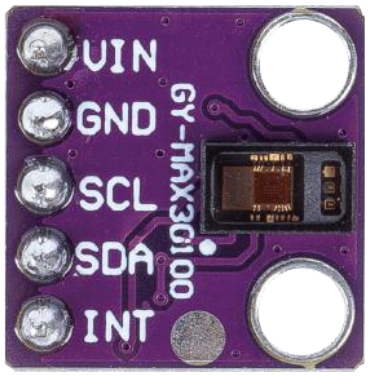
Pulse Oximeter and Heart Rate Sensor (MAX30102)



The MAX30102 is an advanced sensor module designed for tracking heart rate and blood oxygen levels (SpO2). Manufactured by Maxim Integrated, it combines pulse oximetry and heart rate monitoring into a compact package, making it a popular choice for wearable health and fitness applications.

Features

- Reflective Pulse Oximetry: The module employs a reflective-based system, offering the flexibility to work from different parts of the body, not just the fingertip.
- On-board LEDs: It comes equipped with red and infrared LEDs, aiding in SpO2 and heart rate measurements.
- High Sensitivity: The MAX30102 can detect even the slightest pulses and variations, granting it high accuracy in readings.
- FIFO Data Storage: Features an integrated First-In, First-Out (FIFO) data storage system, streamlining the data gathering process.

- I2C Interface: With an I2C communication protocol, it's easily integrated into a wide range of microcontroller-based systems.
- Low Power Consumption: Designed with wearables in mind, the module operates with minimal energy, supporting extended battery life for portable devices.

Specification

- Chip Type: MAX30102
- LED Peak Wavelength: 660nm/880nm
- Supply Voltage: 3.3V or 5V;
- Detection Signal Type: Optical Reflection Signal (PPG)
- Output Signal Interface: I2C Interface
- PCB size: 14 x 14mm
- Working current: <5mA
- Working temperature: -40 ~ +85℃

