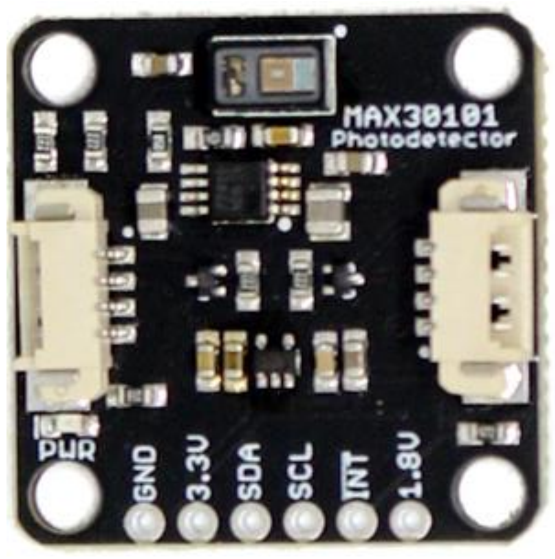
SmartElex Photodetector Breakout – MAX30101



The MAX30101 includes three LEDs and an optical detector in a single package, which can be utilized as a wearable, biosensor for pulse oximeter and heart-rate measurements. Other possible applications include proximity sensing and particle detection by measuring the changes in light that is reflected back from the LEDs.

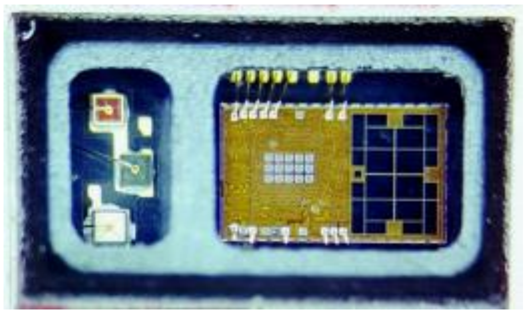
☠ **WARNING:** Our products are **NOT** intended to diagnose or treat any conditions.

Power LED

There is a power status LED to help make sure that your Photodetector Sensor is getting power. You can power the board either through the polarized **connector** or the **breakout pins** (**3.3V** and **GND**) provided. A jumper is available on the back of the board to remove power to the LED for low-power applications (*see **Jumpers** section below*).

MAX30101

The MAX30101 includes three LEDs and an optical detector in a single package. Behind the window on the left, are red, green, and IR LEDs. While behind the window on the right, is a highly sensitive photon detector.



Close up of the MAX30101 sensor.

The working principle of the sensor is that the optical detector measures the changes in the reflected light that was emitted from the LEDs. This is great for various application like detecting particles or for **photoplethysmography**.

(**For more details on the MAX30101, users can check out the datasheet.*)

Characteristic	**Description**
Power	Supply Voltage: **1.7 - 2.0V** Supply Cuurent: **0.6 - 1.1mA** LED Driver: • Red/IR: 3.1-5V • Green: 4.5 - 5.5V
ADC	Resolution: 18-bits (65536 Counts)

LEDs	Wavelength: • IR: 870 - 900nm • Red: 650 - 670nm • Green 530 - 545nm Power: • IR: 6.5mW • Red: 9.8mW • Green 17.2mW
Photodetector	Spectral Range: 640 - 980nm
Temperature Sensor	Range: -40 - 85°C Accuracy: ±1°C
I2C Address	**0x57**

I^{2}C Address

The Photodetector Sensor's I^{2}C address, **0x57**, is factory set.

Connectors

The simplest way to use the Photodetector Sensor is through the connect system. The connectors are polarized for the I^{2}C connection and power. (**They are tied to the corresponding power and I^{2}C breakout pins.*)

Breakout Pins

The board also provides six labeled breakout pins. You can connect these lines to the I^{2}C bus of your microcontroller and power pins (**3.3V** and **GND**), if it doesn't have a connector. The interrupt pin is also broken out to use for triggered events.

Interrupt Pin

The interrupt pins (*active high*) are used to indicate various states of the ADXL313, depending on how they are configured and if they are enabled. The INT pins are pulled down with a **4.7kΩ** resistor.

Jumpers

There are three jumpers on the board.

