

## **RD220 Serial USB RFID Reader**

### **Demonstration Software User Manual**

Document Version 1.0

#### **Revision History**

Revision	Date	Software Version	Description/ Change / Updated / Comment
1.0	March 2013	1.0	1 <sup>st</sup> Release

#### Contents

1.	Introduction	10
2.	Getting Started	10
	2.1 System and Hardware Requirements	10
	2.2 Content in CD	10
	2.3 Software Installation	11
	2.3.1 Driver Installation (For Virtual RS232 on USB)	11
3.	Quick Start with Demonstration Software	14
4.	Demonstration Software Component	16
	4.1 Port-and-Constant Setup	16
	4.1.1 OPEN PORT	16
	4.1.2 CLOSE PORT	17
	4.1.3 SET INIT SEQNUM	17
	4.1.4 SET DEV ID	17
	4.1.5 GET FW VERSION	18
	4.2 RFID standard taps and Reader Configuration tap	18
	4.2.1 RFID standard taps	18
	4.2.1.1 "Standard Setup" Button	18
	4.2.1.1 "Send" Button	18
	4.2.1.2 "Start Send Multi Cmd" Button 4.2.1.3 Reader TX and Rx Speed selection	18
	4.2.1.4 "Enable once check" Check box	19
	4.2.1.5 Command List	19
	4.2.1.6 Parameter setup	19
	4.2.1.7 Command Response	20
	4.2.2 Reader Configuration tap	20
	4.2.2.1 Device ID Setup	20
	4.3 Raw data input and output for developer	21
	4.3.1 "Reset ASIC" button	21
	4.3.2 "ON RF" button	21
	4.3.3 "OFF RF" button	21
	4.3.4 "RF Power Level" Selection	21
	4.3.5 "Set up" button (Power Level)	21
	4.3.6 "Width Restore (<)" button	21
	4.3.7 "Send" button	22
	4.3.8 "Clear" button	22
	4.3.9 Direct TX Input and RX Output	22
	4.3.10 "Clear Logs" button	22
	4.3.11 "Show Raw Pck" Check box	22
	4.3.12 Additional TX Input and RX Output section	22
	4.3.13 "<" button	23
	4.4 Transaction Logs	23
5.	Pi-931 Protocol	24
6.	Using Demonstration Software	26
	6.1 ISO14443A	26

### RD-220 Demonstration Software Manual

6.1.1 ISO14	1443A standard commands	26
6.1.1.1	A_Request	26
6.1.1.2	A_WakeUp	27
6.1.1.3	A_Anticoll	27
6.1.1.4	A_Select	28
6.1.1.5	A_RATS (Request for answer to select)	28
6.1.1.6	A_PPS (Protocol and parameter selection request)	29
6.1.1.7	A_Deselect	30
6.1.2 Perfo	rming Basic operation in ISO14443A standard	30
6.1.3 MIFA	RE	32
6.1.3.1	A LoadKey	33
6.1.3.2	A Authentication	33
6.1.3.3	– A ReadBlock	34
6.1.3.4	A WriteBlock	35
6.1.3.5	A WriteValueBlock	35
6.1.3.6	A WriteSectorTrailerBlock	36
6.1.3.7	A Increment	38
6.1.3.8	_ A Decrement	38
6.1.3.9	A_Restore	38
6.1.3.10	A_Transfer	38
6.1.3.11	A_Req_Anti_LoadKey_Authent	39
6.1.3.12	A_Req_Anti_LoadKey_Authent_Read	40
6.1.3.13	A_Req_Anti_LoadKey_Authent_Write	41
6.1.3.14	A_Increment_Transfer	41
6.1.3.15	A_Decrement_Transfer	42
6.1.3.16	A_Restore_Transfer	43
6.1.3.17	A_Req_Anti_Select	43
6.2 ISO14443E	3	44
6.2.1 ISO14	1443B standard commands	44
6.2.1.1	B_Request	44
6.2.1.2	B_WakeUp	45
6.2.1.3	B_ATTRIB	46
6.2.1.4	B_Halt	46
6.2.1.5	B_Deselect	46
6.2.1.6	B_TransparentWithCRC	47
6.2.1.7	B_TransparentWithoutCRC	48
6.3 ISO15693.		49
6.3.1 ISO15	5693 Standard Command	52
6.3.1.1	Inventory 1slot	53
6.3.1.2	Inventory 16 slot	54
6.3.1.3	Stay Quiet	55
6.3.1.4	ReadSingleBlocks	56
6.3.1.5	WriteSingleBlocks	56
6.3.1.6	Lock Block	56
6.3.1.1	ReadMultipleBlocks	56
6.3.1.2	WriteMultipleBlocks	57
6.3.1.3	Select	58
6.3.1.4	ResetToReady	58
6.3.1.5	WriteAFI	59
6.3.1.6	LockAFI	59
6.3.1.7	WriteDSFID	59
6.3.1.8	Lock DSFID	60
6.3.1.9	Get System Information	60
6.3.1.10	Get Multiple Block Security status	60

# RD-220 Demonstration Software Manual

6.3.2 Exam	nple of ISO15693 standard command usage	61
6.3.3 Custo	om command for SIC5600	62
6.3.3.1	Set EAS	63
6.3.3.2	Reset EAS	63
6.3.3.3	Lock EAS	63
6.3.3.4	EAS Alarm	64
6.3.3.5	Write Password	64
6.3.3.6	Lock Password	64
6.3.3.7	Set Password Mode	64
6.3.3.8	Lock Password Mode	66
6.3.3.9	Get Password Mode	66
6.3.3.10	Load Password	67
6.3.3.11	Kill	69
6.3.3.12	Set OTP	70
6.3.3.13	Get OTP	70
6.3.3.14	Write XUID	71
6.3.3.15	Read XUID	71
6.3.4 Spec	ial command	72
6.4 Pico Tag		73
6.4.1 Pico	Tag standard commands	73
6.4.1.1	ACTALL	74
6.4.1.2	IDENTIFY	74
6.4.1.3	SELECT ASNB	75
6.4.1.4	SELECT SNB	75
6.4.1.5	HALT	75
6.4.1.1	READ	75
6.4.1.1	READ4	76
6.4.2 Spec	ial command	77
6.5 Felica		78
6.5.1 Pollir	ng command	78
6.5.2 Trans	sparentWithCRC	79

