RX111

Sensorless Vector Control of PMSM

R01AN1837EU0100
Rev.1.00
Dec 09, 2013

Introduction

This document describes software design and implementation of sensorless vector control (SVC) of permanent magnetic synchronous motors (PMSM) using Renesas RX111 microcontroller (MCU).

It mainly describes RX111 SVC hardware platform, RX111 MCU, and software implementation. The sensorless vector control algorithm is based on the method introduced in Renesas Application Note REU05B0103-0100/Rev.1.00.The software in this document can be applicable to following devices and platforms.

MCU: RX111 Group
Motor: Three-phase BLDC motor and PMSM
Platform: Renesas MCU plug-in board (P03301-D1-006) and High Voltage Inverter Demo Platform (YMCRPHV2DP)
Control algorithm: Sensorless vector control.

Contents

List of Figures

List of Tables

1. Overview

This document describes software design and implementation of sensorless vector control of permanent magnetic synchronous motor (PMSM) with shunt current measurement using Renesas RX111 microcontroller.

Today, cost effective high performance microcontrollers are available and therefore many design groups are now interested in implementing sensorless vector control of three-phase permanent magnet synchronous motors (PMSM). It has become easy to implement sophisticated advanced motor control schemes into digitized high performance motor control systems.

The RX111 is a 32-bit RX CPU core high-performance microcontroller with a maximum operating frequency of 32MHz and 49 DMIPS. Equipped with multifunction timers (MTU2a, GPT), event link controller (ELC) and high-speed 12-bit A/D converter, and 10-bit A/D converter, the RX111 MCUs are an ideal solution for cost effective high performance motor control solutions.

This document presents RX111 sensorless vector control solution, which has been implemented on the RX111Renesas MCU plug-in board (P03301-D1-006) and High Voltage Inverter Demo Platform (YMCRPHV2DP). It describes hardware platform, RX111 MCU, and software implementation. The sensorless vector control algorithm is based on the method introduced in Renesas' Application Note REU05B0103-0100/Rev.1.00.

Software described in the application note is applicable to following devices and platforms.

- MCU:Rx100 and Rx200 family
- Motor: three-phase BLDC motor and PMSM
- Platform: Renesas' MCU plug-in board and High Voltage Inverter Platform
- Control algorithm: Sensorless Vector Control

2. Hardware Platform

RX111 SVC is implemented with Renesas' High Voltage Inverter Platform and RX111 MCU plug-in board. The hardware setup, shown in *Figure* 1 is versatile and can be applied to any types of motors driven by a three-phase power inverter. The voltage source inverter (VSI) is used to regulate the motor speed of three-phase PMSM by varying frequency and voltage.

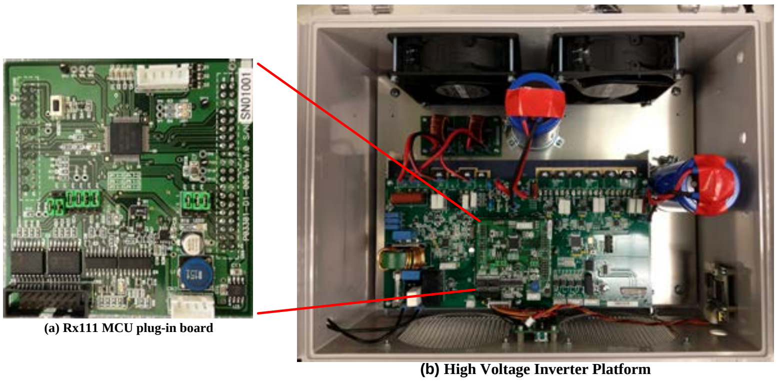

Figure 1 System setup with (a) RX111 MCU plug-in board and (b) High Voltage Inverter Platform

The system consists of High Voltage Inverter Platform and RX111 MCU plug-in board.

The High Voltage Inverter Platform has an input AC/DC rectifier, a DC link and an output DC/AC inverter. It is capable of driving high voltage medium current motors. The bus voltage could be up to DC 400 volts with the current up to 50 amps. The board is designed to measure bus voltage, bus current, and three phase motor currents through shunt current resistors. Each phase current will be amplified with gain 10 and offset to 0 to 5.0V (or 0 to 3.3V). The three phase motor back EMFs are also detected by ADC converters. The speed is input through hall sensor and encoder circuits.

