arm DevSummit

OCTOBER 19-21, 2021

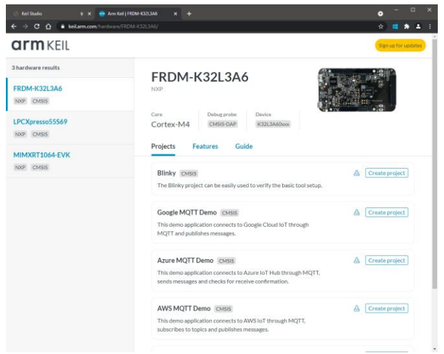
Introducing Keil Studio Cloud

Introducing Keil Studio and cloud-based development for IoT and Embedded applications

Christopher Seidl, Ronan Synnott
Arm

Seamless Support for All Development Phases

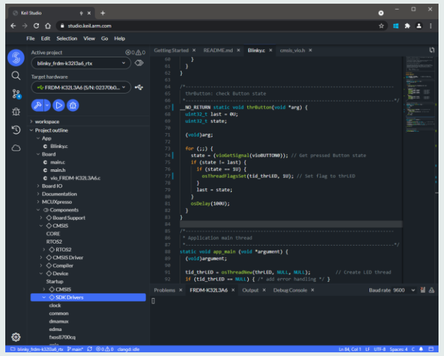
DISCOVER POSSIBILITIES



Enter parameters of your application

- Compare devices
- Evaluation boards
- Reference code examples

EXPLORE REFERENCE DESIGNS



Use online tools for testing

- Explore code
- Zero installation hassle
- Always up to date

DEVELOP APPLICATION

Download reference code and use classic tooling

- Develop and verify custom application functionality
- Extend software framework with additional functionality

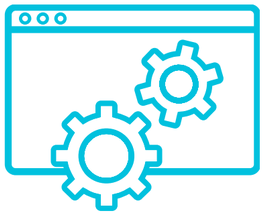
DEPLOY TO CUSTOM DESIGN

Optimize application for mass production

- Retarget device pinout
- Verify system behavior
- Analyze power consumption

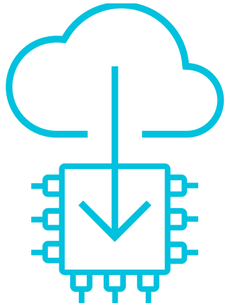
Key advantages of cloud-based embedded development

Software as a Service



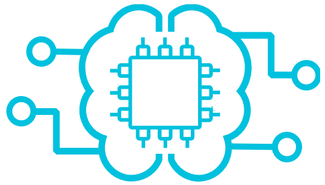
Zero-installation environments, already setup by the provider. Use multiple development platforms with pre-built example to test performance.

Software Updates



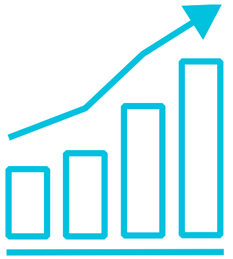
Over-the-air (OTA) programming offers methods to provision and update software of devices that are already in the field.

Data Analytics



Monitor devices to spot anomalies and collect training data for ML algorithms that can be deployed to IoT endpoints.

Continuous Integration



A server running in the cloud contains a tool environment with simulation models and settings specific to your project.

Learn more » www.arm.com/blogs/blueprint/cloud-based-embedded-development

Workflow for CI: Develop Application Code or Test Cases

Flexible workflows addresses the needs of every developer

CI hosted in the Cloud

GitHub – Runners

Commit triggers **GitHub Actions** that starts CI using Arm Compiler for build and/or Arm VHT for testing.

GitHub Commit

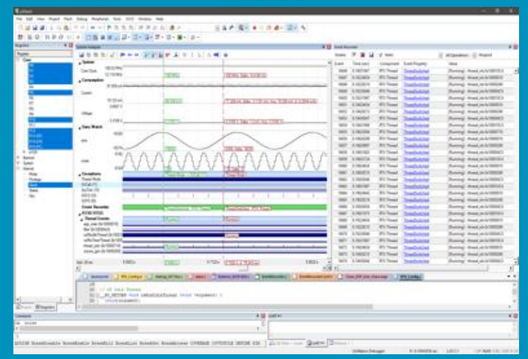
Test Results

Jenkins › CMSIS RTOS2 Validation ›

S	W	Name ↓	Last Success	Last Failure	Last Duration	Fav	# Issues	Robo
●	☔	0. Commit Pipeline	19 days - #159	3 hr 27 min - #168	6 min 57 sec	☆	-	
●	☔	0. Nightly Pipeline	7 days 17 hr - #478	17 hr - #488	1 hr 14 min	☆	-	
●	☁	1. GenPack	4 days 22 hr - #161	3 hr 27 min - #166	12 sec	☆	-	
●	☔	2. Prepare	3 days 10 hr - #1891	10 hr - #1900	7 min 10 sec	☆	-	
●	☔	3.1 Build RTX5 Validation	7 days 16 hr - #1240	3 days 15 hr - #1242	51 min	☆	110	
●	☀	3.2 Build FreeRTOS Validation	3 days 10 hr - #828	N/A	6 min 19 sec	☆	530	
●	☀	4.1 Run RTX5 Validation	8 days 15 hr - #1186	N/A	12 min	☆	-	
●	⛅	4.2 Run FreeRTOS Validation	4 days 10 hr - #792	3 days 10 hr - #793	4 min 22 sec	☆	-	

Cloud flow with IDE in Browser

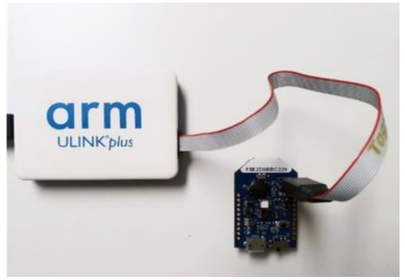
Classic Tools on Desktop (MDK, DS)



Hardware boards on your Desk

Develop Test cases

Deploy to custom hardware



All environments generate Event Log files for off-line analysis

Evaluation Board

MPS3 with FPGA image

Simulation

Target Hardware

Introducing Keil Studio Cloud

- Steps to get started with Keil Studio:
  - **search for and find** relevant examples for the NXP IMXRT1050-EVKB.
  - **import** and **build** an example project.
  - **deploy** the project to the target.
  - **debug** the application via the web browser.

- Example projects:
  - Import a **Blinky** example to get started.
  - Open a **TFLu MicroSpeech** application from GitHub.
  - Run an **MQTT demo application** connecting to AWS.

Please make sure to finish the pre-work first »
github.com/MDK-Packs/KeilStudioWorkshopDevSummit21

arm DevSummit

Getting Started

Key take-aways

In this part you will learn how to:

- Find and import a project from keil.arm.com.
- Use Keil Studio.
- Debug an application on target hardware connected to your PC.

Arm Keil | Cloud-Based Develop

keil.arm.com

arm KEIL

Hardware

Documentation

Keil Studio

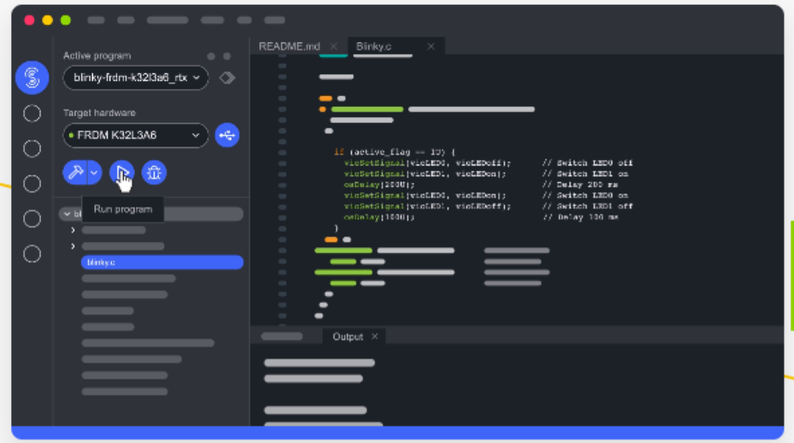
Open Beta

Keil Studio Cloud

Introducing Keil Studio Cloud, a browser-based IDE for IoT, ML and embedded development. Accelerate your next project with zero-installation tools, ready-to-run examples, git integration and web debugging.

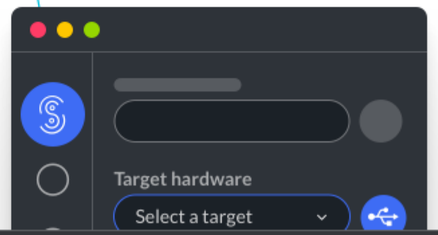
Join the Beta

What's next? See our roadmap



Go to keil.arm.com

A Zero-Installation Environment That Runs in Your Browser



Show all

Arm Keil | Cloud-Based Develop

keil.arm.com

arm KEIL Hardware Documentation Keil Studio

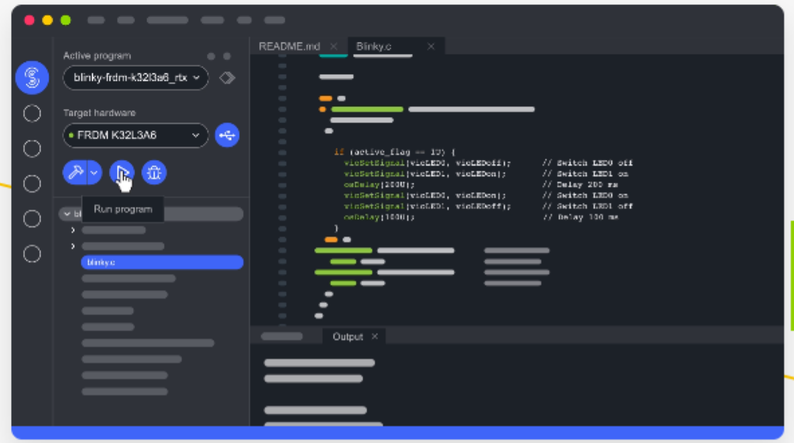
Open Beta

Keil Studio Cloud

Introducing Keil Studio Cloud, a browser-based IDE for IoT, ML and embedded development. Accelerate your next project with zero-installation tools, ready-to-run examples, git integration and web debugging.

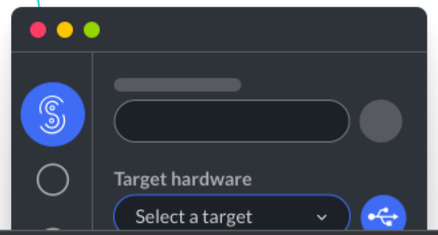
Join the Beta

What's next? See our roadmap



Click on **Hardware** to find the board pages

A Zero-Installation Environment That Runs in Your Browser



Show all

Arm Keil | STK_HC32F120_LQFP4

keil.arm.com/boards/hdsc-stk-hc32f120-lqfp44-080-v10-fc8f9c1/

arm KEIL Hardware Documentation Keil Studio

Hardware (408)

Search by name or vendor

Only show boards with example projects

STK_HC32F120_LQFP44_080_V10
HDSC

STK_HC32M120_LQFP48_050_V11
HDSC

STM32F334-Discovery
STMicroelectronics

STM32L4R9I-EVAL
STMicroelectronics

In the search box, enter “1050” and then select the **EVKB-IMXRT1050_MDK**

STK_HC32F120_LQFP44_080_V10

HDSC

Core	Debug probe	Device
Cortex-M0+	JTAG/SW	HC32F120H8TA

Projects Features Documentation

We don't have any projects for this board yet.

Arm Keil | EVKB-IMXRT1050_MDK

keil.arm.com/boards/nxp-evkb-imxrt1050-mdk-rev-a1-126523e/

arm KEIL Hardware Documentation Keil Studio

Hardware (2)

1050

Only show boards with example projects

EVKB-IMXRT1050
NXP

EVKB-IMXRT1050_MDK
NXP

EVKB-IMXRT1050_MDK Rev. A1

NXP

Core	Debug probe	Device
Cortex-M7	JTAG/SW JTAG/SW	MIMXRT1052DVL6B

Projects Features Documentation

AWS MQTT Demo RTX Ethernet Socket (MW-Network) — Create project
Demonstrates the subscribe-publish workflow of MQTT

AWS MQTT Demo FreeRTOS — Create project
Demonstrates the subscribe-publish workflow of MQTT

Azure MQTT Demo ESP8266 RTX WiFi Socket — Create project
Demonstrates the subscribe-publish workflow of MQTT

