To our customers,

---

**Old Company Name in Catalogs and Other Documents**

On April 1\(^{st}\), 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: [http://www.renesas.com](http://www.renesas.com)

---

April 1\(^{st}\), 2010
Renesas Electronics Corporation

---

Issued by: Renesas Electronics Corporation ([http://www.renesas.com](http://www.renesas.com))

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.
H8SX Family

Two-Phase Excitation Control of a Stepping Motor

Introduction

This application note discusses two-phase excitation control of a stepping motor using the TPU, PPG, and DTC functions incorporated in the H8SX/1657F.

Target Device

H8SX/1657F (advanced mode & 16M-byte address map mode)

Contents

1. Specifications ............................................................................................................... 2
2. Applicable Conditions ............................................................................................... 3
3. Description of Functions ........................................................................................... 4
4. Description of Operation .......................................................................................... 7
5. Description of Software .......................................................................................... 13
6. Flowchart ............................................................................................................... 24
1. Specifications

- A two-phase stepping motor is controlled by using the TPU, PPG, and DTC functions incorporated in the H8SX/1657F.
- The stepping motor is controlled through two-phase excitation and repeats the following sequence: forward rotation → stop → reverse rotation → stop.
- The sample task demonstrates speed-up and slow-down processing without software intervention.
- To protect the driver, a shoot-through current prevention period is set.

Figure 1 shows the connections for two-phase stepping motor control.

![Connections for Two-Phase Stepping Motor Control](image-url)
2. Applicable Conditions

Table 1 Applicable Conditions

<table>
<thead>
<tr>
<th>Item</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating frequency</td>
<td>Input clock: 17.5 MHz</td>
</tr>
<tr>
<td></td>
<td>System clock: 35 MHz</td>
</tr>
<tr>
<td></td>
<td>Peripheral module clock: 35 MHz</td>
</tr>
<tr>
<td></td>
<td>External bus clock: 35 MHz</td>
</tr>
<tr>
<td>Operating mode</td>
<td>Mode 6 (MD2 = 1, MD1 = 1, MD0 = 0)</td>
</tr>
<tr>
<td>Development tool</td>
<td>HEW Version 3.01.06</td>
</tr>
<tr>
<td>C/C++ compiler</td>
<td>H8S, H8/300 SERIES C/C++ Compiler Version 6.00.00 (from Renesas Technology Corp.)</td>
</tr>
<tr>
<td>Compile option</td>
<td>-cpu = h8sxa:24:md, -code = machinecode, -optimize = 1, -regparam = 3, -speed = (register, shift, struct, expression)</td>
</tr>
</tbody>
</table>

Table 2 Section Settings

<table>
<thead>
<tr>
<th>Address</th>
<th>Section Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>H’000000</td>
<td>CV1</td>
<td>Reset vector</td>
</tr>
<tr>
<td>H’000160</td>
<td>CV2</td>
<td>TPU TGI0A interrupt vector</td>
</tr>
<tr>
<td>H’001000</td>
<td>P</td>
<td>Program area</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>Data table storage</td>
</tr>
<tr>
<td>H’FF7000</td>
<td>B</td>
<td>RAM area</td>
</tr>
<tr>
<td>H’FF8560</td>
<td>DDTCV</td>
<td>DTC activation source vector</td>
</tr>
</tbody>
</table>
3. Description of Functions

3.1 Motor Specifications

This sample task uses a permanent magnet-type stepping motor (KP6P8-701 manufactured by Japan Servo Co., Ltd.). Table 3 summarizes the standard specifications of the KP6P8-701.

<table>
<thead>
<tr>
<th>Item</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>KP6P8-701</td>
</tr>
<tr>
<td>Number of phases</td>
<td>2</td>
</tr>
<tr>
<td>Stepping angle [deg./step]</td>
<td>7.5</td>
</tr>
<tr>
<td>Voltage [V]</td>
<td>12</td>
</tr>
<tr>
<td>Current [A/phase]</td>
<td>0.33</td>
</tr>
<tr>
<td>Winding resistance [Ω/phase]</td>
<td>36</td>
</tr>
<tr>
<td>Inductance [mH/phase]</td>
<td>28</td>
</tr>
<tr>
<td>Maximum static torque [mN•m]</td>
<td>78.4</td>
</tr>
<tr>
<td>Detent torque [mN•m]</td>
<td>1.3</td>
</tr>
<tr>
<td>Rotor inertia [g•cm²]</td>
<td>23.7</td>
</tr>
</tbody>
</table>

3.2 H8SX/1657F Functions

The H8SX/1657F functions used to control the stepping motor are described below. Figure 2 is a block diagram of the functions used in this sample task.

- **DTC**
  - Activated by compare-match A of the TPU.
  - Transfers an output pattern in the output pattern table to the NDR register of the PPG. After the transfer of an output pattern, the DTC transfers the pulse period data in the period data table to the TGRB register of the TPU by chain transfer.

- **TPU**
  - Compare-match A: Activates the DTC and PPG.
  - Compare-match B: Clears the timer counter and activates the PPG.

- **PPG**
  - In this sample task, outputs 4 bits of pulse signals that include shoot-through current prevention period (non-overlap time).
  - Compare-match B: Outputs pulse signals of high-to-low transition and suspends output of low-to-high transition.
  - Compare-match A: Outputs the low-to-high pulse signals that are suspended on compare-match B above. (This output is delayed by the time set by TGRA.)
Two-Phase Excitation Control of a Stepping Motor

Stores pulse period data.
The DTC transfers the period data to TGRB by chain transfer.

Stores waveform data for stepping motor control.
Changes the output pattern using DTCA and outputs waveform each time TPU compare match occurs.

Stores waveform data for stepping motor control.

Figure 2  H8SX/1657F’s Functions Used in This Application
3.3 DTC Vector Table

Figure 3 shows the DTC vector table and data arrangement in memory. The DTC transfer information should be stored to the addresses from H'FF6000 in the following order: MRA, SAR, MRB, DAR, CRA, and CRB.

