



silvio93

GUIDA: INIZIARE CON STM32F100RB E KEIL

23 April 2013

Introduzione

Ho deciso di scrivere questa guida per descrivere il procedimento di configurazione e programmazione di un microcontrollore ARM Cortex M3, in particolare l' "STM32F100RBT6B", viste le molteplici complessità incontrate.

La scheda di valutazione utilizzata è [questa](#).

Mentre come compilatore è stato usato [Keil\(MKD-ARM\)](#) .

Prima di poter cominciare la stesura vera e propria del programma si deve configurare a dovere il compilatore, per fare ciò serviranno le librerie ufficiali ST reperibili a [questo link](#).

Queste librerie servono per semplificarci la vita, invece evitare di andare a modificare i singoli bit dei registri, possiamo utilizzare le funzioni proposte dal file di help (il file con estensione .chm).

D'obbligo avere sempre sotto mano il datasheet del micro e, in questo caso, può fare comodo anche quello dalla board.

[Questa](#) è la pagina del micro e [questo](#) il relativo datasheet.



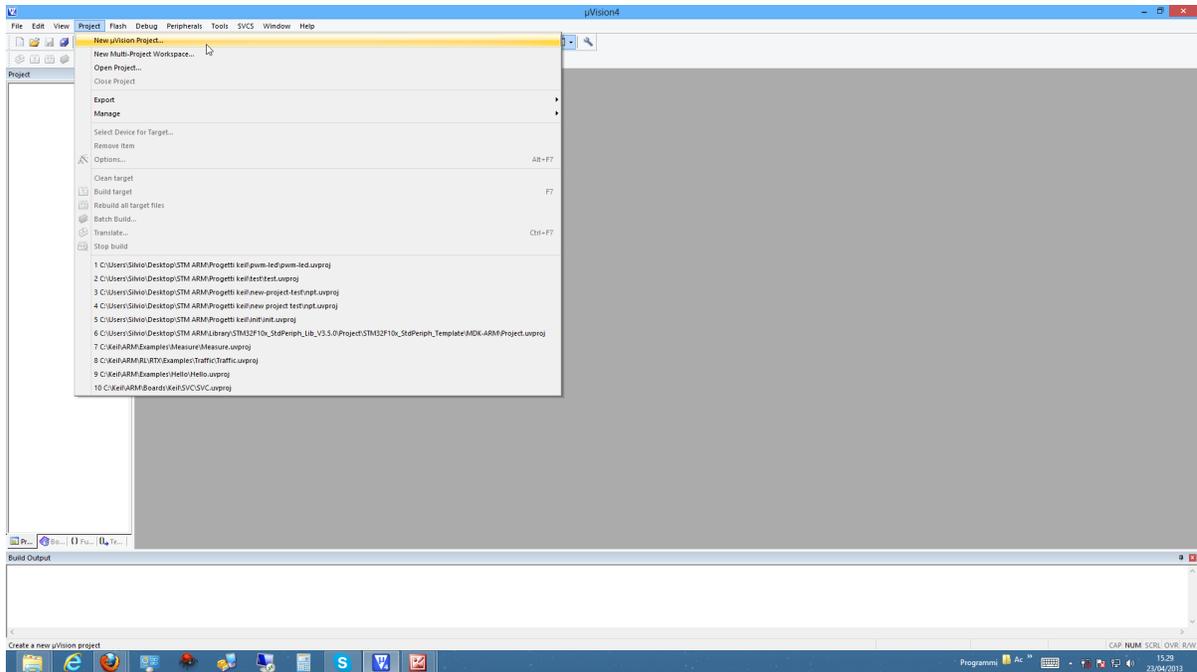
board_stm32vl_discovery.jpg

Primo Programma

Il primo programma di prova consiste nel dimmerare il LED verde(PC9) in modo da simulare un effetto fade. Quindi dopo aver scaricato tutto il necessario basterà installare ed avviare Keil uVision4.

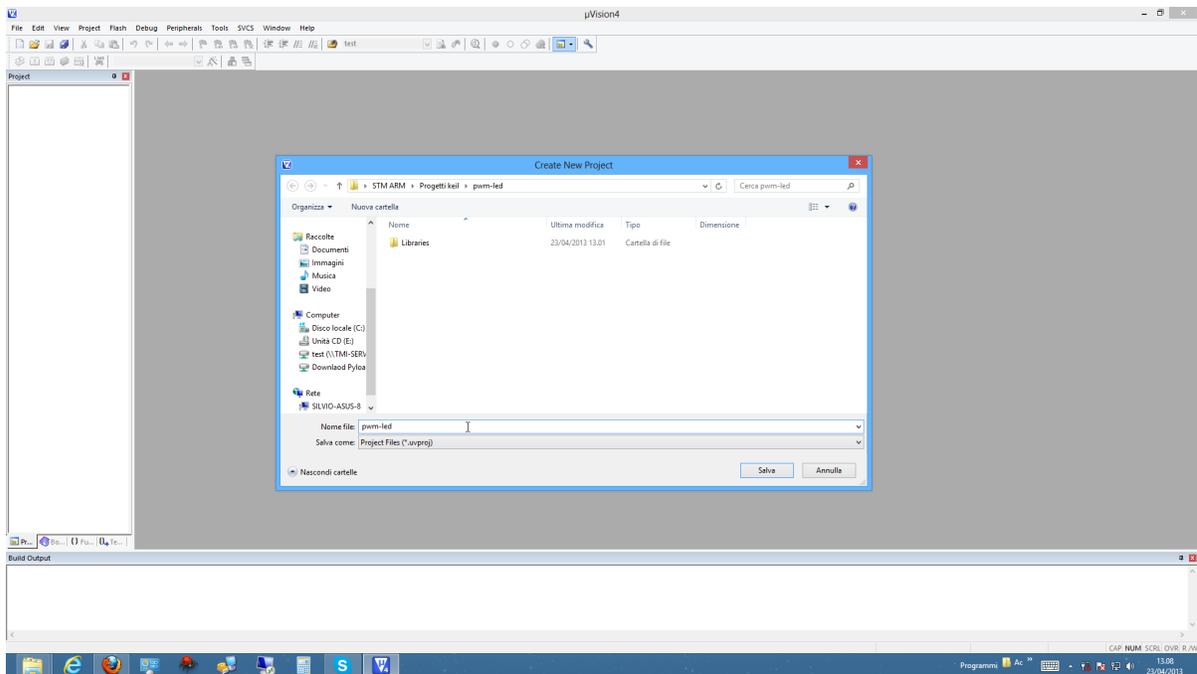
Consiglio di creare una nuova cartella per ogni progetto, in quanto il progetto prevede numerosi file. Nella cartella del progetto è consigliabile copiare ogni volta tutti i file necessari, come i file .h e .c delle librerie, nel nostro caso copiare la cartella "Libraries" presente nel pacchetto ST sopracitato. Inoltre serve il file "stm32f10x_conf.h" che si trova sempre nella libreria sotto "\\Project\\STM32F10x_StdPeriph_Template", da copiare all'interno della cartella del progetto.

Una volta aperto l'IDE vi troverete in questa schermata



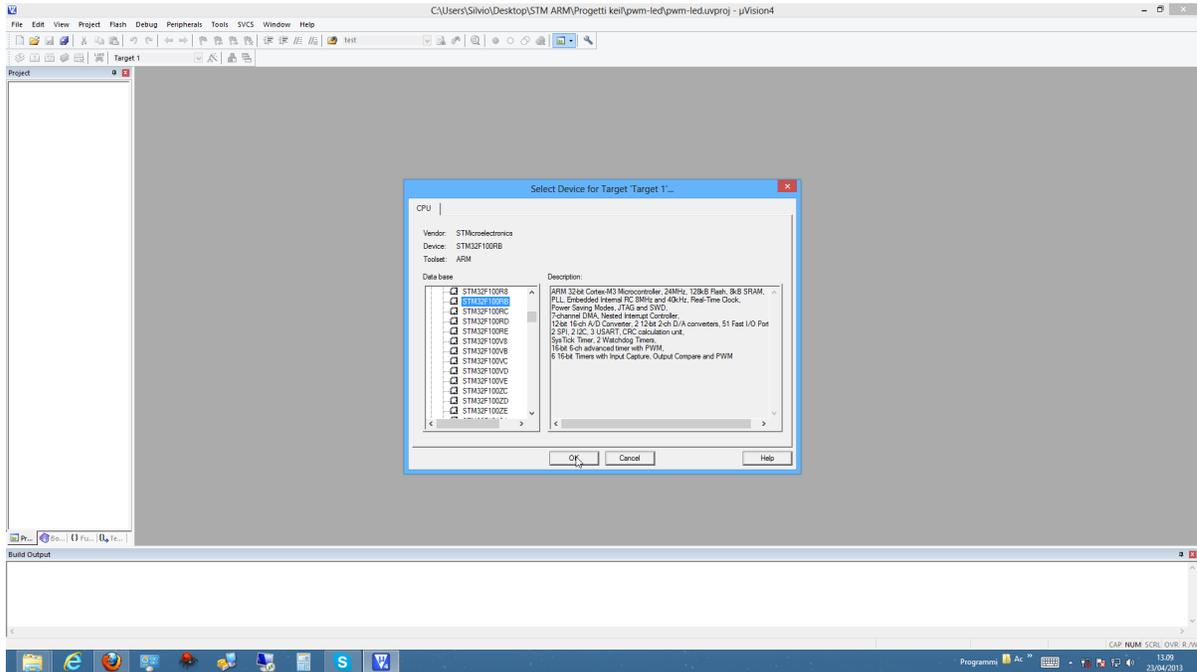
width='600' height='337'

quindi andare su **Project** poi su **New uVision project..**



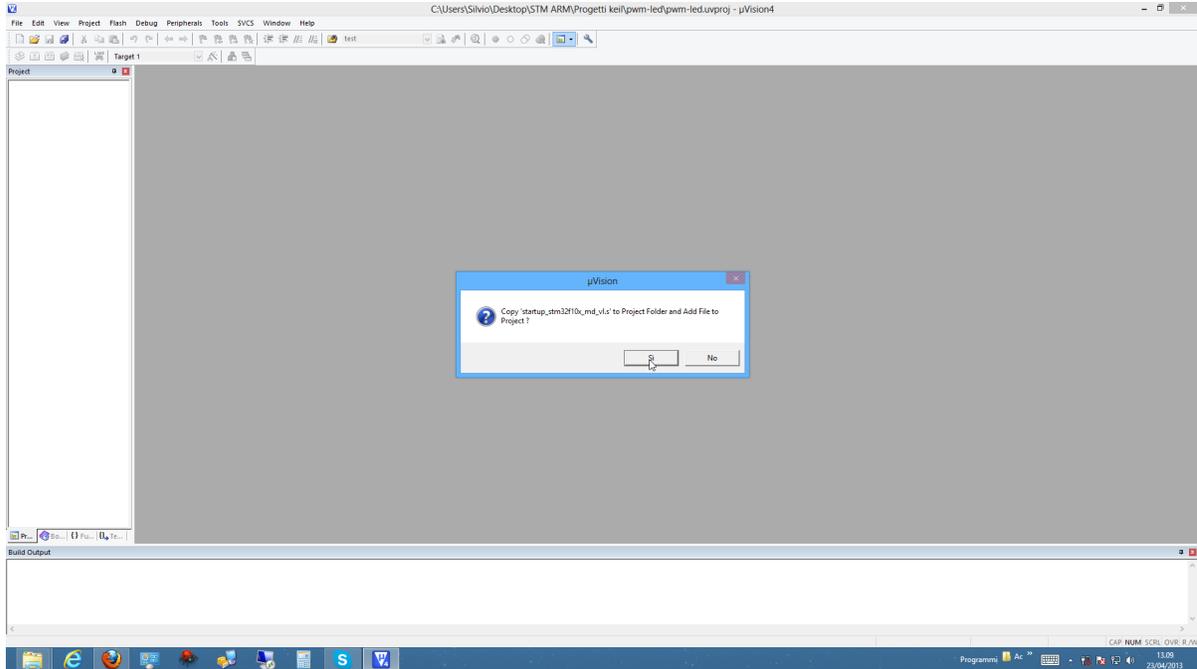
width='600' height='337'

scegliere un percorso dove salvare il progetto e dargli un nome (in questo caso pwm-led)



