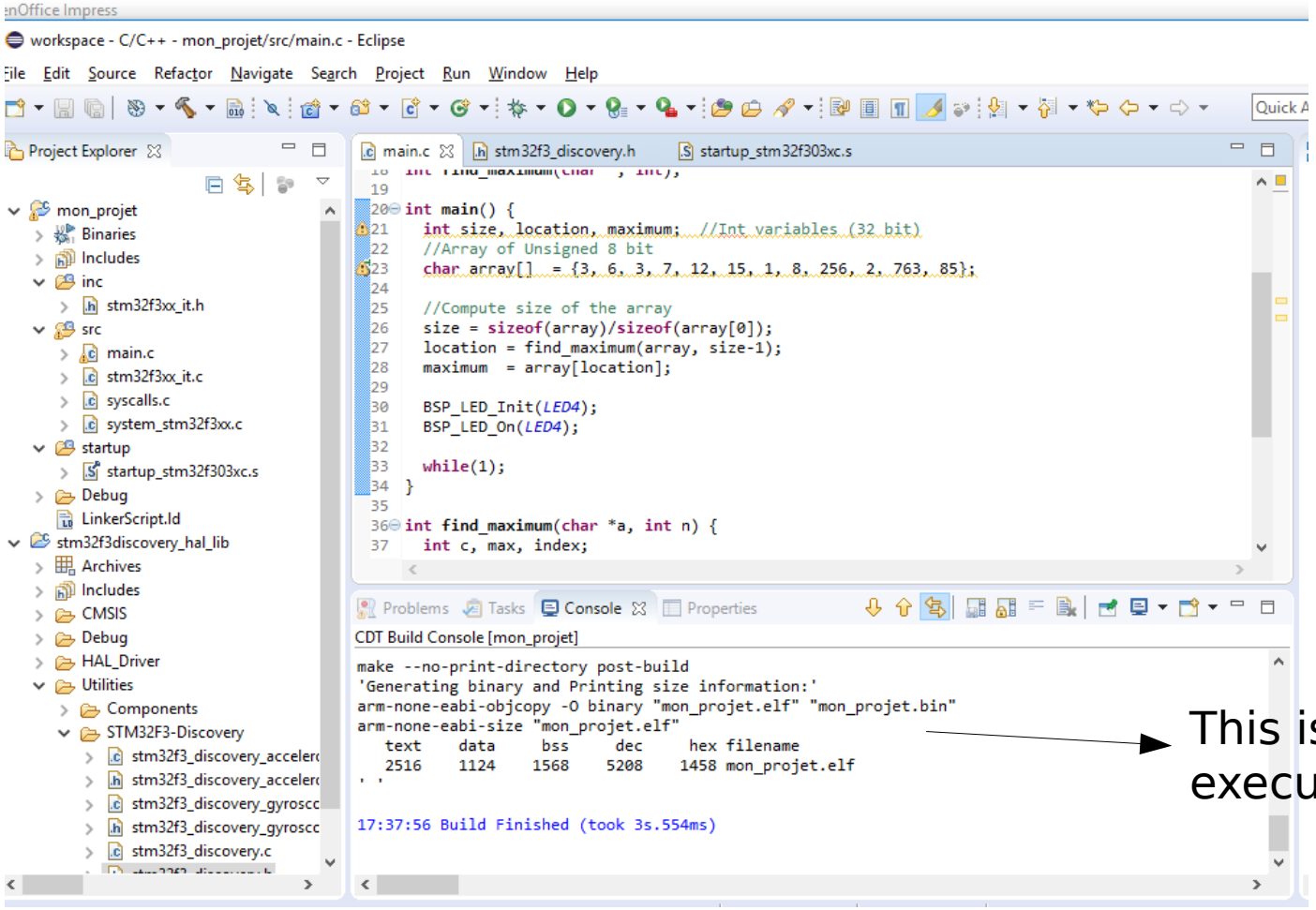
```
20 int main() {
21   int size, location, maximum;  //Int variables (32 bit)
22   //Array of Unsigned 8 bit
23   char array[]  = {3, 6, 3, 7, 12, 15, 1, 8, 256, 2, 763, 85};
24
25   //Compute size of the array
26   size = sizeof(array)/sizeof(array[0]);
27   location = find_maximum(array, size-1);
28   maximum  = array[location];
29
30   BSP_LED_Init(LED4);
31   BSP_LED_On(LED4);
32
33   while(1);
34 }
35
36 int find_maximum(char *a, int n) {
37   int c, max, index;
38   max = a[0];
39   index = 0;
40
41   for (c = 1; c < n; c++) {
42     if (a[c] > max) {
43        index = c;
44        max = a[c];
45     }
46   }
47   return index;
48 }
49
```



