Introduction

Smart areas are being facilitated by recent developments in telecommunications and information systems, particularly those connected to the Internet of Things (IoT). The Internet of Things (IoT) based patient health monitoring system is a cutting-edge technology that allows for real-time monitoring of a patient's health status. This project use webserver which is capable of monitoring heart rate, oxygen level, and body temperature of a person. This data is then analysed and used to assess a person's level of health, which using web server provides interface for healthcare providers to access and monitor the medical data of their patients. This project also uses MLX90614 as a temperature sensor to measure a person's body temperature. ESP 32 is as a microcontroller used to open the access point to receive WIFI signal with other devices to send data output. As the result, we can monitor it LCD display, OLED Display and also in Web Interface.

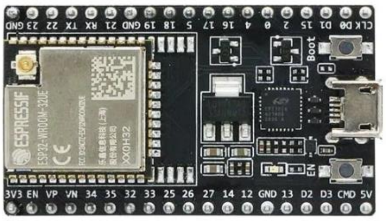
Objective

- Construct the circuit and develop code program using Arduino IDE.
- Collecting information from pulse sensor and temperature sensor for processing on a computer.
- Able to monitor the person's health through the web server, LCD and OLED Display.

Software

List	Purpose	Image
Arduino IDE	Text editor for writing code	ARDUINO
Fritzing	Design Schematic Circuit	fritzing

Component

List	Purpose	Image
ESP 32	Main controller and Wi-Fi Module	
MAX30100 Sensor	Measurement Oxygen level and Heart Rate	
MLX90614 Sensor	Measurement Body Temperature	
I2C LCD Display (16x2)	Display Result	
I2C OLED Display (128 x 64)	Display Result	

ESP 32 Microcontroller and Wi-Fi Module

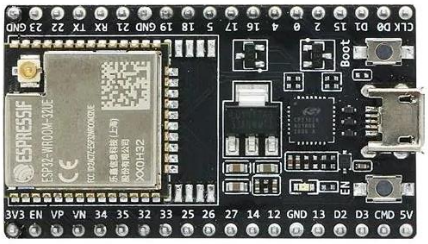

GND 23 22 TX RX 21 GND 19 18 5 17 16 4 0 2 15 D1 D0 CLK
Boot
EN
3V3 EN VP VN 34 35 32 33 25 26 27 14 12 GND 13 D2 D3 CMD 5V

Figure 1 show ESP32

The ESP32 is a low-cost, low-power microcontroller with integrated Wi-Fi and dual-mode Bluetooth capabilities. Some of its key features include:

1. **Dual-Core Processing:** The ESP32 features a dual-core processor, with each core running at up to 240 MHz, providing a powerful and flexible platform for a wide range of applications.
2. **Wireless Connectivity:** The ESP32 is equipped with built-in Wi-Fi and Bluetooth capabilities, enabling it to connect to the internet and other devices wirelessly.
3. **High-Speed Data Transfer:** The ESP32 supports data transfer rates of up to 150 Mbps for Wi-Fi and 1 Mbps for Bluetooth, allowing for fast and reliable communication.
4. **Low Power Consumption**: The ESP32 is designed to consume low power, making it ideal for battery-powered and energy-efficient applications.
5. **Large Memory:** The ESP32 features 520 KB SRAM and up to 4 MB of flash memory, providing ample storage space for programs and data.
6. **Peripheral Interfaces**: The ESP32 provides a variety of peripheral interfaces, including I2C, UART, SPI, I2S, PWM, and ADC, allowing it to interface with a wide range of external sensors and devices.
7. **Comprehensive Development Environment:** The ESP32 is supported by a comprehensive development environment, including an SDK, libraries, and a variety of development tools, making it easy to develop and deploy applications on the platform.

MAX30100 Heart Rate and Oxygen Sensor

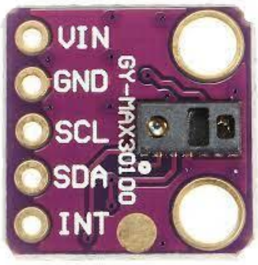


Figure 2 show MAX30100 Heart Rate and Oxygen Sensor

The heart rate oximetry subsystem of the MAX30100 includes ambient light cancellation, a 16-bit sigma-delta ADC, and a unique discrete-time filter. It operates at extremely low power, making it perfect for battery-powered devices. The sensor requires a voltage source between 1.8 and 3.3 volts. Wearable gadgets, fitness assistance devices, medical monitoring devices, and other devices use it. Software can also be used to turn it off with very minimal standby current, enabling the power supply to remain on at all periods. The sensor uses the I2C protocol to send it to the microcontroller.

MLX90614 Sensor

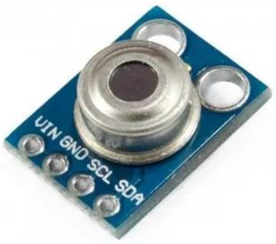


Figure 3 show MLX90614 Temperature sensor

The MLX90614 is a non - contact temperature sensor that uses Infrared (IR) technology. It can detect temperatures ranging from -70 to 382 degrees Celsius. The sensor monitors the temperature of an item and uses the I2C protocol to send it to the microcontroller.

