# uLCD-480RD

# Datasheet

Revision 1.0

Copyright © 2023 4D Systems

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

# Contents

# 1. Description

The uLCD-480RD is a unique 2.1" Round LCD Intelligent Display Module powered by the 4D Systems' DIABLO-16 Graphics Processor.

The 2.1" TFT IPS Round LCD on the uLCD-480RD has a diameter of 480 pixels, however since the display is round there are only ever 240 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 DIABLO-16 processor, a very capable and powerful chip that enables stand-alone functionality, programmed using the 4D Systems Workshop4 IDE Software. The WorkshopIDE enables graphic solutions to be constructed rapidly and with ease due to its design being solely for 4D graphics processors.

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

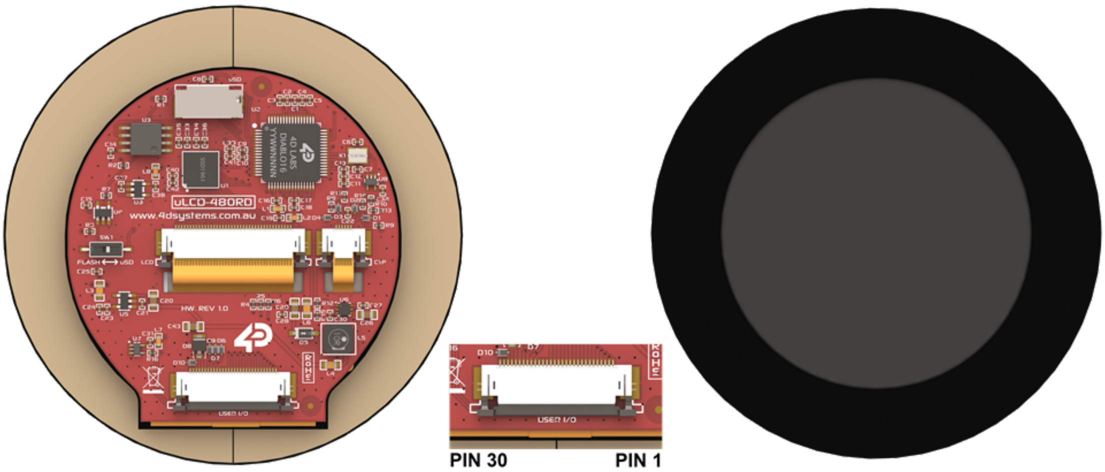
The display module has an array of features including PWM for Sound, selectable micro-SD flash storage or SPI Flash storage, general-purpose I/O including Analog Inputs, multiple TTL Serial, I2C and SPI channels and multiple millisecond resolution timers, amongst many more features.

The uLCD-480RD features a 30 pin ZIF/FFC/FPC 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. All of the IO available on the DIABLO-16 processor have been broken out for the User. It is compatible with pinout found on the gen4-uLCD range of products, and is programmed using the 4D-UPA Programmer - the same as the gen4-uLCD line.

## 2. Features

- Powerful 2.1" Intelligent LCD-TFT IPS display module powered by DIABLO-16.
- 480 x 480 Resolution (Round), RGB 65K true-to-life colors with LED Backlight.
- Available in Non-Touch with CLB, or Capacitive Touch with CLB.
- 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)
- 30pin FPC connection, for all signals, power, communications, GPIO and programming.
- On-board micro-SD memory card connector for multimedia storage and data logging purposes.
- On-board SPI Flash memory, selectable to use instead of the on-board micro-SD memory card.
- 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 6.0V range operation (single supply).
- Module dimensions: 80.0 x 80.0 x 8.5mm
- Weight ~ 32g.
- Display Viewing Area: Round 53.28mm Diam
- RoHS & REACH Compliant.
- CE Compliance pending.

# 3. Pin Configuration and Summary



| 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. |
| 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. |
| 18 | NC | - | Not Connected |
| 19 | AUDIO_OUT | O | Audio Output, PWM, to feed into external amplifier via filter network |
| 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 |

| Pin | Symbol | I/O | Description |
|---|---|---|---|
| | | | 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 module. 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 module. This pin has a 3.3V Level output. |
| 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 | 5V IN | P | Main Voltage Supply +ve input pin. Reverse polarity protected. Range is 4.0V to 5.5V, nominal 5.0V. |
| 29 | NC | - | Not Connected |
| 30 | GND | P | Supply Ground |

**Note**

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

# 4. Hardware Interface - Pins

The uLCD-480RD Module provides both a hardware and software interface. This section describes in detail the hardware interface pins of the device.

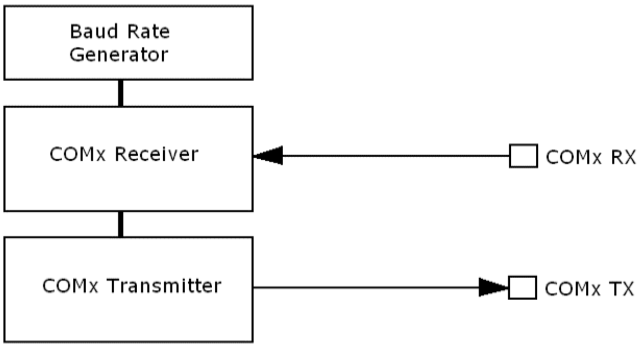
## 4.1. Serial Ports - TTL Level

The DIABLO-16 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 DIABLO-16's serial ports can be used to communicate with external serial devices.

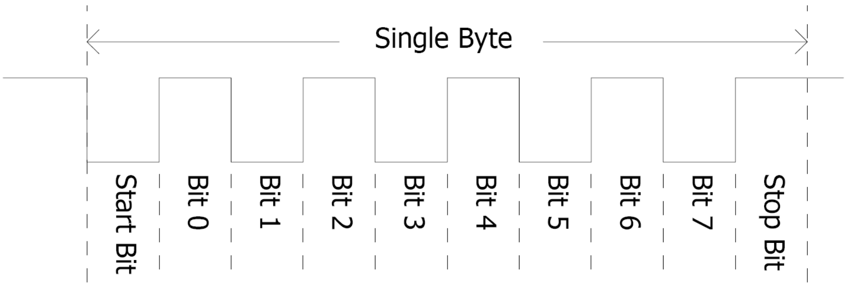
TX/RX0 is referred to as COM0 and is the only one used for programming the DIABLO-16 itself.

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.

**DIABLO-16 Serial TTL Comm Port Configuration Options**

| | TX1 | RX1 | TX2 | RX2 | TX3 | RX3 |
|---|---|---|---|---|---|---|
| **PA0** | | Yes | | Yes | | Yes |
| **PA1** | Yes | Yes | Yes | Yes | Yes | Yes |
| **PA2** | | Yes | | Yes | | Yes |
| **PA3** | Yes | Yes | Yes | Yes | Yes | Yes |
| **PA4** | Yes | Yes | Yes | Yes | Yes | Yes |
| **PA5** | Yes | Yes | Yes | Yes | Yes | Yes |
| **PA6** | Yes | Yes | Yes | Yes | Yes | Yes |
| **PA7** | Yes | Yes | Yes | Yes | Yes | Yes |
| **PA8** | Yes | Yes | Yes | Yes | Yes | Yes |
| **PA9** | Yes | Yes | Yes | Yes | Yes | Yes |
| **PA10** | | Yes | | Yes | | Yes |
| **PA11** | | Yes | | Yes | | Yes |
| **PA12** | Yes | Yes | Yes | Yes | Yes | Yes |
| **PA13** | Yes | 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 DIABLO-16 Internal Functions Manual for information on how to set the DIABLO-16 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.

