R01AN5114EJ0200
Rev.2.00
Apr.01.20

RL78/G1F

120-degree conducting control of permanent magnetic synchronous motor (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_MRSSK_120_CSP_CC_V200 (IDE: CS+ for CC)
- RL78G1F_MRSSK_120_E2S_CC_V200 (IDE: e^{2}studio)

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

The Hall effect sensor and sensorless mode can be changed by rewriting "MTRCONF_SENSOR_MODE" in the configuration definition file "r_mtr_config.h" to 0: HALL and 1: LESS, and compiling.

Reference

- RL78/G1F Group User's Manual: Hardware (R01UH0516EJ0110)
- Application note: '120-degree conducting control of permanent magnetic synchronous motor: algorithm'
(R01AN2657EJ0120)
- Renesas Motor Workbench 2.0 User's Manual (R21UZ0004EJ0202: 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
V8.03.00	CC-RL V1.08.00

e^2studio version	Build tool version
v7.7.0	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 (RTK0EM0001B00012BJ) 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.
     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 Hardware configuration

The hardware configuration is shown below.

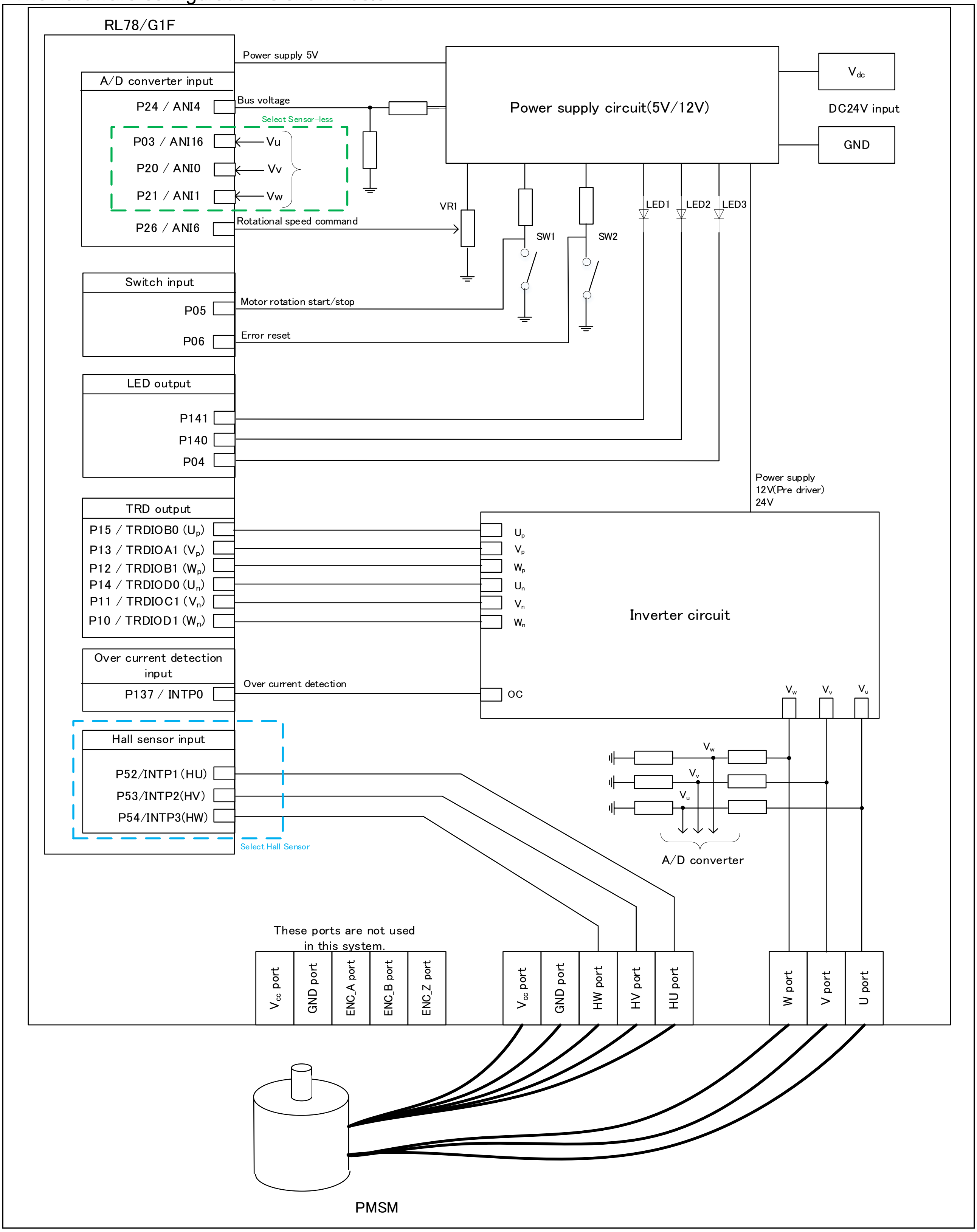


Figure 2-1 Hardware Configuration Diagram

2.2 Hardware specifications

2.2.1 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)	Motor rotation start/stop command
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	Not used in this system
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	U Phase voltage measurement(A/D)
P20 / ANI0	V Phase voltage measurement(A/D)
P21 / ANI1	W Phase voltage measurement(A/D)
P54 / INTP3	Hall effect sensor input [Note] (HU)
P53 / INTP2	Hall effect sensor input [Note] (HV)
P52 / INTP1	Hall effect sensor input [Note] (HW)
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

Note : When Hall effect sensors on motor included in 24V inverter kit are used, equip ferrite core included in this kit with cables for 3 Hall effect sensors to avoid sensor noise.

2.2.2 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
Timer Array Unit (TAU)	- 1 [ms] interval timer - Free-running timer for rotational speed measurement - Delay timer for changing conducting pattern (sensorless control mode)
Timer RD (TRD)	PWM output
External interrupt (INTP1, INTP2, INTP3)	Input signal from Hall effect sensor (Hall effect sensor control mode)
External interrupt (INTP0)	Overcurrent detection

(1) 10-bit A/D converter

The rotational speed command value input, U phase voltage (Vu), V phase voltage (Vv), W phase voltage (Vw), and inverter bus voltage (Vdc) 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: the one-shot conversion mode
- Trigger: Software trigger

(2) Timer Array Unit (TAU)

a. 1 [ms] interval timer

The channel 0 of TAU is used as 1 millisecond interval timer.

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 $\pi/6$ phase from the zero-crossing point.

(3) Timer RD (TRD)

Three-phase PWM output of upper arm chopping is performed using the Complementary PWM Mode.

(4) External interrupt (INTP1, INTP2, INTP3)

Signals from Hall effect sensors are inputted for detection of rotor position.

Both edge mode is used. When the interruption occurs, reading input signals from Hall effect sensor (detection of rotor position), changing conducting pattern and calculation of rotational speed are performed.

(5) External interrupt (INTP0)

An overcurrent is detected by an external circuit.

2.3 Software structure

2.3.1 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
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_fixed.h	Fixed point definition
		r_mtr_common.h	Common definition
		r_mtr_parameter.h	Various parameter definition
		r_mtr_pu_system.h, r_mtr_pu_system.c	Function definition for per-unit
		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_spm_120.h r_mtr_spm_120.c	Function definition for 120-degree conducting control
		r_mtr_interrupt.c	Interrupt function 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'.

2.3.2 Module configuration

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

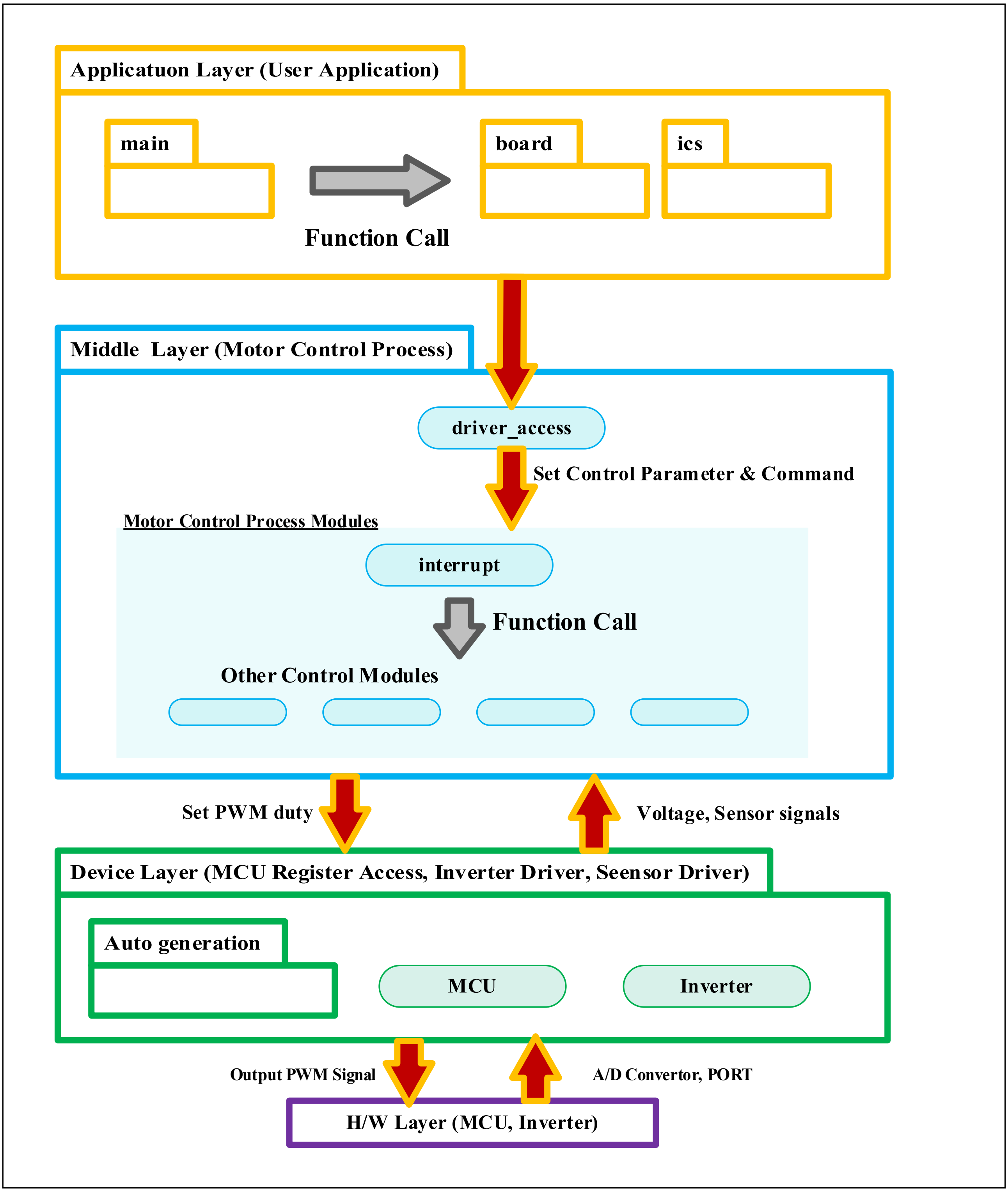


Figure 2-2 Module Configuration of the Sample Programs

2.4 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

Item	**Content**
Control method	120-degree conducting method (chopping upper arm)
Motor rotation start/stop	Determined depending on the level of SW1 (P05) (”Low”: rotation start “High”: stop) or input from Renesas Motor Workbench
Position detection of rotor magnetic pole	Hall effect sensor: Position detection based on interruption of signal from Hall effect sensors (every 60 degrees) Sensorless: Position detection based on induced voltage measured by A/D converters (every 60 degrees) - When position of rotor is detected, PWM duty and conducting pattern are set at same time.
Input voltage	DC24[V]
Main clock frequency	CPU clock: f_{CLK} 32[MHz] TRD clock: f_{HOCO} 64[MHz]
Carrier frequency (PWM)	20 [kHz]
Dead time	2 [μs]
Control cycle	Speed PI control: every 1 [ms]
Rotational speed control range	Hall effect sensor control mode: 530 [rpm] to 3200 [rpm] (Note1) Sensorless control mode: 265 [rpm] to 3200 [rpm] (Note1) Both CW and CCW are supported
Optimization	Default
Processing stop for protection	- 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 3900 rpm (monitored per 1 [ms]) 4. Hall effect sensor interruption or zero-crossing are not detected for 200 [ms]. 5. Detection of unexpected output voltage pattern 6. Detection of overcurrent by external circuit ((low-level input)

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

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 3200 [rpm] : 01FFH to 03FFH	ANI6
	CCW	-3200 [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

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

3.1.3 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 Hall effect sensor interruption in Hall effect sensor control mode or zero-crossing of induced voltage in sensorless control mode, while having the timer of performed free running.

