Health Artificial Intelligence Company

SYMON: Symptom Monitor of COVID-19

FINAL REPORT

Prepared for Assoc. Prof. Fatih Kamışlı

Report Submission Date: 11.06.2021

Company Contact Person: Okyanus Oral
Contact Info: okyanus.oral@metu.edu.tr
+905539109260
Address of the Company: Üniversiteler Mah. METU Electrical and Electronics Dept., 06800 Çankaya/Ankara

Table of Contents

Company Information

HAI-CO. is a company that designs, manufactures and markets a wearable COVID-19 symptom monitor. It is founded by five electrical and electronics engineers from METU. Non-hierarchical functional organizational structure of HAI-CO. can be observed in Figure 1.

Figure 1: The organizational structure of HAI-CO.

The company is composed of signal processing engineers, communication engineers, a hardware-oriented RF design engineer, a software engineer, and an embedded system engineer who are eligible to work on SYMON.

Vision

HAI-CO. aspires to be an international technology company that endeavors to expand and earn people's trust in the fields of health technologies, security systems, and transportation with its unique electronic-based, efficient, and environmentally friendly solutions.

Mission

HAI-CO. aspires to develop new and breakthrough technologies in the field of electronics, resulting in effective, viable, and innovative solutions to humanity's everyday issues.

HAI-CO. and Environment

HAI-CO. positions itself as a sustainable and environmental friendly company. It is concerned with ways to conserve natural resources and recycle them in a way that is safe for human health and the environment.

HAI-CO. and COVID-19

In parallel with its vision and mission, HAI-CO. seeks to contribute the regulation of COVID-19 global pandemic.

Executive Summary

COVID-19 threatens the human lives all around the world with its high transmission rate since December 2019. Until now, 174.9 million people has been infected and death toll has climbed up to 3.77 million globally due to COVID-19 [1]. In order to reduce the transmission rate, people should be warned to take some precautions and informed about the infection risk. to help people with monitoring symptoms and taking proper precautions, Health Artificial Intelligence Company (HAI-CO.) took action and designed and developed SYMON, COVID-19 symptom monitor as a chest strap.

The chest strap has an ability to detect three of the important symptoms of COVID-19, which are cough, fever and palpitation using its three sensors, a microphone, a temperature sensor and an ECG sensor. Additionally, SYMON enables users to log in other symptoms of COVID-19 which cannot be detected using sensors. The overall system consists of a mobile application and a comfortable chest strap carrying these three sensors, a microcontroller and a power bank. After all symptom data are available, the risk of the user is estimated to warn the user. Also, all sensor data taken since the previous data transfer are sent to mobile application to inform the user about its current symptoms and risk. All connections between the chest strap and the mobile application is realized by the embedded Bluetooth Low Energy (BLE) module of the microcontroller.

After the system is powered, SYMON completes different tasks. These are processing sensory data and undetectable symptoms data, estimation of the risk, information transfer between the chest strap and the mobile app and finally, warning and informing the user visually. While sensory data is processed, cough detection is performed applying a correlation filter to the audio signals which is continuously received from the microphone. Fever and palpitation detection are carried out by temperature sensor and heart rate sensor, respectively. Undetectable symptoms are logged in by the user through the mobile application. Then, the risk is estimated reliably by using random forest classifier that is trained by a data set that is created by the team of HAI-CO. In order to inform and warn the user, a user friendly mobile application is designed. Communication link between the chest strap and mobile application is achieved by the BLE module. Finally, all tasks are

integrated as a whole.

Before starting the implementation, the requirements for the product performance are set as an overall cost less than $80, compatibility with different activities, user friendly usage and long battery life in order to achieve an accurate and reliable monitoring device. The further test results have shown that the accuracy of cough detection is 99%, error of temperature measurement and heart rate measurement are ± 0.3 °C and 2.82%, respectively, and risk estimation is completed with an accuracy of 81%. Also, resulting cost is $47, which is less than the set value. The requirement for battery life is satisfied with a charge cycle of 55.4 hours. Because of its 160 g weight, the product is compatible with different activities. Also, mobile application is aimed to be user friendly with its visual indicators and buttons to direct user. Therefore, the objectives are met successfully both by individual subsystems and the integrated overall system.

HAI-CO. intends to provide SYMON as a wearable device in the shape of a chest strap along with a mobile application. The product is ready to be launched.

Chapter 1

Introduction

From its start in December 2019 up to today, COVID-19 has taken the lives of 3,771,669 individuals worldwide [1] becoming a major pandemic and changing social and economic structures significantly.

Ever increasing number of patients made it an urgency to regulate human interaction to sustain occupancy of hospitals below their maximum capacity. Health Artificial Intelligence Company, HAI-CO., offers a solution to this problem. With a wearable COVID-19 Symptom Monitoring Device, SYMON, (see Fig.1.1) users are informed to take necessary precautions if they show the symptoms of the disease. Hence, interactions with the infected can be regulated by the users.

(a) **Front View of SYMON.** (b) **Rear View of SYMON.**

Figure 1.1: Views of SYMON.

SYMON is a wearable chest strap designed to detect the main COVID-19 symptoms: cough, fever and palpitation [2] via its microphone, temperature and ECG sensors respectively. Furthermore, users can enter undetectable COVID-19 symptoms through the user interface of the developed mobile app (see Fig. 1.2). At the end of each day, the device outputs COVID-19 risk estimation score and informs the user to take necessary precautions if needed. Detected symptoms and risk estimations are also shown to the user via the interface of the mobile app.

Overall, SYMON is fully functional and its integration is complete. All subsystems are compatible and set requirements are met.

In this report; entire design effort is summarized, finalized product's spec-

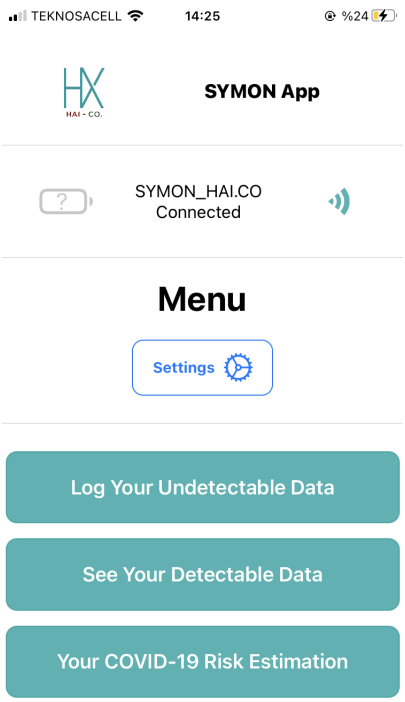


Figure 1.2: Menu View of SYMON App.

ifications and performance results are shown, deliverables and user manual are presented. In Chapter 2, requirement analysis for proper operation is discussed on system and subsystem levels. Specifications and detailed overview of the design process for the overall system and each subsystem are shown in Chapter 3, where compliance with the set requirements are presented in Chapter 4. In order to address the environmental, social and safety related issues of the product, in Chapter 5 resource management and impacts are discussed, it is followed by the deliverables where user manual shown in Chapter 6. At the end, in Chapter 7, product development is concluded.

Chapter 2

Requirement Analysis

2.1 System Level Analysis

SYMON is designed to be a wearable self-monitoring device for detecting the symptoms of COVID-19. In order to fulfill this duty, the requirements that SYMON needs to satisfy include,

- convenient charge cycles that will last for at least 16 hours.
- a reasonable cost for the final product, less than $80.
- being lightweight, less than 200 grams with evenly distributed weight and being suitable to the daily life of a person, i.e. compatibility with different activities.
- being user friendly with an easy and intuitive usage.
- gathering all the relevant symptoms of COVID-19 which are cough, fever, tiredness, shortness of breath, headache, muscle and/or joint pain, chest distress, nausea, vomiting, loss of taste and palpitation.
- providing accurate results of symptom occurrences and COVID-19 risk estimations as well as reliable precaution messages for the relative risks.

2.2 Subsystem Level Analysis

2.2.1 Cough Detection Subsystem Requirement Analysis

One of the major symptoms of COVID-19 is dry cough [2], thus for the development of a COVID-19 symptom monitoring device, detection of coughs is the top priority.

At the beginning of the development; at least 0.7 accuracy, precision, recall and F1 score was set as the performance requirements for a reliable cough detection. However, as the project progressed, it was observed that, for a long duration of active usage, specificity (rejection of false positives) of the algorithm is of great importance. Total duration of possible coughing within a day is considerably shorter than the operation duration of SYMON (16

hours). Hence, if the ratio of misclassification of non cough sounds is high, false positives can easily dominate true positive results. Therefore, ratio of misclassified non-cough sounds to all trials should be minimal, on the order less than 0.001.

2.2.2 Fever Detection Subsystem

Since fever is one of the most common symptoms of COVID-19, SYMON should detect the body temperature with high accuracy and precision. For this purpose, maximum $\pm$ 0.5 °C error was set as a system requirement for the measurement.

2.2.3 Heart-Rate Measurement Subsystem

According to the objectives that were set previously [3], the primary objective for heart rate measurement is ease of use and the compatibility with a wearable device in the form of a chest strap. Therefore, it is essential to select the correct sensor. As for a performance requirement, the maximum approximation error is set to be 5%.

2.2.4 Communication Subsystem

A reliable communication channel is vital between the mobile application and SYMON in order to present data to and acquire data from the user. Therefore, communication subsystem is designed. The subsystem should consume low power since the chest strap has limited power and it must assure the correct data transfer without a loss.

2.2.5 Mobile Application Subsystem

Detectable symptom data of COVID-19, fever, cough, and palpitation are handled by SYMON. However, there are more symptoms that cannot be detected by SYMON but critical to detect having COVID-19. Thus, the related data should be acquired manually for such symptoms. In addition, all the data of detectable and undetectable symptoms, risk score and precaution messages should be displayed to the user via a visual user interface. A mobile application is an appropriate option for this purpose, since in today's world, consumers are reaching for mobile devices numerous times every day to use mobile applications. Therefore, a mobile application subsystem is designed.

The most prominent requirement for this mobile application is that it should be capable of transmitting and receiving the data between the self-monitoring device, SYMON and mobile device via the use of Bluetooth Low Energy (BLE) technology as specified in Communication Subsystem. This must be done in the background in order not to lose any useful data about COVID-19 symptoms.

Another requirement is that the mobile application should have a convenient user interface with simple navigation. For this purpose, visual tools such as scroll bars, buttons, toggles should be included. Additionally, the design should be neat, simple and visually appealing.

Performance is another requirement that is critical for robust mobile application. Its CPU usage should be less than 30%, while memory usage should not exceed 50 MB to maintain high performance and low battery usage of the mobile device.

2.2.6 Risk Estimation Subsystem

As previously mentioned in [3], SYMON is a COVID-19 monitoring device and it should provide an estimated COVID-19 infection risk.

SYMON must achieve a convenient accuracy and sensitivity (recall) score. Sensitivity is an issue of high importance for the use case of SYMON where mislabeled positive COVID-19 users may spread the virus even more. To prevent this, on a data set that can statistically represent a sufficient portion of the population, 80% sensitivity and accuracy are set as important performance requirements for this subsystem.

Chapter 3

System & Subsystem Specifications

3.1 Overall System

3.1.1 General Overview

SYMON the COVID-19 monitor, founded by HAI-CO., consists of a wearable chest strap and a mobile application. SYMON is capable of detecting the wearer's coughs, body temperature and heart-rate via the utilized sensors on the chest strap. The mobile application of the SYMON enables the users to enter the undetectable symptoms and severity levels. In Figure 1.1 and 1.2, the chest strap and the interface of the mobile application is presented. By using a Bluetooth Low Energy (BLE) communication between the mobile device and the chest strap, SYMON merges these symptoms to provide an estimated COVID-19 infection score and precaution information to the user. By doing so, it aims to achieve reliable self-monitoring of the individuals for COVID-19.

SYMON consists of 5 subsystems that are Cough Detection Subsystem, Fever Detection Subsystem, Heart-Rate Measurement Subsystem, Communication Subsystem and Risk Estimation Subsystem and also a mobile application, SYMON App. The block diagram of the overall system is presented in Figure 3.3. All these subsystems run on the microcontroller except from the SYMON App which runs on a mobile device. The utilized microcontroller is an ESP-WROOM32 on a DOIT-DEVKIT-V1 board. The microcontoller is chosen due to its compatibility with many different data types and protocols such as SPI, I^2S and I^2C. Futhermore, it has an included BLE antenna which eliminates the need for purchasing an external Bluetooth module. The microcontroller has a lower cost when compared with most of its peers, though it has more comprehensive processing units and flash memory capabilities, see Table 5.1. Additionally, the connections of the components with the microcontroller and its implementation is given in Figure 3.1 and Figure 3.2. The microcontroller is decided to be powered with a power bank due to its ease of use and high capacity with 4400mAh.

The sensors of the subsystems which are microphone, temperature sensor

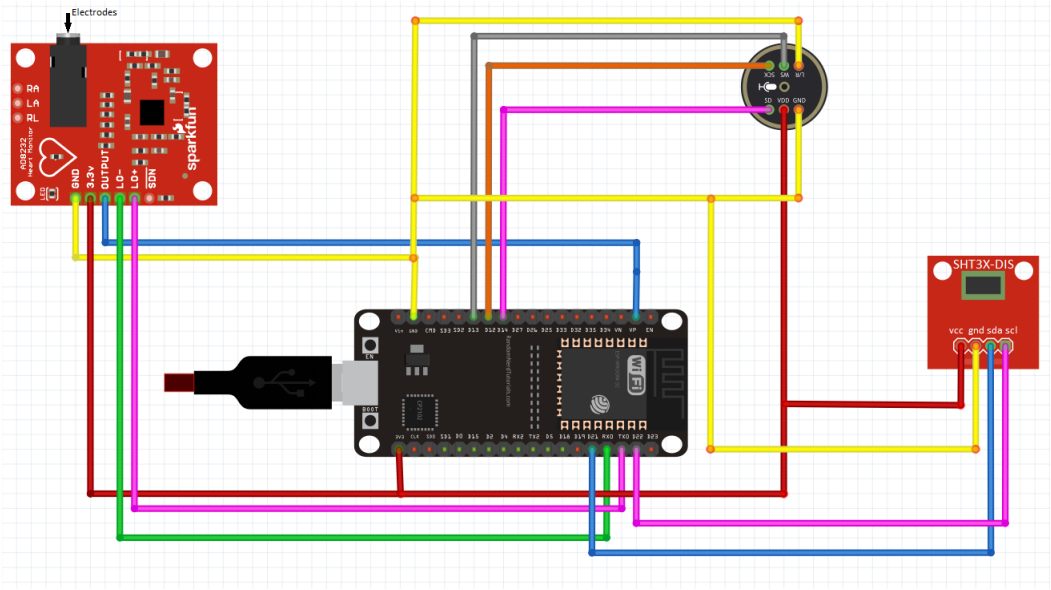


Figure 3.1: The schematic of the connections of components with the microcontroller.

Figure 3.2: Implemented connections of the components with the microcontroller.

and ECG sensor, connects to the microcontroller. The protocols that they use for data exchange with the microcontroller is given in the Table 3.1.

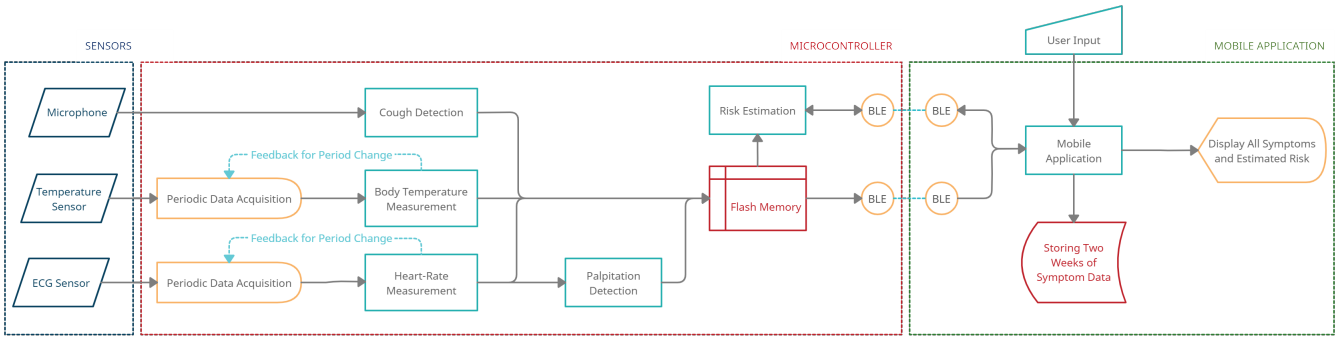


Figure 3.3: The block diagram of the overall system.

Table 3.1: The Table for Sensors of Subsystems, Their Communication Protocols and Output Data Types

Sub-system	**Sensor Name**	**Communication Protocol**	**Output Data Type**
Cough Detection	Microphone- INMP441	I^2S	Digital
Fever Detection	Temperature and Humidity Sensor- SHT31	I^2C	Digital
Heart Rate Measurement	ECG Sensor - AD8232	N/A	Analog

3.1.2 Modifications Through Design Process

A power bank is selected as the power supply of the chest strap. The previously selected power bank which has a capacity of 3350 mAh is changed with a 4400 mAh of another brand. The reason for this modification was the auto-off feature of the previous power bank. Auto-off feature is the mechanism that shuts off the power bank when the current drawn by it is lower than some threshold for some time interval. Since the maximum current drawn by overall system was low in general, the previously selected power bank turned off before the system drawn the required current to keep power bank on (refer to Section 3.1.4 for the detailed power analysis). Therefore, another power bank which does not have an auto-off feature and have a better capacity with 4400 mAh is selected as the power source of the system. The power bank also provides 5 Volts DC and may supply up to 1 Amperes. With this power bank, SYMON is estimated to have a charge cycle of 55.34 hours (refer to Section 3.1.4 for the detailed power analysis). The power bank has a USB type A output port to supply power and for recharging, it has a USB micro B port which receives 5 Volts DC as well.

Another major modification made on the overall system was the removal of the Respiration Rate Measurement Subsystem. This system is removed since it required high computational power and tedious work for calibration

each time the user wears SYMON. There was not any SPI libraries of the selected accelerometer and the other available data protocol of the accelerometer, I^2C, was too slow for motion detection. The fact that this subsystem requires two accelerometers for respiratory motion detection and a continuous computation of the data coming from the both accelerometers almost at the same time, has made it incompatible with the continuous computation of the microphone data. Even though one of the cores of the ESP32 can be utilized for the data acquisition of one of the accelerometers, the need for precise synchronization of the two accelerometer data put a high burden on the restricted computational resources. The cough detection subsystem has preferred over this subsystem for the partition of the computational power. Hence, the Respiration Rate Measurement subsystem was eliminated.

Due to the project specification obliged to HAI-CO. the Risk Estimation subsystem is separated from the Mobile Application subsystem since it is needed to be implemented on the microcontroller. It has been implemented on ESP32 successfully.

Table 3.2: The Table of Weights of the Components

	ESP32-WROOM 32	INMP441	SHT31	AD8232	Powerbank	TOTAL
Weight (g)	10	5	4	20	121	160

3.1.3 Implementation

The flow chart of the overall system is illustrated in Figure 3.4. The operation flow is as follows,

- ***Acquiring Sensory Data:*** The data is obtained from the relevant sensors in order to be conveyed to the preprocessing.
- ***Filtering and Applying Threshold:*** After the data is obtained, it is processed and information is extracted by filtering and/or applying thresholds.
- ***Storing the Detectable Symptom Data:*** Detectable symptoms are stored in the flash memory of the ESP32 in a circular memory. The circular memory consists of 24 binary files (has extension .bin) for each symptom. The symptom data for an hour is written in the corresponding binary files as binary data. In order to indicate the hour that the data is

acquired, the binary file's name is coded with a letter. For instance, the letter A in the name of a binary file indicates that the data is acquired in the first hour, B means that the data obtained at second hour and so on. The clock mechanism implemented inside the ESP32 is explained as a remark at the end of this chapter. After 24 hours, the old data is overwritten by the new data in a circular manner.

- ***Data Transmission and User Input:*** The data transmission from the SYMON to the mobile application is achieved via Bluetooth Low Energy (BLE) and the user can input undetectable symptoms via mobile application interface. The cough calibration process, changing periodicity of the measurements, requesting new measurements and some other features are available through BLE communication which are explained in detail in Section 3.5.

- ***Storage of the Symptom Data:*** In order for users to keep track of their symptoms, 2 weeks of merged undetectable and detectable symptoms data will be available in the mobile application. The 24 hours of detectable symptom data will be kept in the flash memory of the ESP32 as well.

- ***Risk Estimation:*** At the end of each 16 hour periods (or by the request of the user), the risk estimation module will acquire the detectable symptom data from ESP32 flash and undetectable symptom data from the mobile application for that period. The detectable symptom data will be preprocessed, which explained in Section 3.7.2 and will be fed with the undetectable symptom data to the classifier algorithm.

