

## Измеритель пульса на базе платы Arduino

*Кизянов Антон Олегович*

*Приамурский государственный университет имени Шолом-Алейхема*

*Студент*

### Аннотация

В данной статье описан процесс создания макета считывания пульса человека на базе Arduino. Для создания используется плата Arduino, зуммер и датчик пульса MAX30102. Созданный макет позволяет узнать пульс человека, причем каждое биение сопровождается сигналом зуммера и анимацией на OLED дисплее.

**Ключевые слова:** Arduino, MAX30102

### Arduino-based heart rate meter

*Kizyanov Anton Olegovich*

*Sholom-Aleichem Priamursky State University*

*student*

### Abstract

This article describes the process of creating a layout for reading a human heart rate based on Arduino. The Arduino board, the buzzer, and the MAX30102 heart rate sensor are used to create this. The created layout allows you to know the pulse of a person, and each beating is accompanied by a buzzer signal and animation on the OLED display.

**Keywords:** Arduino, MAX30102

Для измерения пульса будет использоваться датчик MAX30102, для отображения OLED дисплей и для озвучивания зуммер.

MAX30102 представляет собой интегрированный пульсо-ксиметрия и сердечного ритма монитор биосенсора модуль. Он включает в себя внутренние светодиоды, фотоприемники, оптические элементы и с помощью библиотеки Sparkfun будет легко измерить эти параметры, еще модуль имеет встроенный датчик температуры, и он на самом деле очень точный и стабильный, но он не предназначен для измерения температуры тела.

Цель исследования – создать систему считывания пульса человека на базе Arduino.

Ранее этим вопросом интересовались О.Н. Головченко, Р.Г. Оганян развивали тему «Организация обработки сигнала с пульсометра» [1] в которой рассматриваются особенности обработки сигнала с пульсометра. Обработка сигнала реализуется на базе пакета прикладных программ

MatLAB. Также в настоящей статье организована обработка и отображение сигналов с пульсометра. Реализация обработки сигналов с пульсометра, позволит применять данную технологию в медицине, для измерения сердечного ритма по средствам монитора ЭКГ. Р.Р. Бикмухаметов, Д.В. Ширшова с темой «Применение датчиков и модулей arduino для реализации проектов лабораторных работ по дисциплинам "микроконтроллеры" и "микропроцессорные системы"» [2], а подробнее про наиболее распространенные модули датчиков Ардуино для решения задач лабораторных работ. Указывается на преимущество использования макетных плат для сборки моделей. Д.А. Страковский, Е.Е. Симаков опубликовали статью «Анализатор воздуха на платформе arduino» [3] рассказали, что роботы задействованы во многих сферах жизни человека. На предприятиях люди не могут обойтись без них. Роботизированные системы помогают человеку обрабатывать данные, следят за соблюдением техники безопасности, помогают заботиться о здоровье. Данная статья посвящена изучению основ проектирования роботизированных устройств на платформе Arduino Uno. Рассмотрены вопросы о влиянии внешней среды на здоровье человека. Разработанное устройство способно предупреждать человека при возникновении неблагоприятных условий как в домашних условиях, так и на производстве.

Для этого потребуется:

- Плата Arduino
- Макетная плата
- Соединительные провода
- Зуммер
- Дисплей OLED 128x32
- Датчик пульса MAX30102

Схема подключения представлена на рисунке 1.