**DIABLO-16 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 | Yes | | |
| **PA15** | Yes | Yes | Yes | | |

Please refer to the DIABLO-16 Internal Functions Manual for information on how to set the DIABLO-16 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:**

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. At least 1 of these pins should be connected to a stable supply voltage in the range of 4.0 Volts to 5.5 Volts DC, the more the better. 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, the more the better.

**RESET (Module Master Reset):**

Module Master Reset pin. An active low pulse of greater than 2µs 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 DIABLO-16 Processor. This pin is a 3.3V level PWM output to drive an external amplifier with DIABLO-16 generated Audio, via an external filtering circuit to turn digital PWM into Analog. Note that AUDENB must be enabled in DIABLO-16 for any input signal to be heard.

## 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, $I^2C$, 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.

**DIABLO-16 Alternate Pin Configurations I/O Support Functions**

| | Pulse Out | PWM Out | Pin Counter | Quadrature In |
|---|---|---|---|---|
| **PA0** | Yes | | | Yes |
| **PA1** | Yes | | | Yes |
| **PA2** | Yes | | | Yes |
| **PA3** | 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 |
| **PA11** | | | | Yes |
| **PA12** | | | | Yes |
| **PA13** | | | | Yes |
| **PA14** | | | | |
| **PA15** | | | | |

**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 DIABLO-16 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.

**DIABLO-16 Alternate Pin Configurations SPI Communications**

| | SPI1 SDO | SPI1 SDI | SPI1 SCK | SPI2 SDO | SPI2 SDI | SPI2 SC |
|---|---|---|---|---|---|---|
| **PA0** | | Yes | | | Yes | |
| **PA1** | Yes | Yes | Yes | Yes | Yes | Yes |
| **PA2** | | Yes | | | Yes | |
| **PA3** | Yes | 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** | | | | | | |

The table below illustrates the GPIO pins you can use for the three different $I^2$C channels available.

**DIABLO-16 Alternate Pin Configurations I2C Communication**

| | $I^2$C1 SDA | $I^2$C1 SCL | $I^2$C2 SDA | $I^2$C2 SCL | $I^2$C3 SDA | $I^2$C3 SC |
|---|---|---|---|---|---|---|
| **PA0** | Yes | Yes | Yes | Yes | Yes | Yes |
| **PA1** | Yes | Yes | Yes | Yes | Yes | Yes |
| **PA2** | Yes | Yes | Yes | Yes | Yes | Yes |
| **PA3** | Yes | Yes | Yes | Yes | Yes | Yes |
| **PA4** | Yes | Yes | Yes | Yes | Yes | Yes |
| **PA5** | Yes | Yes | Yes | Yes | Yes | Yes |
| **PA6** | Yes | Yes | Yes | Yes | Yes | Yes |
| **PA7** | Yes | Yes | Yes | Yes | Yes | Yes |
| **PA8** | Yes | Yes | Yes | Yes | Yes | Yes |
| **PA9** | Yes | Yes | Yes | Yes | Yes | Yes |
| **PA10** | Yes | Yes | Yes | Yes | Yes | Yes |
| **PA11** | Yes | Yes | Yes | Yes | Yes | Yes |
| **PA12** | Yes | Yes | Yes | Yes | Yes | Yes |
| **PA13** | Yes | Yes | Yes | Yes | Yes | Yes |
| **PA14** | | SPECIAL | | SPECIAL | | SPECIAL |
| **PA15** | SPECIAL | | SPECIAL | | SPECIAL | |

Please see I2C section for pins marked SPECIAL.

## 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 DIABLO-16 Internal Functions Manual for more information on how to use the SPI functions, along with the DIABLO-16-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 $I^2C$ 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 $I^2C$.

To map an $I^2C$ 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 $I^2C$ 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 DIABLO-16 Internal Functions Manual for more information on how to use the $I^2C$ functions, along with the DIABLO-16-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 DIABLO-16 Internal Functions Manual for more information on how to use the Pulse Out functions, along with the DIABLO-16-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 DIABLO-16 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 DIABLO-16 Internal Functions Manual for more information on how to use the PWM functions, along with the DIABLO-16-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, OVFfunction);**

*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

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

Please refer to the DIABLO-16 Internal Functions Manual for more information on how to use the Pin Counter functions, along with the DIABLO-16-Processor datasheet.

## 4.10. Quadrature In

There are two Quadrature Input channels available on the DIABLO-16 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 DIABLO-16 Internal Functions Manual for more information on how to use the Quadrature Input functions, along with the DIABLO-16-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 DIABLO-16 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);**

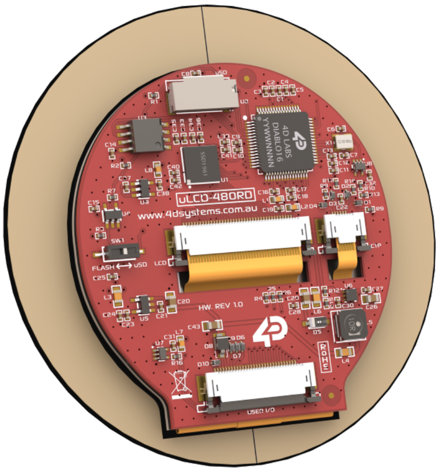
*where:*

'rate' is the number of samples per second,<br>
'samples' is the number of samples to collect per channel,<br>
'1buf' to '4buf' are the buffer addresses for the 4 channels, and<br>
'func' is the user function to call when the number of samples specified has been collected.

Please refer to the DIABLO-16 Internal Functions Manual for more information on how to use the Analog Input functions, along with the DIABLO-16-Processor datasheet.

# 5. Module Features

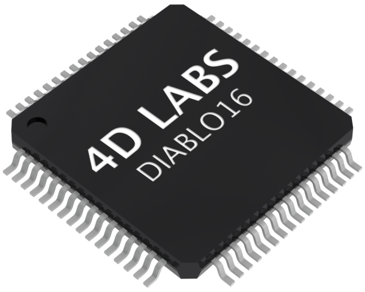
The uLCD-480RD module is designed to accommodate most applications.



Some of the main features of the module are listed below.

## 5.1. DIABLO-16 Processor

The module is designed around the DIABLO-16 Graphics Controller from 4D-Labs.



The DIABLO-16 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 DIABLO-16 graphics processor datasheet for more information.

## 5.2. Audio

Audio playback support in the DIABLO-16 Processor enables the uLCD-480RD module to play audio WAV files stored in the micro-SD memory card. An external amplifier is required, as the output of the uLCD-480RD 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.

