4D LABS SEMICONDUCTORS

Workshop4

Integrated Development Environment

Pixxi Custom Display Configuration User Manual

Document Revision: 1.0

Document Date: 27th March 2020

Table of Content

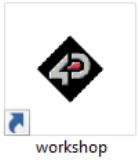
1. Pixxi Custom Display Configuration Introduction

This manual describes the details for configuring the PIXXI-28 and PIXXI-44 processors to interface with any custom display through the 4D Labs Project available only in Workshop4.

The 4D Labs Project Editor allows users to configure the Pixxi processors to interface with virtually any available display driver. The 4D Labs Project Editor does this by writing Display Definitions for the Pixxi processor. The generated display definitions are directly compatible for use in any projects using the same Environments as any other traditional 4D Systems Display.

2. Getting Started

Open Workshop4 by clicking this icon:



By default, Workshop4 will open and display the **Recent** page:

To start the custom Display Definition click the **Create a new 4D Labs Project** button:

3. Display Definition Selection

The main window will be updated for creating or modifying the Display Definitions. Shown in the main window are the recent Display Definitions.

1. **Display Definitions List**
2. **Display Definition**
3. **Incomplete Display Definition**
4. **New Display Definition**
5. **Edit Display Definition**
6. **New Arduino compatible project**
7. **New 4D project**
8. **Device Preview**

Display Definitions List
List of all the recent and template display definitions available for modification

Display Definition
This contains the contents for operating the display driver

Incomplete Display Definition
Display Definitions saved with errors or incomplete entries will be marked as incomplete

New Display Definition
This button will proceed to the workspace where a new Display Definition will be created

Edit Display Definition
This button will proceed to the workspace once a Display Definition is selected from the Display Definition List

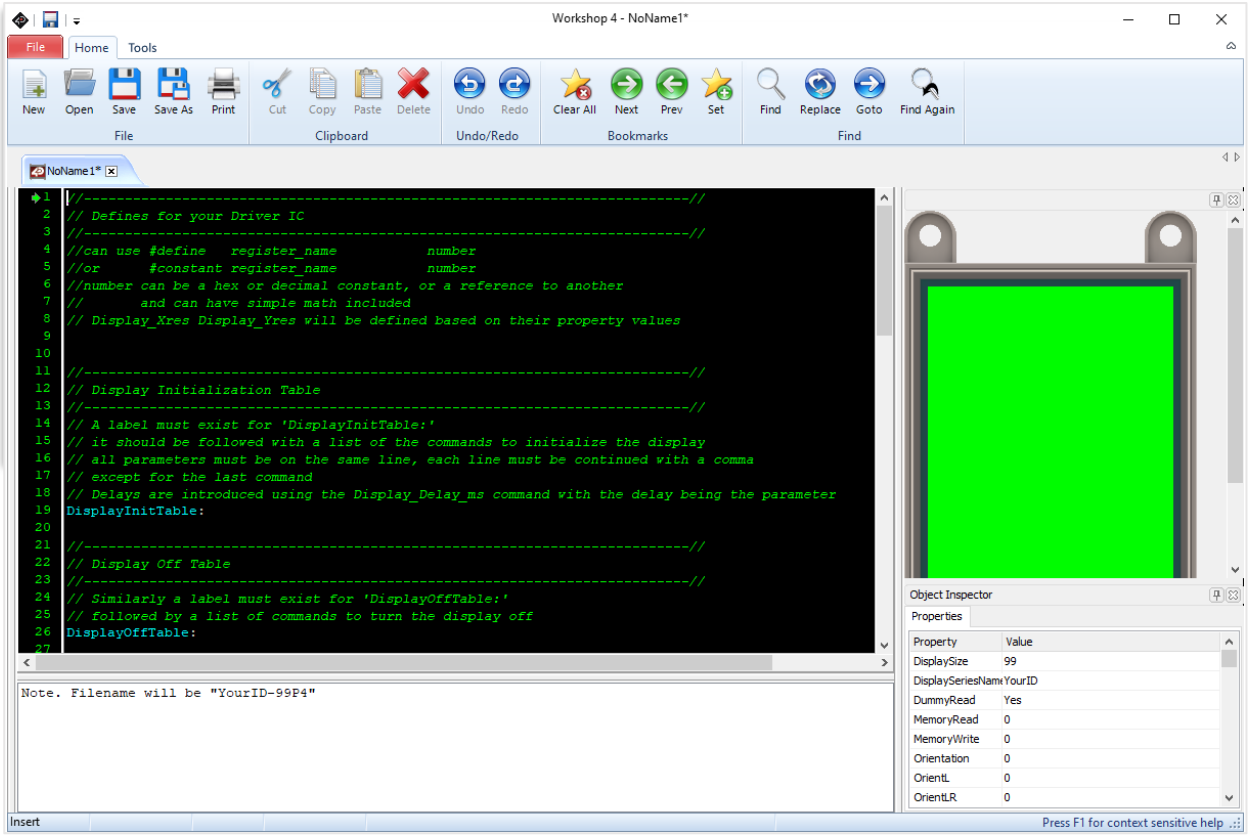
New Arduino compatible project
This button will proceed to Workshop4 environment selection to create project using the selected Display Definition

New 4D project
This button will proceed to Workshop4 environment selection to create project using the selected Display Definition

Device Preview
This shows the preview of the device, clicking this image would change the display orientation for the projects

4. New Display Definition

The workspace for creating new Display definition will show the initial template on the 4DGL Code Configuration. The user will need to fill the required 4DGL configuration and properties in order to deploy a display definition. Once the project is free of errors and saved, the Display Definition files with extension ".4DdisplayDef" and ".inc" are generated. All display definitions are saved to C:\Users\Public\Documents\4D Labs\Display Definitions. The display definitions can only be deleted manually on the same directory.



The display name of the Display Definition is generated based on the Display Definition Properties entry with the format shown below.

