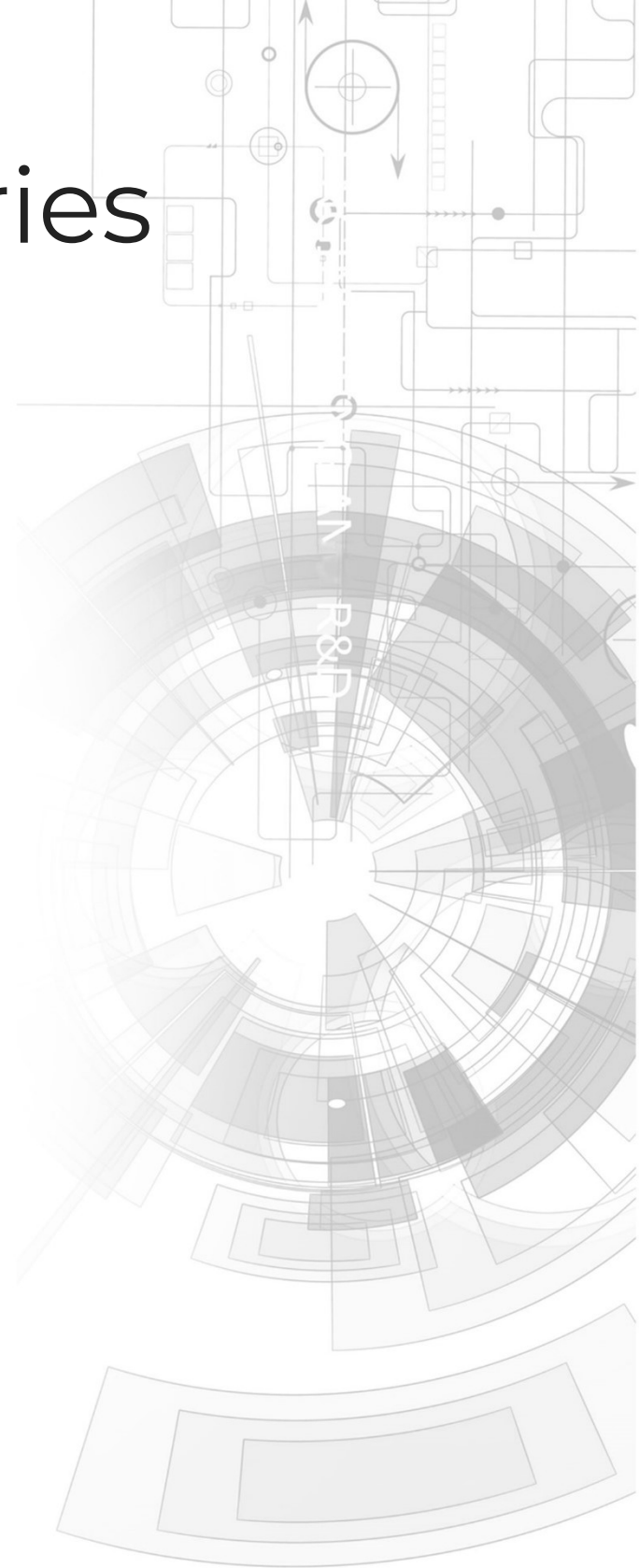


# uLCD-90D Series

uLCD-90DT (Resistive Touch)  
uLCD-90DCT (Capacitive Touch)



## Datasheet

Revision 1.3

Copyright © 2023 4D Systems

Content may change at any time. Please refer to the resource centre for latest documentation.

# Contents

---

1. Description	4
2. Features	5
3. Hardware Overview	6
4. Hardware Interface - Pins	9
4.1. Serial Ports - TTL Level	9
4.2. General Purpose I/O	12
4.3. System Pins	14
4.4. Alternate Pin Function Overview	15
4.5. SPI	18
4.6. I2C	19
4.7. Pulse Out	20
4.8. PWM Out	21
4.9. Pin Counter	22
4.10. Quadrature In	23
4.11. Analog Inputs	24
5. Module Features	25
5.1. DIABLO16 Processor	25
5.2. Audio	25
5.3. SD/SDHC Memory Cards	26
5.4. FAT16	26
6. Display/Module Precautions	27
7. Hardware Tools	28
7.1. 4D Programming Adaptor	28
8. Programming Language	29
9. Workshop4 IDE	30
9.1. PmmC/Firmware Programming	31
10. Display Module Part Numbers	33
11. Starter Kit	34

12. Mechanical Details - Resistive Touch	35
13. Mechanical Details - Capacitive Touch	36
14. Schematic Details (HW REV 1.5)	37
15. Specifications	38
16. Revision History	41

# 1. Description

This 9.0" Diablo16 Integrated Display Module is part of the microLCD range of modules Designed and Manufactured by 4D Systems.

The 9.0" model was designed specifically to cater to the demand for physically large displays, for ease of integration and use, and with careful consideration for space requirements and functionality.

This specific module features a 9.0" colour TFT LCD, with Resistive Touch (DT), or Capacitive Touch (DCT). It is powered by the feature-rich 4D Systems Diablo16 Graphics Processor, which offers an array of functionality and options for any Designer / Integrator / User.

The Diablo16 processor features include 16 customisable GPIO, capable of PWM, Serial, I2C, SPI, Pulse Out, Quadrature Input, and Analog Input.

The 9.0" Diablo16 Integrated Display Module features a TFT LCD and is capable of Touch Detection, microSD memory Storage, GPIO and Communications, along with multiple millisecond resolution timers, and Audio Generation.

This 9.0" module is 100% compatible with the Workshop4 IDE and its 4 different development environments, providing the User with a wealth of options for programming and controlling their system.

Anything designed to run on other 4D Systems displays modules featuring Picaso or Diablo16 Graphic Processors can be run on this Integrated Display Module, with little or no required modifications. Please contact 4D Systems Support Team for assistance if migrating from a previous model.

This module features a 30-pin ZIF socket, designed for a 30-pin FPC cable, for easy and simple connection to an application or mother board, or for connecting to accessory boards for a range of functionality advancements.

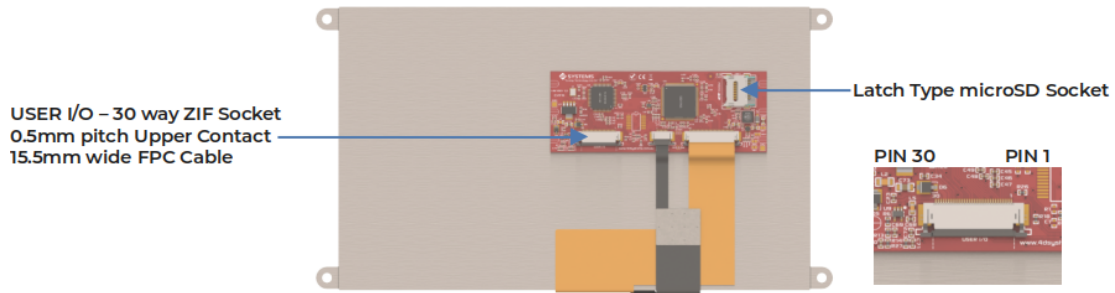
The uLCD-90DT/DCT uses the same PCB found on the popular gen4 modules, such as the gen4-uLCD-70DT, making integration and application sharing easy. Unlike the gen4 modules, the uLCD-90 range does not feature plastic frames with mounting tabs and instead integrates metal mounting ears directly into the display housing.

