RL78/F14

R01AN3809EJ0100
Rev.1.00
Mar.31.2017

Motor control by RL78/F14 micro controller
120 degrees conducting control of blushless DC motor with hall sensor

Summary

This application note aims at explaining the sample program for operating the 3-phase brushless DC motor with hall sensor by 120 degrees conducting method, by using functions of RL78/F14.

The sample program is only to be used as reference and Renesas Electronics Corporation does not guarantee the operations. Before using this sample program, carry out a thorough evaluation in a suitable environment.

Operation checking device

Operations of the sample program are checked by using the following device.

- RL78/F14 (R5F10PLJ)

Contents

1. Overview

This application note describes an example of speed control by brushless DC motor with hall sensor (here in after referred to as BLDC motor) by using micro controller RL78/F14. The speed control is performed by 120 degrees conducting.

1.1 Usage of the system

This system (sample program) enables 120 degrees conducting control by using an RL78/F14 micro controller mounted CPU board, an inverter board for motor control (ECU001-F14-12V [note 1]) and a BLDC motor (BLY171S-15V-8000 [note 2])

Notes: 1. Evaluation board "ECU001-F14-12V" is products of Desk Top Lab Inc.
(http://www.desktoplab.co.jp/)

2. BLDC Motor "BLY171S-15V-8000" is products of Anaheim Automation Inc..
(http://www.anaheimautomation.com/)

1.2 Development environment

(1) Software development environment

Integrated development environment	CS+ for CA, CX (V4.00.00) CS+ for CC (V5.00.00)
	IAR Embedded Workbench (Ver. 7.4.1.4269)
Build tool	CA78K0R (V1.72) CC-RL (V1.04.00)
	EWRL78 (Ver. 2.21.1)

(2) Hardware development environment

On-chip debug emulator	E1
Microcomputer used	RL78/F14(R5F10PLJ)
RL78/F14mounted CPU board	ECU001-F14-12V
BLDC motor	BLY171S-15V-8000

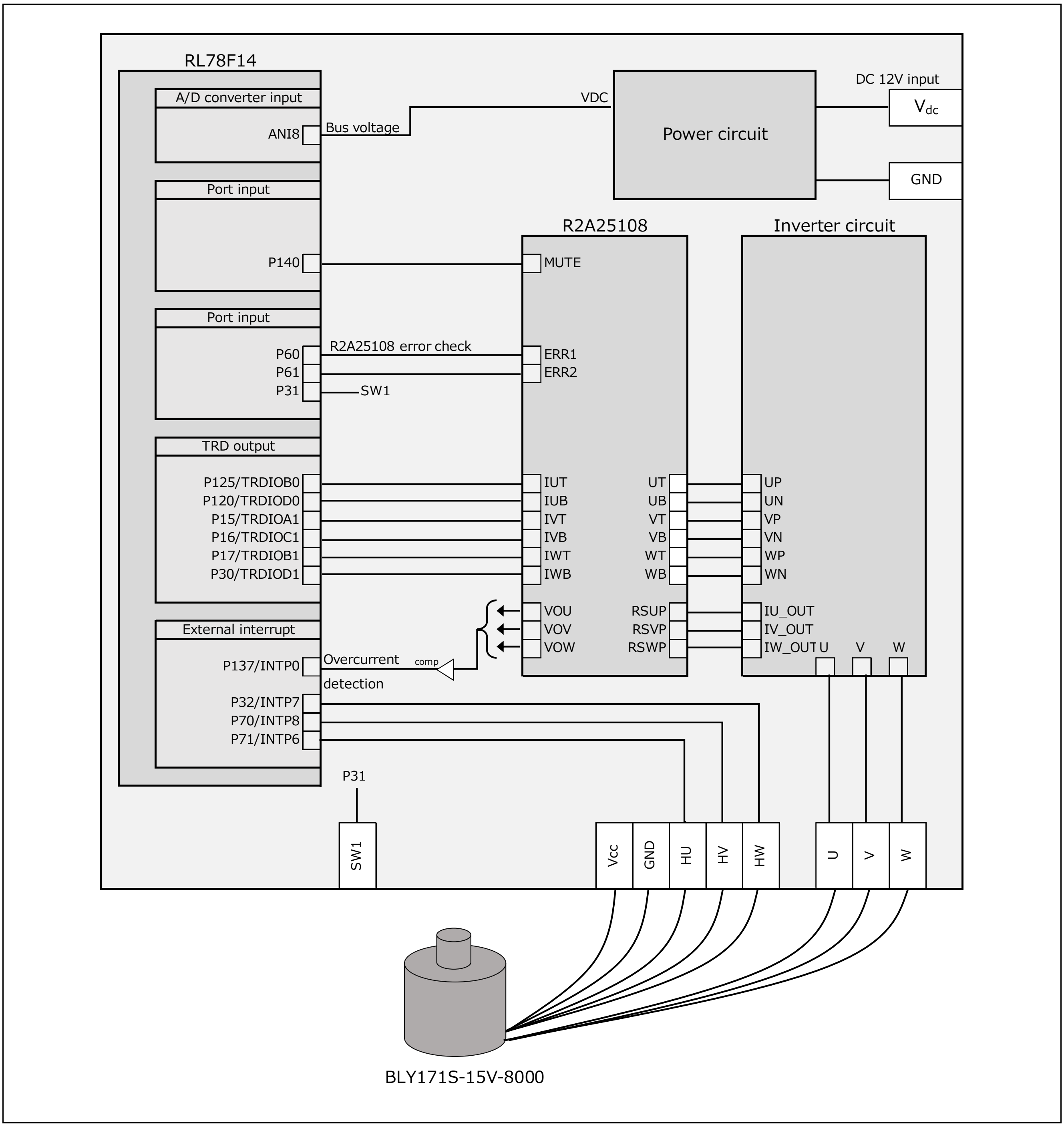
2. System overview

Overview of this system is explained below.

2.1 Hardware configuration

Hardware configuration is shown below.

Figure 2-1 Hardware configuration Diagram



2.2 Hardware specifications

2.2.1 Terminal interface

List of user interface of this system is given in Table 2-1.

Table 2-1 Terminal interface

Terminal name	Function
P71 / INTP6	Hall sensor input (HU)
P70 / INTP8	Hall sensor input (HV)
P32 / INTP7	Hall sensor input (HW)
P86 / ANI8	Bus voltage measurement
P125 / TRDIOB0	Complementary PWM output (U_p)
P120 / TRDIOD0	Complementary PWM output (U_n)
P15 / TRDIOA1	Complementary PWM output (V_p)
P17 / TRDIOB1	Complementary PWM output (W_p)
P16 / TRDIOC1	Complementary PWM output (V_n)
P30 / TRDIOD1	Complementary PWM output (W_n)
P60	ERR1 input
P61	ERR2 input
P140	MUTE output
P31	SW input

2.2.2 Peripheral functions

List of peripheral functions used in this system is given in Table 2-2.

Table 2-2 Peripheral functions List

Peripheral function	Usage
External interruption (INTP6, INTP7,INTP8)	- Hall sensor signal input (position detection) - Hall sensor read-out and external interruption (both edges)
Timer RD (TRD)	PWM output using complementary PWM mode (3positive phases, 3 negative phases)
Port (P60, P61)	Error detection (Over Voltage detection, low voltage detection, heating load short detection)
External interruption (INTP0)	Error detection (Over Current detection)
Port (P15, P16, P17, P30, P120, P125, P140)	- motor control signal with port output - MUTE terminal control signal output
Timer Array Unit (TAU)	- 1 [ms] interval timer - Free-run timer for speed measurement
A/D converter (ANI8)	Bus voltage measurement

2.3 Software specifications

2.3.1 Software file structure

Folders and files structure of the sample program is given below.

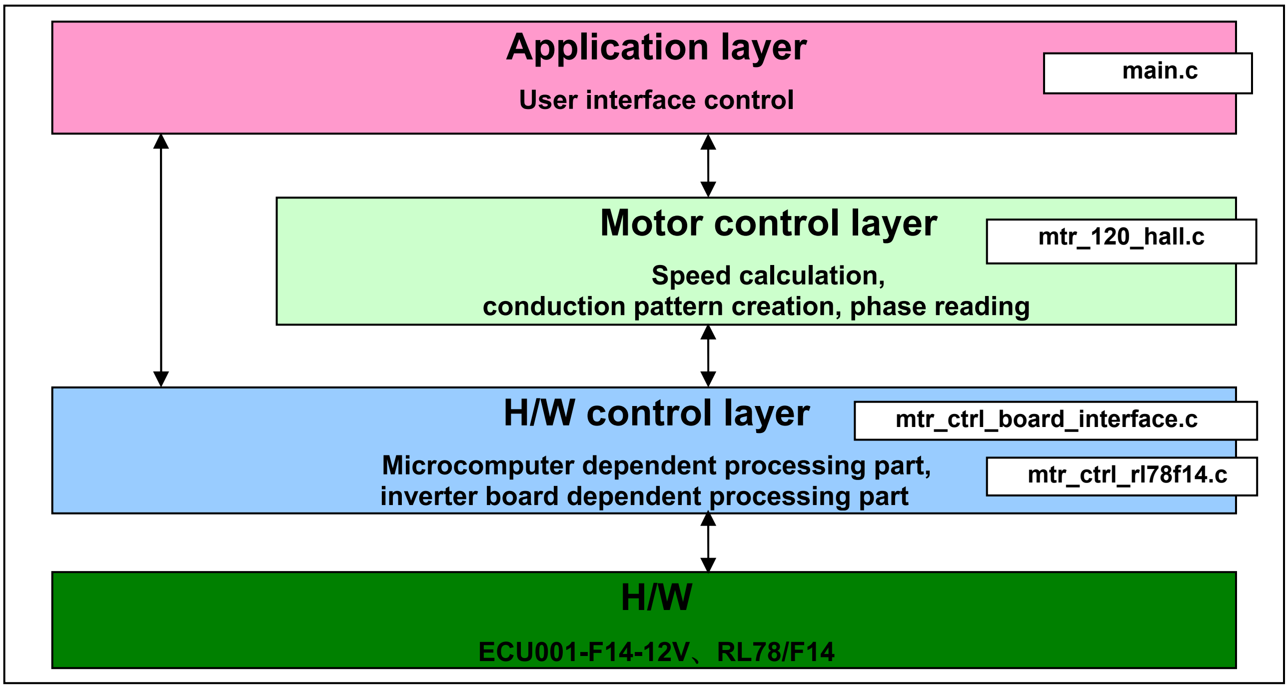
Table2-3 Folders and files structure of sample program

RL78F14 _120_HALL_INTERMIDI	inc	Iodefine.h	SFR definition file(CCRL)
		main.h	Main function, user interface control header
		mtr_120_hall.h	Hall sensor used 120-deg conducting control dependent part header
		mtr_common.h	Header for common definition
		mtr_ctrl_board_interface.h	Board dependent processing part header
		mtr_ctrl_rl78f14.h	RL78/F14 dependent processing part header
	src	main.c	Main function, user interface control
		mtr_120_hall.c	Hall sensor used 120-deg conducting control dependent part
		mtr_ctrl_board_interface.c	Board dependent processing part
		mtr_ctrl_rl78f14.c	RL78/F14 dependent processing part
		mtr_interrupt.c	Interruption handler
	asm	cstat.asm	Startup routine
		hwinit.asm	Hardware initialization
		stkinit.asm	Stack initialization

2.3.2 Modules structure

Module structure of the sample program is described below.

