An Arduino-Controlled Photogate

Calin Galeriu, Becker College, Worcester, MA

It is hard to imagine teaching physics without doing experimental measurements of position as a function of time. These measurements, needed for the determination of velocity and acceleration, are most easily performed with the help of photogates.[1,2] Unfortunately, commercial photogates are rather expensive. Many require the purchase of an additional matching timer or other electronic device. Sometimes special software is also needed. The total bill can easily become prohibitive. For this reason physics teachers have shown considerable interest for cheaper, in-house designed and manufactured photogates.[3-6] The photogate systems described in the literature have their limitations. Some rely on a digital stopwatch and therefore cannot measure time with a precision higher than 0.01 seconds. Some use photoresistors that have a slower response than phototransistors. Some are based on the computer microphone port and therefore cannot handle more than two photogates (two audio channels) at the same time. Extracting the time information from an audio file can also be a challenge for some students. We describe here a photogate system that matches the performance of a commercial one but at a fraction of the cost. The key to this success is the use of an Arduino microcontroller for data collection. The Arduino platform, initially developed for electronics and robotics educational projects, has recently been incorporated in physics labs.[7] The Arduino microcontroller, because of its low cost and because it provides a broad exposure to electronics and computer programming, is an ideal tool for integrated STEM projects.

The photogate housing, shown in Fig. 1, was made with plastic (PVC) water pipe parts, as suggested by Griffin and Stakleff.[8] The housing requires some ½-inch PVC pipe, two ½-inch PVC MIPT plugs, two ½-inch PVC slip x FPT elbows, and two ½-inch slip x slip x slip tees. All the PVC parts add up to $3.72 at Home Depot. From the ½-inch PVC pipe we cut three pieces: two are 3 cm long and one is 13 cm long. A 5-mm diameter hole is drilled in the center of the PVC plugs, and then the LED and the phototransistor are put in place inside. The wires are prevented from moving with the help of two 5-mm tall slices of bottle cork stuck at the ends.

The two photogate electronic circuits,[9] shown in Fig. 2, are based on an infrared LED and a matching phototransistor. The simplest circuit (at the left) uses just one LED, one phototransistor, and two resistors. The phototransistor is in a common-emitter circuit, and it will generate an output signal that transitions from low to high when the path between the LED and the phototransistor is blocked. The more complex circuit (at the right) uses an additional transistor buffer that will ensure optimum noise immunity. This circuit can also drive a load and provide direct TTL compatibility. The phototransistor is in a common-collector circuit, and it will generate an output signal at its emitter that transitions from high to low when the path between the LED and the phototransistor is blocked. The transistor buffer will also act as a logic inverter, and the output signal V_{out} will transition from low to high when the path between the LED and the phototransistor is blocked. Both photogate circuits have produced the same experimental results when used with an Arduino Uno board.

Resistor R1 is 100 Ω, R2 is 5.1 kΩ, R3 is 1 kΩ, and R4 is 47 kΩ. The LED is a TSHA550, a high-speed infrared emitting diode (peak wavelength of 870 nm) manufactured by Vishay Semiconductors (Jameco part number 1950622, $0.39). The phototransistor PT is a UT1893-91-0125, a high-speed and high-sensitive infrared phototransistor (peak sensitivity at 870 nm) manufactured by Ledtech Electronics (Jameco part number 373001, $0.35). The transistor T1 is a PN2222A, an NPN general purpose amplifier manufactured by Fairchild Semiconductor (Jameco part number 178511, $0.06). The voltage V_{cc} of 3.3 V is supplied by the Arduino board, as shown in Fig. 3. If more photogates are needed, then an independent power supply has to be provided, because the Arduino 3.3 V on board regulated voltage supply has a maximum current draw of 50 mA. Another option is to use the 5 V regulated voltage supplied by the Arduino board. The maximum current draw in this case depends on the input voltage source

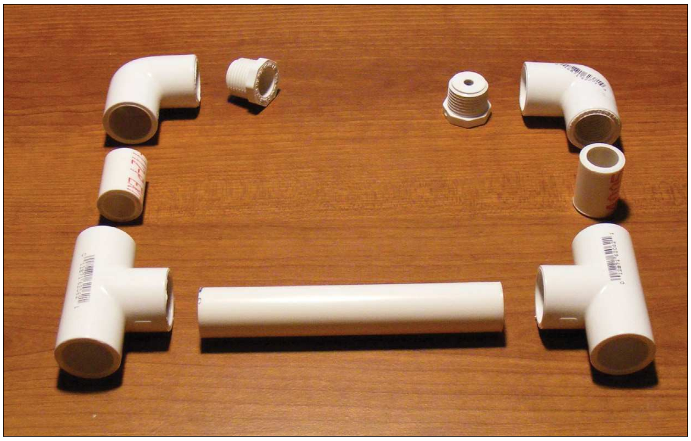


Fig. 1. The photogate housing was made with plastic (PVC) water pipe parts.

