# DLP-2232PB-G
*LEAD-FREE*

## USB / Microcontroller Module
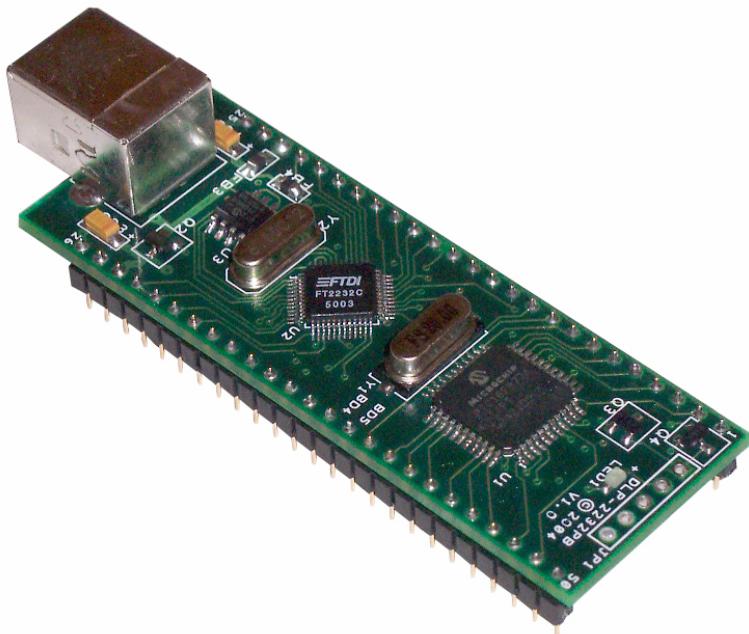
The DLP-2232PB combines the same USB interface used in the DLP-2232M module with a Microchip PIC microcontroller to form a rapid development tool. The 16F877A microcontroller is preprogrammed with basic functionality for accessing the port pins and can be reprogrammed with user hex code via an on-board programming interface, no external programmer required.

**DLP-2232PB FEATURES**

- Send/receive data over USB to a host computer at up to 2 megabits per second
- Standard .9 inch wide DIP footprint, 4 layer pcb with internal ground plane
- 16 digital I/O lines (5 can be configured as A/D inputs) plus the 8-bit data bus available for interfacing to user electronics
- Microchip 16F877A processor with 8K FLASH ROM, 368 bytes RAM, and a multi-channel, 10-bit A/D converter

- "Token I/O" code preprogrammed into the 16F877A's ROM for basic port pin input/output capability including access to the A/D, EEPROM, and external digital temperature sensors. ROM can be easily erased and reprogrammed with built-in Flash programmer or a user-supplied compatible programmer
- On-board Flash programmer takes advantage of 'A' version Microchip microcontroller in that the entire 8K ROM can be programmed typically in under 4 seconds and read in under 1 second
- No in-depth knowledge of USB is required as all USB protocol is handled automatically by the on-board FT2232D and its support circuitry
- Royalty-free device drivers eliminate the need for USB driver development in most cases
- USB bulk or isocronous data-transfer modes
- Required 5V supply can be taken directly from the USB port or supplied by user electronics
- USB 1.1 and USB 2.0 compatible
- USB VID, PID, serial number, and product description strings stored in on-board EEPROM
- Virtual com port (VCP) drivers for:
  - Windows 98 and Windows 98 SE
  - Windows 2000/ME/XP
  - Windows CE**
  - MAC OS-8 and OS-9
  - MAC OS-X**
  - Linux 2.40 and greater
- D2XX (USB direct drivers + DLL S/W interface)
  - Windows 98 and Windows 98 SE
  - Windows 2000/ME/XP

** In planning or under development.

**APPLICATION AREAS**
- Prototype development
- USB ISDN and ADSL modems
- USB interface for digital cameras
- USB interface for MP3 players
- High-speed USB instrumentation
- USB smart-card readers
- Set top box (STB) PC-USB interface
- USB hardware modems
-

## GENERAL DESCRIPTION

The DLP-2232PB provides a cost-effective, microcontroller-based method of interfacing an electronic peripheral to a host computer via USB.

To send data from the peripheral to the host computer, the microcontroller simply writes the byte-wide data into the FT2232D when TXE# is low by placing the data byte on the bus and toggling the WR pin low and then high.  If the FT2232D's transmit buffer fills up or is busy storing the previously written byte, it will take its TXE# high in order to stop further data from being written until some of the FIFO data has been transferred over USB to the host.

