

#include <Wire.h>

#include "MAX30100\_PulseOximeter.h"

#define REPORTING\_PERIOD\_MS 1000

PulseOximeter pox;

uint32\_t tsLastReport = 0;

void onBeatDetected()

{

Serial.println("Beat!");

}

void setup()

{

Serial.begin(115200);

Serial.print("Initializing pulse oximeter..");

// Initialize the PulseOximeter instance

// Failures are generally due to an improper I2C wiring, missing power supply

// or wrong target chip

if (!pox.begin()) {

Serial.println("FAILED");

for(;;);

} else {

Serial.println("SUCCESS");

}

pox.setIRLedCurrent(MAX30100\_LED\_CURR\_7\_6MA);

// Register a callback for the beat detection

pox.setOnBeatDetectedCallback(onBeatDetected);

}

void loop()

{

// Make sure to call update as fast as possible

pox.update();

if (millis() - tsLastReport > REPORTING\_PERIOD\_MS) {

Serial.print("Heart rate:");

Serial.print(pox.getHeartRate());

Serial.print("bpm / SpO2:");

Serial.print(pox.getSpO2());

Serial.println("%");

tsLastReport = millis();

}

}

2.



Code:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

This is a library example for the MLX90614 Temp Sensor

Designed specifically to work with the MLX90614 sensors in the

adafruit shop

----> https://www.adafruit.com/products/1747 3V version

----> https://www.adafruit.com/products/1748 5V version

These sensors use I2C to communicate, 2 pins are required to

interface

Adafruit invests time and resources providing this open source code,

please support Adafruit and open-source hardware by purchasing

products from Adafruit!

Written by Limor Fried/Ladyada for Adafruit Industries.

BSD license, all text above must be included in any redistribution

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <Adafruit\_MLX90614.h>

Adafruit\_MLX90614 mlx = Adafruit\_MLX90614();

void setup() {

Serial.begin(9600);

Serial.println("Adafruit MLX90614 test");

if (!mlx.begin()) {

Serial.println("Error connecting to MLX sensor. Check wiring.");

while (1);

};

}

void loop() {

Serial.print("Ambient = "); Serial.print(mlx.readAmbientTempC());

Serial.print("\*C\tObject = "); Serial.print(mlx.readObjectTempC()); Serial.println("\*C");

Serial.print("Ambient = "); Serial.print(mlx.readAmbientTempF());

Serial.print("\*F\tObject = "); Serial.print(mlx.readObjectTempF()); Serial.println("\*F");

Serial.println();

delay(500);

}

3.



source : <https://how2electronics.com/ecg-monitoring-with-ad8232-ecg-sensor-arduino/>

code:

void setup() {

// initialize the serial communication:

Serial.begin(9600);

pinMode(10, INPUT); // Setup for leads off detection LO +

pinMode(11, INPUT); // Setup for leads off detection LO -

}

void loop() {

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

Serial.println('!');

}

else{

// send the value of analog input 0:

Serial.println(analogRead(A0));

}

//Wait for a bit to keep serial data from saturating

delay(1);

}

Link:

<https://learn.sparkfun.com/tutorials/ad8232-heart-rate-monitor-hookup-guide/all>

4.Blood pressure machine

[](https://www.daraz.com.bd/products/digital-wrist-watch-blood-pressure-monitor-i132674582-s1053840382.html?search=1)

Modify for Arduino compatible Link:

<https://www.youtube.com/watch?v=5zRlYQUBFiw>

https://github.com/carlerickrowan/arduino\_blood\_pressure

code :

#include <Wire.h>

uint32\_t current\_millis;

uint32\_t prev\_millis;