![Diagram of DTC Vector Table and Data Arrangement in Memory]

**Figure 3  Example of DTC Vector Table and Data Arrangement in Memory**
4. Description of Operation

4.1 Stepping Motor Operation

Figure 4 shows an example of two-phase stepping motor operation through two-phase excitation where the step angle of the motor is 7.5 degrees/step. The operation is summarized below.

- When the output pulse is high, the corresponding phase is excited, as shown in figure 4.
- Firstly, phases A and B are excited and the rotor is positioned between phases B and A.
- Next, phases A and B are excited simultaneously and the rotor is positioned between phases A and B. Subsequently, two adjacent phases are excited in the following sequence to cause the rotor to rotate: phases B and A → phases A and B → phases B and A → phases A and B.
- Reverse rotation of the stepping motor is achieved by exciting the phases in the reverse sequence: phases A and B → phases B and A → phases A and B → phases B and A.
- The stepping motor is stopped by holding the phase excitation for a specified period at the last phase of forward or reverse rotation.
Figure 4  Example of Stepping Motor Operation
4.2 Non-Overlap Time

When the output pattern is switched, the shoot-through current prevention period \( n \) (non-overlap time) is inserted as shown in figure 5. The motor driver may be damaged by a turn-off delay that occurs when the excitation pattern is switched. Non-overlap time is inserted to prevent this problem.

![Figure 5 Example of Output with Non-Overlap Time](image)

4.3 Speed-Up and Slow-Down Operation

Speed-up and slow-down operations effectively prevent motor from being out of step. In particular, if a train of short-period pulses is suddenly output, the motor may not be able to handle the load and does not rotate. Speed-up and slow-down operation control is applied to avoid this problem. The speed-up and slow-down operation sequence is described below.

- The pulse period is gradually shortened until the specified number of pulses has been output (Speed up).
- The specified number of pulses with a fixed pulse period is output. (Constant speed).
- The pulse period is gradually extended until the specified number of pulses has been output (Slow down).

![Figure 6 Speed-Up and Slow-Down Operation](image)
4.4 Stepping Motor Control Flow

Figure 7 shows the flowchart of stepping motor control.

![Flowchart of Stepping Motor Control](image-url)

**Figure 7  Flowchart of Stepping Motor Control**
4.5 Example of Four-Phase Pulse Output

Figure 8 shows an example of four-phase pulse output.

![Timing of Stepping Motor Operation](image)

**Figure 8** Timing of Stepping Motor Operation
### Hardware processing

None

### Software processing

**PPG:**
- Set the first and second pulse output values.
- Set up the P00 to P03 pins to function as output pins.
- Set so that the output of pulse output group 0 is triggered by compare-match on TPU channel 0.
- Select direct output and non-overlap operation for all pulse output groups.

**DTC:**
- Enable activation of DTC by TGI0A interrupt.
- Set the DTC vector base register to H'FF8000.
- PTN0 (Motor output pattern transfer information)
  - Increment SAR after transfer
  - DAR is fixed
  - Repeat mode
  - Byte-size transfer
  - Enable chain transfer
  - Transfer source is the repeat area
  - Transfer source address: start address of pattbl
  - Transfer destination address: address of NDRL_B
  - Number of transfers: 4
- CYC0 (Motor period data transfer information)
  - Increment SAR after transfer
  - DAR is fixed
  - Normal mode
  - Word-size transfer
  - Disable chain transfer
  - Transfer source address: start address of uptbl
  - Transfer destination address: address of TGRB_0
  - Number of transfers: specified by "UPTIME"

**TPU:**
- Specify TGRA_0 and TGRB_0 as output compare registers.
- Set non-overlap time.
- Enable TGI0A interrupt requests.
- Start counting by TCNT_0.

---

**Figure 9** Initialization Processing Immediately After Reset
5. Description of Software

5.1 List of Functions

Table 4 lists the functions used in this sample task. Figure 10 shows the function hierarchy in this sample task.

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>init</td>
<td>Initialization routine: Cancels module stop mode, sets clocks, and calls main function.</td>
</tr>
<tr>
<td>main</td>
<td>Main routine: Initializes the TPU, PPG, and DTC and makes settings for forward speed-up operation.</td>
</tr>
<tr>
<td>tgi0a_int</td>
<td>TPU interrupt processing: Controls motor operation in each stage.</td>
</tr>
<tr>
<td>fslowup0</td>
<td>Called after completion of reverse stop operation and changes the DTC transfer modes for forward speed-up operation.</td>
</tr>
<tr>
<td>fconst0</td>
<td>Called after completion of forward speed-up operation and switches the DTC transfer modes for forward constant-speed operation.</td>
</tr>
<tr>
<td>fslowdwn0</td>
<td>Called after completion of forward constant-speed operation and changes the DTC transfer modes for forward slow-down operation.</td>
</tr>
<tr>
<td>rslowup0</td>
<td>Called after completion of forward stop operation and changes the DTC transfer modes for reverse speed-up operation.</td>
</tr>
<tr>
<td>rconst0</td>
<td>Called after completion of reverse speed-up operation and changes the DTC transfer modes for reverse constant-speed operation.</td>
</tr>
<tr>
<td>rslowdwn0</td>
<td>Called after completion of reverse constant-speed operation and changes the DTC transfer modes for reverse slow-down operation.</td>
</tr>
<tr>
<td>frstop0</td>
<td>Called after completion of reverse slow-down operation and changes the DTC transfer modes for stop operation.</td>
</tr>
</tbody>
</table>

Figure 10  Hierarchy of Functions
5.2 Constants

Table 5 lists the constants used in this sample task.

