RENESAS

Customer Notification

78K0R/Fx3

16-Bit Single-Chip Microcontroller

Operating Precautions

78K0R/FB3 Series

78K0R/FC3 Series

78K0R/FE3 Series

78K0R/FF3 Series

78K0R/FG3 Series

Renesas Electronics
www.renesas.com

Document No. R01TU0003ED0103
Date Published: November 2011

Notice

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

2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.

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

4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.

5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.

6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.

7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.
   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti- crime systems; safety equipment; and medical equipment not specifically designed for life support.
   “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.

8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.

9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.

10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority- owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

Table of Contents

A) Related Products

List of related products:

78K0R/FB3

Non A-Version Products

μPD78F1804MCA, μPD78F1805MCA, μPD78F1806MCA, μPD78F1807MCA,
μPD78F1804MCA2, μPD78F1805MCA2, μPD78F1806MCA2, μPD78F1807MCA2

A-Version Products

μPD78F1804AMCA, μPD78F1805AMCA, μPD78F1806AMCA, μPD78F1807AMCA,
μPD78F1804AMCA2, μPD78F1805AMCA2, μPD78F1806AMCA2, μPD78F1807AMCA2

78K0R/FC3

Non A-Version Products

μPD78F1808K8A, μPD78F1809K8A, μPD78F1810K8A, μPD78F1811K8A,
μPD78F1808K8A2, μPD78F1809K8A2, μPD78F1810K8A2, μPD78F1811K8A2

A-Version Products

μPD78F1808AK8A, μPD78F1809AK8A, μPD78F1810AK8A, μPD78F1811AK8A,
μPD78F1808AK8A2, μPD78F1809AK8A2, μPD78F1810AK8A2, μPD78F1811AK8A2

Non A-version Products

μPD78F1812GAA, μPD78F1813GAA, μPD78F1814GAA, μPD78F1815GAA, μPD78F1816GAA,
μPD78F1817GAA, μPD78F1826GAA, μPD78F1827GAA, μPD78F1828GAA, μPD78F1829GAA,
μPD78F1830GAA,
μPD78F1812GAA2, μPD78F1813GAA2, μPD78F1814GAA2, μPD78F1815GAA2, μPD78F1816GAA2,
μPD78F1817GAA2, μPD78F1826GAA2, μPD78F1827GAA2, μPD78F1828GAA2, μPD78F1829GAA2,
μPD78F1830GAA2

A-Version Products

μPD78F1812AGAA, μPD78F1813AGAA, μPD78F1814AGAA, μPD78F1815AGAA,
μPD78F1816AGAA, μPD78F1817AGAA, μPD78F1826AGAA, μPD78F1827AGAA,
μPD78F1828AGAA, μPD78F1829AGAA, μPD78F1830AGAA, μPD78F1812AGAA2,
μPD78F1813AGAA2, μPD78F1814AGAA2, μPD78F1815AGAA2, μPD78F1816AGAA2,
μPD78F1817AGAA2, μPD78F1826AGAA2, μPD78F1827AGAA2, μPD78F1828AGAA2,
μPD 78F1829AGAA2, μPD78F1830AGAA2

Non A-version Products

μPD78F1812K8A, μPD78F1813K8A, μPD78F1814K8A, μPD78F1815K8A, μPD78F1816K8A,
μPD78F1817K8A, μPD78F1826K8A, μPD78F1827K8A, μPD78F1828K8A, μPD78F1829K8A,
μPD78F1830K8A
μPD78F1812K8A2, μPD78F1813K8A2, μPD78F1814K8A2, μPD78F1815K8A2, μPD78F1816K8A2,
μPD78F1817K8A2, μPD78F1826K8A2, μPD78F1827K8A2, μPD78F1828K8A2, μPD78F1829K8A2,
μPD78F1830K8A2