[DisplaySeriesName]-[DisplaySize][Processor][Touch]

[DisplaySeriesName] – defined in the property DisplaySeriesName

[DisplaySize] – defined in the property DisplaySize

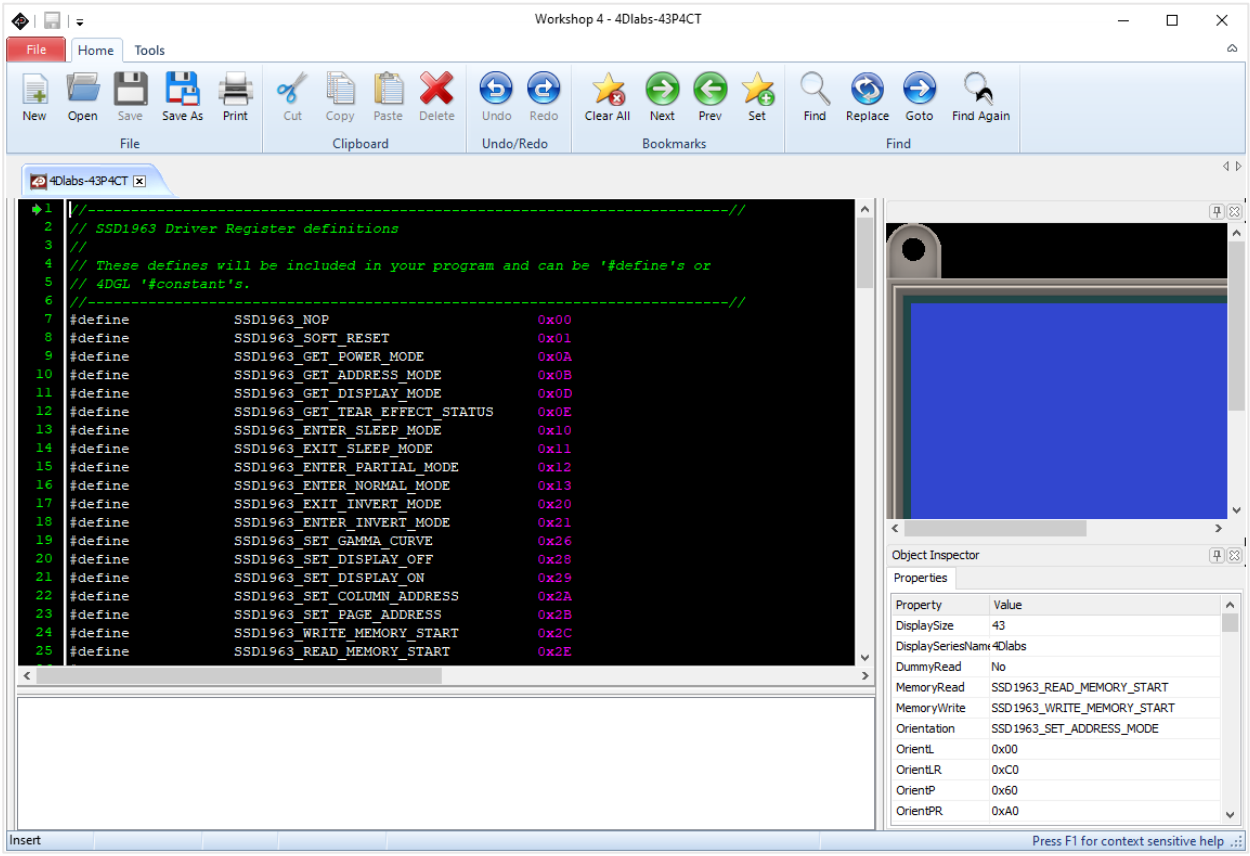
[Processor] – selected 4D Graphics Processor
- **P2** – PIXXI-28
- **P4** – PIXXI-44

[Touch] – selected touch interface type
- **T** – Resistive Touch
- **CT** – Capacitive Touch

5. Edit Display Definition

The workspace will open the selected Display Definition file. Modifications on the properties and 4DGL code configuration can then be carried out on the Display Definition. Once the project is saved free of errors, the Display Definition files with extension ".4DdisplayDef" and ".inc" are saved to C:\Users\Public\Documents\4D Labs\Display Definitions. The display definitions can only be deleted manually on the same directory.

Note: The Workshop4 Display Definition templates cannot be modified directly, rather it can only be saved as a new display definition by changing the display series name.



6. 4DGL Code Configuration

The display constants, initialization, off-sequence, on-sequence and GRAM state machine table contains the key elements in making the display definition function. The definitions, initialization procedure and GRAM access method, should be completed using the display driver manufacturer's datasheet as reference.

```
1  //--------------------------------------------------------------------------//
2  // Defines for your Driver IC
3  //--------------------------------------------------------------------------//
4  //can use #define   register_name          number
5  //or      #constant register_name          number
6  //number can be a hex or decimal constant, or a reference to another
7  //       and can have simple math included
8  // Display_Xres Display_Yres will be defined based on their property values
9
10
11 //--------------------------------------------------------------------------//
12 // Display Initialization Table
13 //--------------------------------------------------------------------------//
14 // A label must exist for 'DisplayInitTable:'
15 // it should be followed with a list of the commands to initialize the display
16 // all parameters must be on the same line, each line must be continued with a comma
17 // except for the last command
18 // Delays are introduced using the Display_Delay_ms command with the delay being the parameter
19 DisplayInitTable:
20
21 //--------------------------------------------------------------------------//
22 // Display Off Table
23 //--------------------------------------------------------------------------//
24 // Similarly a label must exist for 'DisplayOffTable:'
25 // followed by a list of commands to turn the display off
26 DisplayOffTable:
```

6.1. Display Constants

The register addresses and constants are defined in this section for use in the command tables and properties of the display definition project.

```
#define   SSD1963_NOP              0x00
#constant SSD1963_PDI_16BIT565     3
```

6.2. Command Tables

The command tables are utilized by Workshop4 to initialize and operate the displays. Each table follows the format as shown.

```
[Label]:
[Command], [Parameter 1], …, [Parameter N],
 …
[Command], [Parameter 1], …, [Parameter N],
[Command], [Parameter 1], …, [Parameter N]
```

The table is composed of a table type label and a set of command lines. A command line comprising of one or more items separated by commas will occupy a single row in the table. The first item in a command line is a display driver command which can be found in the display driver datasheet. The command can then be followed by its required parameter values as prescribed in the display driver datasheet. Each command line should be terminated by a comma, except for the last line before the next table.

Note: In order to introduce a delay in the program when required, a command line containing the special command "Display_Delay_ms" followed by its delay parameter in milliseconds can be used. For example, to introduce a 1 ms delay the following command line should be inserted:
Display_Delay_ms, 1

6.2.1 Display Initialization

This table contains list of command lines for initializing the display at startup. This table is labelled **"DisplayInitTable"**.

```
DisplayInitTable:
    SSD1963_SET_PLL_MN, 0x23, 0x02, 0x54,
    SSD1963_SET_PLL, 0x01,
    Display_Delay_ms, 1,
    SSD1963_SET_PLL, 0x03,
    SSD1963_SOFT_RESET
```

6.2.2 Display On Table

This table contains list of command lines for turning the display on. This table is labelled **"DisplayOnTable"**.

```
DisplayOnTable:
    SSD1963_SET_DISPLAY_ON
```

6.2.3 Display Off Table

This table contains list of command lines for turning the display off. This table is labelled **"DisplayOffTable"**.

```
DisplayOffTable:
    SSD1963_SET_DISPLAY_OFF
```

6.2.4 GRAM State Machine

This table contains list of command lines for setting up GRAM access of the display driver. This table is labelled **"GRAMStateMachine"**. The table can be composed using the combination of the following GRAM Access command lines and GRAM Address variables.

