**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 | Not For New Designs |
| Core Processor | SH2A-FPU |
| Core Size | 32-Bit Single-Core |
| Speed | 200MHz |
| Connectivity | CANbus, Ethernet, I²C, SCI, SPI, USB |
| Peripherals | DMA, PWM, WDT |
| Number of I/O | 112 |
| Program Memory Size | 1MB (1M x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 128K x 8 |
| Voltage - Supply (Vcc/Vdd) | 3V ~ 3.6V |
| Data Converters | A/D 8x12b |
| Oscillator Type | External |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 176-LQFP |
| Supplier Device Package | - |
| Purchase URL | https://www.e-xfl.com/product-detail/renesas-electronics-america/r5f72167adfa-v1 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 0 | IRQ0F | 0 | R/(W) * | 1: IRQn interrupt has occurred<br>[Setting condition]<br>IRQn input is low<br>Edge detection<br>0: IRQn interrupt request is not detected<br>[Clearing conditions]<br>Cleared by reading IRQnF when the IRQnF = 1, then writing 0 to IRQnF<br>Cleared by executing IRQn interrupt exception handling<br>Cleared when DTC is activated by the IRQn interrupt, and the DISEL bit in MRB of DTC is set to 0<br>Cleared when the setting of IRQn1S or IRQn0S of ICR1 is changed<br>1: IRQn interrupt request is detected<br>[Setting condition]<br>Edge corresponding to IRQn1S or IRQn0S of ICR1 has occurred at IRQn pin |

[Legend]
n = 7 to 0

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 2 to 0 | PFC[2:0] | 101 | R/W | Peripheral Clock (Pϕ) Frequency Division Ratio<br>These bits specify the frequency division ratio of the peripheral clock.<br>000: × 1<br>001: × 1/2<br>010: Setting prohibited<br>011: × 1/4<br>100: Setting prohibited<br>101: × 1/8<br>Others: Setting prohibited |

### (4) Burst ROM (Clock Synchronous)

- CS0WCR

| Bit: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | - | - | - | - | - | - | - | - | BW[1:0] | |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W | R/W |

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | W[3:0] | | | | WM | - | - | - | - | - | - |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R | R | R | R | R/W | R/W | R/W | R/W | R/W | R | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 18 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 17, 16 | BW[1:0] | 00 | R/W | Number of Burst Wait Cycles<br>Specify the number of wait cycles to be inserted between the second or subsequent access cycles in burst access.<br>00: No cycle<br>01: 1 cycle<br>10: 2 cycles<br>11: 3 cycles |
| 15 to 11 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |

**Table 11.28 TIORU_5, TIORV_5, and TIORW_5 (Channel 5)**

| | | | | | Description | |
|---|---|---|---|---|---|---|
| Bit 4 IOC4 | Bit 3 IOC3 | Bit 2 IOC2 | Bit 1 IOC1 | Bit 0 IOC0 | TGRU_5, TGRV_5, and TGRW_5 Function | TIC5U, TIC5V, and TIC5W Pin Function |
| 0 | 0 | 0 | 0 | 0 | Compare match register | Compare match |
| | | | | 1 | | Setting prohibited |
| | | | 1 | X | | Setting prohibited |
| | | 1 | X | X | | Setting prohibited |
| | 1 | X | X | X | | Setting prohibited |
| 1 | 0 | 0 | 0 | 0 | Input capture register | Setting prohibited |
| | | | | 1 | | Input capture at rising edge |
| | | | 1 | 0 | | Input capture at falling edge |
| | | | | 1 | | Input capture at both edges |
| | | 1 | X | X | | Setting prohibited |
| | 1 | 0 | 0 | 0 | | Setting prohibited |
| | | | | 1 | | Measurement of low pulse width of external input signal |
| | | | | | | Capture at trough in complementary PWM mode |
| | | | 1 | 0 | | Measurement of low pulse width of external input signal |
| | | | | | | Capture at crest in complementary PWM mode |
| | | | | 1 | | Measurement of low pulse width of external input signal |
| | | | | | | Capture at crest and trough in complementary PWM mode |
| | | 1 | 0 | 0 | | Setting prohibited |
| | | | | 1 | | Measurement of high pulse width of external input signal |
| | | | | | | Capture at trough in complementary PWM mode |
| | | | 1 | 0 | | Measurement of high pulse width of external input signal |
| | | | | | | Capture at crest in complementary PWM mode |
| | | | | 1 | | Measurement of high pulse width of external input signal |
| | | | | | | Capture at crest and trough in complementary PWM mode |

[Legend]

X: Don't care

- Basic Operation Example of A/D Converter Start Request Delaying Function

  Figure 11.80 shows a basic example of A/D converter request signal (TRG4AN) operation when the trough of TCNT_4 is specified for the buffer transfer timing and an A/D converter start request signal is output during TCNT_4 down-counting.

