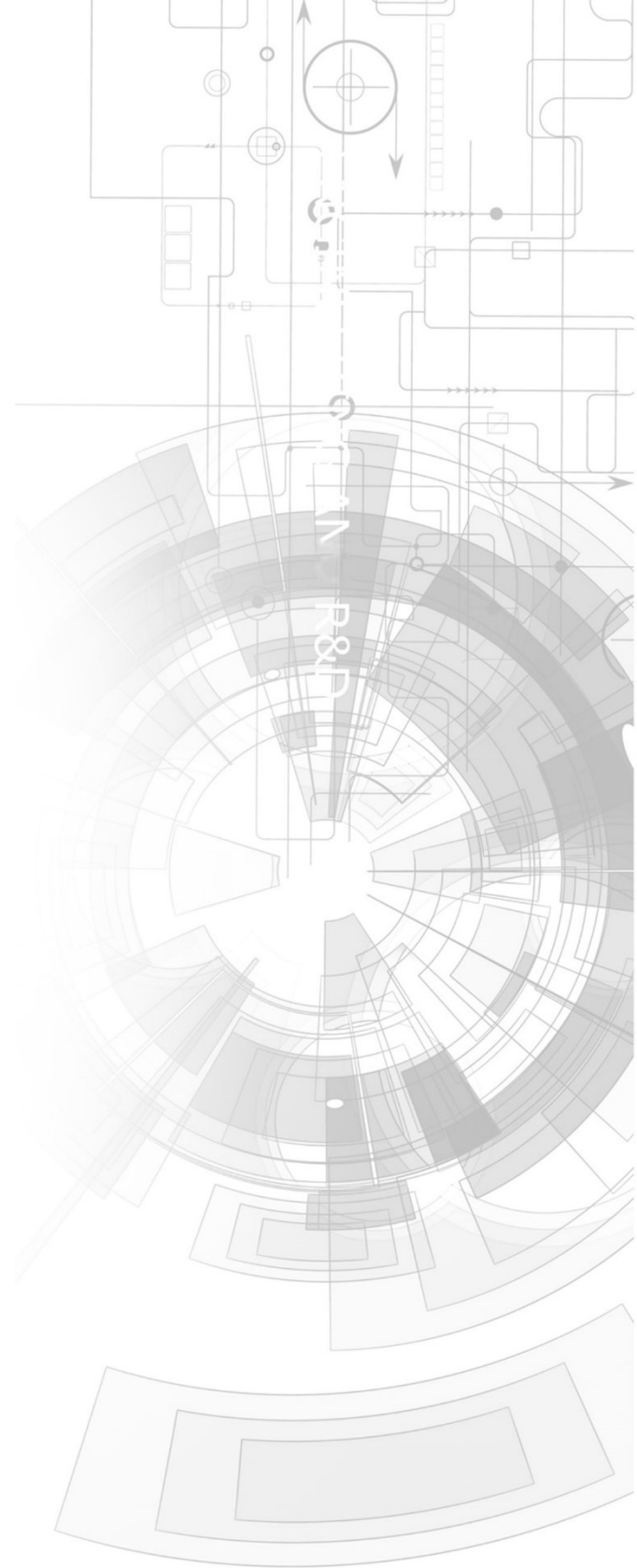


# uLCD-220RD



## Datasheet

Revision 1.5

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. Pin Configuration and Summary	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 Functions - 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. PmmC/Firmware Programming	25
6. Module Features	26
6.1. Display - 1.38" Round TFT Screen	26
6.2. DIABLO16 Processor	27
6.3. Audio	27
6.4. SD/SDHC Memory Cards	28
6.5. FAT16	28
7. Display Precautions	29
8. Hardware Tools	30
8.1. 4D Programming Cable/Adaptor	30
9. Programming Language	31
10. Workshop4 IDE	32
11. Mechanical Details	33

12. Schematic Diagram	34
13. Specifications	35

# 1. Description

The uLCD-220RD is a unique 1.38" Round LCD Intelligent Display Module powered by the 4D Systems' DIABLO16 Graphics Processor. Designed for applications such as gauges and instruments or special applications such as robotics.

The 1.38" Round LCD on the uLCD-220RD has a diameter of 220 pixels, however since the display is round there are only ever 110 pixels from the centre of the display, in any given direction. The pixels are in a round configuration instead of the normal square configuration found on other TFT LCD Displays.

