RL78/G1F Group

R30AN0264EU0100
Rev.1.00
Sept 23, 2016

Motor Start-up: Sensored versus Open Loop Sensorless

Introduction

When starting up a BLDC motor, it is required that you keep in sync with the rotor as it moves though it rotation. This is simple with sensors (typically HALL), but not quite as easy when no sensors are available. This application note compares a sensored start-up with an open-loop sensor-less start-up, comparing and contrasting the pros and cons of each, with specific emphasis on the start-up torque.

Target Device

RL78/G1F, 64 pin version (R5F11BLE)

Contents

1. Overview

This application note provides a comparison of sensored and open-loop sensorless start-up of a BLDC motor. It includes an accompanying demo program for a brushless DC motor (BLDC) using the RL78/G1F microcontroller. The demo program will be described from an "Abstract level", and will only detail the specific areas regarding HALL commutation and open-loop commutation. The reader is referred to the C project files and the Doxygen folder within the project for code documentation. For details on the motor algorithms themselves, please see references in section 11.

In this application note, wherever possible, we will use actual scope shots (or DSO data) rather than simulated or mathematically created data.

In addition, for this application note, we will use the terms "sensored" and "HALL" references almost interchangeably.

NOTE: Although this Application note was verified with a block wound BLDC motor, the theory and operation applies to a sinusoidal wound PMSM motor as well.

Operation checking device

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

- RL78G1F (R5F11BLE)

Target sample programs

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

(1) RL78G1F_START_UP for RL78G1F (R5F11BE) for T2001 Inverter

(Complementary PWM Positive Side, GPIO Negative Side, see section for details)

NOTE: This demo is used to show initial position determination, however, we clearly need to demonstrate the motor running from that position. The algorithm here is based on the CMP BEMF application note. See section 11, item 11.1.2

1.1 Development environment

Table 1-1 shows development environment of the sample programs explained in this application note.

Table 1-1 Development Environment of the Sample Programs

	Sample software	MCU	Inverter board	Motor	E2studio version	Tool Chain
Low-voltage version	(1)	R5F11BLE	T2001 Note 1	Portescap[2]	V5.1.0.02	IAR for RL78 V2.21.1.1833

For purchase and technical support of the low-voltage inverter board T2001, contact Sales representatives and dealers of Renesas Electronics Corporation.

Notes:

1. T2001 is a product of Desk Top Laboratories Inc.
2. Portescap Size 23 BDLC Motor, C-230012-20A
3. Some testing was done on an "off-the-shelf" BLDC power tool motor also.

2. System overview

Overview of this system is explained below.

2.1 Hardware configuration

The hardware configuration is shown below.

