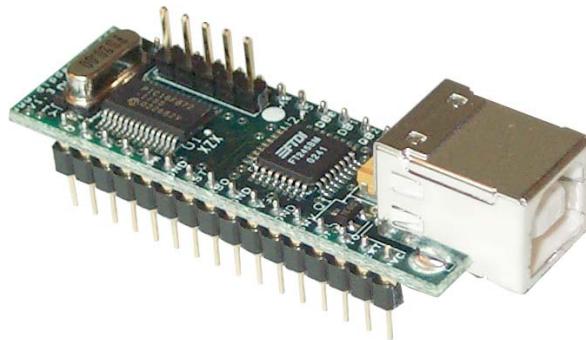


USB / Microcontroller Module



The DLP-245PA combines the same USB interface used in the DLP-USB245M module with a Microchip PIC microcontroller to form a rapid development tool. The 16F872 microcontroller is preprogrammed with basic functionality for accessing the port pins and can be reprogrammed with user hex code via a 5-pin header that is compatible with the DLP-FLASH device programmer (purchased separately).

DLP-245PA FEATURES

- Send/receive data over USB to a host computer at up to 2 megabits per second
- 9 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 16F872 processor with 2K FLASH ROM, 128 bytes RAM, and a multi-channel, 10-bit A/D converter
- “Token I/O” code preprogrammed into the 16F872’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 the DLP FLASH programmer (purchased separately) or a user-supplied compatible programmer
- No in-depth knowledge of USB is required as all USB protocol is handled automatically by the on-board FT245BM and its support circuitry

- Royalty-free device drivers eliminate the need for USB driver development in most cases
- USB bulk or isochronous 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
- USB wireless modems
- USB bar code readers

GENERAL DESCRIPTION

The DLP-245PA 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 FT245BM when TXE# is low. If the FT245BM'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 FT245BM will take RXF# low to let the microcontroller know that at least one byte of data is available. The microcontroller then reads the data until RXF# goes high indicating that no more data is available to 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 FT245BM 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>.

DRIVER SOFTWARE

FTDI's VCP (virtual COM port) driver-executable files are provided royalty free on the condition that they are used only with designs incorporating an FTDI device (i.e. the FT245BM and DLP-245PA). The latest version of the drivers can be downloaded from [dlpdesign.com](http://www.dlpdesign.com) or [ftdichip.com](http://www.ftdichip.com).

The VCP driver download file is a combined set of drivers for Windows 98, Windows ME, and Windows 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-245PA 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-245PA 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 FT245BM 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 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.

Several new features have been recently added to the D2XX drivers. Functions are provided to program the EEPROM (FT_EE_Program) and read the EEPROM (FT_EE_Read). Unused space in the EEPROM is called the user area (EEUA), and functions are provided to access the EEUA. FT_EE_UASize gets its size, FT_EE_UAWrite writes data into it, and FT_EE_UARead is used to read its contents.

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-245PA has the option to accept manufacturer-specific information that is written into the on-board 93C46 EEPROM. Parameters that can be programmed include the VID and the PID identifiers, the manufacturer's product string, and a serial number.

The DLP Design Test Application, available from DLP Design as a separate purchase, can also be used to program the DLP-245PA's EEPROM with manufacturer-specific strings. For more details visit <http://www.dlpdesign.com/test.shtml>.

If you have VCP drivers installed on the PC that will be used to perform the EEPROM write process, you must uninstall these drivers using the uninstaller program (included with the driver files) and install the D2XX drivers prior to running the serializer utility.

QUICK START GUIDE

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

1. Download the DLL version of the device drivers from either dipdesign.com or ftdichip.com. Unzip the drivers onto a blank floppy disk or into a folder on the hard drive.
2. The DLP-245PA can be configured to receive its operating power from the USB port or from user electronics. Pins 14, 15, and 16 allow for this configuration. (Refer to the Pinout Description in the next section for a detailed description of the DLP-245PA electrical interface.)

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

3. Connect the DLP-245PA 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 DLL version of the device drivers was stored in Step 1. Windows will then complete the installation of the device drivers for the DLP-245PA board. The next time the DLP-245PA board is attached, the host PC will immediately load the correct drivers without any prompting. Reboot the PC if prompted to do so.

