Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.

2. 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.

3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.

4. 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.

5. When exporting the 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. You should not use Renesas Electronics products or the 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. 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.

6. 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.

7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. 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 categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. 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 an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics.
The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

“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.

“High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

“Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.

8. 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.

9. 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 system manufactured by you.

10. 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.

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.
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 manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins
   Handle unused pins in accord 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 flow 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 one with a different part number, confirm that the change will not lead to problems.
   - The characteristics of MPU/MCU in the same group but having different part numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different part numbers, implement a system-evaluation test for each of the products.
Table of Contents

Notice 2

General Precautions in the Handling of MPU/MCU Products ............ 4

Chapter 1 Overview .......................................................................................... 7
  1.1 Abstract ........................................................................................................ 7
  1.2 Conditions .................................................................................................... 7
  1.3 Introduction .................................................................................................. 7

Chapter 2 3-Phase BLDC Motor Basics ..................................................... 8

Chapter 3 3-Phase Inverter Topology ......................................................... 10
  3.1 General Inverter Topology ....................................................................... 10
  3.2 Intelligent Power Device for MOSFET Pre-drive (R2A25104KFP) .......... 11
    3.2.1 Description .............................................................................................. 11
    3.2.2 Features ..................................................................................................... 11
    3.2.3 Structure of R2A25104KFP ....................................................................... 12
  3.3 RL78/G14 Specific Inverter Topology .................................................... 13

Chapter 4 Basics of Trapezoidal Commutation (6-step) ............... 14
  4.1 Controlling Phase Voltage ..................................................................... 14
  4.2 Rotor Position ............................................................................................ 16
  4.3 Commutating .............................................................................................. 17

Chapter 5 Implementation ............................................................................ 18
  5.1 System configuration ............................................................................. 18
  5.2 Configuration of Target Board R5F104LE-TB ..................................... 19
  5.3 Motor Configuration ............................................................................... 19
  5.4 Peripherals I/O Assignments .................................................................. 19

Chapter 6 Hardware Configuration ......................................................... 22
  6.1 Motor Control Timer RD ......................................................................... 22
  6.2 Timer Array Unit ...................................................................................... 24

Chapter 7 Software Configuration ............................................................. 25
  7.1 Commutation control ............................................................................ 26
  7.2 Motor starting ........................................................................................... 27

Chapter 8 Software Module Descriptions ............................................. 28
  8.1 Main Routine ............................................................................................ 28
  8.2 INTP1, INTP2, INTP3 Interrupt Service Routine ............................... 29
  8.3 Interval Timer Interrupt Service Routine ............................................. 31
Chapter 9  Program Code (motor driving part only) ..........................32
  9.1  Main Routine..................................................................................32
  9.2  INTTM00 Interrupt Service Rountine............................................34
  9.3  Interrupt Service Routines of external Interrupt INTP1, INTP2, INTP3......35

Chapter 10  Principle Circuit and Schematics........................................36

Chapter 11  Revision History .................................................................41
Chapter 1  Overview

1.1  Abstract

This Application Note shows the implementation of 3-phase BLDC motor drive by 120-degree trapezoidal wave commutation using Renesas MOSFET pre-driver R2A25104KFP and Motor Control Timer RD. The method shown utilizes the HALL sensors in the motor to determine the motors rotor position.

This example applies to MCUs in the RL78/G14, *RL78/F13, *RL78/F14 Group.

* RL78/F13, RL78/F14 are under development.

1.2  Conditions

The explanation of this issue is applied to the following condition:

Applicable MCU: RL78/G14 Group (Device R5F104LE-TB)
MCU operational frequency: 32 MHz
Memory size: ROM 64 KB, RAM 5,5 KB
MCU Supply Voltage: 5V
Peripherals: Timer RD for PWM motor drive

MOSFET Inverter pre-driver: R2A25104KFP

1.3  Introduction

Brushless DC motors (BLDC-motors) are controlled using a three phase power semiconductor inverter bridge. The motor controller requires a rotor position feedback signal for providing the proper commutation sequence to the motor via the power devices in the inverter bridge. Hall sensors are typically used to provide the rotor position feedback.