GRAM Access Command Lines

The following command line is used when a register address or constant is required to be sent as Display Command.

```
GRAM_CONTROL_CONSTANT, [Constant],
```

The following command line is used when a GRAM address variable is required to be sent as Display Data.

```
GRAM_DATA_VAR, [Variable],
```

The following command line can be used in some special case, where a constant is required to be sent as Display Data.

```
GRAM_DATA_CONSTANT, [Constant],
```

The following command line can be used in some special case, where a GRAM address variable is required to be sent as Display Command.

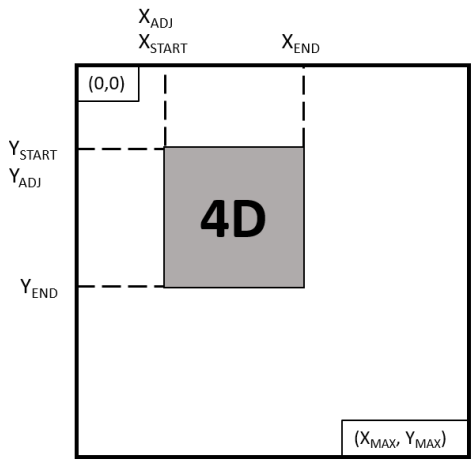
```
GRAM_CONTROL_VAR, [Variable],
```

The GRAM state machine is ended by using the following command line.

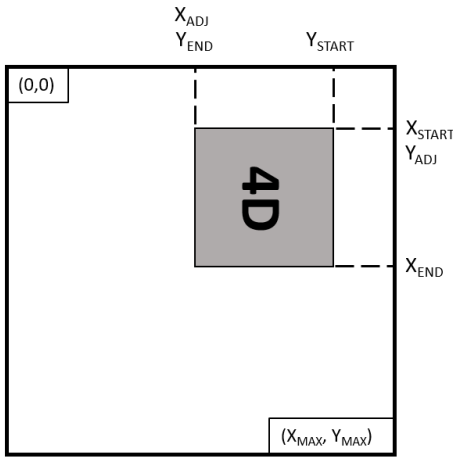
```
GRAM_EXIT
```

GRAM Address Variables

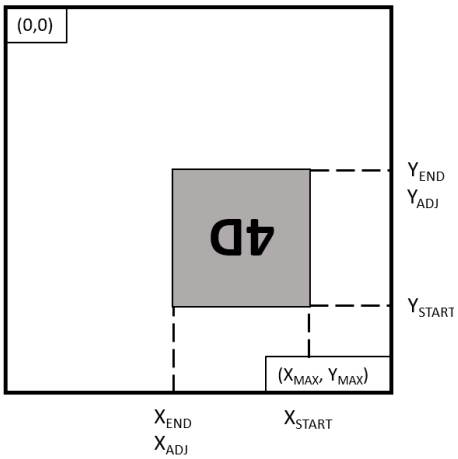
Most display drivers will require memory addresses to define an access/draw area inside the display area. The Pixxi processor provides the following system variables that can be used in the GRAM State Machine for this. The address values of the system variables are relative to the display orientation as shown in the image below.



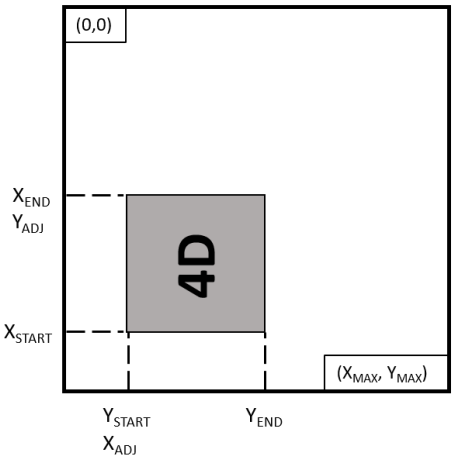
Native Orientation



Rotated 90°



Rotated 180°



Rotated 270°

Address Variables	Description
VAR_X_START	Horizontal start address word (X_{START})
VAR_X_START_HI	Horizontal start address high byte (X_{START})
VAR_X_START_LO	Horizontal start address low byte (X_{START})
VAR_X_END	Horizontal end address word (X_{END})
VAR_X_END_HI	Horizontal end address high byte (X_{END})
VAR_X_END_LO	Horizontal end address low byte (X_{END})
VAR_Y_START	Vertical start address word (Y_{START})

Address Variables	Description
VAR_Y_START_HI	Vertical start address high byte (Y_{START})
VAR_Y_START_LO	Vertical start address low byte (Y_{START})
VAR_Y_END	Vertical end address word (Y_{END})
VAR_Y_END_HI	Vertical end address high byte (Y_{END})
VAR_Y_END_LO	Vertical end address low byte (Y_{END})
VAR_X_START_END	Combined Horizontal addresses (Start in high byte, End in low byte)
VAR_X_END_START	Combined Horizontal addresses (End in high byte, Start in low byte)
VAR_Y_START_END	Combined Vertical addresses (Start in high byte, End in low byte)
VAR_Y_END_START	Combined Vertical addresses (End in high byte, Start in low byte)
VAR_X_START_ADJ	Adjusted horizontal start address (X_{ADJ})
VAR_Y_START_ADJ	Adjusted vertical start address (Y_{ADJ})
VAR_WIDTH	Access/Draw window width word
VAR_WIDTH_HI	Access/Draw window width high byte
VAR_WIDTH_LO	Access/Draw window width low byte
VAR_HEIGHT	Access/Draw window height word
VAR_HEIGHT_HI	Access/Draw window height high byte
VAR_HEIGHT_LO	Access/Draw window height low byte