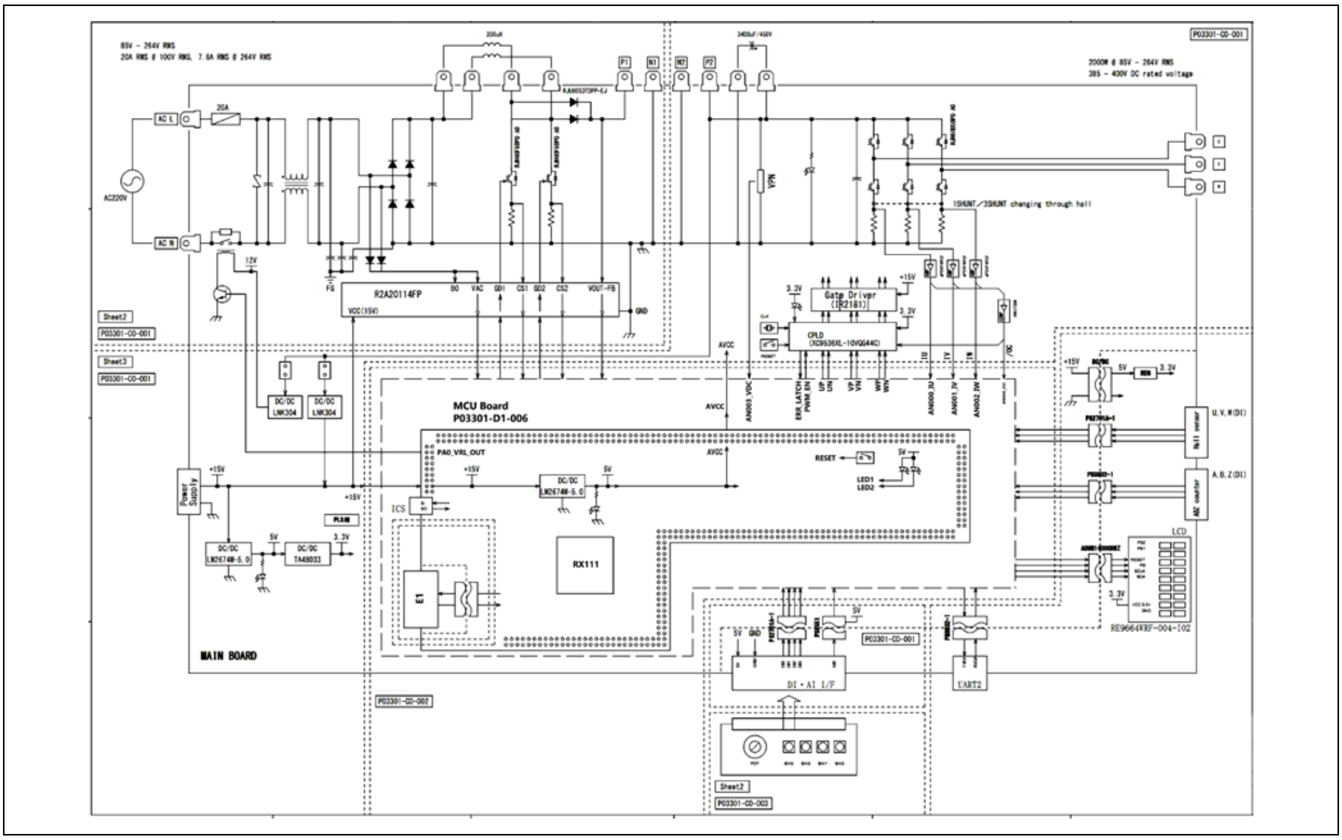

Figure 2 System block diagram for High Voltage Inverter Platform

The central processing unit (CPU) can use any of the board populated with Renesas' R8C, SH, RL78, and RX, which can be directly plugged onto the High Voltage Inverter Platform board. In this demo, RX111 MCU plug-in board is used for sensorless vector control drive shown in *Figure* 2. The demo system has following features:

- Speed sensor inputs of hall sensor, encoder, and tachometer
- Three low side phase current measurement via shunt current measurement using precision resistors and amplifiers with gain ratio 1:10.
- 110 or 220VAC input and output, and bus voltage of 160V or 320V
- Three-phase timer supports multiple PWM modes including complementary pairs with automatic dead-time insertion
- Support all of RX motor MCU board to directly plug in
- Various power modules can be used (10A, 16A and 20A modules).
- LCD display to monitor the operation status
- Support the standalone mode set by potentiometer and push buttons

3. Specification and Performance Data

The major specification data of RX111 sensorless vector control are described as following:

- Input voltage: 110VAC
- Rated bus voltage: 160VDC
- Output voltage: 110VAC
- Rated output power: 1kW
- PWM Switch frequency: 10KHz
- Control loop frequency: 10KHz

- ❖ Current measurement: 3 precision shunt resistor
- ❖ CPU bandwidth: 46.0%
- ❖ Used flash memory: 9.982Kbytes

4. RX111 Microcontroller

RX111 is a 32-bit and 32MHz microcontroller. As part of RX family of microcontrollers, RX111 MCU offers high performance and high throughput required for today's sophisticated designs, especially vector control and other high performance motor control methods. *Figure* 3 is a block diagram of RX111MCU.

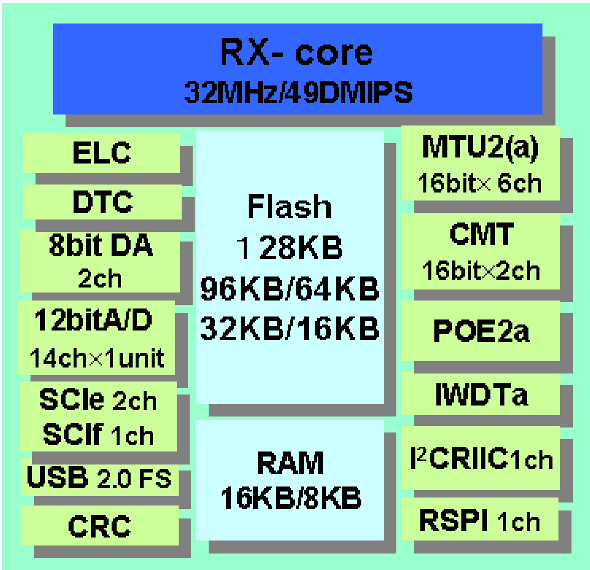

Figure 3 Block diagram of RX111 MCU

Key features of RX111 MCU:

- ❖ High-performance single-chip RISC with Rx core
  - 49DMIPS at 32MHz
  - Built-in 32-bit multiplier
- ❖ Built-in large-capacity memory (ROM/RAM)
  - Flash memory up to 128KB
  - RAM up to 16KB
- ❖ Event link controller (ELC)
  - Module operation by event signals
- ❖ Timers
  - Powerful timer: MTU2(a) (16bit×6ch),
  - Compare match timer (CMT)(16bit×2ch)
  - Port output enable 2 (POE2a)
  - Independent watchdog timer (IDWTa)
  - Realtime clock (RTCc)
- ❖ 12bit A/D converters (14ch x 1unit)
- ❖ Data transfer controller (DTC)
- ❖ DA: 4 channel
- ❖ Serial communication interface (SCIe, SCIf) × 3 channels
- ❖ I2C Bus Interface RIIC : 1ch
- ❖ Serial peripheral interface RSPI: 1ch

- ❖ USB 2.0 host/function module USB

5. System Control Block Diagram

Figure 4 depicts block diagram of a senorless vector control of PMSM based on the Renesas RX111 Microcontroller.

The RX111 timer MTU2a channel 3 and 4 are used to generate 6 PWM signals to drive the motor in the complementary mode. The PWM modulation uses the space vector PWM or the sinusoidal PWM with the third harmonic. The three-phase inverter generates three phase voltages with variable frequency and amplitude to drive the motor to the desired voltage.

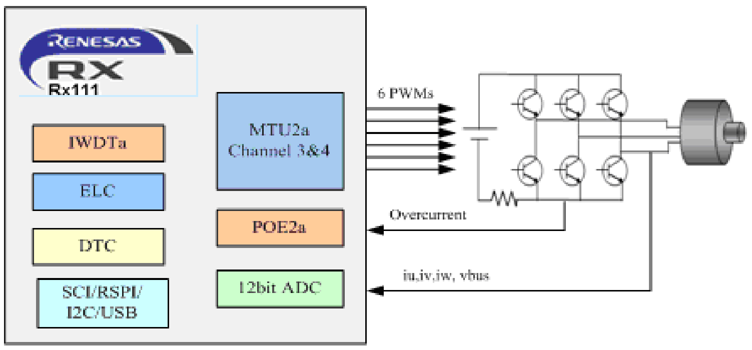

Figure 4 System control block diagram

The motor currents of i_u, i_v, i_w are measured by three shunt resistors via op-amp. The currents of i_u, i_v are measured by 12-bit ADC of channel AN000, and AN001, respectively. In the meantime, the bus voltage is measured by 12-bit ADC unit 0 of channel AN003. The MTU2a channel 3 and 4 are used to generate 6 PWM signals in the complementary mode. The peripherals for the PWM signals are listed in Table 1.

Peripherals Used	RX111 PWM Pin out		
	Signal Name	RX111 Pin-Numbers	Port
MTU2a_3 / TIOC3B	Up	33	PB7
MTU2a_3/ TIOC3D	Un	34	PB6
MTU2a_4/ TIOC4A	Vp	36	PB3
MTU2a_4/ TIOC4C	Vn	37	PB1
MTU2a_4 / TIOC4B	Wp	26	P54
MTU2a_4/ TIOC4D	Wn	25	P55

Table 1 MTU2a 3 and 4 peripherals for PWM signals

Motor phase currents and DC bus voltage are input through 12-bit A/D converters. The conversion mode for ADCs is in the single mode. The channel numbers and it conversion ratio are listed in Table 2.

Item	12-bit ADC Channel	Conversion ratio (actual value/ADC input value)	
Phase v current - iu	AN000	AVCC = 5V	-50A to 50A / 0 to 5V
		AVCC = 3.3V	-33A to 33A / 0 to 3.3V
Phase v current - iv	AN001	AVCC = 5V	-50A to 50A / 0 to 5V

		AVCC = 3.3V	-33A to 33A / 0 to 3.3V
DC bus voltage - vbus	AN003	0-500V / 0-5V	

Table 2 ADC peripherals

6. RX111 Sensorless Vector Control Strategy

Vector control formulation depicted in *Figure* 5 provides a straightforward way to control the flux and the torque in much the same way as the control of DC motors – the flux is controlled by the Id current, while the torque is controlled by the Iq current. Due to its nature, the 3-phase motor has three windings and three currents which are 120 degrees apart. 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. 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.

