PULSE OXIMETER PROJECT

ELECTRONIC PROJECT

PRESENTED BY.:- AMRUT WARAD

PUNE DISTRICT EDUCATION ASSOCIATION

PROF.RAMKRISHNA MORE
ARTS, COMMERCE & SCIENCE COLLEGE

AKURDI, PRADHIKARAN PUNE - 411 044.

DEPARTMENT OF ELECTRONICS

CERTIFICATE

This is to certify that,

Mr. Amrut Prabuling Warad of class **S. Y. B.Sc. (CS)** Div. **B** Roll No. **41436** has satisfactorily completed necessary project work in ELECTRONICS during the academic year 2023-2024.

Exam seat no: _____

Practical in charge: _____

Date: / /

H.O.D Dept. of Electronics	
Internal Examinar	
External Examinar	

INDEX

➢ INTRODUCTION :-

➢ **Project Title**: DIY Pulse Oximeter with Arduino Nano and MAX30102 Sensor

➢ Introduction:

A pulse oximeter is a medical device used to measure the oxygen saturation level (SpO2) and heart rate of an individual. It's a crucial tool in monitoring respiratory health, especially in situations like exercising at high altitudes or during respiratory illnesses. This DIY project aims to create a pulse oximeter using readily available components like the Arduino Nano microcontroller and the MAX30102 sensor module.

➢ Components Needed:

1. Arduino Nano board
2. MAX30102 sensor module
3. OLED display (optional but recommended for displaying SpO2 and heart rate)
4. Connecting wires (jumping wire)
5. Breadboard (optional but recommended for prototyping)

➢ Project Overview:

The MAX30102 sensor module integrates a photodetector and an LED for non-invasive measurement of blood oxygen saturation and heart rate. The module emits light at two different wavelengths (typically red and infrared), and by measuring the light absorption, it can determine the oxygen saturation level in the blood.

- **Project Steps:**

1. **Setup Arduino Nano:** Begin by setting up the Arduino Nano board and installing the necessary development environment (usually the Arduino IDE).

2. **Connect MAX30102 Sensor:** Wire the MAX30102 sensor module to the Arduino Nano according to the pinout specifications. Typically, the sensor module requires connections for power (VCC and GND) and communication (SDA and SCL).

3. **Code Development:** Write the Arduino code to initialize the sensor and continuously read data from it. Use libraries and example code provided by the manufacturer or community to facilitate sensor communication.

4. **Data Processing:** Process the sensor data to extract the heart rate and SpO2 values. This may involve filtering the raw sensor data, performing calculations, and applying algorithms to accurately determine the physiological parameters.

5. **Display Output:** If using an OLED display, integrate code to output the calculated heart rate and SpO2 values onto the display in a user-friendly format.

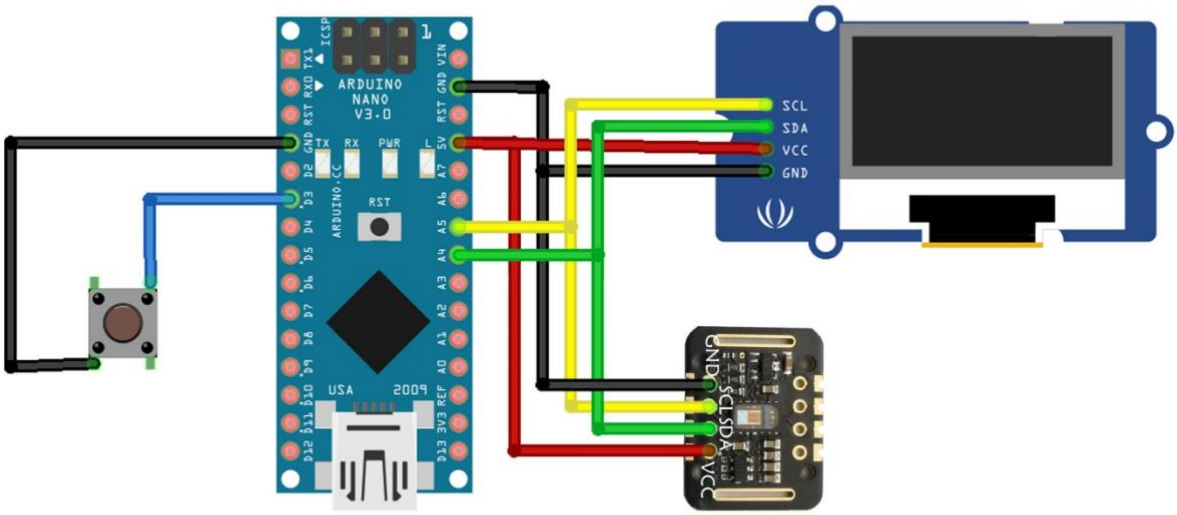
6. **Testing and Calibration:** Test the pulse oximeter by placing it on a finger and comparing the readings to those obtained from a commercial pulse oximeter for accuracy. Calibration may be necessary to ensure reliable measurements.

