心率血氧传感器用户手册

V1.1

发布说明：

日期	版本	内容
20210514	V1.0	初版
20221016	V1.1	更新

YFROBOT

目录

1. 简介

心率血氧传感器（乐高外壳、黑板），采用MAX30102芯片设计制作。其可检测人体脉搏血氧值及心率测量，常被用于健身货医疗保健中，例如：智能手环等。

心率血氧传感器具有统一的兼容乐高积木的安装孔，可轻松完成乐高积木的拼接，实现创意设计。

MAX30102是集成式脉搏血氧仪，心率监测器模块。它包括内部LED，光电探测器，光学元件和低噪声电子设备具有环境光抑制能力。 MAX30102提供完整的系统解决方案，可简化设计过程适用于移动和可穿戴设备。使用 I2C 的标准进行通信接口。可以通过软件关闭模块零待机电流，使电源轨能够始终保持供电。

更多芯片相关信息请参考资料中的数据手册。

⚠ *注意：本产品非医疗器械，测量数据与结果仅供学习参考，不可作为诊疗依据！*

2. 原理说明

光溶积法：利用人体组织在血管搏动时造成透光率不同来进行脉搏和血氧饱和度测量；

光源：采用对动脉血中氧合血红蛋白(Hb02）-和血红蛋白（Hb）有选择性的特定波长的发光极管；

透光率转化为电信号：动脉搏动充血容积委化导致够束光的透光率发生改变，此时由光电变换接收经人体组织反身光线，转变为电信号并将其放大输出。

3. 规格参数

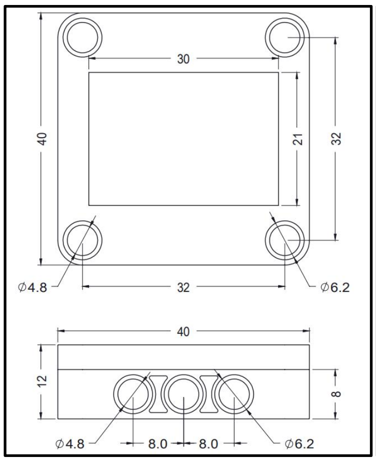
工作电压：DC 3.3-5V

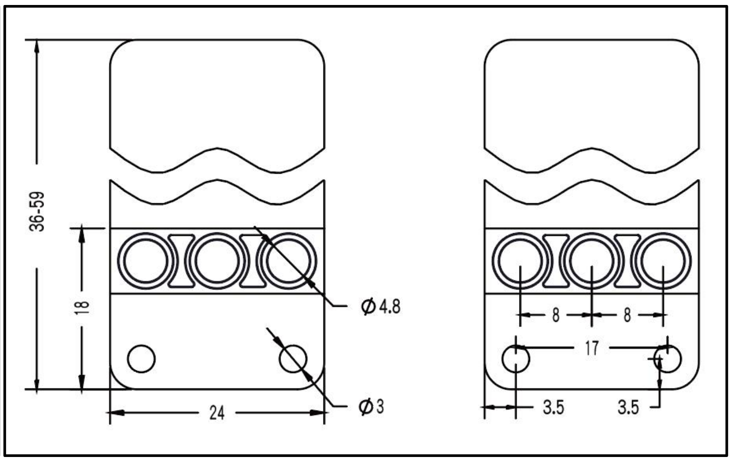
通信方式：I2C，地址0x57

LED峰值波长：660nm/880nm

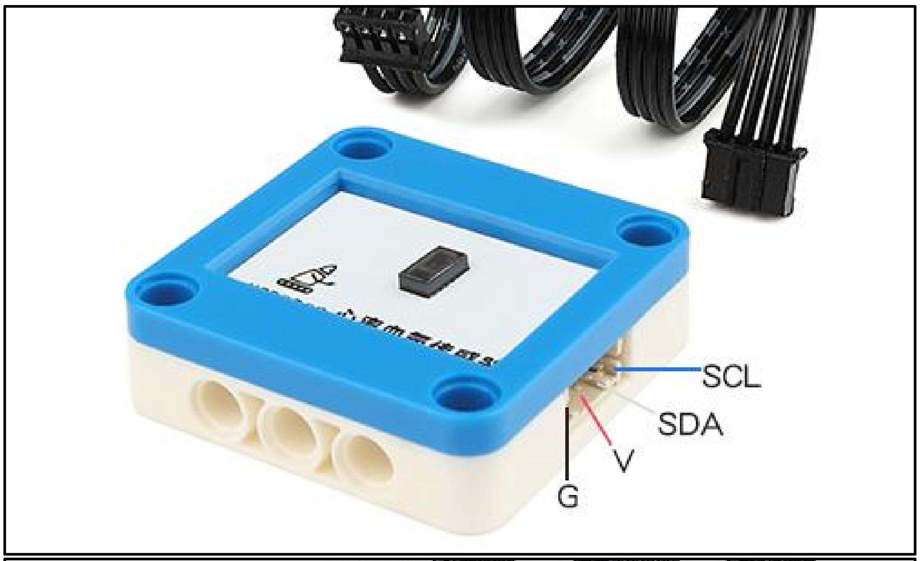
检测信号类型：光反射信号（PPG）

外壳尺寸，单位MM


30
40
21
32
∅4.8
32
∅6.2
40
12
8
∅4.8
8.0
8.0
∅6.2


36-59
18
∅4.8
24
∅3
8
8
17
3.5
3.5

4. 引脚说明


SCL
SDA
V
G

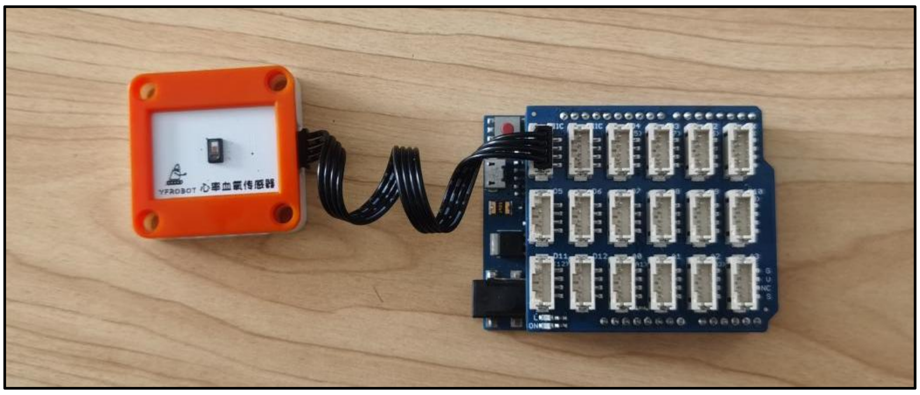
SCL
SDA
V
G

5. 应用示例

⚠ *注意：硬件应用于示例演示，可能需要另购；如有不明请咨询本司客服！*

5.1. 电路连接

心率血氧传感器的 G、V、SDA、SCL分别连接 Arduino UNO的GND、VCC、SDA(A4)、SCL(A5)引脚。



5.2. Arduino IDE示例代码

⚠ *注意：程序需要添加库文件，否则无法正常编译，添加方式见附录2*

复制代码至Arduino IDE中编译上传，并观察结果。

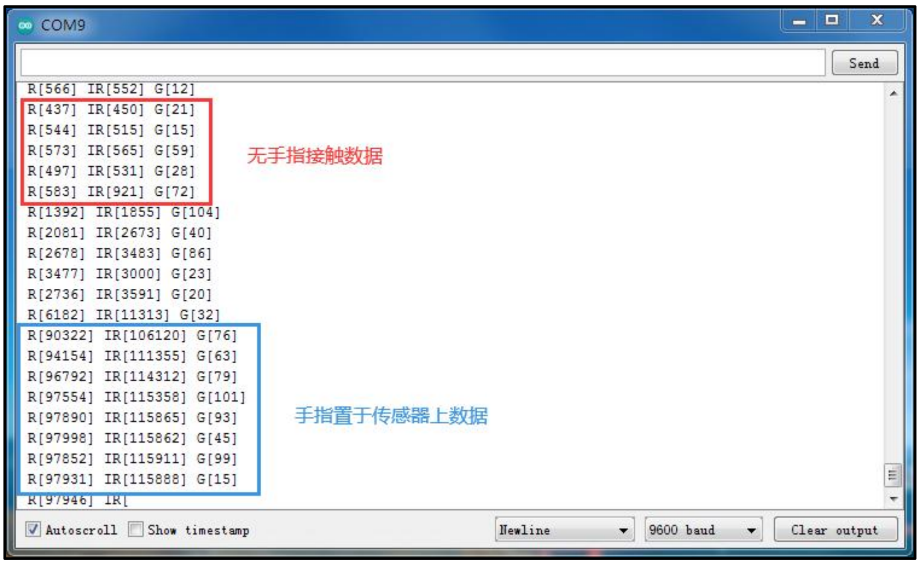
5.2.1. 示例一：原始数据读取

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

MAX30105 particleSensor;

