DSP004 Intelligent Character Display
Technical Manual

Pronto Electronic Displays LLC
www.prontodisplays.com

Doc.# 1031-DSP004-TM REV.03

Table of Contents

1.0 Quick Start Guide

1.1 Connect the Hardware

⚠ **Caution — Hardware Connection Restrictions**

- **DO NOT** power this display from a USB port. The onboard bulk capacitance produces an inrush current spike that can trigger host-port protection or cause the port to lock out. Use a dedicated external 5 V supply only.
- **DO NOT** hot-plug or hot-unplug the 10-pin SIP connector.
- **ALWAYS** ensure all wiring and connections are fully secure before applying primary power to the system.

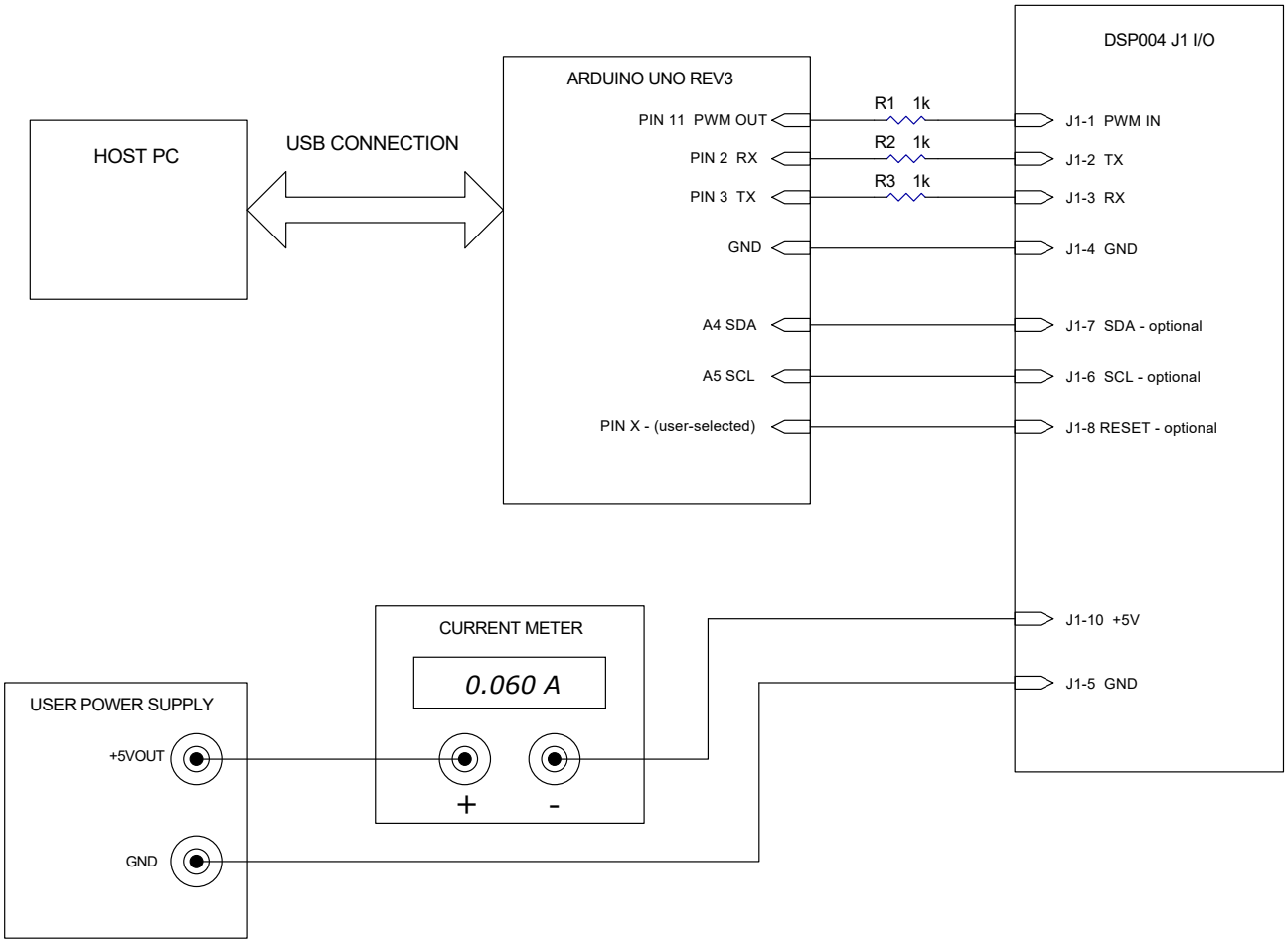


FIGURE 1.1 – DEVELOPMENT SETUP SHOWING POWER, SIGNAL AND COMPUTER CONNECTIONS

⚠ **Caution — Back-Powering Hazard**

Back-powering can occur if the display module remains powered while the host microcontroller (Arduino) is turned off. To ensure safe operation during evaluation, the following hardware protections must be employed.

Hardware Mitigation (UART & PWM Lines)

- Install a **1 kΩ series resistor** on the RX, TX, and PWM lines to limit leakage current, as depicted in Fig. 1.1.
- The display's RESET line already includes an on-board 1 kΩ series resistor; **no external resistor is required**.
- **Do not** add 1 kΩ resistors to the I²C lines (SDA/SCL), as this will corrupt I²C communication. The SDA and SCL pins are pulled up to display board +5V with 4.7k pullup resistors.

Notice — Electrical Damage Risk These evaluation instructions are provided solely as a reference guide. The user is responsible for correct electrical interfacing and wiring. Incorrect wiring, or failure to implement the recommended hardware protections may result in damage to host microcontrollers, peripheral hardware, or associated systems. To the maximum extent permitted by applicable law, Pronto Electronic Displays LLC disclaims all liability for any such damage.

The Arduino Uno Rev3 is used as a reference platform for the following examples.

The first step is to access the on-board integrated menu. This can be done in two ways:

a) **Using a USB Virtual COM Port (VCP).** Connect a USB-VCP to +5V TTL/UART interface to the host computer (not illustrated here). Wire the VCP TX → display RX, VCP RX → display TX, and connect GND → GND, along with the external power supply.

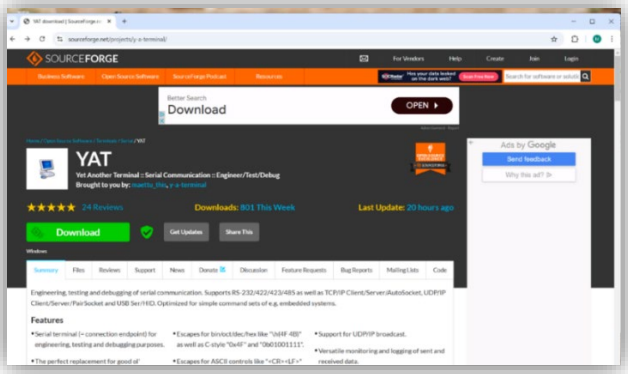
b) **Using an Arduino running the Terminal Bridge sketch.** Upload the Terminal Bridge sketch to the Arduino and connect it—along with the power supply—to the display as shown in the diagram. The sketch may be downloaded from the Pronto Electronic Displays website (www.prontodisplays.com) or entered manually using the copy provided in Appendix A.5.
This method is preferred because it eliminates the need for a separate USB-VCP adapter and avoids the need to disconnect and re-wire when switching to a demonstration sketch. For example, after completing configuration with the Terminal Bridge sketch, simply close the terminal emulator port, then upload the Demonstration sketch to the Arduino.

After uploading the **Terminal Bridge sketch** into the Arduino, connect the Arduino **RX**, **TX**, **PWM** and **GND** lines exactly as shown in the connection diagram above. Verify that the external power supply is set to **5 V output** with a current limit of **1 A** (if available). Connect the supply to the display module as indicated. Note that connecting the PWM line will permit the Demonstration sketch to properly modulate the display brightness when the demo sketch is run.

If the PC operates using Windows, proceed to Section 1.2; for macOS, proceed to Section 1.3.

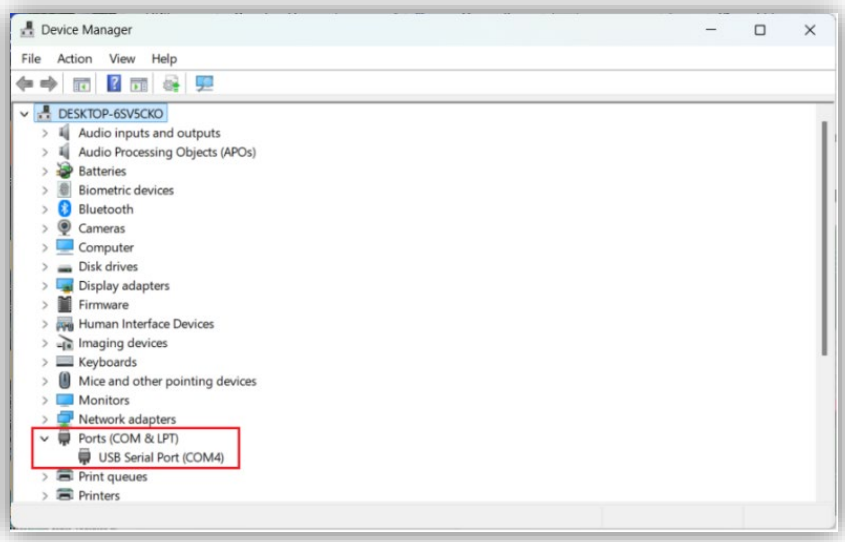
1.2 Configuring a Serial Emulator for Windows Operating System

For Microsoft Windows, YAT may be used. It is available free of charge at sourceforge.net, and is used for these Windows operating system examples.

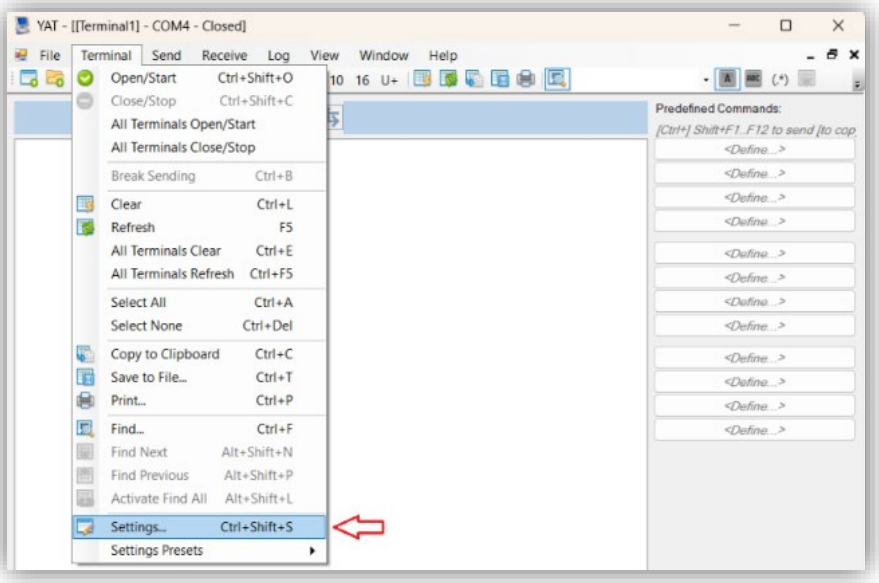


1. The display Rx, Tx and GND should be connected as outlined in Fig.1.1, the USB cable should still be connected to the Arduino, the Arduino should still be running the “Terminal Bridge” sketch, and the display should be powered on. Ensure the “CONFIG” switch on the back of the display is in the “NORM” position.