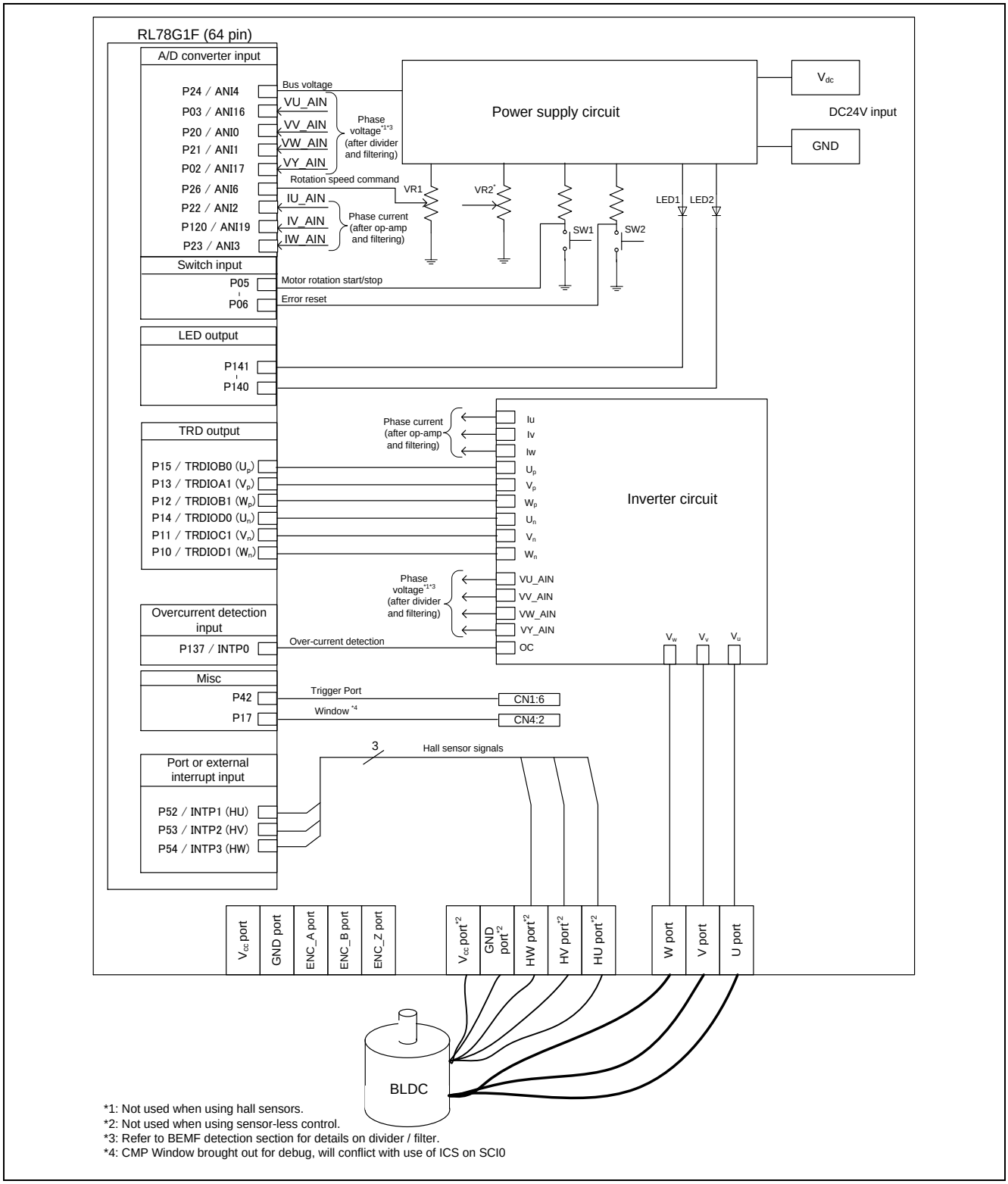

Figure 1: Hardware Configuration Diagram

2.1.1 Initial Position

When starting a BLDC motor, we must know the rotor position in order to produce torque in the correct direction. We will not dig into how this is accomplished on the sensorless since it is somewhat involved and suffice to say has its own Application note. We will only to define how we will do it for this particular application note.

Sensor-less

For sensor-less start-up we will use the Current Mode excitation. This method basically "pings" the windings (not long enough to move the rotor) and evaluates the resulting current which tells us where the rotor flux is positioned relative to the stator providing a 60º resolution of rotor position. The details are somewhat complicated and can be found in reference 11.1.3

HALL

When using HALL sensors, the position of the rotor is given by a direct reading of the (digital output) hall sensors though an I/O port, as they are valid whether the motor is rotating or not. They are strictly based on the position of the rotor flux relative to the 3 HALL sensors mechanically aligned with the stator, thus providing a resolution of 60º (the same as the sensorless Initial position code).

3. Controlling Program Description

In this section we will give an abstract view of the motor control program. For details on the program the reader is referred to the Doxygen output folder (index.html) for code structure, variables, definitions, and call/caller details.

3.1 System States

Figure 2 shows the system states as defined by a call to the Motor Event Function (`R_MTR_ExecEvent()`) and the events it can be called with.

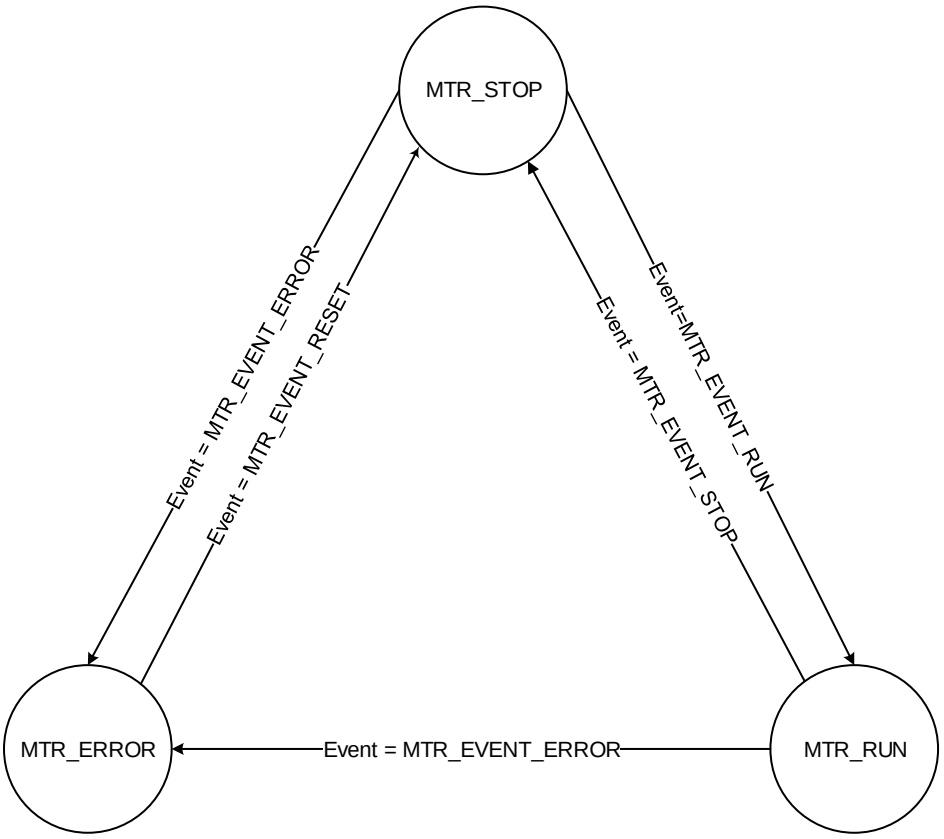

Figure 2: Motor Event States

A call to the function R_MTR_ExecEvent() also makes an indirect (function table) call to the default actions required by the transition into the new state.

