Be Boulder.

PROJECT #2 GUIDE

UCB ECEN 5003 FALL 2016 PROJECT #2: ST ARM CORTEX M4 MBED

TABLE OF CONTENTS

INTRODUCTION

NUCLEO-F401RE

Affordable and flexible platform to ease prototyping using a STM32F401RET6 microcontroller.

The STM32 Nucleo board provides an affordable and flexible way for users to try out new ideas and build prototypes with any STM32 microcontroller line, choosing from the various combinations of performance, power consumption and features.

The Arduino™ connectivity support and ST Morpho headers make it easy to expand the functionality of the STM32 Nucleo open development platform with a wide choice of specialized shields.

The STM32 Nucleo board does not require any separate probe as it integrates the ST-LINK/V2-1 debugger/programmer.

MICROCONTROLLER FEATURES

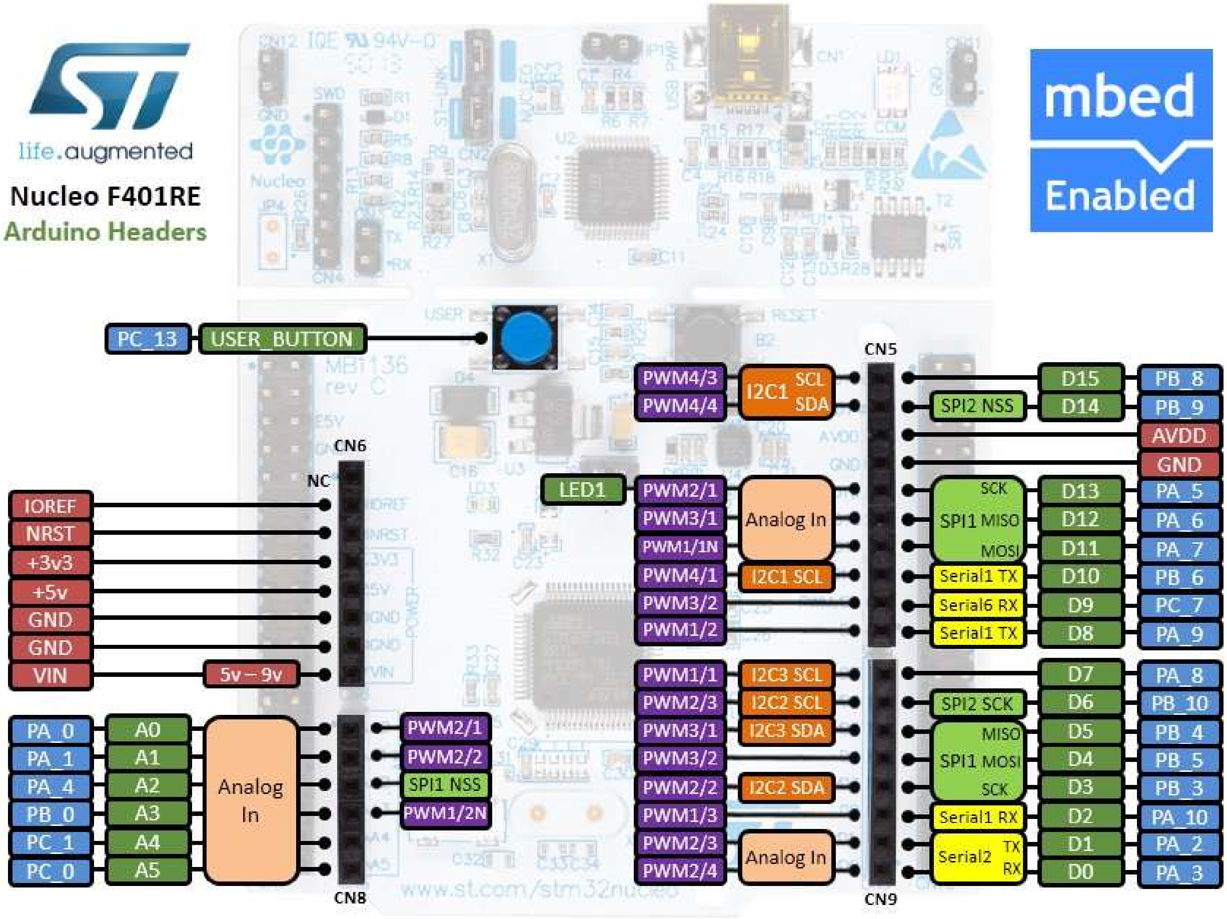
- STM32F401RET6 in LQFP64 package
- ARM®32-bit Cortex®-M4 CPU
- 84 MHz max CPU frequency
- VDD from 1.7 V to 3.6 V
- 512 KB Flash
- 96 KB SRAM
- GPIO (50) with external interrupt capability
- 12-bit ADC with 16 channels
- RTC
- Timers (8)
- I2C (3)
- USART (3)
- SPI (3)
- USB OTG FS
- SDIO

