RX62T

Position Control of PMSM with Encoder

R01AN0899EU0200
Rev. 2.00
Jan. 31, 2014

Introduction

This document presents RX62T position control with a permanent magnet synchronous motor, which has been implemented on RX62T evaluation kit with hall sensors and encoder.

The document describes hardware platform, methodology of position control, control block diagram, software structure, and flow chart of the position measurement and control.

The solution in application note has been implemented with RX62T evaluation kit and a 3-phase 8-pole 24V PMSM with a 1000 line single–ended encoder.

Target Device

RX62T

Contents

1. Overview

Position control plays an important role in various areas such as automation industry, semiconductor industry, etc. Permanent magnet synchronous motors (PMSM) are ideal for advanced position control systems for their potentials of high efficiency, high torque to current ratio, and low inertia, have been widely used in the industrial fields. Various approaches have been made to realize high performance motion control.

With successively improving reliability and performance of digital controllers, advances in Microprocessors (MCU) have greatly enhanced the potential of PMSM in servo position control applications. Digital control can be implemented by MCUs, which make it superior to analog based stepper control, since the controller is much more compact, reliable, and flexible. High performance of PMSM can be obtained by means of field oriented control, which is only realizable in a digital based system.

RX62T is a 32-bit high-performance microcontroller with a maximum operating frequency of 100MHz and 165 DMIPS and single precision floating-point unit (FPU), which is equipped with multifunction timers (MTU, GPT), high-speed 12-bit A/D converter and encoder signal capture for facilitating servo motion control.

In this application note, a RX62T floating point unit (FPU) based position motion control system is proposed. Position regulation is developed to provide both a trajectory generator and a PID controller, which ensures accurate position control and fast tracking. The trajectory generator provides position set-point commands. The position PID controller operates on the position error and outputs a current command. The current regulation with field oriented control is implemented to secure fast dynamic response.

Software developed is applicable to following devices and platforms.

- MCU: RX62T and RX62N
- Motor: three-phase permanent magnetic synchronous motors (PMSM)
- Platform: Renesas RX62T demo kit
- Control algorithm: Encoder based position control

2. System Hardware Setup and Structure

RX62T FPU based position control is implemented with Renesas RX62T evaluation kit and a three-phase PMSM with a1000 line single-ended encoder as shown in Figure 1.

RX62T evaluation kit is a single board inverter, based on the RX series microcontroller RX62T.

- A complete 3-phase inverter on-board with a low voltage motor
- 24Vexternal power supply to provide DC bus voltage, 15V and 5V power supply
- Power devices use Renesas low voltage MOSFETs
- Power rate up to 120watts
- Support 3 shunt and single shunt current measurement
- Easily jumper change from the external amplifiers to the internal PGA
- USB communication with the PC via a H8S2212 MCU
- User GUI to modify motor and control parameters, tune both speed and position control
- Connectors for hall sensors and encoder connections
- LCD display to monitor the operation status
- Support the standalone mode set by potentiometer and push buttons
- Support the second motor drive, signals and connector for another motor control power stage are available

The motor is a 24V 4 pair poles 3-phase permanent magnetic synchronous motor with

- 3 hall sensors
- 1000 line quadrature encoder

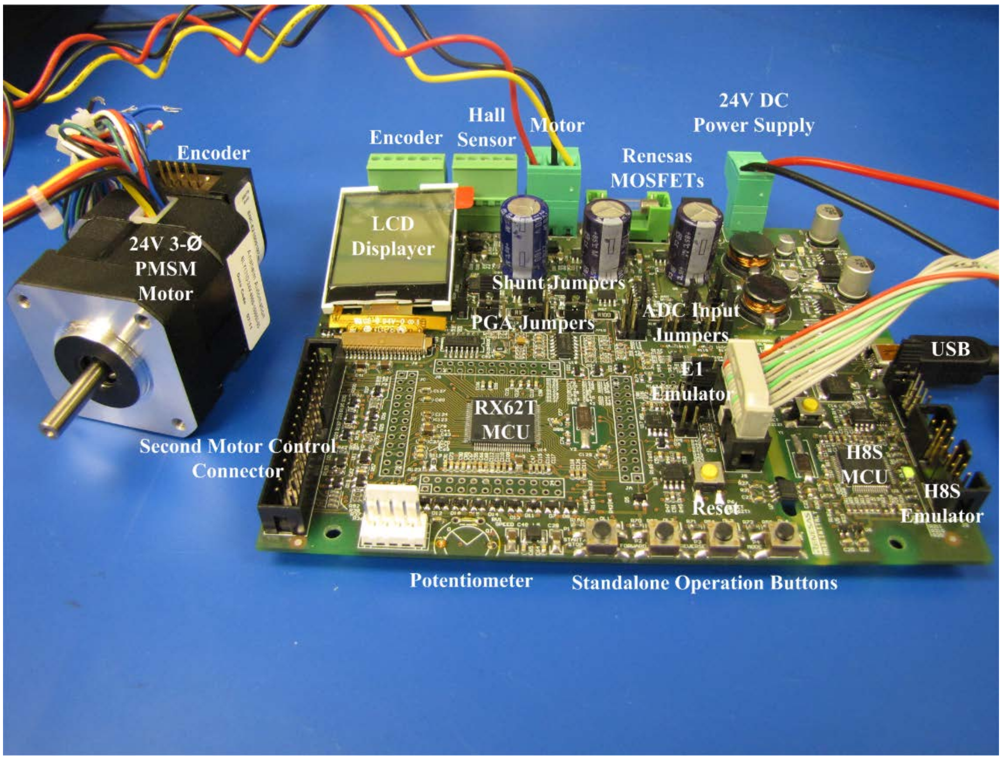

Figure 1 System hardware setup (motor and control platform)

3. Specification and Performance Data

The implementation of position control is based on Renesas evaluation kit and RX62T MCU, the main specification data are described as following:

- Input voltage: 24VDC
- Rated bus voltage: 24V
- Output voltage: 24VAC
- Rated output power: 120W
- PWM Switch frequency: 20KHz
- Control loop frequency: 10KHz
- Current measurement: 3 shunt resistors
- Position measurement: 1000 line quadrature encoder
- Implementation: FPU
- CPU bandwidth: 17%
- Used flash memory: 13.444Kbytes
- Used RAM: 1.725Kbytes
- Used stack : 336bytes

4. RX62T Encoder Capture Function

The RX62T is a 32-bit high-performance microcontroller with a maximum operating frequency of 100MHz and 165 DMIPS and single precision floating-point unit (FPU), which is equipped with multifunction timers (MTU, GPT), high-speed 12-bit A/D converter, and 10-bit A/D converter for facilitating motor control. Figure 2 shows the block diagram of a senorless vector control of PMSM based on the Renesas RX62T Microcontroller.

RX62T has a dedicate function for the encoder measurement as depicted in Figure 2. MTU3 timer external clock input TCLKA, TCLKB, TCLKC, and TCLKD can be used for two-phase encoder pulse inputs. When the MTU3 timer of Channels 1 and 2 are specified by the phase counting mode, an external encoder clock is selected as the counter input clock and TCNT operates as an up/down-counter. The phase difference between two external input clocks is detected and TCNT is incremented or decremented accordingly. The rotor position and speed can be measured by reading the TCNT counts.

