

MECHTRON 3TA4: Laboratory 1

Introduction

This assignment will introduce you to the software and hardware development tools you will use during the semester to work with the STM32L476Discovery development board and the Cortex-M4 ARM Processor so that you may run programs using the Keil uVision SDK. The example code provided with this assignment shows you:

- how to setup a project using Keil uVision;
- how to use timer interrupts to interface with the LEDs in the STM32L476-Discovery board using the "user push-button" on the board.

Prelab

1. Familiarize yourself with the embedded C language and the STM32L4 Standard Peripheral Library that comes with your STM32L4-Discovery software. Specifically, you will need to concentrate on the following material:
 - o Reading Chapters 1, 2, 14 of "Embedded Systems with ARM Cortex-M Microcontrollers in Assembly and C" by Yifeng Zhu.
 - o The STM32L4 Low Layer drivers manual found [here](#).
2. Familiarize yourself with the architecture of the Cortex-M4 processor by
 - o the Cortex-M4 provides a superset of the Cortex-M3 instruction set.
 - o reading the Cortex-M4 reference manual [online](#).
3. Familiarize yourself with the structure of the STM32L476-Discovery board.
 - o The STM32L476-Discovery Board User Guide found on avenue under content/ Manuals & DataSheets/en.DM0017217
4. Familiarize yourself with the timers of the Cortex-M4 in the STM32L476-Discovery Board.
 - o Read corresponding chapters of the reference manual here: [DM00083560.pdf](#).
5. Familiarize yourself with the use of Keil's development tools by working through the first part of the PDF into together with Zip file of code below:
 - o CortexTM-M4 Training STM32L476 Discovery evaluation board using ARM® KeilTM MDK Toolkit: found on avenue under content/ Manuals & DataSheets/en.DM00217936
 - o https://www.st.com/content/ccc/resource/technical/document/user_manual/4/0/8a/04/4b/cf/2a/4c/73/DM00213619.pdf/files/DM00213619.pdf/jcr:content/tranlations/en.DM00213619.pdf
 - o https://www.st.com/content/ccc/resource/technical/document/user_manual/7/4/09/3d/80/f9/39/4c/c7/DM00157440.pdf/files/DM00157440.pdf/jcr:content/tranlations/en.DM00157440.pdf
 - o <https://pdfs.semanticscholar.org/presentation/0a9a/2ab2b2acb501d40e8b2b096ddc22f7fe7192.pdf>

- Explore some project files in the examples directory in your firmware folder.

Hardware

STM32L476-Discovery

The STM32L4-Discovery board you will be using to support the Cortex-M4 processor is a small board providing:

- A target STM32L476 microcontroller with onboard flash memory.
- An embedded ST-LINK/V2 interface to program the flash memory through a mini-USB connection.
- A separate micro-USB connection for applications.
- One user push-button, one reset button and two LEDs.

For this lab you will not need to use additional connectors except for the mini-USB, however, in the rest of the labs a 50-pin header that connects to the P2 set of pins (on the side of the reset button) will be used.

The STM32L476-Discovery is powered via the mini-USB connection and no external power supply will be needed although this option is available. When using the mini-USB connection to power the board you must be very careful that any other peripherals that are using either the 3V or 5V output do not exceed 100mA otherwise you will burn the mini-USB interface on the board.

Software

The software you will use are:

- Keil uVision where you will develop and compile your software.
- Stm32L476-Discovery firmware package
- ST-LINK/V2 link utility and USB drivers.

Procedure

1. Login to the PC at your lab station.
2. Download the Lab1 project from avenue under content/labs/lab1_.
3. Extract the lab1 starter project into your working folder. Make sure it is in a folder called lab1.
4. Connect your stm32l4-discovery board to a USB port of your choice. Let windows try to find the drivers. If the drivers are not found then the st-link usb drivers are missing and you will need assistance from the TAs.
5. Find the shortcut to the Keil uVision IDE. Start uVision.

