

RX62T

R01AN0903EU0201

Rev. 2.01

Jul. 30, 2014

Three-Shunt Sensorless Vector Control of Permanent Magnet Synchronous Motors (PMSM)

Introduction

This document presents the RX62T three-shunt sensorless vector control solution, which has been implemented on a RX62T evaluation kit with a three-shunt current measurement.

It describes the evaluation kit hardware platform, three-shunt current measurement, motor phase current reconstruction method, three-shunt sensorless vector control strategy, and software implementation. The sensorless vector control algorithm is based on the method introduced in Renesas Application Note REU05B0103-0100/Rev.1.00.

Target Device

RX62T

Contents

1. Overview	2
2. Specifications and Performance Data.....	3
3. Hardware Platform	4
4. System Control Block Diagram	5
5. Three-Shunt Motor Phase Current Measurement.....	6
6. Three-Shunt Sensorless Vector Control Strategy.....	12
7. Software Descriptions	13
8. Motor and Control Parameter Tuning Example	16
9. Demonstration Guide	18
Appendix A - References	30

1. Overview

Recent years have seen rapid development and wide applications of microcontrollers in embedded control system such as motor control, and 32-bit floating-point units (FPU) have also been adopted in the high-end motor control applications. The floating point architecture improves performance, simplifies programming, and enables more sophisticated motor control algorithms for advanced capabilities.

The Renesas RX MCU is a high-performance and low cost 32-bit 100MHz/165DMIPS microcontroller. It supports the IEEE-754 standard format for single-precision 32-bit floating-point data. The hardware FPU integrated in the RX core offloads the RX CPU and simplifies decimal-point calculations, boosting overall processing performance and simplifying firmware development.

For the developers of the RX62T based motor control applications, the RX62T evaluation kit provides a complete development platform with all of the hardware and software required to get applications started and shorten time to production by allowing developers to apply what they learn in the evaluation phase to the real-applications.

The kit comes ready to run a Brushless DC (BLDC) motor included. Simply power up the board to explore the 5 different types of motor control solutions: 1) three-shunt sensorless vector control; 2) single-shunt sensorless vector control; 3) three-shunt sensorless vector control with the internal Programmable Gain Amplifier (PGA); 4) single-shunt sensorless vector control with the internal PGA; 5) encoder based position control. Developers can modify the demonstration applications to develop specific motor control applications, and using the Graphical User Interface (GUI), the motor and control parameters can be modified to drive your own motor.

This document presents the RX62T three-shunt sensorless vector control solution, which has been implemented on the RX62T evaluation kit with a three-shunt current measurement. This document describes the evaluation kit hardware platform, three-shunt current measurement, motor phase current reconstruction method, three-shunt sensorless vector control strategy, and software implementation. The sensorless vector control algorithm is based on the method introduced in Renesas Application Note REU05B0103-0100/Rev.1.00. Specifically, this document explains how to modify the solution for different motors and applications, and this document gives an example of parameter tuning.

The software described in this application note is applicable to the following devices and platforms:

- ❖ MCU: RX62T and RX62N
- ❖ Motor: Three-Phase BLDC Motor and PMSM
- ❖ Platform: Renesas Evaluation Kit
- ❖ Control algorithm: Three-Shunt Sensorless Vector Control

2. Specifications and Performance Data

The implementation of the three-shunt sensorless vector control is based on the Renesas evaluation kit and the RX62T MCU. The main specifications are described as following:

- ❖ Input Voltage: 24VDC
- ❖ Rated Bus Voltage: 24V
- ❖ Output Voltage: 24VAC
- ❖ Rated Output Power: 120W
- ❖ PWM Switch Frequency: 20KHz
- ❖ Control Loop Frequency: 10KHz
- ❖ Current Measurement: 3 Shunt Resistors
- ❖ Implementation: FPU
- ❖ CPU Bandwidth: 19.7%
- ❖ Used Flash Memory: 33.816Kbytes
- ❖ Used RAM: 4.387Kbytes
- ❖ Used Stack : 336bytes

3. Hardware Platform

The RX62T evaluation board is a single board with an integrated power inverter with the controller. The hardware includes a low-voltage MOSFET power stage, a communication stage, and a RX62T microcontroller based controller as shown in Figure 1.

The board has the following features:

- ❖ A complete 3-phase inverter on-board with a low-voltage motor
- ❖ 24V external power supply to provide DC bus voltage, 15V and 5V power supply
- ❖ Power devices use Renesas low-voltage MOSFETs
- ❖ Power rate up to 120 watts
- ❖ Supports three-shunt and single-shunt current measurements
- ❖ Easily change jumpers from the external amplifiers to the internal PGA
- ❖ USB communication with the PC via a H8S2212 MCU
- ❖ GUI to both modify the motor and control parameters and tune the speed and position control
- ❖ Connectors for Hall sensors and encoder connections
- ❖ LCD to monitor the operation status
- ❖ Supports the standalone mode set by potentiometer and push-buttons
- ❖ Supports the second motor drive, signals and connector for another motor control power stage

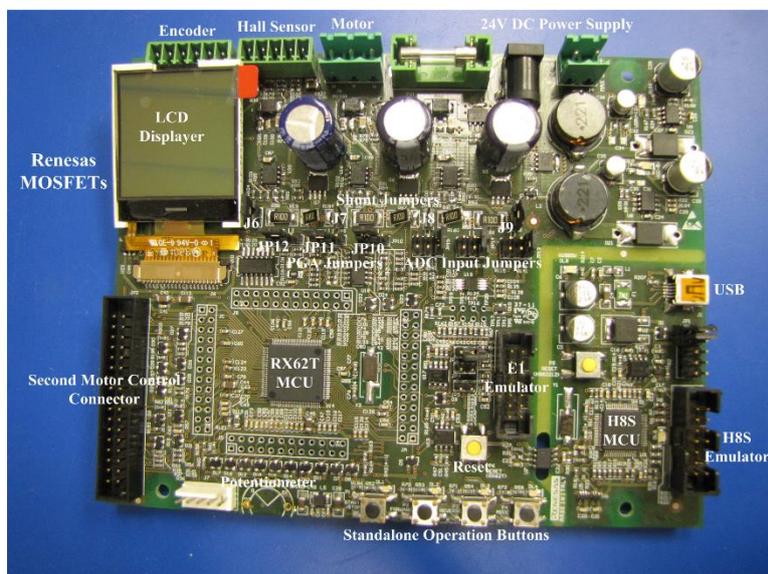


Figure 1 Evaluation Board

4. System Control Block Diagram

The RX62T is a 32-bit high-performance microcontroller with a maximum operating frequency of 100MHz and 165 DMIPS and single precision floating-point unit (FPU), which is equipped with multifunction timers (MTU, GPT), high-speed 12-bit ADC and a 10-bit ADC for facilitating motor control. Figure 2 depicts the block diagram of a sensorless vector control of a PMSM based on the Renesas RX62T Microcontroller.

The evaluation kit is capable of controlling two motors. The RX62T timer MTU3 channels 3 and 4 can be used to drive the second motor, which is not covered in this application note. The MTU3 channels 6 and 7 are used to generate 6 PWM signals to drive the on-board motor in the complementary mode. The PWM modulation uses the space vector PWM or the sinusoidal PWM with the third harmonic. The three-phase inverter uses the Renesas low-voltage MOSFETs, which generates three-phase voltages with variable frequency and amplitude to drive the motor to the desired voltage.

The motor currents of i_u, i_v, i_w are measured by three shunt resistors. The currents of i_u, i_v, i_w are measured by a 12-bit ADC unit0 of channel AN002, AN001 and AN000, respectively, and the bus voltage is measured by a 12-bit ADC unit0 of channel AN003.

The second motor currents of i_u, i_v, i_w are measured by a 12-bit ADC unit1 of channel AN102, AN101, and AN100. The bus voltage is sampled by a 12-bit ADC unit1 of channel AN103.

The speed is given by an external potentiometer, which is input to the 10-bit ADC channel AN1. The MOSFET temperature is measured by the 10-bit ADC channel AN2.

The encoder pulses A and B are input to the TCLKA and TCLKB, and the Z pulse is input to IRQ0. For the second motor, the encoder pulses A and B are input to the TCLKC and TCLKD, and the Z pulse is input to IRQ3. The MTU3 timer has phase counting mode to capture two-phase encoder pulse inputs.

When the motor and the power board have overcurrent, the current ADC sample circuits generate a port output enable (POE) signal to immediately shut down the PWM outputs from the MCU hardware. In addition, the RX62T control system has various system level safety features to comply with the IEC60730 safety standard such as: low-voltage detection (LVD), independent watchdog timer (IWDT), clock stop detection, ADC converter self-diagnosis, and an output port monitor, etc.

The GUI communicates with the RX62T MCU using the USB, and it can display the motor operation status in the real time and tune the motor and control parameters. The board can also run in the standalone mode, and the LCD displays the motor status.

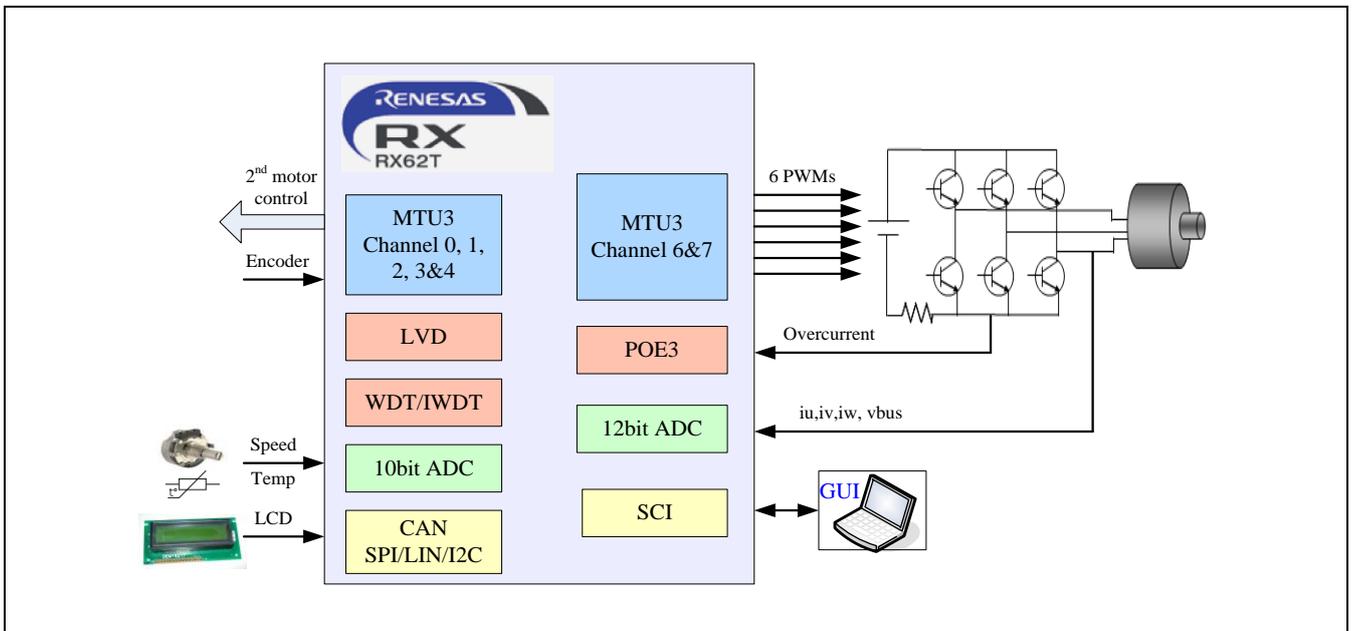


Figure 2 System Control Block Diagram

5. Three-Shunt Motor Phase Current Measurement

5.1 Three-Shunt Current Measurement Method

The three-phase motor currents are usually sampled by:

- ❖ Shunt Resistors
- ❖ LEM Hall Effect Sensors

Shunt resistors are popular current sensors because they provide an accurate measurement at a low cost. LEM Hall Effect current sensors are widely used, because they provide a non-intrusive measurement and are available in a small IC package such as the Allegro Current Sensor IC that combines the sensor and signal-conditioning circuit. Table 1 summarizes the advantages and disadvantages between the two current sensors.

Table 1 Current Sensing Comparison Between Hall Effect Sensor and Shunt Resistor

Current Sensing Method	LEM Hall Effect Sensor	Shunt Resistors
Accuracy	good	good
Cost	high	low
Isolation	yes	no
AC/DC Current	both	both
High-Current	good	poor
DC Offset	no	yes
Intrusive	no	yes
Power Consumption	low	high

Shunt resistors are a popular current-sensing sensor, because of the low cost and good accuracy. The voltage drop across a known low value resistor is measured in order to determine the current flowing through the load. If the resistor is small in magnitude, the voltage drop is small and the measurement doesn't have a major effect on the motor circuit.

The selection criteria of a shunt current resistor requires the evaluation of several trade-offs, including:

- ❖ Increasing the resistance to increase the sensing voltage, which makes the voltage offset and input bias current offset amplifier errors less significant.
- ❖ A large resistance value causes a voltage loss and a reduction in the power efficiency, and also causes a voltage offset to the load in a low-side measurement that may impact the EMI characteristics and noise sensitivity of the system.
- ❖ Special-purpose, low inductance resistors are required if the current has a high-frequency content.
- ❖ The power rating of resistor must be evaluated, because the I^2R power dissipation can produce self-heating and a change in the nominal resistance of the shunt.

Low-side current measurements offer the advantages that the circuitry can be implemented with a low-voltage op amp, because the measurement is referenced to ground. The low-side measurement circuit can use a non-inverting amplifier, and the input impedance of the circuit is equal to the large input impedance of the amplifier.

On the other hand, there are some disadvantages to the low-side resistor. The low-side resistor disrupts the ground path, and the added resistance to the grounding system produces an offset voltage which can cause EMI noise problems. Also, it can't detect a fault where the load is accidentally connected to ground via an alternative ground path.

5.2 Three-Shunt Current Measurement Circuit

Figure 3 shows the evaluation kit hardware circuit for the three-shunt current measurement. Jumpers J6 and J9 are shorted, while jumpers J7 and J8 are open. The composite current of all three MOSFET inverter low-side legs can be measured with a single shunt resistor of 0.1Ω, or the current in each individual leg can be determined with three shunt resistors. Table 2 lists the jumper settings for the three-shunt current measurement.

