# DSP004 Intelligent Character Display
# User Manual

Pronto Electronic Displays LLC
www.prontodisplays.com

Doc.# 1031-DSP004-M REV.02

# Table of Contents

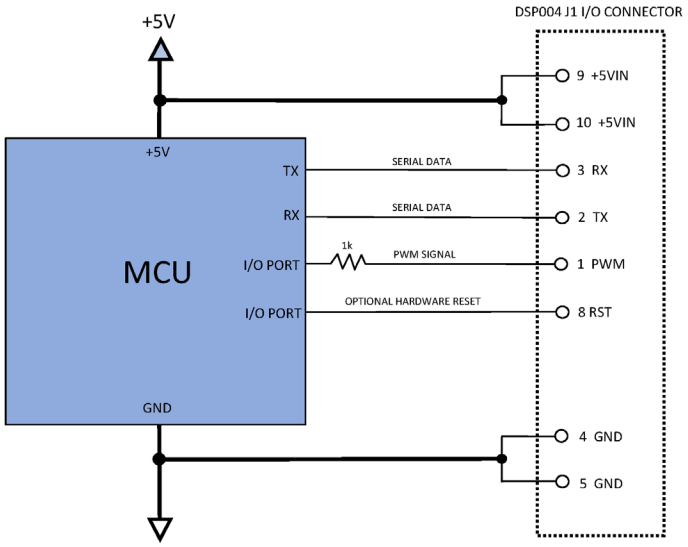
# 1.0 Theory of Operation

## 1.1 Establishing Communication

Communication with the device is established using Serial UART signals. The device supports signals using 5.0V logic. For development purposes, VCP (Virtual COM Port) drivers may be used to establish serial communications to the display UI using USB, with Serial Terminal Emulators providing a user interface through which the user can interact with the device. Refer to document #1032-DSP004-QSG, Quick Start Guide on how to establish communications and configure the serial terminal emulator.

## 1.2 Typical Installation /Application Setup

Once the device has been configured either with the built-in UI and a Serial Emulator(See QSG) or by direct programming of the on-board EEPROM, connect power, ground, RX, TX and PWM pins as shown below. Use a 1k nominal resistor in series with the PWM pin to keep the signal free of excessive ringing and overshoot which will cause malfunctions.



## 1.3 Operating Modes

The unit has two basic modes of operation, being determined by the position of the CONFIG switch. In the NORM position, the unit operates in a fixed factory default consisting of the following settings:

Baud rate= 38400

RXIN EOL Terminator = <CR>

TXOUT EOL Terminator = <CR><LF>

Brightness Mode = Automatic, with minimum brightness 10%, and maximum brightness 100%

Display set to Non-Scrolling Mode

Control Characters Disabled

Software Reset Enabled

Additionally, in the NORM position the UI menu can be accessed by pushing the Menu button; the MENU button is disabled when the CONFIG switch is in the PRGM position. The UI Menu purpose is to permit changing configuration settings, or writing custom characters. The user-programmed operating settings will take effect when the CONFIG switch is in the PRGM position. The UI Menu can be used to:

a) Change Baud rate settings.
b) Change EOL Terminator settings for RXIN.
c) Change Brightness mode/settings.
d) Change Scroll speed settings and scroll enable/disable.
e) Enable/Disable Data Go/Stop control characters.
f) Enable/Disable Scroll Done Flag
g) Enable/Disable Software Reset
h) Design up to 34 custom characters.

The intent of the CONFIG switch is to ensure the user is always able to re-establish communication with the device, by switching to known-working default settings should a user-programmed configuration not perform as expected, or disrupt communications.

## 1.4 On-Board EEPROM

In addition to changing the settings with the UI, it is possible to directly write the settings, including custom characters, to the board, bypassing the UI and thereby allowing high-volume automated programming. This is accomplished using the SDA and SCL pins provided on the I/O connector, which are pin 6 (SCL) and pin 7 (SDA). These pins have internal 4.7k pullups to the board +5V. A thorough treatment of using and programming the on-board EEPROM may be found in Section 4.

## 1.5 Controlling Display Brightness

The unit has three different modes of display brightness control, and these can be set as the default state the unit will operate in when the CONFIG switch is set to PRGM mode.

1) Automatic mode with integrated ambient light sensor
2) Fixed brightness mode
3) External PWM, supplied through the PWM pin 1 on the I/O connector.
   Note external PWM operation may also be engaged by using Smarti™ commands.

## 1.6 Board Power Requirements

The board requires +5VDC with a maximum current requirement of 1.0A. Device current draw changes substantially depending on the illumination brightness level and the total number of LED dots that are on at any given time. For example, in relatively low-light office conditions, and with all standard characters being used, current draw can be 60 mA or less. On the other hand, in the worst-case scenario, which consists of all LED dots being illuminated at maximum brightness, close to 1A of current will be drawn. Input voltage range is 4.75 - 5.3V.

## 1.7 Power Conditioning Circuit

The unit has an on-board power conditioning circuit, consisting of enhanced ESD protection, reverse voltage protection, and overvoltage protection. There is no undervoltage protection.

If reverse voltage is applied to the board, the Green PWR led will not be illuminated, as no power will be applied to the board.

If an overvoltage condition is applied to the board power, the Green PWR led will go off, and the red OVP led will be illuminated; it will stay illuminated until the OVP circuit is reset. To reset the OVP protection and resume normal operation, first turn off the overvoltage power. Then set the power to the correct voltage, and re-apply power.

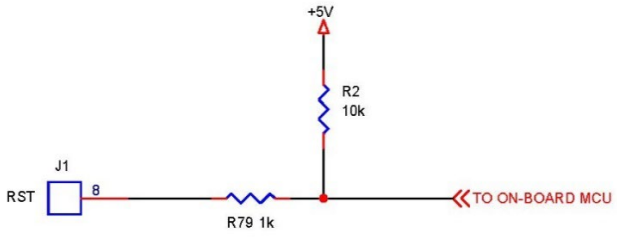
## 1.8 Reset

The unit can undergo a reset by the following conditions:

1. Momentary Power Loss.
2. Command by means of User Interface.
3. Pushing the Reset Button.
4. Changing the position of the “CONFIG” switch.
5. Pulling the Reset pin 8 on the I/O connector to logic low.
6. Sending a Software Reset Smarti™ command.

In all cases when the board is reset, the input serial buffer is flushed. Any previous display data is erased, and the display will show a blank display after the reset.

