**Welcome to E-XFL.COM**

## What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

## Applications of "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Obsolete |
| Core Processor | H8/300H |
| Core Size | 16-Bit |
| Speed | 20MHz |
| Connectivity | CANbus, SCI, SSU |
| Peripherals | LVD, POR, PWM, WDT |
| Number of I/O | 45 |
| Program Memory Size | 56KB (56K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 3K x 8 |
| Voltage - Supply (Vcc/Vdd) | 3V ~ 5.5V |
| Data Converters | A/D 8x10b |
| Oscillator Type | Internal |
| Operating Temperature | -20°C ~ 75°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 64-LQFP |
| Supplier Device Package | 64-LFQFP (10x10) |
| Purchase URL | https://www.e-xfl.com/product-detail/renesas-electronics-america/df36057gfpv |

- BCLR instruction executed

```
BCLR    #0,     @PCR5
```

The BCLR instruction is executed for PCR5.

- After executing BCLR instruction

| | **P57** | **P56** | **P55** | **P54** | **P53** | **P52** | **P51** | **P50** |
|---|---|---|---|---|---|---|---|---|
| Input/output | Output | Output | Output | Output | Output | Output | Output | Input |
| Pin state | Low level | High level | Low level | Low level | Low level | Low level | Low level | High level |
| PCR5 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| PDR5 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- Description on operation

1. When the BCLR instruction is executed, first the CPU reads PCR5. Since PCR5 is a write-only register, the CPU reads a value of H'FF, even though the PCR5 value is actually H'3F.
2. Next, the CPU clears bit 0 in the read data to 0, changing the data to H'FE.
3. Finally, H'FE is written to PCR5 and BCLR instruction execution ends.

As a result of this operation, bit 0 in PCR5 becomes 0, making P50 an input port. However, bits 7 and 6 in PCR5 change to 1, so that P57 and P56 change from input pins to output pins. To prevent this problem, store a copy of the PDR5 data in a work area in memory and manipulate data of the bit in the work area, then write this data to PDR5.

- Prior to executing BCLR instruction

```
MOV.B   #3F,  R0L
MOV.B   R0L,  @RAM0
MOV.B   R0L,  @PCR5
```

The PCR5 value (H'3F) is written to a work area in memory (RAM0) as well as to PCR5.

| | **P57** | **P56** | **P55** | **P54** | **P53** | **P52** | **P51** | **P50** |
|---|---|---|---|---|---|---|---|---|
| Input/output | Input | Input | Output | Output | Output | Output | Output | Output |
| Pin state | Low level | High level | Low level | Low level | Low level | Low level | Low level | Low level |
| PCR5 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| PDR5 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| RAM0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |

## 7.4 Flash Memory Programming/Erasing

A software method using the CPU is employed to program and erase flash memory in the on-board programming modes. Depending on the FLMCR1 setting, the flash memory operates in one of the following four modes: Program mode, program-verify mode, erase mode, and erase-verify mode. The programming control program in boot mode and the user program/erase control program in user program mode use these operating modes in combination to perform programming/erasing. Flash memory programming and erasing should be performed in accordance with the descriptions in section 7.4.1, Program/Program-Verify and section 7.4.2, Erase/Erase-Verify, respectively.

### 7.4.1 Program/Program-Verify

When writing data or programs to the flash memory, the program/program-verify flowchart shown in figure 7.3 should be followed. Performing programming operations according to this flowchart will enable data or programs to be written to the flash memory without subjecting the chip to voltage stress or sacrificing program data reliability.

1. Programming must be done to an empty address. Do not reprogram an address to which programming has already been performed.
2. Programming should be carried out 128 bytes at a time. A 128-byte data transfer must be performed even if writing fewer than 128 bytes. In this case, H'FF data must be written to the extra addresses.
3. Prepare the following data storage areas in RAM: A 128-byte programming data area, a 128-byte reprogramming data area, and a 128-byte additional-programming data area. Perform reprogramming data computation according to table 7.4, and additional programming data computation according to table 7.5.
4. Consecutively transfer 128 bytes of data in byte units from the reprogramming data area or additional-programming data area to the flash memory. The program address and 128-byte data are latched in the flash memory. The lower 8 bits of the start address in the flash memory destination area must be H'00 or H'80.
5. The time during which the P bit is set to 1 is the programming time. Table 7.6 shows the allowable programming times.
6. The watchdog timer (WDT) is set to prevent overprogramming due to program runaway, etc. An overflow cycle of approximately 6.6 ms is allowed.
7. For a dummy write to a verify address, write 1-byte data H'FF to an address whose lower 2 bits are B'00. Verify data can be read in words or in longwords from the address to which a dummy write was performed.

