APPLIED WIRELESS ID

**Applied Wireless Identifications Group, Inc.**
18300 Sutter Blvd., Morgan Hill, CA 95037 • Voice 1-408-825-1100 • Fax 1-408-782-7402 • www.awid.com

# Technical Reference

## AWID Readers – Code Output

## Translating Wiegand and RS-232 Data

*When preparing a system that uses AWID's readers as input devices, the developer may need to relate (a) codes for a serial data interface with (b) Wiegand-type codes programmed into AWID's credentials. This memo shows simple manipulation for translating these data in both directions.*

### Encoding in Credentials

The codes that AWID programs into its credentials (cards, tags and wafers) are specified by the customers in each purchase order. The form of the codes is generally referred to as "Wiegand-type" code. This defines a code that is commonly 26 binary bits in length. It is made up of a "site code" (or "facility code" or "company code"), followed by an "ID number" (or "serial number" or "sequential number"). In the Security industry's standard 26-bit format, the code is a string of 26 binary bits –

Bit no. 1 = error-checking bit, even parity for code bits no. 1 through no. 13.
Bits no. 2 through 9 = site code (8 bits long), with decimal values between 001 and 255.
Bits no. 10 through 25 = ID number (16 bits long), with decimal values between 00 001 and 65 535.
Bit no. 26 = error-checking bit, odd parity for code bits no. 14 through 26.

The 26-bit standard code format, designated 26-bit-STD, is an open code. It may be ordered by all AWID customers.

Other bit lengths, supported by AWID's readers, can be programmed into the AWID credentials. There is another open code, designated 37-bit-AWI. Other formats, with more than 26 bits, are proprietary codes that are reserved for orders from particular customers of AWID.

### Wiegand Data Interface

All AWID readers have the "Wiegand" data interface, for electrical connection to the host system's controller. The code data, read from the credential, are transmitted from the reader's data register either instantaneously or when requested by the controller (using the "Hold" control line that is built into most AWID readers). This interface uses three binary data lines – one for the binary-zero bits in the code's binary string, one for the binary-one bits, and one for the data-common return circuit from the controller's reader port back to the reader. In the proximity (LF) and high-frequency (HF) readers' cable these wires are labeled "Data-0", "Data-1" and "Ground". In the UHF readers' cable they are labeled "Data-0", "Data-1" and "Data Common".

### RS-232 Data Interface

All AWID readers except the Model SR-2400 proximity (low-frequency) reader have also the RS-232 data interface. This is the popular "serial" interface that was the standard input to personal computers before USB ports became standard. (RS-232-to-USB adapters are available from RadioShack and Best Buy and Security suppliers. They handle this conversion very nicely.) The readers' interface uses three wires (separate from the Wiegand interface except for Data-Common) labeled "Transmit Data" (TD), "Receive Data" (RD), and "Ground" or "Data Common" (GND).

### Formats of These Data Outputs

The readers that have both Wiegand and RS-232 data interfaces use different wire sets in the readers' integrated cable. For the **Wiegand** interface, the wires are green, white, and black or blue. For the **RS-232** interface, the wires are orange, violet, and black or blue. The reader's output on these two interfaces is essentially simultaneous.

The **Wiegand** output is a string of binary bits on Data-0 and Data-1 lines, usually a total of 26 bits (depending on the code in the credential).

The **RS-232** output is a string of 80 binary bits on Transmit Data, of which the first 72 bits contain the data encoded in the credential. The string concludes with the 4-bit characters for Carriage Return and Line Feed. The host system generally breaks this 80-bit string into 20 4-bit groups, and then displays this as the hexadecimal characters for the 4-bit groups. Only the 18 characters for the numeric data are visible in the display. In the 26-bit-STD format, there are 7 "real" hexadecimal characters for the 26 binary bits, followed by 11 fill zeros – total 18 hex characters.

## Relationship Between These Data Outputs

These Wiegand and RS-232 data have a definite relationship between each other. The code that is observed for either Wiegand or RS-232 can be translated to the other interface by a fixed routine.

## Translating Wiegand to RS-232

Write the data on separate lines, (a) through (d). This example is for a credential with 26-bit code.

(a) Decimal values for (Parity-Even on bits 1-13) (Site Code) (ID number) (Parity-Odd on bits 14-26).

(b) Binary equivalent for these 4 decimal values. The number of bits is (PE=1) (SC=8) (ID=16) (PO=1); total = 26 bits.

(c) Split the 26-bit string into 7 groups of 4 bits.

(d) Write the hexadecimal character for each group of 4 bits.

These 7 hex characters are the first 7 of 18 characters that appear in the serial display.

***Example***: (The decimal values are from the example in AWID's Technical Reference "Readers – RS-232 Interface".)

(a) (PE=_) (SC=123) (ID=12345) (PO=_); parity bits are calculated after seeing the binary string in step (b)

(b) 0 01111011 0011000000111001 1; removing spaces, 00111101100110000001110011

(c) 0011 1101 1001 1000 0001 1100 1100 (including 2 fill-zeros at the end of the 7th character)

(d) 3 D 9 8 1 C C; displayed as 3 D 9 8 1 C C 0 0 0 0 0 0 0 0 0 0

## Translating RS-232 to Wiegand (as Printed on the Credential)

Write the data on separate lines, (a) through (e). This example is for the same credential as in Wiegand-to-RS-232, above.

