RS232-to-Wiegand Format Converter (RWFC)
Quick-Reference Installation Guide

Revision 2.0 – 12/22/24

Kenneth Delahoussaye Consulting
web: www.kadtronix.com
email: kadtronix@att.net

Copyright © 2015 - 2024
All rights reserved

Table of Contents

1 - Revision History

Rev.	Date	Description
2.0	12/22/24	• Added section for (new) validation commands
1.6	04/30/21	• Added section describing LED indicators
1.5	11/24/19	• Updated photo images.
1.4	01/25/18	• Added section describing the pulse-width option setting.
1.3	12/19/16	• Updated introductory description of the section describing miscellaneous commands
1.2	11/15/16	• Updated description of Wiegand pullups • Revised section describing peripheral activation • Corrected photo reference
1.1	11/14/16	• Clarified availability of the USB cable
1.0	09/01/15	• Initial Release

2 - Introduction

The RS232-to-Wiegand Format Converter (RWFC) provides format conversion for host computers and RS-232 devices including magnetic stripe and barcode readers. It may also be used with specialized devices such as Bluetooth serial dongles. (If the source device has only USB port(s), a USB-to-RS232 adapter will be required.)

The RWFC is a DCE (data communications equipment) device with DB9 female connector. (Please refer to the applicable note below for additional information.) It also features detachable USB cable and AC adapter. The RWFC is ideally suited for use in access control and other applications where it may be necessary to interface a serial device to a Wiegand-based system. It provides seamless format conversion, accepting a RS-232 source input and outputting a Wiegand stream of bits.

Consider these features:

- Accepts RS-232 data source including magstripe or barcode reader
- Source can also be a host PC with USB port (requires USB-to-RS232 adapter)
- Also accepts data from specialized devices including RS-232 wireless dongles
- Outputs standard Wiegand format (26-bit or 8-bit burst)
- Optional custom formats also possible (via Windows host software and/or RS-232 ASCII commands)
- DB-9 (male) connector
- Onboard relay (for optional activation of external device or equipment
- Housed in attractive plastic enclosure
- AC adapter (12vdc)

Notes:

- Some RS-232 sources output extraneous characters that precede the actual data. For instance, magnetic stripe readers are known to output additional characters which are used as track delimiters. Using the Windows host software, you can configure the RWFC to ignore these characters.
- When connecting to another DCE device such as a magstripe card reader, a null-modem adapter is required. When connecting to a DTE (data terminal equipment) device such as a host computer, the adapter is not necessary.
- Source devices or systems without a compatible RS-232 port will require a suitable adapter. For instance, a PC or laptop with only USB ports can utilize a USB-to-RS232 adapter.)
- A wireless interface is possible with the use of a serial Bluetooth dongle. This device has a RS-232 port which can be connected to the RWFC, permitting Wiegand conversion of ASCII data received from a Bluetooth-enabled host.

3 - Setup

Setup primarily involves the following operations:

- Attach the RS-232 source device
- Connect the Wiegand destination device
- Establish DIP switch option settings
- Plug in the AC adapter

If the source device is a host computer (PC or laptop) with no RS-232 ports, a USB adapter cable will be necessary (available from Kadtronix). In this instance, perform the following additional setup steps:

- Install the Windows drivers (using the CD included with the adapter cable)
- Attach the DB9 end of the adapter cable to the RWFC controller
- Attach the USB end of the adapter cable to an available port on the host PC
- Verify that Windows identifies the adapter and loads the drivers

