Compiling STM32

Beginning STM32: The C way

The book uses a Framework with Libopencm3, FreeRTOS, No IDE, arm cross compiler.

git clone <https://github.com/ve3wwg/stm32f103c8t6.git> -> No IDE template

git clone <https://github.com/libopencm3/libopencm3.git> -> libopencm3

<http://www.freertos.org> -> Download Free RTOS

Download libopencm3 & FreeRTOS and place them under No IDE template

Project settings

In ~/stm32f103c8t6/rtos/Project.mk Change the line:

FREERTOS ?= FreeRTOSv10.0.1 -> to match your FreeRTOS **version**.

**Cross compilation**

1. Go to the site https://developer.arm.com.

2. Click on the link “Linux/Open Source.”

3. Scroll down and click on “ARM GNU Embedded Toolchain.”

4. Scroll down and click on the big button labeled “Downloads.” And download the Linux x86\_64 version.

Then, create a system directory /opt (if you do not already have one):

$ sudo -i

# mkdir /opt

7. Change to the /opt directory (as root):

# cd /opt

8. From this point, you’ll unpack your compiler download. Be sure to be specific about your home directory:

# tar xjf ~myuserid/Downloads/gcc-arm-none-eabi-6-2017-q2-update-mac.tar.bz2

Use tar option “j” if the ending of the file is .bz2. Otherwise, use “z” when the ending is .gz.

Once the tar file has been extracted, it may produce a largedirectory name like gcc-arm-none-eabi-6-2017-q2-update. Nowis a good time to shorten that:

# mv gcc-arm-none-eabi-6-2017-q2-update gcc-arm

This will rename the directory to a more manageable name /opt/gcc-arm.

10. Now, exit root and return to your developer session. In that session in order to access the cross compiler from any directory, add the compiler’s bin directory to your PATH:

$ export PATH="/opt/gcc-arm/bin:$PATH"

11. test your cross compiler: $ arm-none-eabi-gcc --version

**Build software**

Right now Makefiles do all the building process so just enter **$make** command once you are placed on /stm32f103c8t6 directory.