

ARTICLE

# Self-health Monitoring and Reporting System for COVID-19 Patients Using CAN Data Logger

Uttam U. Deshpande\* Pooja Hulajatti Shraddha Suryavanshi Vrunda Balgi

Department of Electronics and Communication Engg., KLS Gogte Institute of Technology, Belagavi, Karnataka, 590008, India

ARTICLE INFO

*Article history*

Received: 28 July 2021

Accepted: 10 August 2021

Published Online: 12 August 2021

*Keywords:*

COVID-19

Home isolation

CAN protocol

Data logger

MAX30102

Health-monitoring

ABSTRACT

In the evolving situation of highly infectious coronavirus, the number of confirmed cases in India has largely increased, which has resulted in a shortage of health care resources. Thus, the Ministry of Health and Family Welfare- Government of India issued guidelines for the 'Home isolation of COVID-19 positive patients' methodology for asymptomatic patients or with mild symptoms. During home isolation, the patients are required to monitor and record the pulse rate, body temperature, and oxygen saturation three times a day. This paper proposes a system that can request data from the required sensor to measure the pulse rate, body temperature, or oxygen saturation. The requested data is sensed by the respective sensor placed near the patients' body and sent to the CAN data logger over the CAN bus. The CAN data logger live streams the sensor values and stores the same to an excel sheet along with details like the patient's name, patient's age, and date. The physicians can then access this information.

## 1. Introduction

Bosch developed a serial communication protocol called CAN in 1985 to reduce the complex wiring in vehicles. It provides highly efficient and reliable communication between sensors, actuators, controllers, etc. It is a standard network that is preferred in most embedded applications. Previously, CAN was mainly used by the vehicle industry. From then, it is found in almost every automobile ranging from on-road to off-road. Whereas now, the protocol is also widely used in nearly every area of networked embedded controls, with its applications in various products such as medical equipment, building automation, production machinery, weaving machines, wheelchairs, etc. Most medical devices use CAN as an

underlying embedded network. This paper presents a Self-health monitoring and reporting system using a CAN logger to monitor the body temperature, oxygen level, and heart rate of COVID-19 patients. CAN data logger is an electronic device used to record the data on the CAN bus over time. A microprocessor with sensors and internal memory forms the base of loggers. These loggers can be stand-alone devices or interfaced with a personal computer through software or user interface.

Presently, the entire world is battling the pandemic of COVID-19. With the increase in the number of affected people, the hospital infrastructure worldwide is insufficient to treat and take care of all the patients. It has led to a situation where asymptomatic patients and patients with

*\*Corresponding Author:*

Uttam U. Deshpande,

Department of Electronics and Communication Engg., KLS Gogte Institute of Technology, Belagavi, Karnataka, 590008, India;

Email: [uttamudeshpande@gmail.com](mailto:uttamudeshpande@gmail.com)

mild symptoms are isolated at home. However, during this period, the patients must keep track of certain health parameters such as body temperature, oxygen level, and heart rate regularly and report the same. The patient is provided with a thermometer and pulse oximeter separately to record the parameters. This data then needs to be manually updated in the report sheet. Sometimes, this would lead to manipulating the records from the patients' end due to various reasons affecting the patients' health directly.

This paper introduces a system that can read and store physiological parameters and other details about COVID-19 patients. A graphical user interface is provided with the system to create an account for each patient, measure the health parameters, and store the same in an excel sheet. The implementation of the device is categorized into two functional units. The Sensing Unit comprises Arduino UNO, a temperature sensor, and a pulse oximeter. The CAN data logger comprises of STM32F407 microcontroller. The Sensing unit reads sensor data and sends over the CAN bus when requested by the CAN data logger, responsible for displaying it and saving it on the excel sheet through the user interface provided.