The FLMCR1, FLMCR2, and EBR1 settings are retained, however program mode or erase mode is aborted at the point at which the error occurred. Program mode or erase mode cannot be re-entered by re-setting the P or E bit. However, PV and EV bit settings are retained, and a transition can be made to verify mode. Error protection can be cleared only by a power-on reset.

## 7.6 Programmer Mode

In programmer mode, a PROM programmer can be used to perform programming/erasing via a socket adapter, just as a discrete flash memory. Use a PROM programmer that supports the MCU device type with the on-chip 64-kbyte flash memory (FZTAT64V5).

## 7.7 Power-Down States for Flash Memory

In user mode, the flash memory will operate in either of the following states:

- Normal operating mode
  The flash memory can be read and written to at high speed.
- Power-down operating mode
  The power supply circuit of flash memory can be partly halted. As a result, flash memory can be read with low power consumption.
- Standby mode
  All flash memory circuits are halted.

Table 7.7 shows the correspondence between the operating modes of this LSI and the flash memory. In subactive mode, the flash memory can be set to operate in power-down mode with the PDWND bit in FLPWCR. When the flash memory returns to its normal operating state from power-down mode or standby mode, a period to stabilize operation of the power supply circuits that were stopped is needed. When the flash memory returns to its normal operating state, bits STS2 to STS0 in SYSCR1 must be set to provide a wait time of at least 20 µs, even when the external clock is being used.

# Section 8 RAM

This LSI has an on-chip high-speed static RAM. The RAM is connected to the CPU by a 16-bit data bus, enabling two-state access by the CPU to both byte data and word data.

| Product Classification | | RAM Size | RAM Address |
|---|---|---|---|
| Flash memory version | H8/36057F, H8/36037F | 3 kbytes | H'EC00 to H'EFFF, H'F780 to H'FF7F* |
| | H8/36054F, H8/36034F | 2 kbytes | H'F780 to H'FF7F* |
| Masked ROM version | H8/36057, H8/36037 | 2 kbytes | H'EC00 to H'EFFF, H'FB80 to H'FF7F |
| | H8/36036 | 2 kbytes | H'EC00 to H'EFFF, H'FB80 to H'FF7F |
| | H8/36035 | 2 kbytes | H'EC00 to H'EFFF, H'FB80 to H'FF7F |
| | H8/36054, H8/36034 | 2 kbytes | H'EC00 to H'EFFF, H'FB80 to H'FF7F |
| | H8/36033 | 1 kbyte | H'FB80 to H'FF7F |
| | H8/36032 | 1 kbyte | H'FB80 to H'FF7F |

Note: * When the E7 or E8 is used, area H'F780 to H'FB7F must not be accessed.

### 9.4.2 Port Data Register 6 (PDR6)

PDR6 is a general I/O port data register of port 6.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | P67 | 0 | R/W | Stores output data for port 6 pins. |
| 6 | P66 | 0 | R/W | If PDR6 is read while PCR6 bits are set to 1, the value stored in PDR6 are read. If PDR6 is read while PCR6 bits are cleared to 0, the pin states are read regardless of the value stored in PDR6. |
| 5 | P65 | 0 | R/W | |
| 4 | P64 | 0 | R/W | |
| 3 | P63 | 0 | R/W | |
| 2 | P62 | 0 | R/W | |
| 1 | P61 | 0 | R/W | |
| 0 | P60 | 0 | R/W | |

### 9.4.3 Pin Functions

The correspondence between the register specification and the port functions is shown below.

- P67/FTIOD1 pin

| Register | TOER | TFCR | TPMR | TIORC1 | PCR6 | |
|---|---|---|---|---|---|---|
| Bit Name | ED1 | CMD1, CMD0 | PWMD1 | IOD2 to IOD0 | PCR67 | Pin Function |
| Setting Value | 1 | 00 | 0 | 000 or 1XX | 0 | P67 input/FTIOD1 input pin |
| | | | | | 1 | P67 output pin |
| | 0 | 00 | 0 | 001 or 01X | X | FTIOD1 output pin |
| | | | 1 | XXX | | |
| | | Other than 00 | X | XXX | | |

[Legend]

X: Don't care.

# Section 12 Timer Z

The timer Z has a 16-bit timer with two channels. Figures 12.1, 12.2, and 12.3 show the block diagrams of entire timer Z, its channel 0, and its channel 1, respectively. For details on the timer Z functions, refer to table 12.1.