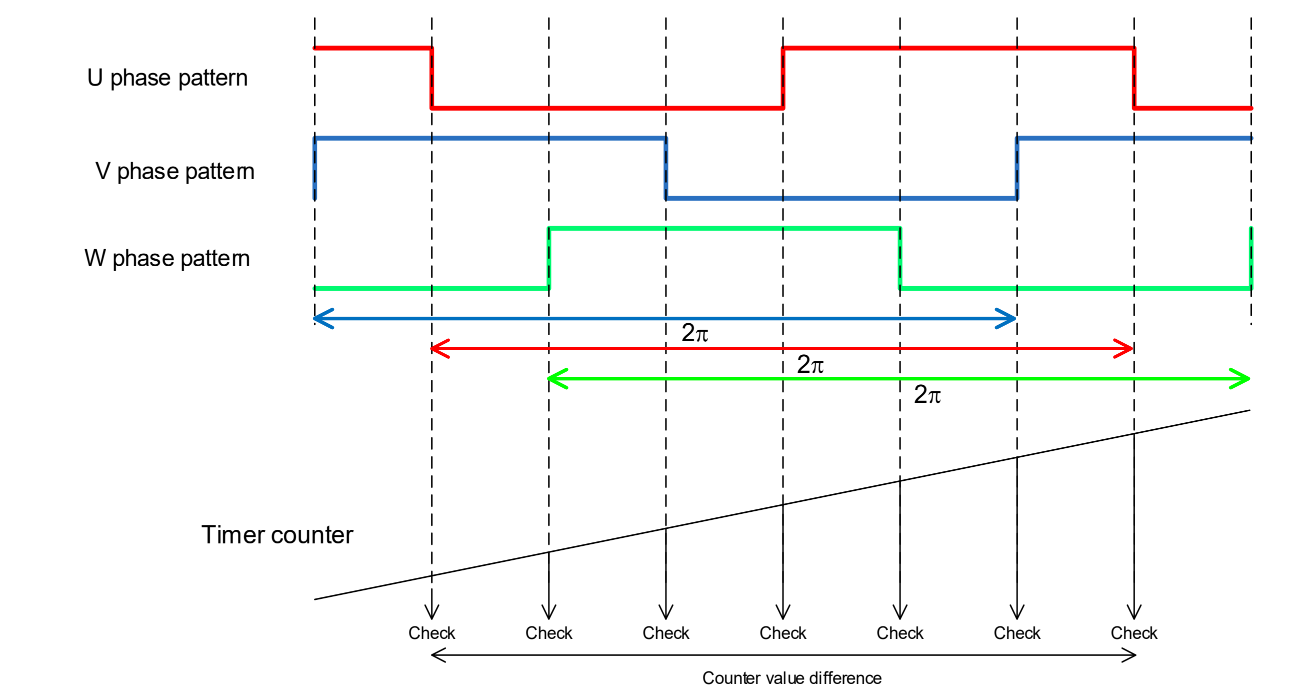


Figure 3-1 Method of Calculation for Rotational Speed

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.

3.1.4 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 3-2.

$$\text{Duty} = \frac{T_{ON}}{T_{ON} + T_{OFF}} \times 100\ [\%]$$

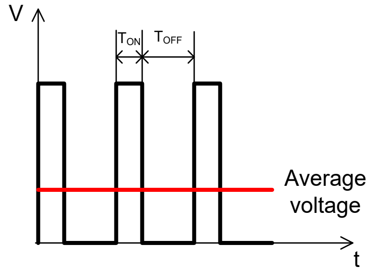


Figure 3-2 PWM Control

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 3-3 and Figure 3-4 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”.

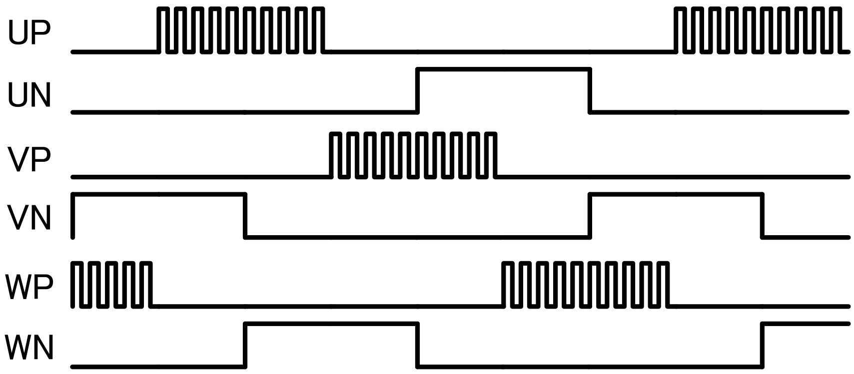


Figure 3-3 Upper Arm Chopping (Non-complementary PWM)

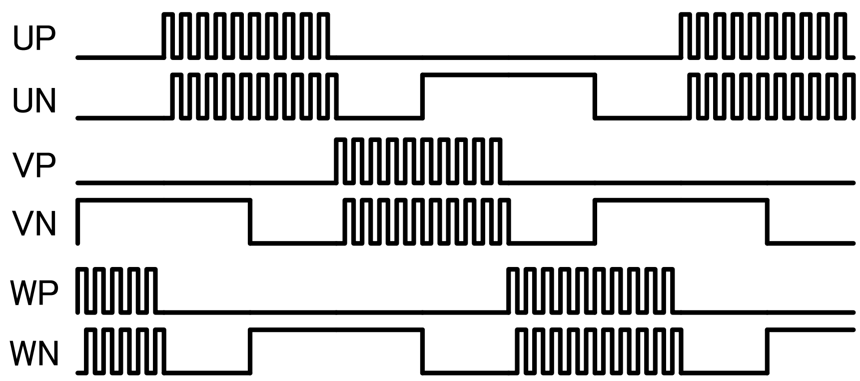


Figure 3-4 Upper Arm Chopping (Complementary PWM)

3.1.5 State transitions

Figure 3-5 shows state transition diagrams of 120-degree conducting control software.

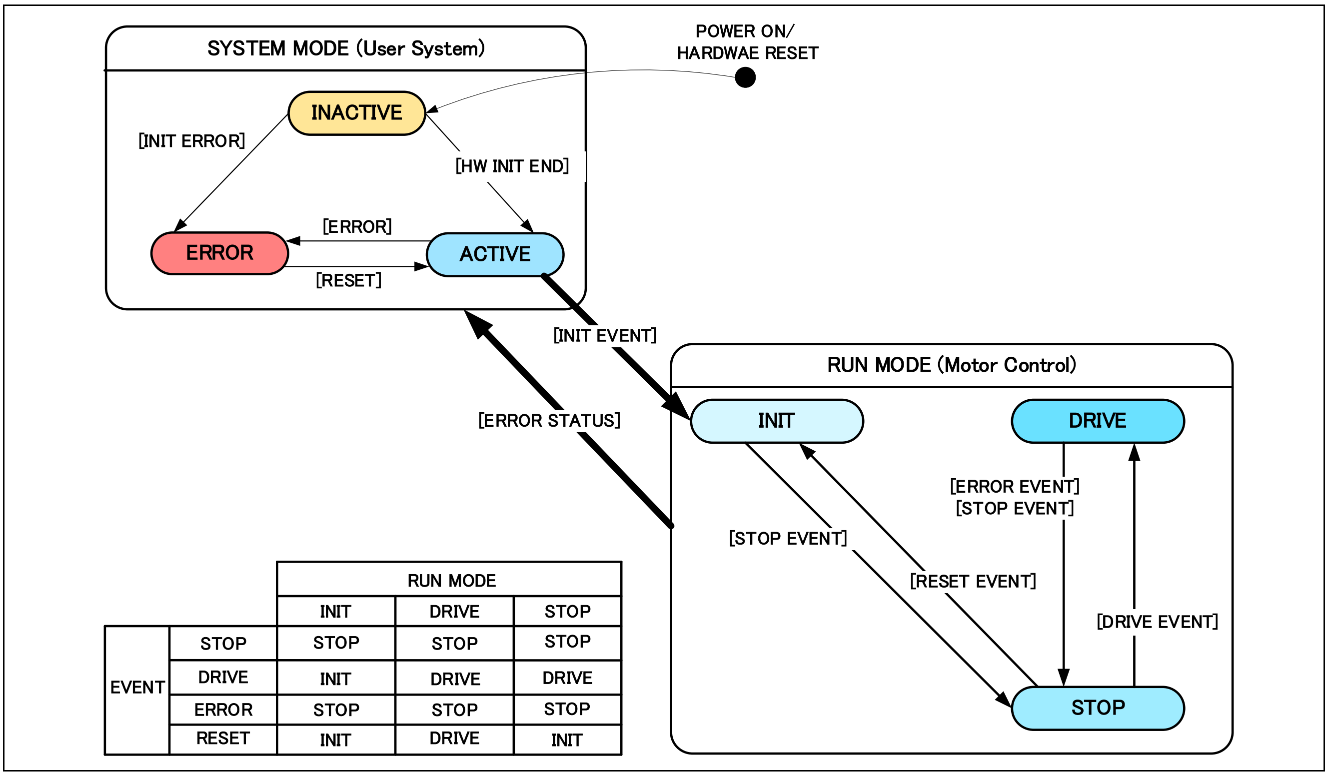


Figure 3-5 State Transition Diagram 120-degree Conducting Control Software

(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 3-5. Each "Event" is caused by occurrence as shown in Table 3-4.

Table 3-4 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

3.1.6 Start-up method

(1) Hall effect sensor control mode

In the Hall effect sensor control mode, after changing to "MTR_MODE_DRIVE", the output pattern is selected from the initial Hall effect sensor signal. Then, voltage is applied and state is changed to PI control mode. The rotational speed is calculated after second hall effect sensor interruption.

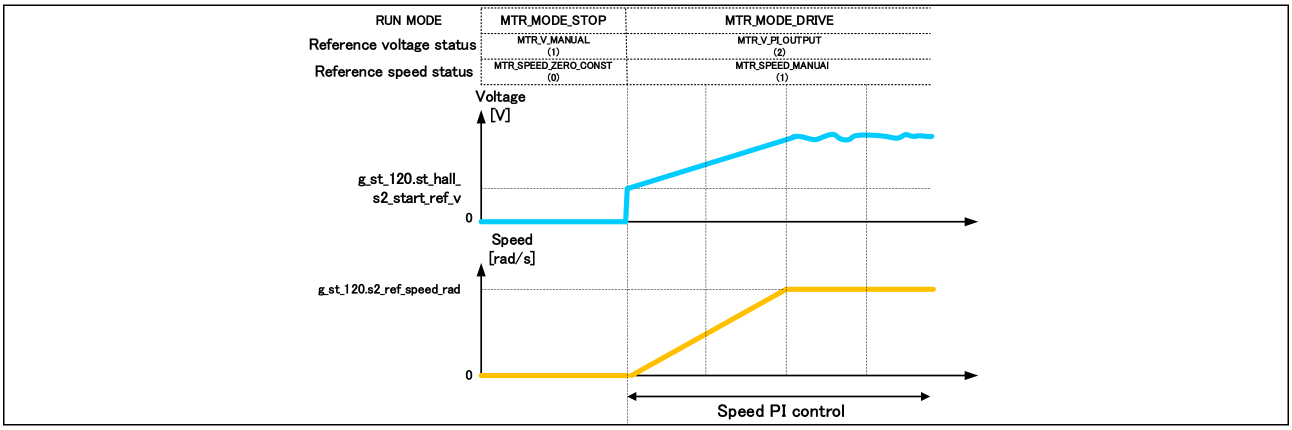


Figure 3-6 Start-up sequence (hall effect sensor control mode)

(2) Sensorless control mode

In sensorless control mode, the position of the magnetic poles is estimated every 60 degrees from induced voltage that is generated from the variation of magnetic flux due to the rotation of the permanent magnet (rotor). However, since the induced voltage is generated by the rotation, at low speed it is not possible to estimate the position of the rotor.