The example below shows a display driver GRAM state machine table.

```
GRAMStateMachine:
    // set x address
    GRAM_CONTROL_CONSTANT,  ILI9225G_HORIZONTAL_RAM_ADDRESS_POSITION_STA,
    GRAM_DATA_VAR,          VAR_X_START,

    GRAM_CONTROL_CONSTANT,  ILI9225G_HORIZONTAL_RAM_ADDRESS_POSITION_END,
    GRAM_DATA_VAR,          VAR_X_END,

    // Set y address
    GRAM_CONTROL_CONSTANT,  ILI9225G_VERTICAL_RAM_ADDRESS_POSITION_STA,
    GRAM_DATA_VAR,          VAR_Y_START,

    GRAM_CONTROL_CONSTANT,  ILI9225G_VERTICAL_RAM_ADDRESS_POSITION_END,
    GRAM_DATA_VAR,          VAR_Y_END,

    GRAM_CONTROL_CONSTANT,  ILI9225G_RAM_ADDRESS_SET_H,
    GRAM_DATA_VAR,          VAR_X_START_ADJ,

    GRAM_CONTROL_CONSTANT,  ILI9225G_RAM_ADDRESS_SET_V,
    GRAM_DATA_VAR,          VAR_Y_START_ADJ,

    GRAM_EXIT
```

7. Display Definition Properties

The properties bar contains the fixed settings of the Display Definition. The properties can utilize and access the constants listed on the Display Constants in the 4DGL Code Configuration through its dropdown list.

7.1. Display Definition Details

The following properties are used to modify the display identification in the Display Definition List. This will become part of the Display Definition name.

Property	Value
DisplaySize	99
DisplaySeriesName	YourID

Option	Description
DisplaySize	Value for display size identification (e.g. 43 for 4.3" diagonal)
DisplaySeriesName	Series name for the display

The following properties are used to modify the aesthetics of the Device Preview in the Display Definition selection window.

Property	Value
VisualDescription	
VisualScreenColor	LIME
VisualTextOverlay	

Option	Description
VisualDescription	Description on the 'Choose your product' screen image
VisualScreenColor	Screen Color on the 'Choose your product' screen image
VisualTextOverlay	Text Overlay on the 'Choose your product' screen image

7.2. Processor Setup

This is used for selecting the graphics processor to be used.

Property	Value
Processor	Pixxi-44

Option	Description
Processor	4D Labs grahics processor type (PIXXI-28 or PIXXI-44)

7.3. Display Resolution

The display X and Y resolution is set in reference to its native orientation.

Property	Value
Xoffset	0
Xres	240
Yoffset	0
Yres	320

Option	Description
Xoffset	X offset for displays having an offset in their native orientation
Xres	X resolution of display in its native orientation (when top left is pixel 0,0)
Yoffset	Y offset for displays having an offset in their native orientation
Yres	Y resolution of display in its native orientation (when top left is pixel 0,0)

7.4. Display Orientation Settings

The following properties are for setting up the display driver draw orientation. The orientation opcodes can be determined through the display driver datasheet.

Property	Value
Orientation	0
OrientL	0
OrientLR	0
OrientP	0
OrientPR	0

Property	Setting
Orientation	Register address for setting display orientation
OrientL	Display driver Landscape opcode
OrientLR	Display driver Landscape-Reversed opcode
OrientP	Display driver Portrait opcode
OrientPR	Display driver Portrait-Reversed opcode

The following properties are for changing the X and Y location in the display driver.

Property	Value
PmmCFlipXY	Yes
PmmCSwapXY	Yes

Property	Setting
PmmCFlipXY	Flip X and Y when display orientation is rotated
PmmCSwapXY	Swap X and Y when display orientation is swapped or reversed

7.5. Graphics Settings

The following properties are used for setting up command opcodes for starting display driver graphics memory read or write operation.

Property	Value
MemoryRead	0
MemoryWrite	0

Property	Setting
MemoryRead	Setup display driver graphics memory for read operation
MemoryWrite	Setup display driver graphics memory for write operation

The format of the pixel data read from the display driver can be specified in here.

Property	Value
DummyRead	Yes
ReadPixelFormat	565
ReadPixelOrder	RGB

Property	Setting
DummyRead	Enable when a dummy read is required to read GRAM
ReadPixelFormat	Select pixel read format (separate RGB or 666; or single word 565)
ReadPixelOrder	Select pixel read order (RGB or BGR)

7.6. SPI Interface Settings

The following properties are for the SPI-based display driver interface settings applicable only for Pixxi-28 processors.

Property	Value
SpiReadExitCmd	Yes
SpiReadFast	Yes
SpiRegister16	No
SpiWriteFast	Yes

Property	Setting
SpiReadExitCmd	Enable use of exit command (0x66) to exit GRAM reading
SpiReadFast	Enable high SPI read speed (Low Speed: ~23 MHz; High Speed: ~35 MHz)
SpiRegister16	In SPI mode, Use 16-bit SPI for register data writes
SpiWriteFast	Enable high SPI write speed (Low Speed: ~23 MHz; High Speed: ~35 MHz)

7.7. Touch Detection Settings

The following settings are for setting up touch detection. There are four available touch detection system for the Pixxi processor.

Property	Value
Touch	(None)

Setting	Meaning
FocalTech	For capacitive touch displays using FocalTech chip (FT5x06, FT6x36, etc.)
Internal	Enables internal touch interface for small resistive displays (3.5” or smaller)
(None)	No touch interface
4DL763 (XPT7603)	For resistive touch display using touch controller chip (4DL-763 or XPT7603)

Note: For smaller displays, resistive touch can be handled internally. For displays above 3.5” an external resistive touch chip (4DL-763 or XPT7603) must be used, this is because the noise picked up by the larger touch panel areas requires a dedicated chip for noise isolation.

Touch calibration values can be set manually for Internal and Resistive touch. These values can be fine-tuned by running a touch calibration utility. The touch calibration utility cannot update the values internally during runtime, hence the need for manual input of the calibration values.

Property	Value
TouchCalMaxX	0x2DC
TouchCalMinX	0xB3
TouchCalMaxY	0x30F
TouchCalMinY	0x81

Property	Setting
TouchCalMaxX	Initial Touch Calibration value for X Maximum
TouchCalMinX	Initial Touch Calibration value for X Minimum
TouchCalMaxY	Initial Touch Calibration value for Y Maximum
TouchCalMinY	Initial Touch Calibration value for Y Minimum

The touch direction of the touch panel can be flipped or swapped through these settings.

