STM32 Discovery Board Projects

Notes:

- This tutorial was written for the STM32F411E-Discovery board and its STM32F411VET microcontroller. If using a different STM Discovery or Nucleo board, basic project setup is the same, except for the microcontroller to be selected.
- 2. This document assumes that the ARM Keil MDK (Version 5.xx) has been installed, along with the STM32F4xx device family pack and ST-Link USB driver. Refer to the tutorial "Installing MDK-ARM" on the course web page.

ARM Keil Microcontroller Development Kit (MDK)

As illustrated in Figure 1, MDK-ARM Version 5.xx (*µVision*) comprises a set of core functions:

- Integrated Development Environment (IDE) with project manager and editor to create projects
- *ARM C/C++ compiler, assembler, and linker* to build projects (any combination of languages)
- Debugger to interactively debug projects on the target board or simulator
- Pack installer

One or more software packs must be installed, each of which provides resources (system/startup code, device drivers, etc.) for a specific family of microcontrollers. The appropriate software pack must be installed for each microcontroller family to be used.

• The required pack for these projects is *Keil::STM32F4xx_DFP*, which supports the *STM32F411VET* microcontroller on the *STM32F411E-Discovery* board.

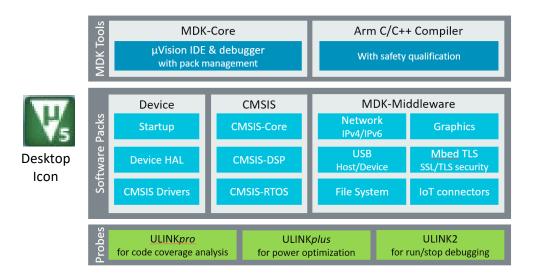


Figure 1. MDK-ARM V5.xx (µVision) comprises the MDK Core and one or more Software Packs



ARM University Program Copyright © ARM Ltd 2013

Creating a New Project for the Discovery/Nucleo Board

 Open μVision and from the menu bar select: Project → New μVision Project. In the Create New Project Window (Figure 1), navigate to the folder in which you want to create the project (create the folder if necessary), enter a project name (ex. Project1), and click Save.

Create New Project							
Computer > ho	omes (\\kudzu.eng.auburn.edu) (H:) 🔸	ELEC2220 • Program1	✓ Search Program1	Q			
Organize 🔻 New folder		Project Folder		!≕ ▼ 🔞			
Documents	Name	Date modified Type	s Size				
🎍 Downloads] ee5250		No items match your search.					
ELEC2220							
📔 ELEC3040_3050							
퉬 elec4200							
🐌 ELEC5250_6250							
🐌 ELEC5260_6260							
Favorites							
\mu grad.www							
File <u>n</u> ame: Program1	Project File			-			
Save as type: Project Files (*.uvproj; *.uvprojx)			•			
) Hide Folders			Save	Cancel			

It is recommended that a separate folder be created for each project.

Figure 1. Creating a new µVision project "Lab1"

2. In the Select Device for Target window (Figure 2), select microcontroller STM32F411VET.

Device	ce for Target 'Target 1'	>
	Software Packs	
Vendor: Device: Toolset: Search:	STMicroelectronics STM32F411VETx ARM	
	Des <u>cription:</u>	
	STM32F411CC STM32F411CC STM32F411CC STM32F411CC STM32F411RC STM32F411RC STM32F411RC STM32F411RC STM32F411VC STM32F411VC STM32F411VE STM32F411	
	OK Cancel Help	

Figure 2. Select the STM32F411VET microcontroller.



ARM University Program Copyright © ARM Ltd 2013

The Architecture for the Digital World[®]

3. In the *Manage Run-Time Environment* window (Figure 3), you can add microcontroller header and startup files, device drivers, and other software pack components to the project by checking the corresponding boxes. (These can also be added later, if desired.)

ELEC 2220 - Introductory assembly language programs:

CMSIS Core and Startup programs are unnecessary for testing simple assembly language programs. **Click OK, without checking any boxes,** to omit these from the project. The assembled program code will be placed beginning at the first byte of program memory.