The reset pin 8 initiates a board reset when it is pulled to logic low. This pin is connected to +5V with an internal 10k pullup, and has a 1k resistor in series with it as shown below:




When controlling the reset pin with an MCU I/O port, ensure the port is either internally pulled up, or set to Hi-Z mode, when not active. When activating a reset function, ensure the pin is pulled low for a minimum of 10 milliseconds.

# 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:

```
*********TERMINAL CONNECTION SUCCESSFUL*********
         Firmware Rev.5.7.4

********OPERATING IN MENU CONFIGURATION********
1. BAUD RATE SET TO DEFAULT 38400
2. RxIn  EOL TERMINATOR SET TO DEFAULT <CR>
  (TxOut EOL Terminator is fixed at <CR><LF>)
3. BRIGHTNESS MODE SET TO AUTOMATIC LIGHT SENSOR:
    Max Display Brightness set to 100%
    Min Display Brightness set to 10%
4. DISPLAY SET TO NON-SCROLLING MODE.
5. CONTROL CHARACTER FUNCTIONS DISABLED.
6. SOFTWARE RESET ENABLED.

******** 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:
```

FIG.2, Headers and Main Menu

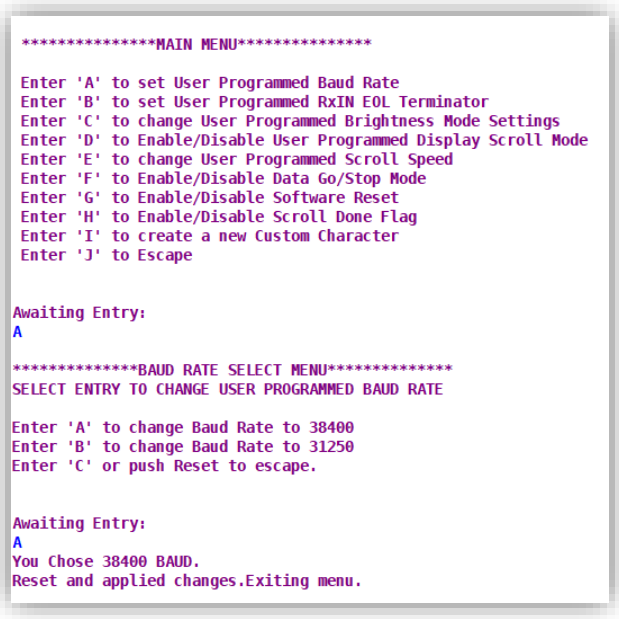
The “OPERATING IN MENU CONFIGURATION” header serves as a message to the user that the unit is currently operating in the default factory configuration. The settings for this configuration are listed in this header to serve as a reference for the user.

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:

```
***************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:
B

**************EOL TERMINATOR SELECT MENU**************
SELECT ENTRY TO CHANGE USER PROGRAMMED RxIN EOL TERMINATOR

Enter 'A' to change to <CR>
Enter 'B' to change to <LF>
Enter 'C' to change to <CR><LF>
Enter 'D' to change to <LF><CR>
Enter 'E' or push Reset to escape.


Awaiting Entry:
D
You Chose <LF><CR>
Reset and applied changes.Exiting menu.
```

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:

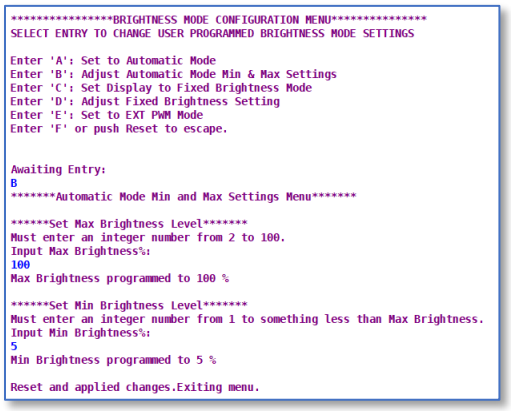
```
***************MAIN MENU***************

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:
C

***************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:
A
Light Sensor set to Automatic Mode.
Reset and applied changes.Exiting menu.
```

### 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. Push the MENU button again to make more selections.

### 2.4.2 Adjust Automatic Mode Min & Max Settings

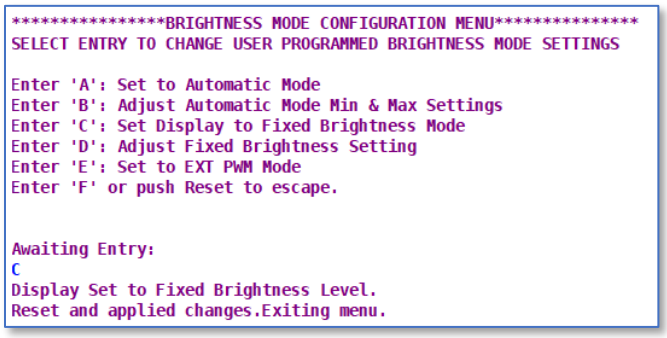
An “B” entry in the “Brightness Mode Configuration Menu” will bring up a dialog in which one may enter min and max brightness levels:




These levels represent a linear percentage of min and max internally-generated PWM duty cycle. Note that the human perception of brightness is more sensitive to lower light levels than to brighter ones, as described by the Weber-Fechner laws; for example, the level of *difference* perceived for the lower PWM percentages of 10-30% , will not be perceived as being the same amount of difference at higher PWM levels of 60-90%.The brightness minimum value determines how dim the display will appear under the dimmest lighting conditions, and the brightness maximum value determines how bright the display will appear under the brightest ambient light conditions. 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.




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 Smarti™ control codes. 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 controls the display scroll mode the unit will operate in as default mode when the CONFIG switch is in the PRGM position.

## 2.6 Change the User Programmed Scroll Speed

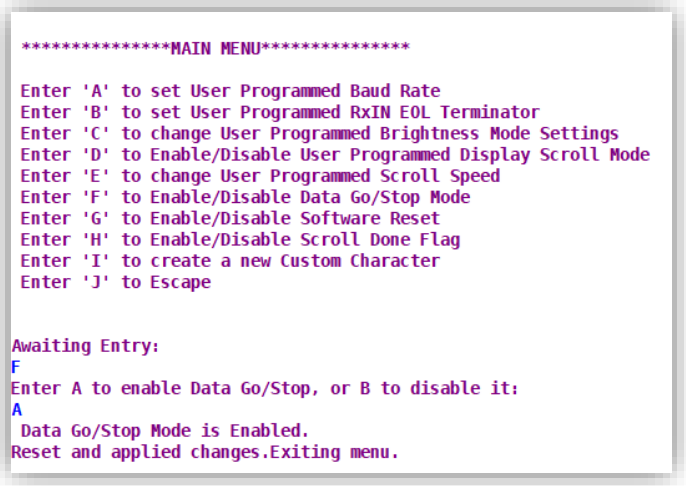
An “E” entry from the Main Menu will present a menu for setting up the Scroll Speed the unit will operate with when it is running in Scrolling mode:

```
***************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:
E
Enter scroll speed value, from 15 (fastest) to 114 (slowest):
22
Scroll speed set to 0.199 seconds.
Reset and Applied Changes.Exiting menu.
```

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 scroll speed in seconds. After completing entry, push the MENU button again to make more selections. When the UI menu is recalled, the scroll speed setting is displayed in seconds, along with the corresponding numerical entry that was made, in the “User Programmed Configuration Settings” header.