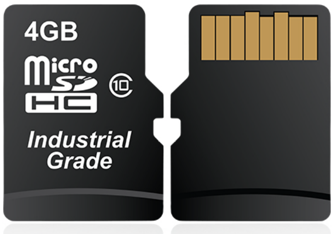
For a complete list of audio commands please refer to the separate document titled:

- DIABLO-16 4DGL Internal Functions

## 5.3. SD/SDHC Memory Cards

The DIABLO-16 processor uses off-the-shelf standard SDHC/SD/microSD memory cards with up to 2GB capacity usable with FAT16 formatting. For any FAT file-related operations, before the memory card can be used it must first be formatted to FAT16. 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 card is now ready to be used in the DIABLO-16-based application.

The DIABLO-16 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.

RMPET, a 4D Systems Tool found in the Workshop4 IDE, is capable of repartitioning and formatting microSD cards to be the appropriate type and format for 4D Systems processors. This tool should be used for all cards.

**Note**

1. An SPI Compatible SDHC/SD/microSD card MUST be used. DIABLO-16 along with other 4D Systems Processors requires SPI mode to communicate with the SD card. If a non-SPI compatible SD card is used, then the processor will not be able to mount the card.
2. Read disturbance is a well-known issue with flash memory devices, such as microSD cards, where reading data from a flash cell can cause the nearby cells in the same memory block to change over time. This issue can be prevented by using industrial-grade microSD cards with read disturb protection. Industrial-grade microSD cards have firmware that actively monitors the read operation and refreshes areas of memory that have high traffic and even move data around to prevent read disturb error from occurring. Furthermore, manufacturers may choose to implement read disturb protection on a specific part of the flash memory only, such that the beginning part of the memory might not be protected. The RMPET utility in Workshop4 is designed to create the first partition at an offset from the start of the microSD card to account for this situation. It is therefore recommended to always partition and format an industrial microSD card using the RMPET utility before using it with 4D Systems processors. Many commercial grade cards designed for Cameras etc, do not handle read disturb well at all, and therefore it is always recommended to use an Industrial grade microSD card with 4D modules. 4D offers one that is tried and tested, on our website.

## 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 DIABLO-16 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.

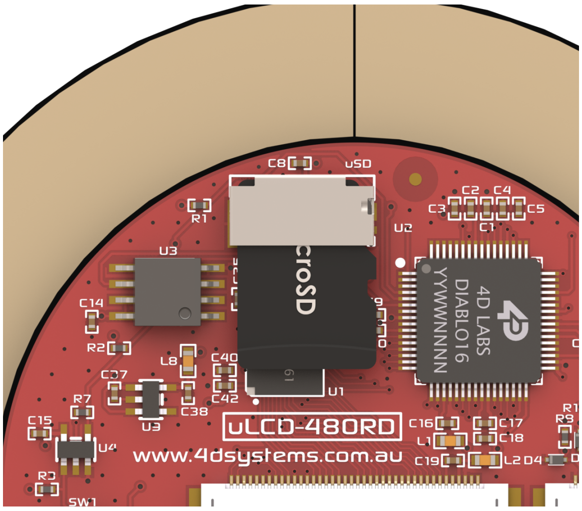
A Max of 4GB can be used by the FAT16 file system. The FAT partition is always first (if it exists). Any space larger than 4GB will be RAW, and can still be used by your 4D Systems module, using different functions. Please refer to the Application Notes.

## 5.5. Micro-SD Socket

The micro-SD socket used on these modules is a small push-pull type.

A micro-SD card is inserted on the component side of the socket, where the card is sitting over top of the components. Press the card gently into the socket, and pull the card out gently to remove it.

Please note that when using the on-board SPI Flash, a card cannot be inserted into the micro-SD socket, and the selector switch (SW1) must be selected to the appropriate media, either microSD or Flash. More information in the next SPI Serial Flash Memory section.

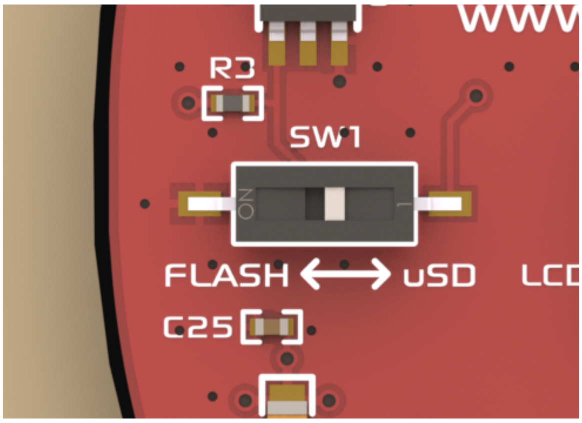


## 5.6. SPI Serial Flash Memory

On the uLCD-480RD there is 32MB of SPI Serial Flash memory mounted on the board, which can be used instead of the micro-SD card if desired.

The SPI Serial Flash Memory and the micro-SD card are mutually exclusive, meaning only one or the other can be used. Both cannot operate together, so a choice is needed at the time of development.

At the left of the modules PCB is a micro-switch (SW1), which can be changed between Flash and uSD selections, to enable the desired storage medium for the project/product in question. This must be done at design time, as the application loaded on to both the Diabl16's internal Flash, and the storage (uSD card or SPI Flash Memory) work together, so this cannot be changed once a project has been loaded, as the application will not work.



Inside the Workshop4 IDE, the selection for Flash or FAT (micro-SD) can be made.

SPI Flash Memory can be used in applications where a micro-SD card is either not permitted, or not desirable. The memory capacity is considerably smaller than micro-SD, so there is also a limitation on what can be stored. Smart project planning of widget types is required. It may be desirable to use a majority of Internal or Inherent widgets when using SPI Flash Memory and leave the SPI Flash Memory itself for GCI widgets / Images etc which cannot be stored any other way. Please refer to our Workshop4 Widgets Reference for more information on the available widget types.

# 6. 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 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, damage will certainly occur. 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 temperature 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 broken backlight
- Display modules have a finite life, which is typically dictated by the display itself, more specifically the backlight. The backlight contains LED's, which fade over time. In the Specifications section is a figure for the typical life of the display, and the criteria are listed.

# 7. Hardware Tools

The following hardware tools are required for full control of the uLCD-480RD module.

## 7.1. 4D Programming Cable/Adaptor

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.

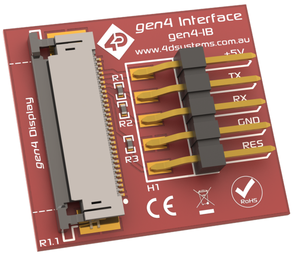
The 4D Programming Cable is a USB to Serial-TTL UART converter cable incorporating the Silabs CP2102 USB to Serial UART bridge IC.