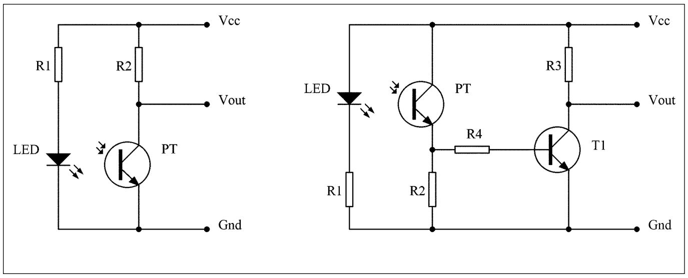


Fig. 2. The two photogate electronic circuits are based on an infrared LED and a matching phototransistor.

DOI: 10.1119/1.4792011

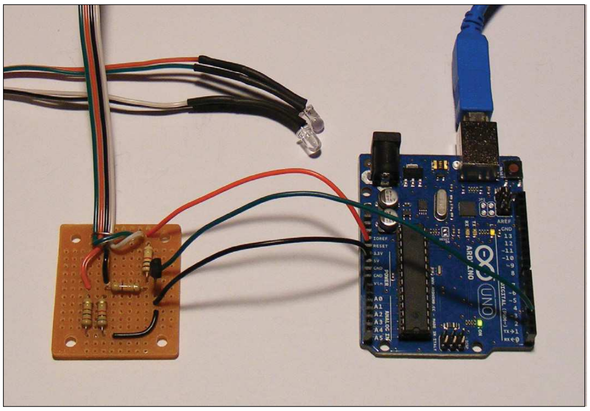


Fig. 3. The photogate circuit and the Arduino board.

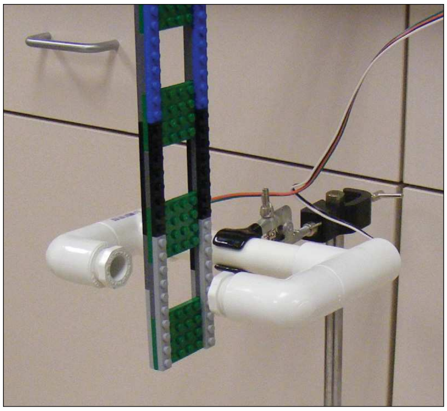
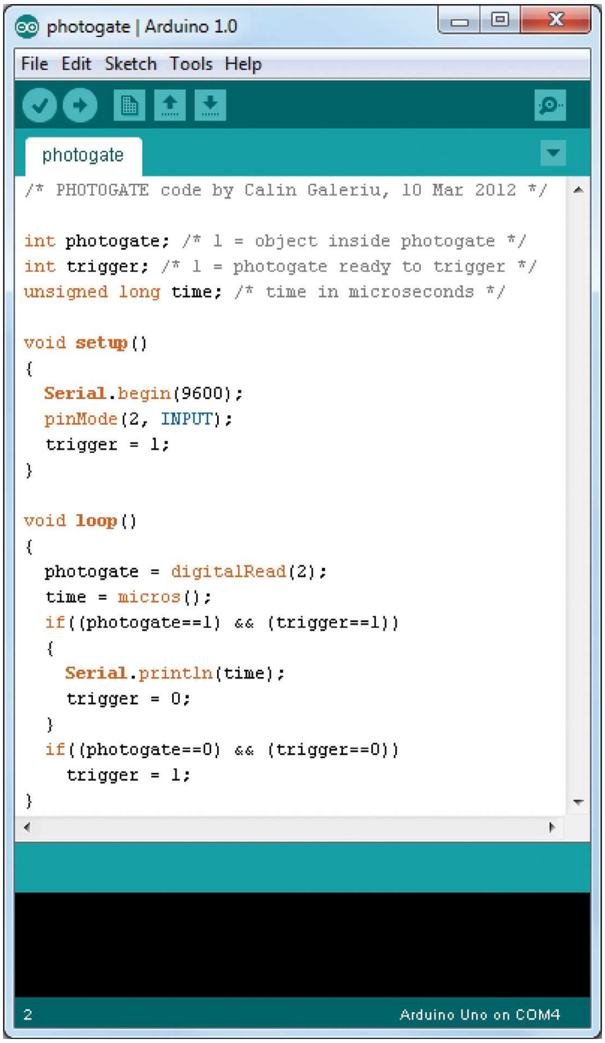