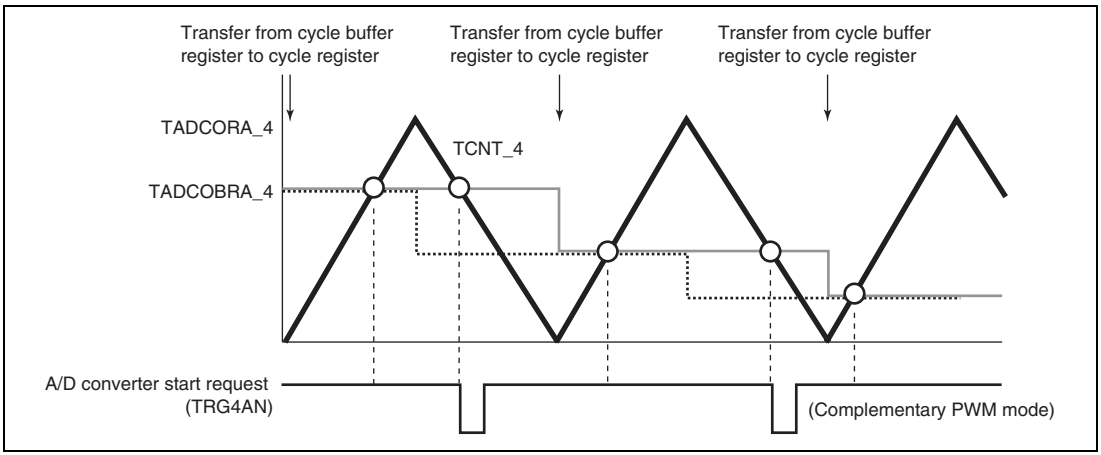


**Figure 11.80 Basic Example of A/D Converter Start Request Signal (TRG4AN) Operation**

- Buffer Transfer

  The data in the timer A/D converter start request cycle set registers (TADCORA_4 and TADCORB_4) is updated by writing data to the timer A/D converter start request cycle set buffer registers (TADCOBRA_4 and TADCOBRB_4). Data is transferred from the buffer registers to the respective cycle set registers at the timing selected with the BF1 and BF0 bits in the timer A/D converter start request control register (TADCR_4).

- A/D Converter Start Request Delaying Function Linked with Interrupt Skipping

  A/D converter start requests (TRG4AN and TRG4BN) can be issued in coordination with interrupt skipping by making settings in the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in the timer A/D converter start request control register (TADCR).

  Figure 11.81 shows an example of A/D converter start request signal (TRG4AN) operation when TRG4AN output is enabled during TCNT_4 up-counting and down-counting and A/D converter start requests are linked with interrupt skipping.

  Figure 11.82 shows another example of A/D converter start request signal (TRG4AN) operation when TRG4AN output is enabled during TCNT_4 up-counting and A/D converter start requests are linked with interrupt skipping.

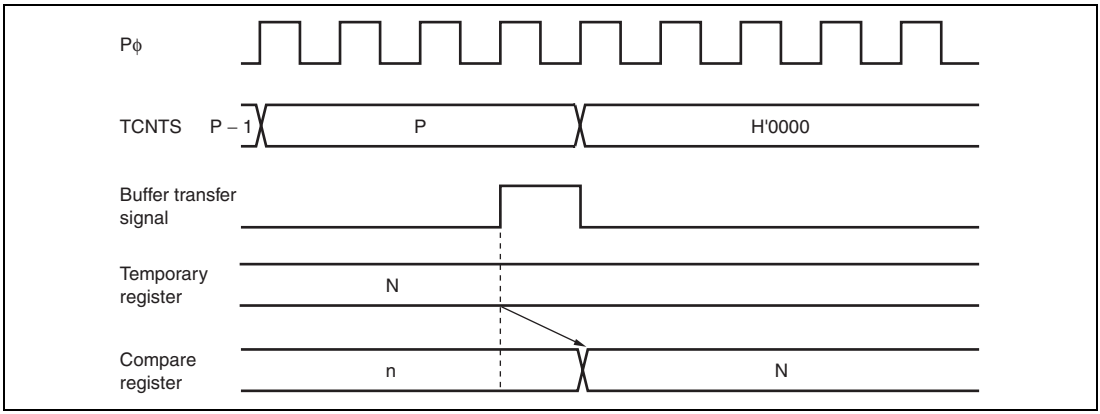


**Figure 11.108 Transfer Timing from Temporary Register to Compare Register**

### 11.6.2 Interrupt Signal Timing

#### (1) TGF Flag Setting Timing in Case of Compare Match

Figures 11.109 and 110 show the timing for setting of the TGF flag in TSR on compare match, and TGI interrupt request signal timing.

