MIDDLE EAST TECHNICAL UNIVERSITY
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

FINAL REPORT
DESIGN STUDIO 9
CONTROLVID-19

ÇAĞDAŞ YARDIMCI	2232965
EGE EKİN CANSIZ	2231561
GÜRAY ÖZGÜR	2167054
KAYA MERT POLATKAYA	2396059
UTKUCAN GENÇ	2331130

PROJECT INITIALIZATION DATE:	20.11.2020
COMPLETION DATE:	11.06.2021
DURATION:	7 MONTHS
BUDGET:	$163

11.06.2021

Contents

1 EXECUTIVE SUMMARY

COVID-19 has a death toll over 3.7 million deaths as of June 2021 and the spread of COVID-19 continues. Even though there are vaccines available, there is a long way to achieve herd immunity. Furthermore, experts report that vaccinated individuals need to continue wearing masks. Therefore, wearing masks is as crucial as ever. This is especially important for groups of people who need to work together in a closed space. Moreover, people who show COVID-19 symptoms increases the chance of spreading the virus in closed spaces.

As Zeyrek, our main mission is to develop technologies for healthcare industry. Due to the COVID-19 pandemic, our mission transforms to developing healthcare related technologies for lessening the impact of pandemic. Our team proposes CONTROLVID-19, a software focused system that makes extensive use of image processing algorithms. This system will allow authorities to automatize the process. With this system, authorities can prevent people without mask from entering the buildings, they can check for possible symptoms or even whether they are wearing their masks properly or not. Solution consists of three subsystems that communicate with each other. These subsystems are Surveillance Subsystem, Temperature Measurement Subsystem, and Lock-Passage Subsystem. Surveillance Subsystem looks for people and checks whether they are wearing a face mask or not. It also checks whether they are wearing their masks properly if they have a face mask on their faces. Temperature Measurement Subsystem measures the body temperature of a person from their hand. It lets people inside if they do not have a high fever and if they wear their masks correctly. Otherwise, it does not allow them to pass through the door. Lock-Passage Subsystem opens the door when someone satisfying the conditions trying to pass through the door. It checks whether that person has passed through the door and increases the count of people inside the building so that nobody passes when the building is too crowded.

The system presented here is not only accurate but also fast to provide a satisfying experience for users. This product has a cost of $163. We will provide operand box with self-cooling system, board computer, camera module, temperature measurement box, a smart passage system and built-in graphical user interface. As our company's business model states, our company's revenue streams are fairly dependent on selling the products that we have developed. Therefore, we will manufacture our product in large numbers in order to increase the revenue streams and profit margin of the product.

In this report, implementations for these subsystems are presented with detailed explanations of the solutions for each subsystem. Furthermore, interactions between these subsystems are shown with diagrams. Requirements are introduced for each subsystem and the subsystems are tested with these requirements in mind. Compliance between and design decisions and tests are stated. Procedures applied for these tests are presented and results of these test are discussed. In addition to user's manual, any potential safety and environmental issues are mentioned, and possible impacts of the product on society are discussed.

2 INTRODUCTION

Authorities impose restrictions to fight against the ongoing COVID-19 pandemic. These restrictions include forcing individuals to wear mask indoors where they gather in groups and isolating individuals infected with COVID-19. Authorities also require businesses to comply with the restrictions and enforce them on their customers. For example, businesses are not allowed to let customers inside if they have a high fever or if there are too many people inside. Businesses currently employ their workers to check their customers. However, that exposes these employees to COVID-19 and it is risky for them.

Zeyrek proposes CONTROLVID-19, a Surveillance-Protective Measure System to assist the authorities in this regard. Right now, the project is finished and it is compliant with the requirements. Moreover, Zeyrek demonstrated the project to the group DS coordinator and is ready for the final presentation and project fair. In this report, not only the system but also the subsystems in detail are explained, whose tests are performed and the results of these tests are presented. It also touches on cost and power analysis, and safety and environmental issues. A detailed report organization can be found below with hyperlinks:

- Section 3 shows the company organization and introduces company members.
- Section 4 describes the subsystems and the overall system. It states the requirements for each subsystem and shows the flowcharts. Section 4.1 describes the project as a whole system. General requirements of the system is stated and overall system operation is explained. Block diagram of the overall system and system level flowchart decision logic is posted. Section 4.2 explains different subsystems of the project in detail, these subsystems include Surveillance Subsystem, Temperature Measurement Subsystem and Lock-Passage Subsystem. In Section 4.2.1, Surveillance Subsystem is explained. Subsystem's requirements are stated. Neural network architecture behind the subsystem is explained to a degree and illustrated with the flowchart of the neural network model. In Section 4.2.2, Temperature Measurement Subsystem is explained. Subsystem's requirements are stated. Temperature measurement with the sensor is explained and important concepts are pointed out. In Section 4.2.3, Lock-Passage Subsystem is explained. Subsystem's requirements are stated. Door lock mechanism is described and illustrated with a figure. Section 4.3 analyses the compatibility and integration of subsystems. Various communication protocols and connections for two boards are discussed. Then, communication and connection types are chosen while reasons are stated.
- Section 5 shows results of various performance tests and analyses results for each subsystem. In Section 5.1, Surveillance Subsystem tests are examined. First test is Training Validation Test, where trained neural network models are validated by test data. Trained model's loss and accuracy is shown by a graph. Second test is Real Time Operation Accuracy Test, where the subsystem is tested for different illumination. Third test is Angle and Distance Test, where the limits of the system is tested.

Fourth test is Frame Rate Test, where detection speed for each case is tested. All results are presented by tables. In Section 5.2, Temperature Measurement Subsystem tests are examined. Temperature is measured by sensor for different subjects, then measurement results are compared with real temperature values. Compared values are demonstrated by a histogram. Error is obtained by fitting the histogram to a statistical model. In Section 5.3, Lock Passage Subsystem tests are examined. Mechanical operation of this subsystem is tested. In Section 5.4, overall system is examined. Overall system's functional operation is tested by applying every possible scenario to the overall system.

- Section 6 is where a list of deliverables is presented. This list includes the delivered hardware and software of the product, services given to the customers and the user manual of the system.
- Section 7 is where resource management of the system is presented. Resource management includes the cost breakdown and the power management analysis.
- Section 8 is where various discussions concerning the product is made. The discussion topics include the safety issues regarding the design, possible applications of the project and the possible effects of the product on the environment.
- Section 9 includes the conclusion of this report.
- Section 10 is references.
- Section 11 is appendices.

3 COMPANY ORGANIZATION

The organizational structure of Zeyrek and company roles for each member is given in Figure 1.

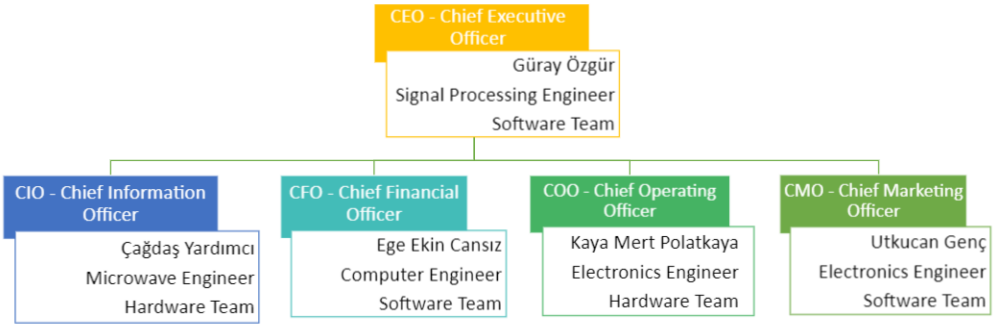

Figure 1: The organizational chart of company

ÇAĞDAŞ YARDIMCI

He is the CIO of Zeyrek. He is a senior year student at Electrical and Electronics Engineering. He specializes in microwave and communications engineering. He had done an internship on distribution transformers in Emek Elektrik Endüstrisi A.Ş. In this project, his strengths are a strong hardware knowledge and his social skills that can strengthen the bonds of the team.

EGE EKİN CANSIZ

He is the CFO of Zeyrek. He is a senior year student at Electrical and Electronics Engineering. He have chosen computer specialization field in his department. He had done internships on embedded systems and biomedical devices at ODTU Technopolis and Bursa. Due to his knowledge about embedded system design and sensors used in biomedical applications, he supports his team in both hardware and software related jobs.

GÜRAY ÖZGÜR

He is the CEO of Zeyrek. He had an important role of the foundation of the company. He is now a senior year student at both Electrical and Electronics Engineering and Mathematics. At the third year of his studies, he attended an exchange programme and has been in Korea, at KAIST, for one semester. He had a work experience at Darkblue Telecommunications and Kuartis at ODTU Technopolis as an Embedded Systems and Machine Learning Intern, respectively. He is interested in the field of signal processing and now searching for an opportunity to continue his master studies in Germany. His theoretical knowledge is his real power, and he is an outside of the box thinker. He loves combining the theory and the application.

KAYA MERT POLATKAYA

He is the COO of Zeyrek. He is a senior year student taking courses from electronics and microwave areas. He had participated in an internship program at MilSOFT. He had worked part-time in Mikrotasarım, a novel ROIC company for infrared imagers. He is currently working in Mikrosens, a low-cost thermal imager company. He is interested in directing his career towards analog integrated circuit and mixed-signal circuit design. He is currently planning on continuing his graduate studies in Europe or USA. He is really experienced in the field because of his work experience. He is a product maker.

UTKUCAN GENÇ

He is the CMO of Zeyrek. He is a senior year student studying Electrical and Electronics Engineering in METU. He has work experience in ASELSAN and MKE as a Test Engineer. Currently, he is working as a part-time Project Engineer in ASELSAN. He chose biomedical engineering and electronics as his specialization fields. He is interested in Python programming, finance and economics. He is a very skilled person when it comes to coding.

4 TOP-DOWN SYSTEM DESCRIPTION

CONTROLVID-19 is a smart mask detection and monitoring system including substantial subsystems that provide system-level requirements in an ordered way. The overall system is aimed to detect the mask and the fever, identify and recognize possible unwanted situations, deny unmasked people, and give warnings when necessary. That's why, all the subsystems that are present are created such that they can satisfy all of the needs coming from the functional requirements and constraints coming from the system itself. These system features and requirements are included to increase the effectiveness of prevention of COVID-19.

4.1 SYSTEM LEVEL DESCRIPTION

Since there are a lot of requirements needed to be performed for fully complete the system, several subsystems are created in order to achieve all of the operations, namely Surveillance Subsystem, Temperature Measurement Subsystem and Lock-Passage Subsystem. Overall description of the system with a block diagram including all the subsystems and interactions between them is given in Figure 2. The detailed explanations of before mentioned subsystems are given in Section 4.2. There are general requirements of the system as well as subsystem requirements of each subsystem. Subsystem requirements are specified in Section 4.2. General requirements of the system is listed below.

General requirements of the system:

- It will deny entrance to risk-carrying people whose temperature is higher than $37.5°C$.
- It will deny entrance to people without masks.
- It will deny entrance to people without properly worn masks.
- It will give entrance to non-risk-carrying people if building is not highly populated.
- It will deny entrance if building is highly populated.
- It will monitor the gathered data to provide a good user-experience.

