

# Course Software Installation

ECE 3436 – Microprocessor Systems

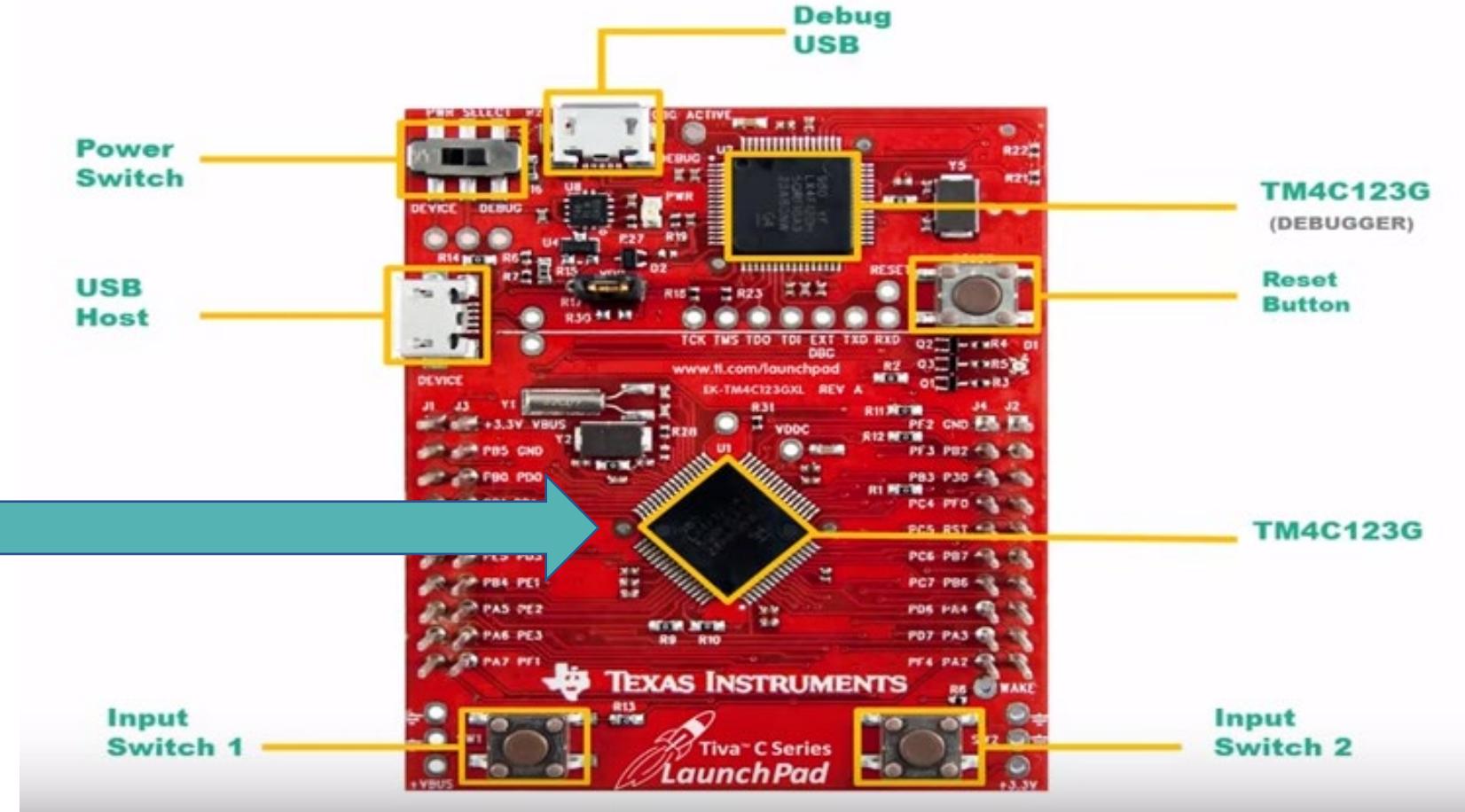
Dr. Diana de la Rosa-Pohl

*\* Some software has been provided by Dr. John Valvano and his team in the  
Department of Electrical and Computer Engineering at The University of Texas at Austin.*

# The Hardware

- The TI LaunchPad

Tiva® C Series MCU  
TM4C123GH6PM



# Development Tools for Tiva C Series MCUs

	 mentor embedded	 IAR SYSTEMS	 ARM KEIL <small>An ARM Company</small>	 Code Composer Studio
<b>Eval Kit License</b>	30-day full function. Upgradeable	32KB code size limited. Upgradeable	32KB code size limited. Upgradeable	Full function. Onboard emulation limited
<b>Compiler</b>	GNU C/C++	IAR C/C++	RealView C/C++	TI C/C++
<b>Debugger / IDE</b>	gdb / Eclipse	C-SPY / Embedded Workbench	μVision	CCS/Eclipse-based suite
<b>Full Upgrade</b>	99 USD personal edition / 2800 USD full support	2700 USD	MDK-Basic (256 KB) = €2000 (2895 USD)	445 USD
<b>JTAG Debugger</b>		J-Link, 299 USD	U-Link, 199 USD	XDS100, 79 USD

# Step 1: Download & Install Keil MDK V5

- download the free version of Keil MDK V5
  - MDK = microcontroller development kit
  - <https://www.keil.com/demo/eval/arm.htm>

# Download Keil

- Enter your information
- The device you are using is the TM4C123
- You must accept the privacy policy.

arm KEIL

Products Download Events Support Videos

Search Keil... + Go

Product Information

Software & Hardware Products

- Arm Development Tools
- C166 Development Tools
- C51 Development Tools
- C251 Development Tools
- Debug Adapters
- Evaluation Boards

Product Brochures

Newsletters

Device Database®

Device List

Compliance Testing

ISO/ANSI Compliance

Validation and Verification

Distributors

Overview

Home / Product Downloads

## MDK-ARM

MDK-ARM Version 5.37  
Version 5.37

Complete the following form to download the Keil software development tools.

### Enter Your Contact Information Below

First Name: Diana

Last Name: de la Rosa-Pohl

E-mail: ddelarosa2@uh.edu

Company: University of Houston

Job Title: Instructional Associate Professor

Country/Region: United States

State/Province: Texas

Phone:

Send me e-mail when there is a new update.  
**NOTICE:**  
If you select this check box, you will receive an e-mail message from Keil whenever a new update is available. If you don't wish to receive an e-mail notification, don't check this box.

Which device are you using?  
(eg, STM32)

TM4C123

Arm will process your information in accordance with the Evaluation section of our [Privacy Policy](#).

Please keep me updated on products, services and other relevant offerings from Arm. You can change your mind and unsubscribe at any time.

Submit Reset

## Download Keil

- The installation file is ~900 MB

The screenshot shows the arm KEIL website with a blue header. The main content area is titled "MDK-ARM" and discusses Version 5.37. It includes a list of installation instructions and a download link for "MDK537.EXE". A red arrow points to the download link.

Product Information

- Software & Hardware Products
  - Arm Development Tools
  - C166 Development Tools
  - C51 Development Tools
  - C251 Development Tools
  - Debug Adapters
  - Evaluation Boards
- Product Brochures
- Newsletters

Device Database®

- Device List

Compliance Testing

- ISO/ANSI Compliance
- Validation and Verification

Distributors

- Overview

MDK-ARM

MDK-ARM Version 5.37  
Version 5.37

- Review the [hardware requirements](#) before installing this software.
- Note the [limitations of the evaluation tools](#).
- [Further installation instructions for MDK5](#)

(MD5:9a3e824a57eb370555826d53f2f6056d)

To install the MDK-ARM Software...

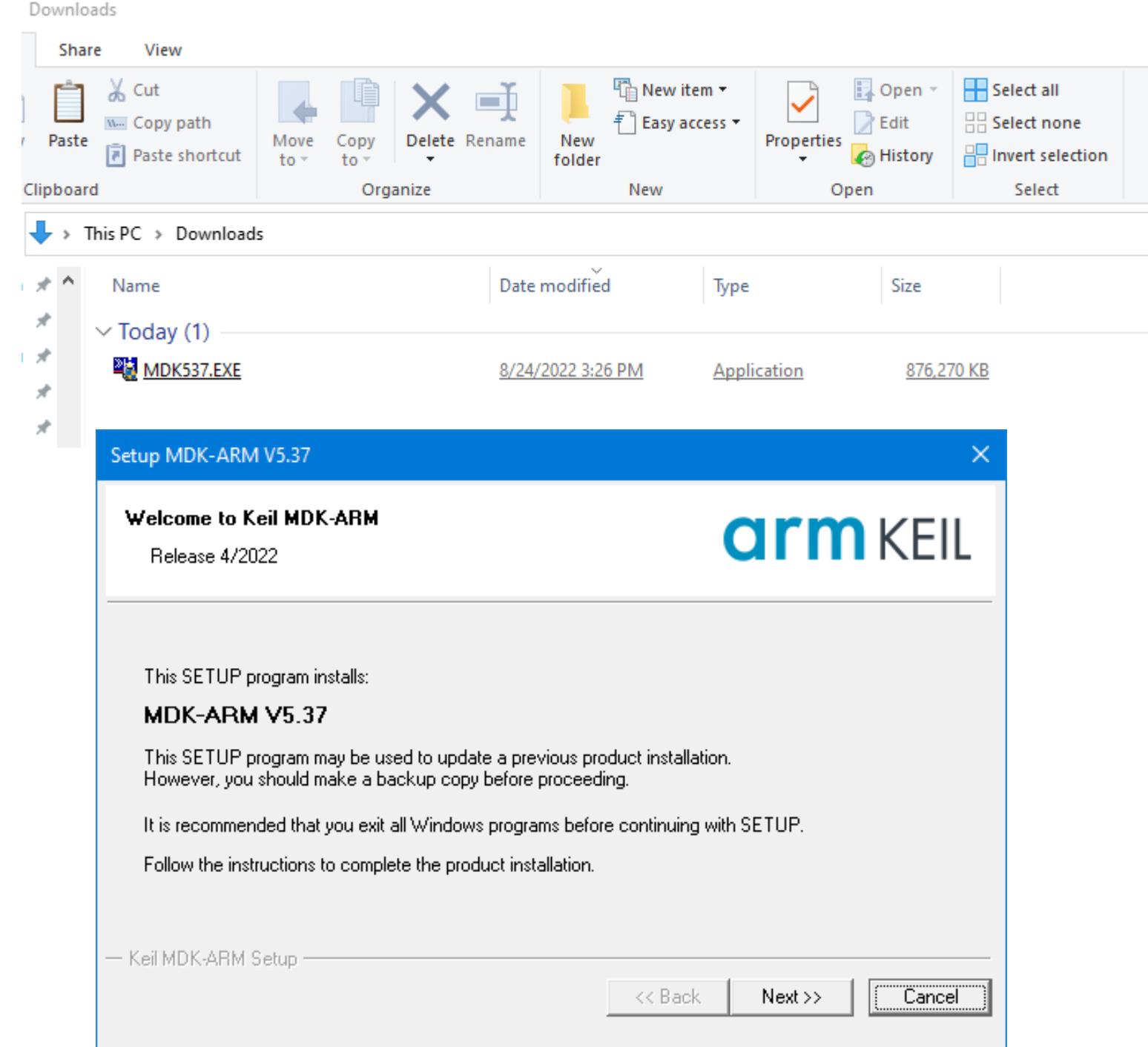
- Right-click on **MDK537.EXE** and save it to your computer.
- PDF files may be opened with Acrobat Reader.
- ZIP files may be opened with PKZIP or WINZIP.

**MDK537.EXE** (876,270K)  
Tuesday, May 3, 2022

- If you are evaluating the tools, be sure to [request a quote](#) for the full version of the tools.