## 2. Literature Review

As illustrated by Priyanka Kakria et al. <sup>[1]</sup>, Veena Sundareswaran et al. <sup>[2]</sup>, and C. Premalatha et al. <sup>[3]</sup>, the Patient monitoring Systems available in the market consists of a microcontroller that could be ATMEGA328 and various health parameter measuring sensors like DS18B20 for temperature, SO00837PS pulse sensor, etc. These systems are plug-and-play devices that collect the data through the sensors and send them to the monitoring system. The monitoring system could be a personal computer, mobile or display interface provided with the device itself. The major setback of using these devices during this pandemic is that the existing devices are not robust. They do not implement any standard secure communication protocol for exchanging the data. These devices are only applicable for personal usage and do not provide any scope to integrate them with the hospital tools and equipment. The devices are mostly capable of only measuring the data and sending it to the monitoring system. Though some systems, as proposed by Prasad Lavgi et al. <sup>[4]</sup>, using CAN protocol, there is no customized application interface that facilitates the patients' data storage.

The systems proposed by K. P. Swain et al. <sup>[5]</sup>, Karumuri Anusha Reddy et al. <sup>[6]</sup>, and Ladlennon C. Banuag et al. <sup>[7]</sup> uses the CAN protocol to monitor the parameters of multiple patients continuously. In these cases, each patient is required to have the sensing unit always attached

to their body. The use of multiple sensing units makes the overall system very expensive.

During this pandemic of COVID-19, it's extremely necessary for the devices being used to be robust, secure, and reliable. Hence, the system illustrated in this paper uses a highly secure CAN protocol for the exchange of data between the sensing node and the data logging unit. This system is interfaced with a customized GUI that provides the medium to log the parameters of multiple patients and is also capable of generating a report that is necessary for analyzing the health of the patient with the timely logged data. The device proposed is designed considering the present coronavirus pandemic situation and is customized in the most suitable form that is useful for the patients in home-isolation and hospitalized.

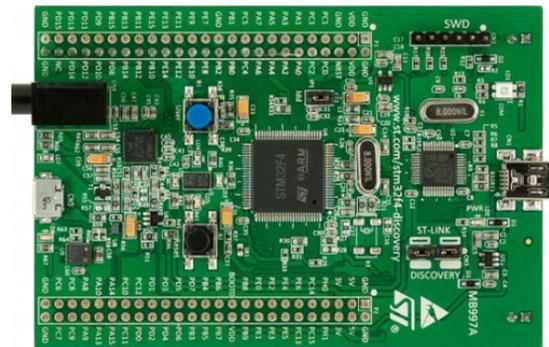


Figure 1. STM32F407VGTx microcontroller

## 3. Experimental Setup

### 3.1 Hardware

#### STM32F407VGTx Microcontroller

Figure 1 illustrates the features of 32F407VGTx Microcontroller. It is a high-performance 32-bit ARM Cortex M4 microcontroller with 100 pins (LQFP100 package). It comes with an FPU core, 1-Mb Flash memory, 192-Kb RAM, and ST-LINK/V2. It also includes LEDs, push buttons and a USB OTG micro-AB connector, etc. Various peripherals like ADC, DAC, USART, SPI, I2C, CAN, TIMER, etc., are available <sup>[8,9]</sup>.

#### MCP2551 CAN transceiver

A high-speed CAN transceiver supports CAN operations of up to 1Mbps with a maximum of 112 nodes connected over the bus and is suitable for 12 V or 24 V operations. It converts the digital signals from the CAN peripheral of a microcontroller to differential signals suitable for the CAN bus and vice versa. It plays a major role in noise-cancellation over the bus <sup>[10]</sup>. Figure 2 highlights MCP2551 CAN transceiver.



Figure 2. MCP2551 CAN transceiver

### FTDI module

It is an USB-to-UART serial converter module and is shown in Figure 3. It has an USB interface, Tx/Rx pins and operates at 3.3 V or 5 V DC. It can be used to convert the data present on USB to serial data suitable for the UART peripheral and vice versa.



Figure 3. FTDI module

### Arduino UNO (ATmega328P)

It is a high-performance 8-bit AVR microcontroller with 32 Kb of Flash memory, 1Kb EEPROM, 2 Kb internal SRAM. It operates at a voltage of 2.7 V-5.5 V and a temperature range of  $-40^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$ . It has many integrated peripherals like TIMERS, PWM, ADC, UART, SPI, etc. A variety of sensors and actuators can be integrated to sense and control the external environment. The Figure 4 indicates the Arduino UNO (ATmega328P) microcontroller.



