An ISO-9001 Company

Grayhill VUI Builder©

Software Developer's Manual

3DUM1940-1

Revision History

Revision	Date	Description
A	Apr-24-2015	Preliminary release
B	Jun-29-2015	Complete manual release • Changed tool name to VUI Builder© • Added Appendix showing objects and other details • Corrected and improved other sections
C	Jul-10-2015	• Added System Info Object • Added new way to access Send Command
D	Aug-24-2015	• Added Information about using new Variable object
E	Oct-03-2015	• Added information about multiple CAN buses • Added information about touch objects being able to kill themselves • Added information about “No Kill” and “One Shot” features added to Timer Object • Added information about new set default feature for backlight • Added information about Bitmap Object Display and Kill Commands that can specify different location for Bitmap Object • Added information about new camera object features • Added information about new Kill All option for Group Object
F	Nov-24-2015	• Added information about new Start/End Touch Event feature • Added information about using “PNG” or “JPEG” files for images
G	May 19, 2016	• Added description of warning issued when removing an object that is referenced by other objects • Updated Download command description • Updated Send Command description
H	Oct 7, 2016	• Added Miscellaneous Commands section, calibration warning
I	Oct 15, 2016	• Added information about new kill option and arrow buttons for Group Object • Added information about button: “Show Where Object Used” • Added note about CAN Monitoring Object limit • Added information about Model 3D70 (7 inch display) which impacted the following areas: o Added third input for Camera Monitoring Object o Added new pin numbers and rules for GPIO Input and Output Objects o Added Analog Input Object o Added Buzzer Object o Added Audio Output Object
J	May 25, 2017	• Added support for Model 3D2104 (10.4 inch display)which impacted the following areas: o Added fourth input for Camera Monitoring Object o Added new pin numbers and rules for GPIO Input and Output Objects o Added Button Object
K	Aug 9, 2019	• Added support for creating USB flash drive project installer • Added support for creating project binary file • New look for main window GUI
L	Jul-30-2021	• Added support for 3D101 model display • Added buzzer level support
N	Feb-28-2022	• Added support for standard or extended CAN message type selection with CAN monitoring and CAN transmit objects
P	Jan-05-2024	• Added menu item to enable/disable Preview clear on object selection • Update revision to sync with Tool and App

Table of Contents

1. Introduction

This Software Developer's Manual provides instructions on how to configure a Grayhill 3Dxx Color Display device using the Grayhill PC-based VUI Builder©(pronounced "view builder") software.

The 3Dxx Color Display programming paradigm involves using the VUI Builder© software to 1) create a group of objects, 2) save the objects as a project, and 3) download the project to the 3Dxx Color Display unit over the CAN bus. Objects programmed into the 3Dxx Color Display can be activated (displayed) or deactivated (killed) by an ECU via CAN message, by other objects (such as timers or touch screen inputs), or by a command from the VUI Builder© software.

The 3Dxx Color Display hardware runs a Grayhill application program under a Linux operating system. This application program reads objects from a project file stored in its internal Flash file system and uses these objects to control operation of the 3Dxx Color Display.

Standalone operation of the 3Dxx Color Display is possible by using the VUI Builder© software to configure touch screen objects. These touch screen objects can be used to change screens and activate (display) or deactivate (kill) other objects, all without the intervention of an ECU.A default object, which is a special type of object that is automatically displayed on startup, can be used to initialize the touch screen operation.

There is also a PC version of the 3Dxx Color Display application software called the 3Dxx Color Display Simulator (hereafter 3Dxx Simulator). This software runs on a Windows PC and shows a window that provides a simulated view of what would appear on the actual 3Dxx Color Display hardware. The 3Dxx Simulator interfaces to the VUI Builder© software using a local pipe interface that works the same way as the CAN bus interface to the actual 3Dxx Color Display hardware. This allows the 3Dxx Simulator to respond to simulated CAN messages from the VUI Builder© software. The 3Dxx Simulator will also respond to mouse clicks and swipes within its preview area the same way the actual 3Dxx Color Display would respond to touch screen inputs.

For a detailed description of the various 3Dxx Color Display objects see Appendix A.

2. Applicable Software Versions

This VUI Builder© Manual describes features found on the following software versions:

- VUI Builder© Tool - Part Number: 3DPG1940-1-N (Rev N)
- 3Dxx Simulator - Part Number: 3DPG1940-3-N (Rev N)
- 3Dxx Linux Application - Part Number: 3DPR1940-2-N (Rev N)

3. Supported Hardware Products

The VUI Builder© Tool and the 3Dxx Simulator software revisions described above currently support these products:

- 3D50 Five Inch Color Display (480 x 800 pixels and up to two CAN buses)
- 3D70 Seven Inch Color Display (480 x 800 pixels and up to two CAN buses)
- 3D2104 10.4 Inch Color Display (1024 x 768 pixels and up to three CAN buses)
- 3D101 10.1 Inch Color Display (1280 x 800 pixels and up to three CAN buses)

The table below summarizes the key features of each of these models. Note that the features of a specific product may vary depending on the purchased hardware configuration.

Model Number	3D50	3D70	3D2104	3D101
Display Size (inches)	5	7	10.4	10.1
Pixel Count (w x h)	800 x 480	800 x 480	1024 x 768	1280 x 800

Model Number	3D50	3D70	3D2104	3D101
Touch Screen Input	Yes	Yes	Yes	Yes
Real Time Clock	Yes	Yes	Yes	Yes
CAN Ports	2	2	3	3
Camera Inputs	2	3	4	4
USB ports	1 (maintenance only)	1 (maintenance only)	1 (maintenance only)	1 (maintenance only)
RS232	1 (maintenance only)	1 (maintenance only)	1 (maintenance only)	1 (maintenance only)
Built-in Ethernet	0	1	1	1
Digital Input (dedicated)	1	4	0	0
Digital Output (dedicated)	1	4	0	0
Digital Input / Output	3	0	4	4
Analog Input	0	2	0	0
Audio Output	No	1 channel	No	No
Buzzer	No	Yes	Yes	Yes

4. CAN Bus Interface to 3Dxx Color Display Hardware

The VUI Builder© software interfaces to a Grayhill 3Dxx Color Display device through a CAN bus interface. The tool currently supports using a GridConnect® USB PCAN adapter. Please visit www.gridconnect.com for details regarding the USB PCAN adapter.

In order for the VUI Builder© software to communicate with the 3Dxx Color Display device, two things must be done:
The USB PCAN adapter software must be installed on the PC being used. For a minimum, only the PCAN USB software needs to be installed, but other tools such as PCAN-View can be installed if desired.
A copy of the Dynamic Link Library (PCANBasic.DLL) provided with the USB PCAN adapter needs to be saved in the same folder as the VUI Builder© executable. Make sure that the 32-bit version is used as opposed to the 64-bit version of the DLL. This file can be found in the GridConnect® web site under “Support -> Product Packages -> PCAN-USB package”. Then look in the download file “usb.zip” under the folder “PCAN-Basic API\Win32\”. If using the CD from a GC-CAN-USB hardware kit, the PCANBasic.dll file can be found in the folder “\PreRelease\PCAN-Basic v3.4 (CAN-FD Support)\Win32”.

5. VUI Builder© Software Installation

The VUI Builder© software can be obtained from this Grayhill web site:

https://www.grayhill.com/vui-builder

The above web site also has the latest version of this manual and some sample 3Dxx Color Display projects.

VUI Builder© software installation does not require running any kind of installation software. All that is required is to place the following files together in a folder on a PC:

- VUI Builder.exe – VUI Builder© executable
- mdhelper.bin – contains J1939standard CAN message definitions
- PCANBasic.dll – DLL to communicate with PCAN interface obtained from the USB PCAN adapter software described above.

6. 3Dxx Simulator Software Installation

The 3Dxx Simulator software can be obtained from this Grayhill web site:

https://www.grayhill.com/vui-builder

The 3Dxx Simulator software installation does not require running any kind of installation software. All that is required is to place the following files together in a folder on a PC (this folder can be different than the folder used for the VUI Builder©):

- 3DxxColorDisplayApp.exe – 3Dxx Simulator executable
- SDL.dll – contains SDL interface library

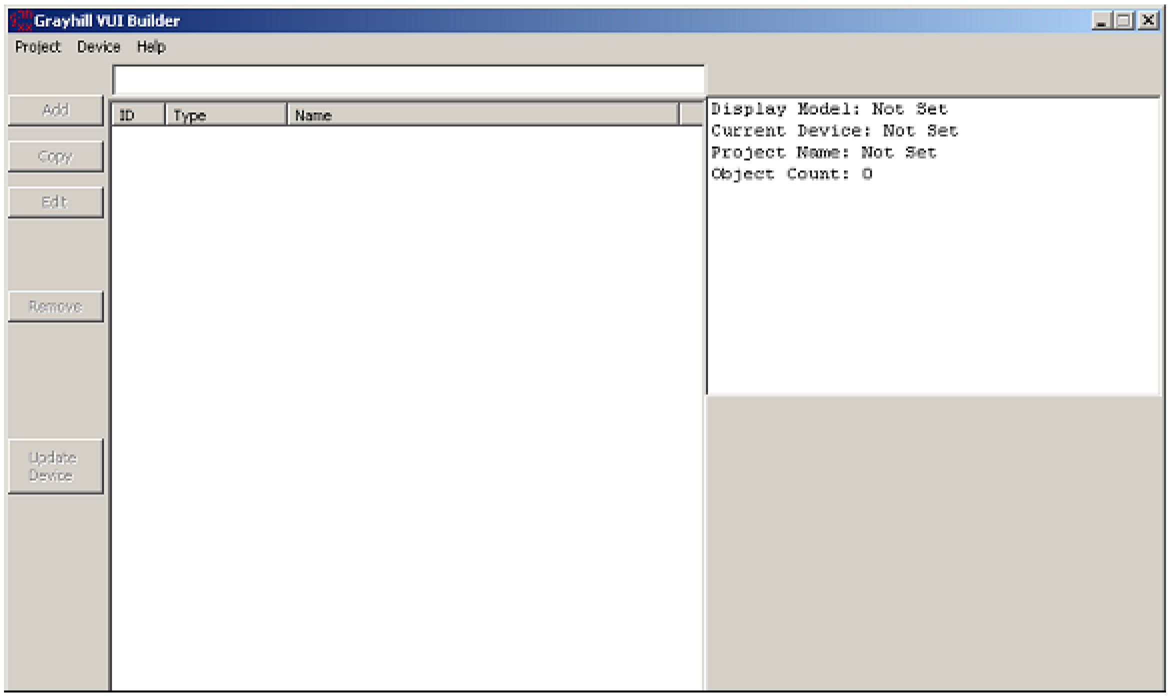
7. VUI Builder© Startup

Double click on "VUI Builder.exe" file to run the VUI Builder©.

Note that VUI Builder© creates a folder in the user's HOMEPATH called "Grayhill\3D50\startup".

If such a folder already exists it should be moved or renamed. By default VUI Builder© looks for saved projects in this folder and saves projects in this folder. Graphical image files associated with these projects are placed in the "Grayhill\3D50\startup\Images" folder which the VUI Builder© will also create if it doesn't exist.

When the application is started the initial dialog will appear as follows:



```
Display Model: Not Set
Current Device: Not Set
Project Name: Not Set
Object Count: 0
```

Figure 1: VUI Builder© Startup Screen

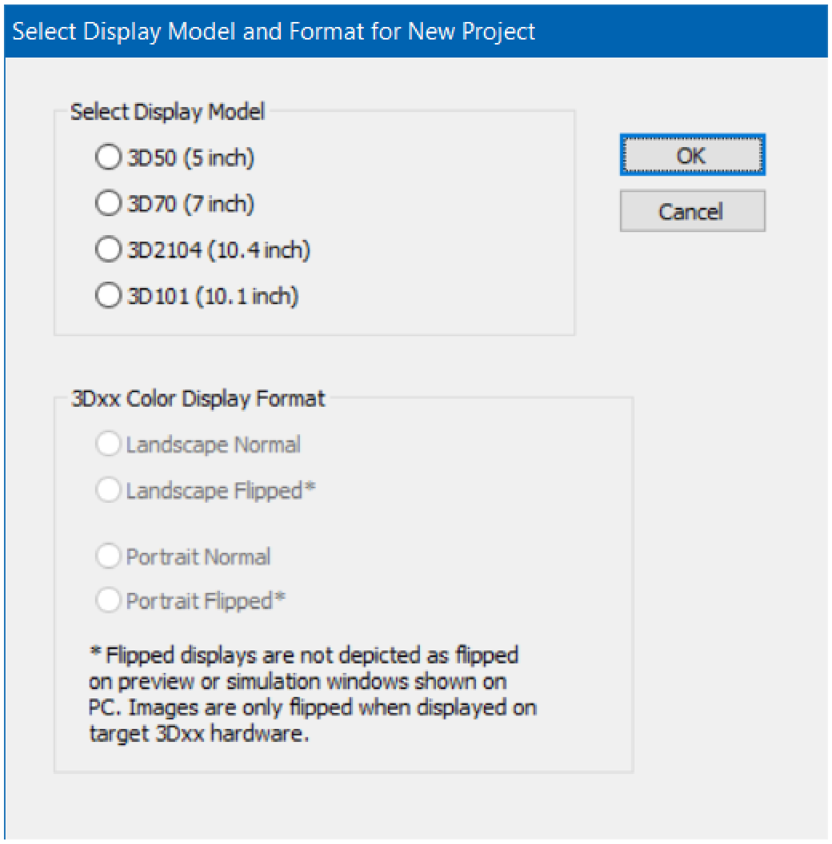
There are three options for getting started:

- Create a new project
- Open an existing project
- Upload existing objects from a device

The first two options can be performed independently of an actual 3Dxx Color Display being connected. The third option requires communication with a 3Dxx Color Display via CAN bus.

8. Creating a New Project

A new project is created by clicking on the "File→New" menu option. This will bring up this display format selection dialog box:


Select Display Model and Format for New Project

Select Display Model
- 3D50 (5 inch)
- 3D70 (7 inch)
- 3D2104 (10.4 inch)
- 3D101 (10.1 inch)

3Dxx Color Display Format
- Landscape Normal
- Landscape Flipped*
- Portrait Normal
- Portrait Flipped*

* Flipped displays are not depicted as flipped on preview or simulation windows shown on PC. Images are only flipped when displayed on target 3Dxx hardware.

Figure 2: 3Dxx Display Format Selection Dialog

Select the model of the target hardware. Note that this model selection can be changed later by using the menu item "Device→Configuration→Change Display Model". Note that certain I/O features are different on different models. Projects will run interchangeably on a different model, but features that do not exist will be ignored.

Next select the display presentation format. In Landscape mode the long dimension of the display is the horizontal dimension and the shorter dimension is presented vertically. In Portrait mode the longer dimension is presented vertically. The difference between "Normal" and "Flipped" is a 180 degree rotation of the display presentation. This option is provided to allow the user to orient the display to take best advantage of the slightly better viewing angle from one side of the display versus the other.

Display orientation (Landscape or Portrait) cannot be changed once it has been selected via this dialog box. That is because such a change would invalidate object coordinates since the width and height of the display are not equal. However, the "Normal" and "Flipped" configuration of a project can be changed at any time by clicking on the "Device→Configuration→Flip Target Display" menu item. If the "Flip" option is selected it will be noted on the title bar of the "3Dxx Display" preview window.

After picking a display format, a file selection dialog is presented so that a file name for the new project can be selected. By default the VUI Builder© starts in the “Grayhill\3D50\startup” folder mentioned above and starts with a default project name of “!!default.mdp”. A different file name may be selected. It is also possible to navigate to a different folder.

Under the folder selected for the project, a directory will be created named “Images”. The “Images” folder will initially be empty, but will be used to hold any image files added to the project.

The project file name selected will be displayed in the “Current Project File Name” box.

9. Adding a New Object

Clicking on the “Add” button on the main Object dialog window will cause the “Add Object” dialog to appear. This dialog allows you to choose from a number of different object types as shown here (note that this object list may be different depending on the target hardware model selected):

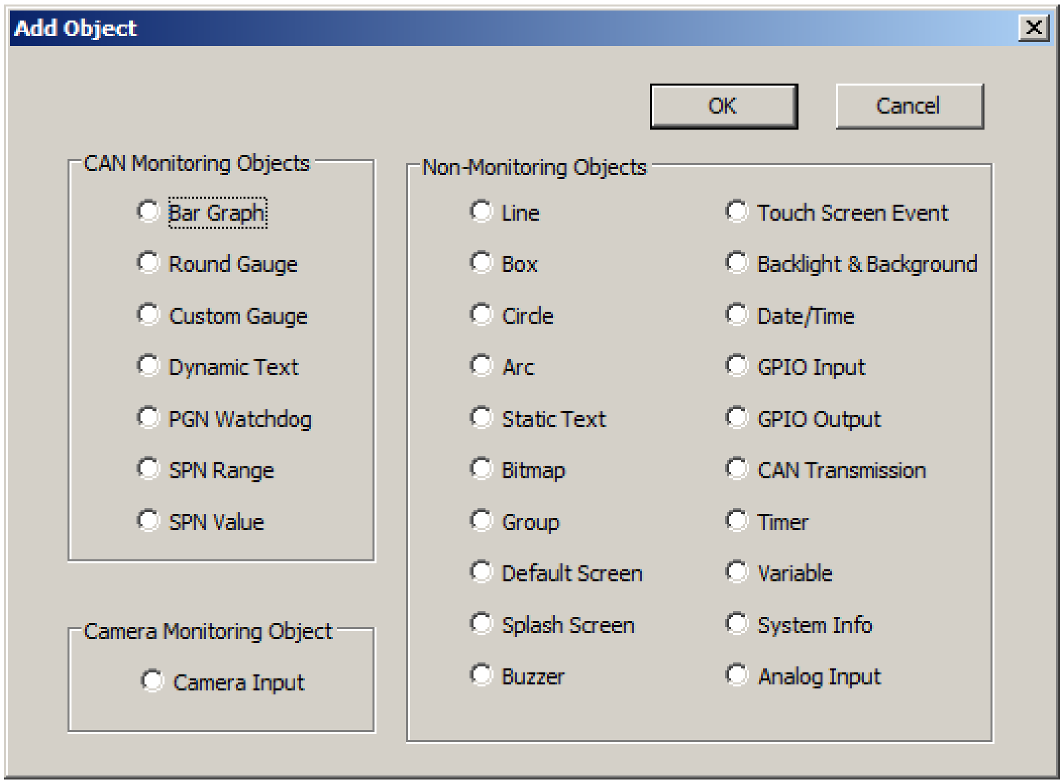


Figure 3: VUI Builder© Add Object Dialog

Select the radio button of the desired object type and click the “OK” button to create the new object.

An edit dialog box unique to the type of object selected will then be displayed that will allow the object to be customized by entering values for several parameters. The details for the various object parameters are described in Appendix A.

10. Editing Object Parameters

When an object is created, its edit dialog box is automatically launched. An object's edit dialog box can also be launched from the main in one of two ways:

- Select the object in the main window object list and then click the "Edit Object" button.
- Double click the object in the main window object list.

