

# ⊕ Demand Peripherals

## BaseBoard4

### Peripheral Controller Card



# Table of Contents

Overview .....	3
Installation and Testing .....	4
<i>Installation</i> .....	4
<i>Testing</i> .....	4
What your project might look like .....	6
Appendix A: BaseBoard4 User Constraints File.....	7
Appendix B: BaseBoard4 Call-Out and Schematics.....	9

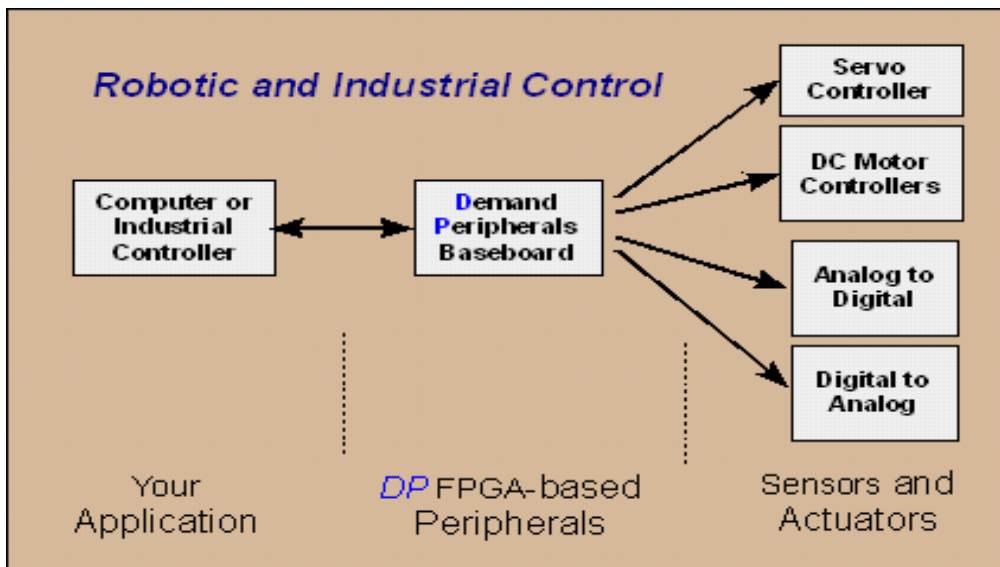
## Overview

The Demand Peripherals BaseBoard4 Peripheral Controller Card features include the following:

- Xilinx 100K Spartan 3E
- USB download of the FPGA code (no JTAG dongle)
- Full-speed USB interface to the host computer
- USB powered (500 ma. maximum)
- Supports both Linux and Windows
- Eight LEDs and 3 pushbuttons
- Three 3-pin headers with power and ground (for servos)
- Four 16-pin expansion connectors each with 8 FPGA pins
- 16-pin connectors can be split into two identical 4 FPGA pin connectors

Besides the Spartan 3E, the BaseBoard4 has a Xilinx CPLD which coordinates the download of code to the FPGA, and an FTDI USB-to-serial adapter. It is the combination of the CPLD and the USB-to-serial adapter that allows the BaseBoard4 to be programmed without use of proprietary drivers or JTAG dongles.

The BaseBoard4 can be a central part of your robot or control system.



# Installation and Testing

## *Installation*

Connect the BaseBoard4 to your host PC using a USB type A to type B cable. Your computer should detect the addition of the FTDI245 USB-to-Serial adapter. If you are using Linux, the appropriate device driver should be loaded automatically for you.

If you are working in a Windows environment you will need to download the proper driver from <http://www.ftdichip.com/Drivers/VCP.htm>. The Virtual COM Port driver exposes the FTDI245 as a COM port. You can locate COM port on your computer by looking in the System folder in the Control panel. More information and an installation guide for the FTDI drivers for Windows and Mac OS X can be found at:

<http://www.ftdichip.com/Documents/InstallGuides.htm>

## *Testing*

This section shows how to validate correct operation of your BaseBoard4 by downloading a pre-compiled Verilog program. The program offers an ASCII command line interface at the USB serial port. The commands in the serial interface allow you to read and write individual FPGA pins. This program should not be confused with DPCore, the FPGA program that contains up to nine user selected peripherals.

