RH850/U2B6

Controlling the Permanent Magnet Synchronous Motor with Resolver by Using RDC3AL and EMU3S

Introduction

This application note describes how to use the RH850/U2B6 Group microcontroller to control the permanent magnet synchronous motor equipped with a resolver by using the RDC3AL and EMU3S.

Target Device

This application note applies to RH850/U2B6.

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

This application note describes how to use the motor control timer (TSG3) of RH850/U2B6.

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.

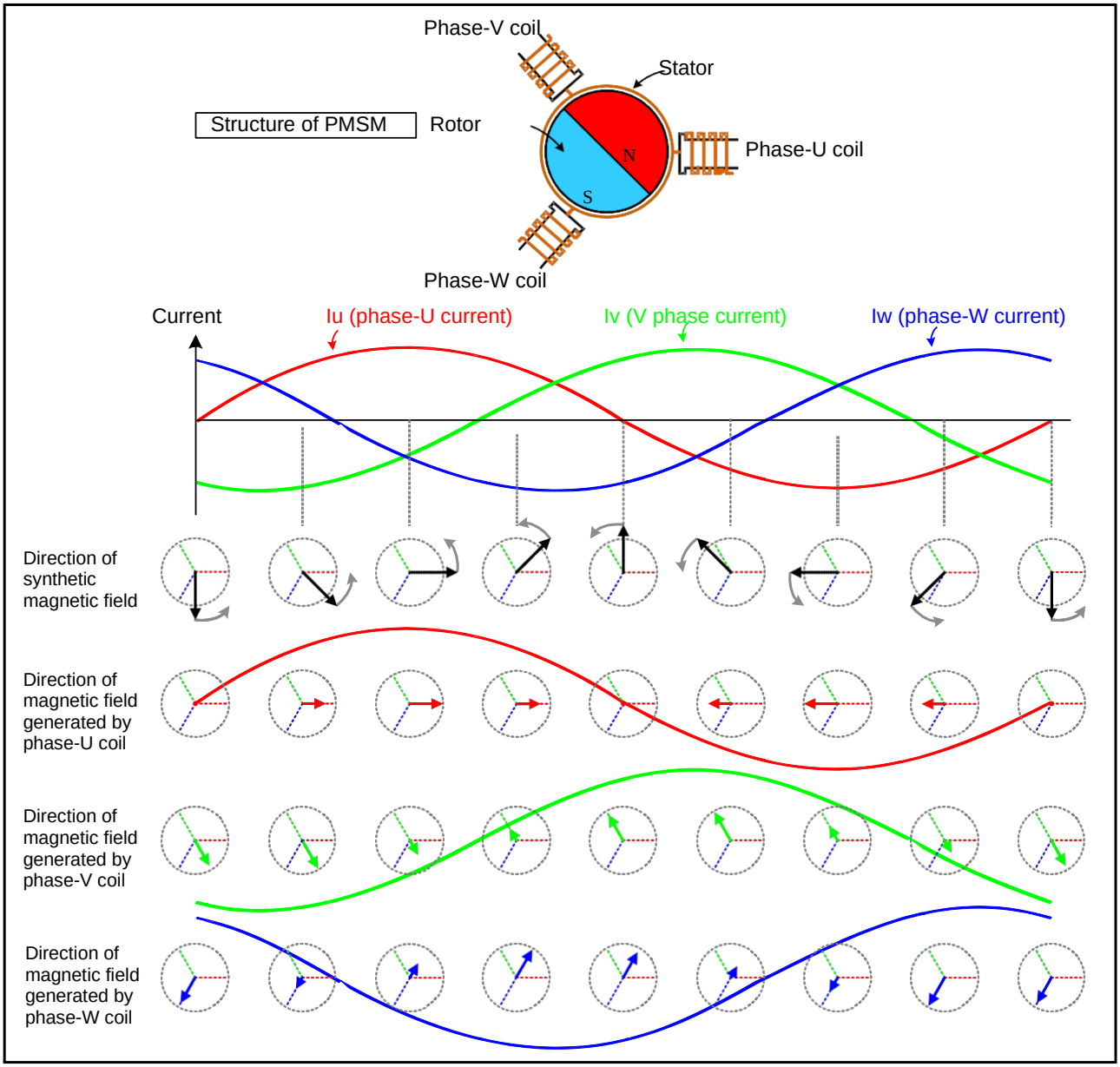


Figure 1.1 PMSM Structure and Synthetic Magnetic Field

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 90o 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.

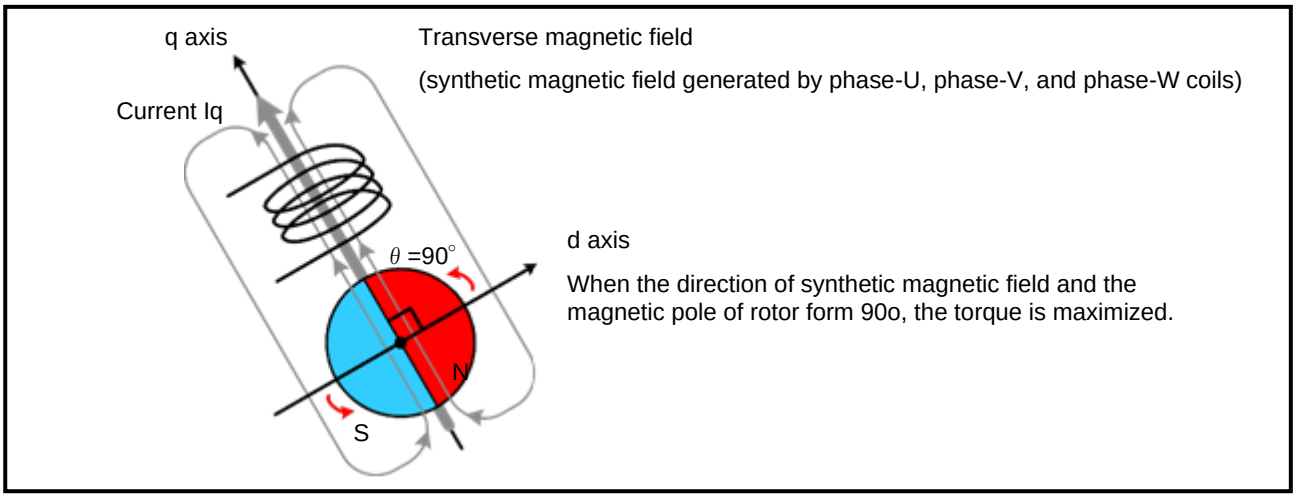


Figure 1.2 Relationship between Synthetic Magnetic Field and Torque

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.

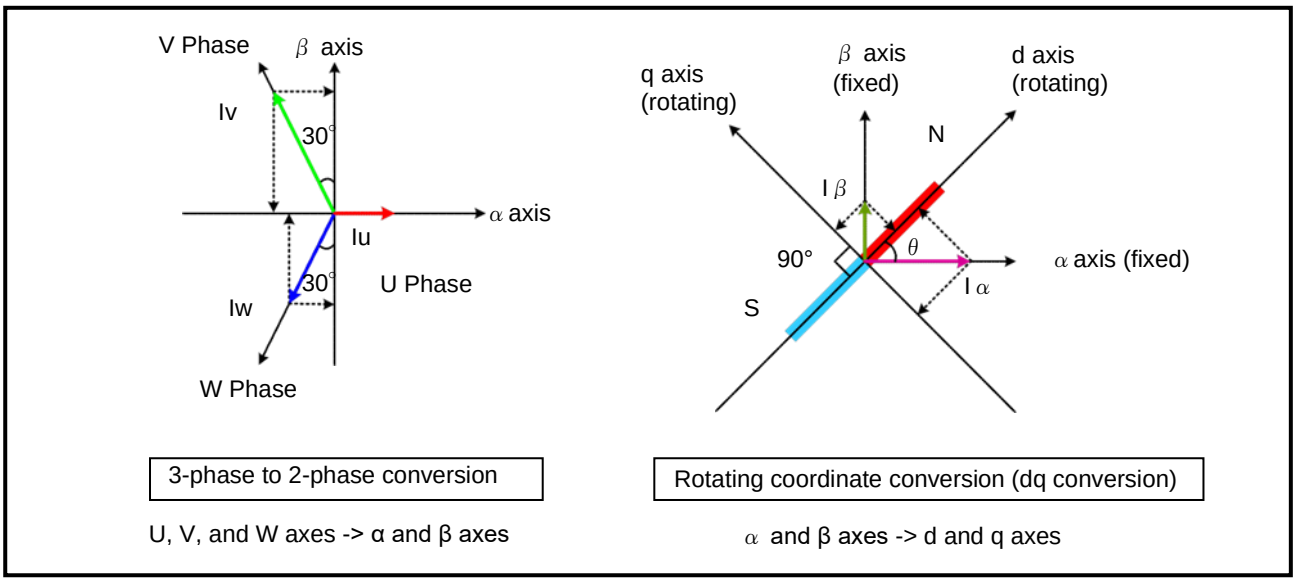


Figure 1.3 Coordinate Conversions

Calculate values of currents Iα and Iβ on the α and β axes by 3-phase to 2-phase conversion of the values of currents Iu, Iv, and Iw that are flowing through the phase-U, phase-V, and phase-W coils and were measured by the n'th measurement.

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

Perform rotating coordinate conversion of Iα and Iβ obtained by the 3-phase to 2-phase conversion to calculate values of currents Id and Iq on the d and q axes.

$$\begin{pmatrix} I_d(n) \\ I_q(n) \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} I_\alpha(n) \\ I_\beta(n) \end{pmatrix}$$

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

$$\varepsilon_d(n) = I_{dt} - I_d(n)$$

$$\varepsilon_q(n) = I_{qt} - I_q(n)$$

$$V_d(n) = V_d(n-1) + K_P \cdot \{\varepsilon_d(n) - \varepsilon_d(n-1)\} + K_I \cdot \varepsilon_d(n) \cdot \Delta t$$

$$V_q(n) = V_q(n-1) + K_P \cdot \{\varepsilon_q(n) - \varepsilon_q(n-1)\} + K_I \cdot \varepsilon_q(n) \cdot \Delta t$$

Remarks:
- Idt: Command value of d-axis current
- Iqt: Command value of q-axis current
- Id(n): d-axis current sampled by n'th sampling
- Iq(n): q-axis current sampled by n'th sampling
- KP: Proportional gain
- KI: Integration gain
- Δt: Sampling time
- ε d: Deviation of d-axis current
- ε q: Deviation of q-axis current

Convert the q-axis and d-axis voltages calculate above by coordinate conversions (fixed coordinate conversion and 2-phas to 3-phase conversion) to obtain output voltages in U, V, and W phases. The equation of fixed coordinate conversion is as follows:

$$\begin{pmatrix} V_\alpha(n) \\ V_\beta(n) \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} V_d(n) \\ V_q(n) \end{pmatrix}$$

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

$$\begin{pmatrix} V_u(n) \\ V_v(n) \\ V_w(n) \end{pmatrix} = \sqrt{\frac{2}{3}} \begin{pmatrix} \cos 0 & \sin 0 \\ \cos\frac{2}{3}\pi & \sin\frac{2}{3}\pi \\ \cos\frac{4}{3}\pi & \sin\frac{4}{3}\pi \end{pmatrix} \begin{pmatrix} V_\alpha(n) \\ V_\beta(n) \end{pmatrix}$$

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.

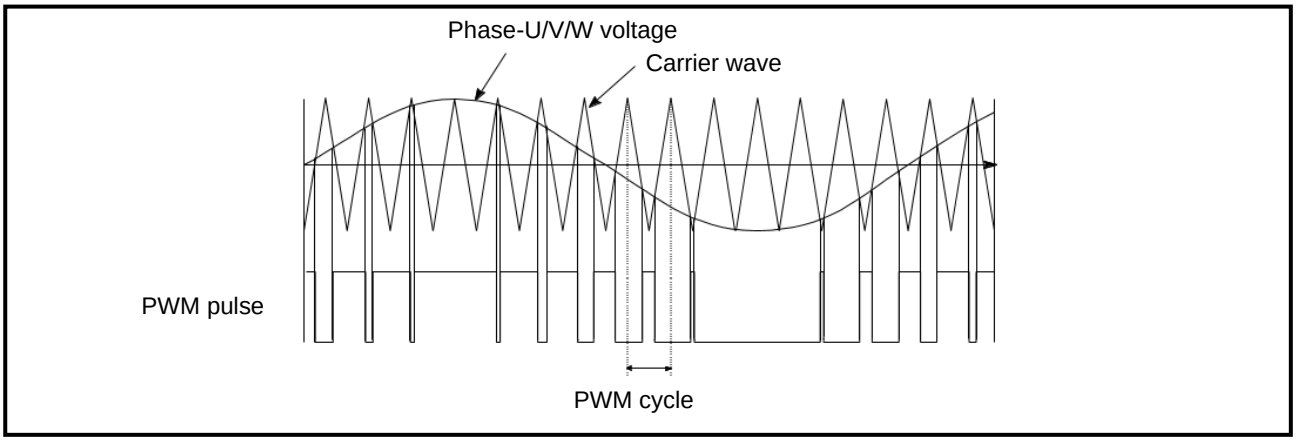


Figure 1.4 Triangular Wave Comparison Method

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 an 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 3.4.5.4 PI Control IP.

$$\begin{bmatrix} vd \\ vq \end{bmatrix} = \begin{bmatrix} Ra + pLd & -\omega Lq \\ \omega Ld & Ra + pLq \end{bmatrix} \begin{bmatrix} id \\ iq \end{bmatrix} + \begin{bmatrix} 0 \\ \omega\varphi a \end{bmatrix} \quad p = \frac{d}{dt}$$

Remarks:

vd, vq : : 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

ω: Angular speed of motor

ϕ: Flux of permanent magnet φa=√(2/3) φ

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:

$$Id = \frac{1}{Ra + sLd}(Vd + \omega LqIq)$$

$$Iq = \frac{1}{Ra + sLq}\left(Vq - \omega\left(LdId + \sqrt{\frac{2}{3}}\varphi\right)\right)$$

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. Typical control example of EMU3S

This application note describes motor control that uses the 12-bit A/D converter (ADCK), R/D converter (RDC3AL), enhanced motor control unit 3S(EMU3S), 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 EMU3S 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.

The following is a typical control example of EMU3S.

- Fully automatic processing operation

  This is a control example when motor control is performed without the intervention of a CPU. For details, refer to "3. Fully automatic processing operation".

- Control operation with CPU processing

  The user, however, can insert user's original processing into the control processing by EMU3S. For details, refer to "4. Control operation with CPU processing".

- Rectangular wave output operation

  As a function of EMU3S, it is a control example when outputting a rectangular wave according to the angle. For details, refer to "5. Square wave output operation".

3. Fully automatic processing operation

3.1 Specifications

In this chapter, the CPU only sets current command values and other control is performed by hardware units alone without intervention by the CPU.

Table 3.1 lists the specifications of motor control. Table 3.2 lists settings of individual modules. Figure 3.1 shows the configuration of a 2-motor control system (using the RH850/U2B6). Figure 3.2 shows the flow of fully automated control processes.

Table 3.1 Motor Control Specifications

Item	Specification
Output waveform	Complemented 3-phase PWM waveform
Carrier frequency	10 kHz (100 µsec/cycle)
Control method	180-degree excitation drive method
Active level	Active high
Short-circuit prevention time (dead time)	4 µsec
Timing of updating compare register and carrier frequency	Trough of the carrier wave
Timing of starting A/D conversion	Trough of the carrier wave
Interrupt	Used (100usec interrupt)
Motor current value	The ADCK0 obtains values of phase-U and phase-V currents.
Motor angle information	The RDC3AL obtains angle information from resolver.

Table 3.2 Module Settings

Peripheral Function	Setting
RDC3AL	• Using the 10-kHz excitation signal generated in the RDC3AL • Automatic setting of PI compensator band
PIC	• Using the signal from EMU30 as a trigger for ADCK0 scan group 4
ADCK	• 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 EMU3S
TSG3	• HT-PWM mode • Carrier frequency: 100 µsec • Dead time: 4 µsec • Direct transfer of Carrier frequency and PWM duty settings from the EMU3S
EMU3S	• 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 ADCK, angle value and angular velocity value from the RDC3AL for computation • Starting A/D conversion in the timing of a trough of the carrier wave • 1usec delay in trigger output for A/D conversion and R/D conversion • Selectable non-interference control/voltage compensation

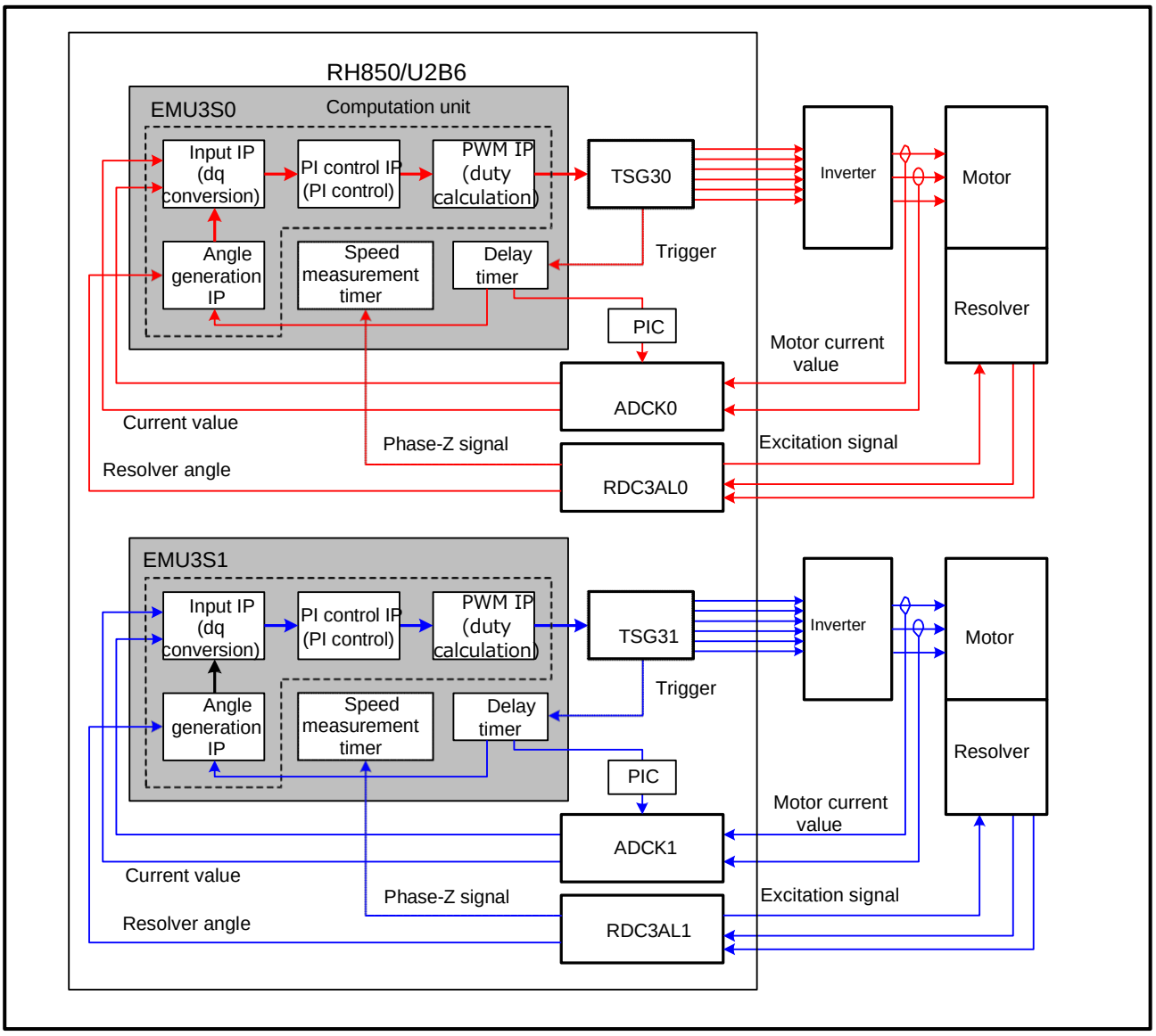


Figure 3.1 Configuration of 2-Motor Control System (using the RH850/U2B6)

