A logo of a globe with yellow rings around it

Description automatically generated

**INDIVIDUAL ASSIGNMENT**

**CT101-3-3-IOT**

**INTERNET OF THINGS: CONCEPTS AND APPLICATION**

**APD3F2411CS(CYB)**

**NAME**   **:** Ricky Bosco Rodrigues

**HAND-OUT DATE**  **:**  2nd January 2025

**HAND-IN DATE**  **:**  21st March 2025

**LECTURER** **:**  Kamalanathan A/L Shanmugam

**WEIGHTAGE** **:**  50%

Table of Contents

[Table of Figures 3](#_Toc193224553)

[Introduction 4](#_Toc193224554)

[1.0 Heart Rate Monitoring System Using UNO 5](#_Toc193224555)

[1.2 Components Required 5](#_Toc193224556)

[1.3 System Design 12](#_Toc193224557)

[1.4 Prototype Screenshot 15](#_Toc193224558)

[1.5 Source Code 16](#_Toc193224559)

[1.6 Limitation 20](#_Toc193224560)

[1.7 Future Enhancement 21](#_Toc193224561)

[Conclusion 22](#_Toc193224562)

[References 22](#_Toc193224563)

# Table of Figures

[Figure 1: ATmega238p 6](#_Toc193224535)

[Figure 2: Mini Breadboard 7](#_Toc193224536)

[Figure 3: LCD with I2C 8](#_Toc193224537)

[Figure 4: Jumper Wires 9](#_Toc193224538)

[Figure 5: Supercell Battery 10](#_Toc193224539)

[Figure 6: MAX30100 11](#_Toc193224540)

[Figure 7: Arduino Uno R3 12](#_Toc193224541)

[Figure 8: LCD I2C 13](#_Toc193224542)

[Figure 9: MAX30100 sensor 14](#_Toc193224543)

[Figure 10: Prototype HMS 15](#_Toc193224544)

[Figure 11: Code 1 16](#_Toc193224545)

[Figure 12: Code 2 17](#_Toc193224546)

[Figure 13: Code 3 18](#_Toc193224547)

[Figure 14: Code 4 19](#_Toc193224548)

# Introduction

The Internet of Things (IoT) is the latest technology for interconnecting physical devices and enabling them to communicate and exchange data through the internet. With the use of computer programs and the addition of sensors, the processes can be automated, the experience enhanced for the user, and things made efficient for many industries, including the medical field. Monitoring health through the internet of things can continuously track the vital signs, such as heartbeats, for the detection of anomalies earlier and the enhancement of patient care (Abdulmalek, et al., 2022).

Arduino is one of the favourite open-source platforms for use in IoT projects owing to its flexibility and ease of use. Input and output pins provided by the Arduino Uno board make it suitable for sensor interfaces, display interfaces, and interfaces for various electronic devices. This is thus the best platform for the design of interactive health-monitoring systems where the data can be collected and visualized by the users in real-time (Banzi, 2011).

This project is concerned with the design of the heart rate measurement system utilizing the pulse sensor and the LCD display using the Arduino Uno. Heart rate is sensed by the pulse sensor and provided feedback by the LCD display in real-time. This system is programmed for heart rates from 30 BPM up to 100 BPM and provides visual feedback by the LCD display. This project showcases the feasibility for the application of the use of the IoT for the health monitoring systems and the ability of the Arduino for the design of cheap and easily available health care.

# 1.0 Heart Rate Monitoring System Using UNO

## 1.2 Components Required

|  |  |
| --- | --- |
| No. | Components |
| 1. | Arduino Uno |
| 2. | Mini BreadBoard |
| 3. | LCD Display 16x2 with 12C Module |
| 4. | Jumper Wires |
| 5. | Supercell Heavy Rechargeable Battery |
| 6. | Max30100 Pulse Heart Rate Sensor |

Table 1: Components Required to Built

**Arduino Uno ATmega328p**

****

Figure 1: ATmega238p

The Arduino Uno is one of the microcontrollers available using the microchip ATmega328P. Owing to its flexibility and ease, the Uno is extremely common for prototyping. It features 6 analogue inputs and 14 digital inputs/output pins, alongside the USB for the purpose of programming. This is the main component for the heart rate measurement system. Utilizing the Arduino IDE and the computer programming language C, the Uno processes the data from the pulse sensor and directs the output towards the display for the display of the heart rate in real-time.

**Mini Breadboard**

Close-up of a white electronic device

AI-generated content may be incorrect.