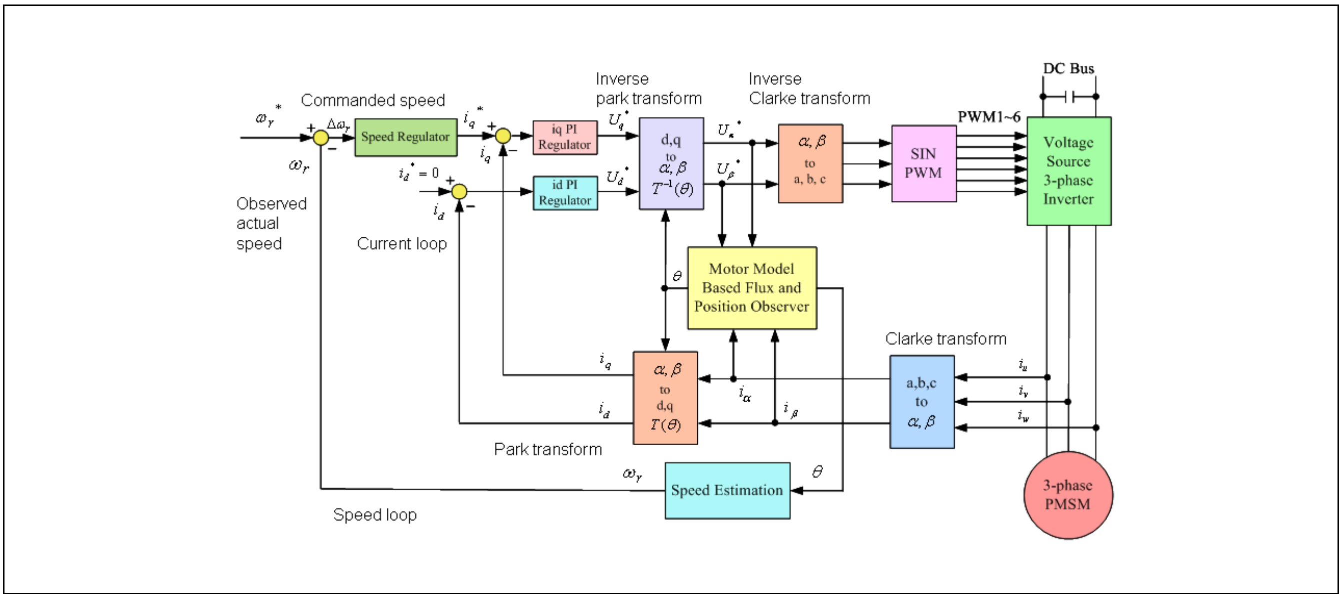

Figure 5 Block diagram of sensorless vector control

All these blocks from the inner current loop shown in *Figure* 5 with two computing blocks known as flux observer and speed estimation, which are a special part of the sensorless vector control formulation. When a sensor is used to measure the rotor angle and speed measurements, these two blocks change significantly.

In sensorless implementation, because there is NO sensor to measure the angle and speed, a motor model is used to calculate the flux and estimate the speed based on the measured phase currents and motor parameters, thus making computations more involved. These computations further involve the use of transcendental functions and filters.

Phase currents measured with ADC are first converted into proper current values. Third-order filters are employed to reduce the noise and other undesired effects and integrate the flux continuously as required. Finally, the inverse tangent ARCTAN function is used to derive the rotor angle. The speed is estimated based on two consecutive rotor angle computations again with some filtering employed.

7. RX111 SVC Software Implementation

The sensorless vector control software has the following features:

- All codes are written in C language with Renesas e2studio IDE;
- The software is modularized according to the SVC block diagram (as shown in *Figure* 6);
- Motor and control parameters are easily tuned through a header file of "customize.h".

7.1 Overall Software Structure

RX111 SVC algorithm is implemented with the complete C code in the standard modules. The overall software architecture is shown in *Figure* 6.

The control has two major blocks: MC modules, and SVC modules. The blocks in MC modules are motor and MCU initialization, parameter definitions, ADC sampling, speed setting and ramp generation, startup procedure, current id and iq regulators, speed regulator, PWM generation and fault protections. The SVC modules include vector control transformations (Clarke and Park transformation), PI controller, flux and speed observer, and PWM duty calculation.

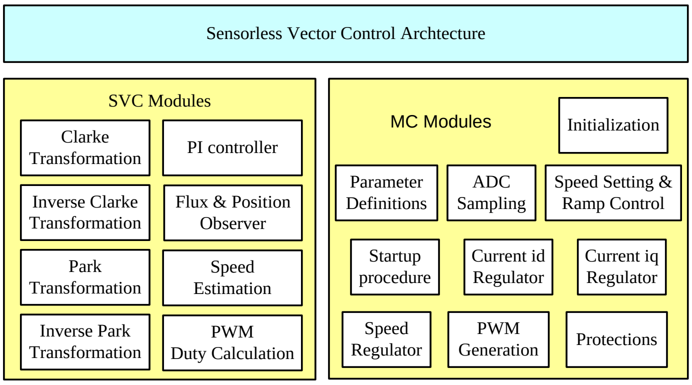

Figure 6 Software architecture of RX111 SVC

7.2 SVC Workspace with Renesas e2studio

RX111sensorless vector control software architecture is similar to the one in Renesas Application Note REU05B0103-0100/Rev.1.00. Shown in *Figure* 7 is the workspace for RX111 sensorless vector control using Renesas e2studio.

