

RR10

Programming Guide

Version 2.0

RR1X Series Programming Guide

Revision History

Rev.	Date	Description
1.0	2008/8/25	New Release

RR1X Series Programming Guide

Table of Contents

Revision History	2
Table of Contents	3
List of Figures	4
List of Tables	5
1. Introduction	12
2. Software Architecture	12
3. Application Programming Interface	13
3.1 Win32	13
3.1.1 Management	13
3.1.2 ISO15693	15
3.1.3 ISO14443	28
3.1.4 Felica	38
3.1.5 ISO18092	38
3.1.6 Mstar Peer-to-Peer	39
3.1.7 Secure	40
3.2 WinCE	43
4. Host Commands and Responses	44
4.1 Communication Interface	44
4.1.1	44
4.1.2 UART	45
4.2 Host Command and Response Format	51
4.3 Host Command Code List	52
4.4 Host Commands and Responses	54
4.4.1 Connection	54
4.4.2 Select RF Parameter	55
4.4.3 Get Version	57
4.4.4 ISO15693 Tag Inventory	59
4.4.5 ISO15693 Tag Access	62
4.4.6 Mifare Ultralight Tag Access	101
4.4.7 ISO14443A Tag Inventory	104
4.4.8 TOPAZ Tag Inventory and Access	105
4.4.9 ST Tag Inventory	109
4.4.10 ST Tag Access	110
4.4.11 ISO14443B Tag Inventory	114
4.4.12 ISO14443B Tag Access	115
4.4.13 Felica Tag Inventory	124
4.4.14 Mstar P2P Tx	125
4.4.15 Mstar P2P Rx	126
4.4.16 ISO18092 Target	127
4.4.17 ISO18092 Initiator	128
5. In-System-Programming	131
6. Appendix	133
6.1 Error Code definition	133
6.1.1 ISO15693 Standard Defined Response Error Code	133
6.2 Tags' State Diagram	134
6.2.1 ISO15693 Tag's State Diagram	134
6.2.2 ISO14443A Tag's State Diagram	135
6.2.3 ISO14443B Tag's State Diagram	136
7. Reference	138

RR1X Series Programming Guide

List of Figures

FIGURE 1 SOFTWARE ARCHITECTURE.....	12
FIGURE 2 USB COMMUNICATION SEQUENCE.....	44
FIGURE 3 UART COMMUNICATION SEQUENCE	45
FIGURE 9 AFI CODING	61
FIGURE 10 IN-SYSTEM-PROGRAMMING SEQUENCE.....	132
FIGURE 11 ISO15693 STANDARD DEFINE RESPONSE ERROR CODE	133
FIGURE 12 ISO15693 TAG'S STATE DIAGRAM	134
FIGURE 13 ISO14443A TAG'S STATE DIAGRAM.....	135
FIGURE 14 ISO14443B TAG'S STATE DIAGRAM.....	136
FIGURE 15 PRINTER PORT OUTLINE	137

RR1X Series Programming Guide

List of Tables

TABLE 1 GENERIC FRAME FORMAT OF HOST COMMAND AND TARGET RESPONSE	51
TABLE 2 FIELD DESCRIPTION OF GENERIC COMMAND / RESPONSE FORMAT	51
TABLE 3 HOST COMMAND LIST.....	52
TABLE 4 FEATURE MAPPING FOR RR1X MODELS.....	53
TABLE 5 CONNECTION COMMAND.....	54
TABLE 6 FIELD DESCRIPTION OF CONNECTION COMMAND.....	54
TABLE 7 RESPONSE OF CONNECTION COMMAND	54
TABLE 8 FIELD DESCRIPTION OF CONNECTION RESPONSE.....	54
TABLE 9 SELECT RF PARAMETER COMMAND.....	55
TABLE 10 FIELD DESCRIPTION OF SELECT RF PARAMETER COMMAND.....	55
TABLE 11 RESPONSE OF SELECT RF PARAMETER COMMAND	56
TABLE 12 FIELD DESCRIPTION OF SELECT RF PARAMETER RESPONSE.....	56
TABLE 13 GET VERSION COMMAND.....	57
TABLE 14 FIELD DESCRIPTION OF GET VERSION COMMAND.....	57
TABLE 15 RESPONSE OF GET VERSION COMMAND	58
TABLE 16 FIELD DESCRIPTION OF GET VERSION RESPONSE	58
TABLE 17 ISO15693 TAG INVENTORY COMMAND.....	59
TABLE 18 FIELD DESCRIPTION OF ISO15693 TAG INVENTORY COMMAND	59
TABLE 19 RESPONSE OF ISO15693 TAG INVENTORY COMMAND	60
TABLE 20 FIELD DESCRIPTION OF ISO15693 TAG INVENTORY RESPONSE.....	60
TABLE 21 ISO15693 TAG ACCESS COMMAND	62
TABLE 22 STAY QUIET OPERATION	62
TABLE 23 FIELD DESCRIPTION OF STAY QUIET OPERATION	62
TABLE 24 RESPONSE OF STAY QUIET OPERATION	62
TABLE 25 FIELD DESCRIPTION OF STAY QUIET OPERATION RESPONSE.....	62
TABLE 26 READ SINGLE BLOCK OPERATION.....	63
TABLE 27 FIELD DESCRIPTION OF READ SINGLE BLOCK OPERATION	63
TABLE 28 RESPONSE OF READ SINGLE BLOCK OPERATION	64
TABLE 29 FIELD DESCRIPTION OF READ SINGLE BLOCK OPERATION RESPONSE	64
TABLE 30 WRITE SINGLE BLOCK OPERATION	65
TABLE 31 FIELD DESCRIPTION OF WRITE SINGLE BLOCK OPERATION	65
TABLE 32 RESPONSE OF WRITE SINGLE BLOCK OPERATION.....	65
TABLE 33 FIELD DESCRIPTION OF WRITE SINGLE BLOCK OPERATION RESPONSE.....	66
TABLE 34 LOCK BLOCK OPERATION	66
TABLE 35 FIELD DESCRIPTION OF LOCK BLOCK OPERATION.....	66
TABLE 36 RESPONSE OF LOCK BLOCK OPERATION	67
TABLE 37 FIELD DESCRIPTION OF LOCK BLOCK OPERATION RESPONSE.....	67

RR1X Series Programming Guide

TABLE 38 READ MULTIPLE BLOCKS OPERATION	68
TABLE 39 FIELD DESCRIPTION OF READ MULTIPLE BLOCKS OPERATION	68
TABLE 40 RESPONSE OF READ MULTIPLE BLOCKS OPERATION	69
TABLE 41 FIELD DESCRIPTION OF READ MULTIPLE BLOCKS OPERATION RESPONSE	69
TABLE 42 SELECT OPERATION	70
TABLE 43 FIELD DESCRIPTION OF SELECT OPERATION	70
TABLE 44 RESPONSE OF SELECT OPERATION	70
TABLE 45 FIELD DESCRIPTION OF SELECT OPERATION RESPONSE	70
TABLE 46 RESET TO READY OPERATION	71
TABLE 47 FIELD DESCRIPTION OF RESET TO READY OPERATION	71
TABLE 48 RESPONSE OF RESET TO READY OPERATION	71
TABLE 49 FIELD DESCRIPTION OF RESET TO READY OPERATION RESPONSE	71
TABLE 50 RESET TO READY OPERATION	72
TABLE 51 FIELD DESCRIPTION OF WRITE AFI OPERATION	72
TABLE 52 RESPONSE OF WRITE AFI OPERATION	72
TABLE 53 FIELD DESCRIPTION OF WRITE AFI OPERATION RESPONSE	72
TABLE 54 LOCK AFI OPERATION	73
TABLE 55 FIELD DESCRIPTION OF LOCK AFI OPERATION	73
TABLE 56 RESPONSE OF LOCK AFI OPERATION	73
TABLE 57 FIELD DESCRIPTION OF LOCK AFI OPERATION RESPONSE	73
TABLE 58 WRITE DSFID OPERATION	74
TABLE 59 FIELD DESCRIPTION OF WRITE DSFID OPERATION	74
TABLE 60 RESPONSE OF WRITE DSFID OPERATION	74
TABLE 61 FIELD DESCRIPTION OF WRITE DSFID OPERATION RESPONSE	74
TABLE 62 LOCK DSFID OPERATION	75
TABLE 63 FIELD DESCRIPTION OF LOCK DSFID OPERATION	75
TABLE 64 RESPONSE OF LOCK DSFID OPERATION	75
TABLE 65 FIELD DESCRIPTION OF LOCK DSFID OPERATION RESPONSE	75
TABLE 66 GET SYSTEM INFORMATION OPERATION	76
TABLE 67 FIELD DESCRIPTION OF GET SYSTEM INFORMATION OPERATION	76
TABLE 68 RESPONSE OF GET SYSTEM INFORMATION OPERATION	76
TABLE 69 FIELD DESCRIPTION OF GET SYSTEM INFORMATION OPERATION RESPONSE	77
TABLE 70 GET MULTIPLE BLOCK SECURITY STATUS OPERATION	78
TABLE 71 FIELD DESCRIPTION OF GET MULTIPLE BLOCK SECURITY STATUS OPERATION	78
TABLE 72 RESPONSE OF GET MULTIPLE BLOCK SECURITY STATUS OPERATION	79
TABLE 73 FIELD DESCRIPTION OF GET MULTIPLE BLOCK SECURITY STATUS OPERATION RESPONSE	79
TABLE 74 READ EAS OPERATION	80
TABLE 75 FIELD DESCRIPTION OF READ EAS OPERATION	80

RR1X Series Programming Guide

TABLE 76 RESPONSE OF READ EAS OPERATION.....	80
TABLE 77 FIELD DESCRIPTION OF READ EAS OPERATION RESPONSE.....	80
TABLE 78 WRITE EAS OPERATION	81
TABLE 79 FIELD DESCRIPTION OF WRITE EAS OPERATION.....	81
TABLE 80 RESPONSE OF WRITE EAS OPERATION	81
TABLE 81 FIELD DESCRIPTION OF WRITE EAS OPERATION RESPONSE.....	81
TABLE 82 LOCK EAS OPERATION.....	82
TABLE 83 FIELD DESCRIPTION OF LOCK EAS OPERATION	82
TABLE 84 RESPONSE OF LOCK EAS OPERATION.....	82
TABLE 85 FIELD DESCRIPTION OF LOCK EAS OPERATION RESPONSE.....	82
TABLE 86 SECURE READ OPERATION.....	83
TABLE 87 FIELD DESCRIPTION OF SECURE READ OPERATION	83
TABLE 88 RESPONSE OF SECURE READ OPERATION.....	84
TABLE 89 FIELD DESCRIPTION OF SECURE READ OPERATION RESPONSE.....	84
TABLE 90 SECURE WRITE OPERATION	85
TABLE 91 FIELD DESCRIPTION OF SECURE WRITE OPERATION.....	85
TABLE 92 RESPONSE OF SECURE WRITE OPERATION	86
TABLE 93 FIELD DESCRIPTION OF SECURE WRITE OPERATION RESPONSE.....	86
TABLE 94 GET OWNER ID OPERATION.....	87
TABLE 95 FIELD DESCRIPTION OF GET OWNER ID OPERATION	87
TABLE 96 RESPONSE OF GET OWNER ID OPERATION.....	87
TABLE 97 FIELD DESCRIPTION OF GET OWNER ID OPERATION RESPONSE	87
TABLE 98 WRITE PWD OPERATION	88
TABLE 99 FIELD DESCRIPTION OF WRITE PWD OPERATION	88
TABLE 100 RESPONSE OF WRITE PWD OPERATION.....	88
TABLE 101 FIELD DESCRIPTION OF WRITE PWD OPERATION RESPONSE.....	88
TABLE 102 LOCK PWD OPERATION.....	89
TABLE 103 FIELD DESCRIPTION OF LOCK PWD OPERATION	89
TABLE 104 RESPONSE OF LOCK PWD OPERATION.....	89
TABLE 105 FIELD DESCRIPTION OF LOCK PWD OPERATION RESPONSE	89
TABLE 106 READ KEY OPERATION	90
TABLE 107 FIELD DESCRIPTION OF READ KEY OPERATION.....	90
TABLE 108 RESPONSE OF READ KEY OPERATION	90
TABLE 109 FIELD DESCRIPTION OF READ KEY OPERATION RESPONSE	90
TABLE 110 WRITE KEY OPERATION	91
TABLE 111 FIELD DESCRIPTION OF WRITE KEY OPERATION.....	91
TABLE 112 RESPONSE OF WRITE KEY OPERATION.....	91
TABLE 113 FIELD DESCRIPTION OF WRITE KEY OPERATION RESPONSE.....	91

RR1X Series Programming Guide

TABLE 114 LOCK KEY OPERATION.....	92
TABLE 115 FIELD DESCRIPTION OF LOCK KEY OPERATION	92
TABLE 116 RESPONSE OF LOCK KEY OPERATION.....	92
TABLE 117 FIELD DESCRIPTION OF LOCK KEY OPERATION RESPONSE	92
TABLE 118 READ REGISTER OPERATION.....	93
TABLE 119 FIELD DESCRIPTION OF READ REGISTER OPERATION	93
TABLE 120 RESPONSE OF READ REGISTER OPERATION.....	93
TABLE 121 FIELD DESCRIPTION OF READ REGISTER OPERATION RESPONSE	93
TABLE 122 WRITE REGISTER OPERATION	94
TABLE 123 FIELD DESCRIPTION OF WRITE REGISTER OPERATION.....	94
TABLE 124 RESPONSE OF WRITE REGISTER OPERATION	94
TABLE 125 FIELD DESCRIPTION OF WRITE REGISTER OPERATION RESPONSE.....	94
TABLE 126 WRITE PWD WITH PWD OPERATION	95
TABLE 127 FIELD DESCRIPTION OF WRITE PWD WITH PWD OPERATION.....	95
TABLE 128 RESPONSE OF WRITE PWD WITH PWD OPERATION	96
TABLE 129 FIELD DESCRIPTION OF WRITE PWD WITH PWD OPERATION RESPONSE	96
TABLE 130 FAST READ SINGLE BLOCK OPERATION.....	97
TABLE 131 FIELD DESCRIPTION OF FAST READ SINGLE BLOCK OPERATION	97
TABLE 132 RESPONSE OF FAST READ SINGLE BLOCK OPERATION	98
TABLE 133 FIELD DESCRIPTION OF FAST READ SINGLE BLOCK OPERATION RESPONSE.....	98
TABLE 134 FAST WRITE SINGLE BLOCK OPERATION	99
TABLE 135 FIELD DESCRIPTION OF FAST WRITE SINGLE BLOCK OPERATION.....	99
TABLE 136 RESPONSE OF FAST WRITE SINGLE BLOCK OPERATION	100
TABLE 137 FIELD DESCRIPTION OF FAST WRITE SINGLE BLOCK OPERATION RESPONSE	100
TABLE 138 MIFARE ULTRALIGHT TAG ACCESS COMMAND.....	101
TABLE 139 READ OPERATION	101
TABLE 140 FIELD DESCRIPTION OF READ OPERATION.....	101
TABLE 141 RESPONSE OF READ OPERATION	102
TABLE 142 FIELD DESCRIPTION OF READ OPERATION RESPONSE.....	102
TABLE 143 WRITE OPERATION.....	103
TABLE 144 FIELD DESCRIPTION OF WRITE OPERATION	103
TABLE 145 RESPONSE OF WRITE OPERATION	103
TABLE 146 FIELD DESCRIPTION OF WRITE OPERATION RESPONSE.....	103
TABLE 147 ISO14443A TAG INVENTORY COMMAND	104
TABLE 148 FIELD DESCRIPTION OF ISO14443A TAG INVENTORY COMMAND.....	104
TABLE 149 RESPONSE OF ISO14443A TAG INVENTORY COMMAND	104
TABLE 150 FIELD DESCRIPTION OF ISO14443A TAG INVENTORY RESPONSE.....	104
TABLE 151 TOPAZ TAG INVENTORY AND ACCESS COMMAND	105

RR1X Series Programming Guide

TABLE 152 WAKE UP OPERATION	105
TABLE 153 FIELD DESCRIPTION OF TOPAZ TAG WAKE UP COMMAND.....	105
TABLE 154 RESPONSE OF WAKE UP OPERATION	105
TABLE 155 FIELD DESCRIPTION OF WAKE UP OPERATION RESPONSE.....	10
TABLE 156 GET ID OPERATION.....	106
TABLE 157 FIELD DESCRIPTION OF TOPAZ TAG GET ID COMMAND	106
TABLE 158 RESPONSE OF GET ID OPERATION.....	106
TABLE 159 FIELD DESCRIPTION OF GET ID OPERATION RESPONSE.....	10
TABLE 160 READ OPERATION	107
TABLE 161 FIELD DESCRIPTION OF READ OPERATION.....	107
TABLE 162 RESPONSE OF READ OPERATION	107
TABLE 163 FIELD DESCRIPTION OF READ OPERATION RESPONSE.....	10
TABLE 164 WRITE OPERATION.....	108
TABLE 165 FIELD DESCRIPTION OF WRITE OPERATION	108
TABLE 166 RESPONSE OF WRITE OPERATION	108
TABLE 167 FIELD DESCRIPTION OF WRITE OPERATION RESPONSE.....	10
TABLE 168 ST TAG INVENTORY COMMAND.....	109
TABLE 169 FIELD DESCRIPTION OF ST TAG INVENTORY COMMAND	109
TABLE 170 RESPONSE OF ST TAG INVENTORY COMMAND.....	109
TABLE 171 FIELD DESCRIPTION OF ST TAG INVENTORY RESPONSE	109
TABLE 172 ST TAG ACCESS COMMAND.....	110
TABLE 173 READ OPERATION	110
TABLE 174 FIELD DESCRIPTION OF READ OPERATION.....	110
TABLE 175 RESPONSE OF READ OPERATION	111
TABLE 176 FIELD DESCRIPTION OF READ OPERATION RESPONSE.....	11
TABLE 177 WRITE OPERATION.....	112
TABLE 178 FIELD DESCRIPTION OF WRITE OPERATION	112
TABLE 179 RESPONSE OF WRITE OPERATION	112
TABLE 180 FIELD DESCRIPTION OF WRITE OPERATION RESPONSE.....	11
TABLE 181 COMPLETION OPERATION	113
TABLE 182 FIELD DESCRIPTION OF COMPLETION OPERATION.....	113
TABLE 183 RESPONSE OF COMPLETION OPERATION	113
TABLE 184 FIELD DESCRIPTION OF COMPLETION OPERATION RESPONSE.....	11
TABLE 185 ISO14443B TAG INVENTORY COMMAND	114
TABLE 186 FIELD DESCRIPTION OF ISO14443B TAG INVENTORY COMMAND.....	114
TABLE 187 RESPONSE OF ISO14443B TAG INVENTORY COMMAND	114
TABLE 188 FIELD DESCRIPTION OF ISO14443B TAG INVENTORY RESPONSE.....	114
TABLE 189 ISO14443B TAG ACCESS COMMAND	115