## 12.1 Features

- Capability to process up to eight inputs/outputs
- Eight general registers (GE): four registers for each channel
  - — Independently assignable output compare or input capture functions
- Selection of five counter clock sources: four internal clocks (ϕ, ϕ/2, ϕ/4, and ϕ/8) and an external clock
- Seven selectable operating modes
  - — Output compare function
    Selection of 0 output, 1 output, or toggle output
  - — Input capture function
    Rising edge, falling edge, or both edges
  - — Synchronous operation
    Timer counters_0 and _1 (TCNT_0 and TCNT_1) can be written simultaneously.
    Simultaneous clearing by compare match or input capture is possible.
  - — PWM mode
    Up to six-phase PWM output can be provided with desired duty ratio.
  - — Reset synchronous PWM mode
    Three-phase PWM output for normal and counter phases
  - — Complementary PWM mode
    Three-phase PWM output for non-overlapped normal and counter phases
    The A/D conversion start trigger can be set for PWM cycles.
  - — Buffer operation
    The input capture register can be consisted of double buffers.
    The output compare register can automatically be modified.
- High-speed access by the internal 16-bit bus
  - — 16-bit TCNT and GR registers can be accessed in high speed by a 16-bit bus interface
- Any initial timer output value can be set
- Output of the timer is disabled by external trigger

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 2 | IOA2 | 0 | R/W | I/O Control A2 to A0 |
| 1 | IOA1 | 0 | R/W | GRA is an output compare register: |
| 0 | IOA0 | 0 | R/W | 000: Disables pin output by compare match |
| | | | | 001: 0 output by GRA compare match |
| | | | | 010: 1 output by GRA compare match |
| | | | | 011: Toggle output by GRA compare match |
| | | | | GRA is an input capture register: |
| | | | | 100: Input capture to GRA at the rising edge |
| | | | | 101: Input capture to GRA at the falling edge |
| | | | | 11X: Input capture to GRA at both rising and falling edges |

[Legend]
X: Don't care

**TIORC:** TIORC selects whether GRC or GRD is used as an output compare register or an input capture register. When an output compare register is selected, the output setting is selected. When an input capture register is selected, an input edge of an input capture signal is selected. TIORC also selects the function of FTIOC or FTIOD pin.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | — | 1 | — | Reserved |
| | | | | This bit is always read as 1. |
| 6 | IOD2 | 0 | R/W | I/O Control D2 to D0 |
| 5 | IOD1 | 0 | R/W | GRD is an output compare register: |
| 4 | IOD0 | 0 | R/W | 000: Disables pin output by compare match |
| | | | | 001: 0 output by GRD compare match |
| | | | | 010: 1 output by GRD compare match |
| | | | | 011: Toggle output by GRD compare match |
| | | | | GRD is an input capture register: |
| | | | | 100: Input capture to GRD at the rising edge |
| | | | | 101: Input capture to GRD at the falling edge |
| | | | | 11X: Input capture to GRD at both rising and falling edges |
| 3 | — | 1 | — | Reserved |
| | | | | This bit is always read as 1. |

**Free-Running Count Operation and Periodic Count Operation:** Immediately after a reset, the TCNT counters for channels 0 and 1 are all designated as free-running counters. When the relevant bit in TSTR is set to 1, the corresponding TCNT counter starts an increment operation as a free-running counter. When TCNT overflows, the OVF flag in TSR is set to 1. If the value of the OVIE bit in the corresponding TIER is 1 at this point, timer Z requests an interrupt. After overflow, TCNT starts an increment operation again from H'0000.

Figure 12.8 illustrates free-running counter operation.