Driving the display and peripherals is the DIABLO16 processor, a very capable and powerful chip that enables stand-alone functionality, programmed using the 4D Systems Workshop4 IDE Software. The Workshop4 IDE enables graphic solutions to be constructed rapidly and with ease due to its design being solely for 4D graphics processors.

The DIABLO16 Processor offers considerable FLASH and RAM upgrades over the PICASO processor, and also provides mappable functions such as I2C, SPI, Serial, PWM, Pulse Out, and Quadrature Input, to various GPIO, and also provide up to 4 Analog Input channels.

The display module has an array of features including PWM for Sound, micro-SD memory storage, general-purpose I/O including Analog Inputs, multiple TTL Serial, I2C and SPI channels and multiple millisecond resolution timers, amongst many more features.

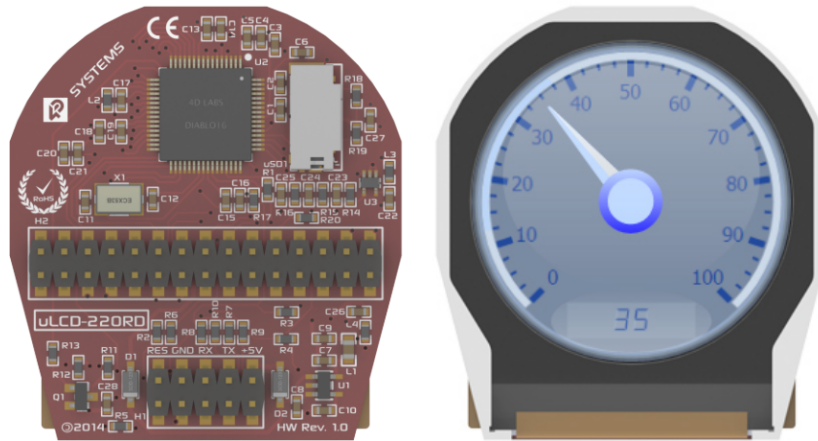
Anything that has been designed in the past to run on a PICASO Processor can theoretically run on this DIABLO16 Module with minor changes. Please ensure you contact the 4D Systems' support team if unsure if upgrading from a PICASO product and wanting to design with or change over to this uLCD-220RD Module.

The uLCD-220RD features a 30-way 2.54mm pitch male pin header, where all of the IO available on the Diablo16 processor have been broken out for the User. It also features a 10-way 2.54mm pitch male pin header which is common to a majority of 4D Systems' display modules, and is used for programming the module with our Workshop4 IDE, using a 4D Programming Cable or Adaptor.

## 2. Features


- Powerful 1.38" Intelligent LCD-TFT display module powered by DIABLO16.
- 220 x 220 Resolution (Round), RGB 65K true-to-life colors with LED Backlight. There is no touch interface on the uLCD-220RD, it is non-touch.
- Easy 5-pin interface to any TTL Serial Host: VCC, TX, RX, GND, RESET
- 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)
- On-board 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 filtered PWM Audio pin driven by WAV files from a micro-SD card, and an external amplifier enable pin allows easy connection to an external amplifier for audio playback.
- Display full colour images, animations, icons and video clips on chosen 4D Systems display.
- Supports all available Windows fonts.
- 4.0V to 5.5V range operation (single supply).
- Module dimensions: 42.4 x 46.9 x 14.1mm
- Weight ~ 15g.
- Display Viewing Area: Round 35.1mm Diam
- RoHS Compliant.
- CE Compliant.

### 3. Pin Configuration and Summary



#### H1 Pinout (Programming Header)

Pin	Symbol	I/O	Description
1	+5V IN	P	Main Voltage Supply +ve input pin. Reverse polarity protected. The range is 4.0V to 5.5V, nominal 5.0V. 500mA minimum required.
2	N/C	-	No Connect – Leave this pin unconnected
3	TX	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 micro-LCD. 5.0V Tolerant Pin. Connected to TX0 via Resistor.
4	N/C	-	No Connect – Leave this pin unconnected
5	RX	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 micro-LCD. 5.0V Tolerant Pin. Connected to RX0 via Resistor.
6	N/C	-	No Connect – Leave this pin unconnected
7	GND	P	Supply Ground
8	N/C	-	No Connect – Leave this pin unconnected
9	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.
10	-N/C	--	No Connect – Leave this pin unconnected

