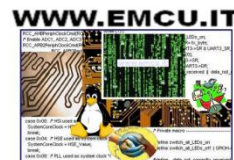




Presenter's name

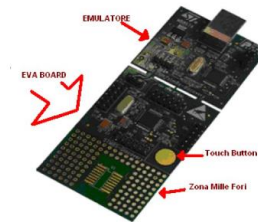
STM8 StartUp



HW and SW tools

HW:

- STM8S-Discovery



SW:

- STM STVP + STVD
- STM8S/A Library
- Cosmic C Compiler
- ST-LINK-v2 (Optional)

LINK

- Useful links are [here](#)

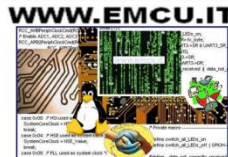
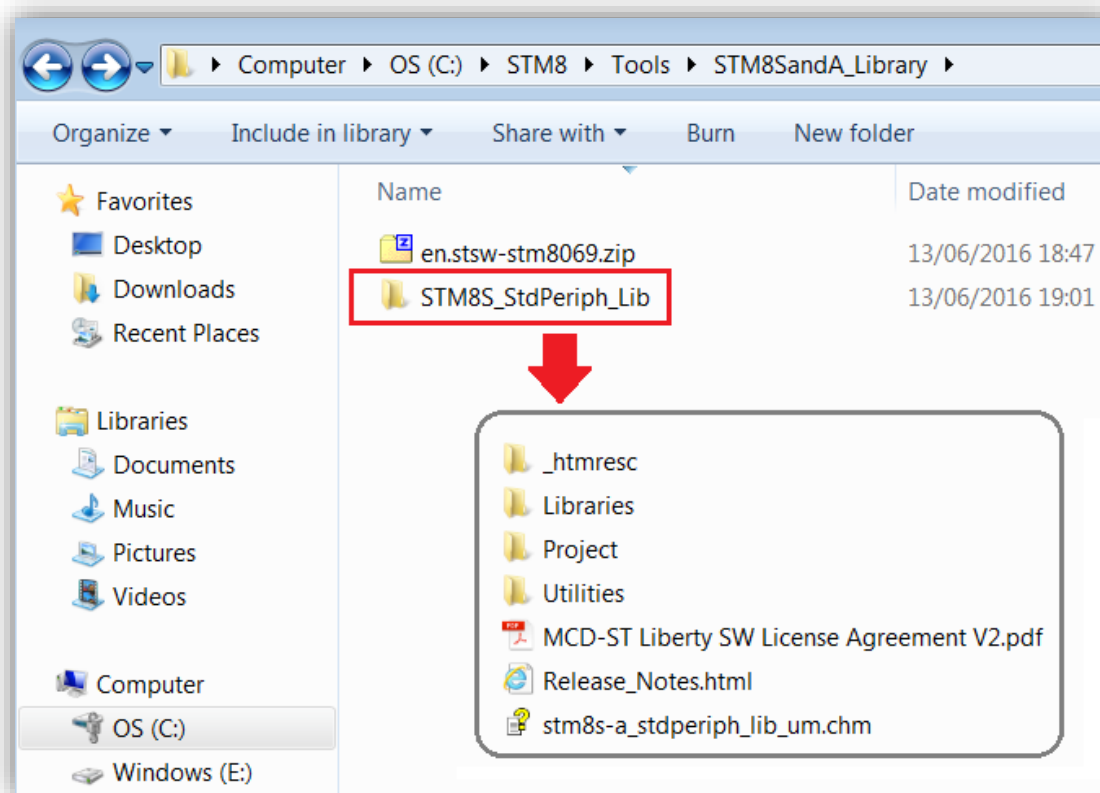
Installations

Please download and install the following SW

- [STM STVP + STVD](#)
- [Cosmic C Compiler](#)

STM8 Library - 1/2

After download of the STM8S/A Library unzip it, you must see something like below.



STM8 Library - 2/2

[stm8s-a_stdperiph_lib_um.chm](#) - is the manual of the library

Libraries - In this folder there are the library (**inc** & **src**).
Also there is a manual that detail the library drivers (**stm8s-a_stdperiph_drivers_um.chm**).

Utilities - is a folder there the are some files used on the discovery and eva boards.

Project - This folder contain other two folder.

STM8S_StdPeriph_Examples - contain a lot of peripheral examples ready to use.

STM8S_StdPeriph_Template - contain a predefined structure ready to use with the C Compiler.

Concepts regarding the way for develop a new SW

- It is a good idea to use the examples for test all the peripheral that you need to use in your project.
- I suggest to test one by one the single peripheral and at the end mix all the peripheral.
- I also suggest to study this manuals.

RM0016: STM8S series and STM8AF series 8-bit microcontrollers

PM0044: STM8 CPU programming manual

AN2860: EMC guidelines for STM8 microcontrollers

AN2867: Oscillator design guide for STM8S, STM8A and STM32 microcontrollers

AN3258: STM8AF and STM8S series HSI oscillator calibration using LIN automatic resynchronization

AN2822: STM8S and STM8A high speed internal oscillator calibration

AN2857: STM8S and STM8A family power management

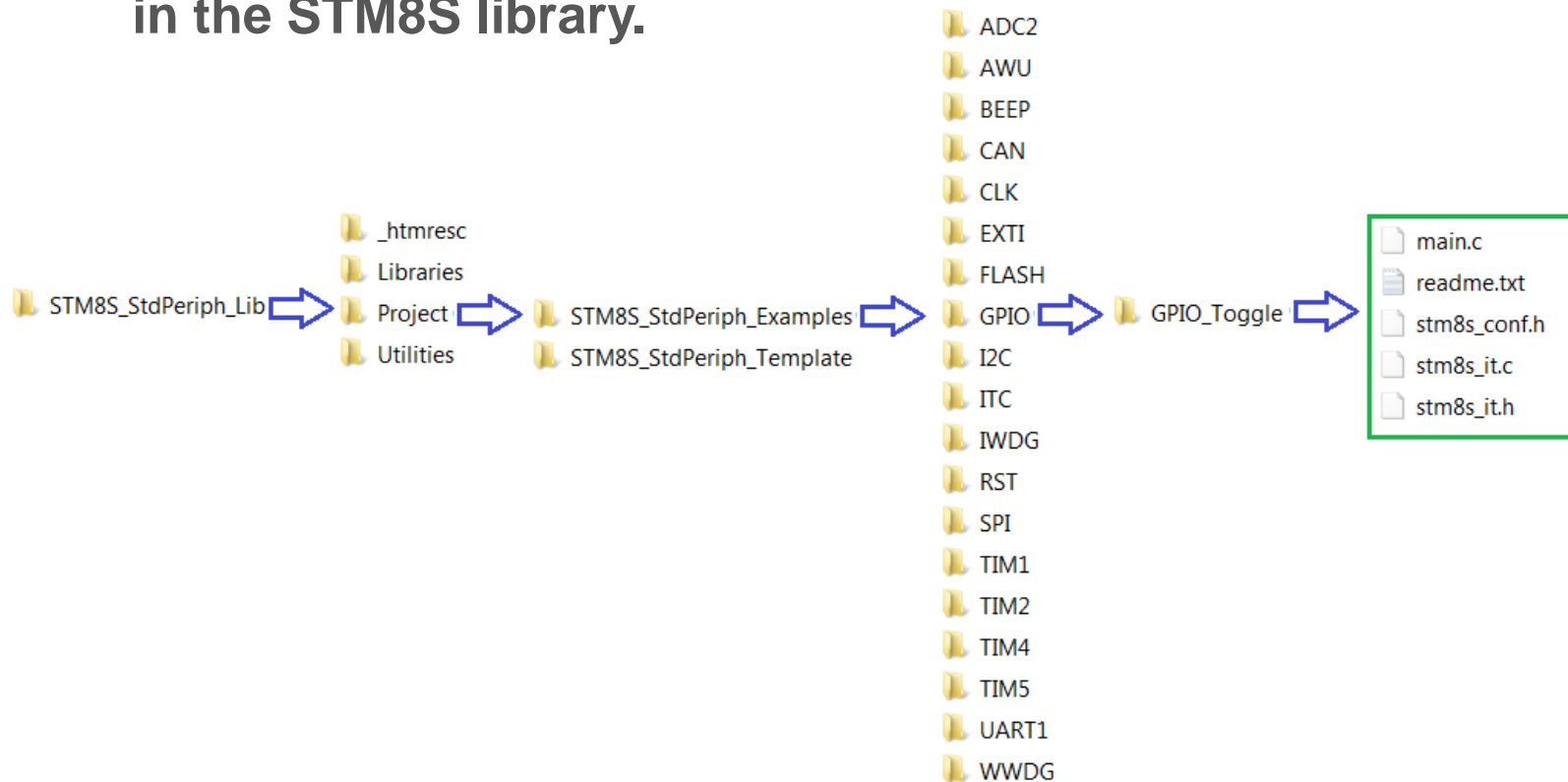
AN2658: Using the analog to digital converter of the STM8S microcontroller

AN3259: RS232 communications with a terminal using the STM8S-DISCOVERY

See also [this](#) tutorials.

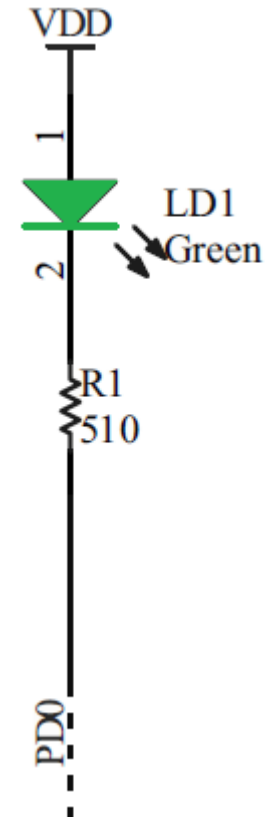
Start a new project for flashing a led 1/18

We need to flashing a LED on STM8S-Discovery.
As I said, we start using a toggle example that is present in the STM8S library.