Therefore, the method to generate a rotating magnetic field by forcibly switching conducting pattern in the synchronous speed regardless position of rotor, is often used.

Figure 3-7 shows the start-up method in the sample software. In “MTR_MODE_DRIVE”, at first, the rotor is drawn in. Second, mode is changed to open-loop drive mode. After detecting the zero-crossing signal three times, mode is changed to PI control mode.

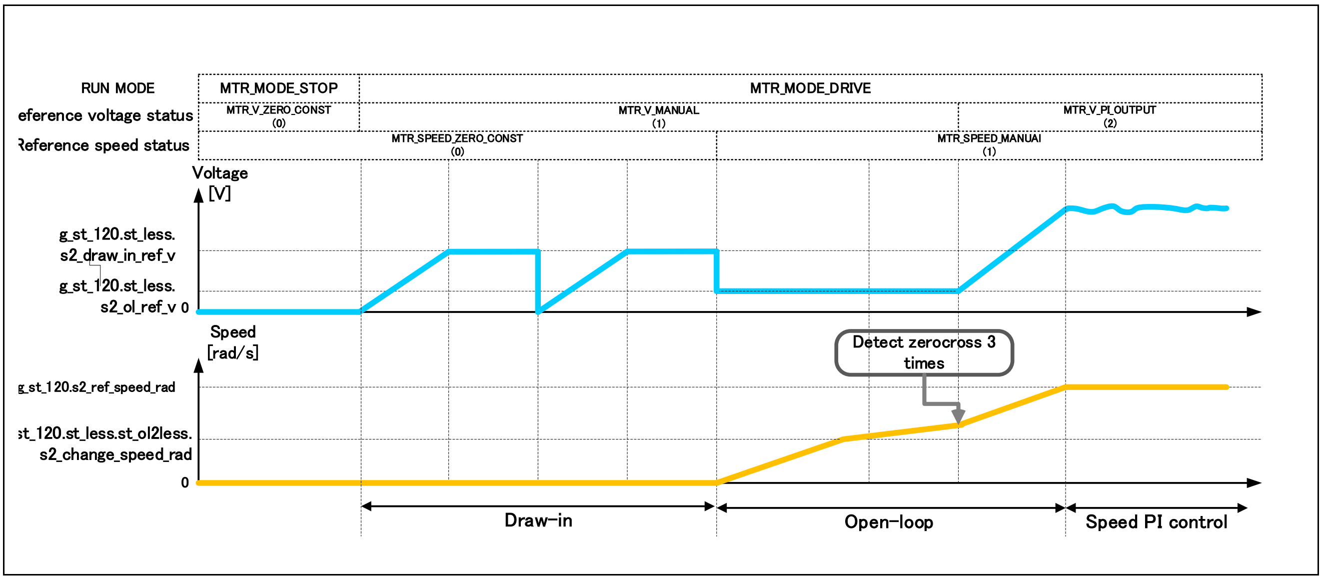


Figure 3-7 Start-up sequence (sensorless control mode)

3.1.7 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 3-5 for settings.

- Overcurrent error for hardware

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

Table 3-5 Setting Value of Each System Protection Function

Kinds of error	**Threshold**	
Overcurrent error	Over current limit [A]	2.0
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]	3900
	Monitoring cycle [ms]	1
Timeout error	Timeout value [ms]	200

3.1.8 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 3-6. Typically, the same base value is used for the physical quantity in same unit. These values are defined in “mtr_pu_system.h”.

Table 3-6 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	[ohm]	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]	Time * Torque / Angular frequency
Kp	[V/(rad/s)]	Use above Voltage/Angular frequency
Kidt	[V/(rad/s)]	Use above Voltage/Angular frequency

As shown in Figure 3-8, 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.

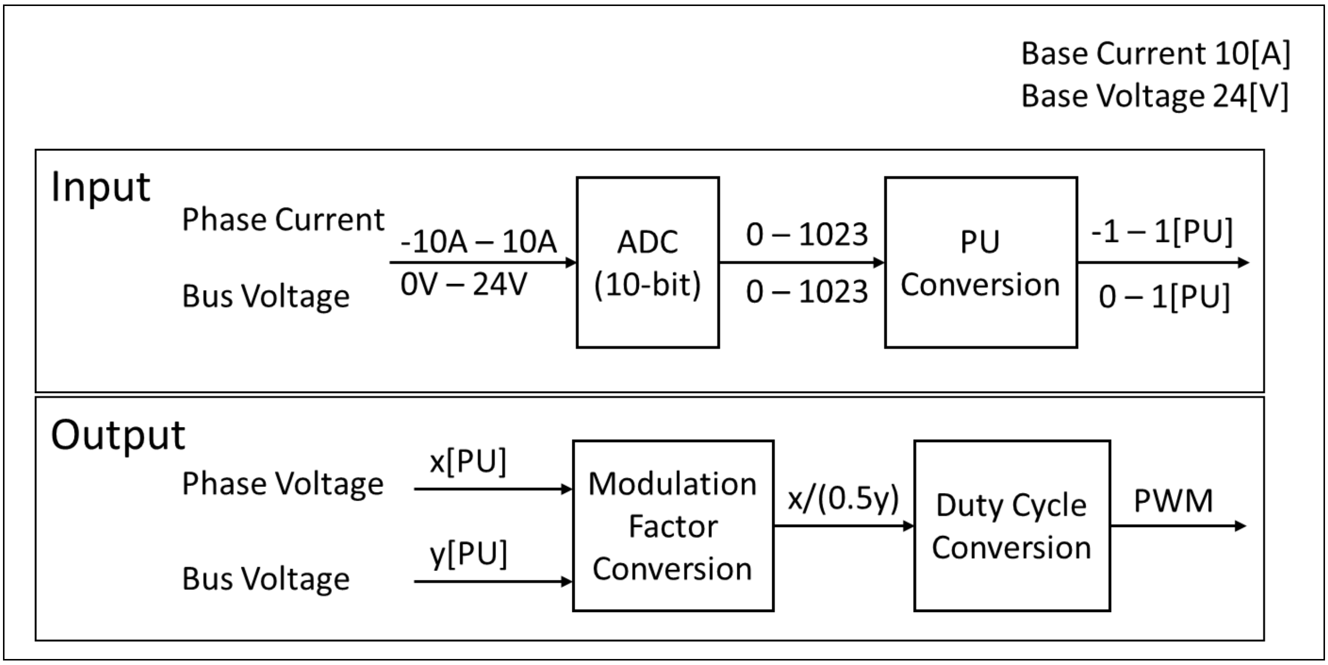


Figure 3-8 Motor control with using PU system

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 3-7 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 ICS 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 ICS UI (GUI) ・input values of command variables to ICS 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 ICS variables

Table 3-8 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 ICS variables ・input values of ICS variables to ICS 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 3-9 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 return: none	LED control depending on motor status and system mode
	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 3-10 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 3-11 List of Functions “r_mtr__ctrl_rl78g1f.h”

file	function	process
r_mtr_ctrl_rl78g1f.h	mtr_clear_trd_imfa() argument: TRDSR0_bit.no0 = 0 return: none	Clear IMFA flag
	mtr_set_tdr03() argument: TDR03 = cnt return: none	Set delay timer
	mtr_start_delay_cnt() argument: TS0L_bit.no3 = 1 return: none	Start delay timer
	mtr_stop_delay_cnt() argument: TT0L_bit.no3 = 0 return: none	Stop delay timer
	mtr_clear_inttm03() argument: TMIF03 = 0 return: none	Clear interruption flag
	clear_wdt() argument: WDTE = 0xAC return: none	Clear watchdog timer (WDT)

Table 3-12 List of Functions "r_mtr_ctrl_rl78g1f.c"

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_check_clk_error / error status return: none	Initialization of clock
	mtr_init_tau argument: none return: none	Initialization of timer array unit (TAU)
	mtr_init_trd argument: none return: none	Initialization of timer RD (TRD)
	mtr_init_intp argument: none return: none	Initialization of external interrupt (INTP)
	mtr_init_ad_converter argument: none return: none	Initialization of A/D converter
	R_MTR_hall_interrupt_enable argument: none return: none	【Hall effect sensor control mode】 Hall effect sensor interruption is enabled
	R_MTR_hall_interrupt_disable argument: none return: none	【Hall effect sensor control mode】 Hall effect sensor interruption is disabled
	R_MTR_get_adc 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_get_v_dcuvw_adc argument: (int16_t) *s2_v_dc / bus voltage (int16_t) *s2_v_uvw / UVW voltages return: none	【sensorless control mode】 Get the results of A/D conversion of bus and UVW voltage
	R_MTR_ctrl_stop argument: none return: none	Stop motor control ・prohibit output of voltage ・Hall effect sensor interruption is disabled 【Hall effect sensor control mode】 ・stop delay timer 【sensorless control mode】
	R_MTR_v_pattern_output 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 3-13 List of Functions "r_mtr_pu_system.c"

file	function	process
r_mtr_pu_system.c	mtr_pu_system_init argument: (float) f4_adc_voltage_scl / scale factor (uint16_t) u2_polepair / number of pole pairs return: none	Initialization of modules of per-unit
	mtr_conv_vdc_pu argument: (int16_t) s2_voltage_adc / voltage (ADC) return: mtr_flex_conv(&st_conv_adc2voltage,s2_voltage_adc) / voltage [PU]	PU conversion calculation of voltage
	mtr_conv_rpm2rad_pu argument: (int16_t) s2_speed_rpm / rotational speed [rpm] return: mtr_flex_conv(&st_conv_rpm2afreq_pu,s2_speed_rpm) / rotational speed [PU]	PU conversion calculation from mechanical [rpm] to electrical [PU] rotational speed
	mtr_conv_rad2rpm_pu argument: (int16_t) s2_ele_speed_pu / rotational speed [PU] return: mtr_flex_conv(&st_conv_afreq2rpm,s2_ele_speed_pu) / rotational speed [rpm]	PU conversion calculation from electrical [PU] to mechanical [rpm] rotational speed
	mtr_flex_conv_init argument: (mtr_frex_conv_t) *st_conv / structure (int8_t) s1_in_q / Q-format of input (int8_t) s1_out_q / Q-format of output (int8_t) s1_margin_q / margin bit of Q-format (float) f4_sf / scale factor return: none	Initialization of structures for per-unit
	mtr_flex_conv 【inline function】 argument: (mtr_flex_conv_t) *st_conv / structure (int16_t) s2_input / input value return: (int16_t)(((int32_t)s2_input*st_conv->s2_sf) >> n) / conversion value	PU conversion calculation process