The following summarizes the MTU3 function for the encoder pulse counting functionality:

- MTU Channel 1 & 2 support 2-phase pulse counting mode which is called “Phase Counting Mode”
- This function covers 4 modes
- At these modes, the counter works as up/down counter. And it is possible to detect the direction of counter with the flag.
- Up/down count by detecting phase difference between phase A and B of encoder on mode1 and mode 4
  - Mode 1: every rising edge & falling edge of both of encoder pulse
  - Mode 4: every rising edge & falling edge of phase B encoder pluses
- Up/down count by two pulse lines which indicate the direction, speed and position.
  - Mode 2: One pulse line and One direction
  - Mode 3: Two pulse lines for each direction
- MTU can detect automatically speed and position data as the pulse width & the pulse. The data of speed and position can be captured every periodic cycle.

In this application, the encoder pulse A and B are input to the TCLKA and TCLKB. The Z pulse is to IRQ0. For the second motor, the encoder pulse A and B are input to the TCLKC and TCLKD. The Z pulse is to IRQ3.

The host communication using the graphic user interface (GUI) is communicated with the RX62T MCU by the USB communication. It can display the motor operation status in the real time, tune the motor and control parameters, and drive the motor for both speed control and position control.

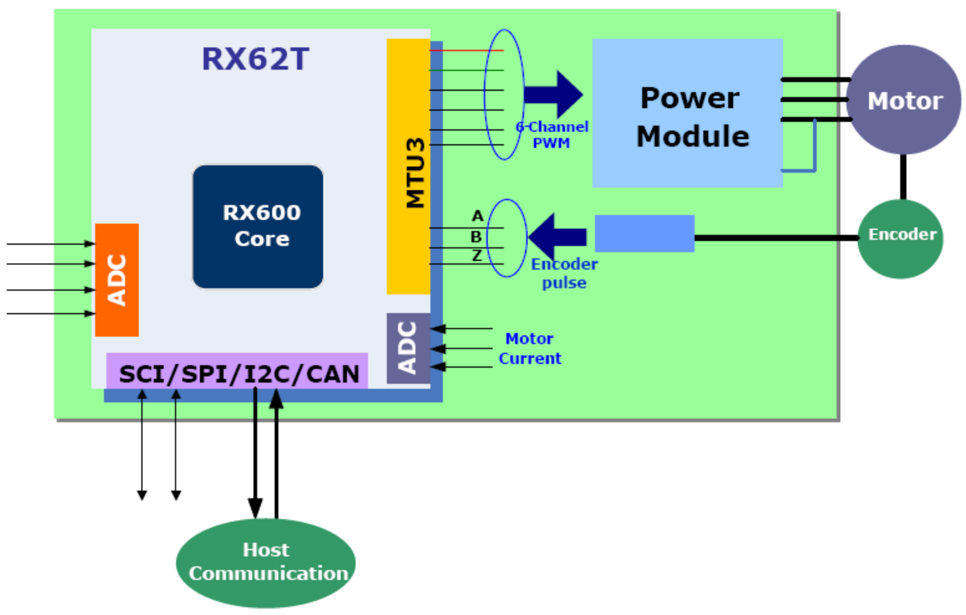

Figure 2 RX62T encoder capture functionality

Table 1 lists the timer register function for Channel 0 to 2 for the encoder capture. The timer MTU enable to automatically detect both the pulse width and the number of pulse of encoder every speed control loop period. It is not necessary of external wiring for any trigger signals. The encoder signals are directly input to Timer external clock; TCLKA and TCLKB as clock source of channel, and also, input command pulse to Timer external clock; TCLKC and TCLKD as clock source of Channel 2.

- Channel 1 counter is counted by every falling edge and rising edge of encoder pulse.
- Channel 0 is used for interval time to generate input capture trigger of Channel 1 and Channel 2, and interrupt of speed control loop.
- Channel 2 measures pulse command input.
- Channel 0 compare match (speed control loop period) can be selected as input capture trigger for Channel 1 internally.
- Channel 1 and Channel 2 external timer clock (encoder pulse or command pulse) can be selected as input capture trigger for Channel 0 internally.

Table 1 MTU timer registers function

Ch	Register/ Counter	Function	Data
0	TCNT0 (Ch0 Counter)	Free Running Timer by internal clock	Interval timer
	TGR0A	Input capture register (trigger by every edge of encoder pulse which is used for ch1 counter clock)	**Pulse width of encoder pulse**
	TGR0B	Output compare register (to create input capture trigger for Ch1 & Ch2)	**Speed Control Loop Period**
	TGR0C	Buffer register of TGRA0	Last data of pulse width
	TGR0D	Output compare register (to create input capture trigger for Ch1 & Ch2)	**Position Control Loop Period**
1	TCNT1 (Ch1 Counter)	Up/Down Counter by **encoder pulse**	Up/Down Counter shows position and speed
	TGR1A	Input capture register (trigger by Ch0/TGR0B compare match)	**The number of pulse of encoder at every speed control loop period**
	TGR1B	Input capture register (trigger by Ch0/TGR0D compare match)	**The number of pulse of encoder at every position control loop period**
2	TCNT2 (Ch2 Counter)	Up/Down Counter by **command input**	**Up/Down counter shows command of speed and position**
	TGR2A	Input capture register (trigger by Ch0/TGR0B compare match)	**The number of pulse of command at every speed control loop period**
	TGR2B	Input capture register (trigger by Ch0/TGR0D compare match)	**The number of pulse of command at every position control loop period**

Figure 3 shows how the MTU captures the encoder signals in phase counting mode. The Channel 1 is coupled with Channel 0 to input 2-phase encoder pulses of a servo motor in order to detect position or speed. Channel 1 is set to phase counting mode 1, and the encoder pulse A-phase and B-phase are input to MTCLKA and MTCLKB. In Channel 0, MTU3_0.TGRC compare match is specified as the TCNT clearing source and MTU3_0.TGRA and MTU3_0.TGRC are used for the compare match function and are set with the speed control cycle and position control cycle. MTU3_0.TGRB is used for input capture, with MTU3_0.TGRB and MTU3_0.TGRD operating in buffer mode. The Channel 1 counter input clock is designated as the MTU3_0.TGRB input capture source, and the widths of 2-phase encoder 4-multiplication pulses are detected. MTU3_1.TGRA and MTU3_1.TGRB for Channel 1 are designated for the input capture function and MTU3_0.TGRA and MTU3_0.TGRC compare matches in Channel 0 are selected as the input capture sources to store the up/down-counter values for the control cycles.

Therefore, the RX62T MTU itself can realize precise detection of the pulse width and the number of pulses, which are needed to estimate motor speed and position. It doesn’t need the load of the CPU hardly to detect those. Also the MTU is able to receive the pulse command as well.