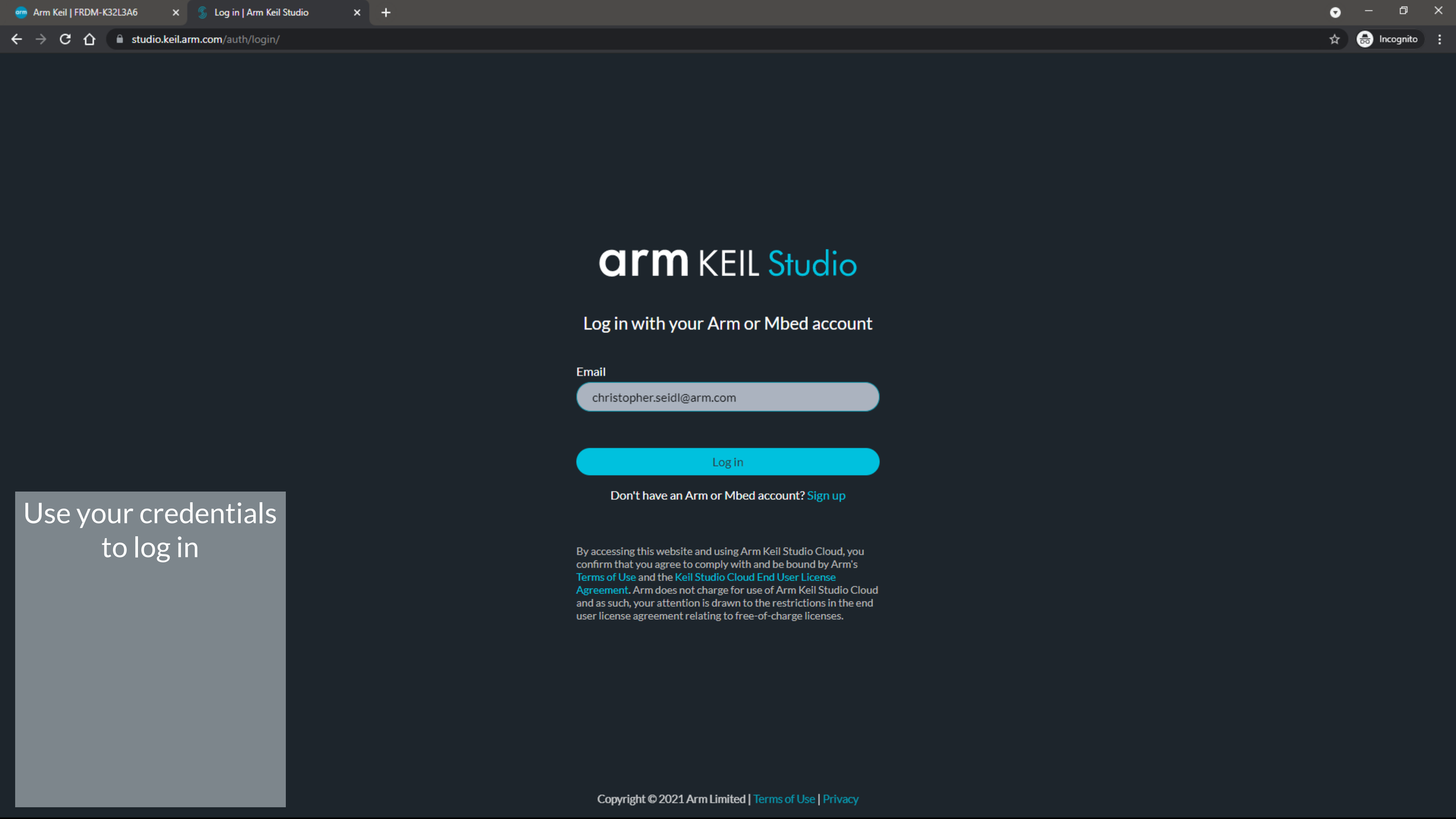
Blinky RTX — Create project
Simple example

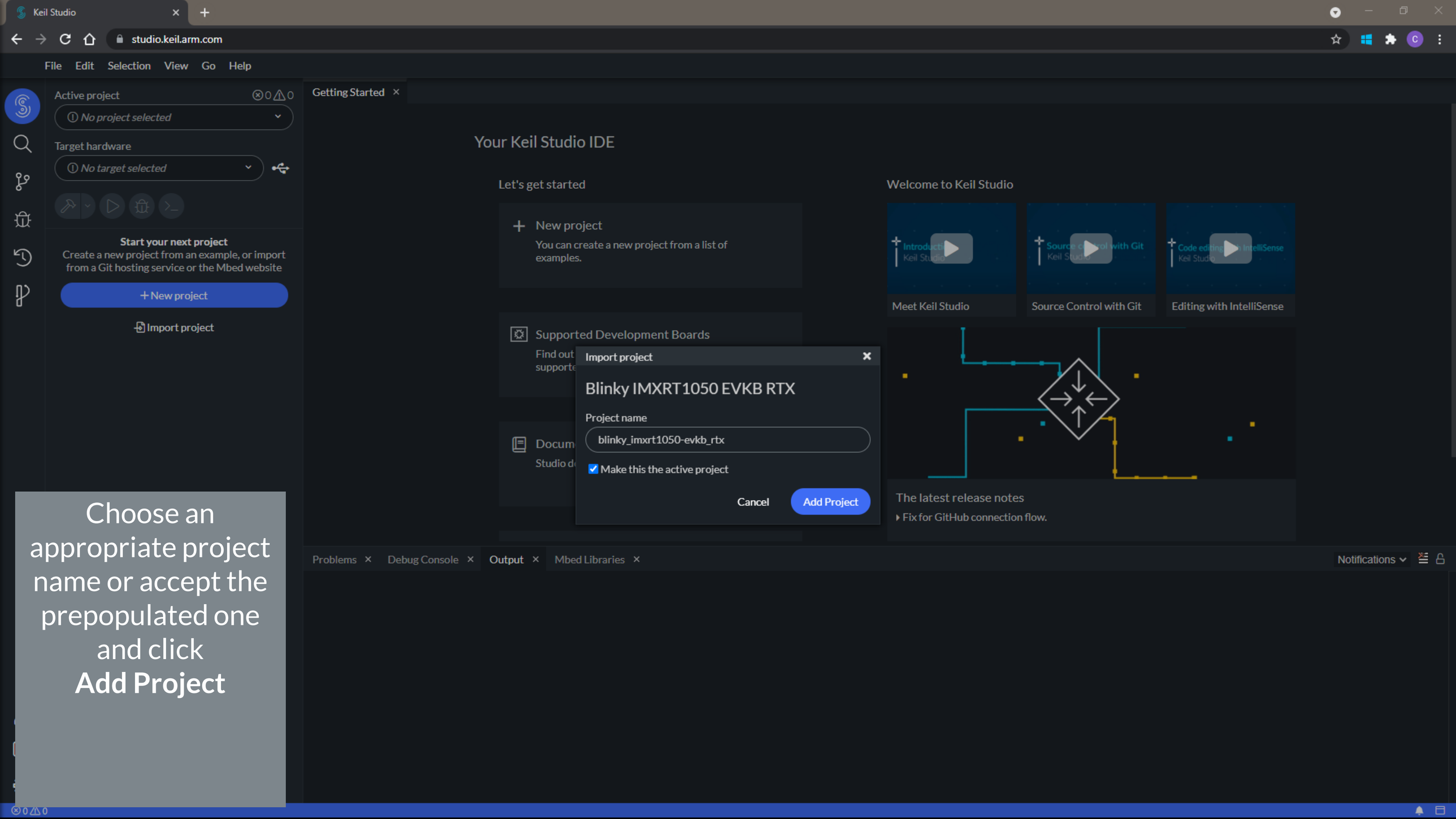
Google MQTT Demo ESP8266 FreeRTOS WiFi Socket — Create project
Demonstrates the subscribe-publish workflow of MQTT

Paho MQTT Demo ESP8266 FreeRTOS WiFi Socket — Create project

With the Blinky project, click **Create project** to import it into Keil Studio

https://studio.keil.arm.com/?cmsis-import-url=https%3A%2F%2Fdownloads.software.api.keil.arm.com%2Fprojects%2F9...


Arm Keil | FRDM-K32L3A6
Log in | Arm Keil Studio
studio.keil.arm.com/auth/login/
Incognito
arm KEIL Studio
Log in with your Arm or Mbed account
Email
christopher.seidl@arm.com
Log in
Don't have an Arm or Mbed account? Sign up
By accessing this website and using Arm Keil Studio Cloud, you confirm that you agree to comply with and be bound by Arm's Terms of Use and the Keil Studio Cloud End User License Agreement. Arm does not charge for use of Arm Keil Studio Cloud and as such, your attention is drawn to the restrictions in the end user license agreement relating to free-of-charge licenses.
Copyright © 2021 Arm Limited | Terms of Use | Privacy
Use your credentials to log in

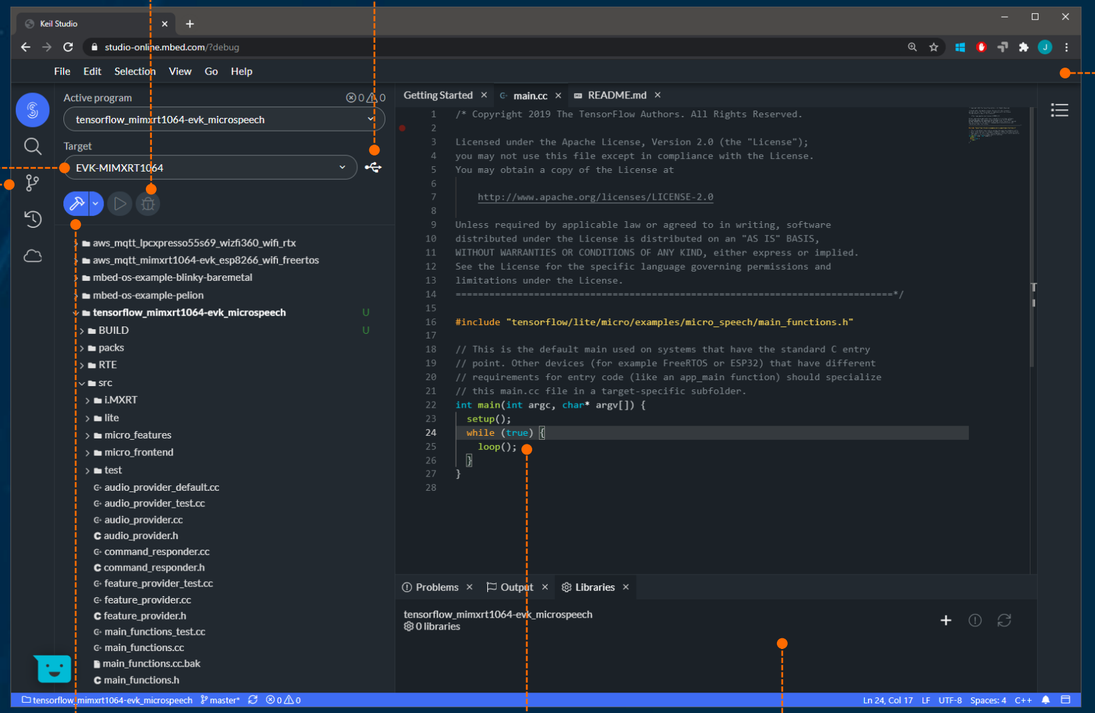

Keil Studio
studio.keil.arm.com
File Edit Selection View Go Help
Active project
No project selected
Target hardware
No target selected
Start your next project
Create a new project from an example, or import from a Git hosting service or the Mbed website
+ New project
Import project
Getting Started
Your Keil Studio IDE
Let's get started
New project
You can create a new project from a list of examples.
Supported Development Boards
Welcome to Keil Studio
Meet Keil Studio
Source Control with Git
Editing with IntelliSense
Import project
Blinky IMXRT1050 EVKB RTX
Project name
blinky_imxrt1050-evkb_rtx
Make this the active project
Cancel
Add Project
The latest release notes
Fix for GitHub connection flow.
Problems
Debug Console
Output
Mbed Libraries
Notifications
Choose an appropriate project name or accept the prepopulated one and click
Add Project

Overview

Debug
Keil Studio Cloud flashing and debug utilises Keil MDK (CMSIS) flashing algorithms and DS debugger RDDI layer.

Detection
Keil Studio device and debug probe detection is undertaken via listing and interrogation of local devices via WebUSB.

Board and MCU data
Board and device data is provided by Arm's **ecosystem API** and is underpinned by CMSIS and Mbed OS data.



IDE
Built on the open source Theia IDE for desktop and browser deployment, and incorporating the Monaco editor.

Source control
Easy integration with CI workflows and source control via git (best with GitHub).

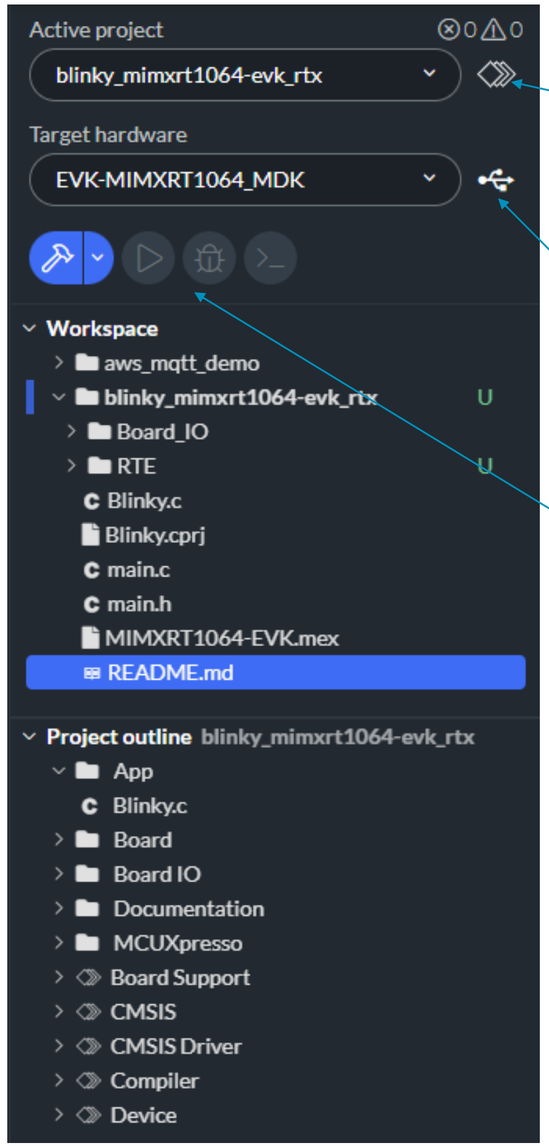
Build
Keil Studio Cloud uses a hosted build service, allowing software to be built rapidly using Arm Compiler 6 for small, efficient binaries.

IntelliSense
Clangd is used to generate the C/C++ IntelliSense for autocompletion and syntax hinting.

