Renesas RA family

RA6T1 Motor Failure Detection Example by TensorFlow Lite for Microcontroller

Introduction

This document describes AI based failure detection example at BLDC motor system with Google TensorFlow Lite for microcontroller. With Renesas RA6T1 Motor Starter Kit, supported software tools, and external Motor Bench, small AI running on RA6T1 eastly detects anomaly condition when the motor system encounters hardware problem. this example will show developer how AI model will be created and integrated into real system and will become first step to design predictive maintenance solution with real system.

Because the example is reference only and Renesas Electronics does not guarantee the operations resulting from use of these instructions.

Supported Kit

RSSK-RA6T1

Supported FSP Version

FSP v 2.0.0 or later

Contents

1. Overview

Figure 1-1 Motor Failure Detection Example System Block Diagram shows the high-level system block diagram of the RA6T1 Motor Failure Detection Example with TensorFlow Lite for micro. This is an e-AI based motor system containing the learned DNN and brushless DC motor control MCU software. The judgment result is displayed on the PC software.

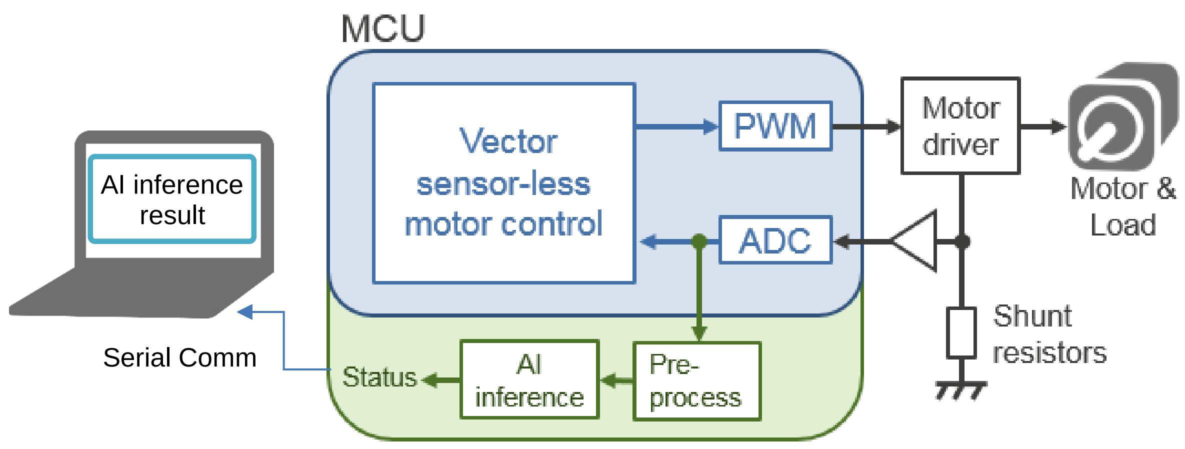


Figure 1-1 Motor Failure Detection Example System Block Diagram

1.1 Pre-Processing

The Pre-Processing in this example performs the following operations.

1. Collect AD converter values of the three-phase current and generate an FFT frame.
2. Pre-processing before learned DNN input data
   A. FFT processing of data frames (frequency spectrum generation)
   B. Feature point extraction from frequency spectrum (learned DNN input data generation)
3. AI inference

The system’s brushless DC motor control employs the sensorless vector control method to monitor the 3 shunt current control with the A/D converter. In this system, focusing on the fact that the waveform of the 3 shunt current changes depending on the state of the motor, this 3 shunt current is used as the input of trained DNN. A/D conversion values are accumulated for a fixed time to obtain waveform data on the time axis.
In input data pre-processing, a frequency spectrum is generated via FFT making it easier for AI to detect feature points of the 3 shunt current waveform. The FFT inputs data units with coefficient of 2 as one frame, but also generates a frame to partially overlap the preceding and following frame to detect changes at frame breaks. This is a common method often referred to as “overlap analysis.” In addition, in the e-AI system with limited storage area, reduction of the DNN network layer is a benefit, allowing extraction and use of areas around the input data feature point.

1.2 AI Inference by TensorFlow Lite for microcontroller

TensorFlow Lite for Microcontrollers (TFLu) by Google is tool for embedded engineer to run TensorFlow models at microcontroller. it is portable, small memory footprint and supporting useful features for embedded AI.

here is list of features.

Has compatibility with TensorFlow
Made for Microcontrollers: it supports BareMetal system and run without RTOS.
Open source project: source file, examples can be cloned from GITHUB. Supporting ARM CMSIS NN
Low memory footprint and easy to optimize system: operations can be selected to reduce memory.
Supports Post-Training Quantization
C++11: MCU Tool chain can support
Further detail can be found at https://www.tensorflow.org/lite/microcontrollers

In this example, simple AI by TFLu is made to detect normality and anomality by following layers.

(1) Input layer FFT-processed U-phase shunt current data is input to the input layer
(2) Hidden layer The hidden layer uses the fully connected layer.
(3) Output layer The output layer outputs the probability of normality and anomality.

Figure 1-2 shows the AI model configuration.

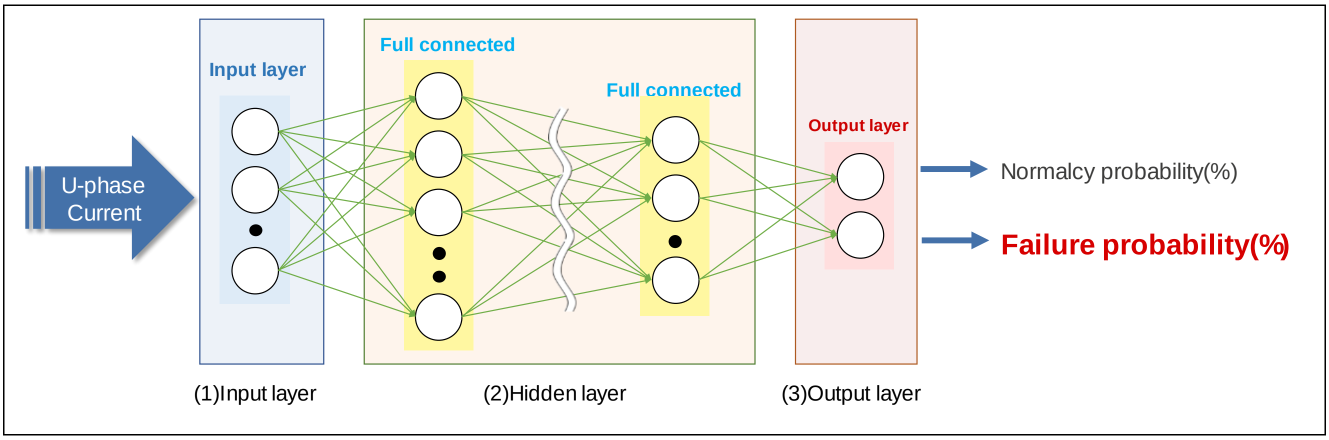


Figure 1-2. AI Model Configuration

2. Hardware Configuration

This section discusses hardware configuration of the RA6T1 Motor Failure Detection Example demo. This example is based on the hardware described in User's Manual [2]: "Motor Control Evaluation System for RA Family - RA6T1 Group" and additional two hardware for communication and motor load. All components can be shown at