Table 5 Description of Constants

<table>
<thead>
<tr>
<th>Constant Name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UPTIME</td>
<td>49</td>
<td>Number of steps for speed-up and slow-down operations</td>
</tr>
<tr>
<td>CNSTTIME</td>
<td>481</td>
<td>Number of steps for constant-speed operation</td>
</tr>
<tr>
<td>STOPTIME</td>
<td>20</td>
<td>Number of steps for stop operation</td>
</tr>
</tbody>
</table>

5.3 Data Table Variables

- **Output Pattern Table**
  
  Data table of output patterns for stepping motor excitation.

  ```
pattbl[4] = {
    0xF6, .... Excites phase B (PO0) and phase A (PO3)
    0xF3, .... Excites phase A (PO3) and phase B (PO2)
    0xF9, .... Excites phase B (PO2) and phase A (PO1)
    0xFC, .... Excites phase A (PO1) and phase B (PO0)
  };
```

- **Period Data Table**

  ```
  uptbl[UPTIME] = {
    0xFFF0, 0xD000, 0x7000, 0x6F00, 0x6E00, 0x6D00, 0x6C00, 0x6B00, 0x6A00, 0x5900, 
    0x5800, 0x5700, 0x5600, 0x5500, 0x5400, 0x5300, 0x5200, 0x5100, 0x5000, 0x4F00, 
    0x4E00, 0x4D00, 0x4C00, 0x4B00, 0x4A00, 0x4900, 0x4800, 0x4700, 0x4600, 0x4500, 
    0x4400, 0x4300, 0x4200, 0x4100, 0x4000, 0x3F00, 0x3E00, 0x3D00, 0x3C00, 0x3B00, 
    0x3A00, 0x3900, 0x3800, 0x3700, 0x3600, 0x3500, 0x3400, 0x3200, 0x3000,
  };
```
### 5.4 Internal Registers

The internal registers used in this sample task are described below.

- **System Clock Control Register (SCKCR)**
  - Address: H'FFFDC4

<table>
<thead>
<tr>
<th>Bit</th>
<th>Bit Name</th>
<th>Setting</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>ICK2</td>
<td>0</td>
<td>System clock (Iஇ) select</td>
</tr>
<tr>
<td>9</td>
<td>ICK1</td>
<td>0</td>
<td>These bits selects the frequency of the system clock supplied to the CPU, DMAC, and DTC.</td>
</tr>
<tr>
<td>8</td>
<td>ICK0</td>
<td>1</td>
<td>001: Input clock × 2</td>
</tr>
<tr>
<td>6</td>
<td>PCK2</td>
<td>0</td>
<td>Peripheral module clock (Pஇ) select</td>
</tr>
<tr>
<td>5</td>
<td>PCK1</td>
<td>0</td>
<td>These bits selects the frequency of the peripheral module clock.</td>
</tr>
<tr>
<td>4</td>
<td>PCK0</td>
<td>1</td>
<td>001: Input clock × 2</td>
</tr>
<tr>
<td>2</td>
<td>BCK2</td>
<td>0</td>
<td>External bus clock (Bஇ) select</td>
</tr>
<tr>
<td>1</td>
<td>BCK1</td>
<td>0</td>
<td>These bits selects the frequency of the external bus clock.</td>
</tr>
<tr>
<td>0</td>
<td>BCK0</td>
<td>1</td>
<td>001: Input clock × 2</td>
</tr>
</tbody>
</table>

- **Module Stop Control Register A (MSTPCR A)**
  - Address: H'FFFDC8

<table>
<thead>
<tr>
<th>Bit</th>
<th>Bit Name</th>
<th>Setting</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>ACSE</td>
<td>0</td>
<td>All-module-clock-stop mode enable</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Enables or disables all-module-clock-stop mode to reduce current consumption by also stopping the bus controller and I/O ports operations when the CPU executes a SLEEP instruction after module stop mode has been set for all the on-chip peripheral modules under control of MSTPCR.</td>
</tr>
<tr>
<td>13</td>
<td>MSTPA13</td>
<td>0</td>
<td>DMA controller (DMAC)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0: Cancels module stop mode.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1: Sets module stop mode.</td>
</tr>
<tr>
<td>12</td>
<td>MSTPA12</td>
<td>0</td>
<td>Data transfer controller (DTC)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0: Cancels module stop mode.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1: Sets module stop mode.</td>
</tr>
<tr>
<td>9</td>
<td>MSTPA9</td>
<td>0</td>
<td>8-bit timers (TMR3, TMR2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0: Cancels module stop mode.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1: Sets module stop mode.</td>
</tr>
<tr>
<td>8</td>
<td>MSTPA8</td>
<td>0</td>
<td>8-bit timers (TMR1, TMR0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0: Cancels module stop mode.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1: Sets module stop mode.</td>
</tr>
<tr>
<td>5</td>
<td>MSTPA5</td>
<td>0</td>
<td>D/A converter (channels 1, 0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0: Cancels module stop mode.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1: Sets module stop mode.</td>
</tr>
<tr>
<td>3</td>
<td>MSTPA3</td>
<td>0</td>
<td>A/D converter (unit 0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0: Cancels module stop mode.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1: Sets module stop mode.</td>
</tr>
<tr>
<td>0</td>
<td>MSTPA0</td>
<td>0</td>
<td>16-bit timer pulse unit (TPU channels 5 to 0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0: Cancels module stop mode.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1: Sets module stop mode.</td>
</tr>
</tbody>
</table>
## Module Stop Control Register B (MSTPCRB)

Address: H'FFFDCA

<table>
<thead>
<tr>
<th>Bit</th>
<th>Bit Name</th>
<th>Setting</th>
<th>Function</th>
</tr>
</thead>
</table>
| 15  | MSTPB15  | 0       | Programmable pulse generator (PPG)  
0: Cancels module stop mode.  
1: Sets module stop mode. |
| 12  | MSTPB12  | 0       | Serial communication interface 4 (SCI_4)  
0: Cancels module stop mode.  
1: Sets module stop mode. |
| 10  | MSTPB10  | 0       | Serial communication interface 2 (SCI_2)  
0: Cancels module stop mode.  
1: Sets module stop mode. |
| 9   | MSTPB9   | 0       | Serial communication interface 1 (SCI_1)  
0: Cancels module stop mode.  
1: Sets module stop mode. |
| 8   | MSTPB8   | 0       | Serial communication interface 0 (SCI_0)  
0: Cancels module stop mode.  
1: Sets module stop mode. |