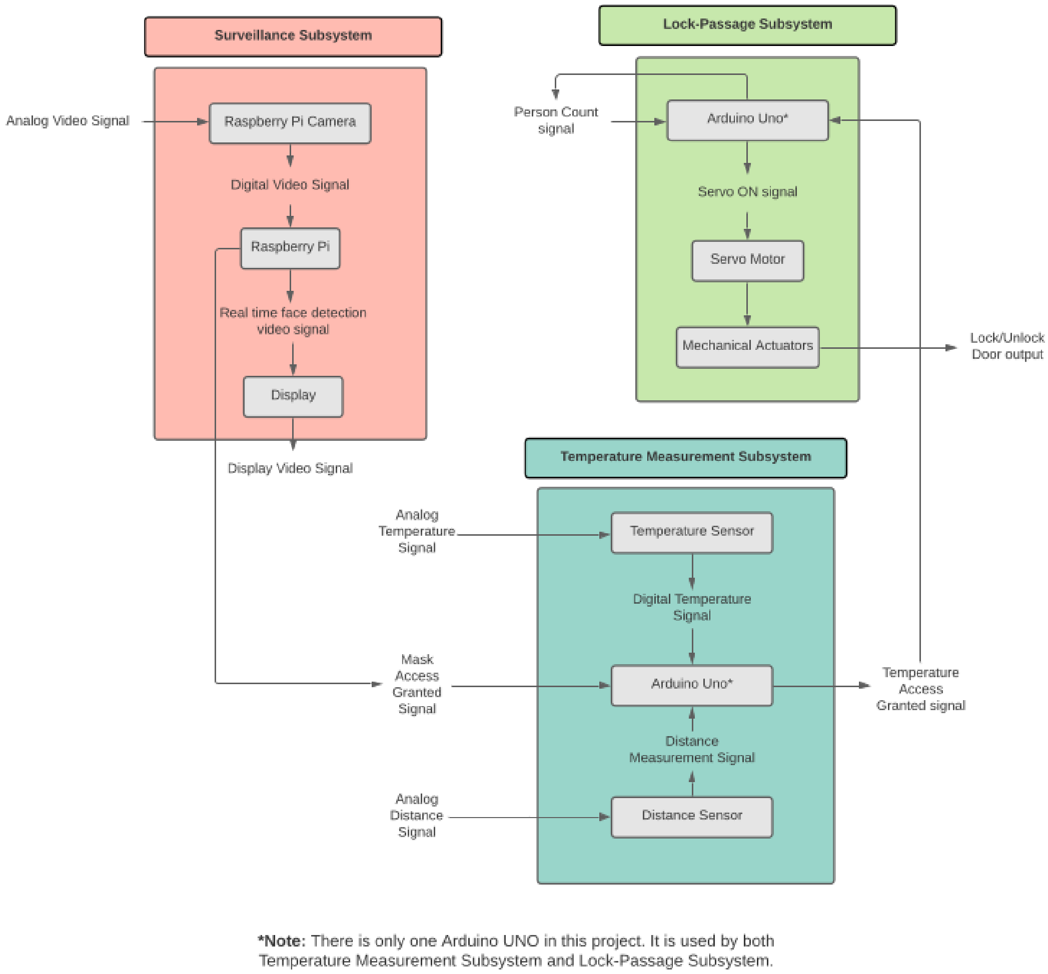

Figure 2: Block diagram of the overall system

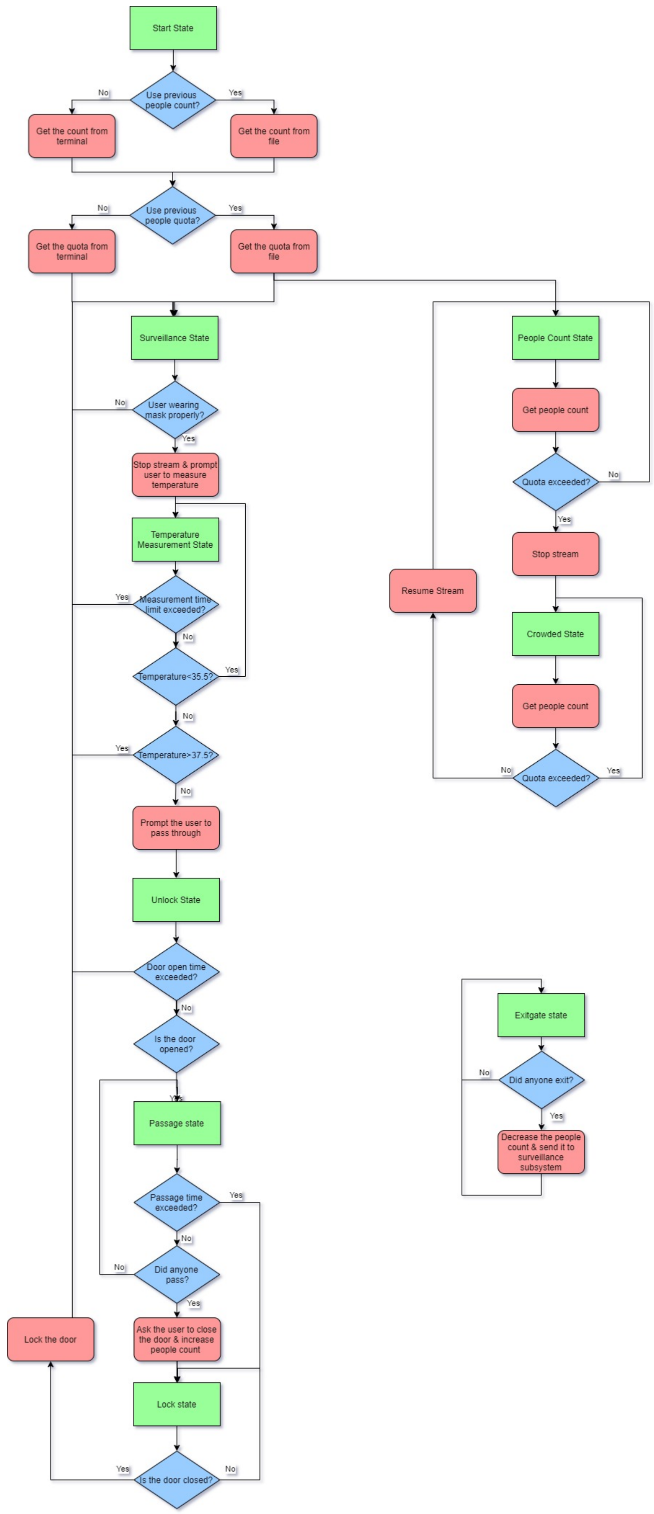

Figure 3: System level flowchart of main decision logic

System operation can be written as follows and a functional flow diagram is as depicted in Figure 3.

System operation:

- It checks whether the capacity is exceeded or not.
- If it detects a person, it checks whether that person is wearing a mask or not.
- If that person is not wearing a mask, the system warns that person to wear a face mask.
- If that person is wearing a face mask, the system checks whether the mask is worn properly or not.
- If the mask is not worn properly, it warns that person to adjust their mask.
- If that person is wearing a face mask properly, the system checks for temperature measurement.
- If the time limit is not exceeded, it checks whether the person has high fever or not.
- If the system detects a high fever, it does not grant enterance.
- If it does not detect a high fever, it unlocks the door.
- If the time limit is not exceeded, it checks whether the person has entered the building or not.
- If the person has entered the building, it increases the count of people in the building and locks the door.
- If anyone has entered left building, it decreases the people count.

4.2 SUBSYSTEM LEVEL DESCRIPTION

Technical drawings, in Appendix A, and real appearances, in Appendix B, of all subsystems are presented in related Appendix, it is not necessary; however, it would be better to look at them to have a better understanding before proceeding.

4.2.1 SURVEILLANCE SUBSYSTEM

The requirements of Surveillance Subsystem is listed as follows:

- **Functional Requirements:**
  - Subsystem must detect human faces.
  - Subsystem must check whether user is wearing a mask.
  - Subsystem must check whether user is wearing a mask properly.
  - Subsystem must show the result of the mask evaluation through a display.
  - Subsystem must exchange information with Temperature Measurement Subsystem.
- **Performance Requirements:**
  - Subsystem must detect properly worn mask at most in 2 seconds.
  - Subsystem must detect properly worn mask with 90% accuracy.
  - Required detections must be done while the subject and the camera has a distance of at least 30 cm and at most 100 cm between them.
  - Required detections must be done while the subject is at the sight of the camera.
  - Required detections must be done while the subject is facing the camera with an angle of at least 10°.
- **Physical Requirements:**
  - Subsystem needs to be physically robust in case of a slight damage.
  - Subsystem must be installable and compatible on any wall surface.

As seen from the requirements, it is the very critical subsystem of the system design. It deals with the one of the most important parts of the problem, i.e. mask detection. Requirements are highly correlated of the requirements of the overall system.

Surveillance Subsystem is designed to check if the user is wearing a mask and if the user is wearing their mask properly. Furthermore, the subsystem should work accurately for different elevations as the users will range from children to elderly. To minimize the possibility of inaccuracy in this subsystem, the subsystem will take several samples and compare them to check the mask usage. If the results of the compared samples are not similar, the system will not grant entrance for the specific sample set and the sample size will be increased temporarily to further decrease the possibility of error.

The subsystem consists of a Raspberry Pi 4B 4 GB, a 3.5 inch LCD screen, a cooling system and a camera module. This subsystem is the core part of the system whose 3D drawings and real-life appearance is available in Appendix A.1 and B.1. Board computer provides whole communication between the other subsystems and processes the real-time image obtained by the camera module. Real-time image and other necessary parameters are shown to the user via the LCD screen. Haar Cascade classifiers are used for face and other facial landmarks including nose. After the face is detected with the Haar Cascade classifies the bounding box is cropped and fed into the neural network model. Haar Cascade as it was the fastest face detection algorithm the team has tried, is used to ensure fast operation of the subsystem and fulfill the requirement concerning frame rate and detection speed. MobileNetV2 neural network architecture is chosen because the architecture is optimized for usage in embedded systems. The architecture is based on an inverted residual structure where the residual connections are between the bottleneck layers. The intermediate expansion layer uses lightweight depth-wise convolutions to filter features as a source of non-linearity. The architecture of MobileNetV2 contains the initial fully convolution layer with 32 filters, followed by 19 residual bottleneck layers [1]. Block diagram of the convolutional neural network is given in Figure 4. Description of these layers are given in Figure 5. Then, the model is trained by using a synthetically generated dataset named MaskedFace-Net created by Cabani et al [2].

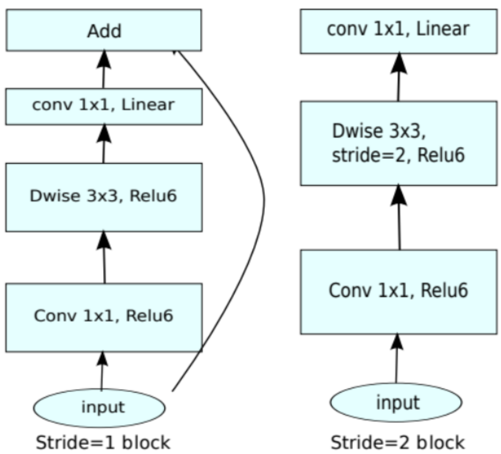

Figure 4: Block diagram of MobileNetV2 architecture

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

Figure 5: Overall architecture of MobileNetV2

Operation starts with the acquisition of the video stream via the mounted camera. The video is then streamed back with the help of the LCD screen. The subsystem first gets the location of the face by using Haar Cascade classifiers. When the face is detected during the stream a rectangular area in the vicinity is cropped and fed to the neural network model for classification. After the classification is made percentages for three different classifications made are gathered. Classifications include "Proper Mask", "Wrong Mask", "No Mask". Their examples are shown in Figure 6.

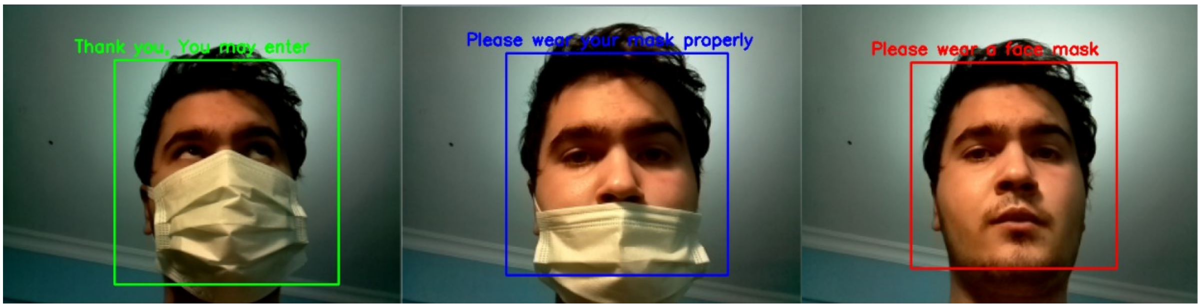

Figure 6: Three different classifications of the neural network model

The highest percentage is accepted as the true classification. The system checks for three consecutive worn mask case and grants entrance accordingly. If this is the case the stream is stopped to conserve power and avoid confusion by the user and "Measure temperature & Pass" line is displayed on the screen. This subsystem is also responsible for the initialization of the code. While the code is starting the terminal prompts the user with the lines shown in the Figure 7.

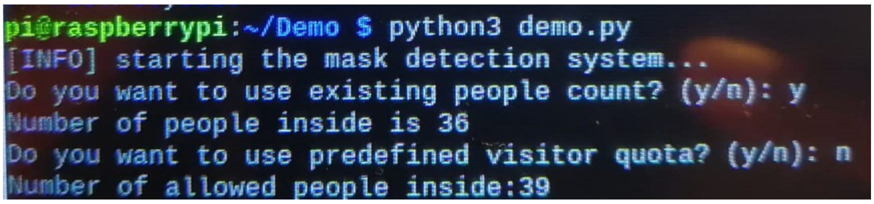

Figure 7: Terminal prompts of the surveillance code during initialization

In the event of a possible power outage, the system should hold the number of people within the building. To do this number of people within the building as well as the people quota is kept within a file which is updated regularly. While starting the program from the terminal, the code asks whether the user wants to use the previous number of people and previous people quota or not, and lets them input the selected parameters if necessary. The subsystem constantly gets the people count from Lock-Passage Subsystem and updates the people count within it. The last operation of the subsystem is to stop streaming if the number of people within the building is equal to the people quota. The subsystem regularly checks Lock-Passage Subsystem for occupancy to achieve this.

Figure 8: Three different classifications of the neural network model

The flowchart of the subsystem illustrating the operations explained is given in Figure 8. The explanation of the subsystem operation with relation to the flowchart is as follows. The subsystem first gets the previous people count and people quota either from the file or from the terminal depending on the user inputs. Then the operation starts. Two different branches run in parallel. This is because the subsystem should not operate when the people quota is exceeded. Hence, the subsystem constantly gets the number of people inside the building from Lock-Passage Subsystem and checks if the quota is exceeded. If the quota is exceeded the stream is stopped and the subsystem does not check for mask.

Moving on to the main operation, the subsystem first checks if there is any face detected by the face detection algorithm, which is Haar Cascade. If so, the rectangular area in the vicinity of the detected face is cropped and fed into the neural network model. If the result coming out of the model points to correct mask usage for three consecutive checks then the subsystem directs the operation to temperature measurement subsystem. Three consecutive checks are used to have a balance between detection speed and detection accuracy. Checking the model for three times increases the accuracy significantly, while the drawbacks in operation speed are hardly recognizable. "confidence" signal is used to check for the three consecutive mask case. When the subsystem decides that the user is wearing their mask appropriately, the subsystem sends out the signal "maskflag=1" to Arduino UNO, stops the stream, prompts the user to check their temperature and waits for the completion of the operation. When Arduino sends "done" signal back, the subsystem sends out "maskflag=1" signal, resumes stream and resumes its operation from the start.

The crosscheck table for requirements of Surveillance Subsystem is presented in Table 1, which illustrates how well each equipment satisfies the requirements.

Table 1: Crosscheck table for Surveillance Subsystem requirements

	MobileNetv2	Haar Cascade	LCD	USB Comm.	RPi Camera	Case
Subsystem must detect human faces		✓			✓	
Subsystem must check whether user is wearing a mask	✓				✓	
Subsystem must check whether user is wearing a mask properly	✓				✓	
Subsystem must show the result of the mask evaluation through a display			✓			
Subsystem must exchange information with other subsystems				✓		
Subsystem must detect properly worn mask at most in 2 seconds	1 sec					
Subsystem must detect properly worn mask with 90% accuracy	> 90%					
Detections must be done while subject and the camera has a distance x of 30cm < x < 100cm	✓	✓			✓	
Required detections must be done while subject is at the sight of the camera	✓	✓			✓	
Required detections must be done while while subject is facing the camera with an angle of at least 10°	✓	✓			✓	
Subsystem needs to be physically robust in case of a slight damage						✓
Subsystem needs to be installable and compatible on any wall surface						✓

4.2.2 TEMPERATURE MEASUREMENT SUBSYSTEM

The requirements of this subsystem is listed as follows:

- **Functional Requirements:**
  - Subsystem must check for temperature from hand.
  - Temperature must be detected with no contact.
  - Subsystem must check for distance.
  - Subsystem must show the measurement results through a display.
  - Subsystem must exchange information with both Surveillance Subsystem and Lock-Passage Subsystem.
- **Performance Requirements:**
  - Temperature must be detected with 0.5°C error at most.
  - Distance between temperature sensor and subject's hand needs to be at least 2cm and at most 3cm.
  - Temperature detection must be done in at most 1 seconds.
- **Physical Requirements:**
  - Subsystem needs to be physically robust in case of a slight damage.
  - Subsystem must be on waist level for hand temperature measurement.

The requirements are necessary to realize the requirement of the system, which is "It will check for fever and if it detects fever, it will deny entry and give a warning."

The subsystem must operate without the need of contact from the user. Reason is that the system is created to check if the user is infected or not and having infected people touch the sensor negates the meaning of creating a fever detection system. For this purpose, a non-contact temperature sensor is used for Temperature Measurement Subsystem whose 3D drawings and real-life appearance is available in Appendix A.2 and B.2. Usually, lenses are used to increase the detectivity of such sensors as they limit the field of view of the sensor and focus on the infrared object. However, due to cost considerations, another method will be used to improve the accuracy of the sensor, which is using an ultrasonic distance sensor. The ultrasonic distance sensor is used to characterize the performance of the temperature sensor with respect to distance. Need for this characterization arises from the operation principle of the temperature sensor.

Temperature sensor operates by averaging the voltage output coming from the FPA (Focal Plane Array). As it is averaged, FOV (Field of View) of the sensor should only cover the observed object which is hand in this case. Hence, the user must hold their hand closer than some threshold distance so that the FPA does not sense the background temperature. If the background temperature is sensed, the object temperature would be underestimated. Hence, the sensor both needs characterization and range. The upper part of the range is coming from the reasons stated before. Lower part of the range is coming from the ultrasonic proximity sensor which can accurately measure the distance no closer than 2 cm. The range is also kept shorter than these upper and lower values because the sensor output is assumed to be linear within the working range and the assumption holds for smaller ranges.