# Install Keil

- run the MDK537.exe file



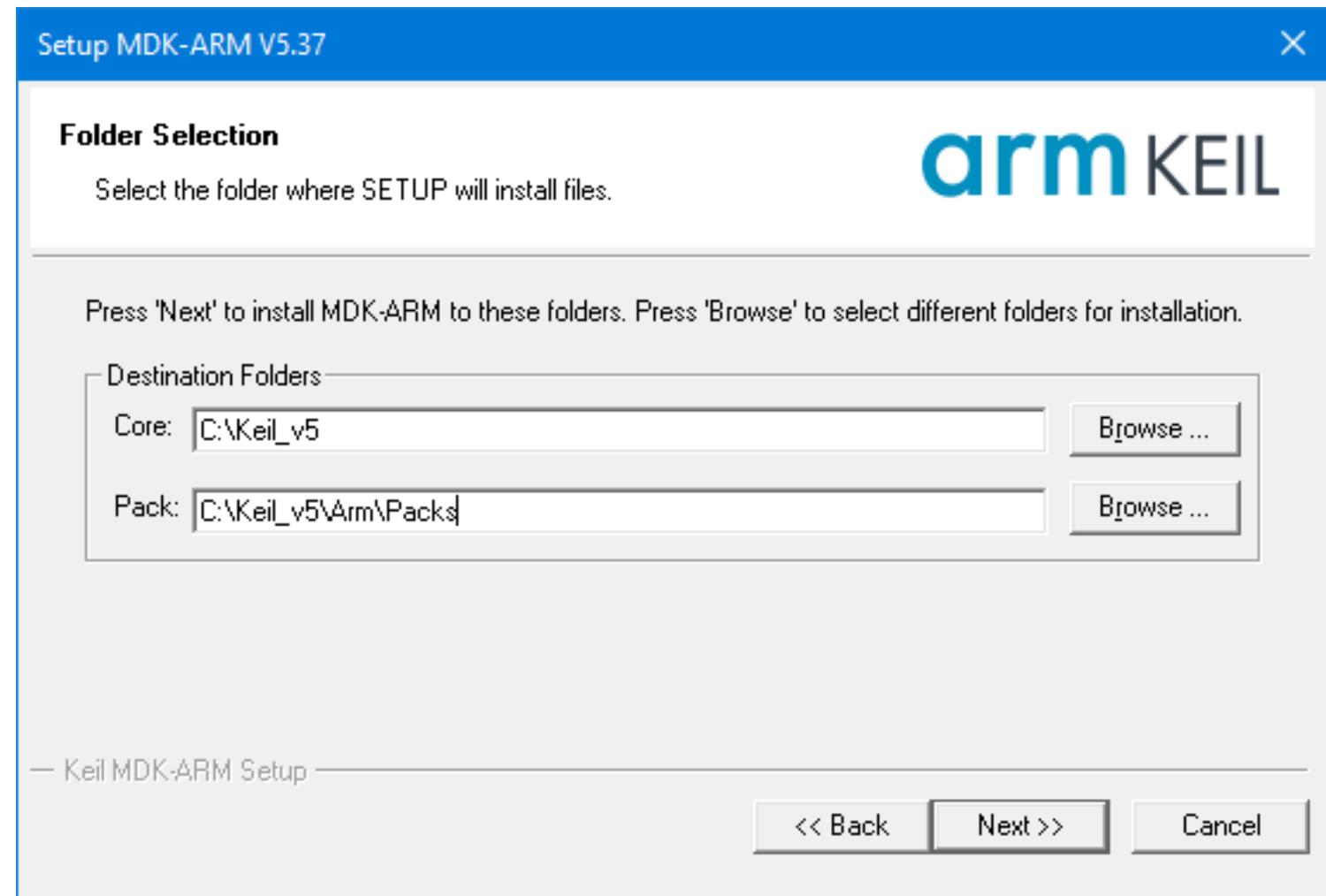
# Install Keil

- agree to the license agreement



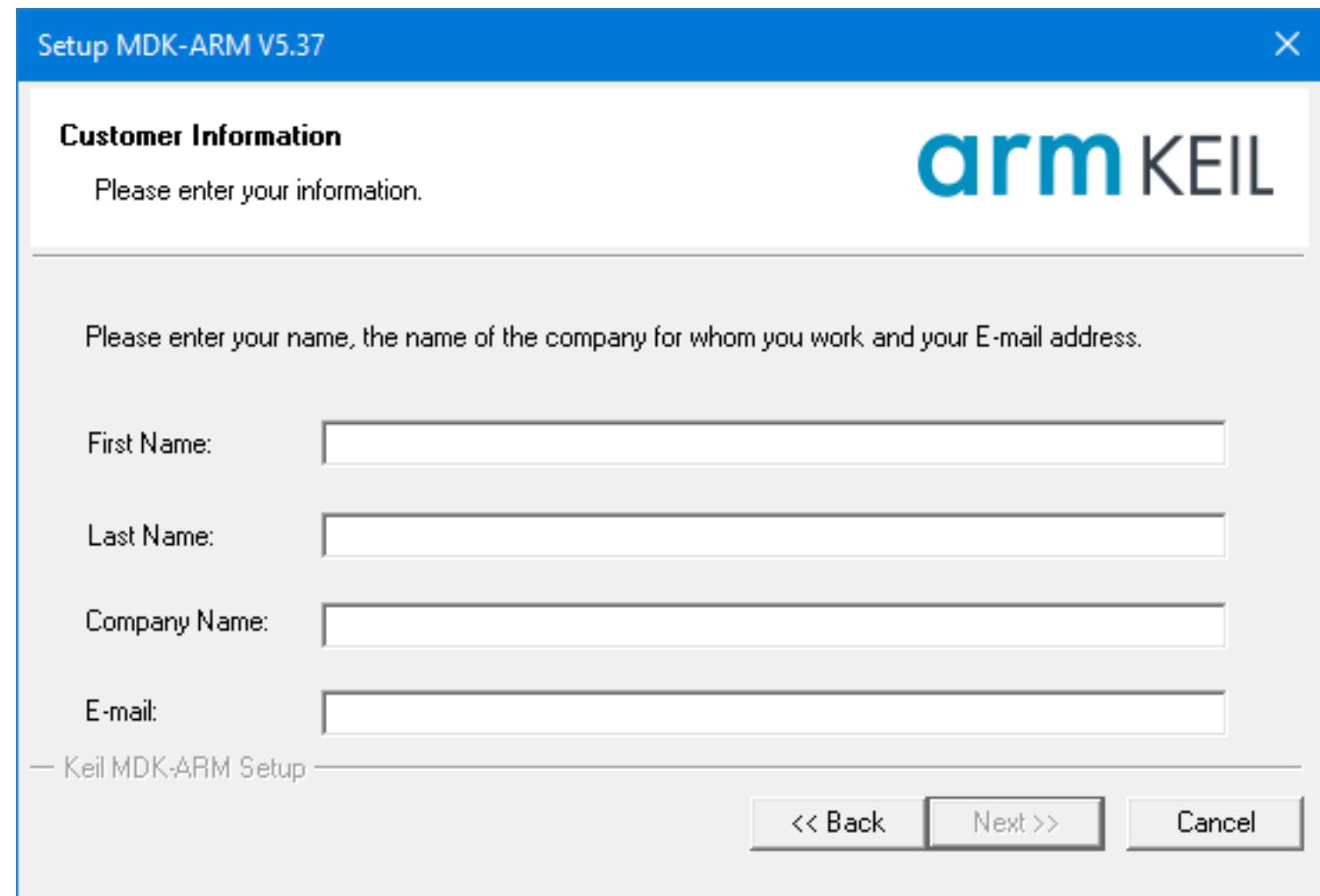
# Install Keil

- Choose where you want to install Keil



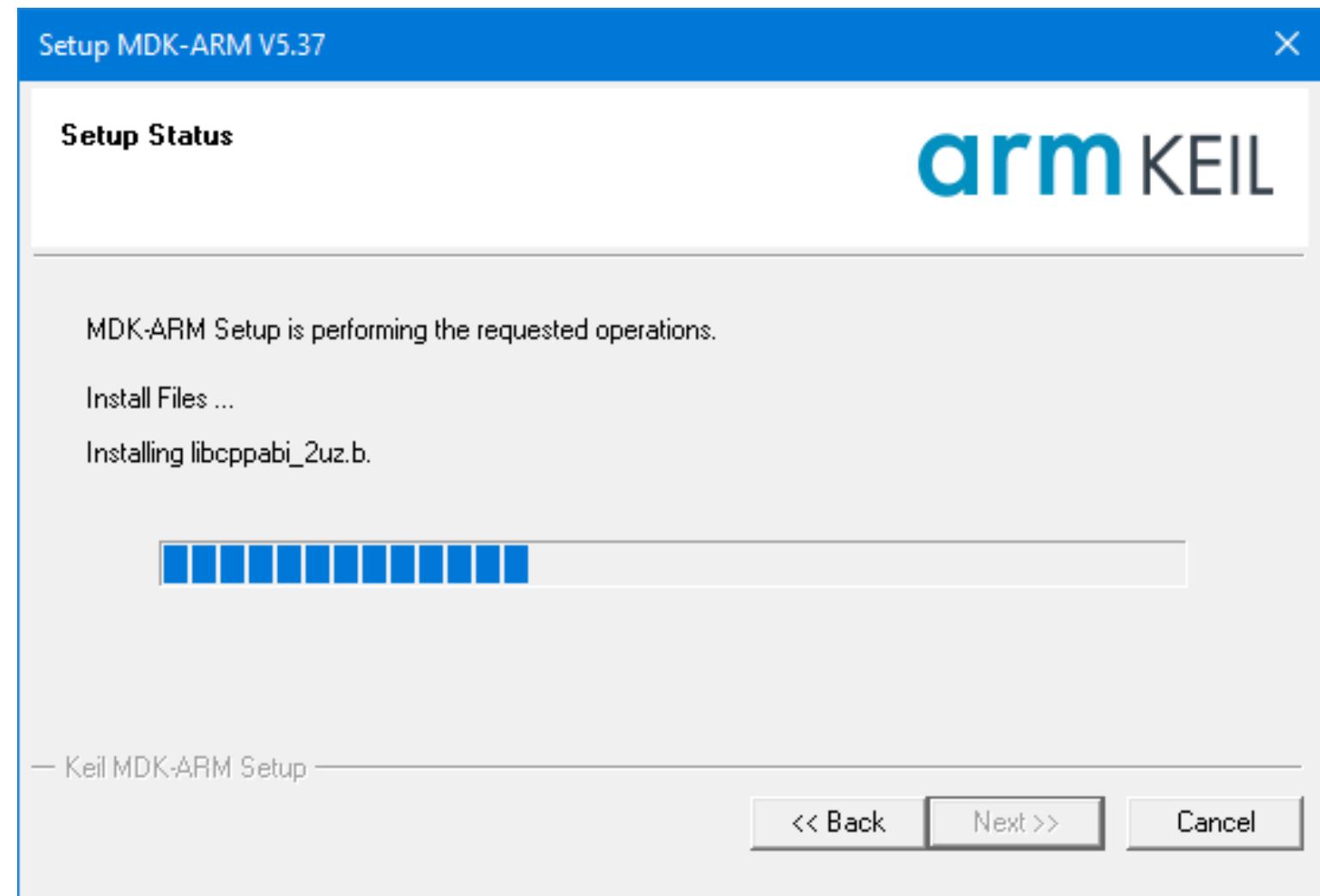
# Install Keil

- Enter YOUR information



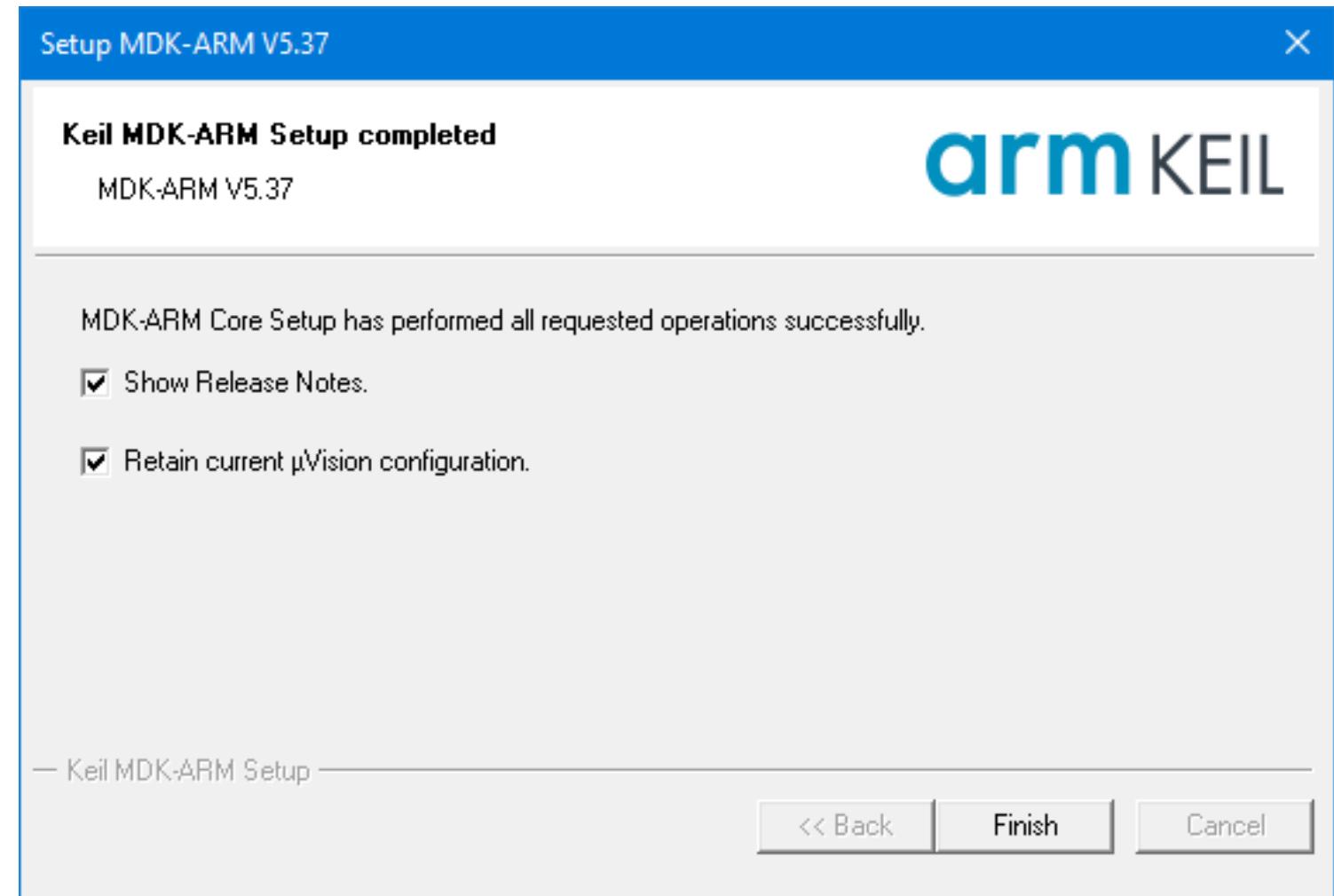
## Install Keil

- click Next and wait for all of the files to be installed
- this process will take a few minutes



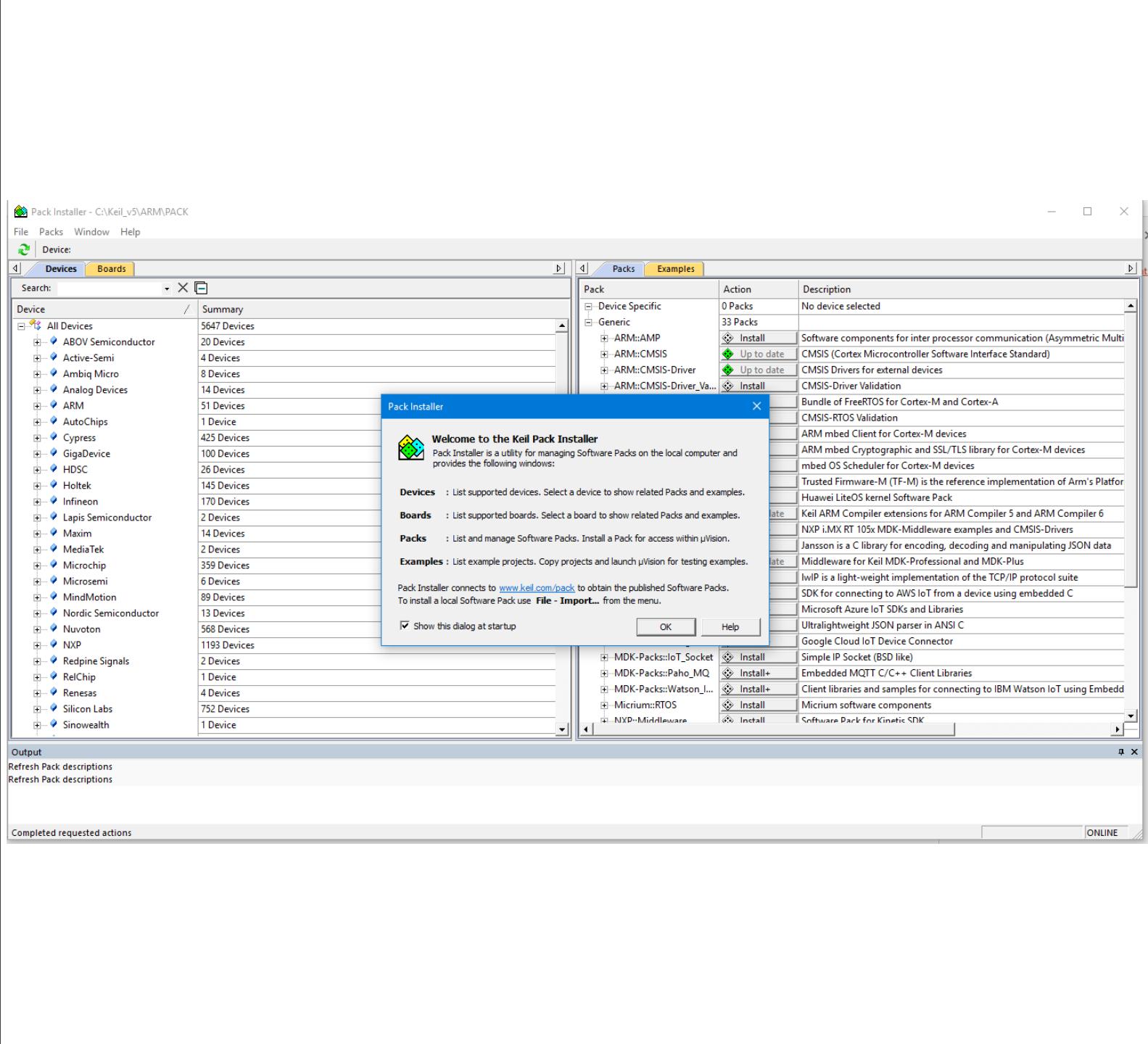
# Install Keil uVision

- Click “Finish”