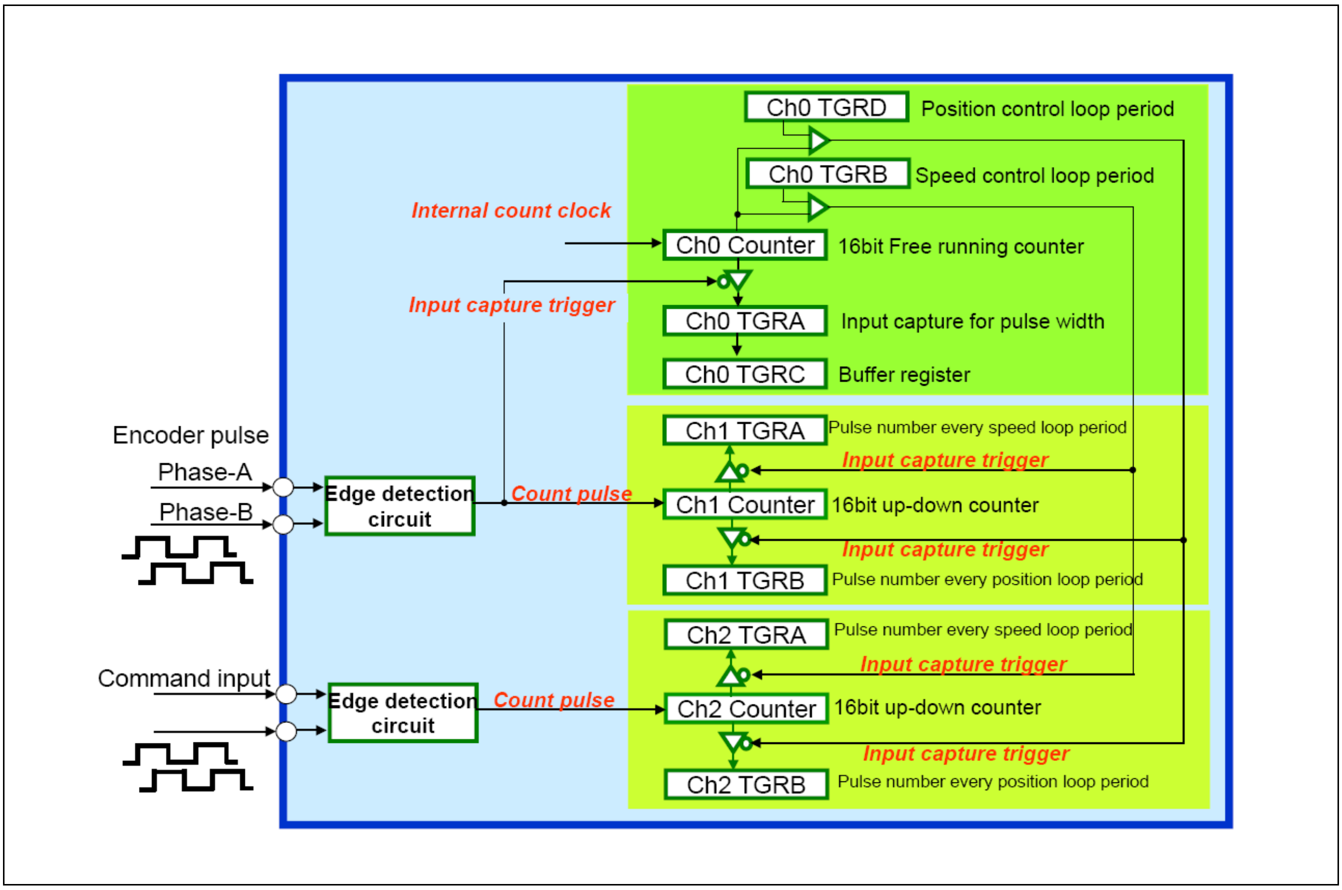

Figure 3 Encoder pulse capture in phase counting mode

5. Encoder Based Position and Speed Calculation

5.1 Position and Speed Measurement

A digital encoder outputs three pulse trains: A, B and Z, as shown in Figure 4. These pulses are fed into a timer unit TCLKA and TCLKB that counts events. Pulses A and B are offset by 1/4th of the distance to give a 90-degree offset, so they are known as quadrature counts. Pulse Z occurs only once per rotation. It is fed into the interrupt input (IRQ0) and zeroes out (resets) the counter MTU2_TCNT. When the pulse Z occurs, the rotor angle with respect to the stator frame produces a definite value, preferably zero. If this value is not zero, it is a constant offset that can be measured. Quadrature counters are designed to count these pulses up or down, depending on whether A comes before or after B. That is, the relationship between A and B indicates the direction of rotation.

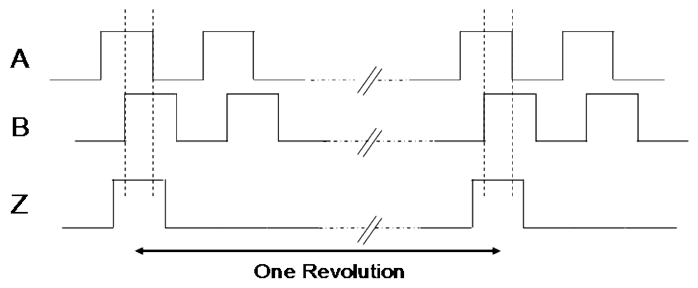

Figure 4 Relationship among the digital encoder pulses A, B and Z

The encoder has been aligned and calibrated with Hall sensor U with zero initial position. The angle is zero count when the Z pulse occurs through the external interrupt IRQ0. From this point onwards it is given a certain count value as the quadrature counter is read. As shown in Figure 5, the phase counting mode 1 is used to up/down count by detecting phase difference between A and B phase. These counts are transformed into a proper angle value for the rotor position.

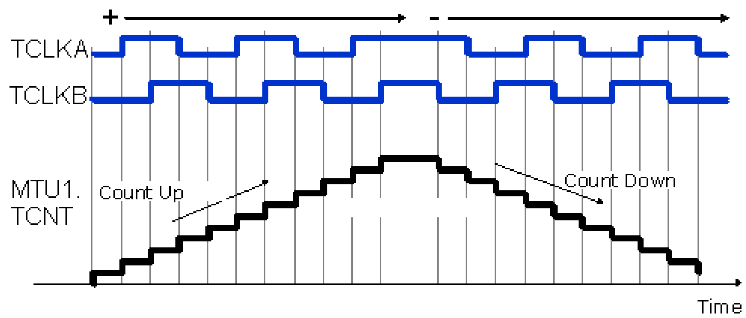

Figure 5 Encoder counting mode operation

Motor speed determines how much the angle of the rotor changes over time. As shown in Figure 6, pulses A and B from the encoder are used at the control loop rate. Two angles are measured at constant time intervals, thus giving the measurements needed to compute speed: delta angle and delta time. Speed is computed by dividing the delta angle $\Delta\theta$ by the delta time.