Table 3-14 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 / ICS structure return: none	Input values of ICS variables to ICS buffer variables
	R_MTR_SetVariables 【inline function】 argument: none return: none	Input values of ICS buffer variables to control variables
	R_MTR_InputBuffParamReset argument: none return: none	Reset ICS 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_Get_Dir argument: none return: (uint8_t) g_st_120.u1_dir / direction of rotation	Get direction of rotation
	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 3-15 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 3-16 List of Functions "r_mtr_spm_120.c"

file name	function	process
r_mtr_spm_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 3-17 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_carrier_interrupt argument: none return: none	Carrier interruption (50 μs) 【Hall effect sensor control mode】 ・obtain bus voltage ・ICS communication 【sensorless control mode】 ・obtain bus and UVW voltages ・draw-in process ・transition from open-loop drive to sensorless control ・rotational speed calculation ・open-loop process ・set delay timer ・ICS communication
	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	ICS 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 return: (int16_t) s2_ref_v_delta / variation of output voltage	PI control calculation (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 3-18 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_hall argument: none return: none	【Hall effect sensor control mode】 TAU0 interruption (1 ms) ・set reference speed and voltage states ・calculate reference speed and voltage ・check error
	mtr_hall_u_interrupt argument: none return: none	【Hall effect sensor control mode】 Process for Hall effect sensor interruption ・set voltage pattern depending on signal from Hall effect sensor ・speed calculation
	mtr_hall_v_interrupt argument: none return: none	【Hall effect sensor control mode】 Process for Hall effect sensor interruption ・set voltage pattern depending on signal from Hall effect sensor ・speed calculation
	mtr_hall_w_interrupt argument: none return: none	【Hall effect sensor control mode】 Process for Hall effect sensor interruption ・set voltage pattern depending on signal from Hall effect sensor ・speed calculation
	mtr_hall_signal_set 【inline function】 argument: none return: none	【Hall effect sensor control mode】 Set voltage patten based on Hall signal
	mtr_hall_signal_process 【inline function】 argument: none return: none	【Hall effect sensor control mode】 Process for Hall effect sensor interruption ・set voltage pattern depending on signal from Hall effect sensor ・rotational speed calculation
	mtr_1ms_interrupt_less argument: none return: none	【sensorless control mode】 TAU0 interruption (1 ms) ・set reference speed and voltage states ・calculate reference speed and voltage ・check error
	mtr_delay_interrupt argument: none return: none	【sensorless control mode】 Delay timer interruption ・stop delay timer ・set voltage pattern
	mtr_draw_in_pattern_set 【inline function】 argument: none return: none	【sensorless control mode】 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	【sensorless control mode】 Estimate position of rotor from zero-crossing of induced voltage

Table 3-19 List of Functions "r_mtr_interrupt.c" [3/3]

file	function	process
r_mtr_interrupt.c	mtr_drive_openloop 【inline function】 argument: none return: none	【sensorless control mode】 Open-loop drive process
	mtr_set_angle_shift 【inline function】 argument: none return: none	【sensorless control mode】 Calculation of delay count after zero-crossing
	mtr_start_delay_timer 【inline function】 argument: (uint16_t) u2_delay_count / delay count return: none	【sensorless control mode】 Start delay timer
	mtr_stop_delay_timer 【inline function】 argument: none return: none	【sensorless control mode】 Stop delay timer
	mtr_openloop_pattern_set 【inline function】 argument: none return: (uint8_t) u2_pattern / voltage pattern	【sensorless control mode】 Set voltage pattern at open-loop drive
	mtr_ol2less_ctrl 【inline function】 argument: none return: none	【sensorless control mode】 Transition from open-loop drive to sensorless control ・comparison of induced voltage with threshold voltage ・count of zero-crossing at open-loop drive ・Judge whether phase of rotor is advanced or delayed
	mtr_get_bemf_voltage 【inline function】 argument: none return: (int16_t) s2_bemf_voltage / BEMF voltage	【sensorless control mode】 Get BEMF voltage
	mtr_ctrl_openloop_phase_ctrl 【inline function】 argument: none return: none	【sensorless control mode】 Judge whether phase of rotor is advanced or delayed

Table 3-20 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 3-21 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_u2_error_status	static uint16_t	Q0	-	Error status management	
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 3-22 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 3-23 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_ramp_limit_speed_rpm	int16_t	Q0	-	Limit of acceleration [rpm/ms]	Mechanical angle
com_f4_ramp_limit_v	float	-	-	Limit of variation of voltage [V/ms]	
com_f4_kp_speed	float	-	-	Proportional gain for speed PI control [V s/rad]	
com_f4_kidt_speed	float	-	-	Integral gain for speed PI control [V s/rad]	
com_f4_start_ref_v	float	-	-	Reference voltage at starting[V]	Hall effect sensor control mode
com_f4_draw_in_ref_v	float	-	-	Reference voltage at draw-in [V]	sensorless control mode
com_f4_ol_ref_v	float	-	-	Reference voltage at open-loop drive[V]	
com_s2_ol2less_speed_rpm	int16_t	Q0	-	Speed allowed to transition to PI control[rpm]	
com_s2_ol2less_ramp_speed_rpm	int16_t	Q0	-	Acceleration at transition to PI control [rpm/ms]	
com_s2_angle_shift_adjust	int16_t	Q0	-	Adjust delay counts	
com_s2_enable_write	int16_t	Q0	-	Variable to allow to input ICS structure	
g_s2_enable_write	int16_t	Q0	-	Variable to allow to input ICS structure	
st_ics_input	mtr_ctrl_input_t	Q0	-	Structure for ICS input	structure

Table 3-24 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	Q15	[1/Ω]	Resistance [PU]	
s2_mtr_ld	int16_t	Q15	[1/H]	D-axis inductance [PU]	
s2_mtr_lq	int16_t	Q15	[1/H]	Q-axis inductance [PU]	
s2_mtr_m	int16_t	Q12	[rad/V s]	Induced voltage constant [PU]	
s2_mtr_j	int16_t	Q7	[1/kg m^2]	Rotor inertia [PU]	

Table 3-25 List of variables "r_mtr_parameter.h / Structure : st_mtr_ctrl_gain_t"

variable	type	Qn	PU	content	remarks
s2_speed_pi_kp	int16_t	Q14	[rad/ V s]	Proportional gain for speed PI control [PU]	
s2_speed_pi_kidt	int16_t	Q15	[rad/ V s]	Integral gain for speed PI control [PU]	

Table 3-26 List of variables "r_mtr_pu_system.h / Structure : mtr_frex_conv_t"

variable	type	Qn	PU	content	remarks
s2_sf	int16_t	Q0	-	Scale factor	
s1_sf_q	int8_t	Q0	-	Q format	
s1_rsft	int8_t	Q0	-	Right shift	

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

variable	type	Qn	PU	content	remarks
st_conv_adc2voltage	mtr_frex_conv_t	-	-	Conversion from A/D to voltage	structure
st_conv_rpm2afreq_pu	mtr_frex_conv_t	-	-	Conversion from rpm to rad/s	
st_conv_afreq2rpm	mtr_frex_conv_t	-	-	Conversion from rad/s to rpm	

Table 3-28 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]	Reference rotational speed [PU]	electric angle
s2_ramp_limit_speed_rad	int16_t	Q13	[s/rad]	Limit of acceleration [PU]	electric angle
s2_ramp_limit_v	int16_t	Q14	[1/V]	Limit of variation of voltage [PU]	
s2_start_ref_v	int16_t	Q14	[1/V]	Reference voltage at starting [PU]	Hall effect sensor control mode
s2_draw_in_ref_v	int16_t	Q14	[1/V]	Reference voltage at draw-in [PU]	sensorless control mode
s2_ol_ref_v	int16_t	Q14	[1/V]	Reference voltage at open-loop drive [PU]	
s2_ol2less_speed_rad	int16_t	Q13	[s/rad]	Speed allowed to transition to PI control [PU]	
s2_ol2less_ramp_speed_rad	int16_t	Q13	[s/rad]	Acceleration at transition to PI control [PU]	
s2_less2ol_speed_rad	int16_t	Q13	[s/rad]	Speed to transition to open-loop drive [PU]	
s2_angle_shift_adjust	int16_t	Q0	-	adjust delay counts	
st_motor	st_mtr_parameter_t	-	-	structure for motor parameter	structure
st_gain	st_mtr_ctrl_gain_t	-	-	structure for PI control	

Table 3-29 List of variables “r_mtr_driver_access.c”

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

Table 3-30 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 3-31 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 3-32 List of variables “r_mtr_spm_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_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_limit	int16_t	Q14	[1/V]	Limit of voltage [PU]	
s2_pre_refp	int32_t	Q27	[1/V]	Previous proportional term [PU]	

Table 3-33 List of variables “r_mtr_spm_120.h / Structure : st_mtr_hall_control_t”

variable	type	Qn	PU	content	remarks
u2_hall_signal	uint16_t	Q0	-	Signal from Hall effect sensor	Hall sensor control model
u2_flag_1st_interrupt	uint16_t	Q0	-	Flag for first hall effect sensor interruption	
s2_start_ref_v	int16_t	Q14	[1/V]	Reference voltage at starting [PU]	

Table 3-34 List of variables “r_mtr_spm_120.h / Structure : st_mtr_ol2less_t”

variable	type	Qn	PU	content	remarks
u2_flag_change	uint16_t	Q0	-	Flag to allow to transition to PI control	sensorless control mode
u2_zc_flag	uint16_t	Q0	-	Flag for zero-crossing detection avoiding commutation	
u2_zc_cnt	uint16_t	Q0	-	Counter for zero-crossing detection	
u2_flag_change_speed	uint16_t	Q0	-	Flag for exceeding change speed	
u2_rotor_pos	uint16_t	Q0	-	Phase of rotor position	
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 3-35 List of variables "r_mtr_spm_120.h / Structure : st_mtr_sensorless_control_t"

variable	type	Qn	PU	content	remarks
u1_state_draw_in	uint8_t	Q0	-	Draw-in state management	sensorless control mode
u1_flag_pattern_change	uint8_t	Q0	-	Flag for zero-crossing detection	
u2_bemf_delay	uint16_t	Q0	-	Delay counts	
u2_bemf_signal	uint16_t	Q0	-	Estimated Hall pattern	
u2_pre_bemf_signal	uint16_t	Q0	-	Previous estimated Hall pattern	
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	uint16_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_less2ol_speed_rad	int16_t	Q13	[s/rad]	Speed to transition to open-loop drive [PU]	
s2_vu_ad	int16_t	Q0	-	Voltage of U phase	
s2_vv_ad	int16_t	Q0	-	Voltage of V phase	
s2_vw_ad	int16_t	Q0	-	Voltage of W phase	
s2_vn_ad	int16_t	Q0	-	Estimated neutral voltage	
s2_draw_in_ref_v	int16_t	Q14	[1/V]	Reference voltage at draw-in [PU]	
s2_ol_ref_v	int16_t	Q14	[1/V]	Reference voltage at open-loop drive [PU]	
s2_angle_shift_adjust	int16_t	Q0	-	Adjust delay counts	
st_ol2less	st_mtr_ol2less_t	-	-	Structure for transition to sensorless control	

Table 3-36 List of variables "r_mtr_spm_120.c"

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

Table 3-37 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	

3.4 Macro definitions of sensorless 120-degree conducting control software

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

Table 3-38 List of Macro definitions “r_mtr_config.h”

Macro	Definition value	Qn	PU	Content	Remarks
RL78_G1F_MRSSK	-	-	-	Select CPU board	
IP_MRSSK	-	-	-	Select inverter board	
MP_TG55L	-	-	-	Select motor parameters	
CP_TG55L	-	-	-	Select control parameters	
ICS_UI	0	-	-	RMW UI	Default
BOARD_UI	1	-	-	RSSK board UI	
MTRCONF_DEFAULT_UI	0/1	-	-	Select UI	BOARD_UI / ICS_UI
NON_COMPLEMENTARY	0	-	-	Non-complementary PWM	
COMPLEMENTARY	1	-	-	Complementary PWM	Default
MTRCONF_PWM_MODE	0/1	-	-	Select non-complementary/ complementary PWM	NON_COMPLEMENTARY / COMPLEMENTARY
HALL	0	-	-	Hall effect sensor	
LESS	1	-	-	Sensorless	Default
MTRCONF_SENSOR_MODE	0/1	-	-	Select sensor to detect position of rotor	HALL / LESS

