

# E2-Studio Integration – Getting Started Guide

## Using Renesas CK-RA6M5

### Description

This document covers the development and testing of the Asset Tracking demo using Renesas’s RA6T2 kit & Reality AI Tools portal. The portal is useful to generate AI models for non-visual sensing applications.

This document contains the following sections:

Lab Section	Demo Steps
Section 1	Prerequisites
Section 2	Data Collection
Section 3	Model Generation
Section 4	Testing the Models
Section 5	Appendix

Sections should be followed sequentially.

<b>Lab Objectives</b> <ul style="list-style-type: none"> <li>Gain hands-on experience on developing a simple AI demo using Renesas CK-RA6M5 + PMOD accelerometer.</li> </ul>	<b>Lab Materials</b> <ul style="list-style-type: none"> <li>Please verify you have the following materials at your lab station. <ul style="list-style-type: none"> <li>PC with Windows 11.</li> <li>Project files provided by Renesas - Reality AI team (link in section 1 of the document).</li> <li>E2-Studio IDE v2023-10 (link in section 1 of the document).</li> <li>Reality AI Tools Account</li> </ul> </li> </ul>
--	--

### Lab Sections

1	Prerequisites.....	3
2	Data Collection .....	6

3	Creating Models .....	15
4	Deploying the model .....	22
5	Appendix.....	25

# 1 Prerequisites

## Overview

This section covers the prerequisites for running the finger friction demo.

## Procedural Steps

**This lab requires the RA6M5 cloud kit and Avnet PMOD accelerometer.**

### Overview

Description    Features    Applications    Related Products

The CK-RA6M5 cloud kit enables users to securely connect to the cloud and explore the features of the Cortex®-M33 based **RA6M5 group of MCUs** and cloud services. With multiple network connectivity options, the kit provides a seamless cloud connectivity experience to most of the global cloud service providers. It is equipped with an onboard Ethernet PHY for Ethernet connectivity. The kit is available with two options for wireless connectivity - Wi-Fi connectivity using **DA16600 Wi-Fi-BLE combo module** Pmod and cellular connectivity with **RYZ014A LTE CAT-M1 module** Pmod. With the Renesas **Flexible Software Package (FSP)**, the kit has complete software stack support using FreeRTOS, Azure RTOS and other middleware stacks, thereby making it an ideal platform for efficiently developing cloud solutions and greatly reducing the time-to-market.

Visit [Renesas Cloud Solutions](#) for complete cloud solution package.



The CK-RA6M5 kit comes with:

1. CK-RA6M5 board
2. Two USB cables.
3. LTE Antenna (not used in this project).
4. Pmod expansion board (not used in this project).

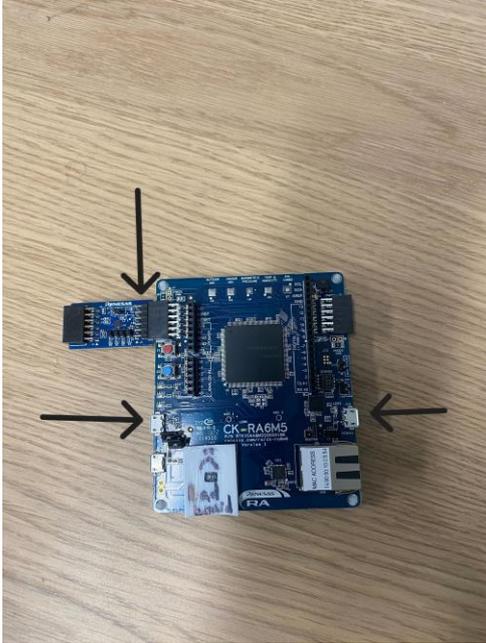
Link: <https://www.renesas.com/us/en/products/microcontrollers-microprocessors/ra-cortex-m-mcus/ck-ra6m5-cloud-kit-based-ra6m5-mcu-group#overview>

Avnet's PMOD Accelerometer board Link: <https://www.avnet.com/shop/us/products/invensense/icm-42670-p-3074457345645573040/>

Assemble the kit as shown in the next section.

Assemble the kit as shown below.

1. Connect the PMOD board to the PMOD1 connection on the board
2. Connect the PC to Micro USB on the boards (Debug on the right side of the image and USB serial on the left).



**This lab requires Renesas E2-Studio IDE 2023-010 or newer with FSP 4.5 ( till 5.2).**

Platform installer available here: [Link](#) Take note of where e2studio is installed.

**This lab requires an account on the Reality AI website.**

Login to Reality Tools (<https://portal.reality.ai/login>) using the username and password provided. Reach out to Renesas – Reality AI team members if you need credentials + instruction documents for the portal.

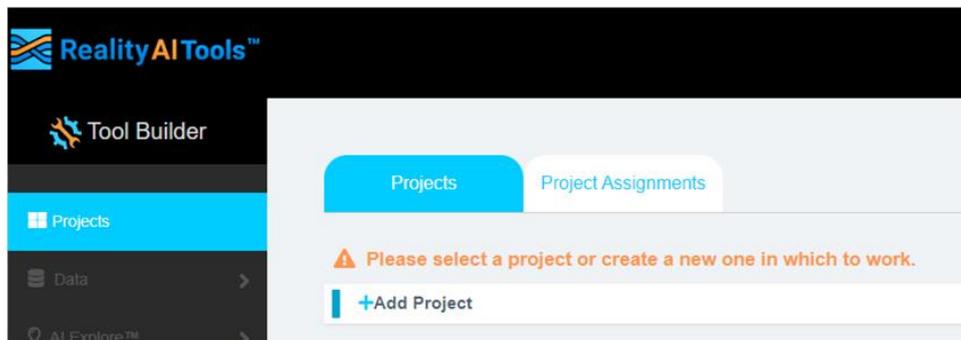


Username  
RenesasUser

Password  
.....

Stay logged in

Login



Leave the browser open in the background.