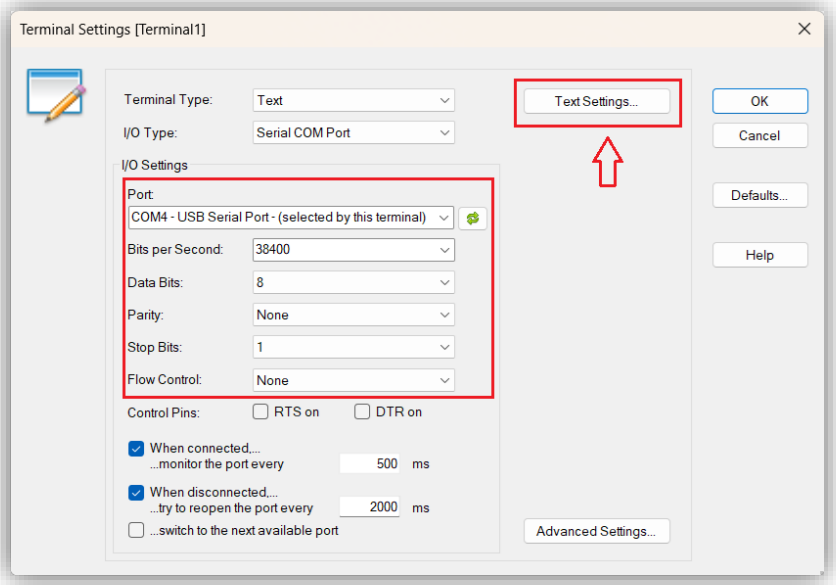
2. Determine the COM port number being used by the display. Do this by opening the Windows Device Manager, and click on “Ports (COM & LPT)”. The appearance of a “USB Serial Port” or some other port name referencing “USB” and “Serial” verifies that the drivers have already installed correctly. Take a note of the COM Port number- in the case of the example below, it is COM4. The window may then be closed.



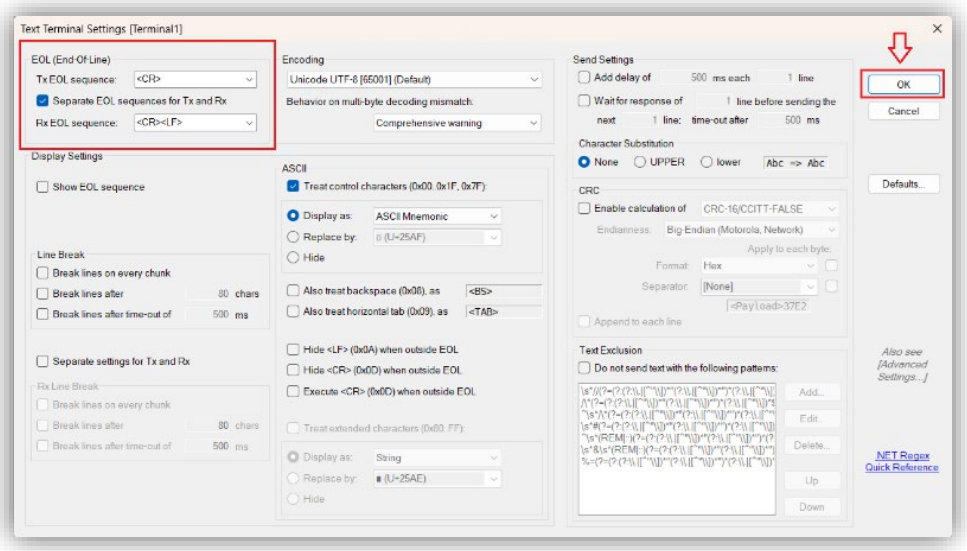
3. Open YAT. On the top menu, go to “Terminal” and click on “Settings”:



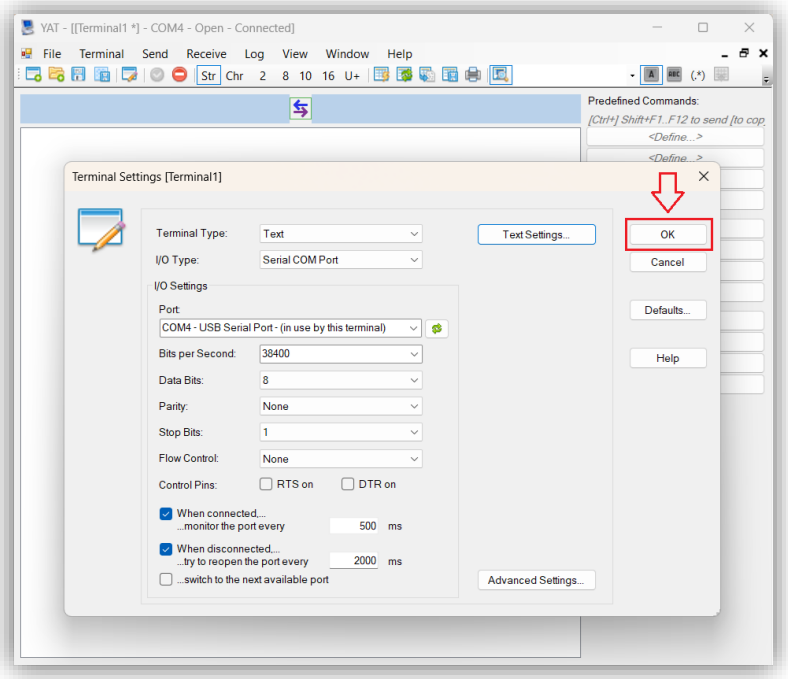
4. In the “I/O Settings” section of the Terminal Settings window (see below):
   a) Under “Port”, select the correct COM port you found in Step 2- in this example case, COM4.
   b) In “Bits per Second”, select 38400.
   c) In “Data Bits”, ensure “8” is selected.
   d) In “Parity”, ensure “None” is selected.
   e) In “Stop Bits” ensure 1 is selected.
   f) In “Flow Control” select None.
   g) Leave all other entries as-is in their default configuration.
   Next, click the “Text Settings” button (see below):



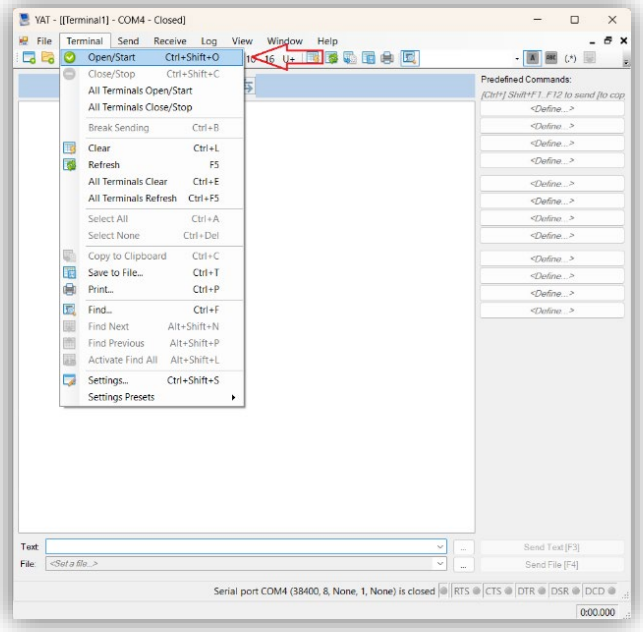
5. In the “Text Terminal Settings” window below, in the “EOL (End of Line)” section:
   a) Click box to enable ‘Separate EOL sequences for Tx and Rx”.
   b) Select <CR> for “Tx EOL sequence”.
   c) Select <CR><LF> for “Rx EOL sequence”.
   d) Leave all other settings as-is in default configuration.
   e) Click “OK” to save settings and close this window:



6. In the “Terminal Settings” window click “OK” to save the settings and close the window:



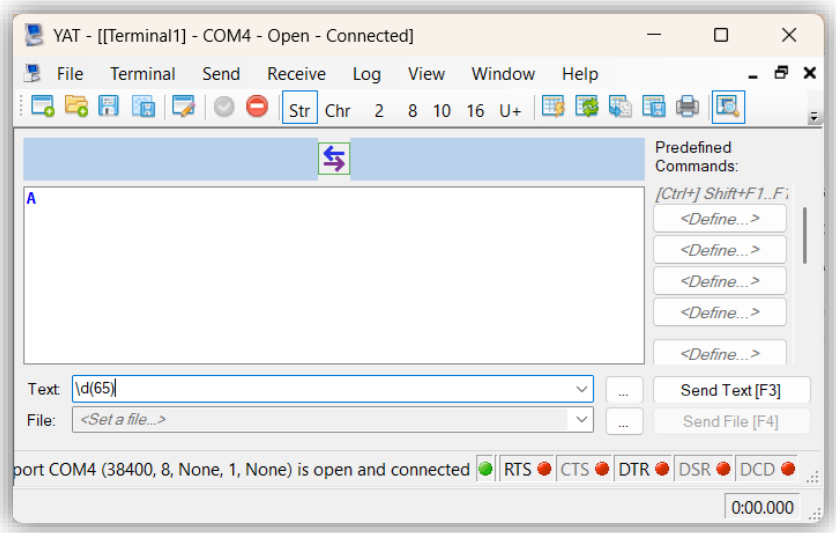
7. Go to “Terminal” and click “Open/Start”:



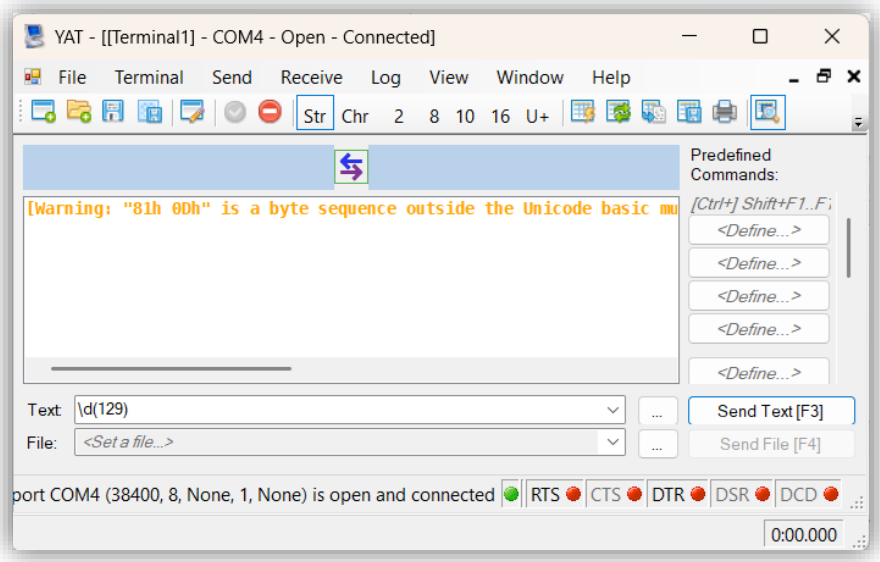
8. Again, ensuring the “CONFIG” switch is set to “NORM” position, push the “MENU” button on the back of the unit. Connection is successful if the user interface menu appears as below. One may have to widen the emulator window to see the entire menu. Verify communications by entering characters and interacting with the user interface. This is done by typing entries into the “Text” window and then clicking the “Send Text” button.

```
*********TERMINAL CONNECTION SUCCESSFUL*********
        Firmware Rev.5.7.7

********OPERATING IN MENU CONFIGURATION********
1. BAUD RATE FIXED TO 38400
2. RxIn  EOL TERMINATOR FIXED TO <CR>
  (TxOut EOL Terminator is fixed at <CR><LF>)
3. BRIGHTNESS MODE FIXED TO AUTOMATIC LIGHT SENSOR:
    Max Display Brightness set to 100%
    Min Display Brightness set to 10%
4. DISPLAY FIXED TO NON-SCROLLING MODE.

******** USER PROGRAMMED CONFIGURATION SETTINGS *******
(Settings applied when Config Switch set to PRGM position)

1. USER PROGRAMMED BAUD RATE: 38400
2. USER PROGRAMMED RxIn EOL TERMINATOR: <CR>
3. USER PROGRAMMED BRIGHTNESS MODE: AUTOMATIC LIGHT SENSOR
    User Programmed Max Brightness Setting: 100%
    User Programmed Min Brightness Setting: 10%
4. USER PROGRAMMED SCROLL MODE SET TO SCROLLING DISPLAY.
5. USER PROGRAMMED SCROLL SPEED SET TO 0.182 SECONDS. NUMERIC ENTRY: 20
6. USER PROGRAMMED DATA GO/STOP MODE ENABLED.
7. USER PROGRAMMED SOFTWARE RESET ENABLED.
8. USER PROGRAMMED SCROLL DONE FLAG ENABLED.


***************MAIN MENU***************

Enter 'A' to set User Programmed Baud Rate
Enter 'B' to set User Programmed RxIN EOL Terminator
Enter 'C' to change User Programmed Brightness Mode Settings
Enter 'D' to Enable/Disable User Programmed Display Scroll Mode
Enter 'E' to change User Programmed Scroll Speed
Enter 'F' to Enable/Disable Data Go/Stop Mode
Enter 'G' to Enable/Disable Software Reset
Enter 'H' to Enable/Disable Scroll Done Flag
Enter 'I' to create a new Custom Character
Enter 'J' to Escape


Awaiting Entry:
```