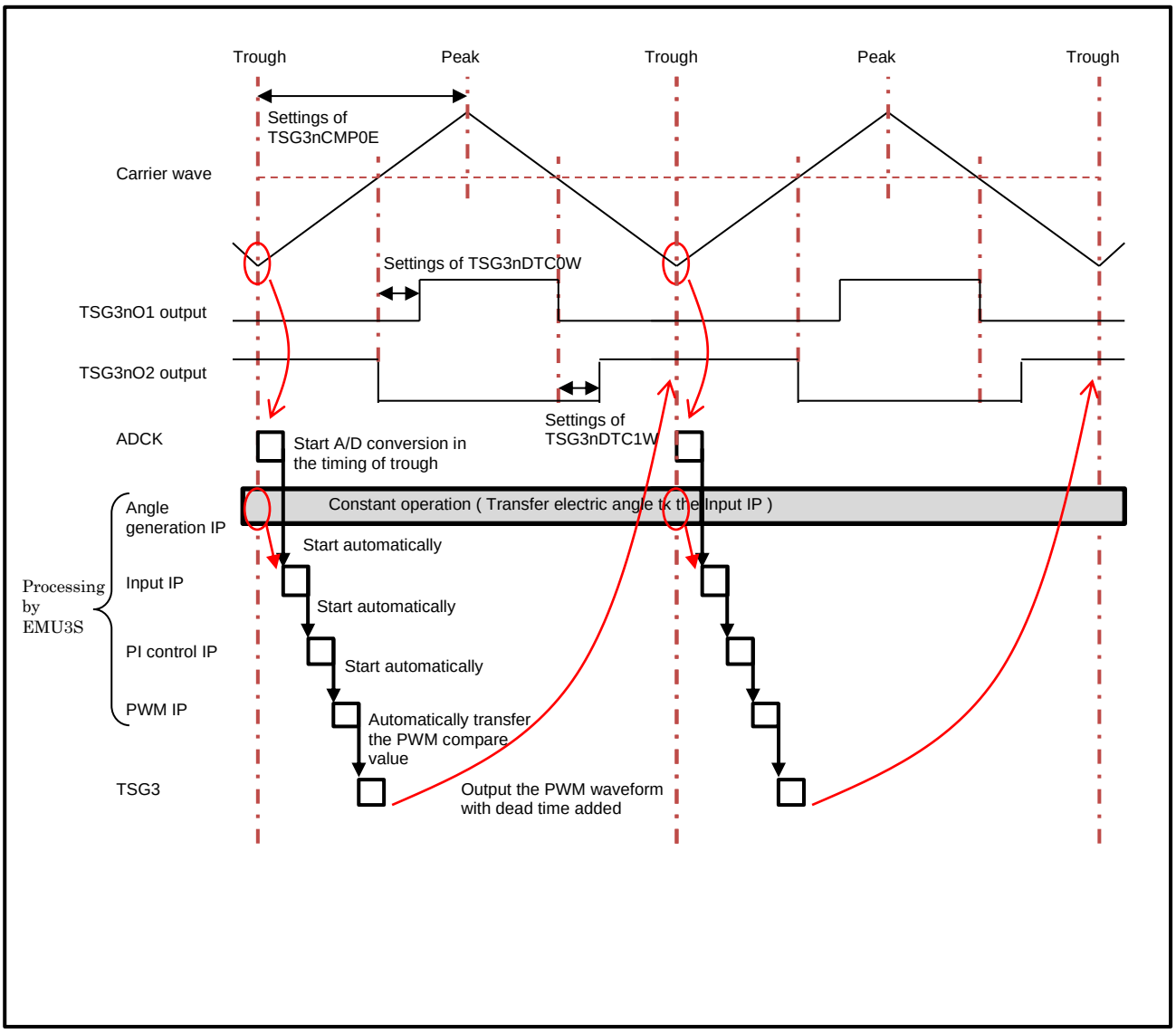


Figure 3.2 Flow of Fully Automated Control

3.2 Operation Check Conditions

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

Table 3.3 Operation Check Conditions

Item	Condition
Microcontroller	RH850/U2B6
Operating frequency	- Main OSC: 20 MHz - CPU clock: 400 MHz Unmodulated high-speed peripheral clock: 80 MHz Unmodulated low-speed peripheral clock: 40 MHz Motor control H/W accelerator clock: 100MHz
Operating voltage	VDD = 1.12 V PVCC = 5.0 V
Integrated development environment	Renesas Electronics' CS+ E8.07.00
C compiler	Renesas Electronics' C Compiler Package for RH850 Family V2.02.00
	Compilation options Default settings of the integrated development environment
Sample code version	Version 1.00

3.3 Pins

Table 3.4 lists the pins to be used and their functions.

Table 3.4 Pins to be Used and their Functions

Module	Pin	Input/Output	Function
ADCK	AN000	Input	Input of measured phase-V current
	AN001	Input	Input of measured phase-W current
	AN002	Input	Input of measured phase-U current
RDC3AL	RDC3AL0S1	Input	Input of resolver signal (cosθ)
	RDC3AL0S3	Input	Input of resolver signal (cosθ)
	RDC3AL0S2	Input	Input of resolver signal (sinθ)
	RDC3AL0S4	Input	Input of resolver signal (sinθ)
	RDC3AL0RSO	Output	Output of excitation signal
	RDC3AL0COM	Output	Output of common voltage for excitation signal
TSG3	TSG30O1	Output	Output of upper-armature phase-U signal
	TSG30O2	Output	Output of lower-armature phase-U signal
	TSG30O3	Output	Output of upper-armature phase-V signal
	TSG30O4	Output	Output of lower-armature phase-V signal
	TSG30O5	Output	Output of upper-armature phase-W signal
	TSG30O6	Output	Output of lower-armature phase-W signal

3.4 Overview of Operation

The following describes motor control operation:

(1) When the EMU3S 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 ADCK0 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 EMU3S computation unit converts the resolver angle obtained from the RDC3AL 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 EMU3S, 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 RDC3AL.

(5) 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) Current command values are set for EMU3S in OSTM interrupt function. Thereafter, a rotation command is given to the motor to be controlled.

3.4.1 R/D Converter 3AL (RDC3AL)

The RDC3AL converts the analog signal output from the resolver into digital signals, and calculates an angle value (resolver angle value) and angular velocity.

3.4.2 Peripheral Interconnection (PIC)

The PIC generates an ADCK hardware trigger signal based on a signal that the EMU3S outputs in synchronization with the carrier wave.

3.4.3 A/D Converter (ADCK)

The ADCK performs A/D conversion of a motor current value, and inputs the converted value to the EMU3S. 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 EMU3S. When the trigger signal is input, the ADCK converts the phase-V current value input from AN000 pin, phase-W current value input from AN001 pin and phase-U current value input from AN002 pin. The phase-W current value is not used in the process.

3.4.4 Motor Control Timer 3 (TSG3)

The TSG3 outputs a 3-phase PWM waveform based on the compare value transferred from the EMU3S.

TSG3 generates duty from compare match of 18-bit counter and 18-bit sub counter (but uses 16-bit for combination with EMU3S) and outputs 3-phase PWM waveform with dead time setting.

Figure 5.1 shows a timing chart for the 3-phase PWM waveform.

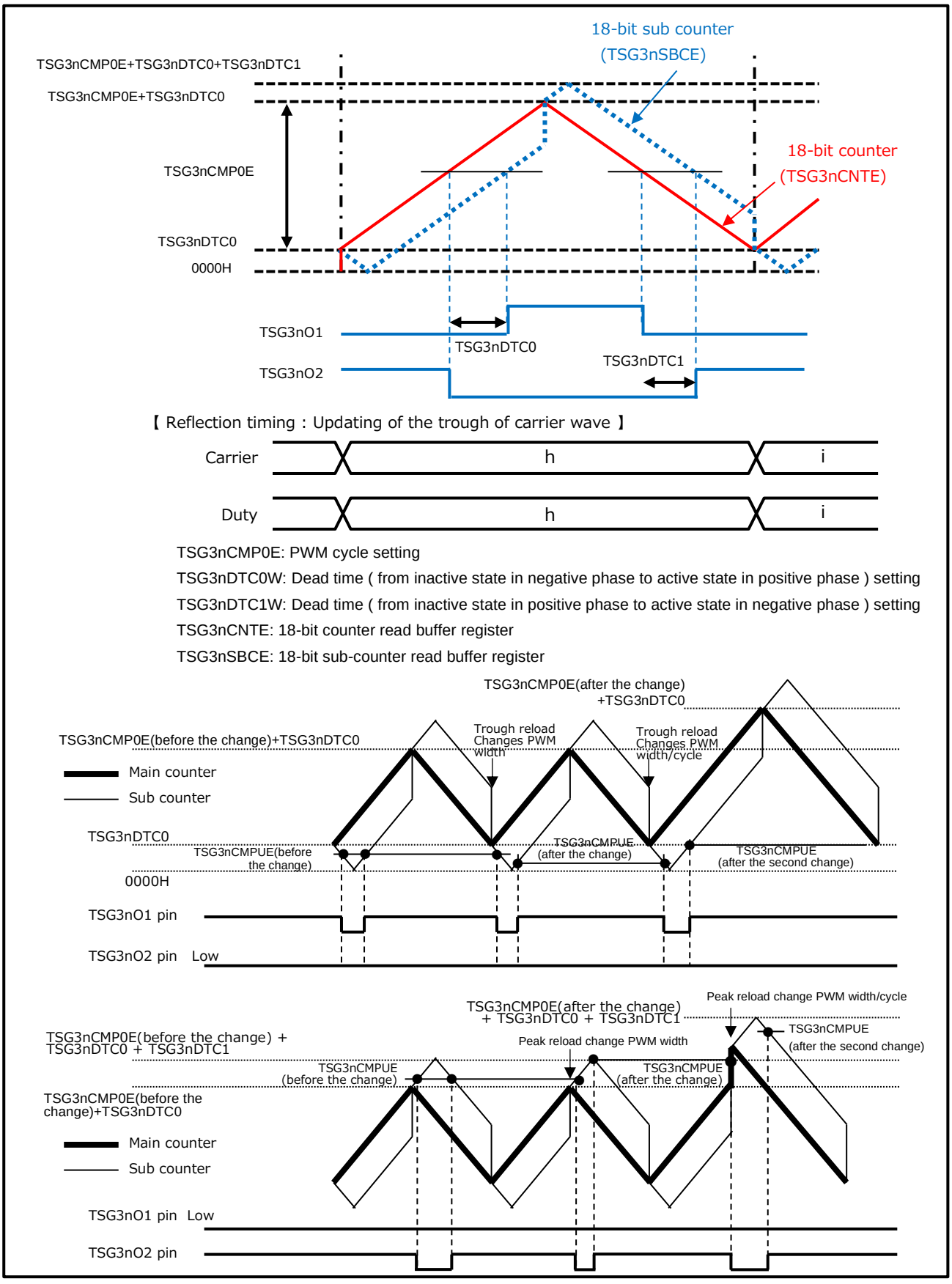


Figure 3.3 Timing Chart for 3-Phase PWM Waveform

3.4.5 Enhanced Motor Control Unit 3S (EMU3S)

3.4.5.1 EMU3S Common Unit

The EMU3S 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)
- SMVAN bit in EMU3nSMCTR register = 1 (Starting A/D converter in the timing of carrier trough)
- ADDATA bit in EMU3nADDCNT register = 80 (1usec delay time in A/D conversion)
- RDDATA bit in EMU3nRDDCNT register = 80 (1usec delay time in the output of R/D conversion trigger)
- EMU3nINT0 to EMU3nINT7 registers = 0000 0000H (Disabling interrupt)
- INTC2EIC38 to EIC45 registers = 00CFH (Disabling interrupt processing)

3.4.5.2 Angle Generation IP

The angle generation IP starts each time the angle data value (high-order 16 bits in the RDC3ALnENC1 register) input from the RDC3AL changes. After adding an offset value (EMU3nANGOFS register) to the value of the RDC3ALnENC1 register, the angle generation IP generates an electric angle. The angle generation IP stores the generated electric angle in the EMU3nTHTEFIX register. Figure 3.4 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 RDC3AL)
- 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.

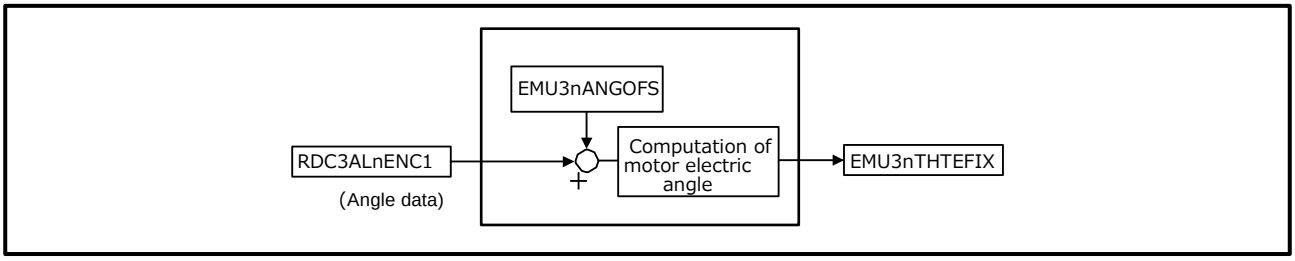


Figure 3.4 Block Diagram for Angle Generation IP

3.4.5.3 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 3.5 shows a block diagram for the input IP.

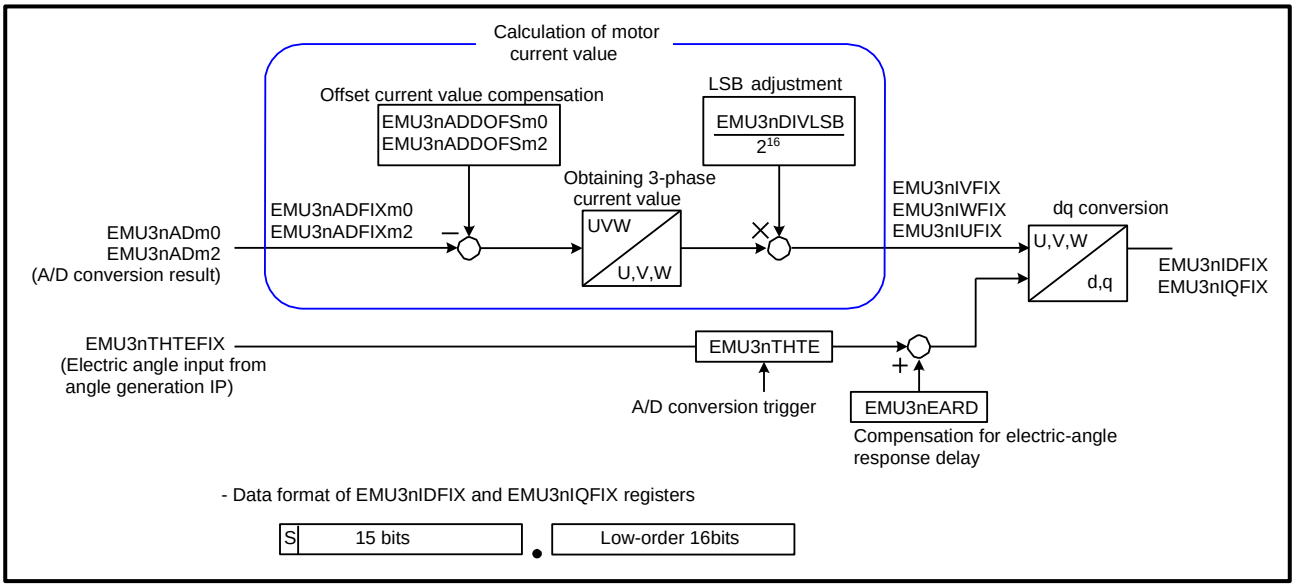


Figure 3.5 Block Diagram for Input IP

(1) Calculation of Motor Current Values

The input IP calculates 3-phase current values from the 2-phase current values stored in the EMU3nADm0 and EMU3nADm2 registers. Calculation results are stored in the EMU3nIVFIX, EMU3nIWFIX, and EMU3nIUFIX 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 3.6 shows the offset compensation for the A/D conversion result.

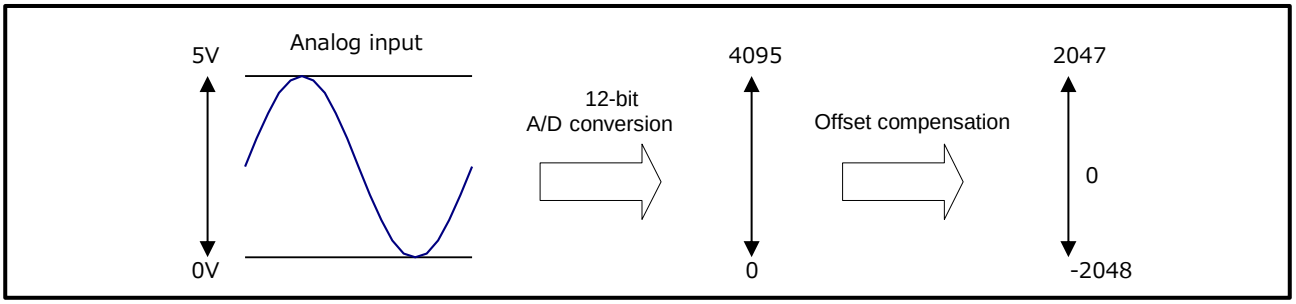


Figure 3.6 Offset Compensation for A/D Conversion Result

$$\mathrm{EMU3nIUFIX} \leftarrow \left((\mathrm{EMU3nAD0FIX} - \mathrm{EMU3nAD0OFS}) \times \mathrm{EMU3nDIVLSB}\right) \gg 16$$
$$\mathrm{EMU3nIVFIX} \leftarrow \left((\mathrm{EMU3nAD1FIX} - \mathrm{EMU3nAD1OFS}) \times \mathrm{EMU3nDIVLSB}\right) \gg 16$$
$$\mathrm{EMU3nIWFIX} \leftarrow -(\mathrm{EMU3nIUFIX} + \mathrm{EMU3nIVFIX})$$

(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 EMU3nIUFIX, EMU3nIVFIX, and EMU3nIWFIX 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 EMU3nIDFIX and EMU3nIQFIX 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]

$$\begin{pmatrix}\mathrm{EMU3nIDFIX}\\ \mathrm{EMU3nIQFIX}\end{pmatrix} = \mathrm{EMU3nSR2} \times \begin{pmatrix}\sin(\theta e + 90°) & -\sin(\theta e + 150°) & -\sin(\theta e + 30°)\\ -\sin(\theta e + 0°) & \sin(\theta e + 60°) & -\sin(\theta e + 120°)\end{pmatrix}\begin{pmatrix}EMU3nIUFIX\\ EMU3nIVFIX\\ EMU3nIWFIX\end{pmatrix}$$

3.4.5.4 PI Control IP

With "1" set in the PIIPTRG bit in the EMU3nIPTRG 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 3.7 shows a block diagram for the PI control IP.

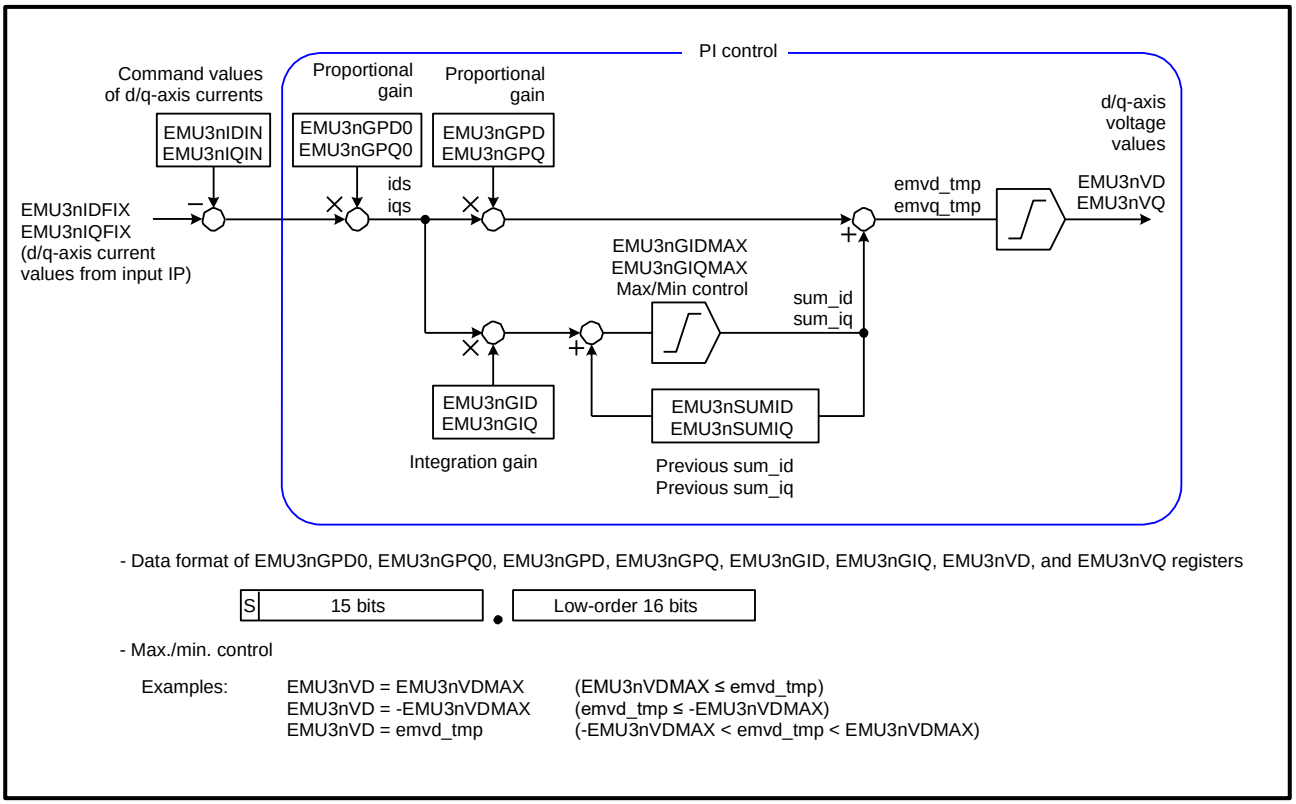


Figure 3.7 Block Diagram for PI control IP

(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)

3.4.5.5 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 3.8 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.