In this application note we will take a look at the RL78/G14 Group driving a 3-phase BLDC motor using the 120° trapezoidal method, also referred to a 6-step. The 6-step method is one of the simplest methods for driving 3-phase BLDC motors. In the past it was done by using discrete logic gates but with the more powerful peripherals available in today’s microcontrollers, such as the RL78 family, we can provide more functionality, better energy usage and higher safety level when driving motors.

The inverter bridge is pre-driven by an intelligent Power Device R2A25104KFP.
Chapter 2  3-Phase BLDC Motor Basics

The structure of the BLDC motor is divided into two parts: The moving part, the rotor, consists of a permanent magnet, and the fixed part, the stator, consisting of phase windings for generating an electromagnetic field.

Figure 1  3-Phase BLDC motor with one pole pair permanent magnet

The rotor is attracted to the electromagnetic field caused by the energized stator windings and starts to rotate. With an appropriate electromagnetic rotating field on the stator, the rotor follows the electromagnet poles and a rotation of the permanent magnet is created and maintained.

The number of the permanent magnet poles can vary. A greater number of poles creates a greater torque for the same current level. For this Application Note a motor with 10 poles is used which means 5 pole pairs (2 poles = 1 pole pair). It defines also the ratio between electrical and mechanical revolution of the motor, which is described in Figure 2.

Figure 2  Relationship between the electrical and mechanical revolution of the motor (5 pole pairs)
The relationship between the electrical and mechanical revolution is given in equation (1).

\[ \omega_e = p \ast \omega_m \]  

(1)

In this equation, \( p \) is the number of pole pairs of the motor, \( \omega_e \) the electrical speed and \( \omega_m \) the motor speed or speed of the rotor. For example, a BLDC motor with 5 pole pairs comes to following equation (2).

\[ \omega_m = \frac{\omega_e}{5} \]  

(2)

This means the electromagnetic field in the stator has to rotate five times to produce one mechanical turn of the rotor as shown in figure 2 above.

The rotor position, which is needed for the synchronization between rotor and rotating field, can be detected using 3 Hall sensors (sensored mode) or Back-EMF detection of the motor (sensorless mode).

In this Application Note we use the sensored mode in which the Hall sensors H1, H2, H3 in the motors are displaced by 120° to each other.

The lack of the brushes in the BLDC motor requires electronic commutation to produce the rotating field on the stator. An external circuit, known as an Inverter circuit, performs the electronic commutation. The topology and the function of the Inverter are described in the next section.
Chapter 3  3-Phase Inverter Topology

3.1  General Inverter Topology

Figure 4 shows the basic Inverter topology for driving 3-phase motors. In case of this application note the control electronics is the RL78/G14. It is important to note that it requires 6 outputs to drive the Inverter section. Timer RD in the RL78 family is perfectly suited for this task and supports “dead-time” to prevent “shoot-through” current on the MOSFETs (figure 3).

If Inverter is build on MOSFETs it is necessary to pre-drive the gates, because every MOSFET has an oxide-insulated gate electrode which builds a capacitor into the transistor. At every changing process this internal capacitor has to be reloaded as fast as possible. For this task the intelligent pre-driver R2A25104KFP suits ideally. The main parts of the R2A25104KFP are described in the next section.

Figure 4 Three-Phase Inverter and motor connections
The circuit contains 6 MOSFETs connected to the high and low side of the DC supply. UT, VT and WT are the high side (upper) transistors and UB, VB and WB are the low side (lower) transistors of the circuit.

3.2 Intelligent Power Device for MOSFET Pre-drive (R2A25104KFP)

3.2.1 Description

R2A25107KFP device is an intelligent Power Device to pre-drive the FET inverter used to drive the three phase brushless motor. This device contains three sets of pre-drivers and is applicable to be used in systems using supply voltages between 7V and 36V.