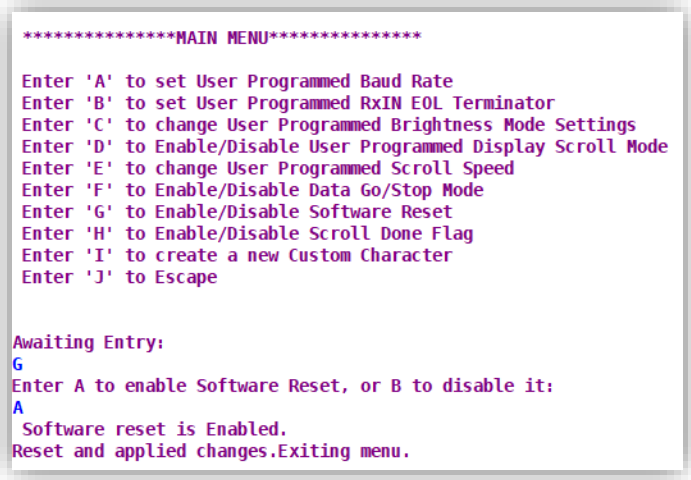
## 2.7 Enable/Disable Data Go/Stop Control Characters

An “F” entry from the Main Menu will present a menu for enabling or disabling the Smarti™ DATA_GO and DATA_STOP control characters. When disabled, the unit will no longer transmit these characters 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 Smarti™ 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.



## 2.9 Enable/Disable Scroll Done Flag

An “H” entry from the Main Menu will present a menu for enabling or disabling the Smarti™ 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”:

```
 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:
 I

 *************** 5x7 CHARACTER WRITING PROGRAM ***************

 34 Custom Characters are available, using ASCII numbers 128-161.

 Enter the ASCII Number:
 130
 You entered: 130
 Accept Entry? (must enter Y or N):
 Y

 ASCII Number 130 has been assigned to this character.

 *** ROW DATA ENTRY SECTION ****

 Entries must be made as a five-bit Binary number.
 Enter Data for Row 1:
 00000

 You entered BINARY:00000
 Accept Entry? (must enter Y or N):
 Y
 Enter Data for Row 2:
 01010

 You entered BINARY:01010
 Accept Entry? (must enter Y or N):
 Y
 Enter Data for Row 3:
```

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 principles 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 represent 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:

```
Enter Data for Row 3:
10101

You entered BINARY:10101
Accept Entry? (must enter Y or N):
Y
Enter Data for Row 4:
10001

You entered BINARY:10001
Accept Entry? (must enter Y or N):
Y
Enter Data for Row 5:
01010

You entered BINARY:01010
Accept Entry? (must enter Y or N):
Y
Enter Data for Row 6:
00100

You entered BINARY:00100
Accept Entry? (must enter Y or N):
Y
Enter Data for Row 7:
00000

You entered BINARY:00000
Accept Entry? (must enter Y or N):
Y
Row Data Entry Completed

Programming User EEPROM ASCII 130 completed.
Reset and applied changes.Exiting menu.
```

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.

## 3.0 Using External PWM for Brightness Control

The PWM pin is pulled to logic high by an internal 10k pullup resistor. Note if a PWM signal is supplied to this pin when the unit is not set to EXT PWM mode, the display may pulsate or go blank, due to interference with the user's PWM and the unit's internally-generated PWM signal which the unit utilizes when in Auto or Fixed Brightness modes. This is why it is important to ensure that no PWM signal is being applied to Pin 1 unless the unit is set for EXT PWM operation. When the CONFIG switch is set the PRGM position, the unit may then operate in EXT PWM mode as a default. Setting EXT PWM as the default may be done with the UI or by directly programming the on-board EEPROM.

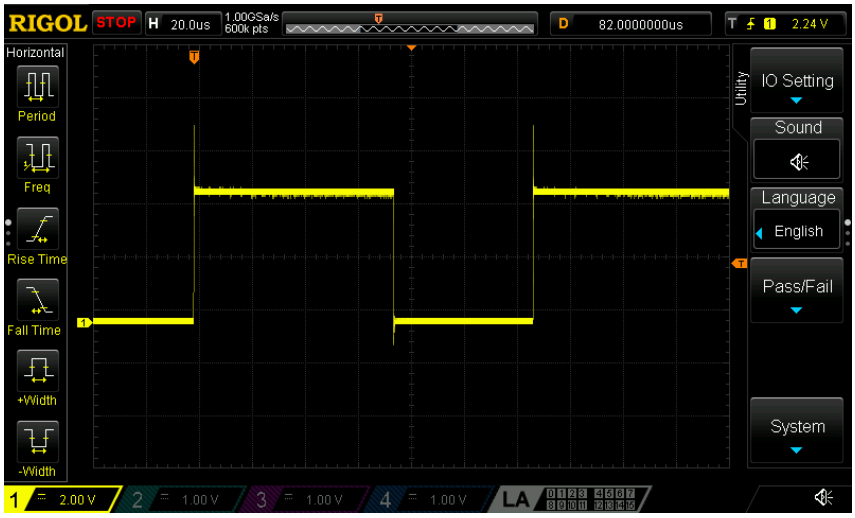
In most cases, EXT PWM is engaged to individual messages using Smarti™ commands, making PWM mode an awkward redundancy if it is set as the default mode. In most cases the default mode setting will be either automatic or fixed brightness mode.

