**Getting started with STM32f429 Discovery Board**

Robert Li (Modified Jan 2023)

This document gives some details of how to download and configure two software components which work together for creating projects that run on STM32f429 Discovery board, which will be used in all labs for Mechtron 2TA4.

1. **Install STM32CubeMX**
   * Download CubeMX from <https://www.st.com/en/development-tools/stm32cubemx.html> .
   * Install CubeMX. This step may require you to install Java Runtime Environment. (in case need the required version of Java JRE, you can download it from: <https://www.oracle.com/java/technologies/javase-jre8-downloads.html> ).
2. **Create Project Using STM32CubeMX**
   * The new version of KEIL uVision requires CubeMX in the process of creating new project. Therefore, you must install CubeMX before using KEIL uVision.
   * You can create your project using Keil uVision, or using CubeMX. A video tutorial on Youtube for getting started with CubeMX may be helpful to you: https://www.youtube.com/watch?v=szMGedsp9jc
   * To create your project using CubeMX, open STM32CubeMX, start project by selecting a ST board and choose “initialize all peripherals with their default mode”

**Note**: *For creating a project for STM32F429I\_DISC boards, if start with selecting a MCU, in the step of configuring clock, there will be no choice of clock source of HSE. This is not the real case for STM32F429I-DISC boards. Therefore, it is better for users to start project by choosing from a ST board, and choose to “initialize all peripherals with their default mode” (if not initialize with default mode, there won’t be HSE available either).*

**For Clock Configuration:**

1. Choose clock source as HSE and set its value to 8 (MHz)
2. Set the PPLM (/M) to 4
3. Set the PPLN (/N) to 72
4. Set the PPLP (/P) to 2 (this will make the SYSCLK=72 MHz)
5. Set the PPLQ (/Q) to 3
6. For “System Clock Mux”, choose PLLCLK
7. Set AHB Prescaler to 1 (this will make the HCLK=72 MHz)
8. Set APB1 Prescaler to 2 (this will make the PCLK1=36 MHz (APB1 peripheral clocks), while APB1 timer clocks is 72MHz)
9. Set APB2 Prescaler to 2 (this will PCLK2=36 MHz (APB1 peripheral clock, while APB2 timer clocks is 72MHz)

**For Pinout and Configuration:**

Categories of Middleware:

1. In the left panel, under Categories of Middleware, select FREERTOS, in the middle panel of “FREERTOS Mode and Configuration”, under the “Mode” section, set “Interface” to Disable
2. Also for the Middleware of “USB\_HOST”, set its “Class for “HS IP” to Disable

Timers:

**Note: *Configuration of timers required only if they are being used in a project e.g. you can skip this step for lab0 but must include for lab1-2.***

Make the Clock Source for all the timers “ Disable” except for TIM3 and TIM4.

* For TIM3: (Configure TIM3 to have an overflow interrupt every second)
  + In the Mode section, Set “Clock Source” to “Internal Clock”
  + In the Configuration section, on tab of Parameter Settings, set:

Prescaler: 7199, which is (72Mhz/10 Khz)-1, which will make the TIM3 counter clock as 10 KHz

Counter Mode: Up

Counter Period: 9999, (which is 10,000-1, for the 10KHz counter clock, count from 0 to 9999 need 1 second)

Internal Clock Divisiion: 0 or No Division.

Auto\_reload Preload: Enable

* + In the Configuration section, on tab of NVIC:

Enable TIM3 global interrupt.

* For TIM4: (Configure TIM4 to have an Output Compare interrupt every 2 second)
  + In the Mode section:

Set “Clock Source” to “Internal Clock”

For Channel 1. Select “Output Compare No Output”

* + In the Configuration section, on tab of Parameter Settings, set:

For Counter Settings:

Prescaler: 7199, which is (72Mhz/10 KHz)-1, which will make the TIM3 counter clock as 10 KHz

Counter Mode: Up

Counter Period: leave it as default or set it to a value larger than 20000, which we will use as the CCR value.

Internal Clock Division: 0 or No Division.