- Voltage: 3.6 to 5V
- Current: 1.5mA
- Range: -69° C to 381° C
- Precision: 0.021° C
- Field of View: 80°

- Ambient Range: -39° C to 124° C
- Distance between sensor and object: 2cm-5cm (estimated).

Methodology

Flowchart of IOT based Patient Health Monitoring System

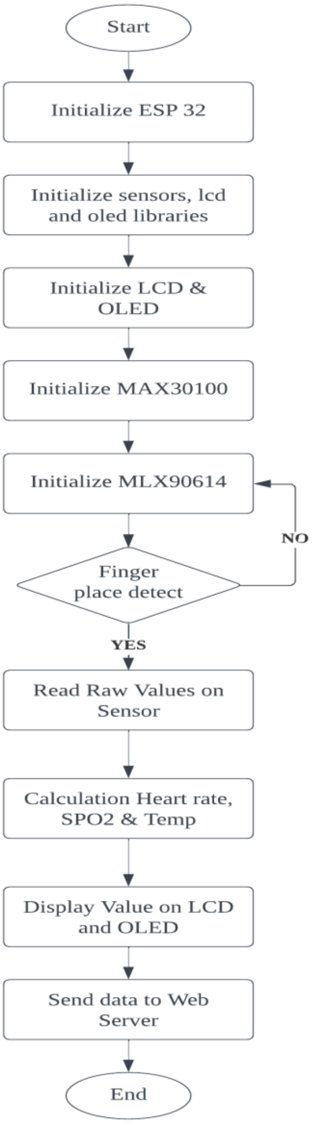

Start
Initialize ESP 32
Initialize sensors, lcd and oled libraries
Initialize LCD & OLED
Initialize MAX30100
Initialize MLX90614
NO
Finger place detect
YES
Read Raw Values on Sensor
Calculation Heart rate, SPO2 & Temp
Display Value on LCD and OLED
Send data to Web Server
End

Figure 4 show Flowchart process

Schematic Circuit

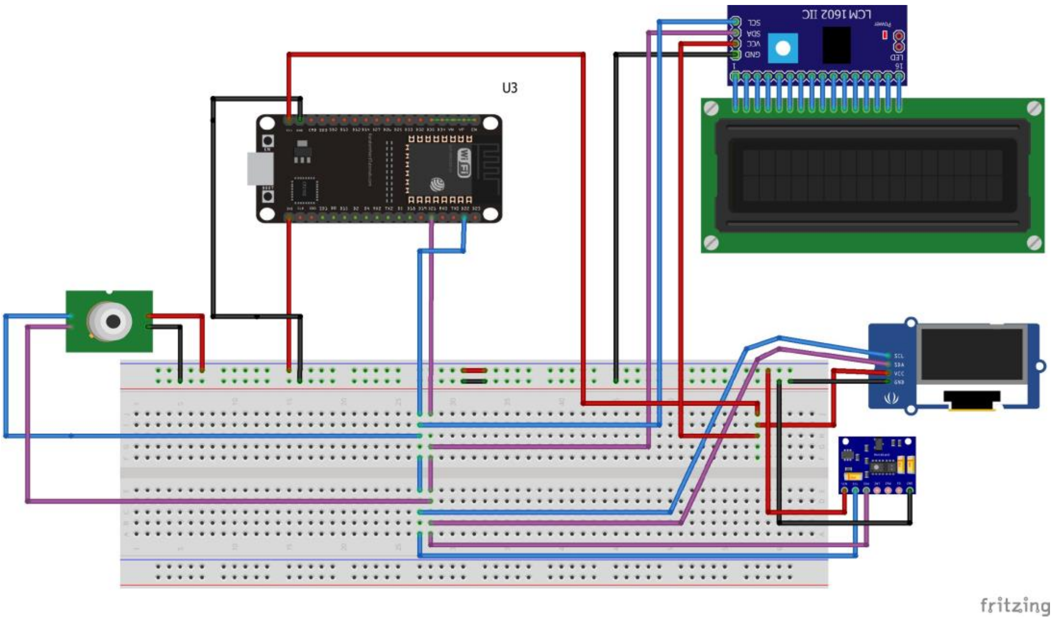


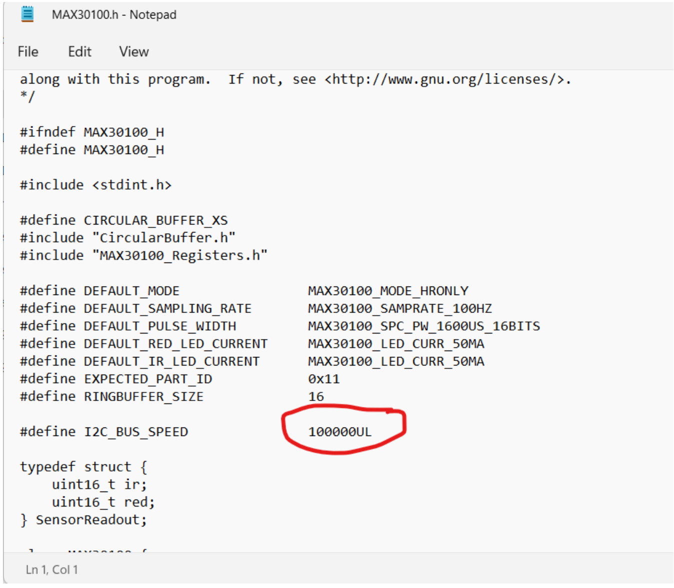
Figure 5 show connection the electronic circuit with sensors

