Application Note

RL78/G1F

R01AN4502EJ0200
Rev.2.00
2021.05.26

Sensorless speed control of 120-degree conducting controlled permanent magnetic synchronous motor with detection of initial position (Implementation)

Summary

This application note explains the sample programs driving a permanent magnetic synchronous motor in the 120-degree conducting method using the RL78/G1F microcontroller and how to use the motor control development support tool, 'Renesas Motor Workbench (RMW)'.

These sample programs are only able to be used as reference and Renesas Electronics Corporation does not guarantee the operations. Please use after carrying out through evaluation in a suitable environment.

Operation checking device

Operations of the sample programs have been checked by using the following device.

RL78/G1F(R5F11BLEAFB)

Target sample programs

The target sample programs of this application note are as follows.

- RL78G1F_RSSK_120_IPD_CSP_CC_V200 (IDE: CS+ for CC)
- RL78G1F_RSSK_120_IPD_E2S_CC_V200 (IDE: e^{2}studio)

RL78/G1F 120-degree conducting control sample program
for RL78/G1F 24V Motor Control Evaluation System

Reference

- RL78/G1F Group User's Manual: Hardware (R01UH0516EJ0112)
- Application note: '120-degree conducting control of permanent magnetic synchronous motor: algorithm' (R01AN2657EJ0120)
- Renesas Motor Workbench V.1.00 User's Manual (R21UZ0004EJ0203: Renesas-Motor-Workbench-V2-0d)
- Renesas Solution Starter Kit 24V Motor Control Evaluation System for RX23T User's Manual (R20UT3697EJ0120)
- RL78/G1F CPU Card User's Manual (R12UZ0014EJ0100)

Contents

1. Overview

This application note explains how to implement the 120-degree conducting control sample programs of permanent magnetic synchronous motor (PMSM) using the RL78/G1F microcontroller and how to use the motor control development support tool, 'Renesas Motor Workbench'. Note that this sample programs use the algorithm described in the application note '120-degree conducting control of permanent magnetic synchronous motor: algorithm'.

1.1 Development environment

Table 1-1 and Table 1-2 show development environment of the sample programs explained in this application note.

Table 1-1 Development Environment of the Sample Programs (H/W)

Microcontroller	Evaluation board	Motor
RL78/G1F (R5F11BLEAFB)	24V inverter board (Note 1) and RL78/G1F CPU Card (Note 2)	TSUKASA TG-55L (Note 3)

Table 1-2 Development Environment of the Sample Programs (S/W)

CS+ version	Build tool version
V8.05.00	CC-RL V1.08.00

e^{2}studio version	Build tool version
2021-04	CC-RL V1.08.00

For purchase and technical support, please contact sales representatives and dealers of Renesas Electronics Corporation.

Notes:

1. 24V inverter board (RTK0EM0001B00001BJ) is a product of Renesas Electronics Corporation.
2. Two kinds of RL78/G1F CPU Card can be used.
   ・RTK0EML240C03000BJ : Renesas Electronics
   ・T5103 : Desk Top Laboratories Inc. (http://desktoplab.co.jp/)
3. TG-55L is a product of TSUKASA ELECTRIC.
   TSUKASA ELECTRIC. (https://www.tsukasa-d.co.jp/en/)

2. System overview

Overview of this system is explained below.

2.1 Modifying the Evaluation Board

For the sample programs in this application note, the 24V Inverter Board and RL78/G1F CPU Card should be modified according to below explanations.

(1) Change from 3 phases to DC-link current detection.

I. Remove R72 and R97.

II. Connect TH1, TH2 and TH3.

III. Short circuit between 2-3 pins on JP3 and set the voltage on shunt resistor to be entered without going through the OP amp on the board.

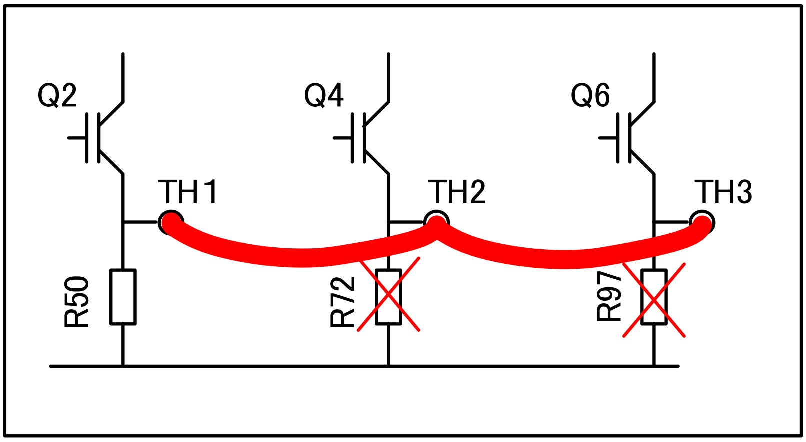
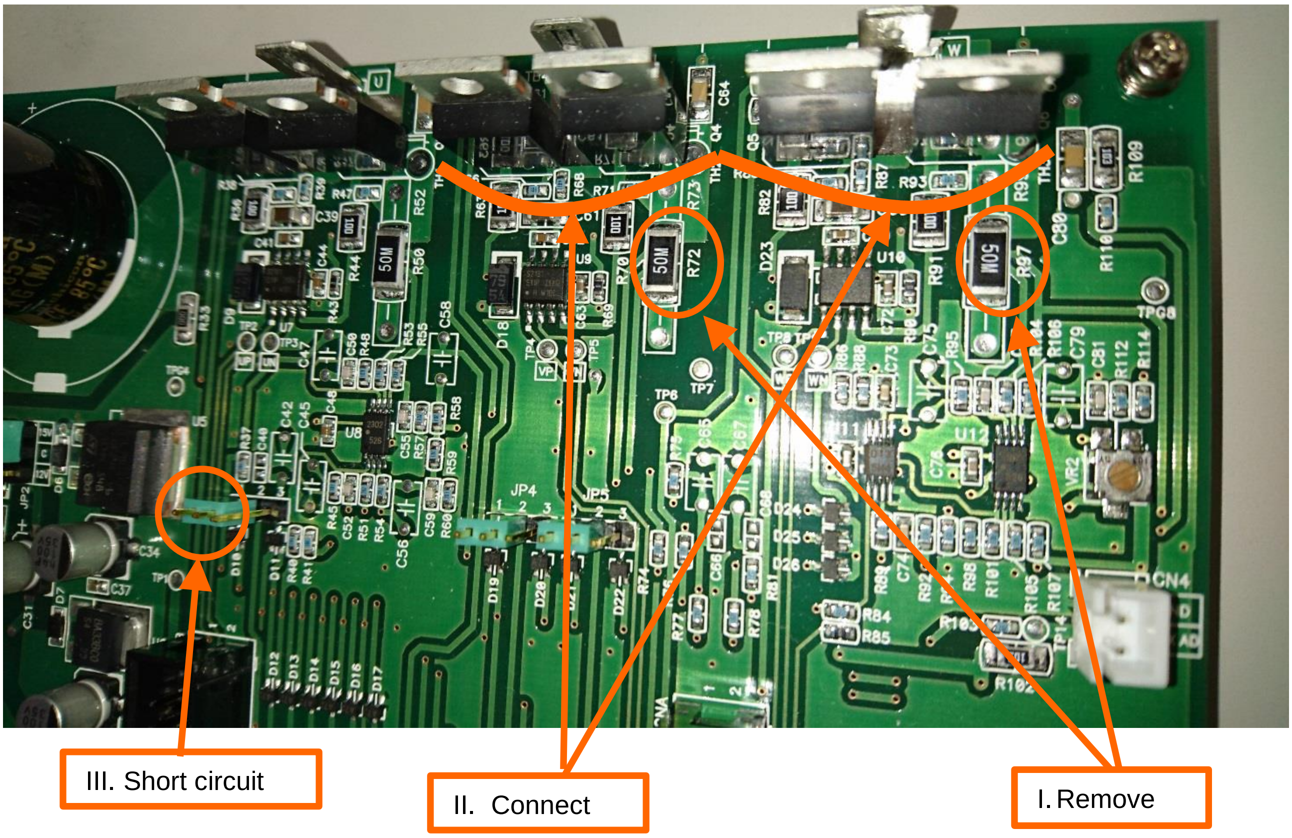


Figure 2-1 Change 3-shunt to 1-shunt



(2) Modification for the circuit to sensing voltage of each phase

I. Mount 10 [kΩ] on R34, R61, and R79.

II. Mount 510 [Ω] on R67.

III. Change resisters R42, R46 and R49 from 470 [Ω] to 1.5 [kΩ]

IV. Remove C49, C51 and C53.

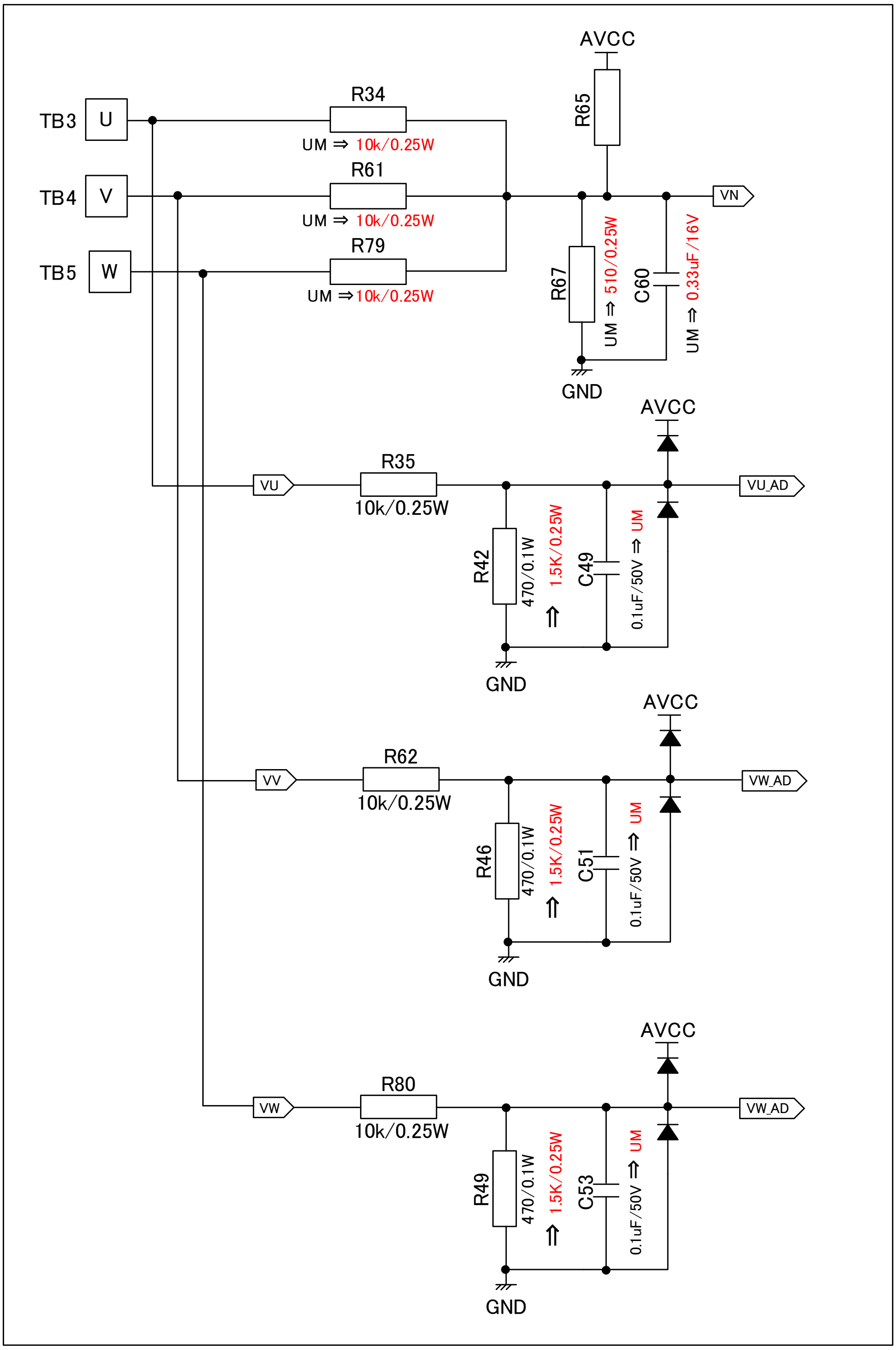
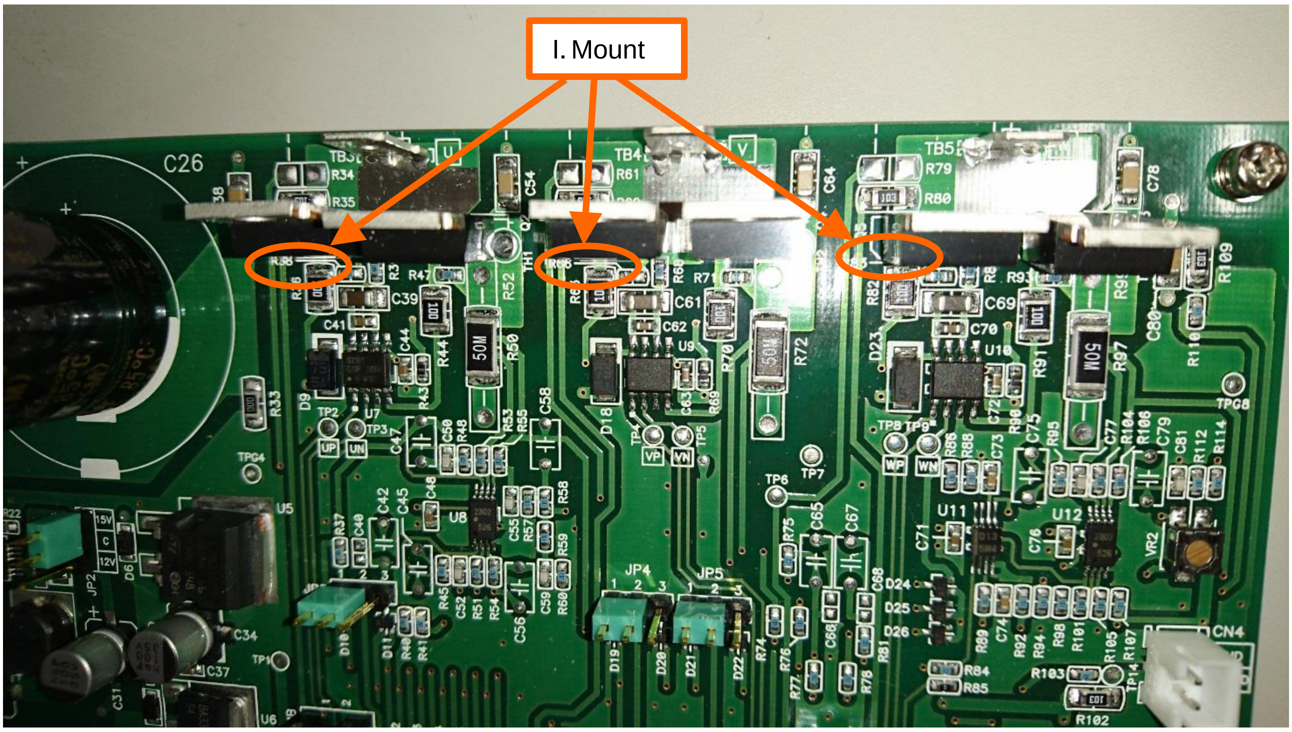
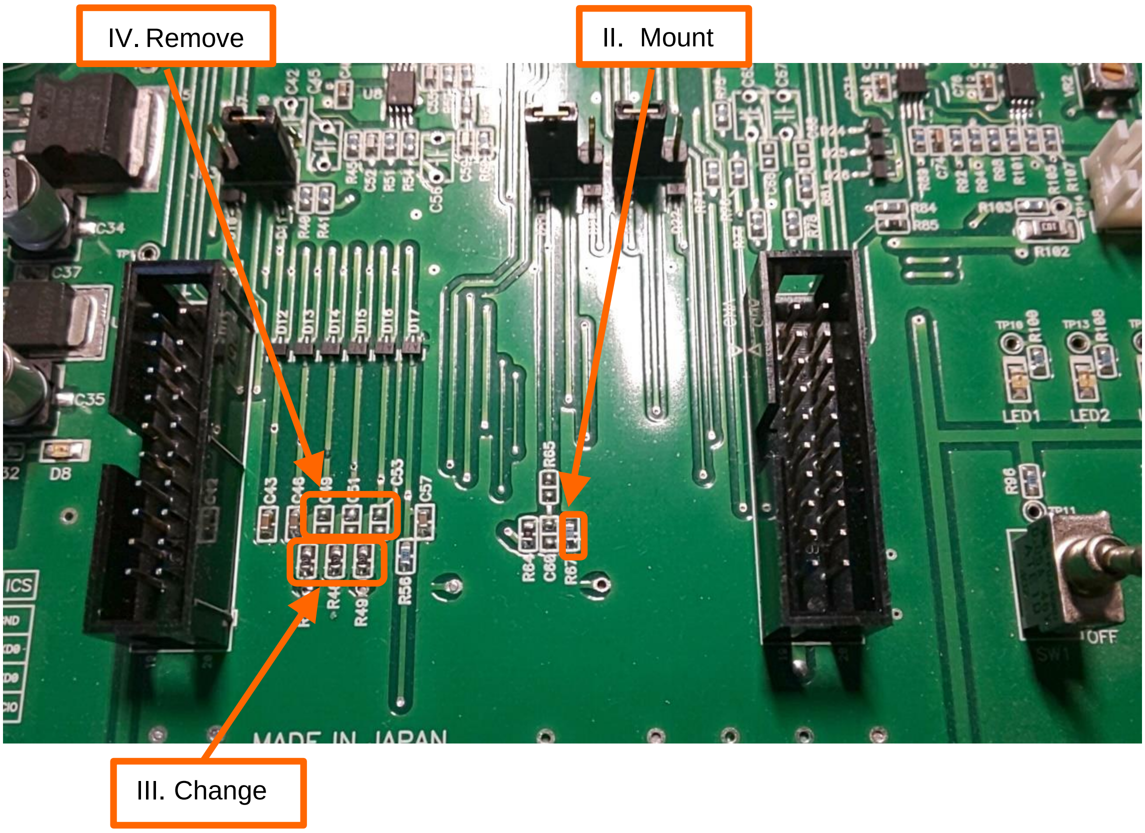


Figure 2-2 Modification for the circuit to sensing voltage of each phase


I. Mount


IV. Remove
II. Mount
III. Change

2.2 Hardware configuration

The hardware configuration is shown below.

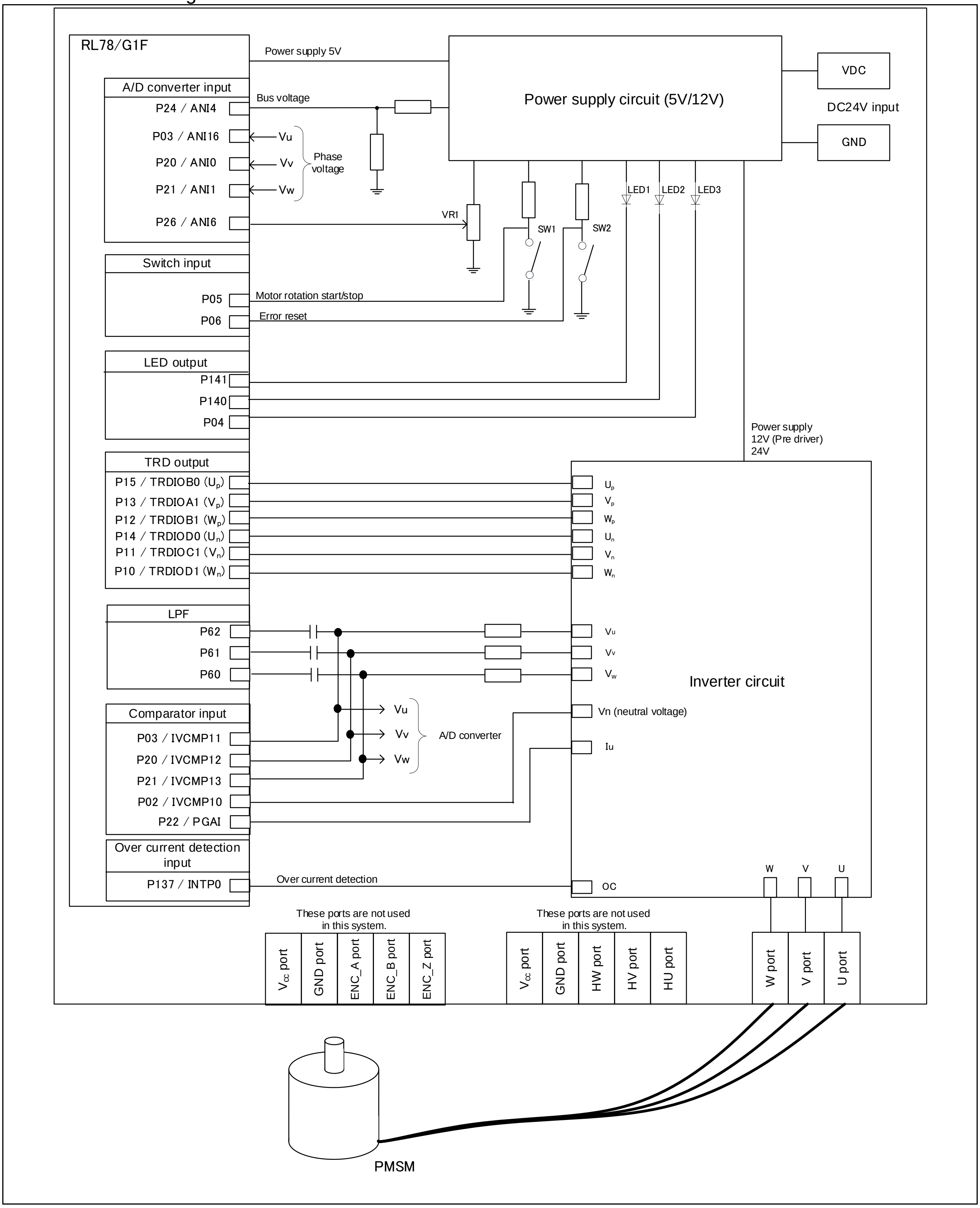


Figure 2-3 Hardware Configuration Diagram

2.3 Hardware specifications

2.3.1 User interface

Table 2-1 is a list of user interfaces of this system.

Table 2-1 User Interface

Item	Interface component	Function
Rotational speed	Variable resistance (VR1)	Rotational speed command value input (analog values)
START/STOP	Toggle switch (SW1)	Start/Stop command for motor rotation
ERROR RESET	Toggle switch (SW2)	Command of recovery from error status
LED1	Yellow green LED	At the time of Motor rotation: ON At the time of stop: OFF
LED2	Yellow green LED	At the time of error detection: ON At the time of normal operation: OFF
LED3	Yellow green LED	Zero-cross is detected by the comparator : ON Others: OFF
RESET	Push switch (RESET1)	System reset

Table 2-2 is a list of port interfaces of RL78/G1F microcontroller of this system.

Table 2-2 Port Interface

R5F11BLEAFB Port name	Function
P24 / ANI4	Inverter bus voltage measurement
P26 / ANI6	For inputting rotational speed command values (analog values)
P05	START / STOP toggle switch
P06	ERROR RESET toggle switch
P141	LED1 ON / OFF control
P140	LED2 ON / OFF control
P04	LED3 ON / OFF control
P03 / ANI16 / IVCMP11	U Phase voltage measurement(A/D) / U Phase zero-crossing compare(use the comparator)(Note1)
P20 / ANI0 / IVCMP12	V Phase voltage measurement(A/D) / V Phase zero-crossing compare(use the comparator)(Note1)
P21 / ANI1 / IVCMP13	W Phase voltage measurement(A/D) / W Phase zero-crossing compare(use the comparator)(Note1)
P02 / IVCMP10	Input virtual neutral voltage(Note1)
P15 / TRDIOB0	PORT output / PWM output (U_p)
P13 / TRDIOA1	PORT output / PWM output (V_p)
P12 / TRDIOB1	PORT output / PWM output (W_p)
P14 / TRDIOD0	PORT output / PWM output (U_n)
P11 / TRDIOC1	PORT output / PWM output (V_n)
P10 / TRDIOD1	PORT output / PWM output (W_n)
P137 / INTP0	PWM emergency stop input at the time of overcurrent detection(Note2)
P22 / PGAI	PWM emergency stop input at the time of overcurrent detection (use PGA+CMP0) (Note2) / current measurement for initial position detection

Note: 1. A/D converters are used at low speed (under 1855[rpm]), and the comparator is used at high speed (over 1855[rpm]).
2. Select the external circuit or “PGA + CMP0” for the detection of overcurrent at compile.

2.3.2 Peripheral functions

Table 2-3 is a list of peripheral functions used in this system.

Table 2-3 List of Peripheral Functions

Peripheral Function	Usage
10-bit A/D converter	- Rotational speed command value input (Board UI mode) - Inverter bus voltage measurement - Voltage of each phase U, V, and W measurement - Offset current measurement
Timer Array Unit (TAU)	- Timer Window for CMP1 output - Free-running timer for rotational speed measurement - Delay timer for changing conducting pattern
Timer RD (TRD)	PWM output
Timer RG(TRG)	1[ms] interval timer
Timer RX(TRX)	- Current rising period measurement
Comparator CMP0	- Judgement for reaching threshold current - Overcurrent detection(Note)
Comparator CMP1	Detection of zero-crossing at U/V/W Phase
Programmable Gain Amp (PGA)	- measurement of current offset - Judgement for reaching threshold current - Overcurrent detection(Note)
External Interrupt (INTP0)	Overcurrent detection(Note)
PWM Option Unit A (PWMOPA)	PWM output cutoff

Note: INTP0 and “PGA+CMP0” are used exclusive. (These are selectable at compile.)

(1) 10-bit A/D converter

The rotational speed command value input, U phase voltage (Vu), V phase voltage (Vv), W phase voltage (Vw), inverter bus voltage (Vdc), and offset current are measured by using the ‘10-bit A/D converter’.

