Grayhill INC.

An ISO-9001 Company

Grayhill Micro Display Configuration Tool

Software Developer’s Manual

3DUM1302-1 Rev D

Revision History

Revision	Date	Description
1		Preliminary
2	06/01/10	Added backlight intensity section; Updated screen shots
3	07/23/10	Added header file generation section; Updated dynamic text formatting information
4	08/19/10	Updated object parameter tables; Revised Appendix sample screens
A	08/20/10	Initial Release
B	05/02/11	Add note on DLL installation
C	06/08/11	Added troubleshooting section and memory size/object count section.
D	07/26/11	Updates for PC Tool version 2.00 and Firmware version 4.0

Table of Contents

1. Introduction

The Software Developer's Manual provides instruction on how to configure a Grayhill MicroDisplay device using the Grayhill MicroDisplay PC Tool software.

The MicroDisplay programming paradigm involves using a PC Tool to 1) create a group of objects, 2) save the objects as a Project, and 3) download the project to the MicroDisplay unit over the CAN bus. Objects programmed into the MicroDisplay can be activated or deactivated by an ECU or by other objects.

Starting with firmware version 4.0 and PC Configuration Tool version 2.0, there is support for limited standalone operation of the MicroDisplay. The soft keys can be used to change screens and menu objects to allow further navigation into screens, all without the intervention of an ECU. See www.grayhill.com/offhwyfiles.aspx to download new firmware and new PC Configuration Tool software.

For a description of MicroDisplay objects see Appendix A.

2. Programming Interface

The Grayhill MicroDisplay Configuration Tool interfaces to a Grayhill MicroDisplay device through a CAN bus interface. The tool currently supports using a GridConnect® USB CAN adapter. Please visit www.gridconnect.com for details regarding the USB CAN adapter.

In order for the Configuration Tool to communicate with the MicroDisplay device, the Dynamic Link Library (PCAN_USB.DLL) provided with the USB CAN adapter needs to be installed in a specific location. The DLL can either be installed in the same directory as the configuration tool executable, or in the following OS-appropriate directory:

Operating System	**Directory**
Microsoft Windows XP	\Windows\system32
Microsoft Windows 7 (64-bit)	\Windows\SysWOW64

Make sure the 32-bit version is used as opposed to the 64-bit version of the DLL.

3. Application Startup

When the application is started, the screen will look like the following:

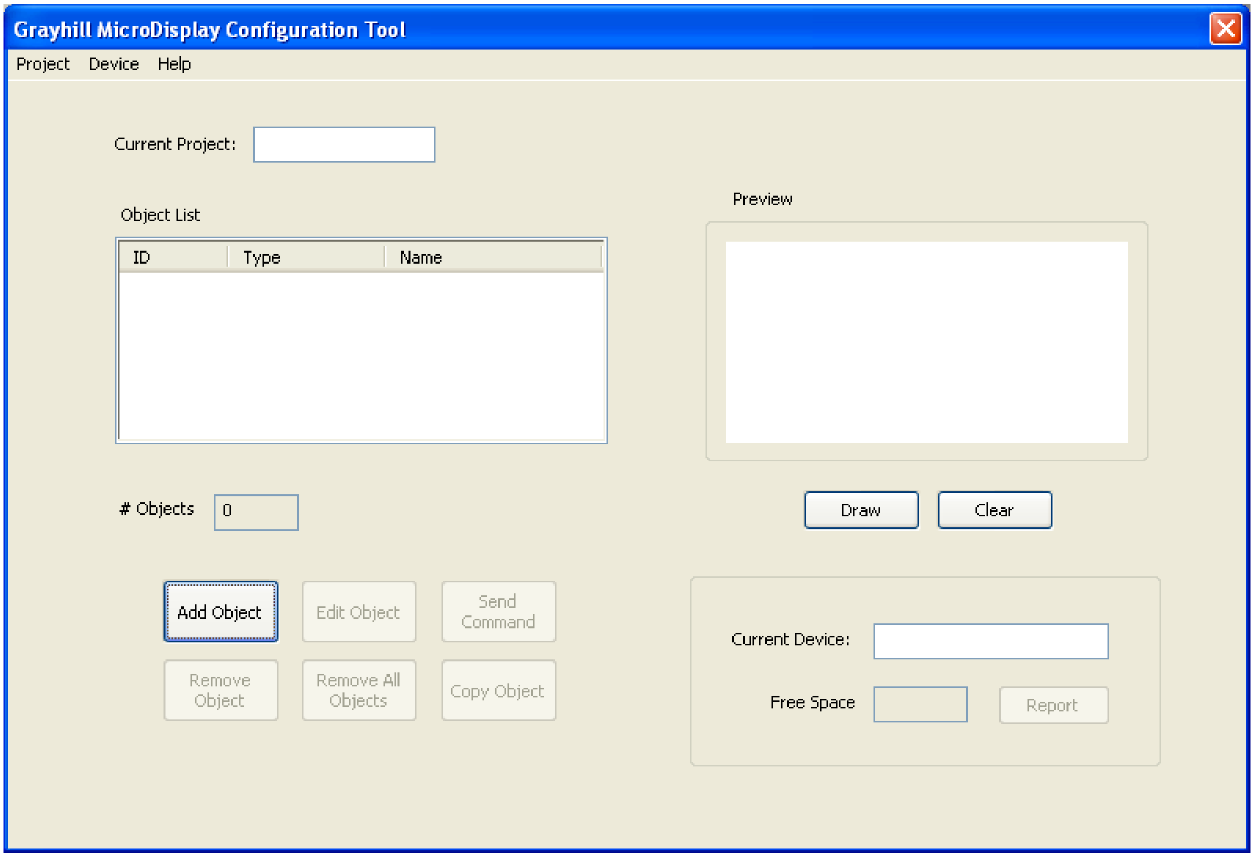


Figure 3-1

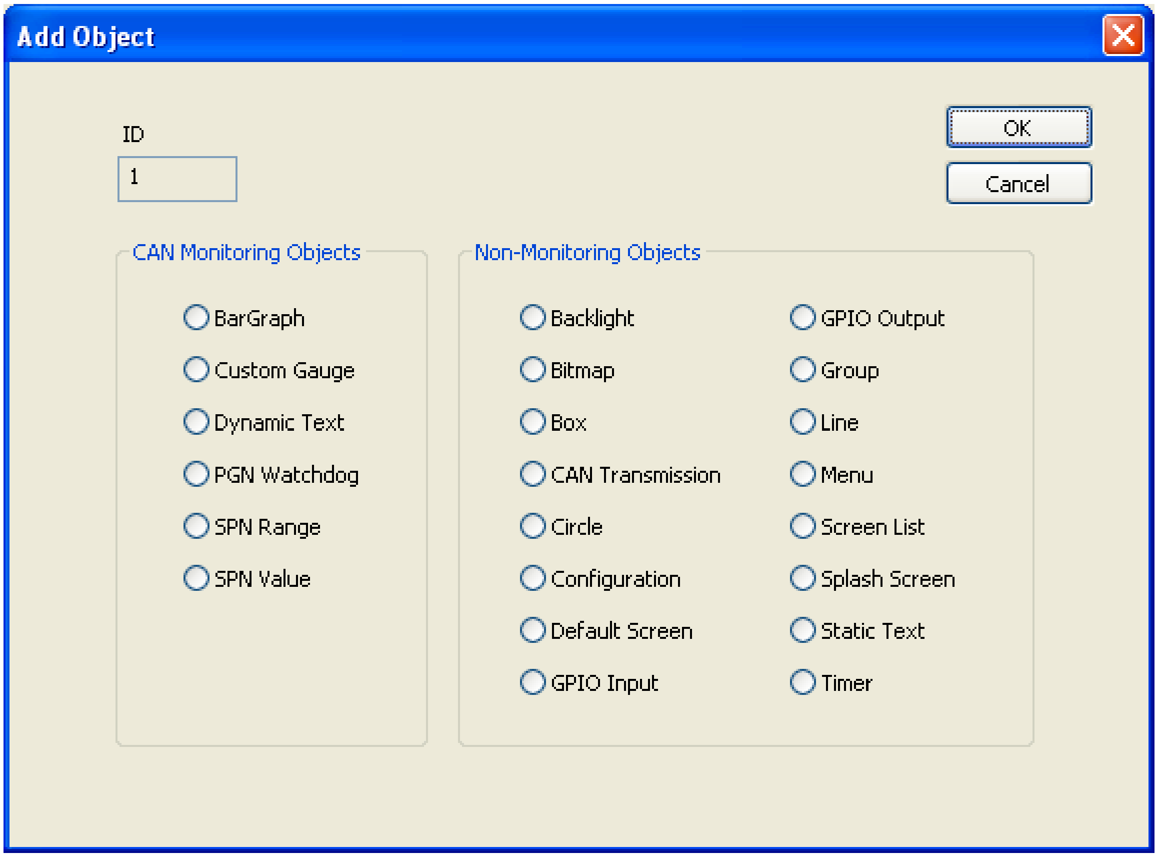
There are 3 options for getting started:

1. Create a new project
2. Open an existing project
3. Upload existing objects from a device

Options 1 and 2 can be performed independent of an actual MicroDisplay being connected. Option 3 requires communication with a MicroDisplay.

4. Creating a New Project

To create a new project, start by clicking on the ‘Add Object’ button. The *Add Object* dialog box will appear. This dialog allows you to choose from a number of different object types:



Add Object

ID

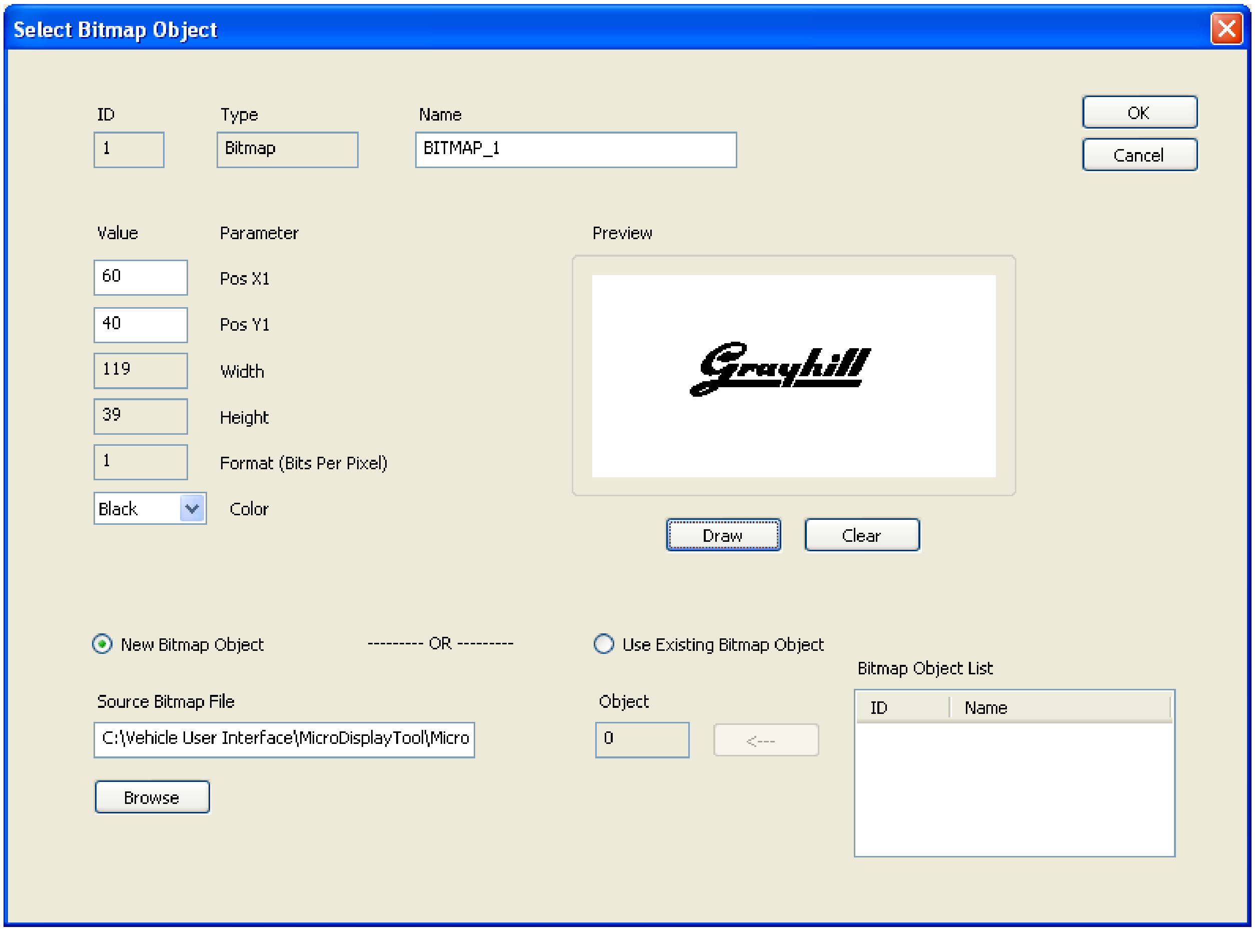
1

Figure 4-1

An object ID number is automatically assigned for the new object. Select the radio button of the desired object type and click ‘OK’.

A new dialog box will then be displayed that will allow you to construct the type of object you have selected by entering values for several parameters. The details for the various object parameters will be discussed in a subsequent section. Once the parameters have been entered, you can for some objects preview what the object will look like by clicking the ‘Draw’ button. The values entered will first be validated, then the object will be drawn in the preview window. If any of the entries are incomplete or invalid, a message will be displayed. To clear the preview window, click the ‘Clear’ button. Once you are satisfied with the object, click ‘OK’ to accept the changes and close the dialog box. Or, click ‘Cancel’ to discard the changes and close the dialog box.

