

Crossed IR Beam Camera/Flash Trigger

by [tyler_durden](#) on January 4, 2008

Table of Contents

- intro: Crossed IR Beam Camera/Flash Trigger 2
- step 1: Gather Electronic Parts 2
 - File Downloads 3
- step 2: The Program 3
 - File Downloads 3
- step 3: Mechanical Construction 4
- step 4: The Controller 5
- step 5: Using The Photo Trigger 6
- Related Instructables 7
- Advertisements 7
 - Customized Instructable T-shirts 7
- Comments 7

intro: Crossed IR Beam Camera/Flash Trigger

This device will trigger a camera or flash unit to automatically take a picture when an object (target) enters a specific location. It uses two, crossed infrared light beams to detect the presence of the target and close a relay that trips the camera or flash unit. Response time is about 2 ms from detection to relay closure, so if your camera doesn't have long shutter-lag, it will capture even fast moving targets.

The optical part of the device consists of two IR LEDs and two Sharp IS471FE optical ICs (OPICs). The optical ICs have built in LED modulators and synchronous detectors, so they won't see light from each other's LEDs. The outputs from the OPICs are connected to an 8 pin PIC microcontroller that handles interpreting input signals and driving the relay and a visible LED that indicates the operating mode. Though there are 11 operating modes, the controller has a very simple user-interface consisting of a pushbutton switch and an LED.

On power up if the beams are properly aligned and unbroken, the LED lights continuously for 1 second then goes dark to indicate the unit is ready to operate in the continuous mode. In that mode the relay will close and remain closed and the LED will light up as long as both IR beams are interrupted. The unit is now ready to connect to your camera.

With some targets you may want to take more than one picture when the target breaks the IR beams. I have included a basic intervalometer function in the controller to allow cameras that don't have a built-in rapid-fire mode to take multiple pictures as long as the IR beams are interrupted. Pushing the mode select button once takes the controller out of continuous mode and puts it in pulse mode. The LED will flash one time to indicate that the relay will close 1 time per second. Some cameras are faster so pushing the button again will move up to 2 pulses per second. By repeatedly pushing the button, the speed will increase from 1 pps all the way to 10 pps, each time flashing the LED to indicate the pulse frequency. Holding the button down for 2.3 seconds resets the unit and takes you back to continuous mode.

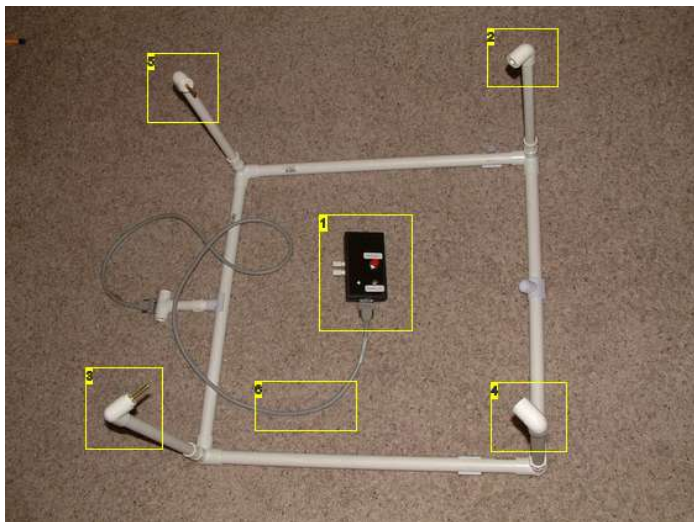


Image Notes

1. Controller
2. IR LED head
3. OPIC head
4. IR LED head
5. OPIC in here
6. 9 pin, straight through, serial cable. Male DB9 on one end, female on the other.

step 1: Gather Electronic Parts

Here are the parts lists for the electronic stuff.