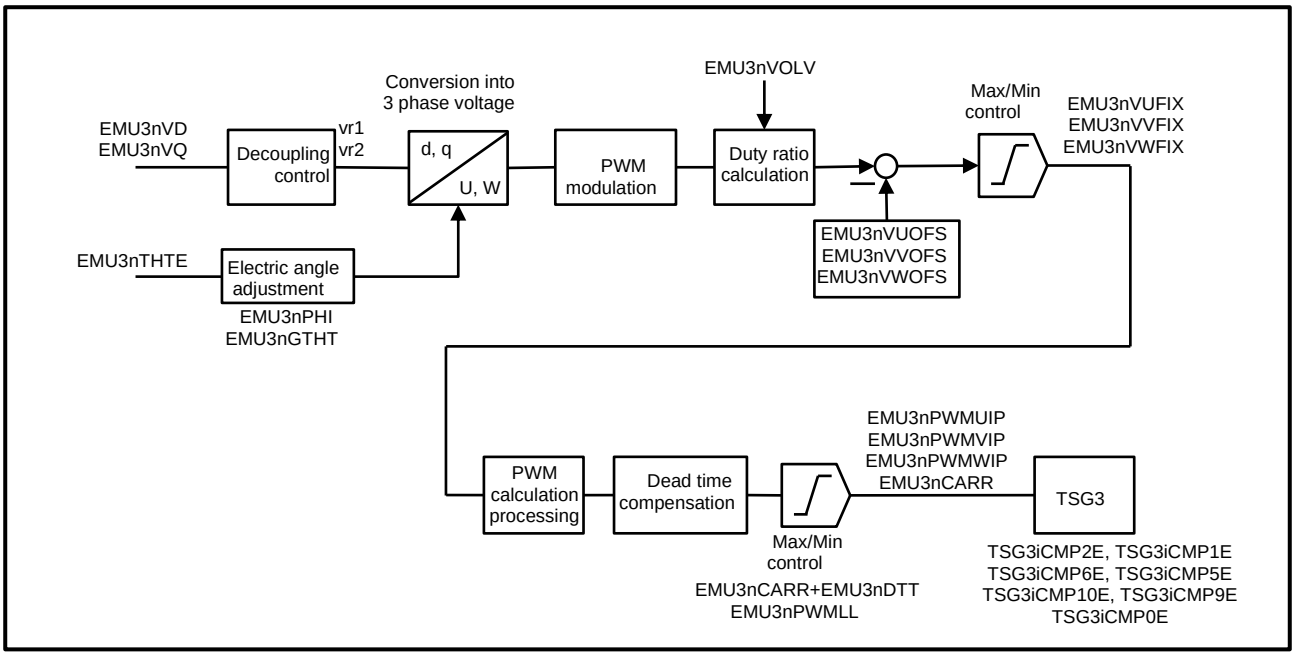


Figure 3.8 Block Diagram for PWM IP

(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. 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

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.

vdi <- 0 + EMU3nVDCRCT

vqi <- 0 + EMU3nVQCRCT

• Non-interference control

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 RDC3AL 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-axis voltage <- d-axis voltage - (ωe × Gωe × Lq × Iq)

q-axis voltage <- q-axis voltage + (ωe × Gωe × Ld × Id + ωe × Gωe × Φ)

(ωe: angular speed of electric angle, Φ: flux, I: current, L: inductance, Gωe: angular speed gain)

[Settings of related registers]

- EMU3nDECVELG register= 0000 0000H
- EMU3nDECFLIX register= 0000 0000H
- EMU3nDECLD register= 0000 0000H
- EMU3nDECLQ register= 0000 0000H

(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]

$$\begin{pmatrix} vuo \\ vwo \end{pmatrix} = EMU3nSR23 \times \begin{pmatrix} \cos(\theta + 0°) & -\sin(\theta + 0°) \\ \cos(\theta + 120°) & -\sin(\theta + 120°) \end{pmatrix} \begin{pmatrix} vdi \\ vqi \end{pmatrix}$$

$$vvo = -(vuo + nwo)$$

(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 vuo,vwo and vvo 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]

$$EMU3nVUFIX = vuo \times \frac{EMU3nPWMK1}{2^{16}} \times \frac{1}{EMU3nVOLV} + EMU3nVUOFS$$

$$EMU3nVUFIX = vuo \times \frac{EMU3nPWMK1}{2^{16}} \times \frac{1}{EMU3nVOLV} + EMU3nVUOFS$$

$$EMU3nVUFIX = vuo \times \frac{EMU3nPWMK1}{2^{16}} \times \frac{1}{EMU3nVOLV} + EMU3nVUOFS$$

(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 = 140H (320) (Dead time: 4 µsec = 12.5 nsec x 320)
- EMU3nPWMK2 register = 01000000H (Adjustment of the range of PWM compare value)
- EMU3nPWMUL register = FFFFH (Setting of the upper limit of PWM compare value)
- EMU3nPWMLL register = 0000H (Setting of the lower limit of PWM compare value)
- EMU3nPWMU register = 1040H (Phases U PWM software input compare value
  (EMU3nCARR+ EMU3nDTT) / 2)
- EMU3nPWMV register = 1040H (Phases V PWM software input compare value
  (EMU3nCARR+ EMU3nDTT) / 2)
- EMU3nPWMW register = 1040H (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:

$$\mathrm{pwmu} = \left(\mathrm{EMU3nVUFIX} \times \frac{(\mathrm{EMU3nCARR} + \mathrm{EMU3nDTT})}{2} \times \mathrm{EMU3nPWMK2}\right) \times \frac{1}{2^{32}} + \frac{(\mathrm{EMU3nCARR} + \mathrm{EMU3nDTT})}{2}$$

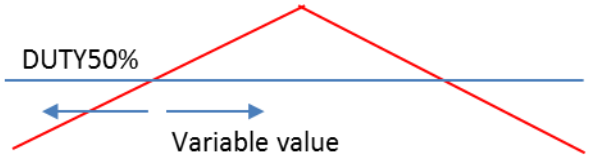
$$\mathrm{EMU3nPWMUIP} = (\mathrm{EMU3nCARR} + \mathrm{EMU3nDTT}) / 2 \quad (\mathrm{EMU3nVOLV} = 0)$$

$$\mathrm{EMU3nPWMUIP} = \mathrm{EMU3nCARR} + \mathrm{EMU3nDTT} \quad \left(\mathrm{pwmu} \geq (\mathrm{EMU3nCARR} + \mathrm{EMU3nDTT} - \mathrm{EMU3nPWMUL})\right)$$

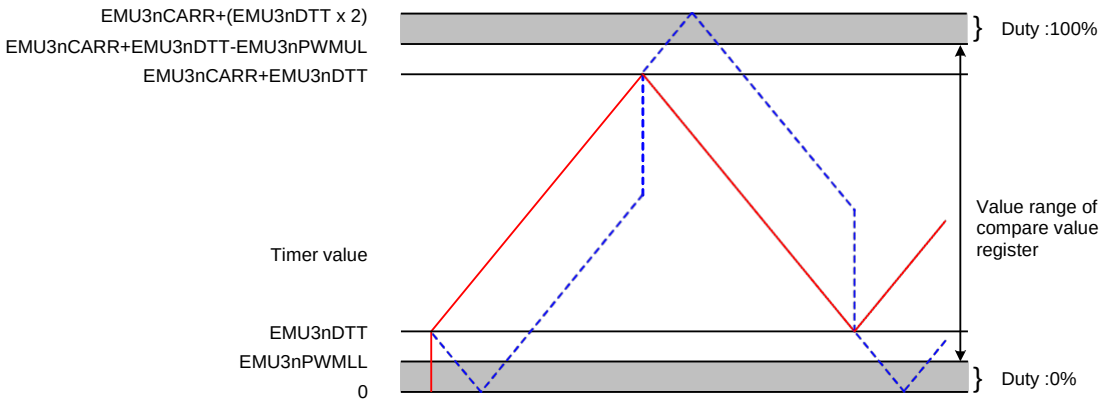
$$\mathrm{EMU3nPWMUIP} = 0 \quad (\mathrm{EMU3nVOLV} = 0)$$

$$\mathrm{EMU3nPWMUIP} = \mathrm{pwmu} \quad (\mathrm{EMU3nVOLV} = 0)$$

$$\text{DUTY50\%} = \frac{EMU3nCARR + EMU3nDTT}{2}$$



$$\text{Variable value} = \left(\text{EMU3nVUFIX} \times \frac{EMU3nCARR + EMU3nDTT}{2} \times EMU3nPWMK2\right) \times \frac{1}{2^{16}}$$



EMU3nCARR : Carrier half-cycle register
EMU3nDTT : Dead-time setting register
EMU3nPWMLL : PWM lower-limit register
EMU3nPWMUL : PWM upper-limit register

Figure 3.9 Limit Processing of PWM Value and Compare Value

3.4.5.6 Examples of Calculation Accuracy

It is assumed that the calculation of EMU3S operates with 32 bits based on 12 bits of input data from ADCK and RDC3AL and outputs it to TSG3 as 16 bits compare value.

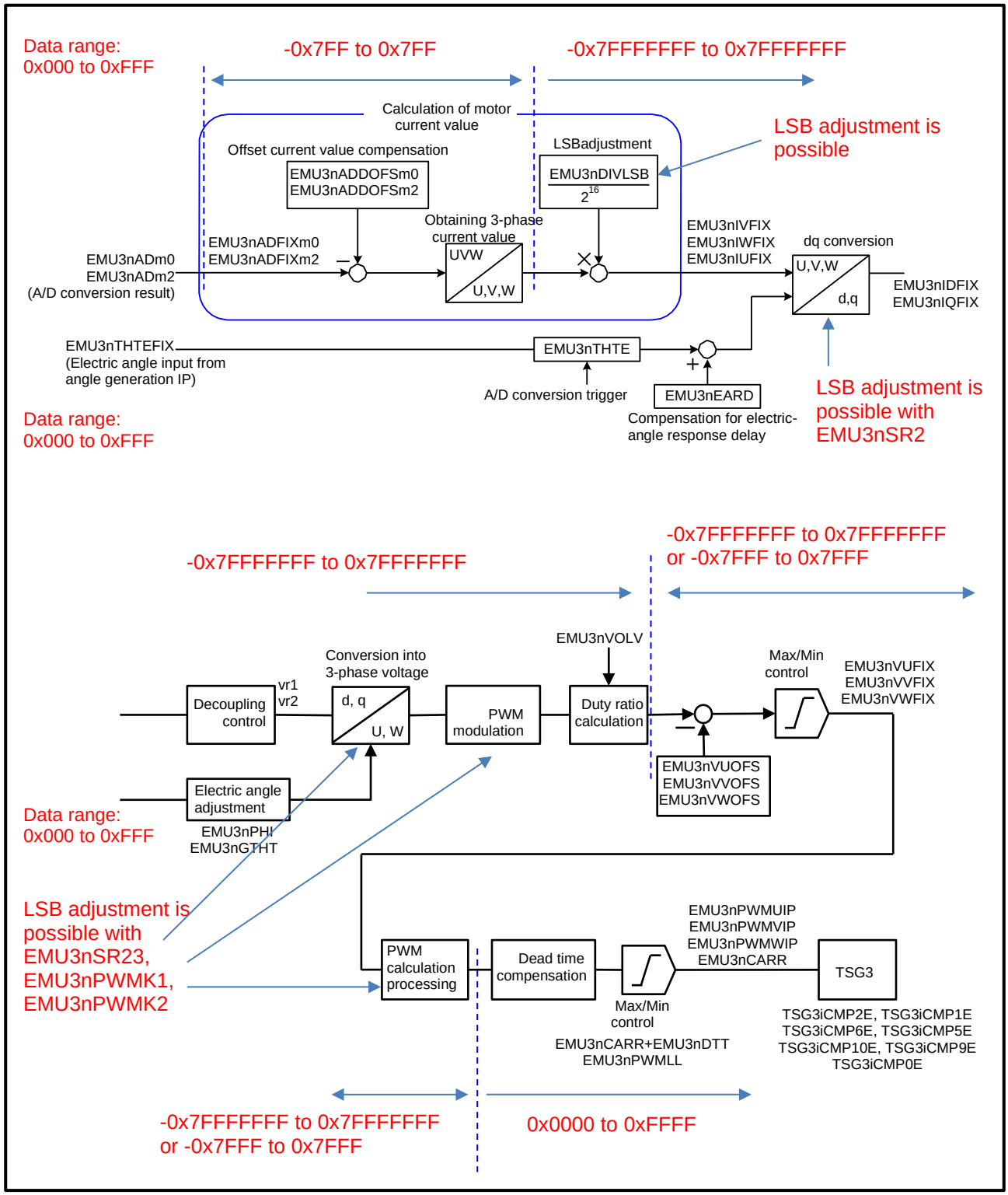


Figure 3.10 Relationship between Input and Output and Calculation Accuracy

3.4.6 File Configuration

Table 3.5 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 3.5 File Configuration

File name	Descript	Remarks
main.c	PE0 main module	PE1: main1.c PE2: main2.c PE1, PE2 main: infinite loop
rdc3al.c	R/D converter 3AL setting module	
pic.c	PIC setting module	
adck.c	A/D converter setting module	
tsg3.c	TSG3 setting module	
emu3s.c	EMU3S setting module	
port.c	Pin setting module	
ostm.c	OSTM setting module	
user_int.c	Implement interrupt operation.	

3.4.7 Functions

Table 3.6 lists functions.

Table 3.6 Functions

Function Name	Description
rdc3al_init	Initializes the R/D converter 3AL.
pic_init	Initializes the PIC.
adck_init	Initializes the A/D converter.
tsg3_init	Initializes the TSG3.
emu3s_init	Initializes the EMU3S.
port_init	Initializes the pins.
wait_mainloop	Wait processing for current command value write processing
set_dq_command	Current command value write processing

3.4.8 Specifications of Functions

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

rdc3al_init

Outline: Initialization of R/D converter 3AL
Header: iodefine.h
Declaration: void rdc3al_init(void)
Description: This function initializes the R/D converter 3AL.
Argument: None
Return value: None
Remarks:

pic_init

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

adck_init

Outline: Initialization of A/D converter
Header: iodefine.h
Declaration: void adck_init(void)
Description: This function initializes the A/D converter.
Argument: None
Return value: None
Remarks:

tsg3_init

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

emu3s_init

Outline: Initialization of EMU3S
Header: iodefine.h
Declaration: void emu3s_init(void)
Description: This function initializes the EMU3S.
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:

ostm_init

Outline: Executes wait processing
Header: iodefine.h
Declaration: void ostm_init (void)
Description: This function initializes the OSTM.
Argument: None
Return value: None
Remarks:

int_ostm_100us

Outline: Writes a current command value to Q-current target register of EMU30
after reading the voltage value from ADCK0 VR3.
Header: iodefine.h
Declaration: void int_ostm_100us (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

3.4.9 Flowcharts

Figure 3.11 shows the flow of main processing. Figure 3.12 shows the flow of RDC3AL initialization. Figure 3.13 shows the flow of PIC initialization. Figure 3.14 shows the flow of ADCK initialization. Figure 3.15 shows the flow of TSG3 initialization. Figure 3.16 shows the flow of EMU3S initialization (1). Figure 3.17 shows the flow of EMU3S initialization (2). Figure 3.18 shows the flow of pin setting. Figure 3.19 shows the flow of OSTM initialization and Figure 3.20 shows the flow of current command value write processing with 100us interrupt. Register synchronous processing is inserted in each function as necessary. However, it is omitted from the flow.

