It's RFID

Firmware Manual

3ALogics Evaluation System Series

This document contains 'firmware configuration and operation' information offered with Evaluation system board. Other hardware and protocol information are not included in this document.

http://www.3ALogics.com

Revision history

Date	Version	Content
2008. 04. 11	0.1	Preliminary release
2008. 04. 15	1.0	1.0 version release

Notice : All referenced brands, product names, service name and trademarks are the property of their respective owners.

*Any*Read™ - is a trademark of 3ALogics.

Disclaimer

The information provided is as is without any type of warranty. To the maximum extent permitted by law, 3ALogics accepts no responsibility on all warranties, including without limitation any implied warranties of merchant ability, fitness for a particular purpose, and non-infringement. The entire risk from the use or performance of the product and documentation remains with the recipient. In no event shall 3ALogics or its suppliers be liable for any consequential, incidental, direct, indirect, special, punitive or other damages whatsoever including without limitation damages for loss of business, profits, business interruption, loss of business information or other losses.

Contact

3Alogics Inc.
7th Fl., Hyundai-office Bldg., 9-4, Sunae-dong, Bundang-gu, Seongnam-si, Gyeonggi-do, 463-783 Korea

TEL : (82)-(31)-715-7117
FAX : (82)-(31)-719-7551

E-mail : rfid@3ALogics.com

homepage: http://www.3ALogics.com

Printed in the Republic of Korea..

Document Contents

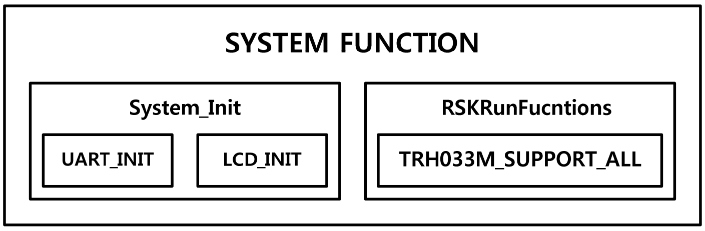
Chapter1 General Information

1.1 OUTLINE

Explanation of Firmware structure and function are given by each role.

1.2 SYSTEM FUNCTION

System function is basic function to operate RSK (RFID Study Kit).

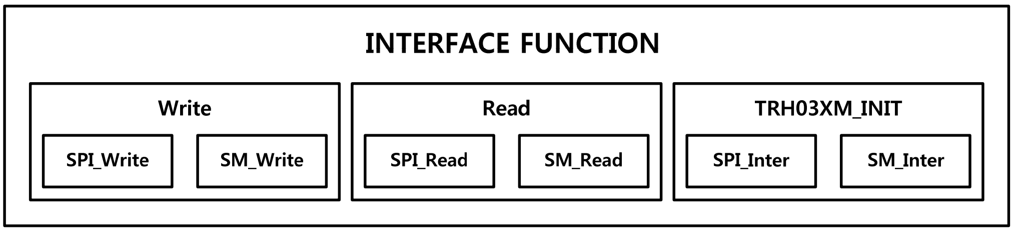


Picture 1-1 System Function Structure

System function is composed as Picture 1-1. UART and LCDFUNCTION are not main functions of Reader IC, thus, not included in this manual. Using RSKRunFuntions() can change functions using RSK board SDK. Basically it's a function to select and activate.

1.3 INTERFACE FUNCTION STRUCTURE

Interface function performs data communication between microprocessor and TRH03XM. For Firmware development it is the most basic function. Picture 1-2 displays a structure of interface function used in RSK (RFID Study Kit).



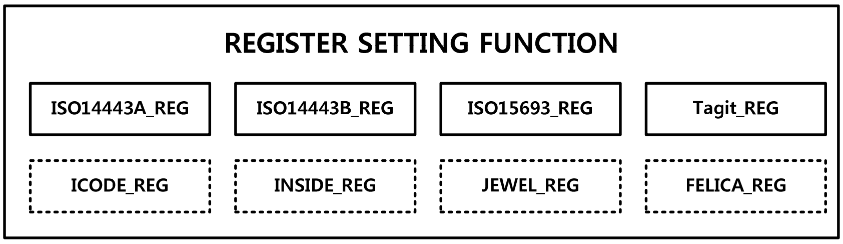
Picture 1-2 Interface Function Structure

Interface function is separated into Read Function, Write Function and Initialization Function. Read and Write Functions are, as indicated, functions to read and write data. Initialization Function sets initialization value of each interface signals.

Read/Write/Initialization functions can be selected from SPI mode and Separated Multiplexed mode. Interface mode is selected by INTER_SEL variable.

1.4 REGISTER SETTING FUNCTION TYPE

Register Setting Function sets register value by each protocol. Therefore, different function is used for different protocol. Picture 1-3 displays Register Setting Function type.



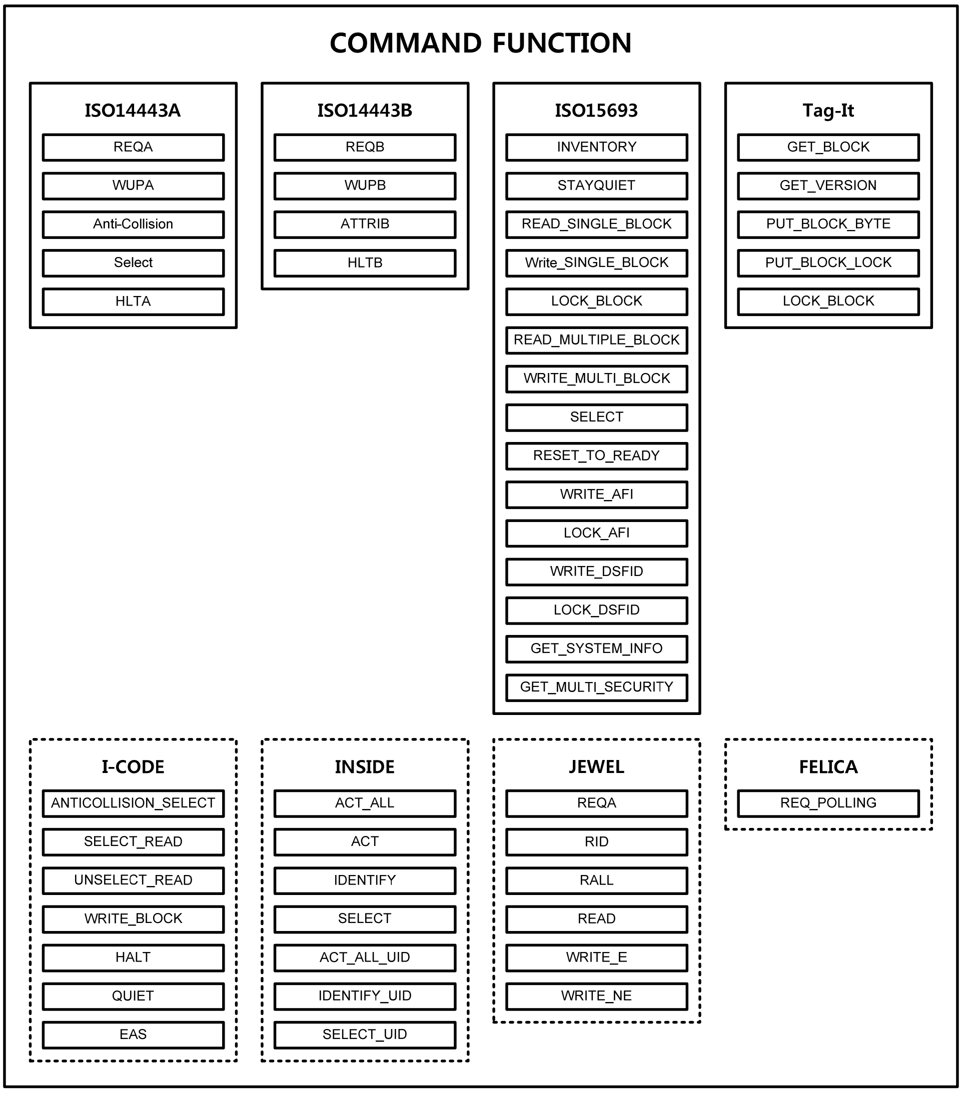
Picture 1-3 Register Setting Function Type

TRH033M supports all above functions, but TRH031M only supports top line functions (solid line) which are protocols TRH031M supports. Since Register Setting Function sets basic communication operation such as encoding/decoding and modulation/demodulation, it must be activated before any communication operation.

1.5 COMMAND FUNCTION STRUCTURE AND TYPE

Command Function performs each protocol command stated in standard document. TRH033M supports 8 protocols and each protocol has about 10 different commands, thus, various Command Function types exist. Also these Command Functions have very complex structure since operation order impact different command actions. Thus, this document will not cover all command included in RSK firmware, but some sample cases of Command Functions will help user to understand how firmware is operated in RSK board.

Picture 1-4 displays basic Command Function by protocol.



COMMAND FUNCTION

ISO14443A
- REQA
- WUPA
- Anti-Collision
- Select
- HLTA

ISO14443B
- REQB
- WUPB
- ATTRIB
- HLTB

ISO15693
- INVENTORY
- STAYQUIET
- READ_SINGLE_BLOCK
- Write_SINGLE_BLOCK
- LOCK_BLOCK
- READ_MULTIPLE_BLOCK
- WRITE_MULTI_BLOCK
- SELECT
- RESET_TO_READY
- WRITE_AFI
- LOCK_AFI
- WRITE_DSFID
- LOCK_DSFID
- GET_SYSTEM_INFO
- GET_MULTI_SECURITY

Tag-It
- GET_BLOCK
- GET_VERSION
- PUT_BLOCK_BYTE
- PUT_BLOCK_LOCK
- LOCK_BLOCK

I-CODE
- ANTICOLLISION_SELECT
- SELECT_READ
- UNSELECT_READ
- WRITE_BLOCK
- HALT
- QUIET
- EAS

INSIDE
- ACT_ALL
- ACT
- IDENTIFY
- SELECT
- ACT_ALL_UID
- IDENTIFY_UID
- SELECT_UID

JEWEL
- REQA
- RID
- RALL
- READ
- WRITE_E
- WRITE_NE

FELICA
- REQ_POLLING

Picture 1-4 Command Function Type

Remark: Non-ISO standard command may change or updated.

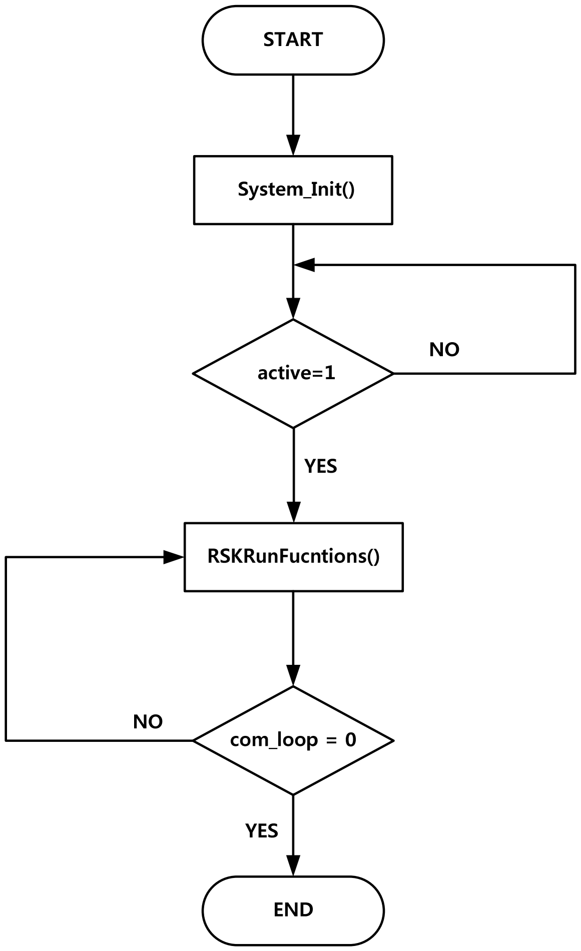
Chapter2 SYSTEM FUNCTION

2.1 OUTLINE

System initialization function and main function is explained.