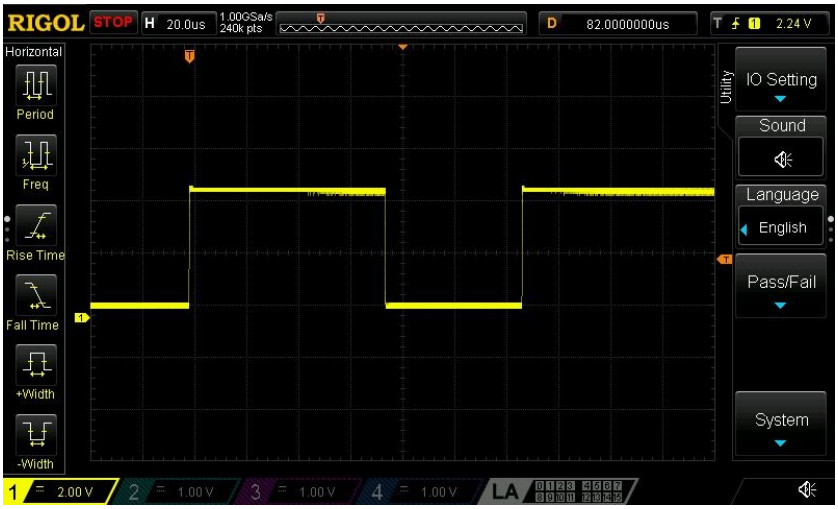
The PWM pin, if not used, should be disconnected. If it is connected to an MCU pin, then while not in use, the pins should either be set to a Hi-Z state, or set to Logic High voltage.

When the display is set to EXT PWM mode, the display will be at maximum brightness level when no external PWM signal is being applied.

The PWM signal applied should be 5V logic level with a frequency range of 6 to 9kHz. Note that frequencies outside this range may be applied; however there is no guarantee that any such frequencies will not create interference artifacts due to the display multiplex characteristics.

It is of utmost importance to ensure a 1k resistor is in series with the signal near the origin point of the PWM signal, as per the diagram in Section 1.2. This is because PWM pins tend to be open-drain outputs with very fast edges. As such, without some form of impedance matching, the unit is almost guaranteed to fail as a result of EMI caused by excessive ringing and overshoot generated by the PWM signal. Below of a screenshot taken at the PWM pin input, without the series 1k EMI suppression resistor. This signal may cause intermittent fails, manifested as missing or garbled characters:






The above screenshot shows the PWM signal input, with a series 1k resistor placed near the PWM generator output. Note the generally clean waveform with minimal overshoot, with no undershoot or ringing being present.

Below are two lines of Arduino Uno code which may be used to generate an 7.812 kHz PWM signal suitable for use with the display unit. Note these settings are only applicable to Arduino Uno which uses the ATMEGA328 microcontroller:

```
10    TCCR2B = 0b00000010; / prescale,  divide by 8
20    TCCR2A = 0b00000011; / configure timer 2 to PWM Fast mode
```

Lines 10 and 20 will create PWM signal in Fast PWM mode for Timer 2 at a frequency of 7.812 kHz, which will be available on Arduino Uno pins D3 and D11.

PWM can be generated from Arduino Uno pins D3 or D11 with the following commands:

analogWrite(11,pwm_factor) or analogWrite(3,pwm_factor)

Where : pwm_factor = an integer number from 0 to 255 with 0 being the lowest duty cycle (always off) and 255 being the highest duty cycle (always on).

When PWM is not being used and it is desired to allow the unit normal operation in other brightness modes such as Automatic or Fixed Brightness, set the PWM pin High by writing a PWM factor = 255. For more information on controlling brightness using PWM see Section 6 on using the Smarti™ suite of ASCII command codes.

## 4.0 Programming the On-Board EEPROM

The on-board EEPROM may be directly programmed by use of the SDA and SCL pins provided on the J1 I/O connector. Ensure the display unit is powered on when programming. The I2C frequency should be set to 100kHz. Programming can take place at any time during the device operation, with the exception of the device undergoing a RESET; although unlikely, ensure that the device is not RESET concurrently with EEPROM programming. The SDA and SCL pins are tied to board +5V power with 4.7k pullups. **Write operations to the on-board EEPROM should only be used for initial development / configuration of the device; EEPROM write operations must not be performed as part of the normal operating profile of the device due to the limited number of lifetime write cycles available for the on-board EEPROM.**

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

| ADDRESS NUMBER | 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 |

## 4.1 Programming Configuration Settings

Note addresses 245-255 are used for setting the operating configuration for the device when the CONFIG switch is placed in the PRGM position. Note also that only integer numbers need to be entered; no bit-masking or other complicated techniques are required, because each address represents only one functionality. The address contents above are shown in decimal values.

## 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 :

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

where: Scroll Speed = the amount of time a character is displayed.

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

## 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 NUMBER FOR ROW 1 DATA | | CHARACTER ASCII # | ADDRESS NUMBER 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 Code for EEPROM Programming

The on-board EEPROM is a Microchip Technology P/N 24AA024-I/SN. Consult the manufacturer datasheet for I2C programming details particular for this chip. Arduino code for reading/writing to the on-board EEPROM is supplied in the appendix for user reference.

# 5.0 Writing Serial Data to the Display using Arduino

This section offers some tips for sending data for display to the unit using Arduino. This section will show how to print basic serial data to the device using the various EOL terminator configurations available to the unit, as well as how to display custom characters. The brief descriptions of the Arduino print commands included here may be supplemented with more detailed information on the Arduino website.

Serial.println(“string”) : The text within quotes will be transmitted with <CR><LF> EOL terminators.

Serial.print(“string”) : The text within quotes will be transmitted without any EOL terminator.

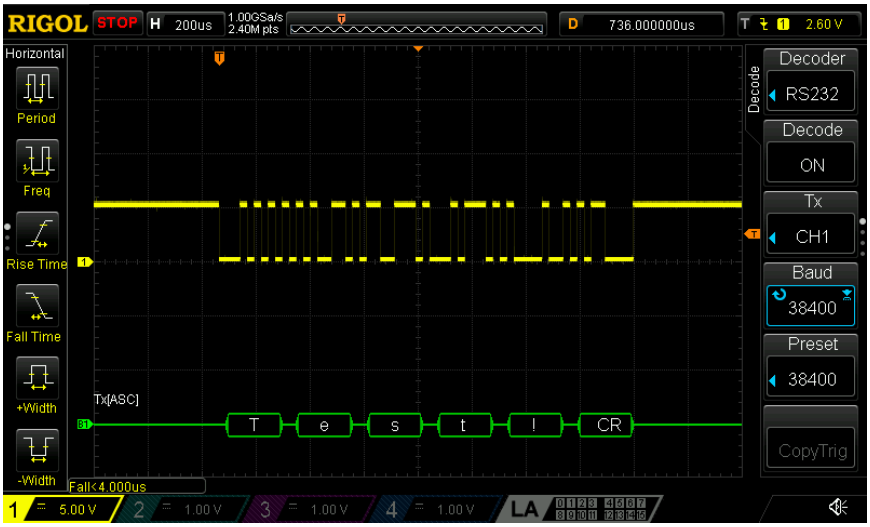
Serial.write(val) : The numeric value within parenthesis will be transmitted. This is useful for sending non-printable characters such as EOL terminators, or custom characters.

