RH850/C1M-A Group

Controlling the Permanent Magnet Synchronous Motor with Resolver by Using RDC3A and EMU3

Introduction

This application note describes how to use the RH850/C1M-A Group microcontroller to control the permanent magnet synchronous motor equipped with a resolver by using the RDC3A and EMU3.

Target Device

RH850/C1M-A Group microcontroller

If you apply this application note to another type of microcontroller, adjust and fully evaluate contents of this application note so that they match the specifications of that microcontroller.
Contents

1. Vector Control of Permanent Magnet Synchronous Motor with Resolver .................................................. 4
   1.1 Operation of PMSM .......................................................................................................................... 4
   1.1.1 Synthetic Magnetic Field and Torque ......................................................................................... 5
   1.2 Concept of Motor Control ................................................................................................................ 5
   1.3 PWM Output Using U/V/W-Phase Voltages .................................................................................... 7
   1.3.1 Triangular Wave Comparison Method ....................................................................................... 7
   1.4 Voltage Equation for PMSM Voltages in dq Coordinate System (Reference Information) .......... 8
   1.4.1 Voltage Equation ....................................................................................................................... 8
   1.4.2 Non-Interference Control .......................................................................................................... 8

2. Specifications ............................................................................................................................................ 9

3. Operation Check Conditions .................................................................................................................. 14

4. Pins ........................................................................................................................................................ 15

5. Hardware Operation ............................................................................................................................... 16
   5.1 Overview of Operation ...................................................................................................................... 16
   5.2 R/D Converter 3A (RDC3A) ............................................................................................................. 16
   5.3 Peripheral Interconnection (PIC) ..................................................................................................... 16
   5.4 A/D Converter (ADCC) .................................................................................................................... 16
   5.5 Motor Control Timer 3 (TSG3) .......................................................................................................... 16
   5.6 Enhanced Motor Control Unit 3 (EMU3) .......................................................................................... 18
   5.6.1 EMU3 Common Unit ..................................................................................................................... 18
   5.6.2 Angle Generation IP .................................................................................................................... 19
   5.6.3 Input IP ........................................................................................................................................ 19
   5.6.3.1 Calculation of Motor Current Values ...................................................................................... 20
   5.6.3.2 dq Conversion .......................................................................................................................... 20
   5.6.4 PI Control IP ................................................................................................................................ 21
   5.6.4.1 PI Control ................................................................................................................................ 21
   5.6.5 PWM IP ....................................................................................................................................... 22
   5.6.5.1 Compensation of d-Axis and q-Axis Voltages/Non-Interference Control .................... 22
   5.6.5.2 Conversion of 3-Phase Voltages ............................................................................................. 24
   5.6.5.3 Calculation of Duty Ratios ..................................................................................................... 24
   5.6.5.4 Calculation of Compare Values ............................................................................................... 26
   5.6.6 Examples of Calculation Accuracy ............................................................................................. 28
   5.7 File Configuration .............................................................................................................................. 29
   5.8 Functions ......................................................................................................................................... 29
   5.9 Specifications of Functions .............................................................................................................. 30
   5.10 Specifications of Functions ............................................................................................................ 32

6. Reference Documents ............................................................................................................................. 39
1. Vector Control of Permanent Magnet Synchronous Motor with Resolver

This section describes the concept of vector control of a permanent magnetic synchronous motor (hereafter called the "PMSM") equipped with a resolver.

1.1 Operation of PMSM

A PMSM has a stator fixed to the motor housing and a rotor, which rotates. The stator has phase-U, phase-V, and W-phase coils mounted at angles of 120 degrees to each other. The rotor has permanent magnets built in.

Applying a voltage across a coil causes a current to flow through it, and thereby generates a magnetic field. When a permanent magnet is placed in the magnetic field, an attractive or repulsive force acts on the permanent magnet. When currents flow through the phase-U, phase-V, and W-phase coils, the rotor moves because its permanent magnets are attracted in a synthetic magnetic field made by vector synthesis of the magnetic fields generated by the coils. When the synthetic magnetic field is rotated, the rotor rotates. A magnetic field that rotates at a constant speed and has a constant magnitude can be generated by passing sinusoidal currents different in phase by 120 degrees through the phase-U, phase-V, and W-phase coils. Figure 1.1 shows the structure of the PMSM and the synthetic magnetic field.

![Figure 1.1 PMSM Structure and Synthetic Magnetic Field](image-url)
1.1.1 Synthetic Magnetic Field and Torque

Figure 1.2 shows the relationship between the synthetic magnetic field and torque. When the magnetic pole of a permanent magnet forms an angle of 90° with the direction of synthetic magnetic field, the torque (force to turn the permanent magnet) is maximized. The synthetic magnetic field perpendicular to the magnetic pole is called the "transverse magnetic field". Here, assume that the direction of the magnetic pole of the permanent magnet is the d axis and the direction of the transverse magnetic field is the q axis.

Efficient rotation of the motor requires appropriate current control to keep a transverse magnetic field generated constantly. To consider generation of the transverse magnetic field, assume that a coil is positioned in the direction of the q axis and the current to be passed through the coil is the q-axis current (Iq). The intensity of the transverse magnetic field is proportional to the intensity of the q-axis current.

If the motor speed is lower than expected, the q-axis current must be increased to increase the torque for acceleration. If the motor speed is higher than expected, the q-axis current must be reduced to reduce the torque for deceleration.

1.2 Concept of Motor Control

To efficiently control the motor, the d-axis and q-axis currents must be controlled according to the motor speed. Therefore, values of d-axis and q-axis currents must be calculated. Because the q-axis current has intensity and direction, which are treated as vector quantities, this control method is called "vector control".

The target values (command values) of d-axis and q-axis currents are determined by the difference between target motor speed and present motor speed. Simple modeling of vector control assumes the D-axis current to be 0 because the D-axis current does not contribute to any torque components. The voltage value to be output next is obtained based on the difference (deviation) between the d-axis current and q-axis current values that is calculated from the command values of d-axis current and q-axis currents and measured coil current. The control operation in which the previous output results (present motor speed and coil current value) are fed back and reflected in the next output is called "feedback control".

To calculate the voltage to be output next from the deviation of present current value, components proportional to the cumulated past deviation are added to those proportional to present deviation. This processing is repeated to control the output voltage and the coil current flowing as the result of voltage output to adjust present motor speed to the target motor speed. This control method is called "proportional-integral (PI) control."

Values of q-axis current and d-axis current are obtained by coordinate conversions (3-phase to 2-phase conversion, then rotating coordinate conversion) of coil currents. Figure 1.3 shows the coordinate conversions.
Calculate values of currents $I_α$ and $I_β$ on the $α$ and $β$ axes by 3-phase to 2-phase conversion of the values of currents $I_u$, $I_v$, and $I_w$ that are flowing through the phase-$U$, phase-$V$, and phase-$W$ coils and were measured by the n'th measurement.

$$\begin{pmatrix} I_α(n) \\ I_β(n) \end{pmatrix} = \sqrt{\frac{2}{3}} \begin{pmatrix} \cos 0 & \cos \frac{2\pi}{3} & \cos \frac{4\pi}{3} \\ \sin 0 & \sin \frac{2\pi}{3} & \sin \frac{4\pi}{3} \end{pmatrix} \begin{pmatrix} I_u(n) \\ I_v(n) \\ I_w(n) \end{pmatrix}$$

Perform rotating coordinate conversion of $I_α$ and $I_β$ obtained by the 3-phase to 2-phase conversion to calculate values of currents $I_d$ and $I_q$ on the d and q axes.

$$\begin{pmatrix} I_d(n) \\ I_q(n) \end{pmatrix} = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} I_α(n) \\ I_β(n) \end{pmatrix}$$

Calculate the deviation between $d$-axis and $q$-axis currents, and then calculate $d$-axis and $q$-axis voltages by PI control.