Table 3-39 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	9.125f	-	-	Resistance [Ω]	
MP_D_INDUCTANCE	0.003844f	-	-	D-axis inductance [H]	
MP_Q_INDUCTANCE	0.004315f	-	-	Q-axis inductance [H]	
MP_MAGNETIC_FLUX	0.02144f	-	-	Induced voltage constant [V s/rad]	
MP_ROTOR_INERTIA	0.000002050f	-	-	Rotor inertia [kgm^2]	
MP_NOMINAL_CURRENT_RMS	0.42f	-	-	Nominal current [A]	

Table 3-40 List of Macro definitions “r_mtr_control_parameter.h”

Macro	Definition value	Qn	PU	content	Remarks
CP_MAX_SPEED_RPM	3200	-	-	Maximum limit of command rotational speed [rpm]	Mechanical angle
CP_MIN_SPEED_RPM	530【Hall effect sensor control mode】/265【sensorless control mode】	-	-	Minimum limit of command rotational speed [rpm]	Mechanical angle
CP_LIMIT_SPEED_RPM	3900	-	-	Maximum limit of estimated rotational speed [rpm]	Mechanical angle
CP_RAMP_LIMIT_SPEED_RPM	120(Hall effect sensor control mode) /6(sensorless control mode)	-	-	Limit of acceleration [rpm/ms]	Mechanical angle
CP_RAMP_LIMIT_V	0.29	-	-	Limit for variation of voltage [V]	
CP_SPEED_PI_KP	0.027410176f	-	-	Proportional gain for speed PI control [V s/rad]	
CP_SPEED_PI_KIDT	0.000861116f	-	-	Integral gain for speed PI control [V s/rad]	
CP_START_REF_V	3.6f	-	-	Initial voltage [V]	Hall effect sensor control mode
CP_OL2HALL_SPEED_RPM	530	-	-	Speed to transition to open-loop drive [rpm]	
CP_DRAW_IN_REF_V	20.0f	-	-	Reference voltage at draw-in[V]	sensorless control mode
CP_OL_REF_V	4.3f	-	-	Reference voltage at open loop [V]	
CP_OL2LESS_SPEED_RPM	530	-	-	Speed allowed to transition to PI control [rpm]	
CP_OL2LESS_SPEED_RAMP_RPM	1	-	-	Acceleration at transition to PI control [rpm/ms]	

Table 3-41 List of Macro definitions “r_mtr_inverter_parameter.h”

Macro	Definition value	Qn	PU	content	Remarks
IP_DEADTIME	2.0f	-	-	Deadtime[us]	
IP_VDC_RANGE	111.0f	-	-	Range of bus voltage [V]	
IP_INPUT_V	24.0f	-	-	Input voltage [V]	
IP_OVERVOLTAGE_LIMIT	28.0f	-	-	Upper limit of voltage [V]	
IP_UNDERVOLTAGE_LIMIT	15.0f	-	-	Lower limit of voltage [V]	

Table 3-42 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 3-43 List of Macro definitions “ICS_define.h”

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

Table 3-44 List of Macro definitions “r_mtr_ics.h”

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

Table 3-45 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_MARGIN	400	-	-	Margin value for VR1	
VR1_SCALING	(CP_MAX_SPEED_RPM+ VR1_MARGIN)/0x0200	-	-	Scaling factor for speed calculation	
VR1_OFFSET	0x1FF	-	-	Offset for VR1	

Table 3-46 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_TAU0_PERIOD	0.001f	-	-	Period of TAU0	
MTR_VDC_SCALING	IP_VDC_RANGE/1023.0f	-	-	Scaling factor to convert to voltage	
MTR_PU_Q_VDC_SCALING	MTR_VDC_SCALING * PU_SF_VOLTAGE * (1 << MTR_Q_VOLTAGE)	Q14	[1/V]	Scaling factor to convert to voltage	PU
MTR_CARRIER_SET	(MTR_PWM_TIMER_FREQ*1000/MTR_CARRIER_FREQ/2)-2 【non-complementary PWM】 / (MTR_PWM_TIMER_FREQ*1000/MTR_CARRIER_FREQ/2)-2 【complementary PWM】	-	-	Resister counts of carrier	
MTR_DEADTIME	2000	-	-	deadtime[ns]	Complementary PWM
MTR_DEADTIME_SET	MTR_DEADTIME*MTR_PWM_TIMER_FREQ/1000	-	-	Resister counts of deadtime	
MTR_NDT_CARRIER_SET	MTR_CARRIER_SET-MTR_DEADTIME_SET	-	-	Resister counts of carrier without deadtime	
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_ADCCH_VR1	6	-	-	A/D converter channel of VR1	
MTR_ADCCH_VDC	4	-	-	A/D converter channel of bus voltage	

Table 3-47 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_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 3-48 List of Macro definitions “r_mtr_ctrl_rl78g1f.h”[3/3]

Macro	Definition value	Qn	PU	content	Remarks
ERROR_NONE	0x00	-	-	None error	
ERROR_CHANGE_CLK_TIMEOUT	0x01	-	-	Timeout error of change of resister for clock	
ERROR_CHARGE_CAP_TIMEOUT	0x02	-	-	Timeout error of capacitor charge	

Table 3-49 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 3-50 List of Macro definitions "r_mtr_fixed.h"

Macro	Definition value	Qn	PU	content	Remarks
MTR_Q_CURRENT	12	-	-	Q-format of current	
MTR_Q_VOLTAGE	14	-	-	Q-format of voltage	
MTR_Q_AFREQ	3	-	-	Q-format of angular frequency	
MTR_Q_FREQ	MTR_Q_AFREQ	-	-	Q-format of frequency	
MTR_Q_RES	15	-	-	Q-format of resistance	
MTR_Q_IND	15	-	-	Q-format of inductance	
MTR_Q_FLUX	12	-	-	Q-format of induced voltage constant	
MTR_Q_INERTIA	7	-	-	Q-format of inertia	
MTR_Q_SPEED_KP	14	-	-	Q-format of proportional gain	
MTR_Q_SPEED_KIDT	15	-	-	Q-format of integral gain	
LSFT_VOLTAGE_2KIDT_AFREQ	MTR_Q_SPEED_KIDT + MTR_Q_AFREQ - MTR_Q_VOLTAGE	-	-	Left shift, (KIDT * speed) to voltage	
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 3-51 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_SPEED_CALC_BASE	125000 * MTR_TWOPI	-	-	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_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	PU
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_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	
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 * 0.8	-	-	MCU stable supply voltage	

Table 3-52 List of Macro definitions “r_mtr_pu_system.h”

Macro	Definition value	Qn	PU	content	Remarks
INV_LOG10_2	1.0f / 0.3010299957f	-	-	1/log10(2)	
PU_BASE_CURRENT_A	MP_NOMINAL_CURRENT_RMS	-	-	Based current for per-unit	
PU_BASE_VOLTAGE_V	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	PU_BASE_FREQ_HZ	-	-	Scale factor to convert from [s] to PU	
PU_SF_RES	PU_BASE_CURRENT_A / PU_BASE_VOLTAGE_V	-	-	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_TIME / PU_SF_AFREQ	-	-	Scale factor to convert from [kg m^2] to PU	
PU_SF_SPEED_KP	PU_SF_VOLTAGE / PU_SF_AFREQ	-	-	Scale factor to convert from [Vs/rad] to PU	
PU_SF_SPEED_KIDT	PU_SF_SPEED_KP	-	-	Scale factor to convert from [Vs/rad] to PU	

Table 3-53 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 3-54 List of Macro definitions “r_mtr_spm_120.h”[1/3]

Macro	Definition value	Qn	PU	content	Remarks
MTR_TIMEOUT_CNT	200	-	-	Counts for timeout	
MTR_INIT_CNT_CARRIER	300	-	-	Initial carrier counts	
MTR_LIMIT_SPEED_RAD	MP_POLE_PAIRS * MTR_TWOPI_60 * CP_LIMIT_SPEED_RPM	-	-	Maximum limit of rotational speed [rpm]	
MTR_HALL2OL_REV_SPEED_RAD	FIX_fromfloat(CP_HALL2OL_REV_SPEED_RPM * PU_SF_AFREQ, MTR_Q_AFREQ)	Q13	-	Speed to transition to PI control at reverse of direction [PU]	Hall effect sensor control mode
MTR_PHASE_ADV	0	-	-	Advanced phase of rotor	sensorless control mode
MTR_PHASE_DLY	1	-	-	Delayed phase of rotor	
MTR_OL2LESS_ZC_CNT	3	-	-	Counts of zero-crossing at transition to PI control	
MTR_AVOID_COMMUTATION	4	-	-	Counts for avoiding to detect zero-crossing after commutation	
MTR_OL2LESS_BEMF_THRESH	3	-	-	Threshold voltage at transition to PI control	
MTR_LESS2OL_HYSTERESIS	50	-	-	Width of speed hysteresis between open-loop and PI control [rpm]	
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_PU_Q_OVERVOLTAGE_LIMIT	FIX_fromfloat(IP_OVERVOLTAGE_LIMIT * PU_SF_VOLTAGE, MTR_Q_VOLTAGE)	Q14	[1/V]	Maximum limit for voltage	PU
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	

Table 3-55 List of Macro definitions “r_mtr_spm_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	5【Hall effect sensor control mode】/ 3【sensorless control mode】	-	-	Voltage pattern at CCW rotation	
MTR_PATTERN_CCW_V_W	1【Hall effect sensor control mode】 /2【sensorless control mode】	-	-		
MTR_PATTERN_CCW_U_W	3【Hall effect sensor control mode】/ 6【sensorless control mode】	-	-		
MTR_PATTERN_CCW_U_V	2【Hall effect sensor control mode】/ 4【sensorless control mod】	-	-		
MTR_PATTERN_CCW_W_V	6【Hall effect sensor control mode】/ 5【sensorless control mode】	-	-		
MTR_PATTERN_CCW_W_U	4【Hall effect sensor control mode】/ 1【sensorless control mode】	-	-		
MTR_PATTERN_NUM	6	-	-	Number of voltage patterns	
MTR_ERROR_NONE	0x00	-	-	No error	
MTR_ERROR_OVER_CURRENT	0x01	-	-	Over current error	
MTR_ERROR_OVER_VOLTAGE	0x02	-	-	Over voltage error	
MTR_ERROR_OVER_SPEED	0x04	-	-	Over speed error	
MTR_ERROR_HALL_TIMEOUT	0x08	-	-	Timeout error for Hall effect sensor control mode	
MTR_ERROR_BEMF_TIMEOUT	0x10	-	-	Timeout error for sensorless control mode	
MTR_ERROR_HALL_PATTERN	0x20	-	-	Hall pattern error	
MTR_ERROR_BEMF_PATTERN	0x40	-	-	BEMF pattern error	
MTR_ERROR_UNDER_VOLTAG E	0x80	-	-	Under voltage error	
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 3-56 List of Macro definitions “r_mtr_spm_120.h”[3/3]

Macro	Definition value	Qn	PU	content	Remarks
MTR_SPEED_ZERO_CONST	0	-	-	Reference speed 0 const mode	
MTR_SPEED_MANUAL	1	-	-	Reference speed manual input mode	
MTR_V_ZERO_CONST	0	-	-	Reference voltage zero const mode	
MTR_V_MANUAL	1	-	-	Reference voltage manual input mode	
MTR_V_PI_OUTPUT	2	-	-	Reference voltage PI output mode	