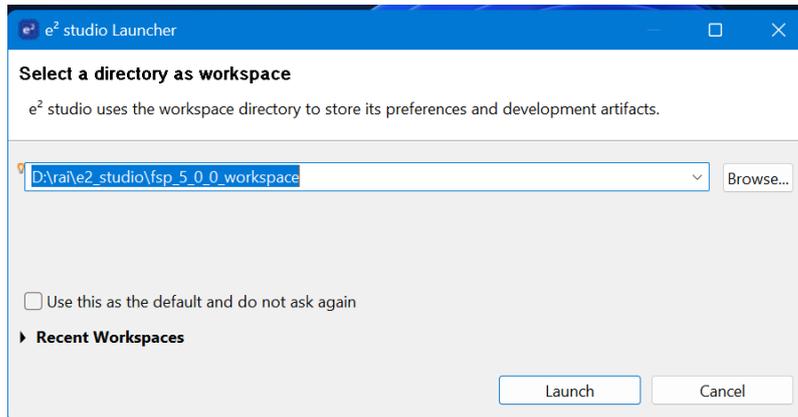
## 2 Data Collection

### Overview

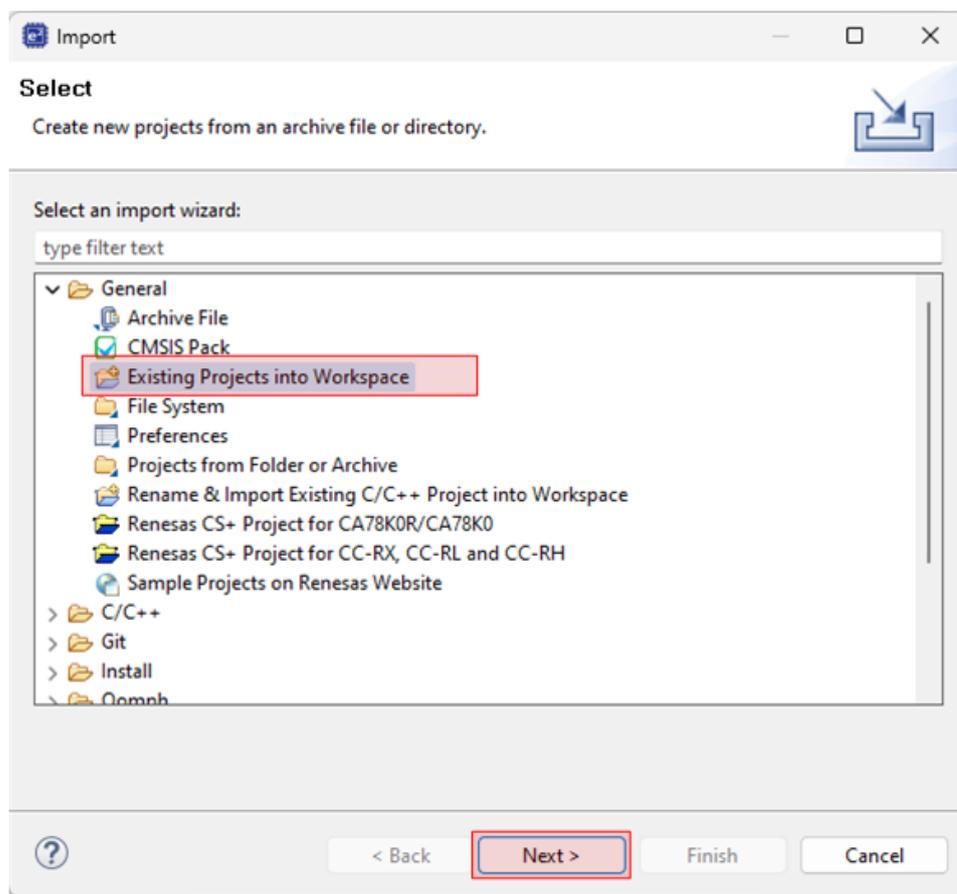
This section covers collecting data with the Reality-AI eclipse plugin using e2studios.

### Procedural Steps

Open E2 Studio IDE and select a workspace. Although the workspace can be in any folder, this lab assumes the workspace shown below. c:\e2\_projects\_4.6

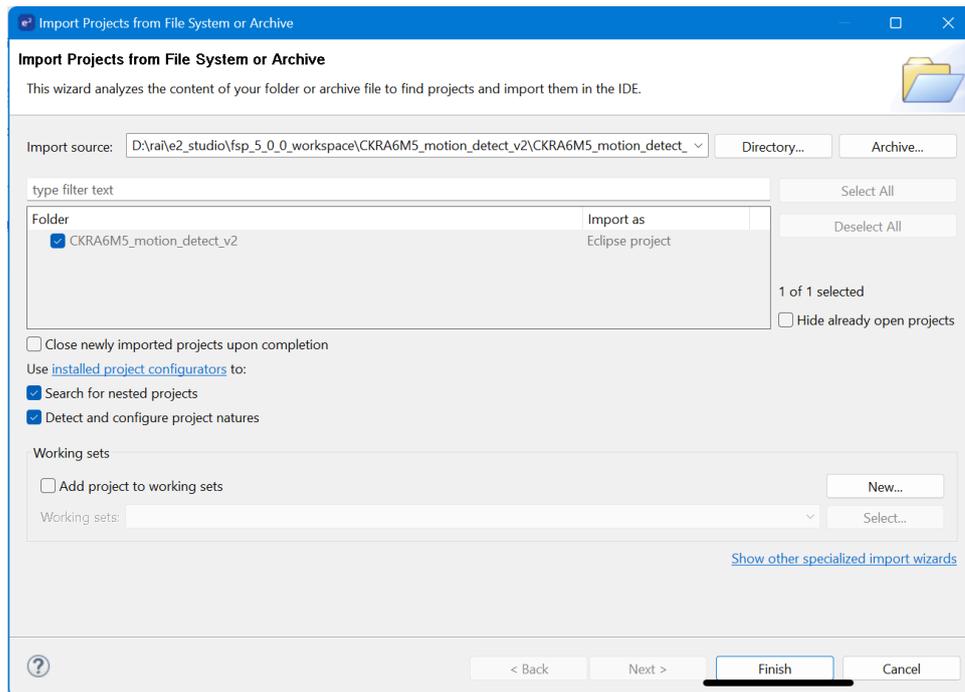


Go to File → Import... Then choose General->Existing Projects into Workspace



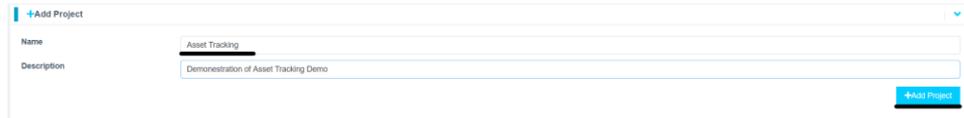
Click Next to continue.

Check “Select archive file” and navigate to the zip file for this lab. This lab contains one project. Make sure it is selected and then click Finish.

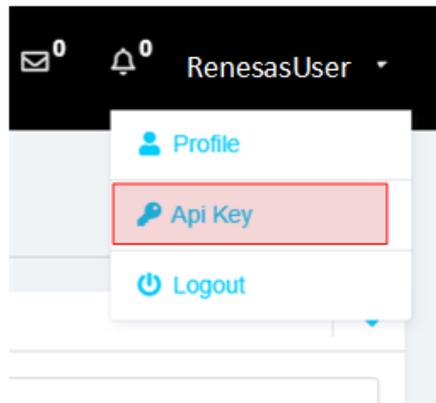


The demo project is imported into the workspace

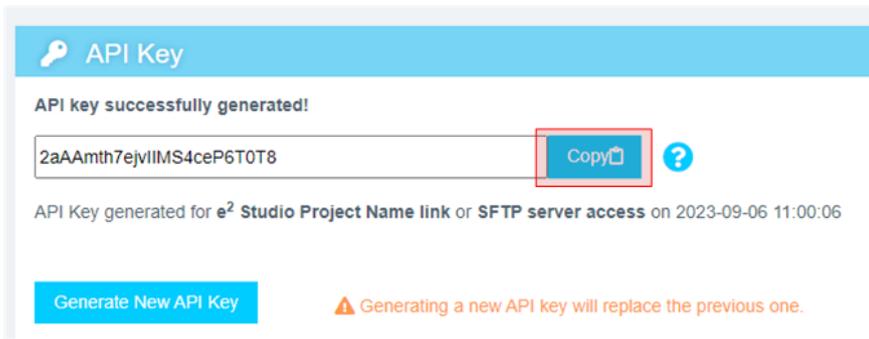
Return to the Reality-AI tool and start a new project. Click on +Add Project, give the project a name and, optionally, a description. Then click the +Add Project button in the lower right.



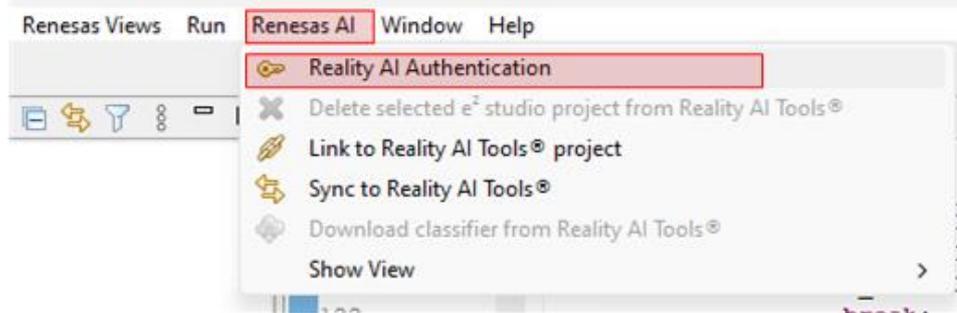
Navigate to the username on the top-right of the screen and select API Key Option.



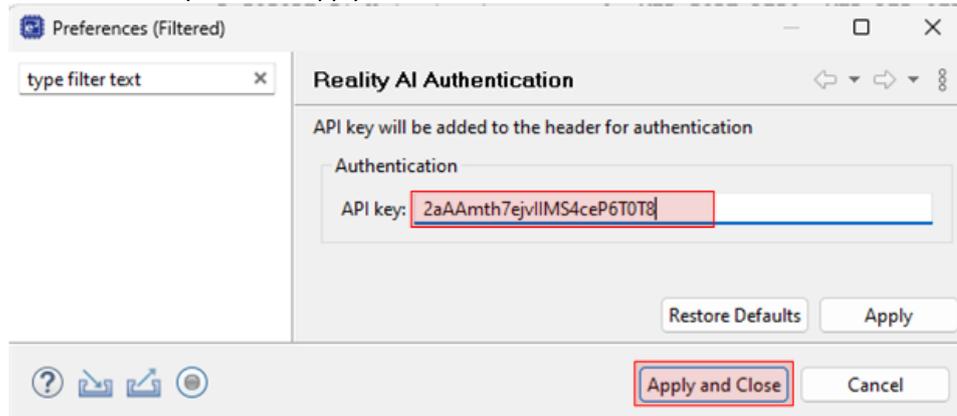
Copy this API key to the clipboard. This key will be used to connect e2 studio IDE and Reality AI Tools.