volatile byte XEEPROM[256] = {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x9c, 0xb0, 0xc4, 0x54, 0xf0, 0x69, 0xca, 0xb3, 0x28, 0x84, 0xa5, 0xc1, 0x80, 0x53, 0x69, 0x0c, 0x38, 0x4f, 0x0c, 0x74, 0xa1, 0x5b, 0x8c, 0x71, 0x34, 0x30, 0x38, 0x30, 0x30, 0x38, 0x31, 0x35, 0x33, 0x31, 0x30, 0x35, 0x00, 0x12, 0x5a, 0x1e, 0xbd, 0x9e, 0x00, 0x00, 0x02, 0x82, 0x30, 0xaa, 0x10, 0xb7, 0x7f, 0x8d, 0x0a, 0xcb, 0x2e, 0xc1, 0xeb, 0x52, 0x20, 0xdc, 0x00, 0x03, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x97, 0x55, 0x57, 0xfc, 0x00, 0x00, 0x00, 0x00, 0x47, 0x4d, 0x54, 0x00, 0x42, 0x53, 0x54, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0a, 0x05, 0x00, 0x02, 0x03, 0x05, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xc4, 0xff, 0xff, 0xff, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0a, 0x00, 0x00, 0x00, 0x02, 0x00, 0x00, 0x00, 0x68, 0xaa, 0xdd, 0x02, 0x46, 0x45, 0x8f, 0x8f, 0x78, 0x56, 0xbb, 0x69, 0x01, 0x04, 0x46, 0x45, 0x8f, 0x8f, 0x78, 0x56, 0xbb, 0x69, 0x01, 0x04, 0x46, 0x45, 0x8f, 0x8f, 0x78, 0x56, 0xbb, 0x69, 0x12, 0x03, 0x46, 0x45, 0x8f, 0x8f, 0x78, 0x56, 0xbb, 0x69, 0x02, 0x02, 0x46, 0x45, 0x8f, 0x8f, 0x78, 0x56, 0xbb, 0x69, 0x12, 0x02, 0x84, 0x79, 0x80, 0x00, 0x00, 0x00, 0x16, 0x01, 0x00, 0x00, 0x00 };

volatile byte rQ, rD;

volatile int data\_count;

volatile int wrote;

volatile uint16\_t count;

volatile uint8\_t sys, dia, hr;

void setup()

{

Serial.begin(115200);

Wire.begin(0x50);

Wire.onReceive(receiveEvent);

Wire.onRequest(requestEvent);

uint16\_t counter = 0;

while (counter <= 256)

{

XEEPROM[counter] = 0x00;

counter++;

}

}

void loop()

{

current\_millis = millis();

if (current\_millis - prev\_millis > 5000)

{

prev\_millis = current\_millis;

if (count == 0)

{

Serial.print(":::");

Serial.print(data\_count);

Serial.print(" count--->");

Serial.println(count);

}

else if (count == 35)

{

Serial.print("Blood Presure Data: ");

Serial.print(sys);

Serial.print("/");

Serial.println(dia);

Serial.println("");

}

else

{

Serial.println("error");

}

count = 0;

// uint16\_t data\_written = 0;

// while (data\_written < 256)

// {

// Serial.print(data\_written);

// Serial.print(": ");

// Serial.println(XEEPROM[data\_written]);

// data\_written++;

// }

}

}

void requestEvent()

{

// // Wire.receive();

// if (data\_count == 1)

// {

// Wire.write(XEEPROM[rQ]);

// }

// // rQ++;

}

void receiveEvent(int iData)

{

data\_count = iData;

if ( iData > 0 )

{

while ( iData-- )

{

rQ = Wire.read();

Serial.println(rQ);

count++;

if (count == 28)

{

sys = rQ;

}

if (count == 29)

{

dia = rQ;

}

if (count == 30)

{

hr = rQ;

}

}

}

// if (data\_count == 1)

// {

// rQ = Wire.read();

// }

// else if (data\_count == 2)

// {

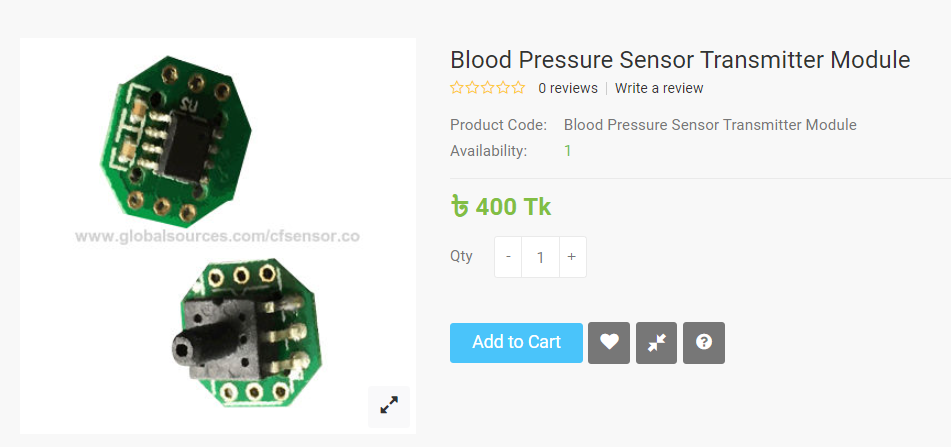
// rQ = Wire.read();

