



## ViSi-Genie Blocked User Buttons

DOCUMENT DATE: 7<sup>th</sup> May 2020  
DOCUMENT REVISION: 1.2



## Description

This application note provides a first hands-on example with ViSi-Genie and describes all the steps related to a project.

Before getting started, the following are required:

- Workshop 4 has been installed according to the document Workshop 4 Installation.
- The user is familiar with the Workshop 4 environment and with the fundamentals of ViSi-Genie, as described in Workshop 4 User Guide and ViSi-Genie User Guide.
- When downloading an application note, a list of recommended application notes is shown. It is assumed that the user has read or has a working knowledge of the topics discussed in these recommended application notes.

## Content

<b>Description .....</b>	<b>2</b>
<b>Content .....</b>	<b>2</b>
<b>Application Overview .....</b>	<b>3</b>
<b>Setup Procedure .....</b>	<b>5</b>
<b>Create a New Project .....</b>	<b>6</b>
<i>Create a New Project .....</i>	<i>6</i>
<b>Design of the Project .....</b>	<b>6</b>
<i>Adding User Buttons .....</i>	<i>6</i>
<i>Create a User Button Matrix .....</i>	<i>7</i>
<i>Create a Non-Momentary User Button .....</i>	<i>11</i>
<i>Create a Momentary button .....</i>	<i>12</i>
<i>Configure the OnChanged Event of the User Buttons .....</i>	<i>14</i>
<b>Build and Upload the Project .....</b>	<b>15</b>
<b>Identify the Messages .....</b>	<b>15</b>
<i>Use the GTX Tool to Analyse the Messages .....</i>	<i>15</i>
Launch the GTX Tool .....	15
<i>The Blocked Momentary User Button .....</i>	<i>16</i>
Change the Block of a Momentary User Button .....	17
<i>The Non-momentary Blocked User Button .....</i>	<i>17</i>
Change the Block of a Non-momentary User Button .....	18
<i>The LSB Value: - Momentary vs. Non-momentary Blocked User Buttons .....</i>	<i>18</i>

<i>Set the Blocks</i> .....	18
Change the Block of the User Buttons to Block 0	18
Change the Block of the User Buttons to Block 1	19
Change the Block of the User Buttons to Block 2	20
Change the Block of the User Buttons to Block 3	21
<b>Proprietary Information</b> .....	<b>23</b>
<b>Disclaimer of Warranties &amp; Limitation of Liability</b> .....	<b>23</b>

## Application Overview

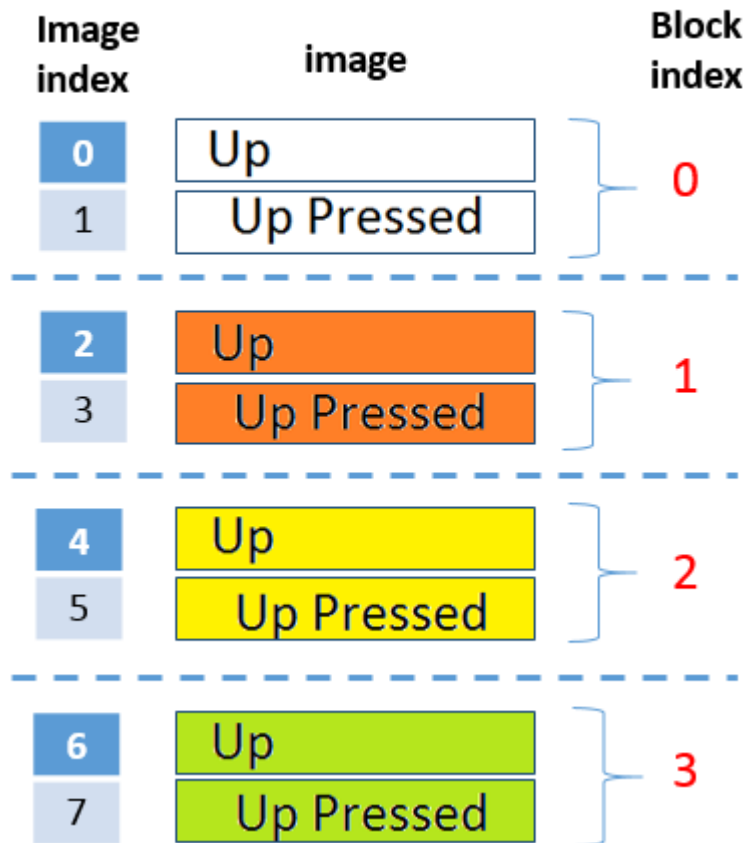
In [ViSi-Genie User Buttons](#), the use of ordinary or non-blocked user buttons was discussed. This application note, on the other hand, introduces the concept of "blocking" or of blocked user buttons. Blocking allows for user buttons to have multiple sets, anyone of which can be activated by the host during runtime. An ordinary or non-blocked momentary user button, for instance, has two states - "Up" and "Up Pressed".

### A non-blocked momentary user button

Image index	image
0	Up
1	Up Pressed

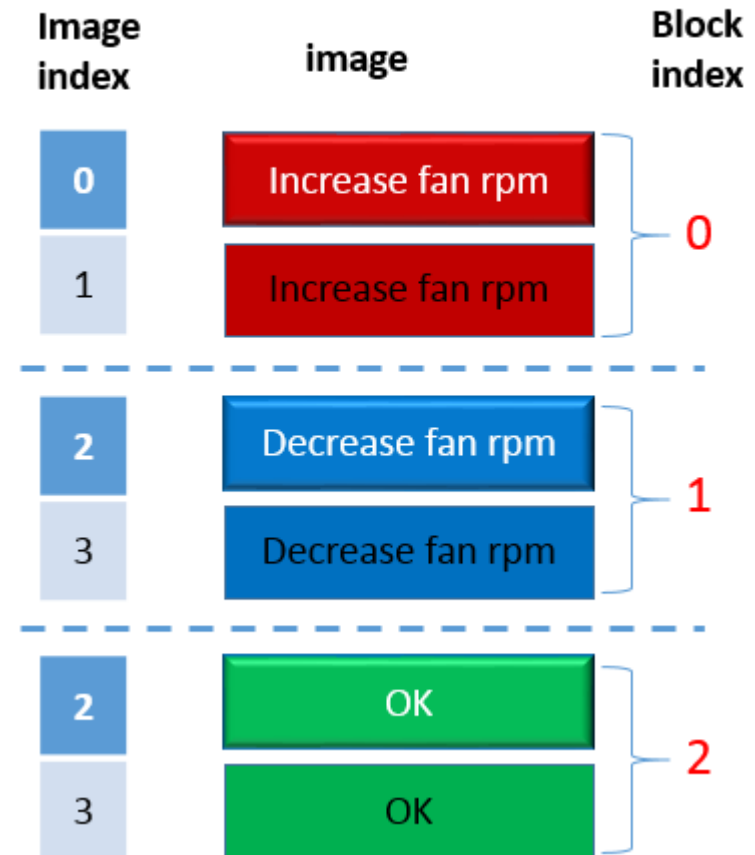
A blocked momentary user button, on the other hand, can have multiple blocks, wherein each block has two states. Shown below is an example of a momentary user button with four blocks.

**A blocked momentary user button**



As shown above, the blocks are indexed from 0 to 3. Anyone of these blocks can be activated by the host using the appropriate commands. The images for the states of different blocks may not need to be similar even. To illustrate,

**A momentary user button w/ 3 blocks**



A host with a temperature sensor and a GUI for controlling the speed of rotation of a fan can display block 0 if the temperature is too high, block 1 if the temperature is too low, and block 2 if the temperature is within range.