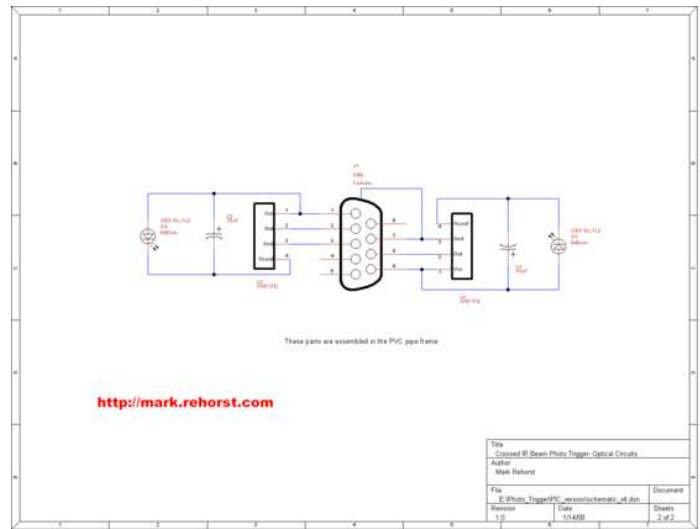
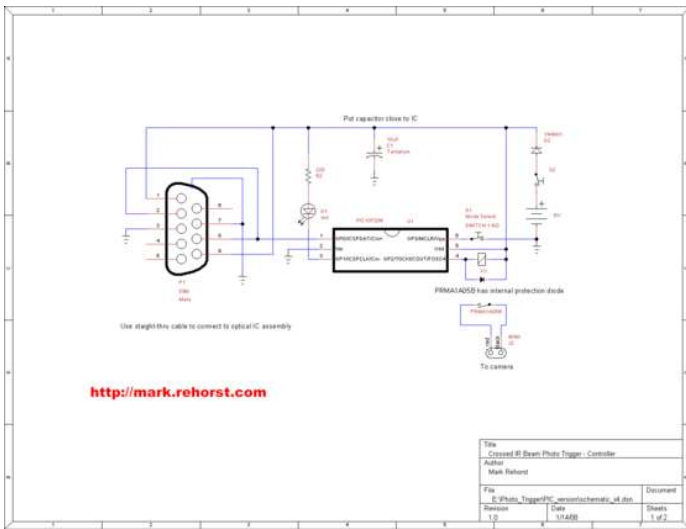
All of the electronics can be obtained from Digikey or other sources. You will need a bunch of different colors of wire, too.

You will need to be able to program the PIC microcontroller- a PICKit2 or ICD-2 or any of hundreds of other programmers can do the job. A suitable programmer will cost about \$20, but once you have it you will find all sorts of projects that can use microcontrollers and will get a lot of use out of it.

When I bought my PICKit2 from digikey I ordered an accessory pack of five PIC10F206 chips with 8 pin DIP adapters. The IC is in a tiny SOT23 package which is fine if you're going to make a PCB but pretty useless for breadboarding and one-off construction projects. The 10F206 is also available in an 8 pin DIP package- I suggest you use it.

I have not provided PCB layout info for the controller here because I didn't use a PCB. The circuit is so simple that it seems sort of silly to make a PCB for it. There are only 4 parts on the board- the relay, the uC, the bypass cap, and a resistor. The circuit requires fewer parts than a 555 timer chip circuit. Just cut some perf board to fit whatever box you're using and wire the thing up. It should take all of 30 minutes start to finish.

The optical circuits are pretty simple- an IC, a cap, and a LED. The LED and optical IC go into diagonally opposite corners of the pipe frame, so you're going to need a bunch of colored wire. I "assembled" the IC and capacitor on small pieces of perf board that fit into cap-plugs for the PVC elbow fittings in the frame- see photos on the next page.



File Downloads



parts_list_v4.txt (867 bytes)

[NOTE: When saving, if you see .tmp as the file ext, rename it to 'parts_list_v4.txt']



schematic_v4.zip (24 KB)

[NOTE: When saving, if you see .tmp as the file ext, rename it to 'schematic_v4.zip']

step 2: The Program

The PIC10F206 is a really simple part- no interrupts and only a 2 level stack, so you can't do any nested subroutines- you'll see liberal use of goto's in the program as a result. The chip is running at 4 MHz using the internal RC oscillator so it executes 1M instructions per second. When an object breaks the IR beams, it takes the IS471 chips about 400 us to change state. From there the uC needs just a few microseconds to detect the change and order the relay to close. The relay takes about 1.5 ms to close resulting in about a 2 ms total delay from beams broken to relay closed.