For example, in the below lines 10 and 20, the text “Test!” will be transmitted followed with <CR> for the EOL terminator:

10 Serial.print(“Test!”);

20 Serial.write(13);

The waveform below shows the serial signal being transmitted with the <CR> EOL terminator, using an oscilloscope with a protocol analyzer:




To display custom characters, use the Serial.write(val) command, where val= the ASCII number of the custom character. For example, the custom character ASCII 135 can be displayed with the command:

Serial.write(135);

Note that if serial data is transmitted with no EOL terminator, the unit will display the contents of the serial buffer after a short timeout interval.

# 6.0 Using Smarti™ Codes

This section explains what Smarti™ codes do, and how to implement them in your display programming.

## 6.1 Overall Software Flow Control Concept

First we may begin with an understanding of how the display unit operates in scroll mode. It comes down to understanding two concepts of how the display handles scrolling messages:

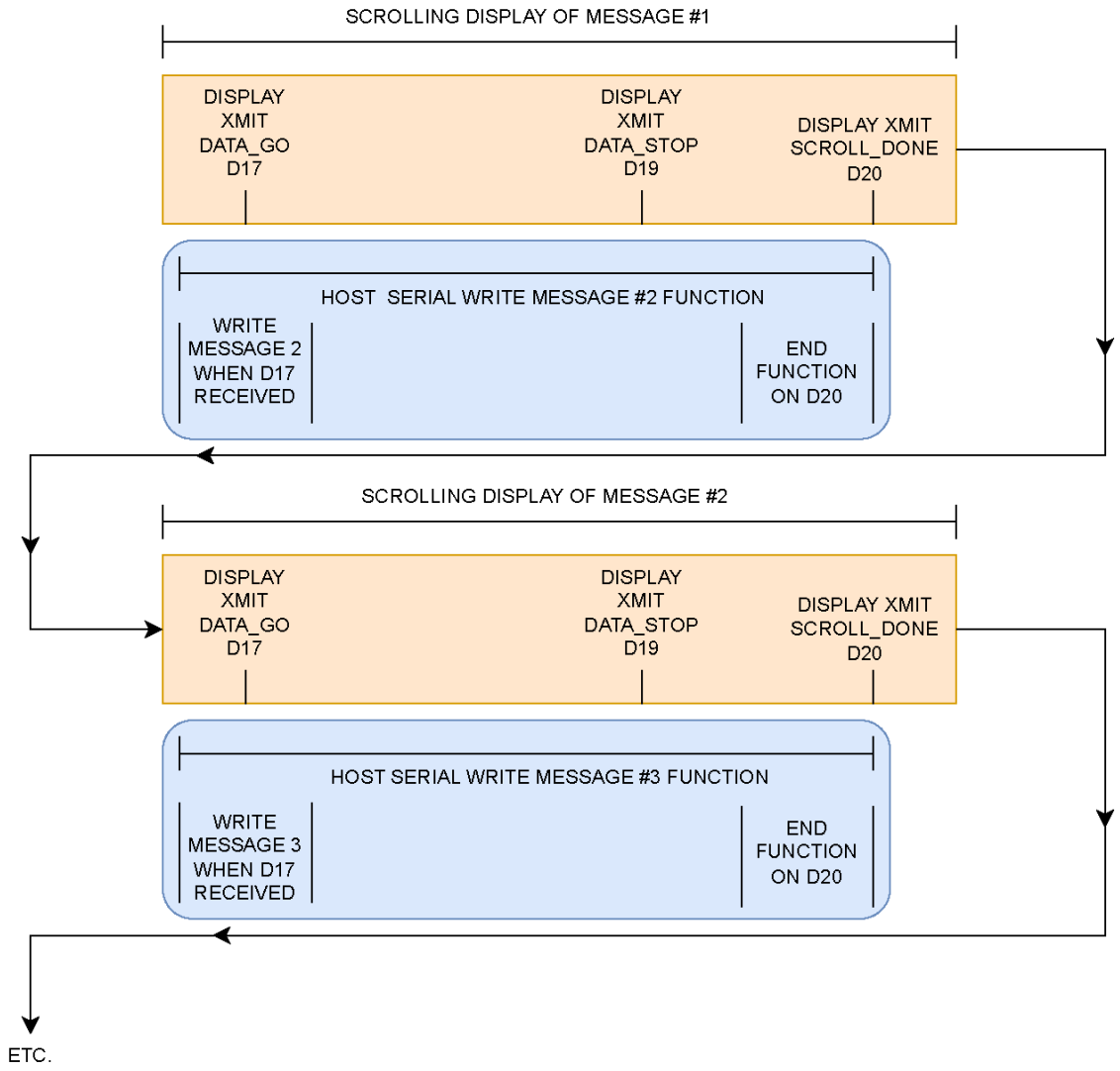
a) The display will repeat the same message if it has not received a new one by the time the scroll has finished.
b) Once the display has begun scrolling a message, new message data can be sent at any time. When the display finishes scrolling the message, the new message in the serial buffer will begin to be scrolled.

   One can deduce from the above behavior what is required to print new scrolling message one after another, reliably:

c) One must know when to send the new scrolling message data.
d) One must make sure new scrolling message data is not overwritten by more new scrolling message data.

Below is a flowchart illustrating how successive scroll writes are produced. When the display is first turned on in scroll mode, it immediately begins transmitting the Smarti™ codes DATA_GO, DATA STOP, and SCROLL_DONE on the TX line. This allows messages to be initially written. Let's assume the first message is being displayed already. When the display transmits a DATA_GO, the Host computer (Arduino or MCU) will then transmit the serial data to the display to write message #2. Host computer then waits until it receives a SCROLL_DONE code from the display unit, at which time the function call used for writing message #2 ends. At this time the function used for writing message #3 begins execution, and waits for a DATA_GO signal before it then transmits the serial data to the display to write message #3. Host computer then waits until it receives a SCROLL_DONE code from the display unit, at which time the function call used for writing message #3 ends, and so on.

It is the fact that function calls to write scrolling message do not end until a SCROLL_END flag is received, which keeps the program flow sequential and prevents data from mistakenly being overwritten.