For example, a MTR_EVENT_ERROR could be any number of items that indicate the motor or the system is not in a condition to run properly, including but not limited to Over-voltage on the VBus, Over-current while motor is running, and Over-speed.

NOTE: For the purposes of the initial position demo, we will only discuss the actions of the INTIAL POSITION functions in terms of being in the MTR_RUN state. For details on the full Motor Algorithm, please see section 0, item 11.1.2.

3.2 Motor States

Figure 3 and 7 show the motor states for the two algorithms in questions, Sensorless and HALL Algorithms respectively.. If you examine the diagram, you can see that after the initial position is determined, the HALL goes right into closed loop operation, whereas the Sensorless method must go into an Open-loop start up state

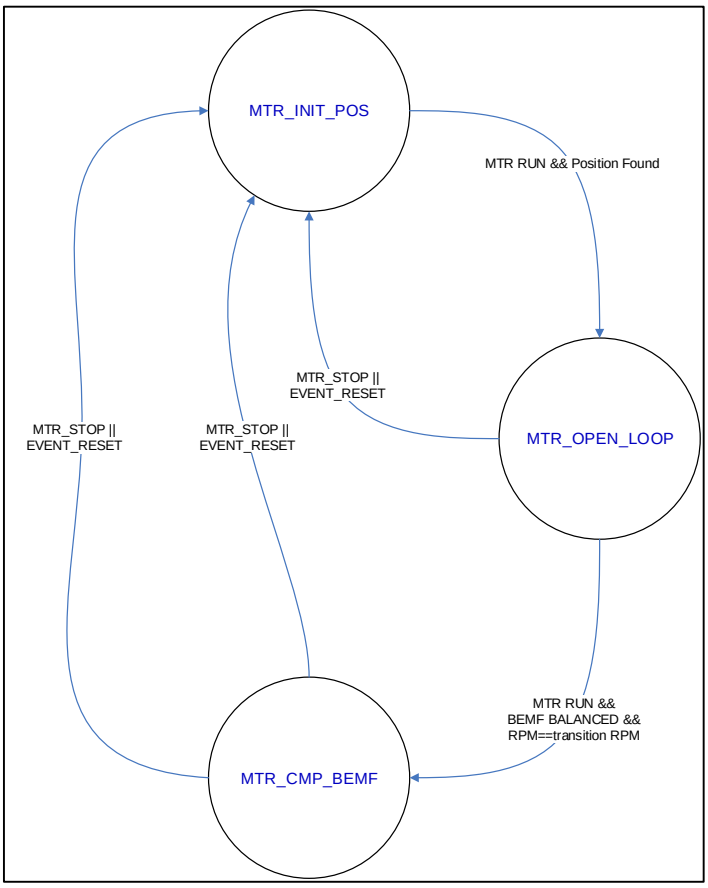

Figure 3: Motor State Diagram, Sensorless

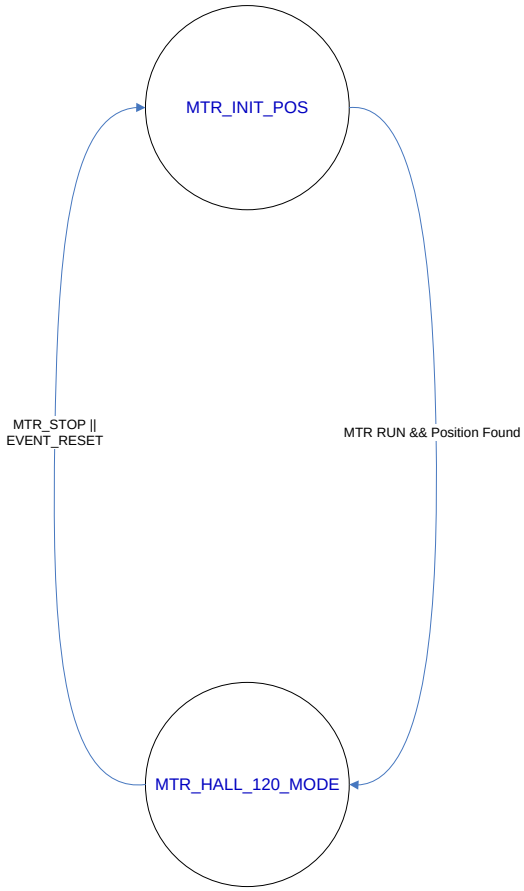

Figure 4: Motor State Diagram, HALL

Motor State transitions are a little harder to track, so a brief description of each.

