Welcome to E-XFL.COM

Understanding Embedded - Microcontroller, Microprocessor, FPGA Modules

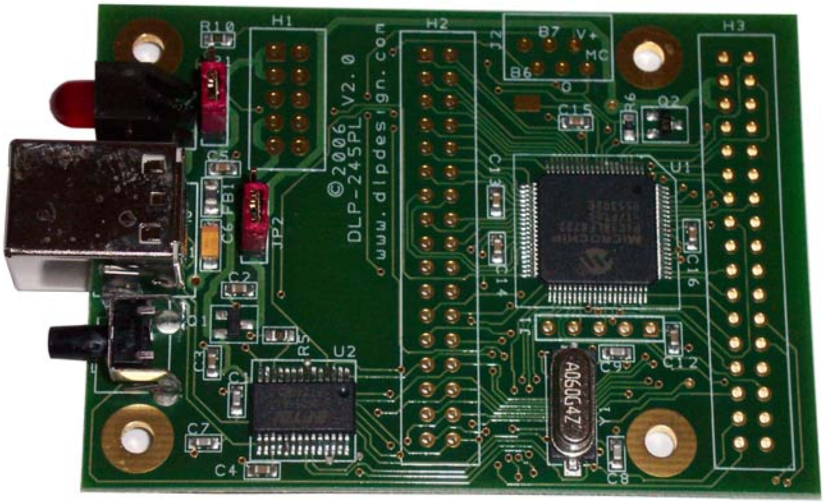
Embedded - Microcontroller, Microprocessor, and FPGA Modules are fundamental components in modern electronic systems, offering a wide range of functionalities and capabilities. Microcontrollers are compact integrated circuits designed to execute specific control tasks within an embedded system. They typically include a processor, memory, and input/output peripherals on a single chip. Microprocessors, on the other hand, are more powerful processing units used in complex computing tasks, often requiring external memory and peripherals. FPGAs (Field Programmable Gate Arrays) are highly flexible devices that can be configured by the user to perform specific logic functions, making them invaluable in applications requiring customization and adaptability.

Applications of Embedded - Microcontroller,

Details	
Product Status	Active
Module/Board Type	MCU, USB Core
Core Processor	PIC16F8722
Co-Processor	FT245RL
Speed	24MHz
Flash Size	128KB
RAM Size	3.75KB
Connector Type	USB - B, Pin Header
Size / Dimension	2.8" x 2" (71.1mm x 50.8mm)
Operating Temperature	-
Purchase URL	https://www.e-xfl.com/product-detail/dlp-design/dlp-245pl-g

DLP-245PL-G

USB / Microcontroller Module (Lead-Free)



The DLP-245PL-G combines the same lead-free USB interface used in the DLP-USB245M-G module with a Microchip PIC microcontroller to form a rapid development tool. The 18LF8722 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 Microchip's MPLAB ICD2 device programmer/debugger (purchased separately).

DLP-245PL FEATURES

- Send/receive data over USB to a host computer at up to 2 megabits per second
- 50 digital I/O lines (16 can be configured as A/D inputs) plus the 8-bit data bus available for interfacing to user electronics
- Microchip 18LF8722 processor with 128K FLASH Program ROM, 3840 bytes RAM, 1024 bytes of data EEPROM and a multi-channel, 10-bit A/D converter
- "Token I/O" code preprogrammed into the 18LF8722's FLASH for basic port pin input/output capability including access to the A/D, EEPROM, and external digital temperature sensors. ROM can be erased and reprogrammed with a device programmer (purchased separately).
- No in-depth knowledge of USB is required as all USB protocol is handled automatically by the on-board FT245RL and its support circuitry
- Royalty-free device drivers eliminate the need for USB driver development in most cases

- 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-chip 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

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-245PL-G 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 FT245RL when TXE# is low. If the FT245RL'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 FT245RL 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 FT245RL 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 **www.dlpdesign.com**.

DRIVER SOFTWARE

FTDI's VCP (Virtual COM Port) driver files are provided royalty free on the condition that they are used only with designs incorporating an FTDI device (i.e. the FT245RL and DLP-245PL-G). The latest version of the drivers can be downloaded from **dlpdesign.com** or **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-245PL-G 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-245PL-G 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 FT245RL 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 FT245RL 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.