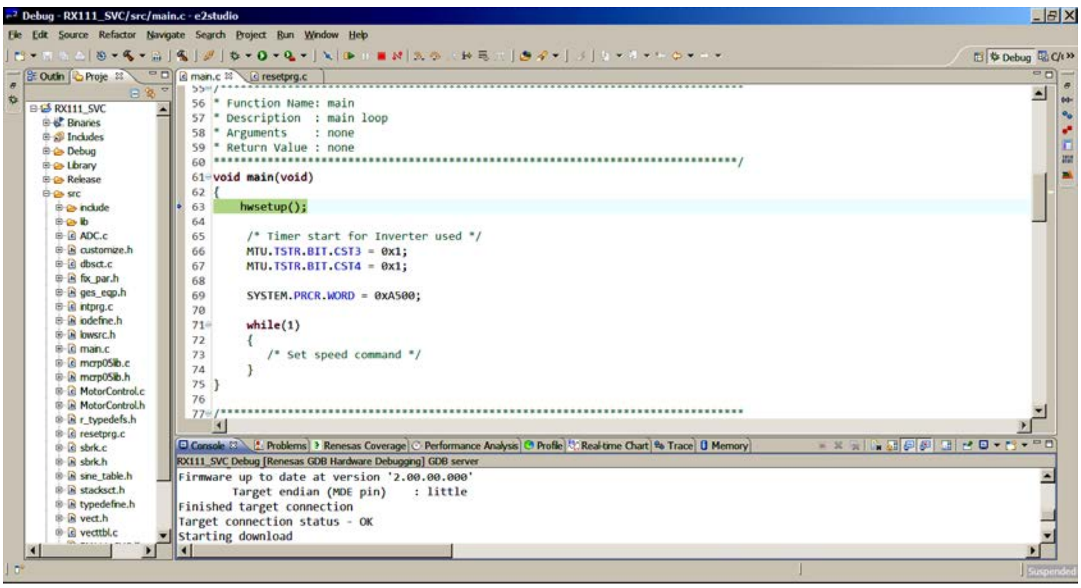

Figure 7 RX111 SVC software workspace with e2studio

The codes include dbsct.c; hwsetup.c, intprg.c; main.c; motorcontrol.c, mcrp05lib.c, resetprg.c, userif.c and vectbl.c. Core sensorless vector control modules for vector control transformation and speed and position observer are put in the mcrp05lib.c.

7.3 Flowchart of RX111 SVC

Figure 8 shows the control logic of RX111 SVC motor control.

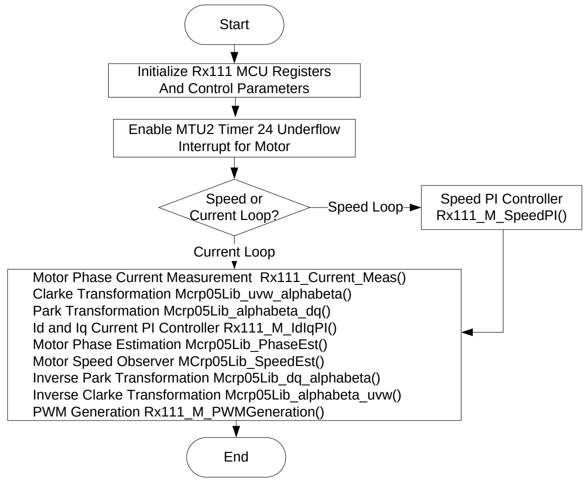

Figure 8 Control logic of SVC

The software first initializes RX111 hardware setup including system clocks, IO definitions, MTU2a timers, ADCs, etc. The SVC control algorithm is executed in MTU2_4 underflow interrupt in 10 kHz frequency. The control starts with the closed current loop control. After the commanded time, it automatically switches to the closed speed loop control with the estimated speed and position.

7.4 Flowchart of MTU2_4 PWM Interrupt

MTU2_4 timer interrupt is to implement RX111 sensorless vector control. *Figure* 9 is the flowchart of the interrupt. It starts with the open loop, and then switches to the closed speed loop.

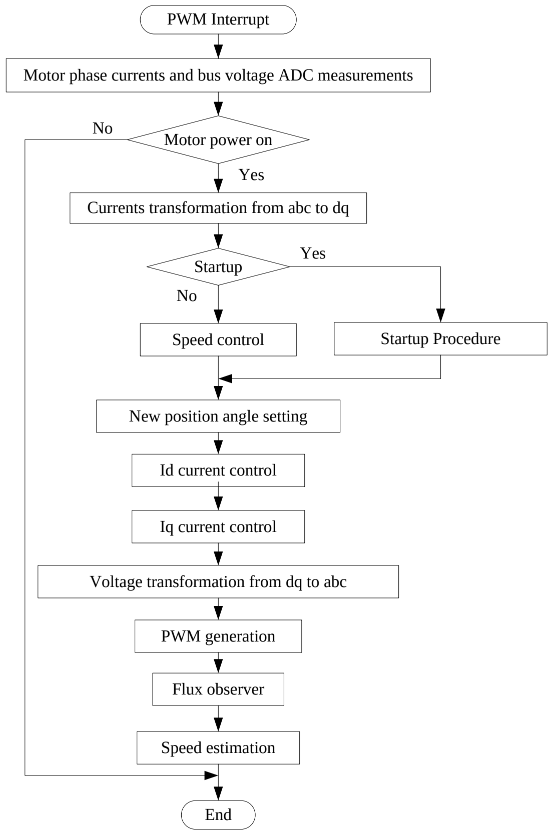

Figure 9 Flowchart of PWM interrupt

The procedures in the interrupt of Rx111_M_Closedloop_ConInt() are:

- Motor phase motor currents and DC bus voltage are first sampled;
- When the motor powers on, the startup procedure handles the open loop starting;
- After the motor starts up at the given time, the system switches into the closed speed loop;
- The rotor position and the speed are estimated in sync with the carrier frequency in order to update the position and the speed timely;
- The current PI controller outputs of v_d and v_q are transformed back to three-phase voltages of v_u v_v and v_w, which are used to calculate PWM duty ratios to drive motor to the desired voltages.