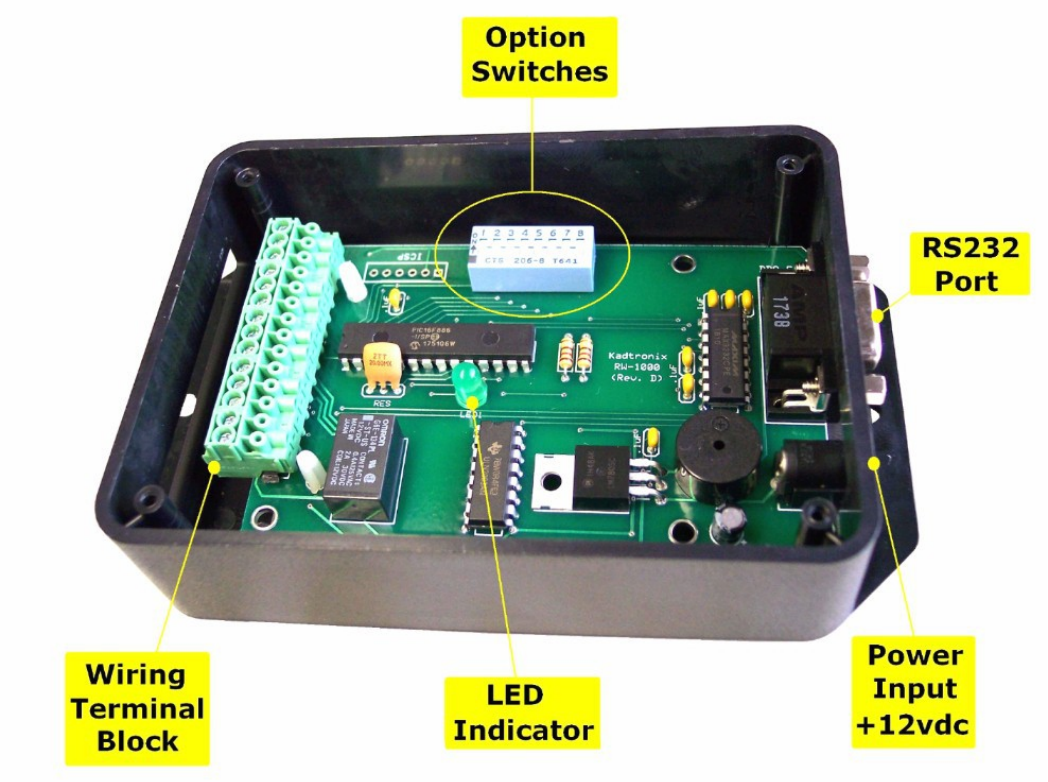
Additional setup & installation details are provided in subsequent sections of this manual.

4 - Hardware Installation

WARNING: Be sure that power is not applied until all wiring has been completed and verified.

Perform the following operations:

- Disconnect the AC power source
- Detach cover by removing the retaining screws.
- Assign desired options. (Refer to "Option Settings".)
- Locate the 14-position wiring terminal connector. (Refer to "TB-1 Wiring Connections".)
- Attach wires associated with the Wiegand output destination
- Connect the RS-232 source device at the DB9 connector. (The RWFC is a Data Communications Equipment [DCE] device. If the source device is also a DCE device such as a magstripe card reader, a null-modem adapter will be needed.)
- Re-attach cover and secure with retaining screws
- Connect the barrel plug of the AC power adapter into the mating receptacle of the RWFC.
- Plug the AC adapter unit into a suitable wall outlet.



5 - LED Indicators

Two LED indicators are provided as described in the table below. (The cover must be detached and removed for viewer accessibility.)

LED Indicators:

Color	Purpose	Description
Green	Status Indicator	The status LED flashes at a steady one-second rate, indicating the firmware is operational and the unit is functioning properly. (The smooth, periodic blink rate may be momentarily disrupted when RS-232 data is received at the DB9 serial port. This is normal behavior.)
Red	Wiegand Activity	This indicator briefly illuminates upon initiating a Wiegand data transmission at the D0 & D1 signal lines. (Wiegand transmission occurs after reception and subsequent conversion of a valid ASCII source string from the DB9 serial port.)

Notes:

- Indicators are not viewer-accessible without removing the cover.
- The Wiegand activity indicator was added as a feature revision and is native to units manufactured after July 2018. (Earlier versions include only the green status LED.)

6 - Option Settings

The RWFC controller has an 8-position set of on-board DIP switches for configuring miscellaneous options. To access the switches, remove the retaining screws from the enclosure and then carefully detach the lid. (Be sure to power off the device prior to opening the enclosure.) The photo above depicts interior elements of the RWFC device including option switches:

Note:

- The RWFC controller must be powered off prior to making any DIP switch changes.

6.1 - Options

The DIP switches permit configuration of the following options:

- Output format
- Source baud rate
- Wiegand pullups

6.1.1 - Output Format (switches 1, 2)

This setting defines the Wiegand format conversion to perform. You may select a pre-stored format type or define a custom format. Configure the switches as follows for the desired output conversion type:

Conversion	**DSW1 Switches**
26-bit Wiegand	1 = off, 2 = off (default)
8-bit burst	1 = off, 2 = on
Custom format	1 = on, 2 = off (see note below)
Reserved	1 = on, 2 = on

Note:

- Custom formatting is possible via RS-232 commands. For custom formatting details, refer to "ASCII Commands".
- The RWFC controller must be powered off prior to making any DIP switch changes.

6.1.2 - RS232 Baud Rate (switch 3)

This setting establishes the data rate for communicating with the source device or host computer (for performing optional host commands). Configure the switches as follows for the desired baud rate:

Baud Rate	**DSW1 Switch**
9600	3 = off (default)
19200	3 = on

Notes:

- Additional RS-232 settings: 8 data bits, no parity, 1 stop bit.
- The RWFC must be powered down & restarted before a new baud rate assignment can take effect.