3.5 Control flows (flow charts)

3.5.1 Main process

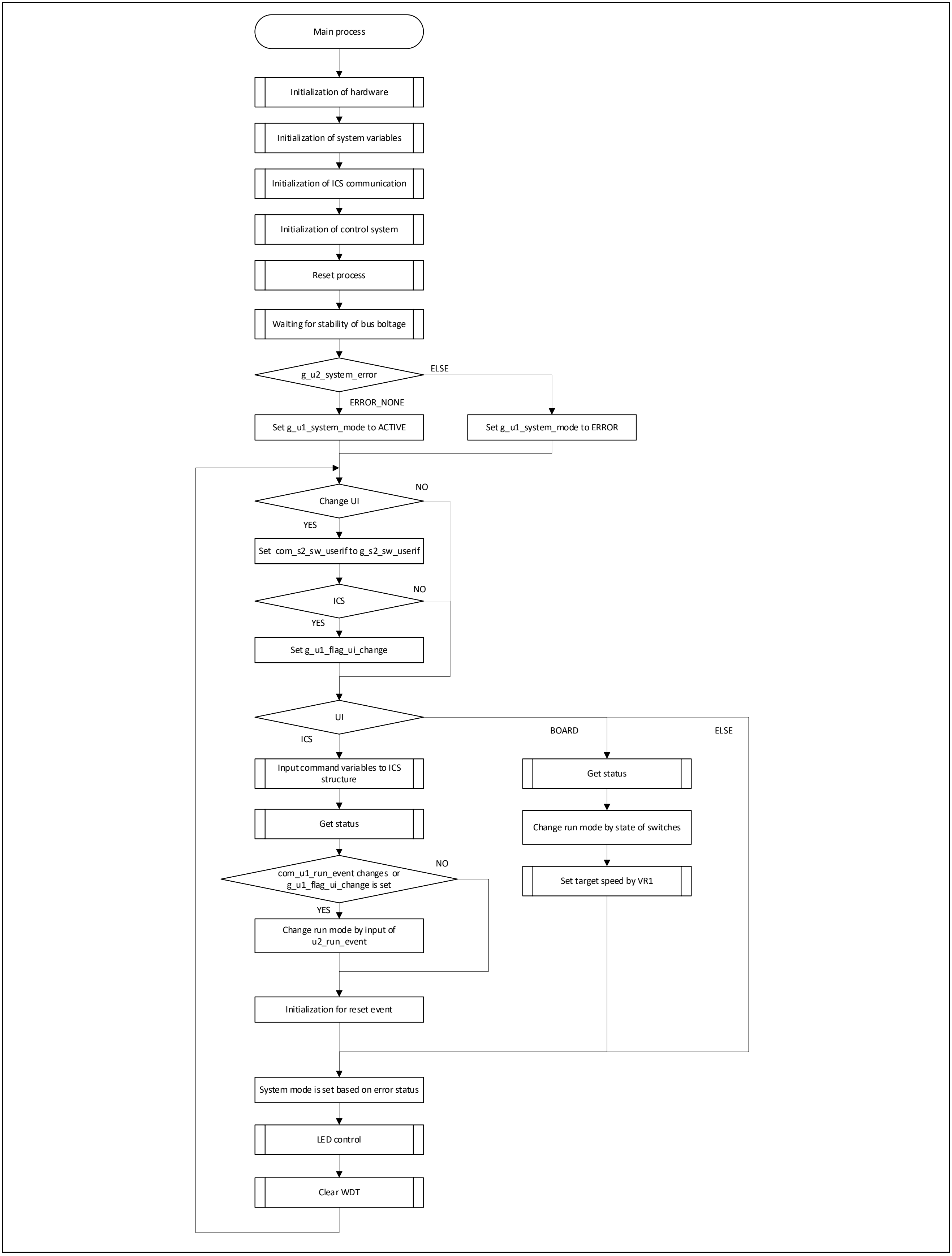


Figure 3-9 Main Process Flowchart

3.5.2 Carrier cycle interrupt handling

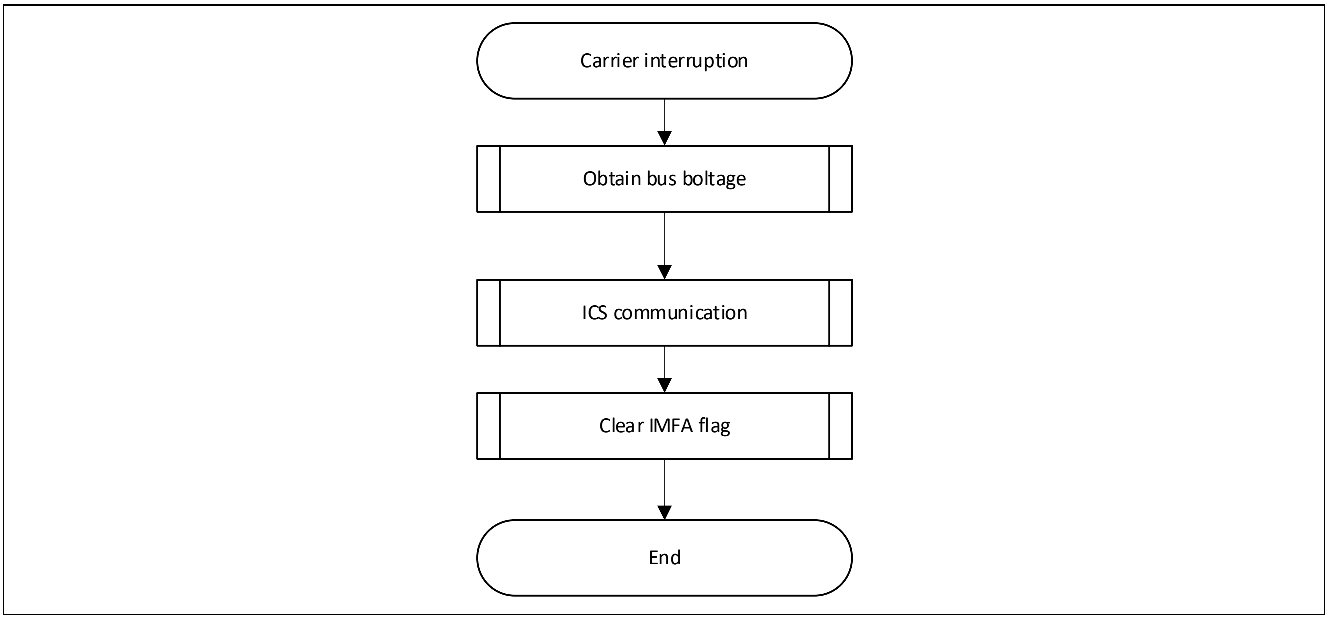


Figure 3-10 50 [μs] Cycle Interrupt Handling (Hall effect sensor control mode)

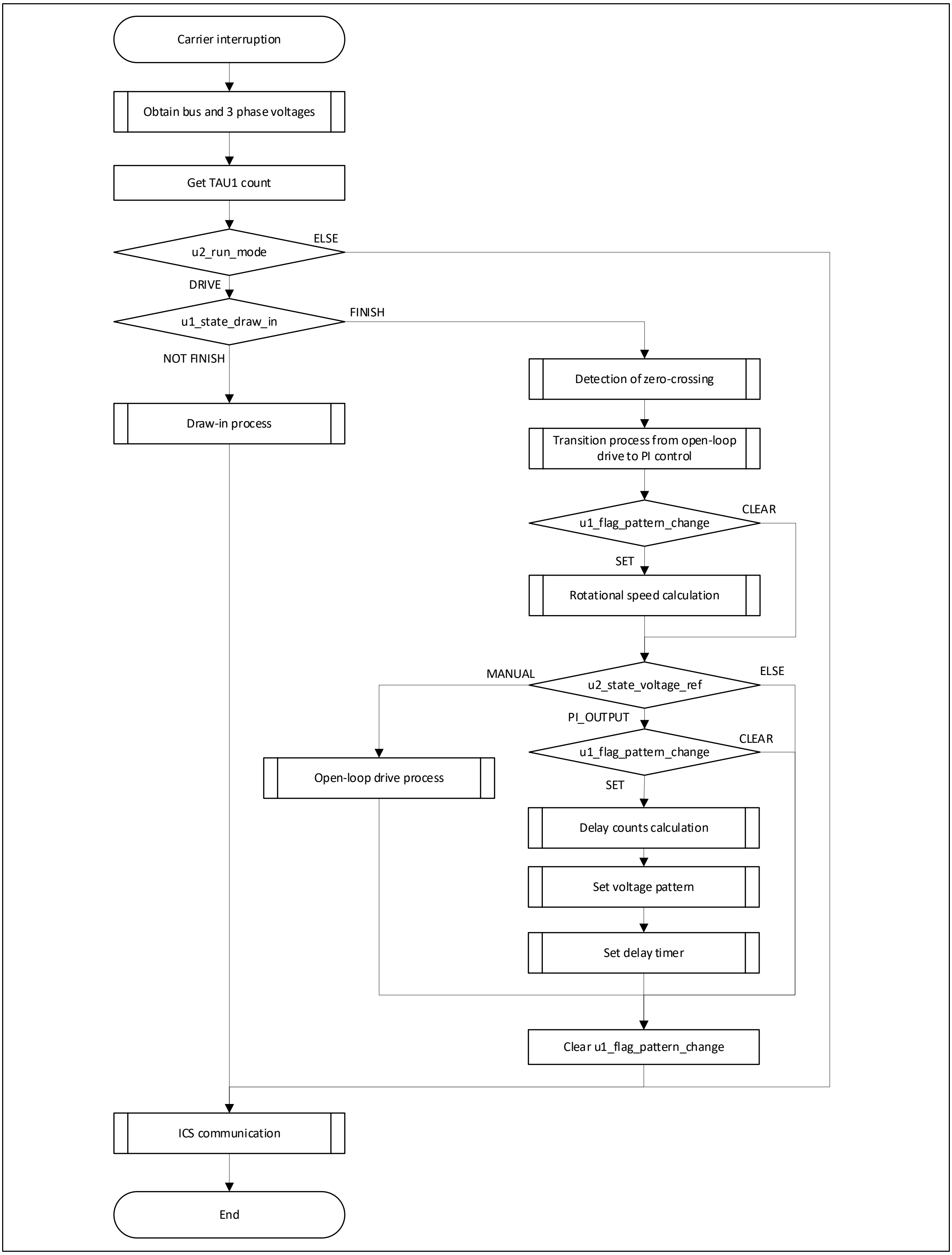


Figure 3-11 50 [μs] Cycle Interrupt Handling (sensorless control mode)