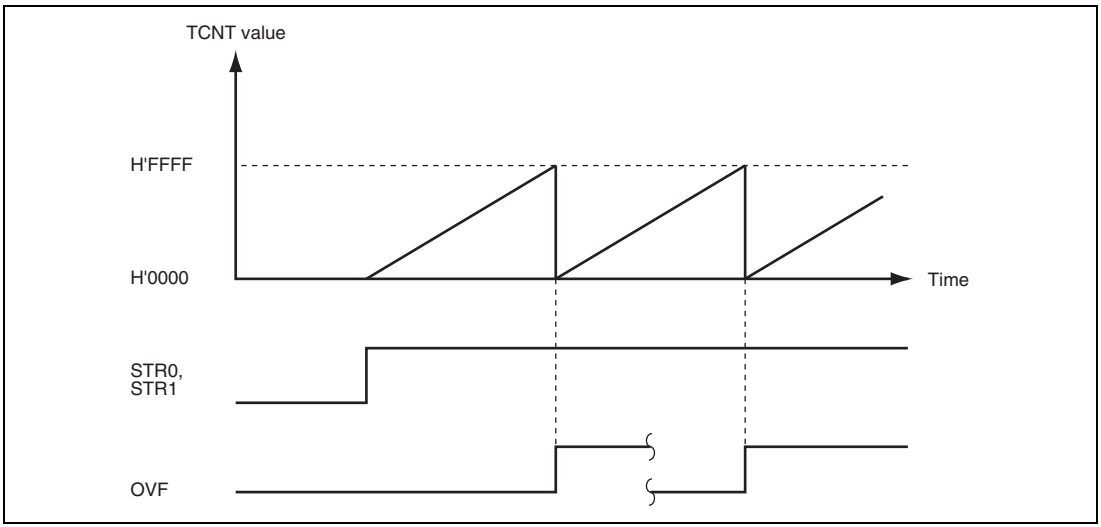


**Figure 12.8 Free-Running Counter Operation**

When compare match is selected as the TCNT clearing source, the TCNT counter for the relevant channel performs periodic count operation. The GR registers for setting the period are designated as output compare registers, and counter clearing by compare match is selected by means of bits CCLR1 and CCLR0 in TCR. After the settings have been made, TCNT starts an increment operation as a periodic counter when the corresponding bit in TSTR is set to 1. When the count value matches the value in GR, the IMFA, IMFB, IMFC, or IMFD flag in TSR is set to 1 and TCNT is cleared to H'0000.

If the value of the corresponding IMIEA, IMIEB, IMIEC, or IMIED bit in TIER is 1 at this point, the timer Z requests an interrupt. After a compare match, TCNT starts an increment operation again from H'0000.

### 12.4.9 Timer Z Output Timing

The outputs of channels 0 and 1 can be disabled or inverted by the settings of TOER and TOCR and the external level.

**Output Disable/Enable Timing of Timer Z by TOER:** Setting the master enable bit in TOER to 1 disables the output of timer Z. By setting the PCR and PDR of the corresponding I/O port beforehand, any value can be output. Figure 12.44 shows the timing to enable or disable the output of timer Z by TOER.

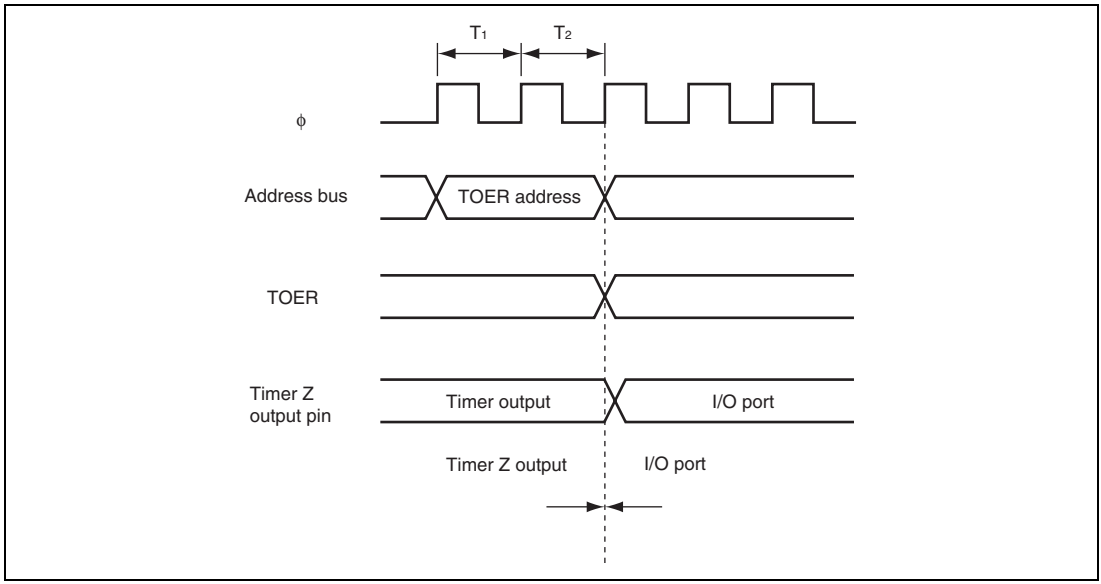


**Figure 12.44 Example of Output Disable Timing of Timer Z by Writing to TOER**

### 14.5.5 Simultaneous Serial Data Transmission and Reception

Figure 14.14 shows a sample flowchart for simultaneous serial transmit and receive operations. The following procedure should be used for simultaneous serial data transmit and receive operations. To switch from transmit mode to simultaneous transmit and receive mode, after checking that the SCI3 has finished transmission and the TDRE and TEND flags are set to 1, clear TE to 0. Then simultaneously set TE and RE to 1 with a single instruction. To switch from receive mode to simultaneous transmit and receive mode, after checking that the SCI3 has finished reception, clear RE to 0. Then after checking that the RDRF and receive error flags (OER, FER, and PER) are cleared to 0, simultaneously set TE and RE to 1 with a single instruction.