RR1X Series Programming Guide

TABLE 190 ATTRIBUTE OPERATION.....	115
TABLE 191 FIELD DESCRIPTION OF ATTRIBUTE OPERATION	115
TABLE 192 RESPONSE OF ATTRIBUTE OPERATION.....	116
TABLE 193 FIELD DESCRIPTION OF ATTRIBUTE OPERATION RESPONSE	11
TABLE 194 READ OPERATION	117
TABLE 195 FIELD DESCRIPTION OF READ OPERATION.....	117
TABLE 196 RESPONSE OF READ OPERATION	118
TABLE 197 FIELD DESCRIPTION OF READ OPERATION RESPONSE.....	11
TABLE 198 WRITE OPERATION.....	119
TABLE 199 FIELD DESCRIPTION OF WRITE OPERATION	119
TABLE 200 RESPONSE OF WRITE OPERATION	119
TABLE 201 FIELD DESCRIPTION OF WRITE OPERATION RESPONSE	11
TABLE 202 LOCK OPERATION	120
TABLE 203 FIELD DESCRIPTION OF LOCK OPERATION.....	120
TABLE 204 RESPONSE OF LOCK OPERATION	120
TABLE 205 FIELD DESCRIPTION OF LOCK OPERATION RESPONSE.....	12
TABLE 206 CHECK PASSWORD OPERATION	121
TABLE 207 FIELD DESCRIPTION OF CHECK PASSWORD OPERATION	121
TABLE 208 RESPONSE OF CHECK PASSWORD OPERATION.....	121
TABLE 209 FIELD DESCRIPTION OF CHECK PASSWORD OPERATION RESPONSE	12
TABLE 210 DESELECT OPERATION.....	122
TABLE 211 FIELD DESCRIPTION OF DESELECT OPERATION.....	122
TABLE 212 RESPONSE OF DESELECT OPERATION	122
TABLE 213 FIELD DESCRIPTION OF DESELECT OPERATION RESPONSE	12
TABLE 214 COUNT OPERATION	123
TABLE 215 FIELD DESCRIPTION OF COUNT OPERATION.....	123
TABLE 216 RESPONSE OF COUNT OPERATION	123
TABLE 217 FIELD DESCRIPTION OF COUNT OPERATION RESPONSE.....	12
TABLE 218 FELICA TAG INVENTORY COMMAND.....	124
TABLE 219 FIELD DESCRIPTION OF FELICA TAG INVENTORY COMMAND.....	124
TABLE 220 RESPONSE OF FELICA TAG INVENTORY COMMAND.....	124
TABLE 221 FIELD DESCRIPTION OF FELICA TAG INVENTORY RESPONSE.....	124
TABLE 222 DATA EXCHANGE OPERATION	127
TABLE 223 FIELD DESCRIPTION OF DATA EXCHANGE OPERATION	127
TABLE 224 RESPONSE OF DATA EXCHANGE OPERATION	127
TABLE 225 FIELD DESCRIPTION OF DATA EXCHANGE OPERATION RESPONSE	12
TABLE 226 ISO18092 INITIATOR COMMAND.....	128
TABLE 227 FIND TARGET OPERATION	128

RR1X Series Programming Guide

TABLE 228 FIELD DESCRIPTION OF FIND TARGET OPERATION.....	128
TABLE 229 RESPONSE OF FIND TARGET OPERATION	129
TABLE 230 FIELD DESCRIPTION OF FIND TARGET OPERATION RESPONSE.....	12
TABLE 231 DATA EXCHANGE OPERATION	130
TABLE 232 FIELD DESCRIPTION OF DATA EXCHANGE OPERATION	130
TABLE 233 RESPONSE OF DATA EXCHANGE OPERATION	130
TABLE 234 FIELD DESCRIPTION OF DATA EXCHANGE OPERATION RESPONSE.....	130

RR1X Series Programming Guide

1. Introduction

This document describes the APIs, host command format, host command flow, and In-System-Programming. For application developers, chapter 3 gives you the details about APIs.

For developers, who want to develop its own DLL or library, chapter 4 gives you the detail format of commands and responses. Besides, chapter 5 provides the detail about In-System-Programming.

2. Software Architecture

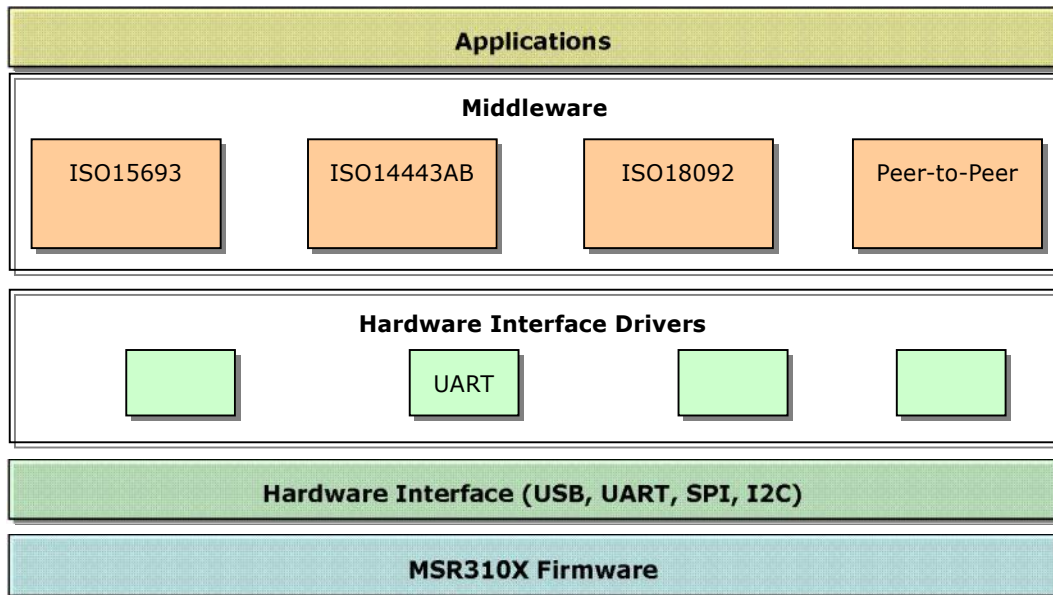


Figure 1 Software Architecture

RR1X Series Programming Guide

3. Application Programming Interface

We provided dynamic link libraries for Windows XP, Windows Vista and WinCE 5.0.

3.1 Win32

3.1.1 Management

3.1.1.1 Connect

Prototype	int Connect(int type, int port, int baudrate)			
Description	Connect to Reader. If connect success, Return handle.			
Parameter	Name	Len (byte)	Direction	Description
	type	1	Input	0: USB 1:Uart
	port	1	Input	Connection port
	baudrate	1	Input	If selected Uart Interface, this parameter shall be setting. Otherwise this parameter set to the "0".
Return Value	>=0: Connect successful, this value is handle -1: The port is already opened -2: The port can not be opened			

3.1.1.2 Disconnect

Prototype	int Disconnect(int handle)			
Description	disconnect Reader			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
Return Value	1: Success -99: Error Handle ID			

3.1.1.3 Get Firmware Version

Prototype	int GetVer(int handle, unsigned char *Ver)			
Description	Get Reader Version			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	*Ver	*	Output	Version Data
Return Value	>0: Version Length 0: Fail			

RR1X Series Programming Guide

3.1.1.4 Reader_Alive

Prototype	int Reader_Alive(int handle)			
Description	Check Reader is alive and return IC's Model ID.			
Parameter	Name	Len	Direction	Description
	Handle	1	Input	Handle ID
Return Value	Return Model ID: 0: MSR3100 1: MSR3101 2: MSR3102 -1: Can't connect to reader			

RR1X Series Programming Guide

3.1.2 ISO15693

3.1.2.1 Inventory

Prototype	int S693Get_UID(int handle, int AFIE NB ,unsigned char AFI, unsigned char *TagSUID)			
Description	Search ISO15693 Tags			
Parameter	Name	Len	Direction	Description
	handle	2	Input	Handle ID
	AFIE NB	1	Input	<ul style="list-style-type: none"> AFI Enable: 0x00 – AFI Flag not set 0x01 – AFI Flag set
	AFI	1	Input	AFI Value
	*TagSUID	8 * Tag counts	Output	Tags data report
Return Value	Tag counts			

3.1.2.2 Stay Quiet

Prototype	int S693StayQuiet(int handle, unsigned char *uid)			
Description	The tag enters the quiet state and doesn't send back response			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	*uid	8	Input	Tag UID
Return Value	0: Success -1: Fail			

3.1.2.3 Read Single Block

Prototype	int S693GetOneBlock(int handle, unsigned char *uid, int BytePerBlock, int BLKNO, unsigned char *rpt, unsigned char *BlkStatus)			
Description	The tag will report the block data			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	*uid	8	Input	Tag UID
	BytePerBlock	1	Input	Block size in bytes
	BLKNO	1	Input	Block ID that want to read
	*rpt	Block size	Output	Tag block data report.
	*BlkStatus	1	Output	Block security status. "0": Not locked, "1": Locked
Return Value	0: Success Other value: Error code			

RR1X Series Programming Guide

3.1.2.4 Write Single Block

Prototype	int S693WriteOneBlock(int handle, unsigned char *uid, int BytePerBlock, int BLKNO, unsigned char *Data, unsigned char OptFlag)			
Description	Write data to the block ID			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	*uid	8	Input	Tag UID
	BytePerBlock	1	Input	Block size in bytes
	BLKNO	1	Input	Block ID
	*Data	Block size	Input	Block data that want to write
	OptFlag	1	Input	If the Option_flag is set, the value is 1, otherwise is 0.
Return Value	0: Success Other value: Error code			

3.1.2.5 Lock Block

Prototype	int S693LockBlk(int handle, unsigned char *uid, int BlkNo, unsigned char OptFlag)			
Description	This function will lock permanently the requested block			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	*uid	8	Input	Tag UID
	BLKNO	1	Input	Block ID that want to lock
	OptFlag	1	Input	If the Option_flag is set, the value is 1, otherwise is 0.
Return Value	0: Success Other value: Error code			

RR1X Series Programming Guide

3.1.2.6 Read Multiple Blocks

Prototype	int S693ReadMultiBlk(int handle, unsigned char *uid, int StartBlk, int BlkNum, unsigned char *BlkData, unsigned char OptFlag))			
Description	The tag will report the block(s) data. Maximum 5 Blocks for each Reading.			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	*uid	8	Input	Tag UID
	StartBlk	1	Input	First block ID
	BlkNum	1	Input	Number of blocks that want to read
	*BlkData	BlkNum * Block size	Output	Tag block data report
	OptFlag	1	Input	If the Option_flag is set, the value is 1, otherwise is 0.
Return Value	0: Success Other value: Error code			

3.1.2.7 Select

Prototype	int S693Select(int handle, unsigned char *uid)			
Description	The tag enters the select state			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	*uid	8	Input	Tag UID
Return Value	0: Success Other value: Error code			

3.1.2.8 Reset to Ready

Prototype	int S693ResetToReady(int handle, unsigned char *uid)			
Description	The tag enters the ready state			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	*uid	8	Input	Tag UID
Return Value	0: Success Other value: Error code			

RR1X Series Programming Guide

3.1.2.9 Write AFI

Prototype	int S693WriteAFI(int handle, unsigned char *uid, unsigned char AFI, unsigned char OptFlag)			
Description	Write AFI value into the tag memory			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	*uid	8	Input	Tag UID
	AFI	1	Input	AFI data
	OptFlag	1	Input	If the Option_flag is set, the value is 1, otherwise is 0.
Return Value	0: Success Other value: Error code			

3.1.2.10 Lock AFI

Prototype	int S693LockAFI(int handle, unsigned char *uid, unsigned char OptFlag)			
Description	This function will lock permanently the AFI			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	*uid	8	Input	Tag UID
	OptFlag	1	Input	If the Option_flag is set, the value is 1, otherwise is 0.
Return Value	0: Success Other value: Error code			

3.1.2.11 Write DSFID

Prototype	int S693WriteDSF(int handle, unsigned char *uid, unsigned char DSFID, unsigned char OptFlag)			
Description	Write DSFID value into the tag memory			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	*uid	8	Input	Tag UID
	DSFID	1	Input	DSFID data
	OptFlag	1	Input	If the Option_flag is set, the value is 1, otherwise is 0.
Return Value	0: Success Other value: Error code			

RR1X Series Programming Guide

3.1.2.12 Lock DSFID

Prototype	int S693LockDSF(int handle, unsigned char *uid, unsigned char OptFlag)			
Description	This function will lock permanently the DSFID			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	*uid	8	Input	Tag UID
	OptFlag	1	Input	If the Option_flag is set, the value is 1, otherwise is 0.
Return Value	0: Success Other value: Error code			

3.1.2.13 Get System Information

Prototype	int S693GetSysInf(int handle, unsigned char *uid, int *BytePerBlk, int *BlkSize, unsigned char *AFI, unsigned char *DSFID, unsigned char *Flag)			
Description	Get System Information			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	*uid	8	Input	Tag UID
	*BytePerBlk	1	Output	Block size in bytes
	*BlkSize	1	Output	Number of blocks
	*AFI	1	Output	AFI value
	*DSFID	1	Output	DSFID value
	*Flag	1	Output	Information flags
Return Value	0: Success Other value: Error code			

RR1X Series Programming Guide

3.1.2.14 Get Multiple Block Security Status

Prototype	int S693BlkLockInf(int handle, unsigned char *uid, int StartBlk, int BlkNum, unsigned char *BlkLockInf)			
Description	Get multiple block security status			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	*uid	8	Input	Tag UID
	StartBlk	1	Input	First block ID
	BlkNum	1	Input	Number of blocks that want to read
	*BlkLockInf	1	Output	Block security status. “0”: Not locked, “1”: Locked
Return Value	0: Success Other value: Error code			

3.1.2.15 Read EAS

Prototype	int S693ReadEAS(int handle, unsigned char *uid, unsigned char *EAS)			
Description	Reader uses this command to get 1 bit EAS data from tag.			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	*uid	8	Input	Tag UID
	*EAS	1	Output	EAS data (0x00 or 0x01)
Return Value	0: Success Other value: Error code			

3.1.2.16 Write EAS

Prototype	int S693WriteEAS(int handle, unsigned char *uid, unsigned char EAS)			
Description	Reader uses this command to write EAS data to tag.			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	*uid	8	Input	Tag UID
	EAS	1	Input	EAS data (0x00 or 0x01)
Return Value	0: Success Other value: Error code			

RR1X Series Programming Guide

3.1.2.17 Lock EAS

Prototype	int S693LockEAS(int handle, unsigned char *uid)			
Description	Reader uses this command to Lock EAS. After EAS has been lock, EAS value can not be changed.			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	*uid	8	Input	Tag UID
Return Value	0: Success Other value: Error code			

3.1.2.18 Secure Read

Prototype	int S693SecReadBlk(int handle, unsigned char *uid, int BytePerBlock, int BLKNO, unsigned char *pwd, int PwdLen, unsigned char *rpt, unsigned char *BlkStatus)			
Description	When password is not default '00000000' and has been written into MSR3200, reader can not get block data by 'Read single command' or 'Read multiple blocks' command. RFID Reader uses this command to read block data if password in command is coincident with password stored in MSR3200.			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	*uid	8	Input	Tag UID
	BytePerBlock	1	Input	Block size in bytes
	BLKNO	1	Input	Block ID that want to read
	*pwd	PwdLen	Input	Password
	PwdLen	1	Input	Password length. If Key Enable, the length is 4 (32 bits), else is 8 (64bits).
	*rpt	Block size	Output	Tag block data report.
	*BlkStatus	1	Output	Block security status. "0": Not locked, "1": Locked
Return Value	0: Success Other value: Error code			

RR1X Series Programming Guide

3.1.2.19 Secure Write

Prototype	int S693SecWriteBlock(int handle, unsigned char *uid, int BytePerBlock, int BLKNO , unsigned char *pwd, int PwdLen, unsigned char *Data)			
Description	When password has been written into MSR3200, reader can not write data into memory by ‘Write single block’ command. RFID Reader uses this command to write data into user data block if password in command is coincident with password stored in MSR3200.			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	*uid	8	Input	Tag UID
	BytePerBlock	1	Input	Block size in bytes
	BLKNO	1	Input	Block ID that want to write
	*pwd	PwdLen	Input	Password
	PwdLen	1	Input	Password length. If Key Enable, the length is 4 (32 bits), else is 8 (64bits).
	*Data	Block size	Input	Block data that want to write
Return Value	0: Success Other value: Error code			

