The University of Texas at Tyler
Department of Electrical Engineering
College of Engineering
Tyler, TX 75799

Primary Design Document
For
An IoT-Based Edge-Intelligent Wearable Sensor Array

A design project to fulfill the requirements of Senior Design in the Departments of Electrical Engineering and The University of Texas at Tyler

The individuals whose names and signatures appear below certify that the narrative, diagrams, figures, tables, calculations, and analyses contained within this document are their original work except as otherwise cited.

Cody Conder	______________________	
	Signature	Date
Lloyd Mcgrath	______________________	
	Signature	Date
Martin Morales Alverez	______________________	
	Signature	Date
Sloke Shrestha	______________________	
	Signature	Date
Timothy Bauer	______________________	
	Signature	Date

Executive Summary

Current wearable sensor arrays require the use of a smartphone to access WiFi and send the data out for cloud processing. While this implementation is adequate for most users, certain demographics are locked out of the benefit of being able to monitor the user's health condition. Additionally, the use of cloud processing means the data has to be processed on an external server. With demand for cloud computing, this proposed system intends to remove the need for a smartphone from the health monitoring framework. The intended benefit of this system is to allow larger access to health monitoring technology and free the framework from reliance on cloud computing. This proposed system is constructed as two separate modules: a processing module on the waist and a data collection module on the wrist. This system also is intended to be scalable as more sensor modules can be added based on user needs. The final design of this system uses the TinyScreen+ on the wrist module for both data collection and display. This TinyScreen+ is attached to an accelerometer and a pulse oximeter. The two vital biological signals collected are heart rate and body motion. The data will then be transferred to the waistband module using Bluetooth to a Teensy 4.1. This Teensy 4.1 processes the data and utilizes machine learning to detect stress in the data.

Table of Contents

List of Figures

List of Tables

1. Project Description

Mobile health innovations are gaining popularity due to innovations in smartphone technology. Smaller, more powerful smartphones are increasingly available to the average consumer at a decreasing cost. The improvement in processing power these smartphones provide counteracts the data processing bottlenecks that might occur in older IoT implementations with limited local processing capability. This has resulted in advancement in smartphone-based health monitoring algorithms[1-4].

Despite the availability of these powerful smartphones and smartphone-based health monitoring algorithms, some are still unable to access this technology. Examples include patients with disabilities, attention-deficit/ hyperactivity disorder (ADHD), or substance use disorder [5]. Therefore, the need for a mobile health monitoring framework which does not rely on a smartphone is evident. The design of a mobile health monitoring framework that can process monitored data and alert the user of abnormalities, without reliance on smartphones, is proposed to fulfill this need.

The project goal is to design a proof-of-concept to achieve a tradeoff between a loss of processing power present in smartphones and a reduction in barriers preventing inclusion in mobile health monitoring systems. Specifically, for the latter, an analysis of biosensors and the means to transmit the data they collect is yet to be done. The feasibility of processing the data, the identification of potential physiological ailments, and then the notification of said ailments to the user will be reviewed in this project. In order to achieve this, the scope of this project includes the measurement of body movement and the heart rate to measure stress as the physical ailment. Technological and price constraints are also studied to measure the utility of the proposed alternative framework.

2. Target Specifications and Ethical Considerations

The designed framework shall aim to address the problems stated in Chapter 1. Additionally, ethical and professional considerations, such as those listed in Table 2.1, must be considered during the design process.

- **Hardware Technical Specifications**
    - The designed framework shall be wearable.
    - The framework device(s) shall be capable of wireless communication.
    - Subsystem(s) within the framework shall draw less than 500 mA.
    - The framework shall be designed to allow for continuous monitoring.
        - Device charging shall not interrupt monitoring.
    - The framework shall monitor motion and heart rate of the user.
    - A chassis shall be designed to house device components.
    - The framework's design shall not include materials which cause skin irritation for the user.
- **Software Technical Specifications**
    - The framework shall use a machine learning (ML) algorithm to detect stress in the user.
        - The ML classifier shall output binary classification for stress detected.
        - The ML classifier shall input motion and heart rate of the user.

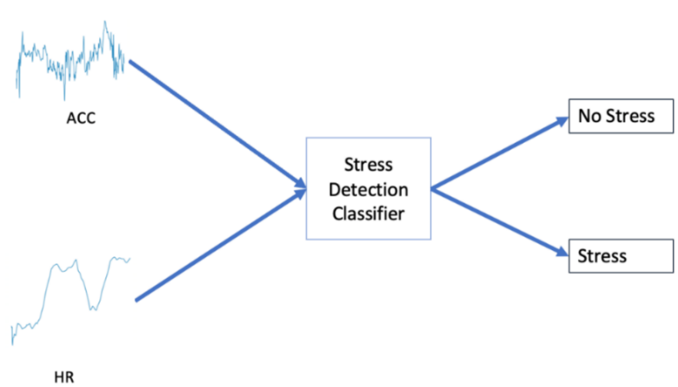


Figure 2.1: The stress detection classifier inputs a window of accelerometer and heart rate data and classifies the signal as stress or no stress conditions.

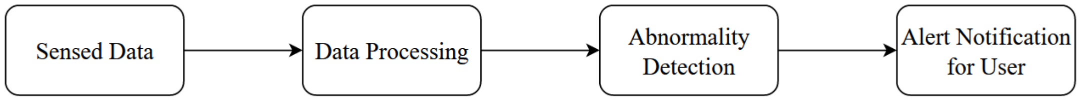


Figure 2.2: A basic view of the framework highlighting key stages. Machine learning algorithms will be used to process data and detect abnormalities.

Table 2.1: Ethical and Professional Considerations.

	Ethical and Professional Considerations
Public health	The designed wearable monitoring system is not intended for treating specific health problems. Wearable health monitoring systems may provide a false sense of safety or anxiety. Biomarker data should not guide treatment without the advice from a healthcare provider. Team members shall promptly communicate and strive to resolve any design concerns which threaten public health.
Safety and welfare	Data obtained from a wearable IoT monitoring device is subject to cybersecurity threats. Some identifiable information can be obtained and used for nefarious purposes which may include risking the welfare of patients. Team members shall strive to protect the privacy of others by complying with ethical design practices.
Global factors	Materials for electronic devices are often produced in countries where unethical labor laws are in place. Team members shall strive to obtain material components from ethically responsible sources.
Societal factors	Widespread use of continuous monitoring wearable devices may result in the normalization of monitoring of the human body by third party entities [6]. As more wearable monitoring devices are used in society, users should be informed as to who has access to the data generated by a worn wearable monitoring device. Team members shall strive to inform individuals using the designed wearable device of what entities have appointed access to the data obtained from the device.
Environmental factors	The wearable device shall incorporate a rechargeable LiPo battery. This type of rechargeable battery contains harmful chemicals that are toxic to ecosystems and the creatures which reside in them. Team members shall properly dispose of batteries which are determined to be unsuitable for the project's design.
Economic factors	Individuals with a cognitive impairment or without constant access to a smartphone are not able to utilize wearable devices which require the processing power of an expensive smartphone. Team members shall design the system to not require a smartphone to process the monitored data. This improves accessibility of the device to those of poorer socioeconomic backgrounds.

3. Engineering Standards and Constraints

Engineering constraints and standards exist to prevent injury and damage to people and property. This proposed project is not immune to abiding by engineering constraints and standards. Below are tables 3.1 and 3.2 list standards and constraints that the project will have to take into consideration while the design process takes place.

Table 3.1: Engineering Standards

Organization	Engineering Standards
IEEE	Wearable monitoring systems require wireless communications should data transmission be necessary. Bluetooth and Wi-Fi, as defined by the IEEE 802.15 and IEEE 802.11 standards, shall be used to handle transmissions if needed.
Safety	The device(s) in the framework shall be designed in a way to minimize risk of choking hazards and skin irritation. Further, the array shall comply with required surface temperature limits set out by regional, national, and international regulations and safety standards. In the U.S. and internationally, the International Electrotechnical Commission standards are used. The current guide for surface temperature limits is set by the IEC's 60950-1 (2005).
Government	The framework shall meet all standards set by the U.S. Consumer Product Safety Commission, applicable European regulations, and other international standards. Additionally, the framework shall adhere to all regulations for wireless medical devices set by the Food and Drug Administration's Center for Devices and Radiological Health (CDRH) [7] due to its status as a class two device.
International	The designed framework could potentially require the 3G, 4G LTE, and CAT – M1 mobile telecommunications networks in order to transmit health data as necessary. In the event it is necessary, interoperability with the 3G, 4G/4G LTE, and CAT – M1 networks is made possible with standards IMT-2000, IMT-Advanced, and IMT-2020, respectively. The International Mobile Telecommunications (IMT) standards are issued by the International Telecommunication Union.

Table 3.2: Engineering Constraints

	Technical Description
Cost	The allocated budget for the project is $1000. Prototypes range in cost depending on the type of components used for construction. Thus, efforts shall be made to select components for the framework in a cost-effective matter to produce a prototype below the project budget.
Speed	The time spent to monitor, process, and alert users of a wearable monitoring system is crucial. This total time accounts for transmission and processing speeds. Due to the use of a machine learning algorithm, transmission speed is left as the most optimizable constraint. As such, only high-speed wireless communication protocols shall be used.
Reliability	Wearable devices are subject to wear and tear. Moreover, wearable items, like all wireless devices in an IoT framework are inherently susceptible to packet-loss from their wireless communications. These aspects of the framework shall be factored into design.
Power	The device will be a wearable device that has a power supply not needing a connection to an outlet. The typical power banks for wearable devices are 3.7V with a range from 100mAh to 500mAh for size. This device will run a minimum of 24 hours before needing to recharge.
Complexity	The framework consists of biometric signals such as movement signals and heart rate signals. The framework shall wirelessly transmit a notification to the user about any adverse events if/when such an event occurs. Programming and machine learning algorithms shall be used to process data and detect adverse health events.
Manufacturability	The framework shall utilize designs using only off-the-shelf components due to unforeseen lead-times impacting the semiconductor industry. Printable Circuit Board (PCB) designs shall not be considered due to similar lead-time issues and added complexity in terms of fabrication.
Sustainability	Energy use of electronics is an important factor to consider for sustainable energy consumption. This can be achieved by striving towards a low power design.

4. Evaluation of Design Alternatives

Figure 4.1 below shows the current system overview. The goal of this section is to expand this rough overview into a specific system overview. Design alternatives were considered at framework level and subsystem level.

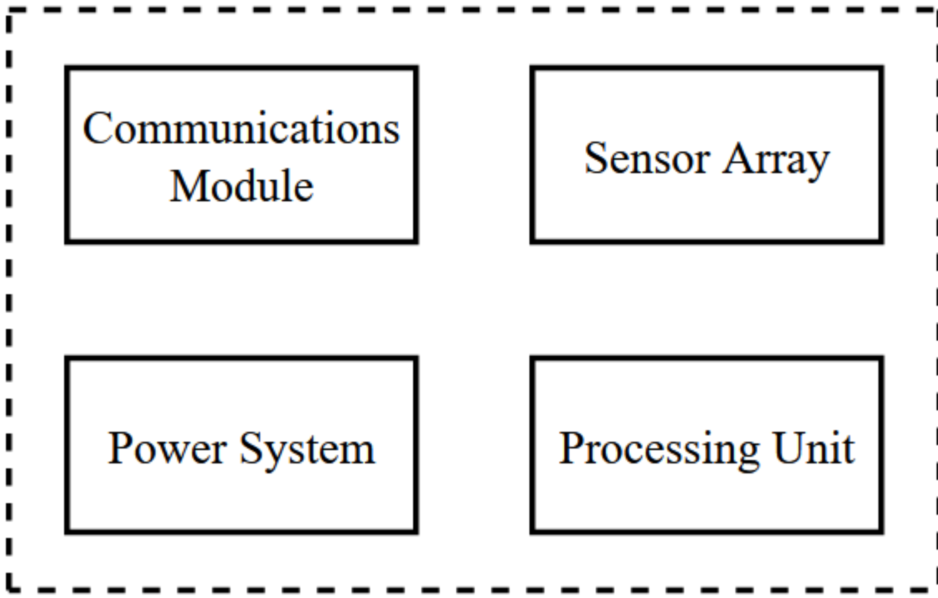


Figure 4.1: A block diagram of necessary components of the proposed framework.