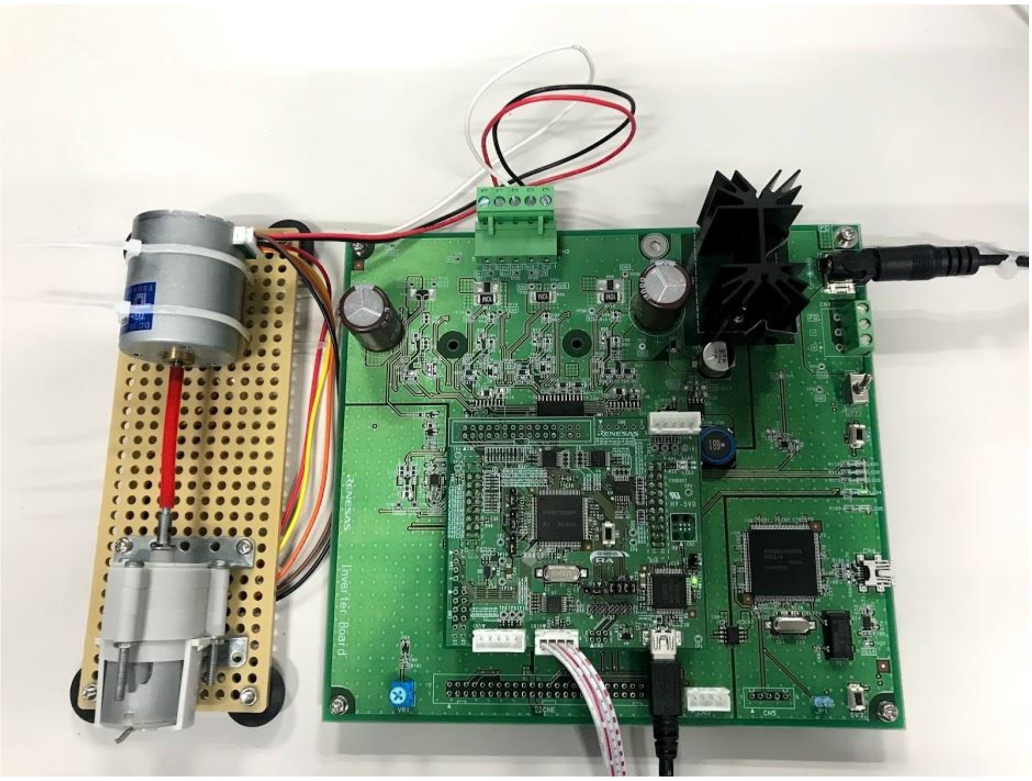


Figure 2-1. RA6T1 RSSK with Motor Bench for demonstration

Item	Name	Manufacturer	Spec/Model No. etc.	QTY
Evaluation board	RA6T1 CPU card	Renesas	RTK0EM0000B10020BJ	1
	Inverter board	Renesas	RTK0EMA170C00000BJ	1
	24V AC adaptor	(general)	refer to Inverter board doc	1
	USB serial converter module	FTDI	FT232RL	1
Motor	Brushless DC motor	Tsukasa Electric Co. Ltd.	Tsukasa Electronics Co. Ltd. TG55-L-KA 24V (bundled with Inverter board)	1
Motor bench	Universal plate	Tamiya Inc.	Item No:70098	1
	Planetary gear box set	Tamiya Inc.	Item No:72001 Use one 4:1 gear unit	1
	Rubber foot	(general)	―	4
	Air tube	(general)	External diameter: 4mm, internal diameter 2.5mm Cut to 52mm	1
	Banding band	(general)	Width: 2mm	4

Table 2-1 List of Hardware

2.1 CPU Card – USB connection module

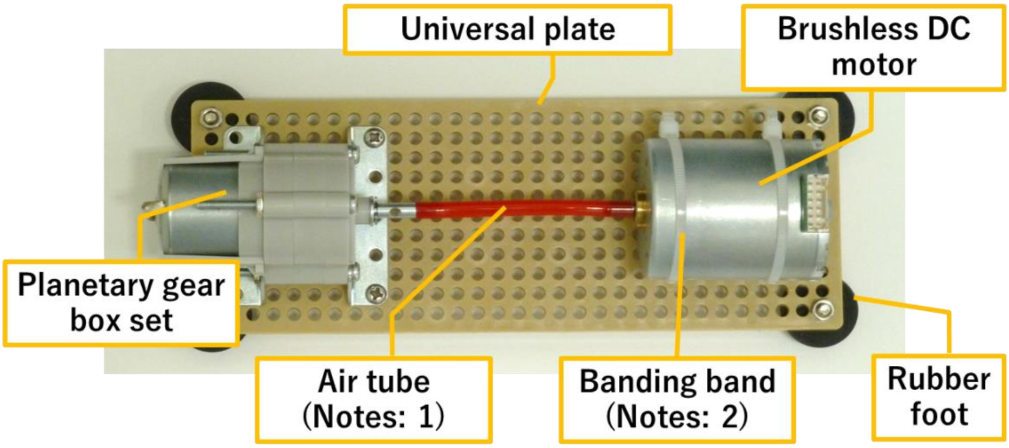
FT232RL communication module is required between CPU card (CN10) and USB port at HOST PC. Physical connection shown at Table 2-2 should be supported in design.