Open e2 studio and navigate to Renesas AI --> Reality AI Authentication  
*(note: If Renesas AI does not appear in the menu, consult the appendix)*

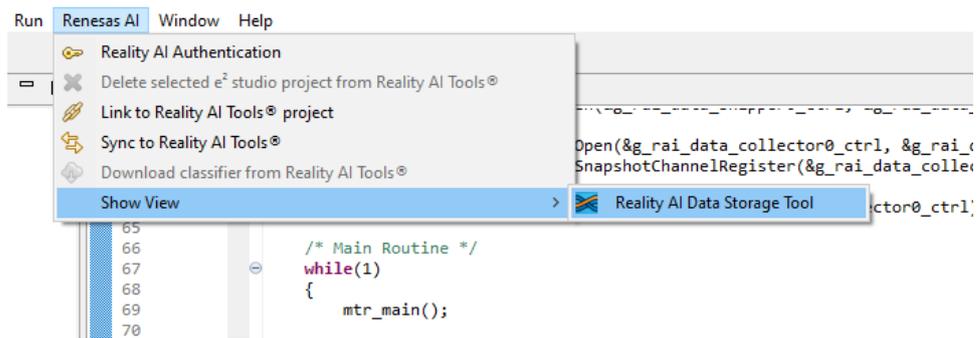


Paste the API key and click Apply and Close

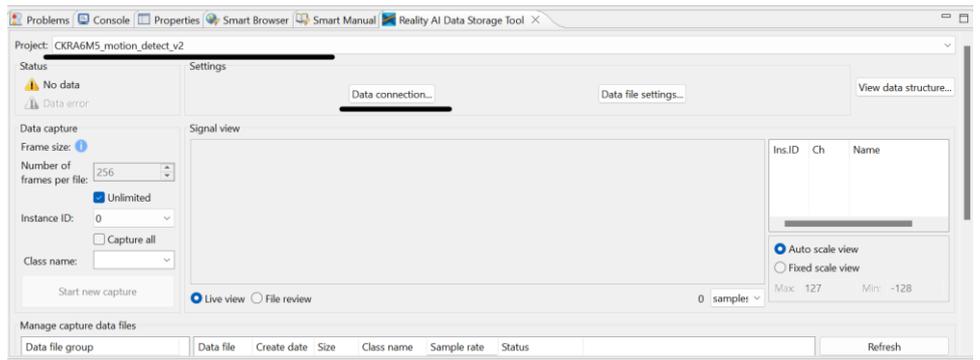


The IDE can now connect to the Reality AI site

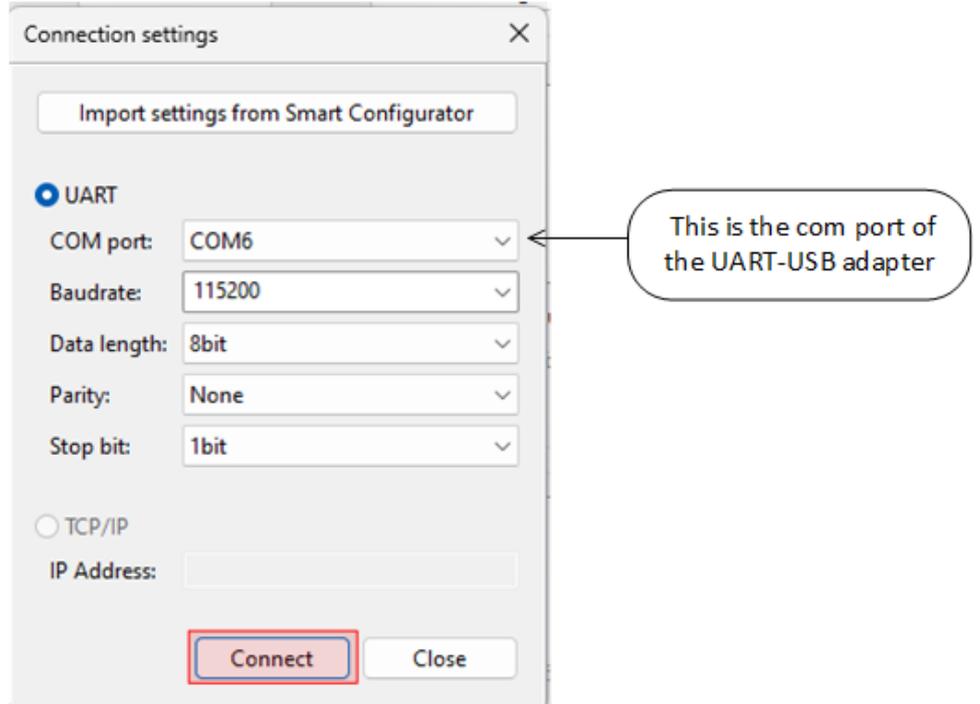
Navigate to Renesas AI --> Show View --> Reality AI Data Storage Tool



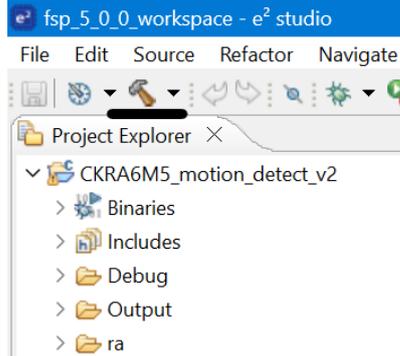
A new window will open in the bottom pane. Click-and-drag the view to the main view area by clicking on the tab and drag-drop in the tab area of the main view. Select the project in the pulldown, then click on "Data Connection" button.



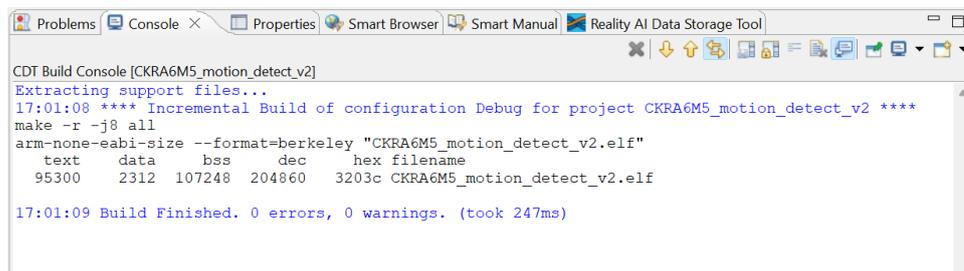
In the menu that pops up, select the COM port, baudrate (115200), and communication protocol parameters (8-N-1). Then press **Connect** --> then press **Close**.



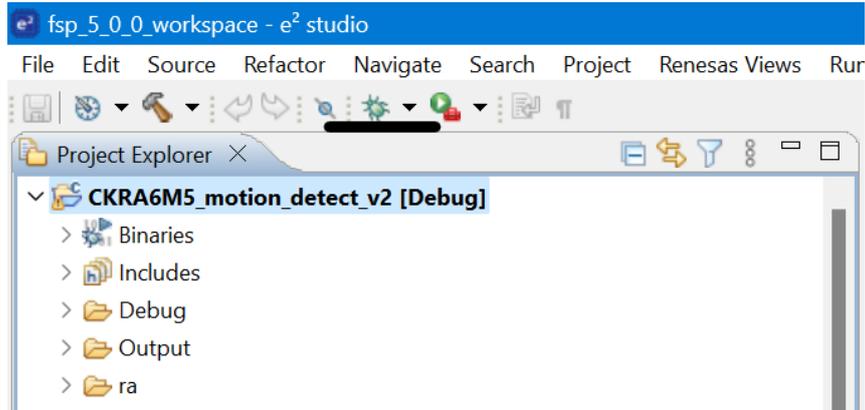
Build the project by clicking the hammer icon



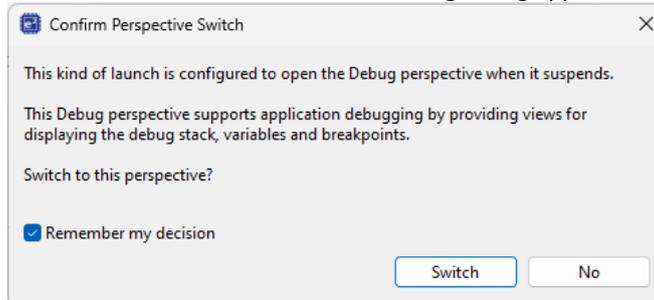
The project should build without error. There will be some warnings, these are expected.