The operation mode is set as below.

- The channel selection mode: the select-mode
- The conversion operation mode: sequential conversion mode (offset current)
  one-shot conversion mode (others)
- Trigger: Software trigger

(2) Timer Array Unit (TAU)

a. Timer Window for CMP1 output

The channels 0 and 2of TAU are used as Timer Window which should be synchronous with PWM on timing and mask CMP1 output.

b. Free-running timer for rotational speed measurement

This channel 1 of TAU is used as free-running counter for rotational speed calculation.

c. Delay timer for changing conducting pattern

The channel 3 of TAU is used as delay timer for changing conducting pattern with π/6 phase from the zero-crossing point.

(3) Timer RD (TRD)

Three-phase PWM output of upper arm chopping with dead time (complementary) is performed using the Complementary PWM Mode. When detecting an overcurrent, the PWM output ports are set to high impedance output using PWM option Unit A.

(4) Timer RG (TRG)

Timer RG (TRG) is used as 1 millisecond interval timer.

(5) Timer RX (TRX)
Timer RX (TRX) is used as timer for measuring period of reaching threshold current during initial position detection.

(6) Comparator CMP0
CMP0 is used to detect overcurrent or judge for reaching threshold current with PGA.

(7) Comparator CMP1
CMP1 is used to detect zero-crossing at high speed (over 1855[rpm])

(8) Programmable Gain Amp (PGA)
PGA is used to measure offset current, detect overcurrent or judge for reaching threshold current with A/D or CMP0.

(9) External interrupt (INTP0)
An overcurrent is detected by an external circuit.

(10) PWM Option Unit A (PWMOPA)
When overcurrent is detected by INTP0 or CMP0, the PWM output from TRD is cutoff.

2.4 Software structure

2.4.1 Software file structure

The folder and file configurations of the sample programs are given below.

Table 2-4 Folder and File Configuration of the Sample Program

Folder		File	Content
config		r_mtr_config.h	Common definition for software configuration
		r_mtr_motor_parameter.h	Configuration definition for motor parameters
		r_mtr_control_parameter.h	Configuration definition for control parameters
		r_mtr_inverter_parameter.h	Configuration definition for inverter parameters
		r_mtr_scaling_parameter.h	Configuration definition for scaling parameters
application	main	main.h main.c	Main function
	board	r_mtr_board.h r_mtr_board.c	Function definition for board UI
	ics	r_mtr_ics.h r_mtr_ics.c	Function definition for Analyzer[Note1] UI
		ICS_define.h	CPU definition for RMW
		RL78G1F_vector.c	Interrupt vector function definition for RMW
		ics_RL78G1F.obj	Communication library for RMW
driver	auto_generation	cstart.asm hdwinit.asm iodefine.h r_stdint.h stkinit.asm	Auto generation files
		mtr_ctrl_mrssk.h, mtr_ctrl_mrssk.c	Function definition for inverter board control
		r_mtr_ctrl_rl78g1f.h, r_mtr_ctrl_rl78g1f.c	Function definition for MCU control
middle	lib	r_dsp.h	Digital signal controller library for CC toolchain
		R_dsp_rl78_CC.lib	DSP library
		r_mtr_common.h	Common definition
		r_mtr_parameter.h	Various parameter definition
		r_mtr_driver_access.h, r_mtr_driver_access.c	Function definition for User access
		r_mtr_statemachine.h, r_mtr_statemachine.c	Function definition for state transition
		r_mtr_120.h, r_mtr_120.c	Function definition for 120-degree conducting control
		r_mtr_interrupt.c	Interrupt function definition
		r_mtr_ctrl_gain_calc.obj	PI gain calculation definition

Note 1: Regarding the specification of Analyzer function in the motor control development support tool 'Renesas Motor Workbench(RMW)', please refer to the chapter 4.The identifier 'ics/ICS (ICS is previous motor control development support tool 'In Circuit Scope') is attached to the name of folders, files, functions, variables related to 'Renesas Motor Workbench'.

2.4.2 Module configuration

Figure 2-4 shows module configuration of the sample programs.

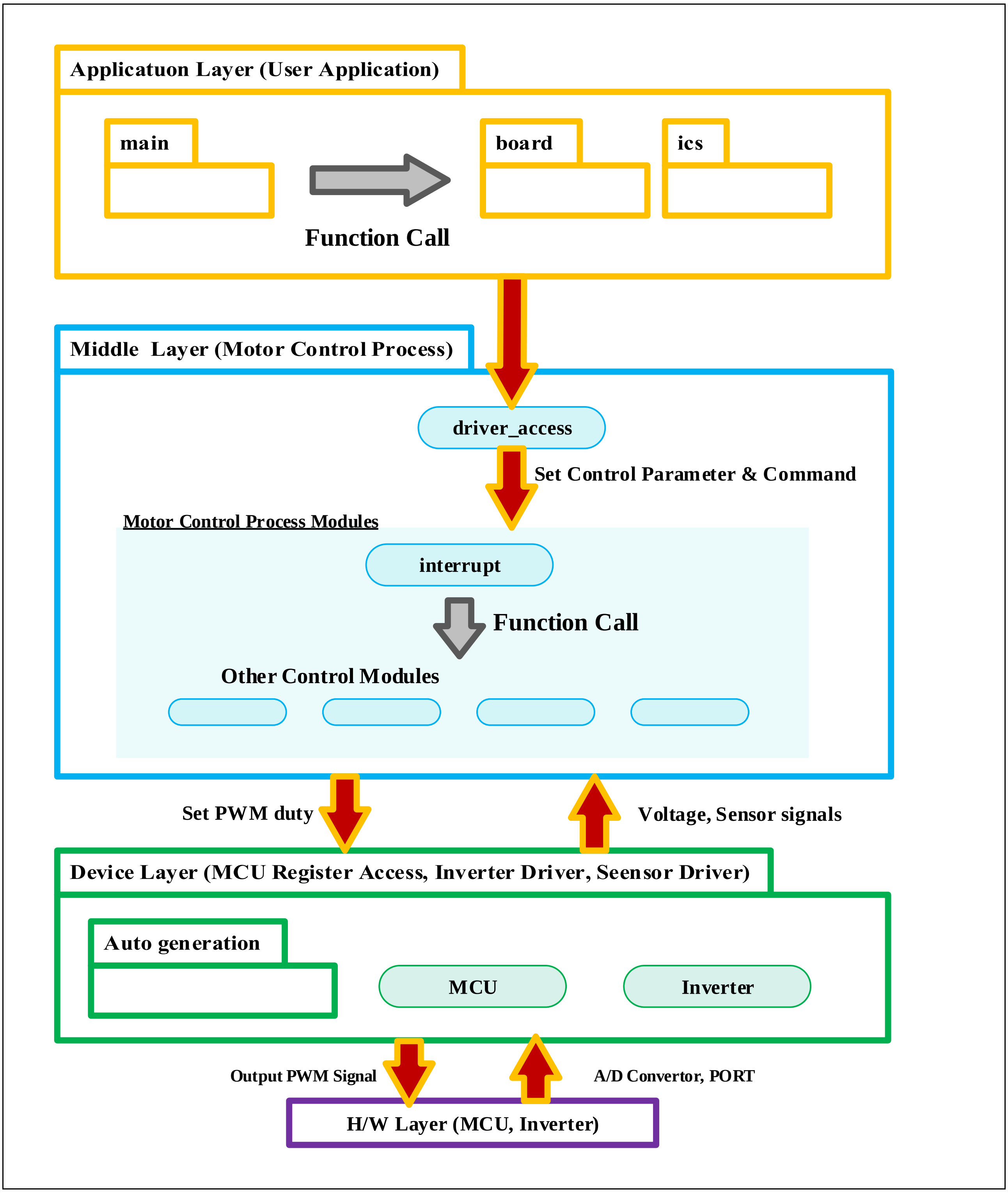


Figure 2-4 Module Configuration of the Sample Programs

2.5 Software specifications

Table 2-5 shows the basic specifications of target software of this application note. For details of 120-degree conducting control, refer to the application note ‘120-degree conducting control of permanent magnetic synchronous motor: algorithm’

Table 2-5 Basic Specifications of Software

<table>
<tr><th>Item</th><th colspan="2">Content</th></tr>
<tr><td>Control method</td><td colspan="2">120-degree conducting method (chopping upper arm)</td></tr>
<tr><td>Motor rotation start/stop</td><td colspan="2">Determined depending on the level of SW1 (P05) (”Low”: rotation start “High”: stop) or input from Renesas Motor Workbench</td></tr>
<tr><td>Position detection of rotor magnetic pole</td><td colspan="2">Position detection based on induced voltage with A/D converters or comparator (by every 60 degrees)
- Under 1855 [rpm], the zero-crossing is detected by A/D converter every 50 [µs]
- Over 1855 [rpm], the zero-crossing is detected by comparator
- At change of voltage pattern, set PWM duty and change commutate pattern at same time.</td></tr>
<tr><td>Input voltage</td><td colspan="2">DC24[V]</td></tr>
<tr><td>Main clock frequency</td><td colspan="2">CPU clock: f_{CLK} 32[MHz]
TRD clock: f_{HOCO} 64[MHz]</td></tr>
<tr><td>Carrier frequency (PWM)</td><td colspan="2">20 [kHz]</td></tr>
<tr><td>Dead time</td><td colspan="2">2 [µs]</td></tr>
<tr><td>Control cycle</td><td colspan="2">Speed PI control: every 1 [ms]</td></tr>
<tr><td>Rotational speed control range</td><td colspan="2">265 [rpm] to 2650 [rpm] (Note1)
Both CW and CCW are supported</td></tr>
<tr><td>Optimization</td><td colspan="2">Default</td></tr>
<tr><td rowspan="2"></td><td>ROM</td><td>954 bytes</td></tr>
<tr><td>RAM</td><td>17337 bytes</td></tr>
<tr><td>Processing stop for protection</td><td colspan="2">- Disables the motor control signal output (six outputs), under any of the following conditions.
1. Inverter bus voltage exceeds 28 V (monitored per 1 [ms])
2. Inverter bus voltage is less than 15 V (monitored per 1 [ms])
3. Rotational speed exceeds 3975 rpm (monitored per 1 [ms])
4. Zero-crossing are not detected for constant period.
5. Detection of unexpected output voltage pattern
6. TRX overflow during initial position detection
7. Detection of overcurrent
- The ports executing PWM output are set to high impedance state when an overcurrent is detected by external circuit (low level input occurs in INTP0 port) or by internal PGA+CMP0.</td></tr>
</table>

Note1 : Please refrain from driving motor overrated speed for a long period.

2.6 User option bytes

The settings of the user option byte area of the RL78/G1F flash memory are shown below.

Table 2-6 User Option Byte Settings

Setting	Address	Value	Description
7877F8	000C0H /010C0H	01111000B	- Uses watchdog timer interval interrupt: does not use interval interrupt - Period when watchdog timer window is open: 100% - Watchdog timer counter operation control: Counter operation possible (After reset is canceled, count begins) - Watchdog timer overflow time: 136 [ms] - Watchdog timer counter operation control: In HALT/STOP mode, counter operation stops
	000C1H /010C1H	01110111B	- LVD settings (reset mode) Rising edge: 3.02 [V] Falling edge: 2.96 [V]
	000C2H /010C2H	11111000B	- Flash operation mode setting: HS (high-speed main) mode - High-speed on-chip oscillator/block frequency fHOCO: 64 [MHz] fIH: 32 [MHz]

3. Descriptions of the control program

The target sample programs of this application note are explained here.

3.1 Contents of control

3.1.1 Motor start/stop

Starting and stopping of the motor are controlled by input from Renesas Motor Workbench or SW1 & VR1.

A general-purpose port is assigned to SW1. The port is read within the main loop. When the port is at a "Low" level, it is determined that the start switch is being pressed. Conversely, when the level is switched to "High", the program determines that the motor should be stopped.

In addition, an analog input port is assigned to VR1. The input is A/D converted within the main loop to generate a rotational speed command value. When the command value is less than Hall effect sensor:530[rpm] /Sensorless: 265[rpm], the program determines that the motor should be stopped.

3.1.2 A/D Converter

(1) Motor rotational speed command value

The motor rotational speed command value can be set by Renesas Motor Workbench input or A/D conversion of the VR1 output value (analog value). The A/D converted VR1 value is used as rotational speed command value, as shown below. Maximum value of conversion ratio is set to achieve maximum speed by VR1 input.

Table 3-1 Conversion Ratio of the Rotation Speed Command Value

Item	Conversion ratio (Command value: A/D conversion value)		Channel
Rotational speed command value	CW	0 [rpm] to 2650 [rpm] : 01FFH to 03FFH	ANI6
	CCW	-2650 [rpm] to 0 [rpm] : 0000H to 01FFH	

(2) Inverter bus voltage

Inverter bus voltage is measured as given in Table 3-2. It is used for modulation factor calculation and over/low voltage detection. (When an abnormality is detected, PWM is stopped).

Table 3-2 Inverter Bus Voltage Conversion Ratio

Item	Conversion ratio (Inverter bus voltage: A/D conversion value)	Channel
Inverter bus voltage	0 [V] to 111 [V] : 0000H to 03FFH	ANI4

(3) U phase, V phase, and W phase voltage

The U, V and W phase voltages are measured as shown in Table 3-3 and used for determining zero-crossing of induced voltage.

Table 3-3 Conversion Ratio of U, V, and W Phase Voltage

Item	Conversion ratio (U, V, and W phase voltage: A/D conversion value)	Channel
U, V, W phase voltages	0 [V] to 111 [V] : 0000H to 03FFH	ANI16, ANI0, ANI1

(4) Offset current

The U phase current is measured as shown in Table 3-4 and used for setting threshold current for CMP0.

Table 3-4 Conversion Ration of Offset current

Item	Conversion ratio (U phase current: A/D conversion value)	Channel
U, V, W phase voltages	0 [A] to 12.5 [A] : 0000H to 03FFH	PGAOUT

Note: For more details of A/D conversion, refer to RL78/G1F User's Manual: Hardware.

3.1.3 PGA (Programmable Gain Amplifier)

A PGA is used to detect current flowing through a shunt resistor whenever overcurrent or initial position is detected.

Table 3-5 Detection of Current Flowing Through Shunt Resistor

Item	Description	Remarks
Select PGA gain	8x	Select from 4/8/16/32x
Select GND	Vss	Vss or PGAGND

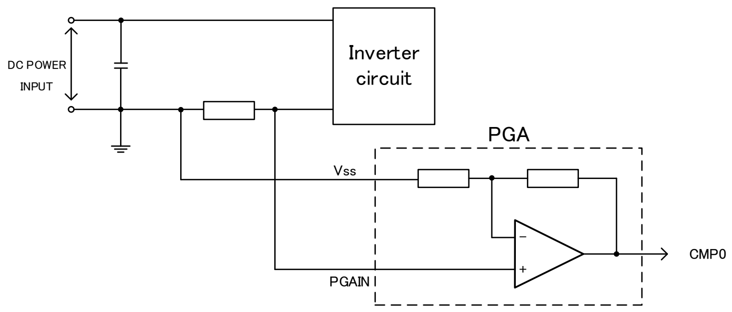


Figure 3-1 Connection Between PGA and Inverter Circuit

3.1.4 Comparator

(1) Detection of current flowing through shunt resistor (CMP0)

Compares the PGA output to the internal D/A converter standard value and detects the current flowing through the shunt resistor. The PGA gain is set to 8x, so it is possible to detect up to 12.5 [A], including offset.

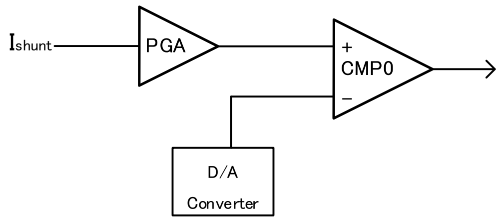


Figure 3-2 Current Detection Using CMP0

(2) Induced voltage zero crossing detection (CMP1)

Selects from 3 phase voltages (Vu, Vv, Vw) or a neutral position voltage (Vn) according to the electrical current pattern for the phase to be input to the CMP1 “+” side or “-” side, when induced voltage zero crossing is detected.

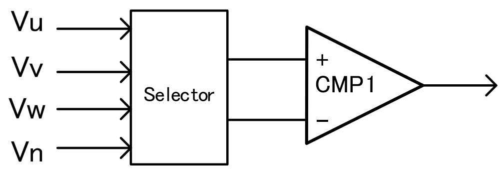


Figure 3-3 Induced Voltage Zero Crossing Detection Using CMP1

Table 3-6 Electrical Current Patterns and Selector Selection Patterns

Selector Selection Pattern	Electrical Current Pattern	'+' Side Input	'－' Side Input
1	High: W Low: V	Vu	Vn
2	High: W Low: U	Vn	Vv
3	High: V Low: U	Vw	Vn
4	High: V Low: W	Vn	Vu
5	High: U Low: W	Vv	Vn
6	High: U Low: V	Vn	Vw

3.1.5 Speed control

In this system, rotational speed is calculated from a difference of the current timer value and the timer value 2π [rad] before. The timer values are obtained when patterns are switched after zero-crossing of induced voltage, while having the timer of performed free running.

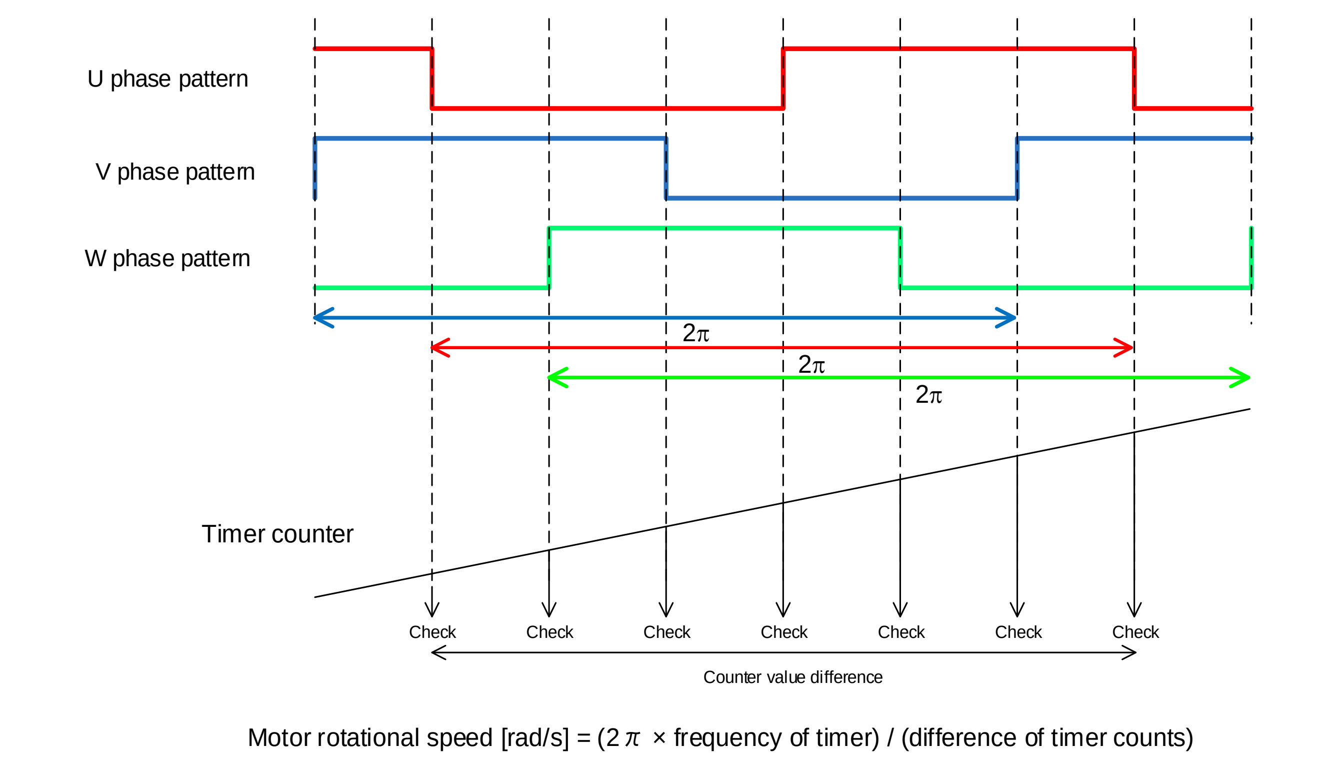


Figure 3-4 Method of Calculation for Rotational Speed

The target sample software of this application note uses PI control for speed control. A voltage command value is calculated by the following formula of speed PI control.

$$v^* = (K_{P\omega} + \frac{K_{I\omega}}{S})(\omega^* - \omega)$$

v^* : Voltage command value ω^*: Speed command value ω: Rotation speed
$K_{P\omega}$: Speed PI proportional gain $K_{I\omega}$: Speed PI integral gain s: Laplace operator

For more details of PI control, please refer to specialized books.

3.1.6 Voltage control by PWM

PWM control is used for controlling output voltage. The PWM control is a control method that continuously adjusts the average voltage by varying the duty of pulse, as shown in Figure 3-5.

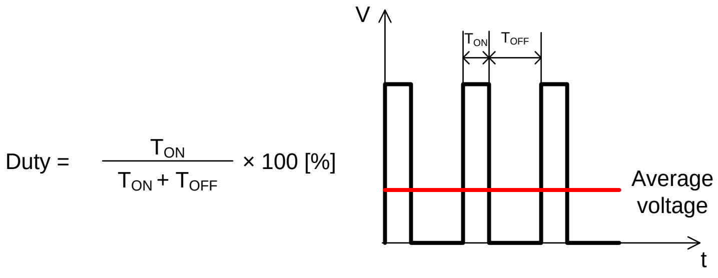


Figure 3-5 PWM Control

Here, modulation factor “m” is defined as follows.

$$m = \frac{V}{E}$$

m: Modulation factor V: Command value voltage E: Inverter bus voltage

This modulation factor is set to resisters for PWM duty in TRD.

In the target software of this application note, upper arm chopping is used to control the output voltage and speed. Figure 3-6 and Figure 3-7 show an example of output waveforms at upper arm chopping. Non-complementary / complementary PWM can be switched by setting the configuration definition file “r_mtr_config.h”.

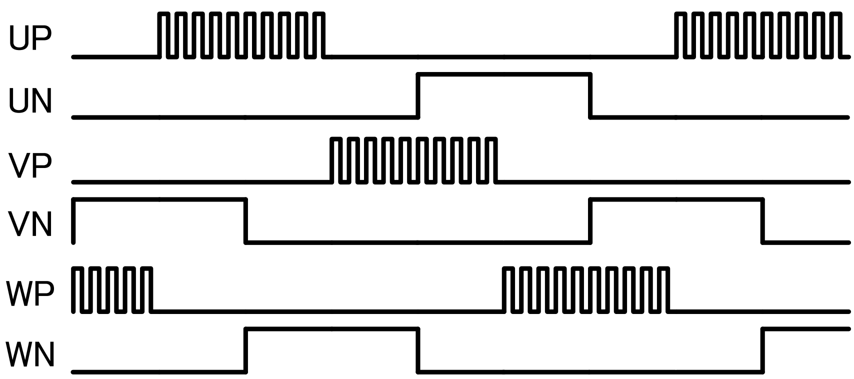


Figure 3-6 Upper Arm Chopping (Non-complementary PWM)

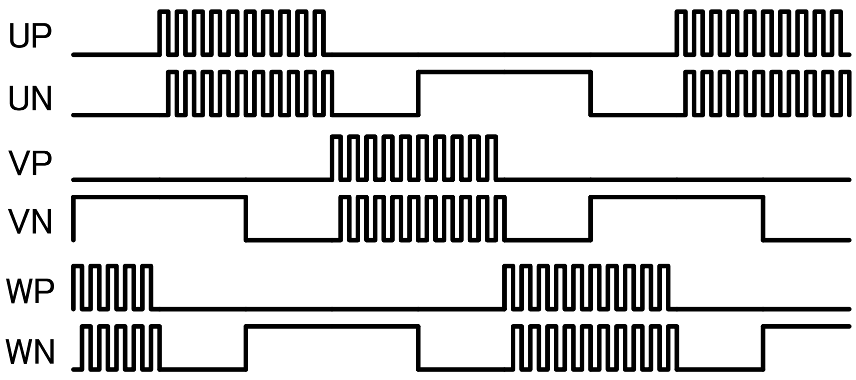
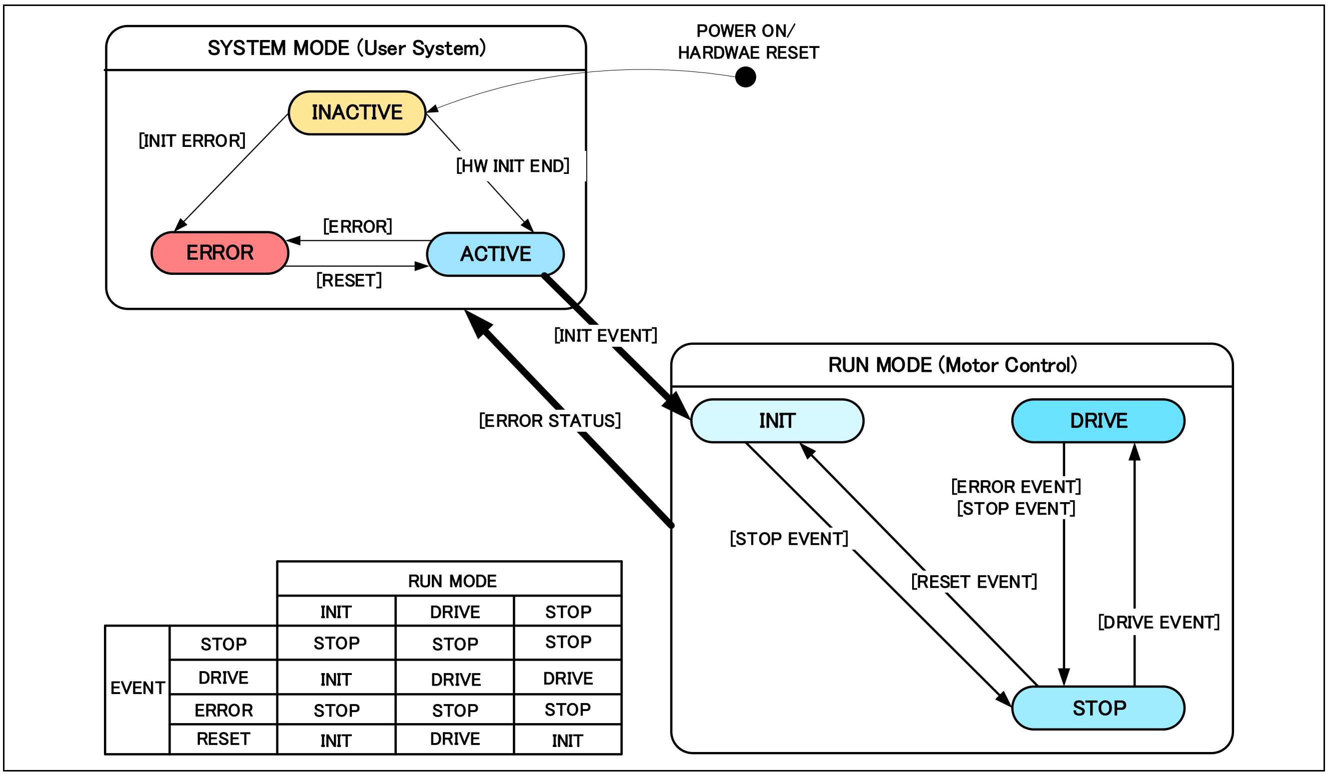


Figure 3-7 Upper Arm Chopping (Complementary PWM)

3.1.7 State transitions

Figure 3-8 shows state transition diagrams of 120-degree conducting control software.