9. To test a custom character, it is necessary to have the terminal emulator send a number, not a string of characters, over the serial line. For example, to test the symbol for the capital letter ‘A’, one needs to send the number 65 over the serial line. It is quite simple to do with YAT. Sending \d(65) in the text entry window will cause the decimal number 65 to be sent, resulting in a capital A being seen on the display unit.



(Note: to send an actual backslash character, send \\ ). Consult the YAT quick start guide for more information).
To test a custom character with an ASCII number of 129 assigned to it, send \d(129) in the text entry window.



Note that a warning message appears in the send/receive window; this is normal and is simply informing the user that 129 is outside the Unicode symbol system.
All custom characters will cause this warning to be displayed.

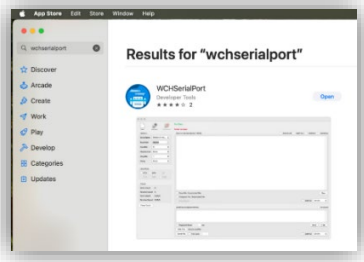
At the factory, ASCII 129 has been assigned as an “UP” arrow for testing purposes. Provided it has not been overwritten by the user, it should appear as below:

Note that YAT configuration settings will be retained: After closing and re-opening YAT, it is not necessary to re-enter the settings.

When a serial terminal program (such as YAT) closes and re-opens a COM port, the serial emulator software may emit a small number of corrupted or invalid bytes during reinitialization. These bytes are not part of any valid message and may briefly appear on the display if a session is restarted while the device is powered and listening. This may cause incorrect characters to appear or to be included in the first message sent after re-opening the session. This behavior is normal for many serial emulator applications and does not indicate a fault in the display. If unexpected characters appear after reopening a terminal session, simply clear the display or resend the intended message.

1.3 Configuring a Serial Emulator for macOS Operating System

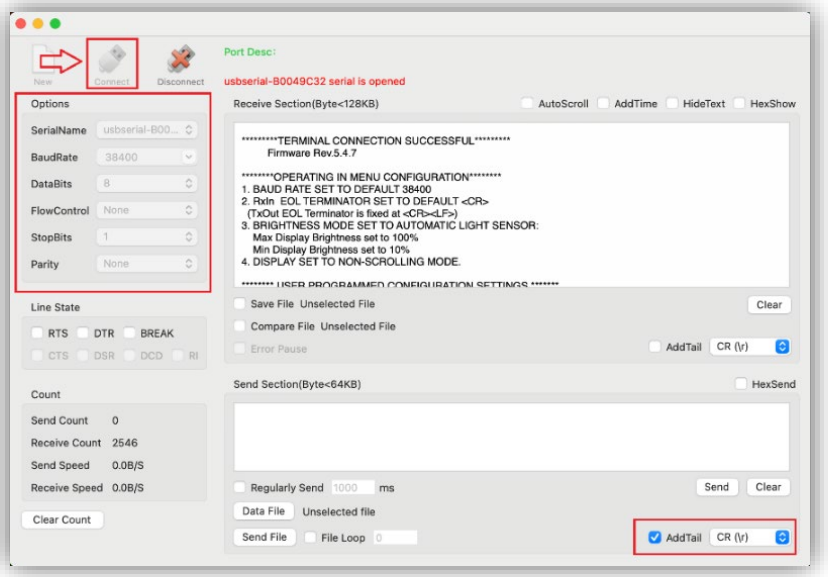
1. For Apple macOS, WCHSerialPort may be used. It is available free of charge in the Apple App Store. Download and install it according to the instructions provided.



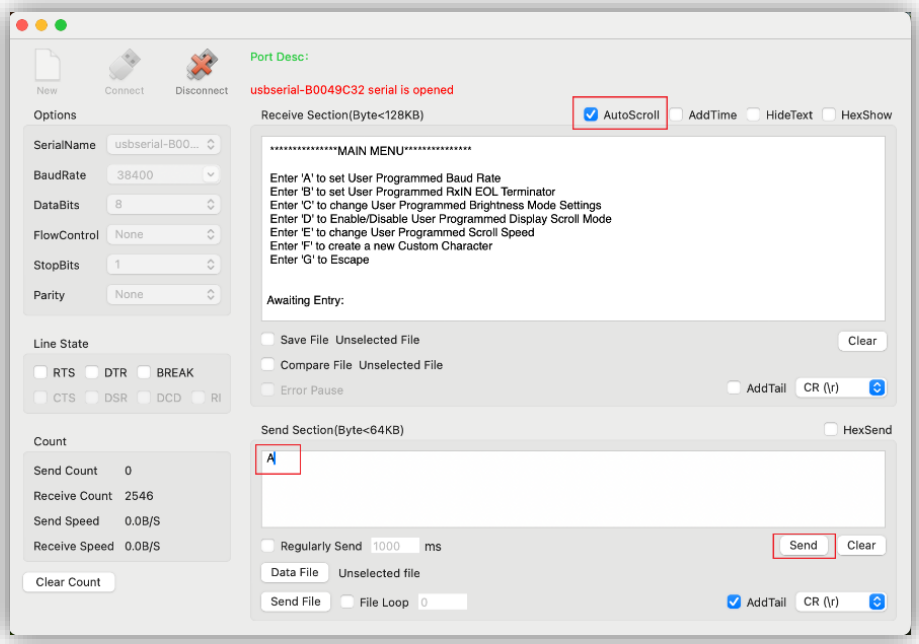
2. The display Rx, Tx and GND should be connected as outlined in Fig. 1.1, the USB cable should still be connected to the Arduino, the Arduino should still be running the “Terminal Bridge” sketch, and the display should be powered on. Ensure the “CONFIG” switch on the back of the display is in the “NORM” position.
3. Open the wshserialport application. In the “Options” section (see screenshot, below):

a) In the “Serial Name” list, select a name referencing “usb” and/or “serial”. This is also an indication that the VCP drivers have been installed.
b) In the “Baud Rate” section, select 38400 from the list.
c) DataBits should be left at the default value of 8.
d) FlowControl should be left at the default value of None.

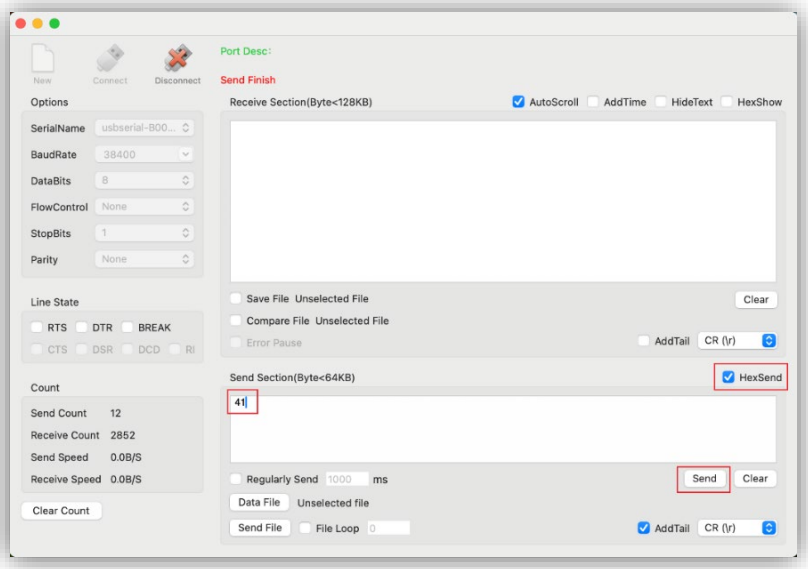
e) StopBits should be left at the default value of 1.
f) Parity should be left at the default value of None.
4. At the lower right in the “Send” section, check “AddTail”. It should be set at default value of CR.
5. At the top left of the window, click “Connect”.
6. If connection is successful, the Menu should appear similar to the screenshot in the Receive Section, as in the figure below.



Scroll down to the bottom of the menu in the Receive Section. Click “AutoScroll” as shown below.



7. Verify communications and functionality by interacting with the menu. Type entries in the “Send Section” and then click the “Send” button, as depicted above.
8. To test a custom character, it is necessary to have the terminal emulator send a number, not a string of characters, over the serial line. For example, to test the symbol for the capital letter ‘A’, one needs to send the number 65 over the serial line. It is quite simple to do with WCHSerialPort. In the “Send” section, check “HexSend” (see below). Next enter the hexadecimal equivalent of Decimal 65, which is 41, end click “Send”. The letter capital “A” should appear on the display unit.



To test a custom character with an ASCII number of 129 assigned to it, send the hexadecimal equivalent for decimal 129, which is 81. At the factory, ASCII 129 has been assigned as an “UP” arrow for testing purposes. Provided it has not been overwritten by the user, it should appear as below:

When the application is closed, upon re-opening you will need to re-select the Serial Name, re-select Baud rate to 38400, and re-check the “Add Tail” box.

When a serial terminal program (such as WCHSerialPort) closes and re-opens a COM port, the serial emulator software may emit a small number of corrupted or invalid bytes during reinitialization. These bytes are not part of any valid message and may briefly appear on the display if a session is restarted while the device is powered and listening. This may cause incorrect characters to appear or to be included in the first message sent after re-opening the session. This behavior is normal for many serial emulator applications and does not indicate a fault in the display. If unexpected characters appear after reopening a terminal session, simply clear the display or resend the intended message.