The integrated circuit contains step down converter, charge pump circuit for the power supply of upper side FET drive, 5V series regulator, watchdog timer, protection circuits as thermal shut down circuit (TSD) and over current detection circuit.

3.2.2 Features

- Wide range operating voltage; 7V to 36V (VBAT, VBAT2)
- On-chip Three phase pre-driver circuit
  - PWM control; up to 20KHz
  - Totem pole type MOSFET gate drive circuit
- On-chip power supplies
  - Step down converter; 6.2V typ.
  - Charge pump circuit for power supply of upper side FET drive; >5V
  - 5V series regulator for MCU; <70mA
- On-chip protections
  - Thermal shut down circuit (TSD)
  - Over current of step down converter
  - Over current detection of motor current
- Watchdog timer circuit (WDT)
- Bandgap reference circuit
- Internal oscillation circuit; 265KHz typ
- 48pin LQFP package
3.2.3 Structure of R2A25104KFP

Figure 5  Block Diagram of R2A25104KFP
3.3 RL78/G14 Specific Inverter Topology

Figure 6 shows the specific architecture of the motor control board connected to the RL78/G14.

Figure 6  Block Diagram of RL78/G14 Inverter Control

Note: Shows the pre-driver part only

The Gates of the upper MOSFETs are modulated by Timer RD. The low site MOSFETs are driven by general I/O port (upper Modulation).
Chapter 4  Basics of Trapezoidal Commutation (6-step)

The 6-step method is one of the simplest methods for driving 3-phase BLDC motors. It is also known as 120-Degree trapezoidal, since it drives each winding for 120-degrees of the electrical rotation and leaves the winding un-driven for 60 degrees. The phase currents of trapezoidal commutation are shown in figure 7.

Please note, although the drive method is simple, this lack of drive for 60 degrees also results in higher torque ripple in the end application. The system designer must decide if this is acceptable or other drive methods should be considered.

Figure 7  Phase Currents of Trapezoidal Commutation

![Figure 7](image_url)

4.1  Controlling Phase Voltage

In chapter 3.3 we showed how we will connect the microcontroller to the power inverter stage to control the gates of the MOSFETs. The speed of the motor is directly proportional to the applied voltage. To vary the speed, one or both of the transistors in the excited pair can be pulse width modulated to produce a variable average DC voltage. Maximum speed is achieved when both transistors are on for 100% of the commutation period so that the full supply voltage is applied to the motor.

Typically modulation techniques are Upper modulation, Lower Modulation, Rotating Modulation or Balanced Modulation. For this application note we will be showing Upper modulation only. Figure 8 shows the basic Upper modulation waveforms. Please note that in this method only the "T" or upper MOSFETs are modulated. UT, VT, WT are the upper transistors and UB, VB, WB are the low site transistors which are set by general purpose ports. U, V, W are the resulting phase voltages.
NOTE: Timer RD in the RL78 Family can do any of the modulation techniques and is not limited to single-sided modulation techniques. This would include sinusoidal modulation for other motor types such as 3-phase induction.
4.2 Rotor Position

We can control voltage on the windings (and indirectly the current) by using a PWM timer, but we must present these signals in the appropriate sequence to properly commutate the motor. In order to do this we must know the rotor position. We will achieve this by the Hall sensors which sense the position of the rotor by using the Hall Effect (figure 9 left site). The sensors can do this because they are positioned relatively to each motor phase winding in the stator coils (see figure 9 right site). Figure 10 shows a typical HALL cell signature. Please note that the state changes every 60° for one electrical cycle. Interrupt input pins of RL78 allow us to read the hall signal and the control software decode them into a 60° rotor position.

In the below figure a motor with 1 pole pair is used, whereas the motor in this application note has 5 pole-pairs.

---

Figure 9 Hall Effect

![Hall Effect Diagram](image)

Figure 10 Typical Hall Signature

![Typical Hall Signature Diagram](image)
4.3 Commutating

So now we can control the voltage, we know where the rotor is so let’s put them together.