Table 2-2 USB-Serial converter cable pin connection list

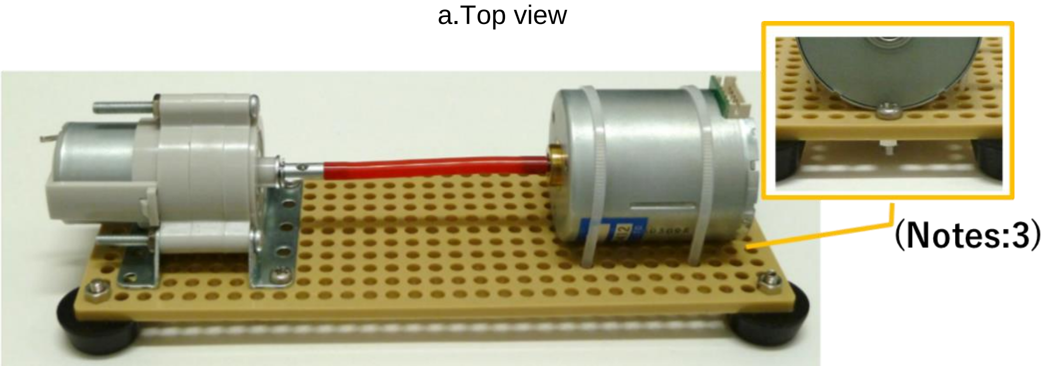
FT232RL module pin assign	CN10 assignment (CPU card)	
Function	**Pin No.**	**Function**
TX	3	RXD
RX	2	TXD
VCC	N.C	-
GND	4	GND

2.2 Simple motor bench assembly

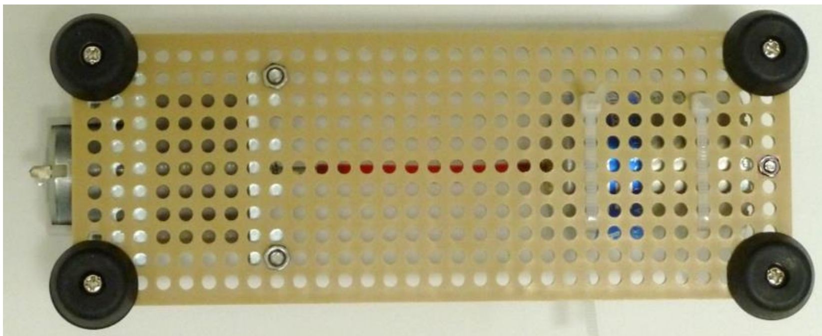
This section describes how to assemble a simple motor bench. The complete bench is shown in Figure 2-2.



a.Top view



b. Side view



c. Bottom view

Notes1. Connect the air tube before fixing the planetary gear box to the universal plate.
Make sure the tube is inserted securely into the base of the shaft.

Notes2. Tightly secure the brushless DC motor to the board with the bands.

Notes3. Screw as a stopper to fix the brushless DC motor.

Figure 2-2. Simple motor bench - Appearance

Figure 2-3 shows the position of the holes for the banding band to secure the load motor.

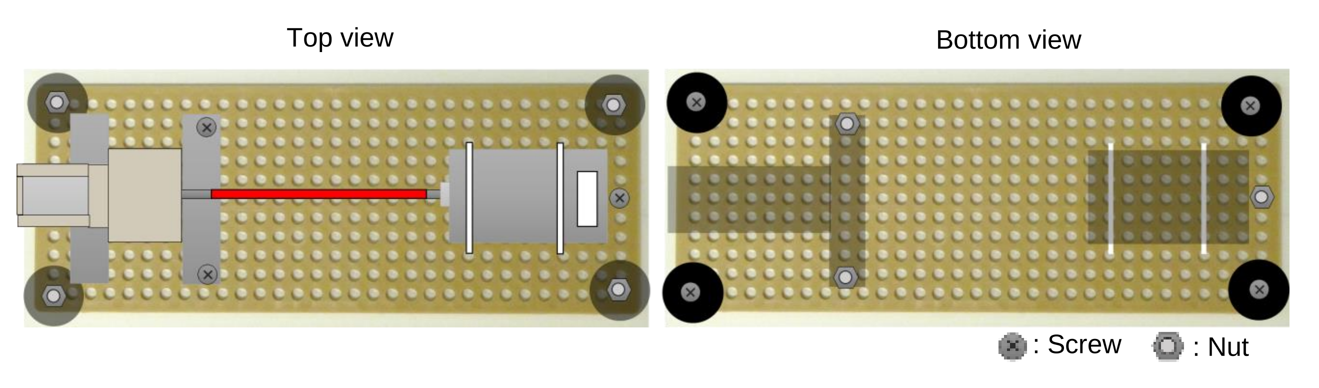


Figure 2-3. Simple motor bench - Parts location

3. Software and Tools

Table 3-1 lists Software and Tools required. For Failure Detection Demo Example, Project contains Motor application and AI application with TensorFlow Lite for Micro. Data Collection Tool and Training Tool will be used for AI model development.

Table 3-1. Software and Tools

Item	Description
Operating Environment	OS: Windows® 10
Integrated Development Environment (IDE)	e^2studio Version: 2020-10 (20.10.0) Build Id: R20200922-0919
Tool Chain	GCC ARM Embedded 8.3.1.20190703
FSP	2.0.0-alpha3+20200828.140790d2
Tools in demo package	Data Collection Tool Training Tool Demo Project: RA6T1 RSSK version 1.00

Demo Project can be imported into e2 Studio in following steps.

Select **File** and chose **Import**

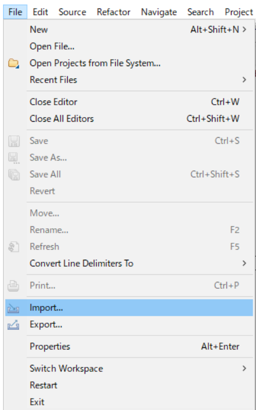


Figure 3-1. Project Import -1

Select **Projects frm Folder or Archive** then go **Next**

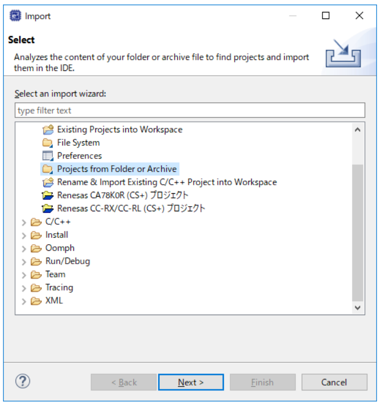


Figure 3-2. Project Import -2

Chose **Archive** then select archive to be imported then select **Finish**

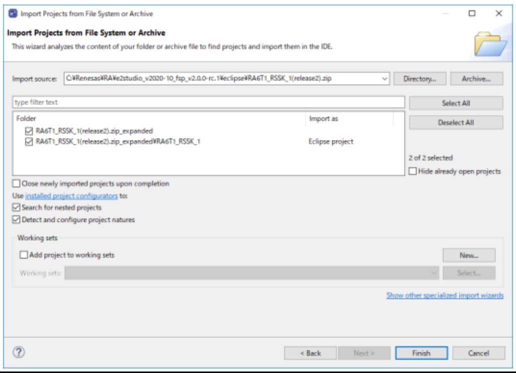


Figure 3-3 Project Import -3

Now project is imported to your environment and it creates directory structure such as Figure 3-4. Directory Structure Figure 3-4.

```
├──.settings
├──ra
├──ra_cfg
├──ra_gen
├──script
└──src
    ├──application
    ├──Comm_tool
    └──eAI_Inference
        └──tensorflow
            ├──tensorflow
            │   ├──core
            │   └──lite
            │       ├──c
            │       ├──core
            │       ├──kernels
            │       ├──micro
            │       │   ├──benchmarks
            │       │   ├──examples
            │       │   │   └──MotorExample
            │       │   │       └──ModelAndInputImage
            │       │   ├──kernels
            │       │   ├──kernels_nn
            │       │   ├──memory_planner
            │       │   └──testing
            │       └──schema
            └──third_party
                ├──flatbuffers
                ├──gemmlowp
                └──ruy
```

Figure 3-4. Directory Structure

4. Failure Detection Demonstration

4.1 Overview

Figure 4-1 shows the system operation flow. The following is an overview of operations.

