To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

Send any inquiries to http://www.renesas.com/inquiry.

RENESAS

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.

M16C/62P

I²C Master Interface Driver using DMA and ACK/NACK Interrupts

Introduction

This application note provides an example of how to use the M16C family I²C mode provided in the UART peripheral. A master implementation using the on chip DMA and ACK/NACK interrupts is shown interfacing to an external I²C EEPROM.

The hardware configuration consisted of a M30627FJPFP (M16C/62P with 512KFLASH/31KRAM/100pin) and a HN58X2404SI (RENESAS 4K I²C EEPROM with 512-word x 8-bit) connected via the M16C UART in I²C mode with external 1Kohm pull-up resistors on the SDA and SCL lines.

The source code has been tested with various M16C devices, UARTs, and compilers (both IAR and RENESAS NC30) with speeds up to 400Kb/s.

The program described in this example uses the I2C library to write and check-out correct writing/reading of a string of characters in the external EEPROM.

The string used is: “Thank you for using RENESAS M16C family I²C (this code uses ACK/NACK interrupt routines)”.

The number of message writes and errors (normally none) are monitored in 2 variables for reliability test purposes.

All tested conditions are validated only if no error was generated during the test. The source code and complete project files is available to download free of charge from our Website.

As M16C and M32C devices share most peripherals, it could be of interest to read following application note “Using Simple I2C Bus Mode on M32C/83.85” (www.renesas.com M16C application note REJ05B0145-0100Z).

Contents

Possible configurations.

The following table shows the possible options supported by current software version.

	M16C26	M16C26A	M16C28	M16C29	M16C6N4	M16C62P
UART0	Not applicable	Not applicable	Not applicable	Not applicable	**Tested**	**Tested**
UART1	Not applicable	Not applicable	Not applicable	Not applicable	*Not tested*	*Not tested*
UART2	**Tested**	*Not tested*	*Not tested*	*Not tested*	**Tested**	**Tested**
DMA1	**Tested**	*Not tested*	*Not tested*	*Not tested*	**Tested**	**Tested**

The options which have been tested were confirmed to work correctly with no errors detected during the test (the other ones should work as well but haven't been tested).

Usage.

1. Abstract of CPU.h

The "cpu.h" file allows choosing the configuration options; this file needs to be modified in order to select the correct configuration setting.

DMA1 used (or not), which UART is used (when this option is available for selected device), which device is used and which compiler (Renesas NC30 or IAR).

```
// -------------------------------------------------------------------------
// DMA use (comment this line if DMA1 is not used for transmission)
// -------------------------------------------------------------------------
#define DMA1_USED
// -------------------------------------------------------------------------
// UART selection (please uncomment one of here bellow only)
// -------------------------------------------------------------------------
//#define UART0
//#define UART1     // NOT TESTED ON THIS UART YET !!!!!!
#define UART2
// -------------------------------------------------------------------------
// CPU selection (please uncomment one of here bellow only)
// -------------------------------------------------------------------------
//#define M16C26
//#define M16C26A
//#define M16C28
#define M16C62P
//#define M16C6N4

//#define EVA_3DK62P
#ifdef  DMA1_USED
        #ifdef  UART0
        #define DMA1_ACK_TRIG_SOURCE   0x0B;   // Trig by UART0 ACK
        #endif
        #ifdef  UART1
          #define DMA1_ACK_TRIG_SOURCE 0x0F;   // Trig by UART1 ACK
        #endif
        #ifdef  UART2
        #define DMA1_ACK_TRIG_SOURCE   0x0D;   // Trig by UART2 ACK
  #endif
#endif
```

2. Detailed description.

The I²C protocol uses following main states for communication:

- START bit.
- Normal communication (8-bit synchronous).
- ACK or NACK bits
- STOP bit.

The M16C devices offer 2 options to implement the communication:

a) ACK/NACK interrupts (used in this example).

b) UART transmit/receive interrupts (used in M32C example mentioned above).

The first option (a) is more suitable in master mode as the master only needs to monitor the ACK/NACK signal from the slave. For slave mode the second option (b) is more appropriate as the slave needs to generate itself the ACK and NACK signals upon reception of data and or address.

Below figure shows the timing and process being used in this application note.