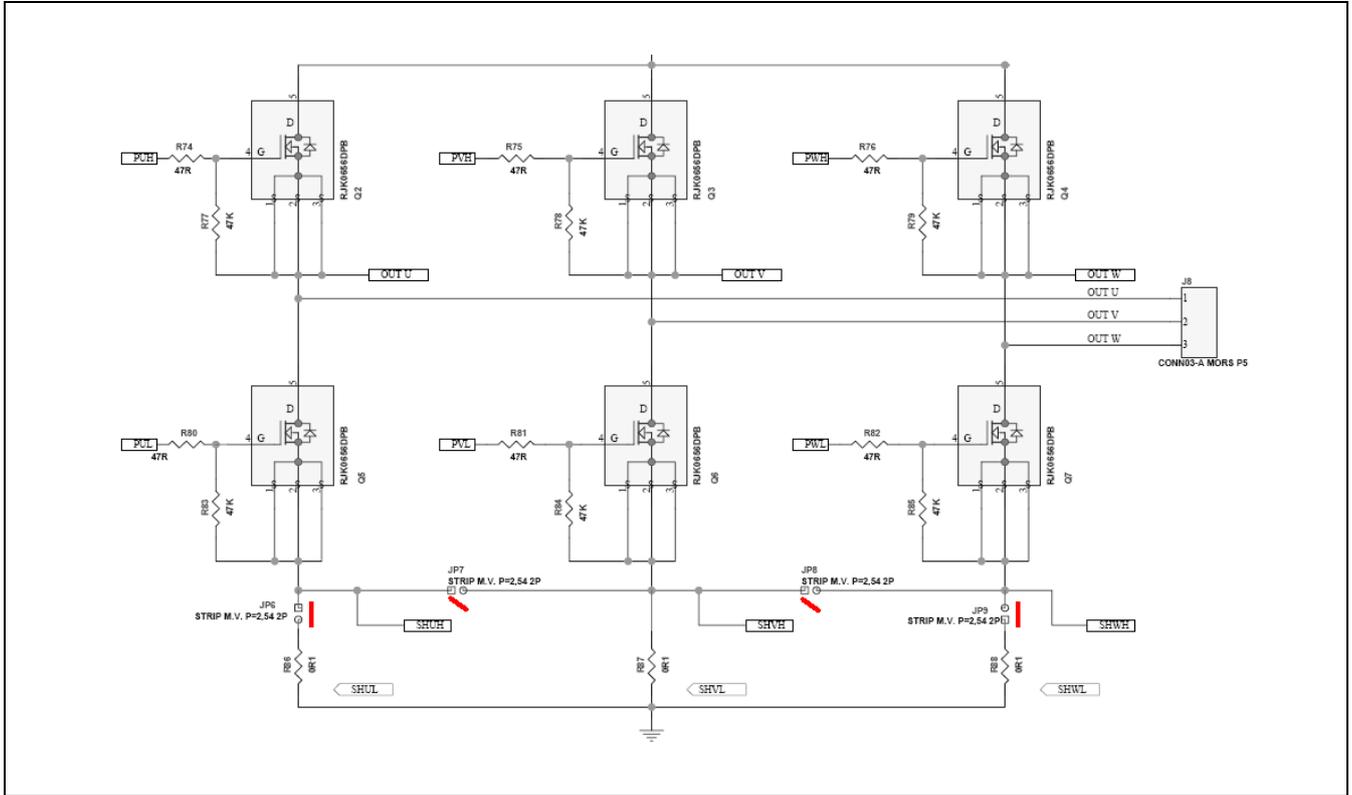


Figure 3 Three-Shunt Current Measurement in the Low-Side Inverter Legs

Table 2 Jumper Settings for the Three-Shunt Current Measurement

Jumper	J6	J7	J8	J9
State	ON	OFF	OFF	ON

5.3 Motor Current Reading from Three Shunts

Figure 4 depicts one leg of the inverter and voltage waveforms. If the low switch is ON and the gate command u^- is high, then:

- ❖ The gate command u^+ is low
- ❖ No current can flow through the high switch, because it is OFF.
- ❖ No current flows through the high diode because it is inversely polarized, so the high diode is also OFF.

If the low-side switch is ON, the phase current flows through the shunt. Therefore, in order to read the motor phase current, the corresponding shunt current should be read when the related low-side switch is ON.

By observing the voltage waveforms obtained with three-phase modulation in Figure 4, it can be seen that two of the three phases are lower than the 90% of the maximum voltage value, so two of the three phases have a low switch ON period greater than 10% of the total period. Therefore, it is always possible to read two currents with a three-shunt measurement system.

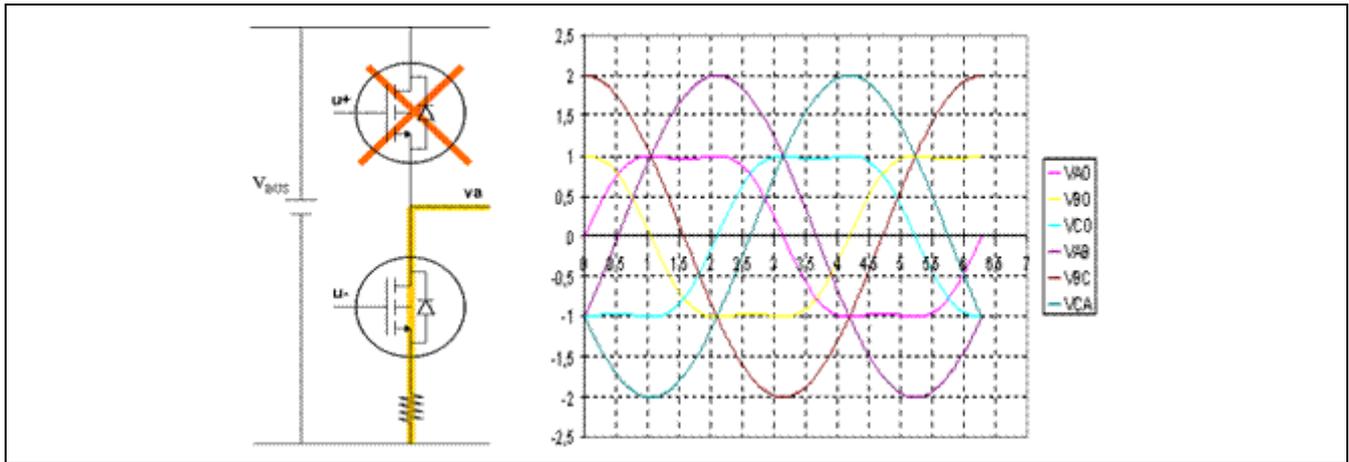


Figure 4 One Leg of the Inverter (Left) and Motor Voltage Waveforms (Right)

Figure 5 shows the PWM waveforms. The three-shunt current reading depends on the PWM duty cycle. The lesser is the duty, the higher voltage is the output, so the lesser time the lower switch is on. Those trees are chosen for which the lower arm on time is higher. Therefore, the conversion in amperes should be made.

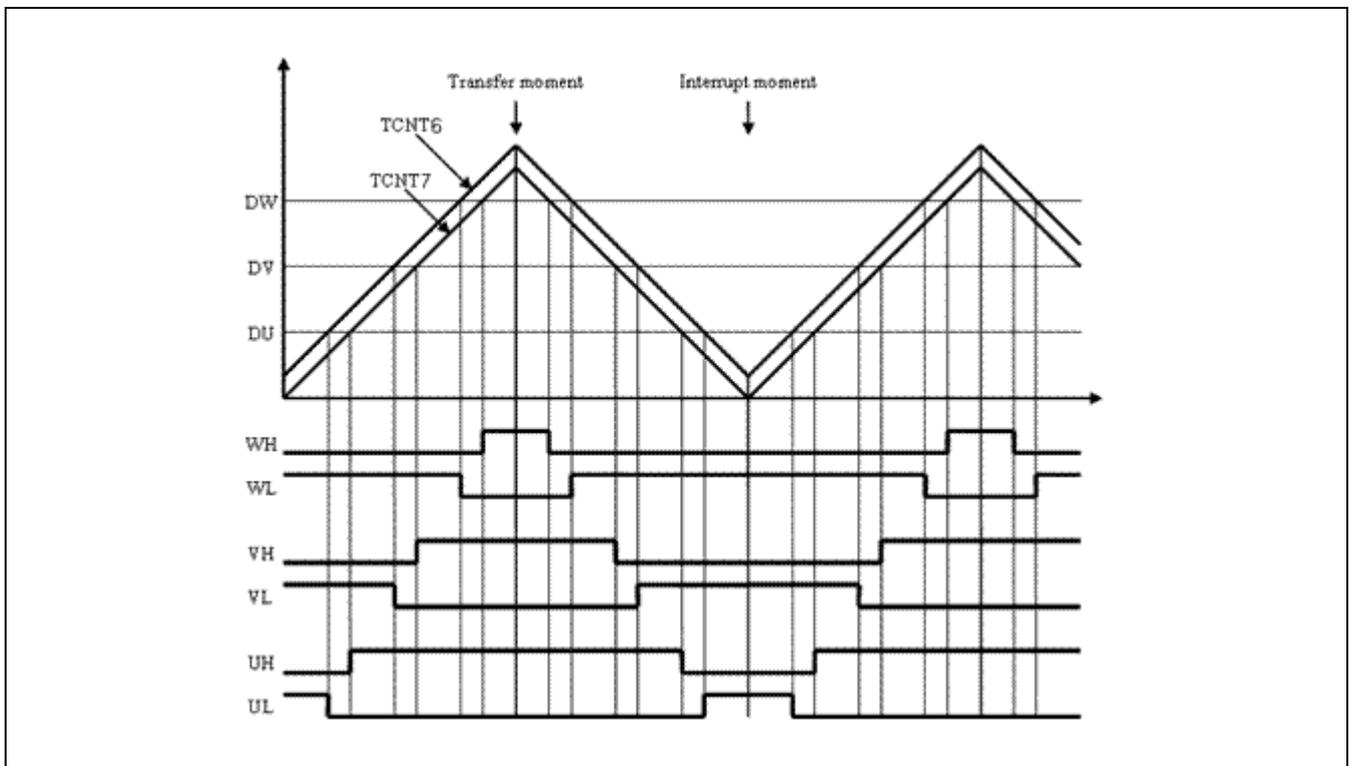


Figure 5 Three-Phase PWM Duties and PWM Waveform

5.4 Three-Shunt Current ADC Sampling and Scaling

Figure 6 shows the three-shunt current sample and amplifier circuit. Due to the current polarity, the offset should be added to shift the maximum negative current to 0 volts. After the offset and amplifier, the three-shunt currents input to the RX62T MCU ADC channels AN000, AN001 and AN002 and are listed in Table 3.

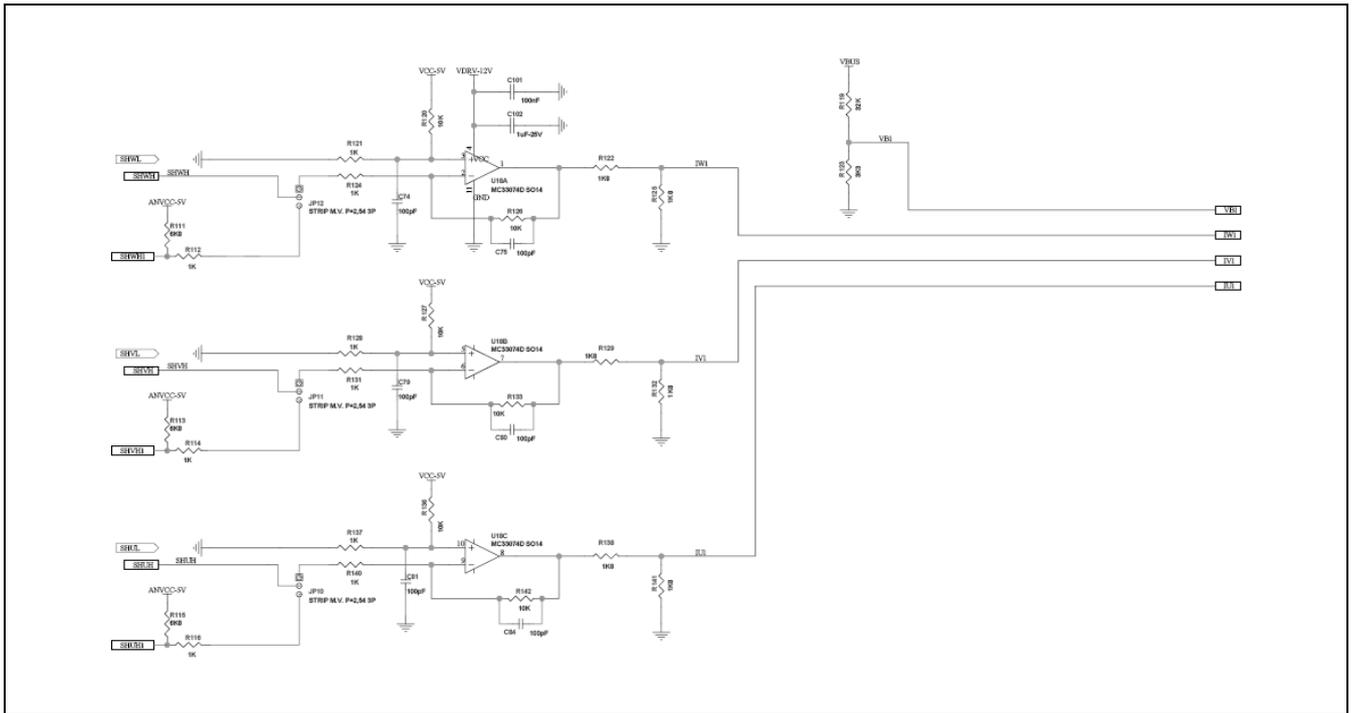


Figure 6 Motor Current Measurement Circuit

Table 3 ADC Conversions

Item	ADC Channel	Conversion Ratio (actual value/ADC input value)
Phase u Current - i_u	AN002	-5A-5A/ 0-5V
Phase v Current - i_v	AN001	-5A-5A/ 0-5V
Phase w Current - i_w	AN000	-5A-5A/ 0-5V
DC Bus Voltage - v_{bus}	AN003	0-50V/0-5V

The signals can be categorized into two main types: bipolar and unipolar. The motor phase currents are examples of bipolar signals, while the DC bus voltage is an example of a unipolar signal.

The shunt resistor used in this application is 0.1Ω, and the measured current range is from -5A to 5A. The gain of the amplifier is set to 5. Figure 7 depicts the current sensing scale translation.