Design alternatives of the framework

Three major questions shaped the design decisions for the overall framework. First: *Should the processing module and sensing module be together or separate?* Sensing and processing modules can be packaged together into a single system. This can be seen in current body sensing devices like smartwatches. The alternative is to separate the processing and sensing modules. This configuration can scale the number of sensing modules. For example, the user could integrate an electromyography module or glucose monitoring module and process data on the local processing unit. Furthermore, because the system would be packaged into smaller pieces, a smaller form factor could be achieved in the framework. Table 4.1 scores the two options. Note that all scoring is done out of 10.

Table 4.1: Decision Matrix Between a Single Module System and a Multiple Module System

	Cost	**Compactness**	**Efficiency**	**Total Score**	**Comments**
Single Modules (monitoring / processing together	7	4	7	18	A single module can be bulky without designing a PCB specifically for the device.
Multiple Modules (monitoring / processing separate)	7	6	7	20	Multiple modules cut down on the bulk a single unit might have. The design also allows the expansion of the network.

* All scoring is done out of 10

Second: *What signals should be collected with this system?* There are multiple biometric signals that can be detected using a wearable device. Examples include blood pressure, skin temperature, heart rate, glucose levels, motion, etc. These sensors can provide important health information about the user. The different bio-signals considered are shown in Table 4.2. Three factors were considered when determining which signals to measure: *comfort of the measurement process, ease of measurement, and clinical significance of the signal*. Comfort of measurement refers to the physical strain on the user. For example, blood pressure is measured with an inflating cuff. This inflation can be painful and uncomfortable. Ease of measurement refers to the effort required to measure the signal. For example, reliable glucose monitoring requires blood monitoring which involves test strips and needles. This measurement would not be considered easy to measure. Finally, clinical significance of the signals was considered. For example, glucose monitoring might not be significant for those who are not diabetic. Table 4.2 shows the decision matrix for the kinds of bio signals to be monitored. Final scores above 20 were considered for the final design, which is: heart rate and motion.

Heart rate sensor generally uses Photoplethysmography (PPG). PPG shines light through the veins and captures the reflected light. The reflected light is modulated according to the amount of blood in the veins. This method is highly effective and low powered. Generally, accelerometer uses differential capacitor structure. These details are explained more clearly in Chapter 5: Final Design.

Table 4.2: Decision Matrix to Determine Which Bio-Signals Will be Monitored

	Ease of Measurement	**Clinical Significance**	**Comfort**	**Final Score**	**Comments**
Heart Rate	8	7	8	**23**	Heart rate is a basic signal that should be monitored.
Skin Temperature	6	5	8	19	Skin temperature is an easy signal to measure and monitor.
Glucose Level	1	8	1	10	Glucose monitoring is very invasive.
Blood Pressure	3	6	3	12	Cuff measurement is the only well tested way to monitor BP.
Motion	8	6	8	**22**	Motion can be measured easily with Inertial Measurement Unit (IMU) sensors.

Third: *Which part of the body will the modules be placed?* There are multiple options for placement of modules in the user's body. Decision matrices were made to decide body area placement from the list of following candidates: wrist, neck, thigh, waist, chest, and ankle. The decision matrices for these options are in Tables 4.3 and 4.4.

Stigma, determined by visibility under common clothing, and comfort are considered for making the decision. Limited mobility due to the device placement can be attributed to lack of comfort. Additionally, for sensing modules, feasibility of motion sensing and heart rate sensing are considered.

Table 4.3: Decision Matrix to Determine Which Area of the Body the Sensors Will be Placed

	Coping Stigma	**Comfort**	**Motion Sensing**	**Heart Rate Sensing**	**Total Score**
Wrist	9	8	9	9	**35**
Neck	3	3	4	8	18
Thigh	8	6	8	4	26
Waist	8	9	8	5	30
Chest	8	6	9	9	32
Ankle	5	5	9	6	25

Table 4.4: Decision Matrix to Determine where on the body the processing module is placed

	Coping Stigma	**Comfort**	**Total Score**
Wrist	7	9	16
Neck	3	3	6
Thigh	8	3	11
Waist	8	9	**17**
Chest	8	6	14
Ankle	5	5	10

Tables 4.3 and 4.4 highlight the wrist and the waist as the optimal placement locations for the sensing and processing modules, respectively. Figure 4.2 visualizes the framework design approach thus far.

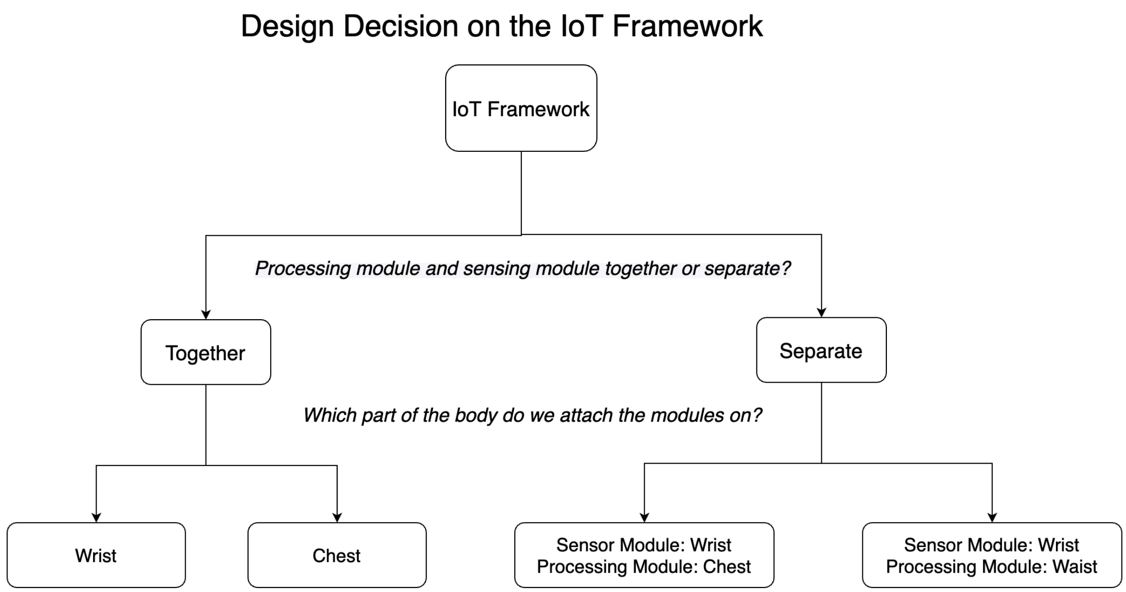


Figure 4.2: Design Decision Tree for the IoT framework designs.

Using the design questions mentioned, four IoT frameworks were considered: sensing and processing modules packaged on the a) wrist and b) chest, c) sensing on wrist and processing on waist, d) sensing on chest and processing on waist.

Since fabricating different modules is out of the scope of this design project, it is not considered; this rules-out option a) and b). Since placing a sensor module on the chest causes comfort issues, particularly in women, option d) was ruled out. Therefore, *option c, sensing on wrist and processing on waist, was chosen for the IoT framework.*

Design alternatives of the subsystems

Subsystem design alternatives that need to be considered are communication technology and comfort for the user.

Communication: There are three separate communication technologies considered: security, range, and ease of implementation.

Table 4.5: Communication technology Decision Matrix

	Security	Range	Ease of Implementation	Total Score
Bluetooth	7	6	8	21
Wi-Fi	5	8	5	18
Cellular	5	10	5	20

Fabric comfort: There are several types of fabric that can be used to avoid skin irritation. The criteria for judging the materials are average price, durability, and skin irritation. A low hypoallergenic score means a lot of skin irritation, while a high score means little to no irritation.

Table 4.6: Waistband Material Decision

	Average Price	Durability	Hypoallergenic	Final Score
Linen	6	9	9	24
Cotton	9	7	6	22
Silk	6	5	7	18
Nylon	8	8	6	22
Polyester	7	6	6	19

Table 4.7: This table shows that the technical specifications outlined in Table 2.1 are adhered to.

	Evaluation Basis
Public health	The design proposed gives the user data that is only light indicators of health, not red flags of health issues. Stress detection will only suggest that there is a possibility that the user is experiencing stress.
Safety and welfare	To adhere to the safety and welfare of the users, allergic reactions of the materials to the skin were considered in the evaluation of alternative materials for waistband.
Global factors	All components will be off the shelf components that are obtained through reputable sites.
Cultural factors	None of the proposed design alternatives posed cultural harm.
Social factors	The factor of stigma was considered when making decisions about choosing body area to place the sensors on to ensure minimum social burden for the user.
Environmental factors	None of the designs pose a threat to the environment.
Economic factors	None of the proposed design uses components which possess economic burden.

Finally, Fig. 4.3 shows the final design overview after the evaluation of design alternatives.

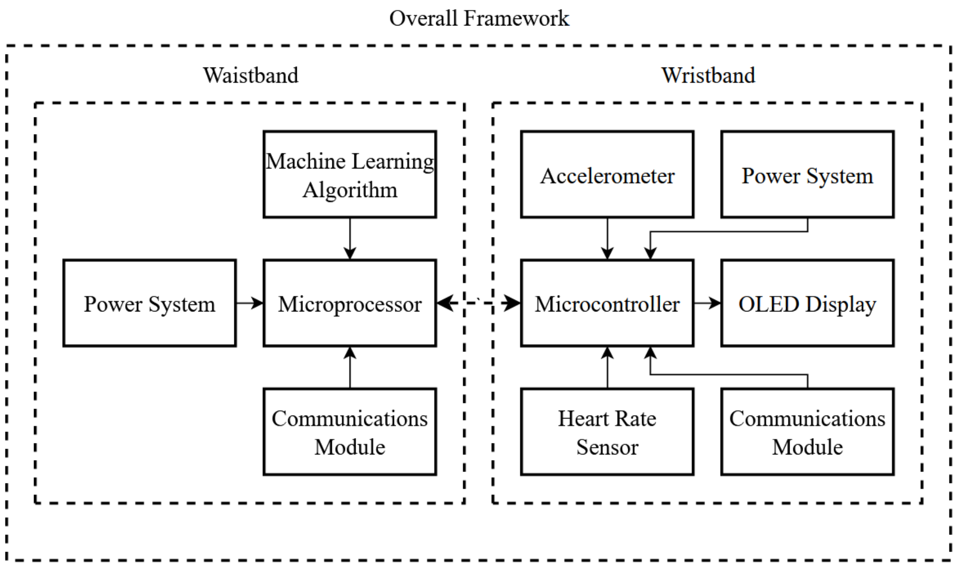


Figure 4.3: A block diagram of necessary components of the proposed framework. This figure expands on Figure 4.2 to highlight changes in the framework architecture following the comparison of alternate designs.

5. Final Design

The design of this framework consists of two components: a wristband device that is used to collect data from the user and allow the user to interface with the device, a waistband to be used for the processing and computing of the machine learning algorithm. The framework will be a closed loop system that does not rely on a phone or cloud computing for processing. The overall design can be seen in Figure 5.1 showing the information exchanged.

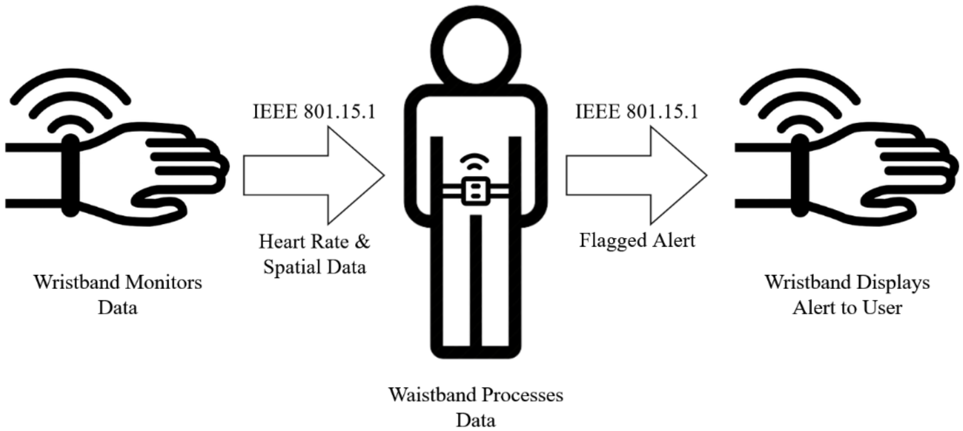


Figure 5.1: A perspective view of the framework and its components. Note that transmission arrows highlight the type of data sent and the communication protocols used.

