**ECE0202: Embedded Processors and Interfacing**

**Lab 1: Installation of KEIL Microcontroller Development Kit (MDK) and Project Creation**

**Due: TBD**

**Objectives**

* **Become familiar with running a project in KEIL**
* **Become familiar with the KEIL debugger interface**
* **Modify assembly code to store custom data into RAM**

**Deliverables – total 50 points**

* **(25 points) Modified code that saves a new hex code of your choice into memory.**
* **(25 points) A screenshot of your last names saved into memory from the debugger.**

Submit your code through Canvas as a \*.s file and the pictures as \*.pdf documents.

Jingtong Hu & David Anderson

1/19/2021

## Section 1: Installation of Keil

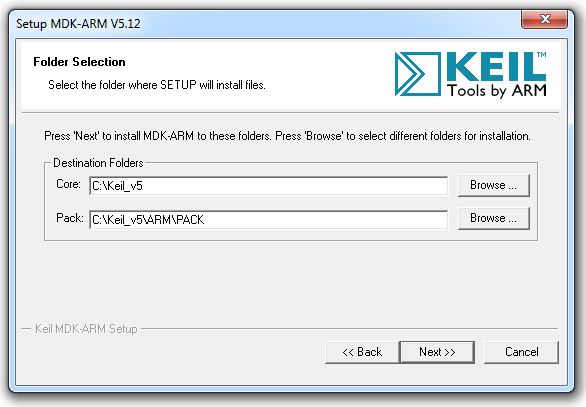
**Warning: Do not connect the lab Kit into your PC or laptop before the software installation completes.** If you connect your kit to PC before installing the USB driver, Windows OS often mistakenly associates a wrong USB driver to the kit. Thus, you will not be able to program the kit. The solution is to go to the control panel and change the USB driver to ST-Link USB driver.

## Step 1: Install Keil MDK-ARM

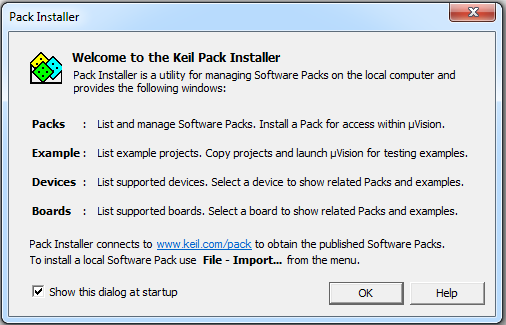
* 1. Download the latest free evaluation version Keil MDK-ARM from the following link:

<https://www.keil.com/demo/eval/arm.htm>.

* Keil MDK-ARM contains µVision 5 IDE (Integrated Development Environment) with debugger, flash programmer and the ARM compiler toolchain.
* The major limitation of the free version is that programs that generate more than 32 Kbytes of code and data will not compile, assemble, or link.
  1. Run the downloaded MDK5xx.exe and install to the default path. The software takes 2GB disk storage space. You can install it to a different driver, instead of the default C drive, if there is limited space in C drive.