Runtime
Software components (from CMSIS-Packs) are configured and managed through a hosted runtime environment in Keil Studio Cloud, and locally for Keil Studio desktop.

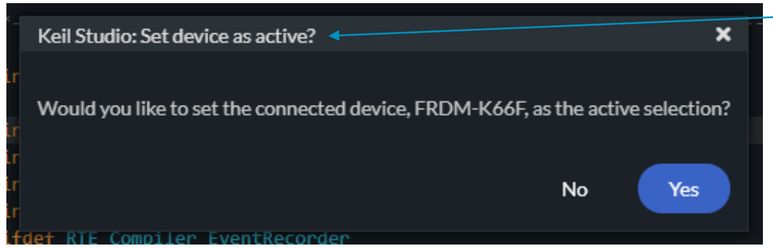
Project explorer

- The project explorer is the control panel for your projects and allows you to select the **active project** (the project that will be built/flashed etc)
- You can also select the active project from the context menu (right click)
- The project explorer shows you errors and warnings for your project
- **Target hardware** (e.g. a development board, custom hardware or MCU) can be selected, but CPRJ format supports only one compatible development board per project
- **Project outline** shows you the logical outline specified in the underlying project file, including the software components used in the project. Keil Studio Cloud uses the new CPRJ format
- **Workspace** lists your folders and files. A user can choose to reveal hidden files (such as workspace settings or project files that are otherwise not visible)

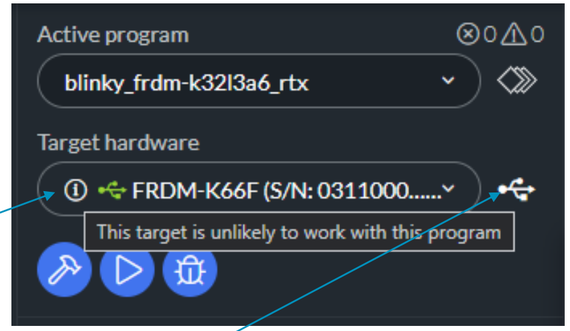


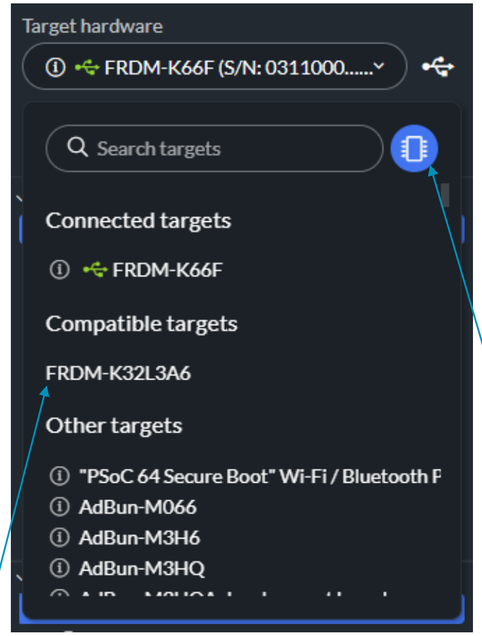
- **Manage software components** launches a panel showing the runtime environment for your application (e.g. CMSIS APIs, software components and associated settings). This is read only in the initial release of Keil Studio Cloud, with the ability to modify software components coming soon
- **USB icon** launches the ability for a supported browser to pair with a development board connected over USB. This launches a native Chrome window to select
- **Build, run and debug** the active project using these controls. Options are enabled or disabled based on the target hardware and whether it is plugged in. You can also open the serial window.

Target hardware



Supported boards that are already paired appear with a modal when you plug the board in allowing you to select it for the active project



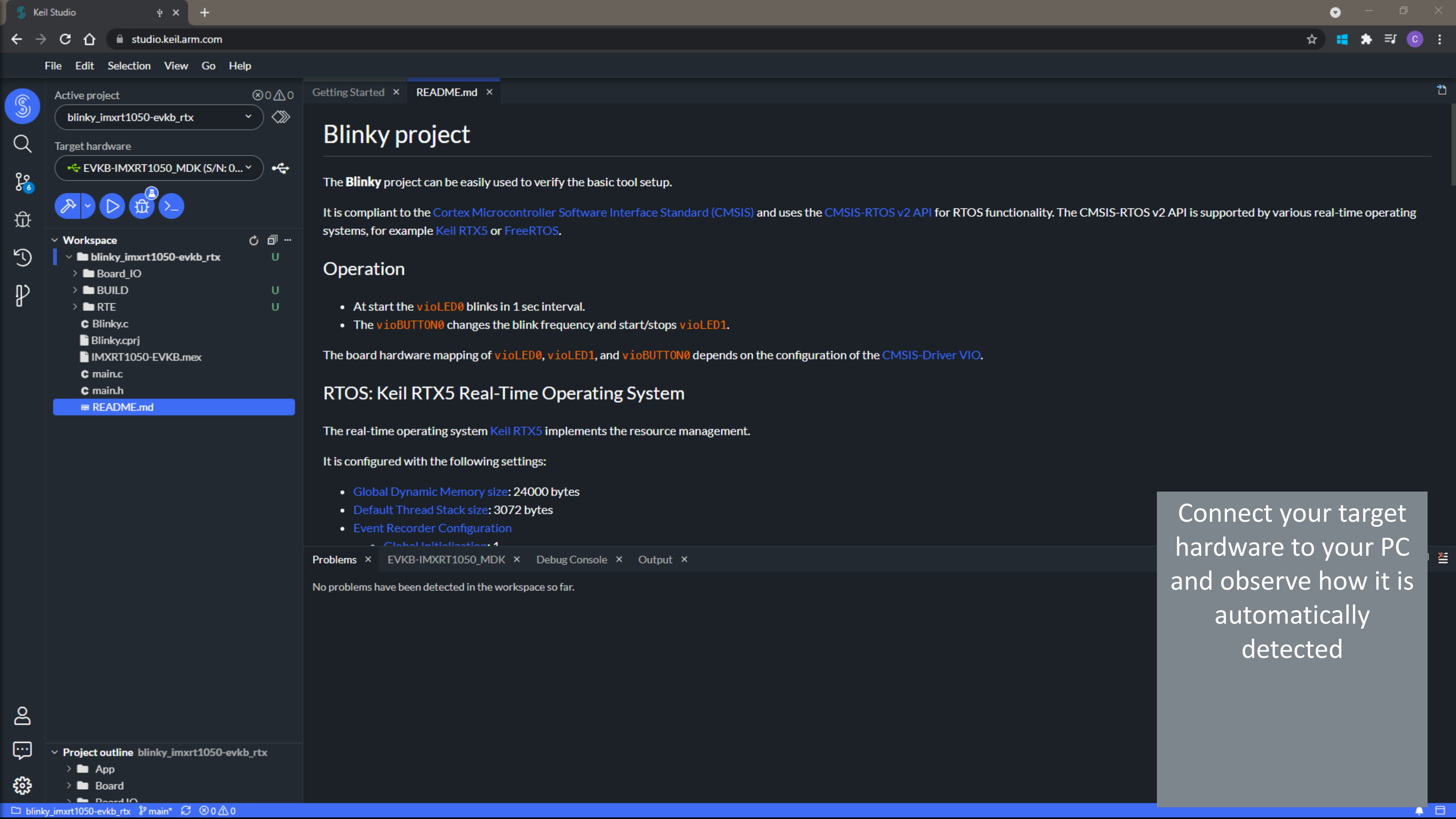


Warnings or information (such as lack of compatibility) are indicated by i

Pair with a supported board using the USB icon and selecting CMSIS-DAP

For CMSIS/CPRJ only the compatible target will result in a successful build/run/debug operation

Selecting the MCU icon allows you to configure a custom target – shown here is an example of the on-board DAPLink device, but the ULINKplus debug probe is also valid


Keil Studio
studio.keil.arm.com
File Edit Selection View Go Help
Active project
blinky_imxrt1050-evkb_rtx
Target hardware
EVKB-IMXRT1050_MDK (S/N: 0...
Workspace
blinky_imxrt1050-evkb_rtx
Board_IO
BUILD
RTE
Blinky.c
Blinky.cprj
IMXRT1050-EVKB.mex
main.c
main.h
README.md
Project outline blinky_imxrt1050-evkb_rtx
App
Board
Getting Started
README.md
Blinky project
The Blinky project can be easily used to verify the basic tool setup.
It is compliant to the Cortex Microcontroller Software Interface Standard (CMSIS) and uses the CMSIS-RTOS v2 API for RTOS functionality. The CMSIS-RTOS v2 API is supported by various real-time operating systems, for example Keil RTX5 or FreeRTOS.
Operation
At start the vioLED0 blinks in 1 sec interval.
The vioBUTTON0 changes the blink frequency and start/stops vioLED1.
The board hardware mapping of vioLED0, vioLED1, and vioBUTTON0 depends on the configuration of the CMSIS-Driver VIO.
RTOS: Keil RTX5 Real-Time Operating System
The real-time operating system Keil RTX5 implements the resource management.
It is configured with the following settings:
Global Dynamic Memory size: 24000 bytes
Default Thread Stack size: 3072 bytes
Event Recorder Configuration
Problems
EVKB-IMXRT1050_MDK
Debug Console
Output
No problems have been detected in the workspace so far.
Connect your target hardware to your PC and observe how it is automatically detected
blinky_imxrt1050-evkb_rtx
main*

CMSIS support

- Support for the CMSIS framework is currently limited to read-only, allowing users to create pre-defined projects based on a range of reference designs
- Each project lists the required software components which are supplied to the IDE and build system by a cache of **CMSIS-Packs**
- A user can view the included software components, configure them and view documentation, but they cannot add or remove software components (this will be a next step)

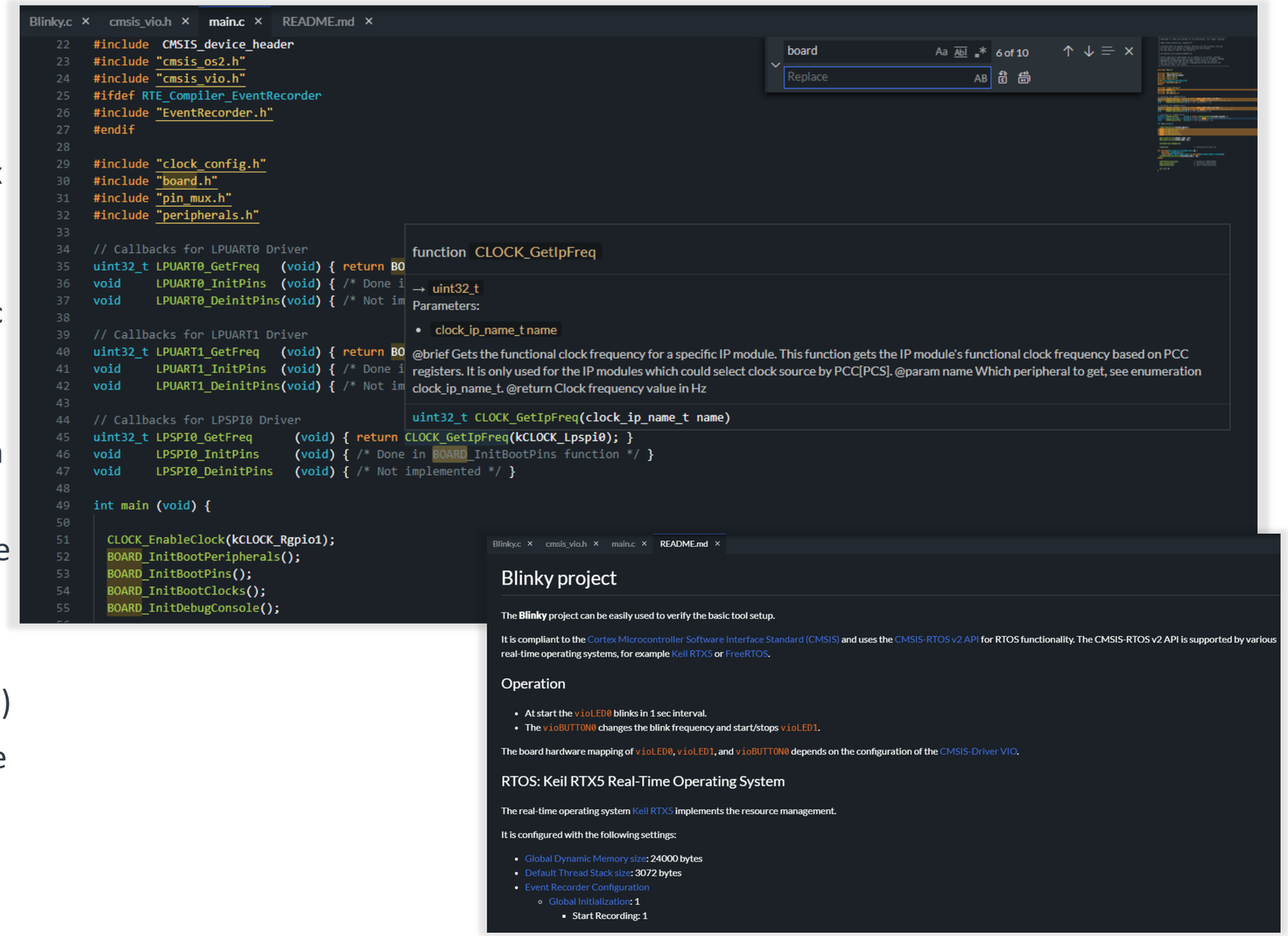
Dependencies of the application are listed, along with vendor, variant and version

