IoT HEALTH MONITORING SYSTEM

GROUP NAME: THINGSPEAK

MOHAMMAD AHMUDUR RAHMAN MAHDI (1828313)

ZUBAIER MOSTAFA (1913515)

SUBMISSION DATE: 09/01/2023

SECTION 1
EMBEDDED SYSTEM DESIGN
INTERNATIONAL ISLAMIC UNIVERSITY MALAYSIA

Executive Summary

With improving technology, different methods of administering health care advice have been on the rise, be it remote or live. In this project, an IoT-based patient health monitoring system is planned and built using Arduino and ESP8266. The system has the ability to gather and communicate patient health information to Thingspeak, an IoT cloud server that allows for remote patient monitoring and real-time recording of the patient's health state. The health professional may recommend the best course of therapy or take urgent action in the event of an emergency based on the facts they have received. It can be implemented to help individuals currently on bed rest on the advice of medical professionals at their homes which opens up the scope of implementation of this project.

Table of Contents

1. Design Problem and Objectives

Patients staying at home after surgery or prescription often need to be checked for heartbeat pulse, body temperature, etc. If there is any abrupt change there is no way for the doctors or professionals to know unless informed through a phone call or take the patient directly to the hospital. This situation is critical for people living in rural areas. Hence, a real-time health monitoring system is required through which doctors or professionals will always stay informed about the patient's condition in real time.

Project Objectives:

- Collect real-time heartbeat in minutes or BPM using a pulse sensor.
- Collect body temperature data using a digital temperature sensor.
- Use ESP8266 to connect the system to the internet and send health data to the Thingspeak IoT server.
- Show real-time data on a 16x2 LCD display.

2. Detailed Design Documentation

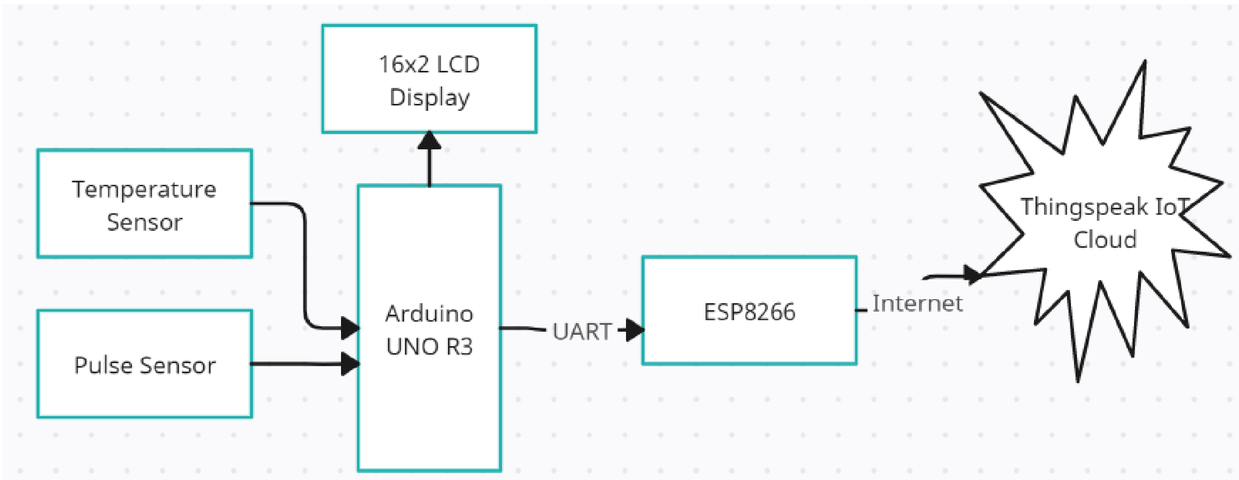

Figure 1: Overall system block diagram.

The IoT-based patient health monitoring system employing an Arduino and ESP8266 is shown in figure 1. BPM and body temperature are measured, respectively, by the pulse sensor and the LM35 temperature sensor. The 16x2 LCD Display on the Arduino shows the code once it has been processed. The ESP8266 Wi-Fi module establishes a wireless connection and transmits data to the IoT device server. Thingspeak is the IoT server in use here. The data may also be seen by logging into the Thingspeak channel from anywhere in the globe.

2.1 Pulse Sensor

A well-designed, low-power, plug-and-play heart-rate sensor for the Arduino is available as the Pulse Sensor. The finger should be placed on the sensor's front, near the heart logo. A pulse sensor operates by flashing green light on the finger and detecting the quantity of light reflected using a photosensor. The green light may be absorbed by arterial blood's oxygenated hemoglobin. The more hemoglobin there is in the blood, the redder it is, and the more green light is absorbed. Blood is pumped through the finger with each pulse, resulting in a change in the quantity of light reflected, which in turn creates a waveform at the photosensor's output. We rapidly start to get a readout of the heartbeat pulse as the green light continues to shine and the photosensor continues to gather data. In order to generate a signal that is substantially bigger, cleaner, and simpler to detect, the generally tiny and noisy photosensor signal is routed through an R/C filter network and then amplified using an Op-Amp.