2.0 Using the Menu

2.1 The Main Menu

Upon successful connection, when the MENU button is pressed, the Main Menu will be displayed:

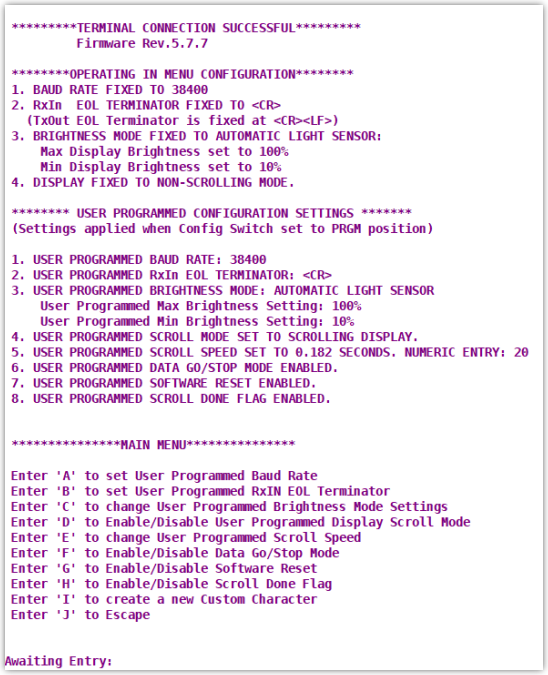


FIG.2, Headers and Main Menu

The “OPERATING IN MENU CONFIGURATION” header serves as a message to the user that the unit is currently operating in the fixed factory configuration. The settings for this configuration are listed in this header to serve as a reference for the user. Note that the EOL terminator for the Tx OUT line is not user-adjustable. It is fixed at <CR><LF>, and this terminator is used when the menu is being printed to the Host computer. Zx DATA_GO, DATA_STOP and SCROLL_DONE flags are sent over the TxOUT line without any EOL terminators.

The “USER PROGRAMMED CONFIGURATION SETTINGS” header displays the device configuration settings that the unit will operate under whenever the “CONFIG” switch is set to the “PRGM” position. The settings for this configuration are listed in this header to serve as a reference for the user. Note the unit comes with settings pre-programmed in the factory, which of course may be overwritten by the user.

The “MAIN MENU” header shows all the configuration settings that can be changed, as well as providing an entry allowing the creation of a custom character. The remainder of this section provides details for each entry made.

Note in all cases upon placing an entry when the unit confirms the choice, it will then perform a reset, and will then display the statement “RESET AND APPLIED CHANGES. EXITING MENU”, at which time the unit then exits out of the Main Menu, and resumes normal operation. Push the MENU button again if it is desired to make more entries from the Main Menu. The newly applied settings will be shown in the “USER PROGRAMMED CONFIGURATION SETTINGS” header, which will appear the next time the MENU button is pushed again.

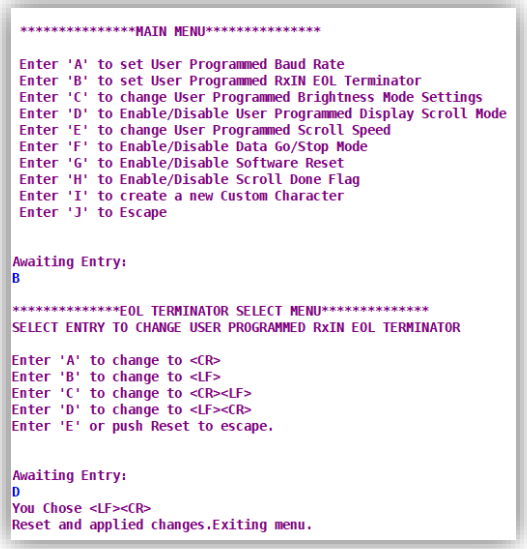
2.2 Setting User Programmed Baud Rate

An “A” entry in the Main Menu results in a Baud rate selection menu being displayed:

A dialog will commence asking the user to place A,B or C.The unit will operate using the selected BAUD rate when the CONFIG switch is set to the “PRGM” position. Push the MENU button again to make more selections.

2.3 Setting the User Programmed RxIN EOL Terminator

A “B” entry in the Main Menu results in a Baud rate selection menu being displayed:



The EOL Terminator that is being selected is for the display RxIN line (Pin 3). (Note the TxOUT EOL terminator for the unit is always fixed at <CR><LF>). The unit will operate using the selected RxIN EOL terminator when the CONFIG switch is set to the “PRGM” position. Push the MENU button again to make more selections.

2.4 The Brightness Mode Configuration Menu

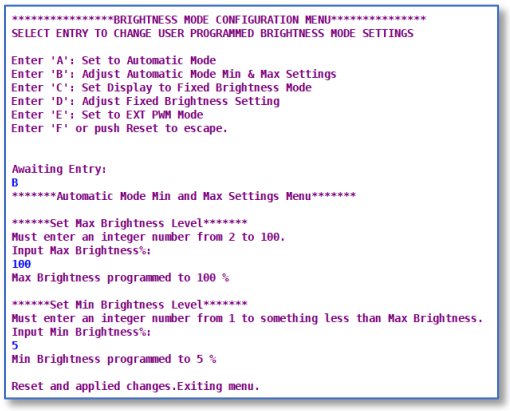
A “C” entry in the Main Menu results with the Brightness Mode Configuration Menu being displayed:

2.4.1 Set to Automatic Mode

An “A” entry in the “Brightness Mode Configuration Menu” will set the unit operating in the Automatic mode when the CONFIG switch is set to the “PRGM” position. In this mode, the device automatically modulates the display brightness depending upon ambient light conditions, utilizing the ambient light sensor which is located on the display side of the device directly alongside the leftmost character. The “min and max” settings currently programmed will be displayed in the “User Programmed Configuration Settings” header when Automatic Mode of operation is selected. Push the MENU button again to make more selections.

2.4.2 Adjust Automatic Mode Min & Max Settings

An “B” entry in the “Brightness Mode Configuration Menu” will bring up a dialog in which one may enter min and max brightness levels:



These levels represent a linear percentage of min and max internally-generated PWM duty cycle. The brightness minimum value determines how dim the display may appear under the dimmest lighting conditions, and the brightness maximum value determines how bright the display may appear under the brightest ambient light conditions. In Automatic mode, the display’s brightness is limited by the ambient-light sensing behavior of the system.

As a result, Automatic mode does not reach the designed 100% brightness level; its maximum output is approximately 95%, even when configured for 100%. To obtain true full-brightness operation, use Fixed Brightness mode set to 100% or External PWM mode with a 100% duty cycle.

After completing entry, push the MENU button again to make more selections.

2.4.3 Set Display to Fixed Brightness Mode

A “C” entry in the “Brightness Mode Configuration Menu” will cause the unit to operate at a display fixed brightness level when the CONFIG switch is in the PRGM position.

```
***************BRIGHTNESS MODE CONFIGURATION MENU**************
SELECT ENTRY TO CHANGE USER PROGRAMMED BRIGHTNESS MODE SETTINGS

Enter 'A': Set to Automatic Mode
Enter 'B': Adjust Automatic Mode Min & Max Settings
Enter 'C': Set Display to Fixed Brightness Mode
Enter 'D': Adjust Fixed Brightness Setting
Enter 'E': Set to EXT PWM Mode
Enter 'F' or push Reset to escape.


Awaiting Entry:
C
Display Set to Fixed Brightness Level.
Reset and applied changes.Exiting menu.
```

The fixed brightness mode setting currently programmed will be displayed in the “User Programmed Configuration Settings” header when the Fixed Brightness Mode of operation is selected.

After finishing the entry, push the MENU button again to make more selections.

2.4.4 Adjust Fixed Brightness Setting

A “D” entry in the “Brightness Mode Configuration Menu” will bring up the “Display Fixed Brightness Settings Menu”:

```
***************BRIGHTNESS MODE CONFIGURATION MENU**************
SELECT ENTRY TO CHANGE USER PROGRAMMED BRIGHTNESS MODE SETTINGS

Enter 'A': Set to Automatic Mode
Enter 'B': Adjust Automatic Mode Min & Max Settings
Enter 'C': Set Display to Fixed Brightness Mode
Enter 'D': Adjust Fixed Brightness Setting
Enter 'E': Set to EXT PWM Mode
Enter 'F' or push Reset to escape.


Awaiting Entry:
D
*******Display Fixed Brightness Settings Menu*******

Must enter an integer number from 1 to 100.
Input Fixed Brightness%:
70
Fixed Brightness programmed to 70 %

Reset and applied changes.Exiting menu.
```

In this menu the user is prompted for entering a percentage number from 1 to100. This number represents a linear percentage of the device’s internal PWM duty cycle. The display illumination will be at a level corresponding to this entry, if the unit is set to operate in the Fixed Brightness Mode. After completing entry, push the MENU button again to make more selections.

2.4.5 Set to EXT PWM Mode

An “E” entry in the “Brightness Mode Configuration Menu” will cause the unit to operate in EXT PWM mode as a default, when the CONFIG switch is set to the PRGM position:

```
***************BRIGHTNESS MODE CONFIGURATION MENU***************
SELECT ENTRY TO CHANGE USER PROGRAMMED BRIGHTNESS MODE SETTINGS

Enter 'A': Set to Automatic Mode
Enter 'B': Adjust Automatic Mode Min & Max Settings
Enter 'C': Set Display to Fixed Brightness Mode
Enter 'D': Adjust Fixed Brightness Setting
Enter 'E': Set to EXT PWM Mode
Enter 'F' or push Reset to escape.


Awaiting Entry:
E
Display Brightness selected to EXT PWM Mode.
Reset and applied changes.Exiting menu.
```

In this mode, the internal ambient light sensing is bypassed, and the display brightness can then be modulated by an external PWM signal supplied by the user on Pin 1. With the unit set to operate in this mode, the unit display brightness will be at maximum level, if no PWM signal is supplied to Pin 1. External PWM Mode can also be enabled/disabled with individual message granularity using Zx control codes. In the “User Programmed Configuration Settings” header, operation in EXT PWM mode will be indicated if this mode of operation is selected. After completing entry, push the MENU button again to make more selections.

This concludes the section on the Brightness Mode Configuration menu. The next section begins again with Main Menu entries.

2.5 Enable/Disable User Programmed Display Scroll Mode

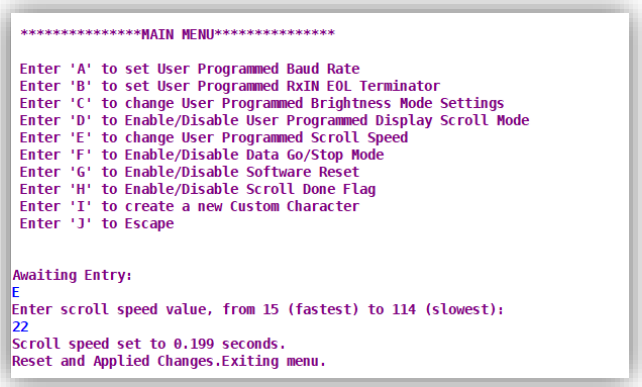
A “D” entry from the Main Menu will present a dialog to the user for enable/disable of the display scrolling feature:

```
***************MAIN MENU***************

Enter 'A' to set User Programmed Baud Rate
Enter 'B' to set User Programmed RxIN EOL Terminator
Enter 'C' to change User Programmed Brightness Mode Settings
Enter 'D' to Enable/Disable User Programmed Display Scroll Mode
Enter 'E' to change User Programmed Scroll Speed
Enter 'F' to Enable/Disable Data Go/Stop Mode
Enter 'G' to Enable/Disable Software Reset
Enter 'H' to Enable/Disable Scroll Done Flag
Enter 'I' to create a new Custom Character
Enter 'J' to Escape


Awaiting Entry:
D
Enter A to enable Scrolling, or B for Non-scrolling mode:
A
User Programmed Display set to Scrolling mode.
Reset and applied changes.Exiting menu.
```

The user entry made determines the display scroll mode the unit will operate in as the default mode, when the CONFIG switch is in the PRGM position. The scroll mode setting currently programmed will be displayed in the “User Programmed Configuration Settings” header.