The DLP-245PA is shipped with default VID, PID, etc. values programmed into the EEPROM. You only need to run the serializer program if you want to change the default values.

At this point, the DLP-245PA is ready for use. Note that the DLP-245PA will appear non-responsive if data sent from the host PC is not read from the FT245BM device by the 16F872 microcontroller.

If changing drivers from the VCP to the DLL type (or vice versa), you must first uninstall the currently loaded drivers. This is accomplished by first disconnecting the DLP-245PA adapter from the host computer and then running the uninstall program for the currently loaded version of the drivers (Ftdiunin.exe for VCP and Ftd2xxun.exe for DLL). These uninstall programs are included with the driver files.

TOKEN I/O

The 16F872 microcontroller on the DLP-245PA 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-245PA also provides access to the 16F872'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 16F872 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 16F872 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_B4 0x34, PIN_B5 0x35

The source code for the Token I/O firmware (developed for the CCS C compiler) is available as a free download from <http://www.dlpdesign.com/usb/245pa.html>. Example Visual C++ source code (for Windows 98/2000/XP) for communicating with the DLP-245PA via the Token I/O firmware is also available for download. The windows source code also contains the port pin definitions listed above.

TOKEN I/O COMMAND SET

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: 5 bytes: “245PA”
Function: This function will return the ID of the 245xx 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 16F872 for a detailed explanation of the conversion clock.)

Example: 0x3, 0xA8, 0x89, 0x81, 0xA3 – Sets all available A/D inputs on the DLP-245PA 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 16F872'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 16F872'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 16F872'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 16F872'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 16F872 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 16F872 port pin B0.

0xAD – DS18S20 Read Data

Parameters: Port Pin – Selects the microcontroller port pin on the 16F872 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 16F872 port pin B0.

0xAE – Reserved

0xAF – Loopback

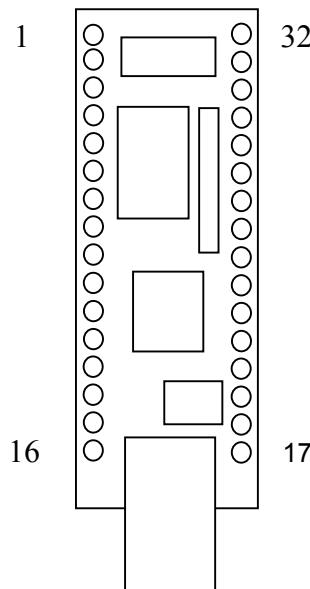
Parameters: Data byte – The byte of data to be looped back to the host.

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.

TABLE 1: DLP-245PA PINOUT DESCRIPTION



Pin #	Description
1	GROUND
2	B0 (I/O) Port Pin E0 connected to the 16F872 microcontroller. A/D Channel 5.
3	B4 (I/O) Port Pin A0 connected to the 16F872 microcontroller. A/D Channel 0.
4	B5 (I/O) Port Pin A1 connected to the 16F872 microcontroller. A/D Channel 1.
5	UPRST (I/O) Port Pin A0 connected to the 16F872 microcontroller.
6	GROUND
7	RESET# (In) Can be used by an external device to reset the FT245BM. Can be left disconnected if not used.
8	RESET0# (Out) Output of the FT245BM's internal Reset Generator. Stays high impedance for ~2ms after VCC>3.5v and the internal clock starts up, then clamps its output to the 3.3v output of the internal regulator. Taking RESET# low will also force RESET0# to go high impedance. RESET0# is NOT affected by a USB Bus Reset.
9	GROUND
10	3V3OUT (Out) Output from the integrated L.D.O. regulator. Its primary purpose is to provide the internal 3.3v supply to the USB transceiver cell and the RSTOUT# pin. A small amount of current (<=5mA) can be drawn from this pin to power external 3.3v logic if required.
11	GROUND
12	SWVCC (Out) Power from EXTVCC (Pin 19) controlled via Pin 10 (POWERN#) of the FT245BM and Q1 MOSFET power switch. R5 and C3 control the power-up rate to help limit inrush current.
13	GROUND
14	VCC-IO (In) 3.0 volt to +5.25 volt VCC to the FT245BM's interface pins 10-12, 14-16, and 18-25.