A-Version Products

μPD78F1812AK8A, μPD78F1813AK8A, μPD78F1814AK8A, μPD78F1815AK8A, μPD78F1816AK8A,
μPD78F1817AK8A, μPD78F1826AK8A, μPD78F1827AK8A, μPD78F1828AK8A, μPD78F1829AK8A,
μPD78F1830AK8A
μPD78F1812AK8A2, μPD78F1813AK8A2, μPD78F1814AK8A2, μPD78F1815AK8A2,
μPD78F1816AK8A2, μPD78F1817AK8A2, μPD78F1826AK8A2, μPD78F1827AK8A2,
μPD78F1828AK8A2, μPD78F1829AK8A2, μPD78F1830AK8A2

78K0R/FE3

Non A-Version Products

μPD78F1818GBA, μPD78F1819GBA, μPD78F1820GBA, μPD78F1821GBA, μPD78F1822GBA, μPD78F1831GBA, μPD78F1832GBA, μPD78F1833GBA, μPD78F1834GBA, μPD78F1835GBA, μPD78F1818GBA2, μPD78F1819GBA2, μPD78F1820GBA2, μPD78F1821GBA2, μPD78F1822GBA2, μPD78F1831GBA2, μPD78F1832GBA2, μPD78F1833GBA2, μPD78F1834GBA2, μPD78F1835GBA2

A-Version Products

μPD78F1818AGBA, μPD78F1819AGBA, μPD78F1820AGBA, μPD78F1821AGBA, μPD78F1822AGBA, μPD78F1831AGBA, μPD78F1832AGBA, μPD78F1833AGBA, μPD78F1834AGBA, μPD78F1835AGBA, μPD78F1818AGBA2, μPD78F1819AGBA2, μPD78F1820AGBA2, μPD78F1821AGBA2, μPD78F1822AGBA2, μPD78F1831AGBA2, μPD78F1832AGBA2, μPD78F1833AGBA2, μPD78F1834AGBA2, μPD78F1835AGBA2

78K0R/FF3

Non A-Version Products

μPD78F1823GKA, μPD78F1824GKA, μPD78F1825GKA, μPD78F1836GKA, μPD78F1837GKA, μPD78F1838GKA, μPD78F1839GKA, μPD78F1840GKA, μPD78F1823GKA2, μPD78F1824GKA2, μPD78F1825GKA2, μPD78F1836GKA2, μPD78F1837GKA2, μPD78F1838GKA2, μPD78F1839GKA2, μPD78F1840GKA2

A-Version Products

μPD78F1823AGKA, μPD78F1824AGKA, μPD78F1825AGKA, μPD78F1836AGKA, μPD78F1837AGKA, μPD78F1838AGKA, μPD78F1839AGKA, μPD78F1840AGKA μPD78F1823AGKA2, μPD78F1824AGKA2, μPD78F1825AGKA2, μPD78F1836AGKA2, μPD78F1837AGKA2, μPD78F1838AGKA2, μPD78F1839AGKA2, μPD78F1840AGKA2

78K0R/FG3

Non A-Version Products

μPD78F1841GCA, μPD78F1842GCA, μPD78F1843GCA, μPD78F1844GCA, μPD78F1845GCA, μPD78F1841GCA2, μPD78F1842GCA2, μPD78F1843GCA2, μPD78F1844GCA2, μPD78F1845GCA2

A-Version Products

μPD78F1841AGCA, μPD78F1842AGCA, μPD78F1843AGCA, μPD78F1844AGCA, μPD78F1845AGCA, μPD78F1841AGCA2, μPD78F1842AGCA2, μPD78F1843AGCA2, μPD78F1844AGCA2, μPD78F1845AGCA2

B) Table of Operating Precautions for 78K0R/Fx3

Table B-1 Summary of restrictions

No.	Outline	78K0R/Fx3				
		Non A-Version			A-Version	
	Rank Note	I (ES)	I (CS)	MP	I (ES)	CS & MP
1	UARTF LIN Automatic Baudrate Mode (Direction of use)	×	×	×	✓	✓
2	UARTF LIN Automatic Checksum Function (Direction of use)	×	×	×	×	×
3	Data Flash Read access during DMA Transfer (Direction of use)	×	×	×	×	×

✓: Not applicable
×: Applicable
- : Not checked

Note The rank is indicated by the letter appearing at the 5th position from the left in the lot number, marked on each product.