As an example, suppose the new object is a Bitmap. The following is an example of what the dialog would look like:



Select Bitmap Object

ID: 1
Type: Bitmap
Name: BITMAP_1

Value	Parameter
60	Pos X1
40	Pos Y1
119	Width
39	Height
1	Format (Bits Per Pixel)
Black	Color

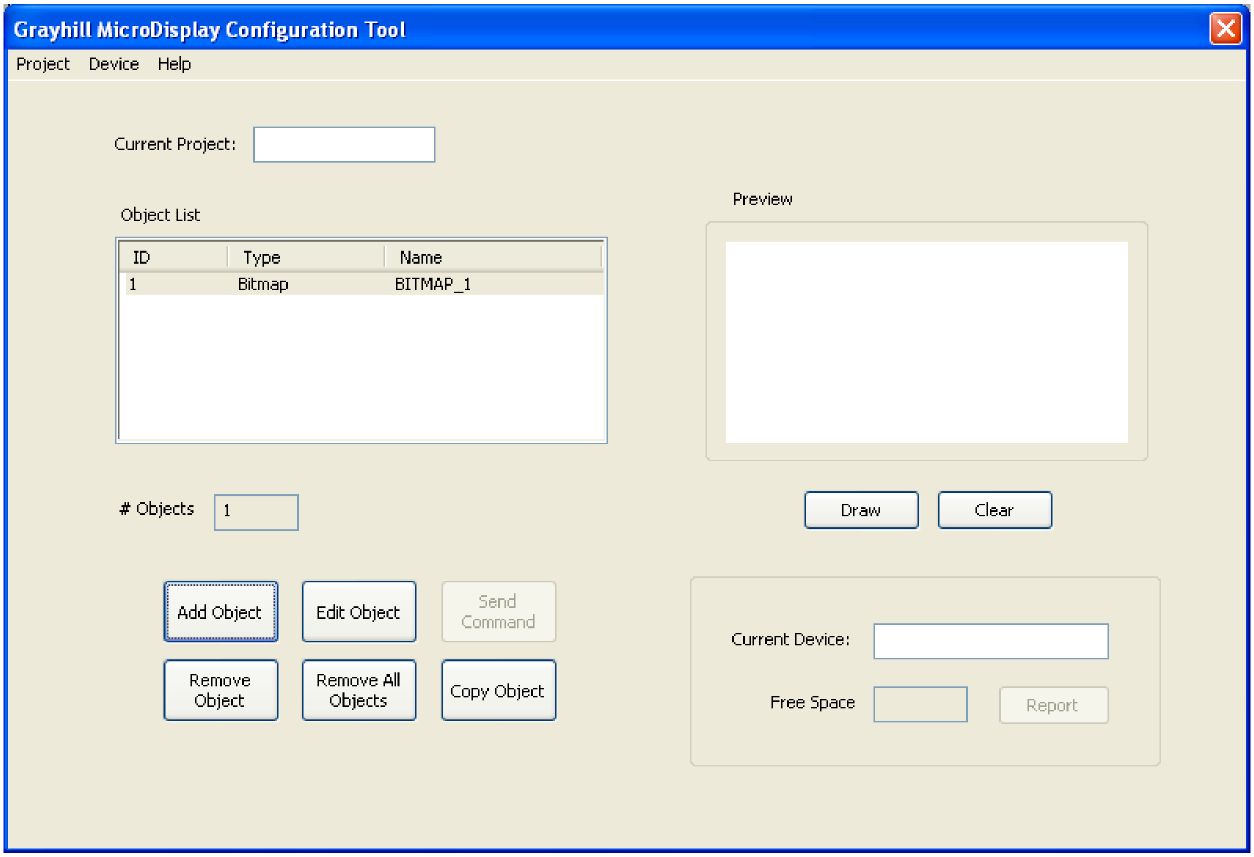
● New Bitmap Object --------- OR --------- ○ Use Existing Bitmap Object

Source Bitmap File: C:\Vehicle User Interface\MicroDisplayTool\Micro

Object: 0

Figure 4-2

In the main window, the newly created object is added to the object list and the object count is updated.



ID	Type	Name
1	Bitmap	BITMAP_1

Objects: 1

Figure 4-3

To save the newly created object in a Project:

1. In the main menu, click ‘Project’
2. Click ‘Save’ or ‘Save As’
3. In the *Save As* dialog box, enter a project filename and choose a directory in which to save the project file
4. Click ‘Save’

The project name will be displayed in the *Current Project* box.

5. Opening an Existing Project

To open an existing project:

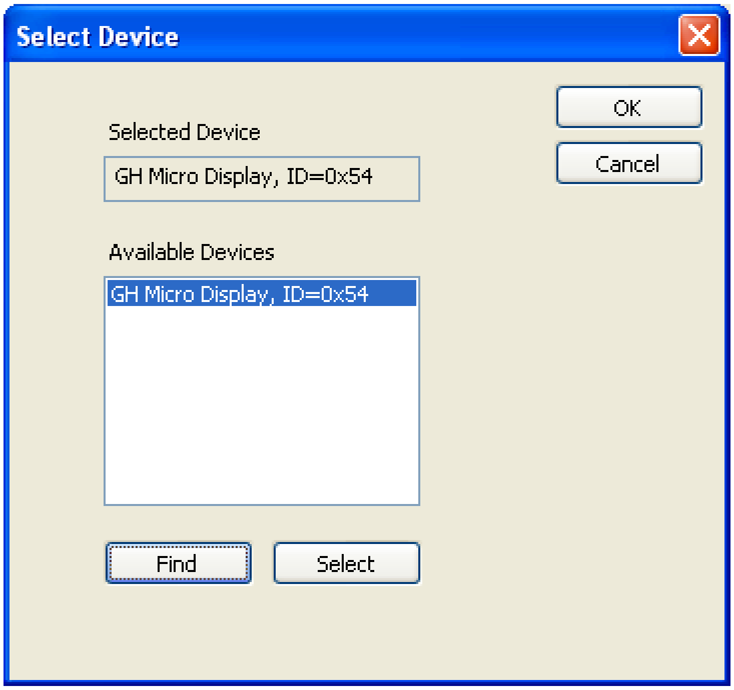
1. In the main menu, click ‘Project’
2. Click ‘Open’
3. Navigate to the desired directory location and select a project file. Project files are saved with extension “.MDP” (<u>M</u>icro<u>D</u>isplay <u>P</u>roject).
4. Click ‘Open’

Objects and their associated data will be read from the project file and added to the object list. The existing objects can then be modified or removed. New objects can also be added to the project.

6. Communicating with a MicroDisplay

Before objects can be uploaded from a MicroDisplay to the configuration tool, communication must be established with the device. Perform the following steps to establish communication with a MicroDisplay:

1. In the main menu, Click ‘Device’. See the Troubleshooting section if this is disabled (grayed out).
2. Click ‘Select’
   The *Select Device* dialog box will be displayed
3. Click the ‘Find’ button
   The application will send a message over the CAN bus soliciting a response from any available devices.
4. If more than 1 MicroDisplay device is found, select the desired unit from the list and click ‘Select’. If a single MicroDisplay device is found, it will automatically be selected
5. Click ‘OK’ to accept the selection and close the dialog box.



Selected Device: GH Micro Display, ID=0x54

Available Devices: GH Micro Display, ID=0x54

Figure 6-1

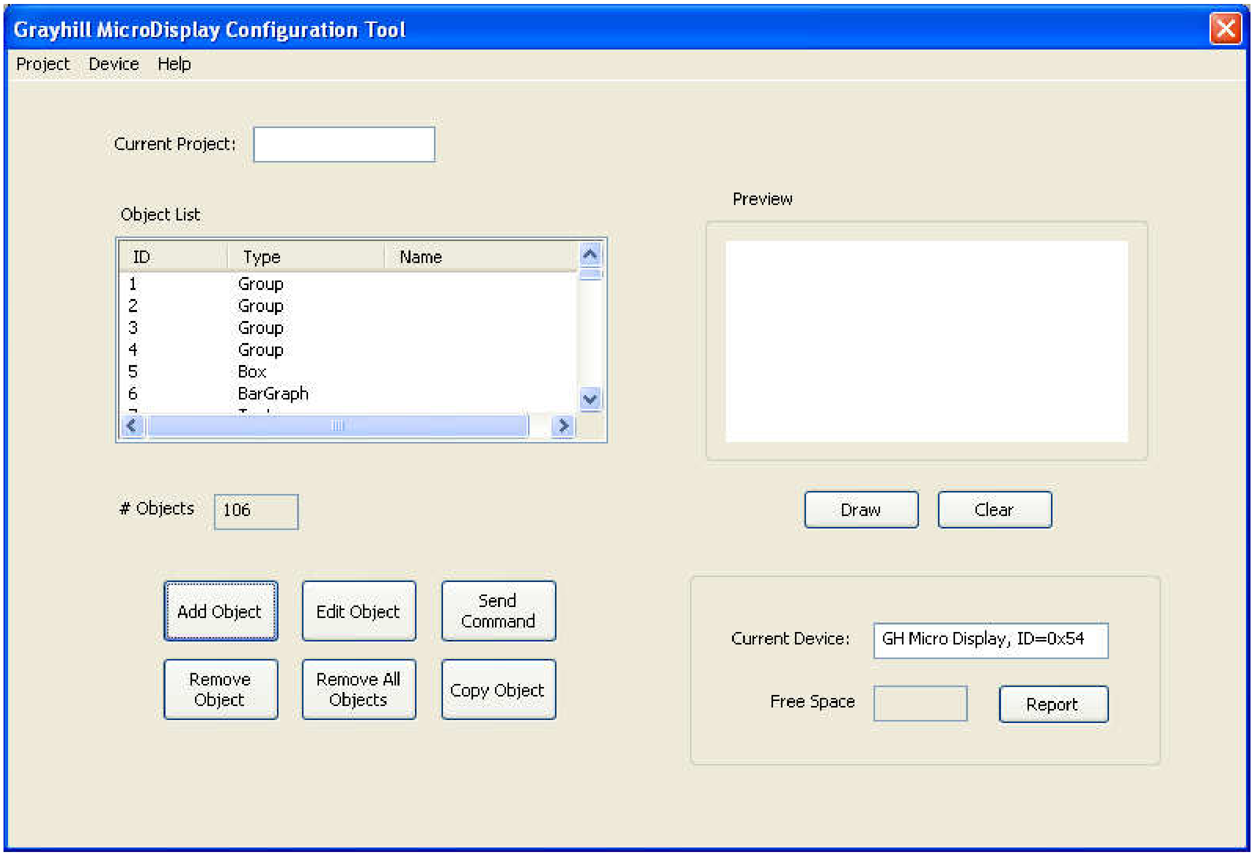
You will then return to the main window. The selected MicroDisplay device will now be displayed in the “Current Device” box.

7. Uploading Objects from a MicroDisplay

Object data can be uploaded to the Configuration Tool from the MicroDisplay, then saved as a project. This is accomplished by performing the following steps:

1. Establish communication with a MicroDisplay device as discussed in the previous section
2. From the main menu, select ‘Device’
3. Select ‘Commands’
4. Click ‘Upload All Objects From MicroDisplay’

The object data will be transferred from the MicroDisplay to the configuration tool. Each object will be added to the object list. However, note that no names are displayed for the objects in the list. This is because the object names are NOT saved in the MicroDisplay – they are used only in the configuration tool. If you are planning to save the uploaded object information as a project, you may wish to add a name for each object. See section 8, *Modifying an Object*, for more information.



Current Project:

Object List

ID	Type	Name
1	Group	
2	Group	
3	Group	
4	Group	
5	Box	
6	BarGraph	

Objects 106

Current Device: GH Micro Display, ID=0x54

Free Space

Figure 7-1

You may save the object information as a project by clicking ‘Project’ in the main menu, followed by ‘Save As’. In the *Save As* dialog box, enter a project filename and directory location, then click ‘Save’. The project name will be displayed in the *Current Project* box.

8. Modifying an Object

To modify an object, select it in the object list and click the ‘Edit Object’ button. A dialog box will be displayed that will allow you to edit the values of the parameters for the object. The object ID and Type parameters cannot be modified. The object Name can always be modified.

SPN Lookup

Screens for object types that use CAN message data contain an “SPN lookup” feature to assist with the object definition. These object screens provide a listbox containing standard J1939 SPN values and descriptions. When an SPN is selected in the list, the values for the associated PGN, Bit Start, and Bit Length parameters are automatically copied to their respective edit control boxes.

9. Removing an Object from a Project

Objects can be removed from a project individually, or collectively as a project.

To remove a single object:

1. Select the object in the object list
2. Click the ‘Remove Object’ button

The object will be removed from the list.

To remove all objects in a project:

1. Click the ‘Remove All Objects’ button.

The object list will be cleared.

Note that the objects are not actually removed from the project file unless you save the project by clicking ‘Project’ then ‘Save’ in the main menu.

10. Copying an Object

To copy an object:

1. Select the object in the object list
2. Click the ‘Copy Object’ button

An exact copy of the object is made and added to the object list, assigned with the first available ID number.

11. Downloading Objects to a MicroDisplay

To download all objects within a project to the selected MicroDisplay device:

1. From the main menu, select ‘Device’
2. Select ‘Commands’
3. Select ‘Download All Objects To MicroDisplay’

The successfully downloaded objects are now saved in Flash memory on the MicroDisplay.

NOTE: Before performing a download to the MicroDisplay, all objects on the device must first be erased using the ‘Erase All Objects On MicroDisplay’ command. See section 15*, Erasing Objects Stored on the MicroDisplay*, for more details.

12. Displaying an Object on a MicroDisplay

Both volatile and non-volatile objects can be displayed on the MicroDisplay.

To display a non-volatile object that is stored in Flash memory on the selected MicroDisplay device:

1. Select an object in the object list
2. Click the ‘Send Command’ button
3. The *Send Command* dialog will appear
4. In the ‘Command’ combo box, select “Display Object”
5. Click the ‘Send’ button
6. The Response box will display the result of the transaction (ACK or NAK)

The object should be visible on the MicroDisplay screen.

A few volatile object types can be displayed on the MicroDisplay without having to be stored in Flash memory. These types include:

- Circle
- Box
- Line
- Static Text

To display a volatile object on the selected MicroDisplay device:

1. Select one of the volatile object types in the object list
2. Click the ‘Send Command’ button
3. The *Send Command* dialog will appear
4. In the ‘Command’ combo box, select the appropriate “Draw” command (“Draw Circle”, “Draw Box”, “Draw Line”, “Draw Text”)
5. Click the ‘Send’ button
6. The Response box will display the result of the transaction (ACK or NAK)

The object should be visible on the MicroDisplay screen.

13. Removing an Object from the MicroDisplay Screen

To remove an object that is displayed on the screen of the selected MicroDisplay device:

1. Select an object in the object list
2. Click the ‘Send Command’ button
3. The *Send Command* dialog will appear
4. In the ‘Command’ combo box, select “Kill Object”
5. Click the ‘Send’ button
6. The Response box will display the result of the transaction (ACK or NAK)

The object should no longer be visible on the MicroDisplay screen. Note that the object is NOT erased from the Flash memory on the MicroDisplay.

14. Clearing the MicroDisplay screen

To clear the screen on the selected MicroDisplay device:

1. From the main menu, select ‘Device’
2. Select ‘Commands’
3. Click ‘Clear MicroDisplay Screen’

The screen on the MicroDisplay should be blank.

15. Erasing Objects Stored on the MicroDisplay

To erase the objects stored on the selected MicroDisplay device:

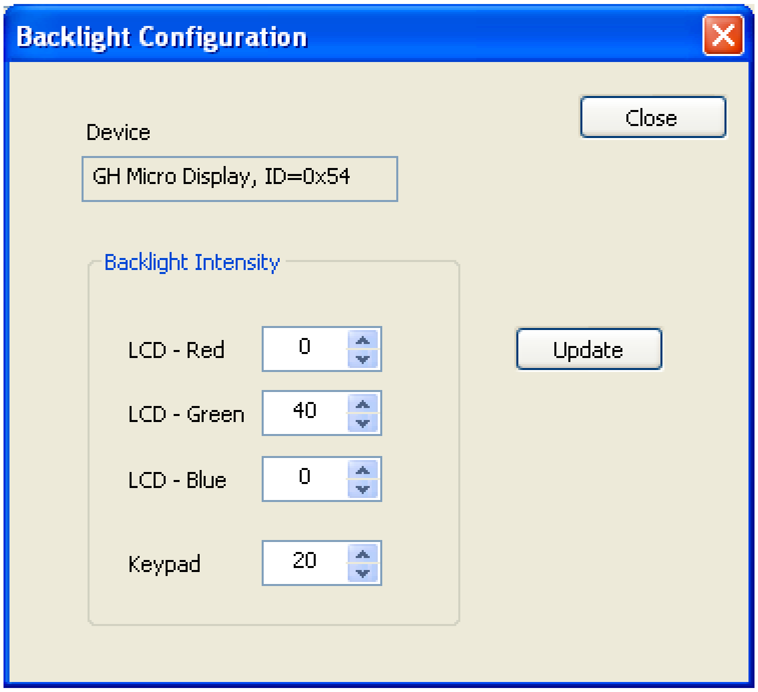
1. From the main menu, select ‘Device’
2. Select ‘Commands’
3. Click ‘Erase All Objects On MicroDisplay’
4. A message box will be displayed asking to confirm that you wish to delete all objects from the device Flash storage - click ‘Yes’ to perform the erase, or ‘No’ to abort

16. Setting the Backlight Intensity on the MicroDisplay

The backlight intensity can be configured on the selected MicroDisplay device.