Figure 2: Mini Breadboard

A mini breadboard is also known as a small, temporary mounting platform for electronic circuit prototyping. It is for holding electronic components such as microcontrollers, sensors, and wires through connections made possible by not using solder. In this experiment, the breadboard is required for physically mounting the Arduino Uno onto the MAX30100 pulse sensor, the display screen, and the parts for swift changes and tests for the circuit.

**LCD Display 16x2 with 12C Module**

****

Figure 3: LCD with I2C

The 16x2 LCD is being used for the display of the heart rate value in real-time, providing a friendly interface for the display of the pulse value. With the display being 16 columns by 2 rows, the display is apt for the display of the heart rate value clearly. The communication from the display and the Arduino is provided by the I2C (Inter-Integrated Circuit) module by keeping the pins required minimum and providing communication from the display towards the Arduino through only two data-carrying wires, making the system efficient.

**Jumper Wires**

A group of multicolored wires

AI-generated content may be incorrect.

Figure 4: Jumper Wires

Jumper wires allow the connections from the breadboard elements to the Arduino. These connections are needed for the connection of the pulse sensor, the LCD display, and the other parts to the Arduino Uno. They allow the flexibility and ease needed for the swift changes during the test and prototyping processes, keeping all the parts easily and securely connected.

**Supercell Heavy Rechargeable Battery**

A black and yellow battery

AI-generated content may be incorrect.

Figure 5: Supercell Battery

The Supercell Heavy Rechargeable Battery is the one delivering the energy required for the overall system. Because the Supercell is rechargeable, the added benefit is the capability for the system to run for longer periods using less replacement. This is the component needed for powering the Arduino Uno, the LCD display, and the pulse sensor for portable operation, making the heart rate measurement system run independently from continuous energy sources.