Make sure the toggle switch is in the OFF position. Start a debug session by clicking the Debug Icon in the toolbar



Click "Allow access" if you get a warning from Windows Defender. Check the "Remember my decision" checkbox and click Switch if the following dialog appears.



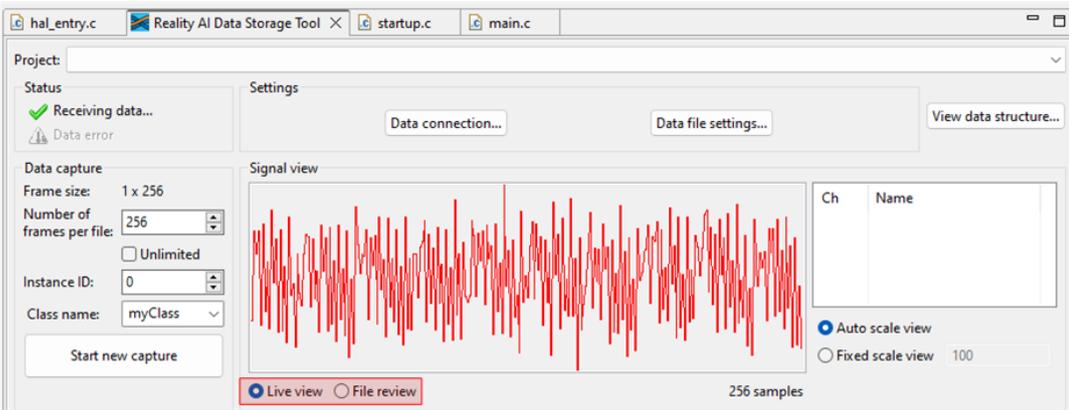
Click the Resume button twice. The program is now running



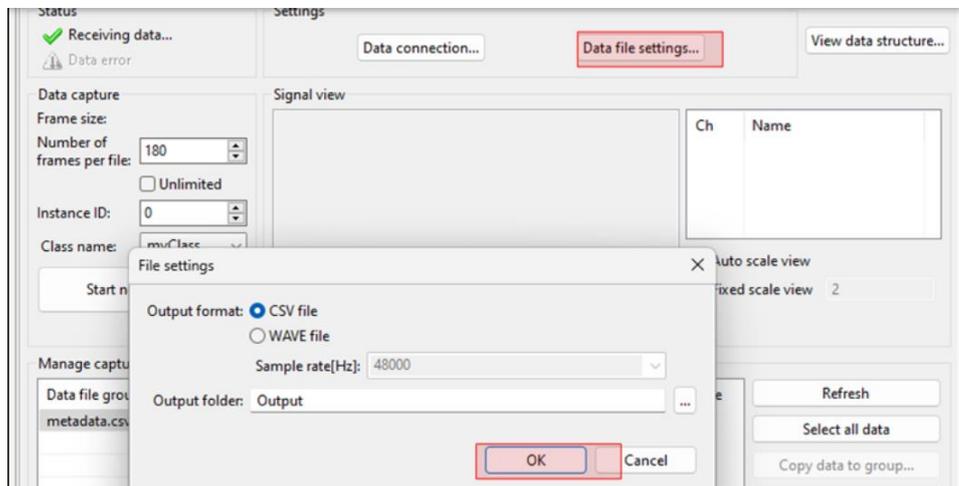
Switch the toggle switch to the ON position. The motor should now be running.

**WARNING: Never stop the debugger while the motor is running. This will cause commutation to stop, possibly leaving a coil energized. This will cause the motor to get very hot. If the motor is not spinning at this stage check the connections and review the material above.**

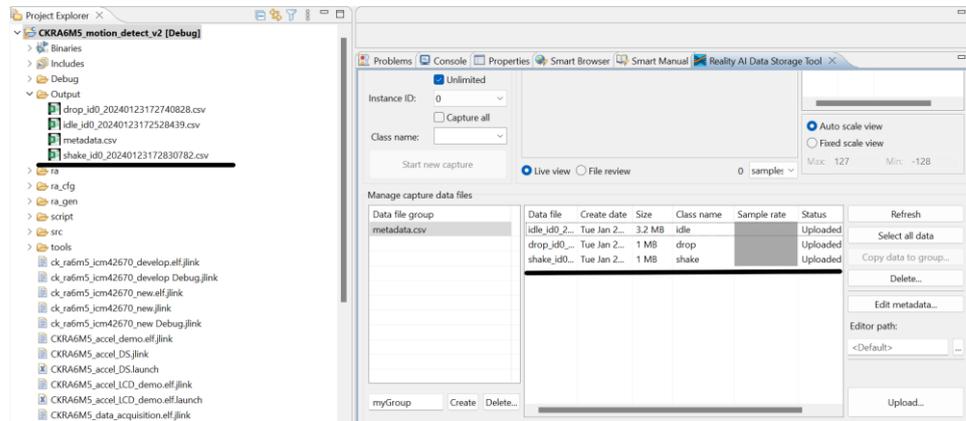
Click on the “Reality AI Data Storage Tool” view and verify the presence of a wave form in the Signal view. If there is nothing showing, click File review and then Live view (highlighted below). If there is still no waveform, recheck the prerequisite steps to make sure the connections are correct.



Click on the Data file settings and verify CVS file and Output folder. The Output folder is relative to the project. Click OK to close the dialog.

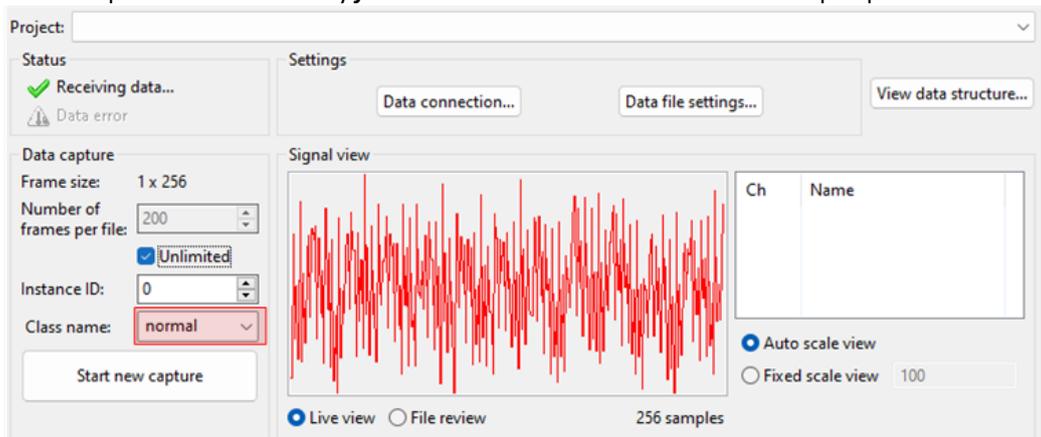


Note that the Data Storage tool reflects the data in the project's Output folder. The data currently there was collected during the lab development and includes date and time information. New data will be collected in the following sections.



There are two methods to collect data: Number of frames per file and Unlimited. If the user selects Unlimited (by checking the Unlimited checkbox) data collection will occur until the user manually stops the collection process. Otherwise (Unlimited box unchecked) the collection will automatically stop once the Number of frames per file is reached. For this project, setting the Number of frames per file to 200 results in approximately 52 seconds of data. The remainder of this lab assumes Unlimited capture.