Figure 4. Arduino UNO

### MCP2515 CAN shield

It is a CAN controller module with an SPI interface.



Figure 5. MCP2515 CAN shield.

### LM35 Temperature sensor

It is a precision temperature device with 3-terminals as shown in Figure 6. It produces an analog voltage that is linearly proportional to the temperature in centigrade. It can accurately measure temperature in the range of  $-55^{\circ}\text{C}$  to  $+150^{\circ}\text{C}$ .



Figure 6. LM35 Temperature sensor

### MAX30102 Pulse oximetry and Heart-rate monitor module

It is used to monitor heart rate and oxygen saturation level in the blood. It is integrated with two LEDs, a photodetector, low noise analog signal processing and optimized optics. It operates within the voltage range of 1.7 V-2 V. It outputs the heart-rate in BPM and the oxygen saturation level in percentage <sup>[12]</sup>. Figure 7 highlights MAX30102 module.



Figure 7. MAX30102 module.

## 3.2 Software

### Keil $\mu$ Vision 5

It is free software used for embedded software development. It is an Integrated Development Environment (IDE) with a text editor, compiler, and hex code generator. It allows verification of the code using the simulator, load the code into the microcontroller and validate the working using the debugger. The peripheral registers can also be

viewed on the debugger.

### Arduino 1.8.9

It is a cross-platform used by developers to write and upload codes in Arduino boards. It provides an interface to include libraries and program different sensors, compile and upload code to Arduino-compatible boards, views the results using Serial monitor and Serial plotter options.

### Microsoft Visual Studio 2019

It is an IDE used to develop GUIs, websites, apps, web services, and computer programs using software development platforms like Windows forms, Windows API, Microsoft Silverlight, etc. It includes a code editor, code refactoring, code profiler, debugger, GUI designers, web, class, etc.

### 3.3 Communication Protocol

#### CAN Protocol

It is a message-based, broadcast-type protocol with differential two-wired communication (CAN\_H and CAN\_L). It is a multi-master, asynchronous, serial communication facility defined by International Standards Organization (ISO). CAN bus supports data speeds up to 1 Mbps. The CAN\_H and CAN\_L lines are terminated by 120Ω resistance on either side to reduce the signal reflection. Due to its high performance and the ease of adding or deleting the nodes, the CAN bus is considered a flexible system. Each node can send and receive messages on the bus but not simultaneously. These messages follow the Standard CAN frame format defined under ISO-11898 [13].

#### UART

Universal Asynchronous Receiver Transmitter is a circuit used for serial communication. It transmits the data asynchronously by using only two wires. The transmitting UART converts the parallel data into serial, and receiving UART converts serial data back into parallel. The transmitting UART attaches additional Start and Stop bits to the 8-bit data package, indicating the start and end of data for receiving UART.

### 3.4 Block Diagram

The proposed system shown in Figure 8 mainly consists of two units, the Sensing unit where all the sensors i.e., the LM35 Temperature sensor read the body temperature. The MAX30102 Pulse oximetry and Heart-rate monitor module read heart rate and oxygen saturation level. They are connected with the CAN data logger and

GUI used to log and store the physiological parameters of multiple patients.

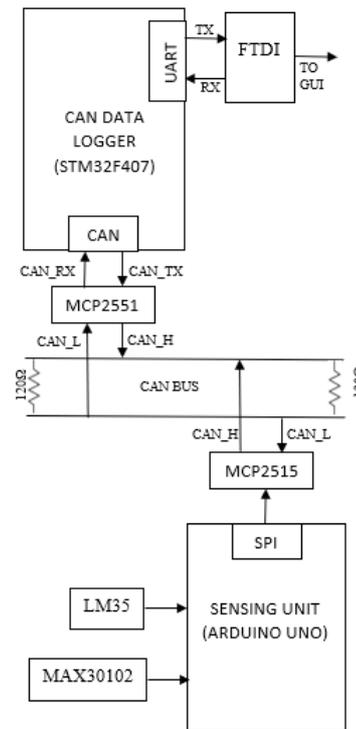


Figure 8. Block diagram

The CAN data logger communicates to the Sensing unit over the CAN bus, and the GUI communicates with the CAN data logger through UART.

