Web Site: www.parallax.com
Forums: forums.parallax.com
Sales: sales@parallax.com
Technical: support@parallax.com

Office: (916) 624-8333
Fax: (916) 624-8003
Sales: (888) 512-1024
Tech Support: (888) 997-8267

TSL235R Light to Frequency Converter (#604-00084) Arduino Demo

Please note: This demo was created to support the 2013 National microMedic Contest kits, which are no longer available.

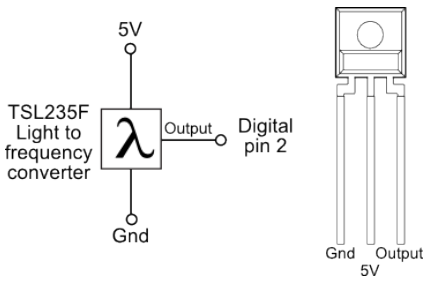
The TSL235R Light to Frequency Converter registers the intensity of light, and outputs a corresponding stream of pulses. The frequency of these pulses is proportional to the brightness of the light. This demo connects the TSL235R to the Board of Education Shield for Arduino and displays sensor values in the Serial Monitor.

- At lower brightness, the sensor outputs a series of pulses at a low frequency.
- Conversely the higher the brightness, the higher the frequency.

A main benefit of the TSL235R sensor is that it has a higher range of light intensities it can respond to, compared to typical all-analog sensors. With appropriate modification of code, it can function in very low or very high light situations.

Connections

To connect the light sensor to the Arduino Shield, attach the three leads of the device as shown in the figure.



Copyright © Parallax Inc.

Programming

To use this example, upload LightSensor sketch to your Arduino, then open the Serial Monitor window. Ensure that the Baud Rate is set at 115200 (note the higher than usual speed used in this sketch). Aim the front of the light sensor toward various light levels, and watch the value change in the Serial Monitor window.

You can adjust the values used in the sketch to increase the affective sensitivity of the sensor. One method is to increase the numeric value in the line

```
if (millis() - lastRead > 5)
```

...to something like 20 or 25. This causes the sketch to update the frequency counting less often, thus accumulating a higher value before displaying it.

High brightness light sources may cause an interruption in the readout through the Serial Monitor. This is due to how the sketch uses one of the Arduino's external hardware interrupt pins. When the light is more intense, the frequency of the sensor increases proportionately, and this causes the interrupt to re-trigger very quickly. The rest of the code in the sketch does not have the chance to execute, causing a delay until the next value appears in the Serial Monitor window. If using the light sensor under high brightness conditions cover with a suitable filter to reduce its overall exposure.

```
volatile unsigned long count = 0;
unsigned long oldcount = 0;
unsigned long temp = 0;
unsigned long lastRead;
unsigned long avg = 0;
unsigned long freq = 0;
long loopcount = 0;

void setup() {
  Serial.begin(115200);                 // Note higher baud rate
  Serial.println("Starting...");
  pinMode(2, INPUT);            // Set pin 2 to input
  digitalWrite(2, HIGH);        // Turn on pullup resistor
  attachInterrupt(0, count_inc, RISING);
}

void loop() {
  if (millis() - lastRead > 5) {
    lastRead = millis();
    temp = count;
    freq = temp - oldcount;
    avg = avg + freq;
    if (loopcount % 50) {
      Serial.println(avg / 50);
      avg = 0;
      loopcount = 0;
    }
    loopcount++;
    oldcount = temp;
```

Copyright © Parallax Inc.

```
  }
}

void count_inc() {
  count++;
}
```