- ***Display of COVID-19 Information:*** The user will have an access to,
  - the COVID-19 label (1 or 0, indicating the presence of the COVID-19)
  - the confidence score (confidence of the algorithm about the label)
  - relevant warning or precaution message according to the label and the confidence score
  - all undetectable and detectable symptom data of last 2 weeks

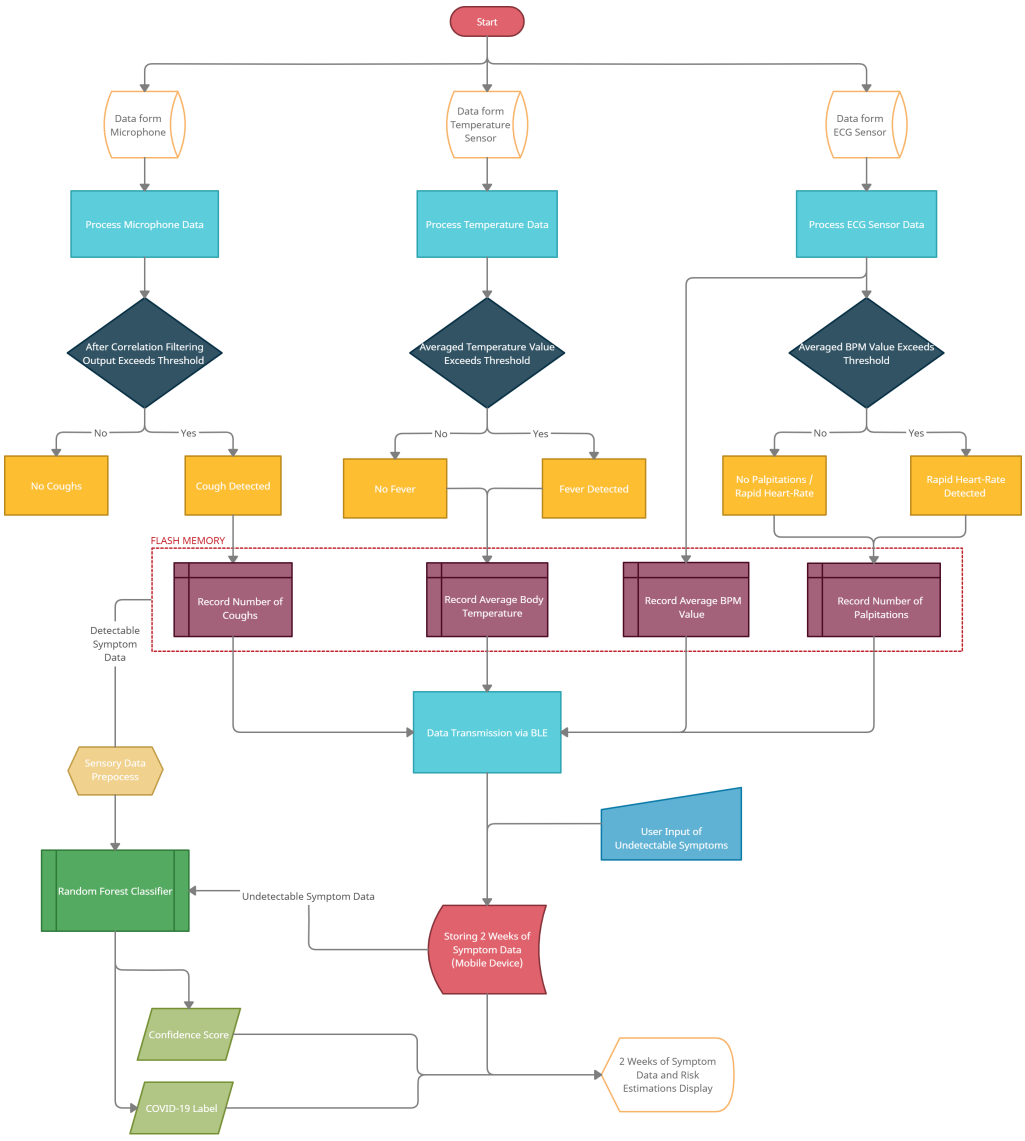


Figure 3.4: The functional flow diagram of the overall system.

Remark: It is important for the wearable device to distinguish the data according to the time it is measured. For ESP32 to acquire the real-world time, it needs to be connected to the internet. Since this may not be possible for users of SYMON, an artificial hourly clock is created by using the timers inside the ESP32. When ESP32 is powered on, the variable "h" is initialized as 0. After completion of each hour, h is incremented by one. This "hour" information is used for writing the obtained measurement data to the correct files in the 24-hour circular memory as well as scheduling the tasks of the measurements.

3.1.4 Power Management

The project is planned to be powered with a power bank of 4400 mAh or 16.3 Wh which provides constant 5V and up to 1A. A USB cable is plugged to the USB post of the personal computer and the other end of the USB cable is cut off. The ground cable of this cut USB cable is connected to the ground pin of the ESP32 and 5V cable of the USB cable is connected to the Vin pin of the ESP32. But before it is connected to Vin, a digital multimeter [4] is connected in series in order to measure current.

An issue of concern is the varying drawn current for the execution of functions of different subsystems. For instance, when a burst of sound occurs, the drawn current increases due to the execution of cough detection algorithm, however, while temperature sensor is working, the drawn current is not as high.

In this approach for calculating the power consumption, the duration of the states of the overall system are considered in an hour and the average drawn current is estimated accordingly. After an average current value is obtained for 16 hours, it is multiplied with the supplied voltage value, 5V to find the average power supplied by the power bank.

In Table 3.3, the measured drawn current by the ESP32 is presented with the respective states, The states are the situations of the overall system when different subsystems are working. The descriptions of the states are also provided in the table. In these states, the BLE communication and writing to the flash operations are included.

Table 3.3: Supplied power and drawn current values for each state.

State	Description	Drawn Current	Supplied Power
Idle	Microphone listens for coughs but there is not any bursting sound	∼71.4mA - ∼75.5mA	357 mW - 377.5 mW
Cough	Presence of a bursting sound	∼95mA - ∼115mA	475mW - 575mW
Temperature	Taking temperature measurement, writing this value to flash and transfer this data through BLE	∼77mA	385mW
Heart-Rate	Taking heart-rate measurement, writing this value to flash and transfer this data through BLE	∼77mA	385mW
Risk Estimation	Risk estimation subsystem is calculating risk and sending through BLE	∼107mA	535mW
	TOTAL	**427mA - 451.5mA**	**2137mW - 2257.5mW**

It should also be considered that ESP32 will not be in the states mentioned in the table at all times. The ECG sensor will be working every 8 hours by default, when there is not an external request sent by the user. The

temperature sensor will be measuring the body temperature with periods of 3 hours, by default. Only the microphone will be working continuously. As it can be seen from the table, when microphone is working in the idle situation, the drawn current is not high but when there is a burst of sound such as coughs or claps, the drawn current can increase up to 115 mA. However, it would not be wrong to assume that high energy sounds will occur only momentarily. Hence, the power consumption of the device will be equal to the worst case power consumption value transiently.

Now that both the current fluctuations and the duration of the high and low currents are considered, the power consumption can be estimated more precisely. A use case is created where the daily usage duration is assumed to be 16 hours and the user does not wear the product for sleep time (8 hours). In this use case, it was planned that there will be two heart-rate measurements at the beginning of the day and at the end of the day which will take 6 seconds. Since the temperature data will be taken in 3 hours intervals, there will be 5 measurements in the day time and these measurements will be taken in 3 seconds. The microphone will be on during 16 hours; however, the high energy sounds are assumed to occur only 10% of this 16 hours duration which corresponds to 96 minutes. The wearable device will calculate the risk according to the symptoms and send the estimated risk data to the mobile application via BLE with 16 hours periods. The risk estimation operation lasts approximately 3 seconds. By combining all of these with the following calculations, the total consumption will be found as follows,

Idle	**1087.2 mAh** = 14.4 hours × 75.5 mA
Cough	**184 mAh** = 1.6 hours × 115 mA
Temperature	**0.321 mAh** = 0.0008 hours × 5 times × 77 mA
Heart-rate	**0.257 mAh** = 0.0016 hours × 2 times × 77 mA
Risk Estimation	**0.0892 mAh** = 0.0008 hours × 1 time × 107mA
TOTAL	**1271.87mAh = ∼1.3Ah**

The total value found is, 1271.87mAh for 16-hours. In order to find an average, we need to divide it to 16. Which will be approximately equal to 79.5 mAh. Therefore, the average usage of overall system in an hour will be 79.5 mAh. Since the selected power bank has a capacity of 4400 mAh, with the mentioned use case, HAI-CO. offers approximately 55.34 hours or approximately 2.3 days of charge cycles for SYMON. Therefore the set

requirements in the Project Proposal Report are met [3].

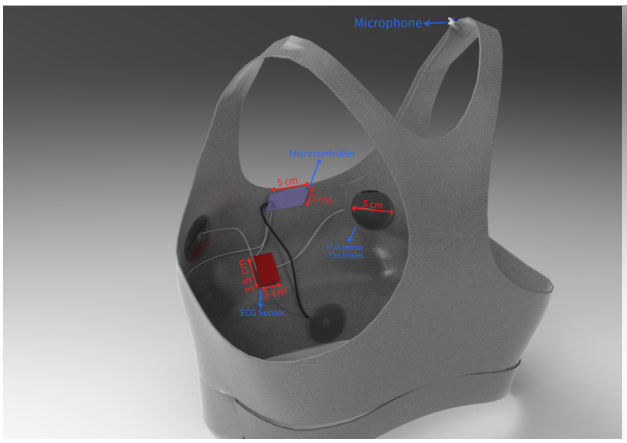
3.1.5 Compatibility with System Requirements

The overall system is mounted on a chest strap which is designed to be lightweight, 160 grams, and ergonomic. The weight of the components is aimed to be distributed evenly on the chest strap to provide a comfortable use and adaptation to any daily life activities. The weight constitution of the overall system is given in Table 3.2. The overall design, places of the sensors and the power bank are shown in the 3D drawing of the chest strap in Figure 3.5. The 3D drawings aim to show the mounted circuitry of the SYMON and general plan inside of the chest strap. For this reason, the components are depicted without a cover. In order to see the final product please refer to Figure 1.1.

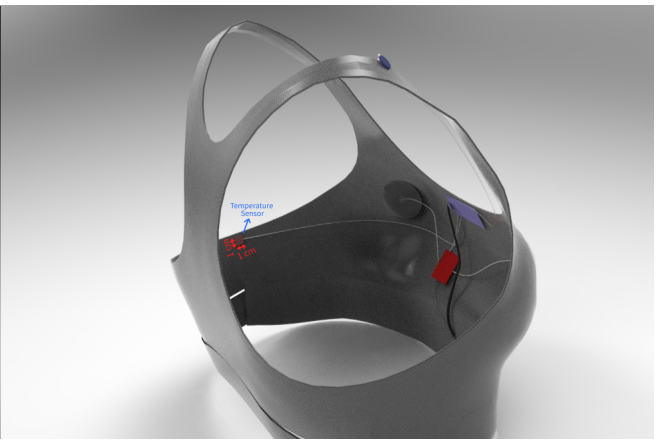
When the requirements that are set in the Section 3.1 are considered, SYMON has achieved,

- a charge cycle of approximately 2.3 days which is much longer than the set requirement which was 16 hours.
- a reasonable cost of $47, which is way below the determined requirement of being less than $80.
- being lightweight with 160 grams, which is below the set requirement of being less than 200 grams.
- gathering all the relevant symptoms through its sensors as well as from its mobile application.
- being user friendly by having an easy to understand mobile application interface.
- being ergonomic, washable and separable.

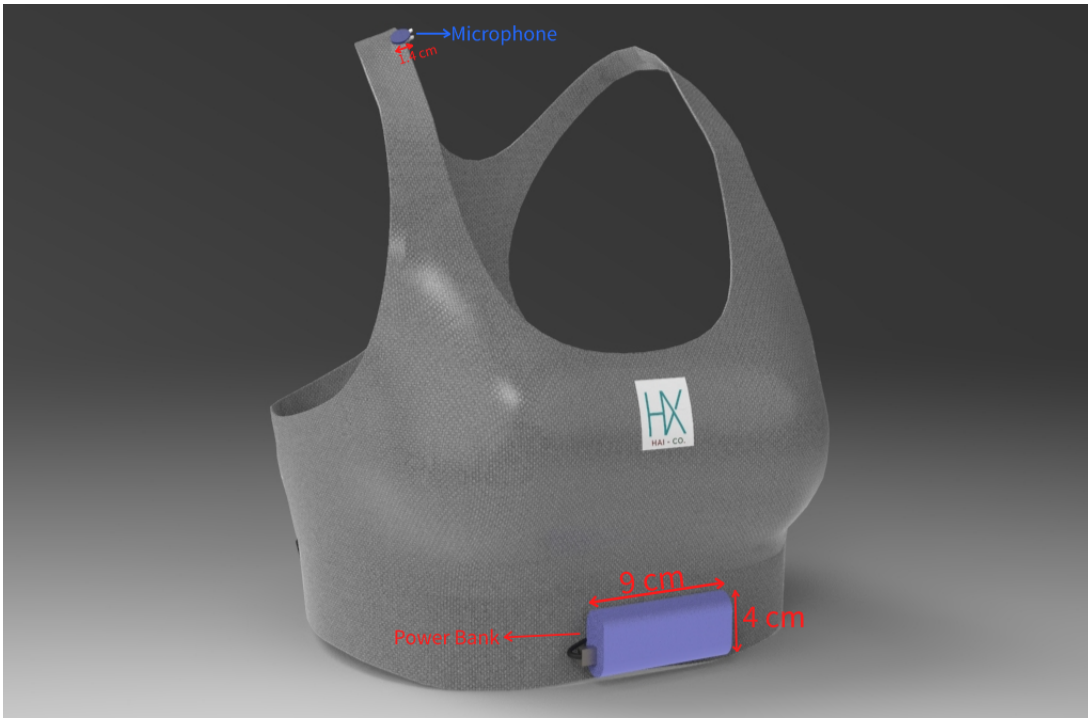
SYMON has achieved the set requirements mentioned in Chapter 2 and is ready to be on the market.



(a) The left side-view of SYMON.



(b) The right side-view of SYMON.



(c) The front view of SYMON.

Figure 3.5: 3 Dimensional Drawings of the SYMON.

3.2 Cough Detection Subsystem

There are variety of methods for cough detection. Methods differ from one another in terms of their applicability and performance for the given operating conditions. For example, respiratory monitoring via chest movement is considered as the gold standard; however, it yields poor performance when the user is not stationary [5]. Detection of cough via sound, where

the results are not deteriorated from the user's mobility, is considered as the second best method for a reliable performance and thus [5] selected to be implemented on SYMON.

3.2.1 Selection of Hardware

To meet with the requirements mentioned in the Section 2.2.1, audio signals must be obtained with satisfactory quality. As a studio standard for audio recordings; with adjustable sampling ratio (and an anti-aliasing filter), high signal to noise ratio (60 dBA), low quantization error (32 bits) and low power consumption (250 µA & 3.3 - 5 V), I2S (maximum of 44100 Hz sampling rate) MEMS microphone, Invensense INMP441, is used [6]. I2S protocol is supported by the selected microcontroller, ESP-WROOM32 [7]. See Table 5.2 for comparison of Invensense INMP441 I2S MEMS microphone with other possible candidates on the market.

3.2.2 Design Process, Tests & Results

During the development of the Cough Detection Subsystem, as obstacles were encountered, operating principles of the subsystem were changed, additional features were added or some features were abandoned. Detailed overview of the design process is given as follows;

- **1 - Initial Subsystem**
  - **1.1 - Priors:** On the initial expiration phase and the voiced phases of a cough, high frequency power peaks, see Fig. 3.6 [8].
  - **1.2 - Implementation:** High frequency power was calculated as the summation of square magnitude of frequency components at frequency bins corresponding to angular frequencies above $\frac{\pi}{2}$. Accordingly coughs were counted when the high frequency power of the signal exceeded its threshold. Since high frequency power of an input cough potentially exceeds the set threshold on the expiration and voiced phases twice, counting the number of coughs had to be regulated. This regulation is done by setting a time margin of 0.25 seconds (expected time between the phases of a cough) between any sequential increase of cough counter [8]. High frequency power outputs and regulation with a time margin is illustrated in Fig. 3.7. For further infor-

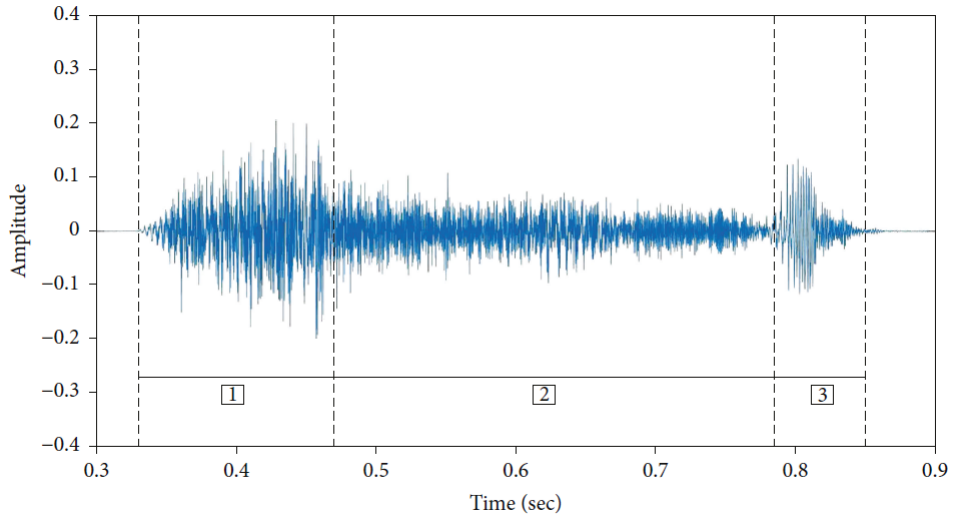


Figure 3.6: A cough sample and its phases: (1) an explosive expiration due to glottis suddenly opening, (2) the intermediate phase with attenuation of cough sounds and (3) the voiced phase due to closing of the vocal cord.

mation on the Initial Cough Detection Subsystem see Conceptual Design Report, Section 3.2.1 [9].

- **1.3 - Tests, Results and Remarks:** The system was prone to high frequency and high power sound inputs. Apart from its accuracy (MATLAB w/ Environmental Noise: 0.69, Hardware Implementation: 0.59, see Table 3.4) being lower than the set performance, 0.7; it had low specificity (MATLAB w/ Environmental Noise 0.45), i.e., unsatisfactory rejection and misclassification of the non-cough input. Furthermore, the ratio of misclassified non-cough sounds (FP) to the total number of trials (TP+TN+FP+FN) was 0.25 (MATLAB w/ Environmental Noise), see Table 3.4. Which was highly undesirable since the false positives can easily dominate the true positives within the operation duration of the subsystem (see Section 2.2.1). Average length of a cough is experimentally found to be around 0.2 to 0.4 seconds, which is consistent with [8]. Hence, number of detection trials within active 16 hours of operation (cough duration is taken 0.2 seconds for the upper bound) is 288 000. That would make 72 000 false positives as the worst case scenario.

- **1.4 - Conclusions:** The performance of the subsystem had to be increased, especially in terms of rejection rate of non-cough sounds.

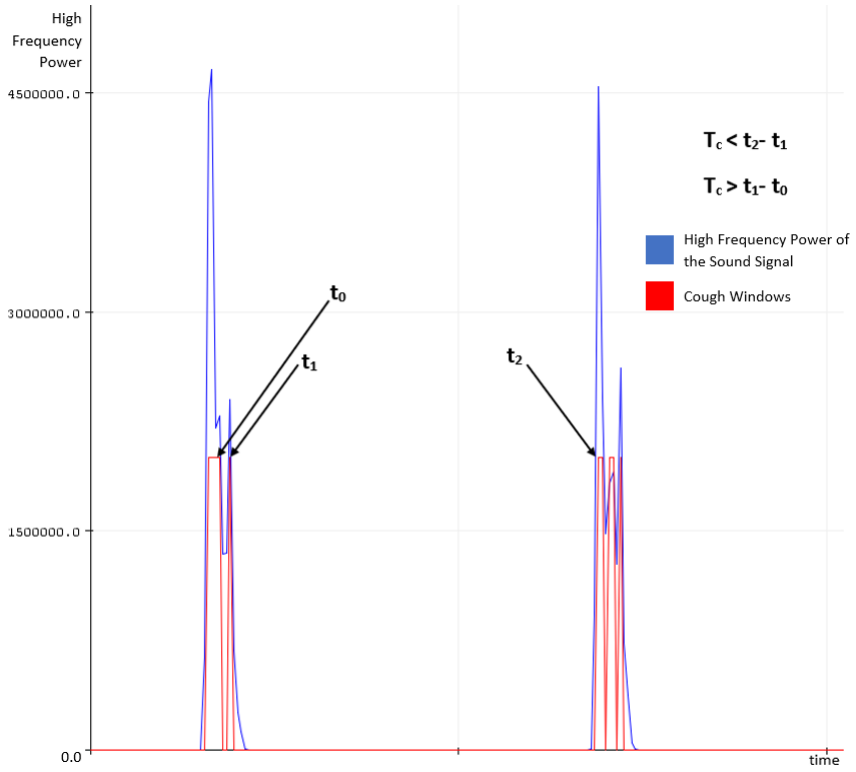


Figure 3.7: High frequency power outputs and time margin regulation (T_c: time margin), cough counter is increased only for t_0 & t_2

- **2 - Neural Network Branching**

  This solution path was chosen to find a viable solution for the problems of the "Initial Subsystem". Its development coincides with the development of the "Improved Subsystem".

  - **2.1 - Priors:** Majority of cough detection and classification methods are being performed via the usage of machine learning algorithms on a static-platforms with high computational power, while the sound data are acquired with wearable devices [8], [10], [5]. Implemented CNN structures on cough detection are fed with Cepstrum or Spectrogram data of the input sounds. Meanwhile, DNN and NN structures use large set of features.

  - **2.2 - Implementation Remarks:** Inference of cough via spectral images increases the dimensionality of the problem significantly. Furthermore, in order to successfully implement the neural network structures, both the training data and the sensory data had to be normalized. With a mismatch of the contents of the training dataset with respect to the real time inputs, the

Table 3.4: Former Cough Detection Subsystem Test Results and Performances

TEST	Performance Measures					True Positives (TP)	False Negatives (FN)	False Positives (FP)	True Negatives (TN)
	Accuracy	Precision	Recall	F1 Score	Specificity				
MATLAB (Skewed Data)	0.84	0.87	0.96	0.91	N.A.	215	8	32	N.A.
MATLAB (w/ Environmental Noise)	0.69	0.65	0.90	0.75	0.45	108	12	56	47
Hardware Implementation	0.59	0.80	0.69	0.74	N.A.	41	18	10	N.A.

outputs can deviate from training performance significantly. Additionally, creating a dataset with the usage of microcontroller limits the flexibility of the development. Also, deviations from an initial training model in terms of the form of the input data require either, again, proper normalization or re-creation of a new dataset.

◦ **2.3 - Conclusions:** With much limited storage space and computational power than personal computers, neural network implementations do not match with the resources and evidently would put a strain on the current hardware.

• **3 - Improved Subsystem**

With the the foreseen obstacles in "Neural Network Branching", this solution path was chosen to find a viable solution for the problems of the "Initial Subsystem". Its development coincides with the trials of the "Neural Network Branching".

◦ **3.1 - Priors:** Considering the dimensionality of the objective problem and the fundamental idea behind the usage of CNNs, it was decided to use correlation filtering to detect coughs real time [11].

◦ **3.2 - Added Features:** Considering the set objectives in Section 2.2.1, the following set of features are added to the subsystem;

+ A trigger mechanism is implemented in order to decrease misclassifications of non-cough instances. Since the detection is now conditioned on a trigger, mean of performed computations within a cycle had also decreased; leading to less energy consumption.

+ To reject high power inputs, an upper threshold is set; coinciding with the intermediate attenuation phase and voiced phase of cough sounds, see Fig. 3.6 [8].

+ Correlation Filtering is used in order to increase the overall performance (accuracy, recall and precision) of the subsystem.

+ Obtained features from correlation output are also used to differentiate and reject common sounds previously misclassified as coughs, also increasing the specificity of the system.

◦ **3.3 - Implementation:** Block diagram of the subsystem depicting its general workflow is shown in Fig. 3.8.

