

---

## USB / Microcontroller Module



The DLP-245SY-G combines the 4<sup>th</sup> generation FT245RL USB interface IC from FTDI with a Parallax SX48 microcontroller to form a rapid development tool. The SX48 microcontroller is preprogrammed with basic functionality for accessing the port pins and can be reprogrammed with user developed firmware via a 5-pin header (4 pins used for compatibility with device programmers and debuggers from other manufacturers).

### **DLP-245SY-G FEATURES**

- Send/receive data over USB to a host computer at up to 7 megabits per second
- 20 digital I/O lines plus the 8-bit data bus available for interfacing to user electronics
- Uvicom SX48 processor with 4K FLASH ROM, 262 bytes RAM, and a 2K (256x8) EEPROM
- Microcontroller Supervisor -- microcontroller held in reset until supply voltage reaches stable operating level
- "Token I/O" code preprogrammed into the SX48's ROM for basic port pin input/output capability including access to the EEPROM, and external digital temperature sensors. ROM can be easily erased and reprogrammed using 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
- 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 EEPROM area of USB IC
- 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/x64

### **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-245SY-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 (up to 3 megabits per second using VCP drivers) 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 FT245RL and DLP-245SY-G). The latest version of the drivers can be downloaded from [dlpdesign.com](http://dlpdesign.com) or [ftdichip.com](http://ftdichip.com). Once loaded, the VCP drivers allow your application software running on the host PC to communicate with the DLP-245SY-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-245SY-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.

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-245SY-G has the option to accept manufacturer-specific information that is written into the EEPROM area of the FT245RL. 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 uses 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.

## **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](http://dlpdesign.com) or [ftdichip.com](http://ftdichip.com). Unzip the drivers onto a blank floppy disk or into a folder on the hard drive.
2. The board can be configured to receive its operating power from the USB port or from user electronics. Pins 18, 19, and 20 allow for this configuration. (Refer to the Pinout Description in the next section for a detailed description of the DLP-245SY-G electrical interface.)

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

3. Connect the DLP-245SY-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-245SY-G. The next time the DLP-245SY-G is

attached, the host PC will immediately load the correct drivers without any prompting.  
Reboot the PC if prompted to do so.

The DLP-245SY-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-245SY-G is ready for use. Note that the DLP-245SY-G will appear non-responsive if data sent from the host PC is not read from the FT245RL device by the SX48 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 disconnecting the DLP-245SY-G adapter from the host computer and then using Add-Remove-Programs from Control Panel.

## **TOKEN I/O**

The SX48 microcontroller on the DLP-245SY-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-245SY-G also provides access to the on-board 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 SX48 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 SX48 will return an error code of 0xAA. For example, setting port PIN\_B1 high would require the following string of bytes:

0x03, 0xA6, 0xB1, 0x01, 0x15

Definition of the Bytes:

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

0xA6 – Command for set port pin high/low

0xB1 – Affected port pin

0x01 – Desired state of port pin

0x15 – Checksum of previous 4 bytes

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

PORT B:

PIN\_B0 0xB0, PIN\_B1 0xB1, PIN\_B2 0xB2, PIN\_B3 0xB3,

PORT C:

PIN\_C0 0xC0, PIN\_C1 0xC1, PIN\_C2 0xC2, PIN\_C3 0xC3,

PIN\_C4 0xC4, PIN\_C5 0xC5, PIN\_C6 0xC6, PIN\_C7 0xC7,

PORT D:

PIN\_D0 0xD0, PIN\_D1 0xD1, PIN\_D2 0xD2, PIN\_D3 0xD3,

PIN\_D4 0xD4, PIN\_D5 0xD5, PIN\_D6 0xD6, PIN\_D7 0xD7

PORT E:

PIN\_E0 0xE0, PIN\_E1 0xE1, PIN\_E2 0xE2, PIN\_E3 0xE3,

PIN\_E4 0xE4, PIN\_E5 0xE5, PIN\_E6 0xE6, PIN\_E7 0xE7

The source code for the Token I/O firmware (developed for the SnXC C compiler from Grich RC) is available as a free download upon purchase of the DLP-245SY-G. Example Visual C++ source code (for Windows 98/2000/XP) for communicating with the DLP-245SY-G via the Token I/O firmware is also available for download upon purchase. 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, 0xA3, 0x3 - 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, 0xA3, 0x1, 0x7 – Sets port pin PIN\_A3 high.