2.2 main() FUNCTION

Picture2-1 displays RSK (RFID Study Kit) Firmware, main() function, Flow Chart.



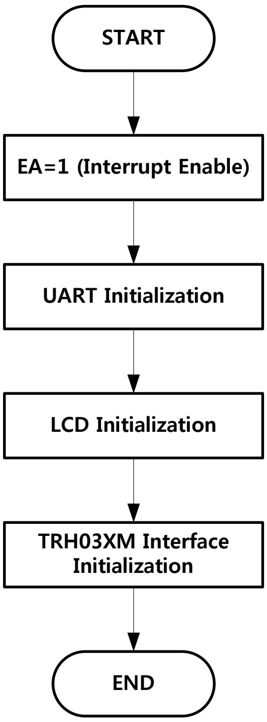
Picture2-1 main() Function Flow Chart

When Firmware is activated, first System_Init() function is executed. System_Init() function initialize hardware used in RSK (RFID Study Kit). LCD, UART, TRH03XM interface, etc are initialized.

When active variable is 1, RSKRunFunctions() is executed. When active variable is 0 than no action is taken. When com_loop variable is 1, continuous execution of a command within RSKRunFunctions() occurs. When com_loop variable is 0, RSKRunFunctions() executes once and stop.

2.3 System_Init() FUNCTION

System_Init() function initialize RSK (RFID Study Kit) hardware. It initializes UART communication, TRH03XM interface and LCD. Picture 2-2 displays Flow Chart of System_Init() function.



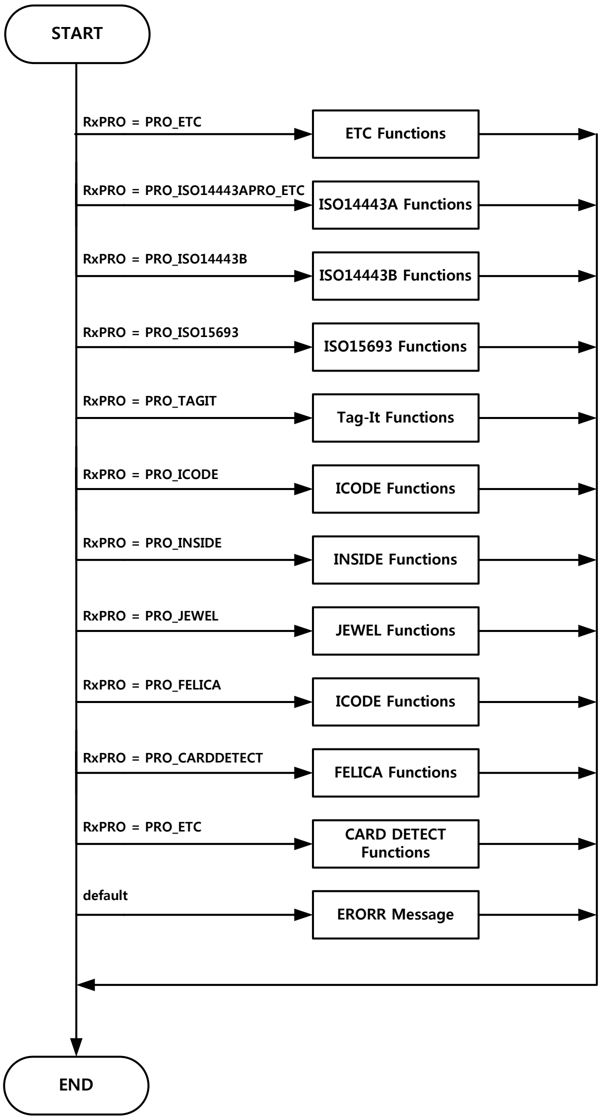
Picture 2-2 System_Init() Function Flow Chart

EA is 8051 microprocessor register that should be set as 1 when using as interrupt. Initial UART setting is communication Baud rate 9600bps, data 8bit, and Stop bit 1. Parity bit is not used.

LCD initialization is hardware setting process prior to RSK board LCD activation. TRH03XM interface is most critical for system initialization. Microprocessor and TRH03xM communication mode are established during this process.

2.4 RSKRunFunction() FUNCTION

RSKRunFunctions() is a function that performs commands from host PC. Picture 2-3 displays RSKRunFunction() function Flow Chart.



Picture 2-3 RSKRunFunction() Function Flow Chart

This function can execute all protocols TRH03XM supports or each selected protocol can be executed. Individually command can be executed selectively.

Executable functions are ISO/IEC 14443A, ISO/IEC 14443B, ISO/IEC 15693, Tag-It, I-CODE, Inside, Jewel, Felica, CARDDETECT, and other miscellaneous functions. Executable function is determined by RxPRO variable value.

Chapter3 INTERFACE FUNCTION

3.1 OUTLINE

This chapter explains about interface function for communication between microprocessor and TRH03XM.

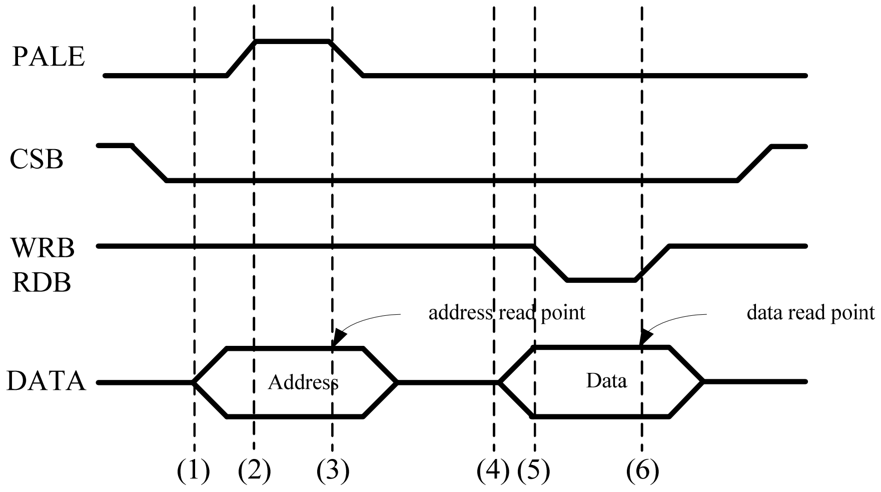
3.2 PARALLEL INTERFACE FUNCTION

3.2.1 SM_Inter() Function

SM_Inter() function is used for setting signal initialization value using parallel interface. Reset automatically executes since RST pin changes from 1 to 0. After Reset, clock signal oscillation obtains delay time for stabilization. It's about 500us. Write command is Dummy command for interface initialization. Thus, actual command is not executed. TRH03XM supports total of 4 types of parallel interface, and Separated Multiplexed mode is used in this circumstances. For detailed information about parallel interface, please refer to "TRH03XM DataSheet" "3.2 PARALLEL INTERFACE".

3.2.2 SM_Read/Write() Function

Read/Write function is very basic and most important function. Even if other higher level functions are well designed, if this function is not providing good performance other functions become ineffective. Also it controls input signal directly it is affected by hardware characteristics. TRH03XM interface communication speed is enough for communicating with normal 8 bit microprocessor. However, for high speed microprocessor, signal to signal timing should be considered for interface functions. Especially for high speed communication, environment characteristics such as temperature and noise can affect performance.



Picture 3-1 Separated Multiplexed Write/Read Function

Picture 3-1 displays each signal using Separated Multiplexed mode. Reviewing RSK firmware, it is identical to above Picture. First (1) address input, (2) PALE to High, (3) return to Low Value. (4) Data entered, (5) and (6) WRB or RDB changes Low to High. Basically SM_Write/Read() function is a function to create waveform as above picture. Entering 0xFF to data bus is to convert bi-directional port to input port.

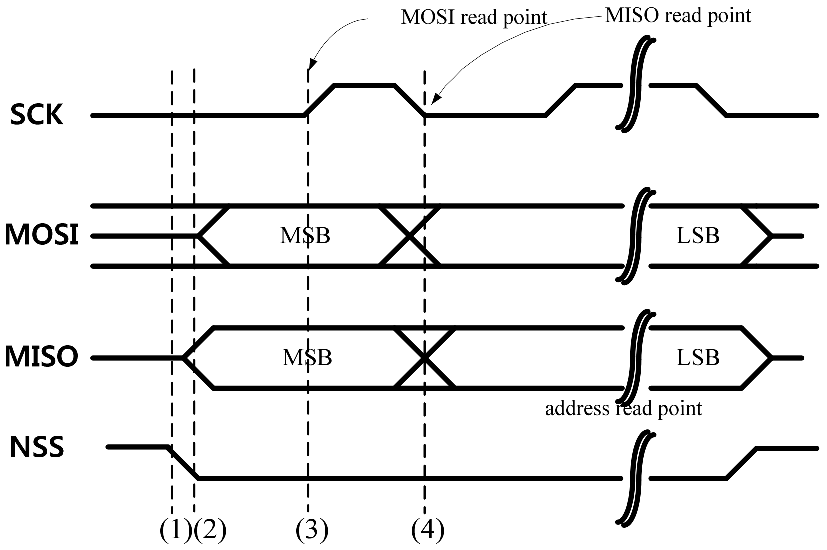
3.3 SPI SERIAL INTERFACE

3.3.1 SPI_Inter() Function

SPI_Inter() function is a command to initialize SPI interface. For SPI_Inter() function also reset is executed and 500us delay time is given. For SPI interface there is no Dummy command so Write command is executed, however, in command register IDLE(0x00) command is written, thus, command is not really executed. For detailed information about SPI interface, please refer to "TRH03XM DataSheet" "3.4 SPI SERIAL INTERFACE".

3.3.2 SPI_Read/Write() Function

SPI_Read/Write() function, as same as SM_Read/Write() function, creates SPI communication signal waveform as displayed in Picture 3-2 below.



Picture 3-2 SPI Interface Function

(1) NSS to Low point in time.   At this time MSB of MISO output occurs.   After (2) point in time, MOSI output, (3) Slave reads MOSI.   (4) MISO changes to next data.   SPI communication data is sent by one bit, to send all address and data, above operation must repeat 16 times.

Chapter4 REGISTER SETTING FUNCTION

4.1 OUTLINE

TRH03XM register setting function by protocol is explained. Explanation for non ISO standard protocol is omitted.

4.2 ISO14443A_REG() FUNCTION

ISO14443A_REG() function is for use of ISO/IEC 14443A standard protocol. Chart 4-1 describes analog setting of ISO14444A_REG() functions.