3.1.2.20 Get Owner ID

Prototype	int S693GetOID(int handle, unsigned char *uid, unsigned char *rpt)			
Description	There is 32 bits factory programmed ID reserved for MStar customers. Reader can get this 32 bits ID for some applications. For example, this ID can be one server IP address on the internet. After reader get this IP address and UID, the application software can hyperlink to server and verify it, or download/upload data between server and tag.			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	*uid	8	Input	Tag UID
	*rpt	4	Output	32 bits Owner ID

RR1X Series Programming Guide

3.1.2.21 Write PWD

Prototype	int S693WritePWD(int handle, unsigned char *uid, int BytePerBlock, int PwdLen, unsigned char *Data)			
Description	User can write password into chip by reader if password has not been locked. After password has been written into MSR3200, reader has to use 'Secure Read' to get data or 'Secure Write' to write data.			
Parameter	Name	Len	Direction	Description
	Handle	1	Input	Handle ID
	*uid	8	Input	Tag UID
	BytePerBlock	1	Input	Block size in bytes
	PwdLen	1	Input	Password length. If Key Enable, the length is 4, else is 8.
	*Data	PwdLen	Input	Password that want to write
Return Value	0: Success Other value: Error code			

3.1.2.22 Lock PWD

Prototype	int S693LockPWD(int handle, unsigned char *uid)			
Description	User can lock password by reader. After password has been locked, reader can not write password into MSR3200 again. Because there is no other command to unlock, user has to very carefully using this command unless password has been proved successfully written.			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	*uid	8	Input	Tag UID
Return Value	0: Success Other value: Error code			

RR1X Series Programming Guide

3.1.2.23 Read Key

Prototype	int S693ReadKey(int handle, unsigned char *uid, unsigned char *rpt)			
Description	User can read key from reader if key is enabled.			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	*uid	8	Input	Tag UID
	*rpt	4	Output	32 bits key

3.1.2.24 Write Key

Prototype	int S693WriteKey(int handle, unsigned char *uid, int BytePerBlock, unsigned char *Data)			
Description	User can write key into chip by reader if key is enabled.			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	*uid	8	Input	Tag UID
	BytePerBlock	1	Input	Block size in bytes
	*Data	4	Input	32 bits key

3.1.2.25 Lock Key

Prototype	int S693LockKey(int handle, unsigned char *uid)			
Description	User can lock key by reader. After key has been locked, reader can not write key into MSR3200 again. Because there is no other command to unlock, user has to very carefully using this command unless key has been proved successfully written.			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	*uid	8	Input	Tag UID
Return Value	0: Success Other value: Error code			

RR1X Series Programming Guide

3.1.2.26 Read Register

Prototype	int S693ReadReg(int handle, unsigned char *uid, int BLKNO, unsigned char *rpt)			
Description	User can read register from reader.			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	*uid	8	Input	Tag UID
	BLKNO	1	Input	Block Number (0 or 1)
	*rpt	4	Output	32 bits data

3.1.2.27 Write Register

Prototype	int S693WriteReg(int handle, unsigned char *uid, int BlkNo, unsigned char *Data)			
Description	User can write register into chip by reader.			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	*uid	8	Input	Tag UID
	BLKNO	1	Input	Block Number (0 or 1)
	*Data	4	Input	32 bits data

3.1.2.28 Write PWD with PWD

Prototype	int S693WritePWD_WithPWD (int handle, unsigned char *uid, int BytePerBlock, int PwdLen, unsigned char *NewPWD, unsigned char *OldPWD)			
Description	User can write password with old-password into chip by reader if password has not been locked. After password has been written into MSR3200, reader has to use 'Secure Read' to get data or 'Secure Write' to write data.			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	*uid	8	Input	Tag UID
	BytePerBlock	1	Input	Block size in bytes
	PwdLen	1	Input	Password length. If Key Enable, the length is 4, else is 8.
	*NewPWD	PwdLen	Input	New Password that want to write
	*OldPWD	PwdLen	Input	Old Password
Return Value	0: Success Other value: Error code			

RR1X Series Programming Guide

3.1.2.29 Fast Inventory- Only for Fujitsu Tag

Prototype	int S693FastGet_UID (int handle, int AFIE NB ,unsigned char AFI, unsigned char *TagSUID)			
Description	Search Fujitsu ISO15693 Tags. This command only for Fujitsu tag			
Parameter	Name	Len	Direction	Description
	handle	2	Input	Handle ID
	AFIE NB	1	Input	<ul style="list-style-type: none"> ■ AFI Enable: 0x00 – AFI Flag not set ■ 0x01 – AFI Flag set
	AFI	1	Input	AFI Value
	*TagSUID	8 * Tag counts	Output	Tags data report
Return Value	Tag counts			

3.1.2.30 Fast Read Single Block- Only for Fujitsu Tag

Prototype	int S693FastReadBlk (int handle, unsigned char *uid, int BytePerBlock, int BLKNO, unsigned char *rpt, unsigned char *BlkStatus)			
Description	The tag will report the block data. This command only for Fujitsu tag			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	*uid	8	Input	Tag UID
	BytePerBlock	1	Input	Block size in bytes
	BLKNO	1	Input	Block ID that want to read
	*rpt	Block size	Output	Tag block data report.
	*BlkStatus	1	Output	Block security status. “0”: Not locked, “1”: Locked
Return Value	0: Success Other value: Error code			

RR1X Series Programming Guide

3.1.2.31 Fast Write Single Block- Only for Fujitsu Tag

Prototype	int S693FastWriteOneBlock (int handle, unsigned char *uid, int BytePerBlock, int BLKNO, unsigned char *Data, unsigned char OptFlag)			
Description	Write data to the block ID. This command only for Fujitsu tag.			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	*uid	8	Input	Tag UID
	BytePerBlock	1	Input	Block size in bytes
	BLKNO	1	Input	Block ID
	*Data	Block size	Input	Block data that want to write
	OptFlag	1	Input	If the Option_flag is set, the value is 1, otherwise is 0.
Return Value	0: Success Other value: Error code			

3.1.2.32 Get ISO15693 DLL Version

Prototype	int Get_ISO15693_DLL_Ver (unsigned char **Ver)			
Description	Get version number and compile date/time information of ISO15693 DLL.			
Parameter	Name	Len	Direction	Description
	**Ver		Output	Pointer to return a buffer pointer.
Return Value	Buffer length			

RR1X Series Programming Guide

3.1.3 ISO14443

3.1.3.1 ISO14443A, Get UID

Prototype	int S43AGetUID(int handle, unsigned char *TagNo, unsigned char *UIDlength, unsigned char *UIDS)			
Description	Get tag UID of ISO-14443A.			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	*TagNo	1	Output	To save found tag amount.
	*UIDlength	1 * Tag counts	Output	To save tag UID length.
	*UIDS	UIDlength * Tag counts	Output	To save tag UID.
Return Value	Return error code: 0: Get UID successful -1: Fail			

3.1.3.2 Mifare Ultralight, Read

Prototype	int UL_READ(int handle, unsigned char ADDR, unsigned char *RES_DATA)			
Description	Read tag block data of Mifare UltraLight. It will response 4 block data from ADDR.			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	ADDR	1	Input	Block address that user want to start read.
	*RES_DATA	16	Output	Tag 4 block data. (1 block has 4 bytes data)
Return Value	Return error code: Data length: Response data length -1: Interface Fail -2: Read Fail			

RR1X Series Programming Guide

3.1.3.3 Mifare Ultralight, Write

Prototype	int UL_WRITE(int handle, unsigned char ADDR, unsigned char *WR_DATA, unsigned char *RES_DATA)			
Description	Write data into Mifare UltraLight tag block.			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	ADDR	1	Input	Block address that user want to write.
	*WR_DATA	4	Input	Block data that user want to write.
	*RES_DATA		Output	Tag response. (AK or NAK)
Return Value	Return error code: Data length: Response data length -1: Interface Fail -2: Write Fail			

3.1.3.4 TOPAZ, Wake up

Prototype	int TOPAZ_REQ(int handle, unsigned char *RES_DATA)			
Description	WUPA command for TOPAZ. It the same as standard WUPA command.			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	*RES_DATA	2	Output	ATQA
Return Value	Return error code: Data length: Response data length -1: Interface Fail -2: WUPA Fail			

3.1.3.5 TOPAZ, Get ID

Prototype	int TOPAZ_RID(int handle, unsigned char *RES_DATA)			
Description	Read UID command for TOPAZ.			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	*RES_DATA	8	Output	Tag Response: HR0 HR1 UID0 UID1 UID2 UID3 CRC1 CRC2
Return Value	Return error code: Data length: Response data length -1: Interface Fail -2: RID Fail			

RR1X Series Programming Guide

3.1.3.6 TOPAZ, Read

Prototype	int TOPAZ_READ(int handle, unsigned char ADDR, unsigned char *UID, unsigned char *RES_DATA)			
Description	According to ADDR read 1 byte data from TOPAZ tag.			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	ADDR	1	Input	Block address that user want to read.
	*UID	4	Input	Tag's UID.
	*RES_DATA	4	Output	Tag Response: AD0 DATA CRC1 CRC2
Return Value	Return error code: Data length: Response data length -1: Interface Fail -2: READ Fail			

3.1.3.7 TOPAZ, Write

Prototype	int TOPAZ_WRITEE(int handle, UINT8 ADDR, UINT8 DATA, UINT8 *UID, UINT8 *RES_DATA)			
Description	According to ADDR write 1 byte data to TOPAZ tag.			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	ADDR	1	Input	Block address that user want to read.
	DATA	1	Input	1 byte data that user want to write.
	*UID	4	Input	Tag's UID.
	*RES_DATA	4	Output	Tag Response: AD0 DATA CRC1 CRC2
Return Value	Return error code: Data length: Response data length -1: Interface Fail -2: WRITE Fail			

RR1X Series Programming Guide

3.1.3.8 ISO14443B, Get PUPI

Prototype	int S43BGetUID_Std(int handle, unsigned char *TagNo, unsigned char *UID, unsigned char *CIDS, unsigned char OPE)			
Description	Get tag UID of ISO-14443B (For ISO standard).			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	*TagNo	1	Output	To save tag numbers that been found.
	*UID	8 * Tag counts	Output	To save tag UID.
	*CIDS	Tag counts	Output	Tag's Chip ID.
	OPE	1	Input	OPE = 0: Not to do ATTRIB command. OPE = 1: Do ATTRIB command
Return Value	Return error code: 0: Get UID successful -1: Fail			

3.1.3.9 ISO14443B, Attribute

Prototype	int ISO43B_ATTRIB(int handle, UINT8 CID, UINT8 *PUPI, UINT8 *RES_DATA)			
Description	ATTRIB command of ISO-14443B (For ISO standard).			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	CID	1	Input	Tag's Chip ID
	PUPI	4	Input	To save tag PUPI.
	*RES_DATA	3	Output	Tag response data. The first byte contains MBLI and CID.
Return Value	Return error code: 0: Get UID successful -1: Interface Fail -2: ATTRIB Fail			

RR1X Series Programming Guide

3.1.3.10 ST, Inventory

Prototype	int S43BGetUID(int handle, unsigned char *TagNo, unsigned char *UID)			
Description	Get tag UID of ISO-14443B (For ST SRI4K).			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	*TagNo	1	Output	To save tag numbers that been found.
	*UID	8 * Tag counts	Output	To save tag UID.
Return Value	Return error code: 0: Get UID successful -1: Fail			

3.1.3.11 ST, Read

Prototype	int S43BReadBlock(int handle, unsigned char *UID, unsigned char Addr, unsigned char *DATA)			
Description	Read tag block data of ISO14443B.			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	*UID	8	Input	Use to get it corresponding CID.
	Addr	1	Input	Block address that user want to read.
	*DATA	4	Output	To save block data.
Return Value	Return error code: 0: Success -1: Fail			

3.1.3.12 ST, Write

Prototype	int S43BWriteBlock(int handle, unsigned char *UID, unsigned char Addr, unsigned char *DATA)			
Description	Write tag block data of ISO14443B.			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	*UID	8	Input	Use to get it corresponding CID.
	Addr	1	Input	Block address that user want to write.
	*DATA	4	Input	Data that want to write.
Return Value	Return error code: 0: Success -1: Fail			

RR1X Series Programming Guide

3.1.3.13 ST, Completion

Prototype	int S43BCompletion(int handle, unsigned char *UID)			
Description	Let tag from SELECTED state to DEACTIVATED state			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	*UID	8	Input	Use to get it corresponding CID.
Return Value	Return error code: 0: Success -1: Fail			

3.1.3.14 ATMEL AT88RF020, Read

Prototype	int AT88RF020_READ(int handle, UINT8 CID, UINT8 ADDR, UINT8 *RES_DATA)			
Description	According to ADDR read 8 byte data from AT88RF020 tag.			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	CID	1	Input	Tag's Chip ID.
	ADDR	1	Input	Block address that user want to read.
	RES_DATA	12 or 4	Output	Tag response data. READ Success: CMD Addr DATA[8] CRCL CRCH NACK: CMD 0xX1 CRCL CRCH
Return Value	Return error code: Data length: Response data length -1: Interface Fail -2: READ Fail			

RR1X Series Programming Guide

3.1.3.15 ATMEL AT88RF020, Write

Prototype	int AT88RF020_WRITE(int handle, UINT8 CID, UINT8 ADDR, UINT8 *WR_DATA, UINT8 *RES_DATA)			
Description	According to ADDR write 8 byte data to AT88RF020 tag.			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	CID	1	Input	Tag's Chip ID.
	ADDR	1	Input	Block address that user want to write.
	*WR_DATA	8	Input	8 byte data that user want to write
	RES_DATA	4	Output	Tag response data. ACK: CMD 0xX0 CRCL CRCH NACK: CMD 0xX1 CRCL CRCH
Return Value	Return error code: Data length: Response data length -1: Interface Fail -2: WRITE Fail			

3.1.3.16 ATMEL AT88RF020, Lock

Prototype	int AT88RF020_LOCK(int handle, UINT8 CID, UINT8 *LCK_DATA, UINT8 *RES_DATA)			
Description	The LOCK command can be executed only after proper password validation has been performed. The last 31 bits of data within the LOCK command are logically ORed within the device with the 31-bit value stored within the LockBits field of Page 0. After the memory has been written, an ACK command will be transmitted. A NACK command is issued if the LOCK command is attempted without previous password validation.			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	CID	1	Input	Tag's Chip ID.
	RES_DATA	4	Output	Tag response data. ACK: CMD 0xX0 CRCL CRCH NACK: CMD 0xX1 CRCL CRCH
Return Value	Return error code: Data length: Response data length -1: Interface Fail -2: LOCK Fail			

RR1X Series Programming Guide

3.1.3.17 ATMEL AT88RF020, Check Password

Prototype	int AT88RF020_CHECK_PWD(int handle, UINT8 CID, UINT8 *PWD, UINT8 *RES_DATA)			
Description	CHECK PASSWORD command is compared to the password stored within Page 3 of the memory. If the input password matches the stored password, user can to access Page 3 and greater than Page3 memory.			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	CID	1	Input	Tag's Chip ID.
	*PWD	8	Input	8 byte password that user want to check.
	RES_DATA	4	Output	Tag response data. ACK: CMD 0xX0 CRCL CRCH NACK: CMD 0xX1 CRCL CRCH
Return Value	Return error code: Data length: Response data length -1: Interface Fail -2: Check Password Fail			

3.1.3.18 ATMEL AT88RF020, Deselect

Prototype	int AT88RF020_DESELECT(int handle, UINT8 CID, UINT8 *RES_DATA)			
Description	When tag state is ACTIVE and receive DESELECT command. Tag will response ACK and enter the HALT state.			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	CID	1	Input	Tag's Chip ID.
	RES_DATA	4	Output	Tag response data. ACK: CMD 0xX0 CRCL CRCH NACK: CMD 0xX1 CRCL CRCH
Return Value	Return error code: Data length: Response data length -1: Interface Fail -2: DESELECT Fail			

RR1X Series Programming Guide

3.1.3.19 ATMEL AT88RF020, Count

Prototype	int COUNT(int handle, UINT8 CID, UINT8 *CNT_DATA, UINT8 *RES_DATA)			
Description	<p>The COUNT command is used to write Page 2. The first six bytes sent by the PCD (referred to as the Signature) are written to the first six bytes of Page 2 unmodified. The last two bytes of data sent by the PCD in the COUNT command are only placeholders and will be ignored.</p> <p>The 16-bit value stored in the counter field of Page 2 is incremented by one each time COUNT is executed. Once the value of the counter reaches 0x8000, no further count operations will be executed, and Page 2 will be effectively locked against further modification.</p> <p>Password validation must occur before the COUNT command is permitted. The chip will compute the new incremented count that will be written into the last two bytes of Page 2 immediately following the incoming 6-byte data field. It is expected that at least part of the 6-byte value will be the result of an externally computed cryptographic operation on the new Counter value, thus permitting some degree of transaction validation.</p> <p>If the write cannot take place (because Counter has a value of 0x8000, Page 2 is locked or no password has been sent), then a NACK command will be issued by the PICC; otherwise, an ACK command will be transmitted after the write has completed.</p>			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	CID	1	Input	Tag's Chip ID.
	*CNT_DATA	6	Input	6 bytes data that user want to write in Page2 Byte0~Byte5.
	RES_DATA	4	Output	Tag response data. ACK: CMD 0xX0 CRCL CRCH NACK: CMD 0xX1 CRCL CRCH
Return Value	<p>Return error code:</p> <p>Data length: Response data length</p> <p>-1: Interface Fail</p> <p>-2: COUNT Fail</p>			

RR1X Series Programming Guide

3.1.3.20 Get ISO14443 DLL Version

Prototype	int Get_ISO14443_DLL_Ver(unsigned char **Ver)			
Description	Get version number and compile date/time information of ISO14443 DLL.			
Parameter	Name	Len	Direction	Description
	**Ver		Output	Pointer to return a buffer pointer.
Return Value	Buffer length			

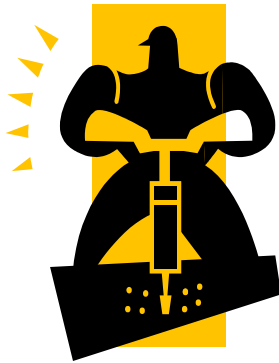
RR1X Series Programming Guide

3.1.4 Felica

3.1.4.1 Get Felica UID

Prototype	int GetUIDPas212Get_UID(int handle, unsigned char *TagSUID)			
Description	Get UID of Felica Tag.			
Parameter	Name	Len	Direction	Description
	handle	1	Input	Handle ID
	* TagSUID	8 * Tag counts	Output	To save tag UID.
Return Value	Tag counts			

3.1.5 ISO18092



RR1X Series Programming Guide

3.1.6 Mstar Peer-to-Peer

3.1.6.1 Send File

Prototype	int ____declspec(dllexport) ____stdcall M3X_Snd_File(int handle, UINT8 *file_names, m3xTxCB pCB);			
Description				
Parameter	Name	Len in byte	Direction	Description
	handle	4	input	Handle ID.
	file_names	*	Input	Filename array to be sent
	pCB	4	Output	Call Back Function pointer
Return Value	Return 0 for success. Return error code on fail.			

