Introduction

This application note is intended for users who understand the functions of the RL78 microcontrollers and who will use this product to design application systems.

The purpose of this application note is to help users understand how to develop dedicated flash memory programmers for rewriting the internal flash memory of the RL78 microcontrollers.

NOTICE:
There are corrections and additions on page 21, 75 and 79 in this document.
3.8 Checksum Command ............................................................................................................................. 30
   3.8.1 Description ........................................................................................................................................ 30
   3.8.2 Command frame and status frame .................................................................................................... 30
   3.8.3 Checksum data frame ....................................................................................................................... 30
3.9 Security Set Command .............................................................................................................................. 31
   3.9.1 Description ........................................................................................................................................ 31
   3.9.2 Command frame and status frame .................................................................................................... 31
   3.9.3 Data frame and status frame ............................................................................................................ 32
3.10 Security Get Command ............................................................................................................................ 34
   3.10.1 Description ........................................................................................................................................ 34
   3.10.2 Command frame and status frame .................................................................................................... 34
   3.10.3 Data frame and security flag ............................................................................................................ 35
3.11 Security Release Command ...................................................................................................................... 36
   3.11.1 Description ........................................................................................................................................ 36
   3.11.2 Command frame and status frame .................................................................................................... 36
CHAPTER 4 UART COMMUNICATION MODE ........................................................................................................ 37
4.1 Command Frame Transmission Processing Flowchart ............................................................................ 37
4.2 Data Frame Transmission Processing Flowchart ..................................................................................... 38
4.3 Data Frame Reception Processing Flowchart ........................................................................................... 39
4.4 Reset Command ........................................................................................................................................ 40
   4.4.1 Processing sequence chart ............................................................................................................... 40
   4.4.2 Description of processing sequence ................................................................................................. 41
   4.4.3 Status at processing completion ........................................................................................................ 41
   4.4.4 Flowchart .......................................................................................................................................... 42
4.5 Baud Rate Set Command ........................................................................................................................... 43
   4.5.1 Processing sequence chart ............................................................................................................... 43
   4.5.2 Description of processing sequence ................................................................................................. 44
   4.5.3 Status at processing completion ........................................................................................................ 44
   4.5.4 Flowchart .......................................................................................................................................... 45
4.6 Block Erase Command .............................................................................................................................. 46
   4.6.1 Processing sequence chart ............................................................................................................... 46
   4.6.2 Description of processing sequence ................................................................................................. 47
   4.6.3 Status at processing completion ........................................................................................................ 47
   4.6.4 Flowchart .......................................................................................................................................... 48
4.7 Programming Command ........................................................................................................................... 49
   4.7.1 Processing sequence chart ............................................................................................................... 49
   4.7.2 Description of processing sequence ................................................................................................. 50
   4.7.3 Status at processing completion ........................................................................................................ 51
   4.7.4 Flowchart .......................................................................................................................................... 52
4.8 Verify Command ...................................................................................................................................... 53
   4.8.1 Processing sequence chart ............................................................................................................... 53
   4.8.2 Description of processing sequence ................................................................................................. 54
   4.8.3 Status at processing completion ........................................................................................................ 54
   4.8.4 Flowchart .......................................................................................................................................... 55
4.9 Block Blank Check Command .................................................................................................................... 56
   4.9.1 Processing sequence chart ............................................................................................................... 56
   4.9.2 Description of processing sequence ................................................................................................. 57
   4.9.3 Status at processing completion ........................................................................................................ 57
   4.9.4 Flowchart .......................................................................................................................................... 58
4.10 Silicon Signature Command ................................................................. 59
4.10.1 Processing sequence chart .............................................................. 59
4.10.2 Description of processing sequence .............................................. 60
4.10.3 Status at processing completion .................................................... 60
4.10.4 Flowchart .................................................................................... 61

4.11 Checksum Command ........................................................................ 62
4.11.1 Processing sequence chart .............................................................. 62
4.11.2 Description of processing sequence .............................................. 63
4.11.3 Status at processing completion .................................................... 63
4.11.4 Flowchart .................................................................................... 64

4.12 Security Set Command ................................................................. 65
4.12.1 Processing sequence chart .............................................................. 65
4.12.2 Description of processing sequence .............................................. 66
4.12.3 Status at processing completion .................................................... 66
4.12.4 Flowchart .................................................................................... 67

4.13 Security Get Command ................................................................. 68
4.13.1 Processing sequence chart .............................................................. 68
4.13.2 Description of processing sequence .............................................. 69
4.13.3 Status at processing completion .................................................... 69
4.13.4 Flowchart .................................................................................... 70

4.14 Security Release Command .......................................................... 71
4.14.1 Processing sequence chart .............................................................. 71
4.14.2 Description of processing sequence .............................................. 72
4.14.3 Status at processing completion .................................................... 72
4.14.4 Flowchart .................................................................................... 73

CHAPTER 5 FLASH MEMORY PROGRAMMING PARAMETER CHARACTERISTICS ....... 75

5.1 Flash Memory Parameter Characteristics of RL78 ............................. 75
5.1.1 Flash memory parameter characteristics in full-speed mode .......... 75
5.1.2 Flash memory parameter characteristics in wide-voltage mode .......... 79

5.2 UART Communication Mode .......................................................... 83
CHAPTER 1  FLASH MEMORY PROGRAMMING

To rewrite the contents of the internal flash memory of the RL78, a dedicated flash memory programmer (hereafter referred to as the "programmer") is usually used. This Application Note explains how to develop a dedicated programmer.

1.1 Overview

The RL78 incorporates firmware that controls flash memory programming. The programming to the internal flash memory is performed by transmitting/receiving commands between the programmer and the RL78 via serial communication.

Figure 1-1.  System Outline of Flash Memory Programming in RL78

![Diagram of System Outline of Flash Memory Programming in RL78]
1.2 Communication Modes

As serial communications for writing the flash memory, single-wire UART communication or two-wire UART communication can be used. By exchanging the master and slave, an optimum communication can be realized.

1.2.1 Single-wire UART communication

The TOOL0 pin is used for single-wire UART communication. The communication conditions are shown below.

Table 1-1. Single-wire UART Communication Conditions

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baud rate</td>
<td>Communication is performed at 115,200 bps until the Baud Rate Set command for baud rate setting command processing is transmitted. The transmission rate is changed to the baud rate set by the Baud Rate Set command from the transmission of the Reset command for baud rate command processing. For details of the settable baud rate, refer to 3.2 Baud Rate Set Command.</td>
</tr>
<tr>
<td>Parity bit</td>
<td>None</td>
</tr>
<tr>
<td>Data length</td>
<td>8 bits (LSB first)</td>
</tr>
<tr>
<td>Start bit</td>
<td>1 bit</td>
</tr>
<tr>
<td>Stop bit</td>
<td>2 bits (programmer → RL78)/1 bit (RL78 → programmer)</td>
</tr>
</tbody>
</table>
1.2.2 Two-wire UART communication

**Figure 1-3. Two-wire UART Communication**

![Diagram of two-wire UART communication](image)

TxD and RxD pins are used for two-wire UART communication. The communication conditions are shown below.

**Table 1-2. Two-wire UART Communication Conditions**

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baud rate</td>
<td>Communication is performed at 115,200 bps until the Baud Rate Set command for baud rate setting command processing is transmitted. The transmission rate is changed to the baud rate set by the Baud Rate Set command from the transmission of the Reset command for baud rate command processing. For details of the settable baud rate, refer to 3.2 Baud Rate Set Command.</td>
</tr>
<tr>
<td>Parity bit</td>
<td>None</td>
</tr>
<tr>
<td>Data length</td>
<td>8 bits (LSB first)</td>
</tr>
<tr>
<td>Start bit</td>
<td>1 bit</td>
</tr>
<tr>
<td>Stop bit</td>
<td>2 bits (programmer → RL78)/1 bit (RL78 → programmer)</td>
</tr>
</tbody>
</table>
1.3 Command List and Status List

The flash memory incorporated in the RL78 can be rewritten by using the commands listed in Table 1-2. The programmer transmits commands to control these functions to the RL78, and checks the response status sent from the RL78, to manipulate the flash memory.

1.3.1 Command list

The commands used by the programmer and their functions are listed below.

<table>
<thead>
<tr>
<th>Command Number</th>
<th>Command Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>00H</td>
<td>Reset</td>
<td>Detects synchronization in communication.</td>
</tr>
<tr>
<td>22H</td>
<td>Block Erase</td>
<td>Erases a specified area in the flash memory.</td>
</tr>
<tr>
<td>40H</td>
<td>Programming</td>
<td>Writes data to a specified area in the flash memory.</td>
</tr>
<tr>
<td>13H</td>
<td>Verify</td>
<td>Compares the contents in a specified area in the flash memory with the data transmitted from the programmer.</td>
</tr>
<tr>
<td>32H</td>
<td>Block Blank Check</td>
<td>Checks the erase status of a specified block in the flash memory.</td>
</tr>
<tr>
<td>9AH</td>
<td>Baud Rate Set</td>
<td>Sets a baud rate and a voltage.</td>
</tr>
<tr>
<td>C0H</td>
<td>Silicon Signature</td>
<td>Reads RL78 information (such as product name and flash memory configuration).</td>
</tr>
<tr>
<td>A0H</td>
<td>Security Set</td>
<td>Sets a security flag, boot block cluster block number, and FSW.</td>
</tr>
<tr>
<td>A1H</td>
<td>Security Get</td>
<td>Reads a security flag, boot block cluster block number, boot area exchange flag, and FSW (flash option).</td>
</tr>
<tr>
<td>A2H</td>
<td>Security Release</td>
<td>Initializes all flash options.</td>
</tr>
<tr>
<td>B0H</td>
<td>Checksum</td>
<td>Reads the checksum value of data in a specified area.</td>
</tr>
</tbody>
</table>
1.3.2 Status list

The following table lists the status codes the programmer receives from the RL78.

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>04H</td>
<td>Command number error</td>
<td>Error returned if a command not supported is received</td>
</tr>
<tr>
<td>05H</td>
<td>Parameter error</td>
<td>Error returned if the value of a parameter to be appended to a command is not appropriate.</td>
</tr>
<tr>
<td>06H</td>
<td>Normal acknowledgment</td>
<td>Normal acknowledgment</td>
</tr>
<tr>
<td>07H</td>
<td>Checksum error</td>
<td>Error returned if transmitted data frame has an abnormality</td>
</tr>
<tr>
<td>0FH</td>
<td>Verify error</td>
<td>Error returned if a verify error has occurred upon verifying data transmitted from the programmer</td>
</tr>
<tr>
<td>10H</td>
<td>Protect error</td>
<td>Error returned if an attempt is made to execute processing that is prohibited by the Security Set command</td>
</tr>
<tr>
<td>15H</td>
<td>Negative acknowledgment (NACK)</td>
<td>Negative acknowledgment</td>
</tr>
<tr>
<td>1AH</td>
<td>Erase error</td>
<td>Erase error</td>
</tr>
<tr>
<td>1BH</td>
<td>IVerify error/Blank error</td>
<td>Internal verify error or blank check error</td>
</tr>
<tr>
<td>1CH</td>
<td>Write error</td>
<td>Write error</td>
</tr>
</tbody>
</table>

Reception of a checksum error or NACK is treated as an immediate abnormal end in this manual. When a dedicated programmer is developed, however, the processing may be retried without problem from the wait immediately before transmission of the command that results a checksum error or NACK. In this event, limiting the retry count is recommended for preventing infinite repetition of the retry operation.

Although not listed in the above table, if a time-out error (BUSY time-out or time-out in data frame reception during UART communication) occurs, it is recommended to shutdown the power supply to the RL78 (refer to 1.5 Shutting Down Target Power Supply) and then connect the power supply again.
1.4 Power Application and Setting Flash Memory Programming Mode

To rewrite the contents of the flash memory with the programmer, the RL78 must first be set to the flash memory programming mode (serial programming mode).

If the TOOL0 pin is at the low level on reset release, the RL78 is first set to the pre-mode. After data for setting a communication mode and the Baud Rate Set command have been transmitted, the RL78 is set to an operation mode of the serial programming mode.

The following figure illustrates a timing chart for setting the flash memory programming mode and selecting the communication mode.

Figure 1-4. Setting Flash Memory Programming Mode and Selecting Communication Mode

Single-wire UART

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Power (VDD) application</td>
</tr>
<tr>
<td>2</td>
<td>TOOL0 = Low level</td>
</tr>
<tr>
<td>3</td>
<td>Reset release</td>
</tr>
<tr>
<td>4</td>
<td>TOOL0 = High level</td>
</tr>
<tr>
<td>5</td>
<td>Start of 1-byte data transmission</td>
</tr>
<tr>
<td>6</td>
<td>End of 1-byte data transmission</td>
</tr>
<tr>
<td>7</td>
<td>Completion of Baud Rate Set command</td>
</tr>
</tbody>
</table>

Two-wire UART

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Power (VDD) application</td>
</tr>
<tr>
<td>2</td>
<td>TOOL0 = Low level</td>
</tr>
<tr>
<td>3</td>
<td>Reset release</td>
</tr>
<tr>
<td>4</td>
<td>TOOL0 = High level</td>
</tr>
<tr>
<td>5</td>
<td>Start of 1-byte data transmission</td>
</tr>
<tr>
<td>6</td>
<td>End of 1-byte data transmission</td>
</tr>
<tr>
<td>7</td>
<td>Completion of Baud Rate Set command</td>
</tr>
</tbody>
</table>
After reset release, 1-byte data is transmitted at 115,200 bps to set the RL78 to the serial programming mode and determine a communication mode. (Note, however, that this data can be set to 00H in the two-wire UART mode even by low-level control at 78.125 μs).

The relationship between the 1-byte data and communication interface is shown below.

<table>
<thead>
<tr>
<th>1-byte Data</th>
<th>Communication Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>3AH</td>
<td>Single-wire UART</td>
</tr>
<tr>
<td>00H</td>
<td>Two-wire UART</td>
</tr>
</tbody>
</table>
1.4.1 Mode setting flowchart

Transition processing to programming mode

-Reset pin low output
  -Tool0 low output
    -VDD pin high output (Target power supply on)
      -Wait
        -Reset pin high output
          -Wait
            -Tool0 pin high output
              -Wait
                -1-byte data (single-wire/two-wire identified value) transmission
                  -Wait
                    -Baud Rate Set command processing
                      -Normal completion

\[t_{TR}\]
\[t_{RT}\]
\[t_{TM}\]
\[t_{MB}\]
1.5 Shutting Down Target Power Supply

After each command execution is completed, shut down the power supply to the target after setting the \texttt{RESET} pin to low level, as shown below.