1. From the main menu, select ‘Device’
2. Select ‘Configuration’
3. Click ‘Backlight’

   The *Backlight Configuration* dialog box will be displayed:



Device

GH Micro Display, ID=0x54

Backlight Intensity

LCD - Red: 0

LCD - Green: 40

LCD - Blue: 0

Keypad: 20

Figure 16-1

The intensity of the LCD red, green, and blue backlights is configurable, as well as the keypad backlight. Values from 0 – 177 can be entered, with 177 being maximum brightness.

4. Specify the desired value for each using the spin control or by typing in a number
5. Click ‘Update’ to send the new configuration to the MicroDisplay

17. Generating a ‘C’ style header file

Once a project has been defined, the objects it contains can be output in a ‘C’ language header file.

1. From the main menu, select ‘Project’
2. Select ‘Generate Header File’

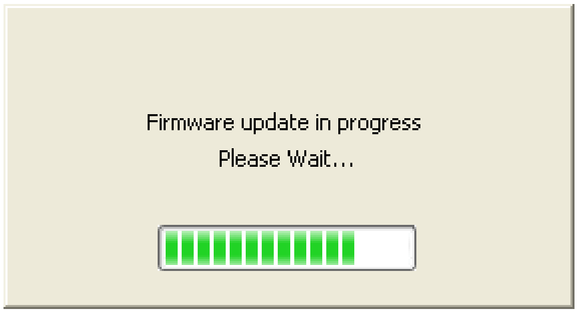
The header file produced will be the same name as the project, but with a .h extension.

18. Performing a Firmware Upgrade

The MicroDisplay firmware can be updated with the configuration tool.

1. From the main menu, select ‘Device’
2. Select ‘Configuration’
3. Click ‘Firmware Upgrade’

A dialog box will be displayed for selecting a firmware .hex file to download to the MicroDisplay. Select the desired hex file, then click ‘Open’. A message box will be displayed warning not to disconnect power from the device during the firmware update procedure. Click ‘OK’ to start the firmware upgrade. The MicroDisplay is placed in bootloader mode and the backlight is turned off. During the update, the configuration tool displays a message box with a progress bar:



Firmware update in progress
Please Wait...

Figure 18-1

When the update process is complete, the MicroDisplay should be running normally and the backlight turned back on.

Appendix A – Micro Display Overview

The intended use of the MicroDisplay is to display J1939 SPN parameter information in real time. It can monitor SPN's that report an analog value, such as coolant temperature. It can monitor SPN's that report operating states, such as the differential lock state being either engaged or disengaged. To do this the MicroDisplay is configured with CAN Monitoring Objects. These objects monitor the J1939 bus and parse the relevant data using the following criteria:

- **PGN** – The PGN the SPN parameter to be displayed belongs to.
- **SA** – The source address of the sending device
- **Bit Start** – The bit location of the least significant bit of the SPN parameter within the J1939 message.
- **Bit Length** – The number of bits the parameter occupies within the J1939 message.
- **Control Bit Start** - The bit location of the control parameter in the event that multiple parameters are multiplexed within the same bit field defined with Bit Start and Bit Length.
- **Control Bit Length** – The number of bits the control parameter occupies.
- **Control Data Value** – The actual value the control field must equal in order to act on the parameter defined with Bit Start and Bit Length.

Multiple Monitoring Objects can be active at the same time, giving the MicroDisplay the ability to replace several existing standard gauges or an entire gauge cluster.

A MicroDisplay screen is composed of one or more objects. An object can be either a Monitoring Object, as described above, or a Non-Monitoring Object, which includes bitmaps, lines, boxes, text, etc. Some Monitoring Objects control the display of other objects. For example, the SPN Range Object displays one of three possible objects depending on if the monitored value falls within a specified range. It will display the Below Range, In Range, or Above Range object if the value is below, within, or above the range respectively.

If it is desired to instantiate several objects at one time the Group Object is used. When instantiated or killed, all objects within the group are then instantiated or killed respectively. Referring to the example above, the SPN Range Object could be assigned the ID of a Group Object containing text, bitmaps, shapes, etc., all of which would be instantiated or killed according to the monitored parameter value.

The Menu Object gives more power to the MicroDisplay by allowing the end user to select which object to display from a list. An object in this scenario would most likely be a group object that could display Monitoring Objects along with text and bitmaps forming a complete screen related to a specific function like engine status, for example.

Each object contains the following parameters that are common to all of the available object types:

Parameter	**Description**
ID #	Object ID number assigned by configuration tool Range: 1 – 65535 ***- Read Only -***
Type	Object Type ***- Read Only -***
Name	ASCII text, length up to 32 characters. By default, for a new object the tool will automatically assign a generic name based on object type and ID #. For example, a new bitmap with object id 3 would be

	assigned the name BITMAP_3. The name can be changed at any time. The name is NOT used by the Micro Display itself; it is only used within the configuration tool for the benefit of the user.

Each object also contains various parameters that are specific to that object type. Objects are categorized as being CAN Monitoring or Non-Monitoring and are described in the following sections.

A.1 Can Monitoring Objects

A.1.1 Bar Graph Object

The Bar Graph Object monitors CAN traffic for an SPN value and adjusts the bar graph value accordingly. The position is specified by the x and y coordinates of the upper left corner of the bar. The bar length (at 100%) and bar width are specified in pixels. The bar graph data can be scaled by specifying the minimum and maximum data values, corresponding to bar graph values of 0% and 100% respectively. One of four possible colors can be specified for the bar, as well as a horizontal or vertical orientation.

Parameter	Description
PGN	Parameter Group Number Range: 0x0 – 0x1FFFF
Source Address	Source address of sending device Range: 0x0 – 0xFF
Bit Start	Starting bit position of the data to use within the associated CAN message data Range: 1 – 64
Bit Length	Bit width of the data to use within the associated CAN message data Range: 1 – 32
Control Data Bit Start	For multiplexed PGN's this is the bit position of the control field that determines the parameter at Bit Start. Set to zero (-0-) if no control field exists. Range: 1 – 64
Control Data Bit Length	Size of the control bit field. Set to zero (-0-) if no control field exists. Range: 1 – 32
Control Data Value	Value the bit field defined by the Control Bit Start and Control Bit Length must equal in order for the object to update itself with the data found at Bit Start
Min Value	Value corresponding to bar graph value of 0%
Max Value	Value corresponding to bar graph value of 100%
Position X1	X-coordinate of upper left corner of bar Range: 0 – 255
Position Y1	Y-coordinate of upper left corner of bar Range: 0 – 127
Length	Length (in pixels) of bar at 100% Range: 1 – 256 (horizontal) or 1 – 128 (vertical)
Width	Width (in pixels) of bar Range: 1 – 128 (horizontal) or 1 – 256 (vertical)
Orientation	Horizontal / Vertical
Color	Black / Dark Gray / Light Gray / White

A.1.2 Dynamic Text Object

The Dynamic Text Object monitors CAN traffic for an SPN value and displays the results as a text string on the MicroDisplay LCD. The text position is specified by the x and y coordinates of the upper left corner of the first character. One of four font sizes can be specified as well as one of four possible colors. The text string is created using the 'C' programming language *printf* style (see Note 1 below).

Parameter	Description
PGN	Parameter Group Number Range: 0x0 – 0x1FFFF
Source Address	Source address of sending device Range: 0x0 – 0xFF
Bit Start	Starting bit position of the data Range: 1 – 64
Bit Length	Bit width of the data Range: 1 – 32
Control Data Bit Start	For multiplexed PGN's this is the bit position of the control field that determines the parameter at Bit Start. Set to zero (-0-) if no control field exists. Range: 1 – 64
Control Data Bit Length	Size of the control bit field. Set to zero (-0-) if no control field exists. Range: 1 – 32
Control Data Value	Value the bit field defined by the Control Bit Start and Control Bit Length must equal in order for the object to update itself with the data found at Bit Start
Resolution	Resolution per bit in units of the SPN
Offset	Offset in units of the SPN
Position X1	X-coordinate for start of first character Range: 0 – 255
Position Y1	Y-coordinate for start of first character Range: 0 – 127
Color	Black / Dark Gray / Light Gray / White
Font Size	6x8 / 8x12 / 12x16 / 24x32
String Format	'C' *printf* style using the %f format specifier. See Note 1 below.

Note 1: Using the 'C' printf style %f format specifier and modifiers

Minimum Field Width modifier - The minimum field width can be specified by placing a decimal number between the % (percent) sign and the f. This ensures a minimum length for the output value by padding with spaces. To pad with 0's (zeroes) instead of spaces, add a 0 before the field width number. An important point to keep in mind is that the output will always be displayed with 6 decimal places if not otherwise specified with the precision modifier (discussed below), which limits the use of the minimum field width modifier.

Examples:

Value to display = 75

Format String	Output
`Speed = %f mph`	`Speed = 75.000000 mph`

`Speed = %10f mph`	`Speed =  75.000000 mph`
`Speed = %010f mph`	`Speed = 075.000000 mph`

Precision modifier - The number of decimal places displayed in the output can be specified with the precision modifier, which is a period followed by a number. The precision modifier is placed between the % sign and the f, immediately following the minimum field width modifier (if used).

Examples:

Value to display = 3.14159265

Format String	Output
`Pi = %.5f`	`Pi = 3.14159`
`Pi = %8.5f`	`Pi =  3.14159`
`Pi = %08.5f`	`Pi = 03.14159`

Justifying output – By default, the output is right-justified in the field. To left-justify the output in the field, place a minus (-) sign immediately after the % sign.

Examples:

Value to display = 98.6

Format String	Output
`Temp = %6.1f deg`	`Temp =   98.6 deg`
`Temp = %-6.1f deg`	`Temp = 98.6   deg`

A.1.3 SPN Value Object

The SPN Value Object monitors CAN traffic and maps an exact SPN value to a specific object. This is intended for SPN’s where the measured data is a state rather than an analog value.

Parameter	Description
PGN	Parameter Group Number Range: 0x0 – 0x1FFFF
Source Address	Source address of sending device Range: 0x0 – 0xFF
Bit Start	Starting bit position of the data Range: 1 – 64
Bit Length	Bit width of the data Range: 1 – 32
Control Data Bit Start	For multiplexed PGN’s this is the bit position of the control field that determines the parameter at Bit Start. Set to zero (-0-) if no control field exists. Range: 1 – 64
Control Data Bit Length	Size of the control bit field. Set to zero (-0-) if no control field exists. Range: 1 – 32
Control Data Value	Value the bit field defined by the Control Bit Start and Control Bit Length must equal in order for the object to update itself with the data found at Bit Start

SPN Value Object List	List size: 32
Bit Field Value	Value to compare against the measured value
Bound Object ID	Object that is displayed if the Bit Field Value matches the measured value exactly

A.1.4 SPN Range Object

The SPN Range Object is used to monitor CAN traffic and display an assigned object based on whether the measured value falls within, above, or below a specified range. The range is defined by the specified Min and Max value parameters.

Parameter	**Description**
PGN	Parameter Group Number Range: 0x0 – 0x1FFFF
Source Address	Source address of sending device Range: 0x0 – 0xFF
Bit Start	Starting bit position of the data Range: 1 – 64
Bit Length	Bit width of the data Range: 1 – 32
Min Value	Minimum value of the bit field Range: 0 – 4,294,967,295
Max Value	Maximum value of the bit field Range: 0 – 4,294,967,295
Control Data Bit Start	For multiplexed PGN's this is the bit position of the control field that determines the parameter at Bit Start. Set to zero (-0-) if no control field exists. Range: 1 – 64
Control Data Bit Length	Size of the control bit field. Set to zero (-0-) if no control field exists. Range: 1 – 32
Control Data Value	Value the bit field defined by the Control Bit Start and Control Bit Length must equal in order for the object to update itself with the data found at Bit Start
Below Range Object ID	Object that is displayed if the bit field value is less than the Min Value
In Range Object ID	Object that is displayed if the bit field value is between the Min Value and Max Value (inclusive)
Above Range Object ID	Object that is displayed if the bit field value is greater than the Max Value

A.1.5 Custom Gauge Object

The Custom Gauge Object provides the ability to create custom gauges that display objects based on a value. This object uses an array of boundary segments composed of a compare value, an active object, and an inactive object. Either the active or inactive object is displayed depending on the comparison between the segment's compare value and the measured value. The segment's compare values MUST be defined in ascending order.

Parameter	Description
PGN	Parameter Group Number containing the SPN parameter to monitor Range: 0x0 – 0x1FFFF
Source Address	Source address of sending device Range: 0x0 – 0xFF
Bit Start	Starting bit position of the data Range: 1 – 64
Bit Length	Bit width of the data Range: 1 – 32
Control Data Bit Start *(see Appendix A)*	For multiplexed PGN's this is the bit position of the control field that determines the parameter at Bit Start. Set to zero (-0-) if no control field exists. Range: 1 – 64
Control Data Bit Length	Size of the control bit field. Set to zero (-0-) if no control field exists. Range: 1 – 32
Control Data Value	Value the bit field defined by the Control Bit Start and Control Bit Length must equal in order for the object to update itself with the data found at Bit Start
Gauge Type	Standard: Multiple segments can be active behaving like a bargraph. Pointer: At most 1 segment is active behaving like a gauge pointer.
Boundary Segment Map	Map size: 50
Boundary Value	Value to compare with the measured value
Object ID – Active	ID of object that is displayed if (1) the boundary value is less than or equal to the measured value for the standard option or if (2) the measured value is between two adjacent segment compare values for the pointer option
Object ID –Inactive	ID of object that is displayed if the Active object is not displayed

A.1.6 PGN Watchdog Object

The PGN Watchdog Object is used to monitor CAN traffic for a specific PGN and source address. If the expected message is received within the specified time period, the Default Object is displayed. If the message is not received within the specified Timeout Period, the Timeout Object is displayed. If the message is received after the timeout has expired, the Timeout Object is removed and the Normal Object re-displayed.

Parameter	Description

PGN	Parameter Group Number containing the SPN parameter to monitor Range: 0x0 – 0x1FFFF
Source Address	Source address of sending device Range: 0x0 – 0xFF
Timeout Period	Maximum amount of time without receiving the expected CAN message before displaying the Timeout Object Units: milliseconds Range: 0 – 10,000
Normal Object	ID of object to display prior to timeout expiration and when expected message is received
Timeout Object	ID of object to display if specified message is NOT received within Timeout Period

A.2 Non-Monitoring Objects

A.2.1 Circle Object

The Circle Object displays a circle on the MicroDisplay LCD. The circle position and size is specified by the center coordinate and radius parameters, respectively. One of four possible colors can be specified, as well as the line thickness (in pixels), and whether the circle is filled or hollow.

Parameter	Description
Position X0	X-coordinate of center Range: 0 – 255
Position Y0	Y-coordinate of center Range: 0 – 127
Radius	Radius of circle Range: 1 – 255
Color	Black / Dark Gray / Light Gray / White
Fill	True / False
Width	Thickness (in pixels) of circle perimeter Range: 1 – 32

A.2.2 Box Object

The Box Object displays a box on the MicroDisplay LCD. The box position and size is specified by the upper left and lower right coordinates, respectively. One of four possible colors can be specified, as well as the line thickness (in pixels) and whether the box is filled or hollow.

