

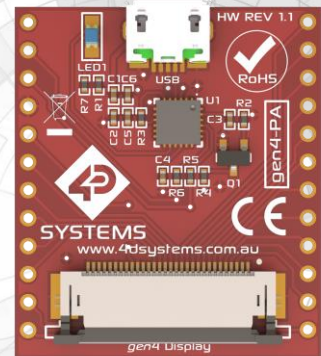


4D SYSTEMS
TURNING TECHNOLOGY INTO ART

gen4-Programming Adaptor

DATASHEET

DOCUMENT DATE: 17th December 2020
DOCUMENT REVISION: 1.3



gen4-PA

Uncontrolled Copy when printed or downloaded.
Please refer to the 4D Systems website for the latest
Revision of this document

Table of Contents

1. Description	3
2. Example Hardware Connections	4
3. FFC Cable	4
4. Schematic Diagram	5
5. Hardware Revision History	6
6. Datasheet Revision History	6
7. Legal Notice	7
8. Contact Information	7

1. Description

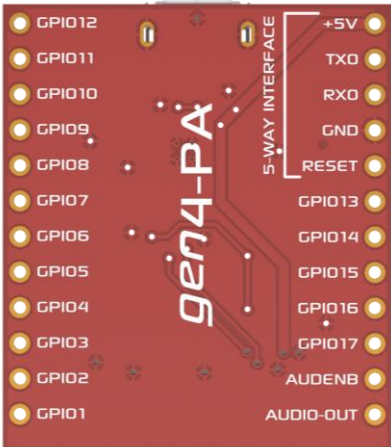
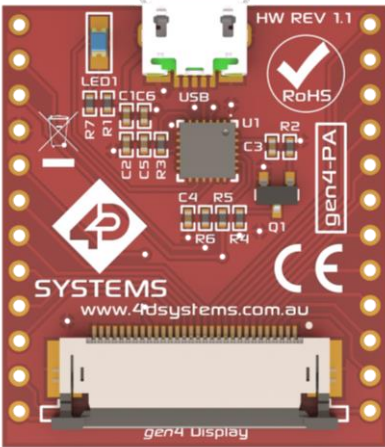
This datasheet covers the gen4-PA (Programming Adaptor) which is compatible with all of the gen4 range of Intelligent Display modules which feature Picaso or Diablo16 graphics processors. It is included in the Starter Kit (SK) packs, and is a quick and easy way to interface and program the gen4 range, without having to break out the signals from the 30-way FFC cable.

The gen4-PA is a programming adaptor for the gen4 display module that also breaks out the signals found on the 30-way FFC cable coming from your gen4 display module. These signals are available via 2.54mm pitch (0.1") pads around the edge of the gen4-PA. It also groups the common 5-way signals found on other 4D Systems products (and the gen4-IB), for easy interface to other products such as our -AR and -PI kits for the Arduino and Raspberry Pi.

The gen4-PA replaces the need for a separate standard programmer, such as the uUSB-PA5 or 4D Programming Cable. The gen4-PA has the circuitry found on the uUSB-PA5-II built into it, and features a microUSB jack, for connection to your PC. Cable is not included with the gen4-PA or in our Starter Kits.

The gen4-PA can be used for programming gen4 display modules, interfacing to a breadboard for prototyping, interfacing to Arduino and Raspberry Pi interfaces (see gen4 -AR and -PI kits), or for interfacing to virtually any host.

The pinout naming, shown on the back of the gen4-PA has been generalised to GPIOx naming, as the gen4-PA can be used with both Picaso and Diablo16 gen4 products, and the naming of GPIO is different for these. Please refer to the following table.