When the host sends data to the peripheral over USB, the FT2232D will take RXF# low to let the microcontroller know that at least one byte of data is available.  The microcontroller then reads the data via the RD# pin until RXF# goes high indicating that no more data is available to read. The current data byte is present on the data bus so long as the RD# pin is held low.  When RD# is returned high, the next byte (if any) is prepared to be read.

By using FTDI's virtual COM port drivers, the peripheral looks like a standard COM port to the application software.  Commands to set the baud rate are ignored--the FT2232D always transfers data at its fastest rate regardless of the application's baud-rate setting.  The latest versions of the drivers are available for download from DLP Design's website at http://www.dlpdesign.com.

## FIRMWARE UPLOAD APPLICATION

As mentioned above, the DLP-2232PB is shipped with its microcontroller pre-programmed with firmware for basic port pin input/output capability including access to the A/D, EEPROM, and external digital temperature sensors.  However, if required, the Flash program memory can be erased and reprogrammed with user supplied hex code using a firmware upload application that can be downloaded from the DLP-2232PB product page at www.dlpdesign.com.  Firmware can also be reprogrammed into the Flash program memory via a 5-pin header and user-supplied compatible programmer.

*** Note that Channel A is used exclusively for updating the Flash program memory of the PIC 16F877A and channel B is used to communicate with the PIC at run-time.  Additionally, using Channel A for firmware upload requires the DLL drivers to be selected for Channel A in the EEPROM setup.  This is the default setup for the DLP-2232PB from the factory.

**DRIVER SOFTWARE**

The new combined Windows VCP / D2XX drivers for FT2232D are now available for on-line download. These act as VCP Drivers for Windows 98 / ME / 2000 / XP and D2XX drivers for Windows 2000 / XP.  In the near future, FTDI will be publishing a version including D2XX support for Windows 98 / ME.

FTDI's drivers are provided royalty free on the condition that they are used only with designs incorporating an FTDI device (i.e. the FT2232D).  The latest version of the drivers can be downloaded from dlpdesign.com or ftdichip.com.

The driver download file is a combined set of drivers for Windows 98 / ME / 2000 / XP.  Unzip the file to a blank floppy disk or folder on your PC.  (The drivers can coexist on the same floppy disk or folder since the INF files determine which set of drivers to load for each operating system version.)  Once loaded, the VCP drivers allow your application software running on the host PC to communicate with the DLP-2232PB as though it were connected to a COM (RS-232) port.

In addition to VCP drivers, FTDI's D2XX direct drivers for Windows offer an alternative solution to the VCP drivers that allow application software to interface with the DLP-2232PB using a DLL instead of a virtual com port.  The architecture of the D2XX drivers consists of a Windows WDM driver that communicates with the FT2232D device via the Windows USB stack and a DLL that interfaces the application software (written in VC++, C++ Builder, Delphi, VB, etc.) to the WDM driver.  An INF installation file, uninstaller program, and D2XX Programmer's Guide complete the package.

The D2XX direct drivers add support for simultaneous access and control of multiple FT2232D and FT245BM devices.  The extended open function (FT_OpenEx) allows the device to be opened either by its product description or serial number, both of which can be programmed to be unique.  The list devices function (FT-ListDevices) allows the application software to determine which devices are currently available for use, again by product description or by serial number.

*Download FTDI Application Notes AN232-03, AN232-05, AN232-06, and AN232-07 for detailed instructions on how to install and remove the drivers.*

**EEPROM WRITE UTILITY**

The DLP-2232PB has the option to accept manufacturer-specific information that is written into the on-board 93C56 EEPROM.  Parameters that can be programmed include the VID and the PID identifiers, the manufacturer's product string, and a serial number.  Additionally, the basic modes of operation for the FT2232D (USB-UART, USB-FIFO, etc) and desired drivers (VCP or DLL) are also selected via the configuration settings in the EEPROM.

*** The default configuration for both channels is 245FIFO and D2XX DLL drivers.

MPROG version 2.2 is the latest EEPROM serializer utility from FTDI for the FT2232D device. MPROG is based on the new combinational VCP / D2XX DLL driver package.  This driver package acts as VCP Drivers for Windows 98 / ME / 2000 / XP and D2XX drivers for Windows 2000 / XP. In the near future, FTDI will be publishing a version including D2XX support for Windows 98 / ME.

MPROG will update the contents of the EEPROM even if the VCP drivers are currently selected in the EEPROM.

