

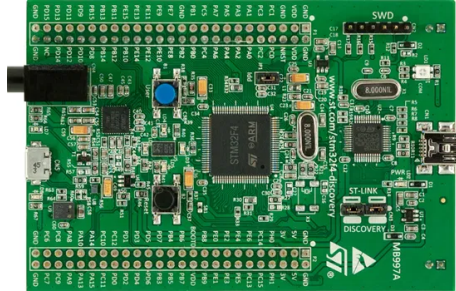
Iztok's HW/SW Sandbox

... with simple and easy to follow recipes!

STM32F4 – Hello World!

Posted on [July 1, 2014](#)

In this tutorial I describe how to *build* and *deploy* a simple “Hello World!” application on the popular ARM-based STM32F4 Discovery board using open source tools (sneak peak, [here](#)).



This tutorial is suitable for total beginners! First, you will *install* the development environment with no hassle in few minutes via apt-get command. I use public and well-maintained repositories whenever possible. Second, in just 3 commands you will *retrieve* the project from the public repository, *build* it and *deploy* the executable on the STM32F4 Discovery board. Finally, you may *explore* the references to learn technologies in greater depth, try to *play* with the existing code or *read* another STM32F4 post.

Enjoy your coding! 😊

1. Prerequisites

- [Ubuntu 14.04 LTS](#) (x86 architecture).
- [STM32F4 Discovery Board](#) (ARM architecture, costs less than 20 EUR).

2. Install Software Dependencies

- Install GNU ARM Embedded Toolchain. You’ll need this cross-toolchain to build executables for the ARM target on the x86 host.

```
sudo apt-get install build-essential git \
    binutils-arm-none-eabi \
    gcc-arm-none-eabi \
    libnewlib-arm-none-eabi
```

- Install [OpenOCD](#) package to enable STM32F4 programming & remote debugging.

```
sudo apt-get install openocd
```

- Download STM32F4 examples.
 1. Clone [my git repository](#). I recommend this options.

```
git clone https://github.com/istarc/stm32.git
```

Privacy & Cookies: This site uses cookies. By continuing to use this website, you agree to their use.
To find out more, including how to control cookies, see here: [Cookie Policy](#)

Close and accept

```
# Download & Extract
mkdir -p ~/stm32f4 && cd ~/stm32f4

# Download stsw-stm32068 file
wget http://bit.ly/1ryC8Tp
unzip stsw-stm32068.zip

cd STM32F4-Discovery_FW_V1.1.0/Project
cd Peripheral_Examples/IO_Toggle
```

3. Build the IO Toggle Project

Now, it's perfect time to get familiar with [GNU make](#). It's a great utility to automate a building process. Create a new file in the IO Toggle directory and name it "Makefile". Copy the code below and paste it inside the Makefile. *Skip this step if you cloned the git repository.*

```
1 # GNU ARM Embedded Toolchain
2 CC=arm-none-eabi-gcc
3 LD=arm-none-eabi-ld
4 AR=arm-none-eabi-ar
5 AS=arm-none-eabi-as
6 CP=arm-none-eabi-objcopy
7 OD=arm-none-eabi-objdump
8
9 # Build Parameters: MCU Flags, Definitions, Includes,
10 # Compile Flags, Linker Script, Linker Flags
11 MCFLAGS=-mcpu=cortex-m4 -mthumb -mlittle-endian \
12 -mfpu=fpv4-sp-d16 -mfloat-abi=hard -mthumb-interwork
13 DEFS=-DUSE_STDPERIPH_DRIVER -DSTM32F4XX
14 INCLUDES=-I. \
15 -I../../Libraries/CMSIS/ST/STM32F4xx/Include \
16 -I../../Utilities/STM32F4-Discovery \
17 -I../../Libraries/CMSIS/Include \
18 -I../../Libraries/STM32F4xx_StdPeriph_Driver/inc
19 CFLAGS=-c $(MCFLAGS) $(DEFS) $(INCLUDES)
20 LDSCRIPT = TrueSTUDIO/IO_Toggle/stm32_flash.ld
21 LDFLAGS=-T $(LDSCRIPT) -specs=nosys.specs $(MCFLAGS)
22
23 # Inputs: C Sources, Assembler Sources
24 SOURCES=main.c stm32f4xx_it.c system_stm32f4xx.c \
25 ../../Utilities/STM32F4-Discovery/stm32f4_discovery.c \
26 ../../Libraries/STM32F4xx_StdPeriph_Driver/src/stm32f4xx_syscfg.c \
27 ../../Libraries/STM32F4xx_StdPeriph_Driver/src/misc.c \
28 ../../Libraries/STM32F4xx_StdPeriph_Driver/src/stm32f4xx_gpio.c \
29 ../../Libraries/STM32F4xx_StdPeriph_Driver/src/stm32f4xx_rcc.c \
30 ../../Libraries/STM32F4xx_StdPeriph_Driver/src/stm32f4xx_exti.c
31 ASMSOURCES=../../Libraries/CMSIS/ST/STM32F4xx/Source/Templates/gcc_ride7/st
32
33 # Outputs: Object Files, ELF Executable &
34 # Converted ELF Executable to Intel HEX format
35 OBJECTS=$(SOURCES:%.c=%.o)
36 OBJECTS+=$(ASMSOURCES:%.s=%.o)
37 EXECUTABLE=iotoggle.elf
38 TARGET=iotoggle.hex
39
40 # Build Rules
41 .PHONY: release
42 release: $(TARGET)
43
44 .PHONY: all
45 all: $(TARGET)
46
47 .PHONY: debug
```