Now let's look into how Smarti™ codes are used to change brightness and scroll speed at the individual message level. To understand how this works requires knowing an important characteristic of how the display handles scrolling messages. There is a blank character added at the end of each message. A SCROLL_END flag is generated during the time that this blank character is starting to be "displayed" on the leftmost position of the display. It is during this time interval when commands to change scroll speed and brightness can be given. In this way these commands can be executed in a manner that is concealed. Below is a scroll writing function that is used in the demonstration software. Note that at the end of the function, scroll speed control characters are sent and PWM brightness is modulated immediately after a SCROLL_DONE flag is received by the Host.

```
149
150   void scrollWrite ( const char* msg, uint8_t speed, int brite )
151   {
152   bool readflag1 = false;
153   bool readflag2 = false;
154   if (brite<0 || brite>255)
155   { brite=22;}// input validation. Set to 22 if entry out of bounds.
156
157     while (readflag1 == false)// Will continue reading
158     {
159       if (mySerial.available()>0)
160       {
161         byte controlByte= mySerial.read();
162         if (controlByte == 17)// Gate the function if received DATA_GO
163         {
164           readflag1 = true;// Breaks loop when recieved DATA_GO
165           mySerial.write(msg);
166           writeEOL(termMode);
167         }
168       }
169     }
170       while (readflag2 == false)// Will continue reading
171     {
172       if (mySerial.available()>0)
173       {
174         byte controlByte= mySerial.read();
175         if (controlByte == 20)// Gate the function if reeived SCROLL_DONE from the last message
176         {
177           readflag2 = true;//
178           mySerial.write(18);// Send Scroll Speed Change Prefix
179           mySerial.write(speed);// Send the speed data byte
180           writeEOL(termMode);// Send EOL terminator
181           delay(15);// Ensure processing time
182           analogWrite(pwm_Pin, brite);
183         }
184       }
185     }
186   }// Function terminates after receiving SCROLL_DONE flag
```

Note that message data can reliably be sent any time between DATA_GO and DATA_STOP time intervals, which varies according to message length. If, in the first loop, a DATA_STOP or SCROLL_DONE is read instead of a DATA_GO, then the message transmission will not occur until the *next* DATA_GO signal to be transmitted is received.

## 6.2 Smarti™ Codes Descriptive Summary

| Smarti™ Control Codes- Rx IN. These must be sent with EOL terminators. | | | |
|---|---|---|---|
| Decimal | ASCII Mnemonic | Opcode Name | Description |
| 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 |

| Smarti™ Control Codes- Tx OUT. These are sent without EOL terminators. | | | |
|---|---|---|---|
| Decimal | ASCII Mnemonic | Opcode Name | Description |
| 17 | DC1 | DATA_GO | Message data send GO signal |
| 19 | DC3 | DATA_STOP | Message data send STOP signal |
| 20 | DC4 | SCROLL_DONE | Message scrolling done |

Note that all of the Smarti™ codes that are transmitted by host to the display on the display Rx input must be sent with the EOL terminators. The type of EOL terminator chosen for the host must also be programmed into the display operating configuration so they can be properly processed by the display. The display can accommodate 4 different types of EOL terminators that can be chosen for the host to use.

SCROLL_SPD: When this is sent the next character that must be written is the scroll speed specifier, which is a number from 15 to 114, which must then be followed by the appropriate EOL terminator. Out-of-range values are ignored by the firmware.

RESET: When issued, it will cause the board to undergo a reset sequence, in an identical way as if the reset button were pushed, or if the reset pin on the I/O connector were used. Serial buffer will be flushed, and the screen will be reset to blank.

SCROLL_ON: Configures the display to operate in Scrolling display mode. If the display is already operating in scroll, mode, this command will have no effect. When sent, the display will clear and will begin sending out DATA_GO, DATA_STOP and SCROLL_DONE flags. Functions which send the SCROLL_ON command should not terminate until a SCROLL_END flag is received, in order to maintain proper program sequential execution.

SCROLL_OFF: Configures the display to operate in Non-Scrolling mode. If the display is already operating in non-scrolling mode, this command will have no effect. When sent, the display will begin sending out DATA_GO, DATA_STOP and SCROLL_DONE flags. Functions which send the

SCROLL_OFF command should send it after a DATA_GO flag is received. Functions which send a SCROLL_OFF command should not terminate until a SCROLL_END flag is received, in order to maintain proper program sequential execution.

EXTPWM_ON: Configures the display to accept PWM signal at the I/O PWM IN pin to modulate display brightness. Will supersede any brightness mode settings the display may be configured to run on based on the EEPROM settings. Functions which send a EXTPWM_ON command when the display is in scrolling mode should send this command when a DATA_GO flag is received, and the function should not terminate until a SCROLL_END flag is received, in order to maintain proper program sequential execution.

EXTPWM_OFF: Configures the display to run any brightness mode settings the display may be configured to run on based on the EEPROM settings. Users should be sure to leave the display PWM input pin at logic high after sending this command in order to permit the display to run in whatever brightness mode is set in accordance with programmed EEPROM settings. Functions which send a EXTPWM_OFF command when the display is in scrolling mode should send this command when a DATA_GO flag is received, and the function should not terminate until a SCROLL_END flag is received, in order to maintain proper program sequential execution.

DATA_GO: Transmission of message data to the display can commence after this flag is transmitted. It is sent as an ASCII Decimal 17 with no EOL terminator.

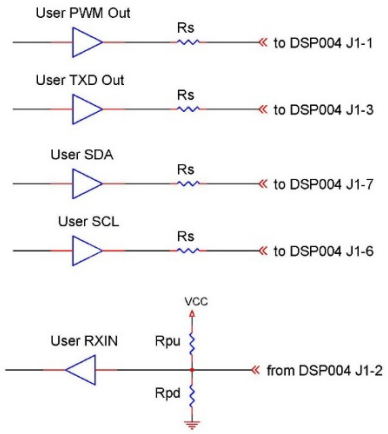
DATA_STOP: Transmission of message data to the display should not be sent after this flag is transmitted. It is sent as an ASCII Decimal 19 with no EOL terminator.

SCROLL_DONE: Transmission of this flag indicates the scrolling message is complete. It is sent as soon as the appended last-character blank space is being displayed on the leftmost display position. Once received, the user may then issue PWM and scroll speed instructions, which will take effect at some time right before the next message is to be displayed. The minimum time interval available to the user to send PWM or scroll speed commands after receiving a SCROLL_DONE flag is defined as one interval of character display time at the fastest scroll speed, which is 138 milliseconds.

Demonstration Arduino software using all of the Smarti™ codes is available for download on the Pronto Displays website. See the appendix for a detail showing Arduino connections for development and running the demonstration software.

