Project name: Automatic Flash Trigger for High Speed Photography

Author: Nicolae-Cosmin VÂRLAN

Purpose: Even best cameras in the world can hardly achieve timings below 1/8000". This is mainly because the shutter which is (usually) a mechanical device needs to expose for an equal amount of time each light-capturing diode. However, the time needed for light (for example from a flash) to expose a dark room is around 1/50.000.000" (if the room is 6m long). This means that we can trigger a flash in a complete dark room and "freeze" practically everything (for the geeks, an object that falls down, accelerated only by gravity, will move for only 1/500mm). The problem that arises is how do we perfectly sync with the falling object to get him in the right position ?

To synchronize the flash with the falling object, the object will interrupt a laser spot. The event will be reported to Arduino which will trigger the flash. The camera will be pointed to where the object is and be set to make a long-time exposure picture (aprox. 2-3 seconds). However, because the room is dark, the camera will not register anything excepting the "flashed" scene. In order to correctly focus the camera, the lights should be opened, the camera should be focused at the distance where the object will be and set on manual focus (so it won't need to re-focus when you will start shooting in the dark room).

Costs: aprox 15USD. Similar device on e-bay: 120USD.

Requirements: 1 Arduino UNO (or compatible)
- 1 RED Laser emitting diode (the version that emits a single spot)
- 1 photoresistor to sense the laser
- 1 MOC 3042 optocuplor (needed for not discharging high-voltage from some flashes into Arduino board)
- 1 Arduino text Serial screen (in my project, I used ADM1602U from Sparkfun)
- 1 Rotary encoder with push option
- 2 330Ω resistors

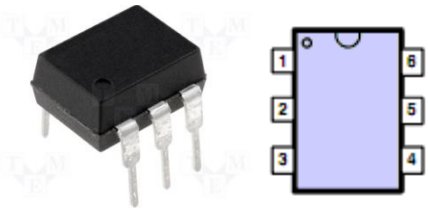
Optionally: Flash + camera + fish tank + some lemons

Description of the project: The lemon will be dropped into the fish tank (which is filled with water). In its way it will interrupt the laser which has the spot on the photoresistor. Missing the light, the photoresistor will report the event to Arduino which, after some time (aprox. 20-30 milliseconds), will light up the led in the optocuplor (you can't actually see the optocuplor light since everything is encapsulated). The optocuplor will close the circuit of the flash which will be triggered. Since the time may vary (different fruits fall down into the fish tank in different ways), you might want to have a way to change that timing - here is where the rotary encoder and the screen will be useful. Also, you can display different stats on the screen.

Connections: Here is how you will connect the hardware:

Arduino pins used: d2, d3, d4, d7, d8, d12, a0, GND, 5V.

The MOC 3042 optocuplor has a small dot near the reference pin. The numbering of its pins will be considered as in the image. Inside there is a LED which will be light up. The anode (+) of this led is PIN1 and the cathode (-) is PIN2. When the led inside is lit, the connection between pins 4 and 6 will be closed.



Connect the PIN1 of MOC to Arduino's pin d12 using a 330 Ω resistor (connect the one side of the resistor to pin 1 and the other side of the resistor to digital 12 of Arduino).

Connect PIN2 of MOC to GND of Arduino.

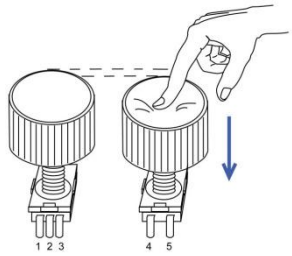
Connect pins 6 and 4 to the flash (it doesn't matter if you are connecting pin 4 or pin 6 to the center pin of the flash). Some flashes also have some input connections that can be used (in the images below you can see such a flash).

The screen has 3 pins: RX, GND, VCC. The *RX pin will be connected to Arduino's d8*, the *GND of the screen to the GND of the Arduino* and the *VCC pin of the screen to the 5V pin* of Arduino.

Te second *resistor should be between the 5V and a0* of Arduino. Also, one side of the *photoresistor should be connected to a0 and the other one to 5V pin of Arduino*.

The laser diode should be connected at 5V and at pin d7 (eventually use a 3rd 330Ω resistor if you want that to be safe).