MTR_INIT_POS: This is the starting state for the Motor operation. It is here after a MTR_EVENT_RESET. It remains here until the System State goes to the MTR_RUN mode at which time this state determines the initial position of the rotor. For the Sensorless algorithm, this is the current "ping" method and take some finite amount of time (typically in milliseconds). For the HALL based algorithm, we just read the value off the HALL sensors on GPIO ports, look this up in a table to translate to Motor STEP and we are ready to run.

Once the initial position of the rotor is determined without an error, the Sensorless transitions to the OPEN Loop state and start spinning the motor. The HALL algorithm can transition right into a closed loop operation.

MTR_OPEN_LOOP: In this state, the motor starts a pre-determined profile to spin the motor to a given RPM. For detail on transition to the CMP_BEMF mode, see the application note reference 11.1.2. *This state is only used by the Sensorless algorithm.*

MTR_CMP_BEMF: The motor stays here until a MTR_STOP event or a MTR_EVENT RESET. Stop is usually done by a commanded RPM of 0 or SW1 turned off. EVENT_RESET can be caused by any number of dynamic events, but most common is over-current. In this state the motor will commutate every time a Zero-cross (rotor position) is detected by the comparator circuit. *This state is only used by the Sensorless algorithm.*

MTR_HALL_OPER: The motor stays here until a MTR_STOP event or a MTR_EVENT RESET. Stop is usually done by a commanded RPM of 0 or SW1 turned off. EVENT_RESET can be caused by any number of dynamic events, but most common is over-current. In this state the motor will commutate every time a HALL sensor (rotor position) changes state. *This state is only used by the Sensorless algorithm.*

IMPORTANT: For the purposes of this START-UP demo program we will only be concerned with the MTR_OPEN_LOOP and the MOTOR_HALL_OPER. For actual motor operation within the algorithm, the reader is refer to section 0, item 11.1.2.

3.3 6-Step State Machine

We are using a trapezoidal drive, sometimes referred to as 6-step, see reference 11.1.5. The state machine for the 6-step is basically the same on the Sensorless and the HALL, with the exception of the state exit conditions which are described below. See Figure 5: Motor 6-Step States Diagram.

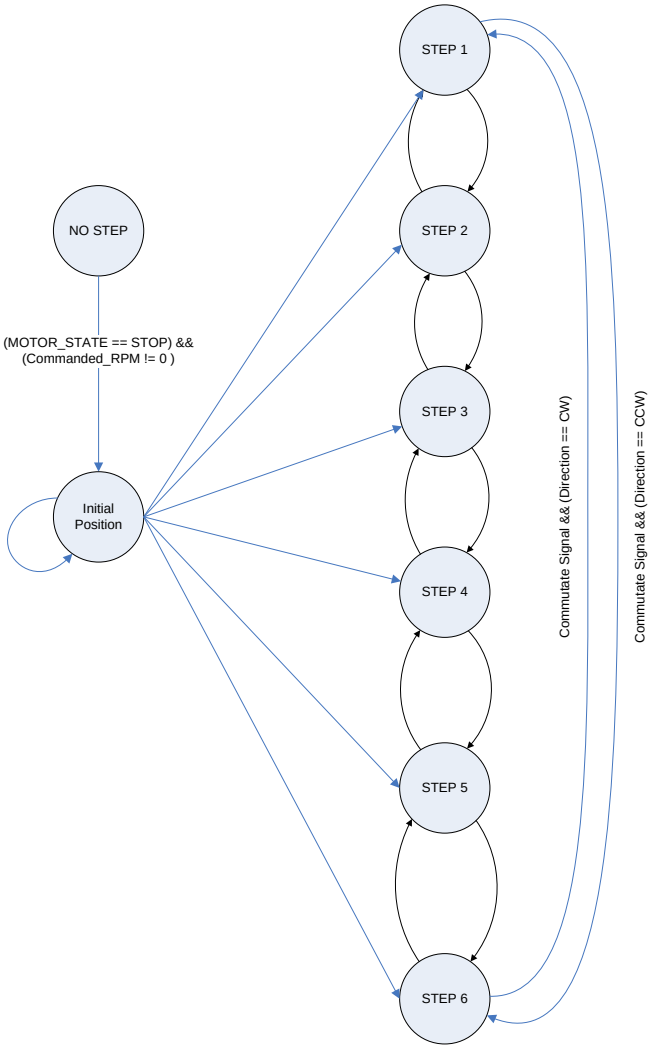

Figure 5: Motor 6-Step States Diagram

3.3.1 Sensorless Step Machine

The Sensorless algorithm exits each STEP state under one of two conditions.

OPEN-LOOP: The first is during open-loop start-up where a "finite" time (commutation time) has expired the state variable is updated and drive outputs are switched to the desired STEP value (next State in Step machine). The "finite" time is derived empirically based on the motor characteristics and the inertial load (system characteristics). The Open-loop commutation time is typically ramped at a rate or slope that corresponds with the load increasing in speed. *This is where the problem lies with Open-Loop start-up, it is typically done at a much higher torque than is required for the given speed so we can guarantee rotation / motion.* The State variable is updated based on the direction the motor is commanded to go.