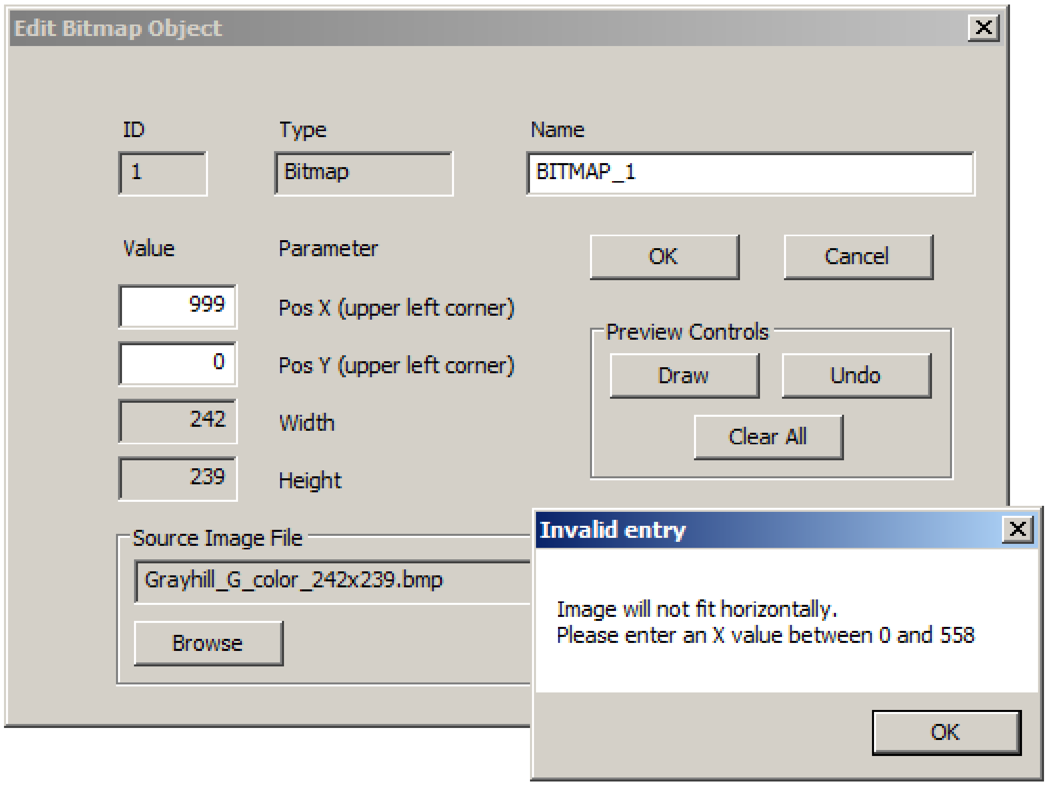
This edit dialog box allows the various parameters of the object to be changed. The object ID and Type parameters cannot be modified, but the object Name can always be modified. The only requirement placed on object names is that they must be unique.

Dialogs for object types that use CAN message data contain an “SPN lookup” feature to assist with the object definition. These object screens provide a list box 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. These edit control boxes are then made read-only and cannot be changed. If these items need to be different than a standard J1939 SPN value, then select the “Custom” item in the list box. This will unlock the PGN, Bit Start, and Bit Length parameters so that any legal values can be entered.

When done editing the object parameters click the “OK” button to save the changes and return to the main window. If the “Cancel” button is clicked, the changes will be discarded and control will return to the main window. If the edit dialog was launched because a new object was created, the object will not be created.

11. Object Parameter Validation

When in an edit object dialog box and the user clicks on the “Draw” or “OK” buttons the VUI Builder© performs a check on all parameters before proceeding. If any parameters are found to be in error, a message box is displayed. In addition to describing what the error is, this message box may also provide limits that must be observed. Sometimes these limits may change based on circumstances. For example, if setting up a Bitmap object, then the limits for the X and Y coordinates where it is to be displayed on the screen will vary depending on how big the image is. If the image width plus the X coordinate exceeds the width of the display, then an error box will appear that will indicate what the allowable values are for the X coordinate. Here is an example:


Edit Bitmap Object

ID: 1
Type: Bitmap
Name: BITMAP_1

Value	Parameter
999	Pos X (upper left corner)
0	Pos Y (upper left corner)
242	Width
239	Height

Source Image File: Grayhill_G_color_242x239.bmp

Invalid entry

Image will not fit horizontally.
Please enter an X value between 0 and 558

Figure 4: Example of Parameter Error Message

After clicking the “OK” button on such an error box, the offending parameter will be highlighted if possible. The originally requested “Draw” or “OK” function will not be executed until all parameter errors are fixed. Only one parameter error is reported at a time.

12. Previewing Objects with VUI Builder©

In order to understand how objects will appear on the 3Dxx Display, there is a 3Dxx Display preview window that opens when a project is opened or created. This preview window will be in the same orientation as defined by the current project (landscape or portrait). If the project specifies that the display is to be flipped on the 3Dxx target hardware, the preview window will still appear right-side up, but the phrase “flipped on target” will appear in the preview window title bar. The preview window title bar will also indicate where the X & Y’s 0, 0 coordinates are located and what the maximum X and Y coordinate values that are allowed for the current display configuration.

In the main window, when an object is selected from the object list, a preview of the object as currently configured is automatically drawn in the 3Dxx Display preview window.

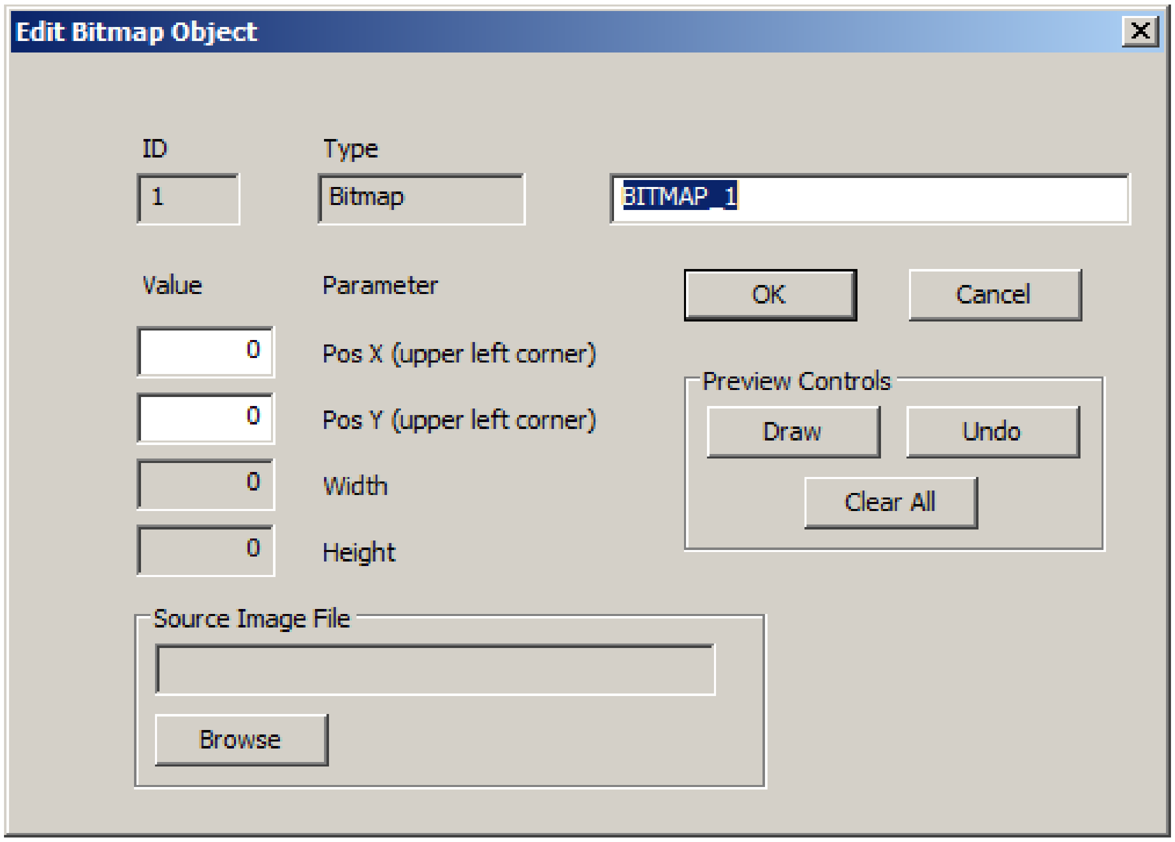
In preview-supported object edit dialog boxes, there are three preview control buttons:

- Draw – will draw a preview of the object as currently configured in the 3Dxx Display preview window
- Undo – will undo the last item drawn. This undo function works for approximately ten items
- Clear – clears the entire 3Dxx Display preview screen

Refer to Appendix A for more details about previews for various objects.

13. Adding a New Object – Bitmap Example

As an example, suppose the new object added is a Bitmap. The following is an example of how the edit dialog would initially appear before any changes are made:



ID: 1
Type: Bitmap
BITMAP_1

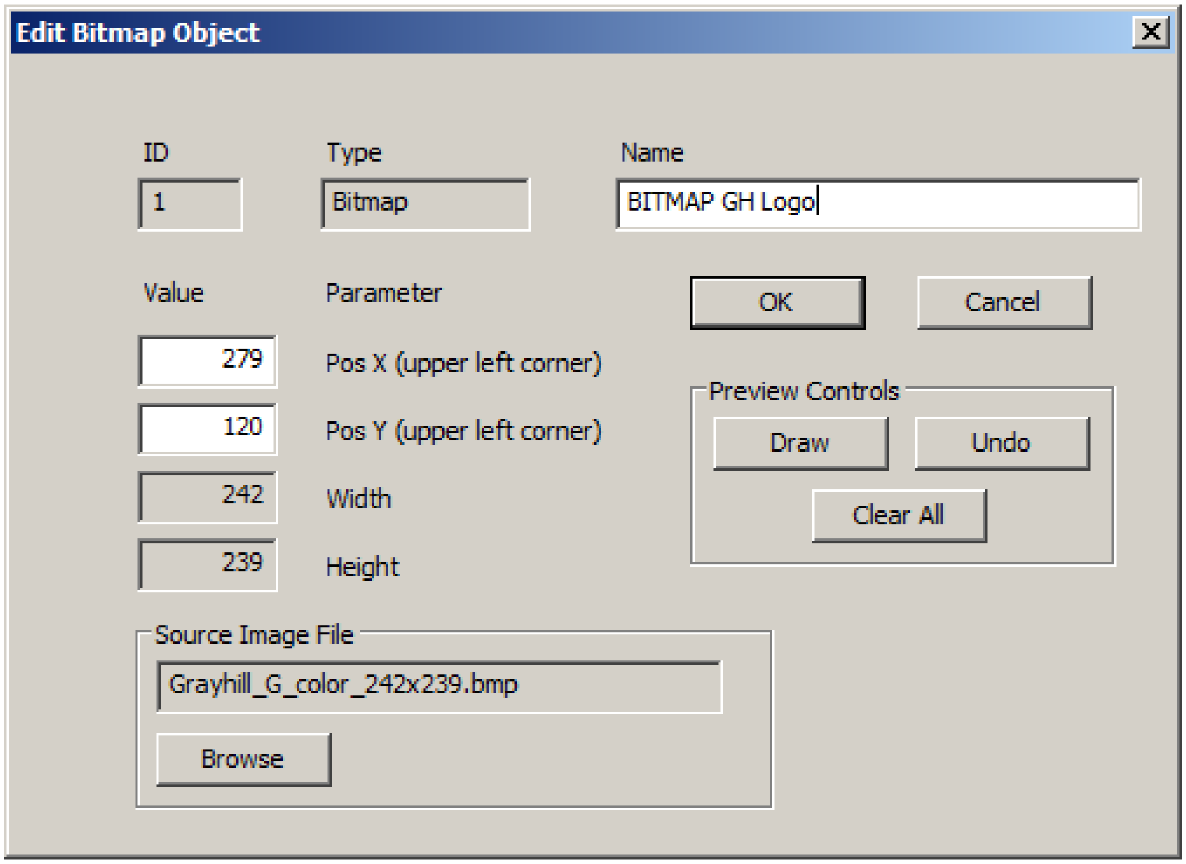
Value	Parameter
0	Pos X (upper left corner)
0	Pos Y (upper left corner)
0	Width
0	Height

Figure 5: Initial "Edit Bitmap Object" Dialog

The following changes are then made:

- Change the object name to "BITMAP GH Logo". It is important that the object names be descriptive because it will make development of the rest of the project easier. Note that spaces are allowed in this name field.
- Click "Browse" button and select an image file to be associated with this object. The VUI Builder© file selection dialog will start with the "Files of type:" selection set to "Bitmap files (*.bmp)"; however, it is possible to change this selection to "PNG" or "JPEG" files. The file selection dialog starts in the "Images" folder where the project file is located, but it is possible to navigate to any folder on the PC to select an image file. Once an image file is selected, the VUI Builder© will automatically copy the file to the "Images" folder where the project file is located (if using the default project file location this will be the folder "Grayhill\3D50\startup\Images"). Since image files are always copied to the project's "Images" folder, only the file name is needed and that is all that appears in the "Source Image File" box. Note that once a Bitmap file is selected, the "Width" and "Height" boxes will be automatically filled in based on information in the Bitmap file.
- Change "Pos X" and "Pos Y" to where the upper left hand corner of the Bitmap should appear on the 3Dxx Color Display when this object is displayed.

The “Edit Bitmap Object” dialog will now look like this:


Edit Bitmap Object

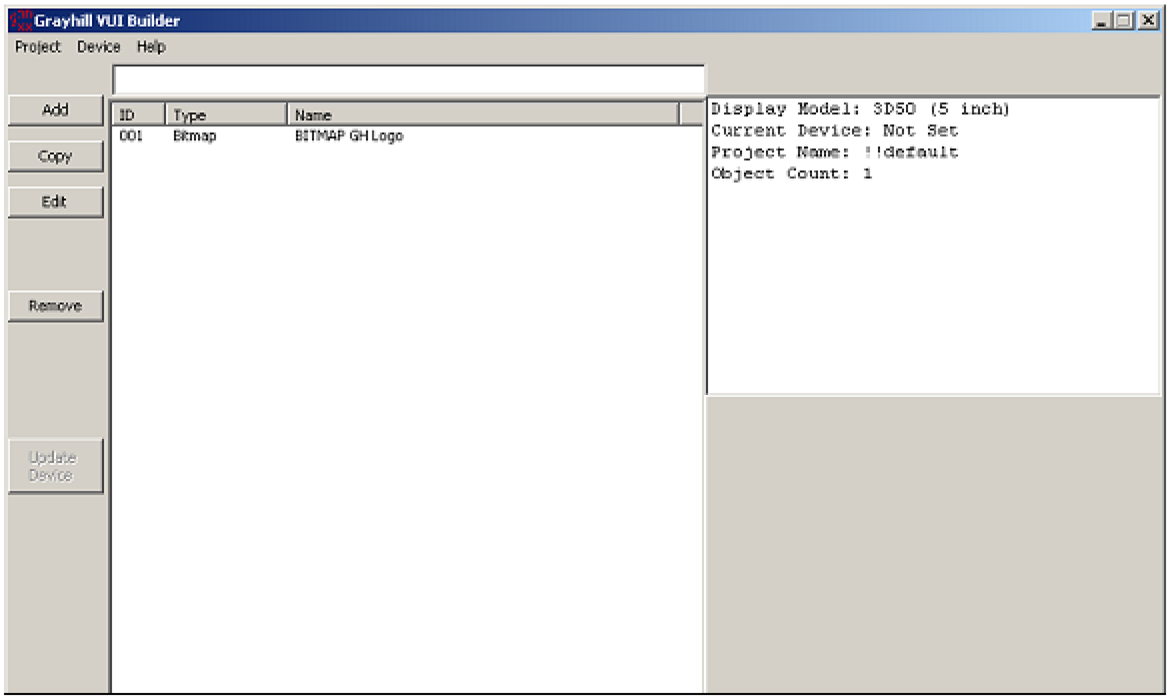
ID	Type	Name
1	Bitmap	BITMAP GH Logo

Value	Parameter
279	Pos X (upper left corner)
120	Pos Y (upper left corner)
242	Width
239	Height

Source Image File: Grayhill_G_color_242x239.bmp

Figure 6: Example of Filled in “Edit Bitmap Object” Dialog

After clicking the “OK” button in the “Edit Bitmap Object” dialog, focus is returned to the main window. In the main window, the newly created object is added to the object list and the object count is updated as shown here:



ID	Type	Name
001	Bitmap	BITMAP GH Logo

```
Display Model: 3D50 (5 inch)
Current Device: Not Set
Project Name: !!default
Object Count: 1
```

Figure 7: Main Window with New Bitmap Object Added

14. Saving a Project

A project can be saved by selecting “Project→Save” or “Project→Save As…” from the main menu. The “Project→Save” selection will save the project to the project file shown in the “Current Project File Name” box. The “Project→Save As…” selection will provide a file selection dialog so that the project can be saved to a different file.

15. Opening an Existing Project

To open an existing project, select “Project→Open…” from the main menu. If there is a project currently open in the VUI Builder© that has not be saved, the user is asked if he wants to save the project before proceeding. The VUI Builder© then clears all objects in its internal list and the objects and their associated name 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.

16. Using Variables

The VUI Builder© and 3Dxx Application Software support the use of up to twenty-six variables. These variables can be used to allow touch screen inputs to modify CAN messages. This can be used, for example, to allow an operator to set parameters like cabin temperature or fan speed by touching on-screen buttons or doing swipe gestures. This is done by linking the set, increment, or decrement variable object operations to touch screen inputs and using the variable values in CAN transmit objects. There is also an option that will display an on-screen keypad so that an operator can enter a value for a variable by tapping numbers.

On the 3D70 Model (seven inch display) there is also an option to have an analog input reading directed to a variable. See the section ***Analog Input Object (Model 3D70 only)*** for more information on this topic.

The value of a variable can be displayed on the 3Dxx Color Display by referencing it one of the CAN monitoring objects. Just select the special "Get Data from Variable" option on the SPN selection list. Then select the desired variable name from the pull-down list.

Variables can also be saved and recalled from FLASH memory so that their value can be restored after a power cycle. In order for the system to properly save variable values, the 3Dxx Color Display must be shut down by removing power from the “VIN Switched” signal on pin 3. Power on the “VIN Positive” signal on pin 1 must be maintained for at least three seconds longer.

Variables are named "A" thru "Z" and are limited to a range of 0 to 65535 (0xFFFF hexadecimal). Variables may also have a mask value. This is useful when the bits in the CAN transmit message that need to be controlled by a variable do not occupy a whole byte on a byte boundary.