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

Pin	Symbol	I/O	Description
1	PA3	I/O/A	Main Voltage Supply +ve input pin. Reverse polarity protected. The range is 4.0V to 5.5V, nominal 5.0V. 500mA minimum required.
2	N/C	-	No Connect – Leave this pin unconnected
3	PA2	I/O/A	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 micro-LCD. 5.0V Tolerant Pin. Connected to TX0 via Resistor.
4	N/C	-	No Connect – Leave this pin unconnected
5	PA1	I/O/A	General Purpose I/O pin with Analog Capability. This pin is 5.0V tolerant when used as a Digital, with a range of 0-3.3V when used as an Analog Input
6	PA11	O	General Purpose I/O. This pin is 5.0V tolerant.
7	PA0	I/O/A	General Purpose I/O pin with Analog Capability. This pin is 5.0V tolerant when used as a Digital, with a range of 0-3.3V when used as an Analog Input
8	PA10	I/O	General Purpose I/O. This pin is 5.0V tolerant.
9	N/C	I	No Connect – Leave this pin unconnected
10	PA14	I	General Purpose Input Only. This pin is 5.0V tolerant.
11	GND	P	Supply Ground
12	PA15	I	General Purpose Input Only. This pin is 5.0V tolerant.
13	GND	P	Supply Ground
14	+5V IN	P	Main Voltage Supply +ve input pin. Reverse polarity protected. The range is 4.0V to 5.5V, nominal 5.0V. 500mA minimum required.
15	N/C	P	No Connect – Leave this pin unconnected
16	AUDIO	I/O	Audio Output for external amplifier, TTL Line level.
17	N/C	-	No Connect – Leave this pin unconnected
18	AUDENB	O	External Amplifier enable pin, to control an external amplifier from the module
19	PA9	I/O	General Purpose I/O. This pin is 5.0V tolerant.
20	RESET	I	Master Reset, Active Low (GND), See <b>H2 Pin 9</b> Description

Pin	Symbol	I/O	Description
21	PA8	I/O	General Purpose I/O. This pin is 5.0V tolerant.
22	3.3V_OUT	O	3.3V Output, limited to approximately 20mA, for external use.
23	PA7	I/O	General Purpose I/O. This pin is 5.0V tolerant.
24	RX0	I	Asynchronous serial port 0 receive pin. COM0
25	PA6	I/O	General Purpose I/O. This pin is 5.0V tolerant.
26	TX0	O	Asynchronous serial port 0 transmit pin. COM0
27	PA5	I/O	General Purpose I/O. This pin is 5.0V tolerant.
28	PA13	I/O	General Purpose I/O. This pin is 5.0V tolerant.
29	PA4	I/O	General Purpose I/O. This pin is 5.0V tolerant.
30	PA12	I/O	General Purpose I/O. This pin is 5.0V tolerant.

 **Note**

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



## 4. Hardware Interface - Pins

The uLCD-220RD Module provides both a hardware and software interface. This section describes in detail the hardware interface pins of the device.

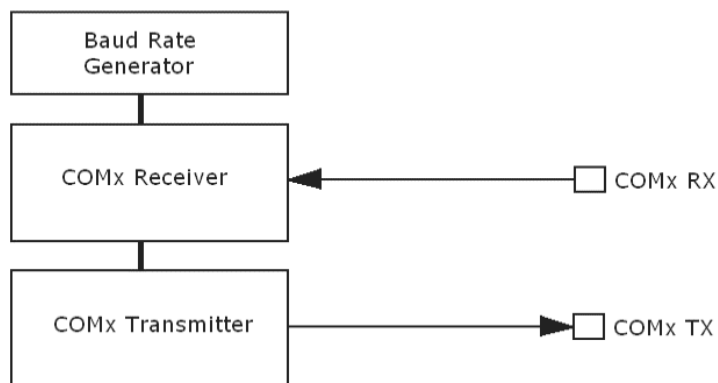
### 4.1. Serial Ports - TTL Level

The DIABLO16 Processor has three hardware asynchronous serial ports (COM1 - COM3) that can be configured on a variety of processor 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.