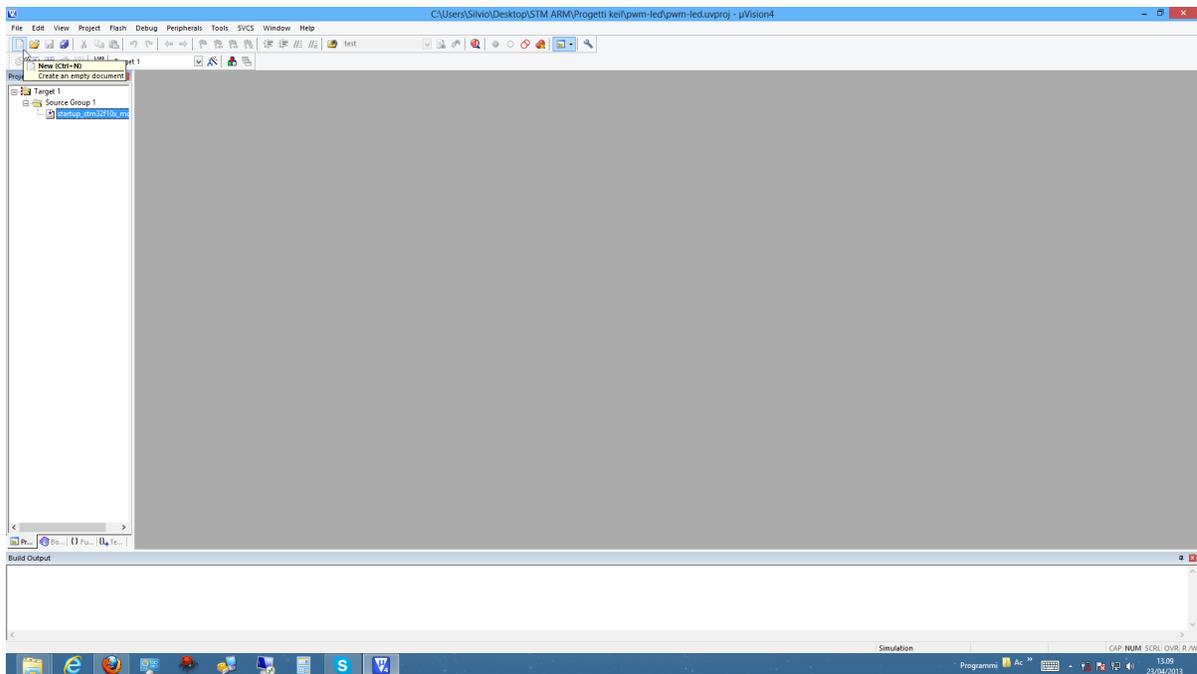
width='600' height='337'

selezionare il micro, in questo caso **STM32F100RB**



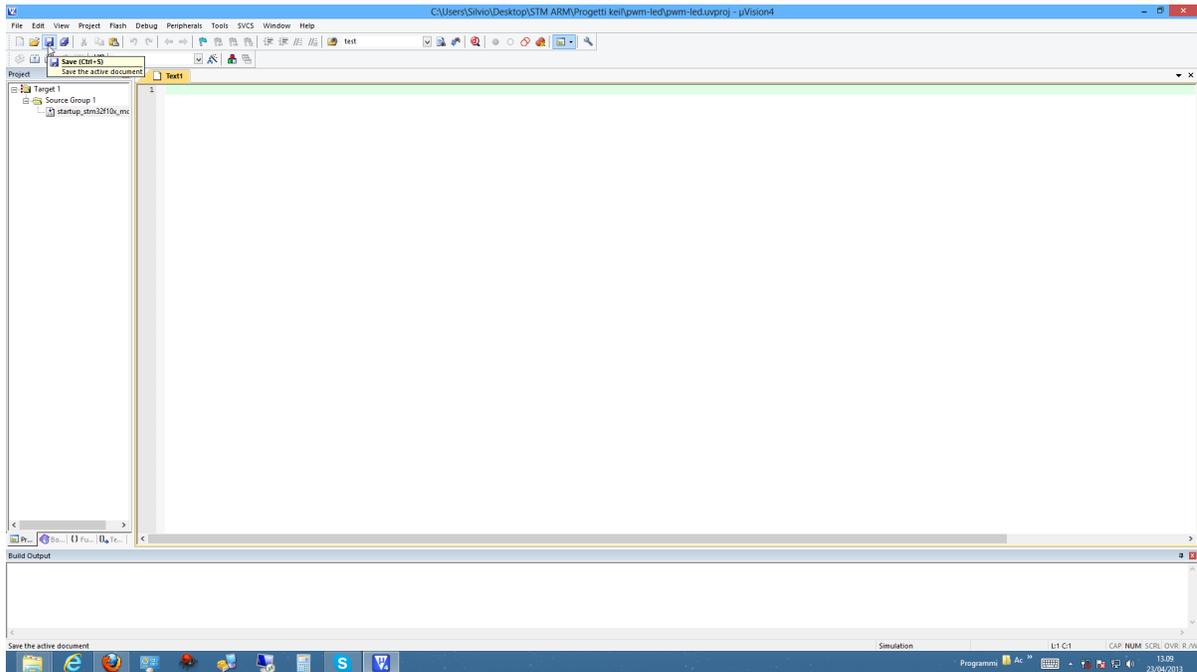
width='600' height='337'

scegliere **Si** per aggiungere il file di startup



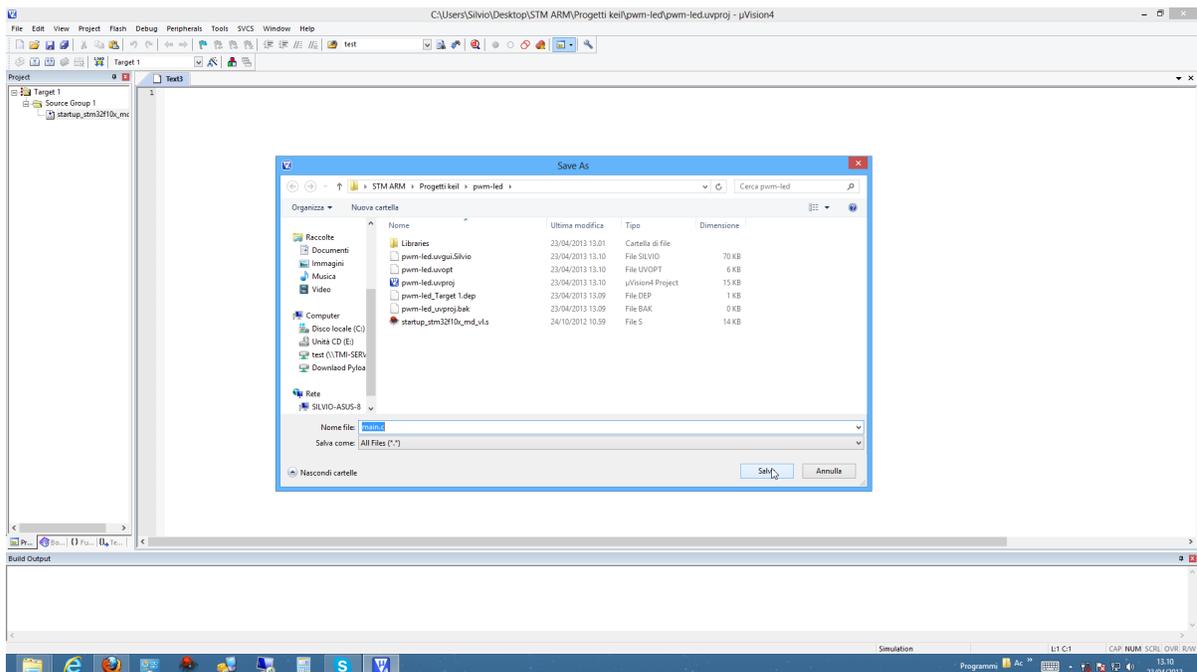
width='600' height='337'

creiamo un nuovo file premendo **New**



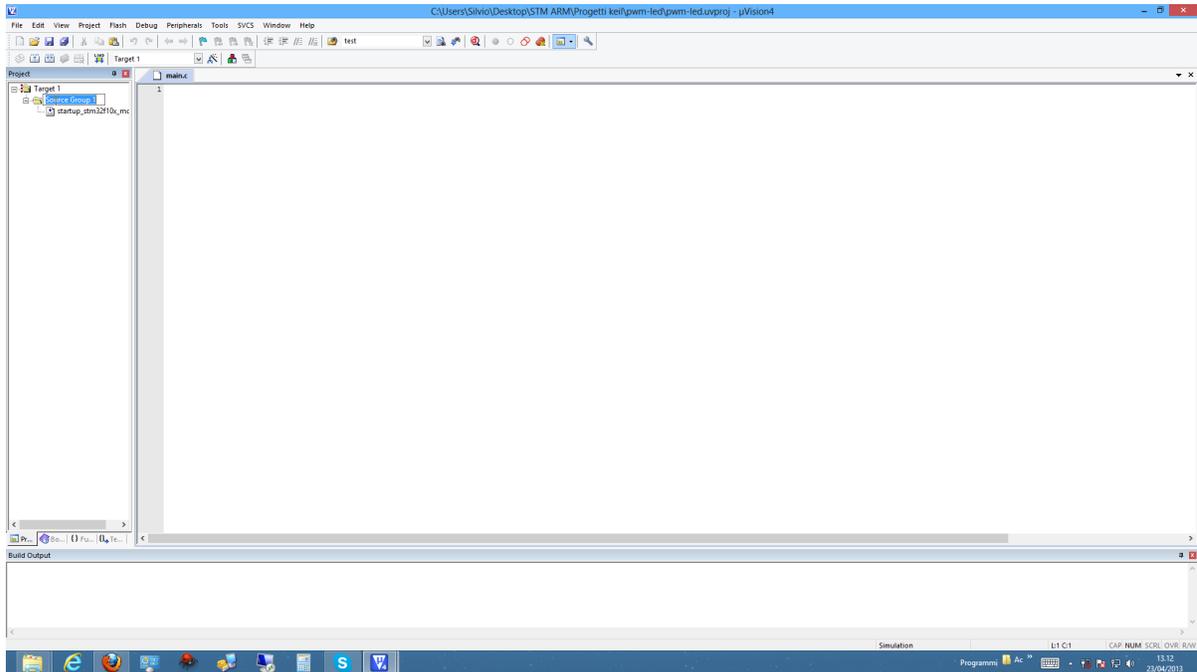
width='600' height='337'

salviamo il file appena creato



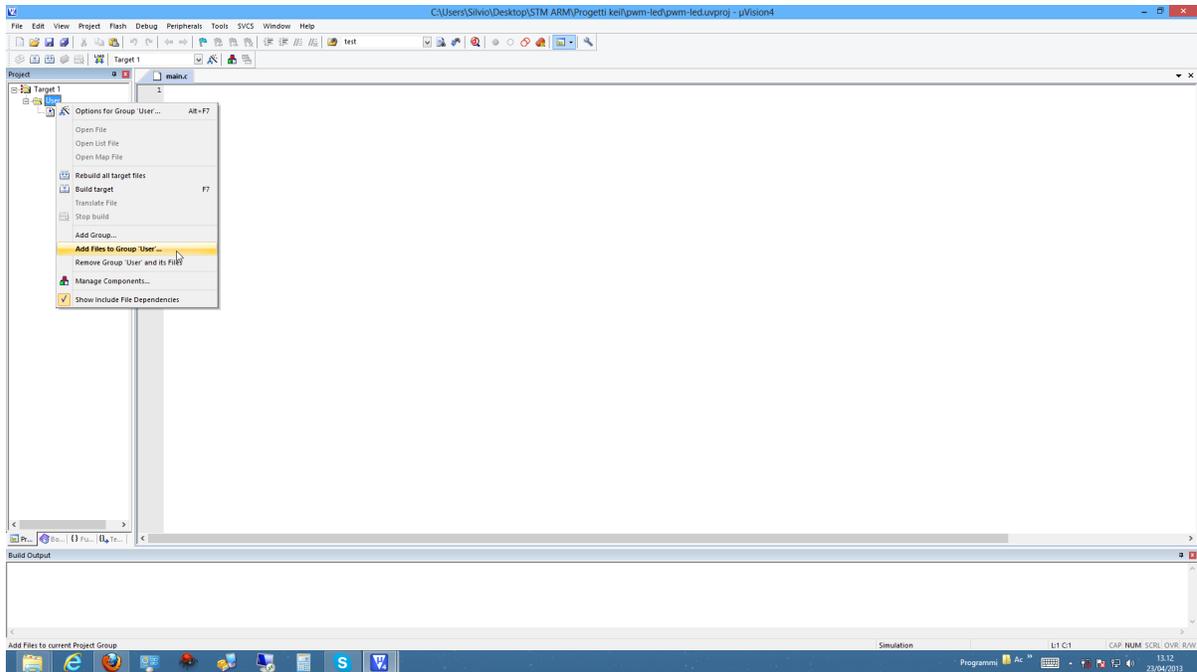
width='600' height='337'