	When interfacing with 3.3v external logic, connect VCC-IO to the 3.3v supply of the external logic; otherwise, connect to the VCC to drive out at 5v CMOS level. This pin must be connected to VCC from the target electronics or EXTVCC.
15	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).
16	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-245PA and target electronics if the USB device is configured for high power.
17	DB7 (I/O) Line 7 of the data bus between the 16F872 and the FT245BM USB-FIFO.
18	DB6 (I/O) Line 6 of the data bus between the 16F872 and the FT245BM USB-FIFO.
19	DB5 (I/O) Line 5 of the data bus between the 16F872 and the FT245BM USB-FIFO.
20	DB4 (I/O) Line 4 of the data bus between the 16F872 and the FT245BM USB-FIFO.
21	DB3 (I/O) Line 3 of the data bus between the 16F872 and the FT245BM USB-FIFO.
22	DB2 (I/O) Line 2 of the data bus between the 16F872 and the FT245BM USB-FIFO.
23	DB1 (I/O) Line 1 of the data bus between the 16F872 and the FT245BM USB-FIFO.
24	DB0 (I/O) Line 0 of the data bus between the 16F872 and the FT245BM USB-FIFO.
25	A5 (I/O) Port Pin A5 connected to the 16F872 microcontroller. A/D Channel 4.
26	A4 (I/O) Port Pin A4 connected to the 16F872 microcontroller.
27	A3 (I/O) Port Pin A3 connected to the 16F872 microcontroller. A/D Channel 3.
28	A2 (I/O) Port Pin A2 connected to the 16F872 microcontroller. A/D Channel 2.
29	A1 (I/O) Port Pin A1 connected to the 16F872 microcontroller. A/D Channel 1.
30	A0 (I/O) Port Pin A0 connected to the 16F872 microcontroller. A/D Channel 0.
31	GROUND
32	GROUND

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.