		RUN MODE		
		INIT	DRIVE	STOP
EVENT	STOP	STOP	STOP	STOP
	DRIVE	INIT	DRIVE	DRIVE
	ERROR	STOP	STOP	STOP
	RESET	INIT	DRIVE	INIT

Figure 3-8 State Transition Diagram 120-degree Conducting Control Software

(1) SYSTEM MODE

"SYSTEM MODE" indicates the operating states of the system. "SYSTEM MODE" has 3 states that are motor drive stop (INACTIVE), motor drive (ACTIVE), and abnormal condition (ERROR).

(2) RUN MODE

"RUN MODE" indicates the condition of the motor control. The state is changed by occurrence of "EVENT".

(3) EVENT

"Event" indicates the change of "RUN MODE". When "EVENT" occurs, "RUN MODE" changes as shown table in Figure 3-8. Each "Event" is caused by occurrence as shown in Table 3-7.

Table 3-7 List of "EVENT"

"EVENT" name	Occurrence factor
STOP	By user operation
DRIVE	By user operation
ERROR	When the system detects an error
RESET	By user operation

3.1.8 Startup method

Using the induced voltage arising from changes in the flux in the permanent magnet (rotor), the sensorless control detects the position of the magnetic pole every 60 degrees. However, induced voltage is generated by rotation, so when the rotational speed is low, the position of the magnetic pole cannot be detected.

However, as a way to start up, there is a method of forcing synchronization regardless of the position of the permanent magnet, by forcibly changing the electrical current pattern to generate a rotating magnetic field.

The startup method using sample software is shown in Figure 3-9. The rotor draw-in happens for the first time when the transition is made to MTR_MODE_DRIVE. Then it is driven in an open-loop, and it starts up smoothly by being transferred to PI control after the induced voltage zero crossing signal is detected 3 times.

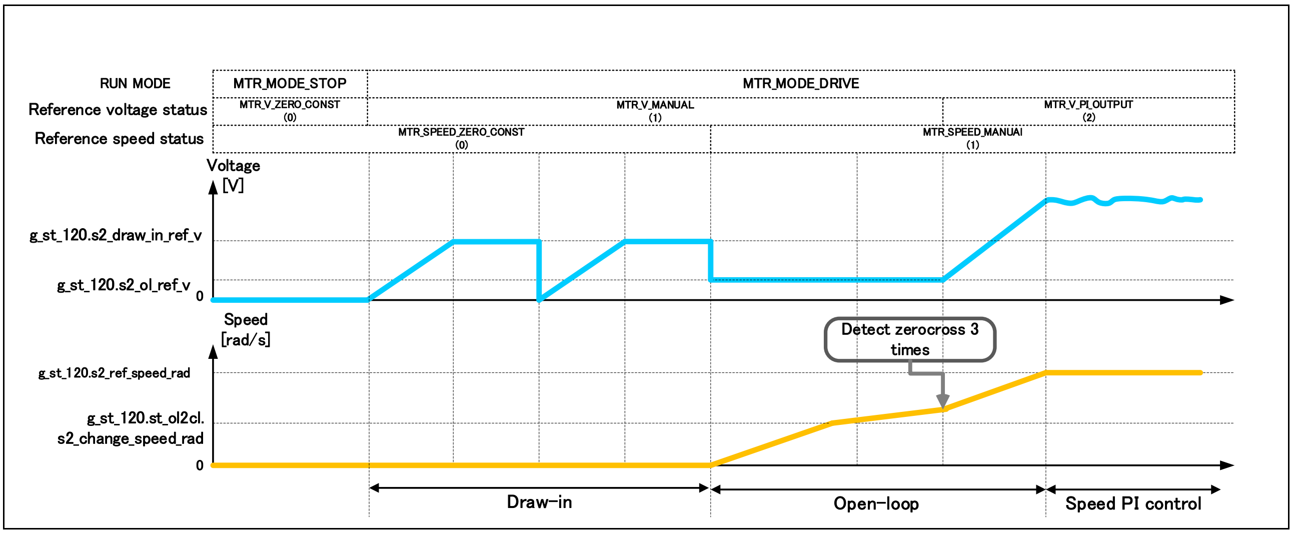


Figure 3-9 Startup Sequence (Using Draw-in)

The startup method using the initial position detection function is shown in Figure 3-10. Instead of draw-in using the position of the magnetic pole, the open-loop process starts after the position of the rotor is detected by the initial position detection process.

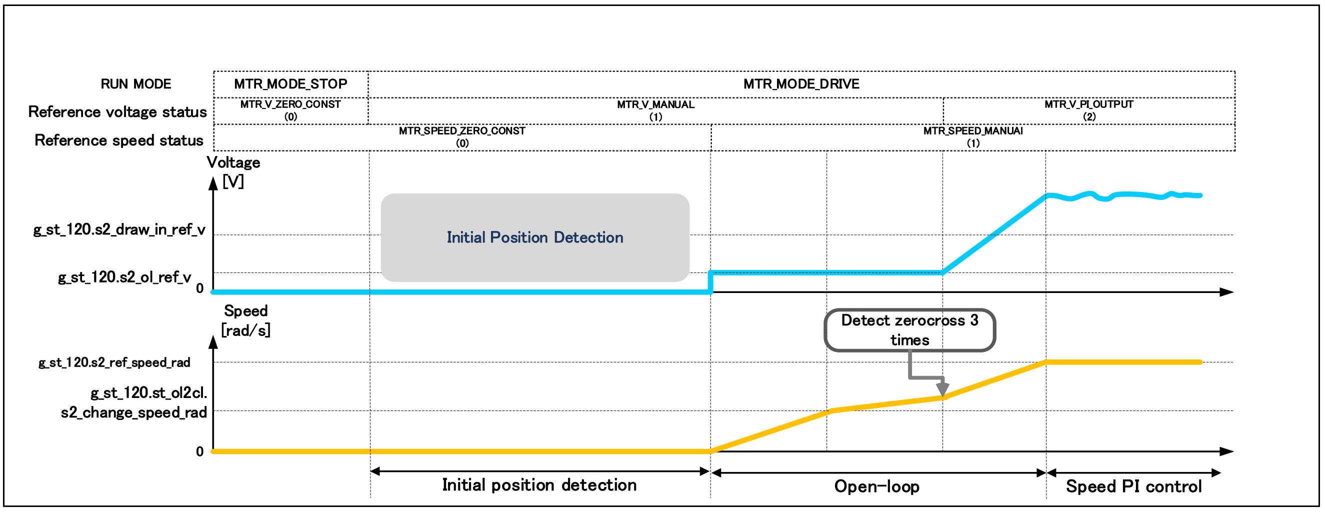


Figure 3-10 Startup sequence (Startup Using Initial Position Detection)

3.1.9 System protection function

This system has the following types of error status and emergency stop functions in case of occurrence of respective error. Refer to Table 3-8 for settings.

- Overcurrent error by external hardware

When an emergency stop signal (over current detection) from the external hardware is detected, voltage output is stopped.

- Overcurrent error by CMP0

Shunt current is monitored by CMP0 + PGA. When shunt current exceeds overcurrent, voltage output is stopped.

- Overvoltage error

The inverter bus voltage is monitored at the overvoltage monitoring cycle. When the inverter bus voltage exceeds overvoltage limit, voltage output is stopped. The threshold value of the overvoltage is set in consideration of the error of resistance value of the detection circuit.

- Low voltage error

The inverter bus voltage is monitored at the low voltage monitoring cycle. When the inverter bus voltage lowers undervoltage limit, voltage output is stopped. The threshold value of the low voltage is set in consideration of the error of resistance value of the detection circuit.

- Rotational speed error

The rotational speed is monitored at the rotational speed monitoring cycle. When the rotational speed exceeds the over speed limit, voltage output is stopped.

- Timeout error

The timeout counter is monitored at the timeout monitoring cycle. When pattern switching by Hall effect sensor interruption in Hall effect sensor control mode or zero-crossing of induced voltage in sensorless control mode don't happen for a timeout period, voltage output is stopped.

- Pattern error

The output voltage pattern is monitored at the pattern monitoring cycle. When unexpected pattern is detected in voltage pattern set from Hall effect sensor in Hall effect sensor control mode or induced voltage in sensorless control mode, voltage output is stopped.

- TRX overflow error

When TRX counter overflows at measuring period of current rising in initial position detection process, voltage output is stopped.

Table 3-8 Setting Value of Each System Protection Function

Kinds of error	Threshold	
Overcurrent error by external hardware	Over current limit [A]	2.0
Overcurrent error by CMP0	Over current limit [A]	1.47
Overvoltage error	Overvoltage limit [V]	28
	Monitoring cycle [ms]	1
Low voltage error	Low voltage limit [V]	15
	Monitoring cycle [ms]	1
Rotational speed error	Speed limit [rpm]	3975
	Monitoring cycle [ms]	1

3.1.10 PU system

The dynamic range of a motor control system based on fixed-point arithmetic needs to be determined appropriately. However, when there is a significant difference between the actual and assumed motor characteristics, due to the dynamic range mismatch, effects of overflow and rounding error become larger. In order to reduce the dependence of dynamic range on motor characteristics, PU (per-unit) system is used in this system.

PU value of a physical quantity is relative value to a base value, and can be derived as follows:

$$PU\ Value = \frac{Physical\ quantity}{Base\ Value}$$

Base value of a physical quantities and gains can be derived from the base current, base voltage, base frequency and base angle. For example, base resistance can be calculated from base voltage and base current:

$$Base\ Resistance = \frac{Base\ Voltage}{Base\ Current}$$

The base current, voltage and frequency should be configured with the motor characteristics in order to minimize the dependence of dynamic range on motor parameters. In this system, rated current, rated voltage and maximum electrical frequency (which derived from maximum velocity and number of pole pairs) are defined as the base current, base voltage and base frequency respectively.

Additionally, since calculations of PI control are in PU unit, gains used in this control system should be converted to PU values. The base values for the respective physical quantities are shown in Table 3-9. Typically, the same base value is used for the physical quantity in same unit. These values are defined in “r_mtr_scaling_parameter.h”.

Table 3-9 PU system base values

Item	Unit	Note
Current	[A]	Rated current (Motor spec)
Voltage	[V]	Rated voltage (Inverter board spec)
Frequency	[Hz]	Maximum rotational speed/ 60 seconds* Number of pole pairs
Angle	[rad]	Constant (2π)
Angular frequency	[rad/s]	Angle * Frequency
Time	[s]	1/Frequency
Resistance	[Ω]	Voltage/Current
Inductance	[H]	Resistance/Angular frequency
Induced voltage constant	[V/(rad/s)]	Voltage/Angular frequency
Torque	[Nm]	Induced voltage constant * Current
Inertia	[kgm^2/rad]	Torque / (Frequency * Angular frequency)
Kp	[V/(rad/s)]	Voltage/Angular frequency
Kidt	[V/(rad/s)]	Voltage/Angular frequency

As shown in Figure 3-11, after the voltage and current information is obtained from A/D conversion, the voltage and current value is converted to a PU unit system and used for control. Since the final output is a dimensionless PWM duty, unit conversion is not needed.

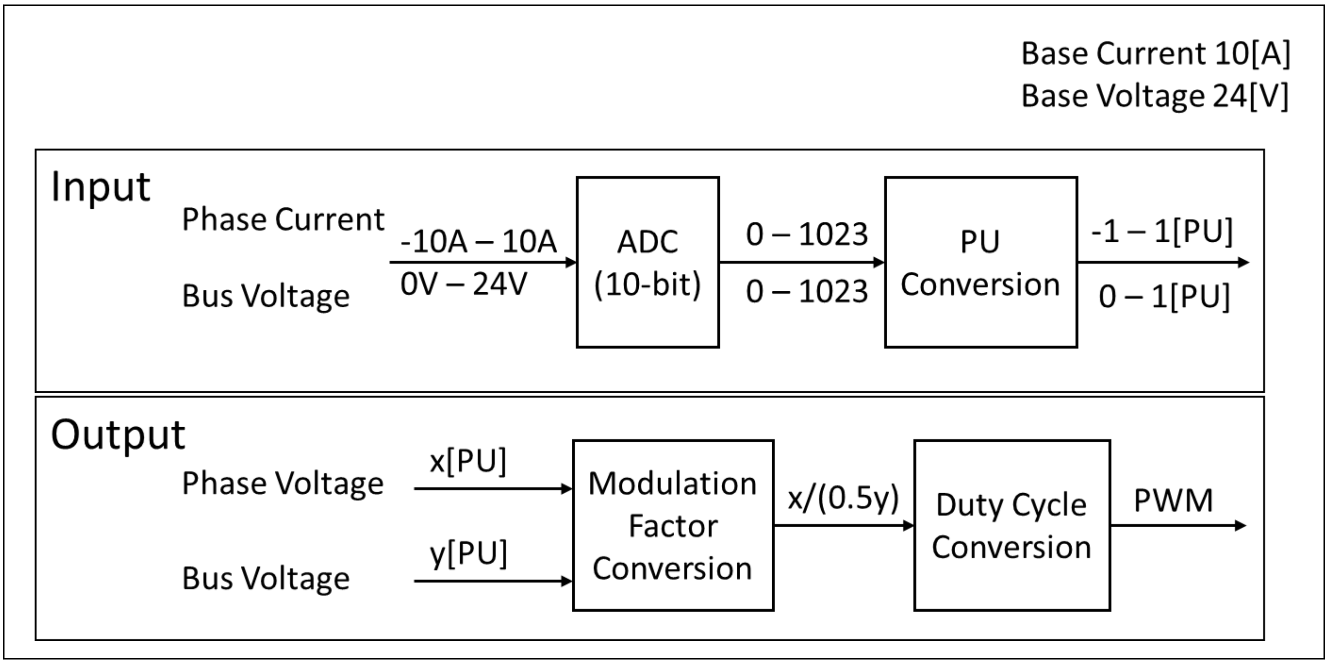


Figure 3-11 Motor control with using PU system

3.1.11 Using saliency to detect initial position during stop

A motor is said to have saliency if the rotor position causes a different amount of rotor magnet magnetic flux through each phase coil. In other words, the magnetic resistance changes according to the position of the rotor. If the magnetic resistance changes into a sine-wave shape, the inductance will also change into a sine-wave shape. As shown in Figure 3-12, inductance changes to twice the number of cycles for each lap of the rotor. In this case, when voltage is applied so that current flows from U→V, V→W, and W→U, the time it takes for the current flowing through the shunt to reach the threshold current value changes according to the position of the rotor. An example of this is shown in Figure 3-13. It therefore takes longer when voltage is applied in the V→W direction than when voltage is applied in the W→U direction.

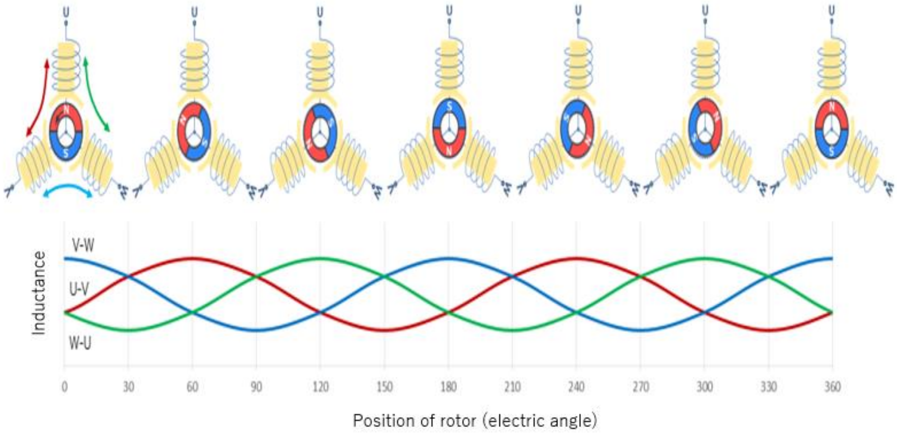


Figure 3-12 Changes in inductance according to rotor position

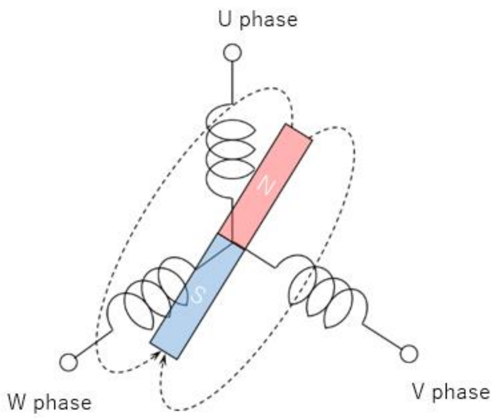


Figure 3-13 Relationship between rotor position and each phase

Here is a description of the rotor position detection method using this phenomenon. A diagram of the angle detection for salient motor used in this system is shown in Figure 3-14. It is distinguished by applying 3 patterns of voltage, measuring the time taken until the current that flows in the shunt resistor reaches the threshold current, and comparing these to detect which direction the rotor is facing each 60 degrees within the 180 degrees of electrical angle.

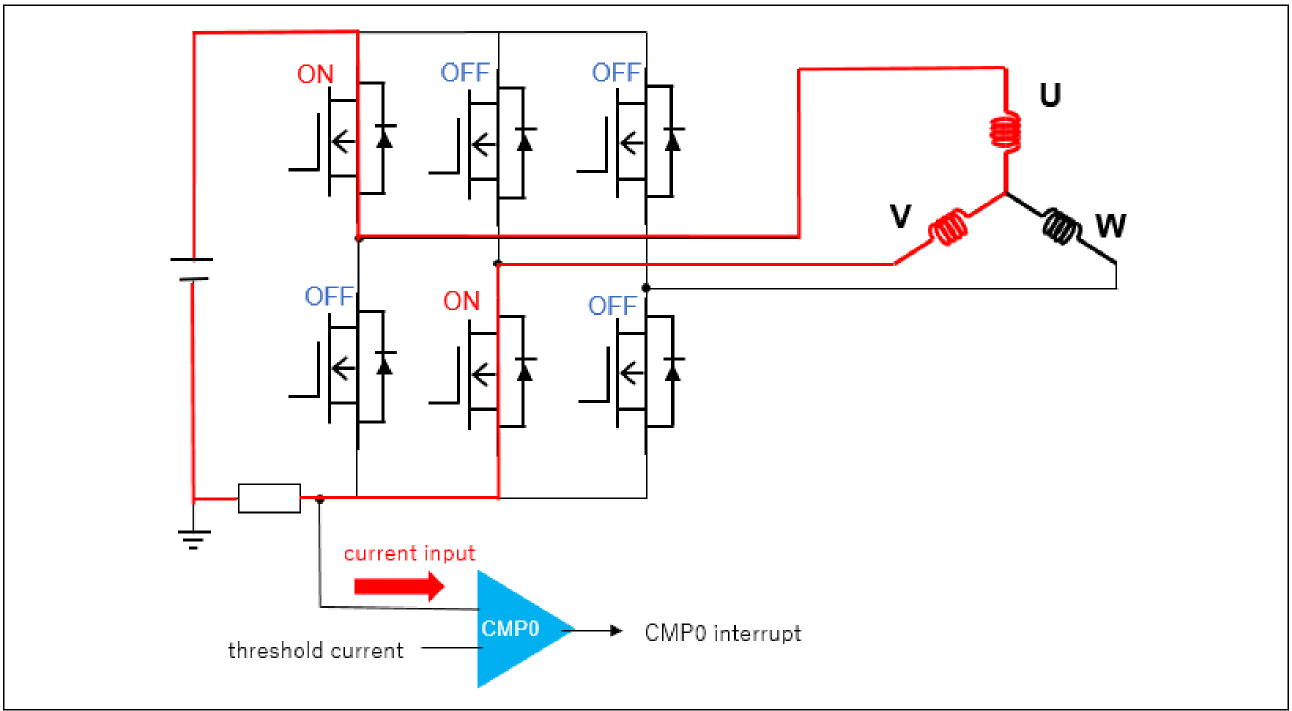


Figure 3-14 Angle detection diagram

The algorithm used in this system detects the time taken to reach the internal base current value using an RL78/G1F timer RX (TRX) and a comparator 0 (CMP0). It uses the TRD complementary PWM mode to apply pulse-shaped voltage to each phase. At this time, the count of the TRX starts at the rising edge synchronization of the TRD. The CMP0 generates an interrupt when it detects that the current flowing through the shunt resistor has reached the threshold current, and it measures the time required to increase the current.

Angle detection is performed every 60 degrees within the 180 degrees of electrical angle from the comparison with the cumulative time measured at each phase. Measurement stops when the cumulative measured time becomes greater than or equal to the threshold in the differential between the highest phase and lowest phase. However, if the differential does not reach the threshold after the maximum number of measurements, it is assumed that saliency is insufficient, and a transition is made to angle detection using saturation characteristics.

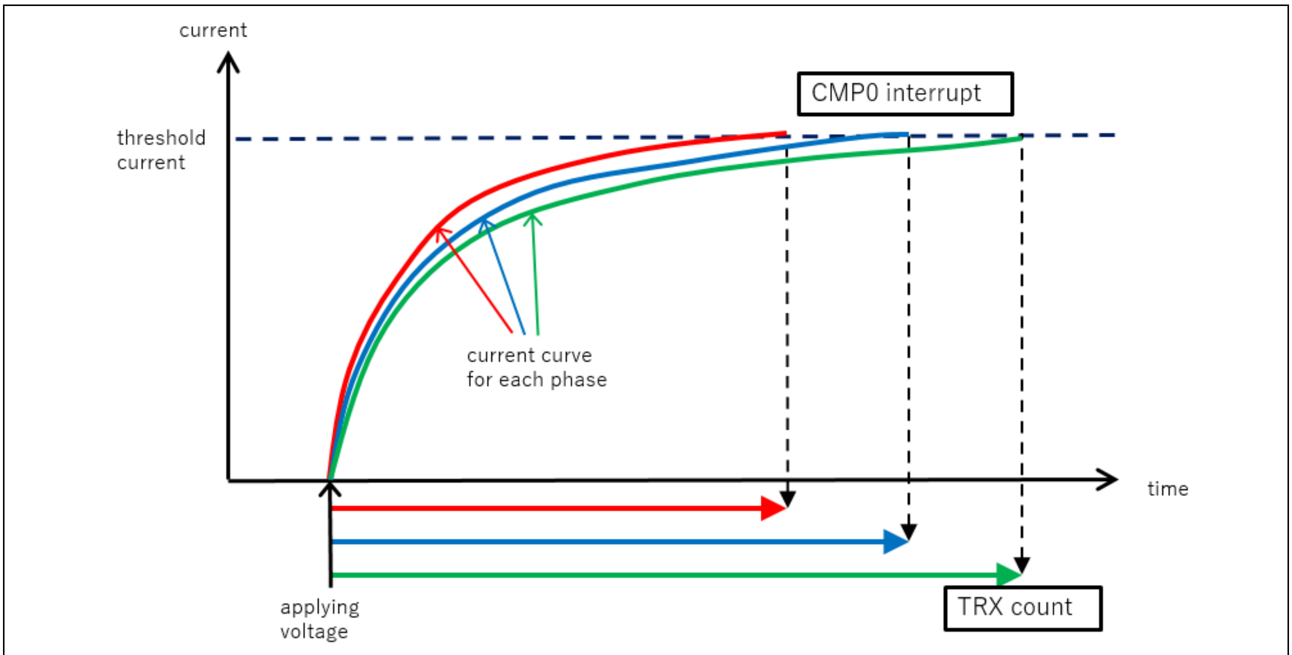


Figure 3-15 Current detection time differential among the 3 Phases

This rotor position detection method also takes measurements to confirm that the motor rotor has sufficient saliency to estimate its initial position. It compares the difference between the maximum value and median value with the difference between the median value and the minimum value of the TRX count, which increments with the current rise of the measured 3 phases, and identifies the maximum value or minimum value phases with the largest difference. Next, it applies voltage in the direction opposite to the identified phase and measures the time required for the current to rise. In this case, the mean value of the TRX count for the 2 phases is compared to the TRX count for the phase with the voltage orientation reversed. If the TRX count of the reversed phase has the same magnitude correlation, saliency is judged to be sufficient, and if it does not, saliency is judged to be too low.

For example, as shown in Figure 3-16, if the rotor is oriented in the 120-degree direction, the phase identified will be the W-U phase because the difference between the maximum value and the median value is greater. It reverses the energization direction from the W-U phase, applies voltage to the U-W phase, and measures the time the current rises. It compares the median value of the U-V, V-W, and W-U phase TRX counts to the U-W phase TRX counts. If the U-W phase TRX count is greater, initial position detection using saliency is judged to be possible, but if it is lower, it is judged not to be possible.