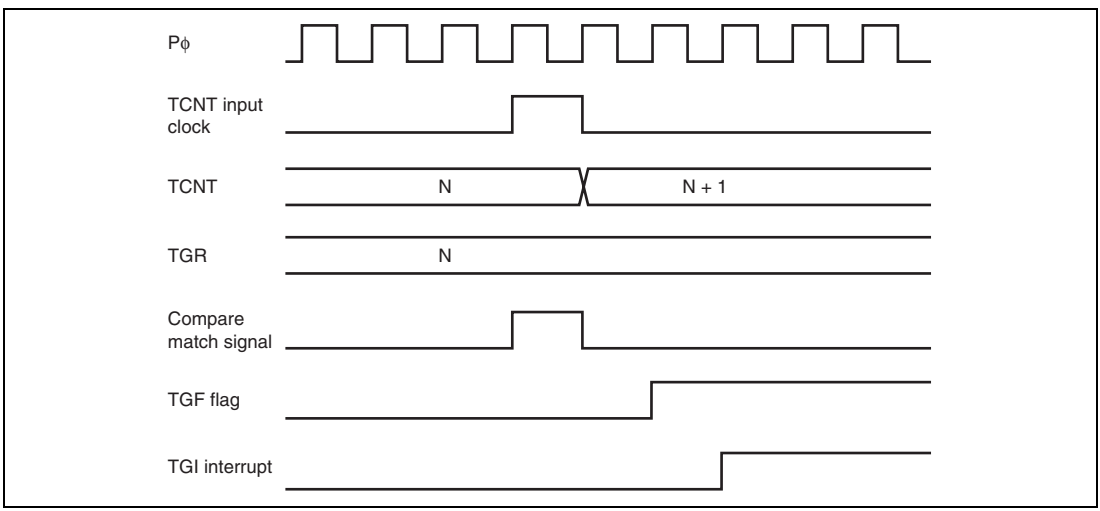


**Figure 11.109 TGI Interrupt Timing (Compare Match)**

### 20.3.6 A/D Data Registers 0 to 7 (ADDR0 to ADDR7)

ADDRs are 16-bit read-only registers. The conversion result for each analog input channel is stored in ADDR with the corresponding number. (See table 20.4.)

The converted 12-bit data is stored in bits 11 to 0.

The initial value of ADDR is H'0000.

After ADDR is read, ADDR can be automatically cleared to H'0000 by setting the ACE bit in ADCR to 1.

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | ADD[11:0] | | | | | | | | | | | |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 to 12 | — | All 0 | R | Reserved |
| 11 to 0 | ADD[11:0] | All 0 | R | 12-bit data |

**Table 20.4 Correspondence between Analog Channels and Registers (ADDR0 to ADDR11)**

| Analog Input Channels | A/D Data Registers |
|---|---|
| AN0 | ADDR0 |
| AN1 | ADDR1 |
| AN2 | ADDR2 |
| AN3 | ADDR3 |
| AN4 | ADDR4 |
| AN5 | ADDR5 |
| AN6 | ADDR6 |
| AN7 | ADDR7 |

### 24.3.37 USB Trigger Register 1 (USBTRG1)

USBTRG1 is a write-only register that generates one-shot triggers to control the transmit/receive sequence for each endpoint. The read value of this register is undefined. Do not write a value to this register using the read value, such as a bit manipulation instruction.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | EP3 PKTE | EP2 PKTE | EP1 RDFN |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | - | - | - | - | - | W | W | W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 3 | — | All 0 | — | Reserved<br>The write value should always be 0. |
| 2 | EP3PKTE | 0 | W | EP3 Packet Enable<br>After one packet of data has been written to the endpoint 3 transmit FIFO buffer, the transmit data is fixed by writing 1 to this bit. |
| 1 | EP2PKTE | 0 | W | EP2 Packet Enable<br>After one packet of data has been written to the endpoint 2 transmit FIFO buffer, the transmit data is fixed by writing 1 to this bit. |
| 0 | EP1RDFN | 0 | W | EP1 Read Complete<br>Write 1 to this bit after one packet of data has been read from the endpoint 1 receive FIFO buffer. The endpoint 1 receive FIFO buffer has a dual-FIFO configuration. Writing 1 to this bit initializes the FIFO that was read, enabling the next packet to be received. |

### 24.3.55 USB Transceiver Test Register 0 (USBTRNTREG0)

USBTRNTREG0 is a test register that controls the on-chip transceiver output signals. Setting PTSTE = 1 enables the transceiver output signals (USD+, USD-) to be set arbitrarily. Table 24.5 shows the USBTRNTREG0 setting and pin output state.