3.1.6.2 Receive File

Prototype	int ____declspec(dllexport) ____stdcall M3X_Rcv_File(int handle, UINT8 *path, m3xRxCB pCB);			
Description				
Parameter	Name	Len	Direction	Description
	handle	4	input	Interface handle returned from connect().
	path	*	Input	Filename array received
	pCB	4	Input	Call Back Function pointer
Return Value	Return 0 for success. Return error code on fail.			

3.1.6.3 Get P2P DLL Version

Prototype	int Get_XFile_DLL_Ver(unsigned char **Ver)			
Description	Get version number and compile date/time information of P2P DLL.			
Parameter	Name	Len	Direction	Description
	**Ver		Output	Pointer to return a buffer pointer.
Return Value	Buffer length			

RR1X Series Programming Guide

3.1.7 Secure

3.1.7.1 MD5 Hash Function

Prototype	void ____declspec(dllexport) Dll_MD5(unsigned char *input, int inputlen, unsigned char output[16])			
Description				
Parameter	Name	Len in byte	Direction	Description
	input	*	Input	Pointer to input data.
	inputlen	4	Input	Length of input data in bytes.
	output	16	Output	Output data array.
Return Value	N/A			

3.1.7.2 RSA Public/Private Key Pair Generator

Prototype	int ____declspec(dllexport) Dll_RSAGenKeyPair(int size, char *keyfilename)			
Description	If keyfilename = "1\0", two key files named 1_pub.key and 1_priv.key is generated.			
Parameter	Name	Len in byte	Direction	Description
	size	4	Input	Key size in bits (256/512).
	keyfilename	*	Input	Pointer to key file name (NULL terminated).
Return Value	Return 0 for success.			
	Return not 0 for fail.			

3.1.7.3 RSA Key File Setup Function

Prototype	int ____declspec(dllexport) Dll_RSASetKey(int mode, int size, char *keyfilename)			
Description	Setup RSA public/private key by corresponding key file. This should be called before using Dll_RSASetKey() and Dll_RSADecrypt().			
Parameter	Name	Len in byte	Direction	Description
	mode	4	Input	0 for public key, 1 for private key.
	size	4	Input	Key size in bits (256/512).
	keyfilename	*	Input	Pointer to key file name (NULL terminated).
Return Value	Return 0 for success.			
	Return not 0 for fail.			

RR1X Series Programming Guide

3.1.7.4 RSA Encryption Function

Prototype	int ____declspec(dllexport) Dll_RSACrypt(int mode, int inputlen, unsigned char *input, unsigned char *output)			
Description	Using RSA public or private key to encrypt input data.			
Parameter	Name	Len in byte	Direction	Description
	mode	4	Input	0 for using public key, 1 for using private key.
	inputlen	4	Input	Length of input data in bytes.
	input	*	Input	Pointer to input data.
	output	*	Output	Pointer to output data.
Return Value	Return 0 for success. Return not 0 for fail.			

3.1.7.5 RSA Decryption Function

Prototype	int ____declspec(dllexport) Dll_RSADecrypt(int mode, int *outputlen, unsigned char *input, unsigned char *output)			
Description	Using RSA public or private key to decrypt input data.			
Parameter	Name	Len in byte	Direction	Description
	mode	4	Input	0 for using public key, 1 for using private key.
	outputlen	*	Output	Pointer to length of output data in bytes.
	input	*	Input	Pointer to input data.
	output	*	Output	Pointer to output data.
Return Value	Return 0 for success. Return not 0 for fail.			

3.1.7.6 AES Key Setup Function

Prototype	void ____declspec(dllexport) Dll_AESSetKey(int mode, unsigned char *key, int keysize)			
Description	Setup AES key.			
Parameter	Name	Len in byte	Direction	Description
	mode	4	Input	0 for encryption, 1 for decryption.
	key	*	Input	Pointer to key content (128 bits).
	keysize	4	Input	Key size in bits (128).
Return Value	N/A			

RR1X Series Programming Guide

3.1.7.7 AES Encryption/Decryption Function

Prototype	void ____declspec(dllexport) Dll_AESCrypt(int mode, unsigned char input[16], unsigned char output[16])			
Description	AES ECB.			
Parameter	Name	Len in byte	Direction	Description
	mode	4	Input	0 for encryption, 1 for decryption.
	input	16	Input	Input data array.
	output	16	Output	Output data array.
Return Value	N/A			

3.1.7.8 Get Secure DLL Version

Prototype	int ____declspec(dllexport) ____stdcall Get_Secure_DLL_Ver(unsigned char **Ver)			
Description	Get version number and compile date/time information of Secure DLL.			
Parameter	Name	Len	Direction	Description
	**Ver		Output	Pointer to return a buffer pointer.
Return Value	Buffer length			

3.2 WinCE

The detail was provided in [MSR3100.chm](#).

RR1X Series Programming Guide

4. Host Commands and Responses

4.1 Communication Interface

RR1X provides four interfaces for connecting with host. Below sections describe the detail communication sequence for each interface.

RR1X Series Programming Guide

4.1.2 UART

RR1X use below UART settings for communication:

Baud Rate: 115200 bps
Parity: None
Data bits: 8
Stop bit: 1

Before host sends command to RR1X or RR1X sends response to host, a handshake sequence exists for synchronization purpose. For the detail, please refer to below sequence chart.

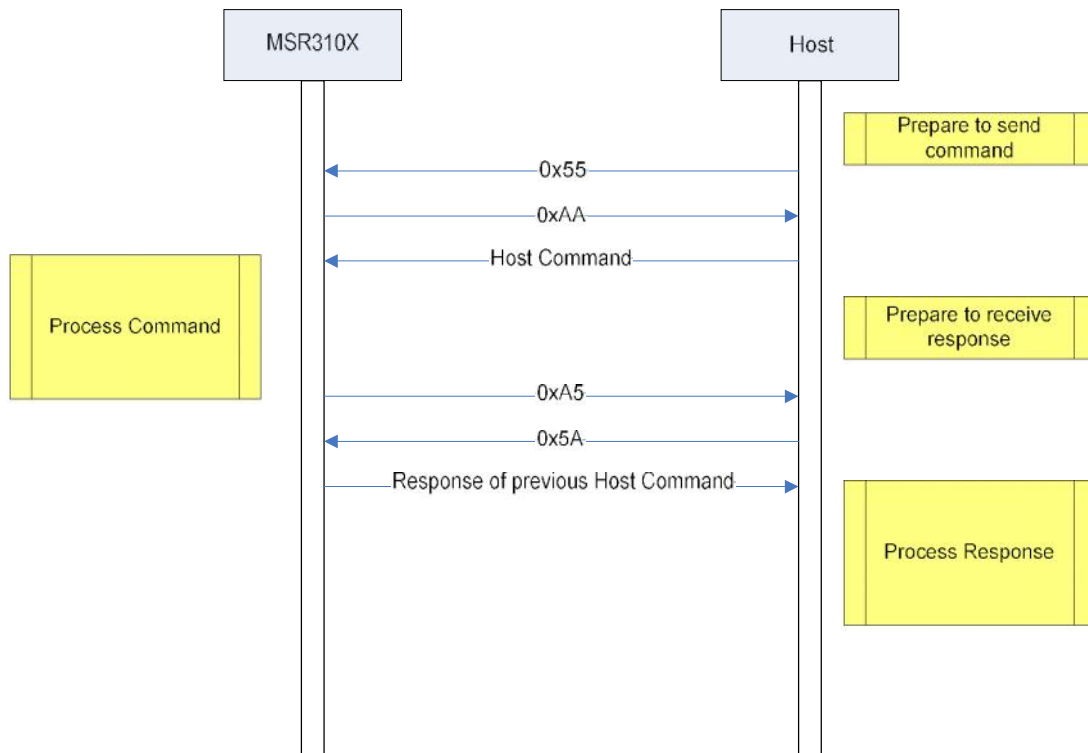


Figure 3 UART communication sequence

RR1X Series Programming Guide

4.2 Host Command and Response Format

Below table shows the generic format of host commands, those were sent from host to RR1X.

The command was sent by host from “Length” field.

Length	Command Code	Parameters / Response
--------	--------------	-----------------------

Table 1 Generic frame format of host command and target response

Field Name	Length (Byte)	Description
Length	1	The total length of command or response including self field.
Command Code	1	The commands supported by RR1X firmware. The detail definition was listed in section 4.3.
Parameters / Response	0~253	The content of parameters and response. The detail of each command was defined in 4.4.

Table 2 Field description of generic command / response format

RR1X Series Programming Guide

4.3 Host Command Code List

Command	Code	Description
Connection	0x01	Test connection and get chip model ID.
Select RF Parameter	0x02	Select and set RF parameters for specific RFID protocol.
Get Version	0x03	Get firmware version.
ISO15693 Tag Inventory	0x06	Inventory ISO15693 compatible tags.
ISO15693 Tag Access	0x07	Access tag's memory block and control tag's state transition.
Mifare Ultralight Tag Access	0x08	Access Mifare Ultralight tag, ISO14443A compatible.
ISO14443A Tag Inventory	0x09	Inventory ISO14443A compatible tags.
TOPAZ Tag Inventory and Access	0x0F	Inventory or access TOPAZ tag, ISO14443A compatible.
ST Tag Inventory	0x0A	Inventory ST tags, ISO14443B RF compatible.
ST Tag Access	0x0B	Access ST tags, ISO14443B RF compatible.
ISO14443B Tag Inventory	0x0C	Inventory ISO14443B tags.
ISO14443B Tag Access	0x0D	Access ISO14443B tags.
Felica Tag Inventory	0x0E	Inventory Felica tags.
Mstar P2P Tx	0x1E	Mstar proprietary Peer-to-Peer Tx transmission command
Mstar P2P Rx	0x1F	Mstar proprietary Peer-to-Peer Rx transmission command.
ISO18092 Target	0x20	Activate as target device, defined in ISO18092.
ISO18092 Initiator	0x21	Activate as initiator device, defined in ISO18092.

Table 3 Host Command List

RR1X Series Programming Guide

Below is the feature mapping for different chip model. Supported feature was marked by character 'V'.

Command	MSR3100	MSR3101	MSR3102
Connection	V	V	V
Select RF Parameter	V	V	V
Get Version	V	V	V
Memory Access	V	V	V
ISO15693 Tag Inventory	V	V	V
ISO15693 Tag Access	V	V	V
Mifare Ultralight Tag Access	V	V	V
ISO14443A Tag Inventory	V	V	V
TOPAZ Tag Inventory and Access	V	V	V
ST Tag Inventory	V	V	
ST Tag Access	V	V	
ISO14443B Tag Inventory	V	V	
ISO14443B Tag Access	V	V	
Felica Tag Inventory	V	V	
Mstar P2P Tx	V		
Mstar P2P Rx	V		
ISO18092 Target	V		
ISO18092 Initiator	V		

Table 4 Feature mapping for RR1X models

RR1X Series Programming Guide

4.4 Host Commands and Responses

The definition of “length” and “command” fields was described in section 4.2. Below sections will focus on descriptions of “parameter / response” field.

4.4.1 Connection

Connection command was used to test the physical link between host and RR1X. Besides, host can recognize RR1X chip mode according to the model ID in response data.

Connection command was specified as below.

Length	Command	Reserve 1	Reserve 2	Checksum
0x06	0x01	0x05	0x0A	

Table 5 Connection command

Field Name	Length (Byte)	Description
Reserve 1	1	It was reserved for future use and was fixed to 0x05 now.
Reserve 2	1	It was reserved for future use and was fixed to 0x0A now.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 6 Field description of connection command

The response of connection command was specified as below.

Length	Command	Reserve 1	Model ID	Checksum
0x06	0x01	0x05		

Table 7 Response of connection command

Field Name	Length (Byte)	Description
Reserve 1	1	It was reserved for future use and was fixed to 0x05 now.
Model ID	1	0x00: MSR3100 0x01: MSR3101 0x02: MSR3102 Others: Reserved
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 8 Field description of connection response

RR1X Series Programming Guide

4.4.2 Select RF Parameter

Select RF parameter command was used to choose RF parameters for different requirement. This may be not necessary, because the RF parameters will be chosen automatically when host request to execute RFID protocol related commands, ex: ISO1569: Tag Inventory...

Select RF parameter command was specified as below.

Length	Command	RF Mode	Checksum
0x05	0x02	As below	

Table 9 Select RF parameter command

Field Name	Length (Byte)	Description
RF Mode	1	0x00: Turn off RF power 0x01: ISO15693 Reader 0x02: ISO14443A Reader 0x03: ISO14443B Reader 0x04: Initiator, Passive, 106 kbps 0x05: Target, Passive, 106 kbps 0x06: Initiator, Passive, 212 kbps 0x07: Target, Passive, 212 kbps 0x08: Initiator, Passive, 424 kbps 0x09: Target, Passive, 424 kbps 0x0A: Initiator, Passive, 848 kbps 0x0B: Target, Passive, 848 kbps 0x0C: Initiator, Active, 106 kbps 0x0D: Target, Active, 106 kbps 0x0E: Initiator, Active, 212 kbps 0x0F: Target, Active, 212 kbps 0x10: Initiator, Active, 424 kbps 0x11: Target, Active, 424 kbps 0x12: Initiator, Active, 848 kbps 0x13: Target, Active, 848 kbps Others: Reserved
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 10 Field description of select RF parameter command

RR1X Series Programming Guide

The response of select RF parameter command was specified as below.

Length	Command	RF Mode	Reserve	Checksum
0x06	0x02	As below	0x00	

Table 11 Response of select RF parameter command

Field Name	Length (Byte)	Description
RF Mode	1	Defined in Table 10.
Reserve	1	It was reserved for future use and was fixed to 0x00 now.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 12 Field description of select RF parameter response

RR1X Series Programming Guide

4.4.3 Get Version

Get version command was used to get firmware version and the information of release date and time.

Get version command was specified as below.

Length	Command	Reserve	Checksum
0x05	0x03	0x00	

Table 13 Get version command

Field Name	Length (Byte)	Description
Reserve	1	It was reserved for future use and was fixed to 0x00 now.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 14 Field description of get version command

RR1X Series Programming Guide

The response of get version command was specified as below.

Length	Command	Major	Minor	Reserve 1	Date	Time	Reserve 2	Reserve 3	Checksum
0x0F	0x03			0x00			0x01	0x01	

Table 15 Response of get version command

Field Name	Length (Byte)	Description
Major	1	Major number of firmware version code.
Minor	1	Minor number of firmware version code.
Reserve 1	1	It was reserved for future use and was fixed to 0x00 now
Date	3	Date[0] store information of month. Date[1] store information of date. Date[2] store information of year. Ex: Date[0] = 6, Date[1] = 23, Date[2] = 8 It means 6/23/2008.
Time	3	Time[0] store information of hour. Time[1] store information of minute. Time[2] store information of second. Ex: Date[0] = 23, Date[1] = 29, Date[2] = 33 It means 23:29:33
Reserve 2	1	It was reserved for future use and was fixed to 0x01 now.
Reserve 3	1	It was reserved for future use and was fixed to 0x01 now.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 16 Field description of get version response

4.4.4 ISO15693 Tag Inventory

ISO15693 Tag Inventory command was used to find tags appeared in RF field.

ISO15693 Tag Inventory command was specified as below.

Length	Command	Enable Fast Mode	Enable AFI	AFI	Checksum
0x07	0x06				

Table 17 ISO15693 Tag Inventory command

Field Name	Length (Byte)	Description
Enable Fast Mode	1	0x00: Normal mode. 0x01: Fast mode. Others: Reserved.
Enable AFI	1	0x00: Do not assign AFI, the AFI filed will be ignored by firmware. 0x01: Assign AFI. Others: Reserved.
AFI	1	Application Family Identifier. The coding is listed in Figure 3.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 18 Field description of ISO15693 Tag Inventory command

RR1X Series Programming Guide

The response of ISO15693 tag inventory command was specified as below.