For non-momentary user buttons, blocking is applied as shown below.

Image index	image	Block index
0	Up	0
1	Up Pressed	
2	Down	
3	Down Pressed	
4	Up	1
5	Up Pressed	
6	Down	
7	Down Pressed	
8	Up	2
9	Up Pressed	
10	Down	
11	Down Pressed	
12	Up	3
13	Up Pressed	
14	Down	
15	Down Pressed	

The section “**Identify the Messages**” discusses the format of the messages coming from and going to a user button using the GTX Tool in Workshop. This section also discusses the format of the message for changing the block of a blocked user button. An understanding of this section is essential for users who intend to interface the display to an external host.

### Setup Procedure

For instructions on how to launch Workshop 4, how to open a ViSi-Genie project, and how to change the target display, kindly refer to the section “**Setup Procedure**” of the application note:

[ViSi Genie Getting Started – First Project for Picaso Displays](#) (for Picaso)

or

[ViSi Genie Getting Started – First Project for Diablo16 Displays](#) (for Diablo16).

## Create a New Project

### Create a New Project

For instructions on how to create a new ViSi-Genie project, please refer to the section “**Create a New Project**” of the application note

[ViSi Genie Getting Started – First Project for Picaso Displays](#) (for Picaso)

or

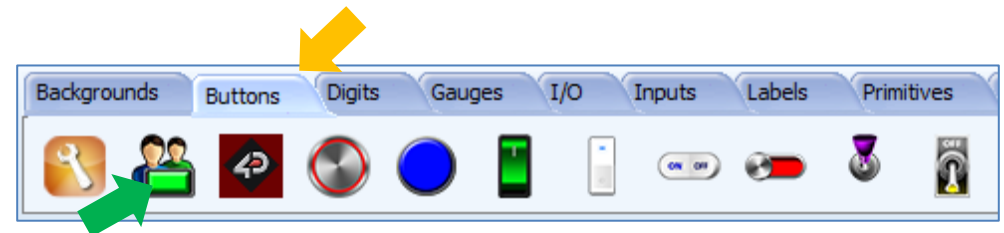
[ViSi Genie Getting Started – First Project for Diablo16 Displays](#) (for Diablo16)

## Design of the Project

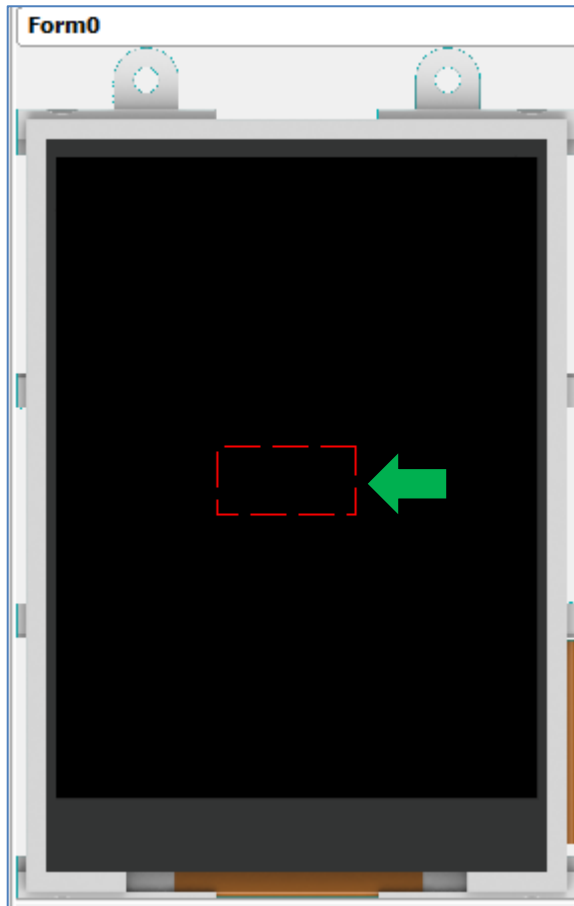
### Adding User Buttons

For an in-depth discussion about user buttons please refer to the application note [ViSi-Genie User Buttons](#). It is discussed here how a user button can be configured to behave as a momentary button, a toggle button, or one among a group of buttons. Each of these configurations has its own application. The application note also discusses the format of the messages coming from and going to a non-blocked user button using the GTX Tool in Workshop.

To add a user button, go to the **Buttons** pane then click on the **user button** icon.




Click on the **WYSIWYG** (What-You-See-Is-What-You-Get) screen to put the object in place. The WYSIWYG screen simulates the actual appearance of the display module screen.

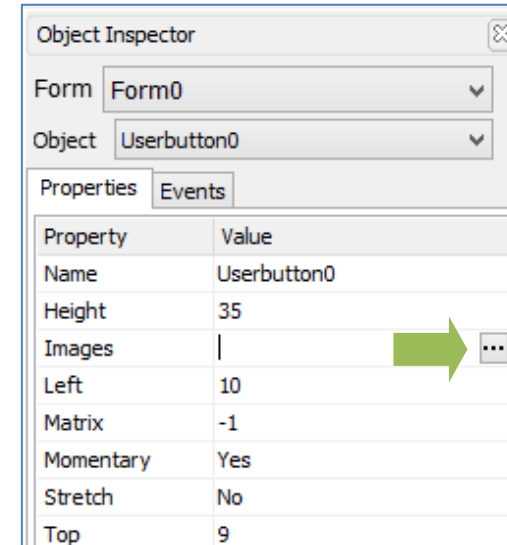


The object can be dragged to any desired location and resized to the desired dimensions. The **Object Inspector** on the right part of the screen displays all

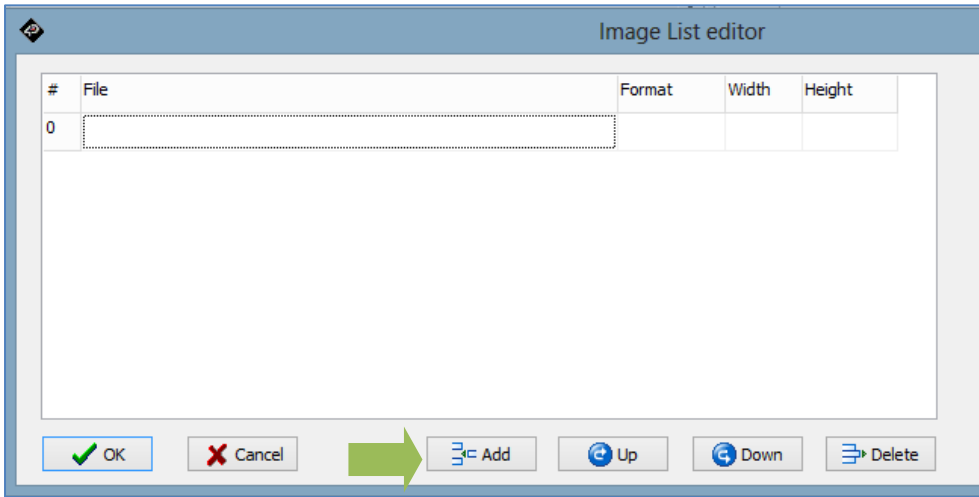
the properties of the newly created user button object named **Userbutton0**. In this application, there are a total of five user buttons, namely Userbutton0, Userbutton1, Userbutton2, Userbutton3, and Userbutton4. The first three user buttons are part of a button matrix, Userbutton3 is a non-momentary user button, and Userbutton4 is a momentary (both) button.