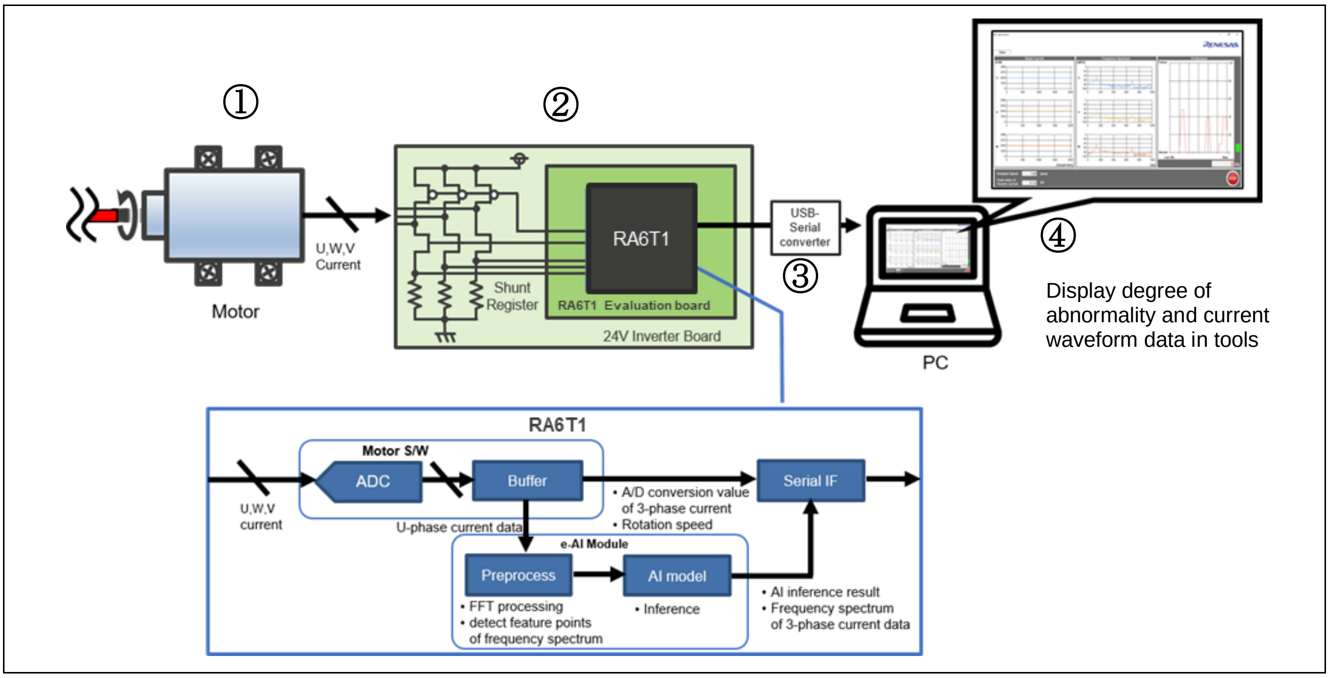


Figure 4-1. System Operation Flow

① Execute sensorless vector control on motor

When power is applied to the 24V inverter board, it is also applied to the RA6T1CPU board, which starts the motor driver operations. See Reference Documents[1] for details on board operations.

② Execute pre-processing for motor drive current data, determine anomaly using e-AI inference

1. A/D conversion value accumulation
   CMT1 generates the 2kHz sampling frequency and acquires the A/D conversion value of the motor 3 shunt current. U and W phase among of 3 shunt current are input to the 12-bit A/D converter. One frame (512 samples) of A/ D conversion values are accumulated for the FFT. From the next frame on, A/D conversion values are accumulated by overlapping 64 samples of the previous frame.
2. Data pre-processing
   The MCU performs the FFT operation using CMSIS DSP. The frequency spectrum resulting from the FFT operation is converted into dBFS. This sample software defines 0dB = 4095 LSB Full Scale. Next, the peak value of the frequency spectrum (excluding the DC component) and the previous and successive 8 samples (A/D conversion values) are selected to extract the frequency spectrum feature points.
3. AI inference
   The extracted feature points are input to the trained DNN, and the probability of the two classes

(normal and anomaly) are output by inference. In this example, the probability of anomality is taken as the degree of anomaly.

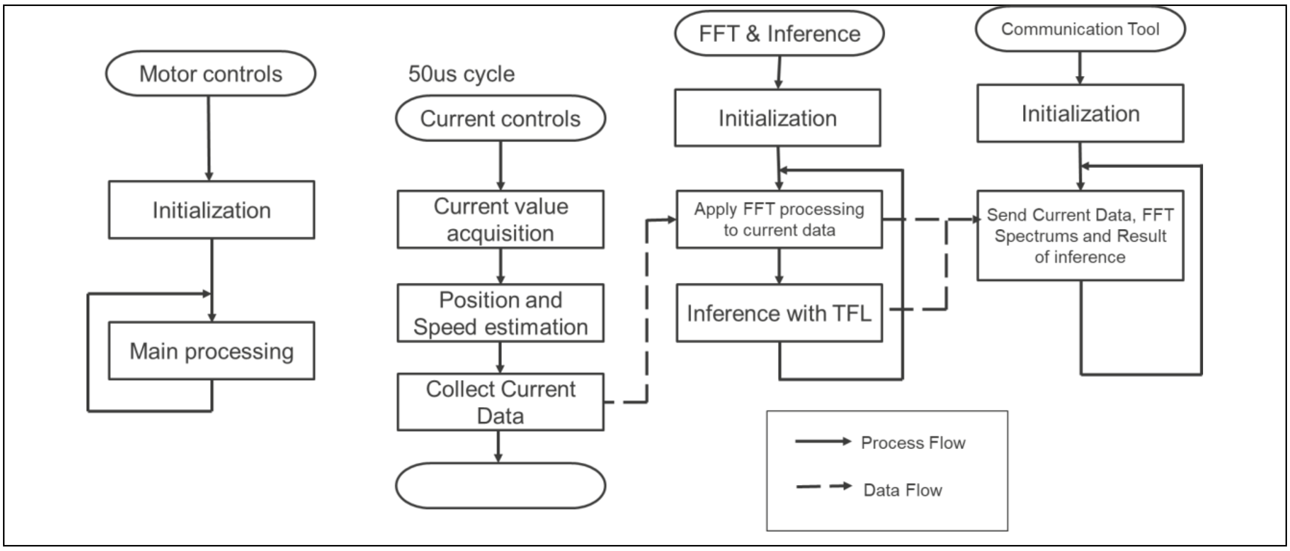


Figure 4-2. Demonstration flowchart

③ Serial communication with PC
Using CN10, data is transferred to the PC using a USB-serial converter cable.