3.5.3 1 [ms] interrupt handling

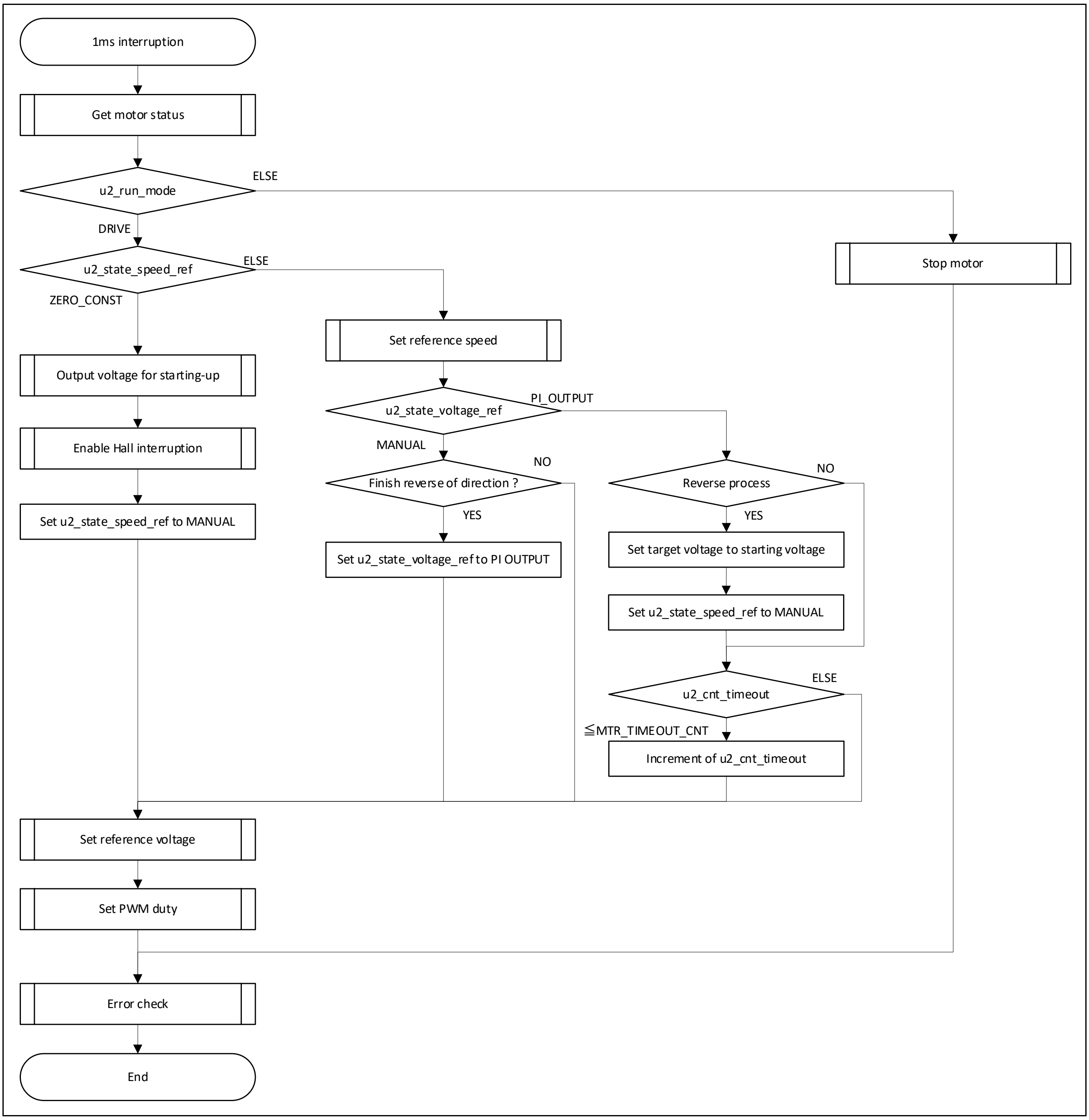


Figure 3-12 1 [ms] Interrupt Handling(Hall effect sensor control mode)

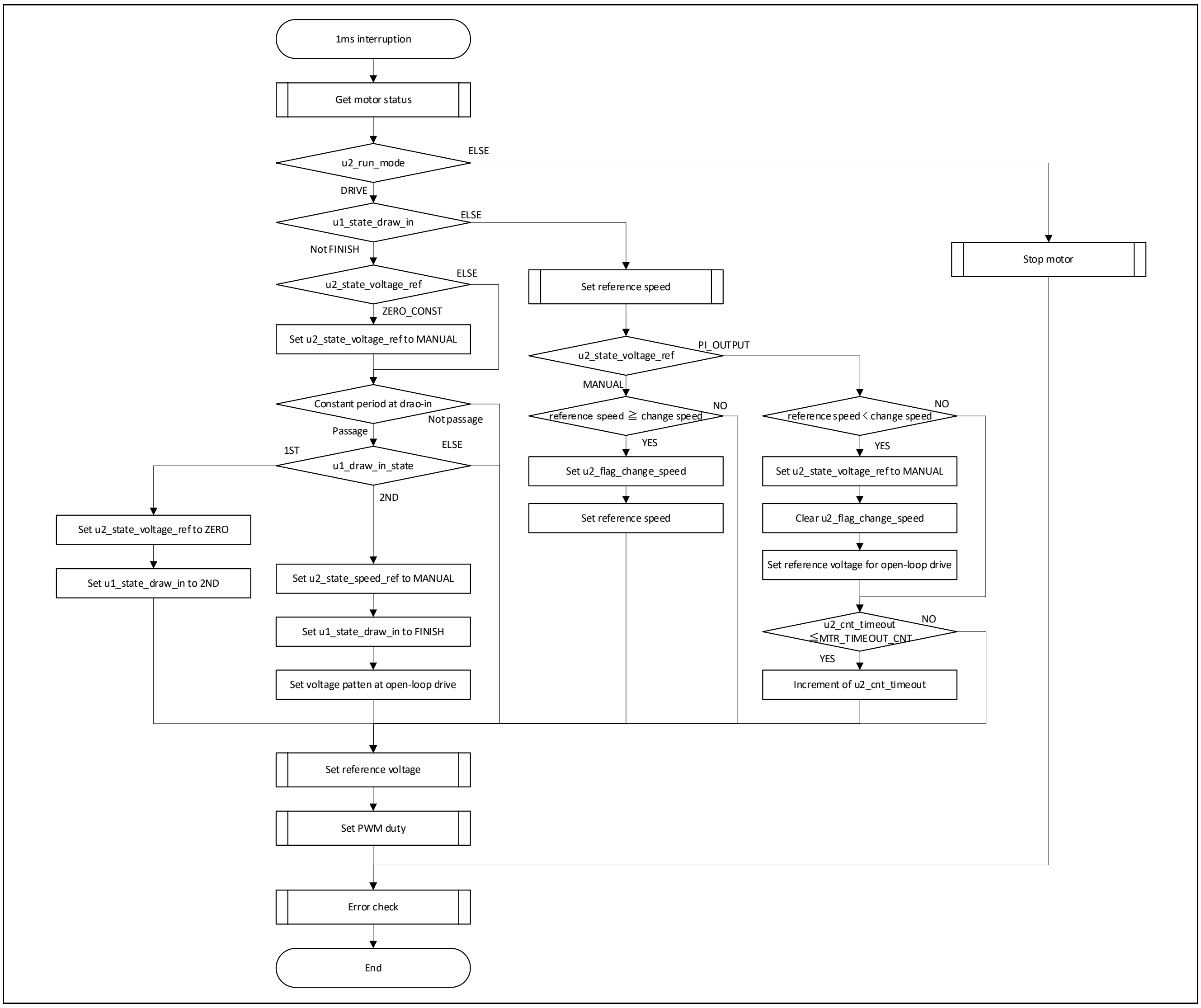


Figure 3-13 1 [ms] Interrupt Handling(sensorless control mode)

