RD220 Serial USB RFID Reader Protocol Manual

Document Version

1.1

Revision History

Revision	Date	Firmware Version	Description/ Change / Updated / Comment
1.0	January 2013	0103	1st Release
1.1	May 2013	0107	Add Hid Key keyboard Set Add auto read UID Add support to read Chinese identity card UID

Contents

1. Protocol Package 3
1.1 Command Frame Format 3
1.1.1 Meaning of byte in Command Frame Format 3
1.1.2 Dev_ID 3
1.1.3 CMD Category 3
1.2 Response frame format 4
1.2.1 Meaning of bytes in Response Frame 4
1.2.2 Response flag (Resp) 5
2. Reader and Reader IC Configuration Command (0x00) 6
2.1 Reader and Reader IC Configuration Command overview (0x00 / 0x01) 6
2.2 Reader Configuration Command (0x00 / 0x01) 7
2.3 Reader-Configuration-Command Response (0x00 / 0x01) 8
3. Polling mode (0x00 + 0xEx) 9
3.1 Reader Configuration Polling mode Command overview 9
3.2 Reader Configuration Polling Command 9
3.3 Reader Configuration Polling Command Response (0x00) 9
3.4 Steps how to configure reader in operating polling mode 9
4. ISO14443A and MIFARE Command (0x0A) 10
4.1 ISO14443A Command Overview (0x0A) 10
4.2 ISO14443A Command (0x0A) 11
4.3 ISO14443A-Command Response (0x0A) 12
4.4 MIFARE-Command Overview (0x0A) 13
4.5 MIFARE Command (0x0A) 14
4.6 MIFARE-Command response (0x0A) 15
4.7 MIFARE PLUS-Command Overview (0x0A) 16
4.8 MIFARE PLUS Command (0x0A) 17
4.9 MIFARE PLUS-Command response (0x0A) 18
5. ISO14443B Command (0x0B) 19
5.1 ISO14443B-Command Overview (0x0B) 19
5.2 ISO14443B Command (0x0B) 20
5.3 ISO14443B-Command Response (0x0B) 21
6. ISO15693 Command (0x0D) 22
6.1 ISO15693-Command overview (0x0D) 22
6.2 ISO15693-Command (0x0D) 23
6.3 ISO15693-Command Response (0x0D) 25
7. Felica Command (0x0C) 28
7.1 Felica-Command overview (0x0C) 28
7.2 Felica -Command (0x0C) 29
7.3 Felica -Command Response (0x0C) 30

1. Protocol Package

1.1 Command Frame Format

Name	SOP	LENG-H	LENG-L	Seq Num	Dev_ID	CMD Category	CMD	Data[0]… Data[n-1]	LRC
Values	0xAA	0x00	0x00						
No. Byte	1-byte	1-byte	1-byte	1-byte	1-byte	1-byte	1-byte	n-byte	1-byte

1.1.1 Meaning of byte in Command Frame Format

Name	Meaning
SOP	Start-of-Package byte (0xAA)
LENG-H	High byte of packet length counting from sequence number to Data[n-1]
LENG-L	Low byte of packet length counting from sequence number to Data[n-1]
Seq Number	Sequence number of Package
Dev_ID	Device ID byte : Silence bit (1Bit MSB) + Device_ID [(1)] (7 Bit)
CMD Category [(4)]	Command Category byte to specify operating standard or reader setup mode
CMD	Command byte in specified Command Category standard
Data[0] … Data[n-1]	Data bytes
LRC	Check sum of the packet which is XORing result from LENGTH to Data[n-1]

(1) The Device_ID can be between 0x00 and 0x7F. The detail of **Dev_ID** is shown in table below.

1.1.2 Dev_ID

Silence Bit [(2)] Dev_ID[7]	**ID** Dev_ID [6:0]	**Meaning**
0	0x00 [(3)]	All devices that receive command operate and respond back to host with Dev_ID of 0x00.
0	0x01 – 0x7F	The ID-matched device operates and responds back to host.
1	-	The operating device will not respond back to host

(2) The silence bit is an option for preventing data collision in reader network from simultaneous answer. User can retrieve last operating result/response from each device by using command “Get Last Response” (0x00-0x02).

(3) The ID (Dev_ID [6:0]) 0x00 can be used in broadcasting if there are multiple readers connected in a network.

(4) The commands are currently divided into 5 categories as shown in table below.

1.1.3 CMD Category

CMD Category	Meaning
0x00	Command Category for general reader function
0x0A	Command Category for ISO14443A and MIFARE/ MIFARE PLUS
0x0B	Command Category for ISO14443B
0x0D	Command Category for ISO15693

1.2 Response frame format

Name	SOP	LENG-H	LENG-L	Seq Num	Dev_ID	CMD Category	FBP CMD	Resp	Data[0]… Data[n-1]	LRC
Values	0xAA	0x00	0x00							
No.Byte	1-byte	1-byte	1-byte	1-byte	1-byte	1-byte	1-byte	1-byte	n-byte	1-byte

1.2.1 Meaning of bytes in Response Frame

Name	Meaning
SOP	Start-of-Package byte (0xAA)
LENG-H	High byte of packet length counting from sequence number to Data[n-1]
LENG-L	Low byte of packet length counting from sequence number to Data[n-1]
Seq Number	Sequence number of operated command packet
Dev_ID	Response device ID byte : 0 (1 Bit) + ID of operating Device (7 Bit)
CMD Category	Operated Command Category byte
FBP CMD	Operated command byte
Resp	Response flag of operated command
Data[0] … Data[n-1]	Response Data Bytes
LRC	Check sum of the packet which is XORing result from LENGTH to Data[n-1]

1.2.2 Response flag (Resp)

Category	Response Name	Resp	Meaning
No Error	Operation success	0x01	Operation is successful.
UART Error	Incomplete packet	0x10	Received data packet is incomplete.
	LRC error	0x11	LRC check sum verification fails.
	UART buffer full	0x12	UART receiver buffer is full.
Communication Protocol Error	Unknown command category	0x20	Device received undefined command category.
	Unknown command	0x21	Device received undefined command.
	Incorrect parameter	0x22	Device received incorrect parameter.
Board Error	Write hardware parameter error	0x30	Writing hardware parameters failed.
ISO14443A Error	HALT error	0xA0	There is a response after HALT command
	MIFARE/PLUS authentication error	0xA1	MIFARE Authentication fails.
	MIFARE access Error	0xA2	Perform an MIFARE authorized operation while crypto engine on reader IC is not turned on.
	MIFARE response Error	0xA3	Card response error message from MIFARE operation.
	ISO14443-4 Header Error	0xA4	ISO14443-4 PCB and CID response to inconsistent.
	MIFARE PLUS MAC Error	0xA5	MIFARE PLUS MAC validation error.
	MIFARE PLUS Response length _Error	0xA6	MIFARE PLUS response length to inconsistent.
	MIFARE PLUS PPC Error	0xA7	MIFARE PLUS Prepare Proximity Check invalid.
	MIFARE PLUS PC Error	0xA8	MIFARE PLUS Proximity Check invalid.
	MIFARE PLUS VPC Error	0xA9	MIFARE PLUS Verify Proximity Check invalid.
	MIFARE PLUS VCSL INF Error	0xAA	MIFARE PLUS Virtual Card Support Last response failed.
ISO14443B Error	ISO14443B Error	0xB0	Card response error message
ISO15693 Error	ISO15693 Error	0xD0	Card response error message from any operation.
RF Communication Error	No Response	0xE0	There is no card or card doesn't respond.
	Framing Error	0xE1	Received frame errors.
	Collision Error	0xE2	Data collision occurs. (ISO14443A / 15693)
	Parity Error	0xE3	Parity verification fails. (ISO14443A)
	CRC Error	0xE4	Received CRC verification fails.
	Invalid Response	0xE5	Received response is invalid or not in expected format.
Reader System Error	Buffer Overflow Error	0xF0	There is an error from FIFO overflow.
	Access E2 Error	0xF1	There is an error from accessing protected area in the EEPROM.
	Write E2 Error	0xF2	There is an error in programming EEPROM.
	Key Error	0xF3	There is an error from loading MIFARE key.
	RD220 Execution timeout	0xF4	The RD220 don't response within defined timeout