Both the wristband and the waistband process data to maximize efficiency. The wristband will be processing the raw data collected from the user and transforming it into usable information. The waistband will take the processed data from the wristband and determine if stress is detected. The wristband will be collecting raw data using a heart rate monitor and an accelerometer.

Heart Rate Sensor

Heart rate will be collected by measuring the volume of blood in the wrist. This is accomplished by an LED blinking and a photometric sensor detecting how much light is absorbed. The more light that is absorbed the larger volume of blood in the wrist at that time. Figure 5.2 shows how the raw data will look. The peaks that are near the max amplitude are taken. The distance between these points is considered the pulse within the vein. A minimum sample rate of 1Hz will be used to achieve low power consumption and still maintain an acceptable sampling rate.

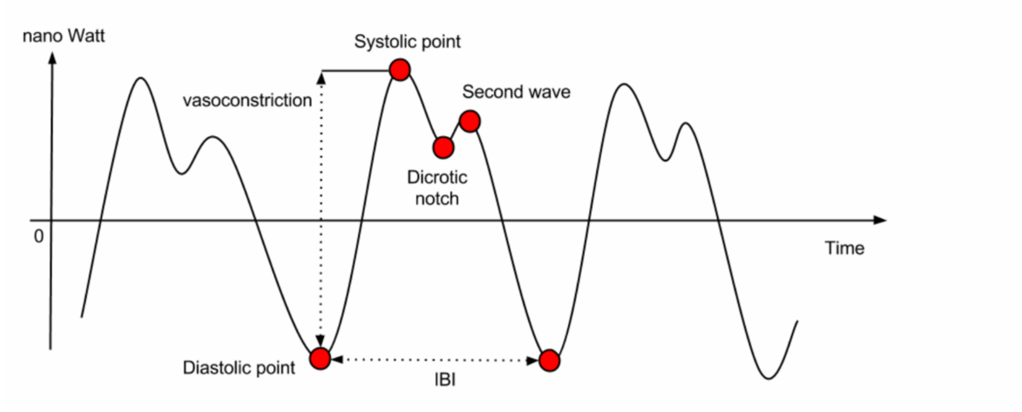


Figure 5.2: The raw data that is collected from the heart rate monitor. The graph shows the amount of LED light that is absorbed over time. This correlates to the volume of blood in the wrist over time. The period between these max amplitudes is what determines the pulse of the user [8].

Accelerometer

The type of accelerometer used in the sensor system works on the differential capacitor structure. This type of accelerometer works by the principle of oscillation and changing capacitive values. When movement occurs, the vibration of electrodes inside the device changes the capacitance which outputs a voltage signal. This signal can be processed into a readable acceleration sensor.

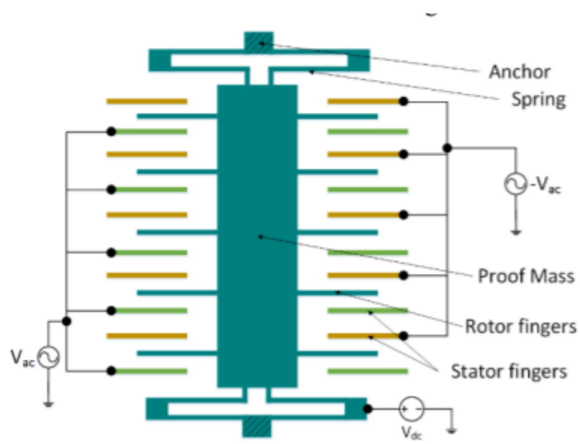


Figure 5.3: The schematic of how a differential capacitive accelerometer works. This figure shows how the different fingers interact with each other when acceleration occurs [9].

Software Wristband

Some processing happens within the wristband. Many small microprocessors have limited input pins. To reduce the number of input pins used on the processor, the raw data will be transmitted to the processor from the sensors through a multiplexer. This allows multiple inputs to share the same data bus but have their own memory address.

An OLED screen will be utilized for the user to interact with the system and obtain information that is processed. This should include the heart rate, motion, and if stress is detected in the user.

The final component of the wristband will be a Bluetooth receiver and transmitter. All components of the wristband will be bought off the shelf. The components will need to use the data collection as stated above. Table 5.1 through Table 5.3 shows the decision matrixes that were used in determining the parts used in prototyping.

Waistband Inference

The waistband will process the data collected from the wristband. To ensure that our framework can house a machine learning model, a pre-trained support vector machine (SVM) will be embedded into the microprocessor on the waistband. The features that were used in the study by Carreiro et. al will be embedded on the processor on the waistband [10]. Using the model will require calculation of features of the signals. The calculated features were standard statistical features like mean and variance of accelerometer signals over 5 a minute window. Furthermore, instantaneous amplitude of the accelerometer signals was used on the window of the accelerometer signals using Hilbert Transformation. More information on feature extraction can be found in Appendix D.

Power

The only hardware needed for the waistband will be a microprocessor and Bluetooth. One thing that will need to be considered for both the wristband and the waistband is management of the voltage coming from the battery. Many off-the-shelf boards will contain battery management systems (BMS), but unfortunately some do not. Batteries supply a range of voltage if directly connected. A consistent voltage is needed with the sensitive components in micro processing boards. This is accomplished using the BMS. The BMS will regulate the current to keep the output voltage constant. The BMS also only operates with a range of voltage. This protects the microprocessor from spikes in voltages as well. The BMS can allow the voltage to stay the same or it can boost or reduce the voltage from the battery.

Software Flowcharts

Figures 5.4 and 5.5 show the software flowcharts of the algorithm used by the wristband and waistband, respectively.

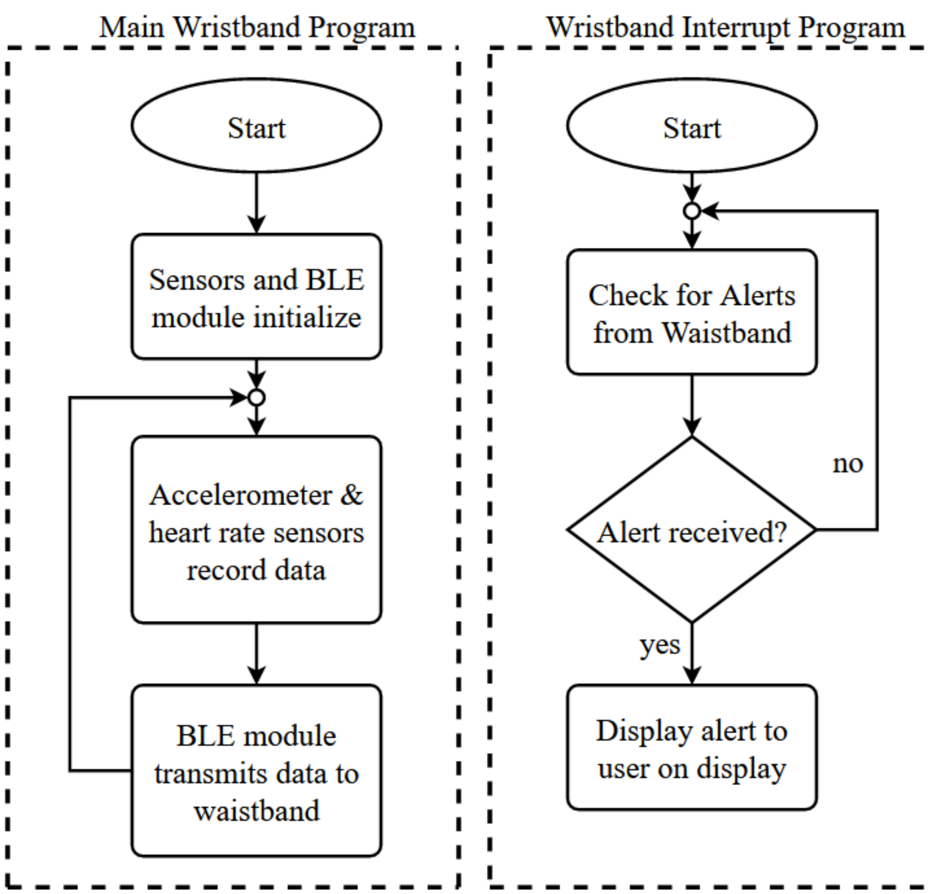


Figure 5.4: Flowchart of wristband algorithm

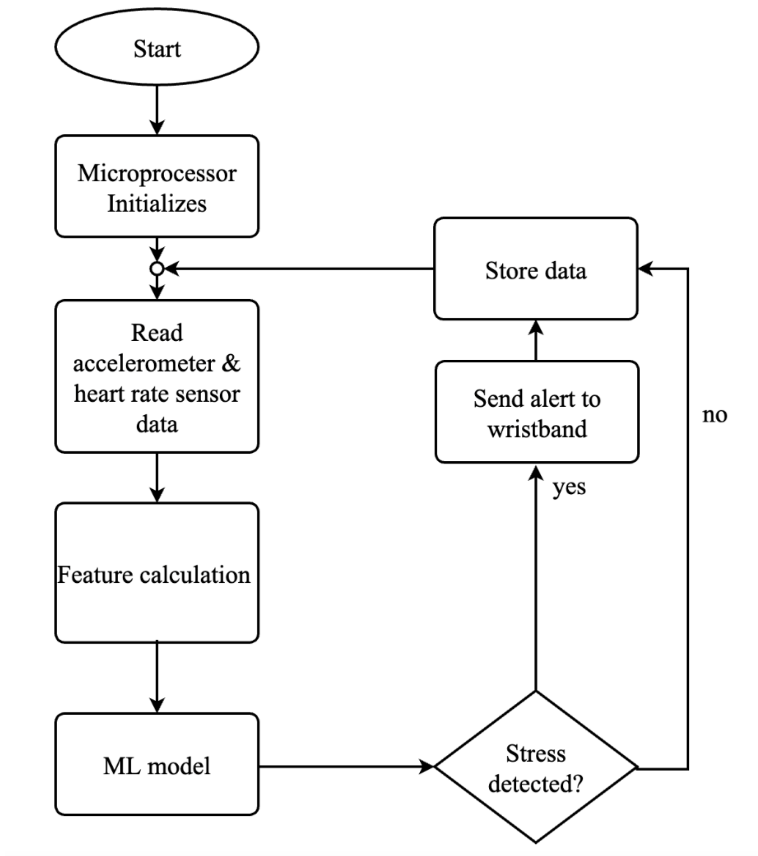


Figure 5.5: Flowchart of waistband algorithm

<u>Prototype Component Decision Matrices</u>

Tables 5.1 – 5.3 are the decision matrices for the prototype components. These components were evaluated on cost, size, and availability. All scoring is out of 10. The P and M columns stand for processing and monitoring units, respectively.

Table 5.1: Decision matrix for microcontrollers

	Cost	Size (mm)	Availability	Standout Features	Score for Units	
					P	M
TinyPico Nano	$19.50	13 x 27	Yes	BLE, Wi-Fi, Integrated LiPo BMS and Voltage Regulator	2	6
Teensy 4.1	$26.85	61 x 18	Yes	ARM Cortex-M7, 18 analog input pins, 55 digital GPIO, SD Card Support	9	1
TinyScreen+	$39.95	25.8 x 25.0	Yes	Integrated 24.4 mm OLED display, integrated LiPo charge/use management, Arduino Zero processor	2	9
Raspberry Pi Zero W	$10.00	66 x 30.5	No	SD Card Support, mini-HDMI, 512 MB of RAM, Single-core 1 GHz processor	8	3

Table 5.2: Decision matrix for heart rate sensor

	Cost	Size (mm)	Availability	Features	Score
AST1041 (Evaluation Board Based on MAX30101)	$19.95	10 x 10	Yes	Typical 600 µA supply current,	8
SEN0203 (Evaluation Board Based on SON3130)	$16.00	28 x 24	Yes	Operating Current < 10 mA, integrated slots for straps	6

Table 5.3: Decision matrix for accelerometer

	Cost	Size (mm)	Availability	Features	Score
SEN-17871 (Evaluation Board Based on KX132-1211)	$13.95	25.4 x 25.4	Yes	Current consumption of 148 µA, built-in sensing functions	4
Adafruit 4554 (Evaluation Board Based on ICM20948)	$14.95	25.7 x 17.7	Yes	3-axis Gyroscope, Accelerometer, Compass	5
Accelerometer Wireling (Board based on BMA250)	$4.95	10 x 10	Yes	Current consumption of 139 µA @ 2kHz data rate, 3-axis, integrated temperature sensor	9

Based on the dimensions of the components comprising the wristband and waistband devices, a 3D modeled housing was designed to contain the components in each device. Figures 5.6 and 5.7 depict the 3D modeled design of the wristband and waistband, respectively. The device housings were modeled using TinkerCAD. Refer to Appendix C for the dimensions.