Set other pins to Hi-Z when shutting down the power supply to the target.

\textbf{Caution} Shutting down the power supply and inputting a reset during command processing are prohibited.

![Figure 1-5. Timing for Terminating Flash Memory Programming Mode](image)

1.6 Command Execution Flow at Flash Memory Rewriting

Figure 1-6 illustrates the basic flowchart when flash memory rewriting is performed with the programmer.

Other than commands shown in Figure 1-6, the Verify command and Checksum command are also supported.
Figure 1-6. Basic Flowchart for Flash Memory Rewrite Processing

Basic flow

Power application to target (see Figure 1-4)

Mode setting (reset release) (see 1.4)

Baud rate setting (see 3.2)

Silicon signature acquisition (Silicon Signature command) (see 3.7)

Command execution

Processing completed?

Target power shutdown processing (see 1.5)

End

Remark The example of each command execution is shown in Figure 1-7.

Reset input and power shutdown during rewriting is prohibited because security information may be lost.
This command is used to check whether data communication between programmer and target device was normally completed.

Figure 1-7. General Command Execution Flow at Flash Memory Rewriting
CHAPTER 2 COMMAND/DATA FRAME FORMAT

The programmer uses the command frame to transmit commands to the RL78. The RL78 uses the data frame to transmit write data or verify data to the programmer. A header, footer, data length information, and checksum are appended to each frame to enhance the reliability of the transferred data.

The following shows the format of a command frame and data frame.

**Figure 2-1. Command Frame Format**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOH</td>
<td>01H</td>
<td>Command frame header</td>
</tr>
<tr>
<td>LEN</td>
<td>(1 byte)</td>
<td>Data length information (00H indicates 256).</td>
</tr>
<tr>
<td>COM</td>
<td>(1 byte)</td>
<td>Command number</td>
</tr>
<tr>
<td>SUM</td>
<td>–</td>
<td>Checksum data for a frame</td>
</tr>
<tr>
<td>ETX</td>
<td>03H</td>
<td>Command frame footer, or footer of last data frame</td>
</tr>
</tbody>
</table>

**Figure 2-2. Data Frame Format**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STX</td>
<td>02H</td>
<td>Data frame header</td>
</tr>
<tr>
<td>LEN</td>
<td>(1 byte)</td>
<td>Data field length</td>
</tr>
<tr>
<td>SUM</td>
<td>(1 byte)</td>
<td>Data frame footer other than the last frame</td>
</tr>
<tr>
<td>ETB</td>
<td>17H</td>
<td>Footer of data frame other than the last frame</td>
</tr>
<tr>
<td>ETX</td>
<td>03H</td>
<td>Command frame footer, or footer of last data frame</td>
</tr>
</tbody>
</table>

The following shows examples of calculating the checksum (SUM) for a frame.
[Command frame]
No command information is included in the following example of a Security Get command frame, so LEN and COM are targets of checksum calculation.

<table>
<thead>
<tr>
<th>SOH</th>
<th>LEN</th>
<th>COM</th>
<th>SUM</th>
<th>ETX</th>
</tr>
</thead>
<tbody>
<tr>
<td>01H</td>
<td>01H</td>
<td>A1H</td>
<td>Checksum</td>
<td>03H</td>
</tr>
</tbody>
</table>

Checksum calculation targets

For this command frame, checksum data is obtained as follows.

00H (initial value) – 01H (LEN) – A1H (COM) = 5EH (Borrow ignored. Lower 8 bits only.)

The Security Get command frame finally transmitted is as follows.

<table>
<thead>
<tr>
<th>SOH</th>
<th>LEN</th>
<th>COM</th>
<th>SUM</th>
<th>ETX</th>
</tr>
</thead>
<tbody>
<tr>
<td>01H</td>
<td>01H</td>
<td>A1H</td>
<td>5EH</td>
<td>03H</td>
</tr>
</tbody>
</table>

[Data frame]
To transmit a data frame as shown below, LEN and D1 to D4 are targets of checksum calculation.

<table>
<thead>
<tr>
<th>STX</th>
<th>LEN</th>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>D4</th>
<th>SUM</th>
<th>ETX</th>
</tr>
</thead>
<tbody>
<tr>
<td>02H</td>
<td>04H</td>
<td>FFH</td>
<td>80H</td>
<td>40H</td>
<td>22H</td>
<td>Checksum</td>
<td>03H</td>
</tr>
</tbody>
</table>

Checksum calculation targets

For this data frame, checksum data is obtained as follows.

00H (initial value) – 04H (LEN) – FFH (D1) – 80H (D2) – 40H (D3) – 22H (D4) = 1BH (Borrow ignored. Lower 8 bits only.)

The data frame finally transmitted is as follows.

<table>
<thead>
<tr>
<th>STX</th>
<th>LEN</th>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>D4</th>
<th>SUM</th>
<th>ETX</th>
</tr>
</thead>
<tbody>
<tr>
<td>02H</td>
<td>04H</td>
<td>FFH</td>
<td>80H</td>
<td>40H</td>
<td>22H</td>
<td>1BH</td>
<td>03H</td>
</tr>
</tbody>
</table>

When a data frame is received, the checksum data is calculated in the same manner, and the obtained value is used to detect a checksum error by judging whether the value is the same as that stored in the SUM field of the receive data. When a data frame as shown below is received, for example, a checksum error is detected.

<table>
<thead>
<tr>
<th>STX</th>
<th>LEN</th>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>D4</th>
<th>SUM</th>
<th>ETX</th>
</tr>
</thead>
<tbody>
<tr>
<td>02H</td>
<td>04H</td>
<td>FFH</td>
<td>80H</td>
<td>40H</td>
<td>22H</td>
<td>1AH</td>
<td>03H</td>
</tr>
</tbody>
</table>

↑ Normally 1BH
2.1 Command Frame Transmission Processing

For details of the flowchart of processing to transmit command frames, read 4.1 Command Frame Transmission Processing Flowchart.

2.2 Data Frame Transmission Processing

The write data frame (user program), verify data frame (user program), and security data frame (security flag) are transmitted as a data frame.

For details of the flowchart of processing to transmit data frames, read 4.2 Data Frame Transmission Processing Flowchart.

2.3 Data Frame Reception Processing

The status frame, silicon signature data frame, security data frame, and checksum data frame are received as a data frame.

For details of the flowchart of processing to receive data frames, read 4.3 Data Frame Reception Processing Flowchart.
CHAPTER 3 DESCRIPTION OF COMMAND PROCESSING

3.1 Reset Command

3.1.1 Description
This command follows the Baud Rate Set command and is used to check if synchronization is performed at the baud rate that has been newly set by the Baud Rate Set command.

3.1.2 Command frame and status frame
Figure 3-1 shows the format of a command frame for the Reset command, and Figure 3-2 shows the status frame for the command.

Figure 3-1. Reset Command Frame (from Programmer to RL78)

<table>
<thead>
<tr>
<th>SOH</th>
<th>LEN</th>
<th>COM</th>
<th>SUM</th>
<th>ETX</th>
</tr>
</thead>
<tbody>
<tr>
<td>01H</td>
<td>01H</td>
<td>00H (Reset)</td>
<td>Checksum</td>
<td>03H</td>
</tr>
</tbody>
</table>

Figure 3-2. Status Frame for Reset Command (from RL78 to Programmer)

<table>
<thead>
<tr>
<th>STX</th>
<th>LEN</th>
<th>Data</th>
<th>SUM</th>
<th>ETX</th>
</tr>
</thead>
<tbody>
<tr>
<td>02H</td>
<td>01H</td>
<td>ST1</td>
<td>Checksum</td>
<td>03H</td>
</tr>
</tbody>
</table>

Remark ST1: Synchronization detection result

See 4.4 Reset Command for details on the flowchart of the processing sequence between the programmer and the RL78, and the flowchart of command processing.
3.2 Baud Rate Set Command

3.2.1 Description
This command is used to set a baud rate (115,200 bps by default) for UART communication and input information on
the data that sets a voltage.
The RL78 determines the operating frequency and programming mode by using voltage setting data and option byte.

