

ARM

Steps to Build ARM MDK Project

1. Install Required Tools

1.1 Arm Keil Studio Pack (MDK v6)

- Install the **Arm Keil Studio Pack (MDK v6)** extension for **VS Code**
- This provides CMSIS, build tools, and project integration

1.2 J-Link Software

- Download the J-Link Software Package from the SEGGER website (**>=V8.92**):
<https://www.segger.com/downloads/jlink/>
- Choose the installer based on your operating system
- Install the software after download
- Add the path to Environment Variables `C:\Program Files\SEGGER\JLink_V892`

J-Link is required for flashing the firmware to the target device.

1.3 cpackget

 **cmsis-toolbox-windows-amd64.zip** 12.56 MB

- Extract and copy the folder to `C:\`
- Add the path to Environment Variables `C:\cmsis-toolbox-windows-amd64\bin`

1.4 ARM Packs

- Download the latest Alif Semiconductor Ensemble 2.1.0 pack and ARM SDS 2.1.0 pack
- `cpackget add ARM::SDS@2.1.0`
- `cpackget add AlifSemiconductor::Ensemble@2.1.0`

2. Clone the ARM MDK Repository

- Clone the ARM MDK repository from GitLab: [ARM-MDK](#)

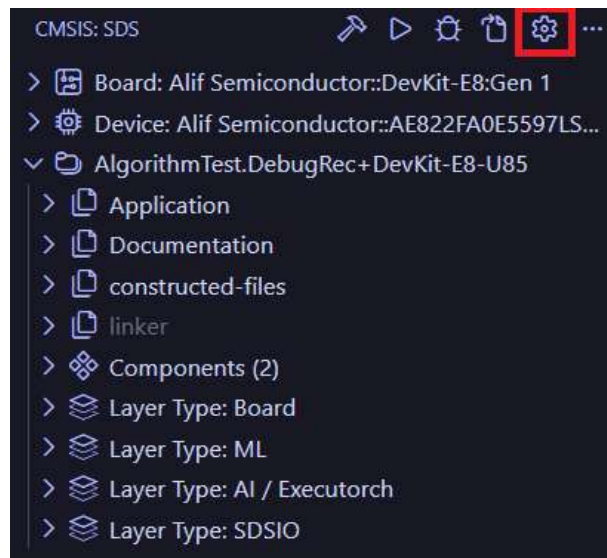
- Open the cloned repository folder in **VS Code**

3. Open the Project in VS Code

- 1. Launch **VS Code**
- 2. Open the project workspace (the cloned repository)
- 3. From the **VS Code Activity Bar**, open the **CMSIS** extension
- 4. Select **Open Solution in Workspace**

4. Configure the Build

- **4.1 Select Target Device**
 - Choose the appropriate device:
 - **AppKit**
 - **DevKit-E8**
- **4.2 Configure Project Settings**
 1. Open the **Explorer Panel**
 2. Click on the **Settings** menu



3. Select:
 - **Target Type**
 - **Project** you want to build
4. Save the configuration

5. Build the Project

- Configure the Active Target and Project as AppKit-E8-AIML-U85/DevKit-U85 and AlgorithmTest respectively by clicking the setting icon in Keil VS code extension and save.

Note: Compatible *exclusively* with **AppKit-E8-U85** and **DevKit-E8-U85**, and requires the **AC6 Compiler**.

The screenshot shows the 'Active Target' settings for AppKit-E8-AIML-U85. The 'Active Target' section is expanded, showing a table with columns: Target Type, Target Set, Board, and Device. Two entries are listed, both for AppKit-E8-AIML-U85 with a '<default>' target set, AppKit-E8-AIML board, and AE822FA0E5597BS0 device. Below this is the 'Projects and Images for AppKit-E8-AIML-U85' section, which includes a table for Projects (DataTest, AlgorithmTest) and Images, with columns for Build Type and Load Settings. The 'Debug Adapter for AppKit-E8-AIML-U85' section is also visible, showing 'J-Link Server' as the adapter and 'Update launch.json and tasks.json' checked. The 'Debug Interface' section at the bottom shows 'Telnet' as the interface, 'Clock (kHz)' set to 4000, and 'Protocol' set to SWD.