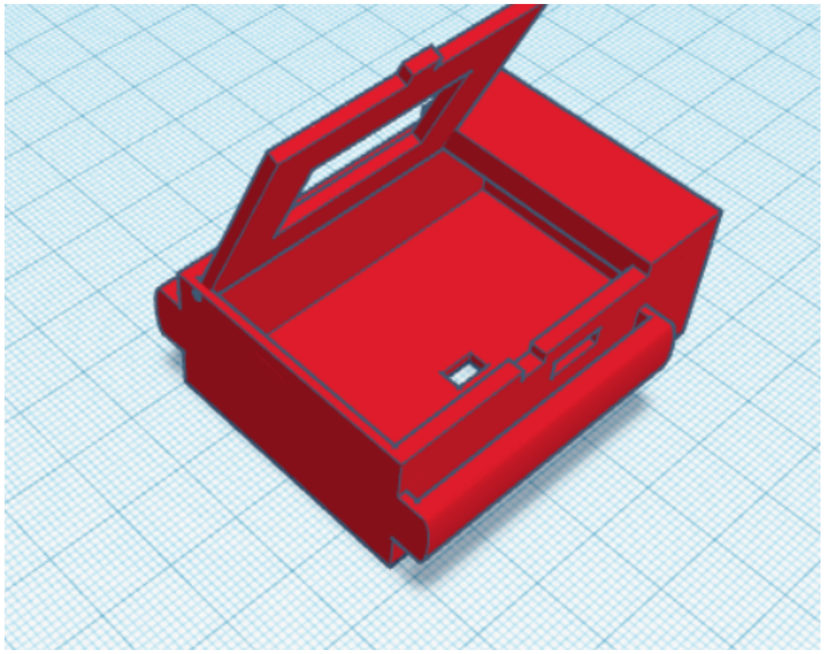


Figure 5.6: 3D model of wristband housing using TinkerCAD.

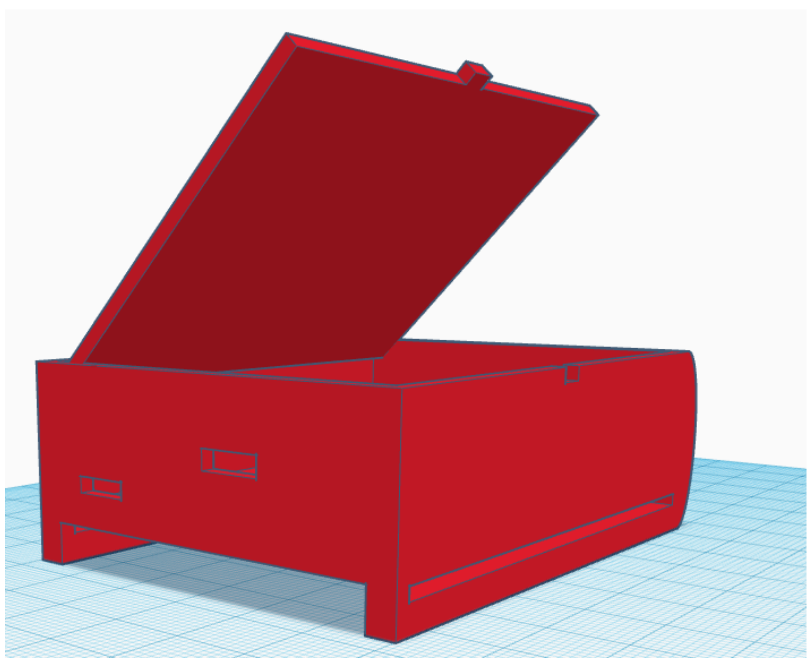


Figure 5.7: 3D model of waistband housing using TinkerCAD.

Prototype Component Design

The configuration of the parts chosen using the decision matrixes previously seen is shown in the following figures. Figure 5.8 shows the physical connections of the boards that will be used for the wristband. Figure 5.9 shows the physical connections of the boards that will be used for the waistband. All parts are off the shelf and were partially chosen based on the ease of use that they can be used together. No voltage changes need to be made as the pin outputs and inputs align with each other. A small voltage change was made in the waistband. An extra battery management system (BMS) had to be added to convert the 3.7V LiPo battery to a 5V source that was acceptable for the Teensy Board. This battery management system is the Adafruit PowerBoost 1000 which provides the correct voltage to the microprocessors in the waistband.

Figure 5.8 represents how the physical components of the wristband will be put together. The figure is organized in a way of how each part will be stacked on top of one another. The lines that run down the right side of the figure represent the physical layers of the wristband. The OLED being on the top of the stack, so the user can easily interact with it. The sensors being on the bottom of the stack, so data can be collected from the user.

Figure 5.9 represents the physical components of the waistband and how they are put together. The figure is set up in the same way as Figure 5.8. The top of the figure represents the top of the stack and the bottom of the figure representing the bottom of the stack. The lines on the right side of the figure represent the physical layers of the components.

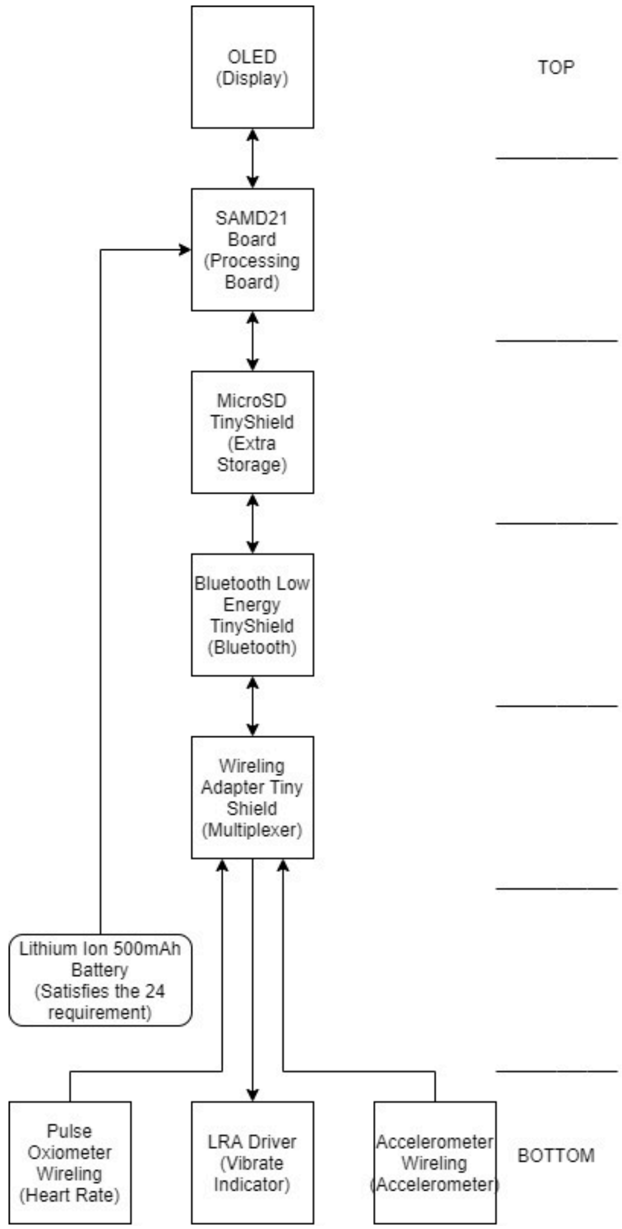


Figure 5.8: Representation of the Physical Component Layout for Wristband Module. The figure represents the level of the component with the lines to the right. The bottom and top are marked.

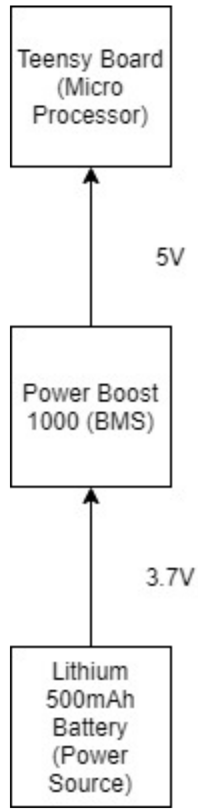


Figure 5.9: Representation of the Physical Layout for the Waistband Module. Unlike Fig 5.8, this only represents the connections of the components.

6. Subsystem Simulation

Since feature extraction has the potential to take some time to process and latency characteristics is important in the IoT framework design, run time analysis was performed using the MATLAB's Hilbert transformation subroutine. As a part of feature extraction pipeline, Hilbert Transformation, mean, standard deviation, and gamma fitting needs to be implemented in the microcontroller on the IoT framework design. Hilbert transformation subroutine consists of the use of Fast Fourier Transform (FFT) and Inverse Fourier Transform (IFT) methods. A more detail explanation of feature extraction is provided in Appendix D: Feature Extraction for Machine Learning. Before performing the run-time analysis, we anticipated the run time to be lower than 1 second on the Teensy 4.1.

The run-time analysis was performed by implementing the feature extraction subroutine in MATLAB. Five minutes dummy data was generated to perform the feature extraction on. The execution time for the feature extraction routine was recorded N = 500 times. Since the analysis was performed in MATLAB on a computer with an operating system, the execution time is not constant in every execution. Certain assumptions were made to estimate the run time in Teensy 4.1 from the run time in the computer.

Dummy Data: The dummy data was generated using the following equation.

$$x[N] = \sin(2\pi \times 1000N) + sin(2\pi \times 10N) + sin(2\pi \times 10N + \pi) + sin(2\pi \times 17N + 2.1\pi) + \Psi$$

Where, Ψ is a random number between 0 and 1. N is the length of the vector of signals.

Window length of five minutes sampled at 32 Hz was considered for feature extraction. The length of the vector can be given as:

Accelerometer: $32 \times 5 \times 60 = 9600$
Heart Rate: $1 \times 5 \times 60 = 300$

Hardware Specification: The simulation was performed on Apple's M1 chip with the following specifications.

a. 3.2 GHz clock speed
b. 16 GB RAM
c. 8 cores

The target hardware for the IoT framework, Teensy 4.1, has the following specifications.

a. 600 MHz
b. 1 MB RAM

Although there are a lot of factors in play to calculate execution speed from one processor to another (number of cores, computer architecture, instruction set, etc.), we assume that the run time only depends on the clock speed. With these assumptions in place, the run time of the feature extraction subroutine can be given as:

$$runtime_{ARM\ cortex\ M7} = \frac{clockspeed_{Apple\ M1\ chip}}{clockspeed_{ARM\ Cortex\ M7}} \times runtime_{MATLAB\ Apple\ M1\ chip}$$

For M1 processor, the mean run time measured for the feature extraction routine was $\mu = 0.0019\ sec$ with standard deviation of $\sigma = 0.0022\ sec$. For Cortex M7 processor, the mean estimated run time was $\mu = 0.0102\ sec$ and $\sigma = 0.0117\ sec$. Figure 6.1 shows the histogram of the run time on Apple's M1 Processor. Figure 6.2 shows the histogram of the estimated run time on Teensy 4.1. The MATLAB script used for the simulation is attached in Appendix E.