6.1.3 - Wiegand Pullup Voltage (switch 4)

Wiegand-based systems typically provide a pullup voltage on data signal lines D0 & D1. If there is no available pullup voltage, the RWFC can be configured to provide it as follows:

Pullup Voltage	**DSW1 Switch**
Disabled	4 = off (default)
Enabled	4 = on

Note:

- By default, output data signals D0 & D1 assume a high impedance state when inactive. During a Wiegand output transaction, the signals are driven to high and/or low output transitions as required to fulfill the data transmission. Once the conversion has completed, the signals return to the high impedance state.
- The high impedance state is especially crucial when there are other device(s) concurrently connected on the D0 & D1 data lines. These may include card readers, access controllers, or other compatible Wiegand items. It is imperative for any attached device to release the data lines (i.e., invoke the high impedance state) after generating a Wiegand transaction so that other devices may enact subsequent transactions.
- The RWFC utilizes 10k resistors to provide a pullup voltage of 4 to 5vdc. If the option is enabled, a pullup voltage is applied when the data lines are inactive (i.e., high impedance).
- Do not attempt to enable pullup voltage if another device already provides it.

6.1.4 - Wiegand Pulse Width (switch 5)

This setting establishes the desired timing for Wiegand pulses. Configure the switches as follows for the desired pulse width:

Pulse Width	DSW1 Switch
50 microseconds	5 = off (default)
100 microseconds	5 = on

Notes:

- In most cases, the default setting will suffice. However, in some instances, a longer pulse width may be required for compatibility with the receiving device.
- Spacing between Wiegand pulses is always 1 millisecond, regardless of the selected pulse width.

6.1.5 - Port Power (switch 8)

The RWFC offers the capability to provide power through the DB-9 connector. This option is useful when the external RS-232 device is port-powered. The output voltage level will be roughly half the input supply voltage. For a 12VDC input, the output port power will be approximately 6VDC.

DB9 Power	DSW1 Switch
Disabled	8 = off (default)
Enabled	8 = on

When port power is enabled, a voltage is applied at DB9 signal outputs DSR (pin 6) and CTS (pin 8). Since the RWFC is a Data Communications Equipment (DCE) device, a null-modem adapter is required when connecting to another DCE device such as a magnetic stripe card reader. With the adapter applied, the RWFC behaves like a DTE device and will apply port power at signals DTR (pin 4) and RTS (pin 7) as indicated in the table below:

RWFC output power	
Without null-modem	With null-modem
DSR (pin 6)	DTR (pin 4)
CTS (pin 8)	RTS (pin 7)

Note: Unless the RS-232 device is port-powered, this option should remain off (disabled).

7 - TB-1 Wiring Connections

The RWFC controller provides a 14-position terminal block with the following screw terminal connections.

Terminal	Description
1	Ground
2	+12VDC in/out (25vdc max)
3	+5V out (200 mA max.)
4	N/A
5	Wiegand D0
6	Wiegand D1
7	Ground
8	Relay contact (N/C)
9	Relay contact (COM)
10	Relay contact (N/O)
11	RS-232 Tx data (also available at DB-9)
12	RS-232 Rx data (also available at DB-9)
13	Ground
14	N/A

Note:

- Only a subset of the above connections will be required in a given application.
- If supplying your own power source at terminals 1 & 2, do not also concurrently provide power with the ac adapter at the barrel connector.
- The maximum allowable input power is +25VDC.

8 - Source Data

The RWFC controller receives source data on the RS-232 port (via DB9 connector). The data is received as standard ASCII character digits. Once received and buffered, this data is converted to the specified format and then promptly transmitted on the Wiegand signal output lines. As an example, consider the following ASCII source string received from the remote device:

00065520

Assuming 26-bit conversion and a default site code of 100, the following composite data will be assembled and formatted by the RWFC for Wiegand output as shown below:

P	**Site Code**								**User Code / ID Number**																**P**
	00100 (64h)								00065520 (FFF0h)																
1	0	1	1	0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1

As a second example, presume 8-bit burst conversion and the following ASCII source string from the remote device:

00000050

The following composite data will be formatted by the RWFC for Wiegand output as shown below:

Burst Data Value							
50 (32h)							
0	0	1	1	0	0	1	0

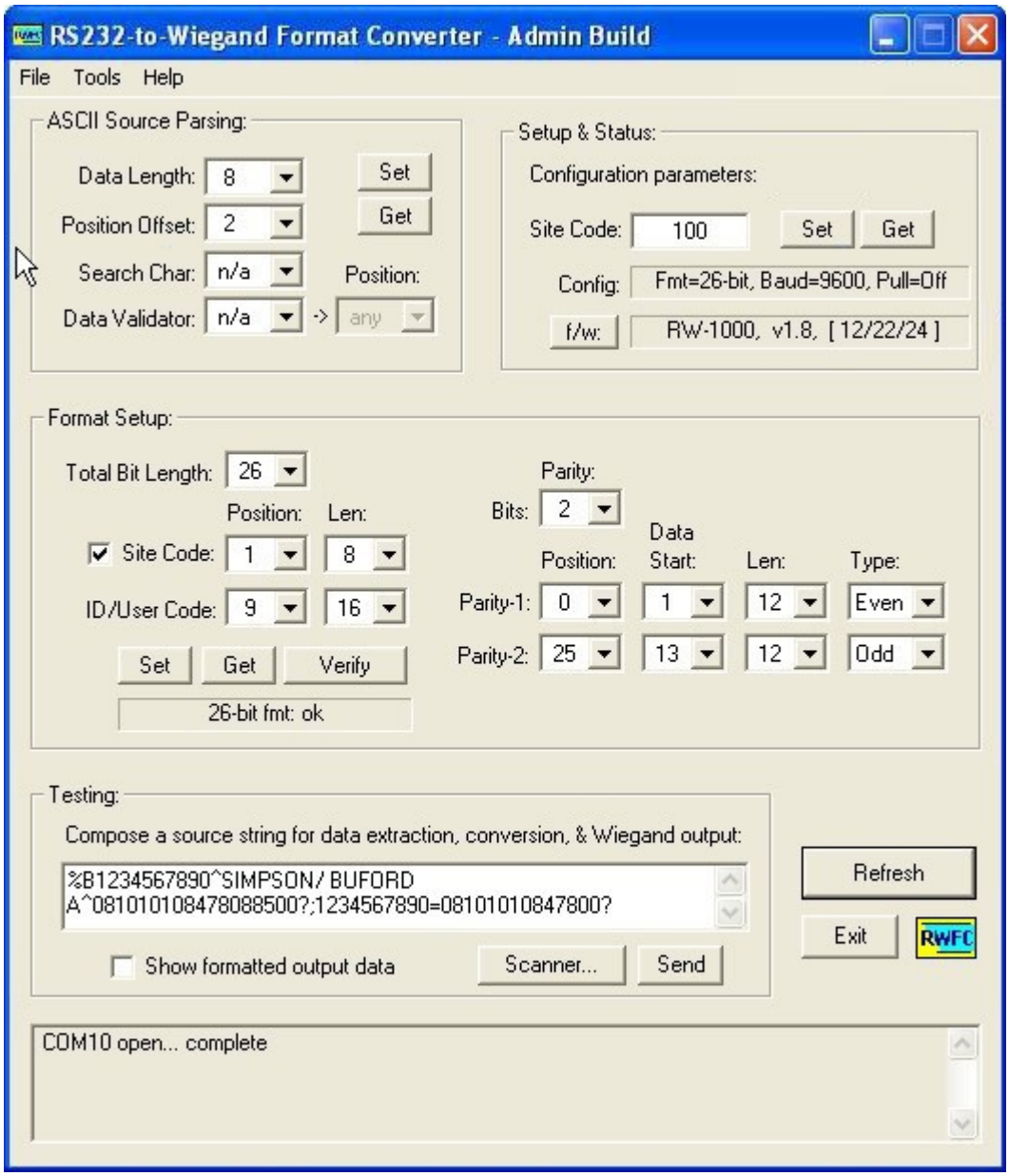
Notes:

- Each received ASCII source string from the host is converted into a composite binary integer prior to being output via Wiegand. For 26-bit Wiegand format, a 16-bit binary value is created (max value = 65535). For 8-bit burst mode, an 8-bit value is created (max value = 255).
- Typically, burst formats contain no site code or parity bits.
- ASCII source strings received from the remote device must be terminated with a carriage-return character (13 dec. or 0D hexadecimal).
- The maximum allowable source data string length is 30 characters.

9 - Windows Host Software

Available RWFC host software for Windows provides functionality for configuring source parsing parameters as well as optional custom formatting. It also provides the capability to issue a user-defined source string for issuing a test Wiegand output. The following is a summary list of software features

- Establish source parsing parameters
- Configure a default site code
- Build any desired custom format (up to 99 bits in length)
- Create user-defined source string and issue Wiegand outputs for testing



These are the requirements for installing the host software on your PC or laptop computer:

- Pentium or compatible CPU
- 8 MB RAM
- 1 MB available hard disk space
- 1 available RS-232 or USB port
- Windows XP or higher

Note:

- Format selection (standard or custom) requires appropriate DIP switch configuration setting. (Refer to “Output Format” for details.)