$$\epsilon_d(n) = I_{dt} - I_d(n)$$
$$\epsilon_q(n) = I_{qt} - I_q(n)$$

$$V_d(n) = V_d(n-1) + K_p \cdot \left\{ \epsilon_d(n) - \epsilon_d(n-1) \right\} + K_i \cdot \epsilon_d(n) \cdot \Delta t$$
$$V_q(n) = V_q(n-1) + K_p \cdot \left\{ \epsilon_q(n) - \epsilon_q(n-1) \right\} + K_i \cdot \epsilon_q(n) \cdot \Delta t$$

Remarks:
- $I_{dt}$: Command value of $d$-axis current
- $I_{qt}$: Command value of $q$-axis current
- $I_d(n)$: $d$-axis current sampled by n'th sampling
- $I_q(n)$: $q$-axis current sampled by n'th sampling
- $K_p$: Proportional gain
- $K_i$: Integration gain
- $\Delta t$: Sampling time
- $\epsilon_d$: Deviation of $d$-axis current
- $\epsilon_q$: Deviation of $q$-axis current
Convert the q-axis and d-axis voltages calculate above by coordinate conversions (fixed coordinate conversion and 2-phase to 3-phase conversion) to obtain output voltages in U, V, and W phases. The equation of fixed coordinate conversion is as follows:

\[
\begin{pmatrix}
V_d(n) \\
V_q(n)
\end{pmatrix} =
\begin{pmatrix}
\cos \theta & -\sin \theta \\
\sin \theta & \cos \theta
\end{pmatrix}
\begin{pmatrix}
V_d(n) \\
V_q(n)
\end{pmatrix}
\]

The equation of 2-phase to 3-phase conversion is as follows:

\[
\begin{pmatrix}
V_d(n) \\
V_q(n) \\
V_r(n)
\end{pmatrix} = \frac{1}{\sqrt{3}}
\begin{pmatrix}
\cos 0 & \sin 0 & 0 \\
\cos \frac{2\pi}{3} & \sin \frac{2\pi}{3} & 0 \\
\cos \frac{4\pi}{3} & \sin \frac{4\pi}{3} & 0
\end{pmatrix}
\begin{pmatrix}
V_u(n) \\
V_v(n) \\
V_w(n)
\end{pmatrix}
\]

1.3 PWM Output Using U/V/W-Phase Voltages

To actually output a PWM waveform from the microcontroller, the phase-U, phase-V, and phase-W voltages (Vu, Vv, and Vw) obtained by two-phase to three-phase voltage conversion of d-axis and q-axis voltages, which are operating quantities, must be reflected in the PWM duty. This section describes how to reflect the voltages.

1.3.1 Triangular Wave Comparison Method

The triangular wave comparison method is a PWM control method. As shown in Figure 1.4, a PWM signal is generated through magnitude comparison between the phase-U/V/W voltage waveform and a triangular wave. This method has still been used widely since the days when PWM signals were generated by using analog circuits. This triangular wave is called the “carrier wave” of which the frequency specifies the operating cycle of PWM. When the phase-U/V/W voltage is higher than the magnitude of triangular wave, the PWM signal is set to the high level (active level). When the phase-U/V/W voltage is lower than the magnitude of triangular wave, the PWM signal is set to the low level (inactive level). Changing the average voltage in such a way enables the phase-U/V/W voltage (sinusoidal wave) to be reproduced in a pseudo manner. For details about this method, see a relevant technical document.

The triangular wave comparison method is also called "triangular wave modulation". Using the triangular wave comparison method, you can perform PWM control of the three-phase voltage of the permanent magnet synchronous motor. Then, the type of PWM output to be used is complemented PWM output (that is, PWM output with dead time).
1.4 Voltage Equation for PMSM Voltages in dq Coordinate System (Reference Information)

1.4.1 Voltage Equation

The voltage equation for the PMSM voltages in the dq coordinate system is shown below. This voltage equation can be used to convert currents Id and Iq into voltages Vd and Vq. Note, however, that this sample program uses PI control for current conversion into voltages. For details about PI control, see Section 5.6.4.1, PI Control.

\[
\begin{bmatrix}
  v_d \\
  v_q
\end{bmatrix} = \begin{bmatrix}
  Ra + pLd & -\omega Lq \\
  \omega Ld & Ra + pLq
\end{bmatrix} \begin{bmatrix}
  id \\
  iq
\end{bmatrix} + \begin{bmatrix}
  0 \\
  \omega \varphi_a
\end{bmatrix}
\]

\[ p = \frac{d}{dt} \]

Remarks:
- \( v_d, v_q \): Armature voltage in individual phase
- \( Ra \): Armature resistance in individual phase
- \( Ld, Lq \): Self-inductance in individual phase
- \( id, iq \): Armature current in individual phase
- \( \omega \): Angular speed of motor
- \( \varphi \): Flux of permanent magnet
- \( \varphi_a = \frac{\sqrt{2}}{3} \varphi \)
- \( p \): Differential operator

For details about the process of deriving this equation, see a relevant technical document.

This equation is not used for actual control (but is only used to derive the equation for non-interference control from the voltage equation).

1.4.2 Non-Interference Control

Laplace transformation transforms the motor voltage equation shown in Section 1.4.1 into the following expressions:

\[
Id = \frac{1}{Ra + sLd} (Vd + \omega Lq iq)
\]

\[
Iq = \frac{1}{Ra + sLq} \left( Vq - \omega \left( Ld id + \frac{2}{3} \varphi \right) \right)
\]

Remarks: \( s \): Laplace operator

These expressions show that the expression for the d axis includes some information on the q axis and the expression for the q axis includes some information on the d axis. Non-interference control refers to the control method in which said information is removed beforehand.

To perform non-interference control, you need to know motor parameters in advance. Note that the Non-interference control is a type of feed forward control.

Actual current control uses PI control. If, however, another factor affects the PI control while currents (Id and Iq) are not controlled constantly, adjustment of the PI control is difficult. Non-interference control is used to avoid such a problem.
2. Specifications

This application note describes motor control that uses the 12-bit A/D converter (ADCC), R/D converter (RDC3A), enhanced motor control unit (EMU3), motor control timer (TSG3), and peripheral interface connection (PIC).

The A/D converter measures the motor current, and the R/D converter obtains motor angle information. The EMU3 performs vector control according to the motor current value and angle information, and then generates a PWM compare value. Based on the PWM compare value, the TSG3 generates and outputs a sinusoidal wave and a PWM waveform by using the rectangular wave comparison method.

In this application note, the CPU only sets current command values and other control is performed by hardware units alone without intervention by the CPU. The user, however, can insert user's original processing into the control processing by EMU3.

The RH850/C1M-A2 supports 2-motor control.

Table 2.1 lists the specifications of motor control. Table 2.2 lists settings of individual modules. Figure 2.1 shows the configuration of a 2-motor control system (using the RH850/C1M-A2). Figure 2.2 shows the flow of fully automated control processes. Figure 2.3 shows the flow of control processes involving processing by the CPU.