Figure 2-1 Module structure of sample program



2.4 Software specifications

Basic specifications of software of this system are given in Table 2-4.

Table2-4 Software basic specifications

Item	Content
Control method	120-deg conducting method
Motor rotation start / stop	- Start by SW1 push down short time. - Stop by SW1 push down long time or driver error detection.
Position detection of rotor magnetic pole	Position detection by hall sensor (every 60 degrees)
Carrier frequency (PWM)	20 [kHz]
Control cycle	Position detection by hall sensor (every 60 degrees). Determination of PWM duty setting and conduction pattern.
Rotation speed control range	800 [rpm] to 5000 [rpm] both CW / CCW. (8pole)
Rotation speed operation	Detects the edge of hall signal then calculates the rotation speed by timer counts for pi [rad] every pi / 3 [rad] hall signal detection. Uses the interval timer for measurement of edge intervals.
Speed control (Speed PI control)	Obtains the speed command value form speed command value setting function, and performs speed control by PI control (5 [ms] cycle).
Processing stop for protection	Disables the motor control signal output (six outputs) under any of the following 3 conditions: 1. Rotation speed exceeds 33000[rpm] (electrical angle). (Monitored for each 1 [ms]) 2. No hall sensor interruption generated for 20 [ms] while the motor is driving. 3. Detect error signal (err1, err2) from pre-driver

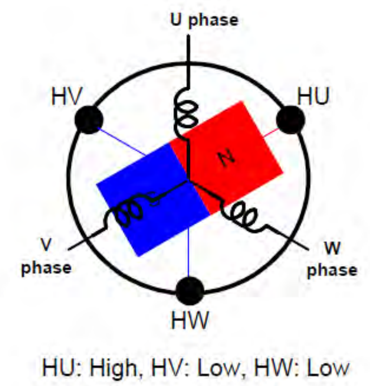
3. Motor control method

120 degrees conducting control and speed control of the BLDC motor with hall sensor, used in the sample program are explained here.

3.1 120 degrees conducting control of the BLDC motor with hall sensor

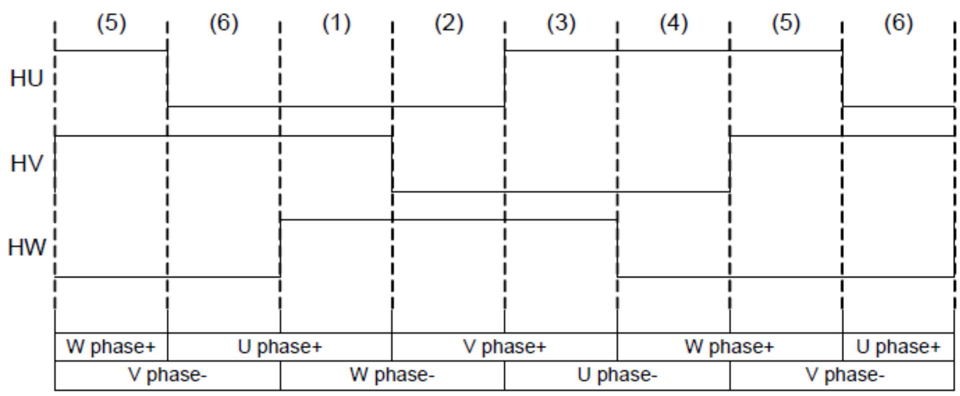
In this system, the hall sensor is used to detect the position of permanent magnet and signals from the hall IC (hall sensor signals) are input to the microcomputer as position information.

Figure 3-1 Example of hall sensor (HU, HV, HW) position and position signal



As shown if Figure 3-1, a hall sensor is allocated every 120 degrees and the respective hall sensor signals are switched depending on direction of rotating magnetic poles. Position information can be obtained every 60 degrees (six patterns for one cycle) by combining these three hall sensor signals.

Figure 3-2 Relation between hall sensor signals and conduction patterns (rotation direction: CCW)

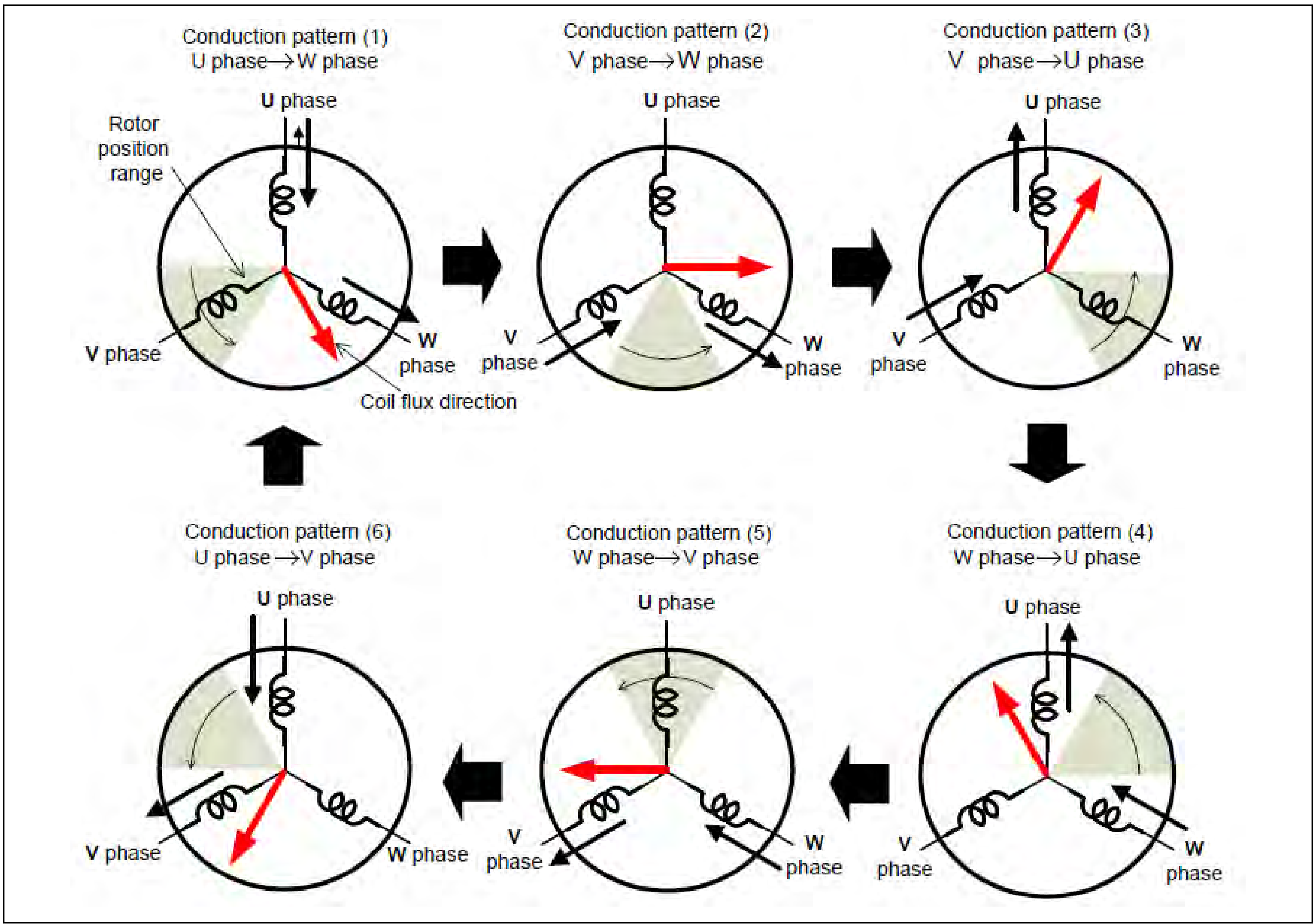


If the conduction patterns of each phase are changed in the switching timing of these hall sensor signals, as shown in Figure 3-2, rotating flux generated as shown in Figure 3-3. Then the rotor has the torque and rotates.

As conduction session of each switching element is 120 degrees, this control method is referred to as 120 degrees conducting control.

The relation between above mentioned six conduction patterns and rotor position ranges is shown in Figure 3-3.