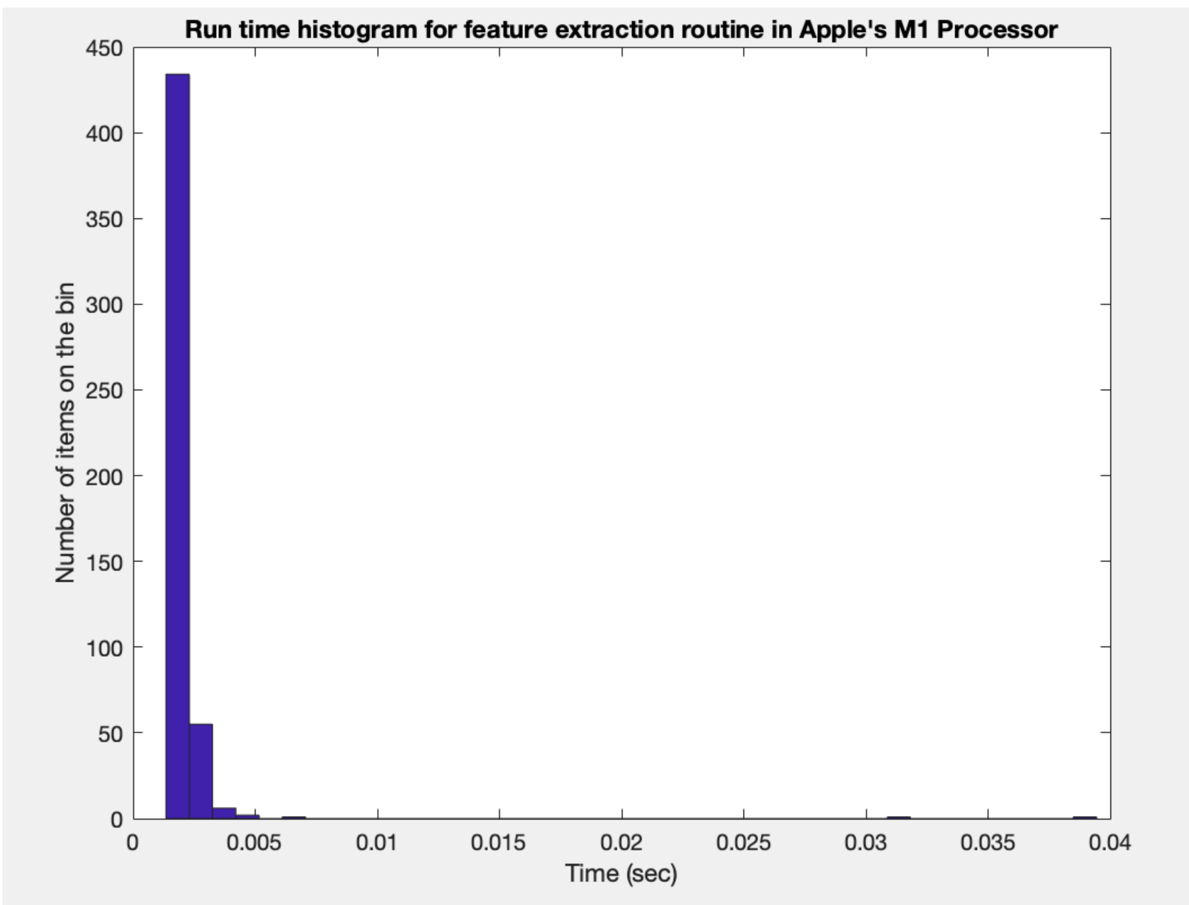


Figure 6.1: Histogram of the run-time in Apple's M1 Processor (N = 500)

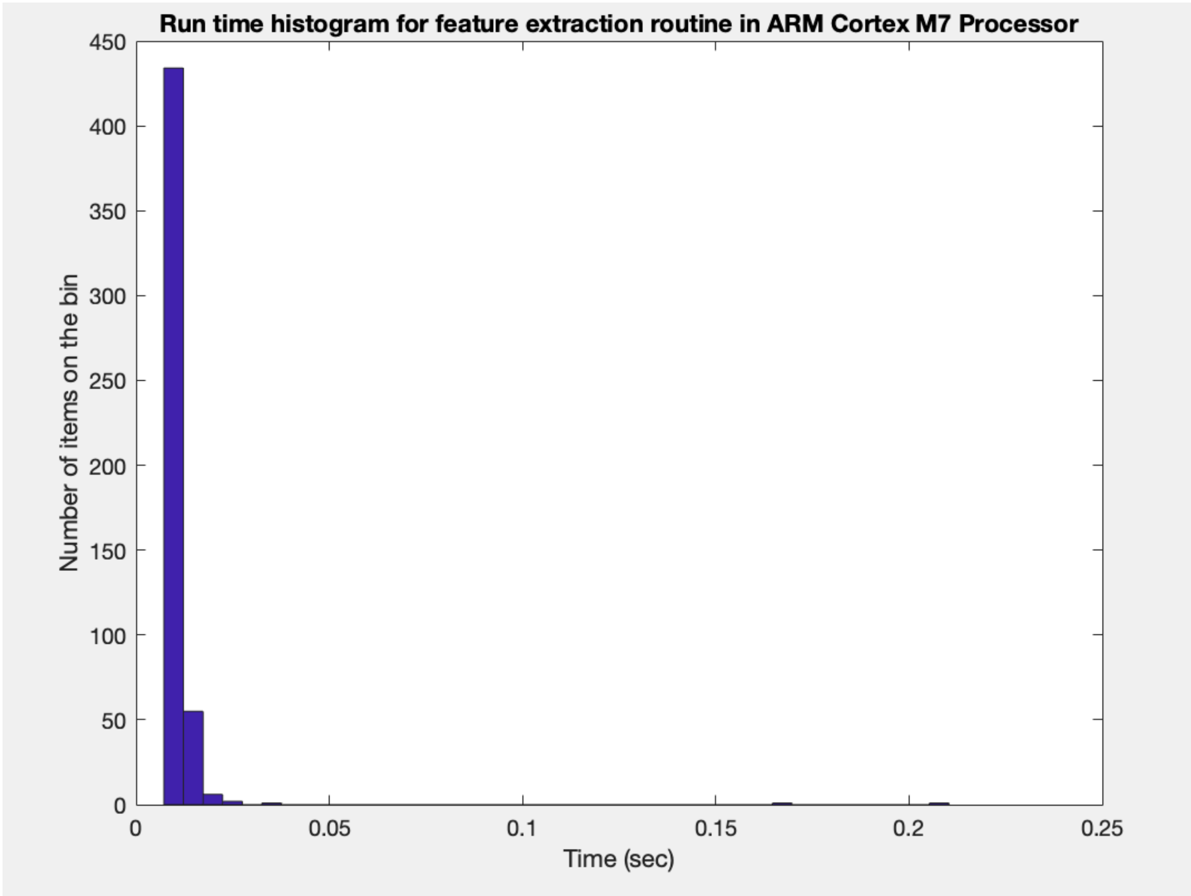


Figure 6.2: Histogram of the estimated run-time in ARM Cortex M7 Processor (N = 500)

The estimated run time on Teensy 4.1 ($\mu = 0.0102\ sec$ and $\sigma = 0.0117\ s$ec) is well below the anticipated one second time by a factor of 100. Therefore, the proposed design does not need to be changed based on the simulation performed.

7. Final Design Specifications

Below is a list of important specifications of the design.

- Weight
    - The total weight of the wristband without casing is 20.09 grams (0.044 lbs.). The casing is expected to be half of the parts' weight. The total weight of the waistband without casing is 20.73 grams (0.046 lbs.) The weight of the casing and band should not exceed more than 2 lbs.

- Size
  - The size of the wristband is 35mm - 25mm - 38.4mm (1.38in-0.98in-1.51in) (length, width, height). The size of the waistband is 45mm-22mm-29.11mm (1.77in-0.87in-1.15in). This does not include the casing and waistband that will be used to support the device on the user.
- Power
  - Both devices will be powered by batteries that are rechargeable. The 500 mAh battery was chosen for its power capacity, while staying of a small size.
- Casing
  - The casing for the wristband will be made of thermoplastic polyurethane (TPU). This is to allow comfort and ability to 3D print the designs. The casing for the waistband was determined using decision matrix in Table 4.6, linen. The waistband will be secured using Velcro.
- Risk
  - One risk for the device is the user slamming device against objects while wearing it on the wrist. The casing will need to absorb collisions. The waistband needs to be water resistant to protect against the user's sweat.
- Overheating
  - The wristband does not use a significant power draw, so overheating is not a concern. The waistband does not use a significant power drawing during operations but can draw up to 2 A while charging. Fast charging USB cables are recommended. To prevent overheating, a warning will be placed stating the device should be turned off during charging.

8. Project Work Plan for Senior Design II

The construction of the prototype, which meets all the design specifications laid out in Chapter 2, is broken down into four phases: Procurement, Software Testing, Prototype Testing, and Final Prototype. In each phase, a corresponding list of deliverables shall be completed by team members. Figure 8.1 depicts the work breakdown structure for the project, where the orange boxes indicate the deliverables for a respective phase. The deliverables shall be completed by the team by following Gantt chart located in Appendix A. A summarized Gantt chart is shown in Table 8.1. The procedure for testing whether the prototype meets all the design specifications is shown in Table 8.2.

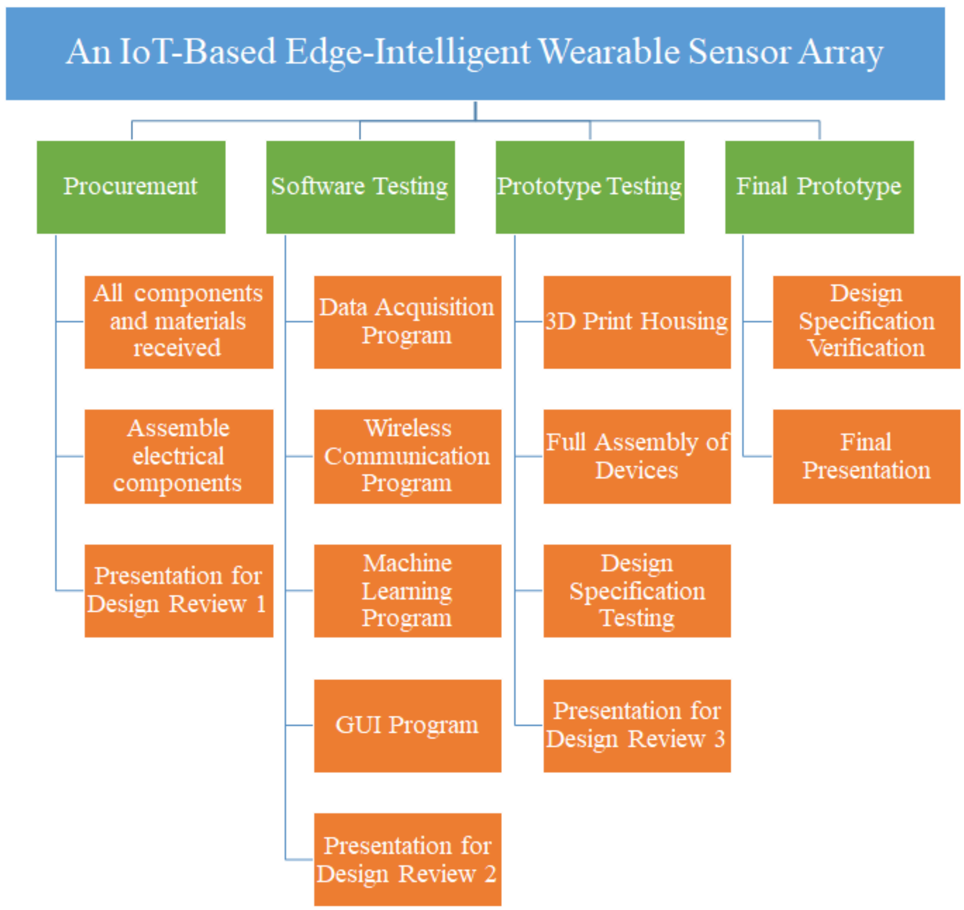


Figure 8.1: Phase-based work breakdown structure.

Table 8.1: Summarized Gantt Chart

Task	Duration (days)	Start	End	December	January	February	March	April
Project	140	12/1/2021	4/20/2022	12/1/21 - 4/20/22				
Phase 1 - Procurement	47	12/1/2021	1/17/2022	12/1/21 - 1/17/22				
Phase 2 - Software Development & Testing	53	1/1/2022	2/23/2022		1/1 - 2/23			
Phase 3 - Prototype Fabrication & Testing	35	2/21/2022	3/28/2022			2/21 - 3/28		
Phase 4 - Final Prototype	24	3/27/2022	4/20/2022				3/27 - 4/20	

Table 8.2: Prototype testing procedures

Target Specifications	Testing Procedure
The framework is wearable	Equip devices on the user. The user must not be in discomfort as a result of wearing the devices.
Devices wirelessly communicate	Verify that the data received by the waistband device matches the data sent by the wristband device.
Subsystems draw less than 500 mA	Connect the ammeter to the output terminal of the battery and the input terminal of the wristband and confirm that the current drawn from the battery is less than 500 mA. Repeat the process for the waistband.
Capable of continuous monitoring, even when charging.	The user shall wear the device until the battery needs to charge. The user will charge the battery and verify that the framework's operation remains unchanged.
Monitors heart rate and motion of the user	The output of the heart rate sensor and the motion sensor shall be observed during operation to confirm proper functionality.
A chassis houses all device components	Verify that the designed housing adequately accommodates all the components which make up the wristband and waistband. Verify alignment of charging ports.
Framework does not cause skin irritation for the user	Verify that only the materials selected for the construction of the devices, as stated in chapters 4 and 5, are used. Additionally, a team member shall wear the devices for at least 10 hours and report any discomfort that may occur.
The machine learning algorithm detects stress	The user shall wear the device and verify that any determination of stress made by the device aligns whether the user feels stress or not.