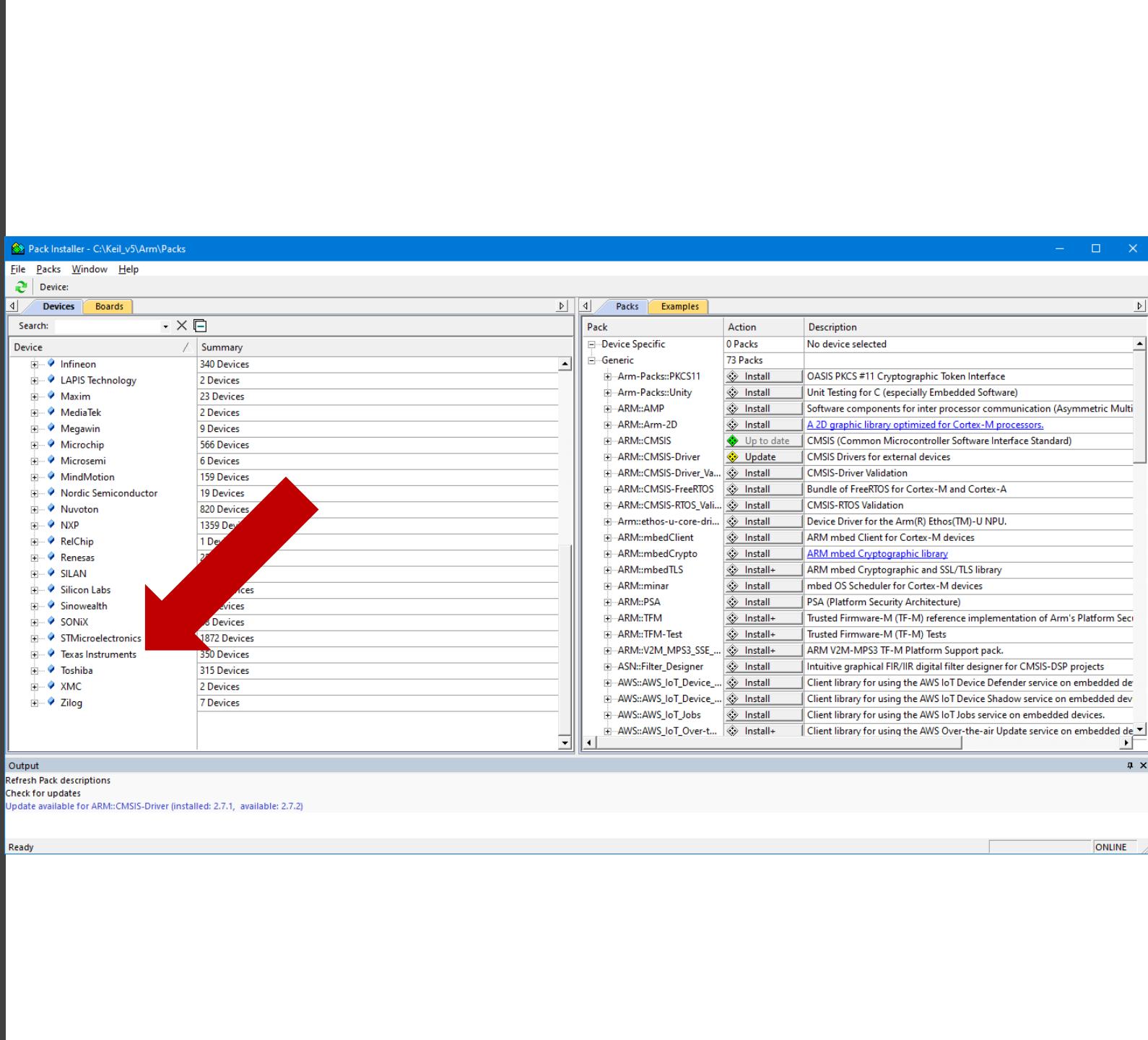
# Install Keil

- Click OK



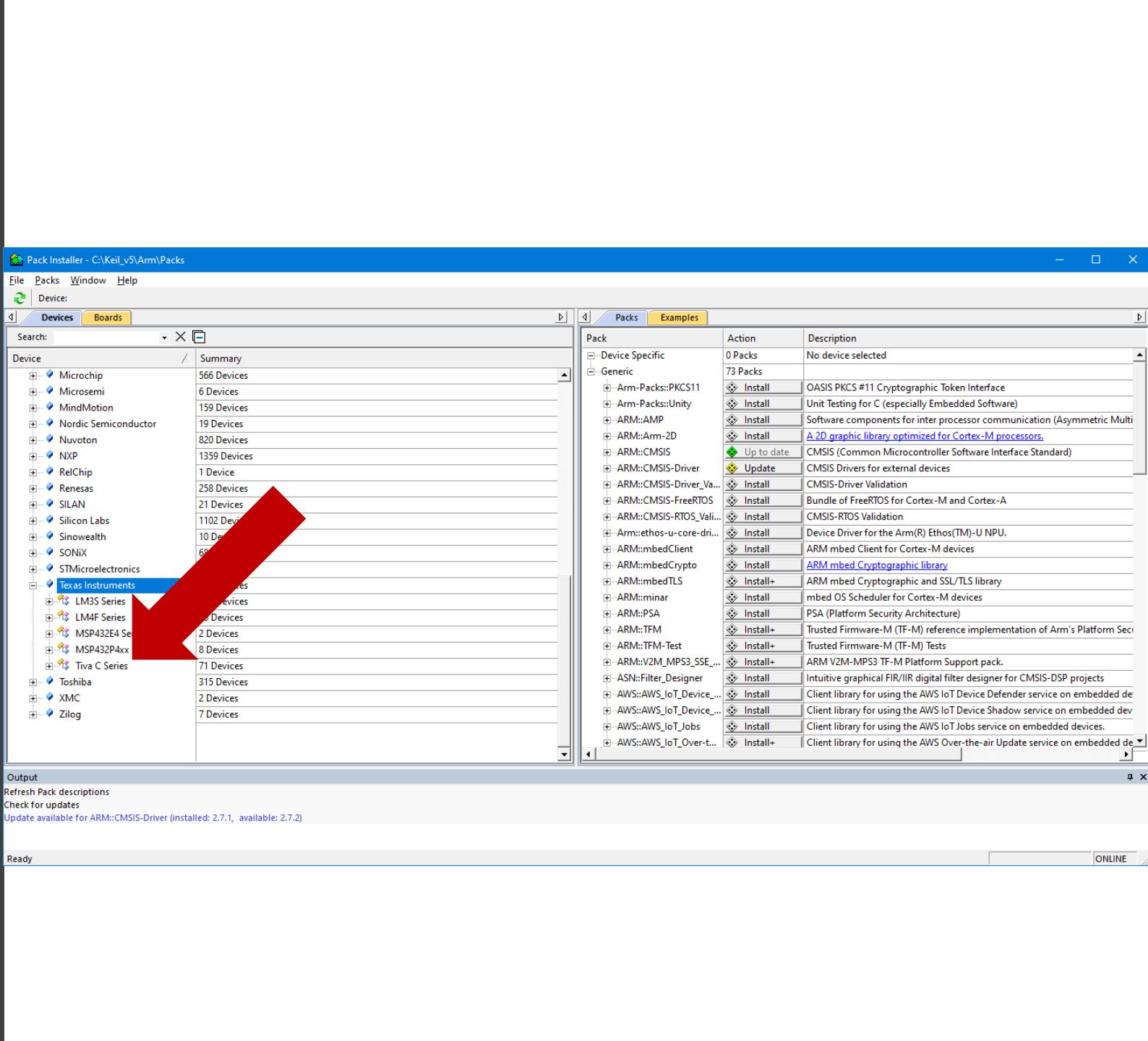
# Install Keil

- Expand the Texas Instruments options



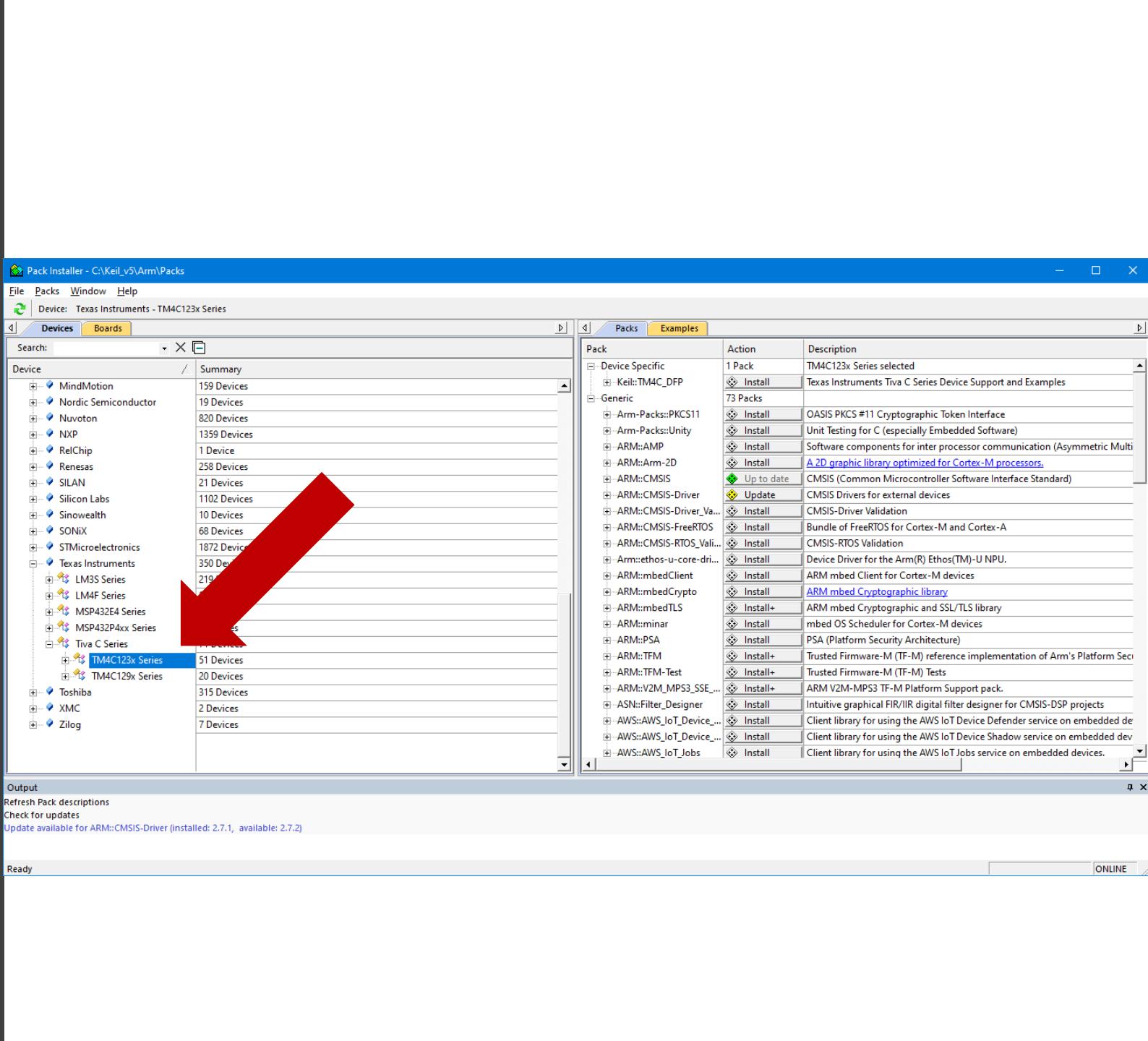
# Install Keil

- Expand the Tiva C Series options



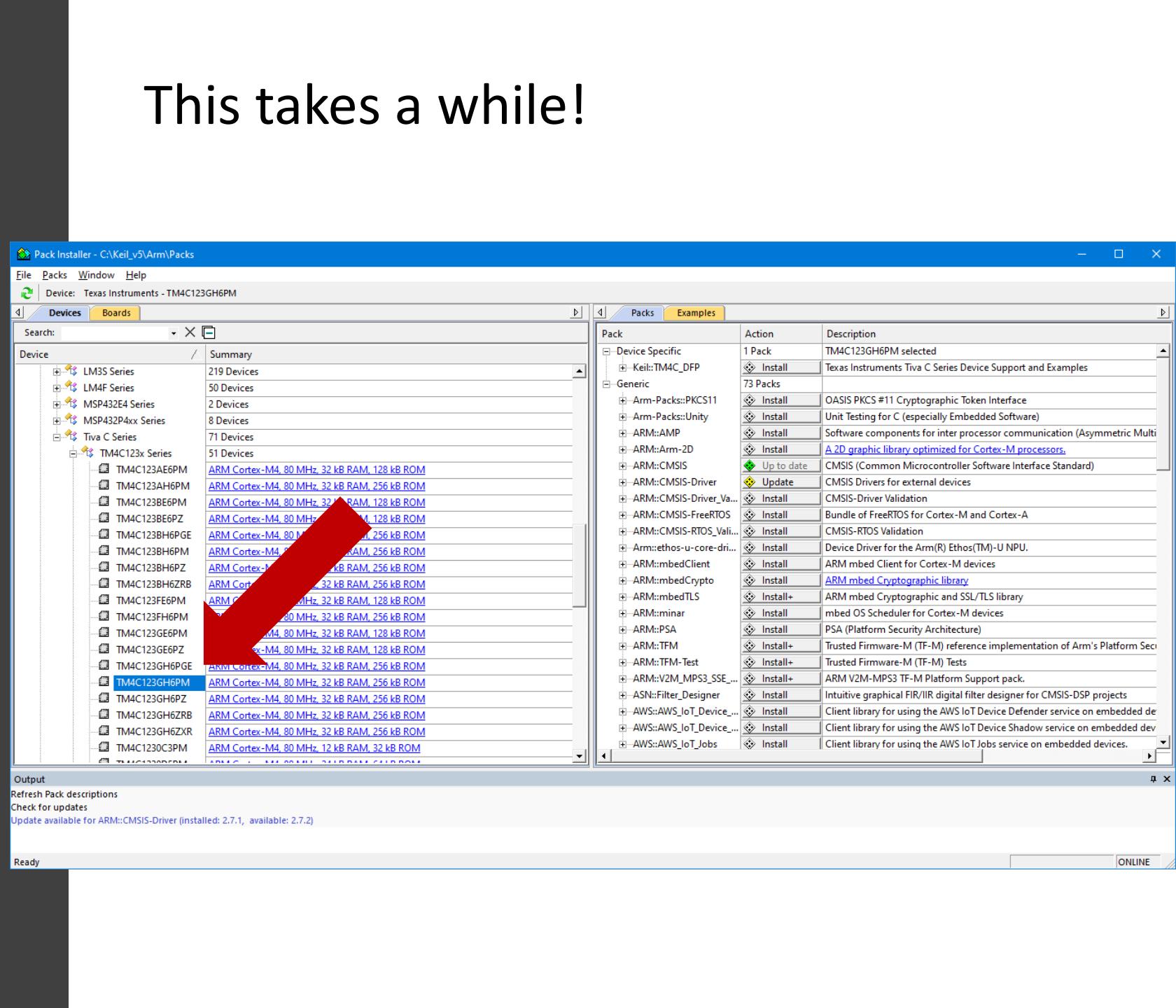
# Install Keil

- Expand the TM4C123 options



# Install Keil

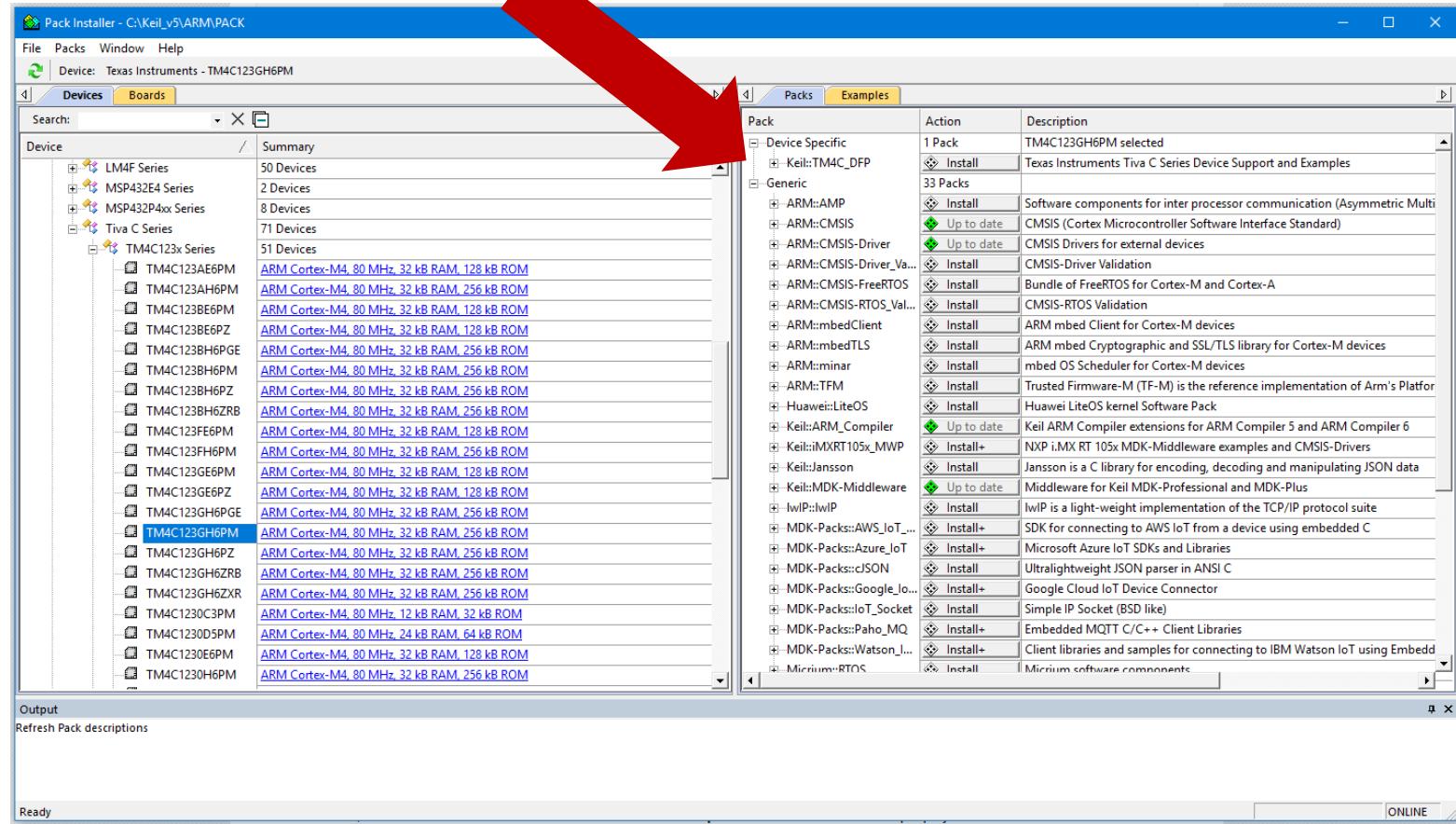
- Select the target device
  - TM4C123GH6PM
- Wait until the download finishes.



This takes a while!

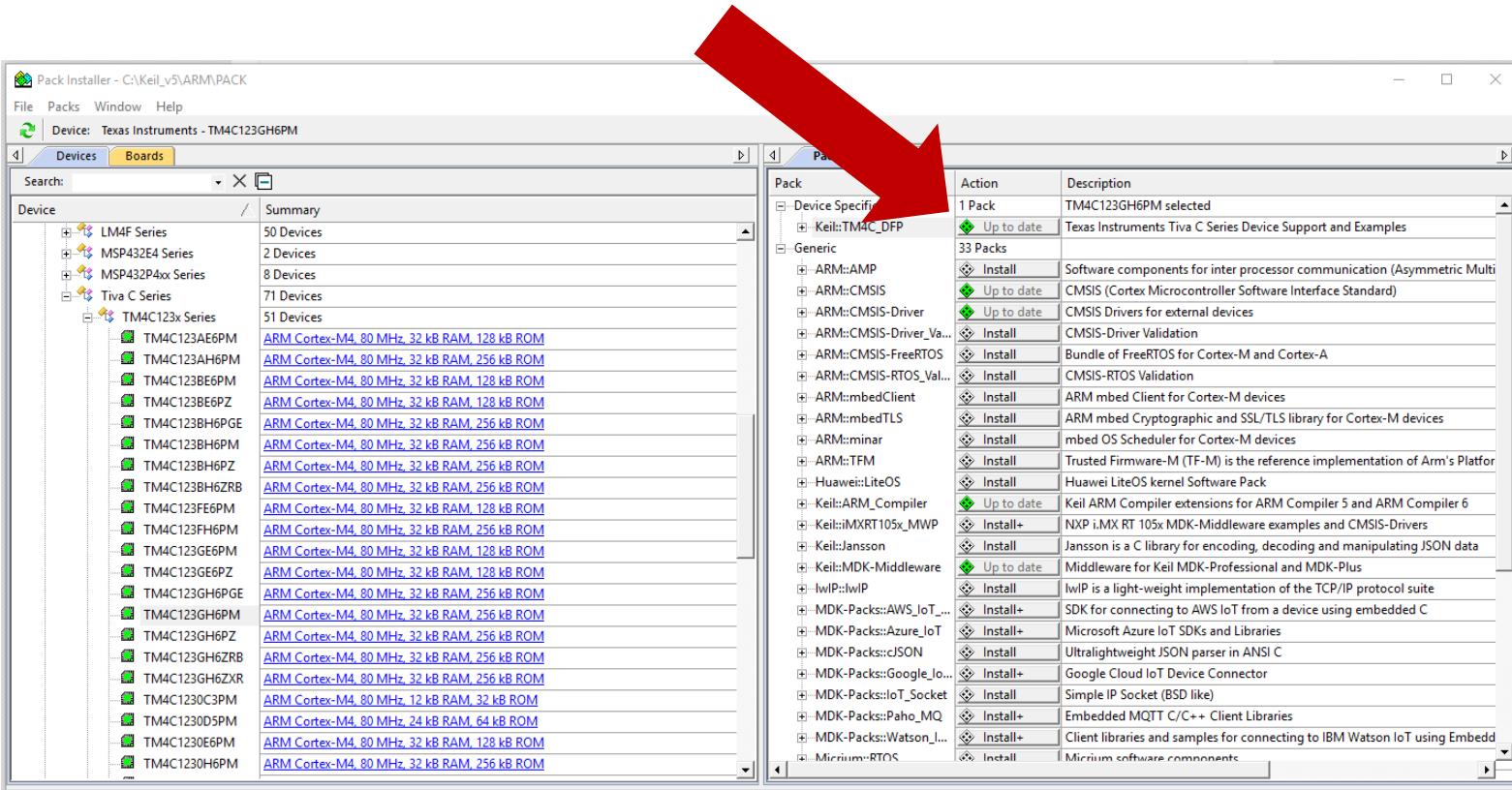
# Install Keil

- Install the Device Specific pack
  - Keil::TM4C\_DFP
- wait until the install completes (see the bar at the bottom)



# Install Keil

- wait until the install completes (see the bar at the bottom)
- You should see “Up to date” as in the image to the right
- You can now close this application



# μVision User's Guide

- Refer to the μVision User's Guide for more information.

<https://developer.arm.com/documentation/101407/latest>

The screenshot shows the μVision User's Guide page on the arm Developer website. The page title is "μVision User's Guide" and it specifies "Version 5.37". The left sidebar contains a "DOCUMENT TABLE OF CONTENTS" with chapters such as "About uVision", "User Interface", "Creating Applications", "Debugging", "Debug Commands", "Debug Functions", "Simulation", "Flash Programming", and "Dialogs". The main content area includes the title "μVision User's Guide", the version "Version 5.37", and a section titled "About this book" which describes the user guide's purpose and contents. A navigation bar at the top includes links for "IP Explorer", "Documentation", "Downloads", "Community", and "Support". A "Next Section" link and a five-star rating icon are also visible.

## Step 2: Download & Install the TI LaunchPad drivers

If you do not have your LaunchPad you can skip to Step 3 for now and come back to Step 2 later.

- download the LaunchPad drivers
- [http://www.ti.com/tool/stellaris\\_icdi\\_drivers](http://www.ti.com/tool/stellaris_icdi_drivers)

# Download Stellaris ICDI Drivers

- Click on the **Download** button for the current drivers
- As of 8/24/22, the filename is
  - **spmc016a.zip**

The screenshot shows the Texas Instruments website with a red header bar. The header includes the Texas Instruments logo, a search bar, and links for Login / Register, English, Ship to, and USD. Below the header, a navigation menu has links for Products, Applications, Design resources, Quality & reliability, Support & training, and About TI. A shopping cart icon is also in the top right. The main content area has a breadcrumb trail: Home / Design resources. The title "STELLARIS\_ICDI\_DRIVERS" is displayed above "Stellaris® ICDI Drivers". Below the title are links for Overview, Downloads, Technical documentation, Related design resources, and Support & training. A large yellow arrow points from the "Downloads" section down to the "SW-ICDI-DRIVERS – Stellaris® ICDI Drivers - Current" link, which is highlighted with a red "Download" button.

TEXAS INSTRUMENTS

Products Applications Design resources Quality & reliability Support & training About TI

Home / Design resources

## STELLARIS\_ICDI\_DRIVERS

### Stellaris® ICDI Drivers

Overview | Downloads | Technical documentation | Related design resources | Support & training

#### Overview

Tiva™ C Series evaluation and reference design kits provide an integrated In-Circuit Debug Interface (ICDI) which allows programming and debugging of the onboard C Series microcontroller. The ICDI can be used with the LM Flash Programmer as well as any of the Tiva-supported toolchains such as Texas Instruments' Code Composer Studio. Only JTAG is supported. To use the ICDI, follow the instructions in the Quick Start document (literature number SPMU287) on this page to install the appropriate drivers on the host computer.

#### Downloads

DRIVER OR LIBRARY

SW-ICDI-DRIVERS – Stellaris® ICDI Drivers - Current

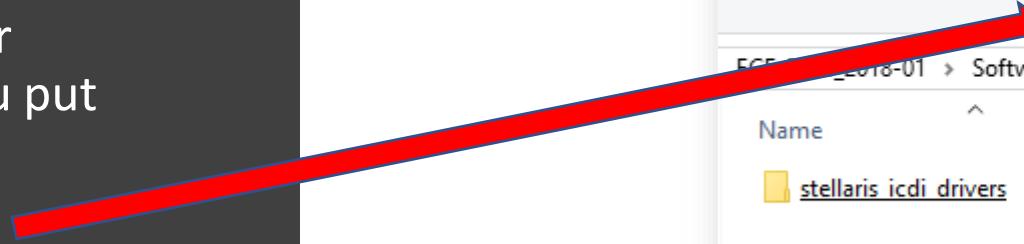
Supported products & hardware

Download

# Download Stellaris ICDI Drivers

- save the zip file to your course software folder (remember where you put this!)
- click on the zip folder
- extract all files
- optional: delete the **zip** folder and just leave the extracted folder

ECE_3436_2018-01 > Software				
Name	Date modified	Type	Size	
MDK474.EXE	2/1/2018 5:03 PM	Application	590,659 KB	
spmc016a.zip	4/10/2018 2:27 PM	Compressed (zipp...)	9,061 KB	



ECE_3436_2018-01 > Software > spmc016a.zip	
Name	Type
stellaris icdi drivers	File folder

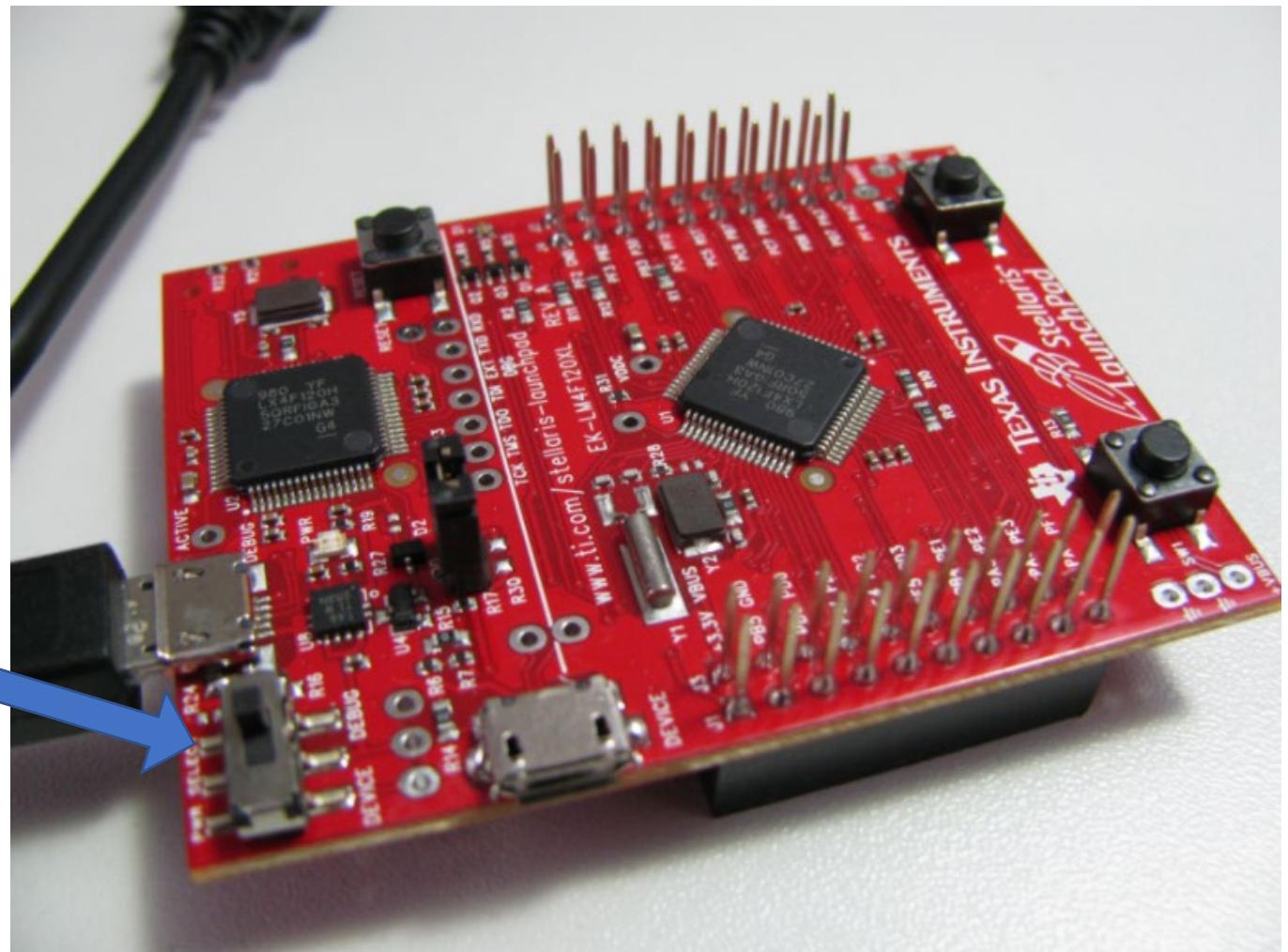
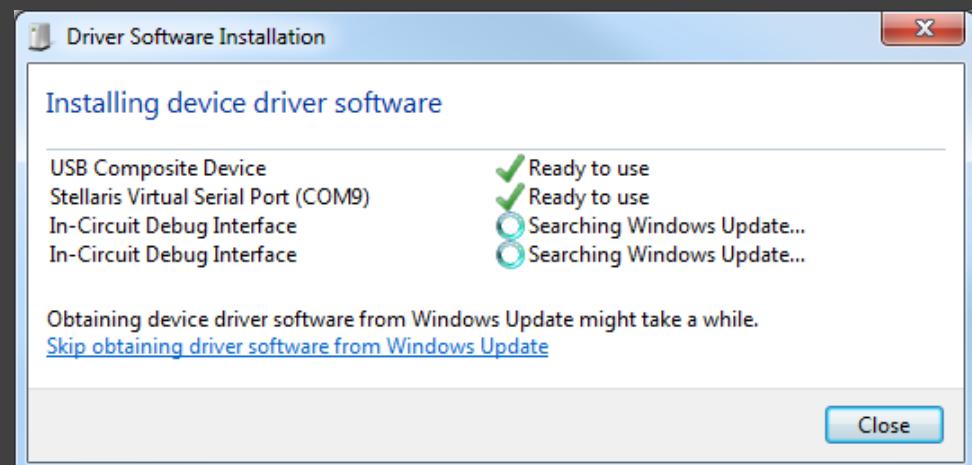
## Download Stellaris ICDI Drivers

- Your driver folder should look like what you see on the right.