### Create a User Button Matrix

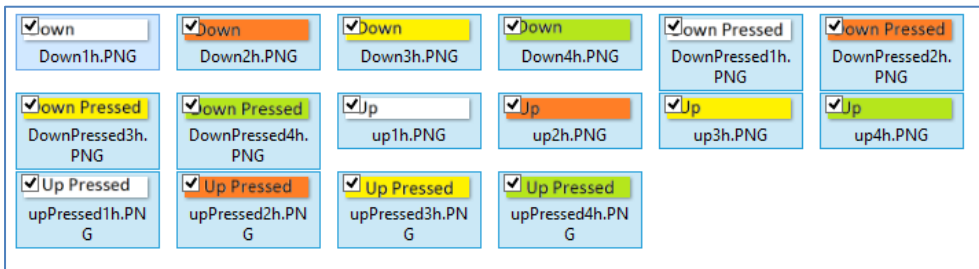
Add three user button objects to Form0. These are Userbutton0, Userbutton1, and Userbutton2. Add images to each of the buttons by using the 'Image List editor'. Click on the button  located at the right of the 'Images' property.



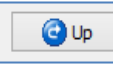

Click on 'Add' to open the file explorer.

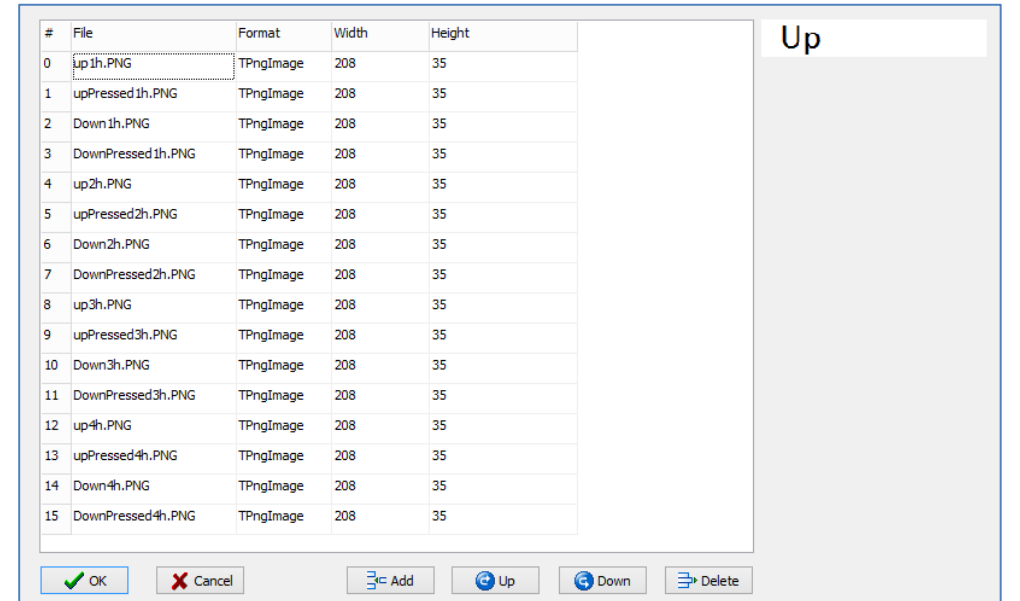


In the file browser, select all of the images inside the folder 'BlockedUserButtons.ImgData'.



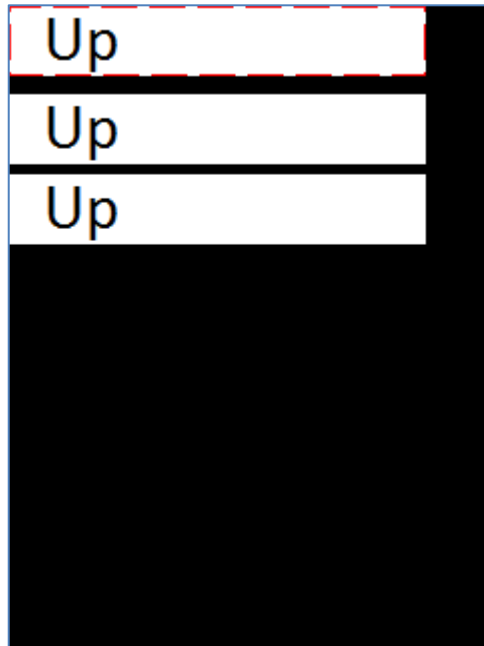
The number of images required is 16 to create 4 Blocks. This is due to the number of states of a non-momentary button.

Arrange the files according to the image below. The   buttons can be used to change the arrangement of the files added. This is necessary for the proper groupings of the image. Each colour represents a block.

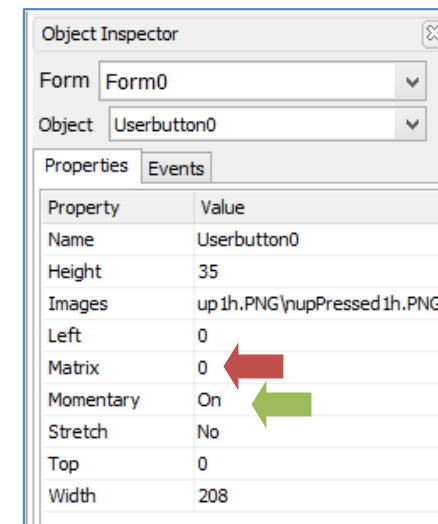




After adding all of the needed images, the three user buttons will now look like the image below. Block 0 represents white coloured buttons.

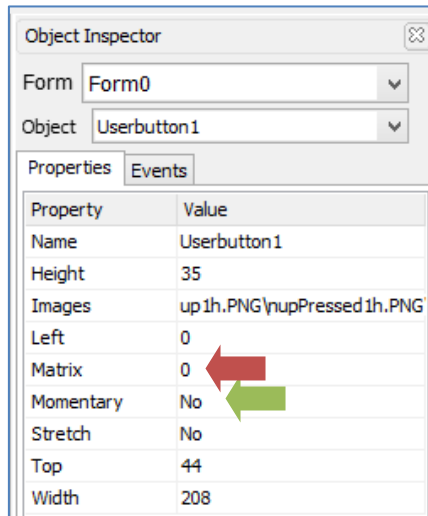


Configure the properties of the three user buttons according to the images below. To group these buttons into a matrix, edit the Matrix property in the object inspector. It is important that all buttons grouped share the same matrix number, otherwise pressing on one button will not release the other buttons of the group.

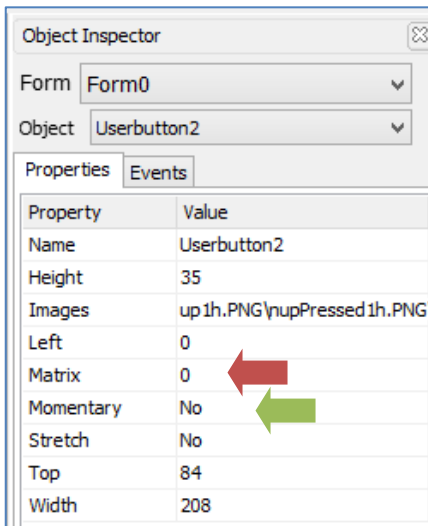


Userbutton0 is set to momentary 'On' so that its initial state is the fourth state or the 'Down Pressed' image.