For example, suppose a parameter for setting a radio frequency that varies from 531 to 1611 needs to be controlled. Let us further assume that this value occupies the top five bits of one CAN message byte and the last six bits of another CAN message byte. This is how to set up the variable object:

- Add variable object to declare Variable “A”
- Set “Minimum Value” to 531
- Set “Maximum Value” to 1611
- Set “Mask Value” (in hexadecimal) to 0xC007 (this leaves the most significant two bits and the least significant three bits fixed)
- Set the “Initial Value” (in hexadecimal) to 0xD0E3 (this sets the eleven bits of interest to 540, the most significant two bits to 11, and the least significant three bits to 011)

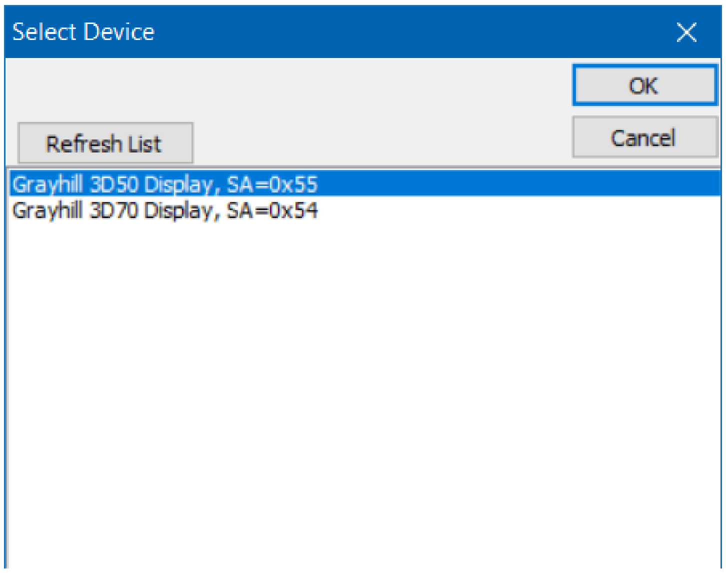
Incrementing and decrementing variable "A" will always leave the two most significant bits and the three least significant bits undisturbed. Also, when incrementing the variable "A", its value will not be allowed to exceed 1611. Attempts to increment past this value will be ignored. Similarly for the decrement case the value will not be allowed to go below 531.

When setting up the CAN Transmission Object, insert "AL" (for variable "A" low byte) and "AH" (for the high byte) in the appropriate bytes of the message. This CAN message will automatically be transmitted (using the "Send Count" and "Transmit Delay" parameters specified) each time the value of the variable “A” changes.

See section in Appendix A called *Variable Object* for more information on configuring Variable objects.

17. Communicating With a 3Dxx Color Display via CAN Bus

Before objects can be downloaded to or uploaded from a 3Dxx Color Display, the CAN communications link must be established. This can be done by selecting the “Device→Select…” item from the main menu. This is the dialog box that will appear if the VUI Builder© finds a 3Dxx Color Display connected to the USB PCAN bus adapter:



Grayhill 3D50 Display, SA=0x55
Grayhill 3D70 Display, SA=0x54

Figure 8: Device Select Dialog with 3D50 and 3D70 Connected Via CAN Bus

If more than one3Dxx Color Display device is found, select the desired unit from the list. If a single 3Dxx Color Display device is found, it will automatically be selected.

Click the "OK" button to accept the selection and close the dialog box. Control will then return to the main object dialog window and the selected 3Dxx Color Display device will now be displayed in the "Current Device:" box.

18. Uploading Objects from a 3Dxx Color Display

Object data can be uploaded to the VUI Builder© from the 3Dxx Color Display and then saved as a project. This is accomplished by selecting the "Device→Commands→Upload All Objects From 3Dxx Color Display" from the main menu.

If there is a project currently open in the VUI Builder© that has not be saved, the user is asked if he wants to save the project before proceeding. The VUI Builder© then clears all objects in its internal list and uploads all the object data from the 3Dxx Color Display. Each object will be added to the object list; however, the names displayed will be generic names based on the object type and object id number. This is because the object names are NOT saved in the 3Dxx Color Display – they are used only in the VUI Builder©. It is possible to edit the object names with VUI Builder© by editing each object. This operation is discussed in a subsequent section.

The object information can be saved as a project by selecting the "Project→Save As…" item in the main menu as described above in section **14.Saving a Project**.

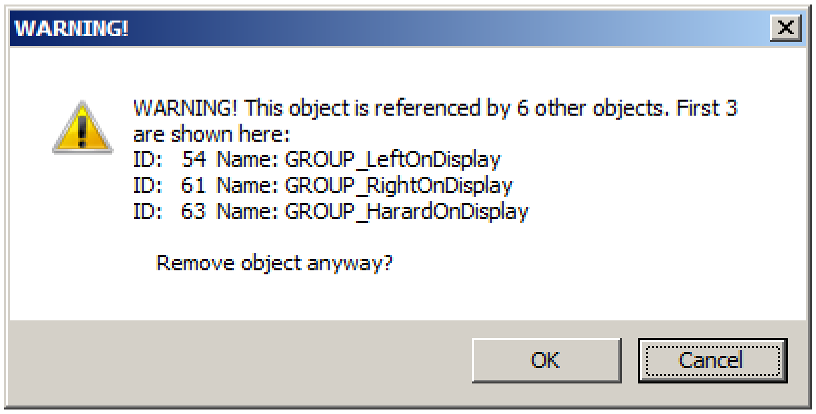
19. Removing an Object from a Project

Individual objects can be removed from a project; however, if the object declares a variable that is referenced elsewhere, then the object may not be removed. Removing objects that are referenced by other objects can cause undesirable results.

To see if an object is referenced elsewhere, select the object in the main list and then click on the "Show Where Object Used" button.

To remove a single object:

- Select the object in the object list.
- Click the “Remove Object” button.
- If the object being removed is referenced by other objects, removing it may cause unexpected behavior of the project. So in this case the following dialog will appear to help identify possible problems:


WARNING!

WARNING! This object is referenced by 6 other objects. First 3 are shown here:
ID: 54 Name: GROUP_LeftOnDisplay
ID: 61 Name: GROUP_RightOnDisplay
ID: 63 Name: GROUP_HarardOnDisplay

Remove object anyway?

Figure 9: Warning When Removing Object Referenced By Other Objects

- If the “OK” button is clicked or if the object being removed is not referenced by any other objects, then object will be removed from the list.

Note that the objects are not actually removed from the project file unless the “Project→Save” main menu item is selected.

20. Copying an Object

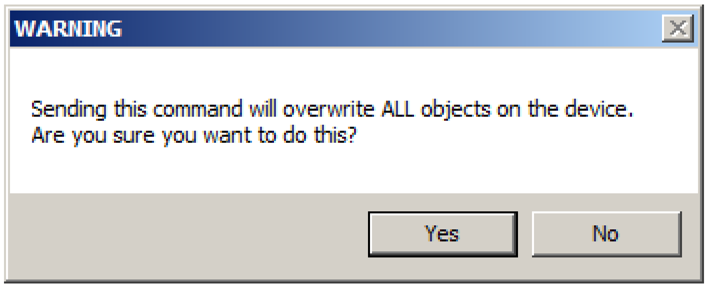
To copy an object:

- Select the object to be copied in the object list.
- 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. The name of the new object will be the same as the original, but the object’s ID number will be appended to the end of the new object’s name to ensure that the new object has a unique name.

21. Downloading Objects to a 3Dxx Color Display

To download all objects in a project to the selected 3Dxx Color Display device select the “Device→Commands→Download All Objects To 3Dxx Color Display”. This command will erase all of the objects in the RAM memory of the 3Dxx Color Display, so before proceeding the following warning dialog is displayed:


WARNING

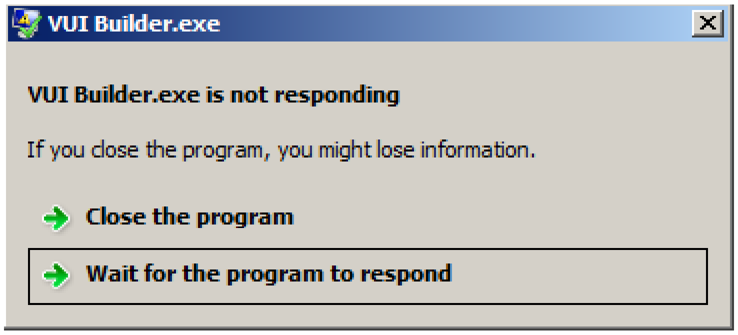
Sending this command will overwrite ALL objects on the device.
Are you sure you want to do this?

Figure 10: Warning Dialog before Doing Download

If the “Yes” option is clicked, then all the objects in the 3Dxx Color Display RAM memory are erased.

Then the objects in the VUI Builder© project are downloaded to the 3Dxx Color Display RAM. If any of the downloaded objects refer to a Bitmap file (can be BMP, PNG, or JPEG format), the Bitmap file will also be downloaded to the 3Dxx Color Display FLASH memory file system. Before downloading a Bitmap file, the 3Dxx Color Display code will see if it already has a correct version loaded by comparing the MD5 sum of the copy it has with the MD5 sum of the file to be downloaded. The file is only downloaded if it is either missing from the 3Dxx Color Display or if the MD5 sums do not match.

A progress dialog box shows which object number is being downloaded or which Font or Bitmap file is being downloaded. If focus is moved away from the VUI Builder© this dialog box may “freeze”. Attempts to click on VUI Builder© dialog boxes may bring up this window (example from Windows 7):


VUI Builder.exe

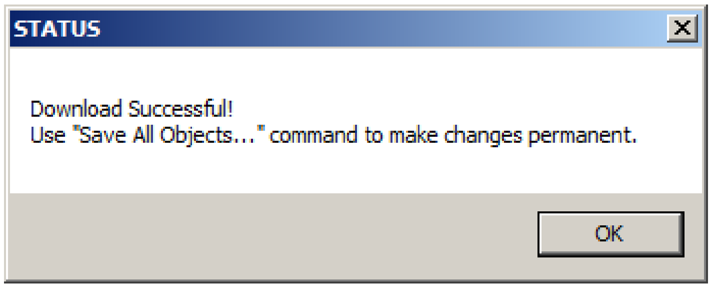
VUI Builder.exe is not responding

If you close the program, you might lose information.

Figure 11: 3DxxColorDisplayTool Not Responding Warning

Just click on the “Wait for the program to respond” choice and when the download is done, everything will return to normal. It may take about two minutes per megabyte of data being downloaded.

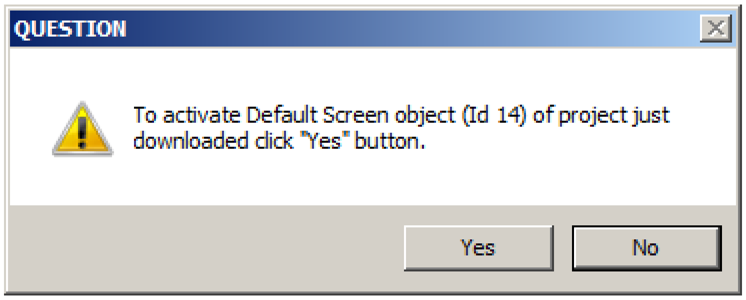
After the download completes successfully, this dialog box appears to remind the user that the objects were only saved to the 3Dxx Color Display RAM (however the Bitmap files, if any, were saved to the flash file system):


STATUS

Download Successful!
Use "Save All Objects..." command to make changes permanent.

Figure 12: Download Successful Dialog

If the project just downloaded contained a Default Screen object, then the following dialog box will appear to get the new project started if desired:



To activate Default Screen object (Id 14) of project just downloaded click "Yes" button.

Figure 13: Question about Activating Default Screen Object

The objects downloaded to RAM can be used and tested while in RAM, but if a "Device→Commands→Reset" operation is performed or if the 3Dxx Color Display power is cycled, then the downloaded objects will be lost. Use the command "Device→Commands→Save All Objects In 3Dxx To Its Default File" under the main menu to save the downloaded objects to the 3Dxx Color Display FLASH file system. Once this has been done, the downloaded objects will be loaded and used when the 3Dxx Color Display is reset or powered up.

After the download completes the VUI Builder© will search the project for a "Default Screen" object. If one is found, the user is asked if he wants to display this object; otherwise the 3Dxx screen will remain blank and no objects will be activated. This means that no touch input or CAN messages will be processed. Alternatively, the "Send Command" option can be used to display any object to get things started.

22. Displaying an Object Loaded on a 3Dxx Color Display

Objects in the 3Dxx Color Display can be displayed by doing the following:

- Select an object in the object list on the VUI Builder©. This assumes that the same set of objects is downloaded to the 3Dxx Color Display.
- Right click the selected object.
- The "Send Command" dialog will appear. (*)
- In the "Command" combo box, select "Display Object" option.
- Click the "Send" button.
- The "Response" box will display the result of the transaction (ACK or NAK).

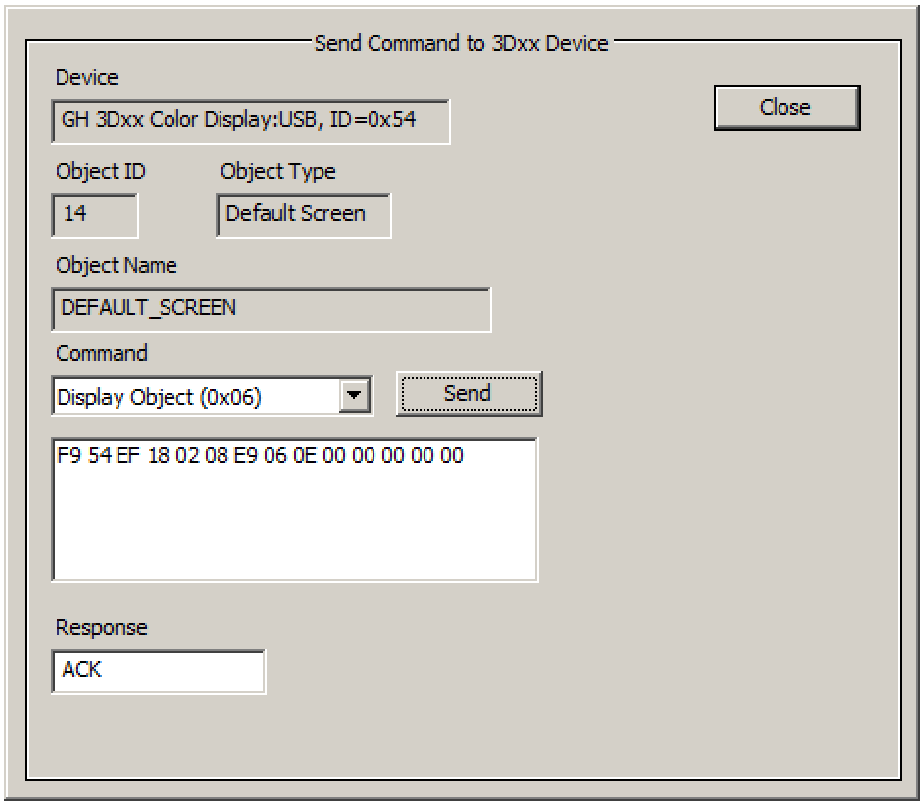
(*) The Send Command dialog for an object can also be launched by right-clicking the object in the object list.

The object will now be visible on the 3Dxx Color Display screen. Note that if the object displayed is a group object or refers to a group (such as a "Default Screen" object that is linked to a group object), then many objects may actually be "displayed".

Also remember that when an object is "displayed" it is also made active if appropriate. For example if the object defines touch screen input events, those touch screen input events will become active and if one of them is activated by touching the screen, then other objects will be "displayed" and possibly "killed".

The Bitmap Object has an additional variant of the "Display Object" command called "Display Object [new loc]". When selected, this command will cause dialog boxes for X and Y coordinates to appear. This then allows the Bitmap Object to be displayed at a different location. If the X or Y coordinates are set such that the image would go past the edge of the display, the offending coordinate will be automatically reduced to keep the entire image on the screen.

Here is what the “Send Command” dialog will look like after a successful Send Command operation:



Device: GH 3Dxx Color Display:USB, ID=0x54
Object ID: 14
Object Type: Default Screen
Object Name: DEFAULT_SCREEN
Command: Display Object (0x06)

```
F9 54 EF 18 02 08 E9 06 0E 00 00 00 00 00
```

Response: ACK

Figure 14: Example of Send Command Dialog

23. Activating an Object Not Loaded on a 3Dxx Color Display

A few object types can be activated or displayed on the 3Dxx Color Display without having to be stored as an object in the 3Dxx Color Display memory. These types are:

- Circle
- Box
- Line
- Arc
- Static Text
- Camera Input
- System Info
- Variable (can only be used with variable objects that set, increment, or decrement variables)

To activate or display such an object on the selected 3Dxx Color Display device:

- Select an object from the object list whose type is one of those listed above.
- Right click the selected object.
- The “Send Command” dialog will appear.
- In the “Command” combo box, select the appropriate command (“Draw Circle”, “Draw Box”, “Draw Line”, “Draw Arc”, “Draw Text”, “Activate Camera”, “Draw System Info”, or “Set Variable”).
- Click the “Send” button.
- The “Response” box will display the result of the transaction (ACK or NAK).

Either the object will appear on the 3Dxx Color Display screen or the variable's value will change as appropriate. Note that attempting to set a variable using the on-screen keypad is possible provided the location specified for the keypad will allow it to fit on the screen. If the keypad will not fit, an error (NAK) is returned and the operation is not performed.

24. Removing an Object from the 3Dxx Color Display Screen

To remove an object that is displayed on the screen of the selected 3Dxx Color Display device:

- Select an object in the object list.
- Right click on the selected object.
- The "Send Command" dialog will appear.
- In the "Command" combo box, select "Kill Object".
- Click the "Send" button.
- The "Response" box will display the result of the transaction (ACK or NAK).

When an object is "killed" its visible parts are erased by painting the area it occupied with the current background color. By default the background color is set to black, but it can be set to another color by using the "Backlight & Background" Object. Also, any items associated with the object will be deactivated. For example, if a touch screen input object is "killed", the touch screen events defined by the object will no longer be active and will not respond.

Note that the object is NOT erased from the memory on the 3Dxx Color Display.

The Bitmap Object has an additional variant of the "Kill Object" command called "Kill Object [new loc]". When selected, this command will cause dialog boxes for X and Y coordinates to appear. This allows the Bitmap Object area at a different location to be erased. If the X or Y coordinates are set such that the erased area would go past the edge of the display, the offending coordinate will be automatically reduced to keep the entire area on the screen.

25. Using VUI Builder© to Simulate CAN Messages

In order to aid in testing CAN Monitoring objects, the VUI Builder© Tool has an automatic CAN message simulator. This simulator will automatically generate a series of CAN messages for a selected object that will contain data values that range from the minimum value to the maximum value expected by the object. After generating messages that vary from the minimum value to maximum value, the simulator will then vary the data values from maximum value to minimum value. The VUI Builder© Tool sets the data increment size and a delay time between each message sent so that each sweep from one end of the data range to the other will take about 10 seconds. This cycle will repeat continuously until the "Stop Simulated Stimulus" command is issued. The following table shows which CAN Monitoring objects can utilize this feature and how the minimum and maximum values are calculated.