Name	Date modified	Type	Size
amd64	8/24/2022 3:53 PM	File folder	
i386	8/24/2022 3:53 PM	File folder	
boot usb.cat	8/24/2022 3:53 PM	Security Catalog	15 KB
boot usb.inf	8/24/2022 3:53 PM	Setup Information	5 KB
stellaris icdi com.cat	8/24/2022 3:53 PM	Security Catalog	9 KB
stellaris icdi com.inf	8/24/2022 3:53 PM	Setup Information	2 KB
stellaris icdi debug.cat	8/24/2022 3:53 PM	Security Catalog	17 KB
stellaris icdi debug.inf	8/24/2022 3:53 PM	Setup Information	5 KB

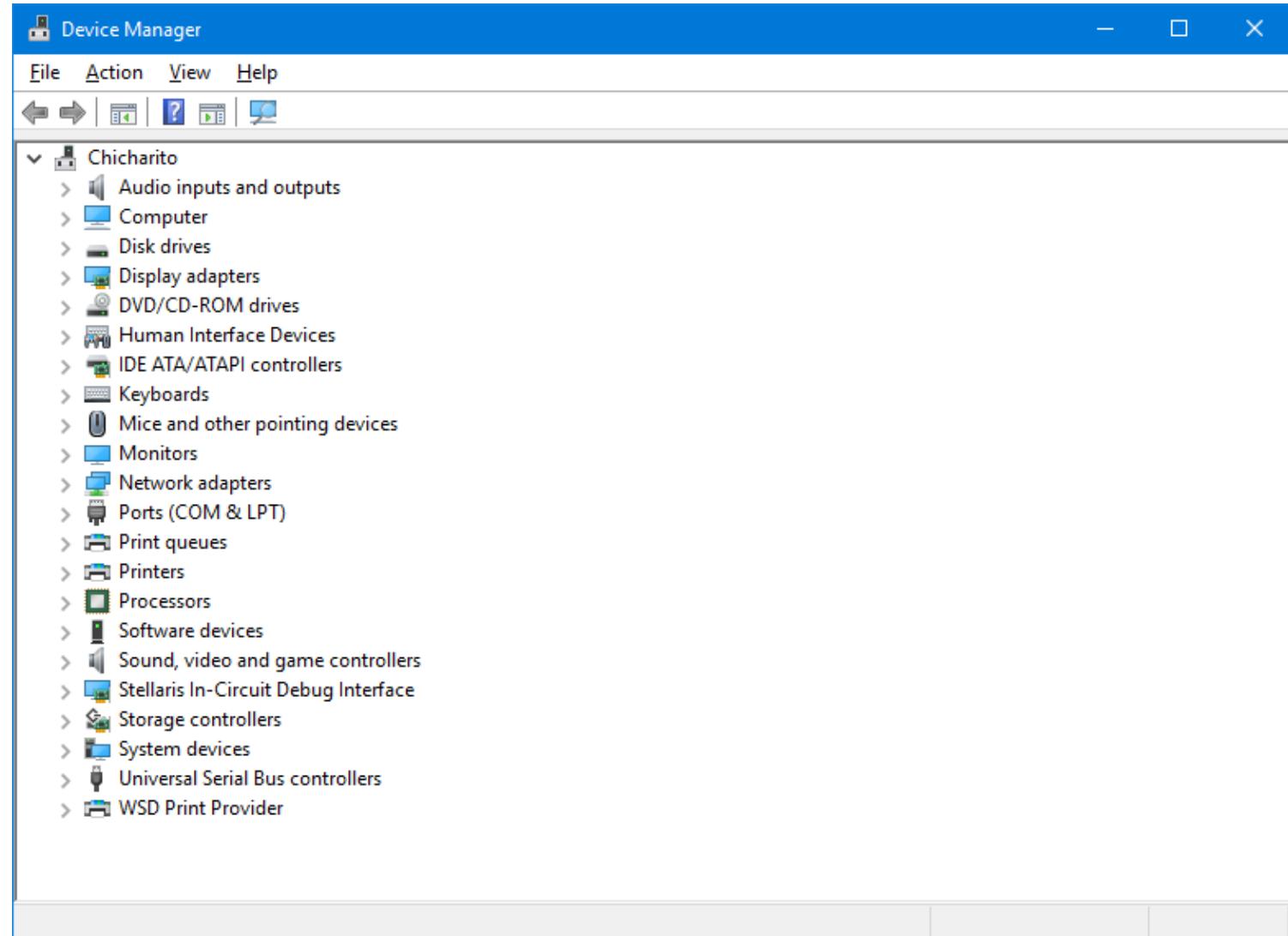
# Install Stellaris ICDI Drivers

- Plug USB cable into the LaunchPad Debug USB port
- Plug other end of USB cable into your computer
- Switch on power on LaunchPad



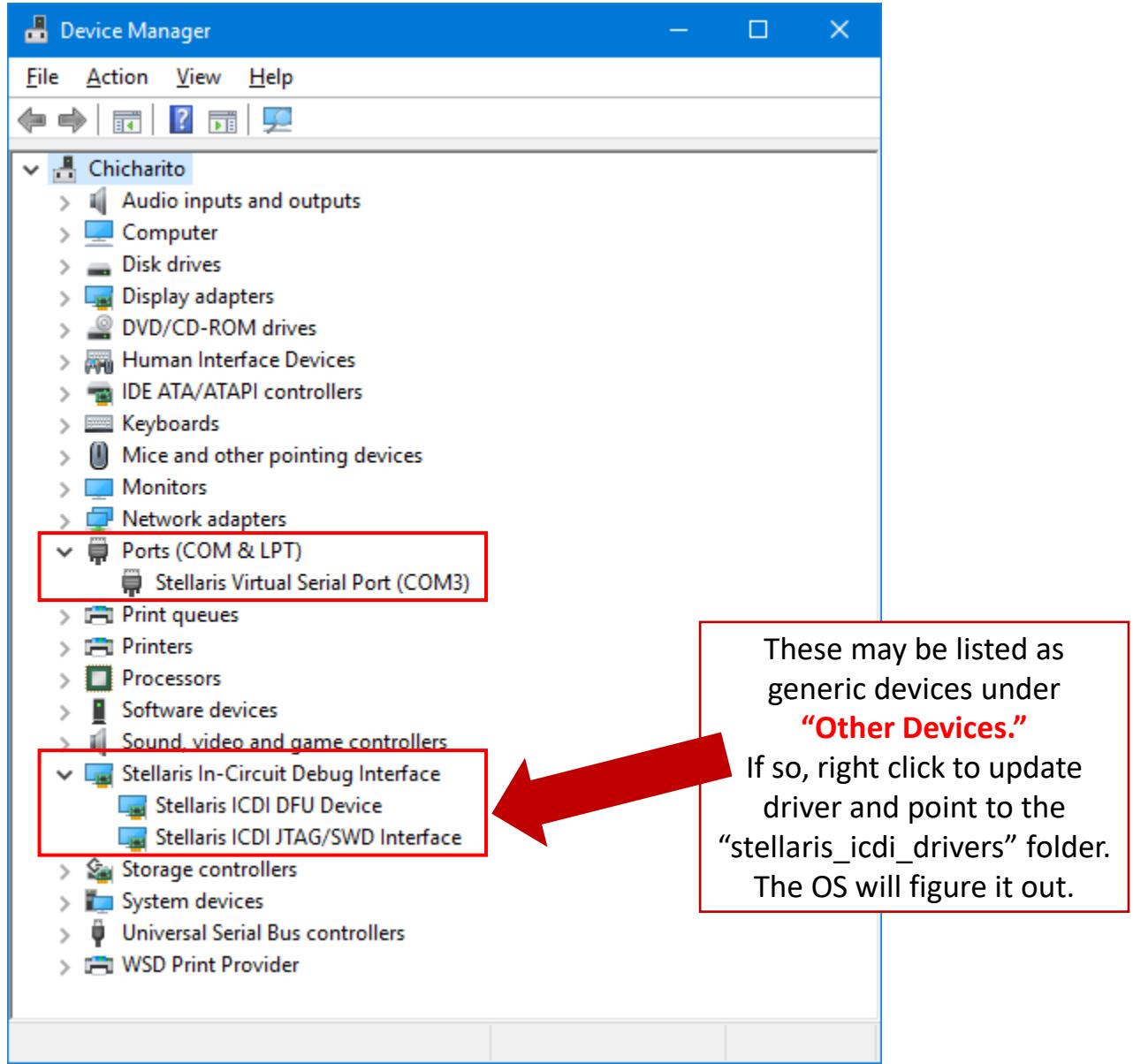
# Install Stellaris ICDI Drivers

- Open your Device Manager



# Install Stellaris ICDI Drivers

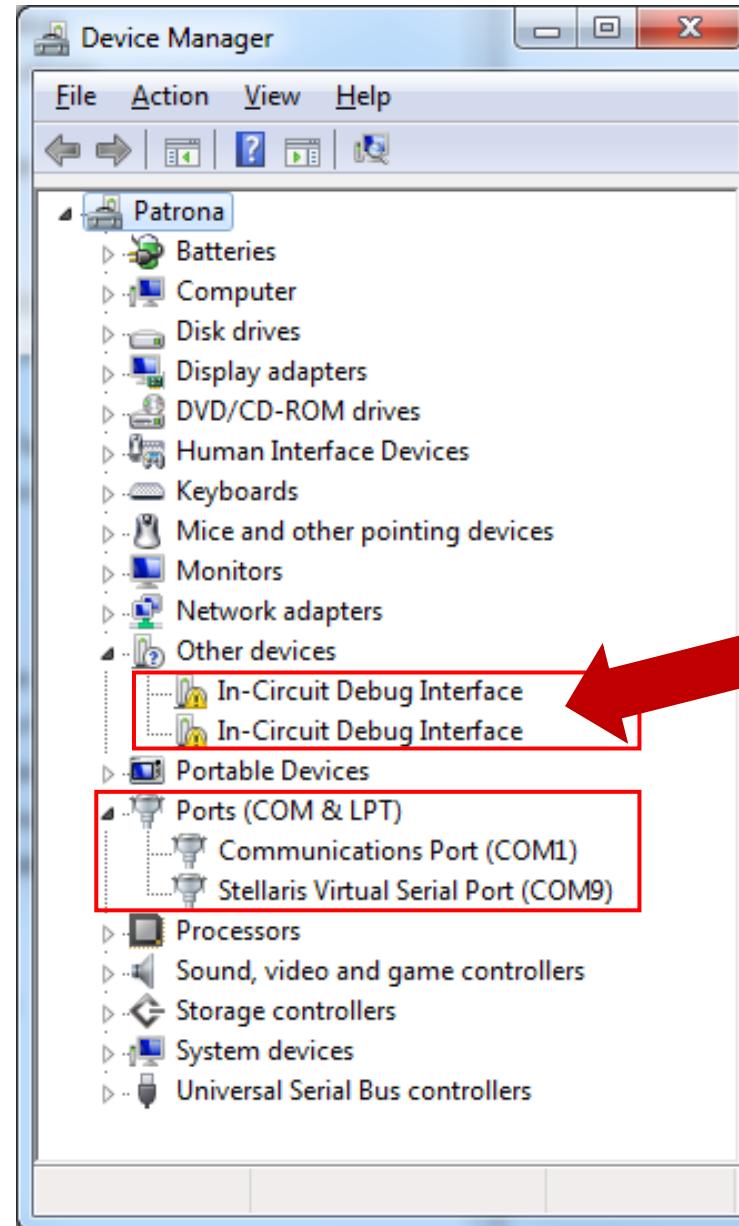
- expand the Ports tab
- update the drivers  
*(need administrator rights)*
  - right click on “Stellaris ICDI DFU Device”
  - select Update driver
  - browse for drivers on your computer
  - you will need to navigate to the folder where you stored the Stellaris drivers (see later slide)



## Install Stellaris ICDI Drivers: “Other Devices” case

- expand the Ports tab
- right click the yellow triangles (indicating missing drivers)
- update the drivers (*need administrator rights*)
  - you will need to navigate to the folder where you stored the Stellaris drivers
- the yellow triangles should disappear (as shown)

### Case when device shows up as “Other Devices”



These may be listed as generic devices under “Other Devices.”

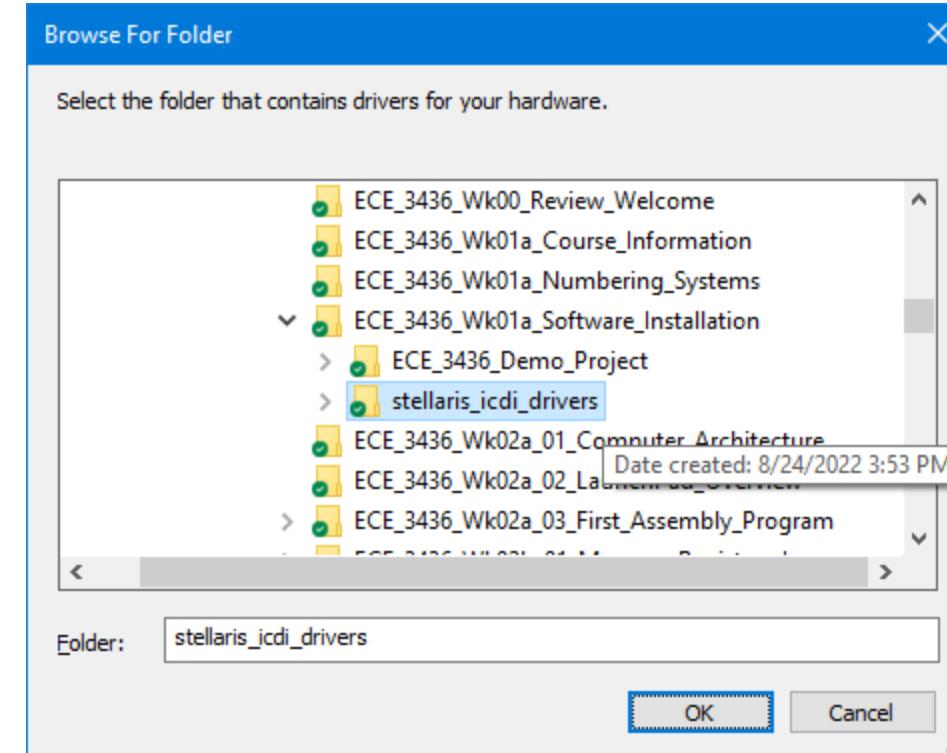
If so, right click to update driver and point to the “stellaris\_icdi\_drivers” folder.

The OS will figure it out.

You need administrator rights.

# Install Stellaris ICDI Drivers

- select the folder where your drivers are located
- Click OK
- Click Next

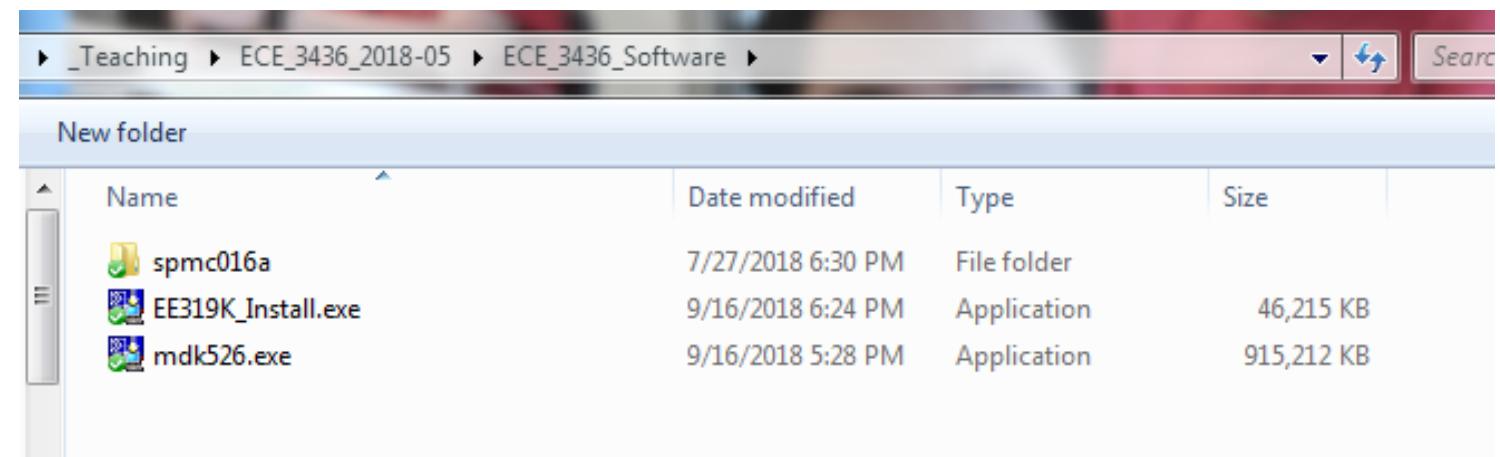


# Step 3: Download & Install the UT EE319K Software

- [http://users.ece.utexas.edu/~valvano/Volume1/EE319K\\_Install.exe](http://users.ece.utexas.edu/~valvano/Volume1/EE319K_Install.exe)
- If the link does not work, get file from **Teams**

# UT EE319K Software

- save the starter configuration executable with your course software

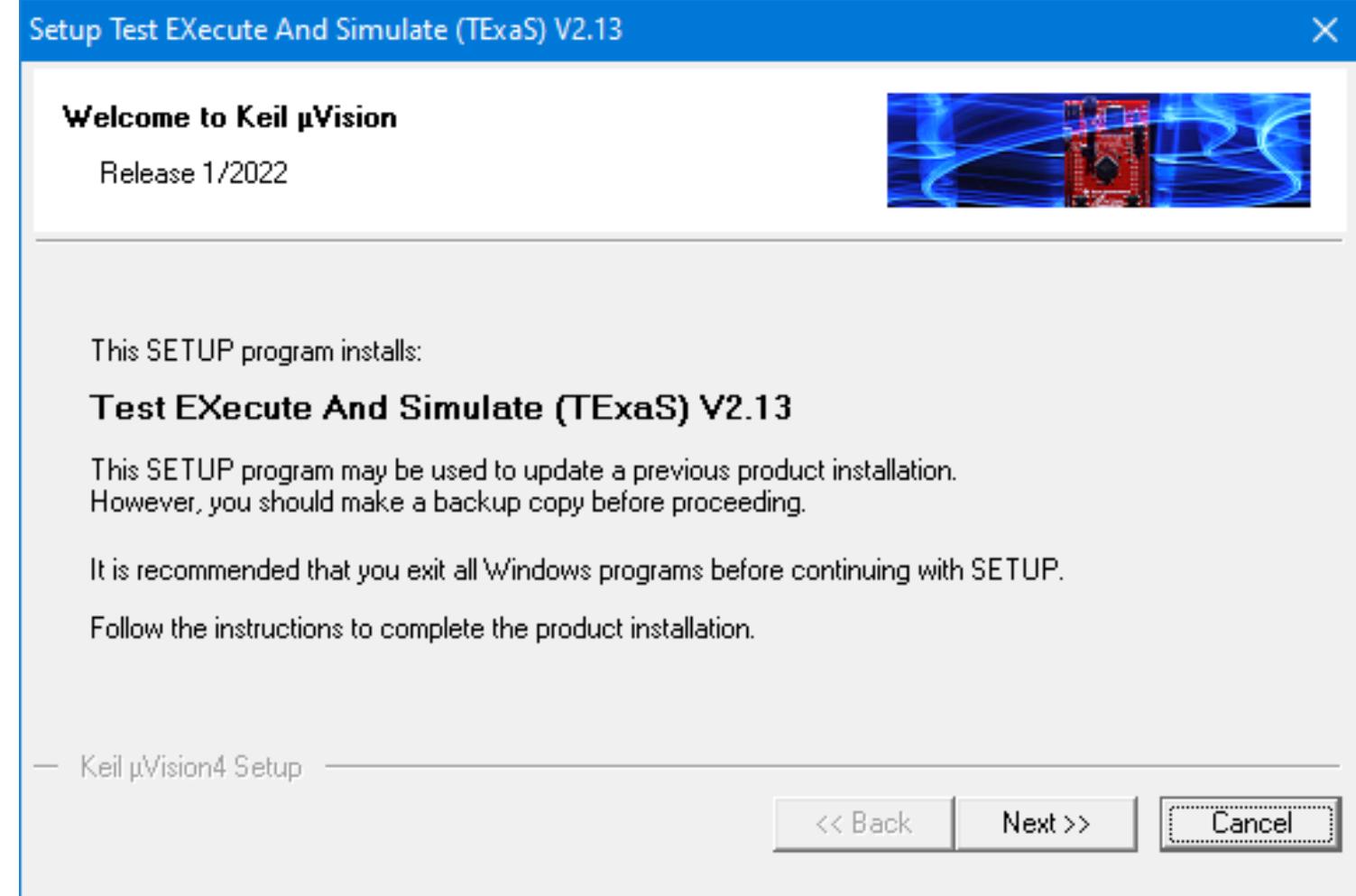


A screenshot of a Windows file explorer window. The address bar shows the path: \_Teaching > ECE\_3436\_2018-05 > ECE\_3436\_Software. The main area is titled "New folder" and displays three items in a grid:

Name	Date modified	Type	Size
spmc016a	7/27/2018 6:30 PM	File folder	
EE319K_Install.exe	9/16/2018 6:24 PM	Application	46,215 KB
mdk526.exe	9/16/2018 5:28 PM	Application	915,212 KB