9. Resources Required

The resources required for this project split into two categories.

1. Necessary and readily available
2. Necessary but not readily available

These resources include both tools, and parts. Table 9.1 lists the required parts that were decided on in Chapter 5 as well as additional parts that are required. These parts should be ordered on time as not to impact the timetable set in Chapter 8. Furthermore, the tools required are listed in 9.2.

Table 9.1: Electronic parts required.

Part	Manufacturer	Function	Price/ea	Quantity
TinyScreen+	TinyCircuits	Wristband Microcontroller	$39.95	1
Accelerometer Wireling	TinyCircuits	Accelerometer	$4.95	1
Bluetooth Low Energy Tinyshield	TinyCircuits	Bluetooth Communication Module	$29.95	2
Pulse Oximeter Sensor Wireling	TinyCircuits	Heart rate sensor	$19.95	1
Wireling Adapter Tinyshield	TinyCircuits	Sensor Adapter board	$9.95	1
LiPo Battery	TinyCircuits	Battery	$6.95	2
5-Pin Wireling Cables, 50 mm (5-pack)	TinyCircuits	Sensor Connections	$2.99	1
Micro USB Cable	TinyCircuits	Processor Connection	$3.95	1
Teensy 4.1 Dev Board	PJRC	Waistband Microprocessor	$26.95	1
PowerBoost 1000 Charger	Adafruit	Charge Management	$19.95	1

Table 9.2: Tools Required

Tool	Function	Readily Available?
3D Printer and Filament	Housing Creation	Yes
Soldering Station	Electrical Connection Construction	Yes

In addition to material resources, labor is required to finish the project. The labor required is focused on software creation as the physical design should be simple in construction.

10. Conclusions

The IoT-based edge-intelligent wearable sensor array is a proof-of-concept framework design that does not rely on a smartphone, or cloud to process the data that is collected. The design consists of two devices: a wristband and a waistband. The wristband collects the two biological signals used in this framework: heart rate and motion. The waistband is used to process the data that is collected from the wristband and determine if stress is detected in the user.

The design uses a Teensy 4.1 board for processing. This board was chosen due to its ability to be expanded upon and incorporate more than the two biological signals used in this design. It is anticipated that, if this concept is successful, the framework will be modular and able to scale the number of biological signals collected based on the user's needs. The design will be constructed into a proof-of-concept prototype in the timeframe detailed in Appendix A.

11.References

The style used to cite the references is that of the IEEE.

[1] E. J. Wang, W. Li, D. Hawkins, T. Gernsheimer, C. Norby-Slycord, and S. N. Patel, "HemaApp: noninvasive blood screening of hemoglobin using smartphone cameras," 2016, pp. 593-604.

[2] D. H. Epstein *et al.*, "Prediction of stress and drug craving ninety minutes in the future with passively collected GPS data," *NPJ digital medicine,* vol. 3, no. 1, pp. 1-12, 2020.

[3] J. Rodríguez-Arce, L. Lara-Flores, O. Portillo-Rodríguez, and R. Martínez-Méndez, "Towards an anxiety and stress recognition system for academic environments based on physiological features," *Computer methods and programs in biomedicine,* vol. 190, p. 105408, 2020.

[4] S. A. Taylor, N. Jaques, E. Nosakhare, A. Sano, and R. Picard, "Personalized multitask learning for predicting tomorrow's mood, stress, and health," *IEEE Transactions on Affective Computing,* 2017.

[5] P. Liu, K. Astudillo, D. Velez, L. Kelley, D. Cobbs-Lomax, and E. Spatz, "Use of mobile health apps in low-income populations: a prospective study of facilitators and barriers," *medRxiv,* 2019.

[6] M. Gross, R. Miller, and A. Pascalev, "Ethical Implementation of Wearables in Pandemic Response: A Call for a Paradigm Shift," 05/20 2020.

[7] "Wireless Medical Devices." Center for Devices and Radiological Health, U.S Food and Drug Administration. https://www.fda.gov/medical-devices/digital-health-center-excellence/wireless-medical-devices (accessed Sept 14, 2021).

[8] E. E4, "Blood Volume Pressure (BVP)," https://support.empatica.com/hc/article_attachments/360032283952/blobid0.png, Ed., ed.

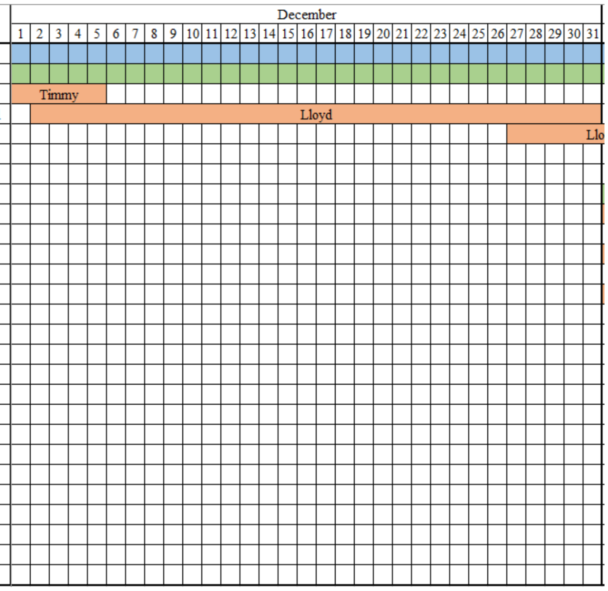
[9] S. Sinha, S. Shakya, R. Mukhiya, R. Gopal, and B. D. Pant, "Design and simulation of MEMS differential capacitive accelerometer," 2014.

[10] S. Carreiro, K. K. Chintha, S. Shrestha, B. Chapman, D. Smelson, and P. Indic, "Wearable sensor-based detection of stress and craving in patients during treatment for substance use disorder: A mixed methods pilot study," *Drug and Alcohol Dependence,* p. 107929, 2020.

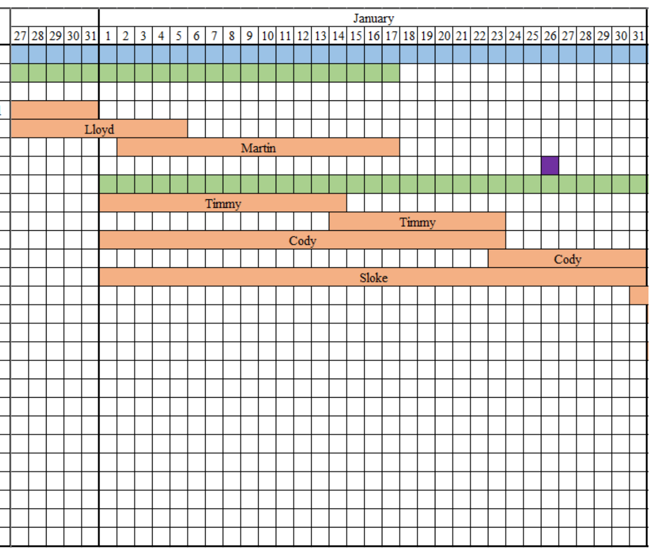
12. Appendices

Appendix A: Gantt Chart

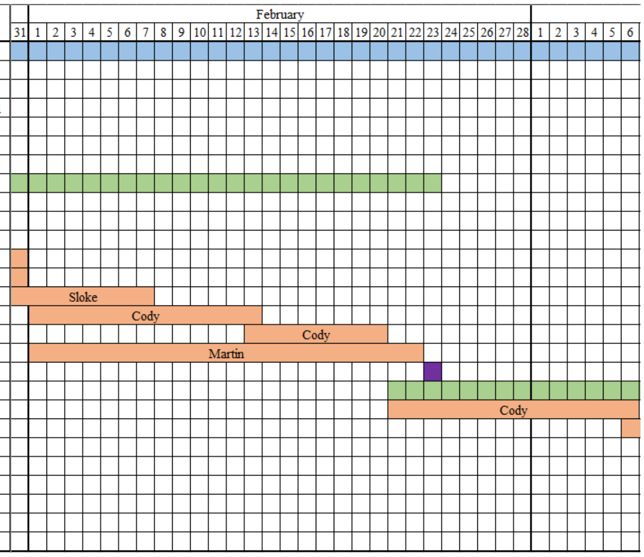
Task	Duration (days)	Start	End
Project	140	12/1/2021	4/20/2022
Phase 1 - Procurement and Circuit Assembly	47	12/1/2021	1/17/2022
Order all components	4	12/1/2021	12/5/2021
Receive all components	29	12/2/2021	12/31/2021
Assemble components required to begin software testing	9	12/27/2021	1/5/2022
Design review 1 presentation	15	1/2/2022	1/17/2022
Design review 1 meeting	1	1/26/2022	1/26/2022
Phase 2 - Software Development and Testing	53	1/1/2022	2/23/2022
Data acquisition program development	13	1/1/2022	1/14/2022
Data acquisition program testing	9	1/14/2022	1/23/2022
Wireless communication program development	22	1/1/2022	1/23/2022
Wireless communication program testing	8	1/23/2022	1/31/2022
Machine learning program development	30	1/1/2022	1/31/2022
Machine learning program testing	7	1/31/2022	2/7/2022
GUI program development	12	2/1/2022	2/13/2022
GUI program testing	7	2/13/2022	2/20/2022
Design review 2 presentation	21	2/1/2022	2/22/2022
Design review 2 meeting	1	2/23/2022	2/23/2022
Phase 3 - Prototype Fabrication and Testing	35	2/21/2022	3/28/2022
Initial prototype construction	13	2/21/2022	3/6/2022
Initial prototype testing	18	3/6/2022	3/24/2022
Design review 3 presentation	14	3/14/2022	3/28/2022
Design review 3 meeting	1	3/30/2022	3/30/2022
Phase 4 - Final Prototype	24	3/27/2022	4/20/2022
Make required adjustments to meet design specifications	18	3/27/2022	4/14/2022
Final presentation	16	4/1/2022	4/17/2022
Final design meeting	1	4/20/2022	4/20/2022



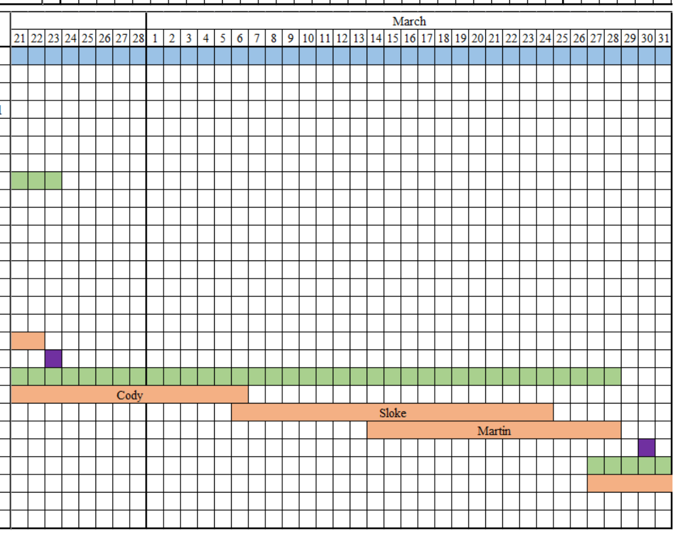
Task	Duration (days)	Start	End
Project	140	12/1/2021	4/20/2022
Phase 1 - Procurement and Circuit Assembly	47	12/1/2021	1/17/2022
Order all components	4	12/1/2021	12/5/2021
Receive all components	29	12/2/2021	12/31/2021
Assemble components required to begin software testing	9	12/27/2021	1/5/2022
Design review 1 presentation	15	1/2/2022	1/17/2022
Design review 1 meeting	1	1/26/2022	1/26/2022
Phase 2 - Software Development and Testing	53	1/1/2022	2/23/2022
Data acquisition program development	13	1/1/2022	1/14/2022
Data acquisition program testing	9	1/14/2022	1/23/2022
Wireless communication program development	22	1/1/2022	1/23/2022
Wireless communication program testing	8	1/23/2022	1/31/2022
Machine learning program development	30	1/1/2022	1/31/2022
Machine learning program testing	7	1/31/2022	2/7/2022
GUI program development	12	2/1/2022	2/13/2022
GUI program testing	7	2/13/2022	2/20/2022
Design review 2 presentation	21	2/1/2022	2/22/2022
Design review 2 meeting	1	2/23/2022	2/23/2022
Phase 3 - Prototype Fabrication and Testing	35	2/21/2022	3/28/2022
Initial prototype construction	13	2/21/2022	3/6/2022
Initial prototype testing	18	3/6/2022	3/24/2022
Design review 3 presentation	14	3/14/2022	3/28/2022
Design review 3 meeting	1	3/30/2022	3/30/2022
Phase 4 - Final Prototype	24	3/27/2022	4/20/2022
Make required adjustments to meet design specifications	18	3/27/2022	4/14/2022
Final presentation	16	4/1/2022	4/17/2022
Final design meeting	1	4/20/2022	4/20/2022