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

MPROG is the latest EEPROM serializer utility from FTDI for the FT245RL device. MPROG is based on the new D2XX drivers and will work on Windows 98, Windows ME, and Windows 2000/XP platforms. You must install the latest release of the D2XX drivers in order to run this application.

If you have VCP drivers installed on the PC that will be used to perform the EEPROM write process, you must uninstall these drivers and install the D2XX drivers prior to running MPROG.

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 **dlpdesign.com** or **ftdichip.com**. Unzip the drivers onto a blank floppy disk or into a folder on the hard drive.

2. The DLP-245PL-G can be configured to receive its operating power from the USB port or from an external power source. Jumper JP1 allows for this configuration. Select JP1 Pins 1 and 2 if the DLP-245PL-G is to take its power from the USB port on the host computer; jumper JP1 Pins 2 and 3 if the DLP-245PL-G is to take its power from your external power source.

3. Jumper JP2 selects the power source for the 18LF8722 microcontroller and the interface pins on the FT245RL USB chip. Select JP2 Pins 1 and 2 if 5 volts from the USB port is to be used to power the microcontroller. Select JP2 Pins 2 and 3 if an external supply is to be used to power the microcontroller.

   Note: The board will not operate until a power source has been selected as mentioned in Steps 2 and 3.

4. Connect the DLP-245PL-G 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-245PL-G board. The next time the DLP-245PL-G 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-245PL-G is shipped with default VID, PID, etc. values programmed into the EEPROM. You only need to run MPROG if you want to change the default values.

At this point, the DLP-245PL-G is ready for use. Note that the DLP-245PL-G will appear non-responsive if data sent from the host PC is not read from the FT245RL device by the 18LF8722 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-245PL-G adapter from the host computer and then running Add-Remove-Programs (via Control Panel) to remove the currently loaded drivers.

TOKEN I/O

The 18LF8722 microcontroller on the DLP-245PL-G 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-245PL-G also provides access to the 18LF8722'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 18LF8722 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 18LF8722 will return an error code of 0xAA. For example, setting port PIN_A1 high would require the following string of bytes:

0x03, 0xA6, 0x2D, 0x01, 0x89

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 below:

PORT A:

PIN_A0 0x2C, PIN_A1 0x2D, PIN_A2 0x2E,
PIN_A3 0x2F, PIN_A4 0x30, PIN_A5 0x31

PORT B:

PIN_B0 0x34, PIN_B1 0x35, PIN_B2 0x36, PIN_B3 0x37,
PIN_B4 0x38, PIN_B5 0x39, PIN_B6 0x3A, PIN_B7 0x3B

PORT C:

PIN_C0 0x3C, PIN_C1 0x3D, PIN_C2 0x3E, PIN_C3 0x3F,
PIN_C4 0x40, PIN_C5 0x41, PIN_C6 0x42, PIN_C7 0x43

PORT D:

PIN_D0 0x44, PIN_D1 0x45, PIN_D2 0x46, PIN_D3 0x47,
PIN_D4 0x48, PIN_D5 0x49, PIN_D6 0x4A, PIN_D7 0x4B

PORT E:

PIN_E0 0x4C, PIN_E1 0x4D, PIN_E2 0x4E, PIN_E3 0x4F,
PIN_E4 0x50, PIN_E5 0x51, PIN_E6 0x52, PIN_E7 0x53

PORT F:

PIN_F0 0x54, PIN_F1 0x55, PIN_F2 0x56, PIN_F3 0x57,
PIN_F4 0x58, PIN_F5 0x59, PIN_F6 0x5A, PIN_F7 0x5B

PORT G:

PIN_G0 0x5C, PIN_G1 0x5D, PIN_G2 0x5E, PIN_G3 0x5F,
PIN_G4 0x60

PORT H:

PIN_H0 0x64, PIN_H1 0x65, PIN_H2 0x66, PIN_H3 0x67,
PIN_H4 0x68, PIN_H5 0x69, PIN_H6 0x6A, PIN_H7 0x6B

PORT J:

PIN_J0 0x6C, PIN_J1 0x6D, PIN_J2 0x6E, PIN_J3 0x6F,
PIN_J4 0x70, PIN_J5 0x71, PIN_J6 0x72, PIN_J7 0x73

The source code for the Token I/O firmware (developed for the CCS C compiler) is available as a free download upon purchase of the DLP-245PL-G from **www.dlpdesign.com**. Example Visual C++ source code (for Windows 98/2000/XP) for communicating with the DLP-245PL-G 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, 0x2F, 0x88 - 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, 0x2F, 0x1, 0x8B – Sets port pin PIN_A3 high.

0xA7 – Return Board ID

Parameters: None
Returns: 5 bytes: “245PL”
Function: This function will return the boardID.
Example: 0x1, 0xA7, A6 – Reads the board ID.

0xA8 – Setup A/D

Parameters: Port Configuration – Selects analog port configuration. [See command 0xA8, setup_AD() in the example firmware for options.]
A/D Conversion Clock – Select the source for the A/D conversion clock. [See command 0xA8, setup_AD() in the example firmware for options.]
Returns: Undefined
Function: This function will select the input channel(s) and the source for the A/D conversion clock. (Refer to the datasheet for the 18LF8722 for a detailed explanation of the conversion clock.)
Example: 0x3, 0xA8, 0x00, 0x7, 0xAC – Sets all available A/D inputs on the DLP-245PL-G to analog mode (0x00) and selects Fosc/64 for 20MHz operation (0x7).

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 (2 bytes) – Selects the zero-based address of the location in the 18LF8722's internal EEPROM for reading. The first address byte sent is the MSByte, and the second is the LSByte.
Returns: 1 Byte: The byte of data read from the EEPROM.
Function: This function will read and return the selected location in the 18LF8722's internal EEPROM.
Example: 0x2, 0xAA, 0x1, 0x2, 0xAB – Reads EEPROM Address 258, and returns a single byte of data.

0xAB – EEPROM Write

Parameters: Address (2 bytes) – Selects the zero-based address of the location in the 18LF8722's internal EEPROM for writing. The first address byte sent is the MSByte, and the second is the LSByte.
Data **–** 8-bit data to be written to the EEPROM.
Returns: Undefined.
Function: This function will write the selected location in the 18LF8722's internal EEPROM.
Example: 0x3, 0xAB, 0x1, 0x2, 0xFE, 0x55 – Writes a value of 0xFE to EEPROM Address 258.

0xAC – DS18B20 Start Convert

Parameters: Port Pin – Selects the microcontroller port pin on the 18LF8722 to be used for communication with a DS18B20 temperature sensor.
Returns: 1 Byte: A single byte indicating the result of attempting 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 communication with a DS18B20 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 DS18B20 for proper communication. The conversion time for the DS18B20 is specified for a maximum of 750 milliseconds.
Example: 0x2, 0xAC, 0x34, 0x9A – Starts a temperature conversion on 18LF8722 Port Pin B0.

0xAD – DS18B20 Read Data

Parameters: Port Pin – Selects the microcontroller port pin on the 18LF8722 to be used for communication with the DS18B20 temperature sensor.
Returns: 9 Bytes: The contents of the scratchpad memory in the DS18B20.
Function: This function will select a port pin for communication with a DS18B20 temperature sensor and request the current contents of the DS18B20'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 DS18B20 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, 0x34, 0x9B – Requests 9-byte temperature data from 18LF8720 Port Pin B0.

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.

