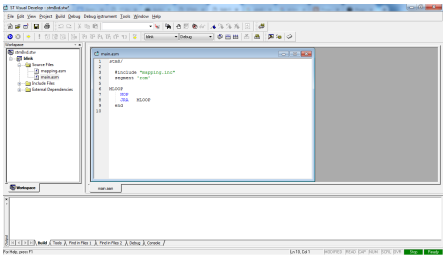
Getting Started with STM8 Assembly Language



Getting Started with STM8 Assembly Language (http://www.fypsolutions.com/tutorials/stm8-assembly-language-hello-world/)

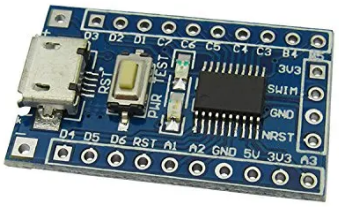
By Abdul Rehman

10 Assembly Language , STM8 , Tutorials ,
Tags: hello-world (http://www.fypsolutions.com/tag/hello-world/), STM8 (http://www.fypsolutions.com/tag/stm8/), stm8 8s003f3p6 (http://www.fypsolutions.com/tag/stm8-8s003f3p6/), stm8 assembly (http://www.fypsolutions.com/tag/stm8-assembly/), stm8 assembly examples (http://www.fypsolutions.com/tag/stm8-assembly-examples/), stm8 assembly language (http://www.fypsolutions.com/tag/stm8-assembly-language/), stm8 basic (http://www.fypsolutions.com/tag/stm8-basic/), stm8s103f3p6 (http://www.fypsolutions.com/tag/stm8s103f3p6/)
26 Dec

Assembly language programming is always a great tool to understand underlying architecture of microcontroller. It also helps us to use any microcontroller at it's full potential. In today's Article we are going to develop a hello world program and we will be using ST Visual develop IDE throughout our tutorial. We are also going to present some very useful links and helping material to better understand stm8 assembly language. So without wasting any further time let's just dive into actual stuff.

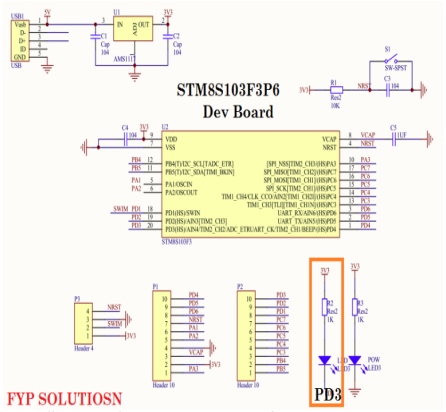
STM8S103F3P Target Board

We are going to target STM8S103F3P microcontroller and will be using stm8s blue pill board for this. You can get this development board via amazon (https://www.amazon.com/Piece-development-board-system-STM8S103F3P6/dp/B077R3TTWW) or any other local store if available. I chooses this board because of its availability in our local store and because of it is very cost effective and cheap. If you find difficulty getting this board you can also purchase this microcontroller directly from ST website or you can choose some similar STM8S microcontroller board.



STM8S103F3P development board

Although you may see USB connector on this board but this is only for powering up this board. This connector has no other functionality rather then providing power to board. There is also one Reset switch to reset microcontroller and one Power LED and one Test LED. Power LED, as it's names suggest is just to indicate power and the TEST led is available for user to program according to their need. According to its schematic this LED is connected to PORTD pin 3 or PD3



(https://i1.wp.com/www.fypsolutions.com/wp-content/uploads/2018/12/STM8S_Board_Schematics.png)
Board Schematic

We will try to target this pin for blinking this on board LED available for us to play with.