Chart 4-1 ISO14443A_REG() Function Analog Setting

Register Name	Flag	Value	Description
TXCONTROL	ModulatorSource	10b	Modulate signal generated from Internal Coder
	F100ASK	1b	Use 100% ASK modulation
	TX2Inv	1b	Reverse 180˚ to TX1 the carrier signal output from TX2
	TX2Cw	0b	Modulates TX2 pin and TX1 pin output signal at the same time
	TX2RFEn	1b	Use TX2 pin
	TX1RFEn	1b	Use TX1 pin
CWCONDUCTANCE	CwConducatance	3Fh	Use maximum output
RXCONTROL1	VGAGain	3h	Set RX VGA Gain to 18.06dB
RXTHRESHOLD	CRO	0b	Not change standard voltage in comparator
	HYR <2:0>	110b	Set comparator Hysteresis Range to 90mv
	CRV <2:0>	000b	Not change standard voltage in comparator

Since ISO/IEC 14443A use 100% ASK, *F100ASK* should be set to 1. *TX2Inv* is set to 1 to reverse TX2 phase. *TX2RFEn* and *TX1RFEn* both should be set to 1 and two TX are used. *VGAGain* value can be adjusted from 00h to 05h but 03h is recommended. Depending on Receiver input strength VGA gain can be adjusted, but in RFID system fixed gain is used instead of automatic gain control. High gain can be used for board or system that is not impacted much by noise but not recommended.

RSK firmware use 03h for *VGAGain.* However, timing is not a problem and reading distance is important, *VGAGain* should be changed to 00h~05h in Firmware and data is received for maximum effect.

Below chart 4-2 displays ISO14443A_REG() function digital setting. For analog setting, board and antenna environment may impact changes in value but in digital setting only special circumstance will impact in value change. CONCONTROL(0x14) is a register to select output data format and to use ISO/IEC 14443A, 0x01 is set.

ADcdMd flag sets decoding method of ISO/IEC 14443A and should be set to 1 for better performance. *RxWait* should be set approximately to 57us.

Chart 4-2 ISO14444A_REG() Function Digital Setting

Register Name	Flag	Value	Description
CODCONTROL	TxCodHigh	000b	Not in use
	TI_Addr	0b	Not in use
	TxCoding	001b	Set Encoding method for ISO 14443A Type
RXCONTROL2	ADcdMd	1b	Improve ISO 14443A decoding method
	Dcdsrc	0b	Decoding of RX received signal
RXWAIT	RxWait	06h	Set receiving delay time to 128/fc X 6 = 57us

4.3 ISO14443B_REG() FUNCTION

ISO14443B_REG() function sets standard setting for ISO/IEC 14443B protocol. Chart 4-3 displays analog setting for ISO14443B_REG().

ISO/IEC 14443B setting is quite similar to ISO/IEC 14443A. However, ISO/IEC 14443A uses ASK 100% but ISO/IEC 14443B uses 10% ASK. Thus, *F100ASK* should be set to 0 and *ModConducatance* value should be set to 10h. For detailed information refer to "TRH03XM CookBook" "9.2 Modulator index adjustment".

Chart 4-3 ISO14443B_REG() Function Analog Setting

Register Name	Flag	Value	Description
TXCONTROL	ModulatorSource	10b	Modulate signal generated from Internal Coder
	F100ASK	0b	ASK modulation depth depends on ModConductance value
	TX2Inv	1b	Reverse 180˚ to TX1 the carrier signal output from TX2
	TX2Cw	0b	Modulates TX2 pin and TX1 pin output signal at the same time
	TX2RFEn	1b	Use TX2 pin
	TX1RFEn	1b	Use TX1 pin
MODCONDUCTANCE	ModConductance	10h	Set modulation conductance value

Below chart 4-4 displays digital setting functions of ISO14443B_REG(). Main difference with ISO/IEC 14443A and ISO/IEC 14443B is BFRAMING register setting. BFRAMING register sets ISO/IEC 14443B protocol signal width. As below set EOF width to 11ETU, EGT width to 38us, and SOF to 11ETU Low 3ETU High.

For ISO/IEC 14443B CRC is used for all command response. Therefore, *TxCRCEn* and *RxCRCEn* setting should be 1. Also *CRCB* should be set to 1 and CRC operation should correspond to ISO/IEC 14443B setting. ISO/IEC 14443B do not use Parity Check method.

Chart 4-4 ISO14443B_REG() Function Digital Setting

Register Name	Flag	Value	Description
CODCONTROL	TxCodHigh	000b	Not in use
	TI_Addr	0b	Not in use
	TxCoding	000b	Set Encoding method to ISO 14443B Type
BFRAMING	EOFWidth	1b	Set EOF width to 11ETU
	CharSpacing	001b	EGT width to 128/fc X 4 = 38us
	SOFWidth	11b	SOF width to 11ETU Low 3 ETU High
REDUNDANCY	CRCWr	0b	Received CRC value not stored in FIFO buffer
	CRCB	1b	CRC operation method set to ISO/IEC 3309
	RxCRCEn	1b	Perform received data CRC operation
	TxCRCEn	1b	Add CRC operated data to transmit data
	ParityOdd	0b	Parity operation method not in use
	ParityEn	0b	Parity operation method not in use

4.4 ISO15693_REG() FUNCTION

ISO15693_REG() sets functions of ISO/IEC 15693 standard protocol. Chart 4-5 displays ISO15693_REG() analog setting.

ISO/IEC 15693 analog setting value is same as ISO/IEC 14443A. ISO/IEC 15693 also uses 100% ASK modulation. Other setting is same as ISO/IEC 14443A.

Chart 4-5 ISO15693_REG() Function Analog Setting

Register Name	Flag	Value	Description
TXCONTROL	ModulatorSource	10b	Modulate signal generated from Internal Coder
	F100ASK	1b	Use 100% ASK modulation
	TX2Inv	1b	Reverse 180˚ to TX1 the carrier signal output from TX2
	TX2Cw	0b	Modulates TX2 pin and TX1 pin output signal at the same time
	TX2RFEn	1b	Use TX2 pin
	TX1RFEn	1b	Use TX1 pin

Below Chart 4-6 is ISO15693_REG() digital setting functions. First set CODCONTROL register to ISO/IEC 15693 protocol. From DECODCONTROL register set *15693R_FS* to 0 and use Slow mode. This feature is only supported in TRH033M. REDUNDANCY register is same as ISO/IEC 14443B.

Chart 4-6 ISO15693_REG() Function Digital Setting

Register Name	Flag	Value	Description
CODCONTROL	TxCodHigh	000b	Not in use
	TI_Addr	0b	Not in use
	TxCoding	111b	Set Encoding method to ISO 15693 standard mode (1 out of 4 coding)
DECODCONTROL	15693R_FS	0b	Use ISO/IEC 15693 Slow mode
	ZrAfColl	0b	Decode received data even after Collision
REDUNDANCY	CRCWr	0b	Received CRC value not stored in FIFO buffer
	CRCB	1b	CRC operation method set to ISO/IEC 3309
	RxCRCEn	1b	Perform received data CRC operation
	TxCRCEn	1b	Add CRC operated data to transmit data
	ParityOdd	0b	Parity operation method not in use
	ParityEn	0b	Parity operation method not in use

Chapter5 COMMAND FUNCTION

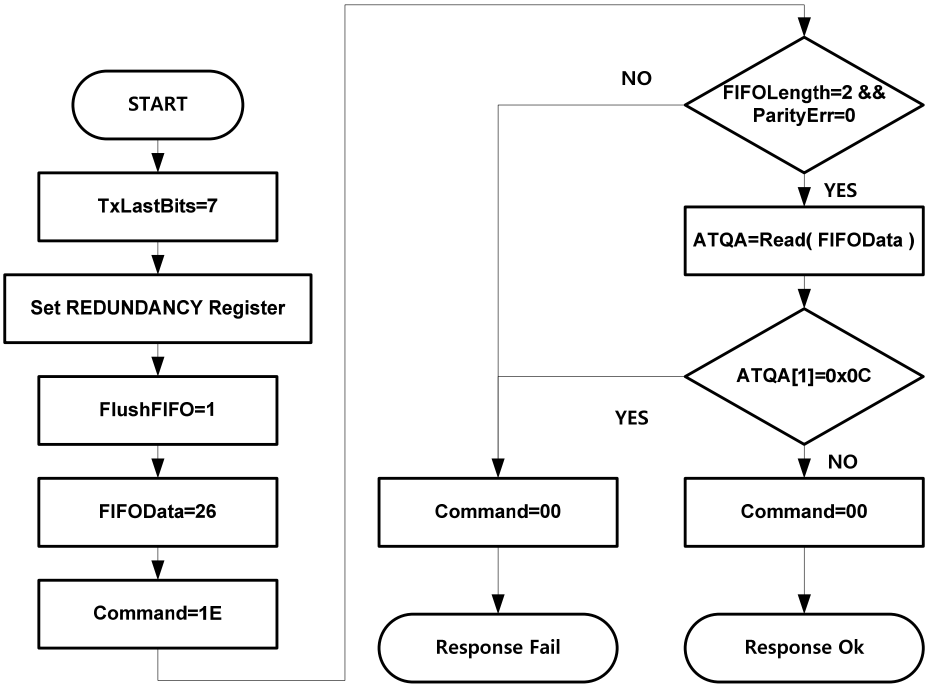
5.1 OUTLINE

Command function for each protocol is explained. Only ISO standard protocol functions are explained.

5.2 ISO/IEC 14443A COMMAND FUNCTIONS

5.2.1 ISO14443A_REQA() Function

Picture 5-1 is ISO14443A_REQA() Function Flow Chart.



Picture 5-1 ISO14443A_REQA() Function Flow Chart

REQA Command receives basic information of tag prior to receiving tag UID. REQA response is called ATQA and this response has tag manufacturer information, tag UID length and Bit frame anti-collision information.

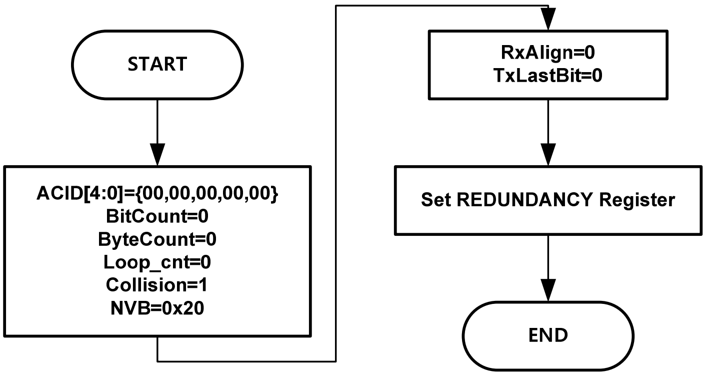
First, REQA is 7 bit 26h(010_0110b). To transmit 7 bit user must set *TxLastBits* to 7. After REDUNDANCY register should be set but since REQA do not use CRC, set both *TxCRCEn* and *RxCRCEn* as 0, and to use Parity, each *ParityOdd* and *ParityEn* to be set as 1. Then, delete FIFO data. Setting *FIFOFlush* as 1, FIFO data is deleted. Lastly, store 26h data to FIFO and store TRANCEIVE(0x1E) to command register to execute transmitting and receiving. Then, wait until transmitting and receiving is complete. For RSK Firmware approximately 500us is needed for this process.