Auto\_reload Preload: Enable

For Output Compare No Output Channel 1 settings:

Set Pulse to : 20,000 ( which will take the TIM4 counter clock 2 second count 20,000 ticks)

* + In the Configuration section, on tab of NVIC:

Enable TIM4 global interrupt.

System Core:

***Only if the use of interrupt mask is required for the “User Button”, proceed as follows:***

Select GPIO in the left panel, in the middle panel of Configuration , select GPIO tab.

Select pin PA0. In the lower part of the middle panel, which is for PA0 configuration:

Set GPIO mode to “External Interrupt Mode with Falling edge trigger detection”

In the middle panel of configuration, select NVIC tab, and enable the EXTI Line0 interrupt.

PA0 is the blue user button on STM32F429\_DISCO board.

Notice the pins PG13 and PG14, which are for the LED3 and LED4 on the STM32F429I-DISC board. The CubeMX has already initialized them so we do not need to do any change here for this project.

**For Project Manager :**

1. Give a Project Name (Use lab0)
2. Choose a project location
3. For Toolchain/IDE, choose “MDK-ARM”

**Then “Generate Code”**

A dialog box confirms successful generation of code. Check the project folder that contains 3 folders: Core, Drivers and MDK-ARM and a file named “lab0.ioc”. Exit from STM32CubeMX.

Open the folder named MDK-ARM and locate a file named “lab0.uvprojx”. This file can be used to open the project in uVision 5, which can be installed as detailed below:

1. **Install Keil uVision 5**

More information about Keil uVision5 and its installation may be found on the Keil’s websites, such as: http://www2.keil.com/mdk5/install/ .

1. Download and install the Keil uVision 5

Download the Keil uVision 5 (Keil MDK-ARM Version 5) .

<http://www2.keil.com/mdk5/install>

or from:

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

1. Install Keil uVision 5
2. Install necessary packs

When install, change the packs’ path from the user’s folder to a more general folder, such as under c:\Keil\_V5\Packs (may need to create a Packs folder under c:\Keil\_V5) . This may be good for a multiuser system.

During installation , the Pack Installeris launched automatically. Some packs will be installed automatically. These packs include:

A: ARM::CMSIS\_

B: Keil::MDK.Middleware

But we need to manually install the firmware package for STMF32F429I-DISCO boards.

In the Device panel, which is at the left side of the Pack Installer window (If the Pack Installer Window is not open anymore, open it by clicking “Project | Manage | Pack Installer…” in Keil uVision5 ), find STMicroelectronics🡪STM32F4 Series🡪STM32F429🡪 STM32F429ZI🡪STM32429ZITx, then in the Pack panel, under section of Device Specific, click Install next to “Keil::STM32F4xx\_DFP”. Please see Figure 1.

The firmware package for Boards STMF32F429I-DISCO boards can also be installed by browsing to the boards name in the Boards panel in the Pack Installer.