In the Reality AI Data Storage Tool, check the Unlimited checkbox. Enter the Class name “Normal” or “idle” Keep the board stationary **for 45 to 60 seconds** and then click the Stop Capture button.



45 to 60 second's elapses...



A new data file appears in the output folder

Manage capture data files

Data file group	Data file	Create date	Size	Class name	Sample rate	Status
metadata.csv	idle_id0_2...	Tue Jan 2...	3.2 MB	idle		Uploaded
	drop_id0_...	Tue Jan 2...	1 MB	drop		Uploaded
	shake_id0...	Tue Jan 2...	1 MB	shake		Uploaded

Change the Class name to “Shake” and repeat the above steps **while lightly shaking the board.**

Once done, again repeat the step by **dropping the board on the table from a height of 4-7 inches.** (Make sure you don’t damage the board. Do it on a softer surface if possible).

This concludes the data capture portion.

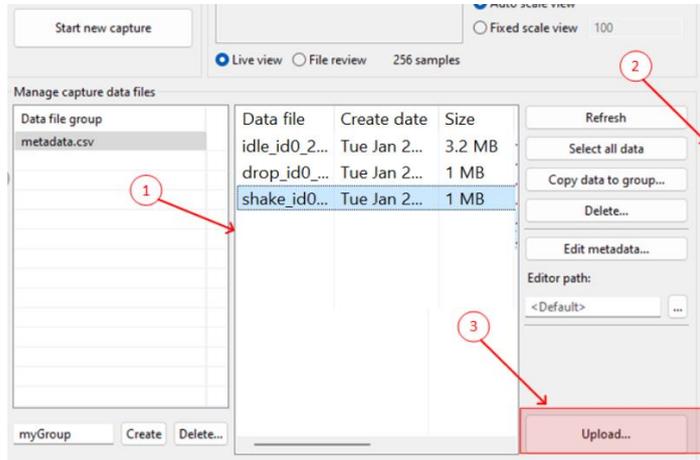
Click the red square icon to terminate the debug session.



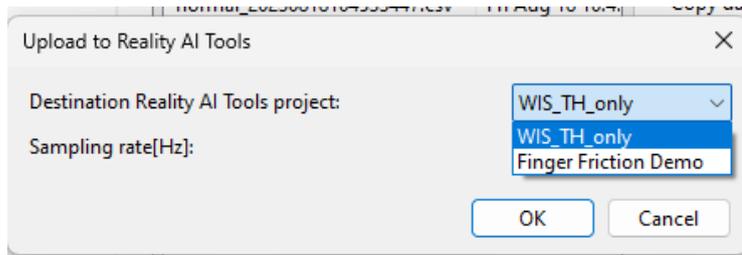
(If you do not see the terminate icon, select the Debug perspective in the upper right corner)

Follow these steps to upload the data files collected in the last section:

1. Select the two files using click + ctrl-click
2. Scroll the view to expose the Upload... button
3. Click the Upload... button

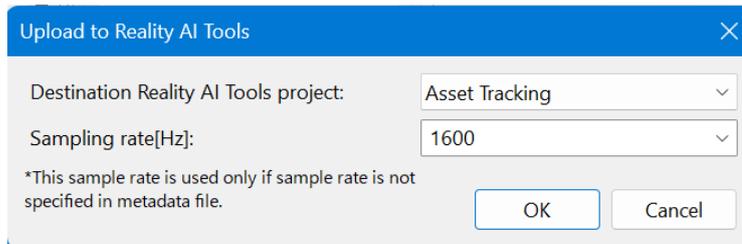


A dialog box should appear with the projects currently defined on the Reality AI website.



Select the appropriate project

Set the Sampling rate to 1600 and press OK



The data files, along with the metadata.csv file will upload, and a confirmation dialog should appear.

**You have finished this section.**

### 3 Creating Models

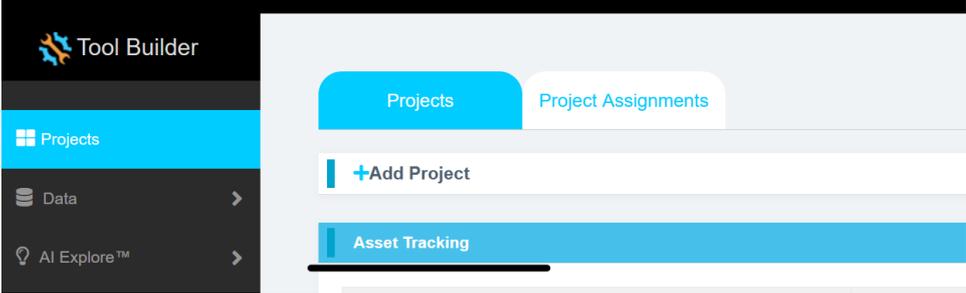
#### Overview

This section covers creating AI models on the collected dataset. A Reality AI Tools account is required.

#### Procedural Steps

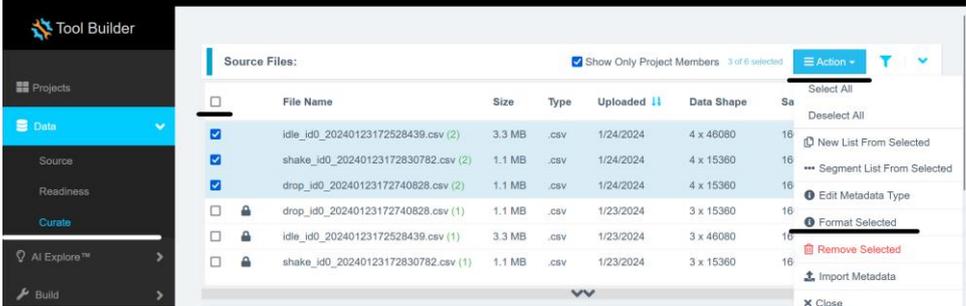
If not already logged in to Reaity AI Tools, use this link: <https://portal.reality.ai/login> with the provided password. Reach out to Renesas – Reality AI team members for credentials + instructions documents for the portal if needed.

**Format the data.** Click on the Asset Tracking project to set active.



Click on the Data tab on the left to expand and follow these steps:

1. Click on Curate.
2. Click on Source Files to expand.
3. Check box next to File Name to select all files
4. Click on the Action button.
5. Click on Format Selected



This opens a dialog box.

Follow these steps:

1. Click on Data to expand the drop down
2. Use the scroll bar to scroll down (or mouse wheel)

Define file format for all selected files

Use this for all selected .csv files with this extension

Display raw data  Display issues only

#1	Data -	#2	Data -	#3	Data -	#4	Data -
1	Index	1 ch	2 ch	3 ch			
2	0	-0.275146484375000000	0.087329104542732240	10.308423042297363000			
3	1	-0.277539074420928960	0.082543946802616120	10.296460151672363000			
4	2	-0.277539074420928960	0.082543946802616120	10.319190025329590000			
5	3	-0.275146484375000000	0.071777343750000000	10.328760147094727000			

Label Row  
Check if CSV has a header row

Sample Rate  
Sample Frequency in Hz:

Delimiter  
Character used to separate columns:

European Decimals  
Check if numbers use commas as a decimals. IE: 1.234.567,89

3. Select Ignore (this column is not part of the data)

Display raw data  Display issues only

5	3		
		Numerical MetaData	-0.017041284590959550
		Sequential #'s / Time Code	
		Date and Time	
		Date only	
		Time only	
		Ignore	

A red circle with the number '3' and an arrow points to the 'Ignore' option in the dropdown menu.

4. Click confirm

Verify that the data shape is now 3 X {num of samples} and the Sample rate is 1600Hz.

