UNIVERSITY OF CENTRAL FLORIDA

Department of Electrical Engineering and Computer Science

Dr. Samuel Richie

Coronavirus Prevention System

Group 24

Emanuel Cicortas – Computer Engineering
Dat Nguyen – Computer Engineering
Ha Tran – Electrical Engineering
Phuong Anh Vu – Computer Engineering

Table of Contents

Contents

Tables

Figures

1 INTRODUCTION

1.1 Executive Summary

The Coronavirus Prevention project is centered on the public safety and personal well-being of every single person who are living currently. From the mask distribution system to the health monitoring and alert system, the Coronavirus Prevent system delivers current technological solutions to both needs. The Coronavirus Prevention system is a concept that stems from the popularity of Pixar's Wall-E. The friendly looking robot can store items within its center compartment and scan its surrounding environment with the need of superficially complex internal or external design. The reason we chose Wall-E as the inspiration for our design is because Wall-E is cute. In the current time of chaos, having something cute sitting on the front door would be pleasant to look at.

Our design of the Coronavirus Prevention system will simplify the design of Wall-E even further. Since our primary purpose is to hand out masks and detect if a person has a fever of any sort, then a single scan when the person first enters the building should be more than enough. Thus, we will be discarding the wheels, as well as the arms, to make the robot stationary. This will help cut down on cost even further and save us much more time as we do not have to implement a navigation system. Another big change that we are making is that instead of the useless button in the front of the robot, we will be making good use of it. By pressing the button, the user will be able to manually open the center compartment without having to wait for the image processing unit to process or in the case that there is a software failure. The visual processing capability will still be there. We hope to retain the cuteness of the machine even after removing some of the preference design's features.

To be able to build a power delivery system alongside a mechanical center compartment, as well as, to learn how to train an effective and efficient machine learning model, we first have to research other projects that have similar features. Projects on building a basic ac-dc step-down converter were looked at when choosing the electrical components, choosing manufacturers, and designing our own power delivery system. Projects that involved Artificial Intelligence and Machine Learning were also extensively looked at to give an idea of where to start on designing a model. In the final phase of the hardware design, we made use of the MSP430 Launchpad reference board as a guide for laying out and interfacing different electrical components. As for the software side, we made use of the publicly available framework library called OpenCV to detect a human face. Then, with the use of the power-efficient, yet powerful NVIDIA Jetson processor to process the image taken and determine if the person is wearing a mask.

The prototype of our Coronavirus Prevention system will be implemented on rectangular acrylic box with an embedded microprocessor and an image processing unit. The box will have an opening, whose door is controlled by a motor that is subsequently controlled by the microcontroller. The microcontroller will not only be in charge of the motor, but it will also control the temperature sensor, and the distance sensor. The only thing that the image processor controls is the camera which takes the picture of the person standing in front of the robot.

It is the intent of our design to have the microcontroller, rather than the image processor, controls most other components in the system. Even though the image processor is much more powerful, it is also more power hungry. The image processor is also more likely to fail due to bad programming codes than is the microcontroller.

1.2 Motivation

SARS-CoV-2 of 2019, or simply COVID-19, is a strand of coronavirus that causes the COVID-19 pandemic that has been spreading quickly across the entire world since November of 2019. Starting March of 2020, the virus arrived on the land of the United States and has been terrorizing many communities and killed many people. As COVID-19 is spreading rapidly in many states in the United States, more than 7 million cases have been reported as of late September of 2020, and more than 204,000 people have lost their lives during the pandemic. As of recently, the number of new infected cases, as well as the number of new deaths, have not climbed as much as before, but the virus is in no way contained. The World Health Organization and public health officials in the States believe that wearing masks and maintaining social distancing of 6 feet or more could further help reduce the risk of transmission of the coronavirus. A new variation of the coronavirus has recently appeared in Europe, and it has already been causing havoc. Fear is brewing up again in the mind of the general public. This is proved by another sudden drop in the United States' stock market. The message advocating for wearing a mask whenever possible is becoming more convincing and more vocal than ever before. Therefore, we too, believe that wearing a mask has serious positive effects during the era of COVID-19, which could save many thousands of lives. Biology experts have said publicly many times that a vaccine will stop the virus in its track, but a vaccine is not expected to be going into mass production until the second quarter of 2021. So right now, the best bet that we have against the pandemic is to wear a mask in public.

Many places are inherently more vulnerable to become a catalyst of spreading the virus. By design, both physically and socially, these places create more opportunities for people to come in contact with another person. With higher contact rate, the rate of transmission will, correspondingly. be higher. These vulnerable places are restaurants, markets, schools, and workplaces. These places tend to be small in size but incur high traffic. Sometimes, people are squished together, shoulder to shoulder. In such cases, keeping a social distance of 6 feet is pretty much impossible. The best way to mitigate the spread of the virus in such situations is to wear a respirator, or as a minimum, a mask. Wearing a mask could help slowdown the COVID-19 spread rate. As a whole, the United States does not and cannot mandate the wear of a mask. However, at state level, many states are requiring people to wear masks when they are not inside their own house. Specifically, in the state of Florida, wearing mask is mandatory. Many markets and restaurants offer free masks for customers who "conveniently" forgot to wear one. However, distributing mask can prove to be costly as many places need to hire a person whose purpose is to solely hand out masks. Buying an automatic mask dispenser is expensive, so such dispensers are not common. Entrepreneurs are capitalizing on the pandemic by selling masks, and anything mask-related at a steep mark up. Inventors are bought by such entrepreneurs to help them capitalize on the profit. Creating cheap and convenient products to help schools and small businesses stay open is just a secondary thought.

Our goal is just that. We want to dive into the venue where most other inventors stray from due to a lack of funds and profit. We want to build a low cost and effective robot that uses machine learning to algorithmically detect faces of people who are not wearing masks. The robot would then automatically open its storage compartment for the none-mask-wearing person to reach in a fetch a mask. By building such a robot, we can help people raise awareness of the importance of wearing mask, so that they can protect themselves and other people around them from getting into direct contact with the coronavirus. Moreover, by making use of the facial mask-wearing recognition, we may potentially be able to help researchers study human behaviors and predict future disease hot spots before it is too late. We may also be able to work with law enforcements to notify them of places where people are gathering but are actively avoiding wearing masks.

1.3 Objective and Goals

Our necessity-based Coronavirus Prevention system aims to provide a cost-effective and easy to use system to distribute single-use masks and monitor the body temperature of those who entered the room. By relying on simple try-and-true components and designs, we try to cut the cost of the production of the robot to as low as possible. COVID-19 is a worldwide pandemic; and currently, the most effective prevention mechanism that we have against it is the use of face masks. And since most of the world’s population are living paycheck-to-paycheck, most people cannot afford an expensive machine for their building/business. To make our machine accessible to as many people as possible, the only option that we have is to lower the price as much as we can.

The Coronavirus Prevention system will have 2 primary functions, detect and monitor. To save energy, the machine will initially start out in low-power mode. For the detection functionality, the machine will make good use of its distance sensor to detect if an object is close enough to the machine. If an object is close enough, the distance sensor will send an interrupt signal to the microcontroller to wake up the entire system. The image processor will then do its job and determine if the object in front of the machine is a human. If the object is a human, then image processor will then check if the person is currently wearing a mask. If a mask if present, then nothing will happen. If the mask is not present, then a mask will be given to that person.

As for the second function of monitoring, it will continuously scan the temperature of the person standing in front of the system. While the image processor is doing its job, the temperature sensor will monitor the temperature. If the person’s temperature is above certain threshold, the temperature sensor will send an interrupt to the microcontroller to alert everyone within the vicinity that a person with a fever has just entered the area. Building managers will not have to worry about the noise complaints as the alert will stop within a couple of seconds.

While the detection and monitoring will be the two primarily functions of the system, the system will also be able to gather data. Since we will be selling the system for cheap, with extremely small, likely 0, margin, this will put our company in the red within a couple of months. We will need a secondary source of revenue to keep the business going, to continue to support our customers. By gathering user data, we will be able to prepare user statistics on health and behaviors. Such data will be extremely valuable to scientists and researchers, who are trying to study trends and correlations. With the data, they will be able to potentially predict the next hot spot of the pandemic or even prevent it beforehand with help of government intervention. When the virus is no longer here, researchers will still be able to use these data to learn more about the behavior of human. The data will be able to show how a person is likely to behave in a stressful situation. With user consent, we will be able to sell these valuable data to the scientists and researchers to help aid their studies and raise capital to fulfill future purchase orders.

1.4 Requirements Specifications

A project will not go anywhere without having a specific set of requirements for the developers to achieve. The requirements ensure that the final product meets or surpasses the outlined measurements and performance. These requirements not only outline the functionality of the Coronavirus Prevention system, but they also lay the foundation for the design and implementation of the machine. These are the standards in which to the final product is to be built upon.

The purpose of the Hardware/Software requirements section below is to outline the specific, verifiable, and measurable outputs the product must achieve. At the bare minimum the finalized product must meet these requirements specified below to meet the intent of the project. These requirements will be verified as per the verification section in TESTING AND EVALUATION.

Further from these strict hardware/software requirements below, the more in-depth functionality requirements and deliverability requirements are listed in section 1.4.3 Functionality.

1.4.1 Hardware Requirements

The hardware requirements below detail what will need to be met in terms of hardware design for the project. They are focused on and may include physical requirements or constraints but are not limited to these two. For example, area, physical interfacing needs between boards and peripherals, feasibility of implementation on PCB, etc. would be located here.

1.4.1.1 Body and Mask Case

The body will provide mounting for the various parts of the system. A mask case will be provided to hold face masks with a sliding door.

1.4.1.2 Mask Dispenser Motor System

Will provide a mask dispenser activated by software. Upon activation the system will provide a mask using motors to open the Mask Case.

1.4.1.3 Button

Will provide an override button to manually activate the Mask Dispenser Motor System.

1.4.1.4 Temperature Sensor

Will provide a human body temperature sensor to the software mounted to the main case.

1.4.1.5 Distance Sensor

Will provide distance measurement intended for accurate temperature measurement to the software.

1.4.1.6 Alert System

Will provide a software-controlled alert system in the case the temperature reading is above a threshold. The alert system will consist of a speaker.

1.4.1.7 Microcontroller Unit (MCU)

Will provide a software-controlled interface for control and status monitoring of peripherals – temperature sensor, distance measurement unit, alert system, and mask dispenser. Will provide I/O to the Image Processor through the PCB.

Total I/O required is 4 to the peripherals and 1 to the Image Processor.

1.4.1.8 Camera

Will provide a camera for the Image Processor suitable for imaging human faces within 50 feet for mask detection.

1.4.1.9 Image Processor

Will provide image processing outputs to detect whether the person is NOT wearing a mask. Will provide I/O to the Microcontroller Unit through the PCB for status and control. Will connect to the Camera.

1.4.1.10 AC-DC Converter and Power Delivery

Will implement a 120V AC-DC voltage converter to provide power for low-power components in the design – MCU and Image Processor.

1.4.1.11 Custom PCB

The AC-DC voltage converter, temperature sensor, distance measurement unit, alert system, microcontroller will be located on the PCB. I/O will be provided for the attached back-end Video Processing Unit to connect to the PCB.

1.4.2 Software Requirements

The software requirements are everything required from the software-sided components for the completion of the project.

Microcontroller

1.4.2.1 Mask Dispensing Control

The MCU will provide capability to drive the motors upon activation and open the mask case.

1.4.2.2 Peripheral Control

The MCU will provide control and status monitoring of the Temperature Sensor, Distance Sensor, and Alert System.

1.4.2.3 Image Processor Interrupt

The MCU will be able to detect a signal from the Image Processor prompting a person with no mask has been detected and react accordingly by opening the mask dispenser and sounding the alert system.

Image Processor

1.4.2.4 Facial/Mask Detection

Will be able to run algorithms for facial detection and correctly determine whether the video from the camera contains a person and if so if they are wearing a mask or not. This will be done in real-time.

1.4.2.5 Microcontroller Interrupt Generation

Will be able to generate an interrupt to the MCU upon a person with no mask detected. The MCU then must react accordingly.

1.4.2.6 Statistics Tracking

Will have a statistics system to track number of people detected passing by, number of people wearing/not wearing masks.

1.4.3 Functionality

The requirements of all the functionalities of the system, as well as how they are going to be implemented are listed in detail in Table 1-1 Requirements and Deliverability Table down below. These requirements provide a rough outline as to how the final product is going to look and function.

Table 1-1 Requirements and Deliverability Table

Requirement	**Requirement Deliverability**
The production cost of the entire system will not exceed $1000 in investment.	The components required for the development of the final product will mostly be second-hand to save on cost.
The system shall be sturdy and reliable enough to sit in one spot for years on end.	The body of the system shall be built out of acrylic/hard plastic to provide both a sturdy foundation and cut back on material cost.
The resolution of the camera shall be no less than 1080 interlaced resolution	We shall purchase the correct device that meets this requirement
The viewing angle of the camera shall be no less than 160 degree of view	We shall purchase the correct device that meets this requirement
The camera needs to have well-calibrated color and not just black and white	We shall purchase the correct device that meets this requirement
The power delivery shall be able to produce at least 30 Watts under sustained load	We shall design the power delivery circuit and use quality products that meets this requirement
The regulator used for the power delivery shall able to withstand the sustained heat output of at least 30 Watts	We shall purchase the correct device that meets this requirement
The door-opening-motor shall be able to move the door in the 90-degree range	The software that controls the motor shall be tuned manually to meet this requirement
The solenoid shall be able to extend far enough the lock the door in place such that people will not be able to force it open without the proper procedures	We shall purchase the correct device and manually tune it until it meets this requirement
The microcontroller shall be able to interact with at least 5 devices at once	We shall purchase the correct device that meets this requirement

Requirement	**Requirement Deliverability**
The temperature sensor shall have the margin of error of +- 1-degree Fahrenheit or less	We shall buy the correct device that meets this requirement
The range sensor shall have the range of at least 2 meters	We shall buy the correct device that meets this requirement
The image processor shall be able to accurately process 1 image at a time	We shall program the processor and design the algorithm in such a way that this requirement is met
The image processor shall support a form of database either in the MySQL fashion or simply as a text file to store data	We shall program the processor correctly to meet this requirement

1.4.4 Specifications

Below sub-sections contain the design specifications that we wish to adhere to for our project. We will do our best to fulfill these specifications such that the final product will perform as we discussed in the following sections.

1.4.4.1 Hardware Components

Mentioned in Table 1-2 Power Delivery Parts below are the are the components necessary to build a reliable power delivery system that will power the entire system. The design of the power delivery is a typical design that has been around for a few decades. If it works well, then there is not a need to change it.

Table 1-2 Power Delivery Parts

Components:	**Quantities:**
Wall Plug	1
1N4001 Diode	7
Capacitors (F):	
470uF (Polarized)	4
100nF	4
Step-down Transformer	1
12V Voltage Regulator	1
Breadboard	1

Components:	**Quantities:**
Wires	~10
Alligator clips	4
Soldering iron	1
Oscilloscope	1
Multimeter	1
5V Voltage Regulator	1
3V Voltage Regulator	1

Now, on to the main system components. Table 1-3 Main System below shows the full list of all the components that are used in the main system. The system incorporates both a microcontroller and a high-power image processing unit with the microcontroller being in charge of the hardware control and the image processor being in charge of the software control. Other miscellaneous equipment that not in the system itself but are necessary for the assembly as the system are also presented in the table. The components are not sorted in any particular order. They were only split into 2 different groups: the main body group and the internal group. The main body group consists of components that can be seen from the outside of the robot without opening it up. The internal components group, as implied, consists of components are essential to the functionality of the machine and cannot be seen, at least not easily, from the outside.

Table 1-3 Main System

Components:	**Quantities:**
Main body	
Acrylic Sheets	6
Drill	1
Hot-glue Gun	1
Acrylic Cutter	1
Push Button Switch	1
Camera	1
Distance Sensor	1
Temperature Sensor	1

Components:	**Quantities:**
Internal:	
Motor	1
Solenoid (door stopper)	1
Microcontroller	1
Image Processor	1
Speaker	1

1.4.4.2 Technical Specifications

For an effective circuit design and an efficient hardware set up of the Coronavirus Prevention system, many specifications will have to be realized before the design process could begin. Without a specific design specification, the design may not work correctly, or it may even explode on first boot. Full system specifications are listed in Table 1-4 Technical Hardware Specifications down below. The specifications are tentative and are subjected to change as the project continues to evolve.

Table 1-4 Technical Hardware Specifications

Component	**Specification**
Wall Plug	USA type A plug standard (Compatible with type B sockets)
Power Delivery	120V_{ac} at 60Hz compatible
Capacitors	50V rated
Transformer	Accepts 120V input
Regulator	Able to withstand heat produced from supplying high output current
Full System Dimension	6 in x 6 in x 6 in
Weight	< 10 kg
Motor	Bidirectional
Microcontroller	Support multiple peripherals interfacing
Camera	1080p+ resolution, supporting multiple image formats
Image Processor	~500GFLOPS

Component	**Specification**
Temperature Sensor	Margin of Error +-1-degree Fahrenheit
Distance Sensor	>1-meter distance
Internal Storage	32GB

1.4.5 User experience

The user experience will be simple and convenient. The user does not have to do anything aside from standing directly in front of the line of sight of the camera. The integrated software will take care of everything else. As the person steps closer to the system, to system will wake itself up from sleep. The system will first take a picture of the person’s face. Then, after a couple of seconds of rigorous processing, the system will determine if the person is wearing a mask or not. If the person is wearing a mask, nothing will happen. If the person is not wearing a mask, the door to the mask container will automatically open, allowing the person to reach in a fetch a mask. While the image processor is busy processing the image taken, the temperature sensor will scan the body temperature of the person. If the person’s body temperature is above a certain threshold, the alarm will go off, else, nothing will happen. Aside from the automatic functionality of the system, there will be 1 single button available for the anyone to use. That button manually overrides all other running processes by terminating them, and upon up the door to the mask container.

2 RESEARCH

As the idea and image of the Coronavirus Prevention System become clearer in our minds, it is essential for us to gather current data on the disease, as well as information on our competitors so that we can position ourselves in a good situation to grow and popularize our product. Since we a new to designing products, we will also need to research technologies that we can make use of to make our task easier.

2.1 Market Research

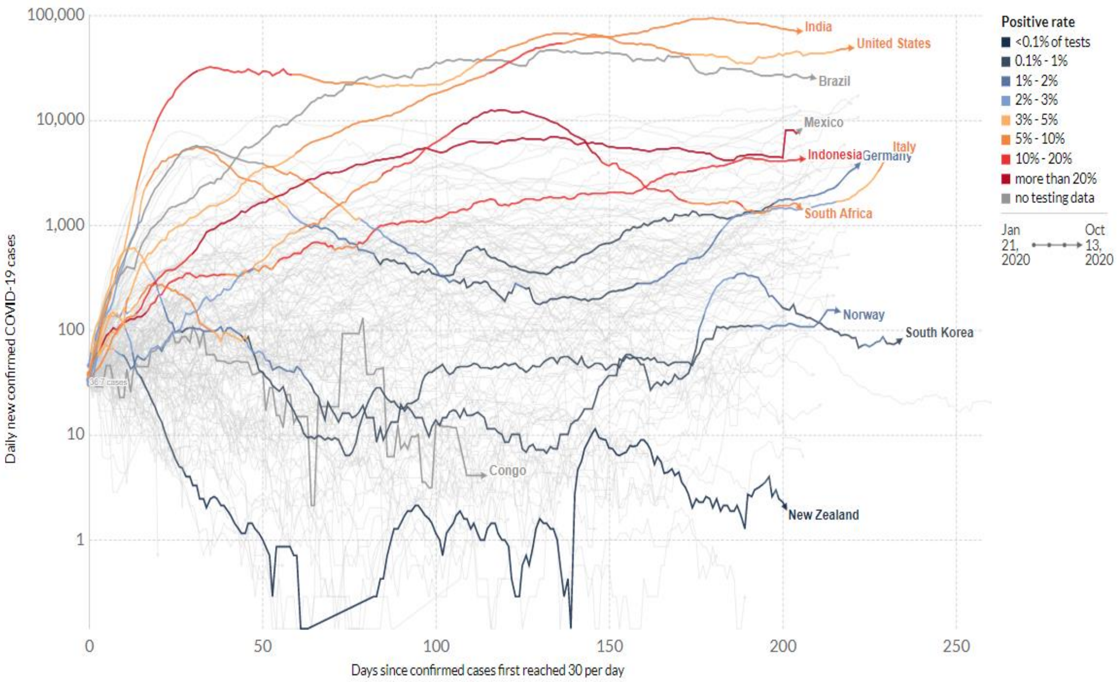
When first launching a product, market research is extremely important. If we have a good product, but we launch it at an unfavorable time, the product will do poorly. In contrast, if our product is highly relevant and is unique enough, it will fly off the shelf faster than the speed of sound. The following sub-sections will go over the current statistics of the novel Coronavirus as well as the timeline as to how long our product can stay relevant for. By design, this is a short-live product, but as long as we can capitalize on the fear, and the desire to stay healthy, then we should be able to popularize the product, even just for a short period of time.

2.1.1 Coronavirus Global and Domestic Statistics

According to the Centers for Disease Control and Prevention (CDC), the novel Coronavirus of 2019 has its origin traced back to November of 2019 in Wuhan, China. Many experts believe that the virus initially came from bats, where they further evolve to infect human who came into contact with bats' droppings. The coronavirus that causes COVID-19 disease is not the same as the common coronavirus that causes the common cold in human. The SARS CoV-2 virus is believed to be a mutation of the SARS CoV virus that once ravaged China back in 2003. Because the bats' bodies are relatively warm, the virus was able to prosper and mutate in such a way that the modern human immune system is no longer able to ward them off.

The novel Coronavirus mainly spreads through the direct contact of the droplets or bodily fluids of infected people, like saliva, tears, urine, etc. This means that most of the infected cases worldwide happened due to close and direct contact with the infected. However, recently, the CDC has confirmed that the SARS CoV-2 strand of the coronavirus had evolved enough to become airborne. Being airborne means that the virus is able to spread much further away from the source than before.

The pandemic spreads everything in the world, and it has turned the world upside down. The mysterious coronavirus outbreak in the Chinese city Wuhan, spread throughout Europe, even all the way to America, the virus has killed millions of people and shut down hundreds of countries. Since January, worldwide, more than 1.09 million deaths have been reported, with potentially many more being misdiagnosed or outright hidden by the corrupted regimes. The chart below shows the progress from January to October of the countries that are heavily affected especially the U.S. Within half a year time, more than 7.89 million cases of infection and 216,000 deaths have been reported in the United States alone. The United States holds the first place for the country with the highest number of infected people, followed closely by India at 7.30 million and Brazil at 5.12 million. This leaderboard is nothing to be proud of. Countries with better public awareness like Japan and Singapore only has 90,000 and 58,000 cases total, respectively. These numbers are 1% that of the number of infected people in the United States. Figure 1 shows the number of daily new infections of COVID-19 in some of the hotspots.

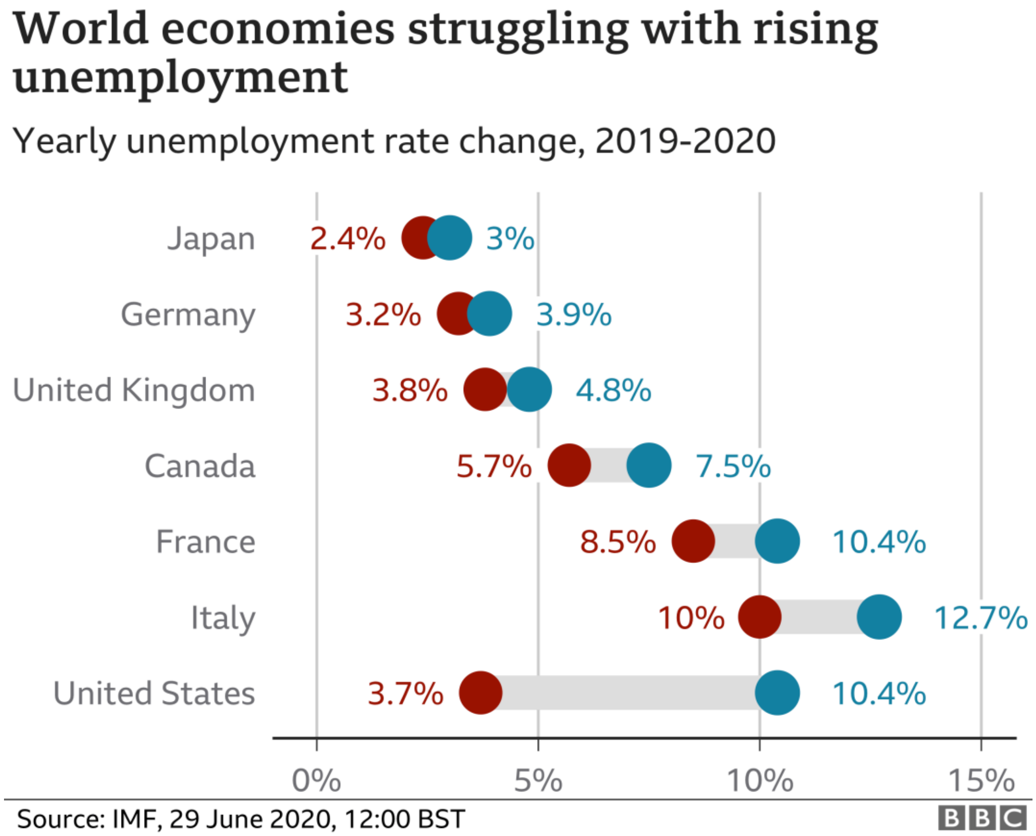


Source: European CDC – Situation Update Worldwide – Last updated 13 October, 10:35 (London time), Official data collated by Our World in Data

CC BY

Figure 2-1 Daily New Coronavirus Cases Trend

There is no way to tell exactly how much of the economy was damaged by the COVID-19. Because of the extremely fast rate of infection and the high mortality rate of the SARS CoV-2 strand, the COVID-19 pandemic had forced hundreds of countries to shut down their economy and close their border for a couple of months. According to the British Broadcasting Corporation, experts predict that the global Gross Domestic Product for 2020 will go down for at least 2.4%. This global economic impact had not been seen since the Great Recession of 2007. This prediction came from the drastic increase in the unemployment rate of multiple countries. Year to year, United States' unemployment rate went from 3.7% to 10.4%. This means that, with the 206,233,681 working age population, according to Trading Economics, the number of unemployed people is 21,448,303 people. Other countries like Italy and France have roughly a 2% year to year increase in unemployment. Italy went from 10% to 12.7% and France went from 8.5% to 10.4%. More information on the unemployment rate in other countries are given in figure 2 down below. Year to date, the Dow Jones industrial has dropped close to 8%. The FTSE 100 Index, which tracks the economy of the United Kingdom has dropped for more than 20%. While the stock market does not directly reflect the health of the economy, as the stock market mainly functions on human emotion, it is still a good indicator of the trend direction that the economy is headed.