### Table 2.1 Motor Control Specifications

<table>
<thead>
<tr>
<th>Item</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output waveform</td>
<td>Complemented 3-phase PWM waveform</td>
</tr>
<tr>
<td>Carrier frequency</td>
<td>10 kHz (100 µsec/cycle)</td>
</tr>
<tr>
<td>Control method</td>
<td>180-degree excitation drive method</td>
</tr>
<tr>
<td>Active level</td>
<td>Active high</td>
</tr>
<tr>
<td>Short-circuit prevention time (dead time)</td>
<td>4 µsec</td>
</tr>
<tr>
<td>Timing of updating compare register and carrier frequency</td>
<td>Trough of the carrier wave</td>
</tr>
<tr>
<td>Timing of starting A/D conversion</td>
<td>Trough of the carrier wave</td>
</tr>
<tr>
<td>Interrupt</td>
<td>Not used</td>
</tr>
<tr>
<td>Motor current value</td>
<td>The ADCC0 obtains values of phase-U and phase-V currents.</td>
</tr>
<tr>
<td>Motor angle information</td>
<td>The RDC3A obtains angle information from resolver.</td>
</tr>
<tr>
<td>Peripheral Function</td>
<td>Setting</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| RDC3A               | • Using the 10-kHz excitation signal generated in the RDC3A  
|                     | • Automatic setting of PI compensator band  
|                     | • BIST and self-diagnosis are not executed  
|                     | • Angle conversion mode 1 is selected |
| PIC                 | • Using the signal from EMU0 as a trigger for ADCC0 scan group 4 |
| ADCC                | • A/D conversion of the virtual channel 0 (phase V), virtual channel 1 (phase W) and virtual channel 2 (phase U) in scan group 4 (SG4)  
|                     | • T&H A/D conversion mode  
|                     | • One scanning operation per trigger in multi-cycle scan mode  
|                     | • Enabling the input of trigger for the hardware in scan group 4  
|                     | • Disabling the output of end interrupt signal for scan group 4  
|                     | • Transferring A/D conversion results to the EMU3 |
| TSG3                | • HT-PWM mode  
|                     | • Carrier frequency: 100 µsec  
|                     | • Dead time: 4 µsec  
|                     | • Direct transfer of Carrier frequency and PWM duty settings from the EMU3 |
| EMU3                | • Starting the input IP automatically with AD end trigger, and starting the IP in other computation units automatically at the end of operation of preceding IP  
|                     | • Using the result of computation by preceding IP for computation  
|                     | • Using the current value from the ADCC, angle value and angular velocity value from the RDC3A for computation  
|                     | • Starting A/D conversion in the timing of a trough of the carrier wave  
|                     | • No delay in trigger output for A/D conversion and R/D conversion  
|                     | • Selectable non-interference control/voltage compensation |
Figure 2.1 Configuration of 2-Motor Control System (using the RH850/C1M-A2)
Figure 2.2 Flow of Fully Automated Control
Figure 2.3 Flow of Control Involving CPU Processing
3. Operation Check Conditions

The sample code described in this application note has been checked for normal operation under the following conditions:

Table 3.1 Operation Check Conditions

<table>
<thead>
<tr>
<th>Item</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microcontroller</td>
<td>RH850/C1M-A2</td>
</tr>
<tr>
<td>Operating frequency</td>
<td>- Main OSC: 20 MHz&lt;br&gt;- PLL0: C1M-A2 320 MHz&lt;br&gt;- PLL1: 80 MHz&lt;br&gt;- CPU clock: 320 MHz&lt;br&gt;Unmodulated high-speed peripheral clock: 80 MHz&lt;br&gt;Unmodulated low-speed peripheral clock: 40 MHz&lt;br&gt;Motor control H/W accelerator clock: 160MHz</td>
</tr>
<tr>
<td>Operating voltage</td>
<td>VDD = 1.25 V&lt;br&gt;PVCC = 5.0 V</td>
</tr>
<tr>
<td>Integrated development environment</td>
<td>Renesas Electronics' CS+ E4.01.00</td>
</tr>
<tr>
<td>C compiler</td>
<td>Renesas Electronics' C Compiler Package for RH850 Family V1.04.00</td>
</tr>
<tr>
<td></td>
<td>Compilation options&lt;br&gt;Default settings of the integrated development environment</td>
</tr>
<tr>
<td>Sample code version</td>
<td>Version 1.10</td>
</tr>
</tbody>
</table>
4. Pins

Table 4.1 lists the pins to be used and their functions.

<table>
<thead>
<tr>
<th>Module</th>
<th>Pin</th>
<th>Input/Output</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADCC</td>
<td>ADCC0I00</td>
<td>Input</td>
<td>Input of measured phase-V current</td>
</tr>
<tr>
<td></td>
<td>ADCC0I01</td>
<td>Input</td>
<td>Input of measured phase-W current</td>
</tr>
<tr>
<td></td>
<td>ADCC0I02</td>
<td>Input</td>
<td>Input of measured phase-U current</td>
</tr>
<tr>
<td>RDC3A</td>
<td>RDC3A0S1</td>
<td>Input</td>
<td>Input of resolver signal (cosθ)</td>
</tr>
<tr>
<td></td>
<td>RDC3A0S3</td>
<td>Input</td>
<td>Input of resolver signal (cosθ)</td>
</tr>
<tr>
<td></td>
<td>RDC3A0S2</td>
<td>Input</td>
<td>Input of resolver signal (sinθ)</td>
</tr>
<tr>
<td></td>
<td>RDC3A0S4</td>
<td>Input</td>
<td>Input of resolver signal (sinθ)</td>
</tr>
<tr>
<td></td>
<td>RDC3A0RSO</td>
<td>Output</td>
<td>Output of excitation signal</td>
</tr>
<tr>
<td></td>
<td>RDC3A0COM</td>
<td>Output</td>
<td>Output of common voltage for excitation signal</td>
</tr>
<tr>
<td>TSG3</td>
<td>TSG30O1</td>
<td>Output</td>
<td>Output of upper-armature phase-U signal</td>
</tr>
<tr>
<td></td>
<td>TSG30O2</td>
<td>Output</td>
<td>Output of lower-armature phase-U signal</td>
</tr>
<tr>
<td></td>
<td>TSG30O3</td>
<td>Output</td>
<td>Output of upper-armature phase-V signal</td>
</tr>
<tr>
<td></td>
<td>TSG30O4</td>
<td>Output</td>
<td>Output of lower-armature phase-V signal</td>
</tr>
<tr>
<td></td>
<td>TSG30O5</td>
<td>Output</td>
<td>Output of upper-armature phase-W signal</td>
</tr>
<tr>
<td></td>
<td>TSG30O6</td>
<td>Output</td>
<td>Output of lower-armature phase-W signal</td>
</tr>
</tbody>
</table>
5. Hardware Operation

5.1 Overview of Operation

The following describes motor control operation:

1. When the EMU3 inputs a trigger signal to the A/D converter via the PIC in the timing of a trough of the carrier wave, the A/D converter performs A/D conversion of the virtual channels 0 to 3 in ADCC0 group 4 to obtain values of the phase-U and phase-V currents of the motor. At the same time, the angle generation IP of the EMU3 computation unit converts the resolver angle obtained from the RDC3A into the electric angle of the motor, and transfers the electric angle to the input IP of the computation unit.

2. After the A/D conversion, conversion results are stored in registers of the EMU3, and the input IP starts automatically. The input IP performs computation, including dq conversion, based on the motor current values and electric angle to calculate the feedback values of d-axis and q-axis currents.

3. When the input IP ends computation, the PI control IP of the computation unit starts automatically. The PI control IP performs PI control based on the feedback values of d-axis and q-axis currents and the command values set by software, and calculates d-axis and q-axis voltages.

4. When the PI control IP ends computation, the PWM IP of the computation unit starts automatically. The PWM IP calculates the PWM compare values (duty values) for phases U, V, and W based on d-axis and q-axis voltages.

   Also, the PWM IP executes non-interference control processing with angular velocity from RDC3A.

5. When the PWM compare values are transferred to the TSG3, and set in the compare registers of the TSG3 in the timing of the next reloading. Then, a 3-phase PWM waveform is output from corresponding TSG3 pins.

6. When evaluation board dependent processing is enabled by macrodef.h, current command values are set for EMU3 in main function. Thereafter, a rotation command is given to the motor to be controlled.

5.2 R/D Converter 3A (RDC3A)

The RDC3A converts the analog signal output from the resolver into digital signals, and calculates an angle value (resolver angle value) and angular velocity.

5.3 Peripheral Interconnection (PIC)

The PIC generates an ADCC hardware trigger signal based on a signal that the EMU3 outputs in synchronization with the carrier wave.