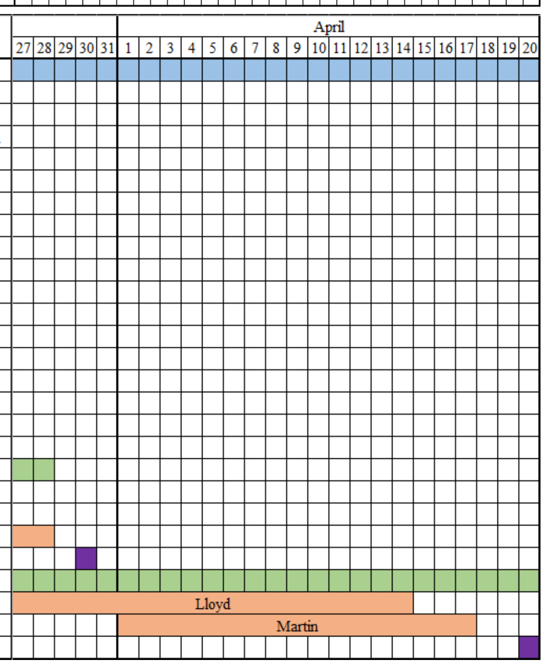
Task	Duration (days)	Start	End
Project	140	12/1/2021	4/20/2022
Phase 1 - Procurement and Circuit Assembly	47	12/1/2021	1/17/2022
Order all components	4	12/1/2021	12/5/2021
Receive all components	29	12/2/2021	12/31/2021
Assemble components required to begin software testing	9	12/27/2021	1/5/2022
Design review 1 presentation	15	1/2/2022	1/17/2022
Design review 1 meeting	1	1/26/2022	1/26/2022
Phase 2 - Software Development and Testing	53	1/1/2022	2/23/2022
Data acquisition program development	13	1/1/2022	1/14/2022
Data acquisition program testing	9	1/14/2022	1/23/2022
Wireless communication program development	22	1/1/2022	1/23/2022
Wireless communication program testing	8	1/23/2022	1/31/2022
Machine learning program development	30	1/1/2022	1/31/2022
Machine learning program testing	7	1/31/2022	2/7/2022
GUI program development	12	2/1/2022	2/13/2022
GUI program testing	7	2/13/2022	2/20/2022
Design review 2 presentation	21	2/1/2022	2/22/2022
Design review 2 meeting	1	2/23/2022	2/23/2022
Phase 3 - Prototype Fabrication and Testing	35	2/21/2022	3/28/2022
Initial prototype construction	13	2/21/2022	3/6/2022
Initial prototype testing	18	3/6/2022	3/24/2022
Design review 3 presentation	14	3/14/2022	3/28/2022
Design review 3 meeting	1	3/30/2022	3/30/2022
Phase 4 - Final Prototype	24	3/27/2022	4/20/2022
Make required adjustments to meet design specifications	18	3/27/2022	4/14/2022
Final presentation	16	4/1/2022	4/17/2022
Final design meeting	1	4/20/2022	4/20/2022



Task	Duration (days)	Start	End
Project	140	12/1/2021	4/20/2022
Phase 1 - Procurement and Circuit Assembly	47	12/1/2021	1/17/2022
Order all components	4	12/1/2021	12/5/2021
Receive all components	29	12/2/2021	12/31/2021
Assemble components required to begin software testing	9	12/27/2021	1/5/2022
Design review 1 presentation	15	1/2/2022	1/17/2022
Design review 1 meeting	1	1/26/2022	1/26/2022
Phase 2 - Software Development and Testing	53	1/1/2022	2/23/2022
Data acquisition program development	13	1/1/2022	1/14/2022
Data acquisition program testing	9	1/14/2022	1/23/2022
Wireless communication program development	22	1/1/2022	1/23/2022
Wireless communication program testing	8	1/23/2022	1/31/2022
Machine learning program development	30	1/1/2022	1/31/2022
Machine learning program testing	7	1/31/2022	2/7/2022
GUI program development	12	2/1/2022	2/13/2022
GUI program testing	7	2/13/2022	2/20/2022
Design review 2 presentation	21	2/1/2022	2/22/2022
Design review 2 meeting	1	2/23/2022	2/23/2022
Phase 3 - Prototype Fabrication and Testing	35	2/21/2022	3/28/2022
Initial prototype construction	13	2/21/2022	3/6/2022
Initial prototype testing	18	3/6/2022	3/24/2022
Design review 3 presentation	14	3/14/2022	3/28/2022
Design review 3 meeting	1	3/30/2022	3/30/2022
Phase 4 - Final Prototype	24	3/27/2022	4/20/2022
Make required adjustments to meet design specifications	18	3/27/2022	4/14/2022
Final presentation	16	4/1/2022	4/17/2022
Final design meeting	1	4/20/2022	4/20/2022



Task	Duration (days)	Start	End
Project	140	12/1/2021	4/20/2022
Phase 1 - Procurement and Circuit Assembly	47	12/1/2021	1/17/2022
Order all components	4	12/1/2021	12/5/2021
Receive all components	29	12/2/2021	12/31/2021
Assemble components required to begin software testing	9	12/27/2021	1/5/2022
Design review 1 presentation	15	1/2/2022	1/17/2022
Design review 1 meeting	1	1/26/2022	1/26/2022
Phase 2 - Software Development and Testing	53	1/1/2022	2/23/2022
Data acquisition program development	13	1/1/2022	1/14/2022
Data acquisition program testing	9	1/14/2022	1/23/2022
Wireless communication program development	22	1/1/2022	1/23/2022
Wireless communication program testing	8	1/23/2022	1/31/2022
Machine learning program development	30	1/1/2022	1/31/2022
Machine learning program testing	7	1/31/2022	2/7/2022
GUI program development	12	2/1/2022	2/13/2022
GUI program testing	7	2/13/2022	2/20/2022
Design review 2 presentation	21	2/1/2022	2/22/2022
Design review 2 meeting	1	2/23/2022	2/23/2022
Phase 3 - Prototype Fabrication and Testing	35	2/21/2022	3/28/2022
Initial prototype construction	13	2/21/2022	3/6/2022
Initial prototype testing	18	3/6/2022	3/24/2022
Design review 3 presentation	14	3/14/2022	3/28/2022
Design review 3 meeting	1	3/30/2022	3/30/2022
Phase 4 - Final Prototype	24	3/27/2022	4/20/2022
Make required adjustments to meet design specifications	18	3/27/2022	4/14/2022
Final presentation	16	4/1/2022	4/17/2022
Final design meeting	1	4/20/2022	4/20/2022



Appendix B: Knowledge and Skills Used in Project

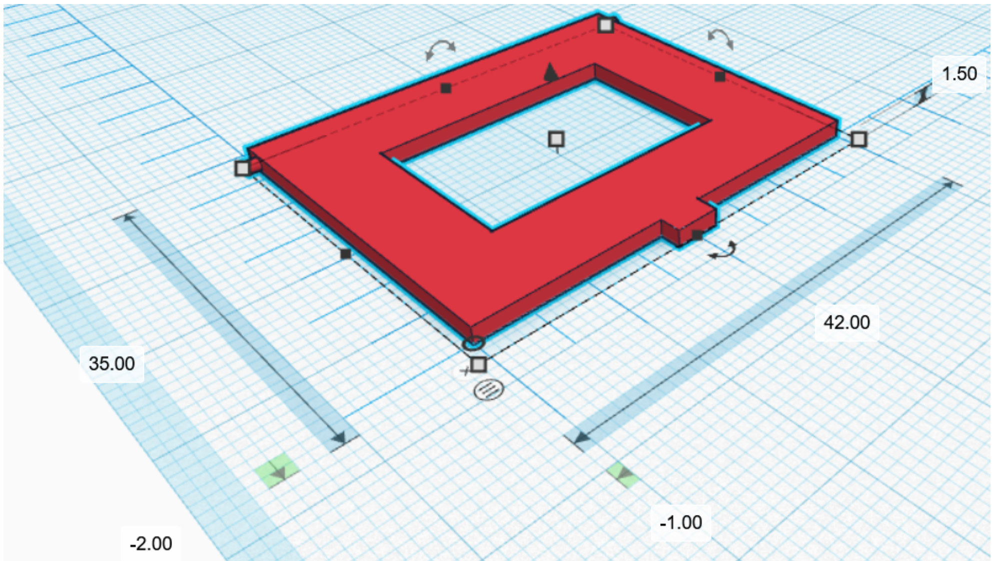
The table below has the course number and the knowledge and skills learned in the course that aided in the design and implementation of this project.

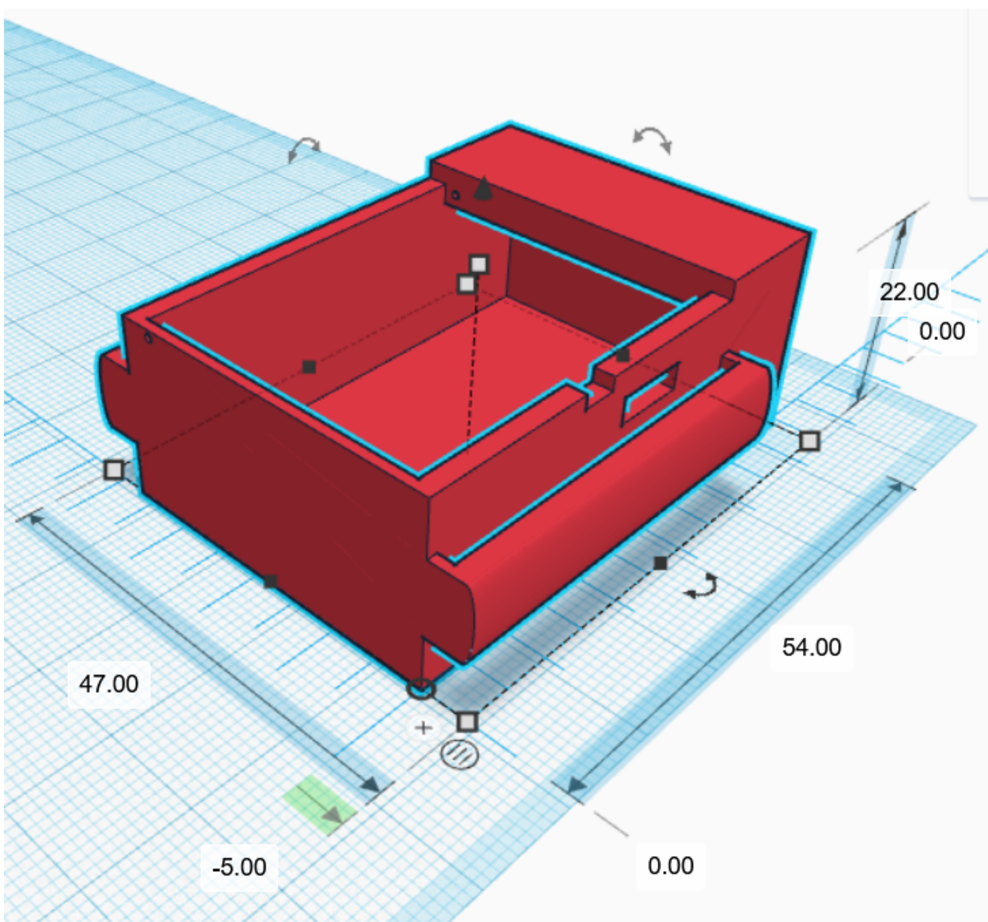
Table B.1: Required knowledge and skills