To check ATQA is received properly, check *ParityErr* flag and *FIFOLength* flag. Since ATQA is configured with 2 byte, *FIFOLength* should be 2 and *ParityErr* value should be 0. If *FIFOLength* value is not 2 or *ParityErr* is not 0, ATQA is not received properly, thus, Response is Failed. When ATQA[1] is 0x0C, it is because of Jewel tag response, thus, processed as Response Fail.

WUPA command also is 7 bit command as REQA. Therefore, operation process is same as REQA. However, WUPA command is 52.

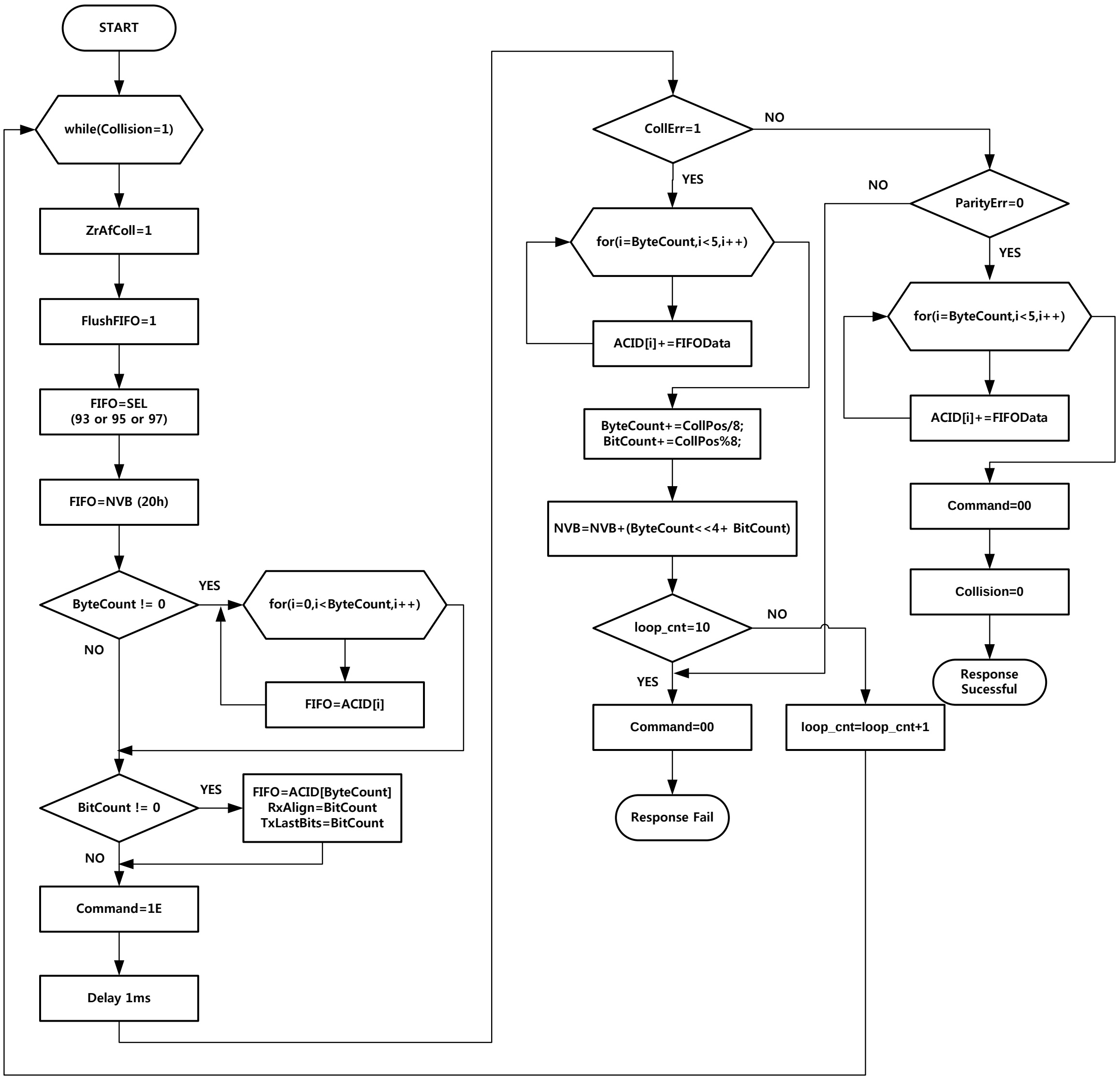
5.2.2 ISO14443A_AntiCollision() Function

ISO14443A_AntiCollision() Function execute Anti-Collision to read tag UID. ISO14443A_AntiCollision() Function has somewhat complicated configuration. Therefore, user must understand ISO/IEC 14443A Anti-Collision algorithm before checking Firmware flow chart. Below Picture 5-2 displays ISO14443A_AntiCollision() Function initialization Function Flow Chart.



Picture 5-2 ISO14443A_AntiCollision() Function Initialization

From Picture 5-2 ACID is variable to store UID. It's made of 5 bytes. BitCount is bit count of last byte received up to that point. ByteCount is total received byte count up to that point. loop_cnt is a variable to exit Anti-Collision loop after collision occurs over 10 times. NVB(Number of Valid Bits) is total bit count of normally received bit count.



Picture 5-3 ISO14443A_AntiCollision() Function Flow Chart

Picture 5-3 is ISO14443A_AntiCollision() Function Flow Chart. Left column displays flow of prior to receiving is executed. Center column displays flow when Collision Error occurs during receiving. Right column flow is when Collision Error does not occur. As seen in above picture, signal re-transmits when Collision Error occurs.

ZrAfColl should be set to 1 so all the data received after Collision to be 0. When *ZrAfColl* is set to 1, portion of previously received UID and current UID received portion can be added. Afterward delete all the FIFO data and input SEL command to FIFO. SEL command is selected by Cascade Level, basically, one value from 93, 95, 97 based on UID length. Next NVB value is inputted to FIFO and this value is currently received UID bit count. This value changes as Anti-Collision executes. Initial value is 20h since SEL command and NVB length are 2 bytes. Now store all received UID to FIFO. If there is no received UID, no data is entered. Since FIFO is stored data by byte, if received UID is not divisible by byte, FIFO is set up to where it is divisible and remaining bit is stored and sent using *RxLastBit* and *TxLastBit.* Transmit is ready and after transmitting, there is about 1ms delay.

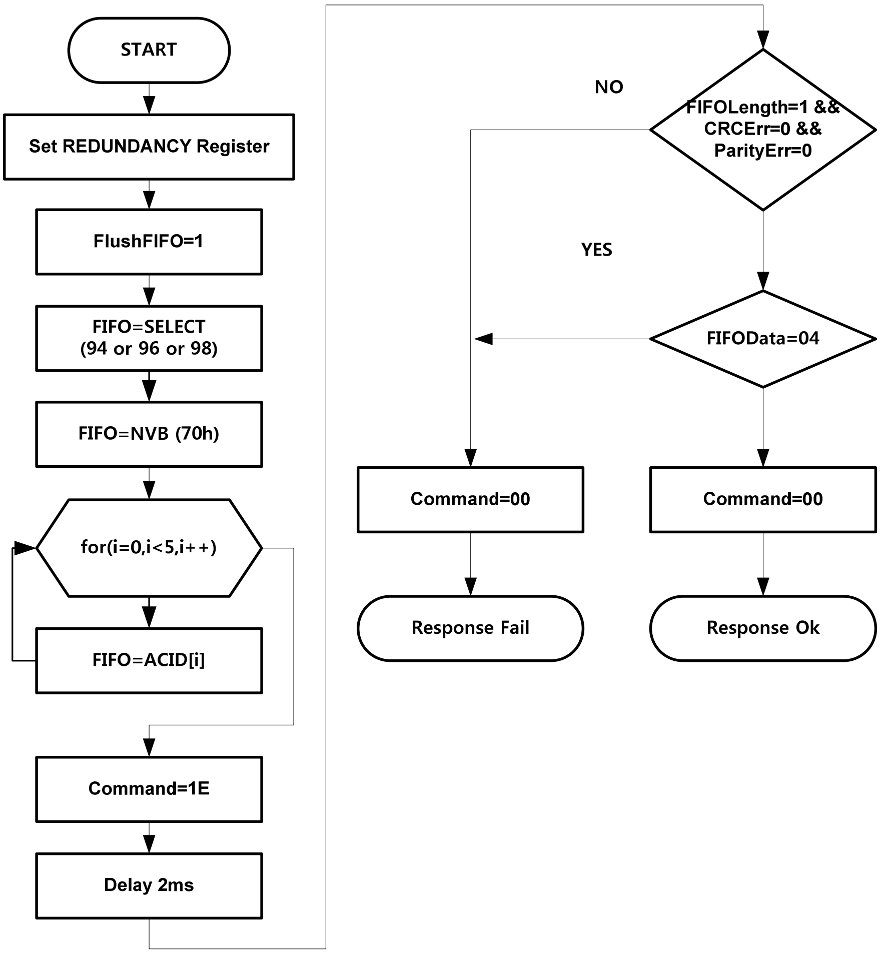
When receiving starts, first Collision Error is detected. IF Collision occurs, UID received in ACID variable is stored. As explained above *ZrAfColl* is used stored in addition. Next update ByteCount, BitCount, and NVB. Afterward confirm loop_cnt to see if Collision Error occurred more than 10 times. If less than 10 occurrence, sending continues. Since ByteCount, BitCount, NVB, and ACID are updated, transmit data is modified.

If Collision Error does not occur, check Parity Error occurrence. If no Parity Error, store received data to ACID and complete Anti-Collision algorithm.

5.2.3 ISO14443A_SELECT() Function

Picture 5-4 is ISO14443A_SELECT() Function Flow Chart. SELECT Command use both CRC and Parity. Therefore, *TxCRCEn* and *RxCRCEn, ParityOdd* and *ParityEn* are set to 1. Same as above functions delete FIFO data and store command. SELECT Command also changes its command to 94, 96 and 98 by Cascade Level. To send NVB, it sends all UID, thus, set to 70. Lastly, transmit after storing tag UID to FIFO. SELECT Command delay time is about 2ms.

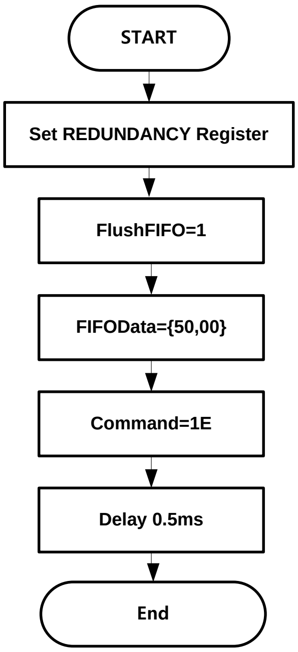
After completion of receiving, check *FIFOLength.* SELECT Command response is one byte and called SAK(SELECT ACK). Same as other commands, check CRC Error and Parity Error, then read SAK from FIFO to confirm value is 04. If SAK is 04, it implies "UID not complete".



Picture 5-4 ISO14443A_SELECT() Function Flow Chart

5.2.4 ISO14443A_HALT() Function

Picture 5-5 ISO14443A_HALT() Function Flow Chart. HALT Command is 50 and 00, and since there is no response, there is no receive part. HALT Function also use CRC and Parity for transmitting, thus, *TxCRCEn* and *RxCRCEn, ParityOdd* and *ParityEn* should be set to 1.