You also need to connect the rotary encoder to the system. This will be done by using d2, d3, d4 and GND of the Arduino. In the image you have a rotary encoder like the one I used. Both *pins 2 and 4 of the rotary encoder are going to the GND of Arduino*. *Pin1 of rotary encoder is going to Arduino's d2, pin3 of rotary encoder to Arduino's d4 and pin5 of rotary encoder to Arduino's pin d4* - the last one is for the pushing action.

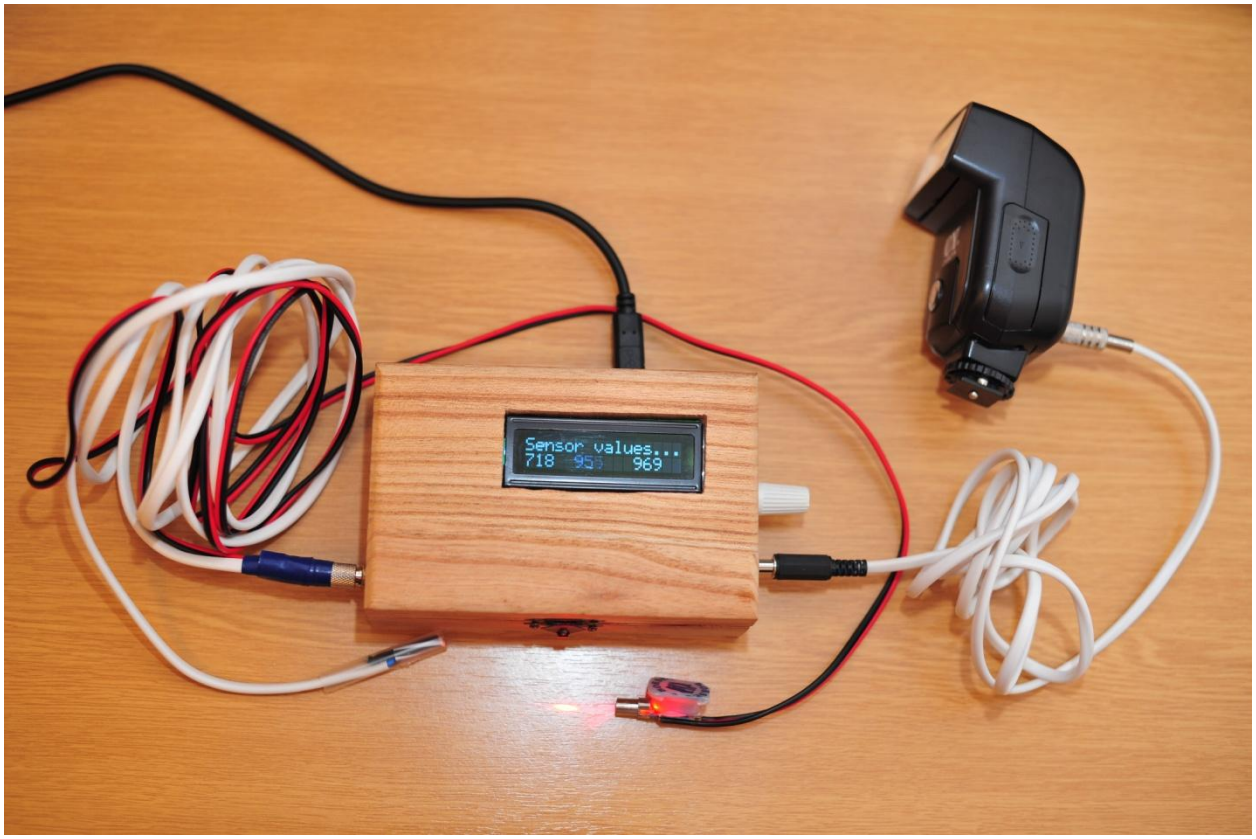


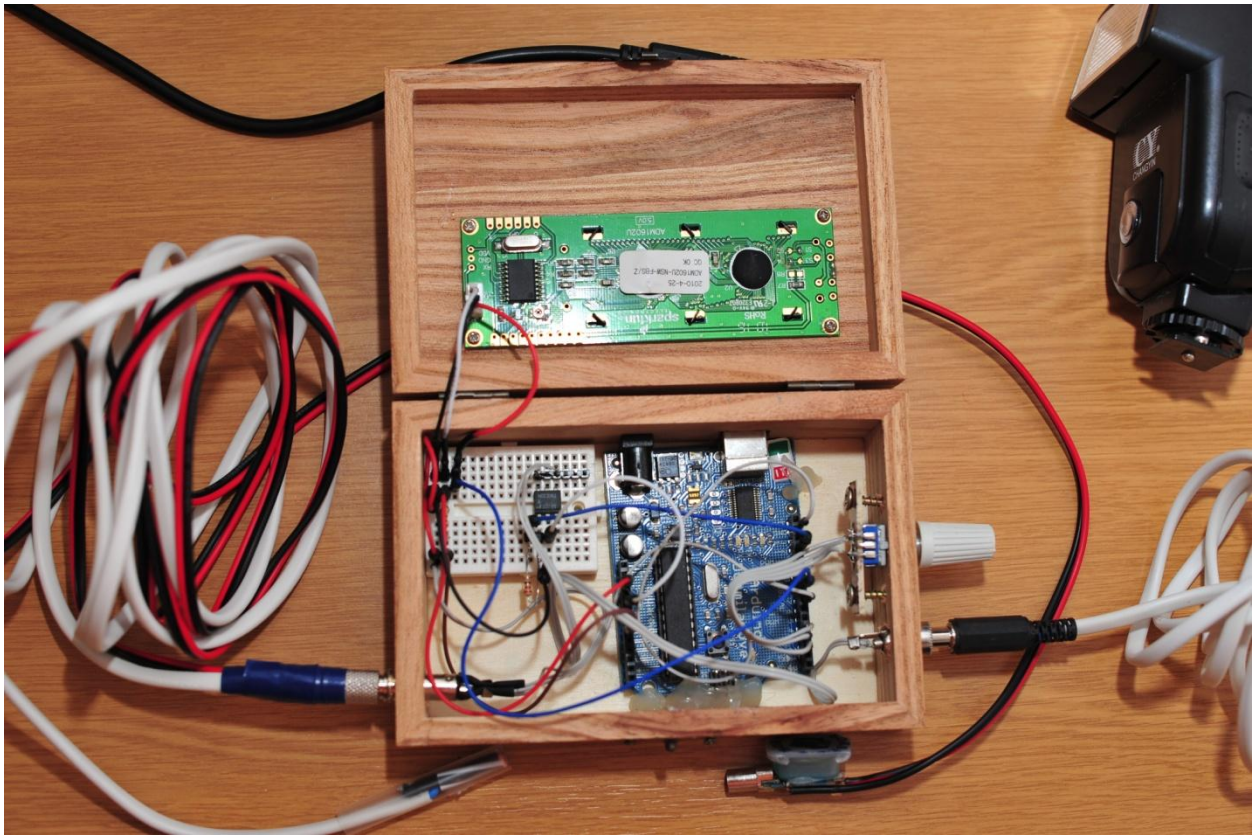
Press the knob to connect 4 and 5

You might want to get longer cables to connect the photoresistor and the laser diode to the Arduino. If you are trying to make photos involving water splashes (water drops flying through the air are nice high speed photography subjects), keep the Arduino and all the electronics as far as possible from the water.

Some photos of the project:

Make your project worth the effort: if you ever think you will use the Arduino board for anything else than the project you just made, probably your project is not that important. So, make it count: put it in a nice box, wire it the best you can, create a nice interface for the user to control the inner settings. In the image below you can see the image displaying the upper and lower bound of the trigger and the current value of the light. If the light will get over or below the thresholds, the flash will be triggered.


Sensor values...
718 953 969



The Coding:

Compile and upload on Arduino the following code. You can modify it as you like (if you decide to add more features to the project).

```
#include <SoftwareSerial.h>

SoftwareSerial mySerial(9,8); // pin 2 = TX, pin 3 = RX (unused)

const int analogInPin = A0;
const int blitz = 12;
const int laser = 7;

// Encoder zone:
const int encoderPin1 = 2;
const int encoderPin2 = 3;
const int encoderSwitchPin = 4; //push button switch
volatile int lastEncoded = 0;
volatile long encoderValue = 0;
long lastencoderValue = 0;
int lastMSB = 0;
int lastLSB = 0;
int sensorValue = 0;
int oldSensorValue = 0;
int lowestSensorValue = 10000;
int highestSensorValue = 0;
int threshold  = 130;// this can be set up
int delayFlash = 20;// this can be set up
int menu = 0;
int option = 1;

void setup()
{
  mySerial.begin(9600); // screen works at 9600 baud
  delay(500); // wait for display to boot up
  pinMode(blitz, OUTPUT);
  pinMode(laser, OUTPUT);

  // encoder:
  pinMode(encoderPin1, INPUT);
  pinMode(encoderPin2, INPUT);
  pinMode(encoderSwitchPin, INPUT);
  digitalWrite(encoderPin1, HIGH); //turn pullup resistor on
  digitalWrite(encoderPin2, HIGH); //turn pullup resistor on
  digitalWrite(encoderSwitchPin, HIGH); //turn pullup resistor on
  attachInterrupt(0, updateEncoder, CHANGE); // intrerupt @ event happens on pin 2 - intrerupt 0
  attachInterrupt(1, updateEncoder, CHANGE); // intrerupt @ event happens on pin 3 - intrerupt 1
}

void loop()
{
    encoderValue=0;
    if (menu==0)
    {
      oldSensorValue=0;
      for(;menu==0;)
      {
        if (abs(oldSensorValue - sensorValue) > 10)
        {
```

```
       goTo(16);
       mySerial.write("                ");
       showNumberAt(sensorValue, 16);
       showNumberAt(threshold, 22);
       showNumberAt(delayFlash, 28);
       oldSensorValue = sensorValue;
    }
    if (abs(encoderValue)>3)
      {
          option++;
          option%=2;
          encoderValue=0;
          oldSensorValue = 0;
          displayMenu();
      }
    if(digitalRead(encoderSwitchPin)){
    }else{
      if (option==0) {menu = 2;  displayMenu(); delay(200);}
      else
      if (option==1) {menu = 1; option = 0; displayMenu(); delay(200);}
    }
    sensorValue = analogRead(analogInPin);
  }
}

if (menu==1)
{
  int refreshWhenTen=0;
  for(;menu==1;)
  {
      int incrementValue = 0;
      //if (encoderValue > 0) {incrementValue=-1; encoderValue=0;}
      //if (encoderValue < 0) {incrementValue=1; encoderValue=0;}
      incrementValue = -encoderValue;
      encoderValue=0;
      if (option==0) threshold += incrementValue;
      if (option==1) delayFlash += incrementValue;
      if(delayFlash < 0) delayFlash=0;
      if(threshold < 0) threshold=0;
      if (incrementValue!=0) displayMenu();
      incrementValue=0;

      if(digitalRead(encoderSwitchPin)){
        // not pressed
      }else{
        if (option==0) {option = 1;  displayMenu(); delay(200);}
        else
        if (option==1) {option = 2;  digitalWrite(laser, HIGH); displayMenu(); delay(200);
                        lowestSensorValue=10000; highestSensorValue=0; }
        else
        if (option==2) {option = 0; menu = 0; digitalWrite(laser, LOW);  displayMenu();
                        delay(200);}
      }
      if(option==2){
        sensorValue = analogRead(analogInPin);
        if (sensorValue < lowestSensorValue) lowestSensorValue = sensorValue;
        if (sensorValue > highestSensorValue) highestSensorValue = sensorValue;
        if ( ++refreshWhenTen == 100 )
        {
           goTo(16); mySerial.write("                ");
           showNumberAt(lowestSensorValue,16);
           showNumberAt(highestSensorValue,27);
```