ES → Engineering Samples

CS → Commercial Samples

MP → Mass Production

C) Description of Operating Precautions for 78K0R/Fx3

Table C-1 No. C1 UARTF LIN Automatic Baudrate Mode (Direction of use)

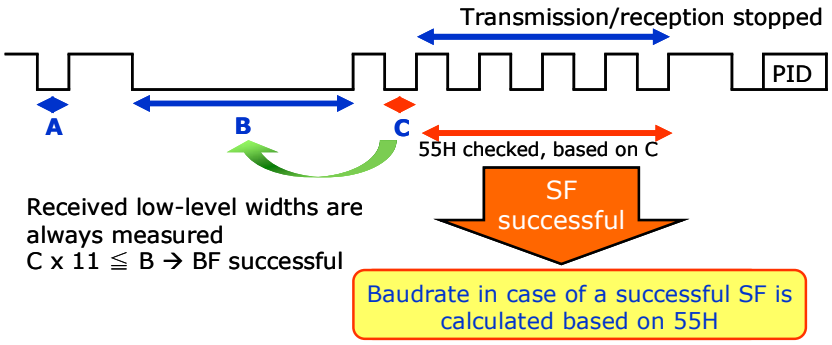
General

Using the Automatic Baud Rate Mode in LIN slave operation the Syncfield (SF) is automatically detected and checked. If the SF fails (no proper 0x55 received) the UARTF will abort the automatic baudrate sequence and wait for the next Syncbreakfield (SBF) to restart the sequence. If the duty cycle of the SF bits deviates from a nominal 50% duty cycle, the internal circuit may not detect the 0x55 pattern of the SF anymore and therefore would abort the automatic baudrate detection sequence. In case of an aborted baudrate detection sequence, the UARTF receive interrupt INTLRx after the PID is not generated.

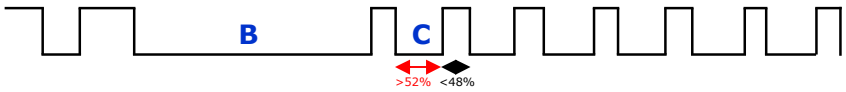
However, there is no issue if the duty cycles of the SF on LRxD0 (LRxD1) is in the range of:
48% ≤ SF duty ≤ 52%

Details

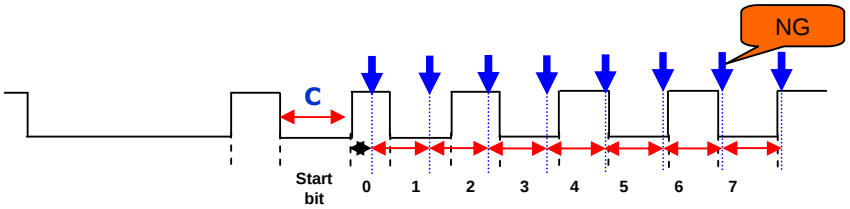
Mechanism of SF detection



However, the SF will be checked based on the low level width “C” of the Start-bit.



If the low level width of the SF will become >52%, the baudrate for the successful SF detection will be changed based on “C” and no valid SF (0x55) can be received anymore.
Due to this the UARTF does not generate the INTLRx and the slave will not respond.



Usecase:

The above described phenomenon with a duty cycle shift on LRxD0, LRxD1 pin could be caused by the LIN transceivers.
For example: LIN Master and LIN Slave are operating on different voltage levels.

Workaround

For the Non A-version products:

If baudrate detection of the SF is required, the countermeasure will be the user has to switch off the Automatic Baudrate Mode and perform the SF baudrate detection by means of software.

For the A-Version Products:

The Automatic Baud Rate Mode (UFnMD1, UFnMD0 = 11B) can be used as described in the User’s Manual. But, please take care for the item No. 2.