7. **Enclosure Design (Optional):** Design and build an enclosure for the Arduino Nanno, MAX30102 sensor, and OLED display to create a compact and portable pulse oximeter device.

- **Conclusion:**

By completing this project, you'll have built a DIY pulse oximeter using an Arduino Nano and the MAX30102 sensor module. This project not only serves as a valuable learning experience in electronics and programming but also provides a practical application in healthcare monitoring and diagnostics.

➢ CIRCUIT DIAGRAM :-



➢ WORKING OF DIAGRAM:-

1. **MAX30102 Sensor:** The MAX30102 sensor is a key component of the pulse oximeter. It consists of an integrated red and infrared LED, a photodetector, and signal processing circuitry. The red and infrared LEDs emit light into the skin, and the photodetector measures the amount of light absorbed by oxygenated and deoxygenated hemoglobin in the blood vessels.

2. **Arduino Nano Board:** The Arduino Nano board serves as the central processing unit of the pulse oximeter. It controls the operation of the MAX30102 sensor, reads the sensor data, performs signal processing algorithms, and displays the results.

3. **Power Supply:** The pulse oximeter requires a stable power supply to operate. This can be provided by connecting the Arduino Nano board to a USB power source or a battery.

4. **Data Processing:** The Arduino Nano board reads the raw sensor data from the MAX30102 sensor, which includes the intensity of the red and infrared light reflected from the skin. Signal processing algorithms are then applied to the raw data to calculate the heart rate and oxygen saturation level (SpO2) of the individual.

5. **Display Output:** The calculated heart rate and SpO2 values are typically displayed on an output device such as an OLED display or sent to a computer for visualization and analysis. The display provides real-time feedback to the user and allows for easy monitoring of vital signs.

6. **User Interaction:** Depending on the design, the pulse oximeter may include buttons or other user interface elements for controlling the

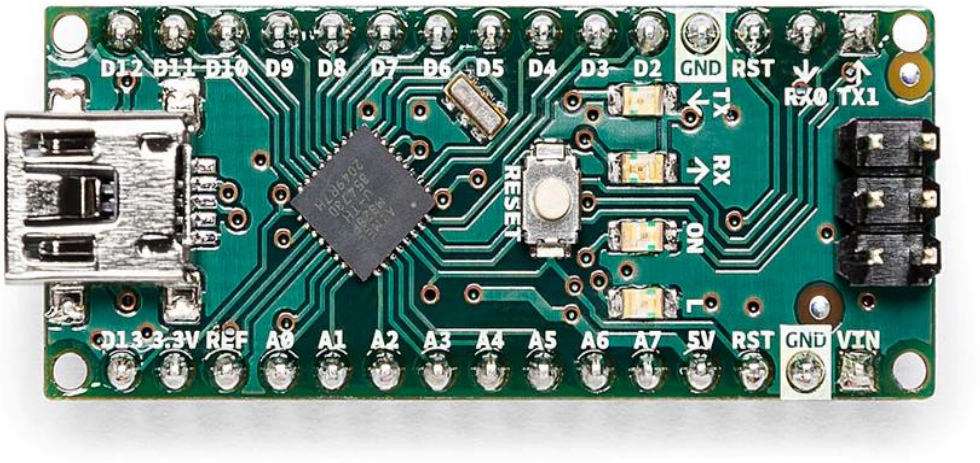
operation of the device, such as starting or stopping measurements, adjusting settings, or navigating menus.