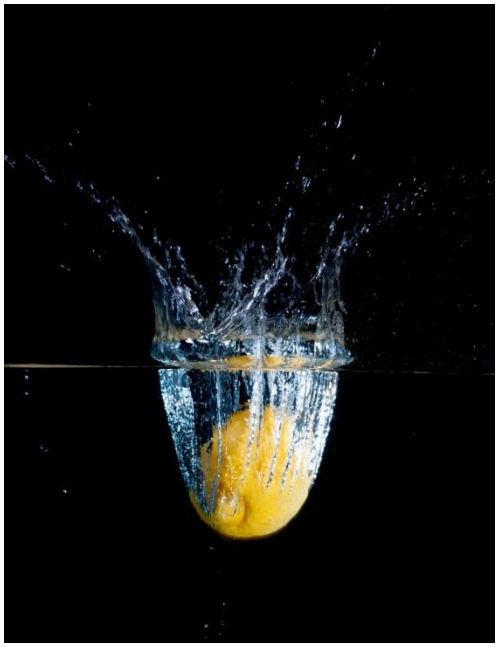
```
              showNumberAt(sensorValue,21);  refreshWhenTen=0;
             }
            oldSensorValue = sensorValue;
          }
      }
    }


    if (menu==2)
    {
      clearLCD();
      digitalWrite(laser, HIGH);
      delay(100); // so the laser has the time to get to the sensor :D
      goTo(16);
      mySerial.write("*Cancel");
      showNumberAt(threshold, 6);
      showNumberAt(delayFlash, 12);
      sensorValue = analogRead(analogInPin);
      oldSensorValue = sensorValue;
      showNumberAt(sensorValue,0);

      for(;menu==2;)
      {
          if(!digitalRead(encoderSwitchPin))
          {
             menu=0; option=1; displayMenu();delay(200);
          }
          sensorValue = analogRead(analogInPin);
          if (abs(oldSensorValue - sensorValue) > 10)
          {
          if(abs(oldSensorValue - sensorValue) > threshold)
            {
              digitalWrite(laser, LOW);
              delay(delayFlash);
              digitalWrite(blitz, HIGH); // start blitz
              delay(1000);
              digitalWrite(blitz, LOW);  // stop blitz
              menu=0; option=1; displayMenu();
            }
            else
            {
              goTo(0);
              mySerial.write("      ");
              showNumberAt(sensorValue,0);
            }
            oldSensorValue = sensorValue;
          }
      }
      digitalWrite(laser, LOW);
    }
}

void displayMenu()
{
  clearLCD();
  if (menu==0)
  {
    if (option==0)
        mySerial.write("*Start  Setup ");
    if (option==1)
        mySerial.write(" Start *Setup");
  }
```

```
  if (menu==1)
  {
    if (option==0)
    {
       goTo(0);
       mySerial.write("*Threshold:");
       showNumberAt(threshold, 12);
       goTo(16);
       mySerial.write(" Delay(ms):");
       showNumberAt(delayFlash, 28);
    }
    if (option==1)
    {
       goTo(0);
       mySerial.write(" Threshold:");
       showNumberAt(threshold, 12);
       goTo(16);
       mySerial.write("*Delay(ms):");
       showNumberAt(delayFlash, 28);
    }
    if(option==2)
    {
          mySerial.write("Sensor values...");
    }
  }
}

// to display a number on the screen at some position:
void showNumberAt(int number, int pos)
{
  goTo(pos);
  int valori[10];
  int nr = number;
  if (number!=0){
    int i;
    for(i=0; nr > 0; i++)
    {
      valori[i] = nr%10;
      nr = nr / 10;
      valori[i]+=48;
    }
    for(int k=1; k<=i; k++)
    {
      goTo(pos + k-1);
      mySerial.write(valori[i-k]);
    }
  }
  else mySerial.write("0");
}

// some commands to deal with the screen display:
void clearLCD(){
   mySerial.write(0xFE);   //command flag
   mySerial.write(0x01);   //clear command.
}

void goTo(int position) { //position = line 1: 0-15, line 2: 16-31, 31+ defaults back to 0
  if (position<16){ mySerial.write(0xFE);   //command flag
              mySerial.write((position+128));    //position
  }else if (position<32){mySerial.write(0xFE);   //command flag
              mySerial.write((position+48+128));    //position
  } else { goTo(0); }
```