### 15.3.2 Master Control Register (MCR)

MCR controls a transition request to halt mode and a software reset request.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 2 | — | All 0 | R/W | Reserved<br>These bits are always read as 0. |
| 1 | HLTRQ | 0 | R/W | Halt Request<br>Halts communication between the TinyCAN and CAN bus. Communication with the CAN bus can be resumed by clearing this bit to 0 and then receiving 11 recessive bits.<br>0: TinyCAN in normal mode<br>1: Halt mode is requested |
| 0 | RSTRQ | 1 | R/W | Reset Request<br>Controls a software reset of the TinyCAN. After a reset has been requested and the initial state is entered, both the RESET bit in GSR and the RHI bit in TCIRR0 are set to 1. When this bit is cleared to 0, communication with the CAN bus is resumed. After powering on, this bit and the RESET bit are always set to 1.<br>0: TinyCAN in normal mode<br>1: Software reset of TinyCAN is requested |

**Internal Arbitration at Transmission:** The TinyCAN transmits untransmitted messages in the priority order from Mailbox 3 to Mailbox 1. The internal arbitration function selects the Mailbox with the highest priority among all transmission request messages. Internal arbitration is based on the three sources given below.

- TXPR/TXCR is set
- Arbitration lost during message transmission
- CAN bus error

**TXPR/TXCR Setting:** Figure 15.7 shows the timing of the TinyCAN internal arbitration caused by the TXPR/TXCR setting. Transmit procedure and operation are as follows.

1. Write data of a transmit message to MCn0, MCn4 to MCn7, and MDn0 to MDn7 [n = 1 to 3] before clearing the MBn bit in MBCR corresponding to the Mailbox of the transmit message to 0 (initial setting).
2. Set the corresponding MBn bit in TXPR to 1 (start condition issuance). Then, the start condition is generated.
3. The internal arbitration for message 1 is determined and the transmit message is transferred to the temporary buffer. After that, even if a transmit request cancellation is issued to the message being transmitted by the DART or MBn bit in TXCR, message 1 is transmitted continuously unless the TinyCAN detects an arbitration loss or error on the CAN bus.
4. After the seventh bit of the EOF has been transmitted (normal message transmission end), the MBn bits in TXPR and TXCR which are corresponding Mailbox are cleared to 0 and the MBn bit in TXACK and the EMPI bit in TCIRR1 are set to 1. At this time, the MBn bit in ABACK is always 0. Then, message transmission is completed.
5. When there is a transmit request other than for message 1, the transmit message is transferred to the temporary buffer and transmitted to the CAN bus after the arbitration for message 2 has been determined. When there is no transmit request other than for message 1, the TinyCAN performs reception.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 2 | CKS2 | 0 | R/W | Transfer clock rate select |
| 1 | CKS1 | 0 | R/W | Sets transfer clock rate (prescaler division ratio) when the internal clock is selected. |
| 0 | CKS0 | 0 | R/W | 000: φ/256<br>001: φ/128<br>010: φ/64<br>011: φ/32<br>100: φ/16<br>101: φ/8<br>110: φ/4<br>111: Reserved |

### 16.3.4 SS Enable Register (SSER)

SSER is a register that sets transmit enable, receive enable, and interrupt enable.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | TE | 0 | R/W | Transmit enable<br>When this bit is 1, transmit operation is enabled. |
| 6 | RE | 0 | R/W | Receive enable<br>When this bit is 1, receive operation is enabled. |
| 5 | RSSTP | 0 | R/W | Receive single stop<br>When this bit is 1, receive operation is completed after receiving one byte. |
| 4 | — | 0 | — | Reserved<br>This bit is always read as 0. |
| 3 | TEIE | 0 | R/W | Transmit End Interrupt Enable<br>When this bit is set to 1, a TEI interrupt request is enabled. |
| 2 | TIE | 0 | R/W | Transmit Interrupt Enable<br>When this bit is set to 1, a TXI interrupt request is enabled. |

## 18.6 Usage Notes

### 18.6.1 Permissible Signal Source Impedance

This LSI's analog input is designed such that conversion accuracy is guaranteed for an input signal for which the signal source impedance is 5 kΩ or less. This specification is provided to enable the A/D converter's sample-and-hold circuit input capacitance to be charged within the sampling time; if the sensor output impedance exceeds 5 kΩ, charging may be insufficient and it may not be possible to guarantee A/D conversion accuracy. However, for A/D conversion in single mode with a large capacitance provided externally, the input load will essentially comprise only the internal input resistance of 10 kΩ, and the signal source impedance is ignored. However, as a low-pass filter effect is obtained in this case, it may not be possible to follow an analog signal with a large differential coefficient (e.g., 5 mV/µs or greater) (see figure 18.6). When converting a high-speed analog signal or converting in scan mode, a low-impedance buffer should be inserted.

### 18.6.2 Influences on Absolute Accuracy

Adding capacitance results in coupling with GND, and therefore noise in GND may adversely affect absolute accuracy. Be sure to make the connection to an electrically stable GND.

Care is also required to ensure that filter circuits do not interfere with digital signals or act as antennas on the mounting board.