Parameter	Description
Position X1	X-coordinate of upper left corner Range: 0 – 255
Position Y1	Y-coordinate of upper left corner Range: 0 – 127
Position X2	X-coordinate of lower right corner Range: 0 – 255
Position Y2	Y-coordinate of lower right corner Range: 0 – 127
Color	Black / Dark Gray / Light Gray / White
Fill	True / False
Width	Thickness (in pixels) of box sides Range: 1 – 32

A.2.3 Line Object

The Line Object displays a line on the MicroDisplay LCD. The line position and size is specified by the x and y coordinates of the two endpoints. One of four possible colors can be specified, as well as the line thickness (in pixels).

Parameter	**Description**
Position X1	X-coordinate of endpoint 1 Range: 0 – 255
Position Y1	Y-coordinate of endpoint 1 Range: 0 – 127
Position X2	X-coordinate of endpoint 2 Range: 0 – 255
Position Y2	Y-coordinate of endpoint 2 Range: 0 – 127
Color	Black / Dark Gray / Light Gray / White
Width	Thickness (in pixels) of line Range: 1 – 32

A.2.4 Bitmap Object

The Bitmap Object displays a bitmap on the MicroDisplay LCD. A bitmap file is selected and the bitmap data is imported from the file. Bitmaps must be 1, 2, 4, or 8 bits-per-pixel in order to be imported by the configuration tool. The MicroDisplay supports only 1 and 2 bpp formats, so 4 or 8 bpp bitmaps are automatically converted to 2 bpp by the configuration tool.

The position is specified by the x and y coordinates of the bitmap upper left corner. The width, height, and bpp format of the bitmap is determined by the imported bitmap file and is not modifiable. For monochrome (1 bpp) bitmaps, one of four possible colors can be specified.

Parameter	**Description**
Position X0	X-coordinate of upper left corner Range: 0 – 255
Position Y0	Y-coordinate of upper left corner Range: 0 – 127
Width	Width in pixels Range: 1 – 256 ***- Read Only -***
Height	Height in pixels Range: 1 – 128 ***- Read Only -***
Format	Bits-per-pixel (bpp) Bitmaps must be either 1,2,4,or 8 bpp to be imported by the configuration tool. Bitmaps of 4 or 8 bpp will automatically be converted to 2 bpp. ***- Read Only -***
Color	Black / Dark Gray / Light Gray / White

A.2.5 Static Text Object

The Static Text Object displays a string of text on the MicroDisplay LCD. The text position is specified by the x and y coordinates of the upper left corner of the first character. One of four font sizes can be specified, as well as one of four possible colors.

Parameter	Description
Position X1	X-coordinate of upper left corner of first character Range: 0 – 255
Position Y1	Y-coordinate of upper left corner of first character Range: 0 – 127
Font	6x8 / 8x12 / 12x16 / 24x32
Color	Black / Dark Gray / Light Gray / White

A.2.6 Group Object

The Group Object holds a collection of other objects as a group. This allows multiple objects to be displayed or removed simultaneously when the group object ID is referenced. A group object may also contain other group objects. The objects within a group object are displayed in order, which is important to keep in mind when objects overlap each other.

Parameter	Description
Object Count	Number of objects included in group
Object ID list	List of object ID numbers included in group

A.2.7 Splash Screen Object

The Splash Screen Object references an object to be displayed at boot up as a splash screen for the specified time duration. Once the time expires, the Splash Screen object is removed and the Default Screen object (if assigned) is displayed.

Parameter	Description
Link ID	ID of object to use as a splash screen
Duration	Length of time (in seconds) splash screen will be displayed Range: 0 – 60

A.2.8 Default Screen Object

The Default Object references an object that is to be displayed following boot up or reset, and after the Splash Screen object (if assigned) has timed out.

Parameter	**Description**
Link ID	ID of object being used as default screen

A.2.9 Timer Object

The Timer Object is used to periodically toggle between two objects. This can be useful to blink an object on screen.

Parameter	**Description**
Interval A	Units: milliseconds Range: 0 – 10000
Interval B	Units: milliseconds Range: 0 – 10000
Interval A Object ID	ID of object to display during Interval A
Interval B Object ID	ID of object to display during Interval B

A.2.10 CAN Transmission Object

The CAN Transmission Object is used to transmit a CAN message when instantiated. The full 29-bit CAN ID is specified, as well as the data length and data bytes (up to 8 maximum). The number of times to transmit the CAN message is specified, as well as the time between consecutive transmissions. A Tx OK Object can be assigned which will be displayed upon a successful transmission of the CAN message.

An option is also available for the object to monitor for a corresponding acknowledgement message.
The Ack message is specified by a PGN, source address, eight data, and eight mask bytes. The object is successfully acknowledged when a message is received that matches the PGN, source address, and non-masked off bits within the data bytes. Upon acknowledgement, the Tx OK Object is displayed and the CAN Object ceases to transmit the remaining count of the CAN transmit message. If the proper Ack message is not received after the send period of the last transmission, the Fault Object is displayed.

Parameter	**Description**
Send Count	Maximum number of times to transmit the CAN message Range: 1 – 65535
Send Period	Time between consecutive CAN messages Units: Milliseconds Range: 1 – 4,294,967,295

Option	Specify whether to monitor for acknowledgement message Ack / No Ack
CAN ID	29 bit identifier of transmitted CAN message Range: 0 – 0x1FFFFFFF
CAN DLC	Data Length of transmitted CAN message Range: 0 – 8
CAN Data	Data bytes of transmitted CAN message
Ack PGN	Parameter Group Number of Ack message Range: 0x0 – 0x1FFFF
Ack Source Address	Source address of Ack message
Ack DLC	Data Length of Ack message Range: 0 – 8
Ack Data	Data bytes of Ack message
Ack Data Mask	Mask values of Ack message to check against Ack data bytes
Fault Object	ID of object to display in the event Ack is expected and not received
Tx OK Object	ID of object to display upon successful CAN message transmission. Object is displayed once.

A.2.11 GPIO Input Object

The GPIO Input Object monitors the three MicroDisplay GPIO inputs and displays objects according to the state of the inputs. Objects can be assigned for the low and high states of all three inputs.

Parameter	**Description**
Input 1 Low Object	ID of object to display if input is low
Input 1 High Object	ID of object to display if input is high
Input 2 Low Object	ID of object to display if input is low
Input 2 High Object	ID of object to display if input is high
Input 3 Low Object	ID of object to display if input is low
Input 3 High Object	ID of object to display if input is high

A.2.12 GPIO Output Object

The GPIO Output Object provides independent control of the two MicroDisplay GPIO outputs. Outputs default to Low (0) on power up. Selecting ‘No Change’ for a value allows one output to be controlled without affecting the other.

Parameter	**Description**
Output 1	Value: Low, High, No Change
Output 2	Value: Low, High, No Change

A.2.13 Menu Object

The Menu Object is used to display a menu that can be navigated by the keypad on the MicroDisplay.

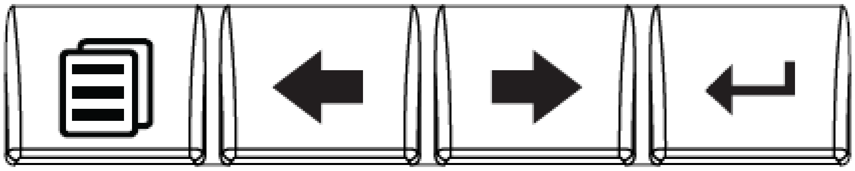


Figure A-1

The type of menu is specified with the Menu Option parameter. There are 2 main types of menu: Navigation and Button-Bound.

Navigation Menu – This is a standard list-oriented type of menu.

For the Navigation menu type, there are additional menu options that may be assigned. The options are as follows:

- **Root Menu –** When this option is selected, the menu item is defined as a Root Menu. Pressing the left most button will cause the menu object with this option set to be instantiated. Only one menu object with this option set should be implemented in a project. If more than one Root Menu is defined, the first one encountered in the list will launch.
- **Menu Loop –** When this option is selected, pressing the left/right buttons will loop through the Selected objects in the menu item array, displaying only the active selected object determined by the cursor position. Most likely the Selected object is a group object that populates the entire screen with other objects. If this option is not selected, all of the Unselected objects in the array are displayed forming a visible list of items to scroll through.
- **Persistent –** When this option is selected, the menu is inhibited from unloading. Select this option if the menu item is the root menu and it is desired that pressing the menu button should have no effect. This is useful when the root menu is in loop mode and the menu should never exit. The Selected object should be a group object that contains monitoring objects, graphics, or another menu object for entering a submenu.

Button-Bound Menu – For this menu type, each button is assigned an object. Most likely the object will be a group object populating the screen with various other objects.

Parameter	**Description**
Menu Option	Navigation (Root / Loop / Persistent) Button-Bound
Timeout Inhibit	Disable timeout feature of menu. When set to true, the menu will stay until a selection is made by pressing the Enter button or the Menu button is pressed. Otherwise, the complete menu system will unload itself and re-display the object from before the root menu was instantiated. The timeout period is approximately 20 seconds.

Background Object	An object to display when the menu is instantiated. This object should occupy the part of the display that doesn't change during the process of navigating through the menu.
Menu Item Array	Array size: 32
Object ID to Display	ID of object to be displayed when ENTER is pressed
Selected Object ID	ID of object that will be displayed when it is the menu item that has cursor focus. Only one item can have cursor focus at a time
Unselected Object ID	ID of object that will be displayed when it is not the menu item that has cursor focus.
Unselected Color	If the Unselected Object is not specified, the Selected Object is displayed with this parameter when the menu item is not selected. Possibilities are any one of the four colors, (white, light gray, dark gray, black)
Selected Color	If the Unselected Object is not specified, the Selected Object is displayed but with different attributes specified by this parameter. Possibilities are any one of the four colors (white, light gray, dark gray, black) or inverted.

A.2.14 Screen List Object

The Screen List Object is useful for scrolling through a list of frequently used screens. It allows the use of the left and right buttons on the display to scroll through a list of defined objects. Each object can be a group object that populates the entire screen with other objects.

Parameter	**Description**
Object Count	Number of objects included in list
Object ID list	List of object ID numbers included in list

A.2.15 Configuration Object

The Configuration Object allows some useful settings to be embedded into a project.

Parameter	**Description**
Button Message Enable	Enable transmission of the button message
Aux I/O Message Enable	Enable transmission of the AUXIO message
Menu Save	Display last Menu selection on Bootup. This will override the Default Screen object, if it exists.

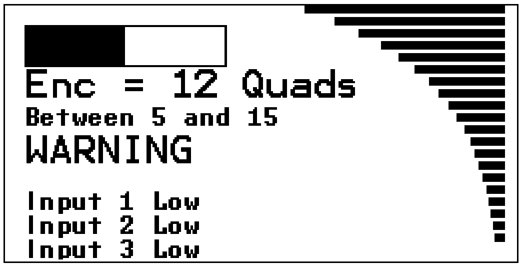
A.2.16 Backlight Object

The Backlight Object is used to control the backlight intensity of the LCD and keypad. When this object is instantiated the backlights are immediately modified. When this object is removed or killed the previously set backlighting values are restored.

Parameter	Description
Red	Red component of the LCD backlight
Green	Green component of the LCD backlight
Blue	Blue component of the LCD backlight
Keypad	Keypad backlight
Set as Default	TRUE/ FALSE If True, store the values as the default for the device

Appendix B – A Sample Screen

The following screenshot is from a project that monitors the messages from a Grayhill 3J series VDC with encoder along with monitoring the three discrete inputs. Refer to Figure A-1



Enc = 12 Quads

Between 5 and 15

WARNING

Input 1 Low

Input 2 Low

Input 3 Low

Figure B-1

The following objects are used in this example.

- Bar Graph
- PGN Watchdog
- Timer
- Static Text
- Dynamic Text
- SPN Range
- Custom Gauge
- GPIO Input

The Bar Graph simply displays a graphical representation of the data being monitored.

The “Enc = 12 Quads” is an example of a CAN Monitoring object that either instantiates another CAN Monitoring object or a Static Text object. The object at the top of the hierarchy is a PGN Watchdog object. This object either instantiates a Dynamic Text object that will show the encoder value or a Static Text object displaying “NO DATA” when no message is received after a specified amount of time. When the VDC is sending messages the Dynamic Text is updated accordingly. If the VDC stops transmitting the monitored messages then the Dynamic Text will be replaced with the text “No Data”. If communication resumes the “No Data” is replace with the Dynamic Text.

The Static Text “Between 5 and 15” is assigned to an SPN Range Object. The range is programmed from 5 to 15. If the encoder value is below, within, or above the range that portion of the screen will read “Below 5”, “Between 5 and 15”, and “Above 15” respectively.

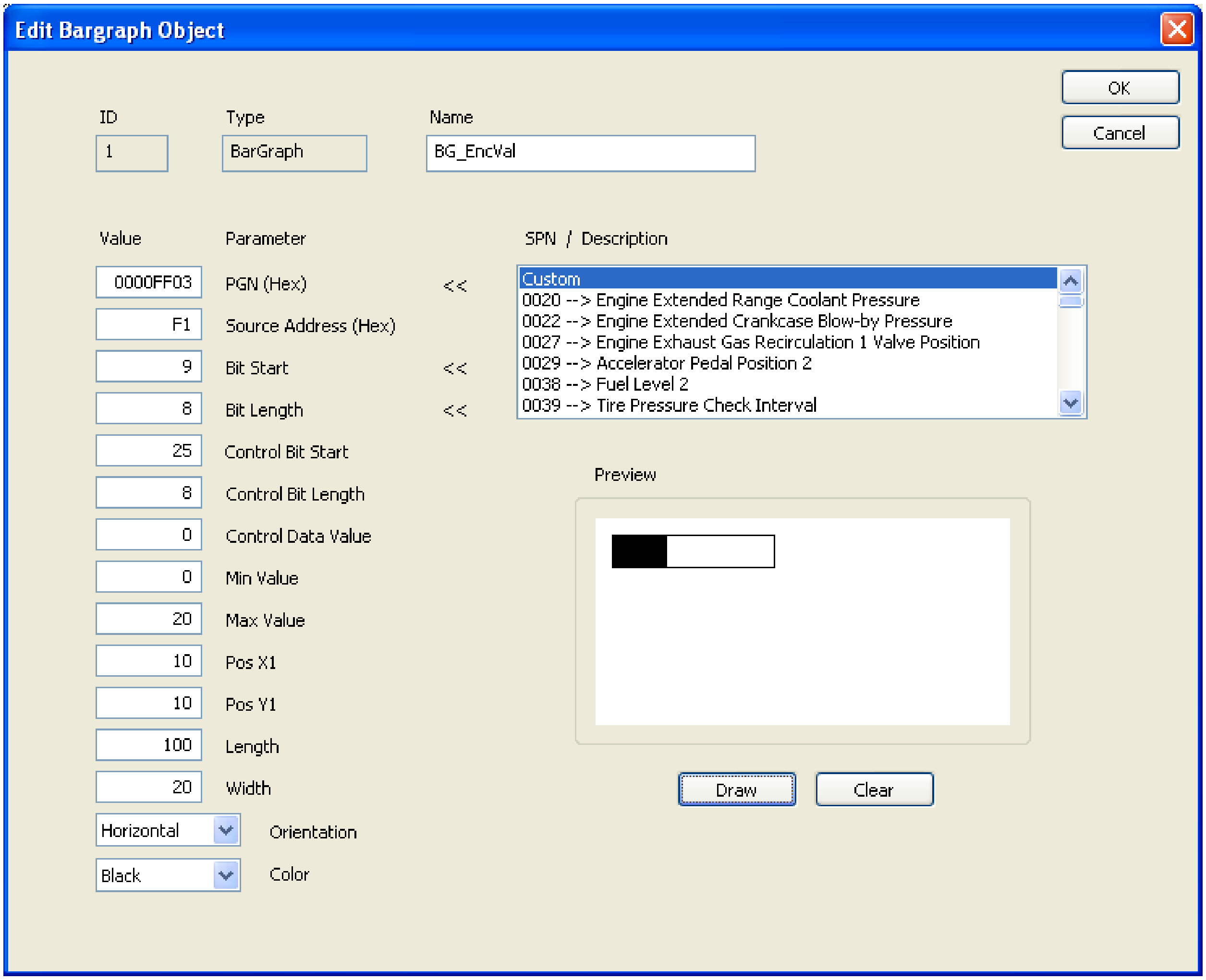
The “WARNING” is another implementation of the SPN Range Object. When the encoder value is above 20 the warning text is displayed and blinking. When below, that area is left blank. Blinking is realized with a Timer object that toggles between two objects. In this example the objects are the Static Text object “WARNING” and the null object.