Components can be expanded to view documentation and further details

Filter search through components

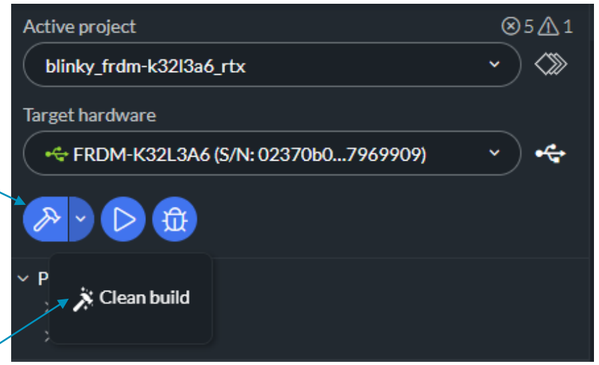
Editor

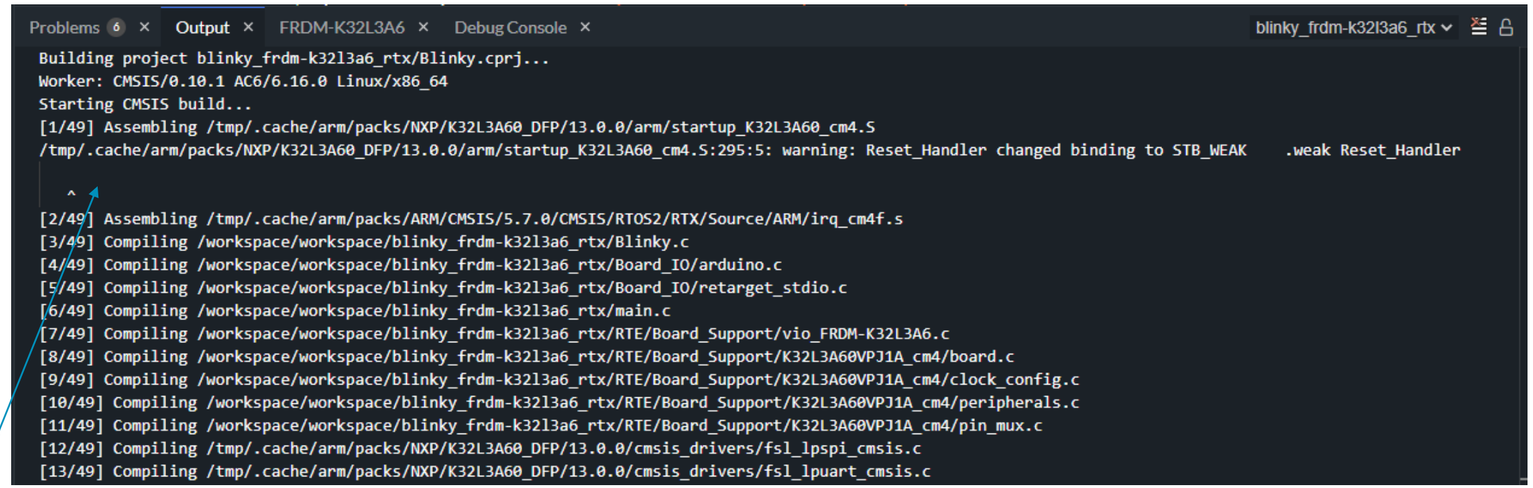
- Keil Studio is using the Monaco editor, developed by Microsoft for VSCode
- Code can be written with API hinting, syntax support for C/C++ and linting (which can be customised)
- Hover over includes, functions, variables etc in order to peek at their definition or docs. Click through to open referenced files
- Editor supports preview mode of markdown for readme or other content
- Editor minimap shows overall code structure
- Loops and functions can be collapsed for readability/clarity
- Code can be refactored (e.g. expand macros)
- In-file find and replace, and workspace-wide search are supported
- User can set preferences for editor (tabs, spaces etc)



Compiling projects

- Compilation is intended to be simple in Keil Studio Cloud – there is limited control over the flags and settings for the compiler
- We use a compiler service to build your project. The project files and dependencies are sent to a cloud based Arm Compiler 6, which returns a binary to your browser
- The default is to provide incremental builds, but the dropdown next to the build icon can be used to select clean builds
- The binary can then be flashed to compatible hardware targets, or downloaded (either use drag + drop programming or a partner programmer)
- At the moment, the compiler flags are taken from the CPRJ project file for CMSIS and are predefined
- Build output is displayed in the Output panel, which passes the information from the build system, including warnings and errors





Keil Studio

studio.keil.arm.com

File Edit Selection View Go Help

Active project

blinky_imxrt1050-evkb_rtx

Target hardware

EVKB-IMXRT1050_MDK (S/N: 0...

Workspace

- blinky_imxrt1050-evkb_rtx U
  - Board_IO
  - BUILD U
  - RTE U
  - Blinky.c
  - Blinky.cprj
  - IMXRT1050-EVKB.mex
  - main.c
  - main.h
  - README.md

Project outline blinky_imxrt1050-evkb_rtx

- App
- Board

Getting Started | README.md

Blinky project

The **Blinky** project can be easily used to verify the basic tool setup.

It is compliant to the Cortex Microcontroller Software Interface Standard (CMSIS) and uses the CMSIS-RTOS v2 API for RTOS functionality. The CMSIS-RTOS v2 API is supported by various real-time operating systems, for example Keil RTX5 or FreeRTOS.

Operation

- At start the `vioLED0` blinks in 1 sec interval.
- The `vioBUTTON0` changes the blink frequency and start/stops `vioLED1`.

The board hardware mapping of `vioLED0`, `vioLED1`, and `vioBUTTON0` depends on the configuration of the CMSIS-Driver VIO.

RTOS: Keil RTX5 Real-Time Operating System

The real-time operating system Keil RTX5 implements the resource management.

It is configured with the following settings:

- Global Dynamic Memory size: 24000 bytes

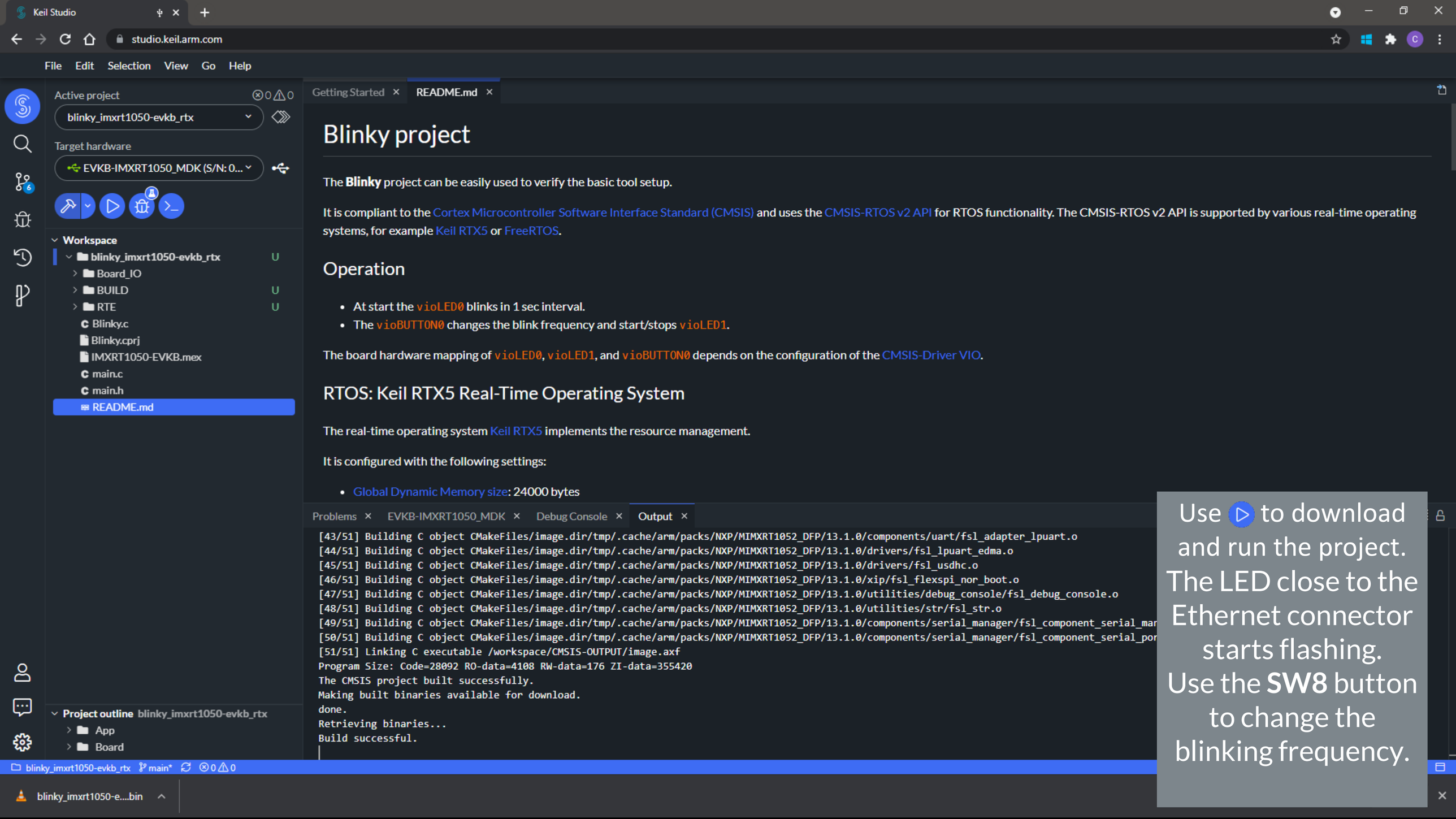
Problems | EVKB-IMXRT1050_MDK | Debug Console | Output

```
[43/51] Building C object CMakeFiles/image.dir/tmp/.cache/arm/packs/NXP/MIMXRT1052_DFP/13.1.0/components/uart/fsl_adapter_lpuart.o
[44/51] Building C object CMakeFiles/image.dir/tmp/.cache/arm/packs/NXP/MIMXRT1052_DFP/13.1.0/drivers/fsl_lpuart_edma.o
[45/51] Building C object CMakeFiles/image.dir/tmp/.cache/arm/packs/NXP/MIMXRT1052_DFP/13.1.0/drivers/fsl_usdhc.o
[46/51] Building C object CMakeFiles/image.dir/tmp/.cache/arm/packs/NXP/MIMXRT1052_DFP/13.1.0/xip/fsl_flexspi_nor_boot.o
[47/51] Building C object CMakeFiles/image.dir/tmp/.cache/arm/packs/NXP/MIMXRT1052_DFP/13.1.0/utilities/debug_console/fsl_debug_console.o
[48/51] Building C object CMakeFiles/image.dir/tmp/.cache/arm/packs/NXP/MIMXRT1052_DFP/13.1.0/utilities/str/fsl_str.o
[49/51] Building C object CMakeFiles/image.dir/tmp/.cache/arm/packs/NXP/MIMXRT1052_DFP/13.1.0/components/serial_manager/fsl_component_serial_mar
[50/51] Building C object CMakeFiles/image.dir/tmp/.cache/arm/packs/NXP/MIMXRT1052_DFP/13.1.0/components/serial_manager/fsl_component_serial_por
[51/51] Linking C executable /workspace/CMSIS-OUTPUT/image.axf
Program Size: Code=28092 RO-data=4108 RW-data=176 ZI-data=355420
The CMSIS project built successfully.
Making built binaries available for download.
done.
Retrieving binaries...
Build successful.
```

