	Name	Signature	Last Updated On
Author of Template	Eyal Shahar		2008
QA manager	<No need for soft copy>		

Approval Conformation

Dept.	Name	Date
R&D VUANCE		

PureRFid™ Protocol Manual PRF-RDR-101 Reader

(For Internal Use Only)

Revision History

Document Version	Date	Author	Changes
1.0	3/24/2008	Etai Sela	First Draft
1.1	3/25/2008	Etai Sela, Nir Segal & Eyal Shahar	Adjustments to the first draft
1.2	3/26/2008	Etai Sela & Eyal Shahar	Update of abbreviations and dip-switches (para 2.3)
1.3	3/30/2008	Eyal Shahar	Technical Writing
1.4	2/04/2008	Etai Sela	Added relays command
1.5	May 28th, 2008	Israel Kalman	Review and changes on the Document
1.6	June 18th, 2008	Israel Kalman	1) Adding all the new commands that we agreed upon in the meeting 2) Some of the commands were changed per the same meeting 3) Two topics should be highlighted: 3.1 Adding error detection when downloading new F/W version to reduce the correction time. Since we cannot correct in the packet level but only on page level there is an issue with the page address. If every page has a fixed address there is no problem. Otherwise there is a problem. 3.2 The "Get Tags" message structure was changed. Now there are two combinations of the LF ACK/NACK. In the messages however we use 3 bits for each of the sensors in parallel (bit per sensor). The tags structure should change accordingly.
1.7	August 4th, 2008	Israel Kalman	Protocol Adjustments
1.8	August 18th, 2008	Etai Sela	Modification to the previous version: Sections 2.6, 3.2.13.2,
1.9	March 10th, 2010	Jason Clute	Spelling & format changes & added protocol for technician mode 2, and Wiegand structure
2.0	May 18th, 2010	Jason Clute	Corrected error on dip switch ID setting, changed document title and name

Approvals Signatures (on P4)

Document Version	Date	Name	Signature (Print your name)
1.1		Etai Sela	
1.1		Israel Kalman	
1.1		Eyal Shahar	

PureRFid©
Driving the Future

PRF-RDR-101 Reader

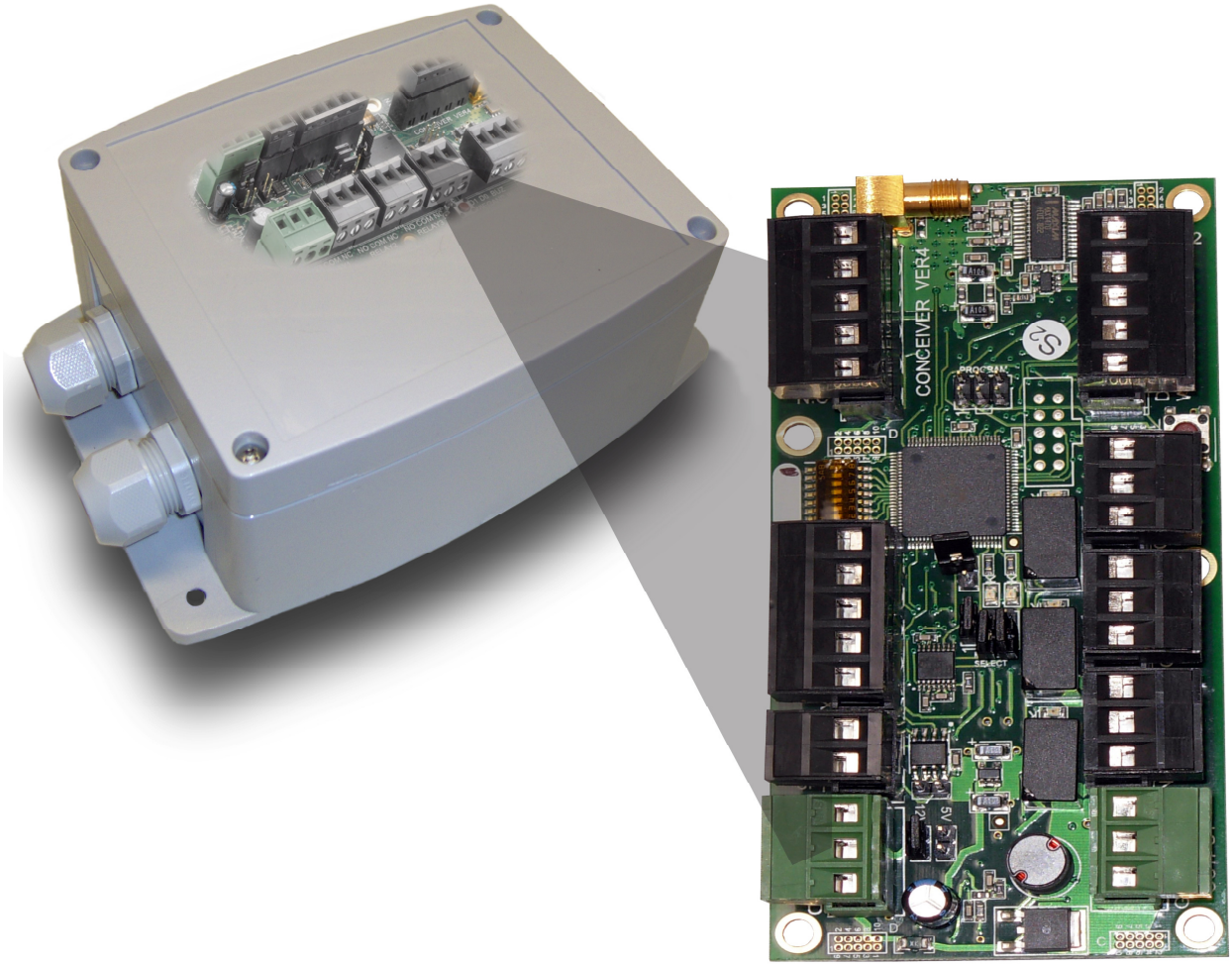


Table of Contents

Abbreviations

API	Application Programming Interface
ASK	Amplitude Shift Keying – Refers to a modulation scheme
GUI	Graphic User Interface
LSB	Least Significant Bit or Byte
MSB	Most Significant Bit or Byte
NC	Normally Closed
NO	Normally Open
PRF-RDR-101	Part number for PureRFid™ Reader
RFID	Radio Frequency Indentification
RSSI	Received Signal Strength Indicator

1. Introduction

1.1 Scope

The PureRFid™ PRF-RDR-101 Reader Controlled Mode Manual describes the operation principles of the Reader in its Controlled or API Mode for receiving commands, providing PureRFid™ Active Tag Data messages and converting various communication protocols. This Manual also lists the protocol output for a stand-alone mode called Technician Mode 1 as well as the Wiegand protocol structure.