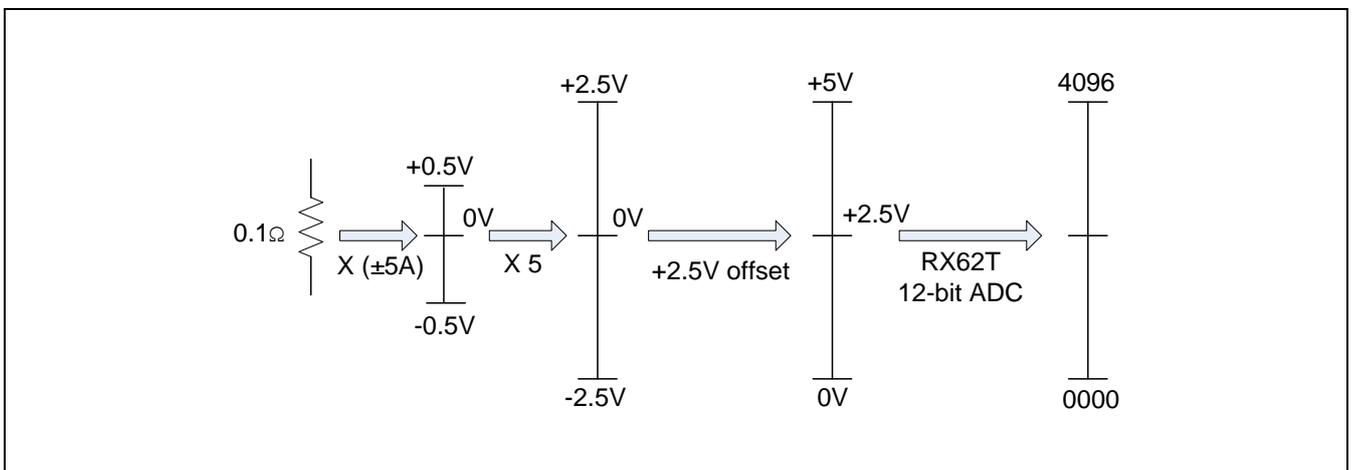


Figure 7 Motor Current Sensing Scale Translation

The shunt resistor senses the maximum ± 5 Amps current to translate the current to the ± 0.5 volts voltage, and the amplifier enlarges this voltage 5 times. The ADC inputs accept the analog input signals (i_u , i_v , and i_w) in the range of 0-5 volts for the RX62T with the ground referenced to 0 volts. The output voltage of the amplifier is added to the 2.5V offset. The ADC input value of 2.5V is taken as the current of zero. The ADC input values from 2.5V to 5V correspond to positive current values, and the inputs from 0 to 2.5V correspond to negative current values. The 12-bit ADC reads the voltage into the digital value from 0 to 4096.

$$i_u = KADI * (AN002 - i_{u_offset})$$

$$i_v = KADI * (AN001 - i_{v_offset})$$

$$i_w = KADI * (AN000 - i_{w_offset})$$

$$KADI = 5 / (4096 * Rshunt * Kamp)$$

Where,

i_u , i_v , and i_w are three-phase motor currents;

i_{u_offset} , i_{v_offset} , and i_{w_offset} are u , v and w phase current offsets;

$AN002$, $AN001$, and $AN000$ are the 12-bit ADC reading values of u , v and w phase shunt currents;

$KADI$ is the motor phase current scaling;

$Rshunt$ is the value of the shunt resistor;

$Kamp$ is the gain of the current amplifier.

5.5 Three-Shunt Current Reading Software Implementation

The software implementation includes the shunt current offset calculation `MC_SetOff()` and the shunt current reading `MC_Readc_TS()`.

- ❖ `MC_SetOff()` reads the offset of the ADC channels when the currents are 0, and it is executed once at the beginning.
- ❖ `MC_Readc_TS()` is called every interrupt at the trough, and it reads the ADC conversion results. Given the duty-cycle values, it calculates which two channels are to be read corresponding to the longer ON period of the lower switches. It has to subtract the channel offsets and converts them into the motor actual currents. The 3rd current is calculated as minus the sum of the other two.

Figure 8 shows the flowchart of the three-shunt current measurements.

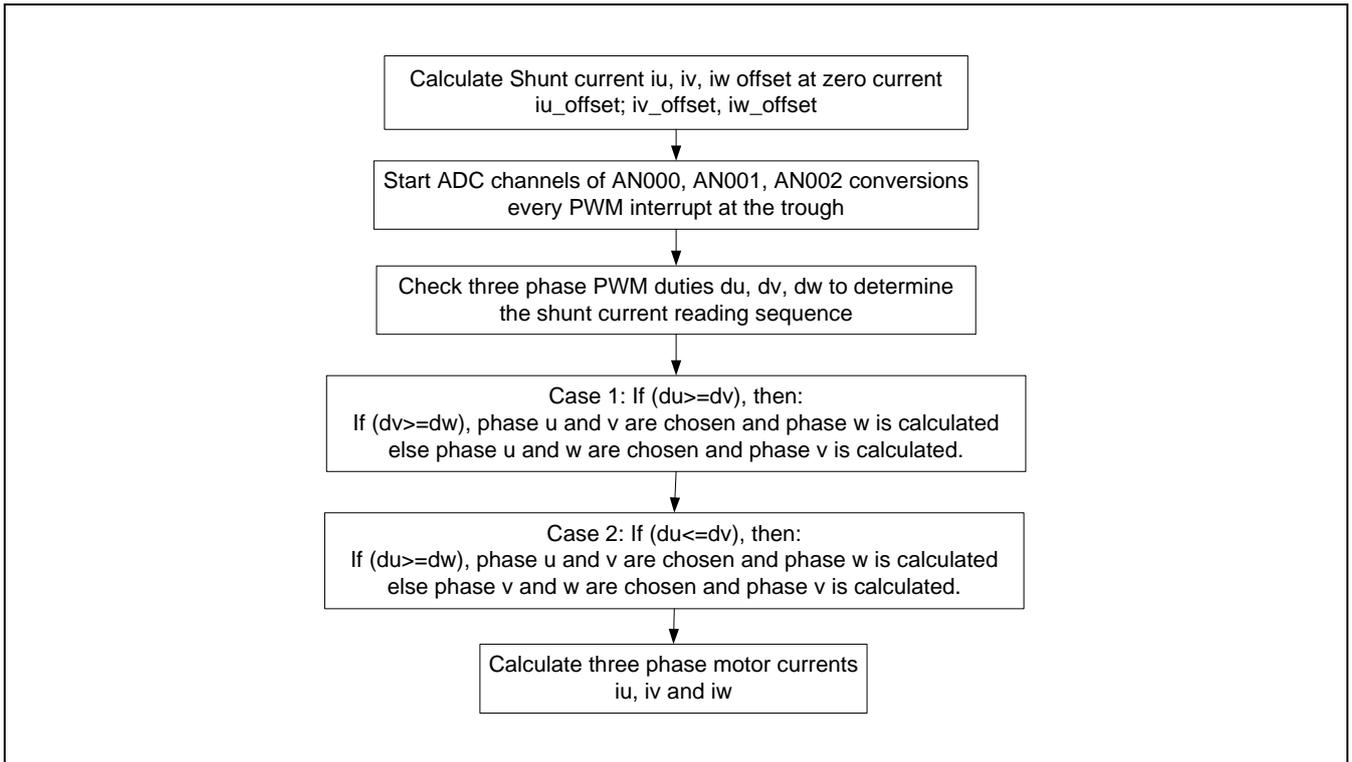


Figure 8 Flowchart of Three-Shunt Current Measurements

6. Three-Shunt Sensorless Vector Control Strategy

The vector control formulation depicted in Figure 9 provides a straightforward way to control the flux and the torque in a similar way to controlling DC motors – the flux is controlled by the I_d current, while the torque is controlled by the I_q current. Due to its nature, the 3-phase motor has three windings and three currents which are 120 degrees apart. Vector formulation uses Clarke and Park transforms to convert the measured phase currents from the u, v, and w frame to first transform them into the static orthogonal (α, β) frame (which is 90 degrees apart) and then the rotor frame. The rotor frame is also an orthogonal frame aligned along the magnetic field axes known as the d and q frame. These transformations use the transcendental functions sine and cosine of the rotor angle; therefore, it is a requirement that the rotor angle is known at the time the calculation is made. Once the currents are transformed into the d and q frame, the control algorithm simply runs the PID or PI loop to calculate the required voltages for the torque and flux. These required voltages (V_{dc} , V_{qc}) are then transformed back into the u, v, and w frame using the inverse Clarke and inverse Park transforms to further calculate the PWM duty cycle.

All these blocks form the inner current loop shown in Figure 9 with two computing blocks known as flux observer and speed estimation, which are special parts of the sensorless vector control formulation. When a sensor is used to measure the rotor angle and speed measurements, these two blocks change significantly.

In sensorless implementation, because there is NO sensor to measure the angle and speed, a motor model is used to calculate the flux and estimate the speed based on the measured phase currents and motor parameters. Therefore making computations is more involved with sensorless implementation, and these computations further involve the use of transcendental functions and filters.

Phase currents measured with the ADC are first converted into proper current values. Third-order filters are employed to reduce the noise and other undesired effects and integrate the flux continuously as required. Finally, the inverse tangent (ARCTAN) function is used to derive the rotor angle. The speed is estimated based on two consecutive rotor angle computations again with some filtering employed. All of these calculations take a significant amount of time and consume the CPU bandwidth. This is where the FPU helps to reduce the CPU bandwidth, and it also reduces the code size.

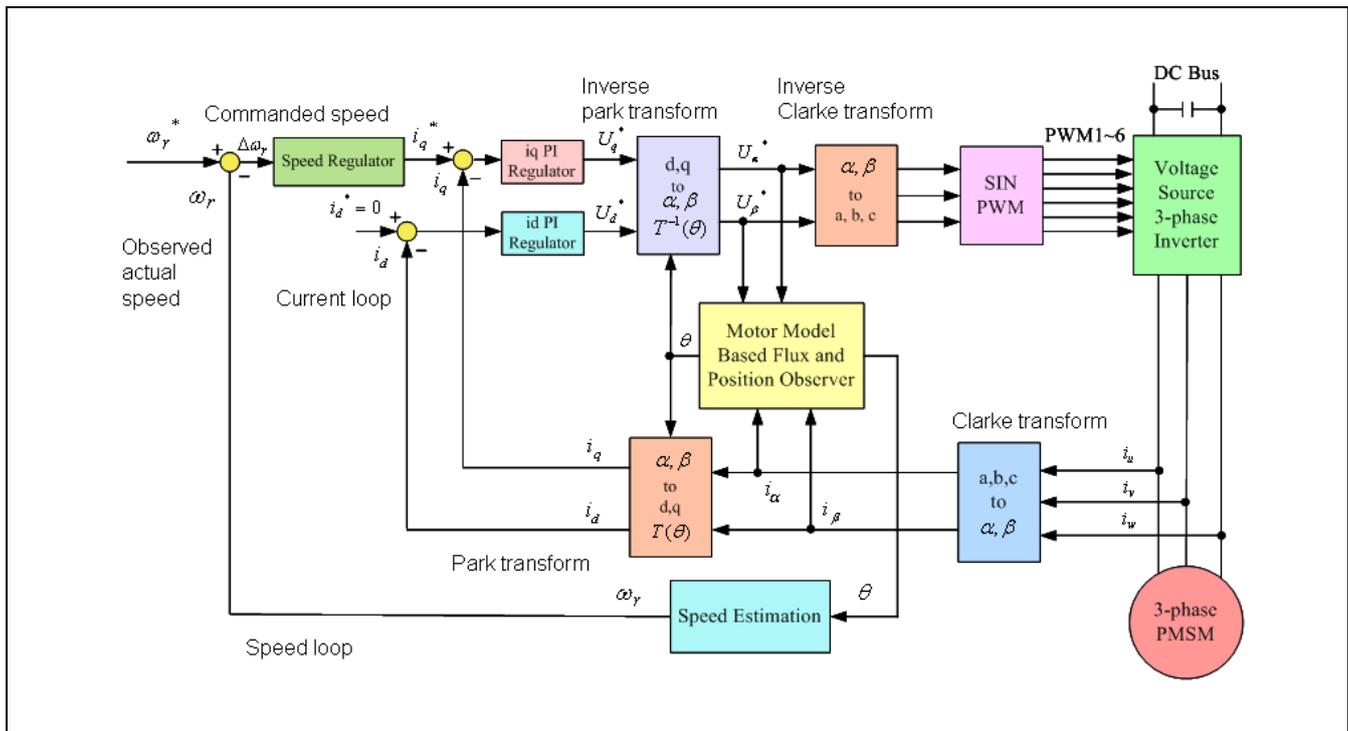


Figure 9 Block Diagram of Sensorless Vector Control

As shown, sensorless vector control (SVC) uses the complex coordinate transformations and motor mathematical model, which requires a large amount of calculations. Thus, SVC necessitates a fast MCU with high computing capability. Currently, most of the SVC implementations are based on the fixed-point MCUs or DSPs. While a few of them adopt the floating-point processors, the cores of those processors are actually fixed-point and they are not true FPU implementation.

7. Software Descriptions

The three-shunt sensorless vector control software has the following features:

- ❖ All codes are written in C language.
- ❖ The software is modularized according to the SVC block diagram (as shown in Figure 9).
- ❖ Core SVC modules can be generally used without any changes.
- ❖ I/O definitions and basic MCU drivers are automatically ported by e²Studio.
- ❖ Motor and control parameters are easily tuned through a header file of “customize.h” and GUI.

7.1 Three-Shunt Sensorless Vector Control Software Implementation

Three-shunt sensorless vector control software architecture is similar to the one in the Renesas' Application Note REU05B0103-0100/Rev.1.00. Figure 10 shows the workspace for three-shunt sensorless vector control using Renesas' e²Studio IDE (Integrated Development Environment).

- ❖ The code includes: dbstc.c; hwsetup.c, intprg.c; main.c; motorcontrol.c; resetprg.c, userif.c and vectbl.c.
- ❖ Core sensorless vector control modules for vector control transformation and speed and position observer are put in the library of 3Shnt_SVC_Lib.lib.