		<table border="1"> <thead> <tr> <th>GEN4-PA</th> <th>DIABLO16</th> <th>PICASO</th> </tr> </thead> <tbody> <tr><td>GPIO1</td><td>PA3</td><td>IO1</td></tr> <tr><td>GPIO2</td><td>PA2</td><td>IO2</td></tr> <tr><td>GPIO3</td><td>PA1</td><td>IO3</td></tr> <tr><td>GPIO4</td><td>PA0</td><td>IO4</td></tr> <tr><td>GPIO5</td><td>PA9</td><td>BUS5</td></tr> <tr><td>GPIO6</td><td>PA8</td><td>BUS4</td></tr> <tr><td>GPIO7</td><td>PA7</td><td>BUS3</td></tr> <tr><td>GPIO8</td><td>PA6</td><td>BUS2</td></tr> <tr><td>GPIO9</td><td>PA5</td><td>BUS1</td></tr> <tr><td>GPIO10</td><td>PA4</td><td>BUS0</td></tr> <tr><td>GPIO11</td><td>PA10</td><td>BUS6</td></tr> <tr><td>GPIO12</td><td>PA11</td><td>BUS7</td></tr> <tr><td>GPIO13</td><td>PA12</td><td>IO5</td></tr> <tr><td>GPIO14</td><td>PA13</td><td>RX1</td></tr> <tr><td>GPIO15</td><td>PA14</td><td>TX1</td></tr> <tr><td>GPIO16</td><td>PA15</td><td>I2C_SCL</td></tr> <tr><td>GPIO17</td><td>N/C</td><td>I2C_SDA</td></tr> </tbody> </table>	GEN4-PA	DIABLO16	PICASO	GPIO1	PA3	IO1	GPIO2	PA2	IO2	GPIO3	PA1	IO3	GPIO4	PA0	IO4	GPIO5	PA9	BUS5	GPIO6	PA8	BUS4	GPIO7	PA7	BUS3	GPIO8	PA6	BUS2	GPIO9	PA5	BUS1	GPIO10	PA4	BUS0	GPIO11	PA10	BUS6	GPIO12	PA11	BUS7	GPIO13	PA12	IO5	GPIO14	PA13	RX1	GPIO15	PA14	TX1	GPIO16	PA15	I2C_SCL	GPIO17	N/C	I2C_SDA
GEN4-PA	DIABLO16	PICASO																																																						
GPIO1	PA3	IO1																																																						
GPIO2	PA2	IO2																																																						
GPIO3	PA1	IO3																																																						
GPIO4	PA0	IO4																																																						
GPIO5	PA9	BUS5																																																						
GPIO6	PA8	BUS4																																																						
GPIO7	PA7	BUS3																																																						
GPIO8	PA6	BUS2																																																						
GPIO9	PA5	BUS1																																																						
GPIO10	PA4	BUS0																																																						
GPIO11	PA10	BUS6																																																						
GPIO12	PA11	BUS7																																																						
GPIO13	PA12	IO5																																																						
GPIO14	PA13	RX1																																																						
GPIO15	PA14	TX1																																																						
GPIO16	PA15	I2C_SCL																																																						
GPIO17	N/C	I2C_SDA																																																						

The gen4-PA utilises the Silicon Labs CP2104 USB to Serial Bridge IC. More information about this can be found from the Silicon Labs website. A link to the driver is available on our website.

- USB 2.0 compliant Full Speed 12Mbps maximum speed.
- Hardware or Xon/Xoff handshaking supported, 300bps to 2Mbps
- UART supports 5, 6, 7, 8 data bits, 1, 1.5, 2 stop bits, odd/even/mark/space and no parity
- Supports Windows 2000 and above, MAC (OSX-8 and above) and Linux (2.4 kernel and above)
- USB powered
- -40 to +85 degrees Celsius temp range

2. Example Hardware Connections

The following pictures illustrate how to connect the gen4-PA to various hardware. Note, the display module illustrated below is the gen4-uLCD-32DT.

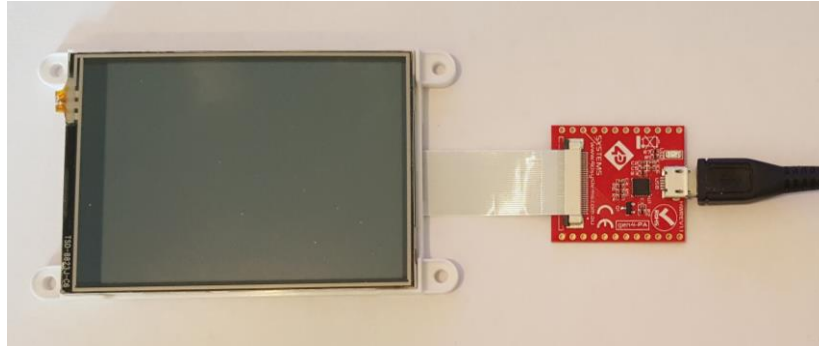


Figure 1. Typical connection of gen4 display module to gen4-PA with USB cable (not included)

When connecting another device (such as an Arduino) to the gen4-PA to interface to the gen4 Display Module, you will not be able to use the TX/RX pins (COM0) on the gen4-PA for the Arduino. The USB chip on the gen4-PA will hold the TX/RX lines and prevent the Arduino being able to communicate with the gen4 Display. To get around this, configure Workshop4 IDE to use a different COM port for communications to the host.

For example, on Diablo display modules, there are 3 additional UARTs which can be assigned to GPIO pins. Configure one of these UART's to use 2 of the Diablo GPIO pins, and then physically wire these 2 pins from the gen4-PA header pads to the Arduino. This leaves the COM0 (TX/RX) pins just for programming the display over USB, and all communications to the host will go out COM1 for example.

In Workshop4, this can be configured easily for the Serial environment, and the ViSi-Genie environment by setting the pins or COM port in the Workshop4 options. In Designer and ViSi, because these are coded 100% by the User, using COM1/2/3 commands instead of COM0 will achieve the same result.

The same can be done for Picaso modules, but it has a fixed COM1 interface predefined.

Check the image in Section 1 (or the Schematic to follow) to show how the Diablo and Picaso pins relate to the markings found on the gen4-PA, so the correct header pads are connected to.