### 0xA7 – Return Board ID

**Parameters:** None  
**Returns:** 5 bytes: "245SY"  
**Function:** This function will return the ID of the 245xx board currently opened.  
**Example:** 0x1, 0xA7, A6 – Reads the board ID.

### 0xAA – EEPROM Read

**Parameters:** Address – Selects the zero-based address of the location in the EEPROM for reading.  
**Returns:** 1 Byte: The byte of data read from the EEPROM.  
**Function:** This function will read the selected location in the 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 EEPROM for writing.

Data – Data to be written to the EEPROM.

**Returns:** Undefined.

**Function:** This function will write the selected location in the 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 SX48 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, 0xB0, 0x1E – Starts a temperature conversion on SX48 port pin B0.

## 0xAD – DS18S20 Read Data

**Parameters:** Port Pin – Selects the microcontroller port pin on the SX48 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, 0xB0, 0x1F – Requests 9 byte temperature data from SX48 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.

**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**

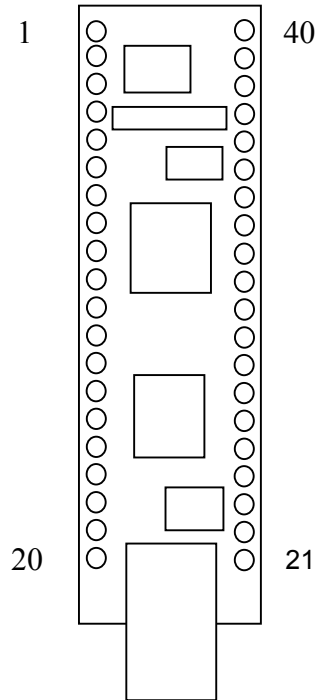
**Parameters:** Data byte – The byte of data to be written to the 8-bit port.

**Returns:** Undefined

**Function:** This function will write the specified data byte to the 8-bit port. Data is latched on the port pins until changed by another command. The port is automatically configured as an output port.