2. Reader and Reader IC Configuration Command (0x00)

2.1 Reader and Reader IC Configuration Command overview (0x00)

Command Name	CMD Category	CMD	Description
Get_Device_ID	0x00	0x00	Get current device ID
Write_Device_ID	0x00	0x01	Write or change device ID
Get_Last_Response	0x00	0x02	Get last response that the reader device answers to host
Test_Communication	0x00	0x03	Test communication by echoing back to host
Get_Firmware_Version	0x00	0x04	Get current firmware version
Buzzer_Active	0x00	0x05	Buzzer actions
LED_Active	0x00	0x06	LED actions
Active_LED_Blink	0x00	0x07	Action_LED blink
Get Keypad_Mode	0x00	0x10	Get HID Keyboard
Set Keypad_Mode	0x00	0x11	Set HID Keyboard
Get Keypad_Type	0x00	0x12	Get HID Keyboard read card type
Set Keypad_Type	0x00	0x13	Set HID Keyboard read card type
Get Keypad_ScanTime	0x00	0x14	Get HID Keyboard scan card time
Set Keypad_ScanTime	0x00	0x15	Set HID Keyboard scan card time
Get Keypad_SameCardTime	0x00	0x16	Get HID Keyboard same card delay time
Set Keypad_SameCardTime	0x00	0x17	Set HID Keyboard same card delay time
Get Keypad_KeySendTime	0x00	0x18	Get HID Keyboard send key time
Set Keypad_KeySendTime	0x00	0x19	Set HID Keyboard send key time
Get Keypad_SendType	0x00	0x1A	Get HID Keyboard send type
Set Keypad_SendType	0x00	0x1B	Set HID Keyboard send type
Reset_ASIC	0x00	0x20	Reset the reader IC
Reset_Device	0x00	0x21	Reset the device
Turn_RF_Field_On	0x00	0x30	Start 13.56-MHz carrier emission
Turn_RF_Field_Off	0x00	0x31	Stop 13.56-MHz carrier emission
Set_Timeout_for_Card_Response(1)	0x01	0x42	Define timeout period for the reader IC in case of no response from card.

(1) The “Set Timeout for card response” command should be performed after executing the following commands
1. A_Config_Reader_43A (0x0A – 0x00)
2. B_Config_Reader_43B (0x0B – 0x00)
3. D_Config_Reader_15693 (0x0D – 0x00).

Else, these commands will supersede the setting timeout value by default value.

2.2 Reader Configuration Command (0x00)

Command Name	CMD Category	CMD	Payload (Data[0]… Data[n-1])
Get_Device_ID	0x00	0x00	-
Write_Device_ID	0x00	0x01	Device_ID
Get_Last_Response	0x00	0x02	-
Test_Communication	0x00	0x03	Data_Echo(0) + … + Data_Echo(N-1)
Get_Firmware_Version	0x00	0x04	Get_Firmware_Version
Buzzer_Active	0x00	0x05	Active_Time
LED_Active	0x00	0x06	Led_No
Active_LED_Blink	0x00	0x07	Active_Time
Get Keypad_Mode	0x00	0x10	-
Set Keypad_Mode	0x00	0x11	Mode
Get Keypad_Type	0x00	0x12	-
Set Keypad_Type	0x00	0x13	Type
Get Keypad_ScanTime	0x00	0x14	-
Set Keypad_ScanTime	0x00	0x15	ScanTime
Get Keypad_SameCardTime	0x00	0x16	-
Set Keypad_SameCardTime	0x00	0x17	SameCardTime
Get Keypad_KeySendTime	0x00	0x18	-
Set Keypad_KeySendTime	0x00	0x19	SendTime
Get Keypad_SendType	0x00	0x1A	-
Set Keypad_SendType	0x00	0x1B	Format + Reverse + Add
Reset_ASIC	0x00	0x20	-
Reset_Device	0x00	0x21	Options 0xFF: Reset Device and All Parameters 0xXX: Reset Device
Turn_RF_Field_On	0x00	0x30	-
Turn_RF_Field_Off	0x00	0x31	-
Set_Timeout_for_Card_Response	0x00	0x42	Timeout Code 0x00: Previous setting 0x01: 1 ms 0x02: 2 ms 0x03: 4 ms 0x04: 8 ms 0x05: 16 ms 0x06: 32 ms 0x07: 64 ms 0x08: 128 ms 0x09: 256 ms 0x0A: 512 ms 0x0B: 1 s

2.3 Reader-Configuration-Command Response (0x00)

Command Name	CMD Category	FBP CMD	Resp	Payload (Data[0]… Data[n-1])
Get_Device_ID	0x00	0x00	0x01	Device_ID
Write_Device_ID	0x00	0x01	0x01	-
			Other	-
Get_Last_Response	0x00	0x02	0x01	CMD_Mode + CMD + Resp + Data_Resp(0) + … + Data_Resp(N-1)
Test_Communication	0x00	0x03	0x01	Data_Echo(0) + … + Data_Echo(N-1)
			Other	-
Get_Firmware_Version	0x00	0x04	0x01	Version(0) + … + Version(3)
Get Keypad_Mode	0x00	0x10	0x01	Mode
Set Keypad_Mode	0x00	0x11	0x01	-
			Other	-
Get Keypad_Type	0x00	0x12	0x01	Type
Set Keypad_Type	0x00	0x13	0x01	-
			Other	-
Get Keypad_ScanTime	0x00	0x14	0x01	ScanTime
Set Keypad_ScanTime	0x00	0x15	0x01	-
			Other	-
Get Keypad_SameCardTime	0x00	0x16	0x01	SameCardTime
Set Keypad_SameCardTime	0x00	0x17	0x01	-
			Other	-
Get Keypad_KeySendTime	0x00	0x18	0x01	SendTime
Set Keypad_KeySendTime	0x00	0x19	0x01	-
			Other	-
Get Keypad_SendType	0x00	0x1A	0x01	Format + Reverse + Add
Set Keypad_SendType	0x00	0x1B	0x01	-
			Other	-
Reset_ASIC	0x00	0x20	0x01	-
Reset_Device	0x00	0x21	0x01	-
Turn_RF_Field_Off	0x00	0x30	0x01	-
Turn_RF_Field_Off	0x00	0x31	0x01	-
Set_Timeout_for_Card_Response	0x00	0x42	0x01	-
			Other	-