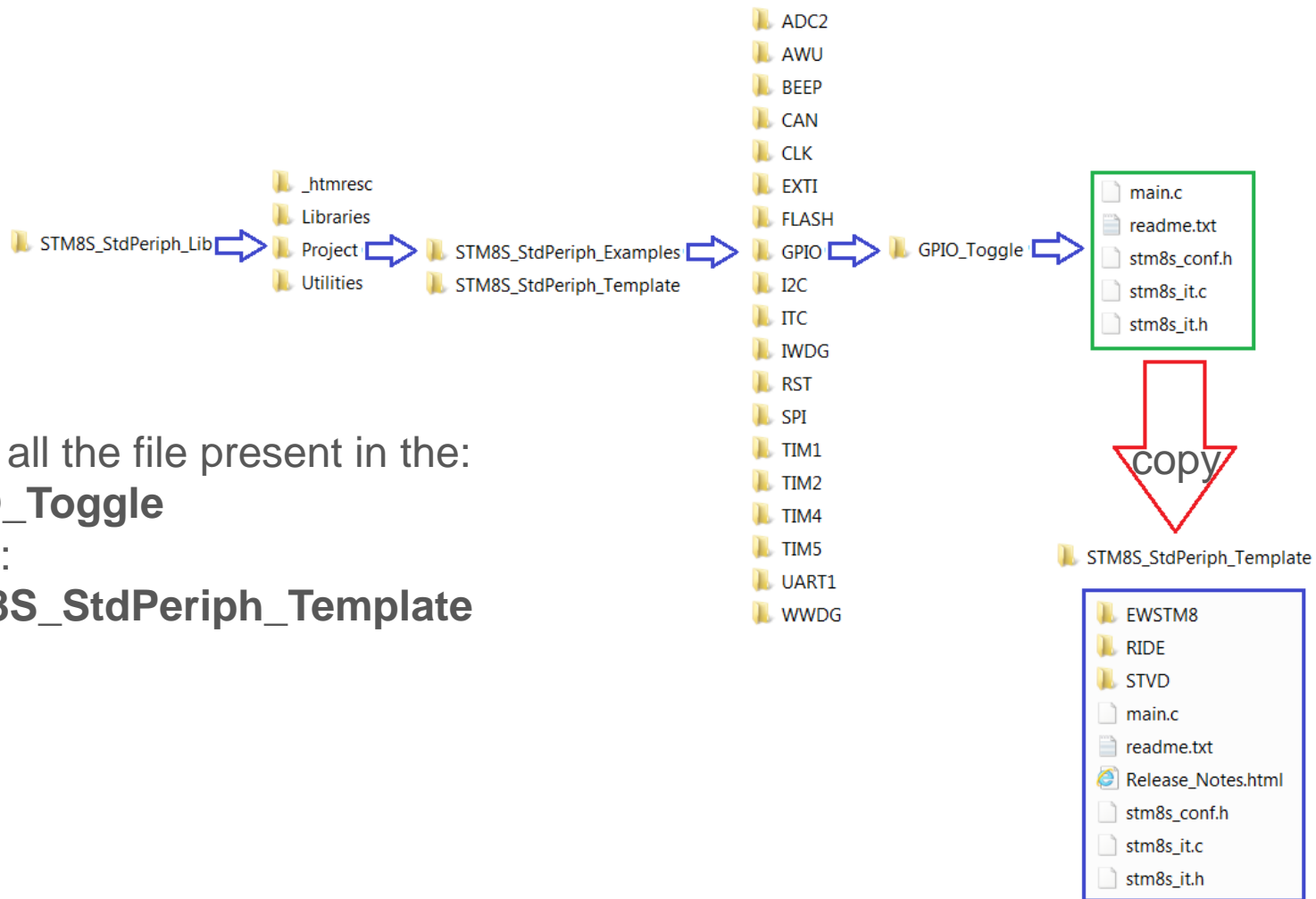


Start a new project for flashing a led 2/18

From the STM8S-Discovery manual
the LED is connected to **PD0**



Start a new project for flashing a led 3/18



Copy all the file present in the:
GPIO_Toggle
in the:
STM8S_StdPeriph_Template

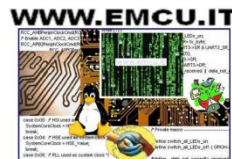
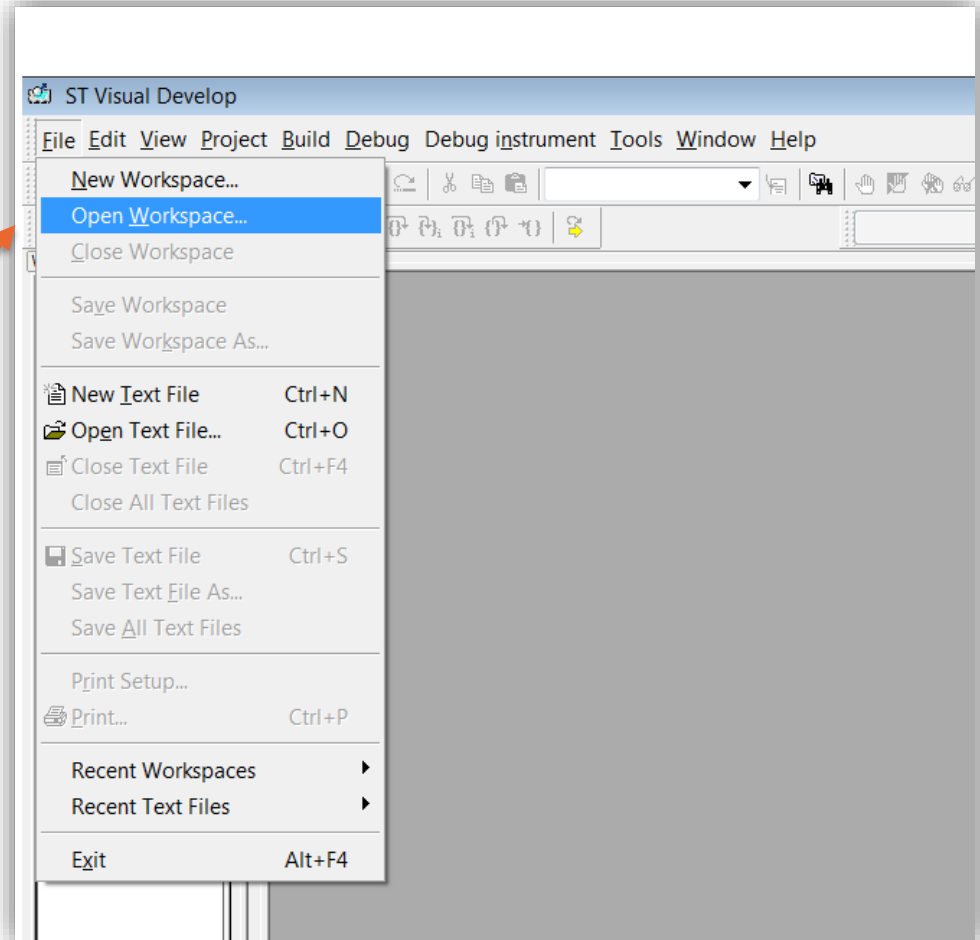
Start a new project for flashing a led 4/18

Run the:



and select:

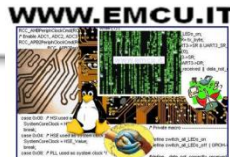
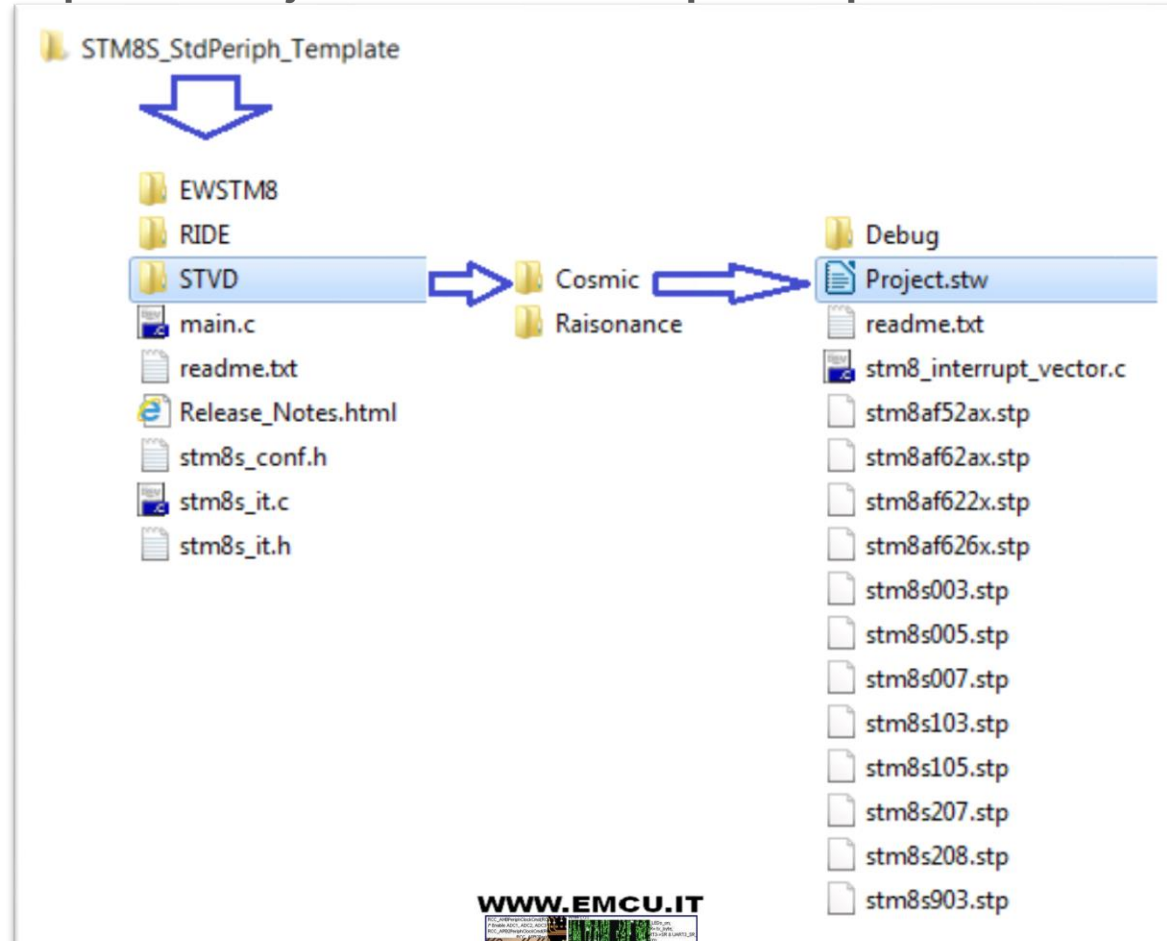
File -> Open Workspace