Length	Command	Tag No.	Reserve	UIDs	Checksum
N	0x06				

Table 19 Response of ISO15693 Tag Inventory command

Field Name	Length (Byte)	Description
Length	1	0x05: No tag be found. “Tag No.” = 0. 0x0E: Found one tag. “Tag No.” = 1. 0x16: Found two tags. “Tag No.” = 2. 0x1E: Found three tags. “Tag No.” = 3. 0x26: Found four tags. “Tag No.” = 4.
Tag No.	1	0x00 ~ 0x04 Others: Reserved.
Reserve	1	Reserved. Don’t care.
UIDs	8* Tag No.	Tags’ UID.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 20 Field description of ISO15693 Tag Inventory response

RR1X Series Programming Guide

AFI most significant nibble	AFI least significant nibble	Meaning VICCs respond from	Examples / note
'0'	'0'	All families and sub-families	No applicative preselection
X	'0'	All sub-families of family X	Wide applicative preselection
X	Y	Only the Yth sub-family of family X	
'0'	Y	Proprietary sub-family Y only	
'1'	'0', Y	Transport	Mass transit, Bus, Airline
'2'	'0', Y	Financial	IEP, Banking, Retail
'3'	'0', Y	Identification	Access control
'4'	'0', Y	Telecommunication	Public telephony, GSM
'5'	'0', Y	Medical	
'6'	'0', Y	Multimedia	Internet services
'7'	'0', Y	Gaming	
'8'	'0', Y	Data storage	Portable files
'9'	'0', Y	Item management	
'A'	'0', Y	Express parcels	
'B'	'0', Y	Postal services	
'C'	'0', Y	Airline bags	
'D'	'0', Y		
'E'	'0', Y		
'F'	'0', Y		

NOTE X = '1' to 'F', Y = '1' to 'F'

The support of AFI by the VICC is optional.

Figure 9 AFI coding

RR1X Series Programming Guide

4.4.5 ISO15693 Tag Access

ISO15693 tag access command was used to access tags memory and control tag's state transition.

ISO15693 tag access command was specified as below. This is a generic format. For each operation, "Parameters" field has different format. The detail was described in below sections.

Length	Command	Operation	Parameters	Checksum
	0x07			

Table 21 ISO15693 tag access command

4.4.5.1 Stay Quiet

Stay Quiet operation will put specific tag into quiet state. The detail command format was specified as below.

Length	Command	Operation	UID	Checksum
0x0D	0x07	0x04	8 bytes UID	

Table 22 Stay Quiet operation

Field Name	Length (Byte)	Description
UID	8	Tag's UID.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 23 Field description of Stay Quiet operation

The response of Stay Quiet operation was specified as below.

Length	Command	Operation	Reserve	Checksum
0x06	0x07	0x04	0x00	

Table 24 Response of Stay Quiet operation

Field Name	Length (Byte)	Description
Reserve	1	It was reserved for future use and was fixed to 0x00 now.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 25 Field description of Stay Quiet operation

RR1X Series Programming Guide

4.4.5.2 Read Single Block

Read Single Block operation was used to read one block data from specific block. The detail command format was specified as below.

Length	Command	Operation	UID	Option	Start Block	End Block	Block Size	Checksum
0x11	0x07	0x02		0x01				

Table 26 Read Single Block operation

Field Name	Length (Byte)	Description
UID	8	Tag's UID.
Option	1	It was reserved for future use and was fixed to 0x01 now.
Start Block	1	0~255.
End Block	1	It should have the value with "Start Block" field
Block Size	1	1~255. Depend on tag's memory configuration.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 27 Field description of Read Single Block operation

RR1X Series Programming Guide

The response of Read Single Block operation was specified as below.

Length	Command	Operation	Result	Block Data	Block Security Status	Checksum
	0x07	0x02				

Table 28 Response of Read Single Block operation

Field Name	Length (Byte)	Description
Length	1	0x06: Operation failed. “Result” field is not equal to 0x00. 0x06 + “Block Size” + 1: Operation success. “Result” field is equal to 0x00.
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Standard defined response error code was defined in Section 6.1.1 Others: Reserved.
Block Data	“Block Size”	This field is absent if “Result” field is not equal to 0x00. The content of specified block.
Block Security Status	1	This field is absent if “Result” field is not equal to 0x00. 0x00: Block was not locked. 0x01: Block was locked. Others: Reserved.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 29 Field description of Read Single Block operation response

RR1X Series Programming Guide

4.4.5.3 Write Single Block

Write Single Block operation was used to write one block data into specified block. The detail command format was specified as below.

Length	Command	Operation	UID	Option	Start Block	End Block	Block Size	Data	Checksum
	0x07	0x03							

Table 30 Write Single Block operation

Field Name	Length (Byte)	Description
Length	1	0x11 + “Block Size”
UID	8	Tag’s UID.
Option	1	0x00: Clear option flag of request flags. 0x01: Set option flag of request flags. Others: Reserved. For the detail of request flags, please refer to ISO15693 standard, section 7.3.1.
Start Block	1	0~255.
End Block	1	It should have the value with “Start Block” field
Block Size	1	1~255. Depend on tag’s memory configuration.
Data	“Block Size”	Data to be written. The data length is specified in “Block Size” field.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 31 Field description of Write Single Block operation

The response of Write Single Block operation was specified as below.

Length	Command	Operation	Result	Checksum
0x06	0x07	0x03		

Table 32 Response of Write Single Block operation

Field Name	Length (Byte)	Description
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Standard defined response error code was defined in Section 6.1.1 Others: Reserved.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

RR1X Series Programming Guide

Table 33 Field description of Write Single Block operation response

4.4.5.4 Lock Block

Lock Block operation was used to lock specified block. The detail command format was specified as below.

Length	Command	Operation	UID	Option	Block ID	Checksum
0x0F	0x07	0x05				

Table 34 Lock Block operation

Field Name	Length (Byte)	Description
UID	8	Tag's UID.
Option	1	0x00: Clear option flag of request flags. 0x01: Set option flag of request flags. Others: Reserved. For the detail of request flags, please refer to ISO15693 standard, section 7.3.1.
Block ID	1	0~255.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 35 Field description of Lock Block operation

RR1X Series Programming Guide

The response of Lock Block operation was specified as below.

Length	Command	Operation	Result	Checksum
0x06	0x07	0x05		

Table 36 Response of Lock Block operation

Field Name	Length (Byte)	Description
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Standard defined response error code was defined in Section 6.1.1 Others: Reserved.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 37 Field description of Lock Block operation response

RR1X Series Programming Guide

4.4.5.5 Read Multiple Blocks

Read Multiple Blocks operation was used to read multiple blocks data from specific blocks. The detail command format was specified as below.

Length	Command	Operation	UID	Option	Start Block	End Block	Block Size	Checksum
0x11	0x07	0x23						

Table 38 Read Multiple Blocks operation

Field Name	Length (Byte)	Description
UID	8	Tag's UID.
Option	1	0x00: Clear option flag of request flags. 0x01: Set option flag of request flags. Others: Reserved. For the detail of request flags, please refer to ISO15693 standard, section 7.3.1.
Start Block	1	0~255.
Number of Block	1	0~255. It should be greater than "Start Block". The maximum block amount in each Read Multiple Blocks operation is five blocks.
Block Size	1	1~255. Depend on tag's memory configuration.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 39 Field description of Read Multiple Blocks operation

RR1X Series Programming Guide

The response of Read Multiple Blocks operation was specified as below.

Length	Command	Operation	Result	Data	Checksum
	0x07	0x23			

Table 40 Response of Read Multiple Blocks operation

Field Name	Length (Byte)	Description
Length	1	0x06: Operation failed. “Result” field is not equal to 0x00. 0x06 + (“Block Size”) * (“End Block”-“Start Block”): Operation success while “Option” field is equal to 0x00 and “Result” field is equal to 0x00. 0x06 + (“Block Size”+1) * (“End Block”-“Start Block”): Operation success while “Option” field is equal to 0x01 and “Result” field is equal to 0x00.
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Standard defined response error code was defined in Section 6.1.1 Others: Reserved.
Data	(“Block Size”) * (“End Block”-“Start Block”) Or (“Block Size+1”) * (“End Block”-“Start Block”)	This field is absent if “Result” field is not equal to 0x00. While “Option” field is equal to 0x00. The basic data unit is “Block Data”. While “Option” field is equal to 0x01. The basic data unit is “Block Data” plus “Block Security Status”. “Block Data” and “Block Security Status” was described in section 4.4.5.2. Multiple basic data units are composed into “Data” field.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 41 Field description of Read Multiple Blocks operation response

RR1X Series Programming Guide

4.4.5.6 Select

Select operation will put specific tag into selected state. The detail command format was specified as below.

Length	Command	Operation	UID	Checksum
0x0D	0x07	0x06		

Table 42 Select operation

Field Name	Length (Byte)	Description
UID	8	Tag's UID.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 43 Field description of Select operation

The response of Select operation was specified as below.

Length	Command	Operation	Result	Checksum
0x06	0x07	0x06		

Table 44 Response of Select operation

Field Name	Length (Byte)	Description
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Standard defined response error code was defined in Section 6.1.1 Others: Reserved.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 45 Field description of Select operation response

RR1X Series Programming Guide

4.4.5.7 Reset to Ready

Reset to Ready operation was used to change specific tag from quiet or selected state to ready state. The detail command format was specified as below.

Length	Command	Operation	UID	Checksum
0x0D	0x07	0x06		

Table 46 Reset to Ready operation

Field Name	Length (Byte)	Description
UID	8	Tag's UID.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 47 Field description of Reset to Ready operation

The response of Reset to Ready operation was specified as below.

Length	Command	Operation	Result	Checksum
0x06	0x07	0x07		

Table 48 Response of Reset to Ready operation

Field Name	Length (Byte)	Description
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Standard defined response error code was defined in Section 6.1.1 Others: Reserved.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 49 Field description of Reset to Ready operation response

RR1X Series Programming Guide

4.4.5.8 Write AFI

Write AFI operation was used to set AFI for specific tag. The detail command format was specified as below.

Length	Command	Operation	UID	Option	AFI	Checksum
0x0F	0x07	0x08				

Table 50 Reset to Ready operation

Field Name	Length (Byte)	Description
UID	8	Tag's UID.
Option	1	0x00: Clear option flag of request flags. 0x01: Set option flag of request flags. Others: Reserved. For the detail of request flags, please refer to ISO15693 standard, section 7.3.1.
AFI	1	Application Family Identifier. The coding is listed in Figure 3.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 51 Field description of Write AFI operation

The response of Write AFI operation was specified as below.

Length	Command	Operation	Result	Checksum
0x06	0x07	0x08		

Table 52 Response of Write AFI operation

Field Name	Length (Byte)	Description
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Standard defined response error code was defined in Section 6.1.1 Others: Reserved.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 53 Field description of Write AFI operation response

RR1X Series Programming Guide

4.4.5.9 Lock AFI

Lock AFI operation was used to set lock memory that used to store AFI. The detail command format was specified as below.

Length	Command	Operation	UID	Option	Checksum
0x0E	0x07	0x09			

Table 54 Lock AFI operation

Field Name	Length (Byte)	Description
UID	8	Tag's UID.
Option	1	0x00: Clear option flag of request flags. 0x01: Set option flag of request flags. Others: Reserved. For the detail of request flags, please refer to ISO15693 standard, section 7.3.1.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 55 Field description of Lock AFI operation

The response of Lock AFI operation was specified as below.

Length	Command	Operation	Result	Checksum
0x06	0x07	0x09		

Table 56 Response of Lock AFI operation

Field Name	Length (Byte)	Description
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Standard defined response error code was defined in Section 6.1.1 Others: Reserved.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 57 Field description of Lock AFI operation response

RR1X Series Programming Guide

4.4.5.10 Write DSFID

Write DSFID operation was used to set data storage format identifier (DSFID) for specific tag. The detail command format was specified as below.

Length	Command	Operation	UID	Option	DSFID	Checksum
0x0F	0x07	0x0A				

Table 58 Write DSFID operation

Field Name	Length (Byte)	Description
UID	8	Tag's UID.
Option	1	0x00: Clear option flag of request flags. 0x01: Set option flag of request flags. Others: Reserved. For the detail of request flags, please refer to ISO15693 standard, section 7.3.1.
DSFID	1	Refer to tag's datasheet.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 59 Field description of Write DSFID operation

The response of Write DSFID operation was specified as below.

Length	Command	Operation	Result	Checksum
0x06	0x07	0x0A		

Table 60 Response of Write DSFID operation

Field Name	Length (Byte)	Description
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Standard defined response error code was defined in Section 6.1.1 Others: Reserved.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 61 Field description of Write DSFID operation response

RR1X Series Programming Guide

4.4.5.11 Lock DSFID

Lock DSFID operation was used to set lock memory that used to store DSFID. The detail command format was specified as below.

Length	Command	Operation	UID	Option	Checksum
0x0E	0x07	0x0B			

Table 62 Lock DSFID operation

Field Name	Length (Byte)	Description
UID	8	Tag's UID.
Option	1	0x00: Clear option flag of request flags. 0x01: Set option flag of request flags. Others: Reserved. For the detail of request flags, please refer to ISO15693 standard, section 7.3.1.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 63 Field description of Lock DSFID operation

The response of Lock DSFID operation was specified as below.

Length	Command	Operation	Result	Checksum
0x06	0x07	0x0B		

Table 64 Response of Lock DSFID operation

Field Name	Length (Byte)	Description
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Standard defined response error code was defined in Section 6.1.1 Others: Reserved.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 65 Field description of Lock DSFID operation response

RR1X Series Programming Guide

4.4.5.12 Get System Information

Get System Information operation was used to get tag's system information. The detail command format was specified as below.

Length	Command	Operation	UID	Checksum
0x0D	0x07	0x0C		

Table 66 Get System Information operation

Field Name	Length (Byte)	Description
UID	8	Tag's UID.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 67 Field description of Get System Information operation

The response of Get System Information operation was specified as below.

Length	Command	Operation	Result	Flag	DSFID	AFI	Block Amount	Block Size	IC Ref	Checksum
	0x07	0x0C								

Table 68 Response of Get System Information operation

RR1X Series Programming Guide

Field Name	Length (Byte)	Description
Length	1	0x06: Operation failed. “Result” field is not equal to 0x00. 0x0D: Operation success. “Result” field is equal to 0x00.
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Standard defined response error code was defined in Section 6.1.1 Others: Reserved.
Flag	1	This field is absent if “Result” field is not equal to 0x00. Bit 0: 1 “DSFID” field is valid. 0 “DSFID” field is invalid. Bit 1: 1 “AFI” field is valid. 0 “AFI” field is invalid. Bit 2: 1 “Block Amount” and “Block Size” fields are valid. 0 “Block Amount” and “Block Size” fields are invalid. Bit 3: 1 “IC Ref” field is valid. 0 “IC Ref” field is invalid. Bit 4 ~ Bit 7: Reserved.
DSFID	1	This field is absent if “Result” field is not equal to 0x00. Refer to tag’s datasheet.
AFI	1	This field is absent if “Result” field is not equal to 0x00. Application Family Identifier. The coding is listed in Figure 3.
Block Amount	1	This field is absent if “Result” field is not equal to 0x00. Tag has “Block Amount”+1 blocks.
Block Size	1	This field is absent if “Result” field is not equal to 0x00. The block size is “Block Size”+1 bytes.
IC Ref	1	This field is absent if “Result” field is not equal to 0x00. Refer to tag’s datasheet.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 69 Field description of Get System Information operation response

RR1X Series Programming Guide

4.4.5.13 Get Multiple Block Security Status

Get Multiple Block Security Status operation was used to get security status of specified blocks. The detail command format was specified as below.

Length	Command	Operation	UID	Flag	Start Block	Block Amount	Checksum
0x10	0x07	0x2C		0x22			

Table 70 Get Multiple Block Security Status operation

Field Name	Length (Byte)	Description
UID	8	Tag's UID.
Flag	1	It was reserved for future use and was fixed to 0x22 now.
Start Block	1	0~255
Block Amount	1	1~10
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 71 Field description of Get Multiple Block Security Status operation

RR1X Series Programming Guide

The response of Get Multiple Block Security Status operation was specified as below.

Length	Command	Operation	Result	Data	Checksum
	0x07	0x2C			

Table 72 Response of Get Multiple Block Security Status operation

Field Name	Length (Byte)	Description
Length	1	0x06: Operation failed. “Result” field is not equal to 0x00. 0x06 +”Block Amount”: Operation success. “Result” field is equal to 0x00.
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Standard defined response error code was defined in Section 6.1.1 Others: Reserved.
Data	”Block Amount”	This field is absent if “Result” field is not equal to 0x00. The content of each byte is as below: 0x00: Block was not locked. 0x01: Block was locked. Others: Reserved.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 73 Field description of Get Multiple Block Security Status operation response

RR1X Series Programming Guide

4.4.5.14 Read EAS

MSR3200 tag proprietary command.

The detail command format was specified as below.

Length	Command	Operation	UID	Checksum
0x0D	0x07	0xA0		

Table 74 Read EAS operation

Field Name	Length (Byte)	Description
UID	8	Tag's UID.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 75 Field description of Read EAS operation

The response of Read EAS operation was specified as below.

Length	Command	Operation	Result	EAS	Checksum
	0x07	0xA0			

Table 76 Response of Read EAS operation

Field Name	Length (Byte)	Description
Length	1	0x06: Operation failed. "Result" field is not equal to 0x00. 0x07: Operation success. "Result" field is equal to 0x00.
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Standard defined response error code was defined in Section 6.1.1 Others: Reserved.
EAS	1	This field is absent if "Result" field is not equal to 0x00. Please refer to MSR3200 datasheet.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 77 Field description of Read EAS operation response

RR1X Series Programming Guide

4.4.5.15 Write EAS

MSR3200 tag proprietary command.

The detail command format was specified as below.

Length	Command	Operation	UID	EAS	Checksum
0x0E	0x07	0xA1			

Table 78 Write EAS operation

Field Name	Length (Byte)	Description
UID	8	Tag's UID.
EAS	1	Please refer to MSR3200 datasheet.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 79 Field description of Write EAS operation

The response of Write EAS operation was specified as below.

Length	Command	Operation	Result	Checksum
0x06	0x07	0xA1		

Table 80 Response of Write EAS operation

Field Name	Length (Byte)	Description
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Standard defined response error code was defined in Section 6.1.1 Others: Reserved.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 81 Field description of Write EAS operation response

RR1X Series Programming Guide

4.4.5.16 Lock EAS

MSR3200 tag proprietary command.

The detail command format was specified as below.

Length	Command	Operation	UID	Checksum
0x0D	0x07	0xA2		

Table 82 Lock EAS operation

Field Name	Length (Byte)	Description
UID	8	Tag's UID.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 83 Field description of Lock EAS operation

The response of Lock EAS operation was specified as below.