Preprocess/segment the data files. Click the Action button and select **\*\*\* Segment List from Selected**

Source Files:  Show Only Project Members 3 of 6 selected

	File Name	Size	Type	Uploaded	Data Shape	Sampl
<input checked="" type="checkbox"/>	idle_id0_20240123172528439.csv (2)	3.3 MB	.csv	1/24/2024	3 x 46080	1600 H
<input checked="" type="checkbox"/>	shake_id0_20240123172830782.csv (2)	1.1 MB	.csv	1/24/2024	3 x 15360	1600 H
<input checked="" type="checkbox"/>	drop_id0_20240123172740828.csv (2)	1.1 MB	.csv	1/24/2024	3 x 15360	1600 H
<input type="checkbox"/>	drop_id0_20240123172740828.csv (1)	1.1 MB	.csv	1/23/2024	3 x 15360	1600 H
<input type="checkbox"/>	idle_id0_20240123172528439.csv (1)	3.3 MB	.csv	1/23/2024	3 x 46080	1600 H
<input type="checkbox"/>	shake_id0_20240123172830782.csv (1)	1.1 MB	.csv	1/23/2024	3 x 15360	1600 H

Action menu options: Select All, Deselect All, New List From Selected, **Segment List From Selected**, Edit Metadata Type, Format Selected, Remove Selected, Import Metadata, Close

### Why Segmentation?

One of the main purposes of generating models from Tools is to deploy them to a variety of Renesas MCU's. To that effect, these models must process live data within a resource constrained environment. So, for practical purposes, a model might be looking at 1 second, 500 ms, or even a smaller window length to make quick predictions on Realtime data. As opposed to a few seconds (or minutes or hours) long data stream. To mimic that effect, we break down the raw training data and feed that to the model generation engine to start learning what it is going to see in a live (production) setting.

Assign window length, overlap, and provide name to the list. Follow these steps:

1. Set Window Length to 256 or 512
2. Click the 50% button to select a 50% offset
3. Give the list a meaningful name
4. Click Submit

Segmentation Method: Sliding CSV Window

Sample Rate: 1000 Hz

Parse a CSV file into a new sample list by **1** using window

Target:  File metadata target column ( type: Class )

Window Length: 256 datapoints, 256 ms

Offset: 128 datapoints, 128 ms

Number of rows between sample start points: **2** 50% (selected), Non-Overlapping, All Shifts

798 estimated samples

Output Sample List: **3** 256\_50\_percent\_overlap (List name available) **4**

Buttons: Cancel, Submit

**The number of estimated samples will be dependent on the amount of data collected. So if your estimated samples don't match the below screenshot, its still okay.**

**Window Length:** The window length determines how much data will be considered by the AI to decide on a classification.

**Offset:** The offset determines how far ahead from the start of the last window the parser moves before creating a new window.

**General Guide: 50% Overlap** is usually a good compromise between covering starting-point variations in the data and too much redundancy.

Use **non-overlapping** windows when you have a great deal of data, with longer offsets. Typically, users will do initial exploration and training on a subset of the available data.

Use **All Shifts** (offset = 1) for testing after you have a suitable candidate classifier, and you want to simulate performance on a stream of new data arbitrarily sampled.

After successfully creating the segmented list, navigate to AI Explore → Classes

List Name	List Type	Data Shape	N Samples	Target Range	Created	Modified	Comments
256_50_percent_overlap	Stream Segments	1 x 256	798	2 Classes	09-13-2023	09-13-2023	<a href="#">segmentationList</a>

Reality AI Tools has options for creating 3 types of AI Models:

**Classes:** When there is labeled categorical data (**option used in the tutorial**). You might have noticed that we uploaded perfectly labeled data in section 1. This is needed for classification models as supervised learning is being performed.

**Values:** When discrete int or float values are used instead of categories. (Example: What is the exact temperature of a machine? Or What is the exact value of tire pressure of a car?). This is also supervised learning.

**Anomalies:** This is an Anomaly Detection module. It is a semi-supervised model where the user only needs examples of Normal data to create a baseline model.

Click on the newly created list and then click on “Start Exploring” to start the feature discovery and model training process.

List Name	List Type	Data Shape	N Samples	Target Range	Created	Modified	Comments	Status
256_50_percent_overlap	Stream Segments	1 x 256	798	2 Classes	09-13-2023	09-13-2023	<a href="#">segmentationList</a>	

Once the explore finishes, select the highest performing model (we recommend model with feature space: Spectral Magnitude) and click on the Create Base Tool button. Hover over each Explanation to get more information about the model.

Favorites	Complexity	KFold Accuracy		Training Separation		Create Base Tool	Explanation	Confusion Matrix
		Overall %	Worst %	Overall %	Worst %			
★		95	95	95	95			
★		82	81	83	82			
★		100	100	100	100			

**What is happening in the background:**

Clicking on the Start Explore button will turn the Reality AI algorithm loose on your data. It will create several optimized feature sets and machine learning models that best fit the classification problem and then summarize each model's accuracy and resource consumption.

**What is the algorithm doing:** AI Explore first creates a **balanced subset** of the sample list (meaning a list in which each class has equal numbers). If the sample list is exceptionally large, it may also sub-sample for a shorter processing time. Explore then runs through a procedure in which the Reality AI algorithms attempt to discover the best possible set of features and machine learning parameters for separating the different training classes represented in the data. The feature sets that are most promising are then used to construct machine learning models, which are trained on the sub-sample and put through a K-Fold validation.

Only the best performing results are displayed. Several hundred different feature sets and machine learning models are compared in a typical AI Explore run.

**What is K-Fold validation:** K-Fold is where a given data set is split into a K number of sections/folds where each fold is used as a testing set at some point. Let us take the scenario of 10-Fold cross validation(K=10). Here, the data set is split into 10 folds. In the first iteration, the first fold is used to test the model and the rest are used to train the model. In the second iteration, the 2nd fold is used as the testing set while the rest serves as the training set. This process is repeated until each fold of the 10 folds has been used as the testing set. **We use K=10 in the AI Explore page.**

Provide a name or use the one suggested. Click Add.

Create Base Tool
✕

---

**Name**

---

**Description**

---

Add

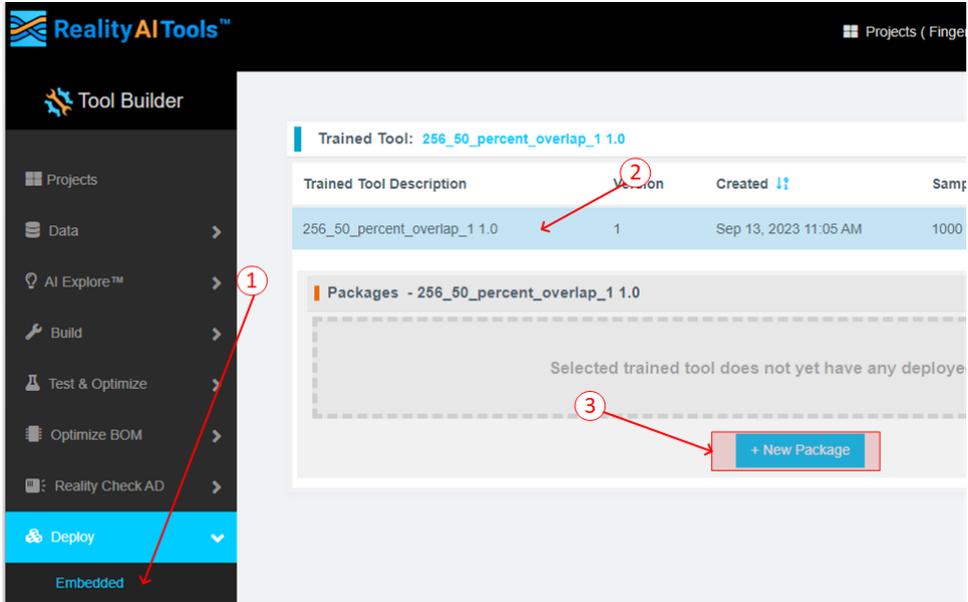
The icon will change, indicating the base tool has been created.

