LED PULSER BOARD SCRIPT GENERATOR

Fermilab

Thomas Wester

July 9, 2012

Introduction

This program has been developed to interface with the LED pulser board developed by Sten Hansen and Terry Kiper at Fermilab. The board has a 4-channel setup that can be programmed to pulse a LED on any or all of the board's channels at a given time based on a 10ns clock.

Features

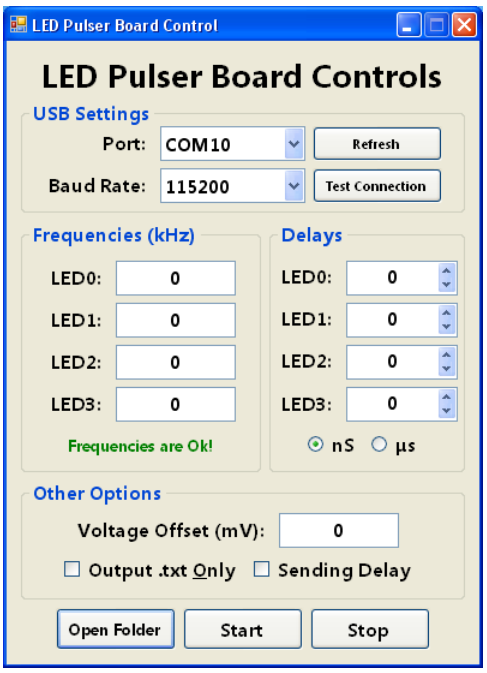
- Easy-to-use GUI interface
- Supports up to 4 LED channels with programmable frequencies and offsets for each
- Built-in voltage control
- Interfaces with the pulser board via USB
- Utilizes Microsoft .NET Framework 4.0

Functionality

The pulser board is capable of sending pulses to each of its four channels at a given time by using a pulse sequence script that contains command-line like entries in a table format. This program aims to simplify the process of writing pulse sequences by generating these scripts based on user-entered values.

The GUI and an Explanation of the Controls

The program will show the following screen on startup:



- Port/Baud Rate – These settings tell the program which USB port to send the files to as well as how fast the program should communicate with the port.
- Refresh/Test Connection – The refresh button refreshes the currently connected USB ports on the computer and adds/removes any entries from the list. The Test connection button sends text to the pulser board and then attempts to read from it.
- Offset – Offsets can be entered either in nS or µS. This can be changed by selecting the appropriate button. Each numeric entry box under the offset heading can be adjusted by the user (with certain guidelines: See Known Issues). **These offsets will move the pulse on the corresponding channel backwards by the entered amount.**
- Frequency Entry Boxes – The user is able to enter a desired frequency in kHz that the LED on the given channel will pulse at.
- Voltage Offset – Any value here will set the starting voltages of all 4 LED channels to the entered amount. The pulser can use values from 0-4095mV.

- Output .txt Only – When the program generates the code for the pulser board to use, it also outputs a .txt file. If the Output .txt Only option is checked, the program will only output the .txt file but will not send the file to the pulser.
- Sending Delay – For the most consistent results, the program can send each line of the generated script to the pulser in short intervals. This is useful because occasionally the board will not be able to read all of the lines at once, especially for longer and more complicated sequences.
- Start Button – Generates a script based on user-entered values and then sends the script to the pulser as well as to a .txt file (The .txt file can be found in the same directory as the .exe file for the program).
- Stop Button – Sends the "s1" command to the pulser which pulses each channel once and then ends any active sequence.
- Open Folder – Opens the current directory that the program is launched from. This is where the .txt file is stored.

Program Details

1. Each frequency value entered by the user is converted to a 10ns time. Then, the program uses a loop that increments a counter while checking if the counter lines up with any of the calculated times. Delays are taken into account by subtracting the delay value from the counter's check.
2. When the counter and the calculated times match, an integer that tells the pulser which channel to pulse is incremented by a certain amount depending on the channel. The values for each channel are in a binary system, so channel 0 has a value of 1, 1 has a value of 2, 2 has a value of 4 and 3 has a value of 8.
3. This information is then outputted as lines that the pulser can read. Both the time and the channel integer value are converted to hexadecimal and separated into two "words" which are read by the pulser as separate lines.
4. The script lines as well as other table marker lines are combined into one file and outputted a text file before going to the pulser.

Known Issues

- Pulses cannot be less than about 50ns away from each other. If two or more frequencies would put pulses too close to each other, the pulser board may not execute the script. User-entered delays must also be 50ns or greater.
- Nanosecond delays must be divisible by 10 (E.g. 100, 120, 130 are acceptable but not 135) for the pulser to recognize them. The program will round to the nearest 10 when set to delay in nanoseconds.
- The pulser board uses a 10ns clock for timing each pulse. Therefore, exact values at a nanosecond level are impossible. The program will round any values to the closest times possible. (For example, a user-entered frequency of 3000 kHz should pulse every 333ns, but the script will actually tell the pulser to pulse at 330ns, 670ns and 1000ns.)
- The program will only run on a Windows computer with the Microsoft .NET Framework v. 4.0 or later installed. The .NET Framework can be downloaded from the Microsoft website here: http://www.microsoft.com/en-us/download/details.aspx?id=17851

References

Kiper, Terry. *LED Pulser Board*. Fermilab. July 15, 2011.

Change Log (7/16/2012)

- New and improved GUI with control grouping
- Auto-correcting frequency and delay entry boxes
- Added the “Open Folder” button
- Added a status bar at the bottom of the form

Change Log (717/2012)

- Changed the focused control at startup to the port combobox
- Color-coded LED channels based on oscilloscope channels

Example Patterns and Oscilloscope View

The following pictures show the program with sample settings and what the user could expect to see on an oscilloscope with these program values entered.