---

## 2. Features

- Powerful 9.0" Intelligent LCD-TFT display module powered by DIABLO16.
- 800 x 480 Resolution, RGB 65K true-to-life colours, TFT Screen with integrated 4-wire Resistive Touch Panel (DT) or Capacitive Touch Panel (DCT).
- 6 banks of 32750 bytes of Flash memory for User Application Code and Data.
- 32Kb of SRAM purely for the User.
- 16 General Purpose I/O pins for user interfacing, which include 4 configurable Analog Inputs.
- The GPIO is variously configurable for alternative functions such as:
  - 3x I2C channels available.
  - 1x SPI dedicated for SD Card and 3x configurable SPI channels available.
  - 1x dedicated and 3x configurable TTL Serial comm ports available.
  - Up to 6 GPIO can be used as Pin Counters.
  - Up to 6 GPIO for PWM (simple and Servo).
  - Up to 10 GPIO for Pulse Output.
  - Up to 14 GPIO can be configured for Quadrature Encoder Inputs (2 channels).
- 30-pin FPC connection, for all signals, power, communications, GPIO and programming.
- Onboard latch-type micro-SD memory card connector for multimedia storage and data logging purposes.
- DOS-compatible file access (FAT16 format) as well as low-level access to card memory.
- A dedicated PWM Audio pin is driven by WAV files from a micro-SD card, and for sound generation, for an external amplifier.
- Display full colour images, animations, icons and video clips.
- Supports all available Windows fonts.
- 4.0V to 5.5V range operation (single supply).
- Module dimensions:
  - (D): 230.7 x 126.4 x 6.4mm (Thickness is 3.4mm LCD + 3.0mm PCB)
  - (DT): 230.7 x 126.4 x 7.9mm (Thickness is 4.9mm LCD + 3.0mm PCB)
  - (DCT): 230.7 x 126.4 x 8.15mm (Thickness is 5.15mm LCD + 3.0mm PCB)
- 4x metal mounting tabs with 4.2mm holes for mechanical mounting using M4 screws
- RoHS, REACH and CE compliant.

### 3. Hardware Overview



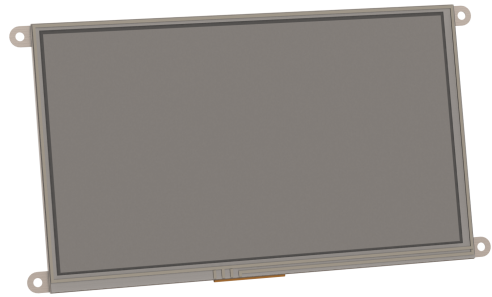
#### USER I/O – 30 Way FPC

Pin	Symbol	I/O	Description
1	GND	P	Supply Ground
2	PA3	I/O/A	General Purpose I/O pin with Analog Capability. This pin has a range of 0-3.3V when used as an Analog Input, and is 3.3V tolerant only. Also used for Bus Read
3	PA2	I/O/A	General Purpose I/O pin with Analog Capability. This pin has a range of 0-3.3V when used as an Analog Input, and is 3.3V tolerant only. Also used for Bus Write
4	PA1	I/O/A	General Purpose I/O pin with Analog Capability. This pin has a range of 0-3.3V when used as an Analog Input, and is 3.3V tolerant only. Positive Supply.
5	PA0	I/O/A	General Purpose I/O pin with Analog Capability. This pin has a range of 0-3.3V when used as an Analog Input, and is 3.3V tolerant only.
6	PA9	I/O	General Purpose Input/Output, 3.3V Level - 5V Tolerant
7	PA8	I/O	General Purpose Input/Output, 3.3V Level - 5V Tolerant
8	PA7	I/O	General Purpose Input/Output, 3.3V Level - 5V Tolerant
9	PA6	I/O	General Purpose Input/Output, 3.3V Level - 5V Tolerant
10	PA5	I/O	General Purpose Input/Output, 3.3V Level - 5V Tolerant
11	PA4	I/O	General Purpose Input/Output, 3.3V Level - 5V Tolerant
12	PA10	I/O	General Purpose Input/Output, 3.3V Level - 5V Tolerant
13	PA11	I/O	General Purpose Input/Output, 3.3V Level - 5V Tolerant
14	PA12	I/O	General Purpose Input/Output, 3.3V Level - 5V Tolerant
15	PA13	I/O	General Purpose Input/Output, 3.3V Level - 5V Tolerant
16	PA14	I/O	General Purpose Input/Output, 3.3V Tolerant only. Special I2C Pin
17	PA15	I/O	General Purpose Input/Output, 3.3V Tolerant only. Special I2C Pin.

Pin	Symbol	I/O	Description
18	NC	-	Not Connected
19	AUDIO_OUT	O	Audio Output, Filtered PWM, to feed into external amplifier
20	AUDENB	O	Audio Amplifier Enable, to enable/disable external amplifier
21	GND	P	Supply Ground
22	RESET	I	Master Reset signal. Internally pulled up to 3.3V via a 10K resistor. An active Low pulse greater than 2 micro-seconds will reset the module. If the module needs to be reset externally, only use open collector type circuits. This pin is not driven low by any internal conditions. The host should control this pin via one of its port pins using an open collector/drain arrangement.
23	RX0	I	Asynchronous Serial Receive pin, TTL level. Connect this pin to the Transmit (Tx) signal of other serial devices. Used in conjunction with the TX pin for programming this microLCD. This pin is tolerant up to 5.0V levels.
24	TX0	O	Asynchronous Serial Transmit pin, TTL level. Connect this pin to the Receive (Rx) signal of other serial devices. Used in conjunction with the RX pin for programming this microLCD. This pin is tolerant up to 5.0V levels.
25	GND	P	Supply Ground
26	5V IN	P	Main Voltage Supply +ve input pin. Reverse polarity protected. Range is 4.0V to 5.5V, nominal 5.0V.
27	5V IN	P	Main Voltage Supply +ve input pin. Reverse polarity protected. Range is 4.0V to 5.5V, nominal 5.0V.
28	NC	-	Not Connected
29	NC	-	Not Connected
30	GND	P	Supply Ground.

 **Note**

**I** = Input, **O** = Output, **P** = Power





## 4. Hardware Interface - Pins

This section describes in detail the hardware interface pins of the device.

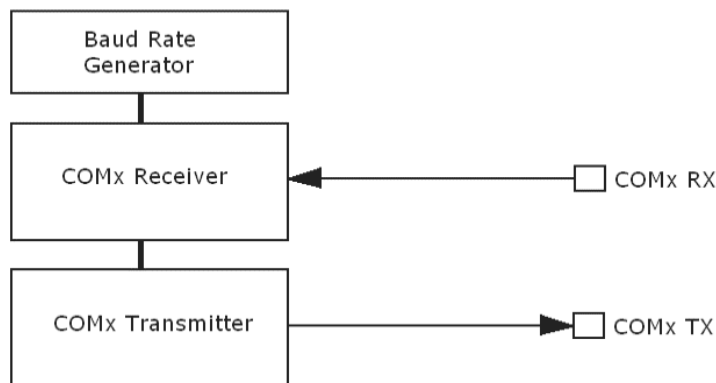
### 4.1. Serial Ports - TTL Level

The DIABLO16 Processor has three hardware asynchronous serial ports (COM1 - COM3) that can be configured on a variety of processors' GPIO pins. TX/RX0 (COM0) is dedicated and its pins are fixed. All of the DIABLO16's serial ports can be used to communicate with external serial devices.

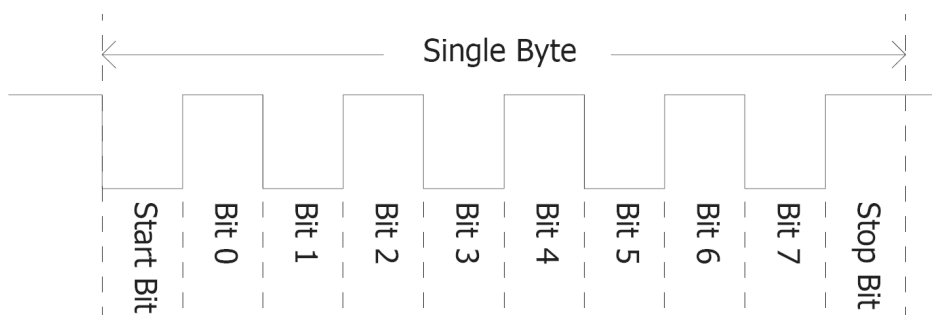
TX/RX0 is referred to as COM0 and is the only one used for programming the DIABLO16 itself.