## Module Stop Control Register C (MSTPCRC)

Address: H'FFFDCC

<table>
<thead>
<tr>
<th>Bit</th>
<th>Bit Name</th>
<th>Setting</th>
<th>Function</th>
</tr>
</thead>
</table>
| 2   | MSTPC2   | 0       | On-chip RAM 2 (H'FFF6000 to H'FFF7FFF)  
0: Cancels module stop mode.  
1: Sets module stop mode. |
| 1   | MSTPC1   | 0       | On-chip RAM 1 (H'FFF8000 to H'FFF9FFF)  
0: Cancels module stop mode.  
1: Sets module stop mode. |
| 0   | MSTPC0   | 0       | On-chip RAM 0 (H'FFFA000 to H'FFFBFFF)  
0: Cancels module stop mode.  
1: Sets module stop mode. |
### DTC Vector Base Register (DTCVBR)
- **Address:** H’FFFD80
- **Function:** 32-bit register that specifies the base address used in vector table address calculation
- **Setting:** H’FF8000

### DTC Control Register (DTCCR)
- **Address:** H’FFFF30

<table>
<thead>
<tr>
<th>Bit</th>
<th>Bit Name</th>
<th>Setting</th>
<th>Function</th>
</tr>
</thead>
</table>
| 4   | RRS      | 0       | DTC transfer information read skip enable  
|     |          |         | 0: Transfer information read skip is not performed.  
|     |          |         | 1: Transfer information read skip is performed when the vector numbers match. |
| 3   | RCHNE    | 0       | Chain transfer enable after DTC repeat transfer  
|     |          |         | 0: Disables the chain transfer after repeat transfer.  
|     |          |         | 1: Enables the chain transfer after repeat transfer. |
| 0   | ERR      | 0       | Transfer stop flag  
|     |          |         | 0: Address error nor NMI interrupt request has not occurred.  
|     |          |         | 1: Address error or NMI interrupt request has occurred. |

### DTC Enable Register B (DTCERB)
- **Address:** H’FFFF22

<table>
<thead>
<tr>
<th>Bit</th>
<th>Bit Name</th>
<th>Setting</th>
<th>Function</th>
</tr>
</thead>
</table>
| 13  | DTCEB13  | 1       | DTC activation enable  
|     |          |         | 0: Disables DTC activation by TGI0A interrupts of the TPU_0.  
|     |          |         | 1: Enables DTC activation by TGI0A interrupts of the TPU_0. |

### PPG Output Control Register (PCR)
- **Address:** H’FFFF76

<table>
<thead>
<tr>
<th>Bit</th>
<th>Bit Name</th>
<th>Setting</th>
<th>Function</th>
</tr>
</thead>
</table>
| 1   | G0CMS1   | 0       | Group 0 compare match select  
|     |          |         | 0: Output of pulse output group 0 is triggered by compare-match on TPU channel 0. |
| 0   | G0CMS0   | 0       | 00: Output of pulse output group 0 is triggered by compare-match on TPU channel 0. |

### PPG Output Mode Register (PMR)
- **Address:** H’FFFF77

<table>
<thead>
<tr>
<th>Bit</th>
<th>Bit Name</th>
<th>Setting</th>
<th>Function</th>
</tr>
</thead>
</table>
| 4   | G0INV    | 1       | Group 0 inversion  
|     |          |         | 0: Inverted output  
|     |          |         | 1: Direct output |
| 0   | G0NOV    | 1       | Group 0 non-overlap  
|     |          |         | 0: Normal operation  
|     |          |         | 1: Non-overlap operation |
### Next Data Enable Register L (NDERL)

<table>
<thead>
<tr>
<th>Bit</th>
<th>Bit Name</th>
<th>Setting</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>NDER7</td>
<td>0</td>
<td>Next data enable 7 to 0</td>
</tr>
<tr>
<td>6</td>
<td>NDER6</td>
<td>0</td>
<td>When a bit in this register is set to 1, data of the corresponding bit in NDRL is transferred to PODRL, triggered by the selected output trigger. Data transfer from NDRL to PODRL does not take place for the bits whose corresponding bits in this register are clear.</td>
</tr>
<tr>
<td>5</td>
<td>NDER5</td>
<td>0</td>
<td>NDER5 is set to 1, data of the corresponding bit in NDRL is transferred to PODRL, triggered by the selected output trigger. Data transfer from NDRL to PODRL does not take place for the bits whose corresponding bits in this register are clear.</td>
</tr>
<tr>
<td>4</td>
<td>NDER4</td>
<td>0</td>
<td>NDER4 is set to 1, data of the corresponding bit in NDRL is transferred to PODRL, triggered by the selected output trigger. Data transfer from NDRL to PODRL does not take place for the bits whose corresponding bits in this register are clear.</td>
</tr>
<tr>
<td>3</td>
<td>NDER3</td>
<td>1</td>
<td>NDER3 is set to 1, data of the corresponding bit in NDRL is transferred to PODRL, triggered by the selected output trigger. Data transfer from NDRL to PODRL does not take place for the bits whose corresponding bits in this register are clear.</td>
</tr>
<tr>
<td>2</td>
<td>NDER2</td>
<td>1</td>
<td>NDER2 is set to 1, data of the corresponding bit in NDRL is transferred to PODRL, triggered by the selected output trigger. Data transfer from NDRL to PODRL does not take place for the bits whose corresponding bits in this register are clear.</td>
</tr>
<tr>
<td>1</td>
<td>NDER1</td>
<td>1</td>
<td>NDER1 is set to 1, data of the corresponding bit in NDRL is transferred to PODRL, triggered by the selected output trigger. Data transfer from NDRL to PODRL does not take place for the bits whose corresponding bits in this register are clear.</td>
</tr>
<tr>
<td>0</td>
<td>NDER0</td>
<td>1</td>
<td>NDER0 is set to 1, data of the corresponding bit in NDRL is transferred to PODRL, triggered by the selected output trigger. Data transfer from NDRL to PODRL does not take place for the bits whose corresponding bits in this register are clear.</td>
</tr>
</tbody>
</table>