I developed the program chip using MPLAB. It is Microchip Tech's free assembler/IDE. I also used my Chinese ICD2 clone (about \$50 on ebay) to actually program the IC. I needed to use a lot of delay loops so I rooted around on the web and found a program called PICLoops here:

<http://www.mnsi.net/~boucher/picloops.html>

PICLoops automatically generates timing loop assembly code for you if you tell it what uC you are using and the clock speed. Later on I ran into a similar on-line program here:

<http://www.piclist.com/techref/piclist/codegen/delay.htm>

The second one will generate delays that are accurate to a single clock cycle where PICLoops isn't quite so accurate. Either is fine for this app because timing isn't critical and the uC is running on an RC oscillator anyway.

The program mainly bops back and forth between checking the mode button and checking to see if the beams are interrupted. The mode switch works by keeping a running count of the number of times the button has been pressed. Each time the button is pressed the delay between the pulses to the relay gets shortened enough to step the pulse frequency by 1 Hz. The biggest part of the code is the different delays used by the pulse modes.

When you change the pulse mode the LED flashes to indicate the new mode. You can tell what the new pulse frequency is by counting the LED flashes- 4 times means 4 Hz, etc. The LED flashes have been timed slow enough that you will be able to count them. If the unit is in 10 Hz pulse mode, pushing the button again takes you back to continuous mode.

There is a watch-dog timer that runs while the program runs. If the timer isn't reset before it overflows, the uC will reset itself. That is why holding the mode button for 2.3 seconds causes the uC to reset to continuous mode. When you push the button, the uC waits for you to release it before doing anything. One of the first things it does after you release it is reset the watch-dog timer. If you don't release the button, the watch-dog timer overflows and restarts the program in the continuous mode.

I have attached the assembly listing file for those who are curious and the .hex file for those who just want to burn the chip and be done with it. I welcome any criticism of my programming technique from any of you PIC assembly experts out there.

Note- the relay closes for 25 ms when it operates in pulse mode. Some cameras may require a longer pulse. That delay is set in line that says "call delay25" near the top of the rlypuls section of the code. If 25 ms is too short for your camera, change that line to say "call delay50", then change the line that says "call delay75" to say "call delay50". That will increase the pulse time to 50 ms and still keep all the pulse frequencies at even 1 Hz steps.

The program only occupies 173 bytes out of the available 512 bytes in the chip, so you can add all sorts of functionality to the thing if you desire, though the user-interface is going to be somewhat limiting.

File Downloads



photo_trigger_4.asm (6 KB)

[NOTE: When saving, if you see .tmp as the file ext, rename it to 'photo_trigger_4.asm']



photo_trigger_4.hex (1 KB)

[NOTE: When saving, if you see .tmp as the file ext, rename it to 'photo_trigger_4.hex']

<http://www.instructables.com/id/Crossed-IR-Beam-CameraFlash-Trigger/>

step 3: Mechanical Construction

I initially tried to make this thing with a 3 foot square of 1/2" pipe but found it was nearly impossible to keep the beams aligned. The distance was too great and the pipe too flexible to maintain beam alignment. I switched to 3/4" pipe and a 2 foot square and now it all stays aligned pretty well. I used most of the 1/2" pipe to make marshmallow blow-guns for my son, Alex, and some of his hoodlum friends.

You will need 3/4" pipe for the main frame and 1/2" pipe for the vertical risers that house the optical ICs and LEDs. You can get 3/4" elbows that have a 1/2" threaded side connection, so get some 1/2" thread adapters too.

My philosophy about dealing with PVC pipe projects is to over-buy the fittings and pipe and return what you don't need when the project is done. That minimizes frustrating trips to the store for a single \$0.30 fitting.

You will need a bunch of different colored wire to connect all this stuff- the LEDs and their ICs are separated by about 6 feet of pipe. You will want to make the wires extra long to allow assembly and taking the thing apart for troubleshooting. Different colors will help you keep straight what connects to what.

The first thing I did was drill holes in the caps and mount the LEDs. I attached extra long wires and used heat-shrink at the LED leads to insulate them. I loosely assembled the pipe frame so I could pull it apart easily and ran the wires through the pipe.