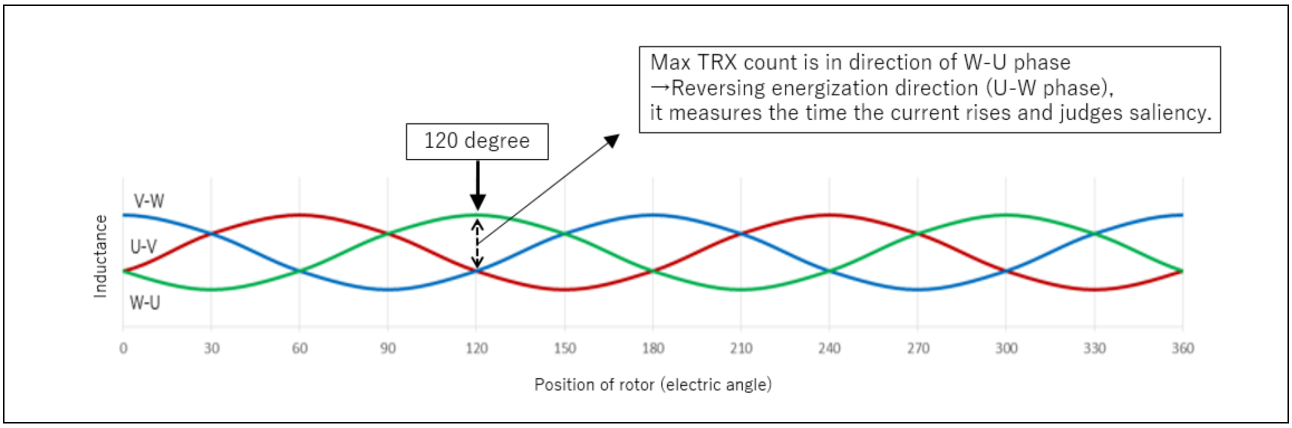


Figure 3-16 Saliency confirmation method

3.1.12 Initial position detection during stop using saturation characteristics

Since the method described above uses the change in inductance due to saliency to estimate the position, it is not possible to determine the polarity (for example, there is no distinction between 60 degrees and 240 degrees). Also, it cannot be applied when a non-salient motor is used. Here, the magnetic saturation characteristics of the motor are used for polarity detection and angle detection with a non-salient rotor.

Due to the limited amount of magnetization that a magnetic material can have, if current is applied to a coil to generate an external magnetic field around the core of the coil, the core goes into a state of saturated magnetization when the external magnetic field exceeds a certain value. If the direction of the external magnetic field through the core is the same as the orientation of the magnetic field generated by the current flowing into the coil, the inductance becomes smaller because magnetization is more saturated than if the directions were opposite. These characteristics are used to judge the orientation of the magnetic pole.

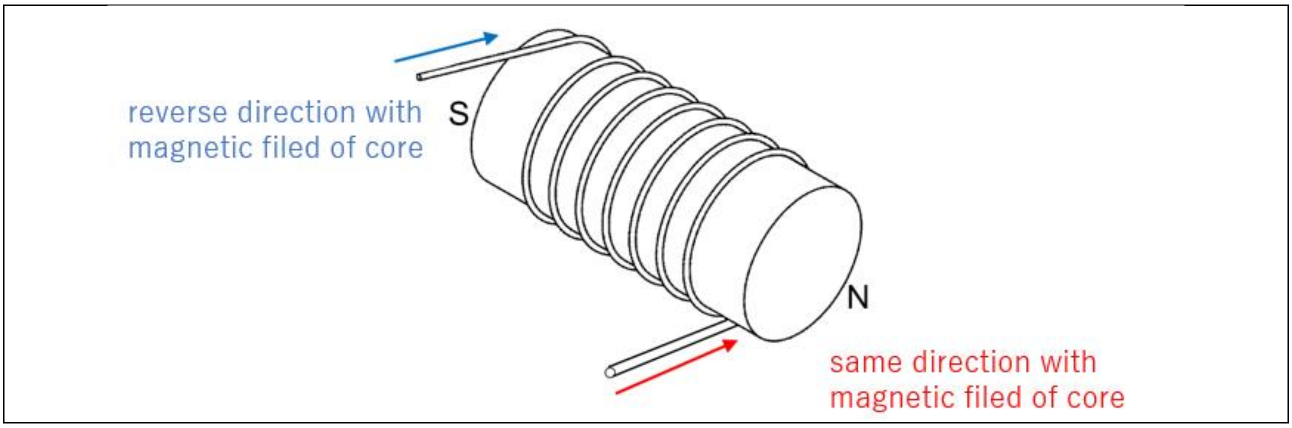


Figure 3-17 Example of magnetic pole wound with coil

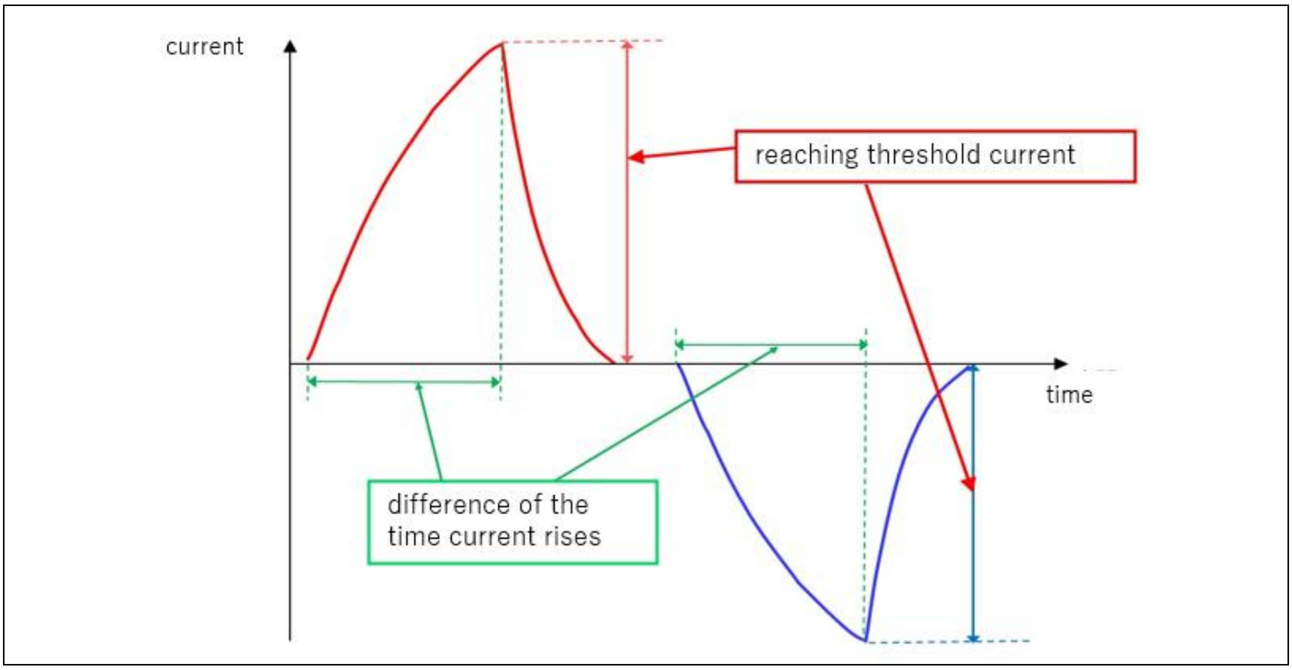


Figure 3-18 Current differential according to direction of applied current

Voltage is applied to the motor as shown in Figure 3-18, and the time required for the current flowing in the shunt resistor to rise is measured by TRX in the same way as it is measured when using saliency. For measurements using saturation characteristics, the TRX count is lowest when the direction of application of the voltage matches the direction of rotation, so this tendency is used to estimate the orientation of the rotor.

If the motor is judged to be salient, in order to detect polarity of rotor, voltage is applied forward and backward based on angle information obtained using saliency. The current rise time is measured, and the polarity of the rotor is determined by comparing the magnitude correlation. If the motor is judged to be non-salient, the current rise time is measured by applying voltage in 6 directions, and the rotor position is estimated as if the rotor were oriented toward the phase with the minimum TRX count value.

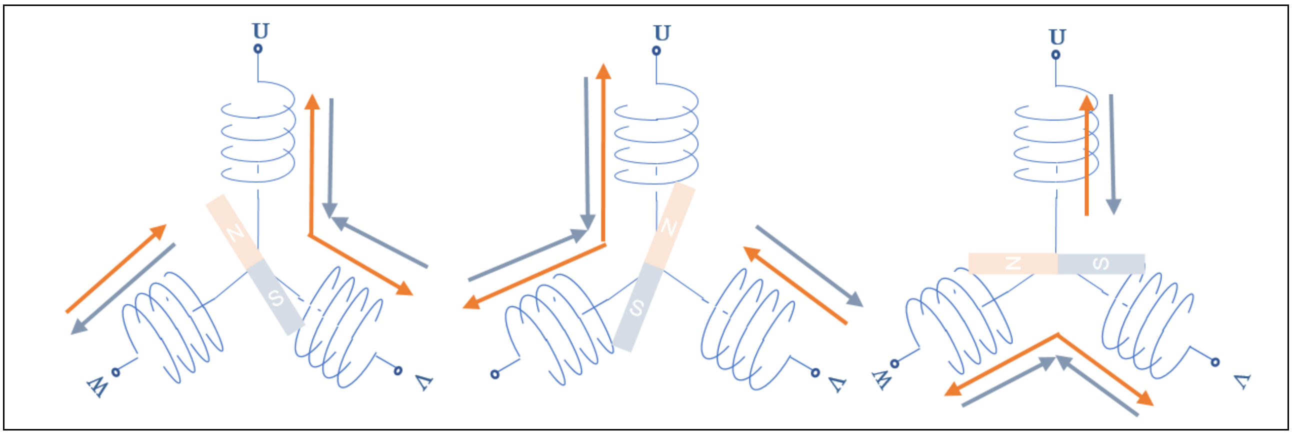


Figure 3-19 Example of applied voltage pattern due to initial position detection using magnetic saturation

Measurement stops when the differential among the directions exceeds the discrimination threshold. However, when the differential integral value does not reach the discrimination threshold even when measurement is carried out at the maximum measurement frequency, if the differential integral value is greater than the discrimination threshold, it is judged as an initial position detection succeeded. if the differential integral value is less than the discrimination threshold, judged as an initial position detection failure.

3.1.13 Zero-crossing detection by Comparator

In the sample programs, at high speed (over 1855[rpm]) the zero-crossing points are detected by CMP1. At that time, it should synchronize the detection timing and PWM ON timing to avoid the effect of other phases. Therefore, we use “TIMER WINDOW” which is generated with TAU channel0 and channel2 to mask the output of CMP1.

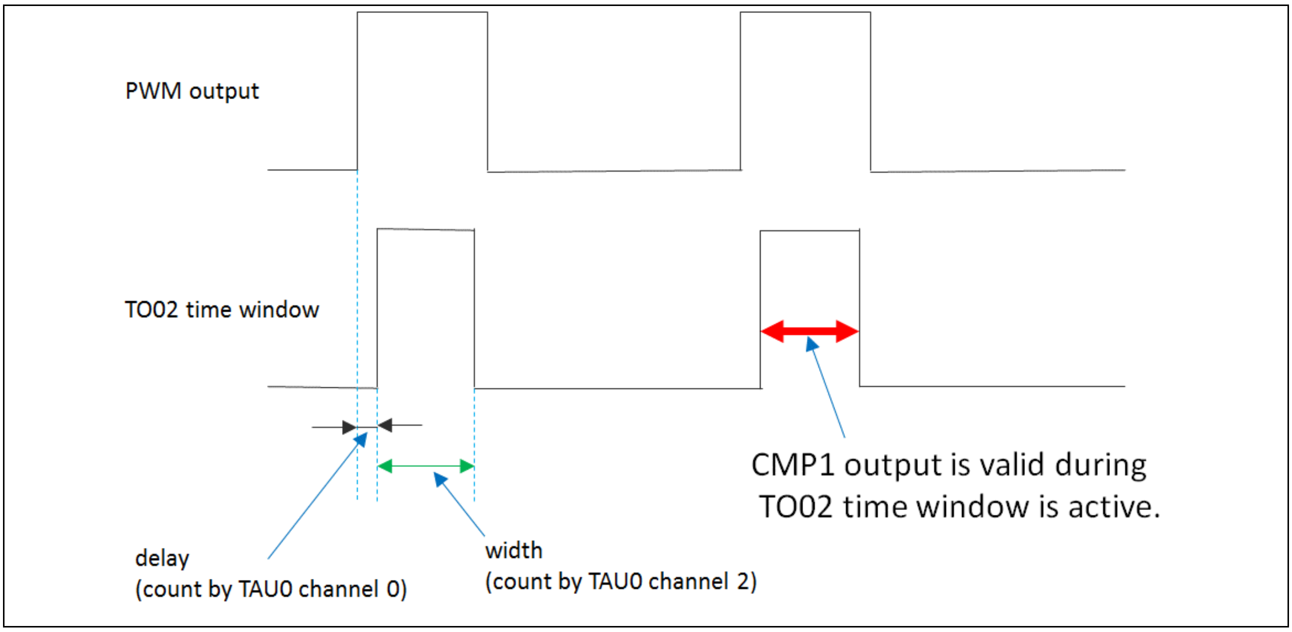


Figure 3-20 Setting of valid periods for CMP1

The change of the method to detect zero-crossing point by using A/D converter or comparator depends on the current estimated rotational speed. To avoid frequent change of the detection method at border value (1855 [rpm]), a hysteresis is provided. (The change from comparator to A/D converter is performed at 1802[rpm].)

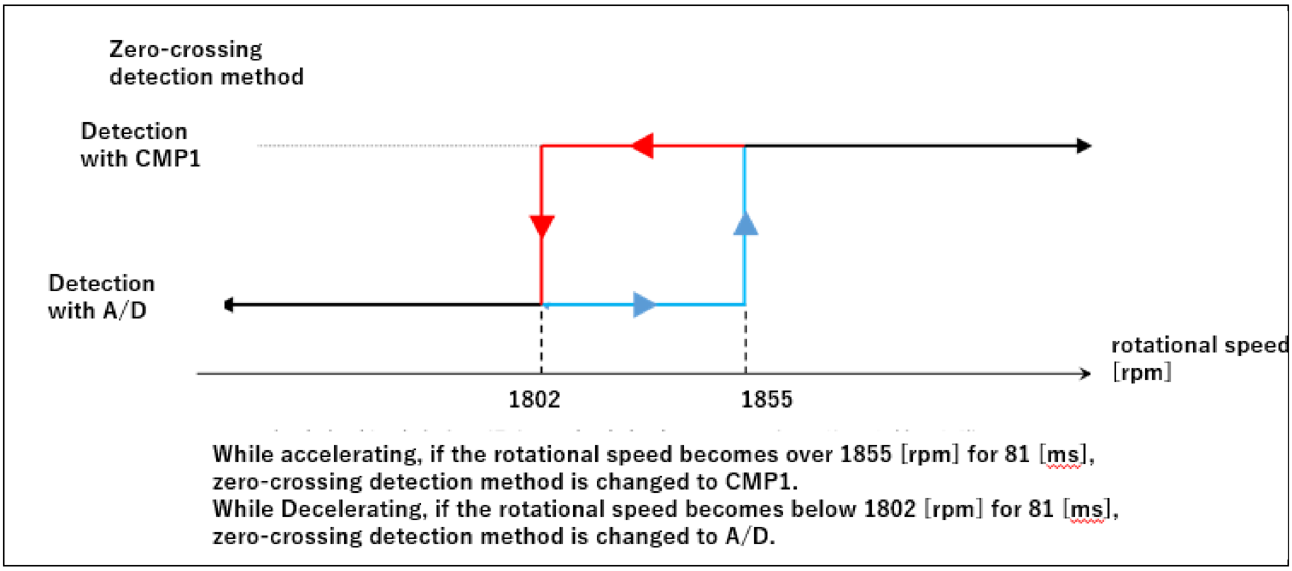


Figure 3-21 Change the method to detect zero-crossing

3.2 Function specifications of 120-degree conducting control software

Lists of functions used in this control program are shown below. Functions not used in this system are undescribed.

Table 3-10 List of Functions "main.c"

file	function	process
main.c	main argument: none return: none	Initialization and main loop ・initialization ⇒initialization of hardware ⇒initialization of system variables ⇒initialization of RMW communication ⇒initialization of control system ⇒reset process ⇒waiting for stability of bus voltage ・main loop ⇒system control depending on input from UI ⇒LED control ⇒clear watch dog timer
	ics_ui argument: none return: none	Process for RMW UI (GUI) ・input values of command variables to RMW variables ・change motor status depending on input event ・initialization of system variables when reset event occurs
	board_ui argument: none return: none	Process of board UI (H/W) ・change motor status depending on state of switch ・determination of command rotational speed by value of VR1
	software_init argument: none return: none	Initialization of system variables ・initialization of variables for main process ・initialization of RMW variables

Table 3-11 List of Functions "r_mtr_ics.c"

file	function	process
r_mtr_ics.c	mtr_set_com_variables argument: none return: none	Preprocess to set control variables ・input values of command variables to RMW variables ・input values of RMW variables to RMW buffer variables
	mtr_ics_variables_init argument: none return: none	Initialization of command variables
	R_MTR_Limit argument: (int16_t) s2_value / target value (int16_t) s2_max / maximum limit (int16_t) s2_min / minimum limit return: (int16_t) s2_temp / limited value	Limit between maximum and minimum values

Table 3-12 List of Functions "r_mtr_board.c"

file	function	process
r_mtr_board.c	mtr_board_led_control argument: (uint8_t) u1_motor_status / motor status (uint8_t) u1_system_status / system status (uint8_t) u1_zc_detection_method / zero-crossing detection method return: none	LED control depending on motor status, system mode and zero-crossing detection method
	mtr_remove_chattering argument: (uint8_t) u1_sw / switch signal (uint8_t) u1_on_off / switch status return: (uint8_t) u1_flag_chattering / flag for chattering	Remove chattering of switch signal

Table 3-13 List of Functions "r_mtr_ctrl_mrssk.c"

file	function	process
r_mtr_ctrl_mrssk.c	R_MTR_GetVr1Ad argument: none return: (uint16_t) u2_ad_data / AD value of VR1	Get VR1 value
	R_MTR_GetSw1 argument: none return: (uint8_t) MTR_PORT_SW1 / state of SW1	Get state of SW1
	R_MTR_GetSw2 argument: none return: (uint8_t) MTR_PORT_SW2 / state of SW2	Get state of SW2
	R_MTR_Led1On argument: none return: none	Turn LED1 on
	R_MTR_Led2On argument: none return: none	Turn LED2 on
	R_MTR_Led3On argument: none return: none	Turn LED3 on
	R_MTR_Led1Off argument: none return: none	Turn LED1 off
	R_MTR_Led2Off argument: none return: none	Turn LED2 off
	R_MTR_Led3Off argument: none return: none	Turn LED3 off

Table 3-14 List of Functions “r_mtr_ctrl_rl78g1f.h”

file	function	process
r_mtr_ctrl_rl78g1f.h	R_MTR_ ClearTrd0Imfa() argument: TRDSR0_bit.no0 = 0 return: none	Clear IMFA flag of TRD
	R_MTR_ ClearTrgImfa() argument: TRGIMFA = 0 return: none	Clear IMFA flag of TRG
	R_MTR_SetTdr03(cnt) argument: TDR03 = cnt return: none	Set delay timer count
	R_MTR_StartDelayCnt() argument: TS0L_bit.no3 = 1 return: none	Start delay timer
	R_MTR_StopDelayCnt() argument: TT0L_bit.no3 = 0 return: none	Stop delay timer
	R_MTR_ClearInttm03() argument: TMIF03 = 0 return: none	Clear interruption flag of delay timer
	R_MTR_ClearWdt() argument: WDTE = 0xAC return: none	Clear watchdog timer (WDT)
	R_MTR_SetTrdgrd0(duty) argument: TRDGRD0 = ((uint16_t)MTR_CARRIER_SET – duty)) return: none	Set TRD general resister D0 for PWM duty
	R_MTR_SetTrdgrc1(duty) argument: TRDGRC1 = ((uint16_t)MTR_CARRIER_SET – duty)) return: none	Set TRD general resister C1 for PWM duty
	R_MTR_SetTrdgrd1(duty) argument: TRDGRD1 = ((uint16_t)MTR_CARRIER_SET – duty)) return: none	Set TRD general resister D1 for PWM duty
	R_MTR_SetCmp1Window(width) argument: TDR02 = width return: none	Set CMP1 timer window
	R_MTR_SetC0rvm(i) argument: C0RVM = i return: none	Set reference current for CMP0
	R_MTR_SetInSelCmp() argument: OPCTL0 &= 0xE7, OPCTL0 |= 0x08 return: none	Set cutoff source of PWMOPA for CMP0
	R_MTR_ClearInSelCmp() argument: OPCTL0 &= 0xE7 return: none	Clear cutoff source of PWMOPA
	R_MTR_StartTrx() argument: TSTART = 1 return: none	Start TRX
	R_MTR_StopTrx() argument: TSTART = 0 return: none	Start TRX
	R_MTR_StartTrd() argument: TRDSTR = 0x0F return: none	Start TRD
	R_MTR_StopTrd() argument: TRDSTR = 0x00 return: none	Stop TRD

Table 3-15 List of Functions “r_mtr_ctrl_rl78g1f.c” [1/2]

file	function	Process
r_mtr_ctrl_rl78g1f.c	R_MTR_InitHardware argument: none return: (uint16_t) u2_init_hw_error / error status	Initialization of peripheral functions
	mtr_init_unused_pins argument: none return: none	Initialization of unused pins
	mtr_init_ui argument: none return: none	Initialization of ports for board UI
	mtr_init_clock argument: (uinit16_t) u2_change_clk_error / error status return: none	Initialization of clock
	mtr_init_tau argument: none return: none	Initialization of TAU
	mtr_init_trd argument: none return: none	Initialization of TRD
	mtr_init_trg argument: none return: none	Initialization of TRG
	mtr_init_trx argument: none return: none	Initialization of TRX
	mtr_init_intp argument: none return: none	Initialization of external interrupt (INTP0)
	mtr_init_cmp0_pga argument: none return: none	Initialization of CMP0 and PGA
	mtr_init_cmp1 argument: none return: none	Initialization of CMP1
	mtr_init_ad_converter argument: none return: none	Initialization of A/D converter
	mtr_init_vuvw_filter argument: none return: none	Initialization of low pass filter on CPU card for voltage
	R_MTR_RecoverForcedShutdow argument: none return: none	Recover from forced shutdown by PWMOPA
	R_MTR_EnableCmp0Intr argument: none return: none	Enable the interrupt of CMP0
	R_MTR_DisableCmp0Intr argument: none return: none	Disable the interrupt of CMP0
	R_MTR_ResetTrd argument: none return: none	Reset TRD

Table 3-16 List of Functions "r_mtr_ctrl_rl78g1f.c" [2/2]

file	function	Process
r_mtr_ctrl_rl78g1f.c	R_MTR_EnableCmp1Intr argument: none return: none	Enable the interrupt of CMP1
	R_MTR_DisableCmp1Intr argument: none return: none	Disable the interrupt of CMP1
	R_MTR_SelectCmp1Input argument: (uint8) u1_p_input / plus side input of CMP1 (uint8) u1_m_input / minus side input of CMP1 return: none	Set the detected signal pattern of CMP1
	R_MTR_SetTauWindowSignal argument: (uint8)u1_signal / signal for TIMER WINDOW return: none	Set the target PWM signal for TIMER WINDOW of CMP1
	R_MTR_GetAdc argument: (uint8_t) u1_ad_ch / channel of A/D conversion return: (int16_t) s2_ad_value / result of A/D conversion	Get the result of A/D conversion
	R_MTR_GetVdcAdc argument: (int16_t) *s2_v_dc / bus voltage return: none	Get the results of A/D conversion of bus voltage
	R_MTR_GetVuvwAdc argument: (int16_t) *s2_v_uvw / UVW voltages return: none	Get the results of A/D conversion of U, V and W voltage
	R_MTR_GetIuAdc argument: (uint16_t) u2_current_offset_cnt / number of measurements return: none	Get the sum of results of A/D conversion of U current offset
	R_MTR_PrepareEnergizePhase argument: (uint8_t) u1_energized_phase / number of energized phases (uint8_t) u1_v_pattern / voltage pattern return: none	Set voltage pattern for initial position detection
	R_MTR_OutputStop argument: none return: none	Stop output of voltage
	R_MTR_LowerArmOn argument: (uint16_t) u2_low_on_period / period for lower arms on return: none	Set all lower arms on
	R_MTR_CtrlStop argument: none return: none	Stop motor control ・stop output of voltage ・stop delay timer
	R_MTR_PatternOutput argument: (uint16_t) u2_pattern / conducting pattern (uint16_t) u2_pwm_duty / duty return: (uint8_t) u1_temp_error_flag / error flag	Set TRD resisters to output PWM ・select output ports ・set PWM duty

Table 3-17 List of Functions “r_mtr_driver_access.c”

file	function	process
r_mtr_driver_access.c	R_MTR_InitControl argument: none return: none	Initialization of control system ・initialization of motor status ・initialization of control variables
	R_MTR_IcsInput argument: (mtr_ctrl_input_t) *st_ics_input / RMW structure return: none	Input values of RMW variables to RMW buffer variables
	R_MTR_SetVariables 【inline function】 argument: none return: none	Input values of RMW buffer variables to control variables
	R_MTR_InputBuffParamReset argument: none return: none	Reset RMW buffer variables
	R_MTR_ExecEvent argument: (uint8_t) u1_event / event return: none	Change motor status and execute event process
	R_MTR_GetStatus argument: none return: (uint8_t) mtr_statemachine_get_status(g_st_120.st_stm) / motor status	Get motor status
	R_MTR_GetErrorStatus argument: none return: (uint16_t) g_st_120.u2_error_status / error status	Get error status
	R_MTR_GetDir argument: none return: (uint8_t) g_st_120.u1_dir / direction of rotation	Get direction of rotation
	R_MTR_GetZcDetectionMethod argument: none return: (uint8_t)g_st_120.u1_zc_detection_method / zero-crossing detection method	Get zero-crossing detection method
	R_MTR_SetSpeed argument: (int16_t) s2_ref_speed_rpm / command rotational speed return: (uint8_t) u1_stop_req / flag for requiring flag	Set command rotational speed
	R_MTR_ChargeCapacitor argument: none return: (uint16_t) u2_charge_cap_error / timeout error	Waiting for stability of bus voltage
	R_MTR_UpdatePolling argument: none return: none	Set control variables