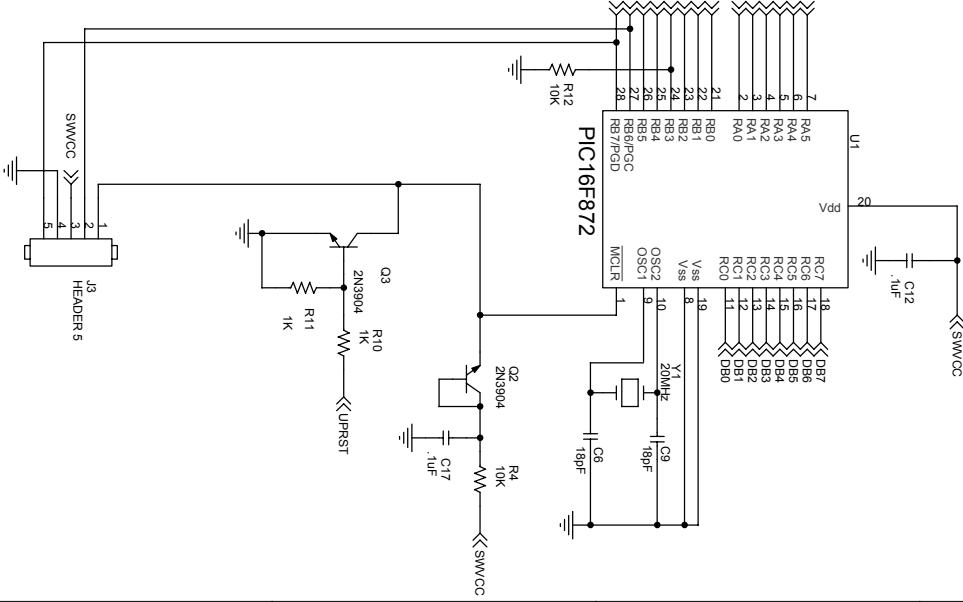
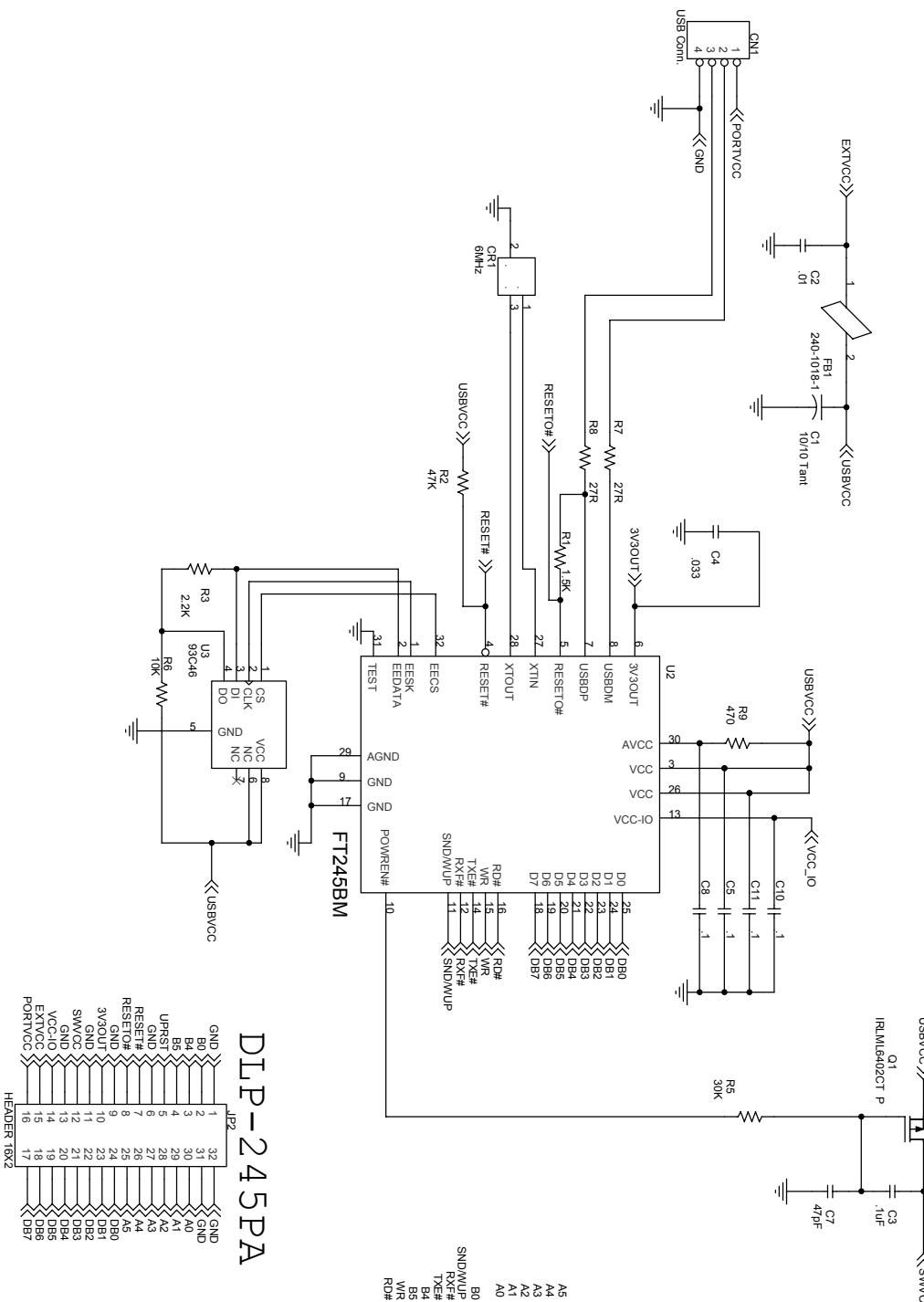
This document provides preliminary information that may be subject to change without notice.

CONTACT INFORMATION

DLP Design
PO Box 503762
San Diego, CA 92150-3762
Phone: 858-513-2777
Fax: 858-513-2777
Email: support@dlpdesign.com
Internet: <http://www.dlpdesign.com>

DLP-245PA
PTC16F872

ELE 698 / 2



Mouser Electronics

Authorized Distributor

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

[DLP Design:](#)

[DLP-245PA](#)