Use to build the project

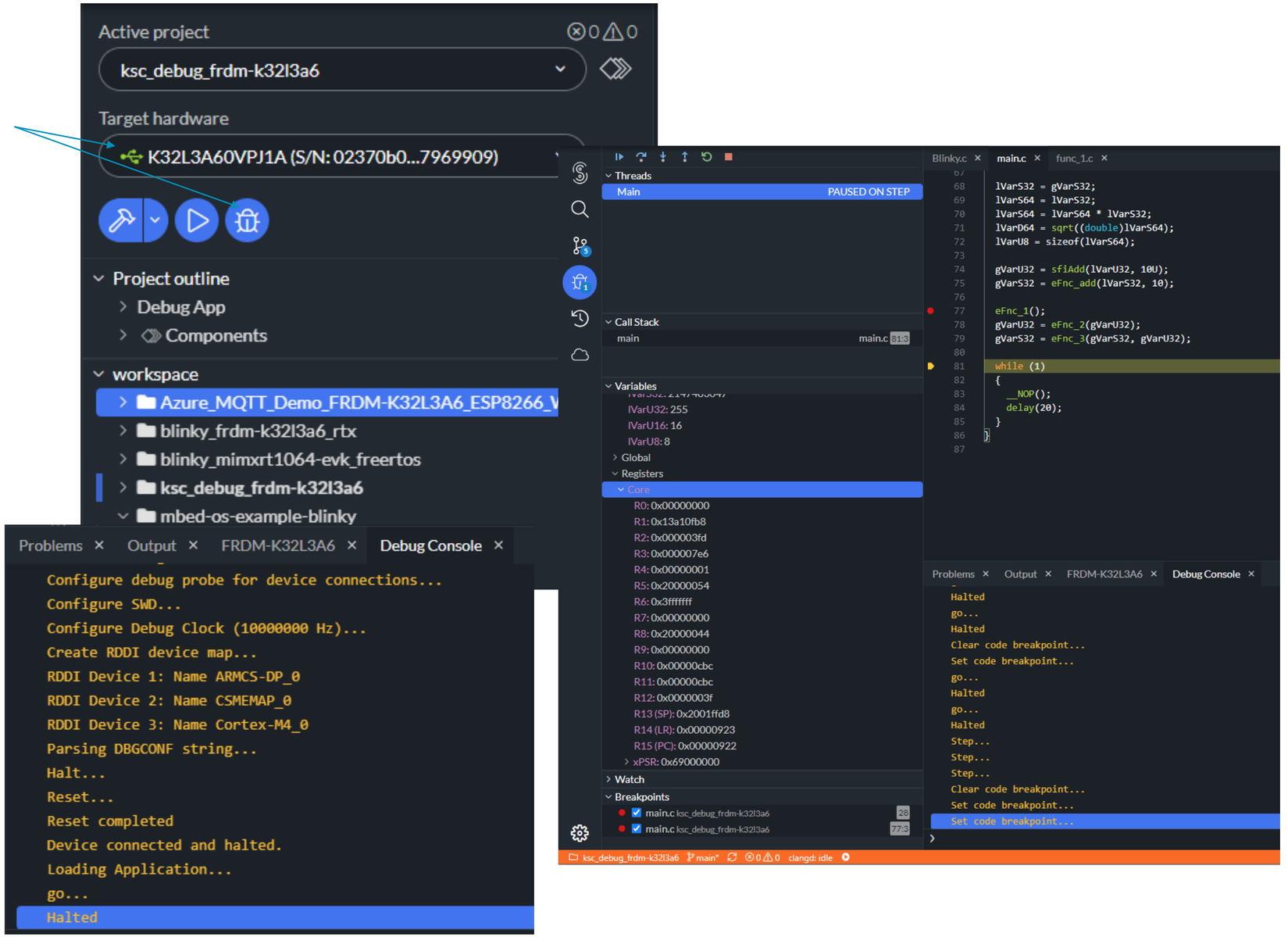
blinky_imxrt1050-evkb_rtx main*

blinky_imxrt1050-e....bin


Keil Studio
studio.keil.arm.com
File Edit Selection View Go Help
Active project
blinky_imxrt1050-evkb_rtx
Target hardware
EVKB-IMXRT1050_MDK (S/N: 0...
Workspace
blinky_imxrt1050-evkb_rtx
Board_IO
BUILD
RTE
Blinky.c
Blinky.cprj
IMXRT1050-EVKB.mex
main.c
main.h
README.md
Project outline blinky_imxrt1050-evkb_rtx
App
Board
Getting Started
README.md
Blinky project
The Blinky project can be easily used to verify the basic tool setup.
It is compliant to the Cortex Microcontroller Software Interface Standard (CMSIS) and uses the CMSIS-RTOS v2 API for RTOS functionality. The CMSIS-RTOS v2 API is supported by various real-time operating systems, for example Keil RTX5 or FreeRTOS.
Operation
At start the vioLED0 blinks in 1 sec interval.
The vioBUTTON0 changes the blink frequency and start/stops vioLED1.
The board hardware mapping of vioLED0, vioLED1, and vioBUTTON0 depends on the configuration of the CMSIS-Driver VIO.
RTOS: Keil RTX5 Real-Time Operating System
The real-time operating system Keil RTX5 implements the resource management.
It is configured with the following settings:
Global Dynamic Memory size: 24000 bytes
Problems
EVKB-IMXRT1050_MDK
Debug Console
Output
[43/51] Building C object CMakeFiles/image.dir/tmp/.cache/arm/packs/NXP/MIMXRT1052_DFP/13.1.0/components/uart/fsl_adapter_lpuart.o
[44/51] Building C object CMakeFiles/image.dir/tmp/.cache/arm/packs/NXP/MIMXRT1052_DFP/13.1.0/drivers/fsl_lpuart_edma.o
[45/51] Building C object CMakeFiles/image.dir/tmp/.cache/arm/packs/NXP/MIMXRT1052_DFP/13.1.0/drivers/fsl_usdhc.o
[46/51] Building C object CMakeFiles/image.dir/tmp/.cache/arm/packs/NXP/MIMXRT1052_DFP/13.1.0/xip/fsl_flexspi_nor_boot.o
[47/51] Building C object CMakeFiles/image.dir/tmp/.cache/arm/packs/NXP/MIMXRT1052_DFP/13.1.0/utilities/debug_console/fsl_debug_console.o
[48/51] Building C object CMakeFiles/image.dir/tmp/.cache/arm/packs/NXP/MIMXRT1052_DFP/13.1.0/utilities/str/fsl_str.o
[49/51] Building C object CMakeFiles/image.dir/tmp/.cache/arm/packs/NXP/MIMXRT1052_DFP/13.1.0/components/serial_manager/fsl_component_serial_ma
[50/51] Building C object CMakeFiles/image.dir/tmp/.cache/arm/packs/NXP/MIMXRT1052_DFP/13.1.0/components/serial_manager/fsl_component_serial_po
[51/51] Linking C executable /workspace/CMSIS-OUTPUT/image.axf
Program Size: Code=28092 RO-data=4108 RW-data=176 ZI-data=355420
The CMSIS project built successfully.
Making built binaries available for download.
done.
Retrieving binaries...
Build successful.
Use ▷ to download and run the project. The LED close to the Ethernet connector starts flashing. Use the SW8 button to change the blinking frequency.
blinky_imxrt1050-evkb_rtx main*
blinky_imxrt1050-e....bin

Debug

- Debugging a device directly from the browser is a new feature in the embedded software space.
- Support is currently limited to CMSIS-DAP v2 and ST-Link devices
- Debug is intended to be simple – run control debug that allows a developer to step through code and gain a better understanding of program execution. We will gradually enhance debug and introduce new features incrementally
- You can set breakpoints in the UI



Set breakpoints and enter debug

- Open `main.c` by clicking on it
- Go to line 61 and set a breakpoint by clicking in the free area left of the line number:

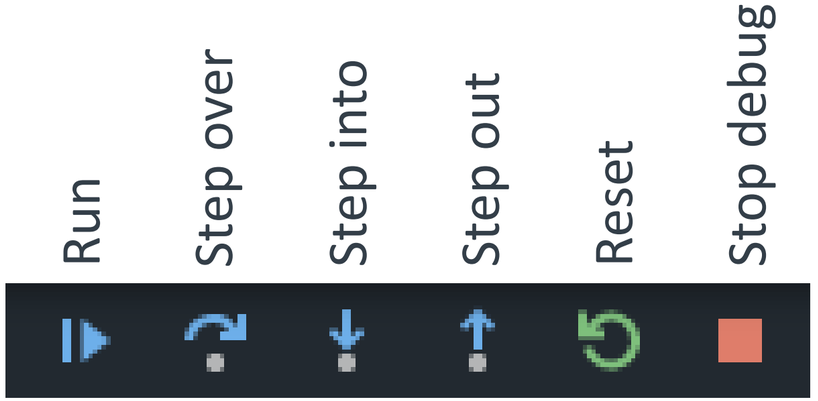
```
51   // Enable ENET_REF_CLK output mode
52   IOMUXC_EnableMode(IOMUXC_GPR, kIOMUXC_GPR_ENET1TxClkOutputDir, true);
53
54   NVIC_SetPriority(ENET_IRQn,    8U);
55   NVIC_SetPriority(USDHC1_IRQn,  8U);
56   NVIC_SetPriority(LPUART3_IRQn, 8U);
57
58   SystemCoreClockUpdate();
59
60 #ifdef RTE_VIO_BOARD
61   vioInit();                                // Initialize Virtual I/O
62 #endif
```

- Open `Blinky.c` and set a breakpoint at line 45:

```
39   for (;;) {
40     if (osThreadFlagsWait(1U, osFlagsWaitAny, 0U) == 1U) {
41       active_flag ^= 1U;
42     }
43
44     if (active_flag == 1U) {
45       vioSetSignal(vioLED0, vioLEDoff);         // Switch LED0 off
46       vioSetSignal(vioLED1, vioLEDon);          // Switch LED1 on
47       osDelay(100U);                            // Delay 100 ms
48       vioSetSignal(vioLED0, vioLEDon);          // Switch LED0 on
49       vioSetSignal(vioLED1, vioLEDoff);         // Switch LED1 off
50       osDelay(100U);                            // Delay 100 ms
51     }
```

- Use the  button to start a debug session
- The program is being flashed, and the debug view is entered

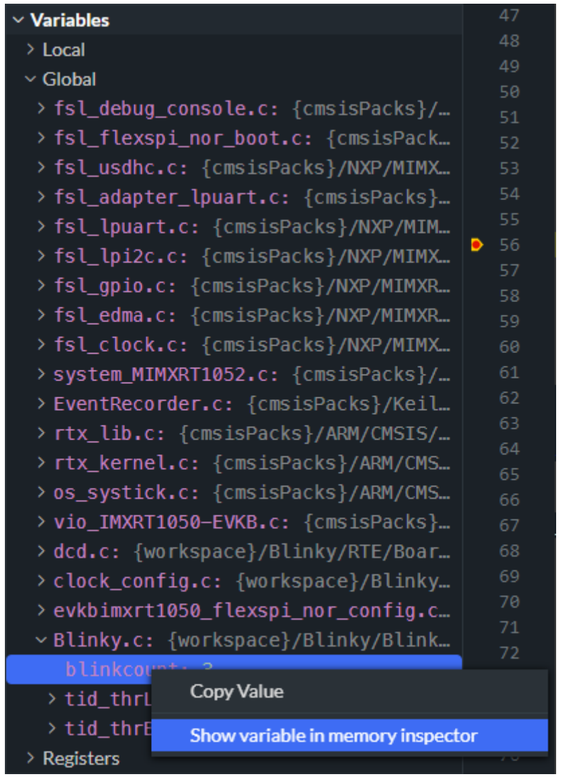
Enter debug mode

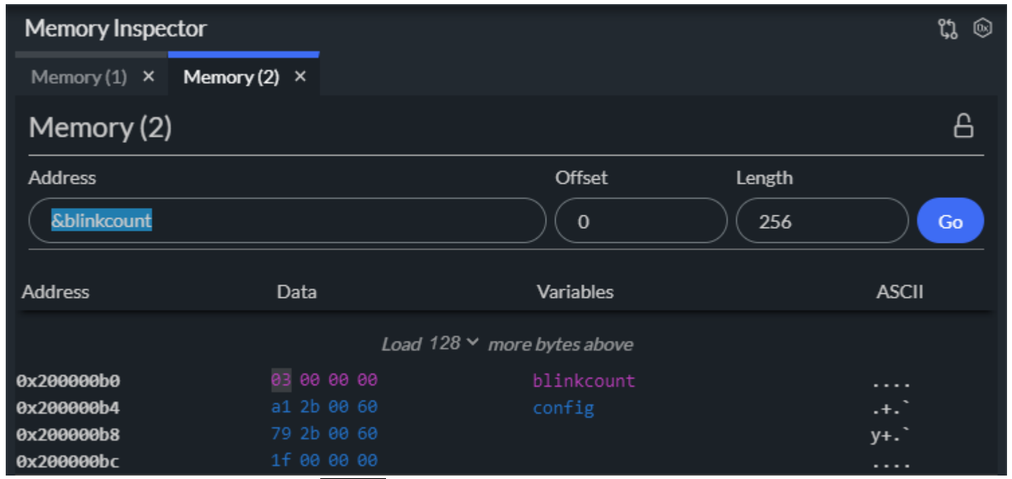


- Program execution stops at ‘main’
- Use the debug controls to run/step through the program:
- Press **SW8** while the program executes, observe how the breakpoint is hit
- Next time the breakpoint is hit, use the **Step into** button to examine the operation of vioSetSignal
- Use the **Step out** button to go back some levels of hierarchy
- **Run** the program and press **SW8**
- Expand **Variables – Local** and observe the values of last and state
- Check the entries in **Variables – Registers** and then step to see the changed **R15 (PC)**
- Go to **Breakpoints** and disable the one for vioInit (main.c, line 61)
- **Run** the program and press **SW8**
- **Exit** the debug mode

Using Memory Inspector

- In `Blinky.c` add:
  - At line 31: `int blinkcount = 0;`
  - At line 56: `blinkcount++;`
- Set a breakpoint at line 56, build, and run the project.
- Once stopped, go to **Variables – Globals – Blinky.c** and right-click on blinkcount. Select **Show variable in memory inspector**:





- Click on the 03 in the data field and enter a new value. Run the project and observe the change.

arm DevSummit

Advanced examples: TFLu Micro-speech

Key take-aways

In this part you will learn how to:

- Import a project from GitHub.
- Use Keil Studio features to work with your GitHub repo.
- Check printf debug output on the serial window.