3.2.2 Command frame and status frame
Figure 3-3 shows the format of a command frame for the Baud Rate Set command, and Figure 3-4 shows the status
frame for the command.

```
Figure 3-3.  Baud Rate Set Command Frame (from Programmer to RL78)

<table>
<thead>
<tr>
<th>SOH</th>
<th>LEN</th>
<th>COM</th>
<th>Command Information (Hex)</th>
<th>SUM</th>
<th>ETX</th>
</tr>
</thead>
<tbody>
<tr>
<td>01H</td>
<td>03H</td>
<td>9AH</td>
<td>D01 D02</td>
<td></td>
<td>03H</td>
</tr>
</tbody>
</table>

Note For details of the command information setting, refer to Table 3-1. If data other than in Table 3-1 is set, a
time-out error will occur.
If a time-out error has occurred, execute a hardware reset and re-set the flash memory programming mode.

Remark D01: Baud rate setting data
D02: Voltage setting data. Data on the voltage supplied to the target when the flash memory is written is
rounded off at the first place below decimal point and transmitted as hexadecimal data.
Example: Voltage D02
3.69 V → 36 → 24H
2.11 V → 21 → 15H

Table 3-1.  Baud Rate Setting Data Format

<table>
<thead>
<tr>
<th>Data</th>
<th>Set Baud Rate (bps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>00H</td>
<td>115,200</td>
</tr>
<tr>
<td>01H</td>
<td>250,000</td>
</tr>
<tr>
<td>02H</td>
<td>500,000</td>
</tr>
<tr>
<td>03H</td>
<td>1,000,000</td>
</tr>
</tbody>
</table>
Figure 3-4. Status Frame for Baud Rate Set Command (from RL78 to Programmer)

<table>
<thead>
<tr>
<th>STX</th>
<th>LEN</th>
<th>Data</th>
<th>SUM</th>
<th>ETX</th>
</tr>
</thead>
<tbody>
<tr>
<td>02H</td>
<td>03H</td>
<td>ST1</td>
<td>D01</td>
<td>D02</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>checksum</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>03H</td>
<td></td>
</tr>
</tbody>
</table>

**Remark**

ST1: Synchronization detection result

D01: Transmitted as hexadecimal data. Wait time and time-out are set based on this frequency.

Example: 32 MHz: 20H

20 MHz: 18H

D02: Sets a programming mode.

To write in the full-speed mode: 00H

To write in the wide-voltage mode: 01H

See 4.5 **Baud Rate Set Command** for details on the flowchart of the processing sequence between the programmer and the RL78, and the flowchart of command processing.
3.3 Block Erase Command

3.3.1 Description
This command is used to erase the content of flash memory of the block with the specified number.
A block can be specified by specifying the first address of arbitrary block in block units.
Erasing cannot be performed, however, if execution of this command is prohibited due to the security setting (see 3.9 Security Set Command).

3.3.2 Command frame and status frame
Figure 3-7 shows the format of a command frame for the Block Erase command, and Figure 3-8 shows the status frame for the command.

**Figure 3-7. Block Erase Command Frame (from Programmer to RL78)**

<table>
<thead>
<tr>
<th>SOH</th>
<th>LEN</th>
<th>COM</th>
<th>Command Information</th>
<th>SUM</th>
<th>ETX</th>
</tr>
</thead>
<tbody>
<tr>
<td>01H</td>
<td>04H</td>
<td>22H</td>
<td>SAL</td>
<td>SAM</td>
<td>SAH</td>
</tr>
</tbody>
</table>

**Remark**
SAH to SAL: Block erase start address (start address of any block)
SAH: Start address, high (bits 23 to 16)
SAM: Start address, middle (bits 15 to 8)
SAL: Start address, low (bits 7 to 0)

**Figure 3-8. Status Frame for Block Erase Command (from RL78 to Programmer)**

<table>
<thead>
<tr>
<th>STX</th>
<th>LEN</th>
<th>Data</th>
<th>SUM</th>
<th>ETX</th>
</tr>
</thead>
<tbody>
<tr>
<td>02H</td>
<td>01H</td>
<td>ST1</td>
<td>Checksum</td>
<td>03H</td>
</tr>
</tbody>
</table>

**Remark**
ST1: Block erase result

See 4.6 Block Erase Command for details on the flowchart of the processing sequence between the programmer and the RL78, and the flowchart of command processing.
3.4 Programming Command

3.4.1 Description
This command is used to write the user program to the flash memory by transmitting write data after having transmitted the write start address and the write end address. Internal verification is then executed after the last data has been transmitted and writing has been completed.

The write start/end address can be set only in the block start/end address units.

Addresses must not be specified extending from the code flash memory to data flash memory.

If both of the status frames (ST1 and ST2) after the last data transmission indicate ACK, the RL78 firmware automatically executes internal verify. Therefore, the Status command for this internal verify must be transmitted.

3.4.2 Command frame and status frame
Figure 3-9 shows the format of a command frame for the Programming command, and Figure 3-10 shows the status frame for the command.

![Figure 3-9. Programming Command Frame (from Programmer to RL78)](image)

<table>
<thead>
<tr>
<th>SOH</th>
<th>LEN</th>
<th>COM</th>
<th>Command Information</th>
<th>SUM</th>
<th>ETX</th>
</tr>
</thead>
<tbody>
<tr>
<td>01H</td>
<td>07H</td>
<td>04H</td>
<td>Programming</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

SAH to SAL: Write start addresses
EAH to EAL: Write end addresses

![Figure 3-10. Status Frame for Programming Command (from RL78 to Programmer)](image)

<table>
<thead>
<tr>
<th>STX</th>
<th>LEN</th>
<th>Data</th>
<th>SUM</th>
<th>ETX</th>
</tr>
</thead>
<tbody>
<tr>
<td>02H</td>
<td>01H</td>
<td>ST1 (a)</td>
<td>Checksum</td>
<td>03H</td>
</tr>
</tbody>
</table>

ST1 (a): Command reception result

3.4.3 Data frame and status frame
Figure 3-11 shows the format of a frame that includes data to be written, and Figure 3-12 shows the status frame for the data.

![Figure 3-11. Data Frame to Be Written (from Programmer to RL78)](image)

<table>
<thead>
<tr>
<th>STX</th>
<th>LEN</th>
<th>Data</th>
<th>SUM</th>
<th>ETX/ETB</th>
</tr>
</thead>
<tbody>
<tr>
<td>02H</td>
<td>00H</td>
<td>Write Data</td>
<td>Checksum</td>
<td>03H/17H</td>
</tr>
</tbody>
</table>

Write Data: User program to be written

![Figure 3-12. Status Frame for Data Frame (from RL78 to Programmer)](image)

<table>
<thead>
<tr>
<th>STX</th>
<th>LEN</th>
<th>Data</th>
<th>SUM</th>
<th>ETX</th>
</tr>
</thead>
<tbody>
<tr>
<td>02H</td>
<td>02H</td>
<td>ST1 (b)</td>
<td>ST2 (b)</td>
<td>Checksum</td>
</tr>
</tbody>
</table>

ST1 (b): Data reception check result
ST2 (b): Write result
3.4.4 Completion of transferring all data and status frame

Figure 3-13 shows the status frame after transfer of all data is completed.

**Figure 3-13. Status Frame After Completion of Transferring All Data (from RL78 to Programmer)**

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>STX</td>
<td>LEN</td>
<td>Data</td>
<td>SUM</td>
<td>ETX</td>
</tr>
<tr>
<td>02H</td>
<td>01H</td>
<td>ST1 (c)</td>
<td>Checksum</td>
<td>03H</td>
</tr>
</tbody>
</table>

**Remark** ST1 (c): Internal verify result

See 4.7 **Programming Command** for details on the flowchart of the processing sequence between the programmer and the RL78, and the flowchart of command processing.
3.5 Verify Command

3.5.1 Description
This command is used to compare the data transmitted from the programmer with the data read from the RL78 (read level) in the specified address range, and check whether they match.

The verify start/end address can be set only in the block start/end address units.
Addresses must not be specified extending from the code flash memory to data flash memory.

3.5.2 Command frame and status frame
Figure 3-14 shows the format of a command frame for the Verify command, and Figure 3-15 shows the status frame for the command.

![Figure 3-14. Verify Command Frame (from Programmer to RL78)](image)

<table>
<thead>
<tr>
<th>SOH</th>
<th>LEN</th>
<th>COM</th>
<th>Command Information</th>
<th>SUM</th>
<th>ETX</th>
</tr>
</thead>
<tbody>
<tr>
<td>01H</td>
<td>07H</td>
<td>13H</td>
<td>SAL</td>
<td></td>
<td>03H</td>
</tr>
</tbody>
</table>

Remark
SAH to SAL: Verify start addresses
EAH to EAL: Verify end addresses

![Figure 3-15. Status Frame for Verify Command (from RL78 to Programmer)](image)

<table>
<thead>
<tr>
<th>STX</th>
<th>LEN</th>
<th>Data</th>
<th>SUM</th>
<th>ETX</th>
</tr>
</thead>
<tbody>
<tr>
<td>02H</td>
<td>01H</td>
<td>ST1 (a)</td>
<td></td>
<td>03H</td>
</tr>
</tbody>
</table>

Remark
ST1 (a): Command reception result

3.5.3 Data frame and status frame
Figure 3-16 shows the format of a frame that includes data to be verified, and Figure 3-17 shows the status frame for the data.

![Figure 3-16. Data Frame of Data to Be Verified (from Programmer to RL78)](image)

<table>
<thead>
<tr>
<th>STX</th>
<th>LEN</th>
<th>Data</th>
<th>SUM</th>
<th>ETX/ETB</th>
</tr>
</thead>
<tbody>
<tr>
<td>02H</td>
<td>00H</td>
<td>00H (= 256)</td>
<td>Checksum</td>
<td>03H/17H</td>
</tr>
</tbody>
</table>

Remark
Verify Data: User program to be verified
Figure 3-17. Status Frame for Data Frame (from RL78 to Programmer)

<table>
<thead>
<tr>
<th>STX</th>
<th>LEN</th>
<th>Data</th>
<th>SUM</th>
<th>ETX</th>
</tr>
</thead>
<tbody>
<tr>
<td>02H</td>
<td>02H</td>
<td>ST1</td>
<td>ST2</td>
<td>03H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(b)</td>
<td>(b)</td>
<td></td>
</tr>
</tbody>
</table>

Remark  ST1 (b): Data reception check result  
ST2 (b): Verify result

Note  Even if a verify error occurs in the specified address range, ACK is always returned as the verify result. The status of all verify errors are reflected in the verify result for the last data. Therefore, the occurrence of verify errors can be checked only when all the verify processing for the specified address range is completed.

See 4.8 Verify Command for details on the flowchart of the processing sequence between the programmer and the RL78, and the flowchart of command processing.
3.6 Block Blank Check Command

3.6.1 Description
This command is used to check if a block in the flash memory, with a specified block number, is blank (erased state).

A block can be specified with the start address of the blank check start block and the last address of the blank check end block. Successive multiple blocks can be specified. However, blocks must not be specified extending from the code flash memory to data flash memory.

To execute the Block Blank Check command alone, set the blank check area specification field (D01) to “00H” regardless of the specified range. Set D01 to “01H” to execute the Block Blank Check command with all the blocks specified and before the flash memory is erased.

3.6.2 Command frame and status frame
Figure 3-18 shows the format of a command frame for the Block Blank Check command, and Figure 3-19 shows the status frame for the command.

Figure 3-18. Block Blank Check Command Frame (from Programmer to RL78)

<table>
<thead>
<tr>
<th>SOH</th>
<th>LEN</th>
<th>COM</th>
<th>Command Information</th>
<th>SUM</th>
<th>ETX</th>
</tr>
</thead>
<tbody>
<tr>
<td>01H</td>
<td>08H</td>
<td>32H</td>
<td>(Block Blank Check)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Remark | SAH to SAL: Block blank check start address (start address of any block)  
SAM: Start address, middle (bits 15 to 8)  
SAL: Start address, low (bits 7 to 0)  
SAH: Start address, high (bits 23 to 16)  
EAH to EAL: Block blank check end address (last address of any block)  
EAM: End address, middle (bits 15 to 8)  
EAL: End address, low (bits 7 to 0)  
EAH: End address, high (bits 23 to 16)  
D01: Blank check specification area  
00H: Specified block (When performing a block blank check for a single block)  
01H: Specified block and flash option (When performing a blank check for the complete area before erasing the chip) |

Figure 3-19. Status Frame for Block Blank Check Command (from RL78 to Programmer)

<table>
<thead>
<tr>
<th>STX</th>
<th>LEN</th>
<th>Data</th>
<th>SUM</th>
<th>ETX</th>
</tr>
</thead>
<tbody>
<tr>
<td>02H</td>
<td>01H</td>
<td>ST1</td>
<td></td>
<td>03H</td>
</tr>
</tbody>
</table>

Remark | ST1: Block blank check result

See 4.9 Block Blank Check Command for details on the flowchart of the processing sequence between the programmer and the RL78, and the flowchart of command processing.
3.7 Silicon Signature Command

3.7.1 Description
This command is used to read information (silicon signature) of the RL78.

3.7.2 Command frame and status frame
Figure 3-20 shows the format of a command frame for the Silicon Signature command, and Figure 3-21 shows the status frame for the command.

Figure 3-20. Silicon Signature Command Frame (from Programmer to RL78)

<table>
<thead>
<tr>
<th>SOH</th>
<th>LEN</th>
<th>COM</th>
<th>SUM</th>
<th>ETX</th>
</tr>
</thead>
<tbody>
<tr>
<td>01H</td>
<td>01H</td>
<td>C0H</td>
<td>Checksum</td>
<td>03H</td>
</tr>
</tbody>
</table>

(Silicon Signature)

Figure 3-21. Status Frame for Silicon Signature Command (from RL78 to Programmer)

<table>
<thead>
<tr>
<th>STX</th>
<th>LEN</th>
<th>Data</th>
<th>SUM</th>
<th>ETX</th>
</tr>
</thead>
<tbody>
<tr>
<td>02H</td>
<td>01H</td>
<td>ST1</td>
<td>Checksum</td>
<td>03H</td>
</tr>
</tbody>
</table>

Remark ST1: Command reception result

3.7.3 Silicon signature data frame
Figure 3-22 shows the format of a frame that includes silicon signature data.

Figure 3-22. Silicon Signature Data Frame (from RL78 to Programmer)

<table>
<thead>
<tr>
<th>STX</th>
<th>LEN</th>
<th>Data</th>
<th>SUM</th>
<th>ETX</th>
</tr>
</thead>
<tbody>
<tr>
<td>02H</td>
<td>16H</td>
<td>DEC</td>
<td>checksum</td>
<td>03H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DEV</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CEN</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>DEN</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>VER</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(3 bytes) (10 bytes) (3 bytes) (3 bytes) (3 bytes)

Remark DEC: Device code
DEV: Device name
CEN: Last address of code flash ROM
Example) In the case of 00FFFFH: FFH, FFH, 00H
DEN: Last address of data flash ROM
Example) In the case of 0F1FFFH: FFH, 1FH, 0FH
000000H is transmitted with a model not supporting the data flash memory.
VER: Firmware version
Example) If version is V1.23: 01H, 02H, 03H
### Table 3-2. Example of Silicon Signature Data (R5F100LE (RL78/G13))

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Content</th>
<th>Length (Byte)</th>
<th>Example of Silicon Signature Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEC</td>
<td>Device code</td>
<td>3</td>
<td>10H 00H 06H</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>52H = ‘R’</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>35H = ‘5’</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>46H = ‘F’</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>31H = ‘1’</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>30H = ‘0’</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20H = ‘0’</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4CH = ‘L ’</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>45H = ‘E’</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20H = ‘ ’</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20H = ‘ ’</td>
</tr>
<tr>
<td>DEV</td>
<td>Device name</td>
<td>10</td>
<td>FFH FFH 00H</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20H = ‘ ’</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CEN</td>
</tr>
<tr>
<td>CEN</td>
<td>Code flash ROM last address</td>
<td>3</td>
<td>FFH</td>
</tr>
<tr>
<td></td>
<td>(00FFFFh)</td>
<td></td>
<td>00H</td>
</tr>
<tr>
<td>DEN</td>
<td>Data flash ROM last address</td>
<td>3</td>
<td>FFH</td>
</tr>
<tr>
<td></td>
<td>(001FFFh)</td>
<td></td>
<td>1FH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>00H</td>
</tr>
<tr>
<td>VER</td>
<td>Firmware version</td>
<td>3</td>
<td>01H</td>
</tr>
<tr>
<td></td>
<td>(V1.23)</td>
<td></td>
<td>02H</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>03H</td>
</tr>
</tbody>
</table>

See 4.10 Silicon Signature Command for details on the flowchart of the processing sequence between the programmer and the RL78, and the flowchart of command processing.
3.8 Checksum Command

3.8.1 Description
This command is used to acquire the checksum data in the specified area.
For the checksum calculation start/end address, specify a fixed address in block units (1 KB) starting from the top of the flash memory.
Addresses must not be specified extending from the code flash memory to data flash memory.
Checksum data is obtained by sequentially subtracting data in the specified address range from the initial value (0000H) in 1-byte units.

3.8.2 Command frame and status frame
Figure 3-26 shows the format of a command frame for the Checksum command, and Figure 3-27 shows the status frame for the command.

Figure 3-26. Checksum Command Frame (from Programmer to RL78)

<table>
<thead>
<tr>
<th>SOH</th>
<th>LEN</th>
<th>COM</th>
<th>Command Information</th>
<th>SUM</th>
<th>ETX</th>
</tr>
</thead>
<tbody>
<tr>
<td>01H</td>
<td>07H</td>
<td>B0H</td>
<td>SAL, SAM, SAH, EAL, EAM, EAH</td>
<td>Checksum</td>
<td>03H</td>
</tr>
</tbody>
</table>

Remark: SAH to SAL: Checksum calculation start addresses
EAH to EAL: Checksum calculation end addresses

Figure 3-27. Status Frame for Checksum Command (from RL78 to Programmer)

<table>
<thead>
<tr>
<th>STX</th>
<th>LEN</th>
<th>Data</th>
<th>SUM</th>
<th>ETX</th>
</tr>
</thead>
<tbody>
<tr>
<td>02H</td>
<td>01H</td>
<td>ST1</td>
<td>Checksum</td>
<td>03H</td>
</tr>
</tbody>
</table>

Remark: ST1: Command reception result

3.8.3 Checksum data frame
Figure 3-28 shows the format of a frame that includes checksum data.

Figure 3-28. Checksum Data Frame (from RL78 to Programmer)

<table>
<thead>
<tr>
<th>STX</th>
<th>LEN</th>
<th>Data</th>
<th>SUM</th>
<th>ETX</th>
</tr>
</thead>
<tbody>
<tr>
<td>02H</td>
<td>02H</td>
<td>CK1, CK2</td>
<td>Checksum</td>
<td>03H</td>
</tr>
</tbody>
</table>

Remark: CK1: Lower 8 bits of checksum data
CK2: Higher 8 bits of checksum data

