



**DLP-
USB245R**
LEAD-FREE

USB-PARALLEL FIFO Module

The DLP-USB245R is DLP Design's smallest USB-to-parallel FIFO interface module and utilizes the popular FT245R IC with FTDIChip-ID™ feature from FTDI. In addition to enabling standard USB-to-FIFO designs, both asynchronous and synchronous bit-bang interface modes are also available.

The DLP-USB245R is available in a lead-free, RoHS-compliant, compact 18-pin, 0.1-inch standard DIP footprint.

FEATURES:

- Single-chip USB-to-parallel FIFO bidirectional data-transfer interface.
- Entire USB protocol handled on the FT245R; no USB-specific firmware programming is required.
- Fully integrated 1024-bit EEPROM storing device descriptors and FIFO I/O configuration.
- Data transfer rates up to 1Mbyte per second.
- 256-byte receive buffer and 128-byte transmit buffer utilizing buffer-smoothing technology to allow for high data throughput.
- FTDI's royalty-free Virtual COM Port (VCP) and direct (D2XX) drivers eliminate the requirement for USB driver development in most cases.
- Unique USB FTDIChip-ID™ feature.
- Configurable FIFO interface I/O pins.
- Synchronous and asynchronous bit-bang interface options with RD# and WR# strobes.
- Device supplied preprogrammed with a unique USB serial number.
- Supports bus-powered, self-powered and high-power bus-powered USB configurations.
- Integrated level converter on FIFO interface for connection to external logic running at between +1.8V and +5V.
- True 5V/3.3V/2.8V/1.8V CMOS drive output and TTL input.
- Configurable I/O pin output drive strength.
- Integrated power-on-reset circuit.
- Fully integrated power-supply filtering; no external filtering is required.
- +3.3V to +5.25V single-supply operation.
- Low operating and USB Suspend current.
- UHCI/OHCI/EHCI host-controller compatible.
- USB 2.0 full-speed compatible.
- -40°C to 85°C extended operating temperature range.

APPLICATION AREAS:

- Upgrading legacy peripherals to USB
- Interfacing MCU/PLD/FPGA-based designs to USB
- USB audio and low-bandwidth video data transfer
- PDA-to-USB data transfer
- USB smart card readers
- USB instrumentation
- USB industrial control
- USB MP3 player interface
- USB FLASH card readers and writers
- Set top box PC-USB interface
- USB hardware modems
- USB wireless modems
- USB bar code readers
- USB software and hardware encryption dongles

DRIVER SUPPORT:

Royalty-Free Virtual COM Port (VCP) Drivers for:

- Windows 98, 98SE, ME, 2000, Server 2003, XP and Server 2008
- Windows XP and XP 64-bit
- Windows Vista and Vista 64-bit
- Windows XP Embedded
- Windows CE 4.2, 5.0 and 6.0
- Mac OS 8/9, OS-X
- Linux 2.4 and greater

Royalty-Free D2XX Direct Drivers (USB Drivers + DLL S/W Interface) for:

- Windows 98, 98SE, ME, 2000, Server 2003, XP and Server 2008
- Windows XP and XP 64-bit
- Windows Vista and Vista 64-bit
- Windows XP Embedded
- Windows CE 4.2, 5.0 and 6.0
- Linux 2.4 and greater

The drivers listed above are all available for free download from the DLP Design website www.dlpdesign.com and FTDI website www.ftdichip.com.

Various third-party drivers are also available for other operating systems; see the FTDI website www.ftdichip.com for details.