Source: IMF, 29 June 2020, 12:00 BST

Figure 2-2 World Unemployment Rates

2.1.2 Potential Cures

COVID-19 is continuing to grow and surpassing more than 9 million cases worldwide. However, there is no cure or vaccine for the virus as of right now. All the scientists in the world are working hard to research and formulate the vaccines to slow down the pandemic. When the pandemic starts spreading in in the United States in March, President Donald Trump has been advertising the malarial drugs chloroquine and hydroxychloroquine as the ultimate cure for the disease caused by the novel coronavirus. He coerced the Food and Drug Administration to approve for the emergency usage of the drugs hydroxychloroquine and chloroquine to combat against the virus. Later, the scientists, as well as Dr. Anthony Fauci noted that while these drugs may treat the disease, without the appropriate studies being conducted, these drugs could prove to be dangerous. These drugs cause higher heartrates in patients, which could potentially increase the risk of a heart arrhythmia.

Many scientists believe that another potential treatment with an extremely high probability of success is the use of antibodies from the body of people who have already gotten the virus and have already made a full recover. The antibodies can be obtained from survivors' blood plasma who have fully recovered from COVID-19 for at least two weeks. Antibodies are what our body produced to fight against uninvited intruders that are damaging our bodies. After the body recovers from the illness, the antibodies remember the weakness of that illness for a period. The memory period of the antibodies is at least 5 months, according to the Health Line Magazine. While the antibodies still remember the virus, if you get infected again, your body will have a much easier time to fight against the invaders. For some diseases, the antibody memory lasts for a lifetime, like with Chicken Pox. For other illnesses like influenza, the antibodies stick around for a year at best. Scientists are still unsure why this is the case, and they are trying to figure out which of the two categories the novel coronavirus belongs to. If the virus triggers the lifetime immune response, then the antibody treatment will be extremely helpful. However, if the COVID-19 disease can only get the antibodies to stick around for a couple of months, then there is very little hope for the antibody treatment. Even if the disease somehow triggers the lifetime antibody response, it is incredibly challenging to have the antibody treatment be a widespread treatment. Since there are not that many blood donors to begin with, obtaining positive SARS CoV-2 antibody plasma will put a lot of pressure on the United States' blood bank. As the results, many pharmaceutical companies are now putting all their resources into research and trials of their very own version of synthetic antibodies to help find a cure to the disease. These companies are raising capital left and right to get their drugs into clinical trials, and subsequently, on the market as soon as possible. Right now, the best way of treating the disease is not get it in the first place, which means staying healthy and avoid social gatherings.

2.1.3 Potential Vaccines

There is currently no authorized vaccine to prevent the infection of SARS CoV-2 in the United States. Traditionally, it takes roughly 5 or more years for a vaccine to be approved for public use. However, due to the severity of the disease, many companies are applying for early approval of their vaccines before they even complete the required testing. According to the UK-based research firm GSK, a typical vaccine needs to go through 3 different trials before they can be officially approved for public use.

Thanks to the government funding, the experimentations of vaccines and medicines for COVID-19 are being conducted with unprecedented speed and scale. The research and approval of a vaccine can be described as a pipelined process. Everything first starts out at the academic research phase. When the researchers have a general idea of which compounds to use, they produce a small sample and pass it along slowly from one laboratory to another laboratory for peer reviews and testing. After they have their peers' approval, the vaccine moves on to the pre-clinical animal testing. And lastly, after being successfully tested on animals, the vaccine moves to a 3-phased clinical trial in people. The development of a vaccine cannot be rushed since it is going to take a lot of time to go from the first discovery to the marketplace. According to the Centers for Disease Control and Prevention, the United States government has sponsored many pharmaceutical companies to produce a viable vaccine by January of 2021. The speed up in vaccine production is dubbed: "Operation Warp Speed". It is possible condense the 5 years trial time of a vaccine down to 18 months by the destruction of the pipeline and parallelize the steps. To cut down the time even further, the operation shortened to required time for the phase 2 and phase 3 human trial. This time shortening of the human trials received many backlashes from the medical communities as the human trials are extremely important, and that most potential vaccine failed the phase 3 human trial. If we recklessly shorten the trial time, we will not know for sure if the vaccine will cause any adverse effects to the vaccinated. Even if the vaccine is completely safe, with no adverse effects, it will be extremely difficult to enter mass production within 10 months period. There will be a shortage of vaccine, and the price of each shot will remain high for a long time. Therefore, the CDC stated on their website that after approval, the COVID-19 vaccine will not be recommended for children and will be recommended to certain high-risk groups.

2.1.4 Current Prevention Efforts

As the COVID-19 wreaked havoc across Europe and the American continent, back in late March of 2020, many countries around the world had issued lockdown to prevent further spread of the pandemic. Some places were on lockdown for 3-4 weeks like with the case of Japan, while others, like the New York state of the United States, were on lockdown for more than 4 months. The mandatory lockdown, while succeeded in preventing the exponential expansion of the COVID-19 pandemic, it forced many businesses, both large and small, into bankruptcy.

After the mandatory lockdown has been lifted, many other prevention strategies were put in place to stop the pandemic from coming back. Some were strictly enforced, while others are left to the discretion of the individuals. One of the more prominent prevention strategies is to limit the social gathering size and enforce the 6 feet social distancing rule. Figure 3 below shows the number of weekly COVID-19 tests and the percentage of newly infected cases. You can see that during the lockdown, the percentage of new positive cases dropped drastically. After the lockdown, however, even though the number of new cases per week does not increase, it does not decrease either and remains relatively flat. This means that the social distancing effort is not that effective, though the spread is somewhat contained. Contained here means that it is not spreading at an unreal rate that could overload the medical system anymore.

Even though the COVID trend seems to be under control for the past couple of months, Winter is quickly approaching. Colder countries are already experienced the comeback of the COVID-19 pandemic. According to Reuters, Madrid, the capital of Spain, experiences 859 COVID-19 cases per 100,000 people, making it Europe’s worst COVID-19 hot spot. Starting from the beginning of October, Madrid and nine nearby municipalities went back into a second lockdown as social distancing alone was not enough in the cold weather. Other countries, like the United Kingdom, are in talk to plan for a potential of a second lockdown as well.

The virus is primarily transmitted through respiratory droplets when people contact in close distance (less than 6 feet). When people cough, sneeze, or talk, it enters the body and leads to respiratory failure, even death if the patient has an underlying disease. Most of the patients had initial symptoms like the flu. However, there are some patients who do not have symptoms or mild symptoms, and it will be ignored. The incubation period of COVID-19 is from 2 to 14 days depends on the immune system of each patient. Therefore, it is necessary to perform a test to accurately. There are also some recommendations from CDC to prevent of getting infection such as avoid touching your eyes, nose, and mouth, use a face mask when physical distancing, clean and disinfect constantly etc.

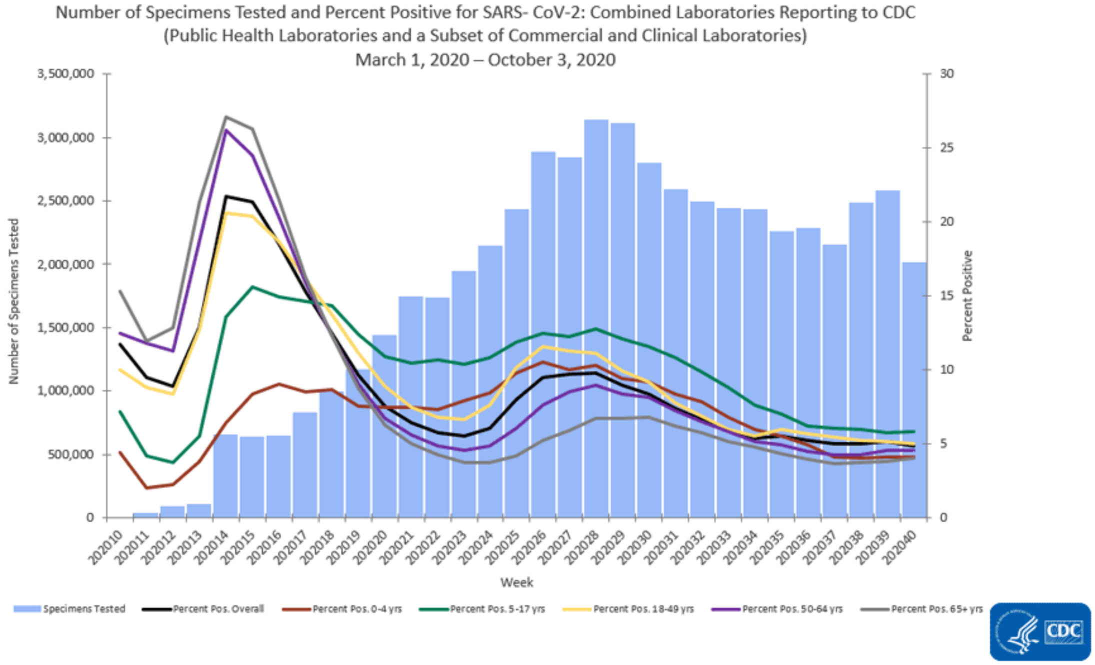


Figure 2-3 COVID Testing Trend

2.2 Competitors Research

As the spread of COVID-19 continues, technology has dramatically evolved to quickly adapt to the current situation and create solutions that can help people to conveniently protect themselves through existing services. The important thing is that COVID-19 is a new disease, so it is not much information about the risk factors for serious illness. Innovative technologies can be game changers to reduce the damage of coronavirus, while waiting for the development of vaccines and treatments. There are many innovative designs and machines that are being used worldwide to prevent further outbreak in the COVID-19 cases. These designs aim to limit the opportunities of direct contact with contaminated surfaces or get infected with the virus from talking with other people who are already infected with the virus.

2.2.1 Mask Vending Machines

The first vending machines have been around since 1883 and was invented by Percival Everitt and introduced at London. Until 1888, the first vending machine finally appeared in the United State. It was designed by the Thomas Adams Gum Company in order to serve selling on New York City train. After the first beverage vending machine was invented in France, William Rowe, American inventor, created the cigarette vending machine, however, with the concerns of underage buyers, it become uncommon. With the development of technology in the 21st century, vending machine is getting more popular with the advanced designs that offers user-friendly environment such as non-contact payment, facial recognition and social media connectivity.

With the COVID-19 pandemic staring at our backs, people have turned to the try and true vending machine as a mean of preventing the disease. People replace to typical content of drinks, food, and office supplies with the critically need masks so that everyone can have easy access to them anytime, anywhere. The idea of mask vending machine can be great choice for a lot of business since they do not require a lot of dedications to install and it is easy to keep them in good conditions such as restocking and collecting the money. However, the cost of vending machine will be $3000 to $10,000, depending on the sizes and features, and it is needed to be placed in location with a lot of traffic that people will be able to notice.

Generally, the purpose of vending machine is to provide automation, deliver the mask products to the customer and it is operated under the control of mechanism. Dispending mask products automatically can save a lot of time as well as making our life more convenient.

Vending machine interface is separated into three main part including: “Choose”, “Browse” and “Confirm”. In the other words, when the customers insert an amount of money, they will be able to select the available products and after that, the machine will generate the items is selected, customer can take out and get the changes back.

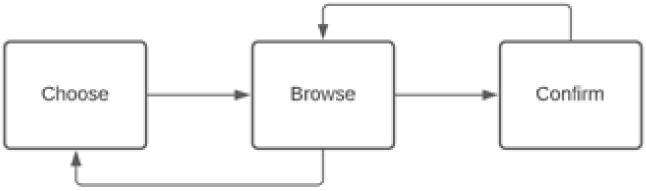


Figure 2-4 Concept Model of Vending Machine

Nowadays, with high level of technology, vending machines also have progress in design to improve the interactive user experience such as touch display, different advanced payment system and using embedded computing power. In general, the vending machine comprise two main parts which are spiral coil and coin mechanism. Firstly, when the customer inserts the coins, vending machine use light sensors to determine the value of coins based on sizes and metal type. The coins will continue go to sorting selection, otherwise they will be returned back to the customers if they are rejected. When inserting money process completed, the machine will wait for user to press button and choose the product. There are variety of products with different range of price, the machine will keep track of the payment and let the customer choose the products they want. If the money inserted is not enough for the product, the machine will ask the customer to insert more, otherwise, they have option to cancel the payment. When the product is selected, it will fall into the release tray after spiral coil is turned 360 degree. Finally, the machine will calculate the number of changes based on the number of products have been purchased, coins will be released one by one until it finishes the payment to the customers.

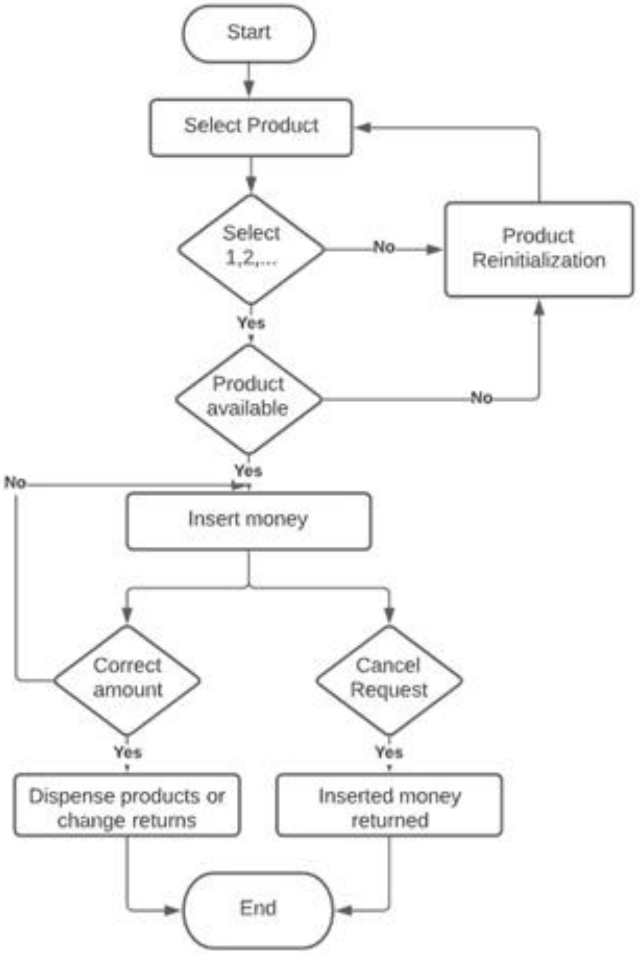


Figure 2-5 Vending Machine Flow Chart

Within the software and hardware requirements, the vending machine is able to work correctly and deliver the products to the customer. After a period of time, it is needed to restock the products and collect the money inside the machine. Maintenance of machine does not require much of human power and using automation in delivering is such a great idea to better customer-engagement for the brands as well as satisfy human needs in daily life. Vending Machine has become wide usage and mask vending machine is expected to have significant growth during the pandemic because of the convenient method of delivery products.

2.2.2 Automatic Hand Sanitizer Dispensers

There are up to 10,000 bacteria per square inch, with the spread of germs, it can lead to different illness around the world, or transfer the illness to different person. It is studied that pathogens can be transmitted from a person to a person through infections on the body or skin to skin with a person who had infected since the skin is considered as one of the main immune defenses for our body in daily life, and the easiest solution to protect our immune system is to make our hands clean. Washing hands after bathroom, coughing, or playing outsides is common, however, with the rapid changes of viruses' structure, it is necessary to wash our hands after touching different public surfaces such as doorknobs, or picking up groceries, and before touching our face. Moreover, places in the bathroom whether it is at home, restaurant and shopping mall are filled with germs from the toilet, sink because people do not give a lot of attention on those things.

The common way to sanitize hand is to washing hands with hand soap, which is found in every public bathroom. However, washing with soap could often take time away from the fast-paced work schedule and it needed to follow the hand washing instructions properly in order to prevent the spread of germs.

It is proven that the solution with more than 60% or more of ethanol or isopropanol could reduce the significant the transmission rate of disease through germs from hands which reduces the transmission of illness. Therefore, a lot of hand sanitizer have been used in public places to prevent the spreading of the virus as well as keeping hand hygiene. As the increased usage of hand sanitizers, it is required to design new dispensing containers to meet human's needs in fast-paced technology. There are many models have push pump to dispense the hand sanitizer to customer's hands, however, this design still requires contacting with the push button to get the liquid. Approaching the next level of advanced dispensers, the Automatic Hand Sanitizer focuses on automation, alcohol-based hand sanitizer dispenser is widely used in hospitals, workplaces, restaurants and many more. Sanitization is defined as reducing the number of germs and the occurrence and growth of viruses, and within most of sanitizing methods, alcohol was determined to be useful for decreasing the bacteria on our hands.

The design of automatic Sanitizer Dispensing machine using Arduino microcontroller with ultrasonic sensor is an effective idea for non-contact hand sanitizer dispenser. It is easy to program on microcontroller since it comprises of ADC and DAC. The input of the microcontroller is using ultrasonic sensor, with the purpose of sensing the distance within our hands. When the ultrasonic sensor senses the hand within the distance of 7 cm or less, the relay gets the pulse from microcontroller and make the pump runs. The pump is between 3V to 12V submersible type, which is able to pumps out the sufficient amount of hand sanitizer onto hands. The advantage of this automatic hand sanitizer is there is no contact needed since the machine is completely automated, which means, the probability of getting infected from direct contact is zero percentage.

Hand sanitizing dispenser is available in different sizes, capacities and either can be self-standing or with hygiene stations. With the fact that, this machine is user friendly, low maintenance cost, and easy to operate, this hand sanitizer dispenser market was valued more than $74 million in 2020, and it is expected to expand quickly in the near future.

The conventional vending machine aside, we do not consider the hand-sanitizer to be strictly a competitor. The hand-sanitizer is designed to keep the hand clean, so that when a person touches his/her face, mouth, or any other bodily opening, he/she will not ingest the virus from their hands. Our glorified mask dispenser system works differently. It provides a way for people to protect their air duct, so that they do not breath in contaminated air. On the grand scheme of things, the hand-sanitizer compliments our Coronavirus Prevention System well. They are more like best friends, rather than market rivals. The target audience for the two products, while similar, are different.

2.3 Technology Research

In this section, we will be exploring the potential technologies that may help us in the implementation of our ideas. None of us have ever design a full system from scratch, nor do we have any research experience. We reckon that the research portion will be the most challenging part in this project. To lighten up the load, our professor advised us to look at other similar ideas and try learning from them. He said in one of the lectures that it would be better would us to look at existing product and figure out how to expand on them, or how to use them in our project would be better than designing everything from scratch. After researching relevant technology, it will be our jobs to figure out how to bridge them all together to form a congruous system.

2.3.1 Microcontrollers

Microcontrollers, in lay man’s term, is just a miniaturized, simplified computer. It is generally used for embedded applications where power efficiency is greatly valued, and where an operating system is not needed. Microcontroller includes a processor, memory, and input/output peripherals on a single chip. Operating System-supporting devices are extremely vulnerable to failure due to a random bit flip that causes the OS to be corrupted. For simple applications where an OS is not needed, a microcontroller is a perfect candidate for the job. Microcontrollers are essentially miniature computers that are designed to controller other simple components without the need of a complex OS. The term microcontroller may not seem to be too common, but microcontrollers exist everywhere around us. They can be found in vehicles, basic robots, radio transceivers, etc.

Like a computer, the microcontroller manages its connected devices by first interpreting data it receives from its input/output peripherals. After the data is successfully received, it is stored in the memory and wait for the processor to process the data. When the data is processed, the processor send out a command to the connected devices via input/output peripherals. While for most small projects, 1 microcontroller is enough to manage the entire system, for larger industrial applications require an array of multiple microcontrollers working together in harmony. Let us take an airplane as an example. There is an individual microcontroller for each of the functions like cruise control, direction control, fuel injection, oxygen control, etc. Unlike a multi-CPU computer that requires a high bandwidth data bridge for the CPUs to communicate with each other, microcontrollers can be set up to work together with minimal effort.

In the past, most microcontrollers are programmed using assembly language. Working directly with hardware via assembly languages allow the microcontrollers to manage the memory usage of the programs more efficiently, thus potentially increase the functionalities of the microcontrollers. Thanks to the advancement in compilers and interpreters, microcontrollers can now be programmed in the C programming language with the comfort of an integrated development environment. The compilers and interpreters can now almost perfectly translate high level programming languages to assembly level instructions. Some of the more advanced microcontrollers even support scripting languages like Python and JavaScript. Microcontrollers come in all shapes and sizes, ranging from a simple 4-bit controller to a more complex 64-bit controller. More basic microcontrollers tend to support the Reduced Instruction Set Computing, or RISC, architecture, while the more advanced microcontrollers are likely to support the Complex Instruction Set Computing, or CISC, architecture. CISC provides more instructions with more addressing modes than RICS. CISC is also much easier to implement and is more efficient in its memory usage. However, there is a big trade off. CISC tends to run into performance degradation issues with small, short-burst instructions due to the much higher number of clock cycles per instruction as compared to RISC. Two of the more popular microcontrollers, the ATmega328 and the MSP430G2553, will be discussed in sections 2.3.1.1 and 2.3.1.2 below.

2.3.1.1 ATmega328 Microcontroller

The Arduino development platform is likely to be the world most popular development environment. ATmega328 is the microcontroller that is used on the Arduino development boards. Since the microcontroller has a strong affiliation with the Arduino platform, there is a strong community support regarding the development on this microcontroller. It is to go-to board for both beginners and advanced developers. ATmega328 is an 8-bit microcontroller that is based on the RISC architecture. Notable features of the ATmega328 microcontroller are noted in the Table 1 down below.

Table 2-1 ATmega328 Features

Component	Feature
CPU Speed	20 MIPS
Program Memory	32KB of flash memory
EEPROM	1KB
Communication Protocol Support	UART, SPI, I2C
SRAM	2KB
General-purpose I/O lines	23
A/D Converter	6-channel 10-bit

Component	**Feature**
Power modes	5
Timers	2 8-bit timers, 1 16-bit timer
PWM enabled	6 pins
General-purpose registers	32

These features, while not overwhelming, provide the necessary development environment for most developers. Since this microcontroller is based on the RISC architecture design, according to Microchip Technology, ATmega328's parent company, this design helps the microcontroller reach the throughput of 10 times faster than that of a conventional CISC microcontroller. The overview of the design of the ATmega328 microcontroller is presented in the Figure 7 down below. Because the ATmega328's processor is an 8-bit processor, the design of the chip was able to be simplified in such a way that the cost of production is much lower than other competitors on the market. Thanks to its support for multiple communication protocols, especially SPI, and the plethora of I/O lines that is has, it is possible to concatenate multiple microcontrollers together to handle more complex tasks that an 8-bit processor may not be able to accomplish.

Block Diagram

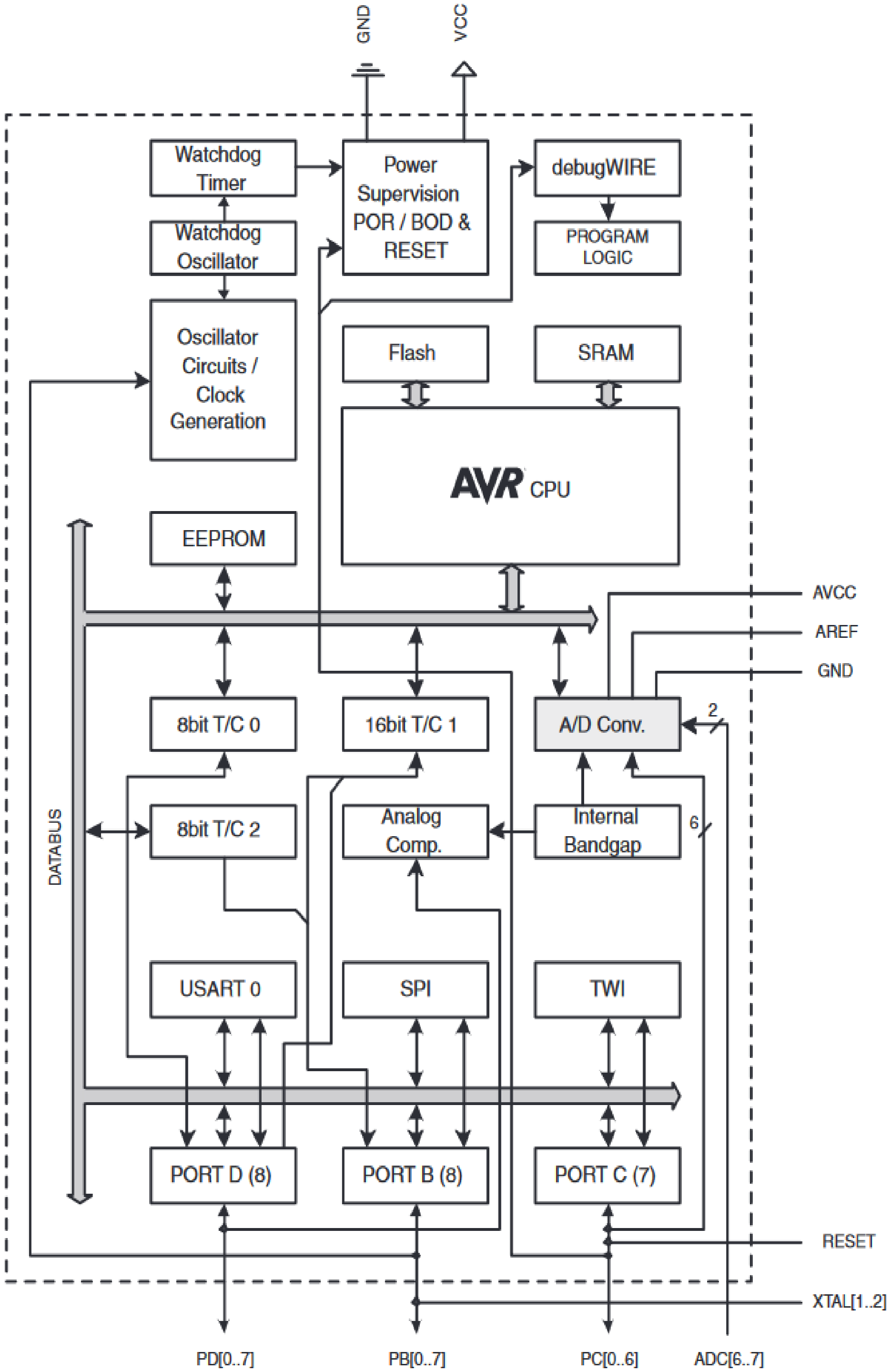


Figure 2-6 ATmega328 Block Diagram

2.3.1.2 MSP430G2553 Microcontroller

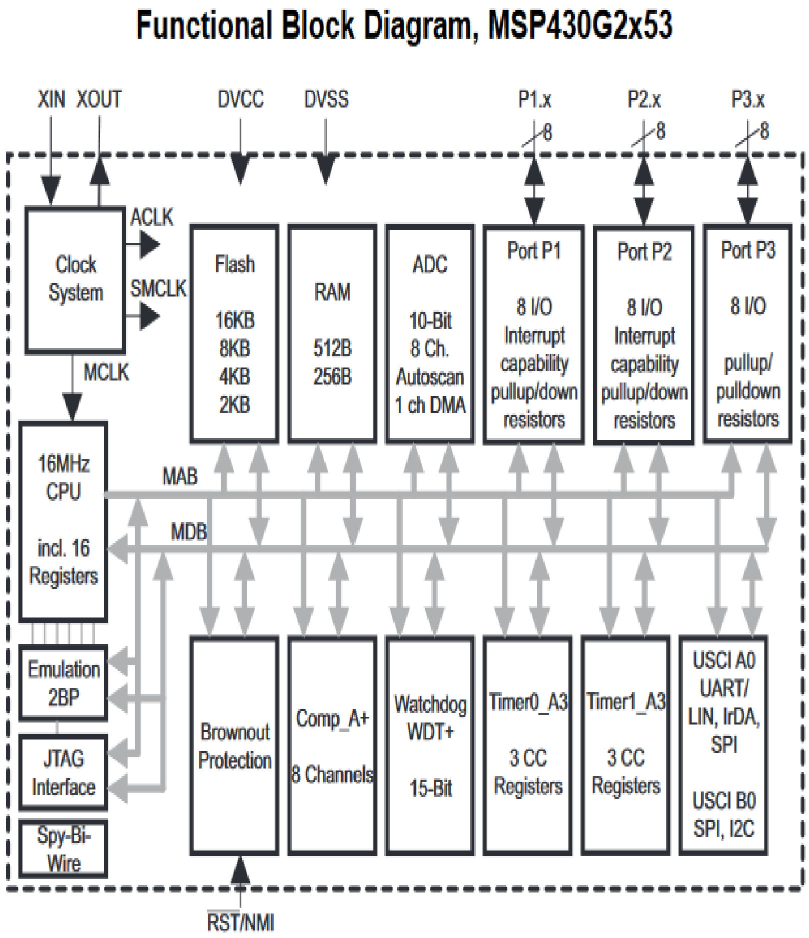
Being one of the world's largest manufacturers of electronics, Texas Instruments' solutions of microcontrollers for embedded projects does not lack behind anyone. One of Texas Instruments' most popular choice is the MSP430 microcontrollers lineup, specifically the MSP430G2553. The MSP430G2553 is the favorite of many embedded instructors as well as professionals because of TI's extremely detailed datasheets, as well as excellent customer supports. Like the ATmega328 microcontroller, the MSP430 line of microcontrollers are also based on the RISC architecture. However, different from the ATmega328, the MSP430G2553 ports a 16-bit processor. This means that in theory, the MSP430G2553 can compute much larger numbers and do much more complex computations as compare to the ATmega328. Notable specifications and features of the MSP430G2553 is noted in the Table 2 down below.