7. **Feedback Mechanism:** The pulse oximeter may incorporate feedback mechanisms to alert the user of abnormal readings or errors, such as audible alarms, visual indicators, or haptic feedback.

   Overall, the working diagram of a pulse oximeter using an Arduino Nano and the MAX30102 sensor involves sensor data acquisition, signal processing, calculation of vital signs, display of results, and user interaction. This integrated system allows for non-invasive monitoring of oxygen saturation and heart rate, making it a valuable tool for healthcare, fitness, and research applications.

➢ COMPONENTS :-

1. Arduino NANO :-



The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328 (Arduino Nano 3.x). It has more or less the same functionality of the Arduino Duemilanove, but in a different package. It lacks only a DC power jack, and works with a Mini-B USB cable instead of a standard one

2. 0.96" I2C OLED :-

The 0.96 inch OLED Display Module – SPI/I2C – 128×64 – 7 Pin (Blue) are one of the most attractive displays available for a microcontroller.

OLED's are the future of displays, as they possess some of the greatest advantages over both conventional display technologies of LCD's and LED's. The most attractive thing about using the *OLED displays* is that they do not need a back-light like conventional LCD/LED screens. The organic material itself has a property known as Electroluminescence (EL), which causes the material to "glow" when stimulated by a current or an electric field. Best energy saving displays ever!!!

3. Max30102 Heart Rate Sensor :-

The MAX30102 heart rate and oxygen sensor operates based on the principle of light absorption by blood in tissues and blood vessels. MAX30102 utilizes infrared (IR) and red light reflection technology to measure the oxygen saturation (SpO2) and heart rate (HR) of the user.

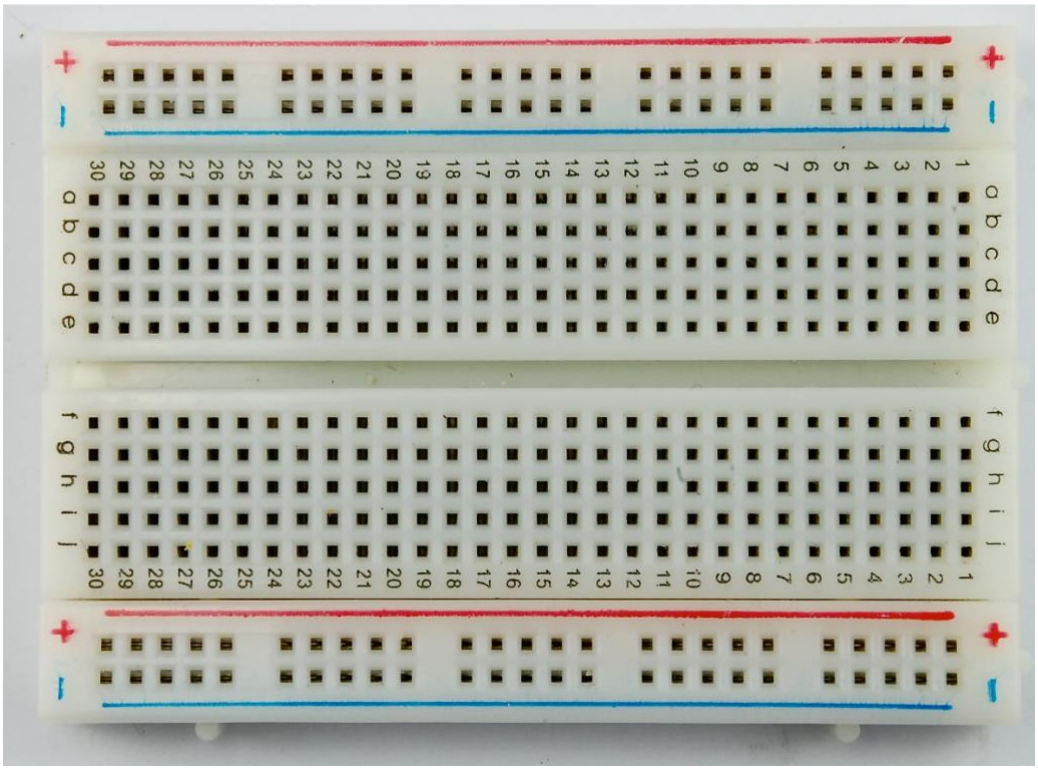
The sensor incorporates infrared and red LEDs, along with a photonic filter, to generate appropriate light that penetrates the skin and red-colored tissues. The infrared LED penetrates deeper into the skin, while the red LED penetrates further away. When the light is projected through the skin, it encounters reflection from the flowing blood in the blood vessels beneath the skin.The sensor employs photodiodes to measure the amount of light reflected by the blood.