**QUICK START GUIDE**

*This guide requires the use of a Windows 98/ME/2000/XP PC that is equipped with a USB port.*

1.  Download the combined Windows VCP / D2XX driver set from either dlpdesign.com or ftdichip.com.  Unzip the drivers onto a blank floppy disk or into a folder on the hard drive.

2.  The DLP-2232PB can be configured to receive its operating power from the USB port or from user electronics.  Pins 23 and 24 allow for this configuration.  (Refer to the Pinout Description in the next section for a detailed description of the DLP-2232PB electrical interface.)

    Note:  The board will not operate until a power source has been selected as mentioned in Step 2.

3.  Connect the DLP-2232PB board to the PC via a standard A-B, 6-foot USB cable.  This action initiates the loading of the USB drivers.  When prompted, select the folder where the device drivers were stored in Step 1.  Windows will then complete the installation of the device

drivers for the DLP-2232PB board.  The next time the DLP-2232PB board is attached, the host PC will immediately load the correct drivers without any prompting.  Reboot the PC if prompted to do so.

At this point, the DLP-2232PB is ready for use.

If you wish to change drivers from the VCP to the DLL type (or vice versa), you must run the MPROG utility and change the setup information store in the EEPROM.  The utility can be downloaded from either www.dlpdesign.com or www.ftdichip.com.  The DLP-2232PB is shipped with default VID, PID, etc. values programmed into the EEPROM.  You only need to run the MPROG utility if you want to change the default values.

**TOKEN I/O**

The 16F877A microcontroller on the DLP-2232PB is shipped from the factory preprogrammed with firmware that provides rudimentary access to the port pins via either the VCP or DLL drivers. Features include the ability to read and write individual port pins as well as 8-bit port reads and writes.

The firmware in the DLP-2232PB also provides access to the 16F877A's A/D converter, EEPROM memory, and communications with digital temperature sensing devices.

Commands sent to the Token I/O firmware must adhere to a specific communications protocol. Each command sequence contains the following information:

| | |
|---|---|
| Byte 0: | Number of bytes in command sequence |
| Byte 1: | Command |
| Byte 2…n-1: | Parameter/Data bytes |
| Byte n: | Checksum |

The checksum is computed by exclusive-OR-ing every byte in the string.  If the same checksum is calculated by the 16F877A upon receipt of the packet, then a single-byte reply of 0x55 is returned to the host.  In the event of a checksum error, the 16F877A will return an error code of 0xAA.  For example, setting port PIN_A1 high would require the following string of bytes:

0x03, 0xA6, 0x29, 0x01, 0x8D

Definition of the Bytes:


0x03 – Number of bytes in command (checksum excluded)

0xA6 – Command for set port pin high/low

0x29 – Affected port pin

0x01 – Desired state of port pin

0x8D – Checksum of previous 4 bytes


The port pins equate to hexadecimal numeric constants as defined here:


PORT A:

PIN_A0  0x28,  PIN_A1 0x29,   PIN_A2  0x2A,

PIN_A3  0x2B,  PIN_A4  0x2C,  PIN_A5  0x2D


PORT B:

PIN_B0 0x30,  PIN_B1 0x31,  PIN_B2 0x32,  PIN_B3 0x33,

PIN_B4 0x34,  PIN_B5 0x35,  PIN_B6 0x36,  PIN_B7 0x37


PORT C:

PIN_C0 0x38,  PIN_C1 0x39,  PIN_C2 0x3A,  PIN_C3 0x3B,

PIN_C4 0x3C,  PIN_C5 0x3D,  PIN_C6 0x3E,  PIN_C7 0x3F


PORT D:

PIN_D0 0x40,  PIN_D1 0x41,  PIN_D2 0x42,  PIN_D3 0x43,

PIN_D4 0x44,  PIN_D5 0x45,  PIN_D6 0x46,  PIN_D7 0x47


PORT E:

PIN_E0 0x48,  PIN_E1 0x49,  PIN_E2 0x4A,  PIN_E3 0x4B,

PIN_E4 0x4C,  PIN_E5 0x4D,  PIN_E6 0x4E,  PIN_E7 0x4F


The source code for the Token I/O firmware (developed for the CCS C compiler) is available as a free download from DLP Design upon purchase.  Example Visual C++ source code (for Windows 98/2000/XP) for communicating with the DLP-2232PB via the Token I/O firmware is also available for download.  The windows source code also contains the port pin definitions listed above.