Figure 3-3 six conduction patterns and rotor position ranges



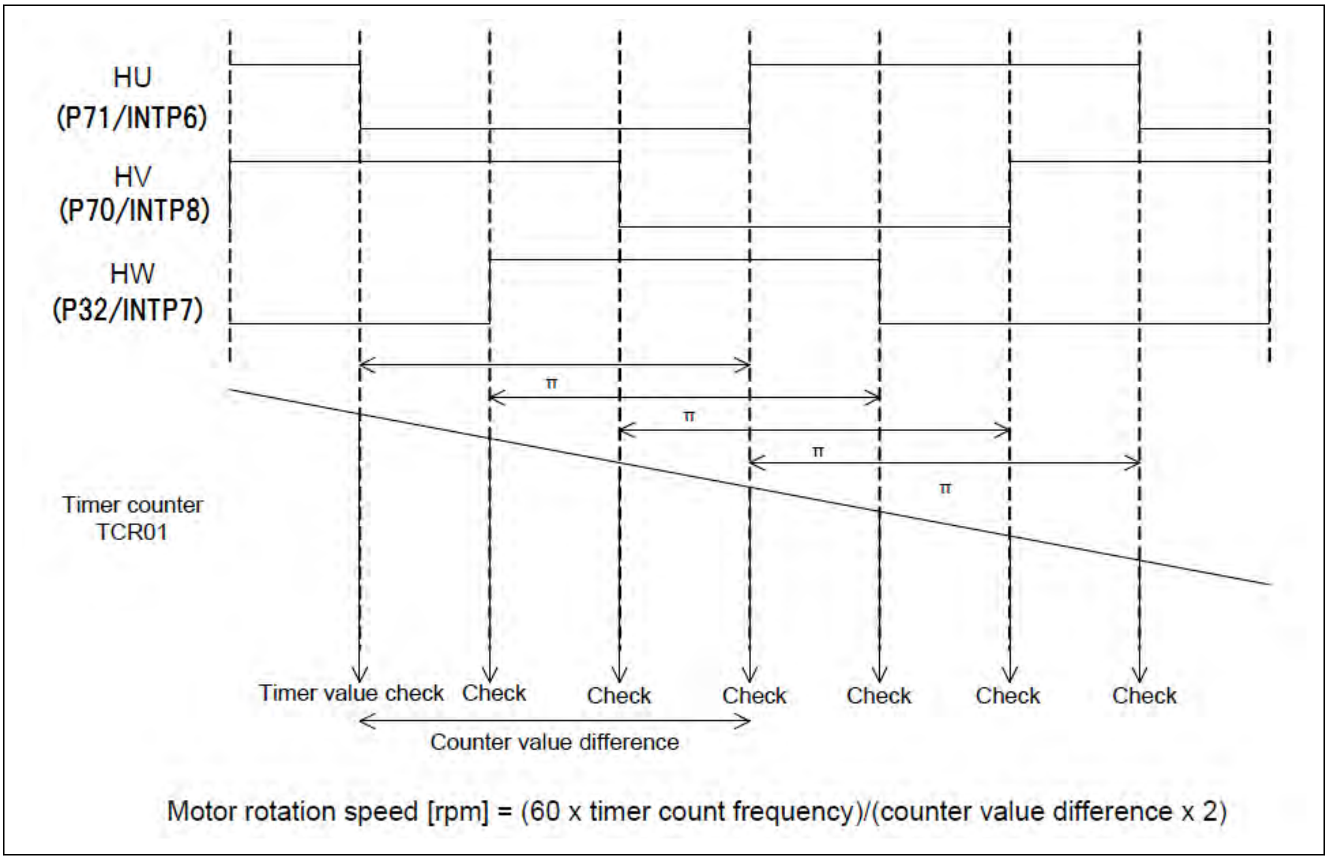
Supplements:

1. The relation between hall sensor signals and conduction patterns shown in Figure 3-3 is set to be suitable for this system. A different motor specification requires setting different conduction patterns appropriate to the system.
2. In the 120 degrees conduction control, only six types of conduction patterns are generated for one cycle and hence in principle, a torque ripple occurs without fail.

3.2 Speed control

In this system, the motor rotation speed is calculated from a difference of the present time value and the timer value pi [rad] before. Timer values are obtained through the external interruption routine by hall sensor signals while having the timer of channel 1 of timer array unit performed free running. This method is applicable even if three hall sensors are not placed at equal spaces.

Figure 3-4 Method of calculating motor rotation speed



This system uses PI control for speed control. A voltage command value at any (discrete) time ‘n’ is calculated by the following formula.

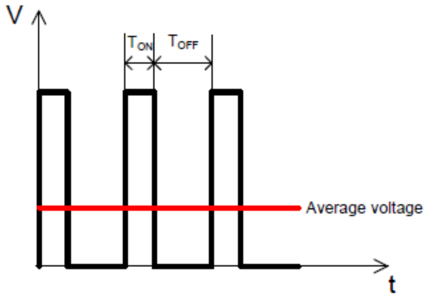
$$V[n] = V[n-1] + K_P \times (err[n] - err[n-1]) + K_I \times err[n]$$

V : Voltage err : Deviation of rotation speed command value and rotation speed calculation value
KP : Proportional gain KI : Integral gain

Also, PWM control is used for the output voltage control. 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

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



Modulation factor "m" is defined as follows.

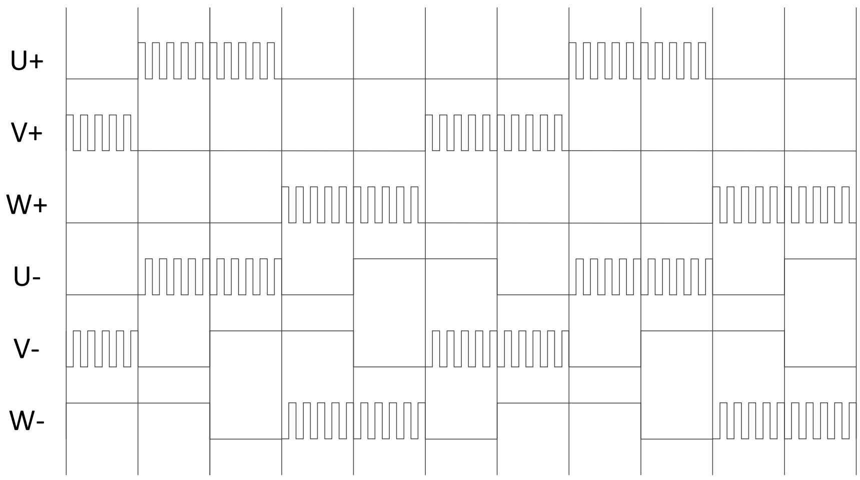
$$m = \frac{V}{E}$$

M: Modulation factor V: Target voltage err: Inverter bus voltage

This modulation factor is reflected in the setting value of resistor that determines the PWM duty.

In this system, complementary PWM chopping method (120 degrees) is adopted and thus output voltage and speed are controlled. An example of motor control signal output waveforms at the time of noncomplementary PWM chopping (120 degrees) is given in Figure 3-6.

Figure 3-6 Waveforms of complementary PWM chopping (120 degrees)



4. Description of peripheral functions used

Peripheral functions used in this system are explained.

Following peripheral functions are explained in this chapter.

- External interruption function
- A/D converter
- Timer Array Unit TAUS function
- Timer RD function

4.1 External interruption function

In this system, external interruptions are set as given in Table 4-1.

Table4-1 External interruption setting details

Interruption	Item	Content	Usage
INTP6, INTP7, INTP8	Valid edge	both edges	Edge detection of hall sensor signal
	Interruption priority level	2	
INTP0	Valid edge	Rising edge	Error detection (Over Current detection)
	Interruption priority level	0	

4.2 A/D converter function

A/D converter converts the analog input to digital value. The target microcontroller (RL78/F14), incorporates one circuit of 10bit A/D converter. Analog input of twelve channels can be converted to digital values by controlling the conversion channel.

In this system, the A/D converter is set as given in Table 4-2.

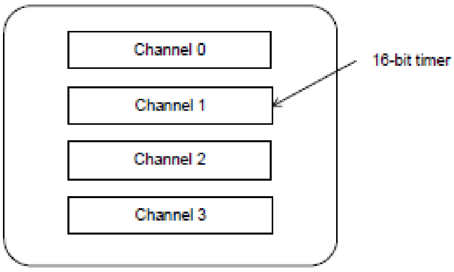
Table 4-2 A/D converter setting details

Channel	Item	Content	Usage
ANI8	Conversion time	3.563 [us]	Inverter bus voltage
	Channel selection mode	Select mode	
	Conversion operation mode	One-shot conversion mode	
	Conversion starting conditions	Software trigger	

4.3 Timer Array Unit TAUS function

The Timer Array Unit TAUS consists of four 16bit timers. Each 16-bit timer called 'Channel' and can be used as an independent timer as well as an advanced timer function by combining multiple channels.

Figure 4-1 Timer Array Unit



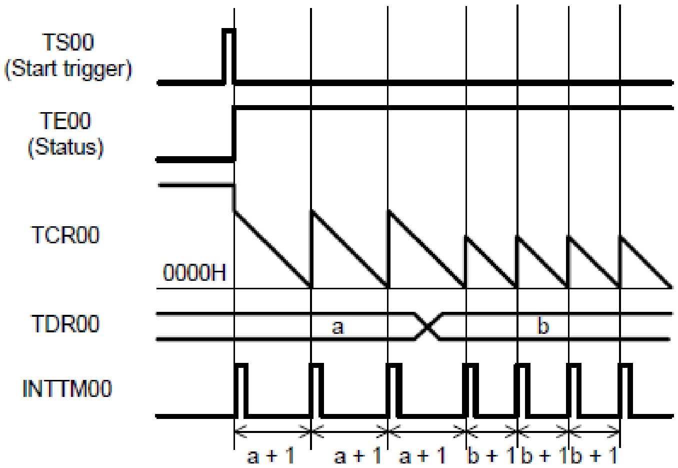
In this system, the Timer Array Unit is set as given in Table4-3.

Table4-3 Timer Array Unit Setting Details

Channel	Item	Content	Usage
Channel 0	Operation mode of timer	Interval timer	Timer for generating 1[ms]
	Source clock	CK00	
	Count clock frequency	24 [MHz]	
	Interruption cycle	1 [ms]	
	Setting value of Timer data resistor 0 (TDR00)	23999 (1[ms]/(1/24[MHz])) - 1)	
Channel 1	Operation mode of timer	Interval timer	Timer for speed calculation
	Source clock	CK01	
	Count clock frequency	125 [kHz]	
	Interruption cycle	524 [ms] (unused)	
	Setting value of Timer data resistor 0 (TDR00)	65535	

Also, basic timings of the interval timer are shown in Figure 4-2.

Figure 4-2 Example of basic timings of interval timer (Example of Channel 0)



4.4 Timer RD function

Timer RD has two 16-bit timers (timer RD0 and timer RD1).

Following four modes are provided in timer RD.

- Timer mode
- Reset synchronous PWM mode
- Complementary PWM mode
- PWM3 mode

In this system, the timer RD is set as given in Table 4-4.