OTHER PROJECTS:

For other projects: check the boxes for **CMSIS** \rightarrow **CORE** and **Device** \rightarrow **Startup**. (CMSIS = Cortex *Microcontroller Software Interface Standard*) as shown in Figure 3, plus boxes for any drivers or other software components you wish to include in the project. The assembled/compiled code for your programs will be placed <u>after</u> that of the device startup program in program memory. The startup program is required for initializing the stack, clocks, and some other system resources, after which it jumps to the first instruction of your application program.

oftware Component	Sel.	Variant	Version	Description
🛷 Board Support		32F469IDISCOVERY ~	1.0.0	STMicroelectronics 32F469IDISCOVERY Kit
🔶 CMSIS				Cortex Microcontroller Software Interface Components
CORE	v		5.5.0	CMSIS-CORE for Cortex-M, SC000, SC300, ARMv8-M, ARMv8.1-M
DSP		Source	1.9.0-dev	CMSIS-DSP Library for Cortex-M, SC000, and SC300
🖉 NN Lib			3.0.0	CMSIS-NN Neural Network Library
🗈 💠 RTOS (API)			1.0.0	CMSIS-RTOS API for Cortex-M, SC000, and SC300
🗄 🚸 RTOS2 (API)			2.1.3	CMSIS-RTOS API for Cortex-M, SC000, and SC300
💠 CMSIS Driver				Unified Device Drivers compliant to CMSIS-Driver Specifications
💠 Compiler		ARM Compiler	1.6.0	Compiler Extensions for ARM Compiler 5 and ARM Compiler 6
💠 Device				Startup, System Setup
Startup	v		2.6.3	System Startup for STMicroelectronics STM32F4 Series
💿 🚸 STM32Cube Framework (API)			1.1.0	STM32Cube Framework
🗄 💠 STM32Cube HAL				STM32F4xx Hardware Abstraction Layer (HAL) Drivers
			i	
lidation Output		Description		

Figure 3. Include the CMSIS CORE and Startup files in the project. (Unless testing simple assembly language programs in ELEC 2220.)



ARM University Program Copyright © ARM Ltd 2013

- 4. Create and add source file(s) to the project in the **Project** pane of the μ *Vision* window (Figure 4). You may do this in either or both of the following ways.
 - Right-click on *Source Group 1*, select *Add New Item to Group 'Source Group 1'*, select a file type (*Asm File (.s) or C File (.c)*), enter the file name (Example: *Program1.s*), and click *Add*. This creates a blank file in the editor pane, into which you can type or paste your source program.
 - For files created outside of μVision, right-click on Source Group 1, select Add Existing Files to Group 'Source Group 1', locate and select file(s) to be added to the project and click Add.
 - Click *Close* after you've created/added all desired source files.

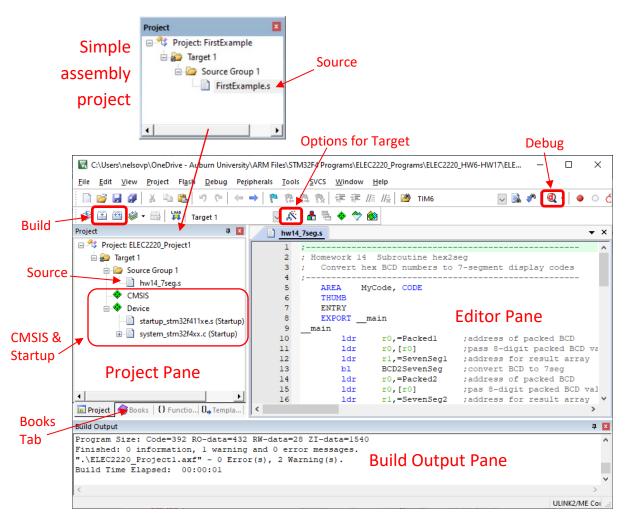


Figure 4. Add source file(s) to Source Group 1 in the Project pane, and build the project. (Note project pane above for simple assembly projects, without startup/CMSIS/)