See 4.11 Checksum Command for details on the flowchart of the processing sequence between the programmer and the RL78, and the flowchart of command processing.
3.9 Security Set Command

3.9.1 Description

This command is used to perform security settings (enabling/disabling of write, block erase, and boot block cluster rewriting, and setting of flash shield window and others). By performing these settings with this command, rewriting of the flash memory by an unauthorized party can be restricted. The security settings performed by this command are also valid for the data flash memory.

Caution Even after the security setting, additional setting of changing from enable to disable can be performed; however, changing from disable to enable is not possible. If an attempt is made to perform such a setting, a protect error (10H) will occur. If such setting is required, the Security Release command must first be executed.

If block erase or boot block cluster rewriting has been disabled, however, the Security Release command cannot be executed. Re-confirmation of security setting execution is therefore recommended before disabling block erase or boot block cluster rewriting, due to this programmer specification.

3.9.2 Command frame and status frame

Figure 3-29 shows the format of a command frame for the Security Set command, and Figure 3-30 shows the status frame for the command.

**Figure 3-29. Security Set Command Frame (from Programmer to RL78)**

<table>
<thead>
<tr>
<th>SOH</th>
<th>LEN</th>
<th>COM</th>
<th>SUM</th>
<th>ETX</th>
</tr>
</thead>
<tbody>
<tr>
<td>01H</td>
<td>01H</td>
<td>A0H</td>
<td>Checksum</td>
<td>03H</td>
</tr>
</tbody>
</table>

(Security Set)

**Figure 3-30. Status Frame for Security Set Command (from RL78 to Programmer)**

<table>
<thead>
<tr>
<th>STX</th>
<th>LEN</th>
<th>Data</th>
<th>SUM</th>
<th>ETX</th>
</tr>
</thead>
<tbody>
<tr>
<td>02H</td>
<td>01H</td>
<td>ST1 (a)</td>
<td>Checksum</td>
<td>03H</td>
</tr>
</tbody>
</table>

**Remark** ST1 (a): Command reception result
3.9.3 Data frame and status frame

Figure 3-31 shows the format of a security data frame, and Figure 3-32 shows the status frame for the data.

Figure 3-31. Security Data Frame (from Programmer to RL78)

<table>
<thead>
<tr>
<th>STX</th>
<th>LEN</th>
<th>Data</th>
<th>SUM</th>
<th>ETX</th>
</tr>
</thead>
<tbody>
<tr>
<td>02H</td>
<td>08H</td>
<td>FLG</td>
<td>BOT</td>
<td>SSL</td>
</tr>
</tbody>
</table>

Remarks
1. FLG: Security flag
   - BOT: Boot block cluster block number
   - SSL: Flash shield window start block number (Lower)
   - SSH: Flash shield window start block number (Higher)
   - SEL: Flash shield window end block number (Lower)
   - SEH: Flash shield window end block number (Higher)
   - RES: Invalid data
2. If the flash shield window is not to be set, set SSL/SSH to 0000H and SEL/SEH to the target device end block number.

Figure 3-32. Status Frame for Security Data Writing (from RL78 to Programmer)

<table>
<thead>
<tr>
<th>STX</th>
<th>LEN</th>
<th>Data</th>
<th>SUM</th>
<th>ETX</th>
</tr>
</thead>
<tbody>
<tr>
<td>02H</td>
<td>01H</td>
<td>ST1 (b)</td>
<td>Checksum</td>
<td>03H</td>
</tr>
</tbody>
</table>

Remark
ST1 (b): Security data write result

The following table shows the contents in the security flag field.

<table>
<thead>
<tr>
<th>Item</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit 7</td>
<td>Fixed to “1”</td>
</tr>
<tr>
<td>Bit 6</td>
<td>Fixed to “1”</td>
</tr>
<tr>
<td>Bit 5</td>
<td>Fixed to “1”</td>
</tr>
<tr>
<td>Bit 4</td>
<td>Programming disable flag (1: Enables programming, 0: Disable programming)</td>
</tr>
<tr>
<td>Bit 3</td>
<td>Fixed to “1”</td>
</tr>
<tr>
<td>Bit 2</td>
<td>Block erase disable flag (1: Enables block erase, 0: Disable block erase)</td>
</tr>
<tr>
<td>Bit 1</td>
<td>Boot block cluster rewrite disable flag (1: Enables boot block cluster rewrite, 0: Disable boot block cluster rewrite)</td>
</tr>
<tr>
<td>Bit 0</td>
<td>Fixed to “1”</td>
</tr>
</tbody>
</table>
The following table shows the relationship between the security flag field settings and the enable/disable status of each operation.

Table 3-4. Security Flag Field and Enable/Disable Status of Each Operation

<table>
<thead>
<tr>
<th>Command</th>
<th>Block Erase</th>
<th>Programming</th>
<th>Security Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target Area</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security Setting Item</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disable programming</td>
<td>√</td>
<td>√</td>
<td>×</td>
</tr>
<tr>
<td>Disable block erase</td>
<td>×</td>
<td>×</td>
<td>√</td>
</tr>
<tr>
<td>Boot block cluster rewrite disable flag</td>
<td>△</td>
<td>√</td>
<td>△</td>
</tr>
</tbody>
</table>

For the relationship between the security function and command, and for security in the self-programming mode, refer to the User's Manual of each product.

See 4.12 Security Set Command for details on the flowchart of the processing sequence between the programmer and the RL78, and the flowchart of command processing.
3.10 Security Get Command

3.10.1 Description
This command is used to acquire security information set to the RL78 (such as writing, block erasure, enabling/disabling rewriting of boot block cluster, and setting of flash shield window).

3.10.2 Command frame and status frame
Figure 3-33 shows the format of a command frame for the Security Get command, and Figure 3-34 shows the status frame for the command.

**Remark** ST1: Command reception result
3.10.3 Data frame and security flag

Figure 3-35 shows the format of a security data frame.

**Figure 3-35.  Security Data Frame (from RL78 to Programmer)**

<table>
<thead>
<tr>
<th>STX</th>
<th>LEN</th>
<th>Data</th>
<th>SUM</th>
<th>ETX</th>
</tr>
</thead>
<tbody>
<tr>
<td>02H</td>
<td>08H</td>
<td>FLG</td>
<td>BOT</td>
<td>SSL</td>
</tr>
</tbody>
</table>

**Remark**
- FLG: Security flag
- BOT: Boot block cluster block number
- SSL: Flash shield window start block number (Lower)
- SSH: Flash shield window start block number (Higher)
- SEL: Flash shield window end block number (Lower)
- SEH: Flash shield window end block number (Higher)
- RES: Invalid data

The following table shows the contents in the security flag field.

**Table 3-5. Contents of Security Flag Field**

<table>
<thead>
<tr>
<th>Item</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit 7</td>
<td>Fixed to “1”</td>
</tr>
<tr>
<td>Bit 6</td>
<td>Fixed to “1”</td>
</tr>
<tr>
<td>Bit 5</td>
<td>Fixed to “1”</td>
</tr>
<tr>
<td>Bit 4</td>
<td>Programming disable flag (1: Enables programming, 0: Disable programming)</td>
</tr>
<tr>
<td>Bit 3</td>
<td>Fixed to “1”</td>
</tr>
<tr>
<td>Bit 2</td>
<td>Block erase disable flag (1: Enables block erase, 0: Disable block erase)</td>
</tr>
<tr>
<td>Bit 1</td>
<td>Boot block cluster rewrite disable flag (1: Enables boot block cluster rewrite, 0: Disable boot block cluster rewrite)</td>
</tr>
<tr>
<td>Bit 0</td>
<td>Boot area exchange flag (“1”: Provided, “0”: None)</td>
</tr>
</tbody>
</table>

See 4.13 Security Get Command for details on the flowchart of the processing sequence between the programmer and the RL78, and the flowchart of command processing.
3.11 Security Release Command

3.11.1 Description
This command is used to initialize the security information set to the RL78 (such as writing, block erasure, enabling/disabling rewriting of boot block cluster, and setting of flash shield window).

The Security Release command can be executed only when all the following conditions are satisfied.
- “Block erase” and “Boot block cluster rewrite” are not prohibited.
  - If these are prohibited, a Protect error occurs.
- The code flash memory and data flash memory are blank.
  - If they are not blank, a Blank error occurs.

Note Only with a model with data flash memory

3.11.2 Command frame and status frame
Figure 3-36 shows the format of a command frame for the Security Release command, and Figure 3-37 shows the status frame for the command.

| Figure 3-36.  Security Release Command Frame (from Programmer to RL78) |
|---------------------------|--------------|------------------|-----------------|
| SOH          | LEN         | COM              | SUM             | ETX            |
| 01H          | 01H         | A2H (Security Release) | Checksum        | 03H            |

| Figure 3-37.  Status Frame for Security Release Command (from RL78 to Programmer) |
|---------------------------|--------------|------------------|-----------------|
| STX          | LEN         | Data             | SUM             | ETX            |
| 02H          | 01H         | ST1              | Checksum        | 03H            |

Remark ST1: Command reception result

See 4.14 Security Release Command for details on the flowchart of the processing sequence between the programmer and the RL78, and the flowchart of command processing.
4.1 Command Frame Transmission Processing Flowchart

Command frame transmission processing

Command frame header (SOH = 01H) transmission

Wait between data transmissions

Data length (LEN) transmission

Wait between data transmissions

Command number (COM) transmission

(LEN - 1) bytes transmitted?

Yes

No

Wait between data transmissions

Transmits 1-byte command information

Wait between data transmissions

Checksum data (SUM) transmission

Wait between data transmissions

Command frame footer (ETX = 03H) transmission

End of command frame transmission
4.2 Data Frame Transmission Processing Flowchart

Data frame transmission processing

Data frame header (SOH = 03H) transmission

Wait between data transmissions

Data length (LEN) transmission

Wait between data transmissions

Command number (COM) transmission

LEN bytes transmitted?

Yes

No

Wait between data transmissions

Transmits 1-byte data

Wait between data transmissions

Checksum data (SUM) transmission

Wait between data transmissions

Last data frame?

Yes

Transmission of last data frame footer (ETX = 03H)

End of data frame transmission

No

Transmission of footer other than those of last data frame (ETB = 17H)
4.3 Data Frame Reception Processing Flowchart

- Data frame reception processing
  - Data frame header (STX = 02H) received?
    - Yes
      - Data length (LEN) received?
        - Yes
          - 1-byte data received?
            - Yes
              - LEN bytes received?
                - Yes
                  - Checksum data (SUM) received?
                    - Yes
                      - Data frame footer received?
                        - Yes
                          - Checksum error?
                            - Yes
                              - End of data frame reception
                            - No
                              - Checksum error
                        - No
                          - End of data frame reception
                        - Yes
                          - Checksum error
                    - No
                      - Reception time-out error
                - No
                  - Reception time-out error
            - No
              - Reception time-out error
        - No
          - Timed out?
            - Yes
              - Reception time-out error
            - No
              - Yes
        - No
          - Reception time-out error
  - No
    - Timed out?
      - Yes
        - Reception time-out error
      - No
        - Yes
4.4 Reset Command

4.4.1 Processing sequence chart

Reset command processing sequence
4.4.2 Description of processing sequence

<1> Waits from the previous frame reception until the next command processing starts (wait time \( t_{SN6} \)).

<2> The Reset command is transmitted by command frame transmission processing.

<3> A time-out check is performed from command transmission until status frame reception.
   If a time-out occurs, a time-out error [C] is returned (time-out time \( t_{CS1} \)).

<4> The status code is checked.

When ST1 = ACK:  Normal completion [A]
When ST1 \( \neq \) ACK:  Abnormal termination [B]

4.4.3 Status at processing completion

<table>
<thead>
<tr>
<th>Status at Processing Completion</th>
<th>Status Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal completion [A]</td>
<td>Normal acknowledgment (ACK)</td>
<td>06H</td>
</tr>
<tr>
<td>Abnormal termination [B]</td>
<td>Checksum error</td>
<td>07H</td>
</tr>
<tr>
<td></td>
<td>Negative acknowledgment (NACK)</td>
<td>15H</td>
</tr>
<tr>
<td>Time-out error [C]</td>
<td>–</td>
<td>The status frame was not received within the specified time.</td>
</tr>
</tbody>
</table>
4.4.4 Flowchart

Reset command processing

Wait from previous frame reception until next command transmission

Command frame transmission processing (Reset)

Status frame received?

Yes

Timed out?

Yes

Time-out error [C]

No

Status = ACK?

No

Abnormal termination [B]

Yes

Normal completion [A]
4.5 Baud Rate Set Command

4.5.1 Processing sequence chart

Baud Rate Set command processing sequence

<1> Waits from the mode setting until the next command transmission
<2> Baud Rate Set command frame transmission
<3> Time-out check for status frame reception
<4> Status frame reception
<5> Re-calculates timing parameters based on the operating frequency and programming mode of the target which have been received in <4>
<6> The baud rate of UART is switched to the value set by the Baud Rate Set command.
<7> Wait from status frame transmission until Reset command transmission
<8> Reset command frame transmission
<9> Time-out check for status frame reception
<10> Status frame reception

Time-out occurs
Status frame receive within specified time
Time-out error [C]

Reception status
[ACK/other than ACK]

Abnormal termination [B]

ACK
Normal completion [A]

Other than ACK

Reception status
[ACK/other than ACK]
4.5.2 Description of processing sequence

<1> Waits from the mode setting until the next command transmission (wait time $t_{MB}$).

<2> The Baud Rate Set command is transmitted by command frame transmission processing.

<3> A time-out check is performed from command transmission until status frame reception.
   If a time-out occurs, a time-out error [C] is returned (time-out time $t_{CS6}$).

<4> The status code is checked.

When ST1 = ACK: Proceeds to <5>.
When ST1 ≠ ACK: Abnormal termination [B]

<5> Re-calculates timing parameters based on the operating frequency and programming mode of the target which have been received.

<6> Switches the baud rate for UART communication to the value set by the Baud Rate Set command.

<7> Waits from the command transmission until the Reset command transmission (wait time $t_{SN6}$).

<8> The Reset command is transmitted by command frame transmission processing.

<9> A time-out check is performed from command transmission until status frame reception.
   If a time-out occurs, a time-out error [C] is returned (time-out time $t_{CS1}$).

<10> The status code is checked.

When ST1 = ACK: Normal completion [A]
When ST1 ≠ ACK: Abnormal termination [B]

4.5.3 Status at processing completion

<table>
<thead>
<tr>
<th>Status at Processing Completion</th>
<th>Status Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal completion [A]</td>
<td>Normal acknowledgment (ACK)</td>
<td>06H The command was executed normally and the synchronization of the UART communication speed has been established between the programmer and the RL78.</td>
</tr>
<tr>
<td>Abnormal termination [B]</td>
<td>Checksum error</td>
<td>07H The checksum of the transmitted command frame does not match.</td>
</tr>
<tr>
<td></td>
<td>Command number error</td>
<td>04H Command other than Baud Rate Set command has been received.</td>
</tr>
<tr>
<td></td>
<td>Parameter error</td>
<td>05H Command information (D01) is illegal. Or, command information (D02) indicates less than 1.8 V.</td>
</tr>
<tr>
<td></td>
<td>Negative acknowledgment (NACK)</td>
<td>15H Command frame data is abnormal (such as invalid data length (LEN) or no ETX).</td>
</tr>
<tr>
<td>Time-out error [C]$^{ext}$</td>
<td>–</td>
<td>Data frame reception was timed out.</td>
</tr>
</tbody>
</table>

**Note** If the Baud Rate Set command has not been completed normally, execute a hardware reset and re-set to the flash memory programming mode.
4.5.4 Flowchart

Baud Rate Set command processing

Wait from mode setting until next command transmission

Command frame transmission processing (Baud Rate Set)

Status frame received?

Yes

Reception status = ACK?

Yes

Re-calculates timing parameters based on the operating frequency and programming mode of the target which have been received.

No

Abnormal termination

No

Timed out?

Yes

Timeout error [C]

No

Wait from mode setting until next command transmission

Command frame transmission processing (fixed)

Status frame received?

Yes

Timed out?

No

Reception status = ACK?

Yes

Normal completion [A]

No

Abnormal termination [B]

Time-out error [C]
4.6 Block Erase Command

4.6.1 Processing sequence chart

Block Erase command processing sequence

- **<1>** Wait from previous frame reception until next command transmission
- **<2>** Block Erase command frame transmission
- **<3>** Time-out check for status frame reception
- **<4>** Status frame reception

- **Time-out error [C]**
- **Other than ACK**
- **Abnormal termination [B]**
- **ACK**
  - Yes
    - Specified block completely erased?
    - No
      - To <1>
    - Yes
      - Normal completion [A]
  - To <4>
4.6.2 Description of processing sequence

<1> Waits from the previous frame reception until the next command transmission (wait time tSNx/tDNx).
<2> The Block Erase command is transmitted by command frame transmission processing.
<3> A time-out check is performed from command transmission until status frame reception.
   If a time-out occurs, a time-out error [C] is returned (time-out time tcs3).
<4> The status code is checked.

When ST1 = ACK: Normal completion [A] if the specified block has been erased.