Characterization is done by moving the user's hand toward the sensor and getting the distance and temperature readings. From these readings the temperature variation with respect to the distance is obtained. The temperature of the user at the time, which was $36.5°C$ was divided with these readings to obtain a multiplier vector. Sensor output can be multiplied with this vector to get the real object temperature. However, to keep things simple this vector is estimated with a linear function and used in the Arduino code. The multiplier vector versus time of flight which is related with the distance is given in Figure 9. The data is then processed in Arduino Uno R3, which gives information to 2x16 LCD screen.

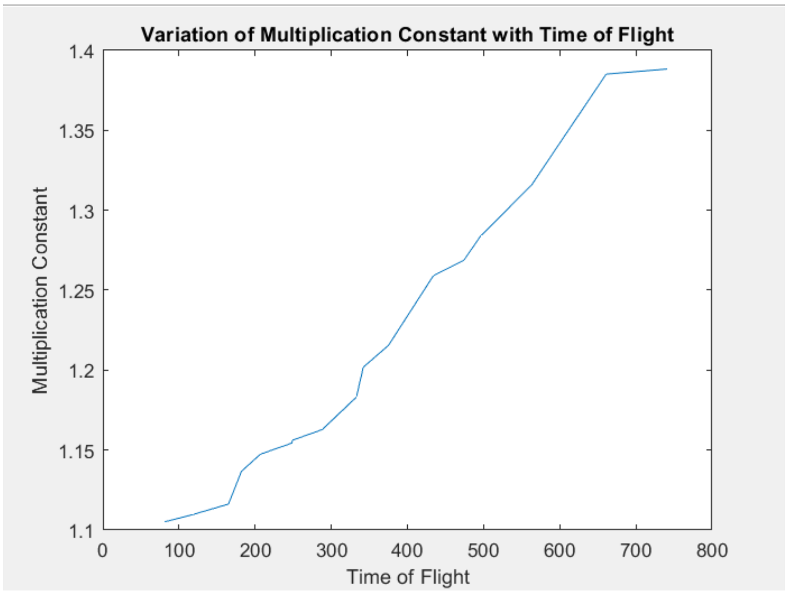

Figure 9: Multiplier vector versus time of flight

Furthermore, the signal flowchart is added in Figure 10 for clarification of the working mechanism of the subsystem.

zeyrek

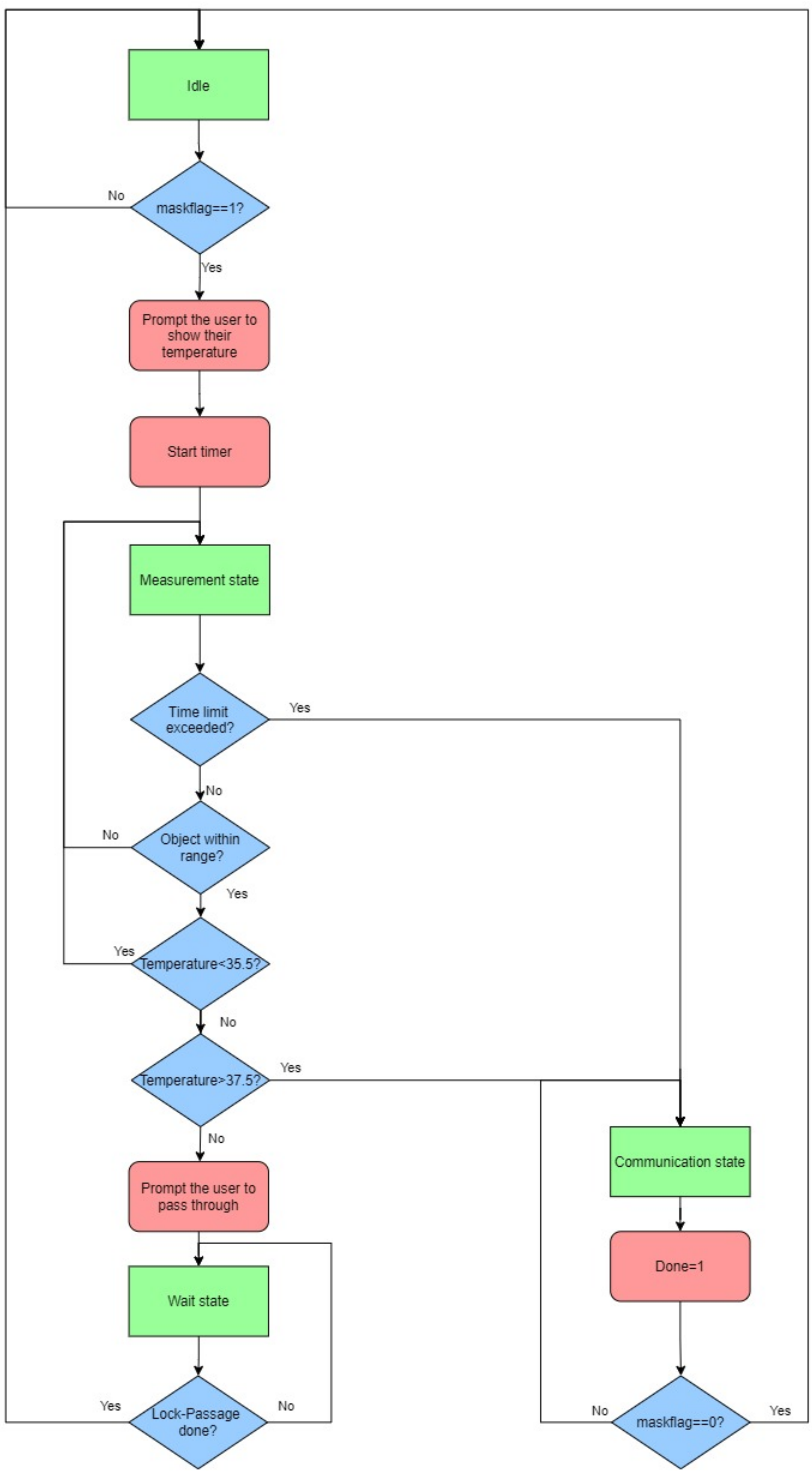

Figure 10: The signal flowchart of Temperature Measurement Subsystem

The explanation of the subsystem operation with relation to the flowchart is as follows. At its initial state the subsystem waits for the "maskflag==1" signal from Surveillance Subsystem. After getting this signal the subsystem prompts the user to show their palm, the subsystem also starts a timer. This timer is used to check how many seconds has passed since the initialization of the subsystem. If the time limit (which is 20 seconds) is exceeded, the subsystem exits the operation and communicates back to Surveillance Subsystem to resume the stream. Within the time limit, the subsystem checks if there is any object within measurement range (which is between 2cm and 3cm). If there is an object within the range, its temperature is measured. If the temperature is lower than 35.5°C, the subsystem returns to the initial state and checks the temperature again. Temperatures lower than 35.5°C are not accepted because a hot external object may be used to fool the system, this range is used to ensure the object within range is close to human body temperature and avoid any bug which may be exploited. However, as temperatures higher than 37.5°C signify high fever. The subsystem immediately exits the operation, and gives warning if a fever is detected.

If the user temperature is within the acceptable range the subsystem prompts the user to pass through and initiates the operation of the Lock-Passage Subsystem. The subsystem then waits for the operation to end to return to its initial state. As a last remark, the operation of the communication state initiates the subsystem's exit from the operation. In this state the subsystem sends out "Done==1" signal to Surveillance Subsystem and waits for the signal "maskflag==0" to exit the operation. This communication flow is used to ensure that there are no errors in communication between the subsystems.

The crosscheck table for requirements of Temperature Measurement Subsystem is presented in Table 2, which illustrates how well each equipment satisfies the requirements.

Table 2: Crosscheck table for Temperature Measurement Subsystem requirements

	Non-contact sensor	Ultrasonic Distance Sensor	LCD	USB Comm.	Case
Subsystem must check for hand temperature	✓				
Temperature must be detected with no contact	✓				
Subsystem must check for distance		✓			
Subsystem must show the measurement results through a display			✓		
Subsystem must exchange information with other subsystems				✓	
Temperature detection must be done in at most 1 seconds.	< 1 sec				
Temperature must be detected with an error of 0.5°C at most.	< 0.33 °C				
Distance between temperature sensor and subject's hand needs to be at least 2cm and at most 3cm		✓			
Subsystem needs to be physically robust in case of a slight damage					✓
Subsystem must be on waist level for wrist temperature measurement					✓

4.2.3 LOCK-PASSAGE SUBSYSTEM

The requirements of this subsystem is listed as follows:

- **Functional Requirements:**
  - Subsystem must unlock the door.
  - Subsystem must lock the door.
  - Subsystem must count the number of people in the building.
  - Subsystem must exchange information with other subsystems.
- **Performance Requirements:**
  - Door needs to be unlocked and locked within at most 3 seconds.
- **Physical Requirements:**
  - Subsystem needs to be physically robust in case of a slight damage.
  - Subsystem needs to give feedback about the passage.

The requirements are necessary to realize the many requirements of the system, since denying or granting entrance is the final result that should be valid for all requirements. For all times, for this subsystem, it is crucial to communicate with other subsystems.

This subsystem consists of a hinge lock controlled by a servo motor and several distance sensors in addition to a magnetic door sensor. Small size and mass of the hinge lock will let us use a relatively smaller, hence cheaper motor for lock mechanism. Here, the hinge lock mechanism is 3D printed and at the same time act as a linear actuator. Passages will be controlled by the data coming from a distance sensor located at the door frame to count the number of people entering the building. The sensor is employed to ensure whether the user enters the building or not. Another distance sensor will be located on the exit door to count the number of people exiting the building. If the number of people in the building exceeds a predetermined threshold, the system will not let people in.

Arduino Uno R3, a servo motor, ultrasonic distance sensors, and a magnetic sensor will be employed for the operation of this subsystem. A linear servo actuator is embodied by ”potentprintables”, which is an open-source 3D printing drawing [4]. Here, some modifications are made and a basic lock mechanism is obtained. It is enough for demonstration, despite it lacks durability. The concept behind its working principle is rather simple. Servo motor runs a pinion gear, then it moves the rack gear, which results in translational movement. The rack and pinion pair is demonstrated in Figure 11. 3D drawings and real-life appearance of Lock-Passage Subsystem is available in Appendix A.3 and B.3.

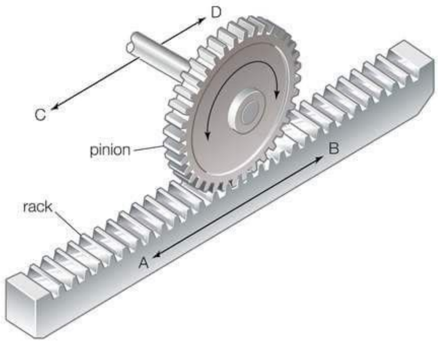

Figure 11: Demonstration of rack and pinion pair employed

The crosscheck table for requirements of Lock Passage Subsystem is presented in Table 3, which illustrates how well each equipment satisfies the requirements.

Table 3: Crosscheck table for Lock Passage Subsystem requirements

	Hinge Lock	Linear Servo Actuator	Arduino	USB Comm.	Case	Indicators
Subsystem must unlock the door	✓	✓				
Subsystem must lock the door	✓	✓				
Subsystem must count the number of people in the building			✓			
Subsystem must exchange information with other subsystems				✓		
Door needs to be unlocked and locked within 3 seconds		✓				
Subsystem needs to be physically robust in case of a slight damage	✓				✓	
Subsystem needs to give feedback about the passage	✓					✓

Furthermore, the signal flowchart of the subsystem is added in Figure 12 for clarification of the working mechanism of the subsystem.

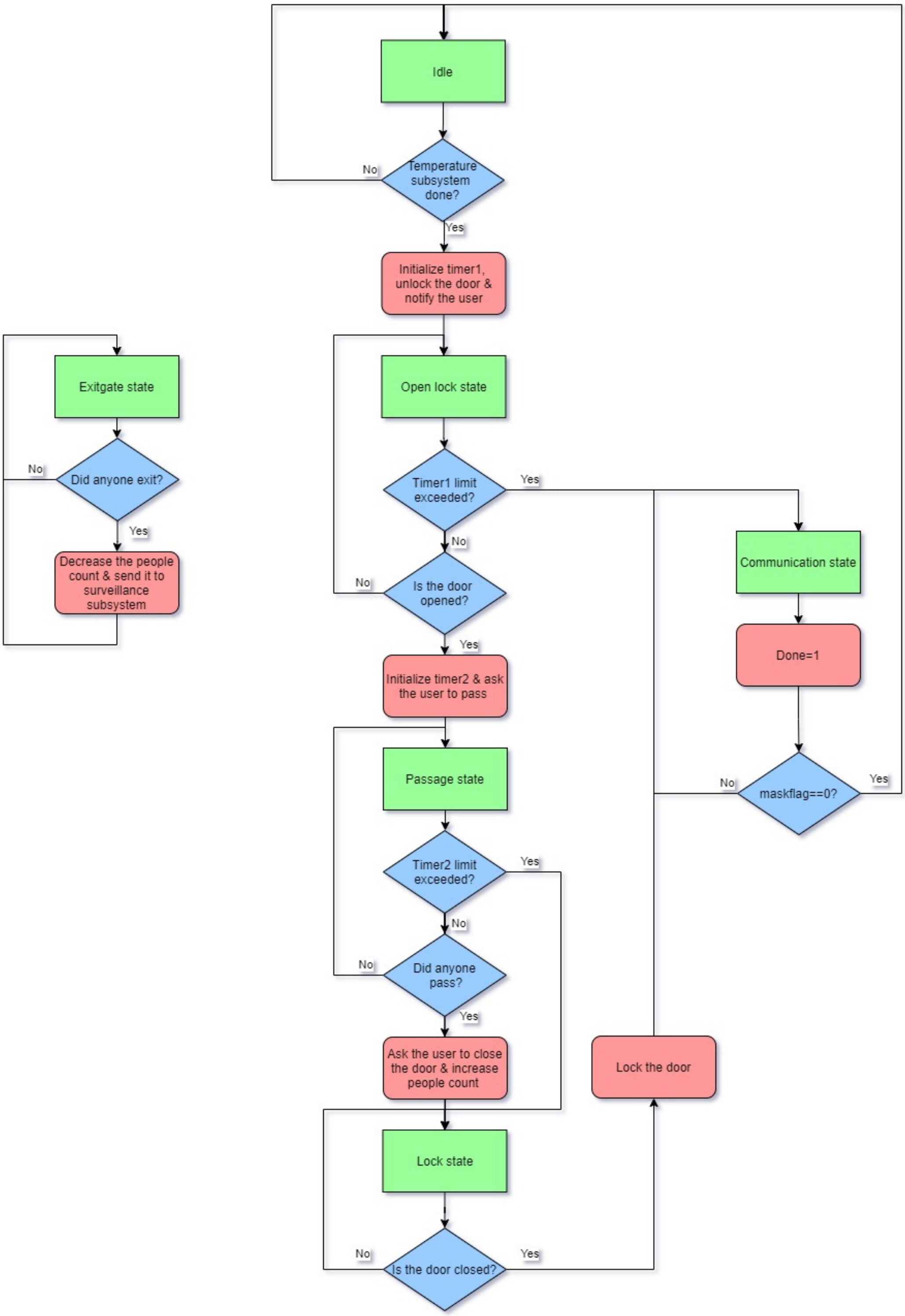

Figure 12: The signal flowchart of Lock-Passage Subsystem

The explanation of the subsystem operation with relation to the flowchart is as follows. Two operations run parallel within the subsystem. The reason for this is the fact that any exit from the building should be checked regardless of which state the system is in. Hence the subsystem always checks the exit for any exit. In the event of an exit the subsystem decreases the people count and prints it via 2x16 LCD screen. It also sends the new people count to Surveillance Subsystem so that stream can continue if the building was crowded before the exit. In its normal operation however, the system waits for Temperature Measurement Subsystem to finish its cycle. When it is done, a timer is initialized and the door is unlocked via actuators. The user is also notified via 2x16 LCD that the door is unlocked. In this state the state of the door, namely "open" or "closed" is checked via a magnetic switch placed on the door frame. The reason for initialization of the timer is to eliminate a possible bug. This bug happens if the user does not open the door when it is unlocked. With the help of the timer the system locks the door back and turn to the mask detection state so that if a user decides that they don't want to enter, the system locks itself up after some time and is not stuck in that state.