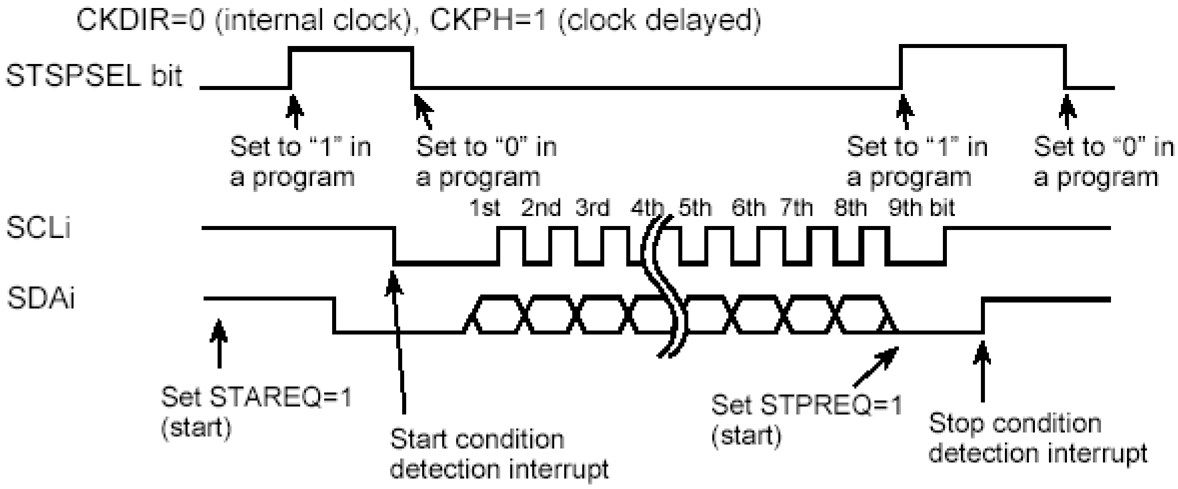


Figure 1

The register UiSMR4 allows automatic generating the START and STOP signals (as well as RESTART which is not used here). This register has only been introduced in the recent M16C devices (the ones mentioned in above tables) and offers the benefit of avoiding direct port toggling to generate these conditions.

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
U0SMR4 to U2SMR4	036Ch, 0370h, 0374h	00h

Bit	Bit Symbol	Bit Name	Function	RW
b0	STAREQ	Start Condition Generate Bit [1]	0 : Clear 1 : Start	RW
b1	RSTAREQ	Restart Condition Generate Bit [1]	0 : Clear 1 : Start	RW
b2	STPREQ	Stop Condition Generate Bit [1]	0 : Clear 1 : Start	RW
b3	STSPSEL	SCL,SDA Output Select Bit	0 : Start and stop conditions not output 1 : Start and stop conditions output	RW
b4	ACKD	ACK Data Bit	0 : ACK 1 : NACK	RW
b5	ACKC	ACK Data Output Enable Bit	0 : Serial I/O data output 1 : ACK data output	RW
b6	SCLHI	SCL Output Stop Enable Bit	0 : Disabled 1 : Enabled	RW
b7	SWC9	SCL Wait Bit 3	0 : SCL "L" hold disabled 1 : SCL "L" hold enabled	RW

3. Start condition generation

A high-to-low transition of the SDA with SCL high is generated in order to start read, write operation.

Bellow 2 lines will force the device to generate the START condition:

UiSMR4 = 0x71; // Start condition generate (STSPSEL=0/STAREQ=1)

UiSMR4 = 0x09; // STSP output enable (STSPSEL=1/STAREQ=1)

After UiSMR4 has been set as mentioned the device generates the START signal which automatically generates a START/STOP interrupt request.