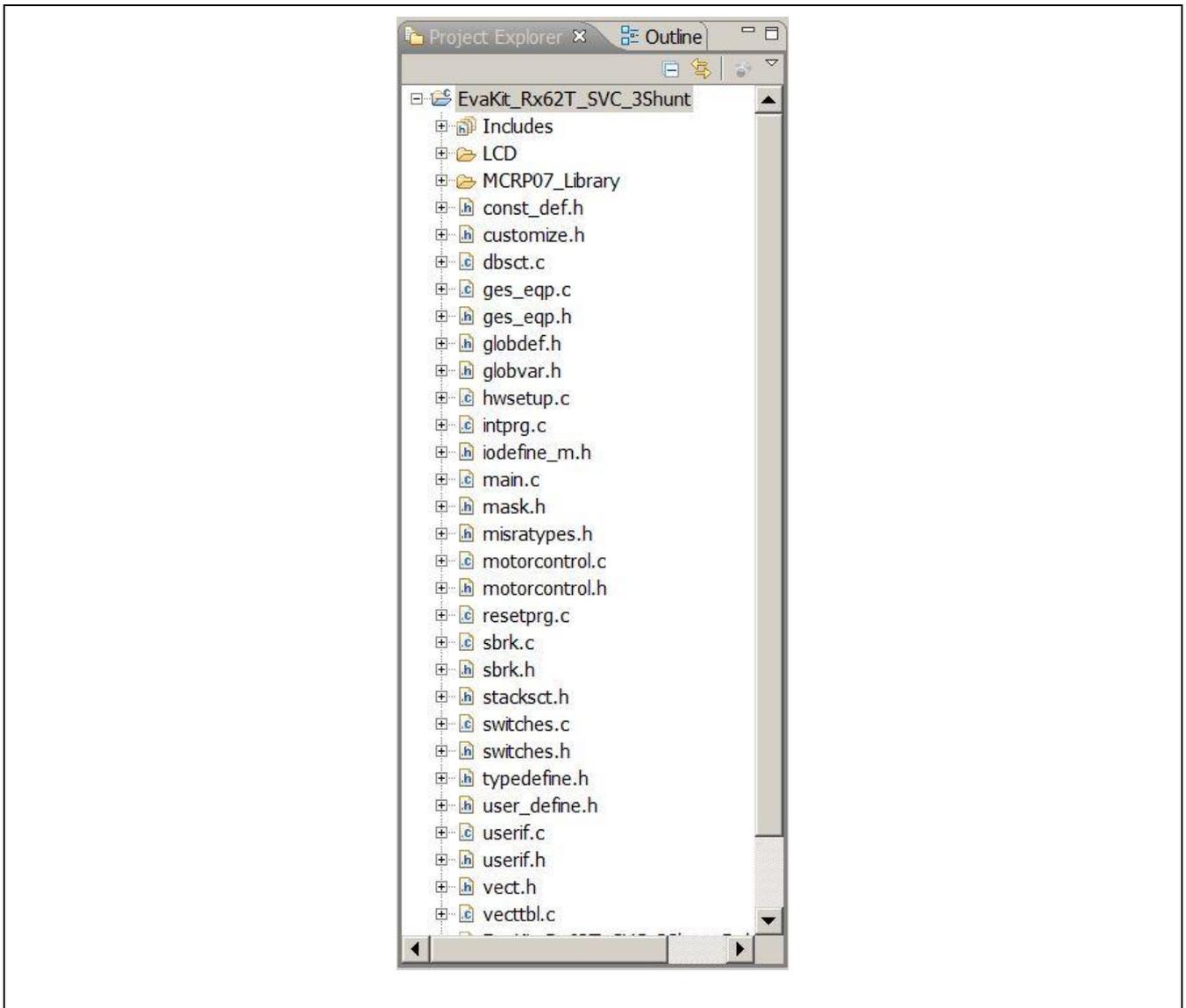


Figure 10 Three-Shunt SVC Software Workspace

7.2 Flowchart of Three-Shunt Sensorless Vector Control

The MTU3 timer interrupt is implemented for three-shunt sensorless vector control. Figure 11 is a flowchart of the PWM interrupt. It starts with the open loop, and then switches to the closed speed loop.

The procedures in the PWM interrupt of MC_ConInt() are:

- ❖ Motor phase currents and DC bus voltages are first sampled;
- ❖ When the motor powers on, the startup procedure handles the open loop;
- ❖ After the motor powers on, the system switches into the closed speed loop;
- ❖ The rotor position and the speed are estimated in sync with the carrier frequency in order to rapidly update the position and the speed;
- ❖ The current PI controller outputs of v_d and v_q are transformed back to the three-phase voltages of v_u , v_v , and v_w , which are used to calculate PWM duty ratios to drive motor to the desired voltages.

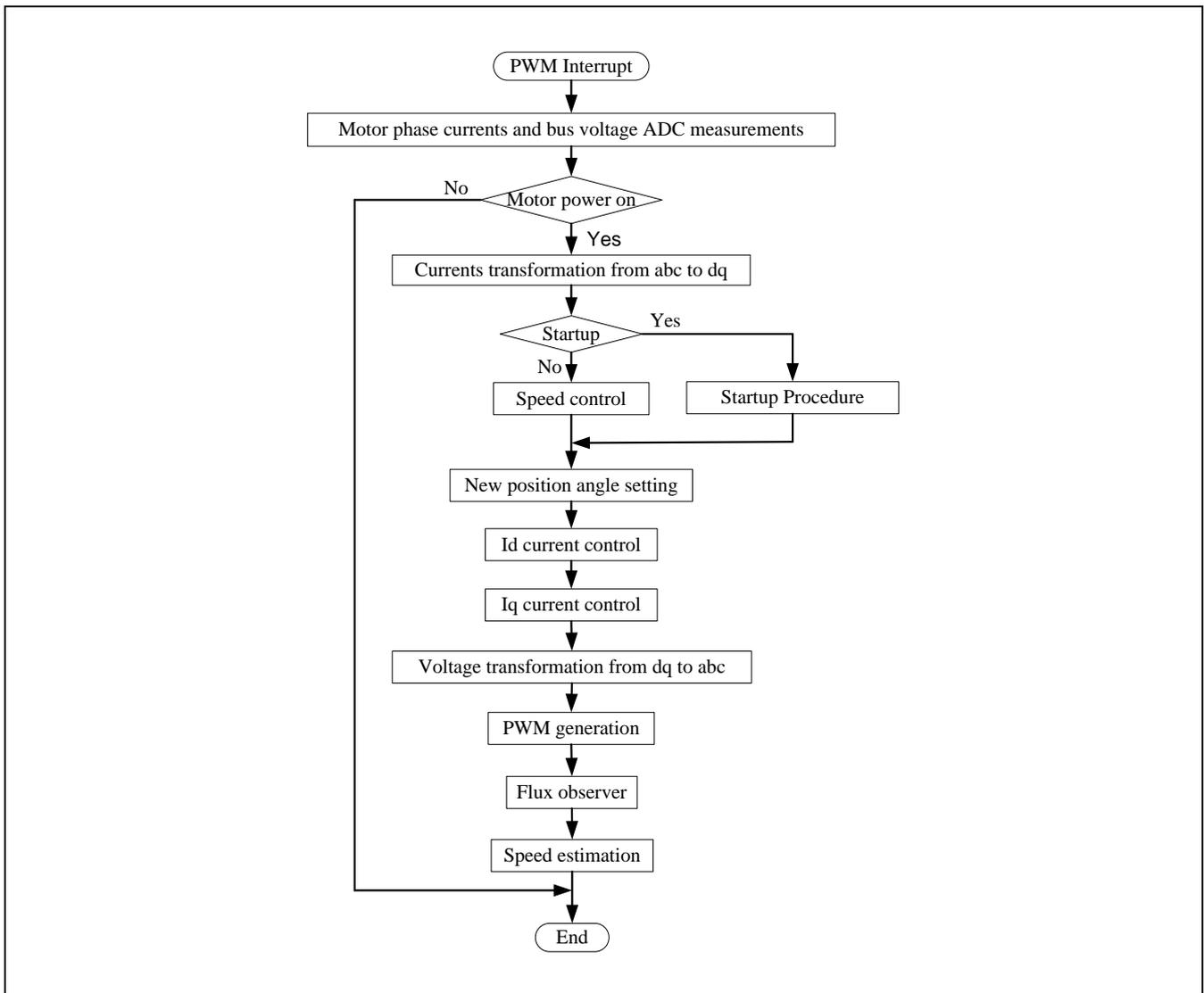


Figure 11 Flowchart of PWM Interrupt

Figure 12 shows a flowchart of the three-shunt sensorless vector control algorithm.

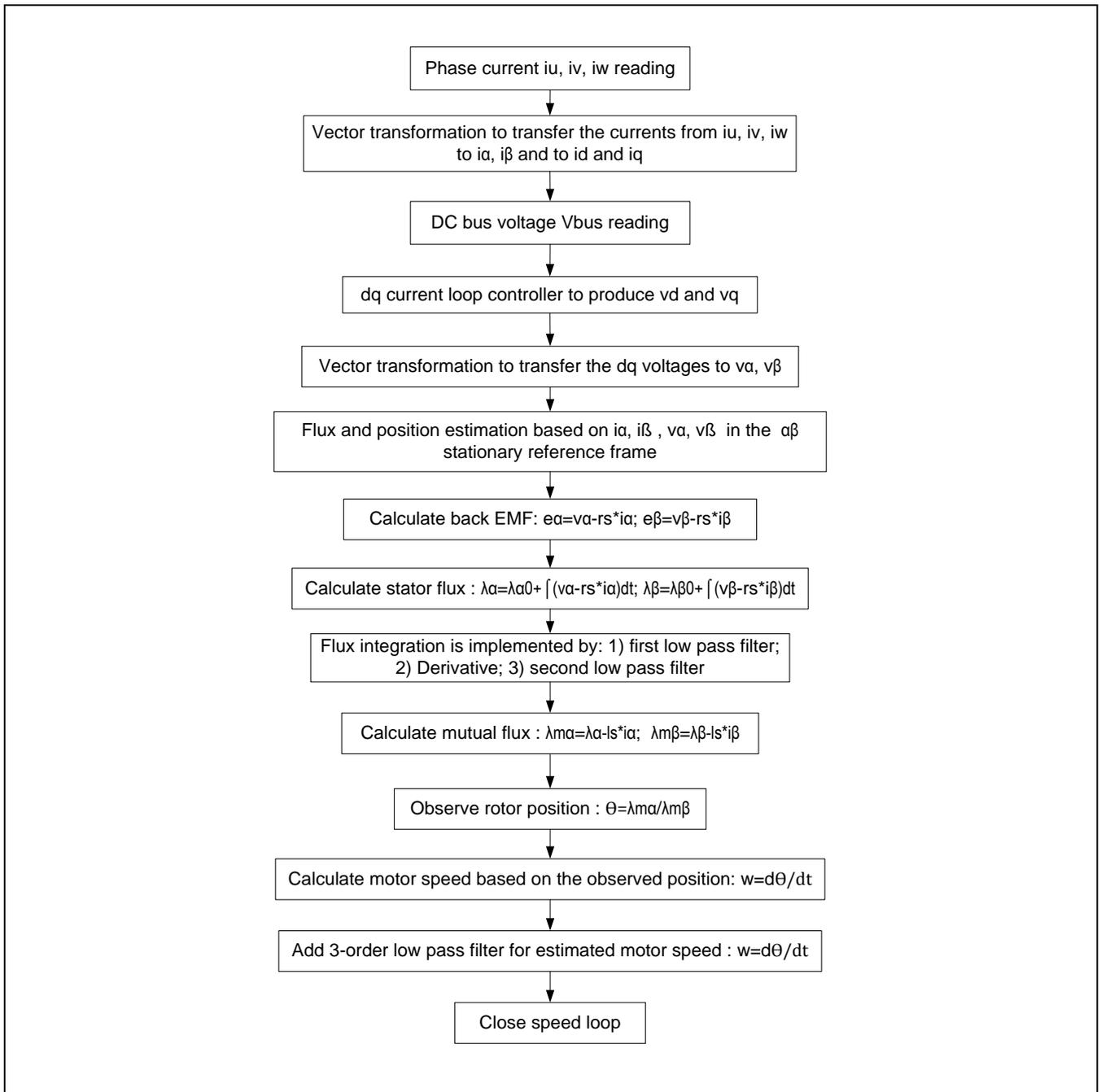


Figure 12 Flowchart of the Three-Shunt Sensorless Vector Control Algorithm

8. Motor and Control Parameter Tuning Example

8.1 Tuning through header file

Shown in Figure 13 is the BLDC motor (BLY171D-24V-4000-1000SI-05), and Table 4 shows the motor's corresponding specifications. The motor is a 24V 4-pole 3-phase BLDC equipped with Hall sensors and a 1000 line per revolution quadratic encoder with index. The rated power is 30 watts, and the rated speed is 4,000 rpm. According to the datasheet, the motor and control parameters have to be properly modified to run the three-shunt sensorless vector control.

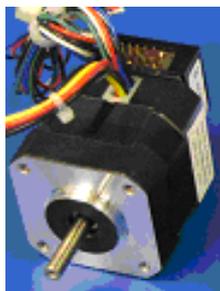


Figure 13 A BLDC Motor Included in the Evaluation Kit

Table 4 Motor Specifications

Motor Poles	8
Phase	3
Voltage	24V
Current	1.5A
Power	30 watts
Speed	4000 rpm
Inductance	2.3 mh
Stator Resistor	1.68Ω
Hall Sensors	3
Encoder	1000 pulses/rev

First, define motor parameters:

```
❖ #define R_STA_CUSTOM      8           // stator phase resistance 0.8 Ω *10
❖ #define L_SYN_CUSTOM     11          // inductance in Henry 2.3mh*10000
❖ #define POLES_CUSTOM     4           // 4 pair of poles
❖ #define I_START_CUSTOM   1.5         // startup current of 1.5A
❖ #define IQ_MAX_CUSTOM    5.0         // maximum iq current of 5.0A
❖ #define RPM_MIN_CUSTOM   500         // minimum motor speed of 500 rpm
❖ #define RPM_MAX_CUSTOM   4000        // maximum motor speed of 4,000 rpm
```

Second, tune control parameters:

```
❖ #define R_ACC_CUSTOM     1000        // acceleration ramp in 1000 rpm/sec
❖ #define KP_CUR_CUSTOM    60          // proportional gain of current controller
❖ #define KI_CUR_CUSTOM    80          // integral gain of current controller
❖ #define KP_SPD_CUSTOM    40          // proportional gain of speed controller
❖ #define KI_SPD_CUSTOM    150         // integral gain of speed controller
```

8.2 Tuning Using the GUI

The motor and control parameters can be tuned through the Renesas GUI as shown in Figure 14. Without modifying the code, the parameters can be set for different motors and applications. There is a parameter window to set up 20 separate parameters. By scrolling up and down through these parameters, the user can decide to make changes to the settings, and “Write” to EEPROM, but this doesn’t change the “customize.h” file. The original values will be restored upon clicking

“Reload”. From Figure 15, it can be seen that these parameters mirror the #defines in the “customize.h” file, and the motor and control parameters can be easily changed using the GUI.