Table 1: CAN Monitoring Objects That Support Simulated CAN Messages

Object Type	Minimum Value	Maximum Value
Bar Graph	As specified by object	As specified by object
Round Gauge	As specified by object	As specified by object
Dynamic Text	0	Ones less than largest value that fits in bit length specified
Custom Gauge	Lowest Boundary value	Highest Boundary value
SPN Value	Lowest Bit Field value	Highest Bit Field value

Here is how to start simulated CAN messages for an object:

- Select an object in the object list.
- Right click the selected object.
- The “Send Command” dialog will appear.
- In the “Command” combo box, select “Display Object”.
- Click the “Send” button.
- In the “Command” combo box, select “Start Simulated Stimulus”.
- Click the “Send” button.
- The “Response” box will display the result of the transaction (ACK or NAK) and the object will be displayed and will show the results of the varying CAN message data.

If connected to the 3Dxx Display device via the CAN bus, the simulated messages will be sent over the CAN bus. If connected to the 3Dxx Simulator, the messages will be sent to 3Dxx Simulator display (see sections **29.3Dxx Simulator** Startup and **30.Communicating With the 3Dxx Simulator** for information on using the 3Dxx Simulator).

26. Clearing the 3Dxx Color Display Screen

To clear the screen on the selected 3Dxx Color Display device select the “Device→Commands→Clear 3Dxx Color Display Screen” from the main menu.

This command will clear the3Dxx Color Display and deactivate any active objects. For example, if any touch screen input events were active, they will no longer respond after this command is issued.

27. Erasing Objects Stored on the 3Dxx Color Display

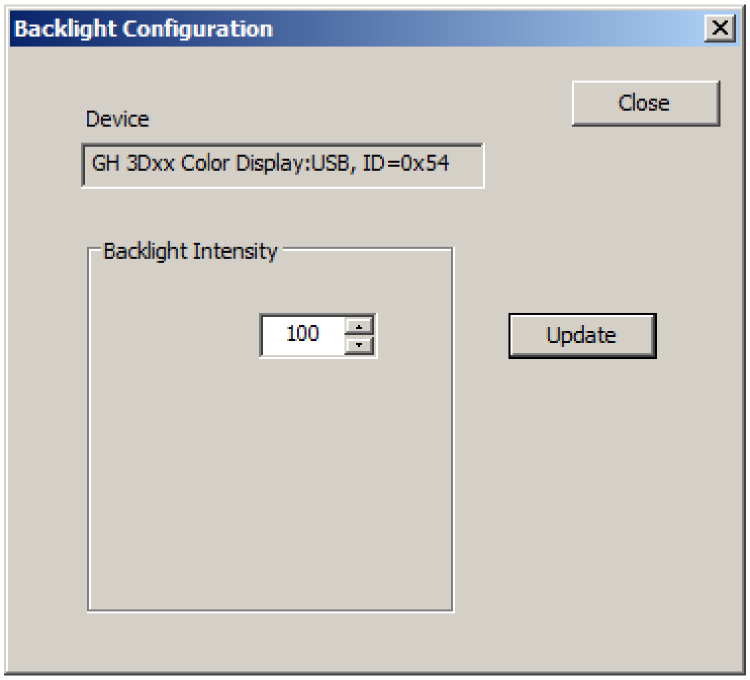
To erase the objects stored on the selected 3Dxx Color Display RAM, select the command “Device→Commands→Erase All Objects On 3Dxx Color Display” from the main menu.

A message box will be displayed asking to confirm that this operation is to be done. Click ‘Yes’ to perform the erase, or ‘No’ to abort.

This operation does not affect the default project file stored in the 3Dxx Color Display FLASH file system. That project will still be used when the power is cycled or when a “Reset” command is sent.

28. Setting the Backlight Intensity on the 3Dxx Color Display

The backlight intensity can be set on the selected 3Dxx Color Display device by selecting the "Device→Configuration→Set Backlight Level" command from the main menu. This will display this dialog:



Device: GH 3Dxx Color Display:USB, ID=0x54

Backlight Intensity: 100

Figure 15: Setting Backlight Level

Specify the desired Backlight Level using the spin control or by typing in a number. The allowable range is zero to 100. Click the "Update" button to send the new setting to the 3Dxx Color Display.

CAUTION! Setting backlight level to zero will turn off the backlight and the display will not be readable at all!

There is also an object that can be used to set the backlight level and that will be discussed in more detail in Appendix A.

29. 3Dxx Simulator Startup

Before starting the 3Dxx Simulator, make sure that the VUI Builder© is running. This will insure that the 3Dxx Simulator will be able to communicate with the VUI Builder©.

Double click on "3DxxColorDisplayApp.exe" file to run the 3Dxx Simulator. When the 3Dxx Simulator starts, it will look for a project file in the same folder in the user's HOMEPATH as discussed in section 7**. VUI Builder© Startup**. It will load the first project file it finds in that folder. Project files have an ".mdp" extension. The "first" project file is the one that appears first when searching alphabetically in the folder. Note that an exclamation point "!" is the lowest "alphabetic" character and so files that start with this character will get picked first. That is why the default project name is "!!default.mdp".

The 3Dxx Simulator opens two windows. One is the size and orientation of the simulated 3Dxx Color Display based on the project file loaded. The other is a small window that provides a few additional commands.

The smaller command window looks like this:

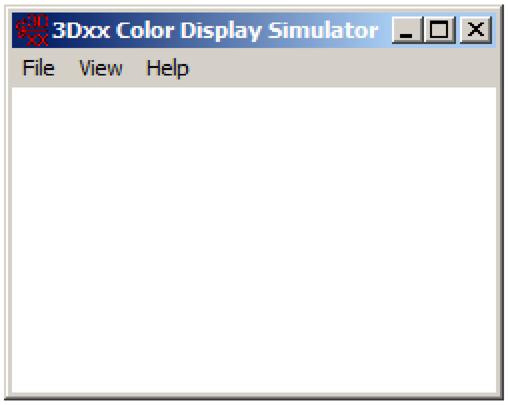


Figure 16: 3Dxx Simulator Command Window

Under the "View" main menu item there are three commands:

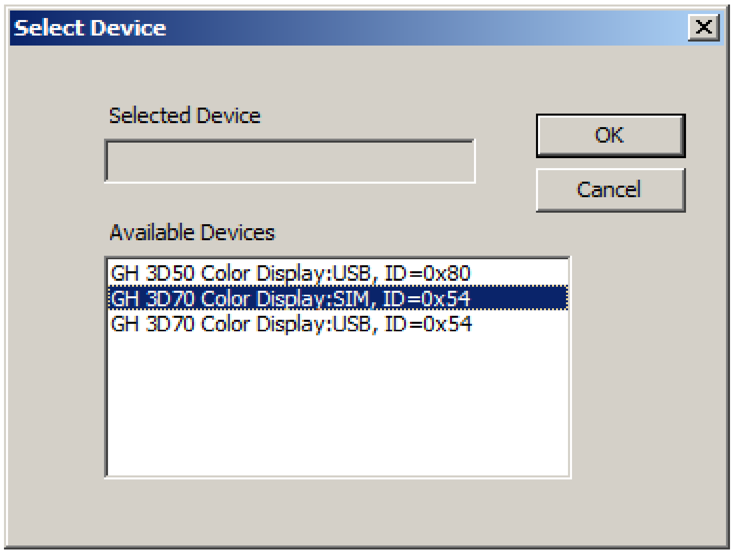
- Clear Screen – will clear the simulated 3Dxx Color Display window.
- Show Color Bars – shows the various shades of color available.
- Show Fonts – shows different fonts available for use in the 3Dxx Color Display objects.

30. Communicating With the 3Dxx Simulator

In order to use the VUI Builder© to modify and test projects on the 3Dxx Simulator, the communications link must be established. This requires doing the following:

- Be sure to start the VUI Builder© before starting the 3Dxx Simulator.
- In the VUI Builder© main window select the "Device→Select…" item from the main menu and select the device labeled "SIM".

This is a sample of "Device Select" dialog box showing a 3Dxx Color Display connected to the USB PCAN bus adapter and a 3Dxx Simulator that is running. The 3Dxx Simulator item is shown selected:



GH 3D50 Color Display:USB, ID=0x80
GH 3D70 Color Display:SIM, ID=0x54
GH 3D70 Color Display:USB, ID=0x54

Figure 17: Device Select for 3Dxx Simulator

Once this communication link is setup, all of the commands used to download, upload, display, draw, and kill objects will now apply to the 3Dxx Simulator instead. The 3Dxx Simulator will also respond to Start and Stop Simulator Stimulus commands sent by the VUI Builder© Tool (see section **25. Using VUI Builder© to Simulate CAN Messages** for more details on this feature). Touch screen inputs can also be tested by clicking and swiping on the simulated display with the Windows mouse.

31. Miscellaneous Commands

Under the "Device" menu item there is a sub-section called "Configuration". Under this sub-section are the following commands:

- Flip Target Display: (previously discussed under the ***Creating a New Project*** section)
- Change Display Model…: (previously discussed under the ***Creating a New Project*** section)
- Calibrate Touch Screen:
  This feature should never be necessary. If selected, the user must touch the screen at five designated points and this information will be used to create a new calibration file for the touch screen. Be sure to fully complete the calibration sequence in order to prevent possible loss of the calibration file. WARNING! If there is a power loss during the calibration sequence, the touch screen may fail to operate. Just run the calibration command again when the power is restored.
- Set Backlight Level:
  If a display device is currently selected, this command can be used to set the backlight setting of the display immediately.

Under the "Device→Commands" menu is a sub-menu called "Draw Test Images". These commands will draw various color bar patterns on the selected display device.

32. Project Operations

Under the "Project" menu item there are sub-menu items for the following operations:

- Create USB Install Stick

  This operation takes the current project and creates a folder containing all the files necessary for installing the project on the display from a USB flash drive. The created folder is called '3D50_update', and it contains the project file, image files, and an install script. The location for saving the folder is specified by the user.

  To use the install package, copy the complete folder to the root of a USB flash drive. With the 3Dxx display running, insert the USB flash drive into the display. The "app update" mechanism of the display will automatically execute the app_update.sh script on the USB flash, which will install the project.

- Create Project Binary

  This operation combines all the necessary files for the current project and saves them as a single 'binary' file. The location for saving the binary file is specified by the user.

  The project binary file that is produced can be downloaded over the CAN bus to the 3Dxx display using a different Grayhill PC tool called the "3DDownloadTool". The 3DDownloadTool can be useful for field personnel who need a simple tool for updating the project software on 3Dxx displays, but do not need the full design functionality of the VUI Builder tool.

  The 3DDownloadTool software is available for download on the Grayhill VUI Builder web page:
  https://www.grayhill.com/vui-builder

Appendix A. 3Dxx Color Display Object Details

Appendix A.1. 3Dxx Color Display Object Overview

The intended use of the 3Dxx Color Display 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 3Dxx Color Display is configured with CAN Monitoring Objects. These objects monitor the J1939 bus and parse the relevant data using the following criteria:

- **CAN Bus Selection** – The 3Dxx Color Display can monitor messages on up to three CAN buses depending on the model.
- **Msg Type** – Extended (29-bit) or Standard (11-bit) or Extended/Standard
- **PGN** – The PGN (16 bits) and the Data Page (one bit) of the SPN parameter to be displayed. (used with Extended message type)
- **SA** – The source address of the sending device. (used with Extended message type)
- **ID** – The 11-bit message ID (used with Standard message type)
- **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 3Dxx Color Display the ability to replace several existing standard gauges or an entire gauge cluster.

A 3Dxx Color Display screen is composed of one or more objects. An object can be a CAN Monitoring Object, as described above, a Camera Monitoring Object, or a Non-Monitoring Object, which includes bitmaps, lines, boxes, text, etc. Some CAN 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 then the Group Object is used. When displayed (instantiated) or killed, all objects within the group are then displayed 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 displayed or killed according to the monitored parameter value.

The Touch Screen Object gives more power to the 3Dxx Color Display by giving the end user the ability to select various options by tapping or swiping the display. Objects used in this scenario would most likely be group objects that would include Monitoring Objects along with text, bitmaps, and other touch screen objects 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:

Table 2: Parameters Common to All Objects

Parameter	Description
ID #	Object ID number assigned by VUI Builder© Range: 1 – 65534 ***- Read Only -***

Param eter	Description
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. These object names can be changed at any time. These names are NOT used by the 3Dxx Color Display itself; they are 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, Camera Monitoring, or Non-Monitoring and are described in the following sections.

Appendix A.2. CAN Monitoring Objects

Most CAN Monitoring objects contain these common CAN message parameters:

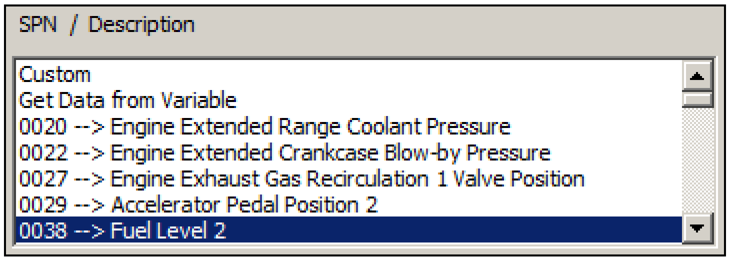
Table 3: Common CAN Monitoring Parameters

Parameter	Description
CAN Bus Select	Selects which CAN bus to monitor. Some 3Dxx Color Displays may have up to three CAN buses.
Msg Type	Extended (29-bit) or Standard (11-bit) or Extended/Standard* **See Notes below**
PGN	Parameter Group Number and Data Page bit Range: 0x0 – 0x1FFFF ***[Note: Applies to Msg Type = Extended]***
ID	11-bit message ID Range: 0x0 – 0x7FF ***[Note: Applies to Msg Type = Standard]***
Source Address	Source address of sending device Range: 0x0 – 0xFF ***[Note: Applies to Msg Type = Extended]***
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 defined at Bit Start. Set to zero (0) if no control field exists. Range: 1 – 64 ***[Note: Applies to Msg Type = Extended]***
Control Data Bit Length	Size of the control bit field. Set to zero (0) if no control field exists. Range: 1 – 32 ***[Note: Applies to Msg Type = Extended]***
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 ***[Note: Applies to Msg Type = Extended]***

The above parameters of PGN, Source Address, Bit Start, and Bit Length can be set automatically by picking a CAN message from a list of standard J1939 messages. The VUI Builder© has over 3000 standard J1939 messages in its library. The user may also select the “Custom” item from this list and is then free to set these parameters to any valid values.

There is also a “Get Data from Variable” option that allows the user to specify a variable to be monitored instead of a CAN message. This is useful for providing the operator with feedback about what value he has selected for a parameter that is controlled by a variable.

This is a sample of the selection list for a few standard J1939 messages and it also shows the “Custom” and “Get Data from Variable” options:


SPN / Description

Custom
Get Data from Variable
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

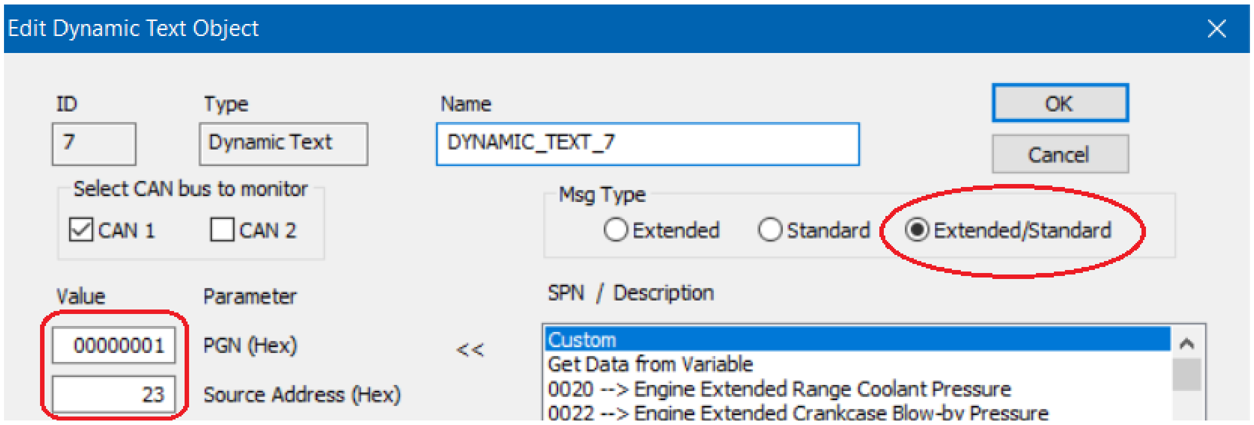
Figure 18: Sample Selection List for Standard J1939 Messages

Many of the CAN Monitoring objects specify data values (i.e. minimum or maximum values). The limits for these data values are based on the number of data bits specified for the data value being monitored; however, if the value is coming from a variable, then the limits are based on the minimum and maximum values selected when the variable is declared.

*** NOTE about CAN message formats for CAN Monitoring objects:**

For Rev K and earlier versions of VUI Builder, all CAN monitor messages were interpreted as Extended (29-bit Id) format. Thus, for CAN messages with an Id <= 0x7FF, no distinction was made between messages of Extended or Standard (11-bit Id) format. To support backward compatibility with existing projects, this behavior is maintained using the Extended/Standard message type. Opening an existing project created with Rev K or earlier versions of VUI Builder will have Extended/Standard selected for the CAN message type.

For example:


Edit Dynamic Text Object

ID: 7
Type: Dynamic Text
Name: DYNAMIC_TEXT_7

Select CAN bus to monitor: [✓ CAN 1] CAN 2

Msg Type: Extended Standard [✓ Extended/Standard]

Value	Parameter
00000001	PGN (Hex)
23	Source Address (Hex)

SPN / Description

Custom
Get Data from Variable
0020 --> Engine Extended Range Coolant Pressure
0022 --> Engine Extended Crankcase Blow-by Pressure

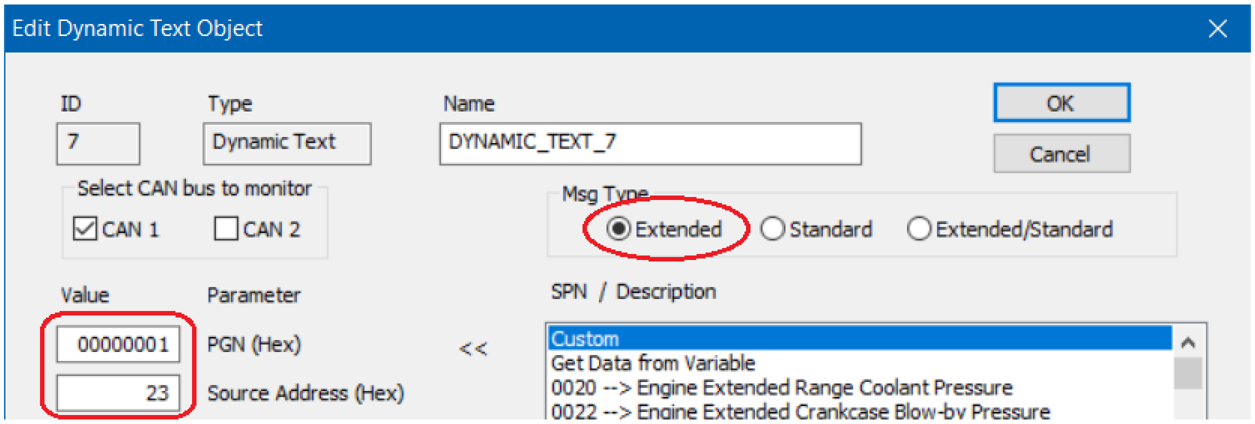
This object would respond to these CAN messages:

0x00000123 (Extended)
0x123 (Standard)

For CAN messages with an Id > 0x7FF, if Extended/Standard type is selected, then only Extended messages apply; Standard messages are not valid.