The motor position is the number of the encoder pulse as N(m)-N(m).

$$\Delta\theta = N(m+1) - N(m)$$

and the motor speed is

$$\omega_r = (N(m+1)-N(m)) / Tsp$$

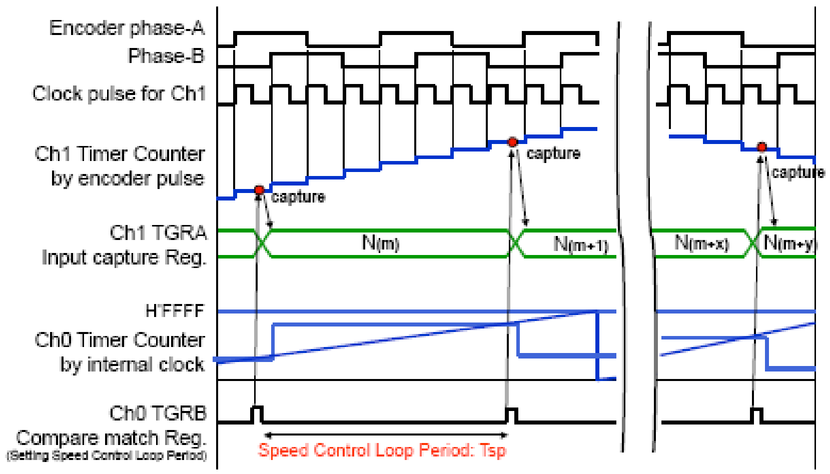

Figure 6 Speed calculation using encoder pulses A and B at control loop rate

5.2 Initial Position Identification

Incremental encoders can only give displacements from the initial position and can't provide absolute position. For PMSM and position control, the initial position is required. Although alignment has been calibrated, the initial starting position before the Z pulse is still unknown.

By means of Hall sensors the rotor initial position can be identified, and further corrected when the rotor starts rotating. Assuming the Hall sensors are located at each phase, as shown in Figure 7. The output signals of the Hall sensors are illustrated in Figure 8. It can be seen that the resolution of the Hall sensor signals are 60° (electrical degree). Table 1 shows the possible combinations corresponding to different positions.

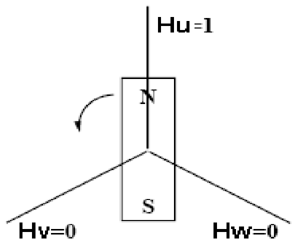

Figure 7 Hall sensors for initial rotor position

From Figure 8 and Table 2, given a specific Hall sensor output combination, the rotor must reside in certain section with a range of 60°. The initial position is determined as follows. When a group of output signals are obtained, for example, (101), we can decide which section the rotor is in (section 1 in this example). We can set the initial position at the center of the section (30° in this example). It can be seen that the maximum error of the initial position is 30°, which occurs when the rotor is at the edge of two regions. However, even with 30° error, the motor is still able to produce sufficient torque to start the motor.

Once the motor starts rotating, the position can be readily corrected when the rotor moves out of the initial region and enters the next section. This position is accurate. In the previous example, when the motor starts rotating in the positive direction from section 1, the rotor position can be corrected when the position is 60°.

Table 2 Relationship between hall sensors and rotor position

Section	Hu	Hv	Hw	Rotor Position
1	1	0	1	0~60
2	1	0	0	60~120
3	1	1	0	120~180
4	0	1	0	180~240
5	0	1	1	240~300
6	0	0	1	300~360

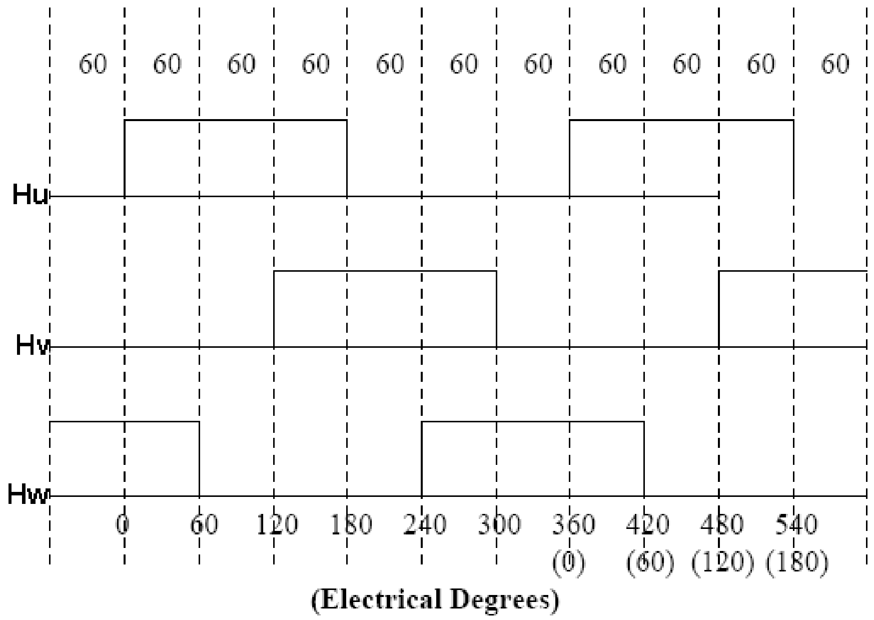

Figure 8 Hall sensor output signals

6. Position Control Strategy

6.1 Block Diagram of Position Control

Figure 9 is block diagram of position control. The position control developed includes two loops. The outer loop is position control to make the motor tracking and holding the given position. The inner loop is current control. Actually it is the torque control loop. The motor currents are sampled through three shunt resistors and converted into the dq axis currents. The control loop here is to control the q axis current for the torque.

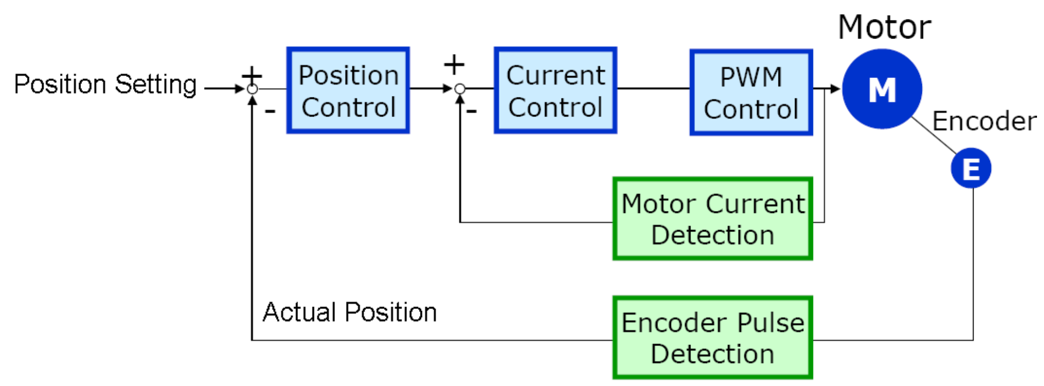

Figure 9 Block diagram of position control

The position control scheme of the PMSM is illustrated in Figure 10. The system has an inner loop of current regulation using vector control, and an outer loop of position regulation. This dual-loop structure ensures the fast torque response by using the vector control, high position accuracy and fast tracking performance with the position controller.

In order to determine the d and q axis currents, the phase currents must be measured. Vector formulation uses Clarke and Park transforms to convert the measured phase currents from the (u, v, w) frame to first transform them in the static orthogonal (a,ß) frame (which is 90 degrees apart), and then, to the rotor frame which is also an orthogonal frame aligned along the magnetic field axes known as the (d,q) frame. These transformations use the transcendental functions sine and cosine of the rotor angle; thus, it is a requirement that the rotor angle is known at the time the calculation is made. The position control requires current sensors, plus an encoder attached to the rotor shaft to measure the rotor position.

Once the currents are transformed in the (d,q) frame, the control algorithm simply runs the PID or PI loop to calculate the required voltages for the torque and flux. These required voltages (Vdc, Vqc) are then transformed back in the (u, v, w) frame using the inverse Clarke and inverse Park transforms to further calculate the PWM duty cycle.