# RD-220 Demonstration Software Manual

### List of Figures

Figure 2 A "Found New Hardware" pops up.11Figure 3 Select "No, not this time".11Figure 4 Select Option "Install from a list or specific location (Advanced)".11Figure 5 Browse to folder "Pi931_Driver" in CD or where the driver is located12Figure 6 Click "Continue Anyway".12Figure 7 Windows is installing the driver.12Figure 8 Droup shows hardware is ready to use13Figure 9 Dop up shows hardware is ready to use13Figure 10 COM Port for Pi-931 Hardware displays in Device Manager.14Figure 12 Demonstration software14Figure 13 TX and RX Speed for reader CODEC in ISO15693 protocol15Figure 14 Sections in Demonstration software16Figure 15 Port-and-Constant Setup16Figure 16 OPEN PORT Menu17Figure 19 SET INIT SEQNUM Menu17Figure 20 SET DEV ID Menu17Figure 20 SET DEV ID Menu18Figure 22 Firmware version report.18Figure 23 Example of button and setup in ISO 15693 tap.19Figure 24 Closs PORT Menu17Figure 25 Example of button and setup in ISO 15693 tap.19Figure 25 Example of Dicti N Jonut and RX Output22Figure 26 Command and setup in Reader Configuration tap.20Figure 27 Raw data input and output for developer.21Figure 28 Command Harme Format24Figure 29 Effect of "Show Raw Pck" in transaction display.22Figure 28 Command Frame Format24Figure 29 Effect of "Show Raw Pck" in transaction displayed in demon	Figure 1 Content in CD	
Figure 3 Select "No, not this time".11Figure 5 Servox to folder "IPISI_Driver" in CD or where the driver is located11Figure 5 Rows to folder "IPISI_Driver" in CD or where the driver is located12Figure 6 Click "Continue Anyway".12Figure 7 Windows is installing the driver.12Figure 8 Installation completes.13Figure 10 COM Port for PI-931 Hardware displays in Device Manager.13Figure 10 COM Port for PI-931 Hardware displays in Device Manager.13Figure 11 Demonstration software14Figure 12 Demonstration software14Figure 13 TX and RX Speed for reader CODEC in ISO15693 protocol15Figure 14 Sections in Demonstration software.16Figure 15 OPEN PORT Menu.16Figure 16 OPEN PORT Menu.17Figure 17 connected com port and communication speed (kbps)17Figure 21 Constrated Sectors report.18Figure 22 SET DEV ID Menu17Figure 23 CSE PORT Menu.17Figure 24 Cast Saved UID" checked box.20Figure 25 Command and setup in ISO 15693 tap.20Figure 25 Command and setup in ISO 15693 tap.20Figure 25 Command and setup in Reader Configuration tap.22Figure 25 Command and setup in Reader Configuration tap.20Figure 25 Command and setup in Reader Configuration tap.20Figure 25 Command and setup in Reader Configuration tap.22Figure 26 Command and setup in Transaction display.22Figure 27 Raw data input and output for developer.21F	Figure 2 A "Found New Hardware" pops up	11
Figure 4 Select Option "Install from a list or specific location (Advanced)"11Figure 5 Browse to folder "Pi931_Driver" in CD or where the driver is located12Figure 6 Click "Continue Anyway"12Figure 7 Windows is installing the driver.12Figure 8 Installation completes.13Figure 9 Dop up shows hardware is ready to use13Figure 11 Demonstration software14Figure 12 Demonstration software14Figure 13 Demonstration software16Figure 14 Sections in Demonstration software16Figure 15 Port-and-Constant Setup16Figure 15 Port-and-Constant Setup16Figure 16 OPEN PORT Menu.17Figure 21 GET FW VERSION Menu17Figure 22 Firmware version report.18Figure 23 Example of button and setup in ISO 15693 tap.19Figure 24 Click Saved UID" checked box.20Figure 25 Errowards and setup in ISO 15693 tap.19Figure 25 Example of button and setup in ISO 15693 tap.19Figure 24 Command and setup in Reader Configuration tap.20Figure 25 Erroward as and UID" checked box.20Figure 25 Command and setup in Reader Configuration tap.20Figure 25 Command and setup in Reader Configuration tap.22Figure 25 Command and setup in Reader Configuration tap.22Figure 26 Command and setup in Reader Configuration tap.22Figure 27 Command and setup in Reader Configuration tap.22Figure 28 Example of Direct TX Input and RX output section.23Figure 28 Comm	Figure 3 Select "No, not this time"	11
Figure 5Browse to folder "Pi931_Driver" in CD or where the driver is located12Figure 6Click "Continue Anyway"12Figure 7Windows is installing the driver.12Figure 8Installation completes.13Figure 9pop up shows hardware is ready to use13Figure 10COM Port for Pi-931 Hardware displays in Device Manager.13Figure 12Demonstration software14Figure 13TX and RX Speed for reader CODEC in ISO15693 protocol15Figure 14Sections in Demonstration software.16Figure 15Ort And Constant Setup16Figure 15Port-and-Constant Setup16Figure 17conceted com port and communication speed (kbps)17Figure 15SET INIT SEQNUM Menu17Figure 20SET DEV ID Menu17Figure 21GET FW VERSION Menu18Figure 22Figure 23ET for Version report.18Figure 24'Load Saved UID" checked box.20Figure 25'Save UID" checked box.20Figure 24'Load Saved UID" checked box.20Figure 25'Save UID" checked box.20Figure 26'Save UID" checked box.20Figure 27Raw data input and output for developer.21Figure 28Example of Direct TX Input and RX output section23Figure 29Effect of 'Show Raw Pck' in transaction display.22Figure 21Command Frame Format24Figure 23Command Frame Format </td <td>Figure 4 Select Option "Install from a list or specific location (Advanced)"</td> <td>11</td>	Figure 4 Select Option "Install from a list or specific location (Advanced)"	11
Figure 6Click "Continue Anyway"12Figure 7Windows is installing the driver.12Figure 8Installation completes.13Figure 10COM Port for PI-931 Hardware displays in Device Manager.13Figure 11Demonstration software14Figure 12Demonstration software14Figure 13TX and RX Speed for reader CODEC in ISO15693 protocol15Figure 14Sections in Demonstration software16Figure 15Port and-Constant Setup.16Figure 16OPEN PORT Menu17Figure 17connected com port and communication speed (kbps)17Figure 18COSE PORT Menu17Figure 21GET FW VERSION Menu17Figure 22Figure 22Firm Vision and setup in ISO 15693 tap19Figure 23GET FW VERSION Menu17Figure 24Cload setup in ISO 15693 tap19Figure 25"Save UID" checked box.20Figure 24Cload setup in ISO 15693 tap19Figure 25"Save UID" checked box.20Figure 25"Save UID" checked box.20Figure 26Command and setup in ISO 15693 tap20Figure 27Save UID" checked box.20Figure 28Example of Ditron and setup in ISO 15693 tap20Figure 29Effect of "Now Naw Pck" in transaction tap.20Figure 29Effect of "Now Naw Pck" in transaction tapslay.22Figure 20Sommand and setup in Reader Configuration tap.20 </td <td>Figure 5 Browse to folder "Pi931_Driver" in CD or where the driver is located</td> <td>12</td>	Figure 5 Browse to folder "Pi931_Driver" in CD or where the driver is located	12
Figure 7Windows is installing the driver.12Figure 8Installation completes.13Figure 9pop up shows hardware is ready to use.13Figure 10COM Port for PI-931 Hardware displays in Device Manager.14Figure 11Demonstration software.14Figure 12Demonstration software.14Figure 13X and RX Speed for reader CODEC in ISO15693 protocol.15Figure 14Sections in Demonstration software.16Figure 15Port-and-Constant Setup.16Figure 16OPEN PORT Menu.17Figure 16OPEN PORT Menu.17Figure 17connected com port and communication speed (kbps)17Figure 20SET DEV ID Menu.17Figure 21SET INIT SEQNUM Menu.17Figure 22Firmware version report.18Figure 23Example of button and setup in ISO 15693 tap.19Figure 23Example of button and setup in ISO 15693 tap.20Figure 24'Load Saved UID' checked box.20Figure 25"Saw UID' checked box.20Figure 26Command and setup in Reader Configuration tap.22Figure 27Raw data input and 0 utput for developer.21Figure 28Example of Direct TX input and RX Output section23Figure 30Usage of additional TX input and RX Output section23Figure 31Transaction display.22Figure 32Command Frame Format24Figure 34Decomposition of raw frame	Figure 6 Click "Continue Anyway"	12
Figure 8 Installation completes.13Figure 9 pop up shows hardware is ready to use13Figure 10 COM Port for Pi-931 Hardware displays in Device Manager.13Figure 11 Demonstration software14Figure 12 Demonstration software14Figure 13 TX and RX Speed for reader CODEC in ISO15693 protocol15Figure 14 Sections in Demonstration software.16Figure 15 Port-and-Constant Setup.16Figure 16 OPEN PORT Menu17Figure 17 connected com port and communication speed (kbps)17Figure 18 CLOSE PORT Menu17Figure 21 GET FW VERSION Menu17Figure 22 Firmware version report.18Figure 22 Firmware version report.18Figure 23 Example of button and setup in ISO 15693 tap.19Figure 24 Command and setup in ISO 15693 tap.19Figure 25 "Save UID" checked box.20Figure 25 "Save UID" checked box.20Figure 26 Command and setup in Reader Configuration tap.20Figure 27 Raw data input and output for developer.21Figure 28 USage of additional TX input and RX output section23Figure 31 Transaction logs text box.23Figure 33 Command Frame Format24Figure 34 Decomposition of an frame format of PI-93125Figure 34 Decomposition of and frame format of PI-93125Figure 34 Nataeder Command.26Figure 34 Decomposition of Antrane format of PI-93127Figure 34 Decomposition of Antrane format of PI-93127Figure 35 Did443A standard comm	Figure 7 Windows is installing the driver	12
Figure 9 pop up shows hardware is ready to use       13         Figure 10 COM Port for PI-931 Hardware displays in Device Manager.       13         Figure 11 Demonstration software       14         Figure 12 Demonstration software       14         Figure 13 TX and RX Speed for reader CODEC in ISO15693 protocol       15         Figure 14 Sections in Demonstration software.       16         Figure 15 Port-and-Constant Setup       16         Figure 16 OPEN PORT Menu       16         Figure 17 connected com port and communication speed (kbps)       17         Figure 18 CLOSE PORT Menu       17         Figure 21 GET FW VERSION Menu       17         Figure 22 Firmware version report.       18         Figure 23 Example of button and setup in ISO 15693 tap.       19         Figure 24 Command and setup in ISO 15693 tap.       19         Figure 25 "Save UID" checked box.       20         Figure 26 Command and setup in Reader Configuration tap.       20         Figure 27 Rw data input and output for developer.       21         Figure 28 ISample of Direct TX Input and RX Output       22         Figure 31 Command rame Format       24         Figure 32 Command net Format       24         Figure 32 Command Frame Format       24         Figure 33 Composition of raw frame form	Figure 8 Installation completes.	13
Figure 10COM Port for Pi-931 Hardware displays in Device Manager.13Figure 11Demonstration software14Figure 12Demonstration software14Figure 13TX and RX Speed for reader CODEC in ISO15693 protocol15Figure 14Sections in Demonstration software16Figure 15Port-and-Constant Setup16Figure 16OPEN PORT Menu17Figure 17connected com port and communication speed (kbps)17Figure 19SET INT SEQNUM Menu17Figure 20SET DEV ID Menu17Figure 21GET DEV IO Menu17Figure 23IST DEV SION Menu18Figure 24Cload Saved UID" checked box20Figure 25Save UID" checked box20Figure 25Command and setup in ISO 15693 tap19Figure 25Command and setup in ISO 15693 tap20Figure 25Save UID" checked box20Figure 25Command and setup in Reader Configuration tap20Figure 25Save UID" checked box20Figure 26Command and setup in Reader Configuration tap22Figure 27Raw data input and output for developer.21Figure 29Effect of "Show Raw Pck" in transaction display.22Figure 31Command Erame Format24Figure 32Command Frame Format24Figure 33Command Frame Format of Pi-93125Figure 34Decomposition of raw frame format of Pi-93125Figure 35D	Figure 9 pop up shows hardware is ready to use	13
Figure 11Demonstration software14Figure 12Demonstration software14Figure 13TX and RX Speed for reader CODEC in ISO15693 protocol15Figure 14Sections in Demonstration software.16Figure 15Port-and-Constant Setup16Figure 16OPEN PORT Menu16Figure 17connected com port and communication speed (kbps)17Figure 18CLOSE PORT Menu17Figure 20SET DEV ID Menu17Figure 21GET FW VERSION Menu17Figure 22FIT WIT SEQNUM Menu17Figure 23Figure 24Stample of button and setup in ISO 15693 tap18Figure 24*Load Saved UID" checked box20Figure 25"Save UID" checked box20Figure 25"Save UID" checked box20Figure 25"Save UID" checked box20Figure 26Command and setup in ISO 15693 tap20Figure 27Raw data input and output for developer.21Figure 28Kample of button and setup in Roader Configuration tap22Figure 29Effect of "Show Raw Pck" in transaction display.22Figure 30Usage of additional TX input and RX output section23Figure 31Transaction Logs text box.23Figure 32Command Frame Format24Figure 33Command Frame Format24Figure 34Decomposition of abridged frame format of PI-93125Figure 35Decomposition of abridged frame format of PI-931 <t< td=""><td>Figure 10 COM Port for Pi-931 Hardware displays in Device Manager.</td><td>13</td></t<>	Figure 10 COM Port for Pi-931 Hardware displays in Device Manager.	13
Figure 12Demonstration software14Figure 13TX and RX Speed for reader CODEC in ISO15693 protocol15Figure 14Sections in Demonstration software16Figure 15Port-and-Constant Setup16Figure 16OPEN PORT Menu16Figure 17connected com port and communication speed (kbps)17Figure 18CLOSE PORT Menu17Figure 19SET INIT SEQNUM Menu17Figure 21SET DEV ID Menu17Figure 21GET FW VERSION Menu18Figure 22Firmware version report.18Figure 23Example of button and setup in ISO 15693 tap19Figure 24Cond Saved UID" checked box20Figure 25Command and setup in Reader Configuration tap20Figure 26Command and setup in Reader Configuration tap20Figure 27Raw data input and output for developer.21Figure 28Example of Direct TX Input and RX Output22Figure 30Usage of additional TX input and RX output section23Figure 31Transaction Logs text box23Figure 32Decomposition of raw frame format displayed in demonstration software25Figure 33Sontamad frame format displayed in demonstration software25Figure 34Decomposition of aw frame format displayed in demonstration software26Figure 35Decomposition of aw frame format displayed in demonstration software26Figure 34Decomposition of aw frame format displayed in demonstration softw	Figure 11 Demonstration software	14
Figure 13TX and RX Speed for reader CODEC in ISO15693 protocol15Figure 14Sections in Demonstration software16Figure 15Port-and-Constant Setup16Figure 15PORT Menu16Figure 17connected com port and communication speed (kbps)17Figure 17SET INIT SEQNUM Menu17Figure 20SET DEV ID Menu17Figure 21GET FW VERSION Menu17Figure 22FIT WARSION Menu18Figure 23EXT PV VERSION Menu18Figure 24'Load Saved UID'' checked box20Figure 25Save UID'' checked box20Figure 26Command and setup in ISO 15693 tap20Figure 27Raw data input and output for developer21Figure 28Example of Direct TX Input and RX Output22Figure 29Effect of 'Show Raw Pck'' in transaction display22Figure 31Transaction Logs text box23Figure 32Command Frame Format24Figure 33Command Frame Format24Figure 34Decomposition of aw frame format of PI-93125Figure 35Decomposition of aw frame format displayed in demonstration software25Figure 34Decomposition of aw frame format displayed in demonstration software26Figure 37A_Requet Command26Figure 34Decomposition of aw frame format displayed in demonstration software25Figure 34Decomposition of aw frame format displayed in demonstration software25 <td>Figure 12 Demonstration software</td> <td>14</td>	Figure 12 Demonstration software	14
Figure 14Sections in Demonstration software16Figure 15Port-and-Constant Setup.16Figure 16OPEN PORT Menu16Figure 17connected com port and communication speed (kbps)17Figure 18CLOSE PORT Menu17Figure 19SET INIT SEQNUM Menu17Figure 20SET DEV ID Menu17Figure 21GET FW VERSION Menu18Figure 22Firmware version report.18Figure 23Kample of button and setup in ISO 15693 tap.19Figure 24"Load Saved UID" checked box.20Figure 25"Save UID" checked box.20Figure 26Command and setup in Reader Configuration tap.20Figure 27Rw data input and output for developer.21Figure 28Example of Direct TX Input and RX Output22Figure 30Usage of additional TX input and RX output section23Figure 31Transaction Logs text box.23Figure 32Command Frame Format24Figure 33Command Frame Format24Figure 34Decomposition of aw frame format displayed in demonstration software25Figure 35Decomposition of abridged frame format displayed in demonstration software26Figure 37A_Requet Command27Figure 38ISO14443A standard commands available in Pi-93127Figure 39A_Wakeup Command.27Figure 44Parameters in A_ARTS command28Figure 44Parameters in A_ARTS command <td< td=""><td>Figure 13 TX and RX Speed for reader CODEC in ISO15693 protocol</td><td>15</td></td<>	Figure 13 TX and RX Speed for reader CODEC in ISO15693 protocol	15
Figure 15Port-and-Constant Setup16Figure 16OPEN PORT Menu16Figure 17connected com port and communication speed (kbps)17Figure 18CLOSE PORT Menu17Figure 19SET INIT SEQNUM Menu17Figure 20SET DEV ID Menu17Figure 21GET FW VERSION Menu18Figure 22Firmware version report.18Figure 23Example of button and setup in ISO 15693 tap19Figure 24"Load Saved UID" checked box20Figure 25Command and setup in ISO 15693 tap20Figure 26Command and setup in Reader Configuration tap20Figure 27Raw data input and output for developer21Figure 28Example of Direct TX Input and RX Output22Figure 30Usage of additional TX input and RX output section23Figure 31Transaction Logs text box.23Figure 32Command Frame Format24Figure 33Command Frame Format displayed in demonstration software25Figure 34ISO14443A command list26Figure 35ISO14443A command27Figure 40Parameters in A_AntiColl command27Figure 41Result ID from CollMaskVal.28Figure 42Parameters in A_AFATS command29Figure 43Parameters in A_PSE command29Figure 44Parameters in A_PSE command29Figure 44Parameters in A_PSE command29Figure 44Parameters in A_PSE c	Figure 14 Sections in Demonstration software	16
Figure 16OPEN PORT Menu16Figure 17connected com port and communication speed (kbps)17Figure 18CLOSE PORT Menu17Figure 19SET INIT SEQNUM Menu17Figure 20SET DEV ID Menu17Figure 21GET FW VERSION Menu18Figure 22FIFW VERSION Menu18Figure 23Example of button and setup in ISO 15693 tap.19Figure 24"Load Saved UID" checked box.20Figure 25"Save UID" checked box.20Figure 26Command and setup in Reader Configuration tap.20Figure 27Raw data input and output for developer.21Figure 28Example of Direct TX Input and RX Output22Figure 30Usage of additional TX input and RX output section23Figure 31Transaction Logs text box.23Figure 32Command Frame Format24Figure 33Command Frame Format displayed in demonstration software25Figure 34ISO14443A command list26Figure 35ISO14443A command26Figure 34ISO14443A command27Figure 40Parameters in A_AntiColl command28Figure 41Result ID from CollMaskVal.28Figure 42Parameters in A_ARATS command29Figure 44Parameters in A_POPS command29Figure 44Parameters in A_POPS command29Figure 44Parameters in A_POPS command29Figure 44Parameters in A_RATS command2	Figure 15 Port-and-Constant Setup	16
Figure 17 connected com port and communication speed (kbps)17Figure 18 CLOSE PORT Menu17Figure 19 SET INIT SEQNUM Menu17Figure 20 SET DEV ID Menu17Figure 21 GET FW VERSION Menu18Figure 22 Firmware version report.18Figure 23 Example of button and setup in ISO 15693 tap.19Figure 24 "Load Saved UID" checked box.20Figure 25 "Save UID" checked box.20Figure 26 Command and setup in Reader Configuration tap.20Figure 27 Raw data input and output for developer.21Figure 28 Example of Direct TX Input and RX Output22Figure 30 Usage of additional TX input and RX Output section23Figure 31 Transaction Logs text box.23Figure 32 Command Frame Format24Figure 33 Command Frame Format24Figure 34 Decomposition of raw frame format of PI-93125Figure 35 Decomposition of raw frame format of PI-93125Figure 38 ISO14443A command list26Figure 39 A_Wakeup Command27Figure 40 Parameters in A_AntiColl command28Figure 41 Result 1D from CollMaskVal28Figure 42 Parameters in A_PPS command29Figure 44 Parameters in A_PPS command29Figure 45 A_Deselect command29Figure 44 Parameters in A_PPS command29 <td>Figure 16 OPEN PORT Menu</td> <td>16</td>	Figure 16 OPEN PORT Menu	16
Figure 18CLOSE PORT Menu17Figure 19SET INIT SEQNUM Menu17Figure 20SET DEV ID Menu17Figure 21GET FW VERSION Menu18Figure 22Firmware version report.18Figure 23Example of button and setup in ISO 15693 tap.19Figure 24"Load Saved UID" checked box.20Figure 25Goswed UID" checked box.20Figure 26Command and setup in Reader Configuration tap.20Figure 27Raw data input and output for developer.21Figure 28Example of Direct TX Input and RX Output.22Figure 30Usage of additional TX input and RX output section23Figure 31Transaction Logs text box.23Figure 32Command Frame Format24Figure 33Command Frame Format of Pi-93125Figure 34Decomposition of raw frame format displayed in demonstration software25Figure 35Decomposition of abridged frame format displayed in demonstration software26Figure 34Nameters in A_AntiColl command27Figure 40Parameters in A_AntiColl command28Figure 41Result 10 from CollMaskVal28Figure 42Parameters in A_PPS command29Figure 44Parameters in A_PPS command	Figure 17 connected com port and communication speed (kbps)	17
Figure 19SET INIT SEQNUM Menu17Figure 20SET DEV ID Menu17Figure 21GET FW VERSION Menu18Figure 22Firmware version report18Figure 23Example of button and setup in ISO 15693 tap19Figure 24"Load Saved UID" checked box20Figure 25"Save UID" checked box20Figure 26Command and setup in Reader Configuration tap20Figure 27Raw data input and output for developer21Figure 28Example of Direct TX Input and RX Output22Figure 29Effect of "Show Raw Pck" in transaction display22Figure 30Usage of additional TX input and RX output section23Figure 31Transaction Logs text box23Figure 32Command Frame Format24Figure 33Decomposition of raw frame format displayed in demonstration software25Figure 34Decomposition of abridged frame format displayed in demonstration software26Figure 35Decomposition of abridged frame format displayed in demonstration software26Figure 34Nakeup Command26Figure 35A_Request Command27Figure 40Parameters in A_AntiColl command28Figure 41Result 1D from CollMaskVal28Figure 42Parameters in A_Select command29Figure 44Parameters in A_PPS command29Figure 45A_Deselect command29Figure 44Parameters in A_PPS command29Figure 45<	Figure 18 CLOSE PORT Menu	
Figure 20SET DEV ID Menu17Figure 21GET FW VERSION Menu18Figure 22Firmware version report18Figure 23Example of button and setup in ISO 15693 tap19Figure 24"Load Saved UID" checked box20Figure 25"Save UID" checked box20Figure 26Command and setup in Reader Configuration tap20Figure 27Raw data input and output for developer21Figure 28Example of Direct TX Input and RX Output22Figure 29Effect of "Show Raw Pck" in transaction display22Figure 30Usage of additional TX input and RX output section23Figure 31Transaction Logs text box23Figure 32Command Frame Format24Figure 33Command Frame Format24Figure 34Decomposition of abridged frame format of Pi-93125Figure 35Decomposition of abridged frame format displayed in demonstration software26Figure 38ISO14443A command27Figure 39A_Wakeup Command27Figure 40Parameters in A_AntiColl command28Figure 41Result ID from CollMaskVal.28Figure 44Parameters in A_PPS command29Figure 44Parameters in A_PS command29Figure 45A_Deselect command29Figure 44Parameters in A_PPS command29Figure 45A_Deselect command29Figure 45A_Deselect command29Figure 45A_Des	Figure 19 SET INIT SEQNUM Menu	17
Figure 21GET FW VERSION Menu18Figure 22Firmware version report18Figure 23Example of button and setup in ISO 15693 tap.19Figure 24"Load Saved UID" checked box.20Figure 25"Save UID" checked box.20Figure 26Command and setup in Reader Configuration tap.20Figure 27Raw data input and output for developer.21Figure 28Example of Direct TX Input and RX Output22Figure 29Effect of "Show Raw Pck" in transaction display.22Figure 30Usage of additional TX input and RX output section23Figure 31Transaction Logs text box.23Figure 32Command Frame Format24Figure 33Command Frame Format24Figure 34Decomposition of raw frame format of PI-93125Figure 35Decomposition of abridged frame format displayed in demonstration software25Figure 38ISO14443A command list26Figure 39A_Wakeup Command27Figure 40Parameters in A_AntiColl command28Figure 41Result ID from CollMaskVal28Figure 42Parameters in A_Select command29Figure 45A_Deselect	Figure 20 SET DEV ID Menu	17
Figure 22Firmware version report	Figure 21 GET FW VERSION Menu	
Figure 23Example of button and setup in ISO 15693 tap.19Figure 24"Load Saved UID" checked box.20Figure 25"Save UID" checked box.20Figure 26Command and setup in Reader Configuration tap.20Figure 27Raw data input and output for developer.21Figure 28Example of Direct TX Input and RX Output22Figure 29Effect of "Show Raw Pck" in transaction display.22Figure 30Usage of additional TX input and RX output section23Figure 31Transaction Logs text box.23Figure 32Command Frame Format24Figure 33Command Frame Format24Figure 34Decomposition of raw frame format of PI-93125Figure 35Decomposition of abridged frame format displayed in demonstration software25Figure 38ISO14443A command list26Figure 39A_Wakeup Command27Figure 40Parameters in A_AntiColl command28Figure 41Result ID from CollMaskVal28Figure 42Parameters in A_ATS command29Figure 44Parameters in A_PEs command29Figure 45A_Deselect command29Figure 44Parameters in A_PEs command29Figure 45A_Deselect command29Figure 45A_Deselect command29Figure 45A_Deselect command29Figure 45A_Deselect command29Figure 45A_Deselect command29	Figure 22 Firmware version report	
Figure 24 "Load Saved UID" checked box.20Figure 25 "Save UID" checked box.20Figure 26 Command and setup in Reader Configuration tap.20Figure 27 Raw data input and output for developer.21Figure 28 Example of Direct TX Input and RX Output22Figure 29 Effect of "Show Raw Pck" in transaction display.22Figure 30 Usage of additional TX input and RX output section23Figure 31 Transaction Logs text box.23Figure 32 Command Frame Format24Figure 33 Command Frame Format24Figure 34 Decomposition of raw frame format of Pi-93125Figure 35 Decomposition of abridged frame format displayed in demonstration software25Figure 38 ISO14443A command list26Figure 39 A_Wakeup Command27Figure 40 Parameters in A_AntiColl command28Figure 41 Result ID from CollMaskVal.28Figure 42 Parameters in A_PPS command29Figure 44 Parameters in A_PPS command29Figure 45 A_Deselect command<	Figure 23 Example of button and setup in ISO 15693 tap	19
Figure 25 "Save UID" checked box20Figure 26 Command and setup in Reader Configuration tap20Figure 27 Raw data input and output for developer21Figure 28 Example of Direct TX Input and RX Output22Figure 29 Effect of "Show Raw Pck" in transaction display22Figure 30 Usage of additional TX input and RX output section23Figure 31 Transaction Logs text box23Figure 32 Command Frame Format24Figure 33 Command Frame Format24Figure 34 Decomposition of raw frame format of Pi-93125Figure 35 Decomposition of abridged frame format displayed in demonstration software25Figure 38 ISO14443A command list26Figure 39 A_Wakeup Command27Figure 40 Parameters in A_AntiColl command28Figure 41 Result ID from CollMaskVal.28Figure 42 Parameters in A_PPS command29Figure 44 Parameters in A_PPS command29Figure 45 A_Deselect command29	Figure 24 "Load Saved UID" checked box	20
Figure 26Command and setup in Reader Configuration tap20Figure 27Raw data input and output for developer.21Figure 28Example of Direct TX Input and RX Output22Figure 29Effect of "Show Raw Pck" in transaction display22Figure 30Usage of additional TX input and RX output section23Figure 31Transaction Logs text box.23Figure 32Command Frame Format24Figure 33Command Frame Format24Figure 34Decomposition of raw frame format of Pi-93125Figure 35Decomposition of abridged frame format displayed in demonstration software25Figure 37A_Request Command26Figure 38ISO14443A standard commands available in Pi-93127Figure 39A_Wakeup Command27Figure 40Parameters in A_AntiColl command28Figure 41Result ID from CollMaskVal28Figure 42Parameters in A_Select command29Figure 44Parameters in A_PPS command29Figure 45A_Deselect command29Figure 45A_Deselect command29	Figure 25 "Save UID" checked box	20
Figure 27Raw data input and output for developer.21Figure 28Example of Direct TX Input and RX Output22Figure 29Effect of "Show Raw Pck" in transaction display22Figure 30Usage of additional TX input and RX output section23Figure 31Transaction Logs text box.23Figure 32Command Frame Format24Figure 33Command Frame Format24Figure 34Decomposition of raw frame format of Pi-93125Figure 35Decomposition of abridged frame format displayed in demonstration software25Figure 36ISO14443A command list26Figure 37A_Request Command26Figure 38ISO14443A standard commands available in Pi-93127Figure 40Parameters in A_AntiColl command28Figure 41Result ID from CollMaskVal28Figure 42Parameters in A_Select command29Figure 44Parameters in A_PPS command29Figure 45A_Deselect command30	Figure 26 Command and setup in Reader Configuration tap	20
Figure 28Example of Direct TX Input and RX Output22Figure 29Effect of "Show Raw Pck" in transaction display22Figure 30Usage of additional TX input and RX output section23Figure 31Transaction Logs text box.23Figure 32Command Frame Format24Figure 33Command Frame Format24Figure 34Decomposition of raw frame format of Pi-93125Figure 35Decomposition of abridged frame format displayed in demonstration software25Figure 36ISO14443A command list26Figure 37A_Request Command27Figure 38ISO14443A standard commands available in Pi-93127Figure 40Parameters in A_AntiColl command28Figure 41Result ID from CollMaskVal28Figure 42Parameters in A_Select command29Figure 44Parameters in A_PPS command29Figure 45A_Deselect command29Figure 45A_Deselect command30	Figure 27 Raw data input and output for developer	21
Figure 29Effect of "Show Raw Pck" in transaction display22Figure 30Usage of additional TX input and RX output section23Figure 31Transaction Logs text box.23Figure 32Command Frame Format24Figure 33Command Frame Format24Figure 34Decomposition of raw frame format of Pi-93125Figure 35Decomposition of abridged frame format displayed in demonstration software25Figure 36ISO14443A command list26Figure 37 <b>A_Request</b> Command26Figure 38ISO14443A standard commands available in Pi-93127Figure 39 <b>A_Wakeup</b> Command27Figure 40Parameters in <b>A_AntiColl</b> command28Figure 41Result ID from <b>CollMaskVal</b> 28Figure 43Parameters in <b>A_ARTS</b> command29Figure 44Parameters in <b>A_PPS</b> command29Figure 45 <b>A_Deselect</b> command29Figure 45 <b>A_Deselect</b> command29	Figure 28 Example of Direct TX Input and RX Output	22
Figure 30Usage of additional TX input and RX output section23Figure 31Transaction Logs text box.23Figure 32Command Frame Format24Figure 33Commond Frame Format24Figure 34Decomposition of raw frame format of Pi-93125Figure 35Decomposition of abridged frame format displayed in demonstration software25Figure 36ISO14443A command list26Figure 37A_Request Command26Figure 38ISO14443A standard commands available in Pi-93127Figure 40Parameters in A_AntiColl command28Figure 41Result ID from CollMaskVal28Figure 42Parameters in A_Select command29Figure 43Parameters in A_PPS command29Figure 44Parameters in A_PPS command29Figure 45A_Deselect command29	Figure 29 Effect of "Show Raw Pck" in transaction display	22
Figure 31Transaction Logs text box.23Figure 32Command Frame Format24Figure 33Command Frame Format24Figure 34Decomposition of raw frame format of Pi-93125Figure 35Decomposition of abridged frame format displayed in demonstration software25Figure 36ISO14443A command list26Figure 37A_Request Command26Figure 38ISO14443A standard commands available in Pi-93127Figure 39A_Wakeup Command27Figure 40Parameters in A_AntiColl command28Figure 41Result ID from CollMaskVal.28Figure 42Parameters in A_Select command29Figure 43Parameters in A_PPS command29Figure 45A_Deselect command30	Figure 30 Usage of additional TX input and RX output section	23
Figure 32Command Frame Format24Figure 33Command Frame Format24Figure 34Decomposition of raw frame format of Pi-93125Figure 35Decomposition of abridged frame format displayed in demonstration software25Figure 36ISO14443A command list26Figure 37A_Request Command26Figure 38ISO14443A standard commands available in Pi-93127Figure 39A_Wakeup Command27Figure 40Parameters in A_AntiColl command28Figure 41Result ID from CollMaskVal28Figure 42Parameters in A_Select command29Figure 43Parameters in A_PPS command29Figure 45A_Deselect command29Figure 45A_Deselect command30	Figure 31 Transaction Logs text box.	23
Figure 33Command Frame Format24Figure 34Decomposition of raw frame format of Pi-93125Figure 35Decomposition of abridged frame format displayed in demonstration software25Figure 36ISO14443A command list26Figure 37A_Request Command26Figure 38ISO14443A standard commands available in Pi-93127Figure 39A_Wakeup Command27Figure 40Parameters in A_AntiColl command28Figure 41Result ID from CollMaskVal28Figure 42Parameters in A_Select command29Figure 43Parameters in A_RATS command29Figure 44Parameters in A_PPS command29Figure 45A_Deselect command29	Figure 32 Command Frame Format	24
Figure 34Decomposition of raw frame format of Pi-93125Figure 35Decomposition of abridged frame format displayed in demonstration software25Figure 36ISO14443A command list26Figure 37A_Request Command26Figure 38ISO14443A standard commands available in Pi-93127Figure 39A_Wakeup Command27Figure 40Parameters in A_AntiColl command28Figure 41Result ID from CollMaskVal28Figure 42Parameters in A_Select command29Figure 43Parameters in A_PPS command29Figure 44Parameters in A_PPS command29Figure 45A_Deselect command30	Figure 33 Command Frame Format	24
Figure 35Decomposition of abridged frame format displayed in demonstration software25Figure 36ISO14443A command list26Figure 37A_Request Command26Figure 38ISO14443A standard commands available in Pi-93127Figure 39A_Wakeup Command27Figure 40Parameters in A_AntiColl command28Figure 41Result ID from CollMaskVal28Figure 42Parameters in A_Select command29Figure 43Parameters in A_RATS command29Figure 44Parameters in A_PPS command29Figure 45A_Deselect command30	Figure 34 Decomposition of raw frame format of Pi-931	25
Figure 36ISO14443A command list26Figure 37 <b>A_Request</b> Command26Figure 38ISO14443A standard commands available in Pi-93127Figure 39 <b>A_Wakeup</b> Command27Figure 40Parameters in <b>A_AntiColl</b> command28Figure 41Result ID from <b>CollMaskVal</b> 28Figure 42Parameters in <b>A_Select</b> command29Figure 43Parameters in <b>A_RATS</b> command29Figure 44Parameters in <b>A_PPS</b> command29Figure 45 <b>A_Deselect</b> command30	Figure 35 Decomposition of abridged frame format displayed in demonstration software	25
Figure 37A_Request Command26Figure 38ISO14443A standard commands available in Pi-93127Figure 39A_Wakeup Command27Figure 40Parameters in A_AntiColl command28Figure 41Result ID from CollMaskVal28Figure 42Parameters in A_Select command29Figure 43Parameters in A_RATS command29Figure 44Parameters in A_PPS command29Figure 45A_Deselect command30	Figure 36 ISO14443A command list	26
Figure 38ISO14443A standard commands available in Pi-93127Figure 39A_Wakeup Command27Figure 40Parameters in A_AntiColl command28Figure 41Result ID from CollMaskVal28Figure 42Parameters in A_Select command29Figure 43Parameters in A_RATS command29Figure 44Parameters in A_PPS command29Figure 45A_Deselect command30	Figure 37 A_Request Command	26
Figure 39A_Wakeup Command27Figure 40Parameters in A_AntiColl command28Figure 41Result ID from CollMaskVal28Figure 42Parameters in A_Select command29Figure 43Parameters in A_RATS command29Figure 44Parameters in A_PPS command29Figure 45A_Deselect command29Figure 45A_Deselect command30	Figure 38 ISO14443A standard commands available in Pi-931	27
Figure 40Parameters in A_AntiColl command28Figure 41Result ID from CollMaskVal28Figure 42Parameters in A_Select command29Figure 43Parameters in A_RATS command29Figure 44Parameters in A_PPS command29Figure 45A_Deselect command30	Figure 39 A_Wakeup Command	27
Figure 41 Result ID from CollMaskVal.       28         Figure 42 Parameters in A_Select command       29         Figure 43 Parameters in A_RATS command.       29         Figure 44 Parameters in A_PPS command       29         Figure 45 A_Deselect command       30	Figure 40 Parameters in A_AntiColl command	28
Figure 42 Parameters in A_Select command       29         Figure 43 Parameters in A_RATS command       29         Figure 44 Parameters in A_PPS command       29         Figure 45 A_Deselect command       30	Figure 41 Result ID from CollMaskVal	
Figure 43 Parameters in A_RATS command       29         Figure 44 Parameters in A_PPS command       29         Figure 45 A_Deselect command       30	Figure 42 Parameters in A_Select command	29
Figure 44 Parameters in A_PPS command    29      Figure 45 A_Deselect command    30	Figure 43 Parameters in A_RATS command	29
Figure 45 A_Deselect command	Figure 44 Parameters in A_PPS command	29
	Figure 45 A_Deselect command	

# RD-220 Demonstration Software Manual

Figure 46	Running multiple commands in ISO14443A	31
Figure 47	A_TransparentWithCRC and A_TransparentWithoutCRC for ISO14443A	31
Figure 48	Available commands related to MIFARE in Pi-931	32
Figure 49	MIFARE Standard commands	32
Figure 50	MIFARE Combo commands	33
Figure 51	A_LoadKey commands	33
Figure 52	A_Authentication commands	34
Figure 53	A_ReadBlock commands	34
Figure 54	A_WriteBlock commands	35
Figure 55	A_WriteValueBlock commands	35
Figure 56	Structure of value block	36
Figure 57	Structure of the sector trailer block	36
Figure 58	Access conditions for data blocks	36
Figure 59	Access conditions for the sector trailer	37
Figure 60	A_WriteSectorTrailerBlock commands	37
Figure 61	A_Increment commands	38
Figure 62	A Decrement commands	38
Figure 63	A_Restore commands	39
Figure 64	A_Transfer commands	39
Figure 65	A_Req_Anti_LoadKey_Authent commands	40
Figure 66	A_Req_Anti_LoadKey_Authent_Read commands	40
Figure 67	A_Req_Anti_LoadKey_Authent_Write commands	41
Figure 68	A_Increment_Transfer commands	42
Figure 69	A_Decrement_Transfer commands	42
Figure 70	A_Restore_Transfer commands	43
Figure 71	A_Restore_Transfer commands	43
Figure 72	ISO14443B command list	44
Figure 73	ISO14443B standard commands available in Pi-931	44
Figure 74	B_Request Command	45
Figure 75	B_WakeUp Command	45
Figure 76	B_ATTRIB Command	46
Figure 77	B_Halt Command	46
Figure 78	B_Deselect Command	47
Figure 79	Example of using <b>B_ TransparentWithCRC</b> with SRI4K card	47
Figure 80	Timeout setting for TransparentWithCRC and TransparentWithoutCRC command	47
Figure 81	Example of running multiple ISO14443B command	48
Figure 82	GUI in ISO15693 Tap	49
Figure 83	ISO15693 standard command	50
Figure 84	command mode in ISO15693	50
Figure 85	State transition diagram of the ISO15693 card	50
Figure 86	Flag in inventory mode	51
Figure 87	Sub mode in Non-inventory mode	51
Figure 88	Standard Commands in ISO15693	52
Figure 89	Example of Inventory 1 slot	53
Figure 90	Example of Inventory 16 slot	54
Figure 91	An example of Using Inventory 16 slot in anti-collision	55