Length	Command	Operation	Result	Checksum
0x06	0x07	0xA2		

Table 84 Response of Lock EAS operation

Field Name	Length (Byte)	Description
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Standard defined response error code was defined in Section 6.1.1 Others: Reserved.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 85 Field description of Lock EAS operation response

RR1X Series Programming Guide

4.4.5.17 Secure Read

MSR3200 tag proprietary command.

The detail command format was specified as below.

Length	Command	Operation	UID	Block ID	Block Size	PWD Length	PWD	Checksum
0x10+”PWD Length”	0x07	0xA3						

Table 86 Secure Read operation

Field Name	Length (Byte)	Description
UID	8	Tag’s UID.
Block ID	1	0~255 Please refer to MSR3200 datasheet.
Block Size	1	1~255 Please refer to MSR3200 datasheet.
PWD Length	1	0x04: Key is enabled 0x08: Key is disabled Others: Reserved Please refer to MSR3200 datasheet.
PWD	“PWD Length”	Password. Please refer to MSR3200 datasheet.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 87 Field description of Secure Read operation

RR1X Series Programming Guide

The response of Secure Read operation was specified as below.

Length	Command	Operation	Result	Block Data	Block Security Status	Checksum
	0x07	0xA3				

Table 88 Response of Secure Read operation

Field Name	Length (Byte)	Description
Length	1	0x04: Operation failed. “Result” field is not equal to 0x00. 0x04 +”Block Size” + 1: Operation success. “Result” field is equal to 0x00.
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Standard defined response error code was defined in Section 6.1.1 Others: Reserved.
Block Data	“Block Size”	This field is absent if “Result” field is not equal to 0x00. The content of specified block.
Block Security Status	1	This field is absent if “Result” field is not equal to 0x00. Please refer to MSR3200 datasheet.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 89 Field description of Secure Read operation response

RR1X Series Programming Guide

4.4.5.18 Secure Write

MSR3200 tag proprietary command.

The detail command format was specified as below.

Length	Command	Operation	UID	Block ID	PWD Length	Block Data	PWD	Checksum
0x13+”PWD Length”	0x07	0xA4						

Table 90 Secure Write operation

Field Name	Length (Byte)	Description
UID	8	Tag’s UID.
Block ID	1	0~255
PWD Length	1	0x04: Key is enabled 0x08: Key is disabled Others: Reserved Please refer to MSR3200 datasheet.
Block Data	4	Data to be written.
PWD	“PWD Length”	Password. Please refer to MSR3200 datasheet.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 91 Field description of Secure Write operation

RR1X Series Programming Guide

The response of Secure Write operation was specified as below.

Length	Command	Operation	Result	Checksum
0x06	0x07	0xA4		

Table 92 Response of Secure Write operation

Field Name	Length (Byte)	Description
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Standard defined response error code was defined in Section 6.1.1 Others: Reserved.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 93 Field description of Secure Write operation response

RR1X Series Programming Guide

4.4.5.19 Get Owner ID

MSR3200 tag proprietary command.

The detail command format was specified as below.

Length	Command	Operation	UID	Checksum
0x0D	0x07	0xA5		

Table 94 Get Owner ID operation

Field Name	Length (Byte)	Description
UID	8	Tag's UID.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 95 Field description of Get Owner ID operation

The response of Get Owner ID operation was specified as below.

Length	Command	Operation	Result	Owner ID	Checksum
	0x07	0xA5			

Table 96 Response of Get Owner ID operation

Field Name	Length (Byte)	Description
Length	1	0x06: Operation failed. "Result" field is not equal to 0x00. 0x07: Operation success. "Result" field is equal to 0x00.
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Standard defined response error code was defined in Section 6.1.1 Others: Reserved.
Owner ID	1	This field is absent if "Result" field is not equal to 0x00. Please refer to MSR3200 datasheet.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 97 Field description of Get Owner ID operation response

RR1X Series Programming Guide

4.4.5.20 Write PWD

MSR3200 tag proprietary command.

The detail command format was specified as below.

Length	Command	Operation	UID	PWD ID	Data	Checksum
0x0F	0x07	0xA6				

Table 98 Write PWD operation

Field Name	Length (Byte)	Description
UID	8	Tag's UID.
PWD ID	1	If Key is enabled, the length of password is 8 bytes. Set PWD ID to 0 for writing lower four bytes of password. And Set PWD ID to 1 for writing upper four bytes of password. Please refer to MSR3200 datasheet.
Data	4	Data to be written.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 99 Field description of Write PWD operation

The response of Write PWD operation was specified as below.

Length	Command	Operation	Result	Checksum
0x06	0x07	0xA6		

Table 100 Response of Write PWD operation

Field Name	Length (Byte)	Description
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Standard defined response error code was defined in Section 6.1.1 Others: Reserved.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 101 Field description of Write PWD operation response

RR1X Series Programming Guide

4.4.5.21 Lock PWD

MSR3200 tag proprietary command.

The detail command format was specified as below.

Length	Command	Operation	UID	Checksum
0x0D	0x07	0xA7		

Table 102 Lock PWD operation

Field Name	Length (Byte)	Description
UID	8	Tag's UID.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 103 Field description of Lock PWD operation

The response of Lock PWD operation was specified as below.

Length	Command	Operation	Result	Checksum
0x06	0x07	0xA7		

Table 104 Response of Lock PWD operation

Field Name	Length (Byte)	Description
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Standard defined response error code was defined in Section 6.1.1 Others: Reserved.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 105 Field description of Lock PWD operation response

RR1X Series Programming Guide

4.4.5.22 Read Key

MSR3200 tag proprietary command.

The detail command format was specified as below.

Length	Command	Operation	UID	Checksum
0x0D	0x07	0xA8		

Table 106 Read Key operation

Field Name	Length (Byte)	Description
UID	8	Tag's UID.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 107 Field description of Read Key operation

The response of Read Key operation was specified as below.

Length	Command	Operation	Result	Key	Checksum
	0x07	0xA8			

Table 108 Response of Read Key operation

Field Name	Length (Byte)	Description
Length	1	0x06: Operation failed. "Result" field is not equal to 0x00. 0x0A: Operation success. "Result" field is equal to 0x00.
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Standard defined response error code was defined in Section 6.1.1 Others: Reserved.
Key	4	This field is absent if "Result" field is not equal to 0x00. Please refer to MSR3200 datasheet.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 109 Field description of Read Key operation response

RR1X Series Programming Guide

4.4.5.23 Write Key

MSR3200 tag proprietary command.

The detail command format was specified as below.

Length	Command	Operation	UID	Key ID	Data	Checksum
0x12	0x07	0xA9				

Table 110 Write Key operation

Field Name	Length (Byte)	Description
UID	8	Tag's UID.
Key ID	1	Please refer to MSR3200 datasheet.
Data	4	Please refer to MSR3200 datasheet.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 111 Field description of Write Key operation

The response of Write Key operation was specified as below.

Length	Command	Operation	Result	Checksum
0x06	0x07	0xA9		

Table 112 Response of Write Key operation

Field Name	Length (Byte)	Description
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Standard defined response error code was defined in Section 6.1.1 Others: Reserved.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 113 Field description of Write Key operation response

RR1X Series Programming Guide

4.4.5.24 Lock Key

MSR3200 tag proprietary command.

The detail command format was specified as below.

Length	Command	Operation	UID	Checksum
0x0D	0x07	0xAA		

Table 114 Lock Key operation

Field Name	Length (Byte)	Description
UID	8	Tag's UID.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 115 Field description of Lock Key operation

The response of Lock Key operation was specified as below.

Length	Command	Operation	Result	Checksum
0x06	0x07	0xAA		

Table 116 Response of Lock Key operation

Field Name	Length (Byte)	Description
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Standard defined response error code was defined in Section 6.1.1 Others: Reserved.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 117 Field description of Lock Key operation response

RR1X Series Programming Guide

4.4.5.25 Read Register

MSR3200 tag proprietary command.

The detail command format was specified as below.

Length	Command	Operation	UID	Register ID	Checksum
0x0D	0x07	0xAB			

Table 118 Read Register operation

Field Name	Length (Byte)	Description
UID	8	Tag's UID.
Register ID	1	0x00 or 0x01 Others: Reserved. Please refer to MSR3200 datasheet.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 119 Field description of Read Register operation

The response of Read Register operation was specified as below.

Length	Command	Operation	Result	Data	Checksum
	0x07	0xAB			

Table 120 Response of Read Register operation

Field Name	Length (Byte)	Description
Length	1	0x06: Operation failed. "Result" field is not equal to 0x00. 0x0A: Operation success. "Result" field is equal to 0x00.
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Standard defined response error code was defined in Section 6.1.1 Others: Reserved.
Data	4	This field is absent if "Result" field is not equal to 0x00. Please refer to MSR3200 datasheet.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 121 Field description of Read Register operation response

RR1X Series Programming Guide

4.4.5.26 Write Register

MSR3200 tag proprietary command.

The detail command format was specified as below.

Length	Command	Operation	UID	Register ID	Data	Checksum
0x12	0x07	0xAC				

Table 122 Write Register operation

Field Name	Length (Byte)	Description
UID	8	Tag's UID.
Register ID	1	0x00 or 0x01 Others: Reserved. Please refer to MSR3200 datasheet.
Data	4	Data to be written.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 123 Field description of Write Register operation

The response of Write Register operation was specified as below.

Length	Command	Operation	Result	Checksum
0x06	0x07	0xAC		

Table 124 Response of Write Register operation

Field Name	Length (Byte)	Description
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Standard defined response error code was defined in Section 6.1.1 Others: Reserved.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 125 Field description of Write Register operation response

RR1X Series Programming Guide

4.4.5.27 Write PWD with PWD

MSR3200 tag proprietary command.

The detail command format was specified as below.

Length	Command	Operation	UID	PWD ID	New PWD	PWD Length	Old PWD	Checksum
0x0F	0x07	0xAD						

Table 126 Write PWD with PWD operation

Field Name	Length (Byte)	Description
UID	8	Tag's UID.
PWD ID	1	If Key is enabled, the length of password is 8 bytes. Set PWD ID to 0 for writing lower four bytes of password. And Set PWD ID to 1 for writing upper four bytes of password. Please refer to MSR3200 datasheet.
New PWD	4	Data to be written.
PWD Length	1	0x04: Key is enabled 0x08: Key is disabled Others: Reserved Please refer to MSR3200 datasheet.
Old PWD	"PWD Length"	Old password. Please refer to MSR3200 datasheet.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 127 Field description of Write PWD with PWD operation

RR1X Series Programming Guide

The response of Write PWD with PWD operation was specified as below.

Length	Command	Operation	Result	Checksum
0x06	0x07	0xAD		

Table 128 Response of Write PWD with PWD operation

Field Name	Length (Byte)	Description
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Standard defined response error code was defined in Section 6.1.1 Others: Reserved.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 129 Field description of Write PWD with PWD operation response

RR1X Series Programming Guide

4.4.5.28 Fast Read Single Block

It's Fujitsu custom command. The Fast Read Single Block command is the same as the Read Single Block command that reads data stored in the specific single-block. The detail command format was specified as below.

Length	Command	Operation	UID	Block ID	Block Size	Checksum
0x0F	0x07	0xF1				

Table 130 Fast Read Single Block operation

Field Name	Length (Byte)	Description
UID	8	Tag's UID.
Block ID	1	0~255.
Block Size	1	1~255. Depend on tag's memory configuration.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 131 Field description of Fast Read Single Block operation

RR1X Series Programming Guide

The response of Fast Read Single Block operation was specified as below.

Length	Command	Operation	Result	Block Data	Block Security Status	Checksum
	0x07	0xF1				

Table 132 Response of Fast Read Single Block operation

Field Name	Length (Byte)	Description
Length	1	0x06: Operation failed. “Result” field is not equal to 0x00. 0x06 + “Block Size” + 1: Operation success. “Result” field is equal to 0x00.
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Standard defined response error code was defined in Section 6.1.1 Others: Reserved.
Block Data	“Block Size”	This field is absent if “Result” field is not equal to 0x00. The content of specified block.
Block Security Status	1	This field is absent if “Result” field is not equal to 0x00. 0x00: Block was not locked. 0x01: Block was locked. Others: Reserved.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 133 Field description of Fast Read Single Block operation response

RR1X Series Programming Guide

4.4.5.29 Fast Write Single Block

It's Fujitsu custom command. The Fast Write Single Block command is the same as the Write Single Block command that writes the single block data included in the request. The detail command format was specified as below.

Length	Command	Operation	UID	Option	Start Block	End Block	Block Size	Data	Checksum
	0x07	0xF2							

Table 134 Fast Write Single Block operation

Field Name	Length (Byte)	Description
Length	1	0x11 + "Block Size"
UID	8	Tag's UID.
Option	1	0x00: Clear option flag of request flags. 0x01: Set option flag of request flags. Others: Reserved. For the detail of request flags, please refer to ISO15693 standard, section 7.3.1.
Start Block	1	0~255.
End Block	1	It should have the value with "Start Block" field
Block Size	1	1~255. Depend on tag's memory configuration.
Data	"Block Size"	Data to be written. The data length is specified in "Block Size" field.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 135 Field description of Fast Write Single Block operation

RR1X Series Programming Guide

The response of Write Single Block operation was specified as below.

Length	Command	Operation	Result	Checksum
0x06	0x07	0xF2		

Table 136 Response of Fast Write Single Block operation

Field Name	Length (Byte)	Description
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Standard defined response error code was defined in Section 6.1.1 Others: Reserved.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 137 Field description of Fast Write Single Block operation response

RR1X Series Programming Guide

4.4.6 Mifare Ultralight Tag Access

Mifare Ultralight Tag Access command was used to access Mifare Ultralight tags appeared in RF field.

Mifare Ultralight Tag Access command was specified as below. This is a generic format. For each operation, “Parameters” field has different format. The detail was described in below sections.

Length	Command	Operation	Parameters	Checksum
	0x08			

Table 138 Mifare Ultralight Tag Access command

4.4.6.1 Read

Read operation will read data from specified address. The detail command format was specified as below.

Length	Command	Operation	Address	Checksum
0x06	0x08	0x00		

Table 139 Read operation

Field Name	Length (Byte)	Description
Address	1	Please refer to Mifare Ultralight datasheet.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 140 Field description of Read operation

RR1X Series Programming Guide

The response of Read operation was specified as below.

Length	Command	Operation	Result	Data	Checksum
	0x08	0x00			

Table 141 Response of Read operation

Field Name	Length (Byte)	Description
Length	1	0x07: Operation failed. “Result” field is not equal to 0x00, or “Address” field contains invalid value. 0x16: Operation success. Others: Reserved.
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Others: Reserved.
Data	1 or 18	1. Tag responses NAK: 0xX0 2. Read success, tag response: 16 bytes Block data(D0 – D15) and 2 bytes CRC.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 142 Field description of Read operation response

RR1X Series Programming Guide

4.4.6.2 Write

Write operation will write data into specified address. The detail command format was specified as below.

Length	Command	Operation	Address	Data	Checksum
0x0A	0x08	0x01			

Table 143 Write operation

Field Name	Length (Byte)	Description
Address	1	Please refer to Mifare Ultralight datasheet.
Data	4	Data to be written.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 144 Field description of Write operation

The response of Write operation was specified as below.

Length	Command	Operation	Result	DATA	Checksum
0x07	0x08	0x01			

Table 145 Response of Write operation

Field Name	Length (Byte)	Description
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Others: Reserved.
DATA	1	AK: Need to check Mifare Ultralight datasheet. NAK: 0xX0, X Don't care.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 146 Field description of Write operation

RR1X Series Programming Guide

4.4.7 ISO14443A Tag Inventory

ISO14443A Tag Inventory command was used to find tags appeared in RF field.

ISO14443A Tag Inventory command was specified as below.

Length	Command	Reserve	Checksum
0x05	0x09	0x00	

Table 147 ISO14443A Tag Inventory command

Field Name	Length (Byte)	Description
Reserve	1	It was reserved for future use and was fixed to 0x00 now.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 148 Field description of ISO14443A Tag Inventory command

The response of ISO14443A Tag Inventory command was specified as below.

Length	Command	Reserve 1	Tag Amount	UID Length	UIDs	Checksum
N	0x09	0x00				

Table 149 Response of ISO14443A Tag Inventory command

Field Name	Length (Byte)	Description
Length	1	0x0E + Length of “UIDs” field.
Reserve 1	1	It was reserved for future use and was fixed to 0x00 now.
Tag Amount	1	0x00 ~ 0x08 Others: Reserved.
UID Length	8	Specify length of each UID by order.
UIDs	n	UIDs. UID length could be 3, 7, or 10 bytes and was specified in “UID Length” field.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 150 Field description of ISO14443A Tag Inventory response

4.4.8 TOPAZ Tag Inventory and Access

Proprietary commands of TOPAZ tag

TOPAZ Tag Inventory and Access command was used to inventory and access TOPAZ tags appeared in RF field.

TOPAZ Tag Inventory and Access command was specified as below. This is a generic format. For each operation, “Parameters” field has different format. The detail was described in below sections.

Length	Command	Operation	Parameters	Checksum
	0x0F			

Table 151 TOPAZ Tag Inventory and Access command

4.4.8.1 Wake Up

The detail command format was specified as below.

Length	Command	Operation	Checksum
0x05	0x0F	0x00	

Table 152 Wake Up operation

Field Name	Length (Byte)	Description
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 153 Field description of TOPAZ Tag Wake Up command

The response of Wake Up operation was specified as below.

Length	Command	Operation	Result	ATQA	Checksum
0x08	0x0F	0x00			

Table 154 Response of Wake Up operation

Field Name	Length (Byte)	Description
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Others: Reserved.
ATQA	2	Please refer ISO14443-3 Table 3 – Coding of ATQA.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 155 Field description of Wake Up operation

4.4.8.2 Get ID

The detail command format was specified as below.