5.4 A/D Converter (ADCC)

The ADCC performs A/D conversion of a motor current value, and inputs the converted value to the EMU3. In the case of the motor control described in this application note, A/D conversion starts with a trigger that the PIC generates in response to a signal output from the EMU3. When the trigger signal is input, the ADCC converts the phase-V current value input from ADCC0I00 pin, phase-W current value input from ADCC0I01 pin and phase-U current value input from ADCC0I02 pin. The phase-W current value is not used in the process.

5.5 Motor Control Timer 3 (TSG3)

The TSG3 outputs a 3-phase PWM waveform based on the compare value transferred from the EMU3. TSG3 generates duty from compare match of 18-bit counter and 18-bit sub counter (but uses 16-bit for combination with EMU3) and outputs 3-phase PWM waveform with dead time setting.

Figure 5.1 shows a timing chart for the 3-phase PWM waveform.
Figure 5.1 Timing Chart for 3-Phase PWM Waveform
5.6 Enhanced Motor Control Unit 3 (EMU3)

5.6.1 EMU3 Common Unit

The EMU3 performs vector control based on motor current values and angle information, and generates PWM compare values. The motor control described in this application note is performed by hardware units alone without intervention by the CPU except the current command setting processing.

[Settings of related registers]

- EMUST bit in EMU3nCTR register = 1 (Starting EMU3n)
- INIPTRG[1:0] bits in EMU3nIPTRG register = 10B (Starting input IP at the end of A/D conversion)
- PIIPTRG bit in EMU3nIPTRG register = 1 (Starting PI control IP at the end of input IP operation)
- PWMIPTRG bit in EMU3nIPTRG register = 1 (Starting PWM IP at the end of PI control IP operation)
- CAVALAD bit in EMU3nADTRG register = 1 (Starting A/D converter in the timing of carrier trough)
- ADDATA bit in EMU3nDDCNT register = 0 (No delay in A/D conversion)
- RDDATA bit in EMU3nDDCNT register = 0 (No delay in the output of R/D conversion trigger)
- EMU3nINT0 to EMU3nINT4 registers = 0000H (Disabling interrupt)
- INTC2EIC56 to EIC60 registers = 00CFH (Disabling interrupt processing)
5.6.2 Angle Generation IP

The angle generation IP starts each time the angle data value (high-order 16 bits in the RDC3AnENC1 register) input from the RDC3A changes. After adding an offset value (EMU3nANGOFS register) to the value of the RDC3AnENC1 register, the angle generation IP generates an electric angle. The angle generation IP stores the generated electric angle in the EMU3nTHTEFIX register. Figure 5.2 shows a block diagram for the angle generation IP.

[Settings of related registers]
- EMU3nANGCTR register = 00H (Inputting angle data and phase-Z pulse from the RDC3A)
- EMU3nRESRLD register = 01H (Specifying "pole number of resolver - 1")

Example: When the resolver pole number is 2, the value of EMU3nRESRLD register is 01H.
- EMU3nANGOFS register = 0000H (Specifying the angle offset value)
- EMU3nPXR register = 0100H (Resolver angle: electric angle = 1 : 1)

Example: When the ratio of resolver angle to electric angle is 1 : 2, the value of EMU3nPXR register is 0200H.

![Figure 5.2 Block Diagram for Angle Generation IP](image)

5.6.3 Input IP

With "B'10" set in the INIPTRG[1:0] bits in the EMU3nIPTRG register, the input IP starts automatically when A/D conversion ends. In the motor control described in this application note, the input IP calculates the current value of the remaining phase from the 2-phase motor current value, and then performs dq conversion on the basis of the calculated current value and the electric angle. The value stored in the EMU3nTHTEFIX register is transferred to the input IP when an A/D trigger occurs, and then stored in the EMU3nTHTE register. Figure 5.3 shows a block diagram for the input IP.

![Figure 5.3 Block Diagram for Input IP](image)
5.6.3.1 Calculation of Motor Current Values

The input IP calculates 3-phase current values from the 2-phase current values stored in the EMU3nAD0 and EMU3nAD2 registers. Calculation results are stored in the EMU3nIVFIX, EMU3nWFIX, and EMU3nUFIX registers. When values are set in the offset compensation registers (EMU3nAD0OFS and EMU3nAD2OFS) and LSB adjustment register (EMU3nDIVLSB) for A/D-converted values, the input IP performs computation according to the set values.

[Settings of related registers]
- EMU3nAD0OFS register = (offset value to be applied when motor current is 0)
- EMU3nAD2OFS register = (offset value to be applied when motor current is 0)
- EMU3nDIVLSB register = 0001 0000H
- CMES bit in EMU3nCTRINMD register = 1 (Selection by CMUVW[2:0] bits)
- CMUVW[2:0] bits in EMU3nCTRINMD register = 100B (Measuring two phases [phases U and V])

[Computation by input IP]

The input IP performs offset compensation for the A/D-converted value in the range from 0 to 4095, and then converts the value into a value from -2048 to 2047. Figure 5.4 shows the offset compensation for the A/D conversion result.

![Offset compensation for A/D conversion result](image)

**Figure 5.4 Offset Compensation for A/D Conversion Result**

\[
\begin{align*}
EMU3nIUFIX & \leftarrow \left(EMU3nAD0FIX - EMU3nAD0OFS \right) \times EMU3nDIVLSB \gg 16 \\
EMU3nIVFIX & \leftarrow \left(EMU3nAD1FIX - EMU3nAD1OFS \right) \times EMU3nDIVLSB \gg 16 \\
EMU3nWFIX & \leftarrow -\left(EMU3nUFIX + EMU3nVFIX \right)
\end{align*}
\]

5.6.3.2 dq Conversion

The input IP performs d-axis/q-axis current conversion by using the phase-U current, phase-V, and phase-W current values stored in the EMU3nUFIX, EMU3nVFIX, and EMU3nWFIX registers and the electric angle value set in the EMU3nTHTE register. As the result of conversion, a d-axis current value and a q-axis current value are stored in the EMU3nDFIX and EMU3nQFIX registers, respectively.

[Settings of related registers]
- FREGIN bit in EMU3nCTRINMD register = 1 (Using the electric angle generated by the angle generation IP)
- EMU3nSR2 register = 0000 D106H (Coefficient of d/q-axis current conversion)

[Computation by input IP]

\[
\begin{bmatrix}
EMU3nIDFIX \\
EMU3nIQFIX
\end{bmatrix} = EMU3nSR2 \times \begin{bmatrix}
\sin(\theta e + 90^\circ) & -\sin(\theta e + 150^\circ) & -\sin(\theta e + 30^\circ) \\
-\sin(\theta e + 0^\circ) & \sin(\theta e + 60^\circ) & -\sin(\theta e + 120^\circ)
\end{bmatrix} \begin{bmatrix}
EMU3nIUFIX \\
EMU3nIVFIX \\
EMU3nWFIX
\end{bmatrix}
\]
5.6.4 PI Control IP

With "1" set in the PIIPTRG bit in the EMU3nIPTTRG register, the PI control IP starts automatically when the input IP ends processing. The PI control IP calculates d-axis and q-axis voltages from the feedback values of d-axis and q-axis currents calculated by dq conversion and the command values of d-axis and q-axis currents set by software. Figure 5.5 shows a block diagram for the PI control IP.

Figure 5.5 Block Diagram for PI control IP

5.6.4.1 PI Control

The PI control IP calculates d-axis and q-axis voltages, which are operating quantities, from the feedback values stored in the EMU3nIDFIX and EMU3nIQFIX registers and the command values set in the EMU3nIDIN and EMU3nIQIN registers. As the result, the d-axis and q-axis voltages are stored in the EMU3nVD and EMU3nVQ registers, respectively.