✓ Explore: Complete 100 %

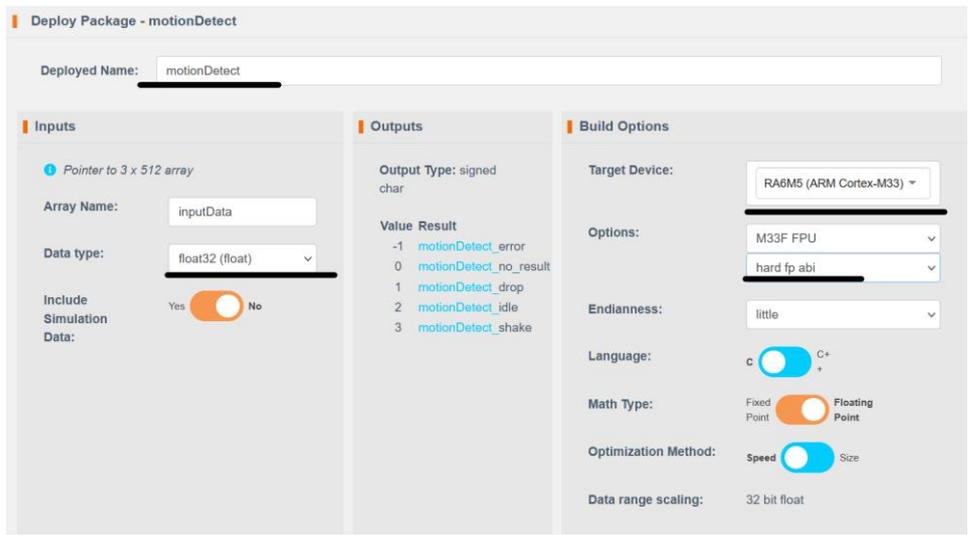
KFold Accuracy		Training Separation		Create Base Tool
Overall %	Worst %	Overall %	Worst %	
95	95	95	95	
82	81	83	82	
100	100	100	100	<span style="border: 1px solid red; padding: 2px 5px;"></span>

Now that the model is ready to be deployed, Follow these steps to produce a new package.

1. Click on Deploy->Embedded.
2. Click on the Trained Tool Description list
3. Click on + New Package



Set the options as indicated. The deployed name should be ***motionDetect\_BC1\_v12*** to match the code in e2studios. Choosing another name will require additional edits to the code. Click ***Generate New Package*** after setting the options.



The package will take a few minutes to generate.

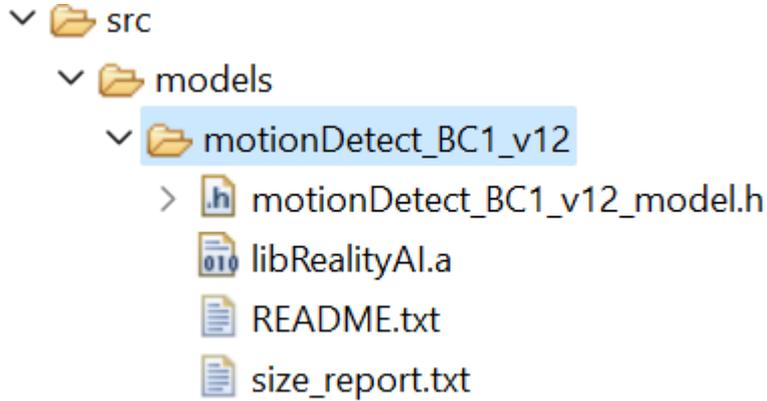
It will take ~10-15 minutes for the model to be available for download. Once ready, download the zip file using the highlighted button.

Package Date	input Data	Parameters	Target	Math Type	Download
Sep 13, 2023 11:21 AM	float *inputData	internal	RA6T2 (ARM Cortex-M33)	float 32	

There are 7 files in the downloaded archive. Copy and paste all files except example\_main.c & readme.txt to the src directory of the E2-Studio project workspace (src --> models), overwriting the current files.

**Note: file names may be different depending on the project name assigned in tools.**

**In the files below, the model function call is specified in motionDetect\_BC1\_v12\_model.h**



**You have finished this section.**

## 4 Deploying the model

### Overview

This section covers deploying the model on hardware.

### Procedural Steps

Open `hal_entry.c` and import the header file. **If the deployed model name was `motionDetect_BC1_v12_model.h` then no changes are required.** Otherwise, edit the `#include` to the deployed model's header file.

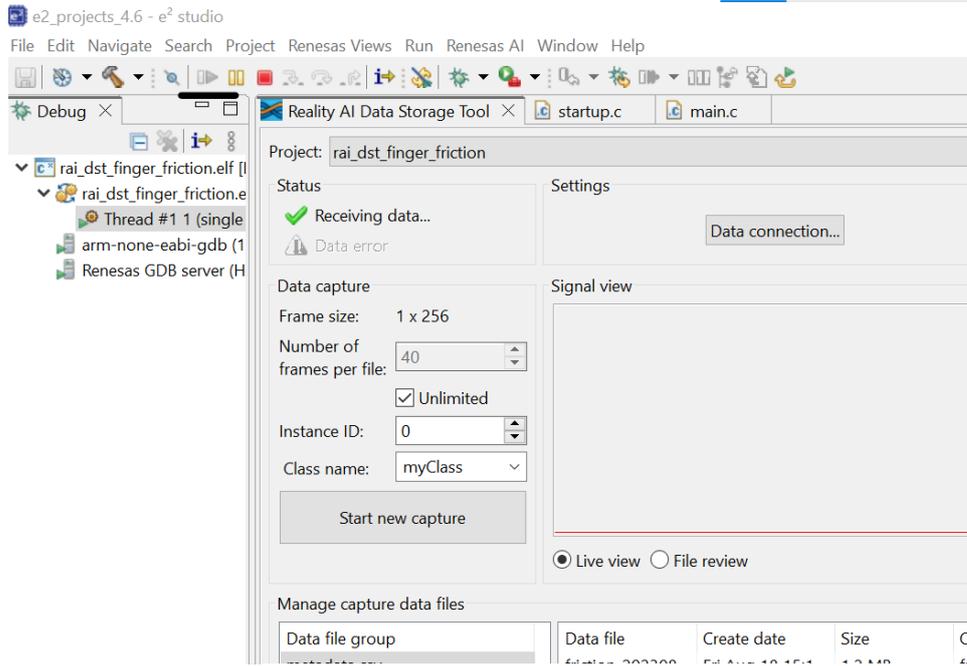
```
hal_entry.c ×
2      * Copyright [2020-2022] Renesas Electronics Corporation and/or its affiliates.
20
21     #include "hal_entry.h"
22     #include "rm_icm42670/rm_icm42670.h"
23
24     #include "models/motionDetect_BC1_v12/motionDetect_BC1_v12_model.h"
25
26     #define RM_ICM42670_EXAMPLE_DELAY_50MS 50
27     #define RM_ICM42670_EXAMPLE_DELAY_1US 10
28     #define RM_ICM42670_EXAMPLE_IRQ_ENABLE 0
29
30     void R_BSP_WarmStart(bsp_warm_start_event_t event);
31     fsp_err_t rm_icm42670_irq_open (rm_icm42670_ctrl_t * const p_api_ctrl);
32
33     bool g_i2c_flag = 0;
```

Modify the following lines of code if the deployed model name was different. Values can be found in the header file included in the last step.

```
434     pred_data.pred_class = motionDetect_BC1_v12_classifier_predictionfloat(input
435     // err = R_SCI_UART_Write(&g_uart5_ctrl, (void*)&pred_data, sizeof
436
437
438     switch (pred_data.pred_class){
439         case motionDetect_BC1_v12_drop: //motionDetect_E2C1_engSeg_v2_drop :
440             blinkDropLed();
441             break;
442         case motionDetect_BC1_v12_in_motion: //motionDetect_E2C1_engSeg_v2_he
443             inMotionLed();
444             break;
445         case motionDetect_BC1_v12_wave: //motionDetect_E2C1_engSeg_v2_wave :
446             waveLed();
447             break;
448         case motionDetect_BC1_v12_idle: //motionDetect_E2C1_engSeg_v2_idle :
449             idleLed();
450             break;
451         case motionDetect_BC1_v12_circle: //motionDetect_E2C1_engSeg_v2_idle
452             circleLed();
453             break;
454         default:
455             clearLeds();
```

Edit lines above by copying the class names from header file included above. **Again, no edits necessary if the deployed model name & number of class configuration matches the above structure.**

Build & debug the project as before. Click on the Resume button twice. Now the board is running on inference mode. Perform the board shake, idle and drop tests and see the LED's change color.



**What should I do if model is not performing well?**

Collect some more data and retrain the model. Usually additional data collection helps in creating better performing models across different conditions.

Double check if the data collection method and testing method are the same. Collecting more data across different variations should help.

**You have finished this section.**

## 5 Testing the Model

### Overview

This section covers testing the model on hardware.