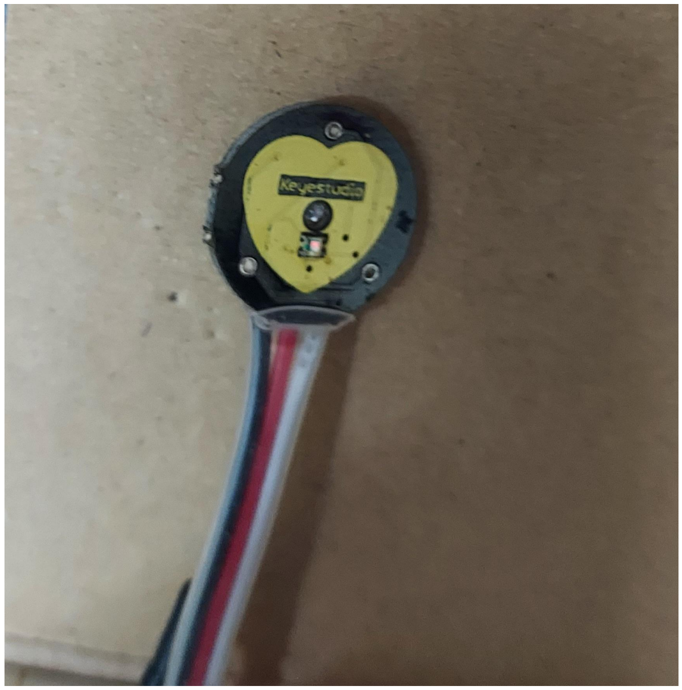

Figure 2: Pulse Sensor.

2.2 LM35 Temperature Sensor

The thermometer LM35 produces an analog output voltage that is proportional to the temperature. It delivers output voltage in degrees Celsius (Celsius). No additional calibration circuitry is needed. The LM35 is 10 mV/degree Celsius sensitive. The output voltage rises as the temperature climbs. For instance, 300 mV denotes 30 °C. It is a 3-terminal sensor that is used to gauge ambient temperatures between -55 °C and 150 °C. The output of the LM35 is more accurate than that of the thermistor.

Figure 3: LM35 Temperature Sensor.

2.3 Generic ESP8266

The generic ESP8266 module is a tiny microcontroller board that, like other microcontrollers, contains RAM, ROM, a clock, communication protocols, and I/O ports. It requires programming to work. Since it lacks a USB interface, we must upload a programme via its UART pins. It supports UART / serial communication protocol, runs at 3.3V, and has (just) two GPIO pins that we won't be using. Through its UART ports, the Arduino gathers and transmits patient data to the ESP8266. The ESP8266 will establish a connection with the pre-programmed server and transmit the data there.

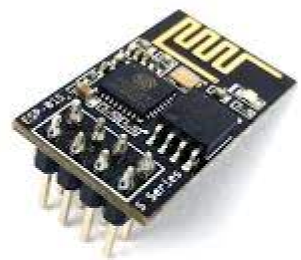

Figure 4: ESP8266 Wifi Module.