#define debug Serial //Uncomment this line if you're using an Uno or ESP
//#define debug SerialUSB //Uncomment this line if you're using a SAMD21

void setup()
{
  debug.begin(9600);
  debug.println("MAX30105 Basic Readings Example");

  // Initialize sensor
  if (particleSensor.begin() == false)
  {
    debug.println("MAX30105 was not found. Please check wiring/power. ");
    while (1);
  }
```

```
    particleSensor.setup(); //Configure sensor. Use 6.4mA for LED drive
}

void loop()
{
    debug.print(" R[");
    debug.print(particleSensor.getRed());
    debug.print("] IR[");
    debug.print(particleSensor.getIR());
    debug.print("] G[");
    debug.print(particleSensor.getGreen());
    debug.print("]");

    debug.println();
}
```

示例一：程序运行结果

串口打印输出Red/IR/Green原始数据。



```
R[566] IR[552] G[12]
R[437] IR[450] G[21]
R[544] IR[515] G[15]
R[573] IR[565] G[59]
R[497] IR[531] G[28]
R[583] IR[921] G[72]
R[1392] IR[1855] G[104]
R[2081] IR[2673] G[40]
R[2678] IR[3483] G[86]
R[3477] IR[3000] G[23]
R[2736] IR[3591] G[20]
R[6182] IR[11313] G[32]
R[90322] IR[106120] G[76]
R[94154] IR[111355] G[63]
R[96792] IR[114312] G[79]
R[97554] IR[115358] G[101]
R[97890] IR[115865] G[93]
R[97998] IR[115862] G[45]
R[97852] IR[115911] G[99]
R[97931] IR[115888] G[15]
R[97946] IR[
```

5.2.2. 示例二：障碍感应

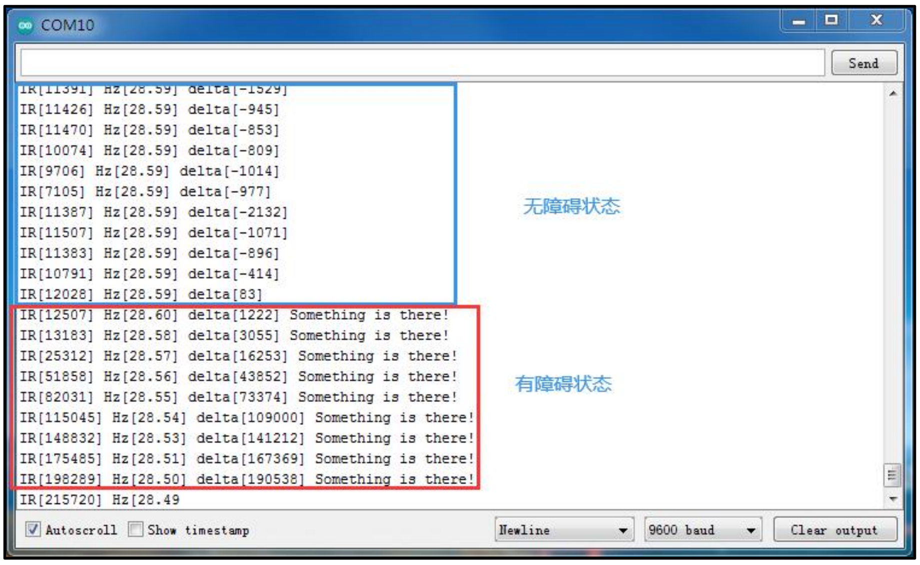
```
#include <Wire.h>
#include "MAX30105.h"