Table 3-18 List of Functions "r_mtr_statemachine.c"

file	function	process
r_mtr_statemachine.c	mtr_statemachine_init argument: (st_mtr_statemachine_t) *p_state_machine / structure for motor status return: none	Initialization of motor status
	mtr_statemacine_reset argument: (st_mtr_statemachine_t) *p_state_machine / structure for motor status return: none	Reset motor status
	mtr_state_machine_event argument: (st_mtr_statemachine_t) *p_state_machine / structure for motor status (void) *p_object / structure for control variables (uint8_t) u1_event / event return: none	Execute event process
	mtr_statemachine_get_status argument: (st_mtr_statemachine_t) *p_state_machine / structure for motor status return: (uint8_t) p_state_machine->u1_status / motor status	Get motor status
	mtr_act_none argument: (st_mtr_statemachine_t) *st_stm / structure for motor status (void) *p_param / structure for control variables return: none	No process is performed
	mtr_act_init argument: (st_mtr_statemachine_t) *st_stm / structure for motor status (void) *p_param / structure for control variables return: none	Initialization of control variables
	mtr_act_error argument: (st_mtr_statemachine_t) *st_stm / structure for motor status (void) *p_param / structure for control variables return: none	Stop motor control
	mtr_act_drive argument: (st_mtr_statemachine_t) *st_stm / structure for motor status (void) *p_param / structure for control variables return: none	Reset control variables
	mtr_act_stop argument: (st_mtr_statemachine_t) *st_stm / structure for motor status (void) *p_param / structure for control variables return: none	Stop motor control

Table 3-19 List of Functions "r_mtr_120.c"

file name	function	process
r_mtr_120.c	mtr_120_motor_default_init argument: (st_mtr_120_control_t) *st_120 / structure for control variables return: none	Initialization of control variables
	mtr_120_motor_reset argument: (st_mtr_120_control_t) *st_120 / structure for control variables return: none	Reset control variables

Table 3-20 List of Functions "r_mtr_ctrl_gain_calc.c"

file name	function	process
r_mtr_ctrl_gain_calc.c	mtr_ctrl_gain_calc argument: (st_mtr_ctrl_gain_t) *st_gain / structure for PI control (st_mtr_design_parameter_t) *st_ctrl_param / structure for PI gain calculation return: none	PI gain calculation

Table 3-21 List of Functions “r_mtr_interrupt.c” [1/3]

file	function	process
r_mtr_interrupt.c	mtr_over_current_interrupt argument: none return: none	Overcurrent detection process ・execute error event ・set error status
	mtr_cmp0_interrupt argument: none return: none	・overcurrent detection process ・Get TRX count during initial position detection
	mtr_carrier_interrupt argument: none return: none	Carrier interruption (50 µs) ・obtain bus and UVW voltages ・initial position detection ・draw-in process ・zero-crossing detection ・transition from open-loop drive to sensorless control ・rotational speed calculation ・open-loop process ・set delay timer ・set CMP1 ・RMW communication
	mtr_measure_current_offset 【inline function】 argument: none return: none	Measure U phase current offset
	mtr_speed_calc 【inline function】 argument: none return: none	Calculation of rotational speed
	mtr_set_chopping_pattern 【inline function】 argument: (uint16_t) u2_pattern / voltage pattern return: none	Set chopping pattern
	mtr_ics_interrupt_process 【inline function】 argument: none return: none	RMW communication
	mtr_set_speed_ref 【inline function】 argument: none return: none	Set reference speed
	mtr_set_voltage_ref 【inline function】 argument: none return: none	Set reference voltage
	mtr_pi_ctrl 【inline function】 argument: (st_mtr_pi_control_t) *pi_ctrl / PI control structure return: (int16_t) s2_ref_v_delta / variation of output voltage	PI control process (velocity form)
	mtr_duty_calc 【inline function】 argument: (int16_t) s2_ref_v / reference voltage (int16_t) s2_vdc_ad / bus voltage return: (uint16_t) u4_temp / duty	Duty calculation
	mtr_abs 【inline function】 argument: (int16_t) s2_value / input value return: (int16_t) s2_temp / conversion value	Conversion to absolute value
	mtr_limit_value 【inline function】 argument: (int16_t) s2_value / input value (int16_t) s2_limit_value / limit value return: (int16_t) s2_temp / conversion value	Limit process

Table 3-22 List of Functions “r_mtr_interrupt.c” [2/3]

file	function	process
r_mtr_interrupt.c	mtr_error_check 【inline function】 argument: none return: none	Error check
	mtr_1ms_interrupt argument: none return: none	TRG interruption (1 ms) ・change reference speed and voltage states ・calculate reference speed and voltage ・check error
	mtr_delay_interrupt argument: none return: none	Delay timer interruption ・stop delay timer ・set voltage pattern
	mtr_cmp1_interrupt argument: none return: none	・Calculates rotational speed ・Set delay timer ・Set voltage pattern
	mtr_draw_in_pattern_set 【inline function】 argument: none return: none	Set voltage pattern in draw-in state
	mtr_detect_zerocross 【inline function】 argument: (st_mtr_sensorless_control_t) *st_less / structure for control variables (uint16_t) *u2_cnt_timeout / timeout counter return: (uint16_t) u2_temp_signal / voltage pattern	Estimate position of rotor from zero-crossing of induced voltage
	mtr_drive_openloop 【inline function】 argument: none return: none	Open-loop drive process
	mtr_set_angle_shift 【inline function】 argument: none return: none	Calculation of delay count after zero-crossing
	mtr_start_delay_timer 【inline function】 argument: (uint16_t) u2_delay_count / delay count return: none	Start delay timer
	mtr_stop_delay_timer 【inline function】 argument: none return: none	Stop delay timer
	mtr_openloop_pattern_set 【inline function】 argument: none return: (uint8_t) u2_pattern / voltage pattern	Set voltage pattern at open-loop drive
	mtr_ol2cl_ctrl 【inline function】 argument: none return: none	Transitions from open-loop to sensorless control ・Compares 3-phase voltage to threshold voltage ・Counts numbers of zero crossings during open-loop ・Detects rotor phase advance/delay

Table 3-23 List of Functions "r_mtr_interrupt.c" [3/3]

file	function	process
r_mtr_interrupt.c	mtr_get_bemf_voltage 【inline function】 argument: none return: (int16_t) s2_bemf_voltage / induced voltage	Get induced voltage
	mtr_openloop_phase_ctrl 【inline function】 argument: none return: none	Detect rotor phase advance/delay
	mtr_measure_inductance_effect 【inline function】 argument: (uint8_t) u1_energized_phase / number of energized phases (uint8_t) u1_v_pattern / voltage pattern return: (uint16_t) u2_trx_cnt / TRX count	Measures the time taken to reach threshold current during initial position detection
	mtr_salient_detect_angle 【inline function】 argument: none return: none	Angle detection process for initial position detection with a salient motor
	mtr_salient_detect_polarity 【inline function】 argument: none return: none	Polarity detection process for initial position detection with a salient motor
	mtr_non_salient_detect_angle 【inline function】 argument: none return: none	Angle detection process for initial position detection with a non-salient motor
	mtr_set_initial_position 【inline function】 argument: none return: none	Set first open-loop pattern depending on the result of initial position detection
	mtr_set_chopping_pattern_cmp 【inline function】 argument: none return: none	Set voltage pattern at sensorless control with CMP1
	mtr_change_cmp1_input 【inline function】 argument: none return: none	Switches input signal when CMP1 detects zero crossing
	mtr_ad2cmp_set_signal 【inline function】 argument: (uint8_t) u1_bemf_voltage / induced voltage value return: none	Sets voltage pattern when changing zero crossing detection system from A/D converter to CMP1
	mtr_cmp2ad_set_signal 【inline function】 argument: (uint8_t) u1_bemf_voltage / signal number return: none	Sets voltage pattern when changing zero crossing detection system from CMP1 to A/D converter
	mtr_set_cmp0_overcurrent 【inline function】 argument: none return: none	Set CMP0 for overcurrent

Table 3-24 List of Functions "r_dsp.h"

file	function	process
r_dsp.h	FIX_fromfloat argument: (float) x / input value n / Q-format return: (int16_t) y / conversion value	Convert float type value x to fixed-point value y in Qn format
	FIX32_fromfloat argument: (float) x / input value n / Q-format return: (int32_t) y / conversion value	Convert float type value x to fixed-point value y in Qn format

3.3 Lists of variables of sensorless 120-degree conducting control software

Lists of variables used in this control program are shown below. However, note that the local variables are not mentioned.

In the sample programs, fixed-point number is used for calculation. Therefore, in advance, some control variables are set in fixed-point number. Bits number in fractional part of fixed-point number is expressed in the Q format. “Qn” means n bits left shift.

Table 3-25 List of variables “main.c”

variable	type	Qn	PU	content	remarks
g_u1_system_mode	static uint8_t	Q0	-	Mode system management	
g_u1_motor_status	static uint8_t	Q0	-	Motor status management	
g_u1_reset_req	static uint8_t	Q0	-	Reset command flag for SW2	
g_u1_stop_req	static uint8_t	Q0	-	Stop command flag for VR1	
g_u1_flag_ui_change	static uint8_t	Q0	-	UI changing flag	
g_u1_error_status	static uint8_t	Q0	-	Error status management	
g_u1_zc_detection_method	static uint8_t	Q0		Zero-crossing detection method	
g_u2_conf_hw	uint16_t	Q0	-	RMW configuration variables	
g_u2_conf_sw	uint16_t	Q0	-		
g_u2_conf_tool	uint16_t	Q0	-		
gui_u1_active_gui	uint8_t	Q0	-		
g_u2_conf_sw_ver	uint16_t	Q0	-		
com_s2_sw_userif	int16_t	Q0	-	Management variable for UI	0: ICS_UI 1: BOARD UI
g_s2_sw_userif	int16_t	Q0	-		
com_u1_run_event	uint8_t	Q0	-	Input event and change run mode	0: MTR_EVENT_STOP 1: MTR_EVENT_DRIVE 2: MTR_EVENT_ERROR 3: MTR_EVENT_RESET
g_u1_run_event	uint8_t	Q0	-		
g_u2_system_error	uint16_t	Q0	-	System error management	

Table 3-26 List of variables “r_mtr_board.c”

variable	type	scale	PU	content	remarks
u1_sw_cnt	static uint8_t	Q0	-	Counter for judgement of chattering	

Table 3-27 List of variables "r_mtr_ics.c"

variable	type	Qn	PU	content	remarks
com_u1_direction	uint8_t	Q0	-	Direction of rotation	0 : CW 1 : CCW
com_u2_mtr_pp	uint16_t	Q0	-	Number of pole pairs	
com_f4_mtr_r	float	-	-	Resistance [Ω]	
com_f4_mtr_ld	float	-	-	D-axis inductance[H]	
com_f4_mtr_lq	float	-	-	Q-axis inductance[H]	
com_f4_mtr_m	float	-	-	Induced voltage constant [Vs/rad]	
com_f4_mtr_j	float	-	-	Rotor inertia[kgm^2]	
com_s2_ref_speed_rpm	int16_t	Q0	-	Command rotational speed ［rpm］	Mechanical angle
com_s2_ol2lcl_speed_rpm	int16_t	Q0	-	Speed allowed to transition to PI control[rpm]	Mechanical angle
com_f4_draw_in_ref_v	float	-	-	Command voltage at draw-in [V]	
com_f4_ol_ref_v	float	-	-	Command voltage at open-loop drive[V]	
com_s2_ramp_limit_speed_rpm	int16_t	Q0	-	Limit of acceleration [rpm/ms]	Mechanical angle
com_s2_ol2cl_ramp_speed_rpm	int16_t	Q0	-	Acceleration at transition to PI control [rpm/ms]	Mechanical angle
com_f4_ramp_limit_v	float	-	-	Limit of variation of voltage [V/ms]	
com_s2_angle_shift_adjust	int16_t	Q0	-	Adjust delay counts	
com_f4_detect_angle_current	float	-	-	Threshold current of angle detection for salient rotor	【IPD】
com_u4_detect_angle_thsld	uint32_t	Q0	-	TRX count value differential of angle detection for salient rotor	
com_u1_detect_angle_max_cnt	uint8_t	Q0	-	Maximum number of times salient rotor angle detection is measured	
com_f4_detect_polarity_current	float	-	-	Threshold current of polarity detection for salient rotor	
com_u4_detect_polarity_thsld	uint32_t	Q0	-	TRX count value differential of polarity detection for salient rotor	
com_u1_detect_polarity_max_cnt	uint8_t	Q0	-	Maximum number of times salient rotor polarity detection is measured	
com_f4_non_salient_current	float	-	-	Threshold current of angle detection for non-salient rotor	
com_u4_non_salient_thsld	uint32_t	Q0	-	TRX count value differential of angle detection for non-salient rotor	
com_u1_non_salient_max_cnt	uint8_t	Q0	-	Maximum number of times non-salient rotor angle detection is measured	
com_u2_current_offset_cnt	uint16_t	Q0	-	Number of measuring current offset	【IPD】 【CMP overcurrent】
com_s2_zc_detect_change_speed_rpm	int16_t	Q0	-	Speed of conversion of zero crossing detection system from A/D converter to CMP1 [rpm]	【CMP zero-crossing】
com_u2_zc_change_judge_cnt	uint16_t	Q0	-	Judgement count for converting zero-crossing detection methods	
com_f4_asr_omega_hz	float	-	-	Frequency for speed PI control [Hz]	【PI gain auto calc】
com_f4_asr_zeta	float	-	-	Damping factor for speed PI control	
com_f4_asr_kp	float	-	-	Proportional gain for speed PI control [V s/rad]	【PI gain manual input】
com_f4_asr_kidt	float	-	-	Integral gain for speed PI control [V s/rad]	
com_s2_enable_write	int16_t	Q0	-	Variable to allow to input RMW structure	
g_s2_enable_write	int16_t	Q0	-	Variable to allow to input RMW structure	
st_ics_input	mtr_ctrl_input_t	Q0		Structure for RMW input	structure

Table 3-28 List of variables “r_mtr_parameter.h / Structure : st_mtr_parameter_t”

variable	type	Qn	PU	content	remarks
u2_mtr_pp	uint16_t	Q0	-	Number of pole pairs	
s2_mtr_r	int16_t	Q17	[1/Ω]	Resistance [PU]	
s2_mtr_ld	int16_t	Q19	[1/H]	D-axis inductance [PU]	
s2_mtr_lq	int16_t	Q19	[1/H]	Q-axis inductance [PU]	
s2_mtr_m	int16_t	Q16	[rad/V s]	Induced voltage constant [PU]	
s2_mtr_j	int16_t	Q12	[1/kg m^2]	Rotor inertia [PU]	

Table 3-29 List of variables “r_mtr_parameter.h / Structure : st_mtr_design_parameter_t”

variable	type	Qn	PU	content	remarks
u2_pp	uint16_t	Q0	-	Number of pole pairs	【PI gain auto calc】
f4_r	float	-	-	Resistance [Ω]	
f4 _ld	float	-	-	D-axis inductance[H]	
f4 _lq	float	-	-	Q-axis inductance[H]	
f4 _m	float	-	-	Induced voltage constant [Vs/rad]	
f4 _j	float	-	-	Rotor inertia[kgm^2]	
f4_speed_omega	float	-	-	Frequency for speed PI control [Hz]	
f4_asr_zeta	float	-	-	Damping factor for speed PI control	
f4_dt	float	-	-	Period of speed control [s]	
f4_pu_sf_asr_kp	float	-	-	Scale factor to convert from [Vs/rad] to PU	
f4_pu_sf_asr_kidt	float	-	-	Scale factor to convert from [Vs/rad] to PU	
u1_q_voltage	uint8_t	Q0	-	Scale factor to convert from [V] to PU	
u1_q_afreq	uint8_t	Q0	-	Scale factor to convert from [rad/s] to PU	
u1_q_asr_kp	uint8_t	Q0	-	Q-format of proportional gain	
u1_q_asr_kidt	uint8_t	Q0	-	Q-format of integral gain	

Table 3-30 List of variables “r_mtr_parameter.h / Structure : st_mtr_ctrl_gain_t”

variable	type	Qn	PU	content	remarks
s2_kp	int16_t	Q14	[rad/ V s]	Proportional gain for speed PI control [PU]	
s2_kidt	int16_t	Q15	[rad/ V s]	Integral gain for speed PI control [PU]	
s2_speed_dz_intg	int16_t	Q14	-	Dead zone for integral term	

Table 3-31 List of variables “r_mtr_driver_access.h / Structure : mtr_ctrl_input_t”

variable	type	Qn	PU	content	remarks
u1_dir	uint8_t	Q0	-	Direction of rotation	
s2_ref_speed_rad	int16_t	Q13	[s/rad]	Command rotational speed [PU]	electric angle
s2_ol2cl_speed_rad	int16_t	Q13	[s/rad]	Speed allowed to transition to PI control [PU]	electric angle
s2_cl2ol_speed_rad	int16_t	Q13	[s/rad]	Speed to transition to open-loop drive [PU]	electric angle
s2_draw_in_ref_v	int16_t	Q14	[1/V]	Command voltage at draw-in [PU]	
s2_ol_ref_v	int16_t	Q14	[1/V]	Command voltage at open-loop drive [PU]	
s2_ramp_limit_speed_rad	int16_t	Q13	[s/rad]	Limit of acceleration [PU]	electric angle
s2_ol2cl_ramp_speed_rad	int16_t	Q13	[s/rad]	Acceleration at transition to PI control [PU]	electric angle
s2_ramp_limit_v	int16_t	Q14	[1/V]	Limit of variation of voltage [PU]	
s2_angle_shift_adjust	int16_t	Q0	-	adjust delay counts	
u1_detect_angle_current	uint8_t	Q0	-	Threshold current of angle detection for salient rotor	【IPD】
u4_detect_angle_thsld	uint32_t	Q0	-	TRX count value differential of angle detection for salient rotor	
u1_detect_angle_max_cnt	uint8_t	Q0	-	Maximum number of times salient rotor angle detection is measured	
u1_detect_polarity_current	uint8_t	Q0	-	Threshold current of polarity detection for salient rotor	
u4_detect_polarity_thsld	uint32_t	Q0	-	TRX count value differential of polarity detection for salient rotor	
u1_detect_polarity_max_cnt	uint8_t	Q0	-	Maximum number of times salient rotor polarity detection is measured	
u1_non_salient_current	uint8_t	Q0	-	Threshold current of angle detection for non-salient rotor	
u4_non_salient_thsld	uint32_t	Q0	-	TRX count value differential of angle detection for non-salient rotor	
u1_non_salient_max_cnt	uint8_t	Q0	-	Maximum number of times non-salient rotor angle detection is measured	
u2_current_offset_cnt	uint16_t	Q0	-	Number of measuring current offset	【IPD】【CMP overcurrent】
s2_zc_ad2cmp_speed_rpm	int16_t	Q13	[s/rad]	Speed of conversion of zero crossing detection system from A/D converter to CMP1 [rpm]	【CMP zero-crossing】
s2_zc_cmp2ad_speed_rpm	int16_t	Q13	[s/rad]	Speed of conversion of zero crossing detection system from CMP1 converter to A/D converter [rpm]	
u2_zc_change_judge_cnt	uint16_t	Q0	-	Judgement count for converting zero crossing detection methods	
st_motor	st_mtr_parameter_t	-	-	structure for motor parameter	structure
st_gain	st_mtr_ctrl_gain_t	-	-	structure for PI control	
st_ctrl_params	st_mtr_design_parameter_h	-	-	Structure for PI gain calculation	

Table 3-32 List of variables “r_mtr_driver_access.c”

variable	type	Qn	PU	content	remarks
st_ics_input_buff	mtr_ctrl_input_t	-	-	Buffer for RMW input	structure
g_u1_trig_enable_write	uint8_t	Q0	-	Flag to allow to input RMW values	

Table 3-33 List of variables “r_mtr_statemachine.h / Structure : st_mtr_statemachine_t”

variable	type	Qn	PU	content	remarks
u1_status	uint8_t	Q0	-	Motor status	
u1_status_next	uint8_t	Q0	-	Next motor status	
u1_current_even	uint8_t	Q0	-	execution event	

Table 3-34 List of variables “r_mtr_statemachine.c”

variable	type	Qn	PU	content	remarks
state_transition_table [MTR_SIZE_EVENT] [MTR_SIZE_STATE]	static uint8_t	Q0	-	Macro array for state transition	
action_table [MTR_SIZE_EVENT] [MTR_SIZE_STATE]	static mtr_action_t	Q0	-	Function array for state transition	

Table 3-35 List of variables "r_mtr_120.h / Structure : st_mtr_pi_control_t"

variable	type	Qn	PU	content	remarks
s2_err	int16_t	Q13	[1/V]	Error between reference and estimated rotational speed [PU]	
s2_pre_err	int16_t	Q13	[1/V]	Previous Error between reference and estimated rotational speed [PU]	
s2_kp	int16_t	Q14	[rad/V s]	Proportional gain for speed PI control [PU]	
s2_kidt	int16_t	Q17	[rad/V s]	Integral gain for speed PI control [PU]	
s2_speed_dz_intg	int16_t	Q14	-	Dead zone for integral term	
s2_pre_speed_rad	int16_t	Q13	[rad/V s]	Previous rotational speed [PU]	【PI gain auto calc】

Table 3-36 List of variables "r_mtr_120.h / Structure : st_mtr_ol2cl_t"

variable	type	Qn	PU	content	remarks
u2_flag_change	uint16_t	Q0	-	Flag to allow to transition to PI control	
u2_flag_change_speed	uint16_t	Q0	-	Flag for exceeding change speed	
u2_zc_flag	uint16_t	Q0	-	Flag for zero-crossing detection avoiding commutation	
u2_rotor_pos	uint16_t	Q0	-	Phase of rotor position	
u2_zc_cnt	uint16_t	Q0	-	Counter for zero-crossing detection	
s2_change_speed_rad	int16_t	Q13	[s/rad]	Speed allowed to transition to PI control [PU]	
s2_ref_speed_rad_buf	int16_t	Q13	[s/rad]	Buff for reference rotational speed [PU]	
s2_ramp_speed_rad	int16_t	Q13	[s/rad]	Acceleration at transition to PI control [PU]	

Table 3-37 List of variables “r_mtr_120.h / Structure : st_mtr_ipd_t”

variable	type	Qn	PU	content	remarks
u1_state_ipd	uint8_t	Q0	-	State of initial position detection	【IPD】
u1_judge_salient	uint8_t	Q0	-	Result of salient jdgement	
u1_flag_cmp0_intr	uint8_t	Q0	-	Flag for CMP0 interrupt	
u1_detect_angle_current	uint8_t	Q0	-	Threshold current of angle detection for salient rotor	
u1_detect_angle_cnt	uint8_t	Q0	-	Number of times salient rotor angle detection is measured	
u1_detect_angle_max_cnt	uint8_t	Q0	-	Maximum number of times salient rotor angle detection is measured	
u1_detect_polarity_current	uint8_t	Q0	-	Threshold current of polarity detection of angle detection for salient rotor	
u1_detect_polarity_cnt	uint8_t	Q0	-	Number of times salient rotor polarity detection is measured	
u1_detect_polarity_max_cnt	uint8_t	Q0	-	Maximum number of times salient rotor polarity detection is measured	
u1_detect_polarity	uint8_t	Q0	-	Result of polarity detection for salient rotor	
u1_non_salient_current	uint8_t	Q0	-	Threshold current of angle detection for non-salient rotor	
u1_nonsalient_cnt	uint8_t	Q0	-	Number of times non-salient rotor angle detection is measured	
u1_non_salient_max_cnt	uint8_t	Q0	-	Maximum number of times non-salient rotor angle detection is measured	
u1_initial_position	uint8_t	Q0	-	Initial position detection judgment result	
u2_temp_trx_cnt	uint16_t	Q0	-	TRX count value acquisition	
u4_sum_trx_detect_angle[3]	uint32_t	Q0	-	TRX count value of angle detection for salient rotor	
u4_sum_trx_salient_check	uint32_t	Q0	-	TRX count value during polarity detection	
u4_sum_trx_detect_polarity[2]	uint32_t	Q0	-	TRX count value of polarity detection for salient rotor	
u4_sum_trx_non_salient[6];	uint32_t	Q0	-	TRX count value of angle detection for non-salient rotor	
u4_trx_diff_detect_angle	uint32_t	Q0	-	TRX count value differential of angle detection for salient rotor	
u4_trx_diff_detect_polarity	uint32_t	Q0	-	TRX count value differential of polarity detection for salient rotor	
u4_trx_diff_non_salient	uint32_t	Q0	-	TRX count value differential in angle detection for non-salient rotor	
u4_detect_angle_thsld	uint32_t	Q0	-	TRX count value differential threshold of angle detection for salient rotor	
u4_detect_angle_percentage	uint32_t	Q0	-	Percentage of TRX count value differential threshold of angle detection for salient rotor	
u4_detect_polarity_thsld	uint32_t	Q0	-	TRX count value differential threshold of polarity detection for salient rotor	
u4_detect_polarity_percentage	uint32_t	Q0	-	Percentage of TRX count value differential threshold of polarity detection for salient rotor	
u4_non_salient_thsld	uint32_t	Q0	-	TRX count value differential threshold of angle detection for non-salient rotor	
u4_non_salient_percentage	uint32_t	Q0	-	Percentage of TRX count value differential threshold of angle detection for salient rotor	