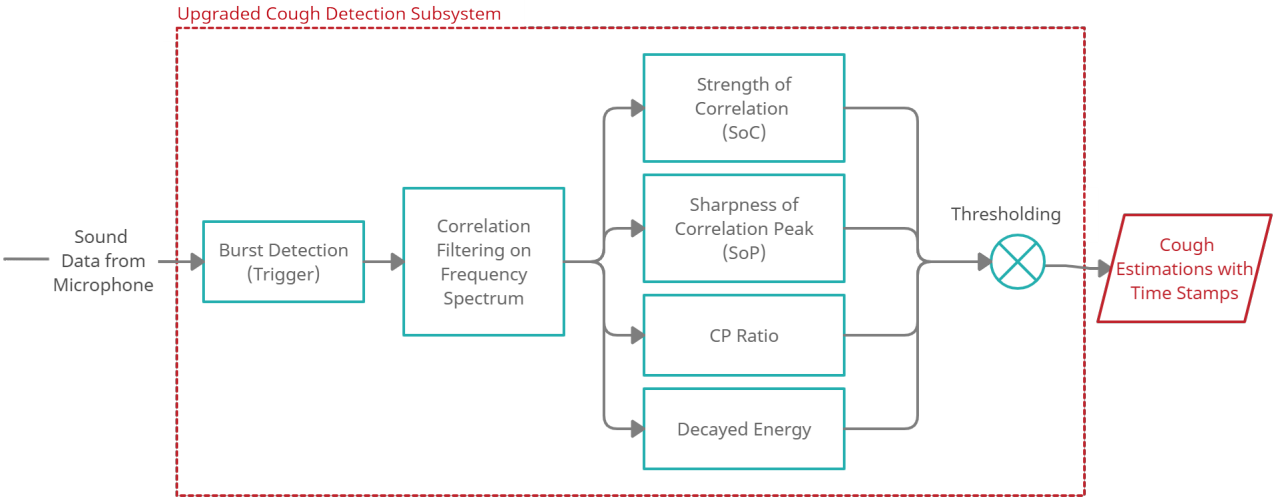


Figure 3.8: Upgraded Cough Detection Subsystem Block Diagram.

Following items describe the design considerations and working principles of each sub-block on the diagram in Fig. 3.8.

→ **Acquisition of Sound Data:**

In order to reduce constant heavy computation, initial phase of a cough is set as the trigger for filtering. Therefore, proper detection of explosive initial phase is of high importance. Cough related frequency information reaches up to 4 kHz [12], while the frequency content of the initial explosive expiration phase reaches up to 13 kHz frequency range, see Fig. 3.9. Within the upper limit of frequencies (13 kHz) on explosive initial phase, other

common sounds' dominant frequencies are present, see Fig. 3.9, and can lead to incorrect detection of the explosive phase. To restrict this interference on the detection of bursts, sampling rate is set as 12 kHz allowing high frequency content up to 6 kHz and rejecting higher frequencies with low pass filtering pre-sampling (there is not any aliasing) by INMP441 I2S microphone module.

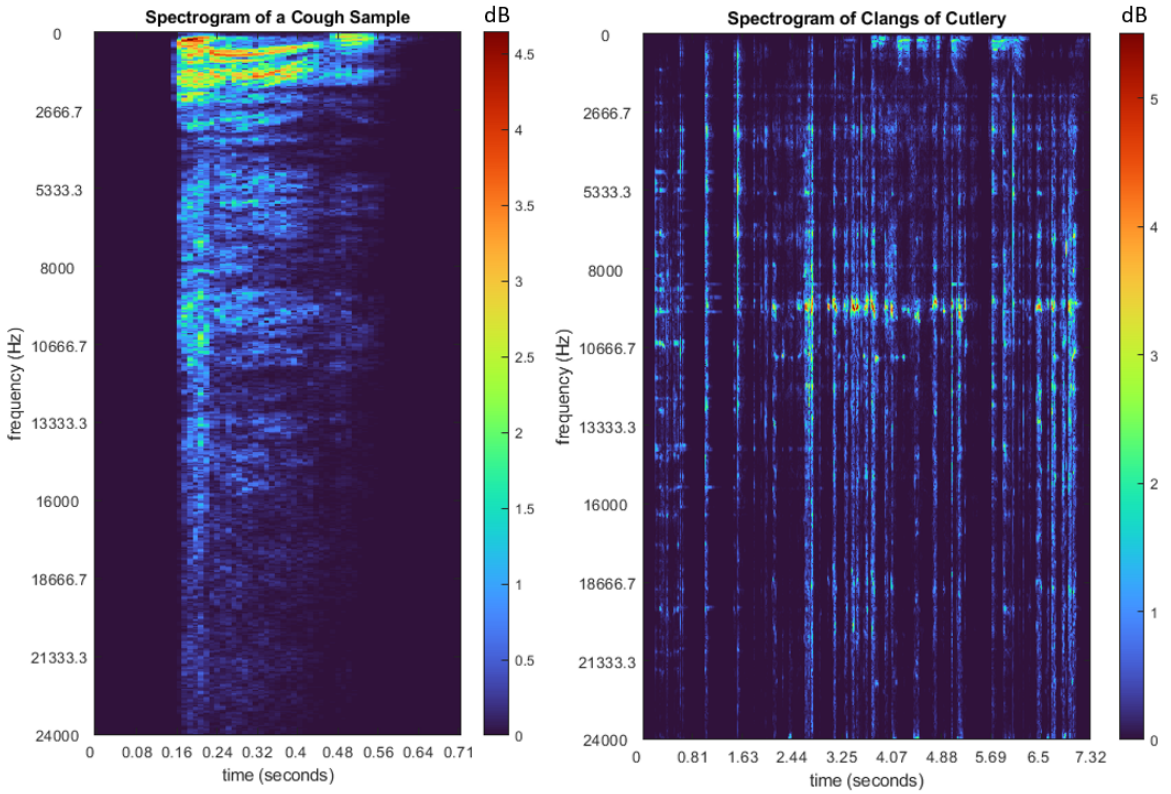


Figure 3.9: Cough and Cutlery Clang Spectrograms.

On experimentation with MATLAB, FFT computation by using 0.02 seconds length Hann windows with 50% overlap yielded clear Fourier magnitude spectrum. The closest power of 2 to the specified window length, 256, is selected as the FFT sample length for STFT implementation on the microcontroller. With a DMA buffer length of 512 samples and 32 bits for each sound sample, FFT values are calculated real time with 50% overlapping Hann windows.

→ **Correlation Filtering:**

To prepare the correlation filter that will be used on Fourier magnitude spectrum, user is asked to cough several times in a silent environment. From this calibration recording, coughs are extracted via detection of high

energy frames, see Fig. 3.10. Then average duration of coughs is calculated. Each cough's spectrogram image is bilinear-interpolated to match the average cough duration. Same size interpolated spectrogram images are averaged. Hence, average cough magnitude spectrogram is obtained. Logarithmic magnitude of the average cough spectrogram is linearly quantized with 8 bits and sent to the microcontroller for it to be used as the correlation filter, see Figure 3.10.

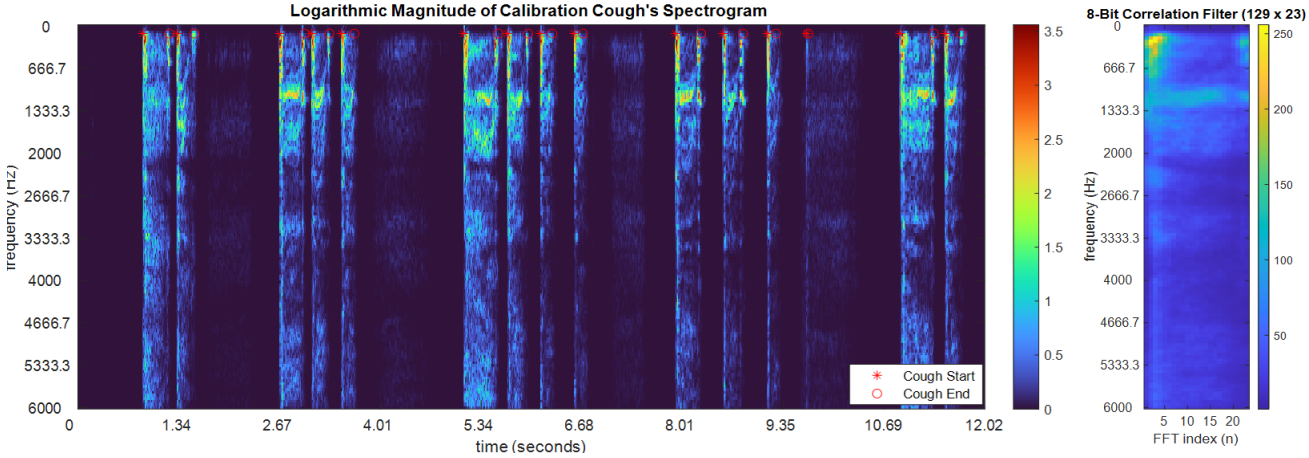


Figure 3.10: Extraction of Coughs from Calibration Recording and Correlation Filter.

In order to use the outputs of a correlation filter for estimation, the filter needs to be slid on the input spectrograms. When the filter overlaps with the cough magnitude spectrum, the strength of correlation peaks. From the outputs, the strength of correlation and sharpness of the correlation peak is primarily used for detection of coughs. In Fig. 3.11, MATLAB implementation of correlation outputs and detected cough locations are shown.

The built-in FFT library of Arduino uses floating points to calculate the spectrum. Storing two-dimensional (129 x Average Cough FFT Length) float array consumes significant space on memory; therefore, the sliding filter implementation on the microcontroller is different than the MATLAB implementation. A circular summation-buffer of length equal to Average Cough FFT Length + 6 + 3 is used to calculate correlation in a sequential manner. Buffer length is chosen accordingly to meet the average length of a cough and necessary calculations.

The procedure of correlation calculation is as follows. Once the filtering mechanism is triggered with a detection of burst, along with the previous

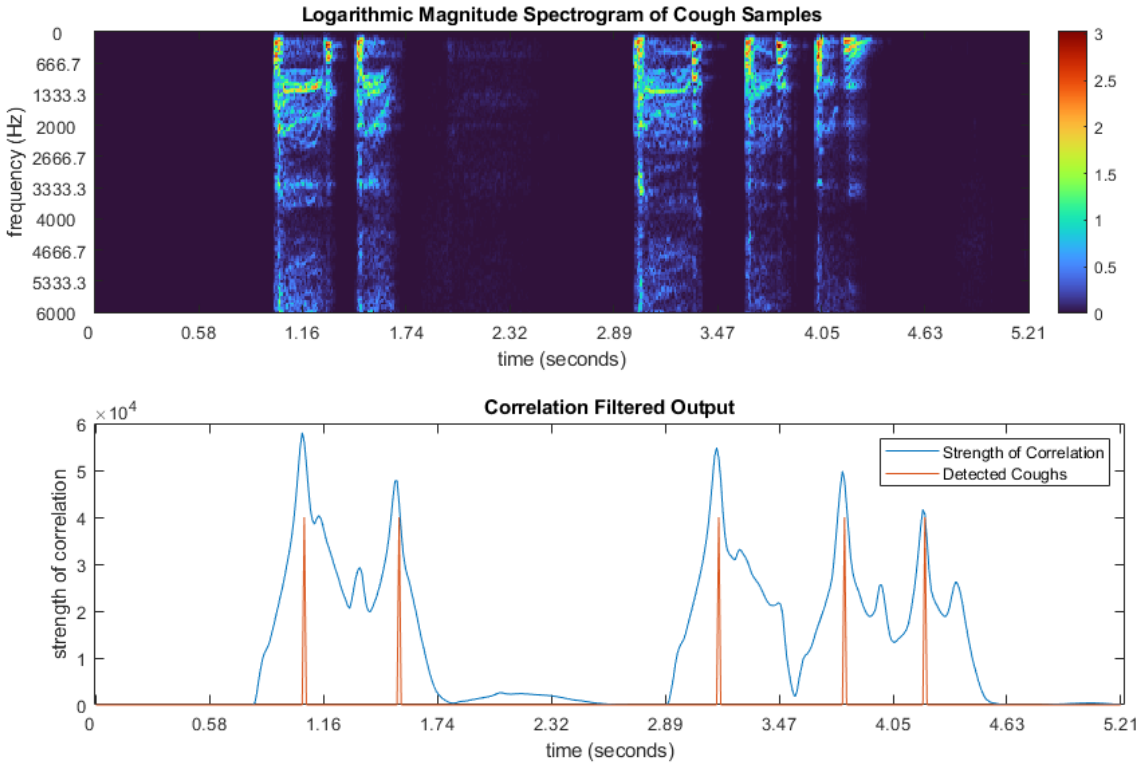


Figure 3.11: MATLAB implementation of correlation outputs and detected cough locations.

3 FFT results before the trigger, FFT arrays (129x1 vectors) are started to be sequentially processed. The sequential calculation of correlation lasts for Average Cough FFT Length + 6 FFT iterations in order to detect peaks properly. At each iteration, dot product of the FFT vector with the correlation filter columns are taken and added to the appropriate bins of circular summation buffer. At each iteration, the buffer's indexes are circularly updated. Last 3 bins of the buffer are the finalized correlation outputs and are used for detection of peaks and obtaining features.

→ **Features, Thresholds & Outputs:**

Correlation outputs are stored at the last 3 bins of the circular summation buffer. From these 3 bins, Strength of Correlation (SoC), Sharpness of Correlation Peaks (SoP) and SoC to SoP ratio (CP ratio) are obtained.

Strength of Correlation is the direct output of the correlation filter, it is an indicator of matching patterns. However, for non-cough inputs, the correlation strength can still reach considerably high values. For example, speech and cough are both generated by human vocal cords and show great

resemblance in their patterns, also forced inputs filling the magnitude spectrum inevitably yield outputs of high correlation (see Fig. 3.12). On the contrary, when the correlation filter is slid on the input, these non-cough patterns either do not form sharp correlation peaks or form flat correlation outputs. Hence, Sharpness of Correlation peaks are also an indicator for proper detection. It is calculated as multiplication of first derivatives on the adjacent points of the correlation peak and is negative for any peak. Its magnitude is a degree of peak sharpness. Furthermore, CP ratio is shown to be an indicator on specificity for patterns. Short bursts of sound form sharp peaks but yield low strength on correlation; meanwhile, sounds with high energy yield strong correlation but flat peaks (see Fig. 3.12). CP ratio is mainly used to reject bursts with high energy such as clapping and clanging noises.

At last, to reduce the number of misclassified outputs due to forced continuous inputs with high energy, an upper energy threshold and a time threshold are set.

An example cough with its detected feature values is shown in Fig. 3.12f.

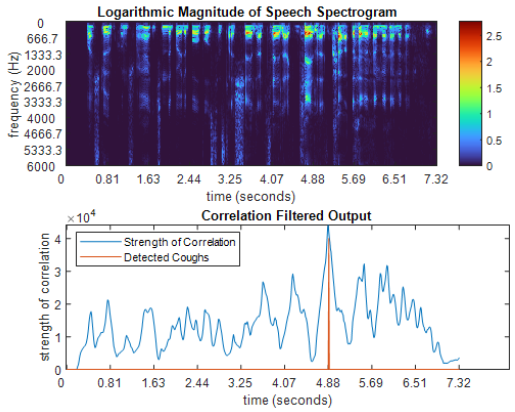
Thresholds for features are found by trial and error.

Cough detection is outputted if all the features are at proper limits, below or over their respective thresholds.

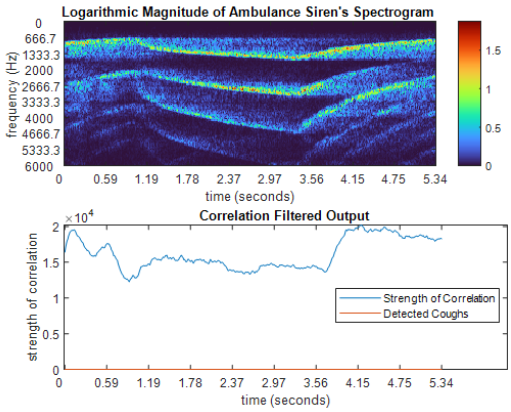
- **3.5 - Tests, Results and Remarks:** The tests are conducted with different classes of sounds (see Appendix B3) that address possible error sources. Also environments that can be encountered in daily life are simulated with appropriate dB levels.

The recordings are played from a speaker. For each class, dB values are adjusted to their respective dB readings in real encounters [13]. For this purpose, Sound Meter mobile application [14] which allows calibration of dB readings pre-recording is used. Calibration is done via matching dB readings to that of speech when speaking. The application shows the mean of dB readings for the specified time period of measurement, for each recording this mean is logged.

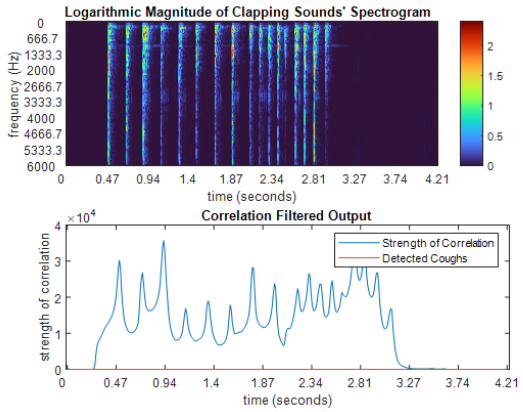
Cough subsystem tests are done by mounting the subsystem on the user as shown in Fig. 3.13a.



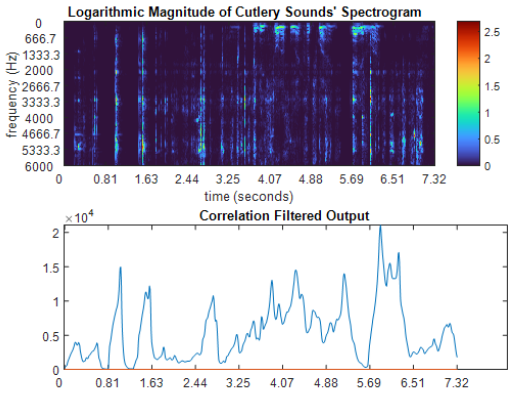
(a) **Speech**.



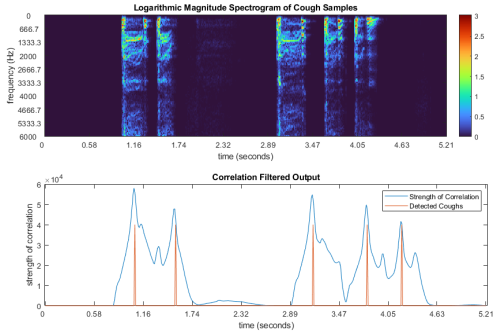
(b) **Ambulance Siren.**



(c) **Clapping.**



(d) **Cutlery.**

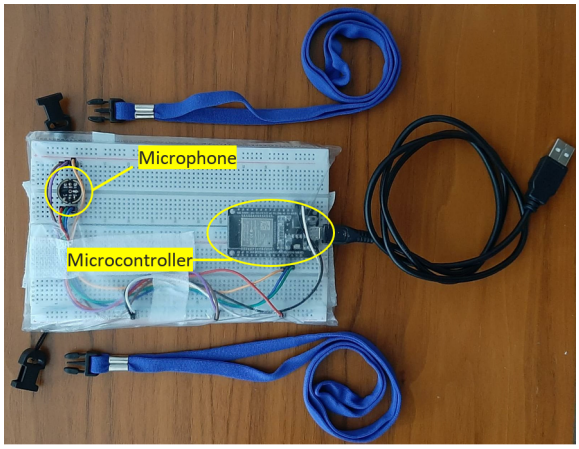


(e) **Coughs.**

```
-> NOF COUGHS/TOTAL FFT          : 1 / 24.00
-> Burst Energy (Phasel, TRIGGER) : 27.43
-> Strength of Correlation       : 3125.69
-> Sharpness of Correlation Peak : 603.73
-> CP Ratio                      : 5.18
-> Energy (Phase2 & 3)           : 393.59
```

(f) **Cough.**

Figure 3.12: a,b,c,d & e :MATLAB implementations; f: Microcontroller implementation of correlation outputs, features and detected coughs for different sounds.



(b) **Cough Submodule Test Equipment.**

(a) **Mounting of the microphone**.

Figure 3.13: Main test equipment and its mounting.

Table 3.5: Upgraded Cough Detection Subsystem Test Results

Sound Content	**File ID**	**Duration (sec)**	**Average dB**	**Max Trials**	**Applied Trials**	**# Cough Groups**	**# TP Cough Groups**	**# FN Cough Groups**	**# FP Cough Groups**	**# Coughs**	**# Coughs Outputted**
	-	-	-	-	-	-	-	-	-	-	-
Quite Residential Area	WC1	85	50	346	344	11	10	1	0	25	18
	NWC1(7/17)	27	50.3	110	0	0	0	0	0	0	0
Restaurant, Speech, Cutlery	O2	392	63.5	1598	0	0	0	0	0	0	0
	WC2	148	63.2	603	248	8	8	0	0	19	14
	NWC2(5/10)	27	72	110	0	0	0	0	0	0	0
Restaurant, Speech, Cutlery, Clangs	O3	190	78.2	774	0	0	0	0	0	0	0
	WC3	163	77.4	664	591	18	12	6	0	42	18
	NWC3(5/8)	23	77.2	94	0	0	0	0	0	0	0
Heavy Traffic, Slow Breaking, Different Vehicle Types	O4	227	75	925	0	0	0	0	0	0	0
	WC4	178	78.3	726	447	14	11	3	0	33	18
	NWC4(7/15)	31	72.3	126	0	0	0	0	0	0	0
Heavy Traffic, Different Vehicle Types and Horns	O5	42	76.8	171	0	0	0	0	0	0	0
	WC5	42	74	171	156	6	4	2	0	12	7
	NWC5(6/19)	36	72.6	147	0	0	0	0	0	0	0
Ambulance Siren	O6	20	82.4	82	0	0	0	0	0	0	0
	WC6	41	80.1	167	226	6	4	2	0	13	5
	NWC6(7/17)	24	78.6	98	0	0	0	0	0	0	0
Office Assembly, Metallic Clangs and dismantling, Dropped Items	O7	216	56.2	880	0	0	0	0	0	0	0
	WC7	216	54.4	880	670	20	16	4	1	47	31
	NWC7(6/16)	27	63	110	0	0	0	0	0	0	0
Phone Ringing, Keyboard Sounds, Printer, Speech, Eraser Rubbing	O8	600	66.5	2446	0	0	0	0	0	0	0
	WC8	600	68.9	2446	821	25	23	2	0	65	38
	NWC8(9/21)	60	65.7	245	0	0	0	0	0	0	0
Speech, Children Crying, Children Shouting, Background Traffic	O9	600	65.8	2446	0	0	0	0	0	0	0
	WC9	594	64.2	2421	794	24	20	4	1	60	30
	NWC9(8/18)	34	69.1	139	0	0	0	0	0	0	0
	TOTAL	**4643**	**-**	**18925**	**4297**	**132**	**108**	**24**	**2**	**316**	**179**

For each class of sounds wearer and non-wearers of the SYMON are asked to cough several times along with raw, original, recordings. Outputs of the subsystem are observed and logged. Test results are shown in Table 3.5.