The position command is an input to the position control system. The motor has an encoder mounted on its rotor to give the quadrature pulses A and B, as well as the zero synch pulse Z. All three of the rotor position signals are sent to the MCU's input-capture and timer/quadrature counter peripheral for making position and speed measurements. The commanded position compares with the actual rotor position. The position regulator uses the traditional PID controller, and outputs the torque control command of iq* to make the motor moving and tracking the commanded position.

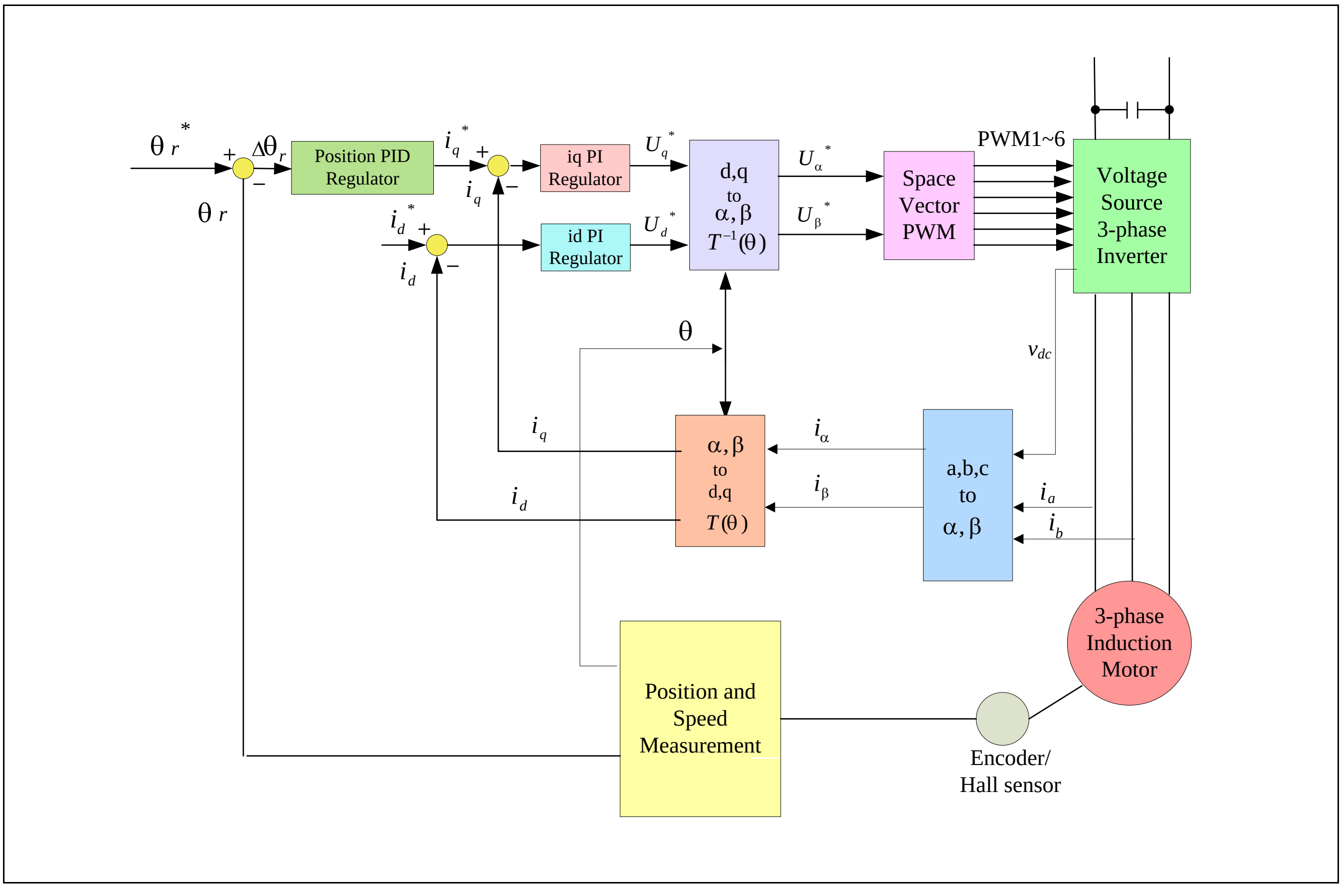

Figure 10 Position control scheme diagram

6.2 Position Control Loop Design

The basic components of a typical servo position control system are depicted in Figure 11. In this figure, the servo position control closes a current loop as described in next section and is modeled simply as a linear transfer function Gireg(s). Of course the servo drive has peak current limits, so this linear model is not entirely accurate; however it does provide a reasonable representation for analysis. For the purposes of this discussion the transfer function of the current regulator or really the torque regulator can be approximated as unity for the relatively lower motion frequencies.

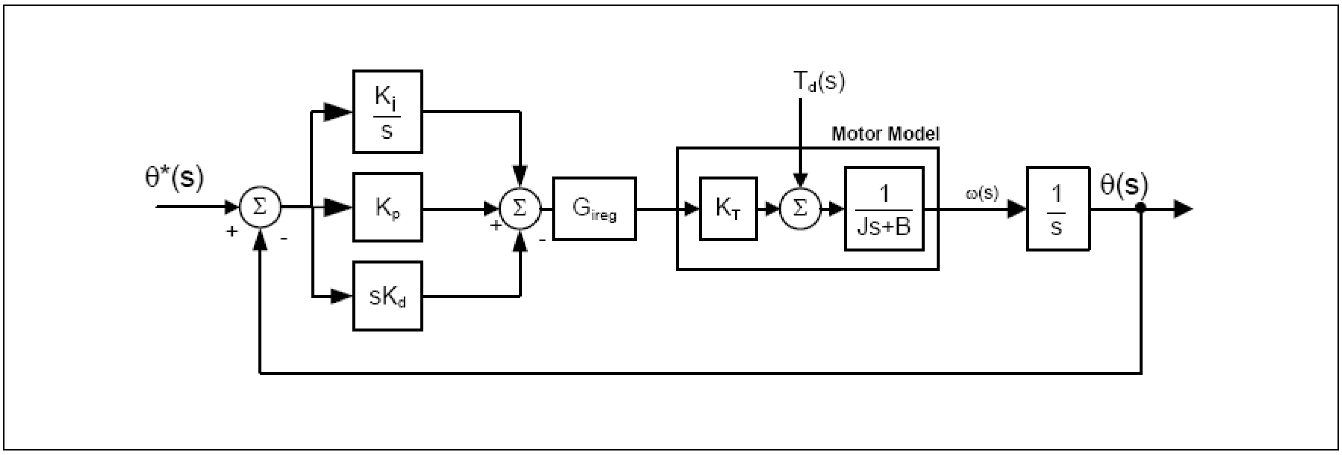

Figure 11 Position PID controller topology

The PMSM is modeled as a lump inertia J, a viscous damping term B, and a torque constant Kt. The lump inertia term is comprised of both the servomotor and load inertia. It is also assumed that the load is rigidly coupled such that the torsional rigidity moves the natural mechanical resonance point well out beyond the position controller's bandwidth. This assumption allows us to model the total system inertia as the sum of the motor and load inertia for the frequencies that can be controlled.