Active Target [\[icon\]](#)

Select target for build, load, and debug. The Target Set stores selected projects, images, and debug adapter. [Edit csolution.yml](#)

Target Type	Target Set	Board	Device
<input type="radio"/> AppKit-E8-AIML-U55	<default>	AppKit-E8-AIML	AE822FA0E5597BS0
<input checked="" type="radio"/> AppKit-E8-AIML-U85	<default>	AppKit-E8-AIML	AE822FA0E5597BS0

Projects and Images for AppKit-E8-AIML-U85 [\[icon\]](#)

Projects	Build Type	Load Settings
<input type="checkbox"/> DataTest	DebugRec	none
<input checked="" type="checkbox"/> AlgorithmTest	DebugRec	image+symbols

[Edit cproject.yml](#) [Edit cproject.yml](#)

Images **Load Settings**

Debug Adapter for AppKit-E8-AIML-U85 [\[icon\]](#)

☒ J-Link Server ☒ Update launch.json and tasks.json

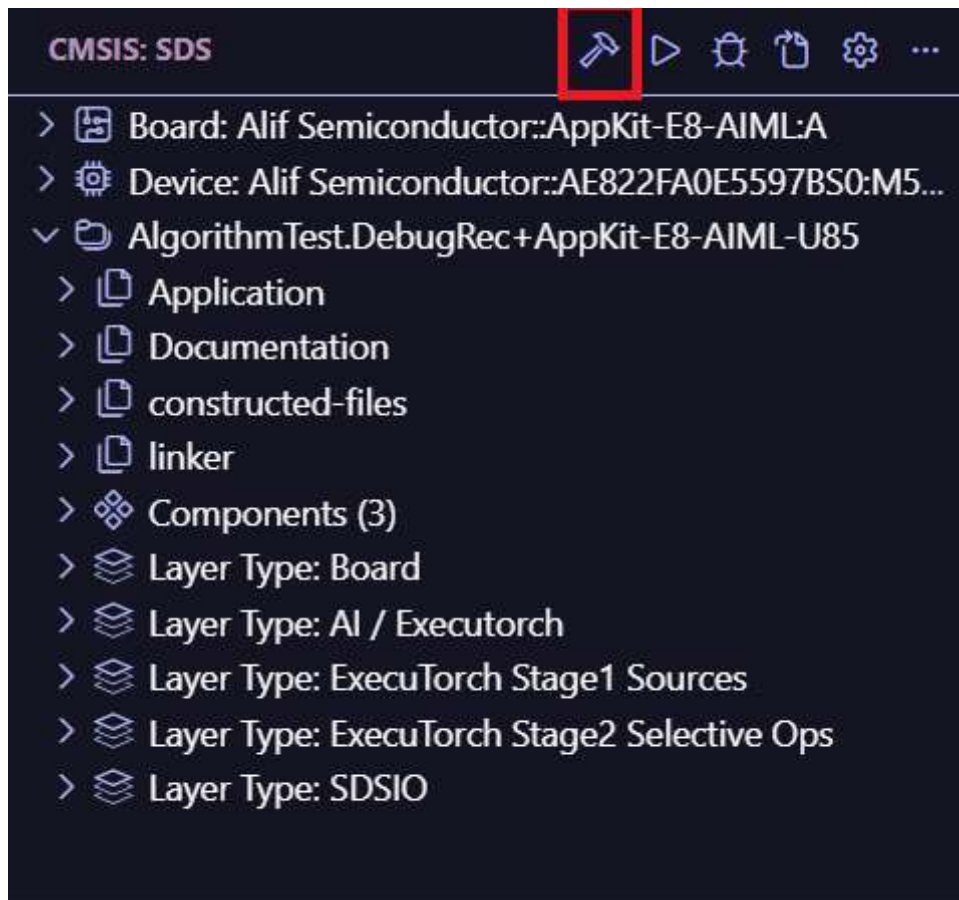
Debug Interface

☐ Telnet

Clock (kHz) 4000

Protocol SWD

- Click on the **Build** icon in Keil VS Code extension



- This will generate:
 - `.elf` file
 - `.hex` file

6. Flash the Firmware

• 6.1 Prerequisites

- J-Link Software must be installed
- Device must be connected

6.2 Flashing Steps

- Click on the **Play** icon in VS Code to flash the firmware binary



- Wait for the flashing process to complete successfully

7. Verification

- Confirm that flashing completes without errors
- Click **Reset** Check UART output (From U4) to verify correct firmware execution.

Rock-Paper-Scissors Sample Images

Note: This is an early release version. The demo functions correctly **only when displayed exactly as shown in the reference images below** and operates properly **only when the distance is less than 0.5 meters**.