Table C-2 No. C2 UARTF LIN Automatic Checksum Function (Direction of use)

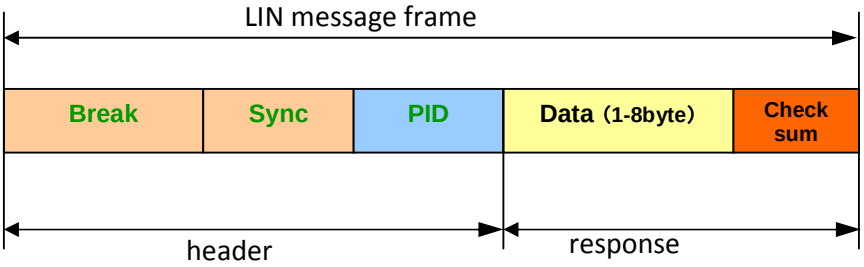
General

Using the UARTF in Automatic Baud Rate Mode (UFnMD1, UFnMD0 =11B) an Automatic Checksum Function (UFnACE = 1) is supported to calculate the checksum during response transmission or response reception automatically.

However, some messages received with a specific data pattern including an incorrect checksum will **not** be detected as **incorrect** by the UARTF in Automatic Checksum Function mode.

- The issue applies to both classic & enhanced (PID included) checksum
- It applies to response reception case
- It applies only to a very specific data / checksum pattern:

➔ When the checksum calculation result is 0x00 (0xFF before inversion), no carry (9th bit position) has been added to the LSB and the received checksum is ≠ 0x00



To get a better understanding for the phenomena, please refer to the below given data pattern examples for the enhanced checksum:

(1) Message received with correct checksum

(Transmitted by master) PID=0x06,
(Transmitted by master) DATA=0x2C,0x3A,0x93,0x00,0x00,0x00,0x00,0x00
(Transmitted by master) CKSM=0x00
Calculated CKSM = 0xFF + CKSM transmitted by master = 0xFF
➔ The message will be treated to be good

(2) Message received with incorrect checksum

(Transmitted by master) PID=0x06,
(Transmitted by master) DATA=0x2C,0x3A,0x93,0x00,0x00,0x00,0x00,0x00
(Transmitted by master) CKSM=0xFF (Not correct for this message)
Calculated CKSM = 0xFF + CKSM transmitted by master = 0x1FE
➔ An incorrect message will be treated to be good

Result: An incorrect checksum is not detected

Workaround
If Automatic Baud Rate mode is selected (UFnMD1, UFnMD0 =11B) clear the bit UFnACE = 0 to disable the Automatic Checksum Function and calculate the corresponding checksum by means of software.

However, the following two cases should be taken under consideration:

Response reception:
Checksum must be calculated by software from the data stored into the buffer and must be compared with the checksum obtained via communication (UFnACE = 0).

Response transmission:
Either, the checksum is calculated by software (UFnACE = 0), added to the end of the response transmission data and transmitted, or the checksum could also be calculated automatically by the hardware of UARTF (UFnACE = 1), **but** if this feature is used please take care for the following caution:
Caution:
In case the software will change back to “Response Reception” a reconfiguration of the UFnOPT1 register becomes necessary. (change UFnACE from “1” to “0”).
But before changing UFnACE, be sure to clear the bits UFnTXE and UFnRXE beforehand (UFnTXE = 0, UFnRXE. = 0).

Table C-3 No. C3 Data Flash Read access during DMA Transfer (Direction of use)

General

In case a Data Flash Read access will be performed exactly at the same timing while any DMA transfer is triggered, there is a possibility for an internal bus conflict between CPU bus and Data Flash bus. Such kind of bus conflict can cause a wrong data to be read from the Data Flash.

Workaround

First of all, the user must keep in mind that a Data Flash Read access can be done on two different manners:

Case 1:

Data Flash Read access directly executed in the user software.

- The workaround for the direct Data Flash Read access is described on the next page(s)

Case 2:

Data Flash Read access via the 'Data Flash Access Library' (FDL) and/or EEPROM Emulation Library (EEL). Both libraries are developed under the responsibility of Renesas.

