GENERAL:

This document describes the data format for the Andover Data Interface to Automation Displays Z-Card. This interface allows a host computer (the Andover system) to control LEDs and to receive switch status messages over a serial link to an annunciation and control panel.

The annunciator panel requires only the connection to AC power and an RS-232 data connection for basic operation.

A DIP switch on the Z-Card determines the baud rate and the switch reporting protocol. The reset switch pushbutton must be pressed after the DIP switch setting is changed.

Pressing the reset switch resets the processor and clears the annunciator's LED displays.

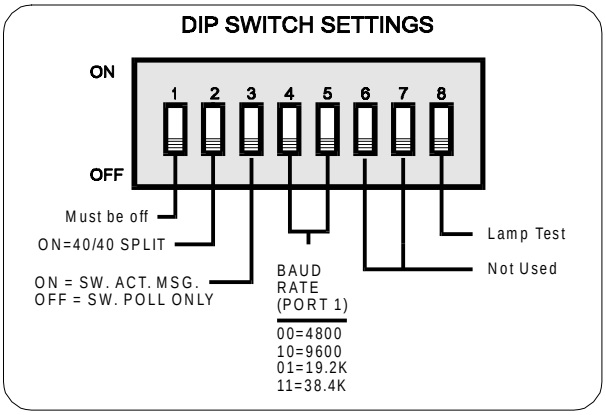
COMMUNICATION FORMAT:

DATA CONNECTION:
RS-232 IN
RS-232 OUT
DC COMMON
No handshake lines used

1 start bit
1 stop bit
8 data bits
no parity

In this application, all RS-232 serial data communication uses the ASCII character set.

Four standard baud rates are available. The maximum baud rate supported in this application is 38.4K baud. The baud rate is determined by the setting of a DIP switch on the Z-Card circuit board (see "DIP SWITCH SETTINGS" drawing). The reset switch must be pressed once after the switch setting is changed to activate the new baud rate or change the switch reporting method.



INDIVIDUAL LED CONTROL COMMANDS:

The three basic commands control the status of individual LEDs. A LED can be turned **ON**, turned **OFF**, or **FLASH**ed at a steady rate. After a LED is commanded to flash, it will continue to flash until commanded otherwise.

The command format is *(abcy)*, where abc represents the LED address in ASCII numerals. LED addresses are numbered consecutively starting with LED #0. Leading zeroes are optional, so addresses can be preceded by leading zeroes, or the leading zeroes can be omitted.

The character *y* signifies the status character. There are three ASCII characters which may be used: ***X, Z*** and ***F***. An ***X*** will turn the addressed point **ON**, and a ***Z*** will turn it **OFF**. An ***F*** will cause the addressed point to **FLASH**. It will continue to **FLASH** until it is commanded **ON** or **OFF**.

Both parenthesis must be inserted as shown. Messages which differ from this format will be ignored. Carriage return, line feed, and space characters are automatically filtered out of the input data stream without affecting the surrounding data.

LED COMMAND EXAMPLES:

(003F) FLASHES LED 3, or
(03F) FLASHES LED 3, or
(3F) FLASHES LED 3
(02Z) turns LED 2 OFF
(00X) turns LED 0 ON

BLOCK FORMATTED LED COMMANDS:

The best interfaces for display panels refresh the display data frequently. This way, if a message is lost or garbled due to transient noise, the display will quickly be restored through repetition of the data. It also makes installation and maintenance much more efficient. If the panel is powered down, it will be rapidly refreshed as soon as power is restored. Likewise, if the communication link is interrupted, the display will be updated when communication is restored.

The block format is designed to accomplish this while minimizing the host computer's overhead. A typical block transmission begins with the left parenthesis, followed immediately by up to 240 status characters. The status characters are the same ones used in the individual LED commands. The first status character controls LED #0, the second controls LED # 1, etc.

As with the individual LED commands, carriage return, line feed, and space characters are automatically filtered out of the input data stream without affecting the surrounding data.

UTILITY LED COMMANDS:

Commands are provided to turn all the LEDs OFF, or to turn all the LEDs ON. The ***(CLR)*** command clears all the LEDs in memory and on the display. This command is useful when the display is being initialized.

The ***(TST)*** command turns all the LEDs on for about 3 seconds, then restores the previous display status. It allows the programmer to provide the same function as a front panel LED TEST switch. In both commands, the parenthesis must be inserted as shown. Messages which differ from this format will be ignored.

Any type of data (garbage or non-valid commands) may be received after the closing parenthesis of a command without affecting the operation of the preceding command.

ACTIVE SENSING MESSAGE:

The Z-Card will send out a **!** character every time it receives a <cr> character input. This allows the host to supervise the data connection and verify that commands are being received by the control graphic.

LED STATUS REQUEST COMMANDS:

The host computer can request the status of any LED by sending a request message containing the LED address.

For example:

($024) Request status of LED #24

The response from the graphic panel would be one of the following three messages:

(F024) LED #24 is flashing

or:

(Z024) LED #24 is off

or:

(X024) LED #24 is on

The response will include the same number of address characters as the request contained, up to four characters. For example, a **($0024)** request message will elicit a response such as **(F0024)**.

SWITCH ACTUATION MESSAGES:

When a switch is pressed, and switch activation messages are enabled by the Z-Card's DIP switch setting, the Z-Card transmits out a message in the format **(Annn)** where nnn = three digit ASCII numeric switch address. When a switch is released, the Z-Card transmits out a message in the format **(Rnnn)**.

SWITCH MESSAGE EXAMPLES:

(A007) Switch #7 activated
(R007) Switch #7 released
(A123) Switch #123 activated
(R123) Switch #123 released

Switches are debounced by the firmware, and the messages are transmitted only when the switch changes state. They always contain three address characters.

SUPPRESSING SWITCH ACTUATION MESSAGES:

The Z-Card DIP switch can be set to suppress all Z-Card initiated switch messages. The host can still use switch request messages to obtain the state of any switch.

If switch actuation messages are disabled, then the switches are latched into memory so that momentary switch actuations will be reported even if the switch request command occurs at a later time, after the switch has been released. Once interrogated, a switch is "unlatched" and cleared.

If switch actuation messages are enabled, then pressing a switch causes immediate transmission of its message. In this mode, the switches are not latched, and when a switch is interrogated, its instantaneous state will be reported.

SWITCH STATUS REQUEST COMMANDS:

The host computer can request the status of any switch by sending a request message containing the switch address.

For example:

(?030) Request status of switch #30

The response from the graphic panel would be one of the following two messages:

(A030) switch #30 is activated

or:

(R030) switch #30 is released

The response will include the same number of address characters as the request contained, up to four characters. For example, a **($0024)** request message will elicit a response such as **(F0024)**. For systems with a large number of switches, and with actuation messages suppressed, it may be useful to use the block transfer message to speed up switch polling activity.

SWITCH STATUS BLOCK TRANSFER:

The host computer can request the status of 80 switches at a time by sending a block request message as follows:

(?SBKx)

for "SWITCH BLOCK x", where x is the number of the switch block status requested. Numbers used for the switch blocks can be 1 -8.

The response from the graphic panel would be the following format:

(yyyAAAAARRRRR...)

Where the "yyy" represents the 3-digit address of the first switch status character. This starting address is determined by the switch block requested. If switch block #1 is requested, the address in the return message will be **"000"**, if switch block 2 is requested the address will be **"080"** etc, in increments of 80.
The message will contain the status of 80 switches with the first status character representing the status of switch #**"yyy"** and the next being switch #**"yyy" + 1**, etc. An "A" means the switch has been activated and an "R" means the switch is released.