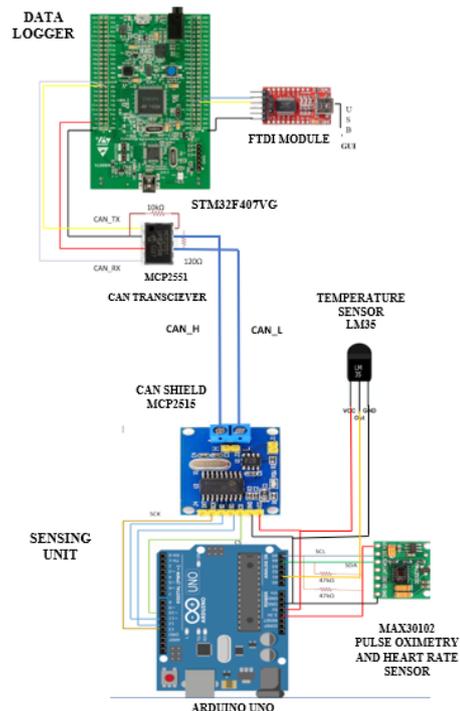


Figure 9. The layout of the proposed system.

## 4. Implementation

### Sensing unit

The microcontroller used is Arduino UNO (ATmega328P). The LM35 temperature sensor, the MAX30102 module is interfaced with it to read the body temperature and heart-rate + oxygen saturation level, respectively. The MCP2515 CAN shield interfaced to communicate over the CAN bus. The sensing unit responds to the CAN data logger's request by reading the data from sensors and sending it over the CAN bus. Figure 9 illustrates the layout of the proposed system.

### CAN data logger

The microcontroller used is STM32F407VGTx. It is the main unit that logs and stores the sensor data. The FTDI module and the MCP2551 CAN transceiver are interfaced with GUI through UART and over CAN bus.

When the user selects a particular sensor on GUI, it sends a CAN remote frame requesting sensor data to the Sensing unit. It then reads the response CAN message and displays the sensor data on GUI.

### GUI

The GUI provides means to interact with the CAN data logger. The patient can create an account, log in to log sensor data, and view previous records. It also provides an inference of the logged data. Multiple patient records can be stored, which can be accessed only by the doctor. The GUI stores details of the patient like name, patient ID, age, gender, address, etc. Figure 10 and 11 displays the flow diagram sensing unit and data logging units.

## 5. Results

The hardware setup for the proposed system is shown in Figure 12.

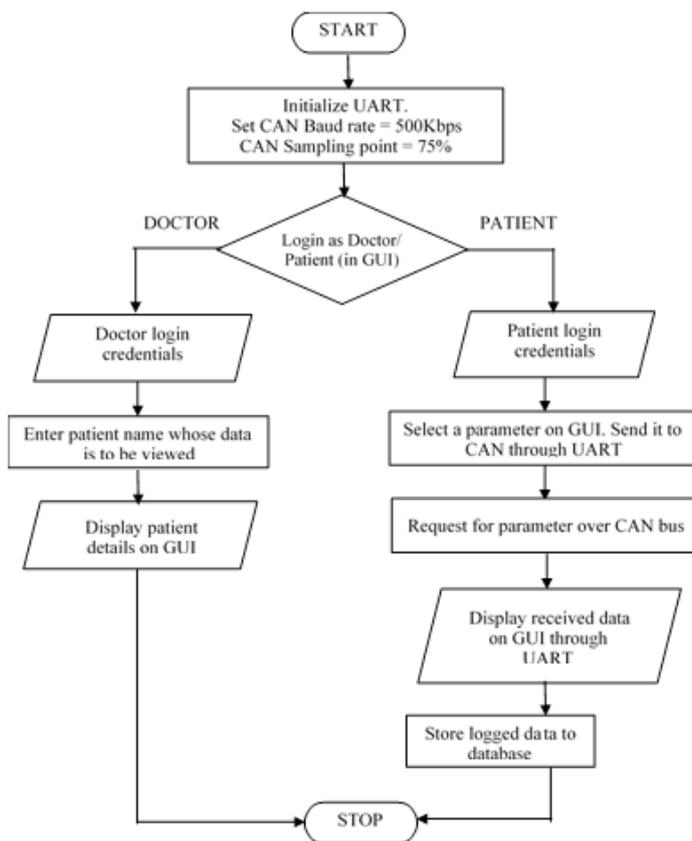


Figure 10. Process flow of Sensing unit.