The right side of the screen is an implementation of the Custom Gauge Object. The bars are either displayed or not according to the encoder value. One caveat is when the value is at the top or bottom of the scale, the top most or bottom most bars toggle between solid and hollow.

Finally, the bottom left corner uses the GPIO Input Object to monitor the three inputs. It simply displays either “Input x Low” or “Input x High”.

Bar Graph: The first object that will be created is the simple bar graph. It will monitor the encoder field within the message sent from the VDC. The following instructions explain how to create this object.

1. Click on “Add Object” then select “Bargraph” and click OK.
2. Fill in the parameters illustrated below and click OK



Edit Bargraph Object

ID: 1
Type: BarGraph
Name: BG_EncVal

Value	Parameter
0000FF03	PGN (Hex)
F1	Source Address (Hex)
9	Bit Start
8	Bit Length
25	Control Bit Start
8	Control Bit Length
0	Control Data Value
0	Min Value
20	Max Value
10	Pos X1
10	Pos Y1
100	Length
20	Width
Horizontal	Orientation
Black	Color

SPN / Description
Custom
0020 --> Engine Extended Range Coolant Pressure
0022 --> Engine Extended Crankcase Blow-by Pressure
0027 --> Engine Exhaust Gas Recirculation 1 Valve Position
0029 --> Accelerator Pedal Position 2
0038 --> Fuel Level 2
0039 --> Tire Pressure Check Interval

Figure B-2

This object parses the VDC CAN message having a PGN value of 0xFF03 from a source address of 0xF1. The portion of the CAN data used to create the bar value is defined by the Bit Start and Bit Length fields. In this example, the Bit Start is 9 and Bit Length is 8. Therefore, the 8 bits of the 2nd data byte comprise the value used for the bar shading. The Control Bit Start, Control Bit Length and the Control Data Value is used for parameters that are multiplexed within the same bit field. In this example, the fourth byte, which holds the button information of the VDC, must be zero in order for the bar graph to update. Turning the encoder with no buttons pressed will cause the bar graph to update. Pressing a button that causes this byte to be non-zero will inhibit the bar graph from updating.

CAN Message ID=XXFF03F1 DLC=8 Data={00 XX 00 YY F0 FF FF FF}

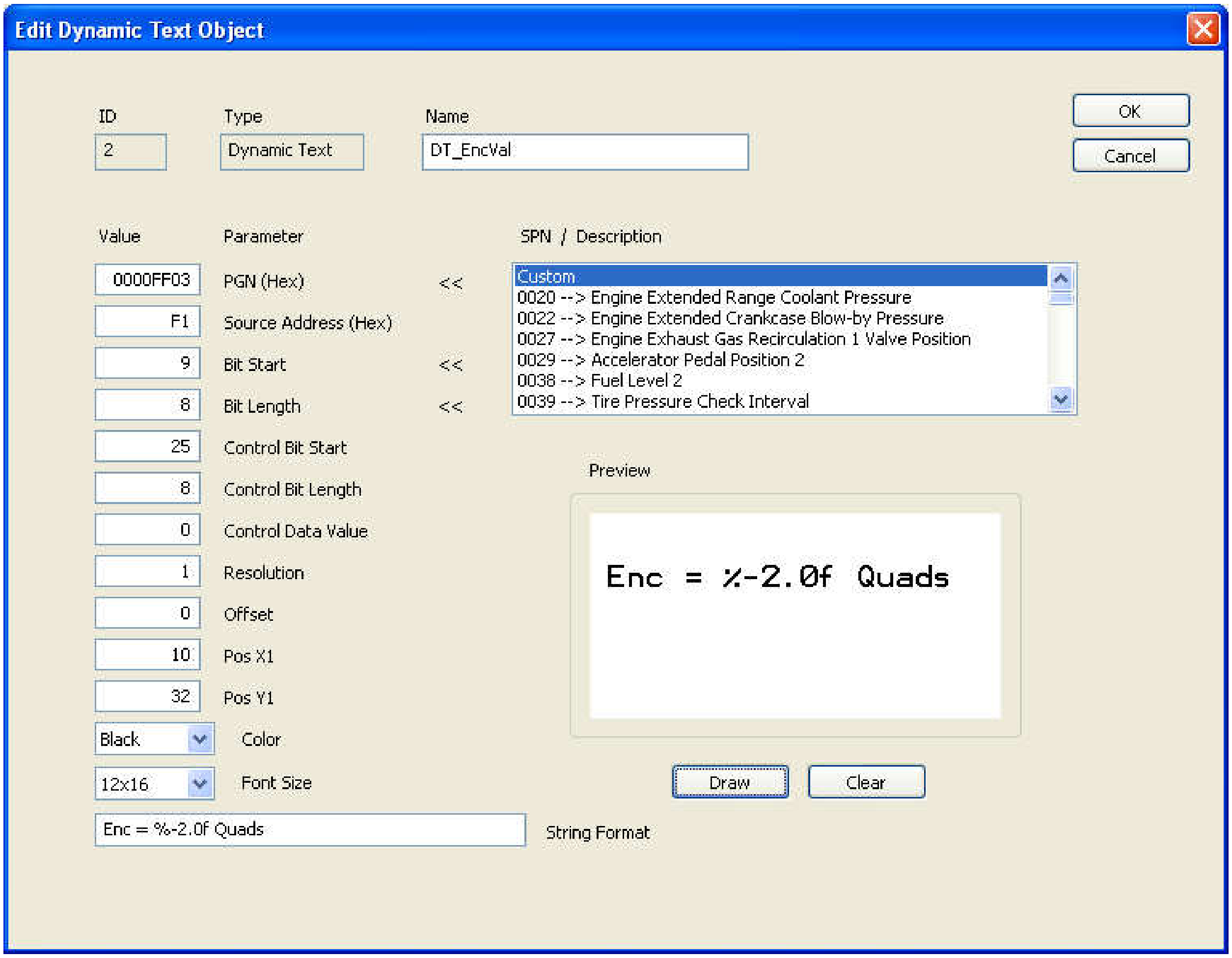
Where: XX is the encoder data the bar graph responds to.
YY is the control field used for multiplexed data.

The Min and Max value fields are used to proportion the value for the indicator according to the following equation:

Bar Pct = (Value/(Max-Min))*100

The bar graph is horizontally oriented with the upper left coordinates (x,y) at (10,10). The full length of the bar graph is 100 pixels and the width is 20 pixels.

PGN Watchdog with Dynamic Text: The next object that will be created is the Dynamic Text. This object allows the monitored value to be displayed as simple text and will be instantiated by the PGN Watchdog. Refer to the following screen shot.



Edit Dynamic Text Object

ID	Type	Name
2	Dynamic Text	DT_EncVal

Value	Parameter	
0000FF03	PGN (Hex)	<<
F1	Source Address (Hex)	
9	Bit Start	<<
8	Bit Length	<<
25	Control Bit Start	
8	Control Bit Length	
0	Control Data Value	
1	Resolution	
0	Offset	
10	Pos X1	
32	Pos Y1	
Black	Color	
12x16	Font Size	

SPN / Description

Custom
0020 --> Engine Extended Range Coolant Pressure
0022 --> Engine Extended Crankcase Blow-by Pressure
0027 --> Engine Exhaust Gas Recirculation 1 Valve Position
0029 --> Accelerator Pedal Position 2
0038 --> Fuel Level 2
0039 --> Tire Pressure Check Interval

Preview

Enc = %-2.0f Quads

String Format: Enc = %-2.0f Quads

Figure B-3

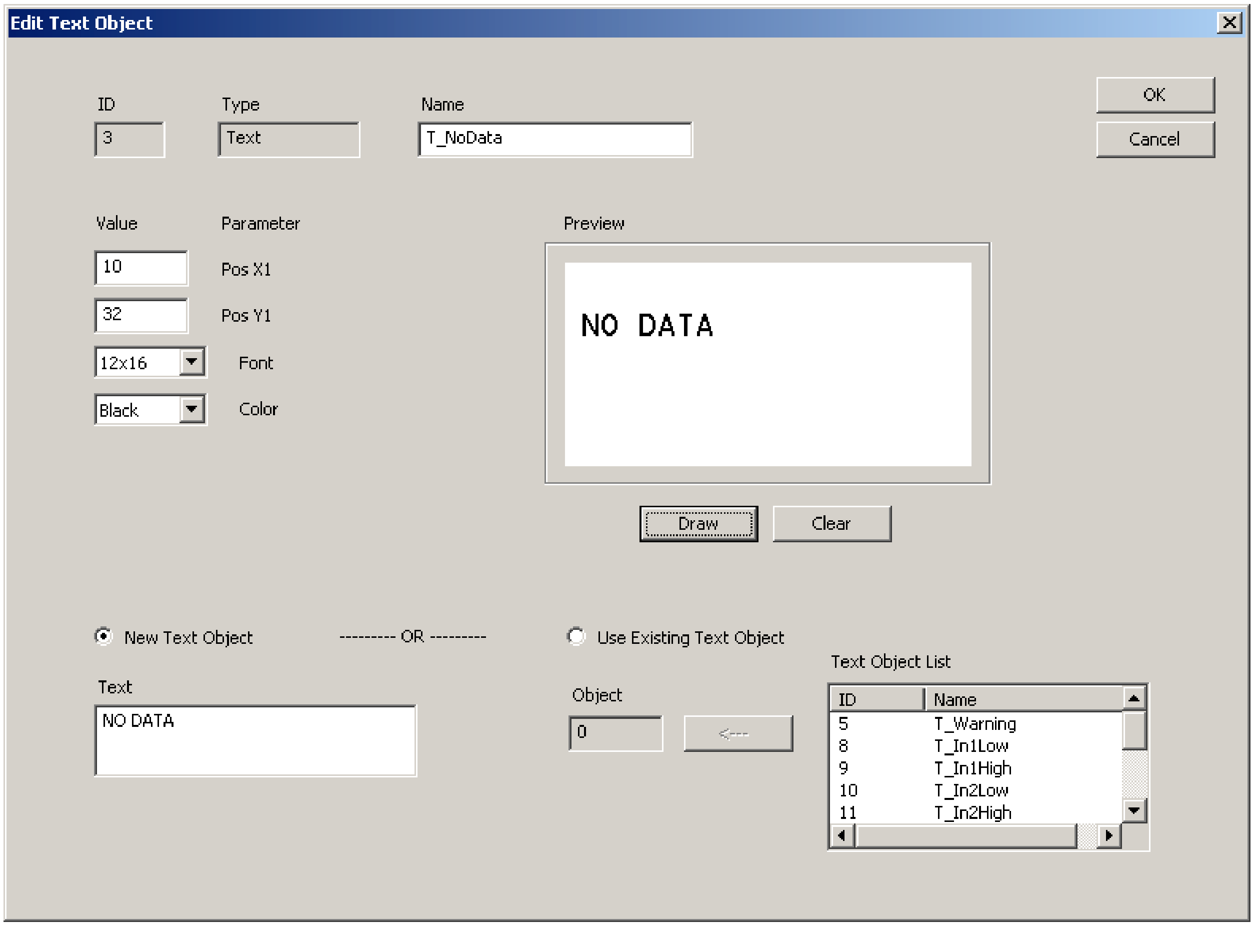
The first seven parameters are the same as for the bar graph illustrated previously. The Resolution is a scalar value the raw data will be multiplied by and the Offset is the value added to the product. In this example, the raw data is displayed with no modifications.

Pos X1 and Pos Y1 refer to the upper left corner of the text on the screen.

The string format controls how the numerical value is displayed and uses C-style formatting of a number of type ‘float’ for the printf statement. A value of 20 will result in the following on the screen.

Enc = 20 Quads

Next, the Static Text object displaying “NO DATA” will be created. Refer to the following screen shot.



ID: 3
Type: Text
Name: T_NoData

Value	Parameter
10	Pos X1
32	Pos Y1
12x16	Font
Black	Color

NO DATA

● New Text Object

○ Use Existing Text Object

Text: NO DATA

Object: 0

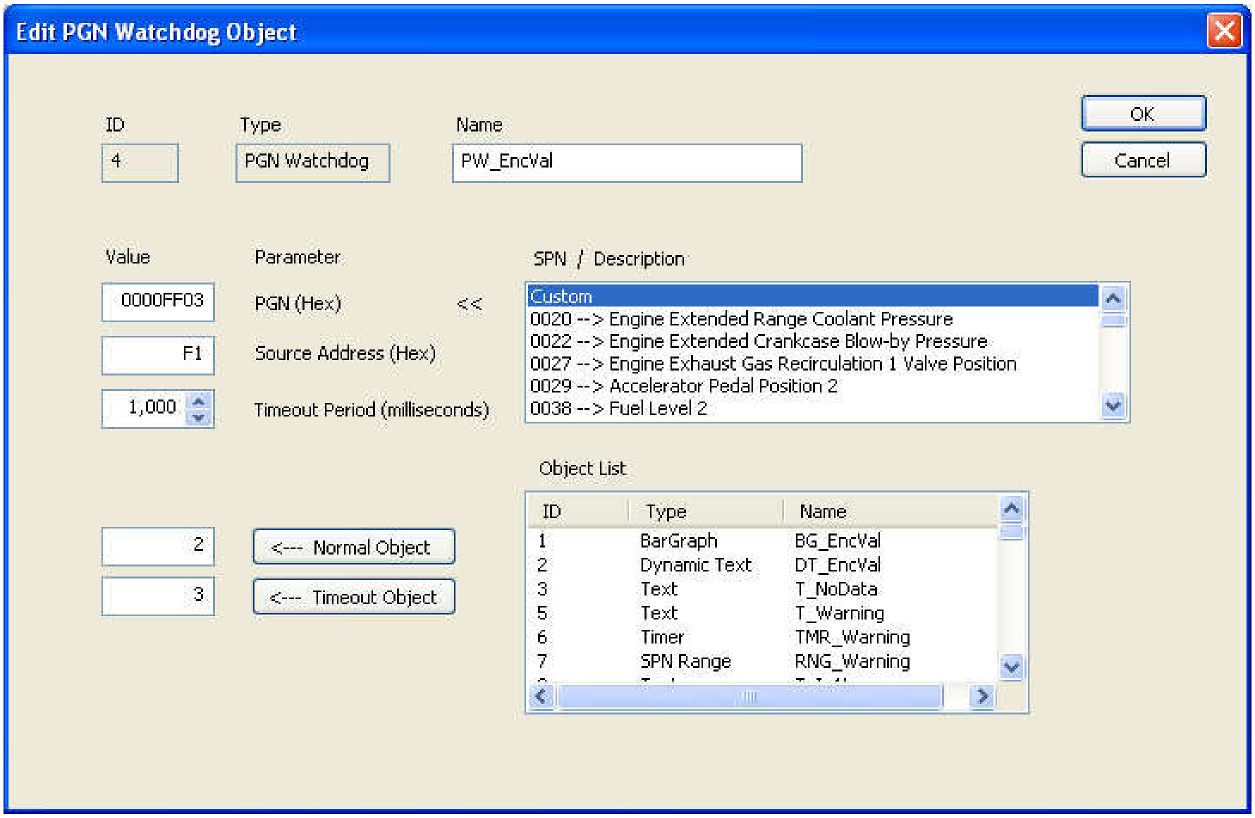
Text Object List

ID	Name
5	T_Warning
8	T_In1Low
9	T_In1High
10	T_In2Low
11	T_In2High

Figure B-4

In this example we want to create a new text object. If another Static Text object was created but with different coordinates than desired, the “Use Existing Text Object” option can be selected assigning new coordinates to the object.