3. Polling mode (0x00 + 0xEx)

3.1 Reader Configuration Polling mode Command overview

Command Name	CMD Mode	CMD	Description
Automatically_Read_UID	0x00	0xE5	A command according to the parameters automatically read the UID.
Enable_Continuous_Run	0x00	0xE6	Enable Continuous mode A command issued to the reader under this mode will be transmitted and receiver response periodically. After a command is operated, reader waits by time **Time_ms_H** and **Time_ms_L** defined from "Set_Op_Delay_Time_ms". The response is continuously sent back to host.
Disable_Continuous_Run	0x00	0xE7	Disable Continuous mode
Set_Op_Delay_Time_ms	0x00	0xE8	Set delay time after each operating command
Get_Op_Delay_Time_ms	0x00	0xE9	Get delay time after each operating command

3.2 Reader Configuration Polling Command

Command Name	CMD Mode	CMD	Payload (Data[0]… Data[n-1])
Automatically_Read_UID	0x00	0xE5	Card_type
Enable_Continuous_Run	0x00	0xE6	-
Disable_Continuous_Run	0x00	0xE7	-
Set_Op_Delay_Time_ms	0x00	0xE8	Time_ms_H(0) + Time_ms_L(1)
Get_Op_Delay_Time_ms	0x00	0xE9	-

3.3 Reader Configuration Polling Command Response (0x00)

Command Name	CMD Mode	FBP CMD	Resp	Payload (Data[0]… Data[n-1])
Automatically_Read_UID	0x00	0xE5	0x01	Card_type + UID(0) + … + UID(x)
Enable_Continuous_Run	0x00	0xE6	0x01	-
Disable_Continuous_Run	0x00	0xE7	0x01	-
Set_Op_ Delay _Time_ms	0x00	0xE8	0x01	-
Get_Op_Delay_Time_ms	0x00	0xE9	0x01	Time_ms_H(0) + Time_ms_L(1)

3.4 Steps how to configure reader in operating polling mode

1. Send "Enable_Continuous_Run" command to switch normal operating mode into polling mode.
2. Send a command user requires operating required operating periodically, e.g. inventory1slot, while the reader is set in polling mode.
3. Operating response from command in (2) is transmitted back periodically at interval set by command "Set_Op_Delay_Time_ms"
4. The operating command will continuously operates until the reader receives "Disable_Continuous_Run" command to stop the operation and switch back to normal mode.
5. User can set delay time after each operating command by using "Set_Op_Delay_Time_ms" command. Note that **Time_ms_H** and **Time_ms_L** are values defined in ms. For example, **Time_ms_H** of "01" and **Time_ms_L** of "00" represents time delay of 256 ms.
6. Note that the default value of the time delay is 50 ms.

Note that

A) Demonstration software does not show this mode due to large amount of data transferred.
However, user can check this capability by using serial monitoring software such HyperTerminal.

B) In this mode, host should have enough memory to accommodate transferred data.

4. ISO14443A and MIFARE Command (0x0A)

4.1 ISO14443A Command Overview (0x0A)

Command Name	Command Category	Command	Description
A_Config_Reader_43A	0x0A	0x00	Configure registers in the RD220 to serve ISO14443A transaction.
A_Config Speed_Reader	0x0A	0x01	Configure CODEC speed in the RD220 for ISO14443A
A_Get_Speed_Reader	0x0A	0x02	Get current CODEC speed from the RD220
A_Request	0x0A	0x10	Perform Request-A command in ISO14443A-3
A_WakeUp	0x0A	0x11	Perform WakeUp-A command in ISO14443A-3
A_Anticoll	0x0A	0x12	Perform anti-collision to get a single card matched condition in field. The condition is that the collide bit is 0 or 1.
A_Select	0x0A	0x13	Perform Select command in ISO14443A-3
A_Halt	0x0A	0x14	Perform Halt command in ISO14443A-3
A_RATS	0x0A	0x15	Perform RATS command in ISO14443A-4
A_PPS	0x0A	0x16	Perform PPS command in ISO14443A-4
A_Req_Anti_Sel	0x0A	0xB0	Perform consecutive ISO14443A-3 operations to get a single card in field and then select. If there is more cascade level, process is recursive until last level.
A_Transparent_With_CRC	0x0A	0xC0	Transmit ISO14443A arbitrary data with CRC appending at the end of the transmission packet
A_Transparent_Without_CRC	0x0A	0xC1	Transmit ISO14443A arbitrary data without CRC appending at the end of the transmission packet

4.2 ISO14443A Command (0x0A)

Command Name	Command Category	Command	Payload (Data[0]… Data[n-1])		
A_Config_Reader_43A	0x0A	0x00	-		
A_Config Speed_Reader	0x0A	0x01	Speed TX/RX		
			(Speed TX) Hi-Nibble	(Speed RX) Lo-Nibble	
			0000b	XXXX	106 kbps for TX
			0001b	XXXX	212 kbps for TX
			0010b	XXXX	424 kbps for TX
			0011b	XXXX	848 kbps for TX
			XXXX	0000b	106 kbps for RX
			XXXX	0001b	212 kbps for RX
			XXXX	0010b	424 kbps for RX
			XXXX	0011b	848 kbps for RX
A_Get_Speed_Reader	0x0A	0x02	-		
A_Request	0x0A	0x10	-		
A_WakeUp	0x0A	0x11	-		
A_Anticoll	0x0A	0x12	Cascade Level + CollMaskVal (1 byte)		
			Cascade Level	0x00	Cascade Level 1 (Default)
				0x01	Cascade Level 1
				0x02	Cascade Level 2
				0x03	Cascade Level 3
			CollMaskVal	0b	Select UID in which 1st collided bit is 0
				1b	Select UID in which 1st collided bit is 1
A_Select	0x0A	0x13	Cascade Level + UID(0) + UID(1) + UID(2) + UID(3)		
			Cascade Level	0x00	Cascade Level 1 (Default)
				0x01	Cascade Level 1
				0x02	Cascade Level 2
				0x03	Cascade Level 3
A_Halt	0x0A	0x14	-		
A_RATS	0x0A	0x15	FSID|CID (Parameter following ISO14443A-4)		
A_PPS	0x0A	0x16	CID + PPS0 + PPS1 (Parameter following ISO14443A-4)		
A_Req_Anti_Sel	0x0A	0xB0	Req Mode + CollMaskVal		
			Req Mode	0x00	Request A
				0x01	Wakeup A
			CollMaskVal	0x00	Select UID in which anti-collision bit is 0
				0x01	Select UID in which anti-collision bit is 1
A_Transparent_With_CRC	0x0A	0xC0	Timeout(1) + TXData(0) + … + TXData(N-1)		
A_Transparent_Without_CRC	0x0A	0xC1	Timeout(1) + TXData(0) + … + TXData(N-1)		

(1) Timeout code is defined in the command “Set Timeout for Card response” (0x01 + 0x42)