Figure 14 Evaluation GUI Interface

INDEX	DESCRIPTION	UNIT	MIN	MAX	VALUE	VALID
1	00. Default Parameters Setting	-	0	32767	0	true
2	01. Minimum Speed	rpm	200	5000	500	true
3	02. Maximum Speed	rpm	1000	20000	2500	true
4	03. Acceleration	rpm/s	1	10000	1000	true
5	04. Deceleration	rpm/s	1	10000	1000	true
6	05. Polar couples	-	1	5	5	true
7	06. Startup Current	Apk/10	0	5000	10	true
8	07. Maximum "q" Current	Apk/10	0	5000	20	true
9	08. Stator Resistance	Ohm/10	0	5000	17	true
10	09. Synchronous Inductance	Henry/10000	0	5000	12	true
11	10. Startup Time	ms	300	10000	1000	true
12	11. Current Loop Kp	-	0	2047	60	true
13	12. Current Loop Ki	-	0	1023	80	true
14	13. Speed Loop Kp	-	0	4095	10	true
15	14. Speed Loop Ki	-	0	4095	100	true
16	15. Startup offset V	V/10	0	32767	0	true
17	16. Startup delta V	V/10	0	32767	0	true
18	17. PI Tuning trigger	-	0	32767	0	true
19	18. Free	-	0	32767	0	true
20	19. Free	-	0	32767	0	true

Figure 15 Parameter Window

Demonstration Guide

8.3 Introduction to the Demonstration Guide

The purpose of this Demonstration Guide is to help users get up and running quickly with the RX62T motor control kit (YMCRPRX62T). The RX62T Microcontroller is pre-programmed to run “Three-shunt Sensorless Vector Control with External Amplifier”. If the demo board has been programmed with a different demo algorithm, reprogram the board via the E1 programmer/debugger to demonstrate the Three-Shunt Sensorless Vector Control of PMSM. Later sections will explain how to (1) setup the demo board, (2) build and debug the demo project with e²studio, and (3) run the GUI application. The user needs to connect the motor and the power supply to experience the efficient motor control capabilities of the Renesas RX62T microcontroller.

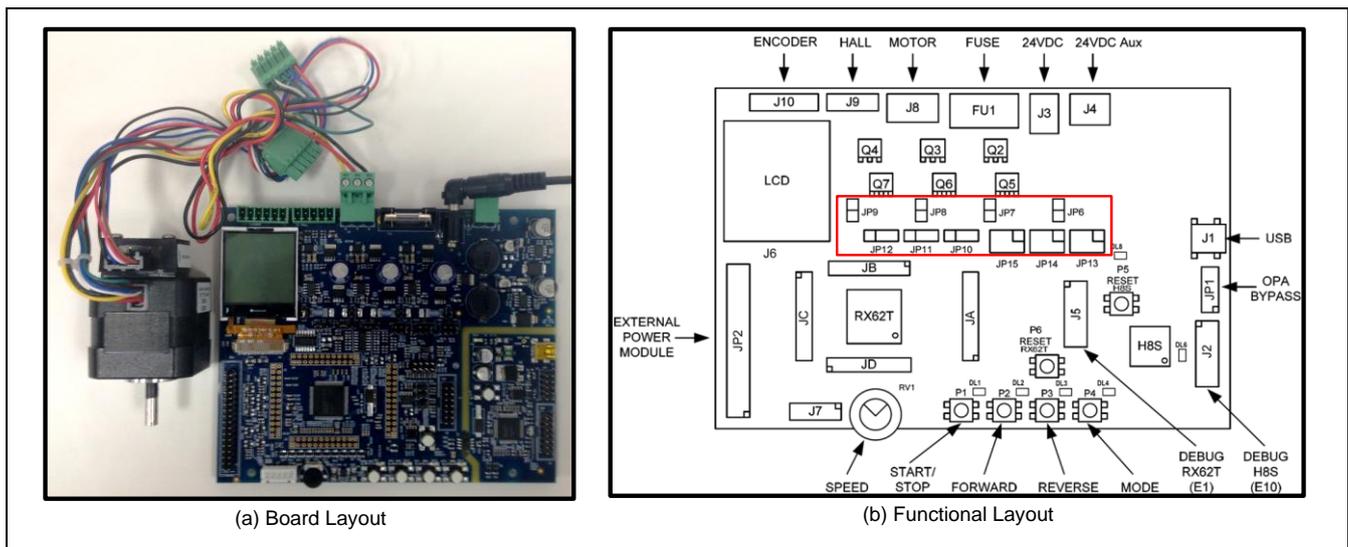
Caution: Do not connect power to the board until all instructions are followed.

The Demo contains the following items:

- RX62T Motor Control Evaluation Board (YMCRPRX62T)
- One BLDC motor with a 3-way Molex connector and encoder cable
- 24V DC power supply
- E1 debugger
- Mini-USB cable
- CD ROM for motor firmware and application GUI

8.4 Demo Board Setup

Figure 16 (a) shows the board with the motor connected to **J8** and the power supply to **J3**. There are four push-buttons, a thumb-wheel potentiometer, a graphic LCD, and a few simple steps to quickly operate the motor out of the box. For debugging or programming, the user needs to connect **J5** with E1. Use the Mini-USB connector, **J1**, in the evaluation board for communication to the GUI.



For standalone mode, press and hold **P4** (mode) button during power cycle or **P6** (reset). Then, release the **P4** button. Now the board is in standalone mode. Press **P1** (start/stop) to start or stop motor. Set **RV1** for motor speed and change motor rotation direction by pressing **P2** (forward) or **P3** (reverse) button. **P4** toggles different modes on the LCD.

Note: The user needs to set speed to 2000 rpm or more to run the motor in this demo. GUI mode will be explained in section 8.6.

8.5 Build Project and Debug Operation with e²studio

To generate the firmware program for demonstration, the provided zip file must be imported to the project workspace using e²studio IDE revision 3.0 or higher. The following steps will explain the procedure for importing the project and setting up the debugger in the e²studio IDE.

8.5.1 Build Project Procedure in e²studio

Before importing the project, the user needs to install e²studio version 3.0 or higher and Renesas compiler CCRX revision v1.02.01. *Note: This demo will only use Renesas compiler CCRX revision v1.02.01.* The user will need to create a new file folder in Windows Explorer. Open the e²studio IDE as shown in Figure 17 and proceed with the following steps:

- Step 1.** Browse or type the newly created file folder path in the “Workspace Launcher” window and click the <OK> button.



Figure 17 e²studio IDE Start-up Windows and Workspace Launcher

- Step 2.** Select “Import” from the “File” pull-down menu.

After selecting “Import”, Figure 18 shows a *Select* dialog box that prompts the user to “Create new projects from an archive file or directory.”

- Step 3.** Select “Existing Projects into Workspace” from the *Select* pop-up dialog box and click the <Next> button.

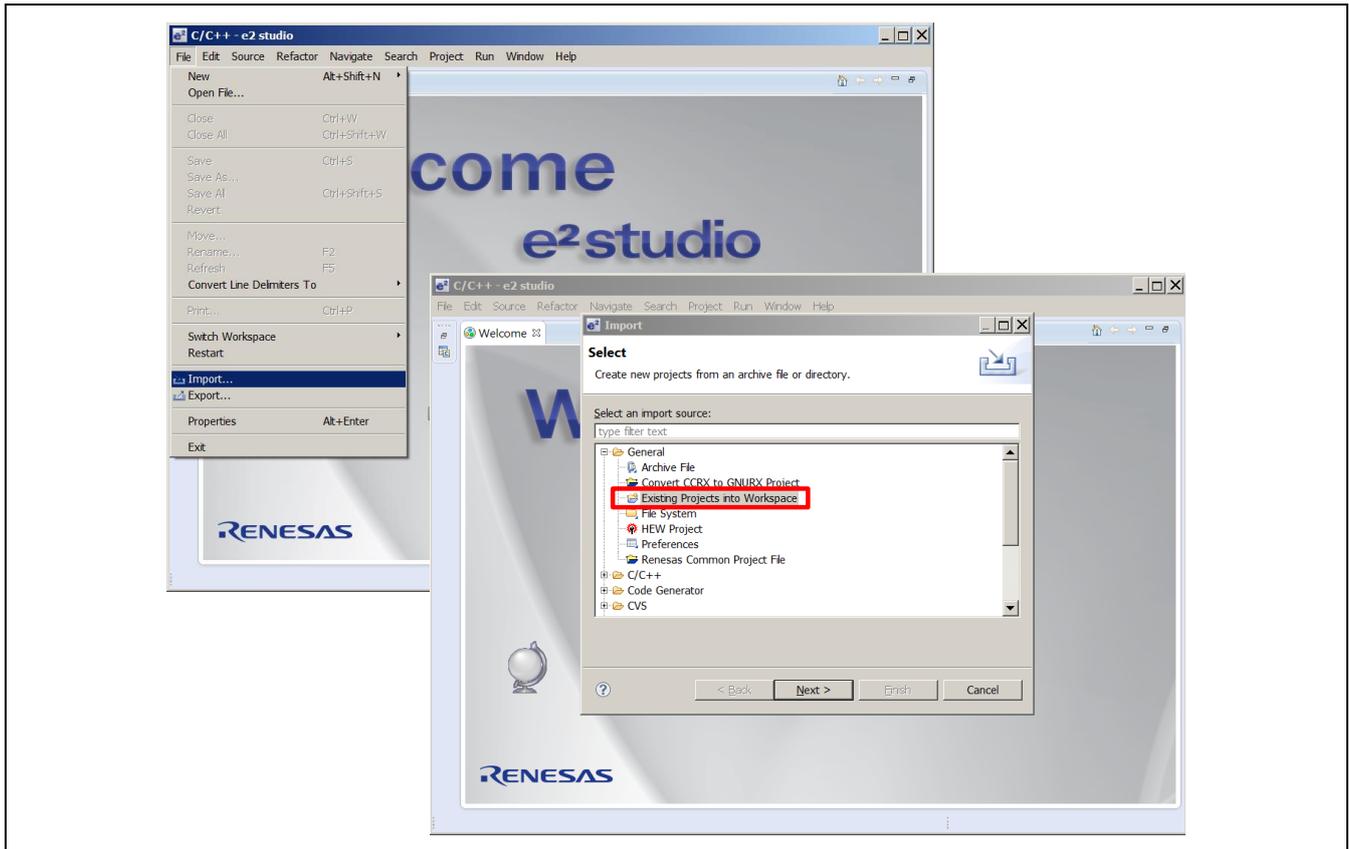


Figure 18 Importing Projects into the Workspace 1 of 2

After clicking the <Next> Button, the *Import Projects* dialog box in Figure 19 prompts the user to “Select a directory to search for existing Eclipse projects.”

- Step 4.** Select the Radio Button “Select archive file” and click <Browse> to locate the Single-Shunt Sensorless Vector Control with Internal PGA zip file to import into the workspace.

The selected project will then appear with a checked box in the *Projects* message box as seen in Figure 19.

- Step 5.** Check the “Add project to working sets” check box, and then click the <Finish> button to complete the project import.

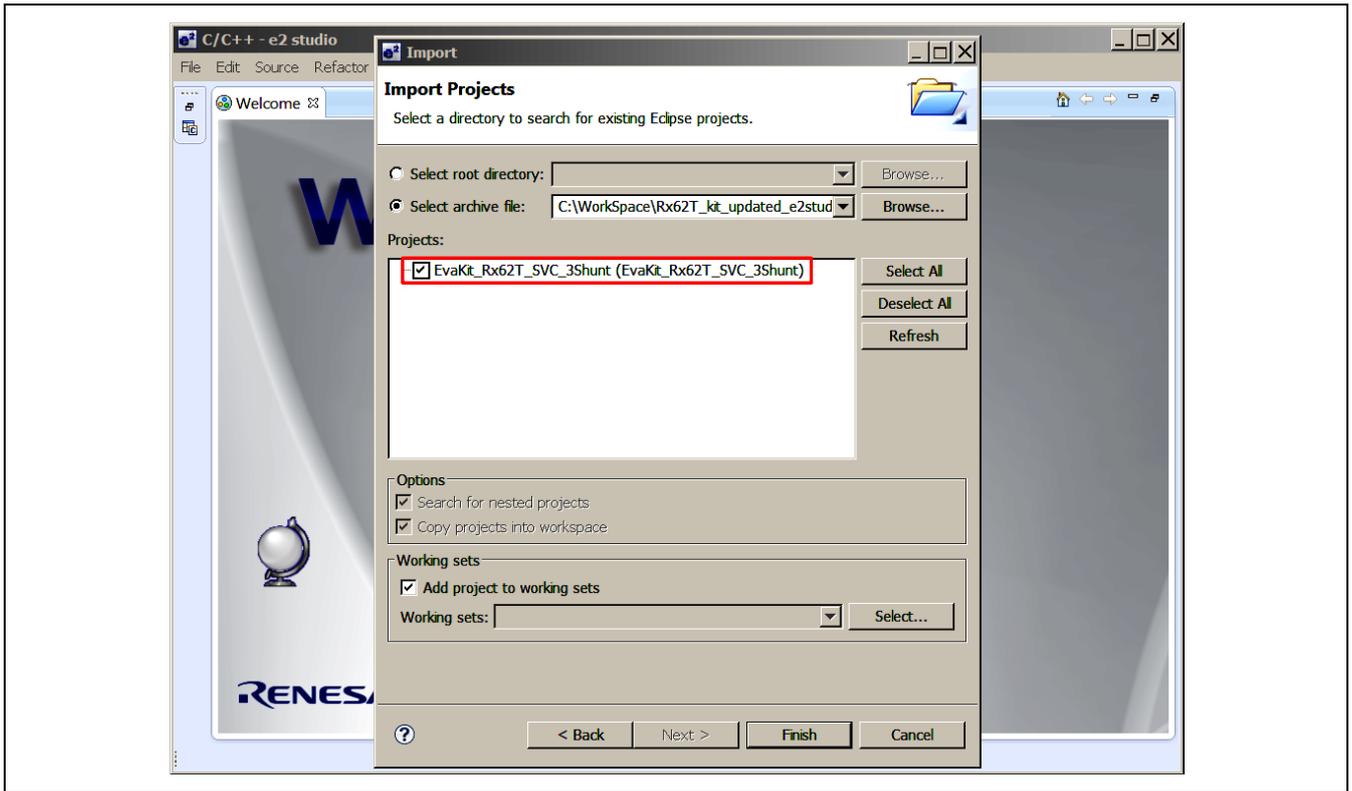


Figure 19 Importing Projects into the Workspace 2 of 2

If the file does not appear with a check box in the *Projects* message box, the selected zip file is the wrong zip file type, or it was not properly exported. If the file already exists in the workspace, then the user will see a message that states, “Some projects cannot be imported because they already exist in the workspace.”

After clicking <Finish>, the imported project is now in the e²studio workspace shown in Figure 20, and the project should be in Debug mode by default.