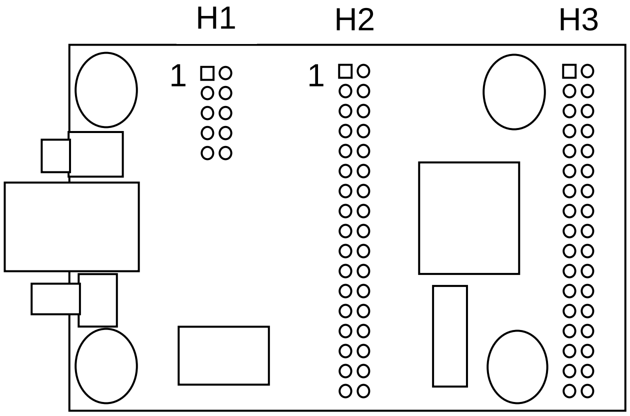
0x55 - Port A Read
0x59 - Port C Read
0x5B - Port D Read
0x5D - Port E Read
0x65 - Port F Read
0x69 - Port H Read
0x6B - Port J Read

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. Analog input ports 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
0x5E - Port E Write
0x66 - Port F Write
0x6A - Port H Write
0x6C - Port J 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, 0x4C – Writes the data byte 0x12 to the 8-bit data bus Port D.

TABLE 1: DLP-245PL PINOUT DESCRIPTION



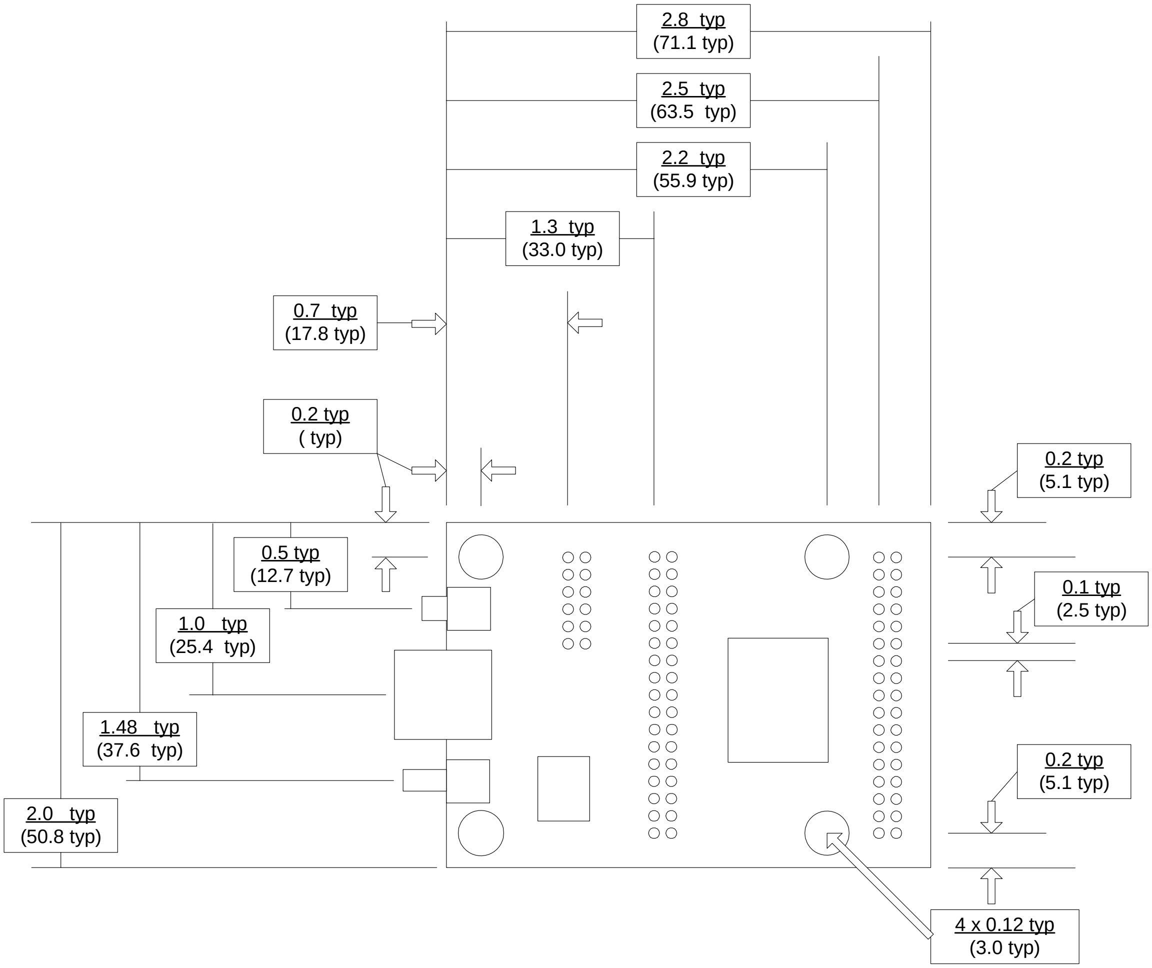
Pin #	H1 Connector Description
1	**EXT5V Input** 5-volt power input for board if not using USB port power.
2	**GROUND**
3	**EXT5V Input** 5-volt power input for board if not using USB port power.
4	**MCLR** Pull to ground to reset the microcontroller.
5	**GROUND**
6	**GROUND**
7	**EXT3-5V** Power input for VCCIO pin on the USB chip and microcontroller. Lower voltages require running the microcontroller at 6MHz (PLL turned off in configuration bits).
8	**GROUND**
9	**EXT3-5V** Power input for VCCIO pin on the USB chip and microcontroller. Lower voltages require running the microcontroller at 6MHz (PLL turned off in configuration bits).
10	**GROUND**