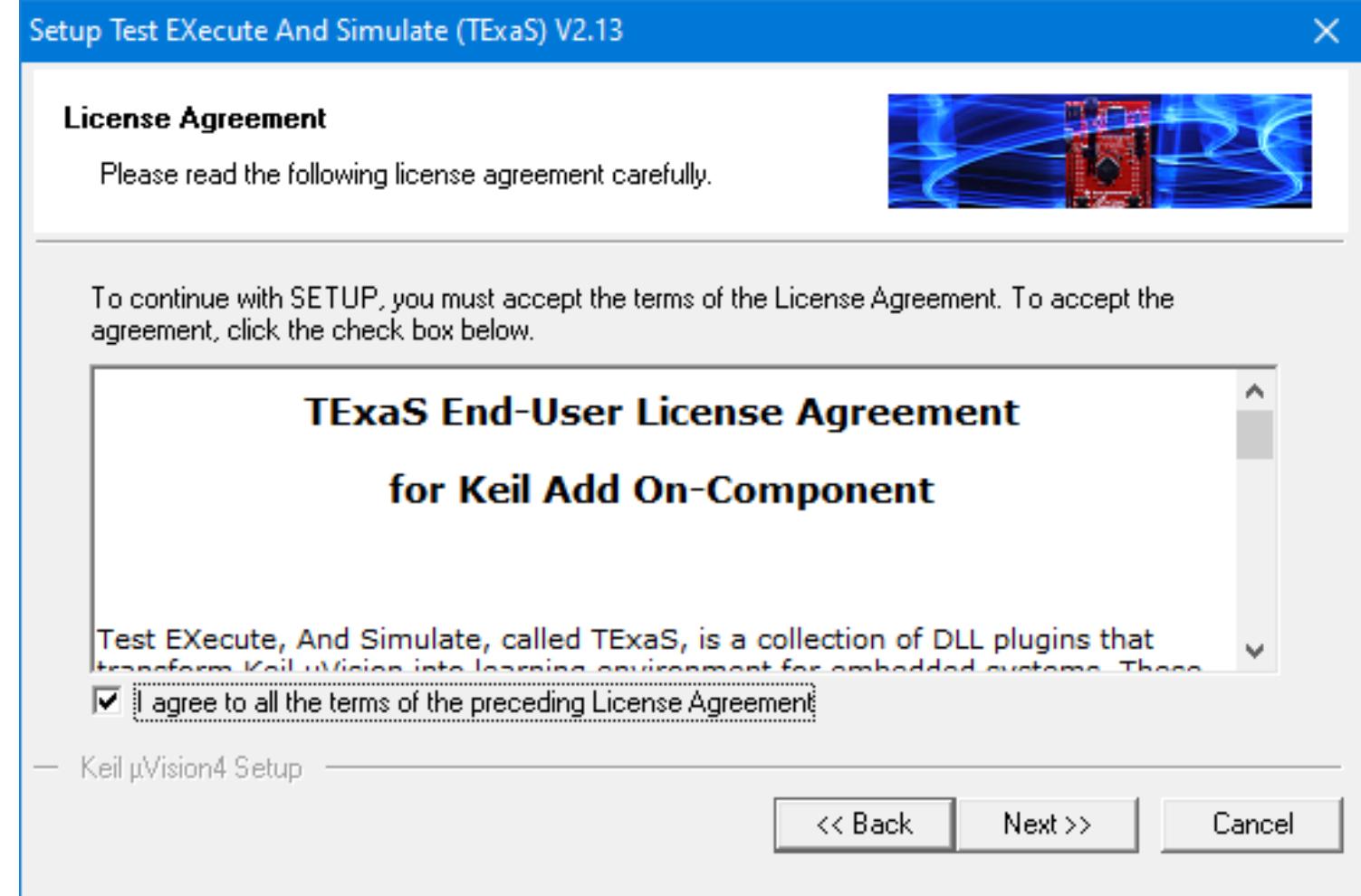
# UT EE319K Software

- run the EE319K executable
- be sure it's V2.13
- click Next



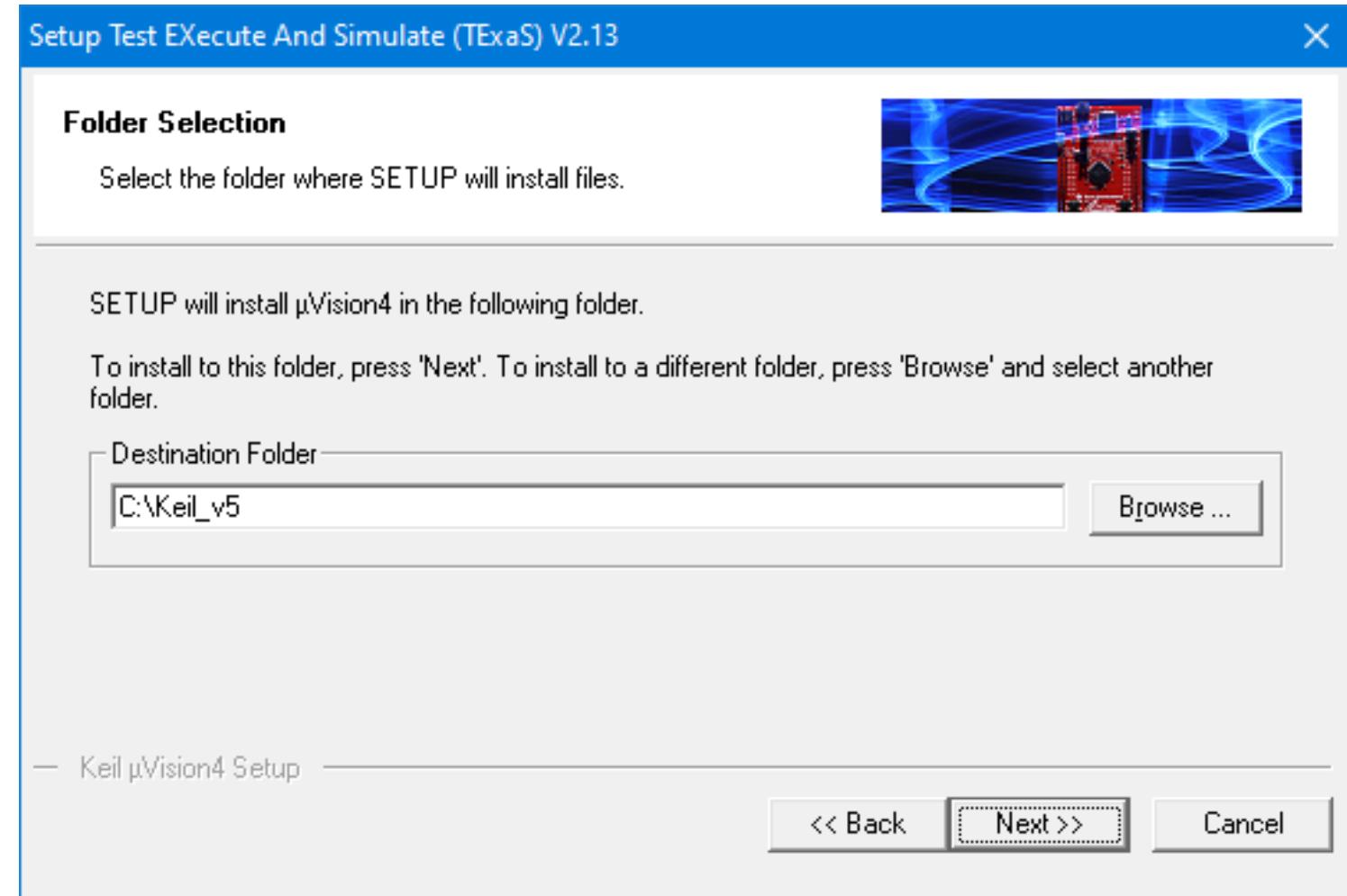
# UT EE319K Software

- read and click if you agree to the license agreement



# UT EE319K Software

- set up in your Keil folder
- click Next



# UT EE319K Software

- enter your information
- click Next

Setup Test EXecute And Simulate (TExaS) V2.13 X

**Customer Information** 

Please enter your information.

Please enter your name, the name of the company for whom you work and your E-mail address.

First Name:

Last Name:

Company Name:

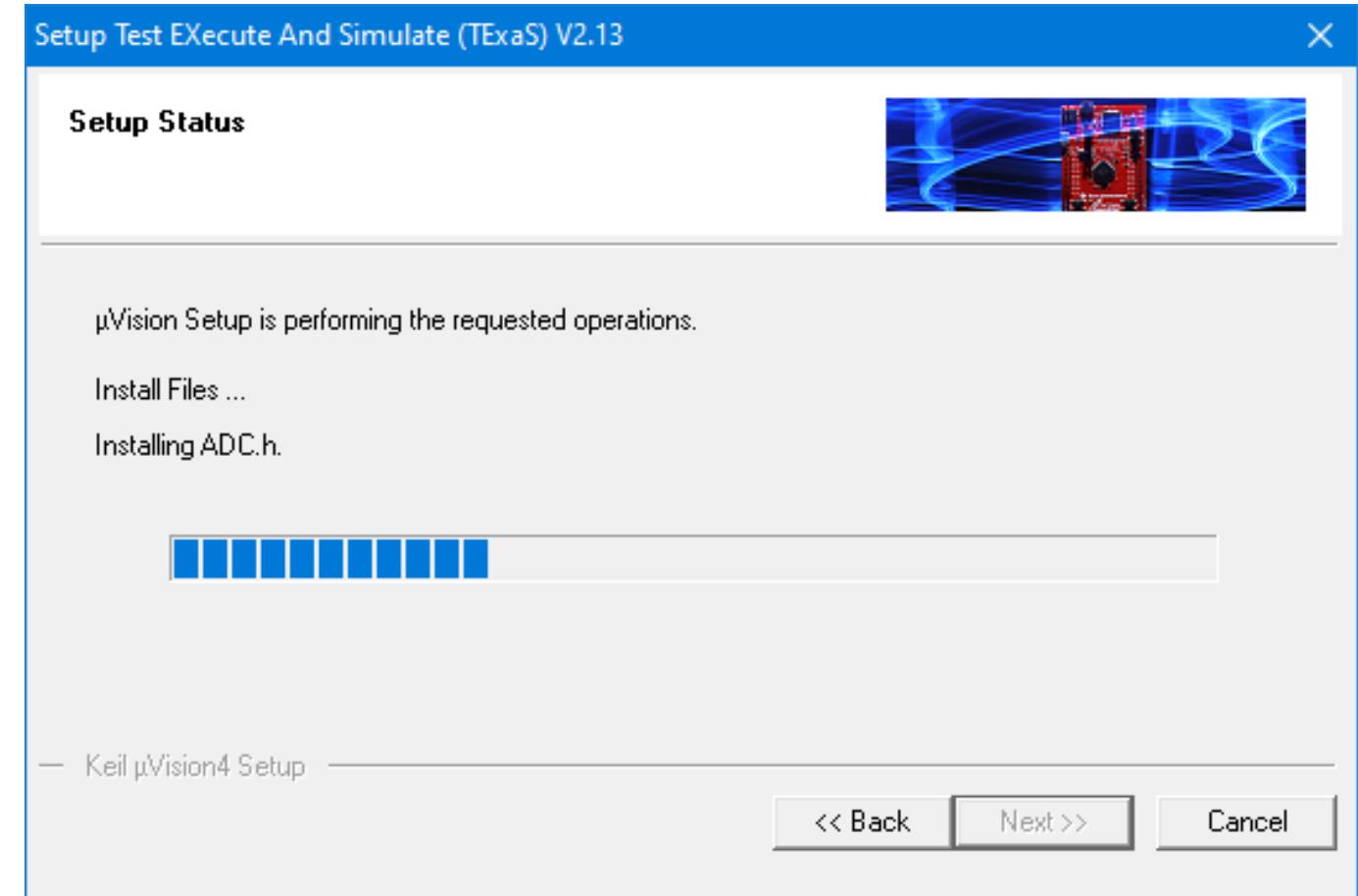
E-mail:

— Keil µVision4 Setup

[<< Back](#) [Next >>](#) [Cancel](#)

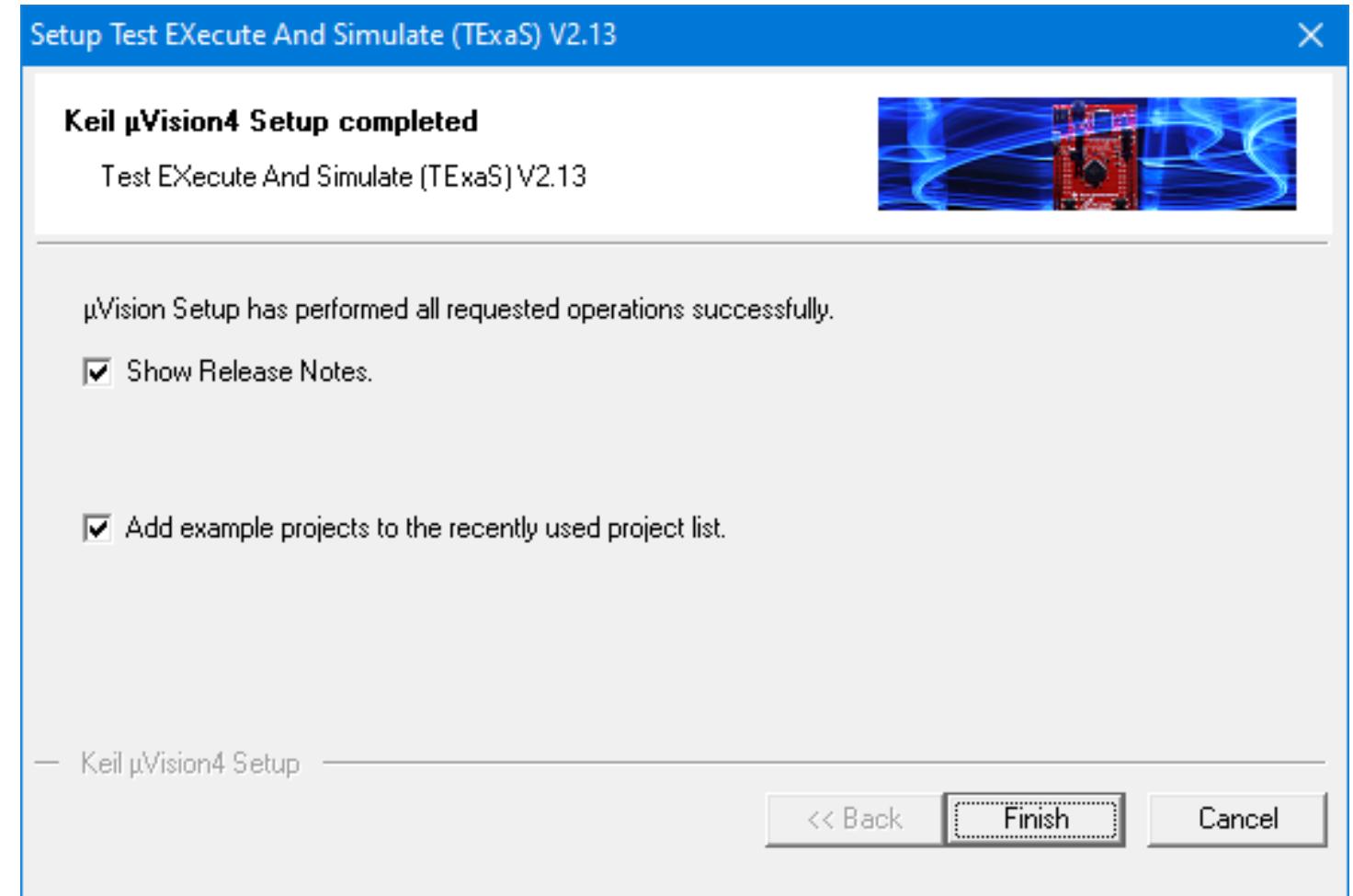
# UT EE319K Software

- wait for files to install



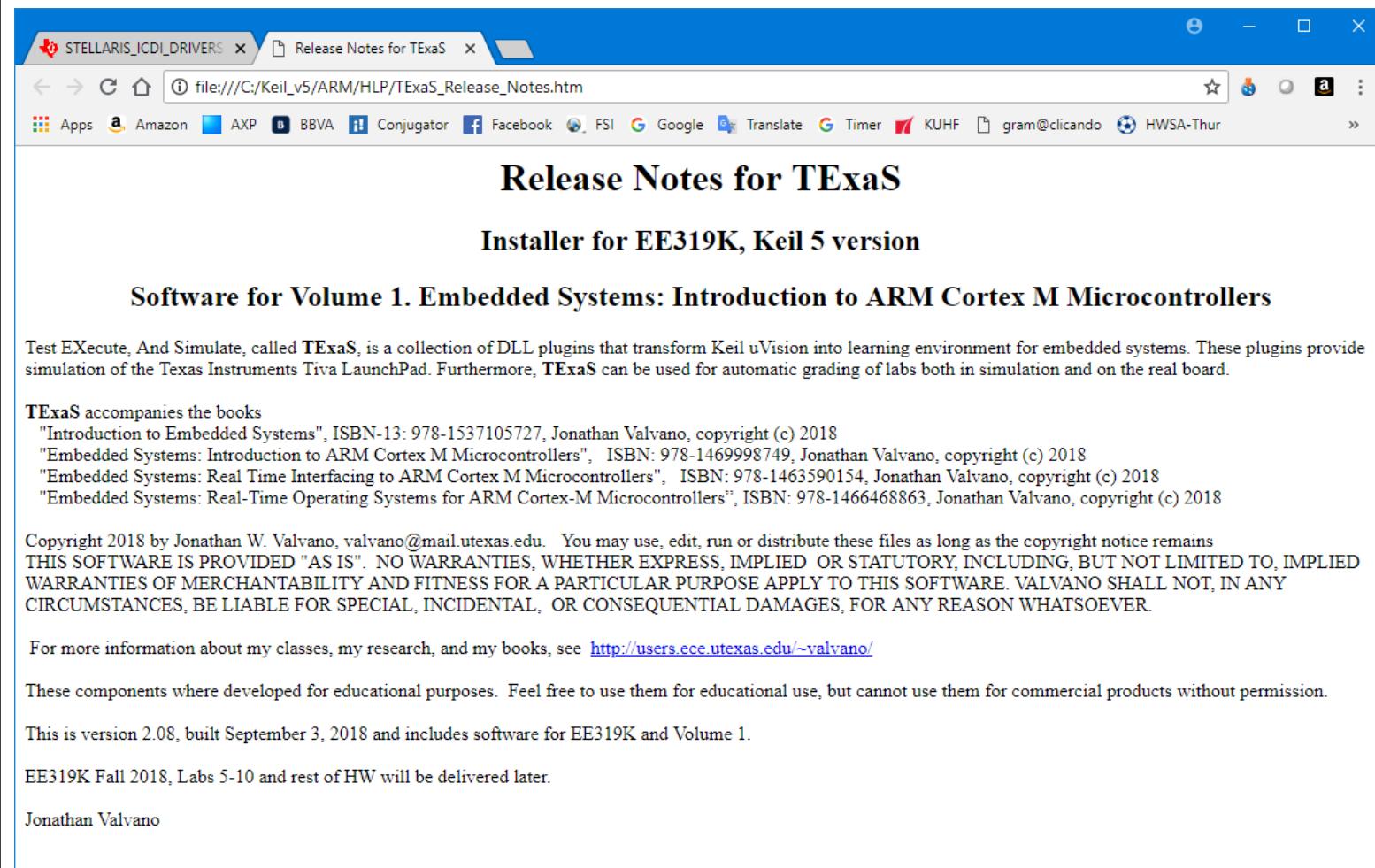
# UT EE319K Software

- the setup should only take a few seconds
- then you should see this dialog box
- leave both boxes checked
- click Finish



# UT EE319K Software

- you will be redirected to the release notes for TExaS



The screenshot shows a Microsoft Edge browser window with the title "Release Notes for TExaS". The address bar displays the URL "file:///C:/Keil\_v5/ARM/HLP/TExaS\_Release\_Notes.htm". The page content is titled "Release Notes for TExaS" and includes sections for "Installer for EE319K, Keil 5 version" and "Software for Volume 1. Embedded Systems: Introduction to ARM Cortex M Microcontrollers". It provides information about the TExaS software, its accompanying books, copyright details, and various disclaimers and notices.

**Release Notes for TExaS**

**Installer for EE319K, Keil 5 version**

**Software for Volume 1. Embedded Systems: Introduction to ARM Cortex M Microcontrollers**

Test EXecute, And Simulate, called **TExaS**, is a collection of DLL plugins that transform Keil uVision into learning environment for embedded systems. These plugins provide simulation of the Texas Instruments Tiva LaunchPad. Furthermore, **TExaS** can be used for automatic grading of labs both in simulation and on the real board.

**TExaS** accompanies the books

- "Introduction to Embedded Systems", ISBN-13: 978-1537105727, Jonathan Valvano, copyright (c) 2018
- "Embedded Systems: Introduction to ARM Cortex M Microcontrollers", ISBN: 978-146998749, Jonathan Valvano, copyright (c) 2018
- "Embedded Systems: Real Time Interfacing to ARM Cortex M Microcontrollers", ISBN: 978-1463590154, Jonathan Valvano, copyright (c) 2018
- "Embedded Systems: Real-Time Operating Systems for ARM Cortex-M Microcontrollers", ISBN: 978-1466468863, Jonathan Valvano, copyright (c) 2018

Copyright 2018 by Jonathan W. Valvano, valvano@mail.utexas.edu. You may use, edit, run or distribute these files as long as the copyright notice remains. THIS SOFTWARE IS PROVIDED "AS IS". NO WARRANTIES, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE APPLY TO THIS SOFTWARE. VALVANO SHALL NOT, IN ANY CIRCUMSTANCES, BE LIABLE FOR SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, FOR ANY REASON WHATSOEVER.

For more information about my classes, my research, and my books, see <http://users.ece.utexas.edu/~valvano/>

These components were developed for educational purposes. Feel free to use them for educational use, but cannot use them for commercial products without permission.

This is version 2.08, built September 3, 2018 and includes software for EE319K and Volume 1.

EE319K Fall 2018, Labs 5-10 and rest of HW will be delivered later.

Jonathan Valvano

# UT EE319K Software

- your Keil folder should look similar to this now
- Of course, the year(s) will change depending on when you take the course.

Name	Date modified	Type	Size
ARM	8/24/2022 3:35 PM	File folder	
EE319KwareSpring2021	3/22/2021 10:18 PM	File folder	
EE319KwareSpring2022	8/24/2022 4:19 PM	File folder	
license terms	3/22/2021 10:01 PM	File folder	
UV4	8/24/2022 3:35 PM	File folder	
TOOLS.INI	8/24/2022 4:19 PM	Configuration sett...	6 KB
Uninstall.exe	4/10/2022 12:55 PM	Application	3,228 KB

Step 4: Install  
the uVision  
Driver Patch

# Software Installation

- Even though you installed the drivers, uVision doesn't show them as an option until you **install a patch.**
- so do that...

<https://www.keil.com/support/docs/4196.htm>

The screenshot shows a web browser displaying a page from the arm Developer website. The URL in the address bar is <https://www.keil.com/support/docs/4196.htm>. The page title is "arm Developer". The main content area displays a knowledgebase article titled "UVISION: Stellaris ICDI Debug Adapter Support". The article includes sections for "Applies To", "Confidentiality", and "Information in this knowledgebase article applies to". A red arrow points from the bottom right towards the "Attachments" section, which lists a file named "MDK\_Stellaris\_ICDI\_AddOn.exe".

Version: 1.0 | [Subscribe](#) | [Search within this document](#)

Applies To: Keil MDK  
Confidentiality: Customer Non-confidential  
Information in this knowledgebase article applies to:  
• MDK v5.29

## QUESTION

I'm working on the device TI TM4C123GH6PM within MDK µVision. When I try starting the application I get the following error message:

```
lmidk-agdi.dll cannot be loaded
```

How can I resolve this issue?

## ANSWER

In MDK v5.29 as well as newer versions of MDK the support for the Stellaris ICDI debug adapter is included in the main MDK installation. There is an add-on Stellaris ICDI installer that supports MDK v5.29. Just download and install it.