Property	Value
TouchFlipX	No
TouchFlipY	No
TouchSwapXY	No

Property	Setting
TouchFlipX	Flip horizontal touch direction
TouchFlipY	Flip vertical touch direction
TouchSwapXY	Swap X and Y touch coordinates (Not applicable for Internal Touch)

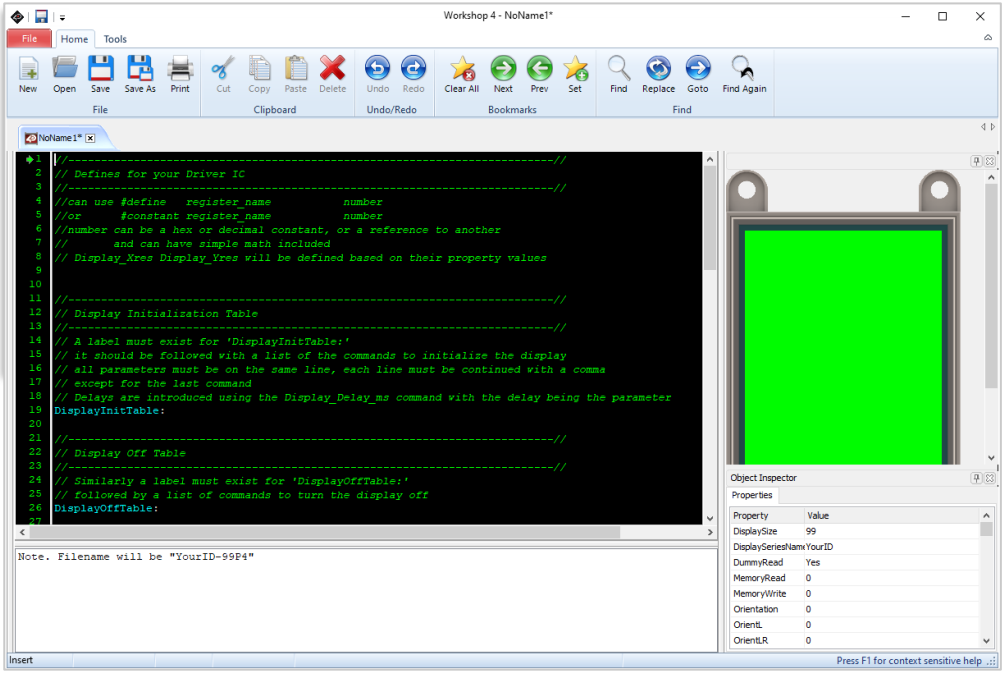
Note: If the values are flipped, the touch calibration utility may not be usable, so it is best to ensure that the hardware is correct.

8. Display Definition Setup Guide

This section presents a walkthrough on the process of creating a Display Definition. Users can also check the working Display Definitions available in the 4D Labs Project of Workshop4 for additional reference.

8.1. Start a New Display Definition

The Pixxi Display Definition is started by opening a new Display Definition by clicking the New Display Definition button in the Display Definition Selection window of the 4D Labs Project. This will create and open a new template for user to begin with for their Display Definition.



8.2. Prepare the Display Constants

The user can start the Display Definition project by defining all of the necessary commands and constants for the Display Constants in the 4DGL Code Configuration, following the format discussed in **Display Constants**. These constants will be used in the succeeding parts of the Display Definition. The users should be able to find the list of display driver commands (example shown below) in the datasheet of their display driver.

8.1. Command List

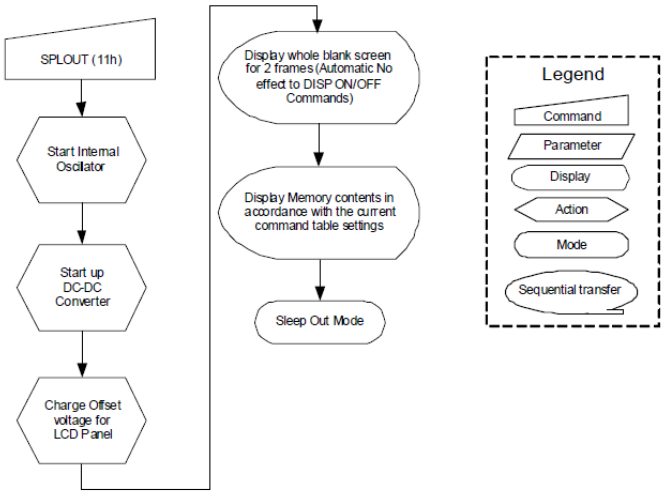
Regulative Command Set													
Command Function	D/CX	RDX	WRX	D17-8	D7	D6	D5	D4	D3	D2	D1	D0	Hex
No Operation	0	1	↑	XX	0	0	0	0	0	0	0	0	00h
Software Reset	0	1	↑	XX	0	0	0	0	0	0	0	1	01h
Read Display Identification Information	0	1	↑	XX	0	0	0	0	0	1	0	0	04h
	1	↑	1	XX	X	X	X	X	X	X	X	X	XX
	1	↑	1	XX	ID1 [7:0]								XX
	1	↑	1	XX	ID2 [7:0]								XX
	1	↑	1	XX	ID3 [7:0]								XX

```
4  #define    READ_DISPLAY_PIXEL_FORMAT    0x0C
5  #define    SLEEP_OUT                    0x11
6  #define    GAMMA_SET                    0x26
7  #define    DISPLAY_OFF                  0x28
8  #define    DISPLAY_ON                   0x29
9  #define    SET_COLUMN_ADDRESS           0x2A
10 #define    SET_PAGE_ADDRESS             0x2B
11 #define    WRITE_MEMORY                 0x2C
12 #define    READ_MEMORY                  0x2E
```

8.3. Write the Command Tables

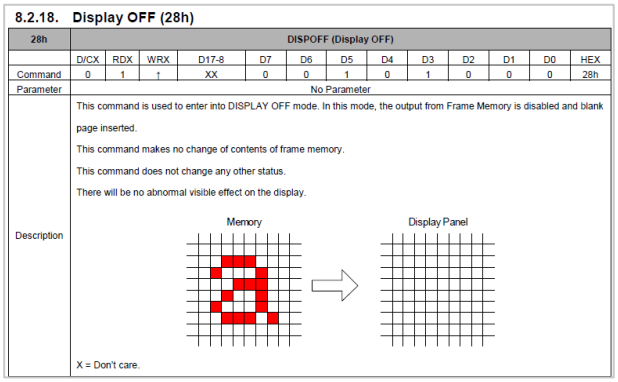
After listing down the necessary constants in the Display Constants, the user can proceed by filling up the command tables. The contents of the command tables should follow the format discussed in **Command Tables**.