The primary features are:

- Full-Duplex 8-bit data transmission and reception.
- Data format: 8 bits, No Parity, 1 Stop bit.
- Independent Baud rates from 300 baud up to 600K baud.
- Single byte transmits and receives a fully buffered service. The buffered service feature runs in the background capturing and buffering serial data without the user application having to constantly poll any of the serial ports. This frees up the application to service other tasks.



A single-byte serial transmission consists of the start bit, 8 bits of data followed by the stop bit. The start bit is always 0, while a stop bit is always 1. The LSB (Least Significant Bit, Bit 0) is sent out first following the start bit. The figure below shows a single-byte transmission timing diagram.



COM0 is also the primary interface for User program downloads and configuration PmmC/Driver programming. Once the compiled 4DGL application program (EVE byte-code) is downloaded and the user code starts executing, the serial port is then available to the user application.

**TX0 pin (Serial Transmit COM0):**

Dedicated Asynchronous Serial port COM0 transmit pin, TX0. Connect this pin to the external serial device receive (Rx) signal. This pin is 5.0V tolerant.

**RX0 pin (Serial Receive COM0):**

Dedicated Asynchronous Serial port COM0 receive pin, RX0. Connect this pin to an external serial device transmit (Tx) signal. This pin is 5.0V tolerant.

**TX1 pin (Serial Transmit COM1):**

Asynchronous Serial port COM1 transmit pin, TX1. Connect this pin to the external serial device receive (Rx) signal. This can be configured to one of the GPIO pins, see the [table](#) below.

**RX1 pin (Serial Receive COM1):**

Asynchronous Serial port COM1 receive pin, RX1. Connect this pin to an external serial device transmit (Tx) signal. This can be configured to one of the GPIO pins, see the [table](#) below.

**TX2 pin (Serial Transmit COM2):**

Asynchronous Serial port COM2 transmit pin, TX2. Connect this pin to the external serial device receive (Rx) signal. This can be configured to one of the GPIO pins, see the [table](#) below.

**RX2 pin (Serial Receive COM2):**

Asynchronous Serial port COM2 receive pin, RX2. Connect this pin to an external serial device transmit (Tx) signal. This can be configured to one of the GPIO pins, see the [table](#) below.

**TX3 pin (Serial Transmit COM3):**

Asynchronous Serial port COM3 transmit pin, TX3. Connect this pin to the external serial device receive (Rx) signal. This can be configured to one of the GPIO pins, see the [table](#) below.

**RX3 pin (Serial Receive COM3):**

Asynchronous Serial port COM3 receive pin, RX3. Connect this pin to an external serial device transmit (Tx) signal. This can be configured to one of the GPIO pins, see the [table](#) below.

DIABLO16 Serial TTL Comm Port Configuration Options					
	TX1	RX1	TX2	RX2	TX3
PA0		Yes		Yes	
PA1	Yes	Yes	Yes	Yes	Yes
PA2		Yes		Yes	
PA3	Yes	Yes	Yes	Yes	Yes
PA4	Yes	Yes	Yes	Yes	Yes
PA5	Yes	Yes	Yes	Yes	Yes
PA6	Yes	Yes	Yes	Yes	Yes
PA7	Yes	Yes	Yes	Yes	Yes
PA8	Yes	Yes	Yes	Yes	Yes
PA9	Yes	Yes	Yes	Yes	Yes
PA10		Yes		Yes	
PA11		Yes		Yes	
PA12	Yes	Yes	Yes	Yes	Yes
PA13	Yes	Yes	Yes	Yes	Yes
PA14					
PA15					

As per the table, not all GPIO can be configured to be every serial port, or RX/TX.

**Note**

Pins **PA4-PA13** are 5.0V tolerant, while pins **PA0-PA3, PA14, PA15** are 3.3V tolerant only. All pins output at 3.3V levels.

Please refer to the [Diablo16 Internal Functions Manual](#) datasheet for information on how to set the DIABLO16 pin mappings.

## 4.2. General Purpose I/O

There are 16 general-purpose Input/Output (GPIO) pins available to the user. These provide flexibility for individual bit operations along with serving collectively for byte-wise operations using the BUS functions

DIABLO16 Alternate Pin Configurations GPIO					
	Digital Input	Digital Output	Bus Read	Bus Write	Analog Read
PA0	Yes	Yes	Yes	Yes	Yes
PA1	Yes	Yes	Yes	Yes	Yes
PA2	Yes	Yes	Yes	Yes	Yes
PA3	Yes	Yes	Yes	Yes	Yes
PA4	Yes	Yes	Yes	Yes	
PA5	Yes	Yes	Yes	Yes	
PA6	Yes	Yes	Yes	Yes	
PA7	Yes	Yes	Yes	Yes	
PA8	Yes	Yes	Yes	Yes	
PA9	Yes	Yes	Yes	Yes	
PA10	Yes	Yes	Yes	Yes	
PA11	Yes	Yes	Yes	Yes	
PA12	Yes	Yes	Yes	Yes	
PA13	Yes	Yes	Yes	Yes	
PA14	Yes	Yes	Yes		
PA15	Yes	Yes	Yes		

Please refer to the [Diablo16 Internal Functions Manual](#) datasheet for information on how to set the DIABLO16 pin mappings.

### PA0-PA3:

General purpose I/O pins, or can serve as Analog Input pins. Each pin can be individually set for INPUT or OUTPUT or ANALOG. Power-Up Reset default is all INPUTS. Digital GPIO can source/sink 10mA. For more information, the [Specifications](#) section. These pins have a **0** to **3.3V** range and have a 12-bit resolution.

For more information, refer to the [Analog Inputs Section](#).

**PA4-PA11:**


General purpose I/O pins. Each pin can be individually set for INPUT or OUTPUT. Power-Up Reset default is all INPUTS. When set as Digital Inputs, the pins are 5V tolerant. Digital GPIO can source/sink 10mA. For more information, see the [Specifications](#) section.

**PA12-PA13:**

PA12 and PA13 are reserved and are used exclusively for the resistive touch controller, or the capacitive touch controller (DT/DCT models). These pins should not be connected else Touch performance may be degraded. There are 4.7K pullup resistors on each of this GPIO. Avoid these pins in normal situations. However, using these pins on non-touch (D model) is OK, or in situations where touch is not required. These pins are 5V tolerant

**PA14-PA15:**

General purpose I/O pins. Each pin can be individually set for INPUT or OUTPUT. Power-Up Reset default is all INPUTS. When set as Digital Inputs, the pins are 3.3V tolerant. Digital GPIO can source/sink 10mA. Under special situations, these pins can be used for a high-speed I2C interface. Please refer to the [table](#) under the [Alternate Pin Function Overview](#) section and the information under the [I2C](#) section.

 **Note**

Digital GPIO pins are 5.0V tolerant, except PA14-PA15, which are 3.3V tolerant only. PA0-PA3 are 3.3V tolerant only. All pins output at 3.3V levels.

### 4.3. System Pins

#### **+5V IN (Module Voltage Input):**

Module supply voltage input pins. Both of these pins should be connected to a stable supply voltage in the range of 4.0 Volts to 5.5 Volts DC. The nominal operating voltage is 5.0 Volts. Note backlight brightness will be lower for voltages under 5.0V.

#### **GND (Module Ground):**

Device ground pins. At least two pins should be connected to the ground.

#### **RESET (Module Master Reset):**

Module Master Reset pin. An active low pulse of greater than 2 $\mu$ s will reset the module. Internally pulled up to 3.3V via a 10K resistor. Only use open collector-type circuits to reset the device if an external reset is required.