On the back of the gen4-PA, is a '5-way interface' marking. This is designed for connecting non-gen4 display modules to the gen4-PA for programming. This utilises the standard 5V, RX, TX, GND, RESET pins which are found on the 30-way FFC connector, for connecting to uLCD or uOLED modules typically, which do not have FFC connectors like gen4. This should not be used for connecting a Host, due to the reasons stated above.

3. FFC Cable

The Standard FFC cable supplied has the following specifications:

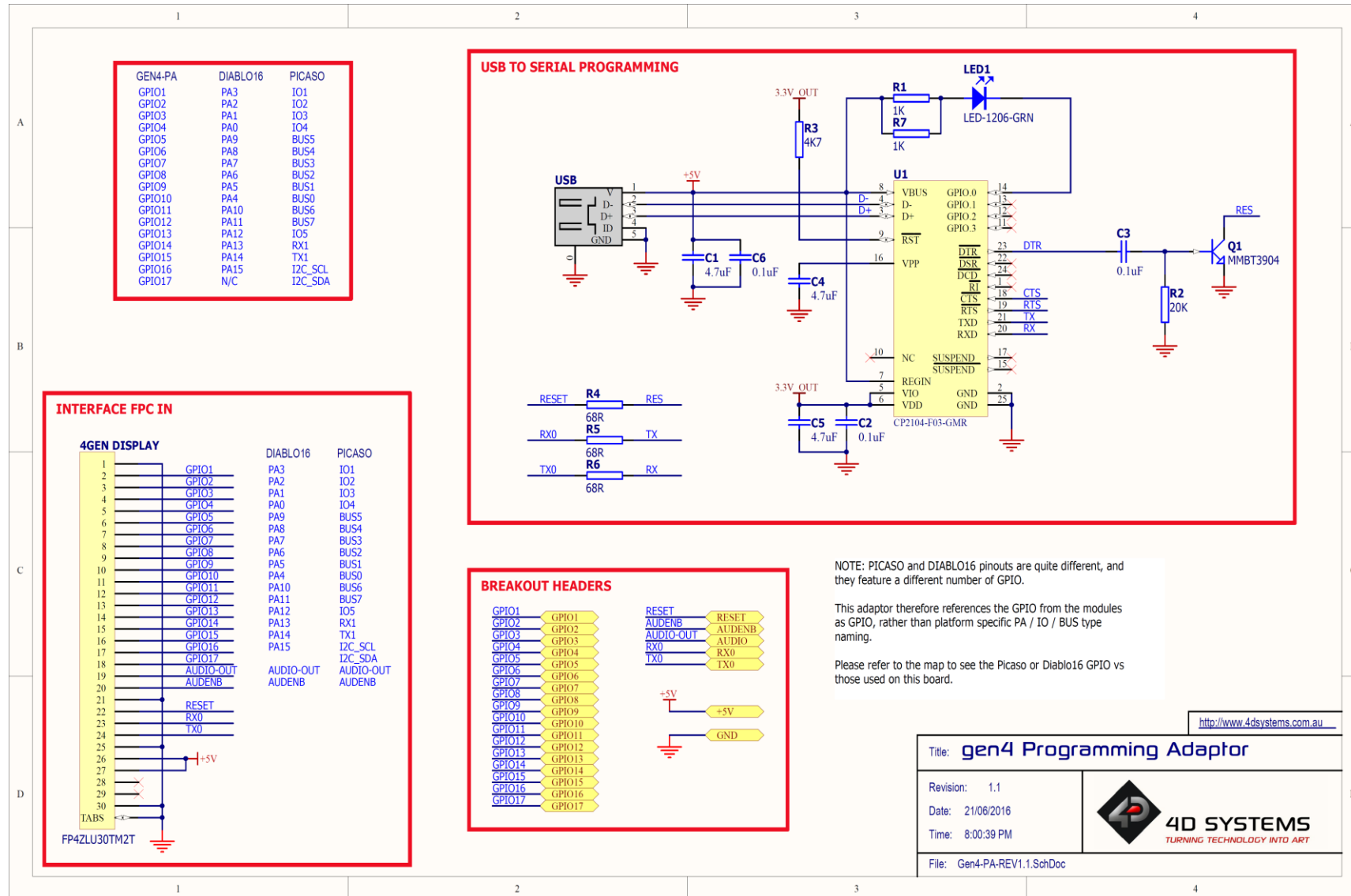
30 Pin Flexible Flat Cable, 150mm Long, 0.5mm (0.02") pitch

Cable Type: AWM 20624 80C 60V VW-1

Heat Resistance 80 Degrees Celsius

Connections on the opposite side at each end (Type B)

4. Schematic Diagram



5. Hardware Revision History

Revision Number	Date	Description
1.1	21/06/2016	Initial Public Release Version

6. Datasheet Revision History

Revision Number	Date	Description
1.1	21/06/2016	Initial Draft
1.2	07/03/2019	Cosmetic Changes to gen4-PA Datasheet
1.3	17/12/2020	Clarification on UART connections to a host via the gen4-PA