	H2 Connector Description
1	**F6** (I/O) Port Pin F6 connected to the 18LF8722 microcontroller. Analog input Channel 11.
2	**F7** (I/O) Port Pin F7 connected to the 18LF8722 microcontroller.
3	**F4** (I/O) Port Pin F4 connected to the 18LF8722 microcontroller. Analog input Channel 9.
4	**F5** (I/O) Port Pin F5 connected to the 18LF8722 microcontroller. Analog input Channel 10.
5	**F2** (I/O) Port Pin F2 connected to the 18LF8722 microcontroller. Analog input Channel 7.
6	**F3** (I/O) Port Pin F3 connected to the 18LF8722 microcontroller. Analog input Channel 8.
7	**H6** (I/O) Port Pin H6 connected to the 18LF8722 microcontroller. Analog input Channel 14.
8	**H7** (I/O) Port Pin H7 connected to the 18LF8722 microcontroller. Analog input Channel 15.
9	**H4** (I/O) Port Pin H4 connected to the 18LF8722 microcontroller. Analog input Channel 12.
10	**H5** (I/O) Port Pin H5 connected to the 18LF8722 microcontroller. Analog input Channel 13.
11	**F0** (I/O) Port Pin F0 connected to the 18LF8722 microcontroller. Analog input Channel 5.
12	**F1** (I/O) Port Pin F1 connected to the 18LF8722 microcontroller. Analog input Channel 6.
13	**A4** (I/O) Port Pin A4 connected to the 18LF8722 microcontroller. Open drain output.
14	**A5** (I/O) Port Pin A5 connected to the 18LF8722 microcontroller. Analog input Channel 4.
15	**A2** (I/O) Port Pin A2 connected to the 18LF8722 microcontroller. Analog input Channel 2.
16	**A3** (I/O) Port Pin A3 connected to the 18LF8722 microcontroller. Analog input Channel 3.
17	**A0** (I/O) Port Pin A0 connected to the 18LF8722 microcontroller. Analog input Channel 0.
18	**A1** (I/O) Port Pin A1 connected to the 18LF8722 microcontroller. Analog input Channel 1.
19	**GROUND**

20	**GROUND**
21	**GROUND**
22	**GROUND**
23	**H0** (I/O) Port Pin H0 connected to the 18LF8722 microcontroller.
24	**H1** (I/O) Port Pin H1 connected to the 18LF8722 microcontroller.
25	**H2** (I/O) Port Pin H2 connected to the 18LF8722 microcontroller.
26	**H3** (I/O) Port Pin H3 connected to the 18LF8722 microcontroller.
27	**DB0** (I/O) Line 0 of the data bus between the 18LF8722 and the FT245RL USB-FIFO.
28	**DB1** (I/O) Line 1 of the data bus between the 18LF8722 and the FT245RL USB-FIFO.
29	**DB2** (I/O) Line 2 of the data bus between the 18LF8722 and the FT245RL USB-FIFO.
30	**DB3** (I/O) Line 3 of the data bus between the 18LF8722 and the FT245RL USB-FIFO.
31	**DB4** (I/O) Line 4 of the data bus between the 18LF8722 and the FT245RL USB-FIFO.
32	**DB5** (I/O) Line 5 of the data bus between the 18LF8722 and the FT245RL USB-FIFO.
33	**DB6** (I/O) Line 6 of the data bus between the 18LF8722 and the FT245RL USB-FIFO.
34	**DB7** (I/O) Line 7 of the data bus between the 18LF8722 and the FT245RL USB-FIFO.