## 0xA5 – Line In – Reads the state of a single port pin

**Parameters:**    Port – Select from available port pins (PIN_A3, PIN_B0, etc)

**Returns:**    1 Byte: State of the port pin (0 or 1)

**Function:**    This function will read the state of a single port pin.  If the data direction for the pin selected was set to output, the direction is first set to input.

**Example:**    0x2, 0xA5, 0x2B, 0x8C - Reads the current state of port pin PIN_A3.


## 0xA6 – Line Out - Sets a single port pin high or low

**Parameters:**    Port – Select from available port pins (PIN_A3, PIN_B3, etc)

    State – 0 or 1

**Returns:**    Undefined

**Function:**    This function will change the output state of a single port pin.  If the data direction for the pin selected was set to input, the direction is first set to output.

**Example:**    0x3, 0xA6, 0x2B, 0x1, 0x8F – Sets port pin PIN_A3 high.


## 0xA7 – Return Board ID

**Parameters:**    None

**Returns:**    6 bytes: "2232PB"

**Function:**    This function will return the ID of the development board currently opened.

**Example:**    0x1, 0xA7, A6 – Reads the board ID.


## 0xA8 – Setup A/D

**Parameters:**    Port Configuration – Selects analog port configuration.  (See command 0xA8 in the example firmware for options.)

    A/D Conversion Clock – Select the source for the A/D conversion clock.  (See command 0xA8 in the example firmware for options.)

**Returns:**    Undefined

**Function:**    This function will select the source for the A/D conversion clock.  (Refer to the datasheet for the 16F877A for a detailed explanation of the conversion clock.)

**Example:**    0x3, 0xA8, 0x89, 0x81, 0xA3 – Sets all available A/D inputs on the DLP-2232PB to analog mode (0x89) and selects Fosc/32 for 20MHz operation (0x81).

### 0xA9 – A/D Conversion

**Parameters:** Analog Port Number – Selects the analog port for the A/D conversion.

**Returns:** 2 Bytes: The 10-bit voltage data; LSB first.

**Function:** This function will set the channel for the A/D conversion, pause 10uS, perform the A/D conversion, and then return 2 bytes to the host (LSB first).  Command 0xA8 must have been previously called to configure the desired analog channel as an A/D input.

**Example:** 0x2, 0xA9, 0x2, 0xA8 – Selects analog channel 2 (AN2), performs the A/D conversion, and returns 2 bytes of data.


### 0xAA – EEPROM Read

**Parameters:** Address – Selects the zero-based address of the location in the 16F877A's internal EEPROM for reading.

**Returns:** 1 Byte: The byte of data read from the EEPROM.

**Function:** This function will read the selected location in the 16F877A's internal EEPROM.

**Example:** 0x2, 0xAA, 0x2, 0xAA – Reads EEPROM Address 2, and returns a single byte of data.


### 0xAB – EEPROM Write

**Parameters:** Address – Selects the zero-based address of the location in the 16F877A's internal EEPROM for writing.
Data **–** Data to be written to the EEPROM.

**Returns:** Undefined.

**Function:** This function will write the selected location in the 16F877A's internal EEPROM.

**Example:** 0x3, 0xAA, 0x2, 0xFE, 0x55 – Writes a value of 0xFE to EEPROM Address 2.

## 0xAC – DS18S20 Start Convert

**Parameters:**   Port Pin – Selects the microcontroller port pin on the 16F877A to be used for communication with the DS18S20 temperature sensor.

**Returns:**   1 Byte: A single byte indicating the result of trying to reset the temperature sensor.

Sensor status definitions:        99: Sensor ready

                                                    8:  Short circuit detected

                                                    2:  No sensor detected

**Function:**   This function will select a port pin for communicating with a DS18S20 temperature sensor and start a temperature conversion.  In addition to power and ground connections, a 1.5K to 4.7K pull-up resistor must be connected between VCC and the data pin of the DS18S20 for proper communication.  The conversion time for the DS18S20 is specified for a maximum of 750 milliseconds.

**Example:**   0x2, 0xAC, 0x30, 0x9E – Starts a temperature conversion on 16F877A port pin B0.

## 0xAD – DS18S20 Read Data

**Parameters:**   Port Pin – Selects the microcontroller port pin on the 16F877A to be used for communication with the DS18S20 temperature sensor.

