System for Remote Health Management of Quarantined Patients

Team 23 — Arnav Ahluwalia, Rohit Kumar and Ishaan Datta
ECE 445 Project Proposal — Spring 2021
TA: Yichi Zhang

1. Introduction

1.1. Objective:

The COVID-19 pandemic has brought to light a wide variety of infrastructure shortcomings in the healthcare industry. Hospitals have been unable to consistently provide services to the high influx of patients, a majority of whose condition isn't critical enough to be admitted. Its been an ongoing struggle to prioritize individuals who are at-risk, leading to a lack of equal access to medical staff and facilities. Generally speaking, this problem can be broken down to a naturally existing low physician-to-population ratio. Whether we consider first-world regions such as New York/Chicago, or third-world countries such as India, high rates of COVID concurrent with high population densities are the major factors that influence how overwhelming the situations would become for the medical infrastructure. Our objective is to develop a low-cost, medical device coupled with a scalable backend service which is capable of monitoring the status of patients while they're at home, allowing for medical professionals to remotely and efficiently keep track of their vital parameters.

1.2. Background:

During the pandemic, monitoring the health of home quarantined patients at scale enables provision of timely medical support. Enabling medical authorities to remotely monitor the health of COVID-19 patients would lessen the load on hospitals and help divert medical resources well in time to people who need it the most. This is particularly relevant for places where the outbreak has progressed to an extent that not enough beds/medical facilities are available to cater to every patient and triaging is being carried out- i.e. medical personnel must tend to higher-risk/seriously ill patients. This is particularly true for developing countries (places where the medical infrastructure isn't expansive enough to cover all patients).

1.3. High-level requirements:

1.3.1. The device should be compact, affordable and be able to accurately measure patient vitals like pulse rate, blood oxygen, temperature and blood pressure with an accuracy greater than 95%.

1.3.2. The device should communicate with an internet-based backend service via WiFi to transmit the measured readings with zero data loss.

1.3.3. The backend service should store patient data securely and help medical personnel in monitoring patient status on near-Realtime basis.

2. Design:

Our design for the patient monitoring system is composed of three overall subsystems. The first comprises of a microcontroller, sensors and a power supply. We'll be using the ATmega328 microcontroller as it has a low cost, is easy to program and has sufficient digital and analog I/O pins to interface with the sensors. All other components have likewise been selected to be cost-efficient and compatible with our microcontroller. The second subsystem is our WiFi module, used by our controller to connect to the patient's home WiFi network. It would enable transmission of sensor data over the internet. The last subsystem consists of a cloud-based backend service intended for receiving and storing health data from the sensors. It would also provide a front-end for medical personal to monitor the health of patients remotely.

2.1. Block Diagram and Physical Design:

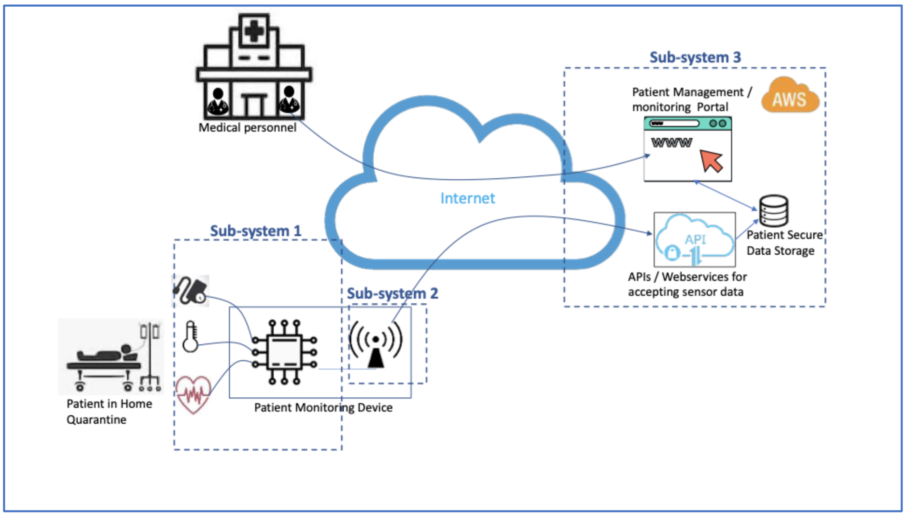

Medical personnel
Sub-system 3
Patient Management / monitoring Portal
AWS
www
Internet
Sub-system 1
API
Patient Secure Data Storage
APIs / Webservices for accepting sensor data
Sub-system 2
Patient in Home Quarantine
Patient Monitoring Device