MAX30105 particleSensor;

long samplesTaken = 0; //Counter for calculating the Hz or read rate
long unblockedValue; //Average IR at power up
long startTime; //Used to calculate measurement rate

void setup()
{
  Serial.begin(9600);
  Serial.println("MAX30105 Presence Sensing Example");

  // Initialize sensor
  if (particleSensor.begin() == false) //Use default I2C port
  {
    Serial.println("MAX30105 was not found. Please check wiring/power. ");
    while (1);
  }

  //Setup to sense up to 18 inches, max LED brightness
  byte ledBrightness = 0xFF; //Options: 0=Off to 255=50mA
  byte sampleAverage = 4; //Options: 1, 2, 4, 8, 16, 32
  byte ledMode = 2; //Options: 1 = Red only, 2 = Red + IR, 3 = Red + IR + Green
  int sampleRate = 400; //Options: 50, 100, 200, 400, 800, 1000, 1600, 3200
  int pulseWidth = 411; //Options: 69, 118, 215, 411
  int adcRange = 2048; //Options: 2048, 4096, 8192, 16384

  particleSensor.setup(ledBrightness, sampleAverage, ledMode, sampleRate, pulseWidth, adcRange);
//Configure sensor with these settings

  particleSensor.setPulseAmplitudeRed(0); //Turn off Red LED
  particleSensor.setPulseAmplitudeGreen(0); //Turn off Green LED

  //Take an average of IR readings at power up
  unblockedValue = 0;
  for (byte x = 0 ; x < 32 ; x++)
  {
    unblockedValue += particleSensor.getIR(); //Read the IR value
  }
  unblockedValue /= 32;

  startTime = millis();
}

void loop()
```

```
{
  samplesTaken++;

  Serial.print("IR[");
  Serial.print(particleSensor.getIR());
  Serial.print("] Hz[");
  Serial.print((float)samplesTaken / ((millis() - startTime) / 1000.0), 2);
  Serial.print("]");

  long currentDelta = particleSensor.getIR() - unblockedValue;

  Serial.print(" delta[");
  Serial.print(currentDelta);
  Serial.print("]");

  if (currentDelta > (long)100)
  {
    Serial.print(" Something is there!");
  }

  Serial.println();
}
```

示例二：程序运行结果

串口打印输出IR数据，当感应到前方障碍时，输出“Something is there!”。

⚠ *注意：程序开始时**需要校准**障碍物红外值，即障碍物置于传感器前一定距离（正常白天室内环境光强下，实测<15cm效果较好），点击主板复位按钮。*



```
IR[11391] Hz[28.59] delta[-1529]
IR[11426] Hz[28.59] delta[-945]
IR[11470] Hz[28.59] delta[-853]
IR[10074] Hz[28.59] delta[-809]
IR[9706] Hz[28.59] delta[-1014]
IR[7105] Hz[28.59] delta[-977]
IR[11387] Hz[28.59] delta[-2132]
IR[11507] Hz[28.59] delta[-1071]
IR[11383] Hz[28.59] delta[-896]
IR[10791] Hz[28.59] delta[-414]
IR[12028] Hz[28.59] delta[83]
IR[12507] Hz[28.60] delta[1222] Something is there!
IR[13183] Hz[28.58] delta[3055] Something is there!
IR[25312] Hz[28.57] delta[16253] Something is there!
IR[51858] Hz[28.56] delta[43852] Something is there!
IR[82031] Hz[28.55] delta[73374] Something is there!
IR[115045] Hz[28.54] delta[109000] Something is there!
IR[148832] Hz[28.53] delta[141212] Something is there!
IR[175485] Hz[28.51] delta[167369] Something is there!
IR[198289] Hz[28.50] delta[190538] Something is there!
IR[215720] Hz[28.49
```

5.2.3. 示例三：温度检测

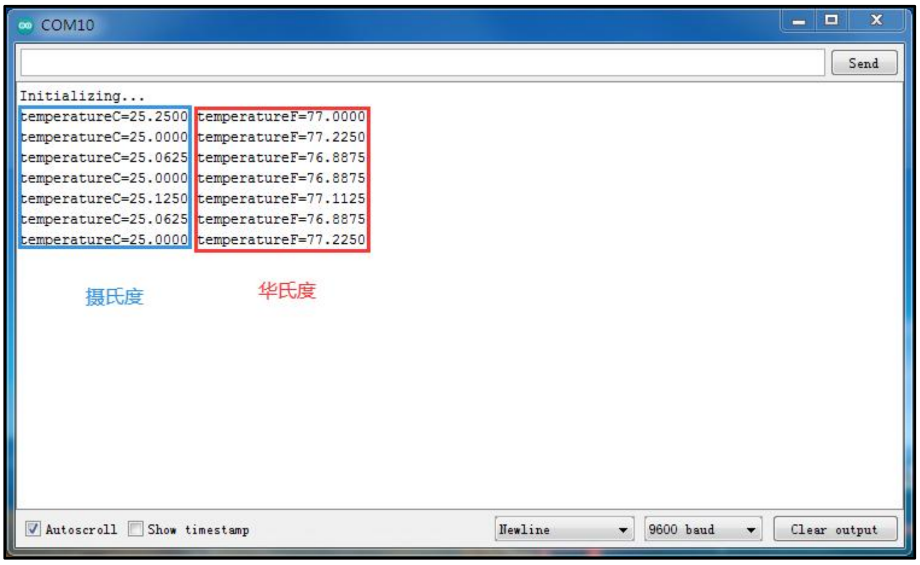
传感器内置温度传感器，可以用于测量当前环境温度。