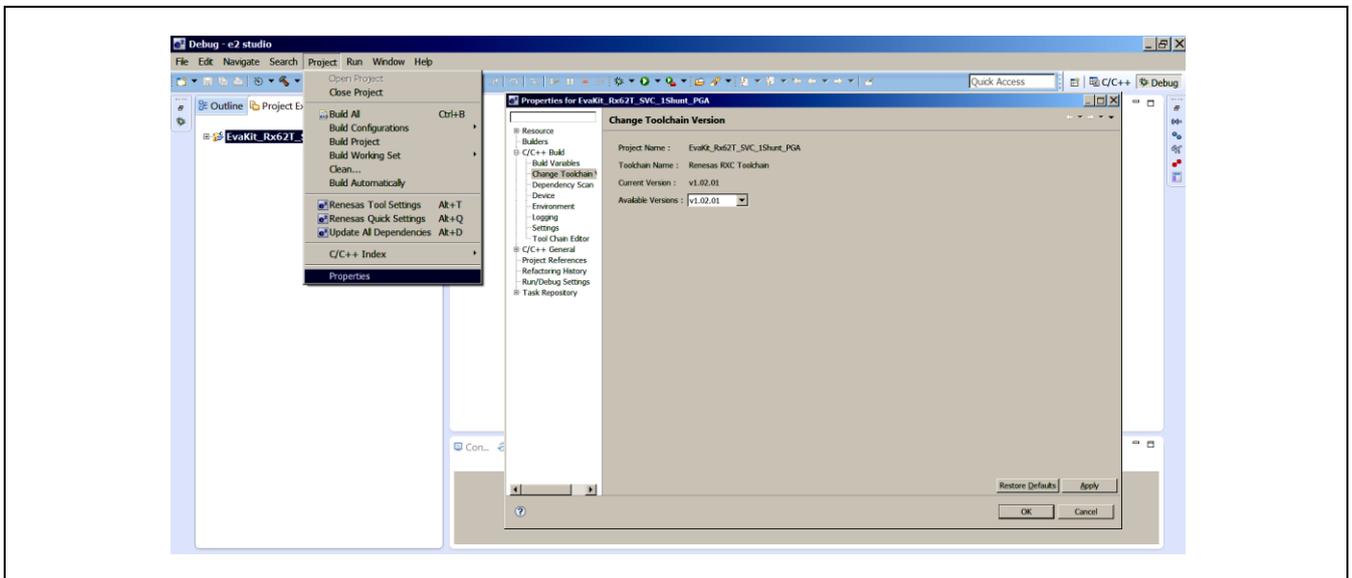


Figure 20 Setting the Toolchain Version in the e²studio Workspace

- Step 6.** Select “Properties” from the “Project” pull-down menu and expand “C/C++ Build” section. Select the “Change Toolchain Version” option and set the “Available Versions” to v1.02.01.
- Step 7.** Select the “Clean” command from the “Project” pull-down menu for cleaning and rebuilding the project.

Figure 21 shows the “Clean” Windows dialog box.

- Step 8.** Check the “Start a build immediately” option and select the Radio buttons “Clean all projects” and “Build the entire workspace.” Then click the <OK> button to clean and rebuild all projects in the workspace.

For debugging, the target firmware (.x file) is generated in the “Binaries” workspace folder shown in Figure 22. For release, set the active project to release mode for building projects. The target firmware (.mot file) is generated in the workspace “Release” folder.

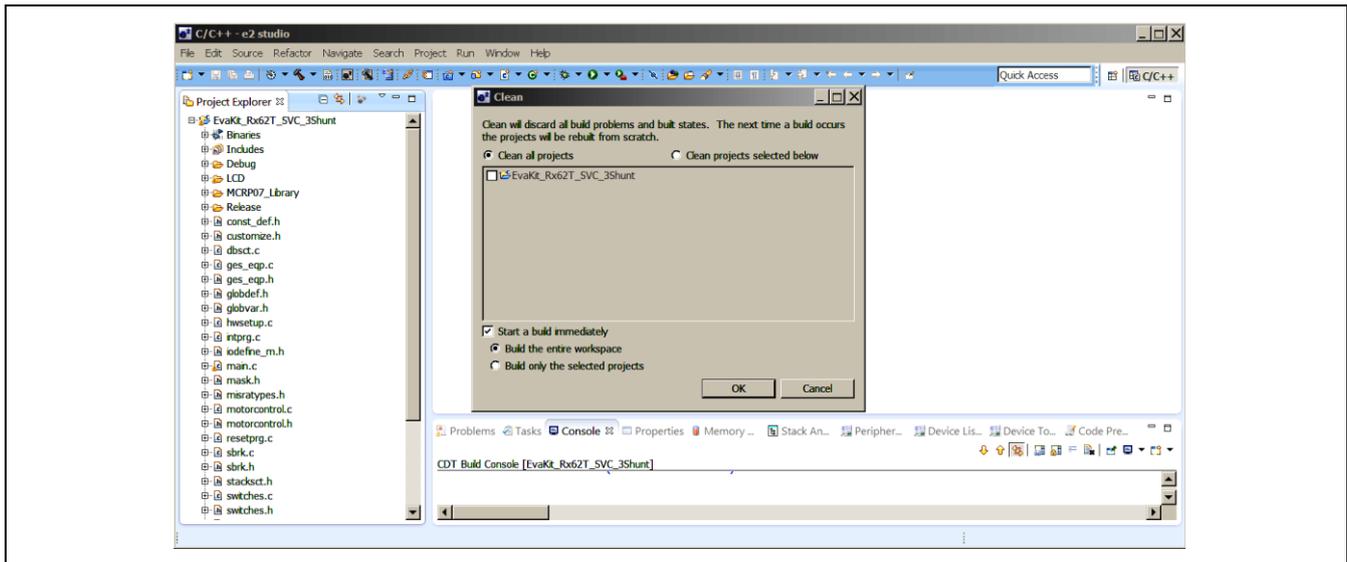


Figure 21 Clean Message Box

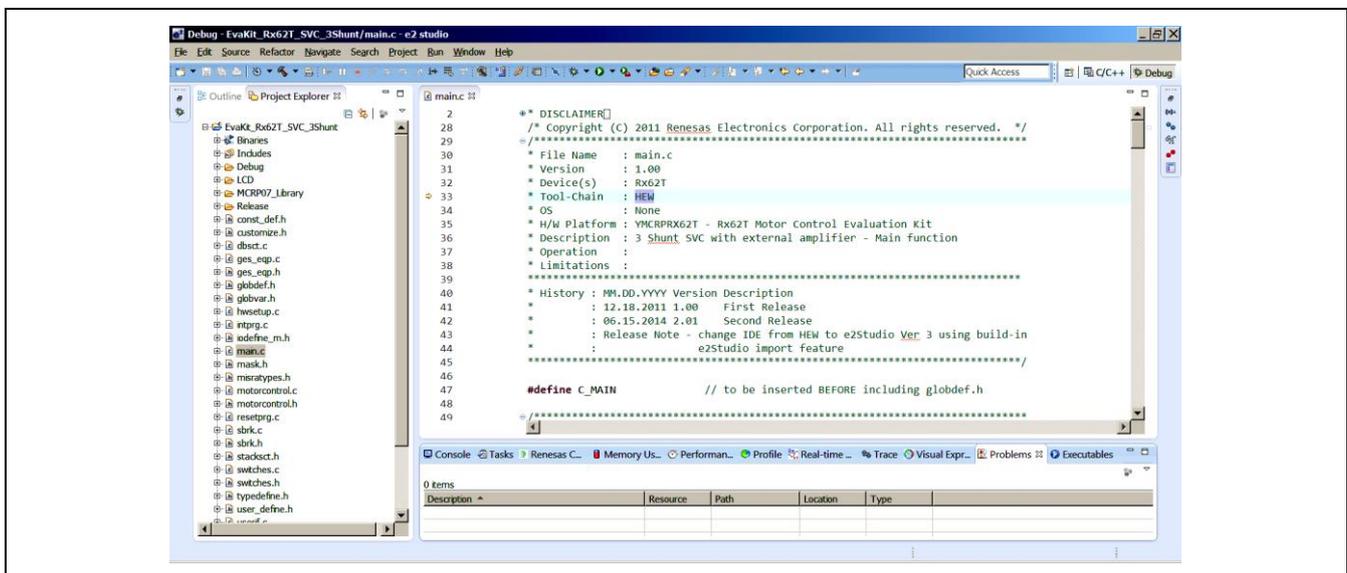
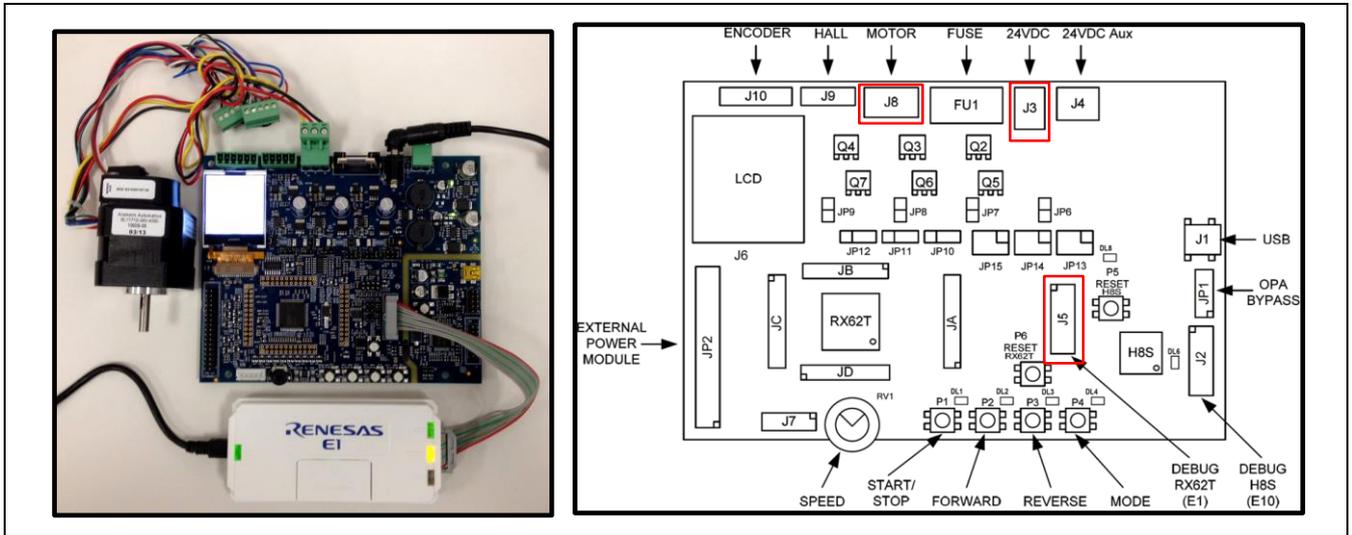


Figure 22 Target Firmware in Workspace

8.5.2 Debug Procedure in e²studio IDE

After generating the target firmware, the user is now ready to setup the debug interface through the E1 debugger. The E1 debugger is necessary as an interface from the software to the hardware. Even if there is no need for any “debugging,” this procedure is still necessary to reprogram the board using the provided algorithms. Connect the 24V DC power to **J3**, the E1 Debugger to **J5**, and the motor to the **J8** connector. The connections are shown in Figure 23. Check the recommended jumper settings for this demo (refer to section 8.4).



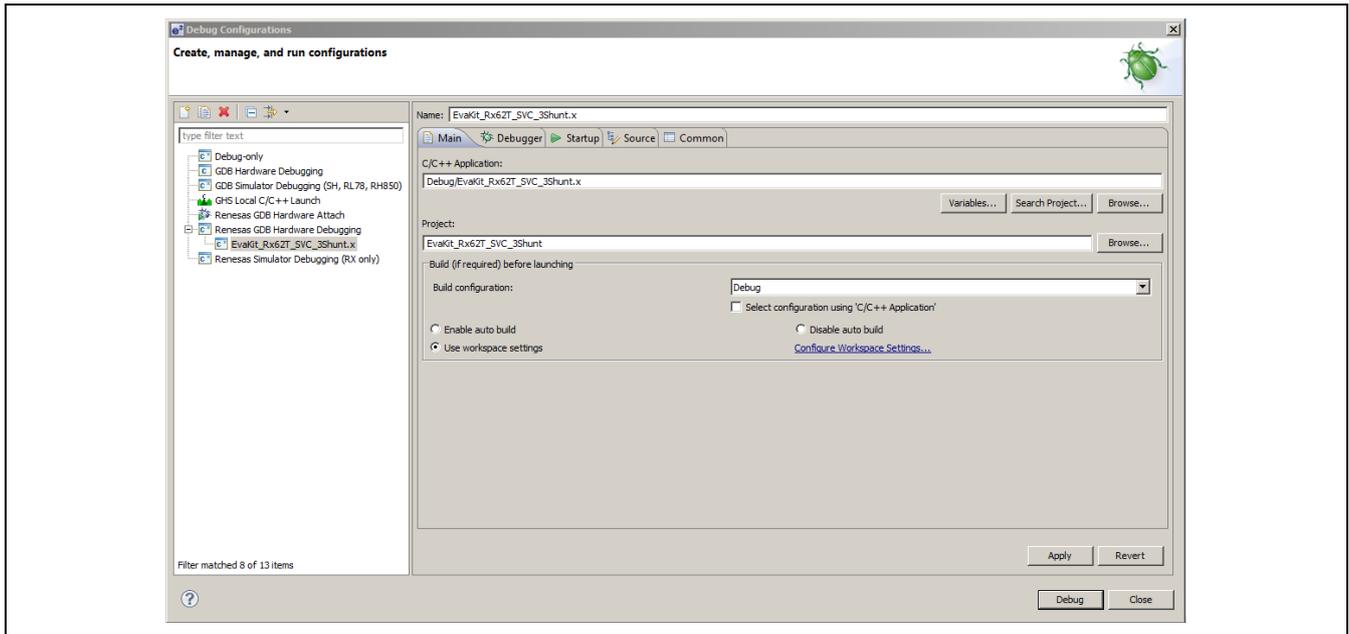


Figure 25 Debug Configuration Main Dialog Box

- Step 4.** Select the “Debugger” tab as shown in Figure 26.
- Step 5.** Select the “GDB Settings” sub-tab under the “Debugger” tab and set the “Debugger hardware” to “E1” and “Target Device” to “R5F562TA”.