Extended CAN message

CAN Monitoring objects can be defined to monitor only Extended CAN messages, regardless of whether the CAN Id is > 0x7FF or <= 0x7FF. For example:


Edit Dynamic Text Object

ID: 7
Type: Dynamic Text
Name: DYNAMIC_TEXT_7

Select CAN bus to monitor: [✓ CAN 1] CAN 2

Msg Type: [✓ Extended] Standard Extended/Standard

Value	Parameter
00000001	PGN (Hex)
23	Source Address (Hex)

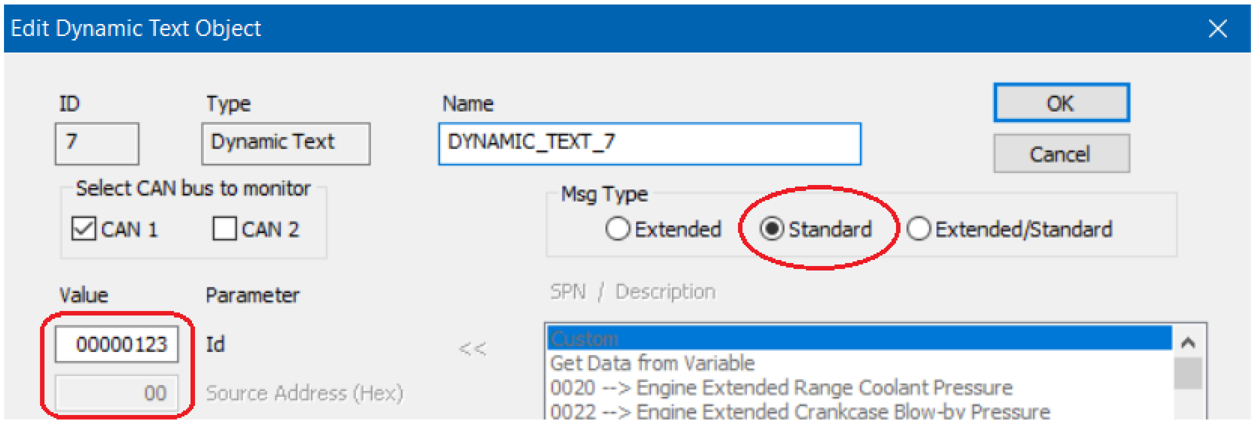
SPN / Description
Custom
Get Data from Variable
0020 --> Engine Extended Range Coolant Pressure
0022 --> Engine Extended Crankcase Blow-by Pressure

This object will respond to this CAN message:
0x00000123 (Extended)

But will NOT respond to this CAN message:
0x123 (Standard)

Standard CAN message

CAN Monitoring objects can be defined to monitor only Standard CAN messages. For Standard messages, only CAN Id's <= 0x7FF are allowed. For example:


Edit Dynamic Text Object

ID: 7
Type: Dynamic Text
Name: DYNAMIC_TEXT_7

Select CAN bus to monitor: [✓ CAN 1] CAN 2

Msg Type: Extended [✓ Standard] Extended/Standard

Value	Parameter
00000123	Id
00	Source Address (Hex)

SPN / Description
Custom
Get Data from Variable
0020 --> Engine Extended Range Coolant Pressure
0022 --> Engine Extended Crankcase Blow-by Pressure

This object will respond to this CAN message:

0x123 (Standard)

But will NOT respond to this CAN message:
0x00000123 (Extended)

*** NOTE about CAN message formats for CAN Transmission object:**

For CAN Transmission objects, the Tx Msg type and Ack Msg type are selected independently.

Transmit Message

For Tx Msg type, the options are

- Extended (29 bit)
- Standard (11 bit)

With existing Rev K and earlier VUI Builder software, the Transmit message was always Extended, regardless of the CAN Id. Therefore, opening an existing project created with Rev K or earlier versions of VUI Builder will have Extended selected for the CAN message type. There is no Extended/Standard option.

Ack Message

For Ack Msg type, the options are

- Extended (29 bit)
- Standard (11 bit)
- Extended/Standard*

For Rev K and earlier versions of VUI Builder, all Ack messages for CAN Tx objects were interpreted as Extended (29-bit Id) format.

Thus, for CAN Ack messages defined with an Id <= 0x7FF, no distinction was made between Extended and Standard (11-bit Id) format. To support backward compatibility with existing projects, this behavior is maintained using the Extended/Standard message type. Opening an existing project created with Rev K or earlier versions of VUI Builder will have Extended/Standard selected for the CAN message type.

For details about the behavior of each option, see the above "*NOTE about CAN message formats for CAN Monitoring objects*".

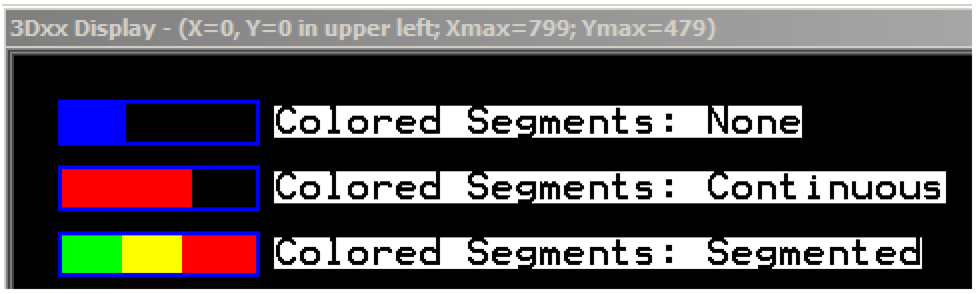
Bar Graph Object

The Bar Graph Object monitors CAN traffic for an SPN value and adjusts the bar graph display accordingly. The position of the upper left corner of the bar is specified by the x and y coordinates. 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. The gauge can be configured for a horizontal or vertical orientation. Horizontal gauges move left to right and vertical gauges grow from bottom to top. The border width (in pixels) and color of the bar graph can be selected. There are also two options to have the bar graph change color based on the value received.

Table 4: Bar Graph Object Parameters

Parameter	Description
Common CAN Message Parameters plus…	
Min Value	Value corresponding to bar graph value of 0% Range: Based on number of data bits specified
Max Value	Value corresponding to bar graph value of 100% Range: Based on number of data bits specified
Position X1	X-coordinate of upper left corner of bar
Position Y1	Y-coordinate of upper left corner of bar
Length	Length (in pixels) of bar including width of border
Width	Width (in pixels) of bar including width of border
Orientation	Horizontal – Bar graph moves left to right Vertical – Bar graph moves from bottom to top
Color	Color of the bar graph border. It is also the color of the bar graph if not using colored segments.
Border Width	Width of border(border stays inside width and length specified above)
Colored Segments	• None – Bar graph is a solid color as defined above and the border is this same color as well. Selecting one of the following two options will show a “Segment Specification” box on the Bar Graph dialog. This box allows up to eight segments to be defined based on either data value or percentage of data range. The data value or percentage indicates the top value for a segment. A different color can be selected for each segment. • Continuous – Bar graph is solid color but color depends on value being graphed. • Segmented – as bar graph grows each segment of it will be the color defined for that segment.
Segment Specifications (up to eight can be defined in ascending order) - Only used for Continuous and Segmented modes -	
Top Value	Top value for a segment expressed as data value or percentage of whole data range
Color	Color to use for whole bar graph if data is at or below segment’s data range and using Continuous mode or color of segment of bar graph if using Segmented mode

Below is a sample of each type of bar graph. Note that for the “Colored Segments: None” case the bar graph is the same color as the border.


3Dxx Display - (X=0, Y=0 in upper left; Xmax=799; Ymax=479)

Colored Segments: None

Colored Segments: Continuous

Colored Segments: Segmented

Figure 19: Bar Graph Samples

Round Gauge Object

The Round Gauge Object monitors CAN traffic for an SPN value and adjusts a round gauge display accordingly. The round gauge data can be scaled by specifying the minimum and maximum data values, corresponding to round gauge pointer (or in some formats the colored background) being at start angle and end angle, respectively. The position of the center of the gauge is specified by the x and y coordinates.

The background can be a solid color filling the whole background, a solid color filling the whole background that changes color based on data, a solid color that changes size and color based on data, multi-colored segments with size and colors based on data, or a fixed image.

The pointer can be omitted (only if using a background that changes size with the data), or it can be a needle or arrow pointer. The needle or arrow pointers can be a fixed color or the color of the pointer can vary based on the data being displayed. The pointer can also be a fixed image that the 3Dxx Color Display software will rotate based on the data being displayed.

Table 5: Round Gauge Object Parameters (Common to All Formats)

Parameter	Description
Common CAN Message Parameters plus…	
Min Value	Value corresponding to pointer being at start angle Range: Based on number of data bits specified
Max Value	Value corresponding to pointer being at end angle Range: Based on number of data bits specified
Center X	X-coordinate of gauge center
Center Y	Y-coordinate of gauge center
Start Angle	This angle corresponds to the minimum data value
End Angle	This angle corresponds to the maximum data value
Background Format	Round Gauge background format selections available are: • All One Fixed Color • All One Color Based on Data & Segments • One Color and Size Based on Data & Segments • Multi-Color Segments and Size Based on Data • Drawn From Image
Pointer Format	Round Gauge pointer format selections available are: • No Pointer • Needle Pointer: Fixed Color • Needle Pointer: Color Based on Data • Arrow Pointer: Fixed Color • Arrow Pointer: Color Based on Data • Drawn From Image

Other parameters of this Round Gauge depend on what background and pointer formats are chosen. Some background and pointer format combinations are not allowed because they would result in a gauge that would not produce clearly visible changes when the input data was changed. If such a selection is made, an error dialog appears and one of the formats must be changed. The following table summarizes which background and pointer formats are allowed. For simplicity, background and pointer formats will be discussed in four separate groups as indicated in this table.

Table 6: Round Gauge Allowable Background and Pointer Format Combinations

Group 1	**Group 2**		**Group 3**		**Group 4**
	Background Format				
Pointer Format	All One Fixed Color	All One Color Based on Data & Segments	One Color and Size Based on Data & Segments	Multi-Color Segments and Size Based on Data	Drawn From Image
No Pointer			OK	OK	
Needle Pointer: Fixed Color	OK	OK	OK	OK	OK
Needle Pointer: Color Based on Data	OK				OK
Arrow Pointer: Fixed Color	OK	OK	OK	OK	OK
Arrow Pointer: Color Based on Data	OK				OK
Drawn From Image	OK	OK	OK	OK	OK

Group 1 –Pointer Formats
The “No Pointer” format is only allowed if the background format is being drawn proportional to the data being graphed. For all other background formats a pointer is needed to be able visualize the data being graphed.

When using an arrow or needle format pointer there are two choices for the pointer’s color: it can be a solid color as selected in the dialog box, or it can change color as defined in one of up to eight data range segments.The pointer length and width can also be specified as shown in table below.

Table 7: Round Gauge Object Parameters (Arrow or Needle Pointer)

Parameter	Description
Radius / Pointer Length	Radius (or pointer length if using background image) Minimum value: 12
Border & Pointer Width	Width of pointer (and of border if not using background image) Range: 1 to 32
Border Color	Color of gauge border (if not using background image)
Pointer Offset	Offset pointer’s starting point away from center of gauge. This is only used with a background image and allows the background image to have a center feature that is not disturbed when the pointer is drawn.
Pointer Color	Used to set pointer color when using “… Fixed Color” pointer format

Parameter	Description
Segment Specifications (up to eight can be defined in ascending order) - Used for "… Color Based on Data" pointer format -	
Top Value	Top value for a segment expressed as data value or percentage of whole data range
Color	Color to use for pointer if data is at or below segment's data range

When using the "Drawn From Image" pointer format, an image file must be provided by clicking the "Pick Image" button. This image file is handled the same way an image file for the Bitmap Object is handled. (See **Section 13Adding a New Object – Bitmap Example** for more details.) When this image file is first specified the VUI Builder© software will provide an initial estimate for the coordinates of the pointer image pivot point and tip. These coordinates are expressed as an offset from the upper left-hand corner of the image. They may be adjusted if necessary. The 3Dxx Applications software automatically removes the background from the pointer image when it rotates it so that the background is not distorted. The 3Dxx Application software determines the background color(s) used in the pointer image by using this algorithm:

1. Examine each pixel around edge of image going two pixels deep.
2. For each pixel examined, remember the color of each unique color there are up to a maximum of ten.
3. Every pixel in the entire pointer image that matches one of these colors is considered a background color and is not copied on top of the background when the pointer image is rotated.
4. If there are more than ten unique colors found, it is considered an error and the pointer image will not be drawn.

Due to this algorithm, it is important that the background color for the pointer image covers all edge pixels going two pixels deep and does not contain more than ten unique colors.

Table 8: Round Gauge Object Parameters (Pointer Drawn from Image)

Parameter	Description
Pick Image	Button to allow selection of image file from file system
Offset to pointer pivot - Image X - Image Y	Offset from the upper left-hand corner of the image to the pointer image's pivot point
Offset to pointer tip - Image X - Image Y	Offset from the upper left-hand corner of the image to the pointer image's tip

Group 2 – Background Format is "All One Fixed Color" or "All One Color Based on Data & Segments"
In these background formats the whole background from start angle to end angle is filled in with one color. The color is either fixed or set by the color defined in one of up to eight data range segments. The size of the gauge is set by the radius. The border width and the border color can be set as desired.

Table 9: Round Gauge Object Parameters (Background Completely Filled)

Parameter	Description
Radius	Radius of round gauge Minimum value: 12
Border &Pointer Width	Width (in pixels) of gauge border (and pointer if applicable) Range: 1 to 32
Border Color	Color of gauge border
Background Color	Used to set background color when using "All One Fixed Color" format
Segment Specifications (up to eight can be defined in ascending order) - Used for "All One Color Based on Data & Segments" format -	

Parameter	Description
Top Value	Top value for a segment expressed as data value or percentage of whole data range
Color	Color to use for whole background if data is at or below segment's data range

Group 2 – Background Format is "One Color and Size Based on Data & Segments" or "Multi-Color Segments and Size Based on Data"
In these background formats the only part of the background is drawn and it is proportional to the data value being displayed. The color is set by the color defined in one of up to eight data range segments. If using "One Color…" format, then the background is drawn using the color from the Segment Specifications corresponding to the data being drawn. If using "Multi-Color…" format, then as the background is drawn the color from the Segment Specifications is used for each segment but drawing stops when background arc is proportional to the data being drawn. The size of the gauge is set by the radius. The border width and the border color can be set as desired. Note that the whole border from start angle to end angle is always drawn.

Table 10: Round Gauge Object Parameters (Background Proportionally Filled)

Parameter	Description
Radius	Radius of round gauge Minimum value: 12
Border & Pointer Width	Width (in pixels) of gauge border (and pointer if applicable) Range: 1 to 32
Border Color	Color of gauge border
Segment Specifications (up to eight can be defined in ascending order)	
Top Value	Top value for a segment expressed as data value or percentage of whole data range
Color	Color to use if data is at or below segment's data range

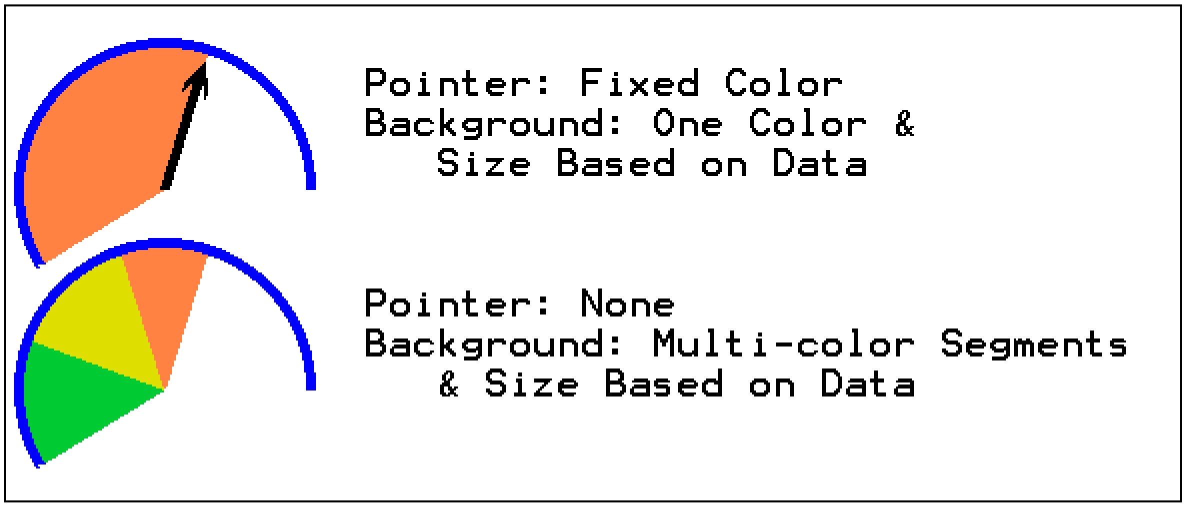

Pointer: Fixed Color
Background: One Color & Size Based on Data
Pointer: None
Background: Multi-color Segments & Size Based on Data

Figure 20: Samples of Round Gauges with Proportional Backgrounds

Group 4 – Background Format is "Drawn From Image"
If using an image for the background, then the size of the gauge is controlled by the size of the image. When using this background format, an image file must be provided by clicking the "Pick Image" button. This image file is handled the same way an image file for the Bitmap Object is handled. (See **Section 13Adding a New Object – Bitmap Example** for more details.) When this image file is first specified the VUI Builder© software will provide an initial estimate for the coordinates of the background image pivot point (it picks the exact center of the image).

These coordinates are expressed as an offset from the upper left-hand corner of the image. They may be adjusted if necessary.

Table 11: Round Gauge Object Parameters (Background Image)

Parameter	Description
Pick Image	Button to allow selection of image file from file system
Offset to pointer pivot - Image X - Image Y	Offset from the upper left-hand corner of the image to the pivot point to use for the pointer

Additional Notes for Round Gauge
The edit dialog for the Round Gauge provides some additional preview drawing controls to aid is setting up the gauge. There are “Draw Min” and a “Draw Max” buttons that will draw the configured gauge as it would appear when graphing the “Min Data Value” and “Max Data value”, respectively. This can aid in setting the Start Angle and End Angle parameters when using an image for the background. The “Draw Moving” button will preview the gauge by varying data values from Min Data Value to Max Data Value and then back to Min Data Value. This action continues until the “OK” button is clicked in the pop-up dialog.