ABSOLUTE MAXIMUM RATINGS

• Storage Temperature	-65°C to +150°C
• Ambient Temperature (Power Applied)	-40 to +85°C
• VCC Supply Voltage	-0.5V to +6.00V
• DC Input Voltage: Inputs	-0.5V to VCC + 0.5V
• DC Input Voltage: High-Impedance Bidirectionals	-0.5V to VCC + 0.5V
• DC Output Current: Outputs	24mA

D.C. CHARACTERISTICS (AMBIENT TEMPERATURE: -40 TO 85°C)

• VCC Operating-Supply Voltage	4.0 - 5.25V
• VCCIO Operating-Supply Voltage	4.0 - 5.25V
• Operating Supply Current	15mA (Normal Operation)
• Operating Supply Current	100uA USB Suspend

1.0 GENERAL DESCRIPTION

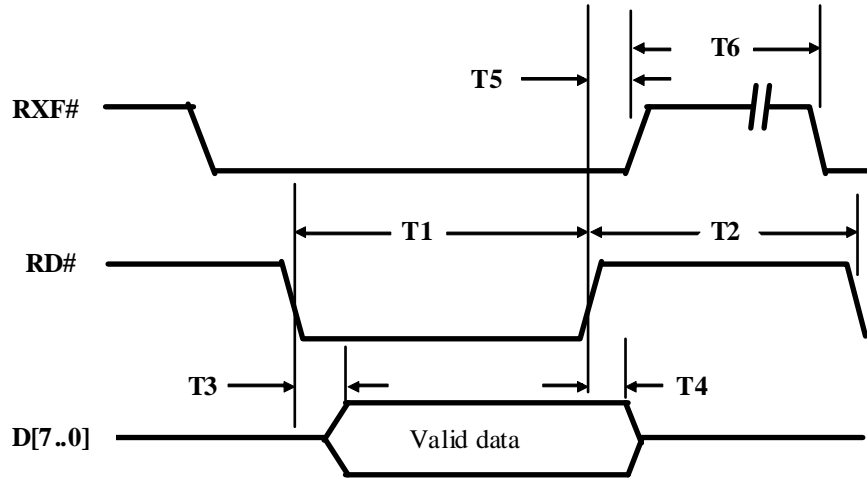
The DLP-USB245M provides an easy, cost-effective method of transferring data to/from a peripheral and a host at up to 8 million bits (1 megabyte) per second. Its simple, FIFO-like design makes it easy to interface to any microcontroller or microprocessor via I/O ports.

To send data from the peripheral to the host computer, simply write the byte-wide data into the module when TXE# is low. If the (128-byte) transmit buffer fills up or is busy storing the previously-written byte, the device takes 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 device will take RXF# low to let the peripheral know that at least one byte of data is available. The peripheral then reads the data until RXF# goes high, thereby indicating that no more data is available to be read.

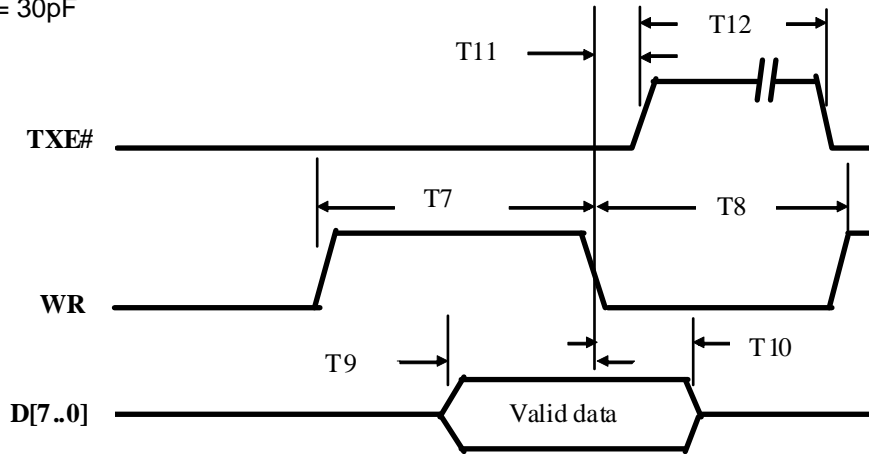
By using FTDI's Virtual COM Port drivers, the peripheral looks like a standard COM port to the application software. Commands to set the baud rate are ignored; the device always transfers data at its fastest rate regardless of the application's baud-rate setting. The latest versions of the drivers are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