4.3 ISO14443A-Command Response (0x0A)

Command Name	Command Category	FBP Command	Resp	Payload (Data[0]… Data[n-1])
A_Config_Reader_43A	0x0A	0x00	0x01 Resp_Err	-
A_Config Speed_Reader	0x0A	0x01	0x01 Resp_Err	-
A_Get_Speed_Reader	0x0A	0x02	0x01 Resp_Err	Speed TX/RX (1)
A_Request	0x0A	0x10	0x01 Resp_Err	ATQA(0) + ATQA(1)
A_WakeUp	0x0A	0x11	0x01 Resp_Err	ATQA(0) + ATQA(1)
A_Anticoll	0x0A	0x12	0x01 Resp_Err	UID(0) + UID(1) + UID(2) + UID(3) (2)
A_Select	0x0A	0x13	0x01 Resp_Err	SAK
A_Halt	0x0A	0x14	0x01 Resp_Err	-
A_RATS	0x0A	0x15	0x01 Resp_Err	Reponse Parameter following ISO14443A-4
A_PPS	0x0A	0x16	0x01 Resp_Err	Reponse Parameter following ISO14443A-4
A_Req_Anti_Sel	0x0A	0xB0	0x01 Resp_Err	Cascade Level + SAK (3) + UID(0) + … + UID(N) (4) Error Location(5)
A_Transparent_With_CRC	0x0A	0xC0	0x01 Resp_Err	RespData(0) + … + RespData(N-1) -
A_Transparent_Without_CRC	0x0A	0xC1	0x01 Resp_Err	RespData(0) + … + RespData(N-1) -

(1) Speed TX/RX code is as shown in input parameter of "A_Config Speed_Reader" (0x0A + 0x01).
(2) The shown UID in response packet starts from least significant byte to most significant byte which reflects to what transmit from tag in chronological order.
(3) The SAK from this combo command is the last SAK from ultimate cascade level.
(4) The response UID from this command can be 4, 7 or 10 depending on cascade level.
(5) Error Location indicates state that error occurs. Possible value of Error Location is as list below

Error Location	Command that error occurs for Error Location1
0x01	A_Request / A_WakeUp
0x02	A_Anticoll
0x03	A_Select

4.4 MIFARE-Command Overview (0x0A)

Command Name	Command Category	Command	Description
A_Loadkey	0x0A	0x30	Load Key to key buffer of RD220 before performing MIFARE authentication
A_Loadkey_E2	0x0A	0x31	Load Key from the EEPROM of RD220 to the key buffer
A_Authentication	0x0A	0x32	Perform MIFARE authentication in specific card sector
A_Read_Block	0x0A	0x33	Read data from target block in authenticated sector
A_Write_Block	0x0A	0x34	Write data to target block in authenticated sector
A_Increment	0x0A	0x35	Increase value block in authenticated sector and store in the card buffer
A_Decrement	0x0A	0x36	Decrease value block in authenticated sector and store the in card buffer
A_Restore	0x0A	0x37	Load value from target block to the card buffer
A_Transfer	0x0A	0x38	Transfer result in the card buffer to target block
A_Req_Anti_Sel_LoadKey_Authent	0x0A	0xA0	Perform multiple commands from power-on to authentication MIFARE card.
A_Req_Anti_Sel_LoadKey_Authent_Rd	0x0A	0xA1	Perform multiple commands from power-on to read data in specific sector.
A_Req_Anti_Sel_LoadKey_Authent_Wr	0x0A	0xA2	Perform multiple commands from power-on to write data in specific sector.
A_Increment_Transfer	0x0A	0xA3	Increase value block and transfer result to another target block
A_Decrement_Transfer	0x0A	0xA4	Decrease value block and transfer result to another target block
A_Restore_Transfer	0x0A	0xA5	Load value from target block and transfer result to another target block. This command is similar to copying from one block to another block.

4.5 MIFARE Command (0x0A)

Command Name	Command Category	Command	Payload (Data[0]… Data[n-1])
A_Loadkey	0x0A	0x30	Key(0) + Key(1) + Key(2) + Key(3) + Key(4) + Key(5)
A_Loadkey_E2	0x0A	0x31	E2_Start_Address
A_Authentication	0x0A	0x32	Select_Key + Blk_Num +UID(0) + UID(1) + UID(2) + UID(3) Select_Key: 0x00 Key A; 0x01 Key B
A_Read_Block	0x0A	0x33	Blk_Num (Block to be read)
A_Write_Block	0x0A	0x34	Blk_Num (Block to be written) + BlkData(0) + … + BlkData(15)
A_Increment	0x0A	0x35	Blk_Num (Block to be increased) + Value(0) + Value(1) + Value(2) + Value(3) (1)
A_Decrement	0x0A	0x36	Blk_Num (Block to be decreased) + Value(0) + Value(1) + Value(2) + Value(3) (1)
A_Restore	0x0A	0x37	Blk_Num (Block in which content to be load to buffer)
A_Transfer	0x0A	0x38	Tran_Blk (Target block where content in buffer to be stored)
A_Req_Anti_Sel_LoadKey_Authent	0x0A	0xA0	Req Mode + CollMaskVal + Select Key + Key(0) + … + Key(5) + BlkNum(Block to be access) Req Mode: 0x00 Request A; 0x01 Wakeup A CollMaskVal: 0x00 Select UID in which anti-collision bit is 0; 0x01 Select UID in which anti-collision bit is 1 Select_Key: 0x00 Key A; 0x01 Key B
A_Req_Anti_Sel_LoadKey_Authent_Rd	0x0A	0xA1	Req Mode + CollMaskVal + Select Key + Key(0) + … + Key(5) + BlkNum (Block to be read) Req Mode: 0x00 Request A; 0x01 Wakeup A CollMaskVal: 0x00 Select UID in which anti-collision bit is 0; 0x01 Select UID in which anti-collision bit is 1 Select_Key: 0x00 Key A; 0x01 Key B
A_Req_Anti_Sel_LoadKey_Authent_Wr	0x0A	0xA2	Req Mode + CollMaskVal + Select Key + Key(0) + … + Key(5) + BlkNum (Block to be written) + Blk Dat(0) + … + Blk Dat(15) Req Mode: 0x00 Request A; 0x01 Wakeup A CollMaskVal: 0x00 Select UID in which anti-collision bit is 0; 0x01 Select UID in which anti-collision bit is 1 Select_Key: 0x00 Key A; 0x01 Key B
A_Increment_Transfer	0x0A	0xA3	Blk Num+ Value(0) + Value(1) + Value(2) + Value(3)* + Tran_Blk Tran_Blk: Target block where the result is transferred
A_Decrement_Transfer	0x0A	0xA4	Blk Num+ Value(0) + Value(1) + Value(2) + Value(3)* + Tran_Blk
A_Restore_Transfer	0x0A	0xA5	Blk Num + Tran_Blk

(1) Note that the Value(3) is a most significant byte of 4-bytes signed value of value block. So, value of Value(3) lying between 0x00 and 0x7F represents positive value. Value of value(3) lying between 0x80 and 0xFF represents negative value. Increment which will result in amount of final value beyond 0x7FFFFFFF is inhibited and cause error in operation. Also, decrement whic h will result in amount of final value below 0x80000000 is inhibited and cause error in operation.