④ Display degree of anomaly and current waveform data in tools
The received data is displayed in numerical values and graph form in the DataCollectionTool (GUI tool) on the PC.

Figure 4-3 shows images of the system in normal and anomaly states. Normal state is defined as when the drive motor and load motor shafts form a single line and anomaly state is defined as when the axis of the two shafts is deviated. In this example, normal and anomaly states are recreated using a simple motor bench, coupling the drive motor and load motor shafts with a tube.

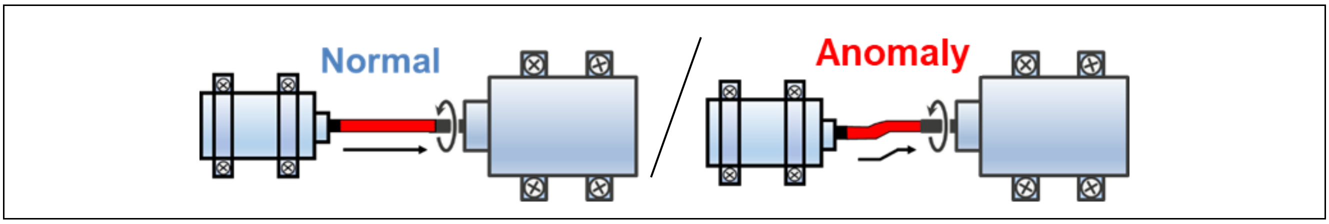


Figure 4-3. Images of Normal and Anomaly States

4.2 Preprocessing specifications

The RA6T1 Motor Failure Detection Application described in this document (referred to as “target system” below) preprocesses motor drive current data for use as AI input data. The following outlines the preprocessing used by the target system.

- Framing
  - Frames the A/D conversion value of motor drive current

- FFT
  - FFT is performed on the A/D conversion value of the motor drive current framed to detect the feature value.
- Data extraction
  - Extract data in the vicinity where the feature is detected.

The following describes preprocessing performed on the actual target system.

① shows the A/D conversion value of the motor drive current which is used as input data. This data must be framed every 512 points. To avoid missing data when collecting data, the frames are set so that 64 points overlap the previous frame. The waveform is output for 3-shunt current.

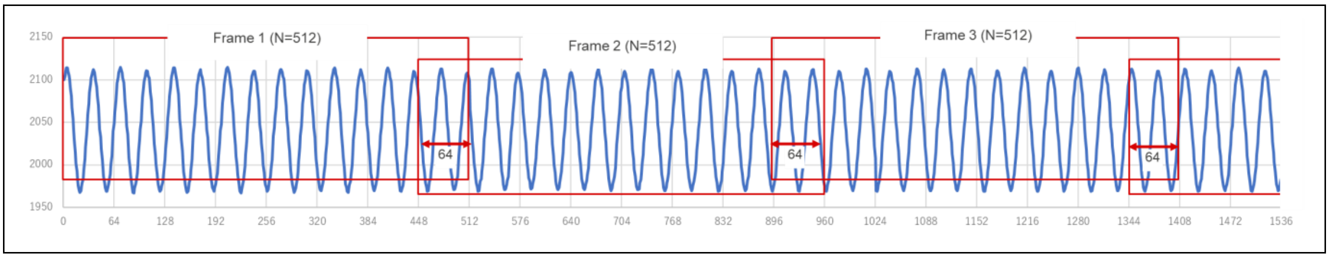


Figure 4-4. A/D Conversion Value of Motor Drive Current

② Feature values cannot be detected on the time axis, so the framed motor drive current A/D conversion value is FFT processed and converted to the frequency axis, as shown in Figure 4-5 graph (a). In the target system, features are detected around the peak value of the fundamental frequency outlined in yellow in Figure 4-5. Data Preprocessing Flow graph (b).

③ A total of 16 points before and after the peak value where the feature value was detected are extracted, as shown in Figure 4-5 graph (c), and used as input data. Only the U phase is used as data for the AI model.

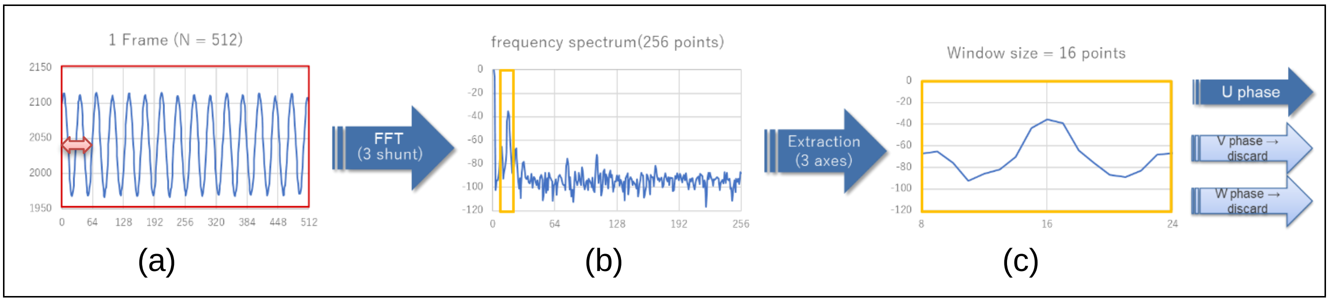


Figure 4-5. Data Preprocessing Flow

5. AI Model Development Flow

Figure 5-1 shows the flowchart of AI model development. This section describes the development sequence based on this flowchart.