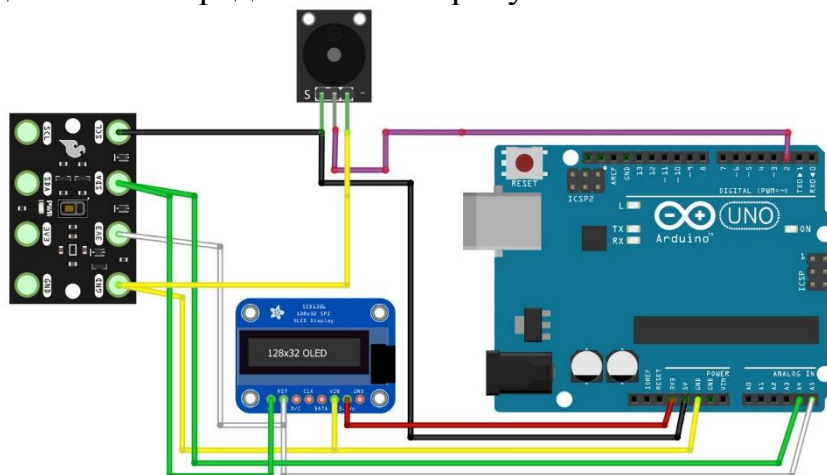


Рис. 1 Схема подключения к плате Arduino

```
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <Wire.h>
#include "MAX30105.h"
#include "heartRate.h"

MAX30105 particleSensorModule;

const byte RATE_SIZE_PIN = 4;
byte rates[RATE_SIZE_PIN];
byte ratePin = 0;
long lastBeatPin = 0;
float beatsPerMinuteCount;
int beatAvgCount;

#define SCREEN_WIDTH_INT 128
#define SCREEN_HEIGHT_INT 32
#define OLED_RESET_INT -1

Adafruit_SSD1306 displaymodule(SCREEN_WIDTH_INT, SCREEN_HEIGHT_INT,
&Wire, OLED_RESET_INT);

static const unsigned char PROGMEM logo2_bmp[] =
{ 3, 192, 240, 6, 113, 140, 12, 27, 6, 24, 14, 2, 16, 12, 3, 16,
  4, 1, 16, 4, 1, 16, 64, 1, 16, 64, 1, 16, 192, 3, 8, 136,
  2, 8, 184, 4, 255, 55, 8, 1, 48, 24, 1, 144, 48, 0, 192, 96,
  0, 96, 192, 0, 49, 128, 0, 27, 0, 0, 14, 0, 0, 4, 0,
};

static const unsigned char PROGMEM logo3_bmp[] =
{ 1, 240, 15, 128, 6, 28, 56, 96, 24, 6, 96, 24, 16, 1, 128, 8,
  32, 1, 128, 4, 64, 0, 0, 2, 64, 0, 0, 2, 192, 0, 8, 3,
  128, 0, 8, 1, 128, 0, 24, 1, 128, 0, 28, 1, 128, 0, 20, 0,
  128, 0, 20, 0, 128, 0, 20, 0, 64, 16, 0x12, 0, 64, 16, 0x12, 0,
  126, 31, 35, 254, 3, 49, 160, 4, 1, 160, 160, 12, 0, 160, 160, 8,
  0, 96, 224, 16, 0, 32, 96, 32, 6, 0, 64, 96, 3, 0, 64, 192,
  1, 128, 1, 128, 0, 192, 3, 0, 0, 96, 6, 0, 0, 48, 12, 0,
  0, 8, 16, 0, 0, 6, 96, 0, 0, 3, 192, 0, 0, 1, 128, 0
};

void setup() {
  displaymodule.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  displaymodule.displaymodule();
  delay(3000);
  // Initialize sensor
  particleSensorModule.begin(Wire, I2C_SPEED_FAST);
  particleSensorModule.setup();
  particleSensorModule.setPulseAmplitudeRed(0x0A);
}

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

  if (irValue > 7000) {
    displaymodule.clearDisplay();
    displaymodule.drawBitmap(5, 5, logo2_bmp, 24, 21, WHITE);
  }
}
```

```
displaymodule.setTextSize(2);
displaymodule.setTextColor(WHITE);
displaymodule.setCursor(50, 0);
displaymodule.println("BPM");
displaymodule.setCursor(50, 18);
displaymodule.println(beatAvgCount);
displaymodule.displaymodule();

if (checkForBeat(irValue) == true)
{
  displaymodule.clearDisplay();
  displaymodule.drawBitmap(0, 0, logo3_bmp, 32, 32, WHITE);
  displaymodule.setTextSize(2);
  displaymodule.setTextColor(WHITE);
  displaymodule.setCursor(50, 0);
  displaymodule.println("BPM");
  displaymodule.setCursor(50, 18);
  displaymodule.println(beatAvgCount);
  displaymodule.displaymodule();
  tone(3, 1000);
  delay(100);
  noTone(3);
  //We sensed a beat!
  long delta = millis() - lastBeatPin;
  lastBeatPin = millis();

  beatsPerMinuteCount = 60 / (delta / 1000.0);

  if (beatsPerMinuteCount < 255 && beatsPerMinuteCount > 20)
  {
    rates[ratePin++] = (byte)beatsPerMinuteCount;
    ratePin %= RATE_SIZE_PIN;
    beatAvgCount = 0;
    for (byte x = 0 ; x < RATE_SIZE_PIN ; x++)
      beatAvgCount += rates[x];
    beatAvgCount /= RATE_SIZE_PIN;
  }
}

if (irValue < 7000) {
  beatAvgCount = 0;
  displaymodule.clearDisplay();
  displaymodule.setTextSize(1);
  displaymodule.setTextColor(WHITE);
  displaymodule.setCursor(30, 5);
  displaymodule.println("Please Place ");
  displaymodule.setCursor(30, 15);
  displaymodule.println("your finger ");
  displaymodule.displaymodule();
  noTone(3);
}
}
```