*4D Programming Cable*

**Note**

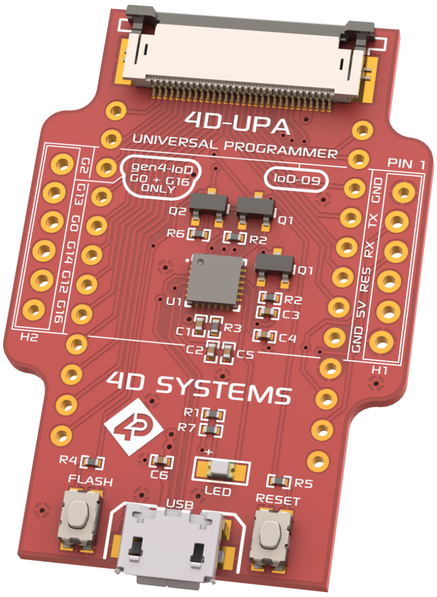
In conjunction with a 4D Programming Cable or a 4D Programming Adaptor, the uLCD-480RD range of display modules require a simple interface board (gen4-IB), as there is no 5-way male header on gen4 modules like other 4D Systems modules.

gen4 Interface boards (gen4-IB) ship with each uLCD-480RD Display Module unless otherwise stated. Simply connect the supplied 30-way FPC cable to the ZIF connector on the gen4 Interface Board, and connect the Programming Cable or Adaptor to the 5-way header on the Interface Board. You are now ready to program using a 4D Programming Cable. This can also be used to interface with other devices such as a 4D Arduino Adaptor Shield, breadboard etc.



*gen4-IB*

An alternative to using a gen4-IB and a 4D Programming Cable is a single all-in-one board called the 4D-UPA (4D Universal Programming Adaptor).



*4D-UPA*

The 4D-UPA minimizes the connections and modules required for programming - creating a single module with a micro USB interface, and DIP style pads for GPIO breakout of all the signals used on the gen4 Display interface, which is useful for development or final product use.

The GPIO naming convention on the 4D-UPA does not reflect the GPIO naming of the actual display module, due to the 4D-UPA being universal and able to be used with many 4D Products. Please review the 4D-UPA datasheet for information on mapping the GPIO naming from this module, with the GPIO naming on the 4D-UPA, to ensure you connect to the correct pins you desire.

**Note**

In addition to these modules, the gen4-PA and uUSB-PA5/uUSB PA5-II can still be used. Please be reminded though that the uUSB-PA5/PA5-II still needs a gen4-IB to be used for programming.

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

# 8. Programming Language

The DIABLO-16 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 about the 4DGL language, please refer to the following documents:

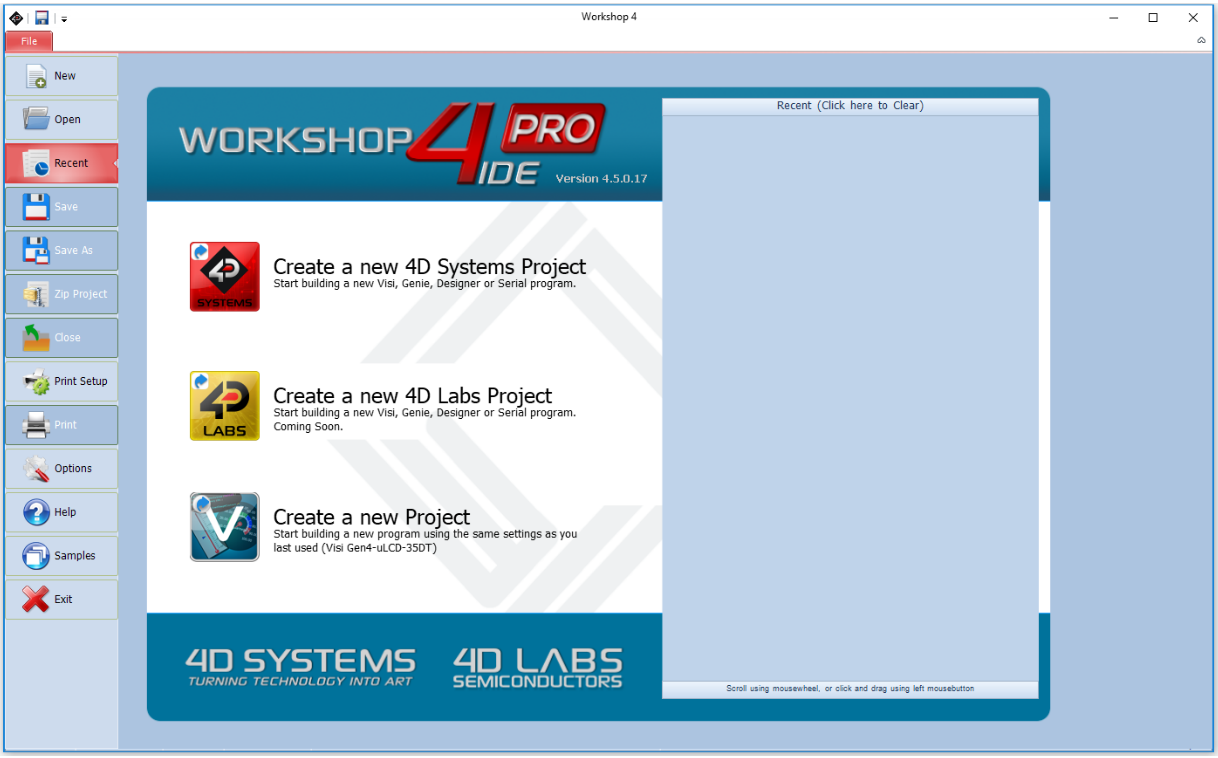
- 4DGL Programmers Reference Manual
- DIABLO-16 Internal Functions Manual

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.



Create a new 4D Systems Project
Start building a new Visi, Genie, Designer or Serial program.

Create a new 4D Labs Project
Start building a new Visi, Genie, Designer or Serial program.
Coming Soon.

Create a new Project
Start building a new program using the same settings as you last used (Visi Gen4-uLCD-35DT)

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.

## 9.1. PmmC/Firmware Programming

The DIABLO-16 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 DIABLO-16 processor, along with product-specific settings and parameters which are required over and above what is found in the PmmC.

