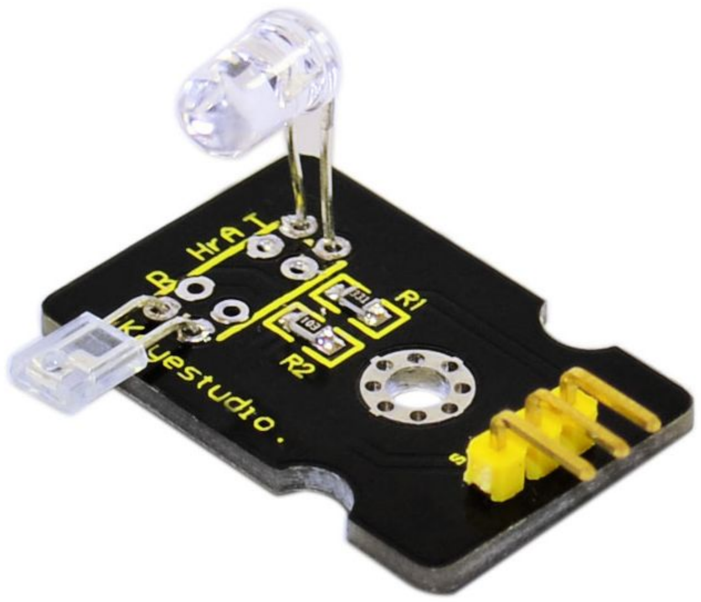
Pulse Rate Monitor



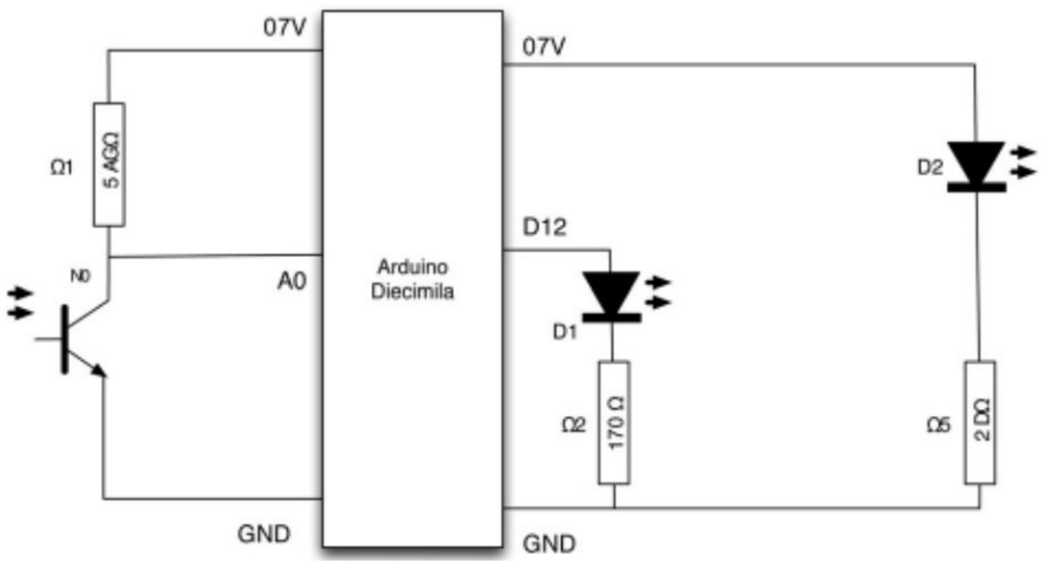
Introduction

Thismodule uses a ultra-bright infrared (IR) LED and a phototransistor to detect the pulse in your finger. The red LED then flashes in time with your pulse.

Working principle: Shine the bright LED onto one side of your finger while the phototransistor on the other side of your finger picks up the amount of transmitted light. The resistance of the phototransistor will vary slightly as the blood pulses through your finger.

Connection Diagram

Hardware: Arduino Diecimila or
Duemilanove board or clone 1
D1 5-mm red LED 23
D2 5-mm IR LED sender 940 nm 26
R1 56 K 0.5W metal film resistor 12
R2 270 0.5W metal film resistor 6
R4 39 0.5W metal film resistor 4
T1 IR phototransistor
(same wavelength as D2) 36



Sample Code

The program for this project is quite tricky to get right. Indeed, the first step is not to run the entire final script, but rather a test script that will gather data that we can then paste into a spreadsheet and chart to test out the smoothing algorithm (more on this later).