Результат работы можно увидеть на рисунке 2.

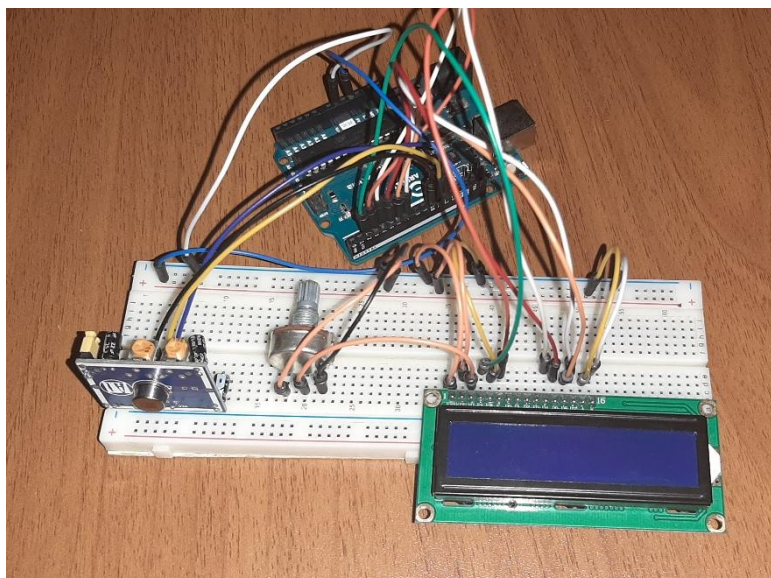


Рис. 2 Схема в собранном состоянии

### Вывод

Результатом статьи стал работающий прототип измерителя пульса с отображением данных на базе Arduino и датчике пульсового оксиметра MAX30102. Схема получилась среднего уровня сложности. Данная схема дает достаточно точный результат, но все равно не может соревноваться с профессиональными приборами.

### Библиографический список

1. Головченко О.Н., Оганян Р.Г. Организация обработки сигнала с пульсометра // Аллея науки. 2018. Т. 3. № 10 (26). С. 966-971. URL: <https://elibrary.ru/item.asp?id=36662678> (Дата обращения: 02.01.2020)
2. Бикмухаметов Р.Р., Ширшова Д.В. Применение датчиков и модулей arduino для реализации проектов лабораторных работ по дисциплинам "микроконтроллеры" и "микропроцессорные системы" // В сборнике: наука сегодня: проблемы и перспективы развития материалы международной научно-практической конференции: в 2 частях. Научный центр «Диспут». 2016. С. 22-24. URL: <https://elibrary.ru/item.asp?id=27588246> (Дата обращения: 02.01.2020)
3. Страковский Д.А., Симаков Е.Е. Анализатор воздуха на платформе arduino // Юный ученый. 2017. № 3 (12). С. 49-56. URL: <https://elibrary.ru/item.asp?id=29463338> (Дата обращения: 02.01.2020)