   Returns to <1> if the specified block has not been erased.
When ST1 ≠ ACK: Abnormal termination [B]

4.6.3 Status at processing completion

<table>
<thead>
<tr>
<th>Status at Processing Completion</th>
<th>Status Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal completion</td>
<td>06H</td>
<td>The command was executed normally and block erase was performed normally.</td>
</tr>
<tr>
<td>Normal acknowledgment (ACK)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Abnormal termination</td>
<td>05H</td>
<td>The specified start address is not the first address of the block.</td>
</tr>
<tr>
<td>Parameter error</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Checksum error</td>
<td>07H</td>
<td>The checksum of the transmitted command frame does not match.</td>
</tr>
<tr>
<td>Protect error</td>
<td>10H</td>
<td>Block erase is prohibited in the security setting. A boot block is included in the specified range and boot block rewrite is prohibited.</td>
</tr>
<tr>
<td>Negative acknowledgment (NACK)</td>
<td>15H</td>
<td>Command frame data is abnormal (such as invalid data length (LEN) or no ETX).</td>
</tr>
<tr>
<td>Erase error</td>
<td>1AH</td>
<td>An erase error has occurred.</td>
</tr>
<tr>
<td>Time-out error [C]</td>
<td>–</td>
<td>The status frame was not received within the specified time.</td>
</tr>
</tbody>
</table>
4.6.4 Flowchart

Block Erase command processing

Wait from previous frame reception until next command transmission

Command frame transmission processing (Block Erase)

Status frame received?

Yes

No

Timed out?

Yes

No

Status = ACK?

Yes

No

Specified block completely erased?

Yes

No

Normal completion[A]

Time-out error [C]

Abnormal termination [B]
4.7 Programming Command

4.7.1 Processing sequence chart

Programming command processing sequence:

1. Wait from previous frame reception until next command transmission
2. Programming command frame transmission
3. Time-out check for status frame reception
4. Status frame reception
5. Status frame received within specified time
6. Data frame (user data) transmission
7. Time-out check for status frame reception
8. Status frame reception
9. Status frame received within specified time
10. Time-out check for status frame reception

Abnormal termination:
- [B] Other than ACK
- [E] Other than ACK
- [D] All data frames transmitted? [Yes/No]
- [C] Time-out occurs
- [A] Normal completion

Time-out error:
- [C] Time-out occurs
- [C] Time-out occurs
- [C] Time-out occurs

Reception status:
- [ACK/other than ACK]
- [ACK/other than ACK]
- [ACK/other than ACK]
4.7.2 Description of processing sequence

<1> Waits from the previous frame reception until the next command transmission (wait time tSNx/tDNx).

<2> The Programming command is transmitted by command frame transmission processing.

<3> A time-out check is performed from command transmission until status frame reception.
   If a time-out occurs, a time-out error [C] is returned (time-out time tCSS).

<4> The status code is checked.

   When ST1 = ACK: Proceeds to <5>.
   When ST1 ≠ ACK: Abnormal termination [B]

<5> Waits from the previous frame reception until the next data frame transmission (wait time tSDx).

<6> User data is transmitted by data frame transmission processing.

<7> A time-out check is performed from user data transmission until data frame reception.
   If a time-out occurs, a time-out error [C] is returned (time-out time tDS).

<8> The status code (ST1/ST2) is checked (also refer to the processing sequence chart and flowchart).

   When ST1 ≠ ACK: Abnormal termination [B]
   When ST1 = ACK: The following processing is performed according to the ST2 value.
      • When ST2 = ACK: Proceeds to <9> when transmission of all data frames is completed.
        If there still remain data frames to be transmitted, the processing re-executes the
        sequence from <5>.
      • When ST2 ≠ ACK: Abnormal termination [D]

<9> A time-out check is performed until status frame reception.
   If a time-out occurs, a time-out error [C] is returned (time-out time tSS).

<10> The status code is checked.

   When ST1 = ACK: Normal completion [A]
   When ST1 ≠ ACK: Abnormal termination [E]
### 4.7.3 Status at processing completion

<table>
<thead>
<tr>
<th>Status at Processing Completion</th>
<th>Status Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal completion [A]</td>
<td>06H</td>
<td>The command was executed normally and the user data was written normally.</td>
</tr>
<tr>
<td>Normal acknowledgment (ACK)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Abnormal termination [B]</td>
<td>05H</td>
<td>The start/end address is out of the flash memory range, the specified start/end address is not the first/end address of the block, or the write start address is larger than the end address. Or, the address range specified by the start/end address extends from the code flash memory to the data flash memory.</td>
</tr>
<tr>
<td>Parameter error</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Checksum error</td>
<td>07H</td>
<td>The checksum of the transmitted command frame or data frame does not match.</td>
</tr>
<tr>
<td>Protect error</td>
<td>10H</td>
<td>Write is prohibited in the security setting. A boot block is included in the specified range and boot block rewrite is prohibited.</td>
</tr>
<tr>
<td>Negative acknowledgment (NACK)</td>
<td>15H</td>
<td>Command frame data or data frame data is abnormal (such as invalid data length (LEN) or no ETX).</td>
</tr>
<tr>
<td>Abnormal termination [C], [D], [E]</td>
<td>1BH, 1CH</td>
<td>The status frame was not received within the specified time. A write error has occurred.</td>
</tr>
</tbody>
</table>

- **A**: Normal completion
- **B**: Abnormal termination
- **C**: Time-out error
- **D**: IVerify error
- **E**: Write error
4.7.4 Flowchart

Programming command processing

Use for previous frame received until next command transmission?

Command frame transmission processing (Programming)

Status frame received?

Yes

Status = ACK?

No

Timed out?

Yes

Time-out error [C]

No

Abnormal termination [B]

Data frame transmission processing (User program)

Status frame received?

Yes

ST1 = ACK?

No

All data frames transmitted?

No

Status frame received?

Yes

Status = ACK?

No

Timed out?

Yes

Time-out error [C]

No

Abnormal termination [B]

ST2 = ACK?

Yes

Abnormal termination [D]

No

ST1 = ACK?

Yes

Abnormal termination [B]

No

All data frames transmitted?

Yes

Status frame received?

Yes

Status = ACK?

No

Timed out?

Yes

Time-out error [C]

No

Abnormal termination [B]

Yes

Normal completion [A]

No

Abnormal termination [E]
4.8 Verify Command

4.8.1 Processing sequence char

Verify command processing sequence

Programmer

RL78

1. Wait for previous frame reception until next command transmission

2. Verify command frame transmission

3. Time-out check for status frame reception

4. Status frame reception

5. Wait for previous frame reception until next data frame transmission

6. Data frame (user data for verify) transmission

7. Time-out check for status frame reception

8. Status frame reception (ST1/ST2)

Reception status

[ACK/other than ACK]

Abnormal termination [B]

Other than ACK

Abnormal termination [B]

All data frames transmitted? [Yes/No]

No

To <5>

Yes

Normal completion [A]

Time-out error [C]

Status frame received within specified time

Other than ACK

Abnormal termination [B]

Time-out error [C]

Status frame received within specified time

Other than ACK

Abnormal termination [B]

Status frame received within specified time

Other than ACK

Abnormal termination [D]

Time-out error [C]

Status frame received within specified time

Other than ACK

Abnormal termination [B]

All data frames transmitted? [Yes/No]

Yes

Normal completion [A]

Time-out error [C]

Status frame received within specified time

Other than ACK

Abnormal termination [B]

Status frame received within specified time

Other than ACK

Abnormal termination [B]

Status frame received within specified time

Other than ACK

Abnormal termination [D]

Time-out error [C]

Status frame received within specified time

Other than ACK

Abnormal termination [B]

All data frames transmitted? [Yes/No]

Yes

Normal completion [A]

Time-out error [C]

Status frame received within specified time

Other than ACK

Abnormal termination [B]

Status frame received within specified time

Other than ACK

Abnormal termination [B]

Status frame received within specified time

Other than ACK

Abnormal termination [D]

Time-out error [C]

Status frame received within specified time

Other than ACK

Abnormal termination [B]

All data frames transmitted? [Yes/No]

Yes

Normal completion [A]
4.8.2 Description of processing sequence

<1> Waits from the previous frame reception until the next command transmission (wait time tSNx/tDNx).

<2> The Verify command is transmitted by command frame transmission processing.

<3> A time-out check is performed from command transmission until status frame reception.
   If a time-out occurs, a time-out error [C] is returned (time-out time tCS2).

<4> The status code is checked.

   When ST1 = ACK: Proceeds to <5>.
   When ST1 ≠ ACK: Abnormal termination [B]

<5> Waits from the previous frame reception until the next data frame transmission (wait time tSD2).

<6> User data for verifying is transmitted by data frame transmission processing.

<7> A time-out check is performed from user data transmission until status frame reception.
   If a time-out occurs, a time-out error [C] is returned (time-out time tDS2).

<8> The status code (ST1/ST2) is checked (also refer to the processing sequence chart and flowchart).

   When ST1 ≠ ACK: Abnormal termination [B]
   When ST1 = ACK: The following processing is performed according to the ST2 value.
   • When ST2 = ACK: If transmission of all data frames is completed, the processing ends normally [A].
     If there still remain data frames to be transmitted, the processing re-executes the sequence from <5>.
   • When ST2 ≠ ACK: Abnormal termination [D]

4.8.3 Status at processing completion

<table>
<thead>
<tr>
<th>Status at Processing Completion</th>
<th>Status Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal completion [A]</td>
<td>Normal acknowledgment (ACK)</td>
<td>06H</td>
</tr>
<tr>
<td>Abnormal termination [B]</td>
<td>Parameter error</td>
<td>05H</td>
</tr>
<tr>
<td></td>
<td>Checksum error</td>
<td>07H</td>
</tr>
<tr>
<td></td>
<td>Negative acknowledgment (NACK)</td>
<td>15H</td>
</tr>
<tr>
<td>Time-out error [C]</td>
<td>–</td>
<td>The status frame was not received within the specified time.</td>
</tr>
<tr>
<td>Abnormal termination [D]</td>
<td>Verify error</td>
<td>0FH (ST2)</td>
</tr>
</tbody>
</table>
4.8.4 Flowchart

1. Verify command processing

   wait from previous frame reception until next command transmission

2. Command frame transmission processing (Verify)

3. Status frame received?
   - Yes
   - No
     - Timed out?
       - Yes
         - tds2
         - Time-out error [C]
       - No
         - ST1 = ACK?
           - Yes
             - tds2
             - ST2 = ACK?
               - Yes
                 - Abnormal termination [B]
               - No
                 - Abnormal termination [D]
           - No
             - Abnormal termination [B]
   - No
     - ST1 = ACK?
       - Yes
         - All data frames transmitted?
           - Yes
             - Normal completion [A]
           - No
             - Abnormal termination [B]
       - No
         - Abnormal termination [B]
4.9 Block Blank Check Command

4.9.1 Processing sequence chart

Block Blank Check command processing sequence

1. Wait from previous frame reception until next command transmission
2. Block Blank Check command frame transmission
3. Time-out check for status frame reception
4. Status frame reception

Time-out error [C]
Other than ACK
Abnormal termination [B]
ACK
Normal completion [A]
4.9.2 Description of processing sequence

<1> Waits from the previous frame reception until the next command transmission (wait time tsNxt/tDNx).
<2> The Block Blank Check command is transmitted by command frame transmission processing.
<3> A time-out check is performed from command transmission until status frame reception.
   If a time-out occurs, a time-out error [C] is returned (time-out time tCS4).
<4> The status code is checked.

When ST1 = ACK: Normal completion [A]
When ST1 ≠ ACK: Abnormal termination [B]

4.9.3 Status at processing completion

<table>
<thead>
<tr>
<th>Status at Processing Completion</th>
<th>Status Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal completion [A]</td>
<td>Normal acknowledgment (ACK)</td>
<td>06H</td>
</tr>
<tr>
<td>Abnormal termination [B]</td>
<td>Parameter error</td>
<td>05H</td>
</tr>
<tr>
<td></td>
<td>Checksum error</td>
<td>07H</td>
</tr>
<tr>
<td></td>
<td>Negative acknowledgment (NACK)</td>
<td>15H</td>
</tr>
<tr>
<td></td>
<td>Blank error</td>
<td>1BH</td>
</tr>
<tr>
<td>Time-out error [C]</td>
<td>–</td>
<td>The status frame was not received within the specified time.</td>
</tr>
</tbody>
</table>
4.9.4 Flowchart

Block Blank Check command processing

Wait from previous frame reception until next command transmission

Command frame transmission processing (Block Blank Check)

Status frame received?

Yes

Timed out?

No

Yes

Time-out error [C]

No

Status = ACK?

Yes

Normal completion [A]

No

Abnormal termination [B]
4.10 Silicon Signature Command

4.10.1 Processing sequence chart

Silicon Signature command processing sequence

Programmer

1. Wait from previous frame reception until next command transmission

2. Silicon Signature command frame transmission

3. Time-out check for status frame reception

4. Status frame reception

5. Time-out check for data frame reception

6. Data frame (Silicon Signature) reception

RL78

Silicon Signature command frame transmission

1. Normal data frame? [Yes/No]

2. Yes

3. Normal completion [A]

4. No

5. Data frame error [D]

6. Time-out error [C]

7. Abnormal termination [B]

8. Other than ACK

9. ACK

Time-out occurs

Time-out error [C]

Status frame received within specified time

Reception status [ACK/other than ACK]

Data frame received within specified time

Data frame (Silicon Signature) reception

Time-out occurs
4.10.2 Description of processing sequence

<1> Waits from the previous frame reception until the next command transmission (wait time \( t_{SNx}/t_{DNx} \)).

<2> The Silicon Signature command is transmitted by command frame transmission processing.

<3> A time-out check is performed from command transmission until status frame reception.
   If a time-out occurs, a time-out error [C] is returned (time-out time \( t_{CS11} \)).

<4> The status code is checked.

   When \( ST1 = ACK \): Proceeds to <5>.
   When \( ST1 \neq ACK \): Abnormal termination [B]

<5> A time-out check is performed until data frame (silicon signature data) reception.
   If a time-out occurs, a time-out error [C] is returned (time-out time \( t_{SD11} \)).

<6> The received data frame (silicon signature data) is checked.

   If data frame is normal: Normal completion [A]
   If data frame is abnormal: Data frame error [D]

4.10.3 Status at processing completion

<table>
<thead>
<tr>
<th>Status at Processing Completion</th>
<th>Status Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal completion [A]</td>
<td>–</td>
<td>The command was executed normally and silicon signature was acquired normally.</td>
</tr>
<tr>
<td>Abnormal termination [B]</td>
<td>Checksum error</td>
<td>07H</td>
</tr>
<tr>
<td></td>
<td>Negative acknowledgment (NACK)</td>
<td>15H</td>
</tr>
<tr>
<td>Time-out error [C]</td>
<td>–</td>
<td>The status frame or data frame was not received within the specified time.</td>
</tr>
<tr>
<td>Data frame error [D]</td>
<td>–</td>
<td>The checksum of the data frame received as silicon signature data does not match.</td>
</tr>
</tbody>
</table>
4.10.4 Flowchart

Silicon Signature command processing

Wait from previous frame reception until next command transmission

Command frame transmission processing (Silicon Signature)

Status frame received?

Yes

Timed out?

No

Status = ACK?

Yes

Abnormal termination [B]

No

Data frame (silicon signature) received?

Yes

Timed out?

No

Normal data frame?

Yes

Normal completion [A]

No

Data frame error [D]

No

Time-out error [C]
4.11 Checksum Command

4.11.1 Processing sequence chart

Checksum command processing sequence

Programmer

<1> Wait from previous frame reception until next command transmission

<2> Checksum command frame transmission

<3> Time-out check for status frame reception

<4> Status frame reception

<5> Time-out check for data frame reception

<6> Data frame (checksum data) reception

RL78

Normal completion [A]

Normal data frame? [Yes/No]

Yes

Data frame error [D]

No

Time-out error [C]

Abnormal termination [B]

Reception status [ACK/other than ACK]

ACK

Other than ACK

Time-out occurs

Status frame received within specified time

Normal data frame? [Yes/No]

Yes

Data frame error [D]

No

Time-out error [C]

Abnormal termination [B]

Reception status [ACK/other than ACK]

ACK

Other than ACK

Time-out occurs

Status frame received within specified time

Normal completion [A]

Normal data frame? [Yes/No]

Yes

Data frame error [D]

No

Time-out error [C]

Abnormal termination [B]
4.11.2 Description of processing sequence

<1> Waits from the previous frame reception until the next command transmission (wait time tSNx/tDNx).

<2> The Checksum command is transmitted by command frame transmission processing.

<3> A time-out check is performed from command transmission until status frame reception.

   If a time-out occurs, a time-out error [C] is returned (time-out time tCS10).

<4> The status code is checked.