(a) Hex characters. Number of "real" characters = number of binary bits divided by 4, rounded up to whole number.

(b) 4 binary bits for each of these hex characters.

(c) Write the binary groups without spaces. Drop the last 2 bits, which will be zeros for a 26-bit code.

(d) Split the binary bits into groups of (1 bit) (8 bits) (16 bits) (1bit); total = 26 bits.

(e) Write the decimal equivalent for each of these groups of bits.

***Example***:

(a) 3 D 9 8 1 C C; these are the first 7 characters of the displayed 3 D 9 8 1 C C 0 0 0 0 0 0 0 0 0 0

(b) 0011 1101 1001 1000 0001 1100 1100

(c) 00111101100110000000111001100; dropping the last 2 zero bits = 00111101100110000001110011

(d) 0 01111011 0011000000111001 1

(e) 0 123 12345 1; this is the original Wiegand code as printed on the credential, with appropriate parity values

## Notes

1. This RS-232-to-Wiegand translation procedure requires knowing the code format of the AWID credential being read by the reader. If the number of "real" hexadecimal characters in the 18-character hex string is 7, the code in the credential is probably format 26-bit-STD.

2. AWID's Model LR-KIT-0-0 Installation Kit contains a short **adapter cable** that is useful when connecting the reader's RS-232 data lines to a PC's serial port, or to an RS-232-to-USB adapter. AWID's adapter cable has 3 spring clips on orange, violet and blue wires at one end, and a 9-pin "D" female connector at the other end.

3. "RS-232" data output from AWID's readers is not true RS-232 in one respect. The amplitude of data pulses is determined by the voltage that is available on the Transmit Data terminal of the host controller's interface. Generally pulse amplitude of about -5 volts DC or greater is sufficient to register in the controller's input circuit.

4. Microsoft's Windows HyperTerminal program is a convenient way to see the reader's RS-232 data output in a string of 18 hexadecimal characters as described in this memo.

5. The "EVAL" reader in AWID's LR-KIT-0-0 Installation Kit is not usable when the task requires either Wiegand or RS-232 data interface from the reader. The "EVAL" reader has no usable data output. Its data interface wires (green, white, orange, violet and blue) are not accessible for external connection.

6. Decimal values for powers of 2 are –

| | | |
|---|---|---|
| $1^{st}$ power = 2 | $6^{th}$ power = 64 | $11^{th}$ power = 2 048 |
| $2^{nd}$ power = 4 | $7^{th}$ power = 128 | $12^{th}$ power = 4 096 |
| $3^{rd}$ power = 8 | $8^{th}$ power = 256 | $13^{th}$ power = 8 192 |
| $4^{th}$ power = 16 | $9^{th}$ power = 512 | $14^{th}$ power = 16 384 |
| $5^{th}$ power = 32 | $10^{th}$ power = 1 024 | $15^{th}$ power = 32 768 |

## Reference

- Technical Reference "Notes on the Name 'Wiegand'".
- Technical Reference "Format Options for Customers".
- Technical Reference "Readers – RS-232 Interface"
- Product sheet "LR-KIT Installation Kit".

(continued on page 4)

## Calculating Decimal Value of a Binary Data Field

This calculation can be used with data fields with either 8 bits or 16 bits of binary data. The industry-standard 26-bit Wiegand-type format has these fields – the facility (site) code, with 8 bits, and the ID number, with 16 bits.

Procedure:

1. Double-click the table (below) to convert it to an Excel spreadsheet.
2. The binary data fields should be written from left to right, most-significant-bit first.
3. In the cells of Column C, write the binary value (1 or 0) for each of the bits in the binary string.
4. After entering each cell's number, press the Enter key to move down to the next cell. The calculation occurs then.
5. The Value on that line in Column E, and the SUM at the bottom of Column E, both change when you press Enter.
6. After you enter all 1's and 0's in Column C, the decimal value of the data field is at the bottom of Column E.

| A | B | C | D | E |
|---|---|---|---|---|
| Bit Number (Left to Right) for 8-bit data field | Bit Number (Left to Right) for 16-bit data field | Binary Value for This Bit (binary-1 or binary-0) | Decimal Value for this Bit Number | Weighted Value (Column C X Column D) |
| --- | 1 | 0 | 32,768 | 0 |
| --- | 2 | 0 | 16,384 | 0 |
| --- | 3 | 0 | 8,192 | 0 |
| --- | 4 | 0 | 4,096 | 0 |
| --- | 5 | 0 | 2,048 | 0 |
| --- | 6 | 0 | 1,024 | 0 |
| --- | 7 | 0 | 512 | 0 |
| --- | 8 | 0 | 256 | 0 |
| 1 | 9 | 0 | 128 | 0 |
| 2 | 10 | 0 | 64 | 0 |
| 3 | 11 | 0 | 32 | 0 |
| 4 | 12 | 0 | 16 | 0 |
| 5 | 13 | 0 | 8 | 0 |
| 6 | 14 | 0 | 4 | 0 |
| 7 | 15 | 0 | 2 | 0 |
| 8 | 16 | 0 | 1 | 0 |
| | | | SUM of Column E >> | **0** |