# RD-220 Demonstration Software Manual

Figure 92	Stay Quiet Command	55
Figure 93	ReadSingleBlock command	56
Figure 94	WriteSingleBlock command	56
Figure 95	Lock Block command	57
Figure 96	ReadMultipleBlock command	57
Figure 97	WriteMultipleBlocks command	58
Figure 98	Select command	58
Figure 99	ResetToReady command	58
Figure 100	WriteAFI command	59
Figure 101	LockAFI command	59
Figure 102	WriteDSFID command	59
Figure 103	LockDSFID command	60
Figure 104	GetSystemInformation command	60
Figure 105	GetMultipleBlockSecuritystatus command	61
Figure 106	Example of ISO15693 standard command usage	61
Figure 107	Custom commands for SIC5600	62
Figure 108	Set EAS command	63
Figure 109	Reset EAS command	63
Figure 110	Lock EAS command	64
Figure 111	Write Password command	64
Figure 112	Lock Password command	65
Figure 113	Set Password Mode command	65
Figure 114	Lock Password Mode command	66
Figure 115	Get Password Mode command	66
Figure 116	Load Password command	68
Figure 117	Example of Kill operation	69
Figure 118	Set OTP command	70
Figure 119	Get OTP command	70
Figure 120	Write XUID command	71
Figure 121	Read XUID command	71
Figure 122	Special commands for ISO15693	72
Figure 123	Timeout setting for TransparentWithCRC (TransparentWithoutCRC) command	72
Figure 124	Reading data from block 0 by using TransparentWithCRC	72
Figure 125	Pico Tag command tap	73
Figure 125 Figure 126	Pico Tag command tap Speed setup for PICO Tag	73 73
Figure 125 Figure 126 Figure 127	Pico Tag command tap Speed setup for PICO Tag State diagram of PicoTag / PicoPass	73 73 74
Figure 125 Figure 126 Figure 127 Figure 128	Pico Tag command tap Speed setup for PICO Tag State diagram of PicoTag / PicoPass ACTALL command and response	73 73 74 74
Figure 125 Figure 126 Figure 127 Figure 128 Figure 129	Pico Tag command tap Speed setup for PICO Tag State diagram of PicoTag / PicoPass ACTALL command and response IDENTIFY command	73 73 74 74 74
Figure 125 Figure 126 Figure 127 Figure 128 Figure 129 Figure 130	Pico Tag command tap	73 73 74 74 74 75
Figure 125 Figure 126 Figure 127 Figure 128 Figure 129 Figure 130 Figure 131	Pico Tag command tap Speed setup for PICO Tag State diagram of PicoTag / PicoPass ACTALL command and response IDENTIFY command SELECT ASNB command SELECT SNB command	73 73 74 74 74 75 75
Figure 125 Figure 126 Figure 127 Figure 128 Figure 129 Figure 130 Figure 131 Figure 132	Pico Tag command tap	73 74 74 74 75 75 76
Figure 125 Figure 126 Figure 127 Figure 128 Figure 129 Figure 130 Figure 131 Figure 132 Figure 133	Pico Tag command tap Speed setup for PICO Tag State diagram of PicoTag / PicoPass ACTALL command and response IDENTIFY command SELECT ASNB command SELECT SNB command HALT command READ command	73 74 74 74 75 75 76 76
Figure 125 Figure 126 Figure 127 Figure 128 Figure 129 Figure 130 Figure 131 Figure 132 Figure 133 Figure 134	Pico Tag command tap	73 74 74 74 75 75 76 76 76
Figure 125 Figure 126 Figure 127 Figure 128 Figure 129 Figure 130 Figure 131 Figure 132 Figure 133 Figure 134 Figure 135	Pico Tag command tap Speed setup for PICO Tag State diagram of PicoTag / PicoPass ACTALL command and response IDENTIFY command SELECT ASNB command SELECT SNB command HALT command READ command Felica command tap	73 74 74 74 75 75 76 76 76 78
Figure 125 Figure 126 Figure 127 Figure 128 Figure 129 Figure 130 Figure 131 Figure 132 Figure 133 Figure 134 Figure 135 Figure 136	Pico Tag command tap	73 74 74 74 75 75 76 76 76 76 78 78

### RD-220 Demonstration Software Manual

#### List of Table

Table 1-1	Card or card that Pi-931 supports	10
Table 4-1	Dev ID	18
Table 5-1	Meaning of byte in Command Frame Format	24
Table 5-2	Meaning of bytes in Response Frame	<u>2</u> 4
Table 6-1	Supported RF transmission speed	19
Table 6-2	operable mode between command mode and card state5	51
Table 6-3	Standard Command for ISO15693	52
Table 6-4	Custom Command for SIC5600	52
Table 6-5	Summarized protection function controlled by PA (Password Allocation ) and SM (Security Mode)	5
Table 6-6	Password mode (PWD Mode)	56
Table 6-7	Kill Enable : Set up Kill-Enable bit	56
Table 6-8	M : Specify password in card used to compare with transmitted password during loading password6	6
Table 6-9	PA (Password Allocation) : Define passwords to be used in protection	57
Table 6-10	) SM ( Security Mode(1:0) ) : Define protection function of the password6	57
Table 6-1	1 OTP Mode	71
Table 6-12	2 L : Lock Control to Lock current status of OTP MODE	'1
Table 6-13	3 OTP MODE(1:0)	71
Table 6-14	4 Pico Product Family	73

#### 1. Introduction

The RD-220 is a series of 13.56MHz RFID reader. The RD-220 supports all major global secured baseband ISO standards namely ISO14443-A, -B, and ISO15693 as well as the MIFARE classic/plus cards.

This manual describes how to use demonstration software for evaluating and exploring various features of RD-220 through providing protocol. The RD-220 protocol contains basic commands as stated in the ISO protocol and combo commands for one-stop operation as well as arbitrary commands for higher layer protocol communication. Performing RFID operation through the providing protocol is suitable for system integrators or hardware/software developers who have to be involved in the 13.56MHz RFID applications. Demonstration software is designed to guide user/developer to quickly understand the RD-220 protocol and be capable to implement your own RFID software applications. Moreover, the open source C# of the demonstration software is also provided in CD for reference.

The cards or cards that RD-220 can support are summarized in Table 1-1.

Table 1-1 Card or card that RD-220 supports	
Standard	Cards or cards that RD-220 can supports
ISO 14443A	MIFARE Family, ISO14443A memory card, ISO14443A smart card
ISO 14443B	Type-B SRI Series (ST), ISO14443B memory card, ISO14443B smart card
ISO 15693	SIC5600 (SIC), I-Code Family (NXP), Card-it (TI), LRI Series (ST), ISO15693 Label

#### 2. Getting Started

Before user can operate the demonstration software, proper operational environment and the following requirements must be prepared.

#### 2.1 System and Hardware Requirements

•	Computer	: PC with USB Port
•	Operating System	: Windows 7, Windows VISTA, Windows XP SP2
•	RFID Reader	: RD-220 Reader
•	Software Requirement	: .NET framework version higher than 4.0 Installed
•	Others	: Card/Tag

#### 2.2 Content in CD

Content in CD, shown in Figure 1, consists of

- Demonstration Software in folder "Demo\_Software"
- Documents namely Module Datasheets and Software Manuals in folder "Documents"
- .Net Framework in folder "Dot\_Net\_Framework"
- RD220 Driver for emulating RS232 on USB in folder "RD220\_Driver"
- Example software source code in "SW\_Source\_Codes"





#### 2.3 Software Installation

- 2.3.1 Driver Installation (For Virtual RS232 on USB)
  - 1. Connect USB cable to the RD-220 reader Hardware. Then, there will be a "Found New Hardware" pop up indication.





2. Windows will ask for the way for driver installation. Select "No, not this time".

<b>尋找新增硬體精靈</b>	
	歡迎使用尋找新增硬體精霊
	Windows將會搜尋您的電腦、硬體安裝 CD 或 Windows Update 網站(您允許的話)來尋找目前的以及已更新的軟 體。 閱讀我們的隱私擺聲明
- mail	Windows是否可以連線到 Windows Update 尋找軟體?
	○是,只有現在徑)
	○ 是,現在以及每次我連接了一個裝置時(匠)
	諸按 [下一步] 繼續。
	<上一步B) (下一步M)>) 取消

Figure 3 Select "No, not this time".

3. Windows will request for driver installation. Select Option "Install from a list or specific location (Advanced)".



Figure 4 Select Option "Install from a list or specific location (Advanced)"



4. Select "Search for the best driver in these locations" and check "Include this location in the search" box. Then, Browse to folder "D:\RD220\_Driver" in CD or where the driver is located. Click "Next".

諸選擇您的	<b>捜尋和安装選項。</b>
<ol> <li>      • 在這   </li> </ol>	些位置中搜尋最好的驅動程式(3)
使用	下列核取方塊來限制或擴充包括本機路徑和可卸除式媒體的預設搜尋,將安 列的最佳驅動程式。
	搜尋可卸除式媒體 (軟碟, CD-ROM)(M)
<b>V</b>	搜尋時包括這個位置(0):
	D:\RD220_Driver 🛛 🔽 瀏覽(R)
○ 不要 選擇 式最	搜尋,我將選擇要安裝的驅動程式(D) 這個選項來從清單中選取裝置驅動程式。Windows 不保證您所選取的驅動程 符合您的硬體。
	[<上一步(B)〕 「下一步(M) > 」 □ 取消

Figure 5 Browse to folder "RD220\_Driver" in CD or where the driver is located

5. Windows ask to ensure installation the driver from SIC as shown in Figure 6; please click "Continue Anyway".

硬體安裝	
♪	您正要爲這個硬體安裝的軟體: RD220 USB CDC Serial Port
	尚未通過Windows標誌測試以確認它與Windows XP的相容性。(告訴我這項測試的重要性。)
	繼結安裝這個軟管合在現在或指來,使您的系統操作不 看定或受損。Microsoft 強烈建議您立即停止這項安 裝,並進絡硬體嚴商索取已通過Windows 標誌測試的 軟體。
	繼續安裝© 停止安裝◎

Figure 6 Click "Continue Anyway"

6. Wait for driver installation.

尋找新增硬體精靈								
精靈安裝軟體時,請稍候 と								
Ţ	RD220 USB CDC	Serial Port						
	$\triangleright$	Ð		D				
		< <u>+</u>	:一步®)(T	-#@))	取消			

Figure 7 Windows is installing the driver.



7. If installation completes, there will a window indicating installation finish as shown below displayed. Please click Finish.



Figure 8 Installation completes.

8. There is the pop up shows hardware is ready to use. At this time, the device is successfully connected to PC and the LED on the module is turn on.



Figure 9 pop up shows hardware is ready to use

9. User can check the number of comport from Device Manager. As shown in Figure 10, the number of comport for this RD-220 hardware in this example is "COM4". Note that the assigned number of comport is different for each computers.



#### 3. Quick Start with Demonstration Software

The demonstration software is "SIC HF SDK.exe" provided in the folder "*Demo\_Software*". There is no software setup required; just double click the "SIC HF SDK.exe". The demonstration software can run either from CD or a copy on hard drive. The GUI of software is shown in Figure 11 and ready to use.

OPEN PORT CLOSE PORT SET INIT SEQNUM SET DEV ID GET FW VERSION ABOUT			
Reader Configulation ISO15693 ISO14443A ISO14443B	General Reader Command	< Cmd	For Note
Device ID	Baset ASIC (SIC0210)	TX 1 Send	
00 Set Device ID 00 Get Device ID	Reset ASIC (SICSSTU)	TX 2 Send	
	ON RF OFF RF	TX 3 Send	
RF Driver Configulation	RF Power Level	TX 4 Send	iī
Differential Close Coupling Network with internal envelope   Write	Level 4   Set up	TX 5 Send	j
Differential Close Coupling Network with internal envelope		TX 6 Send	iU
	< Send Clear	TX 7 Send	í ———
	Cred	TX 8 Send	í
	ТХ	TX.9 Send	í ———
	BX	TX 10 Send	í ————
		TX 11 Send	í ————
	Clear Logs	TX 12 Send	i I
	Communication Logs 📃 Show Raw Pck	<	J
P			
	4		÷

Figure 11 Demonstration software

Following steps, as shown in Figure 12, demonstrate a simple usage in reading UID of ISO15693 card for quick understanding.

VEN PORT CLOSE PORT SET INIT SEQNOM SET DEVID GET FW VERSION	ABOUT	C (md	For Note
Start Send Multi Cmd Send 10 Corlig Reader Freed TX,RX for Reader + Speed Rx for Tag add in Request Rag of expr Cmd (6)	Reset ASIC (SIC3310)     ON RF OFF RF     RF Power Level	TX1       TX2       TX3       TX4	Send Send Send Send
Setup X 1 out of 256 (1.65 kbite/t • FX 1 Sub Low rate (6.62 kbi • ) SET TX • RX • GET Brable Once Check ISO15693 Standard Command ist Cmd Inventory 1 slot Command	Level 4 - Set up Cond Cond TX 0010 0000	r TX 5 TX 6 TX 7 TX 8 TX 9	Send Send Send Send Send
	R 0010 0000000000000000000000000000	TX 10 TX 11 TX 12	Send Send
With Multiple Blocks Select Reset To Ready With AFF Wath SFID Usch AFF Wath SFID Get System information Get Multiple Block Security Status	□         Tx00.53> 0D00           Rx00.53> 0D00 01           Complete communicate!           Tx00.54> 0D01 01           complete communicate!           Tx00.54> 0D1 01           complete communicate!           Tx00.55> 0D10 01 55 34 0F 22 0C	: 00 01 04 E0 12	
custom SIC Command     SetAs       SetEAs        ResetEAS        UID-sd0FL>55     UID-sd0FL>55       UID-sd0FL>20000104E0     13	Complete Communicate!	)	

Figure 12 Demonstration software

- 1) Connect a Pi-931 device to computer and wait until computer recognize COMPORT as shown in Figure 10. Then click "OPEN PORT" Menu tap to query available COMPORT in computer.
  - 2) Available COMPORT in computer is shown. Click the COMPORT number belonging to reader hardware. If connection is successful, there will be the connected comport with communication speed displayed at bottom of the GUI.
  - 3) Click "ISO15693" tap to selection operation related to ISO15693
  - 4) Click "Setup" button to configure reader IC to be ready to transmit and receive ISO15693 frame format
  - 5) Select preferred speed using in RF transaction. In this example, select "1 out of 256 (1.65 Kbits/s)" for TX (downlink) and "1 Sub Low Rate (6.62 Kbits/s)" (Uplink).

Speed TX,RX for Reader + Speed Rx for Tag add in RequestHag	of e	very Cmd —
TX 1 out of 256 ( 1.65 kbits/s - RX 1 Sub Low rate ( 6.62 kbits/s	i 🔻	SET
TX RX	-	GET



- 6) Click "SET" button, shown in Figure 13, to configure codec speed of reader
- 7) Select Command "Inventory 1 Slot" by clicking on the name of that command in order to read the card UID. The selected command will be highlighted.
- 8) Input necessary parameters. The parameters required in this command are Inventory mode, Mark Len and Mark Value. User can left these inputs blank in this case. Then, value of "0" is used.
- 9) Check "Save UID" check box to save readable UID in software buffer. This UID might be used in further operation.
- 10) Ensure that ISO15693 card is placed in operation range of the hardware reader. Click "Send" to transmit the command to reader hardware to read UID of ISO15693 card.
- 11) Transaction between reader hardware and PC is display in communication logs. For more information about response frame format, please refer to "Pi-931 Protocol" Document.
- 12) For standard commands in command list, the response is translated and displayed in command response section for easy understand. In this case, response is successful by response code "01" while DFSID and UID are shown.

Note that user can watch transactions between reader hardware and PC to be ideas in developing application from each step.



#### 4. Demonstration Software Component

This software mainly consists of four sections as shown in Figure 14 namely

- 1) Port-and-Constant Setup
- 2) RFID standard taps and Reader Configuration tap
- 3) Raw data input and output for developer
- 4) Transaction logs.

OPEN PORT CLOSE PORT SET INIT SEQNUM SET DEV ID GET FW VERSION	ABOUT	3			
Reader Configulation ISO15693 ISO14443A ISO14443B		General Reader Command	Cmd		For Note
Device ID		Reset ASIC (SIC9310)	TX 1	Send	
OO Set Device ID OO Get Device ID			TX 2	Send	
	-	ON RF OFF RF	TX 3	Send	
RF Driver Configulation		RF Power Level Set up	TX 4	Send	
Differential Close Coupling Network with integral envelope		Level 4	TX 5	Send	
			TX 6	Send	
		< Clear	TX 7	Send	
		Cmd	TX 8	Send	
		XT	TX 9	Send	
		RX	TX 10	Send	
		Gear Loos	TX 11	Send	
		Communication Lane R. Show Prov. Pol.	TX 12	Send	
		Communication Logs Show Haw Pok	< III	_	+
		•			

#### Figure 14 Sections in Demonstration software

#### 4.1 Port-and-Constant Setup

This section consists of Menus related to hardware setup namely OPEN PORT, CLOSE PORT, SET INIT SEQNUM, SET DEV ID, GET FW VERSION and ABOUT.



Figure 15 Port-and-Constant Setup

#### 4.1.1 OPEN PORT

**OPEN PORT** is used to query and open communication port to the reader device.

OPEN PORT	CLOSE	PORT	SET INIT SE	QNUM	SET DEV ID	GET FW VERSION	ABOU
COM15	IS	015693	ISO14443A	ISO14443	В		

Figure 16 **OPEN PORT** Menu



### RD-220 Demonstration Software Manual

Following steps describe opening the communication port with Pi-931.

- Click **OPEN PORT** menu to search available com port present in computer.
- Available COM ports are shown in menu content under **OPEN PORT** menu.
- Click on the COM port number belonging to reader hardware being operated to open communication.
- If connection is successful, there will be a connected comport with communication speed displayed at bottom of the GUI, as shown in Figure 17.



Figure 17 connected com port and communication speed (kbps)

#### 4.1.2 CLOSE PORT

**CLOSE PORT** is used to close current communication port.

OPEN PORT	CLOSE PORT	SET INIT SEQNUM	SET DEV ID	GET FW VERSION	ABOUT
Reader Configula	ation ISO15693	ISO14443A ISO1444	3B		

#### Figure 18 CLOSE PORT Menu

- Click CLOSE PORT to close current operating COM port.
- The connection shown at bottom of the GUI disappears.

#### 4.1.3 SET INIT SEQNUM

**SET INIT SEQNUM** is used to set initial sequence number **SEQNUM** in transmission command packet. The **SEQNUM**, 1-byte information, is automatically increased after transmission of each packet by this software. As shown in Figure 19, user can input two-hexadecimal number to set the initial sequence number.

OPEN PORT	CLO	OSE PORT	SET I	NIT SEC	NUM	SET DEV ID	GET FW VERSION	ABOUT	
Reader Configula	ation	ISO15693		FF	01444	3B			

Figure 19 SET INIT SEQNUM Menu

#### 4.1.4 SET DEV ID

SET DEV ID is used to define target Device ID in transmission command packet. The Device ID (Dev ID) indicates specific device or all in a network to operate. The Dev ID is consisted of an ID and a silent control bit. The ID can be between 0x00 and 0x7F. The most significant bit in the Dev ID is the silent bit in which the operating devices will not response back to host. The device that receives matched ID from incoming packet will operate and respond back to host. The Dev ID in this setting can be from 0x00 and 0xFF. Then, there will be no response from packet containing the Dev ID between 0x80 and 0xFF. For Device ID of 0x00, all devices will operate and response back to host. The detail of Dev ID is summarized in Table 4-1. User can define Dev ID in packet by entering two- hexadecimal number as shown in Figure 20.

Reader Configulation ISO 15693 ISO 14443A ISO 1444 06		OPEN PORT	CLO	OSE PORT	E PORT SET INIT SEQNUM		SET	DEV ID	GET FW VERSION	ABOUT	
	Γ	Reader Configula	ation	ISO15693	ISO14443A	ISO1444		06			

Figure 20 SET DEV ID Menu

Table 4-1 Dev ID						
Silence Bit <sup>(1)</sup>	Device ID					
Dev_ID[7]	Dev_ID [6:0]	Meaning				
0	0x00 <sup>(2)</sup>	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	0x00	The operating device will not respond back to host				

- (1) The silence bit is an option for preventing data collision in reader network from simultaneous response.
- (2) Device ID 0x80 can use for broadcasting if there are multiple readers connected in a network.

#### 4.1.5 GET FW VERSION

**GET FW VERSION** is used to get firmware version from the connected device. Firmware version is reported as shown in Figure 22

OPEN PORT	CLOSE PORT	SET INIT SEQNUM SE		SET DEV ID	GET FW VERSION	ABOUT	
Reader Configulation	n ISO15693	ISO14443A	ISO14443	3B		ſ	_

Figure 21 GET FW VERSION Menu

#### 4.2 RFID standard taps and Reader Configuration tap

This section consists of three RFID standard taps and Reader Configuration tap. GUI in ISO15693 standard tap is shown in Figure 23. In ISO14443A and ISO14443B tap contains similar component with different commands. For Reader configuration as shown in Figure 26, the provided GUI is for setting up and getting Device ID and RF Driver configuration.

#### 4.2.1 RFID standard taps

Each RFID standard tap contains data transmission button, RF communication speed selection, parameters setup of each associated command as well as response reports from RF transaction of each command.

#### 4.2.1.1 "Standard Setup" Button

"Setup" Button is used to setup parameters in the reader to be ready to transmit and receive following standard in tap name. User must activate this setup before performing any RF-related operations in associated RFID protocol.

#### 4.2.1.1 "Send" Button

"Send" Button is used to transmit command that is currently highlighted in command list

#### 4.2.1.2 "Start Send Multi Cmd" Button

"Start Send Multi Cmd" button is used to transmission multiple commands consecutively that are checked in command list. From example in Figure 23, providing that no any command fail during each operation, *Inventory 1 Slot, Read Single Block, Write Single Block* and *Select* are operated in chronological order respectively. Note that command in the list is run from top to button.

#### 4.2.1.3 Reader TX and Rx Speed selection

This section is used to setup transmission and reception speed of CODEC in reader IC. User must setup the transaction speed before performing any RF-related operations in associated RFID protocol. In addition, current setup speed in the reader can be retrieved by "Get" button.

Reader Configulation ISO15693 ISO1444	ISA ISO14443B	
Start Send Multi Cmd Config Reader Speed TX,RX for Reader + Sp TX 1 out of 256 (1.65 kbits TX 1	Send       Deed Rx for Tag add in Request Rag of every Cmd       /s → RX       1 Sub Low rate (6.62 kbit ▼ SET       RX       GET	
5 Enable Once Check 6 List Cmd Virtertary 1 Slot ReadSingleBlock Virte SingleBlock Virte SingleBlock Contemporal Contemporal Virte SingleBlock Contemporal Contemporal Virte MultipleBlocks Virte MultipleBlocks Virte AFI Lock AFI Cock AFI Cock AFI Cock DSFID Cock DSF	7       ISO15693 Standard Command         Inventory 1 slot Command       Inventory 1 slot Command         Inventory 1 slot Cmd       Inventory 1 slot Cmd         Inventory 1 slot Cmd       Option Flag         © Option Flag       Protocal Extension Flag         Enable AFI       Mask Len         Mask Value       Mask Value         8       Inventory 1 slot Cmd Response         Response> 01 - Resp OK       DSFID> 00         UID> C03398529C2102E0       VID	Ш

Figure 23 Example of button and setup in ISO 15693 tap

#### 4.2.1.4 "Enable once check" Check box

The "Enable Once check" is an option in selecting command on command list. If "Enable Once check" box is checked, a pointed command in command list is checked on one click.

#### 4.2.1.5 Command List

The command list provides air-interface commands in each RFID standard. The commands include primitive command based on operating RFID standard and combo command for one stop operation as well as some proprietary commands for SIC RFID.

#### 4.2.1.6 Parameter setup

The section is for setting up parameters required by each command. The section is different from command to command. User may have to refer to ISO standards for meaning and function of each parameter. If the inputs are left blank, value of "0" is used.

In some command, there is a "Load Saved UID" checked box to load UID from previous operation to use in current operation. For the command required UID in operation, the "Load Saved UID" checked box is provided.

Select Command				
Select Cmd Non Inventory Mod	de			
Option Flag				
Protocal Exten	sion Flag			
Operation Mode	Address Mode 👻			
UID	Load SavedUID			

Figure 24 "Load Saved UID" checked box

#### 4.2.1.7 Command Response

The Command Response section displays and translates operation result and available response of each command. As shown in Figure 25, "01" indicates response is successful without any error. Also, received UID and DSFID in ISO15693 are displayed. The UID in response packet starts from least significant byte to most significant byte which reflects to what transmit from card in chronological order. For more information about meaning of response, please refer to Pi-931 Protocol.

In some command, there is a "Save UID" check box for storing UID form current transaction for further operations. To store UID in software buffer, this checked box must be set prior operating command.