If the user opens the door before the time limit is exceeded, then another timer is set and the system gets into the passage state. The user is again notified via 2x16 LCD that they should pass through the door. In this state, the subsystem checks for the passage via the ultrasonic proximity sensor placed on the door. If the distance read by this sensor is below some value, the system decides that the user has passed through the door. The timer in this state is used to eliminate a bug which may arise if the user opens the door but does not pass through it for a long time. The system stops its operation in this case and waits for a personnel or a user to close the door. Because this is a fault scenario, people count is not increased in this case and the system prompts the user of the personnel that the door should be closed. Likewise, in the normal operation when the user passes through the door frame, people count is increased an sent to Surveillance Subsystem. Also, the system prompts that the door should be closed. State of the door is again checked by the magnetic switch placed on the door. After the door is closed the subsystem operation finishes and communicates with Surveillance Subsystem to go back to its initial state. The subsystem sends "Done==1" signal to the surveillance subsystem and waits for it to return with "maskflag==0" signal. After this state, all the subsystems return to their initial state.

4.3 COMPATIBILITY ANALYSIS OF SUBSYSTEMS

In overall, the system consists of two circuit boards, which are an Arduino UNO and a Raspberry Pi 4B. The board computer is thought as the master, and the Arduino UNO is thought as the slave. Since the Arduino UNO operates up to 5 Volts and the board computer operates up to 3.3 Volts; it is wise to feed Arduino UNO with the board computer (voltages higher than 3.3 Volts may cause irreversible damage.). The ground pins of the both boards must be connected in order to have the same reference voltage on both sides [6].

The communication between the Arduino UNO and Raspberry Pi can be ensured by using *UART* (Universal Asynchronous Reception and Transmission) protocol. The company have chosen the UART protocol as it is more commonly used. There are two options of connecting the boards for UART. First option is connecting the Raspberry Pi GPIO ports to Arduino RX/TX ports, where RX port is for receiving the signal and TX port is for transmitting the signal. Second option is using a USB cable, between USB ports of Arduino and Raspberry Pi. Both boards have a USB to TTL (UART) converter inside, which handles the UART conversion process. Since there are voltage differences (3.3V and 5.5V) between Arduino and Raspberry Pi, it will required to have a 3.3V/5V voltage shifter between connected ports in GPIO-RX/TX connection. Therefore, the company have chosen to use a USB cable connection in serial communication. What is more, serial communication libraries are used by both boards (a Python library for Raspberry Pi and a C library for Arduino) to ease the process [7].

The Lock-Passage Subsystem will not contain any circuit boards, it will be controlled by Arduino UNO. Since the computational workload of the board computer is heavy (the fans and the heat sinks may be inefficient if the board computer also drives a servo motor), the servo in the Lock-Passage Subsystem will get the *on/off* signal from the Arduino UNO.

As can be seen from block diagram of the overall system in Figure 2, Surveillance Subsystem will send out a Mask Access Granted Signal to Temperature Measurement Subsystem via USB connection as mentioned and Temperature Measurement Subsystem will send out a Temperature Access Granted Signal to Lock-Passage Subsystem, which is done in Arduino UNO. Then, Lock-Passage Subsystem will unlock the door.

5 RESULTS AND ANALYSES OF PERFORMANCE TESTS

5.1 SURVEILLANCE SUBSYSTEM

5.1.1 TRAINING VALIDATION TEST

First test is done while the model is being trained. The model is tested with a set of images for different epochs. The results is given in Figure 13. As can be seen from the figure, the model yields around 98% validation accuracy which was 90% previously. The epoch size was kept small to avoid accuracy loss due to overfitting. The number of train data is around 7200 images in total.

Figure 13: Model training results

These results certainly cover the performance requirements of the system in terms of accuracy. However, these tests may not fully represent the real time operation accuracy of the system. Hence, further tests to check the real-time accuracy of the system is done.

5.1.2 REAL-TIME OPERATION ACCURACY TEST

The subsystem is tested for different illumination settings in real-time. To test the subsystem, three different settings were used, which are high illumination, low illumination and moderate illumination.

Three different classifications of the model, which are proper mask, wrong mask and no mask are tested for each of these illumination levels. A desk lamp is placed above the test subject as it allows greater emphasis on the facial features of the test subject. Furthermore, it is presumed that the system will operate in a similar setting where the subsystem is placed below a light source. As the angle of the face is another parameter which could decrease the accuracy, the tester has positioned his face at different angles while taking samples. Distance is also another concern for accuracy, and the distance is also varied during the real-time tests. Hence, the subsystem is tested several times for each illumination setting, different angles and varying distance, the tester then recorded the number of true and false results. These results are rounded up on a scale of 10 to quantize the accuracy results. An example of the real-time test is given for illumination in Table 4.

Table 4: Illumination tests

	No Mask	Wrong Mask	Proper Mask
Low Illumination	10	9.2	10
Moderate Illumination	10	9.8	10
High Illumination	10	9.9	10

Results of the tests show that the model is consistently accurate for no mask and proper mask cases. For wrong mask case, an inherent inaccuracy is present which decreases with increasing illumination, distance and angle. This inaccuracy occurs for extreme angles at distances longer than 2.5 meters. When the illumination level is low, the model accepts wrong mask as proper mask when the horizontal and vertical angle are at their extreme. When the illumination is increased however, this error diminishes. In moderate illumination only several points were observed when the vertical angle of the face is at its extreme. As discussed in the previous reports, the method where a single model with three different classification is used to fulfill the speed requirements. The level of inaccuracy which was previously reported shown in Table 5 is not present anymore thanks to another training. Hence, the chosen method proved to be superior in terms of both accuracy and speed.

Table 5: Previous illumination tests

	Correct Mask	Wrong Mask	No Mask
High Illumination	10	10	10
Moderate Illumination	10	10	8
Low Illumination	10	9	5

5.1.3 ANGLE AND DISTANCE TESTS

To determine the limits of the subsystem, angle and distance tests are conducted. Here, limiting factor is detection of faces. Haar Cascade can not detect faces apart from certain angles and distances, since there is not enough facial landmarks for extreme angles and distances.

For distance test, the other factors kept constant. Direct angle and high illumination is preferred. By using a tape measure, the distance of the subject to camera is measured while increasing distance. After 3 meters, face detection failed. Since 1 meter is selected as the maximum operation distance, and the range is more than enough, accuracy test is not conducted for extreme distances. The effect of the distance on accuracy is negligible in the operating range.

For angle test, the other factors kept constant. 50 cm distance and high illumination is preferred. By using a protractor and a ruler, the position of the face is arranged. Angle is varied according to four different positions, right, left, up, and down. For each mask configuration and direction, the subject increased the angle of his head, and measurements are taken. Maximum acceptable values for angles in each case is noted. The results are obtained as in Table 6.

Table 6: Angle tests

	No Mask	Wrong Mask	Proper Mask
Right Direction	17°	15°	16°
Left Direction	17°	15°	16°
Down Direction	20°	18°	20°
Up Direction	16°	15°	16°

Since an angle of 10° is selected as the maximum operation angle, and the range is more than enough, accuracy test is not conducted for extreme angles. The effect of the angle on accuracy is negligible in the operating range.

5.1.4 FRAME RATE TEST

As the speed of operation is important considering the user experience and detection speed, the tests would be incomplete without a frame rate test. For this test, a new code was generated which counts the number of frames in 60 seconds. To ensure that frame rate does not change for different classifications, the tester has done the frame rate test for four different cases. Namely, "no face" where there is no face within the field of view, "bare face" where the present face is not covered by a mask, "wrong mask" where the present mask does not cover the face properly and "proper mask" where the present mask does cover the face properly. The results of the test were given in Table 7 where number of frames for each case and resulting frame rate for them are presented.

Table 7: Frame rate table

	No Face	Bare Face	Wrong Mask	Proper Mask
Frames per second (fps)	2.53	2.50	2.53	2.55
Number of frames in 60 seconds	152	150	152	153

As the table illustrates, the results for each case are quite similar. The average FPS is around 2.53, which is very close to the frame rate requirement. Another requirement was that the system should detect a masked face accurately within 2 seconds. For our system, 5 frames are present in a 2 second time interval. The system checks for the mask three consecutive times to verify it. The detection time is around 1.2 seconds, which is within the required detection time. As stated before, majority of the design decisions made for the Surveillance Subsystem focused on ensuring fast operation. Hence, this result is as fast as the system can operate with present cost considerations and hardware restrictions.

Table 8: Crosscheck table for tests in Surveillance Subsystem

	Training Accuracy Test	Illumination Test	Angle Test	Distance Test	Frame Rate Test	Integration Test
Subsystem must detect human faces		✓	✓	✓	✓	
Subsystem must check whether user is wearing a mask		✓	✓	✓	✓	
Subsystem must check whether user is wearing a mask properly		✓	✓	✓	✓	
Subsystem must show the result of the mask evaluation through a display		✓	✓	✓	✓	
Subsystem must exchange information with other subsystems						✓
Subsystem must detect properly worn mask at most in 2 seconds		1.2 sec	1.2 sec	1.2 sec	✓	
Subsystem must detect properly worn mask with 90% accuracy	< 97%	~95%	~95%	~95%		
Detections must be done while subject and the camera has a distance x of 30cm < x < 100cm				✓		
Required detections must be done while subject is at the sight of the camera		✓	✓	✓		
Required detections must be done while while subject is facing the camera with an angle of at least 10°			✓			
Subsystem needs to be physically robust in case of a slight damage						✓
Subsystem needs to be installable and compatible on any wall surface						✓

5.2 TEMPERATURE MEASUREMENT SUBSYSTEM

The accuracy of Temperature Measurement Subsystem is assessed by comparing the output of subsystem to the measured value of the temperature. A medical digital thermometer is used for the correct measurement. For a satisfactory result, measured temperature should not differ from the actual temperature more than 0.5 °C. Moreover, the system does not accept inputs if the hand is too close to the subsystem or too far from it. Reason behind that is the ultrasonic distance sensor can not detect distances smaller than 2cm. Getting data from too far away is also problematic because then the scene will interfere with the object temperature and will yield inaccurate results.

To assess the performance of the Temperature Measurement Subsystem, several measurements are taken from the hand at different times and at different distances from the sensor. These sensor measurements are then normalized with the temperature of the person measured by digital thermometer. Normalized values are preferred because the real temperature is not constant and taking measurements at different time intervals would yield inaccurate results. From these data, a histogram is generated which is given in Figure 14. Histogram is approximated as a normal distribution and it's mean and standard deviation are extracted. Mean of the normalized measurements is 1.0005 and the standard deviation of the normalized measurements is 0.003. Empirical calculations on the normal distribution say that the 99.7% of the measurements fall within three standard deviations of the mean. Assuming that the average human body temperature is 36.5°C, 99.7% of the measurements will fall within 0.33°C from the mean, which is equivalent to 0.9% error at most.

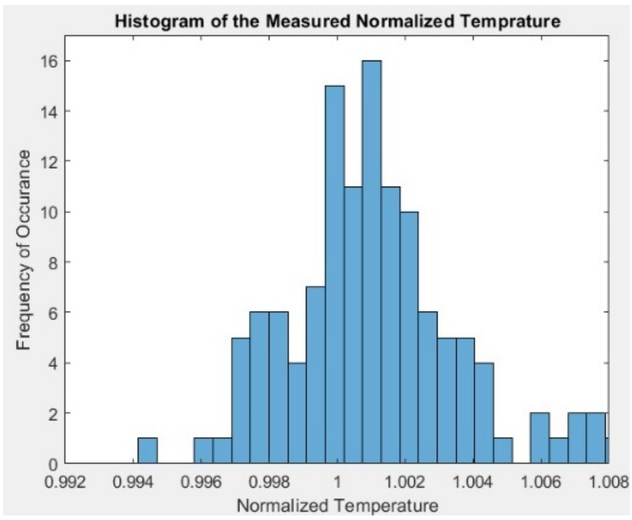

Figure 14: Histogram of the *Measured Normalized Temperature*

Due to pandemic conditions the tests were done by the same individual. Additional tests were done with several other participants to ensure system reliability. A small speed test was also done to check the detection speed of the sensor. Sensor was able to get 4 samples each second during its normal operation which means it can detect the temperature in 0.25 seconds. This number satisfies the speed requirements imposed on the system. All the requirements on the system is shown in a crosscheck table in Table 9 to illustrate the compliance with requirements.

Table 9: Crosscheck table for tests in Temperature Measurement Subsystem

	Accuracy test	Speed test	Integration Test
Subsystem must check for hand temperature	✓		
Temperature must be detected with no contact	✓		
Subsystem must check for distance	✓		
Subsystem must show the measurement results through a display			✓
Subsystem must exchange information with other subsystems		✓	✓
Temperature detection must be done in at most 1 seconds.		< 0.3 sec	
Temperature must be detected with an error of 0.5°C at most.	< 0.33 °C		
Distance between temperature sensor and subject's hand needs to be at least 2cm and at most 3cm	✓		
Subsystem needs to be physically robust in case of a slight damage			✓
Subsystem must be on waist level for wrist temperature measurement			✓

5.3 LOCK-PASSAGE SUBSYSTEM

5.3.1 MECHANICAL OPERATION TEST

For the lock passage subsystem, there are several issues that were tested subsystem-wise. First of all, the lock tongue was tested to check whether it follows a linear path in the lock or not. It is observed that after the lock is mounted on the door, the door is able to lock without any mechanical interruptions under 2 seconds. Secondly, the coherence between the cogwheel and the lock tongue was be tested under externally applied sudden torques. Due to the high torque applied by the servo motor, the cogwheel does not slip during its operation. The cogwheel is also fitted to the lock tongue so that it does not slip during normal operation. This was tested many times during the integration tests phase.

5.4 SYSTEM INTEGRATION TEST

This test involves all the subsystems working together. However, we can say that Lock-Passage Subsystem is responsible for the majority of the logical operations within the system. For instance; checking people count, checking if the door is opened or not, checking if anyone has entered or exited the building and communicating with other subsystems. Main concern of this test is to ensure the proper operation cycle of the overall system.

Firstly, Surveillance Subsystem is checked to see whether it can communicate with Temperature Measurement Subsystem, when the mask is detected. It is observed that Surveillance Subsystem is able to stop the stream and notify Temperature Measurement Subsystem as shown in Figure 15 and Figure 16.

