# Table of Contents

1. Description ........................................................................................................................................... 4
2. Application Overview ............................................................................................................................... 4
3. Setup Procedure .................................................................................................................................... 4
4. Launch Workshop4................................................................................................................................. 5
5. Create a New Project .............................................................................................................................. 6
6. Select ViSi ............................................................................................................................................... 10
7. The Main Screen .................................................................................................................................. 11
   7.1. Area 1: Menus ................................................................................................................................. 12
   7.2. Area 2: Ribbon with Icons ............................................................................................................. 12
   7.3. Area 3: Code .................................................................................................................................. 12
   7.4. Area 4: Form and WYSIWYG Screen ........................................................................................... 13
   7.5. Area 5: Object Inspector ................................................................................................................ 13
   7.6. Area 6: Message Window ............................................................................................................... 14
   7.7. Move the Form and Object Inspector ........................................................................................... 14
8. A First ViSi Project ................................................................................................................................ 16
   8.1. Adding Graphical Objects .............................................................................................................. 16
   8.2. Customizing Graphical Objects ...................................................................................................... 17
   8.3. Changing Between Object Properties ............................................................................................ 18
   8.4. Inserting a Layout into the User Application Program ............................................................... 19
   8.5. Finalizing a Program for Download ............................................................................................. 20
   8.6. Downloading an Application ......................................................................................................... 20
   8.7. Example Application Program ...................................................................................................... 20
9. Objects .................................................................................................................................................... 22
   9.1. Buttons Object ............................................................................................................................... 22
   9.2. Digits Objects ............................................................................................................................... 24
   9.3. Gauges Objects .............................................................................................................................. 25
   9.4. Primitive Objects ........................................................................................................................... 28
   9.5. Input Objects ............................................................................................................................... 30
   9.6. Label Objects ................................................................................................................................ 35
   9.7. System/Media Objects .................................................................................................................... 37
   9.8. Backgrounds .................................................................................................................................. 39
   9.9. Selection Tool ................................................................................................................................ 40
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>10. Connect the Screen</td>
<td>41</td>
</tr>
<tr>
<td>11. Insert the Micro-SD Card</td>
<td>43</td>
</tr>
<tr>
<td>12. Communication Terminal</td>
<td>44</td>
</tr>
<tr>
<td>13. Application Notes</td>
<td>48</td>
</tr>
<tr>
<td>14. Revision History</td>
<td>49</td>
</tr>
<tr>
<td>15. Legal Notice</td>
<td>50</td>
</tr>
<tr>
<td>16. Contact Information</td>
<td>50</td>
</tr>
</tbody>
</table>
1. Description

This manual is dedicated to explaining the ViSi Software Tool, which is part of the Workshop4 IDE suite.

An overview of its basic functionality and uses will be explored. In order to use ViSi, the following items are required:

- Any 4D Systems Display Module which uses a 4D Labs processor
- 4D Programming Cable or Adaptor
- A micro-SD memory card
- 4D Labs Workshop4 IDE

2. Application Overview

It is often difficult to design a graphical display without being able to see the immediate results of the application code. ViSi is the perfect software tool that allows the user to see the instant results of their desired graphical layout.

Additionally, there is a selection of inbuilt dials, gauges and meters that can simply be dragged and dropped onto the simulated module display. From here, each can have properties edited and at the click of a button, all relevant code is produced in the user program. Each feature of ViSi will be outlined with examples below.

3. Setup Procedure

The installer can be downloaded on its product page from 4D Labs website. Using the web browser of your choice and go to www.4dlabs.com.au.
4. Launch Workshop4

There is an alias for 4D Workshop on the desktop:

Launch 4D Workshop by double-clicking on the icon:
5. Create a New Project

Workshop4 opens and displays the Recent page:

To create a new program, there are multiple options:

- Click on the top left-most icon New
• Click on the icon close to **Create a new 4D Systems Project** on top
• Click on the **Create a new Project** to create a project instance based on your last project settings.

