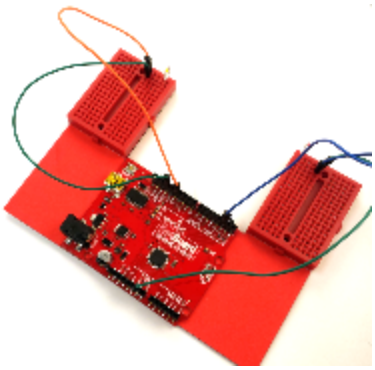
Name: _____ Date: _____

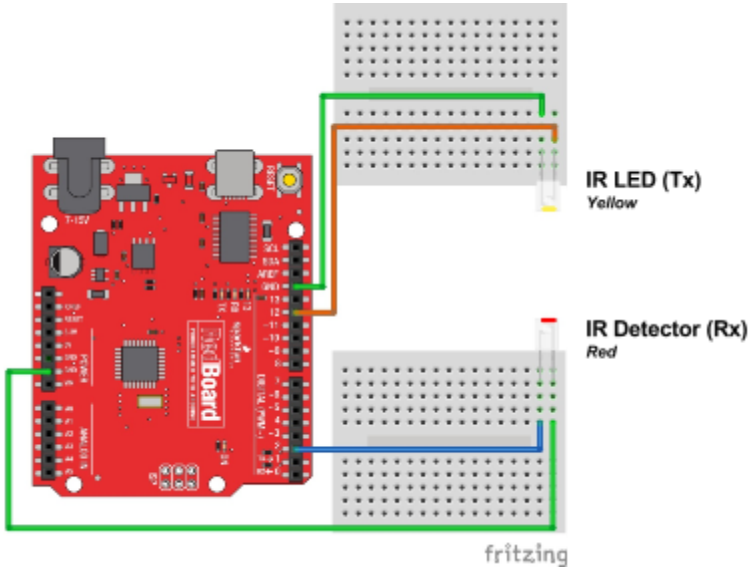
DIY Photogate Timer



Introduction

The photogate is a fundamental tool to have in just about any physics or physical science classroom. A photogate is essentially an IR LED transmit and receive pair separated by some distance - forming a gate. When the IR detector picks up an IR signal, it becomes a short (closing the circuit). We can use this for precise timing applications from characterizing the velocity of a moving buggy to finding the acceleration due to gravity.

Wiring / Hook-up Guide (Fritzing)



Like with the first project, this is a really simple circuit. It involves a single IR LED and an IR Detector. These two components look nearly identical, except for a small colored dot on the top of the element.

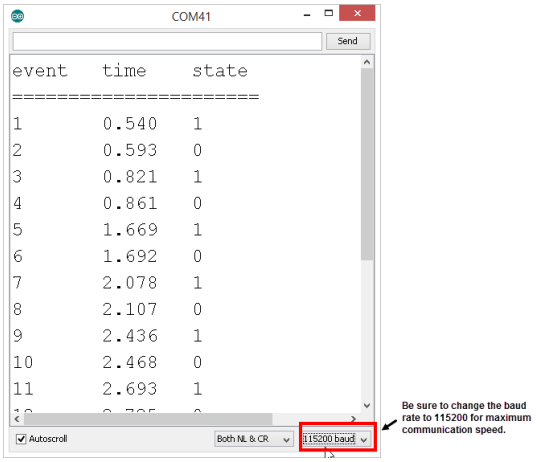
With both of these components, remember that the short leg should be connected to GND. You may also notice a small bulb or plastic lens on the end of each component. Make sure that these are facing each other. Connect the long leg of the IR LED to pin 12 and the long leg of the IR Detector to pin 2.

Example Code

download code: https://codebender.cc/sketch:62316

```
/*****************************************************************
 * DIY PhotoGate Timer
 * Written by: B. Huang, Nov. 17, 2014
 *
 * Hardware connections:
 * IR LED (Emitter): Pin 12
 * IR Detector: Pin 2
 *
 * Upload this example, open up the serial monitor - set to 115200 bps.
 * LED 13 will turn on to show that the beam is being detected.
 *****************************************************************/

// Initialize variables
int IRDetectorPin = 2;  // variable for the detector Pin of the photogate
int IRLEDPin = 12;      // variable for the IR LED Pin of the photogate
char DELIM = '\t';      // data column delimitter character

long baud_rate = 115200;  // use the fastest baud rate poassible.

boolean photoGate;      // stores the state of the photoGate
boolean lastState = 0;      // stores the last State of photoGate used to detect a
change

int eventNum = 1;
long eventTime;          // variable to store the time of the event.

void setup()
{
       Serial.begin(baud_rate);         // setup Serial communication

       pinMode(IRDetectorPin, INPUT_PULLUP);
       pinMode(IRLEDPin, OUTPUT);    // setup IRLEDPin as an output
       pinMode(13, OUTPUT);          // setup debug LED as an output

       digitalWrite(IRLEDPin, HIGH); // turn on the IR LED

       Serial.print("event");
       Serial.print(DELIM);

       Serial.print("time");
       Serial.print(DELIM);

       Serial.println("state");
       Serial.println("======================");
}
// the loop routine runs over and over again forever:
void loop()
{      // read the input pin:
       photoGate = digitalRead(IRDetectorPin);  // Gate is HIGH when IR beam is broken.
       digitalWrite(13, !photoGate); // turns on the LED if the gate is NOT broken

       if (photoGate != lastState)  // if there is a change,
              {
                     eventTime = millis();        // capture the time immediately.
                     Serial.print(eventNum);  // print out the event number
                     Serial.print(DELIM);
                     Serial.print(eventTime / 1000.0, 3); // print out the # of seconds
                                                          // as a floating point decimal
                     Serial.print(DELIM);
                     Serial.println(photoGate); // print out the state of the gate
                     eventNum++; // increment the eventNum
              }
       lastState = photoGate;
}
```

After uploading this code to your Arduino, open up the Serial Monitor. Because we want this to work for any *high speed* timing applications, we want to make sure that the Arduino can keep up. For this, you **must change the baud rate to 115200**. This is the speed that the Arduino sends data back to the computer at.



Now, when you block or un-block the path between the IR LED and the IR detector, you should see an event show up on the Serial Monitor. The "state" indicates if the gate is blocked or not.

You should also see the on-board LED on pin 13 light up when the beam is not blocked. This is a good indicator that the IR LED and detector are properly lined up.

Warning: The IR receiver is susceptible to stray IR from natural light sources. If your classroom has nice windows, you might want to build a baffle or a shield around the IR detector and emitter to help alleviate this.

Labs / Examples / Investigation Ideas

You now have a working photogate timer! Similar units cost upwards of $200 - $300. Great, now what? Here are a few ideas of things to explore:

- **Kinematics**
  - How long does it take for a (ball, square block, index card…) to drop through the gate??
  - Picket fence drop test -- characterize the acceleration due to gravity. (build your own picket fence with electrical tape and clear acrylic).
  - CO_2 race car timing
  - Ball on a ramp
  - Projectile launch velocity
  - SHM -- Pendulum timing
  - SHM -- bouncing mass on a spring.
  - What else???