### Output Data Register L (PODRL)

<table>
<thead>
<tr>
<th>Bit</th>
<th>Bit Name</th>
<th>Setting</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>POD7</td>
<td>0</td>
<td>Output data register 7 to 0</td>
</tr>
<tr>
<td>6</td>
<td>POD6</td>
<td>0</td>
<td>For the bits set to generate pulse outputs by NDERL, the values of the corresponding bits in NDRL are transferred to this register by the output trigger during PPG operation. While a bit in NDERL is set to 1, the CPU cannot write to the corresponding bit of this register. While a bit in NDERL is clear, the initial pulse output value can be set in the corresponding bit of this register.</td>
</tr>
<tr>
<td>5</td>
<td>POD5</td>
<td>0</td>
<td>For the bits set to generate pulse outputs by NDERL, the values of the corresponding bits in NDRL are transferred to this register by the output trigger during PPG operation. While a bit in NDERL is set to 1, the CPU cannot write to the corresponding bit of this register. While a bit in NDERL is clear, the initial pulse output value can be set in the corresponding bit of this register.</td>
</tr>
<tr>
<td>4</td>
<td>POD4</td>
<td>0</td>
<td>For the bits set to generate pulse outputs by NDERL, the values of the corresponding bits in NDRL are transferred to this register by the output trigger during PPG operation. While a bit in NDERL is set to 1, the CPU cannot write to the corresponding bit of this register. While a bit in NDERL is clear, the initial pulse output value can be set in the corresponding bit of this register.</td>
</tr>
<tr>
<td>3</td>
<td>POD3</td>
<td>1</td>
<td>POD3 is set to 1, data of the corresponding bit in NDRL is transferred to PODRL, triggered by the selected output trigger. Data transfer from NDRL to PODRL does not take place for the bits whose corresponding bits in this register are clear.</td>
</tr>
<tr>
<td>2</td>
<td>POD2</td>
<td>1</td>
<td>POD2 is set to 1, data of the corresponding bit in NDRL is transferred to PODRL, triggered by the selected output trigger. Data transfer from NDRL to PODRL does not take place for the bits whose corresponding bits in this register are clear.</td>
</tr>
<tr>
<td>1</td>
<td>POD1</td>
<td>0</td>
<td>POD1 is set to 0, data of the corresponding bit in NDRL is transferred to PODRL, triggered by the selected output trigger. Data transfer from NDRL to PODRL does not take place for the bits whose corresponding bits in this register are clear.</td>
</tr>
<tr>
<td>0</td>
<td>POD0</td>
<td>0</td>
<td>POD0 is set to 0, data of the corresponding bit in NDRL is transferred to PODRL, triggered by the selected output trigger. Data transfer from NDRL to PODRL does not take place for the bits whose corresponding bits in this register are clear.</td>
</tr>
</tbody>
</table>

### Next Data Register L (NDRL)

<table>
<thead>
<tr>
<th>Bit</th>
<th>Bit Name</th>
<th>Setting</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>NDER3</td>
<td>1</td>
<td>Next data 3 to 0</td>
</tr>
<tr>
<td>2</td>
<td>NDER2</td>
<td>1</td>
<td>The contents of this register are transferred to the corresponding bits in the PODRL by the output trigger specified by PCR.</td>
</tr>
<tr>
<td>1</td>
<td>NDER1</td>
<td>0</td>
<td>The contents of this register are transferred to the corresponding bits in the PODRL by the output trigger specified by PCR.</td>
</tr>
<tr>
<td>0</td>
<td>NDER0</td>
<td>0</td>
<td>The contents of this register are transferred to the corresponding bits in the PODRL by the output trigger specified by PCR.</td>
</tr>
</tbody>
</table>

### Timer Start Register (TSTR)

<table>
<thead>
<tr>
<th>Bit</th>
<th>Bit Name</th>
<th>Setting</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>CST0</td>
<td>1</td>
<td>Counter start 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0: Stops counting by TCNT_0 of the TPU.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1: Starts counting by TCNT_0 of the TPU.</td>
</tr>
</tbody>
</table>
Timer Control Register_0 (TCR_0)  
Address: H'FFFFC0

<table>
<thead>
<tr>
<th>Bit</th>
<th>Bit Name</th>
<th>Setting</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>CCLR2</td>
<td>0</td>
<td>Counter clear 2 to 0</td>
</tr>
<tr>
<td>6</td>
<td>CCLR1</td>
<td>1</td>
<td>010: Clear TCNT_0 on compare match or input capture by TGRB_0.</td>
</tr>
<tr>
<td>5</td>
<td>CCLR0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>CKEG1</td>
<td>0</td>
<td>Clock edge 1, 0</td>
</tr>
<tr>
<td>3</td>
<td>CKEG0</td>
<td>0</td>
<td>00: The counter counts the falling edges when an internal clock is input, or counts the rising edges when an external clock is input.</td>
</tr>
<tr>
<td>2</td>
<td>TPSC2</td>
<td>0</td>
<td>Timer prescaler 2 to 0</td>
</tr>
<tr>
<td>1</td>
<td>TPSC1</td>
<td>1</td>
<td>010: The counter clock source is the internal clock Pφ/16.</td>
</tr>
<tr>
<td>0</td>
<td>TPSC0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Timer I/O Control Register H_0 (TIORH_0)  
Address: H'FFFFC2

<table>
<thead>
<tr>
<th>Bit</th>
<th>Bit Name</th>
<th>Setting</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>IOB3</td>
<td>0</td>
<td>I/O control B3 to B0</td>
</tr>
<tr>
<td>6</td>
<td>IOB2</td>
<td>0</td>
<td>0000: TGRB_0 operates as an output compare register and the TIOCB0 pin output is disabled.</td>
</tr>
<tr>
<td>5</td>
<td>IOB1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>IOB0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>IOA3</td>
<td>0</td>
<td>I/O control A3 to A0</td>
</tr>
<tr>
<td>2</td>
<td>IOA2</td>
<td>0</td>
<td>0000: TGRA_0 operates as an output compare register and the TIOCA0 pin output is disabled.</td>
</tr>
<tr>
<td>1</td>
<td>IOA1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>IOA0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Timer Interrupt Enable Register_0 (TIER_0)  
Address: H'FFFFC4

<table>
<thead>
<tr>
<th>Bit</th>
<th>Bit Name</th>
<th>Setting</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>TGIEA</td>
<td>1</td>
<td>TGR interrupt enable A</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0: Disables interrupt request (TGIA) generation by the TGFA bit.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1: Enables interrupt request (TGIA) generation by the TGFA bit.</td>
</tr>
</tbody>
</table>