Table 2-2 MSP430G2553 Features

Component	Feature
CPU Speed	16 MHz
Program Memory	16 KB of flash memory
Communication Protocol support	UART, SPI, I2C
SRAM	256B
General-purpose I/O lines	24
A/D Converter	8-channel 10-bit
Power modes	5
Timers	1 Basic internal clock, 2 16-bit which has 3 Capture/Compare Registers
PWM enabled	3 pins
General-purpose registers	16

Even though the MSP430G2553 microcontroller has a 16-bit processor, it does not utilize all the memory addresses. In fact, the size of both the SRAM and the flash memory are smaller than those of the ATmega328. While this means that the program written to the microcontroller must be small, it also enables the MSP430G2553 to connect to many more devices than the ATmega328 as there are many more usable addresses left. Although the cost of producing a RISC microcontroller is already small due to the simplification of the instructions, the MSP430G2553 aims to cut down the initial cost of production even further than other RISC microcontrollers by reducing the memory size. Although the microcontroller has small memory, but in return, it offers a much greater expansion potential than other microcontrollers on the market. The block diagram for the

MSP430 family is provided in the Figure 8 below. You can see that although both the MSP430G2553 and the ATmega328 are both designed base on the RISC architecture, the implementations are very different. ATmega328 has more usability and quality-of-life features like large memory, while the MSP430G2553 focuses on security and future expandability.


Functional Block Diagram, MSP430G2x53

NOTE: Port P3 is available on 28-pin and 32-pin devices only.

Figure 2-7 MSP430G2553 Block Diagram

2.3.2 Sensors

In order to implement our ideal disease prevention, mask detection robots, it is necessary to have multiple sensors for each of the tasks. While sensors may not seem to be important in the design of a mask detection robot, without them, figuring out the relative position of a person, measuring their temperature, or providing on-demand masks will be exponentially more difficult. Although we do not have any background in the field of sensors technology, implementing a sensor system in our robot may not be too troublesome.

With the abundant of the sensors on the market, we do not have to design them from scratch. Rather, we only need to purchase the correct model for our purpose.

2.3.2.1 Temperature Sensor

A temperature sensor is a device that measures the temperature of its environment. It comes in many different shapes and sizes. In the past some of the most common temperature sensors are the alcohol and mercury-based thermostats. At normal condition, both alcohol and mercury are liquid that have a low boiling points and are extremely sensitive to heat. They are usually put into vacuum-sealed tubes such that as the ambient temperature moves by 1 degree Celsius, the liquid in the tube will expand or contract by 1 millimeter in the same direction. While these are not too accurate as the liquid can slowly permeates through the container and evaporates, they were close enough and are cheap enough for people to buy.

In modern day, gone are the liquid-based thermostats. In their place, the semiconductor-based temperature sensors took reign. There are two basic types of these sensors: contact types and non-contact types. For the contact types, the sensors are required to be in physical contact with the object. They use the principle of conduction to monitor the change in temperature. As the temperature of the contact body increases, more heat is transferred into the probe, which changes the temperature reading of the sensor. As for non-contact temperature sensors, they use the radiation of the object to measure its temperature. All things will lose heat to the environment through detecting and absorbing radiation. There have been several experiments to test if a hot object will lose heat in a vacuum. The results for these experiments have always been positive. Although there is no air in the vacuum, an object still gets colder with time because it radiates off its heat in the form of infrared radiation.

Each of the two types of temperature sensors consists of 3 sub-types. These are electro-mechanical, resistive, and electronic. One of the examples of the electro-mechanical type is the bi-metal thermostat. These are sensors which consist of 2 types of metals with different expansion rates. They do not change much as temperature changes, so they are not fit to monitor something with constantly changing temperature. Their primary usage is a an on/off switch at certain temperature threshold. These switches can be found in old air conditioner designs. An example of the resistive type sensor is the thermistor. Its name came from the combination of the words "thermal" and "resistor". These are resistors whose resistances change as the temperature changes. Most thermistors have negative temperature coefficient. In other words, as the temperature increases, the resistance goes down. Lastly, representing the electronic type sensors is the thermocouple. The thermocouples are small in size and respond extremely well to small changes in temperature. A thermocouple consists of two junctions, each made up of different metal. One junction is kept at a reference temperature (cold), while the other is used to measure (hot). The difference in temperature between the two junctions creates a voltage difference, which can be used to measure the temperature.

Now let us take a look at a more advanced implementation of the non-contact temperature sensors that we are looking to add to our project. The thermal camera is a non-contact sensor that detect infrared radiations and convert them into visual images. Thermal cameras are very different from the typical cameras. Instead of capturing visible lights to form an image, thermal cameras capture the invisible infrared radiations to form an image. Everything emits some form of infrared radiation, that is why infrared cameras can still work in total darkness. Even ice radiates infrared. Although both infrared and visible light are parts of the electromagnetic spectrum, a normal visible light camera cannot capture infrared energy, and the opposite is true for thermal cameras. The resolutions, or the vertical and horizontal pixel counts, of thermal cameras are lower in comparison to visible light cameras. This is because of physics. The wavelengths of infrared radiation are much larger than visible light wavelengths. To accommodate for capturing larger wavelengths, the physical size of each thermal sensor embedded inside the thermal camera needs to be larger than the sensors inside the visible light camera. Coupling the design choice of making the thermal camera package the same size as that of a normal camera with the physically larger infrared sensors, the number of the sensors inside thermal cameras will be less than the number of sensors inside normal cameras. This results in the lower resolutions and fewer pixel counts for thermal cameras than normal cameras, if the package sizes are the same. Modern thermal cameras are extremely sensitive, most can detect temperature difference of as low as 0.01 degree Celsius. This means that as long as two objects next two each other have a subtly different heat signatures, they will show up clearly to the thermal cameras regardless of lighting conditions.

In United States, FLIR has an absolute monopoly on the thermal camera business on the market. Sure, there are other companies who make temperature sensors out there, but for thermal camera, there is pretty much only FLIR in the race. If you search up anything related to thermal cameras on Google, the first thing that pops up will likely be a FLIR product. Most of FLIR products aim toward military, security and research uses, so their prices are rather steep. However, they do have a consumer line of cameras under the name of Lepton for recreational use. We are looking into the specs of the Lepton 3 and Lepton 3.5 as potential thermal cameras for our project.

2.3.2.2 Distance Sensor

Distance sensors are, as the name implies, a type of sensor that detect the relative distance of an object with respect to a reference point. The reference point, in most cases, will be the current location of the sensor. Unlike temperature sensors where there are contact type and non-contact type, all distance sensors are contactless in nature. The device that measures distance based on contact is called a ruler and is not a part of the distance sensor family. It must be noted that, while distance sensor and proximity sensor do similar tasks, they are slightly different, and thus, not the same. The differences are minimal, but there are differences, nonetheless. Proximity sensor senses if an object is within the proximity, or sensing area, of the sensor. While it has an operating range, it does not necessarily detect the distance between itself and the object. Distance sensors do measure the distance between itself and the object using a plethora of methods. It can be said that the proximity sensor is a subclass of the distance sensor. The distance sensors can be divided into 3 sub-categories. These are ultrasonic sensors; infrared, or IR sensors; and laser, or LiDAR sensors. All of these sensors have very similar design scheme, in term of look. All three have 2 signal terminals, one for sending, the other for receiving signal. Figure 9 shows what the HC-SR04 ultrasonic sensor look like. And of course, the looks of other ultrasonic sensors, as well as the infrared sensors and the LiDAR sensors should be very similar.

Ultrasonic sensors are probably the most popular among the distance sensors. They can be found bundled with many different beginner products and like Arduino, or the MakeBlock MBot. If I am not mistaken, the entire lineup of Tesla's electric vehicles utilizes ultrasonic sensors to detect distance and enable autopilot. Ultrasonic sensors can detect the distance of an object by emitting high-frequency ultrasonic waves and receive the deflected waves back. The sequence of action would be something like this: First, the ultrasonic sensor will emit high-frequency sound waves towards the target object. The frequency should be around somewhere between 25kHz and 50kHz. When the soundwaves reach the target, the waves will bounce off and reflect back to the sensor. The receiver module of the sensor will then pick up the waves. The distance is then calculated using the time between sending and receiving the signals. Since ultrasonic waves are technically sound waves, they are not affected by the color of the transparency of the objects. This means that they will work well regardless of the lighting condition of the area. However, because the ultrasonic sensor relies heavily on having the signals bouncing off the surface of an object, if the surface area of the object is rough, with high degree of internal reflection, then the ultrasonic sensor will not work. And since ultrasonic sensors use high-frequency waves, their operating range is not too far. This is the same reason why low-frequency WiFi has much better range than high-frequency WiFi.

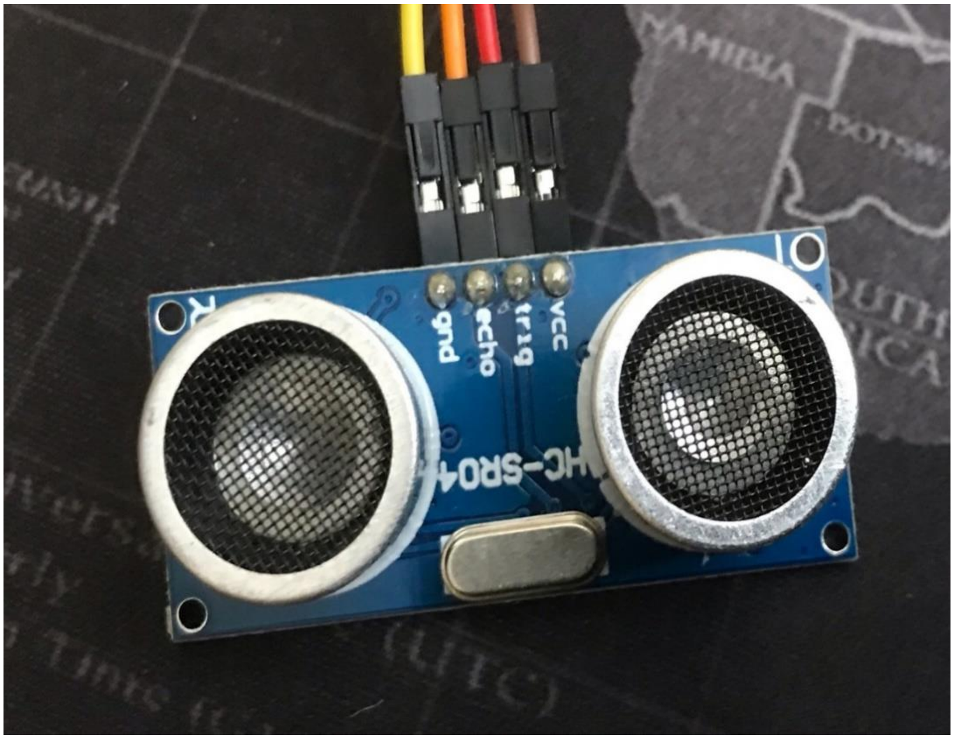


Figure 2-8 HC-SR04 Ultrasonic Sensor

As mentioned previously, the second type of distance sensors are the infrared sensors. Different from the ultrasonic sensor, an infrared sensor can not only detect the distance between itself and an object, but also pinpoint the specific location of the object through clever calculations. Instead of sending a signal that spreads out with distance, infrared sensors rely on infrared light, which is directional, and beam forming. Like the ultrasonic sensor, the infrared sensor also has 2 terminals, one for sending and one for receiving signals. The sending terminal consists of an emitter that emits a beam of light. The receiver is called the position-sensible photodetector (PSD). Because the PSD receiver can detect the angle of reflection of the light, it is then possible for the infrared sensor to compute the precise location of an object, and not just the distance. The order of operation of the infrared sensor is as follow: First, the emitter will emit an infrared light signal. When that that light beam reaches an object, it will be reflected. The PSD will then be able to capture the angle of incident and calculate the distance and the location of the object.

Since the energy of the signal is concentrated in a single beam without spreading out, the light beam can then travel further than the ultrasonic waves. This, in turn, makes the operational range of the infrared sensor much large than that of an ultrasonic sensor. Also, since light waves travel much faster than sound waves, the infrared sensor is able to update its data much earlier than can the ultrasonic sensor. This makes it so that the infrared sensor is more suitable for measure the distance of a constantly moving object. However, there are always some tradeoffs. All good things are almost always accompanied by bad things. Compare the ultrasonic sensor, the infrared sensor is more vulnerable to interference. Like the ultrasonic sensor, the infrared sensor cannot pass through walls, and does not work well with hard, rough objects with high degree of internal reflection. Not only that, the infrared sensor does not work in bright, warm areas, like under sunlight, direct or indirect. Sunlight, as well as hot objects, emit high level of infrared radiation, which interferes with the infrared sensor. When the PSD tries to capture the angle of incidence of the infrared light in an environment polluted with infrared radiations, the PSD will become confused and will not be able to accurately measure the distance of the object.

One of the most popular models of the infrared sensors on the market right now is the SHARP GP2Y0A21YK0F. Figure 10 shows the image of the SHARP infrared sensor. Notice how that its external design is very similar to that of the ultrasonic sensor It is small in size and is quite affordable at around $25 a piece. The sensor can be found bundled with some Arduino kits, or some other educational kits that teach kids to code. Compare to the ultrasonic sensors, the use of the infrared sensors is rather limited, as they can only be used indoor.

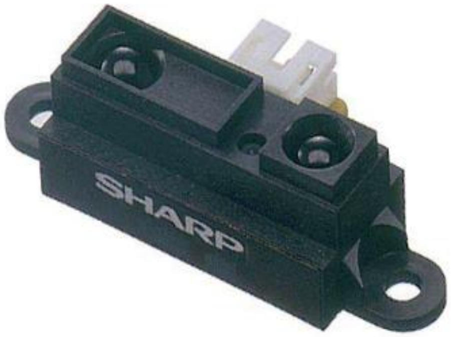


Figure 2-9 SHARP GP2Y0A21YK0F Infrared Sensor

Lastly, let us take a look at the LiDAR sensors. LiDAR is short for Light Detection and Ranging. This means that like with infrared sensors, LiDAR sensors rely on light waves to detect distance of an object. Unlike the infrared sensors, however, instead of using low power infrared light to detect an object, LiDAR uses high energy lasers to detect an object. LiDAR does not rely on beamforming; thus, it does not use the angle of incidence of the light to detect distance. Rather, similar to the ultrasonic sensor, LiDAR uses the time difference between transmitting and receiving of the signals to calculate the distance. An improvement that LiDAR sensors have over ultrasonic sensors are that LiDAR waves are more powerful than ultrasonic waves. This enables the light waves to more effectively bounce off rough objects that absorbs that ultrasonic waves. Being high-energy gives LiDAR an edge in not only detection the distance of an object, but also the 3D shape of the object. The order of operation for a LiDAR sensor is pretty much the same as for an ultrasonic sensor: First, the transmitter sends out a set of light waves. When the waves reach an object, they will bounce off and return to the sensor. The receiver will then receive the waves and use the time it takes for the waves to come back to calculate the object distance.

Because LiDAR’s light waves are powerful, they can be transmitted to a much further distance than both the infrared light and the ultrasonic waves. Under ideal condition, LiDAR can detect an object of up to 12 meters away. Since light waves can travel fast and far, the ideal usage for LiDAR is to measure the distance of fast-moving objects. For this reason, LiDAR sensors can be found in the majority of non-Tesla cars that offer some sort self-driving, driver-assistant features. Another advantage of using a LiDAR sensor over an infrared or an ultrasonic sensor is that LiDAR can detect through walls. LiDAR signals have lower frequency, thus longer wavelength than infrared and ultrasonic waves. This longer wavelength enables LiDAR sensor to scan through glass and thin walls. One of the primary trade-offs of LiDAR sensors are that high-energy LiDAR with wavelength of below 1400nm can cause burn to the cornea. LiDAR with wavelengths at 910nm or lower can cause permanent damage to the eye regardless of the energy output. That being said, as long as the wavelength of the LiDAR sensor is high enough, it should be safe to use.

Unlike infrared sensors and ultrasonic sensors, which are widely available in the DYI enthusiast market, LiDAR sensors are bit more difficult to find. The main reason is that LiDAR, compare to ultrasonic and infrared, is a still a relatively new technology. The newer the technology, the higher premium it demands. The price of a cheap LiDAR sensor could be as much as 100x to 1000x the price of an ultrasonic sensor, and 3x to 7x the price of an infrared sensor. Of course, there will always be cheaper off market options. However, those cheaper ones are generally non-regulated, and could pose a potential health risk to the user.

2.3.2.3 Camera

The most important sensor for this project, let us now talk about the image camera. The primary functionality of this robot is facial recognition. Without a good camera, the project will be a failure.

Obviously, on the consumer's market, photographic cameras can be found anywhere with any price. However, those cameras will likely not fit our needs for this project. For this project, we want a low-cost, high-resolution camera. In order to be able to accurately detect and analyze faces from other structures, clarity is the top priority of our camera. High-resolutions, high-clarity cameras on the market are usually sold for more than $1000. This is because those cameras are built together with a powerful image processing unit. Since we will be doing all the image processing on our side, we do not need a camera with auto-image processing. The cameras we are interested in are development cameras. These cameras are essentially just tiny lens modules that are attached to a printed circuit board to relay information. Since these do not have integrated processors, their prices are much cheaper than other alternatives with the same resolution on the market, almost 50-70% cheaper. Figure 11 below shows the image of a typical development camera module.

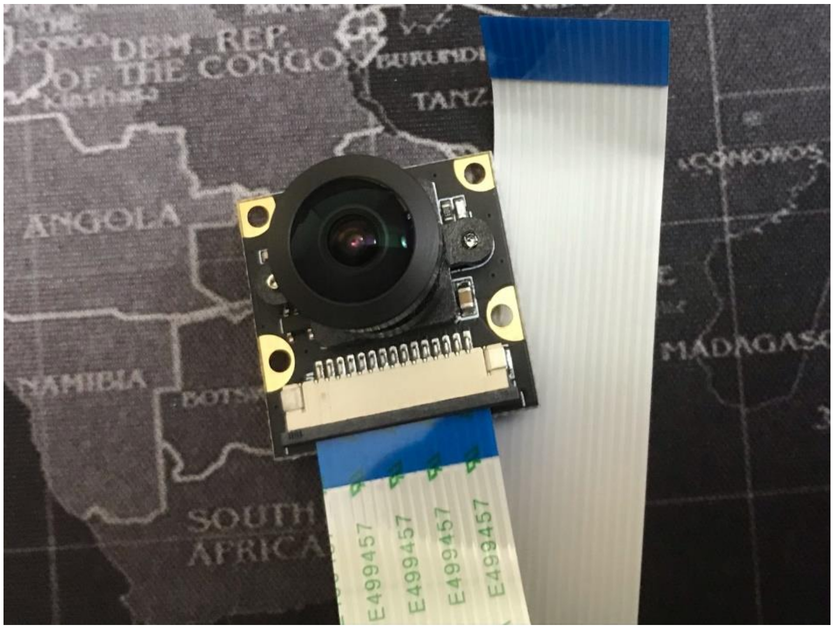


Figure 2-10 Generic Development Camera

2.3.3 Graphics Processors

According to Intel, a graphics processor is one of the most important electronic components in modern computers. A graphics processor can be found anywhere from cars, kiosks, to personal computers and data centers. Pretty much anything that can process and generate different images are considered a graphics processor. Graphics processors are generally designed for rapid parallel processing of similar tasks. These tasks include, but are not limited to playing a video, providing music, and cryptocurrency mining. Compare to a central processing unit, a graphics processor is faster and more power efficient. However, a graphics processor is very slow in continuously switching between dissimilar tasks. A graphics processor cannot, as of yet, replace a central processing unit. The other way around is also true. They work to complement each other in day-to-day computing.

Originally, graphics processors were designed only to accelerate the rendering of 3D graphics. As time goes on, they became more and more flexible and versatile. Currently, they also excel in high performance computing and deep learning. For this project, the functionality that we are interested in is the graphics processor's use in AI and machine learning. Image recognition involves breaking a captured image into hundreds of vectors, which requires a tremendous amount of parallel computing to process each image vector separately. Since image recognition requires parallel computing, it is the perfect fit for a graphics processor, which has high amount of computational capability, and is highly parallel in nature.

According to AMD, graphics processors are divided into 3 main groups: power-efficient, high-performance, and ultra-high performance. Power-efficient processors offer relatively high-resolution graphics output with a low power consumption at sub 15W. These are mainly found it small applications like kiosks, thin client computers, and human-machine interface systems. The high-performance processors offer high image fidelity at a compact form factor. Their power consumptions are around the 50W mark. In term of power efficiency, they offer the best performance per Watt of power. High-performance processors can generally be found in gaming machines and arcades. The ultra-high-performance processors are the next step up. They offer a linear increase in term of performance from the high-performance processors, but the price increase is exponential. Their target audiences are scientists, corporations, and government entities, where cost is not a problem. They can be found in high-end medical imaging devices and aerospace applications. For our purpose of small-scale image processing, the power-efficient models show be more than enough.

As mentioned previously, a graphics processor cannot efficiently work alone by itself. Rather, it requires the flexibility of a central processing unit to establish an operating system for it to work with. Because of this, we believe that the correct path to take is to go with a graphics developing environment, rather than doing something crazy like forcing a microcontroller to run an operating system for the graphics processor. Sections 2.3.5.1, 2.3.5.2, and 2.3.5.3 below show our potential pick for a graphical development environment. These are 3 of the most popular development environments on the market.

2.3.3.1 Raspberry Pi 4

Of the 3 platforms mentioned here, the Raspberry Pi is probably the most popular, with the strongest cult following. Raspberry Pi computers are the first to popularize the concept of a credit card sized computer. It was first launched in 2012 and became a hit almost instantaneously in the enthusiast’s field of computing. Its goal is to enable people of all ages to have easy access to cheap computers to explore the field of computing and learn basic programming languages like Python and Scratch. While not being as powerful as larger computers, the Raspberry Pi line up offers enough performance to do basic web browsing and play high-definition video. Since many have other competitors have entered the market for miniature computers, from smaller companies like SHARP and ASUS, to computer giants like INTEL and NVIDIA. Although Raspberry may no longer be the be all and end all option when it comes to small personal computers, it is still the go-to choice for many developers, who do not care too much about performance. The popularity of the Raspberry Pi could be attributed to its charitable nature. The Raspberry Foundation is registered in the UK as a charitable organization. Raspberry Foundation uses the majority of its profits to donate to schools and low-income communities to help spark the interest of computing in the younger generations.

The newest, and most powerful model for the Raspberry Pi lineup is the Raspberry Pi 4. It offers modern connectivity for fast and reliable communications with other devices. Table 3 below shows the list of features that the Raspberry Pi 4 has.

Table 2-3 Raspberry Pi Features

Component	Feature
CPU	Quad core Broadcom BCM2711
DRAM	2-8G LPDDR4 – 3200MHz
Wifi	IEEE 802.11ac wireless
Bluetooth	5.0 BLE
Ethernet	Gigabit, Power of Ethernet capability
USB	2x 2.0, 2x 3.0
GPIO header	40 pins
Display out	2x micro HDMI, 4-pole composite video
Camera support	2-lane MIPI CSI camera port

While the onboard processor for the Raspberry Pi 4 is powerful enough to do day-to-day coding and computation, machine learning task will be very slow on the chip. With only 4 CPU cores and no dedicated graphics solution, it may not be the best fit for this project, where intensive real-time parallel computing is required. Without parallelism, these programs require rapid context switching, which will overwhelm the weak CPU and cause the program to never terminate.