	H3 Connector Description
1	**USBVCC** 5V power from the USB port.
2	**E0** (I/O) Port Pin E0 connected to the 18LF8722 microcontroller.
3	**SWVCC**
4	**E1** (I/O) Port Pin E1 connected to the 18LF8722 microcontroller.
5	**SWVCC**
6	**E2** (I/O) Port Pin E2 connected to the 18LF8722 microcontroller.
7	**G0** (I/O) Port Pin G0 connected to the 18LF8722 microcontroller.
8	**E3** (I/O) Port Pin E3 connected to the 18LF8722 microcontroller.
9	**G3** (I/O) Port Pin G3 connected to the 18LF8722 microcontroller.
10	**E4** (I/O) Port Pin E4 connected to the 18LF8722 microcontroller.
11	**B0** (I/O) Port Pin B0 connected to the 18LF8722 microcontroller.
12	**E5** (I/O) Port Pin E5 connected to the 18LF8722 microcontroller.
13	**B1** (I/O) Port Pin B1 connected to the 18LF8722 microcontroller.
14	**E6** (I/O) Port Pin E6 connected to the 18LF8722 microcontroller.
15	**B3** (I/O) Port Pin B3 connected to the 18LF8722 microcontroller.
16	**E7** (I/O) Port Pin E7 connected to the 18LF8722 microcontroller.
17	**GROUND**
18	**GROUND**
19	**J0** (I/O) Port Pin J0 connected to the 18LF8722 microcontroller.
20	**C0** (I/O) Port Pin C0 connected to the 18LF8722 microcontroller.
21	**J1** (I/O) Port Pin J1 connected to the 18LF8722 microcontroller.
22	**C1** (I/O) Port Pin C1 connected to the 18LF8722 microcontroller.
23	**J2** (I/O) Port Pin J2 connected to the 18LF8722 microcontroller.
24	**C2** (I/O) Port Pin C2 connected to the 18LF8722 microcontroller.
25	**J3** (I/O) Port Pin J3 connected to the 18LF8722 microcontroller.
26	**C3** (I/O) Port Pin C3 connected to the 18LF8722 microcontroller.
27	**J4** (I/O) Port Pin J4 connected to the 18LF8722 microcontroller.
28	**C4** (I/O) Port Pin C4 connected to the 18LF8722 microcontroller.
29	**LED** (I/O) Port Pin J5 connected to the 18LF8722 microcontroller. LED controlled by 18LF8722.
30	**C5** (I/O) Port Pin C5 connected to the 18LF8722 microcontroller.
31	**J6** (I/O) Port Pin J6 connected to the 18LF8722 microcontroller.
32	**C6** (I/O) Port Pin C6 connected to the 18LF8722 microcontroller.
33	**J7** (I/O) Port Pin J7 connected to the 18LF8722 microcontroller.
34	**C7** (I/O) Port Pin C7 connected to the 18LF8722 microcontroller.

MECHANICAL DRAWINGS (PRELIMINARY)
INCHES (MILLIMETERS) UNLESS OTHERWISE NOTED



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.