Next, mount the IS471 chips and caps on perf board cut to fit into the opening in the end-caps. Drill a hole in the cap and install a piece of 1/4" brass tubing (or whatever you have around). Be sure you know which side of the IS471 is the receiver side! You want it to face your LED, not the bypass cap! Attach wires to the IC board- there will be a total of five connections- Vcc, Gnd, Out, and LED. The fifth wire connects the anode of the LED to Vcc. Decide where you want to put the connector on the pipe frame and make sure the leads to the IC are long enough to reach it.

Mount the connector, run the wires, solder it all together and you're ready to go. Don't forget to solder a ground wire to the shell of the connector. It will help protect everything from static electricity.

Once all the wiring is done, pound the pipe together tightly with a mallet. You should not need glue, and if you glue the pipe together you'll be unable to take it apart to fix problems later. If you want more secure construction, drive a screw through each joint after pounding them together.

When the controller is assembled you will have to align the beams. The relay will close only when BOTH IR beams are interrupted/misaligned. The outputs of the OPICs are normally low, when they can see their light source and go high when the beam is interrupted. So aligning the beams is done as follows:

- 1) Connect the optical frame to the controller.
- 2) Power on. The LED will light and stay lit unless you are extraordinarily lucky. First it lights to indicate continuous mode, then it stays lit because the beams are out of alignment. If the LED goes out it means at least one beam is aligned.
- 3) Assuming that the LED is lit, it indicates that both beams are misaligned. Block one beam with a piece of tape or paper.
- 4) Align the LED as well as you can by twisting the head to point it toward the diagonally opposite OPIC.
- 5) Now start flexing and twisting the OPIC head until the LED goes out, indicating that the beam is aligned.
- 6) Next block the freshly aligned beam, then make the same adjustments to the second beam. When the LED goes out, both beams are aligned and you're ready to take some pictures.

Whenever you power the unit up, check the beams by blocking one then the other. If one beam is misaligned, blocking the other will cause the LED to light. Then you can just realign the one that is out of whack. If the LED lights and stays lit, both beams are out of alignment and you need to follow the procedure detailed above. If you build the thing securely and align the beams for the first time it will take some punishment before you have to do any realignments.



Image Notes

1. IR LED mounted in PVC cap plug.

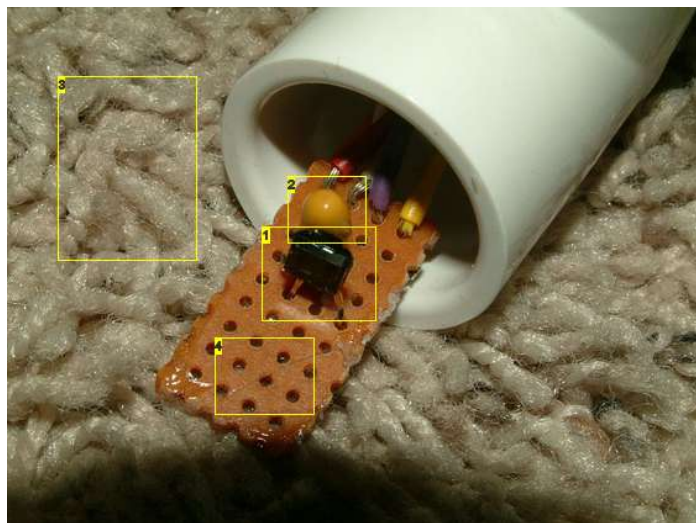


Image Notes

1. IS471FE- make sure it faces the right direction!
2. Bypass cap- I used tantalum, but any electrolytic is OK. Be sure to get the polarity right!
3. Cheesy 70's era shag carpet in my apartment.
4. Extraordinary complexity of the circuit requires specially made circuit board. Or you can just use a piece of perf board...