#### **AUDENB (Audio Enable Output):**

Output dedicated to enabling or disabling an external amplifier, where required.

#### **AUDIO (PWM Audio Output):**

Output specifically for Audio. This pin is a PWM output from the DIABLO16 Processor. This pin is a 3.3V level PWM output to drive an external amplifier with DIABLO16 generated Audio, via an external filtering circuit to turn digital PWM into Analog. Note that AUDENB must be enabled in DIABLO16 for any input signal to be heard.

## 4.4. Alternate Pin Function Overview

Most of the GPIO pins have an alternate function other than being for General Purpose I/O.

GPIO pins can be configured to be SPI, I<sup>2</sup>C, Serial or a range of other functions.

### Note

Not all pins can be configured to be any of the alternate pin functions. Please refer to the tables.

Refer to the tables below that illustrate the GPIO pins you can associate with alternative functions.

### DIABLO16 Alternate Pin Configurations I/O Support Functions

	Pulse Out	PWM Out	Pin Counter	Quadrature In
<b>PA0</b>	Yes			Yes
<b>PA1</b>	Yes			Yes
<b>PA2</b>	Yes			Yes
<b>PA3</b>	Yes			Yes
<b>PA4</b>	Yes	Yes	Yes	Yes
<b>PA5</b>	Yes	Yes	Yes	Yes
<b>PA6</b>	Yes	Yes	Yes	Yes
<b>PA7</b>	Yes	Yes	Yes	Yes
<b>PA8</b>	Yes	Yes	Yes	Yes
<b>PA9</b>	Yes	Yes	Yes	Yes
<b>PA10</b>				Yes
<b>PA11</b>				Yes
<b>PA12</b>				Yes
<b>PA13</b>				Yes
<b>PA14</b>				
<b>PA15</b>				

### Note

- Once you allocate a pin to an alternate function, you can't allocate another pin to the same alternate function.
- Quadrature In requires 2 Pins


The [table](#) above illustrates the GPIO pins you can use for the 4 different I/O Support Functions.

Please refer to the [Diablo16 Internal Functions Manual](#) datasheet for more information on how to set the alternate pin configurations.

The Alternate pin functions have been broken up into a few tables for simplification. There are communication-based functions and I/O support-based functions.


Further information is available in the next sections for each of the alternative pin functions.

The table below illustrates the GPIO pins you can use for the three different SPI channels available.

 <b>DIABLO16 Alternate Pin Configurations SPI Communications</b>					
	<b>SPI1 SDO</b>	<b>SPI1 SDI</b>	<b>SPI1 SCK</b>	<b>SPI2 SDO</b>	<b>SPI2 SDI</b>
<b>PA0</b>		Yes			Yes
<b>PA1</b>	Yes	Yes	Yes	Yes	Yes
<b>PA2</b>		Yes			Yes
<b>PA3</b>	Yes	Yes	Yes	Yes	Yes
<b>PA4</b>	Yes	Yes	Yes	Yes	
<b>PA5</b>	Yes	Yes	Yes	Yes	
<b>PA6</b>	Yes	Yes	Yes	Yes	
<b>PA7</b>	Yes	Yes	Yes	Yes	
<b>PA8</b>	Yes	Yes	Yes	Yes	
<b>PA9</b>	Yes	Yes	Yes	Yes	
<b>PA10</b>		Yes			Yes
<b>PA11</b>		Yes			Yes
<b>PA12</b>	Yes	Yes	Yes	Yes	
<b>PA13</b>	Yes	Yes	Yes	Yes	
<b>PA14</b>					
<b>PA15</b>					



The table below illustrates the GPIO pins you can use for the three different I<sup>2</sup>C channels available.

 <b>DIABLO16 Alternate Pin Configurations I<sup>2</sup>C Communication</b>					
	<b>I<sup>2</sup>C1 SDA</b>	<b>I<sup>2</sup>C1 SCL</b>	<b>I<sup>2</sup>C2 SDA</b>	<b>I<sup>2</sup>C2 SCL</b>	<b>I<sup>2</sup>C3 SDA</b>
<b>PA0</b>	Yes	Yes	Yes	Yes	Yes
<b>PA1</b>	Yes	Yes	Yes	Yes	Yes
<b>PA2</b>	Yes	Yes	Yes	Yes	Yes
<b>PA3</b>	Yes	Yes	Yes	Yes	Yes
<b>PA4</b>	Yes	Yes	Yes	Yes	Yes
<b>PA5</b>	Yes	Yes	Yes	Yes	Yes
<b>PA6</b>	Yes	Yes	Yes	Yes	Yes
<b>PA7</b>	Yes	Yes	Yes	Yes	Yes
<b>PA8</b>	Yes	Yes	Yes	Yes	Yes
<b>PA9</b>	Yes	Yes	Yes	Yes	Yes
<b>PA10</b>	Yes	Yes	Yes	Yes	Yes
<b>PA11</b>	Yes	Yes	Yes	Yes	Yes
<b>PA12</b>	Yes	Yes	Yes	Yes	Yes
<b>PA13</b>	Yes	Yes	Yes	Yes	Yes
<b>PA14</b>		SPECIAL		SPECIAL	
<b>PA15</b>	SPECIAL		SPECIAL		SPECIAL

 **Note**

SPECIAL - please see I<sup>2</sup>C section.

## 4.5. SPI

There are 3 user-configurable SPI channels available for mapping to GPIO, for use by the user for the target application. All 3 SPI channels are Master only, and cannot be configured to be slaves at this time.

The SPI Bus speed is configurable using the **SPIx\_Init()** Function in 4DGL and allows various speeds from 78.125Khz to 17.5Mhz.

Please refer to the [table](#) for details on which GPIO can be configured for SPI.

### Note

The additional SPI channel (SPI0) is dedicated to the memory cards and cannot be reconfigured for alternate uses.

To map an SPI channel to a set of GPIO pins, the following 4DGL functions are used:

- **SPIx\_SCK\_pin(pin);** // Map the SCK pin
- **SPIx\_SDI\_pin(pin);** // Map the SDI pin
- **SPIx\_SDO\_pin(pin);** // Map the SDO pin

where:

'SPIx' is substituted with SPI1, SPI2 or SPI3 accordingly, and

'pin' is the target GPIO pin compatible with that particular pin function.

Chip Select for use with SPI can be any other unused GPIO pin, configured as a Digital Output. The lowering and raising of the selected CS (GPIO) pin is done manually by the user in the 4DGL application.

Please refer to the [Diablo16 Internal Functions Manual](#) datasheet for more information on how to use the SPI functions, along with the [DIABLO16-Processor](#) datasheet.

## 4.6. I2C

There are 3 user-configurable I2C channels available for mapping to GPIO, for use by the user for the target application. All 3 I<sup>2</sup>C channels are Master only, and cannot be configured to be slaves at this time.

Please refer to the [table](#) for details on which GPIO can be configured for I<sup>2</sup>C.

To map an I<sup>2</sup>C Channel to a set of GPIO pins, the following 4DGL function is used:

```
· I2Cx_Open(Speed, SCLpin, SDApin);
```

where:

'I2Cx' is substituted with I2C1, I2C2 or I2C3 accordingly,

'Speed' is the desired I2C Bus speed, and

'SCLpin' and 'SDApin' are the target GPIO pins compatible with that particular pin function.

### Note

The normal I<sup>2</sup>C pins are PA0 to PA13, however, the use of these pins has a few limitations.

1. There is no slew rate control at I2C\_MED.
2. I2C\_FAST is not truly 1MHz.

If either of these restrictions needs to be addressed, a special case of SCLpin = PA14 and SDApin = PA15 exists ONLY for speeds I2C\_MED (which uses slew rate control) and I2C\_FAST (which is truly 1MHz)

Please refer to the [Diablo16 Internal Functions Manual](#) datasheet for more information on how to use the I<sup>2</sup>C functions, along with the [DIABLO16-Processor](#) datasheet.