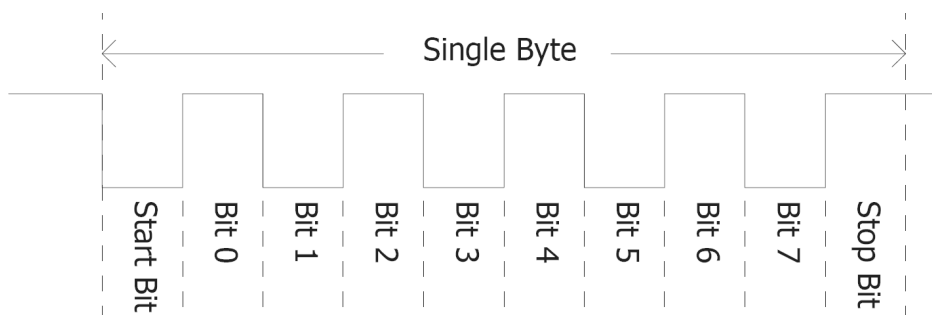
TX/RX0 is referred to as COM0 and is the only one used for programming the DIABLO16 itself.

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 4DGL user program downloads and chip configuration PmmC 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. Refer to the [PmmC/Firmware programming](#) section for more details.

**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 pin is 5.0V tolerant. This can be configured to 1 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 pin is 5.0V tolerant. This can be configured to 1 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 pin is 5.0V tolerant. This can be configured to 1 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 pin is 5.0V tolerant. This can be configured to 1 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 pin is 5.0V tolerant. This can be configured to 1 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 pin is 5.0V tolerant. This can be configured to 1 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					

Please refer to the [Diablo16 Internal Functions Manual](#) 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 General Purpose I/O					
	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		
PA15	Yes		Yes		

Please refer to the [Diablo16 Internal Functions Manual](#) 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. When set as Digital Inputs, the pins are 5V tolerant. Digital GPIO can source/sink 10mA. For more information see the [Specifications](#) section.


When you set pins as Analog Inputs, the pins have a **0 to 3.3V** range and a 12-bit resolution. For more information, see the [Analog Inputs](#) section.

**PA4-PA13:**

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.

**PA14-PA15:**

Input only pins. These pins can only be Digital Inputs.

 **Note**

All GPIO pins are 5.0V tolerant.

### 4.3. System Pins

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

##### **H2 Pin 14, H1 Pin 1:**

Module supply voltage input pins. At least one of these pins must 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 that backlight brightness will be lower for voltages under 5.0V.

#### **3.3V\_OUT (3.3V Output):**

##### **H2 Pin 22:**

3.3V Output for the user limited to approximately 20mA. Used for powering small external devices or components.

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

##### **H2 Pin 11/13, H1 Pin 7:**

Device ground pins. Any of these pins can be used to GND the module, at least one must be connected to the ground.

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

##### **H2 Pin 20, H1 Pin 9:**

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):**

##### **H2 Pin 18:**

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

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

##### **H2 Pin 16:**

Output specifically for Audio. This pin is designed to connect directly to an external amplifier circuit and is a filtered PWM signal from the DIABLO16 Processor.

## 4.4. Alternate Pin Functions - 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, I2C, Serial or a range of other functions.

### Note

Not all pins however can be configured to be any of the alternate pin functions.

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](#) 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.

DIABLO16 Alternate Pin Configurations SPI Communications					
	SPI1 SDO	SPI1 SDI	SPI1 SCK	SPI2 SDO	SPI2 SDI
<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.

DIABLO16 Alternate Pin Configurations I <sup>2</sup> C Communication					
	I <sup>2</sup> C1 SDA	I <sup>2</sup> C1 SCL	I <sup>2</sup> C2 SDA	I <sup>2</sup> C2 SCL	I <sup>2</sup> C3 SDA
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	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	Yes	Yes	Yes
PA11	Yes	Yes	Yes	Yes	Yes
PA12	Yes	Yes	Yes	Yes	Yes
PA13	Yes	Yes	Yes	Yes	Yes
PA14		SPECIAL		SPECIAL	
PA15	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](#) 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](#) 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](#) 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](#) 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 the 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](#) 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 Qencoder1 or Qencoder2 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](#) 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.





## 5. PmmC/Firmware Programming

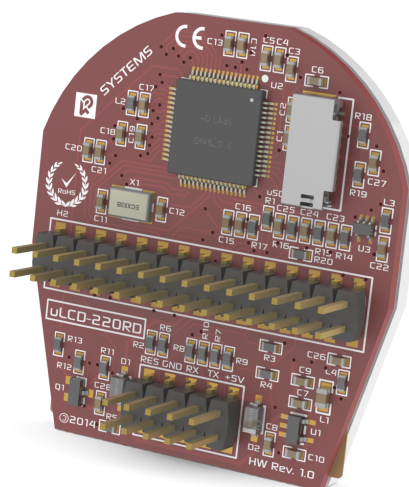
The DIABLO16 processor is a custom graphics processor. All functionality including the high-level commands is built into the chip. This chip level configuration is available as a PmmC (Personality-module-micro-Code) file, which can be likened to traditional Firmware. There is also a Display Driver file, 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 to program 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.