An encoder coupled directly to the motor shaft measures the actual motor position θ (s). External shaft torque disturbances Td are added to the torque generated by the motor's current to give the torque available to accelerate the total inertia J.

Around the current regulator, motor block is the servo position controller that closes the position loop. The basic servo position controller provides both a trajectory generator and a PID controller. The trajectory generator provides only position set-point commands labeled in Figure 9 as θ *(s). The PID controller operates on the position error and outputs a current command.

There are three gains to adjust in the PID controller, Kp, Ki and Kd. These gains all act on the position error defined as:

$$\Delta\theta = \theta^*(s) - \theta(s)$$

Note the superscript “*” refers to a commanded value.

The output of the PID is given mathematically in the time domain as:

$$iq^*(t) = K_p\Delta\theta(t) - K_i\int\Delta\theta(t)dt + K_d\frac{d}{dt}\Delta\theta(t)$$

Loosely speaking, the proportional term affects the overall response of the system to a position error. The integral term is needed to force the steady state position error to zero for a constant position command and the derivative term is needed to provide a damping action, as the response becomes oscillatory. Unfortunately all three parameters are inter-related so that by adjusting one parameter will affect any of a previous parameter adjustment.

Tuning the PID controller can be done if the motor and load parameters are known and the desired frequency response are known. They are adjusted using the following parameters in the header file of “customize.h”.

6.3 Current Control Loop

The current loop is a standard PI type based on the standard Park-Clarke stationary reference frame to rotary reference transformations. The initial rotor position is determined by use of the Hall sensors. Once a Hall transition occurs, the rotor position is then determined by reading the incremental encoder. The basic block diagram for the current vector control is shown in Figure 12.

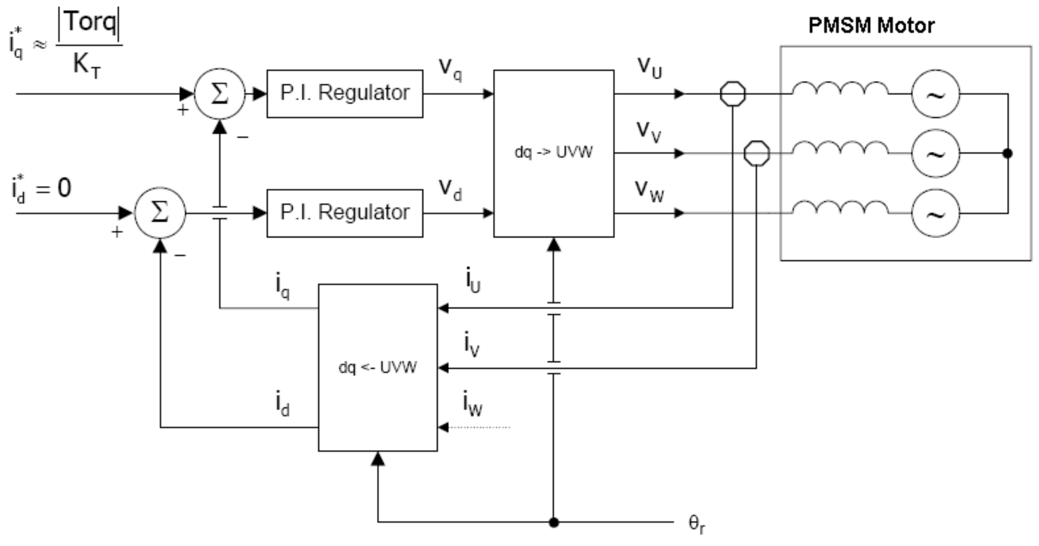

Figure 12 Block diagram of current vector control

Neglecting motor saliency, the commanded q axis current, iq* is linearly related to the commanded torque. The “d” axis current command, id* is set to zero as field weakening is not required. The transformation takes two steps. First, the stationary currents are transformed to an arbitrary stationary pair of orthogonal axes α , β and second, the axes are then rotated to the rotor axes for control purposes.

The typical current PI controller is depicted in Figure 13. Kp and Ki are the proportional gain and integration gain, respectively, which can be adjusted by the software. The hardware gain Kb takes into account the bus voltage.

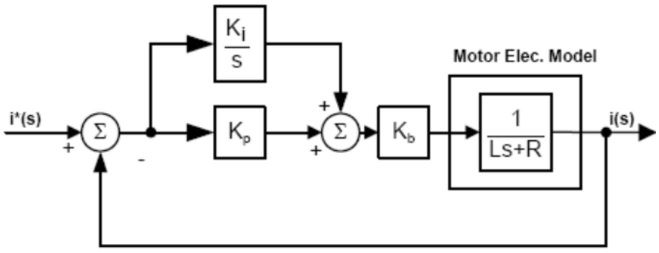

Figure 13 Current PI controller topology

The transfer function of the block diagram is:

$$\frac{i(s)}{i^*(s)} = \frac{\left(\frac{K_p K_b}{L}\right)s + \left(\frac{K_i K_b}{L}\right)}{s^2 + \left(\frac{K_p K_b + R}{L}\right)s + \left(\frac{K_i K_b}{L}\right)}$$

It has a characteristic equation in the form of:

$$s^2 + 2\xi\omega_0 s + \omega_0{}^2 = 0$$

Therefore:

$$K_p = \frac{2L\xi\omega_0 - R}{K_b}$$

$$K_i = \frac{\omega_0{}^2 L}{K_b}$$

The system exhibits the standard second order response with the addition of a real zero. To tune the system, the high frequency of 500Hz needs to be first set for Kp, and then slowly increase the integral term Ki to bring our steady state error to zero.

7. Software Description

7.1 Overall Software Structure

Position control algorithm is implemented with the complete C code using Renesas' RX62T MCU floating point unit. The overall software structure is shown in Figure 14.

