

# Laser Control with the LC880



Many research experiments require the precise control of lasers. The LabSmith LC880 Programmable Experiment Controller handles all aspects of laser timing control, from simple pulsing to sequencing a 2-pulse experiment and beyond.

In this application note, we'll discuss how to implement a pulsed Nd:YAG laser control system using the LC880. We'll expand the system to include safety interlocks and enable switches, in addition to controlling the laser's flash lamp and Q-switch.

## Generating a Laser Pulse Stream

For the first experiment, we will generate a 10 Hz stream of 30  $\mu$ s pulses. According to the manufacturer's specs, the laser will reach maximum population inversion 193  $\mu$ s after the flash lamp fires. Since the Q-switch requires 1  $\mu$ s to engage, we'll trigger it 192  $\mu$ s after the flash lamp. **Figure 1a** shows the timing diagram for this experiment.

We will be using Channel A to provide a 10 Hz Clocked Pulse Stream (i.e., with a 100 ms period) to pulse the laser flash lamp. Channel C will control the Q-switch. The timing modes can be set in dialog boxes (**Figure 2**), and channel logic can be entered using C-style logical assignments.

To help organize the experiment, we'll define Global Variables to represent the outputs of Channels A and C:

`LaserClock = outA;`

`QSwitchOut = outC;`

It is generally good practice to assign memorable variable names and to docu-

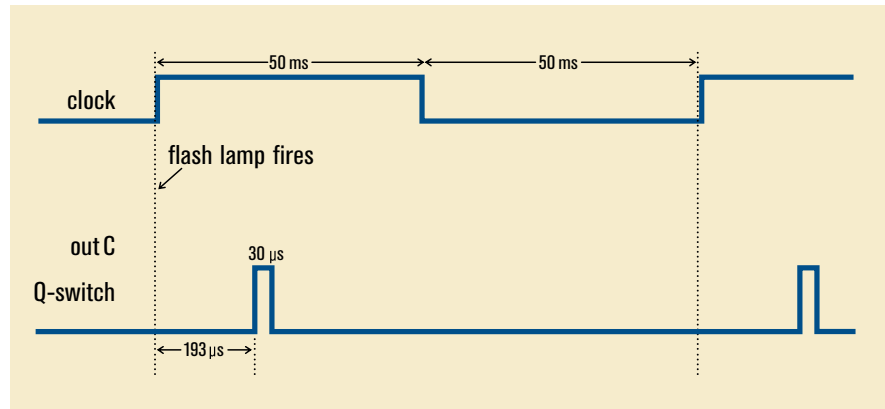


Figure 1a: Timing diagram for generating a 10 Hz laser pulse stream.

Global logic symbols:

`LaserClock = outA; QSwitchOut = outC;`

Channel A:

Free-running clock: High duration: 50.000000 ms; low duration: 50.000000 ms.

Channel C:

Delayed pulse: Delay after trigger 192.000  $\mu$ s then pulse output 30.000  $\mu$ s.

Triggering options: Trigger on rising edge. Unlimited retriggering.

Trigger Input Logic: `inC = LaserClock; //the clock starts the Q-switch delay.`

Figure 1b: Programming for generating a 10 Hz laser pulse stream.

ment your experiments thoroughly to make them easier to understand later.

Channel C will provide the delayed signal to the Q-switch. Operating in Delayed Pulse mode, Channel C (inC) will receive a signal from the laser clock (outA) and will then trigger the Q-switch 192  $\mu$ s later. The LC880 programming for this experiment appears as shown in **Figure 1b**.

## Gating the Control Signal

The manufacturer recommends a 30  $\mu$ s pulse to activate the flash lamp and Q-switch. We can gate the pulse in our first example by employing Channel B, operating in Delayed Pulse Mode. As shown in **Figure 2**, LaserClock (outA) will trigger Channel B, which will then send a 30  $\mu$ s pulse to the flash lamp after minimal delay (< 50 ns). To represent the output to the

flash lamp we'll introduce the variable: `FlashLampOut = outB;`

Channel C will again control the Q-switch. This channel could still be triggered directly by the LaserClock, as in our first

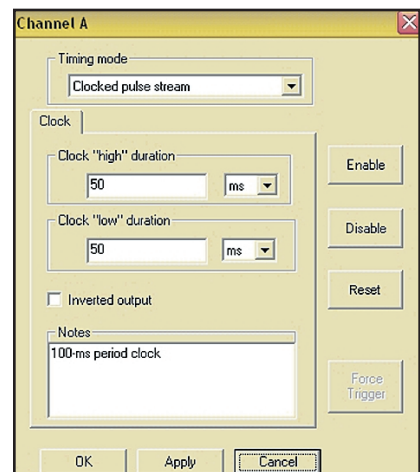


Figure 2: Channel settings can be entered in dialog boxes or through C-style assignments.