The PmmC file and Display Driver file can only be programmed into the device via the COM0 serial port with the aid of Workshop4, the 4D Systems IDE software.

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



## 6. Module Features

The uLCD-220RD module is designed to accommodate most applications. Some of the main features of the module are listed below.

### 6.1. Display - 1.38" Round TFT Screen

The uLCD-220RD module is equipped with a 1.38" Round TFT display. Details of the display are listed below:

- Screen Size: 1.38" diameter, 220x220 resolution, 65K colours, non-touch
- Screen Dimensions: 42.4 x 46.9 x 2.5mm
- Active Viewing Area: 35.1mm Diameter circle
- Pixel Pitch: 0.156(H) x 0.156(V)mm
- Brightness: 171cd/m<sup>2</sup>
- Contrast Ratio: 355:1
- Viewing Angle Above Centre: 61 degrees
- Viewing Angle Below Centre: 64 degrees
- Viewing Angle Left of Centre: 71 degrees
- Viewing Angle Right of Centre: 73 degrees
- Viewing Direction: 6 O'clock
- 2 Parallel LEDs for Backlighting

#### Note

The Displays used are the highest-rated 'Grade A' Displays, which allow for 0-4 defective pixels. A defective pixel could be solid Black (Dead), White, Red, Green or Blue.

## 6.2. 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-LCD 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.

The data sheet for the processor is available from the <http://www.4dsystems.com.au> website:

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

## 6.3. Audio

Audio playback support in the DIABLO16 Processor enables the uLCD-220RD module to play audio WAV files stored in the micro-SD memory card. An external amplifier is required, as the output of the uLCD-220RD is a filtered PWM only, with a separate amplifier enabling signal. 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, 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](#)

## 6.4. SD/SDHC Memory Cards



The module supports micro-SD memory cards via the onboard 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 and high-capacity HC memory cards (4GB and above).

Any cards larger than 4GB need to be partitioned so the first partition is FAT16. See the RMPET tool in the Workshop4 IDE.

### 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.

## 6.5. FAT16

The uLCD-220RD Module uses 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 with the FAT16 option. The formatting of the card can be done on any PC system with a card reader. Select the appropriate drive and choose the FAT16 (or just FAT in some systems) option when formatting.

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.

The FAT partition is always first (if it exists) and can be up to the maximum size permitted by FAT16. Windows 7 will format FAT16 up to 4GB. Windows XP will format FAT16 up to 2GB and the Windows XP command prompt will format FAT16 up to 4GB.

## 7. Display Precautions

- Avoid having to display the same image/object on the screen for lengthy periods. This will 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 will cause the electrodes to corrode. Wipe off any moisture gently or let the display dry before usage.
- 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.

## 8. Hardware Tools

The following hardware tools are required for full control of the uLCD-220RD module.

### 8.1. 4D Programming Cable/Adaptor

The 4D Programming Cable and uUSB-PA5 Programming Adaptor are essential hardware tools to program, customise and test the DIABLO16 Processor.

Either the 4D Programming Cable or the uUSB-PA5 Programming Adaptor can be used.

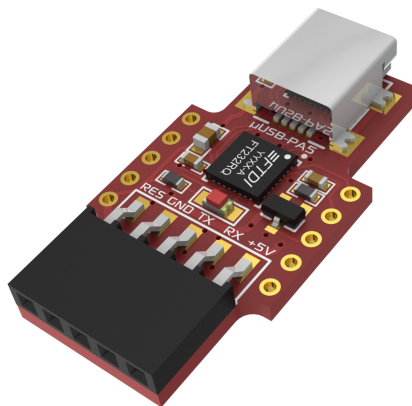
The 4D programming interfaces are used to program a new Firmware/PmmC, Display Driver and for downloading compiled 4DGL code into the processor. They even serve as an interface for communicating serial data to the PC.

The 4D Programming Cable and uUSB-PA5 Programming Adaptor are available from the [4D Systems website](#).