Start a new project for flashing a led 5/18

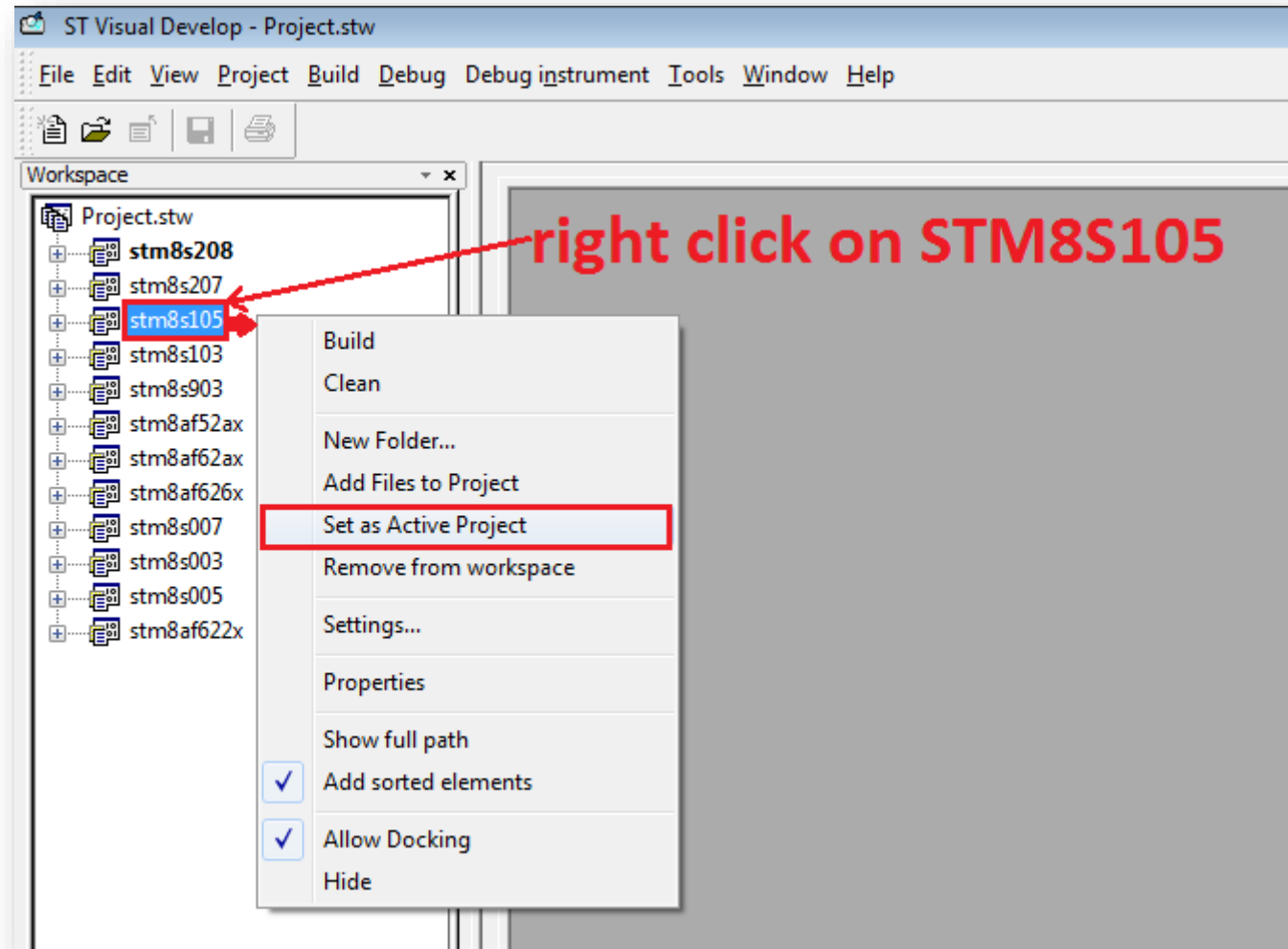
Open Workspace:

C:\STM8S_StdPeriph_Lib\Project\STM8S_StdPeriph_Template\STVD\Cosmic\Project.stw



Start a new project for flashing a led 6/18

The **STM8S-Discovery** is based on:
STM8S105C6T6.
Choose the **STM8105** family, see the figure.



WWW.EMCU.IT



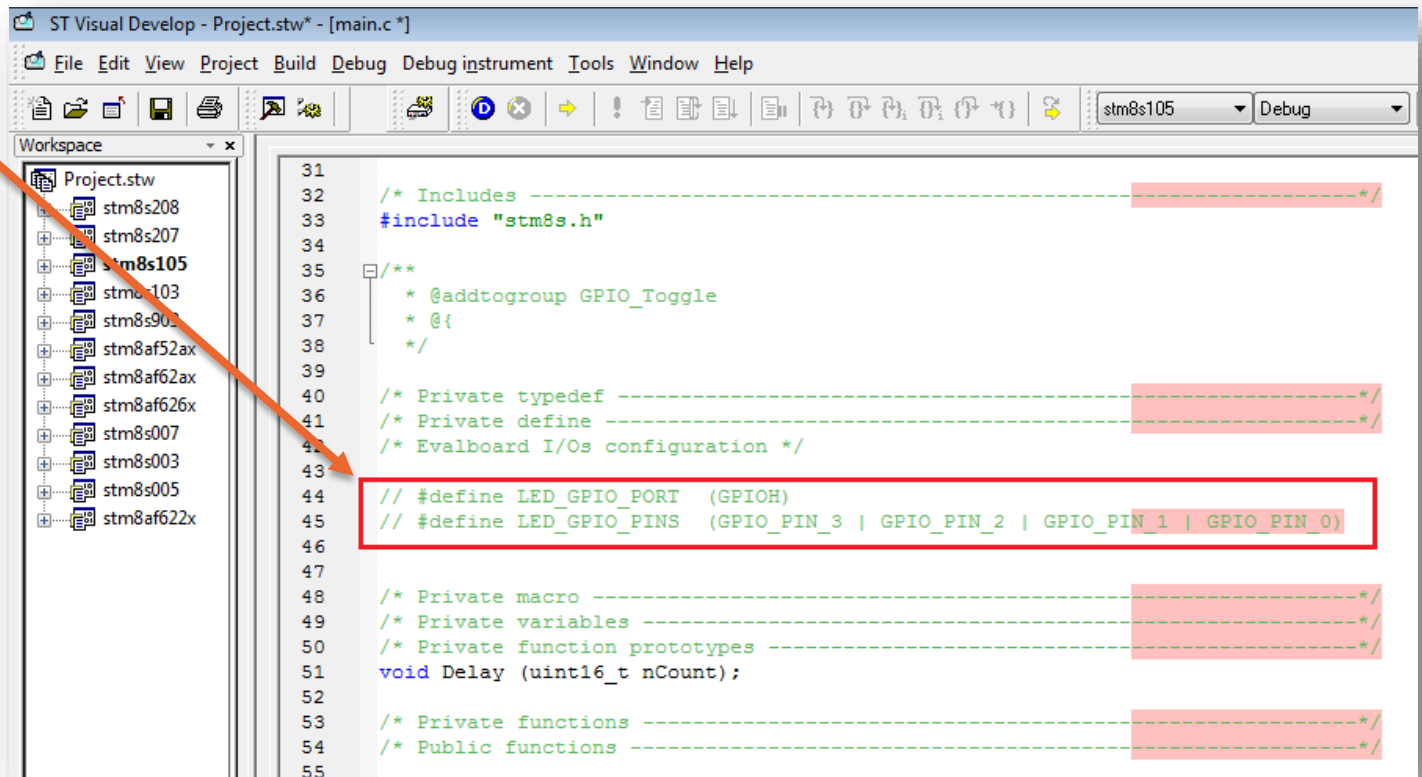
Start a new project for flashing a led 7/18

Comment the two line:

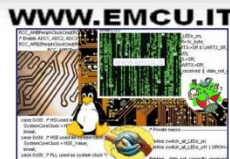
```
#define LED_GPIO_PORT (GPIOH)
```

```
#define LED_GPIO_PINS (GPIO_PIN_3 | GPIO_PIN_2 | GPIO_PIN_1 | GPIO_PIN_0)
```

see the red box.



```
ST Visual Develop - Project.stw* - [main.c *]  
File Edit View Project Build Debug Debug instrument Tools Window Help  
stm8s105 Debug  
Workspace  
Project.stw  
  stm8s208  
  stm8s207  
  stm8s105  
  stm8s103  
  stm8s903  
  stm8af52ax  
  stm8af62ax  
  stm8af626x  
  stm8s007  
  stm8s003  
  stm8s005  
  stm8af622x  
31  
32 /* Includes ----- */  
33 #include "stm8s.h"  
34  
35 /**  
36  * @addtogroup GPIO_Toggle  
37  * @{  
38  */  
39  
40 /* Private typedef ----- */  
41 /* Private define ----- */  
42 /* Evalboard I/Os configuration */  
43  
44 // #define LED_GPIO_PORT (GPIOH)  
45 // #define LED_GPIO_PINS (GPIO_PIN_3 | GPIO_PIN_2 | GPIO_PIN_1 | GPIO_PIN_0)  
46  
47  
48 /* Private macro ----- */  
49 /* Private variables ----- */  
50 /* Private function prototypes ----- */  
51 void Delay (uint16_t nCount);  
52  
53 /* Private functions ----- */  
54 /* Public functions ----- */  
55
```



Start a new project for flashing a led 8/18

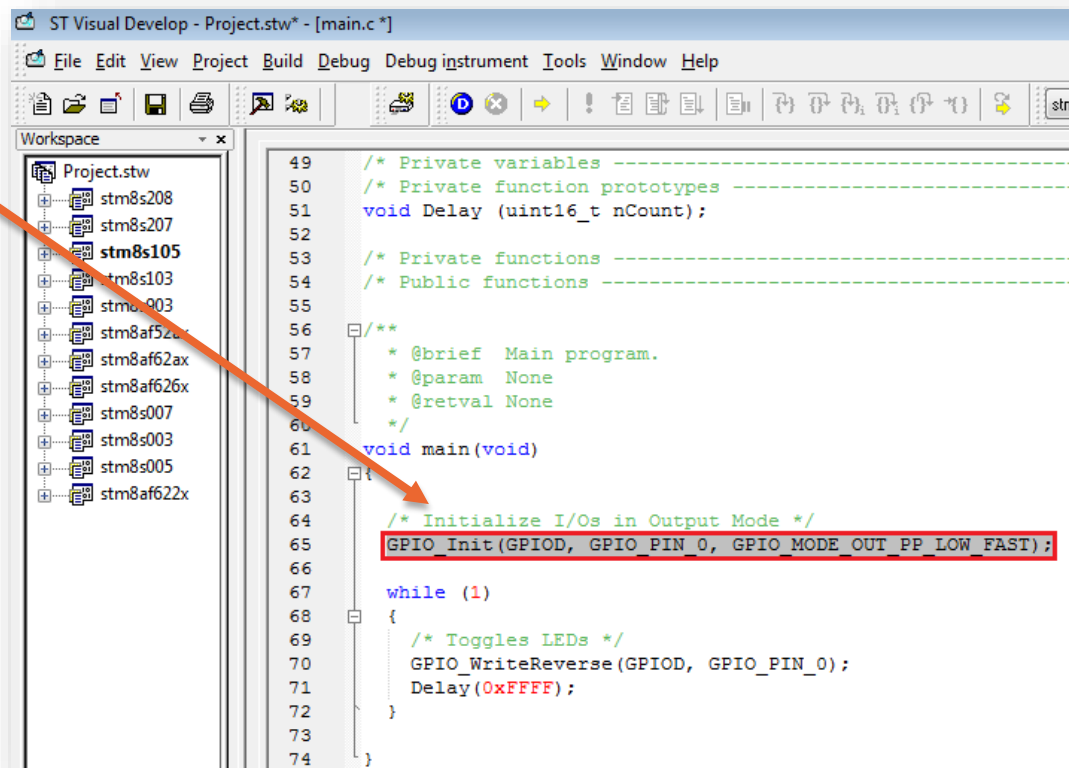
After the sentence:

/ Initialize I/Os in Output Mode */*

Modifay the line as below:

GPIO_Init(GPIOD, GPIO_PIN_0, GPIO_MODE_OUT_PP_LOW_FAST);

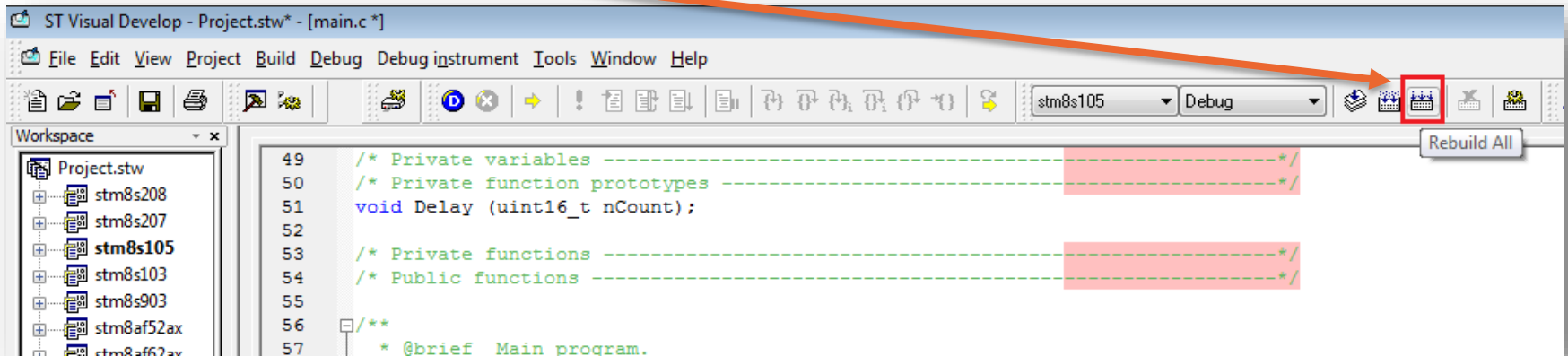
see the red box.



```
ST Visual Develop - Project.stw* - [main.c *]  
File Edit View Project Build Debug Debug instrument Tools Window Help  
Workspace  
Project.stw  
+ stm8s208  
+ stm8s207  
+ stm8s105  
+ stm8s103  
+ stm8s003  
+ stm8af52x  
+ stm8af62ax  
+ stm8af626x  
+ stm8s007  
+ stm8s003  
+ stm8s005  
+ stm8af622x  
49 /* Private variables -----  
50 /* Private function prototypes -----  
51 void Delay (uint16_t nCount);  
52  
53 /* Private functions -----  
54 /* Public functions -----  
55  
56 /**  
57  * @brief Main program.  
58  * @param None  
59  * @retval None  
60  */  
61 void main(void)  
62 {  
63  
64     /* Initialize I/Os in Output Mode */  
65     GPIO_Init(GPIOD, GPIO_PIN_0, GPIO_MODE_OUT_PP_LOW_FAST);  
66  
67     while (1)  
68     {  
69         /* Toggles LEDs */  
70         GPIO_WriteReverse(GPIOD, GPIO_PIN_0);  
71         Delay(0xFFFF);  
72     }  
73  
74 }
```

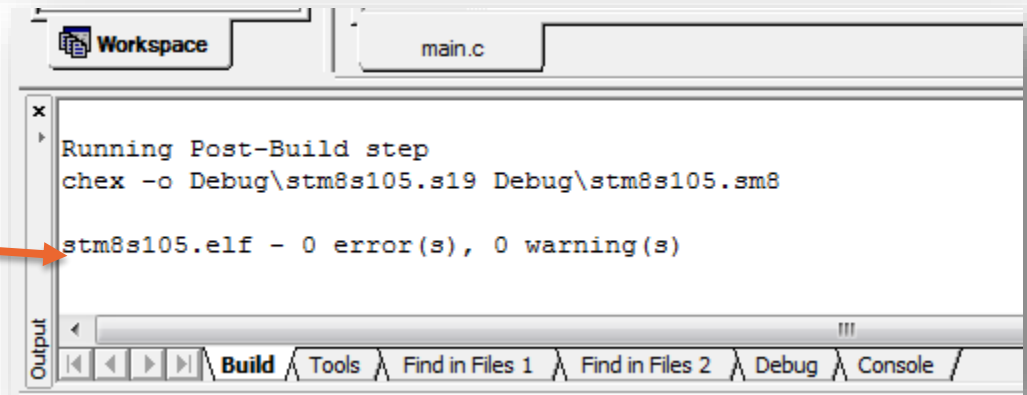
Start a new project for flashing a led 9/18