2.6 Change the User Programmed Scroll Speed

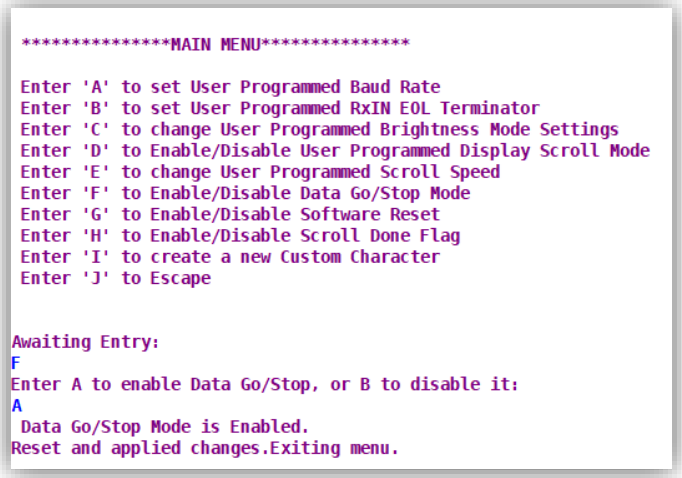
An “E” entry from the Main Menu will present a menu for setting up the Scroll Speed the unit will operate with when it is running in Scrolling mode:



To change the scroll speed, the user enters a positive integer from the range of 15 (fastest) to 114 (slowest). “Scroll Speed” is defined as the time each character remains illuminated; hence, a smaller number results in a faster-scrolling display. When entered, the display confirms the settings and displays the approximate scroll speed in seconds. After completing entry, push the MENU button again to make more selections. When the UI menu is recalled, *and if the unit is programmed to operate in Scrolling Mode*, then the scroll speed setting will be displayed in seconds, along with the corresponding numerical entry that was made, in the “User Programmed Configuration Settings” header. Scroll speed may also be changed using the Zx SCROLL_SPD “scroll speed prefix” control code. Refer to the section of this document “Using Zx Codes” for more details.

2.7 Enable/Disable Data Go/Stop Control Characters

An “F” entry from the Main Menu will present a menu for enabling or disabling the Z code “DATA_GO” and “DATA_STOP” control characters. When disabled, the unit will no longer transmit these control character flags when operating in the Scroll mode.



2.8 Enable/Disable Software Reset Command

A “G” entry from the Main Menu will present a menu for enabling or disabling the Zx™ RESET control character. When disabled, the unit will no longer respond to a software reset command when the unit is operating with the CONFIG switch in the PRGM position.

```
***************MAIN MENU***************

Enter 'A' to set User Programmed Baud Rate
Enter 'B' to set User Programmed RxIN EOL Terminator
Enter 'C' to change User Programmed Brightness Mode Settings
Enter 'D' to Enable/Disable User Programmed Display Scroll Mode
Enter 'E' to change User Programmed Scroll Speed
Enter 'F' to Enable/Disable Data Go/Stop Mode
Enter 'G' to Enable/Disable Software Reset
Enter 'H' to Enable/Disable Scroll Done Flag
Enter 'I' to create a new Custom Character
Enter 'J' to Escape


Awaiting Entry:
G
Enter A to enable Software Reset, or B to disable it:
A
 Software reset is Enabled.
Reset and applied changes.Exiting menu.
```

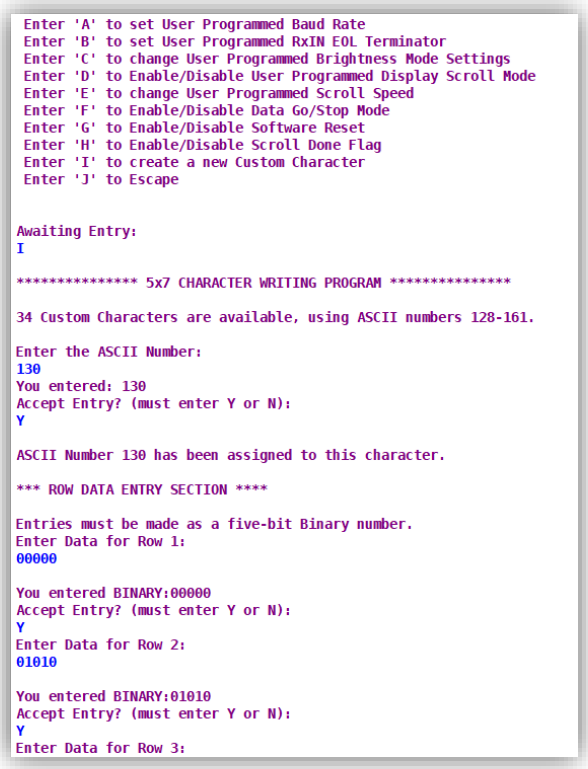
2.9 Enable/Disable Scroll Done Flag

An “H” entry from the Main Menu will present a menu for enabling or disabling the Zx code SCROLL_DONE control character. When disabled, the unit will no longer transmit a SCROLL_DONE control code when the unit is operating in scroll mode with CONFIG switch in PRGM position.

```
***************MAIN MENU***************

Enter 'A' to set User Programmed Baud Rate
Enter 'B' to set User Programmed RxIN EOL Terminator
Enter 'C' to change User Programmed Brightness Mode Settings
Enter 'D' to Enable/Disable User Programmed Display Scroll Mode
Enter 'E' to change User Programmed Scroll Speed
Enter 'F' to Enable/Disable Data Go/Stop Mode
Enter 'G' to Enable/Disable Software Reset
Enter 'H' to Enable/Disable Scroll Done Flag
Enter 'I' to create a new Custom Character
Enter 'J' to Escape


Awaiting Entry:
H
Enter A to enable Scroll Done Flag, or B to disable it:
A
 Scroll Done Flag is Enabled.
Reset and applied changes.Exiting menu.
```

2.10 Create a New Custom Character

An “I” entry in the Main Menu will bring up the “5x7 Character Writing Menu”:



This menu creates a guided dialog in which the user can create a custom character without writing code. In the first part of the dialog, the user is prompted to enter an ASCII number for the character. In the second part of the dialog, the user is asked to enter 5 digits of either ones or zeros, for each character row, of which there are 7. To help understand the principals involved in the row data, the chart below provides an example of row data to be entered for a “heart” symbol:

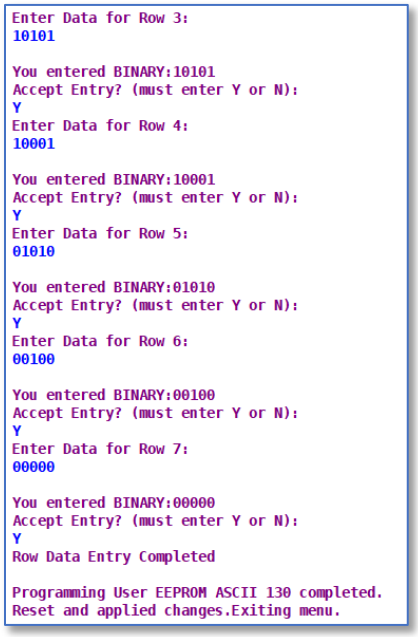
	COL 1	COL 2	COL 3	COL 4	COL 5		
ROW 1	0	0	0	0	0		ROW 1 DATA: 00000
ROW 2	0	1	0	1	0		ROW 2 DATA: 01010
ROW 3	1	0	1	0	1		ROW 3 DATA: 10101
ROW 4	1	0	0	0	1		ROW 4 DATA: 10001
ROW 5	0	1	0	1	0		ROW 5 DATA: 01010
ROW 6	0	0	1	0	0		ROW 6 DATA: 00100
ROW 7	0	0	0	0	0		ROW 7 DATA: 00000

Data for Row 3 can be described as follows:

Row 3 COL1 COL2 COL3 COL4 COL5 Data: 10101

A “1” turns that pixel on, a “zero” turns it off. In all cases the “msb” for the entry represents the state of the Column 1 pixel, the “lsb” entry represents the state for the Column 5 pixel, with Column 5 being the rightmost pixel in the character. Row 1 is the top row, and Row 7 is the bottom row.

The menu exits after the last row entry is completed:



The custom character glyph is stored in the on-board EEPROM and is now available for use. Refer to Section 5 of this document for information on how to display the custom characters. After completing entry, push the MENU button again to make more selections.

2.11 Uploading New sketches after setting configuration

After completing all configuration settings or custom character selections and exiting the UI, you may proceed to upload new sketches, such as the Demonstration sketch.

Before uploading, it is essential to close the serial terminal connection. The Arduino cannot load a new sketch while the COM port is still in use by a terminal emulator.

If using YAT on Windows: select **Terminal → Close/Stop**. **If using WCHSerialPort on macOS:** click **Disconnect**, then upload the new sketch from the Arduino IDE.

3.0 Using Zx Codes™ (Command Set Reference)

3.1 Zx Code Charts

Zx Codes- Rx IN. These must be sent from host with EOL terminators.			
Decimal	ASCII Mnemonic	Opcode Name	Description
12	FF	CLS	Clear Screen
18	DC2	SCROLL_SPD	Scroll Speed Control Prefix
24	CAN	RESET	Software Reset
28	FS	SCROLL_ON	Enable Scrolling Mode
29	GS	SCROLL_OFF	Disable Scrolling Mode
30	RS	EXTPWM_ON	Enable External PWM
31	US	EXTPWM_OFF	Disable External PWM

Zx Codes - Tx OUT. These are sent to host without EOL terminators.			
Decimal	ASCII Mnemonic	Opcode Name	Description
5	ENQ	DATA_GO	Message data send GO flag
6	ACK	DATA_STOP	Message data send STOP flag
20	DC4	SCROLL_DONE	Message scrolling done flag

3.2 Overall Protocol Concept

The scrolling system operates according to two fundamental behaviors:

a) If the display finishes scrolling a message and no new message has been received, the display will repeat the same message.

b) While a message is actively scrolling, the host may transmit the next message at any time within a wide timing window. When the current scroll completes, the display automatically begins scrolling the next message stored in the serial receive buffer.

From these behaviors, the following requirements are implied:

c) To prevent the display from repeating the same message, the next message must be transmitted before the current scroll completes.

d) Message data must be transmitted in the correct real-time order, ensuring that new data does not overwrite the next pending message.

3.2.1 Display-Host Synchronization

When the display is powered on in scrolling mode, it immediately begins transmitting the following Zx Codes on the TX line:

- DATA_GO
- DATA_STOP
- SCROLL_DONE

These codes are emitted continuously as long as scrolling mode is active, even when the display is only scrolling the default blank placeholder message. In this state, “Message 1” (flow chart below) may represent either an actual message or the blank sequence. The flow chart below, Fig. 3.1, depicts the real-time sequencing of data for a scrolling text transmission. It models the function in Appendix A.1.

The host uses these codes to coordinate message transmission:

1. When the display transmits DATA_GO, the host sends the serial data for Message #2.
2. The host then waits for SCROLL_DONE corresponding to Message 1, before returning the function call that sent Message #2 data. This ensures proper real-time sequencing of text transmissions.
3. Having sent SCROLL_DONE for Message #1, the display then begins scrolling Message #2.
4. The host then waits for the next DATA_GO, transmits Message #3, and the sequence continues.