Now, the PGN Watchdog object will be created that will reference the previously created objects. Refer to the following screen shot.



Edit PGN Watchdog Object

ID: 4
Type: PGN Watchdog
Name: PW_EncVal

Value	Parameter
0000FF03	PGN (Hex)
F1	Source Address (Hex)
1,000	Timeout Period (milliseconds)
2	<--- Normal Object
3	<--- Timeout Object

SPN / Description

Custom
0020 --> Engine Extended Range Coolant Pressure
0022 --> Engine Extended Crankcase Blow-by Pressure
0027 --> Engine Exhaust Gas Recirculation 1 Valve Position
0029 --> Accelerator Pedal Position 2
0038 --> Fuel Level 2

Object List

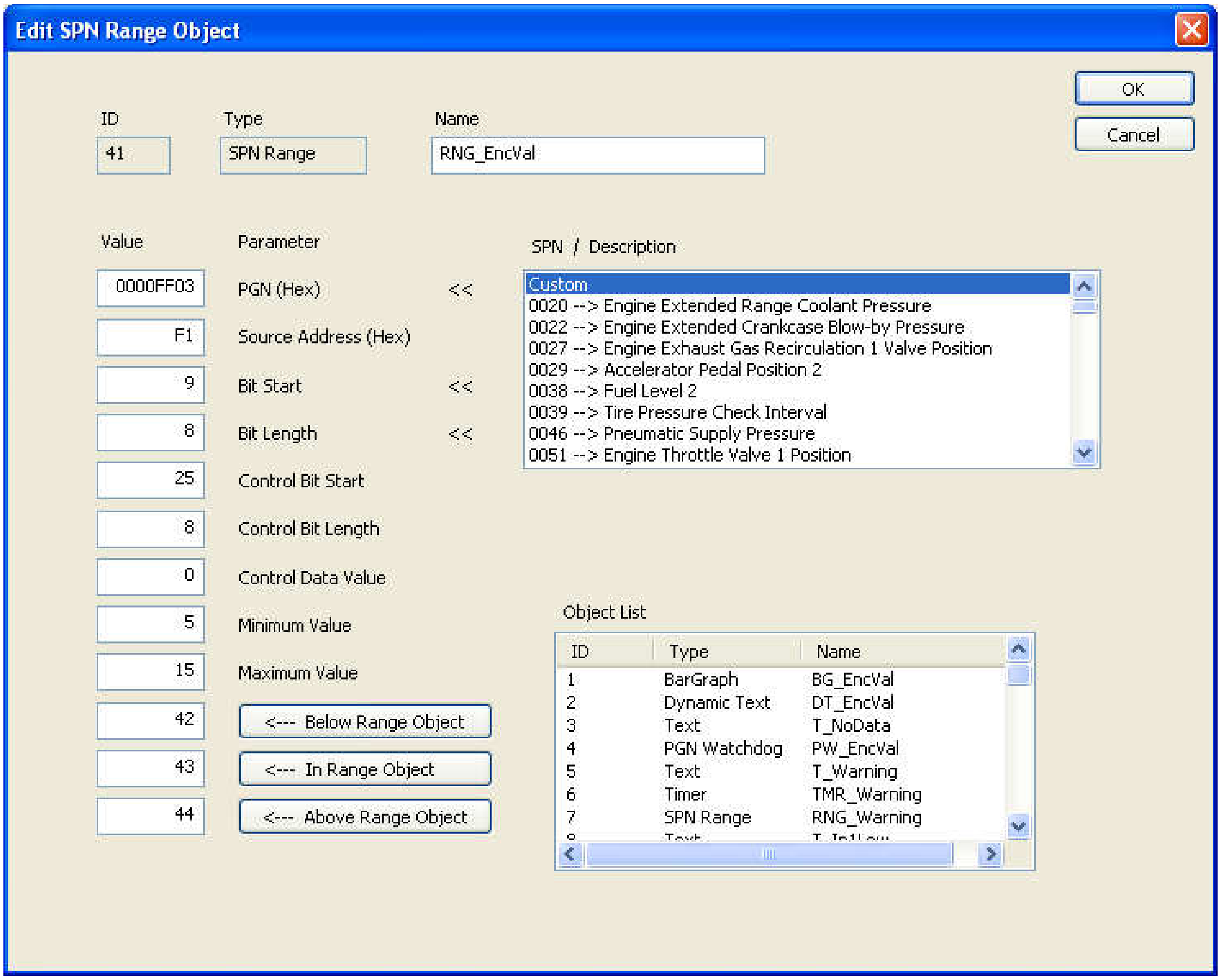
ID	Type	Name
1	BarGraph	BG_EncVal
2	Dynamic Text	DT_EncVal
3	Text	T_NoData
5	Text	T_Warning
6	Timer	TMR_Warning
7	SPN Range	RNG_Warning

Figure B-5

The PGN and source address are the only two parameters checked against incoming messages. The Timeout Period is the amount of time that must pass without receiving a message with the specified PGN and source address before the Normal Object is removed and the Timeout Object is instantiated.

In this example the normal and timeout objects were created first. This does not necessarily need to be the case. If the PGN Watchdog was created first it can always be edited after the others are created.

SPN Range Object: First, create Static Text objects for the texts “Below 5”, “Between 5 and 15”, and “Above 15”. Make sure they have the same coordinates so they occupy the same space. Although, this does not need to be the case. Then, create the SPN Range Object. Refer to the following screenshot for the SPN Range Object.



Edit SPN Range Object

ID: 41
Type: SPN Range
Name: RNG_EncVal

Value	Parameter	
0000FF03	PGN (Hex)	<<
F1	Source Address (Hex)	
9	Bit Start	<<
8	Bit Length	<<
25	Control Bit Start	
8	Control Bit Length	
0	Control Data Value	
5	Minimum Value	
15	Maximum Value	
42	<--- Below Range Object	
43	<--- In Range Object	
44	<--- Above Range Object	

SPN / Description

Custom
0020 --> Engine Extended Range Coolant Pressure
0022 --> Engine Extended Crankcase Blow-by Pressure
0027 --> Engine Exhaust Gas Recirculation 1 Valve Position
0029 --> Accelerator Pedal Position 2
0038 --> Fuel Level 2
0039 --> Tire Pressure Check Interval
0046 --> Pneumatic Supply Pressure
0051 --> Engine Throttle Valve 1 Position

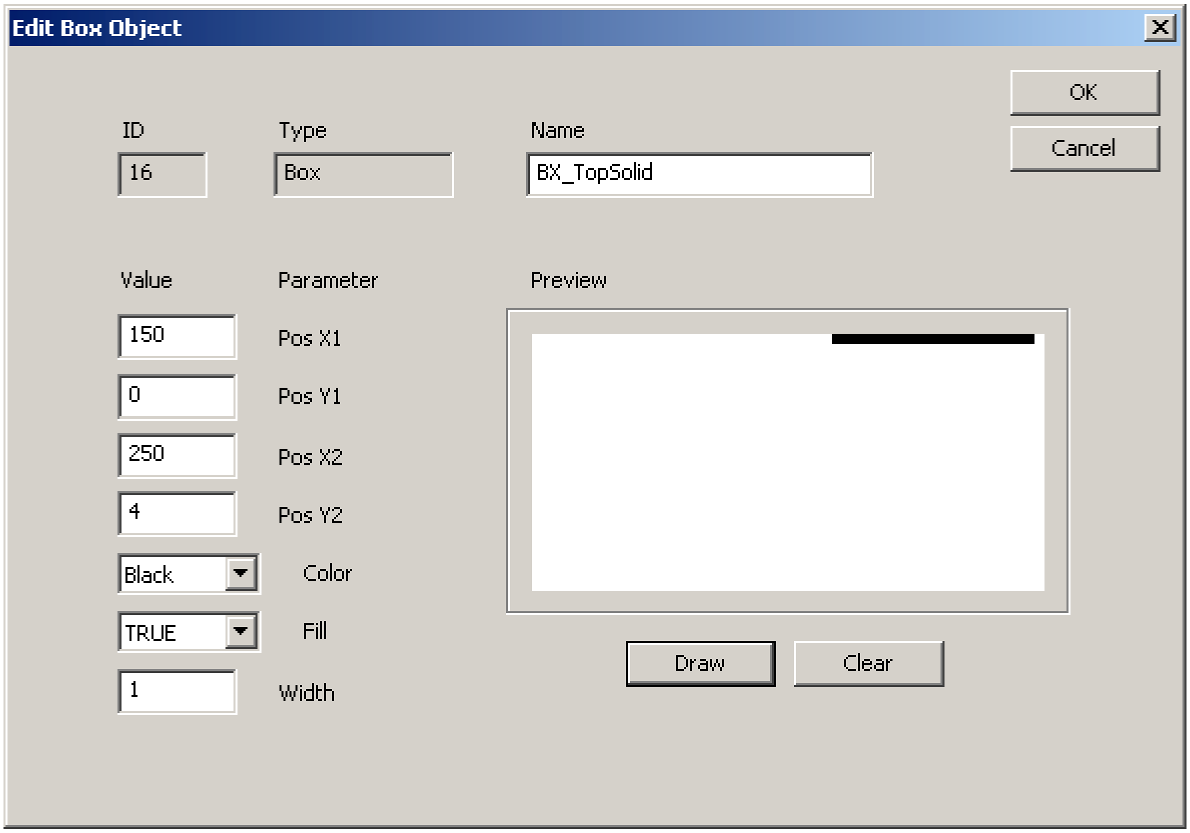
Object List

ID	Type	Name
1	BarGraph	BG_EncVal
2	Dynamic Text	DT_EncVal
3	Text	T_NoData
4	PGN Watchdog	PW_EncVal
5	Text	T_Warning
6	Timer	TMR_Warning
7	SPN Range	RNG_Warning

Figure B-6

The unique parameters for this object are the Minimum Value and the Maximum Value. These values specify the range for checking if the monitored parameter is below, within, or above the range. It is important to note that the range values must be specified using the raw parameter value transmitted on the CAN bus. No scalar or offset is assumed. For example, many temperature ranges specify a range between –40C and +210C with a resolution of 1 deg C/bit (Engine Coolant, SPN 110). If the desired range is 0C to 70C, the values for the minimum value and maximum value need to be the raw values 40 and 110 respectively. These are the values that would be seen if a CAN sniffing tool were used to observe the communication bus.

Custom Gauge Object: The Custom Gauge Object is an object that either instantiates or removes other objects according to the monitored value. As mentioned earlier, the top and bottom bars toggle between solid and hollow fill using a Timer Object. First, create all of the bars with solid fill. Refer to the following screen shot.



Edit Box Object

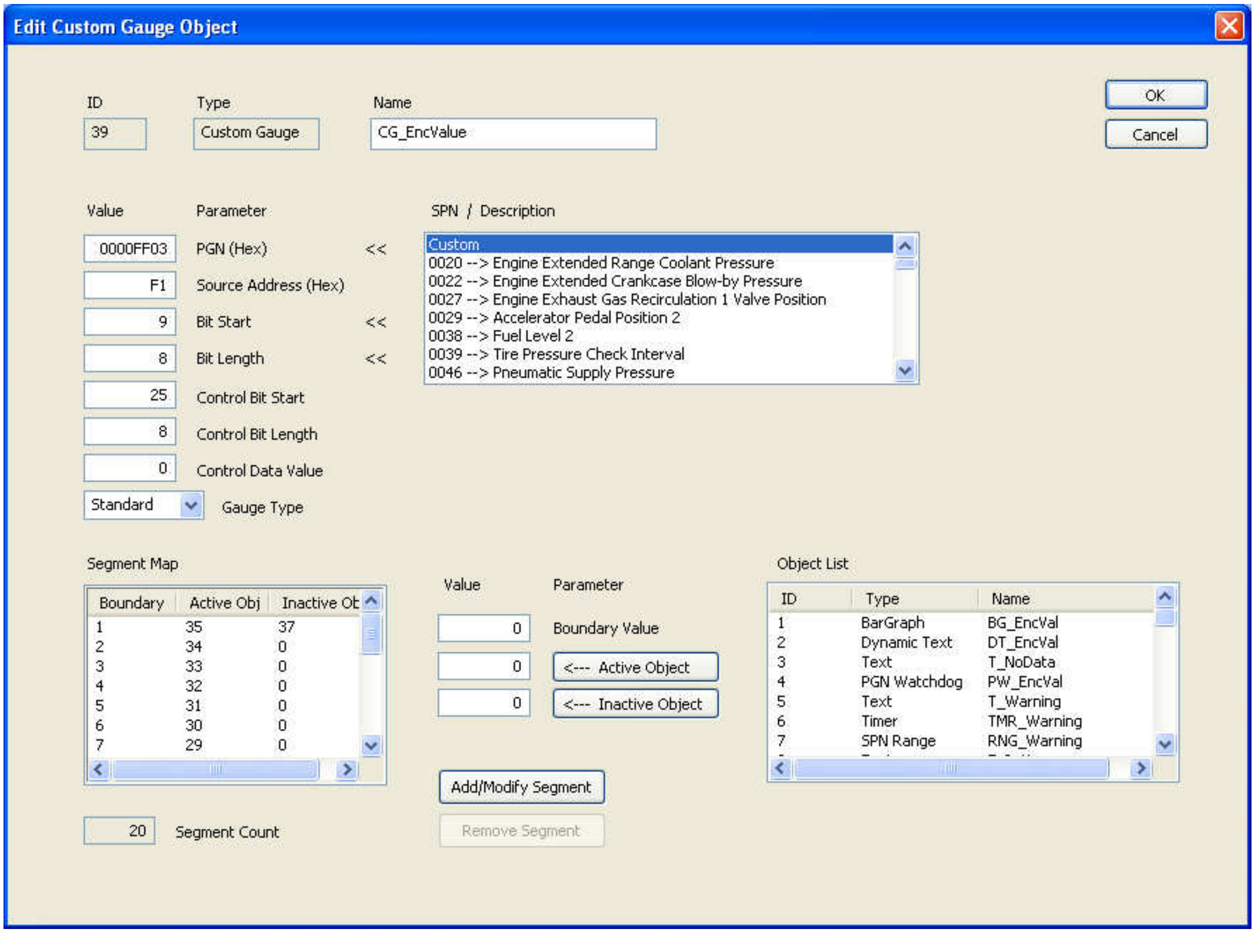
ID	Type	Name
16	Box	BX_TopSolid

Value	Parameter
150	Pos X1
0	Pos Y1
250	Pos X2
4	Pos Y2
Black	Color
TRUE	Fill
1	Width

Figure B-7

Since this is the top bar, select this object in the main screen and click Copy Object then Edit Object. Change the Fill from True to False. Then, a Timer Object can be created that will toggle between the solid and hollow versions. Do the same for the bottom most object.