Better previews of the configured gauge can be obtained using the VUI Builder© simulator feature with the 3Dxx Simulator program.

Custom Gauge Object

The Custom Gauge Object provides the ability to create custom gauges that display objects based on a value received over the CAN bus. This object uses an array of boundary segments composed of a compare value, an active object, and an inactive object. The active and inactive objects are displayed and killed depending on the gauge type and the comparison between the segment’s compare value and the data value from the CAN message. The compare values of the segments are always listed in ascending order, but can be entered in any order. There is a 200 millisecond inhibit time between outputs to prevent display jitter.

Table 12: Custom Gauge Object Parameters

Parameter	Description
Common CAN Message Parameters plus…	
Gauge Type	• Standard: Multiple segments can be active behaving like a bar graph • Pointer: Only one segment at a time is active behaving like a gauge pointer
Segment Map	Map size: 50 segments maximum
Boundary Value	Value to compare with the CAN data value Range: Based on number of data bits specified
Active Object	ID of object that is displayed if (1) the boundary value is less than or equal to the data value for the standard option or if (2) the data value is between two adjacent segment compare values for the pointer option
Inactive Object	ID of object that is displayed if the Active object is not displayed

The standard gauge type will display all active objects that correspond to segments whose boundary value is less than or equal to the value being graphed. The pointer gauge type will only display one active object from the segment whose boundary value is less than or equal to the value being graphed. The figure below shows a comparison of the two types. Both gauges have the same definition except that one is set to standard type and the other is set to pointer type. Note the hollow box shown at the bottom of the pointer type gauge is actually an inactive object assigned to segment 0 of that gauge.

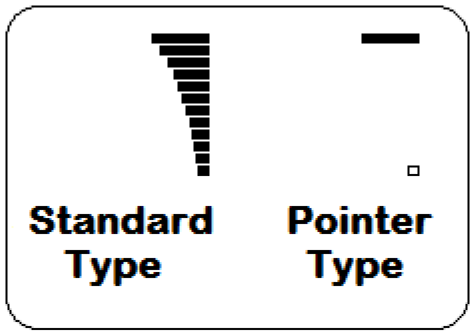


Figure 21: Comparison of Custom Gauge Types

Dynamic Text Object

The Dynamic Text Object monitors CAN traffic for an SPN value and displays the result as a text string on the 3Dxx Color Display LCD. The incoming CAN value is multiplied by the “Resolution” value and then the “Offset” value is added to this result. The final result of this calculation is then processed into an output string using a “C” *printf* style format string (see Note 1 below). The upper left-hand corner of the first character of this output string is displayed at x and y coordinates specified. The font, text color, and text background color of the output string can be selected. There is a 300 millisecond inhibit time between outputs to prevent display jitter.

Table 13: Dynamic Text Object Parameters

Parameter	Description
Common CAN Message Parameters plus…	
Resolution	SPN data is multiplied by this value
Offset	This value is added after SPN data is multiplied by “Resolution”
Pos X	X-coordinate for upper left-hand corner of first character
Pos Y	Y-coordinate for upper left-hand corner of first character
Font	See ***Static Text Object*** section for font types and sizes
Text Color	16-bit color for text
Background Color	16-bit color for text background
String Format	‘C’ *printf* style format specifier (see Note 1 below) - 32 characters maximum -

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`

The edit dialog for the Dynamic Text object provides some additional preview drawing controls to aid in setting up this object. There are “Draw Min” and a “Draw Max” buttons that will display the configured text as it would appear when the CAN input value is zero (0) or the maximum value (based on bit length specified), respectively. There is also a “Draw 66%” button that displays the configured text if the CAN input value was 66% of the maximum possible value based on the specified bit length. Remember that the bit length is specified as part of the Common CAN Message Parameters.

PGN Watchdog Object

The PGN Watchdog Object is used to monitor CAN traffic for a specific PGN and source address, or for a specific message ID. 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.

Table 14: PGN Watchdog Object Parameters

Parameter	Description
PGN	Parameter Group Number containing the SPN parameter to monitor Range: 0x0 – 0x1FFFF ***[Note: Applies to Msg Type = Extended]***
ID	Message ID to monitor Range: 0x0 – 0x7FF ***[Note: Applies to Msg Type = Standard]***
Source Address	Source address of sending device Range: 0x0 – 0xFF ***[Note: Applies to Msg Type = Extended]***
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

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.

Table 15: SPN Range Object Parameters

Parameter	Description
Common CAN Message Parameters plus…	
Min Value	Minimum value of the bit field Range: Based on number of data bits specified
Max Value	Maximum value of the bit field Range: Based on number of data bits specified
Below Range Object	Object that is displayed if the bit field value is less than the Min Value
In Range Object	Object that is displayed if the bit field value is between the Min Value and Max Value (inclusive)
Above Range Object	Object that is displayed if the bit field value is greater than the Max Value

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. When a new value is received, the previous object that was displayed is killed and the new object (if one is found that maps to the new value) is displayed. There is a 300 millisecond inhibit time between outputs to prevent display jitter.

Table 16: SPN Value Object Parameters

Parameter	Description
Common CAN Message Parameters plus…	
SPN Value Object List	List size: 32
Bit Field Value	Value to compare against the SPN value Range: Based on number of data bits specified
Bound Object ID	Object that is displayed if the Bit Field Value matches the SPN value exactly

Appendix A.3. Camera Monitoring Object

Camera Input Object

The Camera Input Object displays a live video feed from the selected video input. Object parameters control the position and size of the video image on the 3Dxx Color Display LCD.

Only one camera input can be active at any one time (i.e., display of multiple simultaneous cameras NOT supported) for each of the following display types:

- model 3D50
- model 3D70
- model 3D2104 running linux kernel 3.0.35

If a second Camera Input Object is displayed, the previously active camera input will be deactivated.

The display of multiple cameras simultaneously is supported on the following display type(s):

- model 3D2104 running linux kernel 4.1.15 or later
- model 3D101 running linux kernel 4.1.15 or later

Certain restrictions apply to simultaneous camera display:

- Up to a maximum of 3 camera inputs can be displayed simultaneously
- Camera inputs 1 and 4 cannot be displayed simultaneously
- Each camera input being displayed must be of the same Layering mode (Foreground or Background), e.g., the display of one Foreground camera and one Background camera is NOT supported.
- Each camera input being displayed must be of the same Type (NTSC or PAL), e.g., the display of one NTSC camera and one PAL camera is NOT supported.

Table 17: Camera Input Object Parameters

Parameter	Description
Camera Input	Model 3D50: Video1 / Video2 Model 3D70: Video1 / Video2 / Video3 Model 3D2104: Video1 / Video2 / Video3 / Video4 Model 3D101: Video1 / Video2 / Video3 / Video4 *Note: some device variants may have fewer or no camera inputs
Camera Input Type	NTSC or PAL
Top Left X	Coordinate of the upper left-hand corner of the camera image on the display
Top Left Y	Coordinate of the upper left-hand corner of the camera image on the display
Width	Width of camera image on display
Height	Height of camera image on display

Parameter	Description
Image Rotation & Flip	One of: • No rotation or flip • Vertical flip & no rotation • Horizontal flip & no rotation • Rotate 180°& no flip • Rotate right 90°& no flip • Rotate right 90°& vertical flip • Rotate right 90°& horizontal flip • Rotate left 90°& no flip
Layering	Foreground or Background
Transparent Color Key	Background Mode only – choose one of: • Black R: 0 G: 0 B: 0 • Red R: 255 G: 0 B: 0 • Green R: 0 G: 255 B: 0 • Blue R: 0 G: 0 B: 255 • Yellow R: 255 G: 255 B: 0 • Cyan R: 0 G: 255 B: 255 • Magenta R: 255 G: 0 B: 255 • White R: 255 G: 255 B: 255
Brightness	Camera image brightness setting Range: 0 to 255
Saturation	Camera image saturation setting Range: 0 to 255
Contrast	Camera image contrast setting Range: 0 to 255
Hue	One of: 0x00, 0x7F, or 0x80

Depending on the display orientation chosen for the project (landscape or portrait and normal or flipped) there are restrictions placed on camera image placement and sometimes its size. A note box will appear on the Edit Camera Object dialog box to describe the restriction for the current display orientation.

The camera inputs are 640 x 480 which results in a 4:3 aspect ratio. When setting the camera image size on the display, if this aspect ratio is not maintained, there will be image distortion. There is a check box on the Edit Camera Object dialog box that if checked will automatically maintain the 4:3 aspect ratio.
When the camera Layering selection is set to Foreground mode, the camera image will cover any graphics drawn on the display where the camera image appears. When the camera is deactivated, the covered graphics will reappear.

When the camera Layering selection is set to Background mode a Transparent Color Key must be selected. The allowable Color Keys and their associated red, green, and blue values are shown in the table above. Any graphics pixels drawn in the area where the camera image appears that match this Color Key will allow the camera image to show through. Pixels of any other color will be opaque and will cover the camera image in that area. This makes it possible to draw graphics over a camera image.

To aid in selecting appropriate settings for the camera input, the Camera Input Object edit dialog has an additional control called “Send Now”. This control is only active if the VUI Builder© is currently connected to a 3Dxx Display via CAN bus. Clicking this button will send the current camera settings to the 3Dxx Display so that the impact of the settings can be immediately viewed on the 3Dxx Display (assuming that the camera is connected to the selected video input).

Note that when the Camera Input Object is previewed on the VUI Builder© preview display or displayed on the 3Dxx Simulator, only a green outline box where the camera image would appear is shown. Inside this green outline box is a brief listing of the parameters that were set in the object.

Appendix A.4. Non-Monitoring Objects

Line Object

The Line Object displays a line on the 3Dxx Color Display LCD. The line position and size is specified by the x and y coordinates of the two endpoints. Any 16-bit color can be selected, as well as the line thickness (in pixels).

Table 18: Line Object Parameters

Parameter	Description
Position X1	X-coordinate of endpoint 1
Position Y1	Y-coordinate of endpoint 1
Position X2	X-coordinate of endpoint 2
Position Y2	Y-coordinate of endpoint 2
Color	16-bit color
Width	Thickness (in pixels) of line Range: 1 – 32

Box Object

The Box Object displays a rectangle on the 3Dxx Color Display LCD. The rectangle position and size is specified by the upper left and lower right coordinates. Any 16-bit color can be selected, as well as whether the rectangle is filled or hollow. If hollow, the line thickness (in pixels) can also be specified.

Table 19: Box Object Parameters

Parameter	Description
Position X1	X-coordinate of upper left corner
Position Y1	Y-coordinate of upper left corner
Position X2	X-coordinate of lower right corner
Position Y2	Y-coordinate of lower right corner
Color	16-bit color
Fill	True / False
Width	Thickness (in pixels) of box sides Range: 1 – 32

Circle Object

The Circle Object displays a circle on the 3Dxx Color Display LCD. The circle position and size is specified by the center coordinate and radius parameters. Any 16-bit color can be selected, as well as whether the circle is filled or hollow. If hollow, the line thickness (in pixels) can also be specified.

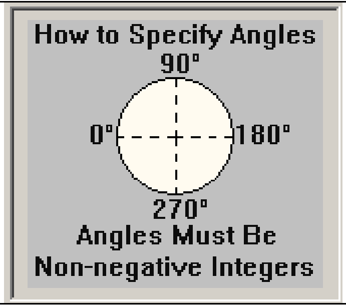
Table 20: Circle Object Parameters

Parameter	Description
Position X	X-coordinate of center
Position Y0	Y-coordinate of center
Radius	Radius of circle
Color	16-bit color
Fill	True / False
Width	Thickness (in pixels) of circle perimeter if “Fill” set to “False” Range: 1 – 32

Arc Object

The Arc Object displays an arc on the 3Dxx Color Display LCD. The arc position and size is specified by the center coordinate, radius, and start and end angle parameters. Any 16-bit color can be selected, as well as whether the arc is filled or hollow. If hollow, the line thickness (in pixels) can also be specified.

Table 21: Arc Object Parameters

Parameter	Description	
Position X	X-coordinate of center	
Position Y0	Y-coordinate of center	
Radius	Radius of arc	
Start Angle	Arc starts at this angle Range 0° to 360°	 How to Specify Angles 90° 0° 180° 270° Angles Must Be Non-negative Integers
End Angle	Arc ends at this angle Range: 0° to 360°	
Color	16-bit color	
Fill	True / False	
Width	Thickness (in pixels) of arc perimeter if “Fill” set to “False” Range: 1 – 32	

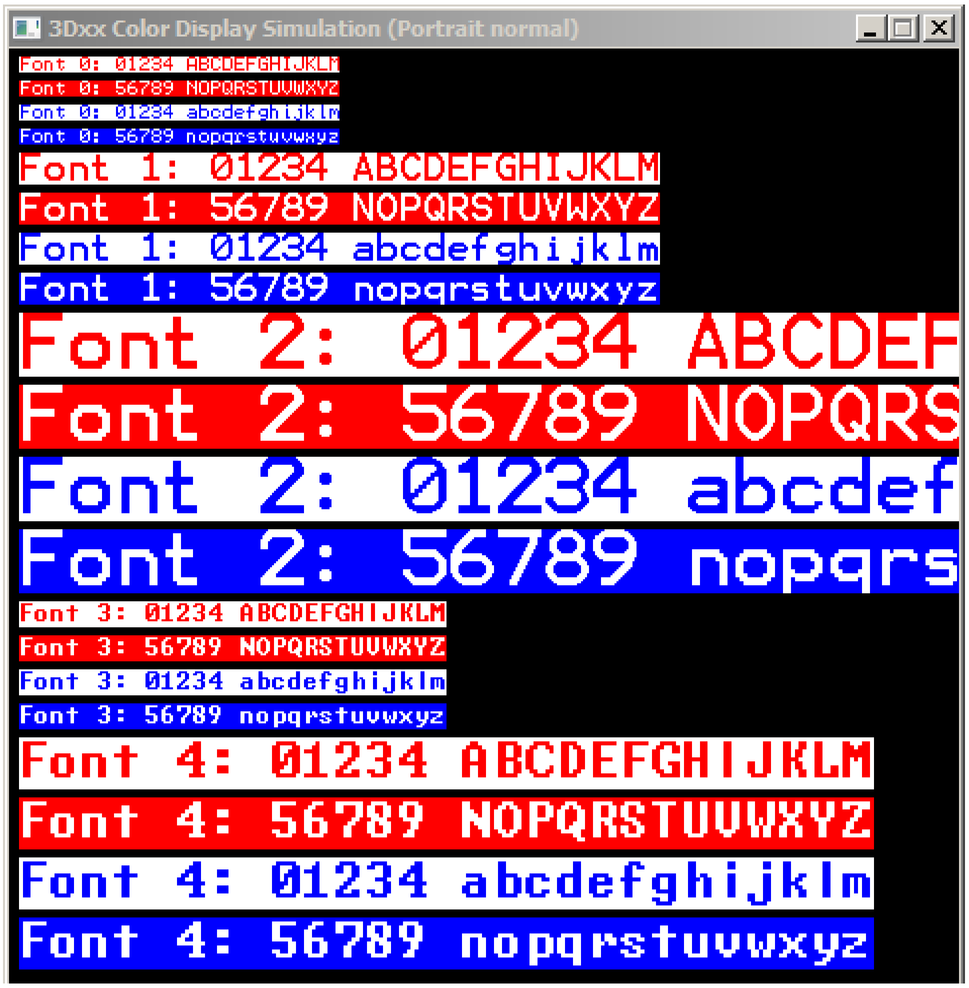
Static Text Object

The Static Text Object displays a string of text on the 3Dxx Color Display LCD. The text position is specified by the x and y coordinates of the upper left corner of the first character. One of five fonts can be specified, as well as the text color and the text background color. The text to be displayed is either entered directly in the edit dialog or can be obtained from another text object. Using another text object can save memory space if the same long message needs to be displayed in different locations and/or with different colors or fonts.

Table 22: Static Text Object Parameters

Parameter	Description			
Position X	X-coordinate of upper left corner of first character			
Position Y	Y-coordinate of upper left corner of first character			
Font		Font Name	Character Width (in pixels)	Character Height (in pixels)
	Font 0	6x8	6	8
	Font 1	12x16	12	16
	Font 2	24x32	24	32
	Font 3	8x12	8	13*
	Font 4	16x24	16	26*
	* - extra row(s) added to provide symmetrical output.			
Text Color	16-bit color for text			
Background Color	16-bit color for text background			
Text	Message to display			

Here are samples of all five fonts:



Font 0: 01234 ABCDEFGHIJKLM
Font 0: 56789 NOPQRSTUVWXYZ
Font 0: 01234 abcdefghijklm
Font 0: 56789 nopqrstuvwxyz
Font 1: 01234 ABCDEFGHIJKLM
Font 1: 56789 NOPQRSTUVWXYZ
Font 1: 01234 abcdefghijklm
Font 1: 56789 nopqrstuvwxyz
Font 2: 01234 ABCDEF
Font 2: 56789 NOPQRS
Font 2: 01234 abcdef
Font 2: 56789 nopqrs
Font 3: 01234 ABCDEFGHIJKLM
Font 3: 56789 NOPQRSTUVWXYZ
Font 3: 01234 abcdefghijklm
Font 3: 56789 nopqrstuvwxyz
Font 4: 01234 ABCDEFGHIJKLM
Font 4: 56789 NOPQRSTUVWXYZ
Font 4: 01234 abcdefghijklm
Font 4: 56789 nopqrstuvwxyz

Figure 22: Font Samples

Bitmap Object

The Bitmap Object displays an image from a file on the 3Dxx Color Display LCD. The user may select an image file from anywhere on the PC and once a file is selected it is copied to the "Images" directory for the current project (See **Section 13Adding a New Object – Bitmap Example** for more details). Currently the 3Dxx Color Display supports BMP, PNG, and JPG files.

The position on the display where the image will appear is specified by the x and y coordinates. They determine where the upper left corner of the image will be placed. The width and height parameters are determined by the imported image file and are not modifiable.

Table 23: Bitmap Object Parameters

Parameter	Description
Position X	X-coordinate of upper left corner
Position Y	Y-coordinate of upper left corner
Width	Width in pixels ***- Read Only -***

Parameter	Description
Height	Height in pixels ***- Read Only -***
File name	File name of the image file to be displayed

Audio Output Object (Model 3D70 only)

The Audio Output Object plays an mp3 file on the 3D70 Color Display and sends the output to the audio mono line out (pins 1, AUDIO OUT, and 2, AUDIO RET, on the B connector). The user may select an mp3 file from anywhere on the PC and once a file is selected it is copied to the "Images" directory for the current project (See **Section 13Adding a New Object – Bitmap Example** for more details).

When an Audio Output object is "displayed", the selected mp3 file will begin playing and program execution will continue and not wait for the mp3 file playback to finish. If another mp3 file was already being played when a new mp3 file play request is received, the previous playback will be stopped and the new file will begin playing (even if it is the same file). If an Audio Output object is killed, it will immediately stop playing.