CLOSED-LOOP: The second is during closed-loop or CMP mode operation where the rotor position is detected by a Zero-cross and the subsequent commutation timer has expired. The state variable is updated and drive outputs are switched to the desired STEP value. The "finite" time is derived empirically based on the motor characteristics and the inertial load (system characteristics). At this point, since we know the rotor position and we are driving in sync with it, the *corresponding current (torque) will typically go way down*. The State variable is updated based on the direction the motor is commanded to go.

3.3.2 HALL Step Machine

CLOSED-LOOP: The based code can typically co right into closed loop, where the rotor position is detected by transitions (state changes) on the HALL sensors. In this case the State variable is actually determined by the HALL values which tell us which state we have entered during the rotation. The state variable is updated and drive outputs are switched to the desired STEP value. This is the major contrast to open-loop start-up, we know where the rotor is, so we can fundamentally drive the torque (current) at the required level to maintain speed and torque during start-up.

NOTE: If you are closing a torque loop, you can actually drive at a low value and let the torque loop increase until you see motion or reach speed.

4. Running the Selected Code

The project will run a 6-step algorithm by either BEMF by comparator or HALL sensors. From an external standpoint both algorithm will functionally operate the same way. See section 10.2 for selecting the proper algorithm that you wish to evaluate the Start-up on. The controls are as follow to get the motor started:

S1 – enables the motor to run.

Potentiometer

- Center detent is 0 RPM,
- CW on the potentiometer increase the RPM in the CW direction. You must select higher than the MIN RPM to run the motor.
- CCW on the potentiometer increase the RPM in the CCW direction.

5. Monitoring Start-up Currents (torque)

The key point of this application note is the comparison of the two types of start-up. To this end we will compare the operating currents during the initial time when the motor starts up and reaches a stable operating speed. Although we could use the current shunts in the Inverter platform and re-construct from capture date, it is easier to just view with Current Clamp probes and a DSO. Our set-up for monitoring start-up current is shown in Figure 6.

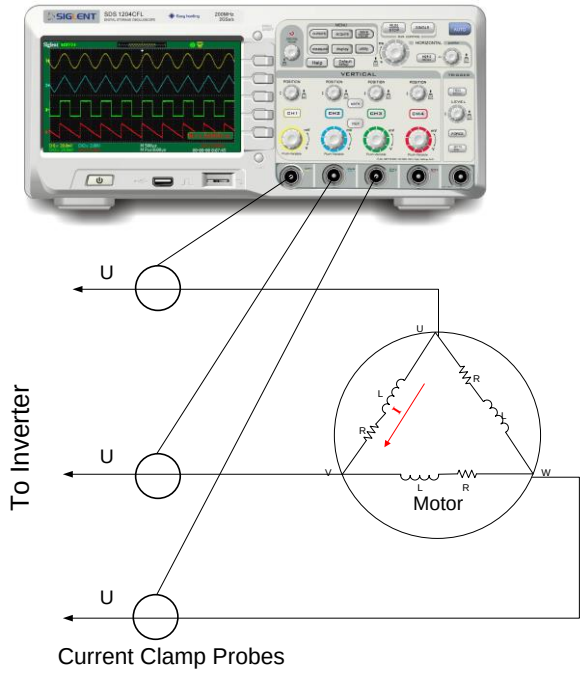

Figure 6: Motor current monitoring set-up

5.1 Open-Loop

Connect the scope as shown in Figure 6: Motor current monitoring set-up. Set the scope to trigger on a change in in current (rise or fall is OK since the current will go in both directions depending on the STEP value. Your current should look similar to that in Figure 7. You will note that the current is almost double (4.0A) at about 1000 RPM versus about 2.0A at 300 RPM (Figure 8) when running closed loop.