**Max30100 Pulse Heart Rate Sensor**

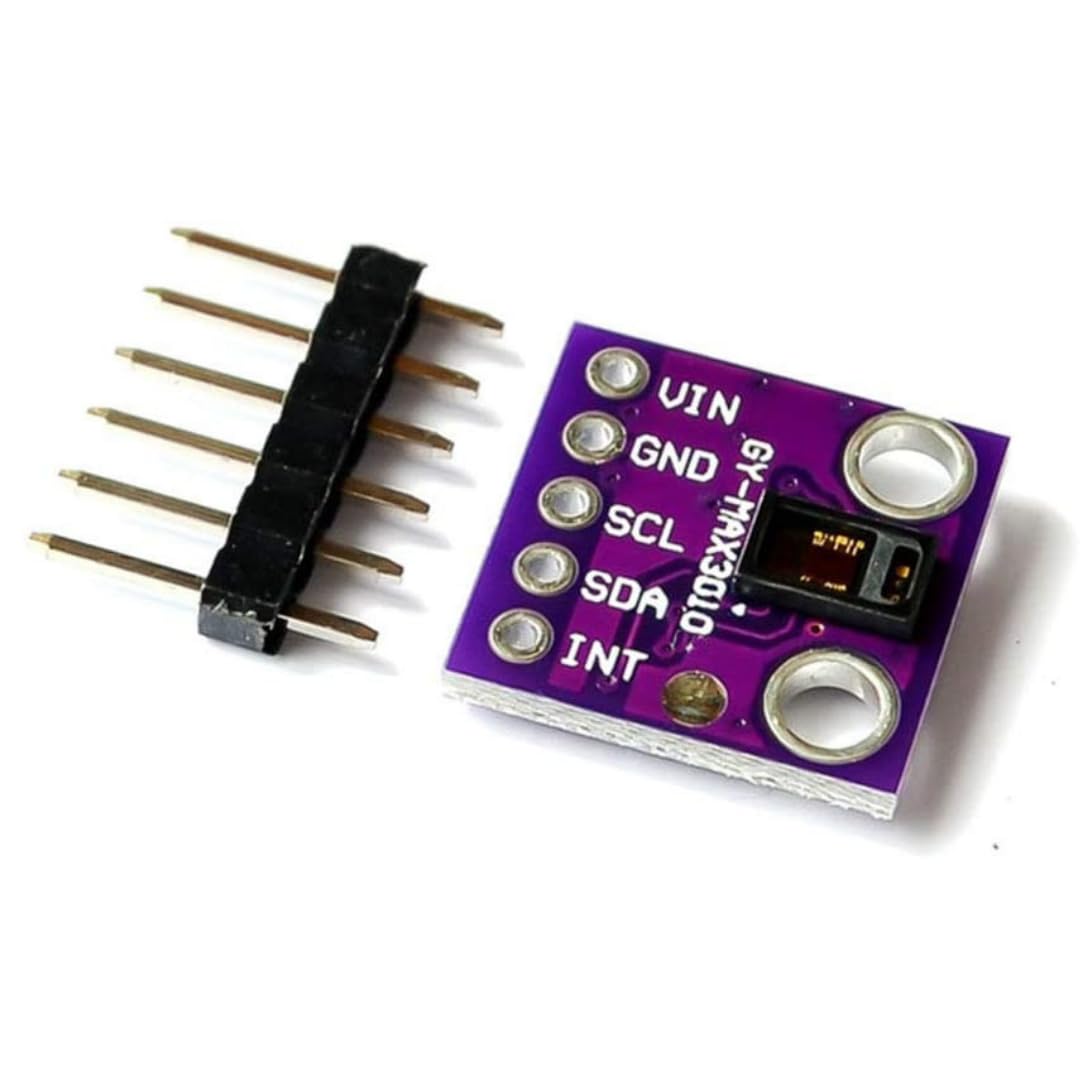


Figure 6: MAX30100

The MAX30100 is also an infrared light heart rate sensor and pulse oximeter and is able to measure heart rate and blood oxygenation levels (SpO2). This sensor is the main component for the heart rate sensor circuit, whereby the sensor collects data from the pulse of the subject by picking up the changes in light absorbed when the blood is passing through the finger. This data is transmitted by the MAX30100 to the Arduino Uno, where the data is computed, and the heart rate is printed onto the display screen.

For this project, all these components complement each other for the efficiency and effectiveness of the heart rate sensor system. The main processor is the Arduino Uno, the pulse data collector is the MAX30100 sensor, the heart rate display is the LCD display, and the breadboard, the jumper cables, and the battery provide the connectivity and the power for the system. These combined allow the heart rate to be detected in real-time, and the data is beneficial for fitness and health purposes (Sensor's, 2017).

## 1.3 System Design

[**TinkerCad Link:**](https://www.tinkercad.com/things/2c2NYYaN7Rs/editel?returnTo=%2Fdashboard%2Fdesigns%2Fcircuits&sharecode=x0e25_ck3U-komp6msxq507m4G2igtJlfkiZl3Ekt4I)

A circuit board with wires connected to it

AI-generated content may be incorrect.

Figure 7: Arduino Uno R3

The simulation presents an Arduino Uno device which connects to a breadboard through jumper wires within the Tinkercad platform. The shown set-up belongs to a heart rate monitoring system, yet it does not display essential elements such as the pulse sensor and LCD I2C display. The breadboard obtains power from the Arduino connection. An Arduino 5V output wire links to the breadboard red power rail and its GND wire connects to the breadboard black power rail for grounding.

The Arduino pins A4 and A5 receive their power through two green and orange jumper wires. The I2C pins on the system serve as communication ports to connect with an LCD I2C display that will display heart rate values. The circuit operates properly through the connection of a black wire to ground. The missing parts, including the pulse sensor and LCD I2C display, will be added later to complete the setup (Hrisko, 2019).

A circuit board with wires and a screen

AI-generated content may be incorrect.

Figure 8: LCD I2C

The system now incorporates an LCD I2C display component in this stage. The Arduino Uno operates this display through I2C protocol that functions with just two communication wires. The Arduino connects the SDA pin of the LCD through A4 and the SCL through A5. The Arduino acquires data transmission and reception capabilities from these two connections. Power supply connections for the LCD are established through the 5V rail on the breadboard which powers its VCC pin and the GND rail which powers its GND pin. This setup ensures that the LCD gets power and can properly communicate with the Arduino to show heart rate readings.

A circuit board with wires connected to it

AI-generated content may be incorrect.

Figure 9: MAX30100 sensor

The setup adds the MAX30100 pulse sensor as the last component. The simulation uses another component to replace the unavailable MAX30100 sensor from Tinkercad. This sensor acts as the vital component to determine heart rate together with oxygen levels.

The communication method of the MAX30100 sensor operates through the same I2C protocol as the LCD I2C display. The SDA pin of the sensor links to A4 on the Arduino platform alongside the SCL pin connected to A5. The sensor can transfer heart rate information to the Arduino system through its connection. Power is required for the sensor operation which connects its VCC pin to the 5V rail and GND pin to the GND rail on the breadboard.

The overall system has been installed successfully. The MAX30100 sensor serves as the heart rate detector and sends data to an Arduino processor which then displays heart rate results on the LCD I2C display.

