

Name:

Date:

Class:

Programs_for_microcontroller

//program 1

//Example 5 of the SparkFun library to measure the heart rate

/*

Optical Heart Rate Detection (PBA Algorithm) using the MAX30105 Breakout

https://github.com/sparkfun/MAX30105_Breakout

This is a demo to show the reading of heart rate or beats per minute (BPM) using a Peripheral Beat Amplitude (PBA) algorithm.

It is best to attach the sensor to your finger using a rubber band or other tightening device. Humans are generally bad at applying constant pressure to a thing. When you press your finger against the sensor it varies enough to cause the blood in your finger to flow differently, which causes the sensor readings to go wonky.

Hardware Connections (Breadboard to Arduino):

-5V = 5V (3.3V is allowed)

-GND = GND

-SDA = A4 (or SDA)

-SCL = A5 (or SCL)

-INT = Not connected

The MAX30105 Breakout can handle 5V or 3.3V I2C logic. We recommend powering the board with 5V

Name:

Date:

Class:

```
but it will also run at 3.3V.

*/

#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(115200);

  Serial.println("Initializing...");

  // Initialize sensor
```

Name:

Date:

Class:

```
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);
}

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();
    }
}
```

Name:

Date:

Class:

```
beatsPerMinute = 60 / (delta / 1000.0);

if (beatsPerMinute < 255 && beatsPerMinute > 20)
{
    rates[ratesSpot++] = (byte)beatsPerMinute; //Store this reading in the array
    ratesSpot %= 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=");
```

Name:

Date:

Class:

```
Serial.print(beatAvg);
```

```
if (irValue < 50000)
```

```
    Serial.print(" No finger?");
```

```
Serial.println();
```

```
}
```

```
// program 2
```

```
//measure oxygen level and heart rate
```

```
/*This demo shows heart rate and SPO2 levels.
```

It is best to attach the sensor to your finger using a rubber band or other tightening device. Humans are generally bad at applying constant pressure to a thing. When you press your finger against the sensor it varies enough to cause the blood in your finger to flow differently, which causes the sensor readings to go wonky.

Hardware Connections (to Arduino):

-5V = 5V (3.3V is allowed)

-GND = GND

-SDA = A4 (or SDA)

-SCL = A5 (or SCL)

-INT = Not connected

The MAX30105 Breakout can handle 5V or 3.3V I2C logic. We recommend powering the board with 5V, but it will also run at 3.3V.

Name:

Date:

Class:

```
*/  
  
#include <Wire.h>  
  
#include "MAX30105.h"  
  
#include "spo2_algorithm.h"  
  
  
MAX30105 particleSensor;  
  
  
#if defined(__AVR_ATmega328P__) || defined(__AVR_ATmega168__)  
  
uint16_t irBuffer[100];  
  
uint16_t redBuffer[100];  
  
#else  
  
uint32_t irBuffer[100];  
  
uint32_t redBuffer[100];  
  
#endif  
  
  
int32_t bufferLength;  
  
int32_t spo2;  
  
int8_t validSPO2;
```

Name:

Date:

Class:

```
int32_t heartRate;

int8_t validHeartRate;


void setup() {

  Serial.begin(115200);


  // Initialize sensor

  if (!particleSensor.begin(Wire, I2C_SPEED_FAST)) {

    Serial.println(F("MAX30105 was not found. Please check wiring/power."));

    while (1);

  }


  byte ledBrightness = 60;

  byte sampleAverage = 4;

  byte ledMode = 2;

  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
```

Class:

Name:

Date:

Class:

```
maxim_heart_rate_and_oxygen_saturation(irBuffer, bufferLength, redBuffer, &spo2, &validSPO2,
&heartRate, &validHeartRate);
```

```
int lastHeartRate = 0;
```

```
int lastSpO2 = 0;
```

```
while (1) {
```

```
    // Dump 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)
```

```
            particleSensor.check();
```

```
        redBuffer[i] = particleSensor.getRed();
```

Name:

Date:

Class:

```
    irBuffer[i] = particleSensor.getIR();

    particleSensor.nextSample();

}

// After gathering 25 new samples recalculate HR and SpO2

    maxim_heart_rate_and_oxygen_saturation(irBuffer, bufferLength, redBuffer, &spo2, &validSPO2,
    &heartRate, &validHeartRate);


// Check if the values are valid

if (validHeartRate && heartRate >= 60 && heartRate <= 120) {

    lastHeartRate = heartRate;

}

if (validSPO2 && spo2 >= 89 && spo2 <= 100) {

    lastSpO2 = spo2;

}


// Print the last valid values

Serial.print("HR=");

Serial.print(lastHeartRate);
```

Name:

Date:

Class:

```
Serial.print(", SPO2=");

Serial.println(lastSpO2);

}

}

// program3:

//Capture and plot Electrocardiogram Signal

void setup() {

Serial.begin(9600);
pinMode(10,INPUT);
pinMode(11,INPUT);

}

void loop() {

if((digitalRead(10)==1)||((digitalRead(11)==1))

{
Serial.println(":D");
}
else{
Serial.println(analogRead(A0));
}
delay(10);

}
```