Power LED

Cutting the **LED** jumper will remove the **1kΩ** resistors and PWR LED from the **3.3V** power. This is useful for low power applications

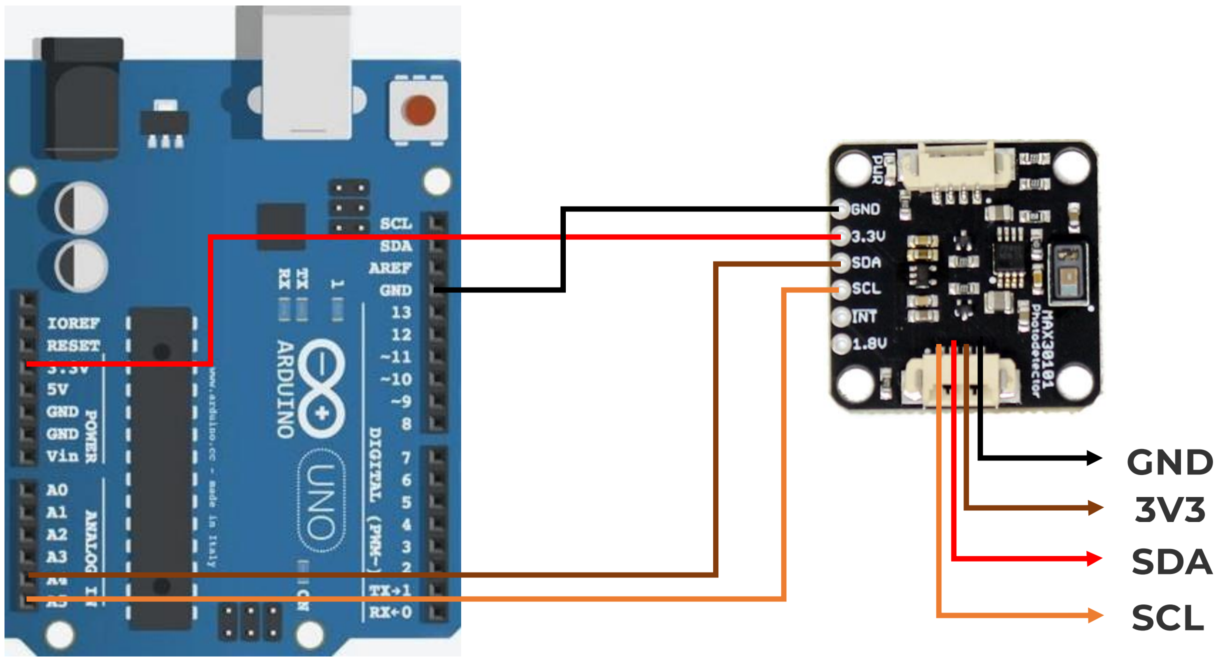
I^{2}C Pull-Up

Cutting the **I^{2}C** jumper will remove the **2.2kΩ** pull-up resistors from the I^{2}C bus. If you have many devices on your I^{2}C bus you may want to remove these jumpers. Be aware that these resistors are also part of the transistor logic level shifting circuit.

Interrupt Pull-up

Cutting the **INT** jumper will remove the **4.7kΩ** pull-up resistors from the interrupt pin.

Wiring



Arduino	MAX30101
A5(SCL)	SCL
A4(SDA)	SDA
3.3V	3V3
GND	GND

Arduino Examples

☠ **Do not rely on our examples for medical diagnosis or any life saving applications**

Note: Particle detection, heart rate measurement, and photoplethysmography (for pulse oximetry) are applications of the MAX30101. These applications require a fundamental understanding of the operating principles of the sensor and a conceptual knowledge of the applications. Although, we provide some examples for these applications; they are primarily for demonstration purposes only and are not supported by SmartElex.

Note: Although there is an example, Maxim has since removed the proximity sensing and particle detection as functionalities of this sensor in their datasheet.

Arduino Library

Daniel Wiese has written a library to work with the SmartElex Photodetector Sensor – MAX30101. You can obtain this library through the Arduino Library Manager by searching for " **MAX3010x Sensor library** ". Find the one written by Daniel Wiese and install the latest version.