## 1.4 Prototype Screenshot

A finger on a finger pointing at a circuit board

AI-generated content may be incorrect.

Figure 10: Prototype HMS

This is the final version of my heart rate monitoring system, which I built using an Arduino Uno, an LCD I2C display, and a MAX30100 pulse sensor. The 9V battery supply enables independent operation of the system without requiring computer connection.

The image shows me touching the MAX30100 sensor to measure my heart rate as well as oxygen levels (SpO2). The sensor transmits data to the Arduino board that proceeds the information for display on the LCD screen. My heart rate stands at 83 BPM while my SpO2 level reads 97% through the screen display which indicates normal parameters.

Real-time display of heart rate and oxygen level values appears clearly on the LCD I2C screen. The system has been configured on a breadboard using correct wiring connections to maintain proper functionality. The yellow LED on the LCD screen shows that power and correct operation of the system are present. The project successfully builds an IoT-based heart rate monitoring system that uses basic components.

A screen shot of a graph

AI-generated content may be incorrect.

Figure 11: Plot graph

The MAX30100 sensor displays heart rate using blue line data while showing oxygen levels by orange line data. When the sensor reaches operational temperature both reading values remain at low levels. Heart rate reaches 100 BPM while SpO2 shows normal values between 98-100% during this period. The sensor measurement values remain steady during this period. The sensor readings experience a quick decrease at the conclusion possibly due to the finger being taken off from the sensor. The sensor operates correctly to monitor live data based on this recorded output.

## 1.5 Source Code

A screenshot of a computer program

AI-generated content may be incorrect.

Figure 12: Code 1

The Arduino code serves as a platform for heart rate monitoring which combines a MAX30100 pulse oximeter sensor together with a 16x2 I2C LCD display. The program integrates all essential libraries needed to operate sensor devices and screen components. The system updates readings every one second because the REPORTING\_PERIOD\_MS value is set to 1000 milliseconds. Both lcd and pox objects serve different purposes in the program where lcd controls the display interface and pox interfaces with the pulse oximeter sensor. The program utilizes sp and hr variables to store reading data about SpO2 and heart rate measurements. A specific heart pattern consisting of an 8-row binary code pattern will be shown on the LCD screen. The system configuration enables it to show heart rate and oxygen levels in real time operations.

A screen shot of a computer program

AI-generated content may be incorrect.

Figure 13: Code 2

This code section performs heartbeat detection while it also starts up the system operations. The onBeatDetected() function activates upon detecting any heartbeat. It prints "Beat!" on the serial monitor. The setup() function executes just during the Arduino startup. The system starts serial communication with 9600 baud rate and displays "Initializing pulse oximeter.." to the serial monitor. After initialization the LCD backlight activates while a custom heart symbol gets stored in the display. During the initial 2-second display period the LCD shows "PATIENT HEALTH" on the first row followed by "MONITORING" on the second row. After the duration it clears the screen. The system configuration makes it possible for the device to show heartbeats immediately upon boot.

A screen shot of a computer program

AI-generated content may be incorrect.

Figure 14: Code 3

The code continues by establishing LCD connections while it initializes the pulse oximeter sensor. The LCD display shows "SYSTEM USING" on row one and "ARDUINO" on row two for two seconds then removes all screen content. Next the program attempts to begin pulse oximeter sensor operation through pox.begin(). The failed initialization triggers the code to display "FAILED" on the serial monitor followed by entering a never-ending loop that prevents further execution. The sensor begins operation and displays the SUCCESS message. The program sets the infrared LED current of the MAX30100 sensor to 30.6mA through the pox.setIRLedCurrent() command to enable heart beat detection. The onBeatDetected() function receives its trigger from the sensor when it detects heartbeats.

A screen shot of a computer program

AI-generated content may be incorrect.

Figure 15: Code 4

This part of the code operates continuously through loop() to show heart rate measurements and oxygen saturation readings. The heartbeat reading from the pulse oximeter sensor remains active through the use of pox.update() function. The program acquires heartbeat data and SpO2 readings from the sensor once per second according to REPORTING\_PERIOD\_MS. The program displays the values through the LCD screen and serial monitor. The heart rate screen on the LCD contains "HEART RATE:" with the heart rate number adjacent to a heart symbol. The screen displays "SpO2 LEVELS:" after which it shows the oxygen saturation percentage. The serial monitor and LCD display the identical information for debugging purposes. The statement tsLastReport = millis(); functions to update the last report timestamp for maintaining one-second value refresh intervals.