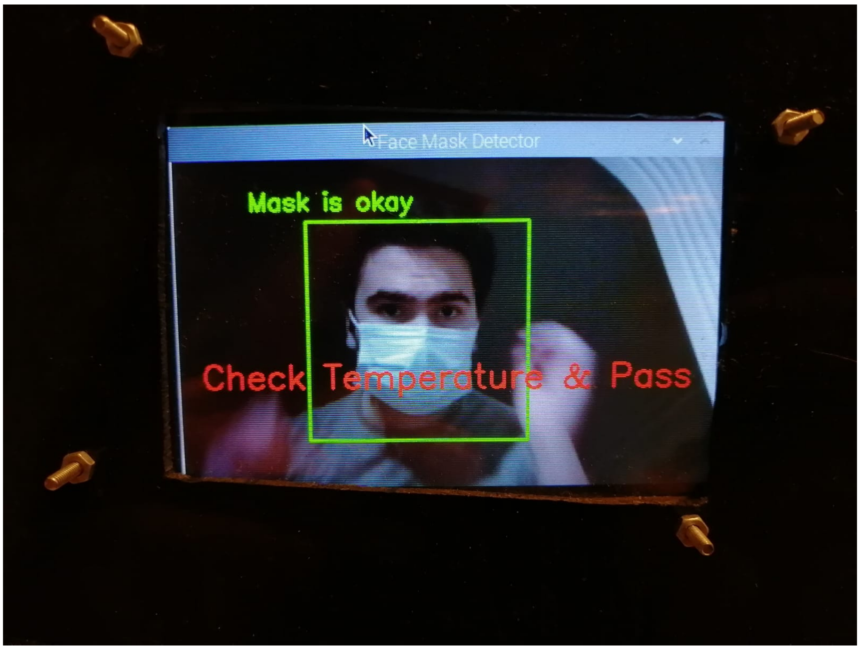

Figure 15: Surveillance Subsystem prompt after proper mask

Figure 16: Temperature Measurement Subsystem prompt for temperature measurement

When the temperature of the user is within the specified limits, Temperature Measurement Subsystem communicates with Lock-Passage Subsystem to open the lock. Figure 17 shows the acquisition of the temperature by the temperature subsystem and Figure 18 shows the lock-passage subsystem prompt after the door is unlocked.

Figure 17: Acquisition of the temperature by the Temperature Measurement Subsystem

Figure 18: Lock-Passage Subsystem prompt for unlocked door

After the door is unlocked, Lock-Passage Subsystem waits for the user to open the door and pass through it. If the user passes through the door the subsystem asks for the door to be shut down and gives the prompt shown in the Figure 19.

Figure 19: Lock-Passage Subsystem prompt for closing the door

In the case that the building becomes too crowded to accept any more visitors, Lock-Passage Subsystem gets into a crowded state and the Surveillance Subsystem stops its operation. The prompt given by the Lock-Passage Subsystem that the building is crowded is given in Figure 20.

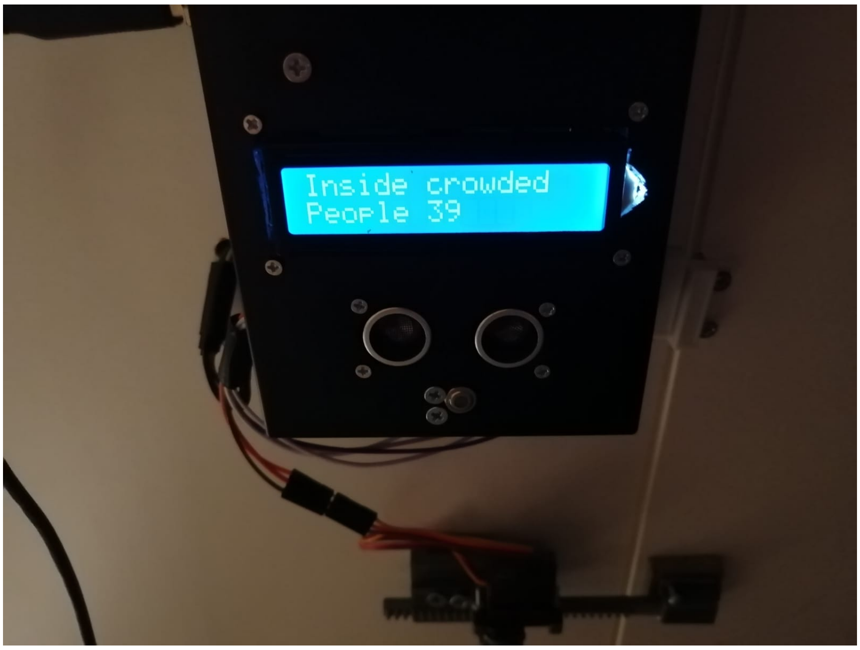

Figure 20: Lock-Passage Subsystem prompt for crowded building

All of these operations are tested numerous times by the tester before the demonstration and during the demonstration of the prototype. Finalized system is accepted to be devoid of any errors in terms of communication between the subsystems and operation flow within the subsystems. Though quantitative results for these tests are not possible to give, correct system operation without flaws after many trials verifies the results of these tests as successful.

6 LIST OF DELIVERABLES

6.1 HARDWARE DELIVERABLES

All subsystems are delivered pre-installed. The mounting and integration of subsystems will be done by staff. There are three main hardware units, which are shown in Appendix B, and the content of each unit can be listed as:

- Surveillance Subsystem
  - 1x Raspberry Pi 4 4GB
  - 1x Raspberry Pi 3.5 Inch LCD Screen
  - 1x Raspberry Pi Camera Module
  - 1x Raspberry Pi Cooling Module
  - 1x Case
- Temperature Measurement Subsystem
  - 1x Arduino UNO R3
  - 1x Non-Contact Temperature Sensor
  - 1x Ultrasound Distance Sensor
  - 1x LCD Display 2x16
  - 1x USB Type-A Male to USB Type-B Male Cable
  - 1x Case
- Lock-Passage Subsystem
  - 1x 3D-Printed Linear Actuator and Hinge Lock Pair
  - 1x Servo Motor
  - 2x Ultrasound Distance Sensors
  - 1x Magnetic Sensor

6.2 SOFTWARE DELIVERABLES

CONTROLVID-19 has 2 main board computing units, which are a Raspberry Pi 4 and an Arduino UNO.

- Raspberry Pi includes the Surveillance Subsystem codes, which are written in Python language. Deep learning models that are used in mask detection codes are trained beforehand. In the training, we extensively make use of deep learning libraries, mainly TensorFlow. Moreover, Raspberry Pi includes serial communication codes written in Python language as well, which communicates with Arduino UNO.

  What is more, we have included the deep learning model training codes in the Raspberry Pi for curious users, where they can investigate how mask detection model training works.

- Arduino UNO includes the Temperature Measurement and Lock-Passage Subsystems codes. Arduino code is written in C++ with an addition of special methods and functions by using Arduino libraries. Temperature Measurement subsystem codes measure temperature and distance with sensors connected to Arduino. Lock-Passage subsystem code controls servo motor and controls the signal flow. Similarly, Arduino UNO has serial communication codes that allow it to communicate with Raspberry Pi.

6.3 USER MANUAL AND SERVICES

The user manual of the product is given in Appendix D, where it includes assembly, initialization, warnings and drawings of the product. Furthermore, the company gives one-year warranty which covers every non-user related complications. Following the one-year warranty period, the company provides both software and hardware maintenance services for monthly subscription.

7 RESOURCE MANAGEMENT

7.1 COST BREAKDOWN AND ANALYSIS

Actual expenditures of the project is calculated as $163, the detailed list of components and their prices are presented in Table 10.

Table 10: Actual expenditures for the end product

Equipment	**Quantitiy**	**Price**
Raspberry Pi 4 4GB	1	$75
Raspberry Pi 3.5 Inch LCD Screen	1	$25
Raspberry Pi Camera Module	1	$15
Temperature Sensor	1	$10
3D Printing	1	$10
Raspberry Pi Cooling Module	1	$5
Arduino UNO R3	1	$5
Screws, Nuts and Enclosure Boxes	1	$5
HC-SR04 Ultrasonic Distance Sensor	4	$4
Servo Motor	1	$3
2x16 LCD Display	1	$2
Magnetic Sensor	1	$3
Cables & Resistors	-	$1
Total Price		**$163**

The total expenditures can be concluded for either prototype or mass productions. The total cost is the product of the total labor in terms of hours and the gross wage rate in terms of $/hr. To be convenient, the gross minimum wage rate per hour is taken as *$2.2.* The total labor can be calculated as product of the number of employees, labor per week and the duration in terms of weeks.

- **Prototype:** The company proposed *6 months* to develop a working prototype in their Proposal Report. Hence, the time spent develop a prototype is equal to *24 weeks.* For a trivial solution, labour per week can be assumed as *6 hours (3 for meetings and 3 for individual & group studies).* Since the company consists of *five people,* the total labor is equal to *720 hours.* Eventually, the engineering cost for prototype developing (including design, integration, tests and reviews) is equal to *$1584.* Including the actual expenditures cost, the total cost for prototype developing is equal to *$1747,* which is illustrated in Figure 21.

- **Mass Production:** Since the softwares are ready to use, only assembly and installation (if preferred by the consumer) are included in the labor. The whole system can be assembled in *1 hour* by *an employee.* If the installation is provided by the company, the labor for installation including the first usage test is *2 hours* for *an* *employee*. In that case, the total labor (assembly & installation) is calculated as *3 hours* per product. In the end, the total mass production cost is equal to the *$6.6*. Including the actual expenditures costs, the total cost is calculated as *$ 169.6*, which is illustrated in Figure 22.

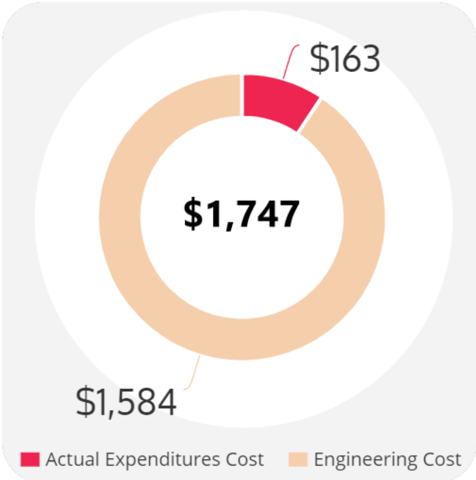

Figure 21: Total cost for prototype development

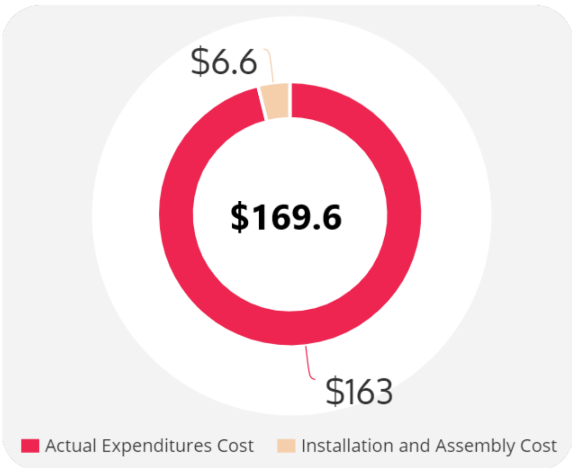

Figure 22: Total cost for mass production (per unit)

7.2 POWER MANAGEMENT ANALYSIS

The power distribution chart for power dissipation of subsystems was given in Critical Design Review Report as in Figure 23. The measurements were taken for each subsystem individually by using a multimeter. However, since the measurements were taken by a multimeter and the power dissipation of the integrated system was not measured, a more complicated and accurate way of measuring is chosen and conducted.

Figure 23: Power distribution chart presented in CDRR

Since there is only one power supply on the system, which is the adapter of the Raspberry Pi 4, a USB power meter was used to measure the power dissipation of the overall system for more accurate results of the power dissipation of the system. As Temperature Measurement Subsystem and Lock-Passage Subsystem share the same Arduino, their power dissipation will not be separated. After the subsystems were realized and integrated, their power consumption were measured with a USB power meter shown in the Figure 24.

Figure 24: USB power meter

As the power consumption is not constant during the operation cycle, the full passage operation was done while monitoring the power meter. The power dissipation for each sample was plotted. The resulting graph is given in the Figure 25.

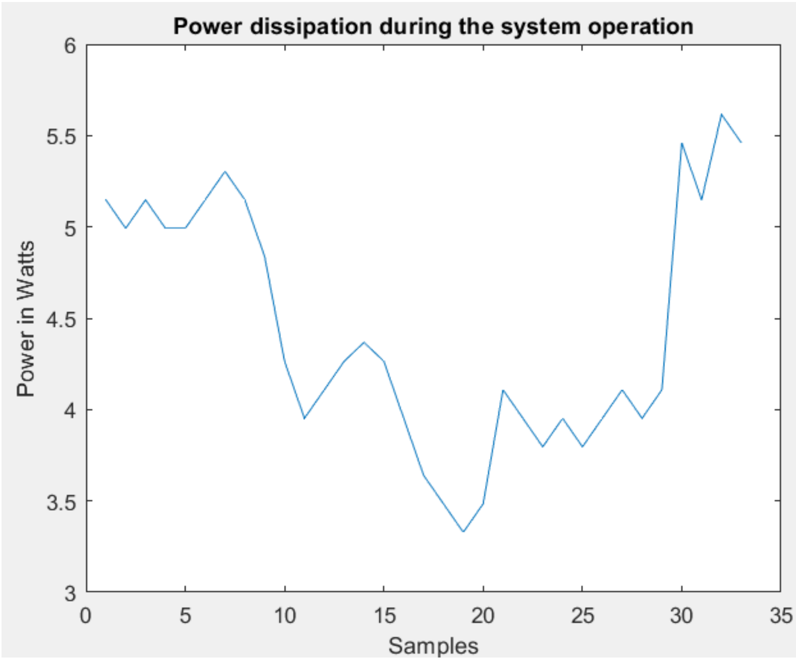

Figure 25: Power dissipation of the operation cycle

As observable in the graph during the operation the power dissipation decreases significantly. The reason for this is the fact that stream is stopped within this period. While the stream continues to flow, the power dissipation of the overall system is around 5.46 W. When Surveillance Subsystem is working on its own without a connection to Arduino, the power dissipation is around 4.84 W, which shows that Temperature Measurement and Lock-Passage Subsystems dissipates 0.62 W while the stream is flowing. To estimate the power dissipation while the system is working, the parts of the graph where the dissipation decreases are averaged. The resulting averaged power dissipation is around 3.98 W. The graph containing the average power dissipation for the frequency of the system usage is given in Table 11. In this graph, the frequent system usage means 10% of the time system is being used, moderate usage means 5% of the time system is being used and infrequent system usage means 10% of the time system is being used. When the system is not used power dissipation is 5.46 W on average and when the system is being used the power dissipation is 3.98 W on average.