2.3.3.2 Jetson Nano

To follow the hype behind the production of the Raspberry Pi and to capitalize on the quickly expanding enthusiast embedded systems development market, in 2014, NVIDIA introduced the Jetson lineup of development environment boards to compete with the then popular Raspberry Pi A. Noticed the short-coming of the Raspberry Pi, NVIDIA utilized its specialty in the production of high-performance graphics solutions to produce a mini computer package that is marketed toward the then novel machine learning developers. Any of the Jetson product then were at least 5x more expensive than the original Raspberry Pi. Because of the difference in features, NVIDA created a split in the market. Developers who just wanted to have fun coding went with the Raspberry Pi, while those who wanted to tinker with AI and machine learning algorithm went with Jetson.

Recently, NVIDIA launched the Jetson Nano board to eat into the market share of the Raspberry Pi and other development boards. The Jetson Nano is advertised for being able to run multiple neural networks in parallel for applications like image classification and speech processing. The list of features of the Jetson Nano is presented in the table 4 below. From NVIDA's documentation, the Jetson Nano's CPU seems to run at a similar clock speed to the Raspberry Pi 4's CPU at 1.5 GHz. However, since the Broadcom BCM2711 is based on the ARM A72 architecture, which improved upon the ARM A57 architecture, the CPU of the Raspberry Pi 4 should be slightly faster than the one inside the Jetson Nano. The Raspberry Pi 4 also has an edge in system responsiveness in that its system memory runs at double the speed of the Jetson Nano.

Table 2-4 Jetson Nano Features

Component	Feature
CPU	Quad core ARM A57
GPU	128 CUDA core Maxwell
DRAM	4 GB LPDDR4 – 1600MHZ
Ethernet	Gigabit
USB	4x 3.0, 1x 2.0 micro-b
GPIO header	14 pins
Display out	1x HDMI, 1x Display Port
Camera support	2-lane MIPI CSI camera port

In exchange for slower RAM speed and much less GPIO headers compare to the Raspberry Pi, the Jetson Nano boasts a fast, dedicated 128 core GPU that can be used to process images. The lack of GPIO headers is also not a concern for the scope of this project since we require a relatively low pin count due to the small number of signals coming out of the graphics processor. The cost of the Jetson Nano board, although is much cheaper in comparison to the high-end Jetson TX2, it is still 3x more expensive than the Raspberry Pi 4. Going from the Raspberry Pi 4 to the NVIDIA Jetson Nano may seem like an expensive downgrade for many developers. However, for the few who work primarily with many high-resolution images and large datasets, the NVIDIA Jetson Nano offers an enormous speed up.

The major plus among the Jetson lineup versus the other micro-processor solutions on the market is the relative strength of its GPU features which are the core of every successful AI implementation. Differing from the CPU made up of primarily one specialized core to run sequential instructions, a GPU is full of individual, non-specific cores in order to perform many simple calculations in parallel. Since a neural network is essentially many addition and multiplication operations needed to be performed in parallel, a GPU is a much better solution for performing inference (the "math" behind mask detection) and training vs. a CPU. For this main reason, the Jetson lineup has a major leg up against the competitors.

Another major plus for the Jetson line is the massive amount of support NVIDIA has set up for its AI and deep learning frameworks, unlike the vast majority of other solutions on the market. All of the NVIDIA Transfer Learning Toolkit API is readily available online in an open-source format with very strong documentation behind it. Additionally, NVIDIA provides some more optimization for performing inference by the TensorRT SDK and runtime engine. These concepts are explained further in the Machine Learning Frameworks section.

2.3.3.3 ASUS Tinker Board S

Among the 3 development environments presented here today, the ASUS Tinker Board is the newest and is likely to be the least known one. ASUS joined the mini-computer hype train back in 2017. ASUS, rather than going for the top-end performance or the low-end price to compete with NVIDIA and Raspberry, decided to choose the middle ground between price and performance. Its price was and still is in between the price of a Raspberry Pi and the NVIDIA Jetson, and so is its performance.

In term of design implementation, instead of offering either high performance CPU or GPU, ASUS decided to offer super-fast storage for its boards. The list of features of the Tinker Board S is presented in table 5 below. Faster storage should help with the responsiveness of the operating systems. However, without a fast CPU to accompany the storage, the desired smoothness may not be realized. Another downside of going with the Tinker Board S rather than the Raspberry Pi 4 or the Jetson Nano is that the Tinker Board S's CPU is a 32-bit CPU. Being 32-bit means that the CPU does not support large address mapping. This means the future connectivity with other devices may be affected as not many devices will be able to connect to the CPU at once. However, the Tinker Board S does have a saving grace. That is its dedicated GPU option. Although the dedicated graphics processor itself is weak, with only 4 cores, it should be more than enough to do basic rendering and small computations. Then again, while it is capable of AI and machine learning workloads, image processing from a camera may not be the most suitable thing for the board. Image processing from a camera requires a smooth, constant inflow of data, which necessitates a large data bus, or multiple small ones. The single lane bus that is supported on the Tinker Board S may not be enough.

Table 2-5 Tinker Board S Features

Component	Feature
CPU	Quad core ARM A17 (32-bit)
GPU	Quad core Mali-T760 MP4
DRAM	2 GB LPDDR3 – 800MHz
Ethernet	Gigabit
Wifi	IEEE 802.11 b/g/n
Bluetooth	4.0 EDR
USB	4x 2.0
GPIO header	28 pins
Display out	1x HDMI
Camera support	1-lane MIPI CSI camera port
Storage	16 GB eMMC

2.3.4 Machine Learning Frameworks and Libraries

Digital image is defined as a sequence of array, or pixels, they are formed in columns and rows, and it is represented in two-dimensional f(x,y) function with a single 8-bit channel in greyscale with different range of pixel values from 0 to 255 (0 represents black color, while 255 represents white color) or 16- bit in “actual color” with values ranges from 0 to 65535. Each of the R, G, and B channel has 16 bits. This means the total number of bits per pixel is 48-bits. Each of the channel can be adjusted independent of each other to create many different colors. The number of colors that a 16-bits system can produce is in the trillions of colors. 2^48 = 281,474,967,710,656 different colors to be exact. It must be noted that there is also a 3x 8-bits channels variant of image displaying technology. However, it is not as common as a single 8-bits greyscale channel or 3 16-bits RGB channels. Image processing is one subset of computer Vision, that targets on improving the pictorial information, which an image such as video captured from a digital camera or images are defined as input, and the output is the extraction from the image to get related patterns or useful information interpretations from the images. Facial features consist of different regions inside the face including face structure, angles, face cut, and facial detection focuses on eliminate the unwanted background to identify the presence person’ faces. This term, computer Vision, has been used in a wide range of disciplines from medical field to entertainment, geographic information to industrial aspects, or personal purposes. Figure 12 below shows the general block diagram of a facial recognition operating sequence.

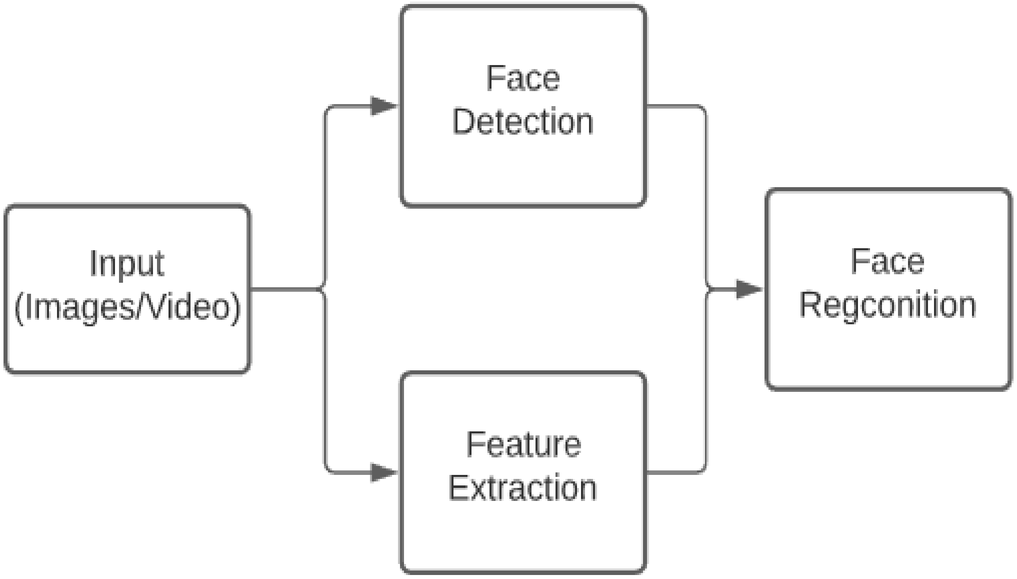


Figure 2-11 Block Diagram of Facial Detection

There are a lot of new and exciting opportunities in the field of image processing that researchers have been working on. There are seemingly and infinite amount of challenges for researchers to work on. They have to flexibly and quickly adapt with new changes in standards and technology and find new solutions in the near future to help better the world. With the rapid changes of technology every day in computer science and engineering, facial detection has become one of, if not the most interesting technology being used in various applications in today’s society. It has the capability of identifying and isolating human faces in digital images from the busy background that a picture may have. There are many new approaches and techniques that have been explored in order to capture an image with high accuracy. This requires powerful hardware, coupled with complex software algorithm to achieve facial recognition from different angles, regardless of the quality of the image. Although a lot of effort has gone into making improvements to the level of accuracy in detection and recognition, there are still some issues the arose due to the different lighting conditions, environmental noises, and the large disparity between image sizes. There are many frameworks and language libraries with different methods of increasing the overall effectiveness of a facial recognition system. Some of the more common, reliable, and robust frameworks and libraries in the following subsections. While this is not an exhaustive list of frameworks and libraries, it should be able to give readers a general idea of the current state of the facial recognition technology.

2.3.4.1 OpenCV

OpenCV is short for Open-Source Computer Vision Library. Its primary purpose is to help accelerate computer algorithms to achieve real-time computer vision. The library boasts more than 2500 algorithms, from simple to complex. It includes both old and novel algorithms to help developers started in the computer vision endeavor. OpenCV's algorithms are best known for their ability to accurately distinguish objects, as well as the ability to track multiple in-frame objects at once. With a good enough training data set, OpenCV's algorithms should be able to detect human faces quickly and accurately without any problems.

The OpenCV project was originally started in 1999 as a part of Intel's research project headed by Gary Bradsky. Its purpose was to expanding computer vision field to the general public by producing commercial applications with the help of rigorous research. OpenCV's alpha version was first released in 2000 to attract more developers and researchers onboard with its great benefits including, but are not limited to, advanced capabilities in face detection, face recognition, and others Machine Learning methods. OpenCV was in beta from 2001 through 2005 as it attempted to squash many major bugs that could freeze systems. Its first stable release was in 2006. With the continuous development, Open CV is now able to support multiple algorithms regarding to Computer Vision and Machine Learning. It also supports a wide variety of programming languages like C++, Python, MATLAB, and Java. However, the primary interface of OpenCV is written in C++.

The thing that is most attractive about OpenCV is its support for a wide variety Operating Systems such as Windows, Linux, Android, and MacOS X, and its cross-platform compatibility. OpenCV is widely used in large corporations like Honda and Toyota to train their self-driving models. Since OpenCV algorithms are primarily small in size and are fast and responsive, they are suitable for cars, where embedded processors may not be too powerful. OpenCV is also used by Google, Yahoo, and Microsoft to better optimize their search engines and data tracking algorithms. Large corporations gravitate toward open-source projects to maximize their profit margins by minimizing the development time. OpenCV creates an excellent framework to Machine Learning models upon.

A computer program that is able to determine positive picture or negative picture is known as a classifier. A classifier can sort data into labeled classes from the massive number of input face and non-face pictures. This allows end users to manipulate data and manually classify them the way the users wanted. OpenCV provides users with two well-known face detection algorithms. These are Haar-Features Classifier and LBP Classifier.

A Haar-features is like a kernel in the convolution neural network. In case you do not know about convolution neural network, here is a bit of detail. Convolution neural network is just another class for analyzing visual imagery. The primary key for Haar classifier object recognition is Haar-like features, these features consist of adjacent rectangular group of pixels in a specific location in order to determine the relative light and dark areas based on the contrast variances between pixel groups. Haar-like features can be increased or decreased in the size of pixel depends on the area being examined. For example, in human face, there are some regions have darker color, Haar features for face detections will use the two adjacent rectangles to focus on eyes, which is always darker than the cheek and nose regions. Through some complicated math, the pixels are then assigned to positions within a matrix. Each of the matrix entry is called a kernel. To manipulate or to study each pixel, you first need to manipulate the corresponding matrix index. Haar-features classifier works similarly, but instead of processing an image kernel-by-kernel, Haar-features classifier work on a larger scale, with each group being a different feature of the image. A downside of Haar-feature classifier is that each feature needs to be determined manually, rather than through an algorithm like with Convolution neural network. Because of this, Haar-features classifier may not be able to detect a face with slightly different features like a face with a missing eye, or deformed features, for example, even though it was trained specifically to detect faces.

Compared to the Haar-features classifier, an LBP cascade classifier may be easier to understand and implement. It should be faster to train as well. Cascade classifiers are just the concatenation of many different simple classifier schemes together. In a cascade classifier, the data traverse from one scheme to the next. Think of it as a pipeline, similar to the one used in computer architecture. Each pipeline stage weeds out more and more of the negatives. Unlike the computer architecture pipeline where each stage has approximately the same execution time, a LBP stage gets more complex and takes longer to compute the further down the pipeline it gets. The LBP cascade classifier is just a variant of the cascade classifier, where you detect objects based on the Local Binary Pattern, hence, LBP. In theory, each object will look very different from each other, which mean each object will have its own unique Local Binary Pattern. However, in practice, many things look similar to each other, thus having similar, or sometimes even exactly the same binary pattern. This could be seen in nature, where a hornet looks extremely similar to a bee. One of the ways the human eyes can tell them apart is through the size of the animal: the larger one is likely to be the hornet and the smaller one is likely to be the bee. This form of characterization does not work with an image. In an image, we can only tell the general characteristics of the object in the front and not its size. Similarly, the LBP algorithm can only assign binary pattern to the object based on the 2D data that it has, rather than based on the 3D data. Because many things look similar in 2D plane, those objects would be assigned with similar LBP values.

There are many other classifiers that OpenCV supports, but we will just discuss about these 2 classifiers to keep this section short. The Haar-features classifier and the LBP cascade classifier's brief comparisons are presented in the table 6 below. These two are two of the more popular classifiers supported by OpenCV.

Table 2-6: Haar Classifier vs LBP Classifier

Algorithm	Advantages	Disadvantages
Haar Cascades	- Highly accurate detection - Low false positive rate	- Longer training time - Computationally complex and slow - Less robust to occlusion
LBP Cascades	- Short training time - Computationally simple and fast - Robust to occlusion	- Less accurate - High false positive rate

2.3.4.2 Amazon Rekognition

The performances of deep learning regarding to face detection, face recognition, face expression, age and gender are generally very poor on consumer hardware. They require complex context switching alongside many parallel computations. Thus, the more reasonable outcome to this problem is to offload the workload to powerful datacenter machines. Amazon has been successfully capturing the market needs by providing a working deep learning framework that is available on the cloud. Amazon Rekognition is Amazon's solution to a cloud-based Software-as-a-service (SAS) computer vision framework. It was first launched in 2016 with the initial ability of identifying objects and providing highly accurate facial analysis. This service requires no prior experience in machine learning to use since the customers just need to add images or video to the Amazon Rekognition API, it can quickly analyze the images and store the data in an Amazon-owned database located in Amazon S3. People say that Amazon's main business is e-commerce, but I beg to differ. Amazon's primary source of income is in the database hosting and API providing business. Amazon's cloud services are the largest cloud infrastructure on the Internet as of 2020.

There are lots of features that Amazon Rekognition provides. These include celebrity recognition, text detection, face- based user verification and so on. The service does not require any deep learning or machine learning experience from the user, therefore, anyone from different background who have passions in computer vision can take advantage of the service to fulfill their dreams. They can analyze images and video footages easily with minimal amount of coding required. This service allows the customer to analyze millions of data quickly and accurately using the Amazon's complex deep learning technology and powerful server hardware. As Amazon Rekognition was designed in-house by Amazon, it is able to work seamlessly with Amazon S3 and AWS Lambda. As each AWS service is well-put-together to ensure the smooth transition of the process, users can be certain that their data is secured with Amazon's amazing infrastructure. An example of Amazon's seamless integration of front-end process and back-end database would be: photos of visitors to the residence is stored in Amazon S3; every time the person arrives, a Lambda automatic detection function would be triggered, letting Amazon Rekognition API identify the guest. Amazon suite of cloud services is free to try out. Users are able to analyze up to 5,000 images and 1,000 minutes of video per month, for the first 12 months with the Amazon Rekognition tiered pricing model. Amazon's cloud services integration model can be seen in figure 13 below

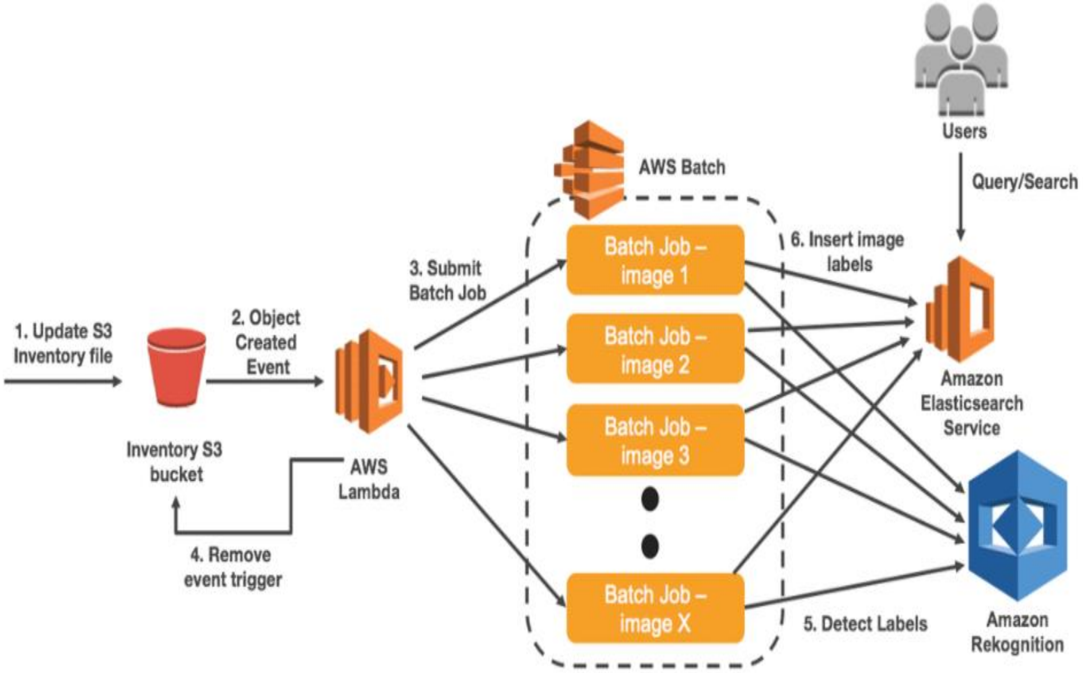


Figure 2-12 Amazon Rekognition Structure

However, there is a study conducted by Massachusetts Institute of Technology researchers, which have shown that Amazon Rekognition has gender and ethnics bias in particular scenarios. In details, the system misclassified darker skinned women as men 31 percent of the time. This is bad news to Amazon in term of future customer acquisition, however, algorithms are constantly changing, so the bug should not persist for too long. Besides, it is the user/programmer's responsibility to improve upon Amazon's algorithms, as what Amazon provides are only the backbones to make the work faster and easier. With so much efforts have been made in order to improve the face analysis, Amazon Rekognition is still a good technology that provides low-cost deep learning and computing services that have the potential to do good in modern day society. In addition, Amazon Rekognition requires zero Machine Learning experience, and it can be used in real-time analytics with complex data set. It is easy to learn and cheap to use.

2.3.4.3 Google's FaceNet

FaceNet is a unique framework that mainly utilizes deep convolutional neural networks to perform specific tasks in the domain of deep learning and machine learning. Specifically, Google's FaceNet, as the name implies, excels in facial detection, verification, and clustering. The concept of FaceNet as a facial recognition framework was first introduced by Google researchers in 2015. Older prototypes of FaceNet can be found on Github as a public domain project. However, newer revisions are only available for internal use at Google. FaceNet provides a unified embedding environment for recognition tasks, which uses Euclidean space for finding the similarity within the facial images. This algorithm of using the Euclidean space is called “one-shot learning method” by Google developers. In other words, this method can classify the images of a person through obtaining shapes, appearances, as well as special identifications to maximize the accuracy of the detection. This enables the algorithm to accurately differentiate between similar-looking people. The primary difference of FaceNet with other facial recognition techniques is that it generates the embedding data from mapping images while not using any bottleneck layers to create recognition. For example, once the embedding data is generated from similarities between faces, it can be used as feature vector.

FaceNet uses a batch of layers as input into the traditional deep convolutional neural network (CNN), followed by the L2 normalization. Essentially, FaceNet utilizes kernels to establish the embedding data that corresponds to the similarities between faces and uses triplets to classify the model structure. FaceNet then output retrieved vectors directly into the 128-dimentioanl vector space. Figure 14 [36] below shows the sequence of action for the FaceNet’s vector separation algorithm.

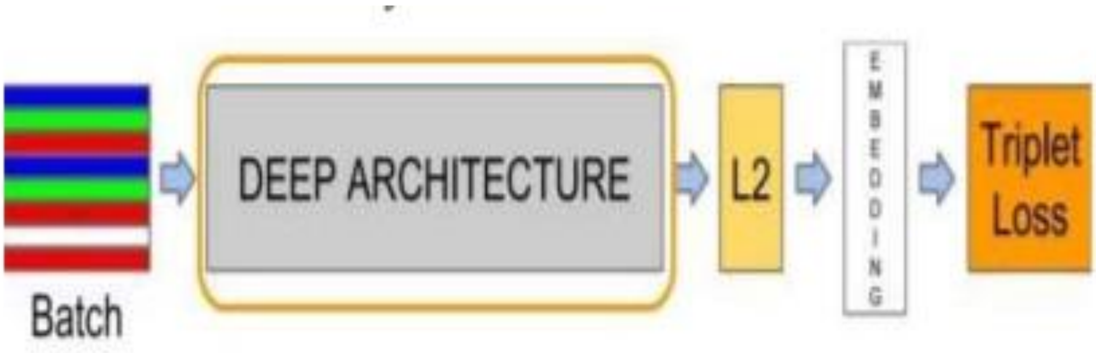


Figure 2-13 FaceNet Structure

In the training process, FaceNet use triplet loss for matching faces. This is done by the collection of anchors, positive and negative images. The vector sequences are compared and the result is noted and classified as a class depending on the person or object such that whenever such sequence appears again, FaceNet will that know that the sequence belongs so which class, enabling fast recognition time. Figure 15 [36] shows the FaceNet’s learning sequence by using triplet loss.

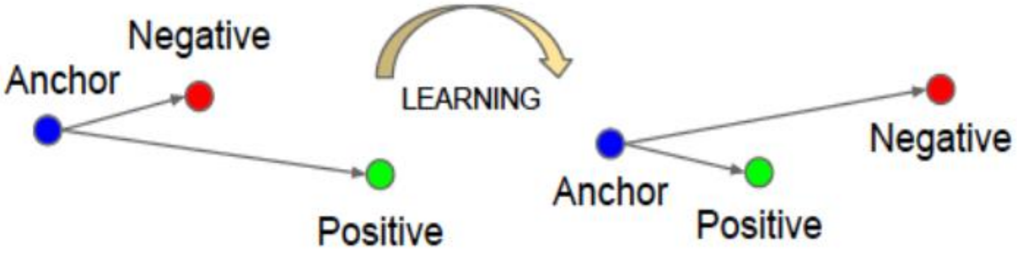


Figure 2-14 Triplet Loss

The classification of each of the triplet loss is defined as follow: an anchor image is defined as image of a specific person; a positive image is the images of a person that may or may not be the same person as the image holds by the anchor; while negative image is an image of any other unrelated objects like a dog, a monkey, or a rock, for example. The requirements for these three elements can be summarized with the triplet loss function. The goal of the function is to minimize the distance of anchor image and positive image, while maximizing between the anchor image with the negative image. In the other words, the closer of anchor with positive images, compared with the negative images, the better, since they have major similarities, while negative images have distinct identity from the anchor. A triplet loss function can be defined in figure 16 [36] below as:

$$\sum_{i}^{N} \left[\| f(x_i^a) - f(x_i^p) \|_2^2 - \| f(x_i^a) - f(x_i^n) \|_2^2 + \alpha \right]_+$$

Figure 2-15 Triplet function

In this formular, *a*,*p* and *n* represent the anchor, positive and negative images, respectively. The function embeds an image into the d-dimensional Euclidean space. Then, the FaceNet algorithm will run to find all possible sets of triplets from the image. These triplets will converge slowly since the model is not able to learn much from many of the sets as majority of them contain irrelevant data. Therefore, it is important to select triplets that violate the triplet constraint in order to ensure fast convergence and make quick improvement of the model. Essentially, we want to select hard positive and hard negative using argmax and argmin. There are major some issues with this approach such as poor training results, and the infeasibility of computing argmax and argmin for the entire training set. However, it is possible to avoid these issues by creating triplets online using mini batch or computing hard positive and hard negative in every n iteration.

This FaceNet system trains the convolution neural network using Stochastic Gradient Descent with backprop and AdaGrad. The training model of FaceNet may be different depending on the applications. Additionally, FaceNet uses two different types of convolution neural networks. These CNNs are Zeiler&Fergus architecture and GoogleLeNet style Inception models. While Zeiler&Fergus architecture has 140 million parameters, which requires 1.6 billion FLOPS of computational power per image, Inception model has significant fewer parameters and FLOPS requirement, such that it can run smoothly on devices with weak processors like mobile phones.

2.3.4.4 NVIDIA Transfer Learning Toolkit

One of the major draws to using the NVIDIA Jetson line of graphics processors as explained in the previous section 2.3.3.2 Jetson Nano is the fact that the NVIDIA Transfer Learning Toolkit (TLT) is available for usage. This is not a feature that is available if we choose another device for the graphics processing portion of the design.