Example - Heartrate

Once you've got the library installed, open the **MAX30105PulseoximeterHeartrate** sketch.

```
#include <MAX3010x.h>
#include "filters.h"

// Sensor (adjust to your sensor type)
MAX30105 sensor;
```

```
const auto kSamplingRate = sensor.SAMPLING_RATE_400SPS;
const float kSamplingFrequency = 400.0;

// Finger Detection Threshold and Cooldown
const unsigned long kFingerThreshold = 10000;
const unsigned int kFingerCooldownMs = 500;

// Edge Detection Threshold (decrease for MAX30100)
const float kEdgeThreshold = -2000.0;

// Filters
const float kLowPassCutoff = 5.0;
const float kHighPassCutoff = 0.5;

// Averaging
const bool kEnableAveraging = true;
const int kAveragingSamples = 50;
const int kSampleThreshold = 5;

void setup() {
  Serial.begin(9600);

  if(sensor.begin() && sensor.setSamplingRate(kSamplingRate)) {
    Serial.println("Sensor initialized");
  }
  else {
    Serial.println("Sensor not found");
    while(1);
  }
}

// Filter Instances
HighPassFilter high_pass_filter(kHighPassCutoff, kSamplingFrequency);
LowPassFilter low_pass_filter(kLowPassCutoff, kSamplingFrequency);
Differentiator differentiator(kSamplingFrequency);
MovingAverageFilter<kAveragingSamples> averager;

// Timestamp of the last heartbeat
long last_heartbeat = 0;

// Timestamp for finger detection
long finger_timestamp = 0;
bool finger_detected = false;

// Last diff to detect zero crossing
float last_diff = NAN;
bool crossed = false;
```

```
long crossed_time = 0;

void loop() {
  auto sample = sensor.readSample(1000);
  float current_value = sample.red;

  // Detect Finger using raw sensor value
  if(sample.red > kFingerThreshold) {
    if(millis() - finger_timestamp > kFingerCooldownMs) {
      finger_detected = true;
    }
  }
  else {
    // Reset values if the finger is removed
    differentiator.reset();
    averager.reset();
    low_pass_filter.reset();
    high_pass_filter.reset();

    finger_detected = false;
    finger_timestamp = millis();
  }

  if(finger_detected) {
    current_value = low_pass_filter.process(current_value);
    current_value = high_pass_filter.process(current_value);
    float current_diff = differentiator.process(current_value);

    // Valid values?
    if(!isnan(current_diff) && !isnan(last_diff)) {

      // Detect Heartbeat - Zero-Crossing
      if(last_diff > 0 && current_diff < 0) {
        crossed = true;
        crossed_time = millis();
      }

      if(current_diff > 0) {
        crossed = false;
      }

      // Detect Heartbeat - Falling Edge Threshold
      if(crossed && current_diff < kEdgeThreshold) {
        if(last_heartbeat != 0 && crossed_time - last_heartbeat > 300) {
          // Show Results
          int bpm = 60000/(crossed_time - last_heartbeat);
          if(bpm > 50 && bpm < 250) {
```

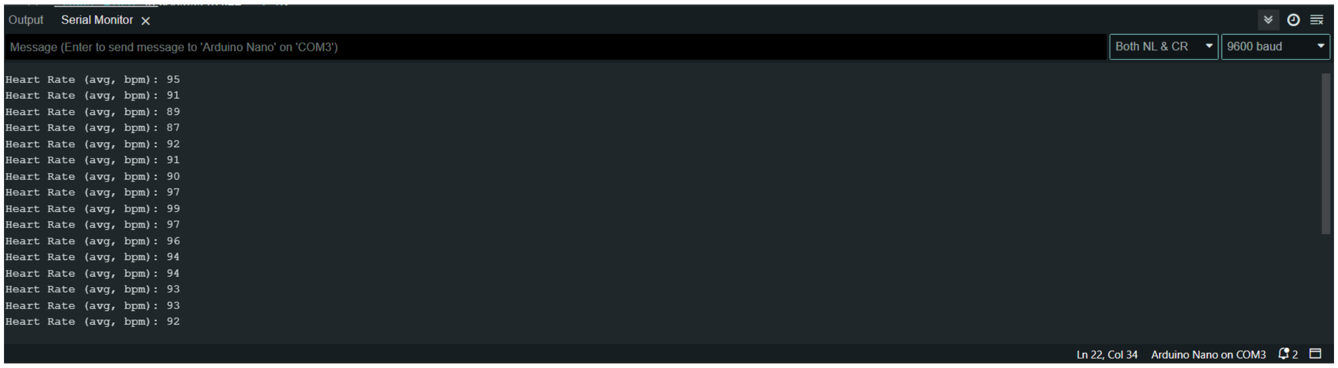
```
      // Average?
      if(kEnableAveraging) {
        int average_bpm = averager.process(bpm);

        // Show if enough samples have been collected
        if(averager.count() > kSampleThreshold) {
          Serial.print("Heart Rate (avg, bpm): ");
          Serial.println(average_bpm);
        }
      }
      else {
        Serial.print("Heart Rate (current, bpm): ");
        Serial.println(bpm);
      }
     }
    }

    crossed = false;
    last_heartbeat = crossed_time;
   }
  }

  last_diff = current_diff;
 }
}
/////////////////////////////////////////////END/////////////////////////////////////////
```

Then load it onto your Uno. Open your favorite Serial Terminal to see the printed values.



```
Heart Rate (avg, bpm): 95
Heart Rate (avg, bpm): 91
Heart Rate (avg, bpm): 89
Heart Rate (avg, bpm): 87
Heart Rate (avg, bpm): 92
Heart Rate (avg, bpm): 91
Heart Rate (avg, bpm): 90
Heart Rate (avg, bpm): 97
Heart Rate (avg, bpm): 99
Heart Rate (avg, bpm): 97
Heart Rate (avg, bpm): 96
Heart Rate (avg, bpm): 94
Heart Rate (avg, bpm): 94
Heart Rate (avg, bpm): 93
Heart Rate (avg, bpm): 93
Heart Rate (avg, bpm): 92
```