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PTSTE | - | - | - | SUS PEND | txenl | txse0 | txdata |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W | R | R | R | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | PTSTE | 0 | R/W | Pin Test Enable<br>Enables test control for the on-chip transceiver output pins (USD+/USD-). |
| 6 to 4 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 3 | SUSPEND | 0 | R/W | On-Chip Transceiver Output Signal Setting<br>SUSPEND: Sets the on-chip transceiver suspend (SUSPEND) signal.<br>txenl: Sets the on-chip transceiver output enable (txenl) signal.<br>txse0: Sets the on-chip transceiver single-ended 0 (txse0) signal.<br>txdata: Sets the on-chip transceiver data (txdata) signal. |
| 2 | txenl | 0 | R/W | |
| 1 | txse0 | 0 | R/W | |
| 0 | txdata | 0 | R/W | |

## 24.8 DMA Transfer

### 24.8.1 Overview

This module allows DMA transfer for endpoints 1, 2, 4, and 5, excluding transfer of word and longword. If endpoint 1 contains at least one byte of valid receive data, a DMA transfer request is issued to endpoint 1. If there is no valid data in endpoint 2, a DMA transfer request is issued to endpoint 2. If endpoint 4 contains at least one byte of valid receive data, a DMA transfer request is issued to endpoint 4. If there is no valid data in endpoint 5, a DMA transfer request is issued to endpoint 5.

When EP1DMAE or EP4DMAE in the USBDMA setting register is set to 1 to allow DMA transfer, 0-length data received for endpoint 1 or 4 is ignored. When DMA transfer is set, it is unnecessary to write 1 to the EP1RDFN, EP2PKTE, EP4RDFN, and EP5PKTE bits in USBTRG1 or USBTRG2. (However, the PKTE bit in USBTRG1 or USBTRG2 must be set to 1 for data with a size less than the maximum number of bytes.) For EP1 and EP4, the FIFO buffer automatically becomes empty when the received data has been completely read. For EP2 and EP5, the FIFO automatically becomes full when the maximum number of bytes (64 bytes) is written to the FIFO, allowing the data in the FIFO to be transmitted. (See figures 24.16 and 24.19.)

### 24.8.2 DMA Transfer for Endpoints 1 and 4

If the received data for EP1 is transferred by DMA, when the currently selected data FIFO becomes empty, processing equivalent to writing 1 to the EP1RDFN bit in USBTRG1 is automatically performed in the module. Therefore, do not write 1 to the EP1RDFN bit in USBTRG1 after reading the data on one side of the FIFO. If 1 is written to the EP1RDFN bit, correct operation cannot be guaranteed.

For example, if 150-byte data is received from the host, processing equivalent to writing 1 to the EP1RDFN bit in USBTRG1 is automatically performed internally in the three places in figure 24.16. Since this processing is performed when the data on the currently selected FIFO becomes empty, the processing is automatically performed in the same way even if data of 64 bytes or less is transferred.

Similarly, if the received data for EP4 is transferred by DMA, when the currently selected data FIFO becomes empty, processing equivalent to writing 1 to the EP4RDFN bit in USBTRG2 is automatically performed in the module. Therefore, do not write 1 to the EP4RDFN bit in USBTRG2 after reading the data on one side of the FIFO. If 1 is written to the EP4RDFN bit, correct operation cannot be guaranteed.

| Name | Abbreviation | R/W | Address | Access Size |
|---|---|---|---|---|
| PAUSE frame receive counter register | RFCF | R/W | H'FFFC 3160 | 32 |
| PAUSE frame retransmit counter register | TPAUSECR | R | H'FFFC 3168 | 32 |
| Broadcast frame receive count register | BCFRR | R | H'FFFC 316C | 32 |

### 25.3.25 PAUSE Frame Retransmit Counter Register (TPAUSECR)

TPAUSECR is a counter that indicates the number of times a PAUSE frame is retransmitted.

| Bit: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | – | – | – | – | – | – | – | – | TXP[7:0] | | | | | | | |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 8 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 7 to 0 | TXP[7:0] | All 0 | R | PAUSE Frame Retransmit Count |

### 26.2.5 Receive Descriptor List Start Address Register (RDLAR)

RDLAR is a 32-bit readable/writable register that specifies the start address of the receive descriptor list. Descriptors have a boundary configuration in accordance with the descriptor length indicated by the DL bits in EDMR. This register must not be modified during reception, and must be modified while the RR bit in the E-DMAC receive request register (EDRRR) is 0 (reception-disabled state).