**Returns:**   9 Bytes: The contents of the scratchpad memory in the DS18S20.

**Function:**   This function will select a port pin for communicating with a DS18S20 temperature sensor and request the current contents of the DS18S20's scratchpad memory.  A conversion (command 0xAC) must have been previously requested for the data to be current.  Additionally, host software must wait at least 750 milliseconds for the conversion to complete before reading the 9 bytes of data.  In addition to power and ground, a 1.5K to 4.7K pull-up resistor must be connected between VCC and the data pin of the DS18S20 for proper communications.  (For an example of how to calculate .1°C resolution temperature data from the 9 bytes of return data, refer to the example Windows source code.)

**Example:**   0x2, 0xAD, 0x30, 0x9F – Requests 9 byte temperature data from 16F877A port pin B0.

**0xAE – Reserved**

**0xAF – Loopback**

| | |
|---|---|
| **Parameters:** | Data byte – The byte of data to be looped back to the host. |
| **Returns:** | 1 Byte: The data byte written. |
| **Function:** | This function will echo the specified byte of data back to the host.  The port pins are not affected. |
| **Example:** | 0x2, 0xAF, 0x12, 0xBF – Requests the data byte 0x12 to be echoed back to the host. |

**0x55 – Port A Read**

**0x59 – Port C Read**

**0x5B – Port D Read**

| | |
|---|---|
| **Parameters:** | None |
| **Returns:** | 1 Byte: The 8-bit data read from the selected port. |
| **Function:** | This function will read the specified 8-bit port and return the results to the host.  The port is automatically configured as an input port.  Port A must be configured for digital I/O (command 0xA8) prior to using this command. |
| **Example:** | 0x1, 0x5B, 0x5A – Requests a read from the 8-bit data bus Port D. |

**0x56 – Port A Write**

**0x5A – Port C Write**

**0x5C – Port D Write**

| | |
|---|---|
| **Parameters:** | Data byte – The byte of data to be written to the 8-bit port. |
| **Returns:** | Undefined |
| **Function:** | This function will write the specified data byte to the 8-pit port.  Data is latched on the port pins until changed by another command.  The port is automatically configured as an output port.  Port A must be configured for digital I/O (command 0xA8) prior to using this command. |
| **Example:** | 0x2, 0x5C, 0x12, 0xBF – Writes the data byte 0x12 to the 8-bit data bus Port D. |

**TABLE 1: DLP-2232PB PINOUT DESCRIPTION**

25                                     1

26                                   50
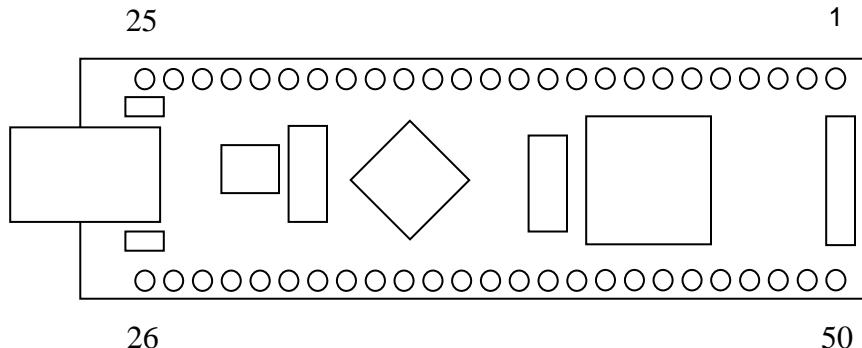
| Pin # | Description |
|---|---|
| 1 | **GROUND** |
| 2 | **B5** (I/O) Port Pin B5 connected to the 16F877A microcontroller. |
| 3 | **UPRST** (In) 16F877A Reset input. Can be left disconnected if not used. |
| 4 | **A0** (I/O) Port Pin A0 connected to the 16F877A microcontroller. A/D Channel 0. |
| 5 | **A1** (I/O) Port Pin A1 connected to the 16F877A microcontroller. A/D Channel 1. |
| 6 | **A2** (I/O) Port Pin A2 connected to the 16F877A microcontroller. A/D Channel 2. |
| 7 | **A3** (I/O) Port Pin A3 connected to the 16F877A microcontroller. A/D Channel 3. |
| 8 | **A4** (I/O) Port Pin A4 connected to the 16F877A microcontroller. Open drain output. |
| 9 | **A5** (I/O) Port Pin A5 connected to the 16F877A microcontroller. A/D Channel 4. |
| 10 | **LED** (I/O) Port Pin E0 connected to the 16F877A microcontroller.  The on-board LED can be driven by 16F877A microcontroller or by external user electronics if E0 is configured as an input. |
| 11 | **GROUND** |
| 12 | **NO CONNECT** |
| 13 | **NO CONNECT** |
| 14 | **NO CONNECT** |
| 15 | **DB7** (I/O) Line 7 of the data bus between the 16F877A and the FT2232D USB-FIFO. |
| 16 | **DB6** (I/O) Line 6 of the data bus between the 16F877A and the FT2232D USB-FIFO. |
| 17 | **DB5** (I/O) Line 5 of the data bus between the 16F877A and the FT2232D USB-FIFO. |
| 18 | **DB4** (I/O) Line 4 of the data bus between the 16F877A and the FT2232D USB-FIFO. |
| 19 | **DB3** (I/O) Line 3 of the data bus between the 16F877A and the FT2232D USB-FIFO. |
| 20 | **DB2** (I/O) Line 2 of the data bus between the 16F877A and the FT2232D USB-FIFO. |
| 21 | **DB1** (I/O) Line 1 of the data bus between the 16F877A and the FT2232D USB-FIFO. |
| 22 | **DB0** (I/O) Line 0 of the data bus between the 16F877A and the FT2232D USB-FIFO. |
| 23 | **EXTVCC** (In) Use for applying main power (4.4 to 5.25 volts) to the module.  Connect to |