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



*4D Programming Cable*



*uUSB-PA5 Programming Adaptor*

---

## 9. Programming Language

The uLCD-220RD Module uses the DIABLO16 processor, which belongs to a family of processors powered by a highly optimised softcore virtual engine, EVE (Extensible Virtual Engine).

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 about 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.

## 10. 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 Down-Loader 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.

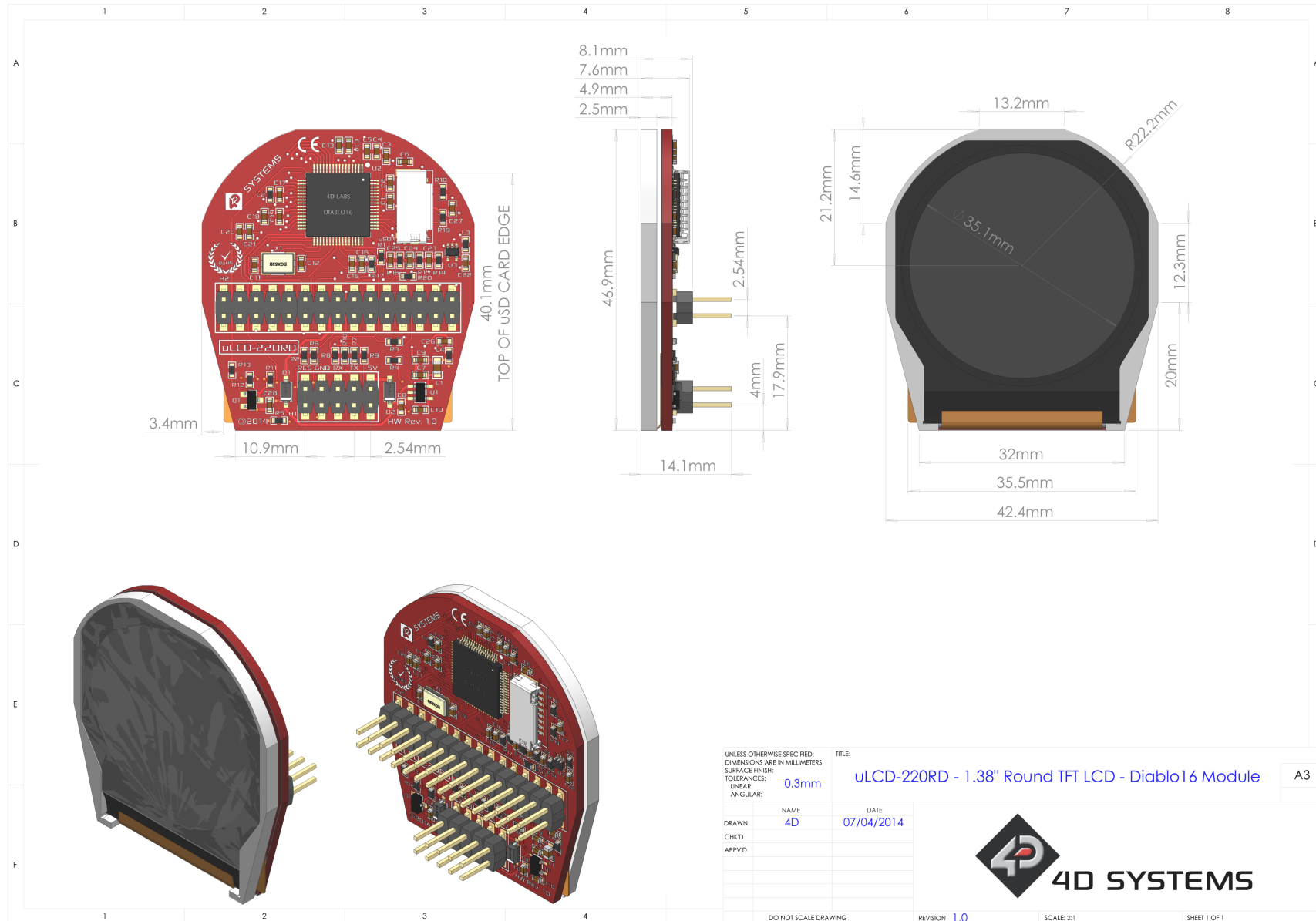
- The **Designer** environment enables the user to write 4DGL code in its natural form to program the range of 4D System's intelligent displays.
- A visual programming experience, suitably called **Visi**, enables drag-and-drop type placement of objects to assist with 4DGL code generation and allows 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](#).