Unlike the previous frameworks, TLT provides previously trained models for re-training, pruning, and in general faster deployment than if the user were to come up with their own model and use one of the frameworks mentioned previously. This is one of many major upsides to using TLT.

TLT provides various open source, previously trained models based on different inference scenarios such as object detection, facial detection in IR, vehicle detection, etc. By taking an existing model that has been optimized by NVIDIA over large-scale databases of images numbering orders of magnitude above what any individual consumer could perform training with on their own, the user is already starting off with a precise and efficient network. Although at this point the model is not trained for the application of mask detection. The solution is simple: simply re-train the network on our own. This will undoubtedly result in a network more efficient and precise than anything we can come up with from scratch.

Another major plus for TLT is that we will not need to spend time putting together an environment since it runs inside of a Docker container separated from the rest of whatever Linux OS is running on top of it. Similar to an OS, all dependencies and software required for training the model are provided inside of the Docker container. The only thing used in common with the OS above is the Kernel.

2.3.5 Cloud Service Providers

Today, the cloud datacenter infrastructures are becoming more mature than ever before. The percentage of firms using cloud services has increased from 48% to 84% within just five short years. Not only that, the current global market size of $371.4 billion in 2020 is expected to grow up to $832 billion by 2025. Due to COVID-19 pandemic, many governments around the world have enforced social distancing laws by forcing physical businesses to shut down, making many people move to remote work and study. This is the primary reason for the sudden surge in cloud service users and subscriptions. As people increasingly work from home, they require a reliable way to access their companies' databases without the need of going to their companies' physical locations. They also need to be able to collaborate with their colleagues in real-time on some projects. This creates an incentive for workers and companies to subscribe to cloud services to virtualize their work quickly and efficiently.

Cloud computing is defined as the practice of using a remote server networks, which are hosted on the internet, to complete the required works without having to be in physical contact with other people. In cloud computing, huge arrays of supercomputer servers are used to store, manage, and process data. Cloud servers can be access at any time, from anywhere, without the need to be on the same local network. They also allow multiple users to access different instances of the same file at the same time without having file-data conflicts. Cloud servers are preferred over local servers because they are versatile and can be deployed at a moment notice. This means that the expandability of a cloud server is theoretically limitless. They are also cheaper to maintain as compared to a physical server, and they do not take up any valuable physical real estate of the company.

Cloud computing is much less capital intensive when compare to local database hosting. However, there is one downside to cloud servers that a local network server does not have. The primary concerns for a cloud server are security and service reliability. Basically, trust is what cloud service providers are trying to attain. Companies of all industries and sizes utilize the cloud to run their operations. If the server is down for even one minute, the potential loss for some businesses could be thousands or even millions of dollars. This is why reliability of a cloud infrastructure is important. As for security, obviously, hosting the server on the Internet is in no way as secure as hosting the server locally. An adversary could easily have access to the server if they just subscribe to the same cloud infrastructure. Whereas for local servers, the adversary would need to go to the company's location and have physical access to the hardware. That being said, cloud service providers, big and small, are trying their best to mitigate the security problems.

Currently, there are only a dozen or so companies which are providing Cloud computing services. Each cloud service provider is unique, and their cost of operation is different as well. They provide different services depending on the need of the individual or corporation, and there is a large price range across different services. Some are aimed toward individuals and small companies like Linode. Some others are aiming for large corporation like IBM. And of course, there are those which aim for both demographics as well, like Amazon AWS, for example. Some of the more popular cloud service providers are Amazon Web Services (AWS), Kamatera Cloud, Linode, Cloudways, Google Cloud Platform, IBM, Dell Technologies, Hewlett-Packard Enterprise, etc. With so many different options available, picking the right cloud provider is not easy, especially for novice developers. There are many factors to take into consideration when choosing a cloud service provider. These include, but are not limited to the amount cloud-based services offered, ease of use, price, hybrid or multi-cloud capabilities, etc. Although there are many providers on the market, there are very few which could stand up to the might of Amazon Web Services. AWS is the largest cloud service provider as of quarter 4 of 2020. The analysis on AWS and other popular cloud service providers are provided in the subsections below.

2.3.5.1 Amazon Web Services

Amazon Web Services (AWS) is known as a cloud computing platform that belongs to Amazon. AWS is as known as an infrastructure-as-a-service (IaaS) model of cloud computing. Originally, Amazon built these technologies to solve their own problems of hosting one of the internet's busiest websites. They need their website to be able to work at the highest efficiency and fastest performance at all the time, regardless of website traffic. Being the highest grossing e-commerce business on Earth, a minute of down time for Amazon could translate to thousands of dollars of revenue loss, so Amazon had to make its server has invulnerable as possible. Over time, Amazon’s servers became so large that their e-commerce business can no longer saturate the machines, wasting computational power and electricity sitting idle. They observed that that most of the other companies are trying to adapt to the digital age by hosting their own servers. The server designs fulfill very similar purposes to Amazon’s own server, which gave Jeff Bezos a revolutionary idea: Since we have a lot of computational power that is wasted sitting idle, why don’t we sell it? As a result, in 2006, Amazon started selling subscriptions to use their servers on the open market with some of the basic services such as S3 for storage, compute, database for enterprises at a discounted price for the first few years. As enterprises grow more and more reliant on Amazon’s infrastructure, they continue to purchase Amazon’s services even after the trial period. AWS has their physical database locations in many different regions all over the world. This broad availability zones means that users can put their computing algorithms and data in the AWS cloud and access them from almost anywhere. Today, Amazon Web Services had grown bigger than ever before. Amazon now offers 175 fully featured services from data centers globally. There are services for lots of different thing especially AWS wavelength; as long as you can think of something, Amazon is likely to have a service that pertains to that particular need. AWS wavelength is built to accommodate for the up-and-coming 5G expansion. It offers optimized broad connectivity at high speed for mobiles edge computing applications. The infrastructure of AWS wavelength is essentially an array of large cell towers to connect from one city to another. Within a large cell tower’s vicinity, there are many small towers to help relay and strengthen transmit signals. This helps ensure that the users can always have access to the fastest network anywhere. A lot of businesses now are trying to transfer to cloud computing instead of investing money in building their own IT infrastructure because by leveraging the power of cloud computing, they are able to access thousands of servers virtually within minutes at an extremely low cost. For cloud computing, AWS is second to none in term of speed and reliability and is suitable for large businesses with high up time. With that being said, AWS is a network backbone for a lot of corporations of all industries and sizes such as Netflix, Twitch, LinkedIn, Facebook etc.

One of the big concerns that have sparked the interest of many researchers in recent decades is that whether the cloud and its services are secure. Since the dawn of cloud computing, there have been hundreds of reported cross-VM attacks targeted at the software and hardware vulnerability of a cloud-based environment. There may be even more attacks that were not publicly mentioned or have gone undetected. Some of the recent notable attacks are RAMBleed and Last-Level Cache Side-Channel attacks. The principle behind RAMBleed is the Rowhammer attack. The gist of Rowhammer is that DRAM cells lose charge over time, and by continuously accessing the memory rows next to the targeted row, attackers can partially charge the targeted row, inducing a bit flip. When accessing a row in the DRAM, all the cells in the row are recharged. Because the DRAM cells are so small, are so located so close to each other, when a row is charged, there will be some current that is leaked to the adjacent rows, charging them. When the charge in the adjacent rows reach a certain level, there is a small chance for one, or several bits to change its value. The bit flip has several implications. First, if the bit flip happens in kernel memory, the attacker could potentially gain illegal access to the system, i.e. a privilege escalation attack. Second, the bit flip in user memory could potentially corrupt sensitive data. Third, in an SGX-protected cloud environment, a bit flip causes the machine to reboot. This is a mechanism from Intel's SGX to protect user's data. However, due to this very behavior of SGX-protected system, the attacker can continuously cause bit flip by rowhammering, causing the cloud server to enter a boot loop. This is essentially the basis of a Denial of Service Attack (DOS) or if the attacker attacks multiple servers at the same time, it would create a (DDOS) attack. Intel SGX is a software implementation of a secure lock such that only authorized users are allowed to use certain parts of the memory. It pretty much creates an isolated environment for a process to run. While that process is running, no other processes can have access to the memory addresses that are reserved. And once that process is terminated, the data inside the memory is flushed in such a way that it is impossible for anyone to retrieve. A popular defense against the Rowhammer attacks and its variants is to use DRAM with Error-correction codes (ECC DRAM). By ECC's design scheme, a memory block with 1-bit error can be corrected, and a memory block with 2-bits error can be detected. However, with roughly 8 thousand bits per row, the probability of having more than 2-bits flip is high. Obviously, RAMBleed has since already been mitigated. Note that I used the term mitigated and not eliminated. As long as the fundamental design of the hardware does not change, future attacks that exploit similar hardware vulnerabilities are possible. You can read more about Inter SGX link from the link posted in the Appendix section below.

The Last-Level Cache (LLC) attack is easier to explain. Essentially, since the CPU share its LLC with all of its core, an attacker can snoop sensitive data from the LLC. Let a victim process runs on core 1 with VM1, and the attacker process runs on core 2 with VM2. As the victim process does its job, it will continuously fetch and write data to and from the memory by first accessing the cache. Since the attacker process is running on the same machine, it also has access to the same LLC. The attacker process can just read in data from the LLC to know what the victim is doing. This is only possible because of the cost-cutting strategy of cloud service providers. Since server computers have processers with many cores, it is very likely that 1 customer will not be able to use up all the cores. Thus, cloud service providers tend to put 2 or more customers on the same physical machine using virtual machines, or VMs, to make them think that they have their own machine.

In fact, AWS is quite serious about security and privacy for their customers as well as for themselves. The attacks mentioned previously did not happen on AWS systems, but rather, on some of Amazon's competitors. AWS has built an amazingly secure technology platform which has a lot of layers of physical and operational security. This includes hiring armed guards to protect their database locations and continuously upgrade the software to mitigate bugs. They also regularly audit and maintain their systems to make sure that the services are bug-free and safe for the customers. By using AWS, users are able to build their work on the most secure global infrastructure. As an addon bonus, users also have the ability to encrypt their data, move them to different providers without the fear of migration data loss, and manage their data from anywhere. For large and fast-growing corporations sharing data source between many different employees is the most cumbersome challenge they have to face. Moreover, the data in question could be so diverse that it would require the companies to build an expensive and complex infrastructure. By using a cloud database, they do not have to worry about such things. There are many types of database and AWS services that Amazon offers, as described in Table 7 below. Basically speaking, if there is a need, Amazon has a solution. For example, if you need a number of small and simple databases, then Relational Database Service (RDS) is usually the best way to go because it eliminates the need for unit testing, diagnosing performance and concurrency issues. On the other hand, if the business grows up and needs more performance, control, and functionality, EC2 should be better because it is able to run the process local to the database, and it has more replication options, extensions and is overall, cheaper than other available options.

Table 2-7 AWS Database Services

Database Type	Use Cases	AWS service
Relational	Traditional applications, ERP, CRM, e-commerce	Aurora, RDS, Redshift
Key-value	High-traffic web apps, e-commerce systems, gaming applications	DynamoDB
In-Memory	Caching, session management, gaming leaderboards, geospatial applications	ElastiCache for Memcahed, Elasticache for Redis
Document	Content management, catalogs, user profiles	DocumentDB
Wide column	High scale industrial apps for equipment maintenance, fleet management, and route optimization	Keyspaces
Graph	Fraud detection, social networking, recommendation engines	Neptune
Time Series	IoT applications, DevOps, industrial telemetry	Timestream
Ledger	Systems of record, supply chain, registrations, banking transactions	QLDB

2.3.5.2 Oracle Database

Like many other cloud service providers, Oracle cloud is also a cloud computing platform that is provided by Oracle Corporation. Oracle Corporation may not seem to be familiar to younger generations of developers out there, but most engineers and computer scientists are likely to have come into contact with Oracle in one form or another. Oracle Database is built on top of SQL, known as multi-model database of Oracle Corporation, which provide clients with relation database management system. It is considered one of the most cost-effective solution to manage information and applications for enterprise grad computing and data warehousing. Enterprise grid computing generates large pools of industrial standard, which means each new system is provisioned from the pool. Moreover, with this architecture, there is no peak workloads since the capacity can be added or relocated from the resource pools. With this new IT architecture that provide low-cost enterprise information system, this architecture also helps group of independent, software components and modular hardware connect and rejoin as desired. With this particular grid computing architecture, it benefits IT system including better quality of service as well as lower cost and its flexibility. In addition, Oracle is the corporation that developed and standardized the Java programming language. Java is popular for its integrated memory management system, its numerous numbers of built-in classes, and its cross-platform compatibility via the Java Virtual Machine (JVM). Oracle is also the developer behind the easy-to-use NetBeans IDE. Even if you have never used Java before, you have likely been recommended the NetBeans IDE for its simplicity and ease of use with popular languages like C, C++, and PHP, just to name a few. Oracle Cloud is a set of services that are developed and maintained by Oracle and the respective the third party. Oracle Corporation is known as a huge technology company that provides servers, storage, networks, and application services. By allowing users to access Oracle Cloud infrastructure-as-a-service without introducing the need to apply distributed architecture patterns to the software, Oracle was able to simplify the process of setting up and managing accounts and make it so that users can easily set up their desired databases. Oracle calls this the self-service hosting.

Oracle, like Amazon, has a vertical integration of their cloud services. They do not outsource, but rather, build their own infrastructure, platform, software, data, etc. As you know, or will know, databases come in many different types, and each type has many different implementations. Oracle owns one of the most popular, free, and open-source implementations of the relational database type. It is called MySQL. A detailed comparison of different database types will be provided in section 2.3.8. In Oracle Database, they offer many service options to meet a wide variety of use cases. When we choose to deploy our database in the cloud, we all have different needs. To accommodate for the large user base, Oracle has created a plethora of services, each of the services has different features as well as functionalities. These services are described in the Table 8 below. Moreover, the Oracle database is very helpful to use if the corporation is expected to drastically change in size in either direction. With just a click of the button, users will be able to add or remove CPU cores, memory size, or storage size. Since Oracle’s infrastructure is relatively young compared to Amazon’s, it is true that the expandability of an Oracle database is limited. However, Oracle is quickly expanding its infrastructure to accommodate for larger databases. From Oracle’s advertisement page, Oracle Cloud can process large, concurrent data streams much better than any competitors in the market. And this is the reason why Oracle cloud is popular with corporations where constant, multiple concurrent data streams are necessary like banks and healthcare services. While Oracle database is ideal for individuals and small companies thanks to its low cost, ease of use and high availability, it is also a valuable asset to large corporations, which place high importance to fast transfer speed for high volume data. A few of the more notable users of Oracle Cloud are Citi, PNC, Bank of America, UnitedHealth Group, etc. More information about Oracle Cloud database solutions can be found in table 8 below.

Table 2-8 Oracle Database Services

	Database on Oracle Cloud Infrastructure	**Virtual Machine or Bare Metal**	**Exadata Cloud Service**	**Autonomous Database**
Manage ment	Customer	Customer	Customer	Customer
Max DB size	40TB	9TB-40TB	Petabytes	128TB
CPU core range	1-24	2-52	1-100's	1-128
Storage	Block:Eslastic per GB	Block or Local NVMe disks	Local disks, NVMe flash cards: Fixed	Exdata, Flash cards: Variable
Backups	Manual	Customer initiated	Customer initiated	Automated
Data Guard	Manual	Across ADs, Across Regions via VCN peering	Across ADs, Across Regions via VCN peering	Across ADs
RAC	Not Available	Available in VM	Available	Available

Compared to Amazon Web Services, Oracle boasts to have better data transfer performance at a better price. And the price difference is not that small either. According to Oracle's website, the price difference between an Oracle Cloud subscription to an equivalent AWS subscription is as much as 80% on average. For a 1620 Terabytes worth of data transfer, Oracle Cloud infrastructure FastConnect charges $918 per month. This is compared to the $34,020 per month for AWS DirectConnect. What more, if the users do not use that much bandwidth at all, the cost for Oracle Cloud could be completely free. For 40GB worth of database storage and 10 or less Terabyte of data transfers per month, Oracle users incur $0 cost. This is compared to AWS $92.20 per Terabyte per month. Oracle also advertises their solutions as 60x faster than those from Amazon. While it is true that Oracle's infrastructure as a whole is newer, and thus, faster than that of Amazon, 60x may be a bit too optimistic. The 60x number takes into account the large user base of Amazon. Obviously speaking, the more users a network has, the more congested it will be. Since Amazon has more users than Oracle, it is no wonder that Amazon network is more congested, and thus, much slower.

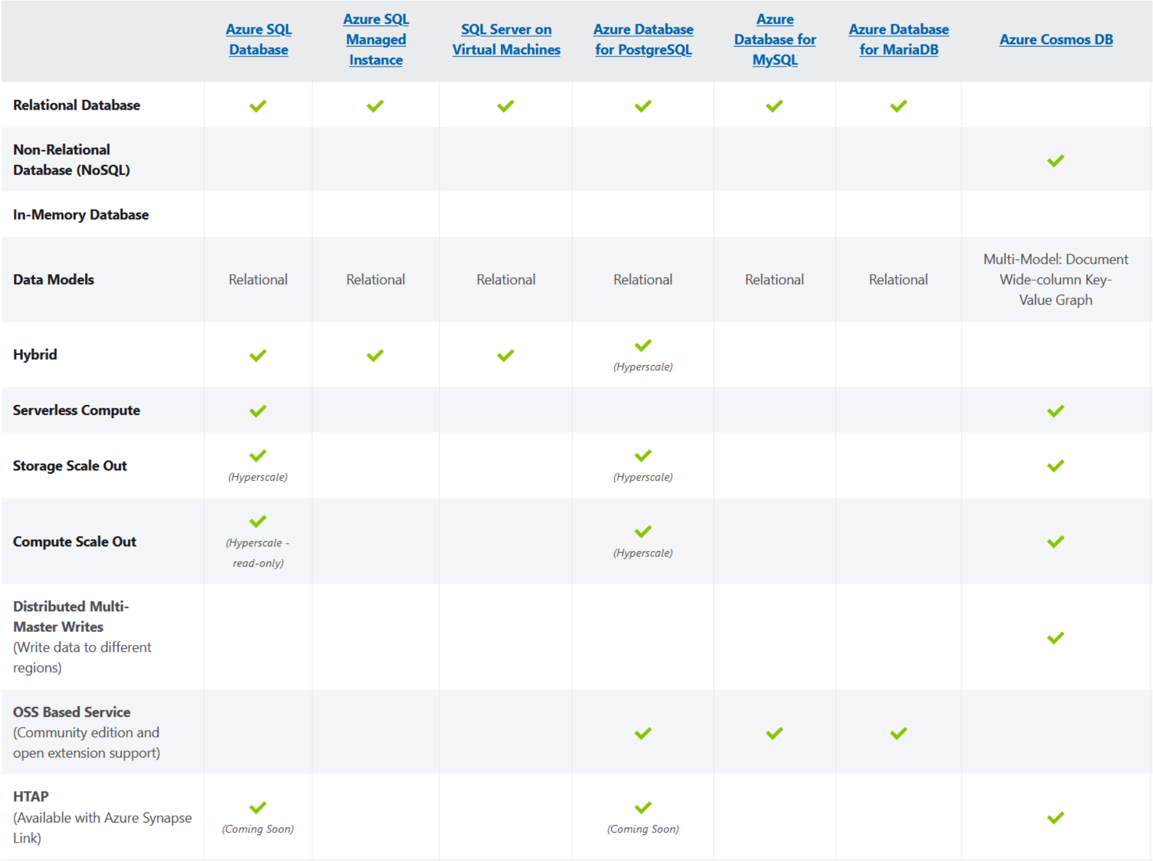
2.3.5.3 Microsoft Azure

Microsoft Azure is yet another popular cloud computing service. Being a cloud service from Microsoft, one of the largest companies in the world, Azure does not lack behind neither Amazon nor Oracle in terms of features and reliability. Azure cloud offers compute, analytics, storage, networking, and 200 other services. Unlike Oracle Cloud, which targets individuals and small businesses, and AWS, which targets large and small businesses alike, Azure focuses its attention to primarily large businesses, specifically those that are in the S&P 500. Azure wants to help businesses manage their challenges and meet their organizational goals. All types of businesses are suitable for Microsoft Azure; however, e-commerce and financial institutions would be able to utilize Azure's features the best. What more is Azure's friendliness toward third parties and open-source APIs, which enable users to switch easily and seamlessly to Azure from other cloud solutions without much trouble.

While Amazon focuses on implementing and polishing in-house approaches for database services, and Oracle chooses to fully support hundreds of different of database implementations, Microsoft goes with a hybrid approach for their database solutions. First, Microsoft develops and maintains in-house solutions like Azure SQL and Azure Cosmos DB to earn more revenue and better support. With in-house solutions, Microsoft employees will be able to diagnose the problems and provide better and faster supports to the customers more easily. This also lowers the employee training time and cost for Microsoft, as there are only limited number of things that employees will need to learn to support customers well. Second, Microsoft knows that since databases have been around for a long time, many people will be hesitant to switch from a try-and-true method to a new database implementation, so Microsoft also offers support to popular database solutions like MySQL, MariaDB and PostgreSQL. Notice that except for Azure Cosmos DB, all other database solutions mentioned are Relational, SQL-based databases. If a business needs to store large, structured data like images, music, graphs, and maps, Microsoft Azure may not be the best service provider to go with. Or if a business requires the use of noSQL or non-Relational databases, then Oracle Cloud, or AWS would be better alternatives. Else, if you want excellent supports from both community and Microsoft regarding all things SQL, then Azure is a good platform to choose. A bit more information about the Microsoft Azure's database solutions can be found in the table 9 down below.

Regarding the price of Azure Cloud, Microsoft does not publicly disclose the prices of their computing and storage solutions. However, Microsoft boasts that Azure's SQL database is up to 85% cheaper than the Amazon counterpart. We do not have a mean to confirm this. That being said, Microsoft is offering the trial of their services for free for the first 12 months, plus the initial 10GB of database storage free. Microsoft keeps its pricing a secret to not have to race Amazon to the bottom, hurting both companies together. Also, from what I have heard, Microsoft charges differently depending on your geographical location. If you are located in Africa, or South East Asia, your cost for Microsoft Azure may be much cheaper than if you are in Europe or America. As far as I know, both Amazon and Oracle charge a flat rate regardless of your location.

Table 2-9 Microsoft Azure Database Services



	Azure SQL Database	Azure SQL Managed Instance	SQL Server on Virtual Machines	Azure Database for PostgreSQL	Azure Database for MySQL	Azure Database for MariaDB	Azure Cosmos DB
Relational Database	✓	✓	✓	✓	✓	✓	
Non-Relational Database (NoSQL)							✓
In-Memory Database							
Data Models	Relational	Relational	Relational	Relational	Relational	Relational	Multi-Model: Document Wide-column Key-Value Graph
Hybrid	✓	✓	✓	✓ *(Hyperscale)*			
Serverless Compute	✓						✓
Storage Scale Out	✓ *(Hyperscale)*			✓ *(Hyperscale)*			✓
Compute Scale Out	✓ *(Hyperscale - read-only)*			✓ *(Hyperscale)*			✓
Distributed Multi-Master Writes (Write data to different regions)							✓
OSS Based Service (Community edition and open extension support)				✓	✓	✓	
HTAP (Available with Azure Synapse Link)	✓ *(Coming Soon)*			✓ *(Coming Soon)*			✓

2.3.6 Local Database Hosting

We want to know all of our options, so here, we are exploring local databases to see whether cloud or local database is best fit for our project. Even with the intensive security measures and extensive features of using a cloud database to store data, it is still much more reassuring the have the data locally. As a whole, a local database is much more secure than a cloud database. The security of a local storage is in the hand of the storage owner, rather than in the hand of those who you do not know. Sure, a by hosting the database locally, you have to take into account temperature, humidity, and power redundancy. But, a local database, if managed correctly, is immune to all sorts of Denial of Service and man-in-the-middle attacks. Since a local database is not connected to the Internet, but rather to the internal network, it is virtually impossible to gain illegal access to the data files without having to be physically in the vicinity. When working on a highly confidential project, a locally hosted database is second to none. You do not have to worry about your Internet service provider snooping your relay package. You do not have to worry about your files being accessed by a potentially untrusted cloud service provider. And most of all, you do not have to worry about cross-VM attacks that could potentially steal your data or making it so that you cannot access your files anymore.

Another major advantage of going with a locally hosted server, rather than a cloud server is the transmission speed and data access latency. Data is transmitted as an electric signal through copper wires. This means it is impossible for data to instantaneous teleport from the remote cloud server to your computer due to physics. The speed of an electron through a copper wire is roughly 1.3e6 m/s. This means that in order for the first bit of the data file to travel from your location to a remote database that is 500km away, it would take 0.38 second. This may not seem to be much but remember that this is just the first bit of the data. To transfer a large data file, the computer needs to first ping the server, then wait for the server to respond before sending the first package. Since the data pipeline is small, a file may need to be split into multiple smaller data packages. This transmission operation could take several minutes, to as much as a couple of hours, depending on the size of the data. Sure, Internet Service Providers are upgrading their infrastructure to support fiber and hasten the transfer speed, but fiber is still not common. If your company functions based on high-frequency data transfer and a single millisecond difference is too large, then a cloud server is not adequate. For example, if your company is a stock trading firm where you need extremely quick access to data to first compute gain probability and then store that data, then using a cloud server could lose you a lot of money from the latency. With a locally hosted server, you can access data instantly without having to wait. This is due to the physical location of you and your server. The closer you are, the less time is needed to receive the data, even if the transfer speed is the same.

Databases come in many different flavors, all of which aim to address the same concerns, that is to transmit the data to and from the users as fast as possible and being as easy to use as possible. These database types are centralized, decentralized / distributed, personal, end-user, commercial, noSQL operational, relational, object-oriented, and graph databases. The naming conventions of these database types are very boring, and they pretty much describe how the database is designed. Each of the currently available database solutions belong to one of these classifications. The purpose of database is to store and organize the information, and the database server is the primary solution to solve the problems regarding to data management. In general, a server manages a large amount of information, which users can access and manipulate the data conveniently. Therefore, for most database system, the fundamental requirements are scalability, high performance, security and provide effective solutions for catastrophic hardware and software failures recovery.