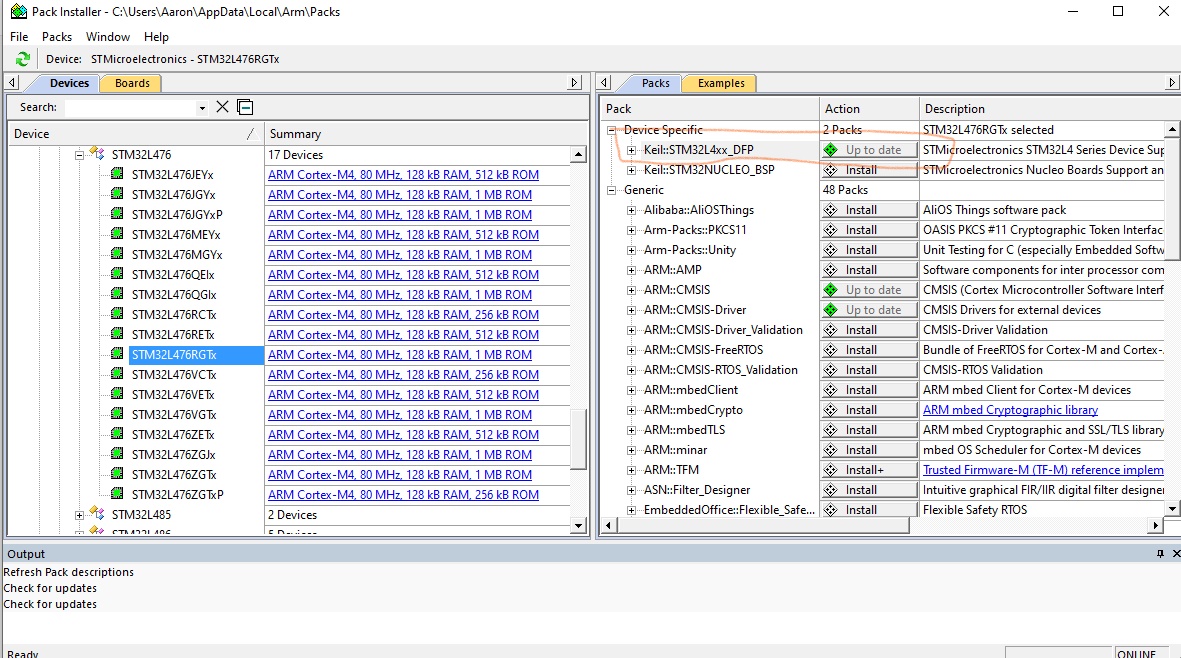


After the core software is installed, a dialog will show up to install Keil Pack. It automatically downloads selected components (called packs) from <http://www.keil.com/dd2/pack/>



Click OK and then the following window shows up.

Select the device **STM32L476RGTx**, install **STM32L4xx\_DFP**.



## Step 2: Install ST-Link USB Driver

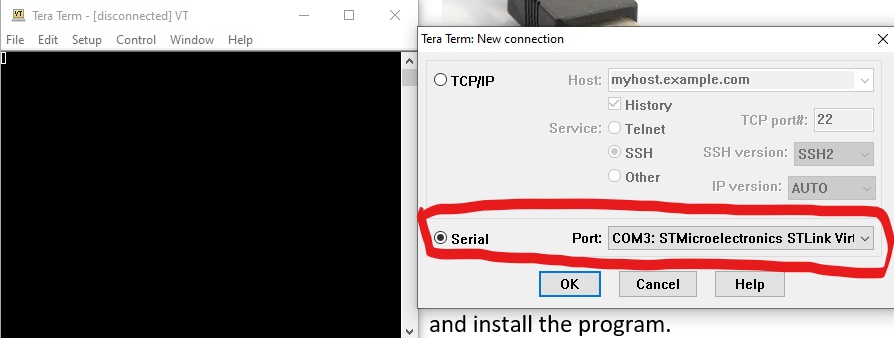
* Do not connect the lab kit before you install the USB driver for ST-Link.
* Go to the directory **C:\Keil\_v5\ARM\STLink\USBDriver** and run **stlink\_winusb\_install.bat** in **administrator mode**.
* Now you can connect the lab kit to computer via a "Type A to mini-B" USB cable. The lab kit should be correctly recognized as “STM32 STLink NODE L476RG”



## Section 2: Install Tera Term

## Since Nucleo-64 board does not have a LCD display, we will install a virtual terminal program, Tera Term, on our computer to display information. Please download the installer from <https://ttssh2.osdn.jp/index.html.en> and install the program.

## After installation, launch the program and select Serial Port: COM3:STMicroelectronics STLink Virtual COM Port and click OK.



## Section 3: Project Creation

## Summary

This tutorial takes the following the kit as an example of creating a project in Keil IDE for assembly programs.