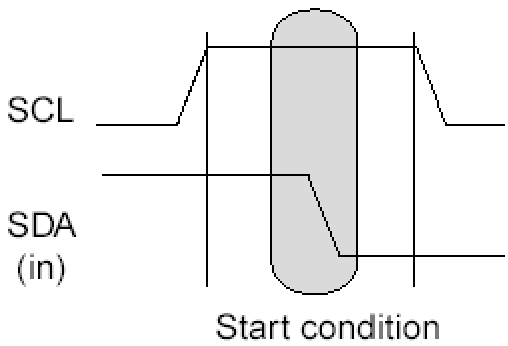


Figure 2

The first action to be done in the START interrupt routine section is to clear the START signal request by setting new value in UiSMR4 register:

UiSMR4 = 0x00; // clear : STSPSEL=0

4. Stop Condition generation

A low-to-high of the SDA with the SCL high is a stop condition. In the case of write operation, a stop condition terminates the write data input and places the device in an internally-timed write cycle to the memories.

Similarly to START condition, the STOP condition is initiated by configuring the UiSMR4 register:

UiSMR4 = 0x74;   // SCL=SDA="H"

UiSMR4 = 0x0C;   // Stop condition (STSPSEL=1/STPREQ=1)

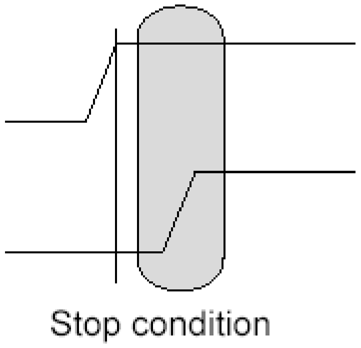


Figure 3

After UiSMR4 has been set as mentioned the device generates the STOP signal which automatically generates a START/STOP interrupt cause.

The first action to be done in the STOP interrupt routine section is to clear the STOP signal request by setting new value in UiSMR4 register:

UiSMR4 = 0x30;   // ACK data output "H"

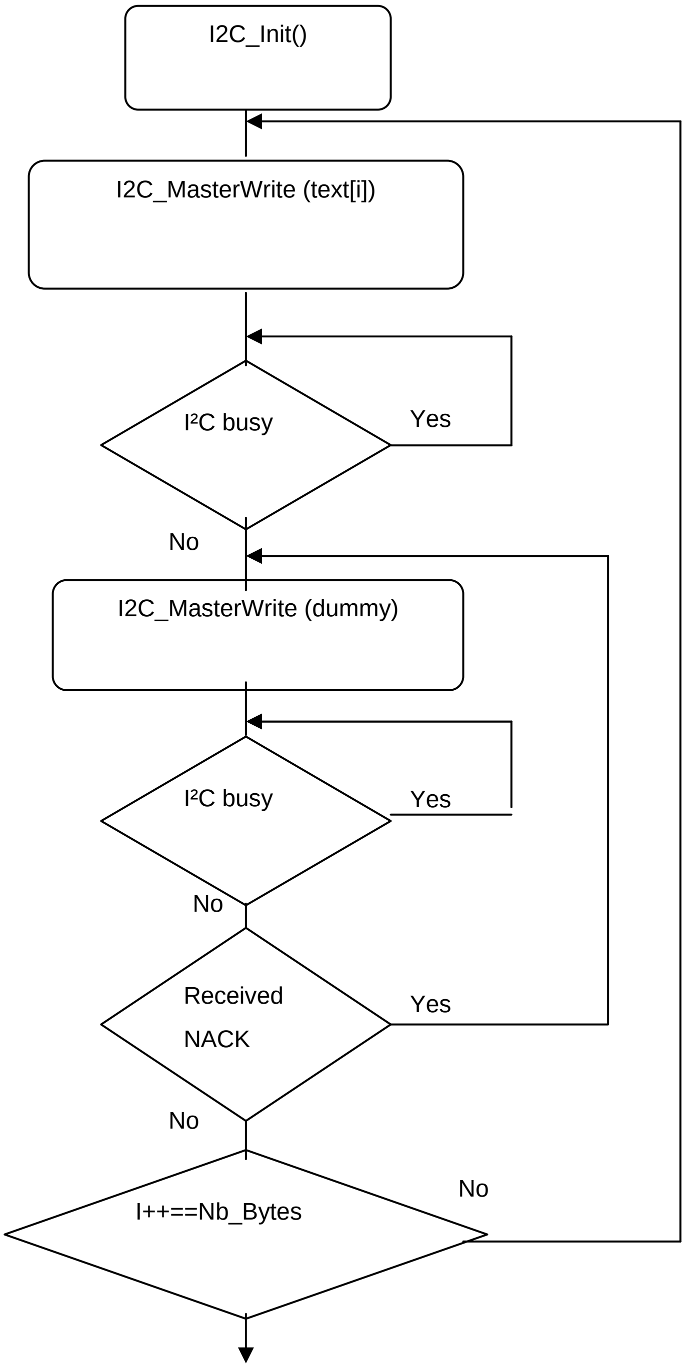
5. Interrupt causes

Below interrupt vectors are used in order to process data transmission and reception.