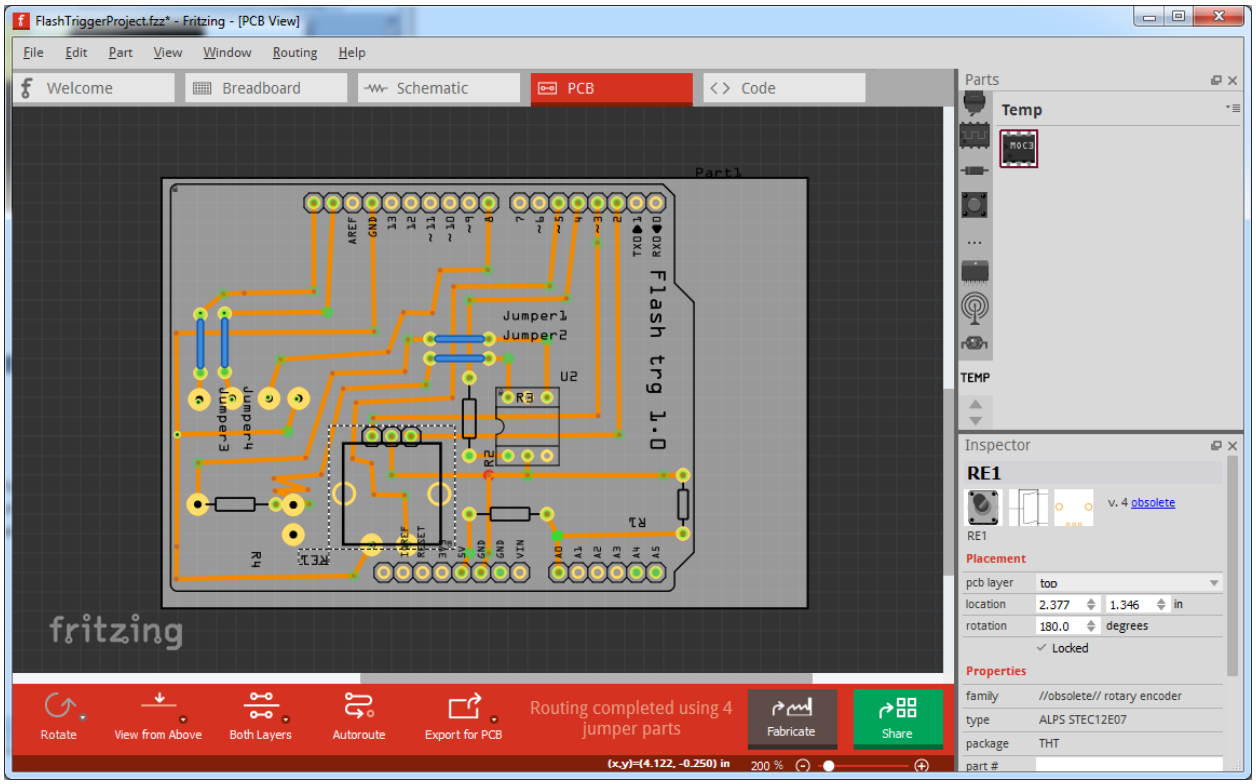
```
}

void backlightOn(){  //turns on the backlight
    mySerial.write(0x7C);   //command flag for backlight stuff
    mySerial.write(157);    //light level.
    delay(10);
}
void backlightOff(){  //turns off the backlight
    mySerial.write(0x7C);   //command flag for backlight stuff
    mySerial.write(128);     //light level for off.
    delay(10);
}

// some commands to deal with the rotary encoder:
void updateEncoder(){
  int MSB = digitalRead(encoderPin1); //MSB = most significant bit
  int LSB = digitalRead(encoderPin2); //LSB = least significant bit
  int encoded = (MSB << 1) |LSB; //converting the 2 pin value to single number
  int sum  = (lastEncoded << 2) | encoded; //adding it to the previous encoded value
  if(sum == 0b1101 || sum == 0b0100 || sum == 0b0010 || sum == 0b1011) encoderValue ++;
  if(sum == 0b1110 || sum == 0b0111 || sum == 0b0001 || sum == 0b1000) encoderValue --;
  lastEncoded = encoded; //store this value for next time
}
```

Miscelaneous: You can attach other sensors to the device. Just modify the thresholds from it's menu and you can set-up easily use the device with sound activation, temperature activation, presence or any other sensor you might think of.

Obvious, you might want to see the result: the photos I took using this device (keep in mind that I am also curious about what you've done). Here there are 3 samples:



PCB version: We also provided a schematic for printing your own shield for Arduino UNO. If you decide to use the following layout, keep in mind that the screen used for this second version of the project communicates via I2C (Via SCL/SDA pins). You will need to rewrite some parts of the previous code to make it work correctly.



The Fritzing source file can be found in the same archive with this file. Fritzing software can be downloaded from http://fritzing.org/home/**.Ş**