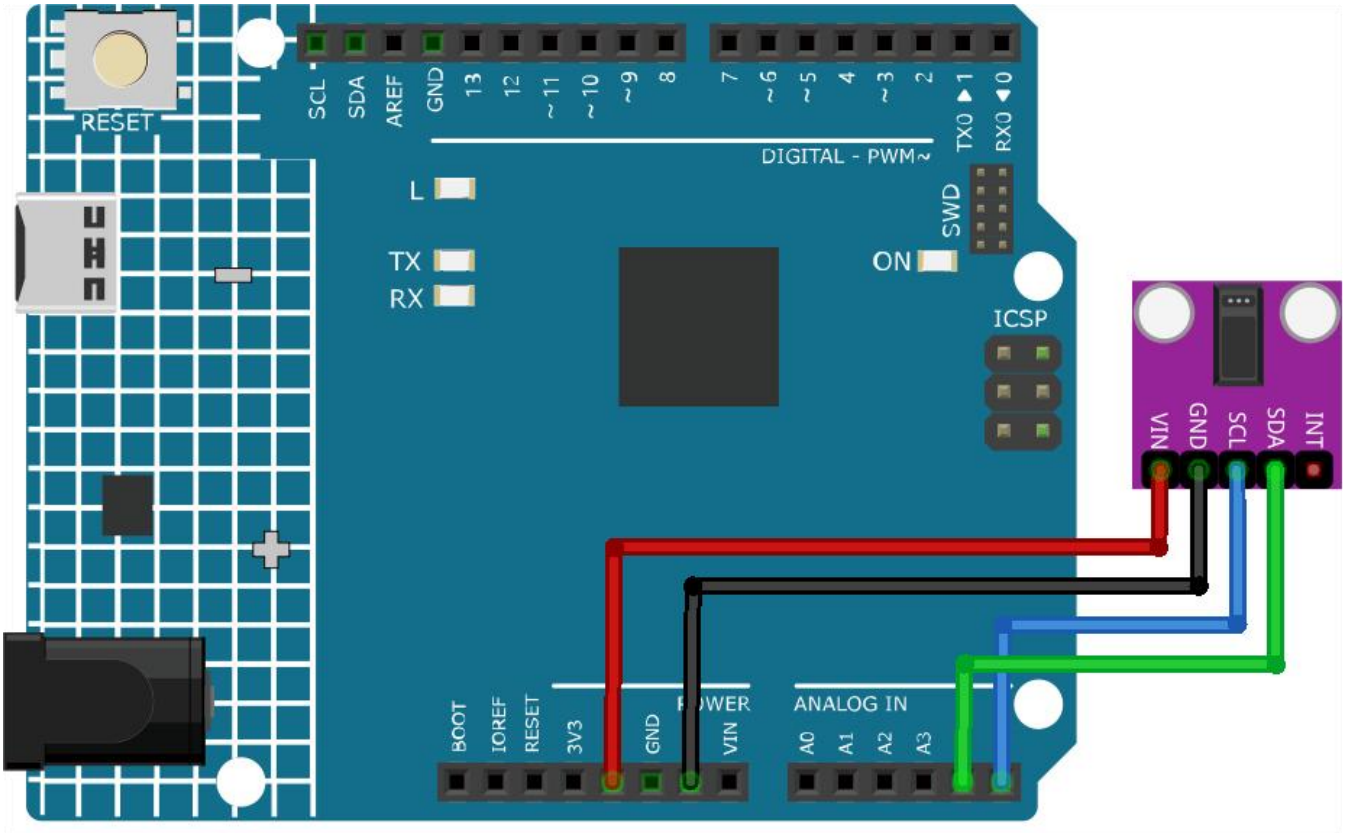
Pinout

- VCC: This is the positive power supply input from the main control.
- GND: Ground connection.
- SCL: serial clock pin for the I2C interface.
- SDA: serial data pin for the I2C interface.
- INT: the Interrupt pin of the IC.

How to Use in Arduino?

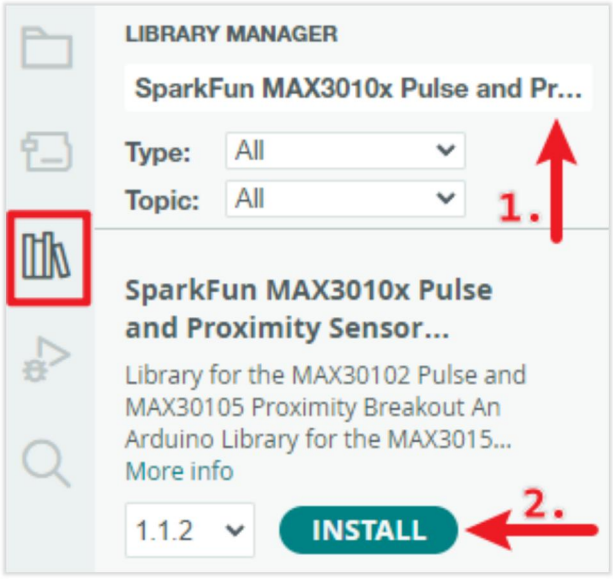
Build the Circuit

Heart Rate Sensor Module	Arduino Board
VCC	5V
GND	GND
SCL	A5
SDA	A4
INT	



Copy and Upload the Code

1. Search and install "**SparkFun MAX3010x Pulse and Proximity Sensor Library**" library in Arduino IDE.



2. Copy and paste the following code to the Arduino IDE.

```
#include <Wire.h>
#include "MAX30105.h"
#include "heartRate.h"

MAX30105 particleSensor;

const byte RATE_SIZE = 4;    //Increase this for more averaging. 4 is good.
byte rates[RATE_SIZE];       //Array of heart rates
byte rateSpot = 0;
long lastBeat = 0;   //Time at which the last beat occurred

float beatsPerMinute;   //Current BPM value
int beatAvg;            //Average BPM value

void setup() {
```

```
  Serial.begin(9600);
  Serial.println("Initializing...");

  // Initialize sensor
  if (!particleSensor.begin(Wire, I2C_SPEED_FAST))  //Use default I2C port, 400kHz speed
  {
    Serial.println("MAX30102 was not found. Please check wiring/power. ");
    while (1) ; //Infinite loop to stop the program
  }
  Serial.println("Place your index finger on the sensor with steady pressure.");

  particleSensor.setup();                    //Configure sensor with default settings
  particleSensor.setPulseAmplitudeRed(0x0A);  //Turn Red LED to low to indicate sensor is running
  particleSensor.setPulseAmplitudeGreen(0);  //Turn off Green LED
}

void loop() {
  long irValue = particleSensor.getIR();

  if (checkForBeat(irValue) == true) {
    //Calculate beatsPerMinute
    long delta = millis() - lastBeat;
    lastBeat = millis();
    beatsPerMinute = 60 / (delta / 1000.0);

    if (beatsPerMinute < 255 && beatsPerMinute > 20) {  //Check if the BPM value is within a valid range
      rates[rateSpot++] = (byte)beatsPerMinute;          //Store this reading in the array
```

```
        rateSpot %= RATE_SIZE;                                    //Wrap
variable

        //Take average of readings
        beatAvg = 0;
        for (byte x = 0; x < RATE_SIZE; x++)
          beatAvg += rates[x];
        beatAvg /= RATE_SIZE;
      }
    }

    //Print the IR value, current BPM value, and average BPM value to
the serial monitor
    Serial.print("IR=");
    Serial.print(irValue);
    Serial.print(", BPM=");
    Serial.print(beatsPerMinute);
    Serial.print(", Avg BPM=");
    Serial.print(beatAvg);

    if (irValue < 50000)
      Serial.print(" No finger?");

    Serial.println();
}
```

3. Choose the board and port you use.

4. Click the **Upload** button.

5. Open the **serial monitor** in the upper right corner of the Arduino IDE, set the baud rate to **9600** in the serial monitor, and then you can see the output.