On Table 3.5, cough groups refer to a collection of coughs. Cough group is a consecutive sequence of coughs lasting less than 1 seconds. The groups contain 1 high energy major cough, followed by several low energy minor coughs. Cough groups are illustrated on Fig. 3.14. The objective of cough detection subsystem is to detect whether the user shows coughing symptom, hence correct detection of major coughs, equivalently cough groups, is satisfactory and the performance results are as intended.

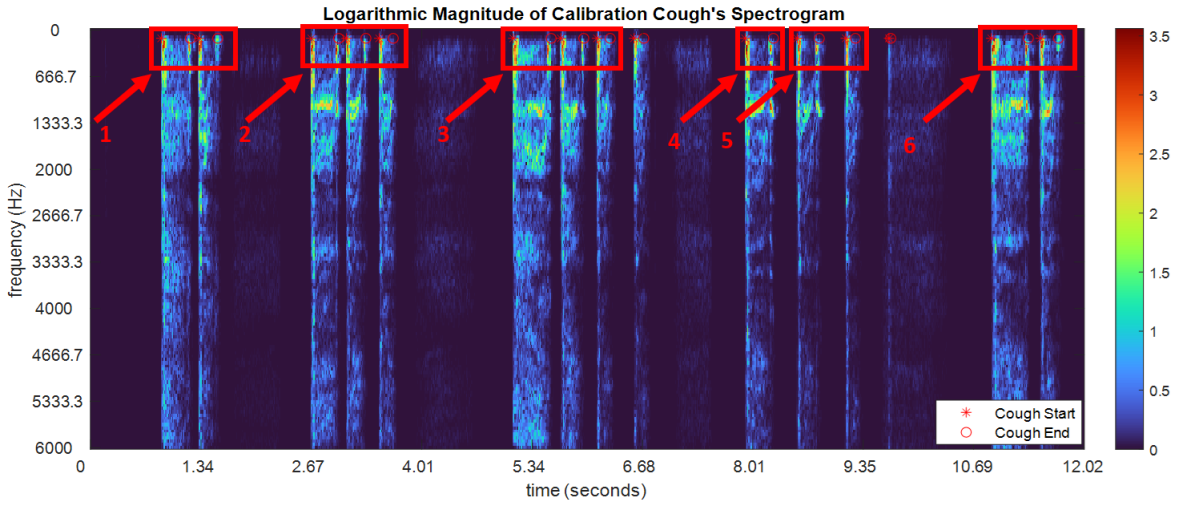


Figure 3.14: Spectrogram of 6 cough groups (windowed) and coughs (*: start, o: end).

The nomenclature of file IDs at Table 3.5 are as follows,

- O ⟨#⟩: Original recording of sound class without any coughs.
- WC⟨#⟩: Recording of sound class with the coughs of the wearer of SYMON.
- NWC ⟨#⟩(x/y): Recording of sound class with the coughs of non-wearer of SYMON (x cough groups and total of y coughs).

Because of the trigger mechanism, the sound bursts of the user are put into consideration strictly. Hence, apart from the WC⟨#⟩, the system did not give any cough outputs (rejected all non-cough sounds and the coughs of non-wearers) or did not considered to filter the inputs.

Table 3.6: Performance Results of the Upgraded Cough Detection Subsystem

Considered Trials	Accuracy	Specificity	Precision	Sensitivity-Recall	F1
Max Trials (18791 TN)	0.998626	0.999894	0.981818	0.818182	0.892562
Applied Trials (4163 TN)	0.993949	0.99952			

Table 3.7: Performance Comparison of Former and Upgraded Cough Detection Subsystem

Performance Criteria	**Cough Detection Subsystem**	
	Former	**Upgraded**
Accuracy	0. 69	0.99394
Specificity	0.45	0.99952
Precision	0.65	0.98181
Sensitivity-Recall	0.90	0.81818
F1	0.75	0.89256
(FP) / (TP+TN+FP+FN)	0.25	0.0001
FP in 16 Hours	**72 000**	**25**

Since the tests are done continuously in time and not on a data set with discrete number of sound packages, performance results are calculated based on number of trials considered or the maximum number of considerations possible within the recording duration, see Table 3.6.

There is a significant increase in the performance of the subsystem with respect to its former version. Increase in performance is shown in Table 3.7 (see Table 3.6 and Table 3.4 for individual performance results).

For all performance criteria, except sensitivity, the upgraded system is significantly better while satisfying the set performance criteria on all aspects. Most importantly (FP) / (TP+TN+FP+FN) ratio decreased significantly. Worst case scenario on the number of misclassified non-cough sounds decreased to 25 for 16 hours of operation (16 hours / (#FP per second)). Considering that the cough data will be sent to the mobile device periodically during the day, 25 misclassified non-cough sounds is negligibly small on hourly basis (1.5 FP/hour).

- **3.4 - Conclusions:** The performance of the subsystem meets the required performance criteria. Therefore, the system was decided to be implemented on the final product.

• 4 - Finalized Subsystem

- **4.1 - Priors:** "Improved Subsystem" is directly used as the "Finalized

Subsystem" without any modifications on detection of coughs and with minor modifications on extraction of correlation filter.

∘ **4.2 - Added Features:** On the "Improved Subsystem", user specific correlation filter was extracted on MATLAB and it was intended to be transferred to the mobile application. However, due to restrictions on the extent of usage of mobile devices, following modifications are made;

+ Extraction of user specific correlation filter is transferred to the microcontroller.

+ In order to handle possible deviations from the empirically obtained threshold values, threshold calibration is automated on the microcontroller.

+ Using the empirically obtained threshold values as a basis, proper calibration of thresholds and filter coefficients are enforced.

∘ **4.3 - Implementation:** Calibration schematic is shown in Fig. 3.15. Working structure of the calibration is as follows;

→ **Acquisition of Sound Data:**

When a cough calibration command is received, system starts to wait for a burst of sound. This is the same trigger mechanism implemented on improved cough detection subsystem.

After a burst is detected, the system expects to capture the cough of the user. Therefore, it is required for the user to start calibration in a silent environment.

To capture the frequency content of sounds, FFTs are calculated as described in "∘ 3.3 - Implementation → Acquisition of Sound Data", which is the same configuration used in cough detection.

→ **Cough Calibration - Calibrating the Correlation Filter:**

It is observed that correlation filter content mostly resembles the spectrogram of a major cough (first cough) from a given cough group. Therefore, it is sufficient to capture the major cough only from the user. Hence, the system requires only coughing once for the extraction of the correlation filter.

Total number of FFTs in the correlation filter was previously set as Average Cough FFT Length (described in "∘ 3.3 - Implementation → Correlation Filtering"). However, since the current system only captures the major

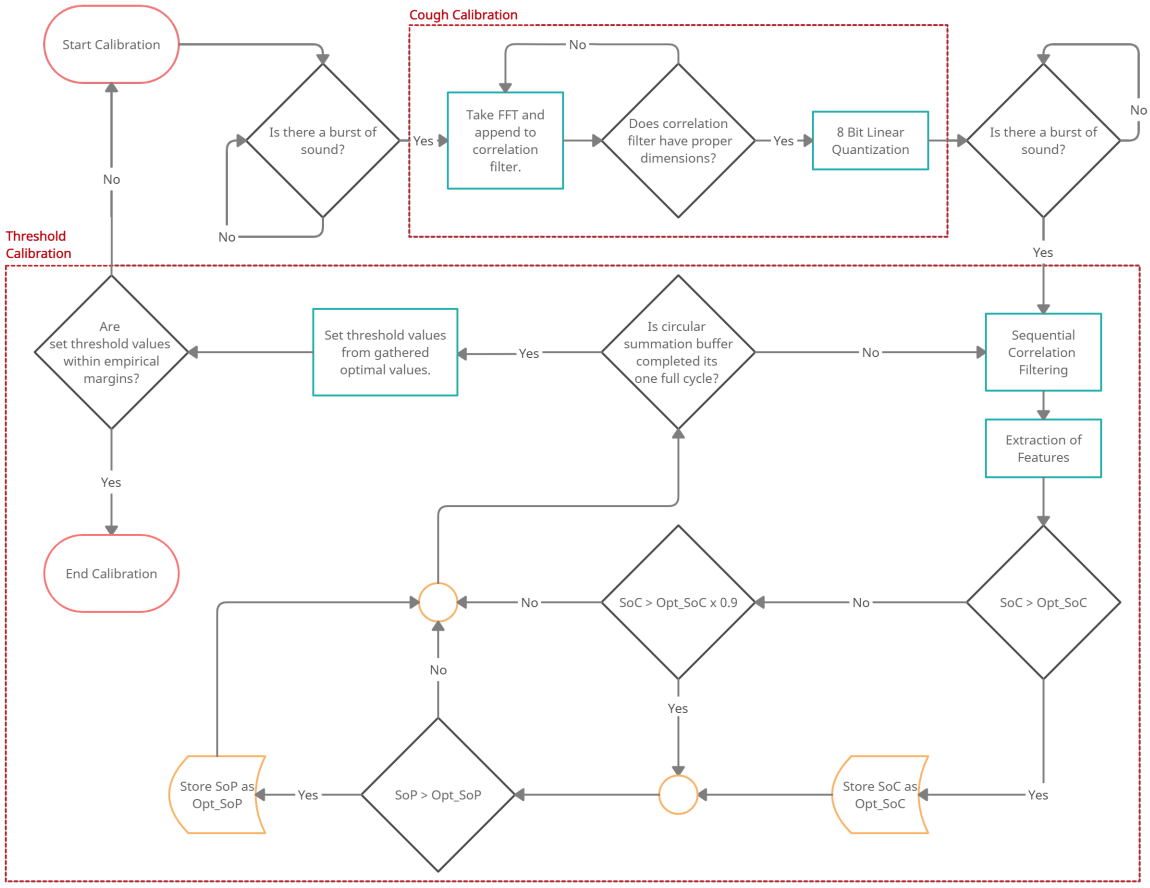


Figure 3.15: Calibration Scheme for Cough Detection

cough, filter length is fixed to 23 FFTs (which was the Average Cough FFT Length experimentally found in MATLAB, see Fig. 3.11).

Logarithmic magnitude of FFT calculations are appended to form logarithmic magnitude spectrogram of a single cough, similar to that of Fig. 3.9. Once all the FFT calculations are completed, logarithmic magnitude of the captured cough spectrogram is linearly quantized with 8 bits and filter coefficients are obtained.

→ **Threshold Calibration - Gathering Optimal Threshold Values:** After Cough Calibration finishes, system again starts to wait for a burst of sound. Therefore user is, again, required to cough once in a silent environment. The objective is to watch the output features during filtering with the calibrated correlation filter and then to select threshold values appropriately.

Once a burst is detected, sequential correlation filtering starts and lasts

for 29 FFT iterations, which corresponds to Average Cough FFT Length + 6 FFTs as it is implemented in cough detection (see "∘ 3.3 - Implementation → Correlation Filtering"). At each iteration, outputted feature values are observed. From the observed features, the feature tuple that would give the best description of a cough is saved. Then, this optimal feature tuple is used to set thresholds accordingly.

Assuming that all coughs have the same correlation output, it is ultimately desired to set the system such that detection of coughs is only outputted when the strength of correlation (SoC) is at its maximum (optimum). Therefore, in an optimal world, thresholds would be set to their respective feature values when SoC is at its maximum. However, neither the coughs of the user are identical nor the physical components are ideal. To account for different coughs, thresholds are needed to be set with a margin; and for non-ideal components and possible calculation errors, optimal feature values should be searched in the proximity of the maximum SoC.

Decision mechanism and the search for the optimal feature values are summarized on Fig. 3.15. At the end of all the FFT iterations for Threshold Calibration, optimal strength of correlation (Opt_SoC) is the maximum of the calculated SoC, while the optimal sharpness of peak (Opt_SoP) is the maximum sharpness of peak obtained within the 10% range of Opt_SoC.

On the finalized system thresholds are set as follows, where LTh_x and UTh_x are the lower and upper thresholds for the feature x;

$$LTh_{SoC} = Opt\_SoC \cdot (1 - 0.07)$$
$$LTh_{SoP} = Opt\_SoP \cdot (1 - 0.95)$$
$$UTh_{SoP} = Opt\_SoP \cdot (1 + 1.5)$$
$$LTh_{CP} = \frac{LTh_{SoC}}{UTh_{SoP}}$$
$$UTh_{CP} = LTh_{CP} + 6$$

Note that the set margins are found empirically and thus not strict. Also upper and lower thresholds for signal energy are set such that the energy of the explosive and voiced phases (see "∘ 3.3 - Implementation → Features, Thresholds & Outputs") are within the set boundaries. Time and burst thresholds are kept at fixed values. To further decrease false positives for

persistent forced inputs, time threshold is increased from 1 seconds to 1.5 seconds.

After the thresholds are calibrated, system checks if all thresholds are non-zero and CP thresholds are within the empirical boundaries. If all thresholds are acceptable, system finishes calibration and stores calibrated filter coefficients and calibrated thresholds to the microcontroller's flash memory. If thresholds are not acceptable, system clears all optimal feature values and re-starts calibration procedure from Cough Calibration.

- **4.4 - Conclusion:** With the newly added calibration features, cough detection subsystem is finalized. The system is fully integrated to the overall system. With proper calibration, the system performs as intended. Note that, even though calibration results are checked before completion, proper calibration cannot be ensured and is dependant on the user. If the calibration requirements are not met or the system performance shows significant deviations, calibration procedure should be manually re-started.

3.3 Fever Detection Subsystem

According to WHO, fever is one of the most common and serious symptoms of the COVID-19 [2]. As a chest strap, SYMON detects fever by measuring body temperature of the user, which is realized by fever detection subsystem. After taking measurements, via the communication subsystem past and current body temperatures are sent to the mobile application. According to the values, a warning is provided to the user visually. The subsystem is implemented via a digital temperature sensor SHT31. As of our final product, it is placed beneath the right armpit. The slit on the chest strap is placed accordingly. since the measurement error of the selected sensor is $\pm$ 0.3 °C , resulting fever information and current body temperature value are highly reliable.

The measurements are taken with 3 hours intervals since body temperature doesn't change very often. If the severity increases, automatically, the 3 hours intervals decreased to 1 hour intervals. Additionally, user have a chance to change periodicity of the measurements externally and to request current body temperature anytime using the interface of the mobile application. Moreover, 3 hour interval is selected considering a trade-off. If the temperature measurements are taken constantly, power consumption would

be much higher; however, the user data would be much better sampled. 3 hour periods are selected as our system specific optimum, under the assumption that temperature of the user does not have rapid fluctuations.

The output of the subsystem is directed to flash memory to be used in risk estimation and to be sent to mobile application via integrated BLE module of the microcontroller.

3.3.1 Design Process

Selection of the Sensor

A digital temperature sensor, SHT-31, is preferred to measure body temperature. It has $\pm$ 0.3 °C accuracy, which is enough to satisfy the requirement stated in Section "Requirement Analysis". Additionally, the integration of the sensor to the overall system is very easy, thanks to its size and compatibility with I^2C communication protocol. [15]. It also has low power consumption and is available in Turkey with an acceptable price of $10 when compared with other sensors 5.3.

Working Principle of the Sensor

When enough power is supplied (2.4 - 5.5 V) to the sensor module by the microcontroller, it measures physical values which is to be converted to ratiometric analog voltages. After a calibration is done, analog voltage values are converted to analog temperature values in Celsius, using data processing and linearization blocks inside the sensor module. Then, the analog temperature values are converted to digital temperature data using its own ADC [15].

Algorithm

SHT-31 senses its environmental temperature. Then, this data is directed to the microcontroller via I^2C bus. At every 3 hours, temperature is measured sequentially. Then, the sequentially measured data is averaged to give a more reliable estimate. The average temperature value is saved to the flash memory of the microcontroller to be used in risk estimation task. Then, the average is compared with a threshold value which is 37.2 °C (99 F), which is fever temperature value for human body according to the US National Library of Medicine (NLM) [16]. After the comparison, if current

body temperature is higher than the threshold, the measurement periodicity is decreased from 3 hours to 1 hour. If the temperature value is below the threshold again, the measurement periodicity increases to 3 hours again.

The resulting subsystem is illustrated in the functional block diagram in Figure 3.16.

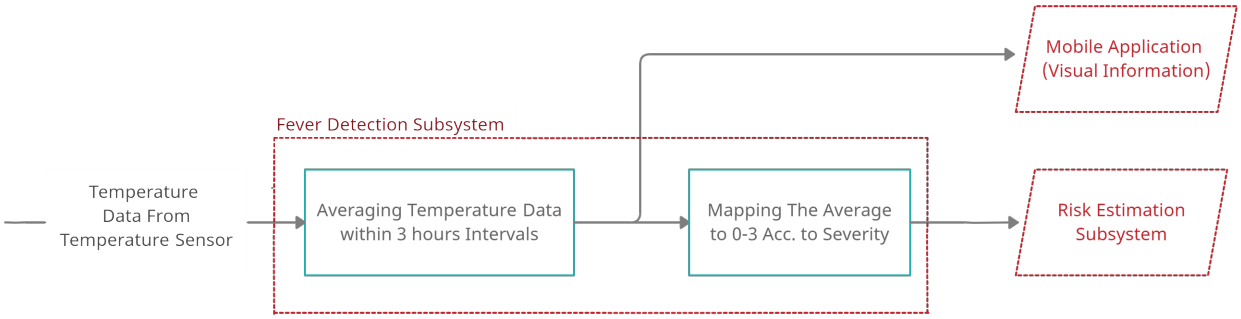


Figure 3.16: The functional block diagram of the fever detection subsystem.

3.3.2 Tests and Results

To verify that the requirement for this subsystem is satisfied, temperature data measured from SHT-31 is compared to the data of a commercial thermometer, Beurer. After waiting approximately 5 minutes, the temperature data of SHT-31 has reached to the body temperature from the room temperature as seen in the 'Initial SHT31 Data' curve in Figure 3.17. After this point, stabilization of the body temperature is occurred and plotted as the 'Stable SHT31 Data' curve in Figure 3.17. The 'Stable SHT31 Data' curve is compared to the curve of the commercial thermometer which is named as 'Commercial Therm.' in Figure 3.17.

As it can be seen in Figure 3.17, the values of 'Stable SHT31 Data' and the values of 'Commercial Therm.' become closer after the initialization time. If the data of the commercial thermometer is used as a reference temperature, the error due to the measurement of SHT-31 is less than $\pm$ 0.3 °C , which is less than the error that is set for the requirement of this subsystem, $\pm$ 0.5 °C as stated under Section 2.2.2. That is, this amount of accuracy is enough to satisfy the objectives that were previously set for the project and demonstrates the reliability of SHT31.

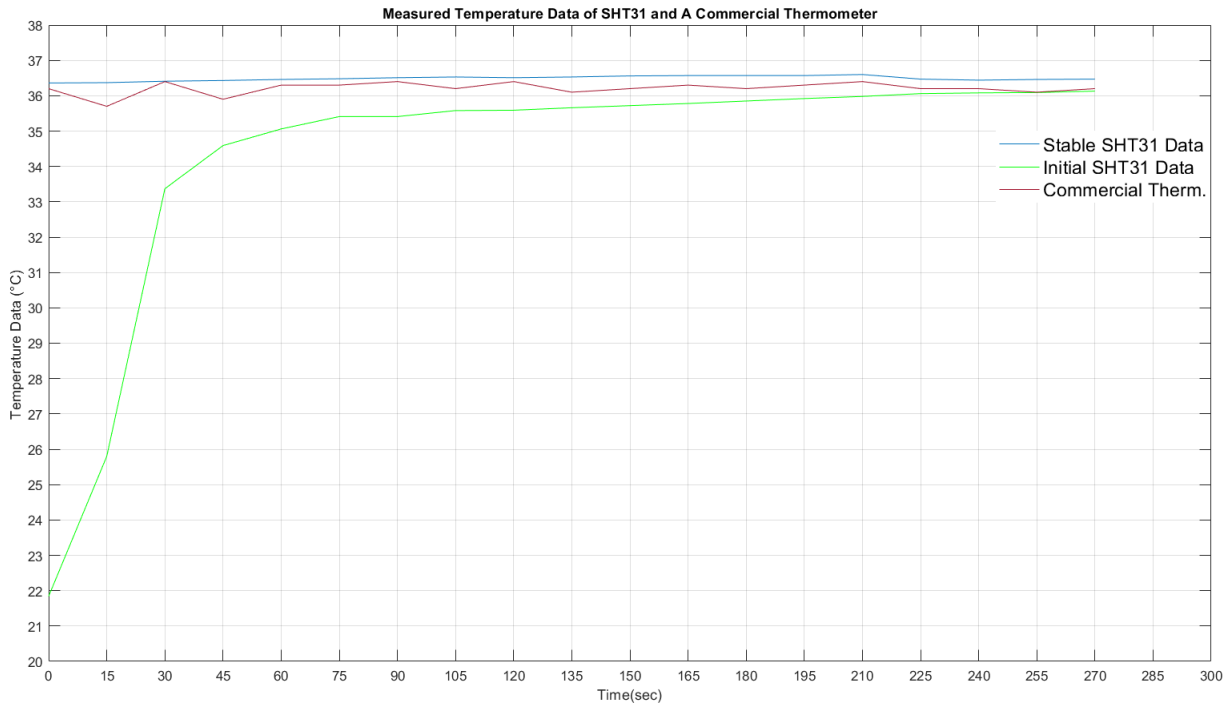


Figure 3.17: Test result of the fever detection subsystem.

3.4 Heart-Rate Measurement Subsystem

Palpitation is one of the detectable symptoms of COVID-19 according to Center for Disease Control and Prevention [17]. The main purpose of rapid heart rate detection is to provide information about the fast rhythm via the number of palpitations. It is implemented via an analog ECG sensor AD8232 which is compatible with a shape of SYMON. It also provides highly accurate measurements that allow to obtain conclusive results of the risk estimation process.

3.4.1 Overall Subsystem

The module processes real-time ECG signals, measures cardiac peaks and returns the required data of palpitation occurrences for the risk estimation process as well as heart-rate data.

The pulse value is returned after ECG signals are measured and processed twice a day, in the morning and in the evening by the microcontroller. Once the heart rate is above 110 bpm, the palpitation log initialises.

The periodicity of the measurements can be adjusted both manually. It is set to be more frequent when the risk of being infected increases. Also, the

user may adjust the periodicity from the Mobile Application.

The data obtained by this module is logged to the flash memory's respective file for that acquisition time. Then it is used by the risk estimation subsystem and also sent to the mobile application via BLE.