e chiamiamolo **main.c**



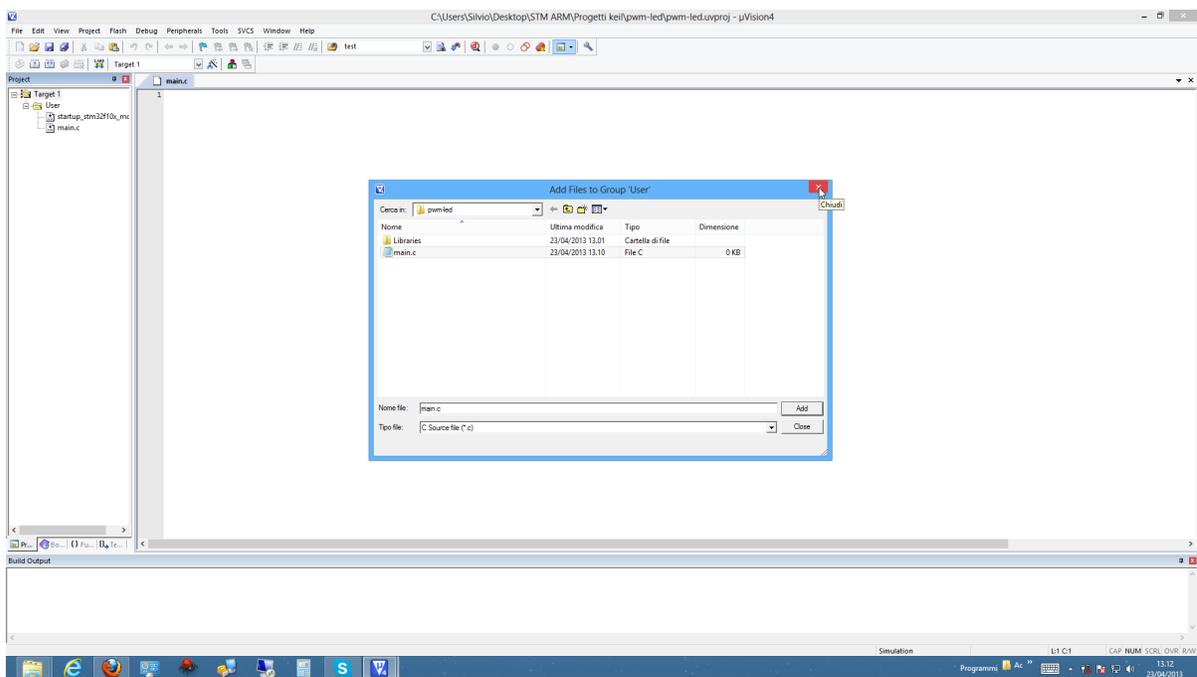
width='600' height='337'

nella finestra di sinistra (Project) possiamo rinominare i gruppi a piacimento (premendo f2), in questo caso **User**



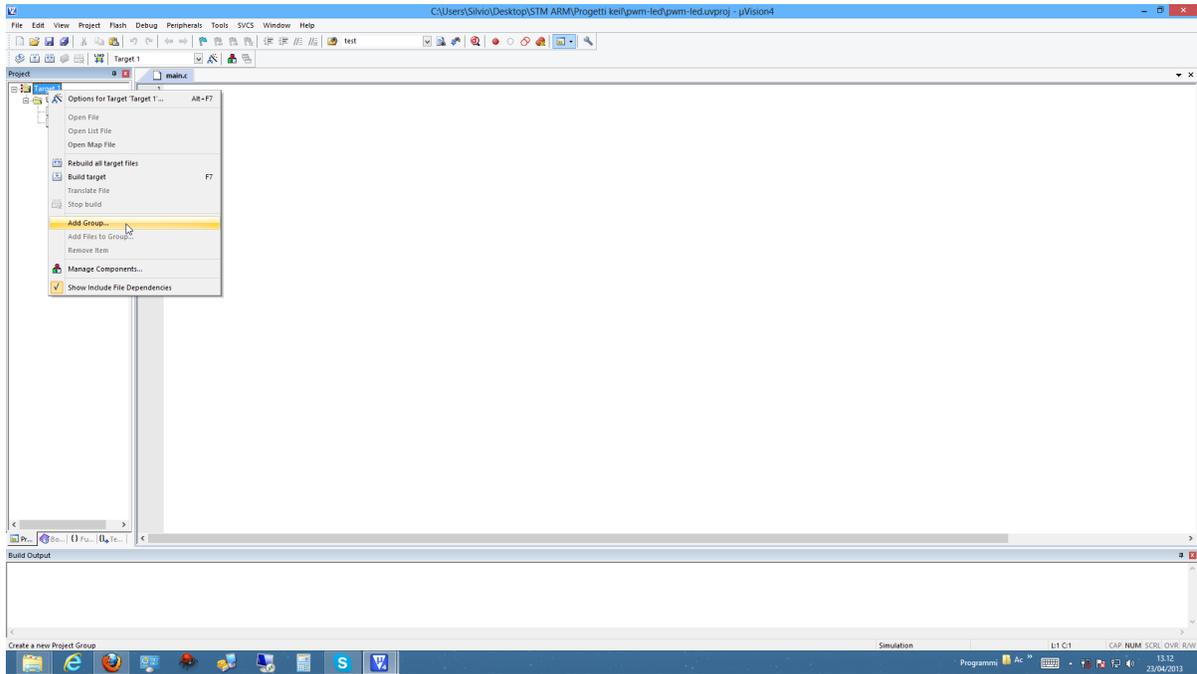
width='600' height='337'

aggiungiamo altri file .c al gruppo user tramite **click dx -> Add Files to Group 'User'...**



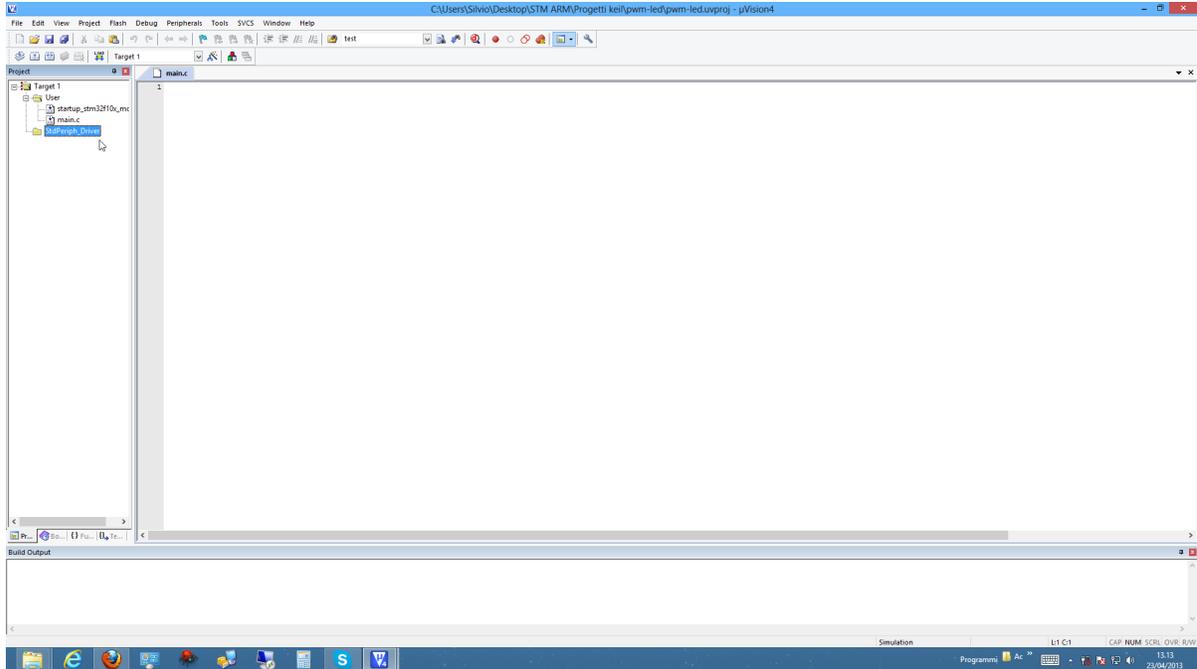
width='600' height='337'

aggiungiamo il file precedentemente creato **main.c**



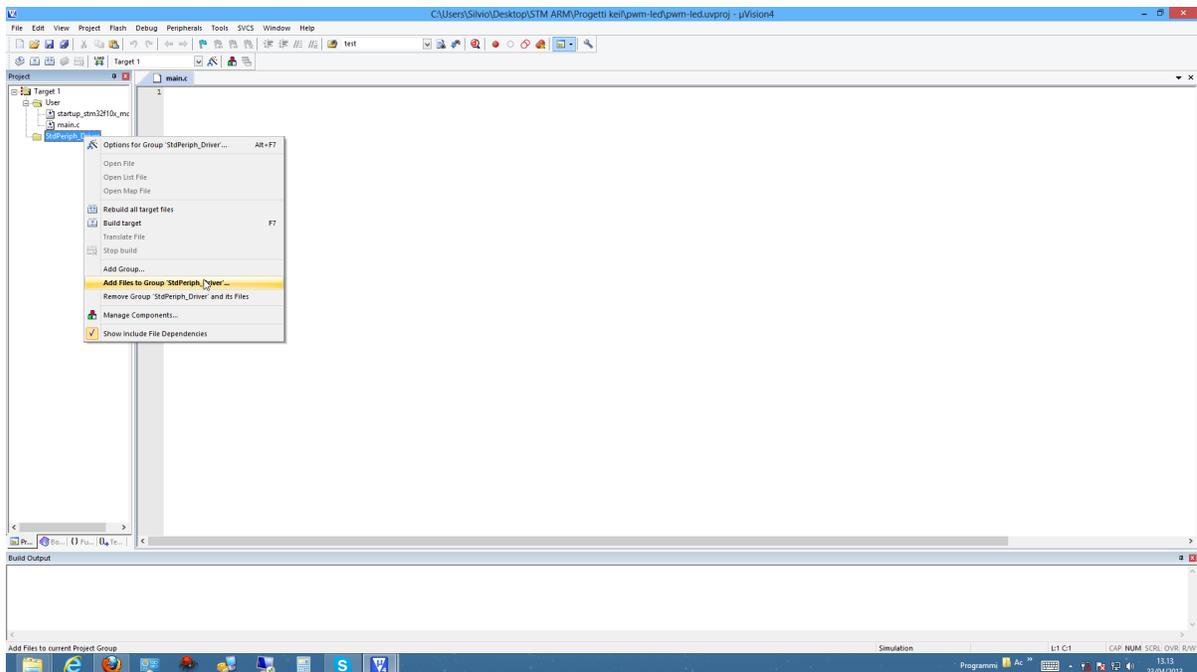
width='600' height='337'

aggiungiamo un nuovo gruppo da **Target 1** -> **click dx** -> **Add Gruop...**



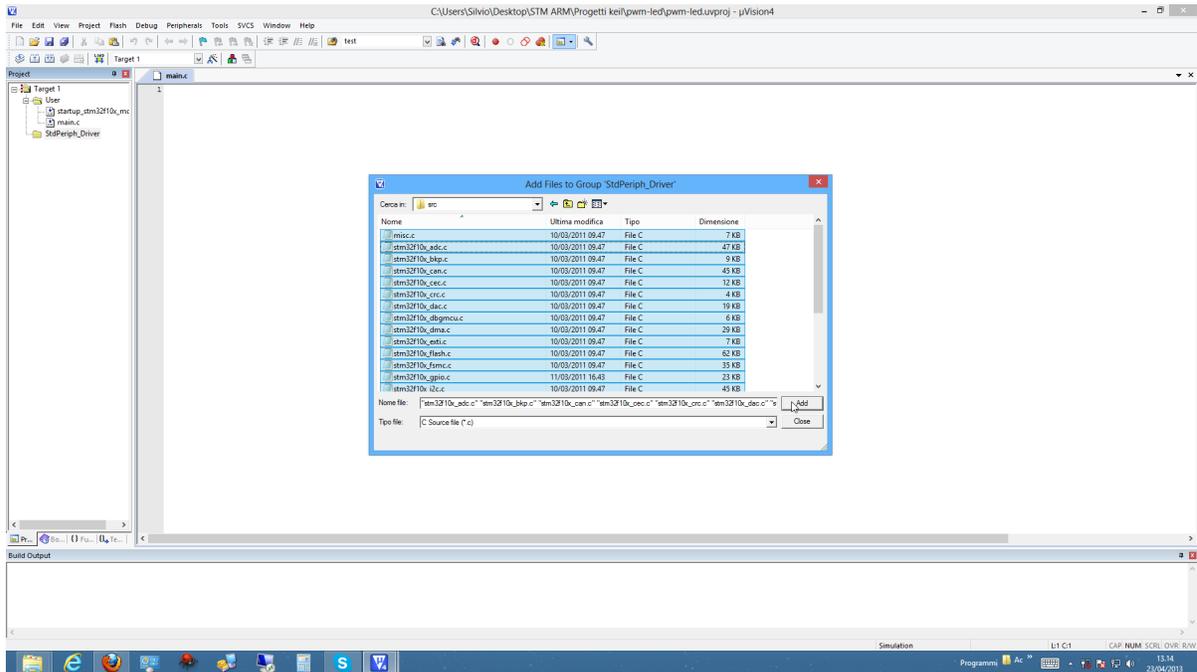
width='600' height='337'

chiamiamolo **StdPeriph_Driver** (i nomi dei gruppi sono soggettivi)



