QSD Monitoring Network

Measuring LaserMet Shutter States

T. Barrett

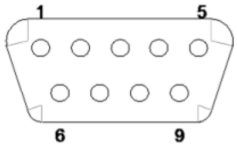
July 2018

Measuring LaserMet Shutter States

At the output of each of the 20 m seed fibres around the labs, as well as in front of every laser, is an *LS10-12 LaserMet* safety laser beam shutter, capable of blocking several watts of optical power. These shutters are connected to a safety interlock system, and when the interlock goes into a tripped state (for example, when lab door is left open for too long) the shutter power is removed and the flag falls down under gravity, blocking the beam. It is useful to be able to monitor the states of these shutters around the building, to know exactly where there is laser light present in the line at any one time.

a)

Pin	Function
1	+12 to 24V DC power to shutter
2	0V
3	Remote open input
4	Not Used (Internally connected to pin 1)
5	'Open' status output
6	'Closed' status output
7 - 9	Optional electrical Interlock option



b)

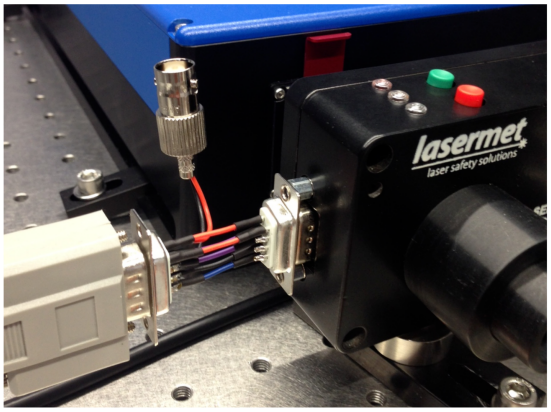


FIGURE 1: Remote monitoring of the LaserMet shutter states. a) Pin allocations for the 9-pin subD connector. b) An in-line adapter to connect the 'Open status output' pin to an external device via BNC cable.

The shutter is mounted on the optical bench and interfaced with a 9-pin subD cable, as can be seen in Fig. 1 b). Page 3 of the *LS10-12* user manual shows that there is a 'Open status output' on pin 5 of the subD, with the pin allocations given in Fig. 1 a). When the shutter is opened (which is indicated by the red light on top of the shutter switching on), then the power supply of the system, which is +22 V in this case, is sent to pin 5. This signal allows remote monitoring of the shutter state.

In order to get access to this 'open status output' pin, we made some adapters which go in-line with the shutter, connecting the centre pin of a BNC cable to pin 5, and the shield of a BNC to ground on pin 2. The adapter is shown after installation in Fig. 1 b). When the shutter is open, the centre pin of the BNC will get +22 V with respect to the outer shield. This allows access to the 'open' indicator signal, whilst avoiding having to cut into the existing cable, and the adapter can simply be plugged in.

One problem is that the shutter 'open status' pin receives +22 V, which is too high for the Arduino digital input pins, which are tolerant only up to +5 V on most models (and only to +3.3 V on the *Due*). Therefore, to convert the higher voltages into lower ones, an optocoupler circuit was designed, and is shown schematically in Fig. 2 a). The optocoupler used is the *Vishay VO2630*, which is a dual channel version - allowing two shutters to be interfaced with the same circuit. The datasheet states that a minimum of 5 mA is needed to turn on the input side LEDs, and so a 4 kΩ resistor is placed in series with the input from the shutter, to allow

this current to flow. The output sides of the optocoupler then connect to two spare digital input pins on the Arduino microcontroller. The truth table in the *VO2630* datasheet (page 3) shows that there is an inversion of the logic - i.e. when the input is high, the output is pulled low. Therefore, in the Arduino sketch, when an input LOW level is detected, this is interpreted as a 'shutter open' signal, and vice versa. The $4\,\mathrm{k}\Omega$ pull-up resistors on the output side ensure that the digital input pins remain HIGH when the shutters are closed and the optocoupler LEDs are off. This HIGH level voltage is now only $+5\,\mathrm{V}$, which is safe to pass to the microcontroller digital input pins. The whole circuit was made into a compact PCB, with two independent channels, a 3D view of which is shown in Fig. 2 b).

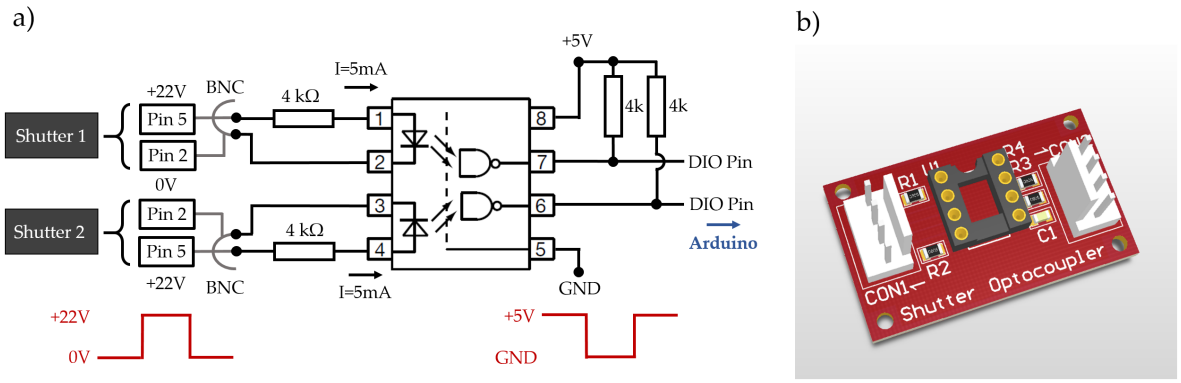


FIGURE 2: a) Schematic of the 'Shutter Optocoupler' circuit. The circuit takes the +22 V signal output from the LaserMet shutters when they are open, and converts it down to +5 V, which can be safely read by an Arduino microcontroller's digital input pins. b) A 3D view of the small PCB, made to facilitate two independent conversion channels, for two shutters in the lab.

Arduino Code

```
/*   04_LaserMetShutterReader
     Code for reading in the states of two LaserMet shutters

*/

const int Pin_Shutter_1 = 2;
const int Pin_Shutter_2 = 3;
int ShutterState_1 = 0;
int ShutterState_2 = 0;

void setup() {

  Serial.begin(9600);
  pinMode(Pin_Shutter_1, INPUT);
  pinMode(Pin_Shutter_2, INPUT);

}

void loop() {

  if (digitalRead(Pin_Shutter_1) == HIGH) {
    ShutterState_1 = 0;
  } else {
    ShutterState_1 = 1;
  }

  if (digitalRead(Pin_Shutter_2) == HIGH) {
    ShutterState_2 = 0;
  } else {
    ShutterState_2 = 1;
  }

  Serial.print("Shutter 1 State = ")
  Serial.print(ShutterState_1)
  Serial.print(", Shutter 2 State = ")
  Serial.println(ShutterState_2)

  delay(1000)

}
```