NUCLEO FEATURES

- Two types of extension resources
  - Arduino Uno Revision 3 connectivity
  - STMicroelectronics Morpho extension pin headers for full access to all STM32 I/Os
- On-board ST-LINK/V2-1 debugger/programmer with SWD connector
  - Selection-mode switch to use the kit as a standalone ST-LINK/V2-1
- Flexible board power supply
  - USB VBUS or external source (3.3 V, 5 V, 7 - 12 V)
  - Power management access point
- User LED (LD2)
- Two push buttons: USER and RESET
- USB re-enumeration capability: three different interfaces supported on USB
  - Virtual Com port
  - Mass storage (USB Disk drive) for drag'n'drop programming
  - Debug port

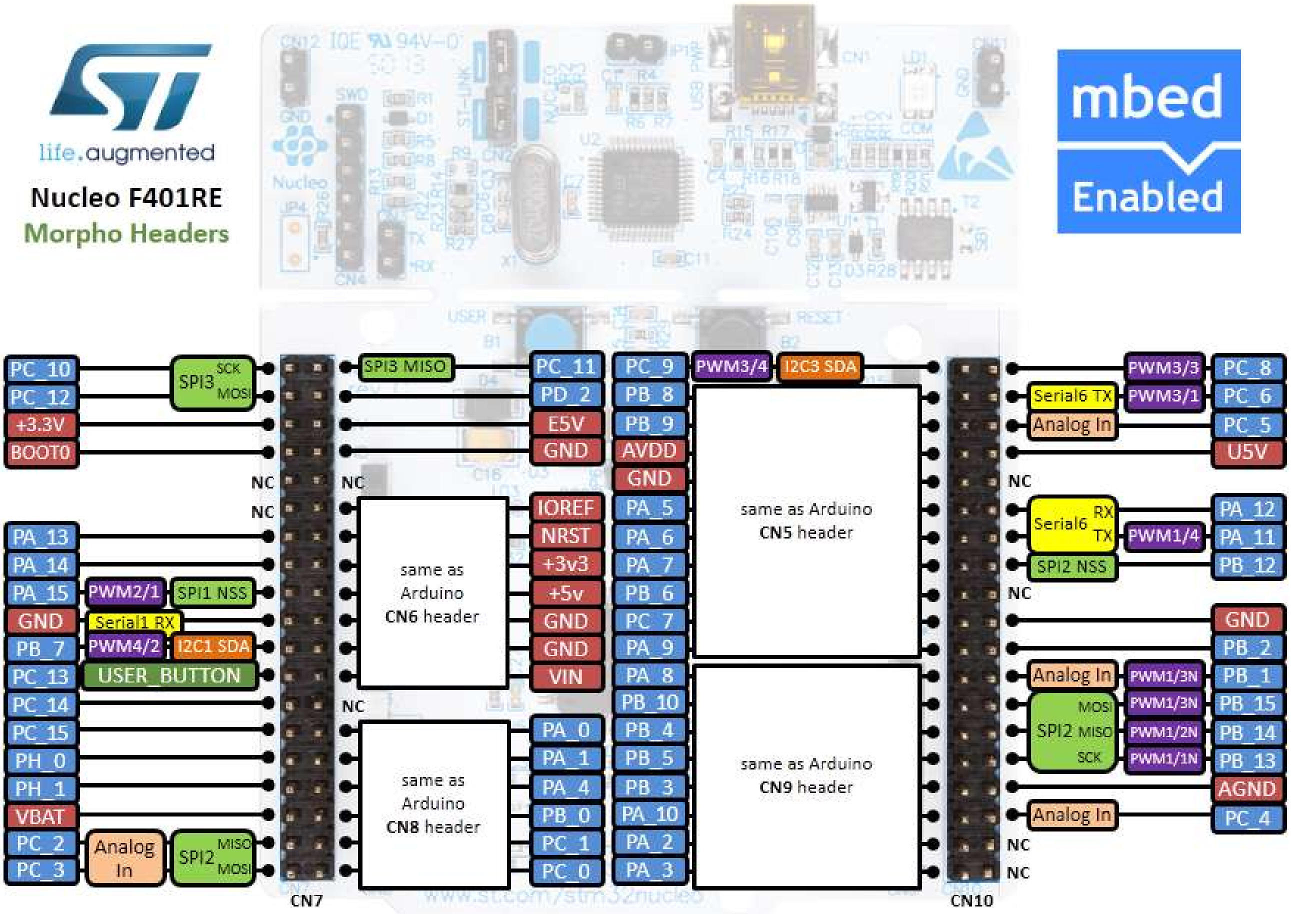
NUCLEO PINOUT

ARDUINO-COMPATIBLE HEADERS



MORPHO HEADERS

These headers give access to all STM32 pins.