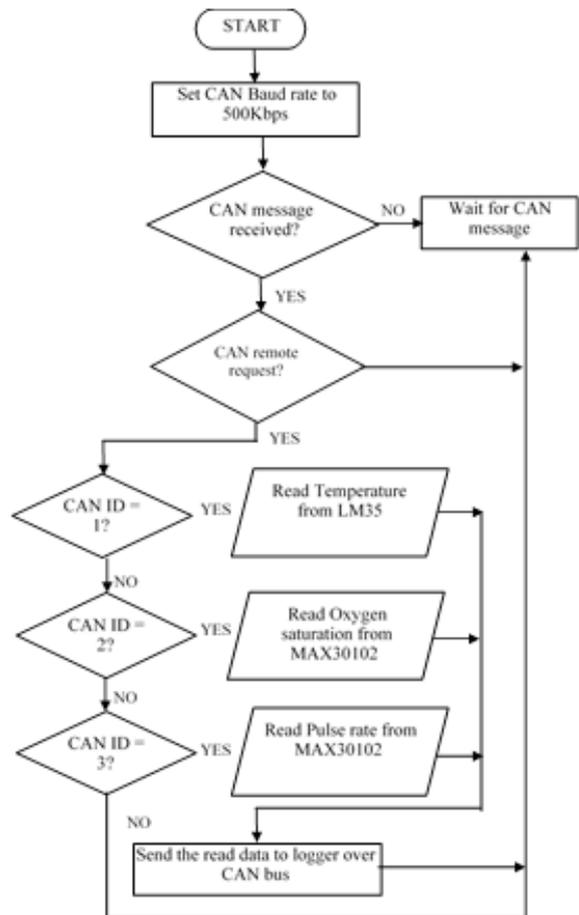
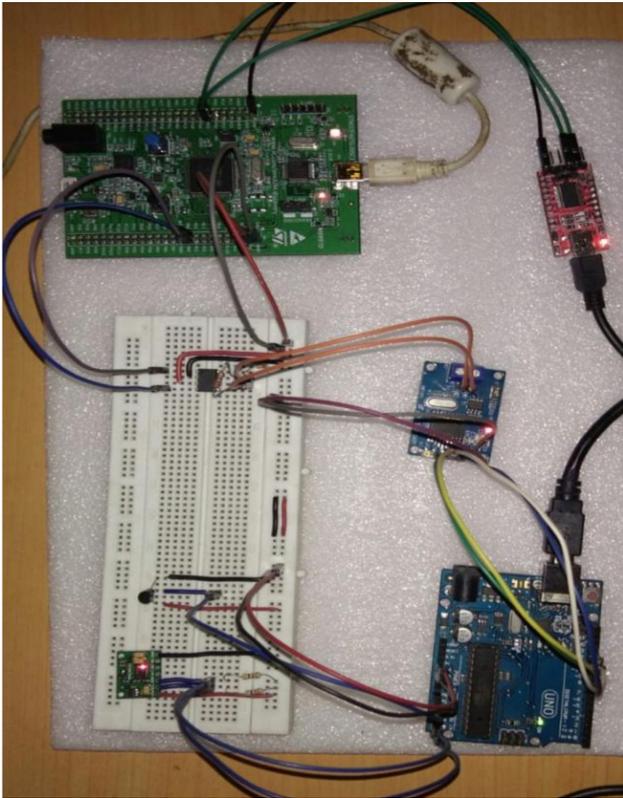


Figure 11. Process flow of CAN data logger and GUI.



**Figure 12.** The hardware setup of the proposed system

A registration page is provided for every new patient to enter his/her details. It creates an excel sheet to store data of that particular patient.

**Figure 13.** Registration page for new user

After creating an account, the patient has to login with a name and ID to log the health parameters.

**Figure 14.** Login page for patient

When a user is successfully logged-in, the patient has to place the index finger on the sensor. The GUI provides the option to measure and save the body temperature, blood oxygen level, and heart rate. It also gives an overall inference of patients' health. The patient can also view his previous records.

