Technical Reference

Readers – RS-232 Interface

Instructions for Installers and Programmers

A. General Information

1. The following instructions apply to all of AWID's readers -- low frequency (proximity), high frequency and ultra-high frequency -- *except* the SR-2400 proximity reader. All AWID readers except the SR-2400 contain both Wiegand and RS-232 data interfaces; the SR-2400 has Wiegand interface only.

2. In the Production process, AWID programs the credentials (encoded cards, tags and wafers) that are read by AWID's LF and UHF readers with between 26 bits and 50 bits of binary code, in a variety of bit formats. Wiegand-type formats typically contain two data fields, and two parity error-checking bits. Data fields are usually a site code (or facility code), a card ID number, and perhaps special fields such as issue number, company code, etc.

3. Each field contains only binary data. The binary data in each data field start with the most-significant bit (MSB), and end with the least-significant bit (LSB).

4. The decimal equivalent of these binary data may be printed during production by AWID on the surface of the credentials, or on a removable paper label. The common sequence of printed data on cards is: *first*, site code (or facility code); *second*, card ID number; and *third*, an identifier like "AWID" followed by the number of bits in the format – all on one printed line.

5. The reader communicates with a credential that is within range of the reader. If the reader recognizes the card, tag or wafer as a valid AWID credential, and if redundant reads yield the identical bit stream, the reader then transmits the credential's code in a burst of data almost simultaneously on the Wiegand interface's Data-0 and Data-1 lines and on the RS-232 Transmit Data (TD) line. This memo describes the RS-232 data output from the reader.

6. Regardless of the number of bits in the credential's code, the reader always sends out 20 bytes – 18 for data, then CR and LF. These bytes are converted in the reader to the RS-232 protocol.

7. Significant data characters are transmitted first. Trailing zeros are added to fill the 18 data bytes. For example, if the credential's code uses the 26-bit format, the RS-232 transmission is 7 bytes for "real" data, followed by 11 fill zero bytes.

8. This process limits the application to transmission and reconstruction of *numeric* data only (no alphabetical characters or symbols). **Note:** The programmer who uses this memo *must* know the bit format of the binary data in AWID's credentials. This includes location of parity bits, number of data fields, uses of the data fields, and bit positions assigned to each data field.

9. Power must be applied to the reader in the usual way. See the Installation Sheet or Instructions for the reader's power requirements.

10. AWID's readers comply with the RS-232 cable length standard of 75 feet. Actual cable length may vary with the equipment used.

B. Wiring the Reader to a PC

To transmit data from an AWID reader to a PC's serial communications port, connect these wires * –

For All LF (Proximity), HF and uAxcess Readers	For All LR-Series UHF Long-Range Readers
Reader power ground (black wire)......to DC power supply -	Reader power negative (black wire)... to DC power supply -
Reader TD (violet wire) ....to PC RD (pin 2)	Reader TD (orange wire) .... to PC RD (pin 2)
Reader RD (orange wire) ....to PC TD (pin 3)	Reader RD (violet wire) .... to PC TD (pin 3)
Reader data common (*black* wire) ......to PC GND (pin 5)	Reader data common (*blue* wire) ...... to PC GND (pin 5)
Reader power positive (red wire)........to DC power supply +	Reader arming (yellow wire) .... to Reader negative (black wire)
	Reader power positive (red wire)....... to DC power supply +

* For a PC with only USB ports, use a serial adapter with a 9-pin "D" connector. Serial-to-USB adapters are available at RadioShack and Best Buy and local computer suppliers. See Figure 1.

Alternative 2-wire connection: When the reader's RD line is not connected, the reader's TD line transmits data at TTL logic levels. The pulses are positive-going from about 0 volts to +5 volts. If the host device can receive input data as standard TTL pulses, this 2-wire connection may suffice. This eliminates the need to use the host's TD line as a source of voltage for the reader's RS-232 output pulses. Other wiring in Section B (above) remains the same.

C. Configuring the Computer's Communications Port

1. Communications port selection: For PCs with a serial port, the port is usually COM1 or COM2. For PCs with only USB ports, this may be designated as COM Power or COM5.
2. RS-232 data format: Each byte consists of 1 start bit, 8 data bits, and 1 stop bit.
3. Data rate: For all readers except the UHF uAxcess series – 9600 bits per second. For uAxcess readers – 57.6 kilobits per second.
4. Parity: No parity check.
5. Emulation: ANSI (formerly ASCII).
6. Flow control: None.
7. For 3-wire connection to host device: AWID's reader does not generate data pulses at full RS-232 amplitude internally. Data pulse voltage is related to the voltage supplied by the host.
   - If the host's TD line supplies -12 VDC to the reader's RD line, the voltage measured on the reader's TD line is about -9 VDC to +9 VDC.
   - If the host's TD line supplies -9.8 VDC, the RS-232 pulses are about -9 VDC to +5 VDC.
   - If the host's TD line supplies -5.5 VDC, the RS-232 pulses are about -5.5 VDC to +5 VDC.