| | | PORTVCC if the module is to be powered by the USB port (typical configuration). |
|---|---|---|
| 24 | | **PORTVCC** (Out) Power from USB port.  Connect to EXTVCC if module is to be powered by the USB port (typical configuration).  500mA is the maximum current available to the DLP-2232PB and target electronics if the USB device is configured for high power. |
| 25 | | **GROUND** |
| 26 | | **GROUND** |
| 27 | | **VCCUSB** (Out) Filtered power applied to pin 23. |
| 28 | | **SWVCC** (Out) Power from EXTVCC (Pin 23) controlled via PWREN of the FT2232D and Q2 MOSFET power switch.  R5 and C9 control the power-up rate to help limit inrush current. |
| 29 | | **AC3** (I/O) Port Pin AC3 connected to the FT2232D USB IC. |
| 30 | | **AC2** (I/O) Port Pin AC2 connected to the FT2232D USB IC. |
| 31 | | **AC1** (I/O) Port Pin AC1 connected to the FT2232D USB IC. |
| 32 | | **AC0** (I/O) Port Pin AC0 connected to the FT2232D USB IC. |
| 33 | | **NO CONNECT** |
| 34 | | **NO CONNECT** |
| 35 | | **NO CONNECT** |
| 36 | | **NO CONNECT** |
| 37 | | **NO CONNECT** |
| 38 | | **NO CONNECT** |
| 39 | | **NO CONNECT** |
| 40 | | **GROUND** |
| 41 | | **C0** (I/O) Port Pin C0 connected to the 16F877A microcontroller. |
| 42 | | **C1** (I/O) Port Pin C1 connected to the 16F877A microcontroller. |
| 43 | | **C2** (I/O) Port Pin C2 connected to the 16F877A microcontroller. |
| 44 | | **C3** (I/O) Port Pin C3 connected to the 16F877A microcontroller. |
| 45 | | **C4** (I/O) Port Pin C4 connected to the 16F877A microcontroller. |
| 46 | | **C5** (I/O) Port Pin C5 connected to the 16F877A microcontroller. |
| 47 | | **C6** (I/O) Port Pin C6 connected to the 16F877A microcontroller. |
| 48 | | **C7** (I/O) Port Pin C7 connected to the 16F877A microcontroller. |
| 49 | | **B0** (I/O) Port Pin B0 connected to the 16F877A microcontroller. |
| 50 | | **GROUND** |

**Mechanical Dimensions**
**Inches (mm)**

# DLP-2232PB

20.000

FT2232C

6.000

2.8 typ
(71.1 typ)

.10 typ
(2.54 typ)

.29 typ
(7.4 typ)

.65 typ
(16.5 typ)

.50 typ
(12.7 typ)

.29 typ
(7.4 typ)

.52 typ dia
(13.2 typ)

.019 typ dia
(.46 typ)

.16 typ
(4.2 typ)

.9 typ
(22.9 typ)

.29 typ
(7.3 typ)