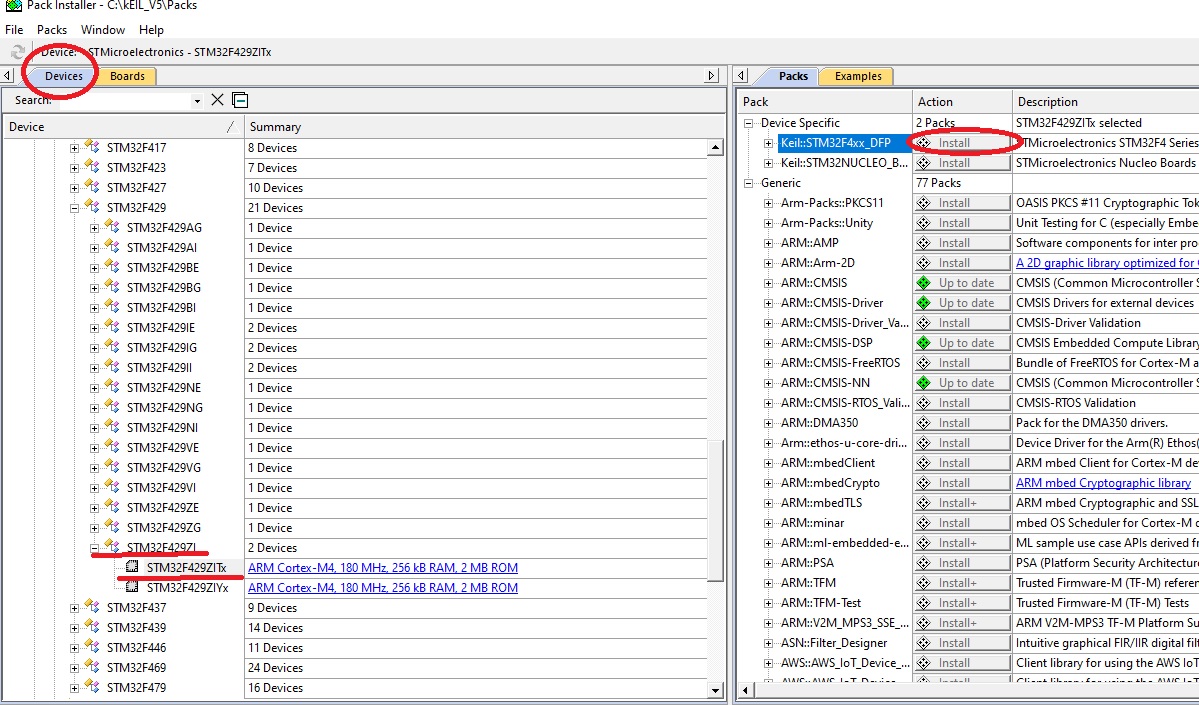


Figure 1: Pack Installer

1. For the STM32CubeF4 firmware library. Users DO NOT NEED to download it, because when install the packs, the firmware library is installed under …/Packs/Keil/STM32F4xx\_DFP/
2. **Open and build the project created by STM32CubeMX**

In uVision 5, go to Project | Open Project…., and then brows the project folder, find and select the project named lab0.uvprojx under the folder “MDK-ARM”, then open the project. Follow the instructions given in lab0.pdf document to finish your lab0 project.

1. **Create a new project using Keil uVision directly (The following are for reference only)**
2. In uVision 5, go to Project | New uVision Project….., select a folder for project and give the project a name, then click Save, to save the project.
3. In the window to Select Device for Target, choose device: STMicroelectronics | STM32F4 Series | STM32F429 | STM32F429ZI | STM32F429ZITx, then click OK. The window “Manage Run-Time Environment” will open. The “Manage Run-Time Environment” window can also be opened from uVision by selecting : Project | Manage | Run-Time Environment.

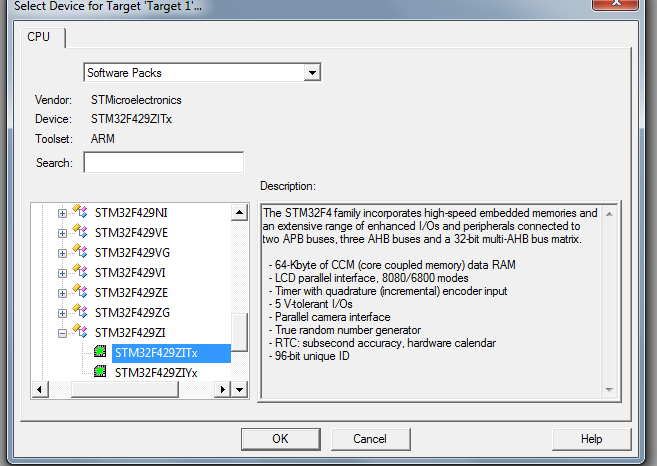


Figure 2: Select Device for STM32F429I-DISCO

1. In the “Manage Run-Time Environment” window, select the following options:

1. Board Support | Buttons (API) | Buttons
2. Board Support | LED (API) | LED
3. CMSIS | Core
4. Device | Startup
5. Device | STM32Cube Framework(API) | Classic , and then click Resolve button at the bottom of the Manage RTE window to solve the package dependency issues. ( by clicking Resolve button at this moment, the following package will be selected automatically:

* Device | STM32Cube HAL | Common
* Device | STM32Cube HAL | Cortex
* Device | STM32Cube HAL | GPIO
* Device | STM32Cube HAL | PWR
* Device | STM32Cube HAL | RCC