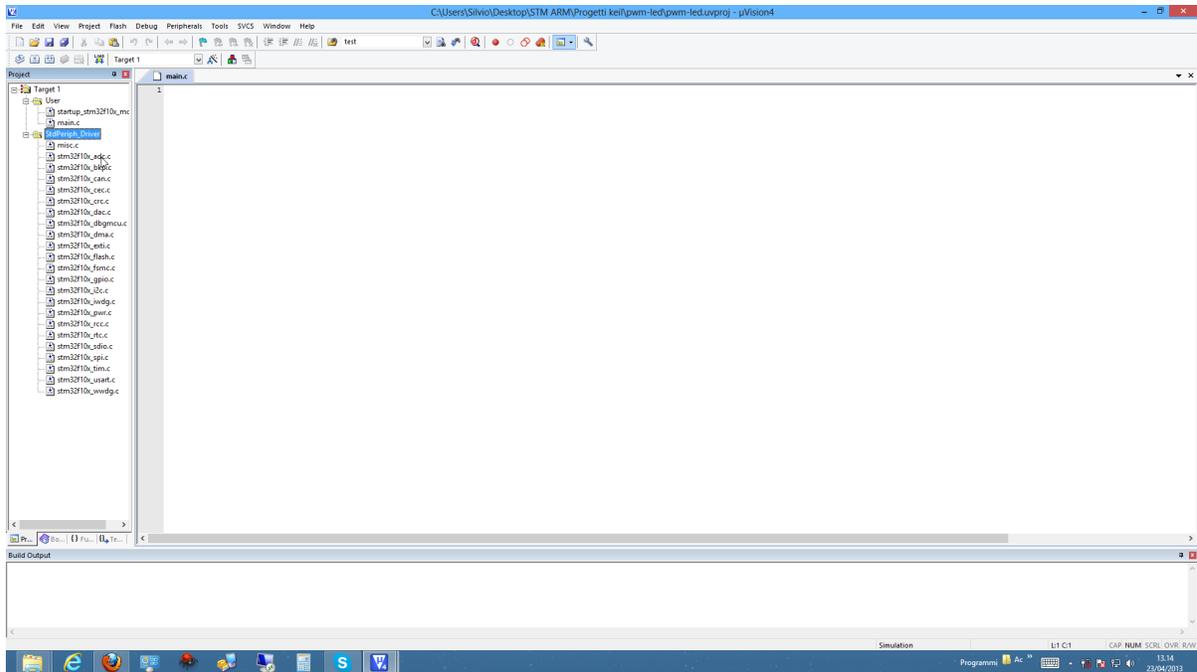
width='600' height='337'

per aggiungere file al gruppo, come sopra, **click dx sul gruppo -> Add Files to Group..**



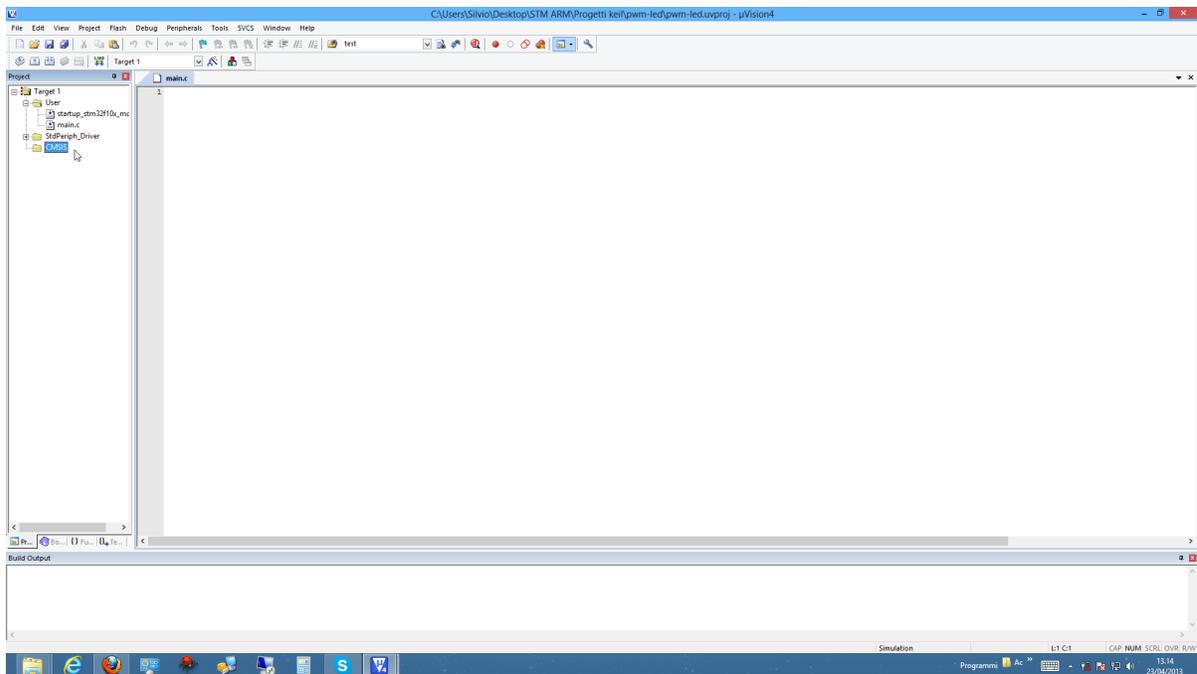
width='600' height='337'

aggiungiamo tutti i file, sempre con estensione .c, del percorso **\Libraries\STM32F10x_StdPeriph_Driver\src**



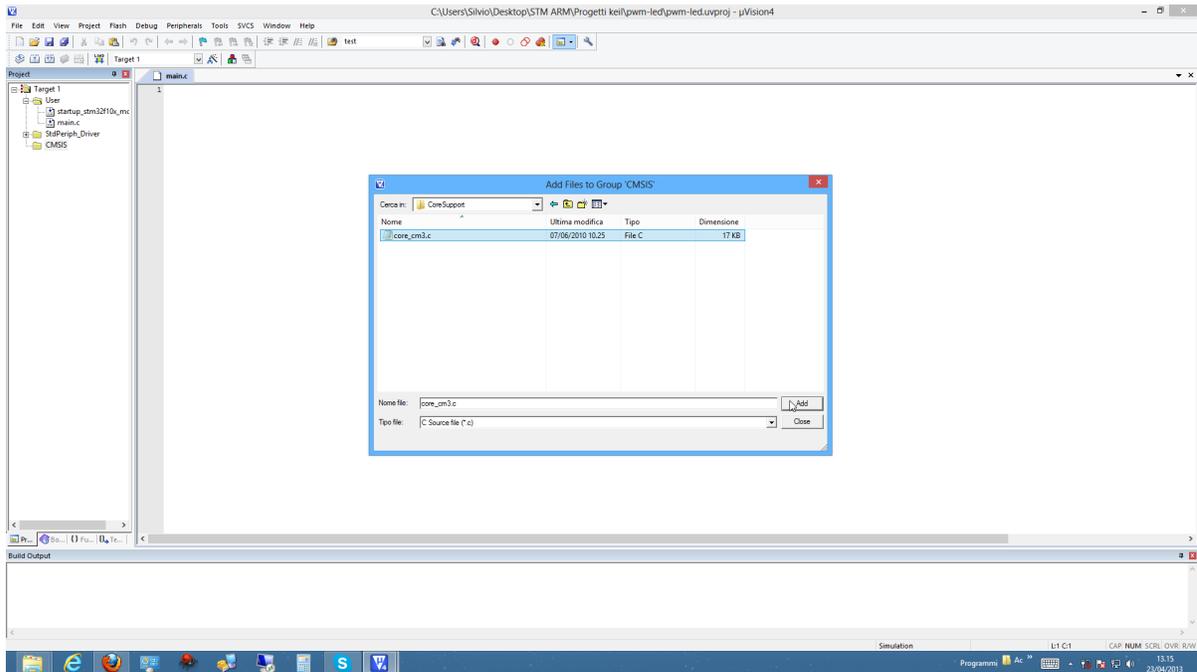
width='600' height='337'

questo dovrebbe essere il risultato



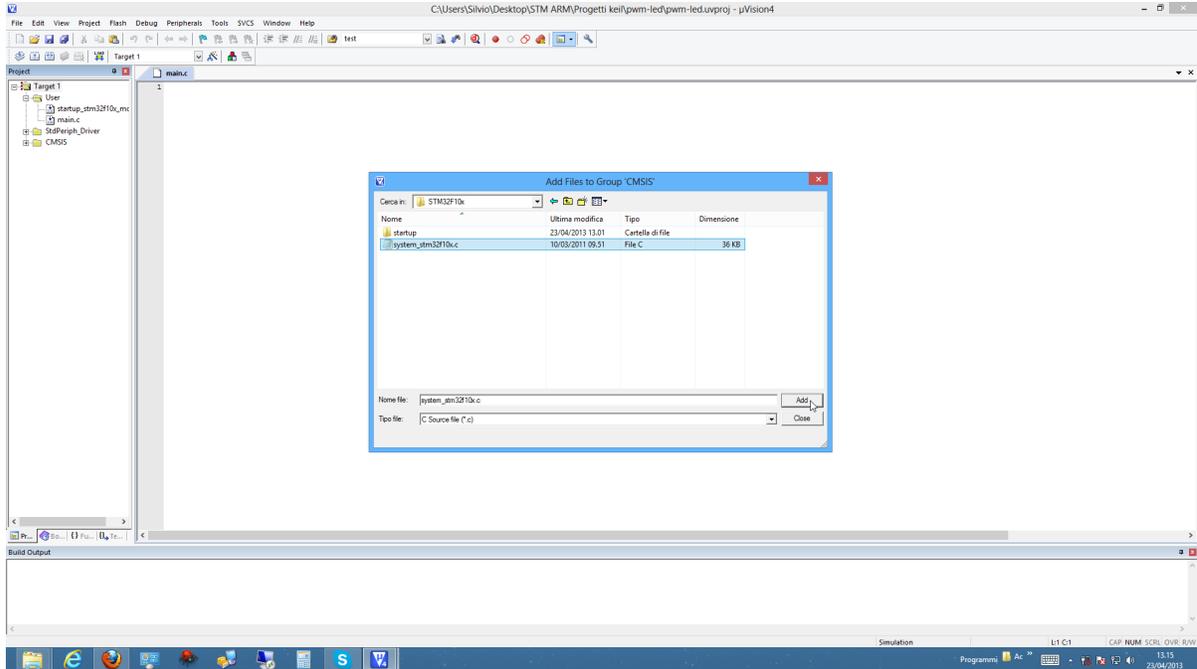
width='600' height='337'

aggiungiamo un altro gruppo e chiamiamolo **CMSIS**



width='600' height='337'

aggiungiamo il file **core_cm3.c** che si trova in **\Libraries\CMSIS\CM3\CoreSupport**



width='600' height='337'

e il file **system_stm32f10x.c** che si trova in **\Libraries\CMSIS\CM3\DeviceSupport\ST\STM32F10x**