Length	Command	Operation	Checksum
0x05	0x0F	0x01	

Table 156 Get ID operation

Field Name	Length (Byte)	Description
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 157 Field description of TOPAZ Tag Get ID command

The response of Get ID operation was specified as below.

Length	Command	Operation	Result	Data	Checksum
0x0C	0x0F	0x01			

Table 158 Response of Get ID operation

Field Name	Length (Byte)	Description
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Others: Reserved.
Data	6	Format: HR0 HR1 UID0 UID1 UID2 UID3 HR0: 0x11 HR1: Reserved. Don't care.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 159 Field description of Get ID operation

RR1X Series Programming Guide

4.4.8.3 Read

The detail command format was specified as below.

Length	Command	Operation	Address	UID	Checksum
0x0A	0x0F	0x02			

Table 160 Read operation

Field Name	Length (Byte)	Description
Address	1	Please refer to TOPAZ datasheet.
UID	4	Please refer to TOPAZ datasheet.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 161 Field description of Read operation

The response of Read operation was specified as below.

Length	Command	Operation	Result	Data	Checksum
	0x0F	0x02			

Table 162 Response of Read operation

Field Name	Length (Byte)	Description
Length	1	0x06: Operation failed. “Result” field is not equal to 0x00. 0x0A: Operation success. “Result” field is equal to 0x00. Others: Reserved.
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Others: Reserved.
Data	4	DATA[0]: ADDR. The same as “Address” field in command. DATA[1]: DATA. DATA[2]: CRCL. DATA[3]: CRCH. Please refer to TOPAZ datasheet.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 163 Field description of Read operation

RR1X Series Programming Guide

4.4.8.4 Write

The detail command format was specified as below.

Length	Command	Operation	Address	Data	UID	Checksum
0x0A	0x0F	0x03				

Table 164 Write operation

Field Name	Length (Byte)	Description
Address	1	Please refer to TOPAZ datasheet.
Data	1	Data to be written.
UID	4	Please refer to TOPAZ datasheet.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 165 Field description of Write operation

The response of Write operation was specified as below.

Length	Command	Operation	Result	Data	Checksum
0x0A	0x0F	0x03			

Table 166 Response of Write operation

Field Name	Length (Byte)	Description
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Others: Reserved.
Data	4	DATA[0]: ADDR. The same as “Address” field in command. DATA[1]: DATA. The same as “Data” field in command. DATA[2]: CRCL. DATA[3]: CRCH. Please refer to TOPAZ datasheet.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 167 Field description of Write operation

RR1X Series Programming Guide

4.4.9 ST Tag Inventory

ST Tag Inventory command was used to find tags appeared in RF field.

ST Tag Inventory command was specified as below.

Length	Command	Reserve	Checksum
0x05	0x0A	0x00	

Table 168 ST Tag Inventory command

Field Name	Length (Byte)	Description
Reserve	1	It was reserved for future use and was fixed to 0x00 now.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 169 Field description of ST Tag Inventory command

The response of ST Tag Inventory command was specified as below.

Length	Command	Reserve 1	Tag Amount	Chip IDs	UIDs	Checksum
N	0x0A	0x00				

Table 170 Response of ST Tag Inventory command

Field Name	Length (Byte)	Description
Length	1	0x0E + Length of “UIDs” field.
Reserve 1	1	It was reserved for future use and was fixed to 0x00 now.
Tag Amount	1	0x00 ~ 0x08 Others: Reserved.
Chip IDs	8	Chip IDs. How many Chip IDs are valid depends on “Tag Amount” field.
UIDs	8 * Tag Amount	Every tag has 8 bytes UID. How many UIDs are valid depends on “Tag Amount” field.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 171 Field description of ST Tag Inventory response

RR1X Series Programming Guide

4.4.10 ST Tag Access

ST Tag Access command was used to access tag's information.

ST Tag Access command was specified as below. This is a generic format. For each operation, "Parameters" field has different format. The detail was described in below sections.

Length	Command	Operation	Parameters	Checksum
	0x0B			

Table 172 ST Tag Access command

4.4.10.1 Read

The detail command format was specified as below.

Length	Command	Operation	Chip ID	Address	Checksum
0x07	0x0B	0x00			

Table 173 Read operation

Field Name	Length (Byte)	Description
Chip ID	1	Please refer to ST RFID Tag datasheet.
Address	1	Please refer to ST RFID Tag datasheet.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 174 Field description of Read operation

RR1X Series Programming Guide

The response of Read operation was specified as below.

Length	Command	Operation	Result	Data	Checksum
	0x0B	0x02			

Table 175 Response of Read operation

Field Name	Length (Byte)	Description
Length	1	0x06: Operation failed. “Result” field is not equal to 0x00. 0x0A: Operation success. “Result” field is equal to 0x00. Others: Reserved.
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Others: Reserved.
Data	4	The content of specified address. Please refer to ST RFID Tag datasheet.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 176 Field description of Read operation response

RR1X Series Programming Guide

4.4.10.2 Write

The detail command format was specified as below.

Length	Command	Operation	Chip ID	Address	Data	Checksum
0x0A	0x0B	0x01				

Table 177 Write operation

Field Name	Length (Byte)	Description
Chip ID	1	Please refer to ST RFID Tag datasheet.
Address	1	Please refer to ST RFID Tag datasheet.
Data	4	Data to be written.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 178 Field description of Write operation

The response of Write operation was specified as below.

Length	Command	Operation	Result	Checksum
0x06	0x0B	0x01		

Table 179 Response of Write operation

Field Name	Length (Byte)	Description
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Others: Reserved.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 180 Field description of Write operation

RR1X Series Programming Guide

4.4.10.3 Completion

The detail command format was specified as below.

Length	Command	Operation	Chip ID	Checksum
0x06	0x0B	0x02		

Table 181 Completion operation

Field Name	Length (Byte)	Description
Chip ID	1	Please refer to ST RFID Tag datasheet.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 182 Field description of Completion operation

The response of Completion operation was specified as below.

Length	Command	Operation	Result	Checksum
0x06	0x0B	0x02		

Table 183 Response of Completion operation

Field Name	Length (Byte)	Description
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Others: Reserved.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 184 Field description of Completion operation

RR1X Series Programming Guide

4.4.11 ISO14443B Tag Inventory

ISO14443B Tag Inventory command was used to find tags appeared in RF field.

ISO14443B Tag Inventory command was specified as below.

Length	Command	Reserve	Checksum
0x05	0x0C	0x00	

Table 185 ISO14443B Tag Inventory command

Field Name	Length (Byte)	Description
Reserve	1	It was reserved for future use and was fixed to 0x00 now.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 186 Field description of ISO14443B Tag Inventory command

The response of ISO14443B Tag Inventory command was specified as below.

Length	Command	Reserve 1	Tag Amount	Reserve	PUPIs	Checksum
N	0x0C	0x00				

Table 187 Response of ISO14443B Tag Inventory command

Field Name	Length (Byte)	Description
Length	1	0x0E + Length of “PUPIs” field.
Reserve 1	1	It was reserved for future use and was fixed to 0x00 now.
Tag Amount	1	0x00 ~ 0x08 Others: Reserved.
Reserve	8	Reserved for future use. Don’t care.
PUPIs	4 * “Tag Amount”	Each PUPI has four bytes.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 188 Field description of ISO14443B Tag Inventory response

RR1X Series Programming Guide

4.4.12 ISO14443B Tag Access

ISO14443B Tag Access command was used to access tag's information.

ISO14443B Tag Access command was specified as below. This is a generic format. For each operation, "Parameters" field has different format. The detail was described in below sections.

Length	Command	Operation	Parameters	Checksum
	0x0D			

Table 189 ISO14443B Tag Access command

4.4.12.1 Attribute

The detail command format was specified as below.

Length	Command	Operation	CID	PUPI	Checksum
0x0A	0x0D	0x00			

Table 190 Attribute operation

Field Name	Length (Byte)	Description
CID	1	Card ID
PUPI	4	Tag's ID
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 191 Field description of Attribute operation

RR1X Series Programming Guide

The response of Attribute operation was specified as below.

Length	Command	Operation	Result	Data	CRC	Checksum
	0x0D	0x00				

Table 192 Response of Attribute operation

Field Name	Length (Byte)	Description
Length	1	0x06: Operation failed. “Result” field is not equal to 0x00. 0x09: Operation success. “Result” field is equal to 0x00. Others: Reserved.
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Others: Reserved.
Data	3	Data[0]: Bit0~Bit3 CID Bit4~Bit7 MBLI Data[1]: Lower byte of CRC16 Data[2]: Higher byte of CRC16
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 193 Field description of Attribute operation response

RR1X Series Programming Guide

4.4.12.2 Read

Proprietary command of ATMEL AT88RF020 tag

The detail command format was specified as below.

Length	Command	Operation	CID	Address	Checksum
0x07	0x0D	0x01			

Table 194 Read operation

Field Name	Length (Byte)	Description
Chip ID	1	The tag's CID by ATTRIB command assign.
Address	1	The memory page address that user want to read.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 195 Field description of Read operation

RR1X Series Programming Guide

The response of Read operation was specified as below.

Length	Command	Operation	Result	Data	Checksum
	0x0D	0x01			

Table 196 Response of Read operation

Field Name	Length (Byte)	Description
Length	1	0x06: Operation failed. “Result” field is not equal to 0x00. 0x0A: Operation success. “Result” field is equal to 0x00. Others: Reserved.
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Others: Reserved.
Data	4 or 12	1. When tag response NAK: CMD 0xX1 CRCL CRCH 2. When read success tag response : CMD Addr DATA[8] CRCL CRCH * CMD = CID 0x4
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 197 Field description of Read operation response

RR1X Series Programming Guide

4.4.12.3 Write

Proprietary command of ATMEL AT88RF020 tag

The detail command format was specified as below.

Length	Command	Operation	CID	Address	Data	Checksum
0x0F	0x0D	0x02				

Table 198 Write operation

Field Name	Length (Byte)	Description
Chip ID	1	The tag's CID by ATTRIB command assign.
Address	1	The memory page address that user want to write.
Data	8	Data to be written.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 199 Field description of Write operation

The response of Write operation was specified as below.

Length	Command	Operation	Result	Data	Checksum
0x0A	0x0D	0x02			

Table 200 Response of Write operation

Field Name	Length (Byte)	Description
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Others: Reserved.
Data	4	1. When tag response NAK: CMD 0xX1 CRCL CRCH 2. When tag response AK: CMD 0xX0 CRCL CRCH * CMD = CID 0x3
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 201 Field description of Write operation

RR1X Series Programming Guide

4.4.12.4 Lock

Proprietary command of ATME LAT88RF020 tag

The detail command format was specified as below.

Length	Command	Operation	CID	Data	Checksum
0x0A	0x0D	0x03			

Table 202 Lock operation

Field Name	Length (Byte)	Description
CID	1	The tag's CID by ATTRIB command assign.
Data	4	Lock bits that user want to assign.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 203 Field description of Lock operation

The response of Lock operation was specified as below.

Length	Command	Operation	Result	Data	Checksum
0x0A	0x0D	0x03			

Table 204 Response of Lock operation

Field Name	Length (Byte)	Description
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Others: Reserved.
Data	4	1. When tag response NAK: CMD 0xX1 CRCL CRCH 2. When tag response AK: CMD 0xX0 CRCL CRCH * CMD = CID 0x2
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 205 Field description of Lock operation

RR1X Series Programming Guide

4.4.12.5 Check Password

Proprietary command of ATMEL AT88RF020 tag

The detail command format was specified as below.

Length	Command	Operation	CID	PWD	Checksum
0x0E	0x0D	0x04			

Table 206 Check Password operation

Field Name	Length (Byte)	Description
CID	1	The tag's CID by ATTRIB command assign.
PWD	8	Please refer to ATMEL AT88020 tag datasheet
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 207 Field description of Check Password operation

The response of Check Password operation was specified as below.

Length	Command	Operation	Result	Data	Checksum
0x0A	0x0D	0x04			

Table 208 Response of Check Password operation

Field Name	Length (Byte)	Description
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Others: Reserved.
Data	4	1. When tag response NAK: CMD 0xX1 CRCL CRCH 2. When tag response AK: CMD 0xX0 CRCL CRCH * CMD = CID 0x6
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 209 Field description of Check Password operation

RR1X Series Programming Guide

4.4.12.6 Deselect

Proprietary command of ATMEL AT88RF020 tag

The detail command format was specified as below.

Length	Command	Operation	CID	Checksum
0x06	0x0D	0x05		

Table 210 Deselect operation

Field Name	Length (Byte)	Description
CID	1	The tag's CID by ATTRIB command assign.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 211 Field description of Deselect operation

The response of Deselect operation was specified as below.

Length	Command	Operation	Result	Data	Checksum
0x0A	0x0D	0x05			

Table 212 Response of Deselect operation

Field Name	Length (Byte)	Description
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Others: Reserved.
Data	4	1. When tag response NAK: CMD 0xX1 CRCL CRCH 2. When tag response AK: CMD 0xX0 CRCL CRCH * CMD = CID 0xA
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 213 Field description of Deselect operation

RR1X Series Programming Guide

4.4.12.7 Count

Proprietary command of ATME LAT88RF020 tag

The detail command format was specified as below.

Length	Command	Operation	CID	Data	Checksum
0x0C	0x0D	0x06			

Table 214 Count Operation

Field Name	Length (Byte)	Description
CID	1	The tag's CID by ATTRIB command assign.
Data	6	Signature field data that user want to write.
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 215 Field description of Count Operation

The response of Count Operation was specified as below.

Length	Command	Operation	Result	Data	Checksum
0x0A	0x0D	0x06			

Table 216 Response of Count operation

Field Name	Length (Byte)	Description
Result	1	0x00: Success. 0x60: SOF error. 0x63: CRC error. Others: Reserved.
Data	4	1. When tag response NAK: CMD 0xX1 CRCL CRCH 2. When tag response AK: CMD 0xX0 CRCL CRCH * CMD = CID 0xE
Checksum	2	Checksum is calculated from first byte to the previous byte of "Checksum" field. The LSB of checksum was sent first.

Table 217 Field description of Count Operation

RR1X Series Programming Guide

4.4.13 Felica Tag Inventory

Felica Tag Inventory command was used to find Felica tags appeared in RF field.

Felica Tag Inventory command was specified as below.

Length	Command	Checksum
0x04	0x0E	

Table 218 Felica Tag Inventory command

Field Name	Length (Byte)	Description
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 219 Field description of Felica tag Inventory command

The response of Felica Tag Inventory command was specified as below.

Length	Command	Tag No.	UIDs	Checksum
	0x0E			

Table 220 Response of Felica Tag Inventory command

Field Name	Length (Byte)	Description
Length	1	0x05 + (8* Tag No)
Tag No.	1	0x00 ~ 0x04 Others: Reserved.
UIDs	8* Tag No	Tags' UID.
Checksum	2	Checksum is calculated from first byte to the previous byte of “Checksum” field. The LSB of checksum was sent first.

Table 221 Field description of Felica Tag Inventory response

4.4.14 Mstar P2P Tx

TBD

4.4.15 Mstar P2P Rx

TBD

RR1X Series Programming Guide

4.4.16 ISO18092 Target

Data Exchange operation will try to send data to target device. The detail command format was specified as below.

Length	Command	DID	NAD	Data(*)
	0x20			

Table 222 Data Exchange operation

Field Name	Length (Byte)	Description
Length	1	0x04 + length of “Data” field.
DID	1	Device ID of Initiator.
NAD	1	Bit 0 ~ Bit 3: NAD of Target. Bit 4 ~ Bit 7: NAD of Initiator.
Data	N	Data received from Initiator or data will be sent to Initiator.

Table 223 Field description of Data Exchange operation

The response of Data Exchange operation was specified as below.

Length	Command	Result
0x03	0x20	

Table 224 Response of Data Exchange operation

Field Name	Length (Byte)	Description
Result	1	0x00: Success. 0x01: Time out. 0x04: CRC error. Others: Reserved.

Table 225 Field description of Data Exchange operation

RR1X Series Programming Guide

4.4.17 ISO18092 Initiator

ISO18092 Initiator command was used to activate as an Initiator role, defined in ISO18092 standard.

ISO18092 Initiator command was specified as below. This is a generic format. For each operation, “Parameters” field has different format. The detail was described in below sections.

Length	Command	Operation	Parameters
	0x21		

Table 226 ISO18092 Initiator command

4.4.17.1 Find Target

Find Target operation will try to find NFC target in the field. The detail command format was specified as below.

Length	Command	Operation	Type	DIDi	NADi	DoPSL	DSR	LR	Data(*)
	0x21	0x02							

Table 227 Find Target operation

Field Name	Length (Byte)	Description
Length	1	0x09 + length of “Data” field.
Type	1	0x04: Passive, 106 kbps 0x06: Passive, 212 kbps 0x08: Passive, 424 kbps Others: Reserved.
DIDi	1	Bit 0 ~ Bit3: Device ID of Initiator Bit 4 ~ Bit7: If one of these bits is set, the Device ID, represented by Bit 0 ~ Bit3, becomes valid.
NADi	1	Bit 0 ~ Bit3: NAD of Initiator Bit 4 ~ Bit7: If one of these bits is set, the NAD, represented by Bit 0 ~ Bit3, becomes valid.
DoPSL	1	If the value of “DoPSL” field is greater than zero, RR1X will do PSL after ATR. For the detail, please refer to ISO18092 standard.
DSR	1	Bit 0 ~ Bit3: DS Bit 4 ~ Bit7: DR
LR	1	Bit 0 ~ Bit3: LR Others: Reserved.
Data	N	Data will be used while finding target device.

Table 228 Field description of Find Target operation

The response of Find Target operation was specified as below.

RR1X Series Programming Guide

Length	Command	Result
0x03	0x21	

Table 229 Response of Find Target operation

Field Name	Length (Byte)	Description
Result	1	0x00: Success. 0x01: Time out. 0x04: CRC error. Others: Reserved.