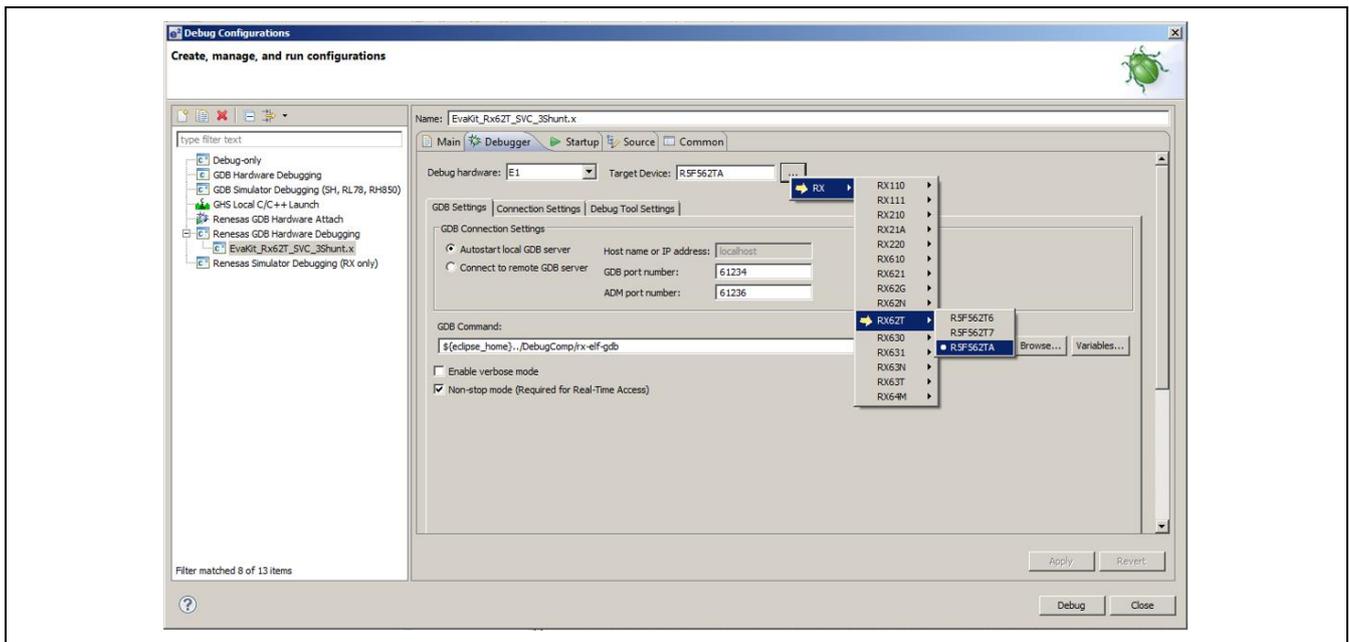


Figure 26 Debug Configuration Debugger Dialog Box

- Step 6.** Select the “Connection Settings” sub-tab and change the “External Frequency” value to 12.00 MHz and “JTag Clock Frequency” to 12.38 MHz. Set “Power” to “No.”
- Step 7.** Select the “Debug Tool Settings” sub-tab under the “Debugger” tab and select “Big Endian” in the “Endian” setting under “Memory.” Then click the <Apply> button.

The typical debug settings for this demo are shown with RED boxes in Figure 27 and in Figure 28.

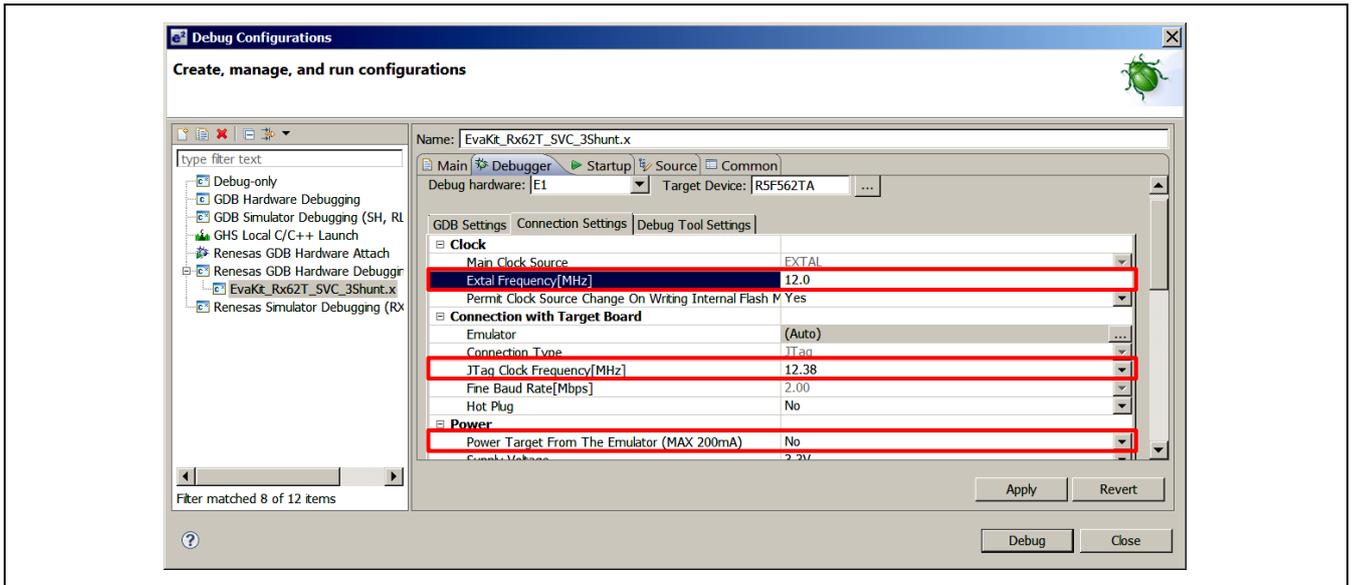


Figure 27 Debug Configuration Dialog Box 1 of 2

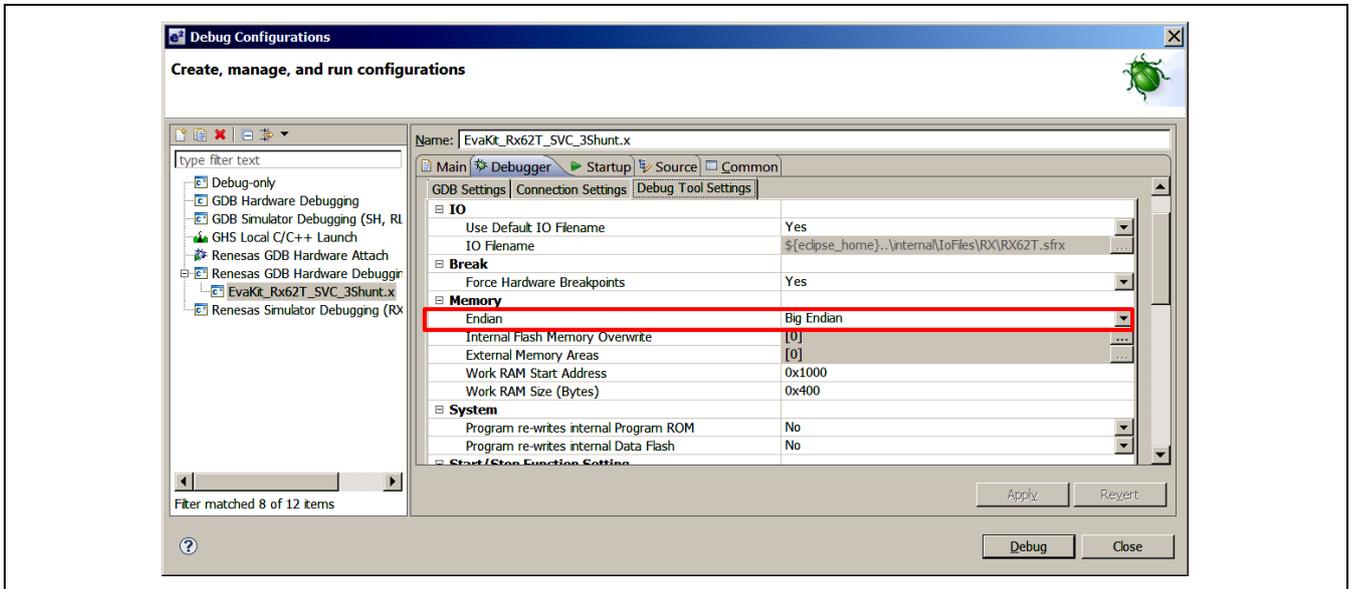


Figure 28 Debug Configuration Dialog Box 2 of 2

Step 8. Check the target board power is ON and verify the connections through the PC, E1 debugger, and the target board.

Clicking the <Debug> button in the “Debug Configurations” dialog box will download the firmware to the target board.

Step 9. Click the “Resume” icon  or press the F8 Key to run the program. This may require the user to press the “Resume” icon or F8 multiple times depending on the delay in the code. The icon should turn gray when the program is running.

The LED **DL1** will blink at about a one second rate continuously while running the target board

8.6 GUI Operation

This operation requires the demo board to be connected to the PC using the supplied Mini-USB cable.

Step 1. Connect the Mini-USB cable to **J1** from the PC

LED **DL8** is on when the USB bus power is applied to evaluation board. The PC will recognize the new hardware and will launch the driver installation screen. Follow the instruction from the “Message Box” to install respective USB device driver.

Note: Separate instructions for the USB device driver installation are provided in the Quick Start Guide or the driver will install automatically depending on the CD ROM installer. Figure 29 shows the necessary connections and LED designations.

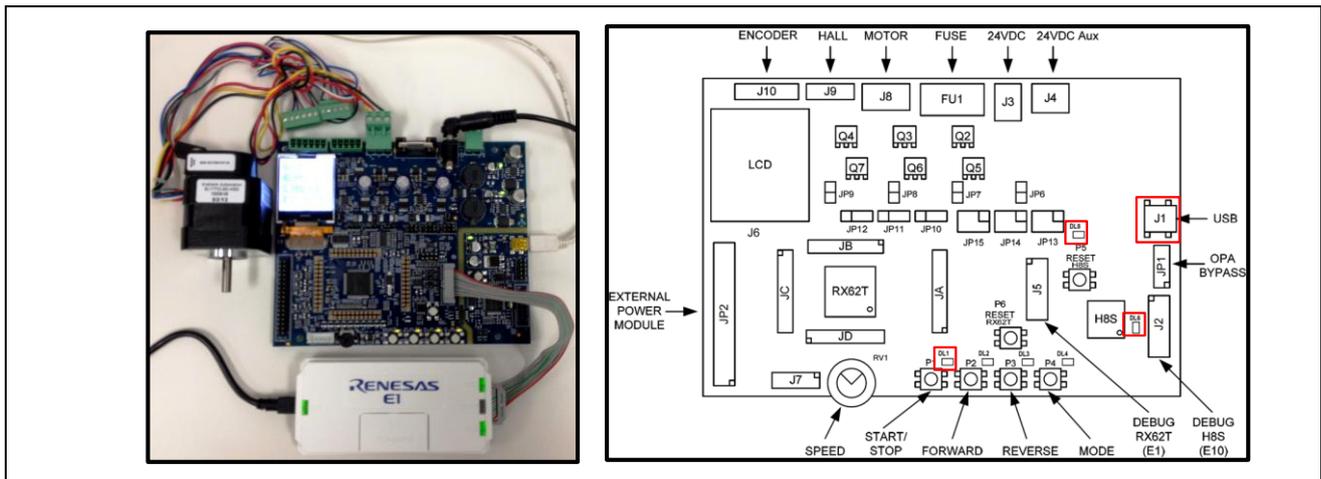


Figure 29 Running the Demo with e²studio and the GUI

Step 2. Start the GUI program by double clicking on the Motor Control Demo icon  or select the “Motor Control Demo” program from the Windows taskbar “Start” in “All Programs” under the “Motor Control Demonstrator” folder.

The GUI program screen will launch and display as shown in Figure 30. For a serial port update, wait for a few seconds to get the “Connect” button highlighted and then proceed to Step 3.

Note: If the “Connect” button is not highlighted, the GUI couldn’t find the correct USB device driver for COM port setting.

Step 3. Click the “Communication Settings” tab on the top left of the GUI seen in Figure 30 and select “Auto detect” under the serial port drop-down tab.

Step 4. Click <Connect>.

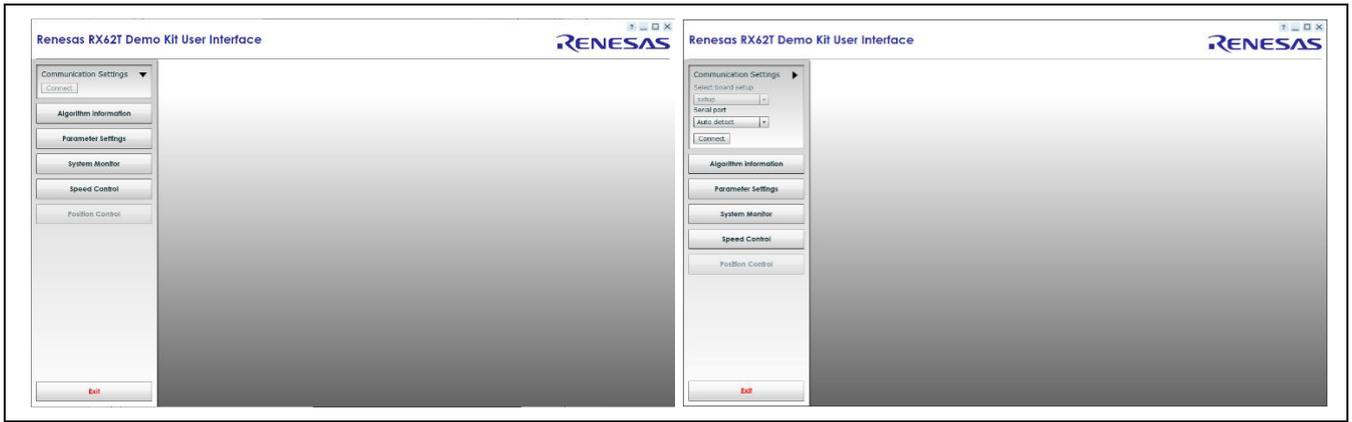


Figure 30 Connecting to the Motor Control Demo GUI 1 of 2

After successfully connecting with the target board, the “Communication Settings” area will change to a green color and the “Connect” button will change to “Disconnect.” The LED **DL6** will blink while communicating between the target board and GUI.

Figure 31 shows the GUI after a successful connection.