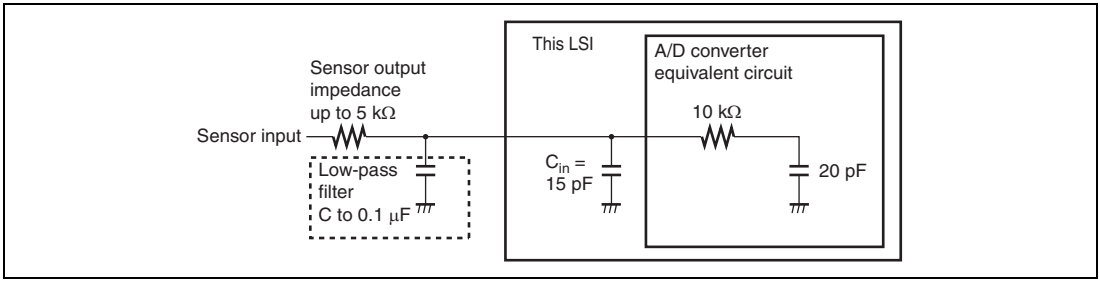


**Figure 18.6 Analog Input Circuit Example**

| Register Name | Abbre-viation | Bit No | Address | Module Name | Data Bus Width | Access State |
|---|---|---|---|---|---|---|
| Message control 0 [4] | MC0[4] | 8 | H'F624 | TinyCAN | 8 | 4 |
| Message control 0 [5] | MC0[5] | 8 | H'F625 | TinyCAN | 8 | 4 |
| Message control 0 [6] | MC0[6] | 8 | H'F626 | TinyCAN | 8 | 4 |
| Message control 0 [7] | MC0[7] | 8 | H'F627 | TinyCAN | 8 | 4 |
| Message control 1 [0] | MC1[0] | 8 | H'F628 | TinyCAN | 8 | 4 |
| Message control 1 [4] | MC1[4] | 8 | H'F62C | TinyCAN | 8 | 4 |
| Message control 1 [5] | MC1[5] | 8 | H'F62D | TinyCAN | 8 | 4 |
| Message control 1 [6] | MC1[6] | 8 | H'F62E | TinyCAN | 8 | 4 |
| Message control 1 [7] | MC1[7] | 8 | H'F62F | TinyCAN | 8 | 4 |
| Message control 2 [0] | MC2[0] | 8 | H'F630 | TinyCAN | 8 | 4 |
| Message control 2 [4] | MC2[4] | 8 | H'F634 | TinyCAN | 8 | 4 |
| Message control 2 [5] | MC2[5] | 8 | H'F635 | TinyCAN | 8 | 4 |
| Message control 2 [6] | MC2[6] | 8 | H'F636 | TinyCAN | 8 | 4 |
| Message control 2 [7] | MC2[7] | 8 | H'F637 | TinyCAN | 8 | 4 |
| Message control 3 [0] | MC3[0] | 8 | H'F638 | TinyCAN | 8 | 4 |
| Message control 3 [4] | MC3[4] | 8 | H'F63C | TinyCAN | 8 | 4 |
| Message control 3 [5] | MC3[5] | 8 | H'F63D | TinyCAN | 8 | 4 |
| Message control 3 [6] | MC3[6] | 8 | H'F63E | TinyCAN | 8 | 4 |
| Message control 3 [7] | MC3[7] | 8 | H'F63F | TinyCAN | 8 | 4 |
| Message data 0 [0] | MD0[0] | 8 | H'F640 | TinyCAN | 8 | 4 |
| Message data 0 [1] | MD0[1] | 8 | H'F641 | TinyCAN | 8 | 4 |
| Message data 0 [2] | MD0[2] | 8 | H'F642 | TinyCAN | 8 | 4 |
| Message data 0 [3] | MD0[3] | 8 | H'F643 | TinyCAN | 8 | 4 |
| Message data 0 [4] | MD0[4] | 8 | H'F644 | TinyCAN | 8 | 4 |
| Message data 0 [5] | MD0[5] | 8 | H'F645 | TinyCAN | 8 | 4 |
| Message data 0 [6] | MD0[6] | 8 | H'F646 | TinyCAN | 8 | 4 |
| Message data 0 [7] | MD0[7] | 8 | H'F647 | TinyCAN | 8 | 4 |
| Message data 1 [0] | MD1[0] | 8 | H'F648 | TinyCAN | 8 | 4 |
| Message data 1 [1] | MD1[1] | 8 | H'F649 | TinyCAN | 8 | 4 |
| Message data 1 [2] | MD1[2] | 8 | H'F64A | TinyCAN | 8 | 4 |
| Message data 1 [3] | MD1[3] | 8 | H'F64B | TinyCAN | 8 | 4 |

## 21.2 Register Bits

The addresses and bit names of the registers in the on-chip peripheral modules are listed below. The 16-bit register is indicated in two rows, 8 bits for each row.