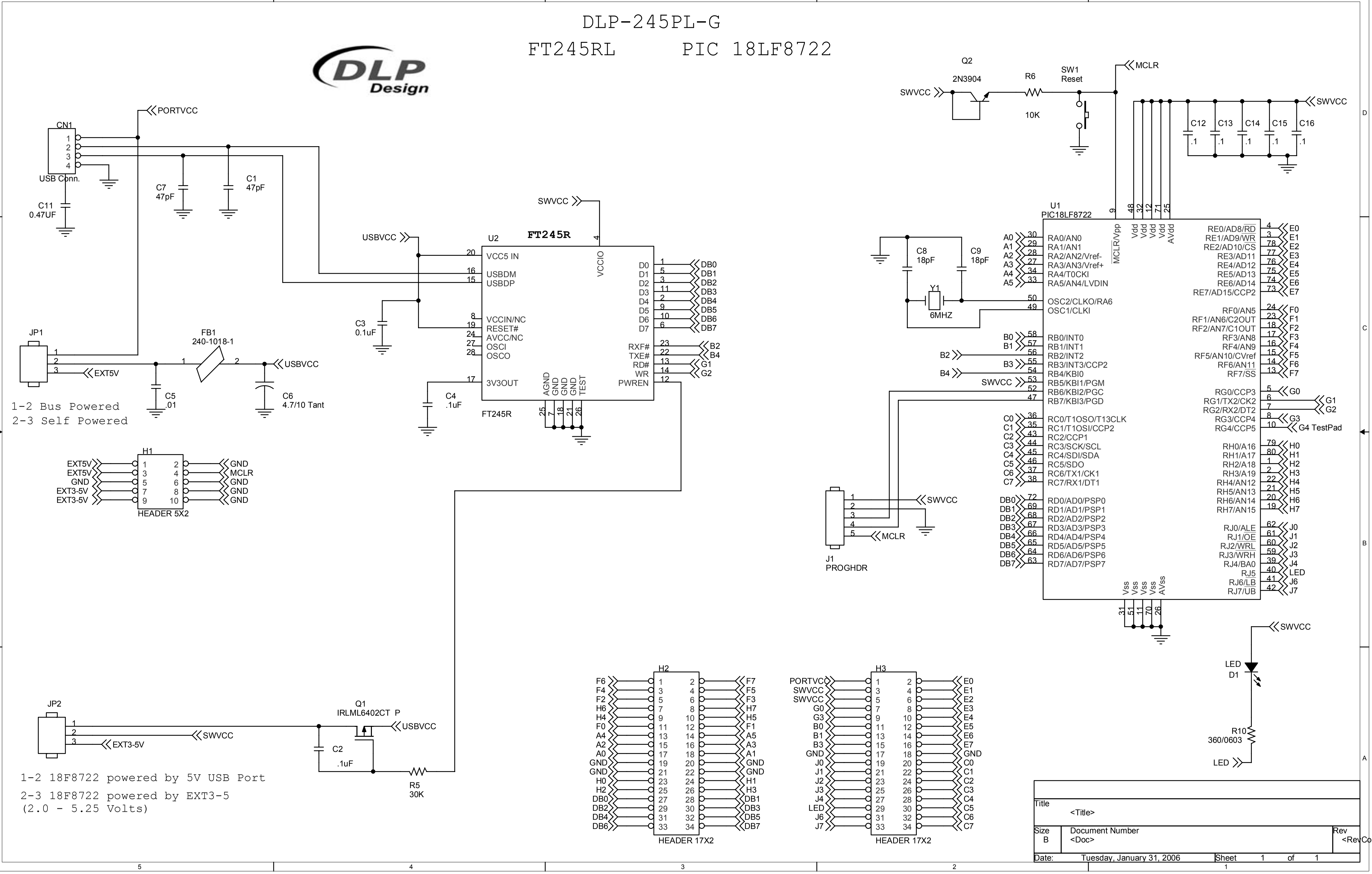
CONTACT INFORMATION

DLP Design, 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-245PL-G

FT245RL PIC 18LF8722



DLP Design
CN1
USB Conn.
1
2
3
4
PORTVCC
C11
0.47UF
C7
47pF
C1
47pF
JP1
1
2
3
EXT5V
FB1
240-1018-1
USBVCC
C5
.01
C6
4.7/10 Tant
1-2 Bus Powered
2-3 Self Powered
H1
EXT5V
EXT5V
GND
EXT3-5V
EXT3-5V
GND
MCLR
GND
GND
GND
HEADER 5X2
SWVCC
USBVCC
U2
FT245R
VCC5 IN
USBDM
USBDP
VCCIN/NC
RESET#
AVCC/NC
OSCI
OSCO
3V3OUT
VCCIO
AGND
GND
GND
GND
TEST
D0
D1
D2
D3
D4
D5
D6
D7
RXF#
TXE#
RD#
WR
PWREN
DB0
DB1
DB2
DB3
DB4
DB5
DB6
DB7
B2
B4
G1
G2
C3
0.1uF
C4
.1uF
FT245R
JP2
EXT3-5V
SWVCC
Q1
IRLML6402CT P
USBVCC
C2
.1uF
R5
30K
1-2 18F8722 powered by 5V USB Port
2-3 18F8722 powered by EXT3-5
(2.0 - 5.25 Volts)
H2
HEADER 17X2
F6 F4 F2 H6 H4 F0 A4 A2 A0 GND GND H0 H2 DB0 DB2 DB4 DB6
F7 F5 F3 H7 H5 F1 A5 A3 A1 GND GND H1 H3 DB1 DB3 DB5 DB7
H3
HEADER 17X2
PORTVCC SWVCC SWVCC G0 G3 B0 B1 B3 GND J0 J1 J2 J3 J4 LED J6 J7
E0 E1 E2 E3 E4 E5 E6 E7 GND C0 C1 C2 C3 C4 C5 C6 C7
Q2
2N3904
SWVCC
R6
10K
SW1
Reset
MCLR
C12 C13 C14 C15 C16
.1
SWVCC
U1
PIC18LF8722
MCLR/Vpp
Vdd
Vdd
Vdd
Vdd
AVdd