Timer Status Register_0 (TSR_0)  
Address: H'FFFFC5

<table>
<thead>
<tr>
<th>Bit</th>
<th>Bit Name</th>
<th>Setting</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TGFB</td>
<td>0</td>
<td>Input capture/output compare flag B</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Indicates the occurrence of TGRB input capture or compare match.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0: Input capture or compare match has not occurred.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1: Input capture or compare match has occurred.</td>
</tr>
<tr>
<td>0</td>
<td>TGFA</td>
<td>0</td>
<td>Input capture/output compare flag A</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Indicates the occurrence of TGRA input capture or compare match.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0: Input capture or compare match has not occurred.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1: Input capture or compare match has occurred.</td>
</tr>
</tbody>
</table>

Timer Counter_0 (TCNT_0)  
Address: H'FFFFC6

Function: A 16-bit readable/writable register that is incremented according to the setting of TCR_0.  
Setting value: H'0065
• Timer General Register A_0 (TGRA_0) Address: H'FFFFC8
  — Function: A 16-bit register that is compared with the counter in output compare operation.
  — Setting value: H'0064

• Timer General Register B_0 (TGRB_0) Address: H'FFFFCA
  — Function: A 16-bit register that is compared with the counter in output compare operation.
  — Setting value: H'FF00
5.5 DTC Transfer Information Settings

5.5.1 PTN0

PTN0 is the motor output pattern transfer information. The addresses given below are those of the RAM area to which PTN0 is allocated.

- **DTC Mode Register A (MRA)**
  
  Address: H'FF6000

<table>
<thead>
<tr>
<th>Bit</th>
<th>Bit Name</th>
<th>Setting</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>MD1</td>
<td>0</td>
<td>DTC mode 1, 0</td>
</tr>
</tbody>
</table>
| 6   | MD0      | 1       | 00: Normal mode  
|     |          |         | 01: Repeat mode |
| 5   | Sz1      | 0       | DTC data transfer size 1, 0 |
| 4   | Sz0      | 0       | 00: Byte-size transfer  
|     |          |         | 01: Word-size transfer |
| 3   | SM1      | 1       | Source address mode 1, 0 |
| 2   | SM0      | 0       | 0X: SAR is fixed.  
|     |          |         | 10: SAR is incremented after transfer.  
|     |          |         | 11: SAR is decremented after transfer. |

Note: X = Don't care

- **DTC Mode Register B (MRB)**
  
  Address: H'FF6004

<table>
<thead>
<tr>
<th>Bit</th>
<th>Bit Name</th>
<th>Setting</th>
<th>Function</th>
</tr>
</thead>
</table>
| 7   | CHNE     | 1       | DTC chain transfer enable  
|     |          |         | 0: Disables chain transfer.  
|     |          |         | 1: Enables chain transfer. |
| 6   | CHNS     | 0       | DTC chain transfer select  
|     |          |         | 0: Chain transfer every time.  
|     |          |         | 1: Chain transfer only when transfer counter value = 0. |
| 4   | DTS      | 1       | DTC transfer mode select  
|     |          |         | 0: Specifies the destination as repeat or block area.  
|     |          |         | 1: Specifies the source as repeat or block area. |
| 3   | DM1      | 0       | Destination address mode 1, 0 |
| 2   | DM0      | 0       | 0X: DAR is fixed. |

Note: X = Don't care

- **DTC Source Address Register (SAR)**
  
  Address: H'FF6000

  — Function: A 32-bit register that specifies the source address of data to be transferred by the DTC. The lower 24 bits are valid in short address mode.

  — Setting value: pattbl

- **DTC Destination Address Register (DAR)**
  
  Address: H'FF6004

  — Function: A 32-bit register that specifies the destination address to which data is transferred by the DTC. The lower 24 bits are valid in short address mode.

  — Setting value: &NDRL_B

- **DTC Transfer Count Register A (CRA)**
  
  Address: H'FF6008

  — Function: A 16-bit register that specifies the number of DTC data transfers.

  — Setting value: H'0404
5.5.2 CYC0

CYC0 is the motor period data transfer information. The addresses given below are those of the RAM area to which CYC0 is allocated.

- **DTC Mode Register A (MRA)**
  
  **Address:** H'FF600C

<table>
<thead>
<tr>
<th>Bit</th>
<th>Bit Name</th>
<th>Setting</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>MD1</td>
<td>0</td>
<td>DTC mode 1, 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>00: Normal mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>01: Repeat mode</td>
</tr>
<tr>
<td>6</td>
<td>MD0</td>
<td>0</td>
<td>DTC data transfer size 1, 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>00: Byte-size transfer</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>01: Word-size transfer</td>
</tr>
<tr>
<td>5</td>
<td>Sz1</td>
<td>0</td>
<td>Source address mode 1, 0</td>
</tr>
<tr>
<td>4</td>
<td>Sz0</td>
<td>1</td>
<td>00: SAR is fixed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10: SAR is incremented after transfer.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>11: SAR is decremented after transfer.</td>
</tr>
</tbody>
</table>

**Note:** X = Don't care

- **DTC Mode Register B (MRB)**
  
  **Address:** H'FF6010

<table>
<thead>
<tr>
<th>Bit</th>
<th>Bit Name</th>
<th>Setting</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>CHNE</td>
<td>0</td>
<td>DTC chain transfer enable</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0: Disables chain transfer.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1: Enables chain transfer.</td>
</tr>
<tr>
<td>6</td>
<td>CHNS</td>
<td>0</td>
<td>DTC chain transfer select</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0: Chain transfer every time.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1: Chain transfer only when transfer counter value = 0.</td>
</tr>
<tr>
<td>4</td>
<td>DTS</td>
<td>0</td>
<td>DTC transfer mode select</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0: Specifies the destination as repeat or block area.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1: Specifies the source as repeat or block area.</td>
</tr>
<tr>
<td>3</td>
<td>DM1</td>
<td>0</td>
<td>Destination address mode 1, 0</td>
</tr>
<tr>
<td>2</td>
<td>DM0</td>
<td>0</td>
<td>0X: DAR is fixed.</td>
</tr>
</tbody>
</table>

**Note:** X = Don't care

- **DTC Source Address Register (SAR)**
  
  **Address:** H'FF600C

  — Function: A 32-bit register that specifies the source address of data to be transferred by the DTC. The lower 24 bits are valid in short address mode.