## 4.7. Pulse Out

Pulse Out is used to create a single pulse of set duration on the selected pin of choice, which is inverted in polarity to the current state of the pin.

This '*inversion of polarity*' means if a Pin is currently held HI, and Pulse Out is executed on that Pin, the pin will pulse LO and then return to HI. Same with vice versa, if currently LO and Pulse Out are executed on that Pin, it will pulse HI and then return to LO.

This is available in both blocking and non-blocking versions.

Please refer to the [table](#) for details on which GPIO can be configured to Pulse Out.

### Note

Each Pulse Out request needs at least a 1ms lead time due to the scheduling of the event with the internal 1ms timer.

To enable the Pulse Out function on a GPIO pin, the following 4DGL functions are used:

- **pin\_Pulseout(pin, value);** //Non-Blocking
- **pin\_PulseoutB(pin, value);** //Blocking

where:

'pin' is the target GPIO pin compatible with that particular pin function, and

'value' is the length of the pulse in milliseconds.

Please refer to the [Diablo16 Internal Functions Manual](#) datasheet for more information on how to use the Pulse Out functions, along with the [DIABLO16-Processor](#) datasheet.

---

## 4.8. PWM Out

There are 6 PWM channels available to be configured by the user, with 4-time bases available for selection. The PWM can be configured to be used in Servo Mode, or Simple Mode.

Please refer to the [table](#) for details on which GPIO can be configured to PWM.

Servo Mode allows a millisecond input value with 0.01ms resolution, which runs at a frequency of approximately 50Hz or 50pps (20ms). The position of the servo is determined by the width of the pulse. Generally, 1.5ms is 90 degrees, 1ms is 0 degrees and 2ms is 180 degrees. Servos however vary, and the DIABLO16 PWM control can be adjusted to suit most applications.

Simple Mode allows a percentage input value with a resolution of 0.1%, which runs at a frequency of approximately 70KHz.

To enable the PWM output on a GPIO pin, the following 4DGL function is used:

· **PWM\_Init(pin, mode, value);**

where:

'pin' is the GPIO compatible with the particular pin function,

'mode' is the type of PWM to generate, and

'value' is the parameter that defined the PWM pulse itself.

Please refer to the [Diablo16 Internal Functions Manual](#) datasheet for more information on how to use the PWM functions, along with the [DIABLO16-Processor](#) datasheet.

---

## 4.9. Pin Counter

There are 6 Pin Counter channels available to be configured by the user, used to count incoming pulses with the ability to call a user function on overflow. The Pin Counter function is available for use in a variety of modes.

The counters can be read and written at any time.

Please refer to the [table](#) for details on which GPIO can be configured to Pin Counter.

To enable the Pin Counter function on a GPIO pin, the following 4DGL function is used:

· **pin\_Counter(pin, mode, OVFunction);**

where:

'pin' is the GPIO pin compatible with this particular function,

'mode' is the type of trigger used to count on such as Rising/Falling/Edge, and

'OVFunction' is the user function to call when the counter overflows if desired.

Please refer to the [Diablo16 Internal Functions Manual](#) datasheet for more information on how to use the Pin Counter functions, along with the [DIABLO16-Processor](#) datasheet.

## 4.10. Quadrature In

There are two Quadrature Input channels available on the DIABLO16 processor, which requires 2 GPIO pins each.

Please refer to the [table](#) for details on which GPIO can be configured for Quadrature Input.

Quadrature Input allows a quadrature encoder to be connected, and the position counter and delta counter can be read at any time.

To enable the Quadrature Input function on a set of GPIO pins (2 pins required), the following 4DGL function is used:

· **Qencoderx(PHApin, PHBpin, mode);**

where:

'Qencoderx' is substituted for Quencoder1 or Quencoder2 accordingly,

'PHApin' is the pin connected to the A Phase of the Encoder,

'PHBpin' is the pin connected to the B Phase of the Encoder, and

'mode' is not currently used so is to be set to zero (0).

Please refer to the [Diablo16 Internal Functions Manual](#) datasheet for more information on how to use the Quadrature Input functions, along with the [DIABLO16-Processor](#) datasheet.

## 4.11. Analog Inputs

Please refer to the [table](#) under the [General Purpose I/O](#) section for details about pins you can configure to analog inputs.

The analog inputs on the DIABLO16 have a range of 0V to 3.3V, each with a max resolution of 12 bits. You can read the analog inputs using either the standard mode, averaged mode or high-speed mode.

- **Standard Mode** results are immediately read in a sample. Standard Mode can read over 40000 values per second. Operates at 12-bit.
- **Averaged Mode** results are immediately read in 16 samples and their average is returned. Averaged Mode can read ~20000 values per second. Operates at 12-bit.
- **Highspeed Mode** collects a user-specified number of samples at a user-specified rate/frequency and can execute a user function when complete. The updated value updates ~250000 times across 1-4 channels. Operates at 10-bit.

To enable a GPIO as an Analog Input for Standard or Averaged modes, you must use the following 4DGL function to set the pin:

- **pin\_Set(mode, pin);**

where:

'mode' is the desired mode defined above, either Standard or Averaged, and

'pin' is the GPIO compatible with this function which is to become an Analog Input.

For highspeed mode, you must the following 4DGL function to set the pin and define the parameters:

- **ana\_HS(rate, samples, 1buf, 2buf, 3buf, 4buf, func);**

where:

'rate' is the number of samples per second,

'samples' is the number of samples to collect per channel,

'1buf' to '4buf' are the buffer addresses for the 4 channels, and

'func' is the user function to call when the number of samples specified has been collected.

Please refer to the [Diablo16 Internal Functions Manual](#) for more information on how to use the Analog Input functions, along with the [DIABLO16-Processor](#) datasheet.

### Note

Analog Inputs are 0V-3.3V tolerant only. Do not apply voltages outside of this range as you will damage Diablo16.



## 5. Module Features

The uLCD-90DT/CT displays are designed to accommodate most applications where a large physical size display module is required. Some of the main features of the module are listed below.

### 5.1. DIABLO16 Processor

The module is designed around the DIABLO16 Graphics Controller from 4D-Labs.



The DIABLO16 is a smart Controller and the interface to the TFT-LCDs is almost plug-n-play.

All of the data and control signals are provided by the chip to interface directly with the display.

Powerful graphics, text, images, animation and countless more features are built right inside the chip.

You can refer to the [DIABLO16 graphics processor datasheet](#) for more information.

### 5.2. Audio

Audio playback support in the DIABLO16 Processor enables this module to play audio WAV files stored in the micro-SD memory card. Filtered PWM audio is generated and made available on the AUDIO pin of the 30-way ZIF connector, ready to interface with an audio amplifier on your application board.

A simple instruction enables the user to play/pause/stop audio files while continuing the execution of the user application code, such as display updates, touch recognition, communications, etc. The audio system also allows real-time pitch change of audio samples.

For a complete list of audio commands please refer to the separate document titled:

- [DIABLO16 4DGL Internal Functions](#)

### 5.3. SD/SDHC Memory Cards

The module supports micro-SD memory cards via the onboard latch-type micro-SD connector. The memory card is used for all multimedia file retrieval such as images, animations and movie clips. The memory card can also be used as general-purpose storage for data logging applications. Support is available for off-the-shelf micro-SD (< 4GB) and high-capacity HC memory cards (4GB and above).



Memory cards up to 32GB in size can be used, however, it must be noted that only a portion of this can be used by the FAT16 file system. See the [FAT16](#) section below for more details.

#### Note

A microSD card capable of SPI is a requirement for all 4D Systems' display modules powered by Goldelox, Picaso or Diablo16 Processors. If a non-SPI compatible card is used, it will simply fail to mount, or may cause intermittent issues resulting in lockups and crashing of the application. Please refer to the [4D Systems website](#) for microSD cards offered by 4D Systems.