C8
18pF
C9
18pF
Y1
6MHZ
RA0/AN0
RA1/AN1
RA2/AN2/Vref-
RA3/AN3/Vref+
RA4/T0CKI
RA5/AN4/LVDIN
OSC2/CLKO/RA6
OSC1/CLKI
RB0/INT0
RB1/INT1
RB2/INT2
RB3/INT3/CCP2
RB4/KBI0
RB5/KBI1/PGM
RB6/KBI2/PGC
RB7/KBI3/PGD
RC0/T1OSO/T13CLK
RC1/T1OSI/CCP2
RC2/CCP1
RC3/SCK/SCL
RC4/SDI/SDA
RC5/SDO
RC6/TX1/CK1
RC7/RX1/DT1
RD0/AD0/PSP0
RD1/AD1/PSP1
RD2/AD2/PSP2
RD3/AD3/PSP3
RD4/AD4/PSP4
RD5/AD5/PSP5
RD6/AD6/PSP6
RD7/AD7/PSP7
Vss
Vss
Vss
Vss
AVss
RE0/AD8/RD
RE1/AD9/WR
RE2/AD10/CS
RE3/AD11
RE4/AD12
RE5/AD13
RE6/AD14
RE7/AD15/CCP2
RF0/AN5
RF1/AN6/C2OUT
RF2/AN7/C1OUT
RF3/AN8
RF4/AN9
RF5/AN10/CVref
RF6/AN11
RF7/SS
RG0/CCP3
RG1/TX2/CK2
RG2/RX2/DT2
RG3/CCP4
RG4/CCP5
RH0/A16
RH1/A17
RH2/A18
RH3/A19
RH4/AN12
RH5/AN13
RH6/AN14
RH7/AN15
RJ0/ALE
RJ1/OE
RJ2/WRL
RJ3/WRH
RJ4/BA0
RJ5
RJ6/LB
RJ7/UB
A0 A1 A2 A3 A4 A5
B0 B1 B2 B3 B4
C0 C1 C2 C3 C4 C5 C6 C7
DB0 DB1 DB2 DB3 DB4 DB5 DB6 DB7
E0 E1 E2 E3 E4 E5 E6 E7
F0 F1 F2 F3 F4 F5 F6 F7
G0 G1 G2 G3 G4 TestPad
H0 H1 H2 H3 H4 H5 H6 H7
J0 J1 J2 J3 J4 LED J6 J7
J1
PROGHDR
SWVCC
MCLR
LED
D1
R10
360/0603
LED
SWVCC
Title
<Title>
Size
B
Document Number
<Doc>
Rev
<RevCode>
Date:
Tuesday, January 31, 2006
Sheet
1
of
1