The command line interface program is called Pinctrl and is available at the Demand Peripherals web site. Download it from:

<http://www.demandperipherals.com/downloads/pinctrl.bin>

Installing an FPGA program onto a BaseBoard4 is very easy and consists of just pressing the load button near the center of the BaseBoard4 and copying the program binary file to the serial port. So, assuming Windows assigned the new COM port as COM5, the DOS Shell command to install Pinctrl would be:

```
copy pinctrl.bin /B COM5: /B
```

The /B is required to make sure the copy process does a byte for byte copy without any tab expansion or CRLF modifications.

The Linux commands are equally simple. Assuming the FTDI USB-to-serial port is installed as ttyUSB0, the commands would be:

```
stty --file=/dev/ttyUSB0 -opost  
cat pinctrl.bin > /dev/ttyUSB0
```

The green LED on the BaseBoard4 comes on after a successful download.

The next step after a successful download is to open a terminal emulator to the serial port. Use the HyperTerminal program on XP, and minicom or kermit on Linux. Windows Vista users need to download and install a terminal emulation program such as putty (<http://www.chiark.greenend.org.uk/~sgtatham/putty>).

Open the terminal emulation program to the USB serial port assigned to the BaseBoard4. The Pinctrl command set is fully documented in its user manual on the Demand Peripherals web site. The only command needed to test the BaseBoard4 is the "pin" command.

The pin command sets or gets the value of a pin. When a value of one or zero is specified it sets a pin as an output and sets the state to the value specified. Specifying the value as a question mark set the pin as an input and reads the value of the pin. The response to a pin read replaces the question mark with the value of the pin. All pins are configured as inputs until the first write to a pin. Pin commands have the letter P followed by the pin number, an equal sign and a one, zero, or a question mark. The syntax of the command is:

```
[Pp]<pin#>=[01?]\n
```

Use the following Pinctrl command to light LED0:

```
p32=1
```

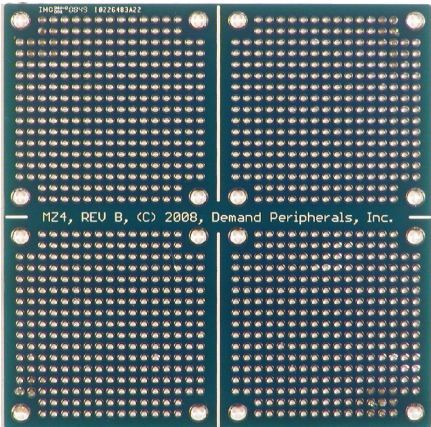
The following command reads the value of button S1. Try this command with the button pressed and with it open.

```
p40=?
```

The Pinctrl program has its own numbering scheme for the FPGA pins and the numbering does not match the physical FPGA pin numbers. The Pinctrl numbers for the LEDs are 32 to 39 and the buttons are from 40 to 42.

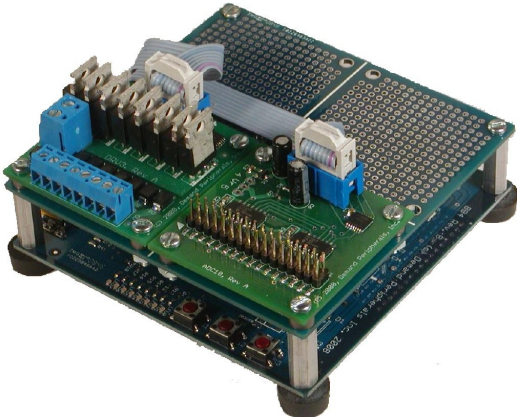
# What your project might look like

Most robotic and industrial control system require some custom circuitry. You might want to consider the Demand Peripherals WireWrap card for your custom circuits. It has four quadrants of plated through holes where you can mount headers, connectors, and components. The card looks like this.



1. WireWrap Card

This card can be mounted with standoffs to the Demand Peripherals BaseBoard4 and to custom cards of your own design. The result might look similar to the following photograph of a stack with the BaseBoard4 on bottom, the WireWrap in the middle, and two daughter card on top. Ribbon cables connect the daughter cards to the BaseBoard4.