[Settings of related registers]
- FSUMIQ bit in EMU3nPICTR register = 1 (Using the result of computation by EMU as the q-axis integral term)
- FSUMID bit in EMU3nPICTR register = 1 (Using the result of computation by EMU as the d-axis integral term)
- EMU3nIDIN register: Setting of command value for the d axis
- EMU3nIQIN register: Setting of command value for the q axis
  - EMU3nGPD0 register = 0001 0000H (The d-axis proportional gain is 1.)
  - EMU3nGPQ0 register = 0001 0000H (The q-axis proportional gain is 1.)
  - EMU3nGID register = 0000 0100H (The d-axis integration gain is 1/256.)
  - EMU3nGIQ register = 0000 0100H (The q-axis integration gain is 1/256.)
  - EMU3nGPD register = 0001 0000H (The d-axis proportional gain is 1.)
  - EMU3nGPQ register = 0001 0000H (The q-axis proportional gain is 1.)
• EMU3nGIDMAX register = 0000 0800H (Maximum value of d-axis integral term)
• EMU3nGiQMAX register = 0000 0800H (Maximum value of q-axis integral term)
• EMU3nVDMAX register = 7FFF FFFFH (Maximum value of d-axis operating quantity)
• EMU3nVQMAX register = 7FFF FFFFH (Maximum value of q-axis operating quantity)

5.6.5 PWM IP

With “1” set in the PWMIPTRG bit in the EMU3nIPTRG register, the PWM IP starts automatically when the PI control IP ends processing. The PWM IP calculates phase-U output, phase-V output, and phase-W output voltages from the d-axis and q-axis voltages calculated by the PI control IP. The PWM IP then calculates the duty ratio and value of compare register for each phase of PWM waveform output. Figure 5.6 shows a block diagram for the PWM IP. When the SETDEC bit of EMU3nPWMCTR is 1, non-interference control is executed on the d-axis and q-axis voltages. When the SETDEC bit of EMU3nPWMCTR is 0, voltage compensation is executed.

![Block Diagram for PWM IP](image)

Figure 5.6 Block Diagram for PWM IP

5.6.5.1 Compensation of d-Axis and q-Axis Voltages/Non-Interference Control

You can either compensate or execute non-interference compensation on the d-axis and q-axis voltages calculated by the PI control IP. The motor control described in this application note can switch by #define setting of macrodef.h. Set compensation values according to control requirements. When executing axis voltage compensation/non-interference control, write “1” to the FPWMREFPER bit in the EMU3nREFCTR register to reflect the value of the registers in the internal circuit of the EMU.

Note that the maximum value of the d-axis and q-axis voltages must be set no matter of setting of FPWMREFPER bit of EMU3nREFCTR register.
[Settings of related registers]

- EMU3nVD2MAX register = 7FFF FFFFH
- EMU3nVQ2MAX register = 7FFF FFFFH

- Voltage compensation (Selects USE_EMU_NINF_CTR OFF via macrodefh.h)
  Compensate by EMU3nVDCRCT register and EMU3nVQCRCT register. Set an appropriate value to each parameter depending on control. In this sample program, correction amount is 0 (no compensation) as the value is set to 0.

[Settings of related registers]
- EMU3nVDCRCT register = 0000 0000H (No compensation)
- EMU3nVQCRCT register = 0000 0000H (No compensation)

Also, by setting "0" in the EMU3nGPD0 and EMU3nGPQ0 registers, you can set parameters for 3-phase voltage conversion by using the EMU3nVDCRCT and EMU3nVQCRCT registers without using PI control.

\[ v_{di} \leftarrow 0 + EMU3nVDCRCT \]
\[ v_{qi} \leftarrow 0 + EMU3nVQCRCT \]

- Non-interference control (Selects USE_EMU_NINF_CTR ON via macrodefh.h)
  The d-axis and q-axis voltages are corrected by the following expression by using EMU3nDECVELG register, EMU3nDECFLUX register, EMU3nDECLD register, EMU3nDECLQ register, angular speed obtained from RDC3A and the d-axis and q-axis currents.
  Set an appropriate value to each parameter depending on control. In this sample program, correction amount is 0 (no compensation) as the value is set to 0.

\[ d-\text{axis voltage} \leftarrow d-\text{axis voltage} - (\omega_e \times G_{\omega_e} \times L_q \times I_q) \]
\[ q-\text{axis voltage} \leftarrow q-\text{axis voltage} + (\omega_e \times G_{\omega_e} \times L_d \times I_d + \omega_e \times G_{\omega_e} \times \Phi) \]

(\(\omega_e\): angular speed of electric angle, \(\Phi\): flux, I: current, L: inductance, \(G_{\omega_e}\): angular speed gain)

[Settings of related registers]
- EMU3nDECVELG register = 0000 0000H
- EMU3nDECFLUX register = 0000 0000H
- EMU3nDECLD register = 0000 0000H
- EMU3nDECLQ register = 0000 0000H
5.6.5.2 Conversion of 3-Phase Voltages

The PWM IP performs conversion of 3-phase voltages by using the values of the EMU3nVD and EMU3nVQ registers calculated by the PI control IP and the electric angle value stored in the EMU3nTHTE register so as to obtain phase-U and phase-W voltages. The PWM IP can also adjust the electric angle by using the d-axis reference voltage register (EMU3nPHI) and electric angle adjustment register (EMU3nGTHT) in consideration of the delay in PWM waveform output after motor current measurement. Note that the motor control described in this application note excludes this adjustment.

[Settings of related registers]
- FLININIP bit in EMU3nPWMCTR register = 1 (Using the result of computation by EMU as the electric angle)
- SETVEL bit in EMU3nPWMCTR register = 1 (Using the RDC speed)
- SETDEC bit in EMU3nPWMCTR register = 1 (When selected non-interference control)
- EMU3nSR23 register = 0000 D106H (Coefficient of 3-phase voltage conversion)
- EMU3nPHI register = 0000H (No adjustment of the phase of electric angle)
- EMU3nGTHT register = 0100H (No adjustment of the phase of electric angle)

[Computation by PWM IP]
\[
\begin{align*}
\begin{bmatrix}
\nu_u0 \\
\nu_w0
\end{bmatrix} &= EMU3nSR23 \times \begin{bmatrix}
\cos(\theta + 0^\circ) & -\sin(\theta + 0^\circ) \\
\cos(\theta + 120^\circ) & -\sin(\theta + 120^\circ)
\end{bmatrix} \begin{bmatrix}
\nu_d1 \\
\nu_q1
\end{bmatrix} \\
\nu_v0 &= -(\nu_u0 + \nu_w0)
\end{align*}
\]

5.6.5.3 Calculation of Duty Ratios

The PWM IP calculates duty ratios of 3-phase PWM waveform from the phase-U voltage and phase-W voltage values generated by 3-phase voltage conversion and the values of the input voltage register (EMU3nVOLV) and digit alignment register (EMU3nPWMK1). The PWM IP adds offset values to the calculated duty ratios, and performs maximum/minimum control processing. The results of processing are stored in the phase-U output voltage register (EMU3nVUFIX), phase-V output voltage register (EMU3nVVFIX), and phase-W output voltage register (EMU3nVWFIX) for duty ratio calculation. The motor control described in this application note excludes offset addition. Each modulation wave is mixed into \(\nu_u, \nu_w, \text{ and } \nu_v\) depending on the setting of SETHARM bit of EMU3nPWMCTR register. The motor control described in this application note excludes modulation.