Table 3-38 List of variables “r_mtr_120.h / Structure : st_mtr_cmp_zc_t”

variable	type	Qn	PU	content	remarks
u1_carrier_skip	uint8_t	Q0	-	Counter for CMP1 interrupt permit delay	【CMP zero-crossing】
u1_flag_change_zc_detection	uint8_t	Q0	-	Zero-crossing detection method conversion flag	
u1_cmp_signal_num	uint8_t	Q0	-	Counter for CMP1 judgment delay	
u2_zc_change_judge_cnt	uint16_t	Q0	-	Judgement count for converting zero crossing detection methods	
u2_zc_change_cnt	uint16_t	Q0	-	Counter when converting zero crossing detection methods	
u2_cmp_window_width	uint16_t	Q0	-	Timer window width	
s2_zc_ad2cmp_speed_rad	int16_t	Q13	[s/rad]	Speed of conversion of zero crossing detection system from A/D converter to CMP1	
s2_zc_cmp2ad_speed_rad	int16_t	Q13	[s/rad]	Speed of conversion of zero crossing detection system from CMP1 to A/D converter	

Table 3-39 List of variables “r_mtr_120.h / Structure : st_mtr_120_control_t” [1/2]

variable	type	Qn	PU	content	remarks
u1_ref_dir	uint8_t	Q0	-	Reference rotational direction	
u1_dir	uint8_t	Q0	-	Rotational direction	
u1_flag_charge_cap	uint8_t	Q0	-	Flag for charge capacitor	
u1_state_draw_in	uint8_t	Q0	-	Draw-in state management	
u1_flag_pattern_change	uint8_t	Q0	-	Flag for zero-crossing detection	
u1_flag_pattern_error	uint8_t	Q0	-	Flag for voltage pattern error	
u1_flag_finish_ipd	uint8_t	Q0	-	Flag for finishing initial position detection	
u1_state_control_mode	uint8_t	Q0	-	State management for control mode	
u1_state_driving	uint8_t	Q0	-	State management for driving speed	
u1_first_rotation_cnt	uint8_t	Q0	-	Count first 6 pattern changes for speed calculation	
u1_bemf_signal	uint8_t t	Q0	-	Estimated Hall pattern	
u1_pre_bemf_signal	uint8_t	Q0	-	Previous estimated Hall pattern	
u1_v_pattern_ring_buff	uint8_t	Q0	-	Ring buffer for timer counter for speed calculation	
u1_zc_detect_method	uint8_t	Q0	-	Method of zero-crossing detection	
u2_run_mode	uint16_t	Q0	-	Run mode	
u2_state_speed_ref	uint16_t	Q0	-	Speed reference mode	
u2_state_voltage_ref	uint16_t	Q0	-	Voltage reference mode	
u2_speed_timer_cnt	uint16_t	Q0	-	Value of timer counter	
u2_pre_speed_timer_cnt	uint16_t	Q0	-	Previous value of timer counter	
u2_timer_cnt_buff[6]	uint16_t	Q0	-	Timer counts for 60 degree	
u2_timer_cnt_sum	uint16_t	Q0	-	Total counter value for speed calculation	
u2_v_pattern	uint16_t	Q0	-	Voltage pattern	
u2_pwm_duty	uint16_t	Q0	-	PWM duty	
u2_cnt_timeout	uint16_t	Q0	-	Counter for timeout error	
u2_bemf_delay	uint16_t	Q0	-	Delay counts	
u2_cnt_ol_speed	uint16_t	Q0	-	Counter for patten change at open-loop drive	
u2_ol_pattern_period	uint16_t	Q0	-	Period for pattern change at open-loop drive	
u2_cnt_draw_in	uint16_t	Q0	-	Counter for pattern change at draw-in	
u2_v_const_period	uint16_t	Q0	-	Period for pattern change at draw-in	
u2_ol_v_pattern	uint16_t	Q0	-	Voltage pattern at open-loop drive	
u2_ol_v_pattern_num	Uint1_t	Q0	-	Ring buffer for voltage pattern at open-loop drive	
u2_cnt_carrier	uint16_t	Q0	-	Counter every carrier interruption	
u2_pre_cnt_carrier	uint16_t	Q0	-	Previous value of carrier counter	
s2_speed_rad	int16_t	Q13	[s/rad]	Rotational speed [PU]	
s2_ref_speed_rad	int16_t	Q13	[s/rad]	Command rotational speed [PU]	
s2_ref_speed_rad_ctrl	int16_t	Q13	[s/rad]	Command rotational speed for control [PU]	
s2_ramp_limit_speed_rad	int16_t	Q13	[s/rad]	Limit of acceleration [PU]	

Table 3-40 List of variables “r_mtr_120.h / Structure : st_mtr_120_control_t” [2/2]

variable	type	Qn	PU	content	remarks
s2_vdc_ad	int16_t	Q14	[1/V]	Line voltage [PU]	
s2_ref_v	int16_t	Q14	[1/V]	Target voltage [PU]	
s2_ref_v_ctrl	int16_t	Q14	[1/V]	Reference voltage [PU]	
s2_ramp_limit_v	int16_t	Q14	[1/V]	Voltage change range limit value [PU]	
s2_cl2ol_speed_rad	int16_t	Q13	[s/rad]	Transition speed from sensorless control to open-loop [PU]	
s2_vu_ad	int16_t	Q0	-	U phase voltage	
s2_vv_ad	int16_t	Q0	-	V phase voltage	
s2_vw_ad	int16_t	Q0	-	W phase voltage	
s2_vn_ad	int16_t	Q0	-	Virtual neutral voltage	
s2_draw_in_ref_v	int16_t	Q14	[1/V]	Draw-in command voltage [PU]	
s2_ol_ref_v	int16_t	Q14	[1/V]	Startup voltage [PU]	
s2_angle_shift_adjust	int16_t	Q0	-	Virtual hall pattern delay adjustment value	
u4_error_status	uint32_t	Q0	-	Error status	
u4_sum_current_offset	uint32_t	Q0	-	Sum of measurement of U phase current offset	
u1_use_cmp0	uint8_t	Q0	-	Selection of function of CMP0	
u1_state_current_offset	uint8_t	Q0	-	State management for measurement of U phase current offset	
u2_current_offset	uint16_t	Q0	-	U phase current offset	
u2_current_offset_cnt	uint16_t	Q0	-	Number of measurements of current offset	
st_ol2cl	st_mtr_ol2cl_t	-	-	Sensorless switching process structure	
st_ipd	st_mtr_ipd_t	-	-	Structure for initial position detection	
st_cmp	st_mtr_cmp_zc_t	-	-	Structure for CMP1 zero crossing	
st_stm	st_mtr_statemachine_t	-	-	Structure for state machine	
st_motor	st_mtr_parameter_t	-	-	Structure for motor parameter	
st_pi_speed	st_mtr_pi_control_t	-	-	Structure for PI control	

Table 3-41 List of variables “r_mtr_120.c”

variable	type	Qn	PU	content	remarks
g_u1_cnt_ics	volatile uint8_t	Q0	-	Counter for period of RMW communication	

Table 3-42 List of variables “r_mtr_interrupt.c”

variable	type	Qn	PU	content	remarks
g_st_120	st_mtr_120_control_t	-	-	Structure for 120 conducting control	structure
g_u2_ol_v_pattern_table [2][7]	uint16_t	Q0	-	Array for voltage pattern	
g_u2_chopping_pattern_table [2][6]	uint16_t	Q0	-	Array for chopping pattern	
g_u1_cnt_ics	static uint8_t	Q0	-	Decimation counter for RMW process	

3.4 Macro definitions of sensorless 120-degree conducting control software

Lists of macro definitions used in this control program are shown below.

Table 3-43 List of Macro definitions "r_mtr_config.h"

Macro	Definition value	Qn	PU	Content	Remarks
IP_MRSSK_IPD	-	-	-	Select inverter board	
MP_TG_55L_KA	-	-	-	Select motor parameters	
CP_TG_55L_KA	-	-	-	Select control parameters	
ICS_UI	0	-	-	RMW UI	Default
BOARD_UI	1	-	-	RSSK board UI	
MTRCONF_DEFAULT_UI	0/1	-	-	Select UI	BOARD UI / RMW UI
IPD	0	-	-	Initial position detection	Default
DRAW_IN	1	-	-	Draw-in	
MTRCONF_START_MODE	0/1	-	-	Selection of starting method	IPD / Draw-in
USE_CMP_ZC_DETECTION	0/1		-	Use of zero-crossing detection with CMP1	Not use / Use(default)
OC_INTP0	0	-	-	INTP0	Default
OC_PGACMP0	1	-	-	PGA+CMP0	
MTRCONF_OC_DETECTION	0/1	-	-	Selection of overcurrent detection method	INTP0 / PGA+CMP0
MANUAL_INPUT	0	-	-	Manual input	
AUTO_CALC	1	-	-	Auto calculation	Default
MTRCONF_PI_GAIN	0/1	-	-	Selection of setting PI gain	Manual input / auto calculation
NON_COMPLEMENTARY	0	-	-	Non-complementary PWM	
COMPLEMENTARY	1	-	-	Complementary PWM	Default
MTRCONF_PWM_MODE	0/1	-	-	Select non-complementary/ complementary PWM	NON_COMPLEMENTARY / COMPLEMENTARY
STAR	0	-	-	Star wiring	Default
DELTA	1	-	-	Delta wiring	
MTRCONF_MOTOR_CONNECTION	0/1	-	-	Selection of motor wiring	Star / Delta

Table 3-44 List of Macro definitions "r_mtr_motor_parameter.h"

Macro	Definition value	Qn	PU	content	Remarks
MP_POLE_PAIRS	2	-	-	Number of pole pairs	
MP_RESISTANCE	8.991693f	-	-	Resistance [Ω]	
MP_D_INDUCTANCE	0.003775972f	-	-	D-axis inductance [H]	
MP_Q_INDUCTANCE	0.004239326f	-	-	Q-axis inductance [H]	
MP_MAGNETIC_FLUX	0.02161693f	-	-	Induced voltage constant [V s/rad]	
MP_ROTOR_INERTIA	0.000002049285f	-	-	Rotor inertia [kgm^2]	
MP_NOMINAL_CURRENT_RMS	0.42f	-	-	Nominal current [A]	
MP_RATED_SPEED	2650	-	-	Nominal speed [rpm]	

Table 3-45 List of Macro definitions “r_mtr_control_parameter.h”

Macro	Definition value	Qn	PU	content	Remarks
CP_MAX_SPEED_RPM	2650	-	-	Maximum limit of command rotational speed [rpm]	Mechanical angle
CP_LIMIT_SPEED_RPM	3975	-	-	Maximum limit of estimated rotational speed [rpm]	Mechanical angle
CP_MIN_SPEED_RPM	265	-	-	Minimum limit of command rotational speed [rpm]	Mechanical angle
CP_OC_LIMIT	1.47f	-	-	Overcurrent limit [A]	
CP_OL2CL_SPEED_RPM	530	-	-	Speed allowed to transition to PI control [rpm]	Mechanical angle
CP_DRAW_IN_REF_V	7.55f	-	-	Command voltage at draw-in[V]	
CP_OL_REF_V	3.77f	-	-	Command voltage at open-loop [V]	
CP_RAMP_LIMIT_SPEED_RPM	5	-	-	Limit of acceleration [rpm/ms]	Mechanical angle
CP_OL2CL_SPEED_RAMP_RPM	1	-	-	Acceleration at transition to PI control [rpm/ms]	Mechanical angle
CP_RAMP_LIMIT_V	0.15f	-	-	Limit for variation of voltage [V]	
CP_ASR_OMETA_HZ	12			Frequency for speed PI control [Hz]	【PI gain auto calc】
CP_ASR_ZETA	1			Damping factor for speed PI control	
CP_ASR_KP	-	-	-	Proportional gain for speed PI control [V s/rad]	【PI gain manual input】
CP_ASR _KIDT	-	-	-	Integral gain for speed PI control [V s/rad]	
CP_OL2CL_SPEED_MARGIN	53	-	-	Rotational speed margin for transition of control mode [rpm]	Mechanical angle
CP_OL2CL_ZC_CNT	3	-	-	Number of zero-crossings for transition to PI control	
CP_SKIP_SPIKE_NOSE	3	-	-	Carrier counts for avoiding spike noise	
CP_ZC_DETECT_CHANGE_RPM	1855	-	-	Speed of conversion of zero crossing detection system from A/D converter to CMP1 [rpm]	【CMP zero-crossing】
CP_ZC_CHANGE_JUDGE_CNT	81	-	-	Judgement count for converting zero crossing detection methods	
CP_ZC_CHANGE_MARGIN	53	-	-	Rotational speed margin for zero-crossing detection method [rpm]	
CP_CURRENT_OFFSET_CNT	768	-	-	Number of measurements of U phase current offset	【IPD】【CMP overcurrent】【CMP zero-crossing】
CP_DETECT_ANGLE_CURRENT	0.3f	-	-	Threshold current of angle detection for salient rotor	【IPD】
CP_DETECT_ANGLE_TRX_THSLD	1626	-	-	TRX count value differential of angle detection for salient rotor	
CP_DETECT_ANGLE_DISCHARGE	612	-	-	Discharge period of angle detection for salient rotor	
CP_DETECT_POLARITY_CURRENT	0.84f	-	-	Threshold current of polarity detection for salient rotor	
CP_DETECT_POLARITY_TRX_THSLD	1984	-	-	TRX count value differential of polarity detection for salient rotor	
CP_DETECT_POLARITY_DISCHARGE	1194			Discharge period of polarity detection for salient rotor	
CP_NON_SALIENT_CURRENT	0.84f	-	-	Threshold current of angle detection for non-salient rotor	
CP_NON_SALIENT_TRX_THSLD	992	-	-	TRX count value differential of angle detection for non-salient rotor	
CP_NON_SALIENT_DISCHARGE	1194	-	-	Discharge period of angle detection for non-salient rotor	
CP_BSC_CHARGE_PERIOD	100	-	-	Period of charging bootstrap capacitor	

Table 3-46 List of Macro definitions "r_mtr_inverter_parameter.h"

Macro	Definition value	Qn	PU	content	Remarks
IP_DEADTIME	2	-	-	Deadtime[us]	
IP_CURRENT_RANGE	20	-	-	Current scaling range [A]	
IP_VDC_RANGE	111	-	-	Range of bus voltage [V]	
IP_INPUT_V	24	-	-	Input voltage [V]	
IP_CURRENT_LIMIT	2	-	-	Current limit for inverter board [A]	
IP_OVERVOLTAGE_LIMIT	28	-	-	Upper limit of voltage [V]	
IP_UNDERVOLTAGE_LIMIT	15	-	-	Lower limit of voltage [V]	
IP_SHUNT_RESISTANCE	0.05f	-	-	Shunt resistance [Ω]	
IP_AMPLIFICATION_GAIN	8	-	-	Current amplification gain	

Table 3-47 List of Macro definitions “r_mtr_scaling_parameter.h”

Macro	Definition value	Qn	PU	content	Remarks
PU_SF_VOLTAGE	74	-	-	Scaling factor to convert to voltage	
PU_BASE_CURRENT	MP_NOMINAL_CURRENT_RMS	-	-	Based current for per-unit	
PU_BASE_VOLTAGEV	IP_INPUT_V	-	-	Based voltage for per-unit	
PU_BASE_FREQ_HZ	CP_MAX_SPEED_RPM* MP_POLE_PAIRS/60	-	-	Based frequency for per-unit	
PU_BASE_ANGLE_RAD	MTR_TWOPI	-	-	Based angle for per-unit	
PU_SF_CURRENT	1.0f / PU_BASE_CURRENT_A	-	-	Scale factor to convert from [A] to PU	
PU_SF_VOLTAGE	1.0f / PU_BASE_VOLTAGE_V	-	-	Scale factor to convert from [V] to PU	
PU_SF_FREQ	1.0f / PU_BASE_FREQ_HZ	-	-	Scale factor to convert from [Hz] to PU	
PU_SF_AFREQ	PU_SF_FREQ/PU_BASE_ANGLE_RAD	-	-	Scale factor to convert from [rad/s] to PU	
PU_SF_TIME	1.0f / PU_SF_FREQ_HZ	-	-	Scale factor to convert from [s] to PU	
PU_SF_RES	PU_SF_VOLTAGE / PU_SF_CURRENT	-	-	Scale factor to convert from [Ω] to PU	
PU_SF_IND	PU_SF_RES / PU_SF_AFREQ	-	-	Scale factor to convert from [H] to PU	
PU_SF_FLUX	PU_SF_VOLTAGE / PU_SF_AFREQ	-	-	Scale factor to convert from [Vs/rad] to PU	
PU_SF_TORQUE	PU_SF_FLUX * PU_SF_CURRENT	-	-	Scale factor to convert from [Nm] to PU	
PU_SF_INERTIA	PU_SF_TORQUE / (PU_SF_FREQ * PU_SF_AFREQ)	-	-	Scale factor to convert from [kg m^2] to PU	
PU_SF_RPM_RAD	1.0f / CP_MAX_SPEED_RPM	-	-	Scale factor to convert from [rpm] to PU	
PU_SF_RAD_RPM	CP_MAX_SPEED_RPM	-	-	Scale factor to convert from PU to [rpm]	
PU_SF_ASR_KP	PU_SF_VOLTAGE / PU_SF_AFREQ	-	-	Scale factor to convert from [Vs/rad] to PU	
PU_SF_ASR_KIDT	PU_SF_VOLTAGE / PU_SF_AFREQ	-	-	Scale factor to convert from [Vs/rad] to PU	
MTR_Q_CURRENT	14	-	-	Q-format of current	
MTR_Q_VOLTAGE	14	-	-	Q-format of voltage	
MTR_Q_AFREQ	13	-	-	Q-format of angular frequency	
MTR_Q_CTRL_TIME	18	-	-	Q-format of control period	
MTR_Q_RESISTANCE	17	-	-	Q-format of resistance	
MTR_Q_INDUCTANCE	19	-	-	Q-format of inductance	
MTR_Q_FLUX	16	-	-	Q-format of induced voltage constant	
MTR_Q_INERTIA	12	-	-	Q-format of inertia	
MTR_Q_ASR_KP	14	-	-	Q-format of proportional gain	
MTR_Q_ASR_KIDT	17	-	-	Q-format of integral gain	
RSFT_AFREQ_KP_2VOLTAGE	MTR_Q_SPEED_KP + MTR_Q_AFREQ – MTR_Q_VOLTAGE	-	-	Right shift, (KP * speed) to voltage	
RSFT_AFREQ_KIDT_2VOLTAGE	MTR_Q_SPEED_KIDT + MTR_Q_AFREQ – MTR_Q_VOLTAGE	-	-	Right shift, (KIDT * speed) to voltage	
RSFT_AFREQ_FLUX_2VOLTAGE	MTR_Q_FLUX + MTR_Q_AFREQ – MTR_Q_VOLTAGE	-	-	Right shift, (speed * induced voltage) to voltage	

Table 3-48 List of Macro definitions "main.h"

Macro	Definition value	Qn	PU	content	Remarks
MODE_INACTIVE	0x00	-	-	Inactive mode	
MODE_ACTIVE	0x01	-	-	Active mode	
MODE_ERROR	0x02	-	-	Error mode	
SIZE_STATE	3	-	-	Number of states	

Table 3-49 List of Macro definitions "ICS_define.h"

Macro	Definition value	Qn	PU	content	Remarks
RL78	-	-	-	CPU definition	

Table 3-50 List of Macro definitions "r_mtr_ics.h"

Macro	Definition value	Qn	PU	content	Remarks
ICS_ADDR	0xFE00	-	-	Address of RMW	
ICS_INT_LEVEL	2	-	-	RMW interrupt level setting	
ICS_NUM	0x40	-	-	Data size of RMW communication	
ICS_BRR	15	-	-	RMW bit rate register selection	
ICS_INT_MODE	0	-	-	RMW interrupt mode setting	
ICS_DECIMATION	4	-	-	decimation for RMW communication	

Table 3-51 List of Macro definitions "r_mtr_board.h"

Macro	Definition value	Qn	PU	content	Remarks
SW_CHATTERING_CNT	10	-	-	Counts for judgement to remove chattering	
VR1_OFFSET	0x1FF	-	-	Offset for VR1	

Table 3-52 List of Macro definitions "r_mtr_ctrl_rl78g1f.h"[1/3]

Macro	Definition value	Qn	PU	content	Remarks
MTR_CARRIER_FREQ	20.0f	-	-	Frequency of carrier	
MTR_TAU0_FREQ	32.0f	-	-	Frequency of TAU0	
MTR_PWM_TIMER_FREQ	64.0f	-	-	Frequency of PWM timer	
MTR_TRG_PERIOD	0.001f	-	-	Period of TRG	
MTR_WAIT_FOR_CMP0_ENABLE	50	-	-	Count of waiting for CMP0 interrupt enabled	【IPD】
MTR_WINDOW_DELAY_OFFSET	3	-	-	Delay count for timer window	
MTR_WINDOW_ADJUST	0	-	-	Adjust count for timer window	
MTR_MAX_WINDOW_WIDTH	1472			Maximum count for timer window	
MTR_OL2CL_BEMF_THRESH	3			Threshold for measuring induced voltage	
MTR_DEADTIME_SET	IP_DEADTIME*MTR_PWM_TIMER_FREQ	-	-	deadtime[ns]	
MTR_NDT_CARRIER_SET	MTR_CARRIER_SET-MTR_DEADTIME_SET	-	-	Resister counts of carrier without deadtime	【Complementary PWM】
MTR_CARRIER_SET	(MTR_PWM_TIMER_FREQ*1000/MTR_CARRIER_FREQ/2)-2 【Non-complementary PWM】 / MTR_NDT_CARRIER_SET+MTR_DEADTIME_SET 【Complementary PWM】	-	-	Carrier count of PWM	
MTR_HALF_CARRIER_SET	MTR_CARRIER_SET/2	-	-	Resister counts of half carrier	
MTR_PORT_HALL_U	P5_bit.no2	-	-	U phase Hall effect sensor input port	
MTR_PORT_HALL_V	P5_bit.no3	-	-	V phase Hall effect sensor input port	
MTR_PORT_HALL_W	P5_bit.no4	-	-	W phase Hall effect sensor input port	
MTR_PORT_UP	P1_bit.no5	-	-	U phase (positive phase) output port	
MTR_PORT_UN	P1_bit.no4	-	-	U phase (negative phase) output port	
MTR_PORT_VP	P1_bit.no3	-	-	V phase (positive phase) output port	
MTR_PORT_VN	P1_bit.no1	-	-	V phase (negative phase) output port	
MTR_PORT_WP	P1_bit.no2	-	-	W phase (positive phase) output port	
MTR_PORT_WN	P1_bit.no0	-	-	W phase (negative phase) output port	
MTR_PORT_SW1	P0_bit.no5	-	-	SW1 input port	
MTR_PORT_SW2	P0_bit.no6	-	-	SW2 input port	
MTR_PORT_LED1	P14_bit.no1	-	-	LED1 output port	
MTR_PORT_LED2	P14_bit.no0	-	-	LED2 output port	
MTR_PORT_LED3	P0_bit.no4	-	-	LED3 output port	
MTR_TAU1_CNT	TCR01	-	-	TAU1 count resister	
MTR_TRX_CNT	TRX	-	-	TRX count resister	
MTR_ADCCH_VR1	6	-	-	A/D converter channel of VR1	
MTR_ADCCH_VDC	4	-	-	A/D converter channel of bus voltage	

Table 3-53 List of Macro definitions “r_mtr_ctrl_rl78g1f.h”[2/3]

Macro	Definition value	Qn	PU	content	Remarks
MTR_ADCCH_VU	16	-	-	A/D converter channel of U phase voltage	
MTR_ADCCH_VV	0	-	-	A/D converter channel of V phase voltage	
MTR_ADCCH_VW	1	-	-	A/D converter channel of W phase voltage	
MTR_ADCCH_IU	2	-	-	A/D converter channel of U phase current	
MTR_ADCCH_IV	19	-	-	A/D converter channel of V phase current	
MTR_ADCCH_IW	3	-	-	A/D converter channel of W phase current	
MTR_OC_INTR_MASK	PMK0	-	-	INTP0 interruption mask	
MTR_DISABLE_OC_INTR	1	-	-	Disable INTP0 interruption	
MTR_ENABLE_OC_INTR	0	-	-	Enable INTP0 interruption	
MTR_TAU_TIMER_UP	0x20	-	-	Trigger for TIMER WINDOW of CMP1 Select	
MTR_TAU_TIMER_UN	0x40	-	-		
MTR_TAU_TIMER_VP	0x60	-	-		
MTR_TAU_TIMER_VN	0x80	-	-		
MTR_TAU_TIMER_WP	0xA0	-	-		
MTR_TAU_TIMER_WN	0xC0	-	-		
MTR_INPUT_SELECT_VN	0	-	-	Select Input signal of CMP1	
MTR_INPUT_SELECT_VU	1	-	-		
MTR_INPUT_SELECT_VV	2	-	-		
MTR_INPUT_SELECT_VW	3	-	-		
MTR_PATTERN_ERROR	0	-	-	Voltage pattern	
MTR_UP_PWM_VN_ON	1	-	-		
MTR_UP_PWM_WN_ON	2	-	-		
MTR_VP_PWM_UN_ON	3	-	-		
MTR_VP_PWM_WN_ON	4	-	-		
MTR_WP_PWM_UN_ON	5	-	-		
MTR_WP_PWM_VN_ON	6	-	-		
MTR_UP_ON_VN_PWM	7	-	-		
MTR_UP_ON_WN_PWM	8	-	-		
MTR_VP_ON_UN_PWM	9	-	-		
MTR_VP_ON_WN_PWM	10	-	-		
MTR_WP_ON_UN_PWM	11	-	-		
MTR_WP_ON_VN_PWM	12	-	-		
MTR_U_PWM_VN_ON	13	-	-		
MTR_U_PWM_WN_ON	14	-	-		
MTR_V_PWM_UN_ON	15	-	-		
MTR_V_PWM_WN_ON	16	-	-		
MTR_W_PWM_UN_ON	17	-	-		
MTR_W_PWM_VN_ON	18	-	-		
MTR_UP_ON_V_PWM	19	-	-		
MTR_UP_ON_W_PWM	20	-	-		
MTR_VP_ON_U_PWM	21	-	-		
MTR_VP_ON_W_PWM	22	-	-		
MTR_WP_ON_U_PWM	23	-	-		
MTR_WP_ON_V_PWM	24	-	-		