Compile all



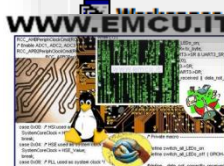
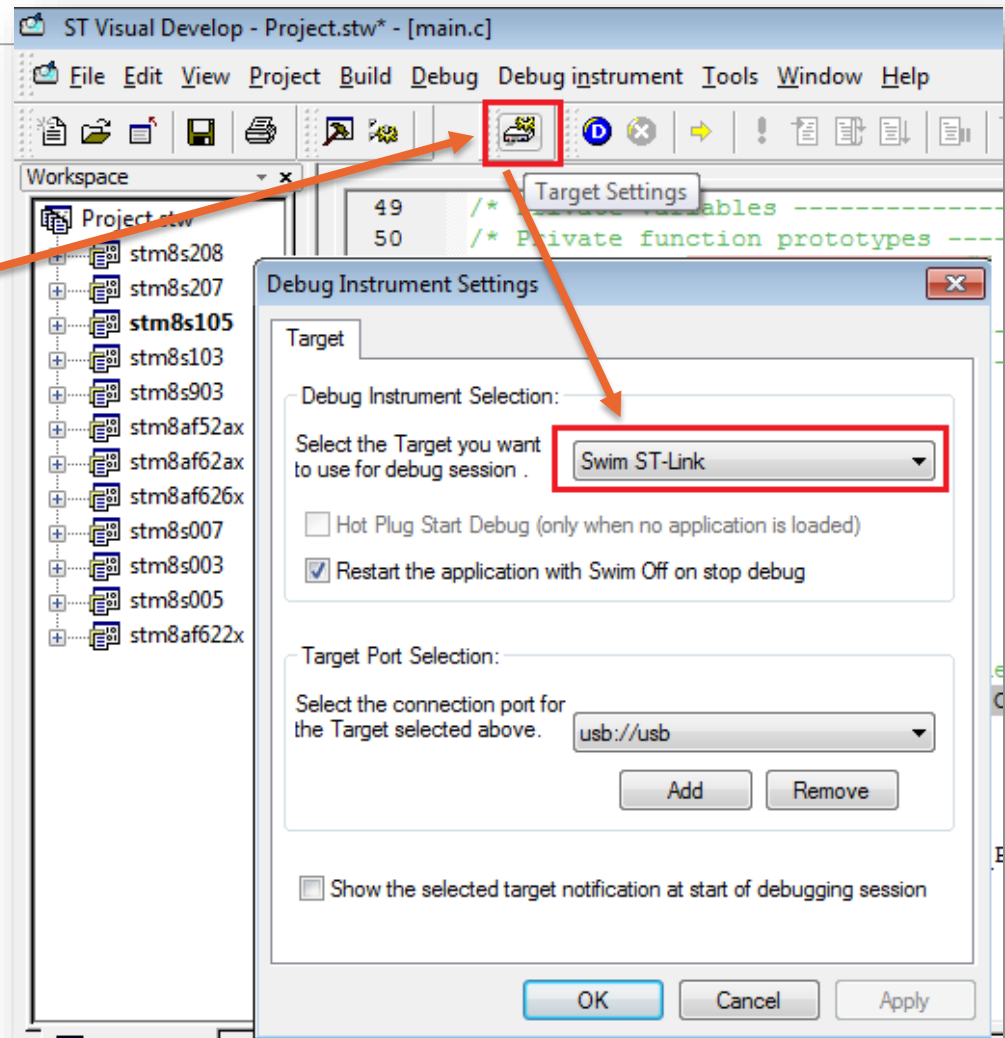
You must see:

stm8s105.elf - 0 error(s), 0 warning(s)



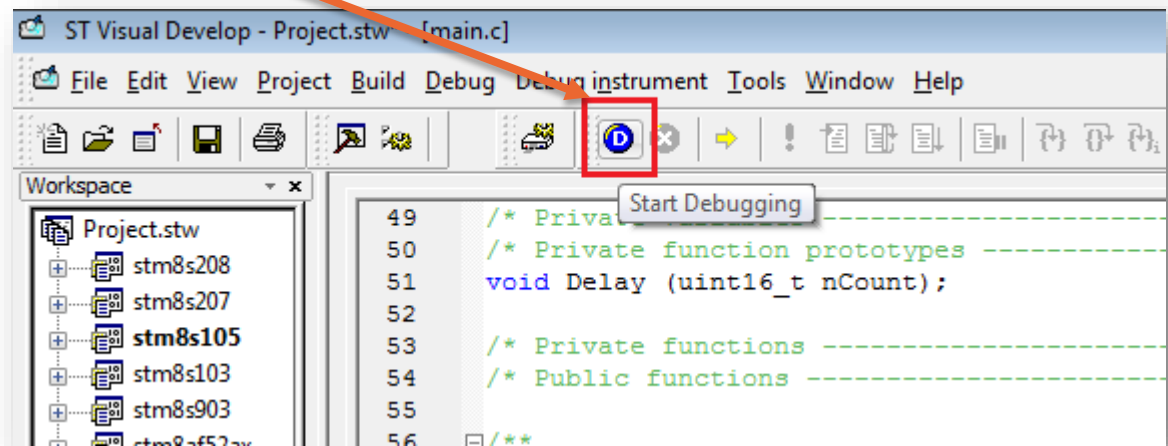
Start a new project for flashing a led 10/18

Before to do the debug, please control the following setup



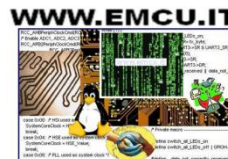
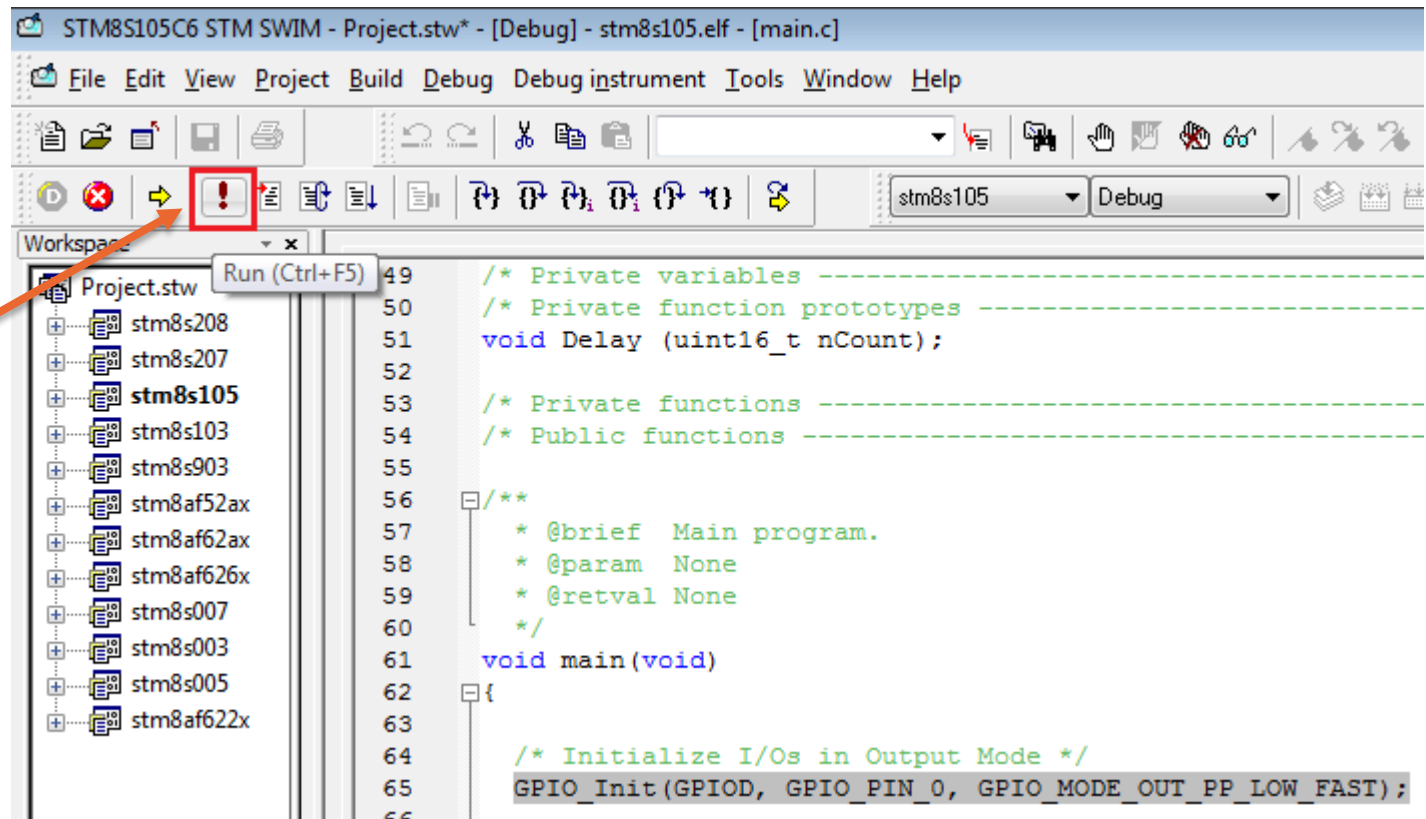
Start a new project for flashing a led 11/18

Now press on the debug button



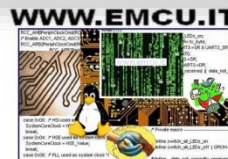
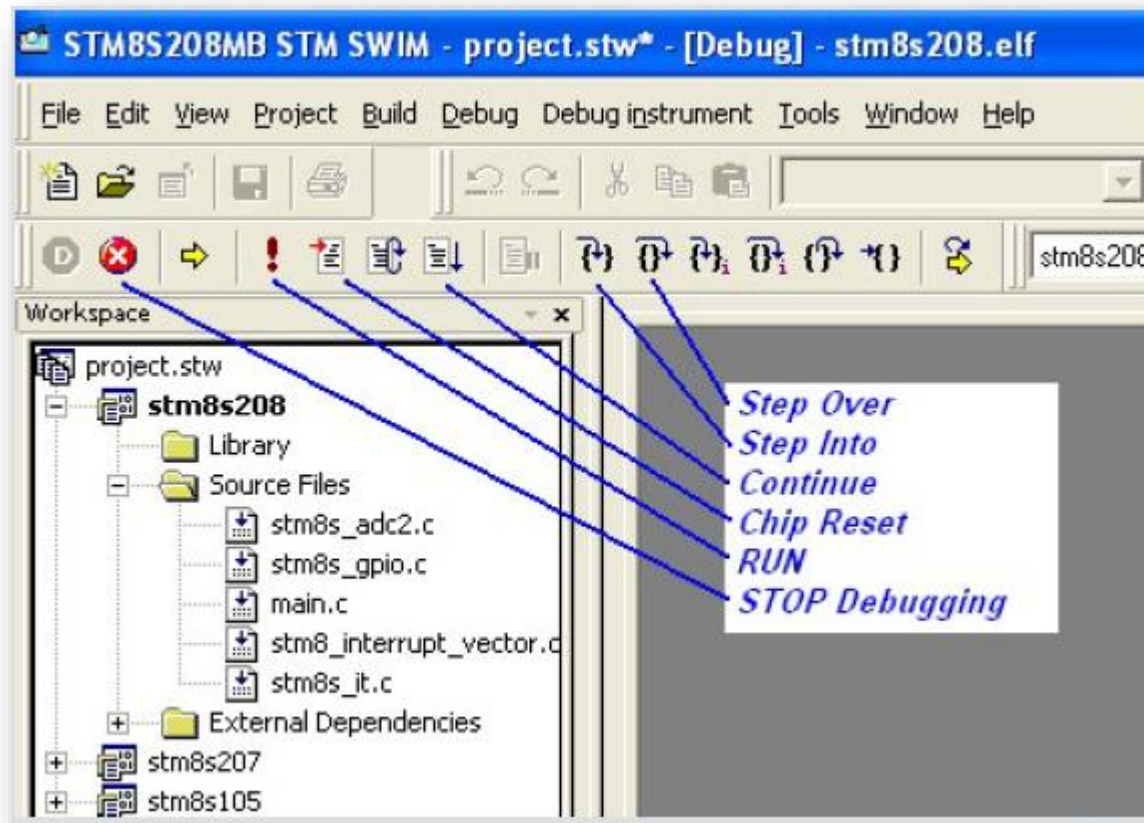
Start a new project for flashing a led 12/18

For run the program press on the RUN button. See the LED that must flashing.