Figure 1 : Overall System

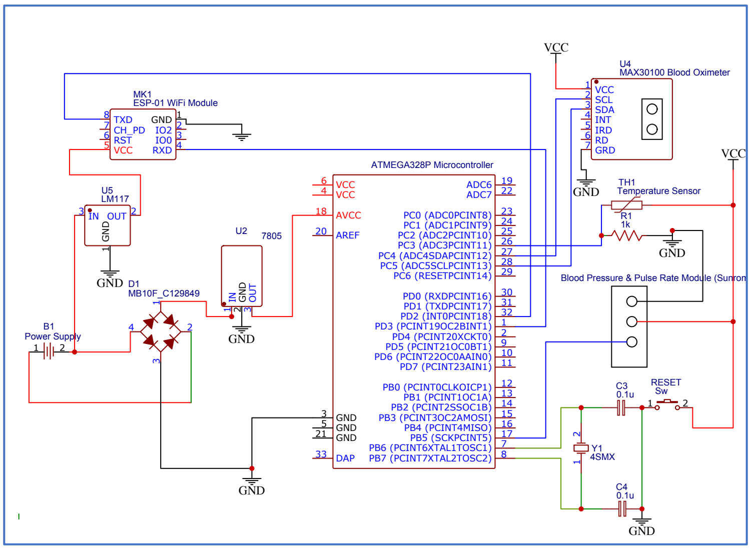


Figure 2: Subsystem 1 & 2 Schematic

Functional Overview:

2.1.1.**Subsystem** 1:

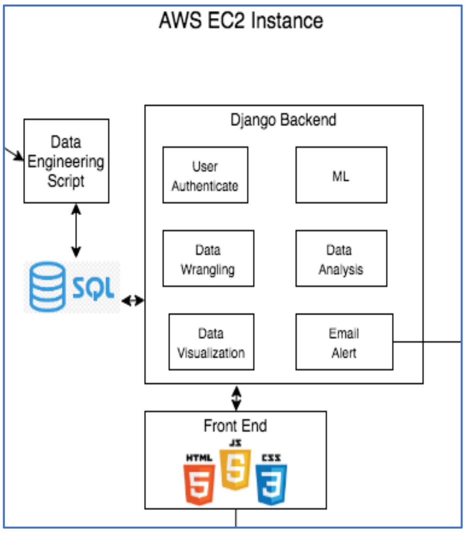
2.1.1.1. Sensors for measuring patient health data: We propose to use Heartbeat, BP, Blood pulse oximeter and Temperature measurement sensors

2.1.1.2. Atmega Microcontroller for capturing health sensor inputs: The microcontroller would form the heart of the system that captures sensor data and does suitable protocol conversions as needed to send the data periodically to the backend application.

2.1.2. **Subsystem** 2: Internet connectivity interface: We would use a WiFi interface module for providing internet connectivity to the device.

2.1.3. **Subsystem** 3: We will implement an application with a cloud-based Backend which would carry out user provisioning (for patients as well as health professionals), storing, capturing and the display of patient data via a website- this application would essentially store the patient data, permit analytics on patient data and display it on a website for monitoring by a medical doctor. The platform should also have an email-based alerting mechanism that gets triggered by tunable health parameters obtained from the device, allowing it to indicate if/as soon as a patient needs medical attention. The Self-service web portal on the cloud to register patients and link their personal details with the unique device ID present with them. Web portal on the cloud for authorized medical personnel to view patient data. The portal would have sufficient security controls – authenticated access and encrypted data storage to keep patient data secure.

Figure 3: Subsytem 3



2.2. Block Requirements:

2.2.1. Subsystem 1- Monitoring Device

2.2.1.1. **Power Supply:** A power supply module- 230V AC to 5V DC would be required to power the microcontroller and all other IC's in our device. Two voltage regulators (LM35 and LM1117) would be used to supply 5V and 3.3V respectively. *Requirement: The WiFi module and blood-oxygen sensor needs 3.3V; all the other components need 5V DC*

2.2.1.2. **Microcontroller:** Once the device has been turned on, the ATmega328 controller would receive input from the four sensors and send the data to our software backend using the ESP8266 WiFi module. A buzzer would sound if the vital parameters aren't within a normal predefined range. The flash memory of our microcontroller would be preloaded with firmware for carrying out the above

functions.
Requirement 1: Capture data from the sensors and do required conversions, eg- Analog to Digital.
Requirement 2: Check if sensor readings are within certain thresholds; if not, sound a buzzer.
Requirement 3: Utilize the WiFi module (ESP8266 SoC) to transmit the patient's vital parameters to the backend service using HTTPS over TCP/IP protocol.

2.2.1.3. Sensors

2.2.1.3.1. **Temperature Sensor**: We've chosen to use the TI- LM35 temperature sensor, which provides analog readings within 0.5 °C (at 25°C). *Requirement: Must be able to consistently measure a patient's temperature with a maximum deviation of ±5% as compared to a clinical thermometer and display the results in degree centigrade.*

2.2.1.3.2. **Blood Pressure and Pulse Rate Sensor**: An over-the-counter blood pressure sensor would be utilized to measure patient's systolic/diastolic blood pressure and pulse rate. These readings would be output to the microcontroller using a UART interface.
Requirement 1: Must be able to consistently measure the patient's blood pressure and pulse rate with > 95% degree of accuracy.
Requirement 2: The over-the-counter device should have a serial interface (UART) to provide external output to the microcontroller.