   When ST1 = ACK: Proceeds to <5>.
   When ST1 ≠ ACK: Abnormal termination [B]

<5> A time-out check is performed until data frame (checksum data) reception.

   If a time-out occurs, a time-out error [C] is returned (time-out time tSD10).

<6> The received data frame (checksum data) is checked.

   If data frame is normal: Normal completion [A]
   If data frame is abnormal: Data frame error [D]

4.11.3 Status at processing completion

<table>
<thead>
<tr>
<th>Status at Processing Completion</th>
<th>Status Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal completion [A]</td>
<td>–</td>
<td>The command was executed normally and checksum data was acquired normally.</td>
</tr>
<tr>
<td>Abnormal termination [B]</td>
<td>Parameter error</td>
<td>05H</td>
</tr>
<tr>
<td>Checksum error</td>
<td>07H</td>
<td>The checksum of the transmitted command frame does not match.</td>
</tr>
<tr>
<td>Negative acknowledgment (NACK)</td>
<td>15H</td>
<td>Command frame data is abnormal (such as invalid data length (LEN) or no ETX).</td>
</tr>
<tr>
<td>Time-out error [C]</td>
<td>–</td>
<td>The status frame or data frame was not received within the specified time.</td>
</tr>
<tr>
<td>Data frame error [D]</td>
<td>–</td>
<td>The checksum of the data frame received as checksum data does not match.</td>
</tr>
</tbody>
</table>
4.11.4 Flowchart

Checksum command processing

Wait from previous frame reception until next command transmission

Command frame transmission processing (Checksum)

Status frame received?

Yes

Status = ACK?

Yes

Data frame (checksum data) received?

Normal data frame?

Yes

Normal completion [A]

No

Data frame error [D]

No

Abnormal termination [B]

No

Timed out?

Yes

Time-out error [C]

No

Timed out?

Yes

Time-out error [C]

No
4.12 Security Set Command

4.12.1 Processing sequence chart

Security Set command processing sequence

Programmer

- <1> Wait from previous frame reception until next command transmission
- <2> Security Set command frame transmission
- <3> Time-out check for status frame reception
- <4> Status frame reception
- <5> Wait from previous frame reception until next data frame transmission
- <6> Data frame (security data) transmission
- <7> Time-out check for status frame reception
- <8> Status frame reception

RL78

- Reception status [ACK/other than ACK]
- Time-out occurs: Time-out error [C]
- Other than ACK: Abnormal termination [B]
- Status frame received within specified time: Time-out error [C]
- Other than ACK: Abnormal termination [D]
- ACK: Normal completion [A]
4.12.2 Description of processing sequence

<1> Waits from the previous frame reception until the next command transmission (wait time \( t_{SNx}/t_{DNx} \)).

<2> The Security Set command is transmitted by command frame transmission processing.

<3> A time-out check is performed from command transmission until status frame reception.
    If a time-out occurs, a time-out error [C] is returned (time-out time \( t_{CS7} \)).

<4> The status code is checked.

    When \( ST1 = \text{ACK} \): Proceeds to <5>.
    When \( ST1 \neq \text{ACK} \): Abnormal termination [B]

<5> Waits from the previous frame reception until the next data frame transmission (wait time \( t_{SD7} \)).

<6> The data frame (security setting data) is transmitted by data frame transmission processing.

<7> A time-out check is performed until status frame reception.
    If a time-out occurs, a time-out error [C] is returned (time-out time \( t_{DS7} \)).

<8> The status code is checked.

    When \( ST1 = \text{ACK} \): Normal completion [A]
    When \( ST1 \neq \text{ACK} \): Abnormal termination [D]

4.12.3 Status at processing completion

<table>
<thead>
<tr>
<th>Status at Processing Completion</th>
<th>Status Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal completion [A]</td>
<td>Normal acknowledgment (ACK)</td>
<td>06H</td>
</tr>
<tr>
<td>Abnormal termination [B]</td>
<td>Parameter error</td>
<td>05H</td>
</tr>
<tr>
<td>Checksum error</td>
<td>07H</td>
<td>The checksum of the transmitted command frame or data frame does not match.</td>
</tr>
<tr>
<td>Protect error</td>
<td>10H</td>
<td>An already prohibited flag is to be enabled.</td>
</tr>
<tr>
<td>Negative acknowledgment (NACK)</td>
<td>15H</td>
<td>Command frame data is abnormal (such as invalid data length (LEN) or no ETX).</td>
</tr>
<tr>
<td>Time-out error [C]</td>
<td>–</td>
<td>The status frame or data frame was not received within the specified time.</td>
</tr>
<tr>
<td>Abnormal termination [D]</td>
<td>Erase error</td>
<td>1AH</td>
</tr>
<tr>
<td></td>
<td>IVerify error</td>
<td>1BH</td>
</tr>
<tr>
<td>Write error</td>
<td>1CH</td>
<td></td>
</tr>
</tbody>
</table>
4.12.4 Flowchart

- Security Set command processing
- Wait for previous frame reception until next command transmission
- Command frame transmission processing (Security Set)
- Status frame received?
  - Yes: Status = ACK?
    - Yes: Normal completion [A]
    - No: Abnormal termination [D]
  - No: Timed out?
    - Yes: Timeout error [C]
    - No: Wait for previous frame reception until next data frame transmission
- Data frame transmission processing (Security data)
- Status frame received?
  - Yes: Timed out?
    - Yes: Timeout error [C]
    - No: Status = ACK?
      - Yes: Normal completion [A]
      - No: Abnormal termination [E]
4.13 Security Get Command

4.13.1 Processing sequence chart

Security Get command processing sequence

Programmer

<1> Wait from previous frame reception until next command transmission

<2> Security Get command frame transmission

<3> Time-out check for status frame reception

<4> Status frame reception

<5> Time-out check for data frame reception

<6> Data frame (security data) reception

RL78

tSNx/tDNx

tCSS

tSOS

Time-out occurs

Status frame received within specified time

Reception status [ACK/other than ACK]

ACK

Other than ACK

Abnormal termination [B]

Time-out occurs

Data frame received within specified time

Normal data frame? [Yes/No]

Yes

Normal completion [A]

No

Data frame error [D]

Time-out error [C]
4.13.2 Description of processing sequence

<1> Waits from the previous frame reception until the next command transmission (wait time tSNx/tDNx).

<2> The Security Get command is transmitted by command frame transmission processing.

<3> A time-out check is performed from command transmission until status frame reception.
   If a time-out occurs, a time-out error [C] is returned (time-out time tcsa).

<4> The status code is checked.
   
   When ST1 = ACK: Proceeds to <5>.
   When ST1 ≠ ACK: Abnormal termination [B]

<5> A time-out check is performed until data frame (security data) reception.
   If a time-out occurs, a time-out error [C] is returned (time-out time tside).

<6> The received data frame (version data) is checked.
   
   If data frame is normal: Normal completion [A]
   If data frame is abnormal: Data frame error [D]

4.13.3 Status at processing completion

<table>
<thead>
<tr>
<th>Status at Processing Completion</th>
<th>Status Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal completion [A]</td>
<td>–</td>
<td>The command was executed normally and security setting data was set normally.</td>
</tr>
<tr>
<td>Abnormal termination [B]</td>
<td>Checksum error</td>
<td>07H</td>
</tr>
<tr>
<td></td>
<td>Negative acknowledgment (NACK)</td>
<td>15H</td>
</tr>
<tr>
<td>Normal acknowledgment (ACK)</td>
<td>06H</td>
<td>The command was transmitted normally.</td>
</tr>
<tr>
<td>Time-out error [C]</td>
<td>–</td>
<td>The status frame or data frame was not received within the specified time.</td>
</tr>
<tr>
<td>Data frame error [D]</td>
<td>–</td>
<td>The checksum of the data frame received as security data does not match.</td>
</tr>
</tbody>
</table>
4.13.4 Flowchart

Security Get command processing

Wait from previous frame reception until next command transmission

Command frame transmission processing (Security Get)

Status frame received?

Yes

Status = ACK?

Yes

Normal completion [A]

No

Abnormal termination [B]

No

Data frame (security data) received?

Yes

Timed out? $t_{SS}$

Yes

Time-out error [C]

No

Normal data frame?

Yes

Timed out? $t_{SD}$

Yes

Time-out error [C]

No

Data frame error [D]
4.14 Security Release Command

4.14.1 Processing sequence chart

The Security Release command can be executed only when all the code flash area and data flash area are blank. Therefore, the block erase processing indicated by the dotted line above must be executed in advance.
4.14.2 Description of processing sequence

<1> Waits from the previous frame reception until the next command transmission (wait time tSNx/tDNx).

<2> The Block Erase command is transmitted by command frame transmission processing.

<3> A time-out check is performed from command transmission until status frame reception.
   If a time-out occurs, a time-out error [C] is returned (time-out time tCS3).

<4> The status code is checked.

When ST1 = ACK: Proceed to <5> if all the blocks have been erased.
   Return to <1> if the specified block has not been erased.
When ST1 ≠ ACK: Abnormal termination [B]

<5> Waits from the previous frame reception until the next command transmission (wait time tSN3).

<6> The Security Release command is transmitted by command frame transmission processing.

<7> A time-out check is performed from command transmission until status frame reception.
   If a time-out occurs, a time-out error [C] is returned (time-out time tCS3).

<8> The status code is checked.

When ST1 = ACK: Normal completion [A]
   Execute the following processing.
   To execute Security Release command alone:
      Output the low level to the Reset pin and terminate.
   To execute a command immediately after Security Release command:
      Execute mode re-setting processing.

When ST1 ≠ ACK: Abnormal termination [B]

4.14.3 Status at processing completion

The following table shows the status codes that may be output after the Security Release command has been executed.

<table>
<thead>
<tr>
<th>Status at Processing Completion</th>
<th>Status Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal completion [A]</td>
<td>Normal acknowledgment (ACK)</td>
<td>06H</td>
</tr>
<tr>
<td>Abnormal termination [B]</td>
<td>Checksum error</td>
<td>07H</td>
</tr>
<tr>
<td></td>
<td>Protect error</td>
<td>10H</td>
</tr>
<tr>
<td></td>
<td>Negative acknowledgment (NACK)</td>
<td>15H</td>
</tr>
<tr>
<td></td>
<td>Blank error</td>
<td>1BH</td>
</tr>
<tr>
<td></td>
<td>Erase error/IVerify error/Write error</td>
<td>1AH/1BH/1CH</td>
</tr>
<tr>
<td>Time-out error [C]</td>
<td>–</td>
<td>The status frame was not received within the specified time.</td>
</tr>
</tbody>
</table>
The Security Release command can be executed only when all the code flash area and data flash area are blank. Therefore, the block erase processing indicated by the dotted line above must be executed in advance.
Mode re-setting processing

Low output to Reset

Low output to TOOL0

Reset release

TOOL0 pulled up

Waits for mode setting data.

Mode setting 1-byte data
@115,200 bps

Communication mode determined

Mode setting data
Single-wire UART: 3AH
Two-wire UART: 00H

Single-wire UART: Input/output Tool0
Two-wire UART: Input RxD, output TxD

Baud Rate Set command processing

Normal completion
CHAPTER 5  FLASH MEMORY PROGRAMMING PARAMETER CHARACTERISTICS

This chapter describes the characteristics of parameter transmitted between the programmer and the devices (RL78) in the flash memory programming mode. Refer to the user’s manual of the devices (RL78) for electrical specifications when designing a programmer.

5.1 Flash Memory Parameter Characteristics of RL78

5.1.1 Flash memory parameter characteristics in full-speed mode

(1) Flash memory programming mode setting time

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>MIN.</th>
<th>TYP.</th>
<th>MAX.</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOOL0↓ to RESET↑</td>
<td>tTR</td>
<td></td>
<td>tSU</td>
<td></td>
</tr>
<tr>
<td>RESET↑ to TOOL0↑</td>
<td>tRT</td>
<td>723μs</td>
<td>16μs</td>
<td></td>
</tr>
<tr>
<td>TOOL0↑ to Receive mode info</td>
<td>tTM</td>
<td>16μs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receive mode info to Receive Baud Rate Set Command</td>
<td>tMB</td>
<td>62μs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RESET↑ to Receive Baud Rate Set Command</td>
<td>tRB</td>
<td>100ms</td>
<td></td>
<td>100ms</td>
</tr>
</tbody>
</table>

Notes 1. The location indicated differs depending on the setting of the option byte.

- Option byte (0000C3H.bit6) = 0: RESET↑ to Baud Rate Set Command reception: within 100 ms
- = 1: TOOL0↑ to Baud Rate Set command reception: within 100 ms

To permit both of the option byte settings, complete the reception of the Baud Rate Set command within 100 ms from RESET↑ (refer to the User’s Manual of each device (tSUINIT)).

2. The flash memory programmer needs the specified wait time before transmission.
## (2) Programming characteristic

<table>
<thead>
<tr>
<th>Wait</th>
<th>Condition</th>
<th>Symbol</th>
<th>MIN.</th>
<th>MAX.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Between data transmissions</td>
<td>Data reception</td>
<td>t_{SR}</td>
<td>0 μs</td>
<td>Note 3</td>
</tr>
<tr>
<td></td>
<td>16 MHz ≤ f_{CLK} ≤ 32 MHz</td>
<td></td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.75 MHz ≤ f_{CLK} &lt; 16 MHz</td>
<td></td>
<td>136/f_{CLK} - 8 μs</td>
<td>Note 3</td>
</tr>
<tr>
<td></td>
<td>Data transmission</td>
<td>t_{DT}</td>
<td>6/f_{CLK}</td>
<td>Note 2</td>
</tr>
<tr>
<td></td>
<td>16 MHz ≤ f_{CLK} ≤ 32 MHz</td>
<td></td>
<td>10/f_{CLK}</td>
<td>Note 3</td>
</tr>
<tr>
<td>From status frame transmission until data frame transmission</td>
<td>Check Sum command</td>
<td>t_{SD10}</td>
<td>48/f_{CLK}+</td>
<td></td>
</tr>
<tr>
<td></td>
<td>15564/f_{CLK} x BLK Note 2, 4</td>
<td></td>
<td>30720/f_{CLK} x BLK</td>
<td></td>
</tr>
<tr>
<td>From status frame transmission until data frame reception (1)</td>
<td>Programming command</td>
<td>t_{SD5}</td>
<td>41/f_{CLK}</td>
<td>Note 3</td>
</tr>
<tr>
<td>From status frame transmission until data frame reception (2)</td>
<td>Verify command</td>
<td>t_{SD2}</td>
<td>41/f_{CLK}</td>
<td>Note 3</td>
</tr>
<tr>
<td>From status frame transmission until data frame reception (3)</td>
<td>Security Set command</td>
<td>t_{SD7}</td>
<td>32/f_{CLK}</td>
<td>Note 3</td>
</tr>
<tr>
<td>From status frame transmission until data frame reception (4)</td>
<td>Security Get command</td>
<td>t_{SD8}</td>
<td>139/f_{CLK}</td>
<td>Note 3</td>
</tr>
<tr>
<td></td>
<td>212/f_{CLK}</td>
<td>Note 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>From status frame transmission until data frame reception (5)</td>
<td>Signature command</td>
<td>t_{SD11}</td>
<td>340/f_{CLK}</td>
<td>Note 2</td>
</tr>
<tr>
<td></td>
<td>512/f_{CLK}</td>
<td>Note 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>From status frame transmission until next command frame reception</td>
<td>Reset command</td>
<td>t_{SN1}</td>
<td>51/f_{CLK}</td>
<td>Note 3</td>
</tr>
<tr>
<td></td>
<td>Verify command</td>
<td>t_{SN2}</td>
<td>54/f_{CLK}</td>
<td>Note 3</td>
</tr>
<tr>
<td></td>
<td>Block Erase command</td>
<td>t_{SN3}</td>
<td>51/f_{CLK}</td>
<td>Note 3</td>
</tr>
<tr>
<td></td>
<td>Block Blank Check command</td>
<td>t_{SN4}</td>
<td>51/f_{CLK}</td>
<td>Note 3</td>
</tr>
<tr>
<td></td>
<td>Programming command</td>
<td>t_{SN5}</td>
<td>51/f_{CLK}</td>
<td>Note 3</td>
</tr>
<tr>
<td></td>
<td>Baud Rate Set command</td>
<td>t_{SN6}</td>
<td>67 μs</td>
<td>Note 3</td>
</tr>
<tr>
<td></td>
<td>Security Set command</td>
<td>t_{SN7}</td>
<td>51/f_{CLK}</td>
<td>Note 3</td>
</tr>
<tr>
<td></td>
<td>Security Release command</td>
<td>t_{SN8}</td>
<td>51/f_{CLK}</td>
<td>Note 3</td>
</tr>
<tr>
<td>From data frame transmission until next command frame reception</td>
<td>Security Get command</td>
<td>t_{SN9}</td>
<td>44/f_{CLK}</td>
<td>Note 3</td>
</tr>
<tr>
<td></td>
<td>Checksum command</td>
<td>t_{SN10}</td>
<td>44/f_{CLK}</td>
<td>Note 3</td>
</tr>
<tr>
<td></td>
<td>Signature command</td>
<td>t_{SN11}</td>
<td>44/f_{CLK}</td>
<td>Note 3</td>
</tr>
</tbody>
</table>

**Notes**

1. This is as a guide of the time-out time.
2. The Flash memory programmer must be ready for receiving communication data within the specified time.
3. The Flash memory programmer needs the specified time before transmission.
4. The detail of the symbol is as follows.
   - **BLK**: Number of blocks (in 1024-byte units)
5. The device operates at either 0.75 MHz or 1 MHz since reset release until reception of the Baud Rate Set command. Calculate the time on the assumption that f_{CLK} = 0.75 MHz so that communication can be executed during this period.
## (3) Command characteristics

<table>
<thead>
<tr>
<th>Command</th>
<th>Symbol</th>
<th>Condition</th>
<th>MIN.</th>
<th>MAX.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Reset</strong></td>
<td>tCS1</td>
<td>–</td>
<td>58/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>255/fCLK&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
<tr>
<td><strong>Verify</strong></td>
<td>tCS2</td>
<td>Code flash</td>
<td>58/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>335/fCLK&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Data flash</td>
<td>58/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>351/fCLK&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>tCS2</td>
<td>Code flash</td>
<td>64/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>11981/fCLK&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Data flash</td>
<td>64/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>11980/fCLK&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
<tr>
<td><strong>Block Erase</strong></td>
<td>tCS3</td>
<td>Code flash</td>