Table 3-54 List of Macro definitions "r_mtr_ctrl_rl78g1f.h"[3/3]

Macro	Definition value	Qn	PU	content	Remarks
ERROR_NONE	0x00	-	-	Default	
ERROR_CHANGE_CLK_TIMEOUT	0x01	-	-	Timeout error of change of resister for clock	
ERROR_CHARGE_CAP_TIMEOUT	0x02	-	-	Timeout error of capacitor charge	
ERROR_DRIVING	0x03	-	-	ERROR at driving mode	

Table 3-55 List of Macro definitions "r_mtr_common.h"

Macro	Definition value	Qn	PU	content	Remarks
MTR_TWOPI	2*3.14159265359f	-	-	2π	
MTR_TWOPI_60	MTR_TWOPI/60	-	-	2π/60	
MTR_CW	0	-	-	CW	
MTR_CCW	1	-	-	CCW	
MTR_ON	0	-	-	ON	
MTR_OFF	1	-	-	OFF	
MTR_CLR	0	-	-	Flag clear	
MTR_SET	1	-	-	Flag set	

Table 3-56 List of Macro definitions "r_mtr_parameter.h"

Macro	Definition value	Qn	PU	content	Remarks
MTR_SPEED_PI_LIMIT_V	IP_INPUT_V	-	-	Output voltage limit at PI control	
MTR_MAX_DRIVE_V	IP_INPUT_V * 0.90f	-	-	Maximum output voltage	
MTR_MIN_DRIVE_V	IP_INPUT_V * 0.0f	-	-	Minimum output voltage	
MTR_MCU_ON_V	IP_INPUT_V * 0.8f	-	-	MCU stable supply voltage	
MTR_SPEED_CALC_BASE	125000 * MTR_TWOPI	-	-	Calculation parameter to convert the timer counter to rotational speed	
MTR_PU_Q_SPEED_CALC_BASE	FIX32_fromfloat(MTR_SPEED_CALC_BASE * PU_SF_AFREQ, MTR_Q_AFREQ)	Q13	[s/rad]	Calculation parameter to convert the timer counter to rotational speed	
MTR_OL_SPEED_CALC_BASE	MTR_CARRIER_FREQ * 1000 * MTR_TWOPI / MTR_PATTERN_NUM	-	-	Calculation parameter to convert rotational speed to timer counter at open-loop drive	
MTR_PU_Q_OL_SPEED_CALC_BASE	FIX32_fromfloat(MTR_OL_SPEED_CALC_BASE * PU_SF_AFREQ, MTR_Q_AFREQ)	Q13	[s/rad]	Calculation parameter to convert rotational speed to timer counter at open-loop drive	
MTR_SPEED_DZ_INTG_MIN	0	-	-	Minimum value of dead zone for integral term	
MTR_REF_CURRENT_BASE	(float)IP_AMPLIFICATION_GAIN*IP_SHUNT_RESISTANCE*256/5	-	-	Scaling factor for setting threshold current of CMP0	
MTR_PU_Q_OVERVOLTAGE_LIMIT	FIX_fromfloat(IP_OVERVOLTAGE_LIMIT * PU_SF_VOLTAGE, MTR_Q_VOLTAGE)	Q14	[1/V]	Maximum limit for voltage	
MTR_PU_Q_UNDERVOLTAGE_LIMIT	FIX_fromfloat(IP_UNDERVOLTAGE_LIMIT * PU_SF_VOLTAGE, MTR_Q_VOLTAGE)	Q14	[1/V]	Minimum limit for voltage	
MTR_PU_Q_SPEED_LIMIT	FIX_fromfloat(MTR_LIMIT_SPEED_RAD * PU_SF_AFREQ, MTR_Q_AFREQ)	Q13	[s/rad]	Maximum limit for speed	
MTR_PU_Q_MCU_ON_V	FIX_fromfloat(MTR_MCU_ON_V * PU_SF_VOLTAGE, MTR_Q_VOLTAGE)	Q14	[1/V]	MCU stable voltage	
MTR_PU_Q_MAX_DRIVE_V	FIX_fromfloat(MTR_MAX_DRIVE_V * PU_SF_VOLTAGE, MTR_Q_VOLTAGE)	Q14	[1/V]	Maximum output voltage	
MTR_PU_Q_MIN_DRIVE_V	FIX_fromfloat(MTR_MIN_DRIVE_V * PU_SF_VOLTAGE, MTR_Q_VOLTAGE)	Q14	[1/V]	Minimum output voltage	
MTR_PU_Q_SPEED_CALC_BASE_1ST	MTR_PU_Q_SPEED_CALC_BASE/6	Q13	[s/rad]	Calculation parameter to convert the timer counter to rotational speed at first speed calculation	
MTR_PU_Q_SPEED_CALC_BASE_2ND	MTR_PU_Q_SPEED_CALC_BASE/3	Q13	[s/rad]	Calculation parameter to convert the timer counter to rotational speed at second speed calculation	
MTR_PU_Q_SPEED_CALC_BASE_3RD	MTR_PU_Q_SPEED_CALC_BASE/2	Q13	[s/rad]	Calculation parameter to convert the timer counter to rotational speed at third speed calculation	
MTR_PU_Q_SPEED_CALC_BASE_4TH	MTR_PU_Q_SPEED_CALC_BASE*2/3	Q13	[s/rad]	Calculation parameter to convert the timer counter to rotational speed at fourth speed calculation	
MTR_PU_Q_SPEED_CALC_BASE_5TH	MTR_PU_Q_SPEED_CALC_BASE*5/6	Q13	[s/rad]	Calculation parameter to convert the timer counter to rotational speed at fifth speed calculation	

Table 3-57 List of Macro definitions "r_mtr_statemachine.h"

Macro	Definition value	Qn	PU	content	Remarks
MTR_MODE_INIT	0x00	Q0	-	Initialization mode	
MTR_MODE_DRIVE	0x01	Q0	-	Drive mode	
MTR_MODE_STOP	0x02	Q0	-	Stop mode	
MTR_SIZE_STATE	3	Q0	-	Number of states	
MTR_EVENT_STOP	0x00	Q0	-	Stop event	
MTR_EVENT_DRIVE	0x01	Q0	-	Run event	
MTR_EVENT_ERROR	0x02	Q0	-	Error event	
MTR_EVENT_RESET	0x03	Q0	-	Reset event	
MTR_SIZE_EVENT	4	Q0	-	Number of events	

Table 3-58 List of Macro definitions "r_mtr_120.h"[1/3]

Macro	Definition value	Qn	PU	content	Remarks
MTR_TIMEOUT_CNT	(uint16_t)(1.0f/((float)MP_POLE_PAIRS*CP_MIN_SPEED_RPM/60)*1000*2)	-	-	Counts for timeout	
MTR_INIT_CNT_CARRIER	300	-	-	Initial carrier counts	
MTR_DETECT_ANGLE_MAX_CNT	20	-	-	Maximum number of times salient rotor angle detection is measured	
MTR_DETECT_ANGLE_PERCENTAGE	30	-	-	Percentage of TRX differential threshold of angle detection for salient rotor	
MTR_DETECT_POLARITY_MAX_CNT	20	-	-	Maximum number of times salient rotor polarity detection is measured	
MTR_DETECT_POLARITY_PERCENTAGE	30	-	-	Percentage of TRX differential threshold of polarity detection for salient rotor	
MTR_NON_SALIENT_MAX_CNT	20	-	-	Maximum number of times non-salient rotor angle detection is measured	
MTR_NON_SALIENT_PERCENTAGE	10	-	-	Percentage of TRX differential threshold of angle detection for non-salient rotor	
MTR_ENERGIZED_2_PHASES	0	-	-	2 phases energized	
MTR_ENERGIZED_3_PHASES	1	-	-	3 phases energized	
MTR_REVERSE_DIRECTION	3	-	-	Polarity inversion	
MTR_PRE_JUDGE	0	-	-	Pre-judgement of salient	
MTR_SALIENT	1	-	-	Salient	
MTR_NON_SALIENT	2	-	-	Non-salient	
MTR_ENERGIZE_U2V	0	-	-	Voltage pattern during 2-phase energizing	
MTR_ENERGIZE_V2W	1	-	-		
MTR_ENERGIZE_W2U	2	-	-		
MTR_ENERGIZE_V2U	3	-	-		
MTR_ENERGIZE_W2V	4	-	-		
MTR_ENERGIZE_U2W	5	-	-		
MTR_ENERGIZE_W2UV	0	-	-	Voltage pattern during 3-phase energizing	
MTR_ENERGIZE_U2VW	1	-	-		
MTR_ENERGIZE_V2WU	2	-	-		
MTR_ENERGIZE_UV2W	3	-	-		
MTR_ENERGIZE_VW2U	4	-	-		
MTR_ENERGIZE_WU2V	5	-	-		
MTR_MAX_PHASE	0	-	-	Max count phase	
MTR_MN_PHASE	1	-	-	Minimum count phase	
MTR_POLARITY_NONE	0	-	-	Polarity default value	
MTR_POLARITY_POSITIVE	1	-	-	Polarity positive direction	
MTR_POLARITY_NEGATIVE	2	-	-	Polarity negative direction	
MTR_DRAW_IN_1ST_PATTERN	1	-	--	Voltage pattern at first draw-in	
MTR_DRAW_IN_2ND_PATTERN	2	-	--	Voltage pattern at second draw-in	
MTR_PHASE_ADV	0	-	--	Advanced phase of rotor	
MTR_PHASE_DLY	1	-	--	Delayed phase of rotor	
MTR_CMP0_IPD	0	-	-	Initial position detection	
MTR_CMP0_OC	1	-	-	Overcurrent	
MTR_ZC_ADC	0	-	-	Zero-crossing with A/D converter	
MTR_ZC_CMP	1	-	-	Zero-crossing with CMP1	

Table 3-59 List of Macro definitions "r_mtr_120.h"[2/3]

Macro	Definition value	Qn	PU	content	Remarks
MTR_PATTERN_CW_V_U	2	-	-	Voltage patten at CW rotation	
MTR_PATTERN_CW_W_U	3	-	-		
MTR_PATTERN_CW_W_V	1	-	-		
MTR_PATTERN_CW_U_V	5	-	-		
MTR_PATTERN_CW_U_W	4	-	-		
MTR_PATTERN_CW_V_W	6	-	-		
MTR_PATTERN_CCW_V_U	3	-	-	Voltage pattern at CCW rotation	
MTR_PATTERN_CCW_V_W	2	-	-		
MTR_PATTERN_CCW_U_W	6	-	-		
MTR_PATTERN_CCW_U_V	4	-	-		
MTR_PATTERN_CCW_W_V	5	-	-		
MTR_PATTERN_CCW_W_U	1	-	-		
MTR_PATTERN_NUM	6	-	-	Number of voltage patterns	
MTR_ERROR_NONE	0x0000	-	-	No error	
MTR_ERROR_OVER_CURRENT	0x0001	-	-	Over current error	
MTR_ERROR_OVER_VOLTAGE	0x0002	-	-	Over voltage error	
MTR_ERROR_OVER_SPEED	0x0004	-	-	Over speed error	
MTR_ERROR_HALL_TIMEOUT	0x0008	-	-	Timeout error for Hall effect sensor control mode	
MTR_ERROR_BEMF_TIMEOUT	0x0010	-	-	Timeout error for sensorless control mode	
MTR_ERROR_HALL_PATTERN	0x0020	-	-	Hall pattern error	
MTR_ERROR_BEMF_PATTERN	0x0040	-	-	BEMF pattern error	
MTR_ERROR_UNDER_VOLTAGE	0x0080	-	-	Under voltage error	
MTR_ERROR_IPD_TRX_OVERFLOW	0x0100			TRX overflow error during initial position detection	
MTR_ERROR_UNKNOWN	0xff	-	-	Undefined error	
MTR_DRAW_IN_NONE	0	-	-	No operation	
MTR_DRAW_IN_1ST	1	-	-	First draw-in	
MTR_DRAW_IN_2ND	2	-	-	Second draw-in	
MTR_DRAW_IN_FINISH	3	-	-	Draw-in finished	

Table 3-60 List of Macro definitions “r_mtr_120.h”[3/3]

Macro	Definition value	Qn	PU	content	Remarks
MTR_SPEED_ZERO_CONST	0	-	-	Reference speed 0 mode	
MTR_SPEED_MANUAL	1	-	-	Reference speed manual input mode	
MTR_V_ZERO_CONST	0	-	-	Reference voltage 0 mode	
MTR_V_MANUAL	1	-	-	Reference voltage manual input mode	
MTR_V_PI_OUTPUT	2	-	-	Reference voltage PI output mode	
MTR_CURRENT_OFFSET_NONE	0	-	-	Default state for measurement U phase current offset	
MTR_CURRENT_OFFSET_MEASURE	1	-	-	Measuring U phase current offset	
MTR_CURRENT_OFFSET_FINISH	2	-	-	Finish measurement of U phase current offset	
MTR_IPD_NONE	0	-	-	Default state	
MTR_IPD_SALIENT_ANGLE	1	-	-	Angle detection for salient rotor	
MTR_IPD_SALIENT_POLARITY	2	-	-	Polarity detection for salient rotor	
MTR_IPD_NON_SALIENT	3	-	-	Angle detection for non-salient rotor	
MTR_IPD_FINISH	4	-	-	Finish initial position detection	
MTR_IPD_UNDETECTED	5	-	-	Initial position detection failure	
MTR_IPD_ERROR	6	-	-	Initial position detection error	
MTR_IPD	0	-	-	Initial position detection	
MTR_OPENLOOP	1	-	-	Open-loop	
MTR_SPEED_CONTROL	2	-	-	Speed control	
MTR_STOP	0	-	-	Motor stop	
MTR_LOW_SPEED	1	-	-	Low-speed driving	
MTR_MIDDLE_SPEED	2	-	-	Middle-speed driving	
MTR_HIGH_SPEED	3	-	-	High-speed driving	

3.5 Control flows (flow charts)

3.5.1 Main process

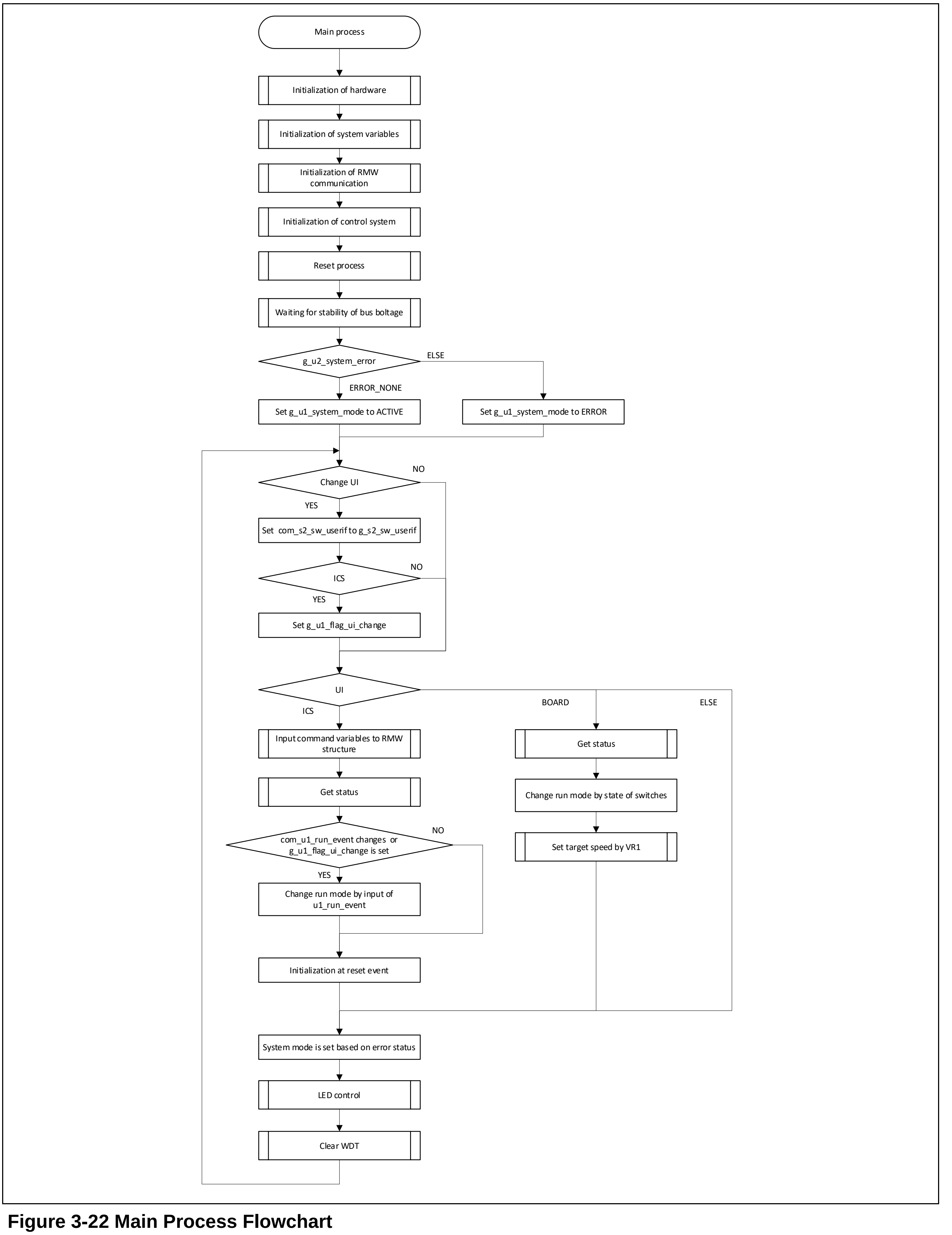
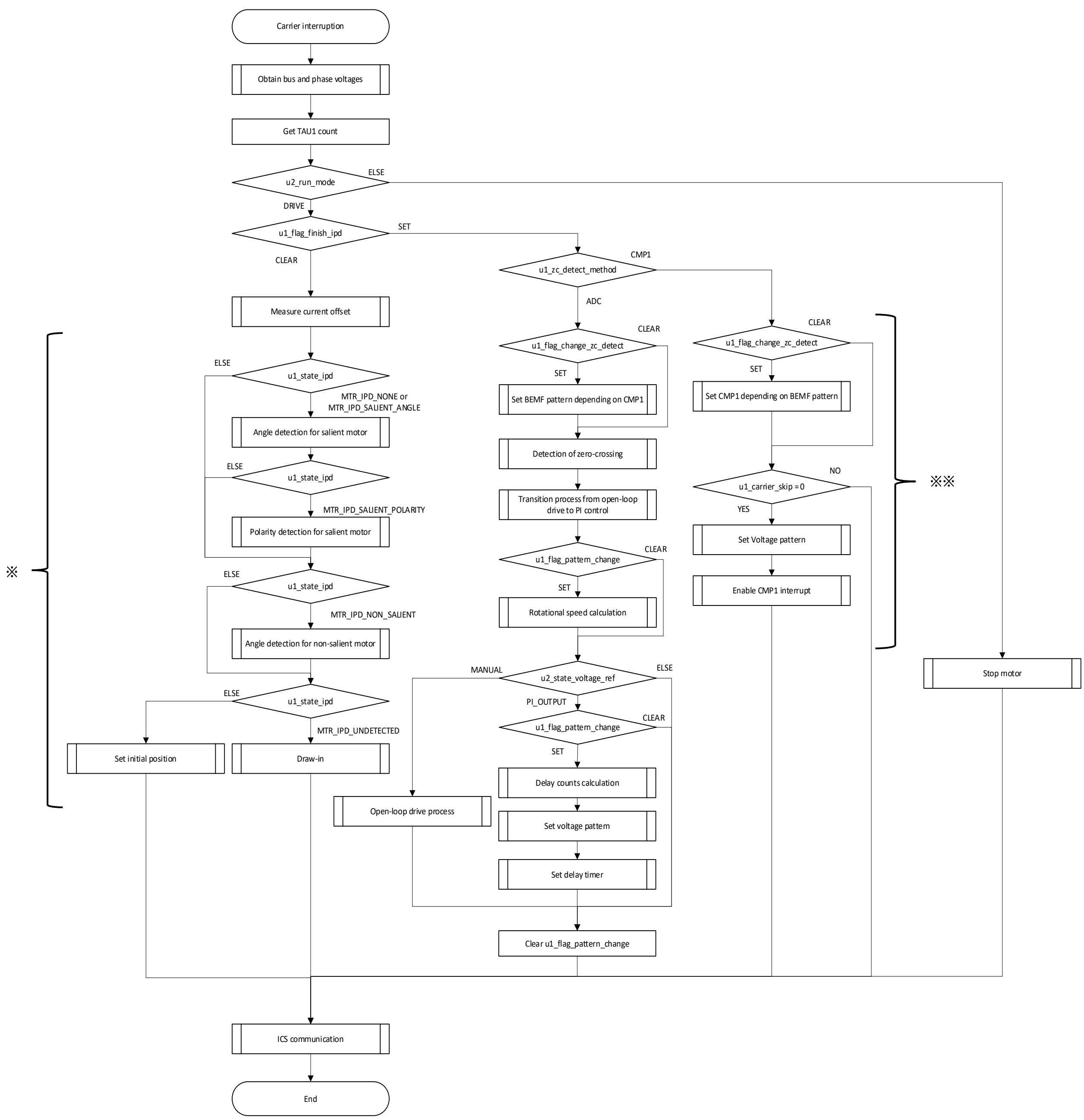


Figure 3-22 Main Process Flowchart

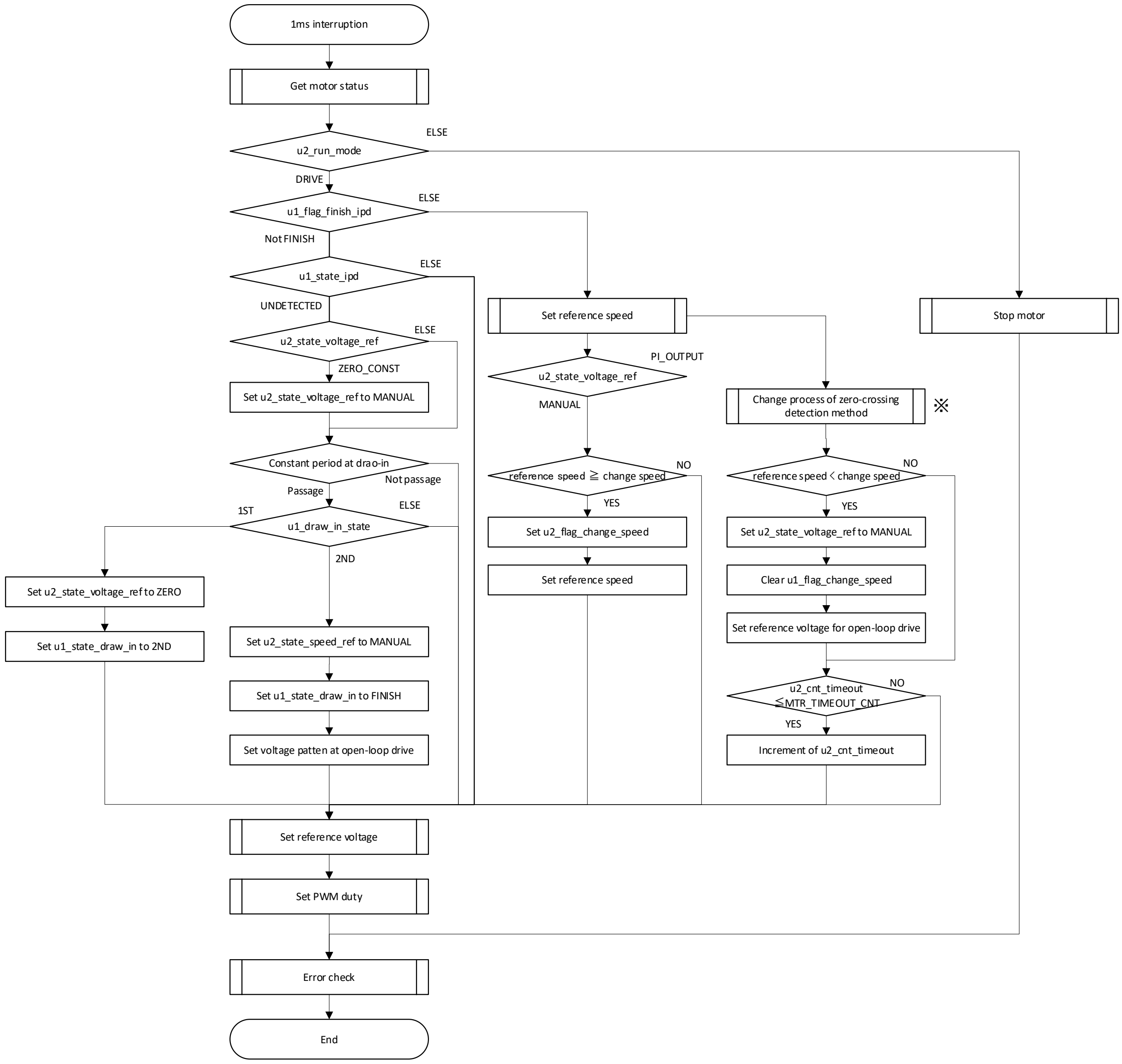
3.5.2 Carrier cycle interrupt handling



※When initial position detection is not used, motor is started up by draw-in.
※※ When zero-crossing detection with CMP1 is not used, this process is not executed.

Figure 3-23 50 [μs] Cycle Interrupt Handling

3.5.3 1 [ms] interrupt handling



※ When zero-crossing detection with CMP1 is not used, this process is not executed.