A questo punto si può copiare il codice nel main.c

```

#include "stm32f10x.h"

TIM_TimeBaseInitTypeDef  TIM_TimeBaseStructure;
TIM_OCInitTypeDef  TIM_OCInitStructure;

uint16_t CCR1_Val = 0;
int check=0;
uint16_t PrescalerValue = 0;
int fade_time = 0x1FFFF;

/* Function Definition */
void RCC_Configuration(void);
void GPIO_Configuration(void);

int main(void){

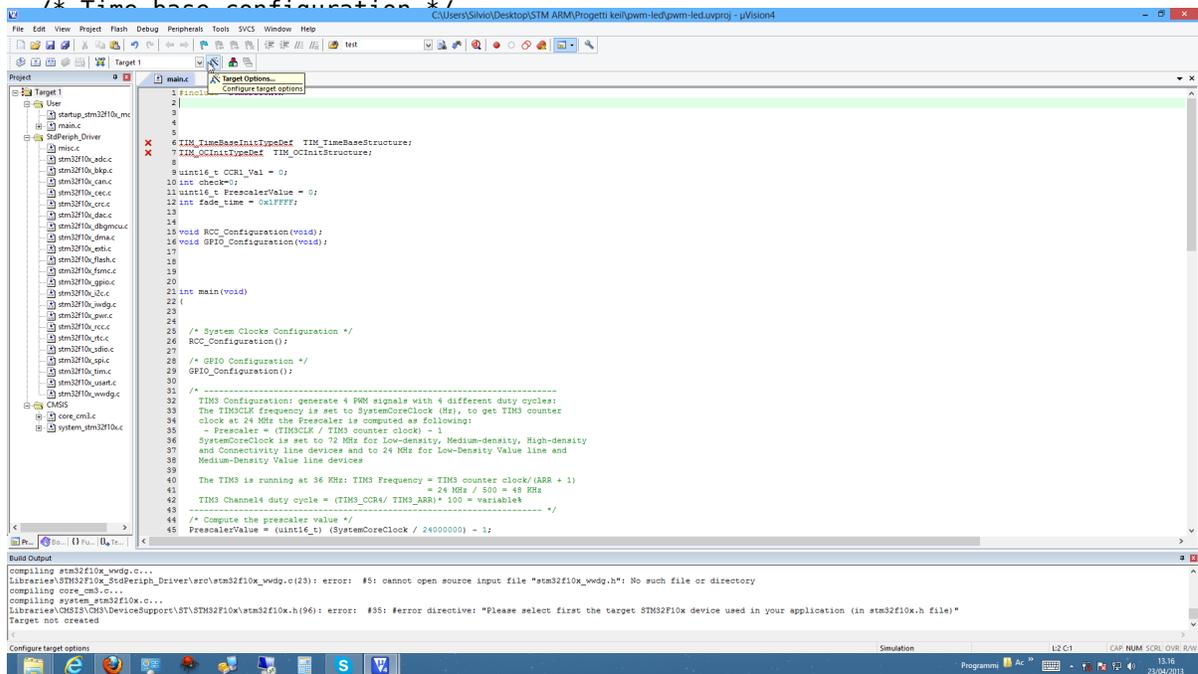
    /* System Clocks Configuration */
    RCC_Configuration();

    /* GPIO Configuration */
    GPIO_Configuration();

    /* -----
Prima di poter compilare però ci sono ancora alcuni passaggi da seguire
TIM3 Configuration: generate 4 PWM signals with 4 different duty cycles:
    The TIM3CLK frequency is set to SystemCoreClock (Hz), to get TIM3 counter
    clock at 24 MHz the Prescaler is computed as following:
        - Prescaler = (TIM3CLK / TIM3 counter clock) - 1
    SystemCoreClock is set to 72 MHz for Low-density, Medium-density, High-density
    and Connectivity line devices and to 24 MHz for Low-Density Value line and
    Medium-Density Value line devices

    The TIM3 is running at 36 KHz: TIM3 Frequency = TIM3 counter clock/(ARR + 1)
                                   = 24 MHz / 500 = 48 KHz
    TIM3 Channel4 duty cycle = (TIM3_CCR4/ TIM3_ARR)* 100 = variable%
    ----- */
    /* Compute the prescaler value */
    PrescalerValue = (uint16_t) (SystemCoreClock / 24000000) - 1;

```

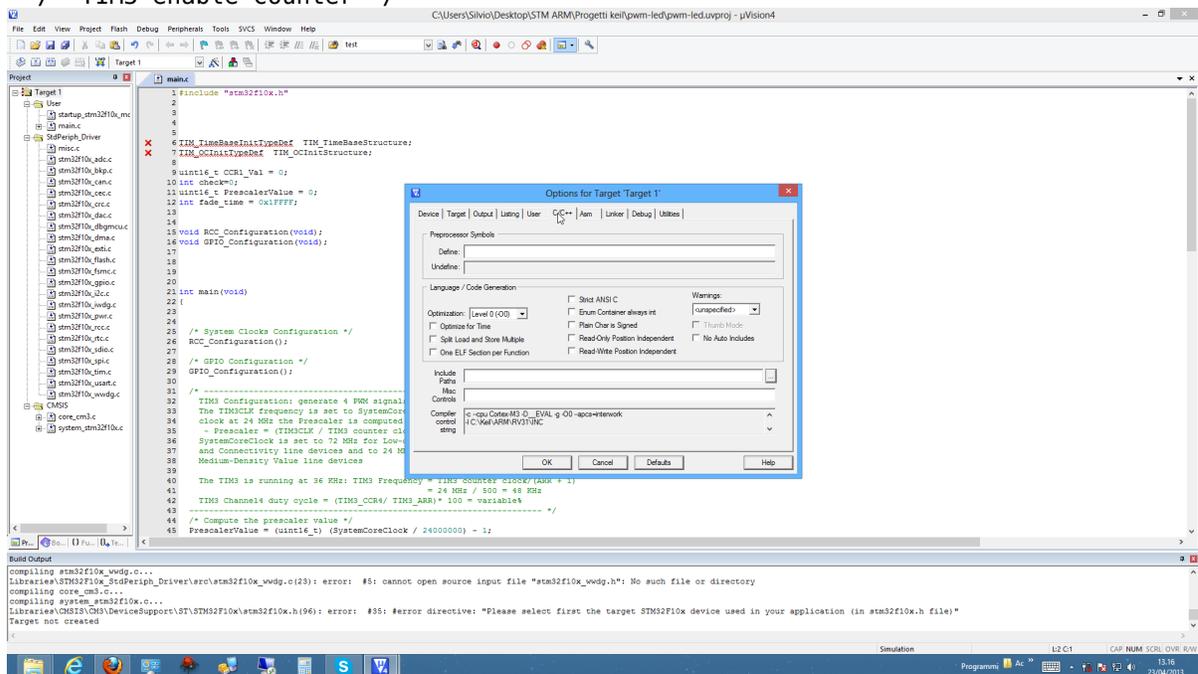


TIM_OC4PreloadConfig(TIM3, TIM_OCPreload_Enable);

TIM_ARRPreloadConfig(TIM3, ENABLE);

andare su **Target Options...**

/* TIM3 enable counter */



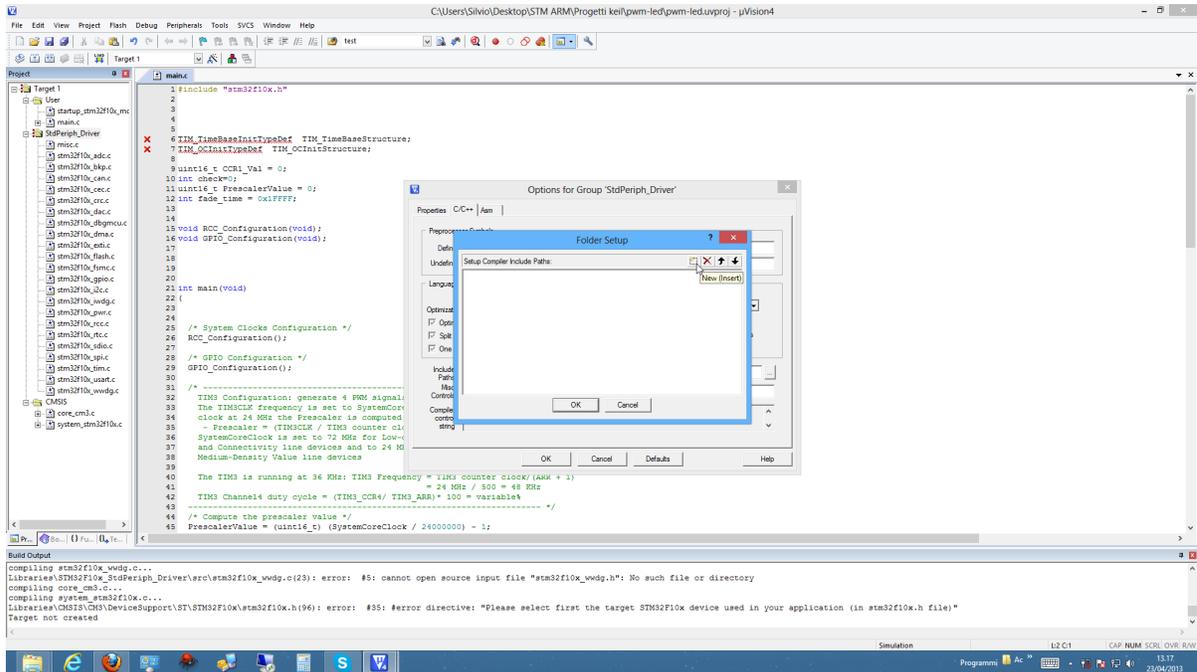
}

}

width='600' height='337'

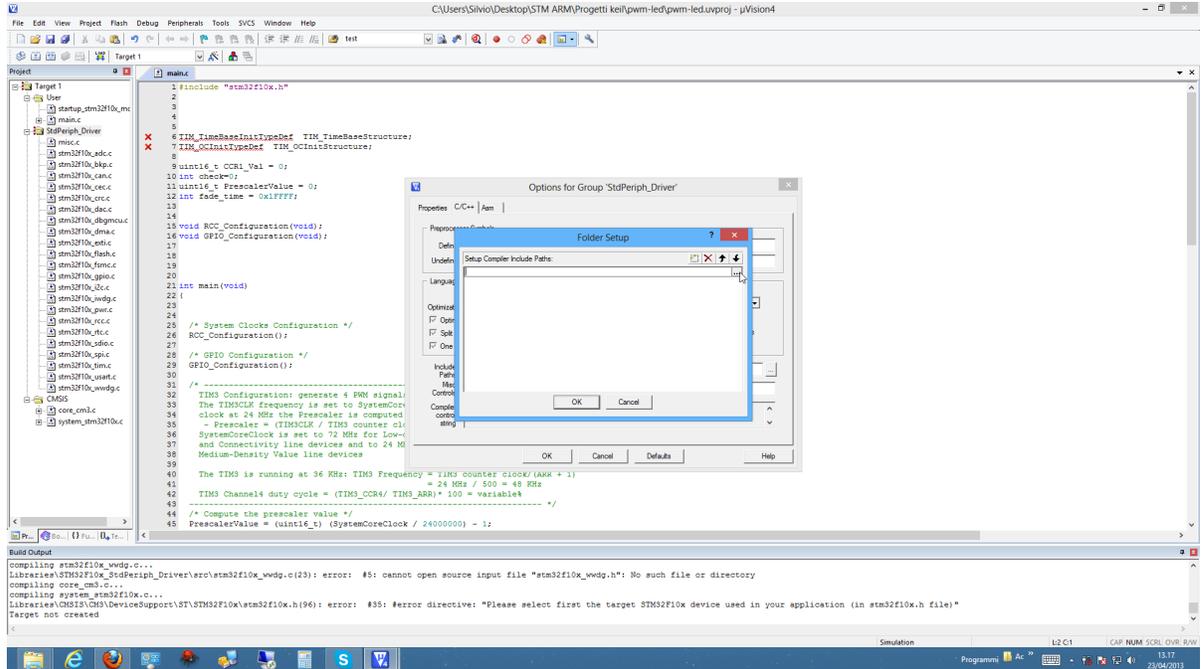
width='600' height='337'