<td>58/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>67731/fCLK+255098 μs&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Data flash</td>
<td>58/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>281423/fCLK+264790 μs&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
<tr>
<td><strong>Block Blank Check</strong></td>
<td>tCS4</td>
<td>Code flash</td>
<td>58/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>3805/fCLK&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Data flash</td>
<td>58/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>2503/fCLK+86 μs+ (5827/fCLK+318 μs)×BLK&lt;sup&gt;Note 1, 3&lt;/sup&gt;</td>
</tr>
<tr>
<td><strong>Programming</strong></td>
<td>tCS5</td>
<td>Code flash</td>
<td>58/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>1432/fCLK&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Data flash</td>
<td>58/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>346/fCLK&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>tCS5</td>
<td>Code flash</td>
<td>64/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>113502/fCLK+71753 μs&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Data flash</td>
<td>64/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>309870/fCLK+219761 μs&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>tCS6</td>
<td>Code flash</td>
<td>1294/fCLK+37 μs&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>1732/fCLK+36 μs+ (7096/fCLK+892 μs)×BLK+ (182/fCLK+17 μs)×N&lt;sup&gt;Note 1, 3&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Data flash</td>
<td>282/fCLK+22 μs&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>397/fCLK+30 μs+ (28382/fCLK+3568 μs)×BLK&lt;sup&gt;Note 1, 2&lt;/sup&gt;</td>
</tr>
<tr>
<td><strong>Baud Rate Set</strong></td>
<td>tCS6</td>
<td>–</td>
<td>58 μs&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>4735 μs&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
<tr>
<td><strong>Security Set</strong></td>
<td>tCS7</td>
<td>–</td>
<td>58/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>168/fCLK&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>tCS7</td>
<td>–</td>
<td>60/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>277095/fCLK+1027564 μs&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
<tr>
<td><strong>Security Get</strong></td>
<td>tCS9</td>
<td>–</td>
<td>58/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>154/fCLK&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
<tr>
<td><strong>Security Release</strong></td>
<td>tCS9</td>
<td>Model with data flash</td>
<td>58/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>146110/fCLK+511868 μs+ (1457/fCLK+80 μs)×CBLK+ (5827/fCLK+318 μs)×DBLK+ (203/fCLK+18 μs)×N&lt;sup&gt;Note 1, 4&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model without data flash</td>
<td>58/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>145783/fCLK+511837 μs+ (1457/fCLK+80 μs)×CBLK+ (203/fCLK+18 μs)×N&lt;sup&gt;Note 1, 4&lt;/sup&gt;</td>
</tr>
<tr>
<td><strong>Checksum</strong></td>
<td>tCS10</td>
<td>Code flash</td>
<td>58/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>203/fCLK&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Data flash</td>
<td>58/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>219/fCLK&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
<tr>
<td><strong>Signature Get</strong></td>
<td>tCS11</td>
<td>–</td>
<td>58/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>111/fCLK&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
</tbody>
</table>

**Notes**

1. This is as a guide of the time-out time.
2. The Flash memory programmer must be ready for receiving communication data within the specified time.
3. The details of the symbols are as follows.
   - **BLK**: Number of blocks (in 1024-byte units)
   - **N**: Number of times Flash memory is to be accessed

Expression (Result of division in parentheses is rounded off at decimal point):

\[
\frac{End \ address - Start \ address}{40000H} + 1
\]

**Example**

Start address = 00000H & End address = 3FFFFH → N = 1
Start address = 3FC00H & End address = 403FFH → N = 2
4. The details of the symbols are as follows.

- **CBLK**: Total number of blocks of CodeFlash
  
  **Example**: Code Flash size = 64 KB → CBLK = 64

- **DBLK**: Total number of blocks of DataFlash
  
  **Example**: Data Flash size = 4 KB → DBLK = 4

- **N**: Number of times Flash memory is to be accessed
  
  **Expression (Rounded off at decimal point)**: CBLK/256
  
  **Example**: CBLK = 256 → N = 1
  
  CBLKK = 384 → N = 2
### 5.1.2 Flash memory parameter characteristics in wide-voltage mode

#### (1) Flash memory programming mode setting time

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>MIN.</th>
<th>TYP.</th>
<th>MAX.</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOOL0↓ to RESET↑</td>
<td>tTR</td>
<td></td>
<td>tSU</td>
<td></td>
</tr>
<tr>
<td>RESET↑ to TOOL0↑</td>
<td>tIT</td>
<td>723 μs</td>
<td>tSU</td>
<td></td>
</tr>
<tr>
<td>TOOL0↑ to Receive mode info</td>
<td>tTW</td>
<td>16 μs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receive mode info to Receive Baud Rate Set Command</td>
<td>tMB</td>
<td>62 μs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RESET↑ to Receive Baud Rate Set Command</td>
<td>tMB</td>
<td></td>
<td>100 ms</td>
<td></td>
</tr>
</tbody>
</table>

#### Notes

1. The location indicated differs depending on the setting of the option byte.
   - Option byte (000C3H.bit6) = 0: RESET↑ to Baud Rate Set Command reception: within 100 ms
   - = 1: TOOL0↑ to Baud Rate Set command reception: within 100 ms
   - To permit both of the option byte settings, complete the reception of the Baud Rate Set command within 100 ms from RESET↑ (refer to the User’s Manual of each device (TSUNIT)).
2. The flash memory programmer needs the specified wait time before transmission.
## (2) Programming characteristic

<table>
<thead>
<tr>
<th>Wait</th>
<th>Condition</th>
<th>Symbol</th>
<th>MIN.</th>
<th>MAX.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Between data transmissions</td>
<td>Data reception&lt;sup&gt;Note 5&lt;/sup&gt; 16 MHz ≤ fCLK &lt; 32 MHz</td>
<td>t&lt;sub&gt;dr&lt;/sub&gt;</td>
<td>0 μs&lt;sup&gt;Note 3&lt;/sup&gt;</td>
<td>16/CLK&lt;sup&gt;Note 3&lt;/sup&gt; = 8 μs&lt;sup&gt;Note 3&lt;/sup&gt;</td>
</tr>
<tr>
<td>Between data transmissions</td>
<td>Data transmission&lt;sup&gt;Note 4&lt;/sup&gt;</td>
<td>t&lt;sub&gt;DR&lt;/sub&gt;</td>
<td>6/CLK&lt;sup&gt;Note 1&lt;/sup&gt;</td>
<td>10/CLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
</tr>
<tr>
<td>From status frame transmission until data frame transmission</td>
<td>Check Sum command</td>
<td>t&lt;sub&gt;SD10&lt;/sub&gt;</td>
<td>48/CLK + 15564/CLK X BLK&lt;sup&gt;Note 2, 4&lt;/sup&gt;</td>
<td>72/CLK&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
<tr>
<td>From status frame transmission until data frame reception (1)</td>
<td>Programming command</td>
<td>t&lt;sub&gt;SD6&lt;/sub&gt;</td>
<td>41/CLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>From status frame transmission until data frame reception (2)</td>
<td>Verify command</td>
<td>t&lt;sub&gt;SD2&lt;/sub&gt;</td>
<td>41/CLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>From status frame transmission until data frame reception (3)</td>
<td>Security Set command</td>
<td>t&lt;sub&gt;SD7&lt;/sub&gt;</td>
<td>32/CLK&lt;sup&gt;Note 3&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>From status frame transmission until data frame reception (4)</td>
<td>Security Get command</td>
<td>t&lt;sub&gt;SD8&lt;/sub&gt;</td>
<td>139/CLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>212/CLK&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
<tr>
<td>From status frame transmission until data frame reception (5)</td>
<td>Signature command</td>
<td>t&lt;sub&gt;SD11&lt;/sub&gt;</td>
<td>340/CLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>512/CLK&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
<tr>
<td>From status frame transmission until next command frame reception</td>
<td>Reset command</td>
<td>t&lt;sub&gt;SN1&lt;/sub&gt;</td>
<td>51/CLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>From status frame transmission until next command frame reception</td>
<td>Verify command</td>
<td>t&lt;sub&gt;SN2&lt;/sub&gt;</td>
<td>54/CLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>From status frame transmission until next command frame reception</td>
<td>Block Erase command</td>
<td>t&lt;sub&gt;SN3&lt;/sub&gt;</td>
<td>51/CLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>From status frame transmission until next command frame reception</td>
<td>Block Blank Check command</td>
<td>t&lt;sub&gt;SN4&lt;/sub&gt;</td>
<td>51/CLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>From status frame transmission until next command frame reception</td>
<td>Programming command</td>
<td>t&lt;sub&gt;SN5&lt;/sub&gt;</td>
<td>51/CLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>From status frame transmission until next command frame reception</td>
<td>Baud Rate Set command</td>
<td>t&lt;sub&gt;SN6&lt;/sub&gt;</td>
<td>67 μs&lt;sup&gt;Note 3&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>From status frame transmission until next command frame reception</td>
<td>Security Set command</td>
<td>t&lt;sub&gt;SN7&lt;/sub&gt;</td>
<td>51/CLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>From status frame transmission until next command frame reception</td>
<td>Security Release command</td>
<td>t&lt;sub&gt;SN9&lt;/sub&gt;</td>
<td>51/CLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>From data frame transmission until next command frame reception</td>
<td>Security Get command</td>
<td>t&lt;sub&gt;SN8&lt;/sub&gt;</td>
<td>44/CLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>From data frame transmission until next command frame reception</td>
<td>Checksum command</td>
<td>t&lt;sub&gt;SN10&lt;/sub&gt;</td>
<td>44/CLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>From data frame transmission until next command frame reception</td>
<td>Signature command</td>
<td>t&lt;sub&gt;SN11&lt;/sub&gt;</td>
<td>44/CLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td></td>
</tr>
</tbody>
</table>

**Notes**

1. This is as a guide of the time-out time.
2. The Flash memory programmer must be ready for receiving communication data within the specified time.
3. The Flash memory programmer needs the specified time before transmission.
4. The detail of the symbol is as follows.
   - BLK: Number of blocks (in 1024-byte units)
5. The device operates at either 0.75 MHz or 1 MHz since reset release until reception of the Baud Rate Set command. Calculate the time on the assumption that fclk = 0.75 MHz so that communication can be executed during this period.
## (3) Command characteristics

<table>
<thead>
<tr>
<th>Command</th>
<th>Symbol</th>
<th>Condition</th>
<th>MIN.</th>
<th>MAX.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reset</td>
<td>tCS1</td>
<td>–</td>
<td>58/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>255/fCLK&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
<tr>
<td>Verify</td>
<td>tCS2</td>
<td>Code flash</td>
<td>58/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>335/fCLK&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Data flash</td>
<td>58/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>351/fCLK&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Code flash</td>
<td>64/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>11981/fCLK&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Data flash</td>
<td>64/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>11980/fCLK&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
<tr>
<td>Block Erase</td>
<td>tCS3</td>
<td>Code flash</td>
<td>58/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>59455/fCLK+265331 µs&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Data flash</td>
<td>58/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>248862/fCLK+299307 µs&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
<tr>
<td>Block Blank Check</td>
<td>tCS4</td>
<td>Code flash</td>
<td>58/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>1732/fCLK+36 µs+ (1259/fCLK+278 µs)xBLK+ (199/fCLK+57 µs)xN&lt;sup&gt;Note 1, 3&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Data flash</td>
<td>58/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>2494/fCLK+168 µs+ (5035/fCLK+1110 µs)xBLK&lt;sup&gt;Note 1, 3&lt;/sup&gt;</td>
</tr>
<tr>
<td>Programming</td>
<td>tCS5</td>
<td>Code flash</td>
<td>58/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>1432/fCLK&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Data flash</td>
<td>58/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>346/fCLK&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Code flash</td>
<td>64/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>107803/fCLK+138891 µs&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Data flash</td>
<td>64/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>287076/fCLK+488315 µs&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>tSS5</td>
<td>–</td>
<td>276/fCLK+57 µs&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>398/fCLK+58 µs+ (17403/fCLK+29293 µs)xBLK&lt;sup&gt;Note 1, 3&lt;/sup&gt;</td>
</tr>
<tr>
<td>Baud Rate Set</td>
<td>tCS6</td>
<td>–</td>
<td>58 µs&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>4735 µs&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
<tr>
<td>Security Set</td>
<td>tCS7</td>
<td>–</td>
<td>58/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>168/fCLK&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>tCS8</td>
<td>–</td>
<td>60/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>242909/fCLK+1075967 µs&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
<tr>
<td>Security Get</td>
<td>tCS9</td>
<td>–</td>
<td>58/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>154/fCLK&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
<tr>
<td>Security Release</td>
<td>tCS10</td>
<td>Model with data flash</td>
<td>58/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>128408/fCLK+534723 µs+ (1259/fCLK+278 µs)xCBLK+ (5035/fCLK+1110 µs)xDBLK+ (199/fCLK+57 µs)xN&lt;sup&gt;Note 1, 4&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model without data flash</td>
<td>58/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>128084/fCLK+534653 µs+ (1259/fCLK+278 µs)xCBLK+ (199/fCLK+57 µs)xN&lt;sup&gt;Note 1, 4&lt;/sup&gt;</td>
</tr>
<tr>
<td>Checksum</td>
<td>tCS11</td>
<td>Code flash</td>
<td>58/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>203/fCLK&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Data flash</td>
<td>58/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>219/fCLK&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
<tr>
<td>Signature Get</td>
<td>tCS11</td>
<td>–</td>
<td>58/fCLK&lt;sup&gt;Note 2&lt;/sup&gt;</td>
<td>111/fCLK&lt;sup&gt;Note 1&lt;/sup&gt;</td>
</tr>
</tbody>
</table>

**Notes**

1. This is as a guide of the time-out time.
2. The Flash memory programmer must be ready for receiving communication data within the specified time.
3. The details of the symbols are as follows.
   - **BLK**: Number of blocks (in 1024-byte units)
   - **N**: Number of times Flash memory is to be accessed
   - Expression (Result of division in parentheses is rounded off at decimal point):
     \[(\text{End address}/40000H) – (\text{Start address}/40000H) + 1\]
   - **Example**
     - Start address = 00000H & End address = 3FFFFH → N = 1
     - Start address = 3FC00H & End address = 403FFH → N = 2
4. The details of the symbols are as follows.
   CBLK: Total number of blocks of CodeFlash
   [Example] Code Flash size = 64 KB → CBLK = 64
   DBLK: Total number of blocks of DataFlash
   [Example] Data Flash size = 4 KB → DBLK = 4
   N: Number of times Flash memory is to be accessed
   Expression (Rounded off at decimal point): CBLK/256
   [Example] CBLK = 256 → N = 1
       CBLKK = 384 → N = 2
### 5.2 UART Communication Mode

In the figure below, TOOL0 is illustrated as two separate lines for the sake of description, but it is actually a single line. The VDD level of TOOL0 can be achieved by using a pull-up resistor (the pin is Hi-Z).

(a) Data frame

<table>
<thead>
<tr>
<th>Single-wire UART</th>
<th>RL78 → Programmer</th>
<th>Programmer → RL78</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOOL0 (output)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOOL0 (input)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Two-wire UART</th>
<th>RL78 → Programmer</th>
<th>Programmer → RL78</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOOLTxD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOOLRxD</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
(b) Programming mode setting

**Single-wire UART**

<table>
<thead>
<tr>
<th>VDD</th>
<th>RESET</th>
<th>TOOL0</th>
<th>&quot;3Ah&quot; @ 115,200 bps</th>
<th>1-byte data for setting mode</th>
<th>Baud Rate Set command frame</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>tTR</td>
<td>tBT</td>
<td>tTM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>tMB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>tMB</td>
</tr>
</tbody>
</table>

**Two-wire UART**

<table>
<thead>
<tr>
<th>VDD</th>
<th>RESET</th>
<th>TOOL0</th>
<th>FP_TxD</th>
<th>FP_RxD</th>
<th>&quot;00h&quot; @ 115,200 bps</th>
<th>1-byte data for setting mode</th>
<th>Baud Rate Set command frame</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>tTR</td>
<td>tBT</td>
<td>tTM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>tMB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>tMB</td>
</tr>
</tbody>
</table>
(c) Reset command/Block Erase command/Block Blank Check command/Baud Rate Set command/Security Release command

**Single-wire UART**