The first command table that should be filled is the **Display Initialization** table. This table is responsible for initializing the display driver by executing the startup command sequence.




```
34  DisplayInitTable:
35      INTERFACE_CONTROL,          0x01, 0x01, 0x00,
36      UNDOCUMENTED_0xEF,          0x03, 0x80, 0x02,
37      POWER_CONTROL_B,            0x00, 0xF2, 0xA0,
38      // POWER_CONTROL_B,         0x00, 0xD9, 0x30, WONT WORK - GIVES WHITE SCREEN
39      POWER_ON_SEQUENCE_CONTROL,  0x64, 0x03, 0x12, 0x81,
40      POWER_CONTROL_A,            0x39, 0x2C, 0x00, 0x34, 0x02,
41      DRIVER_TIMING_CONTROL_B,    0x00, 0x00,
42      DRIVER_TIMING_CONTROL_A,    0x85, 0x10, 0x7A,
43      POWER_CONTROL_1,            0x21,                 //VRH[5:0]
44      POWER_CONTROL_2,            0x11,                 //SAP[2:0];BT[3:0]
45      VCOM_CONTROL_1,             0x3F, 0x3C,
46      VCOM_CONTROL_2,             0xC6,                       // 0xD2
47      PIXEL_FORMAT_SET,           0x55,
```

The next tables that should be filled are the **Display On Table** and **Display Off Table** which contains the command sequence for turning the display on and off respectively. In this example, the display driver has a command for turning on and another for turning off the display, these are then listed in the respective command tables.



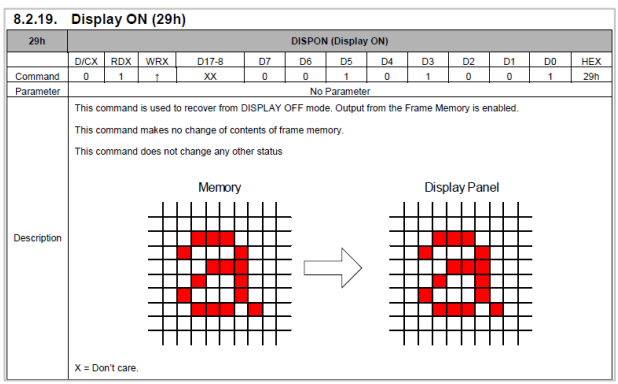
8.2.18. Display OFF (28h)

28h	DISPOFF (Display OFF)												
	D/CX	RDX	WRX	D17-8	D7	D6	D5	D4	D3	D2	D1	D0	HEX
Command	0	1	↑	XX	0	0	1	0	1	0	0	0	28h
Parameter	No Parameter												

Description:
This command is used to enter into DISPLAY OFF mode. In this mode, the output from Frame Memory is disabled and blank page inserted.
This command makes no change of contents of frame memory.
This command does not change any other status.
There will be no abnormal visible effect on the display.

Memory → Display Panel

X = Don't care.



8.2.19. Display ON (29h)

29h	DISPON (Display ON)												
	D/CX	RDX	WRX	D17-8	D7	D6	D5	D4	D3	D2	D1	D0	HEX
Command	0	1	↑	XX	0	0	1	0	1	0	0	1	29h
Parameter	No Parameter												

Description:
This command is used to recover from DISPLAY OFF mode. Output from the Frame Memory is enabled.
This command makes no change of contents of frame memory.
This command does not change any other status

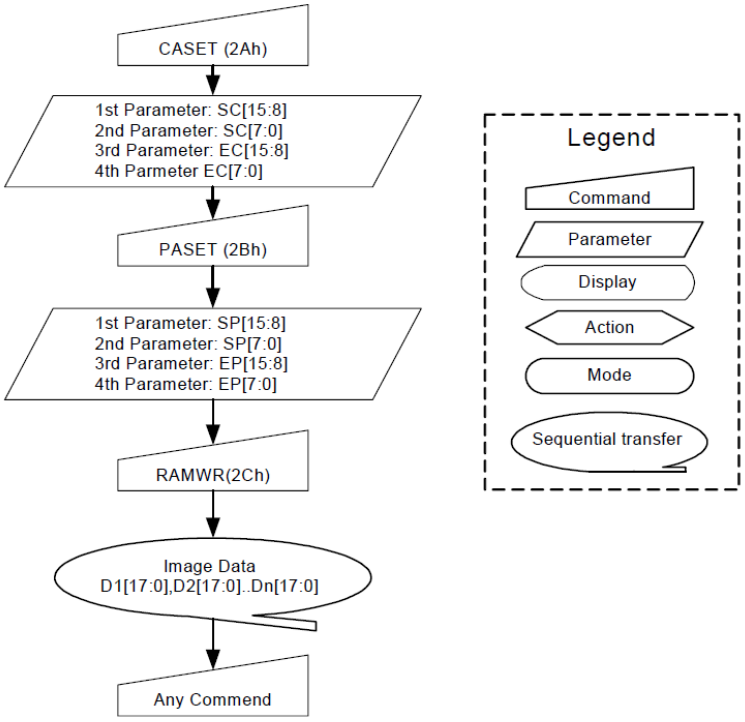
Memory → Display Panel

X = Don't care.

```
60  DisplayOffTable:
61      DISPLAY_OFF
62
63  DisplayOnTable:
64      DISPLAY_ON
```

The last section of the 4DGL Code Configuration is the **GRAM State Machine**, where the command sequence for setting up the GRAM access is located. The GRAM State Machine is responsible for setting up the display driver GRAM access/draw area prior to writing or reading

In this example, the GRAM draw window is initiated by sending the horizontal/column address register command followed by the horizontal start and end addresses, and the vertical/page address register command followed by the vertical start and end addresses as shown in the display driver GRAM setup flowchart below.



Based on the flowchart above, the horizontal/column address register command is first sent using the **GRAM_CONTROL_CONSTANT** command as Display Command.