Build the project

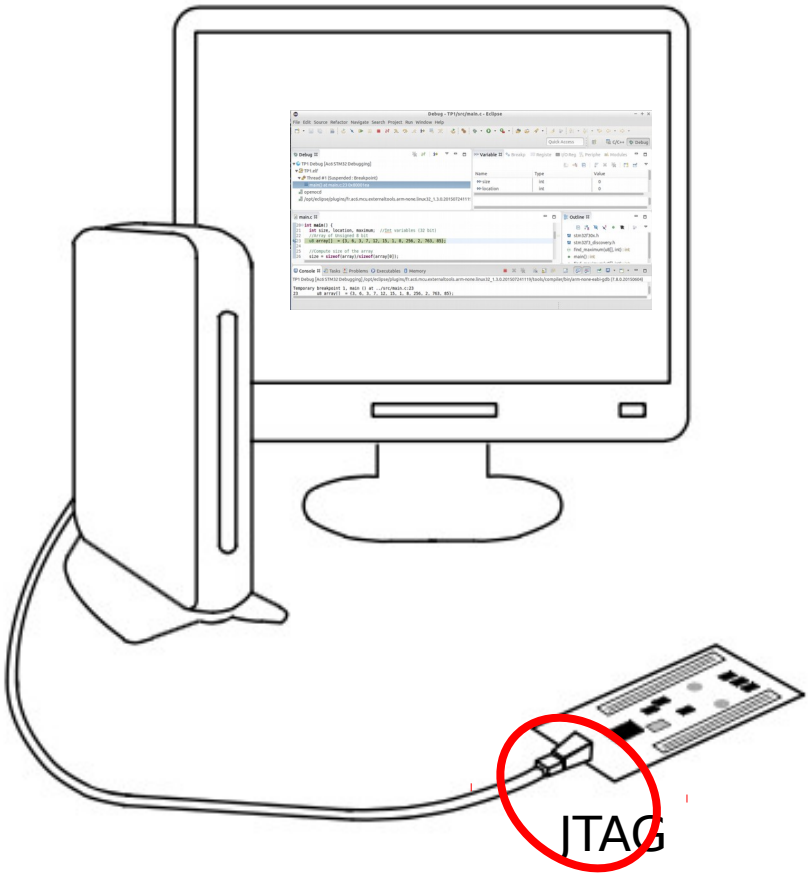


Build Result



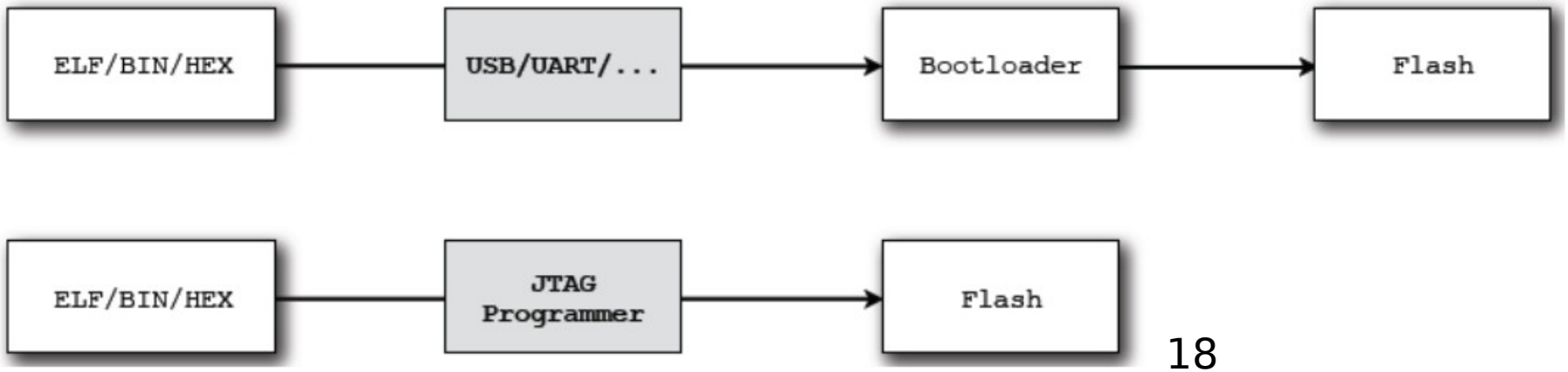
Execute your code

- **We have to send the binary code to the board**



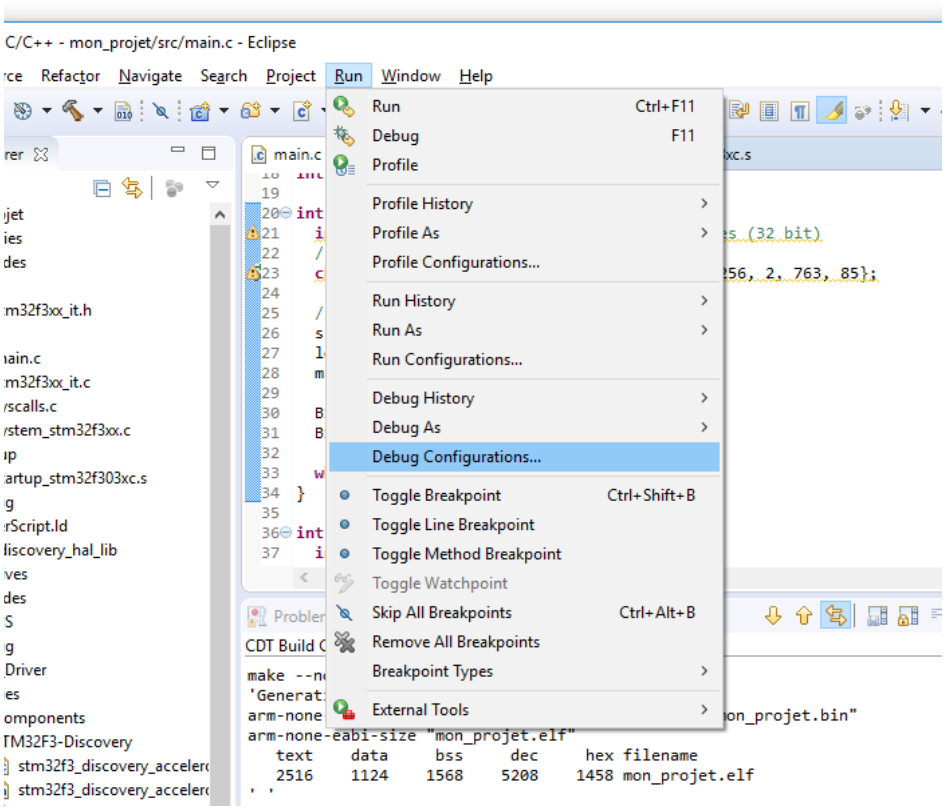
JTAG for programming

- To program a device we have two alternatives:
  - Using a USB / UART / ... connection in bootloader mode
  - **Using JTAG and programmer to**