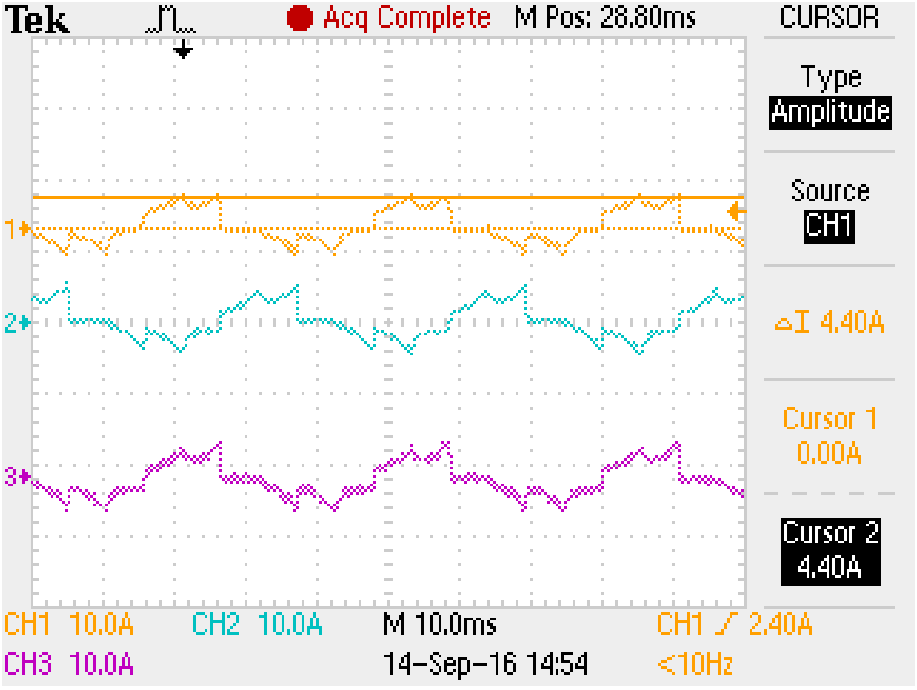

Figure 7: Open loop Start-up current

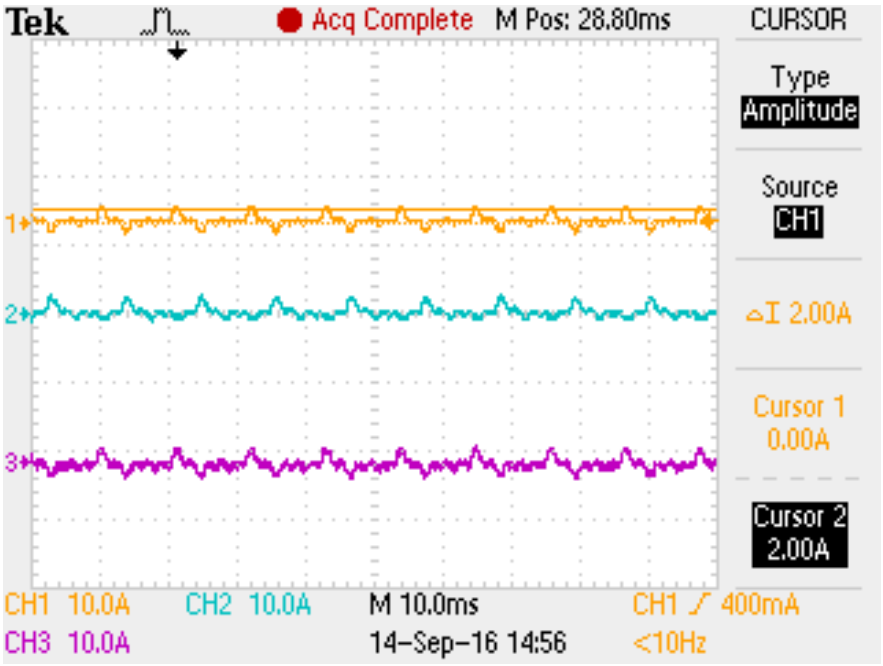

Figure 8: Closed loop Sensor-less

5.2 HALL Based

Connect the scope as shown in Figure 6: Motor current monitoring set-up. Set the scope to trigger on a change in in current (rise or fall is OK since the current will go in both directions depending on the STEP value. Your current should look similar to that in Figure 9.

NOTE: The current bump at the beginning is the current used to overcome the "stiction". This was not "tuned" in this demo and is arbitrarily large. It can be reduced based on the motor.

You will notice that once we overcome the "stiction" of the resting rotor, the current quickly goes to about 400mA, much less than the open loop start-up current. Also it should be noted that unlike the Sernsor-less, you can almost immediately close the speed and torque loops *if required* since you have position and speed data after the first couple of HALL "ticks".