Table 24: Audio Output Object Parameters

Parameter	Description
File name	File name of the mp3 file to be played

Group Object

The Group Object holds a collection of other objects. This allows multiple objects to be displayed or removed (killed) when the group object ID is referenced. A group object may also contain other group objects. The objects within a group object are displayed or killed in order, which is important to keep in mind when objects overlap each other. There are arrow buttons to the right of the box listing the group's items. These arrow buttons can be used to move a single selected item in the group's list up or down in the group's order.

There are also three options to control group object handling. The first option just displays each object listed in order. The second option will kill all objects before the objects in the group list are displayed. This can be useful when switching to a new screen page and everything needs to be reset. The third option will kill each object listed. This can be useful for stopping some CAN monitoring objects that are no longer needed when it is undesirable to reset the whole screen.

Table 25: Group Object Parameters

Parameter	Description
Group option selector	The option selector allows selection of one of the following options: • Display each object listed in order • Kill all objects and then display each object listed • Kill each object listed in order
Group Objects	List of objects included in group – listed in order in which objects will be displayed or killed. Each Object's ID Number, Type, and Name are shown.
Available Objects	List of available objects to include in group. Each Object's ID Number, Type, and Name are shown.

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.

Table 26: Default Screen Object Parameters

Parameter	Description
Object ID	An object is selected from a list of available objects in the project and then the Object ID and Object Name are automatically filled in.
Object Name	

Splash Screen Object

The Splash Screen Object references an object to be displayed when the display is first powered up. This object remains active for the specified time duration. Once the time expires, the Splash Screen object is removed and the Default Screen object (if assigned) is displayed.

Table 27: Splash Screen Object Parameters

Parameter	Description
Object ID	An object is selected from a list of available objects in the project and then the Object ID and Object Name are automatically filled in.
Object Name	
Duration	Length of time (in seconds) splash screen will be displayed Range: 1 – 60

Touch Screen Event Object

The Touch Screen Event Object is used to allow an operator to select different displays and options on the 3Dxx Color Display by simply tapping, touching, or swiping the surface of the screen. A Touch Screen Event Object can be used to define either swipe event responses, tap event responses, or responses to the start of a touch and end of that touch. These three types of event definitions will be covered separately below.

Touch Screen Swipe Event Definition

There can only be one object defining swipe events active at any given time. If a second such object is activated (displayed), its definitions will override the previous ones. Actions can be assigned to each of the four possible gestures: right, left, up, and down.

A gesture is detected when the operator’s finger moves in the specified direction by a certain minimum amount (currently 150 pixels) without moving in an orthogonal direction by more than a certain amount (currently 100 pixels). Gestures that do not fit these parameters are ignored. The gesture is not acted upon until the operator removes his finger from the display’s surface.

Once a gesture is detected, the 3Dxx Display will first kill the object(s) specified (if any). The specified object may be a group in order to have multiple objects killed. For convenience there is a “Kill All” option. This is useful when a gesture is used to navigate to a new screen and all currently active objects need to be killed. Remember that the “Kill All” will also deactivate all Touch Screen Events (including Taps, Start/End Events, and Swipes). It is also possible for the object to specify itself to be the object to kill. This will cause the touch object to become inactive after it finishes displaying the object specified by the current gesture (if any).

Next the 3Dxx Display will display the object specified (if assigned). Again the object displayed can be a group object in order to activate and display multiple objects. The object is not allowed to display itself.

Table 28: Touch Screen Swipe Event Parameters

Gesture	Object(s) to Kill	Object to Display
Swipe Right	None, All, or Selected	None or Selected
Swipe Left	None, All, or Selected	None or Selected
Swipe Up	None, All, or Selected	None or Selected

Gesture	Object(s) to Kill	Object to Display
Swipe Down	None, All, or Selected	None or Selected

Touch Screen Tap Event Definition

A Tap Event object specifies an area on the 3Dxx Display and what to do if the operator taps anywhere within this defined area. In addition to an area, the user may also specify a minimum time that is used to determine if the tap is a short tap or a long tap. Taps that happen faster than this minimum time are considered short taps and taps that take longer than the specified time are considered long taps.

There can be as many objects defining tap events as desired. If the tap areas overlap, usually the object that is activated (displayed) first will take precedence over any later objects that define actions for the same area. This order may be different if other tap objects were killed and new ones created.

A tap is detected when the operator's finger touches the 3Dxx Display and does not move more than a specified distance (currently 80 pixels). The center of the tap must fall within an area that has been defined by a currently active Touch Screen Event Object. Taps that do not fit these parameters are ignored. The tap is not acted upon until the operator removes his finger from the display's surface.

Once a tap is detected, the 3Dxx Display will first kill the object(s) specified (if any). The specified object may be a group in order to have multiple objects killed. For convenience there is a "Kill All" option. This is useful when a tap is used to navigate to a new screen and all currently active objects need to be killed. Remember that the "Kill All" will also deactivate all Touch Screen Events (including Taps, Start/End Events, and Swipes). It is also possible for the object to specify itself to be the object to kill. This will cause the touch object to become inactive (after it finishes displaying the object specified, if any, by the tap gesture) and will remove any text or graphics the touch object drew.

Next the 3Dxx Display will display the object specified (if assigned). Again the object displayed can be a group object in order to activate and display multiple objects. A touch object is not allowed to display itself.

Table 29: Touch Screen Tap Event Parameters

Tap Type	Object(s) to Kill	Object to Display
Short	None, All, or Selected	None or Selected
Long	None, All, or Selected	None or Selected

There are three ways to configure the appearance of a Tap Event Object on the 3Dxx Display. The first way is the "no graphics" method. This just defines a touch area on the screen and what objects to kill and display for short and long taps. This is useful when a bitmap image with button graphics is already being displayed on the screen.

When a draw preview of this type of object is selected, the defined touch area will flash white and then black on the preview screen so that the dimensions of the touch area can be observed against the bitmap image being used. This flashing does not take place when doing a preview draw of a group of objects that may contain such tap event "no graphics" objects.

Table 30: "No Graphics" Tap Event Object Parameters

Parameter	Description
Minimum Milliseconds for Long Tap	Tap must last longer than this time to be considered "long" Range: 300 to 2000 milliseconds
Position X	X-coordinate of upper left corner of tap area
Position Y	Y-coordinate of upper left corner of tap area
Button Width	Width of tap area
Button Height	Height of tap area

The second method to configure the appearance of a Touch Screen Tap Event Object on the 3Dxx Display is to specify a bitmap image. The bitmap image will be displayed and its size will determine the size of the tap touch area. The bitmap image is handled as described under the ***Bitmap Object*** section above.

Table 31: Bitmap Image Tap Event Object Parameters

Parameter	Description
Minimum Milliseconds for Long Tap	Tap must last longer than this time to be considered "long" Range: 300 to 2000 milliseconds
Position X	X-coordinate of upper left corner of tap area
Position Y	Y-coordinate of upper left corner of tap area
Button Image	File name of bitmap image to use
Button Width	Width of tap area (automatically set by image size) ***- Read Only -***
Button Height	Height of tap area (automatically set by image size) ***- Read Only -***

The final method of specifying the Touch Screen Tap Event appearance will optionally draw a box the size of the tap area and will optionally center text inside this area. The color of the box and its border width can be specified. The optional text allows a selection of font, text color, and text background color.

Table 32: Button Box and/or Text Tap Event Object Parameters

Parameter	Description
Minimum Milliseconds for Long Tap	Tap must last longer than this time to be considered "long" Range: 300 to 2000 milliseconds
Position X	X-coordinate of upper left corner of tap area
Position Y	Y-coordinate of upper left corner of tap area
Button Width	Width of tap area
Button Height	Height of tap area
Border Width	Width of box's border (use 0 for no box) Range: 0 to ½ smallest dimension of box (allows box to be filled in)
Border Color	16-bit color for button border
Button Text	Text to center in tap area (may be blank for no text)
Font	See ***Static Text Object*** section for font types and sizes
Text Color	16-bit color for text
Background Color	16-bit color for text background

Touch Screen Start and End Touch Event Definition

A Start and End Touch Event object specifies an area on the 3Dxx Display and what to do when the operator starts touching anywhere within this defined area and what to do when he finally removes his finger from the display. Currently the Start Touch event is not recognized until about 100 milliseconds after the operator starts touching the display. Touches shorter than this are considered to be Taps and will not be processed by a Start Touch Event (therefore such a short Tap will probably be ignored). Once the Start Touch Event has been recognized, it does not matter how much or where the operator moves his finger on the display. The only event that will register after a Start Touch Event is an End Touch Event which is triggered when the operator removes his finger from the display.

There can be as many objects defining Start/End Touch events as desired. If the defined areas overlap, usually the object that is activated (displayed) first will take precedence over any later objects that define actions for the same area. This order may be different if other touch objects were killed and new ones created.

Once a Start Touch Event is detected, the 3Dxx Display will first kill the object(s) specified (if any). The specified object may be a group in order to have multiple objects killed. For convenience there is a “Kill All” option. Remember that the “Kill All” will also deactivate all Touch Screen Events (including Taps, Start/End Events, and Swipes). It is also possible for the object to specify itself to be the object to kill. This will cause the touch object to become inactive (after it finishes displaying the object specified, if any, and performing the End Touch actions, if any) and will remove any text or graphics the touch object drew.

Next the 3Dxx Display will display the object specified (if assigned). Again the object displayed can be a group object in order to activate and display multiple objects. A touch object is not allowed to display itself.

Table 33: Touch Screen Start/End Touch Event Parameters

Touch Event Type	Object(s) to Kill	Object to Display
Start Touch	None, All, or Selected	None or Selected
End Touch	None, All, or Selected	None or Selected

There are three ways to configure the appearance of a Start/End Touch Event Object on the 3Dxx Display. They are the same as the methods listed for the Tap Event processing above. They use all of the same parameters except for the Minimum Milliseconds for Long Tap parameter which is not used for a Start/End Touch Event Object.

Backlight & Background Object

The Backlight & Background Object is used to control the backlight intensity of the LCD and/or set the background color for the display. The Backlight Command controls how this object will affect the backlight setting. It can change the backlight setting to a given percentage or leave it unchanged. If used to set a default setting, then this setting will be used whenever the display is erased (i.e. by doing a “Kill All”). The Backlight Command can also be set to increment or decrement the current backlight setting by the amount specified. This is useful when such an object is linked to a Touch Screen Event because this can allow the operator to adjust the backlight setting with a touch input.

If the Backlight is set to 0% the display cannot be seen at all. When decrementing the backlight setting it is limited to a minimum value of 10% so that the display remains visible; however, the "Set to" commands can set the backlight to a setting less than 10%. The Increment command will not go past 100%. Increment and decrement commands that try to go past the limits will just set the backlight to the limit.

By default the background color is set to black on the 3Dxx Color Display. If, for example, a bar graph gauge is being displayed, the unfilled area of the gauge will be black. Also killing a Bitmap Object is accomplished by drawing a filled rectangle the size of the Bitmap Object using the current background color. Changing the background color will cause these areas to be set to whatever color that is specified by this object. This object can also be used to fill the whole display with the new background color selected.

Table 34: Backlight & Background Object Parameters

Parameter	Description
Backlight Controls	
Backlight Command	Select one of: • No Change – No change to backlight • Set to – Change backlight to setting given below • Increment by – Increment level by amount given below • Decrement by – Decrement level by amount given below
Backlight (%)	Amount to set/increment/decrement backlight Range: 0 to 100%
Set new value as default	If checked, makes the new backlight setting the default setting (used when display is erased)

Parameter	Description
Background Color Controls	
Set default background color	Set new display background color if checked
Background color	16-bit color to use for display background
Fill screen now with background color	If this box is checked, fills whole display with background color when this object is displayed

Buzzer Object

For models with an internal buzzer, this object is used to turn the buzzer on or off, or to sound the buzzer for a specified amount of time. The last object "displayed" takes precedence over any previous command. So if the buzzer is commanded to sound for two seconds and then one second later it is commanded to sound for five seconds, the buzzer will sound for a total of six seconds. Conversely, if the buzzer is commanded to sound for four seconds and two seconds later a buzzer off object is "displayed", the buzzer will be silenced immediately. A "Kill All" operation will also silence the buzzer.

Table 35: Buzzer Object Parameters

Parameter	Description
Buzzer Command	Choose one of: • Turn Buzzer Off • Time Buzzer On for Specified Milliseconds • Time Buzzer On for Specified Seconds • Time Buzzer On
The following parameters only appear when appropriate:	
Buzzer time in milliseconds	Range: 100 to 10000
Buzzer time in seconds	Range: 1 to 3600
Buzzer Level (%)	Range: 10 to 100 (increments of 10)

Date/Time Object

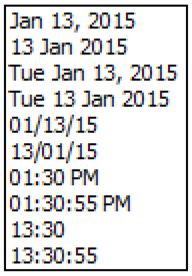
A Date/Time Object can do one of three things: it can display the current date, it can display the current time, or it can launch a special dialog that allows the operator to set the date and time using touch screen input. This object can only be displayed on 3Dxx Color Display hardware that has Real-Time Clock (RTC) hardware.

Table 36: Date/Time Object Parameters

Parameter	Description
Date/Time Command or Display Format	Choose one of: • Date format • Time Format • Command to launch Date/Time Input Dialog
Text Color	16-bit color for text
Background Color	16-bit color for text background
X Position	X-coordinate of upper left corner of first character
Y Position	Y-coordinate of upper left corner of first character
Font	See ***Static Text Object*** section for font types and sizes

When an object to display the date or time is displayed, it will continue to update the date or time automatically every second until the object is killed. The date or time display appearance can be controlled by selecting a font, as well as the text color and the text background color. There are several formats available for displaying the date or the time.

The object edit dialog will report an error if the combination of X and Y coordinates, font choice, and format selection will not fit on the 3Dxx Display. Here is a sample of the formats available:

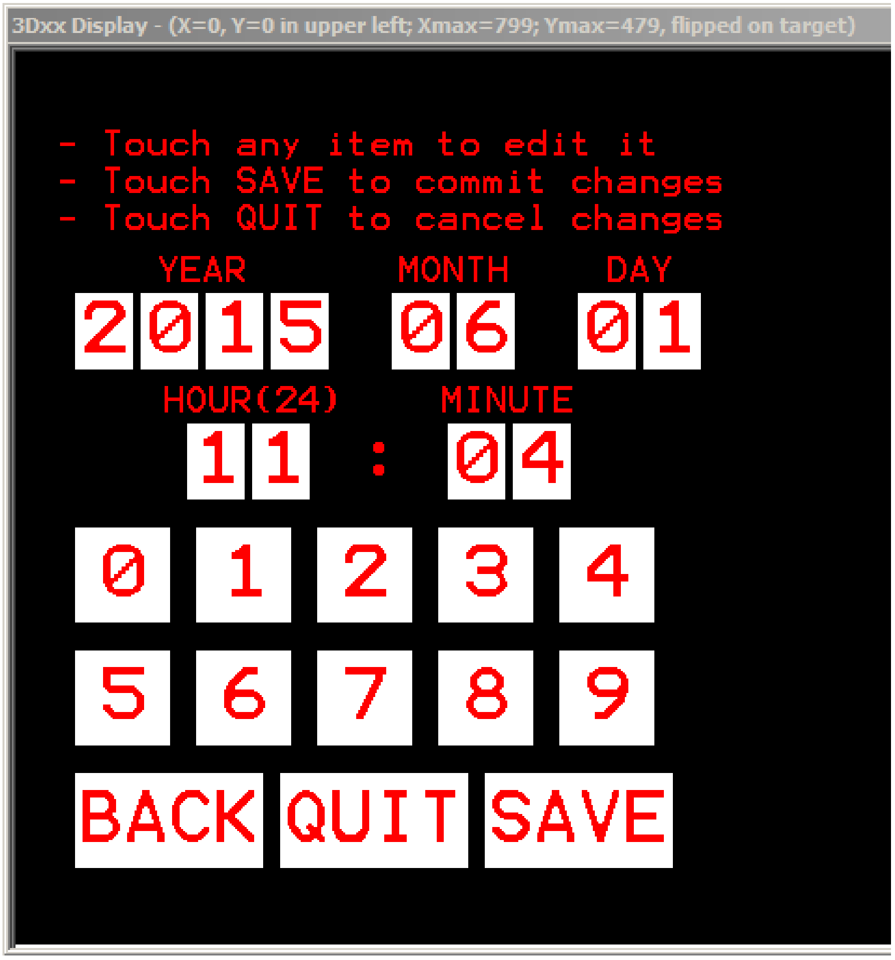


Jan 13, 2015
13 Jan 2015
Tue Jan 13, 2015
Tue 13 Jan 2015
01/13/15
13/01/15
01:30 PM
01:30:55 PM
13:30
13:30:55

Figure 23: Sample of Date/Time Display Formats

The date/time setting dialog requires that the 3Dxx Display hardware has touch screen input hardware installed. This special input dialog will cover half of the 3Dxx Display screen (left half if using Landscape Mode or top half if using Portrait Mode). This dialog uses a black background, but the text messages and buttons will use the Text Color and Text Background Color specified.

While this dialog is active, all CAN monitoring is suspended and touch inputs are only active for this dialog. Any Camera inputs are deactivated. When the dialog exits the display will be restored and all CAN monitoring and touch screen event processing will resume as they were before; however, any camera input that was selected will remain deactivated.



3Dxx Display - (X=0, Y=0 in upper left; Xmax=799; Ymax=479, flipped on target)

- Touch any item to edit it
- Touch SAVE to commit changes
- Touch QUIT to cancel changes

YEAR 2015 MONTH 06 DAY 01

HOUR(24) 11 : MINUTE 04

Figure 24: Date/Time Setting Dialog Preview

When the dialog starts the third digit of the year will be flashing to indicate that this is the next digit to be input using one of the numeric tap areas provided. When the operator taps a numeric input digit, the next digit is automatically selected. The operator may tap any date/time digit to redirect input to that digit. He may also swipe up, down, left, or right to advance to the next parameter. There is also a “BACK” button to allow the operator to select the previous digit. The operator may tap the “QUIT” or “SAVE” touch areas to exit the dialog. The “QUIT” option will not change the current date or time setting, but the “SAVE” option will update the current date and time and save the new setting to the battery-backed RTC hardware.

GPIO Input Object

The GPIO Input Object defines what object is to be displayed when any of the 3Dxx Color Display GPIO inputs changes state. Objects can be assigned for the low to high and high to low transitions for each input. There is also an “Ignore” check box if no object is to be displayed for a given transition. The inputs are monitored once every 10 milliseconds and an input must be found in the same high or low state for 8 consecutive samples before it is considered to be in a high or low state.

Note that Pin 4 on the 3D50 (five inch) Color Display connector is a dedicated input only pin. Pins 6, 7, and 8 are I/O pins that can be used to output a signal or input a signal. In order for these pins to function as input pins, the corresponding output must be set to low. All Model 3D50 inputs have a 51.1Kohm pull down.

On the 3D70 (seven inch) Color Display each of the four inputs are dedicated and so operate independently of any output pins. Each input pin is configured with a 12.5Kohm pull-down.