Interrupt function name	result
UARTi Bus Collision Detect	Start condition detection or stop condition detection
NACK interrupt (shared with Transmit Interrupt)	No acknowledgment detection (NACK) Rising edge of SCLi 9th bit
ACK interrupt (shared with Receive Interrupt)	Acknowledgment detection (ACK) Rising edge of SCLi 9th bit

Main program flowchart.

1. **Main routine (write portion)**

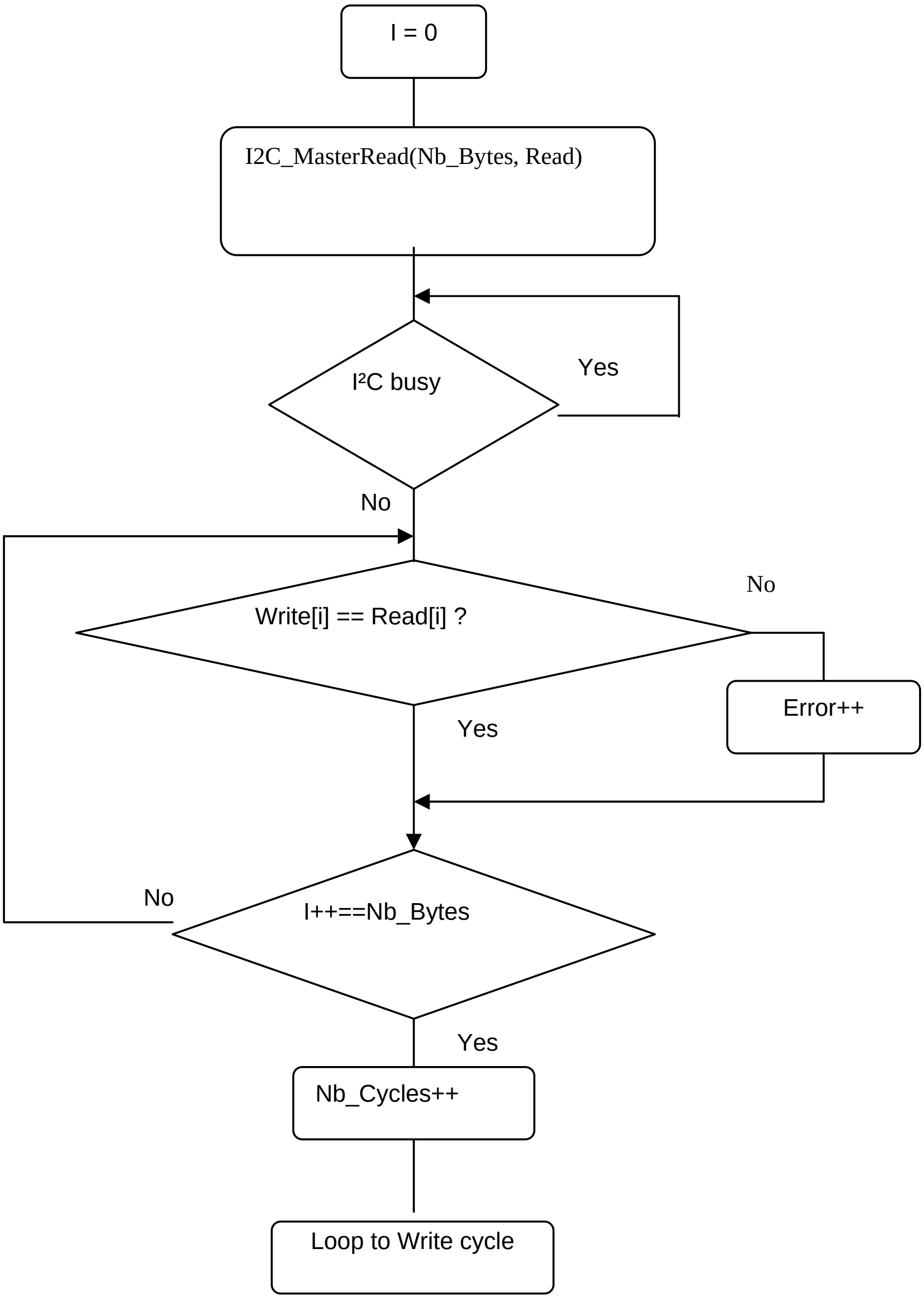


Write data to EEPROM

Test if EEPROM has finished writing

Loop until EEPROM is not busy

2. Main routine (verify portion)



Functions description.

Only 3 functions are currently accessible from application software, these are I2C_init, I2C_MasterWrite and I2C_MasterRead. The other functions described below shouldn't be called by the application software (most of them are interrupt functions). The pins used for I²C interface should be kept in input mode.

1. **void I2C_init (void)**

This function is to be called in the processor initialization section. It prepares the UART settings to work in I²C mode.

2. **I2C_MasterWrite(address, *ptrData, length)**

This function will start a write process to slave peripheral. Input parameters are device address, pointer to data to be written (ptrData) and number of bytes to be written (length). Inside the function a global bit variable "busy" (CurrentStatus.bit.busy) is used to monitor I²C bus status to avoid accessing the bus while a cycle is not completed. When a new cycle is started this boolean is set true, at the end of the cycle the driver code clears this Boolean to false to allow a new cycle to start (see below I2C_StartStopDetection function) .

If a new request comes while the "busy" bit is set, the driver returns an error and nothing is done.

If "busy" Boolean is false it is turned true and driver global variables are initialized with passed parameters (these will be shared and modified by interrupt functions).

3. **I2C_MasterRead(address, *ptrData, length)**

This function will start a read process from slave peripheral. Input parameters are device address, pointer to buffer where data is to be stored (ptrData) and number of bytes to be read (length). In the function a global bit variable "busy" (CurrentStatus.bit.busy) is used to monitor I²C bus status to avoid accessing the bus while a cycle is not completed. The function of the "busy" bit is same as in the I2C_Master function.