Table 4-4 Timer RD setting details

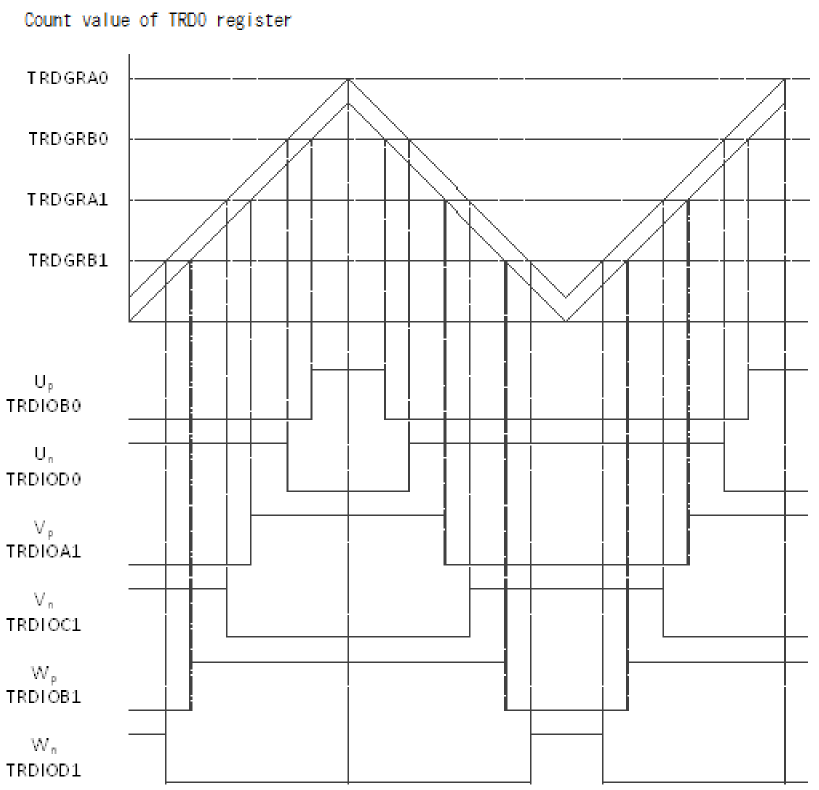
Timer used	Item	Content	Usage
Timer RD	Mode used	Complementary PWM mode	6 phase PWM output
	PWM cycle	50 [us]	
	Dead time	2.0 [us]	
	Count frequency	48 [MHz]	
	Output level	Initial output is “Low”, Active level is “High”	
	Buffer operation	Valid	
	Pulse output forced shutdown control	Valid (Output value at the time of shutdown: Hi-Z)	
	Output port	Refer to Figure 4-3	

Note:

1. In complementary PWM mode, the timer RD outputs a waveform by combining the counters and resistors of timer RD0 and timer RD1.

An example of PWM output waveform is shown in Figure 4-3.

Figure 4-3 Example of PWM output waveform in complementary PWM mode



4.5 Calculation of PWM duty setting using modulation factor

This part summarizes how to set duty in complementary PWM mode.

As first, calculate positive phase active level width by using modulation rate in section 3.2. Next, calculate setting value of TRDGRB0, TRDGRA1, and TRDGRB1 registers that output positive phase active level width.

Positive phase active level width = PWM cycle * modulation rate

TRDGRB0 = TRDGRA1 = TRDGRB1 = TRDGRA1 - TRD0 + 1 - Positive phase active level width

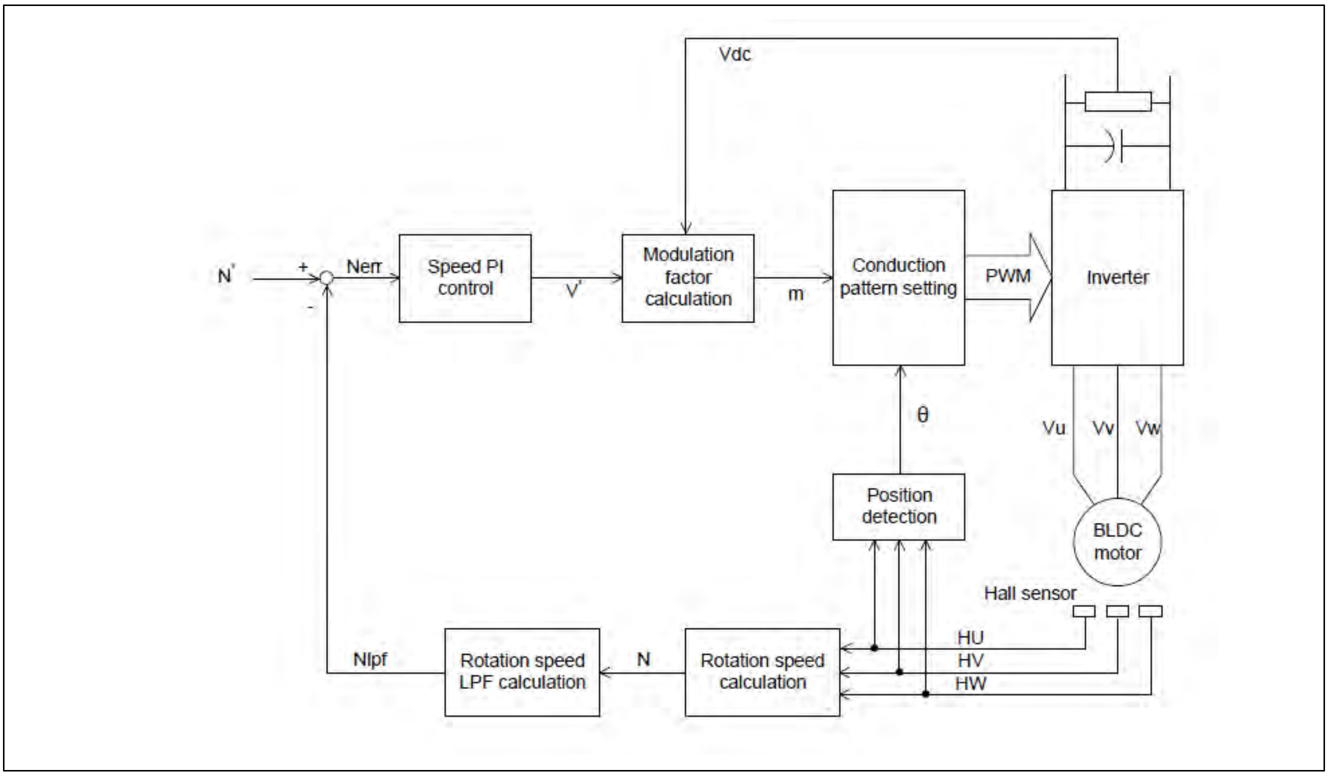
5. Description of control program

Control program of this system (120 degrees conducting control of brushless DC motor with hall sensor) is explained here.

5.1 Control block diagram

In the sample program a motor is driven by open loop control for the 100 [ms] after activation (operation mode during this period is called at BOOT mode). After that, control is performed according to the follow black diagram.

Figure 5-1 Control block diagram



Name	Meaning
N	Rotation speed
Nlpt	Rotation speed after LPF
N*	Rotation speed command value
Nerr	Rotation speed deviation
V*	Voltage command value
Vdc	Inverter bus voltage
m	Modulation factor
PWM	PWM output signal
Vu, Vv, Vw	Phase voltage
HU, HV, HW	Hall sensor signal
θ	Rotor position

Function is given below.

(1) Position detection of permanent magnet

Permanent magnet position is detected by reading the port values within the hall sensor interruption function (HU, HV, and HW. Both edge). The hall sensor signal that input to microcontroller is supposed to digital signal.

(2) Rotation speed calculation

Rotation speed is calculated from the timer counter (TCR01) within the hall sensor interruption function. The rotation speed calculation value is used in speed control.

(3) Speed control

Speed control is using PI control. The output value of speed PI control is set as a voltage command value.

(4) Processing stop for protection

Processing stop for protection prevents the motor or inverter from breakage due to overcurrent or overvoltage.

5.2 Contents of control

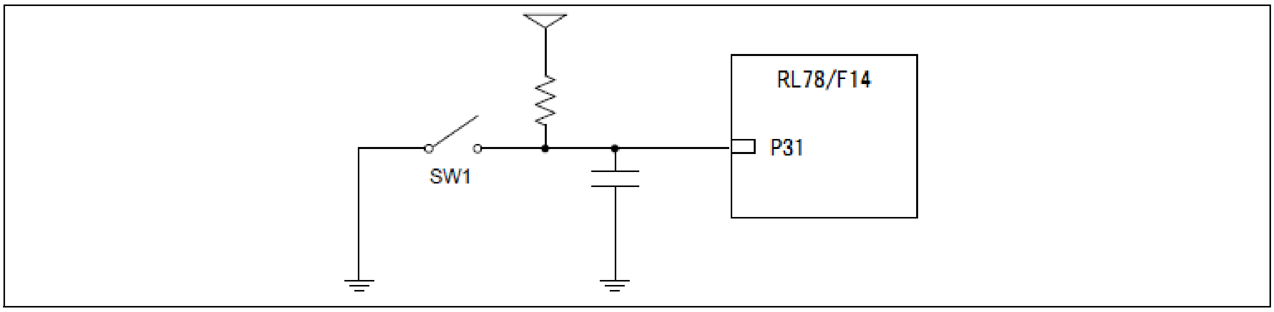
5.2.1 Motor start/stop

Starting and stopping the motor are controlled by input from SW1.

A general-purpose port (P31) is assigned to SW1. The sample program reads the P31 port within the main loop. When SW1 is pushed short time, it is judged that the start switch is on. On the other hand, when SW1 is pushed long time, the program determined to stop the motor.

When push SW1 while the motor is start, a rotation speed command value is changed by software.

Figure 5-1 Conceptual diagram of start switch external circuit



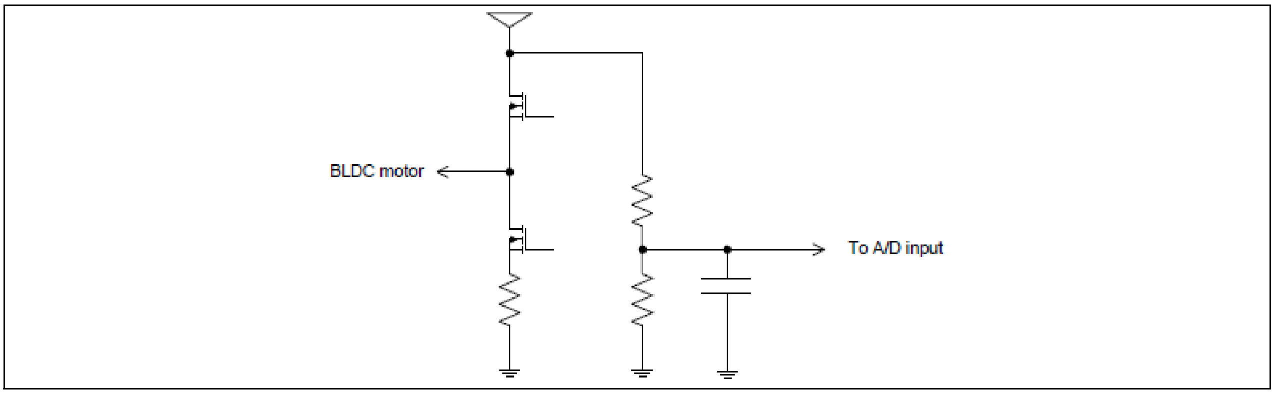
5.2.2 Inverter bus voltage

Inverter bus voltage is measured as given in Table . It is use for modulation factor calculation san overvoltage detection. (When an overvoltage is detected, PWM is stopped.)