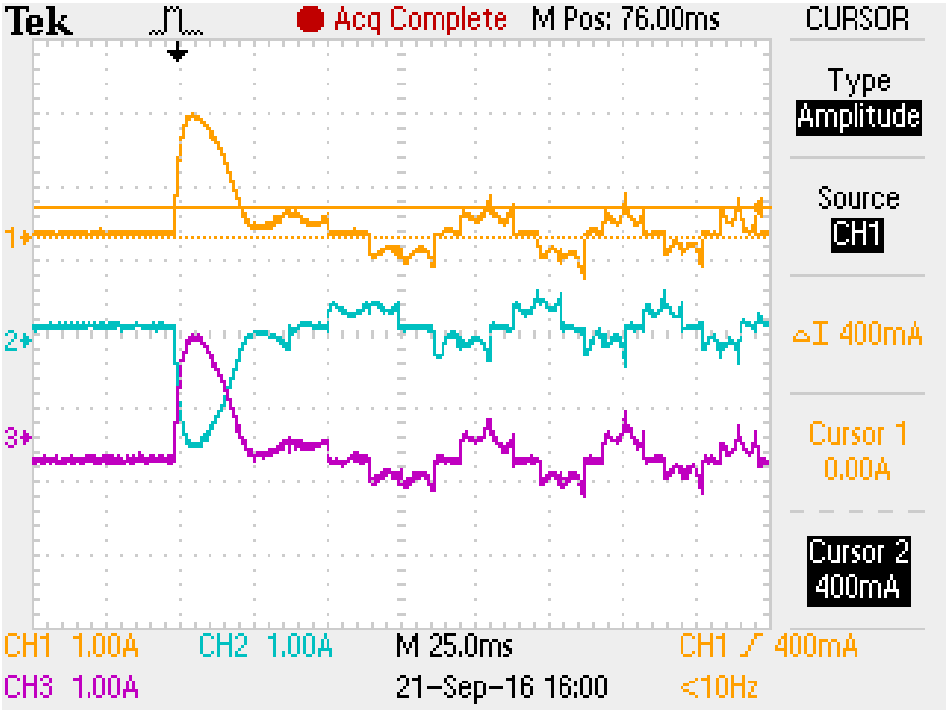

Figure 9: HALL based Start-up Current

6. Pros and Cons of Open-Loop Start-up

6.1 Pros

- Low cost, removes sensors and the associated wires
- 3 less pins utilized on MCU
- Higher hardware reliability (i.e. no sensors or wiring to fail)

6.2 Cons

- more difficult start-up procedure (empirically derived usually)
- a little more difficult to detect “stuck rotor” in open loop mode.
- may require second timer to commutate based on Zero-cross detection

7. Pros and Cons of HALL Based Start-up

7.1 Pros

- Easier start-up procedure, can close loops immediately for control
- Easy to detect “stuck rotor” even on start-up.
- No extra timer for commutation, HALL signal is commutation point[1]

7.2 Cons

- higher cost associated with HALL sensors and the associated wires
- lower hardware reliability (i.e. sensors and/or wiring to fail). In addition, the HALL sensors are typically in the motor where they are exposed to higher temperatures.
- 3 more pins utilized on MCU

8. Summary

Open-loop start up can save you money and increase reliability with its reduction in the components and wiring related to HALL sensors, but has the accompanying issues with producing torque during the spin-up process. It is still quite good solution where loads are somewhat predictable, for example, fans and RC propeller motors.

In applications where position and high torque are critical, the HALL sensors and the associated “ease of start-up “ still find a home, for example Motion control and traction applications..

[1] Unless doing phase advance which may require timer channel.

9. Miscellaneous Operations related to Motor Control

9.1 Over-current (INTP0 operation for this app note)

The demo is currently set-up to use the INTP0 signal as the overcurrent indicator. This is useful in the event you over-drive during alignment or Current Excitation for finding initial position. The platform that this was tested on was multi-shunt with the proper "window" comparator built onto the inverter. See references in section 11, item 11.1.3

10. Demo Project

10.1 Importing and Building

1) Open the e2studio workspace you want project located in
2) Use the standard Import Feature of e2studio to Import an Existing Project
3) Browse (root directory) and select the sub-directory *Workspace\RL78G1F_START_DEMO* where you unzipped the file.
4) Select *RL78G1F_ START_DEMO* and FINISH.
5) If asked, Browse to your IAR install location and click APPLY, then OK
6) Project should look like Figure 10:

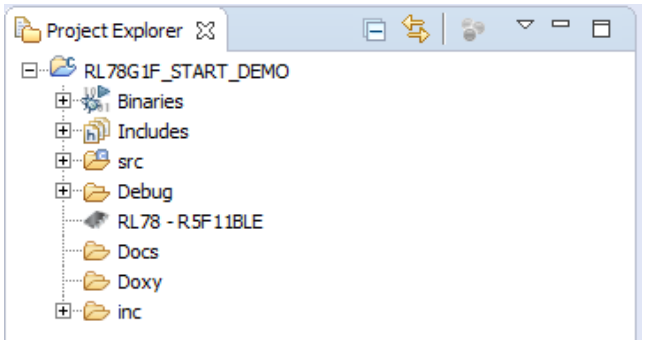

Figure 10: Project Explorer Panel after Import

7) Select menu Project Build ➔All (or ).

NOTE: there may be some warnings due to the use of volatiles in the program.

10.2 Algorithm Selection

The project will be built with the selected algorithm by uncommenting the #define at line 72 in mtr_common.h.

```
// Algorithm Option
//#define HALL_COMMUTATION
```

10.3 Motor Selection

The project is built with parameters derived for the Portescap Motor. We have provided a means of switching to other motors. USER_MOTOR can be used when tuning for your particular motor, just change the #define in mtr_common.h and search on USER_MOTOR for the areas that may need to be tuned / changed.

```
// Motor Options
#define USER_MOTOR 0
#define PORTESCAP 1
#define OTS_PT_MOTOR 2
// set the motor used here
#define MOTOR_USED  PORTESCAP
```

10.4 Tips and Tricks

10.4.1 Open-Loop "tuning"

Although this application note is not really about the algorithms and more about actually producing the initial torque, there is something that bears discussion. Open loop start-up is typically empirically derived. The values we use for the Portescap motor can be found starting at line 79 of mtr_ssa24d_less_120.h of this project

We have exaggerated the open-loop start (lower its ramp rate) to make it more visible and apparent while running this application note. Even in open loop start-up mode, the motor can start faster, but will still require larger currents to guarantee rotation.