  — Setting value: uptbl

- **DTC Destination Address Register (DAR)**
  
  **Address:** H'FF6010

  — Function: A 32-bit register that specifies the destination address to which data is transferred by the DTC. The lower 24 bits are valid in short address mode.

  — Setting value: &TGRB_0

- **DTC Transfer Count Register A (CRA)**
  
  **Address:** H'FF6014

  — Function: A 16-bit register that specifies the number of DTC data transfers.

  — Setting value: UPTIME
5.6 RAM Usage

Table 6 describes the RAM usage in this sample task.

<table>
<thead>
<tr>
<th>Type</th>
<th>Variable Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTC_tag</td>
<td>PTN0</td>
<td>Output pattern transfer information for the motor</td>
</tr>
<tr>
<td>DTC_tag</td>
<td>CYC0</td>
<td>Period data transfer information for the motor</td>
</tr>
<tr>
<td>unsigned char</td>
<td>nextmode0</td>
<td>Sets stepping motor operating mode.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: Forward speed-up control</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: Forward constant-speed control</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2: Forward slow-down control</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3: Stop control</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4: Reverse speed-up control</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5: Reverse constant-speed control</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6: Reverse slow-down control</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7: Stop control</td>
</tr>
</tbody>
</table>
6. Flowchart

6.1 init Function

```
init

H'80 → CCR
Initialize CCR to disable interrupts.

H'0111 → SCKCR
Select multiplication factors for the clocks.
\( I_0, P_0, B_0 = \times 2, \times 2, \times 2 \)

H'0000 → MSTPCRA
H'0003 → MSTPCRB
H'FF00 → MSTPCRC
Cancel module stop mode of all the modules.

main
```
6.2 Main Function

```plaintext
main

nextmode0 = 1
Set the initial value of motor control mode.

PODRL = H'09
Set the first pulse output value.

NDERL = H'0F
Set up PPG output pins.

PCR = H'4E
Select the compare match output trigger: for Group 0, compare match of TPU0.

PMR = H'FF
For all groups, select
- Direct output
- Non-overlap operation

NDRL_B = H'FC
Set the next pulse output value.

tmp = DTCCR
DTCCR = H'00
Stop the DTC and clear flags.

PTN0.DTC1.SAR = pattbl
Set the start address of output pattern table as transfer source.

PTN0.DTC2.DAR = (&NDRL_B)
Set NDRL_B as transfer source.

PTN0.DTC2.MRB = H'90
- Execute chain transfer.
- Disable DTC interrupts.
- Specify the source as the repeat area.
- Fix DAR after transfer.

PTN0.CRA = H'0404
- Number of transfers = 4
- Transfer counter = 4

CYC0.DTC1.SAR = uptbl
Set the start address of period data table as transfer source.

CYC0.DTC1.MRA = H'48
- Select repeat transfer mode.
- Select byte-size transfer.
- Increment SAR after transfer.

CYC0.DTC2.DAR = (&TGRB_0)
Set TGRB_0 as transfer source.

CYC0.DCT2.DAR = (&TGRB_0)
Intensity of TGRB_0.

CYC0.DCT2.MRB = H'00
- Disable chain transfer.
- Disable DTC interrupts.
- Fix DAR after transfer.

CYC0.CRA = UPTIME
- Transfer counter = UPCNT

DTCERB.b13 = 1
Enable DTC activation by TGI0A interrupt.

TCNT_0 = H'0065
Load TCNT_0 with initial value.

TCRH = H'00
Set TGRA_0 and TGRB_0 as output compare registers (output disabled).

TGRA_0 = H'0064
Set the non-overlap period.

TGRB_0 = H'FF00
Set the period of PPG output trigger.

TPU_TCR_0 = H'42
- Clear TCNT_0 on compare match with TGRB_0 .
- Counter clock = internal clock Pφ/16

TPU_TCR_0 = H'41
Enable TGI0A interrupt requests.

TPU_TCR_0 = H'40
Clear the TPU flag.

I = 0
Enable interrupts.

TPU_TSTR = H'01
Start counting by TCNT_0.

DTVCBR = H'FF8000
Set the interrupt vector address to H'FF8000.
```

6.3 tgi0a_int Function

![Flowchart]

- **tgi0a_int**
- **nextmode0**: ?

- **fslowup0**: Forward speed-up
  - nextmode0 = 1
  - Go on to forward constant-speed mode.

- **fconst0**: Forward constant-speed
  - nextmode0 = 2
  - Go on to forward slow-down mode.

- **fslowdwn0**: Forward slow-down
  - nextmode0 = 3
  - Go on to rotation stop mode.

- **frstop0**: Rotation stop
  - nextmode0 = 4
  - Go on to reverse speed-up mode.

- **rslowup0**: Reverse speed-up
  - nextmode0 = 5
  - Go on to constant-speed mode.

- **rconst0**: Reverse constant-speed
  - nextmode0 = 6
  - Go on to reverse slow-down mode.

- **rslowdwn0**: Reverse slow-down
  - nextmode0 = 7
  - Go on to rotation stop mode.

- **frstop0**: Rotation stop
  - nextmode0 = 0
  - Go on to forward speed-up mode.