4.6 MIFARE-Command response (0x0A)

Command Name	Command Category	FBP Command	Resp	Payload (Data[0]… Data[n-1])
A_Loadkey	0x0A	0x30	0x01 Resp_Err	-
A_Loadkey_E2	0x0A	0x31	0x01 Resp_Err	-
A_Authentication	0x0A	0x32	0x01 Resp_Err	-
A_Read_Block	0x0A	0x33	0x01 Resp_Err	BlkData(0) + … + BlkData(15)
A_Write_Block	0x0A	0x34	0x01 Resp_Err	-
A_Increment	0x0A	0x35	0x01 Resp_Err	
A_Decrement	0x0A	0x36	0x01 Resp_Err	
A_Restore	0x0A	0x37	0x01 Resp_Err	
A_Transfer	0x0A	0x38	0x01 Resp_Err	
A_Req_Anti_Sel_LoadKey_Authent	0x0A	0xA0	0x01 Resp_Err	UID(0) + UID(1) + UID(2) + UID(3) Error Location1(1)
A_Req_Anti_Sel_LoadK_Authent_Rd	0x0A	0xA1	0x01 Resp_Err	UID(0) + UID(1) + UID(2) + UID(3) + BlkData(0) + … + BlkData(15) Error Location1(1)
A_Req_Anti_Sel_LoadK_Authent_Wr	0x0A	0xA2	0x01 Resp_Err	UID(0) + UID(1) + UID(2) + UID(3) Error Location1(1)
A_Increment_Transfer	0x0A	0xA3	0x01 Resp_Err	- Error Location2(1)
A_Decrement_Transfer	0x0A	0xA4	0x01 Resp_Err	- Error Location2(1)
A_Restore_Transfer	0x0A	0xA5	0x01 Resp_Err	- Error Location2(1)

(1) The error location for combo command is shown in table below.

Error Location	Command that error occurs for Error Location1	Command that error occurs for Error Location2
0x01	A_Request / A_WakeUp	A_Increment / A_Decrement / A_Restore
0x02	A_Anticoll	A_Transfer
0x03	A_Select	-
0x04	A_LoadKey	-
0x05	A_Authent	-
0x06	A_Read Block / A_Write Block	-

4.7 MIFARE PLUS-Command Overview (0x0A)

Command Name	Command Category	Command	Description
A_MP_Write_Perso	0x0A	0x40	This command is used to change the data and AES keys from the initial delivery configuration to a customer specific value
A_MP_Commit_Perso	0x0A	0x41	This command is used to finalize the personalization
A_MP_First_Authentication	0x0A	0x42	Perform first authentication in specific card sector
A_MP_Follow_Authentication	0x0A	0x43	Following authenticate
A_MP_Read_Block	0x0A	0x44	Read data from target block in authenticated sector
A_MP_Write_Block	0x0A	0x45	Write data to target block in authenticated sector
A_MP_Proximity_Check	0x0A	0x4A	Perform the precise measurement for the proximity check
A_MP_Virtual_Card_Support	0x0A	0x4B	Check, if the Virtual Card Concept is supported
A_MP_Virtual_Card_Support_Last	0x0A	0x4C	Check if the Virtual Card Concept is supported, communicate reader capabilities and retrieve the UID
A_MP_Select_Virtual_Card	0x0A	0x4D	Select the Virtual Card
A_MP_Deselect_Virtual_Card	0x0A	0x4E	Deselect the Virtual Card

4.8 MIFARE PLUS Command (0x0A)

Command Name	Command Category	Command	Payload (Data[0]… Data[n-1])
A_MP_Write_Perso	0x0A	0x40	Select ISO14443A command + Blk_Num + Key (0) + … + Key (15) Select ISO14443A: 0x00 ISO14443A-3 command; 0x01 ISO14443A-4 command
A_MP_Commit_Perso	0x0A	0x41	Select ISO14443A command Select ISO14443A: 0x00 ISO14443A-3 command; 0x01 ISO14443A-4 command
A_MP_First_Authentication	0x0A	0x42	Blk_Num + Key (0) + … + Key (15) + PCDcapsLen + PCDcaps
A_MP_Follow_Authentication	0x0A	0x43	Blk_Num + Key (0) + … + Key (15)
A_MP_Read_Block	0x0A	0x44	Blk_Num + Ext + Options Options: 0x00 MAC/Cmd + Encrypted + No MAC/Resp; 0x01 MAC/Cmd + Encrypted + MAC/Resp; 0x02 MAC/Cmd + Plain + No MAC/Resp; 0x03 MAC/Cmd + Plain + MAC/Resp; 0x04 No MAC/Cmd + Encrypted + No MAC/Resp; 0x05 No MAC/Cmd + Encrypted + MAC/Resp; 0x06 NO MAC/Cmd + Plain + No MAC/Resp; 0x07 NO MAC/Cmd + Plain + MAC/Resp
A_MP_Write_Block	0x0A	0x45	Blk_Num + Options + BlkData(0) + … + BlkData(15) Options: 0x00 Encrypted + No MAC/Resp; 0x01 Encrypted + MAC/Resp; 0x02 Plain + No MAC/Resp; 0x03 Plain + MAC/Resp
A_MP_Proximity_Check	0x0A	0x4A	Iteration + Select Use Kmac + Key (0) + … + Key (15) Select Use Kmac: 0x00 Use Key; 0x01 Use MAC Key
A_MP_Virtual_Card_Support	0x0A	0x4B	IID (0) + … + IID (15)
A_MP_Virtual_Card_Support_Last	0x0A	0x4C	VCKenc (0) + … + VCKenc (15) + IID (0) + … + IID (15) + VCKmac (0) + … + VCKmac (15) + PCDcapsLen + PCDcaps
A_MP_Select_Virtual_Card	0x0A	0x4D	SVCKey (0) + … + SVCKey (15) + PICCaps + UIDLen + UID
A_MP_Deselect_Virtual_Card	0x0A	0x4E	

4.9 MIFARE PLUS -Command response (0x0A)

Command Name	Command Category	FBP Command	Resp	Payload (Data[0]… Data[n-1])
A_MP_Write_Perso	0x0A	0x40	0x01 Resp_Err	-
A_MP_Commit_Perso	0x0A	0x41	0x01 Resp_Err	-
A_MP_First_Authentication	0x0A	0x42	0x01 Resp_Err	-
A_MP_Follow_Authentication	0x0A	0x43	0x01 Resp_Err	-
A_MP_Read_Block	0x0A	0x44	0x01 Resp_Err	BlkData (0) + ... + BlkData (16) + … + ExtBlkData (0) + ... + ExtBlkData (16)
A_MP_Write_Block	0x0A	0x45	0x01 Resp_Err	-
A_MP_Proximity_Check	0x0A	0x4A	0x01 Resp_Err	-
A_MP_Virtual_Card_Support	0x0A	0x4B	0x01 Resp_Err	-
A_MP_Virtual_Card_Support_Last	0x0A	0x4C	0x01 Resp_Err	RndQ (0) + ... + RndQ (5) or RndQ (8)
A_MP_Select_Virtual_Card	0x0A	0x4D	0x01 Resp_Err	-
A_MP_Deselect_Virtual_Card	0x0A	0x4E	0x01 Resp_Err	-

5. ISO14443B Command (0x0B)

5.1 ISO14443B-Command Overview (0x0B)

Command Name	Command Category	Command	Description
B_Config_Reader_43B	0x0B	0x00	Configure registers in the RD220 to serve ISO14443B transaction.
B_Config_Speed_Reader	0x0B	0x01	Configure CODEC speed in the RD220 for ISO14443B
B_Get_Speed_Reader	0x0B	0x02	Get current CODEC speed from the RD220
B_Request	0x0B	0x10	Perform Request-B command and anti-collision process to get a single card in ISO14443B-3. The first card responding complete UID is reported.
B_WakeUp	0x0B	0x11	Perform Wakeup-B command in ISO14443B-3
B_ATTRIB	0x0B	0x12	Perform ATTRIB command in ISO14443B-3
B_Halt	0x0B	0x13	Perform Halt command in ISO14443B-3
B_GUID	0x0B	0x14	Read Chinese identity card
B_Transparent_With_CRC	0x0B	0xC0	Transmit ISO14443B arbitrary data with CRC appending at the end of the transmission packet
B_Transparent_WithOut_CRC	0x0B	0xC1	Transmit ISO14443B arbitrary data without CRC appending at the end of the transmission packet

