

# ENSE 452 - Lab1

## Keil uVision and simple development

Trevor Douglas  
Software Systems Lab Instructor  
University of Regina

### 1 Objective

The objective of this lab is to get you refreshed with using the Keil uVision tools and program the Nucleo-64 board. You will also put your project under source control (revision control). First you'll get the LED on the board to blink at a certain rate. As you build your code make sure to commit changes and push to your repository as that is where your code will be evaluated.

### 2 Reference Material

Many documents are available on urcourses, under 'Class Resources', but some of the most useful are listed below. There may be updated versions of these documents available online, which you are welcome to substitute.

- STM32 Programmer's Manual: PM0056
- STM32 Reference Manual: RM0008
- Nucleo Board Schematic: MB1136
- STM32F103RB Data Sheet

Our target board is the Nucleo-64, and it has a lot of fun peripherals. The brain of this board is a STMicro STM32F103RB microcontroller, with 128 KiB of Flash, and 20 KiB of static RAM and a 72MHz clock.

### 3 Equipment and Software Requirements

- PC for development.
- Keil uVision software.

- Nucleo-64 development kit and USB Serial cable.
- Git (Git client or use command line).

## 4 Procedure

### 4.1 Phase 1 - Setting up your GIT Repository

You may have already done so in the class, but if not please create a GIT repository of your choice. Some popular choices are GitHub or BitBucket. You will have to create an account to do this. Clone your repository to a local directory on your computer. Now create a folder inside your repository and name it Labs. All your lab work will go into this folder. You must add

### 4.2 Phase 2 - Create a Project

Blinky! (The hello world of embedded)

- Create a folder in you Labs folder named Lab1.
- Search and Select the STM32F103RB device.
- Create a project in Keil uVision. Include the CMSIS core and the Device Startup.
- Add these files to source control, commit and then push to your repository. These are the files that should be put under source control. Once you build your executable a number of build or derived files are created. These should not be put under source control.
- Write the code that will turn on the on board LED.
- While you are writing this code make commits and push to your repository.

### 4.3 Phase 3

At this stage your repository should contain working up to date code. Now test that your repository is working by:

- In a new folder on your host computer (not in your repository) clone your Lab 1 code.
- You should now have a working uVision project with just source and setup code.
- Compile this code and ensure that it all still works.
- Congratulations your repository should be working fine.
- Complete your code to blink the LED at approximately 1s intervals.

## 5 What To Submit

On URCourses put the link to your git repository in the submission link. The TA will checkout the version (date) submitted at the deadline.