Note: Another option is displayed on the image above. **Create a new 4D Labs Project** is an upcoming Workshop4 feature which allows to create a project with a customized display module powered by 4D Labs’ processors.

Both of the first two options update the main window with the selection of the screen:
Select the screen, here the gen4-uLCD-32DT

The selected screen is displayed:
Orientation is portrait by default. To set it to landscape, just click on the image of the screen to rotate it:

Press **Next** to proceed:
6. Select ViSi

The main window now asks for the kind of project:

To select ViSi, just click on the blue arrow. The development environment is now displayed:
7. The Main Screen

Let’s discuss the different areas. There are six different areas, from left to right, for top to bottom:

1. Menus
2. Ribbon with icons
3. Code
4. Form and WYSIWYG screen where to place the objects
5. Object inspector, where properties and events are defined
6. Messages about errors, warnings and notices
7.1. Area 1: Menus

The menus include standard Windows options. Each menu displays a specific ribbon.

7.2. Area 2: Ribbon with Icons

For the Home menu, the ribbon includes file-related, editing, project compile and download, and other buttons grouped in panes:

The icons related to the project files include New project, Open project, Save project, Save as project, Print project, and Build project.

The Compile button compiles the project while the Comp’nLoad button compiles and uploads it to the processor.

7.3. Area 3: Code

On top of the code, the open projects are displayed:

Click on the tab to open a project or on the cross to close it.
7.4. Area 4: Form and WYSIWYG Screen


The active form is displayed there, with its objects. Objects are picked from the panes and can be resized and moved.

Click on an object to select it.

7.5. Area 5: Object Inspector

The object inspector provides all the information on the selected object:

- properties, as size and position;
- and events, where actions are defined.
7.6. Area 6: Message Window

The message window displays errors, warnings and notices after the project is built.

Before starting using the Workshop4, we need to connect the screen and prepare a micro-SD card.

7.7. Move the Form and Object Inspector

The Form and Object Inspector can be moved for a customised interface.

Click on the title bar of the form...

...to activate it...

...then click-and-drag to place the form on the desired place...
...and finally release the button.

In case the Form and Object Inspector are hidden, just click on the relevant button on the **View** menu to display them again:
8. A First ViSi Project

8.1. Adding Graphical Objects

Become familiar with the ViSi layout by cycling through the tabs in the Widgets menu. Each tab contains a different category of graphical objects.

Try placing a meter gauge by selecting the Gauges tab and then click on the first dial icon, followed by clicking anywhere on the display area. The meter will be placed in the area, which is now ready for positioning and customization if necessary.

The following snapshot shows the gauge positioned at the top of the display area with editable properties below.
8.2. Customizing Graphical Objects

Now that the object has been placed, it can be fully customized to the desired layout by the user. Each object can be edited in one of two ways; either interactively on the display, or manually using the various fields underneath the display.

Take some time to experiment with the properties available to change the way an object appears on the screen. The easiest approach to achieving the optimal layout is by adjusting the size and position of the object directly on the display by moving the cursor over each entity. It should be noted that the background display or overall binding entity is referred to as Form1.

Form1 can be edited by selecting it from the dropdown box in the Object Inspector. Once happy, move onto changing features such as colours, fonts, captions,... in the property fields below.

Refer to the next image, which demonstrates this concept.

There is no Z-order in ViSi. Objects are always drawn on the display ‘background’. The display background is the Form background (colour or image) with the ‘Backgrounds’ objects ‘added’. For objects that are transparent, or have a transparent aspect (eg corners of WinButtons) this means what shows through the transparent region will be the Form background + ‘Backgrounds’, not what may appear underneath when viewing in Workshop. Any visual effects that need to be ‘underneath’ objects should be included in a form background image.
8.3. Changing Between Object Properties

If there are multiple graphical objects on the screen, it may be necessary to change between object properties. Do this by using the dropdown box located at the top of the **Object Inspector**. Alternatively, each object can be edited by clicking on the object in the module display area.
8.4. Inserting a Layout into the User Application Program

When an image layout is finalised, it is ready to have the relevant code behind each object inserted into the user program. From inside the application program, find the desired place to insert the image created using ViSi.