A motor manufacturer’s data sheet will typically tell you “…when you see this HALL signature, drive these phase windings”. Figure 11 shows a typical commutation sequence. For this motor, when we see the HALL signature for STEP1, we drive UT and VB. The rotor will move because torque is being caused by the magnetic fields in the stator coils at the correct angle to the magnets on the rotor. When we see the Hall signature change state to indicate Step 2, we switch commutation sequence from UT and VB to UT and WB, and the rotor will continue to move.

This will continue for the entire cycle until we are back at step one and the process repeats.

Figure 11 Typical Commutation Sequence

<table>
<thead>
<tr>
<th>Transistor</th>
<th>Graphic</th>
<th>Angle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>UT VT WT</td>
<td>0° - 60°</td>
</tr>
<tr>
<td></td>
<td>UB VB WB</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>UT VT WT</td>
<td>60° - 120°</td>
</tr>
<tr>
<td></td>
<td>UB VB WB</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>UT VT WT</td>
<td>120° - 180°</td>
</tr>
<tr>
<td></td>
<td>UB VB WB</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>UT VT WT</td>
<td>180° - 240°</td>
</tr>
<tr>
<td></td>
<td>UB VB WB</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>UT VT WT</td>
<td>240° - 300°</td>
</tr>
<tr>
<td></td>
<td>UB VB WB</td>
<td></td>
</tr>
<tr>
<td>Step 6</td>
<td>UT VT WT</td>
<td>300° - 360°</td>
</tr>
<tr>
<td></td>
<td>UB VB WB</td>
<td></td>
</tr>
</tbody>
</table>

= Transistor active
Chapter 5 Implementation

5.1 System configuration

As you can see the MCU is supplied by 5V-Relulator of intelligent pre-driver R2A25104KFP. Furthermore overcurrent detection is done by this device. The external Watchdog Timer is used for software shut down. Position detection is done by Hall sensors connected to external interrupt inputs of RL78.
5.2 Configuration of Target Board R5F104LE-TB

The R5F104LE-TB device is a member of the high performance RL78 Family 16-bit microcontrollers. The configuration of the device and the operating environment used in this application is listed below:

- CPU: R5F104LE
- Operating Clock: 32MHz
- Supply Voltage: 5V
- Internal ROM: 64 KB
- Internal RAM: 5,5 KB

5.3 Motor Configuration

- Max. Speed: 4000 rpm
- Max. Voltage: 24V
- Poles: 10 (5 pole-pairs)

5.4 Peripherals I/O Assignments

Table 1 lists all the pins of the R5F104LE-TB device and the ones used in this application are described with their associated function.