3.4.2 Design Process

Selection of the Sensor

The sensor has to measure the electrical signal that occurs due to heart beat. The ECG signal has a specific waveform shape and consists of certain peaks, see Figure 3.18. To obtain the pulse value, which can be further used for palpitation counter, QRS complex has to be detected. For the simplification of the calculation process, detection of QRS complex was reduced to the detection of R-peaks only.

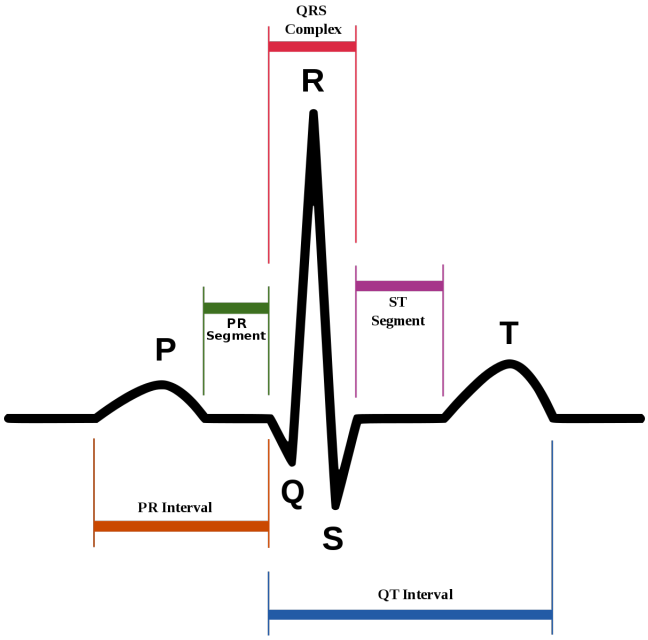


Figure 3.18: ECG of a heart rate in normal sinus rhythm .

The main objective for the sensor selection was ability to be mounted on the chest strap. Hence, available digital sensors are not applicable because they can only be placed on a transparent area. Beyond this fact, analog sensor provides highly accurate measurements compared to optical counterparts [18]. Ultimately, analog ECG sensor AD8232 was chosen to implement the Heart-Rate Measurement Subsystem. AD8232, can be found in Turkey at a price of $ 11.2. It has relatively low power consumption with 170 μA [19].

Working Principle of the Sensor

The ECG sensor consists of three electrodes. Black and blue electrodes measure the potential difference in the chest during the electrical activity of the heart, while the red one functions as a ground. The placement of the electrodes is shown on Figure 3.19. Hence, it is compatible with the shape of SYMON and can be easily mounted on the chest strap.

AD8232 chip serves as an amplifier of the signal because the potential difference between the electrodes is relatively small. The signal is further being filtered in order to remove possible galvanic interference. However, during the measurements the user should stay still in order to prevent noise interference caused by movement.

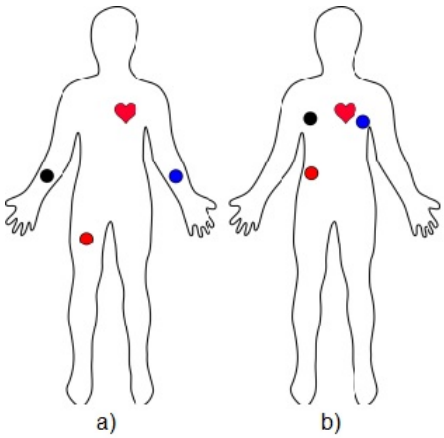


Figure 3.19: The electrode placements at the Eithoven Triangle: a) Limb Placement b) Chest Placement .

Algorithm

The algorithm is designed to process ECG signals, detect R-peaks, measure time interval between them, calculate the pulse value and count the number of palpitations. The pipeline is presented on figure 3.19. The bpm value is used to inform the user via Mobile Application, while the number of palpitations are further sent to risk estimation subsystem.

R-peaks are detected when the signal crosses a threshold value. This threshold corresponds to 85% of the maximum of the ECG signal's amplitude Next, time interval between two consecutive peaks is measured. The bpm is calculated by dividing one minute by the time interval. In order to provide a more precise value, measurements are averaged. A threshold is applied to the averaged value in order to warn user about rapid heart rate and count the

Figure 3.20: The functional block diagram of the heart rate subsystem

number of palpitations. The threshold for the rapid beat is set to be 110 bpm as it corresponds to fast-beating [20]. When this rapid heart-rate occurs, the period is increased such that the heart-rate measurement is obtained every hour. When the heart-rate is decreased back to a value below 110 bpm, the period of the subsystem is turned back to its default period, that is 8 hours.

The palpitation log is created inside the heart-rate measurement subsystem. This enables SYMON to keep the number of palpitation episodes for a duration of 24 hours. When the rapid heart rate is detected by the module, the number of palpitations value for that hour is increased by one. If another measurement is taken in the same hour which results in rapid heart rate, the number of palpitations value for that hour again is increased by one. At the beginning of a new hour, the value is reset to 0. with this upgrade, the need for palpitation data for estimating the risk has been fulfilled.

3.4.3 Tests and Results

The performance of the subsystem is tested in terms of accuracy and threshold activation.

The real-time pulse value, obtained by AD8232 and the microcontroller is compared to the data measured by the smartwatch. During the testing procedure, the wearer stayed still. The values are compared and the approximation error is found to be 2.82% . It meets the set requirement of approximation error being less than 5%. Hence, obtained pulse value was concluded to be reliable. The comparison results are shown on the Figure 3.21.

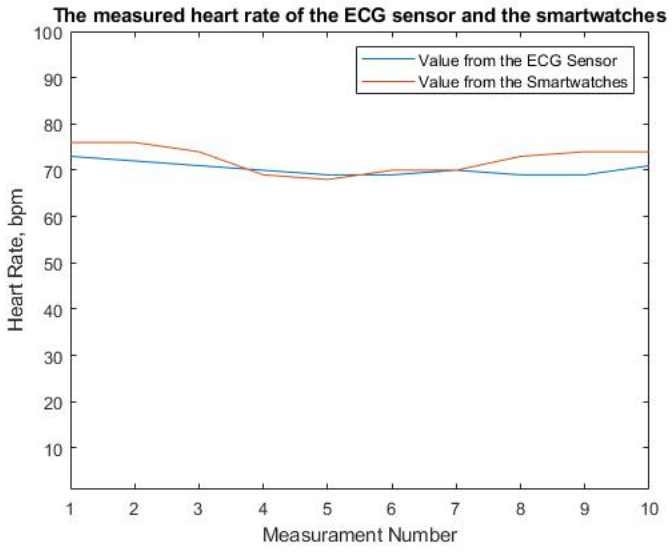


Figure 3.21: The measured heart rate of the ECG sensor and the smartwatches.

During another testing procedure the heart rate of the user was artificially increased by performing exercises. When the electrodes were connected the module, it displayed alert message 'rapid heart rate' and the bpm value. While resting, the heart rate dropped below the threshold, 'normal heart rate' information and the bpm were shown on the serial monitor of the microcontroller. It was concluded that threshold activation works accurately and indicates that trigger for the palpitations counter works successfully.

3.5 Communication Subsystem

In order to satisfy the requirements, Bluetooth Low Energy (BLE) module of the microcontroller is preferred rather than other communication protocols such as WiFi or Bluetooth Classic. When compared with other communication methods, BLE has lower power consumption because it is designed to be in sleep mode when data is not sent or received. Also, it is used for short ranges, which means power released to air is lower than the modules for long ranges. Furthermore, since BLE was embedded on the used microcontroller it does not require additional purchases or external integration [21].

3.5.1 Implementation

In the case of project SYMON, data transfer will be two sided meaning that, both devices can be used as server, which broadcasts data, or client,

which listens data as can be shown in Figure 3.22 according to the situation. The detectable symptom data will be provided to mobile application while undetectable symptom data and requests will be sent to ESP32 from the mobile application through BLE connection. This two sided communication is also crucial for cough calibration process.

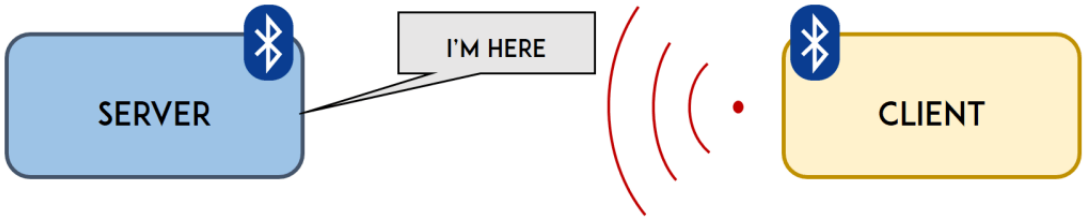


Figure 3.22: Advertising server and client in BLE.

In the setup code of the Arduino sketch (setup function runs only once upon powering on and is not executed again), the BLE characteristics are defined and initialized. This operation makes BLE of ESP32 visible to other devices.

Cough Calibration Communication

To start the cough calibration process, mobile application sends a char array as "CC" (initials of Cough Calibration). When ESP32 receives this value, it starts the filter calibration part of the cough calibration and sends "FCS" (initials of Filter Calibration Started) to mobile application. When mobile application receives this value, it requests the user to cough once. After user coughs, the threshold calibration starts automatically and ESP32 sends "TCS" (initials of Threshold Calibration Started) to mobile application for requesting the user to cough one more time. Once again, the mobile application requests the user to cough once. After this last cough, if the calibration is successful, ESP32 sends char array "CCS" (initials of Cough Calibration Successful), otherwise it sends "CCU" (initials of Cough Calibration Unsuccessful). When CCU is sent from ESP32 to mobile application, the cough calibration process is started once again automatically.

Sending Sensory Data from ESP32 to Mobile Application

For sending values from ESP32 to mobile application, the BLE sending operation has been defined as a task in the main code. Inside the task, the measurement values are read from the flash memory of the ESP32 and

merged as a char array. The arrays are constructed such that, first character will be the indicator of the symptom. For instance, if first letter is T, it means that the rest of the data of the array is about temperature measurement subsystem. The other possible values for this char are C which indicates the data belongs the cough detection subsystem and H indicating the data belongs to heart-rate measurements.

The second character of the char array indicates the time of the obtained data. As explained in Section 3.1.3, the created hourly clock inside the ESP32 enables the classification of the acquired data according to their time instances. For instance, as in the codes for the circular flash, if this second char is A, it means that the linked data to it is acquired in the first hour that the ESP32 is powered on. If it is B, it means data belongs to the second hour, etc. Since there are 24 hours, the letters values can be from A to X.

The remaining part of the array consists of the numbers which belongs to the measurement result. To give a better explanation, Figure 3.23 shows the structure of an generic array and 3.24 shows three examples of the coded sensory data that is sent from ESP32 to mobile application.

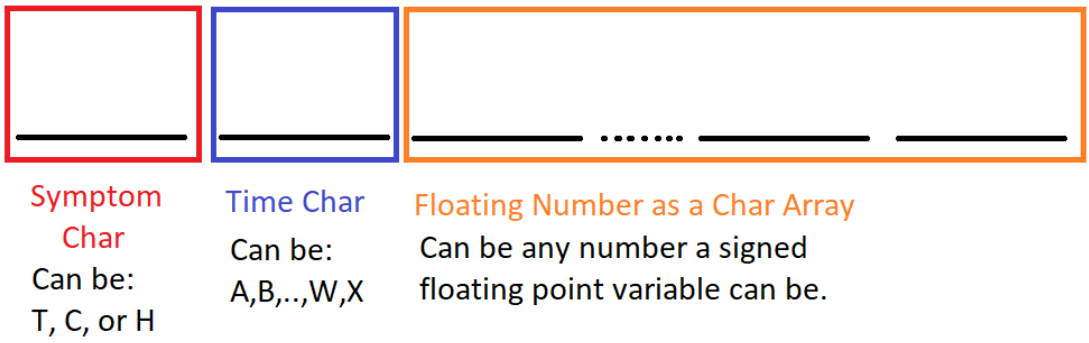


Figure 3.23: The general form of coding the data for sending from ESP32 to mobile application.

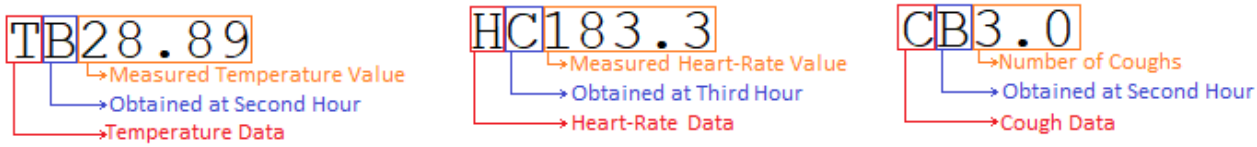


Figure 3.24: Some examples of BLE sent data from ESP32 to mobile application.

When the BLE characteristic of the ESP32 is changed to the described array, the microcontroller starts advertising to be noticed as in Figure 3.22. If any other BLE device is connected to the microcontroller, the data transfer starts.

When BLE period (the period of detectable data synchronization) is larger than 1, the detectable symptom data since the last exchange is sent as a bulk. For example, when BLE period is 3, last 3 data of the temperature measurement should be send. Therefore, the sent array can be such as TF37.23/E36.12/D36.54. As explained above, T indicates that the rest of the array belongs to temperature measurement subsystem. F indicates the 6th hour and the floating point number next to it is the most recent temperature data (the data acquired at the 6th hour). The data belonging to different hours are separated by using "/". In this array, D is the furthest in terms of time and the data next to it is the oldest data. The data next to D is also the first data that is acquired after the previous BLE connection.

If a subsystem did not take any measurement in an hour, "-1" is written by default. If there are not any coughs detected, "0" is written to respective bins.

Requests from Mobile Application & Warnings from ESP32

In order to enhance the user experience, some features such as changing periodicity of the measurements, requesting measurements, requesting risk estimation are added. The requests are encoded as a char array and sent from mobile application to ESP32. These arrays and the functions they evoke are presented in Table 3.8. The requests for changing period are sent with the new period value that is represented with * in the table. * is a char and its value changes from B to X, B corresponding to 1 and X corresponding to 24. When period is set as 1 for a subsystem, for the heart-rate measurement as an example, it starts to take measurements per hour.

ESP32 also sends some warning messages to mobile application. These messages are presented in Table 3.9. The first two char arrays in this table are the arrays that are sent to inform the mobile application that the periodicity of the temperature or heart-rate system has become one. This case occurs when a temperature or heart rate value exceeds their set thresholds. In order to detect the symptoms better, their periodicity are changed to be 1 (represented with letter B) inside the ESP32 automatically and the

Table 3.8: The requests that are sent from mobile application to ESP32.

Char Array	**Command**
RMT	Request Measurement - Temperature
RMH	Request Measurement - Heart-Rate
*CPT**	Change Period - Temperature - *: New Period
*CPH**	Change Period - Heart-Rate - *:New Period
*CPB**	Change Period - BLE - *:New Period
RE	Request Risk Estimation

Table 3.9: The warnings that are sent from ESP32 to mobile application.

Char Array	**Warning**
PCTA	Periodicity Changed - Temperature - A:New Period
PCHA	Periodicity Changed - Heart-Rate - A: New Period
XRE	Risk Estimation Unsuccessful
SS	Stay Still

information of this alteration is sent to mobile application with these two arrays.

"XRE" is sent when ESP32 has no prior undetectable symptom data. In this case, 8 of the symptoms that are needed for risk estimation are missing. Therefore the subsystem sends a warning message to mobile application to inform the user to log undetectable data before requesting risk estimation. "SS" is sent 10 seconds before periodic heart-rate measurements are taken since the user needs to stay still during heart-rate is measuring.

3.5.2 Tests and Results

The tests of this subsystem are conducted using the Serial Port of the Arduino IDE and the terminal of the Xcode, the software that the mobile application is coded. Different cases are inspected in the testing procedure.

Testing of Cough Calibration

The cough calibration messages are tested both for the successful and unsuccessful cases. The test results are represented in the Figure 3.25 and Figure 3.26. As can be seen from the figures, in the first case, the CC value is received from the mobile application and filter and threshold calibrations start one after the other. The received FCS and TCS values by the mobile application can be seen on the Xcode terminal side. And since it is the successful case, at the end of the process, CCS is sent to the mobile application. In the other case, since threshold calibration is unsuccessful (indicated by TH Pass=0 in the Serial Port), CCU is sent. The new calibration is started automatically upon sending CCU value. The values are sent by ESP32 to the mobile application and received by the mobile application (and vice verse) as intended.

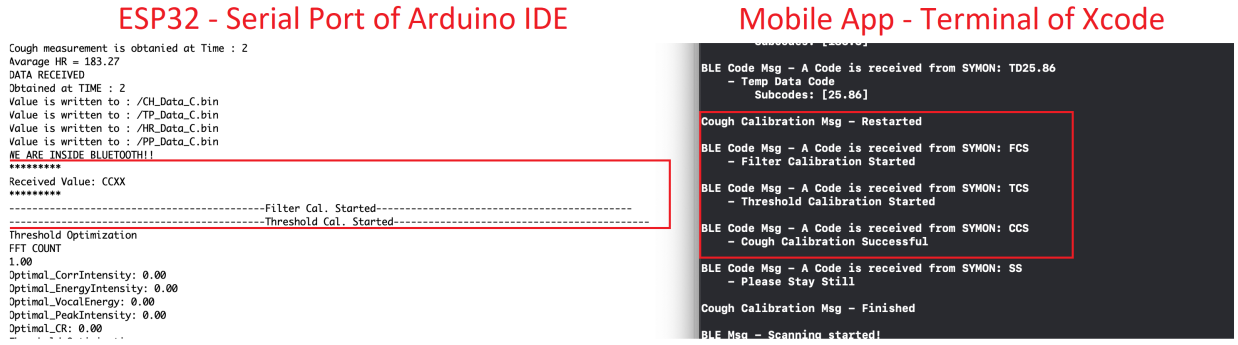


Figure 3.25: Successful cough calibration case.

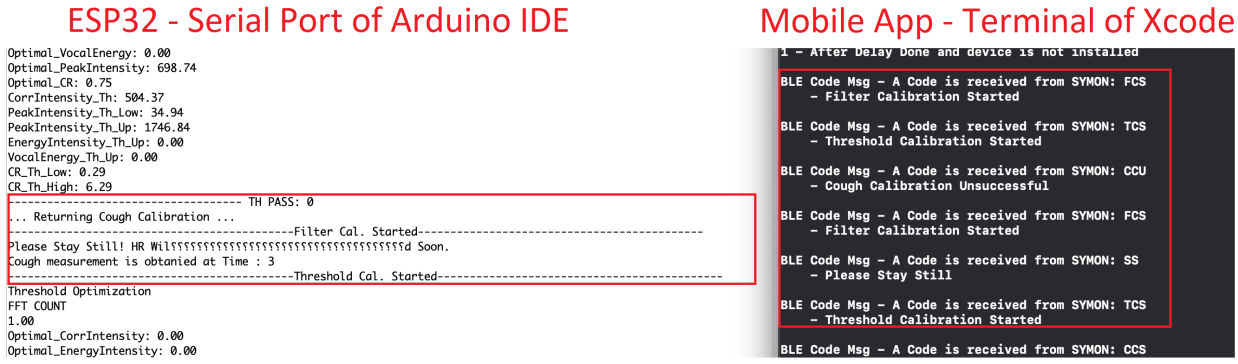


Figure 3.26: Unsuccessful cough calibration case.

Testing of Requesting Measurements

In these tests, a temperature measurement, see Figure 3.27 and a heart-rate measurement is requested, see Figure 3.28. These requests are received by the ESP32 and it takes the relevant measurements and sends the requested values to the mobile application. Hence, the feature works as intended.

Figure 3.27: Requesting temperature measurement case.

Figure 3.28: Requesting heart-rate measurement case.

Testing of Period Change

The command of changing period is sent by the mobile application for systems: temperature measurement, heart-rate measurement and BLE communication subsystem. These requests are received by the ESP32. Hence, it can be said that the communication for this functionality works as intended. Since there was not any indicator of this period changing command on the Xcode terminal, in the Figure 3.29 terminal part is omitted.

Testing of Risk Estimation

Both the successful and unsuccessful risk estimation cases are tested for this part. Unsuccessful risk estimation case occurs when SYMON initializes

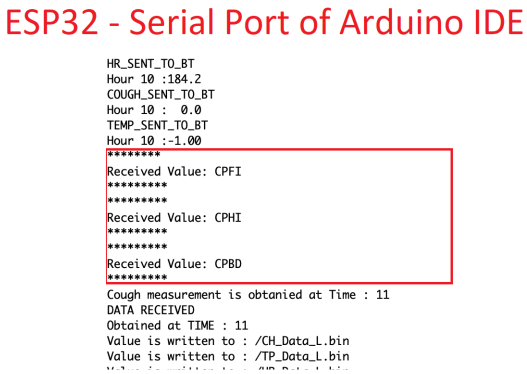


Figure 3.29: Period changing command case.

the periodic risk estimation however cannot find a previously logged undetectable symptom value. Since risk estimation is periodic with 16 hours, in Figure 3.30, on the ESP32 side, it can be seen that time is 16 and risk estimation will start automatically. However since there was no undetectable symptom data sent previously, it sends "XRE" value to the mobile application indicating the unsuccessful risk estimation attempt.

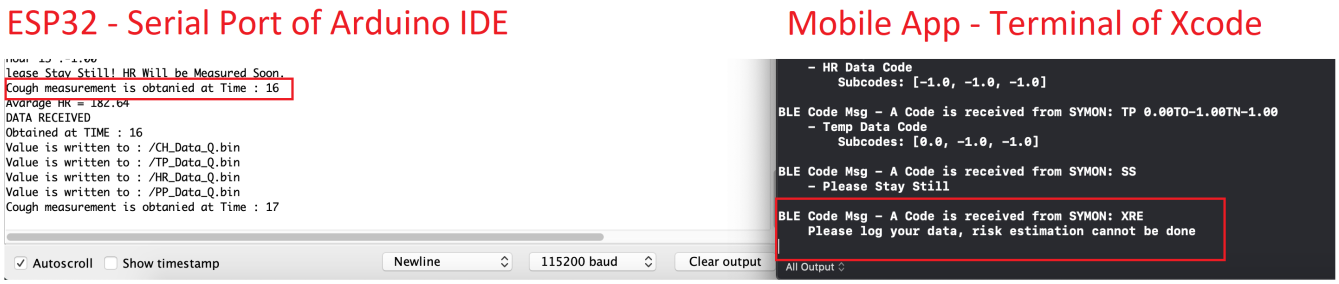


Figure 3.30: Unsuccessful risk estimation case.

In the successful estimation case, the risk estimation request is sent along with undetectable symptom data and as a result, estimated risk and confidence score is sent back to mobile application by ESP32 which can be seen from Figure 3.31.

Testing of Periodic Sensory Data Transfer

The data of the symptoms are acquired by the sensor periodically and it is sent through BLE to the mobile application without a need of requesting measurements. This function is tested in this part. And in Figure 3.32, it can be seen that the values obtained from the sensors connected to ESP32

Figure 3.31: Successful risk estimation case.

are sent successfully to the mobile application.