2.0 TIMING DIAGRAMS



TIME	DESCRIPTION	MIN	MAX	UNIT
T1	RD# Active Pulse Width	50	-	nS
T2	RD# to RD# Pre-Charge Time	50 + T6	-	nS
T3	RD# Active to Valid Data*	20	50	nS
T4	Valid Data Hold Time from RD# Inactive*	0	-	nS
T5	RD# Inactive to RXF#	0	25	nS
T6	RXF# Inactive After RD Cycle	80	-	nS

*Load = 30pF



TIME	DESCRIPTION	MIN	MAX	UNIT
T7	WR Active Pulse Width	50	-	nS
T8	WR to WR Pre-Charge Time	50	-	nS
T9	Valid Data Setup to WR Falling Edge*	20	-	nS
T10	Valid Data Hold Time from WR Inactive*	0	-	nS
T11	WR Inactive to TXE#	5	25	nS
T12	TXE# Inactive After WR Cycle	80	-	nS

*Load = 30pF

3.0 APPLICATION NOTES

USB devices transfer data in packets. If data is to be sent from the PC, a packet is built up by the application program and is sent via the device driver to the USB scheduler. This scheduler adds a request to the list of tasks that the USB host controller will perform. This will typically take at least 1 millisecond to execute because it will not pick up the new request until the next USB frame (the frame period is 1 millisecond). There is, therefore, sizeable overhead (depending upon your required throughput) associated with moving data from the application to the USB device. If data is sent one byte at a time by an application, this will severely limit the overall throughput of the system.

It must be stressed that in order to achieve maximum throughput, application programs should send or receive data using buffers and not individual characters.

4.0 DRIVER SOFTWARE

FTDI's VCP (Virtual COM Port) driver-executable files are provided royalty free on the condition that they are only used with designs incorporating an FTDI device (i.e. the FT245R and DLP-USB245R). The latest version of the drivers can be downloaded from either www.dlpdesign.com or www.ftdichip.com.

The CDM driver download file is a combined set of drivers for the Windows operating system and contains both the VCP and D2XX driver versions. To download, simply unzip the file to a 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 will allow your application software—running on the host PC—to communicate with the DLP-USB245R 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 FT245R device 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 FT245BR device via the Windows USB stack and a DLL that interfaces with the application software (written in VC++, C++ Builder, Delphi, VB, etc.) to the WDM driver.

The D2XX direct drivers add support for simultaneous access and control of multiple FT245R 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 either product description or serial number.

Additional 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 AN_103 and AN_104 for detailed instructions on how to install the drivers on XP and Vista platforms.

5.0 EEPROM WRITE UTILITY

The DLP-USB245R has the option to accept manufacturer-specific information that is written into EEPROM memory. 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 programming utility for the FT245R device. You must install the latest release of the CDM drivers in order to run this application. If you have CDM drivers installed on the PC that is to perform the EEPROM write process, you can run MPROG and update the EEPROM contents with either mode (VCP or D2XX) active.

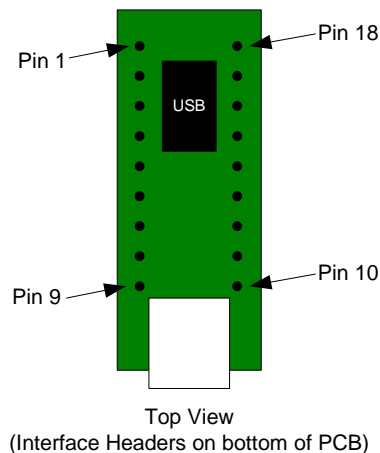
6.0 QUICK START GUIDE

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

1. Download the CDM device drivers from either www.dlpdesign.com or www.ftdichip.com. Unzip the drivers into a folder on the hard drive.
2. Connect the DLP-USB245R board to the PC via a USB 'A' to mini-B 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-USB245R module. The next time the DLP-USB245R module 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-USB245R is ready for use. Note that the DLP-USB245R will appear non-responsive if data sent from the host PC is not read from the FT245R device by an attached microcontroller, microprocessor, DSP, FPGA, ASIC, etc.