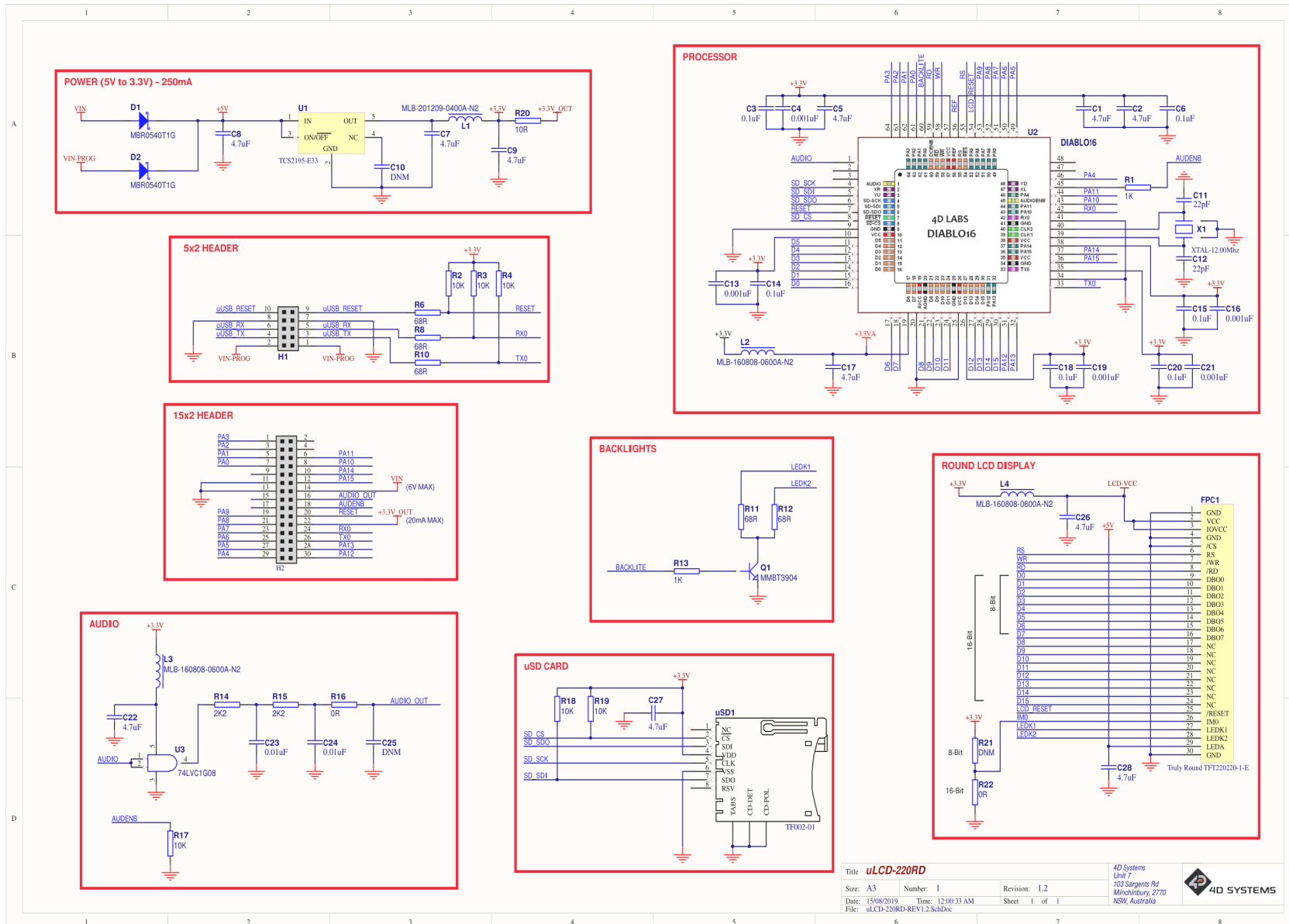
The Workshop4 IDE is available from the [4D Systems website](#).



# 11. Mechanical Details



# 12. Schematic Diagram



Title <b>uLCD-220RD</b>		4D Systems Unit 7 103 Sargeants Rd Ministry, 2770 NSW, Australia	
Size: A3	Number: 1	Revision: 1.2	
Date: 15/08/2019	Time: 12:00:33 AM	Sheet 1 of 1	
File: uLCD-220RD-REV1.2.SchDoc			

## 13. 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)	All GPIO pins, 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