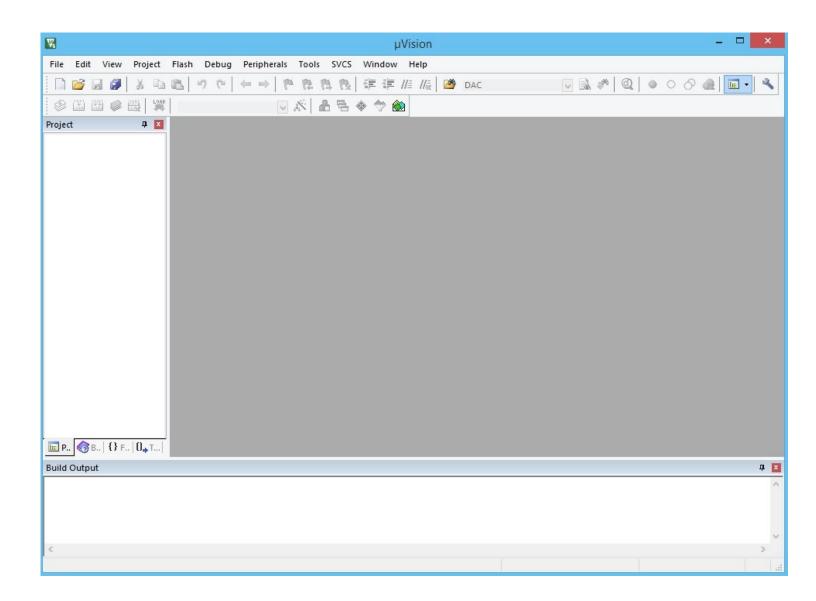
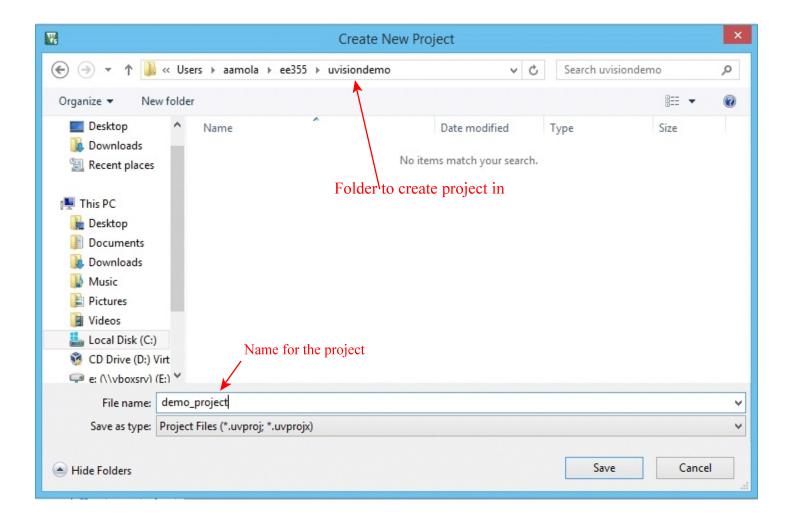
#### Keil uVision Quick Start Guide

The goal of this guide is to illustrate the steps needed to create a new uVision project, add an existing source file to the project, build (i.e. compile) the project, and download the new program to a NXP Freedom development board.

- 0) Create a folder (directory) in which a new project will be created.
- 1) With a browser navigate to the class webpage and download the flashled2019.c program source code file into the directory you created for your project (it is at the bottom of the class web page in the Downloads section)
- 2) Start the Keil uVision program. A screen like (or similar) to that below should appear. (The screen shots shown in this document were made with uVision version 5.24)



- 3) Click **Project** on the top task bar. A drop-down box will open.
  - Select New uVision Project and the Create New Project window will open. See below.
  - Navigate to the folder you wish to put the project in.
  - Enter a project name.
  - Click Save.



After clicking Save a new window will open to allow choice of a specific microcontroller that you wish to target with the design (program) you are creating.

Next page.

This is the first of several windows used to specify the specific brand and model of microcontroller that you wish to use for this project.

:	Select Device for Target 'Target 1'	×
Device Software Packs Vendor: <unknown> Device: <unknown> Toolset: <unknown></unknown></unknown></unknown>	<b>_</b>	
Search: ARM NXP Texas Instruments	Description:	^
	<	>
	OK Cancel	Help

Clock the + box left of NXP

Select Device for Target 'Target 1'
Device Software Packs Vendor: <unknown> Device: <unknown> Toolset: <unknown> Search:</unknown></unknown></unknown>
Description:
ARM NXP KLxx Series Texas Instruments
< > > ×
OK Cancel Help

We are interested in using the KLxx series of microcontrollers

Expand the KLxx series options by clicking on the + box to its left.