5.2 ISO14443B Command (0x0B)

Command Name	Command Category	Command	Payload (Data[0]… Data[n-1])		
B_Config_Reader_43B	0x0B	0x00	-		
B_Config_Speed_Reader	0x0B	0x01	Speed TX/RX (Speed TX) Hi-Nibble	(Speed RX) Lo-Nibble	
			0000b	XXXX	106 kbps for TX
			0001b	XXXX	212 kbps for TX
			0010b	XXXX	424 kbps for TX
			0011b	XXXX	848 kbps for TX
			XXXX	0000b	106 kbps for RX
			XXXX	0001b	212 kbps for RX
			XXXX	0010b	424 kbps for RX
			XXXX	0011b	848 kbps for RX
B_Get_Speed_Reader	0x0B	0x02	-		
B_Request	0x0B	0x10	AFI + Num_of_Slots		
			Num_of_Slots	0x00	1 slot
				0x01	2 slots
				0x02	4 slots
				0x03	8 slots
				0x04	16 slots
B_WakeUp	0x0B	0x11	AFI + Num_of_Slots		
B_ATTRIB[1]	0x0B	0x13	PUPI(0) + … + PUPI(3) + Param(1) + … + Param(4) + Highlayer(0) + … + Highlayer(n-1)		
B_Halt	0x0B	0x14	PUPI(0) + PUPI(1) + PUPI(2) + PUPI(3)		
B_GUID	0x0B	0x15	-		
B_Transparent_With_CRC	0x0B	0xC0	Timeout[2] + TXData(0)+ .. + TXData(N-1)		
B_Transparent_WithOut_CRC	0x0B	0xC1	Timeout[2]+ TXData(0)+ .. + TXData(N-1)		

(1) Input parameters for this command is defined in the ISO14443B standard.

(2) The timeout code is defined in the command “Set Timeout for Card response” (0x01 + 0x42)

5.3 ISO14443B-Command Response (0x0B)

Command Name	Command Category	FBP Command	Resp	Payload (Data[0]… Data[n-1])	
B_Config_Reader_43B	0x0B	0x00	0x01 Resp_Err	- -	
B_Config_Speed_Reader	0x0B	0x01	0x01 Resp_Err	- -	
B_Get_Speed_Reader	0x0B	0x02	0x01 Resp_Err	Speed TX/RX -	
B_Request(1)	0x0B	0x10	0x01	PUPI(0) + PUPI(1) + PUPI(2) + PUPI(3) + App_Data (4 Bytes) + Protocol_Info (3 Bytes)	
				App_Data	AFI (1st byte) CRC_B(AID) (2nd & 3rd bytes) Numbers of App (4th byte)
				Protocol_Info	Bit_Rate_Capability (1st byte) Max_Frame_Size | Protocol_Type (2nd byte) FWI | ADC | FO (3rd byte)
			Resp_Err	-	
B_WakeUp(1)	0x0B	0x11	0x01	PUPI(0) + PUPI(1) + PUPI(2) + PUPI(3) + App_Data (4 Bytes) + Protocol_Info (3 Bytes)	
				App_Data	AFI (1st byte) CRC_B(AID) (2nd & 3rd bytes) Numbers of App (4th byte)
				Protocol_Info	Bit_Rate_Capability (1st byte) Max_Frame_Size | Protocol_Type (2nd byte) FWI | ADC | FO (3rd byte)
			Resp_Err	-	
B_ATTRIB(1)	0x0B	0x12	0x01 Resp_Err	MBLI|CID + HighlayerRes(0) + ..+ HighlayerRes(n-1)	
B_Halt	0x0B	0x13	0x01 Resp_Err	-	
B_GUID	0x0B	0x14	0x01 Resp_Err	GUID(0) + … + GUID(9)	
B_Transparent_With_CRC	0x0B	0xC0	0x01 Resp_Err	RespData(0) + … + RespData(N-1) -	
B_Transparent_WithOut_CRC	0x0B	0xC1	0x01 Resp_Err	RespData(0) + … + RespData(N-1) -	

(1) For more information about output parameters, please refer to ISO14443B standard.

6. ISO15693 Command (0x0D)

6.1 ISO15693-Command overview (0x0D)

The commands in this section are provided as described in mandatory and optional command in ISO15693-3.

Command Name	Command Category	Command	Description
D_Config_Reader_15693	0x0D	0x00	Configure registers in the RD220 to serve ISO15693 transaction
D_Config_Speed_Reader	0x0D	0x01	Configure CODEC speed in the RD220 for ISO15693
D_Get_Speed_Reader	0x0D	0x02	Get current CODEC speed from the RD220
Inventory 1 slot	0x0D	0x10	Perform Inventory 1-slot command
Inventory 16 slot (1)	0x0D	0x11	Perform Inventory 16-slot command
Stay Quiet	0x0D	0x12	Perform Stay-Quiet command
Read Single Blocks	0x0D	0x13	Read data Block
Write Single Blocks	0x0D	0x14	Write data Block
Lock Block	0x0D	0x15	Lock data block
Read Multiple Blocks	0x0D	0x16	Read multiple Block
Write Multiple Blocks	0x0D	0x17	Write multiple Blocks
Select	0x0D	0x18	Perform select command
Reset to Ready	0x0D	0x19	Perform Reset-to-Ready command
Write AFI	0x0D	0x1A	Write AFI value
Lock AFI	0x0D	0x1B	Lock AFI value
Write DSFID	0x0D	0x1C	Write DSFID value
Lock DSFID	0x0D	0x1D	Lock DSFID value
Get System Information	0x0D	0x1E	Get system information
Get Multiple Block Security status	0x0D	0x1F	Get multiple block security status
D_Transparent_With_CRC	0x0D	0xC0	Transmit ISO15693 arbitrary data with CRC appending at the end of the transmission packet
D_Transparent_WithOut_CRC	0x0D	0xC1	Transmit ISO15693 arbitrary data without CRC appending at the end of the transmission packet
Send_1_Pulse	0x0D	0xC2	Send 1 pulse to indicate next slot in anti-collision process

(1) For the inventory 16 slot, the reader transmits pulses to indicate end of each slot automatically and reports all response present in each slot both successful and failed responses. There is no need to transmit “Send_1_Pulse” manually.