### Procedural Steps

The default model will cover the following conditions:

1. **Idle** – Keep the board on the table in idle condition. Clear/ no LED's.
2. **In-motion** – Move the board around to generate a motion gesture. You will see MEGENTA color on RGB for this.
3. **Circle** – Move the board in a small circle. This will have BLUR color in RGB.
4. **Wave** – Move the board in the form of water wave (or sine wave). This will have YELLOW color in RGB led.
5. **Drop** – Drop the board from the height of 4-7 inches on a soft pad on the table (example: you can use a mouse pad). Megenta, Yellow and Red blinking LED's for this motion.

We prefer if you use a mouse pad or some soft material on the table to experiment with the demo. Also dropping it from a bigger height (> 30 inches) may loosen the connection of the PMOD sensor and risk damaging the board.

Note: If you change the class and their respective LED configuration by editing Lines 440, 443, 446, 449, 452 & 455. Remove or call each LED configuration for a different class if you prefer.

```

434     pred_data.pred_class = motionDetect_BC1_v12_classifier_predictionfloat(input
435     //     err = R_SCI_UART_Write(&uart5_ctrl, (void*)&pred_data, size
436
437
438     switch (pred_data.pred_class){
439     case motionDetect_BC1_v12_drop: //motionDetect_E2C1_engSeg_v2_drop :
440         blinkDropped();
441         break;
442     case motionDetect_BC1_v12_in_motion: //motionDetect_E2C1_engSeg_v2_he
443         inMotionLed();
444         break;
445     case motionDetect_BC1_v12_wave: //motionDetect_E2C1_engSeg_v2_wave :
446         waveLed();
447         break;
448     case motionDetect_BC1_v12_idle: //motionDetect_E2C1_engSeg_v2_idle :
449         idleLed();
450         break;
451     case motionDetect_BC1_v12_circle: //motionDetect_E2C1_engSeg_v2_idle
452         circleLed();
453         break;
454     default:
455         clearLeds();

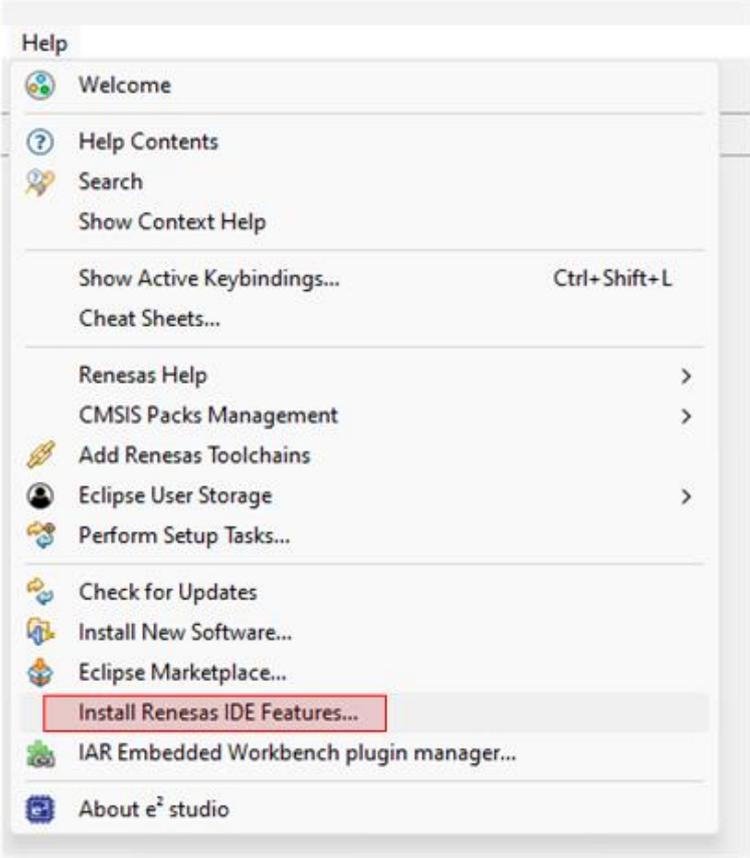
```

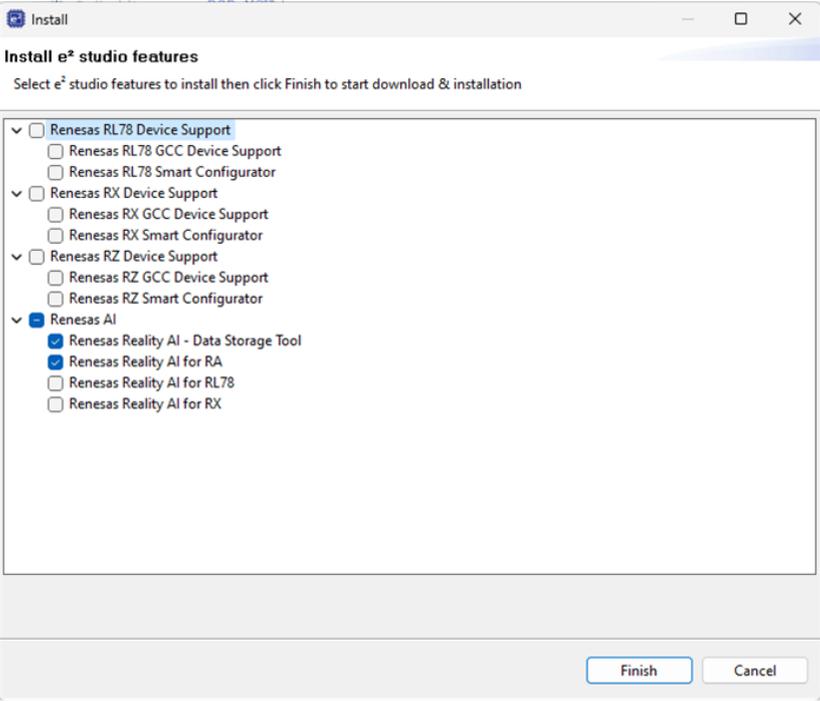
## 6 Appendix

### Overview

If the Reality AI items do not appear in the title bar perform the following steps.

### Procedural Steps

	<p>Click on Help -&gt; Install Renesas IDE Features...</p>  <p>The screenshot shows the Eclipse IDE Help menu. The menu items are: Welcome, Help Contents, Search, Show Context Help, Show Active Keybindings... (Ctrl+Shift+L), Cheat Sheets..., Renesas Help, CMSIS Packs Management, Add Renesas Toolchains, Eclipse User Storage, Perform Setup Tasks..., Check for Updates, Install New Software..., Eclipse Marketplace..., Install Renesas IDE Features... (highlighted with a red box), IAR Embedded Workbench plugin manager..., and About e<sup>2</sup> studio.</p>
	<p>Select the Reality-AI features and click Finish.</p>

	 <p><b>Install e² studio features</b> Select e² studio features to install then click Finish to start download &amp; installation</p> <ul style="list-style-type: none"><li>▼ <input type="checkbox"/> Renesas RL78 Device Support<ul style="list-style-type: none"><li><input type="checkbox"/> Renesas RL78 GCC Device Support</li><li><input type="checkbox"/> Renesas RL78 Smart Configurator</li></ul></li><li>▼ <input type="checkbox"/> Renesas RX Device Support<ul style="list-style-type: none"><li><input type="checkbox"/> Renesas RX GCC Device Support</li><li><input type="checkbox"/> Renesas RX Smart Configurator</li></ul></li><li>▼ <input type="checkbox"/> Renesas RZ Device Support<ul style="list-style-type: none"><li><input type="checkbox"/> Renesas RZ GCC Device Support</li><li><input type="checkbox"/> Renesas RZ Smart Configurator</li></ul></li><li>▼ <input checked="" type="checkbox"/> Renesas AI<ul style="list-style-type: none"><li><input checked="" type="checkbox"/> Renesas Reality AI - Data Storage Tool</li><li><input checked="" type="checkbox"/> Renesas Reality AI for RA</li><li><input type="checkbox"/> Renesas Reality AI for RL78</li><li><input type="checkbox"/> Renesas Reality AI for RX</li></ul></li></ul> <p>Finish Cancel</p>
	<p>Restart e2studios when prompted.</p>
<p><b>You have finished this section.</b></p>	

**END OF LAB  
THANK YOU**