Figure 25 "Save UID" checked box

#### 4.2.2 Reader Configuration tap

Reader Configulation ISO15693 ISO14443A ISO14443B					
Device ID 1 00	Set Device ID 00 Get Device ID				
- RF Driver Configula	tion				
Diffential Close Coupling Network with internal envelope   Write					
Diffential Close Coupling Network with internal envelope   Read					

Figure 26 Command and setup in Reader Configuration tap

#### 4.2.2.1 "Device ID" Setup

"Set Device ID" is for programming Device ID for individual reader. Only device ID between 0x00 and 0x7F is allowed. Also, device ID of connected device can be read from "Get Device ID" button.

#### 4.2.2.2 "RF Driver Configuration" Setup

This section is for setup driver configuration of the hardware. The driver configuration is information used by microcontroller to set up driver characteristic to serve specific RF topology in Pi931 family. This information is stored in EEPROM of SIC9310. To make the reader properly operable and achieve the highest performance, the driver configuration must be specifically set to match to RF topology of hardware. The RF topology can be one of these configurations namely

- 1. Differential driver with internal envelop detector,
- 2. Differential driver with external envelop detector,
- 3. Single ended driver with external envelop detector,
- 4. 50-ohm-output Class-E driver with external envelop detector.

For every Pi-931 reader manufactured from silicon craft, the "RF Driver Configuration" is pre-programmed to match RF driver topology of the reader hardware. So, the "RF Driver Configuration" is provided for users who designs their own hardware based on Pi931-MD module and requires to reconfigure hardware constant to different RF driver topology.

#### 4.3 Raw data input and output for developer

This section is provided for developers to communicate with the hardware by hexadecimal code. Example of usage this section are transmitting commands not provided in command list. This is a common situation in 13.56-MHz RFID application development. Moreover, this section can be used for debugging developing applications. To use this section, users have to input command in Hexadecimal format in TX and decode responses from hardware shown in RX by user defined programming or manually. For more information about module communication protocol and hexadecimal code, please refer to Pi-931 protocol document. The section is consisted of six components, which will be described below.

General Reader Command	Cmd	12		For Note
1 Reset ASIC (SIC9310)	TX 1		Send	
	TX 2		Send	
2 ON RF 3 OFF RF	TX 3		Send	
RF Power Level	TX 4		Send	
Level 4 V	TX 5		Send	
	TX 6		Send	
< Send Olear	TX 7		Send	
Cmd	TX 8		Send	
TX	TX 9		Send	
RX	TX 10		Send	
10 Clear Logs	TX 11		Send	
	TX 12		Send	
Communication Logs Show Raw Pck	•			*

Figure 27 Raw data input and output for developer

#### 4.3.1 "Reset ASIC" button

This button is used to reset reader IC (SIC9310)

#### 4.3.2 "ON RF" button

This "ON RF" button starts 13.56-MHz carrier emission.

#### 4.3.3 "OFF RF" button

This "OFF RF" button stops 13.56-MHz carrier emission.

#### 4.3.4 "RF Power Level" Selection

This pull down menu is for selecting transmitting strength of 13.56-MHz carrier.

#### 4.3.5 "Set up" button (Power Level)

The Set up button is for configuring strength of transmitting carrier following defined RF Power Level.

#### 4.3.6 "Width Restore (<)" button

This button is used to restore width and show all part of sub panel "RFID standard taps".



#### 4.3.7 "Send" button

The "Send" button is used to transmit command and data in TX box in section 4.3.3.

#### 4.3.8 "Clear" button

The "Clear" button is used to clear content in section 4.3.3.

#### 4.3.9 Direct TX Input and RX Output

TX Input consists of command box and its associated data to be send to RFID reader while RX output consists of feedback command mode and response data from RFID reader. As shown in Figure 28, "0x0A13", which performs select command in ISO14443A, is the transmitted command and "0x001CBB4FCD" is the transmitted data to RFID reader. In RX box, "0x0A13" is feedback command to confirm operation and "0x0108" is response data from RFID reader.



#### Figure 28 Example of Direct TX Input and RX Output

#### 4.3.10 "Clear Logs" button

The "Clear Logs" button is used to clear all logs event in Transaction Log display.

#### 4.3.11 "Show Raw Pck" Check box

If "Show Raw Pck" is checked, all every transaction between reader and host is displayed in full format. This will help user understand frame format structure of Pi-931 Protocol from header to checksum. The effect of "Show Raw Pck" is shown in Figure 29.

TX> AA 00 06 5 RX> AA 00 0E 5	5 00 0D 10 5 00 0D 10	00 00 01 00	4E C0 33	98 52	9C 21	02 E	0 21	←→	<b>V</b>	Show Raw Pck
Complete Commu	nicate!									
Rx00_56> 0D10 Complete Commu	01 00 C0 3	3 98 53	2 9C 21	L 02 E	D			<b></b>		Show Raw Pck

Figure 29 Effect of "Show Raw Pck" in transaction display

#### 4.3.12 Additional TX Input and RX Output section

The function of this section is similar as described in 4.3.4. The purpose of this section is for testing consecutive arbitrary command unavailable in command list. User can review each result in the Transaction Log. In addition, the box on the most right is for recording and comment. This help developer recognize meaning of each hexadecimal command. Usage of this section is shown in Figure 30 Usage of additional TX input and RX output sectionFigure 30.

<	Cmd			For Note
TX 1	0A00		Send	Setup CODEC to ISO14443A
TX 2	0A01	00	Send	Set CODEC speed to 106 k for Tx and Rx
TX 3	0A11		Send	WakeUp A
TX 4	0A12	0000	Send	Anticoll A, Casecade level =0, Coll Mask Val = 0
TX 5	0A13	00DB2DF198	Send	Select Card, Casecade level =0, UID = DB2DF198
TX 6	0A15	01	Send	RATS command, FSD = 16 Byte, CID = 1
TX 7	0A16	D1110A	Send	Protocol/Parameter select, CID =1, Rx = 424, Tx = 424
TX 8	0A01	22	Send	Change CODEC speed to 424k for Tx and Rx
TX 9	0AC0	01C2	Send	Deselect
TX 10		I	Send	
TX 11			Send	
TX 12			Send	
TX 13			Send	

Figure 30 Usage of additional TX input and RX output section

#### 4.3.13 "<" button

The "<" button in "Additional TX Input and RX Output section" is used to extend or shrink this section for easy accessibility. When the symbol on the button is "<", click the button extend the section. Once, the symbol on the button become ">", click the button shrink this section.

#### 4.4 Transaction Logs

If "Communication Logs" check box is checked, transaction between host and the device is logged and displayed in this section as shown in Figure 31.

Tx00\_64> 0A11 Rx00\_64> 0A11 01 04 00 Complete Communicate! Tx00\_65> 0A12 00 00 Rx00\_65> 0A12 01 1C BB 4F CD Complete Communicate! Tx00\_66> 0A13 00 1C BB 4F CD Rx00\_66> 0A13 01 08 Complete Communicate! Tx00\_67> 0A14 Rx00\_67> 0A14 01 Complete Communicate!

Figure 31 Transaction Logs text box.

#### 5. Pi-931 Protocol

To operate with Pi-931 module effectively, user should prior be familiar with technical terms, transaction speeds and parameters required in ISO14443 and ISO15693 as well as MIFARE. The command available in this demonstration software is directly mapped from existing command in the protocol. So, user can understand how to interact with card step by step by manually inputting and activating commands at appropriate state of card and can learn how to use our protocol and compose correct structure from this demonstration software.

To create application based on Pi-931 module, user should be able to transmit command in hexadecimal code and review response from the reader device. The Pi-931 protocol itself is basically a series of hexadecimal code. The command frame format is shown Figure 32 while Table 5-1 summarize meaning of bytes in frame. For the response frame, frame format detail is shown Figure 33 and Table 5-2.

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

Table 5-1 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 (7 Bit)			
CMD Category	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]			

Figure 32 Command 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

Figure 33 Command Frame Format

Table 5-2 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]			

### RD-220 Demonstration Software Manual

The relation between hexadecimal of command and response displayed in transaction logs window and frame format as stated above is shown in Figure 34. This example shows performing inventory1slot in ISO15693. To display only necessary numbers in command and response packet, user can uncheck "Show Raw Pck" as explain in 4.3.9. Figure 35 depicted decomposition of abridged frame format.



Figure 34 Decomposition of raw frame format of Pi-931



Figure 35 Decomposition of abridged frame format displayed in demonstration software

#### 6. Using Demonstration Software

#### 6.1 ISO14443A

Figure 36 shows GUI in ISO14443 tap. In the tap, user can transmit single or multiple commands, setup transaction speed as well as review card response. Command list consists of standard commands, MIFARE commands, Combo commands and transparent commands.

Reader Conf	igulation ISO15693 ISO	014443A ISO14443B
Sta	rt Send Multi Cmd	J Send
Config Re	TX 106 KBits/s	er ▼ RX 106 KBits/s ▼ SET ↓ RX ▼ GET
List Cmd A. Ma A. A. Are A. J. M. A. A. Are A. J. A. Are A. J. A. Are A. J. A. Are A. A. Are A. A. Ma A. A. Ma A. J. Ma	Enable Once Check tandard Command	ISO14443A Standard Command None Command

Figure 36 ISO14443A command list

#### 6.1.1 ISO14443A standard commands

The standard commands used in ISO14443A-3 and -4 are illustrated in Figure 38, showing relation of each command in card-state transition diagram. Hence, each command shall be applied at appropriate state. The commands in this section are **A\_Request**, **A\_WakeUp**, **A\_Anticoll**, **A\_Select**, **A\_RATS**, **A\_PPS**, **A\_Halt** and **A\_Deselect**. For more information about the commands, please refer to ISO14443-3 and -4 standards and Pi-931 Protocol.

#### 6.1.1.1 A\_Request

The **A\_Request** command GUI is shown Figure 37. This command is used to probe if ISO14443A cards are in there field. This command requires no input.

List Cmd		A_Request Command
	E	A_Request Cmd
A_Decrement A_Restore A_Transfer	-	Response > 01 - Resp OK ATQA> 0400 - UIDSize = SingleLvI , BitFrame/

Figure 37 A\_Request Command

### RD-220 Demonstration Software Manual



Figure 38 ISO14443A standard commands available in Pi-931

#### 6.1.1.2 A\_WakeUp

The **A\_WakeUp** command GUI is shown Figure 39. This command is used to probe if ISO14443A cards are in there field. This command requires no input.



Figure 39 A\_Wakeup Command

#### 6.1.1.3 A\_Anticoll

The **A\_Anticoll** command GUI is shown Figure 40. The **A\_Anticoll** not only transmits command as in the standard but also automatically seek out a complete UID from cards in the RF field. Providing that the cards performed anti-collision are in the same range, the parameter CollMaskVal, as shown in Figure 40, in **A\_Anticoll** can be used to select a card that condition matches. If the CollMaskVal is 0, the reader selects the UID from the card

### RD-220 Demonstration Software Manual

where the first collided bit is 0, and vice versa. The example of CollMaskVal usage is shown in Figure 41. If CollMaskVal is 1, UID of "3A 2A 10 92" is selected. If CollMaskVal is 0, UID of "0A BD 0F 92" is selected. The collision occurs at the fifth bit where first byte response from one card is 0x3A (00111010) and other is 0x0A (00001010). Note that shown UID in response packet starts from least significant byte to most significant byte which reflects to what transmits from card in chronological order. However, other factors such as transmission energy, size of antenna and coupling factor from card and others in RF environment introduce performance of this feature CollMaskVal. This feature can work well in reader equipped with proper antenna. Moreover, the **A\_Anticoll** supports UID cascade level from level-1 to level-3.



Figure 40 Parameters in A\_AntiColl command



Figure 41 Result ID from CollMaskVal

#### 6.1.1.4 A\_Select

The **A\_Select** command GUI is shown Figure 42. This command requires complete UID from the **A\_Anticoll** command before operating. The **A\_Select** supports UID cascade level from level-1 to level-3.

#### 6.1.1.5 A\_RATS (Request for answer to select)

The **A\_RATS** command GUI is shown Figure 43. This command requires FSD (Maximum PCD frame size) and CID (Card Identifier). This command shall be activated after card is selected the active state by **A\_Select** command. The response from this command indicates card capabilities namely transaction speed, Frame wait time, Start-up Frame guard time, NAD support, CID support and History bytes. For more information about meaning of the parameters, please refer to ISO14443-4. This command is in ISO14443A-4 which is the beginning point in entering smartcard protocol ISO7816.

### RD-220 Demonstration Software Manual

List Cmd	A_Select Command
Standard Command	A Select Cmd
A_Request	Cascade Level Default value
A_WakeUp	
A_AntiColl	UID DB2DF198
A_Select	Load SavedUID
Mifare Command	
A LoadKey	
A_Authentication	
A_ReadBlock	
A_WriteBlock	
A_WriteValueBlock	
A_WriteSectorTrailerBlock	
A_Increment	A_Select Cmd Response
A_Decrement	Response> 01 - Resp OK
A_Restore	SAK> 28 - UID Complete ,Compliant Lv4
A_transfer	
Figure 42 Paramete	ers in A_Select command
List Critic	ARAISCOmmand
Standard Command	A_RATS Command
A_Request	A_RATS Command A_RATS Cmd
A_Request	A_RATS Command A_RATS Cmd Param FSD (Max PCD frame size ) 10.0 +
□Standard Command ▲ A_Request ☑ A_WakeUp ☑ A_AntiColl	A_RATS Command A_RATS Cmd Param FSD (Max PCD frame size ) 16 Bytes V
Standard Command     A_Request     A_WakeUp     A_AntiColl     A_Select	A_RATS Command A_RATS Cmd Param FSD (Max PCD frame size ) 16 Bytes V CID 0 V
Standard Command     A_Request     A_WakeUp     A_AntiColl     A_Select     A_Halt     A_Halt	A_RATS Command A_RATS Cmd Param FSD (Max PCD frame size ) 16 Bytes V CID 0 V
A_Request     A_AWakeUp     A_AttiColl     A_AttiColl     A_Select     A_Halt     A_RATS     A PPS     E	A_RATS Command A_RATS Cm Param FSD (Max PCD frame size ) 16 Bytes V CID 0 V
A_Request     A_ArtiColl     A_BATS     A_BATS     A_Deselect	A_RATS Command A_RATS Cmd Param FSD (Max PCD frame size ) 16 Bytes • CID 0 •
Standard Command     A_Request     A_Request     A_ArtiColl     A_ArtiColl     A_Select     A_Halt     A_RATS     A_Deselect    Wfare Command	A_RATS Command A_RATS Cmd Param FSD (Max PCD frame size ) 16 Bytes V CID 0 V
Image: Standard Command         A_Request         A_ArtiColl         A_Select         A_Halt         A_RATS         A_Deselect        Mirare Command         A_LoadKey	A_RATS Command A_RATS Cmd Param FSD (Max PCD frame size ) 16 Bytes V CID 0 V
all condent of the second of	A_RATS Command A_RATS Cmd Param FSD (Max PCD frame size ) 16 Bytes V CID 0 V
Image: Standard Command         A_Request         A_AWakeUp         A_AntiColl         A_Select         A_Halt         A_BATS         A_Deselect         A_LoadKey         A_Authication         A_RateCommand	A_RATS Command A_RATS Cmd Param FSD (Max PCD frame size ) 16 Bytes • CID 0 •
Standard Command         A_Request         A_ArtiColl         A_ArtiColl         A_Select         A_Halt         A_PPS         A_Deselect        Wifare Command         A_LoadKey         A_Authentication         A_ReadBlock         A_WriteBlock	A_RATS Command A_RATS Cmd Param FSD (Max PCD frame size ) 16 Bytes • CID 0 •
Standard Command     A_Request     A_Request     A_ArtiColl     A_ArtiColl     A_ArtiColl     A_Select     A_Halt     A_PPS     A_Deselect    Mfare Command     A_LoadKey     A_Authertication     A_ReadBlock     A_WriteBlock     A_WriteValueBlock     A_WriteValueBlock	A_RATS Command A_RATS Cmd Param FSD (Max PCD frame size ) 16 Bytes V CID 0 V
A_Request     A_Nequest     A_WakeUp     A_AntiColl     A_Select     A_Hat     A_Beselect     A_Hat     A_Deselect    Mfare Command A_LoadKey     A_WriteValueBlock     A_WriteValueBlock     A_WriteSectorTrailerBlock     A_WriteSectorTrailerBlock	A_RATS Command A_RATS Cmd Param FSD (Max PCD frame size ) 16 Bytes V CID 0 V
A_Request     A_Available     A_Available     A_Available     A_AntiColl     A_Select     A_AntiColl     A_Select     A_Halt     A_RATS     A_Deselect     A_Halt     A_AratS     A_Deselect     A_LoadKey     A_Authentication     A_ReadBlock     A_WriteBlock     A_WriteBlock     A_WriteSectorTrailerBlock     A_Increment     A_Docement	A_RATS Command A_RATS Cmd Param FSD (Max PCD frame size ) 16 Bytes • CID 0 •
A_Request     A_Request     A_AttGol     A_AttGol     A_AttGol     A_AttGol     A_AttGol     A_Select     A_Halt     A_BATS     A_Deselect     A_Halt     A_BATS     A_Deselect     A_LoadKey     A_AttGol     A_AttGol     A_MiteBlock     A_WriteBlock     A_WriteSectorTrailerBlock     A_Decrement     A_Decrement     A_Betoge	A_RATS Command A_RATS Cmd Param FSD (Max PCD frame size ) 16 Bytes • CID 0 • A_RATS Cmd Response Response> 01 - Resp OK
Standard Command     A_Request     A_Request     A_AntiCol     A_ArtiCol     A_ArtiCol     A_ArtiCol     A_Belect     A_Hait     A_PPS     A_Desselect    Mfare Command     A_LoadKey     A_Authentication     A_ReadBlock     A_WriteBlock     A_WriteBlock     A_WriteBlock     A_WriteBlock     A_WriteSectorTrailerBlock     A_Norsent     A_Restore     A_Restore     A_Restore     A_Restore     A_Restore     A_Restore	A_RATS Command A_RATS Cmd Param FSD (Max PCD frame size ) 16 Bytes • CID 0 • A_RATS Cmd Response A_RATS Cmd Response Response 01 - Resp OK TL> 0E
Image: Standard Command       A         A_Request       A         A_AratiColl       A         A_AratiColl       A         A_AratiColl       A         A_AratiColl       A         A_Astricoll       A         A_Astricoll       A         A_Astricoll       A         A_Astricoll       A         A_Astricoll       A         A_LoadKey       A         A_Authentication       A         A_AddBlock       A         A_WriteBlock       A         A_WriteBlock       A         A_Iorement       A         A_Decrement       A         A_Restore       A         A_Transfer       *	A_RATS Command A_RATS Cmd Param FSD (Max PCD frame size ) 16 Bytes • CID 0 • A_RATS Cmd Response A_RATS Cmd Response Response> 01 - Resp OK TL> 0E TD> 78 TA> 33
A. Control      A. Arato     A. AntiColl     A. AntiColl     A. AntiColl     A. AntiColl     A. Select     A. Hat     A. Arato     A. Arathertication     A. Arator	A_RATS Cmd Param FSD (Max PCD frame size ) 16 Bytes • CID 0 • A_RATS Cmd Response A_RATS Cmd Response Response> 01 - Resp OK TL> 0E TD> 78 TA> 33 Taq Tx Bit Rate - 106kbit 212kbit 424kbit
A_Request     A_Request     A_WakeUp     A_AttiColl     A_Select     A_Halt     A_RATS     A_PPS     A_Deselect     A_Halt     A_RATS     A_Deselect     A_Udhentication     A_ReadBlock     A_WriteBlock     A_WriteBlock     A_WriteSlock     A_WriteSlock     A_WriteSlock     A_Decrement     A_Decrement     A_Decrement     A_Restore     A_Transfer     *	A_RATS Command A_RATS Crid Param FSD (Max PCD frame size ) 16 Bytes • CID 0 • A_RATS Crid Response A_RATS Crid Response Response> 01 - Resp OK TL> 0E T0> 78 TA> 33 Tag Tx Bit Rate - 106kbit 212kbit 424kbit Tag Rx Bit Rate - 106kbit 212kbit 424kbit
Image: Standard Command         A_Request         A_Nequest         A_AntiColl         A_AntiColl         A_Select         A_Halt         A_BATS         A_Deselect         A_LoadKey         A_LoadKey         A_Attribulction         A_Ration         A_NeadBlock         A_WriteBlock         A_WriteSectorTrailerBlock         A_Increment         A_Decrement         A_Restore         A_Transfer	A_RATS Cmd Param FSD (Max PCD frame size ) 16 Bytes CID 0 A_RATS Cmd Response A_RATS Cmd Response Response>01 - Resp OK TL> 0E TD> 78 TA> 33 Tag Tx Bit Rate - 106kbit 212kbit 424kbit Tag Rx Bit Rate - 106kbit 212kbit 424kbit TB> 91 TB> 91

Tics 02 PICC - Not Support NAD ,Support CID HistoryBytes> 3131313054434F5350

#### 6.1.1.6 A\_PPS (Protocol and parameter selection request)

The **A\_PPS** command GUI is shown Figure 44. This command is used to select operating protocol and parameter to use in further operation. This command requires CID, PPSO and PPS1. This command shall be activated next to **A\_RATS** command. After executing this command, user shall set the transaction speed match to what card response. For more information about meaning of the parameters, please refer to ISO14443-4.

List Cmd	A_PPS Command
Standard Command     A_Request     A_NakeUp     A_AntColl     A_AtColl     A_AtColl     A_Atat     A_Atat     A_ARTS     A_PPS     A_Deselect    Mifare Command A_LoadKey     A_Authentication     A_ReadBlock     A_WriteSlock     A_WriteSlock     A_WriteSlock     A_WriteSlock     A_WriteSlock     A_WriteSlock	A_PPS Cmd PPSS CID 0 ▼ PPS0 ✓ Transmit PPS1 PPS1 Speed Tag Send (DSI) Div 4(424 KBits ▼ (Rx of Reader) Speed Tag Receive (DRI) Div 4(424 KBits ▼ (Tx of Reader)
A_Increment A_Decrement A_Restore A_Transfer	A_PPS Cmd Response Response> 01 - Resp OK PPSS> D0 - CID = ID0

Figure 44 Parameters in A\_PPS command

### RD-220 Demonstration Software Manual

#### 6.1.1.7 A\_Deselect

The **A\_Deselect** command GUI is shown Figure 45. This command is used to quit operation of ISO14443-4 or higher layer to idle state of ISO14443-3. This command requires no input and, actually, this command is made up of **A\_TransparentWithCRC.** For more information about meaning of the parameters, please refer to ISO14443-4.



Figure 45 **A\_Deselect** command

#### 6.1.2 Performing Basic operation in ISO14443A standard

Figure 46 shows steps in performing basic command in ISO14443A from wake up to Halt through "Start Send multiple cmd" button. The steps are described as follows.

- 1. Select ISO14443A Tap
- 2. Click "Setup" to initialize reader to received ISO14443A frame
- 3. Define transaction speed. For ISO14443A, speed must be start from 106kbps for both TX and RX.
- 4. Click "Set" to configure transaction speed
- 5. Select command to be run by checking box in front of the command. In this example, **A\_WakeUp**, **A\_Anticoll**, **A\_Select** and **A\_Halt** are selected.
  - 5.1 Select Cascade Level in **A\_Anticoll** command to Default value and ensure that "Save UID" box is checked.
  - 5.2 Select Cascade Level in **A\_Select command** to Default value and ensure that "Load Saved ID" box is checked
- 6. Click "Start Send Multi Cmd" button to run all selected command in defined order. Ensure that ISO14443A card is placed in vicinity of reader.
- 7. If no error occurs, transaction results of each step are shown.
- 8. Results from running each command are shown in associated response box.

Note that if user requires running command not in provided order; please use "Raw data input and output" section as described in 4.3.10.





Figure 46 Running multiple commands in ISO14443A

If commands to be transmitted are not provided in command list such as command for smart card, user can use **A\_TransparentWithCRC** and **A\_TransparentWithoutCRC** to directly transmit hexadecimal code to air. User may have to select preferred timeout period for no response, if some smartcard take long time in processing before response. The **A\_TransparentWithCRC** and **A\_TransparentWithoutCRC** GUI is shown in Figure 47.