KeilChris/TFLmicrospeech: Speec

github.com/KeilChris/TFLmicrospeech/tree/main

Search or jump to... / Pull requests Issues Marketplace Explore

KeilChris / TFLmicrospeech Public

Unwatch 1 Star 0 Fork 1

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags Go to file Add file Code

KeilChris Revert "VSI folder removed" ... a065096 4 minutes ago 10 commits

Board_IO	Initial commit	yesterday
Driver_Audio	Initial commit	yesterday
RTE	Initial commit	yesterday
VSI	Revert "VSI folder removed"	4 minutes ago
micro_speech/src	Initial commit	yesterday
.gitignore	Initial commit	yesterday
IMXRT1050-EVKB.mex	Initial commit	yesterday
	commit	yesterday
	e README.md	yesterday
	commit	yesterday
	commit	yesterday
	"VSI folder removed"	4 minutes ago
	commit	yesterday

Go to **github.com/KeilChris/TFLmicrospeech** and use Fork to create a fork in your own GitHub profile

README.md

arm Keil Studio Import

Micro Speech Example for NXP IMXRT1050-EVKB

This example runs a voice recognition model for two keywords ("yes" and "no") on the IMXRT1050-EVKB. It uses the

About

Speech example for TFLu

Readme

Apache-2.0 License

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Contributors 2

KeilChris Christopher Seidl

thegecko Rob Moran

Languages

C++ 56.3% C 38.0% Assembly 4.9% Python 0.8%

KeilChris/TFLmicrospeech: Speec

github.com/KeilChris/TFLmicrospeech/tree/main

Search or jump to... / Pull requests Issues Marketplace Explore

KeilChris / TFLmicrospeech Public

Unwatch 1 Star 0 Fork 1

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags Go to file Add file Code

KeilChris Revert "VSI folder removed" ... a065096 4 minutes ago 10 commits

Board_IO	Initial commit	yesterday
Driver_Audio	Initial commit	yesterday
RTE	Initial commit	yesterday
VSI	Revert "VSI folder removed"	4 minutes ago
micro_speech/src	Initial commit	yesterday
.gitignore	Initial commit	yesterday
IMXRT1050-EVKB.mex	Initial commit	yesterday
LICENSE	Initial commit	yesterday
README.md	Update README.md	yesterday
main.c	Initial commit	yesterday
main.h	Initial commit	yesterday
microspeech.IMXRT1050-EVKB.cprj	Revert "VSI folder removed"	4 minutes ago
microspeech.c	Initial commit	yesterday

README.md

arm Keil Studio Import

Micro Speech Example for NXP IMXRT1050-EVKB

This example runs a voice recognition model for two keywords ("yes" and "no") on the IMXRT1050-EVKB. It uses the

About

Speech example for TFLu

Readme

Apache-2.0 License

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Contributors 2

KeilChris Christopher Seidl

thegecko Rob Moran

Languages

C++ 56.3% C 38.0% Assembly 4.9% Python 0.8%

In the TFLmicrospeech repo in **your** GitHub profile, press

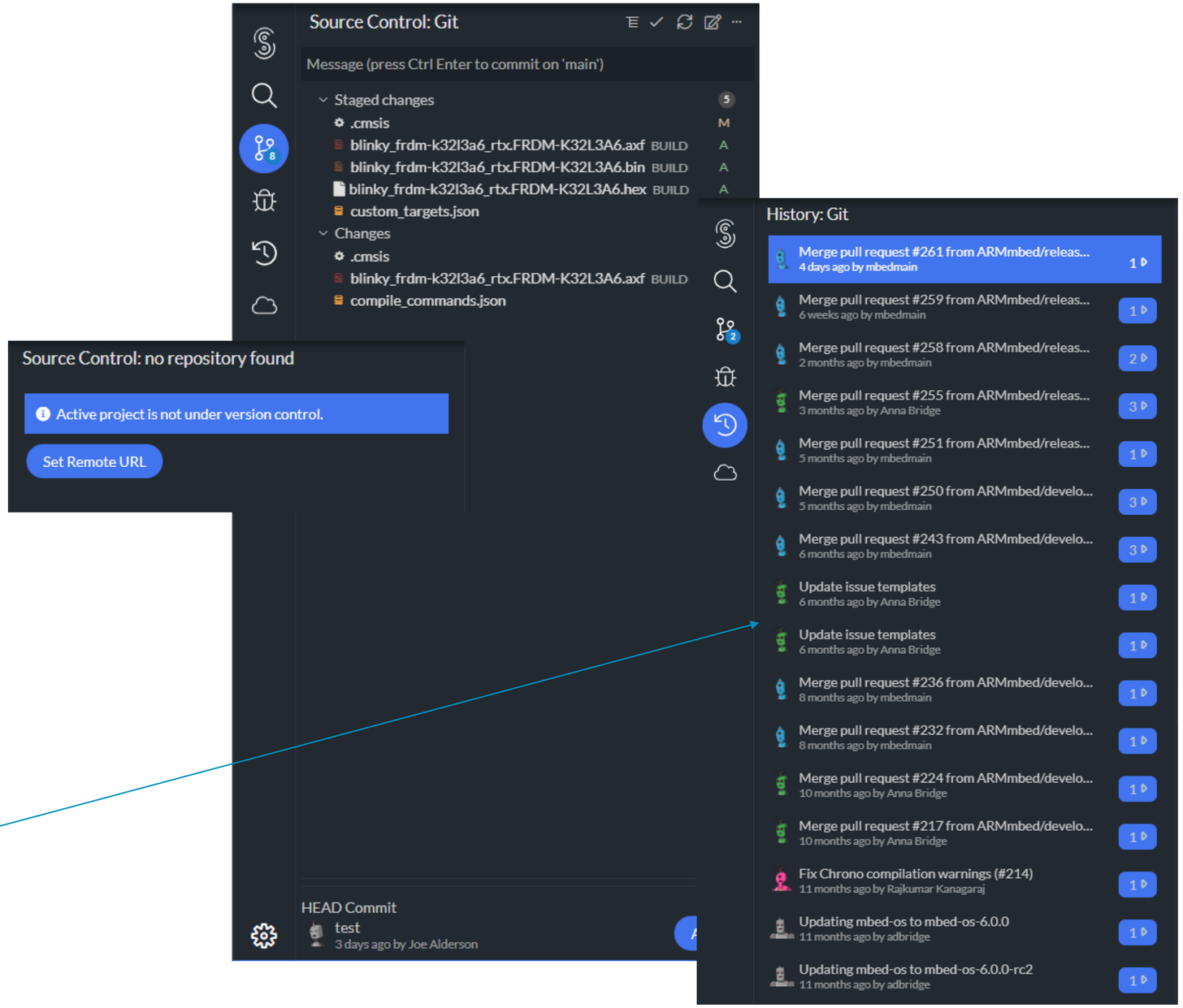
arm Keil Studio Import

to import the project into Keil Studio

Alternatively, you can use **File – Import Project...** in Keil Studio to open the example

Source control

- Comprehensive source control integration allows you to carry out most common git actions directly from the IDE, allowing you to take advantage of GitHub integrations and actions
- Projects are not under version control by default unless they are imported into the workspace. You can choose to keep them under version control
- Track changes to files and see their status reflected in the workspace
- Commit changes with commit messages
- Switch branches
- Compare files between branches or commits
- View the history of a repository or file
- Create a new repository and publish committed code to GitHub



Keil Studio

studio.keil.arm.com

File Edit Selection View Go Help

Active project 42 5

tflmicrospeech

Target hardware

MIMXRT1052DVL6B (S/N: 0227...

Workspace

- aws_mqtt_demo_imxrt1050-evkb_esp8...
- blinky_imxrt1050-evkb_rtx
- tflmicrospeech U

main.c | vio_IMXRT1050-EVKB.c | Release Notes | README.md | Blinky.c | rtx_delay.c

arm Keil Studio Import

Micro Speech Example for NXP IMXRT1050-EVKB

This example runs a voice recognition model for two keywords ("yes" and "no") on the IMXRT1050-EVKB. It uses the EVK's built-in microphone (P1). The outcome of the speech recognition is shown in the serial monitor of Keil Studio Cloud.

RTOS: Keil RTX5 Real-Time Operating System

The real-time operating system Keil RTX5 implements the resource management.

It is configured with the following settings:

- Global Dynamic Memory size: 24000 bytes
- Default Thread Stack size: 3072 bytes
- Event Recorder Configuration
  - Global Initialization: 1
    - Start Recording: 1

Refer to Configure RTX v5 for a detailed description of all configuration optio

Board: NXP IMXRT1050-EVKB

The tables below list the device configuration for this board. The board layer f

Make sure that the baud rate of the serial window is set to **115200**:

Problems 47 | Debug Console | Output | MIMXRT1052DVL6B

Support: SDK Project Template:

Baud rate 115200

```
Heard silence (161) @700ms
Heard silence (164) @800ms
Heard silence (148) @900ms
Heard silence (141) @1000ms
Heard silence (142) @1100ms
Heard yes (149) @4200ms
Heard no (145) @5700ms
Heard yes (145) @7600ms
Heard no (150) @9400ms
Heard unknown (141) @10600ms
Heard unknown (142) @12200ms
Heard unknown (141) @14300ms
Heard unknown (151) @15900ms
Heard unknown (178) @17500ms
Heard no (146) @18400ms
Heard unknown (146) @19000ms
Heard unknown (141) @21200ms
Heard unknown (152) @22900ms
```

The Serial Monitor (here: **MIMXRT1052DVL6B**) shows the output of the application. Say *yes* or *no* to the microphone located at P1 on the board.

tflmicrospeech main* 42 5

arm DevSummit

Advanced examples: AWS MQTT

Using CMSIS-Packs from AWS

Under construction

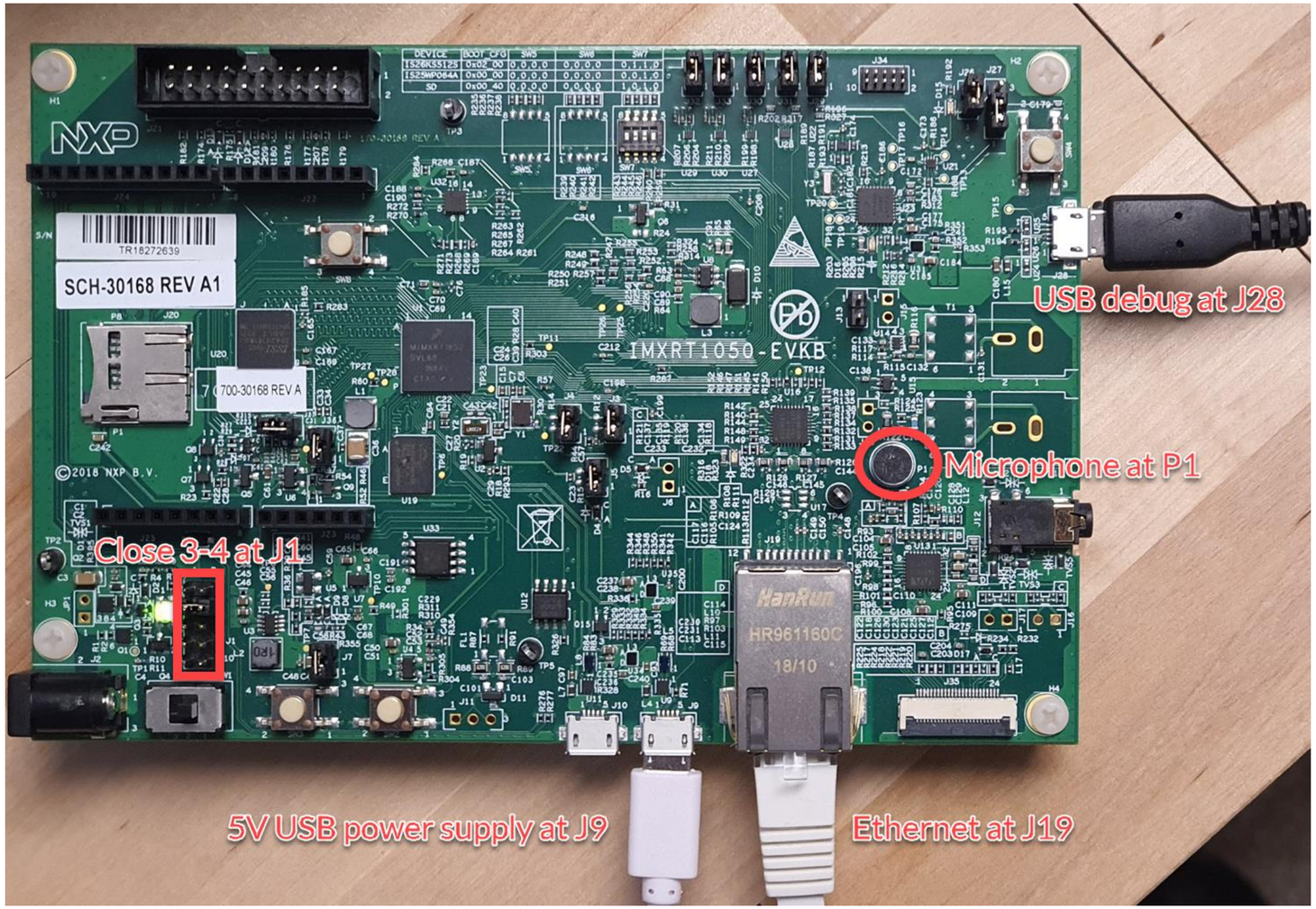
Key take-aways

In this part you will learn how to:

- Distinguish between example projects on keil.arm.com.
- Rapidly connect to the cloud.
- Check MQTT messages in the serial window and on the cloud server.