spostarsi nel campo in basso **Include Paths** e premere il tasto sulla destra



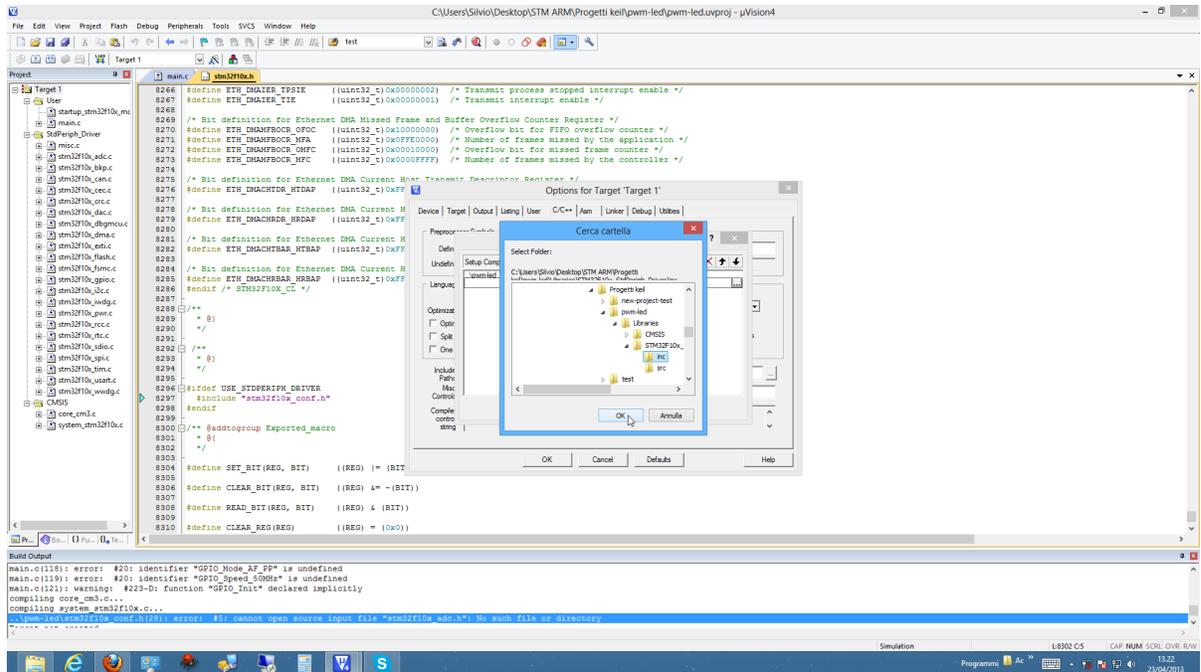
width='600' height='337'

premere **New**



width='600' height='337'

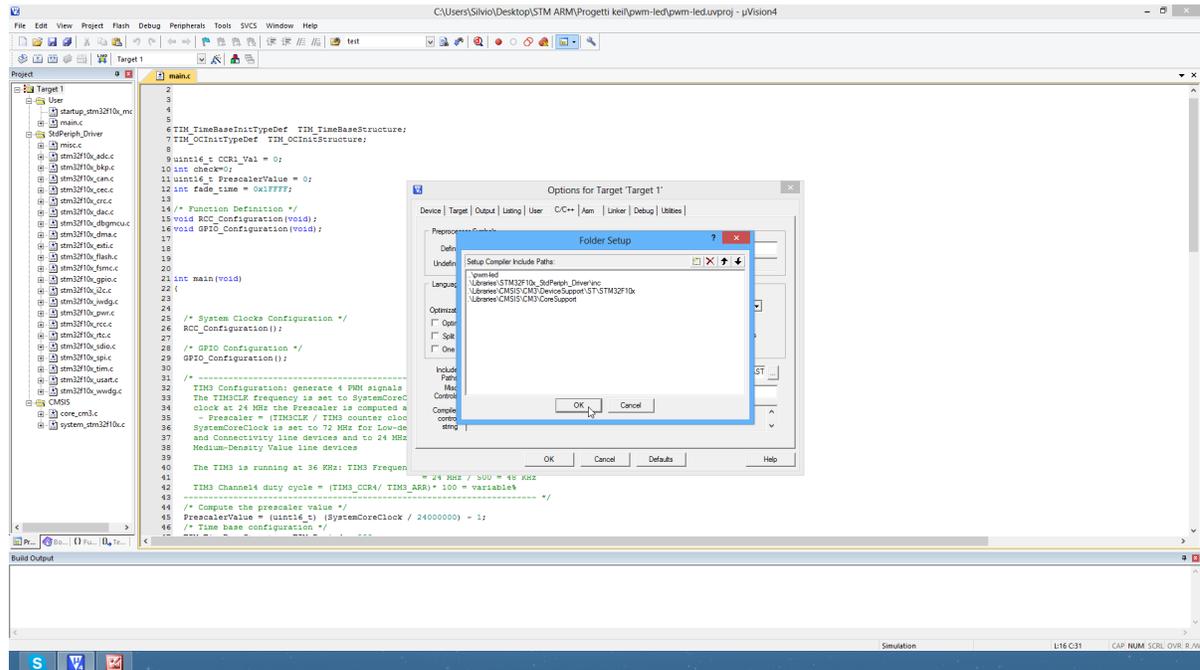
poi il pulsante ...



width='600' height='337'

quindi inserire i seguenti percorsi:

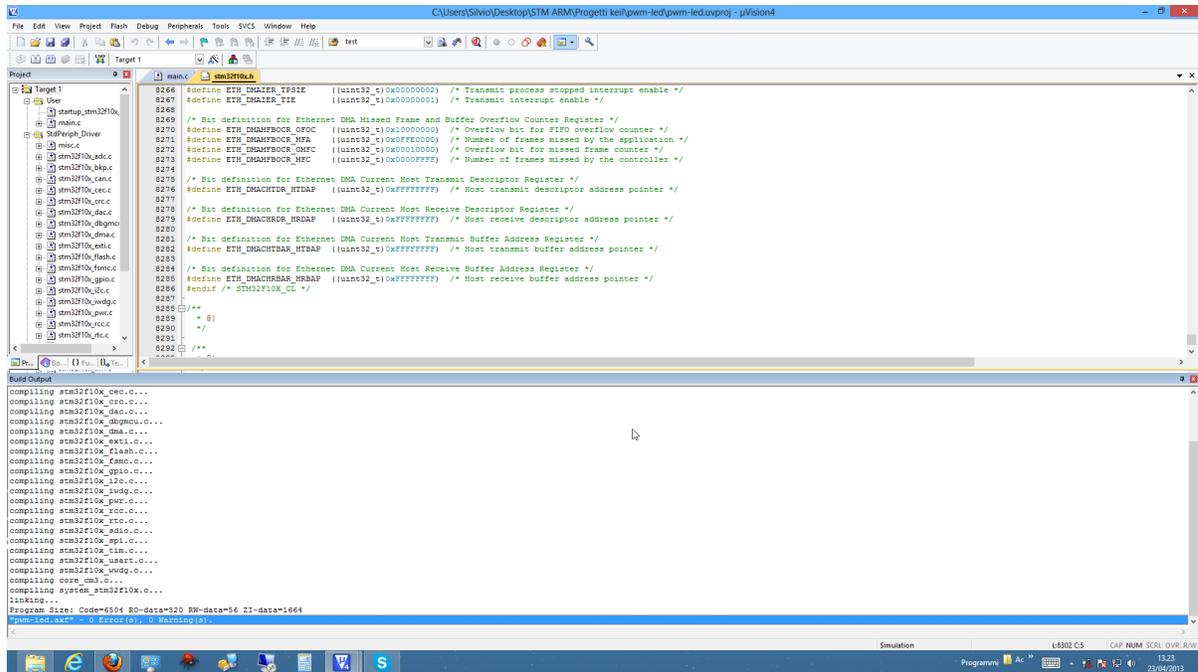
- \pwm-led (perchè la mia cartella principale del progetto è pwm-led)
- \pwm-led\Libraries\STM32F10x_StdPeriph_Driver\inc
- \pwm-led\Libraries\CMSIS\CMS3\DeviceSupport\ST\STM32F10x
- \pwm-led\Libraries\CMSIS\CMS3\CoreSupport



width='600' height='337'

dovrebbe essere come nell'immagine

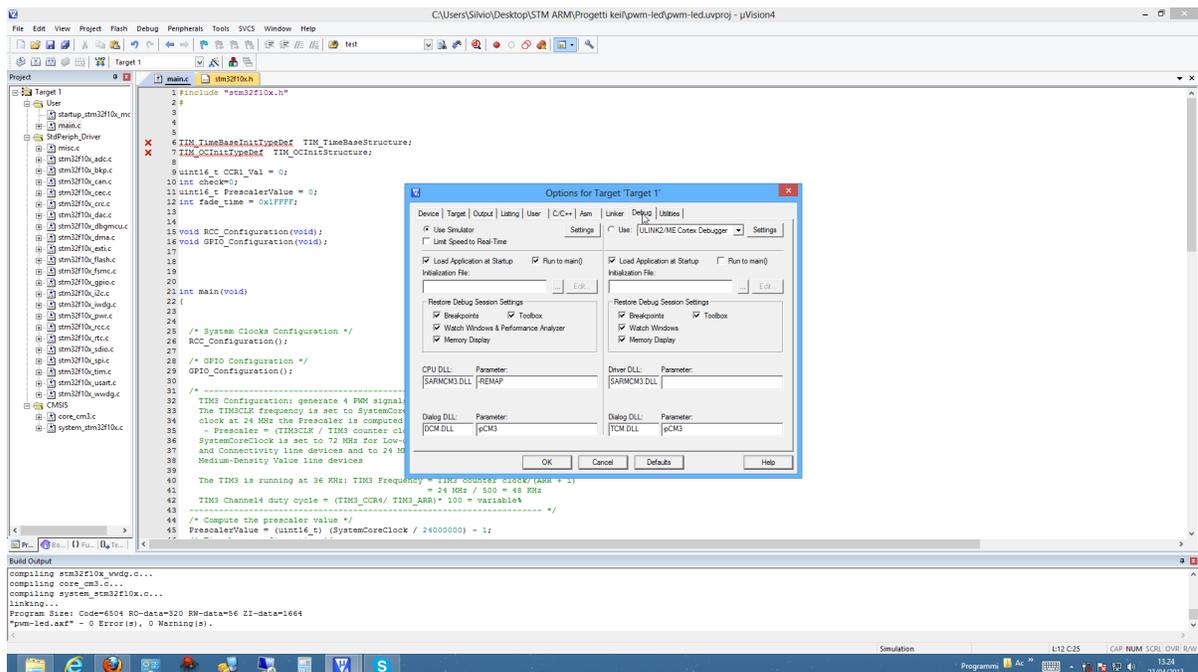
OK ora si può compilare (premendo f7)



width='600' height='337'

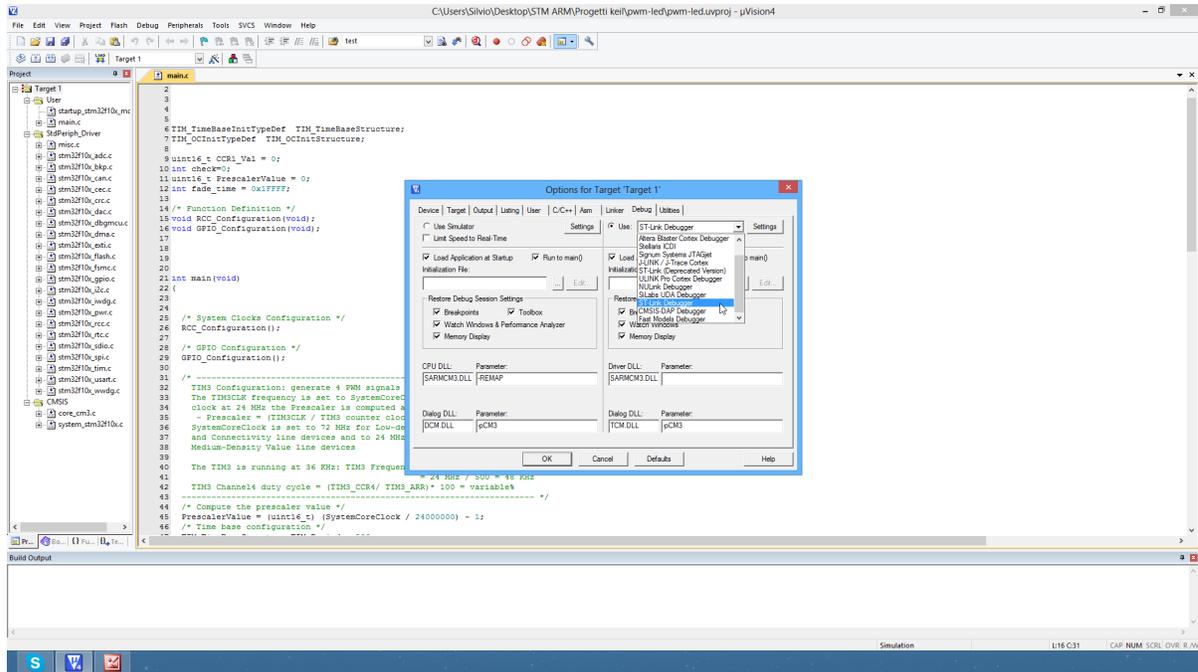
se il risultato è come quello in foto, il procedimento è andato a buon fine