| Bit: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RDLA[31:16] | | | | | | | | | | | | | | | |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RDLA[15:0] | | | | | | | | | | | | | | | |
| Initial value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W: | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | RDLA[31:0] | All 0 | R/W | Receive Descriptor Start Address<br>The lower bits are set according to the specified descriptor length.<br>16-byte boundary: RDLA[3:0] = 0000<br>32-byte boundary: RDLA[4:0] = 00000<br>64-byte boundary: RDLA[5:0] = 000000 |

### 27.5.3 Automatic Adjustment of Bit Rate

When this LSI is started in boot mode, it measures the low-level (H'00) period of the data that is continuously sent from the host in asynchronous SCI communications. During this measurement, set the SCI transmit/receive format to 8-bit data, 1 stop bit, and no parity, and set the bit rate to 9,600 bps or 19,200 bps. This LSI calculates the bit rate of the host SCI by means of the measured low-level period, and then sends H'00 to the host after completing the bit rate adjustment. When the host has received H'00 successfully, it must send H'55 to this LSI. If the host has failed to receive H'00, restart this LSI in boot mode to calculate and adjust the bit rate again. When this LSI has received H'55, it returns H'E6 to the host, or when it has failed to receive H'55, it returns H'FF.

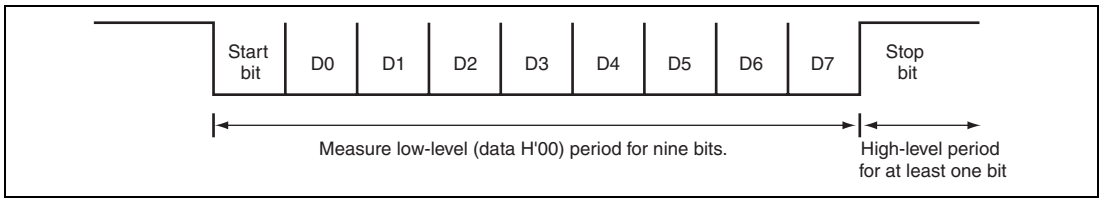


**Figure 27.7 SCI Transmit/Receive Format for Automatic Adjustment of Bit Rate**

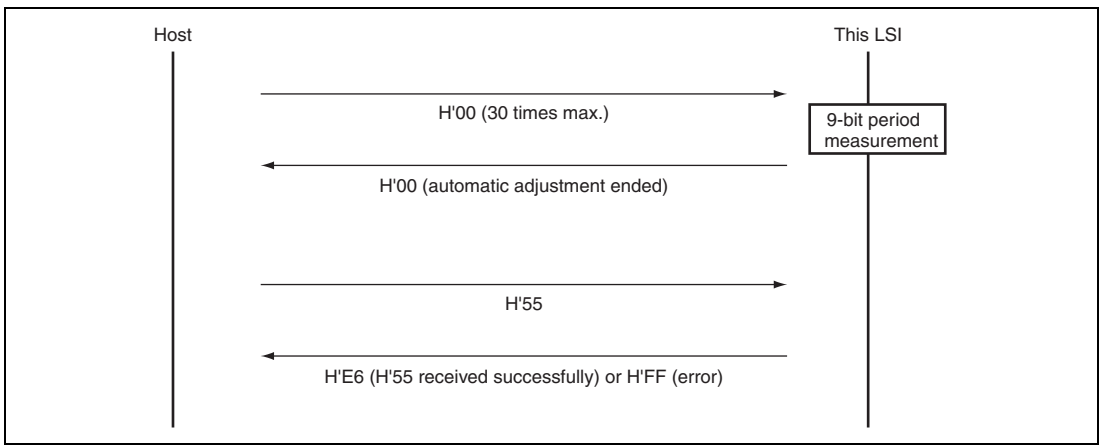


**Figure 27.8 Communication Sequence between Host and This LSI**

The bit rate may not be adjusted correctly depending on the bit rate of the host SCI or the peripheral clock frequency of this LSI. Satisfy the SCI communications condition as shown in table 27.5.

### (10) Programming Size Inquiry

In response to a programming size inquiry command sent from the host, this LSI returns the programming size.

| | | | | |
|---|---|---|---|---|
| Command | H'27 | | | |
| Response | H'37 | Size | Programming size | SUM |

[Legend]

Size (1 byte): Number of characters included in the programming size field (fixed at two)

Programming size (2 bytes): Programming unit (bytes)

SUM (1 byte): Checksum