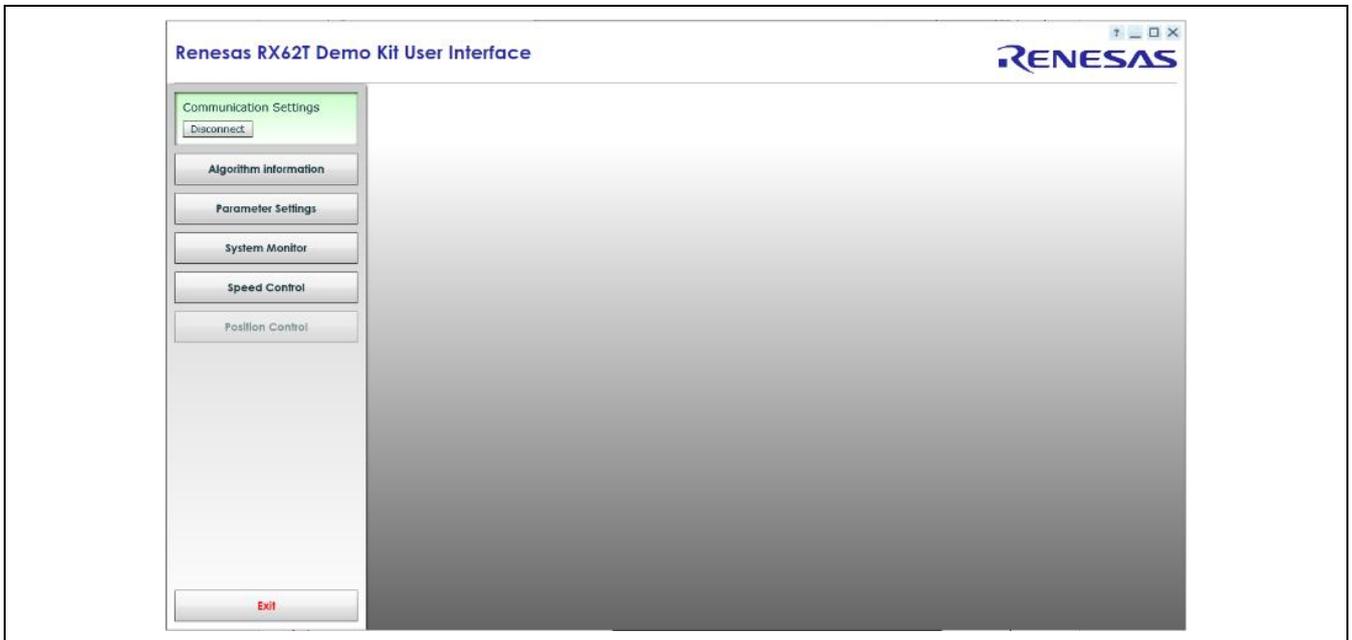


Figure 31 Connecting to the Motor Control Demo GUI 2 of 2

The GUI will detect the programmed algorithm. In this case the “Three-Shunt Sensorless Vector Control with External Amplifier” will be used. After connection, the “Speed Control” button is active while the “Position Control” button is grayed out. The user can check with the “Algorithm Information” message box which shows a valid algorithm. Clicking the “Verify Jumper Settings” button shows Table 5 in the GUI. Figure 32 shows the “Algorithm Information” dialog box. Follow the below procedure for using the GUI.

The LED **DL1** will be blinking continuously while running demo with no fault occurrence. If a fault occurs, the LED **DL2** will flash and **DL1** will remain illuminated without flashing. If a fault occurs, press **P6** (reset) and check if **DL1** begins to blink. If pressing **P6** does not fix the fault, disconnect and reconnect the E1 debugger and the Mini-USB and reprogram the board using the steps discussed in section 8.5.2

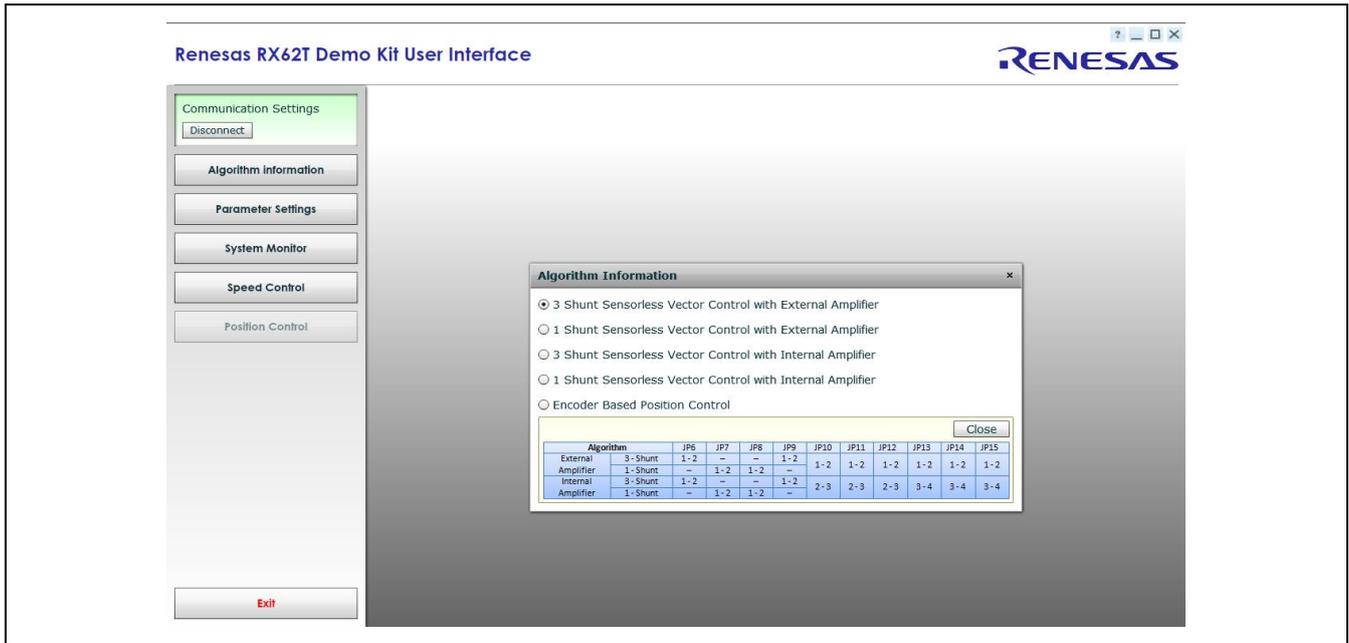


Figure 32 Algorithm Information in the GUI Application

Step 5. Click the <Speed Control> button.

Step 6. Set the speed arbitrarily by dragging the indicator needle to the right or left as shown in Figure 33. The speed can also be manually typed into the dialog box below the needle shown in Figure 33.

Note: The user needs to set the directional speed value from 2000 to 4000 rpm to run this demo.

The motor shaft should rotate with the setting speed. Returning the control needle to zero position stops the motor. By default, the demo sets parameter values for speed.

- Minimum speed 2000 rpm
- Maximum speed 4000 rpm
- Acceleration 5000 rpm/s
- Deceleration 5000 rpm/s
- Startup time 1000 ms

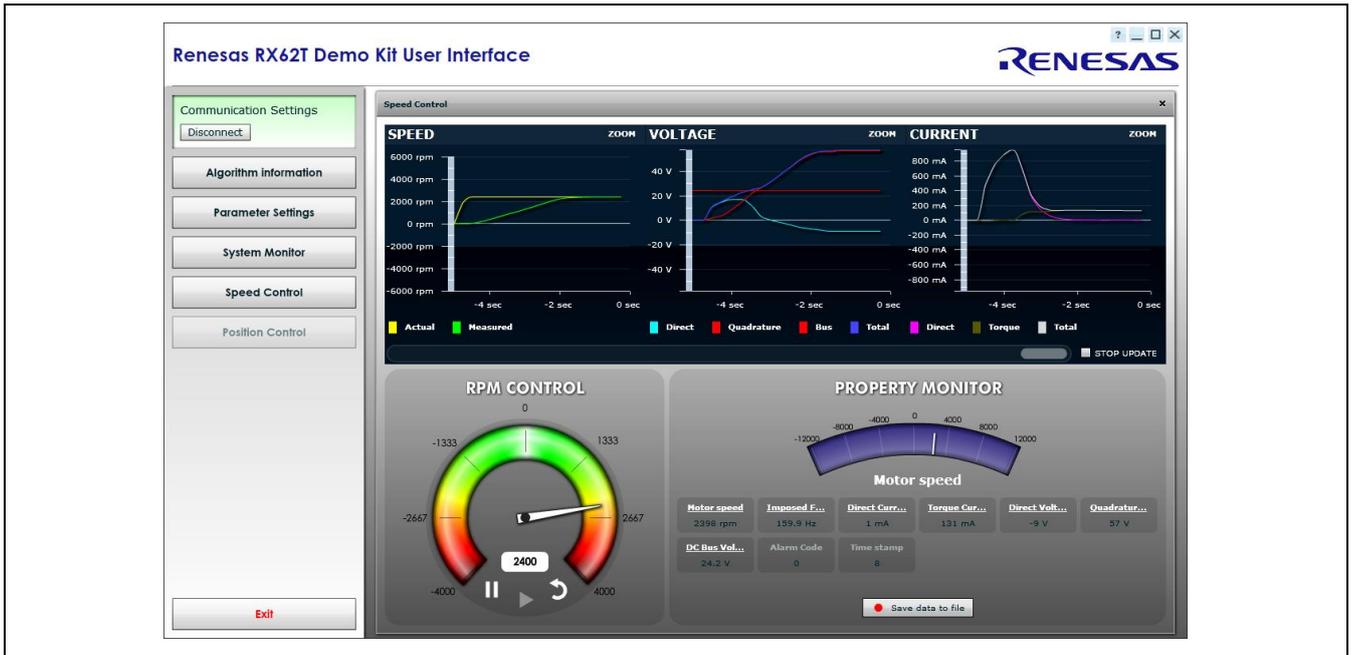


Figure 33 Setting the Speed to Turn the Motor in the GUI Application

Step 7. Click the “Parameter Settings” button

The “Parameter Settings” feature can be used to manually adjust the preset variables using the GUI. Figure 34 shows the “Parameter Settings” dialog box.

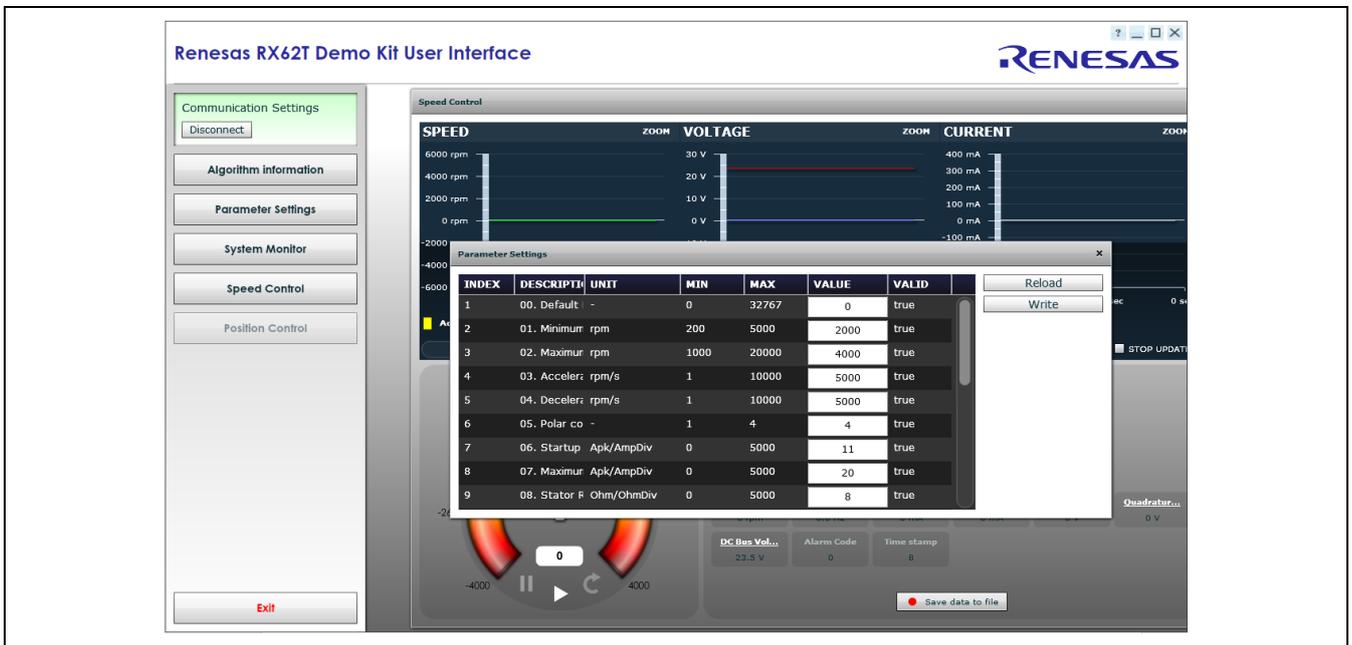


Figure 34 Changing the Parameter Settings in the GUI Application

If standalone mode is used the target board will “Disconnect” and no longer be communicating with the GUI. The LED **DL6** will be on, but it will no longer be blinking. In order to reconnect from standalone mode, press **P6** (reset) and “Connect” to the GUI using the User Interface.

To terminate the GUI application, return the control needle to zero position, press the “Disconnect” button and then press the “Exit” button to close the application.

Appendix A - References

1. RX62T Group User's Manual: Hardware, R01UH0034EJ0110, April 20, 2011
2. DevCon 2010 Courses:
 - ID-620C, Complete Motor Control Integration with RX62T.
 - ID-623C, Understanding Sensorless Vector Control with Floating-Point Unit (FPU) Implementation.
3. Application Note of Sensorless Vector Control of Three-Phase PMSM Motors, REU05B0103-0100/Rev.1.00, March, 2009
4. Application Note of Mcrp05: Brushless AC Motor Reference Platform, REU05B0051-0100, Feb, 2009
5. Huangsheng Xu, Rohan Hubin, and Dave Cocca, "Sensorless Vector Control of PMSM Motor Using One Shunt Current Detection", IEEE-IAS 2008, Oct. 5-9, Edmonton, Alberta, Canada.
6. Huangsheng Xu, and Yashvant Jani, "Understanding Sensorless Vector Control for Brushless DC Motors", ESC-2008, Embedded System Silicon Valley Conference, April 15-17, San Jose, California, USA.

Website and Support

Renesas Electronics Website

<http://www.renesas.com/>

Inquiries

<http://www.renesas.com/inquiry>

All trademarks and registered trademarks are the property of their respective owners.

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Nov. 18, 2011.	—	First edition issued
2.00	Jan. 31, 2014.	—	Second edition issued
2.01	Jul. 30, 2014.	18	Demonstration Guide added

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the “Handling of MPU/MCU Products” and the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins

Handle unused pins in accord 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 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. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

The state of the product is undefined at the moment 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 moment 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 moment 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 moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

Access to reserved addresses is prohibited.

- The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has 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. Moreover, 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.

5. Differences between Products

Before changing from one product to another, i.e. to one with a different type number, confirm that the change will not lead to problems.

- The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type numbers, implement a system-evaluation test for each of the products.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
 2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
 3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
 4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
 5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
"Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots etc.
"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.
Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
 6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
 7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or systems manufactured by you.
 8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
 9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
 10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
 11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
 12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.
- (Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.
(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.



SALES OFFICES

Renesas Electronics Corporation

<http://www.renesas.com>

Refer to "<http://www.renesas.com/>" for the latest and detailed information.

Renesas Electronics America Inc.

2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited

1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited

Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH

Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.

7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.

Unit 301, Tower A, Central Towers, 555 LanGao Rd., Putuo District, Shanghai, China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited

Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.

13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.

80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.

Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.

12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141