<table>
<thead>
<tr>
<th>PIN No.</th>
<th>PIN Name</th>
<th>Mode Setting</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>P120/ANI19</td>
<td>Not used</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>P43</td>
<td>Not used</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>P42</td>
<td>Not used</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>P41</td>
<td>Not used</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>P40/TOOL0</td>
<td>TOOL0</td>
<td>TOOL0 (Target board)</td>
</tr>
<tr>
<td>6</td>
<td>RESET</td>
<td>RESET</td>
<td>RESET-Input</td>
</tr>
<tr>
<td>7</td>
<td>P124/XT2/EXCLKS</td>
<td>XT2</td>
<td>Subsystem Clock (Target board)</td>
</tr>
<tr>
<td>8</td>
<td>P123/XT1</td>
<td>XT1</td>
<td>Subsystem Clock (Target board)</td>
</tr>
<tr>
<td>9</td>
<td>P137/INTP0</td>
<td>INTP0</td>
<td>Test switch (Target board)</td>
</tr>
<tr>
<td>10</td>
<td>P122/X2/EXCLK</td>
<td>X2</td>
<td>Main System Clock (Target board)</td>
</tr>
<tr>
<td>11</td>
<td>P121/X1</td>
<td>X1</td>
<td>Main System Clock (Target board)</td>
</tr>
<tr>
<td>Pin</td>
<td>Description</td>
<td>Pin</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>----------------------</td>
<td>------</td>
<td>----------------------</td>
</tr>
<tr>
<td>12</td>
<td>REGC</td>
<td>13</td>
<td>VSS</td>
</tr>
<tr>
<td>14</td>
<td>EVSS</td>
<td>15</td>
<td>VDD</td>
</tr>
<tr>
<td>16</td>
<td>EVDD</td>
<td>17</td>
<td>P60/SCLA0</td>
</tr>
<tr>
<td>18</td>
<td>P61/SDAA0</td>
<td>19</td>
<td>P62/SSi00</td>
</tr>
<tr>
<td>20</td>
<td>P63</td>
<td>21</td>
<td>P31/TI03</td>
</tr>
<tr>
<td>28</td>
<td>P71/KR1/SI21/SDA21</td>
<td>29</td>
<td>P70/KR0/SCK21/SCL21</td>
</tr>
<tr>
<td>30</td>
<td>P06</td>
<td>31</td>
<td>P05</td>
</tr>
<tr>
<td>32</td>
<td>P30/INTP3/RTC1HZ/SCK00/SCL00/TRJO0</td>
<td>33</td>
<td>P50/INTP1/SI00/RxD0/TOOLR0/SDA00/TRGIOA</td>
</tr>
<tr>
<td>34</td>
<td>P51/INTP2/SO00/TxD0/TOOLTxD/TRGIOB</td>
<td>35</td>
<td>P52</td>
</tr>
<tr>
<td>36</td>
<td>P53</td>
<td>37</td>
<td>P54</td>
</tr>
<tr>
<td>38</td>
<td>P55</td>
<td>39</td>
<td>P17/TI02/TO02/TRDIOAO/TRDCLK0</td>
</tr>
<tr>
<td>40</td>
<td>P16/TI01/TO01/INTP5/TRDIOC0</td>
<td>41</td>
<td>P15/SCK20/SCL20/TRDIOBO</td>
</tr>
<tr>
<td>42</td>
<td>P14/RxD2/SI20/SDA20/TRDIOB0</td>
<td>43</td>
<td>P13/TxD2SO20/TRDIOA1</td>
</tr>
<tr>
<td>44</td>
<td>P12/SO11/TRDIOB1</td>
<td>45</td>
<td>P11/SI11/SDA11/TRDIOC1</td>
</tr>
<tr>
<td>46</td>
<td>P10/SCK11/SCL11/TRDIOB1</td>
<td>47</td>
<td>P146</td>
</tr>
<tr>
<td>48</td>
<td>P147/ANI18</td>
<td>49</td>
<td>P27/ANI7</td>
</tr>
<tr>
<td>50</td>
<td>P26/ANI6</td>
<td>51</td>
<td>P25/ANI5</td>
</tr>
<tr>
<td>52</td>
<td>P24/ANI4</td>
<td>53</td>
<td>P23/ANI3</td>
</tr>
<tr>
<td>54</td>
<td>P22/ANI2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pin</td>
<td>Description</td>
<td>Status</td>
</tr>
<tr>
<td>---</td>
<td>-----</td>
<td>-------------------------------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>55</td>
<td>P21/ANI1/AVREFM</td>
<td>Not used</td>
<td></td>
</tr>
<tr>
<td>56</td>
<td>P20/ANI0/AVREFP</td>
<td>Not used</td>
<td></td>
</tr>
<tr>
<td>57</td>
<td>P130</td>
<td>Not used</td>
<td></td>
</tr>
<tr>
<td>58</td>
<td>P04/SCK10/SCL10</td>
<td>Not used</td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>P03/ANI16/Si10/RxD1/SDA10</td>
<td>Not used</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>P02/ANI17/SO10/TxD1</td>
<td>Not used</td>
<td></td>
</tr>
<tr>
<td>61</td>
<td>P01/TO00/TRGCLKB/TRJIO0</td>
<td>Not used</td>
<td></td>
</tr>
<tr>
<td>62</td>
<td>P00/TI00/TRGCLKA</td>
<td>P00/Output external WDT Clock</td>
<td></td>
</tr>
<tr>
<td>63</td>
<td>P141/PCLBUZ1/INTP7</td>
<td>Not used</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>P140/PCLBUZ0/INTP6</td>
<td>Not used</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 6  Hardware Configuration

6.1 Motor Control Timer RD

The timer used in the RL78/G14 for 3-phase complementary PWM is Timer RD, as shown in Figures 13 and 15. In Complementary PWM mode, it offers three high-side and three low-side PWM outputs, each with adjustable dead-time between the upper and lower pairs. These outputs are connected to the gates of the Inverter MOSFET via the pre-driver (figure 14).