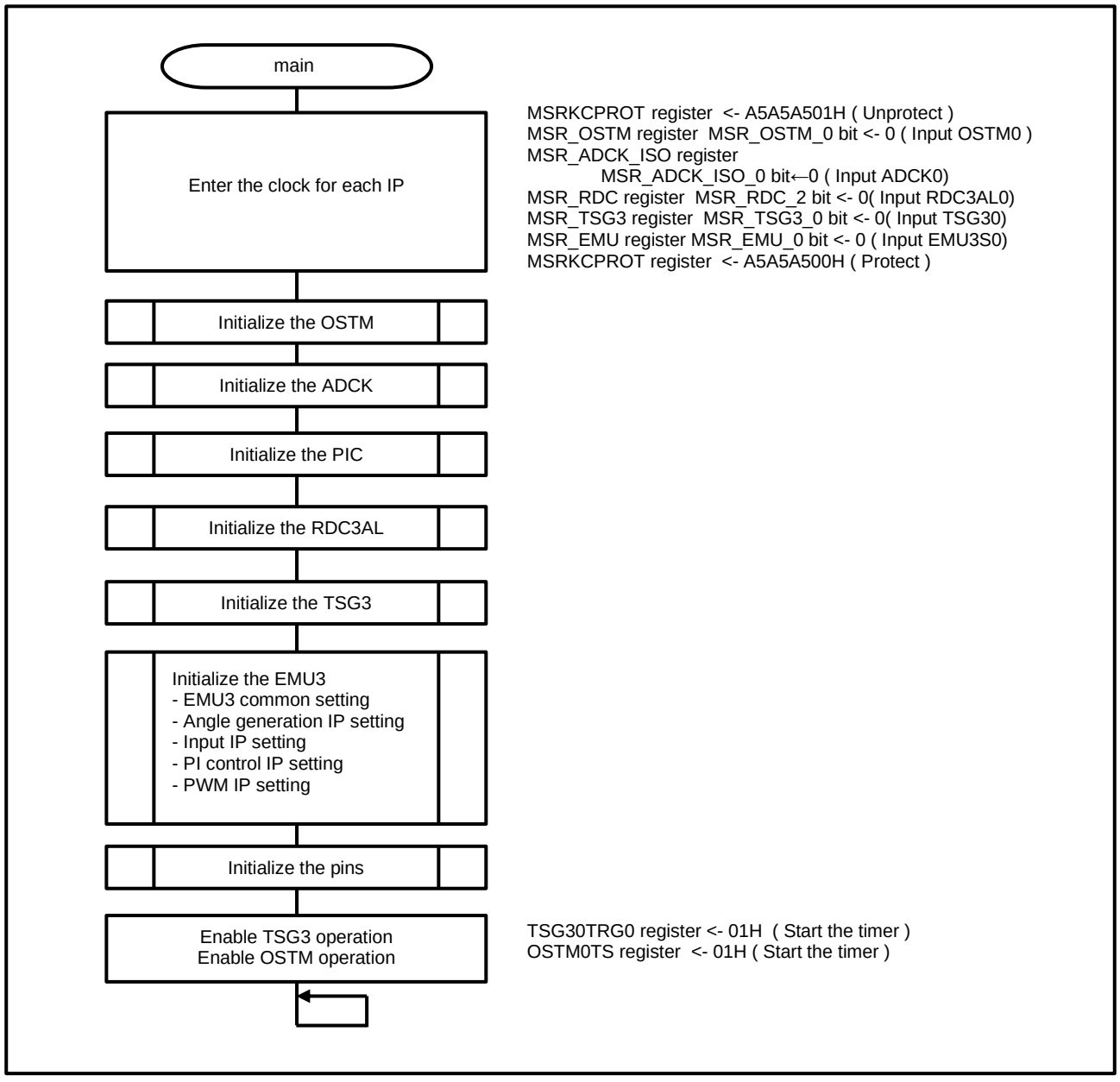


Figure 3.11 PE1 Main Processing

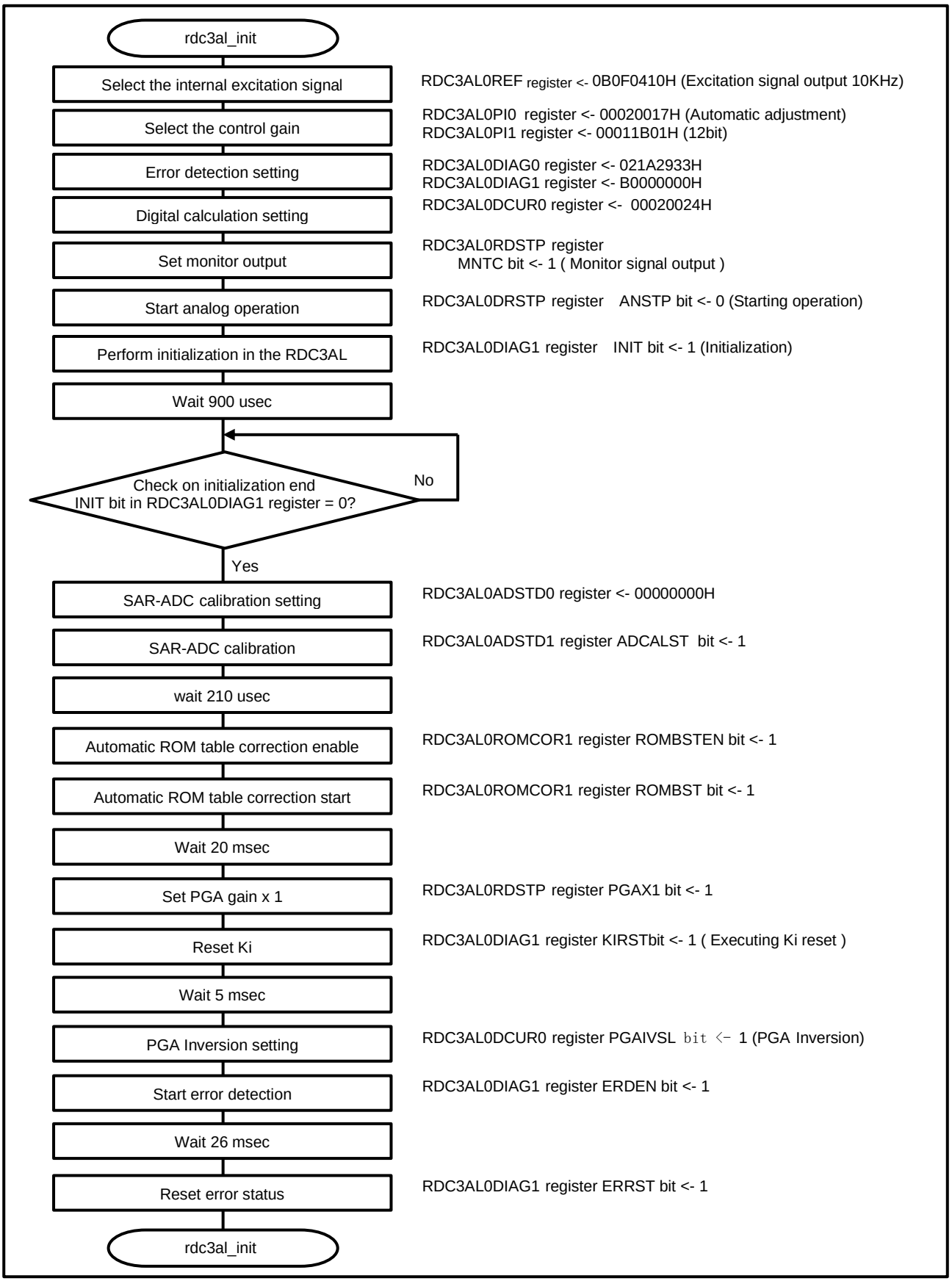


Figure 3.12 RDC3AL Initialization

Figure 3.13 PIC Initialization

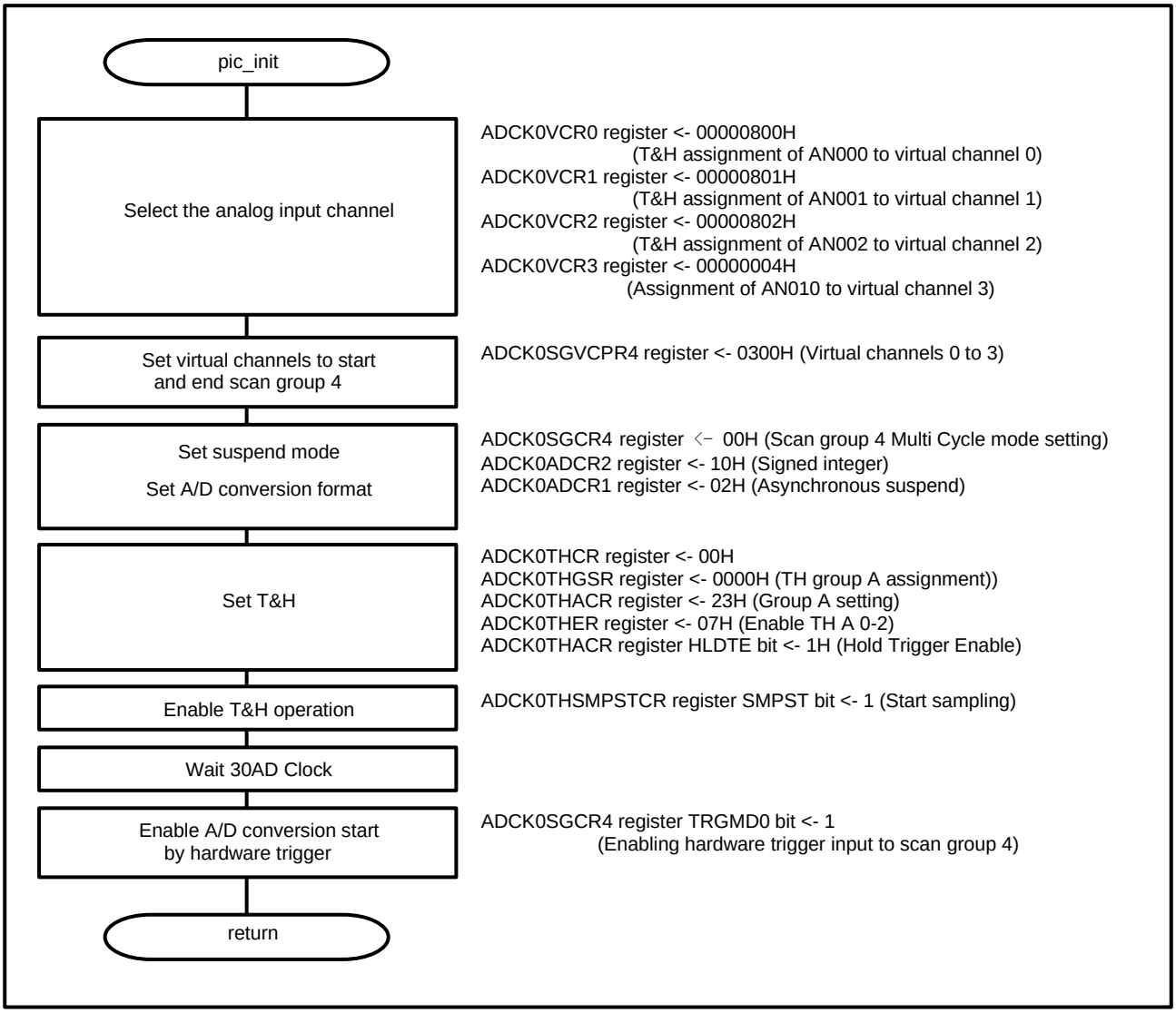


Figure 3.14 ADCK Initialization

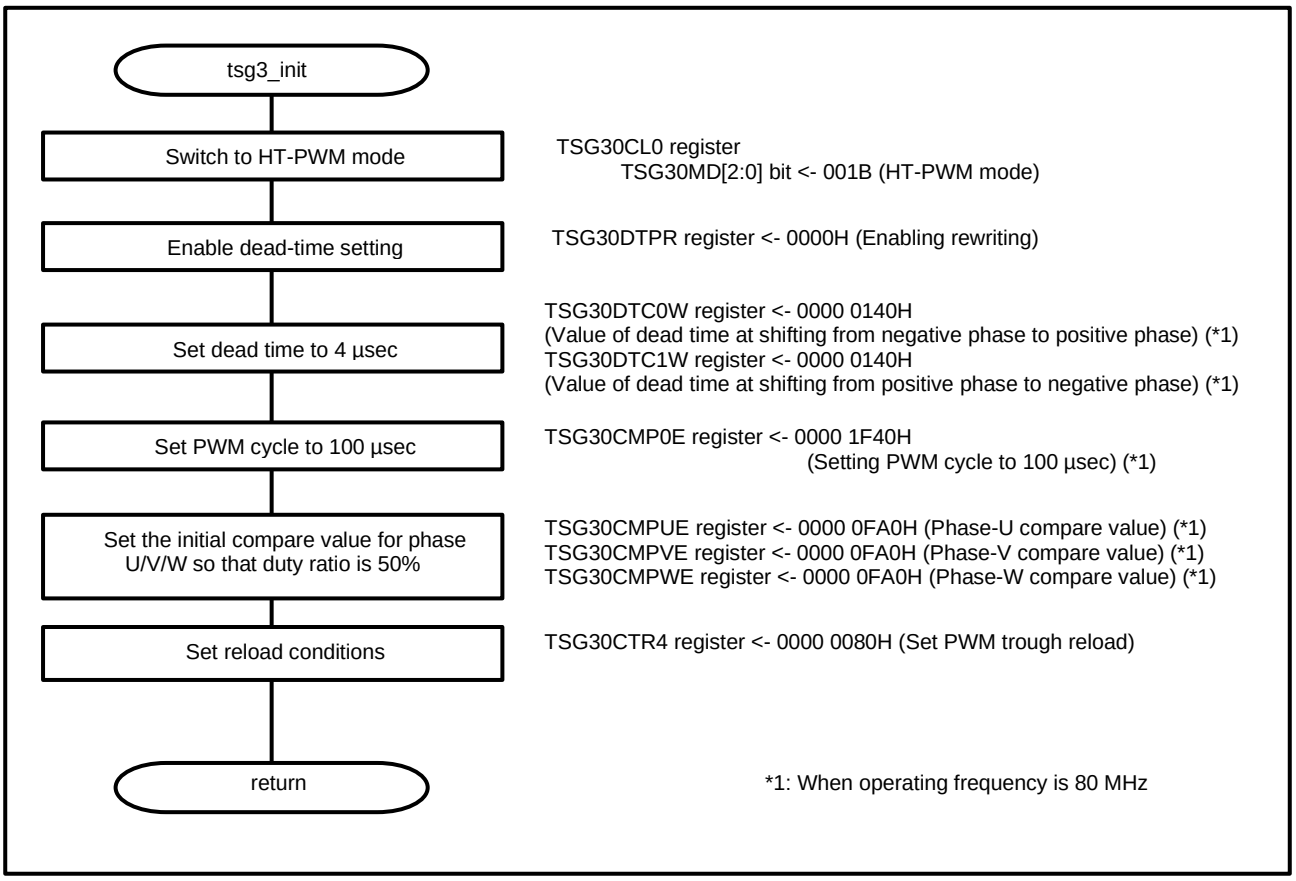


Figure 3.15 TSG3 Initialization

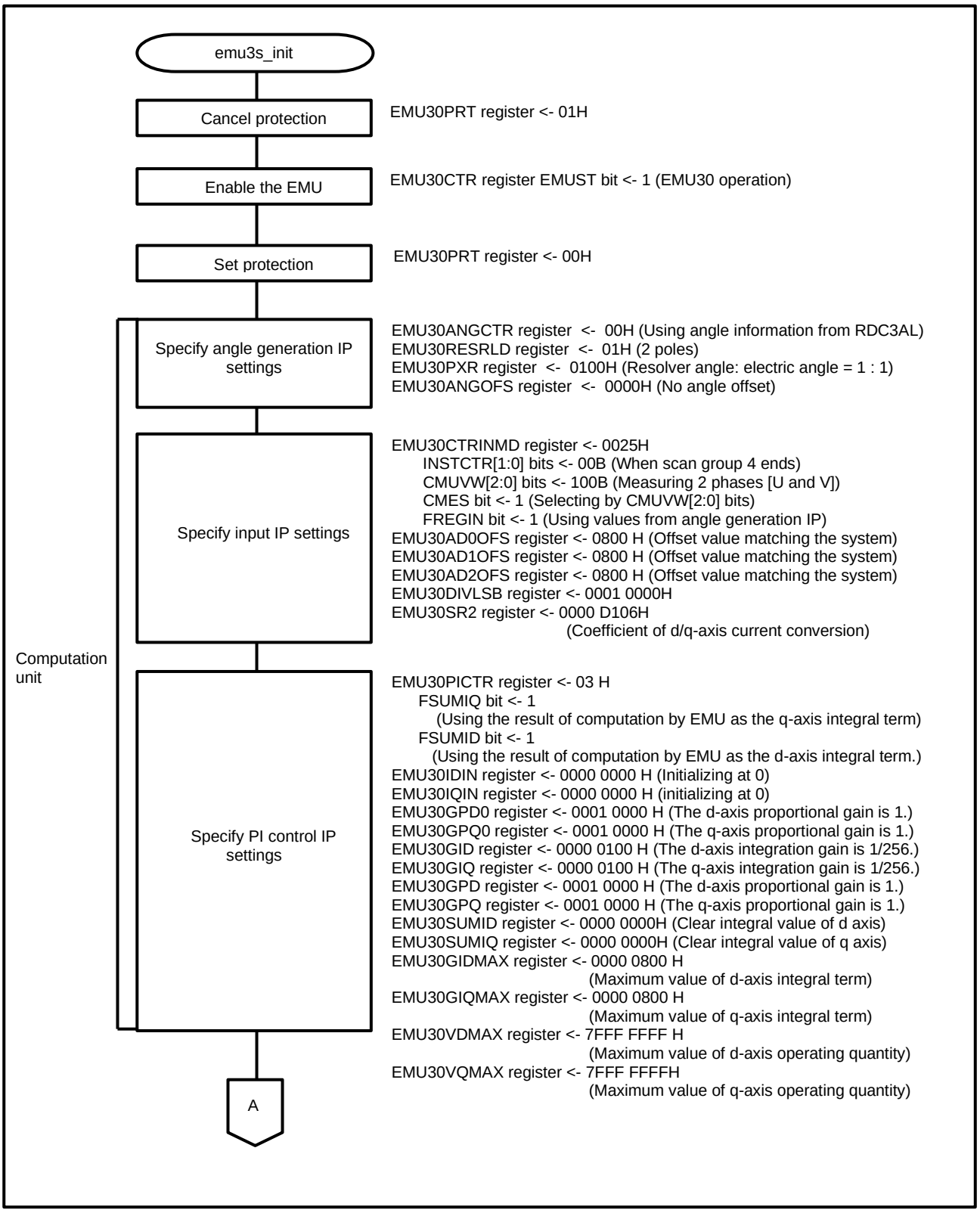


Figure 3.16 EMU3S Initialization (1)

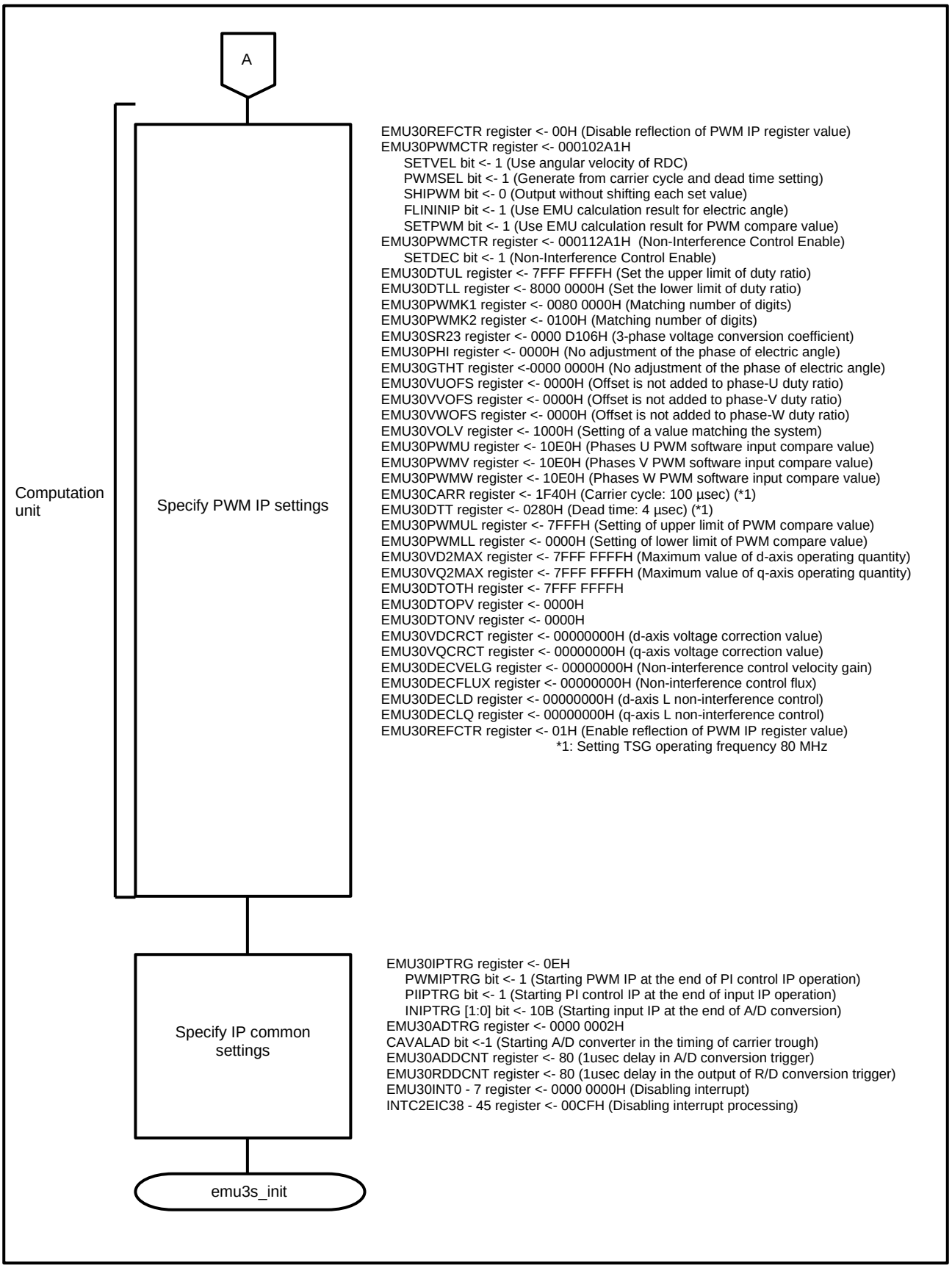


Figure 3.17 EMU3S Initialization (2)