Microcontroller Features

According to Datasheet (https://www.st.com/resource/en/datasheet/stm8s103f2.pdf) of this microcontroller, STM8S103F3 has following features

- 16 MHz advanced STM8 core with Harvard architecture and 3-stage pipeline
- Program memory: 8 Kbyte
- Data memory: 640 byte true data EEPROM; endurance 300

Categories
8051-8052 (http://www.fypsolutions.com/category/assembly-language/8051-8052/)
Android (http://www.fypsolutions.com/category/android/)
Arduino (http://www.fypsolutions.com/category/arduino/)
Assembly Language (http://www.fypsolutions.com/category/assembly-language/)
Atmega8 (http://www.fypsolutions.com/category/atmega8/)
C (http://www.fypsolutions.com/category/c-tutorial/)
Computer Vision (http://www.fypsolutions.com/category/computer-vision/)
ESP8266 (http://www.fypsolutions.com/category/esp8266/)
Examples (http://www.fypsolutions.com/category/examples/)
Flutter (http://www.fypsolutions.com/category/android/flutter/)
FPGA (http://www.fypsolutions.com/category/fpga/)
Image Processing (http://www.fypsolutions.com/category/image-processing/)
Java (http://www.fypsolutions.com/category/java/)
Kotlin (http://www.fypsolutions.com/category/android/kotlin/)
MATLAB (http://www.fypsolutions.com/category/matlab/)
Microchip PIC16 (http://www.fypsolutions.com/category/microchip-pic16/)
MySQL (http://www.fypsolutions.com/category/mysql/)
NLP (http://www.fypsolutions.com/category/nlp/)
OpenCV-Python (http://www.fypsolutions.com/category/opencv-python/)
PHP (http://www.fypsolutions.com/category/php/)
PIC-CCS (http://www.fypsolutions.com/category/pic-ccs/)
Python (http://www.fypsolutions.com/category/python/)
Raspberry PI (http://www.fypsolutions.com/category/tutorials/raspberry-pi/)
React-js (http://www.fypsolutions.com/category/react-js/)
RPi 3 (http://www.fypsolutions.com/category/rpi-3/)
STM32 (http://www.fypsolutions.com/category/stm32/)
STM32F103 (http://www.fypsolutions.com/category/stm32f103/)
STM8 (http://www.fypsolutions.com/category/assembly-language/stm8/)
STM8 Cosmic C (http://www.fypsolutions.com/category/stm8-cosmic-c/)
Tutorials (http://www.fypsolutions.com/category/tutorials/)

Up to 27 external interrupts on 6 vectors
16-bit general purpose timer, with 3 CAPCOM channels (IC, OC or PWM)
8-bit basic timer with 8-bit prescaler · Auto wake-up timer
Window watchdog and independent watchdog timers
UART, SmartCard, IrDA, LIN master mode, I2C, SPI
10-bit, ±1 LSB ADC with up to 5 multiplexed channels, scan mode and analog watchdog
96-bit unique key for each device

Flashing the board

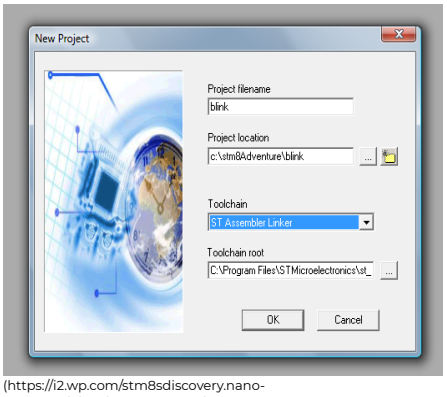
To upload final object file into this microcontroller or to flash this development board we need a ST-LINK Utility (https://www.st.com/en/development-tools/st-link-v2.html) and device. You should not assume that you are going to program this board like an Arduino board via USB cable.

ST-Link V2

If you are unable to figure out how to do this? Than in this blogpost about program stm8 board with st-link v2 (https://blog.junaid.site/2018-01-05-program-stm8s-development-board/), writer had explained it very well.

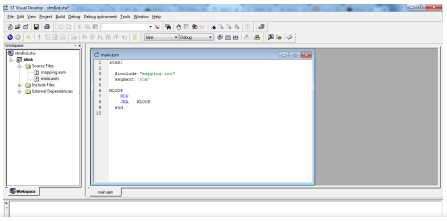
Writing your first Assembly Program

Now its time to write a quick code and see something actually happening. So first of all just download and install STVD for stm8 (https://www.st.com/content/st_com/en/products/development-tools/software-development-tools/stm8-software-development-tools/stm8-programmers/stvd-stm8.html). After installing you will see its shortcut on desktop and also in start menu. So just go and start simply you do with any other application. Now Create a blank workspace and after creating a workspace, create a new empty project. While creating a project it will give option to select the compiler type and in this dialog box you need to select ST Assembler Linker option. After selecting this the editor will automatically add a path you do not need to change that path.