Figure 13  Register Diagram of Timer RD in Complementary PWM Mode

Figure 14  Connection Diagram of Timer RD
The timer is 16-bit, allowing any value up to that point for a peak value. We use a PWM maximum (TRDGRA register value) of 1662, which translates to 3324 clocks per PWM period. That count value provides a 20 kHz PWM frequency. This frequency is chosen because it is outside the audible frequency range (> 16 kHz) and on the other hand it does not cause too high switching losses. The timer counter register, TRD0 counts up to the value of TRDGRA, then turns around and counts down until another counter, TRD1, reaches zero, then it counts up again. The difference between the two counters is the dead-time value which is preloaded before the timer is started (figure 15).

![Output Model of Complementary PWM Mode](image)

**Figure 15** Output Model of Complementary PWM Mode

Remark: \[ I = 0 \text{ or } 1 \]
6.2 Timer Array Unit

The Timer Array Unit is used to realize a 16-bit **Interval Timer** function. This Interval Timer generates a fixed interrupt named INTTM00 every 5 ms. This Interrupt is used to generate a predefined commutation sequence for motor starting ([Chapter 7.2](#)).

![Figure 16 Interval Timer operation of the Timer Array Unit](image-url)
Chapter 7  Software Configuration

Required resources for the example:

- Required RAM 220 Byte
- Required ROM 1 Kbyte
- Port 0.0 Clock for external Watchdog Timer.
- Port 1.0, 1.1, 1.4 3 Port Pins for lower Inverter MOSFETs.
- Port 7.0 MUTE Port-Pin, enables Inverter Outputs.
- Port 7.1, 7.2 2 Ports Pins for overcurrent detection.
- Port INTP (-1,-2,-3) Interrupt Inputs for Hallsensor feedback signal.
- INTP (-1,-2,-3) Interrupt generated by Hallsensors.
- Timer Channel 0 16-bit Interval Timer (200Hz), generates fixed Interrupt.
- INTTM00 Interrupt of Interval Timer (every 5 ms).
- Timer RD Motor Control Timer.
- TRDIO (-B0,-A1,-B1) 3 Timer RD Outputs for upper Inverter MOSFETs.
7.1 Commutation control

In order to obtain the points where to commutate the motor, the Halls are connected to Interrupt Input ports. At this commutation points an Interrupt (INTP1, INTP2, INTP3) is generated at every Hall change. This Interrupt is used to change the required Ports respectively to enable or disable the correct Timer RD Outputs.

For this application note the motor was driven in a counter clockwise (CCW) direction only (as viewed from the motor shaft end). The commutation sequence for the inverter transistors to achieve CCW rotation of the motor is as follows.

![Figure 17: Commutation control while motor is running](image)
7.2 Motor starting

As described in the previous section the control software needs an external Interrupt for commutation control. This Interrupt is generated while motor is rotating only. Therefore we have to start the motor without that external interrupt function by reading the absolute rotor position. This is done by reading Hall sensors at the beginning of program. After this is done a software state machine is entered to start the motor up to a fixed speed using a predefined commutation sequence. This is done until an external Interrupt signal is received by external Interrupt pin of MCU. If an external Interrupt is generated the motor is commutated using the Hall Interrupts (described in chapter 7.1).

Figure 18 Predefined commutation sequence to start the motor
Chapter 8  Software Module Descriptions

With the use of flow charts this section describes the major processes of the software.

8.1 Main Routine

Figure 19 shows the main program loop that is entered after a system RESET.