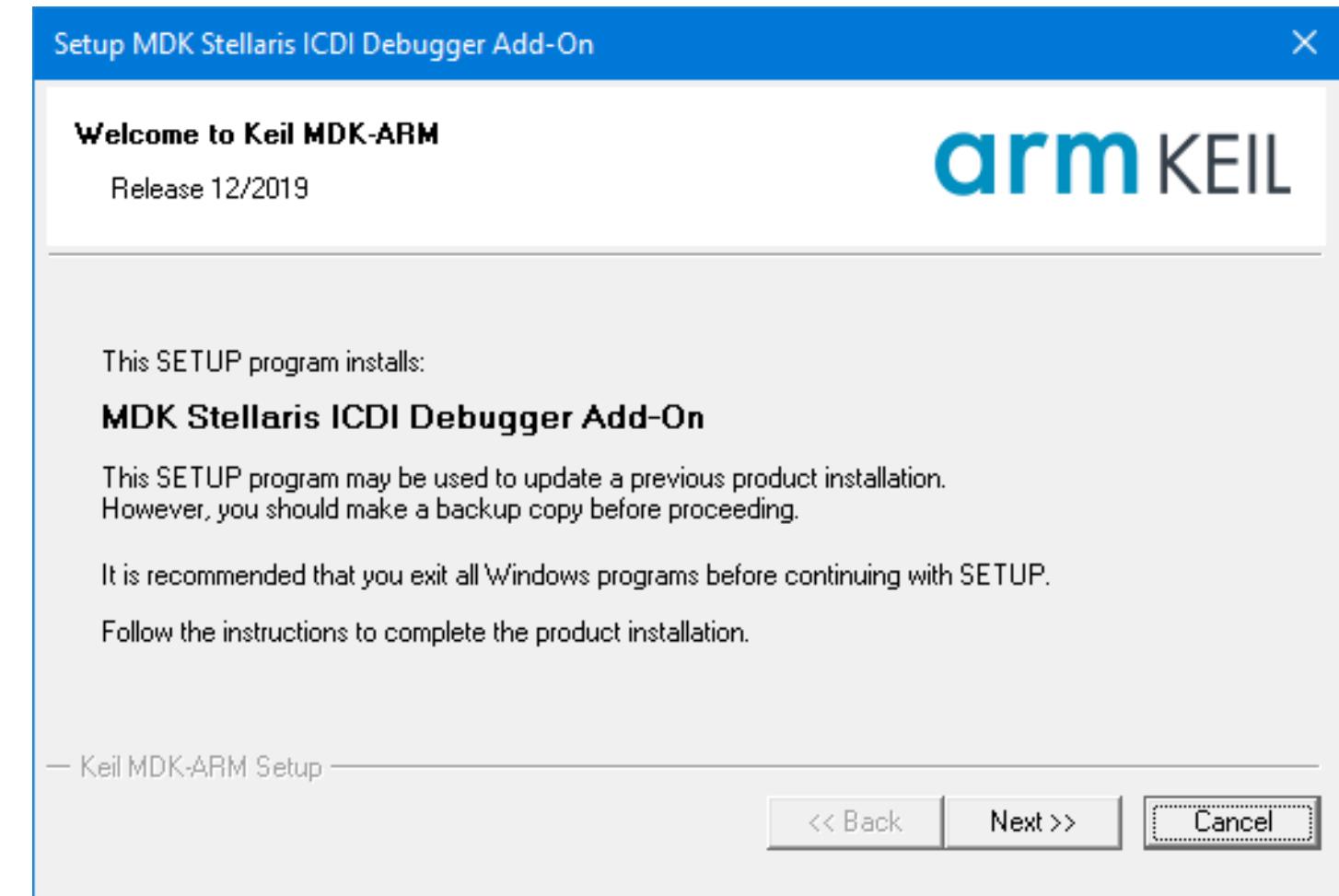
## Attachments

- [MDK\\_Stellaris\\_ICDI\\_AddOn.exe](#)

## Software Installation

- click Next

<https://www.keil.com/support/docs/4196.htm>



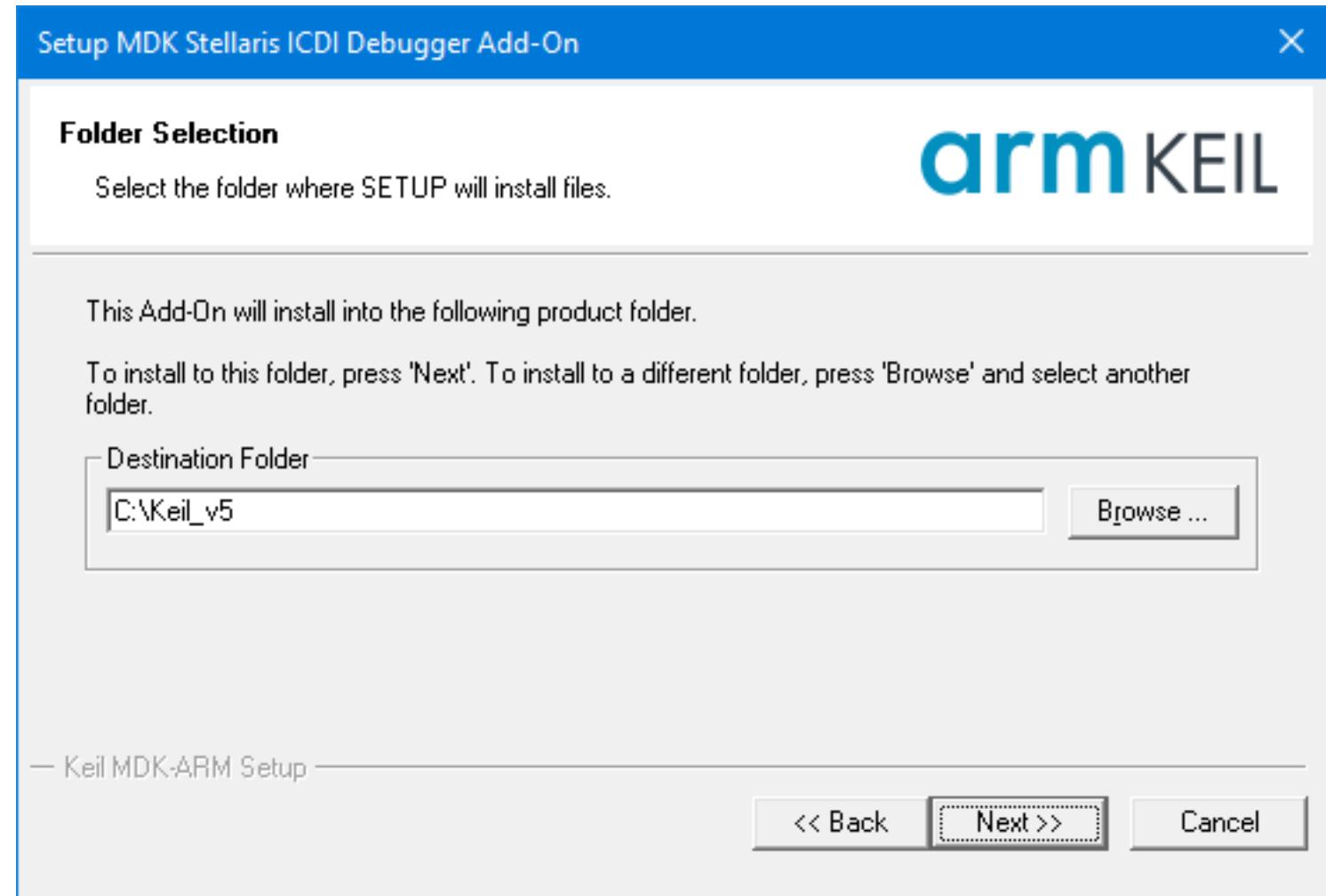
# Software Installation

- agree and click Next



## Software Installation

- agree and click Next



# Software Installation

- enter YOUR information

Setup MDK Stellaris ICDI Debugger Add-On X

**Customer Information**

Please enter your information.

arm KEIL

Please enter your name, the name of the company for whom you work, and your E-mail address.

First Name:

Last Name:

Company Name:

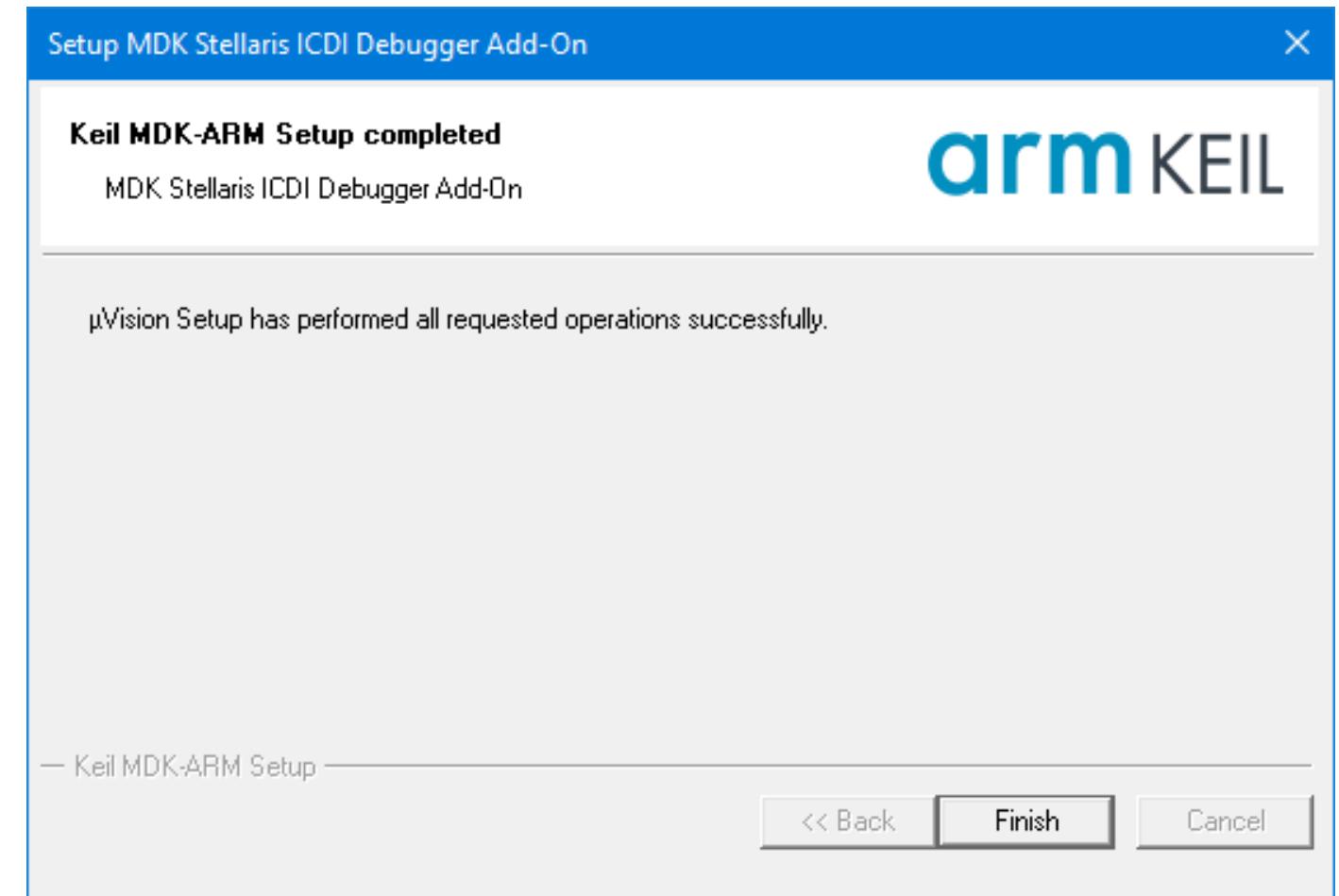
E-mail:

— Keil MDK-ARM Setup —

[« Back](#) [Next >>](#) [Cancel](#)

## Software Installation

- click Finish



## Step 5: Test Your Software in Simulator

## Software Simulator Test

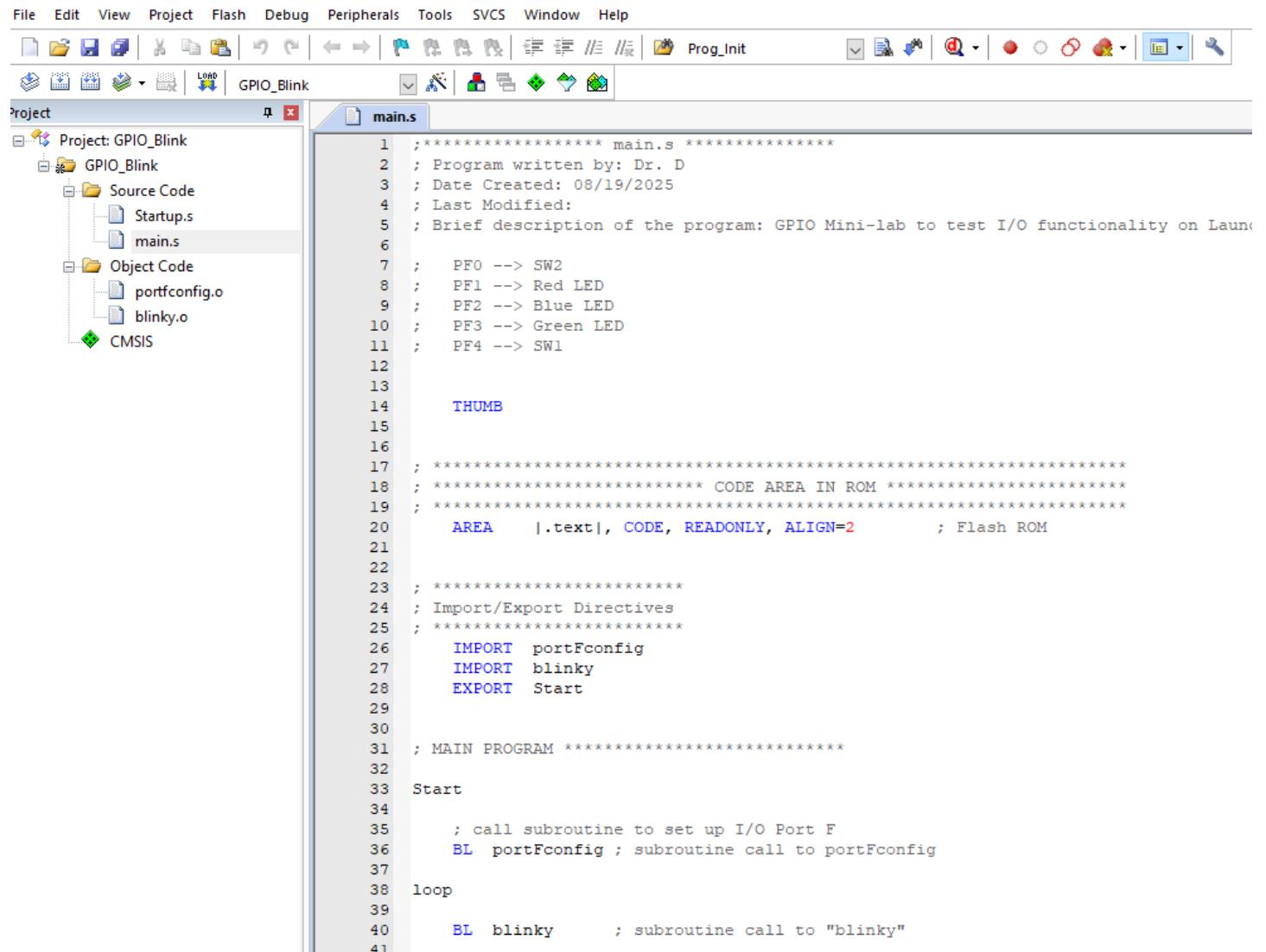
- open the **Lab0** project file in uVision
- the project file always ends in **.uvprojx**



Name	Status
 <a href="#">blinky.o</a>	
 <a href="#">GPIO_Blink.uvprojx</a>	
 <a href="#">main.s</a>	
 <a href="#">portfconfig.o</a>	
 <a href="#">Startup.s</a>	

# Software Simulator Test

- if the **main.s** file is not showing, double-click on the filename



The screenshot shows a software development environment with a toolbar at the top and a menu bar below it. The menu bar includes File, Edit, View, Project, Flash, Debug, Peripherals, Tools, SVCS, Window, and Help. Below the menu bar is a toolbar with various icons. The main window has a title bar "GPIO\_Blink" and a tab labeled "main.s". On the left is a "Project" pane showing a tree structure for the "GPIO\_Blink" project, which contains "Source Code" (with "Startup.s" and "main.s" selected), "Object Code" (with "portconfig.o" and "blinky.o"), and "CMSIS". The right pane displays the assembly code for "main.s".

```
1 ; **** main.s ****
2 ; Program written by: Dr. D
3 ; Date Created: 08/19/2025
4 ; Last Modified:
5 ; Brief description of the program: GPIO Mini-lab to test I/O functionality on LaunchPad
6
7 ; PF0 --> SW2
8 ; PF1 --> Red LED
9 ; PF2 --> Blue LED
10 ; PF3 --> Green LED
11 ; PF4 --> SW1
12
13
14 THUMB
15
16
17 ; **** CODE AREA IN ROM ****
18 ; **** AREA .text, CODE, READONLY, ALIGN=2 ; Flash ROM ****
19 ; **** Import/Export Directives ****
20 AREA .text, CODE, READONLY, ALIGN=2 ; Flash ROM
21
22
23 ; **** Import/Export Directives ****
24 ; Import/Export Directives
25 ; ****
26 IMPORT portFconfig
27 IMPORT blinky
28 EXPORT Start
29
30
31 ; MAIN PROGRAM ****
32
33 Start
34
35 ; call subroutine to set up I/O Port F
36 BL portFconfig ; subroutine call to portFconfig
37
38 loop
39
40 BL blinky ; subroutine call to "blinky"
41
```

# Software Simulator Test

- rebuild (assemble) the entire project by clicking on the icon with 2 down arrows



- you should see a Build Output message like that shown to the right

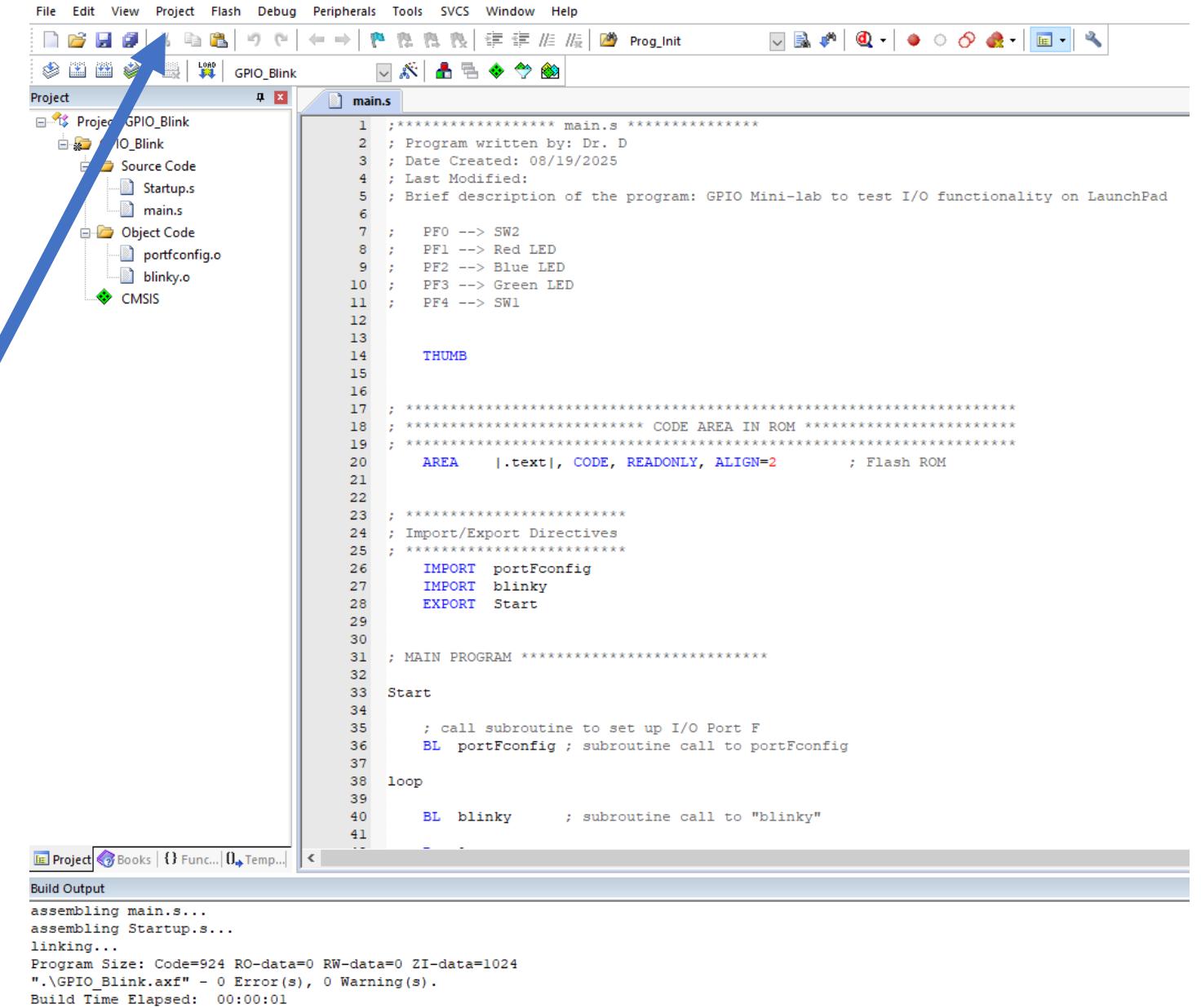
The screenshot shows a software development environment with the following interface elements:

- Menu Bar:** File, Edit, View, Project, Flash, Debug, Peripherals, Tools, SVCS, Window, Help.
- Toolbar:** Includes icons for file operations, project management, and debugging.
- Project Explorer:** Shows the project structure: Project: GPIO\_Blink, which contains Source Code (Startup.s, main.s), Object Code (portconfig.o, blinky.o), and CMSIS.
- Code Editor:** The main.s file is open, displaying assembly code for a GPIO Blink application. The code includes directives for memory areas (.text), imports for portFconfig and blinky, and exports for Start. It also defines a main program loop calling portFconfig and blinky.
- Build Output:** A window at the bottom shows the build process and results:

```
assembling main.s...
assembling Startup.s...
linking...
Program Size: Code=924 RO-data=0 RW-data=0 ZI-data=1024
".\GPIO_Blink.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed: 00:00:01
```

# Software Simulator Test

- make sure that you are in simulator mode
  - goto the **Project** menu
  - select **Options for Target 'GPIO\_Blink'**