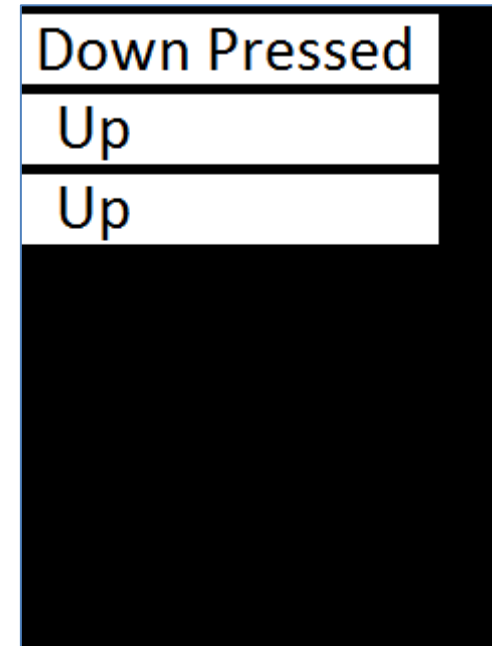
Properties for UserButton1:




Properties for Userbutton2:

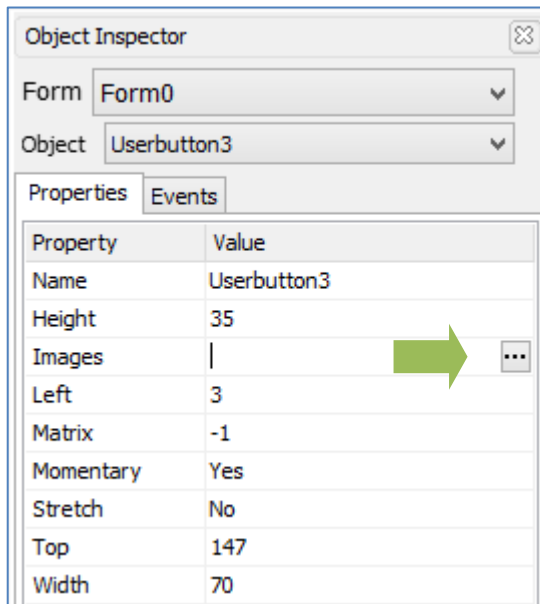


After changing the properties the output should now look like the image shown below.



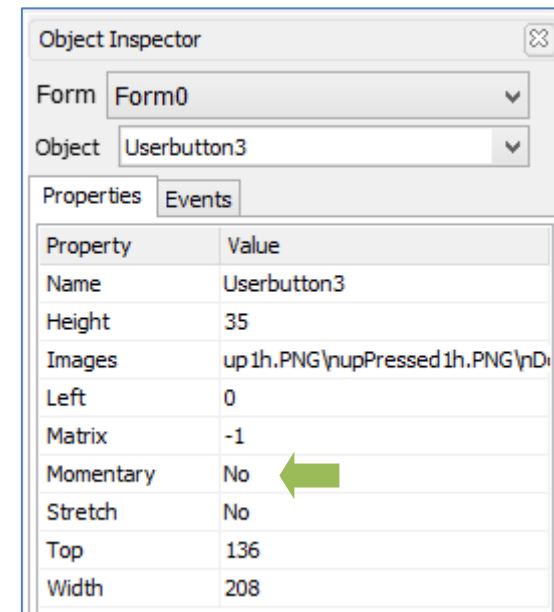
### Create a Non-Momentary User Button

Add another user button object to Form0. This is Userbutton3. Add images in the 'Image List editor' by clicking on  located at the right of the 'Images' property. Add the same images added to each of the matrix user buttons.

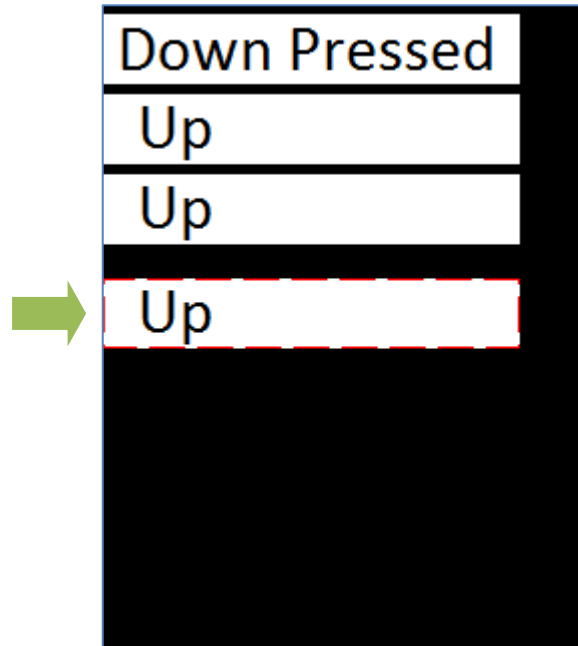


Since Userbutton3 is a non-momentary user button, it has the same number of states with that of a user button in a matrix (Userbutton0, Userbutton1, and Userbutton2). Thus, the images that are to be added to Userbutton3 are also the same images added to the matrix buttons Userbutton0, Userbutton1, and Userbutton2.


Set the 'Momentary' property of Userbutton3 to 'No'.

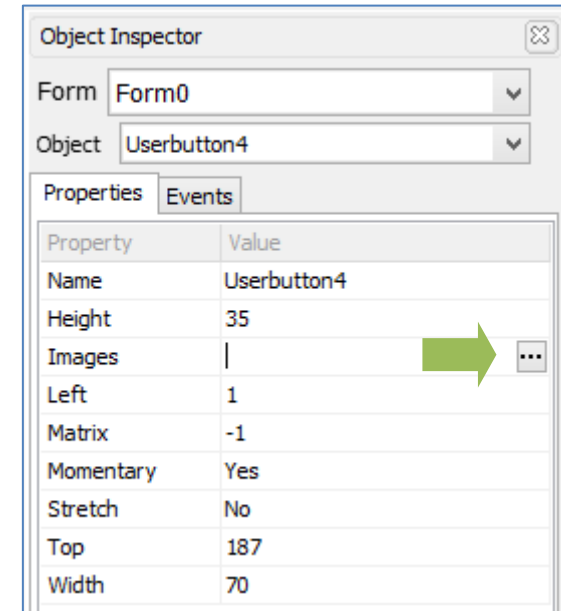


After changing the properties the output should now look like the image shown below.