List Cmd	A_TransparentWithCRC Command
A_LoadKey	A TransparentWithCRC Cmd
A_Authentication	
A_ReadBlock	PreviousSetting -
A_WriteBlock	TxRF
A_WriteValueBlock	
A_WriteSectorTrailerBlock	A
A_Increment	
A_Decrement	
A_Restore	
A_Transfer	
Mifare Combo Command	
A_Req_Anti_LoadKey_Authent	
A_Req_Anti_LoadKey_Authent_	
A_Req_Anti_LoadKey_Authent_	
A_Increment_Transfer	-
A_Decrement_Transfer	
A_Restore_Transfer	A Transparent/WithCRC Cmd Response
Special Command	
A_TransparentWithCRC	
A Transparent Without CRC	

Figure 47 A\_TransparentWithCRC and A\_TransparentWithoutCRC for ISO14443A

#### 6.1.3 MIFARE

MIFARE is encryption applications based on ISO14443A and operate on transaction speed of 106kbps. Before performing MIFARE command, a card must be prior selected in active state. The MIFARE commands Pi-931 can support are shown in Figure 48. There are basic MIFARE commands and MIFARE combo commands. The basic command are single operation commands as stated in MIFARE datasheet while combo commands are consecutive specific-operation commands from REQA to specific command in MIFARE such as read and write. The combo commands ease user to shorten reader operating time, development time and complexity of coding in user application.



Figure 48 Available commands related to MIFARE in Pi-931

Relation between Pi-931 commands in MIFARE state transition diagram are shown in Figure 49 and Figure

50.



Figure 49 MIFARE Standard commands

RD-220 Demonstration Software Manual



Figure 50 MIFARE Combo commands

#### 6.1.3.1 A\_LoadKey

The **A\_LoadKey** command GUI is shown Figure 51. This command requires a 48-bit hexadecimal key matched the key in memory sector on card being authenticated. The supplied key is loaded into key buffer in reader IC to be used authentication.

List Cmd	A_LoadKey Command
List Cmd	A_LoadKey Command A_Loadkey Cmd Key FFFFFFFFFF
A_Decrement A_Restore A_Transfer	A_Loadkey Cmd Response Response > 01 - Resp OK

Figure 51 A\_LoadKey commands

#### 6.1.3.2 A\_Authentication

The A\_Authentication command GUI is shown Figure 52. Parameter as follows shall be specified.

- Select Key : Key A or Key B on card used in authentication
- Block Num : The block address to be accessed
- UID : Unique ID of card to be accessed

This operation shall be performed after executing **A\_LoadKey**. For typical 1k-byte MIFARE card, every four blocks is governed by keys (A and B) of each sector. For example, block 4, 5, 6 and 7 rely on the same key stored in block. If block 4 was authenticated, block 5, 6 and 7 can also be accessed with re-authentication. If authentication is successful, reader will return successful code "0x01" as shown in Figure 52.

List Cmd Standard Command A_Request A_WakeUp A_WakeUp A_ArtiColl A_Select A_Halt A_RATS A_PPS A_DeselectMfare Command V A_LoadKey A_LoadKey A_Mathentication A_ReadBlock A_WriteBlock		A_Authentication Command A_Authentication Cmd Select Key Key A Block Num 5 UID 92D94856 VID S2D94856 VID Load SavedUID
A_Increment A_Decrement A_Restore A_Transfer	Ŧ	A_Authentication Cmd Response Response> 01 - Resp OK

Figure 52 A\_Authentication commands

#### 6.1.3.3 A\_ReadBlock

The **A\_ReadBlock** command GUI, shown Figure 53, is to read data from target block address. Therefore, the target block address to be read is a required parameter. User should read data with in authenticated sector. Else, the card will response error and accessing process must be restart from **A\_Request** command. This operation shall be performed after successful authentication.

Software automatically reports characteristic of received data whether it is data block, value block or Sector Trailer Block. Reading result from plain data block, value block and sector trailer block are shown in Figure 53A, Figure 53B and Figure 53C respectively.



C) Sector Trailer block

D) Sector Trailer Block with control block unchecked

Figure 53 A\_ReadBlock commands



#### 6.1.3.4 A\_WriteBlock

The **A\_WriteBlock** command GUI, shown Figure 54, is to write data to target block address. Parameters as follows shall be specified.

Block Num : The target block address to be written

Block Data : 16-byte data coded in hexadecimal format

This operation shall be performed after successful authentication.



Figure 54 A\_WriteBlock commands

#### 6.1.3.5 A\_WriteValueBlock

The **A\_WriteValueBlock** command GUI, shown Figure 55, is to write value to target block address. Parameters as follows shall be specified.

Address value : a 1-byte data storing in value block for indicating address of back up block

Value

: a 4-byte data coded in 2's complement format

In this demonstration software, user can either input amount of value in hexadecimal or decimal number. To input value in hexadecimal, use box "Data Value (MSB)". To input value in decimal, input data in box "Data Value (Dec)". For hexadecimal, value of 0x00000000 to 0x7FFFFFFF is positive number from 0 to 2,147,483,647 while value of 0x80000000 to 0xFFFFFFFF is negative number from -2,147,483,648 to -1. For decimal number, data shall be within -2,147,483,648 to 2,147,483,647.

Block Num : The target block address to be written.

This operation shall be performed after successful authentication.

Note that value and address value is redundantly stored in 16-byte blocks as depicted in Figure 56.



Figure 55 A\_WriteValueBlock commands



# RD-220 Demonstration Software Manual

Value Blocks																
Byte #	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Description		Va	lue			_Va	alue			Va	lue		Adr	_Adr	Adr	_Adr
Note _ <b>Value</b> is inversion of <b>Value</b> Note _ <b>Adr</b> is inversion of <b>Adr</b>																

Figure 56 Structure of value block

#### 6.1.3.6 A\_WriteSectorTrailerBlock

The sector trailer located in last block of each sector contains secret key A and key B, which return logical "0" when read, and the access condition for the four blocks of each sector. The structure of the sector trailer block and access bit organization are shown in Figure 57. The accessibility of each block is controlled by three control bits namely C1x, C2x and C3x. The access condition is redundantly stored with its inversion to secure data protection. There are different meanings about accessibility for the sector trailer block and data blocks as shown in Figure 58 and Figure 62. Note that byte 9 in the sector trailer block is unused area.

	Sector Trailer Blocks															
Byte #	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	Кеу А					Acce	ss Bits		Kev B							

Access Bits											
Bit #	7	6	5	4	3	2	1	0			
Byte 6	_C23	_C22	_C21	_C20	_C13	_C12	_C11	_C10			
Byte 7	C13	C12	C11	<b>C10</b>	_C33	_C32	_C31	_C30			
Byte 8	C33	C32	C31	C30	C23	C22	C21	C20			
Byte 9											

Access bit C1x C2x C3x Description Block0 **C20 C30 C10** Data block 0 in the sector Block1 C11 C21 **C31** Data block 1 in the sector Block2 C12 C22 C32 Data block 2 in the sector **C13** C23 Data block 3 in the sector Block3 **C33** 

Note  $\_\mathbf{Cxx}$  is inversion of  $\mathbf{Cxx}$ 

Figure 57 Structure of the sector trailer block

Acc	ess bi	ts	Access cond	Application			
C1	C2	C3	read	write	increment	decrement, transfer, restore	
0	0	0	key A B <mark>[1]</mark>	key A B1	key A B1	key A B1	transport configuration
0	1	0	key A B <mark>[1]</mark>	never	never	never	read/write block
1	0	0	key A B <mark>[1]</mark>	key B <sup>1</sup>	never	never	read/write block
1	1	0	key A B <mark>[1]</mark>	key B <sup>1</sup>	key B <sup>1</sup>	key A∣B¹	value block
0	0	1	key A B <mark>[1]</mark>	never	never	key A∣B¹	value block
0	1	1	key B <mark>[1]</mark>	key B <sup>1</sup>	never	never	read/write block
1	0	1	key B <mark>[1]</mark>	never	never	never	read/write block
1	1	1	never	never	never	never	read/write block
Acce	ess bit	s	Access condit	ion for			Application

[1] if Key B may be read in the corresponding Sector Trailer it cannot serve for authentication (all grey marked lines in previous table). Consequences: If the reader tries to authenticate any block of a sector with key B using grey marked access conditions, the card will refuse any subsequent memory access after authentication.

Figure 58 Access conditions for data blocks
Access bits		Access condition for				Remark			
			KEYA		Access	s bits	KEYB		
C1	C2	C3	read	write	read	write	read	write	
0	0	0	never	key A	key A	never	key A	key A	Key B may be read
0	1	0	never	never	key A	never	key A	never	Key B may be read
1	0	0	never	key B	key A B	never	never	key B	
1	1	0	never	never	key A B	never	never	never	
0	0	1	never	key A	key A	key A	key A	key A	Key B may be read, transport configuration
0	1	1	never	key B	key A B	key B	never	key B	
1	0	1	never	never	key A B	key B	never	never	
1	1	1	never	never	key AIB	never	never	never	

**Remark:** the grey marked lines are access conditions where key B is readable and may be used for data.

Figure 59 Access conditions for the sector trailer

The GUI for A\_WriteSectorTrailerBlock is shown in Figure 60. Parameters as follows shall are required.

Key A : A 6-bytes secret key

Key B : A 6-bytes secret key

Access Condition : The access condition for Sector Trailer, Block0, Block1 and Block2 the number of 0 to 7 is selectable which a decimal result of bit "C1x-C2x-C3x"

Block Num : Target block to be written

To ease beginner to understand the meaning of each access condition mode, the demonstration software summarizes accessibility of read and write of each block. Also, the check block "Original" can be optionally unchecked to display compatibility of each command.

Before write the sector trailer, user must be aware the effect of written configuration because some configuration may be irreversible. Hence, the card is permanently locked in such mode.

		· · ·	
List Cmd	A_WriteSectorTrailerBlock Command	Original is not checked 🔲 Original	Original is checked 🛛 📝 Original
A_Deselect	Key A FFFFFFFFFF		
A_Usedect 	Key A       FFFFFFFFFF         Key B       FFFFFFFFFF         Access Condition       FFFFFFFFF         Access Condition       For DataBlock 0         For DataBlock 0       I         For DataBlock 1       I         For DataBlock 1       I         For DataBlock 1       I         For DataBlock 2       I         Body Interview       I         Ving DataBlock 0, when 7       I         Vot write       Stress the stress of the stress o	Using Access Condition Mode 7 Use Main Key B and Auxiliary Key A Lock Access Condition Lock both KEYSEC A & B Not Readable KEYSEC B Mode 7 Using Pitableck 0, When Authen Key A Not Writable Not Increasable Not Decrease. Transfer. Restore able I Using DataBlock 0, When Authen Key B Not Writable Not Decrease. Transfer. Restore able Not Decrease. Transfer. Restore able Not Decrease. Transfer. Restore able Not Decrease. Transfer. Restore able Using DataBlock 1, When Authen Key A Writable Decrease. Transfer. Restore able Using DataBlock 1, When Authen Key B Writable Readable Increasable Decrease. Transfer. Restore able Using DataBlock 2, When Authen Key A Writable Readable Increasable Decrease. Transfer. Restore able Using DataBlock 2, When Authen Key B Writable Readable Increasable Decrease. Transfer. Restore able Using DataBlock 2, When Authen Key B Writable Readable Increasable Decrease. Transfer. Restore able Using DataBlock 2, When Authen Key B Writable Readable Increasable Decrease. Transfer. Restore able Using DataBlock 2, When Authen Key B Writable Readable Increasable Decrease. Transfer. Restore able	Using Access Condition Mode 0 Write : Can not access Read : Use key A Using KEYSEC & y A H Read : Can not access or Force zero Using KEYSEC B Write : Use key A Read : Can not access or Force zero Using DataBlock 0 Mode 7 Write : Can not access Increase : Can not access Decrease, Iransfer, Restore : Can not access Decrease, Iransfer, Restore : Can not access Using DataBlock 1 Mode 0 Write :: Use key B , A Read :: Use key B , A Increase : Use key B , A Decrease Iransfer Restore : Use key B , A Read :: Use key B , A Increase : Use key B , A Read :: Use key B , A Increase : Use key B , A Read :: Use key B , A Decrease : Use key B , A Read :: Use key B , A Read :: Use key B , A Read :: Use key B , A Noter :: Use key B , A Read :: Use key B , A Noter :: Use key B , A Increase :: Use key B , A Noter :: Use key B , A Noter :: Use key B , A Noter :: Use key B , A Decrease ::

Figure 60 A\_WriteSectorTrailerBlock commands

Note that the demonstration software formats sector trailer from data in GUI and sent to reader by normal write command.

### 6.1.3.7 A\_Increment

The **A\_Increment** command GUI is shown Figure 61A. Incremental value shall be specified either in hexadecimal or decimal value. Figure 61B shows example of incremental result of 200 from 100 as in Figure 55. Increment which will result in amount of final value beyond 0x7FFFFFF (2,147,483,647) is inhibited and cause error in operation. Note that result from increment is stored in card buffer. User must store to target address by using command **A\_Transfer**. This operation shall be performed after successful authentication.

List Cmd	A_Increment Command	List Cmd	A ReadBlock Command
Standard Command     A_Request     A_WakeUp     A_MaiColl     A_Select     A_Halt     A_RATS     A_PPS     A_Deselect    Mifare Command     A_LoadKey     A_Authentication     A_ReadBlock     A_WriteBlock     A_WriteBlock     A_WriteBlock     A_WriteSectorTrailerBlock     A_Decrement     A_Decrement     A_Reatore     A_Transfer     T	A_Increment Cmd Block Num 6 Increment value (MSB) Increment value (Dec) 200 A_Increment Cmd Response Response> 01 - Resp OK	A_Request A_Request A_WateUp A_AttiColl A_AttiColl A_Hat A_RATS A_PPS A_Deselect A_Hat A_RATS A_PPS A_Deselect A_LoadKey A_Authentication A_WriteValueBlock A_WriteValueBlock A_WriteValueBlock A_UmteSectorTrailerBlock A_Increment A_Decrement A_Restore A_Transfer	A_ReadBlock Collimand A_ReadBlock Cmd Block Num 6 A_ReadBlock Cmd Response Ø Original Detail for Sector Trailer Response> 01 - Resp 0K BlockData> 2C01000003FEFFFF2C01 Value Block Address - 06 DatavalueMs - 0000012C DatavalueDec = 300
		- \ .	

A) A\_Increment command GUI Figure 61 A\_Increment commands

B) Incremental result

# 6.1.3.8 A Decrement

The **A Decrement** command GUI is shown Figure 62A. Decreasing value shall be specified either in hexadecimal or decimal value. Figure 62B shows example of decrement of 100 from 300 as in Figure 61. Decrement which will result in amount of final value below 0x80000000 (-2,147,483,648) is inhibited and cause error in operation. Note that result from increment is stored in card buffer. User must store to target address by using command **A Transfer**. This operation shall be performed after successful authentication.





# 6.1.3.9 A\_Restore

The **A\_Restore** command GUI is shown Figure 63. Parameter for this command is block address in which content will be loaded into card buffer. This operation shall be performed after successful authentication.

# 6.1.3.10 A\_Transfer

The **A\_Transfer** command GUI is shown Figure 64. Parameter for this command is target block address where active content in card buffer will be written. Once content in card buffer is used, the content can not used again. Transferring inactive content causes error response. Figure 64 shows data transfer to block 5. This operation shall be performed after successful authentication.

List Cmd	A_Restore Command A_Restore Cmd Block Num 6
A_Increment A_Decrement A_Restore A_Transfer	A_Restore Cmd Response Response>01 - Resp OK

Figure 63 A\_Restore commands



A) A\_Transfer command GUI

B) Transfer result



# 6.1.3.11 A\_Req\_Anti\_LoadKey\_Authent

The **A\_Req\_Anti\_LoadKey\_Authent** is a combo command performing ISO14443A from card power-on to authentication as listed below.

- 1. A\_Request or A\_Wake\_up
- 2. A\_Anticoll
- 3. A\_Select
- 4. A\_Loadkey
- 5. A\_Authent

This command requires parameters as follows before operating.

Req Mode	: Select Request or Wakeup command used to probe card in field
CollMaskVal	: Select CollMaskVal to select card as explain in A_Anticoll command
Select Key	: Select Key A or Key B on card used in authentication
Кеу	: 48-bit key
Block Num	: The block address to be accessed

If operation is successful, reader will return UID and card will be ready to use MIFARE commands such as write, read, restore and transfer. Figure 65A shows successful authentication while Figure 65B shows error from authentication. Notice that error code in Figure 65B is "05". This reflects to state that error occurs. The error state is authentication.

# RD-220 Demonstration Software Manual

List Cmd	A_Req_Anti_LoadKey_Authent Command	List Cmd	A_Req_Anti_LoadKey_Authent Command
A_LoadKey A_Authentication A_AreadBlock A_WriteSlock A_WriteSlock A_WriteSlock A_NorteSectorTrailerBlock A_Increment A_Decrement A_Restore A_TransferMfare Combo Command A_Req_Anti_LoadKey_Authent A_Req_Anti_LoadKey_Authent A_Increment Transfer A_Increme	A_Req_Anti_LoadKey_Authent Cmd Req Mode Request Cmd  CollMaskVal  Select Key Key  FFFFFFFFFF Block Num 6		A_Req_Anti_LoadKey_Authent Cmd Req Mode Request Cmd ▼ CollMaskVal 0 ▼ Select Key Key A ▼ Key 00000000000 Block Num 6
A Restore_TransferSpecial Command A_Transparent WithCRC A_Transparent WithoutCRC	A_Req_Anti_LoadKey_Authent Cmd Response Vare UID Response>01 - Resp OK UID> 92D94856	A_Restore_Transfer	A_Req_Anti_LoadKey_Authent Cmd Response Save UID Response> A1 - Resp Err Err Data> 05
A) Authent	cication with correct key	B) Authent	ication with wrong key

Figure 65 A\_Req\_Anti\_LoadKey\_Authent commands

# 6.1.3.12 A\_Req\_Anti\_LoadKey\_Authent\_Read

The **A\_Req\_Anti\_LoadKey\_Authent\_Read** is a combo command performing ISO14443A command from card power-on to read block as listed below.

- 1. A\_Request or A\_Wake\_up
- 2. A\_Anticoll
- 3. A\_Select
- 4. A\_Loadkey
- 5. A\_Authent
- 6. A\_ReadBlock

This command requires parameters as follows before operating.

Req Mode	: Select Request or Wakeup command used to probe card in field
CollMaskVal	: Select CollMaskVal to select card as explain in $\ensuremath{\textbf{A}_{Anticoll}}$ command
Select Key	: Select Key A or Key B on card used in authentication
Кеу	: 48-bit key

Block Num : The block address to be read

Figure 66 shows reading result from block 6 from this combo command.

List Cmd	A_Req_Anti_LoadKey_Authent_Read
A_LoadKey A_Authentication A_ReadBlock A_WriteBlock A_WriteBlock A_WriteBlock A_UriteBlock A_Increment A_Decrement A_Decrement A_Reatore A_TransferMfare Combo Command A_Rea_Anti_LoadKey_Authent A_Rea_Anti_LoadKey_Authent A_Increment_Transfer A_Increment_I	A_Req_Anti_LoadKey_Authent_Read Cmd Req Mode Request Cmd ▼ CollMaskVal 0 ▼ Select Key Key A ▼ Key FFFFFFFFF Block Num 6
A_Restore_TransferSpecial Command A_TransparentWithORC A_TransparentWithoutCRC *	A_Req_Anti_LoadKey_Authent_Read Cmd Response ♥ Original Detail for Sector Trailer Response> 01 - Resp 0K UID> 92094856 BlockData> C800000037FFFFFC800 Value Block Address - 06 DataValueMeb - 000000C8
	DataValueDec = 200
gure 66 A_Req_Anti_Loa	dKey_Authent_Read commands

Page 40 of 79



### 6.1.3.13 A\_Req\_Anti\_LoadKey\_Authent\_Write

The **A\_Req\_Anti\_LoadKey\_Authent\_Write** is a combo command performing ISO14443A command from card power-on to write block as listed below.

- 1. A\_Request or A\_Wake\_up
- 2. A\_Anticoll
- 3. A\_Select
- 4. A\_Loadkey
- 5. A\_Authent
- 6. A\_WriteBlock

This command requires parameters as follows before operating.

Req Mode	: Select Request or Wakeup command used to probe card in field
CollMaskVal	: Select CollMaskVal to select card as explain in A_Anticoll command
Select Key	: Select Key A or Key B on card used in authentication
Кеу	: 48-bit key
Block Num	: The block address to be written

Figure 67 shows writing result into block 6 from this combo command.

List Cmd	A_Req_Anti_LoadKey_Authent_Write
List Cmd  A_LoadKey  A_Authentication  A_ReadBlock  A_WriteBlock  A_WriteValueBlock  A_WriteSectorTrailerBlock  A_Increment  A_Decrement  A Restore	A_Req_Anti_LoadKey_Authent_Write A_Req_Anti_LoadKey_Authent_Write Cmd Req Mode Request Cmd  CollMaskVal 0 Select Key Key FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
A_Transfer A_TransferMifare Combo Command A_Req_Anti_LoadKey_Authent A_Req_Anti_LoadKey_Authent A_Req_Anti_LoadKey_Authent A_Increment_Transfer	Block Num 6 Block Data 00000000000000000000000000000000000
A_Decrement_Iransfer A_Restore_TransferSpecial Command A_TransparentWithCRC A_TransparentWithoutCRC	A_Req_Anti_LoadKey_Authent_Write Cmd Response Save UID Response> 01 - Resp OK UID> 92D94856

Figure 67 A\_Req\_Anti\_LoadKey\_Authent\_Write commands

# 6.1.3.14 A\_Increment\_Transfer

The A\_Increment\_Transfer is a combo command performing MIFARE command as listed below.

# 1. A\_Increment

### 2. A\_Transfer

This command requires parameters as follows before operating.

Address value	: A 1-byte data storing in value block for indicating address of back up block
Block Num	: The block address to perform increment
Value	: A 4-byte data coded in 2's complement format or decimal number between
	-2,147,483,648 and 2,147,483,647.

# RD-220 Demonstration Software Manual

Transfer Block : Target block address where increasing value will be written

This operation shall be performed after successful authentication. Figure 68 shows incremental result in block 6 from 100 by 100. Note that increment which will result in amount of final value beyond 0x7FFFFFFF (2,147,483,647) is inhibited and cause error in operation.



A) Input in **A\_Increment\_Transfer** 

B) Incremental result

# 6.1.3.15 A\_Decrement\_Transfer

The A\_Decrement\_Transfer is a combo command performing MIFARE command as listed below.

Figure 68 A\_Increment\_Transfer commands

#### 1. A\_Increment

### 2. A\_Transfer

This command requires parameters as follows before operating.

Address value	: A 1-byte data storing in value block for indicating address of back up block
Block Num	: The block address to perform decrement
Value	: A 4-byte data coded in 2's complement format or decimal number between
	-2,147,483,648 and 2,147,483,647.
Turnefen Die els	. To work block a debugger of the second second state with the second second

Transfer Block : Target block address where decreasing value will be written

This operation shall be performed after successful authentication. Figure 69 shows decrement result in block 6 from 200 by 3000. Note that decrement which will result in amount of final value below 0x80000000 (-2,147,483,648) is inhibited and cause error in operation.