The screenshot shows a software development environment with a toolbar at the top containing various icons. Below the toolbar is a menu bar with options: File, Edit, View, Project, Flash, Debug, Peripherals, Tools, SVCS, Window, Help. The 'Project' menu is highlighted with a blue arrow. To the right of the menu bar is a 'Project' window showing a tree view of files: Project GPIO\_Blink, IO\_Blink, Source Code (containing Startup.s and main.s), Object Code (containing portfconfig.o and blinky.o), and CMSIS. The main code editor window displays the assembly language file 'main.s'. The code in 'main.s' is as follows:

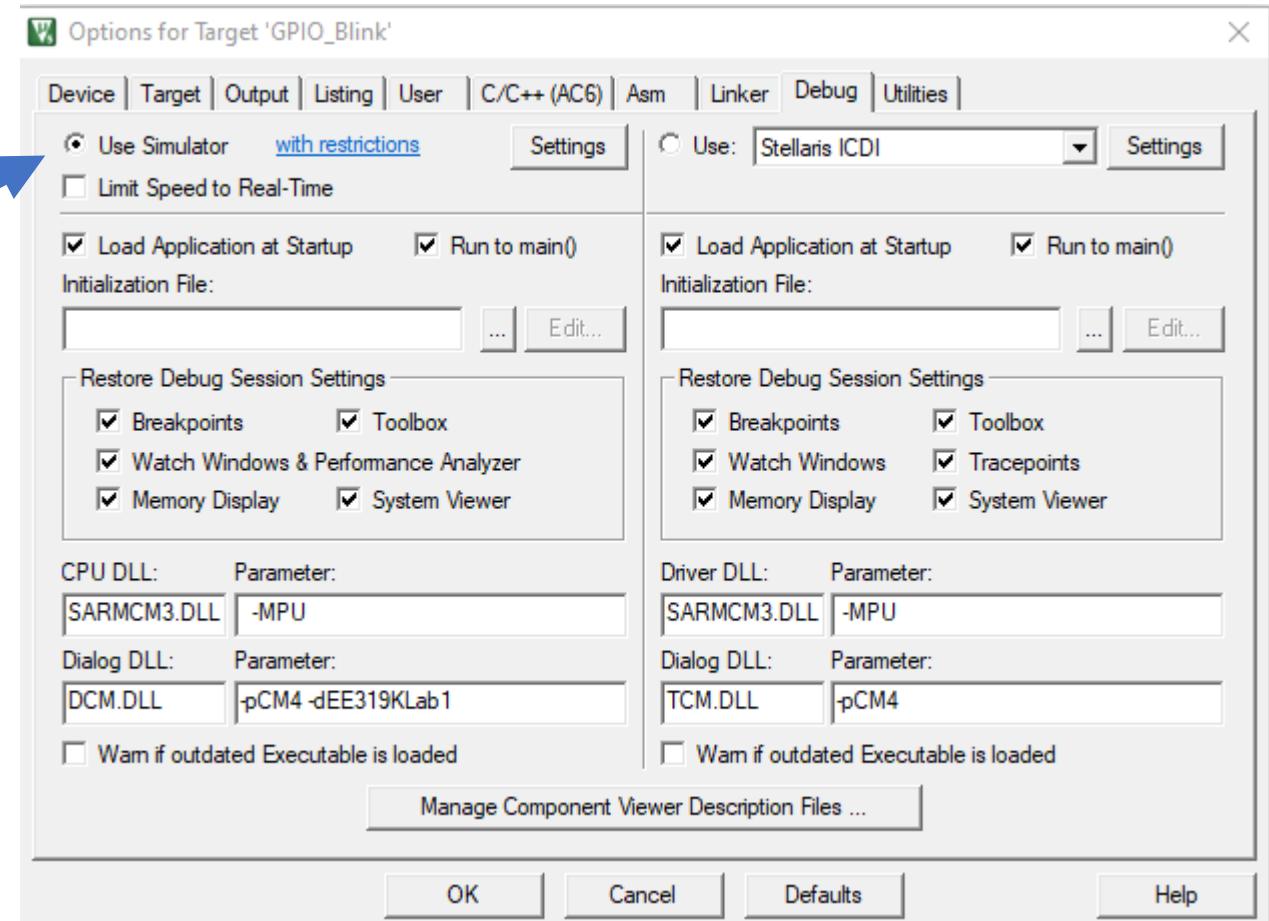
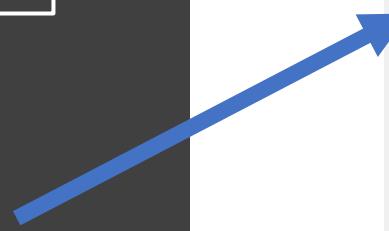
```
1 ;***** main.s *****
2 ; Program written by: Dr. D
3 ; Date Created: 08/19/2025
4 ; Last Modified:
5 ; Brief description of the program: GPIO Mini-lab to test I/O functionality on LaunchPad
6
7 ; PF0 --> SW2
8 ; PF1 --> Red LED
9 ; PF2 --> Blue LED
10 ; PF3 --> Green LED
11 ; PF4 --> SW1
12
13        THUMB
14
15
16
17 ; **** CODE AREA IN ROM ****
18 ; ***** Import/Export Directives ****
19 ; ***** Import/Export Directives ****
20     AREA    .text|, CODE, READONLY, ALIGN=2          ; Flash ROM
21
22
23 ; ***** Import/Export Directives ****
24 ; Import/Export Directives
25 ; ***** Import/Export Directives ****
26     IMPORT  portFconfig
27     IMPORT  blinky
28     EXPORT  Start
29
30 ; MAIN PROGRAM *****
31
32 Start
33
34     ; call subroutine to set up I/O Port F
35     BL  portFconfig ; subroutine call to portFconfig
36
37 loop
38
39     BL  blinky      ; subroutine call to "blinky"
40
41
```

At the bottom of the interface, there is a 'Build Output' window showing the build process:

```
Build Output
assembling main.s...
assembling Startup.s...
linking...
Program Size: Code=924 RO-data=0 RW-data=0 ZI-data=1024
".\GPIO_Blink.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed: 00:00:01
```

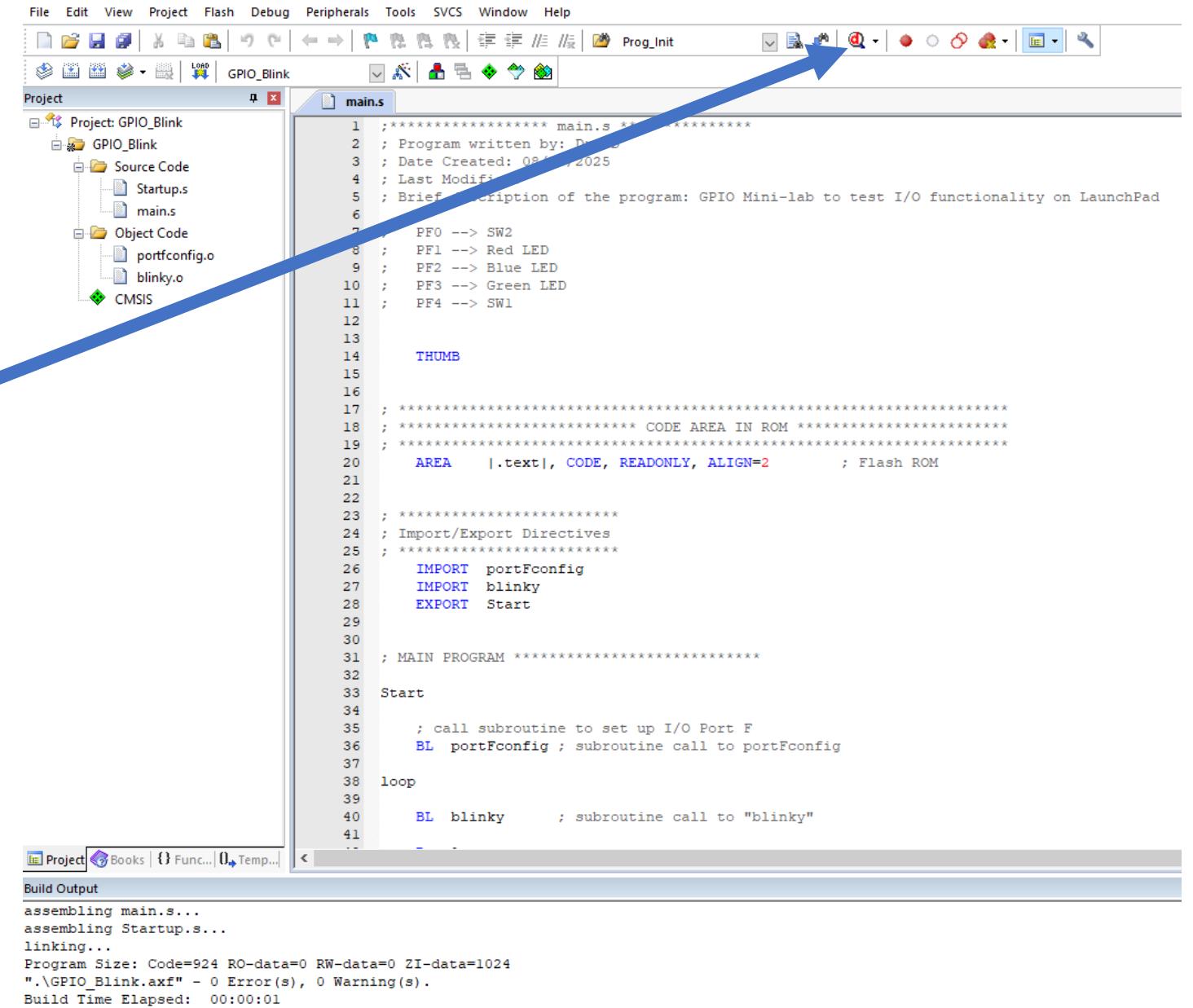
# Software Simulator Test

- make sure that “Use Simulator” is selected in the Debug tab



# Software Simulator Demo

- begin a debugging session 
- this starts the simulator



File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Project: GPIO\_Blink

main.s

```
1 ;***** main.s *****
2 ; Program written by: David L. Johnson
3 ; Date Created: 08/17/2025
4 ; Last Modified: 08/17/2025
5 ; Brief Description of the program: GPIO Mini-lab to test I/O functionality on LaunchPad
6
7 ; PF0 --> SW2
8 ; PF1 --> Red LED
9 ; PF2 --> Blue LED
10 ; PF3 --> Green LED
11 ; PF4 --> SW1
12
13 THUMB
14
15
16
17 ; **** CODE AREA IN ROM ****
18 ; **** Import/Export Directives ****
19 ; ****
20 AREA .text, CODE, READONLY, ALIGN=2 ; Flash ROM
21
22
23 ; **** Import/Export Directives ****
24 ; Import/Export Directives
25 ; ****
26 IMPORT portFconfig
27 IMPORT blinky
28 EXPORT Start
29
30 ; MAIN PROGRAM ****
31
32 Start
33
34 ; call subroutine to set up I/O Port F
35 BL portFconfig ; subroutine call to portFconfig
36
37 loop
38
39 BL blinky ; subroutine call to "blinky"
40
41
```

Build Output

```
assembling main.s...
assembling Startup.s...
linking...
Program Size: Code=924 RO-data=0 RW-data=0 ZI-data=1024
".\GPIO_Blink.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed: 00:00:01
```

# Software Simulator Demo

• now you  
are in  
simulation  
mode

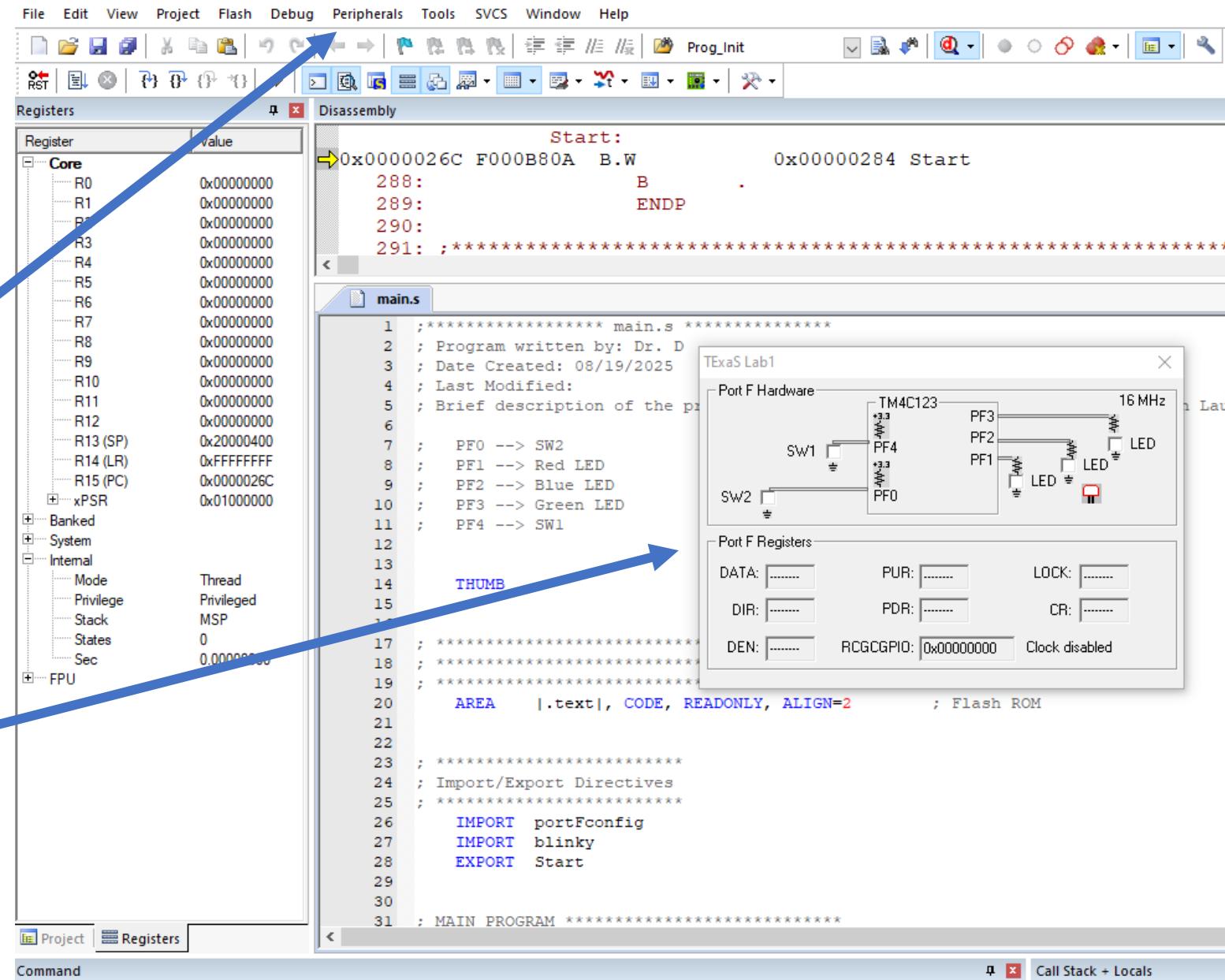
The screenshot shows a software development environment for a microcontroller. The top menu bar includes File, Edit, View, Project, Flash, Debug, Peripherals, Tools, SVCS, Window, and Help. The toolbar contains various icons for file operations, simulation, and debugging. The left pane displays the 'Registers' window, which lists the core registers (R0-R15, xPSR) and system registers (Banked, Internal, FPU) with their current values. The right pane has two sections: 'Disassembly' and 'main.s'. The Disassembly view shows assembly code starting at address 0x00000026C, including instructions like B.W, ENDP, and a series of asterisks. The main.s view shows the source code for the program, which includes comments about the program's purpose, pin assignments (PFO-SW2, PF1-Red LED, PF2-Blue LED, PF3-Green LED, PF4-SW1), memory areas (CODE AREA IN ROM, Flash ROM), and imports/exports (portFconfig, blinky, Start). The assembly code includes directives like AREA, CODE, READONLY, ALIGN=2, and EXPORT.

Register	Value
R0	0x00000000
R1	0x00000000
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x20000400
R14 (LR)	0xFFFFFFFF
R15 (PC)	0x0000026C
xPSR	0x01000000

```
Start:  
0x00000026C F000B80A B.W      0x000000284 Start  
288: B  
289: ENDP  
290:  
291: ;*****  
  
main.s  
1 ;***** main.s *****  
2 ; Program written by: Dr. D  
3 ; Date Created: 08/19/2025  
4 ; Last Modified:  
5 ; Brief description of the program: GPIO Mini-lab to test I/O functionality on Laun  
6  
7 ; PFO --> SW2  
8 ; PF1 --> Red LED  
9 ; PF2 --> Blue LED  
10 ; PF3 --> Green LED  
11 ; PF4 --> SW1  
12  
13  
14 THUMB  
15  
16  
17 ;*****  
18 ; ***** CODE AREA IN ROM *****  
19 ;*****  
20 AREA |.text|, CODE, READONLY, ALIGN=2 ; Flash ROM  
21  
22  
23 ;*****  
24 ; Import/Export Directives  
25 ;*****  
26 IMPORT portFconfig  
27 IMPORT blinky  
28 EXPORT Start  
29  
30  
31 ; MAIN PROGRAM *****
```

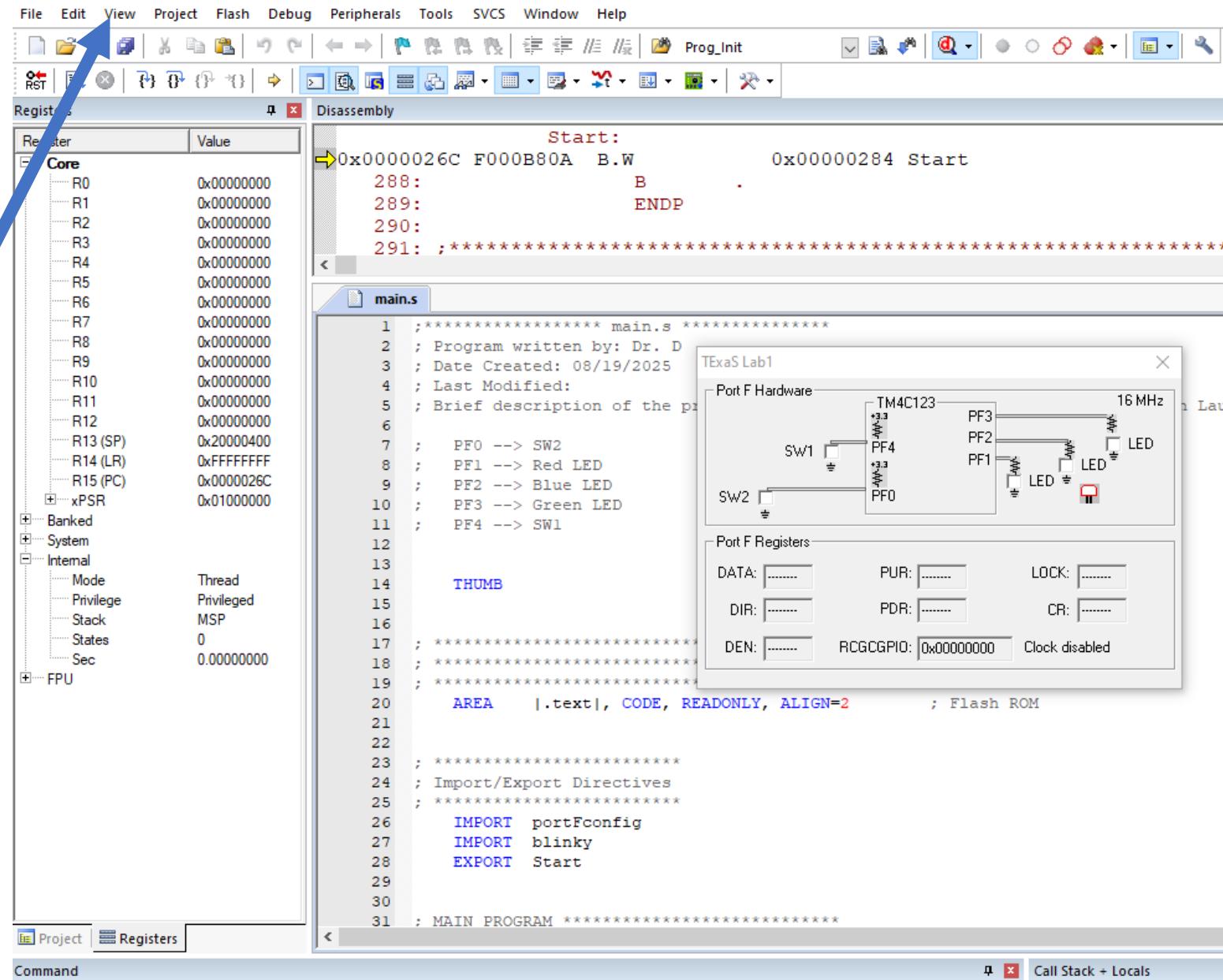
# Software Simulator Demo

- goto Peripherals menu and select “TExaS Port F”
- this opens the graphical simulator for Port F
  - this is what the UT software was for