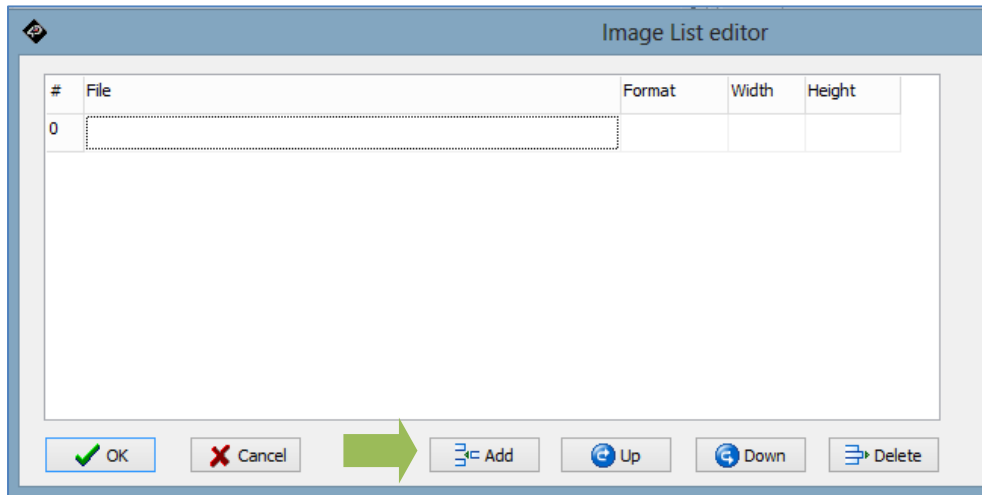


### Create a Momentary button

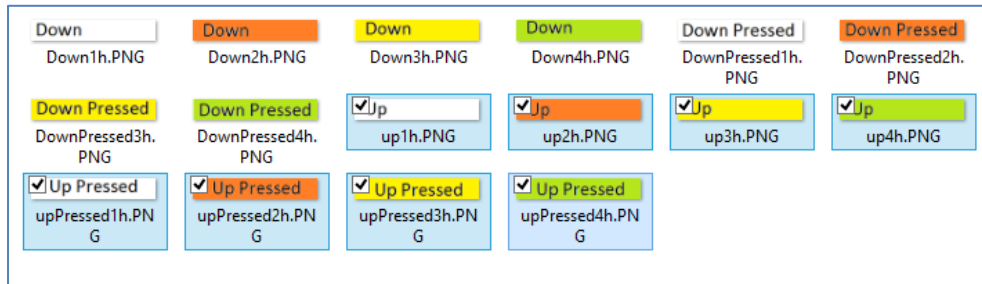
Add another user button object to Form0. This is Userbutton4. Add images in the 'Image List editor' by clicking on  located at the right of 'Images' property.



Click on 'Add' to open browser/file explorer.

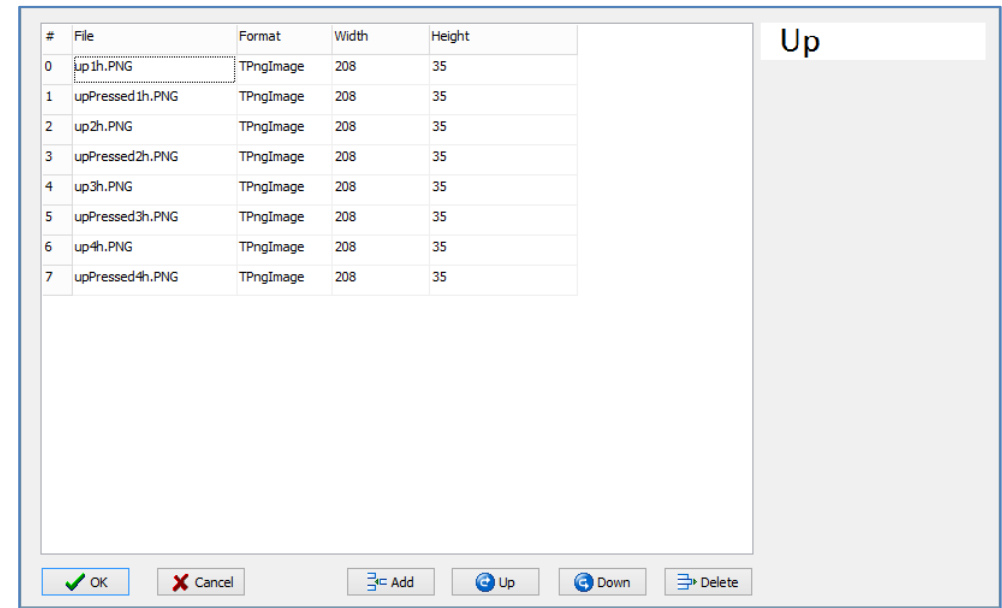


In the browser, inside the folder 'BlockedUserButtons.ImgData', select all of the images that contains 'up'.

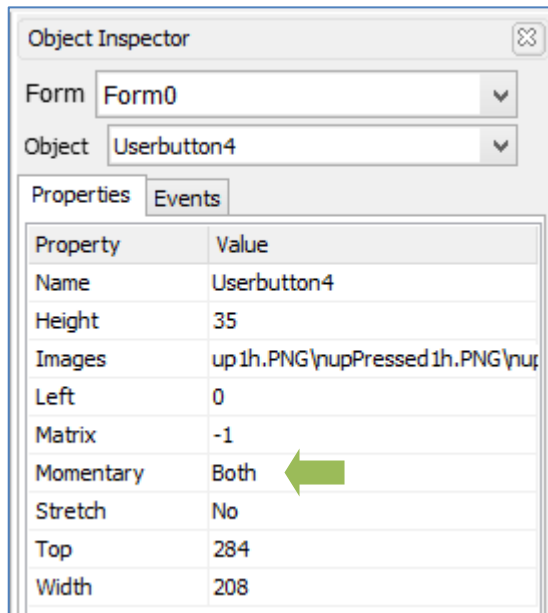


This time the number of images required is 8 to create 4 Blocks. This is due to the number of states of a Momentary Button.

Arrange the files according to the image below. The Up and Down buttons can be used to change the arrangement of the files added.

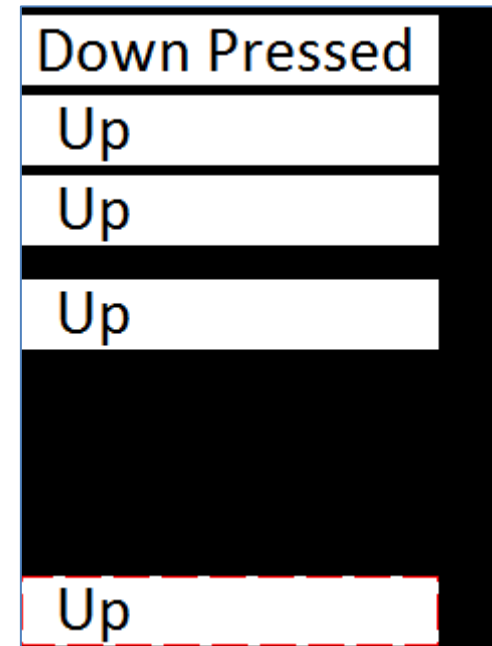


Set the 'Momentary' property of Userbutton4 to 'Both'.



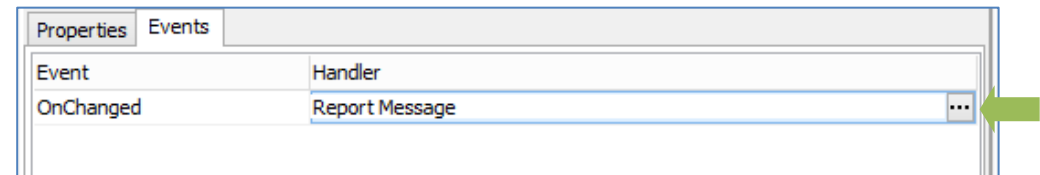
**Note:** Momentary 'Both' and 'Yes' are different in terms of how report messages occur. A momentary 'Both' reports a value of 1 if pressed, and if released it reports a value of '0'. A momentary 'Yes' reports a value of 1 if pressed and reports nothing if released.

The final output of the Genie program is shown below.