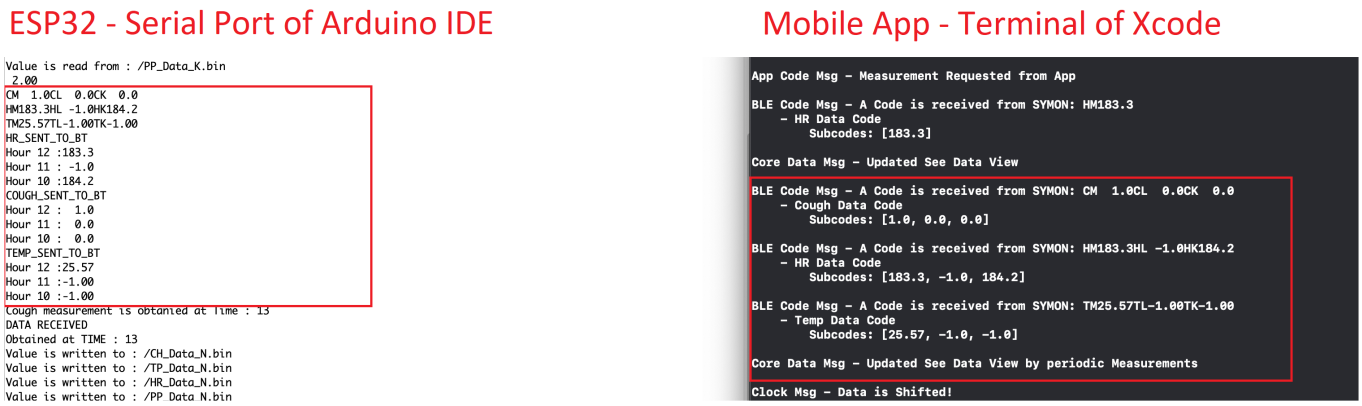


Figure 3.32: Periodic sensory data transfer case.

Conclusion

As a result, these tests demonstrate that the sensor data of SYMON can be transmitted to mobile application without any loss. Additionally, as it can be seen from [22], power usage of BLE module does not contribute highly to the overall power usage of the system. These suggests that the communication subsystem satisfies the requirements that are set in Chapter 2.

3.6 Mobile Application Subsystem

Mobile application subsystem is designed in order to give an interface of SYMON to the user. Front-end of the mobile application offers a user-friendly, visually appealing interface, while back-end of the application has the powerful functionalities including Bluetooth Low Energy (BLE) functions in order to be able to communicate with SYMON. In the following parts, the design process and the structure of the mobile application are

explained.

3.6.1 Design Process

Selection of Platform

Mobile application is developed as a native application, which means that it is compatible for a specific platform. The application is built for iOS platform using the Swift programming language. Even though an app for multi-platform offers lower cost and wider users, the cross-platform app development results in less powerful functions and lower performance of the system than that of native app development [23]. Since the performance was set as a significant requirement in Section 2.2.5, development of a native app was preferred. To reach more users, the mobile app should be developed either on iOS or Android, which dominate the mobile operating system market with 99% share [24]. iOS platform was chosen due to the fact that Apple designs all of their hardware and software components of their products. Therefore, the iOS platform is quicker and more responsive [25], which simplifies the development of the app to make it more powerful.

Main Approach

Mobile application is designed based on MVVM (Model-View-ViewModel) paradigm, which is a software architectural pattern that separates back-end logic from front-end of the app. The paradigm deals with three parts of the mobile application which are model, viewmodel and view. Model is UI independent back-end of the mobile application, which contains data and logic of the system, whereas view only reflects the model. On the other hand, viewmodel binds the view to the model which acts as an interpreter in the system.

In the designed mobile application, the model consists of functions about persistent update and storage of the gathered data about COVID-19 symptoms. This can be achieved with the use of a library called Core Data that is compatible with the Swift programming language [26]. Unless the Core Data model is used, all the data will be lost and returned to the default values whenever the application is terminated. Since not only the last hour data for the COVID-19 risk estimation is used as explained in Section 3.7, the persistence of the data of at least last 16 hours is critical for risk estimation.

Also, for the purposes of displaying the user's data statistics, it is decided to demonstrate the last 2 week's data to the user. Therefore, symptoms data for 14 days are stored and updated at each day. To be able to handle this update, another library called Calendar is used such that the change in the day and the change in the hour can be detected and the updates can be applied. Also, for the logic of updates, up-to-date calendar information is persisted in Core Data model.

Functions about BLE connection such as advertising, connecting/disconnecting to the peripherals, discovering services, reading/writing to characteristics' values are designed inside the back-end of the mobile application in the model. These can be implemented with the use of Core Bluetooth library, which is compatible with the Swift programming language [27]. Thus, the communication between mobile app and SYMON can be supported with these functions. When one of the commands, specified in Table 3.9, is received, the corresponding functions written in the back-end are executed. To be able to detect the command coming from SYMON accurately, a decoding system is also designed in the back-end of the mobile app with the use of basic switch-case statement codes. With the accurate decoding and execution of corresponding functions in the back-end, the front-end of the app updates itself continuously. Moreover, the transmission of the commands from mobile app to SYMON, which are listed in Table 3.8, are handled in the back-end logic of the app. However, this back-end logic is triggered by front-end tools such as pressing the specific button. The algorithm about this interaction can be seen from Figure 3.33

As the front-end of the app, convenient and visually appealing user interface is designed, since it is set as one of the most significant requirements in the Section 2.2.5. The interaction between different views with different buttons can be seen in Figure 3.33. Navigation between different views are designed to be done by the use of buttons. Each button has a label text of the button names in Figure 3.33, which guides the user about the logic of mobile application and makes the app simple as possible.

3.6.2 Structure

The developed application has eight different views, where the detailed features are explained as following:

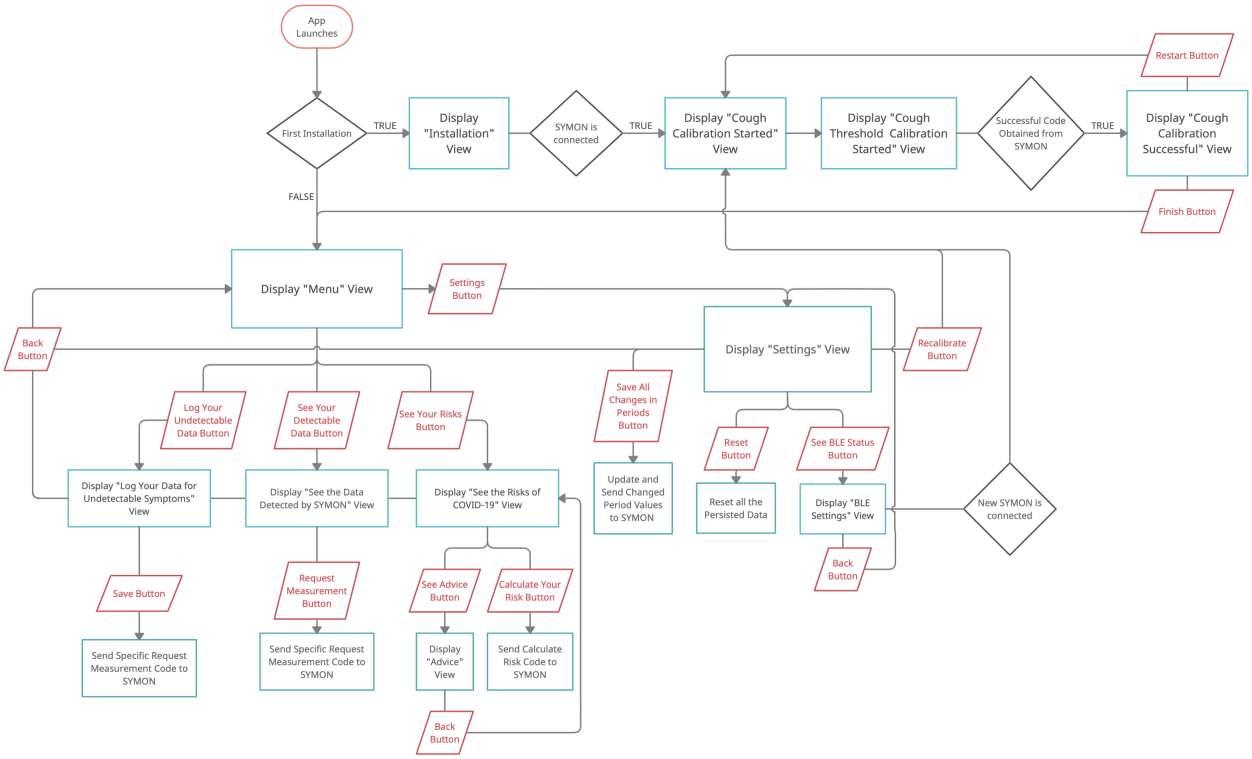
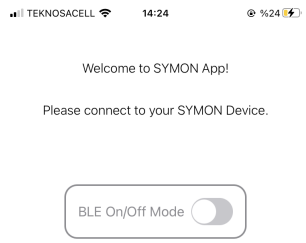


Figure 3.33: Block diagram of mobile application's user interface.

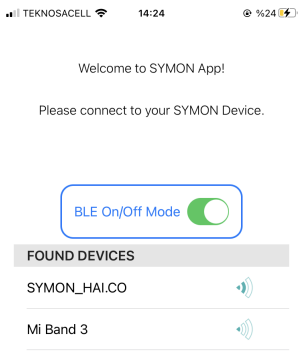
Installation View:

Application has primarily two distinct states, which are 'not installed' and 'installed'. The former one represents that the app is newly downloaded and it waits to be connected to a SYMON device by the user. If the state of the app is the former one, when the application launches, the 'Installation' View is displayed to the user, which can be seen in 3.34a. Inside the view, there is a text welcoming the user to the application, named SYMON App. Another message is also written to guide the user to turn the BLE on in order to connect the app to SYMON app. When the button is pressed to turn the BLE on, the devices that are found are listed in the view, which can be seen in Figure 3.34b. Also, a visual tool near the name of the found device is used to demonstrate the strength of the BLE connection. In order to connect one of the found device, the user should tap on the region where the corresponding device name and its strength symbol are displayed.

After the connection, the application automatically sends the "Cough Calibration" command that is explained in Section 3.5. This is because the application is newly installed and there is no cough sound in SYMON de-



(a) **Screenshot of Installation View**.



(b) **Screenshot of Installation View after the user turned BLE on.**

Figure 3.34: Screenshots of Installation Views with different possibilities.

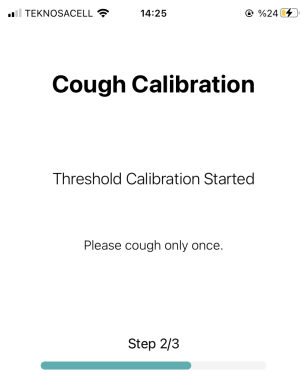
vice that is to be used for cough detection subsystem. Then, the application waits for the response command of "Cough Calibration Started" from SYMON. Whenever this command is received, "Installation" view is closed and "Cough Calibration" views are displayed. After that, the "Installation" view will never be displayed as the mobile application state is converted to 'installed'.

Cough Calibration Views:

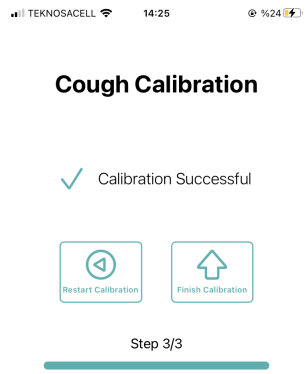
After the device is firstly connected by mobile app, "Cough Calibration" command is automatically sent. As a result, "Calibration Started" command is received and corresponding view is displayed to the user, which can be seen from Figure 3.35a. This is the first step of the cough calibration process, which is also indicated to the user in the view. Also, a message that reminds the user to cough only once is written. Whenever the user coughs properly, "Threshold Calibration" command is received and the "Threshold Calibration Started" view is displayed as the second step of cough calibration process, which can be seen from Figure 3.35b. The same message that reminds the user to cough only once is written. Whenever the user coughs properly again, the "Cough Calibration Successful" command is received and "Cough Calibration Successful" view is displayed as the final step, which can be seen from Figure 3.35c. If the cough calibration cannot be done properly,

then the system automatically restarts the cough calibration process. Otherwise, the user can either press "Restart Calibration" or "Finish Calibration" buttons. The former one restarts the cough calibration process, while the latter one closes the view and opens "Menu" view.

(a) **Screenshot of Cough Calibration Started View.**



(b) **Screenshot of Threshold Calibration Started View**



(c) **Screenshot of Cough Calibration Successful View.**

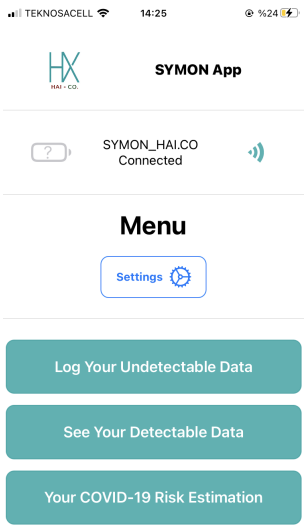
Figure 3.35: Screenshots of Cough Calibration Views at different steps.

Menu View:

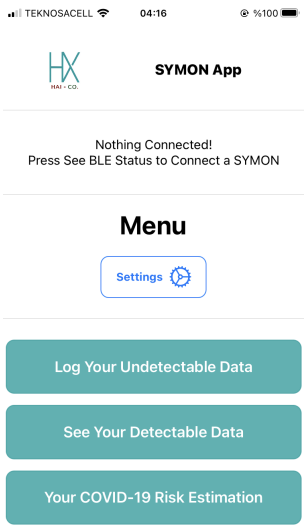
When the application is launched and it is in the 'installed' state, "Menu" view is automatically displayed to guide the user what to do in the mobile application. As it can be seen in Figure 3.36, there are four buttons that are labeled as "Settings", "Log Your Undetectable Data", "See Your Detectable Data" and "Your COVID-19 Risk Estimation". Each of them navigates to their corresponding views in the mobile application according to their label texts, which makes the app user-friendly. Additionally, in this view, there is a region with text indicating whether a device is connected or not. When the device is not connected, it warns and guides the user to connect a device as it can be seen from Figure 3.36b, whereas when the device is connected, the text includes the name of the device as seen from Figure 3.36a.

Log your Data for Undetectable Symptoms View:

When "Log Your Undetectable Data" button is tapped in the "Menu" view, this view, which can be seen in Figure 3.37, is displayed. The purpose



(a) **Screenshot of Menu View when connected.**

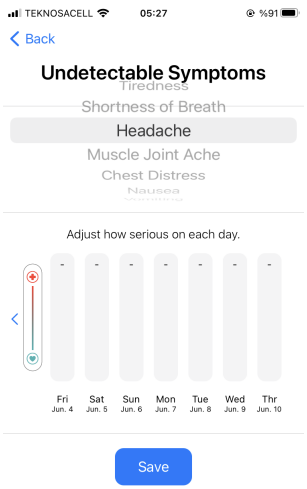


(b) **Screenshot of Installation View when not connected.**

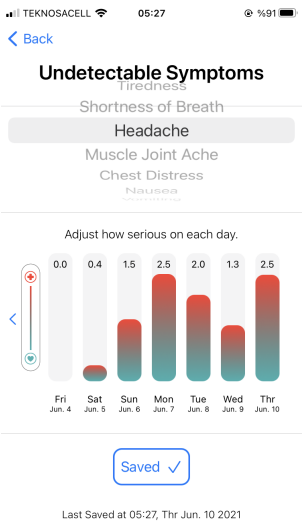
Figure 3.36: Screenshots of Menu Views with different possibilities.

of this view is to obtain the data from the user for undetectable COVID-19 symptoms. To estimate the risk score properly and reliably, the data for each symptom is asked to the user for each day with its severity values. To make this understandable and easy, scroll bars for each day are used to adjust the severity level whose values ranging from 0 to 3, as seen in Figure 3.37b. Since it is decided to store the data for two weeks, data of other seven days can be seen by scrolling to the left or right by dragging gesture or tapping to the appeared left or right button.

To prevent unwanted changes continuously, a "Save" button is added to log the data; therefore, unless the "Save" button is tapped, no data will be persisted. Last saved information is also displayed under the "Save" button to remind the users when they have modified before. When the day is changed, the view is updated by back-end of the app with the use of the library called Calendar and new day's data will seem 'not logged' as '-'. This type of bars can be seen from Figure 3.37a. The purpose of this symbol is to make the users realize that they never logged any data for the corresponding day. However, if "Save" button is pressed, then all the 'not logged' data for each symptom is converted to 'logged' data with value 0 automatically. This idea is implemented in order to simplify user experience by asking the user to log only the symptoms that they carry. Additionally, when "Save" button



(a) **Screenshot of Log Your Undetectable Data View without pressing the 'Save' button.**



(b) **Screenshot of Log Your Undetectable Data View pressing the 'Save' button.**

Figure 3.37: Screenshots of Log Your Undetectable Data Views with different possibilities.

is pressed, the app automatically sends "Risk Estimation" command with newly logged data to the device in order to obtain up-to-date. The purpose of this is to make the app up-to-date and responsive to the new changes.

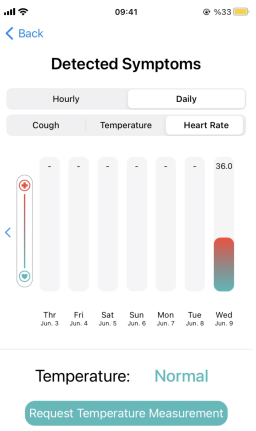
See Your Data Detected by SYMON View:

When "See Your Detectable Data" button is tapped in the "Menu" view, this view, which can be seen in Figure 3.38, is displayed. The purpose of this view is to display responsively the total number of coughs, average body temperature values and average heart rate values of the user for last 14 days, where all the data was obtained from self-monitoring device, SYMON. The data can be seen as either hourly or daily. For testing and demonstration purposes, the hourly data is converted to minutely data, which can be seen from Figure 3.38a. Similar to the "Log your Data for Undetectable Symptoms" View, the user can see the data of other seven days by scrolling to the left or right by dragging gesture or tapping to the appeared left or right button.

This view is developed to be responsive, which means that it continuously updates itself whenever new values are arrived at the app from SYMON. Also, the shifting of the data according to the date is implemented. This is achieved by timer implementation inside the app.

One feature of this view is that the user is able to request measurement at

(a) **Screenshot of See Your Data View for minutely temperature values .**



(b) **Screenshot of See Your Data View for daily temperature values .**

Figure 3.38: Screenshots of See Your Data Views with different possibilities.

anytime. This can be done by pressing the button which labeled as "Request Cough Measurement", "Request Temperature Measurement" or "Request Heart Rate Measurement". When the button is pressed, the corresponding command is sent to the device, and the response is waited. When the expected data arrives, the modification is implemented and the user is able to see the data instantly. In addition, when the button is pressed and the device is not connected, then the system gives a pop-up message to the user to warn about the situation.

Additionally, a text indicating whether the system severity level is normal, low and high is displayed under the graphics, which can be seen from 3.38. This is designed to inform the user about the severity of the detected symptoms data.

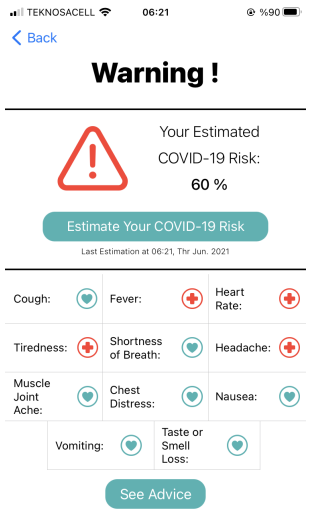
See Your Risks of COVID-19 View:

When "Your COVID-19 Risk Estimation" button is tapped in the "Menu" view, this view, which can be seen in Figure 3.39a, is displayed. The main purpose of this view is to display the risk score and non-detailed user's data about COVID-19 symptoms. Red cross symbols are used to warn the user that the corresponding symptom has high severity level, while green heart symbols are for low severity level.

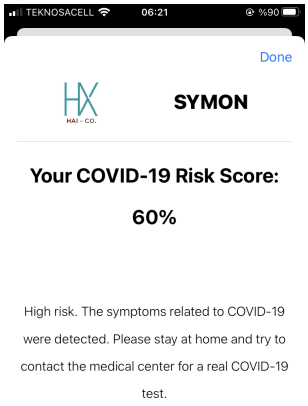
There is a button under the risk score, which is labeled as "Estimate Your COVID-19 Risk". When this button is pressed, "Risk Estimation"

command is sent to the device and the new risk score value is expected from the device. Whenever the risk score is arrived, the displayed value is updated and the date information under the button is updated. If the device is not connected, then the system gives a pop-up message to the user to warn about the situation. Another pop-up message is given if the button is pressed without logging any undetectable symptoms data. This is to prevent the calculation of risk score with inadequate number of symptoms data.

There is a button down on the image named "See Advice". When the user taps on this button, a new view named as "Advice" view as seen in 3.39b is displayed to give some advice to the users according to the level of their risk. The exact warning messages and the



(a) **Screenshot of See Your Risks of COVID-19 View**.



(b) **Screenshot of Advice for Your Risk Score View.**

Figure 3.39: See Your Risks of COVID-19 and Advice for Your Risk Score Views.

Based on the estimated risk score, "Advice" view displays the warning message that includes information about the risk being infected and several precautions. The messages will be in the following form:

- $= \mathbf{0\%}$: No risk. Congratulations, you are healthy! Don't forget to wear the masks and wash your hands.
- $> \mathbf{0\%}$ & $\leq \mathbf{25\%}$: Low risk. Some symptoms were detected. Good news, they are not critical. Take a deep breath and have some rest. Don't forget to wear the masks and wash your hands.

- $>$ **25% &** $\leq$ **50%** : Moderate risk. Sir, you are getting sick!. It can be 50% common cold. Be careful with your nutrition an sleep cycle! Don't forget to wear the masks and wash your hands frequently.
- $>$ **50% &** $\leq$ **75%** : High risk. The symptoms related to COVID-19 were detected. Please stay at home and try to contact the medical center for a real COVID-19 test.
- $>$ **75% &** $\leq$ **99%** : Extreme risk. Houston, we have a problem! Please contact the medical center and do the real COVID-19 test as soon as possible.

Since HAI-CO. is not a certified medical organisation, it was decided that the maximum risk can be 99% rather than 100%.

Settings View:

When "Settings" button is tapped in the "Menu" view, this view, which can be seen in Figure 3.40, is displayed. The main purpose o this view is to offer the user to customize the device. At top of the view, there is a section called "See BLE Status". It navigates to the "Adjust BLE Status" view, where the user can modify the BLE connection of the app.