Design System

Figure 5 show the connection between sensor, lcd and oled display. It is made up of several blocks. All processes will be controlled by the ESP 32. We can program ESP 32 and link circuits to it. The MAX30100 integrated IC, which is a tiny, low-cost integrated solution for sensing heart rate and pulse oximeter, is used in the Pulse Oximeter and Heart Rate Sensor module as well as ambient temperature using the MLX90614 temperature sensor. The ESP32 output will be displayed on the web through Wi-Fi. All Pulse Oximeter has an IR led and a RED led. If one places one's finger on this sensor, the red-light travels into one's skin, and the IR led receives it. The O2 level in the blood is measured using both red and infrared LEDs. The MLX90614 is a digital temperature sensor that could be used to monitor temperature. The sensor measures body temperature using infrared radiation without requiring any physical touch, both sensors (MAX30100 and MLX90614) communicate with the microcontroller via the Inter-Integrated Circuit (I2C) protocol, which detects a beat after a finger is placed on the MLX30100 for ten times and sends the sensor into shutdown mode, with the most recent reading saved and sent serially to the Wi-Fi module via Inter-Integrated Circuit (I2C). At the diagram also show all the component such as MAX30100, MLX90614, OLED & LCD Display are using 12C protocol which uses the SCL 22 and SDA 21 ports on the ESP 32 board. The both sensor is connected to 3.3 V powered, LCD and OLED Display connected to 5V.

Problem when integrated MAX30100 and MLX90614 sensor

The problem encountered when wanting to integrate the MAX30100 and MLX90614 sensors is that the sensors cannot be read simultaneously. So the solution we do is to change in MAX30100 library sets the I2C bus speed to 400kHz (I2C_BUS_SPEED in MAX30100.h). Change the I2C bus speed to 100Khz and the result is MAX30100 and MLX90614 sensors can be read together. It’s because common I2C bus speeds are the 100 kbit/s standard mode and the 400 kbit/s fast mode. Figure 6 show the changes in MAX30100.h library.



```
along with this program.  If not, see <http://www.gnu.org/licenses/>.
*/

#ifndef MAX30100_H
#define MAX30100_H

#include <stdint.h>

#define CIRCULAR_BUFFER_XS
#include "CircularBuffer.h"
#include "MAX30100_Registers.h"

#define DEFAULT_MODE                MAX30100_MODE_HRONLY
#define DEFAULT_SAMPLING_RATE       MAX30100_SAMPRATE_100HZ
#define DEFAULT_PULSE_WIDTH         MAX30100_SPC_PW_1600US_16BITS
#define DEFAULT_RED_LED_CURRENT     MAX30100_LED_CURR_50MA
#define DEFAULT_IR_LED_CURRENT      MAX30100_LED_CURR_50MA
#define EXPECTED_PART_ID            0x11
#define RINGBUFFER_SIZE             16

#define I2C_BUS_SPEED               100000UL

typedef struct {
    uint16_t ir;
    uint16_t red;
} SensorReadout;
```

Figure 6 show the changes in MAX30100.h library

Design and implementation of a web server and web page

One of the most crucial components of the system is the web server. It is in charge of processing requests from the Microcontroller and the sensor device. Before doing web server operations we need to know the basics of html and CSS code to design a website. Then, once the html and CSS code is finished developing, we need to include the HTTP server code for it to start responding. We have integrated Asynchronous JavaScript and XML (AJAX) Code. So that we can request data from the server asynchronously (in the background) without refreshing the page. So, after successful upload the code program into ESP 32, the IP address will show in serial monitor and use the IP address to search in existing web browser using smartphone or laptop. So, after enter the IP address we can monitor BPM, Sp02 and Body Temperature values from any smartphone or PC connected to the same Wi-Fi. These parameters are updated every second by doing a silent GET request on the server and updating the parameters on the page with AJAX.

Conclusion

In conclusion, the objective for this project was successfully achieved because the sensor can be read accurately and is able to display values through LCD, OLED Display and can also be monitored through wireless communication to the website. This project is quite useful for patients or non-patients who are at home, as it allows for quick treatment and monitoring of the patient's condition. In a separate approach, it will help medical professionals by preventing them from being exposed to disease as well as eliminating face-to-face follow-ups that are currently conducted in isolation. In hospitals and clinics, medical workers will always be needed, This system will help in reducing the number of patients visiting the hospital, allowing staff to focus on the most critical cases and, to some extent, minimizing stress and anxiety associated with exposure to and interaction with so many patients.