7.0 PINOUT DESCRIPTION

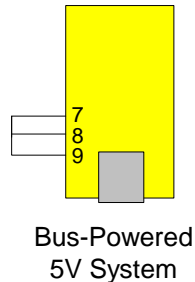


PIN #	DESCRIPTION
1	GROUND
2	DB2 - FIFO Data Bus Bit 2
3	DB7 - FIFO Data Bus Bit 7
4	DB5 - FIFO Data Bus Bit 5
5	DB3 - FIFO Data Bus Bit 3
6	PWREN# - Goes low after the module is configured by USB, then high during USB Suspend. This output can be used to control an external P-channel, logic-level MOSFET switch. Enable the interface pull-down option when using the PWREN# pin in this way. PWREN# should be pulled to VCCIO with 10k-Ohm resistor.
7	VCCIO - +1.8V to +5.25V supply for the FIFO interface. Connect this pin to an external power supply to drive out at +3.3V levels (or another voltage within the specified range), or connect to EXTVCC (Pin 8) to drive out at the +5V CMOS level.
8	EXTVCC - Use for applying main power (4.0 to 5.25 volts) to the module. Connect to PORTVCC (Pin 9) if the module is to be powered by the USB port (typical configuration).
9	PORTVCC - Power from the USB port. Connect to EXTVCC (Pin 8) if the module is to be powered by the USB port (typical configuration). 500mA is the maximum current available to the USB adapter and target electronics if the USB device is configured for high power.
10	GROUND
11	RD# - When pulled low, RD# takes the 8 data lines from a high-impedance state to the current byte in the FIFO's buffer. Taking RD# high returns the data pins to a high-impedance state and prepares the next byte (if available) in the FIFO to be read.
12	WR - When taken from a high to a low state, WR reads the 8 data lines and writes the byte into the FIFO's transmit buffer. Data written to the transmit buffer is sent to the host PC within the TX buffer timeout value (default 16mS) and placed in the buffer that was created when the USB port was opened. The FT245R allows the TX buffer timeout value to be reprogrammed to a value between 1 and 255mS.
13	DB6 - FIFO Data Bus Bit 6
14	TXE# - When high, do not write data into the FIFO. When low, data can be written into the FIFO by strobing WR high, then low. During reset this signal pin is tri-state. Data is latched into the FIFO on the falling edge of the WR pin.
15	RXF# - When low, at least 1 byte is present in the FIFO's receive buffer and is ready to be read with RD#. RXF# goes high when the receive buffer is empty. During reset this signal pin is tri-state. If the Remote Wakeup option is enabled in the internal EEPROM, during USB Suspend Mode (PWREN#=1) RXF# becomes an input. This can be used to wake up the USB host from Suspend Mode by strobing this pin low for a minimum of 20ms which will cause the device to request a resume on the USB bus.
16	DB1 - FIFO Data Bus Bit 1
17	DB4 - FIFO Data Bus Bit 4
18	DB0 - FIFO Data Bus Bit 0

8.0 DEVICE CONFIGURATION EXAMPLES

USB Bus-Powered and Self-Powered Configurations

Figure 1.



The figure above illustrates a typical USB bus-powered configuration. A USB bus-powered device gets its power from the USB bus. Basic rules for USB bus-powered devices are as follows:

1. On plug-in to USB, the module and external circuitry should draw no more than 100mA on the 5-volt line from the USB host.
2. In USB Suspend Mode, the module and external circuitry should draw no more than 2.5mA.
3. A bus-powered, high-power USB device (one that draws more than 100mA) should use one of the CBUS pins configured as PWREN# to keep the current below 100mA on plug-in and below 2.5mA on USB Suspend.
4. A design that consumes more than 100mA in total cannot be plugged into a USB bus-powered hub.
5. No USB target system can draw more than 500mA from the USB bus. The power descriptors in the internal EEPROM of the FT245R should be programmed to match the total current drawn by the target system.