This manual is intended to be used by developers wishing to integrate some type of software or firmware application to perform communication and/or data handling with the PureRFid™ Reader. This manual DOES NOT contain any technical information regarding hardware installation and configuration. Please refer to the PRF-RDR-101 User Guide for these specifications.

1.2 Product Overview

PureRFid Inc has developed the PureRFid™ Reader to operate as an RFID reader and protocol converter under one platform to be used in conjunction with the PureRFid™ Active Tags (ASK). The Reader performs the following functions:

- Receives, decodes and validates data from the PureRFid™ Active Tags.
- Can operate in either standalone or network architectures for designing coverage areas as required.
- Outputs the relevant tag data through a reader network to a host unit (PC).
- Performs filtering of tags for Site Code and RSSI.

Converts the tag data into following protocols:

- RS485
- RS232
- 26 Bit Wiegand
- Dry Contact Relay NO & NC

The Receiver is incorporating the following interfaces:

- RF Receiver (RF Receiver and Demodulator)
- Wiegand Interface
- RS485 Interface
- RS232 Interface
- NC/NO Relay Interface
- Power Interface
- SMA Antenna Interface

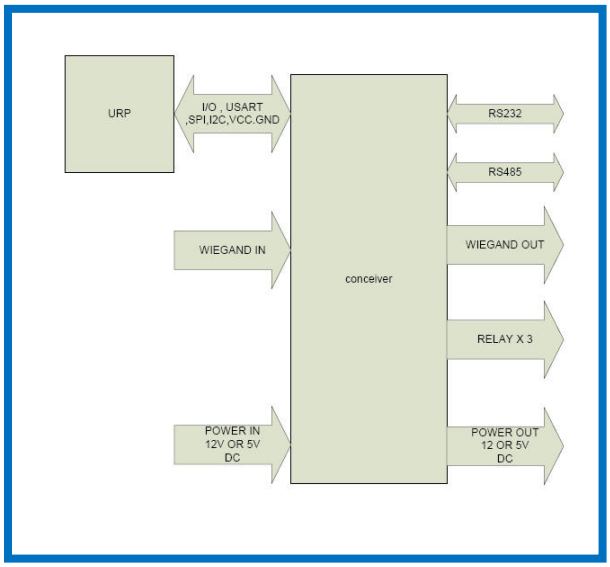


Figure 1.2.1 – Receiver High Level Block Diagram

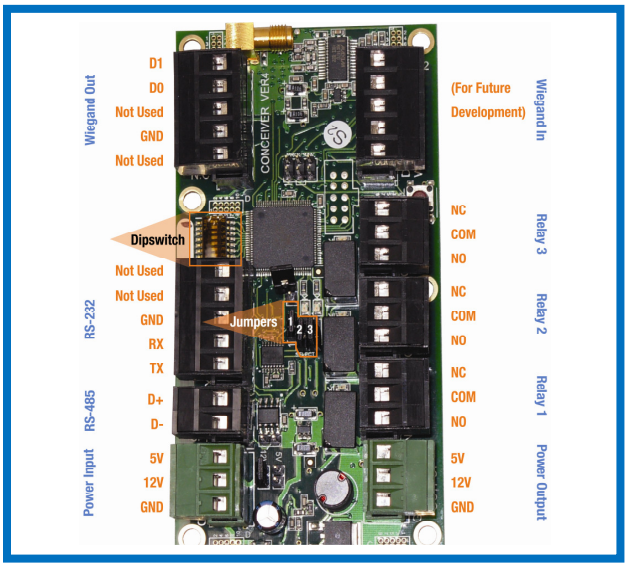


Figure 1.2.2 – The PureRFid™ Reader

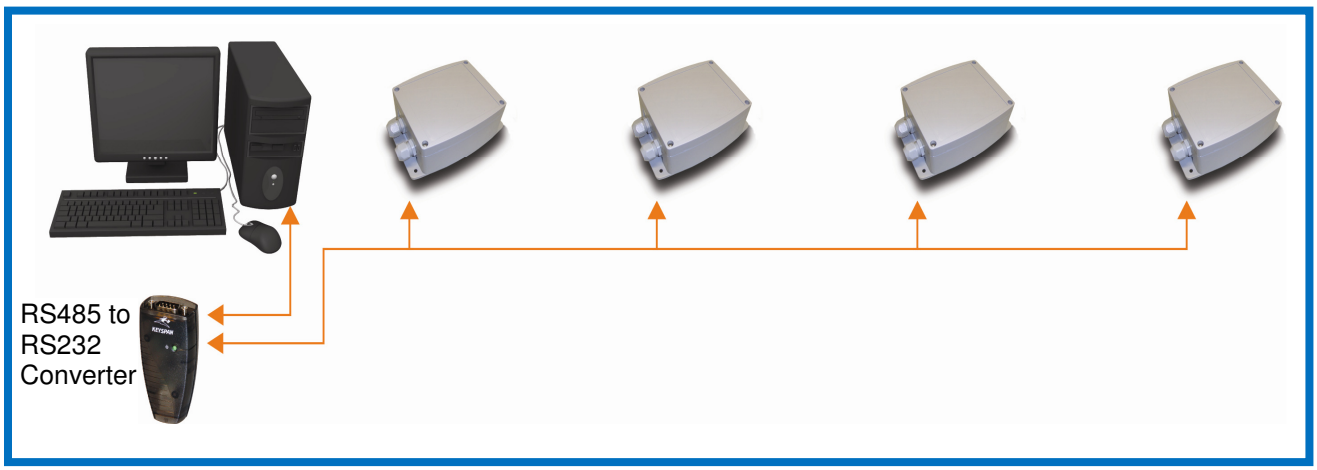


Figure 1.3 – Receivers Network

2. Protocol and Addressing

2.1 Modes of Operation

The Reader is a sophisticated bus-converter and PureRFid™ Tags receiver. The Reader can operate in one of two main modes of operation:

- **Technician 1 or Standalone Mode:** In this mode, the Reader manages the data flow autonomously based on its internal hardware configuration (mode-select Jumpers) and provides data output through RS232, RS485 & 26 Bit Wiegand protocols simultaneously.
- **Controlled or API Mode:** In this mode, the external host (application PC or similar) controls the Reader by a set of commands and responses (protocol). Therefore, the Reader's operation is managed by the host unit (and its F/W or S/W). The Reader can be connected to its host by using one of the following interfaces: RS232, RS485.

2.1.1 Controlled or API Mode

The Reader is set into the Controlled or API Mode by setting the Mode Select jumpers as described in the PRF-RDR-101 User Guide. By using the Jumpers on the board, the Reader becomes a controlled unit that can operate in a Reader network or as a single node reader. The Reader's commands library enable the following functionalities:

- Reception of transmissions from the Pure-RF tags
- Control of the on-board relays
- Management of the on-board 26 Bit Wiegand Output

The basic principle of the Controlled Mode operation is poll-response. The master (host) sends a request to a slave (Reader unit) and waits for its response (timeout limited). The units cannot respond without a request being sent first.

The Controlled Mode has two sub modes that can be set by the protocol commands:

- Main Software Mode (Designated as **M**) - in which the F/W is controlling the Reader
- BootLoader Mode (Designated as **B**) – in which a special separate S/W package is managing the download of a new F/W version from the host to the Reader.

There are commands that operate only in one of the sub-modes and some commands can operate in both, as designated herein in the title of each command.

2.1.2 Technician 1 or Standalone Mode

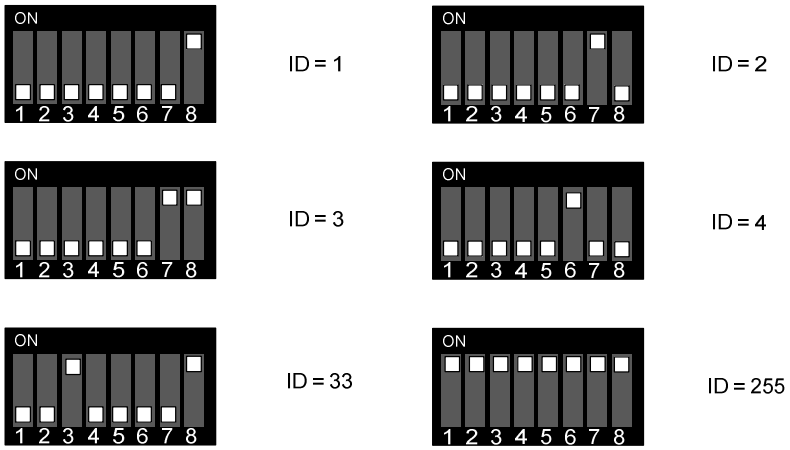
The Reader is set into the Technician 1 or Standalone Mode by setting the Mode Select jumpers as described in the PRF-RDR-101 User Guide. By using the Jumpers on the board, the Reader becomes a standalone unit that can operate as a single Reader that will receive PureRFid Active Tags and output each message using RS232, RS485, or 26 Bit Wiegand protocols.

2.2 Reader Identification

In its Controlled or API Mode the Reader needs an ID for addressing the Tag messages and distinguishing between Readers in a RS485 network. The Controlled Mode ID is configured using the on board dip-switches. Address 0 is saved for broadcast messages and cannot be used as a unit ID. The addresses available for units are from 1 to 255.

Each switch on the dip-switch represents one bit of the ID byte. Switch number 8 is the LSB.

The following is an example of the dip-switch settings:



Note: Every time the dip-switch is changed, the power must be cycled. If Readers are connected on the same RS485 network each Reader needs a different ID.

2.3 Protocol Utilization

For getting Tag messages from the Readers, the host needs to send the **Get Tags** command sequentially to all Readers on the Reader network. The host will get in response the tag information from each addressed Reader respectively.

The Wiegand interfaces and relay operation commands are used per specific activity (the commands are not used continuously).

2.4 Communication Interfaces

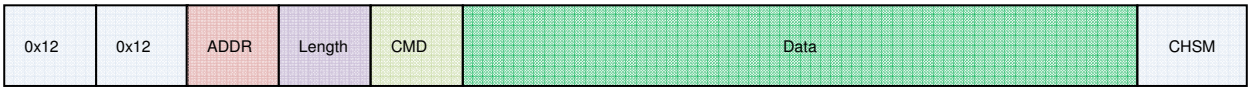
The communication with the Reader is performed by one of the following interfaces:

- RS485 port
- RS232 port

The baud rate is 57600 bps, Data Bits 8, Parity none, Stop Bits 1, Flow Control none.