GETTING STARTED

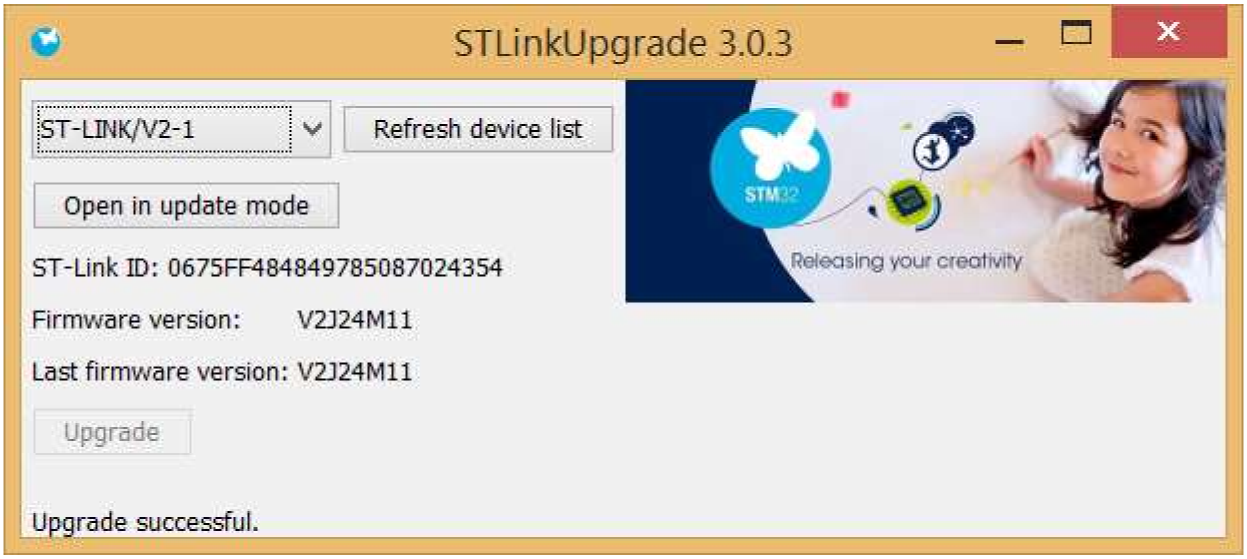
This video shows how to get started with ARM mbed Integrated Development Environment using STM32 Nucleo platform:

https://youtu.be/BrMw5TNQROo

NUCLEO ST-LINK/V2 DRIVER INSTALLATION AND FIRMWARE UPGRADE – VERY IMPORTANT!

- Install the ST-LINK/V2 driver before connecting the Nucleo board to your PC the first time. Follow this LINK for all details.

- For optimum performances, ensure that the Nucleo ST-LINK/V2 firmware is upgraded to the latest version. Follow this LINK for all details. CAUTION: the upgrade software in the link may not work on your system. The program to run is now listed as STLinkUpgrade.jar. Running it will give you this window:



ST-Link ID: 0675FF484849785087024354

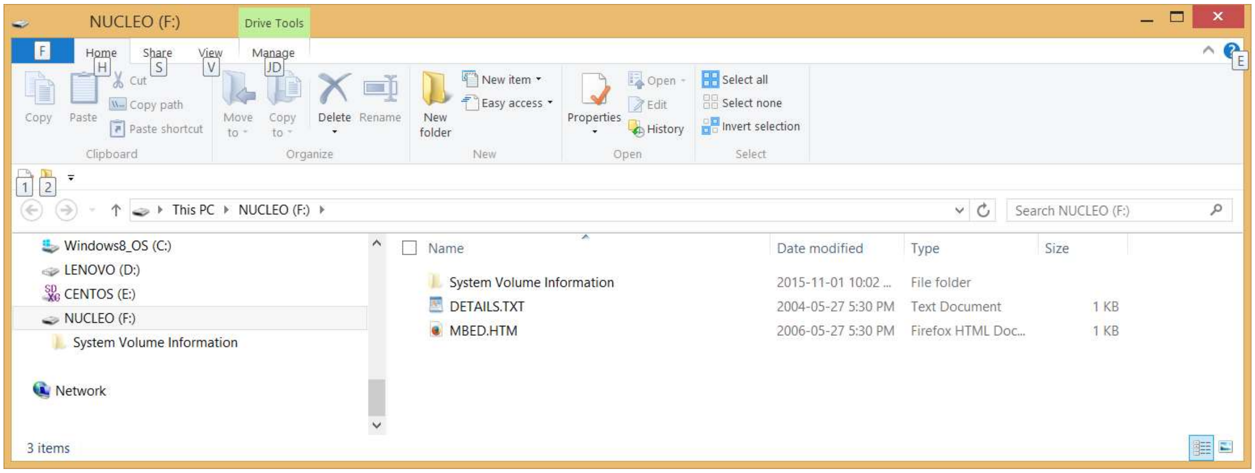
Firmware version: V2J24M11

Last firmware version: V2J24M11

Upgrade successful.

You must click on "Open in update mode" to perform the update. Note that the latest version is V2J24M11. Many boards will have V2J24M4 or similar initially, so need to be updated.

When you are successful, you will see this window when connecting to the Nucleo board:



Name	Date modified	Type	Size
System Volume Information	2015-11-01 10:02 ...	File folder	
DETAILS.TXT	2004-05-27 5:30 PM	Text Document	1 KB
MBED.HTM	2006-05-27 5:30 PM	Firefox HTML Doc...	1 KB

TECHNICAL REFERENCES

For more information, please refer to:

- STM32F401RE Microcontroller
- Nucleo board
- SDK changes log

I. SETUP MBED

GETTING STARTED WITH MBED

1. CONNECT YOUR MBED-ENABLED NUCLEO 401RE TO A PC

Use the USB lead to connect your Nucleo 401RE to a PC, using the miniUSB connector. The power LED light will come on, indicating it has power. After a few seconds of activity, the PC will recognize the mbed Microcontroller as a standard USB drive.

2. CLICK THE MBED.HTM LINK TO GET LOGGED IN

Go to the new USB Drive, and click MBED.HTM to open it in a web browser.

If you do not have an mbed account, choose "Signup", and create your mbed Account. Otherwise, log in with your normal username and password.

This will give you access to the website, tools, libraries and documentation.

3. HAVE FUN!

II. SETUP MDK

REQUIREMENTS

- Software:
  - Keil MDK
    - If have not already done this in Project 1, you can download a free version of Keil MDK-ARM tool (MDK-Lite) here: https://www.keil.com/arm/demo/eval/arm.htm. You do not need a serial number or license key for this version.
    - If you are seeing error messages that indicate you need the Keil MDK legacy support, you may download the Keil MDK5 legacy support at: http://www.keil.com/download/files/mdkcm510.exe

- Hardware:
  - Nucleo F401RE: http://mbed.org/platforms/ST-Nucleo-F401RE/

HARDWARE SETUP

To enable the download and debugging features on your Nucleo F401RE board, use the following steps to upgrade the firmware:

1. Download the firmware from: http://mbed.org/teams/ST/wiki/Nucleo-Firmware
2. Follow the steps on the web page.

More details on debugging can be found at:

https://mbed.org/blog/entry/Debugging-on-mbed-enabled-platforms/

PROJECT DEVELOPMENT WITH MDK

LOADING A PROJECT INTO MDK

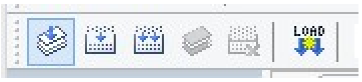
Project files are displayed in File Explorer with the Microvision icon, shown above. To load a project into MDK, do one of the following:

- Menu: Within MDK, select menu item Project->Open Project…, navigate to the project directory, and select the .uvproj file.
- File Explorer: double click on the .uvproj file.

Note that for all lab exercises, the Keil MDK project has been created for you, so that you can simply open and modify using Keil MKD as required.

Alternatively, you can use the mbed online compiler to create and develop a project (follow this link http://mbed.org/handbook/mbed-Compiler for details on how to do this). The online project can also be downloaded and ported to Keil MDK running on your local machine (see http://www.keil.com/appnotes/docs/apnt_207.asp for details on how to do this).