### Configure the OnChanged Event of the User Buttons

A user button can be configured to report a message to an external host controller when its status has changed. Go to its object inspector and configure the **onChanged** event as shown below. Do this for all of the user buttons.



## Build and Upload the Project

For instructions on how to build and upload a ViSi-Genie project to the target display, please refer to the section “**Build and Upload the Project**” of the application note

[ViSi Genie Getting Started – First Project for Picaso Displays](#) (for Picaso)

or

[ViSi Genie Getting Started – First Project for Diablo16 Displays](#) (for Diablo16).

The uLCD-32PTU and/or the uLCD-35DT display modules are commonly used as examples, but the procedure is the same for other displays.

## Identify the Messages

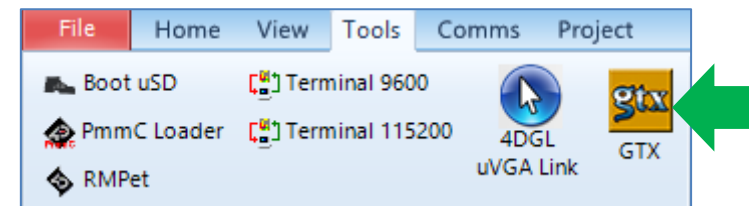
The display module is going to receive and send messages from and to an external host. This section explains to the user how to interpret these messages. An understanding of this section is necessary for users who intend to interface the display to a host. The [ViSi Genie Reference Manual](#) is recommended for advanced users.

### Use the GTX Tool to Analyse the Messages

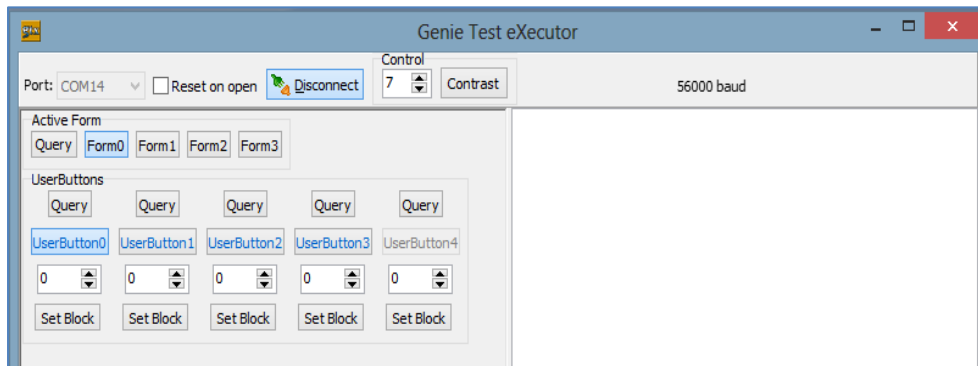
Using the GTX or **Genie Test eXecutor** tool is the first option to get the messages sent by the screen to the host. Here the PC will be the host. The GTX tool is a part of the Workshop 4 IDE. It allows the user to receive, observe, and send messages from and to the display module. It is an essential debugging tool.

### Launch the GTX Tool

Under the Tools menu click on the GTX tool button.



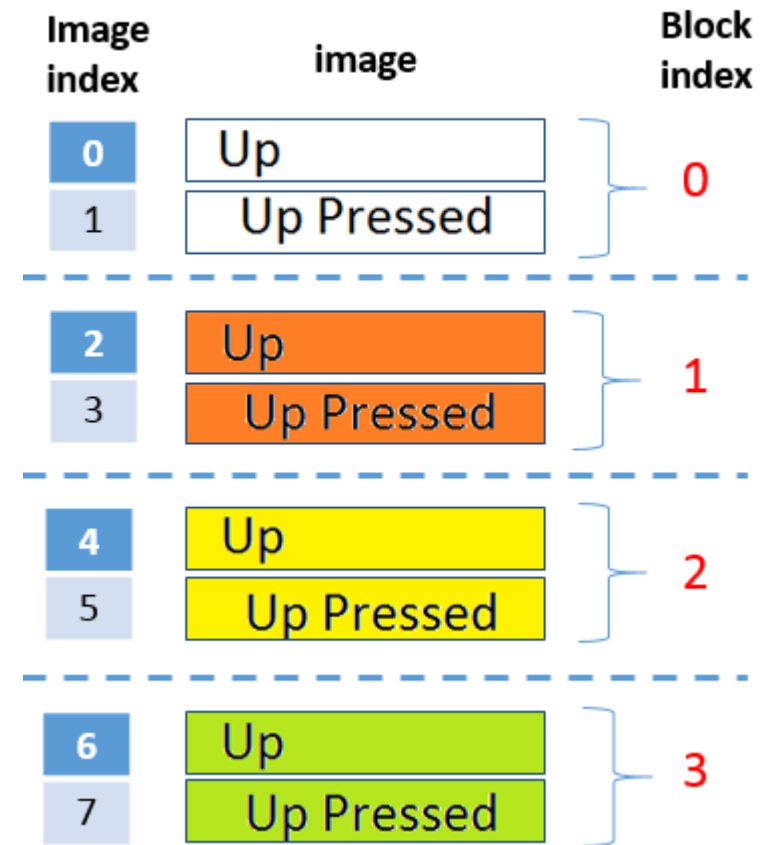
The Genie Test eXecutor window appears.



### The Blocked Momentary User Button

Momentary user buttons have two states. Image 0 or 'Up' is the 'released' state and image 1 or 'Up Pressed' is the 'pressed' or 'touched' state. A momentary user button initially starts at image 0 and goes to image 1 if pressed. If it is released it will go back again to image 0. To create N Blocks, it is needed to have N\*2 images.

### A blocked momentary user button





### Change the Block of a Momentary User Button

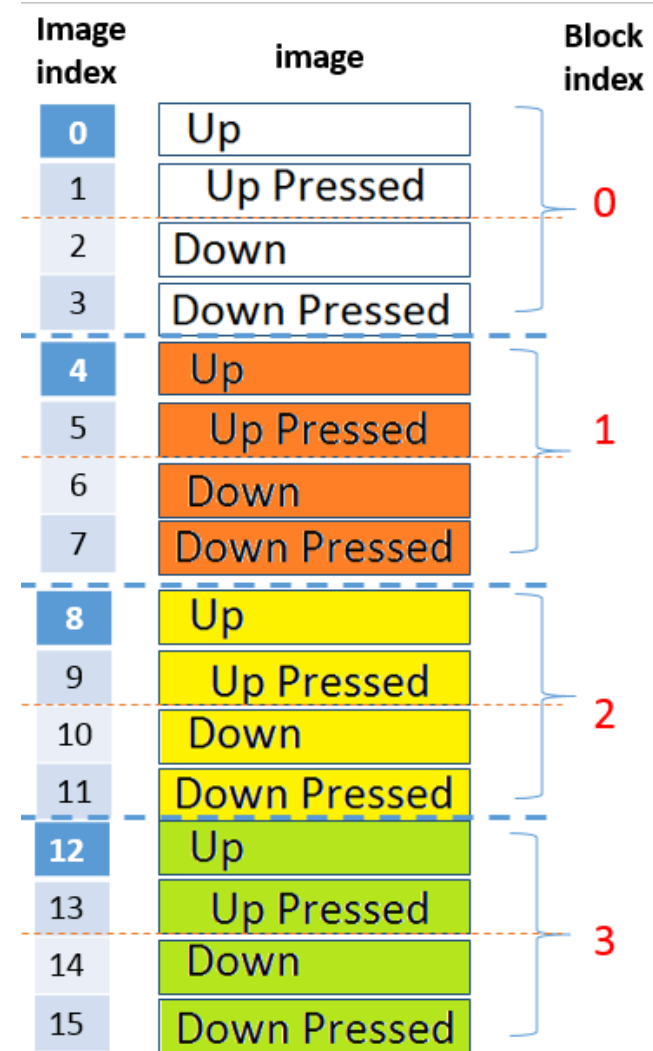
To be able to change the block, it is needed to set the block by writing to the user button. For example, to change the block of a momentary user button to block 3:

Command	Object Type	Object Index	Value MSB	Value LSB	Checksum
0x01	0x21	0x04	0x80	0x03	0xA7
WRITE_OBJ	User button	Fourth	0x8003		

When writing to a user button to change blocks, the MSB (most significant byte) has a constant value of 0x80 always. To change the block of a momentary user button to block 'n', the LSB (least significant byte) value is equal to the block index 'n' itself.