Course	Knowledge/Skills
COCS 1336	Introductory C++ programming
EENG 2301	Introductory MATLAB and Python Programming
EENG 4307	Fundamentals of Microprocessors Structure and Assembly Programming
EENG 4311	Fundamentals of Signal Processing
EENG 4312	Fundamentals of Communication Theory
EENG 4350	Fundamentals of Biosensors and Advanced Bio-signal Processing
EENG 4351	Introduction to Internet of Things and Device to Device Communication

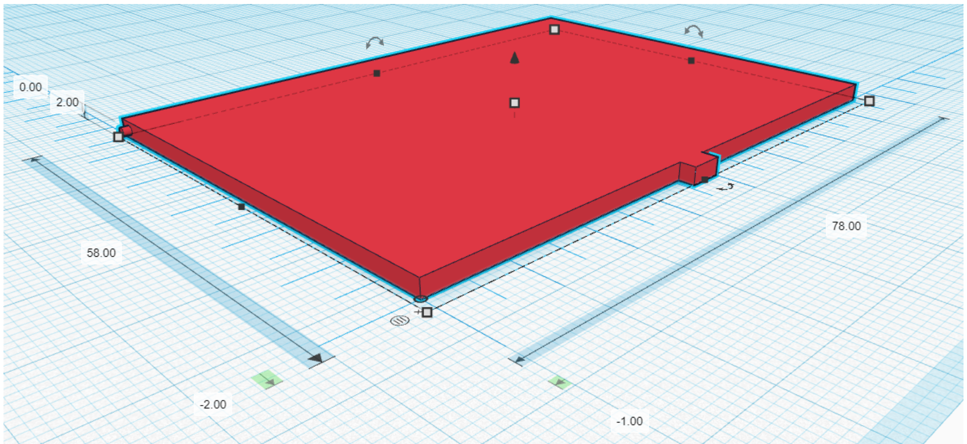
Appendix C: Housing Dimensions for Devices

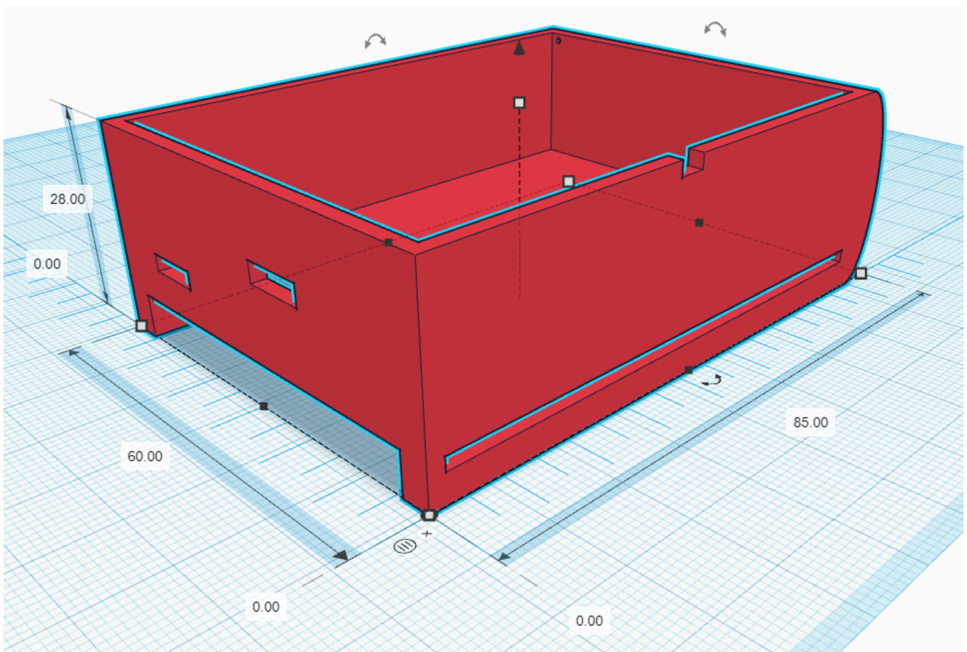
Monitoring Device Housing Dimensions (in Millimeters):





Processing Device Housing Dimensions (in Millimeters):





Appendix D: Feature Extraction for Machine Learning

In order to obtain a feature as an input for the machine learning, sliding window is employed on the time series. On each window of the time series, statistical features: mean and variance were calculated. Furthermore, analytic signals were calculated for each axis of accelerometer signal using Hilbert Transformation. Instantaneous amplitude of the accelerometer signal was calculated using the calculated analytic signal. If x an accelerometer signal, and $\hat{x}$ is the Hilbert transform of the signal, instantaneous amplitude of the signal is given as:

$$a(t) = \sqrt{x^2 + \hat{x}^2}$$

It turns out that $a(t)$ has a long-tailed distribution. Gamma distribution, $\Gamma(\alpha, \beta)$ can be fitted to the distribution of $a(t)$, where α and β are shape and scale parameters respectively. These parameters are used as features for machine learning. Table C.1 shows the modality and the respective features calculated from a five-minute window of the signal. Figure D.1 shows the input-output diagram of the feature extraction routine.

Table D.1: Modality and corresponding features calculated

Modality	**Features**
Acceleration	Mean, Standard Deviation, Shape, Scale
Heart Rate	Mean, Standard Deviation

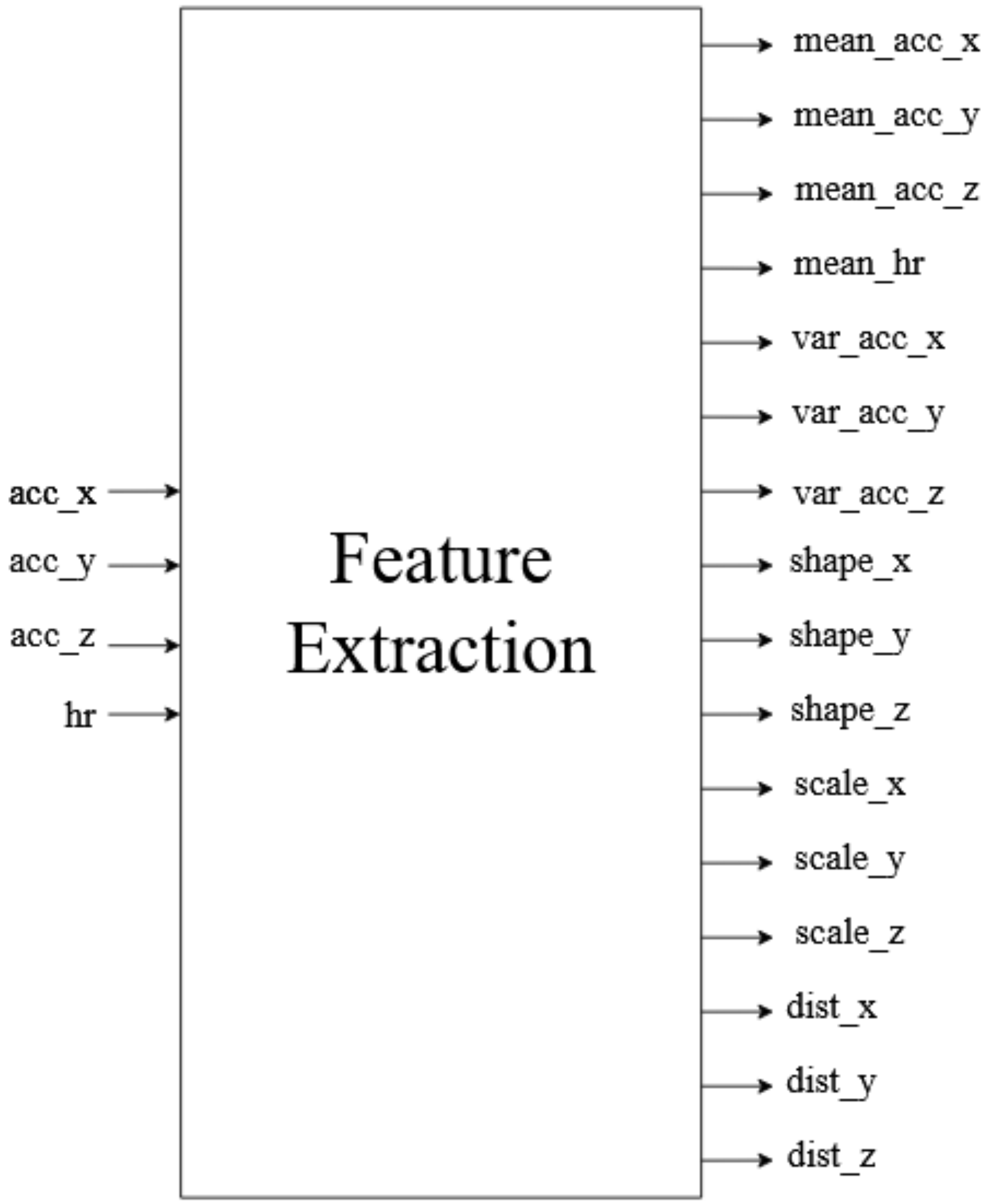


Figure D.1: I/O diagram of feature extraction routine

Appendix E: MATLAB Script for Subsystem Simulation

```
clc;
clear;
close all;

%% clock speeds for Arm M7 and Apple M1 Chip in MHz

am1 = 3200; %apple m1
m7 = 600; %arm cortex m7

%% Dummy data setup
fs_acc = 32; %sampling frequency of accelerometer
N_acc = fs_acc*5*60;

fs_hr = 1; %sampling frequency of heart rate
N_hr = fs_hr*5*60;

t_acc = linspace(0,20,N_acc);
t_hr = linspace(0,20,N_hr);

t_diff = []; %accumulator for run-times


for counter = 1:500


    % Generating dummy data
    accx = sin(2*pi*1000*t_acc) + sin(2*pi*10*t_acc) + sin(2*pi*10*t_acc + 
pi) + sin(2*pi*17*t_acc + 2.1*pi) + rand(1,length(t_acc));
    accy = accx + rand(1,length(t_acc));
    accz = accx + rand(1,length(t_acc));

    hr = sin(2*pi*1000*t_hr) + sin(2*pi*10*t_hr) + sin(2*pi*10*t_hr + pi) 
+ sin(2*pi*17*t_hr + 2.1*pi) + rand(1,length(t_hr));

    start = tic;

    % feature extraction
    calc_features(accx, accy, accz, hr);
    t_diff = [t_diff, toc(start)];

end

hist(t_diff, 40);
title("Run time histogram for feature extraction routine in Apple's M1 
Processor");
xlabel("Time (sec)");
ylabel("Number of items on the bin");

disp("Mean and std of run time Apple's M1 Processor");
disp("mean: "); disp(mean(t_diff*1000));
disp("std: "); disp(std(t_diff*1000));
```

```
figure();
hist(t_diff*am1/m7, 40);
title("Run time histogram for feature extraction routine in ARM Cortex M7
Processor");
xlabel("Time (sec)");
ylabel("Number of items on the bin");

disp("Mean and std of run time ARM Cortex M7 Processor");
disp("mean: "); disp(mean(t_diff*1000*am1/m7));
disp("std: "); disp(std(t_diff*1000*am1/m7));

function [] = calc_features(accx, accy, accz, hr)
%% Calculates features for machine learning
    mean_accx = mean(accx); % Feature 1
    mean_accy = mean(accy); % Feature 2
    mean_accz = mean(accz); % Feature 3
    mean_hr = mean(hr); % Feature 4

    std_accx = var(accx); % Feature 5
    std_accy = var(accy); % Feature 6
    std_accz = var(accz); % Feature 7
    std_hr = var(hr); % Feature 8

    abs_accx = abs(hilbert(accx - mean_accx));
    abs_accy = abs(hilbert(accy - mean_accy));
    abs_accz = abs(hilbert(accz - mean_accz));

    shape_scl_x = gamfit(abs_accx);
    shape_scl_y = gamfit(abs_accy);
    shape_scl_z = gamfit(abs_accz);

    shape_x = shape_scl_x(1); % Feature 9
    shape_y = shape_scl_y(1); % Feature 10
    shape_z = shape_scl_z(1); % Feature 11

    scale_x = shape_scl_x(2); % Feature 12
    scale_y = shape_scl_y(2); % Feature 13
    scale_z = shape_scl_z(2); % Feature 14

    dist_x = sqrt(shape_x.^2 + scale_x.^2); % Feature 15
    dist_y = sqrt(shape_y.^2 + scale_y.^2); % Feature 16
    dist_z = sqrt(shape_z.^2 + scale_z.^2); % Feature 17

end
```