Figure 19  Flow Chart of main

**Initialization and operation**

- General Initialization (Clock, Ports etc.)
- Start Timer RD
- motor_positioning();
- Start Timer Channel 0
- Enable Interrupt: INTP1, INTP2, INTP3

Port toggle (P0.0)  
*For external watchdog timer*

Overcurrent detected?  
Yes  
Set MUTE Pin (P7.0) to Low  
*Motor stops running*

No  

RESET
After resetting the hardware registers and the various timers are initialised and enabled before the program enters an endless loop. The function `motor_position` is used to detect the absolute rotor position for motor starting.

At start of endless loop a Port toggle for external Watchdog timer is generated. Than an overcurrent check is done: Is overcurrent detected the motor stops running and endless loop starts again, is it not detected the endless loop starts up again immediately.

### 8.2 INTP1, INTP2, INTP3 Interrupt Service Routine

Figures 20, 21, 22 show the Interrupt Service Routines to commutate the motor while running.

**Figure 20**  INTP1 Interrupt Service Routine

- **INTP1**
  - *External Interrupt H1*
- **Stop INTTM00**
- **Falling edge**
  - Yes
  - **TRDOER1 = 0xFD; Enable TRDIOB0 Output**
  - **P1 = 0x10; Set P14 = 1**
- **No (rising edge)**
- **End of INTP1**
Figure 21  INTP2 Interrupt Service Routine

INTP2
*External Interrupt H2*

Stop INTTM00

Falling edge

No (rising edge)

Yes

TRDOER1 = 0xEF;
Enable TRDIOA1 Output

P1 = 0x02;
Set P11 = 1

End of INTP2

Figure 22  INTP3 Interrupt Service Routine

INTP3
*External Interrupt H3*

Stop INTTM00

Falling edge

No (rising edge)

Yes

TRDOER1 = 0xEF;
Enable TRDIOB1 Output

P1 = 0x01;
Set P10 = 1

End of INTP3
8.3 Interval Timer Interrupt Service Routine

Figure 23 shows a presentation of the motor state machine. Each state in the diagram represents a different pair of transistors energized in the inverter. The state machine is synchronized to an interval timer (Timer Channel 0) running at 5 ms intervals.

Figure 23  Interrupt Service Routine of Interval Timer
Chapter 9  Program Code (motor driving part only)

9.1 Main Routine

Hauptschleife

void main(void)
{
    DI(); //disable all interrupts
    P1 = 0x00; //set Port 0 to low
    TRDOER1 = 0xff; //disable all Timer RD outputs
    P7_bit.no0 = 1; //set mute to high
    EI(); //enable all interrupts

    R_TMR_RD0_Start(); //start Timer RD
    R_TAU0_Channel0_Start(); //start INTM00 interrupt

    R_INTC1_Start();
    R_INTC2_Start(); //enable Hall interrupts
    R_INTC3_Start();

    while (1U)
    {
        P0_bit.no0 = ~P0_bit.no0; //Port Toggle for extern WDT
        if (P7_bit.no2 == 0) // If overcurrent is detected...
            P7_bit.no0 = 1; //set MUTE Pin = Low
    }
}

Register settings to start Timer RD

void R_TMR_RD0_Start(void)
{
    TRDSR0 = _E0_TMRD_TRDSR0_DEFAULT_VALUR; /* clear TRD0 each interrupt request */
    TRDSR1 = _C0_TMRD_TRDSR1_DEFAULT_VALUR; /* clear TRD1 each interrupt request */
    TRDIF0 = 0U; /* clear TMRD0 interrupt flag */
    TRDMK0 = 0U; /* enable TMRD0 interrupt */
    TRDIF1 = 0U; /* clear TMRD1 interrupt flag */
    TRDMK1 = 0U; /* enable TMRD1 interrupt */
    TRDSTR |= _03_TRD_COUNT_STATR_INITIAL_VALUE; /* start TMRD counter */
}