3.5.4 Overcurrent interrupt handling

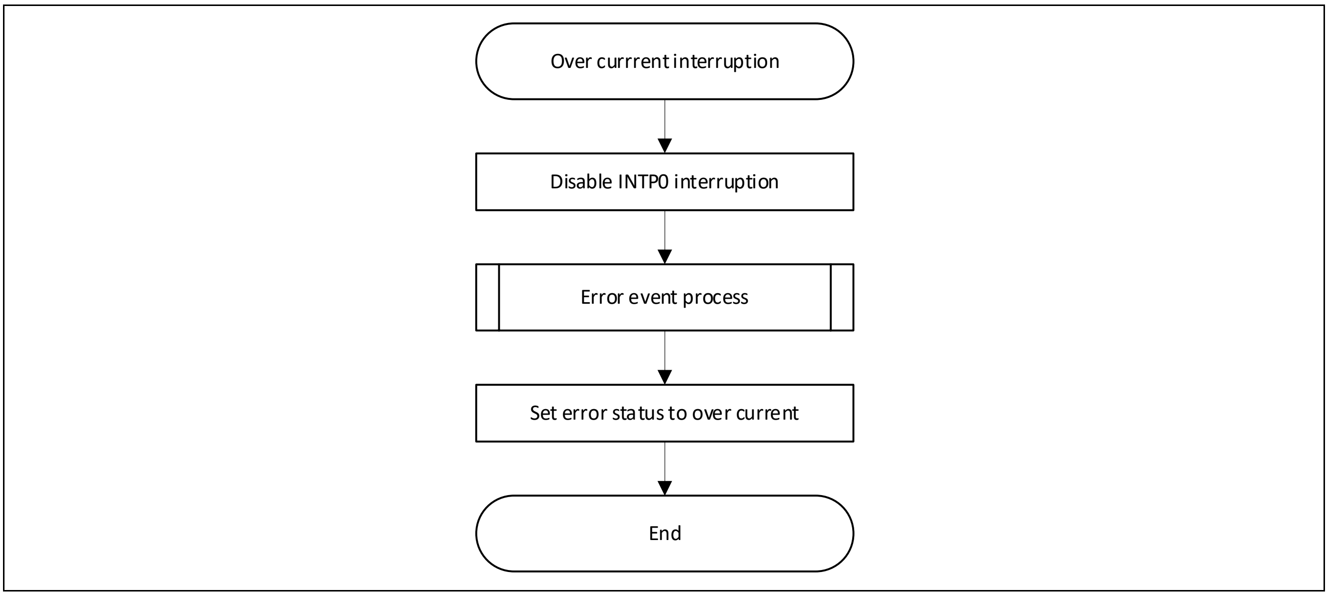


Figure 3-14 Over Current Detection Interrupt Handling

3.5.5 Hall effect sensor interrupt handling

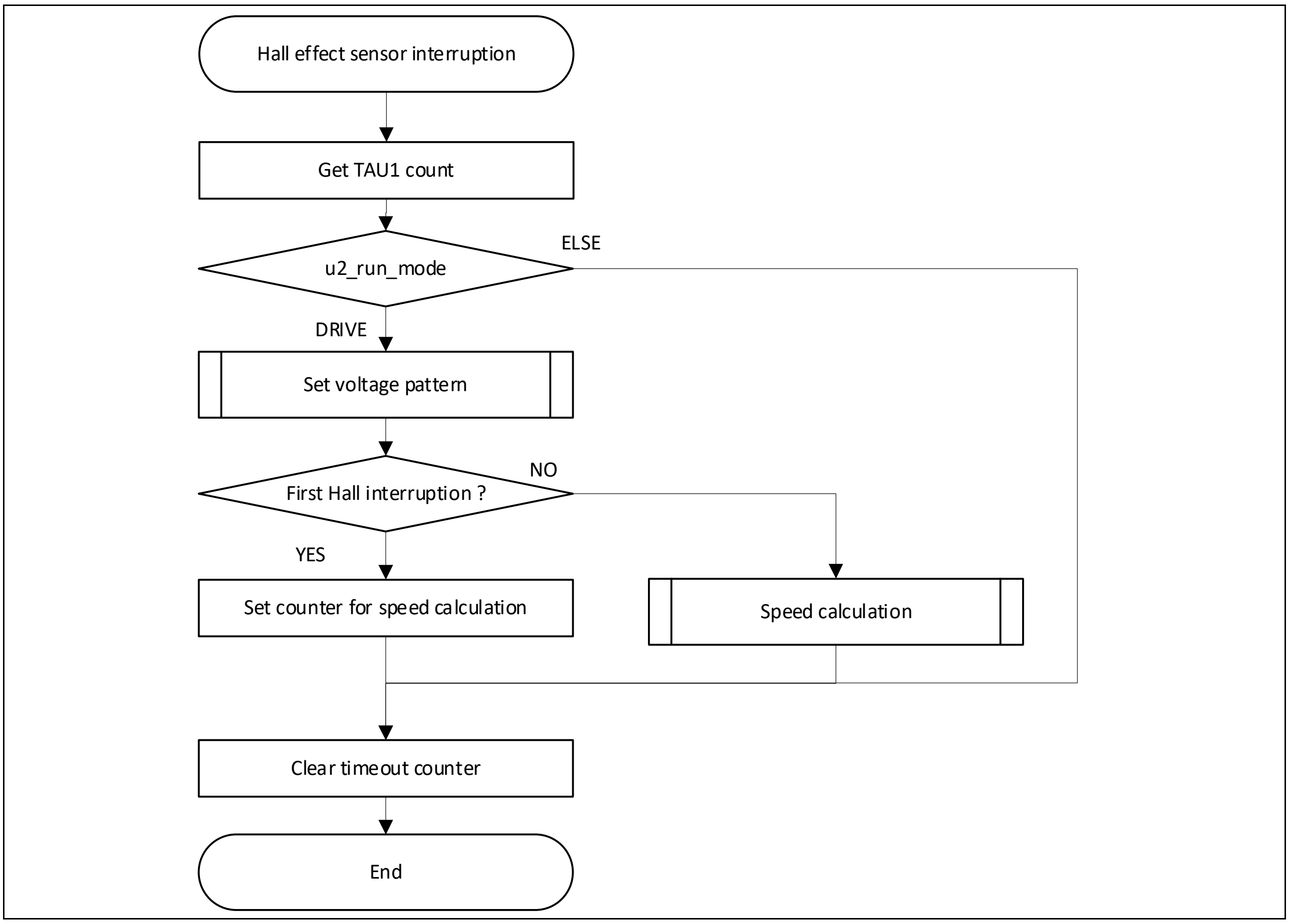


Figure 3-15 Hall effect sensor interrupt handling

3.5.6 Delay timer interrupt handling

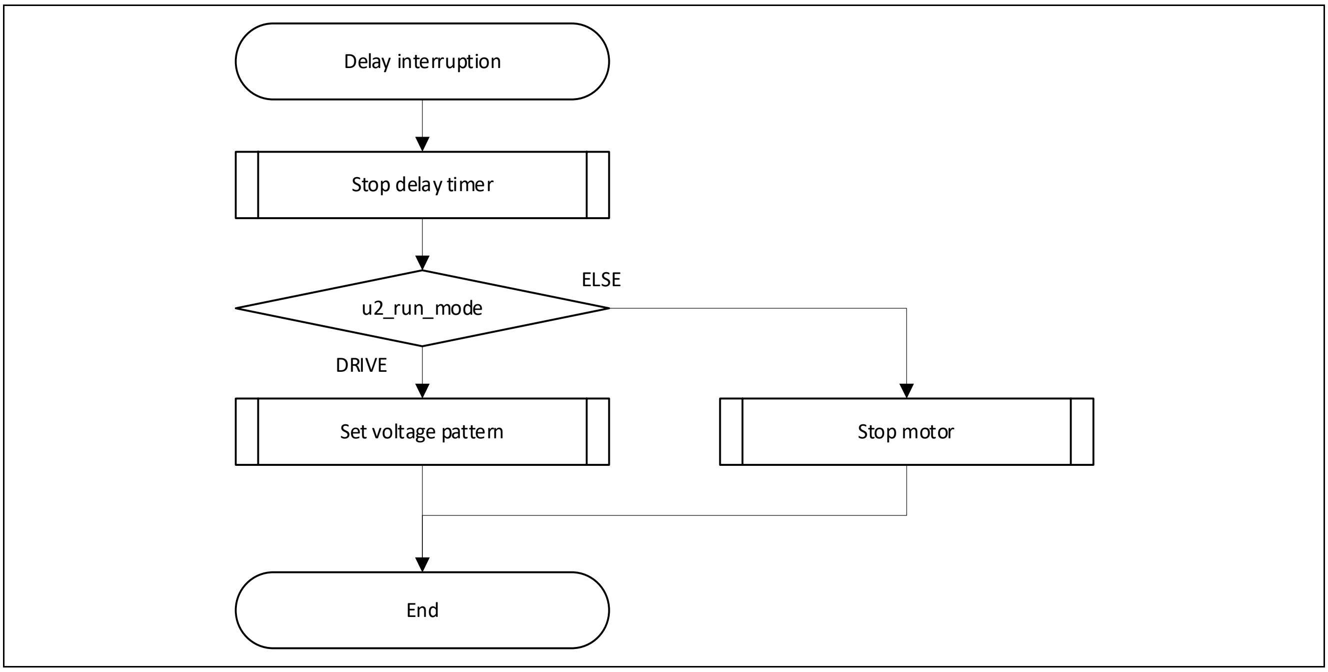


Figure 3-16 Delay Timer Interrupt Handling

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 2.0 User's Manual' for usage and more details. You can find 'Renesas Motor Workbench' on Renesas Electronics Corporation website.

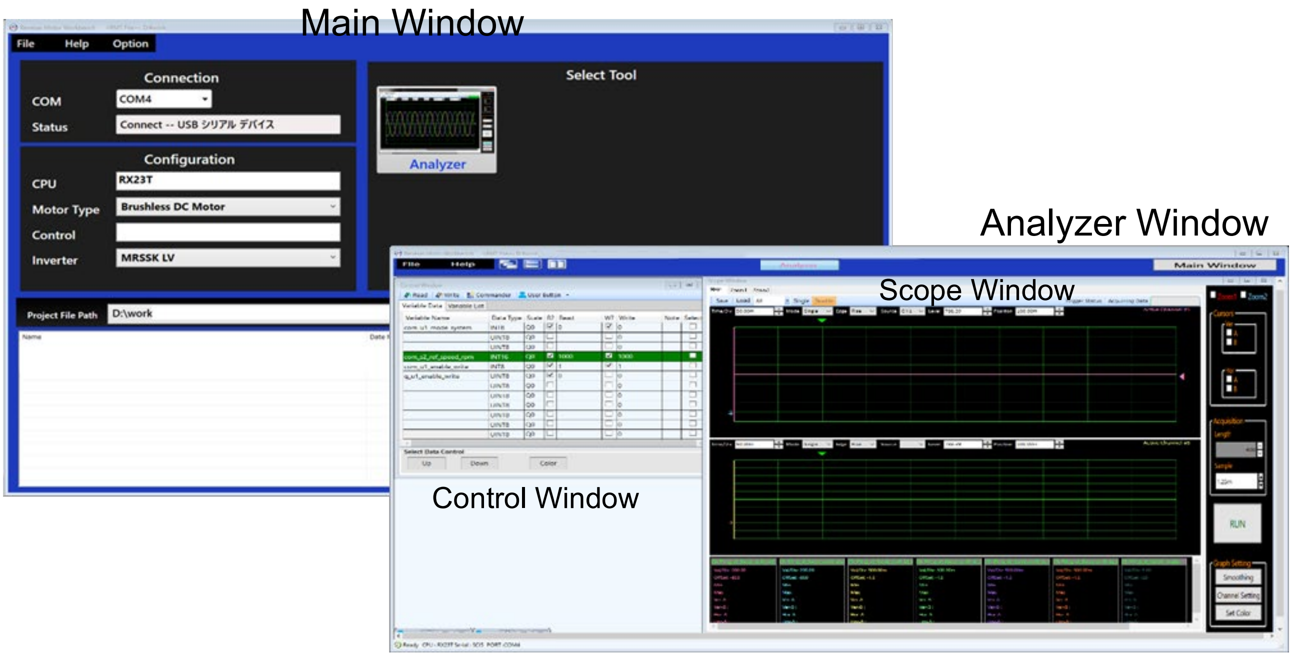
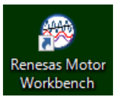


Figure 4-1Renesas Motor Workbench– Appearance

<u>Set up for Renesas Motor Workbench</u>

(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

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_motor.u2_mtr_pp】
com_f4_mtr_r	float	Resistance [Ω]	【g_st_120.st_motor.s2_mtr_r】
com_f4_mtr_ld	float	D-axis inductance[H]	【g_st_120.st_motor.s2_mtr_ld】
com_f4_mtr_lq	float	Q-axis inductance[H]	【g_st_120.st_motor.s2_mtr_lq】
com_f4_mtr_m	float	Induced voltage constant [V s/rad]	【g_st_120.st_motor.s2_mtr_m】
com_f4_mtr_j	float	Rotor inertia[kgm^2]	【g_st_120.st_motor.s2_mtr_j】
com_s2_ref_speed_rpm	int16_t	Command rotational speed［rpm］	【g_st_120.s2_ref_speed_rad】
com_s2_ramp_limit_speed_rpm	int16_t	Limit of acceleration [rpm/ms]	【g_st_120.s2_ramp_limit_speed_rad】
com_f4_ramp_limit_v	float	Limit of variation of voltage [V/ms]	【g_st_120.s2_ramp_limit_v】
com_f4_kp_speed	float	Proportional gain for speed PI control [V s/rad]	【g_st_120.st_pi_speed.s2_kp】
com_f4_kidt_speed	float	Integral gain for speed PI control [V s/rad]	【g_st_120.st_pi_speed.s2_kidt】
com_f4_start_ref_v	float	Reference voltage at starting[V]	【g_st_120.st_hall.s2_start_ref_v】
com_f4_draw_in_ref_v	float	Reference voltage at draw-in [V]	【g_st_120.st_less.s2_draw_in_ref_v】
com_f4_ol_ref_v	float	Reference voltage at open-loop drive[V]	【g_st_120.st_less.s2_ol_ref_v】
com_s2_ol2less_speed_rpm	int16_t	Speed allowed to transition to PI control[rpm]	【g_st_120.st_less.st_ol2less.s2_change_speed_rad】
com_s2_ol2less_ramp_speed_rpm	int16_t	Acceleration at transition to PI control [rpm/ms]	【g_st_120.st_less.st_ol2less.s2_ramp_speed_rad】
com_s2_angle_shift_adjust	int16_t	Adjust delay counts	【g_st_120.st_less.s2_angle_shift_adjust】
com_s2_enable_write	int16_t	Variable to allow to input ICS 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 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.

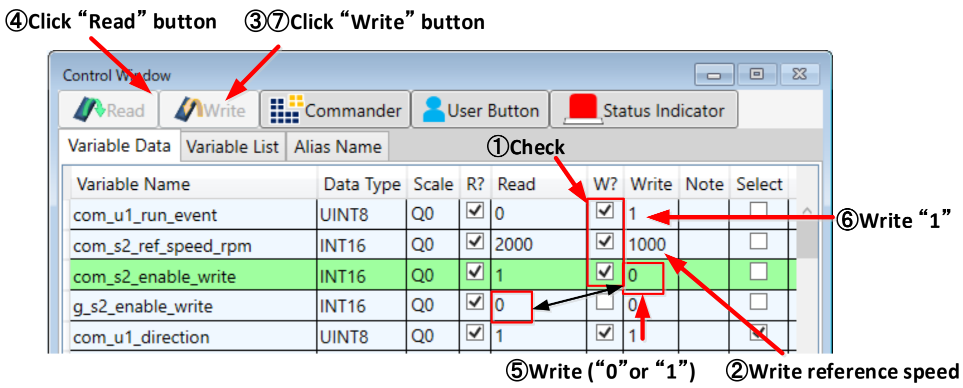


Figure 4-2 Procedure - Driving the motor

・Stop the motor

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

(2) Click the “Write” button.

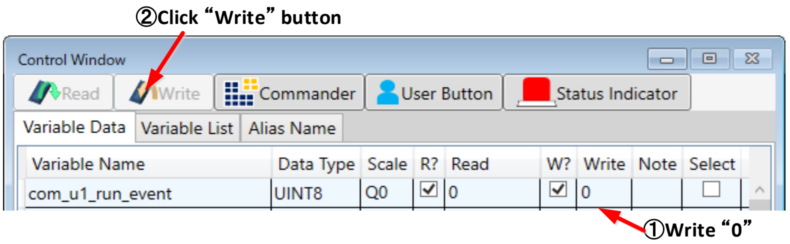


Figure 4-3 Procedure - Stop the motor

・Error cancel operation

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

(2) Click the “Write” button.

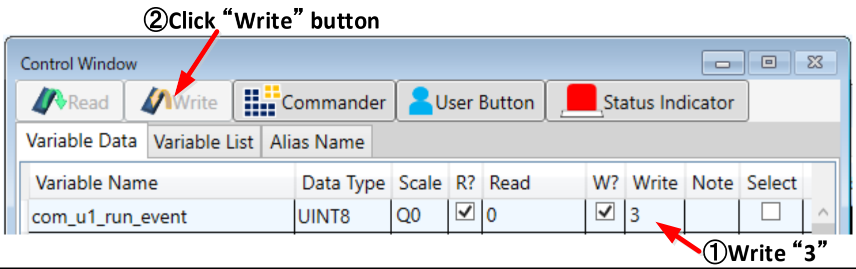


Figure 4-4 Procedure - Error cancel operation

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Oct.1.2016	-	First edition issued
1.10	Apr.5.2016	-	-Follow IDE Version Up -Addition to follow Renesas CPU board -Addition to follow “Renesas Motor Workbench” -Clerical corrections
2.00	Apr.1.2020	-	Significant changes in accordance with sample program

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

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

1. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on

   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins

   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Corporate Headquarters

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

Trademarks

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/.