7.5 Functions in Main Code

Shown in *Figure* 10 are function calls in main.c. The initializations include motor and control parameters; MTU2a PWM timer registers. The current sensor offsets are calculated before the PWMs turn on. The while loop waits for the PWM interrupt. The PWM interrupt of Rx111_M_Closedloop_ConInt() is executed in the 10 kHz frequency.

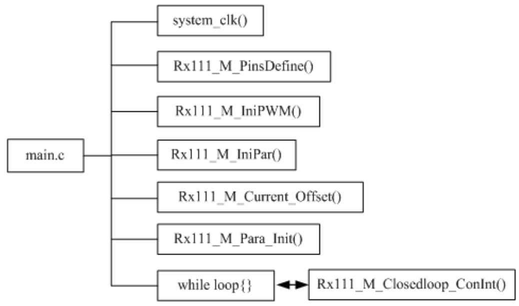

Figure 10 Block diagram of function calls in main.c

7.6 Functions in Motor Control Code

The functions in motorcontrol.c are shown in *Figure* 11.

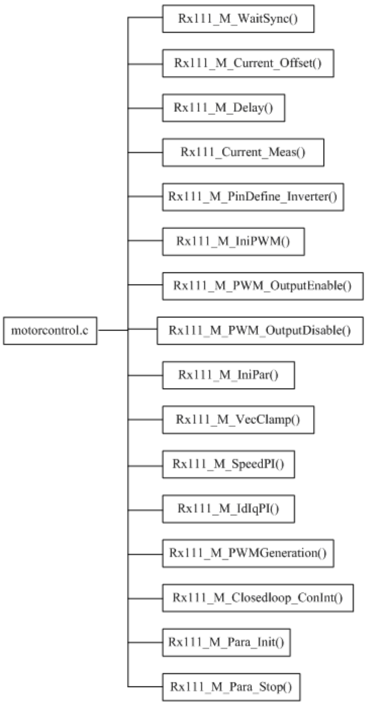

Figure 11 Functions in motorcontrol.c

The motorcontrol.c is a major code for SVC of dual motor control, which contains most of functions and function calls to implement SVC.

- Rx111_M_Current_Offset() calculates motor current sensor offsets;
- Rx111_M_Current_Meas measures motor phase currents;
- Rx111_M_Delay() generates the time delay;
- Rx111_M_WaitSync() is a time synchronization function;
- Rx111_M_PinDefine_Inverter() initializes port registers;
- Rx111_M_IniPWM() initializes MTU2a timer Channel 3 &4 PWM registers;
- Rx111_M_PWM_OutputEnable() starts PWMs and enables the MTU2_4 PWM outputs;
- Rx111_M_PWM_OutputDisable() stops PWM outputs;
- Rx111_M_InitPar() initializes motor and control parameters;
- Rx111_M_IdIqPI() is the motor current Id and Iq regulators;
- Rx111_M_SpeedPI() is motor speed PI regulator;
- Rx111_M_PWMGeneration() generates 6 PWM signals;
- Rx111_SpeedRamp() accelerates and decelerates the reference speed to the desired speed;

- ❖ The PWM interrupt of Rx111_M_Closedloop_ConInt() executes the SVC algorithm – startup, speed loop, current loops, vector control transformation and PWM generation;

7.7 Functions in SVC Motor Control Code

Functions in the SVC modules of mcrp05Lib.c include the functions of vector control transformations, flux and phase observers and speed estimation.