Following the command are the four Horizontal Address parameters as follows:

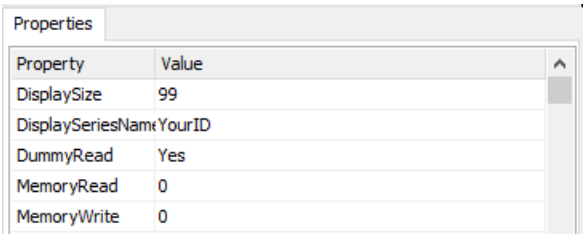
Horizontal Start Address High Byte	**VAR_X_START_HI**
Horizontal Start Address Low Byte	**VAR_X_START_LO**
Horizontal End Address High Byte	**VAR_X_END_HI**
Horizontal End Address Low Byte	**VAR_X_END_LO**

The corresponding GRAM Address variables are sent through **GRAM_DATA_VAR** as Display Data. The same is done with the vertical/column address setup. With the GRAM access set the GRAM state machine is ended by the commandline **GRAM_EXIT**.

```
67  GRAMStateMachine:
68      GRAM_CONTROL_CONSTANT,  SET_COLUMN_ADDRESS,
69      GRAM_DATA_VAR,          VAR_X_START_HI,
70      GRAM_DATA_VAR,          VAR_X_START_LO,
71      GRAM_DATA_VAR,          VAR_X_END_HI,
72      GRAM_DATA_VAR,          VAR_X_END_LO,
73      GRAM_CONTROL_CONSTANT,  SET_PAGE_ADDRESS,
74      GRAM_DATA_VAR,          VAR_Y_START_HI,
75      GRAM_DATA_VAR,          VAR_Y_START_LO,
76      GRAM_DATA_VAR,          VAR_Y_END_HI,
77      GRAM_DATA_VAR,          VAR_Y_END_LO,
78      GRAM_EXIT
```

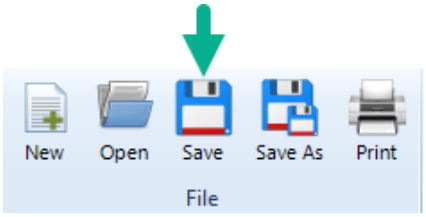
8.4. Set the Display Definition Properties

With the Command Tables completed, the Display Definition Properties can then be filled up to complete the display definition. The details for each item here are discussed in the **Display Definition Properties** section. The Display Definition Properties can also utilize the constants listed on the Display Constants in the 4DGL Code Configuration through a dropdown list.

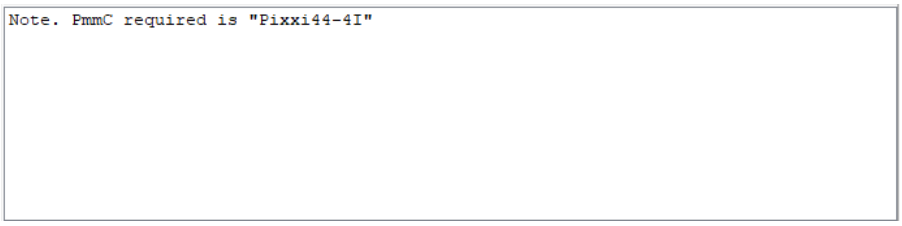


8.5. Save the Display Definition

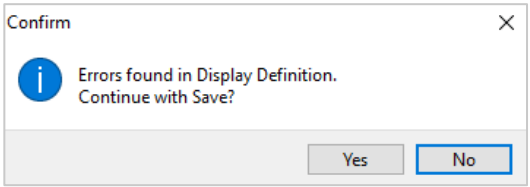
The Display Definition will only be generated after saving it by using the **Save** button in the Home tab.



When the Display Definition is saved without any errors, the status bar will notify the user on which specific PmmC to load in their processor before using the Display Definitions. The Display Definitions are then ready for use in any 4D projects in Workshop4.



If Display Definition contains errors during this process, it will notify the user with a message as shown below. If the user proceeds with the save, the display definition will be marked incomplete and rendered unavailable for use with any 4D projects.



9. Application

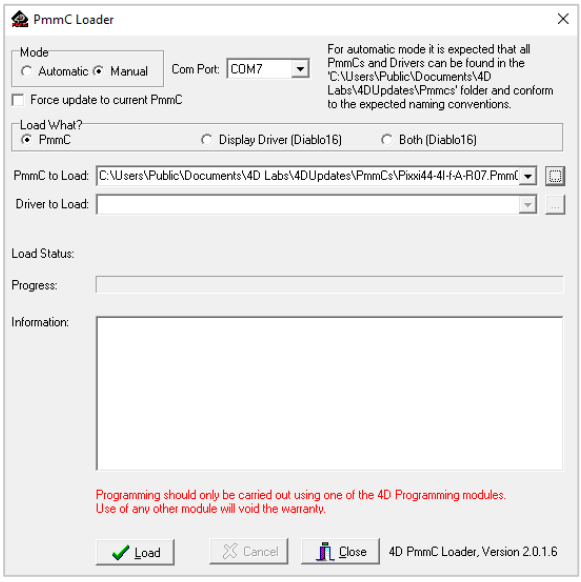
The Display Definitions in 4D Labs Project are available for use in any project in Workshop4. For the Pixxi processor to use the Display Definitions, the PmmC must first be loaded. The PmmC required is specified by Workshop4 after saving the Display Definitions or by selecting one by referring to the tables below.

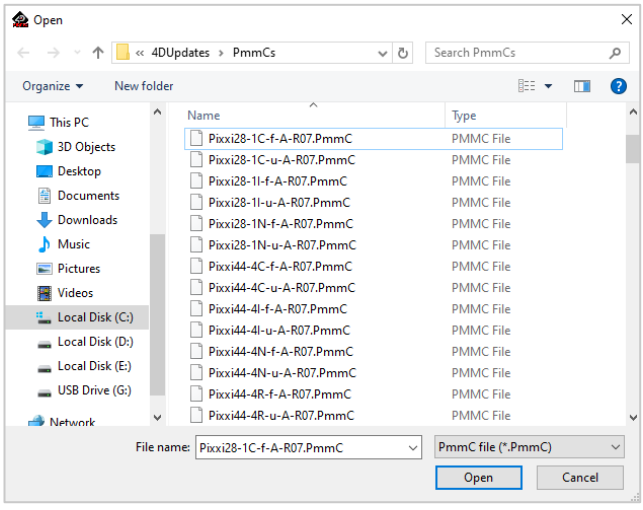
PmmC	Description
Pixxi28-1C	PIXXI-28 processor with Capacitive Touch
Pixxi28-1I	PIXXI-28 processor with Internal Resistive Touch
Pixxi28-1N	PIXXI-28 processor with No Touch
Pixxi44-4C	PIXXI-44 processor with Capacitive Touch
Pixxi44-4I	PIXXI-44 processor with Internal Resistive Touch
Pixxi44-4N	PIXXI-44 processor with No Touch
Pixxi44-4R	PIXXI-44 processor with Resistive Touch controller chip

Depending on the memory storage interface required in the application, the PmmC variant should be selected and loaded.

PmmC Variant	Description
-u	PmmC variant with support for uSD card interface
-f	PmmC variant with support for serial flash memory chip interface

The required PmmC can only be manually loaded to the Pixxi Processor through the PmmC Loader Utility available in Workshop4.