// rD = Wire.read();

// XEEPROM[rQ] = rD;

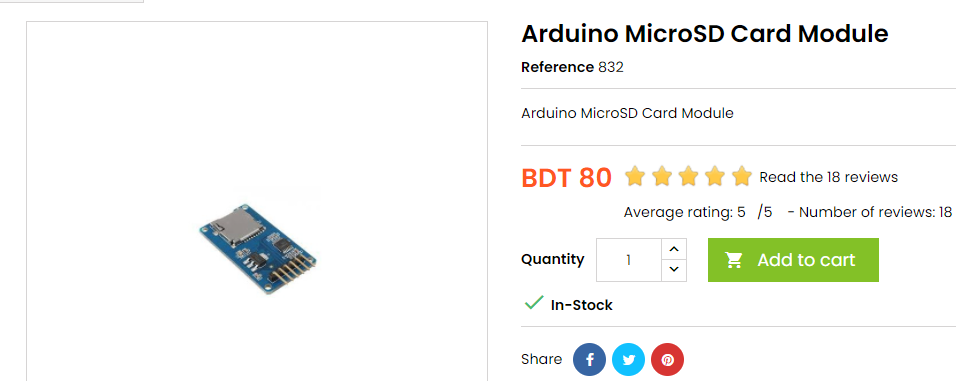
// }

}

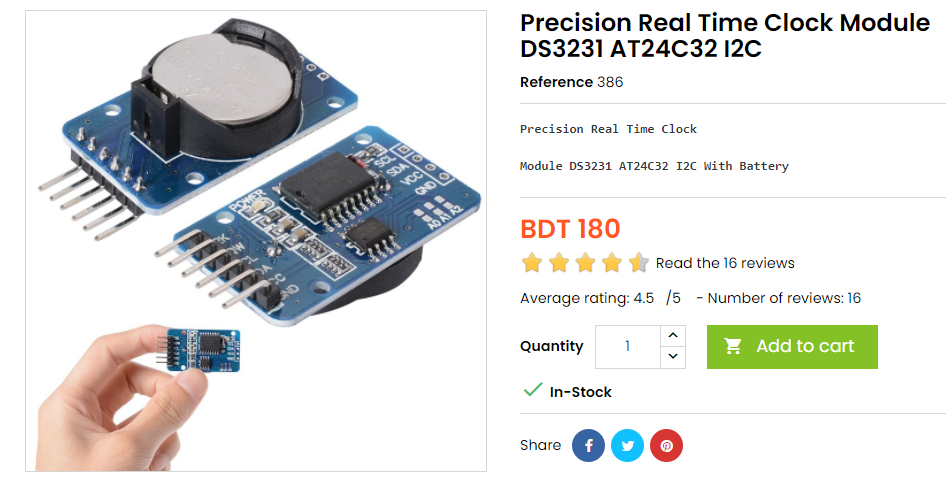
alternate [](http://tspares-bd.com/index.php?route=product/product&product_id=1700)

[](https://it.aliexpress.com/item/4000119040996.html?spm=a2g0o.productlist.0.0.102f20e4c4pFnS&algo_pvid=137066fd-d23a-4651-9cc9-7af6a2bbad8a&algo_exp_id=137066fd-d23a-4651-9cc9-7af6a2bbad8a-0)

5.



<https://create.arduino.cc/projecthub/electropeak/sd-card-module-with-arduino-how-to-read-write-data-37f390>

6. 

Code:

#include <DS3231\_Simple.h>

DS3231\_Simple Clock;

void setup() {

Serial.begin(9600);

Clock.begin();

Serial.println();

Serial.println();

}

void loop()

{

// Create a variable to hold the data

DateTime MyDateAndTime;

// Ask the clock for the data.

MyDateAndTime = Clock.read();

// And use it

Serial.print("Hour: "); Serial.println(MyDateAndTime.Hour);

Serial.print("Minute: "); Serial.println(MyDateAndTime.Minute);

Serial.print("Second: "); Serial.println(MyDateAndTime.Second);

Serial.print("Year: "); Serial.println(MyDateAndTime.Year);

Serial.print("Month: "); Serial.println(MyDateAndTime.Month);

Serial.print("Day: "); Serial.println(MyDateAndTime.Day);

Serial.print("End Of Program (RESET to run again)");

while(1);

}