4. Jumper wires :-

Jumper wires are used for making connections between items on your breadboard and your Arduino's header pins. Use them to wire up all your circuits

Jumper wires are simply wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering. Jumper wires are typically used with breadboard and other prototyping tools in order to make it easy to change a circuit as needed. Fairly simple. In fact, it doesn't get much more basic than jumper wires.

5. Breadboard :-



Certainly! A breadboard is a fundamental tool used in electronics prototyping, including with Arduino projects. Here's some information about breadboards and how they're used with Arduino:

A breadboard is a rectangular plastic board with a grid of small holes on its surface. These holes are interconnected internally in a specific pattern, usually referred to as tie points or nodes. Breadboards are used to create temporary circuits without the need for soldering. Components such as resistors, LEDs, wires, and sensors can be inserted into the holes to quickly build and test circuits.

➢ ADVANTAGES :-

1. **Affordability:** Arduino Uno boards are relatively inexpensive, making them accessible for hobbyists and small-scale projects. This affordability extends to the components required for building a pulse oximeter, such as sensors and display modules.

2. **Open Source and Customization:** Arduino is an open-source platform, which means the code and designs are freely available for modification and customization. This allows individuals to tailor the pulse oximeter to their specific needs, whether it's adding additional features or adapting it for different use cases.

3. **Educational Tool:** Building a pulse oximeter with Arduino can serve as an educational project, teaching users about electronics, programming, and medical monitoring principles. It provides hands-on experience with sensors, data processing, and display output.

4. **Prototyping and Rapid Development:** Arduino's simplicity and ease of use make it ideal for rapid prototyping. With the Arduino platform, users can quickly assemble and test different iterations of the pulse oximeter design, facilitating the development process.

5. **Integration with Other Systems:** Arduino Uno boards can easily interface with other systems and devices through various communication protocols such as serial communication, I2C, or SPI. This allows the pulse oximeter to be integrated into larger projects or connected to external displays, data loggers, or IoT platforms.

➢ DISADVANTAGES :-

1. **Limited Processing Power:** Arduino Uno boards have limited processing power and memory compared to more advanced microcontrollers or dedicated pulse oximeter devices. This limitation may restrict the complexity of algorithms or real-time processing capabilities, especially for high-resolution sensor data.

2. **Accuracy and Calibration:** Achieving high accuracy and reliability in pulse oximetry measurements requires precise calibration and signal processing techniques. While Arduino-based pulse oximeters can provide useful estimates, they may not match the accuracy of medical-grade devices without careful calibration and validation.

3. **Hardware Constraints:** Arduino Uno boards have limited input/output pins and may lack certain features required for advanced sensor interfacing or signal conditioning. Additional components or external circuitry may be needed to overcome these hardware constraints, increasing complexity and cost.

4. **Power Consumption:** Arduino Uno boards are not optimized for low power consumption, which can be a concern for battery-powered applications or continuous monitoring scenarios. Careful power management strategies may be necessary to prolong battery life or ensure uninterrupted operation.

5. **Regulatory Compliance:** If the pulse oximeter is intended for medical use or commercial distribution, regulatory compliance with relevant standards and guidelines (e.g., FDA regulations) may be required. Ensuring compliance can involve additional testing, documentation, and certification processes, which may add complexity and cost to the project.

➢ APPLICATIONS:-

1. **Healthcare Monitoring:** Pulse oximeters are extensively used in healthcare settings for monitoring patients' oxygen saturation levels (SpO2) and heart rates. Arduino-based pulse oximeters can be employed for non-invasive monitoring of patients in hospitals, clinics, or home healthcare environments

2. **Sports and Fitness:** Athletes and fitness enthusiasts can use pulse oximeters during exercise to monitor their oxygen levels and heart rates. Arduino-based pulse oximeters can be integrated into wearable devices or fitness trackers to provide real-time feedback on performance and recovery.