2.4 Circuit Diagram & Results

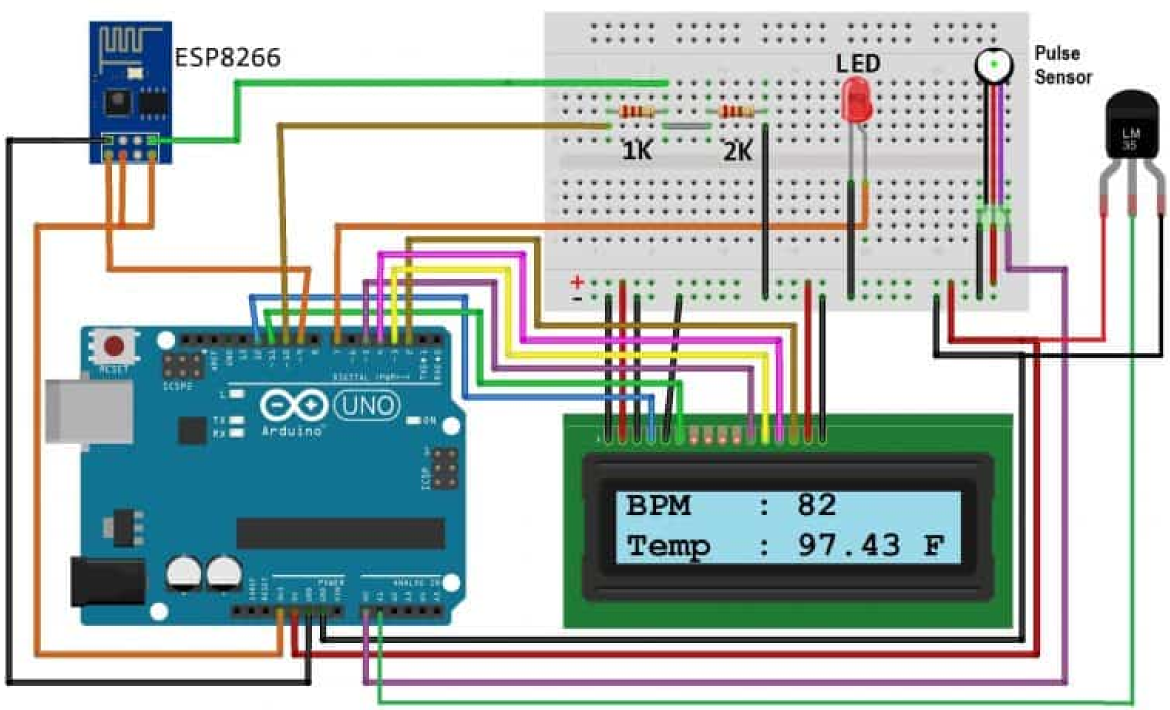

Figure 5: Circuit Diagram for designing IoT-based patient health monitoring system.

The steps to construct the circuit are shown below:

I. Pins 4, 6, 11, 12, 13, and 14 of LCD are connected to Digital Pins 12, 11, 5, 4, 3, and 2 of Arduino.
II. Pins 2, and 15 of LCD are connected to VCC.
III. Pins 1, 3, 5, and 16 of LCD are connected to GND.
IV. The LED is connected to digital pin 7 of Arduino through a 220-ohm resistor.
V. Make a voltage divider by connecting 2.2K & 1K ohm resistors. This will convert 5V to 3.3V which is suitable for the RX pin of ESP8266. Thus the RX pin of the ESP8266 is connected to pin 10 of Arduino through the resistors.
VI. The TX pin of the ESP8266 is connected to pin 9 of the Arduino.
VII. The Pulse Sensor output pin is connected to analog pin A0 of Arduino and the other two pins to VCC & GND.
VIII. LM35 Temperature Sensor output pin is connected to A1 of Arduino and the other two pins to VCC & GND.

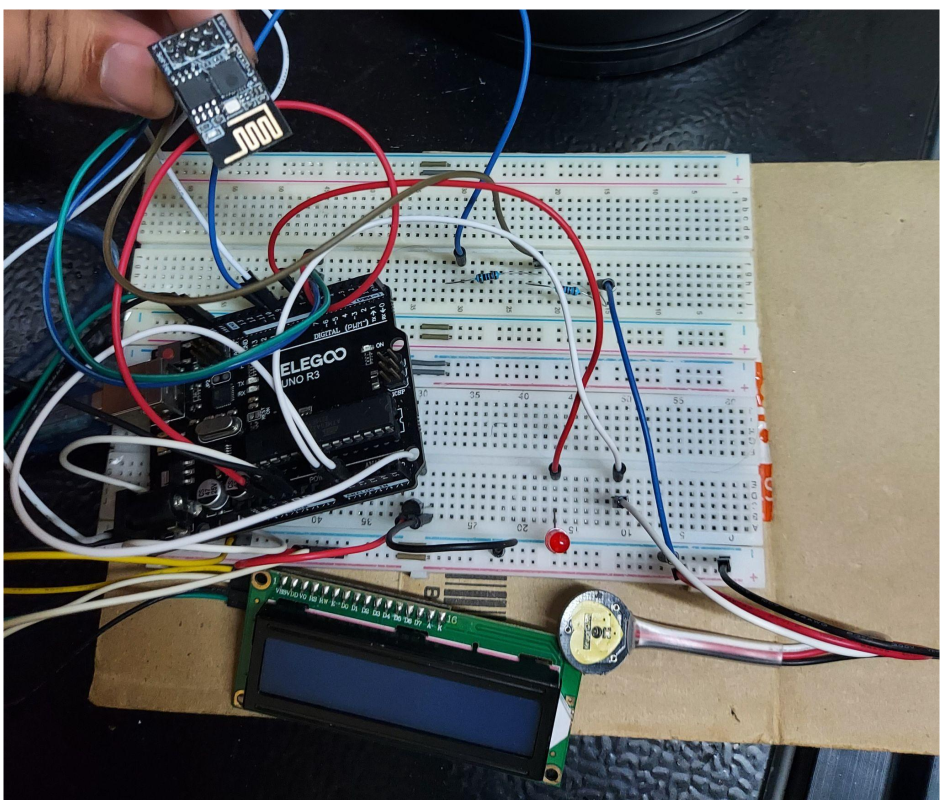

Figure 6: Experimental diagram.