Ensuring that each scroll-write function does not return until SCROLL_DONE is received maintains strict real-time sequencing and prevents message data from being overwritten or lost.

Note: The host firmware example here does not use the DATA_STOP marker for internal processing. It remains defined in the protocol to provide an explicit end-of-message delimiter for hosts that require framed message boundaries.

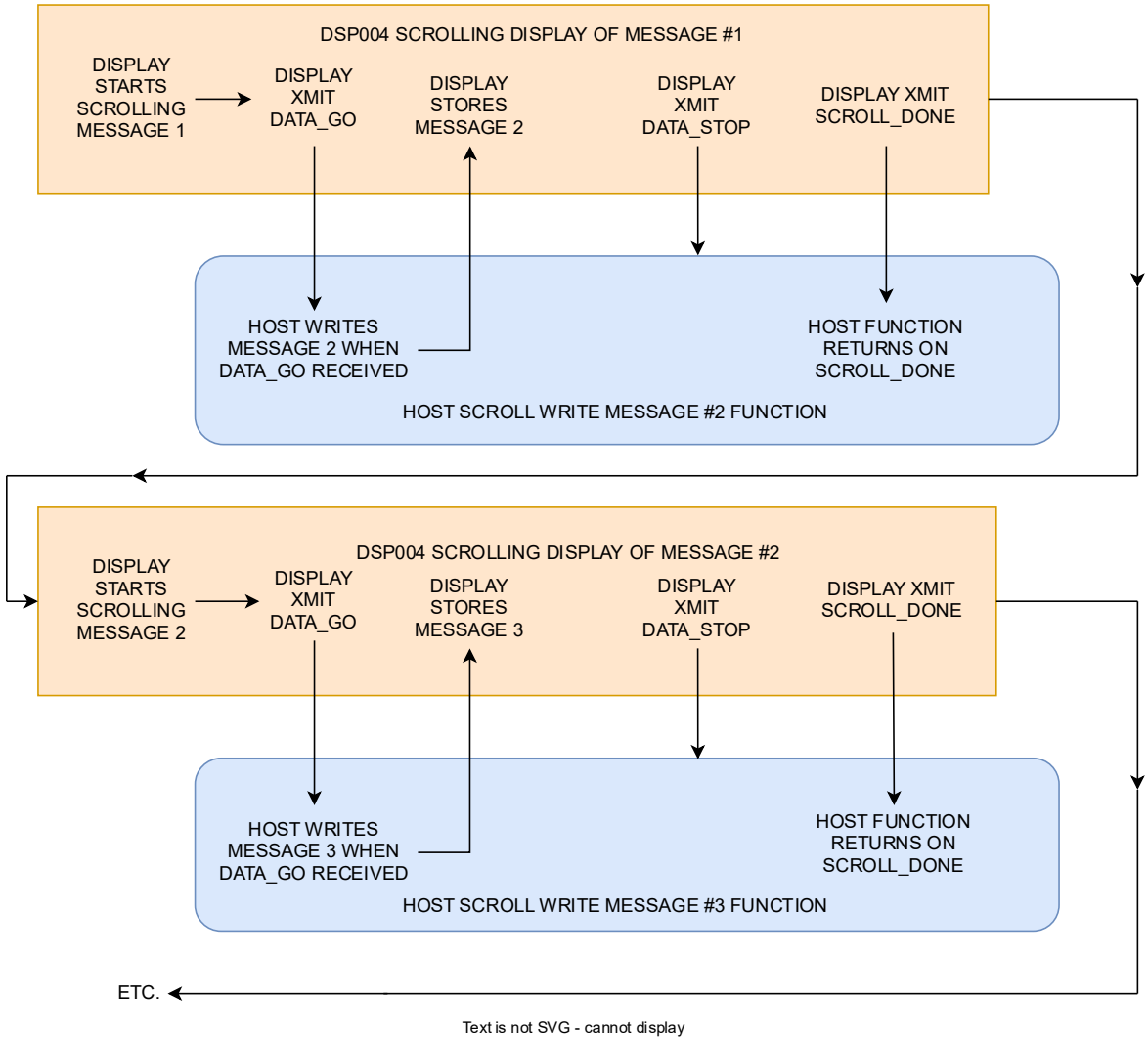


FIGURE 3.1 – REAL-TIME MESSAGE WRITING SEQUENCE IN SCROLLING WRITE OPERATION

3.2.2 Adding Zx Codes for Message-level settings control

Zx Codes can be incorporated directly into high-level message-handling functions for both scrolling and non-scrolling modes. When these functions are implemented in accordance with the operating principles defined in this section, the host can transmit messages in the correct real-time sequence while applying scroll-speed and brightness controls with per-message granularity.

Although the examples and demonstration code use blocking Arduino-style loops, the underlying sequencing rules apply equally to non-blocking architectures. Developers may implement these rules using timers, state machines, or event-driven logic as appropriate for their application.

All the Zx Codes except for the SCROLL_OFF code, take effect immediately upon receipt by the display.

Because those commands apply instantly, those Zx codes must be transmitted only after the previous message has fully completed scrolling; sending those Zx codes during an active scroll could alter the appearance or behavior of the message currently on the display.

To support proper timing for Zx codes, the display automatically appends a blank character to the end of every scrolling message. A SCROLL_END flag is generated at the moment this blank character begins appearing in the leftmost display position. During this time, the display is blank: this permits brightness modulation or scroll speed adjustments to be made before the upcoming text is displayed.

At the fastest scroll speed, the blank character remains on the display for approximately 138 milliseconds. *This interval defines the minimum time during which Zx codes may be transmitted for the upcoming message.*

The display unit requires a minimum 30 ms interval between data transmissions to ensure all incoming information is fully parsed and processed. If sending back-to-back Zx codes, allow 30 ms after each Zx code's EOL terminator. If a Zx code is sent after a text write in non-scrolling mode, ensure a minimum 30 ms delay from the text write's EOL terminator before transmitting the Zx code.

3.2.3 SCROLL_OFF Zx Code- Special Case

SCROLL_OFF is the only Zx code that does not take effect immediately. It must be transmitted between DATA_GO and DATA_STOP flags so the scrolling mode can wind down properly. Once this command is sent, the display finishes the current scrolling message before turning scrolling mode off.

The flowchart below illustrates how Zx code commands are incorporated into a scrolling message flow structure. It also includes a host "SCROLL_DONE" function at the end, which uses the SCROLL_OFF Zx code to turn off scrolling mode operation of the display.

The function in the chart "HOST SCROLL WRITE MESSAGE #2" models the function found in the Appendix A.1, a "Basic Scroll Write Function". The function in the chart "HOST SCROLL_OFF FUNCTION" models the function found in Appendix A.2, and the function in the chart "HOST SCROLL_OFF FUNCTION" models the function found in Appendix A.3.

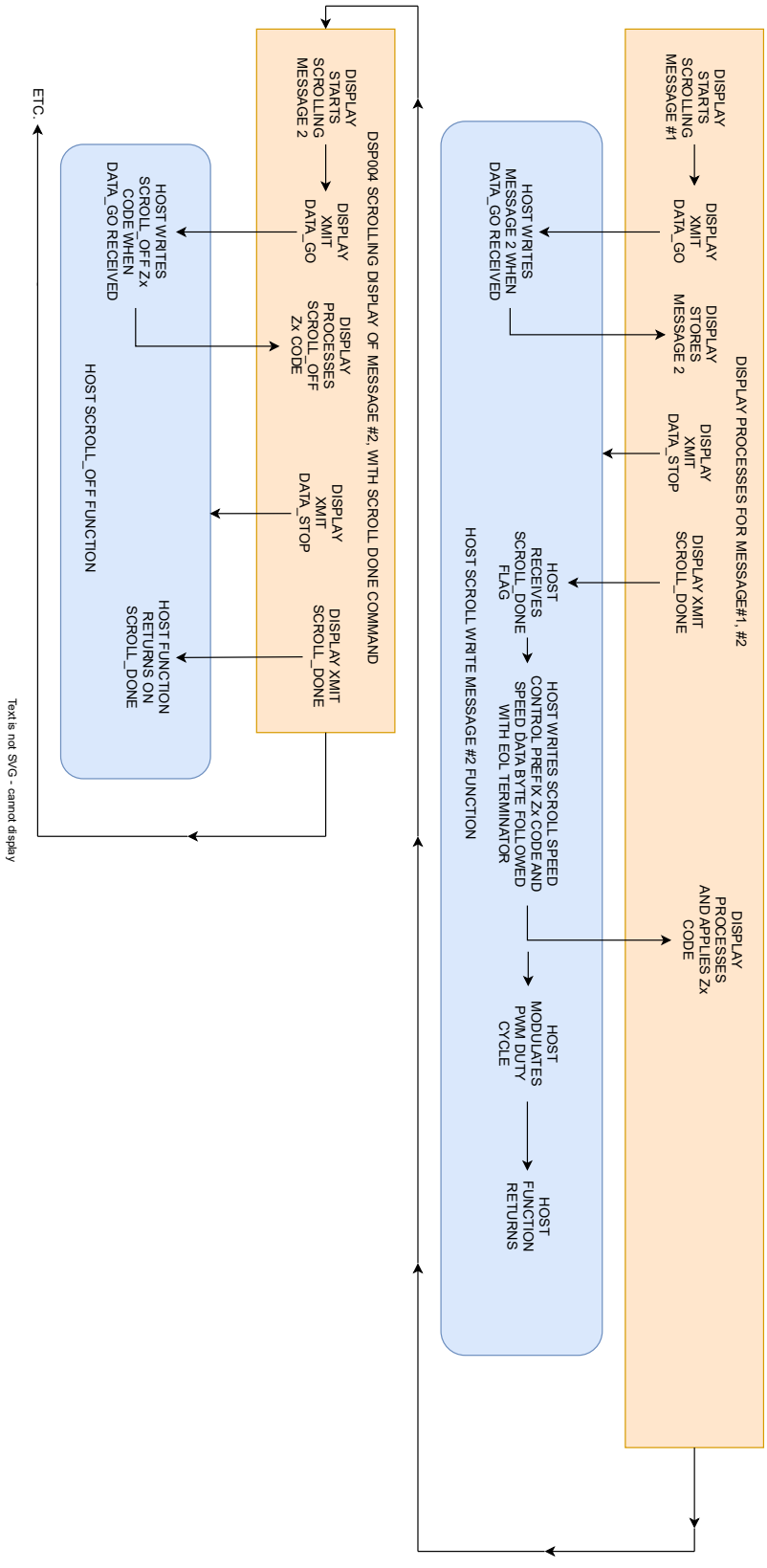


FIGURE 3.2 – REAL-TIME MESSAGE FUNCTION DETAIL SHOWING INCLUSION OF Zx CODES

3.3 Zx Codes Detailed functional descriptions and summary

Note: All Zx Codes transmitted from the host to the display's Rx input must be terminated with the same configured EOL sequence that is used for the transmitted text. The host's chosen EOL terminator must match the display's operating configuration so the display can correctly parse incoming commands.

3.3.1 CLS Clear Screen

Clears the display and serial buffers immediately in both scrolling and non-scrolling modes. All configuration settings remain unchanged.

- In scrolling mode, the current message is cleared immediately.
- After clearing in scrolling mode, the display begins transmitting DATA_GO, DATA_STOP, and SCROLL_DONE flags in preparation for receiving new scrolling-mode message data.
- Useful for priority-interrupt of an ongoing scrolling message.