2.5 Packet Structure

The packets are built in the following structure:



Byte Num.	Purpose	Contents
1 & 2	Sync. Bytes	**0x12**
3	Unit Number (**ADDR**)	The address of the Reader on the network: 1-255 for addressed units 0 for broadcast command
4	Message **Length**	Specifies message length from right after this Byte until after the Check Sum.
5	Command (**CMD**)	Odd numbers for requests; Following even number for related reply.
6-X	**Data**	
X+1	Checksum (**CHSM**)	Sum of all the bytes in the message

3. Commands

3.1 Command List

No	Name	Description	Broadcast Permitted	Main / Bootloader
Basic commands				
0x01	**Get Info**	Receives general identification information from the unit	No	M + B
0x03	**Get Status**	Receives information about unit status	No	M + B
0x05	**Get Debug**	Receives debug information string	No	M
0x07	**Power Control**	Controls unit power	No	M
0x09	**BootLoader Control**	Enters/Exits bootloader mode	No	M + B
0x0B	**Download Firmware**	Writes firmware when in bootloader mode	Yes	B
0x0D	**Firmware Checksum**	Reads firmware checksum	No	B
Info commands				
0x17	**Noise Level**	Reads the noise level in the vicinity of the reader	No	M
Tags commands				
0x31	**Get Tags**	Retrieves the tags picked up by the reader	No	M
Control commands				
0x3F	**Empty Tags Buffer**	Deletes the content of the tag messages buffer	Yes	M
0x41	**Set/Get Name**	Sets/Gets the recader's name (20 letters string)	No	M + B
0x43	**Set/Get RSSI**	Sets/Gets RSSI threshold in Reader	No	M
0x45	**Set/Get Site Code**	Sets/Gets Site Code of Reader	No	M
Wiegand Commands				
0x55	**Send Wiegand Card**	Sends cards through the Wiegand output interface	No	M
Relay Commands				
0x5D	**Activate Relay**	Activates one of the relays on the board	No	M

3.2 Commands Description

3.2.1 Command 0x01 – Get Info (M + B)

Description: Getting general identification information from the unit

3.2.1.1 Request

Byte type	Byte #	Value	Purpose	Contents
Sync.	1 & 2	0x12	Synchronizing the data flow	2X(0x12)
ADDR	3	0x01 to 0xFF	Unit Number	1-255 for units No broadcast permitted
Length	4	0x02	Message Length	Length of message (2 Bytes)
CMD	5	0x01	**Get Info** Command Number	Odd numbers for requests; Following even number for related reply.
CHSM	6		Checksum byte	Sum of all the bytes in the message.

3.2.1.2 Reply

Byte type	Byte #	Value	Purpose	Contents
Sync.	1 & 2	0x12	Synchronizing the data flow	2X(0x12)
ADDR	3	0x01 to 0xFF	Replying Unit Number	1-255, same as request ADDR
Length	4	0x1E	Message Length	Length of message
CMD	5	0x02	**Get Info** Reply Number	The reply number follows the request number.
DATA	6	0 to 0xFF	Protocol Version	BCD encoded protocol version – F/W encoded
DATA	7	0x01	Device Class	Receiver's class is 0x01 – F/W encoded
DATA	8	0x01	Device Sub-Class	Receiver's sub-class is 0x01 – F/W encoded
DATA	9	0 to 0xFF	Firmware Version	BCD encoded firmware version – F/W encoded
DATA	10-29	String	Receiver's name	Set by user. ASCII string of 20 characters.
DATA	30-33	Integer	Receiver's Serial Number	A unique factory set number for each Receiver.
CHSM	34		Checksum byte	Sum of all the bytes in the message.

3.2.2 Command 0x03 – Get Status (M + B)

Description: Getting information about the status of the Receiver unit. The Reader collects data and calculates operational status indicators.

3.2.2.1 Request

Byte type	Byte #	Value	Purpose	Contents
Sync.	1 & 2	0x12	Synchronizing the data flow	2X(0x12)
ADDR	3	0x01 to 0xFF	Unit Number	1-255 for units No broadcast permitted
Length	4	0x02	Message Length	Length of message (2 Bytes)
CMD	5	0x03	**Get Status** Command Number	Odd numbers for requests; Following even number for related reply.
CHSM	6		Checksum byte	Sum of all the bytes in the message.

3.2.2.2 Reply

Byte type	Byte #	Values	Purpose	Contents
Sync.	1 & 2	0x12	Synchronizing the data flow	2X(0x12)
ADDR	3	0x01 to 0xFF	Unit Number	1-255 for units No broadcast permitted
Length	4	0x0A	Message Length	Length of message
CMD	5	0x04	**Get Status** Reply Number	The reply number follows the request number.
DATA	6	0x01=Main 0x02=Bootloader	Unit's sub mode of operation	0x01=Main 0x02=Bootloader
DATA	7-10	0x00 – 0xFFFFFFFF	Uptime	Uptime (in seconds) since last reset.
DATA	11	0x01 - 0x64	Average Processor Workload	Not defined/implemented yet. One Byte reserved to provide indication on Receivers processor workload. Average workload is since last read till present. Has to be stored in E^2
DATA	12	0x01 - 0x64	Number of Resets	Number of resets since last read. Cycling the Power does not reset the counter.
DATA	13		Number of Tag messages in Receiver's buffer	
CHSM	14		Checksum byte	Sum of all the bytes in the message.

3.2.3 Command 0x07 – Power Control (M)

Description: This Command and response are for controlling and monitoring the on/off of RF receiver of the Reader and for restarting the Reader.

3.2.3.1 Request

Byte type	Byte #	Value	Purpose	Contents
Sync.	1 & 2	0x12	Synchronizing the data flow	2X(0x12)
ADDR	3	0x01 to 0xFF	Unit Number	1-255 for units No broadcast permitted
Length	4	0x03	Message Length	Length of message (3 Bytes)
CMD	5	0x07	**Power Control** Command Number	Odd numbers for requests; Following even number for related reply.
DATA	6	0x01-0x04	Mains Power Options	0x01=RF Power Off (command) 0x02=RF Power On (command) 0x03=Reset (command) 0x04=Inquiry (data request)
CHSM	7		Checksum byte	Sum of all the bytes in the message.