(https://i2.wp.com/stm8sdiscovery.nano-age.co.uk/_/rsrc/1283691553553/adventures-in-stm8-assembler/getting-started-with-the-blinking-led/new-project.png)
Selection of toolchain

In microcontroller selection screen, you need to select STM8S103F3P for our development board. You can choose other device if your board comes with any other STM8 Chip. Once you are done the project window will show three auto generated directories. There is **main.asm** file in **source Files** Directory. You can remove all the lengthy pre-written code and just come up with following simple and clean code.



Popular Tags

- 8051 (HTTP://WWW.FYPSOLUTIONS.COM/TAG/MU8051/)
- ANDROID (HTTP://WWW.FYPSOLUTIONS.COM/TAG/ANDROID/)
- ARDUINO (HTTP://WWW.FYPSOLUTIONS.COM/TAG/ARDUINO/)
- ARDUINO TEMPERATURE METER (HTTP://WWW.FYPSOLUTIONS.COM/TAG/ARDUINO-TEMPERATURE-METER/)
- ARDUINO TEMPERATURE SENSOR PROJECT (HTTP://WWW.FYPSOLUTIONS.COM/TAG/ARDUINO-TEMPERATURE-SENSOR-PROJECT/)
- BASIC (HTTP://WWW.FYPSOLUTIONS.COM/TAG/BASIC/)
- BASIC-CODE (HTTP://WWW.FYPSOLUTIONS.COM/TAG/BASIC-CODE/)
- CIRCUIT (HTTP://WWW.FYPSOLUTIONS.COM/TAG/CIRCUIT/)
- COMPUTER VISION (HTTP://WWW.FYPSOLUTIONS.COM/TAG/COMPUTER-VISION/)
- COSMIC C STM8 (HTTP://WWW.FYPSOLUTIONS.COM/TAG/COSMIC-C-STM8/)
- COSMIC STM8S C COMPILER (HTTP://WWW.FYPSOLUTIONS.COM/TAG/COSMIC-STM8S-C-COMPILER/)
- DIGITAL SIGNAL PROCESSING (HTTP://WWW.FYPSOLUTIONS.COM/TAG/DSP/)
- DS18B20 ESP8266 (HTTP://WWW.FYPSOLUTIONS.COM/TAG/DS18B20-ESP8266/)
- DS18B20 TEMPERATURE SENSOR (HTTP://WWW.FYPSOLUTIONS.COM/TAG/DS18B20-TEMPERATURE-SENSOR/)
- FREQUENCY (HTTP://WWW.FYPSOLUTIONS.COM/TAG/FREQUENCY/)
- GOOGLE SHEETS (HTTP://WWW.FYPSOLUTIONS.COM/TAG/GOOGLE-SHEETS/)
- GPIO (HTTP://WWW.FYPSOLUTIONS.COM/TAG/GPIO/)
- HELLO-WORLD (HTTP://WWW.FYPSOLUTIONS.COM/TAG/HELLO-WORLD/)
- IMAGE PROCESSING (HTTP://WWW.FYPSOLUTIONS.COM/TAG/IMAGE-PROCESSING/)
- IOT (HTTP://WWW.FYPSOLUTIONS.COM/TAG/IOT/)
- JSON (HTTP://WWW.FYPSOLUTIONS.COM/TAG/JSON/)
- KOTLIN (HTTP://WWW.FYPSOLUTIONS.COM/TAG/KOTLIN/)

Basic Program in STVD for STM8

You need to understand one very basic thing about STM8 assembly language

Indentation in Code

indentation here matters a lot. Indentation in STM8 assembly denotes as columns and according to reference manual the first column is denoted as Labels. So if you put any opcode in first column then that will be treated as label if it meet labels criteria or will be prompted as error during compile time.