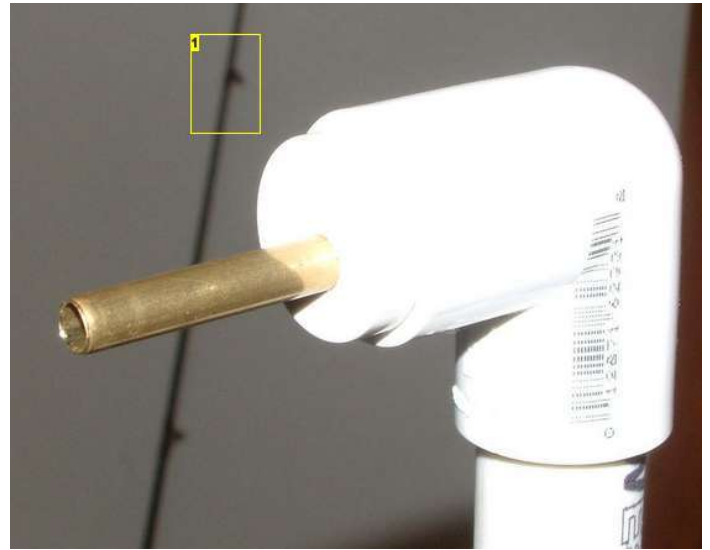


Image Notes
1. Fishing rod...optional!

Image Notes

1. Clean fingers! Always wear gloves to protect your hands when you work with tools. I am in dental school so I wash my hands a LOT, and you should too! And stop biting your nails!

step 4: The Controller

I built the controller in a plastic box I picked up for much too high a price at Fry's electronics. You can use almost anything as long as it is big enough. This box was designed for a 9V battery but I needed to use 6V so the battery space is wasted. I could have easily fit the circuit board in the 9V battery compartment.

Whatever box and switches you use, plan the layout and make sure everything will fit together when you try to close it up.

Note that there is a diode connected in series with the battery. It is there to bring the supply voltage down to an acceptable level for the uC which is rated for 5.5V maximum Vcc. Even with the diode, the part is running at the limit with fresh batteries so don't get any fancy ideas about running at 9V unless you add a 5V regulator. I toyed with the idea of using a PIC12HV615 instead because it has a built-in shunt regulator, but the swing between minimum and maximum currents are too much for the shunt regulator so I'd have to complicate the circuit a bit to get it to work. I wanted to keep this really simple, mostly because I'm lazy but also because I have other projects going and I wanted to finish this one ASAP.

The relay I used has a built-in protection diode shown but not labeled on the schematic. The diode protects the uC from the inductive reverse voltage kick that occurs when you fire a pulse into an inductor like a relay coil. If you use a different relay be sure to add a diode with the polarity shown or maybe you can kiss your uC goodbye the first time the relay fires. The uC can safely sink about 25 mA from one pin so choose a relay with a high resistance coil. The PRMA1A05 has a 500 Ohm coil so it only takes 10-12 mA to close it.

I wanted to use some nice thin, light cables with RJ-11 connectors but all the connectors I found at Fry's were PCB mount parts so I ended up going old-school with DB9s. Serial cables are dirt cheap and the screws will keep the connectors from falling out. You really only need to connect 3 wires (Vcc, Gnd, and the combined outputs of the two IS471FEs) between the optical assembly and the controller so you can use almost any connector/cable you like, even a stereo mini plug and jack.