Fig. 4. The "picket fence" was made with Lego bricks.

used (USB port or power jack). The photogate output V_{out} is connected to the digital input pin 2 of the Arduino board. These connections are made with 22-gauge solid hook-up wires. The small perforated circuit board, with copper solder rings on one side, has dimensions of 1.5×1.75 in^2 (All Electronics part number PC-1, $0.75, with matching plastic case part number TB-1, $1.90). The LED and the phototransistor are connected to the circuit board with a 24-gauge solid four-conductor intercom wire from RadioShack. It is a good idea to remind the students soldering the photogate circuit that the shorter leg of the LED is the cathode and is negative, while the shorter leg of the phototransistor is the collector and is positive.

Arduino is an open source electronics prototyping platform, originally designed for hobby or educational use. The hardware and the software are very well documented on the Arduino website (arduino.cc/en/). The Arduino Uno ($29.95 from Jameco) used in our photogate system relies on an ATmega328 microprocessor and has 14 digital input/output pins, six analog inputs, a 16-MHz crystal oscillator, and a USB connection. The Arduino board can draw power from the USB connection, or it can use an ac-to-dc adapter. The microcontroller on board is programmed using the Arduino programming language, which is based on C/C++. Detailed tutorials are posted on the Arduino website, along with many code examples, and even a reference book.[10]

We have used this photogate system for a picket fence experiment.[6,11] In this experiment we drop an object with alternating transparent and opaque stripes, the "picket fence," between the arms of a photogate, measuring the times at which the infrared light beam is blocked. We have built our picket fence out of Lego parts, as shown in Fig. 4, using six 6 x 4 plates, ten 1 x 4 plates, twelve 1 x 8 plates, and eight 1 x 10 plates. All these Lego bricks can be ordered from shop.lego.com/en-US/Pick-A-Brick-ByTheme for a total of $7.50 plus shipping. We have measured the distance between the 6 x 4 plates with a caliper, for maximum accuracy. There are 6.375 cm from the bottom of one 6 x 4 plate to the bottom of the next one.



```
/* PHOTOGATE code by Calin Galeriu, 10 Mar 2012 */

int photogate; /* 1 = object inside photogate */
int trigger; /* 1 = photogate ready to trigger */
unsigned long time; /* time in microseconds */

void setup()
{
  Serial.begin(9600);
  pinMode(2, INPUT);
  trigger = 1;
}

void loop()
{
  photogate = digitalRead(2);
  time = micros();
  if((photogate==1) && (trigger==1))
  {
    Serial.println(time);
    trigger = 0;
  }
  if((photogate==0) && (trigger==0))
    trigger = 1;
}
```

Fig. 5. The Arduino code for the picket fence experiment.

The Arduino will send to the computer, through the USB cable, the six times (in microseconds) at which the infrared beam is blocked by the 6 x 4 Lego plates. The Arduino code listed in Fig. 5 was specifically designed to perform this task. The output voltage from the photogate is fed into the Arduino digital pin number 2, from where it is read and saved into the variable *photogate* every time the loop function is

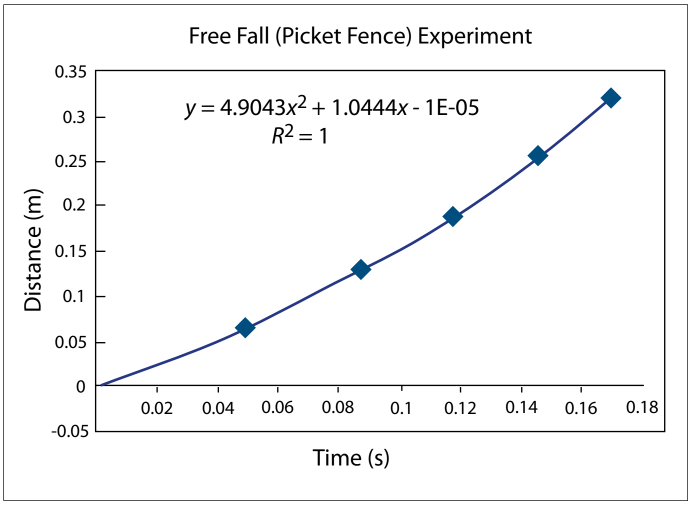


Fig. 6. The graph of distance vs time for an object in free fall.