Figure 3-24 1 [ms] Interrupt Handling

3.5.4 Overcurrent interrupt handling

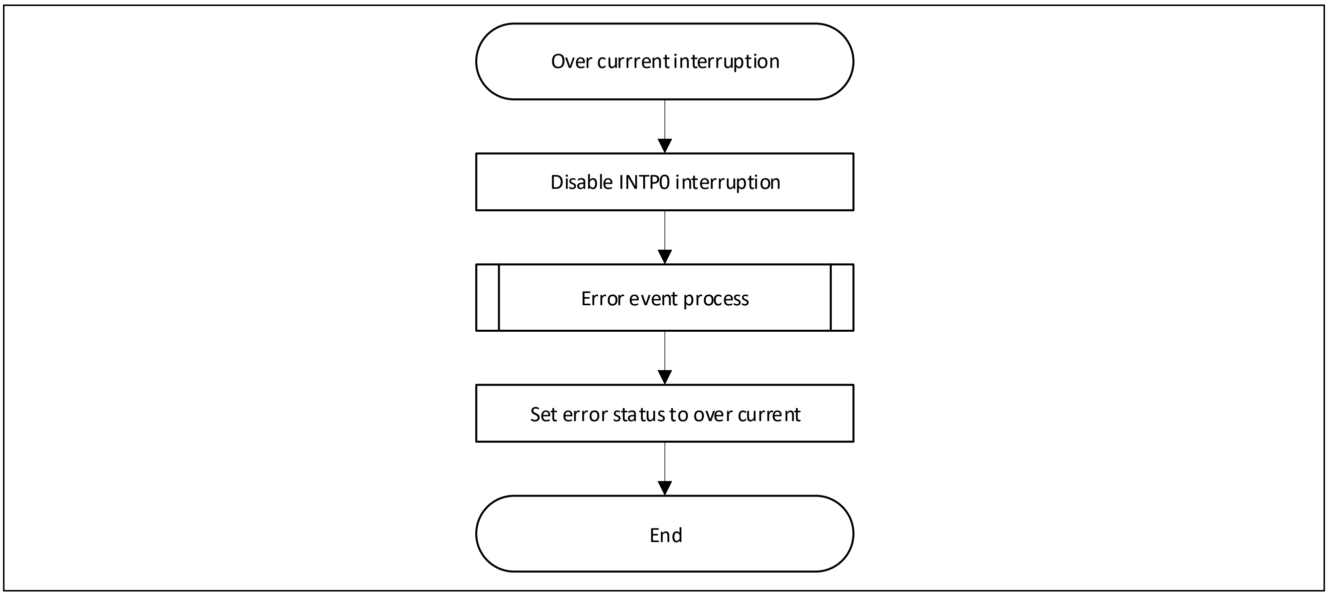
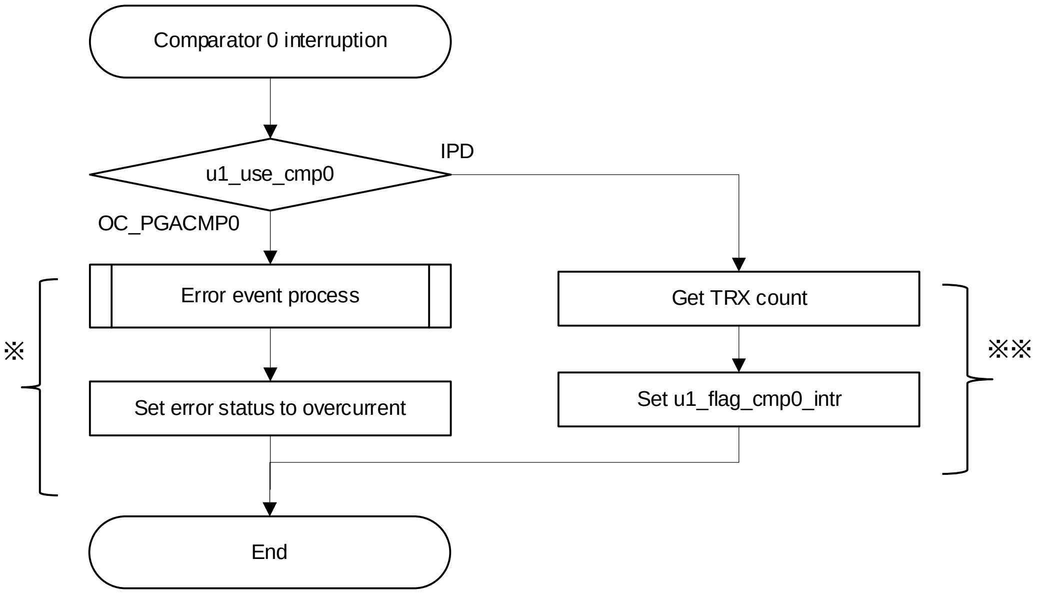


Figure 3-25 Over Current Detection Interrupt Handling

3.5.5 Comparator interrupt handling



※ When overcurrent detection with CMP0 is not used, this process is not executed.
※ When initial position detection is not used, this process is not executed.

Figure 3-26 Comparator 0 interrupt handling

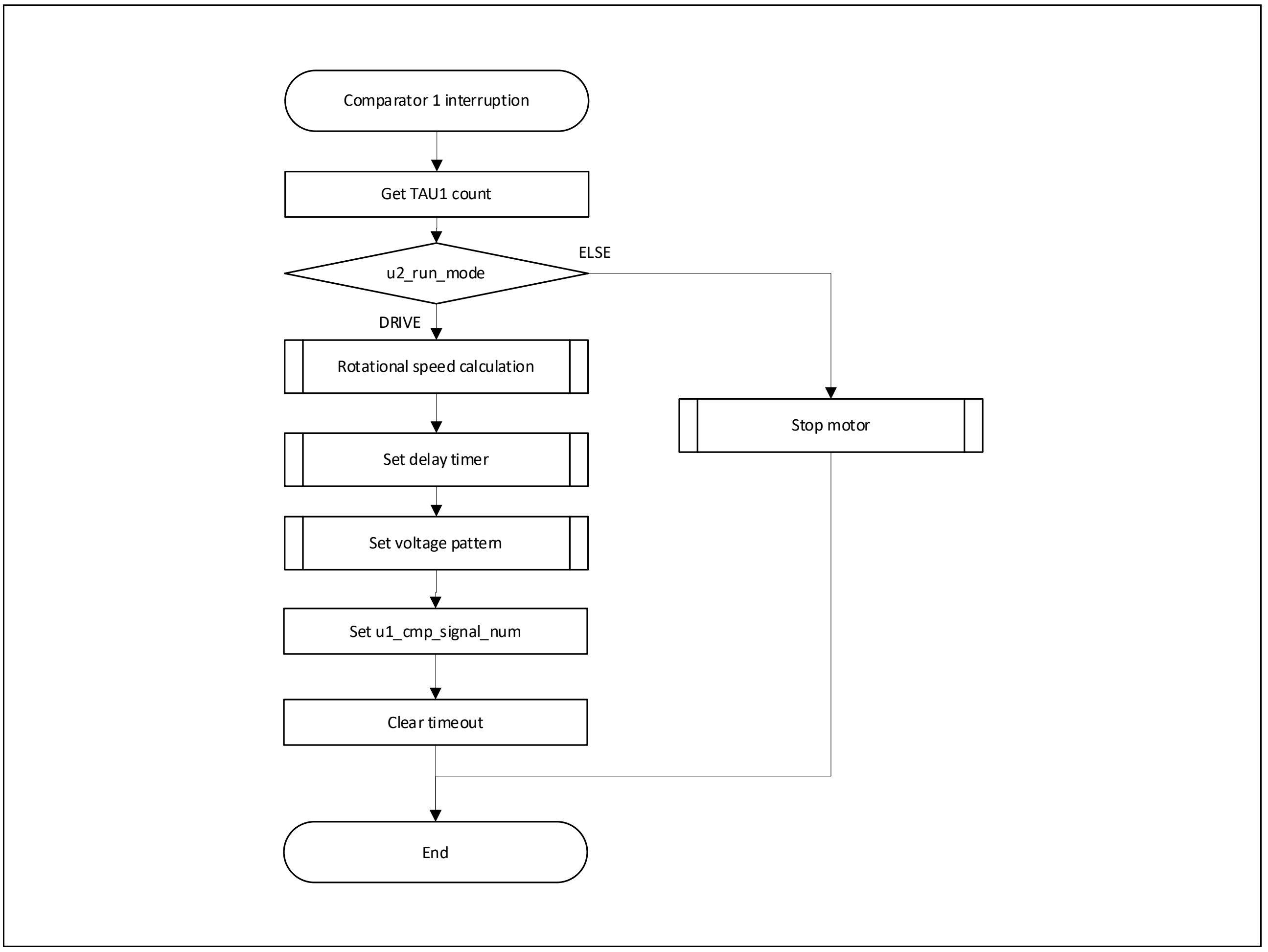


Figure 3-27 Comparator 1 interrupt handling

3.5.6 Delay timer interrupt handling

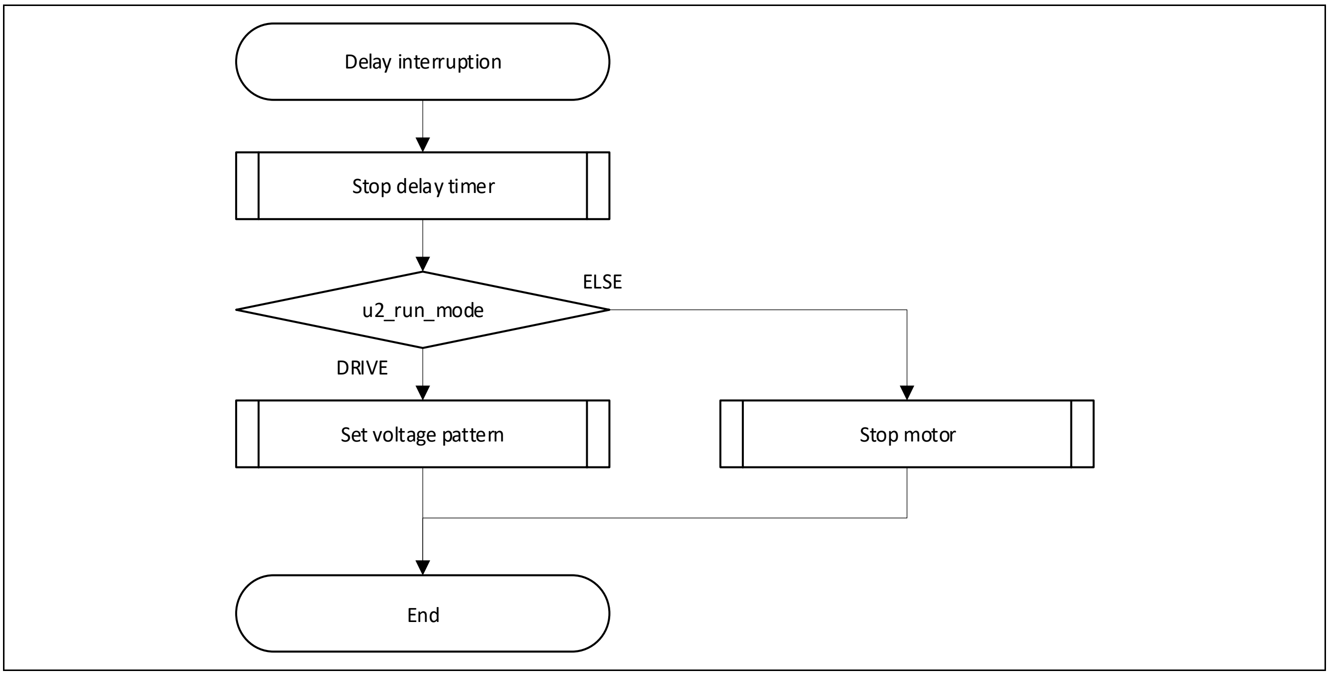


Figure 3-28 Delay Timer Interrupt Handling

4. Usage of Motor Control Development Support Tool, 'Renesas Motor Workbench'

4.1 Overview

In the target sample programs described in this application note, user interfaces (rotating/stop command, rotation speed command, etc.) based on the motor control development support tool, 'Renesas Motor Workbench' can be used. Please refer to 'Renesas Motor Workbench V 2.0 User's Manual' for usage and more details. You can find 'Renesas Motor Workbench' on Renesas Electronics Corporation website.

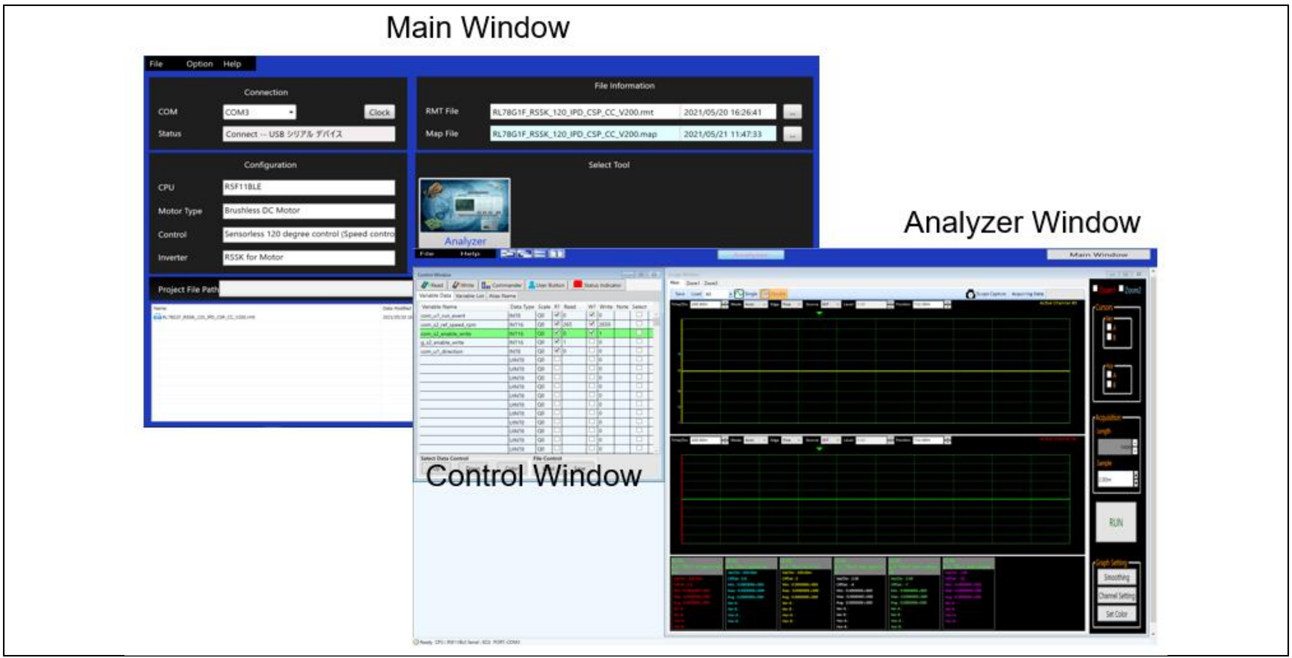
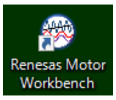


Figure 4-1 Renesas Motor Workbench– Appearance

Set up for Renesas Motor Workbench

(1) Start 'Renesas Motor Workbench' by clicking this icon. 

(2) Drop down menu [File] -> [Open RMT File(O)].
And select RMT file in '[Project Folder]/application/ics/'.
(3) Use the 'Connection' COM select menu to choose the COM port for Motor RSSK.
(4) Click the 'Analyzer' icon in right side of Main Window. (Then, "Analyzer Window" will be displayed.)
(5) Please refer to 4.3' Operation Example for Analyzer' for motor driving operation.

4.2 List of variables for Analyzer

Table 4-1 is a list of variables for Analyzer. These variable values are reflected to the protect variables when the same values as g_s2_enable_write are written to com_s2_enable_write. However, note that variables with (*) do not depend on com_s2_enable_write.

Table 4-1 List of Variables for Analyzer [1/2]

variable	type	content	remarks ([ ]: reflection variable name)
com_u1_run_event (*)	uint8_t	Input event and change run mode 0: Stop event 1: Drive event 2: Error event 3: Reset event	【g_u1_run_event】
com_s2_sw_userif (*)	int16_t	Management variable for UI 0: Analyzer use (default) 1: Board user interface use	【g_s2_sw_userif】
com_u1_direction	uint8_t	Direction of rotation 0：CW 1：CCW	【g_st_120.u1_ref_dir】
com_u2_mtr_pp	uint16_t	Number of pole pairs	【g_st_120.st_mtr.u2_mtr_pp】
com_f4_mtr_r	float	Resistance [Ω]	【g_st_120.st_mtr.s2_mtr_r】
com_f4_mtr_ld	float	D-axis inductance[H]	【g_st_120.st_mtr.s2_mtr_ld】
com_f4_mtr_lq	float	Q-axis inductance[H]	【g_st_120.st_mtr.s2_mtr_lq】
com_f4_mtr_m	float	Induced voltage constant [V s/rad]	【g_st_120.st_mtr.s2_mtr_m】
com_f4_mtr_j	float	Rotor inertia[kgm^2]	【g_st_120.st_mtr.s2_mtr_j】
com_s2_ref_speed_rpm	int16_t	Command rotational speed ［rpm］	【g_st_120.s2_ref_speed_rad】
com_s2_ol2cl_speed_rpm	int16_t	Speed allowed to transition to PI control[rpm]	【g_st_120.st_ol2cl. s2_change_speed_rad】
com_f4_draw_in_ref_v	float	Command voltage at draw-in [V]	【g_st_120.s2_draw_in_ref_v】
com_f4_ol_ref_v	float	Command voltage at open-loop drive[V]	【g_st_120.s2_ol_ref_v】
com_s2_ramp_limit_speed_rpm	int16_t	Limit of acceleration [rpm/ms]	【g_st_120.s2_ramp_limit_speed_rad】
com_s2_ol2cl_ramp_speed_rpm	int16_t	Acceleration at transition to PI control [rpm/ms]	【g_st_120.st_ol2cl. s2_ramp_speed_rad】
com_f4_ramp_limit_v	float	Limit of variation of voltage [V/ms]	【g_st_120.s2_ramp_limit_v】
com_s2_angle_shift_adjust	int16_t	Adjust delay counts	【g_st_120.s2_angle_shift_adjust】
com_f4_detect_angle_current	float	TRX count value differential of angle detection for salient rotor	【g_st_120.st_ipd. u1_detect_angle_current】
com_u4_detect_angle_thsld	uint32_t	Maximum number of times salient rotor angle detection is measured	【g_st_120.st_ipd. u4_detect_angle_thsld】
com_u1_detect_angle_max_cnt	uint8_t	Threshold current of polarity detection for salient rotor	【g_st_120.st_ipd. u4_detect_angle_max_cnt】
com_f4_detect_polarity_current	float	TRX count value differential of polarity detection for salient rotor	【g_st_120.st_ipd. u1_detect_polarity_current】
com_u4_detect_polarity_thsld	uint32_t	Maximum number of times salient rotor polarity detection is measured	【g_st_120.st_ipd. u4_detect_polarity_thsld】
com_u1_detect_polarity_max_cnt	uint8_t	Threshold current of angle detection for non-salient rotor	【g_st_120.st_ipd. u4_detect_polarity_max_cnt】
com_f4_non_salient_current	float	TRX count value differential of angle detection for non-salient rotor	【g_st_120.st_ipd. u1_non_salient_current】
com_u4_non_salient_thsld	uint32_t	Maximum number of times non-salient rotor angle detection is measured	【g_st_120.st_ipd. u4_dnon_salient_thsld】
com_u1_non_salient_max_cnt	uint8_t	Number of measuring current offset	【g_st_120.st_ipd. u4_non_salient _max_cnt】

Table 4-2 List of Variables for Analyzer [2/2]

variable	type	content	remarks ([ ]: reflection variable name)
com_u2_current_offset_cnt	uint16_t	Number of measuring current offset	【g_st_120.u2_current_offset_cnt】
com_s2_zc_detect_change_speed_rpm	int16_t	Speed of conversion of zero crossing detection system from A/D converter to CMP1 [rpm]	【g_st_120.st_cmp. s2_zc_ad2cmp_speed_rad】
com_u2_zc_change_judge_cnt	uint16_t	Judgement count for converting zero crossing detection methods	【g_st_120.st_cmp. s2_zc_change_judge_cny】
com_f4_asr_omega_hz	float	Frequency for speed PI control [Hz]	【g_st_120.st_asr.s2_kp】
com_f4_asr_zeta	float	Damping factor for speed PI control	【g_st_120.st_asr.s2_kidt】
com_f4_asr_kp	float	Proportional gain for speed PI control [V s/rad]	【g_st_120.st_asr.s2_kp】
com_f4_asr_kidt	float	Integral gain for speed PI control [V s/rad]	【g_st_120.st_asr.s2_kidt】
com_s2_enable_write	int16_t	Variable to allow to input RMW structure	【g_s2_enable_write】

4.3 Operation Example for Analyzer

An example of motor driving operation using Analyzer is shown below. For operation “Control Window” is used. Refer to ‘Renesas Motor Workbench V 2.0 User’s Manual’ for “Control Window”.

・Driving the motor

(1) The [W?] check boxes contain checkmarks for “com_u1_run_event”,“com_s2_ref_speed_rpm”, “com_s2_enable_write”
(2) Input a reference speed value in the [Write] box of “com_s2_ref_speed_rpm”.
(3) Click the “Write” button.
(4) Click the “Read” button. Confirm the [Read] box of “com_s2_ref_speed_rpm”, ”g_s2_enable_write”.
(5) Input a same value of “g_s2_enable_write” in the [Write] box of “com_s2_ref_speed_rpm”.
(6) Input a value of “1” in the [Write] box of “com_u1_run_event”.
(7) Click the “Write” button.

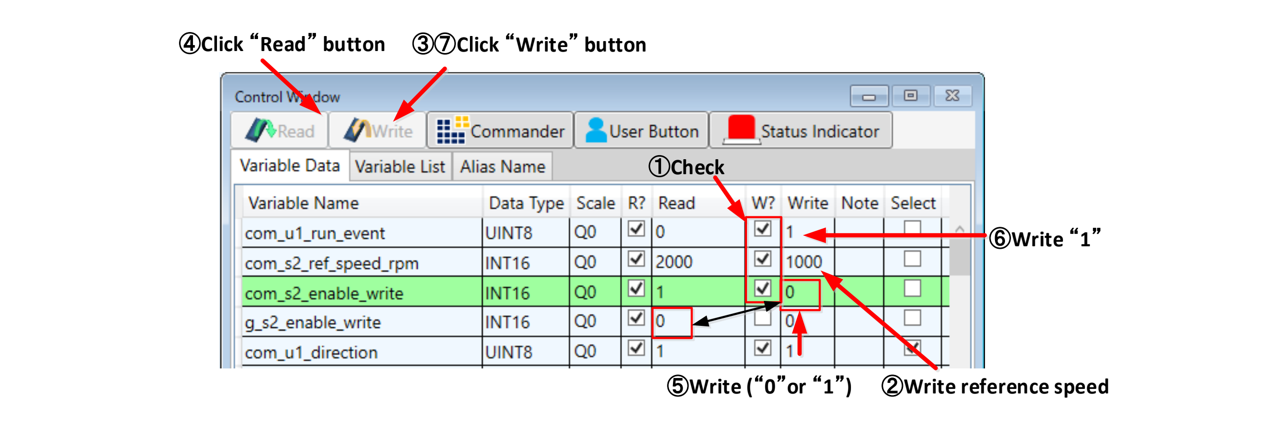


Figure 4-2 Procedure - Driving the motor

・Stop the motor

(1) Type a value of “0” in the [Write] box of “com_u1_run_event”
(2) Click the “Write” button.

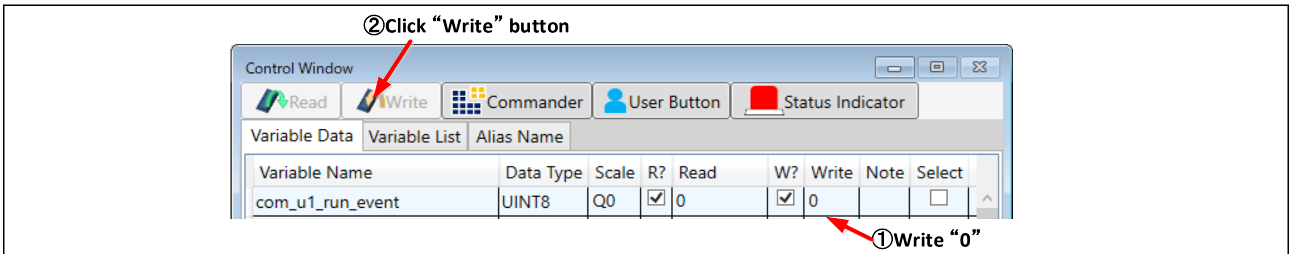


Figure 4-3 Procedure - Stop the motor

・Error cancel operation

(1) Type a value of “3” in the [Write] box of “com_u1_run_event”
(2) Click the “Write” button.

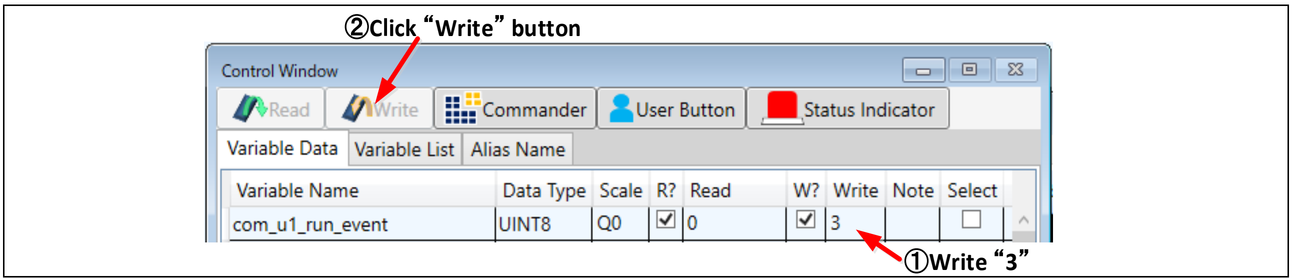


Figure 4-4 Procedure - Error cancel operation

Revision History

Rev.	Date	Description	
		Page	**Summary**
1.00	Sep.19.2018	—	First edition issued
2.00	May.26.2021	—	- Modify constitution of folders and files, change states and control parameters - Change measuring method, add process of initial position detection for non-salient motor and implement judgement of saliency for initial position detection - Transition from open-loop driving to sensorless control after detecting zero-crossing -Implement auto-calculation of PI gain

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/.