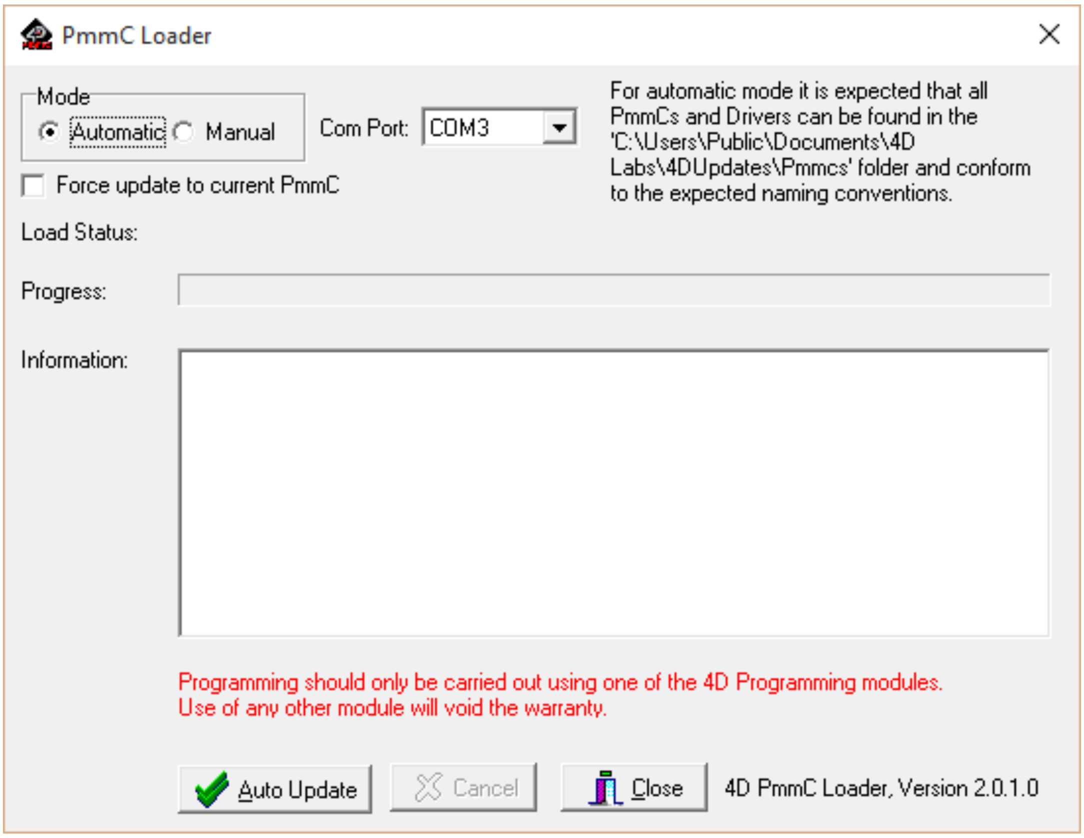
The PmmC file and Display Driver file can only be programmed into the device via the COM0 serial port with the aid of Workshop4, the 4D Systems IDE software.

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

Mode: ● Automatic ○ Manual

Com Port: COM3

For automatic mode it is expected that all PmmCs and Drivers can be found in the 'C:\Users\Public\Documents\4D Labs\4DUpdates\Pmmcs' folder and conform to the expected naming conventions.

□ Force update to current PmmC

Load Status:

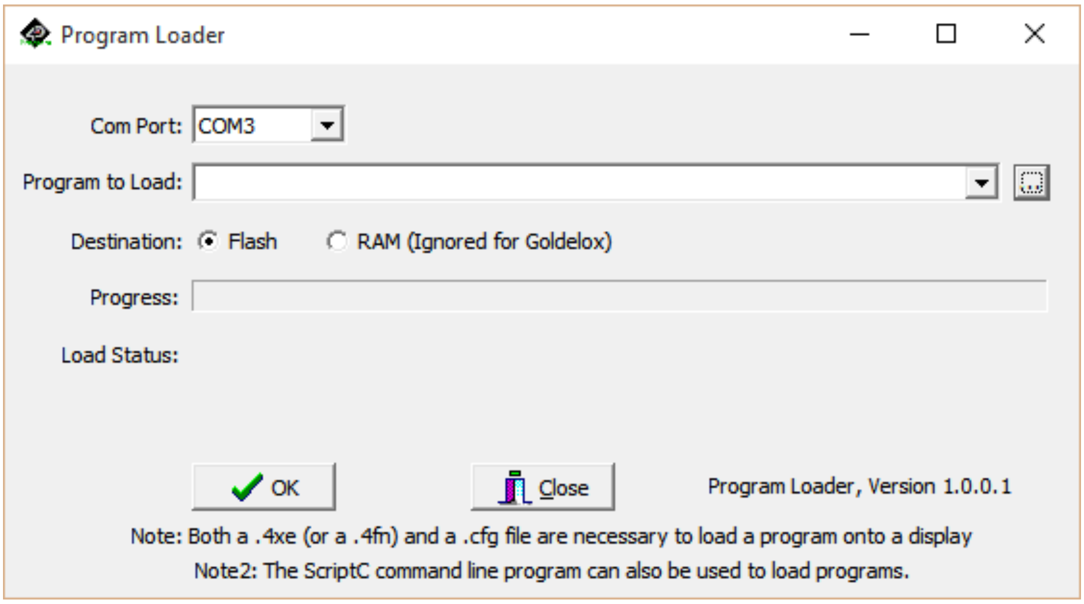
Progress:

Information:

Programming should only be carried out using one of the 4D Programming modules.
Use of any other module will void the warranty.

4D PmmC Loader, Version 2.0.1.0

**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.



Program Loader

Com Port: COM3

Program to Load:

Destination: ● Flash ○ RAM (Ignored for Goldelox)

Progress:

Load Status:

Program Loader, Version 1.0.0.1

Note: Both a .4xe (or a .4fn) and a .cfg file are necessary to load a program onto a display
Note2: The ScriptC command line program can also be used to load programs.

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**.

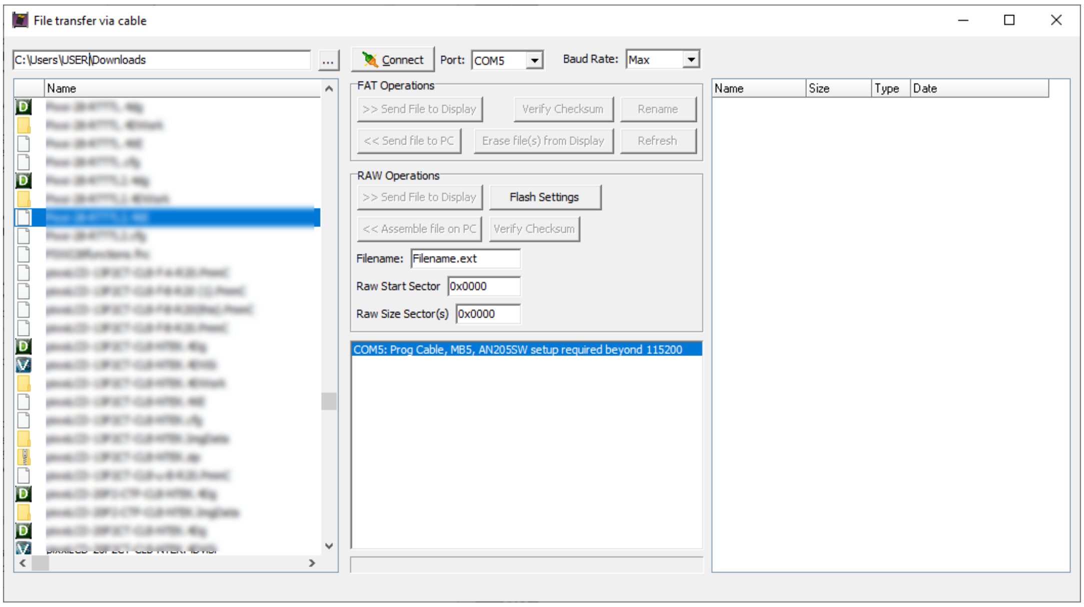
## 9.2. File Transfer

The DIABLO-16 processor is capable of transferring files over the serial UART from the PC. This is used when the SPI Serial Flash Memory is loaded by Workshop4, but it can also be used to load the micro-SD card if desired.