2.2.1.3.3. **Blood Oxygen Sensor**: We shall use the MAX30100 Pulse Oximeter Integrated Circuit to measure the patient's blood-oxygen saturation.
Requirement 1: Must be able to consistently measure a patient's blood oxygen levels with > 97% degree of accuracy.

2.2.1.4. **Subsystem 2- Wifi Module**: For enabling connectivity to the internet, our microcontroller would interface with the ESP8266 microchip- equipped with a TCP/IP stack, this would be utilized to transmit the readings from the microcontroller to the backend service via home wifi network. *Requirement 1: Interface with the microcontroller to allow the SOC's configuration in order to provide the required functionalities, as well as receive patient data. Requirement 2: The ESP8266 would be programmed to run a mini HTTP server, allow the end user to log on to the device using a browser, and set the home SSID and password.*
Requirement 3: Communicate over HTTP to the backend service and transmit the patient data in the required format.

2.2.2. **Subsystem 3- Backend service**: The backend will run inside a docker container within an AWS EC2 instance.

2.2.2.1. **Data Engineering**: We will have a python script to receive the sensor data over HTTP and store it in a MySQL database. Patient's personally identifiable information entered into the system by medical personnel shall be stored using AES 256 bit symmetric encryption.
Requirement 1: Sensor data received from the web-application will be parsed and stored into the database.

Requirement 2: Patient personally identifiable information (names, addresses, telephone numbers, email addresses) will be encrypted prior to storage in the database.

2.2.2.2. **Django Backend:** The core of our backend will be run on a Django web application. This is where we have our portal, user authentication, data visualization & analytics, a script that performs an email alert system whenever medical attention is needed.
Requirement 1: The backend will implement RESTful APIs to accept sensor data .
Requirement 2: The portal for medical personnel will have authentication, searching, visualization and alerting features.
Requirement 3: The backend software will be patched against any vulnerabilities that may compromise patient data.
Requirement 4: Audit trails (logs) will be maintained for access to the portal.

2.2.2.3. **Front End:** The front end will display the portal and data analytics. We will use HTML, CSS and Javascript.
Requirement 1: The portal should be capable of being rendered in Chrome/ Internet Explorer browsers.

2.3. **Risk Analysis:** The Blood Oxygen diagnostic is apparently the most important for Covid patients. A Blood Oxygen reading below 90-92% indicates that the patient needs medical care in a hospital. While BP measurements are in scope, we are not planning to have ECG as it would be difficult for a layman to conduct an ECG test. Measurement technique and accuracy of the BP (and upto an extent Blood Oxygen) sensors themselves is actually a risk that has to be taken into account. BP measurement home devices have been often known to have 5% or more error margin compared to a mercury based one at clinics. Usage technique by patients also affects the accuracy of the measurement. Some of the recommended mitigations around these risks are - (a) Use BP home devices that have listed error rates be low 5%. (b) Teach the patient in correct use of a device - say rest for five minutes before BP measurement, take a Blood Oxygen reading from one finger in both hands (to cater to some patients having circulatory issues) etc. Since we will be storing a series of measurements for each patients over time, we shall also be able to calculate the deviation of successive reading from a baseline for each patients. Percent deviation from baseline in in itself can be a metric that can be used assess deterioration of condition of a patient- despite, say a fixed + or -5% error in the home measuring device compared to a clinic device.

3. **Safety and Ethics**:

3.1. **Safety:** Safety considerations for the project basically include the standard set of precautions taken while working on electrical engineering projects. The device would not be working above 5V DC for any of its components. Precautions would be taken as per our lab guidelines for soldering the different components. Due to the ongoing pandemic, team members would ensure maintaining appropriate social distancing and sanitary measures while working together or in the lab.

3.2. **Ethics, Security & Privacy:** One of the biggest concerns with recording, transmitting and storing patient health data is maintaining privacy and security. The US HIPAA act, EU’s GDPR and other such privacy regulations treat patient health data as “sensitive personal information”, needing the highest level of security and privacy. This includes ensuring-

3.2.1. **Security of Data in Transit:** We have two options for security of health data in transit - encryption and anonymization.

3.2.1.1. **Encrypting data in transit-** Modules exist in the ATmega328 microcontroller which allow encryption of data. An option could be pre-provisioning the same public key in all devices and having the corresponding single private key securely stored in the cloud. On the device, the public key can be used for encrypting all outbound data.

3.2.1.2. **Anonymizing outbound information**- We plan on adopting a simpler approach of anonymizing the transmitted data such that it does not contain any personal details (like names, addresses, phone number, etc.). During device provisioning stage, the patient/ assistant would need to enter personal details on a provisioning (secure) website and link the device unique number with it (the unique number can be printed on the side of each device). Thereafter, all outbound data has just this number along with sensor metrics - no personal details would be sent.

3.2.2.**Security of Data at Rest:** At the cloud end, the data would be stored in an encrypted state once it is linked with personal details. We propose to use AES 256-bit symmetric encryption.

Criteria for success:

- Patient data captured by the device, streamed over the Internet to the cloud-based application and visible on a website to authorized personnel.
- Accurate monitoring of patient vitals and triggering of the alerting system based on predefined parameters.