The Architecture for the Digital World[®]

ARM University Program Copyright © ARM Ltd 2013 Before going to Step 5, click the "Options for Target" button, and select the Linker tab as shown below. Check the box "Use Memory Layout from Target Dialog" (if not already checked) and click OK to close the window.

avica Tarr	et Output Listing User C/C++ (AC	G) Arm Linker De	bug Utilities	
✓ Use Mer	nory Layout from Target Dialog RW Sections Position Independent RO Sections Position Independent Search Standard Libraries ort 'might fail' Conditions as Errors	<u>X</u> /O Base: <u>R</u> /O Base: R/ <u>W</u> Base <u>d</u> isable Warnings:	Dx08000000 Dx20000000	
Scatter File			.	Edit 1
				Luit
<u>M</u> isc controls				

- 5. After creating and saving source files, the project must be "built" by compiling/assembling the source files, correcting any syntax errors as necessary, and then linking the files into one downloadable program. This is performed by:
 - Click on the *Build* icon above the project pane (or from the menu bar select *Project* → *Build Project*). This will recompile/assemble only source file(s) that have changed.
 - OR: Click on the *Rebuild* icon above the project pane (or from the menu bar select *Project* → *Rebuild all target files*). This will compile/assemble all source files.
 - Read the messages in the Build Project pane. You must see the message

ARM University Program

Copyright © ARM Ltd 2013

".\MyProject.axf" - Error(s)", k Warning(s)

before you can debug the program. (*MyProject.axf* is the object file (binary machine code) to be programmed into the microcontroller. If there are any errors, this file will NOT be created, and attempting to enter the debugger will fail.)

- If there are one or more errors/warnings, scroll up through the messages in the Build Project pane to identify the reason for each and where they are located in your program. Any errors must be corrected before the object file will be created.
- **NOTE:** The project must be rebuilt if you change any device, target, compiler, assembler, or linker options.



- 6. There are several options for testing and debugging projects with the μ *Vision* debug tools.
 - a) Load the program into the **simulator** for testing, rather than the physical microcontroller (*See* to Step 7 below.)
 - b) Download the program to the **Discovery board**, program it into the on-chip flash memory of the microcontroller and execute the program in the target hardware. *(See Step 8 below.)*
 - c) Execute a program that was **previously programmed** into the flash memory of the target microcontroller. (*See Step 9 below.*)
- 7. To download the compiled program to the <u>simulated</u> microcontroller's memory and initiate a debug session with the simulator, set up the project Debug options as follows.
 - From the menu bar, select Project → Options for Target 'Target 1' (or click on the Options for Target icon next to the target name in the menu bar.)
 - Select the *Debug* tab (Figure 5) and check the *Use <u>Simulator</u>* button.
 - If you wish to use a debug initialization file (described in Step 10 below), enter the file name in the *Initialization File* box below the *Use Simulator* button.
 - Click OK to close the Options for Target window.

Options for Target 'Target 1'							
Device Target Output Listing User C/C++ Asm Linker Debug Utilities							
 Use Simulator I Limit Speed to Real-Time 	Settings	○ <u>U</u> se: ST-Link	k Debugger				
✓ Load Application at Startup Initialization File: debug.ini	Run to main()	✓ Load Applicat Initialization File:	ion at Startup 🔽 Run to main()				
Restore Debug Session Setting Breakpoints Watch Windows & Perfore Memory Display	Toolbox nance Analyzer	Restore Debug Session Settings Image: Breakpoints Image: Toolbox Image: Breakpoints Image:					
CPU DLL: Parameter: SARMCM3.DLL -MPU -REMAI	2	Driver DLL: SARMCM3.DLL	Parameter: -MPU				
Dialog DLL: Parameter: DCM.DLL PCM4]	Dialog DLL:	Parameter: pCM4				
OK Cancel Defaults Help							

Figure 5. Target Options Debug tab. "Use Simulator" selected, and debug initialization file specified.