### 5.4. FAT16

All 4D Systems display modules featuring 4D Labs processors use off-the-shelf standard SDHC/SD/micro-SD memory cards (SPI Compatible Only) with up to 4GB capacity usable with FAT16 formatting.

For any FAT file-related operations, before the memory card can be used it must first be formatted correctly. Built into Workshop4 is a tool created by 4D, called RMPET (please refer to the Tools menu, in any Environment, inside the Workshop4 IDE). RMPET allows the User to easily partition and format microSD cards, to make their file system ready to be used with 4D Systems modules. The formatting of the card can be done on any PC system with a card reader.

The DIABLO16 Processor also supports high-capacity HC memory cards (4GB and above). The available capacity of SD-HC cards varies according to the way the card is partitioned and the commands used to access it. Refer to the 4D Systems RMPET tool in the Workshop4 IDE.

A Max of 4GB can be used by the FAT16 file system. The FAT partition is always first (if it exists). Any space larger than 4GB will be RAW, and can still be used by your 4D Systems module, using different functions. Please refer to the [Application Notes](#).

---

## 6. Display/Module Precautions

- Avoid having to display the same image/object on the screen for lengthy periods. This can cause a burn-in which is a common problem with all types of display technologies. Blank the screen after a while or dim it very low by adjusting the contrast. Better still; implement a screen saver feature.
- Moisture and water can damage the display. Moisture on the surface of a powered display should not cause any problems, however, if water is to enter the display either from the front or from the rear, or come in contact with the PCB, it will damage. Wipe off any moisture gently or let the display dry before usage. If using this display module in an environment where it can get wet, ensure an appropriate enclosure is used.
- Dirt from fingerprint oil and fat can easily stain the surface of the display. Gently wipe off any stains with a soft lint-free cloth.
- The performance of the display will degrade under high temperatures and humidity. Avoid such conditions when storing.
- Do not tamper with the display flex cable that is connected to the control board. This may affect the connection between the display and the driving circuitry and cause failure.
- Displays are susceptible to mechanical shock and any force exerted on the module may result in deformed zebra stripes, a cracked display cell and a broken backlight.
- Always use the mounting holes on the module's corner plates to mount the display where possible.
- Display modules have a finite life, which is typically dictated by the display itself, more specifically the backlight. The backlight contains LEDs, which fade over time. In the [Specifications section](#) is a figure for the typical life of the display, and the criteria are listed. The resistive Touch model features a touch-sensitive film over the display which is sensitive to pressure. When mounting the display module in an enclosure, you should not apply pressure to the surface of the display by the enclosure. It could result in false touches or the touch will simply not function at all.

## 7. Hardware Tools

The following hardware tool is required for full control of the uLCD-90DT/DCT display module.

### 7.1. 4D Programming Adaptor

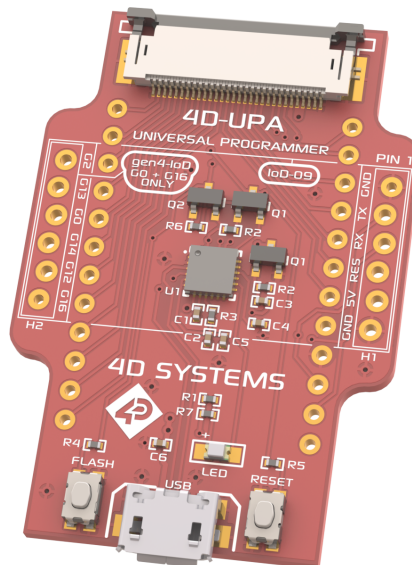
The 4D-UPA (Universal Programming Adaptor) is an essential hardware tool to program, customise and test the DIABLO16 Processor.

The 4D programming interfaces are used to program a new Firmware/PmmC, Display Driver and for downloading compiled 4DGL code into the processor. They even serve as an interface for communicating serial data to the PC.

The 4D-UPA Programming Adaptor are available from the [4D Systems website](#).

Using a non-4D programming interface could damage your processor, and **void your Warranty**.

The 4D-UPA is an all-in-one programming board, compatible with many 4D Systems products.



The 4D-UPA is a single module with a micro USB interface, and DIP-style pads for GPIO breakout of all the signals used on the Display interface, which is useful for development or final product use. Standard 2.54mm (0.1") male pin headers can be mounted so the module signals can be interfaced to easily.

Please refer to the [4D-UPA datasheet](#) for more information.

## 8. Programming Language

The DIABLO16 processor belongs to a family of processors powered by a highly optimised softcore virtual engine called EVE (Extensible Virtual Engine). EVE was designed and created by 4D Systems in the early 2000s, and should not be confused with FTDI's solution of EVE, which was developed a decent decade or so later.

EVE is a proprietary, high-performance virtual machine with an extensive byte-code instruction set optimised to execute compiled 4DGL programs.

4DGL (4D Graphics Language) was specifically developed from the ground up for the EVE engine core. It is a high-level language that is easy to learn and simple to understand yet powerful enough to tackle many embedded graphics applications.

4DGL is a graphics-oriented language allowing rapid application development, and the syntax structure was designed using elements of popular languages such as C, Basic, Pascal and others.

Programmers familiar with these languages will feel right at home with 4DGL. It includes many familiar instructions such as `IF..ELSE..ENDIF`, `WHILE..WEND`, `REPEAT..UNTIL`, `GOSUB..ENDSUB`, `GOTO`, `PRINT` as well as some specialised instructions `SERIN`, `SEROUT`, `GFX_LINE`, `GFX_CIRCLE` and many more.

For detailed information pertaining to the 4DGL language, please refer to the following documents:

- [4DGL Programmers Reference Manual](#)
- [Diablo16 Internal Functions Manual](#)

To assist with the development of 4DGL applications, the Workshop4 IDE combines a full-featured editor, a compiler, a linker and a downloader into a single PC-based application. It's all you need to code, test and run your applications.

4DGL is available to be written in two of the four environments offered by the Workshop4 IDE, Designer and ViSi. The other two environments, Serial and ViSi-Genie do not directly use 4DGL by the User (Except in Workshop4 Pro, for ViSi-Genie), however, it is present in the background. Serial is an application that runs, and that is written in 4DGL. ViSi-Genie automatically generates 4DGL itself based on what is configured in the GUI. More information about each follows.

## 9. Workshop4 IDE

Workshop4 is a comprehensive software IDE that provides an integrated software development platform for all of the 4D family of processors and modules. The IDE combines the Editor, Compiler, Linker and Downloader to develop complete 4DGL application code. All user application code is developed within the Workshop4 IDE.



The Workshop4 IDE supports multiple development environments for the user, to cater to different user requirements and skill levels.

- The **Designer** environment enables the user to write 4DGL code in its natural form to program the range of 4D System's intelligent displays.
- A visual programming experience, suitably called **Visi**, enables drag-and-drop type placement of objects to assist with 4DGL code generation and allows the user to visualise how the display will look while being developed.
- An advanced environment called **Visi-Genie** doesn't require any 4DGL coding at all, it is all done automatically for you. Simply lay the display out with the objects you want, set the events to drive them and the code is written for you automatically. This can be extended with additional features when a Workshop4 PRO license is purchased from the 4D Systems website. Extended Advanced features for Visi-Genie are available in the PRO version of WS4. Further details are explained in the [Visi Genie](#) section of the Workshop4 documentation.
- A **Serial** environment is also provided to transform the display module into a slave serial module, allowing the user to control the display from any host microcontroller or device with a serial port.

For more information regarding these environments, refer to the [Workshop4 manuals](#).

The Workshop 4 IDE is available from the [4D Systems website](#).

---

## 9.1. PmmC/Firmware Programming

The DIABLO16 processor is a custom graphics processor. All functionality including the high-level commands is built into the chip. This chip-level configuration is available as a PmmC (Personality-module-micro-Code) file, which can be likened to traditional Firmware. There is also a Display Driver file, which separates specific display settings from the PmmC, unlike on the PICASO processor where everything is combined.