The process of loading a micro-SD card can take some time, depending on the size of the files needing to be transferred, and it is often faster to remove the micro-SD card from the display module, place it into your PC using a Media/Card reader, and copying the files directly. However in cases where this is not possible, transferring over the serial UART is available.

Additional to Workshop4 natively using File Transfer during its programming phase, File Transfer can also be run manually by opening the File Transfer application from the Windows Start Menu, or from the /DEP folder inside the Workshop4 installation folder.

When loading content on to the SPI Serial Flash Memory manually, you must erase the memory before writing new files.



File transfer via cable

C:\Users\USER\Downloads

Port: COM5

Baud Rate: Max

Filename: Filename.ext

Raw Start Sector 0x0000

Raw Size Sector(s) 0x0000

COM5: Prog Cable, MB5, AN205SW setup required beyond 115200

File Transfer can be used for loading SPI Serial Flash Memory or the micro-SD card. When actioning the programming process from Workshop4, the appropriate one will be loaded based on your hardware and software settings.

If using the 4D Programming Cable, some software modifications to the USB chip inside the cable are required in order to transfer at full speed. Please refer to this **post** on the 4D Systems Forum, for details on the change. When using the 4D-UPA, uUSB-PA5-II or gen4-PA, this is not required, it only relates to the **4D Programming Cable**.

## 10. 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 Woskshop4 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-480RD (-CTP-CLB or -CLB) Display Module
- gen4 Interface Module (gen4-IB)
- Universal Programming Adaptor (4D-UPA)
- 4GB Industrial 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 gen4 display to gen4-IB or 4D-UPA

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.

# 11. Display Module Part Numbers

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

**Example:**

- uLCD-480RD-CTP-CLB
- uLCD-480RD-CLB

*where:*

uLCD - microLCD Display Family

480 - Display resolution (480x480)

R - Round Display

D - DIABLO-16 Processor

CTP - Capacitive Touch

CLB - Cover Lens Bezel

**Note**

- A product without a CTP in the part number is a non-touch variant.
- Cover Lens Bezels (CLB) are glass fronts for the display module with overhanging edges, which allow the display module to be mounted directly into a panel using special adhesive on the overhanging glass. Available for both non-touch and capacitive touch.

# 12. Cover Lens Bezel - Tape Spec

The perimeter of the CLB display modules features double-sided adhesive tape, designed to stick directly onto a panel, enclosure, box etc without the need for any mounting screws or hardware.

The tape used is 3M 9495LE tape, which uses well-known and strong 3M 300LSE adhesives. The double-sided adhesive has a thickness of 0.17mm once the backing has been removed.

More information on this adhesive can be found on the 3M website.

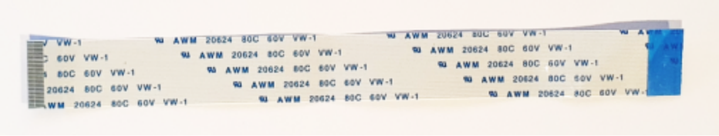
## 13. FFC Cable information

The FFC cables supplied by 4D Systems (included with products) have the following specifications:

- **30 Pin** Flexible Flat Cable, 150mm Long, 0.5mm (0.02") pitch
- Cable Type: AWM 20624 80C 60V VW-1
- Heat Resistance 80 Degrees Celsius
- Connections on the opposite side at each end (Type B)

**Note**

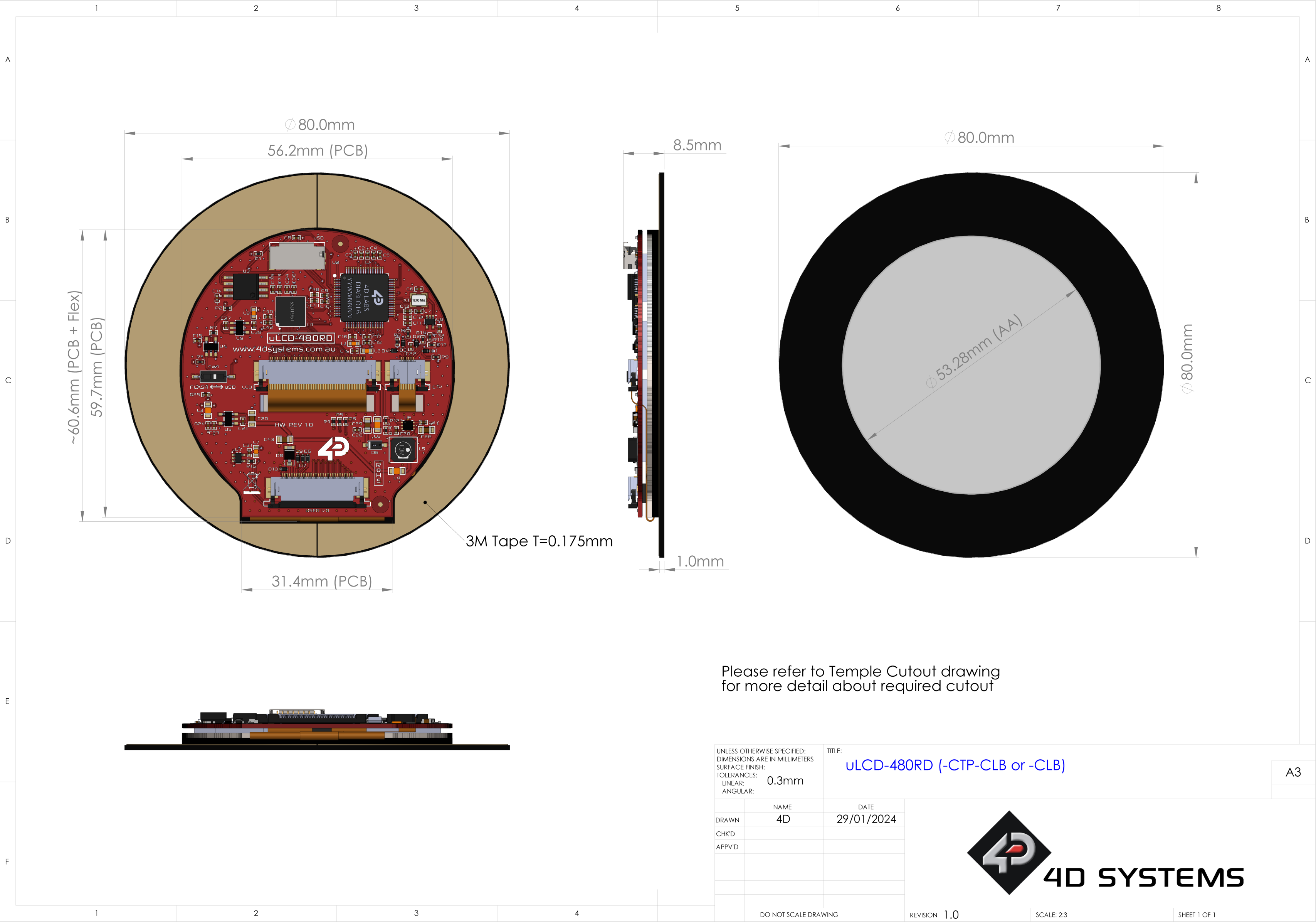
You can get different cable lengths by contacting the 4D Systems sales team.