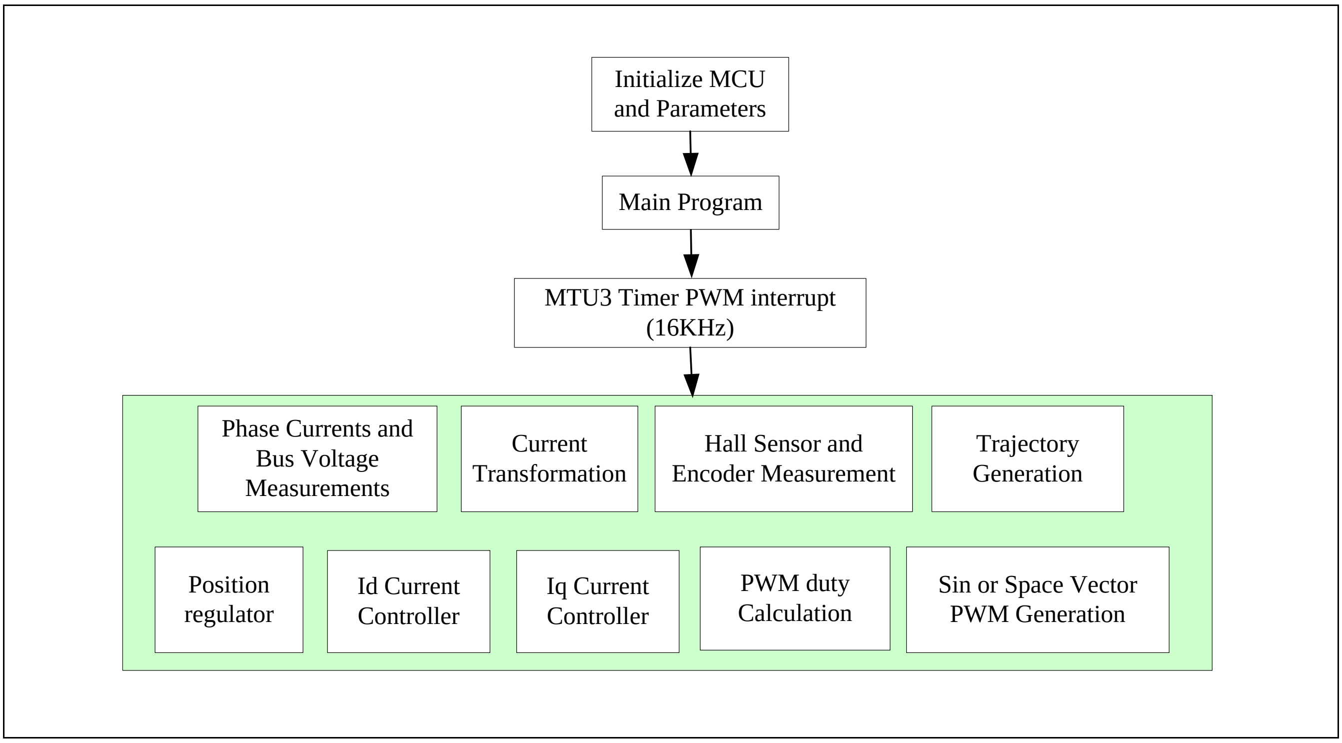

Figure 14 Position software architecture

The procedures include:

- initializations of RX62T MCU, motor and control parameters
- current offsets calculation
- bus voltage and phase currents measurements
- hall sensor and encoder reading
- initial position identification
- rotor position calculation
- vector control transformation
- motion profile - trajectory generation
- position regulator
- current controllers
- PWM duty calculation
- space vector PWM generation

7.2 Software e2Studio Workspace

Shown in Figure 15 is the workspace for position control using Renesas' e2Studio IDE.

- All codes are written in the floating point C language;
- The software is modularized according to the position control block diagram (as shown in Figure 10);
- I/O definitions and basic MCU drivers are automatically ported by e2Studio from HEW;
- Motor and control parameters are easily tuned through a header file of "customize.h" and GUI user interface.

The codes include dbsct.c; hwsetup.c, intprg.c; main.c; mcrplibf.c; motorcontrol.c; resetprg.c; userif.c and vectbl.c.

- dbsct.c includes structures used by the runtime library both to clear un-initialized global variables and to write initial values into initialized global variable sections.
- hwsetup.c is hardware initializations.
- vecttbl.c contains the array of addresses of ISRs.

- resetpr.c has functions called just after reset.
- intprg.c is entry points for all of standard ISRs vectors.
- main.c including: initialization of control parameters, MTU3 timer, interrupts, serial communication, encoder capture definitions; and uploading eeprom parameters. The current sensor offsets are calculated before the output of PWMs. The while loop executes parameter update and SCI communication with graphic user interface.
- The motorcontrol.c is a major code for position control, which contains most of functions and function calls to implement position control.
- Mcrplibf.c mainly includes vector control transformations – Clarke, Park, and inverse Clarke and Park transformations, and sine and space vector PWM generation.

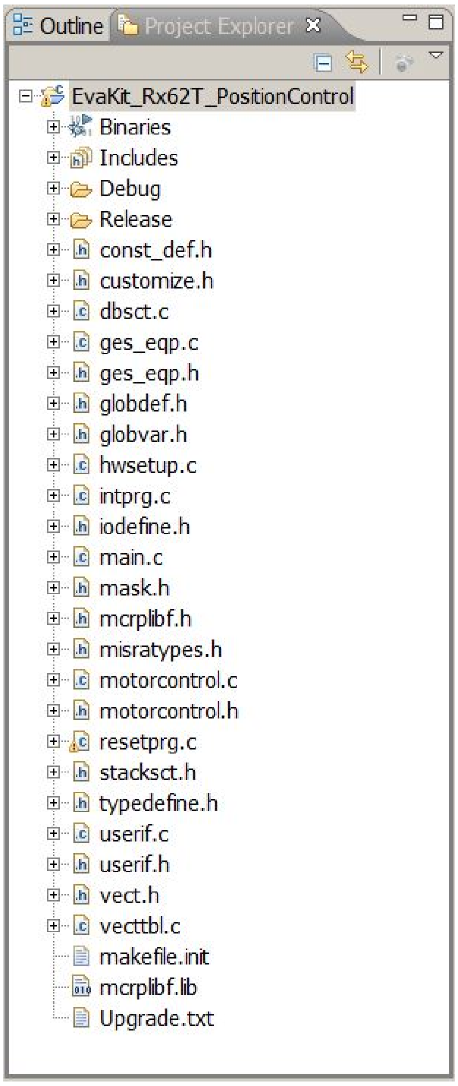

Figure 15 Encoder counting mode operation position control software workspace

7.3 Hall and Encoder Based Position and Speed Measurement

Figure 16 is a flowchart of position measurement. The procedures for the position measurement based on hall sensors and encoder are:

- Initialize hall sensor and encoder capture timer registers and I/O ports;
- Identify the rotor initial position using hall sensor;
- Move the motor to capture the position using encoder pulses;
- Calibrate the rotor position once the hall commutation changes;
- After calibration, recalculate the rotor position;
- Check encoder Z pulse and reset the position offset and encoder pulse capture timer count;
- Calculate the rotor position and motor speed.

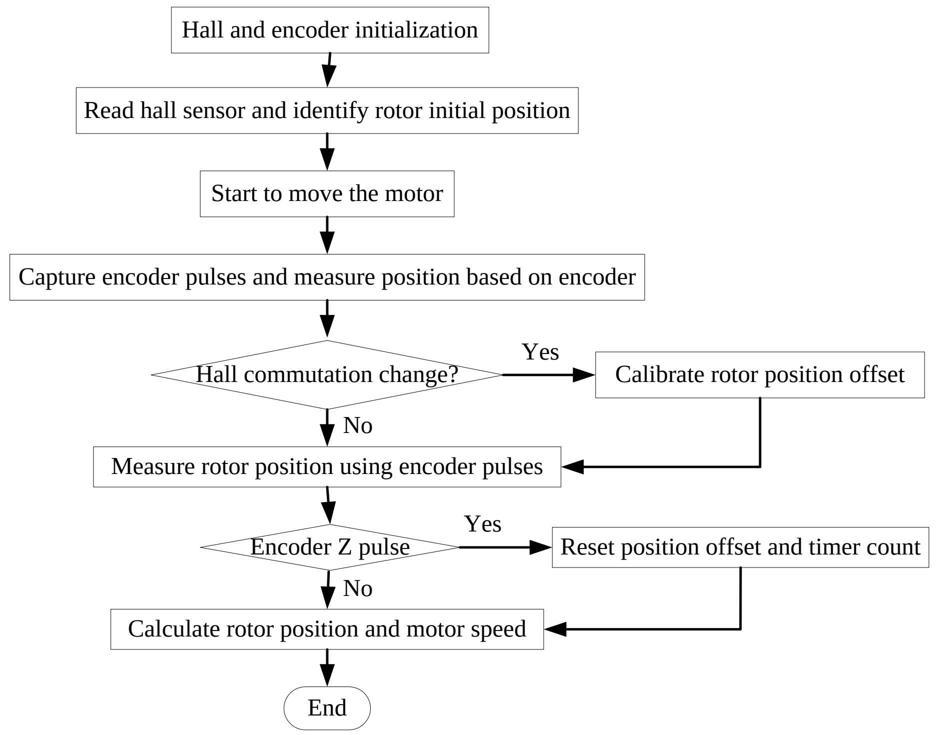

Figure 16 Encoder counting mode operation Flowchart of position and speed measurement

7.4 PWM Interrupt for Position Control

The position profile generation and position control are put in the PWM interrupt with 16 kHz carrier frequency. Figure 17 is a flowchart of PWM interrupt.