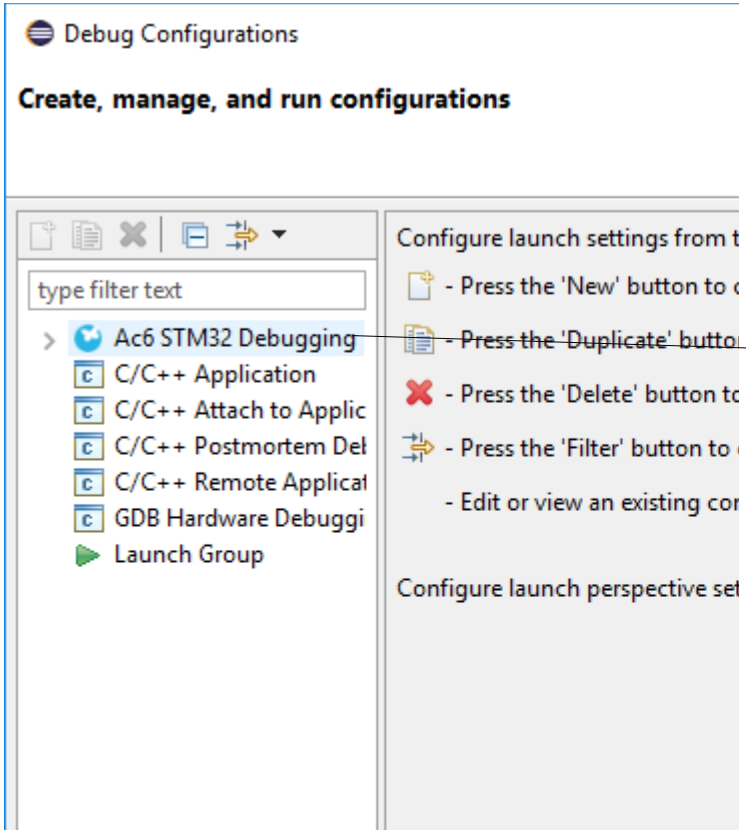


JTAG for programming

- To use JTAG we have to pass through the DEBUGGER
  - It use OpenOCD (http://openocd.org/)