Note: If using PWREN# (available using the CBUS), the I/O pin should be pulled to VCCIO using a 10k-ohm resistor.

Figure 2.

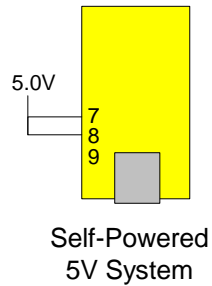


Figure 2 illustrates a typical USB self-powered configuration. A USB self-powered device gets its power from its own power supply and does not draw current from the USB bus. Basic rules for USB self-powered devices are as follows:

1. A self-powered device should not force current down the USB bus when the USB host or hub controller is powered down.
2. A self-powered device can take as much current as it likes during normal operation and during USB Suspend as it has its own power supply.
3. A self-powered device can be used with any USB host and both bus- and self-powered USB hubs.

Figure 3.

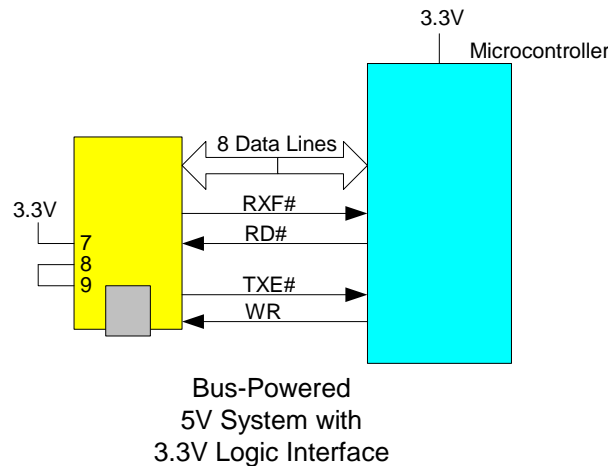


Figure 3 shows how to configure the DLP-USB245R to interface with a 3.3V logic device. In this example, the target electronics provide the 3.3 volts via the VCCIO line (Pin 7) which, in turn, will cause the FT245R interface I/O pins to drive out at the 3.3V level.

9.0 BUS-POWERED CIRCUIT WITH POWER CONTROL

USB bus-powered circuits need to be able to power down in USB Suspend Mode in order to meet the $\leq 2.5\text{mA}$ total Suspend current requirement (including external logic):

Figure 4.