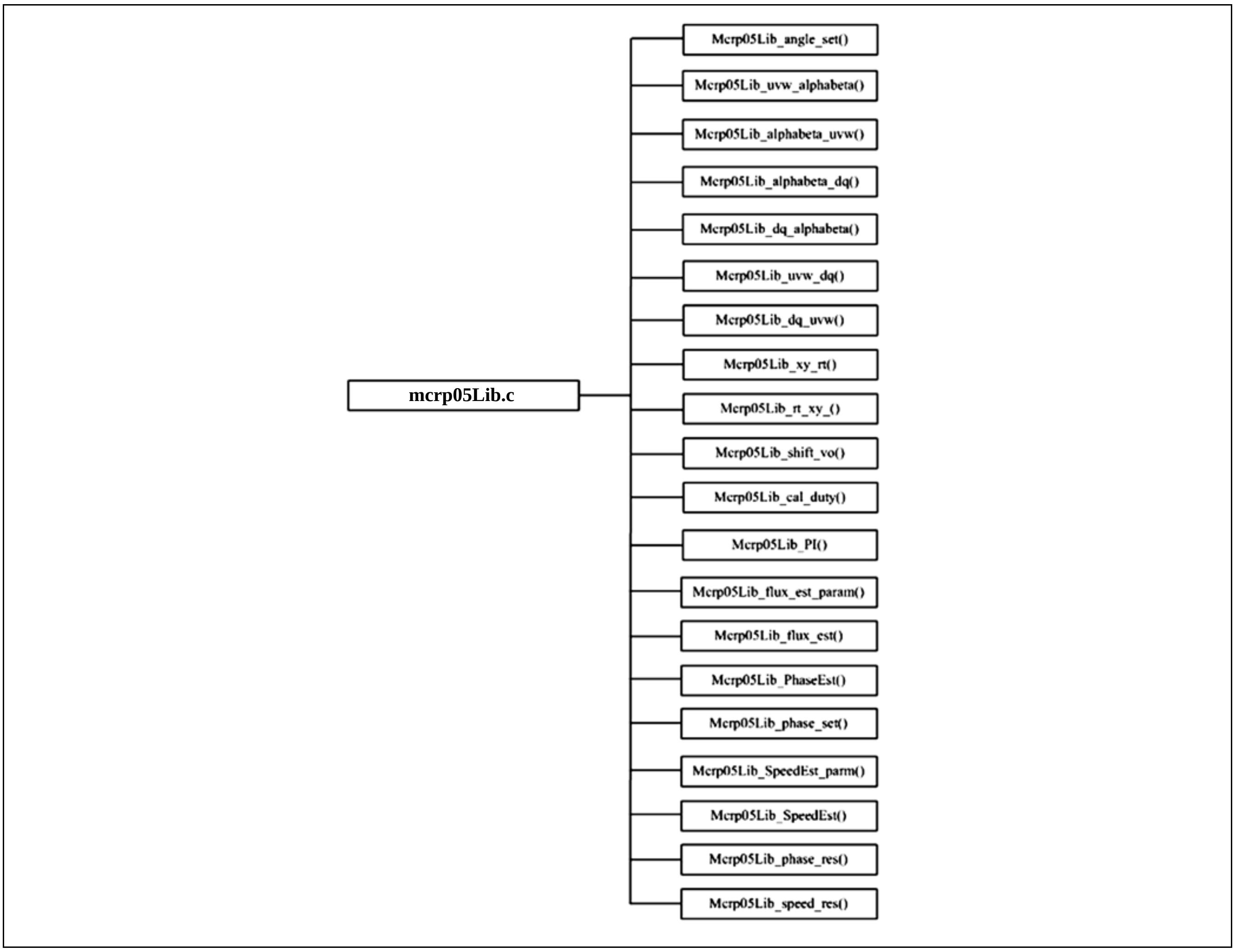

Figure 12 Functions in Mcrp05Lib.c

Figure 12 shows the functions in the code.

- ❖ Mcrp05Lib_angle_set()transfers the motor phase angle to the sine and cosine values for Park and inverse Park transformations;
- ❖ Mcrp05Lib_uvw_alphabeta() is the Clarke transformation;
- ❖ Mcrp05Lib_alphabeta_uvw () is the inverse Clarke transformation;
- ❖ Mcrp05Lib_alphabeta_dq () is the Park transformation;
- ❖ Mcrp05Lib_dq_alphabeta () is the inverse Park transformation;
- ❖ Mcrp05Lib_uvw_dq () is the unitary transformation;
- ❖ Mcrp05Lib_dq_uvw() is the inverse unitary transformation;
- ❖ Mcrp05Lib_xy_rt () is to transfer the x, y vectors to the amplitude and angle;
- ❖ Mcrp05Lib_rt _xy () is to transfer the amplitude and angle to the x, y vectors;
- ❖ Mcrp05Lib_shift_vo() outputs the voltage shift.
- ❖ Mcrp05Lib_cal_duty() calculates the duty cycle according to the desired output voltage;
- ❖ Mcrp05Lib_flux_est_param() sets up the flux estimation filter parameters;

- Mcrp05Lib_flux_est() is flux estimation;
- Mcrp05Lib_PhaseEst() estimates the rotor position;
- Mcrp05Lib_phase_set() pre_sets the phase estimation vectors;
- Mcrp05Lib_SpeedEst_param()sets up the speed estimation filter parameters;
- Mcrp05Lib_SpeedEst() estimates the speed;
- Mcrp05Lib_phase_res() resets the motor phase estimation;
- Mcrp05Lib_speed_res() resets the motor speed estimation;
- Mcrp05Lib_PI() is the PI control for the current and speed controller;

8. RX111 SVC Motor and Control Parameter Tuning

8.1 Tuning through header file of "customize.h"

Shown in *Figure* 13 are a BLDC motor and its data sheet. The motor is a 2-pole 3-phase BLDC motor. The rated power is 0.25 HP. The maximum speed is 2500 rpm. According to the data sheet, motor and control parameters have to be properly modified to run SVC.

Figure 13 A physical BLDC motor for parameter tuning

According to the motor data sheet, motor and control parameters can be properly modified through the header file of "customize.h". In the "customize.h", both motor 1 and motor 2 parameters are defined as following.