2. Stacked card example

## Appendix A: BaseBoard4 User Constraints File

The Xilinx development tools use a User Constraints File (.ucf) to relate the Verilog or VHDL logical name to the physical FPGA pin. The UCF can optionally specify the pin current limits and whether or not to use pull-up or pull-down resistors on the pin.

The User Constraints File for the BaseBoard4 given below. Use this file if you wish to create your own Verilog or VHDL programs for the BaseBoard4.

```
# This is a "user constraints file" that describes
# the pin-out of the Demand Peripherals BaseBoard4.

# Clocks
NET "CK50"      LOC = "P38" ; # 50.0 MHz clock
NET "CK12"      LOC = "P39" ; # 12.5 MHz clock

# Buttons
NET "BNTN1"     LOC = "P13" ; # Button 1
NET "BNTN2"     LOC = "P30" ; # Button 2
NET "BNTN3"     LOC = "P69" ; # Button 3

# LEDs
NET "LED<0>"    LOC = "P70" ; # LED 0
NET "LED<1>"    LOC = "P71" ; # LED 1
NET "LED<2>"    LOC = "P62" ; # LED 2
NET "LED<3>"    LOC = "P66" ; # LED 3
NET "LED<4>"    LOC = "P67" ; # LED 4
NET "LED<5>"    LOC = "P68" ; # LED 5
NET "LED<6>"    LOC = "P63" ; # LED 6
NET "LED<7>"    LOC = "P65" ; # LED 7

# USB interface
NET "RXF_"      LOC = "P89" ; # New data is available
NET "RD_"       LOC = "P43" ; # Active low read data
NET "TXE_"      LOC = "P88" ; # Transmit buffer empty (not)
NET "WR_"       LOC = "P47" ; # Write data on positive edge
NET "USBD<0>"   LOC = "P44" ; # USB Data 0
NET "USBD<1>"   LOC = "P41" ; # USB Data 1
NET "USBD<2>"   LOC = "P40" ; # USB Data 2
NET "USBD<3>"   LOC = "P36" ; # USB Data 3
NET "USBD<4>"   LOC = "P35" ; # USB Data 4
NET "USBD<5>"   LOC = "P34" ; # USB Data 5
NET "USBD<6>"   LOC = "P33" ; # USB Data 6
NET "USBD<7>"   LOC = "P32" ; # USB Data 7

# 3-pin headers
NET "GPIO0"     LOC = "P49" ; # GPIO on 3-pin header #0
NET "GPIO1"     LOC = "P24" ; # GPIO on 3-pin header #1
NET "GPIO2"     LOC = "P26" ; # GPIO on 3-pin header #2