## 7.0 EMI/RFI Considerations

Emissions can be minimized with a source termination resistor placed near the user's signal outputs, and with a parallel termination scheme placed at the users RXIN input pin as shown below.




As mentioned earlier in section 1.2, a resistor of nominal 1k should be placed in series with the PWM line near the user point of signal origin, to reduce unwanted EMI from fast PWM edges, ensuring proper operation of the display.

The values for Rs, Rpu and Rpd depend on the type of transmission medium and connectors used. Rs typically can have a value of 50 to 220 ohms.

Rpu and Rpd should be chosen such that their parallel resistance is equal to the impedance of the RXIN transmission line. Additionally, Rpd should be chosen such that (VOH/Rpd) is less than 25mA. For example, with a transmission line of impedance of 123 ohms, values could be Rpu=180 Ohms, Rpd= 390 Ohms. Output current with VOH=5.0V would be (5V/390 Ohms) = 13mA.

## 8.0 Troubleshooting Table

| **Symptom** | **Probable Cause** | **Corrective Action** |
|---|---|---|
| 1. Display is blank and unresponsive | a) Menu button has been pushed | 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 garbled | a) Incorrect Baud rate<br>b) Incorrect EOL Terminator<br>c) Out-of-range ASCII number being transmitted<br>d) Missing series resistor on PWM output | a) Verify unit is operating with correct Baud rate setting<br>b) Verify unit is operating with correct EOL Terminator setting<br>c) Verify only printable ASCII numbers 32-126 are being transmitted. For custom characters, verify ASCII numbers are within range of 128-161 and that the glyphs used have been properly programmed.<br>d) Add a 1k resistor in series with the PWM line, close to user signal origin point. |
| 4. Characters not displayed near end of message in scrolling mode | a) Too many characters have been transmitted<br>b) Wrong EOL terminator | a) Unit accepts messages of maximum 40 characters in length.<br>b) Check unit settings to ensure proper EOL terminator selection. |
| 5. Display appears only 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. |
| 6. Display flashes and/or pulsates | a) PWM signal applied with unit not configured to operate in EXT PWM mode<br>b) PWM signal frequency is too low | a) Ensure CONFIG switch is in PRGM position; also ensure PRGM mode operation is set to EXT PWM.<br>b) PWM frequency should be 6-9kHz. |
| 7. OVP trips | a) Overvoltage applied to VIN pin<br>b) Unit operating in a hot environment | a) Verify correct power input voltage: 4.75V - 5.3V<br>b) Remove unit to a cooler location <40 deg. C., or lower the power input voltage |
| 8. Menu does not appear | a) CONFIG switch in PRGM position<br>b) Incorrect Baud rate or EOL terminator | a) Set CONFIG switch to NORM position<br>b) Verify serial emulator is properly configured |

| | | |
|---|---|---|
| 9. Display exhibits inconsistent behavior | a) Intermittent connections<br>b) Power supply has excessive noise | a) Check for worn, loose, or broken connectors on +5V, GND, Rx and Tx pins.<br>b) Check +5VDC for high switching transient spikes; change power supply if required. |
| 10. Display pulsates when testing brightness with light source | a) Testing Light source is pulsewidth modulated, creating stroboscopic interference | a) Switch to a non-dimmable LED or an incandescent light source for testing brightness function |

For technical support, contact: support@prontodisplays.com

Note: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

-Reorient or relocate the receiving antenna.

-Increase the separation between the equipment and receiver.

-Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.

-Consult the dealer or an experienced radio/TV technician for help.

**CAN ICES-3(B) / NMB-3(B)**

# Appendix A

## A.1 Specification (Header) File for MC24AA024 Class Object

```
1
2   /*
3   Specification File for 24AA024
4   Copyright 2025 Pronto Electronic Displays LLC
5
6   I2C Address Writing Info Note: RW bit is not included in the "data word": the Wire library
    supplies that bit during compilation.
7   Note address must consist of control bits 1010 with A2 A1 A0 suffix added. Below table shows
    what to use for the address, depending on how A2,A1 and A0 are wired either high or low.
8
9
10              A2   A1   A0          0b        Hex
11              0    0    0    0b1010000      0x50
12              0    0    1    0b1010001      0x51
13              0    1    0    0b1010010      0x52
14              0    1    1    0b1010011      0x53
15              1    0    0    0b1010100      0x54
16              1    0    1    0b1010101      0x55
17              1    1    0    0b1010110      0x56
18              1    1    1    0b1010111      0x57
19
20  */
21
22  #ifndef M24AA024_h
23  #define M24AA024_h
24
25  class MC24AA024
26  {
27       private:
28       int device_addr;
29       uint8_t data_addr;
30       uint8_t databyte;
31
32       public:
33
34       MC24AA024(uint8_t addrl)//read address into constructor
35       {
36            device_addr = addrl;
37       }
38
39       void writeDataByte(uint8_t,uint8_t);// (address,data)
40
41       uint8_t readDataByte(uint8_t);//  // return readbyte from (address)
42
43  };
44
45  #endif
46
47
```

## A.2 Implementation File (.cpp) for MC24AA024 Class Object using Arduino

```
1
2   /*
3   Implementation File (.cpp) for MC24AA024
4   Copyright 2025 Pronto Electronic Displays LLC
5   */
6
7
8   #include <Wire.h>
9   #include<MC24AA024.h>
10  #include<arduino.h>
11
12                  void MC24AA024 :: writeDataByte(uint8_t addr,uint8_t data)
13  {
14        data_addr = addr;
15        databyte = data;
16        Wire.beginTransmission(device_addr);
17        Wire.write(data_addr);
18        Wire.write(databyte);
19        Wire.endTransmission();
20        delay(5);
21  }
22
23  uint8_t MC24AA024 :: readDataByte( uint8_t addr)
24  {
25        data_addr = addr;
26        uint8_t readbyte=0;
27        int i = 0;
28        Wire.beginTransmission(device_addr);
29        Wire.write(data_addr);
30        Wire.endTransmission(false);
31        Wire.requestFrom(device_addr,1);
32
33        if(Wire.available())
34              {
35                          readbyte = Wire.read();// Read the data in the buffer
36              }
37
38        return readbyte;
39  }
40
```

## A.3 Program code using MC24AA024 Class Object

The following Arduino program code may be used for testing read/write to MC24AA024 using the MC24AA024 Class Object.