Select Device for Target 'Target 1'	×
Device	
Software Packs	
Vendor: <unknown></unknown>	
Device: <unknown></unknown>	
Toolset: <unknown></unknown>	
Search:	
Description:	
ARM NXP KLxx Series KL0x KL0x KL1x KL1x KL2x KL2x KL3x KL4x KL4x KL4x KL8x Texas Instruments	~
OK Cancel	Help

Expand the selection of KL2x microcontrollers by clicking on the + box to its left.

Once a specific device (microcontroller chip) is selected, a high level summary of its characteristics is displayed. For class you need to select the MKL25Z128xxx4 device pointed to with the red arrow in the picture below:

Select [	Device for Target 'Target 1'	x
Device Software Packs Vendor: NXP Device: MKL24Z64xox4 Toolset: ARM Search:	Description:	
ARM NXP KLxx Series KL0x KL1x KL1x KL2x MKL24Z32xxx4 (dep) MKL24Z5128xxx4 (dep) MKL25Z128xxx4 (dep)	Kinetis L series MCUs, based on the ARM® Cortex®-M0+ core, offer ultra-low power and small size with scalable integration and ease-of- use. The Kinetis KL2x is an ultra-low-power MCU family that adds a full- speed USB 2.0 On-the-Go (OTG) controller or a full-speed crystal-less USB 2.0 device controller in addition to the Kinetis KL1x series. The Kinetis KL2x MCU family is compatible with Kinetis K20 MCUs (based on ARM® Cortex®-M4), and with all other Kinetis KL1x, KL2x, KL3x, and KL4x series MCUs, providing a migration path to lower and higher performance and feature integration.	~
	OK Cancel Help	

Click OK and the highlighted device will be the target for this project.

oftware Component	Sel.	Variant		Version	Description
w 🚸 CMSIS					Cortex Microcontroller Software Interface Components
🕬 🚸 CMSIS Driver					Unified Device Drivers compliant to CMSIS-Driver Specifications
🕬 Compiler		ARM Compile	r	1.4.0	Compiler Extensions for ARM Compiler 5 and ARM Compiler 6
🕸 Device					Startup, System Setup
🐳 🕸 File System		MDK-Pro	~	6.10.0	File Access on various storage devices
🕂 💠 Graphics		MDK-Pro	~	5.46.5	User Interface on graphical LCD displays
🕸 Network		MDK-Pro	~	7.8.0	IPv4/IPv6 Networking using Ethernet or Serial protocols
RTOS		FreeRTOS		10.0.1	FreeRTOS Real Time Kernel
🚸 USB		MDK-Pro	~	6.12.4	USB Communication with various device classes
alidation Output		Descripti	on		<b>b</b>

# Next the libraries that will be used in this design are chosen. Only two are needed.

### Click on CMSIS and Device.

oftware Component	Sel.	Variant		Version	Description
🚸 CMSIS					Cortex Microcontroller Software Interface Components
CORE				5.1.1	CMSIS-CORE for Cortex-M, SC000, SC300, ARMv8-M
DSP				1.5.2	CMSIS-DSP Library for Cortex-M, SC000, and SC300
🔷 NN Lib				1.0.0	CMSIS-NN Neural Network Library
🗄 🚸 RTOS (API)				1.0.0	CMSIS-RTOS API for Cortex-M, SC000, and SC300
🗄 🚸 RTOS2 (API)				2.1.2	CMSIS-RTOS API for Cortex-M, SC000, and SC300
🚸 CMSIS Driver					Unified Device Drivers compliant to CMSIS-Driver Specifications
🚸 Compiler		ARM Compile	r	1.4.0	Compiler Extensions for ARM Compiler 5 and ARM Compiler 6
🚸 Device					Startup, System Setup
Startup				1.0.0	System Startup for NXP MKL24Z4 Devices
🚸 File System		MDK-Pro	~	6.10.0	File Access on various storage devices
🚸 Graphics		MDK-Pro	~	5.46.5	User Interface on graphical LCD displays
🚸 Network		MDK-Pro	~	7.8.0	IPv4/IPv6 Networking using Ethernet or Serial protocols
🚸 RTOS		FreeRTOS		10.0.1	FreeRTOS Real Time Kernel
🚸 USB		MDK-Pro	×	6.12.4	USB Communication with various device classes
alidation Output		Descripti			

Note the select boxes and click those for CORE under CMSIS and Startup under Device. The other libraries are not needed.