On the 3D2104 (10.4 inch) Color Display all pins are I/O pins that can be used to output a signal or input a signal. In order for these pins to function as input pins, the corresponding output must be set to low. All Model 3D2104 inputs have a 12Kohm pull down.

On the 3D101 (10.1 inch) Color Display all pins are I/O pins that can be used to output a signal or input a signal. In order for these pins to function as input pins, the corresponding output must be set to low. All Model 3D101 inputs have a 12Kohm pull down.

On all models, an input must be less than 2.2 volts to be detected as low and must be greater than 2.7 volts to be detected as high.

Table 37: GPIO Digital Input Object Parameters

3D50 Pins	3D70 Pins	3D2104 Pins, 3D101 Pins	Object to Activate on Input Signal Transition
Input 1 (Pin 4)	Input 1 (Pin 4 Connector A)	I/O 1 (Pin 10)	Object to display when input goes from high to low
			Object to display when input goes from low to high
I/O 1 (Pin 6)	Input 2 (Pin 8 Connector B)	I/O 2 (Pin 21)	Object to display when input goes from high to low
			Object to display when input goes from low to high
I/O 2 (Pin 7)	Input 3 (Pin 9 Connector B)	I/O 3 (Pin 32)	Object to display when input goes from high to low
			Object to display when input goes from low to high
I/O 3 (Pin 8)	Input 4 (Pin 10 Connector B)	I/O 4 (Pin 9)	Object to display when input goes from high to low
			Object to display when input goes from low to high

GPIO Output Object

The GPIO Output Object provides independent control of the 3Dxx Color Display GPIO outputs. Outputs default to low (0) on power up. When in the low state the pin is left floating (except for Pins 6, 7, and 8 on a Model 3D50 and all pins on Model 3D2104 and Model 3D101 which have the 51.1Kohm pull down). If a shared I/O pin is set low then the pin may be used as an input. When a pin is set high it is connected to Vbatt and may source up to 200 mA. Selecting 'No Change' for a value allows one output to be controlled without affecting the others.

On Model 3D50 (five inch display) Pin 5 is a dedicated output only pin on the 3D50 Color Display connector. Pins 6, 7, and 8 are I/O pins that can be used to output a signal or input a signal. In order for these pins to function as input pins, the corresponding output must be set to low.

On Model 3D70 (seven inch display) all four output pins (which are on connector B) are dedicated and not shared with input pins.

On Model 3D2104 (10.4 inch display) all digital output pins are I/O pins that can be used to output a signal or input a signal. In order for these pins to function as input pins, the corresponding output must be set to low.

On Model 3D101 (10.1 inch display) all digital output pins are I/O pins that can be used to output a signal or input a signal. In order for these pins to function as input pins, the corresponding output must be set to low.

Table 38: GPIO Digital Output Object Parameters

3D50 Parameter	3D70 Parameter	3D2104 Parameter, 3D101 Parameter	Description
Output 1 (Pin 5)	Output 1 (Pin11*)	I/O 1 (Pin 10)	Value: Low, High, No Change
I/O 1 (Pin 6)	Output 2 (Pin12*)	I/O 2 (Pin 21)	Value: Low, High, No Change
I/O 2 (Pin 7)	Output 3 (Pin13*)	I/O 3 (Pin 32)	Value: Low, High, No Change
I/O 3 (Pin 8)	Output 4 (Pin14*)	I/O 4 (Pin 9)	Value: Low, High, No Change

* On Model 3D70 all GPIO Output Pins are on Connector B

Analog Input Object (Model 3D70 only)

The Analog Input Object defines what to do with analog readings taken on one of two inputs (pin 4 or pin 5 on Connector B). The Analog Input object can be used to read resistance, voltage, or current with respect to the analog return pin (pin 7 on Connector B). The readings (which are sampled once every 10 milliseconds) can be directed to a previously declared variable. By referencing this same variable in a CAN monitoring object such as Bar Graph or Dynamic Text these readings can be displayed in many different ways.

Optionally, the Analog Input object can also detect transitions and then display or kill objects when these transitions are detected. There are two basic transitions detected, low to high and high to low. For each transition, a threshold value is specified. A number of samples required above and below the threshold are also specified (samples are collected once every 10 milliseconds). If the Hysteresis check box is checked, then a transition will not be detected again until the opposite transition has been detected. Either transition may be detected first.

Table 39: Analog Input Object Parameters (Model 3D70 only)

Parameter	Description
Analog Input	Choose one of: • Analog Input 1 (Pin 4 on Connector B) • Analog Input 2 (Pin 5 on Connector B)
Measurement Mode	Choose one of: • 0 – 5 Volts (returns: 0 – 5000 millivolts)

Parameter	Description
	• 0 – 1.5 Kohms (returns: 0 – 1500 ohms) • 0 – 10 Volts (returns: 0 – 10,000 millivolts) • 0 – 5 Kohms (returns: 0 – 5000 ohms) • 0 – 20 milliamps (returns: 0 – 20000 microamps)
Send Data to Variable	Select a previously declared variable – Allows analog input to be "monitored" with a CAN monitoring object such as Bar Graph or Dynamic Text by referencing the same variable in one of those objects.
Hysteresis On	Check box to turn on hysteresis (once a transition is detected, it will not be acted upon again until the opposite transition has been detected)
Low to High Transition	
Threshold	Analog input must change from being below this value to being at or above this value for the specified number of samples in order to detect a low to high transition.
# Samples Below	Number of continuous samples that must be below threshold
# Samples Above	Number of continuous samples that must be at or above threshold
Kill Object ID	ID of object to kill when low to high transition is detected
Display Object ID	ID of object to display when low to high transition is detected
High to Low Transition	
Threshold	Analog input must change from being above this value to being at or below this value for the specified number of samples in order to detect a high to low transition.
# Samples Above	Number of continuous samples that must be above threshold
# Samples Below	Number of continuous samples that must be at or below threshold
Kill Object ID	ID of object to kill when high to low transition is detected
Display Object ID	ID of object to display when high to low transition is detected

CAN Transmission Object

The CAN Transmission Object is used to transmit a CAN message when instantiated ("displayed"). The full 29-bit (Extended message type) or 11-bit (Standard message type) CAN ID is specified, as well as which CAN bus to use for the transmission, the data length, and the 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, and up to eight data and mask bytes. The object is successfully acknowledged when a message is received on the selected CAN bus 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.

If the CAN transmit data contains any variable references, the CAN Transmission Object will transmit the object with the current variable values as soon as it is instantiated. After the transmission sequence is completed as defined above, the CAN Transmission Object will begin waiting for changes to any of the referenced variables. If it detects a change in a variable that would change the transmit message data bytes, then the CAN Transmission Object will begin a new transmission cycle with the new data. This will continue until the CAN Transmission Object is killed.

Table 40: CAN Transmission Object Parameters

Parameter	Description
CAN Bus Select	Select which CAN bus to use for transmission
Msg Type	Extended (29-bit) or Standard (11-bit)
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	***For Msg Type = Extended*** 29 bit identifier of transmitted CAN message, includes Priority, Data Page bit, PGN (16 bits) and Source Address (8 bits) Range: 0 – 0x1FFFFFFF ***For Msg Type = Standard*** 11 bit identifier of transmitted CAN message Range: 0 – 0x7FF
CAN DLC	Data Length of transmitted CAN message Range: 0 – 8
CAN Data	Data bytes of transmitted CAN message. These bytes may contain a reference to a variable. See the **Using Variables** section for more details.
ACK PGN	Parameter Group Number of ACK message Range: 0x0 – 0x1FFFF ***[Note: Applies to Msg Type = Extended]***
ACK ID	11 bit identifier of ACK message Range: 0x0 – 0x7FF ***[Note: Applies to Msg Type = Standard]***
ACK Source Address	Source address of ACK message ***[Note: Applies to Msg Type = Extended]***
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.

Timer Object

The Timer Object can be used to periodically toggle between two objects (which can be useful to blink an object on the screen) or to perform a one-time delay function (which can be used to display a pop-up message).

Table 41: Timer Object Parameters

Parameter	Description
Time ON	Units: milliseconds Range: 0 – 10000 (May use a variable for this time by inserting a declared variable name into the edit box.)
“No Kill” check box	Check this box to not have the Time ON object killed after ON time expires.
Time ON Object ID & Object Name	ID and Name of object to display during Time ON
Time OFF	Units: milliseconds Range: 0 – 10000 (If Time OFF is zero then timer stops after one cycle.) (May use a variable for this time by inserting a declared variable name into the edit box.)
“No Kill” check box	Check this box to not have the Time OFF object killed after OFF time expires. Ignored if OFF time is zero.
Time OFF Object ID & Object Name	ID and Name of object to display during Time OFF (This object may be left blank.)

In detail, the Timer Object works as follows:

1. When the Timer Object is first displayed, it will display the Time ON object.
2. When the Time ON delay has been completed, the Timer Object will kill the Time ON object (if the associated “No Kill” check box isn’t checked) and display the Time OFF object (if any).
3. If the Time OFF time is zero, then the Timer Object doesn’t do anything else; otherwise, after the Time OFF delay has been completed, the Timer Object will kill the Time OFF object (if the associated “No Kill” check box isn’t checked) and return to step #1.

Variable Object

The Variable object has several operation options, one of which is a declare operation. Before a variable can be used in a project, it must be declared by creating an object naming the variable with the declare operation selected. This declaration object never needs to be "displayed"; these objects are automatically processed whenever a project is read by the 3Dxx Application Software or the VUI Builder©. If a declare operation Variable object is "displayed" or "killed", it has no effect.

All declared variables are initialized when the project is loaded. The object that declares an object has an option to recall the variable’s initial value from the FLASH file system. If this option is not selected or if the value of the variable is not available from the FLASH file system, then the initial value listed in the declare object for the variable is used.

The table below shows the parameters that need to be set for the Declare Variable operation.

Table 42: Variable Object Parameters for Declare Operation

Parameter	Description
Variable Operation	• Declare Variable
Select Variable	Select variable to declare (A – Z)

Parameter	Description
Minimum Value(in decimal)	Variable part of variable is not allowed to go below this value.
Maximum Value(in decimal)	Variable part of variable is not allowed to go above this value.
Mask Value (in hex)	16-bit mask to specify which bits should not change. Bits set to one do not change, but bits can only be set in consecutive groups at the beginning and end of the 16-bit value.
Initial Value (in hex)	16-bit value that sets state of bits that do not change (as indicated by mask value) and sets initial value of variable bits if initial value cannot be read or is not supposed to be read from FLASH. See section **16. Using Variables** above for more details.
Recall initial value from FLASH	Check this box if variable's initial value should come from a value saved from previous time unit was powered up.

One way to change the value of a variable is create an object that sets, increments, or decrements the value of the variable. Each time such an object is "displayed", the value of the variable is adjusted accordingly. Note that for the increment and decrement operations it is possible to specify how much the variable is to be incremented or decremented. This makes it possible, for example, to have one button increment the variable by one and another button increment the variable by ten.

The table below shows the parameters that need to be set for a Variable object that is doing a Set, Increment, or Decrement operation.

Table 43: Variable Object Parameters for Set, Increment, and Decrement Operations

Parameter	Description
Variable Operation	Choose one of: • Set Variable to Value Below • Increment Variable by Value Below • Decrement Variable by Value Below
Select Variable	Select variable that has been declared (A – Z)
Minimum Value (in decimal)	Read Only value obtained from variable's declaration
Minimum Value (in decimal)	Read Only value obtained from variable's declaration
Set, Increment, or Decrement Value	Enter decimal value to be used for this variable operation

The other way to set a variable is to have the operator enter a new value by using an on-screen keypad. The on-screen keypad will appear on the screen at the position specified when this object is "displayed". The keypad uses a black background with a gray and white border, but the text color and the keypad button color can be selected as shown below. The table below shows the parameters that need to be set for a Variable object that is using the on-screen keypad.

Table 44: Variable Object Parameters for Set Via Keypad Operation

Parameter	Description
Variable Operation	• Set Via On-Screen Keypad
Select Variable	Select variable that has been declared (A – Z)
Minimum Value (in decimal)	Read Only value obtained from variable's declaration
Minimum Value (in decimal)	Read Only value obtained from variable's declaration
Variable's Initial Value (in decimal)	Read Only value obtained from variable's declaration
Position X	X-coordinate of upper left corner of on-screen keypad

Parameter	Description
Position Y	Y-coordinate of upper left corner of on-screen keypad
Font	The keypad buttons always use Font 2 (24x32), but this parameter specifies the font to use for the line showing the number being input. See ***Static Text Object*** section for font types and sizes
Text Color	16-bit color for text on buttons and the input line
Background Color	16-bit color for button background and input line background

While this on-screen keypad is active, all CAN monitoring is suspended and touch inputs are only active for this keypad. Any Camera inputs are deactivated. The keypad exits when the operator touches the “QUIT” or “SAVE” buttons and then the display will be restored and all CAN monitoring and touch screen event processing will resume as they were before; however, any camera input that was selected will remain deactivated.

The following shows an example of an on-screen keypad using Font 1 (12x16) for the Font, red for the Text Color and green for the Background Color:

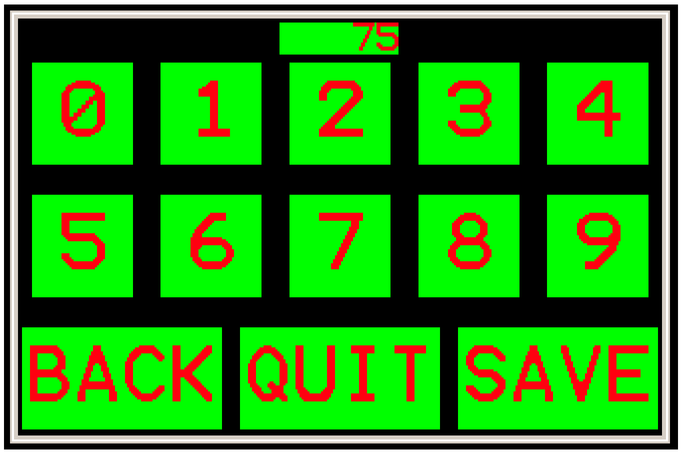


Figure 25: Example of On-Screen Keypad for Setting a Variable

System Info Object

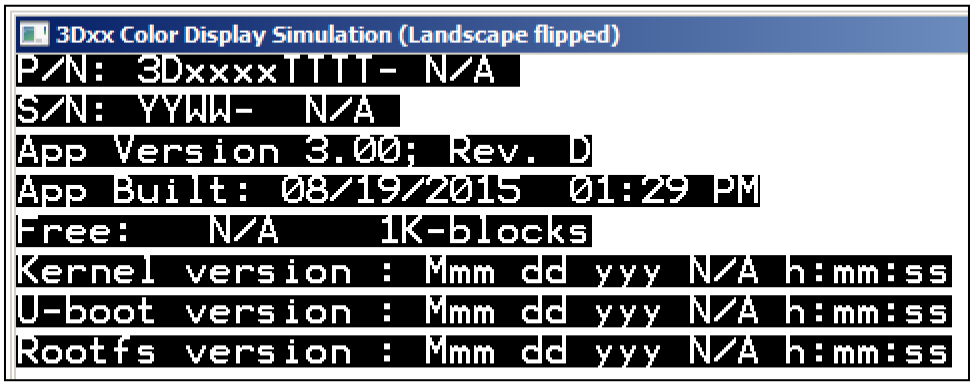
The System Info Object is used to display internal system information on the 3Dxx Color Display. The placement of the output report as well as the font and colors used can be selected. The table below shows the choices available.

Table 45: System Info Object Parameters

Parameter	Description
System Info to Show	Choose one of: • System Part Number • System Serial Number • Application Version • Application Build Date&Time • File System Free Space • Kernel Version • U-boot Version • Rootfs Version
Position X	X-coordinate of upper left corner of first character
Position Y	Y-coordinate of upper left corner of first character
Font	See ***Static Text Object*** section for font types and sizes

Parameter	Description
Text Color	16-bit color for text
Background Color	16-bit color for text background

The following figure shows the preview output for each of the possible System Info Object reports. The actual format on the 3Dxx Color Display device may be slightly different. On the 3Dxx Simulator only the “App Version” and “App Built” reports are real, the other reports will show “n/a” or format place holders instead of real information.



```
P/N: 3DxxxxTTTT- N/A
S/N: YYWW-  N/A
App Version 3.00; Rev. D
App Built: 08/19/2015  01:29 PM
Free:    N/A    1K-blocks
Kernel version : Mmm dd yyy N/A h:mm:ss
U-boot version : Mmm dd yyy N/A h:mm:ss
Rootfs version : Mmm dd yyy N/A h:mm:ss
```

Figure 26: System Info Object Report Samples

Button Object (Model 3D2104 only)

The Button Object detects up or down button presses on the 3D2104 keypad. A Button Object must be assigned to 1 of the 7 keypad buttons on the 3D2104 display. Pressing the button causes a Button Down event, releasing the button causes a Button Up event. Button Up and Button Down events can be configured to either Display or Kill an object.

Table 46: Button Object Parameters (Model 3D2104 only)

Parameter	Description
Button Number	Keypad button 1 - 7
Button Down Kill Object	None, All, or Selected
Button Down Display Object	None or Selected
Button Up Kill Object	None, All, or Selected
Button Up Display Object	None or Selected

Appendix B. Memory Capacity and Usage

The 3Dxx Color Display allocates 0.5 Mbytes for object storage. Note that all graphics are stored as files in the FLASH file system and so only the name of the file takes up space in the object storage area.

The 3Dxx Color Display typically has approximately 1.5 GBytes of FLASH file system space available for storing graphics files.

This table shows some typical object storage requirements:

Table 47 - Memory Used by Various Object Types

Object Type	Size in Bytes
CAN Monitoring	
Bar Graph	46 for monochrome type or 96 for color segment type
Round Gauge	102 plus following possible add-ons: Image for background:18 + (1 byte per character in file name) * Image for pointer: 18 + (1 byte per character in file name) *
Custom Gauge	428
Dynamic Text	74
PGN Watchdog	20
SPN Range	38
SPN Value	26 + (6 bytes per object mapped)
Camera Monitoring	
Camera Input	28
Non-CAN Monitoring Objects	
Line	18
Box	18
Circle	16
Arc	20
Static Text	16 + (1 byte per character) *
Bitmap	16 + (1 byte per character in file name) *
Audio Output	6 + (1 byte per character in file name) *
Group Object	8 + (2 bytes per object in group)
Default Screen	10
Splash Screen	10
Touch Screen Event	36 + (1 byte per character in file name or text label if used) *
Backlight & Background	12
Buzzer	10
Date/Time	18
GPIO Input	22
GPIO Output	10
Analog Input	32
CAN Transmission	56
Timer	18
Variable	20
System Info	18
Button	16

* These items will be padded so that the object size will be even.

Appendix C. Troubleshooting

The 'Device' menu item is disabled (grayed out). The program requires the use of the 32 bit version of PCANBasic.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.

If the PCAN USB hardware is installed check these two things:

1. Verify that the PCANBasic.DLL is located in the same directory as the "VUI Builder.exe" file.
2. Verify that the PCAN software was installed and that the USB device driver installed itself when the PCAN USB device was connected.