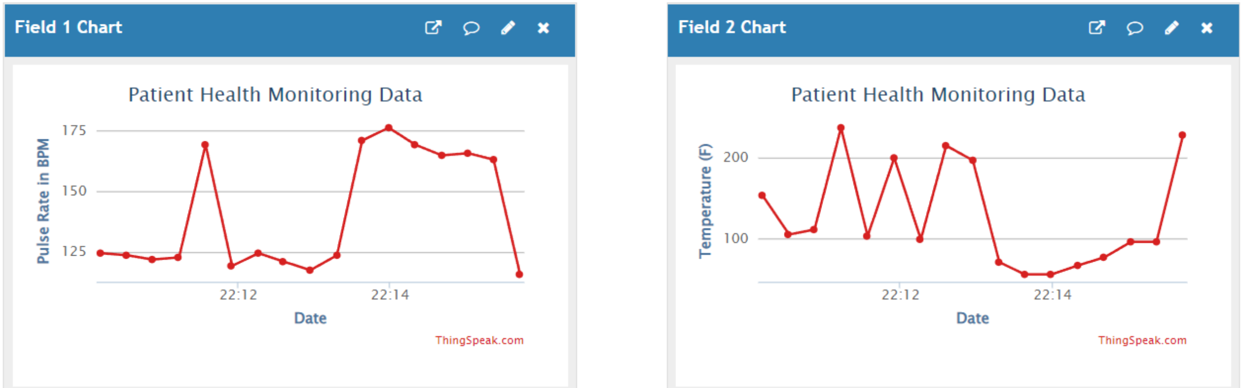

Figure 7: Real-time monitoring through ThingSpeak Server.

The figure above shows 17 entries made and their timestamps in the ThingSpeak IoT Server.

2.5 Coding

The Arduino code for the health monitoring system can be found at the following link: https://github.com/AhmadurRahmanMahdi/EmbeddedSystem/tree/main/IoT_Health_Monitoring_System

2.6 Communication

The communication between the microcontroller and PC is achieved through the USB port. The ESP8266 does not have a USB port. So, the communication between the microcontroller and ESP8266 is done through UART.

2.7 Interrupt

Our code made use of interrupts in various places. Software interrupts were used after reading the sensor data. These interrupts allow the processor to multitask, changing priorities from tasks as required, going from low to high priority ones.

2.8 Memory

Our project made use of the basic mcu memory, which is flash memory and also the SRAM. Flash memory is used when any physical data is stored which does not change its value during program execution. The SRAM however, is used for data that is read and written repetitively for the mcu. In our project, it is the sensor values that require this type of memory.

3. Safety

The entire circuit does not require any high-voltage connections. It must, however, be noted that the ESP8266 operates at 3.3V, not at the regular 5V level of the Arduino Uno. Hence, it is highly recommended to use a voltage divider while providing a power supply to the ESP8266 through the 5V voltage rails. The TX port of the ESP8266 will not send any data to the IoT-based server if it is not being operated at a 3.3V level. These precautions will guarantee safety against any possible short circuits of the ESP module.

4. Conclusion

We have come to the end of our mini-project. This was a very good topic as it introduced us to the world of IoT. The topic was selected because it has very wide and good implementations in healthcare which expands day by day because of the various advancements in technology. The application is not limited to healthcare. The same approach can be undertaken for data collection in other critical sectors such as the automobile or aerospace industry. Replacing the current sensors with ones that are relevant to that particular application will incorporate IoT into that specific industry. IoT is the future of everything and it is a very interesting area one can delve into as a hobby or further one's career.

References

IOT based Health Monitoring System: Arduino project. DIY Electronics Projects. Retrieved January 8, 2023, from https://electronics-project-hub.com/iot-based-health-monitoring-system-arduino/

.IOT based patient health monitoring using ESP8266 & Arduino. How To Electronics. Retrieved January 8, 2023, from https://how2electronics.com/iot-patient-health-monitoring-system-esp8266/

LM35 Temperature Sensor: Sensors & Modules. ElectronicWings. (n.d.). Retrieved January 8, 2023, from https://www.electronicwings.com/sensors-modules/lm35-temperature-sensor

Last Minute Engineers. (2022, November 25). *In-depth: Detect, Measure & Plot Heart Rate Using Pulse Sensor & Arduino*. Last Minute Engineers. Retrieved January 8, 2023, from https://lastminuteengineers.com/pulse-sensor-arduino-tutorial/