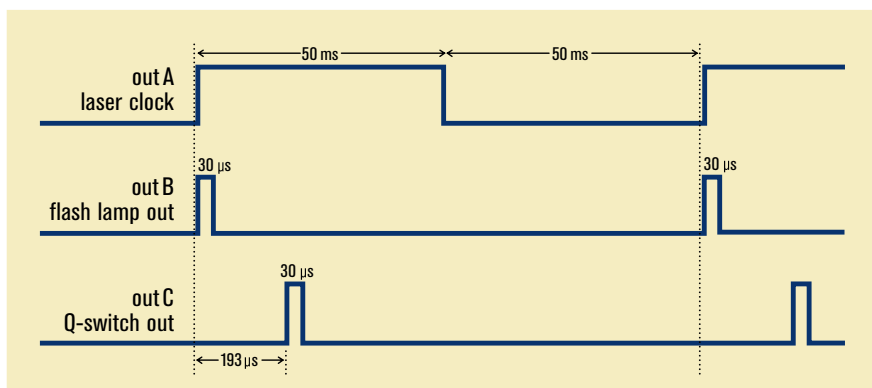


Figure 3a: Gating the control signal with a delayed trigger channel.

#### Global logic symbols:

LaserClock = outA; FlashLampOut = outB; QSwitchOut = outC;

#### Channel A:

Free-running clock: High duration: 50.000000 ms; low duration: 50.000000 ms.

#### Channel B:

Delayed pulse: Delay after trigger 20 ns then pulse output 30.000 μs.  
Triggering options: Trigger on rising edge. Unlimited retriggering.  
Trigger Input Logic: inB = LaserClock; //The clock triggers the flash lamp.

#### Channel C:

Delayed pulse: Delay after trigger 192.000 μs then pulse output 30.000 μs.  
Triggering options: Trigger on rising edge. Unlimited retriggering.  
Trigger Input Logic: inC = FlashLampOut; //The flash lamp triggers the Q-switch.

Figure 3b: Programming for gating the control signal.

```
InterlockOK = not in1;
QSwitchEnable = not in2;
LaserClock = outA;
FlashLampOut = outB;
QSwitchOut = outC;
```

#### Channel A:

Free-running clock: High duration: 50.000000 ms; low duration: 50.000000 ms.

#### Channel B:

Delayed pulse: Delay after trigger 50 ns then pulse output 30.000 μs.  
Triggering options: Trigger on rising edge. Unlimited retriggering.  
Trigger Input Logic: inB = LaserClock and Interlock OK;  
//The clock triggers the flash lamp only if the Interlock is closed.

#### Channel C:

Delayed pulse: Delay after trigger 192.000 μs then pulse output 30.000 μs.  
Triggering options: Trigger on rising edge. Unlimited retriggering.  
Trigger Input Logic: inC = FlashLampOut and QSwitchEnable;  
//The flash lamp triggers the Q-Switch only if the Q-Enable switch is closed.

Figure 3c: Programming for gating the control signal, with interlocks.

example. However, it makes most sense to trigger the Q-switch 192 μs after the flash lamp fires – thus, outA triggers inB, and outB triggers inC. Though the LC880 accuracy is the same in either arrange-

ment, careful attention to the way events are coupled can make programming easier in complicated experiments. The LC880 programming for this example appears as shown in **Figure 3b**.

## Adding a Safety Interlock and Q-switch Enable

Lastly, we'll introduce two switches, a Safety Interlock and a Q-switch Enable, connected to short in1 and in2 of the LC880 to ground when closed. The system will only operate when the Interlock is closed. With the Interlock closed and the Q-switch disabled, the laser can be pulsed to reach thermal equilibrium without emitting any light. Once the system stabilizes, the operator can enable the Q-switch, and the pulse stream will begin.

We'll define these global variables (note: LC880 inputs are normally weakly pulled high internally to simplify the use of switches):

```
InterlockOK = not in1;
//Closing the switch pulls in1 low.
QSwitchEnable = not in2;
//Closing the switch pulls in2 low.
```

We want Channel B to remain disabled until the Interlock is closed. Channel C should be disabled until the Q-enable switch is closed. The logic, then, for Channels B and C is:

```
inB = LaserClock AND InterlockOK;
inC = FlashLampOut AND QSwitchEnable;
```

The entire program appears in **Figure 3c**.

## Conclusion

The LC880's precision timing allows for complex control of lasers and other laboratory devices. Combinatorial logic makes it easy to expand your experiment in an orderly fashion. Whether controlling material removal, PIV, or 2-pulse experiments, the LC880 is an indispensable tool for fast laser setup and reliable experimental results.



4659 Las Positas Rd. Ste. C ■ Livermore, CA 94551  
Phone (925) 292-5161 ■ Fax (925) 454-9487  
www.labsmith.com ■ info@labsmith.com