| Register Abbreviation | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Module Name |
|---|---|---|---|---|---|---|---|---|---|
| MCR | — | — | — | — | — | — | HLTRQ | RSTRQ | TinyCAN |
| GSR | — | — | ERPS | HALT | RESET | TCMPL | ECWRG | BOFF | |
| BCR1 | — | TSG22 | TSG21 | TSG20 | TSG13 | TSG12 | TSG11 | TSG10 | |
| BCR0 | SJW1 | SJW0 | BRP5 | BRP4 | BRP3 | BRP2 | BRP1 | BPR0 | |
| MBCR | — | — | — | — | MB3 | MB2 | MB1 | — | |
| TCMR | MSTTC | — | — | — | — | — | PMR97 | PMR96 | |
| TXPR | — | — | — | — | MB3 | MB2 | MB1 | — | |
| TXCR | — | — | — | — | MB3 | MB2 | MB1 | — | |
| TXACK | — | — | — | — | MB3 | MB2 | MB1 | — | |
| ABACK | — | — | — | — | MB3 | MB2 | MB1 | — | |
| RXPR | — | — | — | — | MB3 | MB2 | MB1 | MB0 | |
| RFPR | — | — | — | — | MB3 | MB2 | MB1 | MB0 | |
| TCIRR1 | — | — | — | WUPI | — | — | OVRI | EMPI | |
| TCIRR0 | OVLI | BOFI | EPI | ROWI | TOWI | RFRI | DFRI | RHI | |
| MBIMR | — | — | — | — | MB3 | MB2 | MB1 | MB0 | |
| TCIMR1 | — | — | — | WUPIM | — | — | OVRIM | EMPIM | |
| TCIMR0 | OVLIM | BOFIM | EPIM | ROWIM | TOWIM | RFRIM | DFRIM | RHIM | |
| REC | REC7 | REC6 | REC5 | REC4 | REC3 | REC2 | REC1 | REC0 | |
| TEC | TEC7 | TEC6 | TEC5 | TEC4 | TEC3 | TEC2 | TEC1 | TEC0 | |
| TCR | TSTMD | WREC | FERPS | ATACK | DEC | DRXIN | DTXOT | INTILE | |
| UMSR | — | — | — | — | MB3 | MB2 | MB1 | MB0 | |
| MC 0 [0] | DART | NMC | — | — | DLC3 | DLC2 | DLC1 | DLC0 | |
| MC 0 [4] | ID20 | ID19 | ID18 | RTR | IDE | — | ID17 | ID16 | |
| MC 0 [5] | ID28 | ID27 | ID26 | ID25 | ID24 | ID23 | ID22 | ID21 | |
| MC 0 [6] | ID7 | ID6 | ID5 | ID4 | ID3 | ID2 | ID1 | ID0 | |
| MC 0 [7] | ID15 | ID14 | ID13 | ID12 | ID11 | ID10 | ID9 | ID8 | |
| MC 1 [0] | DART | NMC | — | — | DLC3 | DLC2 | DLC1 | DLC0 | |