Privacy & Cookies: This site uses cookies. By continuing to use this website, you agree to their use.
To find out more, including how to control cookies, see here: [Cookie Policy](#)

Close and accept

```

53     @echo "Objcopy from ELF to IHEX complete!\n"
54
55 $(EXECUTABLE): $(OBJECTS)
56     $(CC) $(LDFLAGS) $(OBJECTS) -o $@
57     @echo "Linking complete!\n"
58
59 %.o: %.c
60     $(CC) $(CFLAGS) $< -o $@
61     @echo "Compiled "$<"\n"
62
63 %.o: %.s
64     $(CC) $(CFLAGS) $< -o $@
65     @echo "Assambled "$<"\n"
66
67 .PHONY: clean
68 clean:
69     rm -f $(OBJECTS) $(EXECUTABLE) $(TARGET)

```

Now, you'll use this Makefile to automagically build the project (you could also do it manually, but it's a tedious work!).

```
make clean && make
```

4. Deploy the IO Toggle Project

Deploy the iotoggle executable using OpenOCD (don't forget to plug-in the STM32F4 board via USB). The OpenOCD driver will automatically detect the appropriate USB port.

```

sudo openocd \
-f /usr/share/openocd/scripts/board/stm32f4discovery.cfg \
-c "program iotoggle.elf verify reset"

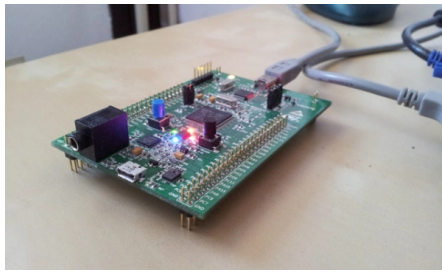
```

If everything goes well, you should see the output text below and blinking LEDs on the STM32F4 board.

```

...
Info : STLINK v2 JTAG v14 API v2 SWIM v0 VID 0x0483 PID 0x3748
Info : Target voltage: 2.900177
Info : stm32f4x.cpu: hardware has 6 breakpoints, 4 watchpoints
target state: halted
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x08002060 msp: 0x20020000
** Programming Started **
auto erase enabled
Info : device id = 0x10016413
Info : flash size = 1024kbytes
wrote 16384 bytes from file iotoggle.elf in 0.844525s (18.946 KiB/s)
** Programming Finished **
** Verify Started **
verified 10384 bytes in 0.116926s (86.727 KiB/s)
** Verified OK **
** Resetting Target **

```



Whoohoo! Now it's time to celebrate!

If you want to change the LED blinking order or delay, you can make appropriate changes in the main.c file. Then rebuild and deploy the project (use section 3 and 4, respectively).

Many things can go wrong in software development. One of them are software bugs. Once they are discovered, they can be effectively mitigated by using STM32F4 in-circuit debugger, but that's another story (click [here](#)).

Q: Deployment fails repeatedly due to some OpenOCD issue. 😞 Is there a workaround?

A: Yes, the current official Ubuntu package (Aug 2014) contains a prehistoric OpenOCD version. You should build a newer version from scratch. I provide step-by-step instructions here (2nd section, it takes less than 3 minutes to build it). When you are done, just return here and continue as nothing happened. It will work out of the box. 😊

6. References

This tutorial is based on Benjamin's robotics blog on "[Get started with the STM32F4 on Ubuntu Linux](#)". The Makefile is based on Nabil's blog on "[Programming the STM32F4 DISCOVERY with the Bus Blaster](#)". The Homer Simpson Whoohoo cover is from [FB Cover Street](#).

Share this:



Like

Be the first to like this.

Related

[STM32F4 – In-circuit Debugging \[Revisited\]](#)
In "Embedded Systems"

[STM32F4 - Deploy FreeRTOS Embedded OS in under 10 seconds!](#)
In "Embedded Systems"

[STM32F4 - Unit Testing with CppUTest or GoogleTest \[Part I.\]](#)
In "Embedded Systems"



About istarc

Embedded Systems Developer.

[View all posts by istarc →](#)

This entry was posted in [Embedded Systems](#), [STM32F4](#) and tagged [ARM](#), [Embedded Systems](#), [Open Source](#), [Software Development](#). Bookmark the [permalink](#).

Privacy & Cookies: This site uses cookies. By continuing to use this website, you agree to their use.