The uVision screen should now look something like this:

C:\Users\aamola\ee355\uvisiondemo\demo_project.uvprojx - µVision - C	×
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help	
🗋 😂 🗟 🖇 🛝 🖏 🖏 🤟 🗢 🔶 隆 隆 隆 🎼 🎼 🎼 🏙 DAC 🛛 🔽 🐨 🐼 🧶 🔴 🖉 🍓	- 3
🔗 🕮 🕮 🥔 👯   Target 1 🕢 🐼 📥 🖷 🧇 🐡 🎒	
Project 🕂 🗜 🔟	
Project: demo_projec   Image: Target 1	
Build Output	<b>Д</b> 🗙
	~
<	>
ULINK2/ME Cortex Debugger	

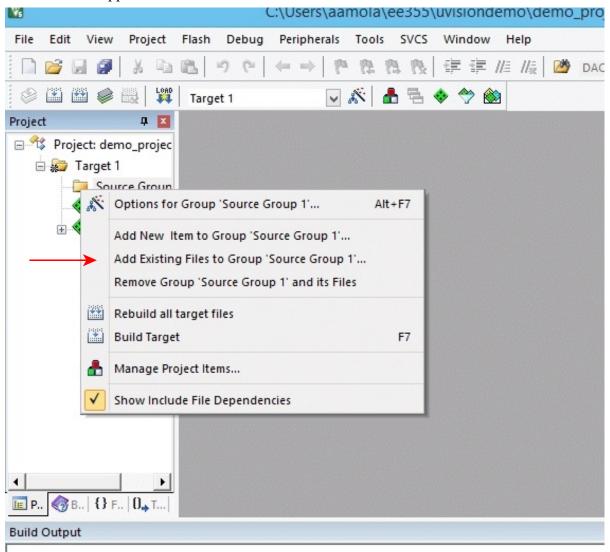
Note that the left panel titled Project has in it icons and the name of the project created above. Also, there is an icon and the text Target 1.

Expand the Target item (click on the + box to its left). A Source Group icon will appear.

Right click on the folder symbol to the left of Source Group text.

This will open a menu box as shown on the next page.

#### Select Add Exiting Files to Group



A new box will appear: ----->

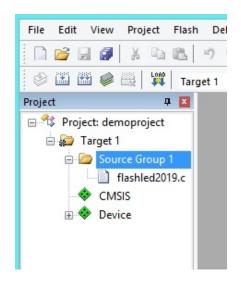
Click on a source file.

Click Add

Click Close

1	Add Files to Group 'Source	Group 1'	×
Look in: 🚺	uvisiondemo 💌	← 🗈 💣 📰 ▾	
Name	A	Date modified	Ту
listings		1/19/2020 8:38 PM	Fi
Objects		1/19/2020 8:38 PM	Fi
📗 RTE		1/19/2020 8:38 PM	Fi
flashled.	2019.c	1/19/2020 8:26 PM	С
<			>
File name:	flashled2019	Add	
Files of type:	C Source file (*.c)	✓ Close	

The left (Project) panel should look something like this after adding a source file: ------



If you double click on the source file flashled2019.c it will open the file in a text editor like this:

W	C:\Users\aamola\ee355\uvisiondemo\demoproject.uvprojx - µVision – 🗖 🗙
File Edit View Project Flash	Debug Peripherals Tools SVCS Window Help
🗋 🗋 🛃 🕼 🕺 🕹 🛍 🗎	り ♡   ← →   🍢 陰 陰 良   律 津 //ε //ε 🖄 🖉 🔽 🔽 🔍 🔍 🔶
😵 🍱 🏙 🥔 🔜 🔤 Targe	et 1 🔽 🔊 📩 📥 🖷 🧇 🐡 🎯
Project 📮 🗵	flashled2019.c
Project: demoproject Target 1 Source Group 1 I I I I I I I I I I I I I I I I I I I	<pre>1 // **********************************</pre>
► ► ► ► ► ► ► ► ► ► ► ► ► ► ► ► ► ► ►	22 ; 22 ; <

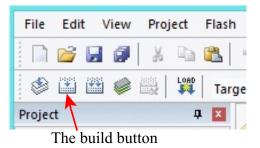
If there are additional source or header files you can open several and there will be a tab for each in the edit window area allowing one at a time to be open for editing.

If you are starting a new project without an existing source file (or files) then instead of adding existing file(s) you would select Add New Item to the source files and an empty file will be created. You then can use the editor and write a program.

#### **BUILDING A PROJECT**

Once a program is added to the project or is written it needs bo be built. During the build process a system startup C-file, an assembly file, some library files, and header files will be added to the project. During the build process source files will be compiled and linked together to create a binary file to be downloaded to the microcontroller.