3.3.2 SCROLL_SPD – Scroll Speed Control Prefix

After sending the SCROLL_SPD command, the host must immediately transmit a scroll-speed value from 15 to 114, followed by the EOL terminator.

- Out-of-range values are not interpreted as valid scroll-speed commands.
- If no speed value is sent, the speed will assume the numeric speed entry value 32.
- Scroll Speed command and speed value must be sent after SCROLL_DONE flag has been received by the host computer.

3.3.3 RESET – Software Reset

Initiates a full reset sequence identical to pressing the hardware reset button or asserting the reset pin.

- The serial buffer is flushed and the display is cleared.
- If the unit is in PRGM mode, all configuration parameters revert to the defaults stored in EEPROM.
- Allow a 300mS setup time before sending any new data/commands to the display unit.
- Recommended to be sent during non-scrolling operation to retain control of real-time execution.

Software-Reset Command Disabled Mode Behavior

When the software-reset command is received while Reset Mode is disabled in the display module, the module does not perform a reset. Instead, it outputs a single **"?"** character to the display. This character serves as a diagnostic indicator that a reset command was issued but intentionally ignored due to the current configuration.

The appearance of this indicator allows engineers to quickly identify unintended or residual reset transmissions from the host system. Under normal operation, no reset commands should be sent when Reset Mode has been disabled. A visible "?" therefore provides an immediate

cue that the host firmware may still be issuing reset requests (for example, from a watchdog routine, initialization loop, or legacy code path).

3.3.4 SCROLL_ON – Enable Scrolling Mode

Places the display into scrolling mode.

- If already in scrolling mode, the command has no effect.
- When activated, the display clears and begins transmitting DATA_GO, DATA_STOP, and SCROLL_DONE flags.
- Functions issuing SCROLL_ON should not return until a SCROLL_END flag is received to maintain correct real-time program sequencing.

3.3.5 SCROLL_OFF – Disable Scrolling Mode

Places the display into non-scrolling mode.

- If received when display is already in non-scrolling mode, the display shows “?”.
- If sent while a scrolling message is active, the display finishes the current message, sends a final SCROLL_DONE, and then exits scrolling mode.
- SCROLL_OFF must be sent between DATA_GO and DATA_STOP flags.
- Functions issuing SCROLL_OFF should not return until a SCROLL_END flag is received to maintain correct real-time program sequencing.
- If a SCROLL_OFF function is called while the display is already in non-scrolling mode, the function will stall waiting for a DATA_GO flag, which is only generated during scrolling operation.

3.3.6 EXTPWM_ON – Enable External PWM

Enables brightness control via the PWM IN pin.

- Takes effect immediately.
- Overrides any EEPROM-configured brightness mode.
- When operating in this mode, ensure the EXT PWM Pin has a valid PWM signal applied, as the display is blanked when logic low level is applied to the PWM pin.

3.3.7 EXTPWM_OFF – Disable External PWM

Returns brightness control to the EEPROM-configured mode.

- Takes effect immediately.
- The PWM IN pin should be held at logic high or connected to HI-Z pin after this command is sent, to ensure proper display operation.

3.3.8 DATA_GO – Message data send GO flag

Indicates that the display is ready to receive message text and the SCROLL_OFF command. Transmission of these items may begin immediately after DATA_GO is received.

3.3.9 DATA_STOP – Message data send STOP flag

Indicates that message text or data or SCROLL_OFF command may no longer be sent. Additional text must wait until the next DATA_GO is received.

3.3.10 SCROLL_DONE – Message scrolling done flag

Indicates that the current scrolling message has finished.

- Sent when the appended trailing blank character reaches the leftmost display position.
- The duration of this blank-character interval depends on the configured scroll speed (138 ms at maximum scroll speed).
- During this interval, Zx Codes for scroll speed and brightness may be sent, as well as any external PWM signal adjustments. These changes apply to the upcoming message.

3.4 Arduino Sketch using Zx Codes

Refer to the documentation download link on the Pronto Electronic Displays website (www.prontodisplays.com) to download the Arduino demonstration sketch. It requires no custom libraries.

4.0 Programming the On-Board EEPROM

The on-board EEPROM can be programmed through the SDA and SCL pins on the J1 I/O connector. The display unit must be powered on during programming. The I²C bus frequency should be set to 100 kHz. Programming may occur at any point during device operation except when the device is undergoing a RESET. Ensure that the device is not in a reset condition while EEPROM write operations are active. The SDA and SCL lines are pulled up to the display +5 V power through 4.7 kΩ resistors.

Caution: *Write operations to the on-board EEPROM are intended only for initial development and device configuration. EEPROM writes must not be performed during normal device operation due to the limited lifetime write-cycle endurance of the on-board EEPROM.*

4.1 Configuration Settings by programming on-board EEPROM

Below is a memory map for the on-board EEPROM:

ADDRESS	FUNCTION	DATA DESCRIPTION
0-237	CUSTOM CHARACTER GLYPH DATA	ROW DATA FOR 34 CUSTOM CHARACTERS, ASCII 128-161
238-244	USER DEFINED	USER DEFINED
245	BAUD RATE CODE NUMBER	01=38400 02=31250
246	EOL TERMINATOR CODE NUMBER	01=<CR> 02=<LF> 03=<CR><LF> 04=<LF><CR>
247	BRIGHTNESS MODE CODE NUBER	01=AUTO MODE 02=FIXED BRIGHTNESS MODE 03=EXTERNAL PWM MODE
248	FIXED BRIGHTNESS LEVEL	INTEGER NUMBER 1-100
249	MIN BRIGHTNESS LEVEL	INTEGER NUMBER 1-100
250	MAX BRIGHTNESS LEVEL	INTEGER NUMBER 1-100
251	SCROLL ENABLE	01= SCROLLING DISPLAY 02= NON-SCROLLING DISPLAY
252	SCROLL SPEED	INTEGER NUMBER 15-114
253	DATA_GO/STOP ENABLE	01= ENABLE DATA_GO/STOP 02= DISABLE DATA_GO/STOP
254	SOFTWARE RESET ENABLE	01= SOFTWARE RESET ENABLE 02= SOFTWARE RESET DISABLE
255	SCROLL DONE FLAG ENABLE	01= ENABLE SCROLL DONE FLAG 02=DISABLE SCROLL DONE FLAG

Note addresses 245–255 define the device's operating configuration. These settings take effect when the CONFIG switch is placed in the PRGM position. Only integer values are required; no bit-masking or composite fields are used, as each address controls a single function. All address contents shown above are expressed in decimal format.

4.2 Programming Scroll Speed

Address 252 is reserved for the scroll speed code number, and should be an integer number >= 15 up to and including 114. The formula equation for the number entry to scroll speed (secs) is defined:

Approximate Scroll Speed (Seconds) = (0.0087) *(Number entry) + 0.008

where: Approximate Scroll Speed = the amount of time a character is displayed in one position.

Number entry= the integer number that is written to address 252.

Note that scroll speed calculation is approximate and should not be relied upon for critical system timing. The asynchronous scrolling engine cannot be synchronized with any external signal.

4.3 User-Defined EEPROM Addresses

Note that addresses 238 through 244 are available for user-defined applications. Some uses that may be programmed could include serial numbers, manufacturing documentation, etc.

4.4 Addresses for Custom Characters

The memory data location for custom ASCII characters is organized into individual addresses in memory based on what the ASCII number for that character is. For example, for ASCII 128, data for Row 1 will start at address zero, Row 2 data at address 1, Row 3 data at address 2, and so on, with Row 7 data occupying address 6. The chart below may be used for correctly loading custom character glyph data into memory, if directly programming the on-board EEPROM:

CHARACTER ASCII #	ADDRESS FOR ROW 1 DATA		CHARACTER ASCII #	ADDRESS FOR ROW 1 DATA
128	0		145	119
129	7		146	126
130	14		147	133
131	21		148	140
132	28		149	147
133	35		150	154
134	42		151	161
135	49		152	168
136	56		153	175
137	63		154	182
138	70		155	189
139	77		156	196
140	84		157	203
141	91		158	210
142	98		159	217
143	105		160	224
144	112		161	231

4.5 Out-of-range ASCII character indicator

The display will indicate a question mark “?” if it parses a single ASCII character sent which is not a Zx code which is less than ASCII 32, or if the character is greater than ASCII 161.

5.0 Troubleshooting Table

Symptom	Probable Cause	Corrective Action
1. Display is blank and unresponsive	a) Menu button has been pushed-awaiting entry	a) Push the Reset button
	b) PWM Pin is at Logic Low	b) When not used, PWM pin should be unconnected, or connected to a digital pin set to logic high or Hi-Z mode.
	c) Unit is operating in non-scrolling mode but is being written to with scrolling-mode programming	c) Check CONFIG switch is in PRGM position, and that unit is programmed to operate in Scrolling mode.
2. Unit stays at maximum brightness level	a) Unit operating in EXT PWM mode	a) Apply PWM signal to modulate brightness, or change operating modes.
3. Characters not displayed near end of message in scrolling mode	a) Too many characters have been transmitted	a) Unit accepts messages of maximum 40 characters in length.
	b) Wrong EOL terminator	b) Check unit settings to ensure proper EOL terminator selection.
4. Display appears, but after a short delay	a) Transmission EOL terminator is missing/wrong	a) Check your transmission settings to ensure the correct EOL terminator is being sent.
5. OVP trips	a) Overvoltage applied to VIN pin	a) Verify power supply voltage and supply current is in accordance with the datasheet specifications.
	b) Unit operating in a too-warm environment	b) Remove unit to a cooler location and lower operating voltage if required

Symptom	Probable Cause	Corrective Action
6. Display flashes and/or pulsates	a) PWM signal applied with unit not configured to operate in EXT PWM mode b) PWM signal frequency is outside recommended frequency range	a) Ensure CONFIG switch is in PRGM position; also ensure PRGM mode operation is set to EXT PWM. b) PWM frequency should be 5-9kHz.
7. Menu does not appear	a) CONFIG switch in PRGM position b) Incorrect Baud rate or EOL terminator	a) Set CONFIG switch to NORM position b) Verify serial emulator is properly configured
8. Characters not displayed near end of message in scrolling mode	a) Too many characters have been transmitted b) Wrong EOL terminator	a) Unit accepts messages of maximum 40 characters in length. b) Check unit settings to ensure proper EOL terminator selection.
9) Characters garbled/inconsistent	a) Incorrect Baud rate b) Incorrect EOL Terminator c) Intermittent connections d) Signal integrity problem	a) Verify unit is operating with correct Baud rate setting b) Verify unit is operating with correct EOL Terminator setting c) Verify Rx, Tx and GND connections are not loose, intermittent or missing d) Inspect all signals for signal integrity problems such as excessive ringing or overshoot and take remedial measures as required

6.0 Mounting Holes, Cutout Area and Mounting Hardware

NOTE: This drawing assumes that the mounting hardware and procedure used are the same as those specified in the *Bezel/Window and Mounting Hardware Kit Instructions*. Use of different hardware or alternative mounting procedures may change the mechanical stack-up; in these cases, dimensional tolerances may require adjustment to ensure proper fit and alignment.