ARM University Program Copyright © ARM Ltd 2013

- 8. To download the compiled program to the <u>Discovery board</u>, program it into the microcontroller's flash memory, and initiate a debug session, set up the project Debug options as follows.
 - From the menu bar, select Project → Options for Target 'Target 1' (or click on the Options for Target icon next to the target name in the menu bar.)
 - Select the *Debug* tab (Figure 6), check the *Use* button and select *ST-Link Debugger*.
 - If you wish to use a debug initialization file (described in Step 10 below), enter the file name in the *Initialization File* box below the *Use ST-Link Debugger* button.

V Options for Target 'Target 1'						
Device Target Output Listing User C/C++ Asm	Linker Debug Utilities					
C Use Simulator Settings	ⓒ Use: ST-Link Debugger Settings					
☑ Limit Speed to Real-Time						
✓ Load Application at Startup ✓ Run to main()	Load Application at Startup Run to main()					
Initialization File:	Initialization File:					
Edit	debug.ini Edit					
Restore Debug Session Settings	Restore Debug Session Settings					
I Breakpoints I Toolbox	✓ Breakpoints ✓ Toolbox					
Watch Windows & Performance Analyzer	Watch Windows					
Memory Display System Viewer	Memory Display System Viewer					
CPU DLL: Parameter:	Driver DLL: Parameter:					
SARMCM3.DLL -MPU -REMAP	SARMCM3.DLL -MPU					
Dialog DLL: Parameter:	Dialog DLL: Parameter:					
DCM.DLL PCM4	TCM.DLL -pCM4					
, ,	1					
OK Car	ncel Defaults Help					

Figure 6. Target Options Debug tab. "Use ST-Link Debugger" selected.

(Continue on next page for setting up the ST-Link Debugger)



ARM University Program Copyright © ARM Ltd 2013

 Click on Settings next to ST-Link Debugger to produce the Cortex-M Target Driver Setup Window (Figure 7), and select Port SW (Single-Wire debug) under Target Com.
 If Serial Number is blank, install/reinstall the ST-Link USB driver on your computer.

Debug Trace Flash Download Pack Debug Adapter Unit: ST=LINK/V2 Shareable ST-Link	SW De	vice IDCODE 0x2BA01477	Device Name ARM CoreSight SW	-DP (ARM Core	Move
Serial Number: 30FF6E065647373721571043 Version: HW: V2 FW: V2J37S0 Check version on start Target Com Port: SW Req: 10 MHz Selected: 1.800 MHz		omatic Detection nual Configuration	ID CODE: Device Name: Dodate IR Ien: IR IEN: IEN: IR IEN: IR IEN: IEN: IR IEN: IR IEN: IEN: IR IEN: IR IEN:	AF	Down e; 0
Debug Connect & Reset Options Connect: Normal Reset: Autodetect Reset after Connect Stop after Reset	_	Cache Optio	<u>2</u> ode <u>□ V</u> erify	l Options Code Download load to <u>∏</u> ash	

Figure 7. Single-Wire (SW) Debug Adapter port selected.

 Click on the *Flash Download* tab of the Cortex-M Target Driver Setup Window (Figure 8) and under *Programming Algorithm*, click the *ADD* button and select *STM32F4xx 512kB Flash*.

Then you can close the setup and options windows.

Cortex-M Target Driver Setup				×
Debug Trace Flash Download	Pack			
Download Function Erase Full Chip Frase Sectors Do not Erase	 ✓ Program ✓ Verify ✓ Reset and F 		Ngorithm x20000000 Size: 0x00001000]
Programming Algorithm	Device Size	Device Type	Address Range	-
STM32F4xx 512kB Flash	512k	On-chip Flash	08000000H - 0807FFFH	-
		Start: [Size:	-
	Add	Remove		
			OK Canc	el <u>A</u> pply

Figure 8. STM32F4xx flash programming algorithm selected



ARM University Program Copyright © ARM Ltd 2013

- 9. To initiate a debug session for a program previously programmed into the microcontroller's flash memory on the Discovery board, set up the project Debug options as in step 8 above, but on the Debug tab (Figure 6) uncheck the button *Load Application at Startup*, located beneath the *Use ST-Link Debugger* button. This will cause the debugger to be entered without downloading the program, thereby using the program currently in the microcontroller's flash memory.
- 10. A **debug initialization file** is useful to set up any desired initial register and memory values and other debug options when the debugger is first opened. This is a plain text file that contains one or more debugger commands; these commands may also be entered in the Debugger's Command Window (see Figure 9).