10 - ASCII Commands

The RWFC controller receives optional ASCII host commands on the RS-232 port (via DB9 connector). Commands are subdivided into the following types:

- Peripheral activation
- Source data parsing
- Wiegand formatting
- Miscellaneous commands

The RWFC is capable of receiving not only host commands but also source data. (Refer to “Source Data” for details.) The two types are easily distinguishable in that commands are prepended with a special three-byte header sequence. The command header is shown below, indicated in both ASCII text and hexadecimal representations:

Command Header	
ASCII Text	**Numeric (Hex)**
\SOH \STX >	01 02 3E

All transactions (host commands and source data) must be appropriately followed by a termination character as shown below:

Terminator	
ASCII Text	**Numeric (Hex)**
\CR	0D

The example below illustrates a complete command sequence and is shown in both ASCII text and hexadecimal representations:

Command Example (Activate Relay)	
ASCII Text	**Numeric (Hex)**
\SOH \STX > ! 0 5 \CR	01 02 3E 21 30 35 0D

Notes:

- The command format contains no spaces. (Spaces are included in the example above only for clarity.)
- All command sequences must be preceded by the special 3-byte header. (Subsequent command listings have omitted the header for the sake of clarity.)
- Commands which assign a parameter setting also allow you to poll/retrieve the current setting. To do this, simply issue the command without parameters.
- All host commands (and source data) must be terminated with a carriage-return character (13 dec. or 0D hexadecimal). (Subsequent command listings have omitted the terminator for the sake of clarity.)

10.1 - Peripheral Activation

Peripheral activation commands are used for activating optional onboard peripherals including relay and/or piezo beeper. Available commands are listed below:

- Activate Relay
- Activate Piezo Beeper

10.1.1 - Activate Relay

The onboard relay (if equipped) may be used to activate an external device or specialized equipment. It can be commanded in one of two available methods: 1) activated for a timed duration, or 2) activated on or off for an indefinite period. The command format is defined below:

Command	Syntax	Example	Comment
Activate Relay	`!xxxxx`	`!00005`	Activate relay for 5 seconds
	`RELAY=n`	`RELAY=1`	Activate relay (on)
Note: Onboard relay has a switching capacity of: 0.4A@125vac / 2A@30vdc.			

Field designators:

- xxxxx = 5-digit relay activation period in seconds
- n = activation designator: 0=off, 1=on

10.1.2 - Activate Piezo Beeper

The onboard piezo beeper (if equipped) may be used for audible annunciation and can be commanded in one of two available methods: 1) cyclically annunciate on and off for a specified number of reptititions, or 2) annunciate once for a specified duration. The command formats are defined below:

Command	Syntax	Example	Comment
Activate Beeper	`&nn`	`&15`	Sound the beeper 1.5 seconds
	`&xrrr`	`&x005`	Cycle the beeper (on & off) 5 times

Field designators:

- nn = 2-digit value indicating the activation period in hundreds of milliseconds
- rrr = 3-digit value indicating the number of beeper repetitions to perform

10.2 - Source Data Parsing

Source data parsing allows the RWFC to properly locate and extract data characters from the received ASCII serial data string. It is necessary when the string contains extraneous or unused characters which are to be ignored such as headers, trailers, prepended zeros, etc. (For instance, magnetic stripe readers are known to output additional characters which are used as track delimiters.) There are two available data parsing methods for handling this situation:

- Set data start index & length
- Set search character

Additionally, an optional validation character may be applied for the purpose of validating an extracted data field:

- Set validation index
- Set validation character

Notes:

- The search character may be used in combination with the data start index. When a search character is defined, the start index takes precedence. That is, searching for the specified character will not begin until a minimum number of source character(s) have been received, identified by: start index + 1.
- The start index defines the expected location of the desired data (or the minimum position within the composite source string to invoke an optional search character, if defined).
- The RWFC can accept a composite source string of virtually any length. However, the number of extracted data characters should be limited to 10 digits. (This is the maximum length which can be converted and stored into a four-byte unsigned integer, i.e., 4,294,967,295.)
- Extracted data character(s) must contain only numeric digits '0' - '9'.

Available commands are listed below:

- Set Data Start Index
- Set Data Length
- Set Search Character
- Set Validation Index
- Set Validation Character

Note that each command must be terminated with a carriage-return character (13 dec. or 0D hexadecimal).