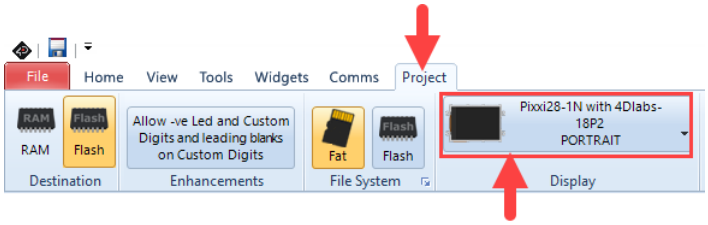


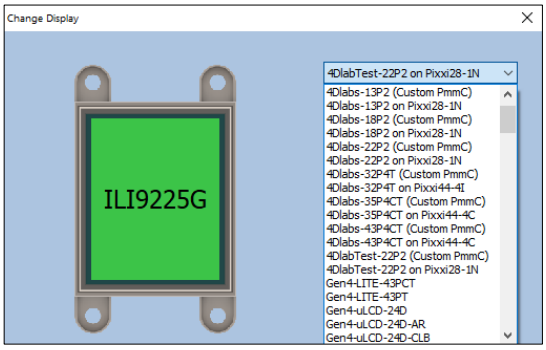


Note: In case the user wants to have a PmmC fully customized for use with their display for product deployment, they can contact 4D Systems to have their Display Definitions converted to a full custom PmmC.

New 4D project can be started through the **Create a new 4D Labs Project** window by selecting a Display Definition in the list, then pressing the **New Arduino Compatible Project** or **New 4D Project** button.

Additionally, the Display Definitions can also be applied for use with other existing 4D projects via the **Change Display** option found in the **Project** tab.

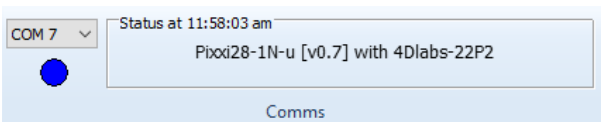




The ".inc" file is then included as a "#inherit" in the second line of any program using this display definition. This section is automatically managed by Workshop so no modification should be done here.

```
1  #platform "Pixxi44-4I"
2  #inherit "ILI9341_3.2_IT.inc"     // *** Display Definition File. Do not move or remove!
```

The Workshop4 Communication tab should also be able to detect and confirm the loaded Display Definition on the display.



10. Test Programs

Test programs are also provided in order to check if the supplied definitions and initialization are in correct working order. Failure to test with the provided test programs might result to failure in observing potential issues that might arise when doing certain operations (e.g. program issue, display issue, driver issue, or hardware issue).

These tests are designed to be small, so that they can be run on small displays without alteration. Using a magnifying glass would is advisable to assist in verifying the accuracy of the output graphics and in ensuring correct display timings. The following test programs can be found in Workshop4 in **File** > **Samples** > **Pixxi Designer**

ChkRect
This program draws a Blue rectangle on the screen. It should ensure the correct resolution and timing settings of the display. If any side of the rectangle is missing, blurred, or moving; the timing parameters supplied by the display vendor should be rechecked, or maybe the hardware glass is obscuring the active display area.

ChkReadPix
This program validates the following parameters concerning pixel transfer operations with the display driver.

Parameters	Test Method
Correct RGB order on write	The text 'BLUE' is shown in colour blue on the display
Simple readpixel	'ReadPixel OK' shown on the display
Complex readpixel	'0 Verify Errors' shown on the display

The image below will show on the display if everything is correct:

ChkOrient

This program validates the following parameters concerning correct display driver orientation commands.

Parameters	Test Method
Orientation Commands	The same information appears in each corner of the screen when rotated. The text and colour of it differs in every orientation.
Transparent Text	The first part of the text will have a line behind it.
Opaque Text	The second part of the text will not have a line behind it.

The image below will show on the display if everything is correct.

The Top Left Bottom Right (TLBR) colour boxes are shown in every orientation with the correct order shown in the image below. Any errors occurring here could be coming from incorrect GRAM setup, i.e. incorrect usage of system variables in the GRAM state machine addresses.

This program can also be used for testing uSD functionality of the Pixxi processor with the display. To activate this test, the following line below should be uncommented.

```
11  //#constant USD
```

Once enabled, the "tlbr.dat" and "tlbr.gci" should be copied to a FAT16 formatted uSD to be read by the Pixxi processor. These files are located at C:\Users\Public\Documents\4D Labs\Pixxi Designer. This small graphic will be added on the display for all orientation:

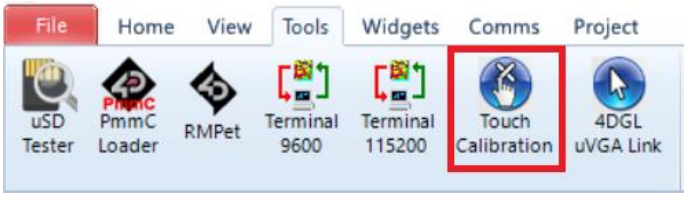
The image below will show on the display if everything is correct:

11. Resistive Touch Interface Calibration

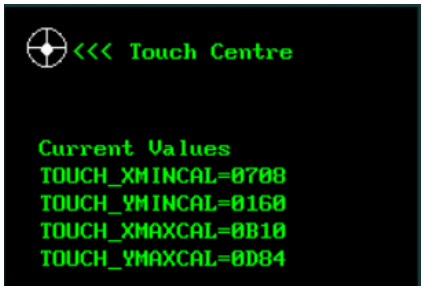
The resistive touch interface will need calibration before use. Discussed in this section are the calibration steps for the resistive touch panels with Internal or External touch controllers.

Initial step is to upload the working display definition to the display module by updating it with the correct PmmC and the display definition, check the **Application** section for more details.

Open any project using the display definition in Workshop4, go to Tools tab, then press the Touch Calibration tool as shown below. This will upload a small program to the display for calibration purposes.



Follow the instructions for calibrating as written on the screen.



After completing the calibration steps, take note of the values written on the display, this will then be used to change the calibration values in the display definition.

Input the values into the Touch Calibration properties accordingly as shown below.

Property	Value
TouchCalMaxX	0x2DC
TouchCalMinX	0xB3
TouchCalMaxY	0x30F
TouchCalMinY	0x81

After this, save the display definition and upload it again to the display module. The display module will now have calibrated resistive touch screen values stored in its display definition.

12. Revision History

Revision	Revision Content	Revision Date
1.0	Initial public release	27/03/2020

13. Legal Notice

13.1. 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 makes changes to Specifications or written material without prior notice at any time.

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

13.2. 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.

14. Contact Information

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