List Cmd	A_Decrement_Transfer Command	List Cmd	A_ReadBlock Command
A_LoadKey     A_Authentication     A_ReadBlock     A_WriteBlock	A_Decrement_Transfer Cmd Block Num 6 Decrement value (MSB)	<ul> <li>✓ A_LoadKey</li> <li>✓ A_Authentication</li> <li>✓ A_ReadBlock</li> <li>✓ A_WriteBlock</li> </ul>	A_ReadBlock Cmd Block Num 6
A_WriteValueBlock A_WriteSectorTrailerBlock A_Increment	Decrement value (Dec) 3000	A_WriteValueBlock A_WriteSectorTrailerBlock A_Increment	A_ReadBlock Cmd Response Ø Original Detail for Sector Trailer
A_Decrement A_Restore A_Transfer	TransferBlock <sup>6</sup>	A_Decrement A_Restore A_Transfer	Response> 01 - Resp OK BlockData> 10F5FFFFEF0A000010F5 Value Block Address - 06 DatavalueMb - EFFFF510
A_Req_Anti_LoadKey_Authent A_Req_Anti_LoadKey_Authent A_Req_Anti_LoadKey_Authent		A_Req_Anti_LoadKey_Authent	DatavalueDec = -2800
A_Rec_Arin_Coditivey_Adition. A_Increment_Transfer		A_nerement_Transfer	
A_restore_inansier	A_Decrement_Transfer Cmd Response Response> 01 - Resp OK	A_TransparentWithoutCRC	
A) Input in <b>A_D</b>	ecrement_Transfer	B) Di	ecrement result

Figure 69 A\_Decrement\_Transfer commands



## 6.1.3.16 A\_Restore\_Transfer

The A\_Decrement\_Transfer is a combo command performing MIFARE command as listed below.

### 1. A\_Restore

## 2. A\_Transfer

This command requires parameters as follows before operating.

Block Num : The block address in which content will be loaded into card buffer

Transfer Block : Target block address where active content in card buffer will be written

This operation shall be performed after successful authentication. Figure 70 shows decrement result in transferring data from block 6 to block 5.

List Cmd  A_LoadKey  A_Authentication  A_ReadBlock  A_WiteBlock  A_WiteBlock  A_WiteSlock  A_Uncrement  A_Decrement  A_Decrement  A_Req_Anti_LoadKey_Authent  A_Req_Anti_LoadKey_Authent  A_Req_Anti_LoadKey_Authent  A_Req_Anti_LoadKey_Authent  A_Recment_Transfer  A_Decrement_Transfer  A_TransparentWthCRC	A_Restore_Transfer Command A_Restore_Transfer Cmd Block Num 6 TransferBlock 5 A_Restore_Transfer Cmd Response Response> 01 - Resp OK	List Cmd	A_ReadBlock Command A_ReadBlock Cmd Block Num 5 A_ReadBlock Cmd Response ♥ Original Detail for Sector Trailer Response> 01 - Resp 0K BlockData> 10F5FFFEF0A000010F5 Value Block Address - 06 DataValueMsb - FFFF510 DataValueDec = -2800
A_ITansparentWithoutCRC  A_TransparentWithoutCRC A) Input in A Re	estore Transfer	A_TransparentWithCRC	fer result

Figure 70 A\_Restore\_Transfer commands

### 6.1.3.17 A\_Req\_Anti\_Select

The A\_Req\_Anti\_Select is a combo command used in getting UID and selecting ISO14443A card from card power-on to authentication as listed below.

- 1. A\_Request or A\_Wake\_up
- 2. A\_Anticoll
- 3. A\_Select

This command requires parameters as follows before operating.

- Req Mode : Select Request or Wakeup command used to probe card in field
- CollMaskVal : Select CollMaskVal to select card as explain in A\_Anticoll command

Figure 71 A\_Restore\_Transfer commands



# 6.2 ISO14443B

The ISO14443B commands list of Pi-931 is shown Figure 72.

Start Send Multi Cmd		Send	
Config Re	ader Speed TX,RX for Reader TX 106 KBits/s	▼ RX 106 KBits/s ▼ SET	
List Cmd	Enable Once Check tandard Command quest akeUp TRIB It iselect vecial Command ansparent WithORC ansparent WithoutCRC	ISO14443B Standard Command None Command	

Figure 72 ISO14443B command list

# 6.2.1 ISO14443B standard commands

Figure 73 depicts the relation of the commands with card state transition diagram. Each command shall be applied at an appropriate state. The commands in this section are **B\_Request**, **B\_WakeUp**, **B\_ATTRIB**, **B\_Halt** and **B\_Deselect**, **B\_TransparentWithCRC** and **B\_TransparentWithoutCRC**. For more information about the commands, please refer to ISO14443-3 standards and Pi-931 Protocol.



Figure 73 ISO14443B standard commands available in Pi-931

# 6.2.1.1 B\_Request

The GUI of the **B\_Request** is shown Figure 74. The AFI value and the number of slot are required parameters. The **B\_Request** command of Pi-931 not only transmits the REQB command following ISO14443B but also performs anti-collision if the number of slot more than 1 is specify. The first card of which PUPI is completely received in anti-collision slot is reported. Hence, the slot Marker is already embedded; that correspond command

for Slot Marker is not provided. However, user can transmits by the slot market via transparent command. The responses from this command, called ATQB, are PUPI, application data and protocol information. The demonstration software decomposes the responses to show card properties. A Pseudo-Unique PICC Identifier (PUPI) is used to differentiate cards during anti-collision. User can check "Save PUPI" check block to collect in software buffer to use in further operation. The Application data field is used to inform the PCD which applications are currently installed in the PICC. The Protocol Info field indicates the parameters supported by the card. For more information, please refer to ISO14443-3.



Figure 74 B\_Request Command

# 6.2.1.2 B\_WakeUp

The operation of **B\_WakeUp** is similar to **B\_Request**. Different from **B\_Request**, **B\_WakeUp** can also wake up card in Halt state. The **B\_WakeUp** GUI is shown in Figure 75.

List Cmd  B_Request  B_Request  B_Halt  B_Deselect  Special Command B_TransparentWthCRC  B_TransparentWthoutCRC	B_WakeUp Command B_WakeUp Cmd AFI Value Num of Slots 1 →
*	B_WakeUp Cmd Response Save PUPI Response > 01 - Resp OK ATOB Response PUPI> BAA185D8 App Data> 11223311 AFI = 11 CRC_B = 2233 NumOfApp = 11 Protocalinfo> F781A3 BitRateCap = F7 Tag Must Use same Bit Rate both Tx ; Tag Tx Bit Rate - 106kbit 212kbit 424 Tag Tx Bit Rate - 106kbit 212kbit 424 Tag Tx Bit Rate - 106kbit 212kbit 424 MaxFirameSize = Size256bytes ProtocalType = Tag Compliant level 4 III

Figure 75 B\_WakeUp Command

# 6.2.1.3 B\_ATTRIB

This command is used to select operating protocol and parameter to use in further operation of smartcard. This command requires Param1, Param2, Param3 and Param4 as stated in ISO14443B. This command shall be activated next to **B\_Request** or **B\_WakeUp** command. After executing this command, user shall set the transaction speed match to what card response. For more information about meaning of the parameters, please refer to ISO14443-3 and -4. The GUI of **B\_ATTRIB** as shown in Figure 76 aids user in composing Param1, Param2, Param3 and Param4.

List Cmd	B_ATTRIB Command B_ATTRIB Cmd PUPI 8A5E085E Param1 ♥ Load SavedPUPI TR0 (Guard Time) Default Va ♥ TR1 (Synchronization Time) Default Va ♥ Disable EOF Disable SOF Param2 Speed TX/RX for Tag TX 106 KBits/♥ RX 106 KBits/♥ Max Frame Size 16 ♥ Param3 ♥ PICC compliant with ISO/IEC 14443-4 Param4 CID 0 ♥
	Direct         Param1 Param2 Param3 Param4           Input         00         00         01         00           Higher layer         Image: Compare the second

Figure 76 **B\_ATTRIB** Command

# 6.2.1.4 B\_Halt

The **B\_Halt** command is used to put card in Halt state. The GUI of **B\_Halt** is shown in Figure 77. The PUPI is a required parameter. This command shall be applied after receiving ATQB from **B\_Request** or **B\_wakeup**.

List Cmd	B_Hoth Cond	HaltCommand
B_Request B_WakeUp B_ATTRIB	PUPI	A8CA3163
B_Hat     B_Deselect    Special Command     B_TransparentWithCRC     B_TransparentWithoutCRC		
Ŧ	B_Halt Cmd Res Response> 01 -	ponse Resp OK

Figure 77 **B\_Halt** Command

## 6.2.1.5 B\_Deselect

The **B\_Deselect** is used to exit active state. This command requires no input and, actually, this command is made up of **B\_TransparentWithCRC.** The GUI of **B\_Deselect** is shown in Figure 77



Figure 78 B\_Deselect Command

# 6.2.1.6 B\_TransparentWithCRC

If commands to be transmitted are not provided in command list such as command for smart card or slot marker, user can input raw hexadecimal code in **B\_TransparentWithCRC** to invoke operation as stated in the RF protocol. Before executing, user might need to adjust timeout for card response to be suitable for user applications. The timeout setting option in the GUI is shown in Figure 80.

The example of "initiate" command in SRI4K card is shown in Figure 79. User shall review results in transaction logs windows. For example, subsequent operations from Figure 79 to read data are shown in Figure 81. For more information about SRI4K, please refer to the datasheet from ST.

List Cmd	B_TransparentWithCRC Command B_TransparentWithCRC Cmd Timeout TxRF		
	0600		
	P. Tarana With CDC Card Damage		
•	B_ Iransparent With CHC Cmg Hesponse		

Figure 79 Example of using **B\_TransparentWithCRC** with SRI4K card

List Cmd	B_Transpa	arentWithCRC Command
B_Request B_WakeUp B_ATTRIB B_Hait B_DeselectSpecial Command B_TransparentWithCRC B_TransparentWithoutCRC	Timeout TxRF	PreviousSetting       PreviousSetting       1 ms       2 ms       4 ms       8 ms       16 ms       32 ms       64 ms       128 ms       256 ms       512ms       1 s
Ţ	-B_Transparent\	WithCRC Cmd Response

Figure 80 Timeout setting for TransparentWithCRC and TransparentWithoutCRC command

Tx00_F3>	OBCO 05 <u>06 00</u> ◀── Initiate()
Rx00_F3>	OBCO 01 <u>E8</u> ◀── ID
Complete	Communicate!
Tx00_F4>	OBCO O5 <u>OE E8</u> <b>→ Select()</b>
Rx00_F4>	OBCO 01 E8
Complete	Communicate!
Tx00_F5>	OBCO 05 <u>08 00</u> <b>▲ Read(0)</b>
Rx00_F5>	OBCO 01 <u>00 00 00 00</u> <b>▲ Data Block 0</b>
Complete	Communicate!
Tx00_F6>	OBCO 05 <u>O8 O1</u> ← Read(1)
Rx00_F6>	OBCO 01 <u>FF FF FF FF</u> ← Data Block 1
Complete	Communicate!

Figure 81 Example of running multiple ISO14443B command

# 6.2.1.7 B\_TransparentWithoutCRC

The operation of this command is similar **to B\_TransparentWithCRC**. The different is there is no calculated CRC appending at the end of transmitted RF frame.

# RD-220 Demonstration Software Manual

## 6.3 ISO15693

The GUI in ISO15693 tap is shown in Figure 82 in which user can select RF transaction speed and operating command. The command list consists of ISO15693 standard commands, Custom command for SIC RFID card IC and special command as is shown in Figure 83. The RF transaction speeds for downlink and uplink are summarized in Table 6-1.

Reader Configulation ISO15693 ISO14443/	A ISO14443B	
Start Send Multi Cmd	Send	
Config Reader Speed TX,RX for Reader + Speed TX 1 out of 256 (1.65 kbits/s TX		Speed TX,RX for Reader + Speed F           TX         1 out of 256 (1.65 kbits/s ▼           TX         1 out of 256 (1.65 kbits/s )           1 out of 4         (26.48 kbits/s )
Enable Once Check List Cmd Inventory15lot Inventory16Slot StayQuiet ReadSingleBlock WriteSingleBlock LockBlock EnadMutikaBlastice	ISO15693 Standard Command Inventory 1 slot Command Inventory 1slot Cmd Inventory Mode RFU Rag Option Rag Protocal Extension Rag	
Virte Multiple Blocks     Select     Reset To Ready     Write AFI     Lock AFI     Write DSFID     Lock DSFID     Get SystemInformation	Mask Len Mask Value	RX 1 Sub Low rate (6.62 kbit ▼ RX 1 Sub Low rate (6.62 kbits/s) 1 Sub High rate (26.48 kbits/s) 1 Sub Ultra high rate (52.96 k 2 Sub Low rate(6.67 kbits/s) I \$2 Sub High rate (26.69 kbits/s)
Get Multiple Block Security Status GCustom SIC Command SetEAS ResetEAS	Inventory1slot Cmd Response	

Figure 82 GUI in ISO15693 Tap

Table 6-1 Supported RF transmission speed				
Transaction	Supported RF Coding	RF Speed	Description	
Downlink	1 out of 4	26.48 Kbits/s	1 out of 4 coding	
	1 out of 256	1.65 Kbits/s	1 out of 256 coding	
Uplink	1 Sub Low rate	6.62 Kbits/s	One subcarrier at Low data rate	
	1 Sub High rate	26.48 Kbits/s	One subcarrier at High data rate	
	1 Sub Ultrahigh rate	52.96 Kbits/s	One subcarrier at Ultrahigh data rate	
	2 Sub Low Rate	6.67 Kbits/s	Two subcarrier at Low data rate	
	2 Sub High Rate	26.69 Kbits/s	Two subcarrier at High data rate	

The ISO15693 command in Pi-931 requires operating mode supplied in command packet before executing. Command mode in operation can be primarily divided into two groups namely inventory mode and non-inventory mode. Inventory mode is for accessing the UID while non-inventory is for any normal card operations. The non-inventory mode can be subdivided into three modes namely non-address mode, address mode and select mode.

Figure 84 illustrates command modes and related operable command. Similar to other RFID standards, card itself do occupy a specific state at certain time. Transition state diagram in ISO15693 is shown in Figure 85. In each state, card reacts to transmitted command differently. Table 6-2 summarizes operable mode between command mode and card state.

# RD-220 Demonstration Software Manual



Table 6-2 operable mo	Table 6-2         operable mode between command mode and card state					
Command	Inventory Mode		Non-Inventory Mode			
Mode		Address Mode	Non-Address Mode	Selected Mode		
Card State		( with UID )	( without UID)			
Ready State	Yes	Yes	Yes	No		
Selected State	Yes	Yes	Yes	Yes		
Quiet State	No	Yes	No	No		

In inventory mode, there are four control flags as shown in Figure 86. The RFU Flag, Option Flag and Protocol Extension Flag can be optionally checked to perform special functions. The functions of these flags are depended on purpose of card IC manufacturer. For Enable AFI flag, if this flag is checked, AFI value must be supplied. The benefit of the AFI (Application family identifier) is to pre-discriminate type of card and reduces the number of card in anti-collision process.





The non-inventory mode is for general operations. In this mode, UID is not required. Except quiet state, all cards in field will be affected. The address mode is for specific-card operation. In this mode, UID is required and the card UID match will only be affected. The select mode is for performing group operation. All cards in selected state

will be affected in this mode. Selection of Non-inventory in the demonstration software is shown in Figure 87. Meaning and functionality of the flags in non-inventory mode is depended on card IC manufacturer.

Note that the inventory mode is mandatory for command "Inventory1Slot" and "Inventory16Slot". Command "Select" and "Stay Quiet" must only be used in address mode. Then, UID is mandatory for these commands. The GUI in demonstration software guides applicable modes for each command.

### 6.3.1 ISO15693 Standard Command



Figure 88 Standard Commands in ISO15693

Table 6-3 Standard Command for ISO15693						
Command Name	Description	Mandatory Input*				
Inventory 1 slot	Perform Inventory 1-slot command	MaskLen, MaskValue				
Inventory 16 slot <sup>(1)</sup>	Perform Inventory 16-slot command	MaskLen, MaskValue				
Stay Quiet	Perform Stay-Quiet Command	UID				
Read Single Blocks	Read Block	Block Address				
Write Single Blocks	Write Block	Block Address, Data Block				
Lock Block	Lock block	Block Address				
Read Multiple Blocks	Read Multiple Block	Block Address, Num of Block				
Write Multiple Blocks	Write Multiple Blocks	Block Address, Num of Block, Block size, Data Multi Block				
Select	Perform Select Command	UID				
Reset to Ready	Perform Reset-to-Ready Command	-				
Write AFI	Write AFI value	AFI Value				
Lock AFI	Lock AFI value	-				
Write DSFID	Write DSFID value	DSFID Value				
Lock DSFID	Lock DSFID value	-				
Get System Information	Get system information	-				
Get Multiple Block Security status	Get multiple block security status	Block Address, Num of Block				

### 6.3.1.1 Inventory 1slot

Command **Inventory1slot** is used in determining UID. Except from four flags in inventory mode, parameter Mask Len and Mask Value must be supplied. Mask Value is part of UID used in selecting cards in anti-collision process. Only cards in which part of UID matches to Mask Value will response back. Mask Len is a one -byte data specifying length in bit of Mask Value to be used. Mask Value and Mask Len shall be input in hexadecimal number. If these parameters are input with "00" or left blank, all cards will response. UID in response packet starts from least significant byte to most significant byte, which reflects to what transmit from card in chronological order.

List Cmd	Inventory 1 slot Command Inventory 1slot Cnd Inventory Mode RFU Rag Option Rag Protocal Extension Rag Enable AFI Mask Len 04 Mask Value 08 Inventory 1slot Cmd Response	List Cmd	Inventory 1 slot Command Inventory Node Inventory Mode Option Flag Protocal Extension Flag Enable AFI Mask Len 04 Mask Value 09 Inventory 1slot Cmd Response
SetEAS	Response> E0 - Resp Er	ectador Sic connicio	Save UID           Response> 01 - Resp OK           DSFID> 00           UID> 7996230C000104E0
A) Widsk Let	Inventory 1 slot Command Inventory 1 slot Command Inventory 1slot Cmd RFU Rag Option Rag Protocal Extension Rag Enable AFI	B) Wlask Ler	Inventory 1 slot Command Inventory 1 slot Command Inventory Islot Cmd Inventory Mode RFU Flag Option Flag Protocal Extension Flag Enable AFI
Select ResetToReady WriteAFI LockAFI WriteDSFID LockDSFID GetSystemInformation	Mask Len 0C Mask Value 7906	Select ResetToReady WriteAFI LockAFI WriteDSFID LockDSFID GetSystemInformation	Mask Len 00 Mask Value 7906
GetMultipleBlockSecurityStatus Custom SIC Command SetEAS ResetEAS C) Mask Len	Inventory1slot Cmd Response Save UID Response> 01 - Resp OK DSFID> 00 UID> 7996230C000104E0 = 0C, Mask Value = 7906	GetMultipleBlockSecuntyStatus GetMultipleBlockSecuntyStatus ResetEAS The SetEAS The SetE	Inventory1slot Cmd Response Save UID Response>01 - Resp OK DSFID> 00 UID>7996230C000104E0 OD, Mask Value = 7906

Figure 89 Example of Inventory 1 slot

Figure 89 shows example in executing **Inventory1slot** to a single card with various value of Mask Len and Mask value. The UID of card is "79 96 23 0C 00 01 04 E0". Then, UID can be written in chronological order from left to right hand side as follows.

UID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Hex	79	96	23	0C	00	01	04	EO
Transmitted Binary	1001 1110	0110 1001	1100 0100	0011 0000	0000 0000	1000 0000	0010 0000	0000 0111

A) Mask Len = 04, Mask Value = 08

Four bits of "0001" representing "8" is transmitted with this command. This part of UID is not matched the LSB part the card UID. Therefore, there is no response from card.

UID Hex	79
UID in Transmitted binary	<b>1001 1110</b>
Transmitted mask value	<b>0001</b>



# B) Mask Len = 04, Mask Value = 09

Four bits of "0001" representing "8" is transmitted with this command. This part of UID is matched the LSB part the card UID. Therefore, card response a complete UID.

UID Hex	79
UID in Transmitted binary	0001 1110
Transmitted mask value	0001

# C) Mask Len = 0C, Mask Value = 7906

12 bits of "0001 1110 0110" is transmitted with this command. This part of UID is not matched the LSB part the card UID. Therefore, card response a complete UID.

UID Hex	79	96
UID in Transmitted binary	0001 1110	<b>0110</b> 1001
Transmitted mask value	0001 1110	0110

D) Mask Len = 0D, Mask Value = 7906

13 bits of "0001 1110 0110 0" is transmitted with this command. This part of UID is not matched the LSB part the card UID. Therefore, there is no response from card.

UID Hex	79	96
UID in Transmitted binary	0001 1110	<b>0110 1001</b>
Transmitted mask value	0001 1110	<b>0110 0</b>

### 6.3.1.2 Inventory 16 slot

Command **Inventory16slot** is used in anti-collision process and determining UID. This command reduces data collision by spreading response from multiple cards into 16 slots systematically. Card will response if Mask Value with in specific Mask Len matches. This is similar to Mask Len and Mask Value in **Inventory 1slot**. In addition, card answers in slot number coincident to 4-bit number in UID next to mask value. Figure 90 shows example in executing **Inventory16slot** to a single card with various value of Mask Len and Mask value. The UID of card is "79 96 23 0C 00 01 04 E0".

A) Mask Len = 00, Mask Value = 00 (None)

The card responds in  $9^{th}$  slot because the UID next to Mask Value is 9 as shown in red number below. UID = "79 96 23 0C 00 01 04 E0"

B) Mask Len = 08, Mask Value = 79 (None)

The card responds in  $6^{th}$  slot because the UID next to Mask Value is 6 as shown in red number below. UID = "79 96 23 0C 00 01 04 E0"

List Cmd	Inventory 16 slot Command	List Cmd	Inventory 16 slot Command
Standard Command     Inventory1Slot     Inventory1Slot     StayQuiet     ReadSingleBlock     WriteSingleBlock     LockRlock     ReadMultipleBlocks	Inventory 16slot Cmd Inventory Mode RFU Rag Option Rag Protocal Extension Rag Enable AFI	Standard Command     Inventory1Slot     Inventory1Slot     StayQuiet     ReadSingleBlock     WriteSingleBlock     LockBlock     ReadMultipleBlocks	Inventory 16slot Cmd Inventory Mode RFU Rag Option Rag Protocal Extension Rag Enable AFI
Write Multiple Blocks Select Reset To Ready Write AFI Lock AFI Lock DSFID Lock DSFID Get System Information	Mask Len Mask Value	WriteMultipleBlocks Select ResetToReady WriteAFI LockAFI WriteDSFID LockDSFID GetSystemInformation	Mask Len 08 Mask Value 79
GetMultipleBlockSecurityStatus Custom SIC Command SetEAS ResetEAS ▼	Inventory16slot Cmd Response Response>01 - Resp OK 	Get Multiple Block Security Status	Inventory16slot Cmd Response Response> 01 - Resp OK slot 6 SubResp> 01 - SubResp OK DSFID> 00 UID> 7996230C000104E0
A) Mask Len =	00, Mask Value = 00 (None)	B) Mask Ler	n = 04, Mask Value = 09

Figure 90 Example of Inventory 16 slot

The response from this command is a concatenated frame of response in each slot. For more information, please refer to "Pi-931 Module Protocol".

Diagram in Figure 91 shows an idea of anti-collision process by using scanning technique. The collision slot, which is slot 3 and slot E in this example, in the first round, is possibly that multiple cards have responded in the same slot. Hence, the **Mask Value** is set to the number of slot that collision occurred and **Mask Len** is increased by 4. This process is repetitive until there is no collision detected and processing unit collect the UID in slot there is no collision. Practically, error may cause errors in collision slot.



Figure 91 An example of Using Inventory 16 slot in anti-collision

# 6.3.1.3 Stay Quiet

The **Stay Quiet** put UID-specific card in quiet state. Therefore, UID is mandatory parameter for this command.