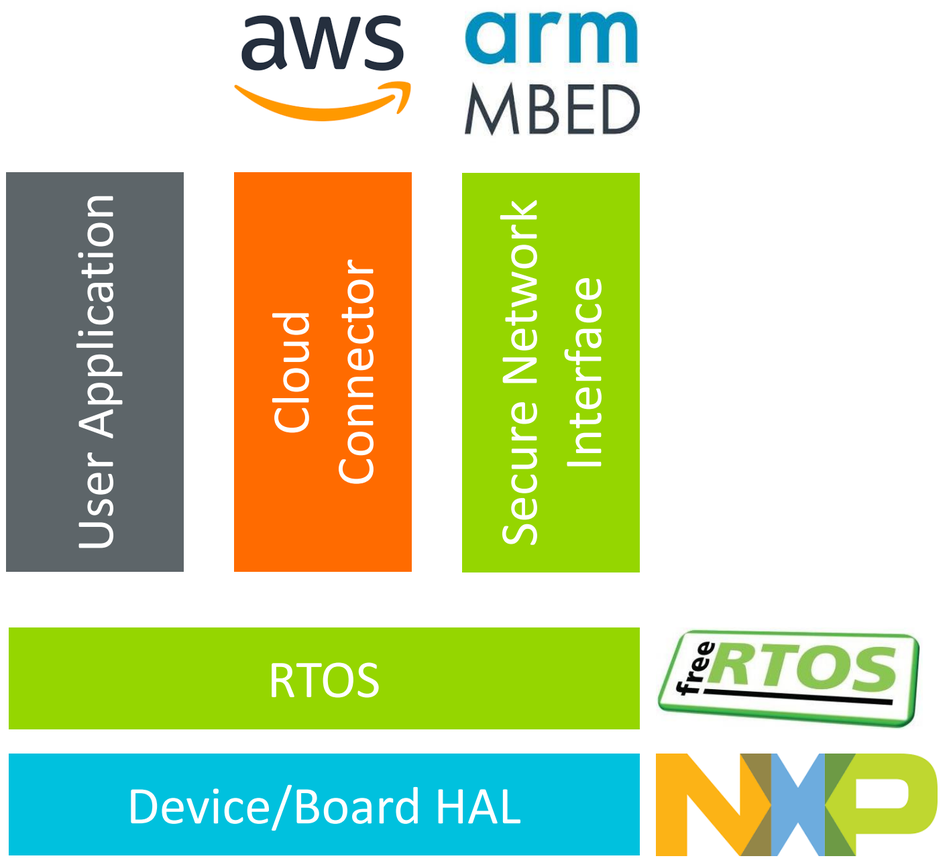
Hardware setup

- Connect your board with an Ethernet cable to your router:



Reference Code Examples for IoT on Cortex-M

Simplified view to the software building blocks for IoT endpoints



- **Device/Board HAL**: abstraction of processor and peripherals with hardware specific configuration
- **RTOS**: thread and resource management
- **Secure Network Interface**: encrypted internet connection using different interfaces (Ethernet, WiFi, …)
- **Cloud Connector**: protocol interface to cloud provider; here, we are using CMSIS-Packs provided by AWS
- **User Application**: custom functionality of endpoints

Arm Keil | EVKB-IMXRT1050_MDK

keil.arm.com/boards/nxp-evkb-imxrt1050-mdk-rev-a1-126523e/

arm KEIL Hardware Documentation Keil Studio

Hardware (411)

Search by name or vendor

Only show boards with example projects

- EVKB-IMXRT1050_MDK — NXP
- FRDM-K20D50M — NXP
- FRDM-K22F — NXP
- FRDM-K28F — NXP

With the **AWS MQTT Demo (FreeRTOS)** project, click **Create project** to import it into Keil Studio

EVKB-IMXRT1050_MDK Rev. A1

NXP

Core	Debug probe	Device
Cortex-M7	JTAG/SW JTAG/SW	MIMXRT1052DVL6B

Projects Features Documentation

AWS MQTT Demo RTX Ethernet Socket (MW-Network) — Create project
Demonstrates the subscribe-publish workflow of MQTT

AWS MQTT Demo FreeRTOS — Create project
Demonstrates the subscribe-publish workflow of MQTT

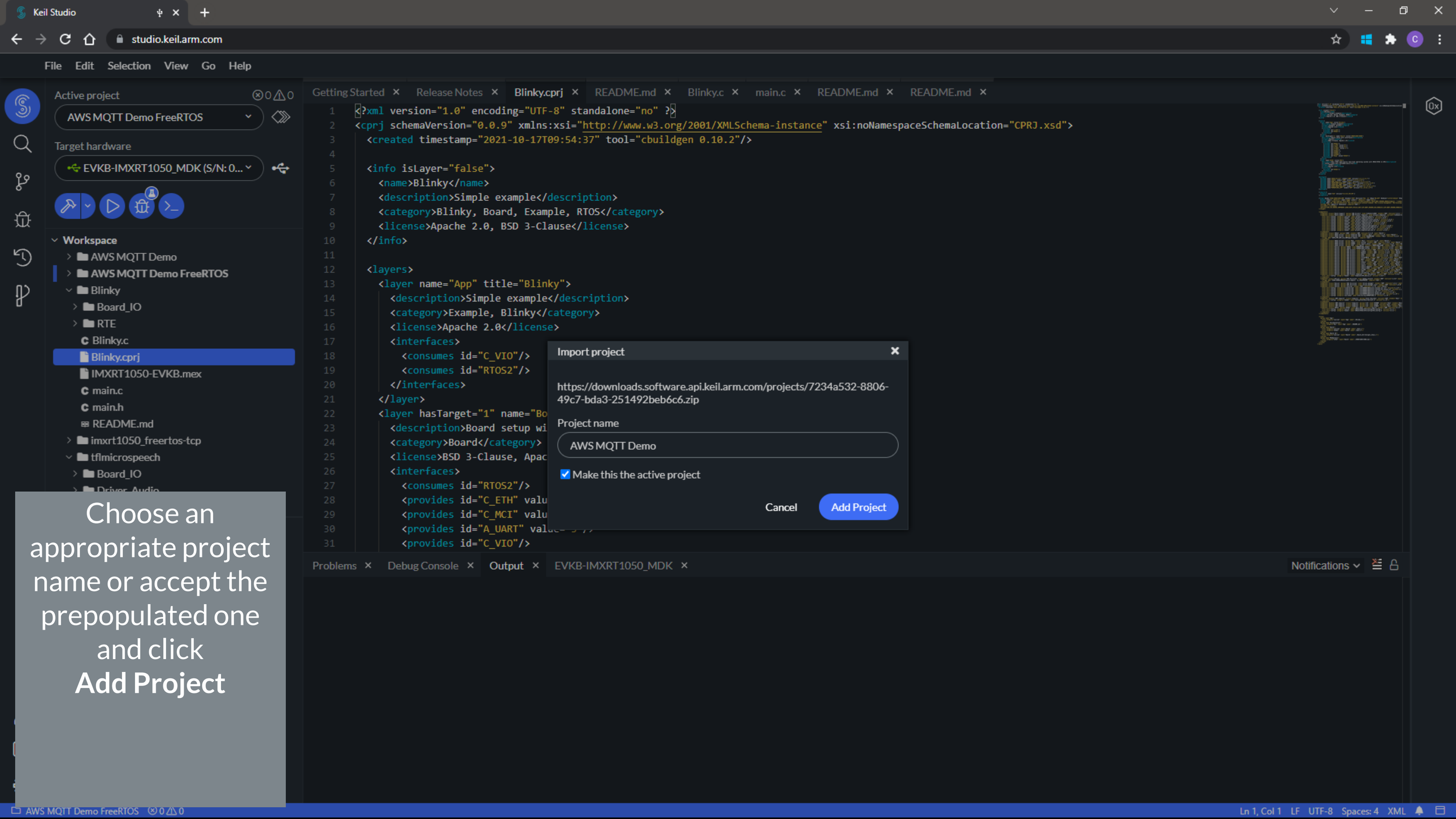
Azure MQTT Demo ESP8266 RTX WiFi Socket — Create project
Demonstrates the subscribe-publish workflow of MQTT

Blinky RTX — Create project
Simple example

Google MQTT Demo ESP8266 FreeRTOS WiFi Socket — Create project
Demonstrates the subscribe-publish workflow of MQTT

Paho MQTT Demo ESP8266 FreeRTOS WiFi Socket — Create project

https://studio.keil.arm.com/?cmsis-import-url=https%3A%2F%2Fdownloads.software.api.keil.arm.com%2Fprojects%2...

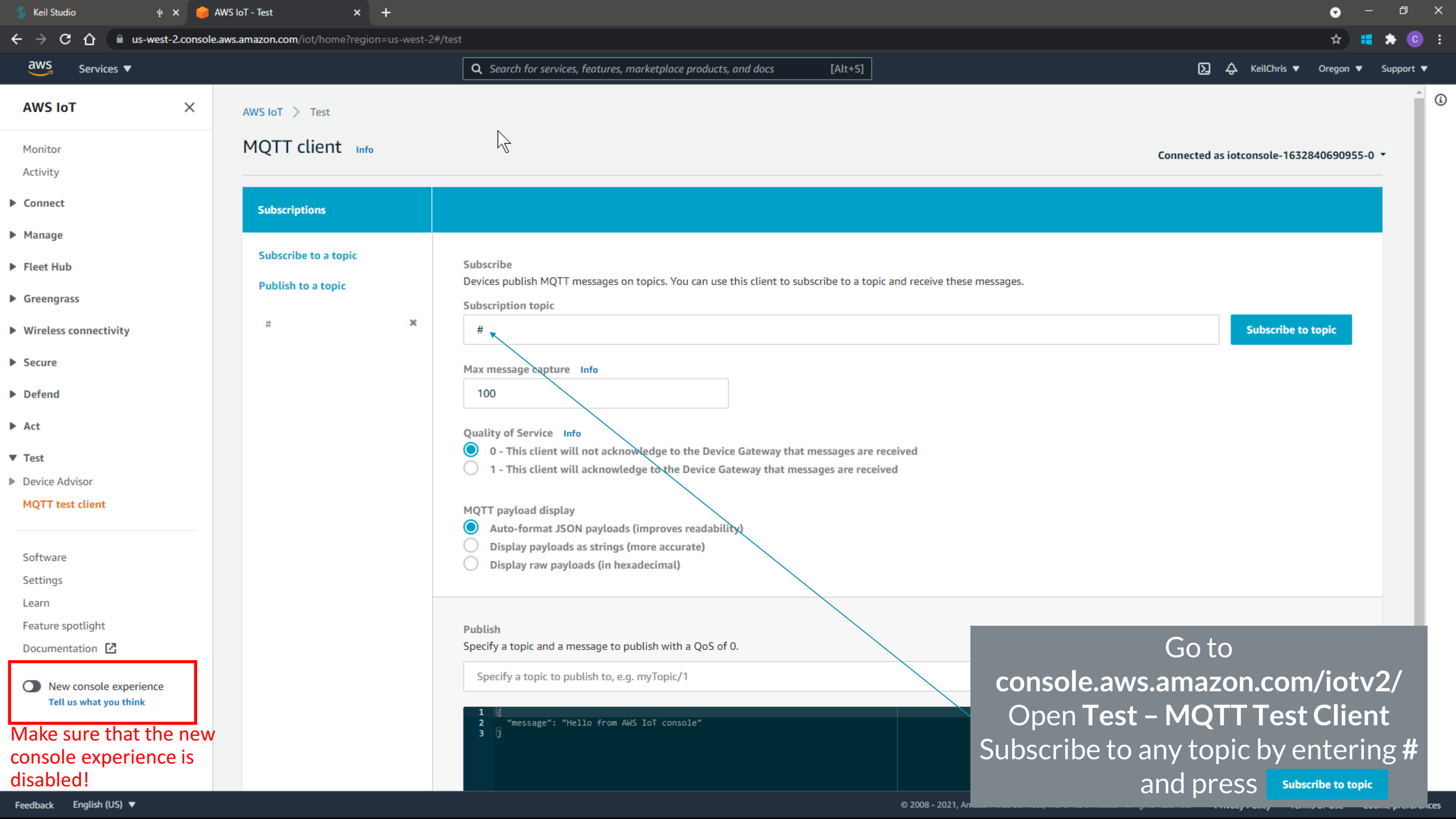

Keil Studio
studio.keil.arm.com
File Edit Selection View Go Help
Active project
AWS MQTT Demo FreeRTOS
Target hardware
EVKB-IMXRT1050_MDK (S/N: 0...
Workspace
AWS MQTT Demo
AWS MQTT Demo FreeRTOS
Blinky
Board_IO
RTE
Blinky.c
Blinky.cprj
IMXRT1050-EVKB.mex
main.c
main.h
README.md
imxrt1050_freertos-tcp
tflmicrospeech
Board_IO
Getting Started
Release Notes
Blinky.cprj
README.md
Blinky.c
main.c
README.md
README.md
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<cprj schemaVersion="0.0.9" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="CPRJ.xsd">
<created timestamp="2021-10-17T09:54:37" tool="cbuildgen 0.10.2"/>
<info isLayer="false">
<name>Blinky</name>
<description>Simple example</description>
<category>Blinky, Board, Example, RTOS</category>
<license>Apache 2.0, BSD 3-Clause</license>
</info>
<layers>
<layer name="App" title="Blinky">
<description>Simple example</description>
<category>Example, Blinky</category>
<license>Apache 2.0</license>
<interfaces>
<consumes id="C_VIO"/>
<consumes id="RTOS2"/>
</interfaces>
</layer>
<layer hasTarget="1" name="Bo
<description>Board setup wi
<category>Board</category>
<license>BSD 3-Clause, Apac
<interfaces>
<consumes id="RTOS2"/>
<provides id="C_ETH" valu
<provides id="C_MCI" valu
<provides id="A_UART" valu
<provides id="C_VIO"/>
Import project
https://downloads.software.api.keil.arm.com/projects/7234a532-8806-49c7-bda3-251492beb6c6.zip
Project name
AWS MQTT Demo
Make this the active project
Cancel
Add Project
Problems
Debug Console
Output
EVKB-IMXRT1050_MDK
Notifications
Choose an appropriate project name or accept the prepopulated one and click
Add Project
AWS MQTT Demo FreeRTOS
Ln 1, Col 1
LF
UTF-8
Spaces: 4
XML

Build the ready-to-run example

- For this pre-built example, you'll need:
  - Your AWS credentials as described here: https://github.com/MDK-Packs/Documentation/tree/master/AWS_Thing

- Open the files:
  - `./amazon-freertos/demos/include/aws_clientcredential_keys.h` and enter:
    - `keyCLIENT_CERTIFICATE_PEM`
    - `keyCLIENT_PRIVATE_KEY_PEM`
  - `./amazon-freertos/demos/include/aws_clientcredential.h` and enter:
    - `clientcredentialMQTT_BROKER_ENDPOINT`
    - `clientcredentialIOT_THING_NAME`

- Build, download, and start a debug session


Keil Studio
AWS IoT - Test
us-west-2.console.aws.amazon.com/iot/home?region=us-west-2#/test
Services
Search for services, features, marketplace products, and docs [Alt+S]
KeilChris
Oregon
Support
AWS IoT
Monitor
Activity
Connect
Manage
Fleet Hub
Greengrass
Wireless connectivity
Secure
Defend
Act
Test
Device Advisor
MQTT test client
Software
Settings
Learn
Feature spotlight
Documentation
New console experience
Tell us what you think
Make sure that the new console experience is disabled!
AWS IoT > Test
MQTT client Info
Connected as iotconsole-1632840690955-0
Subscriptions
Subscribe to a topic
Publish to a topic
#
Subscribe
Devices publish MQTT messages on topics. You can use this client to subscribe to a topic and receive these messages.
Subscription topic
#
Subscribe to topic
Max message capture Info
100
Quality of Service Info
0 - This client will not acknowledge to the Device Gateway that messages are received
1 - This client will acknowledge to the Device Gateway that messages are received
MQTT payload display
Auto-format JSON payloads (improves readability)
Display payloads as strings (more accurate)
Display raw payloads (in hexadecimal)
Publish
Specify a topic and a message to publish with a QoS of 0.
Specify a topic to publish to, e.g. myTopic/1
{
"message": "Hello from AWS IoT console"
}
Go to
console.aws.amazon.com/iotv2/
Open Test – MQTT Test Client
Subscribe to any topic by entering #
and press
Subscribe to topic
Feedback
English (US)

Keil Studio | AWS IoT - MQTT test client

studio.keil.arm.com

File Edit Selection View Go Help

Active project: aws_mqtt_demo_imxrt1050-evkb_es...