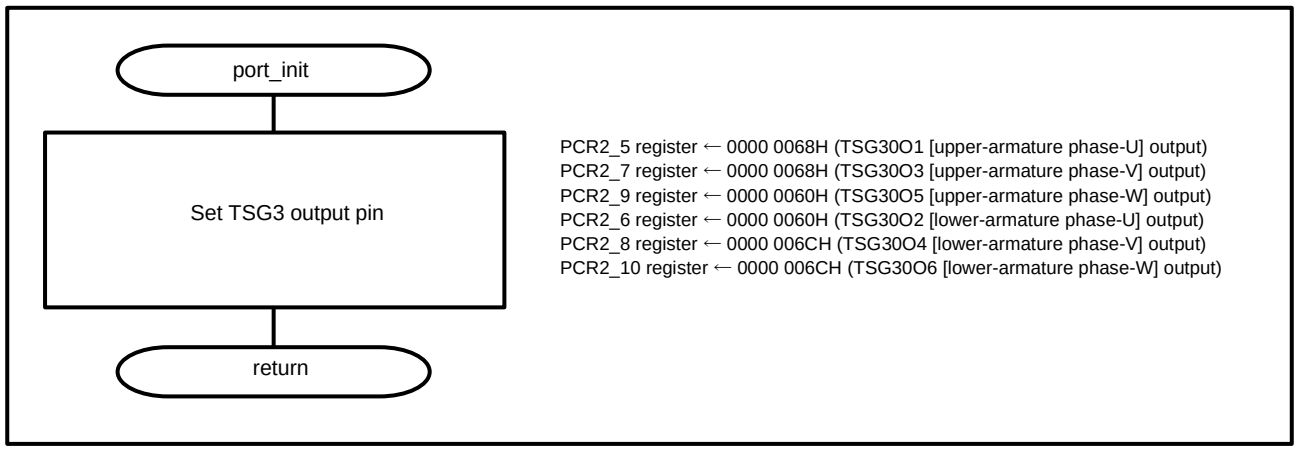


Figure 3.18 Pin setting

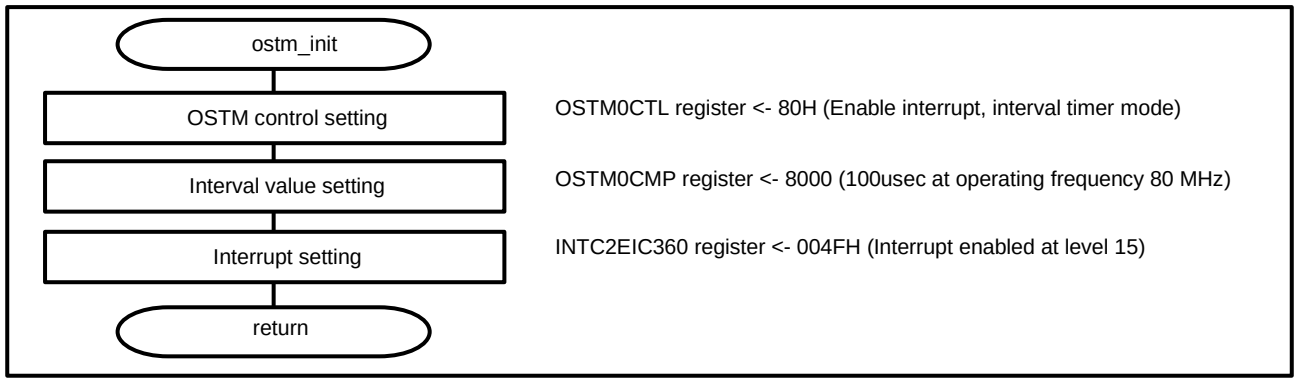


Figure 3.19 OSTM Initialization

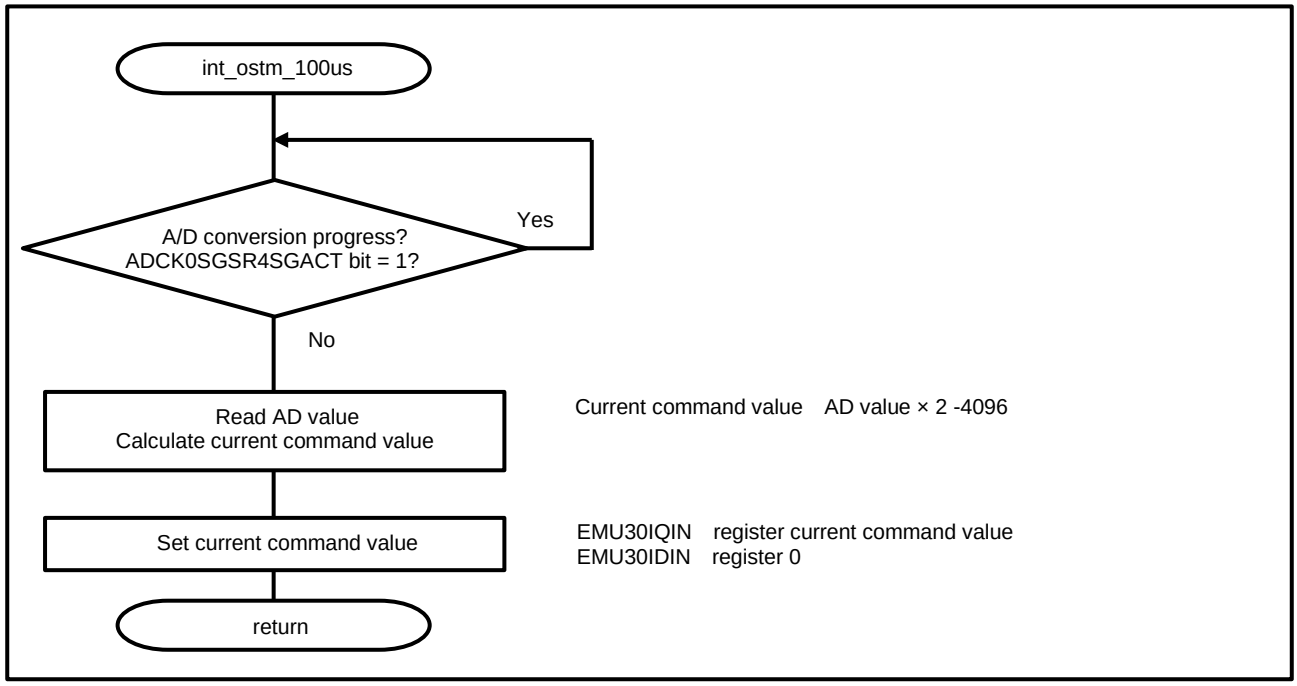


Figure 3.20 Current command value write processing with 100us interrupt

4. Control operation with CPU processing

4.1 Specifications

In this chapter, the CPU sets the current command value and performs feedback processing (interrupt).

Table 4.1 lists the specifications of motor control. Table 4.2 lists settings of individual modules. Figure 4.1 shows the configuration of a 2-motor control system (using the RH850/U2B6). Figure 4.2 shows the flow of control processes involving processing by the CPU.

Table 4.1 Motor Control Specifications

Item	Specification
Output waveform	Complemented 3-phase PWM waveform
Carrier frequency	10 kHz (100 µsec/cycle)
Control method	180-degree excitation drive method
Active level	High
Short-circuit prevention time (dead time)	4 µsec
Timing of updating compare register and carrier frequency	Trough of the carrier wave
Timing of starting A/D conversion	Trough of the carrier wave
Interrupt	Used (WAIT transition interrupt, 100usec interrupt)
Motor current value	Obtain the values of phase-U and phase-V currents by the ADCK0
Motor angle information	Obtain the angle information from resolver by the RDC3AL.

Table 4.2 Module Settings

Peripheral Function	Setting
RDC3AL	Refer to Table 3.2
PIC	Refer to Table 3.2
ADCK	Refer to Table 3.2
TSG3	Refer to Table 3.2
EMU3S	• Starting the input IP automatically with AD end trigger • Starting PWM IP with software trigger from CPU • Using the current value from the ADCK, angle value and angular velocity value from the RDC3AL for computation • Starting A/D conversion in the timing of a trough of the carrier wave • 1usec delay in trigger output for A/D conversion and R/D conversion • Selectable non-interference control/voltage compensation

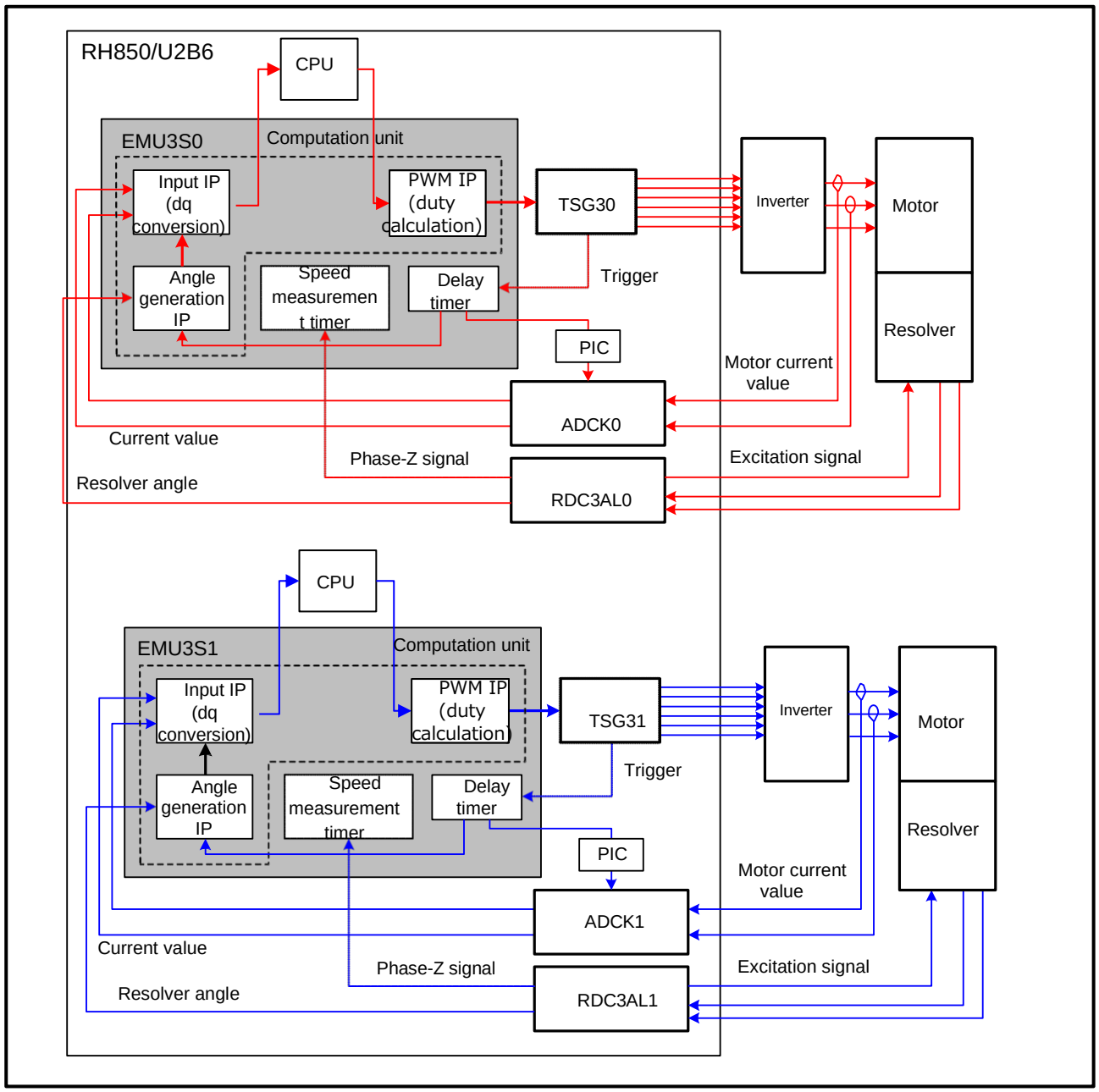


Figure 4.1 Configuration of 2-Motor Control System (using the RH850/U2B6)

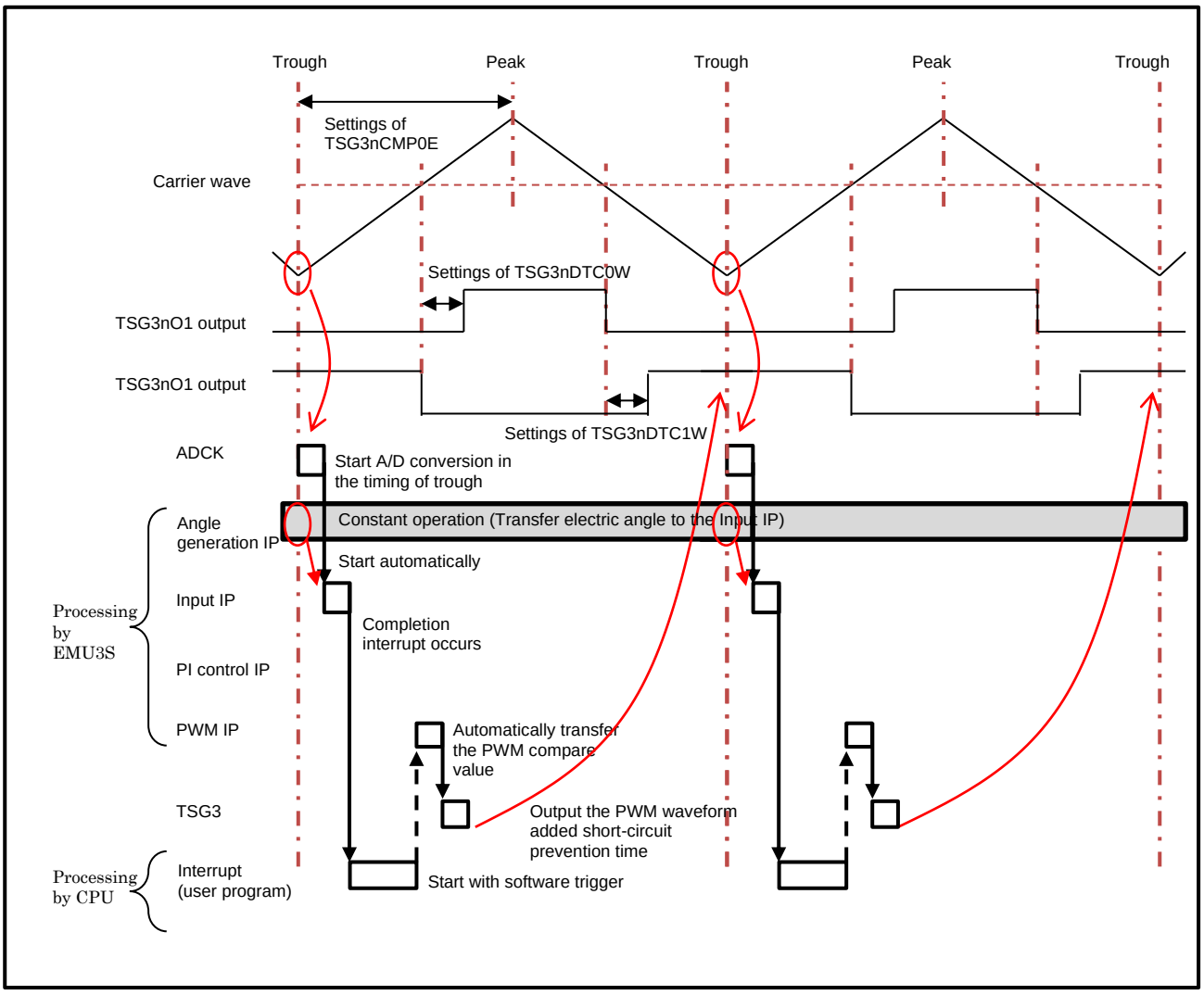


Figure 4.2 Flow of Control Involving CPU Processing

4.2 Operation Check Conditions

Refer to “3.2 Operation Check Conditions”

4.3 Pins

Refer to “3.3 Pins”

4.4 Overview of Operation

The following describes motor control operation:

(1) When the EMU3S 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 ADCK0 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 EMU3S computation unit converts the resolver angle obtained from the RDC3AL 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 EMU3S, 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, interrupt is occurred and EMU3S is put in the WAIT state. Feedback calculation is performed by CPU processing. The result is reflected in the PWM IP register. When the reflection is completed, the PWM IP is booted from the software.

(4) The PWM IP calculates the PWM compare values (duty values) for phases U, V, and W based on d-axis and q-axis voltages.

(5) 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) Current command values are set for EMU3S in OSTM interrupt function. Thereafter, a rotation command is given to the motor to be controlled.

4.4.1 R/D Converter 3A (RDC3AL)

Refer to “3.4.1 R/D Converter 3AL (RDC3AL)”

4.4.2 Peripheral Interconnection (PIC)

Refer to “3.4.2 Peripheral Interconnection (PIC)”

4.4.3 A/D Converter (ADCK)

Refer to “3.4.3 A/D Converter (ADCK)”

4.4.4 Motor Control Timer 3 (TSG3)

Refer to “3.4.4 Motor Control Timer 3 (TSG3)”

4.4.5 Enhanced Motor Control Unit 3S (EMU3S)

4.4.5.1 EMU3S Common Unit

The EMU3S performs vector control based on motor current values and angle information, and generates PWM compare values. In this chapter, since the feedback processing is performed by the CPU instead of the PI control IP, it transits to WAIT after the input IP is completed, generates a CPU interrupt, and starts the PWM IP from the software after the feedback processing is completed.

[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 = 0 (Starting PI control IP with software trigger from CPU)
- PWMIPTRG bit in EMU3nIPTRG register = 0 (Starting PWM IP with software trigger from CPU)
- SMVAN bit in EMU3nSMCTR register = 1 (Starting A/D converter in the timing of carrier trough)
- ADDATA bit in EMU3nADDCNT register = 80 (1usec delay time in A/D conversion)
- RDDATA bit in EMU3nRDDCNT register = 80 (1usec delay time in the output of R/D conversion trigger)
- EMU3nFUNCFLGRPA0 register = 00C0H (Transition to IDLE after input3 end)
- EMU3nFUNCWAITGRPA register = 0301H (Starting from func(pwm1) of PWM IP)
- EMU3nINT0 registers = 0080 0000H (Interrupt at WAIT transition)
- EMU3nINT1 to EMU3nINT7 registers = 0000 0000H (Disabling interrupt)
- INTC2EIC38 registers = 0040H (Enable interrupt processing, interrupt level 0)
- INTC2EIC39 to EIC45 registers = 00CFH (Disabling interrupt processing)

4.4.5.2 Angle Generation IP

Refer to “3.4.5.2 Angle Generation IP”

4.4.5.3 Input IP

Refer to “3.4.5.3 Input IP”

4.4.5.4 PI Control IP

Nothing is done in this operation example. Interrupt processing is performed instead.

4.4.5.5 PWM IP

After the interrupt processing is completed, the PWM IP is started by the trigger from the software. The PWM IP uses the d-axis voltage and q-axis voltage written to the register to calculate by interrupt processing instead of the PI control IP. Obtain the U / V / W phase output voltage from that voltage value. Next, calculate the duty ratio and compare register value of each phase in the PWM waveform output.

The subsequent processing is the same as "3.4.5.5 PWM IP", so refer to that.

4.4.6 File Configuration

Table 4.3 shows the changes made from "3.4.6 File Configuration". Files automatically generated in the integrated development environment are excluded.

Table 4.3 File Configuration

File name	Descript	Remarks
user_int.c	Interrupt processing by Wait transition	

4.4.7 Functions

Table 4.4 shows the changes made from "3.4.7 Functions".

Table 4.4 Functions

Function Name	Description
int_shift2wait	Interrupt when transitioning from input IP to WAIT

4.4.8 Specifications of Functions

The function specifications of the sample code that have changed from "3.4.8 Specifications of Functions" are shown below.

int_shift2wait

Outline: Reads the result of the input IP, performs feedback processing, writes to the register of PWM IP, and starts PWM IP.
Header: iodefine.h
Declaration: void int_shift2wait (void)
Description: Feedback processing
Argument: None
Return value: None
Remarks:

4.4.9 Flowcharts

The flowcharts that have changed from "3.4.9 Flowcharts " are shown in Figure 4.3 to Figure 4.5.