To find out more, including how to control cookies, see here: [Cookie Policy](#)

Close and accept

**su** says:

March 19, 2015 at 3:12 pm

Hey when i execute the command- `sudo openocd \`
`-f /usr/share/openocd/scripts/board/stm32f4discovery.cfg \`
`-c "program iotoggle.elf verify reset"`

i get the following output and i am not able to proceed.

discovery.cfg \-c "program iotoggle.elf verify reset"

Open On-Chip Debugger 0.7.0 (2013-10-22-08:31)

Licensed under GNU GPL v2

For bug reports, read

<http://openocd.sourceforge.net/doc/doxygen/bugs.html>

Runtime Error: embedded:startup.tcl:47: Can't find /usr/share/ocencd/scripts/board/stm32f4discovery.cfg
in procedure 'script'

at file "embedded:startup.tcl", line 47

Please let me know what i should do to move ahead.

thanks for the previous reply and in advance thanks for next reply.

[Reply](#)

**istarc** says:

March 22, 2015 at 10:13 pm

Hi, try the following workaround.

1. "sudo updatedb"
2. "locate stm32f4discovery.cfg"
3. Provide appropriate path to stm32f4discovery.cfg to -f switch.

[Reply](#)

**su** says:

March 26, 2015 at 3:26 pm

ok i will try these. thanks

**stappers** says:

May 18, 2016 at 1:02 pm

the backslash means "continuing line"

it is used for getting one logic line on several physical lines.

Example

```
cat \  
/etc/hosts
```

is actual

```
cat /etc/hosts
```

The trick is that the backslash the last character on a physical line.

HTH

Groeten

**stappers** says:

May 18, 2016 at 1:14 pm

oh, by the way, it is "openocd", not "opencd".

So add an extra o in

Runtime Error: embedded:startup.tcl:47: Can't find /usr/share/opencd/scripts/board/stm32f4discovery.cfg

[Reply](#)**bay** says:

May 30, 2016 at 9:28 am

Open On-Chip Debugger 0.7.0 (2013-10-22-08:31)

Licensed under GNU GPL v2

For bug reports, read

<http://openocd.sourceforge.net/doc/doxygen/bugs.html>

srst_only separate srst_nogate srst_open_drain connect_deassert_srst

Info : This adapter doesn't support configurable speed

Info : STLINK v2 JTAG v14 API v2 SWIM v0 VID 0x0483 PID 0x3748

Info : Target voltage: 2.884248

Info : stm32f4x.cpu: hardware has 6 breakpoints, 4 watchpoints

target state: halted

target halted due to debug-request, current mode: Thread

xPSR: 0x01000000 pc: 0x08000228 msp: 0x20000690

** Programming Started **

auto erase enabled

Error: couldn't open iotoggle.elf

** Programming Failed **

Help!

[Reply](#)**istarc** says:

June 30, 2017 at 7:22 am

Hi, could you please try with sudo. I guess openocd doesn't have sufficient privileges to access the device. Br, Iztok

[Reply](#)**bay** says:

May 30, 2016 at 9:30 am

Open On-Chip Debugger 0.7.0 (2013-10-22-08:31)

Licensed under GNU GPL v2

For bug reports, read

<http://openocd.sourceforge.net/doc/doxygen/bugs.html>

srst_only separate srst_nogate srst_open_drain connect_deassert_srst

Info : This adapter doesn't support configurable speed

Info : STLINK v2 JTAG v14 API v2 SWIM v0 VID 0x0483 PID 0x3748

Info : Target voltage: 2.884248

Info : stm32f4x.cpu: hardware has 6 breakpoints, 4 watchpoints

target state: halted

target halted due to debug-request, current mode: Thread

xPSR: 0x01000000 pc: 0x08000228 msp: 0x20000690

Privacy & Cookies: This site uses cookies. By continuing to use this website, you agree to their use.
To find out more, including how to control cookies, see here: [Cookie Policy](#)

[Close and accept](#)

[Reply](#)**davngr** says:

October 3, 2016 at 2:41 am

Hi, Thanks for the tutorial,

After executing the last command "sudo openocd \ -f /usr/share/openocd/ ...", I get the following output.

"

Open On-Chip Debugger 0.7.0 (2013-10-22-17:42)

Licensed under GNU GPL v2

For bug reports, read

<http://openocd.sourceforge.net/doc/doxygen/bugs.html>

Info : This adapter doesn't support configurable speed

Info : STLINK v2 JTAG v27 API v2 SWIM v6 VID 0x0483 PID 0x3748

Info : Target voltage: 3.269103

Info : stm32f4x.cpu: hardware has 6 breakpoints, 4 watchpoints

target state: halted

target halted due to debug-request, current mode: Thread

xPSR: 0x01000000 pc: 0x0800019c msp: 0x20000450

** Programming Started **

auto erase enabled

Info : device id = 0x10076413

Info : flash size = 512kbytes

"

and thats it. It doesnt write anything. Any Idea why this is happening?

[Reply](#)