List Cmd		Stay Quiet Command
Standard Command     Inventory1Slot     Inventory16Slot     StayQuiet	•	StayQuiet Cmd Non Inventory Mode I RFU Flag
ReadSingleBlock		Option Flag
LockBlock	Ξ	Protocal Extension Flag
Read Multiple Blocks		Operation Mode Address Mode
WriteMultipleBlocks		UID 7996230C000104E0
Reset To Ready		
WriteAFI		
WriteDSFID		
LockDSFID		
GetSystemInformation		
Custom SIC Command		StayQuiet Cmd Response
SetEAS		Response> 01 - Resp OK
ResetEAS	Ŧ	

Figure 92 Stay Quiet Command

#### 6.3.1.4 ReadSingleBlocks

The **ReadSingleBlocks** performs data reading from a specified block. Block address is only a require parameter.

List Cmd		Read Single	Blocks Command
Standard Command     Inventory1Slot     Inventory16Slot     StayQuiet		ReadSingleBlock Cm Non Inventory Mode	d e
ReadSingleBlock		Option Flag (Blog)	ock security status )
WriteSingleBlock LockBlock ReadMultipleBlocks Select ResetToReady WriteAFI LockAFI WriteDSFID CockDSFID GetSystemInformation	E	Protocal Extens Operation Mode Block Addess	ion Flag Non Address Mode
GetMultipleBlockSecurityStatusCustom SIC Command SetEAS ResetEAS	•	ReadSingleBlock Cm Response> 01 - Resp Block Data> 555555	d Response o OK 55

Figure 93 ReadSingleBlock command

### 6.3.1.5 WriteSingleBlocks

The **WriteSingleBlocks** performs data programming to specified block. Block address and data to be written are required parameters. The GUI, as shown in Figure 94, provides 4-byte data to be default block length. For some special card in which block data is wider than the default, user can check "Over 4 bytes" box to accept input up to 8 bytes.

List Cmd	Write Single Blocks Command	List Cmd	Write Single Blocks Command
Standard Command     Inventory 1Slot     Inventory 1Slot     StayQuiet     ReadSingleBlock     WriteSingleBlock	WriteSingleBlock Cmd Non Inventory Mode RFU Rag Option Rag	Standard Command     Inventory1Slot     Inventory16Slot     StayQuiet     ReadSingleBlock     WriteSingleBlock	WriteSingleBlock Cmd Non Inventory Mode RFU Flag Option Flag
LockBlock	Operation Mode	LockBlock	Protocal Extension Flag     Operation Mode
WriteMultipleBlocks		WriteMultipleBlocks	Non Address Mode
Reset To Ready	Block Addess 5 June 2010	Reset To Ready	Block Addess 5 Void SavedUID
LockAFI WriteDSFID	Data Block 11223344	LockAFI     WriteDSFID	Data Block 1122334455667788
LockDSFID	Over 4 Bytes	LockDSFID	✓ Over 4 Bytes
GetSystemInformation GetMultipleBlockSecurityStatus Custom SIC Command SetEAS ResetEAS	WriteSingleBlock Crnd Response Response> 01 - Resp OK	GetSystemInformation GetMultipleBlockSecurityStatusCustom SIC Command SetEAS ResetEAS	WriteSingleBlock Cmd Response Response>01 - Resp OK

Figure 94 WriteSingleBlock command

#### 6.3.1.6 Lock Block

The **LockBlock** command performs content lock in a specific block from modification later. The required parameter is a block address.

### 6.3.1.1 ReadMultipleBlocks

The **ReadMultipleBlocks** performs data reading from a specified range of data. Parameters as follows must be specified before executing.

Block Address : The first block address to be written

Num of Blocks : The number of additional block from first block to be written

The demonstration software can divide and display data for each block as shown Figure 96, providing that the block size, parameter in software, be specified correctly. Note that cards from some manufacturers might not support this command.

List Cmd	Lock Block Command  LockBlock Cmd Non Inventory Mode RFU Rag Option Rag Protocal Extension Rag Operation Mode Non Address Mode Block Addess 5 V Load SavedUID
GetSystemInformation GetMultipleBlockSecurityStatus Custom SIC Command SetEAS ResetEAS	LockBlock Cmd Response

Figure 95 Lock Block command

List Cmd	* E	Read Multiple ReadMultipleBlocks Non Inventory Mod RFU Rag Option Flag ( Bl Protocal Extens	e Blocks Command Cmd le lock security status ) sion Flag
		Operation Mode Block Addess Num Of Blocks	Non Address Mode
	Ŧ	ReadMultipleBlocks Block Size Response> 01 - Res	Cmd Response 4 pp OK
		order 0 Block Data> 222222 order 1 Block Data> 333333	333
		order 2 Block Data> 444444 order 3 Block Data> 555555	555

Figure 96 ReadMultipleBlock command

# 6.3.1.2 WriteMultipleBlocks

The **WriteMultipleBlocks** performs data reading from a specified range of data. Parameters as follows must be specified before executing.

Block Address : The first block address to be written

Num of Blocks : The number of additional block from first block to be written

Block Size : The number of bytes in one block

Data Multi Block : Consecutive data for all block to be written

The demonstration software can divide and display data for each block as shown Figure 96, providing that the block size, parameter in software, be specified correctly. Note that cards from most manufacturers might not support this command.

List Cmd	_	Write Multiple	Blocks Command
Standard Command     Inventory1Slot     Inventory16Slot     StavQuiet	^	WriteMultipleBlocks Non Inventory Moo	Cmd Je
ReadSingleBlock		Option Flag	
ViteSingleBlock LockBlock	E	Protocal Extens	sion Flag
ReadMultipleBlocks		Operation Mode	Non Address Mode 🔻
Select     Select     Reset To Ready     Write AFI     Lock AFI     Uotk DSFID     Lock DSFID     Get System Information     Get Multiple Block Security Status    Custom SIC Command     SetEAS     ResetEAS	-	Block Addess Num Of Blocks BlockSize Data Multi Block	0 V Load SavedUll 2 + 1 4 0000000011111111222 22222

Figure 97 WriteMultipleBlocks command

# 6.3.1.3 Select

The Select put UID-specific card in selected state. UID is mandatory parameter for this command.

List Cmd		Select	t Command
Standard Command     Inventory1Slot     Inventory16Slot     StayQuiet	*	Select Cmd Non Inventory Mod RFU Flag	e
ReadSingleBlock		Option Flag	
ViteSingleBlock	Ξ	Protocal Extens	ion Flag
ReadMultipleBlocks		Operation Mode	Address Mode 👻
WriteMultipleBlocks			
Select		UID	7996230C000104E0
Reset To Ready           Write AFI			Load SavedUID
LockAFI			
WriteDSFID			
LockDSFID			
Get System Information			
		Select Cmd Respons	e
SetEAS		Response> 01 - Res	р ОК
ResetEAS	Ŧ		

Figure 98 Select command

# 6.3.1.4 ResetToReady

The **ResetToReady** puts card into ready state. No parameter, except UID in selected mode, is required for this command.

List Cmd		Reset to Ready Command
Standard Command     Invertory1Slot     Invertory1Slot     StayQuiet     ReadSingleBlock     WriteSingleBlock     LockBlock     WriteMultipleBlocks     Select     NeekToReady     WriteAFI     LockAFI     WriteDSFID     LockDSFID     GetSystemInformation	E	Reset To Ready Cmd Non Inventory Mode RFU Rag Option Rag Protocal Extension Rag Operation Mode Non Address Mode  Value Load SavedUID
GetMultipleBlockSecurityStatus		Reset To Ready Cmd Response
SetEAS		Response> UI - Resp OK
ResetEAS	*	

Figure 99 ResetToReady command



## 6.3.1.5 WriteAFI

The WriteAFI performs AFI programming. One bytes data for AFI shall be supplied before executing.



Figure 100 WriteAFI command

# 6.3.1.6 LockAFI

The **LockAFI** performs AFI programming. Once the AFI was locked, AFI content cannot be modified later. No parameter, except UID in selected mode, is required for this command.

List Cmd	Lock AFI Command
Standard Command     Inventory1Slot     Inventory1Slot     StayQuiet     ReadSingleBlock     WriteSingleBlock     LockBlock     WriteMultipleBlocks     WriteMultipleBlocks     WriteMultipleBlocks     WriteMultipleBlocks     WriteMultipleBlocks     WriteMather     IockAFI     WriteDSFID     IockDSFID	LockAFI Cmd     Non Inventory Mode     RFU Rag     Option Rag     Protocal Extension Rag     Operation Mode Non Address Mode      Von Address Mode     Load SavedUID
Get SystemInformation Get MultipleBlockSecurityStatusCustom SIC Command SetEAS ResetEAS	LockAFI Cmd Response Response> 01 - Resp OK

Figure 101 LockAFI command

#### 6.3.1.7 WriteDSFID

The WriteDSFID performs DSFID programming. One bytes data for DSFID shall be supplied before executing.

List Cmd		Write DS	FID Command
Standard Command     Inventory1Slot     Inventory16Slot     StayQuiet	^	WriteDSFID Cmd Non Inventory Mod	le
ReadSingleBlock		Option Flag	
	E	Protocal Extens	sion Flag
ReadMultipleBlocks		Operation Mode	Non Address Mode 🔻
WriteMultipleBlocks			
✓ Select ✓ ResetToReady			
VriteAFI LockAFI		DSFID value	01 V Load SavedUIL
WriteDSFID			
LockDSFID GetSystemInformation			
GetMultipleBlockSecurityStatus		WriteDSFID Cmd Re	esponse
Custom SIC Command		Response > 01 - Res	sp OK
ResetEAS	Ŧ		

Figure 102 WriteDSFID command



# 6.3.1.8 Lock DSFID

The **LockAFI** performs DSFID programming. Once the DSFID was locked, DSFID content cannot be modified later. No parameter, except UID in selected mode, is required for this command.



Figure 103 LockDSFID command

### 6.3.1.9 Get System Information

The GetSystemInformation retrieve card system information. Information as listed are displayed.

UID	: Unique ID
IC reference	: 8-bit IC reference number
VICC Memory Size	: Card memory size information consisting of block size and number of block
AFI	: Application family identifier
DSFID	: Data storage format identifier



Figure 104 GetSystemInformation command

### 6.3.1.10 Get Multiple Block Security status

The **GetMultipleBlockSecurityStatus** retrieve condition if specific blocks are locked. Parameters as follows must be specified before executing.



Block Address : The first block address to be written

Num of Blocks

: The number of additional block from first block to be written



Figure 105 GetMultipleBlockSecuritystatus command

# 6.3.2 Example of ISO15693 standard command usage

Figure 106 shows example of operating standard command. The operations in this example go through ready mode, selected mode and quiet mode. In each mode, inventory command and read block 0 in three non-inventory mode are performed. User can notice that the operating results coincide with the Table 6-2. For more information about standard command, please refer to ISO15693-3 protocol.



Figure 106 Example of ISO15693 standard command usage

# 6.3.3 Custom command for SIC5600

SIC5600 is an ISO15693 card IC from silicon craft. To communicate with SIC5600 requires SIC-defined custom commands. The supported custom command in Pi-931 is shown in Figure 107 and summarized in Table 6-4. For more information about SIC5600, please refer to SIC5600 datasheet.

Table 6-4 Custom Command for SIC5600					
Command Name	Description	Mandatory Input*			
Set EAS <sup>(1)</sup>	Enable EAS mode ( If EAS mode is not locked )	-			
Reset EAS	Disable EAS mode ( If EAS mode is not locked )	-			
Lock EAS <sup>(2)</sup>	Lock current status of EAS mode	-			
EAS Alarm	Invoke SIC5600 to transmit EAS code (Card must be preset in EAS mode)	-			
Kill <sup>(3)</sup>	Destroy a specific-UID card	UID, Kill Code			
Load Password	Login to enable accessing card content	UID, PWD Mode, Random Number, Password			
Write Password	Write password 0, password 1 or kill code	UID, Selected Password, Password			
Lock Password <sup>(2)</sup>	Lock current password or kill code	UID, Selected Password			
Set Password Mode <sup>(4)</sup>	Setup password mode or kill enable	UID, PWD Mode			
Lock Password Mode <sup>(2),(4)</sup>	Lock current status of the password mode	UID			
Get Password Mode <sup>(4)</sup>	Get current status of the password mode	UID			
Set OTP <sup>(5)</sup>	Set OTP mode	UID, OTP Mode			
Get OTP <sup>(5)</sup>	Get current status of the OTP mode	UID			
Write XUID <sup>(6)</sup>	Write XUID to a specific-UID card	UID, XUID			
Read XUID (6)	Write XUID from a specific-UID card	UID			



Figure 107 Custom commands for SIC5600

# RD-220 Demonstration Software Manual

## 6.3.3.1 Set EAS

EAS (Electronic Article Surveillance) is a configurable mode in SIC5600. If EAS mode is set, **EAS Alarm** command can invoke card to transmit EAS code. The command **Set EAS** requires no input parameter. Figure 108A shows successful response from **Set EAS** and Figure 108B show response of **EAS Alarm** after performing **Set EAS**.

List Cmd	Set EAS Command	List Cmd	EAS Alarm Command
Select ResetToReady WriteAFI LockAFI LockAFI GetSystemInformation GetMultipleBlockSecurityStatusCustom SIC Command SetEAS ResetEAS LockEAS LockEAS EAS_Alam Kill LoadPassword WritePassword	SetEAS Cmd Non Inventory Mode RFU Rag Option Rag Protocal Extension Rag Operation Mode Non Address Mode  V Load SavedUID	Select ResetToReady WriteAFI LockAFI UriteDSFID CockAFI GetSystemInformation GetSystemInformation GetMultipleBlockSecurityStatusCustom SIC Command SetEAS ResetEAS EAS_Alarm Kill LoadPassword WritePassword	EAS_Alarm Cmd Non Inventory Mode RFU Flag Option Flag Protocal Extension Flag Operation Mode Non Address Mode  Value Load SavedUID
ChickPassword SetPasswordMode LockPasswordMode GetPasswordMode ▼	SetEAS Cmd Response Response> 01 - Resp OK	CockPassword SetPasswordMode CockPasswordMode GetPasswordMode ▼	EAS_Alam Cmd Response Response> 01 - Resp OK EAS> 2FB36270D5A7907FE8B18038D28149

A) Perform Set EAS

B) EAS Alarm Response after performing Set EAS

Figure 108 Set EAS command

### 6.3.3.2 Reset EAS

If EAS mode is reset, **EAS Alarm** command can not invoke card to transmit EAS code. The command **Reset EAS** requires no input parameter. Figure 109A shows successful response from **Reset EAS** and Figure 109B shows no response of **EAS Alarm** after performing **Set EAS**.



Perform Reset EAS B) EAS Alarm response after performing Reset EAS Figure 109 Reset EAS command

# 6.3.3.3 Lock EAS

Command Lock EAS locks EAS setting configuration permanently. After performing Lock EAS, Set EAS and Reset EAS cannot successfully execute as shown in Figure 110.





Figure 110 Lock EAS command

# 6.3.3.4 EAS Alarm

Command **EAS Alarm** can invoke card to transmit EAS code. As shown in Figure 108B and Figure 109B, EAS value for SIC5600 is "2FB36270D5A7907FE8B18038D281497682DA9A866FAF8BB0F19CD112A57237EF".

# 6.3.3.5 Write Password

Command **Write Password** used to program Password 0, Password 1 and Kill code. UID, password value and target password are required parameters for this command. If password being written was locked, programming will not be successful with return code as shown in Figure 111B.

List Cmd       Write Password Command         Write DSFID          LockDSFID          Get SystemInformation       Get SystemInformation         Get MultipleBlockSecurityStatus         Custom SIC Command       SetEAS         SetEAS          LockEAS          LockEAS          LockPassword          Write Password          Wite Password          Option Flag          Protocal Extension Flag          Operation Mode          LockPassword          Write Password          Write Password          Write Password          DockEAS          LockPassword          Write Password          UID       0D7A580000239E0         Select PWD addr       Password 0         Value          LockPassword          Vite Password          Operation Mode          LockPasswordMode          Self PasswordMode          GetPasswordMode
---

A) Write Password successful

B) Perform Write Password while password was locked

Figure 111 Write Password command

# 6.3.3.6 Lock Password

Command Lock Password, the GUI is shown Figure 112, locks current target password value permanently. UID and target password are required parameters for this command.

# 6.3.3.7 Set Password Mode

Command **Set Password Mode** is for enabling Kill-enable bit and defining password mode. If Kill-enable bit is set, **Kill** command can be disable card. Password mode, consists of two control parameters namely Password Allocation (PA) and Security Mode(SM), defines password to be used and accessibility for read and writes SIC5600 content. Functionality of each password mode, which is combination Password Allocation, and Security Mode is summarized in Table 6-5. Figure 113A shows effect from setting Password 0 for read protection and Password 1 for

write protection. As a result, reading block 0 without loading password is inhibit as shown in Figure 113B. UID, Killenable and password mode are required parameters for this command.



Figure 112 Lock Password command

Select       SetPasswordMode Cmd         ResetToReady       Inventory Mode         WriteAFI       IockAFI         LockAFI       Option Flag         Option Flag       Option Flag         Option Flag       Option Flag         Option Flag       Option Flag         UID       OptionStatus        Custom SIC Command       SetEAS         LockAS       UID         DotkPassword       VriteAFI         WritePassword       VoiteAFI         Wite DSFID       Operation Mode         LockBS       Operation Mode         Wite StripeBlock       VritePassword         UID       0D7A5800000239E0         PWD Mode       LockAS         EAS_Alarm       VritePassword         VoitePassword       VoitePassword         VoitePassword       Postected Read and Wr         SectPasswordMode       Protected Read and Wr         SetPasswordMode       Protected Reagonse	List Cmd	Set Password Mode Command	List Cmd	Read Single Blocks Command
Response>01 - Resp OK	Select Reset To Ready Witte AFI Lock AFI Lock AFI Lock AFI Get System Information Get Multiple Block Security StatusCustom SIC Command Set EAS Reset EAS Lock EAS EAS_Lock EAS EAS_Alam Kill V Load Password Lock Password Lock Password Cock Passwo	SetPasswordMode Cmd Non Inventory Mode RFU Rag Option Rag Protocal Extension Rag Operation Mode UID 0D7A5800000239E0 PWD Mode V Load SavedUID Fassword Allocation Use PWD0 for Rd , PWD1 for Wr Security Mode Protected Read and Wr SetPasswordMode Cmd Response Response>01 - Resp OK	Standard Command     Inventory1Slot     Inventory1Slot     StayQuiet     SeadSingleBlock     WriteSingleBlock     WriteSingleBlocks     WriteMultipleBlocks     WriteMultipleBlocks     Select     ResetToReady     WriteAFI     LockAFI     WriteDSFID     LockDFID     LockDFID     LockDFID     GetSystemInformation     GetMultipleBlockSecurityStatus    Custom SIC Command     SetEAS     ResetEAS     ToreadSiteCommand	ReadSingleBlock Cmd       Non Inventory Mode       RFU Rag       Option Rag (Block security status )       Protocal Extension Rag       Operation Mode       Address Mode       UID       0D7A5800000239E0       Block Addess       0       VLoad SavedUID

A) Set Password Mode (PA = 00, SM = 11)

B) Reading during read protection

Figure 113 Set Password Mode command

Table 6-5         Summarized protection function controlled by PA (Password Allocation ) and SM (Security Mode)						
PA (Password Allocation)		SM (Security Mode)		Protection Function		
00b	Use Password 0 for read	00b	No protection	No protection		
	protection and Password	01b	Read Protection Only	Read Protection by Password 0, M = 0		
	1 for write protection	10b	Write Protection Only	Write Protection by Password 1, M = 1		
		11b	Read and Write Protection	Read Protection by Password 0, M = 0 and Write		
				Protection by Password 1, M = 1		
01b	Use Password 0 for all	00b	No protection	No protection		
	protection	01b	Read Protection Only	Read Protection by Password 0		
		10b	Write Protection Only	Write Protection by Password 0		
		11b	Read and Write Protection	Read & Write Protection by Password 0		
10b	Use Password 1 for all	00b	No protection	No protection		
	protection	01b	Read Protection Only	Read Protection by Password 1		
		10b	Write Protection Only	Write Protection by Password 1		
		11b	Read and Write Protection	Read & Write Protection by Password 1		
11b	Use Both Password 0 and	00b	No protection	No protection		
	Password 1 (64 Bits) for	01b	Read Protection Only	Read Protection by Password 0 + Password 1		
	all protection	10b	Write Protection Only	Write Protection by Password 0 + Password 1		
		11b	Read and Write Protection	Read & Write Protection by Password 0 + Password 1		



#### 6.3.3.8 Lock Password Mode

Command Lock Password Mode, the GUI is shown Figure 114, freezes current password mode value permanently. UID is a required parameter for this command.

ist Cmd		Lock Password Mode Command
WriteDSFID         LockDSFID         GetSystemInformation         GetMultipleBlockSecurityStatus        Custom SIC Command         SetEAS         LockEAS         LockEAS         Kill         LoadPassword         WritePassword         LockPasswordMode         SetPasswordMode         SetOTP         GetOTP         Multiple		LockPasswordMode Cmd Non Inventory Mode RFU Rag Option Rag Protocal Extension Rag Operation Mode Address Mode UID 0D7A5800000239E0 VID LockPasswordMode Cmd Response LockPasswordMode Cmd Response Response>01 - Resp OK
ReadXUID	-	

Figure 114 Lock Password Mode command

#### 6.3.3.9 Get Password Mode

Command **Get Password Mode**, the GUI is shown Figure 115, is used to obtain current password mode. UID is a required parameter for this command. The structure of returned password mode is shown in Table 6-6 to Table 6-10.

ist Cmd WriteDSFID LockDSFID GetSystemInformation GetMultipleBlockSecurityStatus Custom SIC Command SetEAS ResetEAS LockEAS		Get Password Get Password Mode C Non Inventory Mod RFU Rag Option Rag Protocal Extens Operation Mode	d Mode Command ind e ion Rag
EAS_Alarm Kill LoadPassword WritePassword LockPassword SetPasswordMode LockPasswordMode	E	UID	0D7A5800000239E0
SetOTP GetOTP WriteXUID ReadXUID	•	GetPasswordMode C Response> 01 - Res PWDMode> 20	md Response p OK

Figure 115 Get Password Mode command

Table 6-6 Password mode (PWD Mode)								
Bit	7	6	5	4	3	2	1	0
Name	0	0	Kill Enable	М	PA (Password Allocation) SM (Security N		rity Mode)	

Ob         Reset Kill-Enable bit in SIC5600 to protect card from Kill command           1b         Set Kill-Enable bit in SIC5600 to enable Kill command to operate	Table	6-7 Kill Enable : Set up Kill-Enable bit
1b Set Kill-Enable bit in SIC5600 to enable Kill command to operate	0b	Reset Kill-Enable bit in SIC5600 to protect card from Kill command
	1b	Set Kill-Enable bit in SIC5600 to enable Kill command to operate.

Table	6-8 M : Specify password in card used to compare with transmitted password during loading password
0b	Use Password 0
1b	Use Password 1



Table	6-9 PA (Password Allocation) : Define passwords to be used in protection
00b	Use Password 0 for read protection and Password 1 for write protection. If card was set in this mode, input PWD shall
	match Password 0 during loading password with M value of 0 to read content. To write data to card, input PWD shall match
	Password 1 during loading password with M value of 1.
01b	Use Password 0 for protection. If card was set in this mode, input PWD shall match Password 0 during loading password
	with M value of 0 to read/write content.
10b	Use Password 1 for protection. If card was set in this mode, input PWD shall match Password 1 during loading password
	with M value of 1 to read/write content.
11b	Use Both Password 0 and Password 1 (64 Bits) for protection. If card was set in this mode, input 8-bytes PWD matched
	Password 0 and Password 1 must be used for loading password with any value of M to read/write content.

Table 6-10 SM (Security Mode(1:0)): Define protection function of the password		
00b	No protection	
01b	Read Protection Only	
10b	Write Protection Only	
11b	Read and Write Protection	

# 6.3.3.10 Load Password

Command Lock Password, the GUI is shown Figure 116A, is used to login to grant accessibility to read or write. Parameters as follows must be specified before executing.