* STM32L476RG MCU (Cortex-M4 with FPU and DSP)

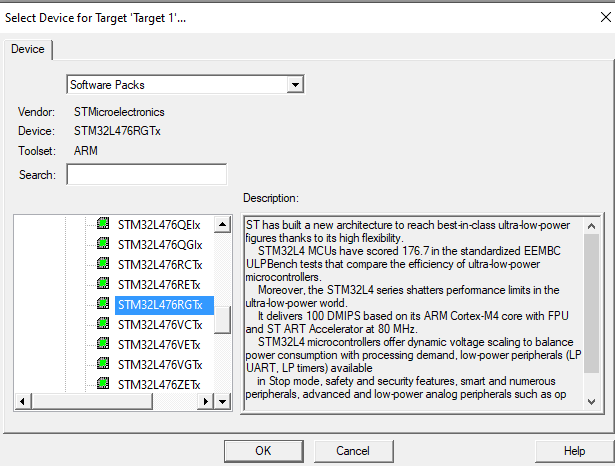
Note that the project does not use the default startup files provided by Keil. You need to download a modified version of ***startup\_stm32l476xx.s*** from Canvas.

## Identifying Target Processor

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kit** | **Processor** | **Core** | **Flash** | **RAM** |
| **STM32L476 Nucleo-64** | **STM32L476RG** | **Cortex-M4 (DSP + FPU)** | **1 MB** | **128 KB** |

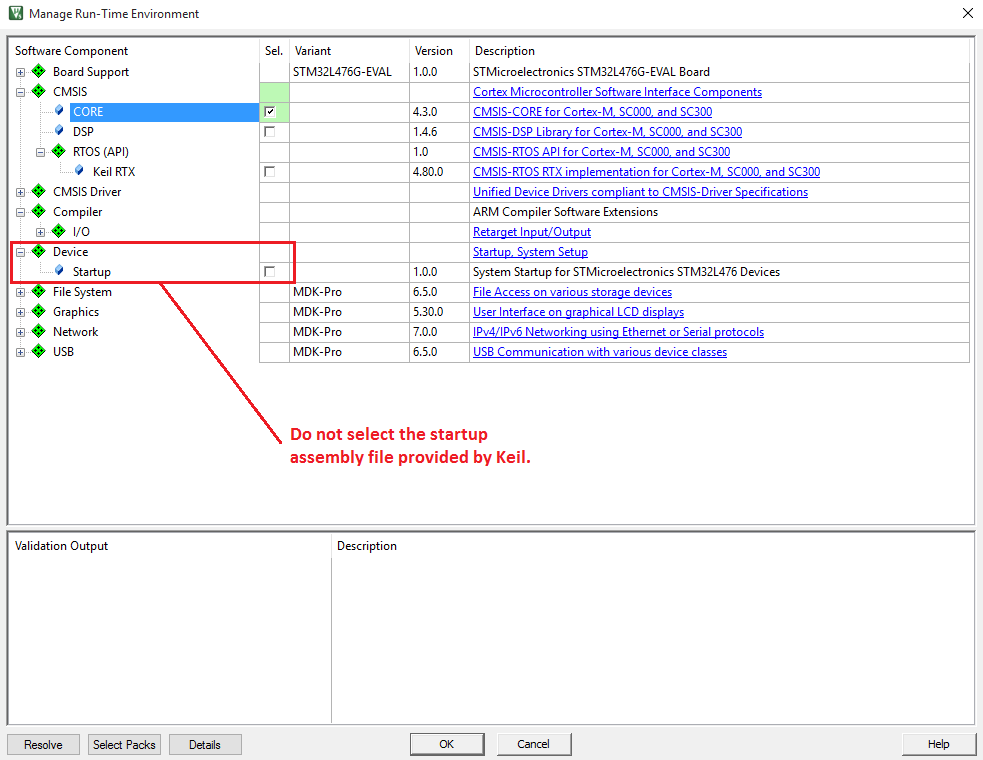
## Steps to create a new project in Keil

1. From the menu **Project** ⟶ **New µVision Project**
2. Give the project a name and select its storage directory. In this tutorial, the project is named as “lab”.
3. Select the device **STM32L4 Series**, and then select **STM32L476RGTx**.



If did not see the targeted processor in the list, click the “**Pack Installer**” button from Menu Project -> Manage and install the component **Keil::STM32L4xx\_DFP**.

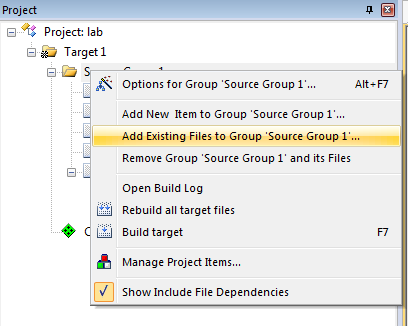
1. Select **CMSIS Core** only. Do NOT select “Device Startup”. Instead, you should use the one provided by the course website.