1.0 typ
(25.4 typ)
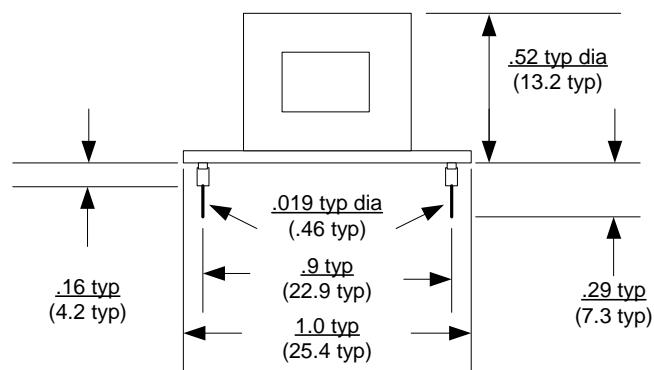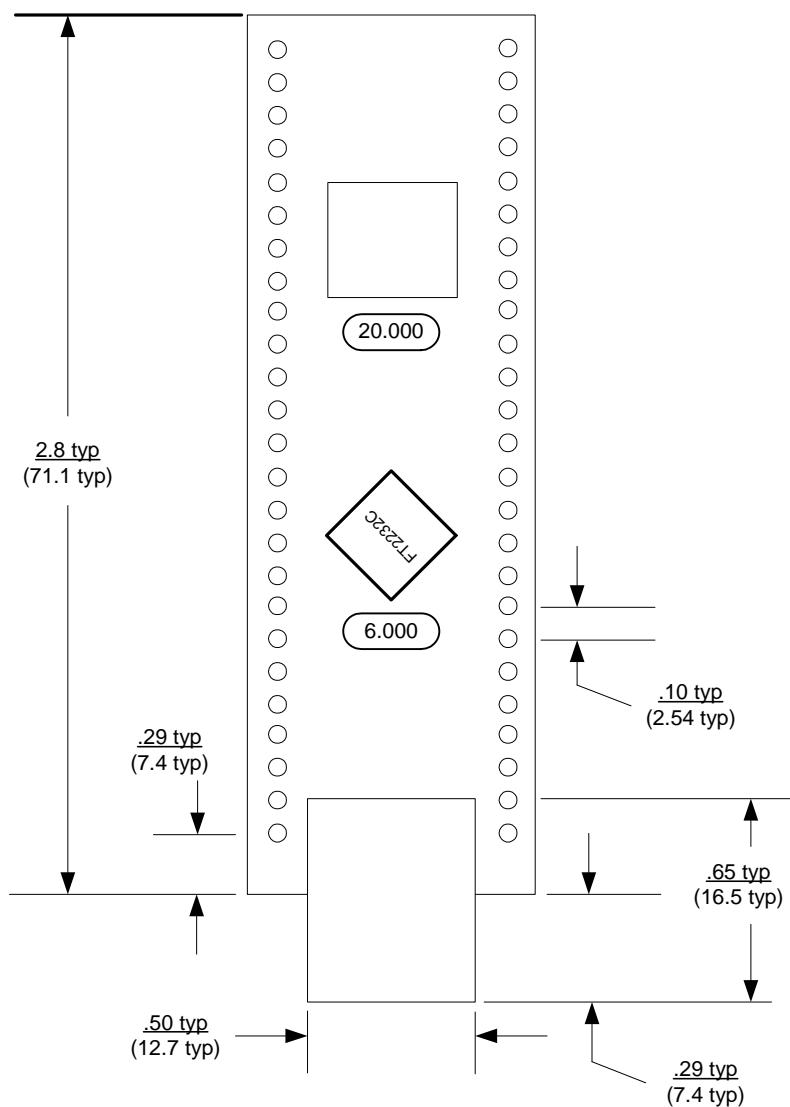
**DISCLAIMER**

Neither the whole nor any part of the information contained within or the product described in this datasheet may be adapted or reproduced in any material or electronic form without the prior written consent of the copyright holder.

This product and its documentation are supplied on an as-is basis, and no warranty as to their suitability for any particular purpose is either made or implied.  DLP Design will not accept any claim for damages whatsoever arising as a result of use or failure of this product.  Your statutory rights are not affected.

This product or any variant of it is not intended for use in any medical appliance, device, or system in which the failure of the product might reasonably be expected to result in personal injury.

This document provides information that may be subject to change without notice.