## 3. Logic Instructions

| Mnemonic | | Operand Size | Addressing Mode and Instruction Length (bytes) | | | | | | | | | Operation | Condition Code | | | | | | No. of States*1 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | #xx | Rn | @ERn | @(d, ERn) | @–ERn/@ERn+ | @aa | @(d, PC) | @@aa | — | | I | H | N | Z | V | C | Normal | Advanced |
| AND | AND.B #xx:8, Rd | B | 2 | | | | | | | | | Rd8∧#xx:8 → Rd8 | — | — | ↕ | ↕ | 0 | — | 2 | |
| | AND.B Rs, Rd | B | | 2 | | | | | | | | Rd8∧Rs8 → Rd8 | — | — | ↕ | ↕ | 0 | — | 2 | |
| | AND.W #xx:16, Rd | W | 4 | | | | | | | | | Rd16∧#xx:16 → Rd16 | — | — | ↕ | ↕ | 0 | — | 4 | |
| | AND.W Rs, Rd | W | | 2 | | | | | | | | Rd16∧Rs16 → Rd16 | — | — | ↕ | ↕ | 0 | — | 2 | |
| | AND.L #xx:32, ERd | L | 6 | | | | | | | | | ERd32∧#xx:32 → ERd32 | — | — | ↕ | ↕ | 0 | — | 6 | |
| | AND.L ERs, ERd | L | | 4 | | | | | | | | ERd32∧ERs32 → ERd32 | — | — | ↕ | ↕ | 0 | — | 4 | |
| OR | OR.B #xx:8, Rd | B | 2 | | | | | | | | | Rd8∨#xx:8 → Rd8 | — | — | ↕ | ↕ | 0 | — | 2 | |
| | OR.B Rs, Rd | B | | 2 | | | | | | | | Rd8∨Rs8 → Rd8 | — | — | ↕ | ↕ | 0 | — | 2 | |
| | OR.W #xx:16, Rd | W | 4 | | | | | | | | | Rd16∨#xx:16 → Rd16 | — | — | ↕ | ↕ | 0 | — | 4 | |
| | OR.W Rs, Rd | W | | 2 | | | | | | | | Rd16∨Rs16 → Rd16 | — | — | ↕ | ↕ | 0 | — | 2 | |
| | OR.L #xx:32, ERd | L | 6 | | | | | | | | | ERd32∨#xx:32 → ERd32 | — | — | ↕ | ↕ | 0 | — | 6 | |
| | OR.L ERs, ERd | L | | 4 | | | | | | | | ERd32∨ERs32 → ERd32 | — | — | ↕ | ↕ | 0 | — | 4 | |
| XOR | XOR.B #xx:8, Rd | B | 2 | | | | | | | | | Rd8⊕#xx:8 → Rd8 | — | — | ↕ | ↕ | 0 | — | 2 | |
| | XOR.B Rs, Rd | B | | 2 | | | | | | | | Rd8⊕Rs8 → Rd8 | — | — | ↕ | ↕ | 0 | — | 2 | |
| | XOR.W #xx:16, Rd | W | 4 | | | | | | | | | Rd16⊕#xx:16 → Rd16 | — | — | ↕ | ↕ | 0 | — | 4 | |
| | XOR.W Rs, Rd | W | | 2 | | | | | | | | Rd16⊕Rs16 → Rd16 | — | — | ↕ | ↕ | 0 | — | 2 | |
| | XOR.L #xx:32, ERd | L | 6 | | | | | | | | | ERd32⊕#xx:32 → ERd32 | — | — | ↕ | ↕ | 0 | — | 6 | |
| | XOR.L ERs, ERd | L | | 4 | | | | | | | | ERd32⊕ERs32 → ERd32 | — | — | ↕ | ↕ | 0 | — | 4 | |
| NOT | NOT.B Rd | B | | 2 | | | | | | | | ¬ Rd8 → Rd8 | — | — | ↕ | ↕ | 0 | — | 2 | |
| | NOT.W Rd | W | | 2 | | | | | | | | ¬ Rd16 → Rd16 | — | — | ↕ | ↕ | 0 | — | 2 | |
| | NOT.L ERd | L | | 2 | | | | | | | | ¬ Rd32 → Rd32 | — | — | ↕ | ↕ | 0 | — | 2 | |

| Instruction | Mnemonic | Instruction Fetch I | Branch Addr. Read J | Stack Operation K | Byte Data Access L | Word Data Access M | Internal Operation N |
|---|---|---|---|---|---|---|---|
| ROTXR | ROTXR.B Rd | 1 | | | | | |
| | ROTXR.W Rd | 1 | | | | | |
| | ROTXR.L ERd | 1 | | | | | |
| RTE | RTE | 2 | | 2 | | | 2 |
| RTS | RTS | 2 | | 1 | | | 2 |
| SHAL | SHAL.B Rd | 1 | | | | | |
| | SHAL.W Rd | 1 | | | | | |
| | SHAL.L ERd | 1 | | | | | |
| SHAR | SHAR.B Rd | 1 | | | | | |
| | SHAR.W Rd | 1 | | | | | |
| | SHAR.L ERd | 1 | | | | | |
| SHLL | SHLL.B Rd | 1 | | | | | |
| | SHLL.W Rd | 1 | | | | | |
| | SHLL.L ERd | 1 | | | | | |
| SHLR | SHLR.B Rd | 1 | | | | | |
| | SHLR.W Rd | 1 | | | | | |
| | SHLR.L ERd | 1 | | | | | |
| SLEEP | SLEEP | 1 | | | | | |
| STC | STC CCR, Rd | 1 | | | | | |
| | STC CCR, @ERd | 2 | | | | 1 | |
| | STC CCR, @(d:16,ERd) | 3 | | | | 1 | |
| | STC CCR, @(d:24,ERd) | 5 | | | | 1 | |
| | STC CCR,@-ERd | 2 | | | | 1 | 2 |
| | STC CCR, @aa:16 | 3 | | | | 1 | |
| | STC CCR, @aa:24 | 4 | | | | 1 | |
| SUB | SUB.B Rs, Rd | 1 | | | | | |
| | SUB.W #xx:16, Rd | 2 | | | | | |
| | SUB.W Rs, Rd | 1 | | | | | |
| | SUB.L #xx:32, ERd | 3 | | | | | |
| | SUB.L ERs, ERd | 1 | | | | | |
| SUBS | SUBS #1/2/4, ERd | 1 | | | | | |