**Example:** 0x2, 0x5C, 0x12, 0xBF – Writes the data byte 0x12 to the 8-bit data bus Port D.

**TABLE 1: DLP-245SY-G PINOUT DESCRIPTION**

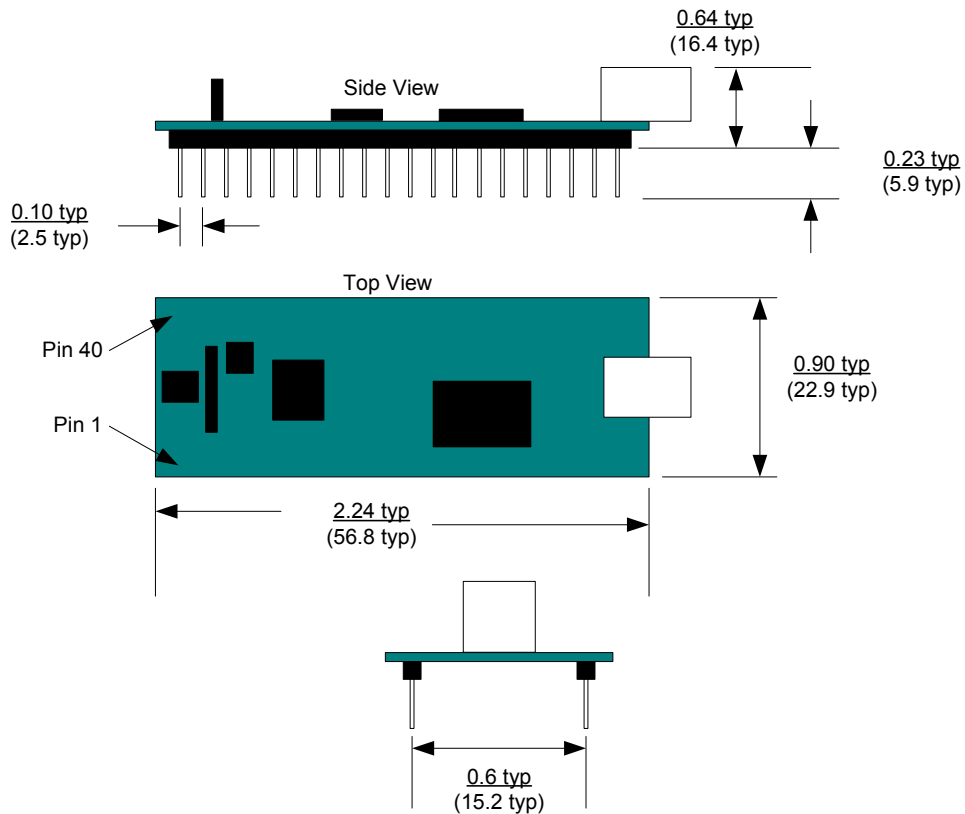


Pin #	Description
1	<b>C0</b> (I/O) Port Pin C0 connected to the SX48 microcontroller
2	<b>C1</b> (I/O) Port Pin C1 connected to the SX48 microcontroller
3	<b>C2</b> (I/O) Port Pin C2 connected to the SX48 microcontroller.
4	<b>C3</b> (I/O) Port Pin C3 connected to the SX48 microcontroller.
5	<b>C4</b> (I/O) Port Pin C4 connected to the SX48 microcontroller.
6	<b>C5</b> (I/O) Port Pin C5 connected to the SX48 microcontroller.

7	<b>C6</b> (I/O) Port Pin C6 connected to the SX48 microcontroller.
8	<b>C7</b> (I/O) Port Pin C7 connected to the SX48 microcontroller.
9	<b>MCLR*</b> (In) Can be used by an external device to reset the SX48. Can be left disconnected if not used.
10	<b>RTCC</b> (In) Input to the SX48's Real-Time Clock/Counter
11	<b>RESET#</b> (In) Can be used by an external device to reset the FT245RL. Can be left disconnected if not used.
12	<b>GROUND</b> Note: This pin was RESETO# in the original DLP-245SY
13	<b>GROUND</b>
14	<b>3V3OUT</b> (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.
15	<b>GROUND</b>
16	<b>SWVCC</b> (Out) Power from EXTVCC (Pin 19) controlled via Pin 10 (POWERN#) of the FT245RL and Q1 MOSFET power switch. R5 and C3 control the power-up rate to help limit inrush current.
17	<b>GROUND</b>
18	<b>VCC-IO</b> (In) 3.0 volt to +5.25 volt VCC to the FT245RL'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.
19	<b>EXTVCC</b> (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).
20	<b>PORTVCC</b> (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-245SY-G and target electronics if the USB device is configured for high power.
21	<b>DB7</b> (I/O) Line 7 of the data bus between the SX48 and the FT245RL USB-FIFO.
22	<b>DB6</b> (I/O) Line 6 of the data bus between the SX48 and the FT245RL USB-FIFO.
23	<b>DB5</b> (I/O) Line 5 of the data bus between the SX48 and the FT245RL USB-FIFO.
24	<b>DB4</b> (I/O) Line 4 of the data bus between the SX48 and the FT245RL USB-FIFO.
25	<b>DB3</b> (I/O) Line 3 of the data bus between the SX48 and the FT245RL USB-FIFO.
26	<b>DB2</b> (I/O) Line 2 of the data bus between the SX48 and the FT245RL USB-FIFO.
27	<b>DB1</b> (I/O) Line 1 of the data bus between the SX48 and the FT245RL USB-FIFO.
28	<b>DB0</b> (I/O) Line 0 of the data bus between the SX48 and the FT245RL USB-FIFO.
29	<b>E0</b> (I/O) Port Pin E0 connected to the SX48 microcontroller.
30	<b>E1</b> (I/O) Port Pin E1 connected to the SX48 microcontroller.
31	<b>E2</b> (I/O) Port Pin E2 connected to the SX48 microcontroller.

32	<b>E3</b> (I/O) Port Pin E3 connected to the SX48 microcontroller.
33	<b>E4</b> (I/O) Port Pin E4 connected to the SX48 microcontroller.
34	<b>E5</b> (I/O) Port Pin E5 connected to the SX48 microcontroller.
35	<b>E6</b> (I/O) Port Pin E6 connected to the SX48 microcontroller.
36	<b>E7</b> (I/O) Port Pin E7 connected to the SX48 microcontroller.
37	<b>B3</b> (I/O) Port Pin B3 connected to the SX48 microcontroller.
38	<b>B2</b> (I/O) Port Pin B2 connected to the SX48 microcontroller.
39	<b>B1</b> (I/O) Port Pin B1 connected to the SX48 microcontroller.
40	<b>B0</b> (I/O) Port Pin B0 connected to the SX48 microcontroller.