There are two possible options for inserting code into the application.

The user can:
- either paste for an individual object;
- or paste all the code in one place for all of the objects created on the screen.

Pasting all of the code can only be done by selecting Form1. All other objects only have the ability to generate code for themselves.

The following images illustrate how to paste code into the user application. Select Form1, then click Paste all Code just below it. Notice all of the code is inserted into the program:
Often, not all code will be needed in the one place in a given user program. For this reason, it will be necessary to paste code for a particular object only.

The next image demonstrates this point. Select an object –here, **Gauge1**– then click on **Paste Code**. Notice that only a couple of lines of code are inserted into the program.

```
// Meter1 1.0 generated 4/23/2017 10:50:34 AM
img_setword(hwd1, [meter], IMAGE_INDEX, frame); // where frame is 0 to 100 (for a displayed 0 to 100)
img_draw(hwd1, [meter1]);
```

8.5. Finalizing a Program for Download

Rounding off a project requires the user to develop the necessary code to control the flow of how the application runs. This will involve writing control loops and sequences in between all of the image screens that have been created using ViSi.

It is now apparent, that this method of application development has now significantly reduced in time, since all image layouts are generated by ViSi, which normally the user would have to do.

Not only is time saved in this respect, but also code will no longer be required to be downloaded onto the module every time a visual change is made. Instant graphical changes are now seen in the simulated display, which means an application download will only need to be done once; at the completion of a user program.

8.6. Downloading an Application

When a ViSi program is compiled, a .gci and .dat file is generated automatically without going through the Graphics Composer software tool. Make sure that the module is plugged into the PC and the COM port is selected. A prompt will appear requiring that the .gci and .dat files are copied onto the micro-SD card before proceeding.

Before clicking OK, Insert the micro-SD card into the PC and locate the .gci and .dat file created during compile.

These will be located in the same folder as the project file. Copy these to the micro-SD card and insert the card back into the module and click OK. The application will finish downloading to the module and the application will now run.

8.7. Example Application Program

The following example was developed by the 4D Team to illustrate some of the features and possibilities with the new ViSi software tool.

The screen exhibits three gauges, a counter and a meter. All three graphic tools are controlled by the same index and hence all reflect the same value between 1 and 100.

Three buttons control the meters titled; Start, Pause and Stop. The Start button will begin the counter counting from either zero or wherever the counter is up to.

The counter resets every time it reaches 100. The Pause button will stop the counter wherever it is up to; whilst the Stop button will reset the counter to zero.

Examine the code to see just how simple it is to generate a fully graphic and interactive program with only a few lines of code. Comments will be placed throughout the code to explain certain areas in detail.
Place the .4dg and .4DVisi file that comes with this application note into a project folder on the PC. Open the .4dg file through ViSi to load the example application. Be sure to follow the steps explained in the previous section for loading the .gci file and .dat file onto the micro-SD in order for the program to work.
9. Objects

ViSi provides an easy method of designing complex Graphics User Interface applications. The user drags and drops objects on his simulated screen and the code is updated accordingly.

Each object is presented with its button on the left and an example on the right when used on a form.

9.1. Buttons Object

The Buttons pane contains the following widgets:

- Win Button

![Win Button Example](image)

- User Button

![User Button Example](image)

- Animated Button

![Animated Button Example](image)
• 4D Buttons

Button01  
Rocker03  
ON  OFF

Button02  
Slider01  
ON  OFF

Rocker01  
Toggle01

Rocker02  
Toggle02
9.2. Digits Objects

The Digits pane contains 4 different displays.

- LED Digits

![LED Digits](image)

- Custom Digits

![Custom Digits](image)

- LED

![LED](image)

- User LED

![User LED](image)
The Gauges pane contains the following:

- Meter

- Gauge

- Angular Meter
- Cool Gauge

- Thermometer

- Tank

- Spectrum

- Scope
• Smart Gauge (PRO version)

This is a highly customizable gauge. The image presented above is a basic smart gauge example.