The Custom Gauge object is created next. Refer to the following screen shot.



Edit Custom Gauge Object

ID	Type	Name
39	Custom Gauge	CG_EncValue

Value	Parameter
0000FF03	PGN (Hex)
F1	Source Address (Hex)
9	Bit Start
8	Bit Length
25	Control Bit Start
8	Control Bit Length
0	Control Data Value
Standard	Gauge Type

SPN / Description

Custom
0020 --> Engine Extended Range Coolant Pressure
0022 --> Engine Extended Crankcase Blow-by Pressure
0027 --> Engine Exhaust Gas Recirculation 1 Valve Position
0029 --> Accelerator Pedal Position 2
0038 --> Fuel Level 2
0039 --> Tire Pressure Check Interval
0046 --> Pneumatic Supply Pressure

Segment Map

Boundary	Active Obj	Inactive Ob
1	35	37
2	34	0
3	33	0
4	32	0
5	31	0
6	30	0
7	29	0

Value	Parameter
0	Boundary Value
0	<--- Active Object
0	<--- Inactive Object

20 Segment Count

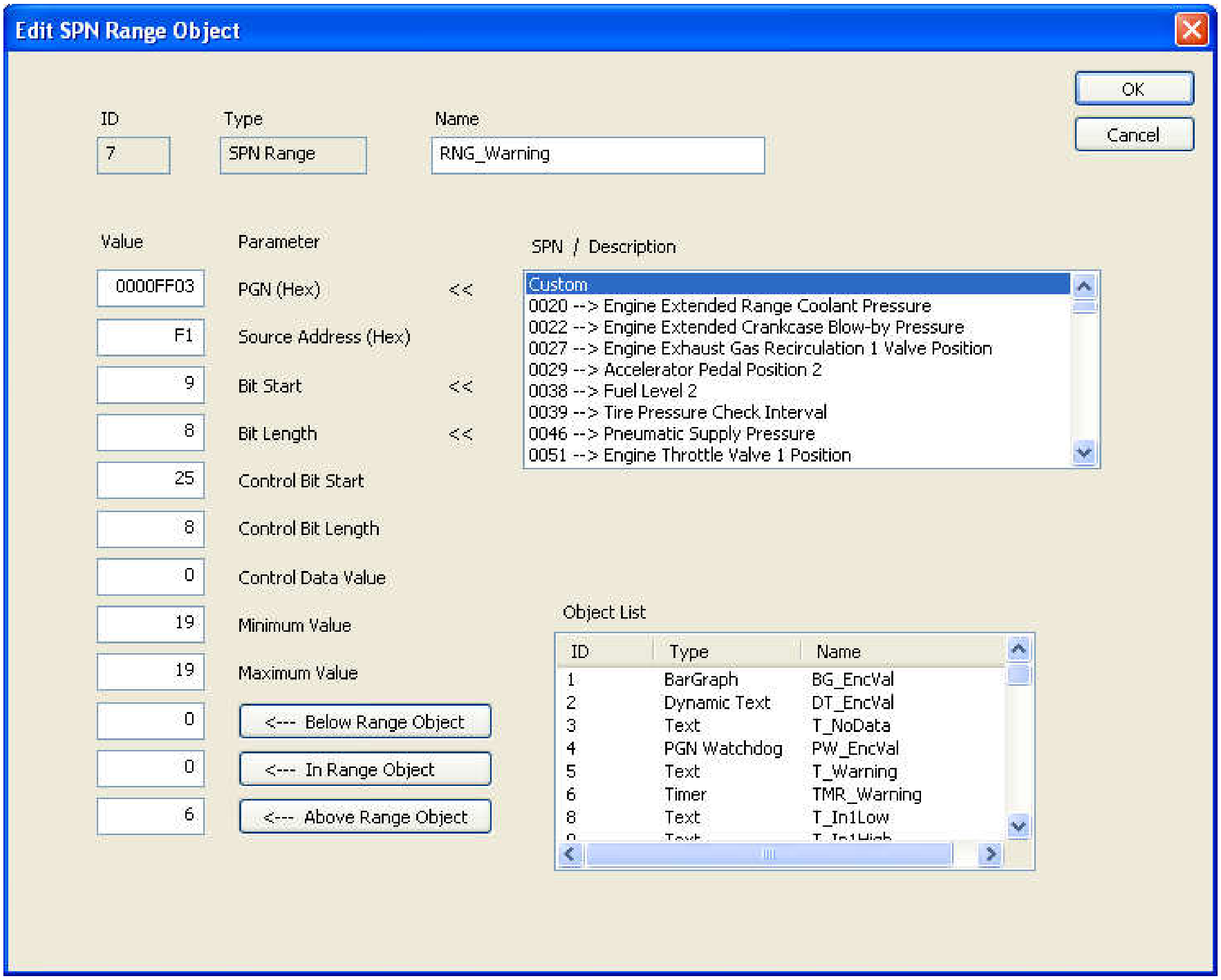
Object List

ID	Type	Name
1	BarGraph	BG_EncVal
2	Dynamic Text	DT_EncVal
3	Text	T_NoData
4	PGN Watchdog	PW_EncVal
5	Text	T_Warning
6	Timer	TMR_Warning
7	SPN Range	RNG_Warning

Figure B-8

The unique parameters for this object are the Gauge Type and the Segment Map. Segment Map maps the boundary value to an object. The object is either active or inactive depending on if the measured parameter is equal to or below the boundary value respectively. The Null Object (value of zero) is used if no object is assigned. The options for Gauge Type are either Standard or Pointer. The Standard type displays all active objects equaling or less than the boundary, otherwise the inactive object is displayed (or nothing in the event of the null object). With Pointer type only the active object is displayed if the value is between adjacent entries. Example: Object 32 is displayed when the measured parameter value is 4. This is better suited for gauges that implement a needle as opposed to bars where only one needle should be displayed at a time.

WARNING message: The ‘warning’ message will blink if the encoder value is 20 or above. The blinking is done with a Timer object that toggles between the ‘warning’ Static Text object and the null object. The Timer object is instantiated by an SPN Value object where the ‘above range object’ is assigned the Timer Object ID just described and the ‘below’ and ‘in range’ object are assigned the null object. Refer to the following screen shot.



ID	Type	Name
7	SPN Range	RNG_Warning

Value	Parameter
0000FF03	PGN (Hex)
F1	Source Address (Hex)
9	Bit Start
8	Bit Length
25	Control Bit Start
8	Control Bit Length
0	Control Data Value
19	Minimum Value
19	Maximum Value
0	Below Range Object
0	In Range Object
6	Above Range Object

SPN / Description

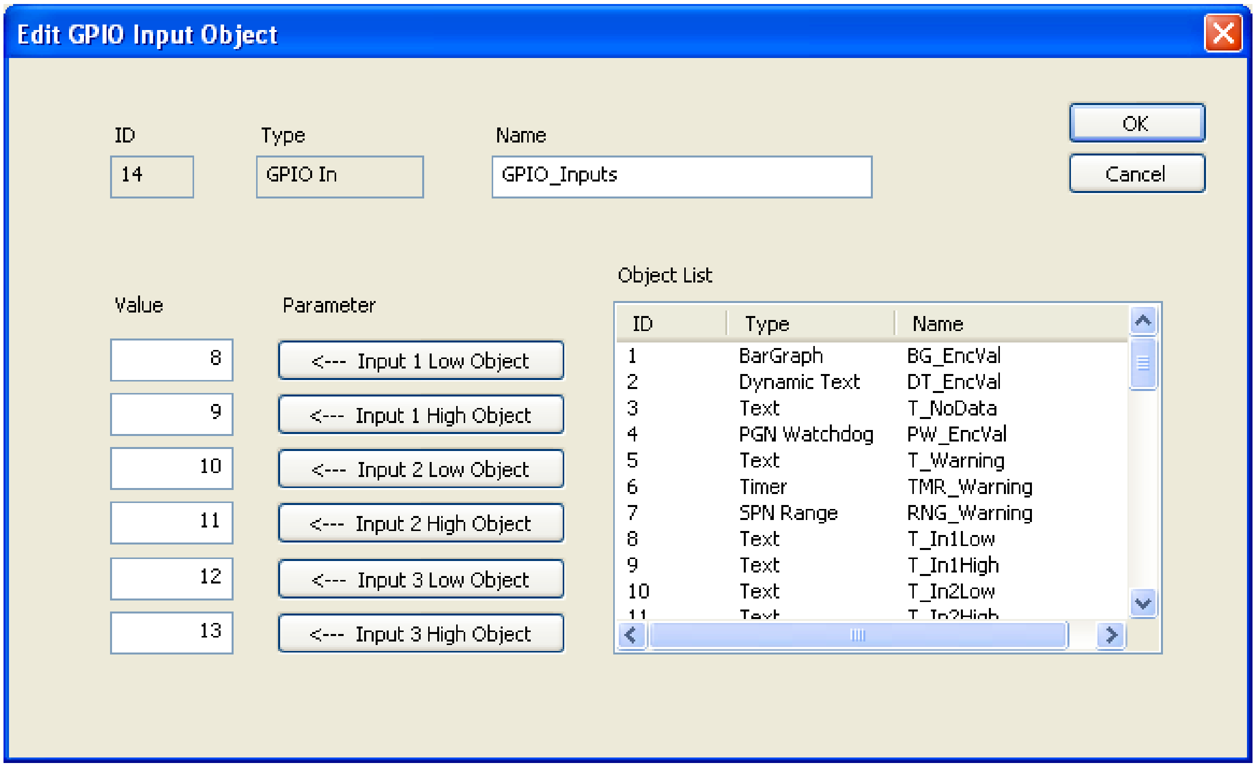
- Custom
- 0020 --> Engine Extended Range Coolant Pressure
- 0022 --> Engine Extended Crankcase Blow-by Pressure
- 0027 --> Engine Exhaust Gas Recirculation 1 Valve Position
- 0029 --> Accelerator Pedal Position 2
- 0038 --> Fuel Level 2
- 0039 --> Tire Pressure Check Interval
- 0046 --> Pneumatic Supply Pressure
- 0051 --> Engine Throttle Valve 1 Position

Object List

ID	Type	Name
1	BarGraph	BG_EncVal
2	Dynamic Text	DT_EncVal
3	Text	T_NoData
4	PGN Watchdog	PW_EncVal
5	Text	T_Warning
6	Timer	TMR_Warning
8	Text	T_In1Low
9	Text	T_In1High

Figure B-9

GPIO Inputs: Finally, a GPIO Input object will be created to monitor the three discrete inputs. For each input, create two Static Text objects with the same coordinates saying “Input 1 Low” and “Input 1 High”. Since only one will be displayed at a time they can occupy the same area on the screen. The same is then done for inputs two and three. Then, the GPIO Input object is created. Refer to the following screen shot.



ID	Type	Name
14	GPIO In	GPIO_Inputs

Value	Parameter
8	<--- Input 1 Low Object
9	<--- Input 1 High Object
10	<--- Input 2 Low Object
11	<--- Input 2 High Object
12	<--- Input 3 Low Object
13	<--- Input 3 High Object

Object List

ID	Type	Name
1	BarGraph	BG_EncVal
2	Dynamic Text	DT_EncVal
3	Text	T_NoData
4	PGN Watchdog	PW_EncVal
5	Text	T_Warning
6	Timer	TMR_Warning
7	SPN Range	RNG_Warning
8	Text	T_In1Low
9	Text	T_In1High
10	Text	T_In2Low
11	Text	T_In2High

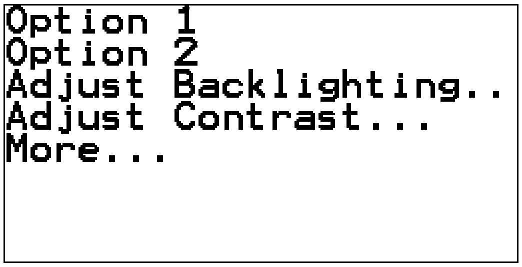
Figure B-10

In this example, text is displayed indicating the level of each input accordingly.

Appendix C – Menu Example with Embedded Objects

This example will demonstrate the menu ability of the MicroDisplay. It will consist of a root menu with five selectable items with one invoking a sub menu and two more invoking embedded objects, these being the Backlight Adjust and Contrast Adjust screens. Scrolling through the selections will simply invert one of the options. This will act as the cursor. Pressing Enter will launch the object assigned to the menu item. Pressing the Menu button (the left most button) will back out one menu level or, in the event that the root menu is displayed, unload the menu and launch the previously displayed object.

The root menu will look as follows:

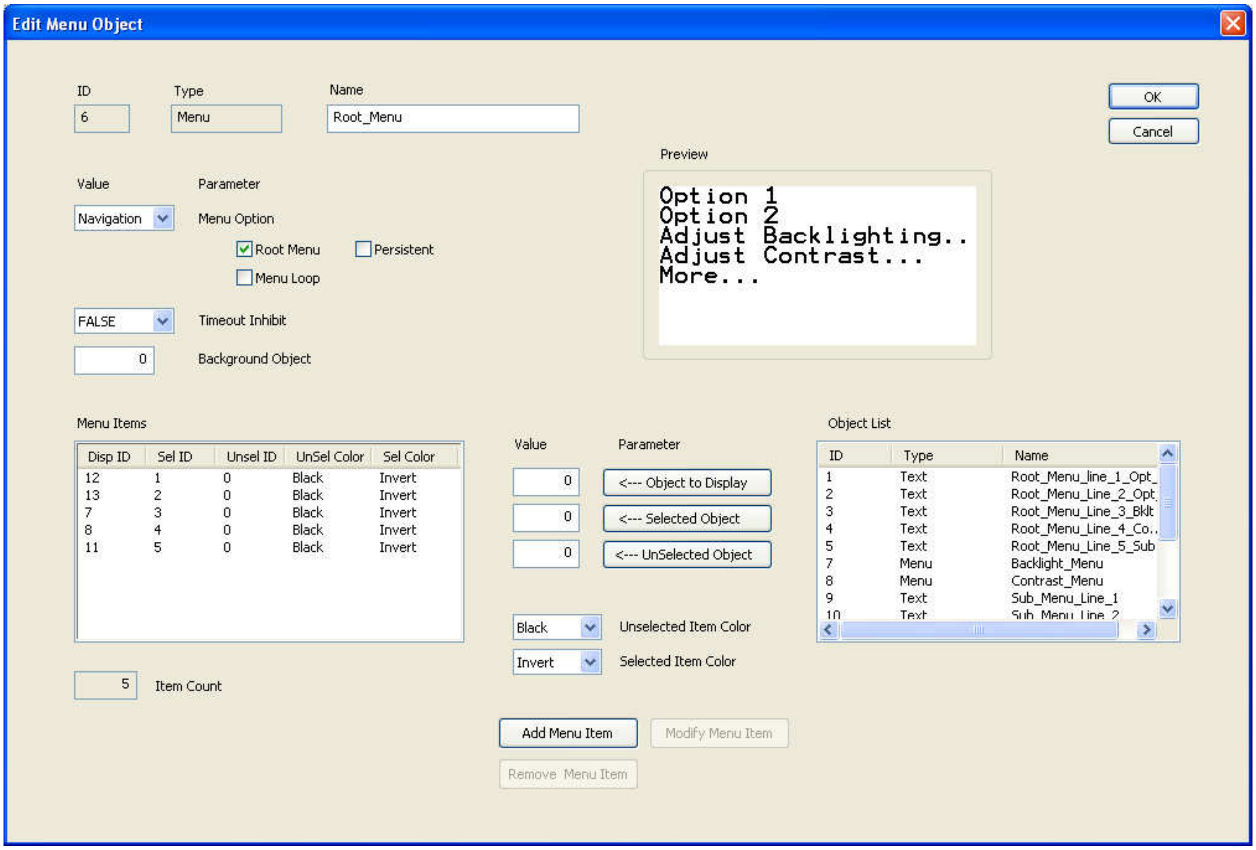


```
Option 1
Option 2
Adjust Backlighting..
Adjust Contrast...
More...
```

Figure C-1

Each line is a Static Text object. These are created first for ease of creating/modifying the Menu Object.