```
#include <Wire.h>
#include "MAX30105.h"   //Get it here: http://librarymanager/All#SparkFun_MAX30105

MAX30105 particleSensor;

void setup()
{
  Serial.begin(9600);
  Serial.println("Initializing...");

  // Initialize sensor
  if (particleSensor.begin() == false) //Use default I2C port, 400kHz speed
  {
    Serial.println("MAX30105 was not found. Please check wiring/power. ");
    while (1);
  }

  //The LEDs are very low power and won't affect the temp reading much but
  //you may want to turn off the LEDs to avoid any local heating
  particleSensor.setup(0); //Configure sensor. Turn off LEDs
  //particleSensor.setup(); //Configure sensor. Use 25mA for LED drive

  particleSensor.enableDIETEMPRDY(); //Enable the temp ready interrupt. This is required.
}

void loop()
{
  float temperature = particleSensor.readTemperature();

  Serial.print("temperatureC=");
  Serial.print(temperature, 4);

  float temperatureF = particleSensor.readTemperatureF(); //Because I am a bad global citizen

  Serial.print(" temperatureF=");
  Serial.print(temperatureF, 4);
```

```
    Serial.println();
    delay(500);
  }
```

示例三：程序运行结果

串口打印输出温度值。



```
Initializing...
temperatureC=25.2500 temperatureF=77.0000
temperatureC=25.0000 temperatureF=77.2250
temperatureC=25.0625 temperatureF=76.8875
temperatureC=25.0000 temperatureF=76.8875
temperatureC=25.1250 temperatureF=77.1125
temperatureC=25.0625 temperatureF=76.8875
temperatureC=25.0000 temperatureF=77.2250
```

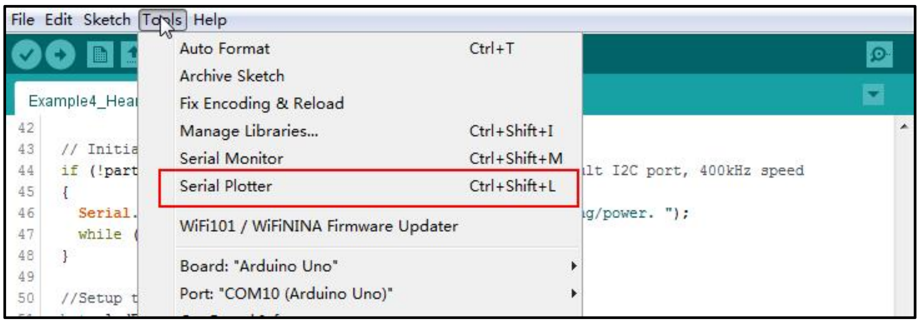
5.2.4. 示例四：绘图仪显示心跳

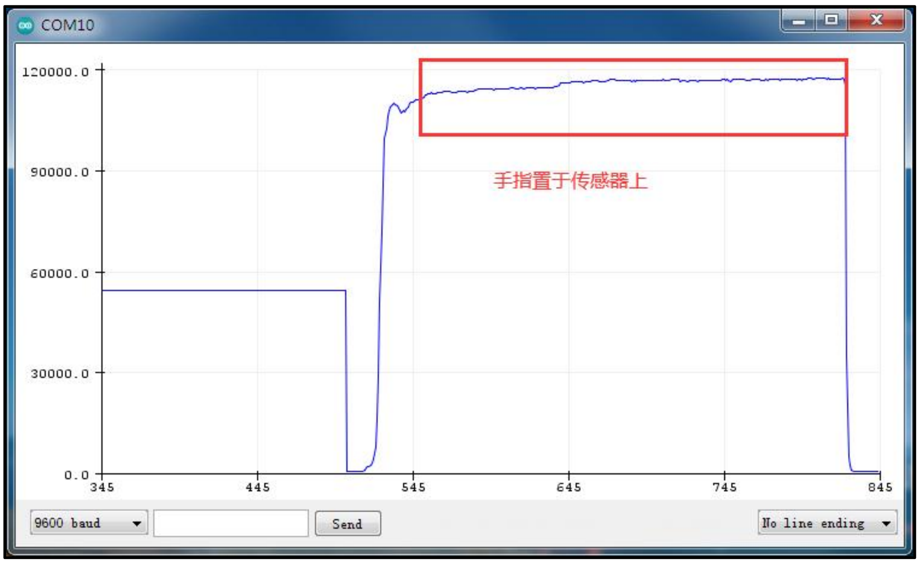
```
  #include <Wire.h>
  #include "MAX30105.h"

  MAX30105 particleSensor;

  void setup()
  {
    Serial.begin(115200);
    Serial.println("Initializing...");