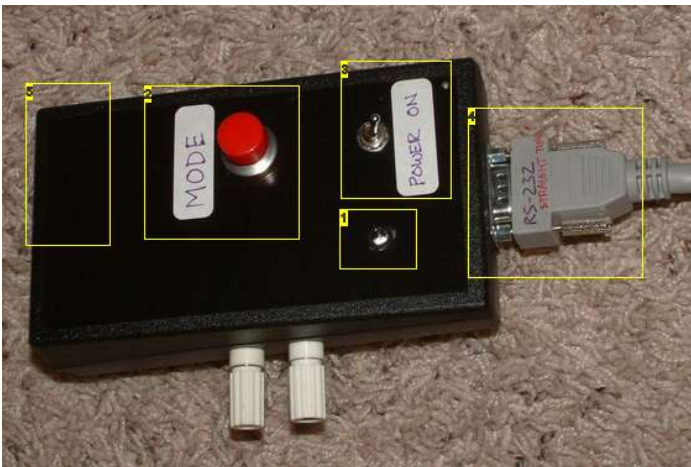
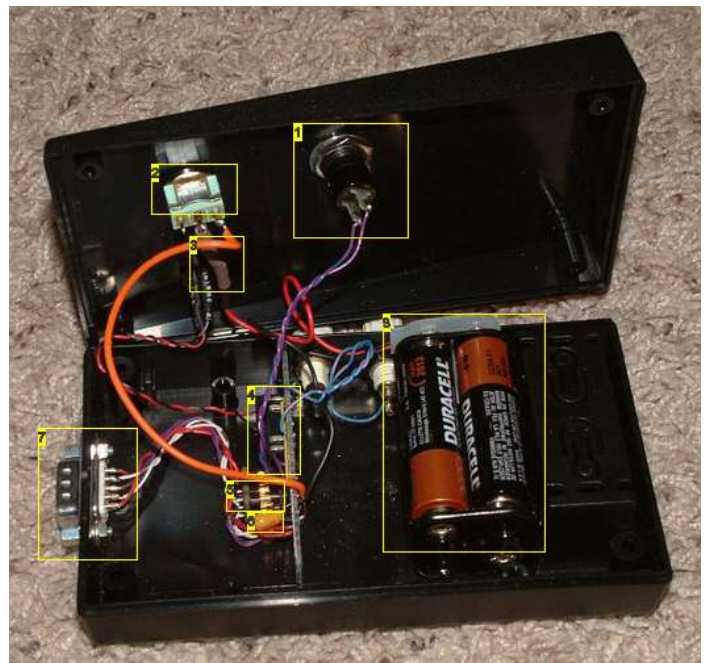
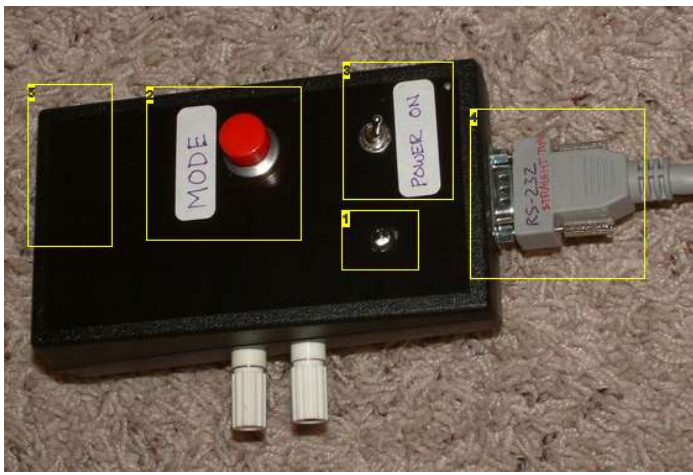


Image Notes

1. LED flashes to indicate operating mode. Also lights when the relay is closed providing visual feedback of system operation.



2. Momentary pushbutton to select operating mode. Press once to go to 1 Hz pulse mode. Press again and again to go to 2, 3,...9, 10 Hz, and back to continuous mode. Hold button down for 2.3 sec. to reset system and go back to continuous mode.
3. Power switch- duh!
4. Standard, cheap, secure straight-thru serial cable to optical frame.
5. Much too expensive, crummy plastic box.

Image Notes

1. Mode select button
2. Power switch
3. Diode inside shrink tubing to drop supply voltage to about 5.5V max.
4. PRMA1A05B relay
5. uC on 8 pin DIP adapter board, plugged into socket.
6. 10uF bypass capacitor.
7. DB9 connector for cable to optical frame.
8. You'll have to guess what these are!

step 5: Using The Photo Trigger

The idea is to set the thing up so that the beams cross where you expect some action to take place. For example, if you want to shoot a hummingbird at a feeder, or a bird entering or exiting a nest, set the frame up with the crossed beam point right where you want it. Then set up the camera pointed at the target and preset the focus, exposure, and white balance (this will minimize the shutter lag time). Test the beam alignment to make sure that BOTH beams are aligned correctly- this is done by waving your hand through each beam individually then through the target area. The LED should light and relay close only when both beams are interrupted. Now set the operating mode- either continuous or pulsed and go away.

You have to know a little about the behavior of your target to get best results. If you want to shoot something that moves fast, you have to take the camera and controller delays into account to predict where the target will be after it interrupts the IR beams. A hummingbird that is hovering in one place can be shot right where the beams cross. A bird or bat that is flying fast might be a couple feet away by the time the camera takes the picture.

The pulsed mode allows cameras that don't have a built in continuous shooting mode to take multiple pictures as long as the beams are interrupted. You can set the pulse frequency as high as 10 Hz, though there aren't many cameras around that can shoot that fast. You'll need to experiment a little to see how fast your camera can shoot.

