LamaPLC: MAX30100/MAX30102 Heart Rate Click Sensor Module

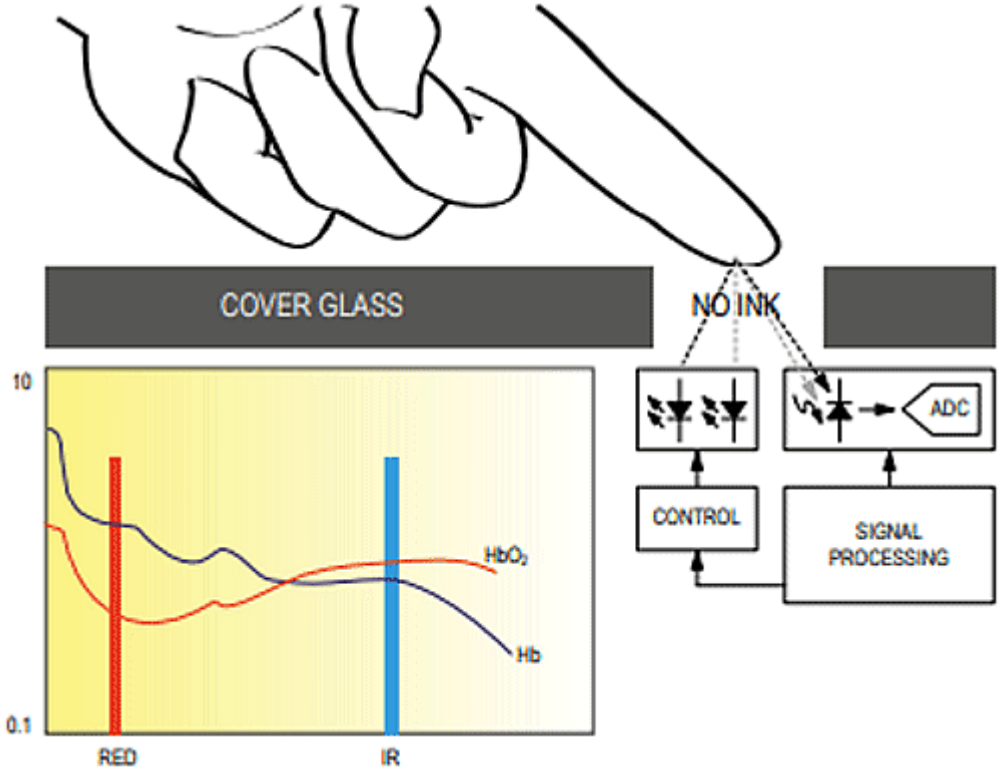
The **MAX30102** and **MAX30100** are integrated pulse oximetry and heart-rate monitor sensor modules commonly used in wearable health devices. While both use red and infrared LEDs to measure blood oxygen (SpO2) and heart rate (BPM), the MAX30102 is the upgraded version with significant technical improvements.

Key Differences

		
Feature	MAX30100	MAX30102 (Upgrade)
FIFO Buffer	16-deep (smaller storage)	32-deep (stores more data)
ADC Resolution	14-bit	18-bit (higher sensitivity)
Voltage	1.8V and 3.3V	1.8V (logic) & 3.3V–5.0V (LEDs)
Temperature Sensor	No internal temperature sensor	Includes on-chip temperature sensor (±1°C accuracy)
Motion Rejection	Basic	Improved ambient light and motion rejection
Interfacing	Both use the I²C protocol (SDA/SCL) The default 7-bit I²C slave address for both sensors is **0x57**. This corresponds to an 8-bit write address of 0xAE and an 8-bit read address of 0xAF	
Accuracy	Calibrated tests show roughly 95.8% to 97% accuracy for heart rate monitoring	

Working of the MAX30100 Oximeter

The sensor consists of a pair of light-emitting diodes, one emitting monochromatic red light at 660 nm and the other emitting infrared light at 940 nm. These wavelengths are chosen because at these wavelengths, oxygenated and deoxygenated hemoglobin have very different absorption properties. As shown in the graph below, there is a clear difference between HbO2 (oxygenated Hb) and Hb (deoxygenated Hb) when subjected to these specific wavelengths.



Pinout

Both modules share the same pin layout for common use cases.

- **VIN:** This is the main power supply input pin for the entire module. You can typically connect this to a 3.3V or 5V source from your microcontroller, as most boards include on-board voltage regulators.
- **GND:** The common ground connection for the power supply and signal reference.
- **SCL:** The I²C Serial Clock input pin. This connects to the SCL pin of your microcontroller (e.g., A5 on an Arduino Uno).
- **SDA:** The I²C Serial Data pin, which is bidirectional. This connects to the SDA pin of your microcontroller (e.g., A4 on an Arduino Uno).
- **INT:** An Active-Low Interrupt output pin. This pin goes low when a specific event (such as a heartbeat) is detected, enabling efficient, event-driven programming. It can be connected to any digital interrupt pin on your microcontroller, but it is optional for basic use.
- **RD:** The Red LED driver connection pin. This is typically left unconnected unless you want to drive the LED manually.
- **IRD:** The Infrared LED driver connection pin. Similar to the RD pin, it's usually left unconnected.

Important Pin Notes

- **Voltage Levels:** The core MAX30102 IC operates internally at 1.8V, with the LEDs running at ~3.3V. The VIN pin on standard breakout boards is designed to accept a wider range (3.3V-5V) because the boards include the necessary voltage regulators for proper operation.
- **Unused Pins:** For standard operation using the I²C interface, you need to connect only VIN, GND, SCL, and SDA. The INT pin provides additional functionality for more advanced projects. The RD and IRD pins are generally unused.
- **Pull-up Resistors:** Most breakout boards already include the necessary pull-up resistors for the I²C lines (SCL and SDA) and the INT pin, simplifying the wiring.

Arduino code

For the MAX30102, the most robust library is the SparkFun MAX3010x Pulse and Proximity Sensor Library.

```
#include <Wire.h>
#include "MAX30105.h" // Works for MAX30102

MAX30105 particleSensor;

void setup() {
  Serial.begin(115200);
  if (!particleSensor.begin(Wire, I2C_SPEED_FAST)) { // 400kHz speed
    Serial.println("MAX30102 was not found. Check wiring/power.");
    while (1);
  }
  particleSensor.setup(); // Configure with default settings
}

void loop() {
  // Get raw infrared and red light data
  Serial.print(" IR: "); Serial.print(particleSensor.getIR());
  Serial.print(" Red: "); Serial.println(particleSensor.getRed());
}
```

MAX30102, MAX30100, Heart Rate Click, sensor, communication, I2C, arduino, code