    // Initialize sensor
    if (!particleSensor.begin(Wire, I2C_SPEED_FAST)) //Use default I2C port, 400kHz speed
    {
      Serial.println("MAX30105 was not found. Please check wiring/power. ");
      while (1);
    }

    //Setup to sense a nice looking saw tooth on the plotter
    byte ledBrightness = 0x1F; //Options: 0=Off to 255=50mA
    byte sampleAverage = 8; //Options: 1, 2, 4, 8, 16, 32
    byte ledMode = 3; //Options: 1 = Red only, 2 = Red + IR, 3 = Red + IR + Green
    int sampleRate = 100; //Options: 50, 100, 200, 400, 800, 1000, 1600, 3200
    int pulseWidth = 411; //Options: 69, 118, 215, 411
    int adcRange = 4096; //Options: 2048, 4096, 8192, 16384

    particleSensor.setup(ledBrightness, sampleAverage, ledMode, sampleRate, pulseWidth, adcRange);
  //Configure sensor with these settings

    //Arduino plotter auto-scales annoyingly. To get around this, pre-populate
    //the plotter with 500 of an average reading from the sensor

    //Take an average of IR readings at power up
    const byte avgAmount = 64;
    long baseValue = 0;
    for (byte x = 0 ; x < avgAmount ; x++)
    {
      baseValue += particleSensor.getIR(); //Read the IR value
    }
    baseValue /= avgAmount;

    //Pre-populate the plotter so that the Y scale is close to IR values
    for (int x = 0 ; x < 500 ; x++)
      Serial.println(baseValue);
  }

  void loop()
  {
    Serial.println(particleSensor.getIR()); //Send raw data to plotter
  }
```

示例四：程序运行结果

串口绘图仪显示IR红外值（心跳）。

打开串口绘图仪：


File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino Uno"
Port: "COM10 (Arduino Uno)"
Example4_Hear



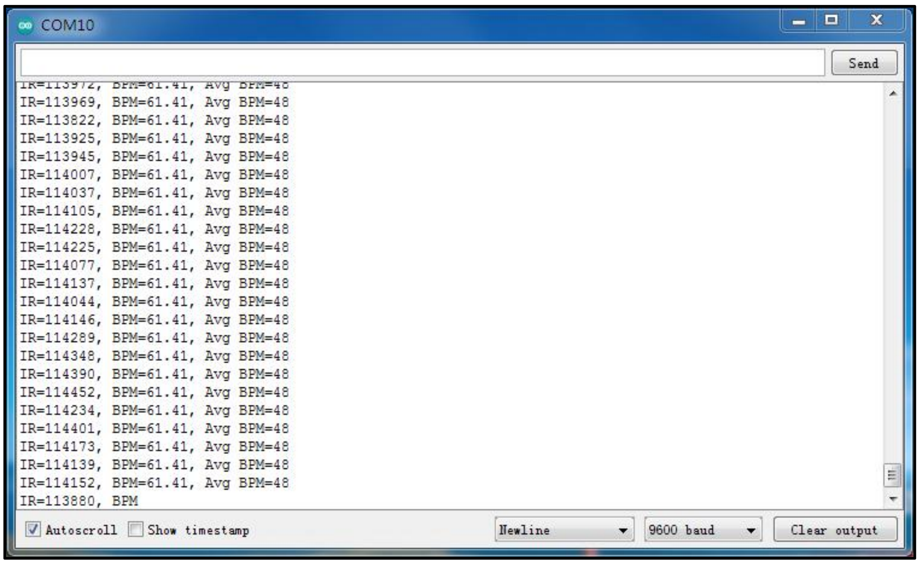
5.2.5. 示例五：心率监测

```
#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;
int beatAvg;

void setup()
{
  Serial.begin(9600);
  Serial.println("Initializing...");
  // Initialize sensor
  if (particleSensor.begin() == false)
  {
    Serial.println("MAX30105 was not found. Please check wiring/power. ");
    while (1);
  }
  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) {
    //We sensed a beat!
    long delta = millis() - lastBeat;
    lastBeat = millis();

    beatsPerMinute = 60 / (delta / 1000.0);

    if (beatsPerMinute < 255 && beatsPerMinute > 20)
    {
      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;
    }
  }

  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();
  }
```

示例五：程序运行结果

串口打印测得值，BPM-心率值（安静状态下的心率在60-100次/分）。



```
IR=113972, BPM=61.41, Avg BPM=48
IR=113969, BPM=61.41, Avg BPM=48
IR=113822, BPM=61.41, Avg BPM=48
IR=113925, BPM=61.41, Avg BPM=48
IR=113945, BPM=61.41, Avg BPM=48
IR=114007, BPM=61.41, Avg BPM=48
IR=114037, BPM=61.41, Avg BPM=48
IR=114105, BPM=61.41, Avg BPM=48
IR=114228, BPM=61.41, Avg BPM=48
IR=114225, BPM=61.41, Avg BPM=48
IR=114077, BPM=61.41, Avg BPM=48
IR=114137, BPM=61.41, Avg BPM=48
IR=114044, BPM=61.41, Avg BPM=48
IR=114146, BPM=61.41, Avg BPM=48
IR=114289, BPM=61.41, Avg BPM=48
IR=114348, BPM=61.41, Avg BPM=48
IR=114390, BPM=61.41, Avg BPM=48
IR=114452, BPM=61.41, Avg BPM=48
IR=114234, BPM=61.41, Avg BPM=48
IR=114401, BPM=61.41, Avg BPM=48
IR=114173, BPM=61.41, Avg BPM=48
IR=114139, BPM=61.41, Avg BPM=48
IR=114152, BPM=61.41, Avg BPM=48
IR=113880, BPM
```

5.2.6. 示例六：血氧饱和度（SpO2）测量