1. Depending on project needs, select other packs under Device | STM32Cube HAL, such as

* TIM (for timers) --(depends on DMA)
* LTDC (for LCD TFT), etc.
* I2C and SPI (Since STM32f429i\_DISCO board has some features that need use I2C and SPI ---stm32f429i\_discovery.c)

1. Click OK to close the Manage Run-Time Environment window

## 

## Figure 3: Manage Run-Time Environment

## In uVision 5’s project window, change the “Source Group 1” to “src”. Then right click it and choose “ Add New Item to Group ‘src”. In the opened dialog window, select “User Code Template”, then in the right part the window, double click the “Device” and then select “STM32Cube Framework : Classic | ‘main’ module for STM32Cube”

Users can rename the project name from “Target 1” to another name, like “lab1”.

Users can also right click the source group “src” to add exiting files instead of creating a new source file.

## 

## Figure4: Add user code to project

*Would suggest that the project (explorer) is organized in the following way:*

* *In the project folder, create three subfolders: src, inc, and stm32f429i\_discovery.* 
  1. *Src(to save source files): main.c, stm32f4xx\_hal\_msp.c(this file contains callback …..) and stm32f4xx\_it.c (should have system\_stm32f4xx.c and startup\_stm32f429xx.s too, but these two file has been in the “Device” source group)*
  2. *Inc(to save header files): main.h, stm32f4xx\_it.h.*
  3. *Stm32f4xx\_discovery: copy the files: stm32f429i\_discovery.c, and other files such as : stm32f429i\_discovery\_eeprom.c, stm32f429i-discovery\_lcd.c , stm32f429i-discovery\_gyroscope.c, stm32f429i-discovery\_io.c… from (Keil ROOT)\…..\Drivers\BSP\STM32F29I-Discovery folder.*

*stm32f4xx\_hal\_msp.c :*

This file contains the MSP (Mcu Specific Pack?) initialization and de-initialization (main routine and callbacks) of the peripheral used in the user application.

*stm32f4xx\_hal\_conf.h*

This file allows the user to customize the HAL drivers for a specific application.

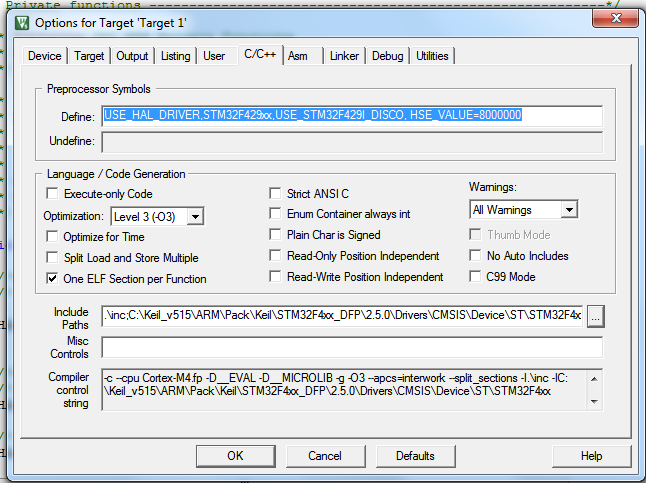
It is not mandatory to modify this configuration. The application can use the default configuration without any modification. (Most times, Users do not need to modify it)

* *In the projects, create 3 corresponding source groups for the above 3 folders. And add files to these groups.*
* *If create project from scratch, the files: system\_stm32f4xx.c, startup\_stm32f429xx.s and stm32f4xx\_hal\_conf.h are created and included in the “Device” source file group. SO, THESE FILES DO NOT need to be included in other folders, otherwise will have compiling errors.*