The procedures in the PWM interrupt of MC_ConInt () are:

- Measure motor phase motor currents and DC bus voltage;
- Calculate motor position and speed using hall sensors and encoder;
- Transfer motor currents into dq currents;
- Current control loop;
- Update trajectory generator and position profile;
- Position control loop;
- PWM generation using space vector PWM modulation or sinusoidal PWM modulation.

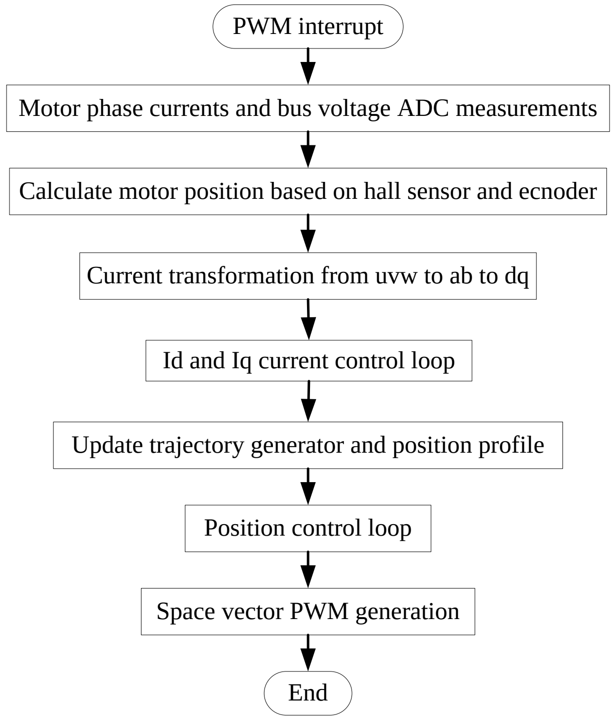

Figure 17 Flowchart of PWM interrupt for position control

8. Motor and Position Control Parameters

8.1 Tuning through header file

According to the motor data sheet and position control requirements, motor and control parameters, and motion profile should be properly tuned.

Motor and control parameters required in the code of “customize.h” include:

- #define ENC_EDGES_CUSTOM 4000 // total encoder Edges/Revolution
- #define PWM_FREQ_CUSTOM 16000 // PWM Frequency in Hz
- #define SAM_FREQ_CUSTOM 16000 // Sample Frequency in Hz
- #define C_POLI_CUSTOM 4 // polar couples number
- #define R_STA_CUSTOM 8 // stator phase resistance in Ohm/OHM_DIV
- #define L_SYN_CUSTOM 10 // synchronous inductance in Henry/HEN_DIV
- #define POS_MIN_CUSTOM 0 // minimum position in counts
- #define POS_MAX_CUSTOM 40000 // maximum position in counts
- #define KP_CUR_CUSTOM 60 // K prop. current control
- #define KI_CUR_CUSTOM 80 // K integ. current control
- #define K_P_POSITION 10 // K prop. position control
- #define K_I_POSITION 12 // K integ. position control
- #define K_D_POSITION 150 // K derivative psotion control

8.2 Operation through GUI

The motor and control parameters can be tuned through Renesas friendly graphic user interface as shown in Figure 18. Without modifying the code, the parameters can be set for the different motors and applications. There is a parameter window to set up 20 parameters. Scrolling up and down through these parameters, the user can make changes to the settings, and “Write” to EEPROM, but this doesn’t change the “customize.h” file. The original values will be restored upon RESET. From Figure 19, it can be seen that these parameters mirror the #defines in the “customize.h” file. The motor and control parameters can be easily changed by the GUI.

In the meantime, the GUI has position control window to set the commanded position, and display the motor actual operation status.

Figure 18 GUI interface of evaluation kit

INDEX	DESCRIPTION	UNIT	MIN	MAX	VALUE	VALID
1	00. Default Parameters Setting	-	0	32767	0	true
2	01. Minimum Speed	rpm	200	5000	500	true
3	02. Maximun Speed	rpm	1000	20000	2500	true
4	03. Acceleration	rpm/s	1	10000	1000	true
5	04. Deceleration	rpm/s	1	10000	1000	true
6	05. Polar couples	-	1	5	5	true
7	06. Startup Current	Apk/10	0	5000	10	true
8	07. Maximum "q" Current	Apk/10	0	5000	20	true
9	08. Stator Resistance	Ohm/10	0	5000	17	true
10	09. Synchronous Inductance	Henry/10000	0	5000	12	true
11	10. Startup Time	ms	300	10000	1000	true
12	11. Current Loop Kp	-	0	2047	60	true
13	12. Current Loop Ki	-	0	1023	80	true
14	13. Speed Loop Kp	-	0	4095	10	true
15	14. Speed Loop Ki	-	0	4095	100	true
16	15. Startup offset V	V/10	0	32767	0	true
17	16. Startup delta V	V/10	0	32767	0	true
18	17. PI Tuning trigger	-	0	32767	0	true
19	18. Free	-	0	32767	0	true
20	19. Free	-	0	32767	0	true

Figure 19 Parameter window

Appendix A - References

1. RX62T Group User's Manual: Hardware, R01UH0034EJ0110, April 20, 2011
2. DevCon 2010 Courses:
   - ID-620C, Complete Motor Control Integration with RX62T.
   - ID 623C, Understanding Sensor-less Vector Control with Floating Point Unit (FPU) Implementation.
3. DevCon 2008 Courses:
   - ID-504, Speed Control using a Digital Encoder and Vector Formulation
4. Application Note of Sensorless Vector Control of three-phase PMSM motors, REU05B0103-0100/Rev.1.00, March, 2009
5. Application Note of Mcrp05: Brushless AC Motor Reference Platform, REU05B0051-0100, Feb, 2009

Website and Support

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

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

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Nov. 18, 2011.	—	First edition issued
2.00	Jan. 31, 2014.	—	Second edition issued

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU 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. 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 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. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment 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 moment 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 moment 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 moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has 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. Moreover, 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.

5. Differences between Products

   Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.

   — The characteristics of an MPU or MCU in the same group but having a different part number may differ in terms of the 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.

RENESAS

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.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, 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 Rd., Putuo District, Shanghai, China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, 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 906, 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 Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 3.0