There is another section called "Periodicity of Data Acquisiton", where the period values can be changed with "Decrement" and "Increment" buttons. In order to update the changed periods, the user must press the button named "Save All Changes in Periods". It sends the changed value's "Change Periodicity" command to the device. If the device is not connected, then the system gives a pop-up message to the user to warn about the situation. Also, if the automatic period changes occur, the period values update itself in this view.

Another section in this view is called "Recalibrate Your Cough". This is to restart the cough calibration process. When the button is pressed, it automatically sends "Cough Calibration" command to the user and respond is waited. As soon as the app receives the respond, "Cough Calibration Started" view is displayed. If the device is not connected, then the system gives a pop-up message to the user to warn about the situation.

Final section in this view is called "Reset". The purpose of this button is to reset all stored data to the factory default values. It also disconnects the

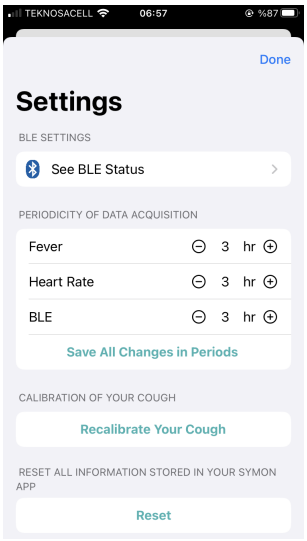


Figure 3.40: Screenshot of Settings View.

device. However, it does not transfer the app into the 'not installed' state, as the installation message will be unnecessary in that part.

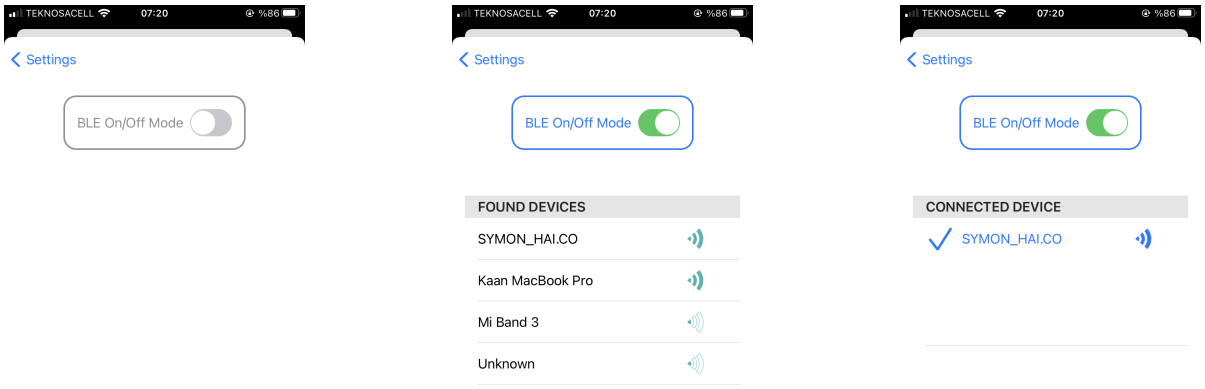
Adjust BLE Status View:

When "See BLE Status" button is tapped in the "Menu" view, this view, which can be seen in Figure 3.41, is displayed. The main purpose of this view is to make it possible to connect any SYMON device to the mobile application anytime or disconnect the BLE connection directly by turning BLE mode off. As seen from Figure 3.41a to Figure 3.41b, found BLE peripherals are listed when the app is not connected to a peripheral and BLE is turned on. When the user taps on one of the found devices, it connects to the device as seen in Figure 3.41c.

Background Tasks:

Functions in BLE model are executed in the background. In other words, even though mobile application is closed, BLE connection continues to receive data from SYMON in order not to lose any information about COVID-19 symptoms of the user. This is critical for the application to be up-to-date and responsive to the events such that it can warn the user immediately for very high risk results. The warnings are held by notifications. The implemented notifications are in the following:

- Please Stay Still - For each heart rate measurement, it is expected to be



(a) **Screenshot of Adjust BLE Status View as BLE turned off.**

(b) **Screenshot of Adjust BLE Status View as BLE turned on.**

(c) **Screenshot of Adjust BLE Status View as connected to device.**

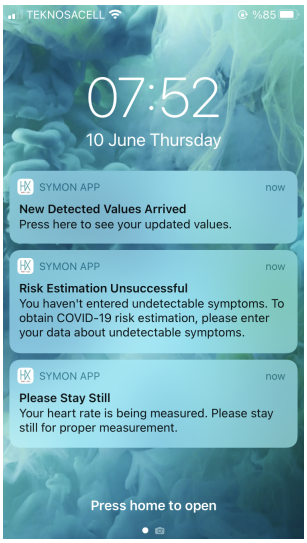
Figure 3.41: Screenshots of Adjust BLE Status Views for different possibilities.

stayed still. Therefore, this notification is sent to warn the user that the heart rate measurement is about to start.

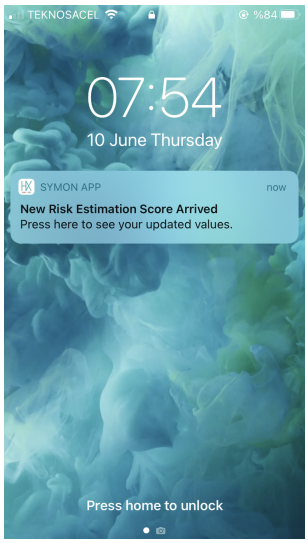
- New Risk Estimation Score Arrived - Whenever up-to-date estimated risk scores are received, this notification is sent to the user.
- New Detected Values Arrived - Whenever up-to-date detectable symptoms data are received, this notification is sent to the user.
- Risk Estimation Unsuccessful - Risk estimation can be failed when the undetectable symptoms are not logged. Then, this notification is sent to the user to warn that risk estimation cannot be done.

3.6.3 Tests and Results

The test is conducted with the usage of Debug Navigator in XCode, which is an IDE developed by Apple for software development [28]. Mobile application is uploaded to an iPhone 7 mobile device and it is connected to a MacBook Pro with an USB cable while running the app from XCode program to make testings. The program automatically tracks the CPU and memory usage in the Debug Navigator. During the testing procedure, the



(a) **Screenshot of Notifications of "Please Stay Still", "New Detected Values Arrived" and "Risk Estimation Unsuccessful".**



(b) **Screenshot of Notification of "New Risk Estimation Score Arrived".**

Figure 3.42: Screenshot of all Notifications.

application is constantly being used and different views and functionalities were used to track every state of the mobile application.

From Figure 3.43, it is seen that CPU usage never exceeds 29% anytime which is less than the specified limit, 30% as stated in Section 3.2.5. Also, it should be noted that the average value is approximately 10% CPU usage, which is very low considering the functionality of the app.

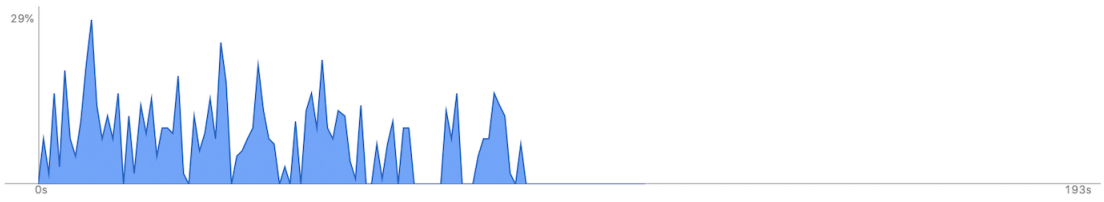


Figure 3.43: CPU usage over time during the usage of the designed mobile application.

From Figure 3.44, it can be seen that memory usage never exceeds 26 MB anytime during the execution of the application. This is much less than the specified limit, 50 MB as stated in Section 2.2.5. This results in very fast execution of the mobile application as there were no flaws or waiting time during the testing process.

The final test was done on the size of the application as the storage is one of the specified requirement. From Table 3.10, the impacts of the mobile

Figure 3.44: Memory usage over time during the usage of the designed mobile application.

Table 3.10: Storage Size of the Designed Mobile Application, SYMON App

App Size	**6 MB**
Documents & Data Size	**557 KB**

application on storage of the mobile device is very low due to the fact that it only takes 6 MB.

To conclude, the test process demonstrates that the designed mobile application has high performance with its CPU, memory usage and storage size such that it can have low battery energy impacts and fast utilization for efficient user experience.

3.7 Risk Estimation Subsystem

The Risk Estimation subsystem is designed to assess the risk of the user having infected with SARS-CoV-2 based on the presence of detectable and undetectable symptoms. The subsystem is implemented on ESP32 and it operates using the internal flash memory of the ESP32 and BLE connection with mobile application. The design process and the working principles of the system are explained in the following parts.

3.7.1 Overall Subsystem

The system's ultimate function is merging the symptom data and assessing a risk of having the virus. For detectable symptoms, a 24-hour circular log in the flash memory of the ESP32 has been created. The risk estimation system reads the 16-hour data from this log and maps the symptom values

to a number between 0 and 3 indicating the severity of the symptom. During the training of the classifier (Random Forest Classifier) both detectable and undetectable symptom intensities are represented on the same scale. The training data is gathered from created surveys; where, for the ease of participants, the intensity scale is fixed between 0 - 3, for both detectable and undetectable symptoms. The risk estimation system acquires undetectable symptom data from mobile application through BLE connection.

The subsystem periodically estimates the risk every 16-hours by default. As an output, the systems gives a "COVID-19 Label", a Boolean indicating having the virus or not, and a "Confidence Score", a floating point number indicating how confident is the model of the outcome. These two values are sent to mobile application through BLE . However, if the undetectable symptoms are not present, meaning that the user has never logged an undetectable symptom data before, the system gives an error message which will be sent to mobile application through BLE. The user is then asked to log undetectable symptom data through mobile application. The functional block diagram of the subsystem can be seen in Figure 3.45.

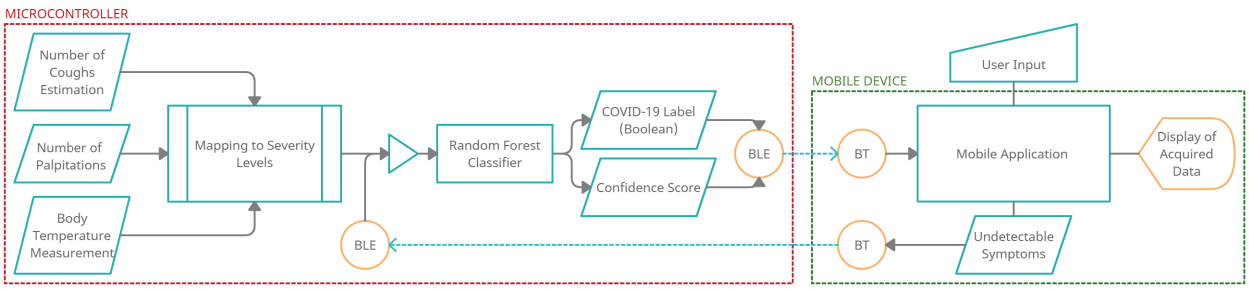


Figure 3.45: The functional block diagram of the risk estimation subsystem.

3.7.2 Design Process and Tests & Results

Acquiring Data for Model Training

Since there were not any data sets which included the severity levels of the symptoms as well as the test results of the subjects, a new data set is created. In order to do that, two surveys are conducted, one in English and other in Turkish. The participation to both surveys in total was 710, consisting of 662 Turkish speaking and 48 English speaking participants. See Appendix A.1.1 for the surveys in both languages. In the surveys, the participants were

asked to scale the severity of 12 symptoms of COVID-19 which are combined from [29] and [30].

The surveys included the questions inquiring if the subjects have been tested or not and their test result if applicable. 380 of the participants were not tested. The symptom data belonging to those who did not tested were not utilizable for the training or testing of the classifier model since it did not indicate any label regarding the virus. Hence, only 303 samples of which 187 had negative labels and 116 had positive labels, were eligible for the training and testing of the classifier model.

Preprocessing of Data

The severity levels indicated in the surveys are None, Low/Few, Mild and Severe/High. These severity levels are mapped to 0,1,2 and 3, respectively. The negative resulted tests are labeled as 0 while positive resulted tests labeled as 1.

The Turkish speaking participants were not required to fill all the symptoms to submit the form. This created blank rows and blank cells in the data. Fully blank submissions were removed completely, while blank cells are set to 0, assuming the corresponding symptom severity as none. Upon deletion of the blank rows, the overall sample count has deduced to 284.

Before feeding the data to the model, the correlation of the symptoms with the COVID-19 test result label is inspected in order to make sure that all the symptoms are correlated and effective in the training. The resulting correlation heat-map can be seen in Figure 3.46. As can be seen from the figure, Diarrhea has correlation with the label (COVID-19 test result) equal to 0.0098. Correlation is negligibly small when compared with the correlation of the other symptoms with the label. Therefore, it was decided to exclude Diarrhea symptom from the data set since it only added up the complexity of the system while not contributing to the decision of the label. The final data set can be observed in Appendix A.2.

Algorithm Selection & Tests

For the risk assessment of the user having the virus, four algorithms were compared according to their compatibility with an embedded system, their receiver operating characteristic (ROC) curves and other metrics. Considered classification algorithms are Logistic Regression (LR), Random Forest

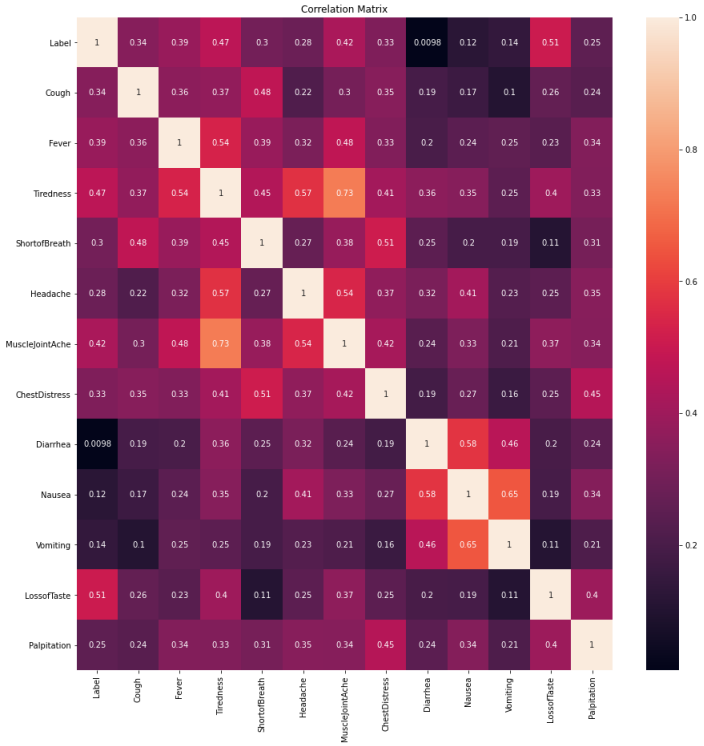


Figure 3.46: The correlation heat-map of symptoms and labels in the data set.

(RF), Support Vector Machine (SVM) which are created with scikit-learn library and Neural Network (NN) model created with Keras.

For the training of all the models, the train-test split was done as 80-20 and random state was chosen as 68. As the solver of the LR, 'newton-cg' has chosen with the inverse of the regularization strength parameter, C, equal to 0.1. For RF model, 40 trees were deployed with a maximum depth of the trees as 2. The SVM is created with 'rbf' kernel type and maximum iteration limit being 10000. Lastly, NN classifier model is created by using Keras. The NN network has 2 dense layers and an output layer. The dense layers included 6 neurons and the output layer included one neuron. The activation functions of the dense layers are ReLu and that of the output layer is sigmoid function. Normal distribution is chosen for weight initialization. Optimizer of the NN is Adam and the loss function is Binary Cross-Entropy. The model is trained with a batch size of 10 and 100 epochs.

First, the ROC of the models are inspected. The SVM algorithm does not classify the data strictly by using the probabilities. Therefore, in the ROC curve plot, it is excluded. The obtained ROC curves are present in

Table 3.11: The performance metrics of the classification models.

	LR	**RF**	**SVM**	**NN**
Accuracy	0.807	0.807	0.789	0.825
Precision	0.800	0.778	0.719	0.833
Recall	0.769	0.808	0.885	0.769
F1 Score	0.784	0.792	0.793	0.800

Figure 3.47 with their corresponding areas under the curve (AUC) given in the legend. The ROC curves were very close to each other and did not provide a significant selection measure.

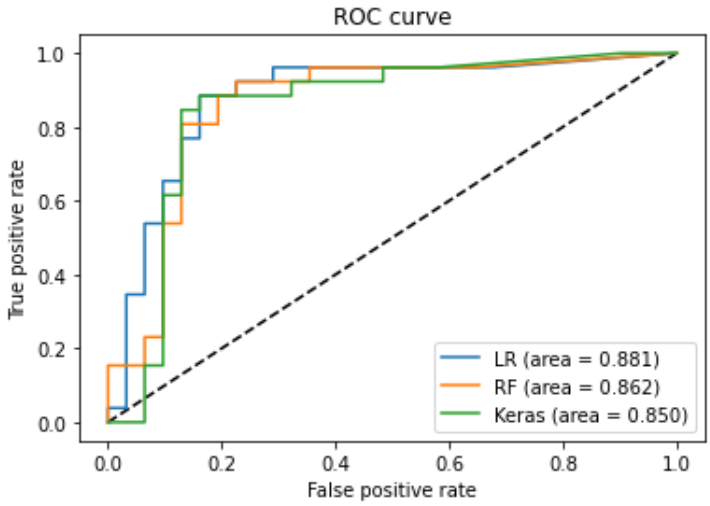


Figure 3.47: The ROC curves for Logistic Regression (LR), Random Forest (RF) and Neural Network (Keras) models with their AUC written in the legend.

Then, models' accuracy, precision, recall and F1 scores are compared. The respective scores are presented in the Table 3.11. As mentioned in Chapter 2, the requirement for this subsystem is to find the presence of COVID-19 with at least %80 accuracy. Also, it is very important to label actual positive values as positive when detecting the vectors of COVID-19. Hence, recall is a very important metric for this case. In terms of recall, most successful model is the SVM with a recall value with 0.885. However, it has the lowest accuracy result with 0.789. The model which has both accuracy and recall higher than %80 is the Random Forest algorithm.

Lastly, the compatibility with an embedded system is compared for these models. It was seen that the most efficient model is the logistic regression

due to its simple nature. However, all of these models can be implemented on the microcontroller efficiently and easily by using C code converter libraries on Python such as "micromlgen".

As a result, due to its compatibility with the embedded systems, high accuracy and recall performance and moderate AUC results, Random Forest algorithm has been chosen to be used as the classifier for COVID-19 risk assessment. By choosing Random Forest as the classifier, the requirement of having an accuracy and sensitivity score higher than %80, as mentioned in Section 2.2.6, is also met.

3.7.3 Implementation

After the classifier model is obtained by the Python implementation and exported to the microcontroller by using micromlgen library, the model is merged with the overall system. In order to feed the model with the obtained symptom data, the processes explained in the following sections are applied to the detectable symptom data while undetectable symptom data received through BLE connection is used as it is since it was already between 0 and 3 and obtained for a day.

Preprocessing of the Data to be Classified

The data obtained from the sensors must be mapped between 0 and 3 since it is the range of severity of the symptom data that the classifier was trained. The three detectable symptoms have different normal and severity ranges. The following parts describe the mappings of these data.

Processing of Temperature Data

The risk estimation subsystem receives the last 16 measurements of temperature and averages them. According to [16], the temperature of the body is considered to be normal in the range 36.1°C - 37.2°C. The averaged temperature values that are equal to or lower than 36.1°C are mapped between 0 and 1 while the values between (36.1°C, 37.2°C] are mapped between 1 and 2, 37.2°C corresponding to 2. Then the values between (37.2°C, 38°C] are considered as a mild fever and they are mapped between 2 and 3. The values higher than 38°C (including 38°C) are mapped to 3. This mapping produces non-integer values as well.

Processing of Cough Data

Cough data is obtained as number of coughs in the last 16 hours from the flash memory of the microcontroller which is logged by the Cough Detection subsystem. This data also should be mapped to the range of 0 and 3. In a population of healthy adults, the geometric mean of the number of coughs in a duration of 24 hours is found to be 18.6 with a standard deviation 3.16 [31]. Hence the total number number of coughs between 0 and 22 in a duration of 16 hours are mapped to a number between 0 and 1 linearly. Also, in the same paper, the geometric mean of number of coughs for a population of people with respiratory diseases is 257 with a standard deviation of 2.34 in 24 hours. Therefore, the number of coughs per day equal to 255 and above are mapped to 3. Since in the literature, there was not any definition for mild and few coughs, the number of coughs between 22 and 257 are mapped between 1 and 3 linearly with the line equation $y = (2 * x + 446)/234$; where y is the value to be sent to classifier and x is the number of coughs.

Processing of Heart Rate Data

Lastly, the palpitation symptom should be considered in order to estimate the risk. The number of palpitations for every hour is kept as a log in the flash memory as explained in Section 3.4.1. The risk estimation subsystems reads the number of palpitations belonging to the last 16 hours. The frequency of the palpitation episodes are rated by an integer between 1 and 7 in [32], where 1 corresponds to once or twice a day and 7 corresponds to sustained. Since SYMON will not be measuring continuously, the 7th rate is not applicable for this case. Therefore, the first 6 cases are adapted (merged by groups of two) for our 0-to-3 rating system. The final severity classes and their corresponding palpitation episode frequencies are as follows,

- Severity Level 1: Once to 5 times in 16 hours.
- Severity Level 2: More than 5 times but less than 32 times in 16 hours.
- Severity Level 3: More than 32 times in 16 hours.

In this mapping, the values are mapped to the integers directly to sustain coherency with the reference paper.

Classification

Once the detectable symptom data is preprocessed and undetectable symptom data are received from mobile application, the symptom values are fed to the model as an array. The output consisting of COVID-19 label along with the confidence score for the resulting label is sent to the mobile application through BLE. In the mobile application, as mentioned in Section 3.6.2, the warning messages are presented to the user.

Chapter 4

Compliance with Requirements

The compliance with the requirements that are set in the Chapter 3 are checked and discussed through the parts of Chapter 4. However, in order to present them in a compact and clear manner, the Table 4.1 is prepared. In this table, the relevant requirements set in Chapter 3 for each subsystem as well as the overall system are shown with their corresponding test values.