3.2.3.2 Reply

Byte type	Byte #	Value	Purpose	Contents
Sync.	1 & 2	0x12	Synchronizing the data flow	2X(0x12)
ADDR	3	0x01 to 0xFF	Unit Number	1-255 for units No broadcast permitted
Length	4	0x05	Message Length	Length of message
CMD	5	0x08	**Power Control** Reply Number	The reply number follows the request number.
DATA	6	0x01-0x03	Current power mode	0x01=RF Power Off 0x02=RF Power On 0x03=Reset After This Reply
DATA	7-8	0x00 – 0xFF	Measured input voltage	Multiplied by 100mV. Nominal value = 120 (12V); Permitted tolerance = 104 to 144
CHSM	9		Checksum byte	Sum of all the bytes in the message.

Note: Remember that Power on/off refers only to the RF receiver of the Reader!

3.2.4 Command 0x09 – BootLoader Control (M + B)

Description: Enter/Exit Bootloader Mode

3.2.4.1 Request

Byte type	Byte #	Value	Purpose	Contents
Sync.	1 & 2	0x12	Synchronizing the data flow	2X(0x12)
ADDR	3	0x01 to 0xFF	Unit Number	1-255 for units No broadcast permitted
Length	4	0x03	Message Length	Length of message
CMD	5	0x09	BootLoader Control Command Number	Odd numbers for requests; Following even number for related reply.
DATA	6	0x01-0x02	New Mode	0x01=Enter Main Mode 0x02=Enter Bootloader Mode
CHSM	7		Checksum byte	Sum of all the bytes in the message.

3.2.4.2 Reply

Byte type	Byte #	Value	Purpose	Contents
Sync.	1 & 2	0x12	Synchronizing the data flow	2X(0x12)
ADDR	3	0x01 to 0xFF	Unit Number	1-255 for units No broadcast permitted
Length	4	0x04	Message Length	Length of message (4 Bytes)
ADDR	5	0x01 to 0xFF	Unit Number	1-255 for units
CMD	6	0x0A	BootLoader Control Reply Number	The reply number follows the request number.
DATA	7	0x01-0x02	Current mode	0x01=Main Mode 0x02=BootLoader Mode 0x03=Non-operable Main (and remaining in BootLoader mode)
CHSM	8		Checksum byte	Sum of all the bytes in the message.

3.2.5 Command 0x0B – Download Firmware (B)

Description: Command for Downloading New Firmware to the Unit

3.2.5.1 Request

Byte type	Byte #	Value	Purpose	Contents
Sync.	1 & 2	0x12	Synchronizing the data flow	2X(0x12)
ADDR	3	0x00 to 0xFF	Unit Number	0-255 for units Broadcast permitted
Length	4	0x14	Message Length	Length of message
CMD	5	0x0B	**Download F/W** Command Number	Odd numbers for requests; Following even number for related reply.
DATA	6-7	0x00-0xFFFF	Packet Number	
DATA	8-23	16 Bytes of F/W	F/W Packet	
CHSM	24		Checksum byte	Sum of all the bytes in the message.

Note: This command can be broadcasted or addressed. When broadcasted there is no Ack, so the transmission is repeated three times to improve reliability. When addressed, there is an Ack., so each packet is transmitted once.

3.2.5.2 Reply (for addressed Requests only)

Byte type	Byte #	Value	Purpose	Contents
Sync.	1 & 2	0x12	Synchronizing the data flow	2X(0x12)
ADDR	3	0x01 to 0xFF	Unit Number	1-255 for units No broadcast permitted
Length	4	0x05	Message Length	Length of message
CMD	5	0x0C	**Download F/W** Reply Number	The reply number follows the request number.
DATA	6-7	0x00 - 0xFFFF	Packet Number	The same number as in the request
DATA	8	0x01 – 0x02	Packet confirmation	0x01=Ok 0x02=Error
CHSM	9		Checksum byte	Sum of all the bytes in the message.

Explanations:

1 F/W Packet = 16 Bytes

1 F/W Page = 16 Packets = 256 Bytes

3.2.6 Command 0x0D – Firmware Checksum (B)

Description: Command for Checking whether the Main Firmware is OK

3.2.6.1 Request

Byte type	Byte #	Value	Purpose	Contents
Sync.	1 & 2	0x12	Synchronizing the data flow	2X(0x12)
ADDR	3	0x01 to 0xFF	Unit Number	1-255 for units No broadcast permitted
Length	4	0x02	Message Length	Length of message (2 Bytes)
CMD	5	0x0D	**F/W Checksum** Command Number	Odd numbers for requests; Following even number for related reply.
CHSM	6		Checksum byte	Sum of all the bytes in the message.

3.2.6.2 Reply

Byte type	Byte #	Value	Purpose	Contents
Sync.	1 & 2	0x12	Synchronizing the data flow	2X(0x12)
ADDR	3	0x01 to 0xFF	Unit Number	1-255 for units No broadcast permitted
Length	4	0x0D	Message Length	Length of message
CMD	5	0x0E	**F/W Checksum** Reply Number	The reply number follows the request number.
DATA	6-7	0x0000 – 0xFFFF	Firmware checksum	2Bytes
DATA	8	0x00- 0x02	Firmware State	0x00 – No firmware 0x01 – Bad firmware 0x02 – Firmware OK
DATA	9-16		Bad Pages List (Not implemented yet)	The list designates 4 F/W pages that have been detected with errors during the download.
CHSM	17		Checksum byte	Sum of all the bytes in the message.

3.2.7 Command 0x17 – Noise Level (M)

Description: Used for reading the level of noise in the vicinity of the reader.

3.2.7.1 Request

Byte type	Byte #	Value	Purpose	Contents
Sync.	1 & 2	0x12	Synchronizing the data flow	2X(0x12)
ADDR	3	0x01 to 0xFF	Unit Number	1-255 for units No broadcast permitted
Length	4	0x02	Message Length	Length of message
CMD	5	0x17	**Noise Level** Command Number	Odd numbers for requests; Following even number for related reply.
CRC	6		Checksum byte	Sum of all the bytes in the message.