```
TOOLO (output)  Command frame  Status frame  Next command frame
```

```
TOOLO (input)  Command frame  Status frame  Next command frame
```

```
tCS1, 3, 4, 6, 9  tSN1, 3, 4, 6, 9
```

**Two-wire UART**

```
TOOLTxD  Command frame  Status frame  Next command frame
```

```
TOOLRxD  Command frame  Status frame  Next command frame
```

```
tCS1, 3, 4, 6, 9  tSN1, 3, 4, 6, 9
```

(d) Verify command/Security Set command

**Single-wire UART**

```
TOOLO (output)  Command frame  Data frame  Data frame (1)  Status frame (1)  Status frame (2)  Data frame (n)  Status frame (n)  Next command frame
```

```
TOOLO (input)  Command frame  Data frame  Data frame (1)  Status frame (1)  Status frame (2)  Data frame (n)  Status frame (n)  Next command frame
```

```
tCS2, 7  tCS2, 7  tCS2, 7  tCS2, 7  tCS2, 7  tCS2, 7  tCS2, 7  tCS2, 7
```

**Two-wire UART**

```
TOOLTxD  Command frame  Data frame  Data frame (1)  Status frame (1)  Status frame (2)  Data frame (n)  Status frame (n)  Next command frame
```

```
TOOLRxD  Command frame  Data frame  Data frame (1)  Status frame (1)  Status frame (2)  Data frame (n)  Status frame (n)  Next command frame
```

```
tCS2, 7  tCS2, 7  tCS2, 7  tCS2, 7  tCS2, 7  tCS2, 7  tCS2, 7  tCS2, 7  tCS2, 7
```
(e) Programming command

Single-wire UART

TOOL0 (output)

TOOL0 (input)

Two-wire UART

TOOLTxD

TOOLRxD

(f) Security Get command/Check Sum command/Signature command

Single-wire UART

TOOL0 (output)

TOOL0 (input)

Two-wire UART

TOOLTxD

TOOLRxD
Website and Support

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

Inquiries
http://www.renesas.com/inquiry
## Revision Record

<table>
<thead>
<tr>
<th>Rev.</th>
<th>Date</th>
<th>Page</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>Nov. 7, 2011</td>
<td>–</td>
<td>First edition issued</td>
</tr>
</tbody>
</table>

All trademarks and registered trademarks are the property of their respective owners.
NOTES FOR CMOS DEVICES

(1) VOLTAGE APPLICATION WAVEFORM AT INPUT PIN: Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between VIL (MAX) and VIH (MIN) due to noise, etc., the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between VIL (MAX) and VIH (MIN).

(2) HANDLING OF UNUSED INPUT PINS: Unconnected CMOS device inputs can be cause of malfunction. If an input pin is unconnected, it is possible that an internal input level may be generated due to noise, etc., causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using pull-up or pull-down circuitry. Each unused pin should be connected to VDD or GND via a resistor if there is a possibility that it will be an output pin. All handling related to unused pins must be judged separately for each device and according to related specifications governing the device.

(3) PRECAUTION AGAINST ESD: A strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it when it has occurred. Environmental control must be adequate. When it is dry, a humidifier should be used. It is recommended to avoid using insulators that easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors should be grounded. The operator should be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with mounted semiconductor devices.

(4) STATUS BEFORE INITIALIZATION: Power-on does not necessarily define the initial status of a MOS device. Immediately after the power source is turned ON, devices with reset functions have not yet been initialized. Hence, power-on does not guarantee output pin levels, I/O settings or contents of registers. A device is not initialized until the reset signal is received. A reset operation must be executed immediately after power-on for devices with reset functions.

(5) POWER ON/OFF SEQUENCE: In the case of a device that uses different power supplies for the internal operation and external interface, as a rule, switch on the external power supply after switching on the internal power supply. When switching the power supply off, as a rule, switch off the external power supply and then the internal power supply. Use of the reverse power on/off sequences may result in the application of an overvoltage to the internal elements of the device, causing malfunction and degradation of internal elements due to the passage of an abnormal current. The correct power on/off sequence must be judged separately for each device and according to related specifications governing the device.

(6) INPUT OF SIGNAL DURING POWER OFF STATE: Do not input signals or an I/O pull-up power supply while the device is not powered. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Input of signals during the power off state must be judged separately for each device and according to related specifications governing the device.
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 example. 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 manufacturers, 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.

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 against the possibility of physical injury, and injury or damage caused by fire and 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 reconfigurable computer 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.

SALES OFFICES
Renesas Electronics Corporation
http://www.renesas.com

Refer to "http://www.renesas.com" for the latest and detailed information.

Renesas Electronics America Inc.
2960 South Boulevard (Santa Clara, CA 95056-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1201 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5941, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Boynes End, Buckinghamshire, SL8 5FH, U.K.
Tel: +44-1628-986-100, Fax: +44-1628-986-900

Renesas Electronics Europe GmbH
Arcadialandstrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 Zhonglu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1150, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 202, 205, A2A Center, No.1523 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6687-7858 / 7898

Renesas Electronics Hong Kong Limited
Unit 19F, Cheung Kong Centre, 133-135, Western Harbour Road, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852-2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 Harbourfront Avenue, #05-10, Keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6213-0200

Renesas Electronics Malaysia Sdn. Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46500 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-8300, Fax: +60-3-7955-8510

Renesas Electronics Korea Co., Ltd.
13F, Santin Landin Bldg., 770-2, Teksin-dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3377, Fax: +82-2-558-5141