Table 230 Field description of Find Target operation response

RR1X Series Programming Guide

4.4.17.2 Data Exchange

Data Exchange operation will try to send data to target device. The detail command format was specified as below.

Length	Command	Operation	DID	NAD	Data(*)
	0x21	0x00			

Table 231 Data Exchange operation

Field Name	Length (Byte)	Description
Length	1	0x05 + length of “Data” field.
DID	1	Device ID of Target.
NAD	1	Bit 0 ~ Bit 3: NAD of Target. Bit 4 ~ Bit 7: NAD of Initiator.
Data	N	Data received from Target or data will be sent to Target.

Table 232 Field description of Data Exchange operation

The response of Data Exchange operation was specified as below.

Length	Command	Result
0x03	0x21	

Table 233 Response of Data Exchange operation

Field Name	Length (Byte)	Description
Result	1	0x00: Success. 0x01: Time out. 0x04: CRC error. Others: Reserved.

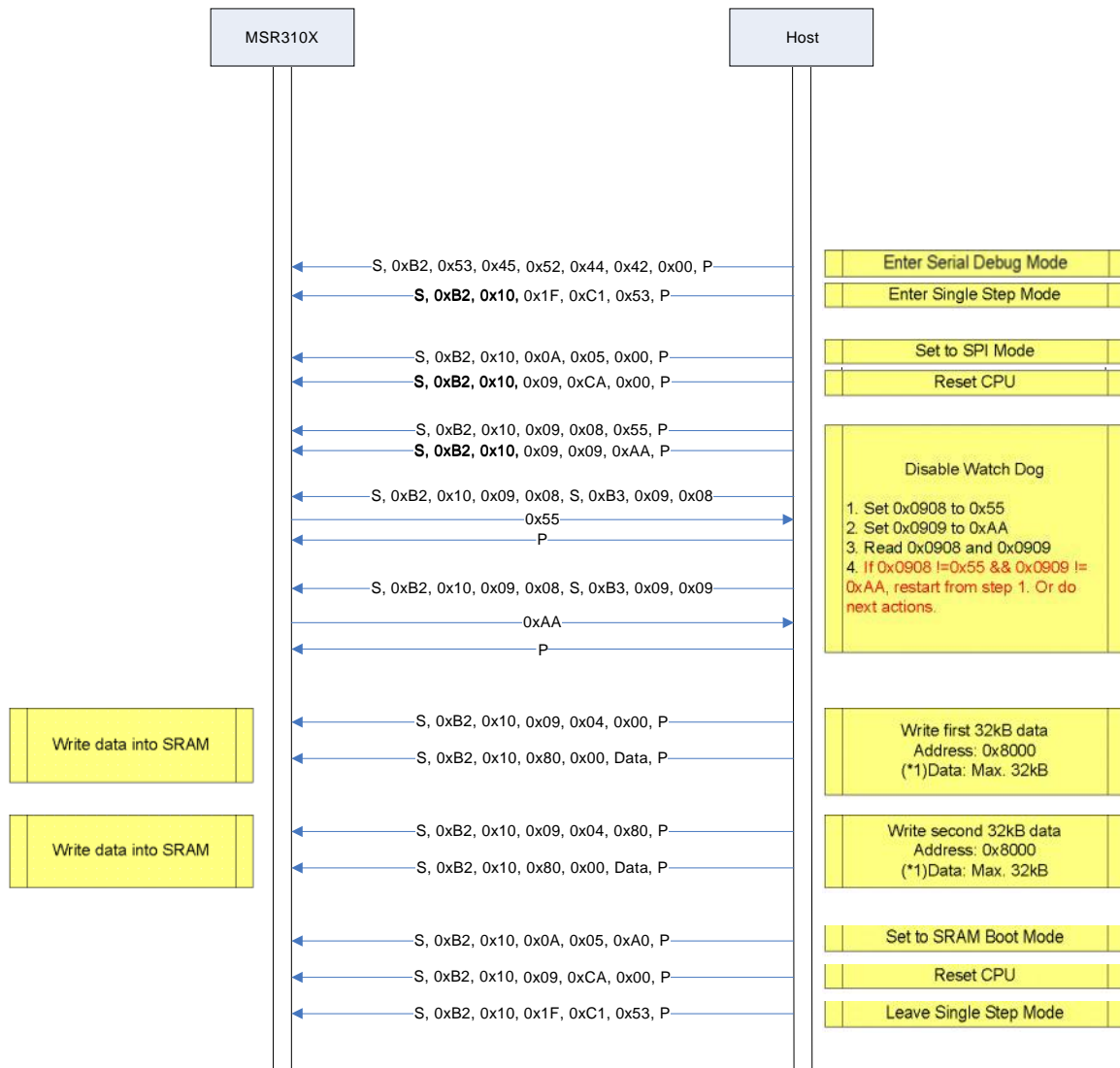
Table 234 Field description of Data Exchange operation response

RR1X Series Programming Guide

5. In-System-Programming

When RR1X activates as a slave device, there may not be a NOR flash for storing RR1X firmware. In this case, the firmware should be downloaded from host at run time. Below section describes the method for downloading firmware into the SRAM of RR1X directly and run it. We called In-System-Programming hereafter.

RR1X Series Programming Guide



(*1) 32 kB is the maximum number. User could send smaller data size and control address offset to write data.
Ex: Write 256 bytes each time. The address increase 256 bytes every time. Detail sequence as below.

Step 1:	0xB2, 0x10, 0x80, 0x00, [256 bytes data]	→ One I2C Transfer Frame
Step 2:	0xB2, 0x10, 0x81, 0x00, [256 bytes data]	→ One I2C Transfer Frame
Step 2:	0xB2, 0x10, 0x82, 0x00, [256 bytes data]	→ One I2C Transfer Frame
:	:	
:	:	
:	:	

(*2) S: I2C Start bit
P: I2C Stop bit

Figure 10 In-System-Programming Sequence

6. Appendix

6.1 Error Code definition

6.1.1 ISO15693 Standard Defined Response Error Code

Error code	Meaning
'01'	The command is not supported, i.e. the request code is not recognised.
'02'	The command is not recognised, for example: a format error occurred.
'03'	The option is not supported.
'0F'	Unknown error.
'10'	The specified block is not available (doesn't exist).
'11'	The specified block is already -locked and thus cannot be locked again
'12'	The specified block is locked and its content cannot be changed.
'13'	The specified block was not successfully programmed.
'14'	The specified block was not successfully locked.
'A0' – 'DF'	Custom command error codes
all others	RFU

Figure 11 ISO15693 Standard Define Response Error Code

6.2 Tags' State Diagram

6.2.1 ISO15693 Tag's State Diagram

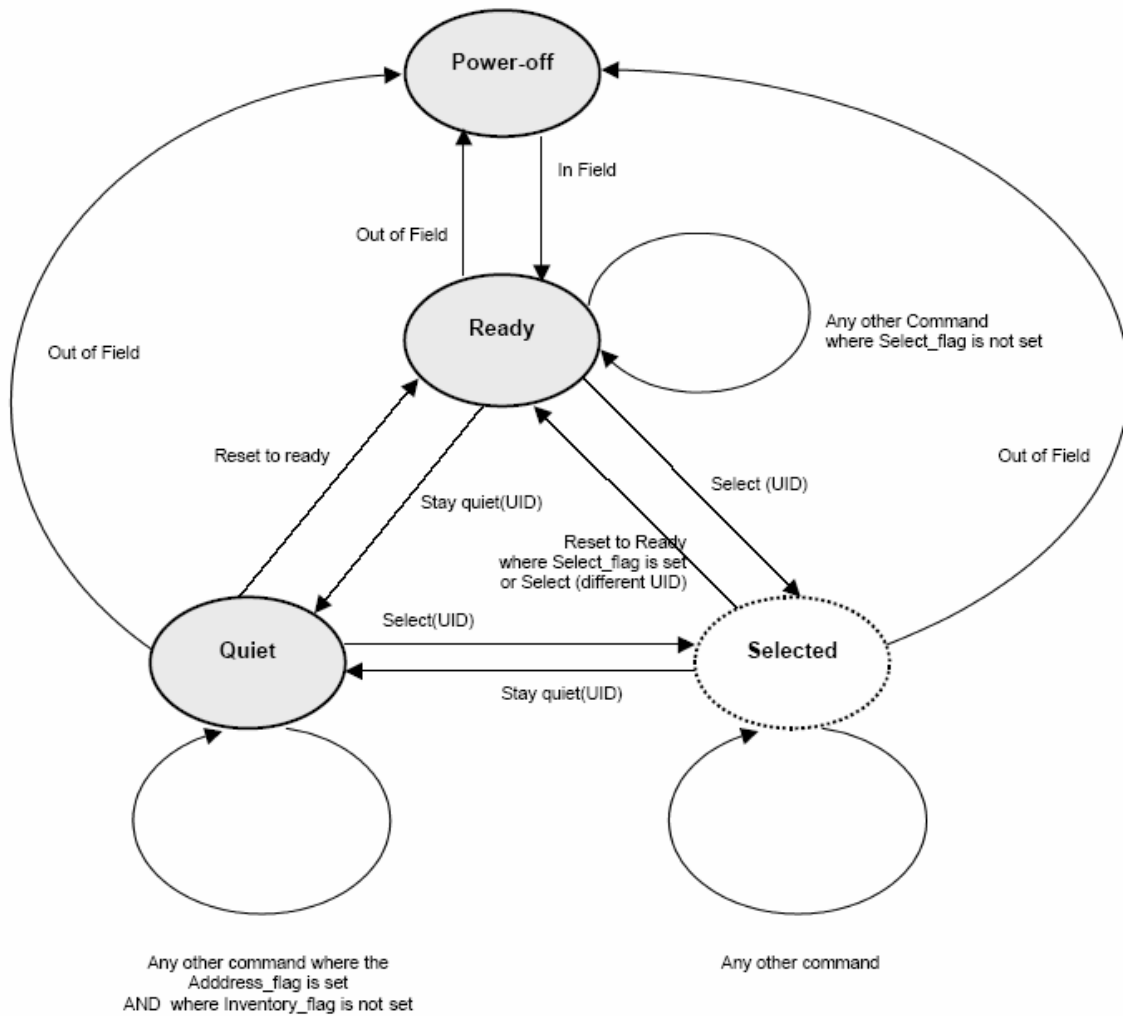


Figure 12 ISO15693 Tag's State Diagram

6.2.2 ISO14443A Tag's State Diagram

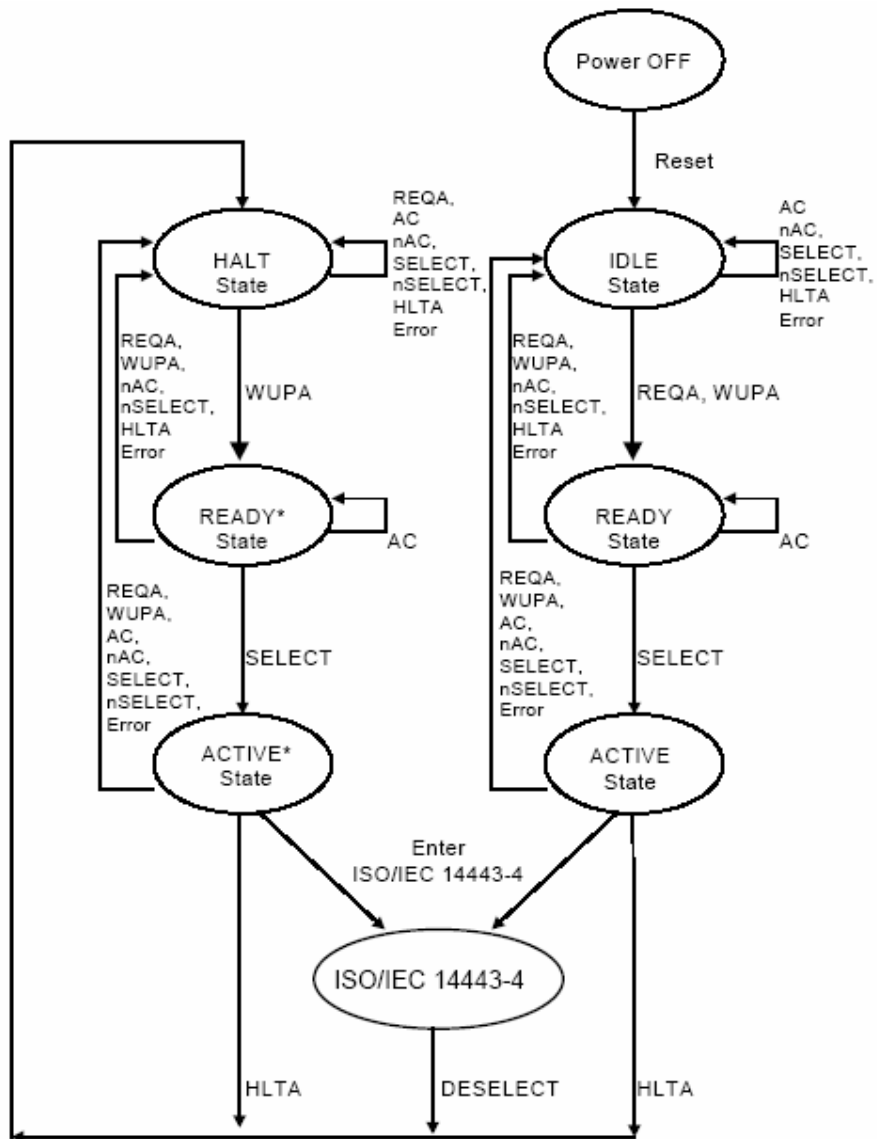


Figure 13 ISO14443A Tag's State Diagram

6.2.3 ISO14443B Tag's State Diagram

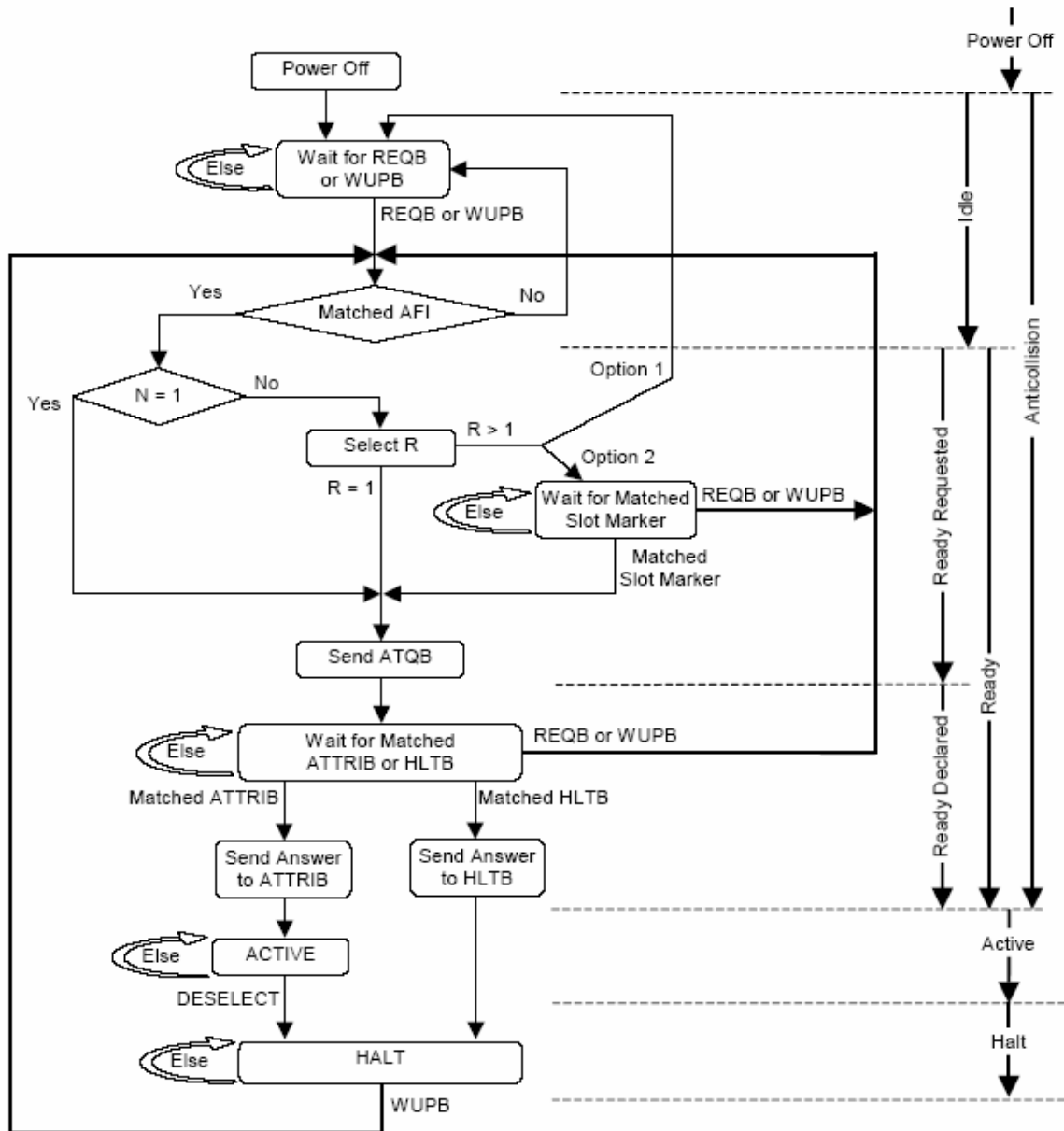


Figure 14 ISO14443B Tag's State Diagram

7. Reference

- ISO15693, Part 3: Anti-collision and transmission protocol
- ISO14443, Part 3: Initialization and anti-collision
- ISO18092, Near Field Communication — Interface and Protocol (NFCIP-1)
- MSR3100 Datasheet