3. **High Altitude Environments:** Individuals traveling or living in high altitude regions can benefit from pulse oximeters to monitor their oxygen levels and acclimatization status. Arduino-based pulse oximeters can be portable and lightweight, making them suitable for outdoor activities such as hiking, mountaineering, or skiing.

4. **Sleep Apnea Monitoring**: Pulse oximeters are used in sleep studies to diagnose sleep disorders such as sleep apnea. Arduino-based pulse oximeters can be integrated into sleep monitoring systems for at-home sleep studies or continuous monitoring of patients with sleep-related breathing problems.

5. **Remote Patient Monitoring:** Pulse oximeters are integral to remote patient monitoring systems, allowing healthcare providers to remotely monitor patients' vital signs and intervene when necessary. Arduino-

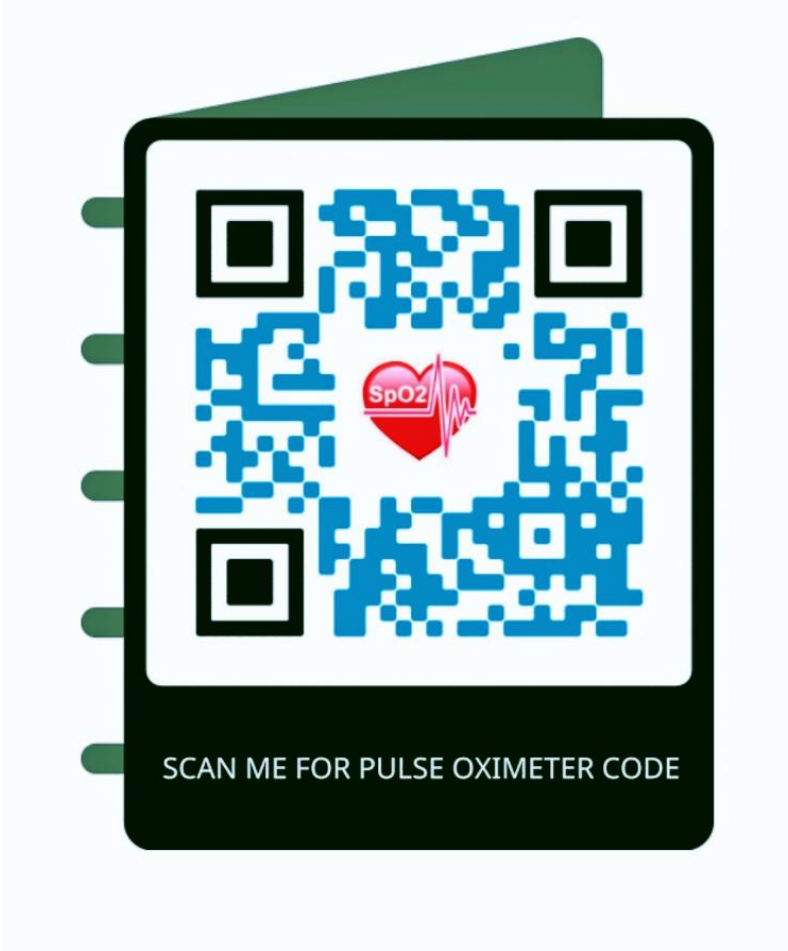
based pulse oximeters can be connected to IoT platforms or mobile apps for remote data transmission and analysis.

6. **Research and Development:** Pulse oximeters are used in research laboratories for studying physiological responses to various stimuli or conditions. Arduino-based pulse oximeters offer a cost-effective and customizable solution for researchers to develop and test new algorithms, sensors, or monitoring techniques.

7. **Education and Training:** Pulse oximeters are valuable educational tools for teaching students about cardiovascular physiology, medical instrumentation, and signal processing. Arduino-based pulse oximeter projects can be incorporated into biomedical engineering courses or workshops to provide hands-on learning experiences.

   Overall, the application of a pulse oximeter built using an Arduino Uno is diverse, spanning healthcare, fitness, research, education, and beyond. Its versatility, affordability, and customization options make it a valuable tool for various users and environments.

➢ SCAN QR FOR PULSE OXIMETER CODE:-




thank
you