stm8/ Label at first line

This label at first line tells assembler linker that the targeted platform is of STM8 microcontroller family. Next **segment 'rom'** Line tells to store code in rom. Segment opcode tells where to place it and we are telling assembler to put code at very beginning of the Flash ROM. The **'rom'** directive is defined in **mapping.asm** file like this

```
WORDS           ; The following addresses are 16 bits long
segment byte at 8080-9FFF 'rom'
```


So according to this definition our code will be started at address 0x8080. It's like **ORG** directive in other kind of Assembly programming.

end directive at last line

There must be an END directive at the last line of code and there must be a line break after that directive. This directive tells assembler that the code has been ended and whenever assembler find this directive it will stop compiling further code even if present.

JRA Instruction

This instruction is short form of **JUMP RELATIVE ALWAYS.** This is force jump instruction and tells controller to jump over specific relative address always. Whenever controller find this instruction it will jump to the provided address without any hesitation and without caring anything in code. So according to this definition it expect one operator which is relative address of a location to jump over. We can provide label here as we did in our code. So our final code looks something like this so far

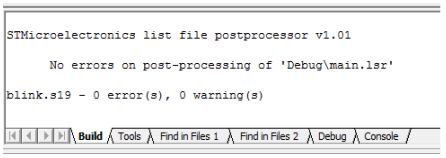
```
stm8/

    #include "mapping.inc"
    segment 'rom'

MLOOP
        NOP
        JRA     MLOOP
    end
```


Now its time to compile our code and see if we are able to compile without facing and error or warnings. To compile our code just press **F7 key** or go to build menu and selection **Build** option. If everything goes well this will show success message something like this

MINI PROJECTS (HTTP://WWW.FYPSOLUTIONS.COM/TAG/MINI-PROJECTS/)
NLP (HTTP://WWW.FYPSOLUTIONS.COM/TAG/NLP/)
NODEMCU ESP8266 GOOGLE SHEETS (HTTP://WWW.FYPSOLUTIONS.COM/TAG/NODEMCU-ESP8266-GOOGLE-SHEETS/)
PIC (HTTP://WWW.FYPSOLUTIONS.COM/TAG/MICROCHIP-PIC/)
PYTHON (HTTP://WWW.FYPSOLUTIONS.COM/TAG/PYTHON/)
RASPBERRY PI (HTTP://WWW.FYPSOLUTIONS.COM/TAG/RASPBERRY-PI/)
STM8 (HTTP://WWW.FYPSOLUTIONS.COM/TAG/STM8/)
STM8 8S003F3P6 (HTTP://WWW.FYPSOLUTIONS.COM/TAG/STM8-8S003F3P6/)
STM8 ASSEMBLY (HTTP://WWW.FYPSOLUTIONS.COM/TAG/STM8-ASSEMBLY/)
STM8 ASSEMBLY EXAMPLES (HTTP://WWW.FYPSOLUTIONS.COM/TAG/STM8-ASSEMBLY-EXAMPLES/)
STM8 ASSEMBLY LANGUAGE (HTTP://WWW.FYPSOLUTIONS.COM/TAG/STM8-ASSEMBLY-LANGUAGE/)
STM8 BASIC (HTTP://WWW.FYPSOLUTIONS.COM/TAG/STM8-BASIC/)
STM8 BLUE PILL (HTTP://WWW.FYPSOLUTIONS.COM/TAG/STM8-BLUE-PILL/)
STM8 EXAMPLE CODE (HTTP://WWW.FYPSOLUTIONS.COM/TAG/STM8-EXAMPLE-CODE/)
STM8 PROGRAMMING (HTTP://WWW.FYPSOLUTIONS.COM/TAG/STM8-PROGRAMMING/)
STM8S (HTTP://WWW.FYPSOLUTIONS.COM/TAG/STM8S/)
STM8S003F3P6 (HTTP://WWW.FYPSOLUTIONS.COM/TAG/STM8S003F3P6/)
STM8S103F3P6 (HTTP://WWW.FYPSOLUTIONS.COM/TAG/STM8S103F3P6/)
STM8S DISCOVERY TUTORIAL (HTTP://WWW.FYPSOLUTIONS.COM/TAG/STM8S-DISCOVERY-TUTORIAL/)
STM32 (HTTP://WWW.FYPSOLUTIONS.COM/TAG/STM32/)
WEB BASED (HTTP://WWW.FYPSOLUTIONS.COM/TAG/WEB-BASED/)