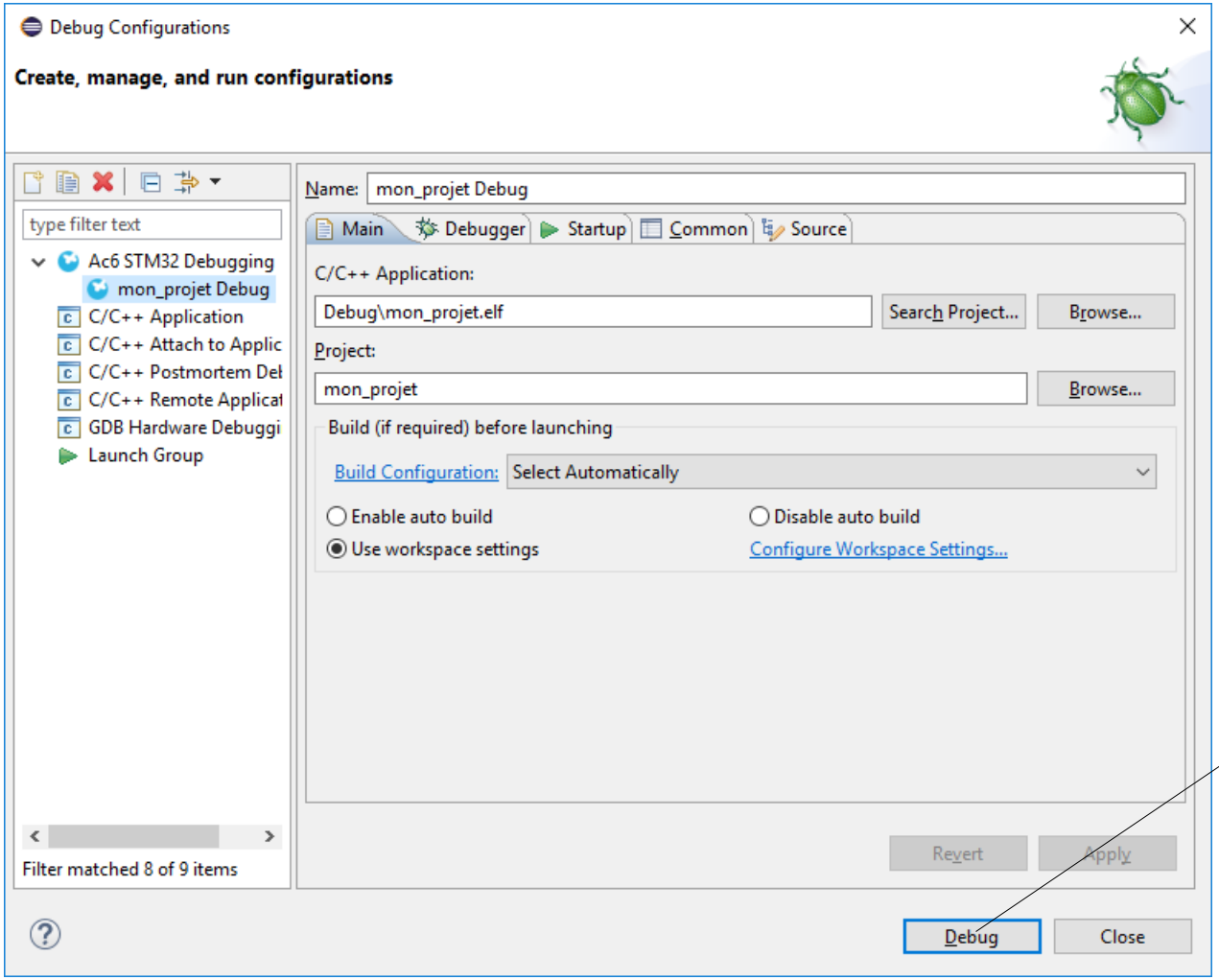


Debug your code



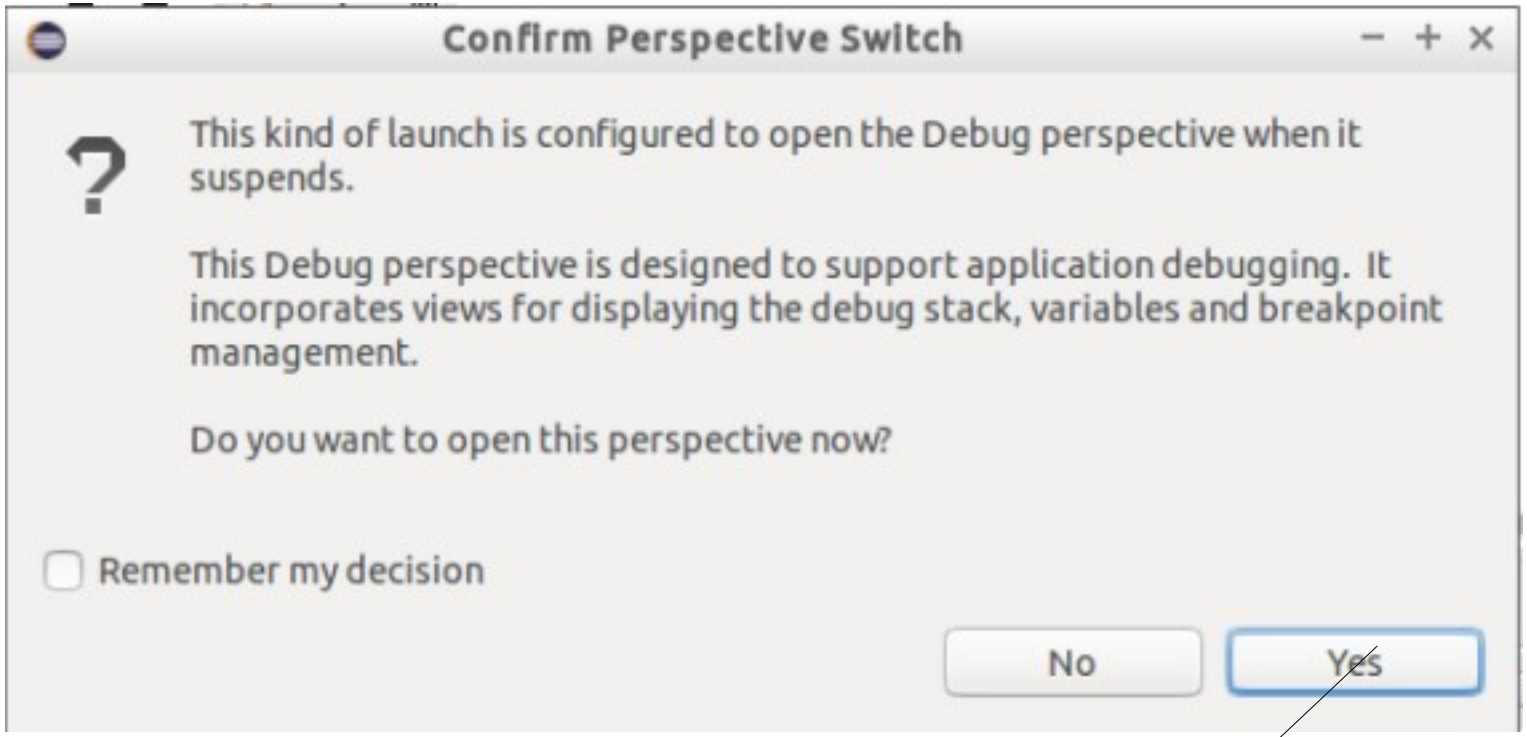
Double click

Debug your code



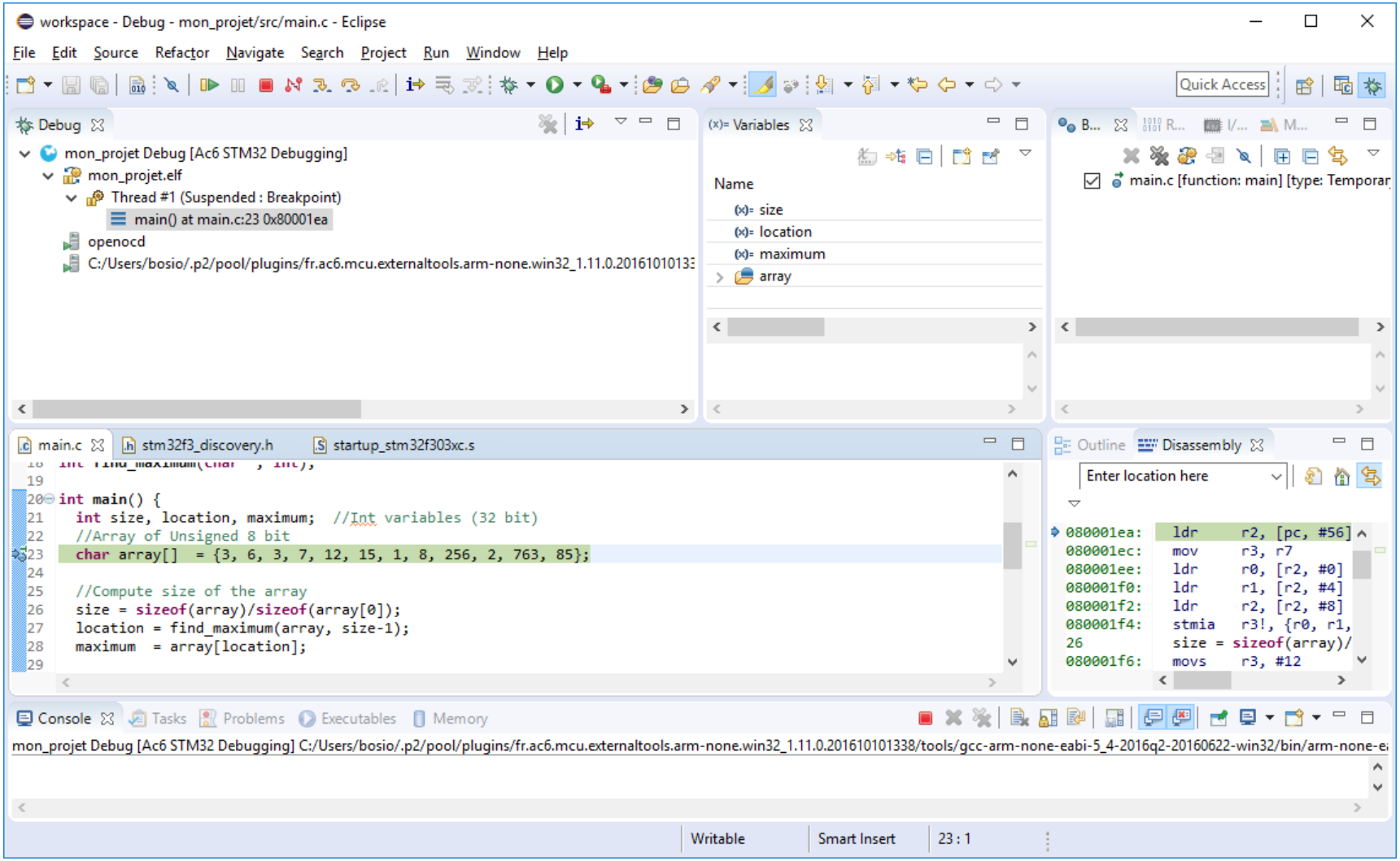
You can start to debug

Debug your code



You activate the debug mode of eclipse

The debug view



Debug yourself

- Try to debug code on the board
- Monitor variables
- Check debug steps buttons
- Is everything working correctly?
- Do the functions works properly?
- Are the results correct?
  - If not why?

Exercise

- Un palindrome est un mot qui reste le même qu'on le lise de gauche à droite ou de droite à gauche (par exemple, PIERRE n'est pas un palindrome, alors que OTTO est un palindrome).
  - Ecrire un programme qui vérifie si une chaîne simple (sans espace) est un palindrome: si oui allumer le LED4, sinon le LED7
  - La chaîne est constante (comme pour l’exemple précédente)