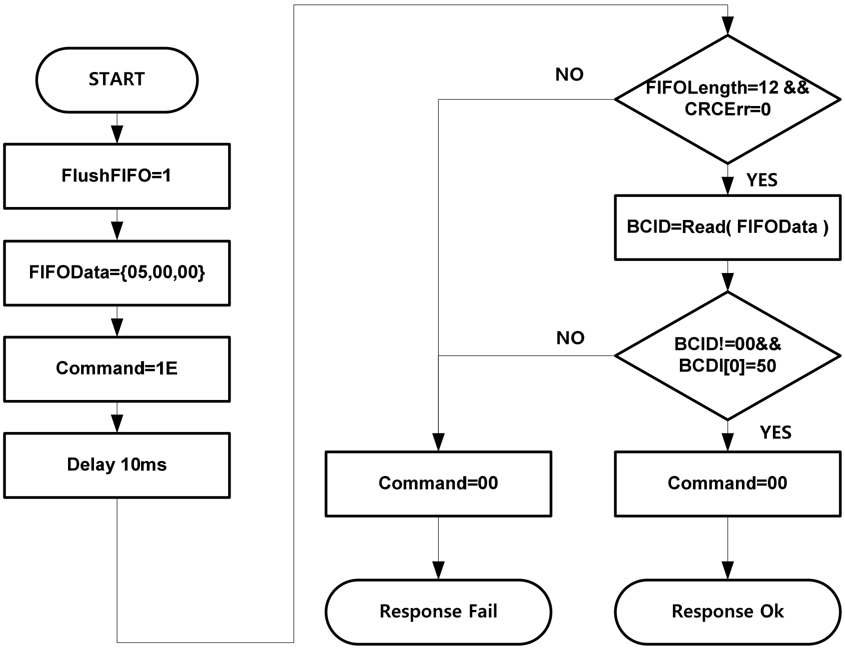


Picture 5-5 ISO14443A_HALT Function Flow Chart

5.3 ISO/IEC 14443B COMMAND FUNCTION

5.3.1 ISO14443B_REQB() Function

Picture 5-6 is ISO14443B_REQB() Function Flow Chart.

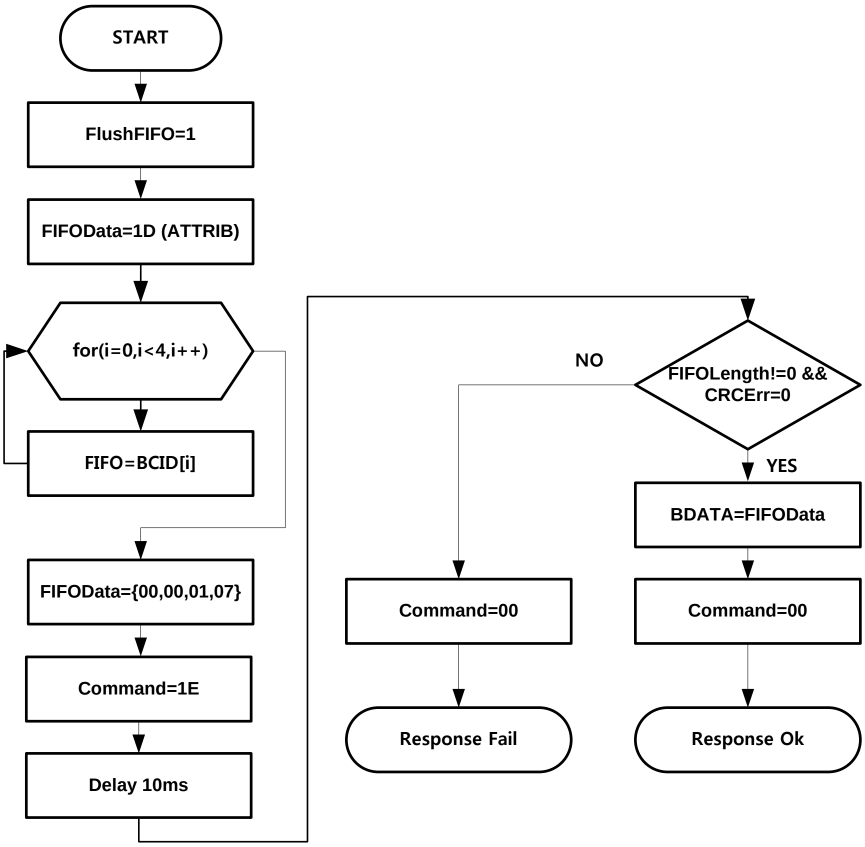


Picture 5-6 ISO14443B_REQB Function Flow Chart

REQB Command of ISO/IEC 14443B is a command to read tag UID. First, delete data from FIFO and store 05, 00, and 00 to FIFO. Then, transmit. For transmit completion and receiving of response, total delay time is about 10ms. After receiving completion, read *FIFOLength* flag to check whether data is 12 byte and CRC error occurrence. If *FIFOLength* is 12 and *CRCErr* value is 0, tag UID is received correctly. Read FIFO data and store to BCID variable. Lastly, confirm BCID first byte is 50 instead of 00. ISO/IEC 14443B tag UID begins with 50 thus, if any UID begins with 00 then incorrect data is received and process as Response Fail. If BCID first byte begins with 50, data is received properly thus store 00 *Command* flag and complete receiving process.

5.3.2 ISO14443B_ATTRIB() Function

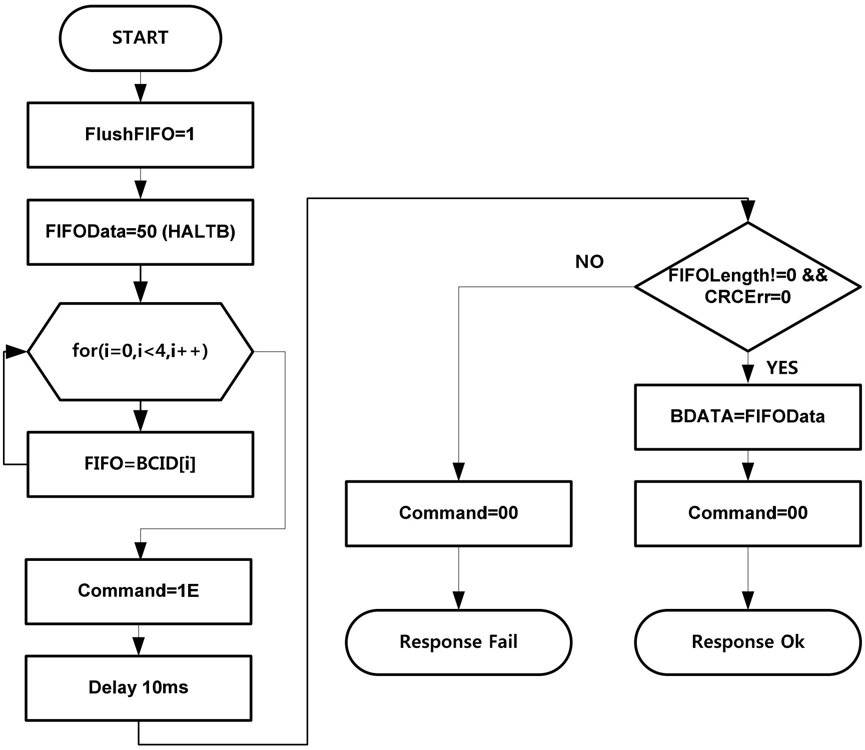
Picture 5-7 is ISO14443B_ATTRIB() Function Flow Chart. ATTRIB Function is commands to change various settings related to protocols. For example, transmit speed, SOF, and EOF use can be selected. ATTRIB Command include a portion of 14443B UID. In Picture 5-7 user can see BCID storing to FIFO before transmitting. BData is variable to store ATTRIB Command response. As other commands, ATTRIB Command checks *FIFOLength* flag and *CRCErr* flag to confirm receiving is completed correctly.



Picture 5-7 ISO14443B_ATTRIB() Function Flow Chart

5.3.3 ISO14443B_HALTB() Function

Picture 5-8 is ISO14443B_HALTB() Function Flow Chart. As seen below, it's very similar to ATTRIB. Therefore, data is stored to FIFO, transmit/receiving, then confirm *FIFOLength* and *CRCErr.* HALTB Command is 50.

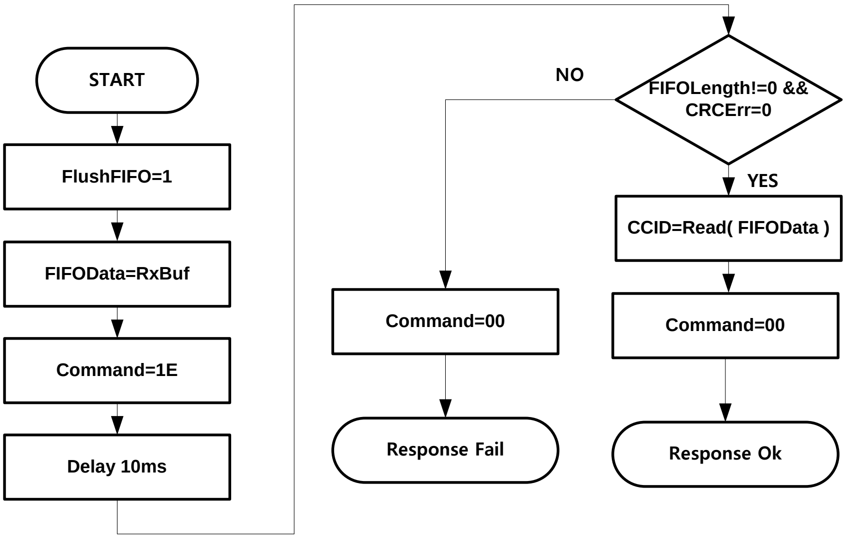


Picture 5-8 ISO14443B_HALTB() Function Flow Chart

5.4 ISO/IEC 15693 COMMAND FUNCTION

5.4.1 ISO15693_INVENTORY() Function

Picture 5-9 is ISO15693_INVENTORY() Function Flow Chart.

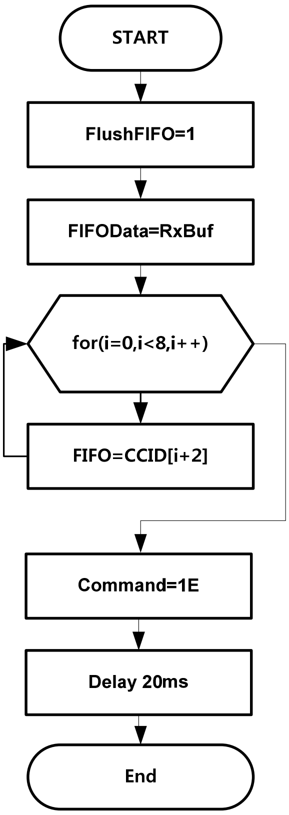


Picture 5-9 ISO15693_INVENTORY() Function Flow Chart

ISO15693_INVENTORY() Function is to execute ISO/IEC 15693 Inventory Command. Inventory Command is a function to read tag UID. ISO/IEC 15693 Firmware is somewhat different from existing ISO/IEC 14443A or ISO/IEC 14443B. ISO/IEC 15693 Functions receives all the FIFO stored commands from host. RxBUf is a register storing data received from host. ISO15693_INVENTORY() Function checks receiving data 10ms after transmitting data received from host.