3.2.7.2 Reply

Byte type	Byte #	Value	Purpose	Contents
Sync.	1 & 2	0x12	Synchronizing the data flow	2X(0x12)
ADDR	3	0x01 to 0xFF	Unit Number	1-255 for units
Length	4	0x04	Message Length	Length of message
CMD	5	0x18	**Noise Level** Reply Number	The reply number follows the request number.
DATA	6-7	0x00 – 0x3FF	Noise level	A number in range of 0 to 1023 representing noise level in the vicinity of the Receiver
CHSM	8		Checksum byte	Sum of all the bytes in the message.

3.2.8 Command 0x31- Get Tags (M)

Description: Used to retrieve tag messages received by the reader.

3.2.8.1 Request

Byte type	Byte #	Value	Purpose	Contents
Sync.	1 & 2	0x12	Synchronizing the data flow	2X(0x12)
ADDR	3	0x01 to 0xFF	Unit Number	1-255 for units No broadcast permitted
Length	4	0x04	Message Length	Length of message
CMD	5	0x31	**Get Tags** Command Number	Odd numbers for requests; Following even number for related reply.
DATA	6	0x01-0x08	Maximum number of tags	The maximum number of tags that the receiver's response can contain (maximum is 8)
DATA	7	0x00 – 0xFF	Package identification number	Random number. The package identification mechanism prevents loss of tags packets
CHSM	8		Checksum byte	Sum of all the bytes in the message.

3.2.8.2 Reply

Byte type	Byte #	Value	Purpose	Contents
Sync.	1 & 2	0x12	Synchronizing the data flow	2X(0x12)
ADDR	3	0x01 to 0xFF	Unit Number	1-255 for units
Length	4	0xFF	Message Length	Length of message
CMD	5	0x32	**Get Tags** Reply Number	The reply number follows the request number.
DATA	6		Status and indication byte	Used to indicate on various receiver properties (TBD)
DATA	7	0x00 – 0xFF	Package identification number	The same number that was sent in the request. If the packet was not delivered for some reason it can be asked again
DATA	8	0x00 – 0x08	Number of tags in the packet	
DATA	9	0x00 – 0x28	Tag messages left in buffer	Represents the number of tag messages left in Receiver's buffer excluding this transmission
DATA	10 + Ten bytes per each Tag	(Details below)	Tags data.	(Details in the table below)
CHSM	Last Byte		Checksum byte	Sum of all the bytes in the message.

Note: In the **Get Tags** request in Byte 7, the package identification number must be different from 1 request to the next. A random value between 00-FF can be used for each request or you can simply alternate 2 different values each time. For example, the first request can be 00, the next request can be 01, then the third request start back over at 00.

Note: The Reader will buffer 50 Tag messages. For each **Get Tags** request you can obtain between 1-8 tag messages from the buffer at a time.

Tag Data Structure Details: Each tag includes all the successive bytes. If there is more than one tag, the tags come consecutively until the end of the package.

Byte No.	Description
1	Site Code
2-4	Tag ID – Factory set for each tag (Can be changed by PureRFid using Initializer Device)
5	Transmission index + type of message from the tag The five lower bits represent the transmission index, between 1 – 30 (Decimal) The three upper bits represent the type of message from the tag, between 0 and 7 (detailed below) Example: if a decimal 69 was received, it is 01000101 (message | index) in binary – transmission index is 5 and the type of message from the tag is 2. Transmission Index is used for filtering reception duplicates. Message represents various properties of the tag as presented in the table hereunder. Transmission Index cannot use "00000" and "11111" since these combinations are reserved for Ack and Nack of the LF channel (Initializer).
6	Activator number, in case of activation. 0 in all other cases.
7-8	RSSI level (0 - 1,024 levels)
9-10	Noise level on receive time (0 - 1,024 levels measured just prior of receiving tag).

The higher 3 bits indicate tag status:

- Bit 8 – Tamper/Panic ON/OFF (0=OFF 1=ON)
- Bit 7 – Motion Sensor ON/OFF (0=OFF 1=ON)
- Bit 6 – Low Battery alert ON/OFF (0=OFF 1=ON)

3.2.9 Command 0x3F – Empty Tags Buffer (M)

Description: Used to clear the tags messages buffer.

3.2.9.1 Request

Byte type	Byte #	Value	Purpose	Contents
Sync.	1 & 2	0x12	Synchronizing the data flow	2X(0x12)
ADDR	3	0x00 to 0xFF	Unit Number	0-255 for units Broadcast permitted
Length	4	0x02	Message Length	Length of message
CMD	5	0x3F	**Empty Tags Buffer** Command Number	Odd numbers for requests; Following even number for related reply.
CHSM	6		Checksum byte	Sum of all the bytes in the message.

3.2.9.2 Reply (for non broadcast commands only!)

Byte type	Byte #	Value	Purpose	Contents
Sync.	1 & 2	0x12	Synchronizing the data flow	2X(0x12)
ADDR	3	0x01 to 0xFF	Unit Number	1-255 for units
Length	4	0x03	Message Length	Length of message
CMD	5	0x40	**Empty Tags Buffer** Reply Number	The reply number follows the request number.
DATA	6	0x00 – 0x01	Success or failure	0x01 – success 0x00 – failure
CHSM	7		Checksum byte	Sum of all the bytes in the message.

3.2.10 Command 0x41 – Set/Get Name (M)

Description: Used to get/set the receiver's name.

3.2.10.1 Request

Byte type	Byte #	Value	Purpose	Contents
Sync.	1 & 2	0x12	Synchronizing the data flow	2X(0x12)
ADDR	3	0x01 to 0xFF	Unit Number	1-255 for units No broadcast permitted
Length	4	0x17	Message Length	Length of message
CMD	5	0x41	**Set/Get Name** Command Number	Odd numbers for requests; Following even number for related reply.
DATA	6	0x01 – 0x02	Get/Set	0x01 – Get Name 0x02 – Set Name
DATA	7-26	String	Receiver's Name	The Receiver's name is an ASCII encoded string. If less than 20 bytes are used for name, the rest should be '\0'.
CHSM	27		Checksum byte	Sum of all the bytes in the message.