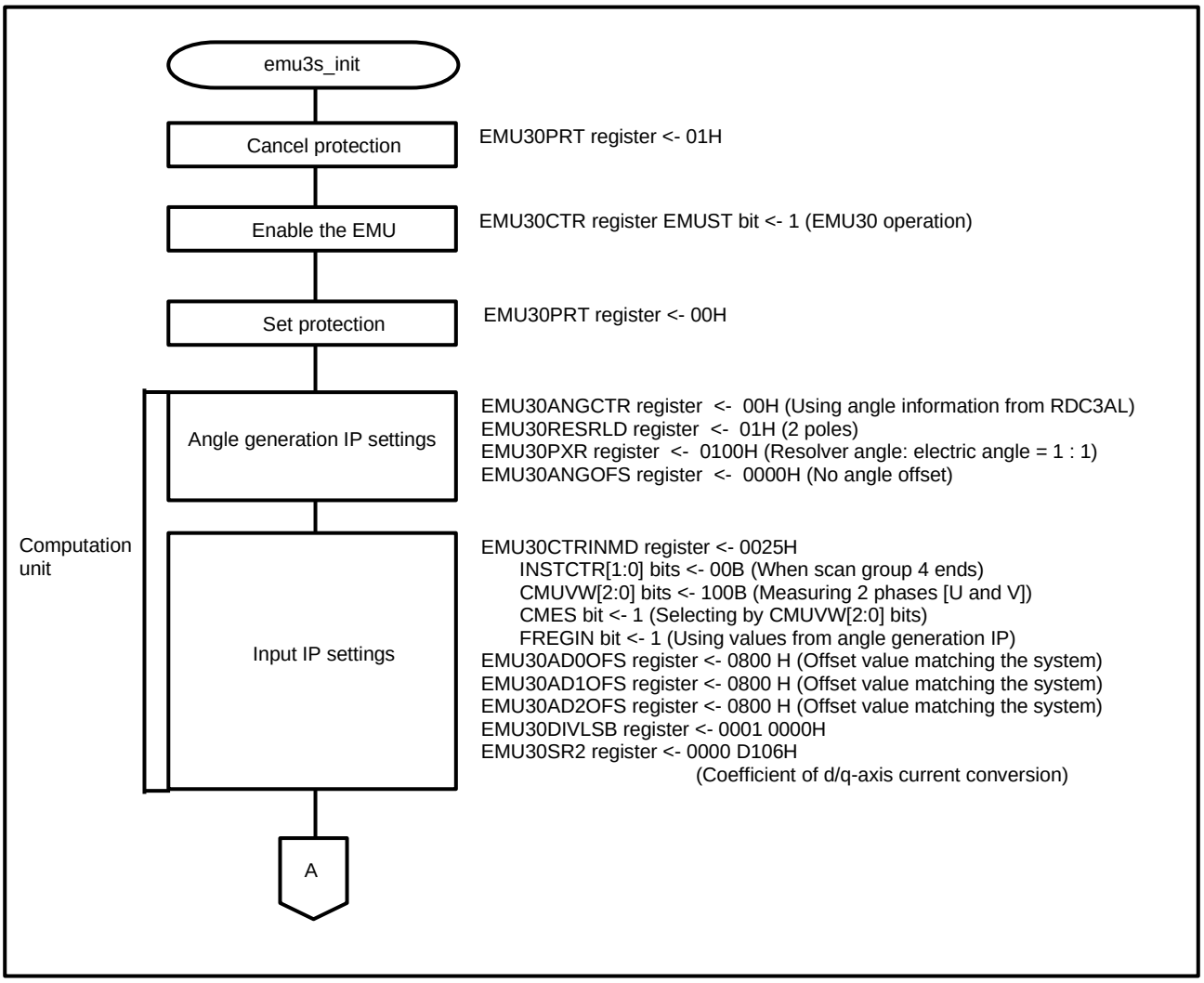


Figure 4.3 EMU3S Initialization (1)

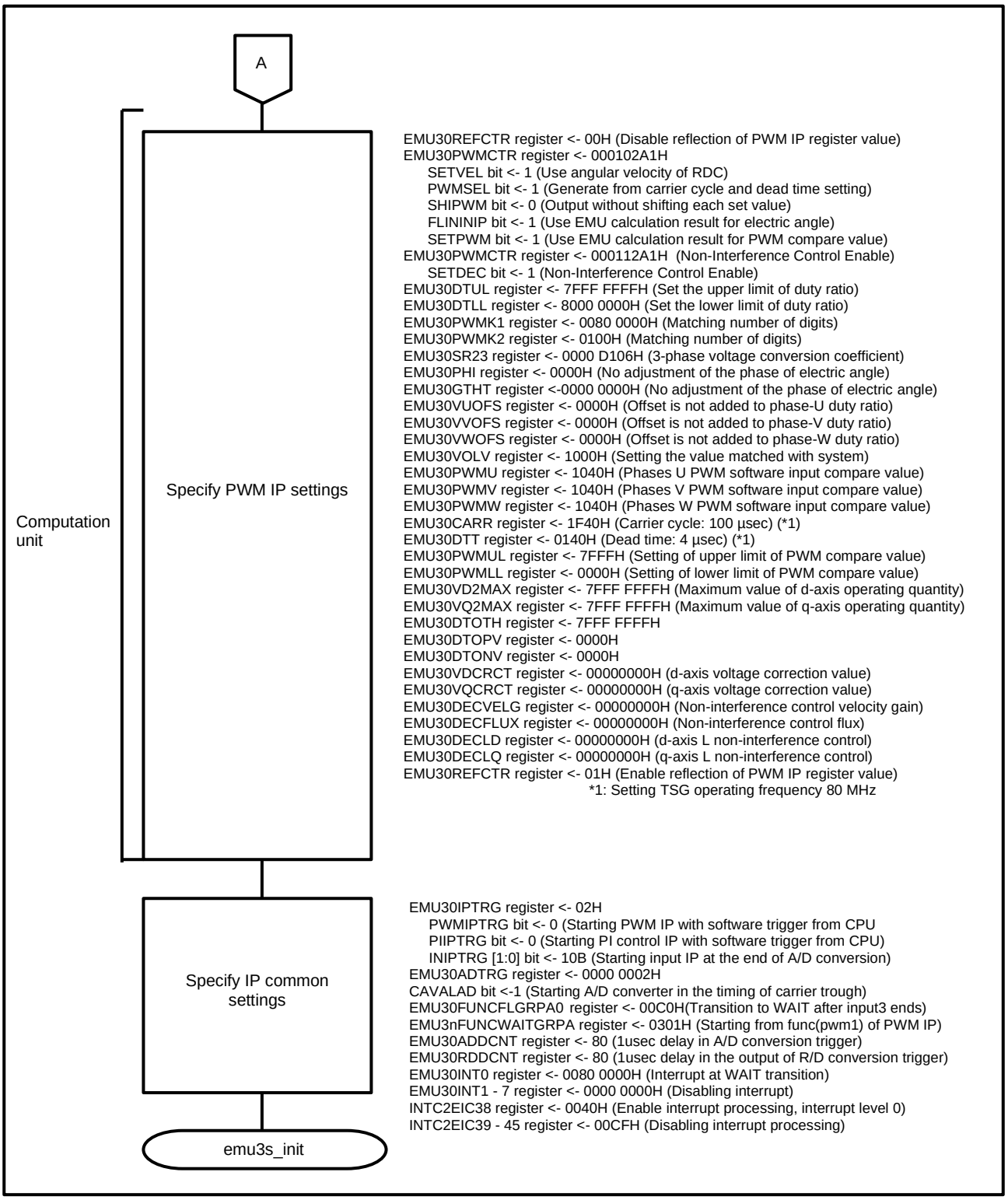


Figure 4.4 EMU3S Initialization (2)

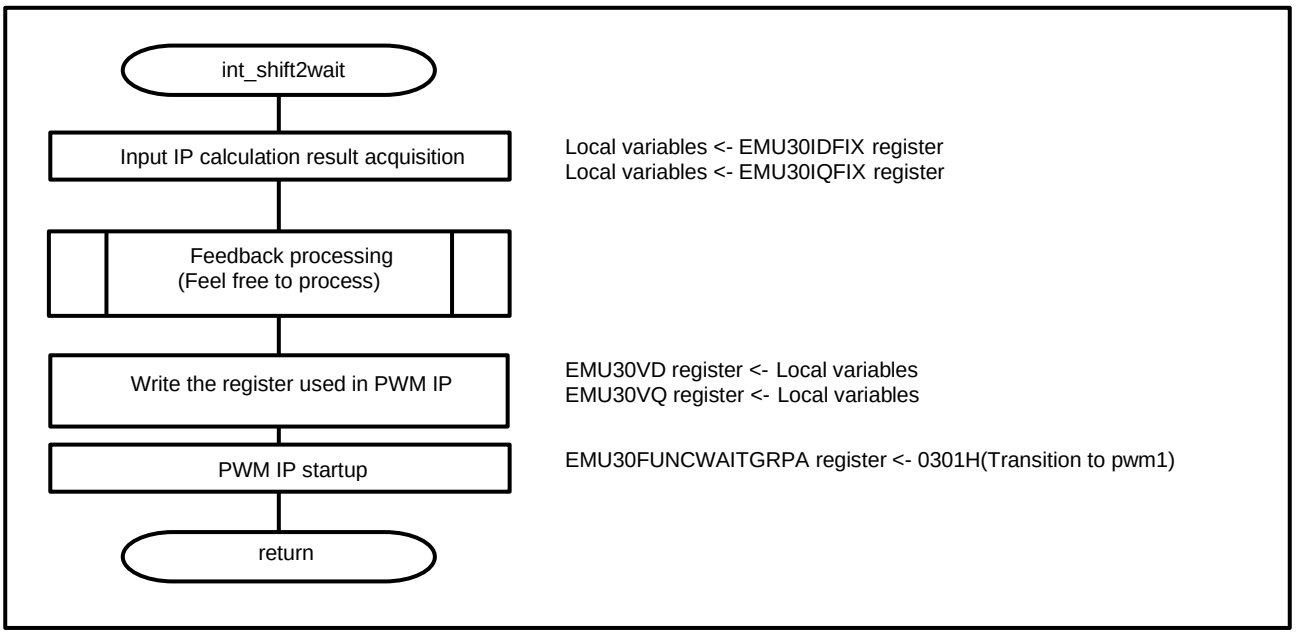


Figure 4.5 Interrupt processing of WAIT transition

5. Rectangular wave output operation

5.1 Specifications

In this chapter, rectangular wave output processing is performed according to the angle.

Table 5.1 lists the specifications of motor control. Table 5.2 lists settings of individual modules. Figure 5.1 shows the configuration of a 2-motor control system (using the RH850/U2B6). Figure 5.2 shows the rectangular wave output image diagram.

Table 5.1 Motor Control Specifications

Item	Specification
Output waveform	Rectangular wave
Control method	180-degree excitation drive method
Active level	Active high
Short-circuit prevention time (dead time)	4 µsec
Compare register update timing	Electric angle 0 degrees
Timing of starting A/D conversion	Electric angle 0 degrees
Interrupt	Used (U-phase compare match interrupt)
Motor current value	obtains values of phase-U and phase-V currents with the ADCK0.
Motor angle information	obtains angle information from resolver with the RDC3AL.

Table 5.2 Module Settings

Peripheral Function	Setting
RDC3AL	Refer to Table 3.2
PIC	Refer to Table 3.2
ADCK	Refer to Table 3.2
TSG3	Output rectangular wave from EMU3S
EMU3S	• Starting the input IP automatically with AD end trigger • Starting PWM IP with software trigger from CPU • Starting A/D conversion in the timing of electric angle 0 degrees • 1usec delay in trigger output for A/D conversion and R/D conversion

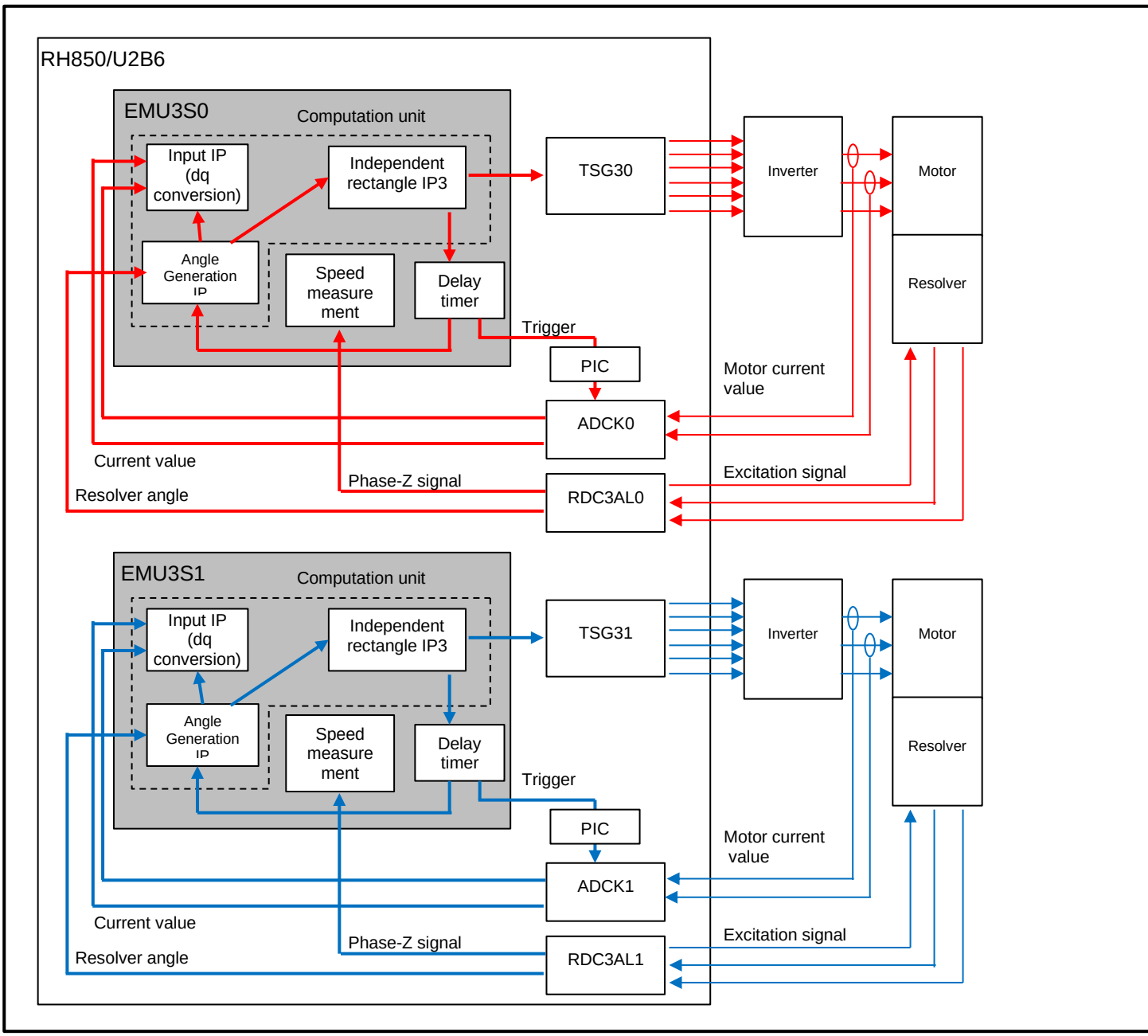


Figure 5.1 Configuration of 2-Motor Control System (using the RH850/U2B6)

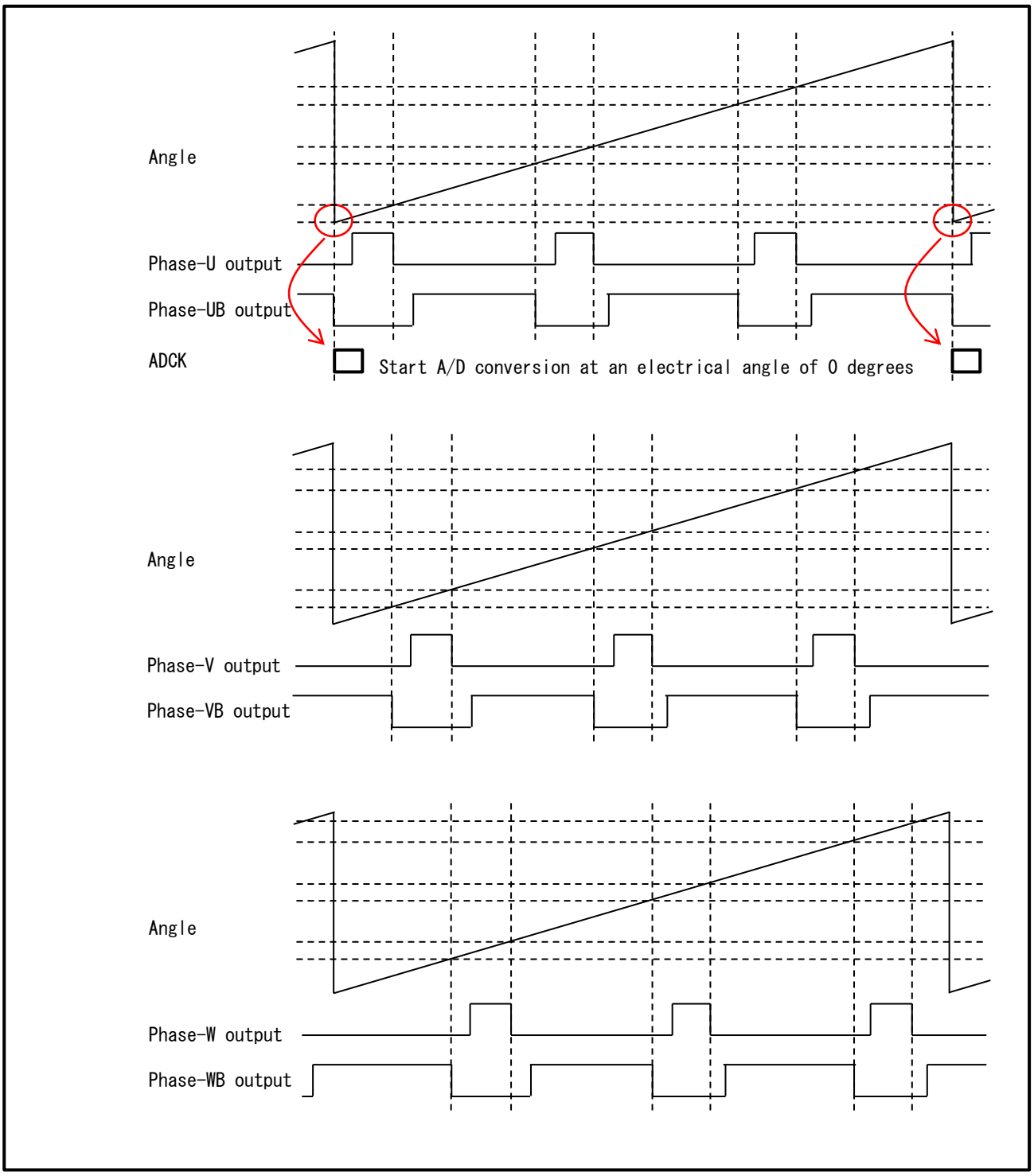


Figure 5.2 Rectangle wave output diagram

5.2 Operation Check Conditions

Refer to “3.2 Operation Check Conditions”

5.3 Pins

Refer to “3.3 Pins”

5.4 Overview of Operation

The following describes motor control operation:

(1) When the EMU3S inputs a trigger signal to the A/D converter via the PIC in the timing of electric angle 0 degrees, the A/D converter performs A/D conversion of the virtual channels 0 to 3 in ADCK0 group 4 to obtain values of the phase-U and phase-V currents of the motor.

(2) The angle generation IP of the EMU3S computation unit converts the resolver angle obtained from the RDC3AL into the electric angle of the motor, and transfers the electric angle to the input IP and Independent rectangle IP3 of the computation unit.

(3) After completed the A/D conversion, conversion results are stored in registers of the EMU3S, 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. After the input IP ends, it transitions to the IDLE state.

(4) The independent rectangle IP3 operates when the electrical angle is changed. When it operates, it compares the set compare value with the electric angle. Update the output according to the comparison result.

(5) The output updated by the independent rectangle IP3 is transferred to TSG3. Then, a 3-phase PWM waveform is output from corresponding TSG3 pins.

5.4.1 R/D Converter 3A (RDC3AL)

Refer to “3.4.1 R/D Converter 3AL (RDC3AL)”

5.4.2 Peripheral Interconnection (PIC)

Refer to “3.4.2 Peripheral Interconnection (PIC)”

5.4.3 A/D Converter (ADCK)

Refer to “3.4.3 A/D Converter (ADCK)”

5.4.4 Motor Control Timer 3 (TSG3)

TSG3 adds a dead time to the rectangular wave output transferred from EMU3S and outputs it. It is possible by setting TSG3 option register 2.

5.4.5 Enhanced Motor Control Unit 3S (EMU3S)

5.4.5.1 EMU3S Common Unit

EMU3S compares the compare value with the electrical angle and generates a rectangular wave. In this chapter, an interrupt and an A/D conversion trigger are generated at an electrical angle of 0 degrees. Transitions to IDLE after completed the input IP.

[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 = 0 (Starting PI control IP with software trigger from CPU)
- PWMIPTRG bit in EMU3nIPTRG register = 0 (Starting PWM IP with software trigger from CPU)
- ADDATA bit in EMU3nADDCNT register = 80 (1usec delay time in A/D conversion)
- RDDATA bit in EMU3nRDDCNT register = 80 (1usec delay time in the output of R/D conversion trigger)
- EMU3nFUNCFLGRPA0 register = 0040H (Transition to IDLE after input3 ends)
- EMU3nINT0 registers = 0008 0000H (Interrupt at WAIT transition)
- EMU3nINT1 to EMU3nINT7 registers = 0000 0000H (Disabling interrupt)
- INTC2EIC38 registers = 0040H (Enable interrupt processing, interrupt level 0)
- INTC2EIC39 to EIC45 registers = 00CFH (Disabling interrupt processing)

5.4.5.2 Angle Generation IP

Refer to “3.4.5.2 Angle Generation IP”

5.4.5.3 Input IP

Refer to “3.4.5.3 Input IP”

After the input IP is completed, it will transition to IDLE.