A PmmC file contains all of the low-level micro-code information (analogy of that of a soft silicon) which defines the characteristics and functionality of the device. The ability of programming the device with a PmmC file provides an extremely flexible method of customising as well as upgrading it with future enhancements.

The Display Driver contains the initialisation and parameters associated with the particular display that is to be connected to the DIABLO16 processor, along with product-specific settings and parameters which are required over and above what is found in the PmmC.

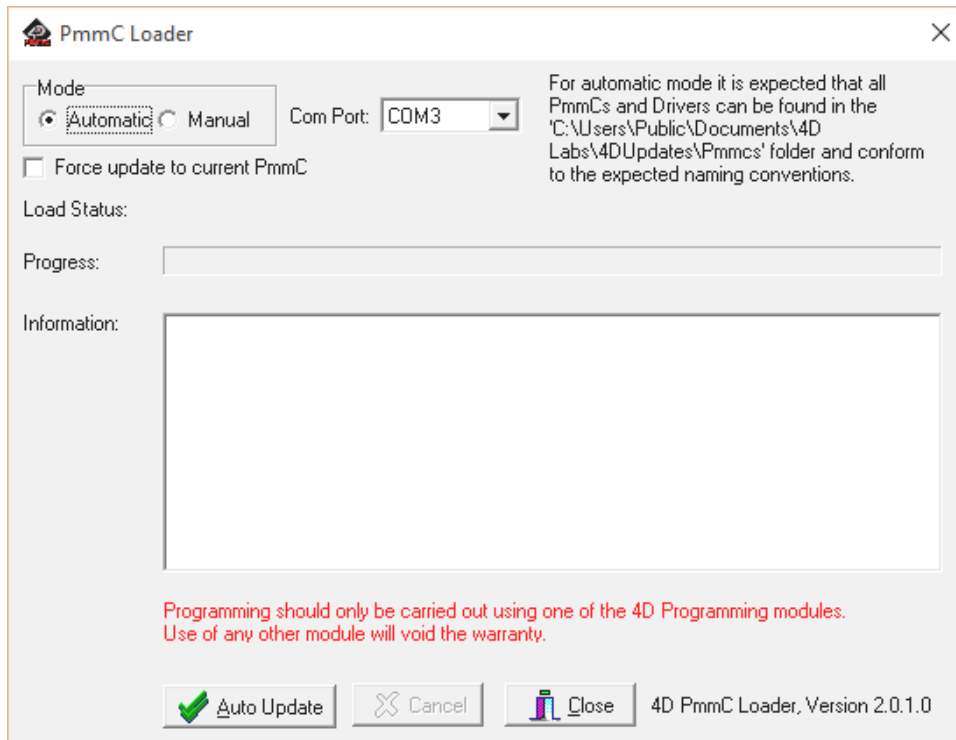
The PmmC file and Display Driver file can only be programmed into the device via the COM0 serial port with the aid of Workshop 4, the 4D Systems IDE software.

Solutions, which remove the need to use Workshop4 to program the display modules, are available for commercial customers requiring batch programming or production line programming. These solutions are practical for production staff and minimize the chance of unwanted or unauthorised modifications on the production line.

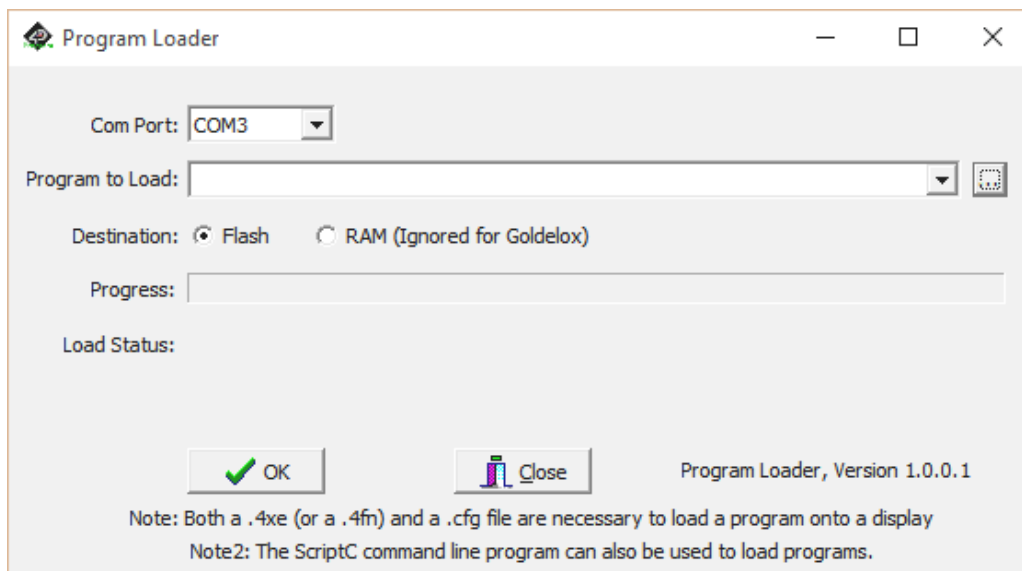
Three solutions come with the Workshop4 IDE, which can be separated for production line computers if required, **ScriptC**, **PmmC Loader** and **Program Loader**.

**ScriptC** is a command line interface that can be controlled from a Batch Script or similar, useful for repetitive loading or when controlled from an external application. Example scripts are provided, showing how to load PmmC, Display Drivers and Applications using a batch script.

**Program Loader** is a GUI interface designed to download Applications to either Flash or RAM, useful for testing or production loading, without the need for Workshop4 itself.



**PmmC Loader** is a GUI interface designed to download PmmC's and Drivers to 4D Systems Processors. It can automatically update existing PmmC's and Drivers present on a module, or manually change or force download to a blank processor or module, overwriting anything written in previously.



Please contact our Support Team for more information on what we can provide. Some solutions also exist for external processor-based loading, if there is a requirement for that. This solution is available under NDA. Using a non-4D programming interface could damage your module, and **void your warranty**.



## 10. Display Module Part Numbers

The following is a breakdown of the part numbers and what they mean.

### Example:

· uLCD-90DCT

uLCD - microLCD Display Family

90 - Display size (9.0")

D - Diablo16 Processor

T - Resistive Touch

CT - Capacitive Touch



### Note

- For part numbers that do not include T or CT, these are non-touch variants.
- Resistive Touch relies on the mechanical flexing of a membrane to trigger touch, unlike Capacitive Touch which is glass and detects your finger with little or no pressure requirement.

---

## 11. Starter Kit

4D Systems highly recommends all first-time buyers of 4D Systems' displays, to purchase the Starter Kit when purchasing their first 4D Systems display solution.

The Starter Kit provides all the hardware that is required to get the User up and running.

Not all development environments and features will be needed by every User. However, purchasing the display solution in a Starter Kit allows you to take full advantage of all of the features of the 4D Systems Display Solution and try out each of the 4D Workshop4 Environments before settling with the preferred feature set.

The **Designer** environment can use every feature of the display, however, depending on the user requirements, a micro-SD (uSD) card may not be required. The uSD card is used when displaying images/video/sound, along with data logging to uSD, and a programming cable or adaptor is required for downloading compiled code and PmmC/Firmware updates.

The **Visi** environment is the same as Designer in terms of feature utilisation, but is image based so requires a uSD card, along with a programming cable.

The **Visi-Genie** environment is also image-based, and therefore requires a uSD card and programming cable also.

The **Serial** environment does not require either a uSD or Programming cable to be used once the module has been configured as a Serial device, however, can use both depending on the user's requirements. The uSD card can be used for such things as storage and display of multimedia files, data logging, and the Programming cable for PmmC/Firmware updates, or changing to one of the other three programming environments.