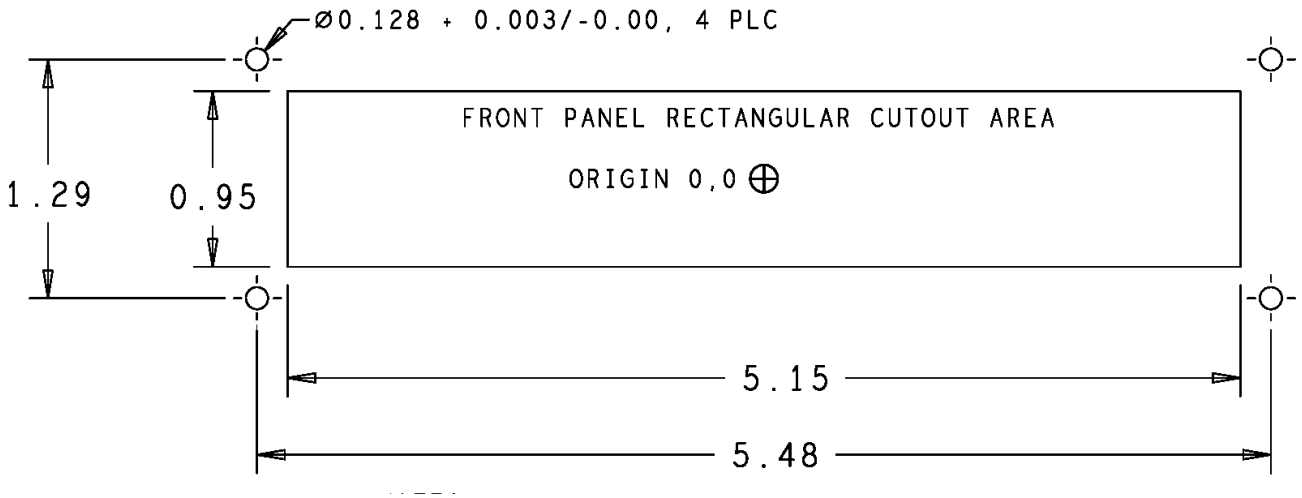


FIGURE 6.1 – MOUNTING HOLE DIMENSIONS, LOCATIONS AND RECOMMENDED CUTOUT AREA

Notes:

1. All dimensions are in Inches.
2. Dimension origin is at the center of the panel cutout.
3. Hole pattern and cutout dimensions are symmetrical about the drawing origin.
4. All hole location dimensions are +/- 0.005 inches.
5. Center cutout width and height dimensions are +/- 0.005 inches.
6. 30-60 mil radius on cutout inner corners, optional.
7. Cutout area includes functional clearance for the on-board light sensor.

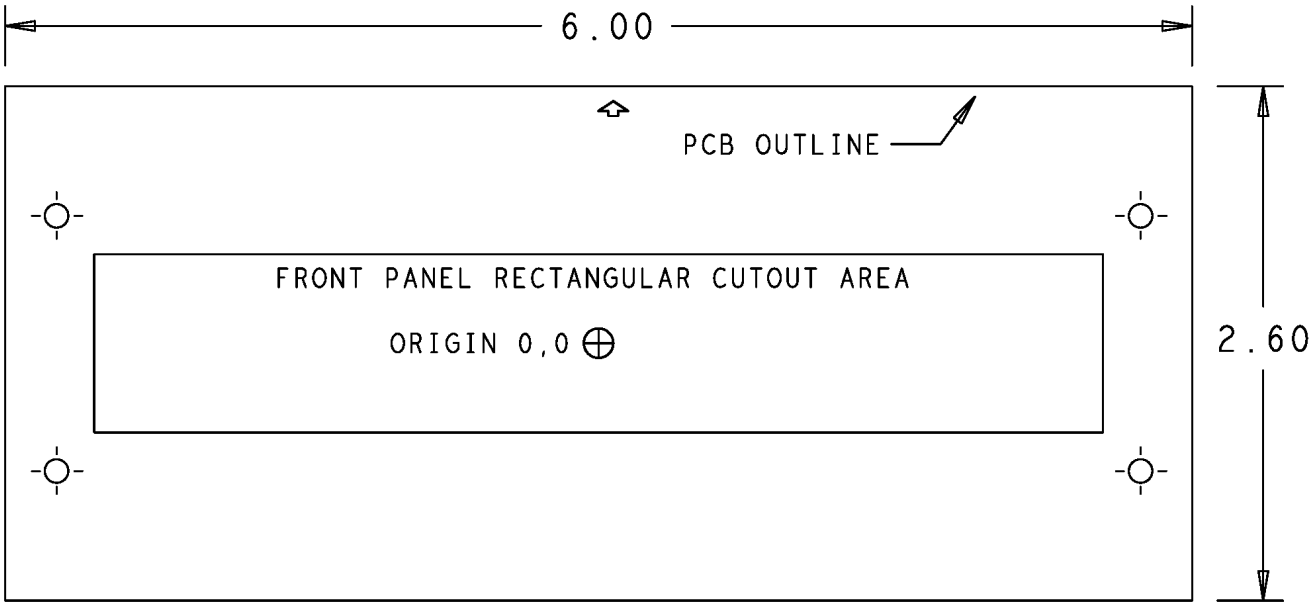


FIGURE 6.2 – PCB MOUNTING GEOMETRY, PANEL REFERENCE

Notes:

1. All dimensions are in Inches.
2. Dimension origin is at the center of the panel cutout.
3. PCB outline dimensions are symmetrical about the drawing origin.
4. PCB outline dimensions are held to +/- 0.020 inches.

Appendix A- Arduino Reference Programming

A.1 Basic "Scroll Write" function

Note: The helper function writeEOL(termMode) sends one of the four supported EOL terminators selected by the user. See Appendix section A.4

```
// This function writes a scrolling message using default PRGM Mode settings for
brightness, speed and PWM mode.
void scrollWrite (const char* msg )
{
bool readflag1 = false;

  while (readflag1 == false)// Will continue reading
  {
    if (mySerial.available()>0)
    {
      byte controlByte= mySerial.read();
      if (controlByte == 5)// 5= DATA_GO
      {
        readflag1 = true;// Breaks loop when received DATA_GO
        mySerial.write(msg);
        writeEOL(termMode);
      }
    }
  }
  readflag1 = false;

    while (readflag1 == false)// Will continue reading
  {
    if (mySerial.available()>0)
    {
      byte controlByte= mySerial.read();
      if (controlByte == 20)// 20= SCROLL_DONE
      {
        readflag1 = true;// Breaks loop when received SCROLL_DONE
      }
    }
  }
}// Function returns after receiving SCROLL_DONE flag
```

Sample function call: scrollWrite("Hello World!");

A.2 "Scroll Write" function including Zx code SCROLL_SPD

```
// This function writes a scrolling message and adjusts speed and brightness for
it.
// This function is meant to be used only when display EXT PWM Mode is enabled,
otherwise display will blink.

void scrollWrite_spd_brite (const char* msg, uint8_t speed, int brite )
{
bool readflag1 = false;

if (brite<0 || brite>255)
{ brite=22;}// input validation. Set to 22 if entry out of bounds.

  while (readflag1 == false)// Will continue reading
  {
    if (mySerial.available()>0)
    {
      byte controlByte= mySerial.read();
      if (controlByte == 5)// 5= DATA_GO
      {
        readflag1 = true;// Breaks loop when received DATA_GO
        mySerial.write(msg);
        writeEOL(termMode);
      }
    }
  }
  readflag1 = false;

    while (readflag1 == false)// Will continue reading
  {
    if (mySerial.available()>0)
    {
      byte controlByte= mySerial.read();
      if (controlByte == 20)// 20= SCROLL_DONE
      {
        readflag1 = true;//
        mySerial.write(18);// Send Scroll Speed Control Prefix
        mySerial.write(speed);// Send the scroll speed data byte
        writeEOL(termMode);// Send EOL terminator
        analogWrite(pwm_Pin, brite);// host-side brightness modulation
      }
    }
  }
}// Function returns after receiving SCROLL_DONE flag
```

Sample function call: scrollWrite_spd_brite("Hello World!", 30,85);

A.3 “scrollModeOff()” function

```
// This function will turn scroll mode off. This Zx code must be sent between
DATA_GO and DATA_STOP Zx code flags.

void scrollModeOff()
{
  bool readflag1 = false;

  while (readflag1 == false)// Continue to wait for DATA_GO
  {
    if (mySerial.available()>0)
    {
      byte controlByte= mySerial.read();
      if (controlByte == 5)// 5= DATA_GO
      {
        readflag1 = true;
        mySerial.write(29);// Send scroll disable command character
        writeEOL(termMode);// Write EOL Terminator
      }
    }
  }
  readflag1 = false;

  while (readflag1 == false)
  {
      if (mySerial.available()>0)
      {
        byte controlByte= mySerial.read();
        if (controlByte == 20)// 20= SCROLL_DONE
        {
          readflag1 = true;// Breaks loop when received SCROLL_DONE flag
        }
      }
  }
}// Function returns when received SCROLL_DONE
```

Sample function call: scrollModeOff();

A.4 "writeEOL()" Function

```
//*** Global Variables for selecting EOL Terminator Used ***
// Un-remark below as required depending on your particular EOL terminator
configuration.

char termMode = 'A'; // Select this for <CR>
//char termMode = 'B'; // Select this for <LF>
//char termMode = 'C'; // Select this for <CR><LF>
//char termMode = 'D'; // Select this for <LF><CR>



void writeEOL( char term_mode)
{
  if (term_mode == 'A')
  {
    mySerial.write (13);// EOL term equals <CR>
  }
    if (term_mode == 'B')
  {
    mySerial.write (10);// EOL term equals <LF>
  }
    if (term_mode == 'C')
  {
    mySerial.write (13);// EOL term equals <CR><LF>
    mySerial.write (10);
  }
    if (term_mode == 'D')
  {
    mySerial.write (10);// EOL term equals <LF><CR>
    mySerial.write (13);
  }
}
```

A.5 Terminal Bridge Program

```
// Terminal Bridge program
// This program allows the user to access and interact with the integrated UI
// When done, close the serial port before attempting to upload other sketches
// Set CONFIG switch to NORM


#include <SoftwareSerial.h>

// Isolate the peripheral to digital pins 2 and 3.
// This completely separates it from the Arduino hardware USB serial lines (Pins 0
and 1).
// RX = 2, TX = 3. DSP004 pin 3 to Arduino pin 3, DSP004 pin 2 to Arduino pin 2.
SoftwareSerial peripheralSerial(2,3);

void setup() {
  // 1. Initialize the USB connection to YAT on your PC.
  // Set YAT's baud rate to 38400 to match this.
 Serial.begin(38400);
 //Serial.begin(31250);
  // 2. Initialize the connection to your peripheral device.
  // Change '38400' to match your specific peripheral's required baud rate.
  peripheralSerial.begin(38400);
  //peripheralSerial.begin(31250);

}

void loop() {
  // PATH A: Peripheral ---> PC (YAT Screen)
  // Check if the peripheral device has sent any data back to the Arduino.
  if (peripheralSerial.available() > 0) {
    // Read the character from the peripheral and pass it strictly to YAT.
    // Because it goes to 'Serial', it will NOT be looped back out to the
peripheral TX line.
    Serial.write(peripheralSerial.read());
  }

  // PATH B: PC (YAT Typing) ---> Peripheral
  // Check if you have typed anything into your YAT terminal window.
  if (Serial.available() > 0) {
    // Read your keystroke from YAT and pass it strictly out to the peripheral
device.
    peripheralSerial.write(Serial.read());
  }
}
```

Document Revisions		
Rev. No.	Date	Engr.
03	9/23/2026	M. Hawkins

END OF DOCUMENT