Table 11: Power dissipation table for different operation frequencies

	Maximum Dissipation	Minimum Dissipation	Average Dissipation
Infrequent usage	5.46 W	3.98 W	5.44 W
Moderate usage	5.46 W	3.98 W	5.39 W
Frequent usage	5.46 W	3.98 W	5.31 W

As it is observable from the table more frequent usage of the system leads to less power dissipation. This is due to the fact that after the mask is verified, the video stream is stopped. Considering the fact that total dissipation of Surveillance Subsystem is 4.84 W and sum of the power dissipation of Temperature Measurement and Lock-Passage Subsystems is 0.62 W, it can be concluded that Surveillance Subsystem is much more dominant in terms of power dissipation. Hence, when the stream is stopped, power dissipation of the overall system decreased considerably. Highest average power dissipation of the system is 5.46 W. Highest average power distribution of subsystems are illustrated in Figure 26.

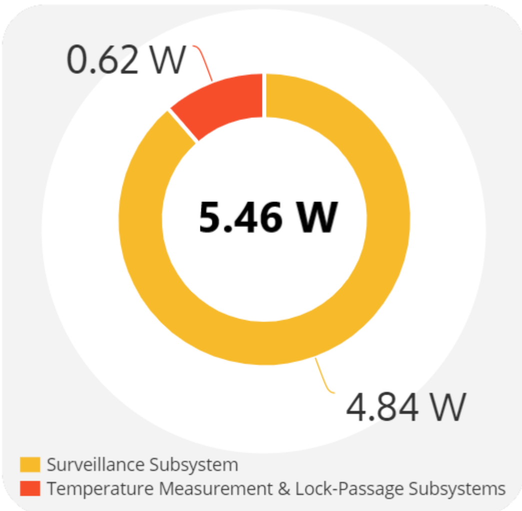

Figure 26: Highest average power distribution chart

8 DISCUSSION

8.1 SAFETY ISSUES & PRECAUTIONS

Since the system is designed for commercial use, safety is one of the most important issues in both designing and marketing phase. This paragraph is dedicated to convey possible safety issues related with the subsystems and the precautions taken by the company in order to avoid any safety problems.

- **Surveillance Subsystem:** In order to commercialise a product, Electromagnetic Compatibility (EMC) and Electromagnetic Interference (EMI) standards have to be satisfied. In order to cope with Electromagnetic Interference (EMI), screening is a solution. In addition to that, screening can be used to reduce the entry of any extraneous matter. Hence, the company chose a non-conducting enclosure for the Surveillance Subsystem. By this means, the inner circuitry is isolated from the environment.*

  In order to reduce the risk of by-pass or internal hardware damage, the components are fixed via screws. The conducting branches of cables are covered with friction tape.

  The shape of the enclosure is selected as wide at the bottom and narrow at the top. By this means, the Surveillance Subsystem exhibits a strong stand on a plain surface.

  Any stray voltage is harmless for human body since the voltage supplied to the whole system is **5 Volts.** The main display is covered with mica; hence no incisive matter is sprinkled due to a incident.

  **No RF circuitry is used in the Surveillance Subsystem. Moreover, the wireless antenna on the subsystem is disabled.*

- **Temperature Measurement Subsystem:** Similar to the Surveillance Subsystem, a non-conducting enclosure is preferred for the Temperature Measurement Subsystem, too. IR temperature sensor, proximity sensor, 2x16 LCD screen and Arduino UNO are fixed on the enclosure in order to prevent any by-pass and internal hardware damage.

  The shape of the enclosure box is chosen as rectangular prism, and the subsystem is mounted on a vertical surface via screws.

  The USB cable used for the serial communication between the Surveillance Subsystem and the Temperature Measurement Subsystem is chosen as a hardened cable. The actual USB cord is covered with resilient metal wires and rigid plastic.

  The operating frequency of the proximity sensor is 40 kHz which indicates the sensor has low EMI index.

- **Lock-Passage Subsystem:** In this subsystem, only very low powered components are used. In order to prevent the lock from breaking, the linear actuator is conditioned on the position of the magnetic sensors.

  The proximity sensors are placed back to back. In addition to that, their operation frequency is 40 kHz. Moreover, there is a rigid barrier, which is a gateway in this case, between the Lock-Passage Subsystem proximity sensors and the Temperature Measurement Subsystem proximity sensor. To conclude, the EMI index of the Lock-Passage Subsystem is expected to be negligible.

8.2 WIDESPREAD APPLICATION EXAMPLE & IMPACTS ON SOCIETY

CONTROLVID-19's use settings are preferred to be confined spaces, where the infection has a greater rate of spreading than open spaces. This is due to the fact that the air inside confined spaces are not refreshed for many hours, in other words virus stays on the air in confined spaces. Therefore, main usage area of CONTROLVID-19 would be closed buildings that masses of people would gather around. This system would make security personnel's checking mask and temperature job easier.

Technological products more or less have an impact on society. Broadly speaking, impact on society of a technological product can be measured by some qualitative factors. Main factors are; *how widespread the product is* and *how well does the product solve the problem.* If the product solves the problem in a better or more practical way, then it will be easier for society to adapt; making the product more widespread. One can also say that the more widespread a product is, the more impact on society a product has. As Zeyrek, we have developed a useful and practical product, but we need to manufacture it on great masses so as to have the societal impact that we desire.

As it is written before, CONTROLVID-19 makes personnel's job easier by assisting them. However, the product is originally designed to be used without supervision. Thus, it can reduce the need for human resources and increase productivity. What is more, widespread use of CONTROLVID-19 may result in less carbon footprint generation than a manual mask and temperature checking process, since automating a process generally decrease the carbon footprint.

8.3 POTENTIAL ENVIRONMENTAL EFFECTS IN WIDESPREAD USE

First of all, heat emitted to the environment while the system is running is not significant. The maximum power delivered to the system (5.46 Watts) is less than the maximum power delivered to a conventional yellow bulb (100 Watts); hence the power economy issue can be counted as insignificant. Furthermore, no chemical ingredients such as batteries are used in the system. However, enclosures of the subsystems are made of plastic. Therefore, they cannot be recycled easily. Nevertheless, their contribution to plastic pollution is small considering that they are not single-use plastics. All in all, it can be said that the product causes little to no harm to environment.

9 CONCLUSION

In this report, the motivation behind designing CONTROLVID-19 and the concerns that Zeyrek have over the ongoing pandemic is stated. Then, the project is described with a top down approach, i.e., it is first described as a system and then it is described at subsystem level. After the subsystems was introduced, a compatibility analysis is conducted. After this discussion, performance tests made for each subsystem are introduced. Furthermore, their validity are discussed and their results are analyzed. Finally, the tests are matched with the design requirements and the compliance with these requirements are shown. After the analysis on the design, a list of deliverables was presented. Product, services and user manual is included in the list of deliverables. The list of deliverables is followed by the resource management where the cost breakdown and power management of the design is presented. Finally, other discussions regarding the design are made. These discussions include the safety issues regarding the design, possible applications of the product and its environmental effects. Since subsystem implementation is so crucial for this product, each subsystem can be summarized by mentioning its details as follows:

Surveillance Subsystem is used to perform a mask check, i.e, to separately check whether someone has a mask on their face and whether they are wearing their masks correctly. As explained with details, Surveillance Subsystem uses Haar Cascade and MobileNetV2 CNN model to process the feed from its camera. The model is trained by using a synthetically generated dataset named MaskedFace-Net created by Cabani et al [2]. Subsystem makes its detection in under 2 seconds with an accuracy of approximately 95%. It is robust under different illumination levels, angles and distances.

Temperature Measurement Subsystem measures the body temperature to detect possibly infected people, which uses a non-contact temperature sensor to measure the body temperature and an ultrasonic distance sensor to improve the sensitivity of temperature sensor by characterizing its performance with respect to the distance. The sensor, as previously discussed, works satisfactorily in the range defined previously. One can argue that this is not a good way to optimize the temperature sensor, however, since we are working in a small range of temperature, this method is sufficient for our purposes. Subsystem makes its detection in under 1 seconds with an error of at most 0.5°C.

Lock-Passage Subsystem controls the entrance and keeps track of entering and exiting passengers. Lock-Passage Subsystem uses a hinge lock and a servo motor to control the hinge lock. Small size and mass of the hinge lock will let us use a relatively smaller, hence cheaper motor for lock mechanism. Such a lock is better suited for our purposes, since the durability of lock is not our main concern. Moreover, the subsystem uses distance sensors to keep track of entering and exiting passengers. Using distance sensors for this purpose provides us with a simple solution for relatively important task. With magnetic sensors, the situation is door is determined, and a decision is taken accordingly. The door is able to lock without any mechanical interruptions under 2 seconds.

10 REFERENCES

[1] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 2018, pp. 4510-4520, doi: 10.1109/CVPR.2018.00474.

[2] Cabani, Adnane & Hammoudi, Karim & Benhabiles, Halim & Melkemi, Mahmoud. (2020). MaskedFace-Net – A Dataset of Correctly/Incorrectly Masked Face Images in the Context of COVID-19. 10.13140/RG.2.2.20336.28165.

[3] "Datasheet Raspberry Pi 4 Model B," Jun-2019. [Online]. Available: https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2711/rpi_DATA_2711_1p0_preliminary.pdf. [Accessed: 10-Dec-2020]

[4] Thingiverse.com. (n.d.). Linear Servo Actuators by potentprintables. Retrieved January 08, 2021, from https://www.thingiverse.com/thing:3170748

[5] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, Kauai, HI, USA, 2001, pp. I-I, doi: 10.1109/CVPR.2001.990517.

[6] Team, P. (2020, October 28). Connecting a Raspberry Pi to an Arduino Uno. Retrieved April 08, 2021, from http://www.deviceplus.com/arduino/connecting-a-raspberry-pi-to-an-arduino-uno/

[7] Oscar, amp; Oscar. (2018, March 23). Connect Raspberry Pi and Arduino with Serial usb cable. Retrieved April 08, 2021, from https://oscarliang.com/connect-raspberry-pi-and-arduino-usb-cable/