Starter Kits typically include:

- uLCD-90DT/DCT Integrated Display Module
- Interface Module (gen4-IB)
- 4D Universal Programming Adaptor (4D-UPA)
- 4GB micro-SD Card
- 5-way cable for easy connection to a breadboard or host via the gen4-IB
- 150mm 30-way FFC cable for connecting the display to gen4-IB or 4D-UPA
- Quick Start Guide

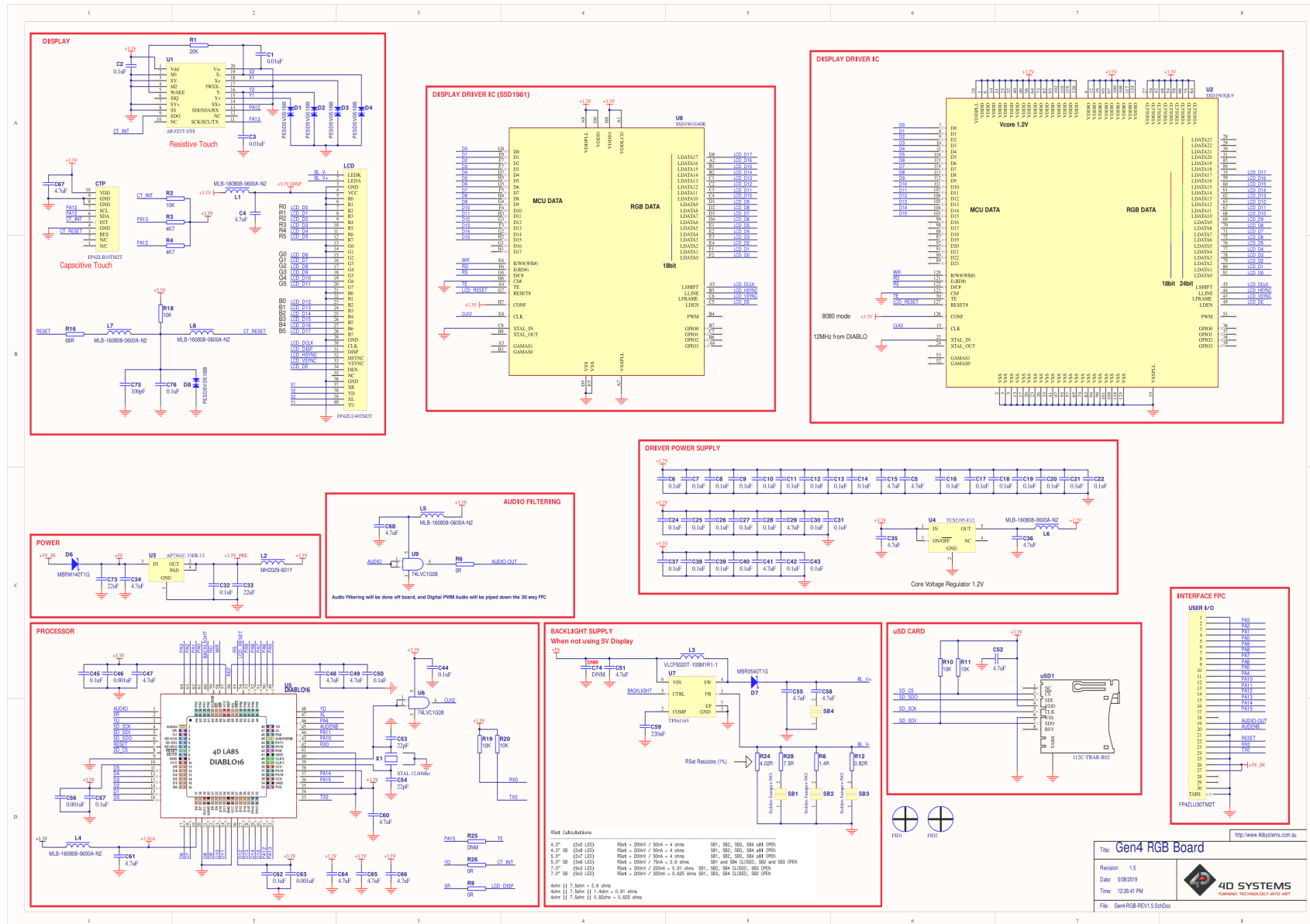
Please refer to the [4D Systems website](#) for the current components included in the Starter Kit.

Simply select the Starter Kit option when purchasing the chosen display module on the 4D Systems shopping cart, or from your local distributor.





# 14. Schematic Details (HW REV 1.5)



## 15. Specifications

### Absolute Maximum Ratings

Operating ambient temperature	-20°C to +70°C
Storage temperature	-30°C to +80°C
Voltage on any digital input pin with respect to GND	-0.3V to 6.0V
Voltage on VCC with respect to GND	-0.3V to 6.0V
Maximum current sunk/sourced by any pin	10.0mA
Maximum current sunk/sourced by all ports	200.0mA

### Note

Stresses above those listed here may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the recommended operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

### Recommended Operating Conditions

Parameter	Conditions	Min	Typ	Max	Unit
Supply Voltage (VCC)	Stable external supply required	4.0	5.0	5.5	V
Processor voltage (VP)		--	3.3	--	V
Operating Temperature		-10	--	+60	°C
Input Low Voltage (VIL)	all pins	0	--	0.2VP	V
Input High Voltage (VIH)	non 5V tolerant pins	0.8VP	--	3.3	V
Input High Voltage (VIH)	PA4-PA13, RX0 and TX0 pins	0.8VP	--	5.5	V
Reset Pulse	External Open Collector	2.0	--	--	μs
Operational Delay	Power-Up or External Reset	500	--	3000	ms


**Global Characteristics Based on Operating Conditions**

Parameter	Conditions	Min	Typ	Max	Units
Supply Current(I <sub>CC</sub> ) (5V Supply used)	uLCD-90D (Contrast = 15)	9	760	--	mA
	uLCD-90DT (Contrast = 15)	9	766	--	mA
	uLCD-90DCT (Contrast = 15)	9	784	--	mA
Display Endurance	Hours of operation, measured to when the display is 50% original brightness	30000	--	--	Hours
Touch Screen Endurance (Resistive Touch)	Number of touches/hits with a 12.5mm tip at a rate of 2x per second with 250gf force	--	1M	--	Touches
	Slide stylus on screen, 100gf force, 60mm/s speed with a 0.8mm polyacetal tip stylus pen	--	100K	--	Slides
Touch Screen Transparency	Resistive Touch	82	--	--	%
	Capacitive Touch	90	--	--	%
Touch Screen Operational Force (Resistive Touch)	Only use Finger or Stylus, do not use anything sharp or metal	20	--	100	CentiNewtons
CTP Hardness	Glass Hardness	--	6	--	Hardness

**LCD DISPLAY INFORMATION**

Parameter	Conditions	Specification
Display Type		TFT Transmissive LCD
Display Size		9.0" Diagonal
Display Resolution		800 x 480 (Landscape/Wide Viewing)
Display Brightness	uLCD-90D (Contrast = 15) - Typical	500 cd/m2
	uLCD-90DT (Contrast = 15) - Typical	450 cd/m2
	uLCD-90DCT (Contrast = 15) - Typical	475 cd/m2
Display Contrast Ratio	Typical	500:1
Display Viewing Angles	Above Centre	50 Degrees
	Below Centre	70 Degrees
	Left of Centre	70 Degrees
	Right of Centre	70 Degrees
Display Viewing Direction		6 O'clock Display (Optimal viewing is from below when in Landscape/Wide mode)
Display Backlighting	White LED's	11x3 Parallel LED's (33 LED's)
Pixel Pitch	Width x Height – Landscape mode	0.2475 x 0.2327mm (non-Square pixels)
Pixel Density	Number of pixels in 1 row in 25.4mm	103 DPI/PPI (Horizontal)
		109 DPI/PPI (Vertical)