1. To create a project, you should download all the template files from Canvas and put them in the project folder. Then right click on the folder “Target 1” and add the following files as shown in the following figure:

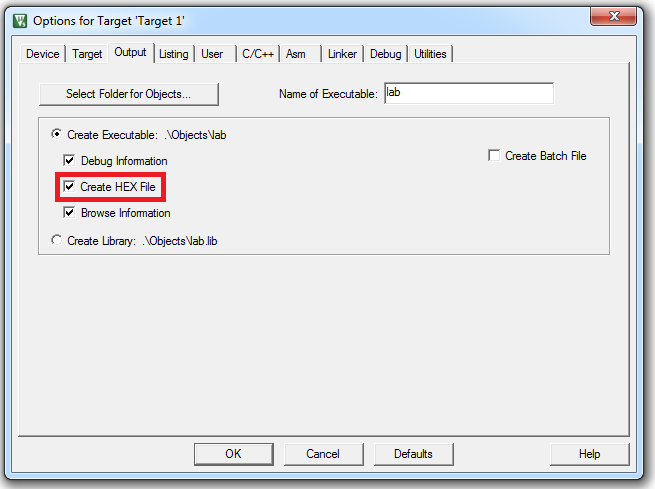
* **startup\_stm32l476xx.s**
* **main.s**
* core\_cm4\_constrants.s
* stm32l476xx\_constants.s
* SysClock.c
* UART.c

Note that SysClock.h and UART.h are not added to the project, but these files must be in the project folder because they are used by SysClokc.c and UART.c

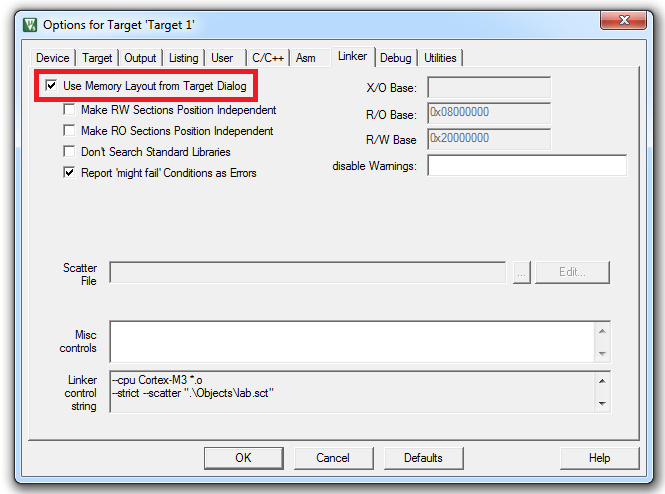


1. **Set Project Properties**

From the menu, click **Project** ⟶ **Option for Target**, Go to the **Output** page, select “**Create HEX file**”

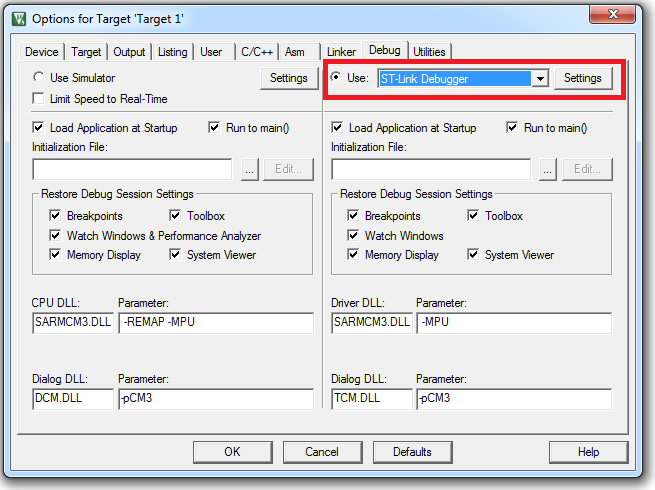


Go to the **Linker** page, select “**Use Memory Layout from Target Dialog**”