Table 5-1 Inverter voltage conversion ratio

Item	Conversion ratio (Inverter voltage Vdc: A/D conversion value)	Channel
Inverter bus voltage	0 [V] to 26 [V] : 000H to 03FFH	ANI8

Figure 5-2 Conceptual diagram of inverter voltage measurement external circuit



5.2.3 Rotation speed operations

The rotation speed is calculated by using the external interruption by hall sensor signal and free-run timer (TAUS channel 01). A counter value of free-run timer is obtained through hall sensor interruption routine, and a difference of the present timer value and the timer value pi [rad] before is calculated. Based on the difference, speed is calculated by the following formula.

Rotation speed (N) = (60 * 125 [kHz]) / {(counter value pi [rad] before－current counter value) * 2}

Notes:

1. 125 [kHz] = (count clock frequency of free-run timer)
2. (* 2) is done as the period for obtaining the counter value is pi [rad]

In this sample program, LPF (migration average) processing is performed for the speed calculation result before speed PI control.

5.2.4 Speed PI control

In this sample program, speed PI control is performed on a 5 [ms] cycle, to avoid the multiple executions of PI control during hall sensor interruption. The voltage command value (V*) is created as given below.

Proportional (P) term: K_P * (current rotation speed deviation – last rotation speed deviation)
Integral (I) term: K_I * (current rotation speed deviation)
Voltage command value (V*) = previous voltage command value + proportional term + integral term

Notes:

1. Proportional gain (K_P): 0.0001
2. Integral gain (K_I): 0.00001

Values of K_P and K_I depend on the used system.

For details of PI control, refer to specialized books.

5.2.5 System protection function

This control program has the following 5 types of error status and enables emergency stop functions in case of occurrence of respective error.

- Pre-driver error

The error signal (ERR1 and ERR2) of over voltage error, under voltage error and short detection error from pre-driver is monitored with general ports, CPU performs emergency stop.

The ERR1 use P60, the ERR2 use P61. A kind of external notification errors and combination of terminals are shown in Table 5-2.

Table 5-2 A kind of external notification errors and combination of terminals

A kind of errors	ERR1 (P60)	ERR2 (P61)
Over voltage detection	Low	High
Under voltage detection	Low	Low
short detection	High	Low
No Error	High	High

- Over current error

An emergency stop signal (overcurrent detection) from hardware forces the program to execute high impedance output to PWM the output port (emergency stop without involving CPU). The INTP0 port is used.

- Rotation speed abnormality error

The rotation speed calculation value is monitored with 1 [ms] interval. When an error value is detected in rotation speed values (in a case of value over 33000 [rpm] (electrical angle)), CPU performs emergency stop.

- Timeout error

When no hall sensor interruption occurs for a certain period (20 [ms]), CPU performs emergency stop.

- Hall sensor signal pattern error

Hall sensor signal patterns for each hall sensor interruption processing are monitored. When an error pattern is detected, CPU performs emergency stop.

5.3 System resources

5.3.1 Interruption

List of interruptions used in this control program is given here.

Table5-3 Interruption resources

Interruption	Interruption handler	Interruption occurrence condition	Main function
Carrier synchronous (INTTRD0)	void mtr_carrier_interrupt(void)	50 [us](20 [kHz])	- Clear counter of motor stop
Interval timer interruption (INTTM00)	void mtr_tau0_interrupt(void)	1 [ms](1 [kHz])	- Speed PI control - Error monitoring - Control start time measurement
Hall sensor interruption (INTP6, INTP7, INTP8)	void mtr_hall_interrupt(void)	Hall sensor signal edge detection	- Rotation speed calculation - Clearing motor stop determination counter value - Conduction pattern setting
Over current detection interruption (INTP0)	void mtr_over_current_interrupt(void)	Over current detection	- Over current protection

5.3.2 Port function

List of port functions used in this control program is given below.

Table5-4 Port functions

Input / output	Port number	Function	Remark
Input	INTP6/P71	Hall sensor signal interruption input (HU)	Perform both edge detection
	INTP8/P70	Hall sensor signal interruption input (HV)	
	INTP7/P32	Hall sensor signal interruption input (HW)	
	P31	SW input	Motor On/Off
	P60	ERR1 port input	Pre-Driver Error
	P61	ERR2 port input	
	INTP0/P137	Over Current input	
Output	P42	MUTE port control signal output	Pre-Driver control
	P125	U phase upper arm motor control signal port output (Up)	Logic setting is 'High' active.
	P120	U phase lower arm motor control signal port output (Un)	
	P15	V phase upper arm motor control signal port output (Vp)	
	P16	V phase lower arm motor control signal port output (Vn)	
	P17	W phase upper arm motor control signal port output (Wp)	
	P30	W phase lower arm motor control signal port output (Wn)	

5.3.3 PWM output part

List of PWM output used in this control program is below.

Table5-5 PWM signal

Input / output	Output port	Function	Remark
Output	TRDIOB0	U phase upper arm motor control signal PWM output (Up)	Logical setting is 'High' active.
	TRDIOD0	U phase lower arm motor control signal PWM output (Un)	
	TRDIOA1	V phase upper arm motor control signal PWM output (Vp)	
	TRDIOC1	V phase lower arm motor control signal PWM output (Vn)	
	TRDIOB1	W phase upper arm motor control signal PWM output (Wp)	
	TRDIOD1	W phase lower arm motor control signal PWM output (Wn)	

5.3.4 A/D converter input signal and used channels

List of used channels of A/D converter used in this control program is given below.

Table 5-6 A/D converter settings

Channel	Measurement signal	Range of setting value	Remark
ANI8	Inverter bus voltage	26 [V] / 5 [V]	Used in modulation factor calculation, over voltage protection

5.4 Function specification

Multiple control functions are used in this control program. Lists of control functions are given below.
For detailed processing, refer to flowcharts or source file.

Table 5-7 List of control functions (1/3)

File name	Function overview	Processing overview
main.c	main() Input: None Output: None	- Hardware initialization function call - User interface initialization function call - Main processing used variable initialization function call - status transition and event execution function call - Main processing -> Main processing execution function call -> Watchdog timer clear function call
	ctrl_ui() Input: None Output: None	- Motor status change - Determination of rotation speed command value and rotation direction
	software_init() Input: None Output: None	Initialization of variables used in the main processing
	check_sw() Input: sw_mode Output: sw_mode	Obtaining the SW pushed status.
	change_ref_speed_in_stages() Input: None Output: None	Change reference speed in stages.
mtr_ctrl_board_interface.c	get_sw1() Input: None Output: (uint8) tmp_port / level of SW1	Obtaining the status of SW1
mtr_interrupt.c	mtr_hall_interrupt() Input: None Output: None	Output pattern determination function call
	Mtr_over_current_interrupt() Input: None Output: None	･Event setting ･Error state
	mtr_tau0_interrupt() Input: None Output: None	- Error check Function call - Calling speed PI control function every 5 [ms] - Open loop starting control
	mtr_carrier_interrupt() Input: None Output: None	Compare match flag (IFMA) clear function call

Table 5-7 List of control functions (2/3)

File name	Function overview	Processing overview
mtr_120_hall.c	R_MTR_InitSequence() Input: None Output: None	Initialization for variables to use for sequence control
	R_MTR_ExecEvent() Input: (uint8)u1_event / Occurred event Output: None	- Changing the status - Calling an appropriate processing execution function for the occurred event
	mtr_act_run() Input: (uint8)u1_state / Motor status Output: (uint8)u1_state /Motor status	- Variable initialization function call upon motor startup - Motor control startup function call - Output pattern determination function call
	mtr_act_stop() Input: (uint8)u1_state / Motor status Output: (uint8)u1_state /Motor status	Motor control stop function call
	mtr_act_none() Input: (uint8)u1_state / Motor status Output: (uint8)u1_state /Motor status	No processing is performed
	mtr_act_reset() Input: (uint8)u1_state / Motor status Output: (uint8)u1_state /Motor status	- Global variable initialization - Wait motor stop
	mtr_act_error() Input: (uint8)u1_state / Motor status Output: (uint8)u1_state /Motor status	Motor control stop function call
	mtr_pattern_set() Input: (uint8)u1_state / Motor status Output: (uint8)u1_state /Motor status	- Speed measurement function call - Obtaining hall sensor patterns - Conduction pattern determination - Motor control signal creation function call
	mtr_speed_calc() Input: None Output: None	Speed measurement calculation processing
	mtr_start_init() Input: None Output: None	Initializing only the variables required for motor startup
	mtr_pi_ctrl_speed() Input: None Output: None	Speed PI control
	R_MTR_SetSpeed() Input: (int16)ref_speed / Rotation speed command value Output: None	Rotation speed command value setting
	R_MTR_SetDir() Input: (uint8)dir / Rotation direction command value Output: None	Rotation direction setting
	R_MTR_GetSpeed() Input: None Output: (int16)g_s2_rpm / Rotation speed calculation value	Obtaining the rotation speed calculation value (electrical angle)
	R_MTR_GetStatus() Input: None Output: (uint8)g_u1_mode_system / Motor status	Obtaining the motor status
	mtr_error_check() Input: None Output: None	Error monitoring and detection
	mtr_25108_err() Input: None Output: None	Error detection from pre-driver (overcurrent, overvoltage, short detection)

Table 5-7 List of control functions (3/3)

File name	Function overview	Processing overview
mtr_ctrl_rl78f14.c	R_MTR_InitHardware() Input: None Output: None	Calling Initialization functions
	R_MTR_InitClock() Input: None Output: None	Initializing clock
	R_MTR_InitIoPort() Input: None Output: None	Initializing I/O ports
	R_MTR_InitTAU() Input: None Output: None	Initializing timer array unit
	R_MTR_InitTRD() Input: None Output: None	Initializing timer RD
	R_MTR_InitADC() Input: None Output: None	Initializing A/D convertor
	R_MTR_InitExtInt() Input: None Output: None	Initializing external interrupts
	init_ui() Input: None Output: None	Initializing user usage peripheral functions
	mtr_ctrl_start() Input: None Output: None	- Enabling hall sensor interruption (INTP6, INTP7, INTP8) - Starting TAU0
	mtr_ctrl_stop() Input: None Output: None	- Disabling hall sensor interruption (INTP6, INTP7, INTP8) - Stopping timer RD - Stopping TAU0 - Changing the motor control output port to inactive status - Waiting motor stop
	mtr_ctrl_error() Input: None Output: None	- Stopping timer RD - Stopping TAU0 - Changing the motor control output port to inactive status
	mtr_change_pattern() Input: (uint8)pattern / Conduction pattern Output: None	- Setting output pattern - Changing the motor status when output pattern error occurs - Event processing selection function call
	mtr_get_adc() Input: (uint8)ad_ch / Conversion channel Output: (int16)s2_temp / A/D conversion result	Executing A/D conversion
	clear_wdt() Input: None Output: None	Clearing the watchdog timer
	mtr_clear_trd0_imfa() Input: None Output: None	Clearing the compare match flag (IMFA)

5.5 Variables list

Lists of variables used in this control program are given below. Note that local variables are not described.