5.4.5.4 PI Control IP

Nothing is done in this operation example. Interrupt processing is performed instead.

5.4.5.5 PWM IP

Nothing is done in this operation example. Interrupt processing is performed instead.

5.4.5.6 Independent rectangular IP3

Compare the compare value with the electric angle calculated by the angle generation IP. Update the output according to the comparison result. The relationship between the output and the compare value in the independent rectangle IP3 is shown in Figure 5.3. The IP3CMPn register determines the end angle of area n and the output in area n.

At this time, maintain the relationship of area n ≤ area n + 1 for the compare value of each region. Also, it is necessary to specify the number of areas used in the IR3VALN register. In the case of Figure 5.3, it is necessary to set EMU30.IR3VALN.BIT.UVL = 6 (using 6 areas from 0 to 5).

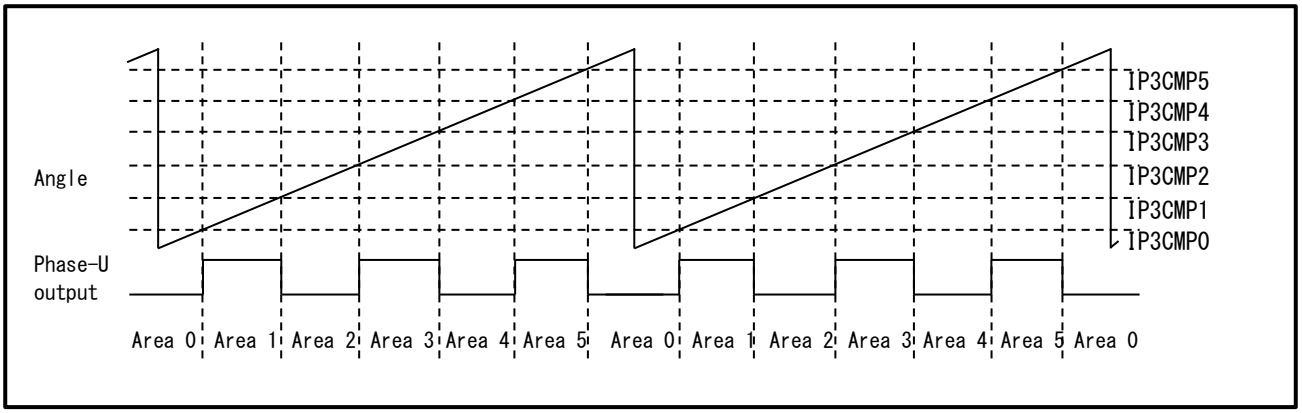


Figure 5.3 Relationship between output and compare value (U phase)

In addition, the compare value for the independent rectangle IP3 can be set to 96 (32 x 3 phases). On the other hand, there are 32 IP3CMPn registers that perform Read / Write, and it is possible to read / write 96 IP3CMPn registers by selecting which phase to perform Read / Write.

Select the write destination in the IR3CMPWRMD register. Select the Read source in the IR3CMPRDMD register. If the batch transfer mode is selected in the CMP bit of the IR3TRSMODE register, generate the trigger of a batch transfer in the IR3TRG register to enable write data.

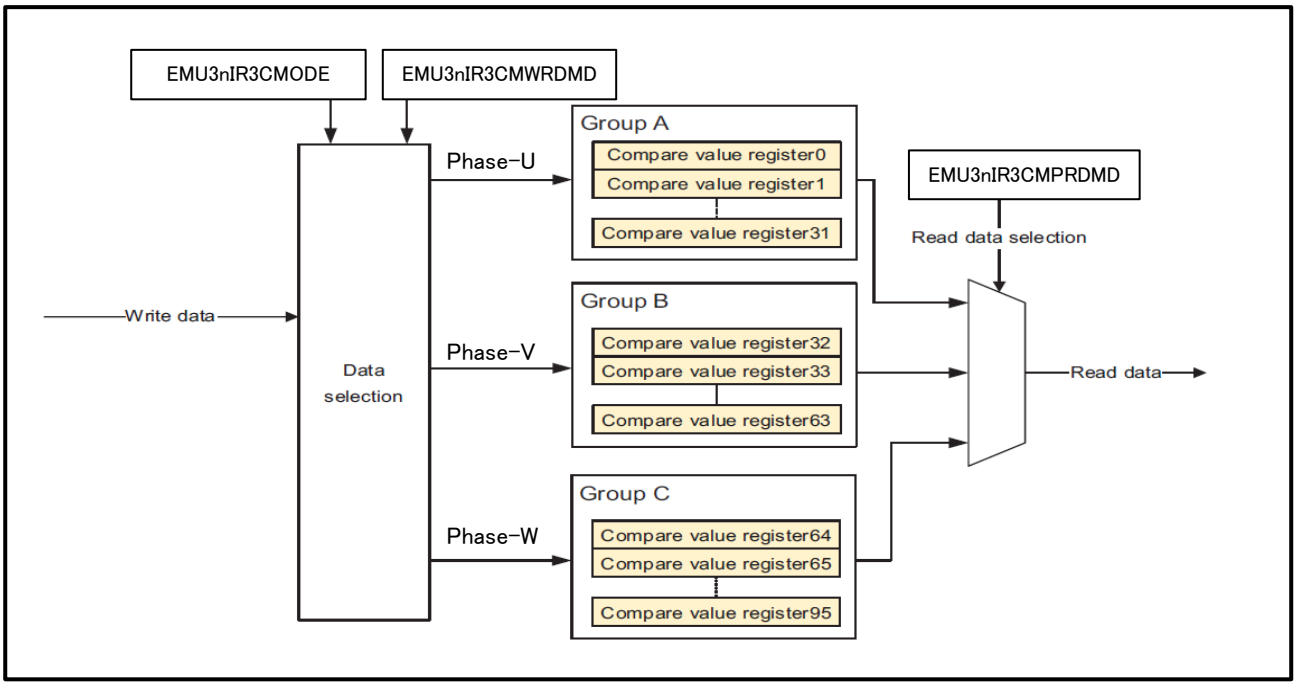


Figure 5.4 Read / Write by IP3CMPn register

In this chapter, set the initial compare values as shown in Table 5.3 to Table 5.5 using the IP3CMPn registers.

Table 5.3 Phase-U initial setting

Phase-U	Set value	remarks	Phase-U	Set value	remarks
CMP0	0x0000	0xF40 to 0x000, Low output	CMP16	0x0800	0x740 to 0x800, Low output
CMP1	0x8040	0x000 to 0x040, High output	CMP17	0x8840	0x740 to 0x840, High output
CMP2	0x0100	0x040 to 0x100, Low output	CMP18	0x0900	0x840 to 0x900, Low output
CMP3	0x8140	0x100 to 0x140, High output	CMP19	0x8940	0x900 to 0x940, High output
CMP4	0x0200	0x140 to 0x200, Low output	CMP20	0x0A00	0x940 to 0xA00, Low output
CMP5	0x8240	0x200 to 0x240, High output	CMP21	0x8A40	0xA00 to 0xA40, High output
CMP6	0x0300	0x240 to 0x300, Low output	CMP22	0x0B00	0xA40 to 0xB00, Low output
CMP7	0x8340	0x300 to 0x340, High output	CMP23	0x8B40	0xB00 to 0xB40, High output
CMP8	0x0400	0x340 to 0x400, Low output	CMP24	0x0C00	0xB40 to 0xC00, Low output
CMP9	0x8440	0x400 to 0x440, High output	CMP25	0x8C40	0xC00 to 0xC40, High output
CMP10	0x0500	0x440 to 0x500, Low output	CMP26	0x0D00	0xC40 to 0xD00, Low output
CMP11	0x8540	0x500 to 0x540, High output	CMP27	0x8D40	0xD00 to 0xD40, High output
CMP12	0x0600	0x540 to 0x600, Low output	CMP28	0x0E00	0xD40 to 0xE00, Low output
CMP13	0x8640	0x600 to 0x640, High output	CMP29	0x8E40	0xE00 to 0xE40, High output
CMP14	0x0700	0x640 to 0x700, Low output	CMP30	0x0F00	0xE40 to 0xF00, Low output
CMP15	0x8740	0x700 to 0x740, High output	CMP31	0x8F40	0xF00 to 0xF40, High output

Table 5.4 Phase-V initial setting

Phase-V	Set value	remarks	Phase-V	Set value	remarks
CMP0	0x0040	0xF80 to 0x040, Low output	CMP16	0x0840	0x780 to 0x840, Low output
CMP1	0x8080	0x040 to 0x080, High output	CMP17	0x8880	0x840 to 0x880, High output
CMP2	0x0140	0x080 to 0x140, Low output	CMP18	0x0940	0x880 to 0x940, Low output
CMP3	0x8180	0x140 to 0x180, High output	CMP19	0x8980	0x940 to 0x980, High output
CMP4	0x0240	0x180 to 0x240, Low output	CMP20	0x0A40	0x980 to 0xA40, Low output
CMP5	0x8280	0x240 to 0x280, High output	CMP21	0x8A80	0xA40 to 0xA80, High output
CMP6	0x0340	0x280 to 0x340, Low output	CMP22	0x0B40	0xA80 to 0xB40, Low output
CMP7	0x8380	0x340 to 0x380, High output	CMP23	0x8B80	0xB40 to 0xB80, High output
CMP8	0x0440	0x380 to 0x440, Low output	CMP24	0x0C40	0xB80 to 0xC40, Low output
CMP9	0x8480	0x440 to 0x480, High output	CMP25	0x8C80	0xC40 to 0xC80, High output
CMP10	0x0540	0x480 to 0x540, Low output	CMP26	0x0D40	0xC80 to 0xD40, Low output
CMP11	0x8580	0x540 to 0x580, High output	CMP27	0x8D80	0xD40 to 0xD80, High output
CMP12	0x0640	0x580 to 0x640, Low output	CMP28	0x0E40	0xD80 to 0xE40, Low output
CMP13	0x8680	0x640 to 0x680, High output	CMP29	0x8E80	0xE40 to 0xE80, High output
CMP14	0x0740	0x680 to 0x740, Low output	CMP30	0x0F40	0xE80 to 0xF40, Low output
CMP15	0x8780	0x740 to 0x780, High output	CMP31	0x8F80	0xF40 to 0xF80, High output

Table 5.5 Phase-W initial setting

Phase-W	Set value	remarks	Phase-W	Set value	remarks
CMP0	0x0080	0xFC0 to 0x080, Low output	CMP16	0x0880	0x7C0 to 0x880, Low output
CMP1	0x80C0	0x080 to 0x0C0, High output	CMP17	0x88C0	0x880 to 0x8C0, High output
CMP2	0x0180	0x0C0 to 0x180, Low output	CMP18	0x0980	0x8C0 to 0x980, Low output
CMP3	0x81C0	0x180 to 0x1C0, High output	CMP19	0x89C0	0x980 to 0x9C0, High output
CMP4	0x0280	0x1C0 to 0x280, Low output	CMP20	0x0A80	0x9C0 to 0xA80, Low output
CMP5	0x82C0	0x280 to 0x2C0, High output	CMP21	0x8AC0	0xA80 to 0xAC0, High output
CMP6	0x0380	0x2C0 to 0x380, Low output	CMP22	0x0B80	0xAC0 to 0xB80, Low output
CMP7	0x83C0	0x380 to 0x3C0, High output	CMP23	0x8BC0	0xB80 to 0xBC0, High output
CMP8	0x0480	0x3C0 to 0x480, Low output	CMP24	0x0C80	0xBC0 to 0xC80, Low output
CMP9	0x84C0	0x480 to 0x4C0, High output	CMP25	0x8CC0	0xC80 to 0xCC0, High output
CMP10	0x0580	0x4C0 to 0x580, Low output	CMP26	0x0D80	0xCC0 to 0xD80, Low output
CMP11	0x85C0	0x580 to 0x5C0, High output	CMP27	0x8DC0	0xD80 to 0xDC0, High output
CMP12	0x0680	0x5C0 to 0x680, Low output	CMP28	0x0E80	0xDC0 to 0xE80, Low output
CMP13	0x86C0	0x680 to 0x6C0, High output	CMP29	0x8EC0	0xE80 to 0xEC0, High output
CMP14	0x0780	0x6C0 to 0x780, Low output	CMP30	0x0F80	0xEC0 to 0xF80, Low output
CMP15	0x87C0	0x780 to 0x7C0, High output	CMP31	0x8FC0	0xF80 to 0xFC0, High output

【Setting value of related-register】

- CMD bit in EMU3nIR3CMODE register = 1 (3-phase independent mode)
- MODE bit in EMU3nIR3CMODE register = 00b (Sawtooth mode)
- EMU3nIR3RDI register = 00H (Positive rotation)
- EMU3nIR3TRSMODE register = 00H (Batch transfer mode)
- EMU3nIR3INTCLEAR register = 07H (Clear interrupt status)
- LOE bit in EMU3nIR3INTEN register = 1 (Enable interrupts below 180 degrees)
- EN bit in EMU3nIR3INTEN register = 00b (Enable interrupts only in phase-U)
- INT0 bit in EMU3nIR3INT0 register = 1 (Interrupt is generated when the phase-U area 0 compare match.)
- EMU3nIR3INT1 register = 00000000H (In phase-V, no interrupt generation by compare match.)
- EMU3nIR3INT2 register = 00000000H (In phase-W, no interrupt generation by compare match.)
- EMU3nIR3ADCCLEAR register = 07H (Clear A / D conversion trigger status)
- LOE bit in EMU3nIR3ADCEN register = 1 (Enable A / D conversion trigger below 180 degrees)
- EN bit in EMU3nIR3ADCEN register = 00b
  (A/D conversion trigger is generated by the compare match of phase-U.)
- ADI0 bit in EMU3nIR3ADC0 register = 1
  (A/D conversion trigger is generated by the compare match of U phase area 0.)
- EMU3nIR3ADC1 register = 00000000H
  (In phase-V, A/D conversion trigger is not generated by compare match.)
- EMU3nIR3ADC2 register = 00000000H
  (In phase-W, A/D conversion trigger is not generated by compare match.)

- EMU3nIR3COFSALL register = 0000H (No electrical angle offset)
- EMU3nIR3COFSU register = 0000H (No electrical angle offset for phase-U)
- EMU3nIR3COFSV register = 0000H (No electrical angle offset for phase-V)
- EMU3nIR3COFSW register = 0000H (No electrical angle offset for phase-W)
- UVL bit in EMU3nIR3VALN register = 32 (Use 32 phase-U compare values)
- VVL bit in EMU3nIR3VALN register = 32 (Use 32 phase-V compare values)
- WVL bit in EMU3nIR3VALN register = 32 (Use 32 phase-W compare values)
- TRG bit in EMU3nIR3TRG register = 1 (Reflect the U / V / W phase compare register)
- EN bit in EMU3nIR3CCTR register = 1 (Enable independent rectangle IP3)

5.4.6 File Configuration

Table 5.6 shows the changes made from "3.4.6 File Configuration". Files automatically generated in the integrated development environment are excluded.

Table 5.6 File Configuration

File name	Descript	Remarks
emu3s.c	Initial setting of independent rectangle IP3	
user_int.c	Phase-U compare match interrupt processing	

5.4.7 Functions

Table 5.7 shows the changes made from "3.4.7 Functions".

Table 5.7 Functions

Function	Description
emu3s_init	Initial setting of angle generation IP and independent rectangle IP3
emu3s_rec_angle_init	Set the compare value of the independent rectangle IP3
emu3s_rec_angle_update	Reflect the compare value of independent rectangle IP3 in the register
emu3s_rec_angle_update_u	Set the phase-U compare value of the independent rectangle IP3
emu3s_rec_angle_update_v	Set the phase-V compare value of the independent rectangle IP3
emu3s_rec_angle_update_w	Set the phase-W compare value of the independent rectangle IP3
emu3s_rec_angle_reflect	Reflects the compare value of the independent rectangle IP3
int_rec_ip3_compare	Interrupt when phase-U compare matches at an electrical angle of 0 degrees

5.4.8 Specifications of Functions

The function specifications of the sample code that have changed from "3.4.8 Specifications of Functions" are shown below.

emu3s_init

Outline: Initial setting of angle generation IP and independent rectangle IP3
Header: iodefine.h
Declaration: void emu3s_init(void)
Description: Initialize the EMU3S (angle generation IP and independent rectangle IP3).
Argument: None
Return value: None
Remarks:

emu3s_rec_angle_init

Outline: Set the compare value of the independent rectangle IP3
Header: None
Declaration: void emu3s_rec_angle_init (void)
Description: Set the values in the compare value array used by the independent rectangle IP3.
Argument: None
Return value: None
Remarks:

emu3s_rec_angle_update

Outline: Reflect the compare value of independent rectangle IP3 in the register
Header: None
Declaration: void emu3s_rec_angle_update (void)
Description: Reflect the compare value of the independent rectangle IP3 to the IP3CMP register.
Argument: Compare value array
Return value: None
Remarks:

emu3s_rec_angle_update_u

Outline: Set the phase-U compare value of the independent rectangle IP3
Header: iodefine.h
Declaration: void emu3s_rec_angle_update_u (void)
Description: Change the setting to write phase-U data to the IP3CMP register and call emu3s_rec_angle_update.
Argument: None
Return value: None
Remarks:

emu3s_rec_angle_update_v

Outline: Set the phase-V compare value of the independent rectangle IP3
Header: iodefine.h
Declaration: void emu3s_rec_angle_update_v (void)
Description: Change the setting to write phase-V data to the IP3CMP register and call emu3s_rec_angle_update.
Argument: None
Return value: None
Remarks:

emu3s_rec_angle_update_w

Outline: Set the phase-W compare value of the independent rectangle IP3
Header: iodefine.h
Declaration: void emu3s_rec_angle_update_w (void)
Description: Change the setting to write phase-W data to the IP3CMP register and call emu3s_rec_angle_update.
Argument: None
Return value: None
Remarks:

emu3s_rec_angle_reflect

Outline: Reflects the compare value of the independent rectangle IP3
Header: iodefine.h
Declaration: void emu3s_rec_angle_reflect (void)
Description: Reflect the value written in the IP3CMP register to the internal holding register.
Argument: None
Return value: None
Remarks:

int_rec_ip3_compare

Outline: Interrupt when phase-U compare matches at an electrical angle of 0 degrees
Header: iodefine.h
Declaration: void int_rec_ip3_compare (void)
Description: Update the IP3CMP register of the independent rectangle IP3 by interrupting in phase-U.
Argument: None
Return value: None
Remarks:

5.4.9 Flowcharts

The flowcharts that have changed from "3.4.9 Flowcharts " are shown in Figure 5.5 to Figure 5.12.