**Figure 15.** Option to log, save and view records

The doctor can view details of any patient by logging-in though with his credentials and enter the patient name and ID. Figures 13 to 17 show the snapshots of the patient registration and data logging procedures.



Figure 16. Interface for doctor

Date	Time	Temperature(°C)	Blood Oxygen(%)	Heart Rate(bpm)
Tuesday, August 4, 2020	4:05:12 PM	29.55	98	63
Tuesday, August 4, 2020	4:10:25 PM	29.17	100	71
Tuesday, August 4, 2020	4:23:28 PM	32.5	100	78

Figure 17. Logged data saved to excel sheet

## 6. Conclusions

The paper successfully shows the implementation of the patient health parameter monitoring system, which is capable of measuring the body temperature, heart rate, and oxygen level of the patient. The device has been interfaced with a GUI capable of requesting the sensor data and adding the requested data to the excel sheet to generate the report with timely logged data. The system illustrated is designed to keep the present coronavirus pandemic situation in mind with an intention to provide a robust, secure, and low-cost solution for keeping track of both the patients in hospital and patients under home isolation.

## References

[1] Priyanka Kakria, N. K. Tripathi, and Peerapong Kiti-pawang, "A Real-Time Health Monitoring System for Remote Cardiac Patients Using Smartphone and Wearable Sensors", International Journal of Telemedicine and Applications Volume 2015, Article ID

373474, 2015.

[2] M. Saranya, R. Preethi, M. Rupasri and Dr. S. Veena. "A Survey on Health Monitoring System by using IOT", International Journal for Research in Applied Science & Engineering Technology, Volume-6, Issue-III, , pp. 778-782, 2018.

[3] C. Premalatha, R.P. Keerthana and R. Abarna. "Human Health Monitoring System", International Research Journal of Engineering and Technology, Volume-6, Issue-1, pp. 914-916, 2019.

[4] Prasad Lavgi and Mr. Lakshaman Kora, "Patient Health Monitoring System using CAN Protocol", International Research Journal of Engineering and Technology, Volume-6, Issue-6, pp. 394-398, 2019.

[5] K.P. Swain, M.V.S. V Prasad, J. Sahoo, S. K. Sahoo, G. Palai, "Patient Monitoring System using CAN and Android", Journal of Research in Science, Technology, Engineering and Management, pp. 101-105, 2016.

[6] Karumuri Anusha Reddy and Narendra Babu Tatini, "Monitoring of Patients Health in Hospitals using CAN Protocol", International Journal of Innovative Technology and Exploring Engineering, Volume-8, Issue-6, April 2019, pp. 1556-1559.

[7] Ladlennon C. Banuag and Joseph Karl G. Salva, "Development of a Patient Monitoring System for Hospital Wards Employing Zigbee Technology and CAN Protocol", 2019 2nd World Symposium on Communication Engineering (WSCE), Nagoya, Japan, pp. 112-116, 2020.

[8] STMicroelectronics, "STM32F405xx and STM32F407xx, ARM Cortex-M4 32b MCU+FPU, 210DMIPS, up to 1MB Flash/192+4KB RAM, USB OTG HS/FS, Ethernet, 17 TIMs, 3 ADCs, 15 comm. interfaces & camera", DocID022152 Rev 8 datasheet, 2016.

[9] STMicroelectronics, "STM32F405/415, STM32F407/417, STM32F427/437 and STM32F429/439 advanced Arm®-based 32-bit MCUs", RM0090 Rev 17 reference manual, 2018.

[10] Microchip Technology Inc., "MCP2551, High-Speed CAN Transceiver", DS21667F datasheet, 2010.

[11] Microchip Technology Inc., "MCP2515, Stand-Alone CAN Controller With SPI™ Interface", DS21801D datasheet, 2005.

[12] Maxim Integrated Products, Inc., "MAX30102, High-Sensitivity Pulse Oximeter and Heart-Rate Sensor for Wearable Health", 2015.

[13] Steve Corrigan, "Introduction to the Controller Area Network (CAN)," Texas Instruments, SLOA101B application report, 2016.