## 1.6 Limitation

The first limitation would be accuracy issues like for the heart rate sensors MAX30100 uses light to measure blood flow, but their accuracy can be affected by movement, poor sensor placement, or external light. If the sensor is not properly attached or if the person moves too much, the readings may be incorrect. Skin tone and blood circulation can also impact accuracy. To solve this limitation, it would be better if the person should not move as it needs to be stationary while checking their heart rate via using touching the MAX30100 sensor.

Another limitation would be the power consumption, heart rate monitoring systems need a reliable power source, especially if they run continuously. Sensors and microcontrollers, like the Arduino Uno, can drain battery life quickly, especially when using wireless communication. To solve this limitation, it is better to use energy efficient components where it would make the battery last longer.

Lastly, would be the environmental interferences, due to the performance of the heart rate the sensors can be affected by the surrounding environment. Changes in temperature and humidity may impact how well the sensor works overtime. Also, signals from other electronic devices can interfere with the sensor, leading to inaccurate readings. To solve this limitation, its better to use protective shielding and filtering techniques in order to maintain accurate and stable readings.

## 1.7 Future Enhancement

Heart rate monitoring system can be improved by adding wireless connectivity such as Wi-Fi or Bluetooth. This will allow the device to send heart rate data to a mobile phone, computer, or cloud storage in real time. With cloud integration, doctors or family members can check the heart rate remotely, which is useful for patients who need continuous monitoring.

Another future enhancement would be a mobile app can be developed to display heart rate readings in an easy-to-understand way. The app can show real-time data, store past readings, and even send alerts if the heart rate is too high or too low. This will make it more user-friendly, as people can access their health information anytime on their smartphones. The app can also connect with fitness trackers and smartwatches, making it more useful for people who exercise regularly.

Since the system runs on battery power, making it more energy-efficient will be important. Using better microcontrollers, such as the ESP32, can help reduce power consumption. Also, adding smart power-saving features, like sleep mode when the sensor is not in use, can help extend battery life. This will make the device last longer between charges, making it more practical for daily use (Real Time Heart Rate Monitoring System Using Arduino, 2023).

# Conclusion

The project proves that IoT technology together with an Arduino Uno platform can successfully monitor heart rate in real time. The integrated system of the MAX30100 pulse oximeter sensor with an LCD display presents an accessible low-cost technology for monitoring both heart rate and SpO2 levels. The system benefits from Arduino hardware because it provides flexibility alongside easy prototyping methods and healthcare-oriented suitability.

Machine functionality exists but the system encounters accuracy challenges because of movements together with power usage problems and environmental resistance. The system weak points can be resolved through best placement of sensors alongside the use of efficient components as well as protective shielding methods.

Integration of wireless connectivity through Wi-Fi or Bluetooth along with cloud-based data storage systems would improve the device for future development. Real-time health monitoring combined with history storage and alert notifications would be achievable through developing an app that would improve user experience. The system's usefulness for consistent application will increase if we optimize its power efficiency.

Healthcare managers can leverage this project because it demonstrates how IoT technology develops affordable portable systems for heart rate monitoring which help patients and medical professionals simultaneously.

# 

# References

Abdulmalek, S., Nasir , A., Jabbar, W. A., Almuhaya, M. A., Bairagi, A. K., Khan, M. A.-M., & Kee, S.-H. (2022). IoT-Based Healthcare-Monitoring System towards Improving Quality of Life: A Review. *NLM*, 1-7.

Banzi, M. (2011). Getting Started with Arduino. *Sarcnet*, 1-3.

Hrisko, J. (2019, June 29). *Arduino Heart Rate Monitor Using MAX30102 and Pulse Oximetry*. Retrieved from MakerPortal: https://makersportal.com/blog/2019/6/24/arduino-heart-rate-monitor-using-max30102-and-pulse-oximetry?srsltid=AfmBOor2e3LX0c2BZUxWObiywAPn2snWp58g76I5nbM2rXlR9Tp6AwlS

*Real Time Heart Rate Monitoring System Using Arduino*. (2023, March 24). Retrieved from Quartz Components: https://quartzcomponents.com/blogs/electronics-projects/real-time-heart-rate-monitoring-system-using-arduino?srsltid=AfmBOoqEljolVypfskT\_dT4TXQ29WJwsqDGdKIQXFwx53F3WUky69YId

Sensor's, A. &. (2017, March 5). *Analog & Digital Sensor's*. Retrieved from LETS ACADEMY: https://leetsacademy.blogspot.com/2017/03/max30100-spo2-sensor-with-arduino.html