Workaround for Case 2:

- The current FDL version (V1.0.2) will be updated to take the aforementioned phenomena under consideration.
- The current EEL version (V1.0.6) will be updated to take the aforementioned phenomena under consideration.

Please contact your local Renesas sales support team to get the target schedules for the updated versions.

Remark:

However, there will be **no** internal bus conflict as described above during any of the following Data Flash accesses:

- Data Flash Write access via FDL
- Verify command via FDL
- Blank Check command via FDL

In other words, these Data Flash access commands can be executed without taking care of any workaround.

Workaround for Case 1:
To prevent a DMA transfer during any of the below listed Data Flash Read instructions (see the list on the next two pages) the user has to suspend the DMA transfer for all DMA channels. To suspend or to release any DMA transfer can be controlled by the DMCALL register, in particular by the bit 'DWAITALL'.
In case, DWAITALL = 1 all DMA channels are forced to wait until DWAITALL becomes "0" again.

Remark:
To keep the DMA suspension as short as possible, we would recommend to perform the corresponding software modifications direct in Assembler language.

Software example before modification:

```
dataflash_read_asm:
     MOVW      HL, AX          // 1 cycle
     MOV       A, C            // 1 cycle
     MOV       ES, A           // 1 cycle
     MOV       A, ES:[HL]      // 4 cycles
     RET                       // 6 cycles
```

Software example after modification:

```
dataflash_read_asm:
     MOVW      HL, AX          // 1 cycle
     MOV       A, C            // 1 cycle
     MOV       ES, A           // 1 cycle
     PUSH      PSW             // 1 cycle  (Note 1)
     DI                        // 4 cycles (Note 2)
     SET1      DWAITALL        // 2 cycles (Note 3)
     NOP                       // 1 cycle  (Note 4, 5)
     NOP                       // 1 cycle  (Note 4, 5)
     MOV       A, ES:[HL]      // 4 cycles
     CLR1      DWAITALL        // 2 cycles (Note 3)
     POP       PSW             // 3 cycles (Note 6)
     RET                       // 6 cycles
```

Cautions ***1:*** Before any read access to the Data Flash will be executed, be sure to wait for minimum two CPU clocks after DWAITALL = 1.

2: Disable all interrupts (DI) before DWAITALL = 1 to prevent the execution of any interrupt service routine during the time when all DMA transfers are kept pending. In case an interrupt service would be executed during the time between DWAITALL = 1 until DWAITALL = 0 a pending DMA transfer would be maybe blocked for a relative long time until the interrupt service is finished.

3: For the above described workaround it is assumed the user application software is always using DWAITALL = 0. If this is not always true, please refer to Note 3

Notes ***1:*** Save the current PSW status, especially the current interrupt status (EI or DI).
2: Disable interrupt (DI) will be necessary to prevent any interrupt service to be executed during the time when any DMA transfer is kept pending. Please, refer to Caution 2 as well.
3: In case the user application software would also modify the bit 'DWAITALL' (e.g. DWAITALL = 1) the corresponding actions need to be taken (e.g. store the current DWAITALL status before SET1 DWAITALL and restore it after DWAITALL = 0).
4: Be sure to wait for minimum two CPU clocks before any Data Flash read instruction is executed after DWAITALL = 1.
For all possible Data Flash read instructions, please refer to the list below.
5: The two instructions 'NOP' could also be replaced by any other instruction, which **do not** perform a Read from the Data Flash (e.g. MOV A, C ;; MOV ES, A ;; …..). However, it is important to wait at least for two CPU clocks by these instructions after DWAITALL = 1.
6: Restore the PSW status to continue the application software with respect to the Interrupt status before any DMA transfer was kept pending (EI or DI)