10.2.1 - Set Data Start Index

This command sets the start index for locating the ID/data segment (sub-string) within the received ASCII source string. It is used in conjunction with Set Data Length to locate and extract pertinent data characters from the string. (The default index is 0, meaning the first character is the presumed to be the beginning location for the data.) The command format is defined below:

Command	Syntax	Example	Comment
Set Start Index	SDSnn	SDS09	Set the start index at position 9 (i.e., the 10th character)

Field designators:

- nn = start index

10.2.2 - Set Data Length

This command sets the length (in character digits) of the ID/data segment (sub-string). It specifies the number of relevant characters within the received ASCII source string. It is used in conjunction with Set Data Start Index to locate and extract pertinent data characters from the string. The command format is defined below:

Command	Syntax	Example	Comment
Set Data Length	SDLnn	SDL08	Set a 8-digit data length

Field designators:

- nn = data length (digits)

Notes:

- This command does not designate the length of the source string. Instead, it specifies the length of the ID/data segment within the source string, though both parameters may be identical.
- The maximum allowable data length is 10 digits.
- The maximum source data string length is 30 characters.

10.2.3 - Set Search Character

This command defines an optional search character within the received composite string for locating the data field. (The desired source capture data field is presumed to begin immediately following the search character, if found.) The command format is defined below:

Command	Syntax	Example	Comment
Set Search Character	`SSCnnn`	`SSC059`	Set semicolon as the desired search character.

Field designators:

- `nnn` = decimal representation for the desired ASCII search character

Note:

- By default, search parsing is disabled (i.e., search character = 0).
- To clear/revoke the search character, simply issue the command with a search parameter of 0, i.e., “SSC000”.

10.2.4 - Set Validation Character

This command defines an optional validation character for validating the data field within an ASCII source string. Similar to the search character defined previously, the validator is useful for discriminating source data lines. It is ideal for applications where source strings contain data fields that are properly formatted and positioned, but are not necessarily relevant for Wiegand conversion. The command is applied in conjuction with Set Validation Index which defines an expected position for the validation character.

As an example, consider two realtime ASCII source strings received via RS-232 from a gate control keypad where visitors are required to enter a 4-digit access code. The first string below was issued by the keypad as the result of an invalid code entry attempt, while the subsequent string indicates a valid access. (The RWFC is presumed to have been configured to accept a data field offset of 5 characters and length of 4 numeric digits.)

- `"CODE 1234 ACCESS ..FAILED"`
- `"CODE 3487 ACCESS ..PASSED!"`

Without a validator, the RWFC will accept data from **both** source input lines, converting and applying the two codes to the Wiegand bus. However, an established validator effectively eliminates the unwanted entry by examining a follow-on character (validator) that is unique only to valid source strings. One possible choice for the above example is the exclamation "!" character. Another possibility is 'P' which occurs in the word, `"PASSED"`.

The command format is defined below:

Command	Syntax	Example	Comment
Set Validator	`SVCnnn`	`SVC033`	Set exclamation “!” as the validator.

Field designators:

- `nnn` = decimal representation for the desired ASCII validation character

Notes:

- By default, validation is disabled (i.e., validator = 0).
- To clear/revoke the validator, simply issue the command with a parameter of 0, i.e., “SVC000”.
- The validator is expected to occur after/following the relevant data field. Any incidental occurrence of the validator that precedes the data field will be ignored. (See also Set Validation Index)
- Source validation was added as a firmware revision (v1.8). It is native to units manufactured December 2024 and later.

10.2.5 - Set Validation Index

This command sets the index for locating a validation character within the received ASCII source string. It is used in conjunction with Set Validation Character for the purpose of validating the data field. The index value defines the validator's expected position within the source string. (Validator position must be located past or beyond the data field.) The default index is 255 which indicates a "floating" validator, meaning the validator may appear at *any* position beyond the data field. The command format is defined below:

Command	Syntax	Example	Comment
Set Validator Index	`SVSnnn`	`SVS014`	Set the validator index at position 14 (i.e., the 15th character)

Field designators:

- nnn = validation index

Notes:

- To assign a floating position index, apply a value of 255, i.e., “SVS255”.
- The validator is expected to occur after/following the relevant data field. Any incidental occurrence of the validator that precedes the data field will be ignored. (See also Set Validation Character.)
- Source validation was added as a firmware revision (v1.8). It is native to units manufactured December 2024 and later.

10.3 - Wiegand Formatting

There are two available Wiegand output format types:

- Standard
- Custom

Standard formats are preexisting, non-customizable outputs and include 26-bit and 8-bit types. Custom formats are user-created, allowing you to define practically any desired output. (Refer to “DIP Switch Settings” to select the desired format.)

Wiegand formatting commands are provided for optional host access and used for the following purposes:

- Retrieving active format parameter settings
- Creating a custom-defined output format

Commands are provided for retrieving active format parameter settings. The active format may be standard or custom-defined. These commands are described in subsequent sections of this manual.