[Settings of related registers]
- EMU3nVUOFS register = 0000H (Offset is not added to 3-phase duty ratio.)
- EMU3nVVOFS register = 0000H (Offset is not added to 3-phase duty ratio.)
- EMU3nVWOFS register = 0000H (Offset is not added to 3-phase duty ratio.)
- EMU3nVOLV register = 1000H: To be adjusted according to the system voltage
- EMU3nPWMK1 register = 0080 0000H (Adjustment of the range of duty ratios)
- EMU3nDTUL register = 7FFF FFFFH (Setting of the upper limit of duty ratio)
- EMU3nDTLL register = 8000 0000H (Setting of the lower limit of duty ratio)
[Computation by PWM IP]

\[
EMU3nVUFIX = vuo \times EMU3nPWMK1 \times \frac{1}{EMU3nVOLV} + EMU3nVUFIX
\]

\[
EMU3nVUFIX = vuo \times EMU3nPWMK1 \times \frac{1}{EMU3nVOLV} + EMU3nVUFIX
\]

\[
EMU3nVUFIX = vuo \times EMU3nPWMK1 \times \frac{1}{EMU3nVOLV} + EMU3nVUFIX
\]
5.6.5.4 Calculation of Compare Values

The PWM IP calculates compare values for phases U, V, and W from the values of the EMU3nVUFIX, EMU3nVVFIX, and EMU3nVWFIX registers, and stores calculation results in the EMU3nPWMUIP, EMU3nPWMVIP, and EMU3nPWMWIP registers. For computation, the PWM IP uses also the values of the carrier cycle register (EMU3nCARR), dead-time setting register (EMU3nDTT), and digit alignment register 2 (EMU3nPWMK2).

[Settings of related registers]
- SHIPWM bit in EMU3nPWMCTR register = 0 (Outputting settings without shifting)
- PWMSEL bit in EMU3nPWMCTR register = 1 (Generation from carrier cycle and dead-time values)
- SETPWM bit in EMU3nPWMCTR register = 1 (Using the result of computation by EMU as the PWM compare value)
- EMU3nCARR register = 1F40H (8000) (Carrier cycle: 100 µsec = 12.5 nsec x 8000)  
  (Output to TSG3, and use it as Carrier cycle. It should be consistent with the setting of TSG3.)
- EMU3nDTT register = 280H (640) (Dead time: 8 µsec = 12.5 nsec x 640)
- EMU3nPWMK2 register = 0100H (Adjustment of the range of PWM compare value)
- EMU3nPWMUL register = FFFFH (Setting of the upper limit of PWM compare value)
- EMU3nPWMUL register = 0000H (Setting of the lower limit of PWM compare value)
- EMU3nPWMU register = 10E0H (Phases U PWM software input compare value )  
  (EMU3nCARR+EMU3nDTT) / 2
- EMU3nPWMV register = 10E0H (Phases V PWM software input compare value )  
  (EMU3nCARR+EMU3nDTT) / 2
- EMU3nPWMW register = 10E0H (Phases W PWM software input compare value )  
  (EMU3nCARR+EMU3nDTT) / 2

[Computation by PWM IP]

The following shows the computing equation for phase U, which is applied also to phases V and W:

\[
\text{pwmu} = \left( \text{EMU3nVUFIX} \times \frac{\text{EMU3nCARR} + \text{EMU3nDTT}}{2} \times \text{EMU3nPWMK2} \right) \times \frac{1}{2^{16}} + \frac{\text{EMU3nCARR} + \text{EMU3nDTT}}{2}
\]

\[
\text{EMU3nPWMUIP} = \frac{\text{EMU3nPWMUIP}}{2} \quad (\text{EMU3nVOLV} = 0)
\]

\[
\text{EMU3nPWMUIP} = \text{EMU3nCARR} + \text{EMU3nDTT} \quad (\text{pwmu} \geq \text{EMU3nCARR} + \text{EMU3nDTT} - \text{EMU3nPWMUL})
\]

\[
\text{EMU3nPWMUIP} = 0 \quad (\text{EMU3nVOLV} = 0)
\]

\[
\text{EMU3nPWMUIP} = \text{pwmu} \quad (\text{EMU3nVOLV} = 0)
\]
Figure 5.7 Limit Processing of PWM Value and Compare Value
5.6.6 Examples of Calculation Accuracy

It is assumed that the calculation of EMU3 operates with 32 bits based on 12 bits of input data from ADCC and RDC3A and outputs it to TSG3 as 16 bits compare value.

**Figure 5.8 Relationship between Input and Output and Calculation Accuracy**

Data range: 0x000 ~ 0xFFF

-0x7FFF ~ 0x7FFF

Calculation of motor current value

-0x7FFFFFFF ~ 0x7FFFFFFF

LSB adjustment

Offset current value compensation

A/D conversion result

Data range: 0x000 ~ 0xFFF

0x000 ~ 0xFFF

-0x7FFFFFFF ~ 0x7FFFFFFF

LSB adjustment is possible

Obtaining 3-phase current value

U.V.W

Calculated motor current value

-0x7FFF ~ 0x7FFF

Compensation for electric-angle response delay

(U.V.W) d,q

LSB adjustment is possible with EMU3nSR2

Data range 0x000 ~ 0xFFF

LSB adjustment is possible with EMU3nSR2, EMU3nPWMK1, EMU3nPWMK2

-0x7FFFFFFF ~ 0x7FFFFFFF

or -0x7FFF ~ 0x7FFF

0x0000 ~ 0xFFF

-0x7FFFFFFF ~ 0x7FFFFFFF

or -0x7FFF ~ 0x7FFF
5.7 File Configuration

Table 5.1 lists the files that are used for the sample code. Note that the list of files excludes the files that are automatically generated in the integrated development environment. This sample program has been confirmed to operate with Renesas' evaluation board. The process to confirm the operation is included.

<table>
<thead>
<tr>
<th>File name</th>
<th>Description</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>main1.c</td>
<td>PE1 main module</td>
<td>PE2: main2.c</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SubCPU: mainsub.c</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PE2, SubCPU main: infinite loop</td>
</tr>
<tr>
<td>rdc3a.c</td>
<td>R/D converter 3A setting module</td>
<td></td>
</tr>
<tr>
<td>pic.c</td>
<td>PIC setting module</td>
<td></td>
</tr>
<tr>
<td>adcc.c</td>
<td>A/D converter setting module</td>
<td></td>
</tr>
<tr>
<td>tsg3.c</td>
<td>TSG3 setting module</td>
<td></td>
</tr>
<tr>
<td>emu3.c</td>
<td>EMU3 setting module</td>
<td></td>
</tr>
<tr>
<td>port.c</td>
<td>Pin setting module</td>
<td></td>
</tr>
<tr>
<td>iodefine.h</td>
<td>Register header file</td>
<td></td>
</tr>
<tr>
<td>macrodef.h</td>
<td>Header file for embedded module</td>
<td>Process for evaluation, ON/OFF</td>
</tr>
<tr>
<td></td>
<td>setting</td>
<td>for EMU3 non-interference control</td>
</tr>
</tbody>
</table>

5.8 Functions

Table 5.2 lists functions.

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rdc3a_init</td>
<td>Initializes the R/D converter 3A.</td>
</tr>
<tr>
<td>pic_init</td>
<td>Initializes the PIC.</td>
</tr>
<tr>
<td>adcc_init</td>
<td>Initializes the A/D converter.</td>
</tr>
<tr>
<td>tsg3_init</td>
<td>Initializes the TSG3.</td>
</tr>
<tr>
<td>emu3_init</td>
<td>Initializes the EMU3.</td>
</tr>
<tr>
<td>port_init</td>
<td>Initializes the pins.</td>
</tr>
<tr>
<td>wait_mainloop</td>
<td>Wait processing for current command value write</td>
</tr>
<tr>
<td>set_dq_command</td>
<td>Current command value write processing</td>
</tr>
</tbody>
</table>
5.9 Specifications of Functions

This section describes the specifications of the functions used in the sample code.