Non rimane che configurare il programmatore/debugger in Keil



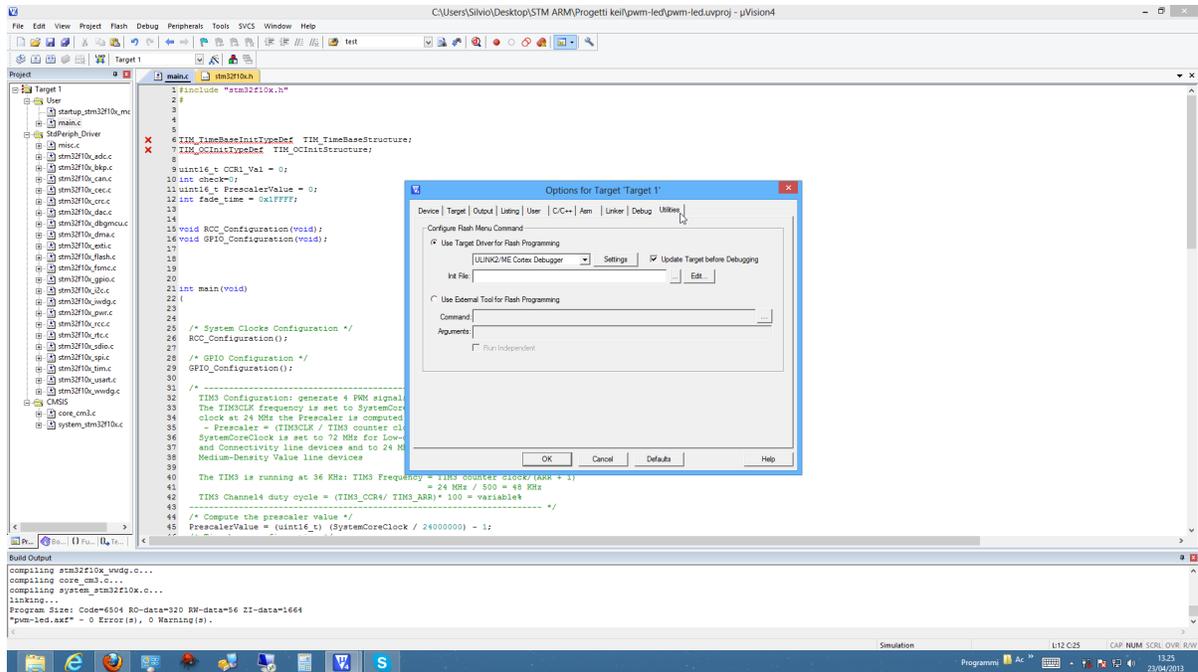
width='600' height='337'

tornare in **Target Options...** e spostarsi su **Debug**



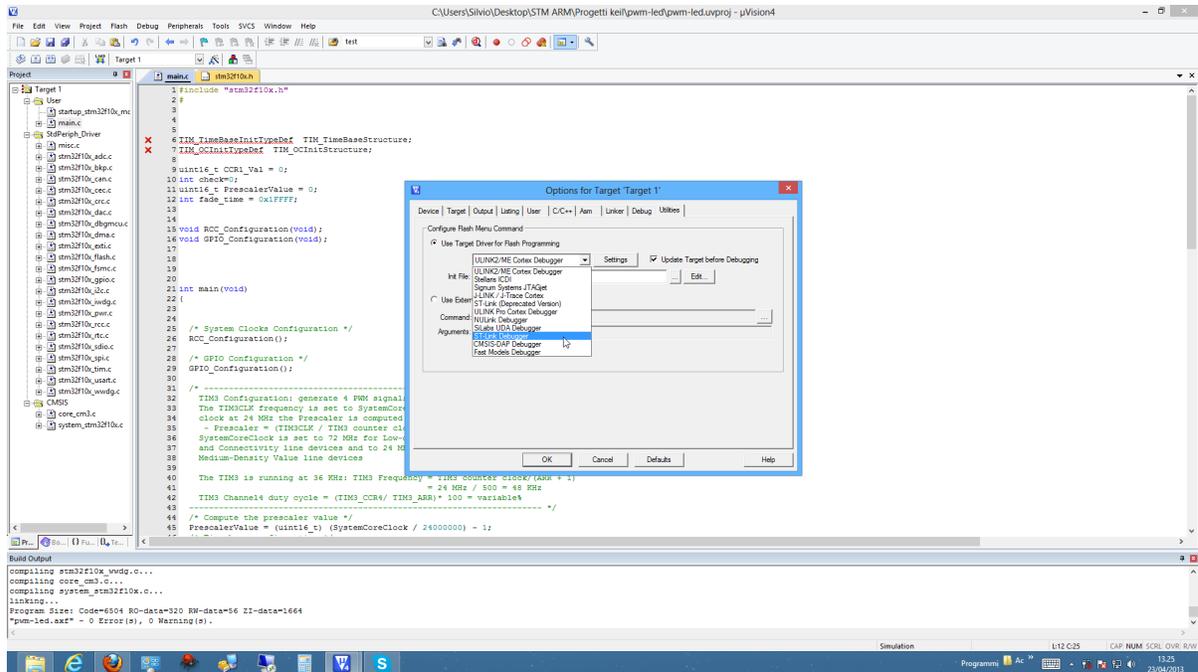
width='600' height='337'

dal menu a tendina selezionare **ST-Link Debugger** e mettere la spunta su **Use**



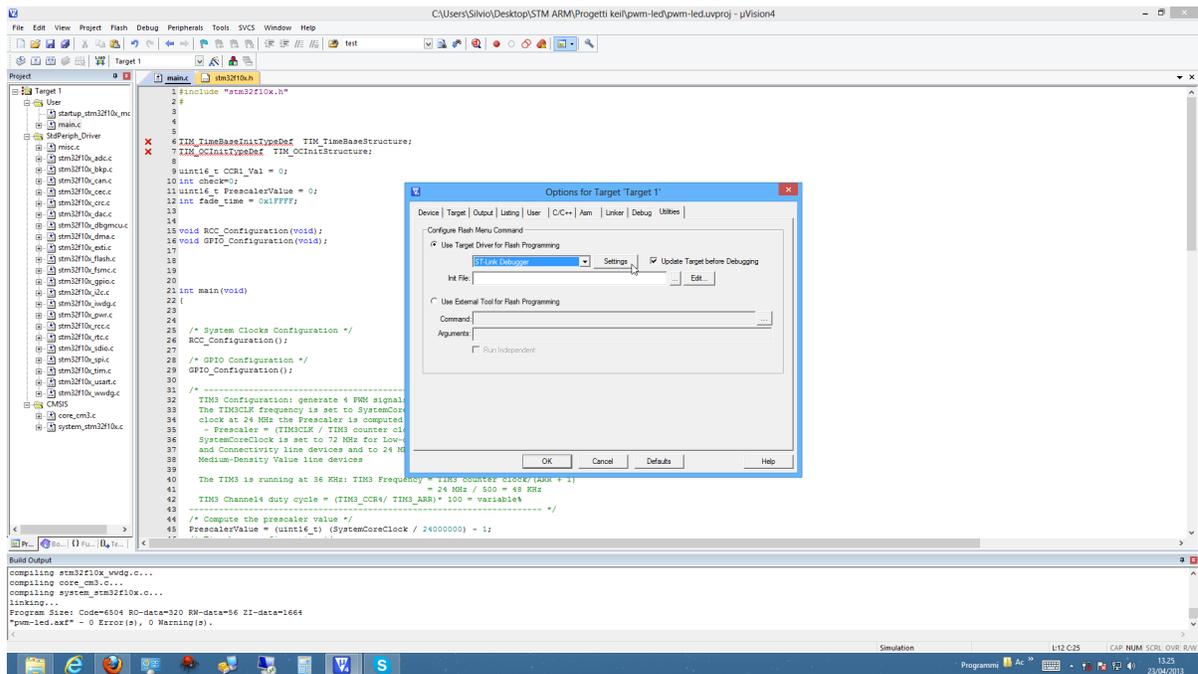
width='600' height='337'

spostarsi su **Utilities**



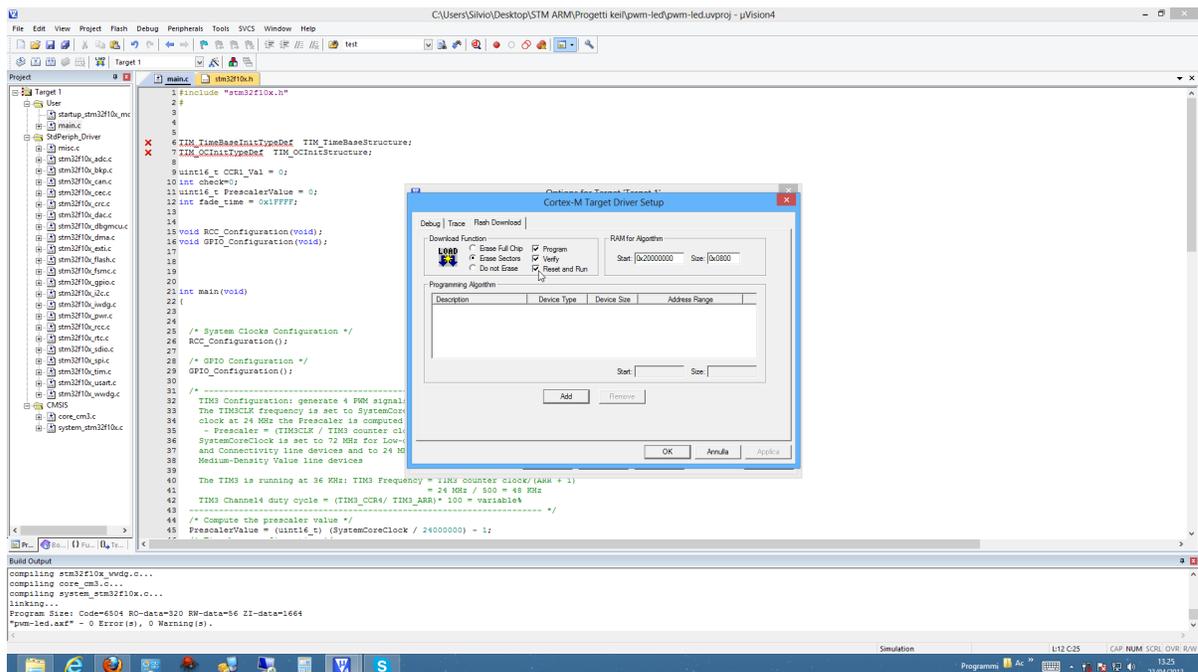
width='600' height='337'