If desired, you may create a custom-defined output format. (To utilize a custom format, the appropriate configuration option must be enabled. Refer to “DIP Switch Settings” for details.) Multiple commands are required to define the entire set of parameters which comprise a format. The following parameters must be specified:

- Total bit length
- Parity mask (bit-location, data-start index, count, even/odd)
- Data length (start index, length)
- Site code (start index, length)

Available Wiegand formatting commands are listed below:

- Format clear
- Set Format Identifier
- Set Parity Mask
- Set Data Field
- Set Site Code Field
- Verify Format
- Assign Default Site Code

Command details including descriptions and syntax are provided in subsequent sections.

For those desiring to create custom formats, it is essential to understand individual field parameters and their relative bit positions. To illustrate this, a sample 26-bit Wiegand string is shown below. (Note that the designated bits will be output to the receiving device in order from left to right, beginning with bit 0.)

P	**Site Code**								**User Code / ID Number**																**P**
1	100 (64h)								65520 (FFF0h)																1
1	0	1	1	0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Notes:

- Bit fields are numbered in order from left to right, beginning with 0, the most-significant bit of the composite output.
- "P" indicates a parity . (The left-most indicator denotes even parity, while the right-most indicator denotes odd parity.)
- Parity and site code are considered optional fields for custom formats.
- Up to two parity indicators may be specified in custom formats.
- Custom formatting must be enabled using the appropriate option switch. (Refer to "DIP Switch Settings" for details.)
- Commands which assign a parameter setting also allow you to poll/retrieve the current setting. In most cases, retrievals are context sensitive. That is, they are applicable to the currently selected format, whether custom or standard. (Refer to "DIP Switch Settings" for details.)
- All commands must be terminated with a carriage-return character (13 dec. or 0D hexadecimal).

10.3.1 - Format Clear

This command clears all custom formatting parameters including length, parity, data specifications and site code information. The command format is defined below:

Command	Syntax	Example	Comment
Format Clear	WGCL	WGCL	Clear all custom formatting parameters.

Field designators:

- none

Note:

- This command may be used to clear preexisting custom format parameters prior to defining a new custom format.
- The command does not apply to standard formats.

10.3.2 - Set Format Identifier

The format identifier establishes the high-level format parameters including overall length and number of parity indicators. The format length defines the total number of bits which comprise the format. The parity count defines the number of single-digit parity indicators contained in the format. The command format is defined below:

Command	Syntax	Example	Comment
Set Format	WGFLnnp	WGFL262	Set 26-bit format length; include 2 parity bits

Field designators:

- nn = format length (bits)
- p = number of parity indicators

Note:

- The maximum format length is 99 bits
- Up to 2 parity indicators may be defined

10.3.3 - Set Parity Mask

Parity provides an indication of data validity. This command establishes a mask, defining specific bits in the format that are used in the parity check. You may specify up to two parity masks (see "Set Format Identifier") and each may be designed as even or odd. The following elements comprise the mask:

- location - defines the bit index where the single-digit parity field resides
- data-start - defines the first bit that will be used in the parity check
- count - defines the number of bits to be used in the parity check
- type - defines the type of parity check to perform: even or odd

The command format is defined below:

Command	Syntax	Example	Comment
Set Parity Mask	WGPnllsscct	WGP1000112E	Set parity-1 at bit 0; compute on bits 1-12, even
		WGP22513120	Set parity-2 at bit 25; compute on bits 13-24, odd

Field designators:

- n = designator (1 or 2)
- ll = bit index defining where the single-digit parity field resides
- ss = starting data bit that will be used in the parity check
- cc = number of data bits to be used in the parity check
- t = type of parity check to perform: 'E' = even, 'O' = odd

As an illustration of parity masking, consider the 26-bit Wiegand data sample shown in the table below. The format requires that the total number of '1' bits located in the first 13 bit fields (identified in green) must be even. Therefore the even parity indicator (located in the left-most bit field) is assigned a '1' (to yield an even result). Likewise, the total number of '1' bits in the last 13 bit fields (identified in red) must be odd. Therefore, the odd parity indicator (located in the right-most bit field) is also assigned a '1' (to yield an odd result).

P	Site Code								User Code / ID Number																P
	00100 (64h)								00065520 (FFF0h)																
1	0	1	1	0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Notes:

- By default, the RWFC attempts to extract the maximum number of data characters from the ASCII string.
- Unless specified otherwise, index values are zero-based.
- Bit fields are numbered in order from left to right, beginning with 0, the most-significant bit of the composite output.
- Parity is an optional field element. If undesired, simply omit the command.
- For even parity, the total number of "1" bits in relevant bit locations (including the parity bit) must be even. Therefore, the parity bit must be assigned to "0" if the total number of "1s" is even. Otherwise, if odd, a "1" is assigned to force the count to be even. For odd parity, the total number of "1" bits in relevant bit locations (including the parity bit) must be odd. Therefore, the parity bit must be assigned to "0" if the total number of "1s" is odd. Otherwise, if even, a "1" is assigned to force the count to be odd.
- A maximum of two parity fields may be defined.