**Note:** For more information on the Smart Gauge object, please refer to the [Smart Widgets Editor User Guide](#).
9.4. Primitive Objects

The Primitives pane offers 6 standard static drawings and 2 dynamic objects.

- Circle

- Rectangle

- Triangle

- Line

- Ellipse
- Panel

- Button
9.5. Input Objects

The Inputs pane contains rotary selectors, linear selectors, keyboards and switches.

- **Knob**

![Knob](image)

The minimum and maximum angles, the back and the handle can be customised.

- **Rotary Switch**

![Rotary Switch](image)

The minimum and maximum angles, the positions and labels, the switch and the winch colours can be customised.

- **Slider**

![Slider](image)

The minimum and maximum values, the vertical or horizontal orientations, the colours can be customised.
- Track-bar

The minimum and maximum values, the vertical or horizontal orientations, the frequency and ticks, the colours can be customised.

- Keyboard
ViSi comes with various defined keyboards:

- QWERTY keyboard, by default,

- Cell-phone keyboard

- Numeric keyboard

- And even a customised keyboard.

The different keyboards are selected by clicking on the **KeyboardType** property:
Click on the button to launch the Keyboard Editor:

The Keyboard Editor allows to select and customise the keyboard:

- **DIP Switch**

  The number of positions of the switch can be specified, 2 as shown or more.

- **Rocker Switch**

  When on, the red LED is turned on.
• Color Picker

This is a highly customizable knob. The image presented above is a basic knob example.

**Note:** For more information on the Smart Knob object, please refer to the Smart Widgets Editor User Guide.

• Smart Knob (PRO version)

• Smart Slider (PRO version)

This is a highly customizable slider. The image presented above is a basic slider example.

**Note:** For more information on the Smart Slider object, please refer to the Smart Widgets Editor User Guide.
9.6. Label Objects

The Labels pane offers three different objects to display text.

- Label

  ![Label Example]

- Static Text

  ![Static Text Example]

- Strings

  ![Strings Example]
The text is defined by:

Font, size, ANSI or Unicode can be defined.
9.7. System/Media Objects

The System pane includes the image, video, form, and user images objects.

- Image

The image is selected through an Open window:
• **Video**

The video is selected through an Open window:

![Open window for selecting video](image)

• **Form**

The Form creates a new empty form and adds it to the project.

![Form creation](image)
• User Images

The user images object represents an easy way to build a slideshow by joining together a sequence of images in one place. The images are selected through the Image List editor window.

9.8. Backgrounds

Background objects can be modified accordingly by the user.

• Border
9.9. Selection Tool

The arrow is used to deselect and object.

To select an object, just click on it: green or red dotted lines appear.

To deselect an object, just click again: the dotted lines disappear.
10. Connect the Screen

Connect the screen to a USB port with the 4D Systems programming cable and select the Comms menu:

Above the Comms section, the violet light mentions no screen is currently connected.

Connect the 4D Systems’ display to the screen and plug the cable into the USB port.

Click on the drop-down list and select the COM port, here COM25.

The light turns yellow while the connection is being established:

Finally, the light goes blue when the connection is established.
The light turns red when no screen is attached to the selected port:

![Image of red light indicating device is not responding]
11. Insert the Micro-SD Card

For Picaso, Picaso Lite, and Diablo16, the micro-SD card shall be FAT16-formatted. Partition can’t exceed 4 GB. For Goldelox, the micro-SD card shall not be formatted at all, it requires the SD card to be RAW.

To connect the micro-SD card, either

- Insert the micro-SD card into the USB adaptor and plug the USB adaptor into an USB port of the PC

Or

- Insert the micro-SD card into a micro-SD to SD card converter and plug the SD card converter into the SD card slot of the PC.