1. Configure the “Option for Target…”:

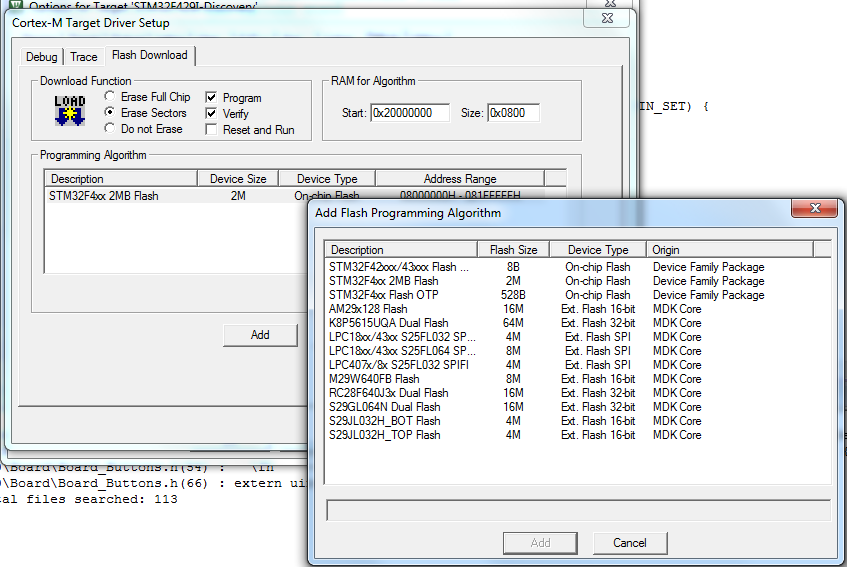
* In the Target tab: check “Use MicroLIB” (sometimes the size of the project image exceeds the maximum allowed for the version of the linker if do not check to use MicroLIB)
* In C/C++ tab:
  + define: “**USE\_HAL\_DRIVER,STM32F429xx,USE\_STM32F429I\_DISCO,HSE\_VALUE=8000000**” ---without quote mark when input.
  + Change Optimization level to Level 3
  + Include Paths:

Customize the include paths for the project. This depends on how the project is organized and where the packs or library files are located.



## Figure 5: Target Driver Setup

* In Debug tab: choose “STLink Debugger” instead of “ULINK2/ME Cortex Debugger” and click button “Settings” beside it. In the opened window “ Cortex-M Target Driver Setup”, set “Port” as “SW” instead of “JTAG”, then click tab “Flash Download” and then make sure the “STM32F4xx 2MB Flash” on-chip flash is the Programming Algorithm.(*Remember to change “Port” form JTAG to SW. otherwise, when download program to board, there will be error message: (“Unknown Target Connected” )*

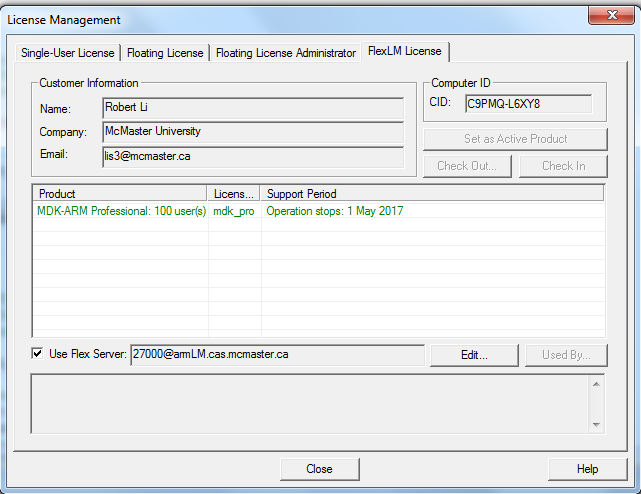


## Figure 6: Target Driver Setup (Also remember here to add, “STM32F4xx 2MB flash”)

1. **License Set up for Keil uVision**

Using the Cube HAL library will make some lab projects exceed the file size limit for the free version of KEIL uVersion5, even if the MicroLIB is used when compiling the project. Our department has a license for licensed uVision5 which is capable of handling those lab projects whose binary size is larger. For some projects you need to set the license for Keil uVision.

To set up license for KEIL uVision5, go to File | License Management…., then in the License Management window, click the FlexLM License tab, then check the checkbox of “Use Flex Server” and input “27000@armLM.cas.mcmaster.ca” in the corresponding input box. See Figure 7.



## Figure7: License Setup for Keil uVision required for some lab projects