5.4.2 ISO15693_STAYQUIET() Function

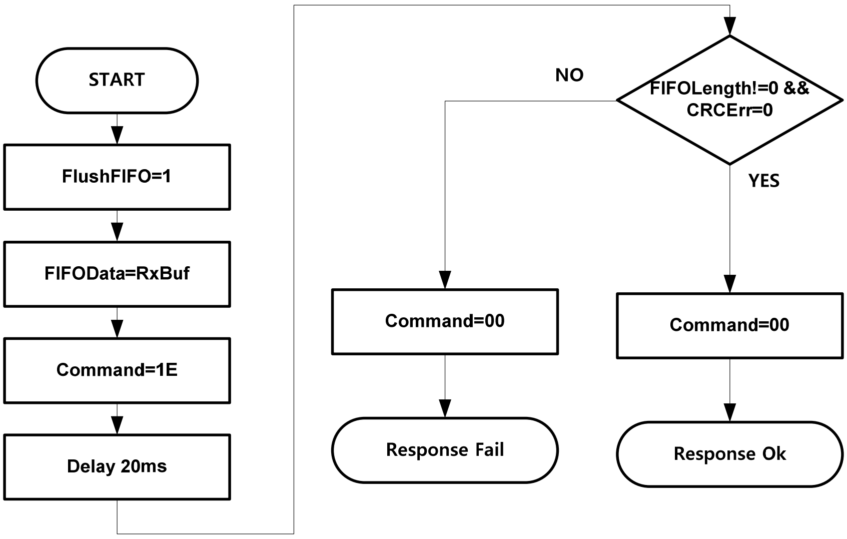
Picture 5-10 is ISO15693_STAYQUIET() Function Flow Chart. StayQuiet Command is command to send tag in quiet mode thus there is no response. Therefore, there is no part to handle receiving. StayQuiet Command includes a UID thus specific tag is selected and executed. Therefore, CCID with ISO/IEC 15693 UID is stored in FIFO and transmitted. StayQuiet Command also as with Inventory Command transmit data after receiving data from host.



Picture 5-10 ISO15693_STAYQUIET() Function Flow Chart

5.4.3 ISO15693_READ_SINGLE_BLOCK() Function

Picture 5-11 is ISO15693_READ_SINGLE_BLOCK() Function Flow Chart. Read Single Block Command reads specific tag block data. Tag memory is divided into blocks and it's size is different by different manufacturer and tag chip. ISO15693_READ_SINGLE_BLOCK() Function also transmit after receiving command from host. Read Single Block Command delay time is about 20ms.



Picture 5-11 ISO15693_READ_SINGLE_BLOCK() Function Flow Chart

5.4.4 ISO15693_WRITE_SINGLE_BLOCK() Function

ISO15693_WRITE_SINGLE_BLOCK() Function executes ISO/IEC 15693의 Write Single Block Command. Write Single Block Command changes tag memory block data. ISO15693_WRITE_SINGLE_BLOCK() Function Flow Chart is same as ISO15693_READ SINGLE_BLOCK() Function. ISO15693_WRITE_SINGLE_BLOCK() Function delay time also is 20ms.

5.4.5 ISO15693_LOCK_BLOCK() Function

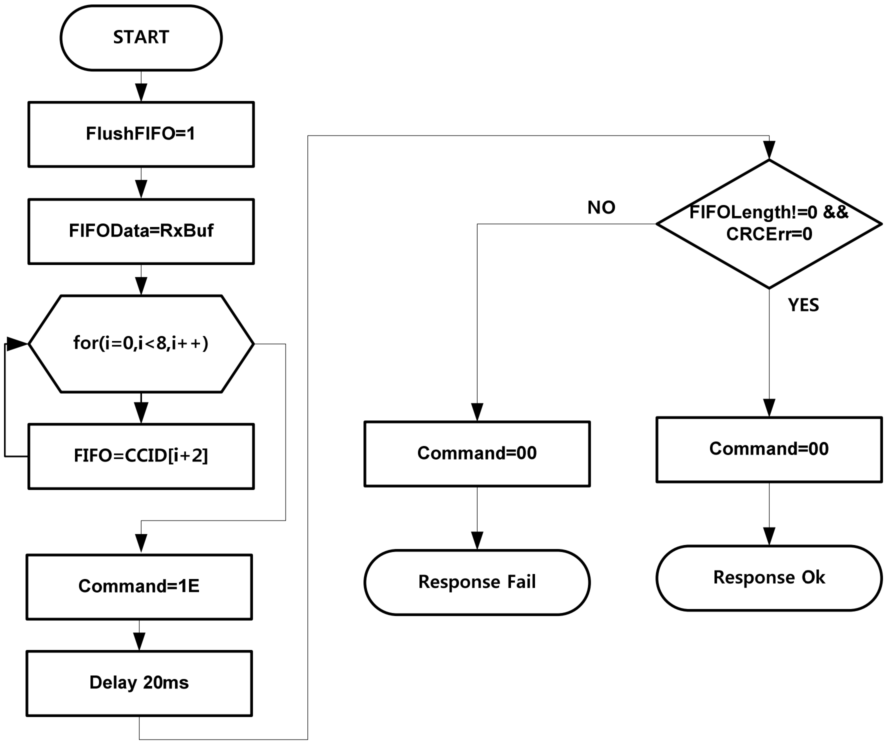
ISO15693_LOCK_BLOCK() Function executes ISO/IEC 15693 Lock Block Command. Lock Block Command locks specified tag block so data can't be changed. ISO15693_LOCK_BLOCK() Function Flow Chart is also same as Picture 5-11 and delay time is 20ms same as Read Single Block Command.

5.4.6 ISO15693_READ_MULTIPLE_BLOCK() Function

ISO15693_READ_MULTIPLE_BLOCK() Function executes ISO/IEC15693 Read Multiple Block Command. Read Multiple Block Command allows reading of multiple blocks simultaneously. ISO15693_READ_MULTIPLE_BLOCK() Function Flow Chart is also same as Picture 5-11. However, large data need to respond thus delay time is 250ms.

5.4.7 ISO15693_SELECT() Function

Picture 5-12 is ISO15693_SELECT() Function Flow Chart. ISO15693_SELECT() Function executes ISO/IEC 15693 Select Command. Select Command replace tags to Selected status. Select Command includes tag UID. From the picture CCID is a register storing previously recognized tag UID.



Picture 5-12 ISO15693_SELECT() Command Flow Chart

5.4.8 ISO15693_RESETTOREADY() Function

ISO15693_RESETTOREADY() Function executes ISO/IEC 15693 Reset-to-Ready Command. Reset-to-Ready Command changes Selected or Quiet mode tag to Ready mode. ISO15693_RESETTOREADY() Function Flow Chart is as in Picture 5-12.

5.4.9 ISO15693_WRITE_AFI() Function

ISO15693_WRITE_AFI() Function executes ISO/IEC 15693 Write AFI Command. Write AFI Command executes changing of AFI (Application family identifier) value. ISO15693_WRITE_AFI() Function Flow Chart is as in Picture 5-12.

5.4.10 ISO15693_LOCK_AFI_BLOCK() Function

ISO15693_LOCK_AFI_BLOCK() Function executes ISO/IEC 15693 Lock AFI Command. Lock AFI Command locks changes in AFI value. ISO15693_LOCK_AFI_BLOCK() Function Flow Chart is same as in Picture 5-12.

5.4.11 ISO15693_WRITE_DSFID() Function

ISO15693_WRITE_DSFID() Function executes ISO/IEC 15693 Write DSFID Command. Write DSFID Command executes changing of DSFID (Data Storage Format Identifier) value. ISO15693_WRITE_DSFID() Function Flow Chat is same as Picture 5-12.

5.4.12 ISO15693_LOCK_DSFID() Function

ISO15693_LOCK_DSFID() Function executes ISO/IEC 15693 Lock DSFID Command. Lock DSFID locks DSFID(Data storage format Identifier) value so value can’t be changed. ISO15693_LOCK_DSFID() Function Flow Chart is as Picture 5-12.

5.4.13 ISO15693_GET_SYSTEM_INFORMATION() Function

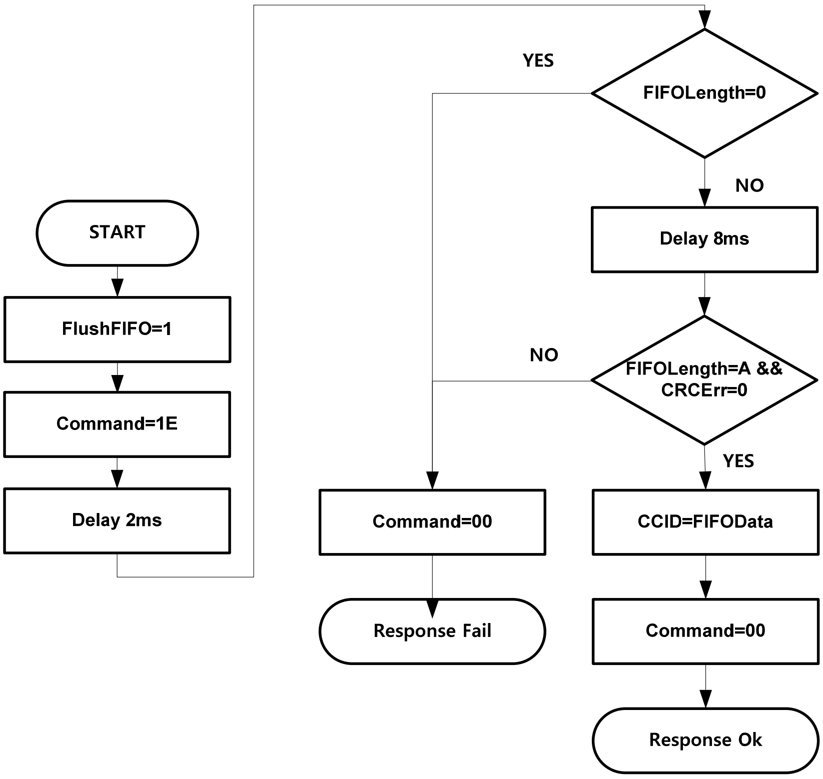
ISO15693_GET_SYSTEM_INFORMATION() Function executes ISO/IEC 15693 Get System Information Command. Get System Information is used to read tag system information. Get System Information response include UID, AFI, DSFID and memory information. ISO15693_GET_SYSTEM_INFORMATION() Function does not include tag UID and Flow Chart is as in Picture 5-11.

5.4.14 ISO15693_GET_MULTIPLE_SECURITY_STATUS() Function

ISO15693_GET_MULTIPLE_SECURITY_STATUS() Function executes ISO/IEC 15693 Get multiple block security status Command. Get multiple block security status Command reads tag memory block lock status. Get multiple block security status Command also does not include tag UID so Flow Chart is same as Picture 5-11. However, Get multiple block security status Command delay time is 250ms.