11 APPENDICES

A 3D DRAWINGS OF THE SUBSYSTEMS

A.1 SURVEILLANCE SUBSYSTEM

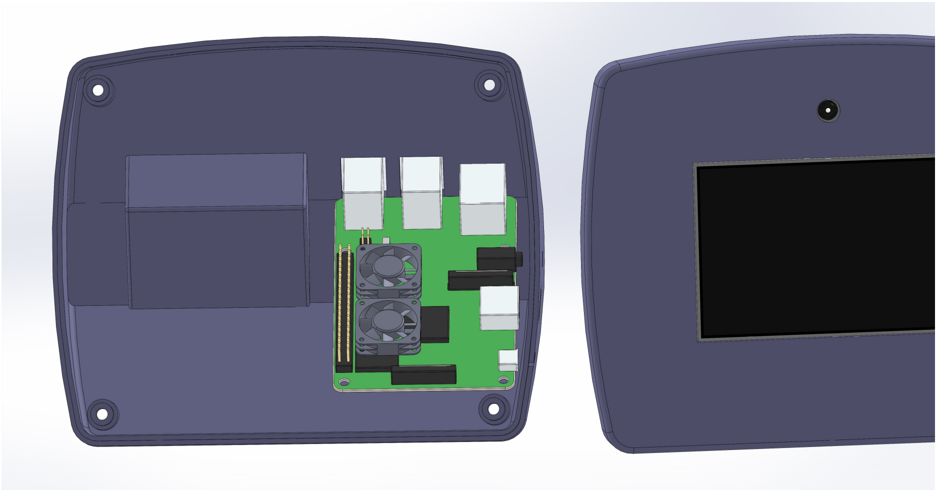

Figure 27: Interior view of the Surveillance Subsystem

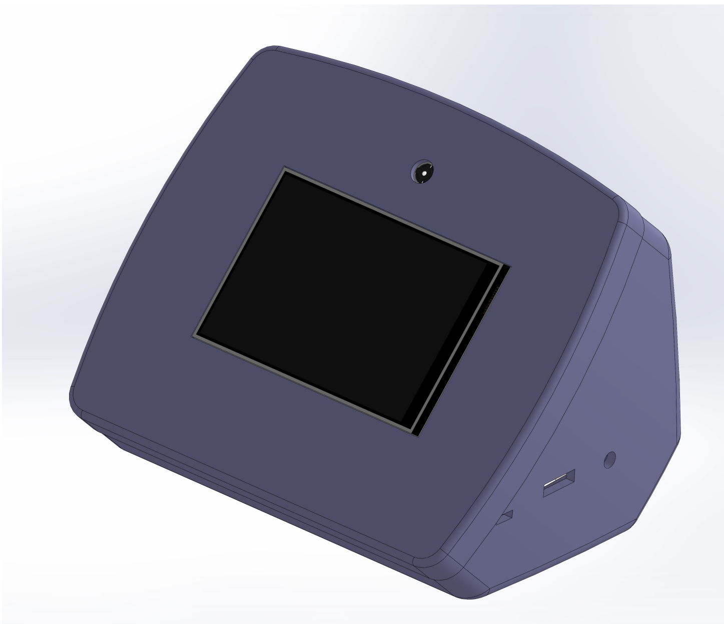

Figure 28: Isometric view of the Surveillance Subsystem

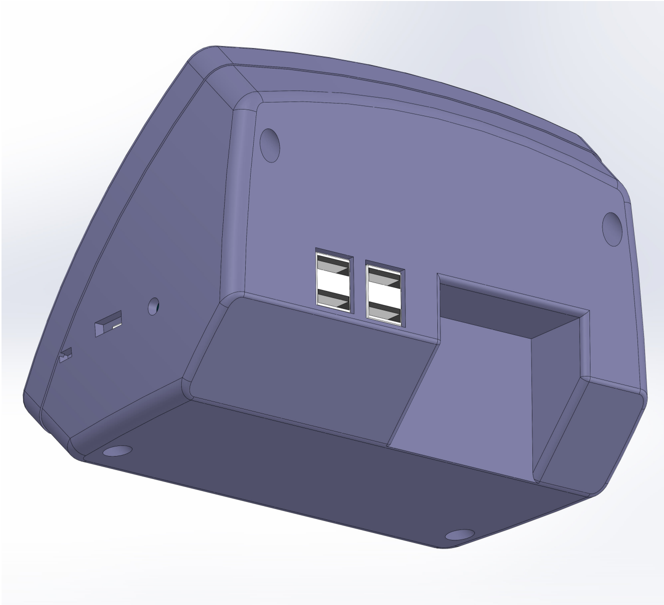

Figure 29: Rear view of the Surveillance Subsystem

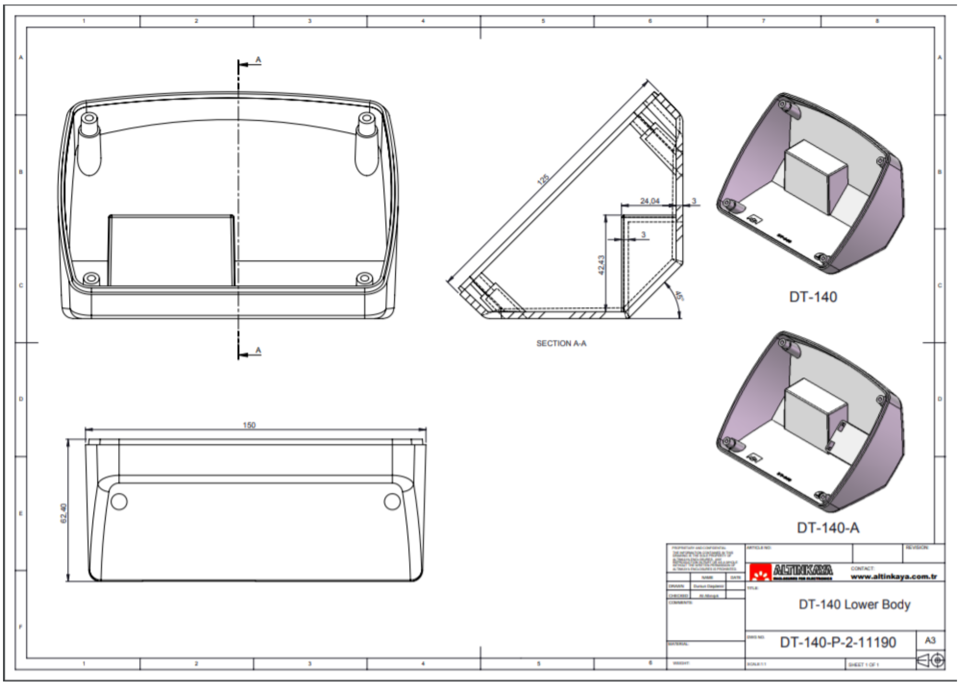

Figure 30: Technical drawing of the Surveillance Subsystem enclosure

A.2 TEMPERATURE MEASUREMENT SUBSYSTEM

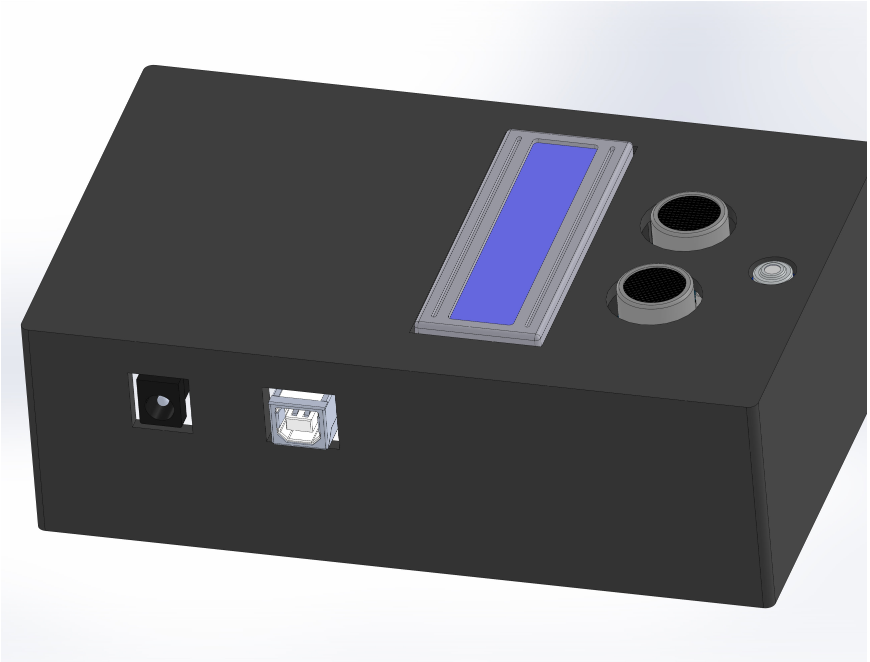

Figure 31: Isometric view of the Temperature Measurement Subsystem

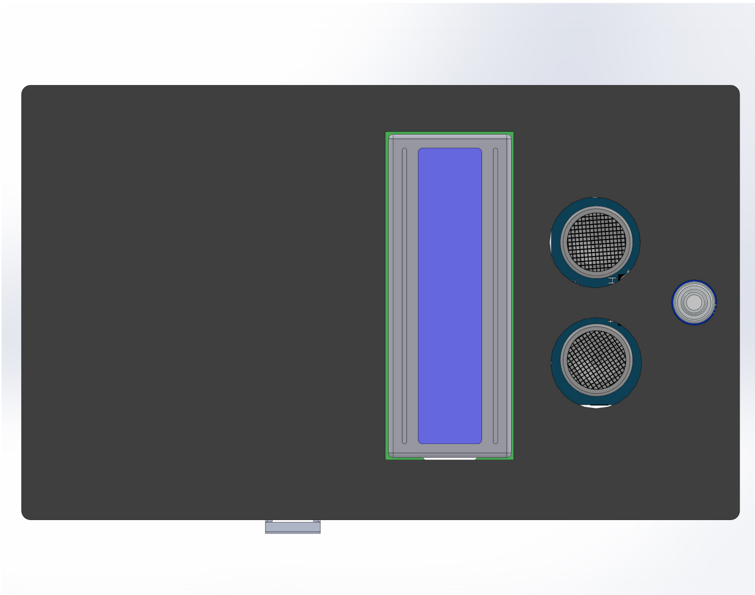

Figure 32: Top view of the Temperature Measurement Subsystem

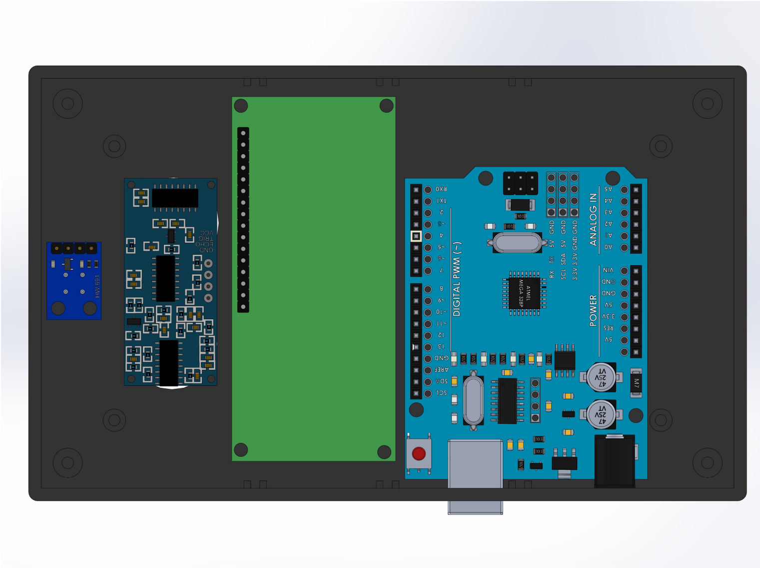

Figure 33: Interior view of the Temperature Measurement Subsystem

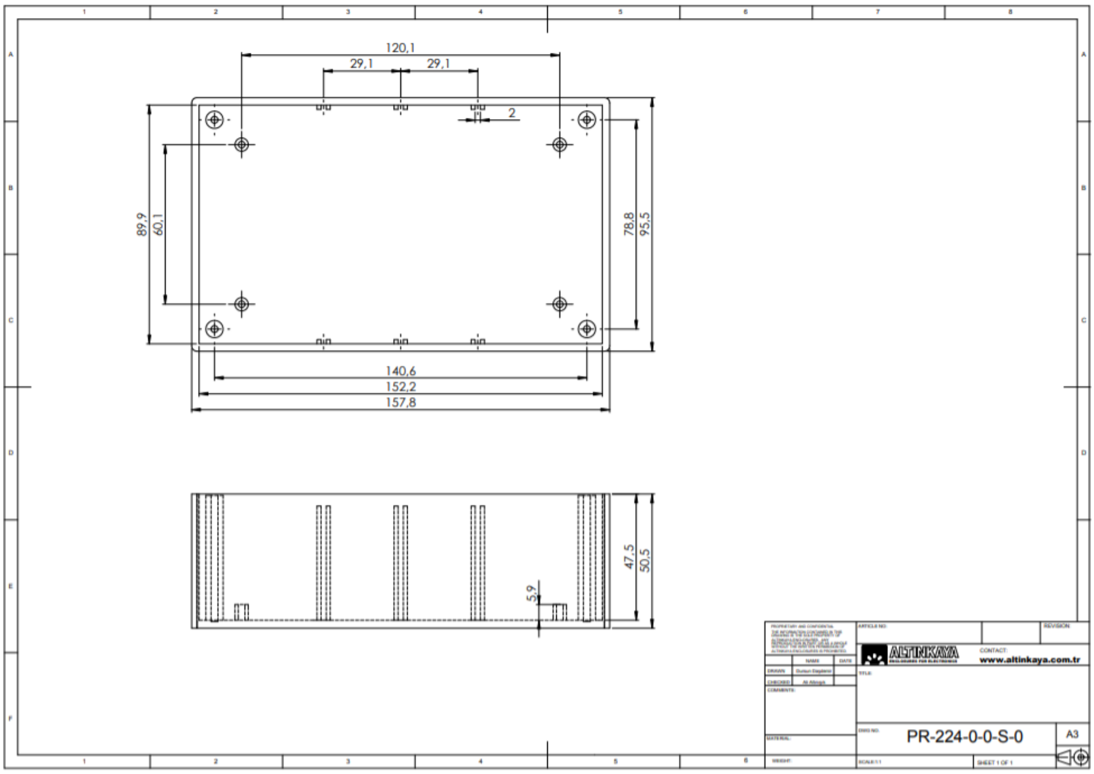

Figure 34: Technical drawing of the Temperature Measurement Subsystem enclosure

A.3 LOCK-PASSAGE SUBSYSTEM

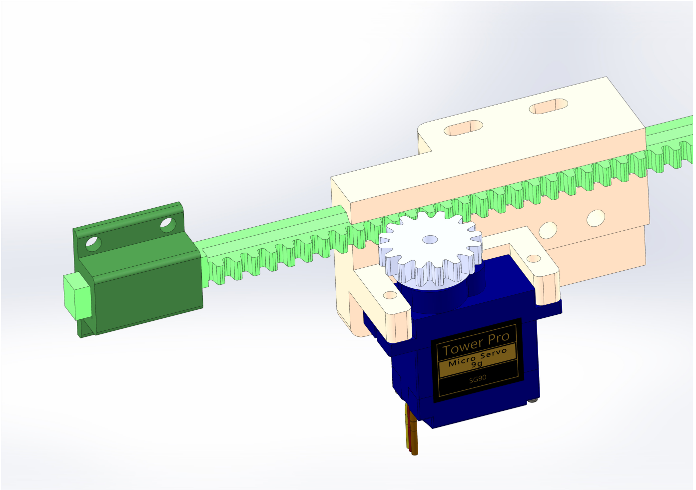

Figure 35: Lock-Passage Subsystem when the lock is locked

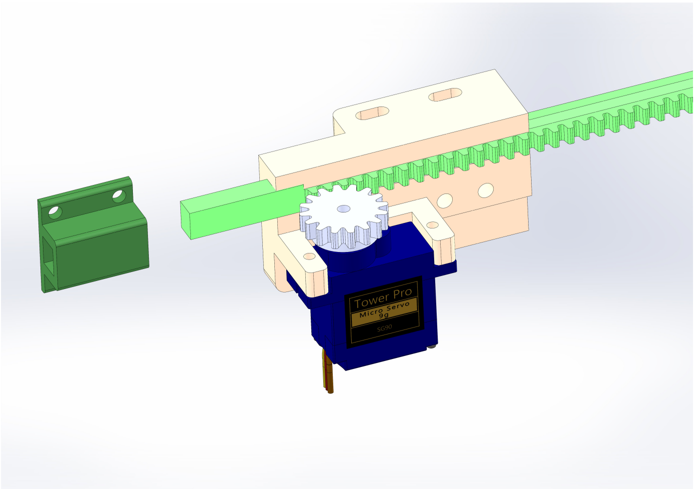

Figure 36: Lock-Passage Subsystem when the lock is unlocked

B REAL-LIFE APPEARANCES OF THE SUBSYSTEMS

B.1 SURVEILLANCE SUBSYSTEM

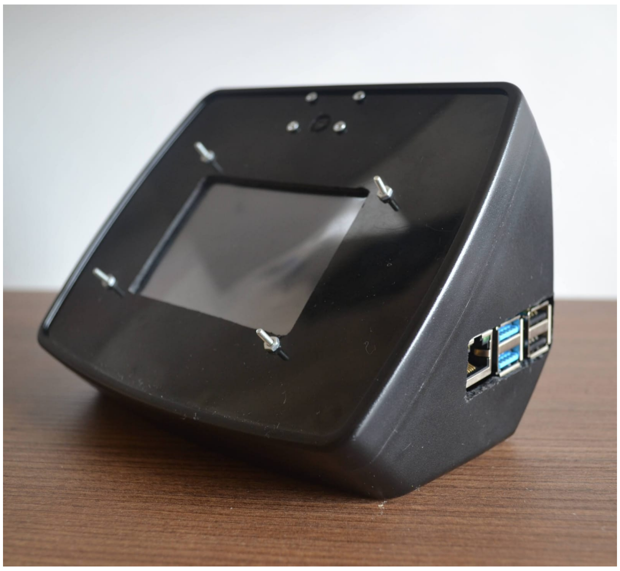

Figure 37: Isometric view of the Surveillance Subsystem

Figure 38: Rear view of the Surveillance Subsystem

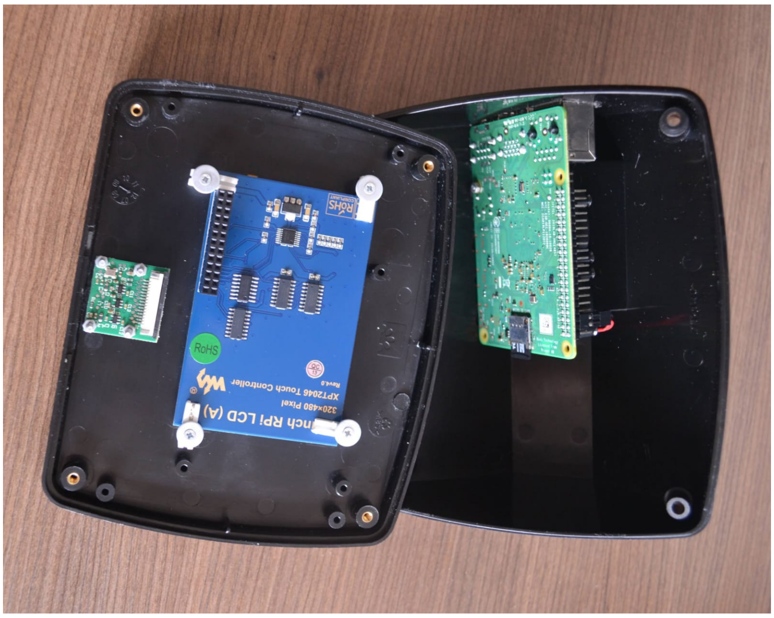

Figure 39: Interior view of the Surveillance Subsystem

B.2 TEMPERATURE MEASUREMENT SUBSYSTEM

Figure 40: Isometric view of the Temperature Measurement Subsystem

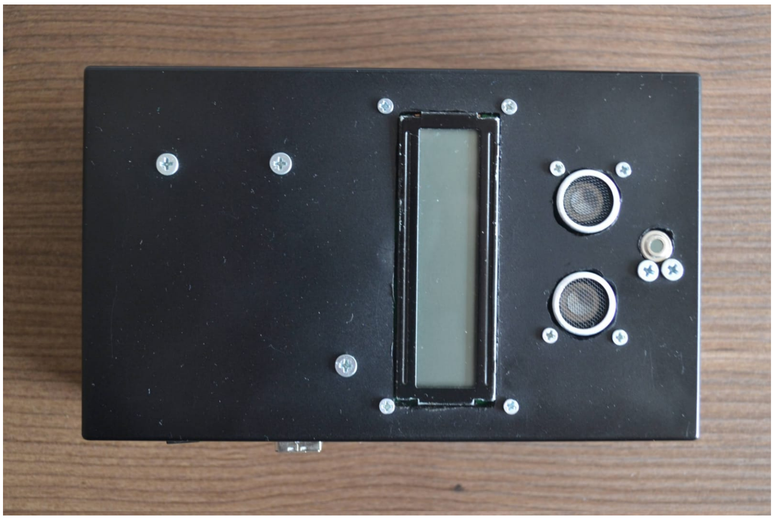

Figure 41: Top view of the Temperature Measurement Subsystem

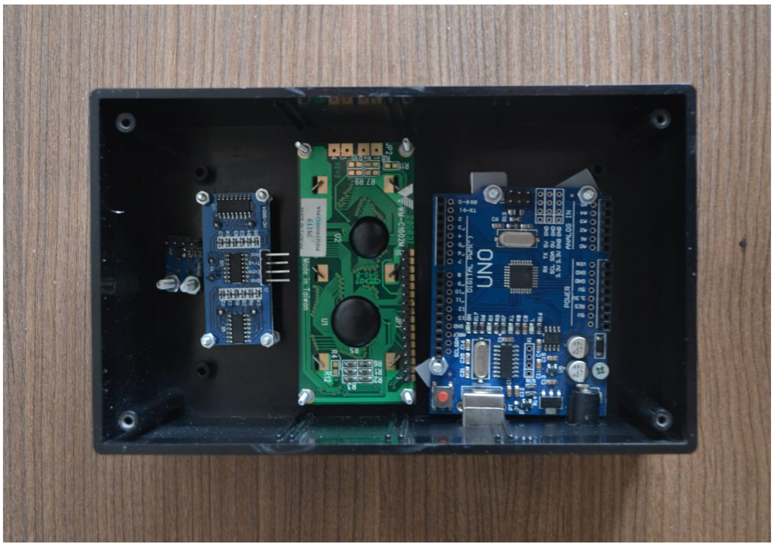

Figure 42: Interior view of the Temperature Measurement Subsystem

B.3 LOCK-PASSAGE SUBSYSTEM

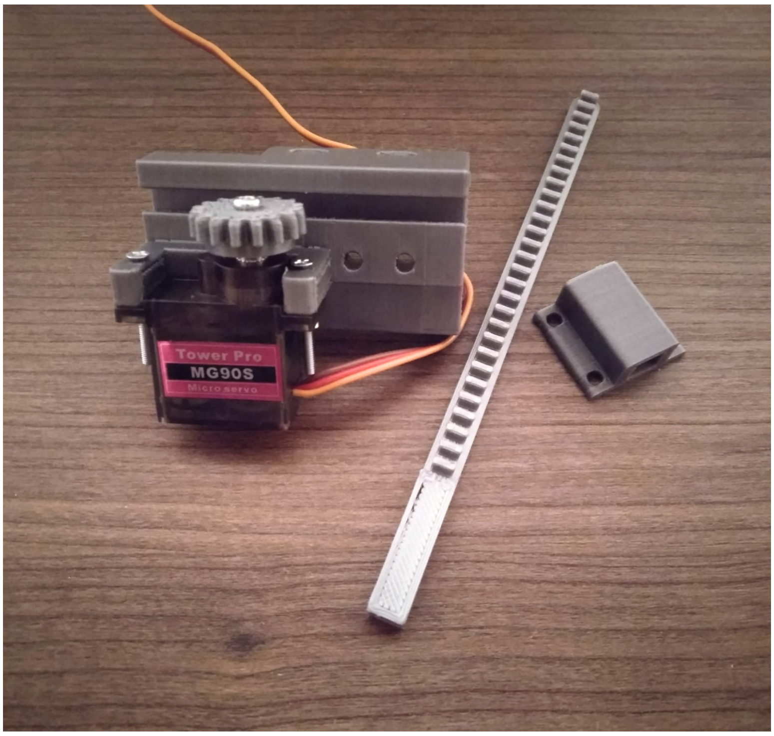

Figure 43: Parts of Lock-Passage Subsystem

C FINAL SETUP OF THE SYSTEM

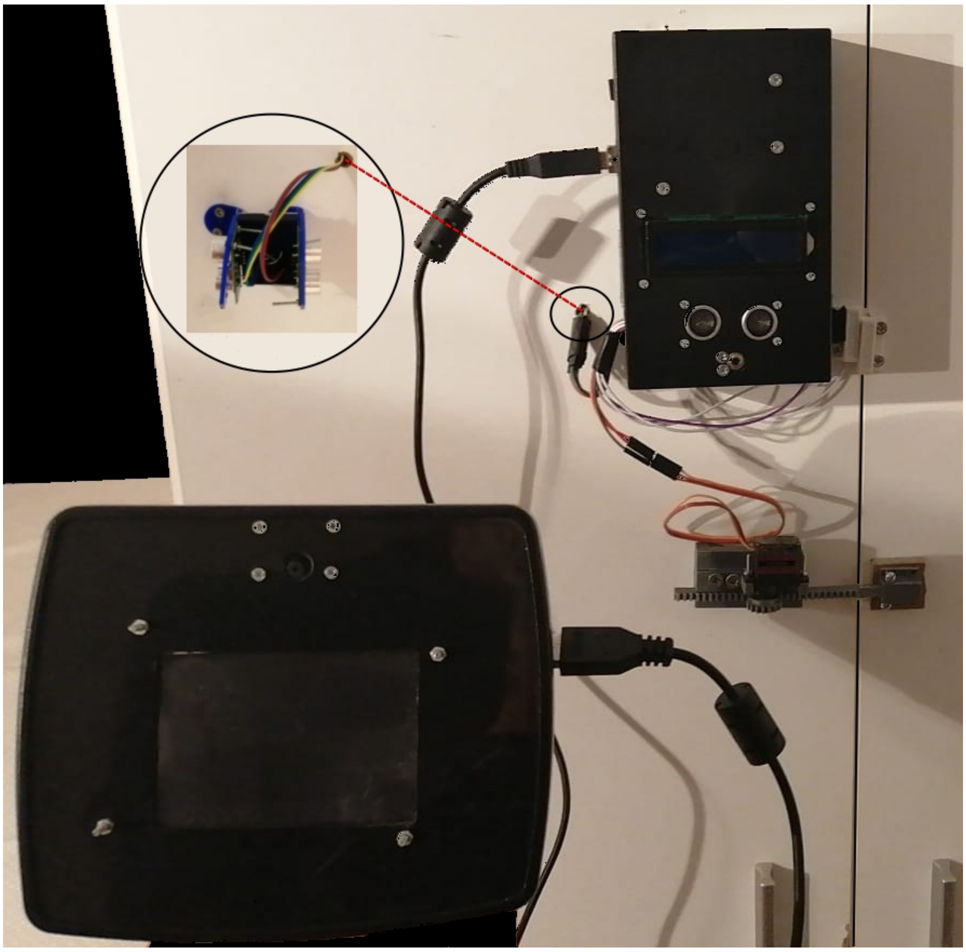

Figure 44: Final setup for CONTROLVID-19

D USER MANUAL

CONTROLVID-19 / User Manual

This user manual is written for the ***single-door*** *version only.*
IMPORTANT NOTE : If the mounting will be done by staff, you may skip to *First Impression* step. Else, read carefully from the beginning.

Before You Begin

First, check the contents right after unpacking. The content list should be as following :

— 1x Surveillance Subsystem
— 1x Temperature Measurement Subsystem
— 1x USB Type-A Male to USB Type-B Male Cable
— 1x Linear Actuator w/ a Servo Motor
— 2x HC-SR04 Ultrasound Proximity Sensor*
— 1x Magnetic Sensor*
— 1x Hinge Lock*
— 20x Screws*

**According to the product purchased (single-door or dual-door), amounts of these components may vary.*

Assembly

Before you begin, make sure that there is no bulge to disturb the linear actuator on your gate.

1. Mount the linear actuator and the hinge lock on the gate using screws. To be convenient, do not install the lock module higher than waist level.
2. Mount the magnetic sensor on the gate such that the dummy part (the part without any connection) is mounted on the moving surface.
3. Install the Temperature Measurement Subsystem on a fixed surface. It is better to install the module on chest level since the measurement is taken from palm.
4. Install the proximity sensors back to back. One sensor must be orthogonal to the entrance. On the other hand, the other sensor must see nothing except intentional interference.
5. Put the Surveillance Subsystem on a fixed surface with a height between chest and waist level.
6. Make necessary connections between the Temperature Measurement Subsystem and the rest. Be careful, **do not connect anything to the Surveillance Subsystem including the power cord.**

Initialization

1. Connect Type-B edge to the Temperature Measurement Subsystem and Type-A edge to a computer with Arduino IDE software installed.
2. Open the file with *.ino* extension.
3. Configure the threshold values for the proximity sensors according to the environment they are built in.
4. Debug the configured file and upload it to the Temperature Measurement Subsystem.
5. Power up the Surveillance Subsystem
6. Connect Type-A edge to the Surveillance Subsystem.
7. Start the Surveillance Subsystem. (This may take a while.)
8. Run the main program.