List of possible Data Flash Read Instructions:

```
- MOV   A, ES:!addr16
- MOV   A, ES:[DE]
- MOV   A, ES:[DE + byte]
- MOV   A, ES:[HL]
- MOV   A, ES:[HL + byte]
- MOV   A, ES:[HL + B]
- MOV   A, ES:[HL + C]
- MOV   A, ES:word[B]
- MOV   A, ES:word[C]
- MOV   A, ES:word[BC]
- MOV   B, ES:!addr16
- MOV   C, ES:!addr16
- MOV   X, ES:!addr16

- MOVW  AX, ES:!addr16
- MOVW  AX, ES:[DE]
- MOVW  AX, ES:[DE + byte]
- MOVW  AX, ES:[HL]
- MOVW  AX, ES:[HL + byte]
- MOVW  AX, ES:word[B]
- MOVW  AX, ES:word[C]
- MOVW  AX, ES:word[BC]
- MOVW  BC, ES:!addr16
- MOVW  DE, ES:!addr16
- MOVW  HL, ES:!addr16

- ADD   A, ES:!addr16
- ADD   A, ES:[HL]
- ADD   A, ES:[HL + byte]
- ADD   A, ES:[HL + B]
- ADD   A, ES:[HL + C]

- ADDC  A, ES:!addr16
- ADDC  A, ES:[HL]
- ADDC  A, ES:[HL + byte]
- ADDC  A, ES:[HL + B]
- ADDC  A, ES:[HL + C]

- SUB   A, ES:!addr16
- SUB   A, ES:[HL]
- SUB   A, ES:[HL + byte]
- SUB   A, ES:[HL + B]
- SUB   A, ES:[HL + C]

- SUBC  A, ES:!addr16
- SUBC  A, ES:[HL]
- SUBC  A, ES:[HL + byte]
- SUBC  A, ES:[HL + B]
- SUBC  A, ES:[HL + C]

- AND   A, ES:!addr16
- AND   A, ES:[HL]
- AND   A, ES:[HL + byte]
- AND   A, ES:[HL + B]
- AND   A, ES:[HL + C]




- OR    A, ES:!addr16
- OR    A, ES:[HL]
- OR    A, ES:[HL + byte]
- OR    A, ES:[HL + B]
- OR    A, ES:[HL + C]
```

List of possible Data Flash Read Instructions: (cont'd)

```
- XOR   A, ES:!addr16
- XOR   A, ES:[HL]
- XOR   A, ES:[HL + byte]
- XOR   A, ES:[HL + B]
- XOR   A, ES:[HL + C]

- CMP   A, ES:!addr16
- CMP   A, ES:[HL]
- CMP   A, ES:[HL + byte]
- CMP   A, ES:[HL + B]
- CMP   A, ES:[HL + C]
- CMP   ES:!addr16,#byte

- CMPS  ES:!addr16
- CMPS  X, ES:[HL + byte]

- ADDW  AX, ES:!addr16
- ADDW  AX, ES:[HL + byte]

- SUBW  AX, ES:!addr16
- SUBW  AX, ES:[HL + byte]

- CMPW  AX, ES:!addr16
- CMPW  AX, ES:[HL + byte]

- MOV1  CY, ES:[HL].bit
- AND1  CY, ES:[HL].bit
- OR1   CY, ES:[HL].bit
- XOR1  CY, ES:[HL].bit

- BT    ES:[HL].bit, $addr20
- BF    ES:[HL].bit, $addr20
```

D) Valid Specification

Item	Date published	Document No.	Document Title
1	July 2011 or later	R01UH0007EJ0500 or later	User's Manual: Hardware

E) Revision History

Item	Date published	Document No.	Comment
1	August 04, 2010	R01TU0003ED0100	1st Release
2	July 15, 2011	R01TU0003ED0101	1st Update: -) Added the A-Version Products -) For No. 1: Added a comment to the 'Workaround' for the A-version -) Added item No. 2 for all products
3	November 10 , 2011	R01TU0003ED0102	2nd Update: -) Added item No. 3 for all products
4	November 21, 2011	R01TU0003ED0103	3rd Update: -) Modified and improved the workaround description for No. 3 -) Added Remark on page 11

Renesas Electronics Europe

R01TU0003ED0103
November 2011