```
#include <Wire.h>
#include "MAX30105.h"
#include "spo2_algorithm.h"

MAX30105 particleSensor;

#define MAX_BRIGHTNESS 255

#if defined(__AVR_ATmega328P__) || defined(__AVR_ATmega168__)
//Arduino Uno doesn't have enough SRAM to store 100 samples of IR led data and red led data in 32-bit format
//To solve this problem, 16-bit MSB of the sampled data will be truncated. Samples become 16-bit data.
uint16_t irBuffer[100]; //infrared LED sensor data
```

```
uint16_t redBuffer[100];   //red LED sensor data
#else
uint32_t irBuffer[100]; //infrared LED sensor data
uint32_t redBuffer[100];   //red LED sensor data
#endif

int32_t bufferLength; //data length
int32_t spo2; //SPO2 value
int8_t validSPO2; //indicator to show if the SPO2 calculation is valid
int32_t heartRate; //heart rate value
int8_t validHeartRate; //indicator to show if the heart rate calculation is valid

byte pulseLED = 11; //Must be on PWM pin
byte readLED = 13; //Blinks with each data read

void setup()
{
  Serial.begin(9600); // initialize serial communication at 115200 bits per second:

  pinMode(pulseLED, OUTPUT);
  pinMode(readLED, OUTPUT);

  // Initialize sensor
  if (!particleSensor.begin()) //Use default I2C port, 400kHz speed
  {
    Serial.println(F("MAX30105 was not found. Please check wiring/power."));
    while (1);
  }

  Serial.println(F("Attach sensor to finger with rubber band. Press any key to start conversion"));
  while (Serial.available() == 0) ; //wait until user presses a key
  Serial.read();

  byte ledBrightness = 60; //Options: 0=Off to 255=50mA
  byte sampleAverage = 4; //Options: 1, 2, 4, 8, 16, 32
  byte ledMode = 2; //Options: 1 = Red only, 2 = Red + IR, 3 = Red + IR + Green
  byte sampleRate = 100; //Options: 50, 100, 200, 400, 800, 1000, 1600, 3200
  int pulseWidth = 411; //Options: 69, 118, 215, 411
  int adcRange = 4096; //Options: 2048, 4096, 8192, 16384
  particleSensor.setup(ledBrightness, sampleAverage, ledMode, sampleRate, pulseWidth, adcRange);
//Configure sensor with these settings
}

void loop()
{
  bufferLength = 100; //buffer length of 100 stores 4 seconds of samples running at 25sps

  //read the first 100 samples, and determine the signal range
  for (byte i = 0 ; i < bufferLength ; i++)
  {
    while (particleSensor.available() == false) //do we have new data?
      particleSensor.check(); //Check the sensor for new data

    redBuffer[i] = particleSensor.getRed();
    irBuffer[i] = particleSensor.getIR();
    particleSensor.nextSample(); //We're finished with this sample so move to next sample

    Serial.print(F("red="));
    Serial.print(redBuffer[i], DEC);
    Serial.print(F(", ir="));
    Serial.println(irBuffer[i], DEC);
  }

  //calculate heart rate and SpO2 after first 100 samples (first 4 seconds of samples)
  maxim_heart_rate_and_oxygen_saturation(irBuffer, bufferLength, redBuffer, &spo2, &validSPO2,
&heartRate, &validHeartRate);

  //Continuously taking samples from MAX30102.  Heart rate and SpO2 are calculated every 1 second
  while (1)
  {
    //dumping the first 25 sets of samples in the memory and shift the last 75 sets of samples to the top
    for (byte i = 25; i < 100; i++)
    {
      redBuffer[i - 25] = redBuffer[i];
      irBuffer[i - 25] = irBuffer[i];
    }

    //take 25 sets of samples before calculating the heart rate.
    for (byte i = 75; i < 100; i++)
    {
      while (particleSensor.available() == false) //do we have new data?
        particleSensor.check(); //Check the sensor for new data

      digitalWrite(readLED, !digitalRead(readLED)); //Blink onboard LED with every data read

      redBuffer[i] = particleSensor.getRed();
      irBuffer[i] = particleSensor.getIR();
      particleSensor.nextSample(); //We're finished with this sample so move to next sample

      //send samples and calculation result to terminal program through UART
      Serial.print(F("red="));
      Serial.print(redBuffer[i], DEC);
      Serial.print(F(", ir="));
      Serial.print(irBuffer[i], DEC);

      Serial.print(F(", HR="));
      Serial.print(heartRate, DEC);

      Serial.print(F(", HRvalid="));
      Serial.print(validHeartRate, DEC);

      Serial.print(F(", SPO2="));
      Serial.print(spo2, DEC);