The camera connection is via a normally open relay contact so you can connect a flash instead of a camera. Then you can shoot in the dark by propping the shutter open and using the controller to fire a flash unit either once or multiple times when an object (a bat, maybe?) breaks the beams. After the flash is tripped, close the shutter. If your flash can keep up you can make some cool multiple exposure shots by using one of the pulse modes.

You can accurately locate the point where the beams cross by attaching some elastic thread to the optical heads. For some targets, that is where you will point and prefocus your camera. The photos below show a Lego man falling through the beams. I dropped him from a couple feet above the beams and you can see he has fallen about 6-8" below the beams in the time it took for the beams to be broken, the relay to close, and the camera to fire. This camera was a Nikon DSLR that probably has small shutter lag when prefocused and exposed. Your results will depend on your camera.

The prototype is now in the hands of the friend who took these pictures (my camera needs to be modified to use remote shutter release). If he produces some more artistic photos using this device I'll try to post them here or on my web site.

Have fun!



Related Instructables



Comments

[16 comments](#) [Add Comment](#)



yoghurtsniffer says:

You should add some hooks on the underside of the pvc so that you can attach a net to catch the item.

Kwl instructable
Good job

thanks

Jul 5, 2008, 12:31 AM [REPLY](#)



tslibertan says:

Excellent.

Last year I made a camera (pulsed laser) trigger for wildlife photography. It had only one beam, and was made out of circuits from Mims' notebooks. 555 ICs etc. It works very well, although it became apparent that a microcontroller would be necessary for anything more sophisticated, and so I've begun to learn about PICs - currently I'm only at the 'flashing LED' stage.

It's with great delight that I see your contraption, which even looks a little like mine (hopefully I uploaded a photo).

May 27, 2008, 6:11 PM [REPLY](#)

I've stolen your .asm file. Those online timing routines sound like they might be very useful. Cheers.



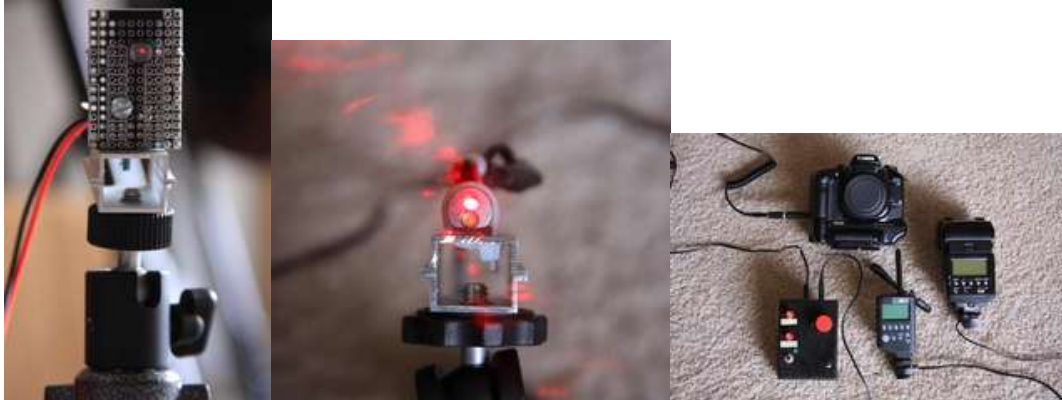
tslibertan says:

May 27, 2008. 6:57 PM [REPLY](#)

In addition to the redness of our knobs, I see that even our confirmation lights and power switches are in the same places.

I feel inspired to design (on paper) my dream all-purpose camera trigger.

Some more pictures I dug up. Detector, laser and some devices the trigger can fire.



viniciuspedrozo says:

May 1, 2008. 7:05 PM [REPLY](#)

You can sale it to me?



clamoring says:

Feb 12, 2008. 2:51 PM [REPLY](#)

Nice use of IR LEDs! Can't wait to see your next project!



GorillazMiko says:

Jan 24, 2008. 3:52 PM [REPLY](#)

Wow... great Instructable! A lot of details, pictures are great, very descriptive, great job, it looks like quite a bit of work, but wow, nice job. +1 rating.



tyler_durden says:

Jan 24, 2008. 9:58 PM [REPLY](#)

Thanks! It was a bit of work, mainly figuring out how get the and keep the optical stuff in alignment. I do a lot of stuff with PVC so I decided to try it for this project too.