Table 4.1: The Table for Subsystems and the Overall System and their Compliance with the Set Requirements

Subsystem	**Requirements**	**Set**	**Achieved**
Cough Detection	Accuracy	>0.7	0.998 (met)
	Precision	>0.7	0.981 (met)
	Sensitivity	>0.7	0.818 (met)
	F1	>0.7	0.892 (met)
	FP rate	<0.001	0.0001(met)
Risk Estimation	Sensitivity	>0.8	0.808 (met)
	Accuracy	>0.8	0.807 (met)
Mobile Application	Maximum CPU Usage	<0.30	0.29 (met)
	Memory Usage	<50 MB	26 MB (met)
	Communication	Two-way	Two-way (met)
Communication	Battery Usage	Low Energy	BLE (met)
Heart Rate Measurement	Location	Chest	Chest (met)
	Absolute Measurement Percentage Error	<0.05	∼0.02 (met)
Fever Detection	Absolute Measurement Error	<0.5 °C	<0.3 °C (met)
Overall	Charge Cycle	>16 hours	64 hours (met)
	Cost	<$80	$ 47 (met)
	Weight	<200 g	160 g (met)
	User Friendliness	User interface & Physical Ergonomics	Intuitive interface & washable, ergonomic cloth (met)

Chapter 5

Resource Management and Impacts

5.1 Cost Management

SYMON is a chest strap consisting of a microcontroller, a microphone, a temperature sensor, a heart rate sensor and a power bank. While deciding which components are used for the implementation of the overall system, candidate components are compared in terms of their compliance with the requirements set for each subsystem and the overall system while considering their costs.

ESP WROOM-32 is chosen as the microcontroller since it has considerably low cost with $ 6.8 with high processing power and low power consumption when it is compared with the other microcontrollers. The comparison can be seen in Table 5.1.

Table 5.1: Comparison Table of Microcontroller

Name	Cost	Power Cons.	Protocols	Memory	Processor	Size & Weight
ESP WROOM 32 [7]	$ 6.8	80 mA for each I/O pin & 3-3.6 V	I^2C & SPI & I^2S	520 KB SRAM & QSPI Flash Memory 4 KB	2.4 GHz frequency & 440 KB ROM & 520 KB SRAM	18.0 mm x 25.5 mm x 3.1 mm & 8 g
Arduino Nano [33]	$ 28	40 mA for each I/O pin & 7-12 V	I^2C & SPI	32 KB & 2 KB Bootloader	ATmega328P 16 MHz SRAM 2KB	18.5 mm x 43.1 mm x & 7 g
Arduino Micro [34]	$ 22	20 mA for each I/O pin & 6-20 V	I^2C & SPI	32 KB & 4 KB Bootloader	ATmega32U 4 & 16 MHz	48 mm x 18 mm x & 13 g
Arduino Mini [35]	$ 2.5	40 mA for each I/O pins & 5-12 V	SPI	32 KB	ATmega32 8 Bit 16 MHz	18 mm x 33 mm x & 2 g

Regarding the requirements of the cough detection subsystem, the MEMS microphone INMP441 is chosen as the best option with its low cost, $ 6.5, besides it has a low power usage and is compatibile with I^2S protocol. The

comparison of INMP441 and other candidate microphones is shown in Table 5.2.

Table 5.2: Comparison Table of Microphones

Microphone	Cost	Power Consumption	Protocol	SNR	Availability in Turkey
INMP441 [6]	$ 6.5	250 µA & 3.3 - 5 V	I^2S	70 dB	Available
LM 393 [36]	$ 0.8	4-5 mA & 3.3-5 V	Digital & Analog Output	54 dB	Available
SPH0645LM4H [37]	$ 7.8	600 µA & 3 V	I^2C	65 dB	Available
smMIC - ICS43434 [38]	$13.95	490µA & 1.8V	I^2S	64 dB	Not Available

In order to obtain temperature data as accurate as possible with a low cost, SHT-31 temperature sensor is chosen. It has a high accuracy of ± 0.3 °C and a low price of $ 10. Some of the features of SHT-31 and other sensors are compared in Table 5.3.

Table 5.3: Comparison Table of Temperature Sensors

Sensor	Cost	Accuracy	Power Cons.	Data Type & Protocols	Availability in Turkey
SHT 31 [15]	$ 10	± 0.3 °C	800 μA & 2.4-5.5 V	Digital & I^2C	Available
SHT 35 [15]	$ 23	± 0.2 °C	800 μA & 2.4-5.5 V	Digital & I^2C	Available
MCP9808 [39]	$ 5	± 0.5 °C	200 μA & 2.7-5.5 V	Digital & I^2C	Available
LM35 [40]	$ 2	± 0.5 °C	60 μA & 4-30 V	Analog	Available
TEMP 117 [41]	$ 25	± 0.1 °C	3.5 μA & 1.8-5 V	Digital & I^2C	Not Available
MAX30208 [42]	$ 1.31*	± 0.1 °C	1.7 μA & 1.7-3.6 V	Digital & I^2C	Not Available

To supply power to the system, considering its low cost with $ 7.5 and high power capacity with 4400 mAh as tabulated in Table 5.4, Trust is preferred over the other candidates and the previous power bank which was chosen at Conceptual Design Report [9].

Regarding its fitness to human body, analog ECG sensor AD8232 is cho-

Table 5.4: Comparison Table of Power Banks

Power Bank	Cost	Capacity (mAh)	Size	Safety
Duracell [43]	$ 10	3350	28x104x197 mm	Yes
Trust [44]	$ 7.5	4400	90x22x47 mm	Yes

Table 5.5: Total Cost Table

Components	ESP WROOM-32	INMP 441	SHT 31	AD8232	Power Bank	Additional Costs	TOTAL
Cost	$ 6.8	$ 6.5	$ 10	$ 11.2	$ 7.5	$ 5	$ 47

sen to obtain accurate pulse measurement with cost of $11.2 other than its counter parts.

After removing the respiratory measurement subsystem, cost of 2 accelerometers, which are LIS3DSH with cost $ 7.52, is excluded from the total cost. At the end, total cost of the components is calculated $ 47 as in Table 5.5.

5.2 Impacts and Discussions

Safety and Precautions

SYMON contains sensitive electronic components and can be damaged if dropped, burned, punctured, or crushed. Hence, the device should be handled carefully. If one of the components is damaged, such as bare wire or a cracked box, the wearer must not use the device. Tens pads may lose their stickiness at high temperatures. Also, the wearer should not attempt to open and repair SYMON. It may both damage the device and user may get injured. For health and safety, device should be kept dry. Furthermore, the device must not be used if the individual has a cardiac pacemaker due to possible electromagnetic interference caused by BLE and/or ECG sensor.

The wearer also must consider that SYMON should not be used as a medical device. It is not designed to diagnose disease, it can be only used for self-monitoring. It cannot substitute professional medical testing. It is required to consult a medical organization to get a confirmed diagnosis.

Impacts on Society

Widespread usage of SYMON is expected to deliver a significant impact on the regulation of the pandemic. Primarily, it is aimed to detect the symptoms of COVID-19, hence the user will be able to detect the health condition at an early stage. It will reduce the transmission rate of the pandemic. Also, it will raise awareness to such health issues and a prominent effect of the electronic device. Therefore, they can improve the technology to detect more symptoms or even to monitor other contagious diseases.

Impacts on Environment

The overall device is made of various materials that include the following: electronic components, copper, plastic, steel, and fabric. When the lifetime of the device is expired, such materials can cause pollution to the environment, especially in large amounts. Hence, HAI-CO. positions itself as a sustainable company and offers recycling services. The users are motivated to send their old devices back to us and get a discount towards the purchase of the new devices. As a result, it is aimed to prevent any pollution that may be caused by materials of product SYMON.

Chapter 6

Deliverables

Chest Strap

A comfortable, lightweight and washable chest strap, which will be worn under clothing in order to place the electronic box and the power bank to the body of the user.

A Box Including Electronics

An electronic enclosure box made of plastic, consisting of HR (heart-rate) module and microcontroller. As necessary attachments; microphone and microphone clipper to mount it beneath collars, temperature sensor and an elastic band to sustain it under the armpit. All components are supplied with cables that have appropriate lengths.

Power Bank

A power bank as a power supply of the overall system with a USB to Micro-USB cable for connecting the power bank to the microcontroller.

Tens Pads

3 tens pads to be mounted on the body of the user to measure heart rate.

SYMON App From App Store

A mobile application that enables the user interface, fed with user inputs for undetectable symptoms and displaying the COVID-19 risk assessment for the user.

A User Manual

A user manual describing how to assemble the hardware and software parts of the product is available as a link under Appendix A.4.

Services

The HAI-CO. Authorised Service Provider (HASP) program is designed for providing users technical support and warranty services. Customer support is intended to work 24/7. The service performs maintenance and repairs

in case of hardware and software errors of the device. The legal guarantee is valid for two years and covers products bought anywhere in Turkey and Russian Federation.

Chapter 7

Conclusion

From education to economics, politics to social life, COVID-19 has been changing and shaping the world faster than ever. Since its first confirmed case on 31 December 2019, it is estimated that 10% of the world's population is infected with COVID-19. Even with the global vaccination attempts, the disease persists where regulation of its infection is still a global emergency. With ever increasing number of infected individuals occupancy of medical centres are at alarming levels.

As supplementary application to vaccination, self-monitoring and taking precautions immediately in case of infection, can be a viable solution to regulate the spread of the virus. Thus, effectively keeping the occupancy of hospitals under control

To contribute to the effort of regulating the rapid spread of the virus, team of HAI-CO. has designed a wearable symptom monitoring device, SYMON

SYMON, as a chest strap, is designed to detect major symptoms of COVID-19 that are fever, cough and palpitation. It also enables users to log their undetectable symptoms through a mobile application and these symptoms include tiredness, shortness of breath, headache, muscle and joint ache, chest distress, nausea, vomiting, loss of taste. By using this symptom data, SYMON estimates the user's risk of having SARS-CoV-2 virus and provides relevant precautionary advice to the user if needed.

In this report, development and overall design process of SYMON is explained. First, the team of HAI-CO. is introduced, then the requirements that overall system and subsystems need to achieve are discussed. The design process of the system and subsystem implementations are examined and test results are presented. Their compatibility with the requirements are inspected and the product is analyzed in terms of its potential environmental and societal impacts and its safety. The resource management for the project is explained and followed by the deliverables. All the requirements that are set in the beginning of the project are shown to be set in the final product.

For cough subsystem, the set requirements were 0.7 accuracy, precision, recall, F1 score and ratio of misclassified non-cough sounds on the order less

than 0.001. With the implemented cough detection subsystem, comprehensive tests with a total duration of 4643 seconds (1 hour 17 minutes) with 132 cough groups and 316 coughs on different dB levels are conducted and all the requirements are shown to be met with 0.993 accuracy, 0.999 specificity, 0.981 precision, 0.818 recall, 0.892 F1 score and ratio of 0.0001 misclassified non-cough sounds. Furthermore, the user specific cough calibration system has been implemented successfully.

The requirements set for the fever subsystem was at least ± 0.5 °C accuracy and low power usage. In the tests conducted on the implemented fever detection subsystem, ± 0.3 °C accuracy is obtained. In the previous report it was shown that the temperature measurement system individually dissipates 40 μA for each digital measurement. Therefore, both requirements are satisfied for fever detection subsystem.

Heart-rate subsystem is decided to have an approximation error at most 5% as a requirement in Chapter 2. In the tests, the subsystem achieved 2.82% approximation error. With this, it met the set requirement.

Communication subsystem is required to transfer the data without any loss and the implementation should have been realized with low price. Since the subsystem is implemented using BLE module of the microcontroller, the subsystem is completed without any extra price and showed in the tests that the data is transferred successfully.

For mobile app subsystem, the requirements were a convenient visual user interface, functionality as BLE connection and persistence of the data for 2 weeks, high performance with less CPU and memory usage and less storage size for better user experience. For these purposes, the app is designed and developed only for iOS platform to increase its performance. All functionalities including BLE connection, persistence of data and displaying the symptoms are implemented. Many visual tools are used inside the app in order to contribute the users to understand the logic behind the app intuitively. Approximately 10% CPU usage, less than 30 MB memory usage and 3.4 MB app storage size are obtained such that robust, fast and low power mobile application is obtained.

Lastly, for the risk estimation, 80% accuracy and sensitivity scores were the most important requirements. By implementing a Random Forest classifier on the microcontroller, the 80% accuracy and sensitivity is met. Another requirement was to provide the users precaution messages according to their

COVID-19 infection scores. The relevant messages for respective risk levels are prepared.

You: All in all, within the planned time interval, a product that can meet the needs of the customers in terms of self-monitoring for COVID-19 is designed and implemented by the team of HAI-CO. The device is shown to comply with the requirements at a reasonable price and is expected to contribute to well-being of the society significantly.

Appendices

Appendix A

A.1 Links for COVID-19 Surveys

A.1.1 Turkish Survey Link

https://forms.gle/tGxVF83QrVX1yGCx5

A.1.2 English Survey Link

https://forms.gle/rribTGTzSp8XELdr5

A.2 The Data Set for Risk Estimation Algorithm

https://docs.google.com/spreadsheets/d/1vTM2eVkwbstcyHJsdnxDtNKaDlzc0KcrifwgDosUWgs/edit?usp=sharing

A.3 Cough Detection Test Recordings

https://drive.google.com/file/d/1ycn0$_a kc$ – $zZkt5UAdc8ysz$ – $d51iGhAGx/view?usp = sharing$

A.4 Link For The User Manual

https://drive.google.com/file/d/1diBaOfrsTGHe0V5jbBKLoLN-Xwubthtg/view?usp=sharing

References

[1] "Covid-19 coronavirus pandemic," Jan. 8, 2021. [Online]. Available: https://www.worldometers.info/coronavirus/. [Accessed: June 9, 2021].

[2] "Coronavirus," 2021 [Online]. Available: https://www.who.int/health-topics/coronavirus#tab=tab_3.. [Accessed: Jan. 8, 2021].

[3] S. Mert, O. Oral, M. Marchenkova, U. Sedef, and K. Okumuş, "Symon: Symptom monitor of covid-19: Project proposal report," Nov. 20, 2020. [Online]. Available: https://drive.google.com/file/d/10FPlZtmy8OkXuzqE0LnTWt9Vy7Bv6-mT/view?usp=sharing. [Accessed: Apr. 5, 2021].

[4] "Jt 830ln dijital multimetre Ölçü aleti - class," [Online]. Available: https://www.direnc.net/jt-830ln-dijital-multimetre-olcu-aleti-class. [Accessed: June 10, 2021].

[5] T. Elfaramawy, C. Latyr Fall, M. Morissette, F. Lellouche, and B. Gosselin, "Wireless respiratory monitoring and coughing detection using a wearable patch sensor network," in *2017 15th IEEE International New Circuits and Systems Conference (NEWCAS)*, 2017, pp. 197–200.

[6] "Datasheet inmp441," May, 2015 [Online]. Available: https://invensense.tdk.com/wp-content/uploads/2015/02/INMP441.pdf. [Accessed: Jan. 8, 2021].

[7] "Datasheet esp32wroom32," 2020 [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf. [Accessed: Jan. 8, 2021].

[8] Y. Shi, H. Liu, Y. Wang, M. Cai, and W. Xu, "Theory and application of audio-based assessment of cough," *Journal of Sensors*, vol. 2018, pp. 1–10, 03 2018.

[9] S. Mert, O. Oral, M. Marchenkova, U. Sedef, and K. Okumuş, "Symon: Symptom monitor of covid-19: Conceptual design report," Jan. 8, 2021. [Online]. Available: https://drive.google.com/file/d/1s0JAfzfVyeX2LfnWe9cNZV_5DLn8IGqO/view?usp=sharing. [Accessed: Apr. 5, 2021].

[10] R. X. A. Pramono, S. A. Imtiaz, and E. Rodriguez-Villegas, "A cough-based algorithm for automatic diagnosis of pertussis,"

PLOS ONE, vol. 11, pp. 1–20, 09 2016. [Online]. Available: https://doi.org/10.1371/journal.pone.0162128

[11] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2544–2550.

[12] R. X. Adhi Pramono, S. Anas Imtiaz, and E. Rodriguez-Villegas, "Automatic identification of cough events from acoustic signals," in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2019, pp. 217–220.

[13] J. Gallagher, "What noises cause hearing loss? — nceh — cdc," Nov. 14, 2020. [Online]. Available: https://www.cdc.gov/nceh/hearing_loss/what_noises_cause_hearing_loss.html. [Accessed: Nov. 16, 2020].

[14] "Abc apps team: Sound meter v.3.6.3," Available: https://play.google.com/store/apps/details?id=com.gamebasic.decibel&hl=en&gl=US., 2021.

[15] "Sht 3x datasheet," May, 2015. [Online]. Available:https://www.farnell.com/datasheets/2000035.pdf. [Accessed: Apr. 8, 2021].

[16] L. J. Vorvick, "Body temperature norms," July 2, 2019. [Online]. Available: https://medlineplus.gov/ency/article/001982.htm. [Accessed: Apr. 5, 2021].

[17] "Long-term effects of covid-19," Nov. 13, 2020. [Online]. Available: https://www.cdc.gov/coronavirus/2019-ncov/long-term-effects.html. [Accessed: Apr. 8, 2021].

[18] J. Horton, P. Stergiou, T. Fung, and L. Katz, "Comparison of polar m600 optical heart rate and ecg heart rate during exercise," *Medicine Science in Sports Exercise*, vol. 49, pp. 2600 – 2607, 12 2017.

[19] "Datasheet ad8232," [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/ad8232.pdf. [Accessed: Jan. 8, 2021].

[20] R. Avram, G. H. Tison, K. Aschbacher, P. Kuhar, E. Vittinghoff, M. Butzner, R. Runge, N. Wu, M. J. Pletcher, G. M. Marcus *et al.*, "Real-world heart rate norms in the health eheart study," *NPJ digital medicine*, vol. 2, no. 1, pp. 1–10, 2019.

[21] "Comparison of ble and wifi," Jun. 29, 2019. [Online]. Available:https://www.netguru.com/codestories/bluetooth-vs-wifi-comparison-for-the-iot-solutions. [Accessed: Apr. 8, 2021].

[22] S. Mert, O. Oral, M. Marchenkova, U. Sedef, and K. Okumuş, "Symon: Symptom monitor of covid-19: Critical design review report," Apr. 9, 2021. [Online]. Available: https://drive.google.com/file/d/1mTQzR4VcA0cdtBPGRUZx8kNw9ud3M6Ij/view?usp=sharing. [Accessed: Jun. 11, 2021].

[23] P. Grzmil, M. Skublewska-Paszkowska, E. Łukasik, and J. Smołka, "Performance analysis of native and cross-platform mobile applications," *Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska*, vol. 7, 2017.

[24] "Mobile operating system market share worldwide," [Online]. Available:https://gs.statcounter.com/os-market-share/mobile/worldwide. [Accessed: Apr. 8, 2021].

[25] "Mobile operating system market share worldwide," [Online]. Available:https://developer.apple.com/xcode/swiftui/. [Accessed: Apr. 8, 2021].

[26] "Core data," [Online]. Available:https://developer.apple.com/documentation/coredata. [Accessed: Apr. 8, 2021].

[27] "Core bluetooth," [Online]. Available:https://developer.apple.com/documentation/corebluetooth. [Accessed: Apr. 8, 2021].

[28] "Examining system impact," [Online]. Available:https://developer.apple.com/library/archive/documentation/ToolsLanguages/Conceptual/Xcode_Overview/ExaminingSystemImpact.html#//apple_ref/doc/uid/TP40010215-CH59-SW1. [Accessed: Apr. 8, 2021].

[29] J. Li, Z. Chen, Y. Nie, Y. Ma, Q. Guo, and X. Dai, "Identification of symptoms prognostic of covid-19 severity: Multivariate data analysis of a case series in henan province," *J Med Internet Res*, vol. 22, no. 6, p. e19636, Jun 2020. [Online]. Available: https://www.jmir.org/2020/6/e19636

[30] Y. Luo, J. Wu, J. Lu, X. Xu, W. Long, G. Yan, M. Tang, L. Zou, D. Xu, P. Zhuo, Q. Si, and X. Zheng, "Investigation of covid-19-related symp-

toms based on factor analysis," *Annals of Palliative Medicine*, vol. 9, pp. 36–36, 06 2020.

[31] N. Yousaf, W. Monteiro, S. Matos, S. S. Birring, and I. D. Pavord, "Cough frequency in health and disease," *European Respiratory Journal*, vol. 41, no. 1, pp. 241–243, 2013. [Online]. Available: https://erj.ersjournals.com/content/41/1/241

[32] B. Huang, H. Yan, L. Hu, G. Wang, J. Meng, W. Li, G. Liu, J. Wang, W. Le, and H. Jiang, "The contribution of psychological distress to resting palpitations in patients recovering from severe coronavirus disease 2019," sep 2020. [Online]. Available: https://doi.org/10.22541%2Fau.160029826.67826939

[33] "Arduino nano datasheet," [Online]. Available:https://www.arduino.cc/en/uploads/Main/ArduinoNanoManual23.pdf. [Accessed: Apr. 8, 2021].

[34] "Arduino micro datasheet," [Online]. Available:http://www.farnell.com/datasheets/1685581.pdf. [Accessed: Apr. 8, 2021].

[35] "Arduino mini datasheet," Sep. 4, 2014. [Online]. Available:https://www.arduino.cc/en/uploads/Main/Arduino-Pro-Mini-schematic.pdf. [Accessed: Apr. 8, 2021].

[36] "Datasheet lm393," Available: https://components101.com/asset/sites/default/files/component_datasheet/Sound-Detection-Sensor-Datasheet.pdf. [Accessed: Apr. 5, 2021].

[37] "Datasheet sph0645lm4h," Available: https://cdn-shop.adafruit.com/product-files/3421/i2S+Datasheet.PDF. [Accessed: Apr. 5, 2021].

[38] "Datasheet smmic - ics43434," Available: https://sensormaestros.com/wp-content/uploads/smMic-ICS43434-DS.pdf. [Accessed: Apr. 5, 2021].

[39] "Mcp9808 datasheet," 2011. [Online]. Available:https://ww1.microchip.com/downloads/en/DeviceDoc/25095A.pdf. [Accessed: Apr. 8, 2021].

[40] "Lm35 datasheet," December,2017. [Online]. Available:https://www.ti.com/lit/ds/symlink/lm35.pdf. [Accessed: Apr. 8, 2021].

[41] "Temp117 datasheet," March ,2019. [Online]. Available:https://www.ti.com/lit/ds/symlink/tmp117.pdf?ts=1617781243889&

ref_url=https%253A%252F%252Fwww.google.com%252F. [Accessed: Apr. 8, 2021].

[42] "Max30208 datasheet," May ,2020. [Online]. Available:https://datasheets.maximintegrated.com/en/ds/MAX30208.pdf. [Accessed: Apr. 8, 2021].

[43] "Duracell features," [Online]. Available:https://www.duracell.com.au/product/duracell-powerbank-3350-mah/. [Accessed: Apr. 8, 2021].

[44] "Trust features," [Online]. Available:https://www.trust.com/en/product/21224-primo-powerbank-4400-mah-black. [Accessed: Apr. 8, 2021].