Next, the Menu Object is created by clicking on New Object then selecting Menu then OK. The completed screen follows.



ID: 6 Type: Menu Name: Root_Menu

Value	Parameter
Navigation	Menu Option
	☑ Root Menu ☐ Persistent
	☐ Menu Loop
FALSE	Timeout Inhibit
0	Background Object

Preview

```
Option 1
Option 2
Adjust Backlighting..
Adjust Contrast...
More...
```

Menu Items

Disp ID	Sel ID	Unsel ID	UnSel Color	Sel Color
12	1	0	Black	Invert
13	2	0	Black	Invert
7	3	0	Black	Invert
8	4	0	Black	Invert
11	5	0	Black	Invert

5 Item Count

Value	Parameter
0	<--- Object to Display
0	<--- Selected Object
0	<--- UnSelected Object
Black	Unselected Item Color
Invert	Selected Item Color

Object List

ID	Type	Name
1	Text	Root_Menu_line_1_Opt_
2	Text	Root_Menu_Line_2_Opt
3	Text	Root_Menu_Line_3_Bklt
4	Text	Root_Menu_Line_4_Co...
5	Text	Root_Menu_Line_5_Sub
7	Menu	Backlight_Menu
8	Menu	Contrast_Menu
9	Text	Sub_Menu_Line_1
10	Text	Sub_Menu_Line_2

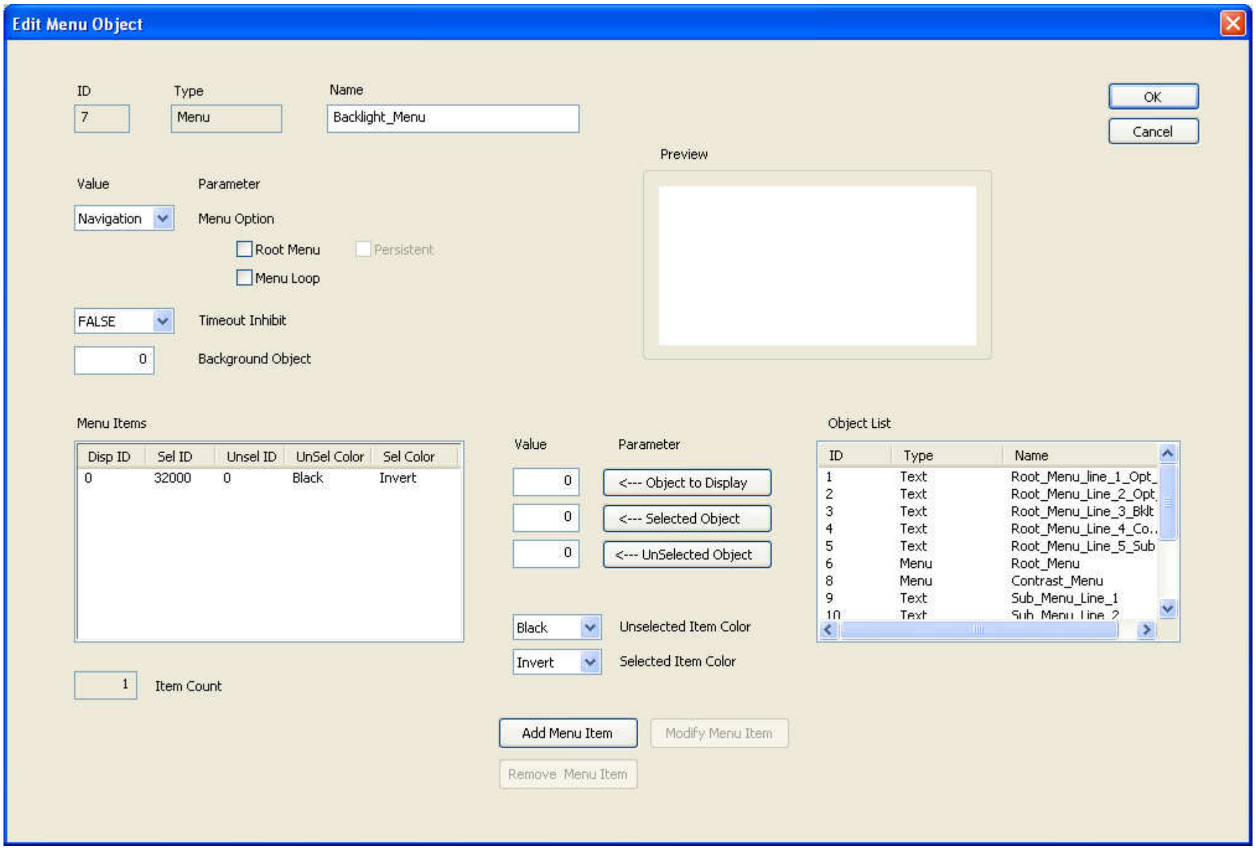
Figure C-2

The Root Menu checkbox must be checked. This enables the MicroDisplay to launch the object when the left most button is pressed.

If the Object to Display is not created yet then assign the same value as the Selected Object. Once these objects are created the menu object can be edited and the Disp ID fields completed.

Next, the sub menu for adjusting the backlighting is created. The Adjust Backlight and Adjust Contrast are embedded objects that are part of the MicroDisplay application firmware. The ID's start at 32000. They are intended to be part of, and launched from a menu system.

A new Menu object is created. Refer to the following screen shot.



ID: 7
Type: Menu
Name: Backlight_Menu

Value: Navigation — Parameter: Menu Option
□ Root Menu □ Persistent
□ Menu Loop
FALSE — Timeout Inhibit
0 — Background Object

Menu Items

Disp ID	Sel ID	Unsel ID	UnSel Color	Sel Color
0	32000	0	Black	Invert

1 — Item Count

Value	Parameter
0	<--- Object to Display
0	<--- Selected Object
0	<--- UnSelected Object
Black	Unselected Item Color
Invert	Selected Item Color

Object List

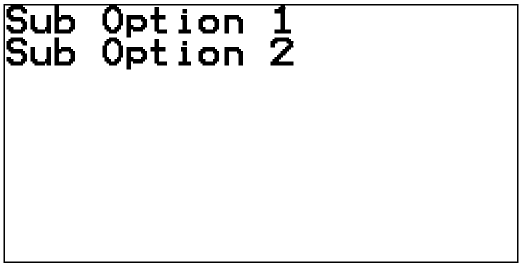
ID	Type	Name
1	Text	Root_Menu_line_1_Opt_
2	Text	Root_Menu_Line_2_Opt
3	Text	Root_Menu_Line_3_Bklt
4	Text	Root_Menu_Line_4_Co..
5	Text	Root_Menu_Line_5_Sub
6	Menu	Root_Menu
8	Menu	Contrast_Menu
9	Text	Sub_Menu_Line_1
10	Text	Sub_Menu_Line_2

Figure C-3

Notice that this menu item only has one entry and that the Selected Object ID is assigned the embedded object's ID of 32000. Nothing gets displayed when pressing the Enter button since its purpose was not to display another object. Because it is assigned as the Selected Object and Time Out Inhibit is not selected, if no buttons are pressed within five seconds the screen will be unloaded and the previously displayed object (before invoking the menu) is re-displayed. Pressing the menu button when the backlight screen is displayed will simply navigate back to the calling menu screen, in this example, the root menu.

The screen for contrast adjusting is done the same way with the exception that the Selected Object's value is 32001.

Next, the sub menu "More…" is created. Refer to the screen shot.



Sub Option 1
Sub Option 2

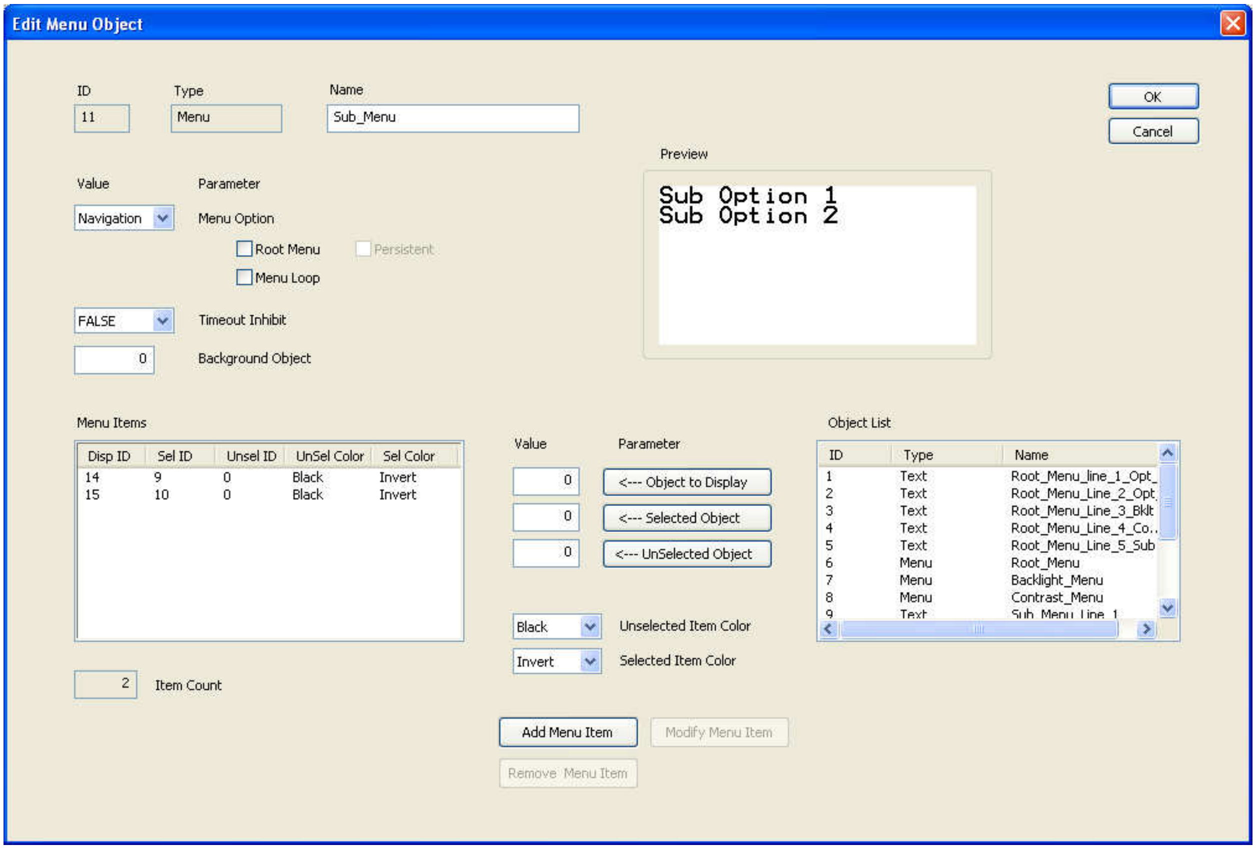
Figure C-4

First, the menu lines are created with Static Text objects. A new menu object is created just like the root menu with the exception that the Root Menu checkbox is unchecked.

The displayed objects are created that will be launched when a selection is made. For this example, they will simply be Static Text objects with the following text:

- Option 1
- Option 2
- Sub Option 1
- Sub Option 2

Finally, both the root menu and sub menu are modified to reference the proper objects. Refer to the screen shots.



ID: 11
Type: Menu
Name: Sub_Menu

Preview

Sub Option 1
Sub Option 2

Value	Parameter
Navigation	Menu Option
	□ Root Menu □ Persistent
	□ Menu Loop
FALSE	Timeout Inhibit
0	Background Object

Menu Items

Disp ID	Sel ID	Unsel ID	UnSel Color	Sel Color
14	9	0	Black	Invert
15	10	0	Black	Invert

2 Item Count

Value	Parameter
0	<--- Object to Display
0	<--- Selected Object
0	<--- UnSelected Object
Black	Unselected Item Color
Invert	Selected Item Color

Object List

ID	Type	Name
1	Text	Root_Menu_line_1_Opt_
2	Text	Root_Menu_Line_2_Opt_
3	Text	Root_Menu_Line_3_Bklt
4	Text	Root_Menu_Line_4_Co..
5	Text	Root_Menu_Line_5_Sub
6	Menu	Root_Menu
7	Menu	Backlight_Menu
8	Menu	Contrast_Menu
9	Text	Sub_Menu_Line_1

Figure C-5

Appendix D – Memory Size

The MicroDisplay has approximately 392 Kbytes of object data storage. The table below gives an indication of how many types of objects can be stored.

Object Type	Size	Max # stored
Full Screen Bitmap (2 bit)	8192	47
Full Screen Bitmap (1 bit)	4096	95
Circles	16	24,500
Boxes	18	21,700
Lines	16	24,500
Bar Graphs	36	10,800
Group Objects	6 + (2 bytes per group item)	392K / (Size)
Static Text	16 + (1 byte per character)	392K / (Size)
Dynamic Text	32 + (1 byte per character)	392K / (Size)
Bitmap (monochrome)	18 + (Pixel Count/8)	392K / (Size)
Bitmap (2 bit Grayscale)	18 + (Pixel Count/4)	392K / (Size)

By far, bitmaps take up the most space depending on their size. If the same bitmap is desired in the same location but on multiple screens, only one needs to be stored and a Group Object then used to reference the bitmap. However, if the location needs to change then an identical bitmap but with new coordinates needs to be created.

The MicroDisplay provides the ability to obtain the amount of memory available for defining additional objects. For the currently selected MicroDisplay, click the ‘Report’ button on the main window. The remaining bytes available for assigning new objects will be displayed in the ‘Free Space’ box.

Appendix E – Troubleshooting

The 'Device' menu item is disabled (grayed out). The program requires the use of the 32 bit version of PCAN_USB.DLL. If the program cannot open the DLL then it will simply disable the 'Device' menu item. This allows users to still create and edit projects if the hardware is not installed on the machine being used.

Two things can cause the program to not be able to open the DLL:

1. The DLL is not found in the same directory as the executable or in the Window's system directory. For 32 bit Windows operating system this is the System32 subdirectory. For 64 bit Windows this is the SysWOW64 subdirectory. The 32 bit version, not the 64 bit version, of PCAN_USB.DLL must be placed in the SysWOW64.
2. Another application has the DLL already opened. Make sure another instance of the Configuration Tool or an application by Peak-System, such as PCANView, is not already open.

It is recommended that only one DLL is present on a computer and that it is located in the system directory. If two applications attempt to use the DLL but the DLLs are in different locations a blue memory dump will occur. Example: a copy of the DLL is in the Configuration Tool's directory and also the system directory. If the Configuration Tool is started, it will open the DLL located in its directory. Then, if PCANView is started it will use either the DLL in its directory, if it exists, or the DLL in the system directory. As long as both use the same DLL, no problems will occur.