Table 5-8 Variables list

Variable name	Type	Content	Remark
g_u1_cnt_speed_pi	uint8	Speed PI control interrupt interval counter	Speed PI control cycle 5[ms]
g_s2_pwm_duty	Int16	Timer RD compere resister setting	-
g_u2_cnt_openloop_mode	uint16	Open loop mode time counter	
g_u2_cnt_wait_stop	Uint16	Wait motor stop counter	Count number of motor stop time
g_u1_flg_wait_stop	uint8	Wait motor stop flag	0: Not waiting motor stop 1: Waiting motor stop
g_u2_run_mode	uint16	Operation mode management	1: Open loop mode 2: Normal operation mode
g_u1_error_status	uint8	Error status management	1: Overcurrent error 2: Overvoltage 3: Over speed error 4: Timeout error 5: Hall sensor pattern error 7: Under voltage error 8: Short error (0xff: Non-definition error)
g_u1_mode_system	uint8	State management	0: Stop mode 1: Run mode 2: Error mode

5.6 Macro definitions

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

Table 5-9 Macro definitions list (1/6)

File name	Macro name	Definition value	Remark
main.h	SW_ON	0	SW active level
	SW_OFF	1	SW inactive level
	SW_MODE_NONE	0	SW1 is not pushed
	SW_MODE_SHORT	1	SW1 pushed short time
	SW_MODE_LONG	2	SW1 pushed long time
	CHATTERING_CNT	100	Chattering removal
	PUSH_CNT	0xF000	Using for judge SW1 pushed long time
	PUSH_CNT_CLR	0x0000	Counter clear value
	SOFT_STOP_SPEED	0	0 speed [rpm] (mechanical angle)
	SOFT_MIN_SPEED	0	min speed [rpm] (mechanical angle)
	SOFT_MAX_SPEED	3000	max speed [rpm] (mechanical angle)
	SOFT_DIFF_SPEED	500	increment / decrement difference speed [rpm] (mechanical angle)
	REF_SPEED_DECEL	0	deceleration reference speed
	REF_SPEED_ACCEL	1	acceleration reference speed

Table 5-9 Macro definitions list (2/6)

File name	Macro name	Definition value	Remark
mtr_ctrl_rl78f14.h	MTR_PWM_TIMER_FREQ	48	Timer RD count frequency [MHz]
	MTR_TAU1_FREQ	93750	Timer Array Unit channel 1 count frequency [Hz]
	MTR_PORT_MODE_HALL_U	PM7.1	U phase hall sensor input port
	MTR_PORT_ MODE_HALL_V	PM7.0	V phase hall sensor input port
	MTR_PORT_ MODE_HALL_W	PM3.2	W phase hall sensor input port
	MTR_PORT_HALL_U	P7.1	U phase hall sensor input port
	MTR_PORT_HALL_V	P7.0	V phase hall sensor input port
	MTR_PORT_HALL_W	P3.2	W phase hall sensor input port
	MTR_PORT_PULLUP_HALL_U	PU7.1	U phase hall sensor input port
	MTR_PORT_PULLUP _HALL_V	PU7.0	V phase hall sensor input port
	MTR_PORT_PULLUP_HALL_W	PU3.2	W phase hall sensor input port
	MTR_PORT_MODE_CTL_HALL_U	PMC7.1	Port mode control
	MTR_PORT_MODE_CTL_HALL_V	PMC7.0	Port mode control
	MTR_PORT_MODE_UP	PM12.5	U phase (positive phase) port mode
	MTR_PORT_MODE _UN	PM12.0	U phase (negative phase) port mode
	MTR_PORT_MODE _VP	PM1.5	V phase (positive phase) port mode
	MTR_PORT_MODE _VN	PM1.6	V phase (negative phase) port mode
	MTR_PORT_MODE _WP	PM1.7	W phase (positive phase) port mode
	MTR_PORT_MODE _WN	PM3.0	W phase (negative phase) port mode
	MTR_PORT_MODE_CTL_UP	PMC12.5	Port mode control
	MTR_PORT_MODE_CTL_UN	PMC12.0	Port mode control
	MTR_PORT_UP	P12.5	U phase (positive phase) output port
	MTR_PORT_UN	P12.0	U phase (negative phase) output port
	MTR_PORT_VP	P1.5	V phase (positive phase) output port
	MTR_PORT_VN	P1.6	V phase (negative phase) output port
	MTR_PORT_WP	P1.7	W phase (positive phase) output port
	MTR_PORT_WN	P3.0	W phase (negative phase) output port

Table 5-9 Macro definitions list (3/6)

File name	Macro name	Definition value	Remark
mtr_ctrl_rl78f14.h	MTR_TAU1_CNT	TCR01	Timer count register for speed measurement
	MTR_PORT_MODE_MUTE	PM14.0	MUTE output port
	MTR_PORT_MUTE	P14.0	MUTE output port
	MTR_PORT_MODE_ERR1	PM6.0	Signal Pre-Driver Error1
	MTR_PORT_MODE_ERR2	PM6.1	Signal Pre-Driver Error2
	MTR_PORT_ERR1	P6.0	Signal Pre-Driver Error1
	MTR_PORT_ERR2	P6.1	Signal Pre-Driver Error2
	MTR_PORT_PULLUP_ERR1	PU6.0	Signal Pre-Driver Error1
	MTR_PORT_PULLUP_ERR2	PU6.1	Signal Pre-Driver Error2
	MTR_PORT_MODE_SW1	PM3.1	LED1 output port
	MTR_PORT_SW1	P3.1	LED1 output port
	MTR_PORT_PULLUP_SW1	PU3.1	LED1 output port
	MTR_PORT_MODE_ADCCH_VDC	PM8.6	Input port mode of ADC (VDC)
	MTR_ADCCH_VDC	8	VDC voltage A/D conversion channel
	MTR_MAX_VDC	12	Limit of the voltage command value [V]
	MTR_VDC_RESOLUTION	26 / 1023	Inverter bus voltage resolution

Table 5-9 Macro definitions list (4/6)

File name	Macro name	Definition value	Remark
mtr_120_hall.h	MTR_CARRIER_FREQ	20.0f	PWM carrier frequency [kHz]
	MTR_DEADTIME_US	2.0f	Dead time value [us]
	MTR_START_DUTY	9.0f	PWM duty initial value [%]
	MTR_PATTERN_CW_U_V	5	CW hall sensor value
	MTR_PATTERN_CW_U_W	4	
	MTR_PATTERN_CW_V_W	6	
	MTR_PATTERN_CW_V_U	2	
	MTR_PATTERN_CW_W_U	3	
	MTR_PATTERN_CW_W_V	1	
	MTR_PATTERN_CCW_U_V	2	CCW hall sensor value
	MTR_PATTERN_CCW_W_V	6	
	MTR_PATTERN_CCW_W_U	4	
	MTR_PATTERN_CCW_V_U	5	
	MTR_PATTERN_CCW_V_W	1	
	MTR_PATTERN_CCW_U_W	3	
	MTR_SPEED_PI_DECIMATION	4	Speed PI control decimation count
	MTR_SPEED_PI_KP	0.0001f	Proportional term gain
	MTR_SPEED_PI_KI	0.00001f	Integral term gain
	MTR_AVG_OLD	0.3f	LPF previous value filter coefficient
	MTR_SPEED_LIMIT	33000	Over speed Threshold[rpm]
	MTR_OVERVOLTAGE_LIMIT	15	Over voltage limit[V]
	MTR_TIMEOUT_CNT	20	Undetected time = MTR_TIMEOUT_CNT * 1[ms]
	MTR_START_CNT	200	Control start time after startup [ms]

Table 5-9 Macro definitions list (5/6)