Start a new project for flashing a led 13/18

Below the principal debug commands.

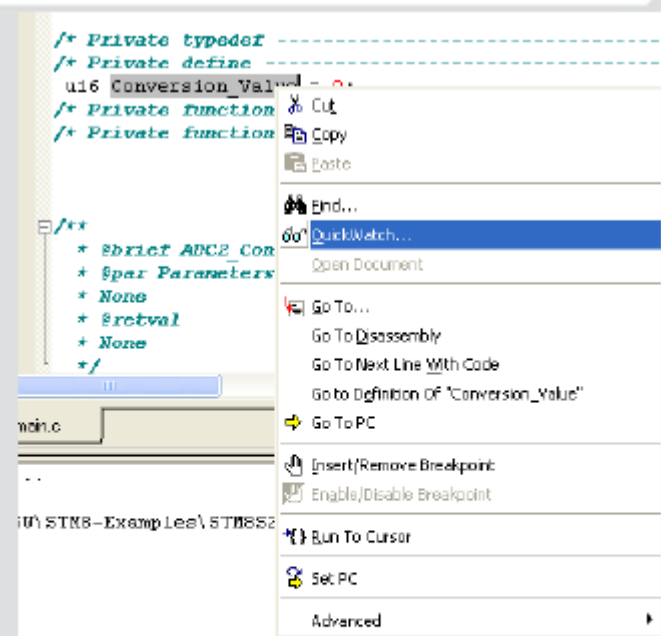
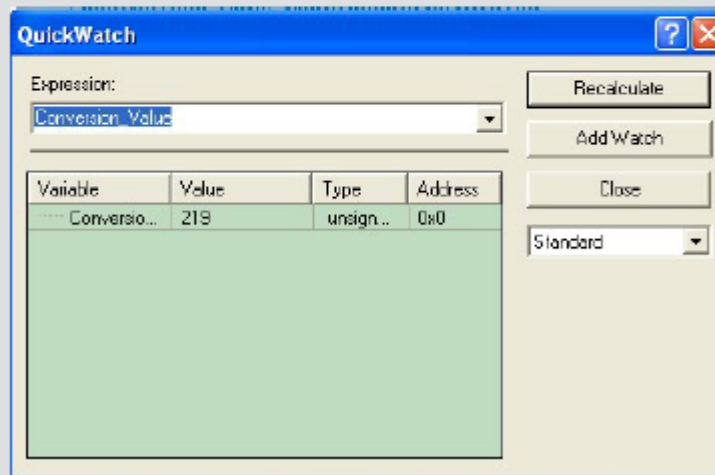


Start a new project for flashing a led 14/18

Below some debug commands, not used in this example.

Watch Variables

Select variable you want to watch
Then right click on **Quickwatch**



click **AddWatch** , a new window with variables will appear.



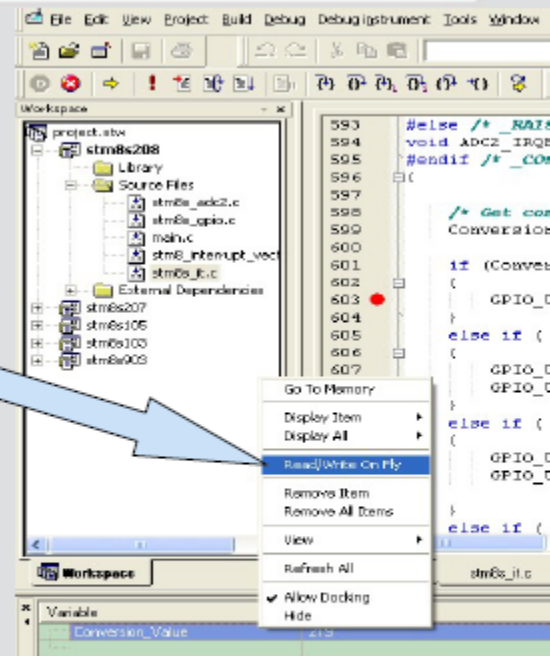
Start a new project for flashing a led 15/18

Below some debug commands, not used in this example.

Watch Variables On Fly

If you want to watch variable content in real time
You have to click on the variable and select:
ReadWrite On Fly

Variables box will change color and variables
values will change in real time



Variable	Value	Type	Address
Conversion_Value	333	unsigned short	0x0

WWW.EMCU.IT

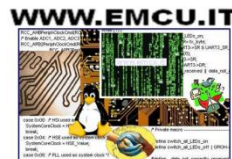
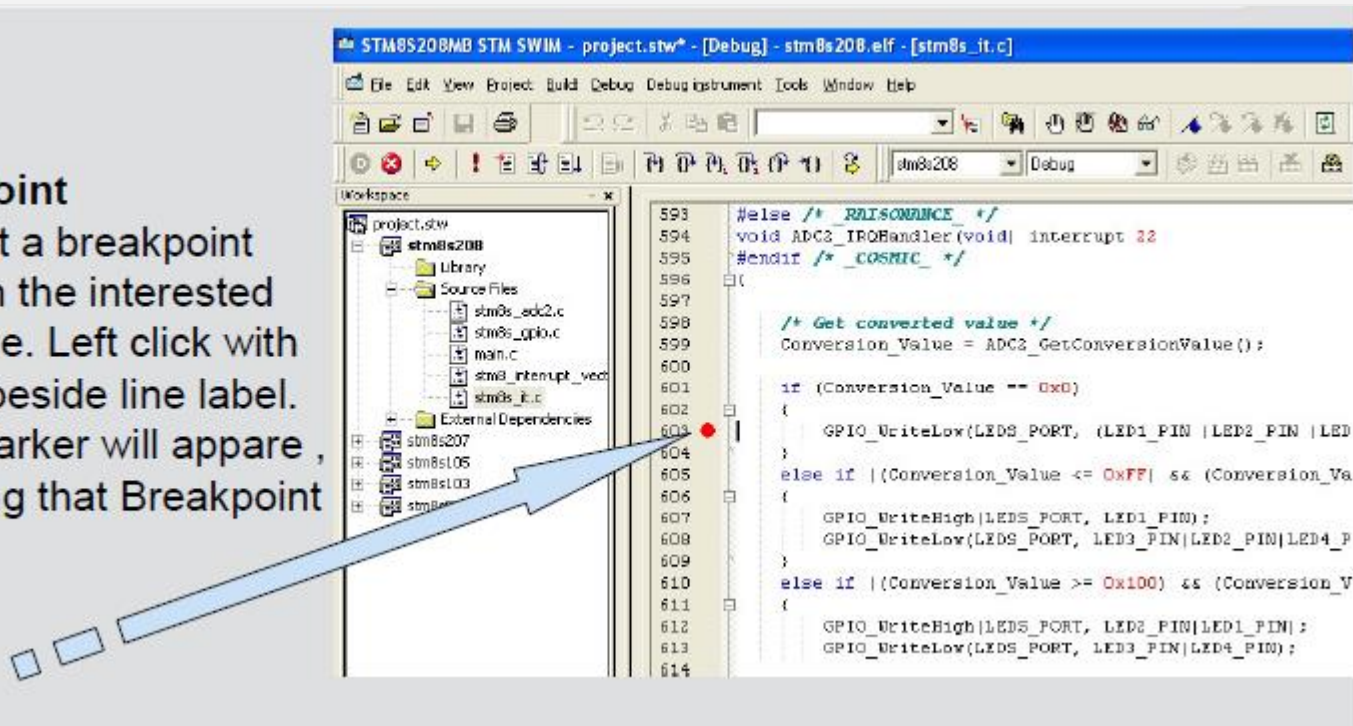


Start a new project for flashing a led 16/18

Below some debug commands, not used in this example.