10.3.4 - Set User Code Field

This command establishes the location and length of the user code (ID number) field. (For applicable format definitions, this field designates the user code / ID number.) The command format is defined below:

Command	Syntax	Example	Comment
Set User Code	WGUssll	WGU0916	Set 16-bit user code length, starting at bit 9

Field designators:

- ss = start data index
- ll = length of data field (32 bits max)

Notes:

- The maximum allowable length of the field is 32 bits.
- Unless specified otherwise, index values are zero-based.

10.3.5 - Set Site Code Field

This command establishes the location and length of an optional site code field. The command format is defined below:

Command	Syntax	Example	Comment
Set Site Code	WGSssll	WGS0108	Set 8-bit site code, starting at bit 1

Field designators:

- ss = site code start index
- ll = length of site code field (32 bits max)

Notes:

- The maximum allowable length is 32 bits.
- Unless specified otherwise, index values are zero-based.
- The site code is an optional field element. If undesired, simply omit the command.
- This command is not to be confused with “Assign Default Site Code”.

10.3.6 - Verify Format

This command evaluates the current format configuration to determine its validity. If the format is invalid, an error message will be reported to the host. The command format is defined below:

Command	Syntax	Example	Comment
Verify format	`WGVF`	`WGVF`	Evaluate the current format.

Field designators:

- none

Notes:

- Though the command is intended primarily for use with custom-defined formats, it may also be applied to standard formats.

10.4 - Miscellaneous Commands

The RWFC can also accept the following miscellaneous commands:

- Assign Default Site Code
- Retrieve Configuration Settings
- Enable Output Status Messages

Command details including descriptions and syntax are provided in subsequent sections. (Please note that each command must be terminated with a carriage-return character (13 dec. or 0D hexadecimal).

10.4.1 - Assign Default Site Code

This command establishes a default site code value for insertion into the Wiegand output bit stream. It is applicable only if the output format specifies a site code. The command format is defined below:

Command	Syntax	Example	Comment
Assign Default Site Code	`WDSnnnnn`	`WDS00100`	Assign default site code value = 100

Field designators:

- nnnnn = 5-digit site code value

Notes:

- A default site code is needed only if the output format utilizes a site code.
- This command is not to be confused with “Set Site Code”.

10.4.2 - Report Configuration Settings

This command retrieves DSW1 DIP switch configuration settings. (Refer to “DIP Switch Settings” for details.) The command format is defined below:

Command	Syntax	Example	Comment
Report Config. Settings	`WRCS`	`WRCS`	Retrieve DSW1 configuration settings

10.4.3 - Enable Output Status Messages

The RWFC provides a status message output option which may be used for purposes of testing and/or troubleshooting. When enabled, the controller will issue a series of ASCII characters for each received source data transaction, providing a visual representation of the corresponding Wiegand transaction. The output will be sent on the RS-232 port (via DB9 connector). It provides a a simple and convenient means of viewing and confirming the Wiegand output bit stream at your computer or similar host device. The command format is defined below:

Command	Syntax	Example	Comment
Enable Output Status Messages	WGTE	WGTE	Enable output status messages
Disable Output Status Messages	WGTD	WGTD	Disable output status messages

As an example, consider a default site code of 100 (64h) and the following ASCII source data (ID number) from the host:

00065520

Assuming the specified conversion format is 26-bit Wiegand, the corresponding data will be formatted as shown below:

P	Site Code								User Code / ID Number																P
	00100 (64h)								00065520 (FFF0h)																
1	0	1	1	0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

A string of twenty-six digits will be issued to the host as an output status message:

ID Number	Site Code	Output Status String
00065520	100	1 0 1 1 0 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1

Note:

- Output status messages will adversely affect timing of Wiegand output signals and should not be used under normal operating conditions.

11 - Care and Maintenance

The product requires no special care or maintenance other than protection from potential damage due to mechanical shock, electrical surges, and vandalism. It is also vital to shield the hardware from exposure to the elements, hazardous environments, and otherwise extreme conditions.

12 - Warranty

This product is warranted for a period of 1 year from the date of purchase and is guaranteed to be free from defects. The warranty covers the entire unit, except if any part or component has been modified or otherwise converted from its original form. The warranty does not cover damage or failure due to neglect, improper use, or unshielded exposure to moisture, power surges, hazardous environments and the like.

IMPORTANT: The customer is responsible to provide protection against potential over-voltage situations including power surges, spikes, and lightning strikes. The use of adequate surge protection is recommended.

13 - Contact Information

Should you have any questions or comments please contact us :

Delahoussaye Consulting
website: http://www.kadtronix.com
email: support@kadtronix.com
phone: 321-757-9280