---

**RDC3A_init**

Outline: Initialization of R/D converter 3A  
Header: iodefienefi.e
Declaration: void rdc3a_init(void)  
Description: This function initializes the R/D converter 3A.  
Argument: None  
Return value: None  
Remarks:

---

**pic_init**

Outline: Initialization of PIC  
Header: iodefienefi.e  
Declaration: void pic_init(void)  
Description: This function initializes the PIC.  
Argument: None  
Return value: None  
Remarks:

---

**adcc_init**

Outline: Initialization of A/D converter  
Header: iodefienefi.e, macrodefi.e  
Declaration: void adcc_init(void)  
Description: This function initializes the A/D converter.  
Argument: None  
Return value: None  
Remarks:

---

**tsg3_init**

Outline: Initialization of TSG3  
Header: iodefienefi.e  
Declaration: void tsg3_init(void)  
Description: This function initializes the TSG3.  
Argument: None  
Return value: None  
Remarks:
emu3_init
Outline: Initialization of EMU3
Header: iodefine.h, macrodef.h
Declaration: void emu3_init(void)
Description: This function initializes the EMU3.
Argument: None
Return value: None
Remarks:

port_init
Outline: Initialization of pins
Header: iodefine.h
Declaration: void port_init(void)
Description: This function initializes the pins.
Argument: None
Return value: None
Remarks:

wait_mainloop
Outline: Executes wait processing
Header: iodefine.h, macrodef.h
Declaration: void wait_mainloop (void)
Description: This function executes wait processing.
Argument: None
Return value: None
Remarks:

set_dq_command
Outline: Writes a current command value to Q-current target register of EMU30 after reading the voltage value from ADCC0 VR3.
Header: iodefine.h, macrodef.h
Declaration: void set_dq_command (void)
Description: This function writes current command value.
Argument: None
Return value: None
Remarks: Q setting (AD read value x 2) - 4096
D setting 0
### 5.10 Specifications of Functions

Figure 5.9 shows the flow of main processing. Figure 5.10 shows the flow of RDC3A initialization. Figure 5.11 shows the flow of PIC initialization. Figure 5.12 shows the flow of ADCC initialization. Figure 5.13 shows the flow of TSG3 initialization. Figure 5.14 shows the flow of EMU3 initialization (1). Figure 5.15 shows the flow of EMU3 initialization (2). Figure 5.16 shows the flow of pin setting. Figure 5.17 shows the flow of wait processing and Figure 5.18 shows the flow of write processing of current command value. Register synchronous processing is inserted in each function as necessary. However, it is omitted from the flow.

![Flowchart of PE1 Main Processing](image)

**Figure 5.9 PE1 Main Processing**
Figure 5.10 RDC3A Initialization

- rdc3a_init
  - Select the internal excitation signal
  - Select the control gain
  - Set the error detection
  - Set the excitation extraction filter
  - Set the threshold for detecting errors
  - Set the monitor output
  - Set the forced gain control
  - Set the automatic amplitude adjustment
  - Set the digital calculation
  - Select the angle calculation mode
  - Start analog operation
  - Perform initialization in the RDC3A
  - Wait 210 usec
  - Check on initialization end
  - RDC3A0DIAG1 register INIT bit = 0?
    - No
      - Execute RDC AD calibration
      - Wait 2 msec
      - Reset Ki
      - Wait 5 msec
      - Set PGA inversion
      - Return
    - Yes
      - RDC3A0ADSTD1 register ADCALST bit ← 1
      - RDC3A0DIAG1 register KIRST bit ← 1 (Executing Ki reset)
      - RDC3A0DCUR0 register PGAIVSL bit ← 0 (No PGA inversion)
**Figure 5.11 PIC Initialization**

```
pic_init
  Set A/D conversion trigger
  PIC2DADCC0TSEL4 register → 0x0002001H (Selecting EMU30 as trigger source for ADCC0 scan group4)
  Return
```

**Figure 5.12 ADCC Initialization**

```
adcc_init
  Select the analog input channel
  ADCC0VCR0 register ← 0x0002000H (T&H assignment of ADCC0I00 to virtual channel 0)
  ADCC0VCR1 register ← 0x0002001H (T&H assignment of ADCC0I01 to virtual channel 1)
  ADCC0VCR2 register ← 0x0002002H (T&H assignment of ADCC0I02 to virtual channel 2)
  ADCC0VCR3 register ← 0x0000004H (Assignment of ADCC0I03 to virtual channel 3) *
  Set virtual channels to start and end scan group
  ADCC0SGVCSP4 register ← 0x00H (Starting with virtual channel 0)
  ADCC0SGVCEP4 register ← 0x03H (Ending with virtual channel 3) *
  Set suspend mode
  ADCC0SGCR4 register ← 0x00H (Scan group 4 Multi Cycle mode setting) *
  ADCC0ADCR2 register ← 0x10H (Signed integer)
  ADCC0ADCR1 register ← 0x02H (Asynchronous suspend)
  Set T&H
  ADCC0THCR register ← 0x00H
  ADCC0THGSR register ← 0x0000H (TH group A assignment)
  ADCC0THACR register ← 0x23H (Group A setting)
  ADCC0THER register ← 0x07H (Enable TH A 0-2)
  ADCC0THACR register HLDTE bit ← 1H (Enable Group A operation)
  Enable T&H operation
  ADCC0THSMPCSTCR register SMPST bit ← 1 (Start sampling)
  Wait for 30AD Clock
  Enable A/D conversion start by hardware trigger
  ADCC0SGCR4 register TRGMD0 bit ← 1 (Enabling hardware trigger input to scan group 4)
  return
* Possible to switch macedef.h
```
**Figure 5.13 TSG3 Initialization**

1. **tsg3_init**
2. **Switch to HT-PWM mode**
   
   TSG30CONF0 register: TSG30MD[2:0] bit 001B (HT-PWM mode)

3. **Enable dead-time setting**
   
   TSG30DTPR register ← 0000H (Enable rewriting)

4. **Set dead time to 4 usec**
   
   TSG30DTC0W register ← 0000 0140H (Value of dead time at shifting from negative phase to positive phase) (*1)
   
   TSG30DTC1W register ← 0000 0140H (Value of dead time at shifting from positive phase to negative phase) (*1)

5. **Set PWM cycle to 100 usec**
   
   TSG30CMP0E register ← 0000 1F40H (Setting PWM cycle 100 usec) (*1)

6. **Set the initial compare value for phase U/V/W so that duty ratio is 50%**
   
   TSG30CMPUE register ← 0000 0FA0H (Phase-U compare value) (*1)
   
   TSG30CMPVE register ← 0000 0FA0H (Phase-V compare value) (*1)
   
   TSG30CMPWE register ← 0000 0FA0H (Phase-W compare value) (*1)

7. **Set reload conditions**
   
   TSG30CTR4 register ← 0000 0080H (Set PWM trough reload)

*1: When operating frequency is 80MHz
Figure 5.14 EMU3 Initialization (1)
**Figure 5.15 EMU3 Initialization (2)**