3.2.10.2 Reply

Byte type	Byte #	Value	Purpose	Contents
Sync.	1 & 2	0x12	Synchronizing the data flow	2X(0x12)
ADDR	3	0x01 to 0xFF	Unit Number	1-255 for units
Length	4	0x16	Message Length	Length of message
CMD	5	0x42	**Set/Get Name** Reply Number	The reply number follows the request number.
DATA	6-25	String	Receiver's Name	The Receiver's name is an ASCII encoded string. If less than 20 bytes are used for name, the rest should be '\0'
CHSM	26		Checksum byte	Sum of all the bytes in the message.

3.2.11 Command 0x43 – Set/Get RSSI Filter (M)

Description: Used to get/set the receiver's RSSI threshold for controlling the maximum range of reading tags. After setting the RSSI threshold, the Receiver transfers only tag receptions with higher RSSI.

3.2.11.1 Request

Byte type	Byte #	Value	Purpose	Contents
Sync.	1 & 2	0x12	Synchronizing the data flow	2X(0x12)
ADDR	3	0x01 to 0xFF	Unit Number	1-255 for units No broadcast permitted
Length	4	0x05	Message Length	Length of message
CMD	5	0x43	**Set/Get RSSI Filter** Command Number	Odd numbers for requests; Following even number for related reply.
DATA	6	0x01 – 0x02	Get/Set	0x01 – Get RSSI 0x02 – Set RSSI
DATA	7-8	Integer	RSSI Threshold	A number up to 1023
CHSM	9		Checksum byte	Sum of all the bytes in the message.

3.2.11.2 Reply

Byte type	Byte #	Value	Purpose	Contents
Sync.	1 & 2	0x12	Synchronizing the data flow	2X(0x12)
ADDR	3	0x01 to 0xFF	Unit Number	1-255 for units
Length	4	0x04	Message Length	Length of message
CMD	5	0x44	**Set/Get RSSI Filter** Reply Number	The reply number follows the request number.
DATA	6-7	Integer	RSSI Threshold Value	A number up to 1023
CHSM	8		Checksum byte	Sum of all the bytes in the message.

3.2.12 Command 0x45 – Set/Get Site Code (M)

Description: Used to get/set the receiver's Site Code for filtering received tags to only tags set to same Site Code.

3.2.12.1 Request

Byte type	Byte #	Value	Purpose	Contents
Sync.	1 & 2	0x12	Synchronizing the data flow	2X(0x12)
ADDR	3	0x01 to 0xFF	Unit Number	1-255 for units No broadcast permitted
Length	4	0x04	Message Length	Length of message
CMD	5	0x45	**Set/Get Site Code** Command Number	Odd numbers for requests; Following even number for related reply.
DATA	6	0x01 – 0x02	Get/Set	0x01 – Get Site Code 0x02 – Set Site Code
DATA	7	Integer	Site Code	A number up to 255 used for Site Code.
CHSM	8		Checksum byte	Sum of all the bytes in the message.

3.2.12.2 Reply

Byte type	Byte #	Value	Purpose	Contents
Sync.	1 & 2	0x12	Synchronizing the data flow	2X(0x12)
ADDR	3	0x01 to 0xFF	Unit Number	1-255 for units
Length	4	0x03	Message Length	Length of message
CMD	5	0x46	**Set/Get Site Code** Reply Number	The reply number follows the request number.
DATA	6	Integer	Site Code	A number up to 255 designating the site code.
CHSM	8		Checksum byte	Sum of all the bytes in the message.

Note: Setting the site code filter to 0x00 will allow all site codes to be received.

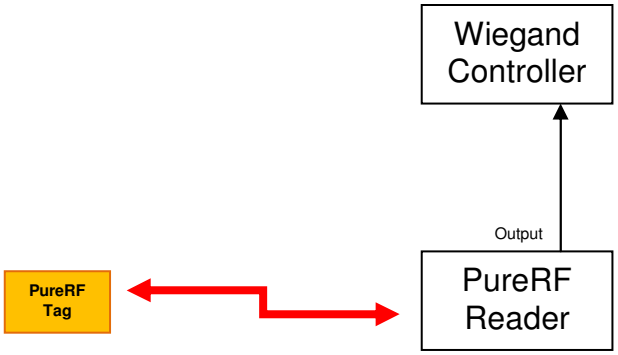
3.3.1 Wiegand Protocol & Relay Output

3.3.1.1 Wiegand Introduction

Wiegand is a type of data protocol that is commonly found and utilized in the card access control industry. It transmits binary information in a serial fashion along three wires: Data 0, Data 1, and a reference to ground. The PureRFid™ Reader will output 26 bit Wiegand data in both stand alone mode and controlled mode. Below are illustrations of both.

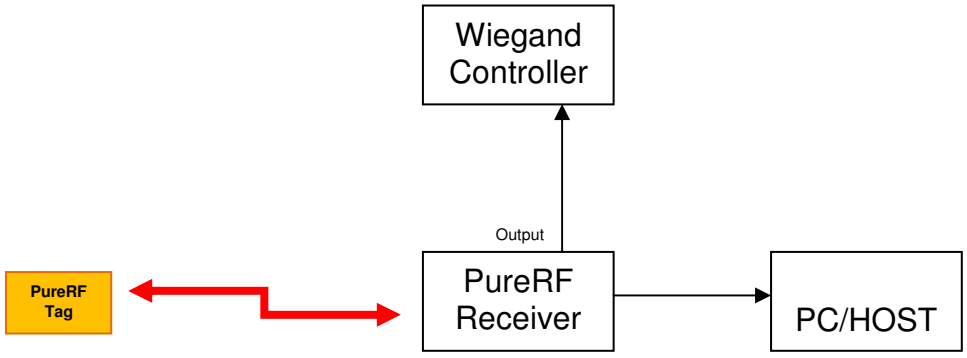
- **Technician 1 or Standalone Mode:** Facility code + Card ID for received RFID tags
- **Controlled or API Mode:** Upon software request command

TECHNICIAN 1 OR STANDALONE MODE



- *Reader receives Tag Message*
- *Reader decodes Wiegand numbering from 3 byte Tag ID*
- *Reader outputs 26 bit Wiegand Protocol data to controller*

CONTROLLED OR API MODE



- *Reader receives Tag Message*
- *Reader sends to PC/Host application for verification*
- *PC sends Wiegand command to Reader*
- *Reader sends Wiegand protocol data to controller*

Note: PC/Host application can generate the Wiegand command without a tag received event

3.3.1.2 Wiegand Format

26 Bit format is the most common bit structure in the card industry. It allows for a 1 byte facility code and a 2 byte card ID. Two bits are used for parity. This provides 255 facility codes with 65,535 card ID's in each facility code.