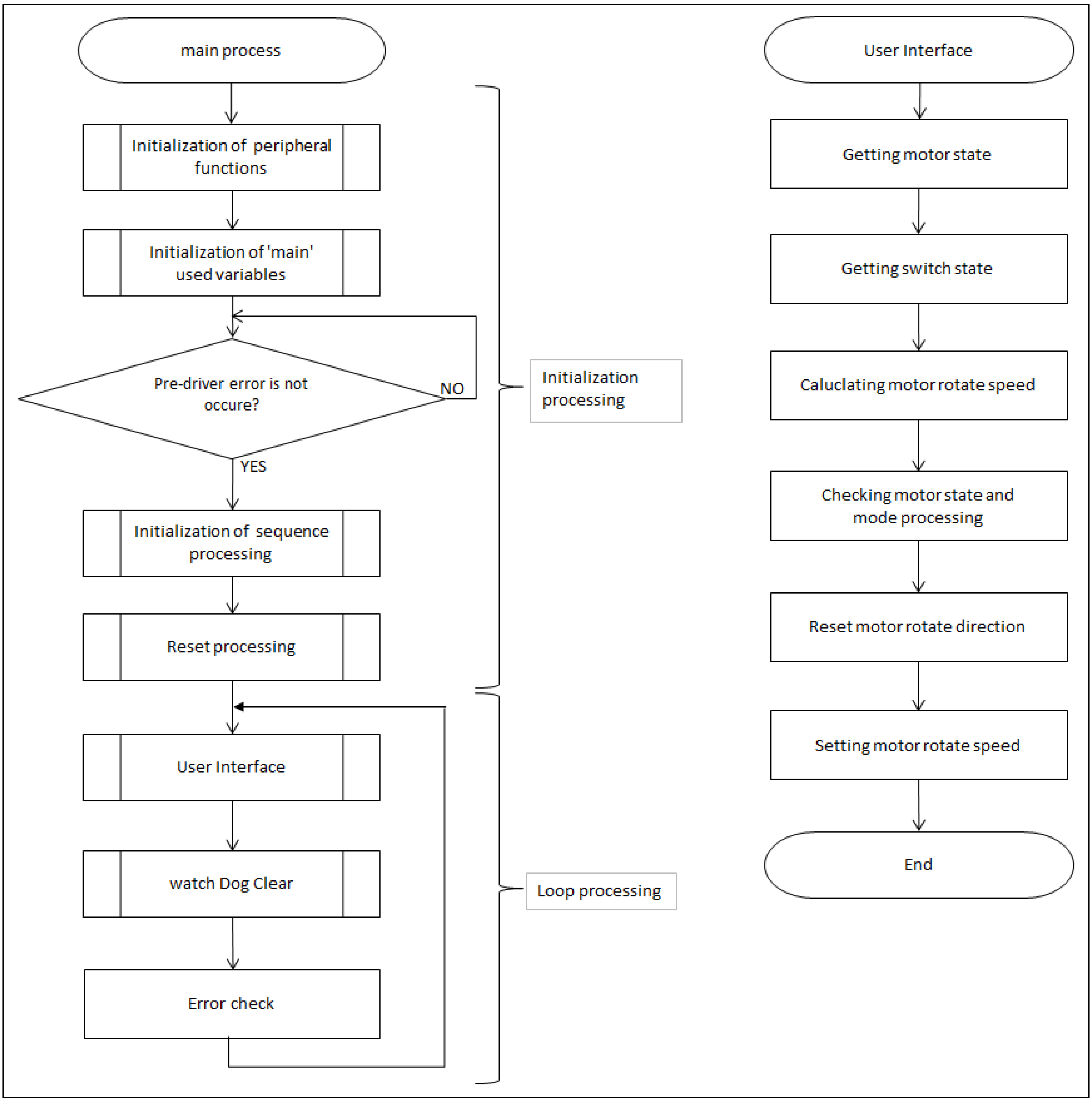
File name	Macro name	Definition value	Remark
mtr_120_hall.h	MTR_CARRIER_SET	(1000 / MTR_CARRIER_FREQ * MTR_PWM_TIMER_FREQ - 1)	Set value of carrier wave frequency
	MTR_PWM_DEAD_TIME	(MTR_PWM_TIMER_FREQ * MTR_DEADTIME_US)	Dead time (period)
	MTR_START_DUTY_SET	(((((MTR_CARRIER_SET + 1) / 100) * MTR_START_DUTY) / 2) + MTR_PWM_DEAD_TIME - 1)	Set value of default duty
	MTR_PWM_PERIOD	(((MTR_CARRIER_SET + 1) / 2) + MTR_PWM_DEAD_TIME)	Set value of PWM period (Set to TRDGA0 register)
	MTR_RATE_DUTY	(MTR_START_DUTY / 100)	Set value of initial voltage
	MTR_MAX_PWM_DUTY	((((MTR_CARRIER_SET + 1) / 100) * 95) - 1)	Max limit value of duty (95%)
	MTR_MIN_PWM_DUTY	((((MTR_CARRIER_SET + 1) / 100) * 5) - 1)	Min limit value of duty (5%)
	MTR_RPM_CALC_BASE	(60 * MTR_TAU1_FREQ / 2)	Constant for speed measurement: 60[sec] * TAU1 timer frequency[Hz] / 6(times)
	MTR_PATTERN_ERROR	0	
	MTR_U_PWM_VN_ON	1	
	MTR_V_PWM_WN_ON	2	
	MTR_W_PWM_UN_ON	3	
	MTR_U_PWM_WN_ON	4	
	MTR_V_PWM_UN_ON	5	
	MTR_W_PWM_VN_ON	6	
	MTR_CW	0	Rotation direction setting value: CW
	MTR_CCW	1	Rotation direction setting value: CCW
	MTR_AVG_NEW	(1 - MTR_AVG_OLD)	LPF new value filter coefficient
	MTR_OVERSIZE_LIMIT	115	Speed difference min.
	MTR_FLG_CLR	0	Constant for flag clear
	MTR_FLG_SET	1	Constant for flag set
	MTR_STOP_WAIT_CNT	200	Wait time[ms]
	MTR_POLE_PAIR	4	Pole pair number of motor

Table5-9 Macro definitions list (6/6)

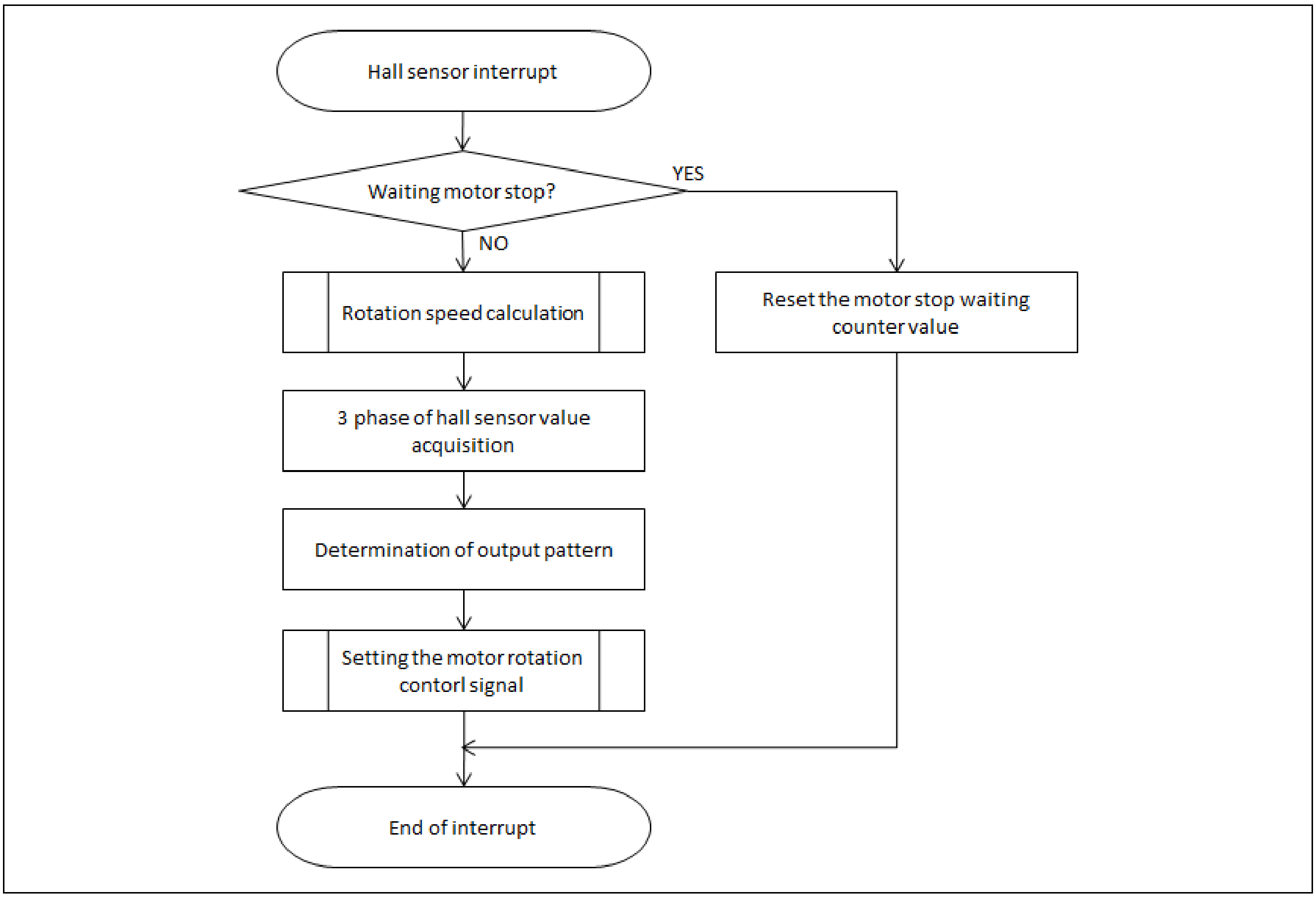
File name	Macro name	Definition value	Remark
mtr_120_hall.h	MTR_INITIAL_MODE	0x00	Initialization mode
	MTR_OPENLOOP_MODE	0x01	Open loop mode
	MTR_HALL_120_MODE	0x02	Normal operation mode
	MTR_OVER_CURRENT_ERROR	1	Overcurrent error
	MTR_OVER_VOLTAGE_ERROR	2	Overvoltage error
	MTR_OVER_SPEED_ERROR	3	Rotation speed abnormality error
	MTR_TIMEOUT_ERROR	4	Timeout error
	MTR_HALL_ERROR	5	Hall sensor pattern error
	MTR_OVER_VOLTAGE_ERROR	7	Under voltage error
	MTR_SHORT_ERROR	8	Short error
	MTR_UNKNOWN_ERROR	0xff	Undefined error
	MTR_MODE_STOP	0	stop status
	MTR_MODE_RUN	1	Rotating status
	MTR_MODE_ERROR	2	Error status
	MTR_SIZE_STATE	3	Status count
	MTR_EVENT_STOP	0	Motor stop event
	MTR_EVENT_RUN	1	Motor startup event
	MTR_EVENT_ERROR	2	Motor error event
	MTR_EVENT_RESET	3	Motor reset event
	MTR_SIZE_EVENT	4	Event count

