

# LED PULSER BOARD

Fermilab  
Terry Kiper

June 27, 2012 (Revision 1.4)

## LED Controller Features

Altera Cyclone 3 FPGA  
Texas Instrument MSP430F2618 micro controller  
USB interface for data, (board power option for USB or 5 volt wall wart)  
Requires USB Device Drivers, see '*LED Controller USB Drivers*'  
4 LED Drive Channels, LEMO connectors to remote "LED BOARDS"

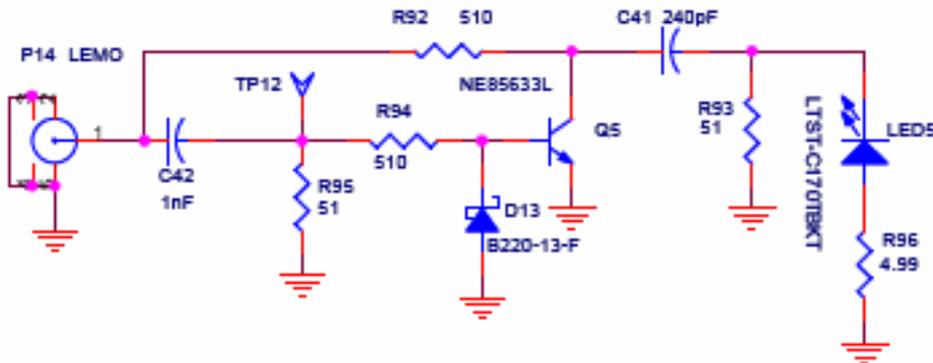
## LED Controller LEMO Connectors

NIM Output P5, Gate Trigger Output  
TTL Input P6, directly pulses LED channels 1, 0  
TTL Input P7, directly pulses LED channels 3, 2  
TTL Output P8, Pulse time matches output at channel 1, 0  
TTL Output P9, Pulse time matches output at channel 3, 2

## LED Drive Circuit Board

Pulsed LED, Lite-On-Inc, Mfg# LTST-C170TBKT  
Source color InGaN Blue Clear 0805 SMD  
Dominant Wavelength 465 nm  
Luminous Intensity 28-180 mcd (IF=20mA)  
Controller drives LEMO outputs with

- DC Offset of positive 0-16.4 volts
- 3 volt pulse



## LED Controller Status LEDs (green)

D7, Local pulse stretched LED Trigger Indicator  
D2, uC heartbeat LED  
D8, uC programmable LED (On when trigger logic enabled)  
D20, +5 Volt power LED

## LED Controller Drive Logic

All channels share the same 3 volt TTL pulse width timing that is adjustable from 10-160nS. Each LED drive channels has a DC level controlled by a 4 channel DAC. This DC voltage level is adjustable at the DAC from 0-4096 mV. The actual DC output level at the LEMO is increased by a factor of 4 by an op amp. A maximum DAC voltage setting of 4096 mV will output approximately 16.4 volts.

## LED Controller Sequence Logic

The FPGA has a 4096x32 bit memory for holding test pulse patterns. Each data word is 32 bits. The upper 28 are compared to a 100MHz counter (10nS/count). If the lower 4 bits are zero it marks the end of the pattern table. Each bit of this lower nibble enables a channel to output a pulse. A pulse pattern can be built by writing these memory locations. Each pattern memory location may be written with a pulse delay of (10nS - 2.684Sec) with 1-4 LEDs active.