When the reader is in the Technician 1 or Standalone Mode it automatically sends a 26 bit message from it's Wiegand Out port each time it receives a tag. The facility code and card ID are parsed from the 3 byte Tag ID as referenced in the Tag Data Details Table in section 3.2.8. The byte Numbers listed in the table below refer to the byte position of the Tag Data Details Table.

Tag ID Byte No.	Description
2-3	2 Byte Card ID (0-66535)
4	1 Byte Facility Code (0-255)

Example: A tag transmitted has a decimal site code of 10 and a tag ID of 328643. The Wiegand numbering is parsed from the Tag ID 328643.

Tag ID = 328643 or in Hex 0x0503C3		
Byte 4	**Byte 3**	**Byte 2**
0x05	0x03	0xC3
5	0x03C3 = 963	

So the Wiegand Facility Code = 5 and the Card ID = 963

3.3.1.3 Command 0x55 – Send Wiegand Card (M)

Description: Used to send card over the Receiver's Wiegand port.

3.3.1.4 Request

Byte type	Byte #	Value	Purpose	Contents
Sync.	1 & 2	0x12	Synchronizing the data flow	2X(0x12)
ADDR	3	0x01 to 0xFF	Unit Number	1-255 for units No broadcast permitted
Length	4	0x05	Message Length	Length of message
CMD	5	0x55	**Send Wiegand Card** Command Number	Odd numbers for requests; Following even number for related reply.
DATA	6	0x00 – 0xFF	Site Code	Card site code
DATA	7-8	0x00 – 0xFFFF	Card Number	Card number
CHSM	9		Checksum byte	Sum of all the bytes in the message.

3.3.1.5 Reply

Byte type	Byte #	Value	Purpose	Contents
Sync.	1 & 2	0x12	Synchronizing the data flow	2X(0x12)
ADDR	3	0x01 to 0xFF	Unit Number	1-255 for units
Length	4	0x02	Message Length	Length of message
CMD	5	0x56	**Send Wiegand Card** Reply Number	The reply number follows the request number.
DATA	6	0x00 – 0xFF	Wiegand port status	Busy, another card is currently being sent Sending the card received in the request
CHSM	7		Checksum byte	Sum of all the bytes in the message.

3.3.1.6 Relay Output Introduction

There are 3 dry contact relays located on the PRF-RDR-101 Reader. The relays will not activate at all in the Technician 1 mode but there are other standalone modes where the relays will activate. Refer to the PRF-RDR-101 User Guide for information on these modes. The status of the each relay can be controlled by PC/Host in Controlled or API mode. The relays can be used to carry small amounts of current to switch local lights or buzzers as well as provide dry contact inputs to access control, alarm panels, etc. Each relay is a single pole connection providing NO, NC contacts.

3.3.1.7 Command 0x5D – Activate Relay (M)

3.3.1.8 Request

Byte type	Byte #	Value	Purpose	Contents
Sync.	1 & 2	0x12	Synchronizing the data flow	2X(0x12)
ADDR	3	0x01 to 0xFF	Unit Number	1-255 for units No broadcast permitted
Length	4	0x04	Message Length	Length of message
CMD	5	0x5D	**Activate Relay** Command Number	Odd numbers for requests; Following even number for related reply.
DATA	6	0x00 – 0x02	Relay number	The number of the relay that going to be activated
DATA	7	0x00 – 0xFF	Activation time	0x00 = relay off 0xFF = relay always on 0x01 – 0xFE = the time in 100ms units that the relay will be on.
CHSM	8		Checksum byte	Sum of all the bytes in the message.

3.3.1.9 Reply

Byte type	Byte #	Value	Purpose	Contents
Sync.	1 & 2	0x12	Synchronizing the data flow	2X(0x12)
ADDR	3	0x01 to 0xFF	Unit Number	1-255 for units
Length	4	0x03	Message Length	Length of message
CMD	5	0x5E	**Activate Relay** Reply Number	The reply number follows the request number.
DATA	6	0x01 – 0x02	Status	0x01 = relay activated successfully 0x02 = error occurred
CHSM	7		Checksum byte	Sum of all the bytes in the message.

4. Technician 1 or Standalone Mode

As stated earlier this mode can be used when a Host/PC is not interfaced with the reader. It will output tag messages in a very basic format autonomously using RS232, RS485, & 26 Bit Wiegand. The Wiegand structure is detailed in section 3.3.1.2. The serial port settings are the same as listed in Section 2.4. The serial format for RS232 & RS485 uses CSV text format. Windows HyperTerminal can be used to view the characters transmitted for each tag message. If no tags are being received then no serial data is transmitted. When a tag is received by the reader, it sends out a simple message for each tag reception on 1 line that is similar to the Tag Data Structure Details listed in section 3.2.8. Each column is separated by a comma. The line is returned for each tag message.

Column No.	Description
1	Tag ID – Factory set for each tag (Can be changed by PureRFid using Initializer Device)
2	Transmission index (incrementing value for each transmission of 1-30)
3	Tag Status (T=Tamper, M=Motion, B=Low Battery)
4	Activator number, in case of activation. 0 in all other cases.
5	RSSI level (0 - 1,024 levels)
6	Noise level on receive time (0 - 1,024 levels measured just prior of receiving tag).

Note: In this mode the entire 1 byte site code and 3 Byte tag ID are combined into one 4 byte tag ID. Site Code and RSSI filtering will still be applied if set using another mode.