**MECHANICAL DIMENSIONS IN INCHES (MM) (PRELIMINARY)**



## **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

1605 Roma Lane

Allen, TX 75013

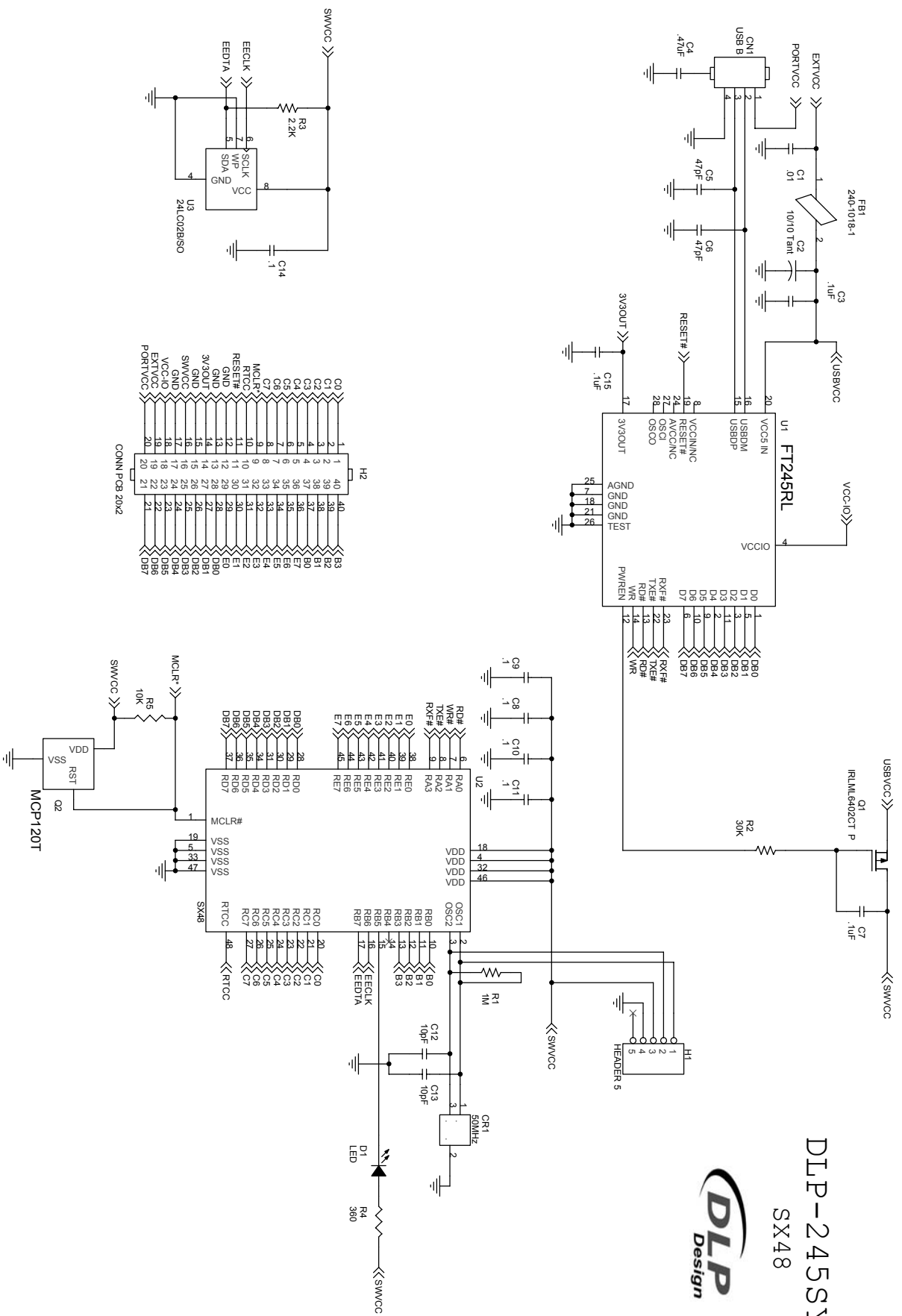
Phone: 469-964-8027

Fax: 415-901-4859

Email: [support@dlpdesign.com](mailto:support@dlpdesign.com)

Internet: <http://www.dlpdesign.com>

# DLP-245SY-G SX48



5 4 3 2 1