## Sequencer Setup Example Pattern (using uC write commands to FPGA registers)

*//writes to WW0 Bits 0-3 control LEDs, LED0=0x1, LED1=0x2, LED2=0x4, LED3=0x8*

```
Cmd 'WB 10 0' //stops sequencer if active
Cmd 'WW 4 0000' //write sequencer address pointer (set to beginning)
Cmd 'WW 0 0000' //delay high
Cmd 'WW 2 1001' //delay low and LED ch0 On (delay is (0x0000100 * 10nS/cnt)=2.56uS)
Cmd 'WW 0 0000' //delay high
Cmd 'WW 2 2002' //delay low and LED ch1 On
Cmd 'WW 0 0000' //delay high
Cmd 'WW 2 3004' //delay low and LED ch2 On
Cmd 'WW 0 0000' //delay high
Cmd 'WW 2 4008' //delay low and LED ch3 On
Cmd 'WW 0 0000' //delay high
Cmd 'WW 2 500F' //delay low and LED ch0-3 On
Cmd 'WW 0 0000' //delay high
Cmd 'WW 2 0000' //delay low (mark end of table)
Cmd 'WB 10 3' //enables sequencer to run data pattern one time (cmd 'WB 10 1' free run).
Cmd 'WB 10 0' //stops sequencer, if still active before next (cmd 'WB 10 0').
```

## LED Controller Menu Help Page 1

HE

### Pulser Control Commands

```
PS - Display Pulser Setup Registers
PA c - ADC Gate Delay, LEMO output, c=0-255(d), 10nS per cnt
PD c - Pulse Delay, c=0-255(d), 10nS per cnt
PE c - Pulser Run Enable, c=0(Off), c=1(single step), c=2(free run)
PG c - Gate (NIM level), LEMO output, c=0-255(d), 10nS per cnt
PW c - Pulse LED on Time, c=0-31(d), 10nS per cnt
PR - Restore Default Memory Timing Pattern and FPGA Control Regs
```

### LED Sequencer Setup

```
SE - LED Sequencer Pattern Setup, ('Enter Key' only for defaults)
```

### LED Single Pulse

```
S1 - LED Single Pulse, outputs one pulse on each LEMO output
```

## DAC (for LED drive voltage, )

WD c d - Write DAC, c=Ch 0-3, Data= 0-4095mV(BCD), (0-16V at LEMO)  
DD - Display DACs (shadowed) set data (BCD), 0-4095mV (BCD)

## FPGA R/W Addr/Data (hex)

RB b e - 8Bit Read, b=BeginAddr, e=EndAddr (Def=BeginAddr)  
WB a d - 8Bit Write a=Addr, d=Data Byte  
RF a c - 8Bit Read a=addr, c=count, addr no increment  
RW a - 16Bit Read a=addr, data  
WW a d - 16Bit Write a=addr, data

## Misc

HE,H1,ID - Help, H1(Addr Map), ID(Code Date/Ver)  
RE - Software restart of uC  
SB n - Set/Flash BAUD, 1=38K, 2=57K, 3=115K, 4=230K  
SN n - Set Serial number (BCD input), use cmd 'SA' to FLASH save.

## Flash

SA - Save Pulser Setup and DAC Regs for Flash  
FL - Load FPGA binary data file (\*.rbf) to Atmel Flash  
FR - Read Flash using last download byte cnt, display SumCheck  
FD d - Display Atmel Flash Page (264 bytes), d= Page Number(dec)

## LED Controller Menu Help Page 2

Help Page 2, FPGA ADDRESS MAP

ADDR	SIZE	NAME
00	(16bits)	Trigger Table High Word (HI+LO= 28bits @ 10nS/Cnt)
02	(16bits)	Trigger Table Low Word (HI+LO=0 marks end of table)
04	(16bits)	Trig Table Pointer (12 bits)
06	(16bits)	Baud Rate Generator (24..15)
08	(16bits)	Baud Rate Generator (16..0), 5.96046 Hz/cnt
0A	(16bits)	Up Time counter (32..24)
0C	(16bits)	Up Time counter (15..0)
0E	(8bits)	ADC Gate (8bits), 10nS/cnt
0F	(8bits)	LED Pulse Delay (8bits), 10nS/cnt
10	(8bits)	CSR, Bit1 (0=Run, 1=Single Step), Bit0(1=Enable)
12	(8bits)	Pulse Width Byte, LED1(Bits7-4), LED0(Bits3-0), 10nS/cnt
13	(8bits)	Pulse Width Byte, LED3(Bits7-4), LED2(Bits3-0), 10nS/cnt
14	(8bits)	Gate Width, R/W Upper 8 bits
15	(16bits)	Test Counter Word, R/W 16 bits