10.4.2 Expressions

Most of the State Machines use enumerated values for their state variables, so it makes it easy to "watch" these in the expressions window to determine the whole state of your "machine". For example, in Figure 11, from our Auto-refresh variable (large R indicator) we can see that:

- Our motor run mode is Open Loop Mode
- The current RPM is 1042
- We are in STEP 4.

It should be noted, that the real time refresh rate is typically in the 100's of milliseconds, so this window cannot always be used for troubleshooting real time values such as motor current sampling, etc. Some other method such as streaming connection to the software may be required.

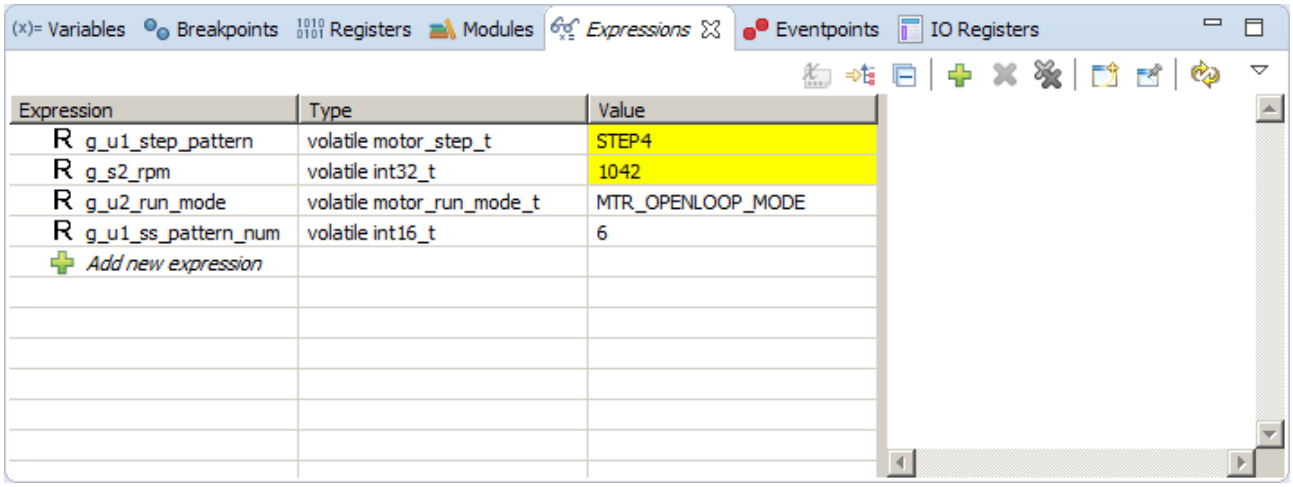

Figure 11: Real time refresh Expressions

10.4.3 Doxygen

In the Doxy folder under the project, the CALLER/CALL graphs, definitions, variables, files structure, etc. can be browsed by clicking on the index.html file under the Project Explorer panel (or maybe opened outside in a conventional Web browser).

After changes are made to the software, you can update the Doxygen files by re-running the wizard and loading the Doxyfile in the Doxy directory.

11. References

The following document were used in the creation of this application note

11.1.1 RL78G1F Group User's Manual: Hardware (R01UH0516EJ0050)

11.1.2 R02AN0227EU0100 – Application Note: 6-Step Control of BLDC by Comparator Zero-cross detection.

11.1.3 R02AN0249EU0100 – Application Note: Initial Position of Sensorless BLDC Motor.

11.1.4 R01AN2657EJ0100 - Application note: 120-degree conducting control of permanent magnetic synchronous motor: algorithm

11.1.5 BLDC Algoithrms and Theory: https://www.renesas.com/en-us/solutions/key-technology/motor-control/motor-algorithms/bldc.html

11.1.6 Applilet for RL78/G14 V1.01.01, used for sample TAU one-shot code generation

11.1.7 Trial series "T2001" 50W 60VA Low Voltage Inverter Unit User's Manual

11.1.8 http://www.electricaltechnology.org/2014/09/comparison-between-star-and-delta-connections.html

12. Appendix A

12.1 Glossary

BEMF – Back Electromotive Force

BLDC Motor – brushless DC motor (typically requires electronic commutation)

DSO – Digital Signal Oscilloscope

MCU – Microcontroller Unit

MSO – Mixed Signal Oscilloscope

PM – Permanent Magnet

PWM - Pulse Width Modulation

TAU – Timer Array Unit

Website and Support

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

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

Revision History

		Description	
Rev.	**Date**	**Page**	**Summary**
1.0	09/23/2016		Initial Release

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.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.
   Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

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.
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, HALII 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 5.0