**DTCERB.b13 = 1**
- Enable DTC activation by TGI0A interrupts.

**TSR_0 &= H'FE**
- Clear the flag.

**End**
6.4 \text{fslowup0} Function

\begin{itemize}
  \item \texttt{fslowup0}
  \item \texttt{dtjad =\&PTN0.DTC1.MRA}
    \begin{itemize}
      \item Set the address of PTN0 as the start address of the information transferred by the DTC.
    \end{itemize}
  \item PTN0.DTC1.SAR = pattbl
    \begin{itemize}
      \item Set the start address of output pattern table as the transfer source.
    \end{itemize}
  \item PTN0.DTC1.MRA = H'48
    \begin{itemize}
      \item Select repeat transfer mode.
      \item Select byte-size transfer.
      \item Increment SAR after transfer.
    \end{itemize}
  \item PTN0.CRA = H'0404
    \begin{itemize}
      \item Number of transfers = 4
      \item Transfer counter = 4
    \end{itemize}
  \item CYC0.DTC1.SAR = uptbl
    \begin{itemize}
      \item Set the start address of period data table as the transfer source.
    \end{itemize}
  \item CYC0.DTC1.MRA = H'18
    \begin{itemize}
      \item Select normal transfer mode.
      \item Select word-size transfer.
      \item Increment SAR after transfer.
    \end{itemize}
  \item CYC0.CRA = UPTIME
    \begin{itemize}
      \item Transfer counter = UPTIME
    \end{itemize}
\end{itemize}

\text{End}
6.5 fconst0 Function

fconst0

dtcad = &PTN0.DTC1.MRA
Set the address of PTN0 as the start address of the information transferred by the DTC.

PTN0.DTC1.SAR = pattbl
Set the start address of output pattern table as the transfer source.

PTN0.DTC1.MRA = H'48
- Select repeat transfer mode.
- Select byte-size transfer.
- Increment SAR after transfer.

PTN0.CRA = H'0404
- Number of transfers = 4
- Transfer counter = 4

CYC0.DTC1.SAR = &uptbl[UPTIME-1]
Set the end address of period data table as the transfer source.

CYC0.DTC1.MRA = H'10
- Select normal transfer mode.
- Select word-size transfer.
- Fix SAR after transfer.

CYC0.CRA = CNSTTIME
- Transfer counter = CNSTTIME

End
6.6  fslowdwn0 Function

```
dtcad = &PTN0.DTC1.MRA
Set the address of PTN0 as the start address of the information transferred by the DTC.

PTN0.DTC1.SAR = pattbl
Set the start address of output pattern table as the transfer source.

PTN0.DTC1.MRA = H'48
- Select repeat transfer mode.
- Select byte-size transfer.
- Increment SAR after transfer.

PTN0.CRA = H'0404
- Number of transfers = 4
- Transfer counter = 4

CYC0.DTC1.SAR = &uptbl[UPTIME-1]
Set the end address of period data table as the transfer source.

CYC0.DTC1.MRA = H'1C
- Select normal transfer mode.
- Select word-size transfer.
- Decrement SAR after transfer.

CYC0.CRA = UPTIME
- Transfer counter = UPTIME

End
```
6.7 rslowup0 Function

```
dtcad = &PTN0.DTC1.MRA
Set the address of PTN0 as the start address of the information transferred by the DTC.

PTN0.DTC1.SAR = &pattbl[3]
Set the end address of output pattern table as the transfer source.

PTN0.DTC1.MRA = H'4C
- Select repeat transfer mode.
- Select byte-size transfer.
- Decrement SAR after transfer.

PTN0.CRA = H'0404
- Number of transfers = 4
- Transfer counter = 4

CYC0.DTC1.SAR = uptbl
Set the start address of period data table as the transfer source.

CYC0.DTC1.MRA = H'18
- Select normal transfer mode.
- Select word-size transfer.
- Increment SAR after transfer.

CYC0.CRA = UPTIME
- Transfer counter = UPTIME

End
```
6.8  rconst0 Function

```
dtcad = &PTN0.DTC1.MRA
Set the address of PTN0 as the start address of the information transferred by the DTC.

PTN0.DTC1.SAR = &pattbl[3]
Set the end address of output pattern table as the transfer source.

PTN0.DTC1.MRA = H'4C
- Select repeat transfer mode.
- Select byte-size transfer.
- Increment SAR after transfer.

PTN0.CRA = H'0404
- Number of transfers = 4
- Transfer counter = 4

CYC0.DTC1.SAR = &uptbl[UPTIME-1]
Set the end address of period data table as the transfer source.

CYC0.DTC1.MRA = H'10
- Select normal transfer mode.
- Select word-size transfer
- Fix SAR after transfer.

CYC0.CRA = CNSTTIME
- Transfer counter = CNSTTIME

End
```
6.9  rslowdwn0 Function

dtcad = &PTN0.DTC1.MRA
Set the address of PTN0 as the start address of the information transferred by the DTC.

PTN0.DTC1.SAR = &pattbl[3]
Set the end address of output pattern table as the transfer source.

PTN0.DTC1.MRA = H'4C
- Select repeat transfer mode.
- Select byte-size transfer.
- Decrement SAR after transfer.

PTN0.CRA = H'0404
- Number of transfer = 4
- Transfer counter = 4

CYC0.DTC1.SAR
= &uptbl[UPTIME-1]
Set the end address of period data table as the transfer source.

CYC0.DTC1.MRA = H'1C
- Select normal transfer mode.
- Select word-size transfer.
- Decrement SAR after transfer.

CYC0.CRA = UPTIME
- Transfer counter = UPTIME

End
6.10 frstop0 Function

frstop0

dtcad = &CYC0.DTC1.MRA
Set the address of CYC0 as the start address of the information transferred by the DTC.

CYC0.DTC1.SAR = uptbl
Set the address of stop time storage memory as the transfer source.

CYC0.DTC1.MRA = H'10
- Select normal transfer mode.
- Select word-size transfer.
- Fix SAR after transfer.

CYC0.CRA = STOPTIME
- Transfer counter = STOPTIME

End
## Revision Record

<table>
<thead>
<tr>
<th>Rev.</th>
<th>Date</th>
<th>Description</th>
<th>Page</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>Mar.04.05</td>
<td>First edition issued</td>
<td>—</td>
<td></td>
</tr>
</tbody>
</table>
Keep safety first in your circuit designs!

1. Renesas Technology Corp. puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage. Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corp. product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corp. or a third party.

2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.

3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for the latest product information before purchasing a product listed herein.

The information described here may contain technical inaccuracies or typographical errors. Renesas Technology Corp. assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors. Please also pay attention to information published by Renesas Technology Corp. by various means, including the Renesas Technology Corp. Semiconductor home page (http://www.renesas.com).

4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.

5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.

6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce in whole or in part these materials.

7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination. Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.

8. Please contact Renesas Technology Corp. for further details on these materials or the products contained therein.