(8Bits use cmds 'RB' or 'WB')  
(16Bits use cmds 'RW' or 'WW')

## Sequencer Menu (example pattern with default setup using command 'SE')

LED outputs are sequenced in two groups, Ch0,Ch1 and Ch2,Ch3  
Seq Pattern, (Ch0,Ch2)On, Delay, (Ch1,Ch3)On, Delay, ALL\_ON, Delay  
LED Lemo DC output level is 4 \* DAC set voltage  
Hit Enter Key on each line for a default sequence

- Enter Initial DAC0 voltage (0-4095mV) 1
- Enter Initial DAC1 voltage (0-4095mV) 1
- Enter DAC0, Step Voltage (0-4095mV) 100
- Enter DAC1, Step Voltage (0-4095mV) 100
- Enter Delay btw LED Pulses (50nS/Cnt) 1
- LED Pulse Patterns per Seq ( 1-1365) 10
- Number of Sequence to repeat( 0-65535) 1000
- LED Sequence Started

## LED Controller USB Drivers

The following are instructions for installing USB drivers on a personal computer. Most of this document is for Windows PC users. **Note: Install (unpack) the drivers before making the first USB connection to the device.**

To get the most recent version of the drivers go to (<http://www.silabs.com>). Search for 'VCP Drivers' or 'CP210X'. Below is an example listing of the driver packages for Mac, Win, and Linux.

**Download the driver package you need. (CP210x USB to UART Bridge VCP Drivers)**

- **Windows 2K (v6.1)**
- **Linux (2.6.x and 2.4.36)**
- **Macintosh OSX**

### Installing the driver on a Windows PC

- Install CP210x driver.
- Now connect the USB port on the PC to the device board.
- New hardware found wizard will open. The wizard should find the drivers and complete the install.
- HyperTerminal can now be used to connect to the device. Connect using the assigned port. This is usually the highest port number that is selectable on the Properties menu under File option for HyperTerminal.
  1. Connect using : usually COM3 (for additional help, see *will not connect to device*)
  2. click on configure button and make the following Port Settings
  3. Bits per second: 115200 (BAUD)
  4. Data bits : 8
  5. Parity : None
  6. Stop bits : 1
  7. Flow control : None

### Will not connect to Device

The following is for Windows users if HyperTerminal will not connect to device.

After installing driver do the following to find the COM port assigned to CP210x USB.

- Note: The Cp210x drivers have been installed.
- Connect DAQ to the PC using a USB cable.
- Right click on 'My Computer" (WinXP example)
- click on Properties
- click Hardware
- click Device Manager
- click Ports (COM & LPT)
- CP210x USB to UART Bridge Controller COM (n). COM (n) is the port number for HyperTerminal.