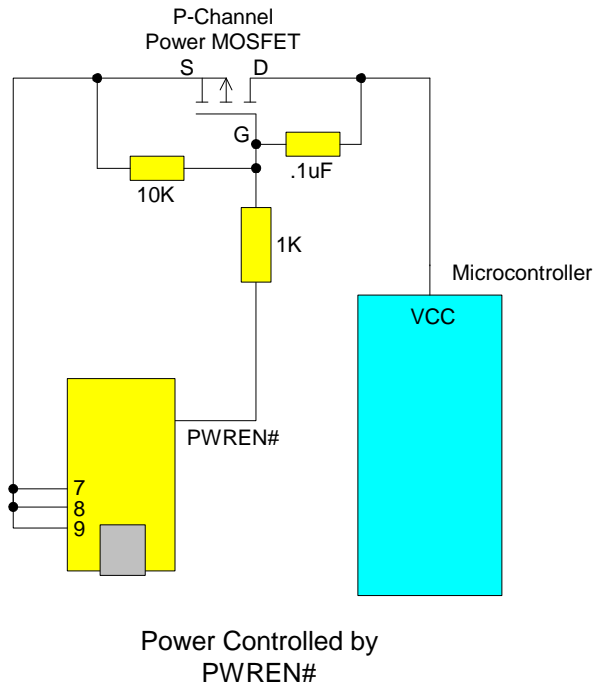
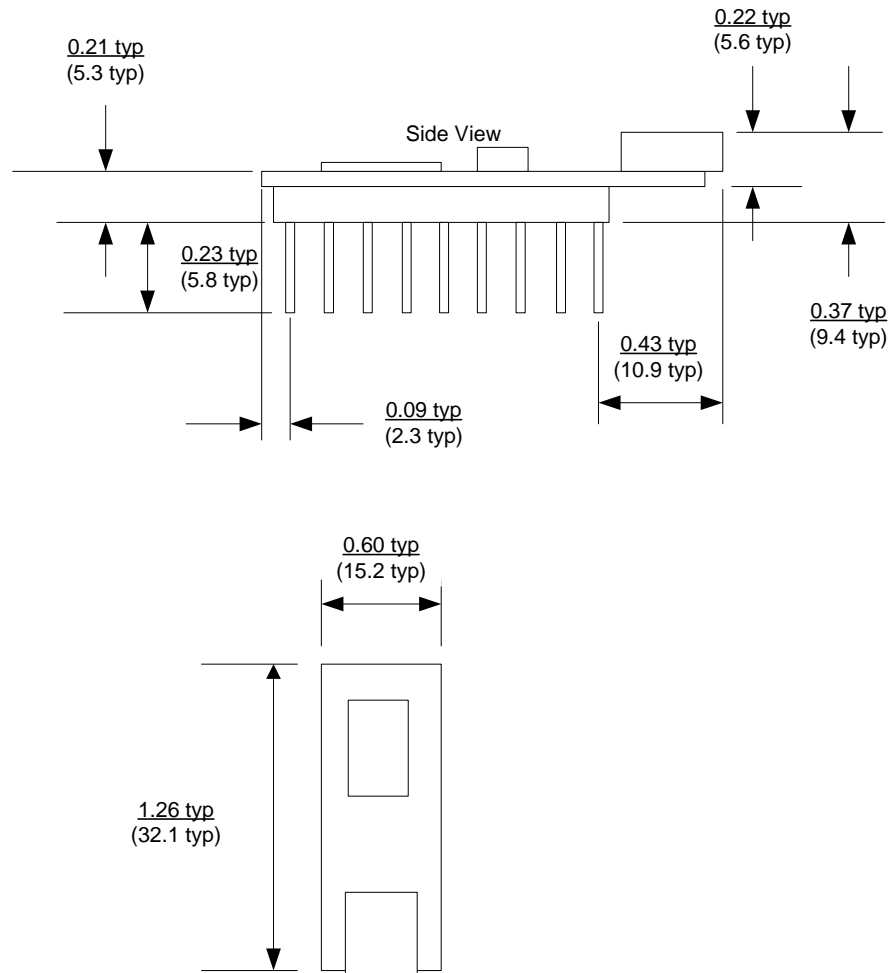


Figure 4 shows how to use a discrete P-Channel MOSFET to control the power to external logic circuits. This “soft-start” circuit accommodates designs that draw more than 100mA at power up.

Please note the following points in connection with power-controlled designs:

1. The logic to be controlled must have its own reset circuitry so that it will automatically reset itself when power is reapplied on coming out of Suspend.
2. Set the soft pull-down option bit in the FT245R EEPROM.
3. For USB high-power, bus-powered applications (ones that consume greater than 100mA, and up to 500mA of current from the USB bus), the power consumption of the application must be set in the Max Power Field in the internal FT245R EEPROM. A high-power, bus-powered application uses the descriptor in the internal FT245R EEPROM to inform the system of its power requirements.
4. PWREN# gets its VCC from VCCIO. For designs using 3.3-volt logic, ensure that VCCIO is not powered down during Suspend.

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



11.0 DISCLAIMER

© DLP Design, Inc., 2008

Neither the whole nor any part of the information contained herein nor the product described in this manual 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, Inc. will not accept any claim for damages whatsoever arising as a result of the 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.

12.0 CONTACT INFORMATION

DLP Design, Inc.
1605 Roma Lane
Allen, TX 75013

Phone: 469-964-8027
Fax: 415-901-4859

Email Sales: sales@dlpdesign.com
Email Support: support@dlpdesign.com
Website URL: <http://www.dlpdesign.com>