Motor Pole	4
Phase	3
Voltage	130 V
Current	1.5 A
Power	1/5 hp
Speed	2500 rpm
Inductance	27 mh
Stator Resistor	5.1Ω
Hall sensors	3

Table 3 Motor data sheet

First, define motor parameters:

- #define R_STA_CUSTOM 51 // stator phase resistance 5.1Ω /10
- #define POLES_CUSTOM 2 // 2 pair of poles

- #define I_START_CUSTOM 15 // startup current of 1.5A in Amps/10
- #define IQ_MAX_CUSTOM 45 // max iq current of 4.5A in Amps/10
- #define RPM_MIN_CUSTOM 500 // minimum motor speed of 500rpm
- #define RPM_MAX_CUSTOM 2500 // maximum motor speed of 2,500rpm

Second, modify control parameters related with hardware platform:

- #define PWM_FREQ_CUSTOM 10000 // PWM Frequency in 20,000Hz
- #define SAMPLE_FREQ_CUSTOM 10000 // Sample Frequency in 20,000 Hz
- #define DEAD_TIME_CUSTOM 2.0 //Switch dead time is 2. µS.

Last, tune control parameters:

- #define R_ACC_CUSTOM 1000 // acceleration ramp in 1000rpm/sec
- #define KP_CUR_CUSTOM 100 // proportional gain of current controller
- #define KI_CUR_CUSTOM 50 // integral gain of current controller
- #define KP_SPD_CUSTOM 300 // proportional gain of speed controller
- #define KI_SPD_CUSTOM 200 // integral gain of speed controller
- #define IRST_FLUX_LOWPASS_TIME_CUSTOM 10
- #define DERIVATIVE_TIME_CUSTOM 1
- #define LAST_FLUX_LOWPASS_TIME_CUSTOM 10
- #define FIRST_SPEED_LOWPASS_TIME_CUSTOM 5
- #define SECOND_SPEED_LOWPASS_TIME_CUSTOM 4
- #define THIRD_SPEED_LOWPASS_TIME_CUSTOM 3

9. MCU Performance Analysis

9.1 CPU Bandwidth

In order to better verify the performance of the developed SVC software described above, a series of experiments has been performed. Evaluations have been made in terms of CPU resources (especially bandwidth) used. Shown in *Figure* 14 is the CPU bandwidth for three shunt SVC at the 10 kHz PWM frequency. It is about 46.0% at10 kHz PWM frequency and 32MHz CPU.

RX111 MCU has an interrupt skipping function for the MTU2a timer. With this feature, the timer compare-match interrupt and underflow interrupt could be skipped several times in the complementary PWM mode. The interrupt skipping value depends on the PWM frequency and sample frequency defined in the header file of “customize.h”. If PWM_FRE_CUSTOM is not the same as SAM_FRE_CUSTOM, the INT_SKIP equals the ratio of these two variables. For instance, if the PWM frequency is set at 10 kHz and the control loop is run with 5 kHz, the PWM interrupt is skipped twice.

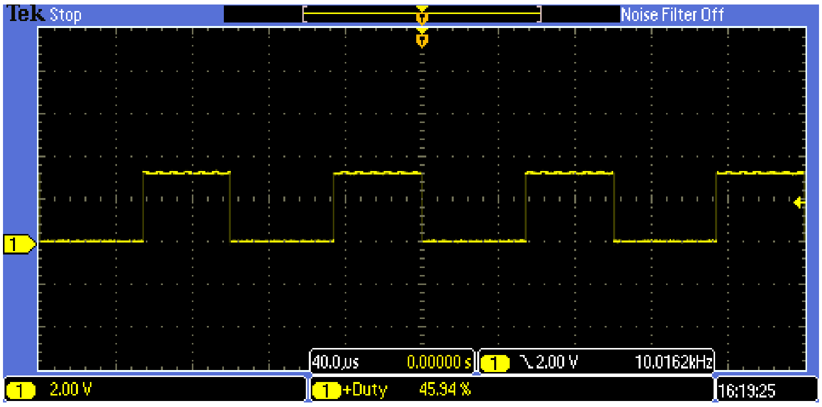

Figure 14 CPU bandwidth of RX111 SVC implementation

9.2 Testing Results

The RX111 SVC has been evaluated in terms of motor start current, speed responses and regulations without load and with some of load. The hardware platforms are described in Section 2. The motor uses a motor – 110V 3-phase Bordine BLDC motor in Section 8 with the DC bus voltage 160V DC.

The motor start current, the phase current at the steady state without load and with some of load are well perform and smooth rotation. The motor starts up with the open loop and the startup current is higher. Once the control switches into the closed loop, the motor current becomes the smaller in transition. And also, with no load, the motor current is very small but it increases with respect to the load applied. The test results demonstrate that Rx210 SVC exhibits very good field-oriented control and that the rotor position is estimated correctly.

When the load is applied or removed, The Rx210SVC handles the speed decrease or increase appropriately. The motor speed responds very quickly and maintains the reference speed. Therefore, the estimation of the angle and speed is accurate, and there is no loss of synchronization

Appendix A - References

1. RX111 Group User’s Manual: Hardware, Rev.0.50, October, 2012
2. Application Note of “Rx62T Three Shunt Sensorless Vector Control of PMSM Motor”, R01AN0903EU0100, Nov. 2011.
3. DevCon 2010 Courses:
   - ID-620C, Complete Motor Control Integration with RX62T.
   - ID-623C, Understanding Sensor-less Vector Control with Floating Point Unit (FPU) Implementation.
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
6. Huangsheng Xu, and Yashvant Jani, “Understanding Sensorless Vector Control for Brushless DC Motors”, ESC-2008, Embedded System Silicon Valley conference, April 15-17, San Jose, California, USA.

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	Dec. 09, 2013.	—	First 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