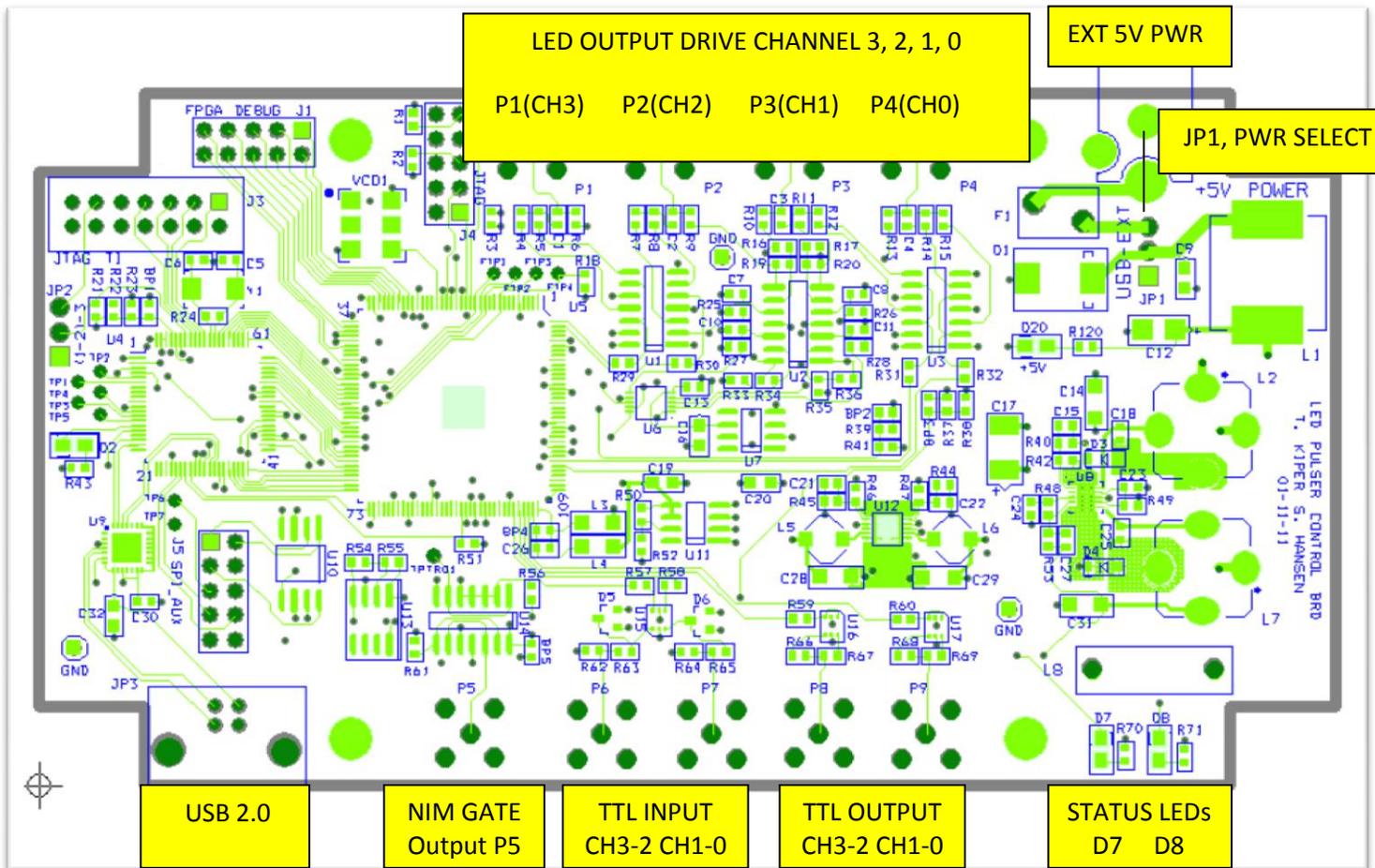
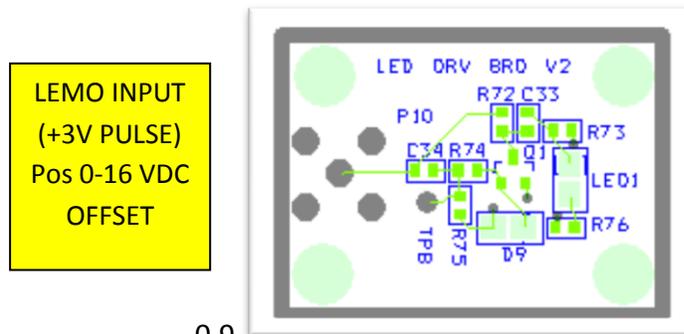


Figure 1, Pulser Card (DIM 5.3x2.84)



0.9

Figure 2, Pulser LED Card (DIM 1.16x0.9)