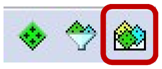
BUILDING THE PROJECT



Build the project using one or more of the following toolbar buttons (listed from left to right):

- Translate current file (e.g. compile or assemble)
- Build the target files whose source files have changed and create output file
- Rebuild all of the target files and create output file

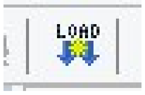
- You may need to install the hardware support package for your platform. This can be done by clicking the package installer button in Keil MDK:



DOWNLOADING THE PROGRAM IMAGE TO THE MICROCONTROLLER PROGAM MEMORY (FLASH MEMORY)

Download the program to the MCU flash using one of these methods:

- Toolbar button:



- Menu: Flash->Download
- Accelerator keys: Alt+A+D

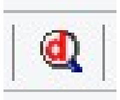
Note: if the download fails, please check:

- The firmware has been installed on mbed board (hardware setup)
- Install Keil MDK5 legacy support (http://www.keil.com/download/files/mdkcm510.exe)

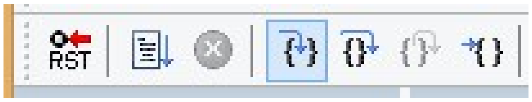
USING THE DEBUGGER

Begin or end a debugger session using one of these methods:

- Toolbar button:



- Menu: Debug->Start/Stop Debug Session
- Accelerator keys: ctrl+F5



Control the target program execution with the following toolbar buttons (shown from left to right above):

- Reset MCU
- Run program execution (F5)
- Stop program execution
- Step one line in program, entering a subroutine (F11)
- Step one line in program, executing and returning from a subroutine (F10)
- Step out of current function (ctrl+F11)
- Run to cursor (ctrl+F10)

Right-clicking on a line of code will bring up a context menu with various options, including:

- Setting and clearing breakpoints
- Adding a variable to a watch window

- Navigating to definitions or uses of symbols (functions and variables)

The View menu can be used to open different windows to help in debugging, including:

- Disassembly
- Symbols
- CPU registers
- Call stack
- Variable watch windows
- Memory windows
- System viewer (MCU and peripheral control registers)

III. MODULE 1. HELLO WORLD, WHAT TIME IS IT?

1. Use the results from Homework E to complete this module.
2. Where (at what address) does the Reset handler begin in the memory map?
3. How much memory is used by the code (Led blinking code for the homework)?
4. Run the mBed Nucleo Example (display_time). Set the time to the current time, and combine this with you're the mBed Nucleo Example (printf) to print the current time to a terminal window on your PC. Capture a screen shot of the terminal window. How much memory is used by this code?
5. Explain the memory model of ARM Cortex-M4 with respect to the code memory, data memory, IRQ handlers and peripherals. Explain with the help of a diagram where required.
6. As a separate project, run either the Dhrystone or the Whetstone benchmark program on your target processor using the code provided, which may need to be modified. If running the Dhrystone, calculate the number of VAX DMIPS.

IV. MODULE 2. BUTTON READ, ADC READ, LED PWM, AND UART

1. From D2L, download the word document Lab_Exercise2_2.zip and Code22.zip and unzip.
2. This module will require you to interface some hardware.
3. Follow the Lab_Exercise2_2 directions, which include exercises 8.1 (button), 8.2(interrupt), 9 (analog), and 11.4 (integration). However please note that any screenshot present in the documents are for your reference, but they may not be in sync with the current version of tools.
4. Answer the questions in the Lab documents.

V. MODULE 3. RTOS THREADS

1. From D2L, download the word document Lab_Exercise3_3.docx.
2. Follow the Lab_Exercise3_3 directions. However, notice you can use the mbed RTOS or the CMSIS RTOS (Keil RTX) for this module.
3. Answer the questions in the Lab document.

VI. MODULE 4. SIGNAL ANALYZER

1. Create a program to more fully evaluate the ST STM32F401RE MCU for use in a Signal Analyzer:
    a. Write a program either using the mbed compiler, or in the Keil MDK, to generate a 1004 Hz tone through the DAC. Simulate this program in Simulink. Please note if there is no DAC module present, a software DAC function would also work and store the samples generated into a .csv or .txt file.
    b. Write a program either using the mbed compiler, or in the Keil MDK, to read a 1004 Hz tone from a file into the ADC and perform a harmonic analysis. Simulate this program in Simulink. Use the samples from your .csv or .txt file from the previous question for the input to the analyzer. (NOTE: please do not use arrays to input values from your csv file to your program).
    c. Blink the User LED at a rate proportional to the ADC input fundamental frequency.
2. Run Splint or an equivalent code checker on your code in 1 of this module. Resolve any errors, explain the warnings.
3. Auto-generate documentation using Doxygen on your code in 1 of this module. Provide either an HTML directory or PDF file documenting your codebase.
4. Estimate the processor load in % of CPU cycles.