The following is a sample initialization file (*debug.ini* in Figures 5 and 6)

```
xPSR=0x01000000 //Set T bit of xPSR Register to Thumb Mode
PC=0x08000000 //Set starting program counter address
E INT 0x20000000 = 5,23,-100,200,201,-200,0,1000,2000
```

- The first line initializes the T (Thumb) bit to 1 in the Processor Status Register (xPSR) to force the processor into Thumb mode. The Cortex-M4 processor in the STM32F411 microcontroller supports only Thumb mode instructions, and its T bit is hard-wired to 1. However, the simulator initializes xPSR to all 0's, thereby setting T=0, which selects ARM mode. Attempting to execute a Thumb instruction in ARM mode will trigger an error exception. Therefore, the user must initialize T=1 at the start of simulation when using the simulator.
- The second line initializes the program counter (PC) to the address of the first instruction to be executed. This will normally be the first byte of flash memory at address 0x08000000. If using the device startup code, the initial PC will be loaded from a vector table and therefore this debug command should be removed.
- The third line enters a sequence of 32-bit data values (separated by commas) into consecutive memory words, beginning at address 0x20000000 (the first byte SRAM). The "INT" indicates that the data are to be encoded and written as 32-bit values. Type "CHAR" would indicate 8-bit data. This E command is useful for entering test data for small programs.
- 11. To initiate a debug session, click the **Debug** icon in the μ Vision window (Figure 4), or select **Debug** \rightarrow Start/Stop Debug Session. Depending on how you set up the debug options, this will open the Debug Window (Figure 9) after:
 - downloading the project code to the simulator memory (if set up via Step 7),
 - downloading the project code to the target board and programming it into the microcontroller's flash memory (if set up via Step 8),
 - skipping the download (if set up via Step 10)

ARM University Program

Copyright © ARM Ltd 2013

At this point, you may run/stop/step the program, set breakpoints, select watch variables, etc. Click on the **Debug** icon to stop the debug session and return to the μ Vision IDE.



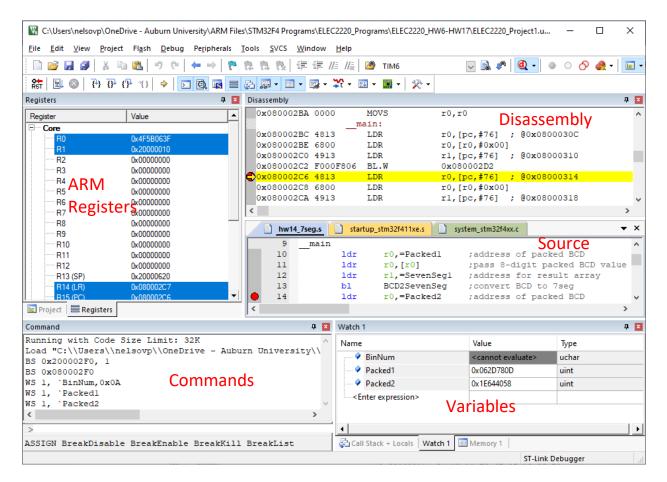


Figure 9. MDK uVision Debug Window.

Use of the debugger is described in the document **Debugging Projects with MDK-ARM**, available on the course web page.

For further information, refer to "*Keil uVision MDK - uVision Lab for the STM32F4 Discovery Board*" (<u>http://www.keil.com/appnotes/docs/apnt_230.asp</u>): App Note 230, Section 20 (Creating your own project from scratch)



ARM University Program Copyright © ARM Ltd 2013