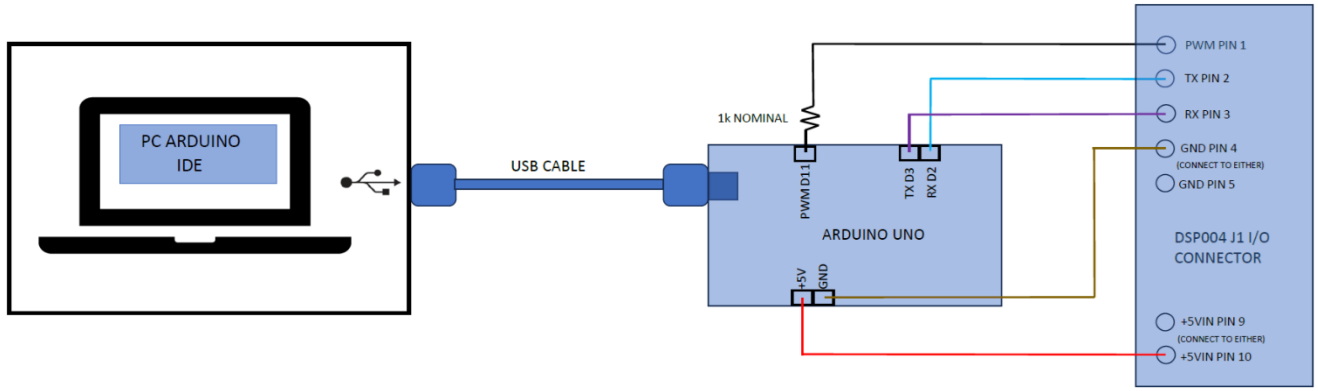
```
1
2  /* MC24AA024 Test Program
3  Copyright 2025 Pronto Electronic Displays LLC
4  This program checks EEPROM functionality by allowing writes/reads to specific
   Addresses.
5  Serial port is used to verify read/write operation.
6  Uses default pins for SDA,SCL.
7  */
8
9  #include <Wire.h>
10 #include <MC24AA024.h>
11
12 // Instantiate MC24AA024 Class Object. Pass the address into the constructor:
13
14 MC24AA024 U1(0x57);// Address used depends on how A0,A1,A2 pins are wired
15
16 void setup()
17 {
18
19 Serial.begin(9600);
20 Wire.begin();
21 Wire.setClock(100000);
22
23 }
24
25 void loop()
26 {
27
28 uint8_t data_address = 163;
29 uint8_t written_data = 47;
30 U1.writeDataByte(data_address,written_data);// (address, data)
31 delay(5);
32 uint8_t read_data;
33 read_data = U1.readDataByte(data_address);
34 Serial.print("Address: ");
35 Serial.println(data_address);
36 Serial.print("Written data: ");
37 Serial.println(written_data);
38 Serial.print ("Read data: ");
39 Serial.println(read_data);
40 Serial.println();
41 delay (5000);
42
43 }
44
45
```

## A.4 Program Code for PWM signal generation using Arduino

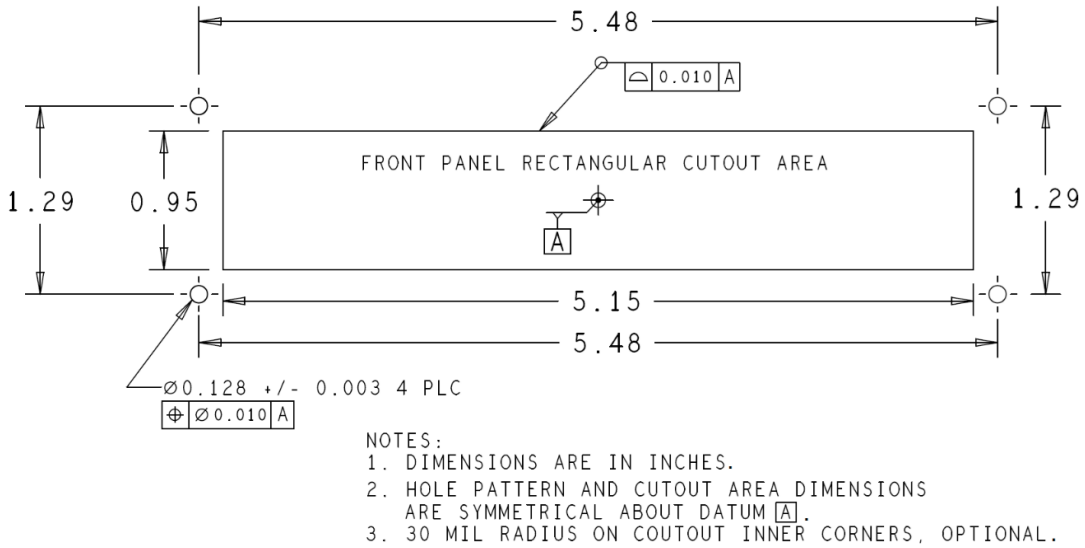
The following code may be used to generate PWM signal to the DSP004 unit using the Arduino Uno. Code is only for use with ATMEGA328 series microcontrollers.

```
/*
Copyright 2025 Pronto Electronic Displays LLC
PWM Program for dimming the DSP004 Display
This program will dim and brighten display in repeated cycles
*/

//************* BEGIN SETUP SECTION ***************

void setup()
{
  // initialize pins

  pinMode(11,OUTPUT);// Pin D11 used for PWM Dimming

// Set PWM parameters, 8kHz
 TCCR2B=0b00000010; // divide by 8 Prescaler
 TCCR2A=0b00000011; // Fast PWM mode for Timer 2

}
//############### END SETUP###############

void loop()
{

        // Begin PWM Dimming Loop

        for (int i = 255; i>0; i--)
          {
            analogWrite(11,i);
            delay(20);
          }
        for (int i = 0; i<256; i++)
          {
            analogWrite(11,i);
            delay(20);
          }

}
```

## A.5 Connecting Arduino for Development and Demo Software




## A.6 Recommended Front Panel Mounting/Cutout Dimensions and Mounting Hardware




Recommended Mounting Hardware:

a) Qty (4) pieces #4-40 x ¼ in. hex standoffs
b) Qty (4) pieces #4-40 x ¾ in. pan head machine screws
c) Qty (4) pieces #4 flat washer
d) Qty (4) pieces #4 split lock washer
e) Qty (4) pieces # 4-40 hex nut

| Document Revisions | | | |
|---|---|---|---|
| Rev. No. | Date | Engr. | Description |
| 01 | 8/21/2025 | M.Hawkins | Document origination, firmware Rev. 5.4.7 |
| 02 | 12/18/2025 | M.Hawkins | Updated for firmware Rev. 5.7.4 |
| | | | |