Like the name implied, in a centralized database, the data for everything is stored in a single secured location. The data can generally be access via LAN and can only be modified from that location. Centralized Databases are maximized for integrity and space. Since everything is stored in a single location, there are minimal redundancy of data and it is easier to coordinate the dataflow accurately. Data coherency and versioning is also easier as there will be less conflicts when everything is managed by a centralized system. And because the data is centralized, it is, in theory, easier to build a strong security system around the database. A major downside to a centralized database system is the search time. As the search process is iterative, go through every single file to find the correct one is slow and not efficient. And since everything is stored in a single place, servicing multiple high-bandwidth connections will contest the network, causing major slow down.

The complete polar opposite of the centralized database is the distributed database. Instead of everything be stored in a single location, in the distributed database system, multiple, small databases are connected to a high-speed network. Think of this as a tree where the root is the network itself, and the children nodes are the small databases. Each small database can have smaller databases as its children nodes. An interesting thing about the distributed database is that the nodes in the network can be on the same relative location or they can be geographically far apart. Despite this set up, to the users, distributed database system appears as a single, unified database. This makes it easier for the user to manage files and access data. The searching and versioning algorithms are handled internally without the users' intervention. Distributed databases are known for their data persistency. If a single location encounters a natural disaster and lose all of its data, the system as a whole will still be fine as such data is only a small portion of the entire system. Also, since the system is distributed, chances are that there will not be an excessive load on any single database, reserving the access speed for the system. A downside to a distributed system is that it is difficult to manage versioning. A file may have multiple different versions, with each version being on a different node. When there comes the time to merge, there may be a conflict that prevents the files from merging. Another downside is the difficulty of securing the transmission between the nodes. While a firewall can be built around each node to keep the data safe, the transmission of data is usually handled by a third-party internet service provider, which may not be secure.

Personal, end user and commercial databases are essentially just variations of the centralized database, where the data are stored in a central location and can be accessed either by one or many users. These are generally can only be accessed within a local network and cannot be accessed from far away. As for the NoSQL database, it is a subset of the distributed database system. A unique thing about the NoSQL database is the access of data to and from the database does not require the use of SQL script. For those who do not know, SQL scripts are essentially just a specially formatted command so that the database knows what to do with a piece of data. SQL commands can be complex, which deters many people from learning about databases. With the NoSQL database, data can be accessed without complex script. It is extremely friendly toward new users, who do not have SQL knowledge.

By definition, an operational database is a database that manages data in real-time. It is designed such that when a piece of data is accessed, or written to, the database is modified almost instantaneously. These databases are optimized for speed. With a conventional centralized or distributed database, look up time may be a problem if the database is large. With operational databases, the lookup time is minimal. Operational databases are primarily used in locations and businesses where the quick flow of data is required. They can be found in warehouses, stock trading firms, and hospitals, just to name a few.

Among all the types of databases, relational databases are the most widely used. They are used for data with pre-defined relationships between them. The database consists of tables with columns and rows, with each column defines a category of data and the rows define the entries. This kind of database is suitable for tasks where the categories of data are pre-determined, and that each entry has a unique ID like a name or a number. Relational databases can be found in most corporations with the task of holding the information of each employee. They are also used for polls and census data. An advantage for using relational databases is that they are fast to update. Since each data entry has a unique key, the search time is in the order of O(1). Another advantage is that it is easier to merge tables in relational databases than in distributed databases. Since each table's information is unique, merging them is just adding a table to the end of another table. Relational databases are known for their ability to connect (relate) one table to another table without having to merge the tables together. Figure 17 below details an example for the on-to-many relational database relationships. There are 3 types of associations for relational databases. First is the one-to-one relationship, where one row in table A is only related to one row in table B. Second type is the one-to-many association, where a row in table A is related to two or more rows in table B. And finally, is many to many, where the relationship is bidirectional such that a row in table A is related in many rows in table B, and a row in B is related to many rows in A. A common example of a many-to-many relationship is students taking classes, where a student can be in multiple classes and a class has multiple students.

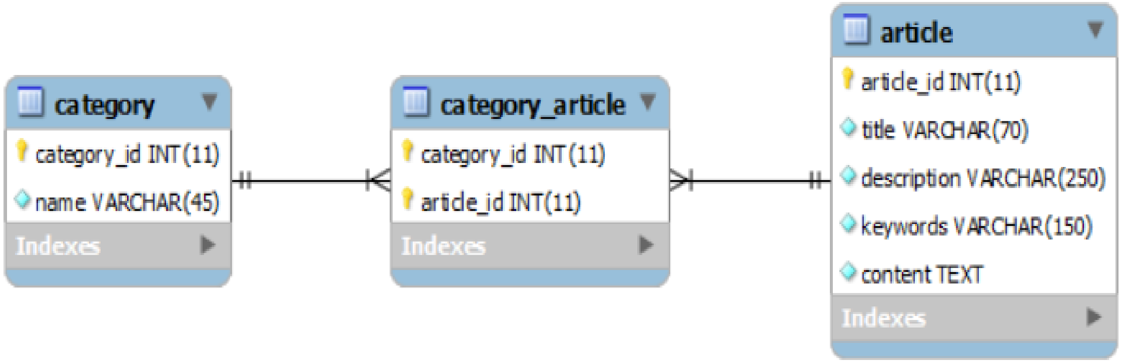


Figure 2-16 One-to-many Relational Database example

Last, but not least, is the Object-oriented database. It is essentially just a specialized version of the relational database. Objects created using object-oriented programming languages like C++ or Java are difficult to be stored inside a database because they do not belong to an already-known datatype. They are their own objects and cannot be categorized with a generic datatype. This is the reason why Object-oriented databases were created. Their main goal is to come up with a scheme to be able to store and organize objects using the relational model. Because each object is unique and are created differently, the databases are organized around data, rather than logic. An advantage of using object-oriented database over the conventional relational database, aside from the ease of storage, is the ease of use. While it is difficult to storage an object in the relational database, it is possible by representing the object as either a hex string or a binary string. However, the flow of the method is the long encode and decode time. With an object-oriented database, the object can be stored directly, saving time by not doing conversion.

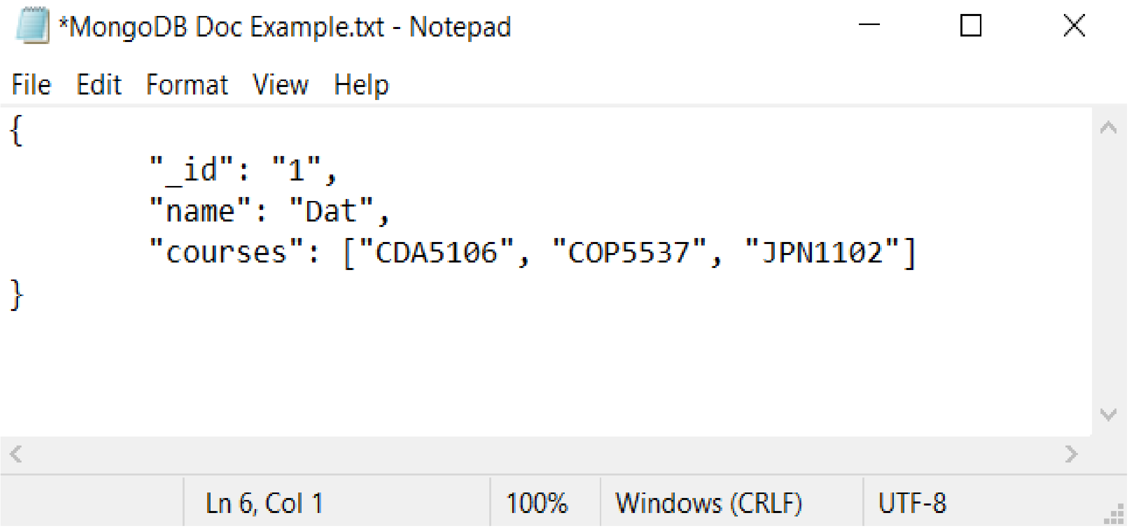
From these types of databases, there are hundreds of thousands of different ways to implement a database. Since the database is hosted locally, pretty much all types of databases are supported. Below subsections 2.3.8.1, 2.3.8.2, and 2.3.8.3 detail some of the more popular ways to establish a database.

2.3.6.1 MongoDB

MongoDB is likely to be second most popular database implementation after MySQL, and the most popular noSQL database. Being a non-structured database, MongoDB essentially eliminates the need for a data table to store user inputs. Non-structured is just another word for being a non-relational database, making MongoDB not reliant on Structured Query Language (SQL) and tables. However, tables are still supported in case they are necessary to do so, and to make the switch from a SQL-based database to MongoDB easy. This means that MongoDB supports both SQL instructions and other types of instructions as well. Hence, MongoDB was named a NoSQL database, or a Not-only Structured Query Language database. Because the database in non-structured, MongoDB is known for its versatility, expandability, and speed. Sure, some relational databases do support non-structured data like with MySQL's support for binary and hex objects, but they were not designed from scratch to support non-structured data like with MongoDB, so the query speed will be much slower.

Having a non-structured database like MongoDB may not seem to be necessary as most of the time, when collecting data, it is the case that data with similar topics will have similar entry field. However, this is not always the case. Sometimes, we would want to add an extra column to the table but only for some of the rows. Doing this in a structured database is a problem and is time consuming, as it requires the user to have a function to update every single row with the new column data. This may not be possible as some entries may not have the data to fill into the field. With a non-structured database, the user can just add a piece of data hanging off the side of the table, without necessary having to add an entirely new row. Non-structured data is not a replacement or an alternative to structured data, but think of it as an add-on, a complement to the structured data. With a NoSQL database like MongoDB, user will not have to worry about if a new data entry will fit the structure of the table.

In MongoDB, like with other NoSQL databases, we have an interesting datatype called Collections. A collection pretty much corresponds to a table in the SQL database, but it is slightly different. We can put objects that we called "documents" into a collection. Essentially, a document is just a fancy name for an entry within a table, but a cool thing about document is that we can have documents with entirely different structures within the same collection without having to modify the collection like we would with a table. A collection does not restrict what the data has to be within a given format. A document is a collection entry, and within a document, there are key-value pairs. The keys represent the fields, or attributes, and the values represent the values of the attributes. You can think of the key as the column header of the table, and the value is an entry within that column. The way we represent such the key-value pairs is the same as how we would represent a data entry within a JSON data package. For those who do not know, JSON, stands for JavaScript Object Notation, is lightweight object format for storing and transporting data. JSON's most common usage is to store frontend user's credentials, and once signaled by an interrupt, the JSON package would be sent to the backend to be processed. The key-value pairs formatting is the key in quotation marks, followed by a colon, and then the value inside the quotation marks. Different key-value pairs are separated by commas, and a document may have as many key-value pairs as it wants. Unlike with a table, the keys within a document do not have to follow any particular order. Users should be cautious that the key most be unique, in other words, no two keys should be the same within a document. This, however, does not apply to the values. A key may have as many values as it wants. Multiple values are inputted as arrays and are encapsulated by a pair of square brackets. Example of a MongoDB JSON document is shown in figure 18 below. The ability to have multiple values within the same field is something that relational databases do not have. For relational databases, for these situations, users must manually create another table to hold the values and leave a reference within the current table such that the current table points to the newly created table, hence relational. And like a JSON package, all key-value pairs within a document are encapsulated by a pair of curly braces. In term of scalability, NoSQL databases are by far the best options. If your dataset is large and you want to add a new attribute to some of the entries, with a NoSQL database like MongoDB, you only need to update what are necessary, however, for a relational database, you either have to create an entirely new table for the attribute or you would need to update the field for every single entry. As the dataset gets larger and larger, more and more abnormalities will appear in the data, making MongoDB more space efficient in the long run.


*MongoDB Doc Example.txt - Notepad
File Edit Format View Help
Ln 6, Col 1
100%
Windows (CRLF)
UTF-8

```
{
	"_id": "1",
	"name": "Dat",
	"courses": ["CDA5106", "COP5537", "JPN1102"]
}
```

Figure 2-17 NoSQL Document Example

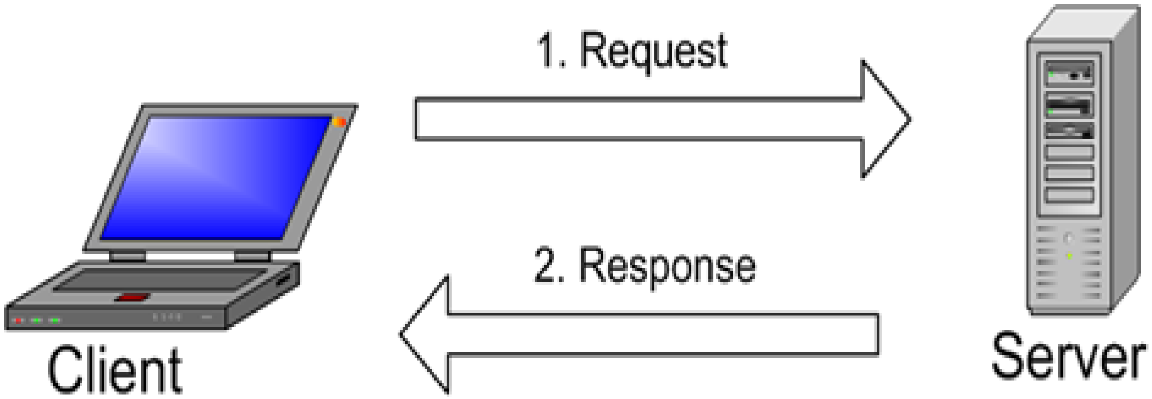
MongoDB query commands pretty much maps one-to-one to the SQL commands. An exception to this is that the “SELECT” command is replaced with the “find()” command in MongoDB. This makes switching to MongoDB from a SQL database very intuitive. MongoDB query commands have been simplified such that they are much easier for a new developer to learn. For example, if you want to fetch from a collection a document with the name “Dat”, in a typical SQL database, you will need to type in a relatively long query string like “SELECT * FROM table WHERE name = \“Dat\””. In MongoDB, however, you can simply type find({“name”: “Dat”}), which is much shorter. Even though MongoDB query commands are shorter on average, the access and transfer speeds may not differ too much from those of MySQL or other database alternatives. For the same set of data with similar ways of organizing the data, the speed of MongoDB is comparable to that of MySQL. However, that is not the be-all-and-end-all. For atypical objects, i.e., objects that cannot be classified as integers, strings, or Booleans, MongoDB’s write and access speed to these objects are much faster than MySQL counterpart. This is because in MongoDB, special objects are stored as-is, while in MySQL, these objects are first encoded as hexStrings before storage, taking longer as a result.

So is MongoDB the ultimate winner in the database war, and that you should use it for every project? No, not necessarily. While it is true that MongoDB is more scalable and is faster in some circumstances, relational databases like MySQL is still better for some situations, the most apparent of which is in modulated projects. For example, each department of a company is in charge of a small component of a project, and each component requires a database, then MySQL would be better than MongoDB in this case. With MySQL, each department will be able to work separately on their own data without having to constantly consult with people from other departments. At the end of the project, instead of merging multiple small databases together and risk creating data conflict, the company can just reference one table from another table without having to modify anything. In the next section, we will dive deeper into MySQL and the SQL commands.

2.3.6.2 MySQL

MySQL is known as an open-source relational database management system (RDBMS) with a client-server operating model. It was originally developed as a side project by Monty Widenius, a co-owner at TcX, in 1979 when he was trying to create a way to simplify his report-writing process. MySQL was first written in BASIC but was later rewritten in C so that it would be compatible with the popular Unix Operating System. At first, this program was called Unireg, rather than MySQL because no SQL codes were involved in the data management process. However, due to the high demand of TcX’s customer for a SQL interface, SQL compatibility was added to Unireg, and it was renamed to MySQL in 1996. Around early 2000, MySQL AB was spun off from TcX to focus its attention solely on the support and development of MySQL. The intellectual property behind MySQL was sold to Sun Microsystems, in 2008. Later, in 2010, Sun Microsystems was acquired by Oracle, and Oracle has practically owned MySQL since then.

A database is defined as location where data is stored and organized. MySQL manages databases based on the relational database model, which means data is organized as tables in computer system. Each table contains row and column, where each column represents a name or attribute of the data, and each row represents an entry in the table. Each entry will have all of the defined attributes. In additional, MySQL allow users modify the database easily and change the structure of the tables to meet expectations. The creation and modification of data within the table is done by the use of SQL codes. MySQL database system is designed to work as a client-server model, which means the users access the data by connecting to RDBMS server. MySQL is becoming so common such that all the big company Facebook, YouTube, Google use for data storage. Moreover, it is also compatible with many computing platforms such as Linux, OS, Windows with primary language is written in C and C++.



This Photo by Unknown Author is licensed under CC BY-SA

Figure 2-18 MySQL Client-Server Model

MySQL works based on client-server structure. When a client wants to access the data, they will connect to the server with a particular network, the client will make request to access the data as desired from the graphical user interface, and the server will provide the output as tables. In the other words, the user will request a specific data using SQL statements, and the server application will respond the request by showing the information asked from the client as long as both sides understand the instructions. While MySQL can be used as stand-alone clients that allow users to interact directly with databases and tables via the use of SQL, this is not the most common way, in which MySQL is used in modern day. MySQL are usually embedded within other programs to provide the capability of relational database. Most of MySQL applications are web-based, rather than local. It is the main component of the LAMP stack, the popular web development software stack. One may think that LAMP stack is not popular simply because the name is not familiar, however, LAMP stack is used as the foundation for many of the popular websites today, like Facebook, WordPress, Wikipedia, Tumblr, Slack, etc.

When mentioning stack, most people will immediately think of the First-In-Last-Out memory structure, however, stack is also used to refer to a collection of software programs that function together to create an application software. These independent software works together such that no additional programs are needed for an application to work. These can be seen as the core components of the applications, while other programs are just add-on peripherals that can be removed at any time. Among all the software development stacks, LAMP stack is likely to be the most popular, as it is introduced as the first application stack in most computing degree programs. What more, it is open-sourced, and thus, free-to-use. For a software development stack to work perfectly, a stack needs to have 4 general components. These are operating system, web server, database, and programming language. For LAMP stack, the operating system is Linux; connectivity is done via the Apache HTTP server; filesystems and data management is done through MySQL relational database; and the programming language is PHP. That being said, recently, many developers are replacing PHP with JavaScript for the front-end development. Figure 20 below shows the visual representation of the LAMP stack. Think of the LAMP stack as just another stack of books where the lower layers support the upper ones. The first layer, Operating System, can be seen as the backbone of the development environment; without the first layer, it would almost be impossible to develop and use programs. Next layer up is the web server software. These are the servers that host applications / websites. It is the server's job to fetch the correct files that store the program's elements for the users to view. The next layer up is the database engine, which stores all the information that are necessary for the end-user and the system. In my opinion, rather than having the database engine being supported by the web server, it would be more appropriate to say that these two components belong to the same hierarchical level, and work hand-in-hand to support the next layer up. The final layer, programming language, is highly dependent on the previous layers to function correctly. Since computers cannot understand human languages, it is the job of the programming language to handle such translations. It tells the computers what to expect and what to do when a user performs a certain action. LAMP stack's goal is to give developers a cost-free environment so that they can concentrate on developing their codes rather than spending time finding the cheapest option for their platform. Because of this, all of LAMP stack's components are open-sourced and are free-to-use.

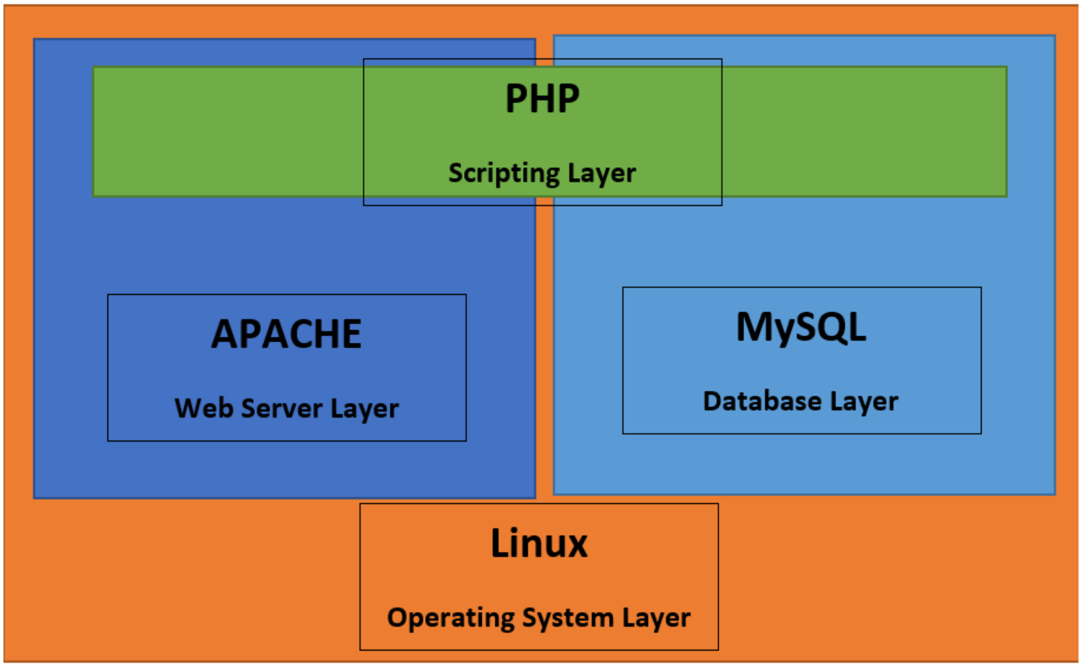


Figure 2-19 LAMP Stack Representation

MySQL is one of the most popular RDBMS software due to the fact that it is user-friendly and easy to access the data from different computing platforms. User can modify the source code as needed, and it is free to install the software including the options to update it. Even though with massive amount of data working with heavy dataset from machine learning, MySQL offers scalability that can work smoothly with excellent speed. In addition, MySQL has been using for years with successful web applications, which means there are huge number of resources for user to use, even for freelance to experts. One of the most important advantage is this database management system provides data security layer, which its Access Privilege System and User Account Management, verification and password encryption are required in order to protect the data.

MySQL consists of multiple different core modules. Note that the term “module” is thrown around rather loosely here, as it denotes pieces of code that logically belongs together and performs a certain critical function, rather than being removable. Here is the general overview of how each of the MySQL components works together. When the server is started on the command line, the Initialization Module parses the configuration file and command-line arguments, and perform a plethora of other initialization tasks, like allocating memory and initializing variables. After the it finishes initialization, Initialization Module calls upon Connection Manager Module to start listening for connections from clients. After the connection is established, the control is passed to the Thread Manager, which determines if the process will requirement one or multiple threads and allocate the resources accordingly. Then, the control is passed to User Module to authenticate the credential of the connection. After authentication, the Commander Dispatcher Module will take control and execute the commands. The Commander Dispatcher controls the Logging Module, Query Cache Module, and Parser to record, cache, and issue commands, respectively. The Parser acts like a multiplexer to redirect the command to different modules depending on the type of the command. The Optimizer, Table Modification Module, Table Maintenance Module, Replication Module, and Status Reporting Module are in charge of SELECT, UPDATE, Repairs, Replication, and Status command, respectively. Once called, each of 5 previously mentioned modules will access the Access Control Module, follow by the Table Manager to fetch and/or update the data. Then, the data will be passed along to the Abstracted Storage Engine to encoding or decoding. Finally, the system control is given to the Client/Server Protocol Module. The Client/Server Protocol Module is responsible for packaging the database responses in the proper format. After packaging, the data is transported to the client via the Low-Level Network I/O. The core components are heavily reliant on the Core API to perform their tasks. Developers are discouraged from using direct libc calls as they could potentially brick the system if not used correctly. Core API provides an additional level of assurance for the reliability of the system. The Core API provides rich functionalities with file I/O, memory management, string manipulation, and the implementations of various data structures and algorithms. The overview of the modules of MySQL is provided in figure 21 below.

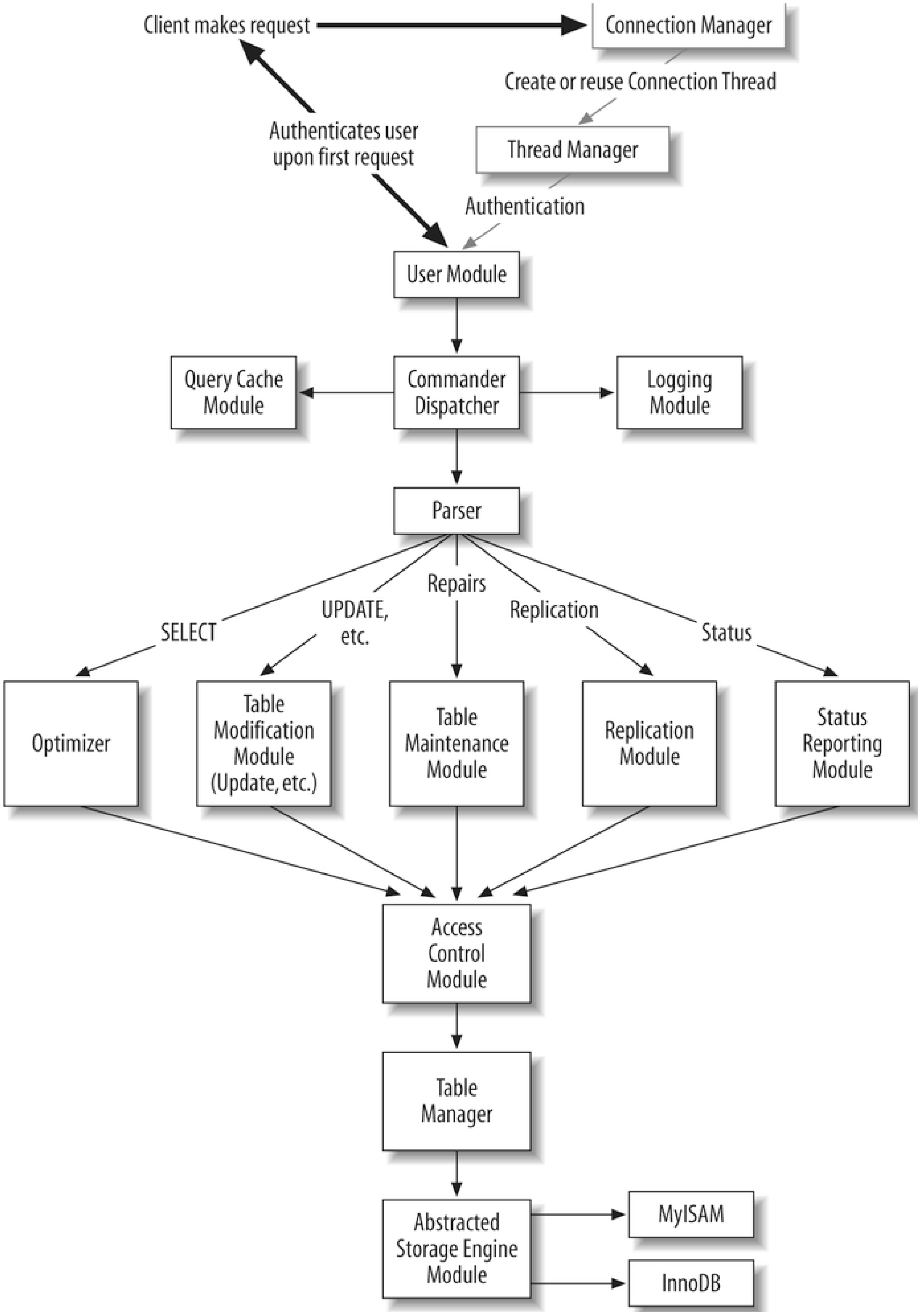


Figure 2-20 High-level Overview of MySQL Modules

While on the topic of MySQL, let us talk a bit about the indispensable element of a relational database, SQL. SQL, stands for Structured Query Language, is a computer language that is used for storing, manipulating, and retrieving data stored in a relational database. The SQL programming language was first developed in 1970 by IBM researchers. The language was called SEQUEL, and its purpose was to manage relational model of data for large, shared data banks. SQL was made publicly available in 1974. Soon, SQL became the standard of the American National Standards Institute (ANSI) in 1986 for relational databases. While some major SQL vendors do modify the language to their desires, most utilize the basic ANSI approved version of SQL for their programs with some extra additions to fit their needs.