dal menu a tendina scegliere sempre **ST-Link Debugger**



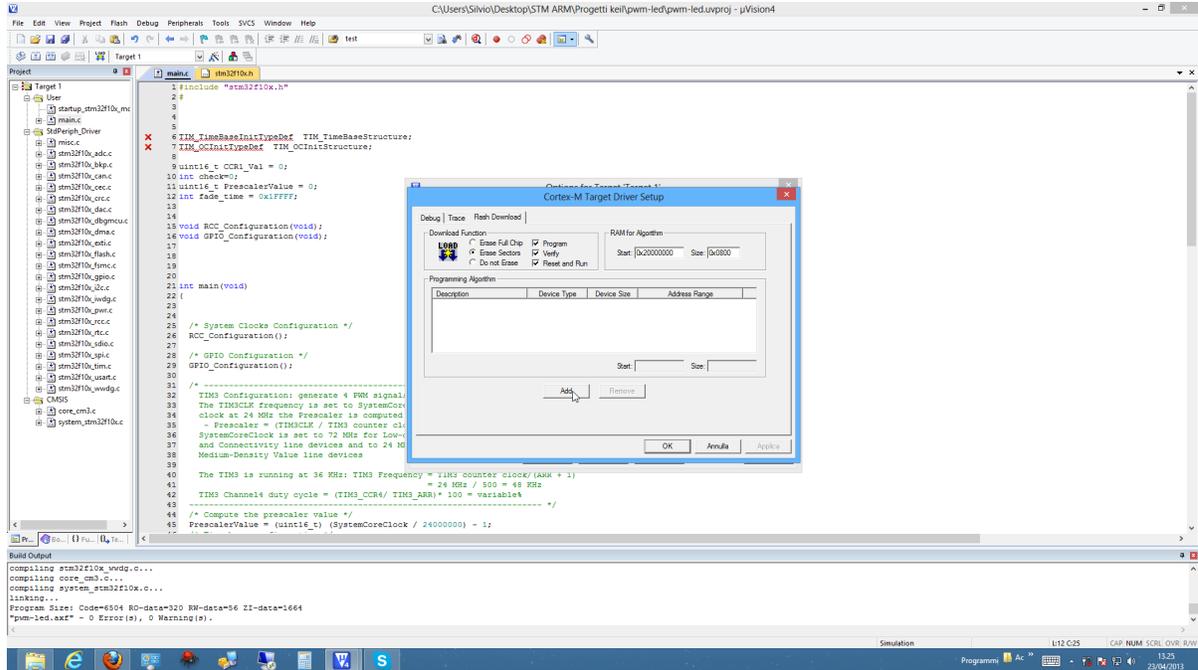
width='600' height='337'

andare su **Settings**



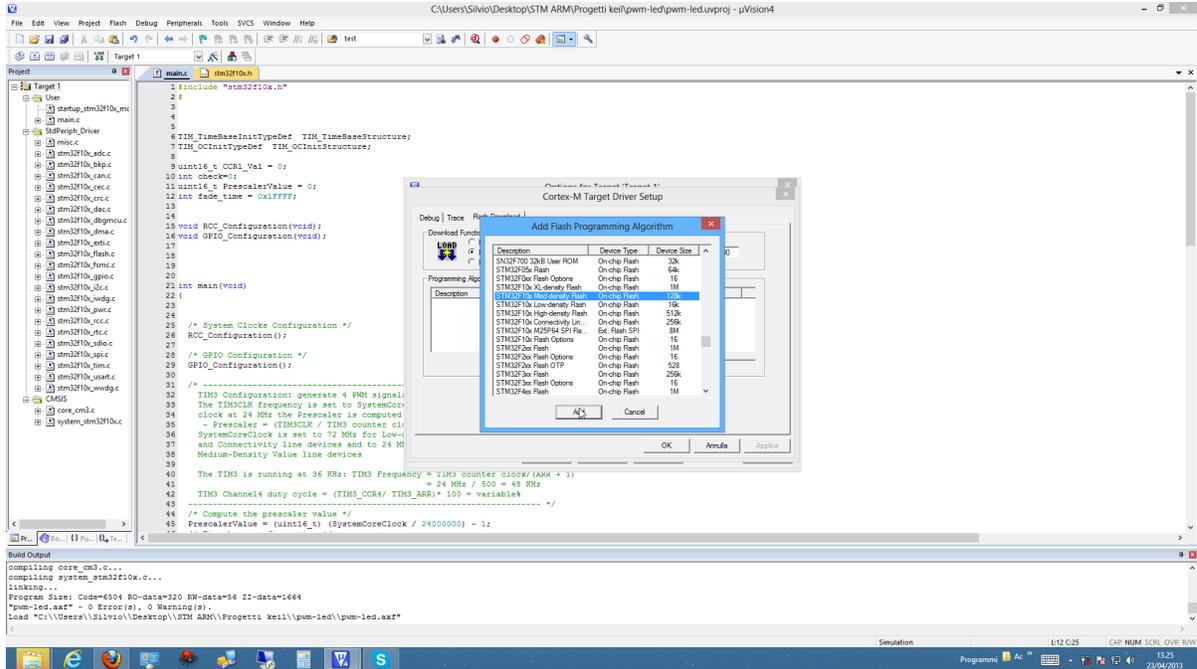
width='600' height='337'

spuntare **Reset and Run** in modo che, una volta scaricato il codice, si avvii senza dover resettare a mano la scheda



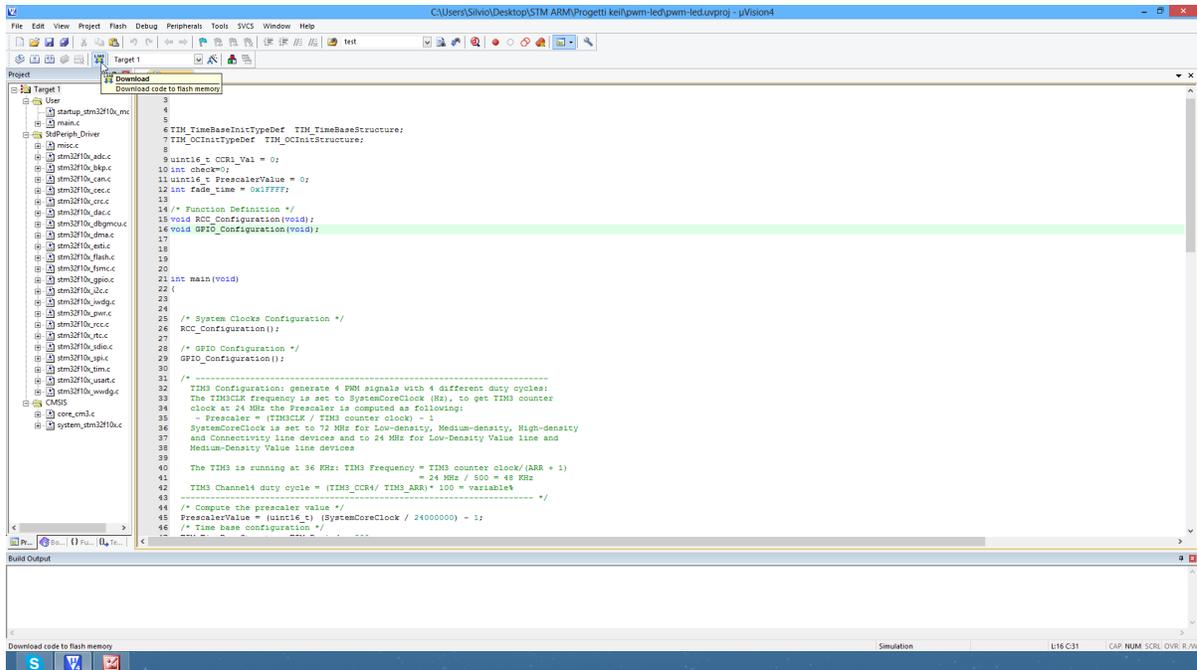
width='600' height='337'

premere **Add**



width='600' height='337'

selezionare **STM32F10x Med-density Flash** ...



width='600' height='337'

Ed il gioco è fatto ora non vi resta che scaricare il codice nella scheda e testare il tutto

Riepilogo

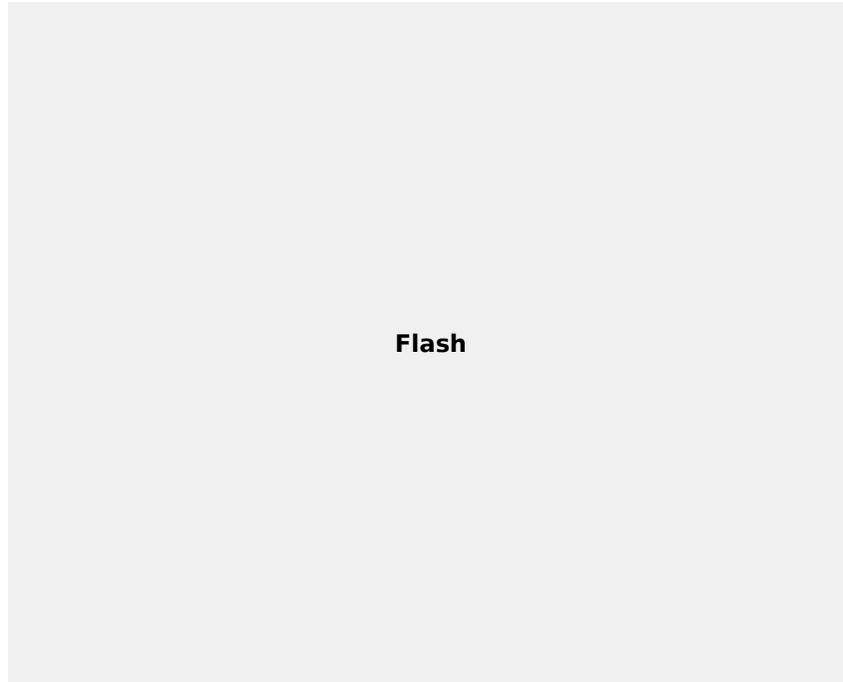
Checklist:

- Libraries + stm32f10x_conf.h
- Add group -> User, StdPeriph_Driver, CMSIS
- User -> Startup + main.c
- StdPeriph_Driver -> tutti i file della cartella src
- CMSIS -> core_cm3.c, system_stm32f10x.c
- Target Options:
 - C/C++:
 - Define -> STM32F10X_MD_VL, USE_STDPERIPH_DRIVER
 - Include Paths :
 - \root (cartella principale del progetto)
 - ... \Libraries\STM32F10x_StdPeriph_Driver\inc
 - ... \Libraries\CMSIS\CM3\DeviceSupport\ST\STM32F10x
 - ... \Libraries\CMSIS\CM3\CoreSupport
 - Debug:
 - ST-Link Debugger
 - Utilities:
 - Target Driver -> ST-Link Debugger
 - Settings:
 - ❖ Reset and run
 - ❖ Add -> STM32F10x Med-density Flash
- Compile & Start

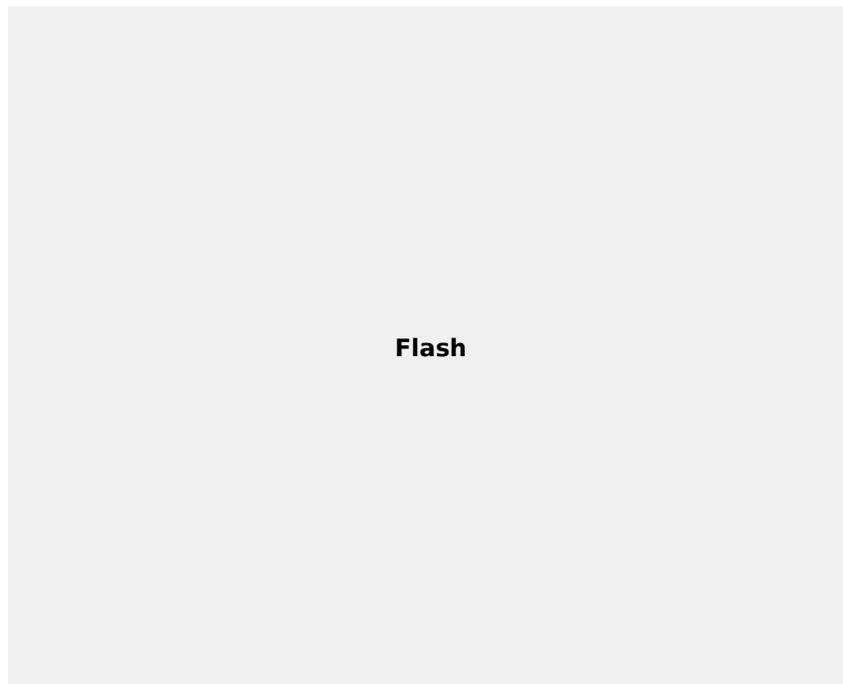
Immagine adf.png

Add-on

Questo è il video di quello che dovrebbe fare la il programma sulla STM32 Discovery



mentre questo è l'analisi all'oscilloscopio della variazione del segnale pwm nel pin PC9



Spero che questo articolo possa essere di aiuto a qualcuno.

Ovviamente sono ben accetti consigli per il miglioramento della guida, e per la correzione di eventuali errori. Ringrazio in generale tutta la community di ElectroYou

per il supporto che offre, ed in particolare a **TardoFreak** che mi ha aiutato a risolvere gli ultimi problemi sulle inclusioni.

Un saluto a tutti

Silvio

Estratto da "<http://www.electroyou.it/mediawiki/index.php?title=UsersPages:Silvio93:guida-iniziare-con-stm32f100rb-e-keil>"