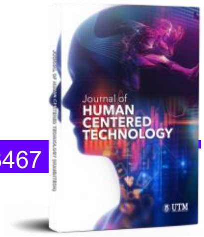
Journal of Human Centered Technology
https://www.humentech.utm.my
E-ISSN: 2821-3467



Development of Mobile and Real-Time Vital Signs Monitoring System with Camera Integration

Muhammad Ikmal Danish Azemi[1], Mohd Riduan Mohamad[1,2*], Muhammad Harist Danish Md Rajuna[1]

[1]Department of Biomedical Engineering and Health Sciences, Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310, UTM Johor Bahru, Johor, Malaysia.
[2]Bioinspired Device & Tissue Engineering Research Group, Department of Biomedical Engineering and Health Sciences, Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310, UTM Johor Bahru, Johor, Malaysia.

*Corresponding Author mohd.riduan@utm.my

Cite: https://doi.org/10.11113/humentech.v4n2.111

Research Article

Abstract:
Pre-hospital and emergency department staff play important roles to treat patients where these two departments are correlated to each other. However, there are few misalignments between them such as dissimilar training and clinical abilities, language barriers, and high-risk clinical environments which are characterized by noise, interruption, and time restriction. The objective of this study was to develop an Internet of Things (IoT) based health monitoring system with real-time camera inside an ambulance. With this system, the emergency department can monitor the current patient's condition in an ambulance while measuring data when paramedics are having difficulties. In conducting the study, the IoT health monitoring system was developed by using ESP32 as the main microprocessor. Few sensors were used including DS18B20, MAX30102, and ESP32-CAM. In addition, the Blynk application was used as the IoT and connected through Wi-Fi, provided by the ESP32. Five respondents were assigned for the data measurement of vital signs specifically heart rate, body temperature, and SpO_2 using both standard and proposed monitoring system to calculate average and percentage error. From the results, the percentage error for heart rate was recorded as 21.56 ± 8.40%, body temperature was measured as 3.14 ± 3.41%, and SpO_2 was recorded as 2.13 ± 0.87%. Moreover, during the device construction, the camera functional was delayed for approximately 5 second. In conclusion, integrating real-time video with vital sign data in IoT-based health monitoring systems provides a comprehensive view of a patient's condition, enabling proactive treatment adjustments and better outcomes.

Keywords: Mobile; Vital signs; IoT; Monitoring system; Camera integration

SUPPLEMENTARY DATA

```
#define BLYNK_TEMPLATE_ID "TMPL6il1DGAwR"
#define BLYNK_TEMPLATE_NAME "Monitoring System"
#define BLYNK_AUTH_TOKEN "MiQIt55glwU9g3hkouPUT5nXpBm9WpAR"

#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

#include <OneWire.h>
#include <DallasTemperature.h>
#include <Wire.h>
#include "MAX30105.h"
#include "spo2_algorithm.h"


// WiFi credentials
char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "POCOF5";
```


Received 11 June 2025; Accepted 01 August 2025
Available online 06 August 2025

```
char pass[] = "pool11111";

// OneWire and DallasTemperature setup
#define ONE_WIRE_BUS 13
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

// Blynk timer
BlynkTimer timer;

// MAX30105 setup
MAX30105 particleSensor;
#define REPORTING_PERIOD_MS 1000

uint32_t tsLastReport = 0;
uint32_t irBuffer[100]; // Infrared LED sensor data
uint32_t redBuffer[100]; // Red LED sensor data
int32_t bufferLength = 100; // Buffer length of 100 stores 4 seconds of samples at 25 samples per second
int32_t spo2; // SPO2 value
int8_t validSPO2; // Indicator to show if the SPO2 calculation is valid
int32_t heartRate; // Heart rate value
int8_t validHeartRate; // Indicator to show if the heart rate calculation is valid

void sendTemperature()
{
 sensors.requestTemperatures();

 Serial.print("Celsius temperature: ");
 Serial.println(sensors.getTempCByIndex(0));
 int tempC = sensors.getTempCByIndex(0);

 Blynk.virtualWrite(V0, tempC);
}

void applyMovingAverageFilter(uint32_t *data, int32_t length, int32_t windowSize)
{
 for (int i = 0; i < length - windowSize + 1; i++)
 {
  int32_t sum = 0;
  for (int j = 0; j < windowSize; j++)
  {
   sum += data[i + j];
  }
  data[i] = sum / windowSize;
 }
}

void sendOximeterData()
{
 for (int i = 0; i < bufferLength; i++)
 {
  while (particleSensor.available() == false)
   particleSensor.check();

  redBuffer[i] = particleSensor.getRed();
  irBuffer[i] = particleSensor.getIR();
  particleSensor.nextSample();
 }

 // Apply moving average filter to the data
```

```
    applyMovingAverageFilter(redBuffer, bufferLength, 5);
    applyMovingAverageFilter(irBuffer, bufferLength, 5);

    maxim_heart_rate_and_oxygen_saturation(irBuffer, bufferLength, redBuffer, &spo2, &validSPO2, &heartRate,
&validHeartRate);

    if (validHeartRate && validSPO2)
    {
     Serial.print("Heart rate: ");
     Serial.print(heartRate);
     Serial.print(" bpm / SpO2: ");
     Serial.print(spo2);
     Serial.println(" %");

     Blynk.virtualWrite(V3, heartRate);
     Blynk.virtualWrite(V4, spo2);
    }
    else
    {
     Serial.println("Invalid readings");
    }

    if (millis() - tsLastReport > REPORTING_PERIOD_MS)
    {
     tsLastReport = millis();
    }
   }

   void setup()
   {
    Serial.begin(115200);

    sensors.begin();

    Blynk.begin(auth, ssid, pass);

    if (!particleSensor.begin(Wire, I2C_SPEED_FAST))
    {
     Serial.println("Failed to initialize MAX30105 sensor");
     for (;;);
    }
    else
    {
     Serial.println("MAX30105 sensor initialized");
     particleSensor.setup();
     particleSensor.setPulseAmplitudeRed(0x1F); // Set higher LED power for better readings
     particleSensor.setPulseAmplitudeIR(0x1F); // Ensure IR LED is also appropriately powered
     particleSensor.setPulseAmplitudeGreen(0);
     particleSensor.setSampleRate(50); // Increase sample rate for better resolution
    }

    timer.setInterval(1000L, sendTemperature);
    timer.setInterval(4000L, sendOximeterData);
   }

   void loop()
   {
    Blynk.run();
    timer.run();
}
```