Check the micro-SD card is mounted, here as drive E:

For Goldelox, if prompted to format the SD card, click no/cancel. Leave the card unformatted and Workshop4 will handle the rest.
12. Communication Terminal

An alternative to the debugger is the Terminal.

To launch the Terminal, select the **Tools** menu...

...and

- Click ‘Terminal connect 9600’ to open the currently selected com port at 9600 baud in the Terminal program.
- Click ‘Terminal connect 115200’ to open the currently selected com port at 115200 baud in the Terminal program.

A new screen appears:

To send the commands on hexadecimal format, press

The commands sent by the host and the messages sent by the screen are the same as with the **Genie Test Executor** debugger.

The white area on the right displays

- In **green** the messages sent to the screen;
- And in **red** the messages received from the screen
Here, the command Set Slider0 to value 0x17 is sent, or 04 00 17 displayed in green on the terminal window.
#platform "uLCD-32PT"
#inherit "4DGL_16bitColours.fnc"
#inherit "VisualConst.inc"
#inherit "4DViSi 4D-AN-1010Const.inc"
#inherit "ledDigitsDisplay.inc"

var x,y;
var mindex;

func main()
putstr("Mounting...");
if (!((disk:=file_Mount(0))){
    while (!((disk := file_Mount(1)))) // check micro-SD is loaded
        putstr("Drive not mounted...");
pause(200);
gfx_Cls();
pause(200);
}
wend

gfx_TransparentColour(0x0020);
gfx_Transparency(ON);

hndl := file_LoadImageControl("4DVISI~1.dat", "4DVISI~1.gci", 1);

img_Show(hndl,iwinbutton1); // meter1 generated 11/8/2011 2:05:10 PM
img_Show(hndl,iwinbutton2); // winbutton2 generated 11/8/2011 2:05:10 PM
img_Show(hndl,iwinbutton3); // winbutton3 generated 11/8/2011 2:05:10 PM
img_Show(hndl, ileddigits1); // leddigits1 generated 11/10/2011 12:56:25 PM
img_Show(hndl, igauge1); // gauge1 generated 11/10/2011 4:39:17 PM
img_Show(hndl, igauge2); // gauge2 generated 11/10/2011 12:59:28 PM

mindex := 0; // This is the index that controls the value of all the meters

touch_Set(TOUCH_ENABLE); // enable touch

repeat
if (touch_Get(TOUCH_STATUS) == TOUCH_PRESSED) // scan for touch
    x := touch_Get(TOUCH_GETX()); // get x coordinate
    y := touch_Get(TOUCH_GETY()); // get y coordinate
    if ((x >= 104 && x <= 224) && (y >= 188 && y <= 223)) // coordinates for Start Button
        img_SetWord(hndl, iwinbutton3, IMAGE_INDEX, 1); // show start button depressed
        img_Show(hndl,iwinbutton3);
pause(200);
        img_SetWord(hndl, iwinbutton3, IMAGE_INDEX, 0); // release start button
        img_Show(hndl,iwinbutton3);
        repeat
            mindex++; // increment the index
            if (mindex == 100) // if the index reaches 100, reset it to 0
                mindex := 0;
            endif
            img_SetWord(hndl, imeter1, IMAGE_INDEX, mindex); // update imeter1
            img_Show(hndl, imeter1);
            ledDigitsDisplay(mindex, ileddigits1+1, 56, 3, 2, 43, 0); // update leddigits1
            img_SetWord(hndl, igauge4, IMAGE_INDEX, mindex); // update igauge4
            img_Show(hndl, igauge4);
            img_SetWord(hndl, igauge4, IMAGE_INDEX, mindex); // update ugaugel
            img_Show(hndl, igauge4);
            img_SetWord(hndl, igauge2, IMAGE_INDEX, mindex); // update igauge2
            img_Show(hndl, igauge2);
            if (touch_Get(TOUCH_STATUS) == TOUCH_PRESSED)
                x := touch_Get(TOUCH_GETX());
                y := touch_Get(TOUCH_GETY());
                if ((x >= 104 && x <= 224) && (y >= 228 && y <= 263)) || ((x >= 104 && x <= 224) && (y >= 268 && y <= 303))
                    break; // if a touch is detected on either of the other buttons, break out
            endif
        endif
    endif
forever
endif