      Serial.print(F(", SPO2Valid="));
      Serial.println(validSPO2, DEC);
    }

    //After gathering 25 new samples recalculate HR and SP02
    maxim_heart_rate_and_oxygen_saturation(irBuffer, bufferLength, redBuffer, &spo2, &validSPO2,
&heartRate, &validHeartRate);
  }
}
```

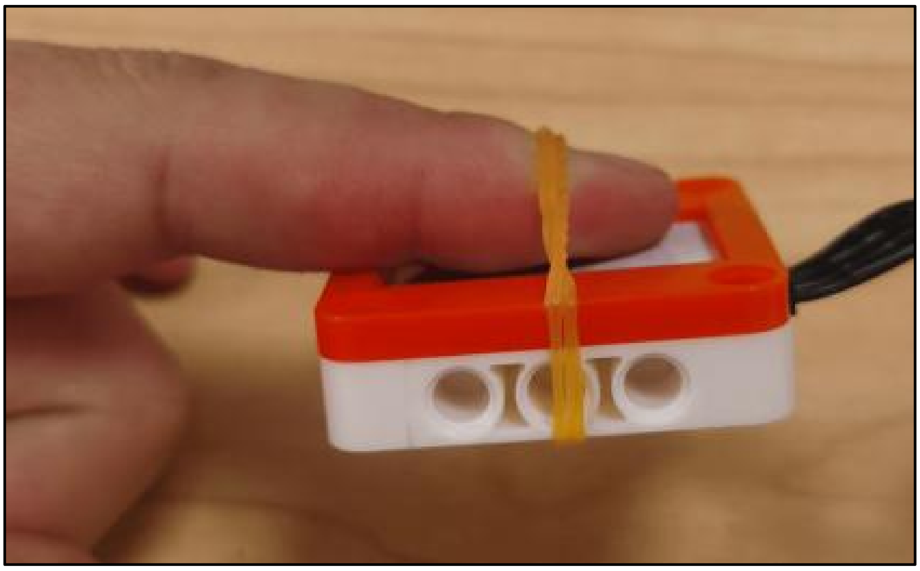
示例六：程序运行结果

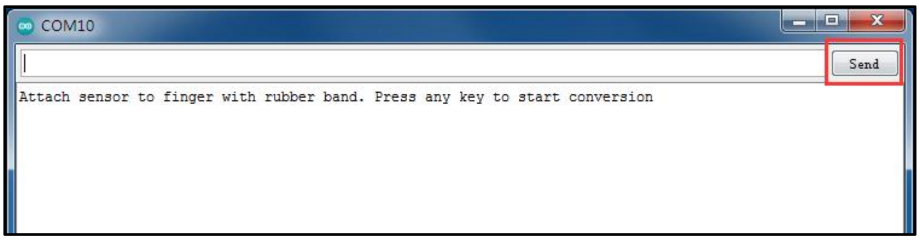
将手指置于传感器上，最好使用橡皮带或其他紧固装置将传感器连接到手指。传感器需要手指持续稳定施加压力。

当您向传感器抵靠手指时，它变化足以使您的手指中的血液流动以不同地流

动，这会导致传感器读数不准确。

打开串口监视器，固定好手指，点击红框“Send”：

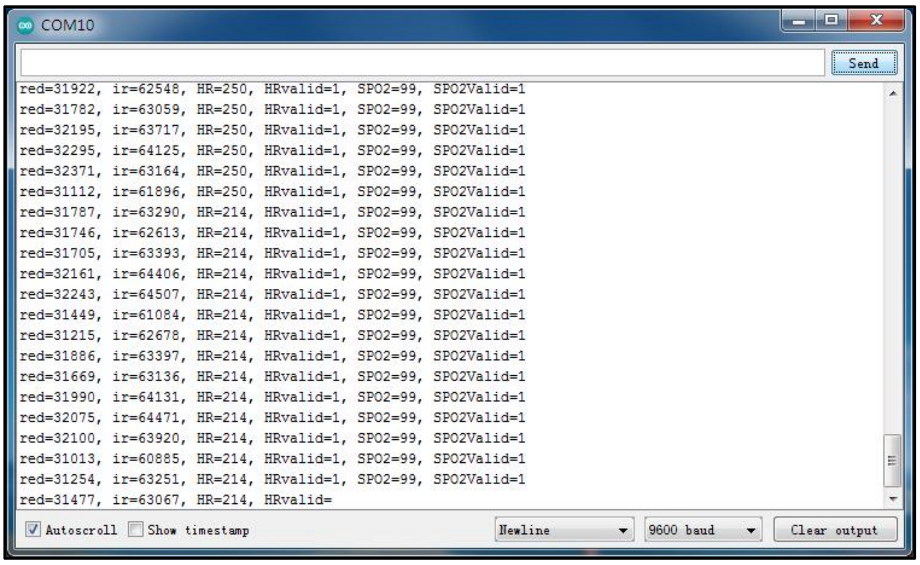