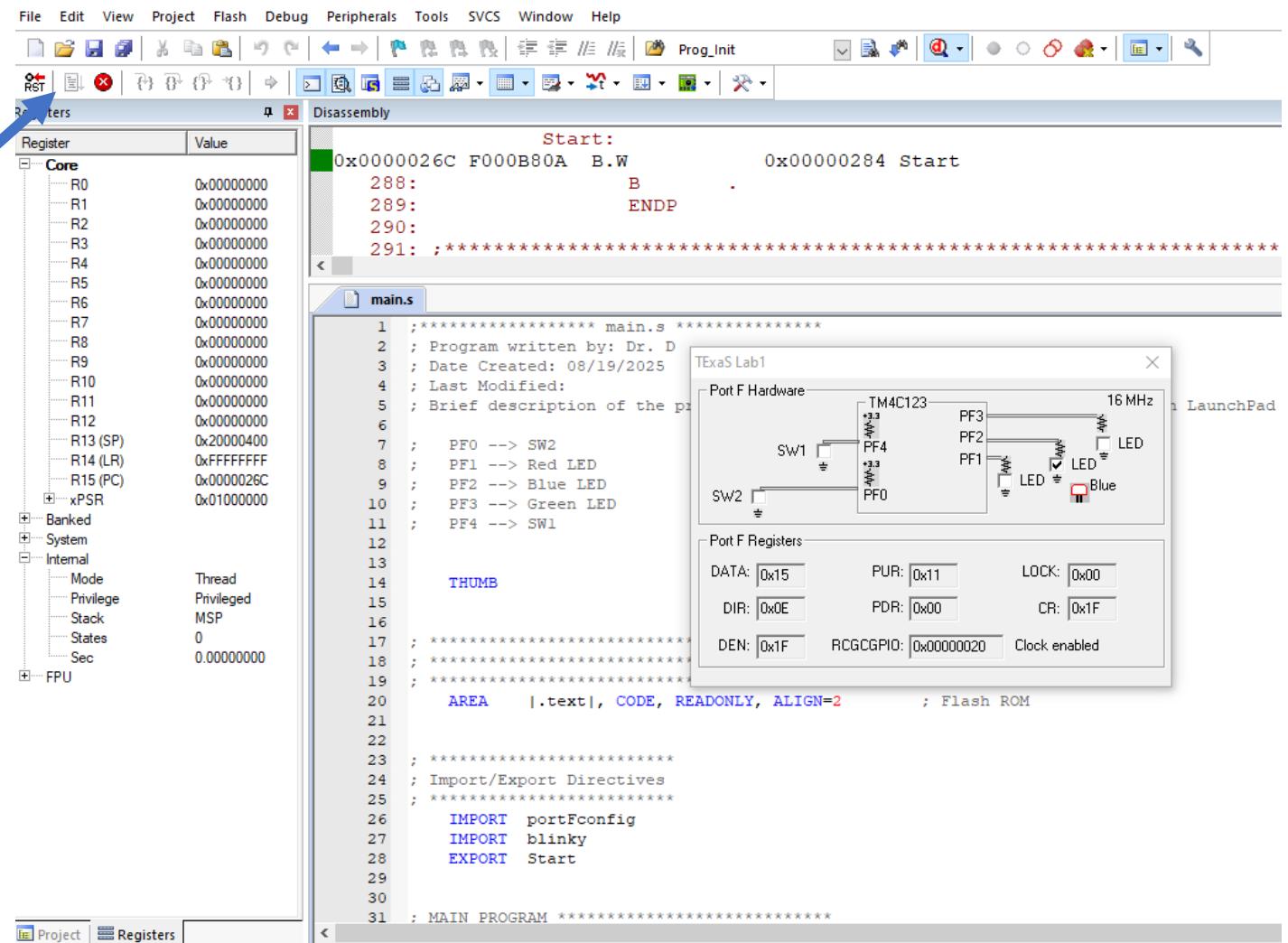
# Software Simulator Demo

- verify that you have selected “**Periodic Window Update**” in the View menu
- you won’t see anything happening if you don’t



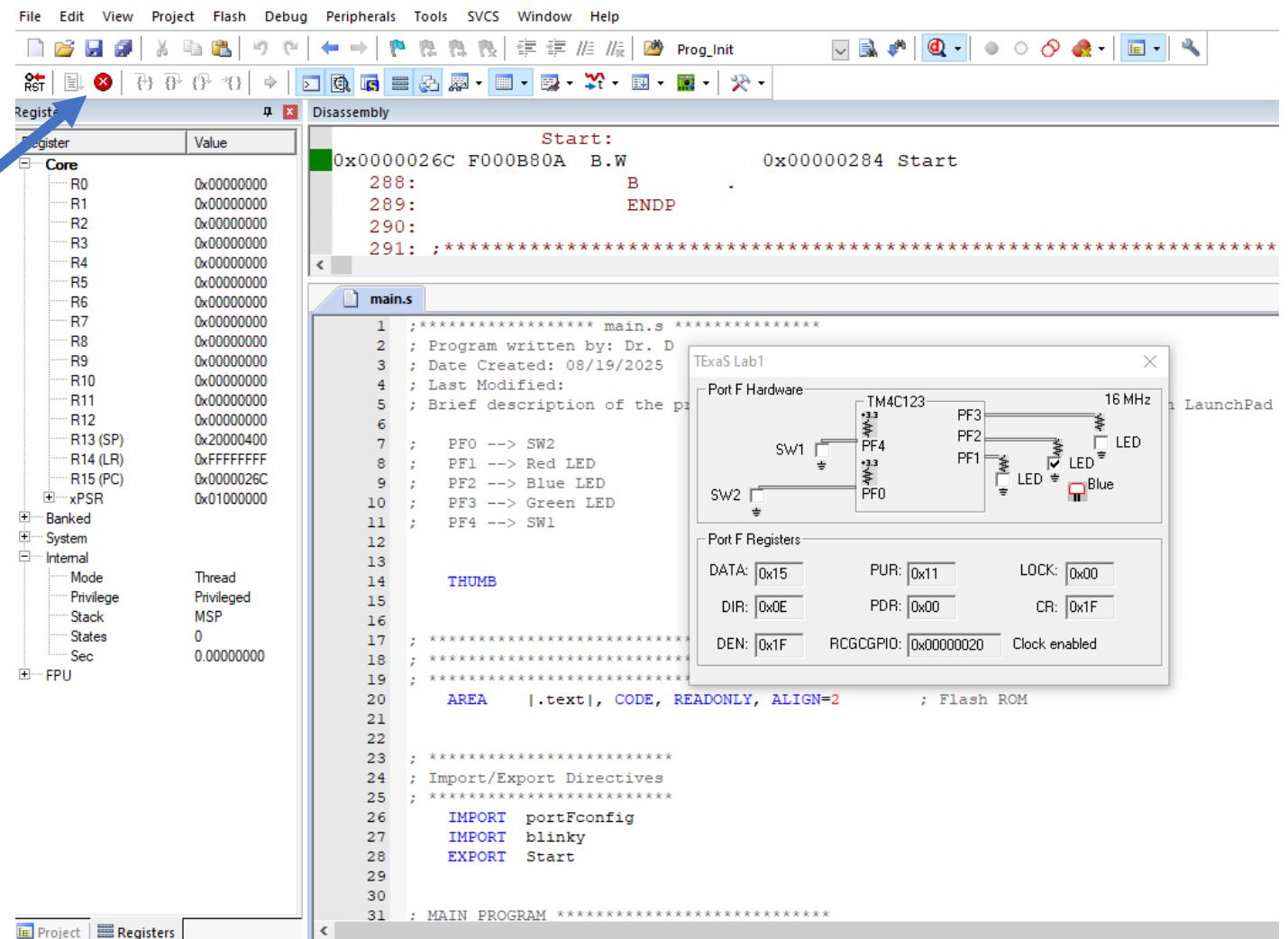
# Software Simulator Demo

- click the **run** icon or use the **F5** key
- you should see each of the LEDs (red, blue, green) cycle and repeat



# Software Simulator Demo

- hit **stop icon** and then the debug icon again to get out of simulation mode

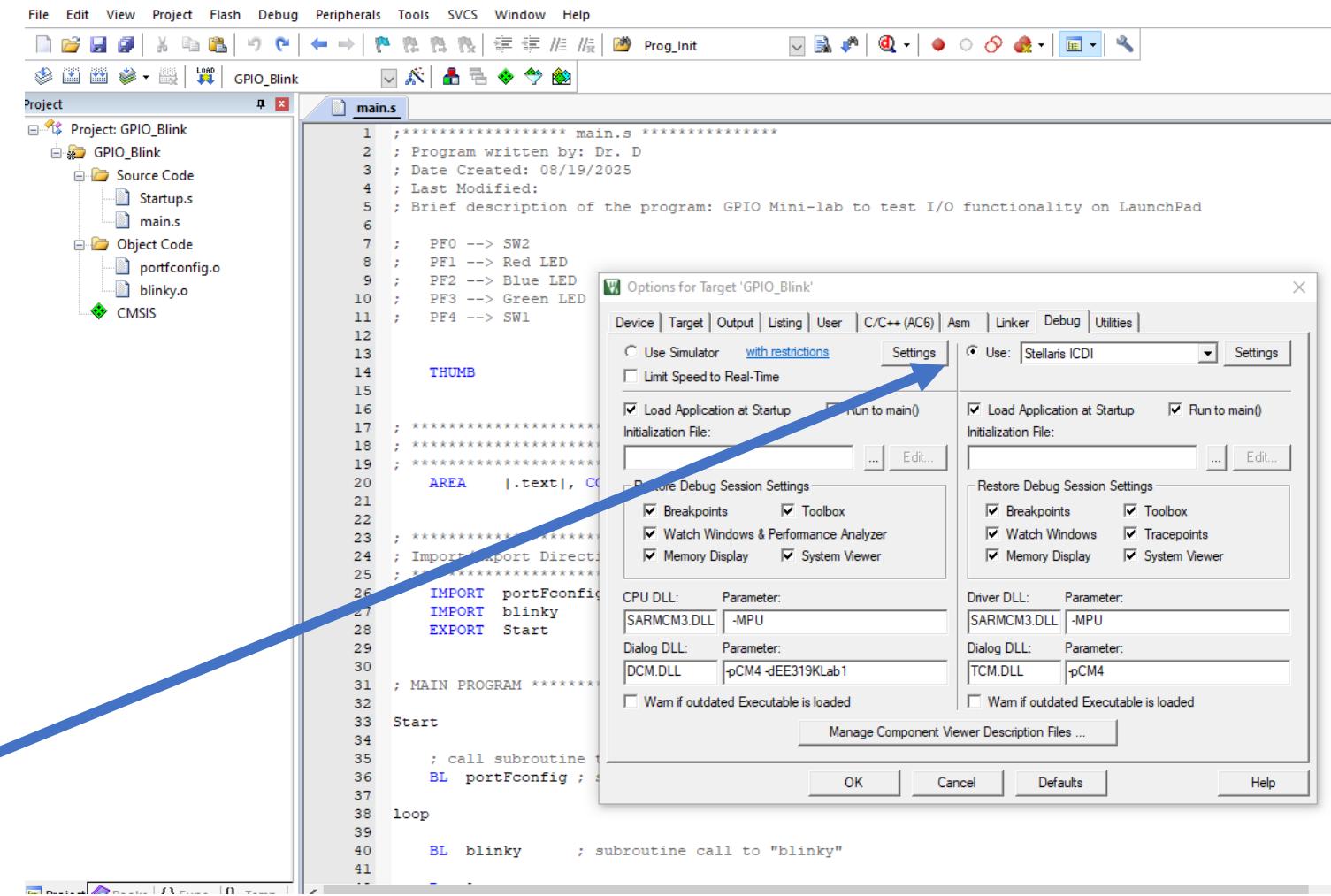


## Step 6: Test Your Hardware

# Hardware Demo

- to test your hardware, switch from simulation mode to the hardware mode
- first, bring up the “Options for Target” dialog box from the Project menu
- then select “Use Stellaris ICDI”

You cannot do this step until you have properly installed the Stellaris drivers! (and the patch)



# Hardware Demo

- **LOAD** your ASSEMBLED program to LaunchPad
- of course, your LaunchPad must be plugged in and powered on



The screenshot shows the Keil MDK-ARM IDE interface. The top menu bar includes File, Edit, View, Project, Flash, Debug, Peripherals, Tools, SVCS, Window, and Help. The toolbar contains various icons for file operations and project management. A red arrow points from the "LOAD" button icon on the left towards the "LOAD" button in the toolbar of the IDE.

The main window displays a project titled "GPIO\_Blink" with a sub-project "GPIO\_Blink". The "Source Code" folder contains "Startup.s" and "main.s". The "Object Code" folder contains "portconfig.o" and "blinky.o". The "CMSIS" folder is also present. The "main.s" assembly code is shown in the editor:

```
1 ;***** main.s *****
2 ; Program written by: Dr. D
3 ; Date Created: 08/19/2025
4 ; Last Modified:
5 ; Brief description of the program: GPIO Mini-lab to test I/O functionality on LaunchPad
6
7 ; PF0 --> SW2
8 ; PF1 --> Red LED
9 ; PF2 --> Blue LED
10 ; PF3 --> Green LED
11 ; PF4 --> SW1
12
13 THUMB
14
15
16
17 ; **** CODE AREA IN ROM ****
18 ; **** Import/Export Directives ****
19 ; ****
20 AREA .text, CODE, READONLY, ALIGN=2 ; Flash ROM
21
22
23 ; **** Import/Export Directives ****
24 ; Import/Export Directives
25 ; ****
26 IMPORT portConfig
27 IMPORT blinky
28 EXPORT Start
29
30
31 ; MAIN PROGRAM ****
32 Start
33
34 ; call subroutine to set up I/O Port F
35 BL portConfig ; subroutine call to portConfig
36
37 loop
38
39 BL blinky ; subroutine call to "blinky"
40
41
```

The bottom status bar shows the build output:

```
Project Books | 0 Func... | 0 Temp...
Build Output
assembling Startup.s...
assembling main.s...
linking...
Program Size: Code=924 RO-data=0 RW-data=0 ZI-data=1024
".\GPIO_Blink.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed: 00:00:01
```

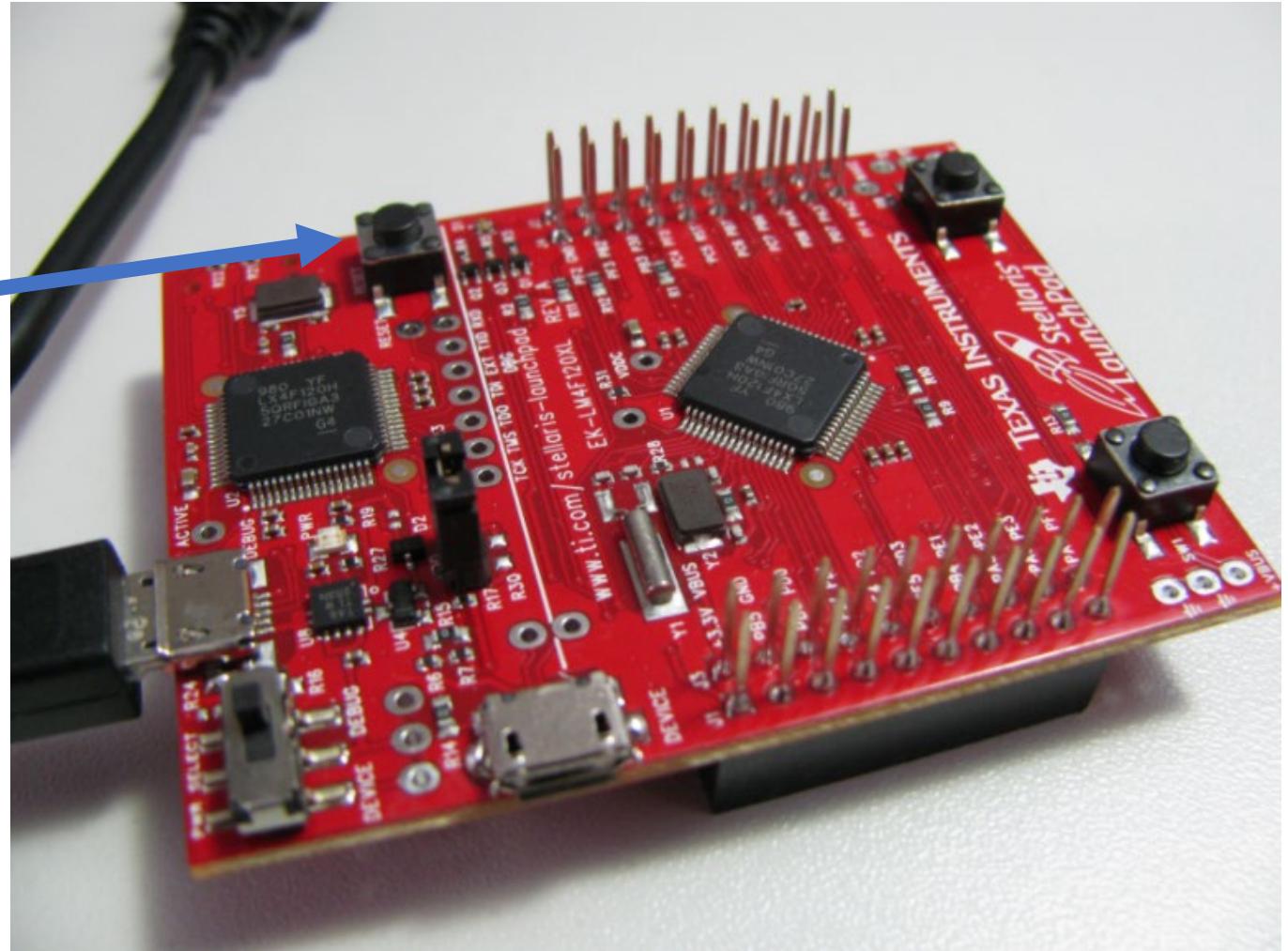
## Hardware Demo

- You should see a message telling you that your program was successfully downloaded to the LaunchPad.

```
Connecting: Mode=JTAG, Speed=1000000Hz
Erase Done.
Programming Done.
Verify OK.
Flash Load finished at 17:23:37
```

## Hardware Demo

- hit the **reset** button on the LaunchPad
- Now sit back and enjoy!



# Keil uVision Resources



# Keil User's Guide

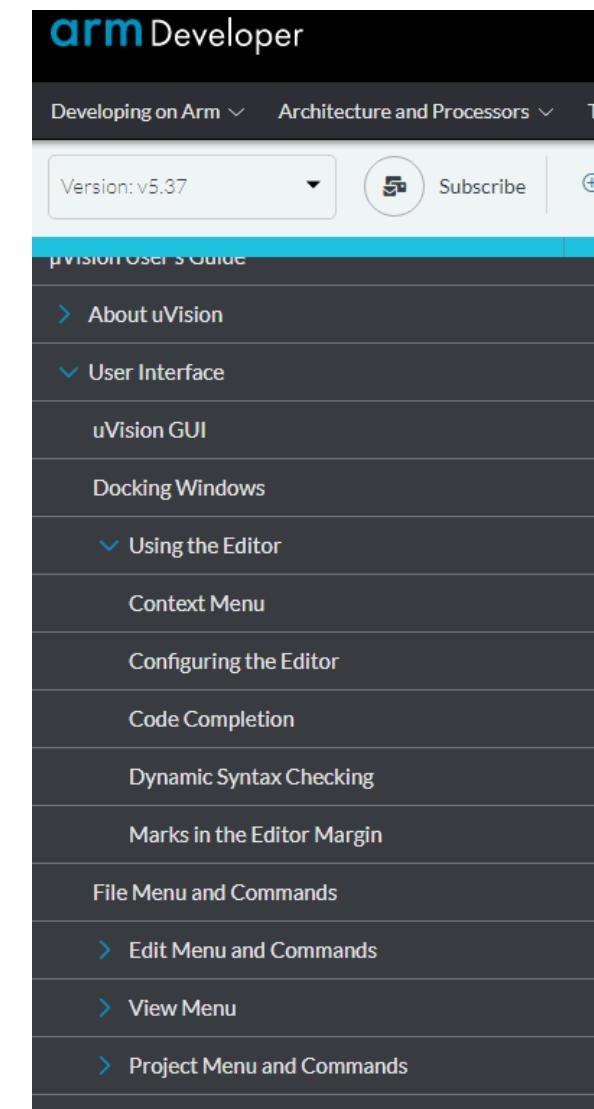
- <https://developer.arm.com/documentation/101407/latest/>

The screenshot shows the arm Developer website with the following details:

- Header:** arm Developer, Developing on Arm, Architecture and Processors, Tools and Software.
- Breadcrumbs:** Home / Documentation / Tools and Software / Keil Products / Keil MDK
- Title:** µVision User's Guide
- Version:** v5.37
- Actions:** Subscribe, Search within this document
- Table of Contents:** DOCUMENT TABLE OF CONTENTS
  - Back to search
  - All Keil MDK Documentation
  - µVision User's Guide
    - About uVision
    - User Interface
    - Creating Applications
    - Debugging
    - Debug Commands
    - Debug Functions
    - Simulation
    - Flash Programming
- Section:** µVision User's Guide Version 5.37 About this book
- Description:** This User's Guide describes the µVision® IDE & Debugger. It covers the creation of applications for ARM-based microcontrollers and embedded systems.
- List:**
  - **About µVision** describes main features, the folder structure, and how to start the IDE.
  - **User Interface** describes the IDE interface with toolbars, menus, and windows.
  - **Creating Applications** describes the creation of applications, including source code. This chapter includes a section with advanced topics.
  - **Debugging** describes the µVision debugger, the advanced debugging techniques for accessing memory, and how to use breakpoints and watch windows.

# Keil Editor Marks

- <https://developer.arm.com/documentation/101407/0537/User-Interface/Using-the-Editor/Marks-in-the-Editor-Margin>



The screenshot shows a web browser displaying the **arm Developer** website. The main content area is titled "Marks in the Editor Margin". Below the title, there is a section titled "Edit Mode" which contains two bullet points: one for a warning icon and one for an error icon. To the left of the main content, a sidebar lists various topics under "uVision User's Guide", including "About uVision", "User Interface", "uVision GUI", "Docking Windows", "Using the Editor", "Context Menu", "Configuring the Editor", "Code Completion", "Dynamic Syntax Checking", "Marks in the Editor Margin", "File Menu and Commands", "Edit Menu and Commands", "View Menu", and "Project Menu and Commands".

## Marks in the Editor Margin

The editor supports coding and debugging through visual marks that are displayed in the margin.

### Edit Mode

- **Warning**. Place the mouse cursor over the icon to view details.
- **Error**. Place the mouse cursor over the icon to view details.

### Edit Mode and Debug Mode

- **Enabled breakpoint**. Breakpoints can be set or removed by clicking into the left margin.
- **Disabled breakpoint**.
- **No Code at Breakpoint**. A breakpoint could not be set because there is no code at that location.
- **Bookmark**.
- **Current cursor line**.

### Debug Mode

- **Next execution statement**. This icon is used in the editor.
- **Next execution statement**. This icon is used in the Disassembly window.