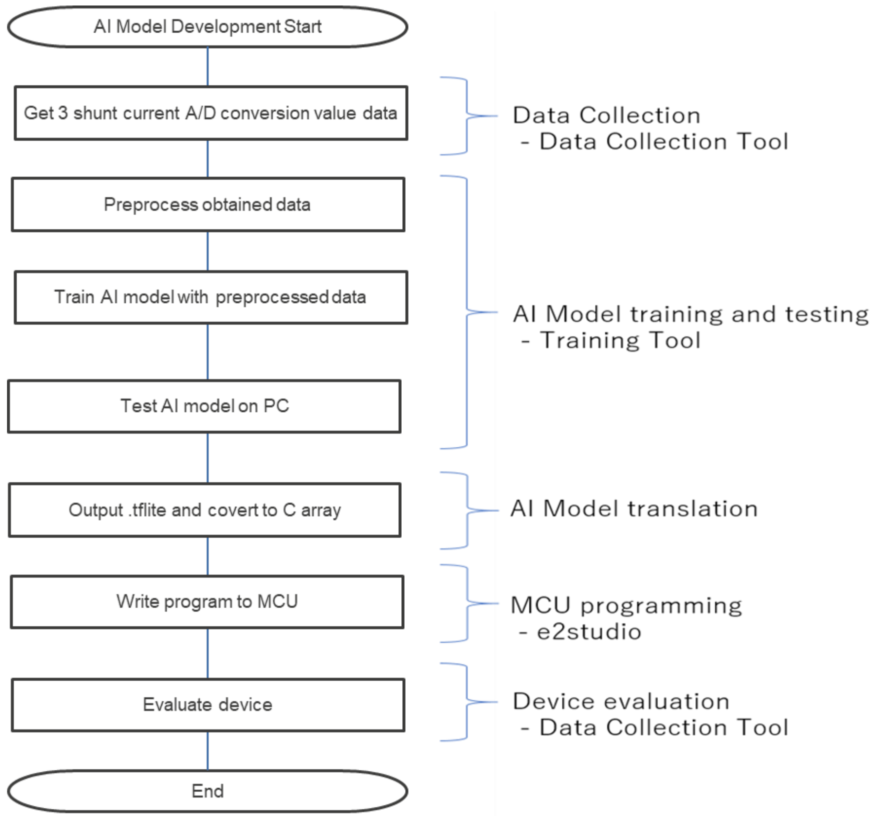


Figure 5-1. AI Model Development Flowchart

5.1 Data Collection

This section describes the sequence for collecting data using the Data Collection Tool. Two types of data are collected: data for training and data for testing. Testing data is used by the Training Tool when testing the AI model.

Data Collection Sequence

① Connect the PC and hardware using an FTDI-manufactured USB serial conversion cable and execute the DataCollectionTool_for_RX.exe file. For instructions on connecting the hardware, see Reference Document [1]. If you open the exe file before connecting the PC with a USB serial conversion cable, the error shown in Figure 5-2 will appear on the screen.

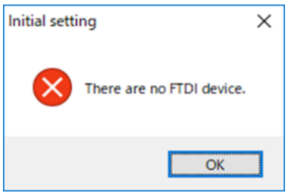


Figure 5-2. Error Dialog

② Change Communication Setting from 5000000(default) to 3000000

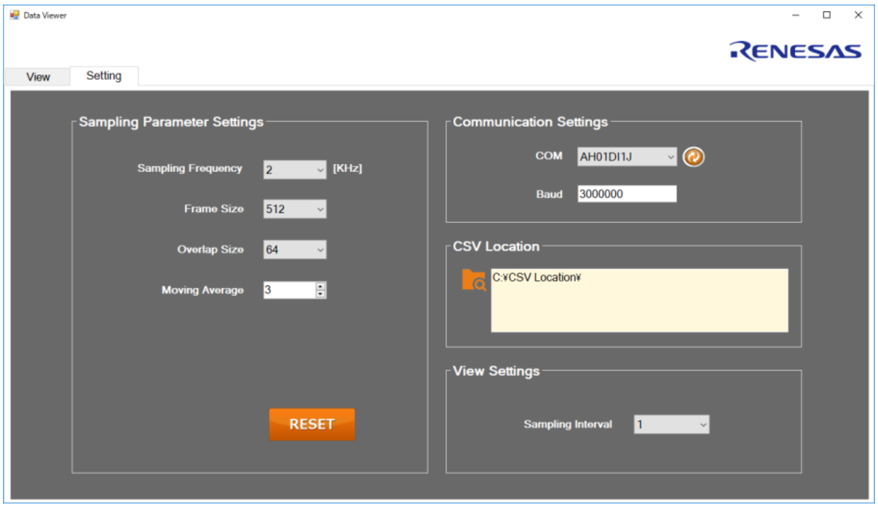


Figure 5-3. Screen Shot During Data Collection

③ From the dropdown list in the lower right corner of the window, select “Save to CSV (combined).” The Training Tool supports the CSV format output in this mode.

④ Press the START button in the lower right corner of the window to initiate data collection and display graphs. To stop the collection and display, press the STOP button found in the same place.

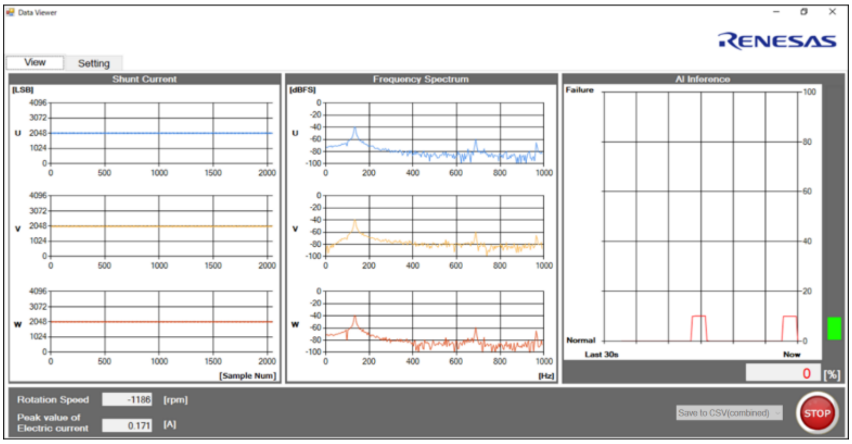


Figure 5-4. Screen Shot During Data Collection

⑤ Next, rename the file used for collected data. Change the names of the files using the naming convention Abnormal_No~.csv and Normal_No~.csv, as shown in Figure 5-5.

Figure 5-5. File Renaming Example

5.2 AI Model Training

This section describes the sequences for training and testing the AI model using the Training Tool. To re-train and test the AI model, use the training data and testing data previously collected by the Data Collection Tool.

AI Model Training Sequence

① Prepare three folders ahead of time and name as follows: Training data, Testing data, AI_Model. Store the collected training data and testing data in their respective folders. The AI_Model folder is for storing the output AI model.

② Execute the e-AI_Training_Tool.exe file. Make sure the program is set to Training mode.

③ Specify each data folder. Specify the folder that stores the training data as "Training Data Set." Specify the folder created for the output AI model as the "Output AI Model."

④ Click START to initiate the training. Preprocessing of the training data will start. When the progress bar shows "100%", preprocessing is complete and "dataframe.csv" is created in the folder where the collected data is stored. When preprocessing finishes, the sequence proceeds to the training process. When training finishes, "Training completed" is displayed as shown on the left in Figure 5-6. If a problem is detected during preprocessing, operations stop and "Training failed" is displayed.

