

GLO-416Y User's Guide

GLO-416s take serial data at 9600bps and display it on a bright yellow OLED screen. This document briefly describes hookup and configuration, and outlines instructions for formatting text on the screen. For indepth coverage, see the online docs at seetron.com/glo416.

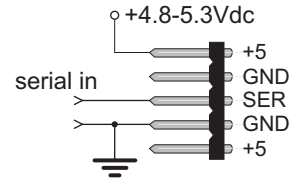
Hookup and Initial Checkout

Connect regulated 5.0Vdc power to the header pins marked +5 and GND. (Either +5 or GND pin may be used, they are duplicated to allow use of a reversible connector.) A factory-configured GLO will display

GLO-416Y
v #.# seetron

If the configuration has been changed by the user, the startup screen may be different (or blank).

To send data to the display, set the computer or microcontroller serial port for 9600bps, N81 (no parity, 8 data bits, 1 or more stop bits), handshaking (flow-control) off. With the correct settings, when you send text like "Hello GLO!" to the display, it will appear on the screen.



Reconfiguring Serial Polarity (SPol Jumper)

GLO defaults to inverted serial input. Inverted is normal: PC comm ports, RS-232 ports and many microcontrollers work this way. Microcontrollers that send serial via a general I/O pin (Basic Stamps, etc.) usually *can* output inverted based on a parameter in the serial-out instruction.

But if a microcontroller outputs serial directly through a "hardware UART" or "USART" pin, the GLO may have to be set for non-inverted ("true") operation. To do so, cut the wire jumper marked **SPol** (Serial Polarity). Check your documentation before doing so; this is a one-way option. If you are unsure, see the online docs at seetron.com/glo416 for additional advice.

Display Setup (DSU) Jumper

The DSU jumper on the serial interface permits startup screen and custom-character data to be written to the GLO's EEPROM. Cutting the DSU jumper write-protects the EEPROM to prevent further changes. Leave this jumper intact until no further changes are required.

Formatting Instructions

GLO accepts instructions that control the appearance and location of displayed text. See the table on page 2. Some instructions, like Clear Screen, are obvious; others are more subtle. See the online docs at seetron.com/glo416 for more.

Instruction	ASCII	Hex	Notes
Home cursor (ctrl-A)	1	0x01	
Select font size (ctrl-B)	2	0x02	
Set default font size (ctrl-C)	3	0x03	
Backspace (ctrl-H)	8	0x08	
Tab: jump to next multiple-of-4 column (ctrl-I)	9	0x09	
Smart linefeed: move down one row (ctrl-J)	10	0x0A	
Vertical tab: move up one row (ctrl-K)	11	0x0B	
Formfeed: clear screen (ctrl-L)	12	0x0C	
Carriage return: move to start of next line (ctrl-M)	13	0x0D	
Turn OLED driver on (ctrl-N)	14	0x0E	
Turn OLED driver off (ctrl-O)	15	0x0F	
Accept cursor-position data (next byte) (ctrl-P)	16	0x10	1
Accept right-alignment data (ctrl-R)	18	0x12	2
Set 7-segment big numbers (ctrl-S)	19	0x13	
Set text-style big numbers (ctrl-T)	20	0x14	
Escape character: begin multipart instruction	27	0x1B	3
NOTES:			
1. Byte after ctrl-P is a byte equal to position + 64 (0x40 hex), or text # followed by space (discarded).			
2. Byte after ctrl-R is a text # '0'-'7' (48-55 dec, 0x30-0x37 hex) indicating max width of right-aligned text			
3. See below.			
4. See online docs at seetron.com/glo416			

Escape Instructions

The Escape character (27 dec, 0x1B hex) starts the following multipart instructions:

Define Custom Character: ESC D and ESC d

GLO-216 has 16 custom characters located above the standard ASCII character set at 128-143 dec (0x80-0x8F hex). The bit pattern of these characters may be changed using **ESC D n [B0-B7]**. That's 11 bytes: Escape, uppercase 'D', a text number '0'-'7' (48-55 dec, 0x30-0x37 hex), then 8 bytes defining the character. Since **n** only covers the first 8 custom chars, substituting lowercase 'd' is used for the upper 8.

Recall Saved Text or Custom Characters: ESC E0 and ESC e0

GLO-216 stores startup text and custom chars in EEPROM. The text is recalled by **ESC E 0**. That's 3 bytes: Escape, uppercase 'E' and the text '0' (zero). The saved text does not clear the display and can contain instructions, a powerful macro capability that may produce unexpected results. See online docs. **ESC e 0** (lowercase 'e') restores the custom characters to their EEPROM-stored patterns.

Save Text or Custom Characters: ESC X0 and ESC x0

To store text sent since the last clear-screen instruction use **ESC X 0**. That's 3 bytes: Escape, uppercase 'X' and the text '0' (zero). **ESC x 0** (lowercase 'x') stores the current set of custom-char patterns.