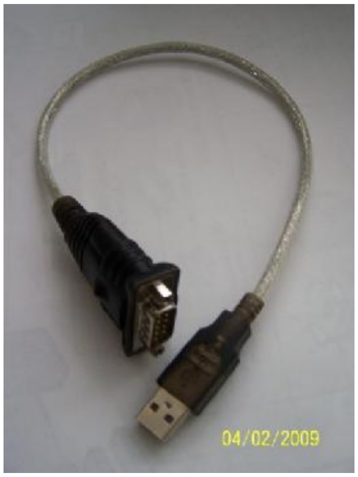

Figure 1. Adapter, serial (9-pin male) to USB. (See Section B.)

8. Connecting the port: In the PC's default state, the port may be disconnected. To open the port and start HyperTerminal – (a) open the menu for the Call tab at the top of the window; (b) select "Start Call".

D. Data Preparation and Transmission

1. The AWID reader reads the credential's code. The reader registers the string of bits.

   Example for a card with 26-bit Wiegand industry-standard format:
   Site code (or facility code) is decimal-123 (binary-01111011 in bit positions 2 to 9).
   Card ID number is decimal-12345 (binary-0011000000111001 in bit positions 10 to 25).
   Parity: Bit position 1 is used to force even parity for bits 1 through 13 (= binary-0 in this example);
   Bit position 26 is used to force odd parity for bits 14 through 26 (= binary-1 in this example).

2. The reader breaks the binary data string into sequential groups of 4 bits. The groups contain bits number 1-4, 5-8, 9-12, 13-16, 17-20, 21-24, 25-28, 29-32, 33-36, … 65-68, 69-72 (a total of 18 bit groups). Trailing zeros fill the groups as necessary.

   Example:
   binary - 0011 1101 1001 1000 0001 1100 1100 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

3. The reader converts these 18 4-bit binary groups to a string of 18 equivalent hexadecimal characters in the range 0-to-9 and A-to-F.

   Example:
   hex - 3 D 9 8 1 C C 0 0 0 0 0 0 0 0 0 0 0

4. The reader considers each of these hexadecimal characters as ASCII code, and then converts the ASCII code to its equivalent hexadecimal code, and appends Carriage Return (hex-0D) and Line Feed (hex-0A).

   Example:
   hex - 33 44 39 38 31 43 43 30 30 30 30 30 30 30 30 30 30 30 0D 0A

5. The reader hardware converts this hexadecimal code to RS-232 and transmits the data on the serial TD line as 40 4-bit binary groups. The host system stores the RS-232 data in a register.

   Example:
   binary - 0011 0011 0100 0100 0011 1001 0011 1000 0011 0001 0100 0011 0100 0011 0011 0000 0011 0000 . . . 0000 1101 0000 1010

E. Reconstruction of Data in the Host System

1. The RS-232 data stored in the host's register are grouped in 20 8-bit bytes.

   Example:
   binary - 00110011 01000100 00111001 00111000 00110001 01000011 01000011 00110000 00110000 . . . 00001101 00001010

2. Host software reads these bytes as pairs of 4-bit BCD numbers. Software drops the non-significant trailing zeros, and also CR and LF.

   Example:
   hex - 33 44 39 38 31 43 43

3. Host software converts these pairs of characters to the equivalent ASCII code.

   Example:
   ASCII - 3 D 9 8 1 C C

4. The binary code for these hex characters is the same as the original binary string in step D.1 (plus 2 trailing zeros for fill).

   Example:
   binary - 0011110110011000000111001100

5. Using the known bit format for the data in the AWID card, this binary string can be converted to the original decimal values and parity bits (after dropping the last two binary-zeros, to reduce the total number of bits to 26 for this example).

6. The program uses the reconstructed data as required by the application.

F. Displaying the RS-232 Data on a PC

1. *For LR-series long-range readers* – Connect the reader's yellow wire to the black wire. This arms the reader for RF transmission to the tag. To display RS-232 data, run AWID's "LRRSettings" program.
2. *If the PC has been used to sync with a PDA* – Click on the icon for "Last HotSync" on the task bar at the lower right corner of the *Desktop* screen. When the pop-up menu appears, click on "Exit".
3. Connect the reader to the PC's serial port, using the wiring in Section B (page 2).
4. *With an older* PC -- Run Microsoft HyperTerminal on the PC. The path in Microsoft Windows is:
   Start button → Programs → Accessories → Communications → HyperTerminal
   If necessary, click on HyperTrm.exe.
5. Configure the PC's serial port. See Section C on page 2.
6. Hold a compatible AWID credential in front of the reader.
7. The credential's code is listed on the monitor in a column, with 18 hexadecimal characters on each line. The data for the credential are the same as in step D.3. To make new data easily visible as they enter at the bottom of a full column, use a *different* credential in front of the reader.