5.4.15 ISO15693_EOF() Function

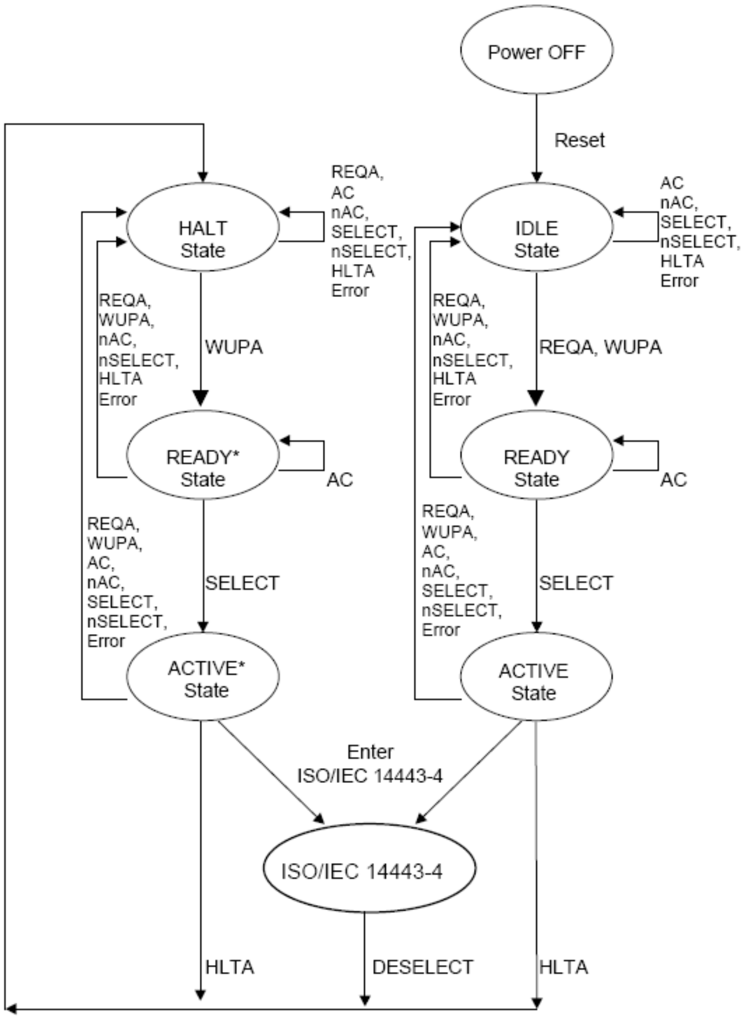
Picture 5-13 is ISO15693_EOF() Function Flow Chart. ISO15693_EOF() Function Command transmits EOF signal and when there is a tag response UID is received. It's used for processing of Anti-Collision algorithm. Since EOF is signal waveform not data, thus, transmits without storing any data in FIFO. After 2ms, if data remains in FIFO, tag is responding to UID. Thus, wait another 8ms and check if FIFO stored data is 10 byte, then bring data from FIFO. When no data in FIFO after 2ms, FIFO stored data is not 10 byte or *CRCErr* flag is 1, then command is processed as no response.



Picture 5-13 ISO15693_EOF() Function Flow Chart

Chapter6 Appendix

6.1 ISO/IEC 14443A State Transition Diagram


Power OFF
Reset
HALT State
REQA, AC nAC, SELECT, nSELECT, HLTA Error
IDLE State
AC nAC, SELECT, nSELECT, HLTA Error
REQA, WUPA, nAC, nSELECT, HLTA Error
WUPA
REQA, WUPA, nAC, nSELECT, HLTA Error
REQA, WUPA
READY* State
AC
READY State
AC
REQA, WUPA, AC, nAC, SELECT, nSELECT, Error
SELECT
REQA, WUPA, AC, nAC, SELECT, nSELECT, Error
SELECT
ACTIVE* State
ACTIVE State
Enter ISO/IEC 14443-4
ISO/IEC 14443-4
HLTA
DESELECT
HLTA

*** AC - ANTICOLLISION Command (matched UID)**
*** nAC - ANTICOLLISION Command (not matched UID)**
*** SELECT - SELECT Command (matched UID)**
*** nSELECT SELECT Command (not matched UID)**
*** DESELECT - DESELECT Command, defined in ISO/IEC 14443-4**

Picture 6-1 ISO 14443A State Transition Diagram

6.2 ISO/IEC 14443B State Transition Diagram

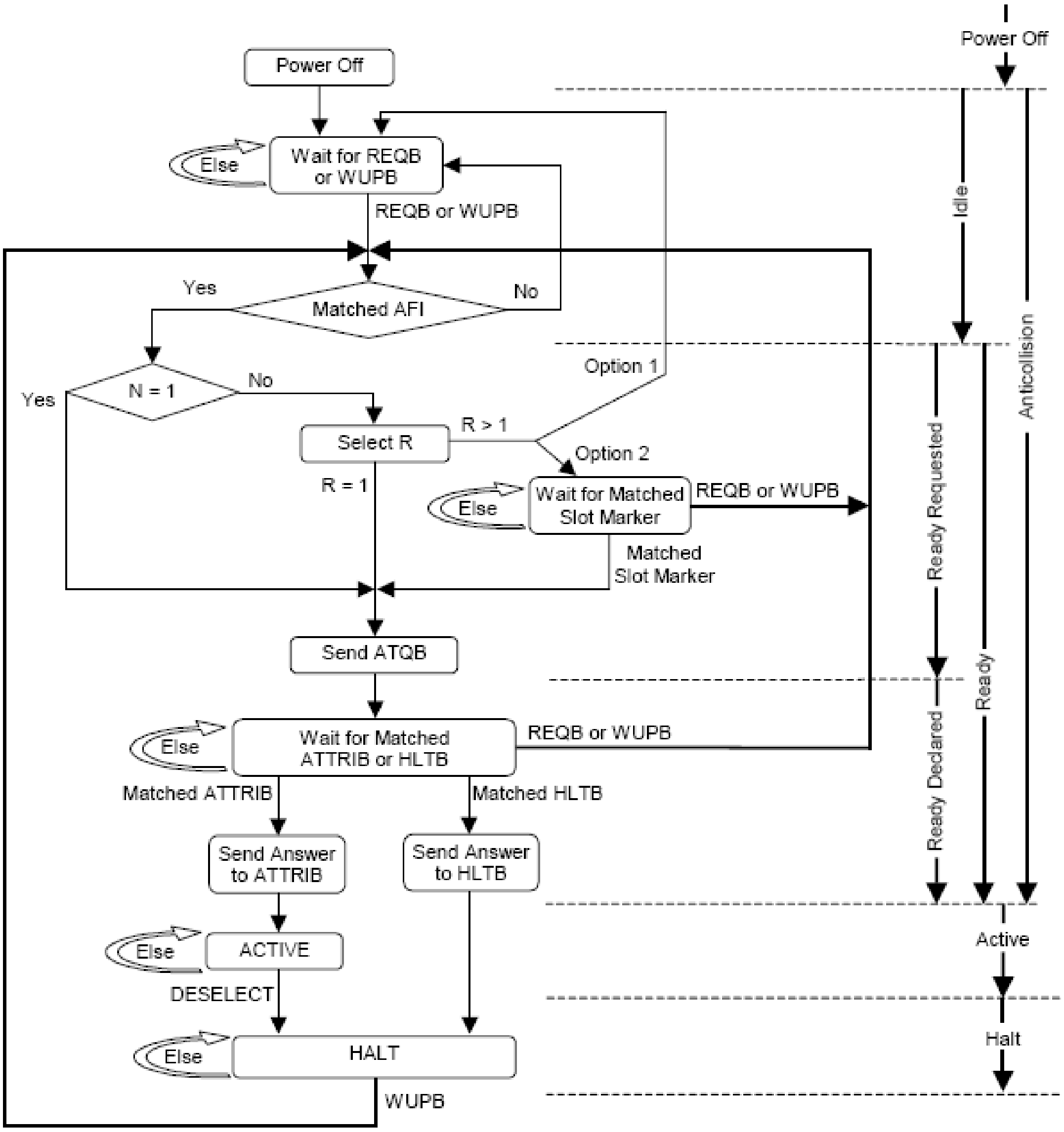


Figure 19 — PICC state transition flowchart example

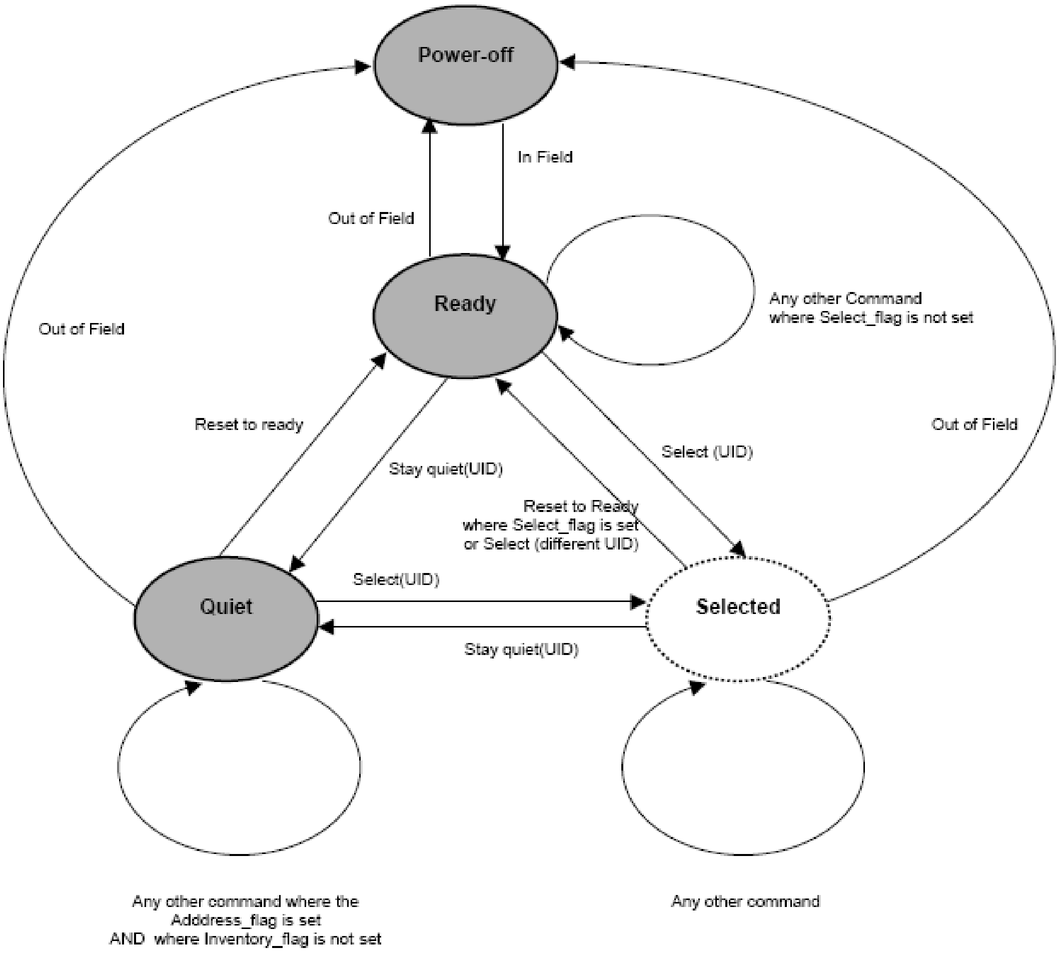
NOTE 1 R is a random number chosen by the PICC in the range from 1 to N (for coding of N see 7.7.4).