- **EMU3REFCTR register** ← 00H (Disable reflection of PWM IP register value)
- **EMU3PWSMCTR register** ← 000102A1H
  - SETVEL bit ← 1 (Use angular velocity RDC)
  - PWMSEL bit ← 1 (Generate from carrier cycle and dead time setting)
  - SHIPWM bit ← 0 (Output without shifting each set value)
  - FLINIP bit ← 1 (Use EMU calculation result for electric angle)
  - SETPWM bit ← 1 (Use EMU calculation result for PWM compare value)
- **EMU3PWSMCTR register** ← 000112A1H (Decoupling control enable)
- **SETDEC bit** ← 1 (Decoupling control enable)
- **EMU3DUL register** ← 7FFF FFFFH (Set the upper limit of duty ratio)
- **EMU3DTLL register** ← 8000 0000H (Set the lower limit of duty ratio)
- **EMU3PWMK1 register** ← 0080 0000H (Matching number of digits)
- **EMU3PWMK2 register** ← 0100H (Matching number of digits)
- **EMU3DPR3 register** ← 0000 D106H (3-phase voltage conversion coefficient)
- **EMU3DPHI register** ← 0000H (No adjustment of the phase of electric angle)
- **EMU3DGTH register** ← 0100H (No adjustment of the phase of electric angle)
- **EMU30VUSFS register** ← 0000H (Offset is not a added to phase-U duty ratio)
- **EMU30VUSFOFS register** ← 0000H (Offset is not added to phase-V duty ratio)
- **EMU30VUSFOFS register** ← 0000H (Offset is not added to phase-W duty ratio)
- **EMU30Vально register** ← 1000H (Setting of a value matching the system)
- **EMU30PWMW register** ← 1040H (Phase U PWM software input compare value)
- **EMU30PWMV register** ← 1040H (Phase V PWM software input compare value)
- **EMU30PWMW register** ← 1040H (Phase W PWM software input compare value)
- **EMU30CARR register** ← 1F40H (Carrier cycle: 100usec) (*1)
- **EMU30DUL register** ← 0280H (Dead time: Busec) (*1)
- **EMU30PWMUL register** ← 7FFF FFFFH (Setting of upper limit of PWM compare value)
- **EMU30PWMLL register** ← 0000H (Setting of lower limit of PWM compare value)
- **EMU30VD2MAX register** ← 7FFF FFFFH (Maximum value of d-axis operating quantity)
- **EMU30VD2MAX register** ← 7FFF FFFFH (Maximum value of q-axis operating quantity)
- **EMU30DTH register** ← 7FFF FFFFH
- **EMU30TOP register** ← 0000H
- **EMU30DONV register** ← 0000H
- **EMU30VDCVFL register** ← 00000000H (d-axis voltage correction value)
- **EMU30VDCVFL register** ← 00000000H (q-axis voltage correction value)
- **EMU30DCVELG register** ← 00000000H (Decoupling control velocity gain)
- **EMU30DECFLUX register** ← 00000000H (Decoupling control flux)
- **EMU30DECLD register** ← 00000000H (d-axis L decoupling control)
- **EMU30DCL register** ← 00000000H (q-axis L decoupling control)
- **EMU30REFCTR register** ← 01H (Enable reflection of PWM IP register value)

*1: Setting TSG operating frequency 80MHz
Figure 5.16 Pin Setting

Figure 5.17 Wait Processing

Figure 5.18 Current Command Value Write Processing
6. Reference Documents

C-Compiler Manuals
CS+ V4.00.00 Integrated Development Environment User's Manual: RH850 Coding
CS+ V4.00.00 Integrated Development Environment User's Manual: RH850 Build
RH850/C1M-A1 C1M-A2 User's Manual Hardware

(Please download the latest versions from the Renesas Electronics' web site.)
## Revision History

<table>
<thead>
<tr>
<th>Rev.</th>
<th>Date</th>
<th>Description</th>
<th>Page</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.10</td>
<td>2017.05.31</td>
<td>-</td>
<td>-</td>
<td>First edition</td>
</tr>
<tr>
<td>0.11</td>
<td>2017.06.09</td>
<td>16</td>
<td>16</td>
<td>Figure 5.1 corrected</td>
</tr>
<tr>
<td>1.00</td>
<td>2018.01.31</td>
<td>7</td>
<td>7</td>
<td>1.4.1 Voltage Equation, expressions corrected</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7</td>
<td>7</td>
<td>1.4.2 Non-Interference Control, expressions corrected</td>
</tr>
<tr>
<td></td>
<td></td>
<td>23</td>
<td>23</td>
<td>5.6.5.3 Calculation of Duty Ratios</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Description added, expressions corrected</td>
</tr>
<tr>
<td></td>
<td></td>
<td>25</td>
<td>25</td>
<td>5.6.5.4 Calculation of Compare Values, Figure 5.7 corrected</td>
</tr>
<tr>
<td></td>
<td></td>
<td>27</td>
<td>27</td>
<td>5.6.6 Examples of Calculation Accuracy added</td>
</tr>
<tr>
<td>1.01</td>
<td>2018.05.31</td>
<td>17</td>
<td>17</td>
<td>5.6.1 Fix EI Level Interrupt Control Register notation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>23</td>
<td>23</td>
<td>5.6.5.2 Delete descriptions of bits removed from User’s Manual</td>
</tr>
<tr>
<td></td>
<td></td>
<td>25</td>
<td>25</td>
<td>5.6.5.4 Clerical corrections</td>
</tr>
<tr>
<td></td>
<td></td>
<td>25</td>
<td>25</td>
<td>5.6.5.4 Change explanation of register</td>
</tr>
<tr>
<td></td>
<td></td>
<td>25</td>
<td>25</td>
<td>5.6.5.4 Add each phase software input compare value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>31</td>
<td>31</td>
<td>Figure 5.9 Delete “See other application notes”</td>
</tr>
<tr>
<td></td>
<td></td>
<td>33</td>
<td>33</td>
<td>Figure 5.12 Add ADCC0SGCR4 register setting</td>
</tr>
<tr>
<td></td>
<td></td>
<td>33</td>
<td>33</td>
<td>Figure 5.12 Fix A/D conversion wait time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>34</td>
<td>34</td>
<td>Figure 5.14 Add SUMID and SUMIQ register setting</td>
</tr>
<tr>
<td></td>
<td></td>
<td>35</td>
<td>35</td>
<td>Figure 5.15 Delete descriptions of bits removed from User’s Manual</td>
</tr>
<tr>
<td></td>
<td></td>
<td>35</td>
<td>35</td>
<td>Figure 5.15 Add PWMCTR setting of Non-Interference Control</td>
</tr>
<tr>
<td></td>
<td></td>
<td>35</td>
<td>35</td>
<td>Figure 5.15 Add each phase software input compare value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>36</td>
<td>36</td>
<td>Figure 5.18 Add comment to conditional branch</td>
</tr>
<tr>
<td>1.10</td>
<td>2018.09.07</td>
<td>9</td>
<td>9</td>
<td>Table 2.2 Add function description in RDC3A setting</td>
</tr>
<tr>
<td></td>
<td></td>
<td>13</td>
<td>13</td>
<td>Table 3.1 Update sample code version to 1.10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>32</td>
<td>32</td>
<td>Figure 5.10 Correct flow in accordance with User’s Manual</td>
</tr>
<tr>
<td></td>
<td></td>
<td>33</td>
<td>33</td>
<td>Figure 5.12 Correct flow in accordance with User’s Manual</td>
</tr>
<tr>
<td>1.11</td>
<td>2021.09.09</td>
<td>-</td>
<td>-</td>
<td>Format update</td>
</tr>
<tr>
<td>1.20</td>
<td>2022.07.11</td>
<td>26</td>
<td>26</td>
<td>Fixed EMU3nDTT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>27</td>
<td>27</td>
<td>Fixed EMU3nDTT in Figure 5.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>37</td>
<td>37</td>
<td>Fixed EMU3nDTT value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4-6</td>
<td>4-6</td>
<td>Correction of figures</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20</td>
<td>20</td>
<td>Figure 5.3 Fix Block Diagram for Input IP notation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>21</td>
<td>21</td>
<td>5.6.3.1 Fix Calculation of Motor Current Values notation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>29</td>
<td>29</td>
<td>Figure 5.8 Fix Relationship between Input and Output and Calculation Accuracy notation</td>
</tr>
</tbody>
</table>
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.
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 or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.

2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving 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, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.

3. You are solely responsible for determining whether what is required are from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.

4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.

5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alterations, conditions are within the ranges specified by Renesas Electronics products.

6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended 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; industrial robots; etc.

“High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or 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 damage (space systems; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.

7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to, unauthorized access to or use of a Renesas Electronics product or any system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INNUVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.

8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.

9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation, and any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.

10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.

11. Renesas Electronics products and technologies shall 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 shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.

12. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.

13. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

Corporate Headquarters
TOYOSU FORESIA, 3-2-24 Toyosu,
Keto-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
Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

© 2021 Renesas Electronics Corporation. All rights reserved.