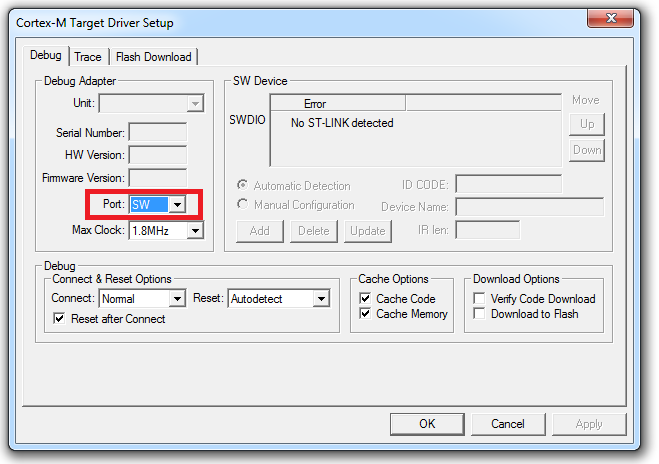


Go to the **Debug** page, select “**ST-Link Debugger”**

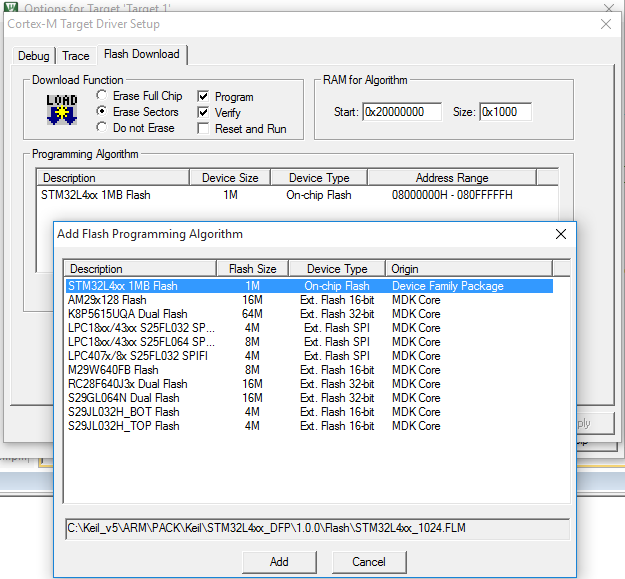
**If you see a firmware upgrade notification, please follow the steps to upgrade the firmware.**



Click “**Settings**” and select “**SW**” (Serial Wire) as the port.

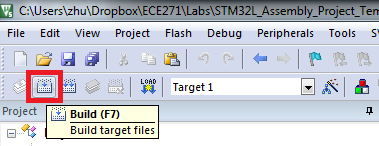


Go to the **Flash Download** page, and verity that **STM32L4xx On-chip Flash** is selected in the Programming Algorithm. If not, click “Add” and select STM32L4xx On-chip flash in the popped dialog.



1. **Compile and run your project**

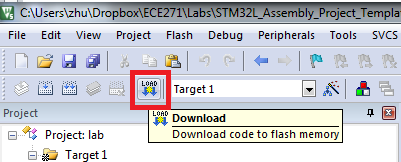
Build the program:



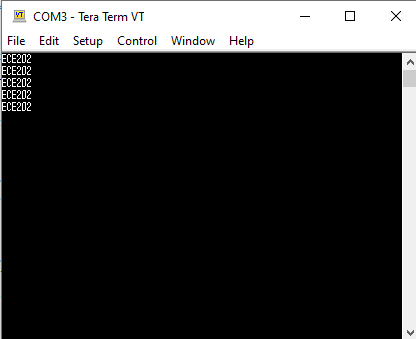
You can ignore the following warning when the linking stage:

.\Objects\lab.sct(8): warning: **L6314W: No section matches pattern \*(InRoot$$Sections)**.

Connect your lab kit to the computer and download the program to the STM32L processor.



Push the black Reset button on the board, and your program will start. By default, ECE0202 will be displayed on the Tera Term each time you push the reset button.



**Please replace the "ECE0202" in main.s with your last name, compile and download again. Your last name will be displayed on the LCD. Please do this for both team members.**

## Section 4: Debug

Follow the steps in course lecture. Take a screenshot of your last names saved into memory from the debugger