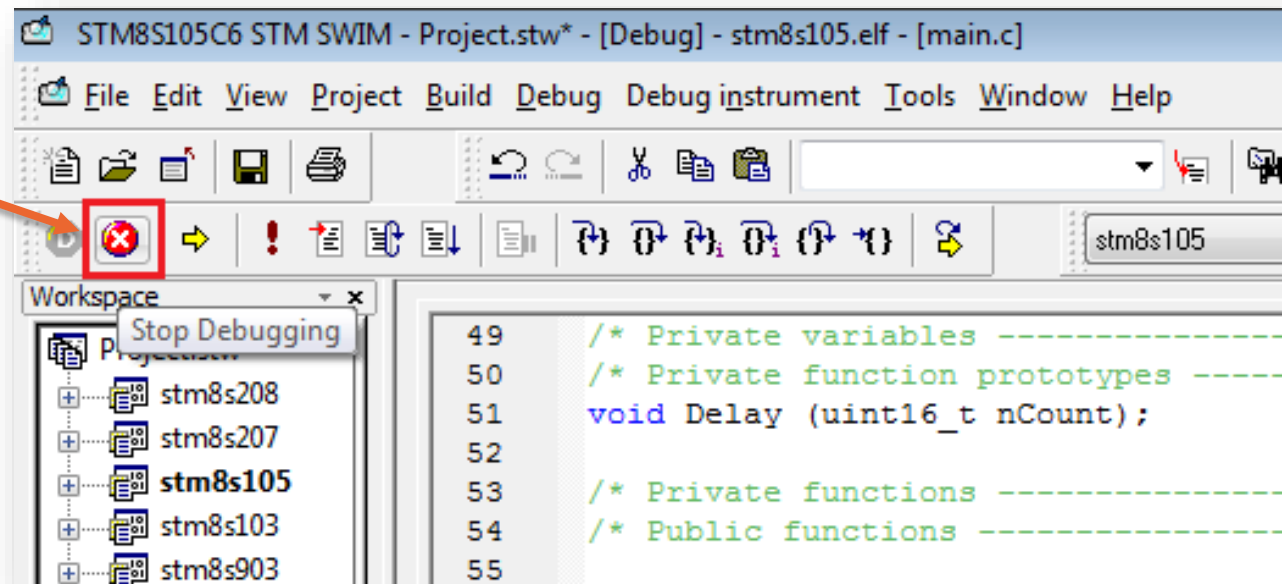
BreakPoint

To insert a breakpoint
Move on the interested
Code line. Left click with
mouse beside line label.
A red marker will appear ,
Indicating that Breakpoint
Is set.



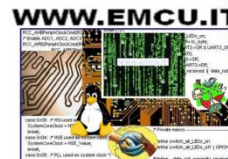
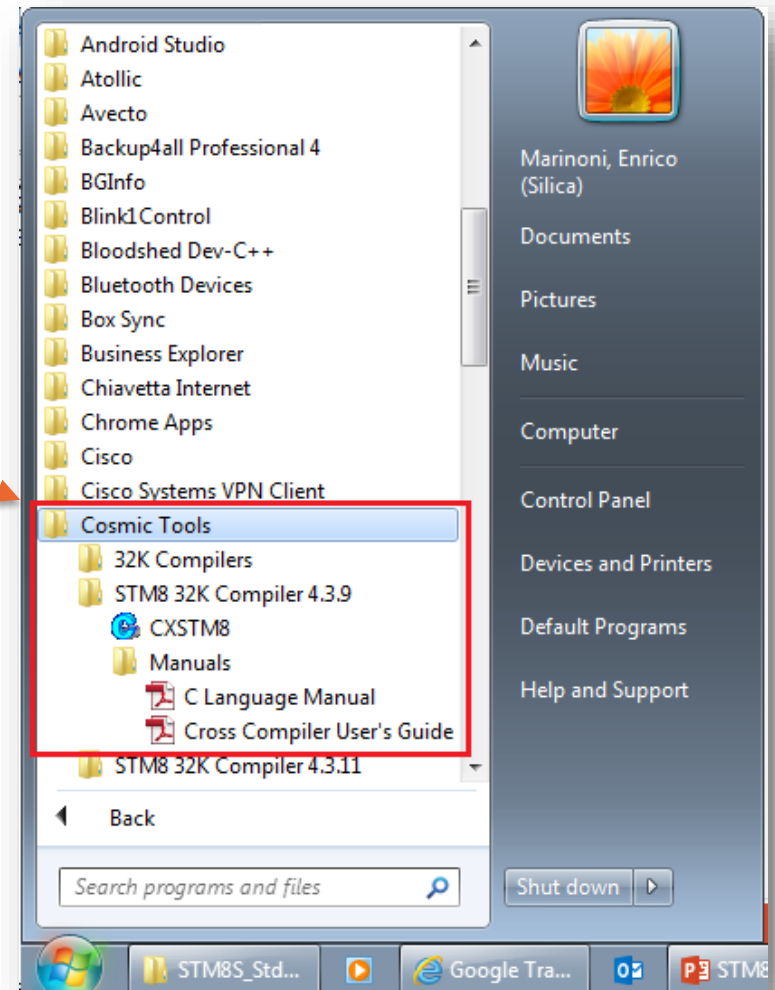
Start a new project for flashing a led 17/18

For close the debug
press on the:
Stop Debug icon



Start a new project for flashing a led 18/18

More info are available on the Cosmic documentations.



Some consideration concerning the SW 1/6

We did some changes in the toggle example because the examples are ready to use for the EvaBoard (see [here](#)).

It is very important to check the following files:

stm8s.h

stm8s_conf.h

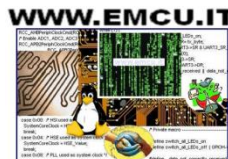
Following you will see a brief explanation of the two files.



25

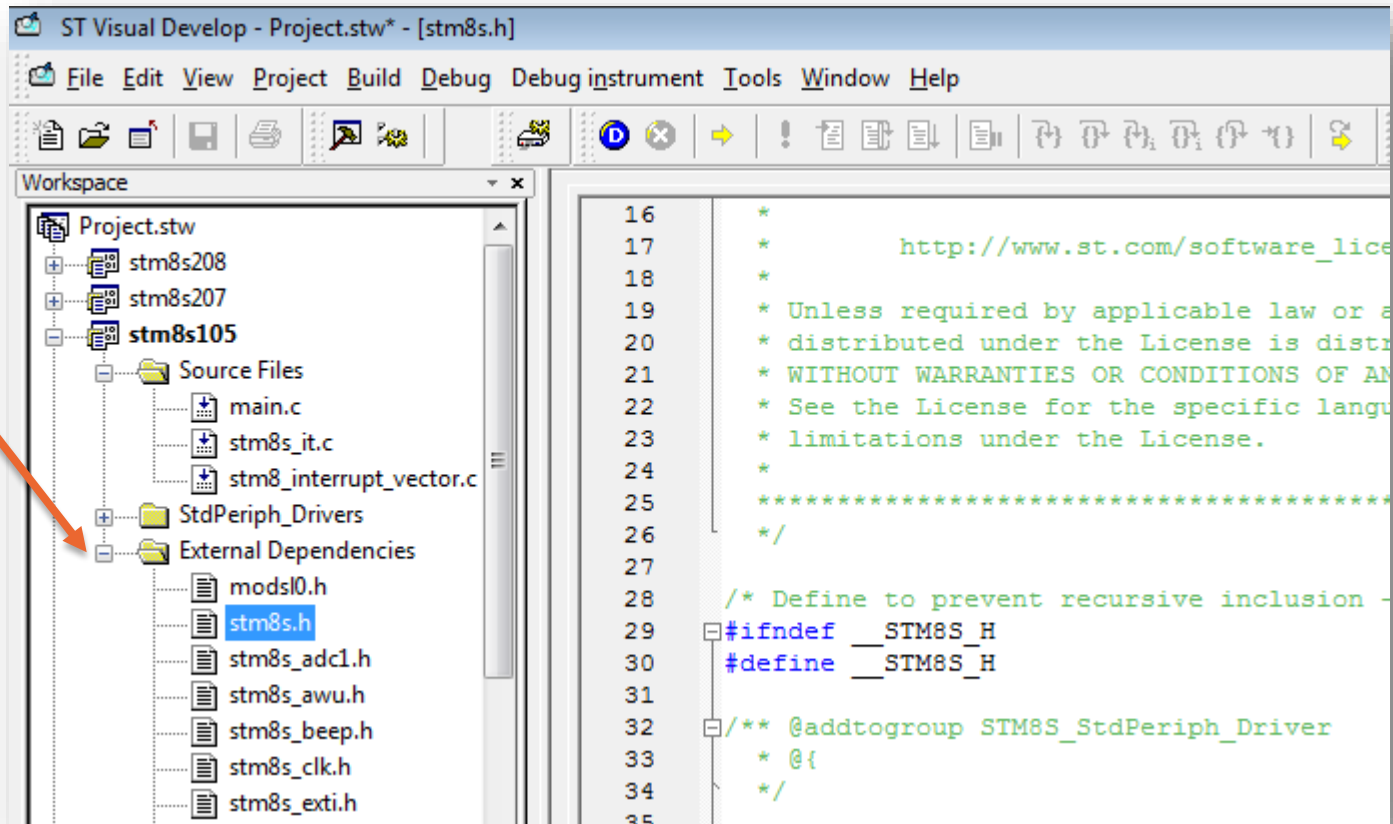


14 June 2016



Some consideration concerning the SW 2/6

The **stm8s.h** is located here



WWW.EMCU.IT



Some consideration concerning the SW 3/6

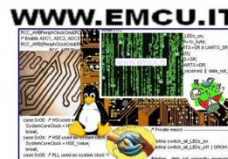
The file **stm8s.h** defines the MCU that we want to use in our project.

See the line from 39 to 50.

The SetUp in this file is supersede from the SetUp of the compiler, for this reason all the lines are commented.

In any case, is a good practice, remove from the comment the MCU that you need to use.

```
36  /* Uncomment the line below according to the target STM8S or STM8A device used in your
37  application. */
38
39  /* #define STM8S208 */          /*!< STM8S High density devices with CAN */
40  /* #define STM8S207 */          /*!< STM8S High density devices without CAN */
41  /* #define STM8S007 */          /*!< STM8S Value Line High density devices */
42  /* #define STM8AF52Ax */        /*!< STM8A High density devices with CAN */
43  /* #define STM8AF62Ax */        /*!< STM8A High density devices without CAN */
44  /* #define STM8S105 */          /*!< STM8S Medium density devices */
45  /* #define STM8S005 */          /*!< STM8S Value Line Medium density devices */
46  /* #define STM8AF626x */        /*!< STM8A Medium density devices */
47  /* #define STM8AF622x */        /*!< STM8A Low density devices */
48  /* #define STM8S103 */          /*!< STM8S Low density devices */
49  /* #define STM8S003 */          /*!< STM8S Value Line Low density devices */
50  /* #define STM8S903 */          /*!< STM8S Low density devices */
51
```



Some consideration concerning the SW 4/6

Again, in the file **stm8s.h** there are the clock definitions that are very important for serial communications and timers.