The PIC programming was the easiest part. I'm already looking at adding another operating mode - inverse continuous. The idea is to trip the relay when the beams are NOT interrupted. Then you can put an object such as a frog in the beam area, and catch a picture of him in the air when he jumps. Or maybe a water balloon onto which you drop a dart or other sharp object. When the balloon breaks, it will no longer interrupt the beams and the camera will snap the picture.

I think it would be worthwhile to make a smaller optical frame for macro work, maybe one that screws into the tripod socket on the camera. It would probably be great for capturing bugs in the air near flowers, etc.

TD



whiteoakart says:

Feb 3, 2008. 12:00 PM [REPLY](#)

hummingbirds! almost impossible for normal people to reliably capture on film. This would work great for hummingbirds.



coryo93 says:

Jan 31, 2008. 7:02 PM [REPLY](#)

Is it fast enough to be able to capture a picture of a moving arrow, as in a bow & arrow, or like a BB from a BB gun?



tyler_durden says:

Jan 31, 2008. 8:45 PM **REPLY**

That depends on a lot of factors. Arrows and BBs are very small objects. You have to break BOTH beams in order to trigger the camera/flash. It would probably be nearly impossible to do it with something as small as BB in the configuration shown. The beams are IR so there's no easy way to check for alignment that is so precise that an object as small as a BB would trigger the thing. The beams may miss each other by a few mm and that would be enough to prevent a BB from ever breaking both beams. An arrow with feathers would probably work. If you want to capture really small stuff, you will need to make the frame assembly smaller so that you will be able to set up and maintain the beam alignment.

If you want to capture something moving fast, you'll need to trigger a flash instead of triggering a camera- they usually have long shutter lag and the arrow will be far away by the time the camera snaps the picture. If you set everything up in the dark and use the trigger device to fire a flash unit instead, you'll get a nearly instantaneous flash. Set up the camera bow and arrow, flash and whatever else you may need and kill the lights. Open the camera's shutter, fire the arrow, and close the shutter again.

Total delay in the photo trigger device is about 2 ms. You may need a couple test shots to see how far your arrow will go in 2 ms and point your camera and flash there. If the arrow moves at 300 ft/sec, in 2 ms it will go about 0.6 feet. If you rely on the camera's shutter to fire you may have to wait 0.1 seconds- the arrow will go 30 feet.

If I wanted to capture a BB or arrow I would probably set up some other trigger device. Maybe a thin plastic membrane with a piezo pick-up attached. Fire the arrow or BB through the plastic and the pickup will generate an electrical signal that will trip a flash unit.

TD



coryo93 says:

Feb 1, 2008. 10:12 AM **REPLY**

Wow. You covered a lot. Thanks. I will look into what you said. Do you know of any instructable that tells you how to create a slow-motion camera like they use on MythBusters?



tyler_durden says:

Feb 1, 2008. 6:20 PM **REPLY**

That would require a video camera that can shoot at higher than normal frame rates. When you play such a video back it then looks like slow motion. I am looking into modifying web cams for variable frame rate video. If I succeed I will write an Instructable. In the meantime, Casio just announced a new digital camera at CES that will be able to shoot at up to 1200 frames per second. See it here:

<http://gizmodo.com/341634/casios-exilim-pro-ex+f1-hands+on-1200-fps-demo-video-and-sample-60+shots+per+second-gallery>

TD



coryo93 says:

Feb 2, 2008. 6:14 AM **REPLY**

Thanks. Good luck on building that web-cam one. Really good luck to you, because that Canon camera was \$999.99.



tyler_durden says:

Feb 2, 2008. 8:38 AM **REPLY**

None of the currently available web cams will ever be able to shoot that fast. Maybe in ten years...

TD



shaunak says:

Jan 27, 2008. 8:54 AM **REPLY**

Very cool!
Clear instructions, nice illustrations.
Keep the good work up!



johnpost says:

Jan 24, 2008. 9:55 PM **REPLY**

Awesome Instructable, with in focus photos even. Great project. I'm building one.