6.2 ISO15693-Command (0x0D)

Command Name	Command Category	Command	Payload (Data[0]… Data[n-1])
D_Config_Reader_15693	0x0D	0x00	-
D_Config_Speed_Reader	0x0D	0x01	Speed TX/RX (Speed TX) Hi-Nibble / (Speed RX) Lo-Nibble 0000b / XXXX : 1 out of 256 (1.65 kbits/s) 0001b / XXXX : 1 out of 4 (26.48 kbits/s) XXXX / 0000b : 1 Sub Low rate (6.62 kbits/s) XXXX / 0001b : 1 Sub High rate(26.48 kbits/s) XXXX / 0010b : 1 Sub Ultra high rate (52.96 kbits/s) XXXX / 0011b : 2 Sub Low rate(6.67 kbits/s) XXXX / 0100b : 2 Sub High rate(26.69 kbits/s)
D_Get_Speed_Reader	0x0D	0x02	-
Inventory 1 slot	0x0D	0x10	Inventory_Mode(1) + [AFI](5) + MaskLen + MaskUIDValue0 + … + MaskUIDValue7
Inventory 16 slot	0x0D	0x11	Inventory_Mode(1) + [AFI](5) + MaskLen + MaskUIDValue0 + … + MaskUIDValue7
Stay Quiet	0x0D	0x12	Non Inventory Mode(2) + UID(0) + … + UID(7)
Read Single Blocks	0x0D	0x13	Non Inventory Mode(2) + [UID(0) + … + UID(7)](4) + BlockAddr
Write Single Blocks	0x0D	0x14	Non Inventory Mode(2) + [UID(0) + … + UID(7)](4) + BlockAddr + Data(0) + Data(1) + .. + Data(Block Len-1)
Lock Block	0x0D	0x15	Non Inventory Mode(2) + [UID(0) + … + UID(7)](4) + BlockAddr
Read Multiple Blocks	0x0D	0x16	Non Inventory Mode(2) + [UID(0) + … + UID(7)](4) + BlockAddr + Num_of_Block
Write Multiple Blocks	0x0D	0x17	Non Inventory Mode(2) + [UID(0) + … + UID(7)](4) + Block_Size + BlockAddr + Num_of_Block + Data(0) + Data(1) + .. + Data(Block Len-1)
Select	0x0D	0x18	Non Inventory Mode(2) + UID(0) + … + UID(7) +
Reset to Ready	0x0D	0x19	Non Inventory Mode(2) + [UID(0) + … + UID(7)](4) +
Write AFI	0x0D	0x1A	Non Inventory Mode(2) + [UID(0) + … + UID(7)](4) + AFI Value
Lock AFI	0x0D	0x1B	Non Inventory Mode(2) + [UID(0) + … + UID(7)](4) +
Write DSFID	0x0D	0x1C	Non Inventory Mode(2) + [UID(0) + … + UID(7)](4) + DSFID Value
Lock DSFID	0x0D	0x1D	Non Inventory Mode(2) + [UID(0) + … + UID(7)](4) +
Get System Information	0x0D	0x1E	Non Inventory Mode(2) + [UID(0) + … + UID(7)](4) +
Get Multiple Block Security status	0x0D	0x1F	Non Inventory Mode(2) + [UID(0) + … + UID(7)](4) + BlockAddr + Num_of_Block
D_Transparent_With_CRC	0x0D	0xC0	Timeout(3) + TXData(0)+ .. + TXData(N-1)
D_Transparent_WithOut_CRC	0x0D	0xC1	Timeout(3) + TXData(0)+ .. + TXData(N-1)
Send_1_Pulse	0x0D	0xC2	-

(1) The Inventory Mode byte is defined as shown in table below.

Bit	7	6	5	4	3	2	1	0
Name	RFU Flag	Option Flag	Protocol Extension Flag	-	AFI mode			
					0001b	Check AFI		
					0000b	Not Check AFI		

If AFI mode = 0001b, AFI byte must be supplied in input command packet. Depending on special mode required by each card, RFU Flag, Option Flag and Protocol Extension can be set to compose a flag following ISO15693.

(2) The Non Inventory Mode is defined as shown in table below.

Bit	7	6	5	4	3	2	1	0
Name	RFU Flag	Option Flag	Protocol Extension Flag	-	Operation mode			
					0010b	Select Mode		
					0001b	Address Mode		
					0000b	Non Address Mode		

(3) The timeout code is defined in the command “Set Timeout for Card response” (0x00 + 0x42)

(4) For these command, UID is optional. If operation mode is Address Mode, UID must be supplied in payload.

(5) Depending on setup flag in Inventory_Mode and Non_Inventory_Mode, Value in bracket [ ] is an optional.

6.3 ISO15693-Command Response (0x0D)

Command Name	Command Category	Command	Resp	Payload (Data[0]… Data[n-1])
D_Config_Reader_15693	0x0D	0x00	0x01	-
			Resp_Err	-
D_Config_Speed_Reader	0x0D	0x01	0x01	-
			Resp_Err	-
D_Get_Speed_Reader	0x0D	0x02	0x01	Speed TX/RX[1]
			Resp_Err	-
Inventory 1 slot	0x0D	0x10	0x01	DSFID + UID(0) + … + UID(7) [2]
			Resp_Err	-
Inventory 16 slot [3]	0x0D	0x11	0x01	Slot_Num + Slot_Resp + Slot_Len + [DSFID + UID] / *Remaining data in ASIC FIFO+ + … Slot_Num + Slot_Resp + Slot_Len + [DSFID + UID] / *Remaining data in ASIC FIFO+ + … Slot_Num: Number of slot that there is tag response Slot_Resp: Response of packet as defined in table 1.2.2 in each slot If Slot_Resp = 0x01, "DSFID + UID" is reported, else remaining data in ASIC FIFO is reported. Slot_Len: Length of data in each sub slot
			Resp_Err	-
Stay Quiet	0x0D	0x12	0x01	-
			Resp_Err	-
Read Single Blocks	0x0D	0x13	0x01	Block Security[4] + Data(0) + … + Data(N)
			Resp_Err	-
			0xD0	Error Code [5]
Write Single Blocks	0x0D	0x14	0x01	-
			Resp_Err	-
			0xD0	Error Code[5]
Lock Block	0x0D	0x15	0x01	-
			Resp_Err	-
			0xD0	Error Code[5]
Read Multiple Blocks	0x0D	0x16	0x01	Block Security[4] + Data[0](0) + … + Data[0](N) + Block Security[4] + Data*1+(0) + … + Data*1+(N) + …
			Resp_Err	-
			0xD0	Error Code[5]
Write Multiple Blocks	0x0D	0x17	0x01	-
			Resp_Err	-
			0xD0	Error Code[5]
Select	0x0D	0x18	0x01	-
			Resp_Err	-
			0xD0	Error Code[5]
Reset to Ready	0x0D	0x19	0x01	-
			Resp_Err	-
			0xD0	Error Code[5]
Write AFI	0x0D	0x1A	0x01	-
			Resp_Err	-
			0xD0	Error Code[5]
Lock AFI	0x0D	0x1B	0x01	-
			Resp_Err	-
			0xD0	Error Code[5]

Write DSFID	0x0D	0x1C	0x01	-
			Resp_Err	-
			0xD0	Error Code[5]
Lock DSFID	0x0D	0x1D	0x01	-
			Resp_Err	-
			0xD0	Error Code[5]
Get System Information	0x0D	0x1E	0x01	Info Flag [6] + UID(0) + … + UID(7) + DSFID + AFI + VICC memory size [7](2 byte) + IC_ref [8]
			Resp_Err	-
			0xD0	Error Code[5]
Get Multiple Block Security status	0x0D	0x1F	0x01	BSS(0) + … + BSS(N)[9]
			Resp_Err	-
			0xD0	Error Code[5]
D_Transparent_With_CRC	0x0D	0xC0	0x01	RespData(0) + … + RespData(N-1)
			Resp_Err	-
D_Transparent_WithOut_CRC	0x0D	0xC1	0x01	RespData(0) + … + RespData(N-1)
			Resp_Err	-
Send_1_Pulse	0x0D	0xC2	0x01	RespData(0) + … + RespData(N-1)
			Resp_Err	-

(1) Speed TX/RX is as shown in input parameter of "D_Config_Speed_Reader".