A build can be started from the Project tab on the upper tool bar or by clicking an icon on a lower tool bar:



During and at completion of the build, information will be displayed in the Build Output window at the bottom of the uVision window. Here is an example after building the flashled program:

```
Build Output

*** Using Compiler 'V5.06 update 5 (build 528)', folder: 'C:\Keil

Build target 'Target 1'

compiling flashled2019.c...

assembling startup_MKL2524.s...

compiling system_MKL2524.c...

linking...

Program Size: Code=704 RO-data=336 RW-data=0 ZI-data=352

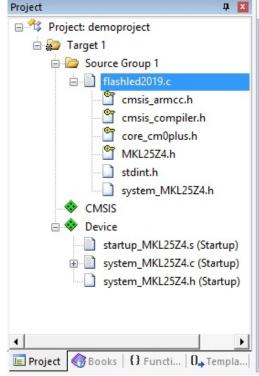
".\Objects\demoproject.axf" - 0 Error(s), 0 Warning(s).

Build Time Elapsed: 00:00:09

Project demoproject

Project: demoproject
```

After building you can see names of additional files added to the project during the build process. Click on the + boxes left of flashled2019.c and the Device item. ----->



The next step is to download to the target processor.

But before download will work a little setup must be done.

Click the **Flash** tab on the top tool bar to get a drop-down menu:



Click on Configure Flash Tools

A window similar to that below will appear. Tabs across the top select various screens for configuring options. If the Debug tab is not highlighted then click on it (the Utilities tab is active as shown below):

Options for Target 'Target 1'	×
Device Target Output Listing User C/C++ Asm Linker Debug Utilities	
Configure Flash Menu Command	
Use Target Driver for Flash Programming	
Use Debug Driver Settings 🔽 Update Target before Debugging	
Init File: Edit	
Command: Arguments: Run Independent	
Configure Image File Processing (FCARM):     Add Output File to Group:	
Source Group 1	
Image Files Root Folder:	
OK Cancel Defaults Help	

Here is the Debug screen with default options showing:

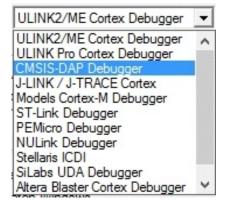
Options for Ta	arget 'Target 1'
Device       Target       Output       Listing       User       C/C++       Asm         O       Use       Simulator       with restrictions       Settings         Imit Speed to Real-Time	Linker Debug Utilities
✓ Load Application at Startup ✓ Run to main() Initialization File:	✓ Load Application at Startup     ✓ Run to main() Initialization File:
Restore Debug Session Settings Breakpoints Toolbox Watch Windows & Performance Analyzer Memory Display System Viewer CPU DLL: Parameter:	Restore Debug Session Settings     Iv Breakpoints     Iv Toolbox     Iv Watch Windows     Iv Memory Display     Iv System Viewer  Driver DLL:     Parameter:
SARMCM3.DLL	SARMCM3.DLL
Dialog DLL: Parameter: DARMCM1.DLL -pCM0+	Dialog DLL: Parameter: TARMCM1.DLL -pCM0+
Manage Component Vie	ewer Description Files
OK Car	ncel Defaults Help

Click the down arrow to the right of the little window with the text

ULINK2/ME Cortex Debugger

A drop-down menu will open with these options:

# Select CMSIS-DAP Debugger



The Debug window should now look like this:

	Options for Ta	arget 'Target 1'	
Device Target	Output Listing User C/C++ Asm	Linker Debug Utilities	
○ Use Simulator		Use: CMSIS-DAP Debugger     Settings	
Load Applicat	ion at Startup 🔽 Run to main()	✓ Load Application at Startup ✓ Run to main() Initialization File:	
	Edit	Edit	
Restore Debug	Session Settings	Restore Debug Session Settings	
Breakpoin	ts 🔽 Toolbox	I Breakpoints I Toolbox	
	indows & Performance Analyzer	Watch Windows	
	isplay Viewer	Memory Display Viswer	
CPU DLL: Parameter:		Driver DLL: Parameter:	
SARMCM3.DLL		SARMCM3.DLL	
Dialog DLL:	Parameter:	Dialog DLL: Parameter:	
DARMCM1.DLL -pCM0+		TARMCM1.DLL pCM0+	
	Manage Component Vi	iewer Description Files	

The default options are ok for all the other fields on this screen as well as screens opened under the other tabs.

Click OK

### NOW THE EXCITING PART - DOWNLOADING

Plug in a USB cable between the SDA connector on the Freedom board and a USB port on the computer running uVision.

Click on the LOAD icon.

Download should start.

F	ile	Edi	t \	/iew	Pro	ject	Flash
*****		2	P	ø	¥		
	٢					LOAD	Targe
Pr	ojec	t				7 4	L X

(remember that the OpenSDA program has to installed on the Freedom board before download will work)

As download proceeds there will be messages displayed in the Output window at the bottom of the screen. If errors occur they will be displayed. An example successful download is this:

Build Output

```
Load "C:\\Users\\aamola\\ee355\\uvisiondemo\\Objects\\demoproject.axf"
Erase Done.
Programming Done.
Verify OK.
Flash Load finished at 21:21:51
```

To start running the program downloaded onto the Freedom board press and release the pushbutton that is on the Freedom board.

Thats it. Of course there are lots of features of the uVision program including debugging that can and should be explored as needed.