Some of the advantages of using SQL are, but are not limited to, the ability for users to describe the data, the ability to create and drop databases and tables, the ability to create view, stored procedure, and functions in a database, and the ability to set different permissions on a table-by-table basis rather than just by database. What more, SQL are embedded within popular programming languages like C++, Java, PHP, and JavaScript. This eliminates the need for the creation of a separate layer of connection between the running program and the SQL decoder. The integration of SQL within modern languages is possible thanks to the development of extensive modules, libraries, and pre-compilers. Just by including a SQL module within a program, users will be able to issues SQL commands just by calling specific functions, rather than needing a dedicated interpreter.

When you are executing a SQL command, the system will determine the best way to carry your request, either by querying or updating, and SQL engine will figure out how to interpret the command. There are 3 main classifications for SQL command keywords. These are Data Definition Language, Data Manipulation Language, and Data Control Language. Data Definition Language includes the CREATE, ALTER, and DROP commands. CREATE handles the creation of database objects like creating a separate database or creating new tables within a database. The ALTER command modifies existing database objects. And the DROP command deletes the entire table or database. The Data Manipulation Language includes SELECT, INSERT, UPDATE, and DELETE command. Each of the name is very self-explanatory. SELECT fetches one or multiple entries in the table; INSERT adds a new entry to the table; UPDATE modifies the existing entry; and DELETE removes an entry from the table. A main difference between CREATE, ALTER, DROP, and INSERT, UPDATE, DELETE is that the former is broader in scale. Data Definition Language commands operate on the entire table or the entire database, while the Data Manipulation Language counterparts only work on entries within a table. An interesting thing to note if you are working with Java programming language is that SELECT, INSERT, and DELETE are considered to be query commands, while UPDATE is considered to be an update command. Lastly, Data Control Language includes GRANT and REVOKE, which gives and takes back privileges to the user, respectively. There is also the special command calls WHERE, which filter out the results. Unlike previously mentioned commands, WHERE is not a standalone command, but rather, it works to supplement other commands. WHERE specifies the attribute needed, such that the database will only return the rows where the condition is true. For example, let us say that we want to fetch all the entries in the table “a” that have the first name “Dat”, then we the following SQL statement can be used: “SELECT * FROM a WHERE first_name = ‘Dat’”. The “*”

from the statement indicates that we want all the attributes of the entries that fit the criteria; if there are no criteria, then the entire table is returned. Subsequently, you can replace “*” with an attributes or multiple attributes such that the database will only return the specified attribute(s) of the entries.

MySQL also includes some other, non-standard commands that may benefit the users. These commands are LOAD, REPLACE, IMPORT, DUPLICATE, etc. The link in the appendix provides the manual for MySQL, which includes the list of all MySQL-compatible commands.

2.3.6.3 SQL Compact

As we know, there are more than $2.5*10^{18}$ bytes being generated and stored every day. Such large amount of data obviously plays a crucial role in decision making for many business operations. Different operation requires different type of database, which gives us the need of using various kinds of databases. It is entirely possible to use one single database to manage a wide variety of data, however, using non-specialized, non-optimized database may cause operation bottleneck as the data transfer speed may suffer from transporting unoptimized data. In today’s market, there are many kinds of databases, though popular ones are generally either relational database or hierarchical database. In this section, we are going to recap a little bit about SQL (Structured Query Language), then we will move on to the SQL Compact database. SQL is a standard language which supports C#, C, C++, and many other languages. SQL Server Compact is a compact relational data base that belongs to Microsoft. It is optimized for applications that run on mobile devices, though it is entirely capable of running on a desktop environment. SQL Compact, like MySQL, can be deployed as a stand-alone, embedded database for client apps. The major difference between Microsoft SQL server (a typical SQL database implementation) and SQL Compact is SQL Compact runs from within the memory that is allocated to the application in which it is hosted. While Microsoft SQL Server, like most other relational databases, runs as a separate application on either the same computer in which the dependent application resides, or inside another computer across a network. For typical relational databases, when we shut down our application, the database will be automatically unloaded. Since the database files are saved into files on the disk, separate from the main application, more space overhead is needed. This would incur unnecessary storage cost for users, especially on mobile devices. This is not the case with SQL Compact, however, as database files are compressed and stored in the same location as the main application, reducing the storage space overhead.

As of today, the latest release of the SQL Compact is public version 4.0. SQL Compact is a free application that can be used for almost anything. It is now even able to run in ASP.NET 4 web hosting scenarios without installing anything from the users, which means we do not need to install or setup anything to their servers in order to use it, but we just need to copy our web application onto any server and use it as a database engine. With that being said, we can build an ASP.NET Web application which have our codes, contents, and SQL Compact database engine. That is how SQL Compact Edition works. Moreover, SQL Compact has small footprint such as 2 MB for a disk and 5 MB approximately for the memory. Its databases rely in a single .sdf file which is 4 GB in size, and it supports both

32 and 64-bit. Not only that, but SQL Compact is also fast, on par with MySQL, in term of transfer speed. The few downsides to SQL Compact are that it does not support large database sizes, with the largest being 4GB, as compared to MySQL database size limit of 65536TB. What more, SQL Compact only supports up to 256 concurrent connections, while other alternatives do not have connectivity limit. This means that SQL Compact is only suited for personal use, or for small-scaled operations that do not require storing a lot of data. The small footprint of SQL Compact is both an upside and a downside. Being small, it is able to store more data within a given space area, however, due to having small address indexing, it is not capable of storing that much data at all. The table 10 below is the summary of some features that SQL Compact has.

Table 2-10 SQL Compact Features

SQL Compact Features
• Single file, code-free database format
• Support for non-admin, embedded installations, no database installation needed
• Integration with Microsoft Visual Studio, WebMaxtrix
• Support for a subset of syntax and data types
• Database for ASP.NET Web Applications and Web Sites
• API enhancements like ADO.NET
• Alternative to JET, XML, SQL Server Express
• Stronger data security with the use of the SHA2 encryption algorithms for encrypting the databases.

3 HARDWARE AND SOFTWARE IMPLEMENTATION

The following sections will detail the choices made for selection of hardware products along with software implementations and the reasoning behind these choices. The research done in the previous sections should provide a reasonable foundation upon which these choices can be made. Compiling reasoning behind these choices include analyzing tradeoffs between implementations in terms of different costs such as those listed in Table 3-1 Design Costs and Importance. The table details the importance of each cost as assessed by the team. Importance is rated on a scale from 1-5 where 5 is in the team’s best interest to adhere to.

Table 3-1 Design Costs and Importance

Cost	Effect if unmet	Importance
Budget	The project will go over budget and result in funds running out, possibly resulting in missed deadlines.	5

Cost	Effect if unmet	Importance
Feasibility of design meeting requirements	The project may not meet critical requirements.	5
Implementation difficulty: time, effort, etc.	The team may be faced with undue stress and time commitments. May miss deadlines.	4
Future expansion	It may not be feasible to easily add additional functionality to the design.	2

It is reasonable to assume that the entire project will be finished by the scheduled end date. By taking precautions in the off chance that the project does not finish in time we can better guarantee a certain set of critical functionalities will be met. The criticality of the requirements from section 1.4 Requirements Specifications along with a brief description and the importance of meeting each requirement is given in Table 3-2 Criticality of Requirements. The criticality of the section is on a scale from 1-3 where 3 is absolutely critical to the completion of the design.

Those sections with higher criticality will need to be completed before the less-critical portions of the design. This will ensure that even if some requirements are not met, the critical path is completed first for baseline functionality of the system.

Table 3-2 Criticality of Requirements

Requirement	Brief Description	Criticality
Body and Mask Case	The body and mask case to house the design	3
Mask Dispenser Motor System	Motor system to open the mask case	3
Button	Button to override and open mask case	2
Temperature Sensor	Human body temperature sensor	1
Distance Sensor	Detecting passing humans	1
Alert System	Alarm system to call out those not wearing a mask, or those with higher-than-normal body temperatures	2
Microcontroller Unit (MCU)	Control of peripherals in sections above	3
Camera	Camera for imaging of faces	3
Image Processor	Main processor for running algorithms to determine if person in the shot is wearing a mask	3
AC-DC Converter and Power Delivery	Converts 120Vac to low voltage for MCU, Image Processor, and peripherals	3
Custom PCB	Custom PCB to implement AC-DC Converter and MCU. Main I/O interconnect for peripherals	3

Requirement	Brief Description	Criticality
(SW) Mask Dispensing Control	MCU controls the mask dispenser and can open or close the case	3
(SW) Peripheral Control	MCU controls and monitors status to the peripherals	2
(SW) Image Processor Interrupt	Image Processor is capable of sending an interrupt to the MCU.	3
(SW) Facial/Mask Detection	Image Processor is capable of using camera to determine whether persons in the image are not wearing masks.	3
(SW) Microcontroller Interrupt Generation	Microcontroller can receive an interrupt from the Image Processor.	3
(SW) Statistics Tracking	Image Processor stores statistics based on people wearing/not wearing masks.	1

3.1 Design Overview Block Diagrams

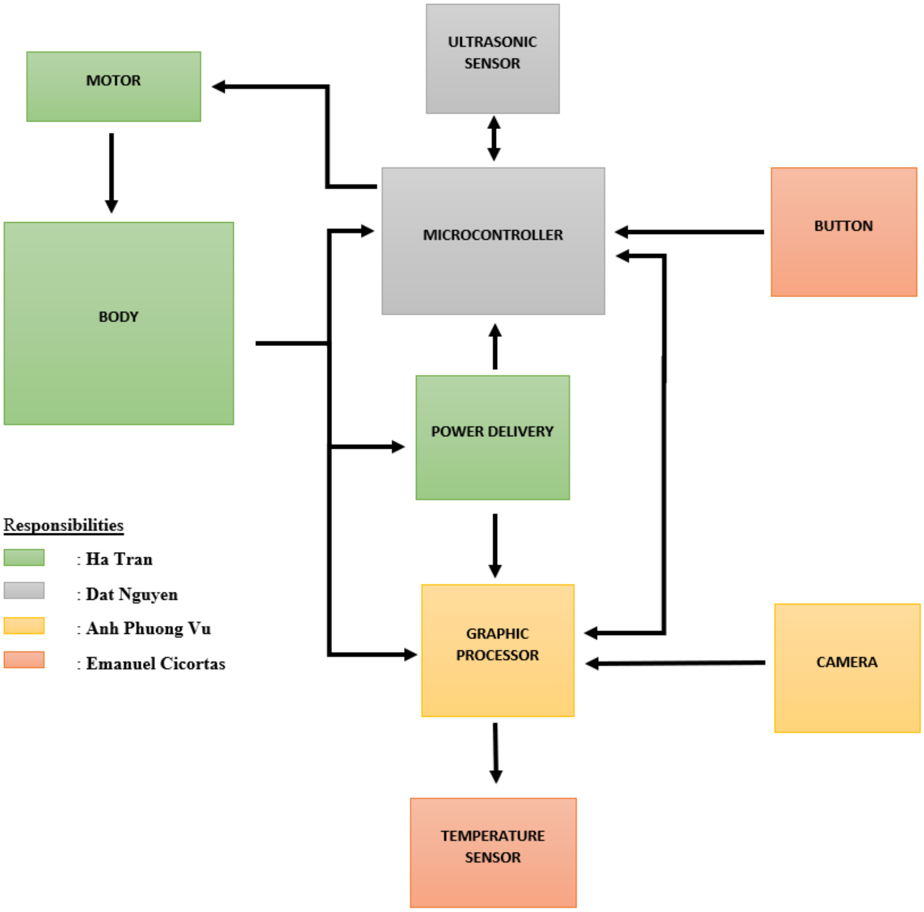


Figure 3-1 Hardware Design Block Diagram

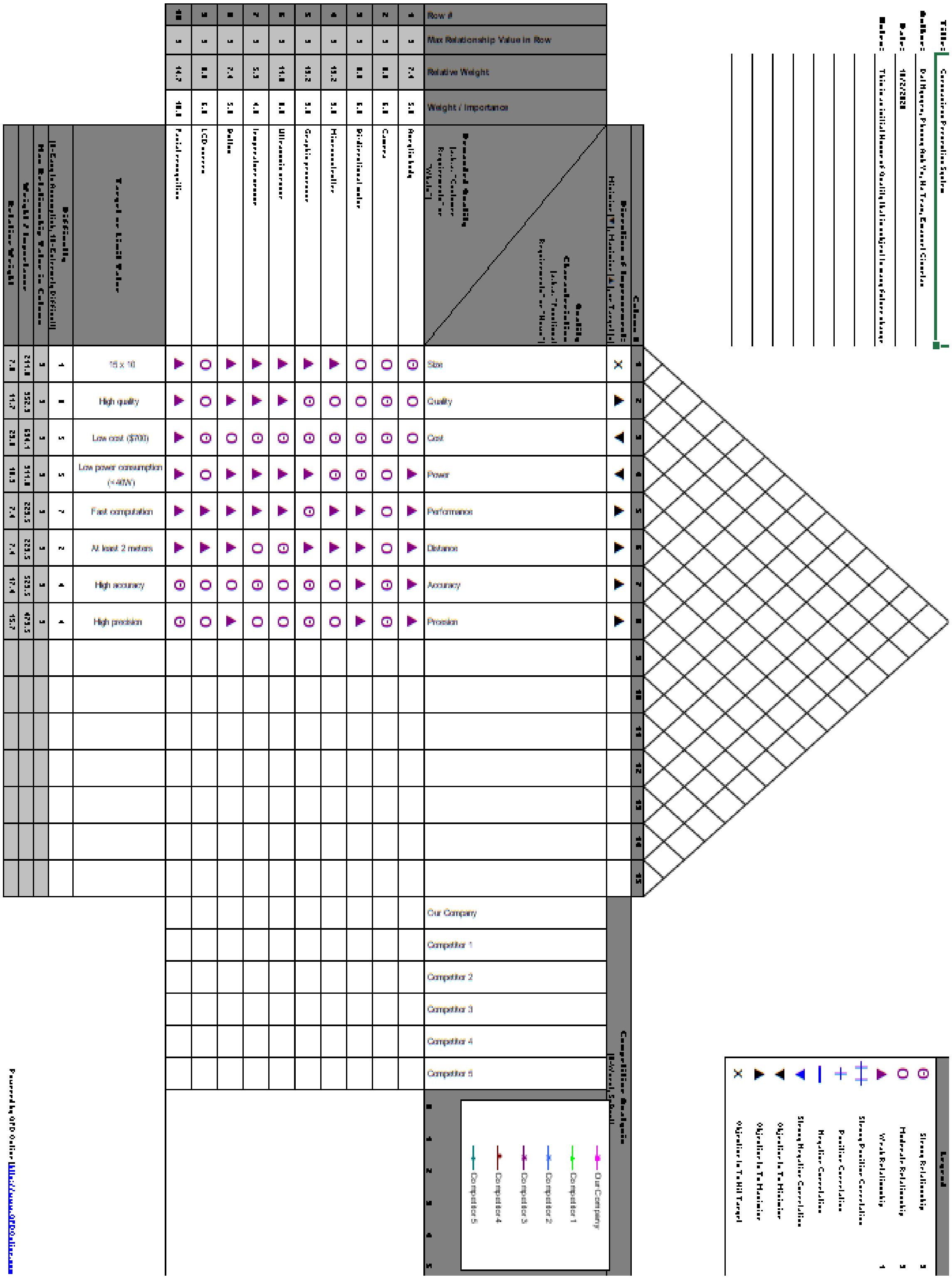
Figure 3-1 above illustrates our plan for implementing the hardware connectivity of the system. The main body will be big enough to house all of the components listed, however, it will not be too big as it will be too expensive. The power delivery system will consist of an AC to AD converter and 3 separate voltage rails. 12V rail will be used to control the motor. 5V rail will be used to power both the motor driver and the Jetson Nano. And finally, the 3V rail will be used to power the microcontroller, as well as the proximity sensor. As for peripheral connectivity, the microcontroller will controller the motor driver, which in turn, moves the motor. The microcontroller will also be in charge of monitoring the statuses of the distant sensor and of the override button. To relaying status, the microcontroller will be connected to the Jetson Nano serially via either GPIO or the UART interface.

The Jetson Nano once received the "go ahead" signal from the microcontroller, will activate its camera, take an image of the subject, and analyze whether the subject is wearing a mask or not. While the analysis is ongoing, the Jetson Nano will also utilize its temperature sensor to measure the temperature of the subject. If the subject's temperature is too high, regardless if the subject is wearing a mask or not, the Jetson will send a signal to the microcontroller for it to ring the alarm. In the situation where the subject's temperature is within the acceptable limit, the facial analysis will continue. Once the result is out, if the subject is wearing a mask, a signal will be sent to the microcontroller to light up a green LED. However, if the subject is not wearing a mask, a different signal will be sent. This signal will make the microcontroller ring the alarm for a few seconds, then the microcontroller will turn on the motor and open up the mask compartment for the subject to reach in a fetch a mask.

Originally, we had wanted to implement an LCD display to greet the person standing in front of the machine and to give them analysis status. However, implementing a large LCD will put an unnecessary strain on the Jetson Nano, which could potentially slow down the computation. If we let the microcontroller control the LCD instead, the LCD will have to be a small, segmented display for the microcontroller to control. This is not a good idea either as if the screen is too small, the subject may not be able to read it, and if they can, some may not even be bothered to read. Hence, we have decided against implementing an LCD display in our machine.

Aside from all the initial computations and the peripheral controls, our design almost makes good use of the storage of the Jetson Nano. We want to contribute our data to science to study about human's behavior to help advance treatments and prevention measures of other devastating diseases in the future. To do so, we implement a MySQL database on the Jetson Nano's Linux system. To make it easy for ourselves, we chose the LAMP stack implementation. We will talk more about the LAMP stack implementation of the database system in section 3.6.3.

3.2 House of Quality Analysis



Title: Coronavirus Prevention System
Authors: Dat Nguyen, Phuong Anh Vu, Hu Tran, Emanuel Cisneros
Date: 10/2/2020
Notes: This is an initial House of Quality (Subject to any future change)

Row #	Max Relationship Value in Row	Relative Weight	Weight / Importance	Demanded Quality (a.k.a. "Customer Requirements" or "Whats")
1	9	7.4	5.0	Amplified body
2	9	8.8	6.0	Camera
3	9	8.8	6.0	Bidirectional motor
4	9	10.2	9.0	Microcontroller
5	9	10.2	9.0	Graphic processor
6	9	11.8	8.0	Ultrasonic sensor
7	9	5.3	4.0	Temperature sensor
8	9	7.4	5.0	Button
9	9	8.8	6.0	LCD screen
10	9	14.7	10.0	Facial recognition

Column #	Direction of Improvement: Minimize (▼), Maximize (▲), or Target (x)	Quality Characteristics (a.k.a. "Functional Requirements" or "Hows")	Facial recognition	LCD screen	Button	Temperature sensor	Ultrasonic sensor	Graphic processor	Microcontroller	Bidirectional motor	Camera	Amplified body	Target or Limit Value	Difficulty (0=Easy to Accomplish, 10=Extremely Difficult)	Max Relationship Value in Column	Weight / Importance	Relative Weight
1	×	Size	▶	○	▶	▶	▶	▶	▶	○	○	⊙	15 x 10	1	9	211.8	7.8
2	▶	Quality	▶	○	▶	▶	▶	⊙	○	○	⊙	○	High quality	8	9	352.9	11.7
3	◀	Cost	▶	⊙	○	⊙	⊙	⊙	⊙	⊙	⊙	○	Low cost ($700)	5	9	694.1	23.8
4	◀	Power	▶	○	▶	▶	▶	▶	⊙	⊙	○	▶	Low power consumption (<40W)	5	9	511.8	18.3
5	▶	Performance	▶	▶	▶	▶	▶	⊙	▶	▶	○	▶	Fast computation	7	9	229.5	7.4
6	▶	Distance	▶	▶	▶	○	⊙	▶	▶	▶	○	▶	At least 2 meters	2	9	229.5	7.4
7	▶	Accuracy	⊙	○	○	⊙	○	⊙	○	▶	⊙	▶	High accuracy	4	9	529.5	17.4
8	▶	Precision	⊙	○	▶	○	○	⊙	○	▶	⊙	▶	High precision	4	9	470.5	15.7
9																	
10																	
11																	
12																	
13																	
14																	
15																	

Competitive Analysis (0=Worst, 5=Best)	Facial recognition	LCD screen	Button	Temperature sensor	Ultrasonic sensor	Graphic processor	Microcontroller	Bidirectional motor	Camera	Amplified body
Our Company										
Competitor 1										
Competitor 2										
Competitor 3										
Competitor 4										
Competitor 5										

Legend		
⊙	Strong Relationship	9
○	Moderate Relationship	3
▶	Weak Relationship	1
‡	Strong Positive Correlation	
+	Positive Correlation	
|	Negative Correlation	
◀	Strong Negative Correlation	
◀	Objective Is To Minimize	
▶	Objective Is To Maximize	
×	Objective Is To Hit Target	

Powered by QFD Online (http://www.QFDOnline.com)

Figure 3-2 House of Quality

From the house of quality from the figure above, it can be seen that there are 9 quality characteristics that we hope to achieve with this project's design. These are: size, quality, cost, power, performance, distance, accuracy, and precision. The sizes of all components, except for the PCB, are fixed at the time of purchase and will not change. Thus, the acrylic housing body should take into account the dimensions of all of the components, as well as the cables connecting them. From a quick glance, a box of dimension 6in x 6in x 6in should be enough to house everything. However, if we take into consideration the masks that we will be putting in there, it starts to become quite cramped. Thus, we have decided to make a box with dimension of 10in x 10in x 10in so that it is easier to work with. We do not want the box to be bigger to save on material cost.

In term of quality, obviously we would want to maximize the quality of the components while staying within the budget. The components selected are all those with a large number of high star reviews. They all belong to reputable manufacturers like NVIDIA, Texas Instruments, and Keurig. And to ensure the ease of use, we made sure that the components selected are popular and have either strong community support (like with MSP430G2553) or strong manufacturer support (like with Jetson Nano).

The cost is likely the most important aspect of this project. Since we are all students, we do not have strong financial backings. Whatever budget we have is from our own savings and our part time jobs. Due to the COVID-19 pandemic, not many companies are looking to sponsors student projects, so we are pretty much on our own. The initial budget we made for this project is $1000 dollars, with the margin of errors of $50, which means we can afford to fork out another $50 over the budget, but not any more than that. To cut back on cost, we opted to go with free and used components whenever possible. Our transformer, microcontroller, and proximity sensors are all used components from other projects. Our resisters and capacitors are all free products we obtained from the school's engineering lab.

Next up is the power and performance as these two go hand-in-hand. The goal is to minimize the power consumption while maximizing the performance of the components. We know that the derivative of power is heat, and since the space inside the housing box is small, it is crucial to minimize the power consumption as much as possible as it is difficult to dissipate heat in small spaces. The box being small compounds with the flammable masks being inside the same container as the electrical components will create a fire hazard. Regulating the power consumption is necessary to ensure the safety of the system. Since we are doing facial recognition, we need high computation power, so we need something that is both powerful and has low power consumption. We ended up going with the NVIDIA Jetson Nano. Although it is nowhere near as powerful as the TX2 counterpart, it consumes much less power, and serves as a great middle ground between power and performance.

For our system to work, we need to have good sensors. For the sensors, the distance that they should be able to measure is at least 1.5 meters. We decided to go with the HC-SR04 proximity sensor to measure the distance as it is advertised to be able to measure up to 4 meters. However, through our testing, it is only able to measure up to around 2 meters. This is substantially less than the advertised range, but it should be sufficed for our application. Our image sensor (camera) is the IMX219 camera module, with the resolution of 3280pix x 2464pix, which should be able to capture an image at 1.5 meters without much trouble.

Lastly are the accuracy and precision. While they sound similar, they are quite different. For easy testing and debugging of our codes, the measurements given by the components must be precise. This means that under the same condition, the device must give the same, or at the very least, close enough result each time. As for accuracy, we went our measurements and computations to be as close to the true value as possible. We know that it is very difficult to achieve 100% precision with our machine learning model on the Jetson Nano, but we want to at least be able to achieve at least a 90% accuracy. For the data collected to be useful at all, the accuracy must be at least this high. Although some scientists and data science journals do accept data of as low as 70% accuracy, to make our results useful to as many people as possible, we want to accuracy to be higher.

3.3 Impact of Realistic Design Constraints

A good project takes into account realistic design constraints to achieve best result while staying true to limits that were established by nature and surrounding mindsets. Design constraints are essentially another set of design requirement. They are not necessarily the widely accepted standards, but rather, the limitations that are imposed upon by the environment and the beliefs of people around. A constraint is a design decision imposed by the environment or a stakeholder that impacts or limits the design. The following 8 subsections describe each of the design constraints and how we plan accommodate them.

3.3.1 Economic

Among all the design constraints, economic constraint is the most relevant to our project. Economic requirements include the costs associated with the development (design, production, and maintenance), and sale of the system. In our case, we only have to worry about design, as this is likely to be a one-off project, so we will not be producing the product in large quantities, nor will we be around to maintain the product after this period. To a using, this system will have little to no economic benefit. People may come to the establishment to try out the new machine, but they will not come back if other factors are not attractive enough. So, in theory, this machine should provide a short-term boost to the business, but on the long run, the business will have to figure something out. As stated, multiple times previously, we are poor students, so the most that we can allocate to this project is $1000. With that being the case, we have to be strategic in choosing our parts, such that the chance of going over the budget is low.