UID	: Card Unique ID
Access Password	: Card Password (0 or 1) used to compare with transmitted password
Password Allocation	: Password protection function in card being access
Security Mode	: Security mode in card being access
Random Number	: 4-byte or 8-byte random number used to encrypt transmitted data on air
Password	: 4-byte or 8-byte password value matched access password

If Password Allocation is "Use PWD 64 bit (PWD0+PWD1)", 8-byte random number and 8-byte password value matched password 0 consecutive with password 1 shall be supplied. Practically, RNG should be altered every time to prevent playback attack

The example of steps in executing **Lock Password** and its results are shown in Figure 116. Assume that card being accessed was previously programmed with password 0 of 00000000 and password 1 of 00000000, and password mode was set in "Password 0 for read protection and Password 1 for write protection" (from Figure 113).

List Cmd Ust Cmd Inventory 1Slot Inventory 1Slot StayQuiet V ReadSingleBlock UockBlock BeadMultipleBlocks	Read Single Blocks Command ReadSingleBlock Cmd Non Inventory Mode RFU Rag Option Rag (Block security status ) Protocal Extension Rag Operation Mode Non Address Mode	List Cmd Standard Command Inventory 1Slot StayQuiet ReadSingleBlock WhiteSingleBlock ReadMultipleBlocks	Write Single Blocks Command Write Single Block Cmd Non Inventory Mode RFU Rag Option Rag Protocal Extension Rag Operation Mode Non Address Mode V
WriteMultipleBlocks Select WriteAFI LockAFI WriteDSFID LockDSFID	Block Addess 0 V Load SavedUID	WriteMultipleBlocks Select ResetToReady WriteAFI LockAFI WriteDSFID LockDSFID GetSystemInformation	Block Addess 0 V Load SavedUID Data Block 11111111 Over 4 Bytes
GetSystemInformation GetMultipleBlockSecurityStatus Custom SIC Command SetEAS ResetEAS	ReadSingleBlock Cmd Response Response > D0 - Resp Err Err Data > A0	Get Multiple Block Security StatusCustom SIC Command SetEAS ResetEAS *	WriteSingleBlock Cmd Response Response> D0 - Resp Err Err Data> A0

A) Reading data from block 0 is protected

B) Programming data from block 0 is protected

# **RD-220 Demonstration Software Manual**

List Cmd



C) Load password with password 0 to access read Successful



E) Load password with password 1 to access write Successful



G) Confirm that data was programmed into block 0

Figure 116 Load Password command

Read Single Blocks Command --Standard Command--. ReadSingleBlock Cmd Inventory1Slot Non Inventory Mode Inventory16Slot RFU Flag StayQuiet Option Flag (Block security status) WriteSingleBlock Protocal Extension Flag LockBlock Operation Mode Address Mode -ReadMultipleBlocks Write Multiple Blocks UID Select 0D7A5800000239E0 Reset To Ready Block Addess Load SavedUID 0 WriteAFI LockAFI WriteDSEID LockDSFID GetSystemInformation GetMultipleBlockSecurityStatus ReadSingleBlock Cmd Response -Custom SIC Command Response>01 - Resp OK Block Data> 00000000 SetEAS ResetEAS

# D) Successful reading data from block "0"

List	Cmd		Write Single Blocks Command
	Inventory1Slot Inventory1Slot StayQuiet ReadSingleBlock UockBlock ReadMultipleBlocks Select ResetToReady WriteAtFl LockAFl WriteDSFID LockDSFID GetSystemInformation GetMultipleBlockSecurityStatus	· III	WriteSingleBlock Cmd Non Inventory Mode RFU Flag Option Flag Protocal Extension Flag Operation Mode Non Address Mode Block Addess Data Block 11111111 Over 4 Bytes
	Custom SIC Command SetEAS ResetEAS LockEAS	Ŧ	WriteSingleBlock Cmd Response Response>01 - Resp OK

Successful programming data from block "0" F)

# RD-220 Demonstration Software Manual

#### 6.3.3.11 Kill

The Kill command will permanently disable card function. The card has been killed will not response to any request from reader. The Kill GUI is shown in Figure 117E. The UID and Kill code are required for this command. Following steps, depicting in Figure 117, demonstrate preparation for kill and Kill operation.



#### A) Set Kill-Enable by Set Password Mode.



#### C) Check Password Mode if Kill-Enable was set.

# B) Write Kill Code by Write Password Command.

List Cmd		Inventory 1 slot Command
Standard Command     Inventory150t     Inventory150t     StayQuiet     ReadSingleBlock     WriteSingleBlock     LockBlock     ReadMultipleBlocks     Select     ResetToReady     WriteAFI     LockAFI     WriteDSFID     LockDSFID     GetSvstemInformation		Inventory Islot Cmd Inventory Mode RFU Rag Option Rag Protocal Extension Rag Enable AFI Mask Len Mask Value
GetMultipleBlockSecurityStatus GetMultipleBlockSecurityStatus SetEAS		Inventory1slot Cmd Response
ResetEAS	*	Response> 01 - Resp OK DSFID> 00 UID> 0D7A5800000239E0

D) Check if card is still alive by inventory



E) kill card by Kill command

F) No response confirms that card was killed

# Figure 117 Example of Kill operation

# RD-220 Demonstration Software Manual

### 6.3.3.12 Set OTP

OTP is one time programmable feature. If an OTP bit is set, the data in associated section can not modified forever. Note that if the OTP is set to once, it can not reset to zero. In the provided GUI of Set OTP as shown in Figure 118, user can lock target page or lock OTP configuration itself.



C) Perform lock OTP configuration By Set OTP

D) Perform lock Page 1 By **Set OTP** failed because OTP configuration was locked.

Figure 118 Set OTP command

# 6.3.3.13 Get OTP

The **Get OTP** command is used to obtain current status of OTP Mode. As shown in Figure 119, OTP Mode of "00" (hexadecimal) was reported. The structure of OTP Mode is shown in Table 6-11 to Table 6-13.

List Cmd	Get OTP Command
Custom SIC Command     SetEAS     ResetEAS     LockEAS     LockEAS     EAS_Alam     Kill     LoadPassword     WritePassword     LockPassword     SetPasswordMode     GetPasswordMode     GetPasswordMode     SetOTP	GetOTP Cmd Non Inventory Mode RFU Rag Option Rag Protocal Extension Rag Operation Mode Non Address Mode UID 0D7A5800000239E0 V Load SavedUID
V GetOTP WriteXUID ReadXUID Special Command TransparentWithCRC TransparentWithoutCRC SendEOF  v	GetOTP Cmd Response Response> 01 - Resp OK OTPMode> 00

Figure 119 Get OTP command



Table 6-11 OTP Mode											
Bit	7	6	5	4	3	2	1	0			
Name	0	0	0	0	0	L	OTP MO	DE(1:0)			

Table 6-12 L : Lock Control to Lock current status of OTP MODE				
0b	Not Lock OTP			
1b	Lock OTP.			
	If Lock OTP bit was set to one, it can not reset back to zero and OTP MODE value can not be altered later.			
Table 6-13 OTP MODE(1:0)				

00b	Not Lock		
01b	Lock Page 0 (Address 0 - 15)		
10b	Lock Page 1 (Address 16 - 31)		

### 6.3.3.14 Write XUID

XUID is a one-time writable 6-byte extended Unique ID which can be in some specific user purpose. If XUID has been once written, XUID cannot be modified later. User can input required XUID value in the GUI of **Write XUID** as shown in Figure 120. UID is required for this command.



Figure 120 Write XUID command

# 6.3.3.15 Read XUID

The Read XUID is used to retrieve XUID. UID is required for this command. Result of 112-bit complete UID and XUID is reported.

List Cmd		Read XUID Command		
Custom SIC Command     SetEAS     ResetEAS     LockEAS     EAS_Alarm     Kil     LoadPassword     WritePassword     LockPassword	•	ReadXUID Cmd Non Inventory Mode RFU Rag Option Rag Operation Mode Non Address Mode		
Set Password Mode Lock Password Mode Get Password Mode Set OTP Get OTP Write XUID Write XUID		UID	0D7A5800000239E0	
Special Command     Transparent WithCRC     Transparent WithoutCRC     SendEOF	•	ReadXUID Cmd Response Response> 01 - Resp OK But Wrong Data format		

Figure 121 Read XUID command

### 6.3.4 Special command

Similar to ISO14443A and B, the special command available in ISO15693 are **TransparentWithCRC**, **TransparentWithoutCRC** and **SendEOF** as shown in Figure 122.

For both transparent commands, user can input raw hexadecimal code to invoke operation as stated in the RF protocol. The example of Reading data from block 0 of ISO15693 card is shown in Figure 124. Also, User shall review result in transaction logs windows. Moreover, user might need to adjust timeout for card response to be suitable to user applications. The option for timeout in the GUI is shown in Figure 123.

The **SendEOF** is used to transmit a gap in ISO15693 anti-collision process to indicate transition to next slot. However, inventory 16 slot command has already embeds transmitting gap in its process. So, the **SendEOF** can be used in conjunction with transparent command.



Figure 122 Special commands for ISO15693



Figure 123 Timeout setting for TransparentWithCRC (TransparentWithoutCRC) command




#### 6.4 Pico Tag

PicoTag, PicoPass are families of contactless memory IC from INSIDE CONTACTLESS. PicoPass is secure version equipped with INSIDE security encryption mode. By using PI931, user cannot access Read/Write area of PicoPass and PicoCrypt except UID. For PicoTag 16K and 2K, the read/write area is unprotected.

Table 6-14 Pico Product Family					
Product	ISO standards	Cryptographic Security	Memory size and personalization		
PicoPass 16KS	ISO15693 &	No	(8 x 2K pages) or (1 x 16K page)		
PicoPass 2KS	ISO14443B		2K bits		
PicoTag 16K	ISO15693	No	1 x 16K page		
PicoTag 2K			2K bits		
PicoTag 2KS		Yes	2K bits		

The radio frequency power and signal interface for data transmission of PicoTag and PicoPass rely on ISO15693-2 and ISO14443B-2. In another word, bit definition of RF signal in both transmission and reception follows ISO15693-2 and ISO14443B-2. Pico-family commands are proprietary and do not conform ISO15693-3. These commands are called "ISO15693-2 and ISO14443B -2 commands". However, PicoPass does support ISO14443B-3 but the IC must be prior set in that mode. The Pi931-XX, where the SIC9310 is used as an RF front-end chip supporting such standards, can communicate to PicoPass contactless IC.

Figure 125 shows GUI in Pico Tag tap. In this tap, user can transmit single or multiple commands, setup transaction speed as well as review card response. Command list consists of standard commands and transparent commands. The commands in this tag are actually realized from transparent command of ISO15693 and ISO14443B. First of all, user shall set up speed in transaction which can either be ISO15693 or ISO14443B. The options are shown in Figure 126.

Reader Conf	igulation ISO1569	3   ISO14	443/	A ISO14443B Pic	оТад	
Start Send Multi Cmd				Send		
Config Re	ader Speed TX,RX for ISO15693	Reader + TX 1 c	Spee ut of	ed Rx for Tag add in	RequestFlag of every C Sub Hight R₂ ▼ SE	imd T
List Cmd	Enable Once Chec	тх ж	1	PICO Sta Active	Indard Command	
ACTALL IDENTIFY SELECT ASNB SELECT SNB			Active All Cmd	PreviousSetting		
REAL	) 04 pacial Command parentWithCRC ;parentWithoutCRC			Active All Cmd Re	sponse	

Figure 125 Pico Tag command tap

Config R	eader		Config Reader	
	Speed TX,RX for	Reader + Speed Rx for Tag add in RequestFlag of every Cmd	- Speed TX,RX for Reader + Speed Rx for Tag add in RequestFlag of every Cmd	d-
Cotup	ISO15693	TX 1 out of 4 🔹 RX 1 Sub Hight Ra 🔹 SET	○ ISO15693 TX 106 KBits/s ▼ RX 106 KBits/s ▼ SET	
Jerup	ISO14443B	TX RX 1 Sub Hight Rate GET	ISO14443B TX RX IC KBits/s GET	

Figure 126 Speed setup for PICO Tag

#### 6.4.1 Pico Tag standard commands

The standard commands used in PicoTag/PicoPass are illustrated in Figure 127, showing relation of each command in card-state transition diagram. Hence, each command shall be applied at appropriate state. The commands in this section are **ACTALL, IDENTIFY, SELECT ASNB, SELECT SNB, HALT, READ** and **READ4**. For more information about the commands, please refer to PicoTag/PicoPass datasheet.



Figure 127 State diagram of PicoTag / PicoPass

#### 6.4.1.1 ACTALL

The **ACTALL** command GUI is shown Figure 128. This command is used to switch card state from Idle to Activated. This command requires no input. Because response for this command is only SOF of standard we select, then the PI931xx report "E1" response error for ISO15693 and "E0" not response for ISO14443B. This is due to the SOF only is treated to be incomplete frame. Please don't care the answer in this state please go to next command.

Config Reader         Speed TX,RX for Reader + Speed Rx for Tag add in RequestRag of every Cmd                              ● ISO15693 TX 1 out of 4 ▼ RX 1 Sub Hight Rε ▼ SET	Setup         State         State <t< th=""></t<>
Enable Once Check PICO Standard Command	Enable Once Check PICO Standard Command
List Cmd     Active All Command       ACTALL     Active All Cmd       IDENTIFY     SELECT ASNB       SELECT SNB     HALT       READ     Active All Cmd Response       READ4    Special Command       TransparentWithCRC     TensparentWithoutCRC	List Cmd Active All Command Standard Command ACTALL DENTIFY SELECT ASNB SELECT ASNB HALT READ READ4Special Command TransparentWithORC TransparentWithoutCRC

Figure 128 ACTALL command and response

#### 6.4.1.2 IDENTIFY

The **IDENTIFY** command is used to get the ASNB (Anti-collision serial number). The card/tag that be able to receive this command shall be in activated state. Result from this command is shown in Figure 129.



Figure 129 IDENTIFY command



#### 6.4.1.3 SELECT ASNB

The **SELECT ASNB** is used to get the real UID. The required parameter is ASNB from **IDENTIFY** command. If PICOtags receive this command correctly, command **READ** and **READ4** will be operable.



Figure 130 SELECT ASNB command

#### 6.4.1.4 SELECT SNB

The **SELECT SNB** is used to get the real UID. The required parameter is UID from **IDENTIFY** command. If PICOtags receive this command correctly, command **READ** and **READ4** will be operable.

Confin Do	adar				
Coning Ne	- Speed TV DV for	Reader +	Coord Dy fo	r Tao add in	RequestFlag of eveny Cmd
	Speed TX, NXTO	Neauer +	Speed I wild		Thequest hag of every clift
Cotup	ISO15693	TX [10	out of 4	🔹 RX [1	Sub Hight Ra 👻 SET
Secup Secup		тх		- RX	⊸ GET
	Enable Once Che	ck		PICO St	andard Command
List Cmd			s	elect seria	al number Command
S	tandard Command		Selec	ct Cmd	
SELE	CT ASNB		· ·	Timeout	PreviousSetting 💌
SELE	ECT SNB				
HALT READ READ4 Special Command TransparentWithCRC TransparentWithoutCRC			UID	39181600FBFF12E0	
		0.0			
			Sele	ct Cmd Resp	oonse
					Save UID
			Resp	onse> 01 - 1	Resp OK
			[UID>	39181600F	BFF12E0
	Figure	2131 9	SELECT S	SNB con	nmand

#### 6.4.1.5 HALT

The **HALT** command is used to put card in HALT state. In this state, card is silent and waits until receive **SELECT ASNB** and **SELECT SNB** again. Similar to **ACTALL** command, the response from this command is SOF only. Then, it is treated to be incomplete frame. Pi931 respond "E0" and "E1" for ISO14443B and ISO15693 respectably. The **HALT** command and response is shown in Figure 132.

#### 6.4.1.1 READ

The **READ** command is used to read data from specified block in card's memory. Card must prior be set in selected state. The block address is required parameter. However, for some family such as PicoPass, content in memory is protected by proprietary encryption. Hence, reading memory block apart from UID and configuration (Block 0 and Block1) results in all "FF". Pi931 can not perform authentication for PicoPass. Example of READ command result is shown in Figure 133.

# SYRIS

# RD-220 Demonstration Software Manual

-Config Re	ader		
Setup	Speed TX,RX for ISO 15693 ISO 14443B	Reader + Sp TX 1 out TX	eed Rx for Tag add in RequestRag of every Cmd of 4    RX 1 Sub Hight R₂    SET RX    GET
	Enable Once Che	ck	PICO Standard Command
List Cmd			Halt Command
ACT/	tandard Command \LL TIFY	···· •	Halt Cmd Response
	CT ASNB		Timeout PreviousSetting
REAL	)		Halt Cmd Response
REAL	04 pecial Command sparent WithCRC sparent WithoutCRC		Response> E1 - Resp Err
	Fig	gure 132	HALT command
Confin De	a dag		
Config Re	eader Speed TX.RX for	Reader + Sc	peed Rx for Tag add in Request Flag of every Cmd
Config Re	Speed TX,RX for ISO 15693	Reader + Sp TX 1 out	beed Rx for Tag add in Request Flag of every Cmd t of 4    RX 1 Sub Hight R₂   SET
Config Re	Speed TX,RX for ISO 15693	TX 1 out	eeed Rx for Tag add in Request Rag of every Cmd t of 4
Config Re Setup	Speed TX,RX for ISO 15693 ISO 14443B Enable Once Che	Reader + Sp TX 1 out TX ck	eeed Rx for Tag add in Request Rag of every Cmd t of 4 • RX 1 Sub Hight Rz • SET • RX • GET PICO Standard Command
Config Re Setup	Speed TX,RX for Speed TX,RX for ISO 15693 Solution 150 14443B Enable Once Che	Reader + Sp TX 1 out TX	eeed Rx for Tag add in Request Rag of every Cmd t of 4
Config Re Setup	sader Speed TX,RX for ISO 15693 SISO 14443B Enable Once Che tandard Command-	Reader + Sp TX 1 out TX ck	eed Rx for Tag add in Request Flag of every Cmd t of 4 • RX 1 Sub Hight Rz • SET • RX • GET PICO Standard Command Read Command Read Cmd
Setup	sader Speed TX,RX for ISO 15693 ISO 14443B Enable Once Che tandard Command- ALL TIFY ECT ASNB ECT SNB	Reader + Sp TX 1 out TX ck	beed Rx for Tag add in Request Flag of every Cmd t of 4 • RX 1 Sub Hight Rz • SET • RX • GET PICO Standard Command Read Command Read Cmd Timeout PreviousSetting •
Setup	sader Speed TX,RX for Speed TX,RX for ISO 15693 Speed TX,RX for ISO 14443B Enable Once Che tandard Command- ALL TTIFY ECT ASNB ECT SNB T D D	Reader + Sp TX 1 out TX ck	beed Rx for Tag add in Request Flag of every Cmd t of 4
Setup List Cmd ACT. IDEN SELE HALL REAL ENA	sader Speed TX, RX for Speed TX, RX for ISO 15693 Speed TX, RX for ISO 14443B Enable Once Che tandard Command	Reader + Sp TX 1 out TX	beed Rx for Tag add in Request Flag of every Cmd t of 4 • RX 1 Sub Hight Rz • SET • RX • GET PICO Standard Command Read Command Timeout Previous Setting • Block Addr 0 Read Cmd Response
List Cmd List Cmd List Cmd ACT/ DEL SELE HAL REA REA Trans	sader Speed TX, RX for Speed TX, RX for ISO 15693 Speed TX, RX for ISO 1693 Speed TX, RX for Speed TX, RX for ISO 1693 Speed TX, RX f	Reader + Sp TX 1 out TX	beed Rx for Tag add in Request Rag of every Cmd t of 4

Figure 133 READ command

#### 6.4.1.1 READ4

The **READ4** command is similar to **READ** but card answers 4 consecutive block. The required input parameter is starting block address. Example of READ command result is shown in Figure 134.

-Config Re	ader		
	Speed TX,RX for	Reader + Sp	eed Rx for Tag add in RequestFlag of every Cmd
Setup	ISO15693	TX 1 out	of 4   RX 1 Sub Hight Ra   SET
	ISO14443B	тх	▼ RX ▼ GET
	Enable Once Che	sk 🛛	PICO Standard Command
List Cmd			Read4 Command
🔳S	tandard Command	A	Read4 Cmd
SELF			Timeout PreviousSetting -
SELE	CT SNB		
HALT	ŗ		Start Block Addr 0
REAL	) N	_	
Sp	ecial Command		Read Cord Reserves
Transparent WithCRC			Read4 Cmd Response
Trans	parentWithoutCRC		Response> 01 - Resp OK
			order 1
			Block Data > 39181600FBFF12E0
			order 2
			Block Data > 12FFFFFE97FFF3C
			order 3
			Block Data > FFFFFFFF9FFFFF
		~	order 4
			Block Data > FFFFFFFFFFFFFFF

Figure 134 **READ4** command



### RD-220 Demonstration Software Manual

#### 6.4.2 Special command

**TransparentWithCRC**, **TransparentWithoutCRC** are provided to support unavailable command in command list. To use these command, user have to know command code as stated in datasheet.

## RD-220 Demonstration Software Manual

#### 6.5 Felica

Felica is contactless smart card IC from SONY. Line coding of Felica is BPSK both transmission and reception. For more information about Felica frame format, please refer to ISO18092. Note that coding and decoding demonstrates in this section is not performed by the SIC9310 reader IC but it is implemented through raw signal decoding by microcontroller. Due to proprietary encryption, user cannot access Read/Write except UID. Figure 135 shows GUI in Felica tap. In this tap, user can read UID and transmit arbitrary command via C\_TransparentWithCRC. First of all, user shall set up speed in transaction to be 212 kbps.

Reader Configulation	ISO15693 IS	6014443A	ISO14443B	PicoTag	Felica	
Start Send Multi Cmd           Config Reader           TX           212 KBits/s           TX		d er v				▼ SET ▼ GET
Enable C List Cmd C_Polling Special Com C_Transparent	nce Check mand WithCRC		F M	elica Cor None Cor	mmand mmand	I

Figure 135 Felica command tap

#### 6.5.1 Polling command

The **C\_Polling** command is used to get UID from Felica card. This command can perform reading both single tag and multiple tags in case of anti-collision. The required parameters are system code, reserved and the number of slot in anti-collision process. Figure 136 shows reading a Felica card in 1 slot while Figure 137 shows reading multiple cards in 16 slots. Note that if there are slots that collision occurs, PI931 reports an error code (such as "E0", "E1") depending on signal characteristic at that time.

Config Reader				
Speed TX,RX for Reader				
TX 212 KBits/s	▼ RX 212 KBits/s ▼ SET			
Setup TX				
Enable Once Check	Felica Command			
	C_Polling Command			
C. Polling	Polling Cmd			
Special Command				
C_TransparentWithCRC	System Code FFFF			
	Reserved 00			
	Time Slot 1 ▼ Slot			
	Polling Cmd Response			
	Response> 01 - Resp OK			
	slot 0			
	SubResp> 01 - SubResp OK			
	UID> 010102147E0C131B100B4B428485D0F			

Figure 136 **C\_Polling** command in reading a Felica card

# SYRIS

## RD-220 Demonstration Software Manual

Config Reader Speed TX,RX for Reader TX 212 KBits/s TX	→ RX 212 KBits/s    SET → RX    GET				
Enable Once Check	Felica Command				
List Cmd	C_Polling Command				
C Polling	Polling Cmd				
Special Command     C. Terrereret/WithCPC	System Code FFFF				
	Reserved 00				
	Time Slot 16 - Slot				
	Polling Cmd Response				
	Kesponse> U1 - Kesp OK				
	slot 0 SubResp> 01 - SubResp OK UID> 000000000000000000000000000000000000				
	slot 2 SubResp> 01 - SubResp OK UID> 010106018704F10F03014B024F4993Ff				
	slot 12 SubResp> 01 - SubResp OK UID> 010102147E0C151B100B4B428485D0F				

Figure 137 **C\_Polling** command in reading multiple Felica card in 16 slots

#### 6.5.2 TransparentWithCRC

The **TransparentWithCRC** command is used to transmit command in hexadecimal code directly to air. Two byte calculated CRC following Felica standard are appended at the end of transmit data frame. For more information, please refer to commands data format in Felica card manual for constructing data bytes.