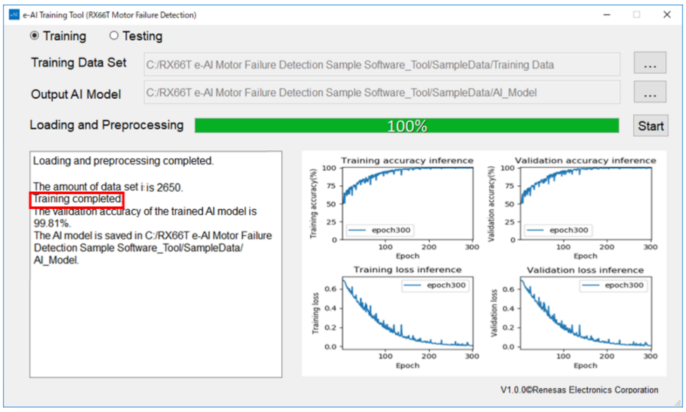


Figure 5-6. Training Completed Screen Shot

⑤ Confirm that the files shown in Figure 5-7 have been created in the AI_Model folder.

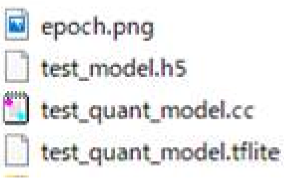


Figure 5-7. AI Model Storage Folder

⑥ Copy data of “test_quant_model_tflite” at “test_quant_model.cc” to “g_motor_model_data” array in the \src\eAI_Inference\tensorflow\tensorflow\lite\micro\examples\MotorExample\ModelAndInputImage \motor_model.cc and compile project to get RA6T1_RSSK_1.elf

⑦ Programming the RA6T1_RSSK_1.elf to the RA6T1 CPU board and execute it.

6. Demonstration

This demo example has confirmed by following data set. These are enclosed in project.

Normal data – 850, 900, 950, 1000 rpm

Anomaly data – 850, 900, 950, 1000 rpm

96% Accuracy is achieved in validation and 850 – 950 rpm was confirmed for normal / anormal detection in real hardware.

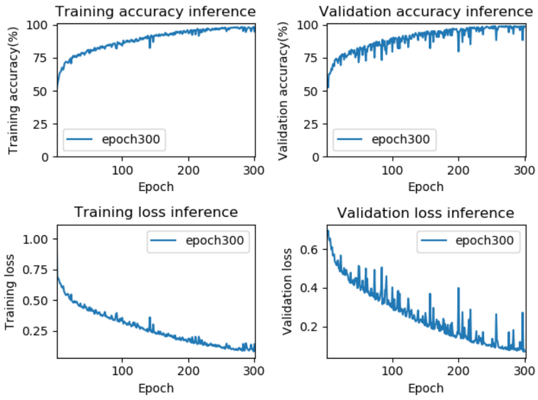


Figure 6-1. AI Model Storage Folder

7. Support Tools

7.1 Data Collection Tool

7.1.1 Overview

The Data Collection Tool is software that collects and displays 3 shunt current data and AI inference values from the MCU. The software comes in an EXE format executable file and does not require installation.

7.1.2 Functional Explanations

This software tool has a **View** tab for displaying all information and a **Setting** tab for setting up operations.

7.1.2.1 View Tab

Figure 7-1 shows the display layout used in the View tab. The numbers in the figure correspond to the numbered function descriptions below.

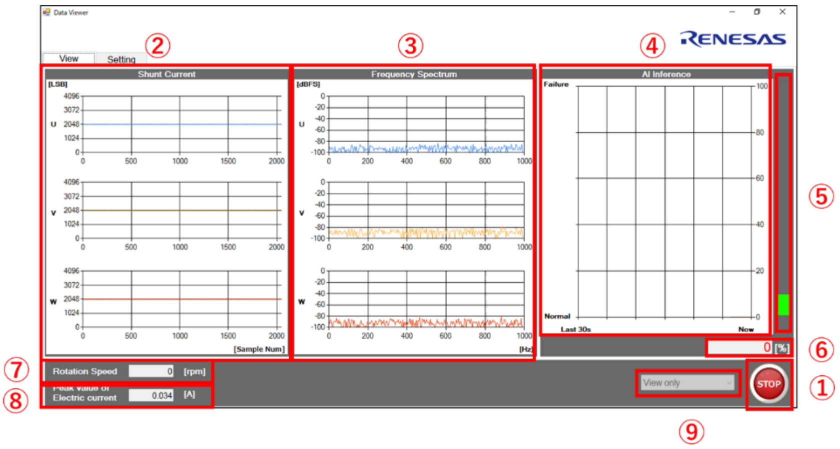


Figure 7-1. View Tab Layout

① Data acquisition START/STOP button

The START button is displayed when the GUI software is started up. Each function is described below.

- the START button is pushed:
  - Data Send Request Commands are sent from the PC to RA6T1, and data is sent from RA6T1 to the PC.
  - Received data is displayed in real time.
- the STOP button is pushed:
  - Data Send Stop Command is sent from the PC to RA6T1 and data acquisition ends.

② 3 shunt current A/D sampling results (magnitude waveform)
3 shunt current sampling data is plotted on a graph as U, V and W.

③ 3 shunt current FFT result (frequency characteristics)
The 3 shunt current waveform data in (2) above are transformed into the frequency spectrum via FFT, converted to dBFS and plotted on a graph.

④ Moving average waveform of AI inference result

The moving average of the abnormality probability output by AI inference is generated and plotted in a waveform graph.

⑤ AI inference result indicator bar

Displays the abnormality probability output by AI inference in a stacked bar graph in 10% increments.

⑥ AI inference result in percentages

Displays the abnormality probability output by AI inference in percentages.

⑦ Numerical value of rotation speed

Displays the motor rotation speed in numerical value.

⑧ Numerical value of peak current value

Displays the numerical value of the 3 shunt current's peak current value, which, in this example, is the U phase current's peak value.

⑨ Log function selection

User selects whether to output log (CSV file) from dropdown list. The CSV file is stored in the "CSV Location" folder immediately under the C drive in the initial settings.

- View only
- Only monitors various data.
- Save to CSV (divided)
- Monitors various data and outputs logs. This setting outputs the sampling waveform and frequency spectrum (dBFS) displayed in (2) and (3) to the file independently for each phase. Data is recorded after a line feed for every FFT frame.
- Save to CSV (combined)

- Monitors various data and outputs logs. This setting outputs the sampling waveforms and frequency spectrums (dBFS) displayed in (2) and (3) together in a single file. Data is added until the acquisition record is completed

7.1.2.2 Setting Tab

Figure 7-2 shows the display shows the display layout used in the Setting tab. The numbers in the figure correspond to the numbered function descriptions below.