4. **_INTERRUPT void I2C_StartStopDetection(void)**

This function is called when a stop or start condition interrupt is generated. The BBS bit in the UiSMR sfr register indicates whether it is the start or stop condition ("1" indicates start and "0" indicates stop condition).

When start condition is the interrupt cause this routine will send the I²C slave address (including R/W bit) with 9^{th} bit set to 1 (to allow slave to generate ACK or NACK) and enable the NACK and ACK interrupts (or DMA instead of ACK interrupt in case it is used to automatically send the data).

When a stop condition is detected both NACK, ACK interrupts are disabled and "busy" bit is cleared to allow later access to the bus for other requests.

5. **_INTERRUPT void I2C_ACK(void)**

This function is called when ACK condition is detected. It is enabled in I²C read or I²C write modes if the DMA is not used or only in I²C read mode if DMA is used.

In I²C write mode this routine will write data buffer contents to UARTi transmit buffer with automatic increment of the data buffer index.

In I²C read mode it stores contents of UARTi receive buffer in the data buffer (with automatic increment of buffer index) and it will put 0x00FF in the UARTi transmit register to acknowledge received byte (9^{th} bit to 0 means ACK signal is sent). The last byte transmitted should be 0x01FF so that slave can put the ACK level (or NACK in case of problem) for acknowledgement of the transmission.

6. **_INTERRUPT void I2C_NACK(void)**

This function allows stopping the on-going I²C transmission by disabling ACK interrupt or DMA.

7. **_INTERRUPT void DMA_interrupt(void)**

Stops DMA function and enables ACK interrupt (to accept last byte).

Conclusion

M16C devices offer a powerful and easy way of handling I²C bus to slave devices without consuming CPU power doing port toggling. This application note is made available to help M16C developers speed up their development.

Website and Support

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

Inquiries
http://www.renesas.com/inquiry
csc@renesas.com

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, 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 products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life
   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas 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 applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.