5.7 Control flow (flow chart)

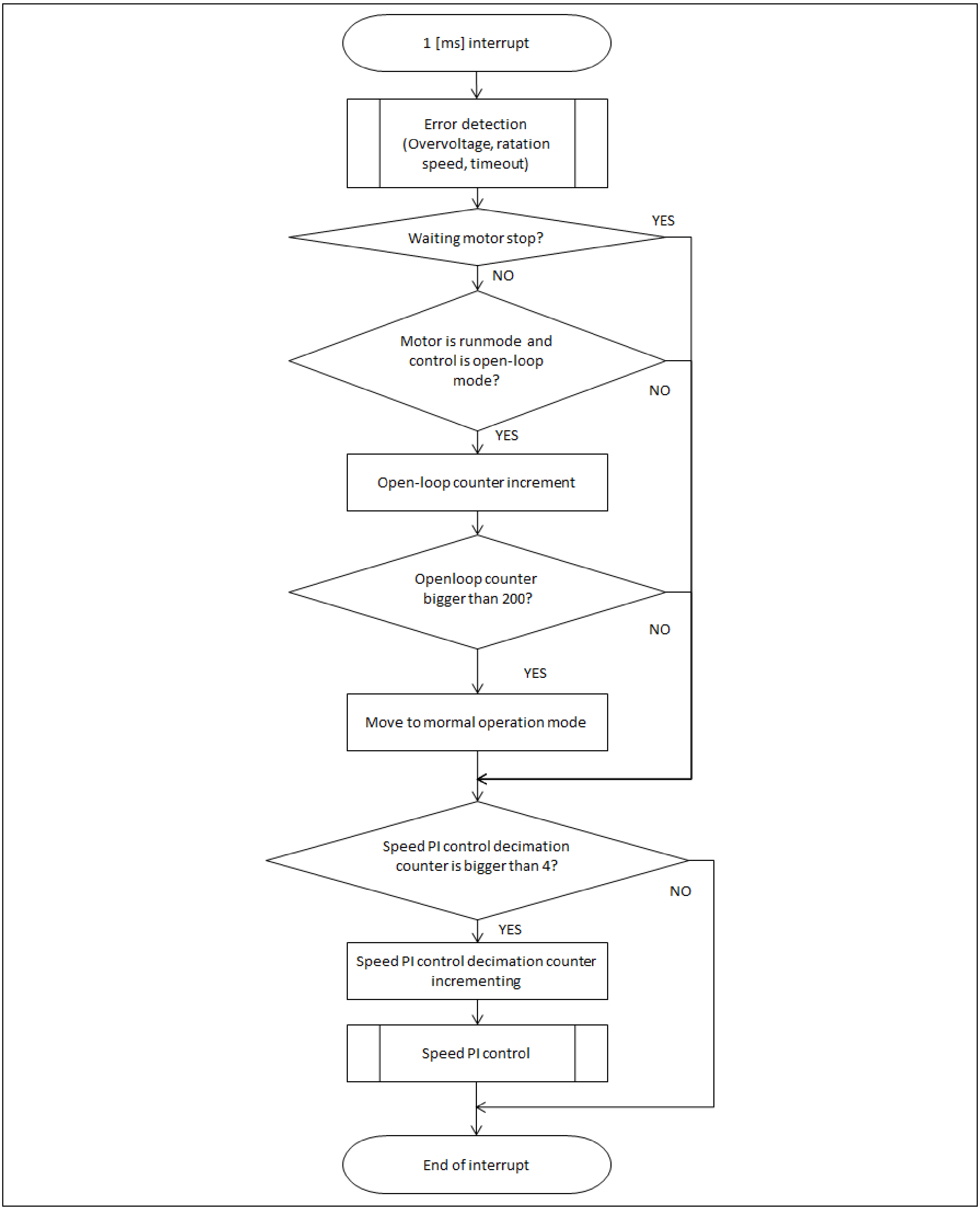
(1) Main process



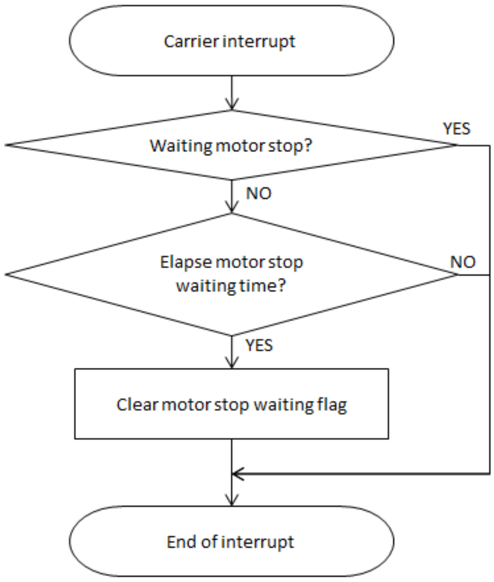
(2) Hall sensor interruption process



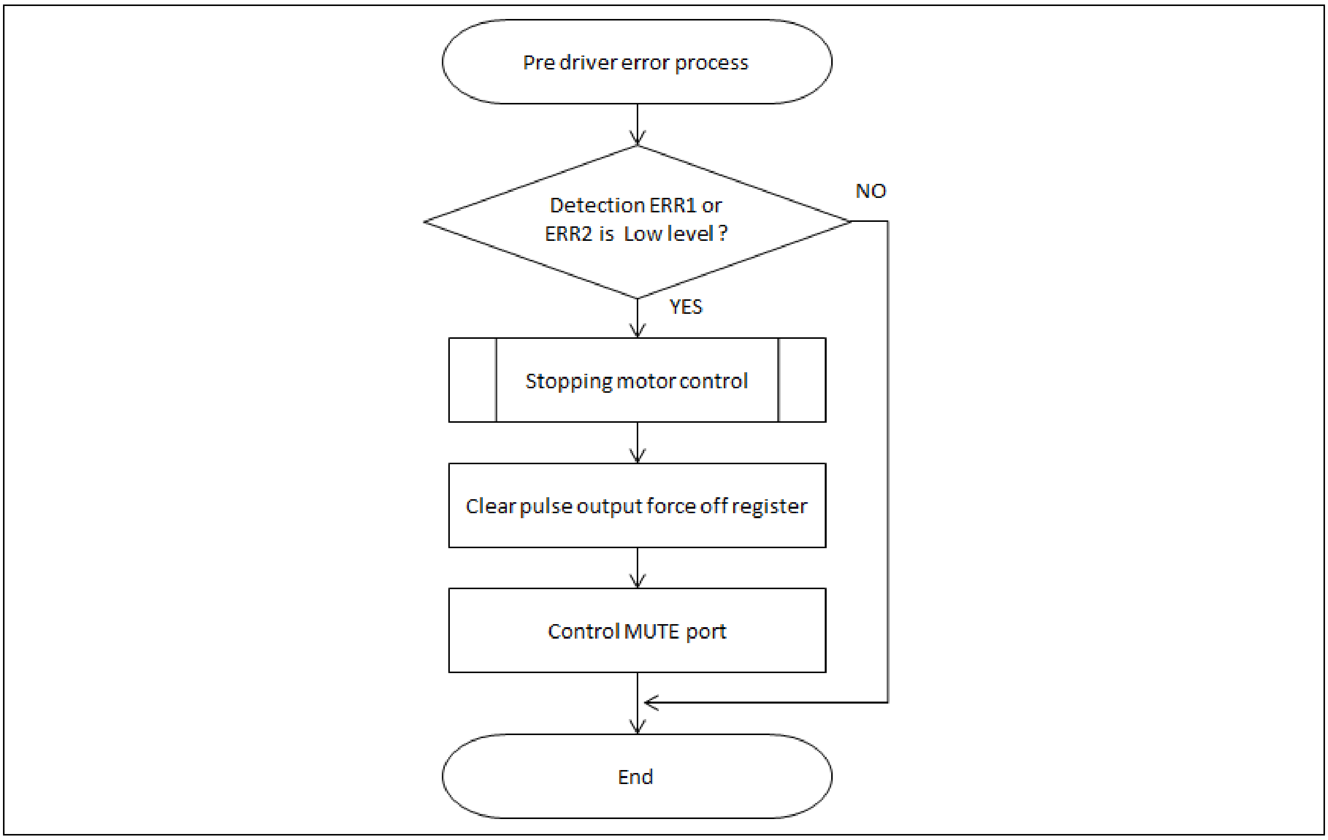
(3) 1 [ms] interruption process



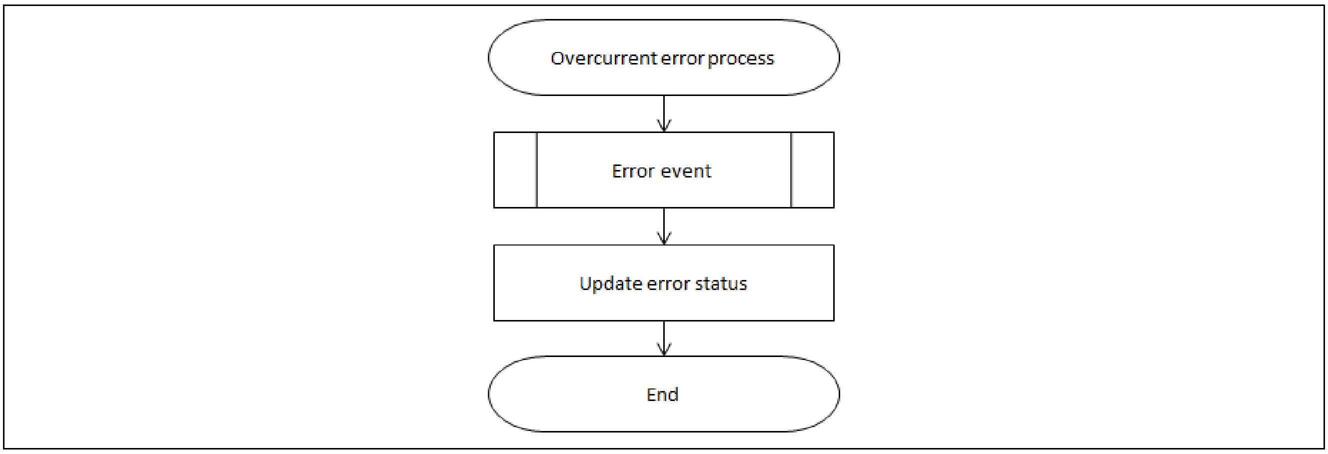
(4) Carrier cycle interruption process



(5) Pre-driver error process



(6) Overcurrent interrupts process



Website and Support

Renesas Electronics Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/contact/

Revision History

Rev.	Data	Description	
		Page	Summary
1.00	Mar.31.2017	-	First edition issued

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

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

Renesas Electronics America Inc.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

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

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL II Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

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

Colophon 6.0