# Expansion headers
NET "JPA<0>"    LOC = "P92" ;
```

```

NET "JPA<1>" LOC = "P78" ;
NET "JPA<2>" LOC = "P79" ;
NET "JPA<3>" LOC = "P94" ;
NET "JPA<4>" LOC = "P98" ;
NET "JPA<5>" LOC = "P83" ;
NET "JPA<6>" LOC = "P84" ;
NET "JPA<7>" LOC = "P85" ;

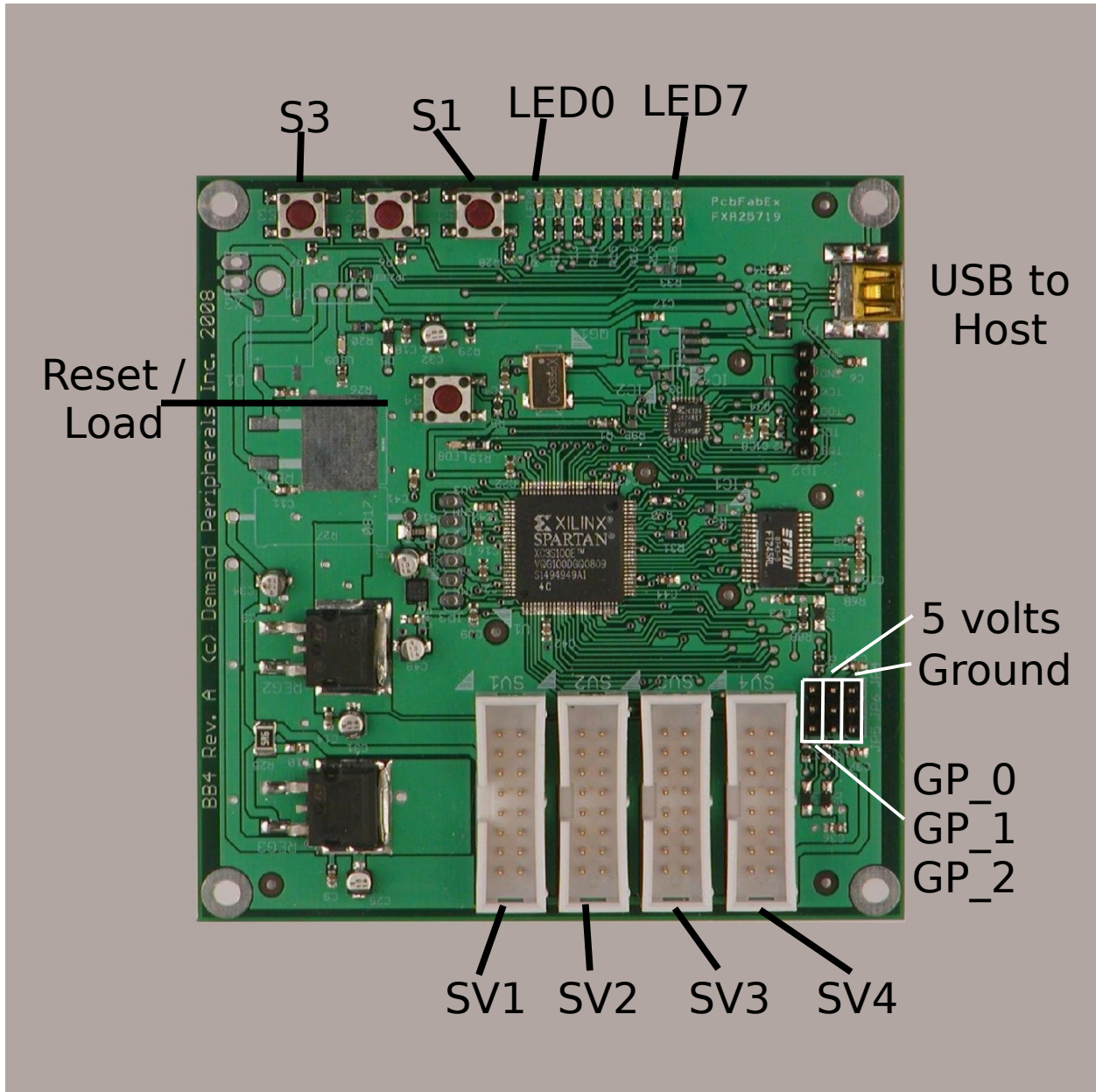
NET "JPB<0>" LOC = "P86" ;
NET "JPB<1>" LOC = "P95" ;
NET "JPB<2>" LOC = "P90" ;
NET "JPB<3>" LOC = "P91" ;
NET "JPB<4>" LOC = "P53" ;
NET "JPB<5>" LOC = "P54" ;
NET "JPB<6>" LOC = "P57" ;
NET "JPB<7>" LOC = "P58" ;

NET "JPC<0>" LOC = "P60" ;
NET "JPC<1>" LOC = "P61" ;
NET "JPC<2>" LOC = "P2" ;
NET "JPC<3>" LOC = "P3" ;
NET "JPC<4>" LOC = "P4" ;
NET "JPC<5>" LOC = "P22" ;
NET "JPC<6>" LOC = "P23" ;
NET "JPC<7>" LOC = "P5" ;

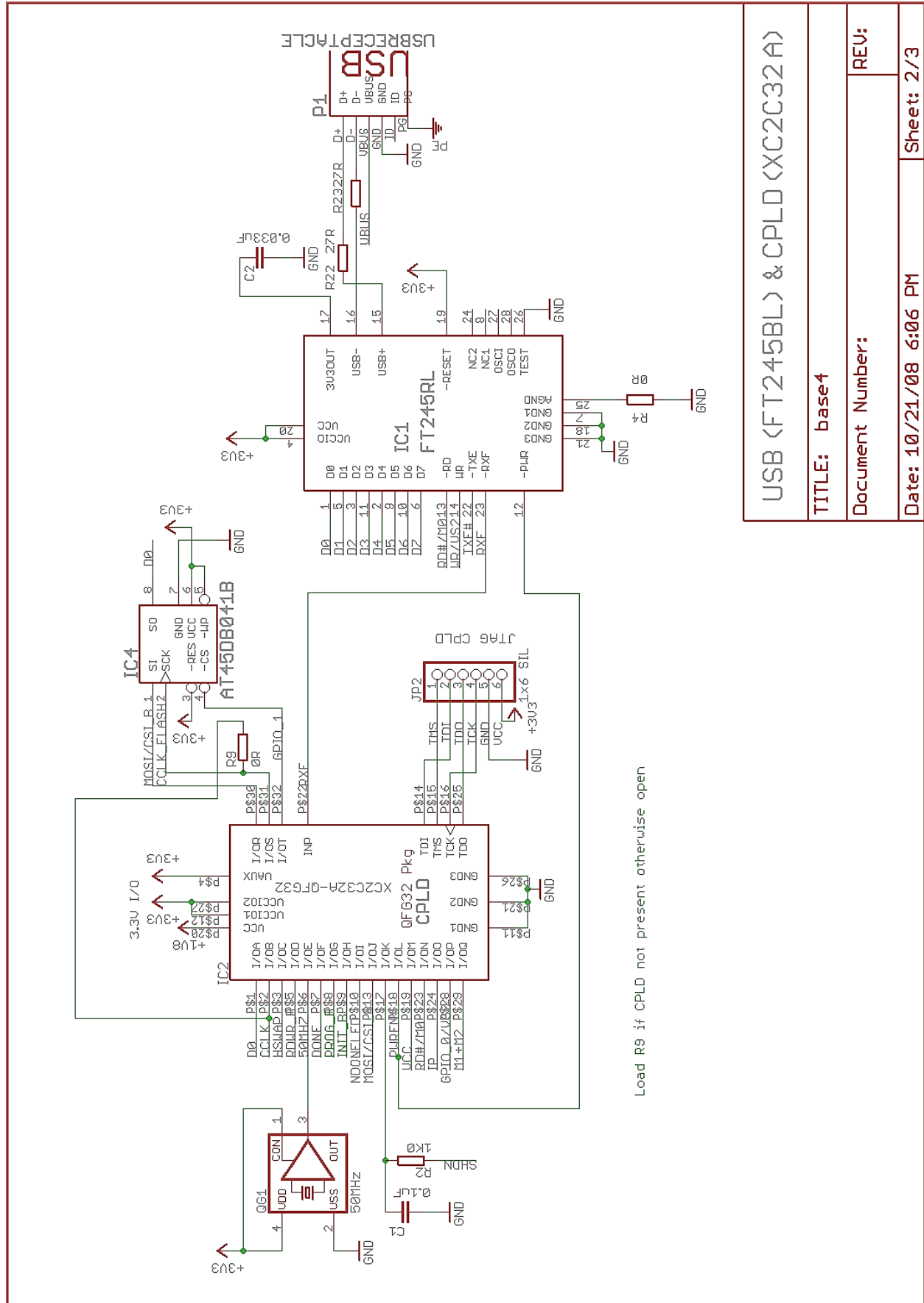
NET "JPD<0>" LOC = "P9" ;
NET "JPD<1>" LOC = "P10" ;
NET "JPD<2>" LOC = "P11" ;
NET "JPD<3>" LOC = "P16" ;
NET "JPD<4>" LOC = "P17" ;
NET "JPD<5>" LOC = "P18" ;
NET "JPD<6>" LOC = "P12" ;
NET "JPD<7>" LOC = "P15" ;