NOTE 2 Option 1 – For PICCs not supporting Slot-MARKER Command (Probabilistic approach).

Option 2 – For PICCs supporting Slot-MARKER Command (Timeslot approach).

Picture 6-2 ISO14443B State Transition Diagram

6.3 ISO/IEC 15693 State Transition Diagram


Power-off
In Field
Out of Field
Ready
Any other Command where Select_flag is not set
Out of Field
Reset to ready
Stay quiet(UID)
Select (UID)
Out of Field
Reset to Ready where Select_flag is set or Select (different UID)
Quiet
Select(UID)
Selected
Stay quiet(UID)
Any other command where the Adddress_flag is set AND where Inventory_flag is not set
Any other command

NOTE 1 The intention of the state transition method is that only one VICC should be in the Selected state at a time.

NOTE 2 The VICC state transition diagram shows only valid transitions. In all other cases the current VICC state remains unchanged. When the VICC cannot process a VCD request (e.g. CRC error, etc.), it shall stay in its current state.

NOTE 3 The Selected state is represented with a dotted line to show its support by the VICC is optional.

Picture 6-3 ISO/IEC 15693 State Transition Diagram

6.4 Firmware download (RSK Board)

Firmware source included with RSK Board is complied by Keil uVsionV2.40a and firmware write procedure is as below.

6.4.1 Firmware download procedure

- Convert to MCU Program mode
- Power reset (Power switch off/on or adapter unplug / plug-in)
- Firmware download (Flip 3.2.2)
- From MCU Program mode convert to operating mode
- Power reset (Power switch off/on or adapter unplug / plug-in)
- RSK system start

6.4.2 MCU Program mode

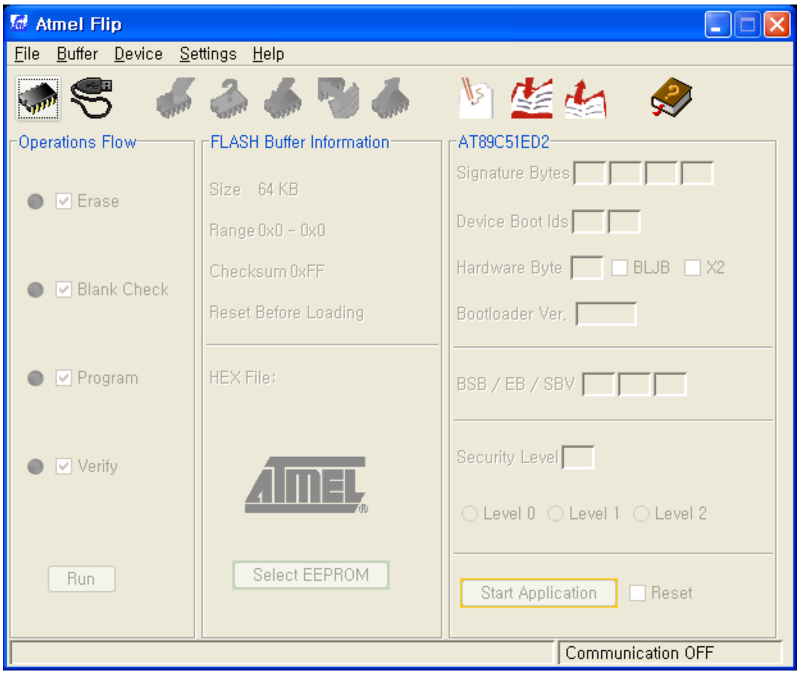
MCU Program mode는 Picture 6-4와 같이 RSK200L은 외부에 위치한 switch의 레버를 상으로 위치시키고 RSK100/300은 JT1의 pin header 1번과 2번을 pin header cap 으로 연결 시키면 Program mode로 전환된다.

(a) RSK200L (b) RSK100/300

Picture 6-4 RSK system program mode switch and pin header

6.4.3 Flip Use Method

For Firmware download program use Atmel Flip (Flexible In-system Programmer) 3.2.2. Flip3.2.2 requires JAVA program. If you installed JVM to operate SDK, you will not have any problem operating Flip3.2.2. To use software please refer to Picture 6-5. Before using Flip, need to RSK System Power Reset.



Size 64 KB

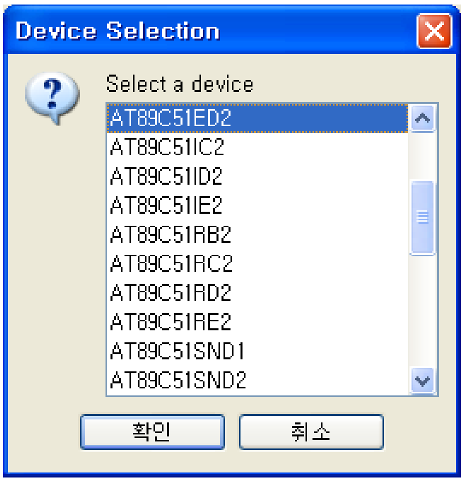
Range 0x0 - 0x0

Checksum 0xFF

Reset Before Loading

Communication OFF

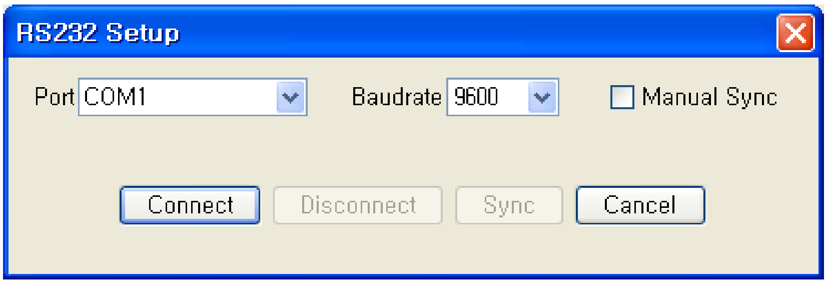
(a) Program Initialization Screen



Select a device

AT89C51ED2
AT89C51IC2
AT89C51ID2
AT89C51IE2
AT89C51RB2
AT89C51RC2
AT89C51RD2
AT89C51RE2
AT89C51SND1
AT89C51SND2

(b) Device select (AT89C51ED2)

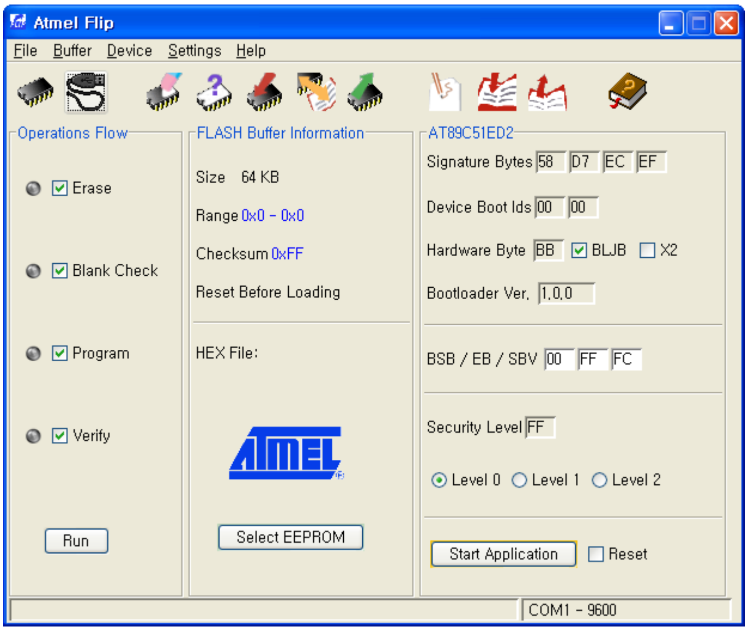


Port COM1

Baudrate 9600

□ Manual Sync

(c) RS232 Setup (Baud-rate :9600bps) : connect



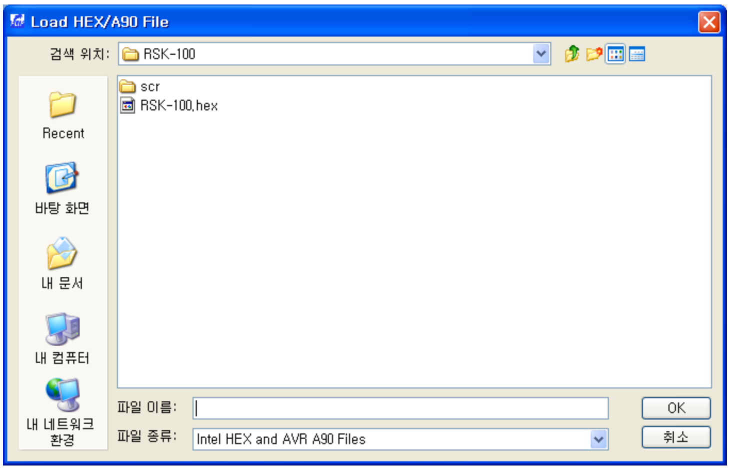
☑ Erase
☑ Blank Check
☑ Program
☑ Verify

FLASH Buffer Information
Size 64 KB
Range 0x0 - 0x0
Checksum 0xFF
Reset Before Loading
HEX File:

AT89C51ED2
Signature Bytes 58 D7 EC EF
Device Boot Ids 00 00
Hardware Byte BB ☑ BLJB □ X2
Bootloader Ver. 1.0.0
BSB / EB / SBV 00 FF FC
Security Level FF
● Level 0 ○ Level 1 ○ Level 2
□ Reset

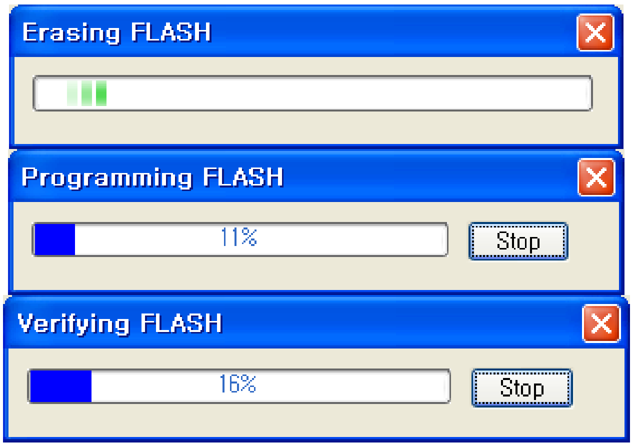
COM1 - 9600

(d) Complete Environment Setting



검색 위치: RSK-100
scr
RSK-100.hex
파일 이름:
파일 종류: Intel HEX and AVR A90 Files

(e) Load HEX file (RSK-100.hex) / hex file path



(f) RUN / download Complete

Picture 6-5 Firmware Download

[Reference]

[1] International Standard ' ISO/IEC 14443 ' , PartII
[2] International Standard ' ISO/IEC 15693 ' , PartII

3ALogics 13.56MHz Multi-protocol RFID reader system *Any*Read^{TM} Firmware User Manual

It's RFID

RFID & Mobile SoC for Ubiquitous Technology

Contact

3Alogics Inc.
7th Fl., Hyundai-office Bldg.,
9-4, Sunae-dong, Bundang-gu,
Seongnam-si, Gyeonggi-do, 463-783 Korea
TEL : 82 – 31 – 715 -7117
FAX : 82 – 31 – 719 -7551
rfid@3alogics.com
http://www.3alogics.com