### The Non-momentary Blocked User Button

Non-momentary user buttons have four states. Image 0 or 'Up' and image 2 or 'Down' are the 'released' states. Image 1 or 'Up Pressed' and image 3 or 'Down pressed' are the 'pressed' or 'touched' states. A non-momentary user button initially starts at image 0 and if pressed it will go to image 1. If it is released it will go to Image 2 and to image 3 if pressed again. The cycle repeats itself. To create N Blocks, it is needed to have N\*4 images.



### Change the Block of a Non-momentary User Button

To change the block of a non-momentary user button to block 3, the correct message is:

Command	Object Type	Object Index	Value MSB	Value LSB	Checksum
0x01	0x21	0x01	0x80	0x06	0xA7
WRITE_OBJ	User button	Second	0x8006		

To change the block of a non-momentary user button to block 'n', the LSB (least significant byte) value is equal to the block index 'n' multiplied by 2.

### The LSB Value: - Momentary vs. Non-momentary Blocked User Buttons

Again, when writing to a user button to change blocks, the MSB (most significant byte) has a constant value of 0x80 always. To change the block of a non-momentary user button to block 'n', the LSB (least significant byte) value is equal to the block index 'n' multiplied by 2. For momentary user buttons, the LSB value is equal to the block index 'n' itself. To further illustrate,

For non-momentary user buttons:

$$LSB\ value = n \times 2$$

For momentary user buttons:

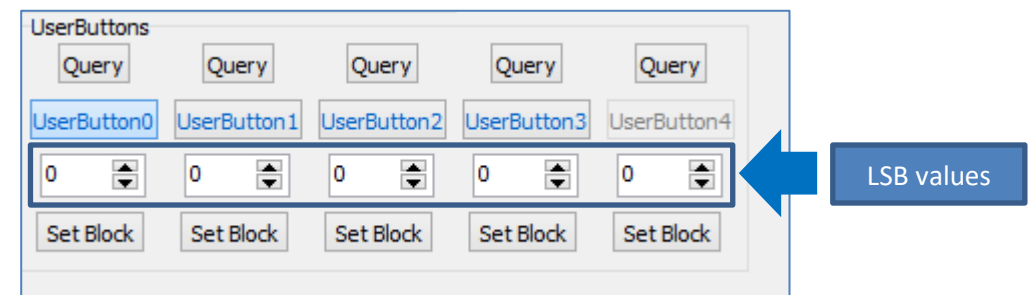
$$LSB\ value = n$$

Where 'n' is the index of the desired block of destination.

### Set the Blocks

#### Change the Block of the User Buttons to Block 0

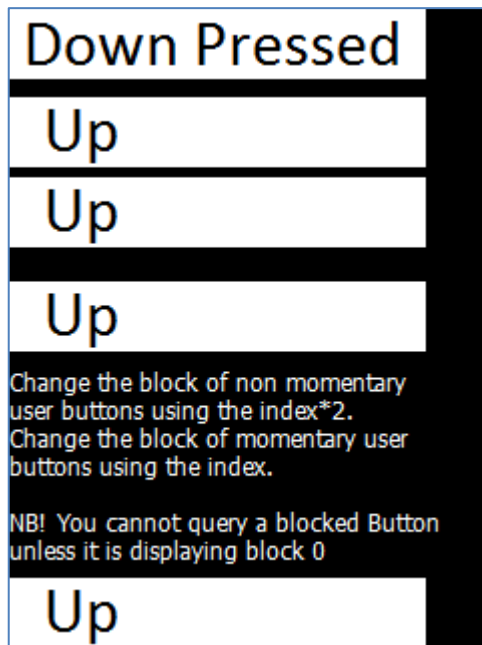
Set all of the LSB values to 0 and click all of the 'Set Block' buttons.



The image below shows the reported messages after changing the block.

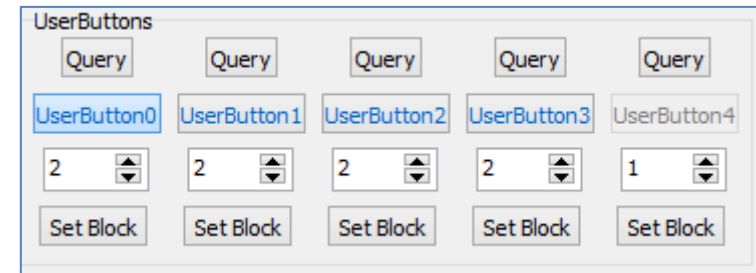
```
Set UserButton Value 15:01:38.366 [01 21 00 80 00 A0]
ACK 15:01:38.438 [06]
Set UserButton Value 15:01:39.046 [01 21 01 80 00 A1]
ACK 15:01:39.095 [06]
Set UserButton Value 15:01:39.580 [01 21 02 80 00 A2]
ACK 15:01:39.620 [06]
Set UserButton Value 15:01:39.993 [01 21 03 80 00 A3]
ACK 15:01:40.025 [06]
Set UserButton Value 15:01:40.742 [01 21 04 80 00 A4]
ACK 15:01:40.778 [06]
```

Block 0 represents white coloured buttons.



### Change the Block of the User Buttons to Block 1

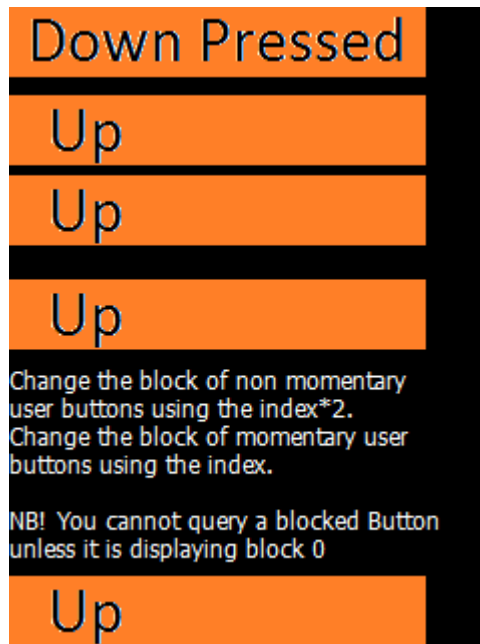
Set all of the LSB values to 2 for the non-momentary buttons and 1 for the momentary button then click 'Set Block'.



The image below shows the reported messages after changing the block.