After running the program, you will see a screen showing the serial port connection is establishing and the program is ready to start. Please keep in mind that, the interface will ask for a headcount quota ; hence while deciding on this quota, follow the social distance regulations announced by the officials.

First Impression

From now on, your system should be ready to use. You should see that the video stream in the Surveillance Subsystem is live.
On the other hand, you should observe that the Temperature Measurement Subsystem is active and **Show Your Mask** is displayed on the 2x16 LCD screen.
Initially, the gateway should be locked since the system locks the door automatically just after it starts.

1. Please proceed to the Surveillance Subsystem with a mask and observe the following cases on the main display :
   (a) No Mask
   (b) Inappropriate Mask
   (c) Appropriate Mask
2. When the system sees a user with appropriate mask, the frame is captured instantly. Now, you should see **Please, check your temperature** warning is displayed.
3. The Temperature Measurement Subsystem works the best between **2** to **3** centimeters range. Show your palm to the Temperature Measurement Subsystem within the optimal range.
4. The temperature margin is decided according to the World Health Organization standards. The margin is pre-defined as 35 °C - 37.5 °C.
5. When the palm temperature is detected, the measured temperature is displayed on the 2x16 LCD screen.
6. If the palm temperature is within the pre-defined temperature margin, the gate should be unlocked.
7. The door stays unlocked until one of the following cases is carried out :
   (a) The door is opened and a passenger is entered.
   (b) The door is not opened for **15 seconds**.
   (c) The door is opened ; but no one has passed within **15 seconds**.**

   ***A short beep sound warns the staff that the door has been left open and it should be closed immediately.*
8. Now, you may proceed through the gateway. Please keep in mind that the door must be closed after your passage is completed.
9. The initial headcount is asked by the interface at the beginning. For simplicity, it can be assumed as **zero**.
10. After your passage is completed, the headcount must be increased by **one**. You may observe the new headcount on the Temperature Subsystem display. When the whole system is in stasis, the headcount is displayed on the 2x16 LCD screen continuously.
11. Now, you may exit by showing your hand to the *blind* proximity sensor. However, if the door is unlocked but not opened, it will be locked again after **15 seconds.**
12. After you exit through the gateway, the headcount must be decreased by **one**. This decreasement can be observed through 2x16 LCD display.
13. If the previous steps are passed smoothly, your CONTROLVID-19 is intact and ready to use. ***CONGRATULATIONS.***

WARNING

1. This system is designed to be contact-free. Please do not touch the sensory areas.
2. **Do not** plug out the power cord and the serial connection USB cable while the system is running.
3. **Do not** force the linear actuator to open. The servomotor is not designed for externally applied torque.
4. **Do not** disturb the enclosures of the modules. If you think that there are any hardware-related issues, please contact with the staff immediately.
5. If the main display is not working properly, you may use an external screen via mini HDMI. However, if you cannot start the external screen, the Surveillance Subsystem might need a firmware alteration. In that case, please contact with the staff and request a firmware alteration.
6. If the 2x16 LCD screen is not working properly, the internal connections may be damaged. In that case, please contact with the staff and request a hardware check.
7. In order to solve an issue by restarting the whole system, please **do not** unplug the power cord at first. You should shutdown the Surveillance Subsystem first via the Raspbian. Then, you may unplug the power cord.

Figure 45: Front page of user manual of CONTROLVID-19

SUBSYSTEMS

Please use the figures below to understand the subsystems.

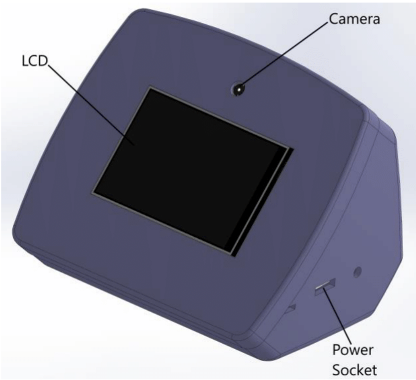

Figure 1: *Front side of Surveillance Subsystem*

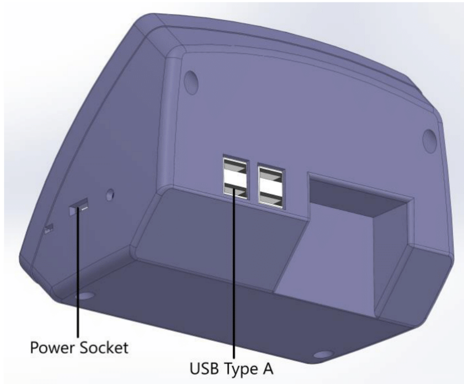

Figure 2: *Rear side of Surveillance Subsystem*

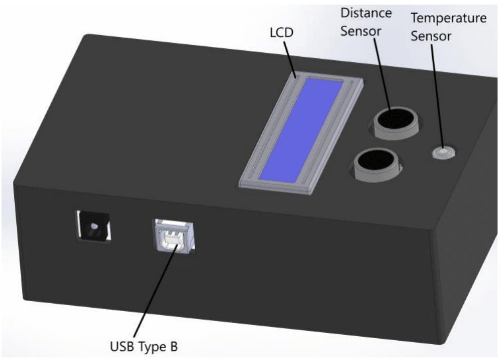

Figure 3: *Temperature Measurement Subsystem*

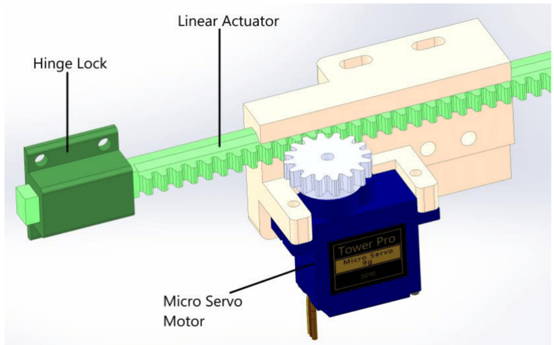

Figure 4: *Lock-Passage Subsystem*

Figure 46: Back page of user manual of CONTROLVID-19