Target hardware: EVKB-IMXRT1050_MDK (S/N: 0...

Run in progress

Workspace
- _AWS_MQTT_Demo U
- Board_Support
- CMSIS_Driver
- Compiler
- Device
- IoT_Client M
  - iot_config.h M
- IoT_Utility
- RTOS
- Security
- app_main.c
- AWS_MQTT_Demo.cprj
- IMXRT1050-EVKB.mex
- iot_demo_logging.h
- iot_demo_mqtt.c

RTX_Config.c | RTX_Config.h | README.md | socket_startup.c | iot_config.h | main.c aws_mqtt_demo_imxrt1050-evkb_esp8266_wifi_freertos | README.md aws_mqtt_demo_imxrt1050-evkb_esp8266_wifi_freerto

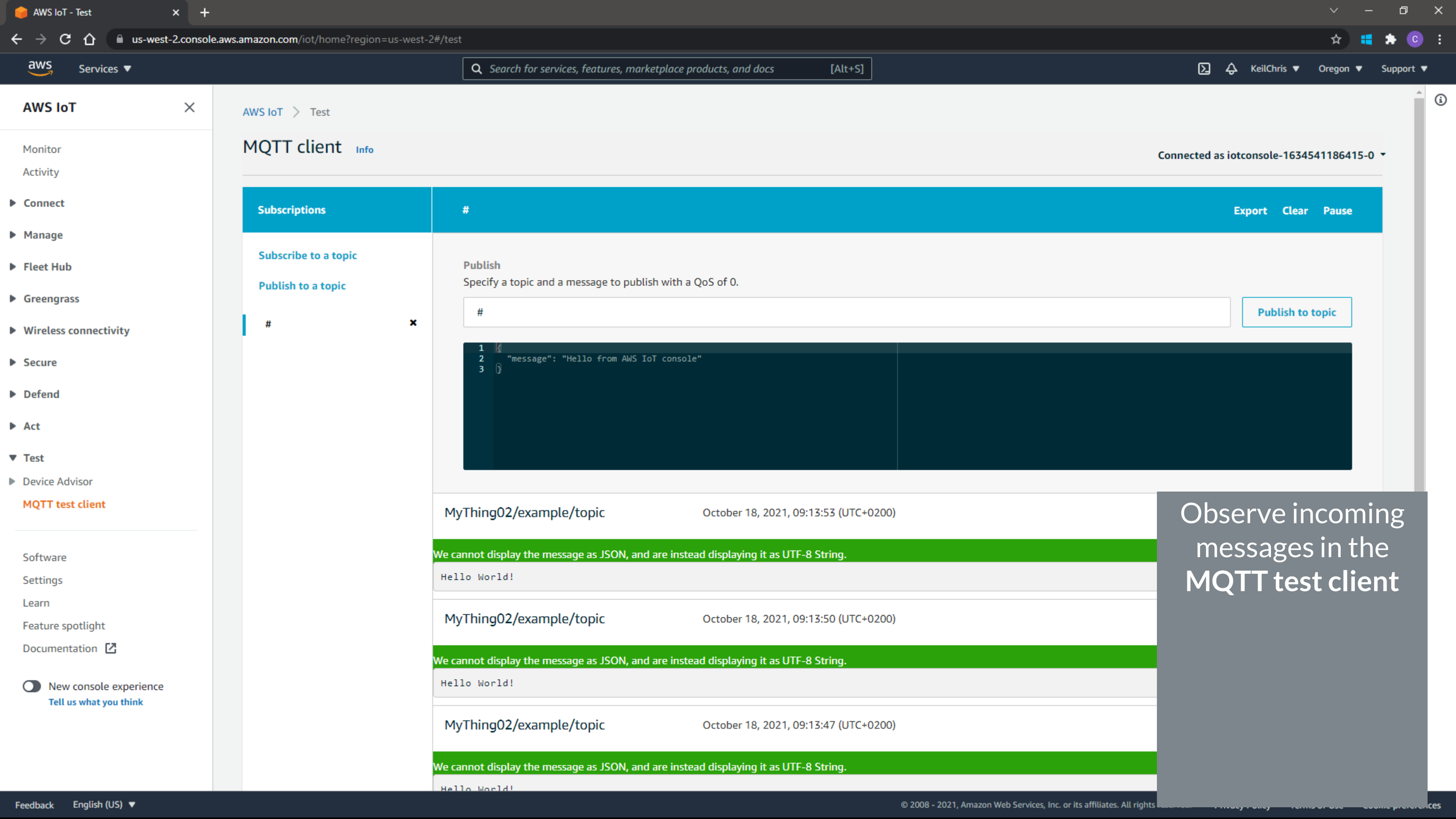
```
### NVIC Configuration

| NVIC Interrupt      | Priority
|:--------------------|:--------
| ENET                | 8
| USDHC1              | 8
| LPUART3             | 8

**STDIO** is routed to debug console through Virtual COM port (DAP-Link, peripheral = LPUART1, baudrate = 115200)

### CMSIS-Driver mapping

| CMSIS-Driver | Peripheral
|:-------------|:----------
| ETH_MAC0     | ENET
| ETH_PHY0     | KSZ8081RNB (external)
| MCI0         | USDHC1
| USART3       | LPUART3

| CMSIS-Driver VIO  | Physical board hardware
|:------------------|:-----------------------
| vioBUTTON0        | User Button SW8 (WAKEUP)
| vioLED0           | User LED (GPIO_AD_B0_09)
| vioMotionAccelero | 3-Axis Accelerometer (FXOS8700CQ)
| vioMotionMagneto  | 3-Axis Magnetometer (FXOS8700CQ)
```

Problems 99+ | EVKB-IMXRT1050_MDK | Debug Console | Output | Baud rate 115200

```
Connecting to WiFi ...
WiFi network connection succeeded!
[INFO ][INIT] SDK successfNFO ][MQTT] MQTT library successfully initialized.
[INFO ][DEM[INFO ][NET] (Network connection 2000c700) TLS handshake succeconnection established.
[INFO ][MQTT] Establishing new MQTT connection.
[INFO ][MQTT] Anonymous metrics (SDK language, SDK version) will be provided to AWS IoT. Recompile with AWS_IOT_MQTT_ENABLE_METRICS set to 0 to disable.
[INFO ][MQTT] (MQTT connection 2000dbc0, CONNECT operation 2000dc68) Waiting for operation completion.
[INFO ][MQTT] (MQTT connection 2000dbc0, CONNECT operation 2000dc68) Wait complete with result SUCCESS.
[INFO ][MQTT] New MQTT connection 20006368 established.
[INFO ][MQTT] (MQTT connection 2000dbc0) SUBSCRIBE operation scheduled.
[INFO ][MQTT] (MQTT connection 2000dbc0, SUBSCRIBE operation 2000dc68) Waiting for operation completion.
[INFO ][M[DEMO] All demo topic filter subscriptions accepted.
[INFO ][Dued.
[INFO ][MQTT] (MQTT connection 2000dbc0) MQTT PUBLISH operation queued.
[INFO ][MQTT] (MQTT connection 2000dbc0) MQTT PUBLISH operation queued.
[INFO ][MQTT] (MQTT connection 2000dbc0) MQTT PUBLISH operation queued.
[INFO ][MQTT] (MQTT connection 2000dbc0) MQTT PUBLISH operation queued.
[INFO ][MQTT] (MQTT connection 2000dbc0) MQTT PUBLISH operation queued.
[INFO ][MQTT] (MQTT connection 2000dbc0) MQTT PUBLISH operation queued.
```

Run the program
Make sure that the baud rate of the serial window (**EVKB-IMXRT1050_MDK**) is set to **115200**
Observe the UART output

aws_mqtt_demo_imxrt1050-evkb_esp8266_wifi_freertos main* 43 6 | Ln 1, Col 1 LF UTF-8 Spaces: 4 Plain Text Program 2


AWS IoT - Test
us-west-2.console.aws.amazon.com/iot/home?region=us-west-2#/test
Services
Search for services, features, marketplace products, and docs [Alt+S]
KeilChris
Oregon
Support
AWS IoT
Monitor
Activity
Connect
Manage
Fleet Hub
Greengrass
Wireless connectivity
Secure
Defend
Act
Test
Device Advisor
MQTT test client
Software
Settings
Learn
Feature spotlight
Documentation
New console experience
Tell us what you think
AWS IoT > Test
MQTT client Info
Connected as iotconsole-1634541186415-0
Subscriptions
Subscribe to a topic
Publish to a topic
#
Export Clear Pause
Publish
Specify a topic and a message to publish with a QoS of 0.
#
Publish to topic
{
"message": "Hello from AWS IoT console"
}
MyThing02/example/topic October 18, 2021, 09:13:53 (UTC+0200)
We cannot display the message as JSON, and are instead displaying it as UTF-8 String.
Hello World!
MyThing02/example/topic October 18, 2021, 09:13:50 (UTC+0200)
We cannot display the message as JSON, and are instead displaying it as UTF-8 String.
Hello World!
MyThing02/example/topic October 18, 2021, 09:13:47 (UTC+0200)
We cannot display the message as JSON, and are instead displaying it as UTF-8 String.
Observe incoming messages in the MQTT test client
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights

arm DevSummit

Next steps

Learn more at DevSummit

TECH TALK WITH Q&A

CI/CD and MLOps Workflow for IoT endpoint development

Reinhard Keil, Arm

On-demand

TECH TALK

Transforming the IoT and Microcontroller Software Development Journey

Sam Taylor, Arm

On-demand

WORKSHOP

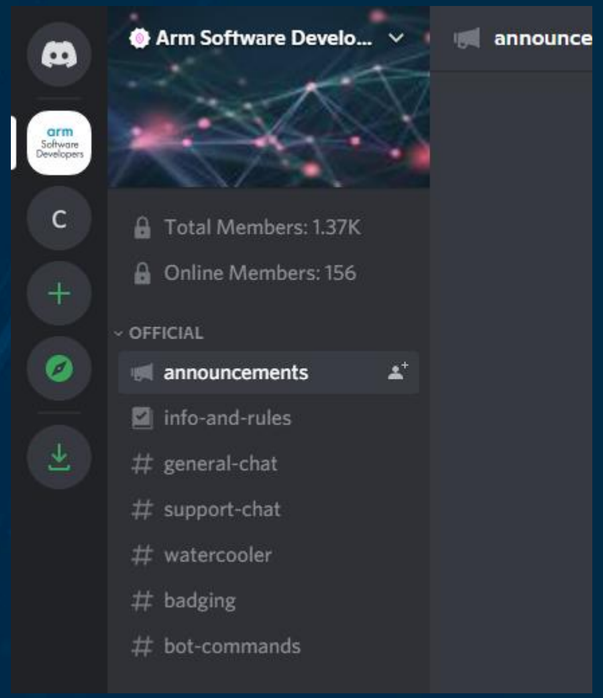
IoT DevOps made simple and scalable in the cloud

Jason Andrews, Arm

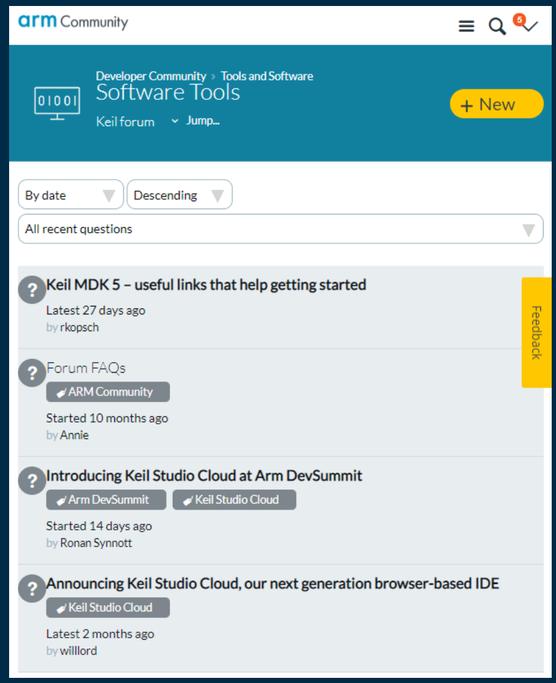
Thu, 9.15 PST, 21 Oct

Stay Connected

keil.arm.com



discord.gg/ArmSoftwareDev



bit.ly/3igNII0

Take our survey » www.surveymonkey.co.uk/r/Keil_Studio

Contact the presenter

NAME

Christopher Seidl
Senior Product Manager, DSG

EMAIL

christopher.seidl@arm.com

arm DevSummit