If you are interfacing with this module directly to your product, and wish to interface via the FFC cable directly, suitable connectors are readily available from many electronics suppliers, such as Digikey, Mouser, Farnell, RS, etc.

A standard 30-pin, 0.5mm pitch, 0.3mm thick FFC, FFC connector. They are available in Top Contact and Bottom Contact, so depending how you orientate the cable on your product, will determine which one you need. Please however take care of the pinout and how it flows from the display module, through the FFC and into your product, to ensure Pin1 and Pin30 are where you expect them to be.

# 14. Mechanical Details



Please refer to Temple Cutout drawing for more detail about required cutout

UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR: 0.3mm
ANGULAR:

TITLE: uLCD-480RD (-CTP-CLB or -CLB)

A3

|  | NAME | DATE |
|---|---|---|
| DRAWN | 4D | 29/01/2024 |
| CHK'D |  |  |
| APPV'D |  |  |

4D SYSTEMS

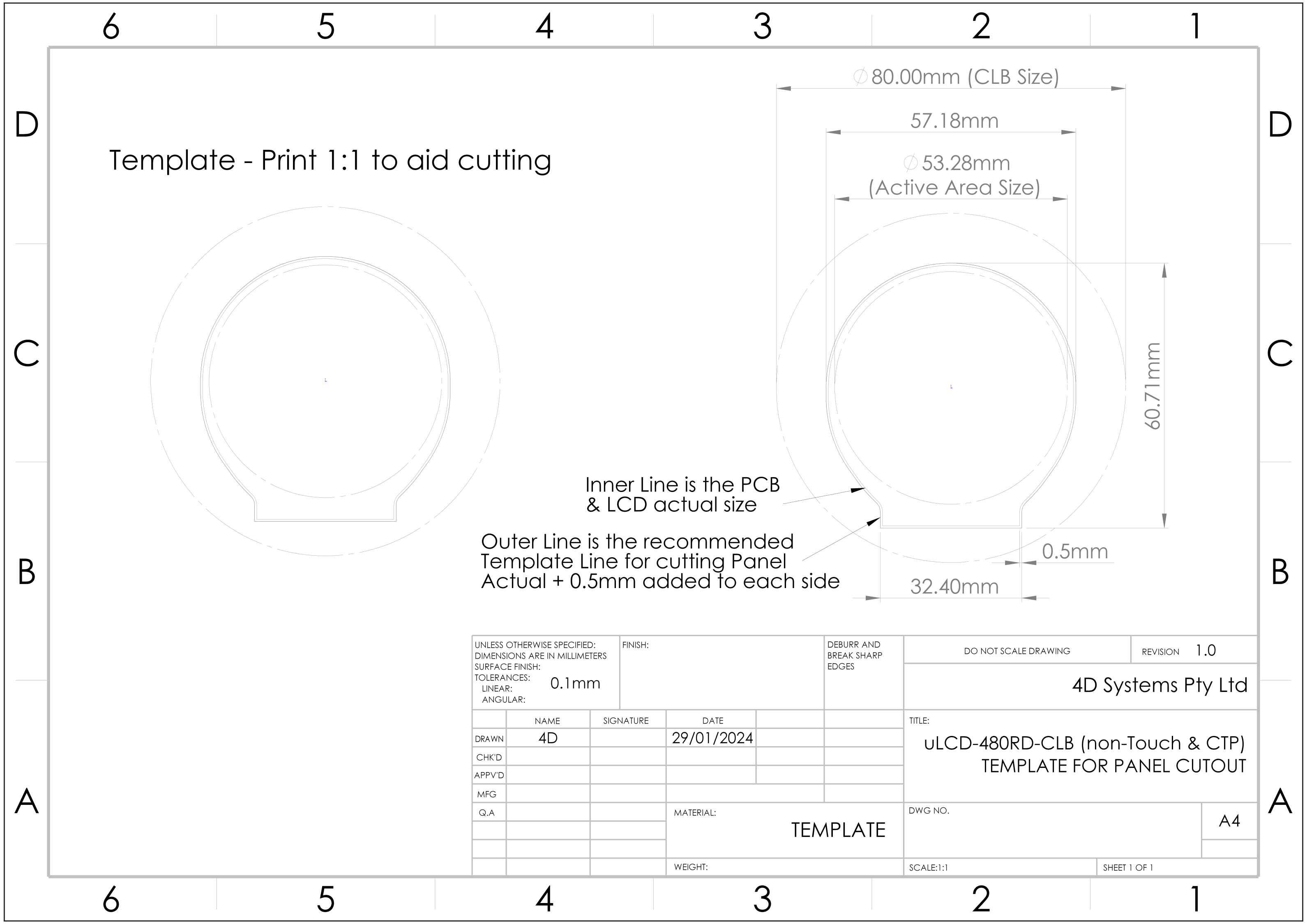
DO NOT SCALE DRAWING

REVISION 1.0

SCALE: 2:3

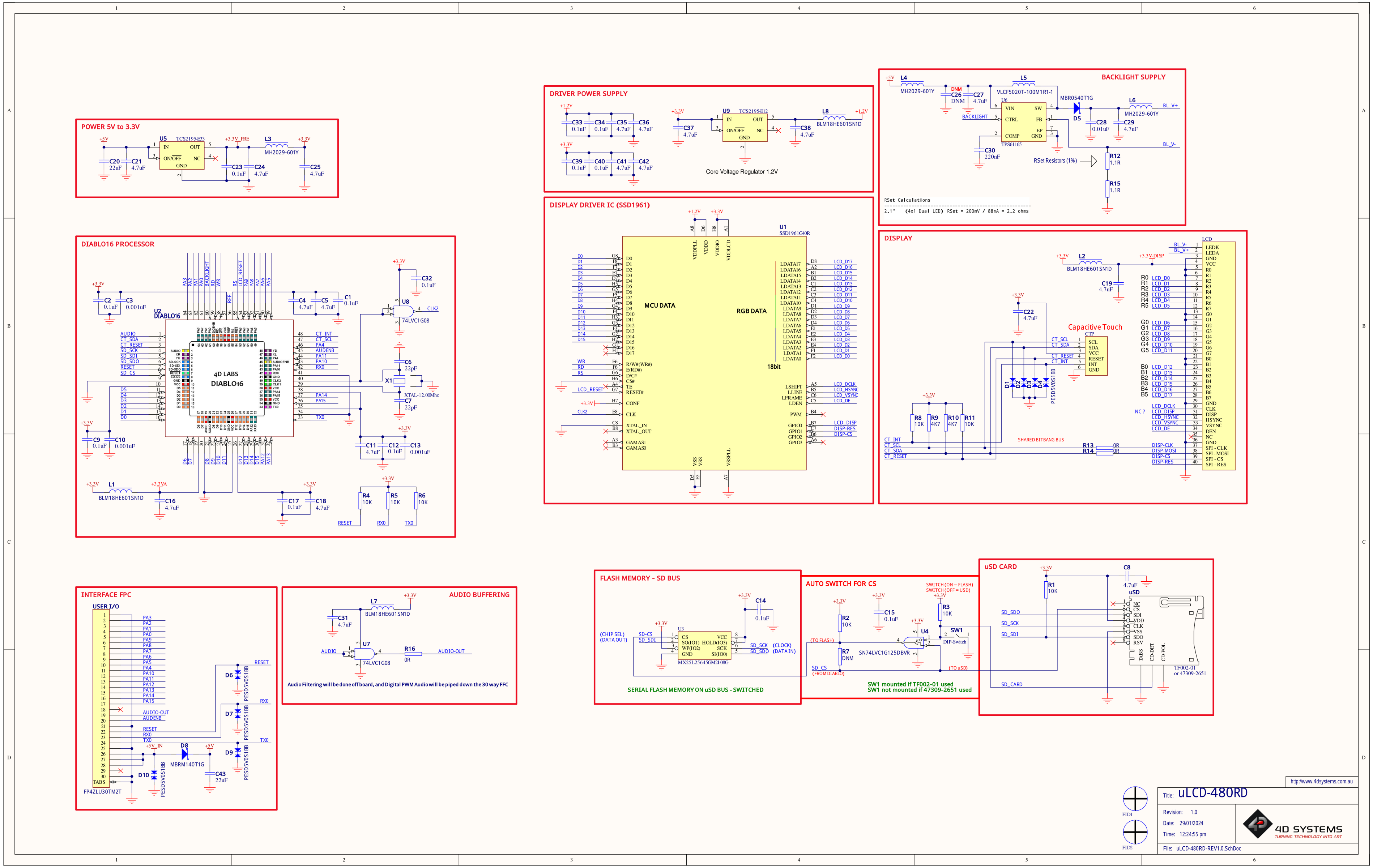
SHEET 1 OF 1

# 15. Mechanical Template Details



Link to DXF file for 1:1 scale printing, available HERE or alternatively HERE.

# 16. Schematic Diagram



# 17. 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 | Units |
|---|---|---|---|---|---|
| Supply Voltage (VCC) | Stable external supply required | 4.0 | 5.0 | 5.5 | V |
| Processor voltage (VP) | | – | 3.3 | – | V |
| 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 |
| Output Low Voltage (VOL) | 3.3V, IOL = 3.4mA | – | – | 0.4 | V |
| Output High Voltage (VOH) | 3.3V, IOL = -2.0mA | 2.4 | – | – | V |
| Capacitive Loading | All pins | – | – | 50 | pF |

**Global Characteristics Based on Operating Conditions**

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| Supply Current (ICC) | 5.0V, uLCD-480RD-CTP-CLB | 10 | 240 | – | mA |
| | 5.0V, uLCD-480RD-CLB | 10 | 230 | – | mA |
| Display Endurance | Hours of operation, measured to when the display is 50% original brightness | 30000 | – | – | H |
| Touch Screen Transparency | | 80 | – | – | % |
| Flash Memory Endurance | DIABLO-16 PmmC Programming | – | 10000 | – | E/W |

**LCD DISPLAY INFORMATION (IPS DISPLAY)**

| Parameter | Conditions | Specification |
|---|---|---|