Register settings to start Timer Channel 0 Interval Timers

void R_TAU0_Channel0_Start(void)
{
    TMIF00 = 0U; /* clear INTTM00 interrupt flag */
    TMMK00 = 0U; /* enable INTTM00 interrupt */
    TS0 |= _0001_TAU_CH0_START_TRG_ON;
}
Register settings to start external Interrupts

void R_INTC1_Start(void)
{
    PIF1 = 0U;    /* clear INTP1 interrupt flag */
    PMK1 = 0U;    /* enable INTP1 interrupt */
}

void R_INTC1_Stop(void)
{
    PMK1 = 1U;    /* disable INTP1 interrupt */
    PIF1 = 0U;    /* clear INTP1 interrupt flag */
}

void R_INTC2_Start(void)
{
    PIF2 = 0U;    /* clear INTP2 interrupt flag */
    PMK2 = 0U;    /* enable INTP2 interrupt */
}


9.2 INTTM00 Interrupt Service Routine

```c
int i = 1;

#pragma vector = INTTM00_vect
__interrupt void R_TAU0_Channel0_Interrupt(void)
{
    switch(i)
    {
        case 1:
            TRDOER1 = 0xfd;  // activate TRDIOB0
            i++;  // i = i+1
            break;
        case 2:
            P1 = 0x01;  // Set P1.0 = 1
            i++;  // i = i+1
            break;
        case 3:
            TRDOER1 = 0xef;  // activate TRDIOA1
            i++;  // i = i+1
            break;
        case 4:
            P1 = 0x10;  // Set P1.0 = 1
            i++;  // i = i+1
            break;
        case 5:
            TRDOER1 = 0xdf;  // activate TRDIOB1
            i++;  // i = i+1
            break;
        case 6:
            P1 = 0x02;  // Set P1.0 = 1
            i=1;  // i = i+1
            break;
        default:
            TRDOER1 = 0xff;  //disable all Timer RD outputs
            P1 = 0x00;  // Set P1 to low
            break;
    }
}
```
9.3 Interrupt Service Routines of external Interrupt INTP1, INTP2, INTP3

Interruptfunction INTP1

#include "INTC1 interrupts.h"

#pragma vector = INTP1_vect
__interrupt void R_INTC1_Interrupt(void)
{
    R_TAU0_Channel0_Stop(); // stop INTTM00
    if (P5_bit.no0 == 0) // 180°
        P1 = 0x10; // Set P1.0 = 1
    else // 0°
        TRDOER1 = 0xfd; // activate TRDIOB0
}

Interruptfunction INTP2

#pragma vector = INTP2_vect
__interrupt void R_INTC2_Interrupt(void)
{
    R_TAU0_Channel0_Stop(); // stop INTTM00
    if (P5_bit.no1 == 0) // 300°
        P1 = 0x02; // Set P1.0 = 1
    else // 120°
        TRDOER1 = 0xef; // activate TRDIOA1
}

Interruptfunction INTP3

#pragma vector = INTP3_vect
__interrupt void R_INTC3_Interrupt(void)
{
    R_TAU0_Channel0_Stop(); // stop INTTM00
    if (P3_bit.no0 == 0) // 60°
        P1 = 0x01; // Set P1.0 = 1
    else // 240°
        TRDOER1 = 0xdf; // activate TRDIOB1
}

Register settings to stop INTTM00 Interrupt

void R_TAU0_Channel0_Stop(void)
{
    TT0 |= _0001_TAU_CH0_STOP_TRG_ON; /* Mask channel 0 interrupt */
    TMMK00 = 1U; /* disable INTTM00 interrupt */
    TMIF00 = 0U; /* clear INTTM00 interrupt flag */
}
Chapter 10 Principle Circuit and Schematics

Figure 24  Picture of the Motor Control Circuit
Figure 25  Schematic Inverter-bridge Board
Figure 27  Schematic Microcontrollerboard
Chapter 11 Revision History

Table 2  Revision History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Change Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0100</td>
<td>3. November 2011</td>
<td>This is the initial version of the document.</td>
</tr>
</tbody>
</table>