executed. Accordingly the variable *photogate* is 1 when there is an object in the photogate, between the LED and the phototransistor, blocking the infrared light. With no object in the photogate, the variable *photogate* is 0. For our picket fence experiment we want to capture and record the time when the infrared light is blocked the very first time by each of the 6 x 4 Lego plates. The program uses another variable, named *trigger,* to identify the exact moment when the variable *photogate* changes from 0 to 1. With no object in the photogate, the variable *trigger* is set to 1 and the system is ready to capture the arrival of an object. The very first time an object is detected, the variable *trigger* is set to 0, thus preventing the recording of all of the later times when the object is still in the photogate. After the object gets out of the photogate, the variable *trigger* is again set to 1, and the system is ready to capture the arrival of the next 6 x 4 Lego plate.

After compiling, downloading, and starting the execution of the Arduino code, we need to open the Serial Monitor window. Then we drop the picket fence. It is very important to drop the picket fence as vertically as possible. The Serial Monitor window will show the six times when the infrared beam was blocked. These six values are easily copied (select the numbers with the mouse, then press CTRL+C) and pasted into a Microsoft Excel spreadsheet. Arduino boards with a clock speed of 16 MHz have a time resolution of 4 μs.

Now we can make a graph of distance versus time for our picket fence in free fall. The experimental data are shown in Fig. 6. In column A we have the time in microseconds, as provided by the Arduino. In column B we have the time in microseconds, with the value of the first time measurement subtracted from all the data. In column C we have the time in seconds. In column D we have the distance in centimeters, and in column E we have the distance in meters. A scatter plot of distance versus time shows that the data points are very well fit by a quadratic function. The constant of gravitational acceleration is twice the coefficient of the square term. We have obtained $g = 9.8$ m/s^2, in excellent agreement with the textbook value.

The author wishes to thank the anonymous referee for helpful comments.

References

1. *Instruction Manual and Experiment Guide for the PASCO Scientific Model ME-9206A and ME-9215A Photogate Timers*, wiki.brown.edu/confluence/download/attachments/28887/Photogate+Timers.pdf.
2. John Wolbeck, “Instantaneous velocity using photogate timers,” *Phys. Teach.* **48**, 262–263 (April 2010).
3. Allen Wootton, “A stopwatch-based photogate timer,” *Phys. Teach.* **38**, 405–407 (Oct. 2000).
4. Michael Horton, “A $400 photogate for $50 or less,” *Phys. Teach.* **46**, 246–247 (April 2008).
5. Michael Horton, “The $10 photogate,” *Phys. Teach.* **48**, 615 (Dec. 2010).
6. Zoltan Gingl, “Sub-$10 sound card photogate variants,” *Phys. Teach.* **49**, 390–391 (Sept. 2011).
7. Eric Ayars, “Using Arduino microcontrollers as inexpensive dataloggers,” poster presented at AAPT National Meeting, February 2010, phys.csuchico.edu/~eayars/publications/Arduino-datalogger.pdf.
8. C. F. Griffin and S. Stakleff, “An easy-to-construct photogate housing,” *Phys. Teach.* **23**, 250 (April 1985).
9. OPTEK Technology, “Application circuits for the phototransistor switch,” *Application Bulletin* **213** (Dec. 2004), www.optekinc.com/pdf/App Bulletin 213-Opto Assemblies.pdf.
10. Brian W. Evans, *Arduino Programming Notebook,* 2nd ed. (Lulu, 2008).

Calin Galeriu *is teaching physics at Becker College and math at Bay Path Regional Vocational High School. He has received a BS degree in physics from the University of Bucharest, an MA degree from Clark University, and a PhD degree from Worcester Polytechnic Institute.*
Becker College, 61 Sever St., Worcester, MA, 01609; Calin.Galeriu@becker.edu