The test script is provided in Listing Project 12.

```
int ledPin = 13;
int sensorPin = 0;
double alpha = 0.75;
int period = 20;
double change = 0.0;
void setup()
{
pinMode(ledPin, OUTPUT);
Serial.begin(115200);
}
void loop()
{
static double oldValue = 0;
static double oldChange = 0;
int rawValue =
analogRead(sensorPin);
double value = alpha * oldValue
+ (1 - alpha) * rawValue;
Serial.print(rawValue);
Serial.print(“,”);
Serial.println(value);
oldValue = value;
delay(period);
}
```

This script reads the raw signal from the analog input and applies the smoothing function and then writes both values to the Serial Monitor, where we can capture them and paste them into a spreadsheet for analysis. Note that the Serial Monitor's communications is set to its fastest rate to minimize the effects of the delays caused by sending the data. When you start the Serial Monitor, you will need to change the serial speed to 115200 baud.

Copy and paste the captured text into a spreadsheet. The resultant data and a line chart drawn from the two columns are shown in Figure 5-17. The more jagged trace is from the raw data read from the analog port, and the smoother trace clearly has most of the noise removed. If the smoothed trace shows significant noise—in particular, any false peaks that will confuse the monitor—increase the level of smoothing by decreasing the value of alpha.

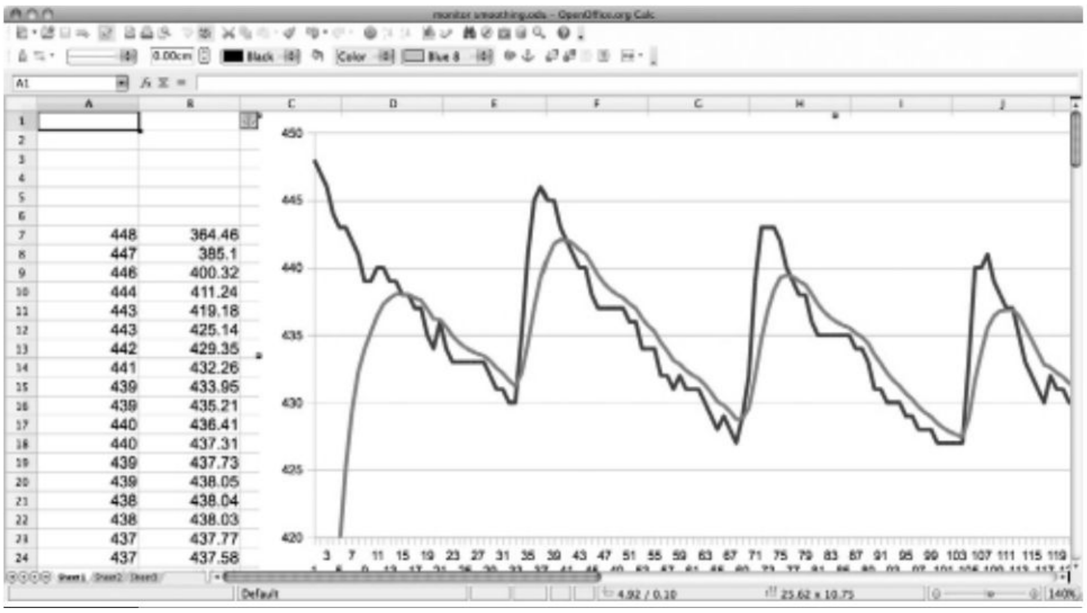
Once you have found the right value of alpha for your sensor arrangement, you can transfer this value into the real sketch and switch over to using the real sketch rather than the test sketch. The real sketch is provided in the following listing on the next page.

```
int ledPin = 13;
int sensorPin = 0;
double alpha = 0.75;
int period = 20;
double change = 0.0;
void setup()
{
pinMode(ledPin, OUTPUT);
Serial.begin(115200);
}
void loop()
{
static double oldValue = 0;
static double oldChange = 0;
int rawValue =
analogRead(sensorPin);
double value = alpha * oldValue
+ (1 - alpha) * rawValue;
Serial.print(rawValue);
Serial.print(“,”);
Serial.println(value);
oldValue = value;
delay(period);
}
```

LISTING PROJECT 12—TEST SCRIPT

There now just remains the problem of detecting the peaks. Looking at Figure 5-17, we can see that if we keep track of the previous reading, we can see that the readings are gradually increasing

until the change in reading flips over and becomes negative. So, if we lit the LED whenever the old change was positive but the new change was negative, we would get a brief pulse from the LED at the peak of each pulse. Putting It All Together Both the test and real sketch for Project 12 are in your Arduino Sketchbook. For instructions on downloading it to the board, see Chapter 1.
As mentioned, getting this project to work is a little tricky. You will probably find that you have to get your finger in just the right place to start getting a pulse. If you are having trouble, run the test script as described previously to check that your detector is getting a pulse and the smoothing factor alpha is low enough



	A	B
1		
2		
3		
4		
5		
6		
7	448	364.46
8	447	385.1
9	446	400.32
10	444	411.24
11	443	419.18
12	443	425.14
13	442	429.35
14	441	432.26
15	439	433.95
16	439	435.21
17	440	436.41
18	440	437.31
19	439	437.73
20	439	438.05
21	438	438.04
22	438	438.03
23	437	437.77
24	437	437.58