(2) The UID in response packet starts from least significant byte to most significant byte, which reflects to what transmit from tag in chronological order.

(3) For Inventory 16 slot, successful and failed response presenting in each slot both are reports.
Examples 1, RX > "AA 00 1D 7D 00 0D 11 01 00 01 09 00 20 E1 22 0C 00 01 04 E0 09 01 09 00 69 96 23 0C 00 01 04 E0 4B".
This response can be decomposed as shown in table below. There are 2 cards response completely in slot 0 and slot 9.

Header	Length-H Length-L	SEQ NUM	DEV ID	CMD MODE & CMD	Resp	Slot_Num	Slot_Resp	Slot_Len	[DSFID + UID] / [Remaining data in ASIC FIFO]		LRC
AA	00 1D	7D	00	0D 11	01						
						00	01	09	00	20 E1 22 0C 00 01 04 E0	
						09	01	09	00	69 96 23 0C 00 01 04 E0	
											4B

Examples 2, RX> "AA 00 29 54 00 0D 11 01 00 01 09 00 20 E1 22 0C 00 01 04 E0 01 01 09 00 D1 DD 22 0C 00 01 04 E0 04 01 09 00 74 96 23 0C 00 01 04 E0 88". This response can be decomposed as shown in table below. There are three cards response in three different slots in this example.

Header	Length-H Length-L	SEQ NUM	DEV ID	CMD MODE & CMD	Resp	Slot_Num	Slot_Resp	Slot_Len	[DSFID + UID] / [Remaining data in ASIC FIFO]		LRC
AA	00 29	54	00	0D 11	01						
						00	01	09	00	20 E1 22 0C 00 01 04 E0	
						01	01	09	00	D1 DD 22 0C 00 01 04 E0	
						04	01	09	00	74 96 23 0C 00 01 04 E0	
											88

Examples 3, RX> "AA 00 20 56 00 0D 11 01 00 01 09 00 20 E1 22 0C 00 01 04 E0 04 E2 0C 00 00 04 00 00 00 00 00 00 00 00 00 87".
This response can be decomposed as shown in table below.

Header	Length-H Length-L	SEQ NUM	DEV ID	CMD MODE & CMD	Resp	Slot_Num	Slot_Resp	Slot_Len	[DSFID + UID] / [Remaining data in ASIC FIFO]		LRC
AA	00 20	56	00	0D 11	01						
						00	01	09	00	20 E1 22 0C 00 01 04 E0	
						04	E2	0C	00 00 04 00 00 00 00 00 00 00 00 00		
											87

For the example 3, there are a successful response in one slot and collision in another slot. In the slot collision occurs, collision response is “00 00 04 00 00 00 00 00 00 00 00” consisting of Response flag (1 bytes), DSFID (1 bytes), UID (8 bytes) and CRC (2 byte) respectively. Note that data after collision is masked to zero. User can further discriminate collided tags by reissuing Inventory 16-slot command with mark value equal to the number of slot that collision occurred and mark length of “0x04”. In this case, mark value is “0x04”.

(4) Block security presents, if option flag in input command packet is set. (5)
Resp of “0xD0” is reported, if Bit Error flag in ISO15693 response is set.
(6) Info Flag is one byte information following ISO15693. The bit detail is shown in table below.

Bit	7	6	5	4	3	2	1	0
Name	0	0	0	0	IC reference	VICC memory size	AFI	DFSID

(7) VICC memory size is two byte information following ISO15693. The bit detail is shown in table below.

Byte	12th byte in pay load								13th byte in pay load							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Name	Number of blocks								0	0	0	Block size in bytes				

Block size is expressed in number of bytes on 5 bits, allowing to specify up to 32 bytes i.e. 256 bits. It is one less than the actual number of bytes. E.g. a value of '1F' indicates 32 bytes, a value of '00' indicates 1 byte.

Number of blocks is on 8 bits, allowing to specify up to 256 blocks. It is one less than the actual number of blocks. E.g. a value of 'FF' indicates 256 blocks, a value of '00' indicates 1 block.

(8) IC_ref is an 8-bits IC reference and its meaning is defined by IC manufacturer.
(9) BSS is block security status.

7. Felica Command (0x0C)

7.1 Felica-Command overview (0x0C)

Command Name	Command Category	Command	Description
C_Config_Reader_Felica	0x0C	0x00	Configure registers in the RD220 to serve Felica transaction
C_Config_Speed_Reader	0x0C	0x01	Configure CODEC speed in the RD220 for Felica
C_Get_Speed_Reader	0x0C	0x02	Get current CODEC speed from the Felica
C_Polling	0x0C	0x10	Perform Felica Polling command
C_Transparent_With_CRC	0x0C	0xC0	Transmit arbitrary data with CRC appending at the end of the transmission packet

7.2 Felica -Command (0x0C)

Command Name	Command Category	Command	Payload (Data[0]… Data[n-1])		
C_Config_Reader_Felica	0x0C	0x00	-		
C_Config_Speed_Reader	0x0C	0x01	Speed TX/RX (Speed TX) Hi-Nibble	(Speed RX) Lo-Nibble	
			0000b	XXXX	212 kbps for TX
			XXXX	0000b	212 kbps for RX
C_Get_Speed_Reader	0x0C	0x02	-		
C_Polling	0x0C	0x10	System Code(0)[1] + System Code(1)[1] + Reserved[2] + #Slot		
			#Slot	0x00	1 slot
				0x01	2 slots
				0x02	4 slots
				0x03	8 slots
				0x04	16 slots
C_Transparent_With_CRC	0x0C	0xC0	Timeout[3] + TXData(0)+ .. + TXData(N-1)		

(1) System Code for this command is defined in the Felica standard. (0xFFFF is default value)

(2) Reserved is a parameter depend on card configuration. (0x00 is default value)

(3) The timeout code is defined in the command “Set Timeout for Card response” (0x01 + 0x42)

7.3 Felica -Command Response (0x0C)

Command Name	Command Category	FBP Command	Resp	Payload (Data[0]… Data[n-1])	
C_Config_Reader_Felica	0x0C	0x00	0x01	-	
			Resp_Err	-	
C_Config_Speed_Reader	0x0C	0x01	0x01	-	
			Resp_Err	-	
C_Get_Speed_Reader	0x0C	0x02	0x01	Speed TX/RX	
			Resp_Err	-	
C_Polling	0x0C	0x10	0x01	[Slot Num + Sub Resp + Sub Len + UID(0) + UID(1) + UID(2) + … + UID(13) + UID(14) + UID(15) + System Code(0) + System Code(1)] + [Repeated data in slot containing data + + …	
				Slot_Num	Number of slot that there is tag response
				Sub Resp	Response of sub packet where it meaning is as defined in table 1.2.2 If Slot_Resp = 0x01, “Sub Len” and “UID” is reported, else remaining data in decoding FIFO in MCU is reported.
				Sub Len	Length of data in each sub slot
				UID	Unique ID
			Resp_Err	-	
C_Transparent_With_CRC	0x0C	0xC0	0x01	RespData(0) + … + RespData(N-1)	
			Resp_Err	-	