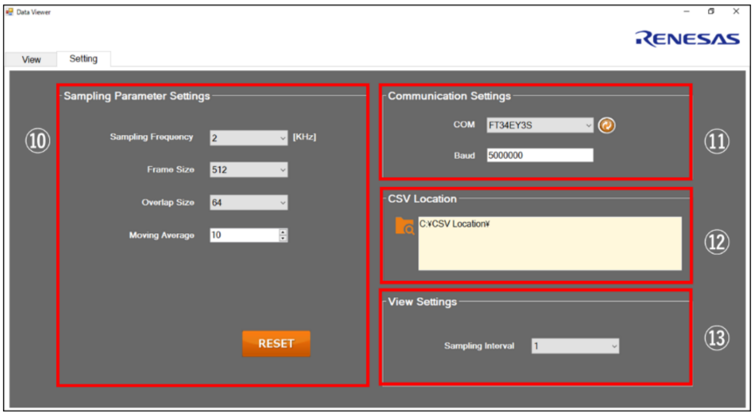


Figure 7-2. Setting Tab Layout

① **Sampling parameter setting**

The trained DNN in this example is optimized to the default setting except for the moving average.

- Sampling Frequency
  Specifies the sampling frequency (1/2/4/8 kHz, default: 2 kHz).
- Frame Size
  Specifies the FFT frame size (128/256/512/1024, default: 512).
- Overlap Size
  Specifies the FFT frame overlap size (16/32/64/128, default: 64).
- Moving Average
  Specifies the moving average of the graph for the AI inference result (specified range: 1 to 100 times, default: 10).

② Communication setting

- COM
  Displays the name of the FTDI device connected to the PC.
- Baud
  Specifies the Baud rate for communications between the MCU and PC (range: 9600 to 5000000, default: 5000000)

③ CSV storage location setting

Specifies the CSV file output location when the View tab is set to output logs.

④ View settings

Specifies the update interval of the view tab (1/2/4/8/16/32/64, default: 1)

7.2 Training Tool

The Training Tool is software that trains and tests the AI model. The software comes in an EXE format executable file and does not require installation. It is bundled with trained DNN and can be retrained. The following is an overview of the Training Tool operations.

・AI model training
・AI model testing
・Testing preprocessing

7.2.1 Function Descriptions

This software has two modes: Training mode, which trains the AI model, and Testing mode, which tests the trained AI model.

7.2.2 System Requirements

This training tool needs following software installed in PC.

Python	3.5.3
Keras	2.3.1
Numpy	1.16.3
Pandas	0.25.3
TensorFlow	2.1.0

7.2.2.1 Training mode

Figure 7-3 shows the screen layout when in training mode. The numbers in the figure correspond to the numbered function descriptions below.

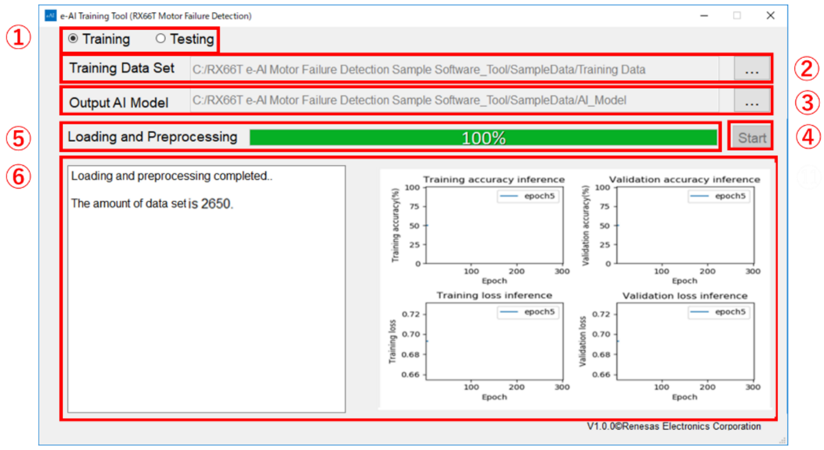


Figure 7-3. Screen Layout in Training Mode

① Mode selection
Select between Training and Testing mode.

② Training Data Set folder setting
Select the folder that stores the training data.

③ Output AI Model folder setting
Select the folder for the Output AI Model.

④Start button
Starts to pre-process the training data and then trains AI model.

⑤ Loading and Preprocessing
Displays the status of the training data preprocessing.

⑥ Training Status display
Displays the status of the training. The accuracy and loss of the AI model are plotted in the graph.

7.2.2.2 Testing Mode

Figure 7-4 shows the screen layout when in Testing mode. The numbers in the figure correspond to the numbered function descriptions below.

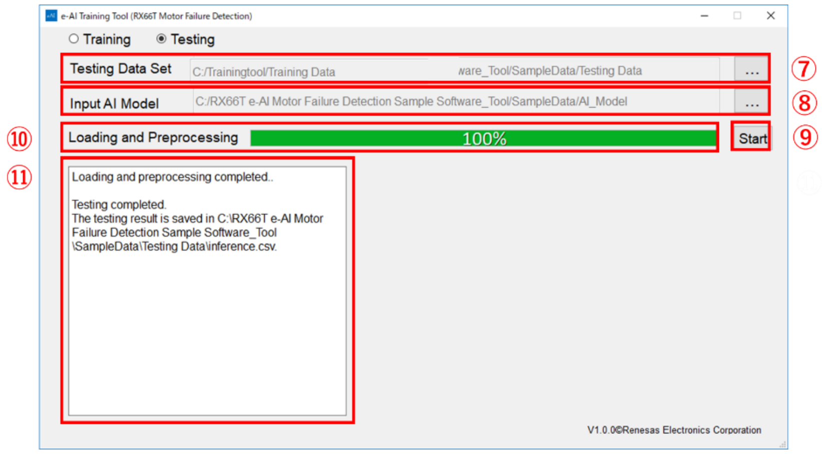


Figure 7-4. Screen Layout in Testing Mode

⑦ Testing Data Set folder setting
Select the folder that stores the testing data.

⑧ Set Input AI Model folder setting
Select the folder that stores the AI model to be tested

⑨ Start button
Starts to pre-process the testing data and then tests the trained AI model.

⑩ Loading and Preprocessing
Displays the status of the testing data preprocessing.

⑪ Testing Status display
Displays the Testing status.

8. Reference Documents

[1] Motor Control Evaluation System for RA Family User's Manual (R12UZ0078EJ0100 Rev 1.00)
[2] e2 studio Integrated Development Environment User's Manual: Getting Started Guide (R20UT4374)

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Oct.26.20	---	Initial Release

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)
   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on
   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state
   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins
   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals
   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin
   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses
   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products
   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/.

Trademarks