It is a must set correct the lines: 108, 110, 117 and 118.

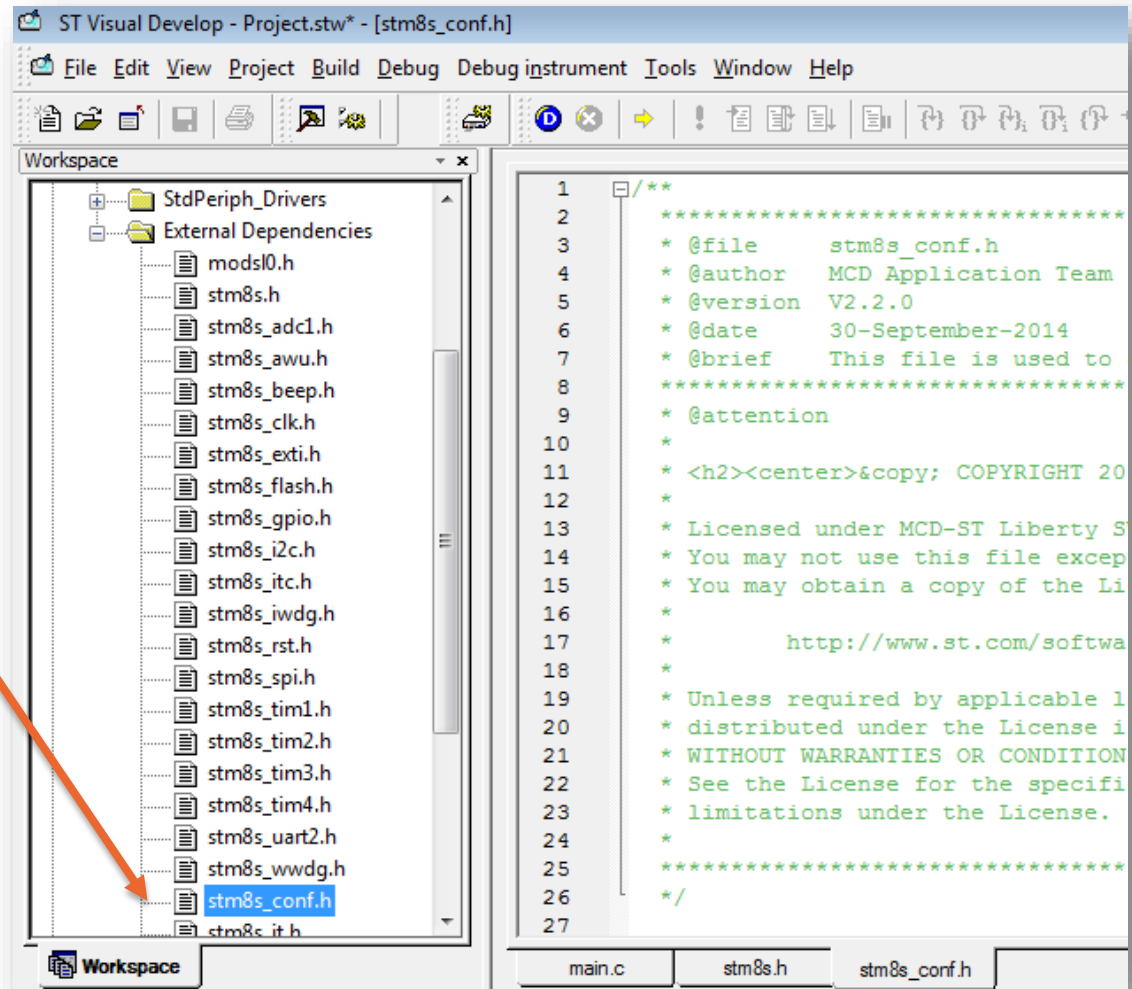
```
98  /**
99  * @brief In the following line adjust the value of External High Speed oscillator (HSE)
100  * used in your application
101
102  * Tip: To avoid modifying this file each time you need to use different HSE, you
103  * can define the HSE value in your toolchain compiler preprocessor.
104  */
105  #if !defined HSE_Value
106  #if defined (STM8S208) || defined (STM8S207) || defined (STM8S007) || defined (STM8AF52Ax) || \
107  defined (STM8AF62Ax) || defined (STM8AF622x)
108  #define HSE_VALUE ((uint32_t)24000000) /* Value of the External oscillator in Hz */
109  #else
110  #define HSE_VALUE ((uint32_t)16000000) /* Value of the External oscillator in Hz */
111  #endif /* STM8S208 || STM8S207 || STM8S007 || STM8AF62Ax || STM8AF52Ax || STM8AF622x */
112  #endif /* HSE_Value */
113
114  /**
115  * @brief Definition of Device on-chip RC oscillator frequencies
116  */
117  #define HSI_VALUE ((uint32_t)16000000) /*!< Typical Value of the HSI in Hz */
118  #define LSI_VALUE ((uint32_t)128000) /*!< Typical Value of the LSI in Hz */
119
```

WWW.EMCU.IT



Some consideration concerning the SW 5/6

The **stm8s_conf.h** is located here



WWW.EMCU.IT



Some consideration concerning the SW 6/6

The file **stm8s_conf.h** enables or disables all peripheral we want to use. Peripheral enable should be done removing comment on the interested peripheral.

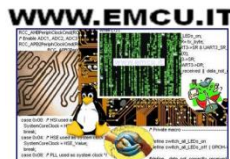
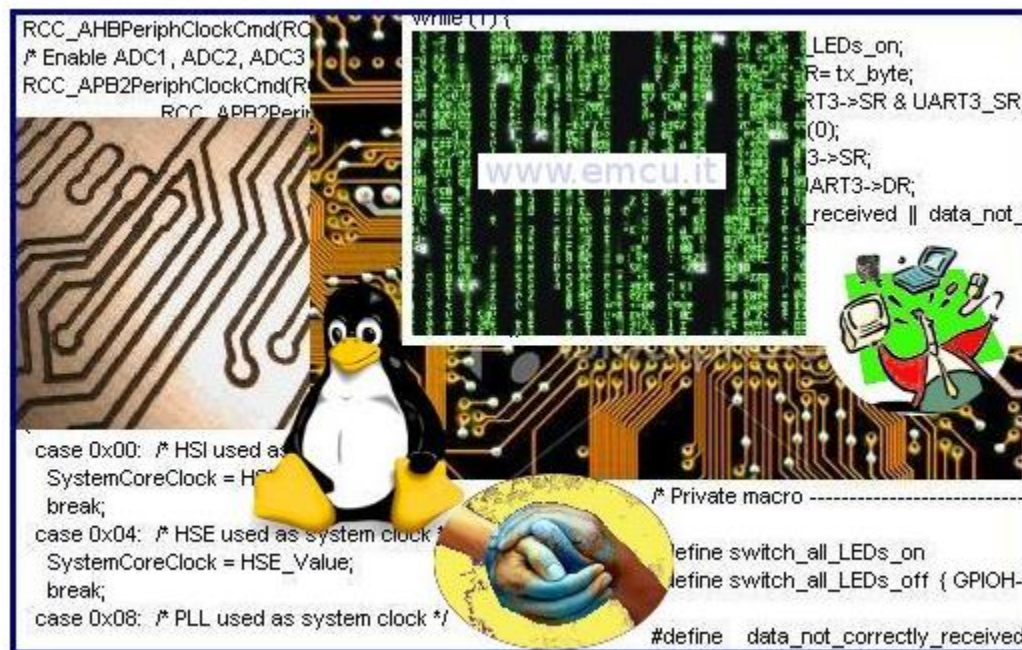
For default all peripherals are enabled.

See the file from line n.35

```
35  /* Uncomment the line below to enable peripheral header file inclusion */
36  #if defined (STM8S105) || defined (STM8S005) || defined (STM8S103) || defined (STM8S003) || \
37      defined (STM8S903) || defined (STM8AF626x)
38      #include "stm8s_adc1.h"
39  #endif /* (STM8S105) || (STM8S103) || (STM8S903) || STM8AF626x */
40  #if defined (STM8S208) || defined (STM8S207) || defined (STM8S007) || defined (STM8AF52Ax) || \
41      defined (STM8AF62Ax)
42      #include "stm8s_adc2.h"
43  #endif /* (STM8S208) || (STM8S207) || (STM8AF62Ax) || (STM8AF52Ax) */
44  #include "stm8s_awu.h"
45  #include "stm8s_beep.h"
46  #if defined (STM8S208) || defined (STM8AF52Ax)
47      #include "stm8s_can.h"
48  #endif /* STM8S208 || STM8AF52Ax */
49  #include "stm8s_clk.h"
50  #include "stm8s_exti.h"
51  #include "stm8s_flash.h"
52  #include "stm8s_gpio.h"
53  #include "stm8s_i2c.h"
54  #include "stm8s_itc.h"
55  #include "stm8s_iwdg.h"
56  #include "stm8s_rst.h"
57  #include "stm8s_spi.h"
58  #include "stm8s_tim1.h"
59  #ifndef STM8S903
60      #include "stm8s_tim2.h"
61  #endif /* STM8S903 */
```

More info...

Extra info are available from [here](http://www.emcu.it).





Thank you.

WWW.EMCU.IT



32



14 June 2016



life.augmented