```

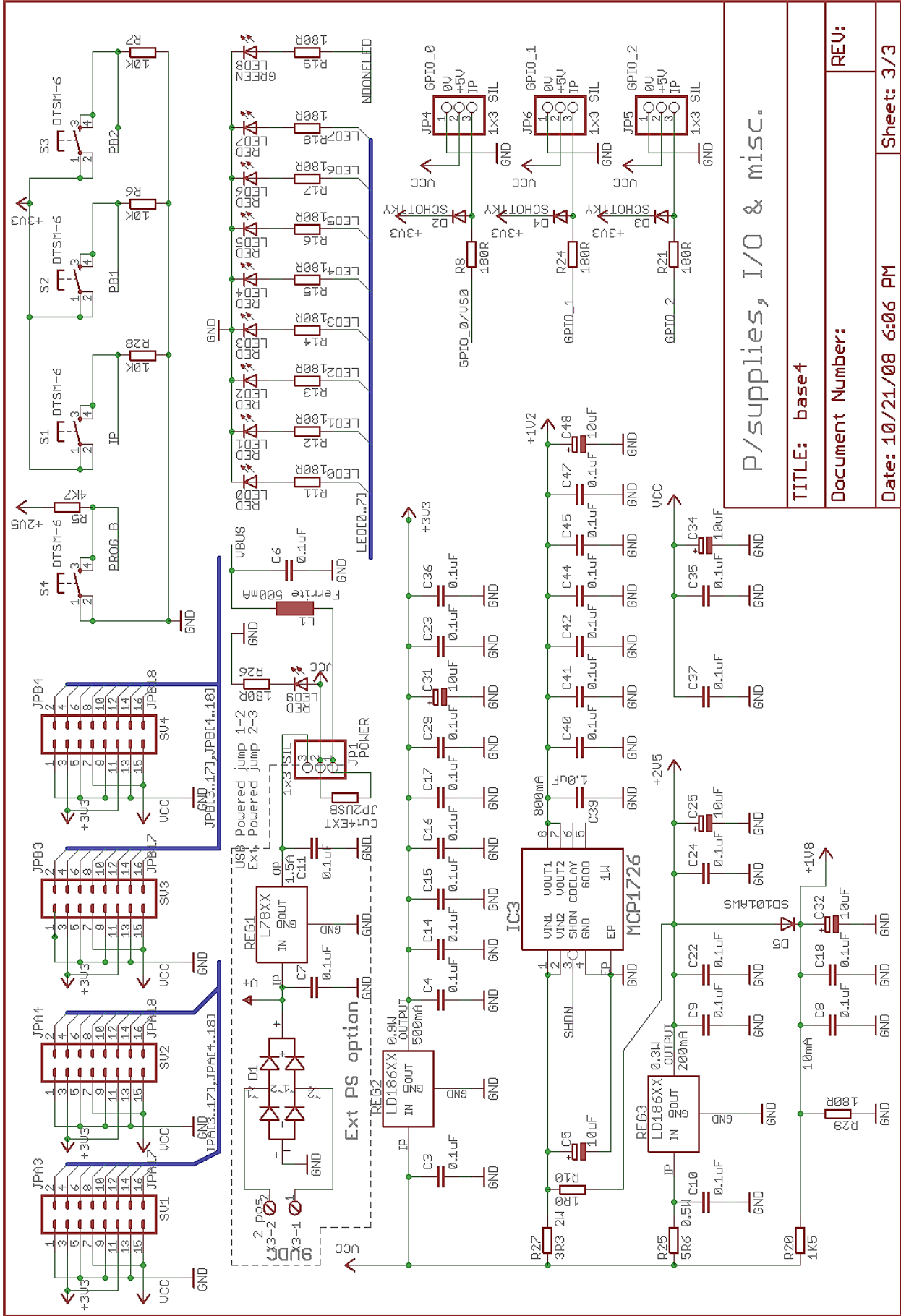
## Appendix B: BaseBoard4 Call-Out and Schematics







USB (FT245BL) & CPLD (XC2C32A)	
TITLE: base4	
Document Number:	REV:
Date: 10/21/08 6:06 PM	Sheet: 2/3



P/supplies, I/O & misc.

TITLE: base4

Document Number:

Date: 10/21/08 6:06 PM

REV:

Sheet: 3/3