if ((x >= 104 && x <= 224) && (y >= 228 && y <= 263)) // coordinates for Pause Button
    img_SetWord(hndl, iwinbutton2, IMAGE_INDEX, 1); // show pause button depressed
    img_Show(hndl,iwinbutton2);
pause(200);
img_SetWord(hndl, iwinbutton2, IMAGE_INDEX, 0); // release pause button
img_Show(hndl, iwinbutton2);
img_SetWord(hndl, imeter1, IMAGE_INDEX, mindex); // update imeter1
img_Show(hndl, imeter1);
ledDigitsDisplay(mindex, ileddigits1+1, 56, 3, 2, 43, 0); // update leddigits1
img_SetWord(hndl, igauge4, IMAGE_INDEX, mindex); // update igauge4
img_Show(hndl, igauge4);
img_SetWord(hndl, igauge1, IMAGE_INDEX, mindex); // update igauge1
img_Show(hndl, igauge1);
img_SetWord(hndl, igauge2, IMAGE_INDEX, mindex); // update igauge2
img_Show(hndl, igauge2);
endif
if ( (x >= 104 && x <= 224) && (y >= 268 && y <= 303) ) // coordinates for Stop Button
    pause(200);
    img_SetWord(hndl, iwinbutton1, IMAGE_INDEX, 1); // show stop button depressed
    img_Show(hndl, iwinbutton1);
endif
forever
endfunc
13. Application Notes

For a more detailed presentation of the objects with examples, please refer to the application notes section of the 4D Labs website.


Under “Environments” choose “ViSi”

All ViSi-related application notes will now appear in the search results.
14. Revision History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Revision Content</th>
<th>Revision Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>First Release</td>
<td>16/11/2012</td>
</tr>
<tr>
<td>1.1</td>
<td>Amended microSD detail</td>
<td>07/07/2013</td>
</tr>
<tr>
<td>1.2</td>
<td>Added the new widgets</td>
<td>23/08/2013</td>
</tr>
<tr>
<td>1.3</td>
<td>Added Z-order description to Section 8.2</td>
<td>22/12/2014</td>
</tr>
<tr>
<td>2.1</td>
<td>Updated formatting and contents, added Smart Widgets discussions</td>
<td>25/07/2017</td>
</tr>
<tr>
<td>2.2</td>
<td>Updated formatting</td>
<td>05/04/2019</td>
</tr>
</tbody>
</table>
15. Legal Notice

Proprietary Information

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

4D Labs Semiconductors 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 Labs Semiconductors products and services is continuous and published information may not be up to date. It is important to check the current position with 4D Labs Semiconductors. 4D Labs Semiconductors reserves the right to modify, update or make changes to Specifications or written material without prior notice at any time.

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

Disclaimer of Warranties & Limitation of Liability

4D Labs Semiconductors makes no warranty, either expressed 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.

Images and graphics used throughout this document are for illustrative purposes only. All images and graphics used are possible to be displayed on the 4D Labs Semiconductors range of products, however the quality may vary.

In no event shall 4D Labs Semiconductors 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 Labs Semiconductors, or the use or inability to use the same, even if 4D Labs Semiconductors has been advised of the possibility of such damages.

4D Labs Semiconductors 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 Labs Semiconductors and its suppliers specifically disclaim any expressed or implied warranty of fitness for High Risk Activities.

Use of 4D Labs Semiconductors’ 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 Labs Semiconductors from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any 4D Labs Semiconductors intellectual property rights.

16. Contact Information

For Technical Support: www.4dlabs.com.au/support
For Sales Support: sales@4dlabs.com.au
Website: www.4dlabs.com.au

Copyright 4D Labs Semiconductors 2000-2019.