Compiled without any error

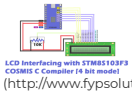
Closing statement

So that's it for today's tutorial. If you followed along this tutorial and end up with a successful compile message, Congratulations than, You had properly setup your development environment and written your first hello world program. This program will do one very important work for your microcontroller which is to keep it busy and stuck it to one loop called **infinite loop** in Embedded Systems. In our next tutorial we will blink LED and see things happening in our Development board.

Useful Resources

1. Device Datasheet (https://www.st.com/resource/en/datasheet/stm8s103f2.pdf)
2. ST Assembler Linker User Manual (https://www.st.com/content/ccc/resource/technical/document/user_manual/b8/61/d0/68/83/82/4f/2b/CD00056470.pdf/files/CD00056470.pdf/jcr:content/transla
3. STM8 Programming Manual (https://www.st.com/content/ccc/resource/technical/document/programming_manual/43/24/13/9a/89/df/45/ed/CD00161709.pdf/files/CD00161709.pdf/jcr:content/
4. ST-Link Utility (https://www.st.com/en/development-tools/st-link-v2.html)
5. ST-Link User manual (https://www.st.com/content/ccc/resource/technical/document/user_manual/65/e0/44/72/9e/34/41/8d/DM00026748.pdf/files/DM00026748.pdf/jcr:content/transl
6. ST Visual Development IDE (https://www.st.com/content/st_com/en/products/development-tools/software-development-tools/stm8-software-development-tools/stm8-programmers/stvd-stm8.html) (https://www.st.com/content/ccc/resource/technical/document/user_manual/b8/61/d0/68/83/82/4f/2b/CD00056470.pdf/files/CD00056470.pdf/jcr:content/transla
7. Another LED Blink Example (http://stm8sdiscovery.nano-age.co.uk/adventures-in-stm8-assembler/getting-started-with-the-blinking-led)
8. (https://www.st.com/content/ccc/resource/technical/document/programming_manual/43/24/13/9a/89/df/45/ed/CD00161709.pdf/files/CD00161709.pdf/jcr:content/ STM8S Board with ST-Link V2 (https://blog.junaid.site/2018-01-05-program-stm8s-development-board/)
9. Programing STM8 with stlink (http://digital-wizard.net/stm8_tutorials/programming_stm8_using_swim?fbclid=IwAR2ZVULXKsqJLt8jH-2WO5OGFPrfj_97gTjDN934kdv4oN3PNsG5WK7V9TY)

Related



(http://www.fypsolutions.com/examples/lcd-4-bit-mode-interfacing-with-stm8-in-c-from-scratch/) LCD 4 bit mode interfacing with STM8 in C from scratch (http://www.fypsolutions.com/examples/lcd-4-bit-mode-interfacing-with-stm8-in-c-from-scratch/)

(http://www.fypsolutions.com/tutorials/stm8-assembly-language-hello-world/registers-of-stm8-for-assembly-programming/) While programming STM8 microcontroller using assembly

(http://www.fypsolutions.com/examples/stm8s-external-interrupt-example/) STM8s External interrupt example (http://www.fypsolutions.com/examples/stm8s-external-interrupt-example/)

Leave a Reply

Your email address will not be published. Required fields are marked *

Enter comment Message...

Full Name | Email | Website

Notify me of follow-up comments by email.

Notify me of new posts by email.

POST COMMENT

This site uses Akismet to reduce spam. Learn how your comment data is processed (https://akismet.com/privacy/).

FYP Solutions Powered By Impressive Business WordPress Theme (https://voilathemes.com/wordpress-themes/impressive-business/)