```
Attach sensor to finger with rubber band. Press any key to start conversion
```

串口打印测得值，SPO2-血氧饱和度（正常不低于94%），HR-心率。

⚠ *注意：本产品非医疗器械，测量数据与结果仅供学习参考，不可作为诊疗依据！*



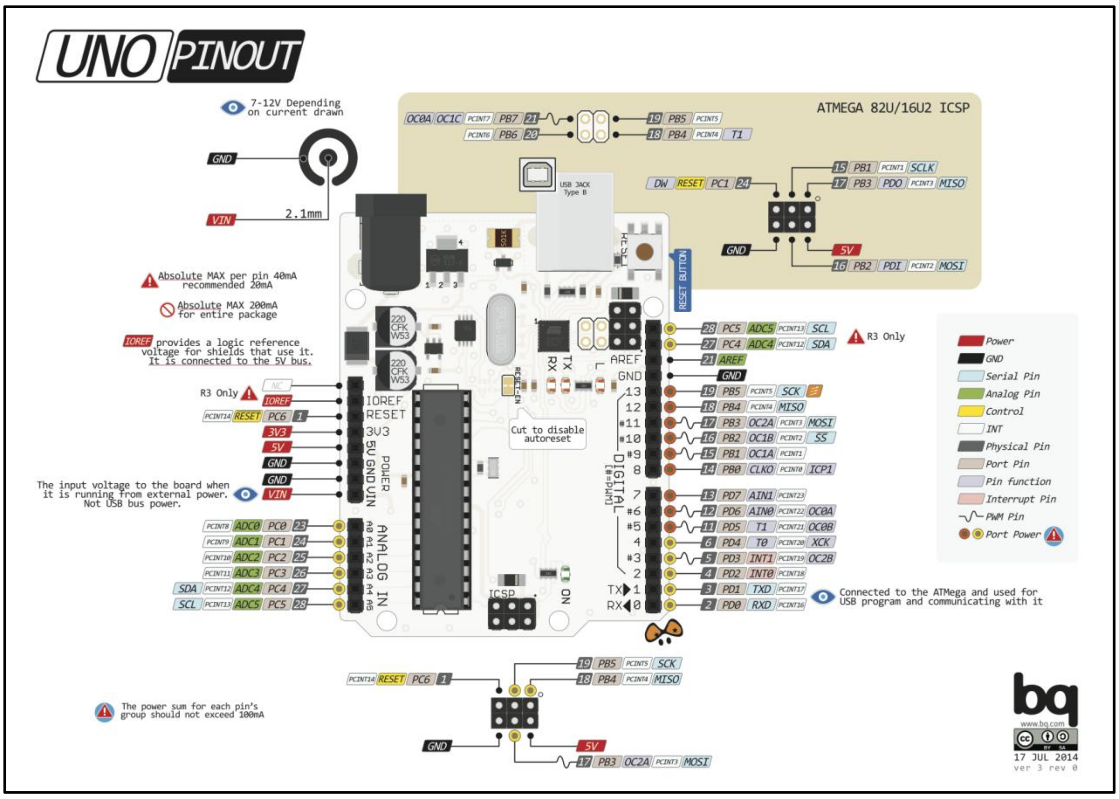
```
red=31922, ir=62548, HR=250, HRvalid=1, SPO2=99, SPO2Valid=1
red=31782, ir=63059, HR=250, HRvalid=1, SPO2=99, SPO2Valid=1
red=32195, ir=63717, HR=250, HRvalid=1, SPO2=99, SPO2Valid=1
red=32295, ir=64125, HR=250, HRvalid=1, SPO2=99, SPO2Valid=1
red=32371, ir=63164, HR=250, HRvalid=1, SPO2=99, SPO2Valid=1
red=31112, ir=61896, HR=250, HRvalid=1, SPO2=99, SPO2Valid=1
red=31787, ir=63290, HR=214, HRvalid=1, SPO2=99, SPO2Valid=1
red=31746, ir=62613, HR=214, HRvalid=1, SPO2=99, SPO2Valid=1
red=31705, ir=63393, HR=214, HRvalid=1, SPO2=99, SPO2Valid=1
red=32161, ir=64406, HR=214, HRvalid=1, SPO2=99, SPO2Valid=1
red=32243, ir=64507, HR=214, HRvalid=1, SPO2=99, SPO2Valid=1
red=31449, ir=61084, HR=214, HRvalid=1, SPO2=99, SPO2Valid=1
red=31215, ir=62678, HR=214, HRvalid=1, SPO2=99, SPO2Valid=1
red=31886, ir=63397, HR=214, HRvalid=1, SPO2=99, SPO2Valid=1
red=31669, ir=63136, HR=214, HRvalid=1, SPO2=99, SPO2Valid=1
red=31990, ir=64131, HR=214, HRvalid=1, SPO2=99, SPO2Valid=1
red=32075, ir=64471, HR=214, HRvalid=1, SPO2=99, SPO2Valid=1
red=32100, ir=63920, HR=214, HRvalid=1, SPO2=99, SPO2Valid=1
red=31013, ir=60885, HR=214, HRvalid=1, SPO2=99, SPO2Valid=1
red=31254, ir=63251, HR=214, HRvalid=1, SPO2=99, SPO2Valid=1
red=31477, ir=63067, HR=214, HRvalid=
```

[✓] Autoscroll [ ] Show timestamp

实际测试结果心率值和示例五中值有很大不同，可能由于算法不同或者单位不同，具体原因不明。

6. 附录

6.1. 附录1-UNO接口说明



UNO PINOUT
7-12V Depending on current drawn
GND
VIN
2.1mm
ATMEGA 82U/16U2 ICSP
Absolute MAX per pin 40mA recommended 20mA
Absolute MAX 200mA for entire package
IOREF provides a logic reference voltage for shields that use it. It is connected to the 5V bus.
R3 Only
The input voltage to the board when it is running from external power. Not USB bus power.
IOREF
RESET
3V3
5V
GND
VIN
POWER
ANALOG IN
A0 A1 A2 A3 A4 A5
AREF
GND
DIGITAL (PWM~)
TX
RX
ICSP
ON
RESET BUTTON
Cut to disable autoreset
Connected to the ATMega and used for USB program and communicating with it
The power sum for each pin's group should not exceed 100mA
Power
GND
Serial Pin
Analog Pin
Control
INT
Physical Pin
Port Pin
Pin function
Interrupt Pin
PWM Pin
Port Power
bq
www.bq.com
17 JUL 2014
ver 3 rev 0

注：UNO官方版本和兼容版本大部分功能都相同

6.2. 附录2-Arduino如何导入库?

教程中有些需要使用库；如何将其导入到自己的Arduino IDE编译器中?

跳转网页查看视频教程：点击跳转。

6.3. 附录3-Mind+如何导入库?

教程中有些需要使用库；如何将其导入到的编译器中?

跳转网页查看教程：点击跳转。

6.4. 附录4-Mixly如何导入库?

教程中有些需要使用库；如何将其导入到的编译器中?

跳转网页查看教程：点击跳转。

6.5. 附录5-MakeCode如何导入扩展?

教程中有些需要使用扩展；如何将其导入到的编译器中?

跳转网页查看教程：点击跳转。

7. 联系我们

YFROBOT网站：www.yfrobot.com / www.yfrobot.com.cn

手机：17696701116（微信/QQ同号）

微信公众号：YFRobotStudio

QQ群：243067479

邮件：yfrobot@qq.com

技术微信　　　　微信公众号