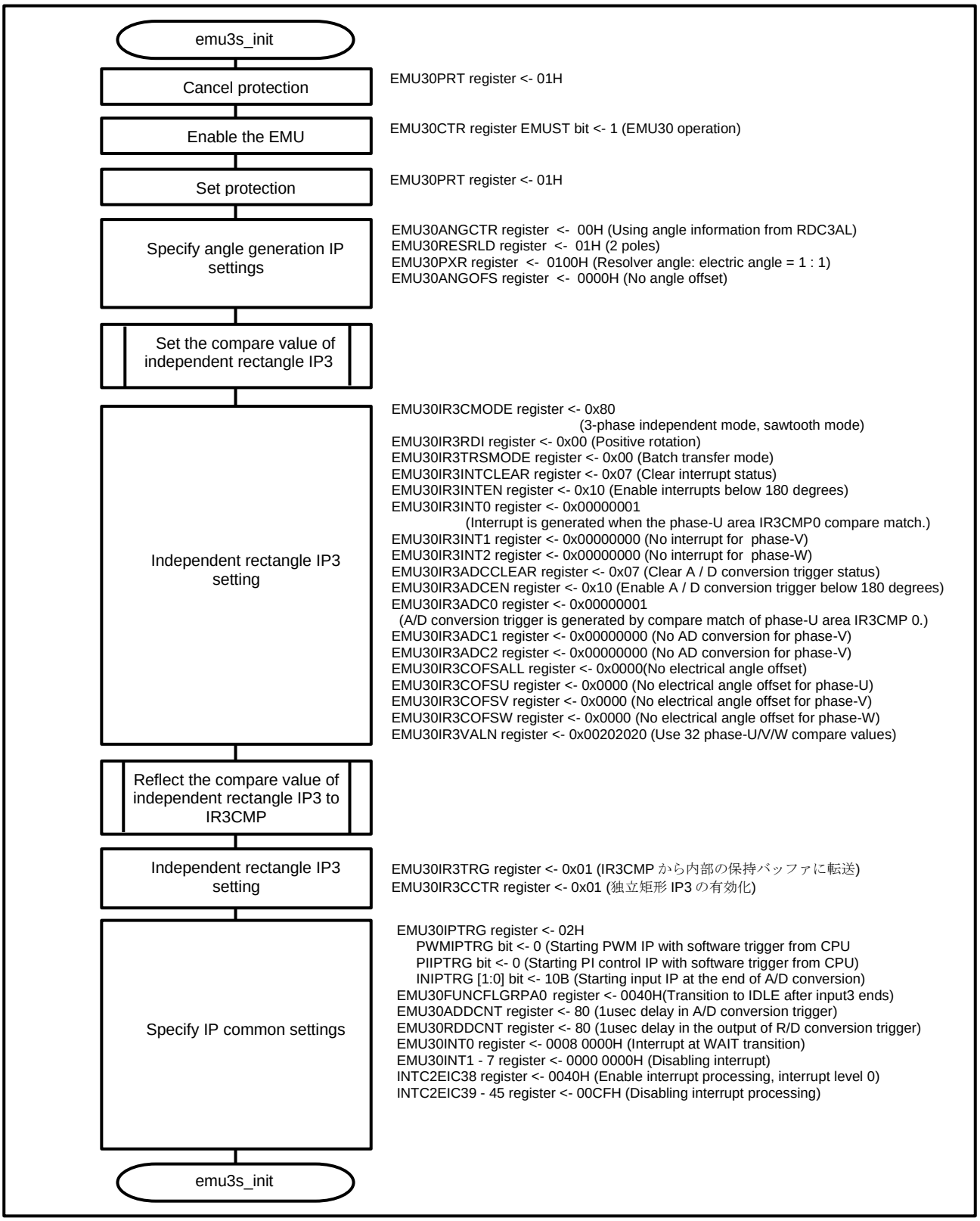


Figure 5.5 EMU3S Initialization

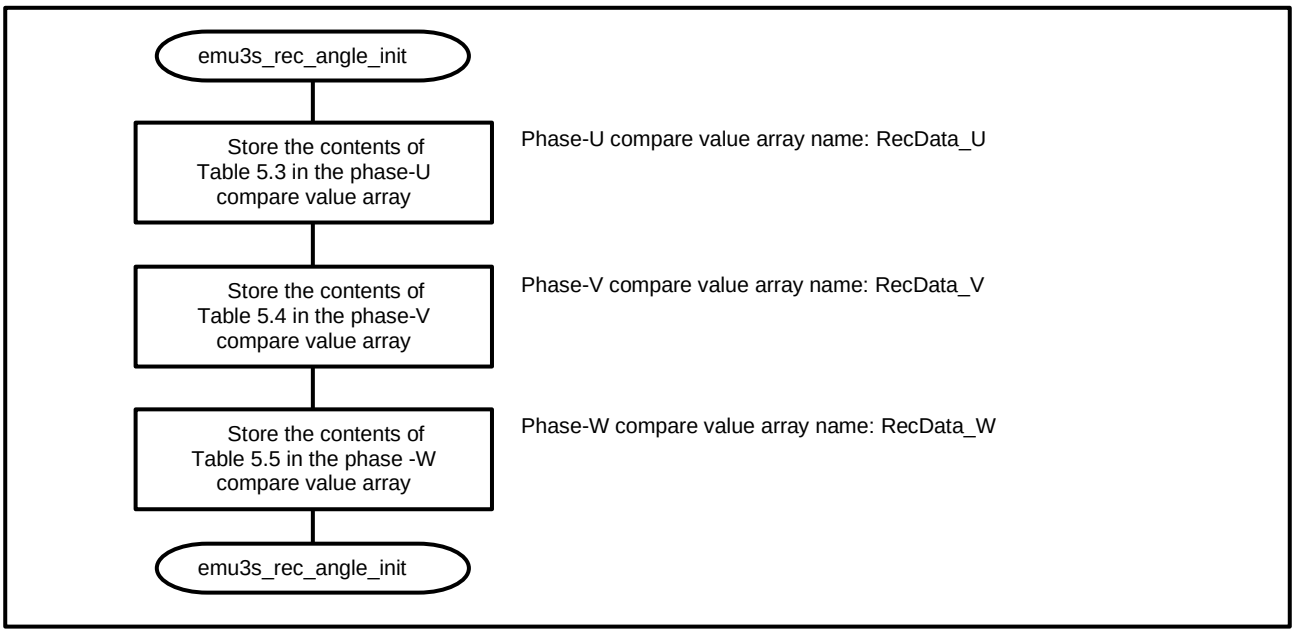


Figure 5.6 Compar value array settings

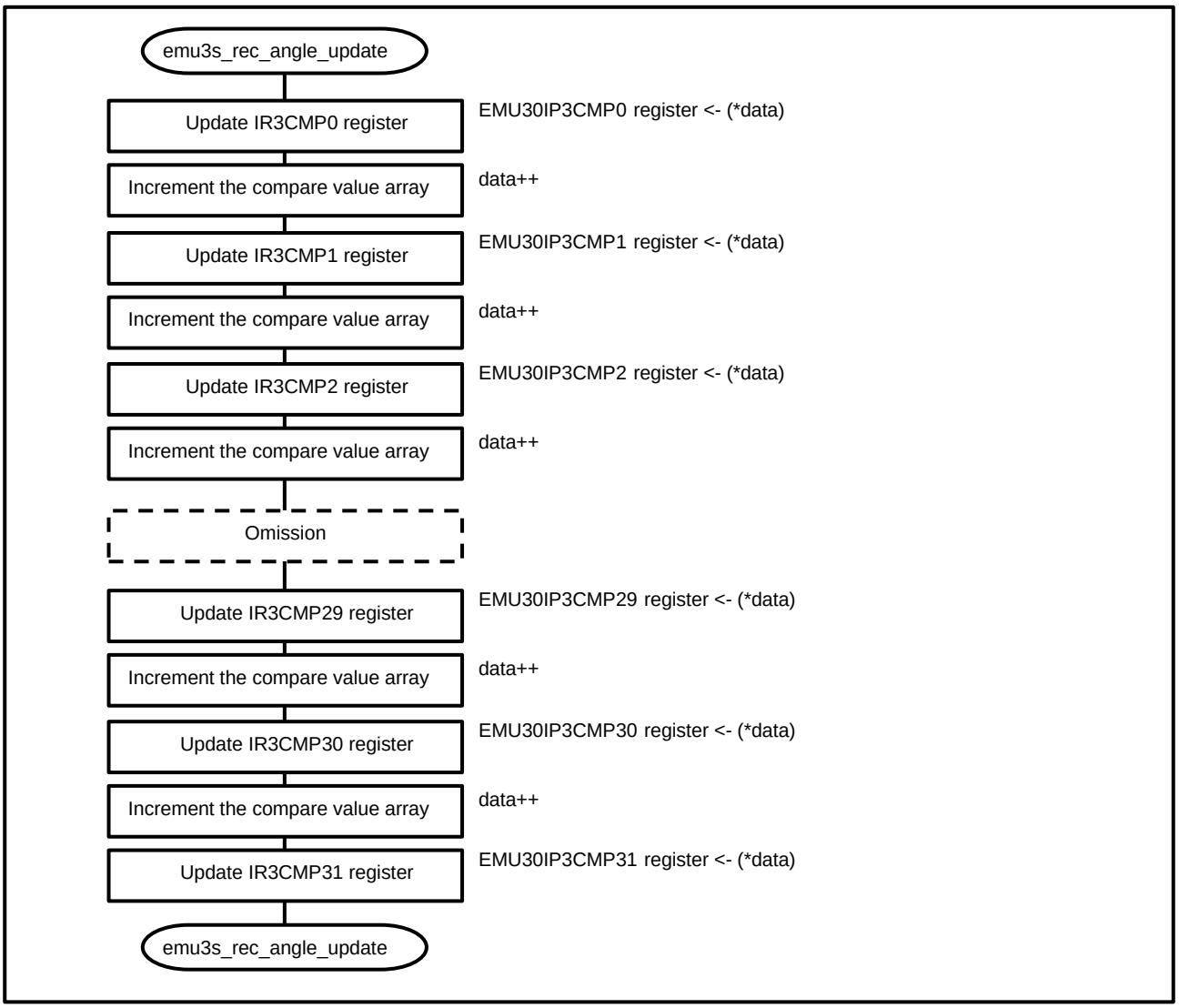


Figure 5.7 IR3CMP register update

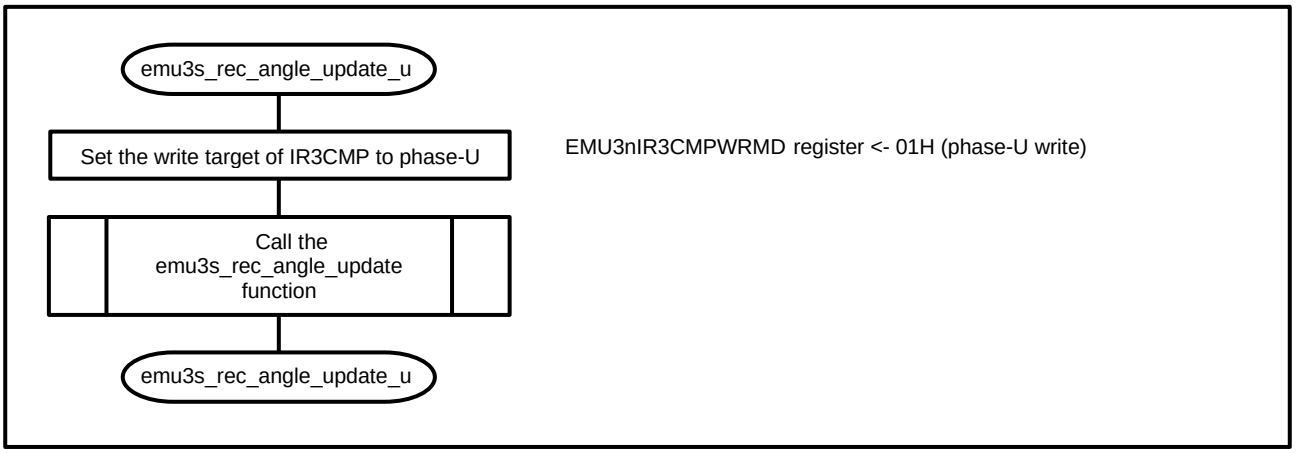


Figure 5.8 IR3CMP register update (phase-U)

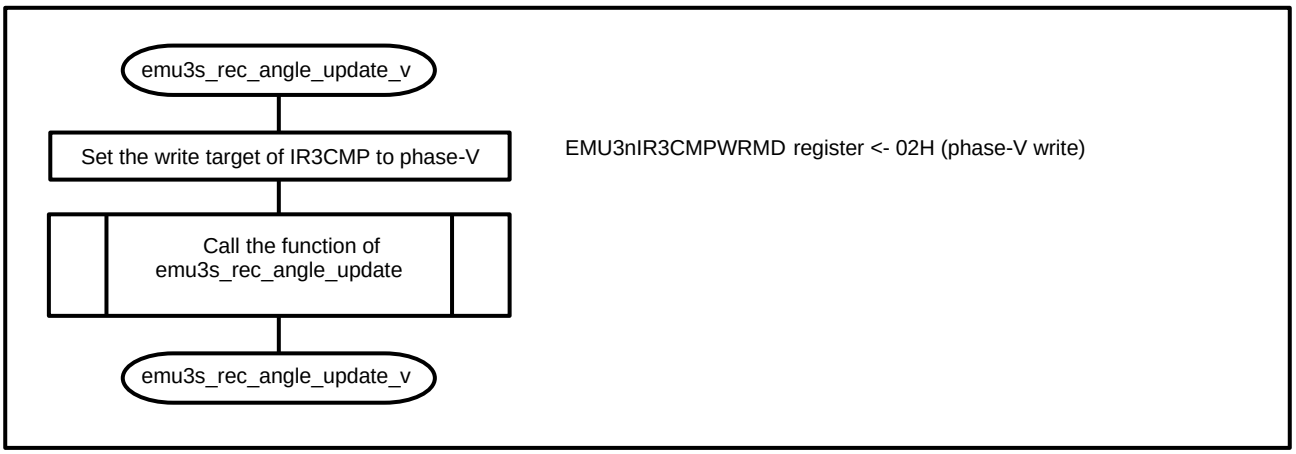


Figure 5.9 IR3CMP register update(phase-V)

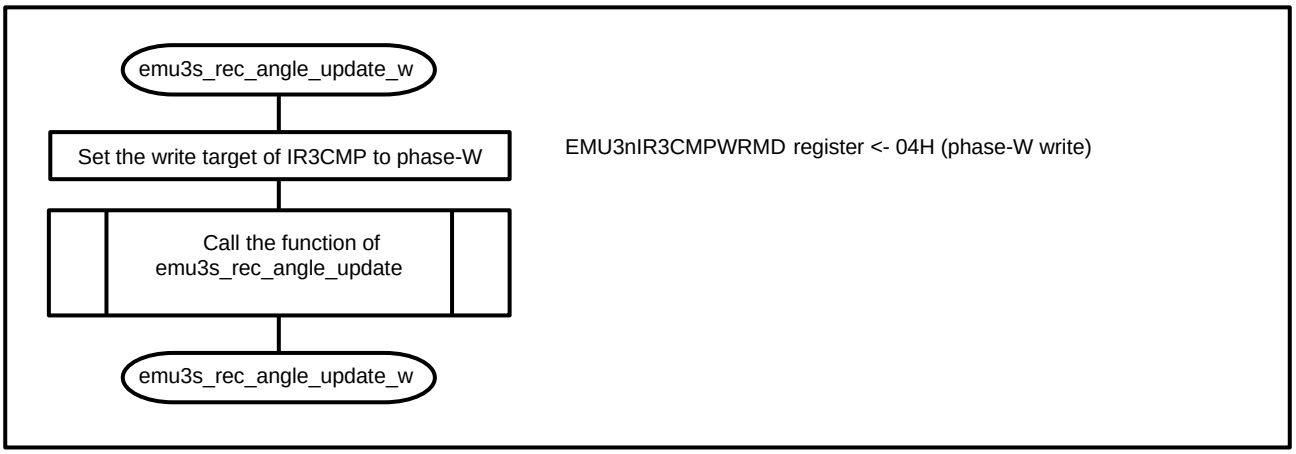


Figure 5.10 IR3CMP register update(phase-W)

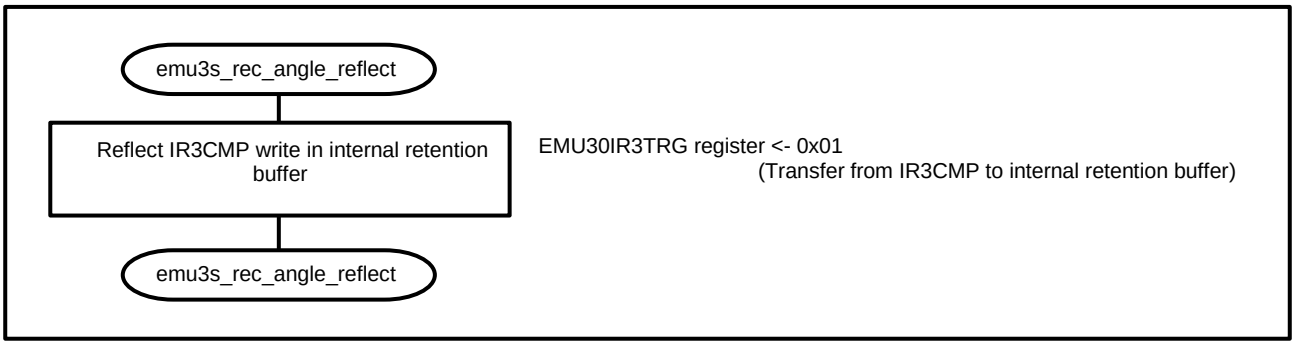


Figure 5.11 Reflection in retention buffer

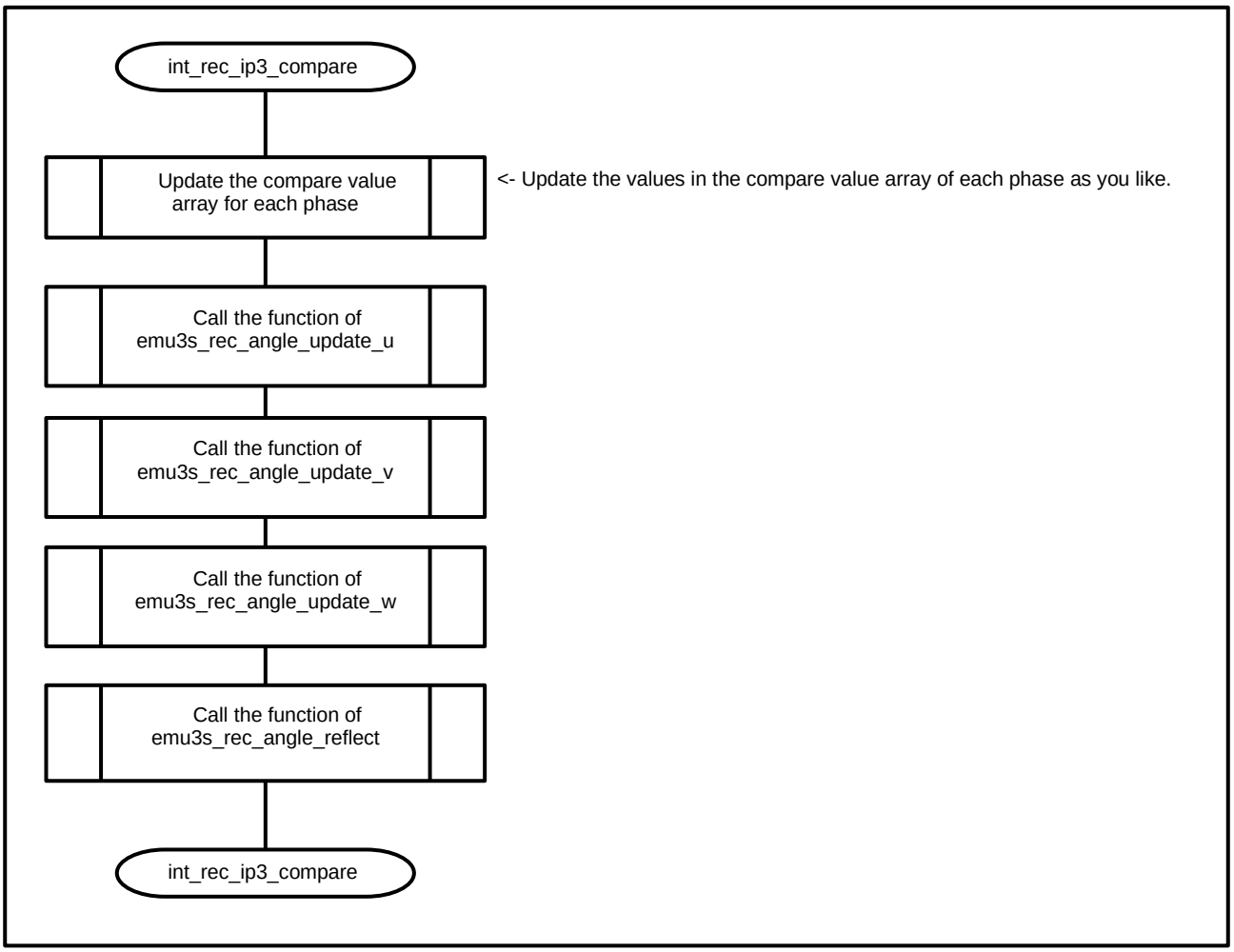


Figure 5.12 Interrupt processing in U-phase compare

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	2022.03.31	-	First edition
1.10	2025.01.31	4, 9, 11, 14, 15, 16, 19, 24, 28, 29	Corrected typos

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. 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 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 alteration, modification, copying or reverse engineering.
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 system; 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 any unauthorized access to or use of a Renesas Electronics product or a 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 INVULNERABLE 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 conditions 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 or 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.
12. 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.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. 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.

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

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/.

© 2021 Renesas Electronics Corporation. All rights reserved.