**CONTACT INFORMATION**

DLP Design, Inc.
1605 Roma Ln.
Allen, TX  75013
Phone:  469-964-8027
Fax:      415-901-4859
Email:   support@dlpdesign.com
Internet:  http://www.dlpdesign.com

**DLP-2232PB MODULE**

FT2232

16 Digital I/O to uP -Plus- 8-bit Data bus (24 total)
1→
2 of which are interrupt input
5 of which are analog input
4 General Purpose I/O from FT2232C

B7 (DTA) <-->
B6 (CLK) -->
0=run 1=Program

General Purpose I/O

Debugger Header (DNS)

PIC16F877A TQFP

CONN PCB 25X2  JP2
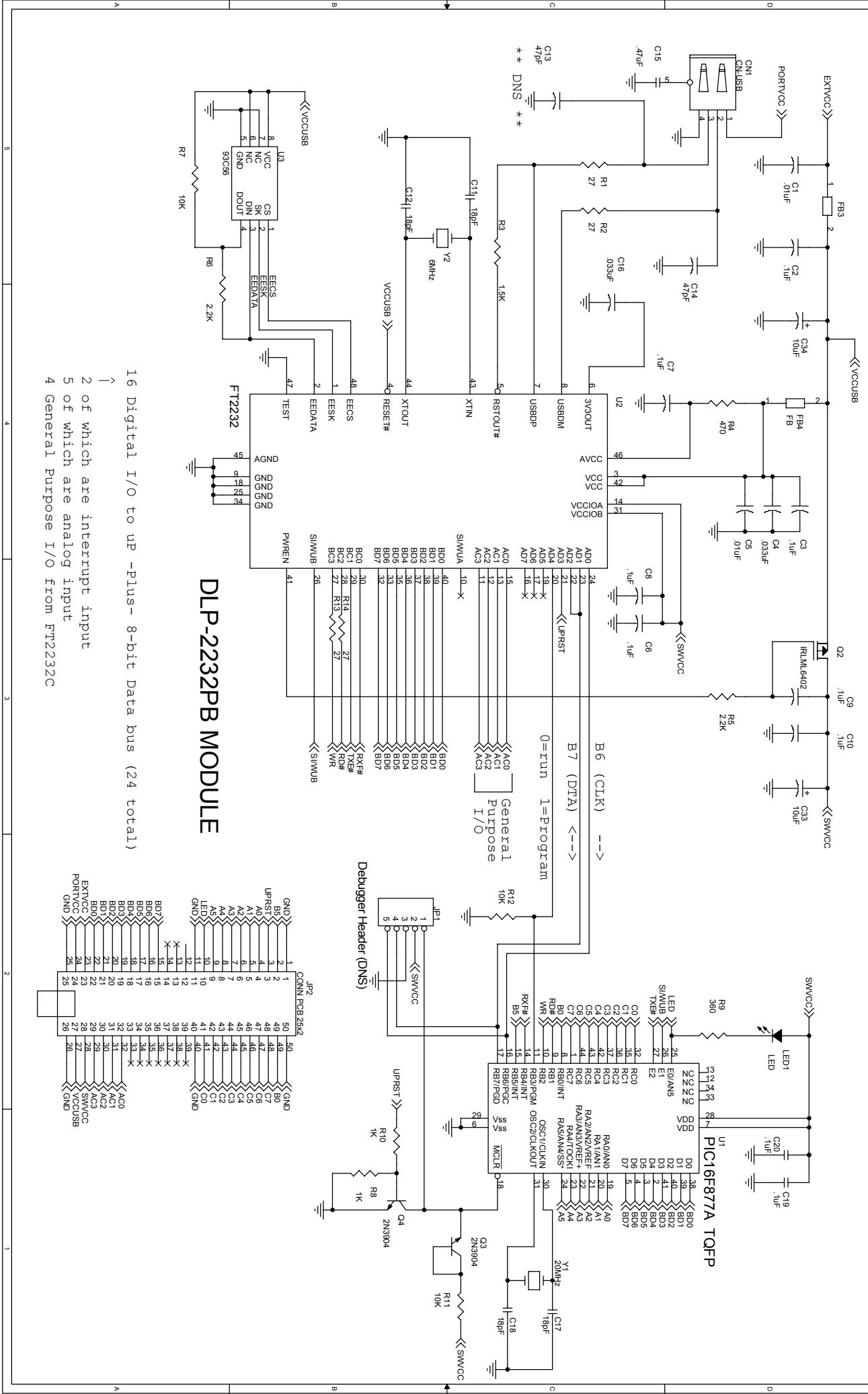
# Mouser Electronics

Authorized Distributor


Click to View Pricing, Inventory, Delivery & Lifecycle Information:


[DLP Design](#):
  [DLP2232PB-CCS](#)  [DLP-2232PB-G](#)