| Display Type | | IPS - TFT Transmissive LCD |
| Display Size | | 2.1" Diagonal (Round Display) |
| Display Resolution | | 480 x 480 (Round) |
| Display Brightness | 5V Supply | 500 cd/m2 (typical) |
| Display Contrast Ratio | Typical | 1000:1 |
| Display Viewing Angles | Above Centre (Typical) | 85 Degrees |
| | Below Centre (Typical) | 85 Degrees |
| | Left of Centre (Typical) | 85 Degrees |
| | Right of Centre (Typical) | 85 Degrees |
| Display Viewing Direction | | ALL (wide viewing IPS Display) |
| Display Backlighting | White LED Backlighting | 1x4 Dual Parallel LED's |
| Pixel Pitch | | 0.111 x 0.111mm (Square pixels) |
| Pixel Density | Number of pixels in 1 row in 25.4mm | 288 DPI/PPI |

## 18. Revision History

**Hardware Revision**

| Revision Number | Date | Description |
|---|---|---|
| 1.0 | 12/12/2023 | Initial public release |

**Datasheet Revision**

| Revision Number | Date | Description |
|---|---|---|
| 0.1 | 30/01/2024 | Internal Use Only |

| ORDERING INFORMATION |
|---|
| **Order Code:** uLCD-480RD-CLB |
| uLCD-480RD-CTP-CLB |
| **Packaging:** Module sealed in an antistatic foam padded 4D Systems box |

# X-ON Electronics

Largest Supplier of Electrical and Electronic Components

***Click to view similar products for*** Display Modules ***category:***

***Click to view products by*** 4D Systems ***manufacturer:***

Other Similar products are found below :

P6171DPR-DC-U P6191PR-DC-U-V3 17978 20135 MIKROMEDIA 3 FOR PIC32MZ CAPACITIVE FPI 19885 RL0F00101000JGDAASA00 1500CSH MOD-OLED-128x64 ULCD-104D ULCD-104DCT GEN4-ESP32-50CT GEN4-ESP32-50CT-CLB GEN4-ESP32-70 GEN4-ESP32-50T GEN4-ESP32-24 GEN4-ESP32-35CT GEN4-ESP32-32 GEN4-ESP32-43T GEN4-ESP32-35 GEN4-ESP32-32CT GEN4-ESP32-24CT GEN4-ULCD-43DT-HR GEN4-ESP32-43CT ESP32-90CT GEN4-ESP32-50 GEN4-ESP32-32CT-CLB ESP32-90T DM2-170H-R10 gen4-uLCD-70P4CT-CLB 6SL32550AA004JA2 6SL32550AA004CA1 3D32HK-200 11769 MIKROMEDIA FOR DSPIC33 MIKROMEDIA FOR PIC24 MIKROMEDIA FOR PIC32 MIKROMEDIA FOR STELLARIS M3 MIKROMEDIA HMI 3.5 RES MIKROMEDIA HMI 5 MIKROMEDIA PLUS FOR FT90X MIKROMEDIA PLUS FOR PIC32MX7 SHIELD MIKROMEDIA 5 FOR TIVA SHIELD MIKROMEDIA CONNECT SHIELD MIKROMEDIA FOR ARM MIKROMEDIA FOR PSOC5LP MIKROMEDIA HMI 4.3 UXB MIKROMEDIA HMI 5 RES MIKROMEDIA HMI 5 UXB MIKROMEDIA HMI 7