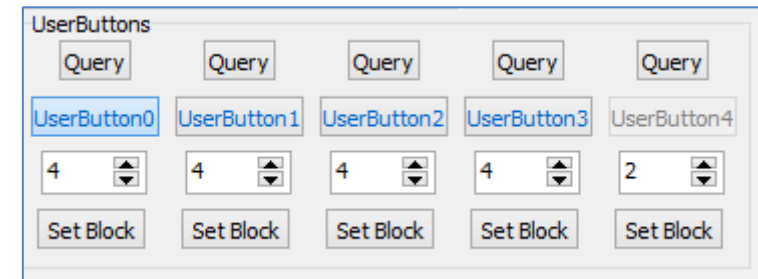
```
Set UserButton Value 15:02:45.176 [01 21 00 80 02 A2]
ACK 15:02:45.234 [06]
Set UserButton Value 15:02:46.998 [01 21 01 80 02 A3]
ACK 15:02:47.044 [06]
Set UserButton Value 15:02:48.308 [01 21 02 80 02 A0]
ACK 15:02:48.353 [06]
Set UserButton Value 15:02:49.886 [01 21 03 80 02 A1]
ACK 15:02:49.918 [06]
Set UserButton Value 15:02:51.041 [01 21 04 80 01 A5]
ACK 15:02:51.074 [06]
```

Block 1 represents orange coloured buttons.



[Change the Block of the User Buttons to Block 2](#)

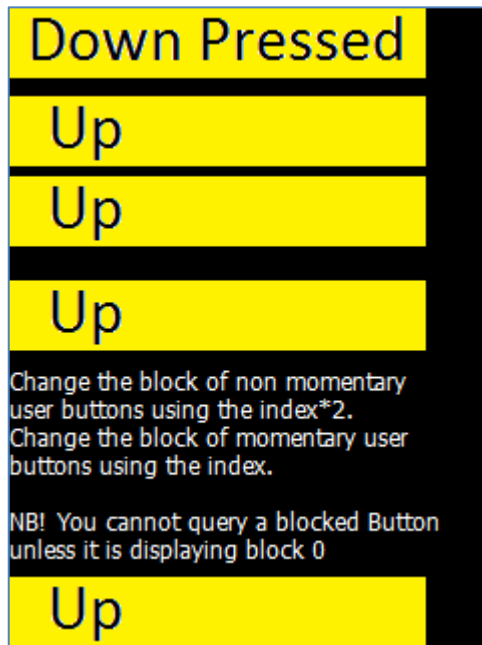
Set all of the LSB values to 4 for the non-momentary buttons and 2 for the momentary button then click 'Set Block'.



The image below shows the reported messages after changing the block.

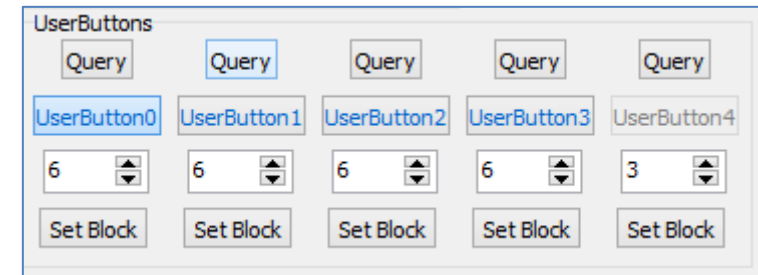
```
Set UserButton Value 15:04:16.528 [01 21 00 80 04 A4]
ACK 15:04:16.573 [06]
Set UserButton Value 15:04:17.401 [01 21 01 80 04 A5]
ACK 15:04:17.448 [06]
Set UserButton Value 15:04:17.971 [01 21 02 80 04 A6]
ACK 15:04:18.012 [06]
Set UserButton Value 15:04:18.736 [01 21 03 80 04 A7]
ACK 15:04:18.763 [06]
Set UserButton Value 15:04:19.282 [01 21 04 80 02 A6]
ACK 15:04:19.323 [06]
```

Block 2 represents yellow coloured buttons.



### Change the Block of the User Buttons to Block 3

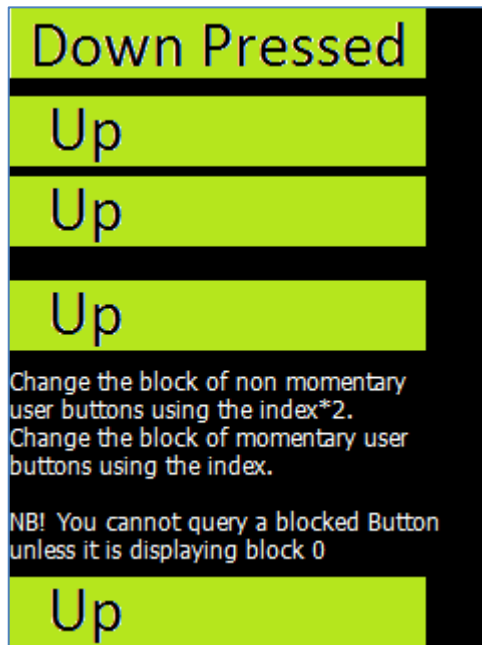
Set all of the LSB values to 6 for the non-momentary buttons and 3 for the momentary button then click 'Set Block'.



The image below shows the reported messages after changing the block.

```
Set UserButton Value 15:21:42.615 [01 21 00 80 06 A6]
ACK 15:21:42.653 [06]
Set UserButton Value 15:21:43.150 [01 21 01 80 06 A7]
ACK 15:21:43.181 [06]
Set UserButton Value 15:21:43.706 [01 21 02 80 06 A4]
ACK 15:21:43.744 [06]
Set UserButton Value 15:21:44.118 [01 21 03 80 06 A5]
ACK 15:21:44.154 [06]
Set UserButton Value 15:21:44.552 [01 21 04 80 03 A7]
ACK 15:21:44.590 [06]
```

Block 3 represents green coloured buttons.



## Proprietary Information

The information contained in this document is the property of 4D Systems Pty. Ltd. and may be the subject of patents pending or granted, and must not be copied or disclosed without prior written permission.

4D Systems endeavours to ensure that the information in this document is correct and fairly stated but does not accept liability for any error or omission. The development of 4D Systems products and services is continuous and published information may not be up to date. It is important to check the current position with 4D Systems.

All trademarks belong to their respective owners and are recognised and acknowledged.

## Disclaimer of Warranties & Limitation of Liability

4D Systems makes no warranty, either expresses or implied with respect to any product, and specifically disclaims all other warranties, including, without limitation, warranties for merchantability, non-infringement and fitness for any particular purpose.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications.

In no event shall 4D Systems be liable to the buyer or to any third party for any indirect, incidental, special, consequential, punitive or exemplary damages (including without limitation lost profits, lost savings, or loss of business opportunity) arising out of or relating to any product or service provided or to be provided by 4D Systems, or the use or inability to use the same, even if 4D Systems has been advised of the possibility of such damages.

4D Systems products are not fault tolerant nor designed, manufactured or intended for use or resale as on line control equipment in hazardous environments requiring fail – safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines or weapons systems in which the failure of the product could lead directly to death, personal injury or severe physical or environmental damage ('High Risk Activities'). 4D Systems and its suppliers specifically disclaim any expressed or implied warranty of fitness for High Risk Activities.

Use of 4D Systems' products and devices in 'High Risk Activities' and in any other application is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless 4D Systems from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any 4D Systems intellectual property rights.