6. Select "Project->Open Project" and select the ".uvproj" file in your starter project. This will load the starter project in your Keil uVision IDE.
7. Make sure the project configuration is correct:
 - Right click the project folder and select "Options for lab1" or select "Project->Options for lab1".
 - Under the "Device" tab make sure the device selected is "STM32L476VG" as printed on your discovery board.
 - Under the "Target" tab make sure "Use MicroLIB" has a checkmark.
 - Under the "Output" tab make sure "Create HEX File" has a checkmark.
 - Under the "C/C++" tab
 - Make sure the "Define" box has "USE_STDPERIPH_DRIVER STM32L4XX" in it (without the quotes). This enables all the source files you include from the stm32l4discovery-fw folder to automatically load the appropriate header files, otherwise you will have to modify the source files and add all the necessary headers.
 - Make sure "One ELF Section per Function" is checked.
 - Make sure your include paths are correct. By default they should be ".\inc;..\STM32Cube_FW_L4_V1.8.0\Libraries\CMSIS\Include;..\STM32Cube_FW_L4_V1.8.0\Libraries\CMSIS\Device\ST\STM32F4xx\Include;..\STM32Cube_FW_L4_V1.8.0\Libraries\CMSIS\Device\ST\STM32F4xx\Source;..\STM32Cube_FW_L4_V1.8.0\Libraries\STM32F4xx_StdPeriph_Driver\inc;..\STM32Cube_FW_L4_V1.8.0\Libraries\STM32F4xx_StdPeriph_Driver\src;..\STM32Cube_FW_L4_V1.8.0\Utilities\STM32F429I-Discovery;..\STM32Cube_FW_L4_V1.8.0\Utilities\Common", however, if you have a different folder structure you will need to update this setting to the appropriate header file folders.
 - Under the "Linker" tab make sure that "Use Memory Layout from Target Dialog" and "report 'might fail' conditions as errors" are both checked.
 - Under the "Debug" tab make sure to select "ST-Link Debugger" in the dropdown beside "Use" and that you are not using the simulator. Then click "Settings" and make sure "SW" (single-wire debug) is selected as the debugging interface under the "Debug" tab; also check that "STM32L4xx Flash" is the only listed item under "Flash Download". Make sure there's a checkmark besides "Run to main()".
 - Under the "Utilities" tab make sure "Use Target Driver for Flash Programming" is selected.
8. If the source files appearing in your "Workspace" window are not found make sure you placed the files in the appropriate directories.
9. Take special note of the startup_stm32l4xx.s assembly file. This file defines the startup procedure before getting to your main() function as well as defines all the available interrupts. This can come in handy when you want to quickly find the name of the function for a specific interrupt (eg. TIM3_IRQHandler, EXTI0_IRQHandler, etc.).
10. Compile your lab by selecting "Project->Build" or by pressing F7 and make sure there are no errors in compilation. If errors are present make sure to carefully investigate what is

missing (ie. header or source files at compile-time, missing dependencies or implicit declarations at linker-time and so forth) before asking assistance from a TA.

11. Load the HEX file produced by the toolchain onto your board by selecting "Flash->Download"
12. Press the "Reset" button on the STM32L4-Discovery board.
13. You should see LED3 blink.
14. Press the "User Button" on the STM32L4-Discovery board and you should see LED3 blinking at a different rate.
15. If you would like to Debug your code step by step select "Debug->Start/Stop Debug Session" or press Ctrl+F5.

Important: For more thorough documentation on using Keil uVision go to your Keil uVision IDE and click the "Books" tabs on the bottom left. This tabs provides you with all the resources and information needed to use Keil uVision as well as information on programming with Cortex processors.

Timing of all functions in this lab, *and most exercises in this course* will be handled by interrupt-driven counters, not by software wait-loops.

Assignment

Using the short demo code downloaded above, modify the code:

1. **[15 pts]** Make LED4 blink instead of LED3 when the reset button is pressed (ie. at the start of your program).
2. **[55 pts]** Pressing the User Button will make the two LEDs blink in a turn-taking manner: LED3 on, LED3 off, LED4 on and then LED4 off, and then repeat. Also, modify the code so that they blink at 1Hz, i.e. for each LED the on-time and off-time should add up to 1 second, and it takes 2 seconds in total to blink both LEDs once.
3. **[30 pts]** Use TIM2 instead of TIM3.

Hint: Look through the comments in the C code and also through the TIM_TimeBase example of the STM32L4-Discovery board TIM examples.

There is no written report for Lab 1, but attendance is mandatory and you will be required to demo your program to the TA. Be prepared to answer questions about your work.