3.3.2 Environmental

Environmental constraints address the impact of the design on the external environment and usage of the earth’s resources. In the 21st century, global warming and natural resource scarcity is the utmost concern for most people and governments around the globe. Many countries, including the United States have been advocating for the Reduce-Reuse-Recycle campaign. Recyclability is defined as the ability to dismantle a product into its constituent materials for reuse in other products. Although it is technically not a constraint placed upon us by the government, we chose to voluntarily adhere to it by using recyclable acrylic as the housing body for our machine, instead of metals or woods. Metals and woods are difficult to recycle, and most of the time, they would end up in the landfills. Acrylic is the best way we know to both cut down on cost and protect the environment at the same time.

Also, since most of the electricity in the United States are produced by coal, we opted for using components with as low energy consumption as possible to lower the machine's carbon footprint. Using low energy consumption devices is our strategy to manage our system's heat output while reducing our electricity usage.

3.3.3 Social

Social and cultural constraints are sometimes grouped together into one group. These constraints address aspects such as benefits, risks, and acceptance of products by the intended user or by society at large. Many great inventions have gone to dusts because people were unwilling to change and accept. To be widely accepted, we believed that a system needs to address a current issue that is highly relevant, and most of all non-polar. The coronavirus pandemic of 2020 is known and feared by many. However, there are those who believe that the pandemic is just a government propaganda, a hoax created by Western governments to make China look bad. We are taking a gamble to with a product that addresses an issue many do not believe. However, we want to be optimistic that the number who are in disbelief are low, and that our product will be accepted by many.

3.3.4 Political

Political constraints address relationships to political, governmental, or union organizations. Currently, due to the COVID-19 pandemic, the United States, as well as many other countries are funneling money into creating an effective vaccine to combat the virus. While the vaccines are being developed, many governments are also funding many manufactures to crank up the production of Personal Protective Equipment (PPE). Since our machine is just a glorified PPE dispenser, so it does not contradict the current political climate in any way, shape, or form.

3.3.5 Legal

The legal constraint is something that is quite difficult to study and research. There are billions of patents, and trademarks out in the wild, so it is difficult to know whether your design violates any patent. However, since we only produce 1 of such machines, and we do not have any desire to make a profit off of it, even if we are violating some patents, it should not matter too much, unless we start selling it. Now, let's say that we are mass producing the machine, then what should we do? To adhere to the legal design constraints, designs should not infringe upon existing patents, copyrights, and trademarks, particularly if the intervention is to sell the product.

One way we could avoid having legal actions taken against us if we are to sell our robot is to do a patent search. Conveniently, the search capability is available on the website of the

United States Patent Office (www.uspto.gov). The most common way of approaching this is to hire a team of professional patent attorneys to conduct a patent search and verify your design for you.

3.3.6 Health and Safety

For any product, the health and safety of its users should be taken into consideration carefully. The health and safety of anyone affected by the final product is an especially important consideration. If users' health is negatively affected from the use of the device could spell legal problems and headaches to the producer. Because of this, IEEE and ANSI have provided a set of guidance on safe level of exposure to radio-frequency electric fields. In our design, the only thing that gets into contact with the user's body is the high frequency sound waves of the ultrasonic sensor. The sound waves are low in energy, so they do not travel that far. What more is that the ultrasonic waves are weak enough to bounce off from the human body, rather than passing through it, making it safe for daily use. Aside from the Corona Prevention System helps people keep themselves healthy even when they are in close proximity with others by providing them with quick access to masks.

3.3.7 Manufacturability

Manufacturability is a prevalent product development paradigm that used to be employed in many engineering organizations. Now a day, people just design the product and outsource the manufacturing to a manufacturing firm to have them deal with it. If the product is difficult to manufacture, the manufacturing team will send to design back to be redesigned. For our design, aside from the PCB, which requires special machinery to produce, most other components come prepackaged and ready-to-go. The acrylic housing body will require some drilling and cutting on our part, but it should not be too unmanageable. To make sure that the PCB is manufacturable, we will have to follow closely the design layout specifications given to us by the PCB manufacture, and layout the components accordingly. The traces should not be too close together, nor should they be too far apart. For the locations of through holes components, they should not overlap if the PCB is double-sided.

3.3.8 Sustainability

Sustainability takes into account not environmental factors, but also economic factors at the same time. To be sustainable, the product must first fit into the required budget. The product needs to use components that are abundant and reuse whenever possible. Our design must be environmentally friendly and be friendly to our budget. The materials must be cheap to obtain and cheap to modify. They must also be recyclable if possible. This means that after use, the materials can be reused in some other ways, or be remanufactured rather than ending up in the landfills. This is quite difficult to do, as a matter of fact. Sustainable materials are usually more expensive than those unsustainable alternatives of the same quality. For example, an electric-powered car is generally more expensive than a gasoline-powered counterpart.

3.4 Facilities and Testing Equipment

Due to the current pandemic, we have very limited options for the locations that we can use to test our set up. Unlike previous semesters where students are able to meet in class and work on their projects in the classroom or in the student union, our options include our own houses, and the limited use Senior Design lab. I guess another "facility" that we could include is Zoom, as we do meet on Zoom from time to time to discuss our ideas and to update our progress on our part of the project.

Among the 3 facilities listed above, the one that we used the most frequent is our houses. Our robot's primary components and functionalities are all very software centric. Very little hardware design and implementations are needed for our project. Because of this, it is entirely possible to work at home. We do our software research, design, and testing on our personal computers. Using our houses as the working facility is rather convenient as we can respond to each other's requests instantaneously. What more, by working at home, we do not need to waste time travelling to school. We could, instead, spend the time difference to do so more research. I think that Zoom is not really a "facility" on its own, but rather, a subsidiary of the home facility. It is not something that we chose voluntarily, but rather, a requirement that arose from us working from home. At home, it is difficult to communicate and get our points across. Thus, it is essential to use a video conferencing program so that we can see and read each other's expressions.

Although our project his very software centric, there are still electrical parts that we need to design, assemble and test. Every system needs a power delivery component to operate, and ours is no exception. Our power delivery requires specialized testing equipment. Sure, we can build the circuit at home on a breadboard, however, we have no way to probe the output voltages or measure the maximum load of the circuit at home. To do so, we need access to a multimeter and an oscilloscope. The mini oscilloscope given to each group by the department may have worked, but for certain parts, we need to probe multiple junctions at once, of which the mini oscilloscope is incapable. For some parts would need someone else to hold the leads for us to properly probe the voltages, so having another group member working on it is good. Also, since some of our parts are used, some of the wires needed to be trim and solder; without a soldering iron at home, such feat is impossible. Listed in the table below is the testing equipment we used to test both our hardware and software. They are listed in no particular order.

Table 3-3 Testing Equipment

Equipment	**Purpose**
Multimeter	Measure the DC operating voltage of each component to make sure they are within specs
Oscilloscope	Measure AC input and output voltage of the transformer. Measure the PWM response of the microcontroller
Soldering Iron	Solder cable leads together

Equipment	**Purpose**
Gator clips	Hold cables in place for easy probing
Breadboard	Test components before soldering them onto the PCB
Code Composer Studio	Microcontroller code debugging
Command line	Machine Learning code debugging
Visual Video	Facial Recognition accuracy testing OpenCV library testing

3.5 Hardware Design

The below 6 subsections detail our design of the hardware components of the robot. If the part was built from scratch, then we will show the build process, as well as the materials used for the part. If the part is bought from a vendor, we will show why we chose such part among all the other alternatives. Then, we will go over the purpose of the part in the grand scheme of functionality of the system.

3.5.1 Housing

The first thing that someone will see when the look at the machine is likely to the outer body. In order to get someone to use the machine, the body must look non-intimidating, but rather, it must be as friendly looking, and as inviting as possible. However, since we are on a budget, we must carefully consider the build material, as well as the design of the housing itself. If we use high-end materials and design the housing to look sophisticated, the building cost would be off the roof. Thus, we opted for something a simple and cute looking design that can be done using cheap materials. This is where our design inspiration, Pixar’s Wall-E, comes in. Pixar was able to design Wall-E to be so simplistic, yet at the same time, adorable. We want to capitalize on this simplicity for our robot to cut down the housing cost as much as possible.

Among all the materials that we can use like wood, aluminum, or molded hard plastic, we have decided to use 1/8” thick acrylic sheets as the body. Although acrylic does not have to elegant look of wood, nor is it as malleable as aluminum or hard plastic, it is much easier for beginners to work with than the alternatives. All we need are a drill and a saw. We will cut each panel separately, sand down the edges, and stick them together using hot glue. Figure 3 below shows a cardboard prototype of our machine’s body. Please note that the cardboard prototype is not to scale with the final version of the robot’s body. The actual body will be roughly 1.5 times as large in all dimensions. The dimensions of the cardboard mockup are measured to be around 6in x 6in x 6in.

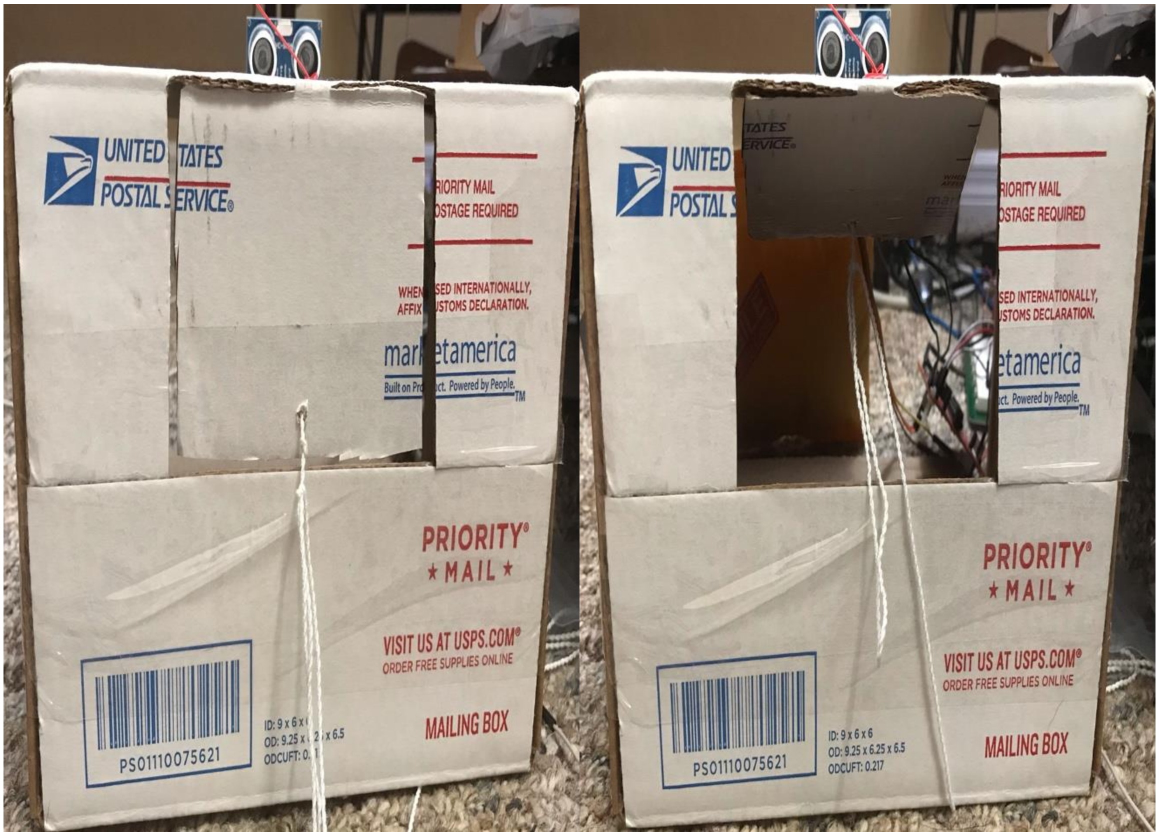


Figure 3-3 Housing, Cardboard Prototype

What we are still undecided about is to whether we should spend some extra money to buy some paint to paint the body to make it look as similar to Wall-E as possible, or should we just leave it clear. Painting the body brown would cost a small amount of money, but we will be able to enhance its similarity to Wall-E, and potentially increase its friendliness with the rusty brown color. However, if we leave the acrylic clear, people will be able to look inside and be able to determine the purpose of the machine at first glance. If people know the machine’s purpose, they will be more likely to test out the machine. If people do not know the machine’s purpose, they will just think of it as a decoration and walk by it. This is such a dilemma.

To add to the robot’s cuteness factor, we plan on adding eyes and arms to the body. The eyes will be huge, and the arms will be short, relatively to the body size to create a sense of neoteny. Neoteny is a human trait where adults would retain some of the features children like having a small head with large eyes. These so called “baby traits” are what human find cute, so by adding large eyes to our robot, we hope that people will give it more attention, and thus more likely to try it out.

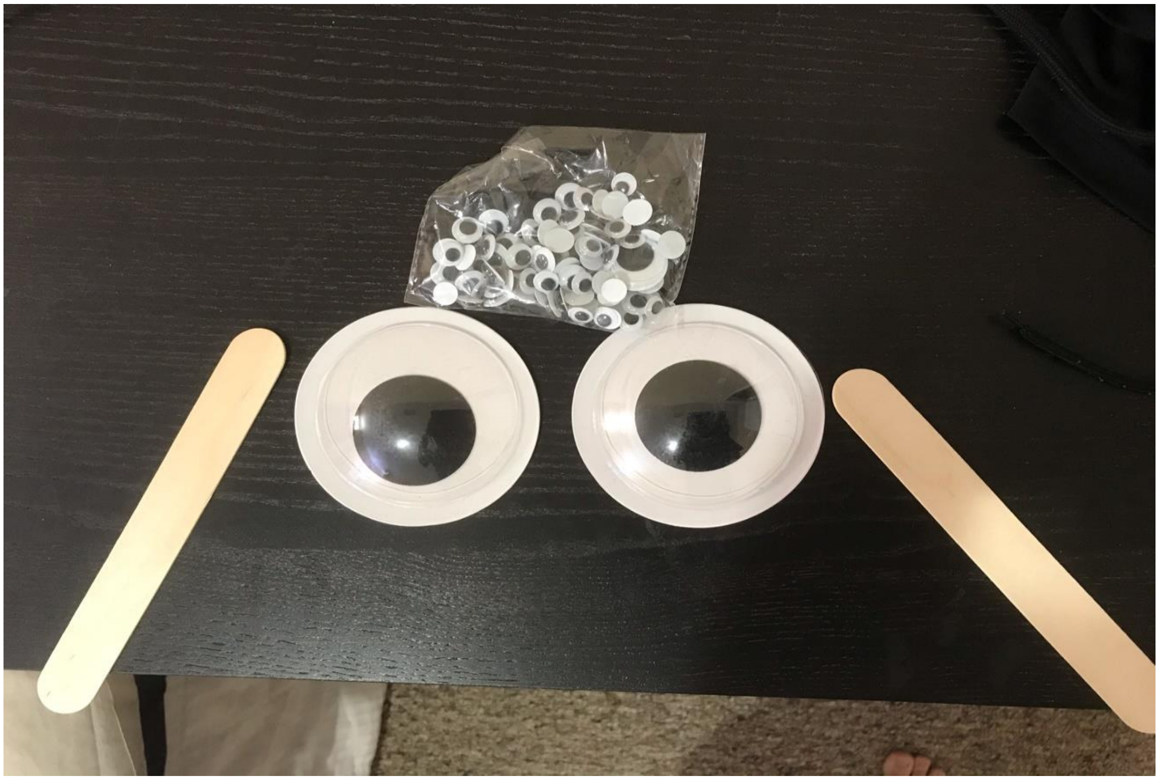


Figure 3-4 Eyes and Arms for the robot

The housing body must be large enough to accommodate all the hardware, as well as some masks. The hardware inside the body will be the Jetson Nano, PCB, and motor. Other components like the ultrasonic sensor and the camera module will be hanged on the outside. From a rough estimation, we found that a dimension of 5in x 5in x 5in would be sufficient to house all the components. However, to make it easier to mount the components and manage cables, we have decided to double all of the dimension to make it more spacious inside. There will be a door at the front of the robot that is controlled by a motor to make it easy for the user to reach in a fetch a mask. The initial design of the door is modelled after the old garage door that is lifted with a string-pulling motor. If this does not work, then we will switch the design to make it look similar to a house door. There will also be a back door for easy internal access if needed.

3.5.2 Motors

The motor in the coronavirus prevention system is just a cherry on top of the sweet pie. It is not required to get the system to be functional, however, having a motor adds to the convenience of the machine as a whole. The motor's purpose is to open and close the front door of the robot's body. Because people are weary of the pandemic, they will not feel comfortable having to push the door in to fetch a mask. This is why we want to make the door to be automatic such that when our machine learning model have determined that the person is not wearing a mask, the microcontroller will turn on the motor to open the door for the person to reach in a fetch a mask.

Initially, we had wanted to use a cheap DC brushless motor to pull the string that is connected to the door. However, after the initial testing, we have discovered that the DC motor is not at all consistent. Given the same activation voltage, same PWM settings, and same active period, the number of rotations in each active period is slightly different from each other. This makes it so that sometimes the door is not shut all the way, or sometimes, the door is only half-opened. The reason behind the inconsistency is because the rotation is generated by charging the magnets. Sometime, the magnets are not charged enough, causing the motor to not spin enough. Some other times, the magnets get charged too quickly and spin the motor too fast; even after the current is cut off from the motor, the angular momentum continues to spin the motor for a bit longer.

This is where the servo motor comes in. Servomotor is a specialized class of DC motors that is capable of enabling precise control of the angular velocity and position of the shaft. Servo motor's limitation over a typical DC motor is that it cannot rotate infinitely. Most servo motors have a 180-degree rotational axis. However, 180-degree is enough for our application. Having a limited spinning range decreases the risk of having the motor spinning too much. A servo motor box is just a gear box. Multiple gears are cascaded together to make it so that the rotatory torque is high. With high torque, the shaft would still be in position even when the motor is off. This makes it ideal for door opening and closing. Figure below shows the image of a mini servo motor.

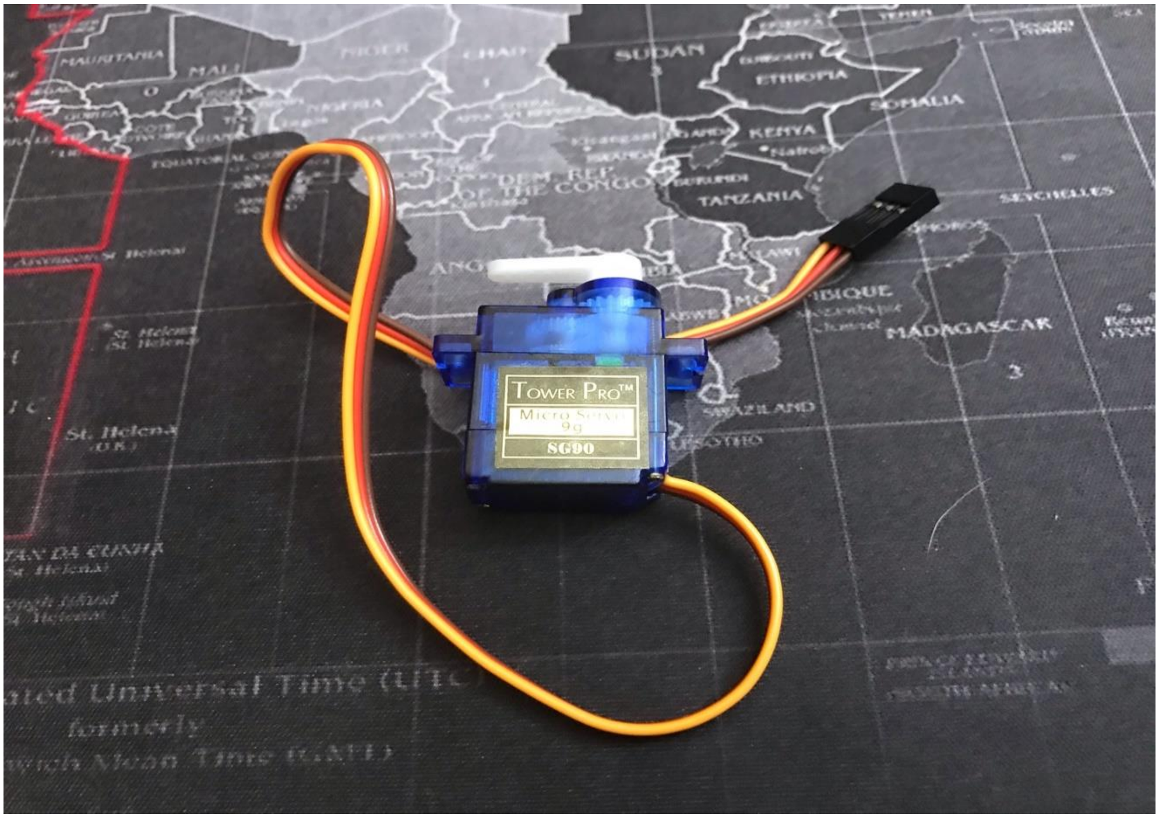


Figure 3-5 Tower Pro SG90 Servo Motor

Both DC motor and servo motor operate in the range of 3V and 12V. We thought that since the MSP430G2553's GPIO pins can output 3.5V when set to high, it should be enough to spin the motor, although slowly. However, this is not the case. While the motors can operate at a range of voltages, they require high current input. Since the MSP430G2553 does not output high current, it will not be able to control the motor. This is why a motor driver like the L293D is necessary to spin the motor. The motor driver takes a separate voltage supply and pump it into the channel(s) that the microcontroller selects. The general pin layout of the L293D motor driver is shown in the figure below.

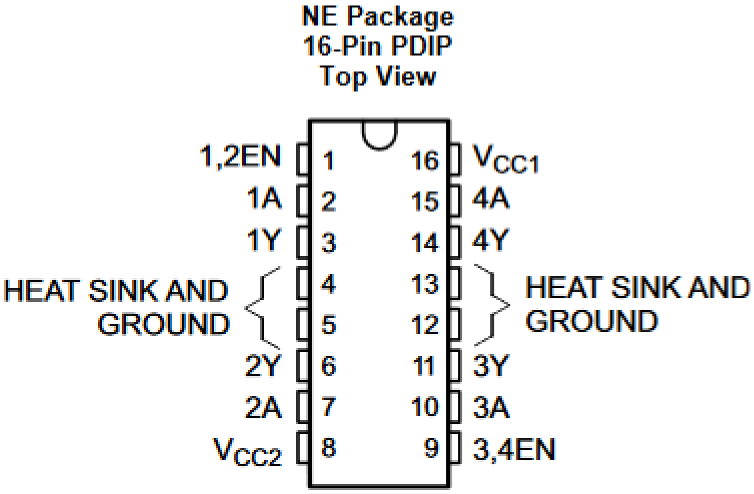


Figure 3-6 L293D Motor Driver Pins

The Vcc1 pin supplies the power to the motor driver, while the Vcc2 pin supply the power to the enabled channels. It is recommended by Texas Instrument that Vcc2 should be in the range of 9V to 12V for the motors to function properly, though we did test the Vcc2 input voltage of 3.3V and 5V and they both worked flawlessly, be it a bit slow. From the figure, you can see that the motor driver supports 4 half motors or 2 full motors. The term half motor is referred to a motor that only spin in 1 direction, while a full motor can spin in either direction depending on the input pins (the A pins). The EN pins tell the driver to read in the voltage level of the corresponding input (A) pins and direct the output current to the corresponding output (Y) pins. How our setup work is that the P1,2EN, P1A, and P2A are connected to our microcontroller via P2.1, P2.0, and P2.3, respectively. To start the motor in one direction, we would send a PWM signal to P1,2EN through P2.1. Then, we would make sure that P2.0 is outputting high and P2.3 is outputting low. This will signal the motor driver to divert the voltage and current to P1A, and set voltage at P2A to 0, creating a complete circuit. To turn the motor the other way, P2.0 will have to output low and P2.3 will have to output high. And finally, to stop the motor, simply disable the PWM signal of P2.1 on the microcontroller.

3.5.3 Sensors and Buttons

For this project, there will be 3 sensors and 1 button that will be implemented. Of those, only 1 sensor is required to accomplish the task determined initially. The rest are superfluous add-ons that aim to improve the quality-of-life for the users. When out the 2 sensors and the button, the machine should still be functioning correctly. However, the codes will be more complex, and the wait time will potentially be longer. The required sensor for this project is the image sensor (photo camera). Without out it, the tasks of facial detection and mask recognition would not be realizable. The quality-of-life enhancements to the machine are the proximity sensor, the temperature sensor, and the override button.

For the proximity sensor, we have decided to go with the HC-SR04 ultrasonic sensor. We did not go with the awesome range of LiDAR or Infrared but chose a relatively short-range ultrasonic sensor because that is all we need. We do not need long distance tracking, but rather, we need high short-range accuracy. While the accuracy of LiDAR does not lack behind that of the ultrasonic sensor, depending on the LiDAR's wavelength, it could be harmful to the health of the users. We do not want to have the risk of having a LiDAR sensor functioning out of specification and cause harm to its users. The LiDAR sensors that are available on the consumer's markets simply do not go through the strict quality control procedures as those made for large corporations. Also, cost is also a factor that needs to be taken into consideration. Since we are on a budget, we cannot afford a high-quality LiDAR sensor. As for those cheaply made Chinese products, we just do not know if they function the way they say they do or if they would just work for one day, and the next they, the frequency increases drastically. We have no way of measuring light waves currently, so the product needs to be from a reputable manufacturer that we can trust. Unlike LiDAR, ultrasonic sensors are available from many US-based reputable manufactures. They are also much cheaper in comparison. We did not choose the infrared sensor because we do not know the lighting conditions of the place where the machine is going to be. If the room has too many infrared signals from sources like sunlight, lightbulbs or computers, the infrared sensor will experience constant interference and the result will not be at all accurate.

At first glance, the ultrasonic sensor may not be necessary in the system. However, it is a way that we can save on energy consumption of the system. If there are no proximity sensors, the Jetson Nano controlled camera will have to continuously be taking pictures, and the Jetson Nano will have to continuously analyze the pictures to detect if there is anyone in front of the machine. Compared to the microcontroller and the proximity sensor, the Jetson Nano and camera use much more power. To conserve power, while the proximity sensor has not detected anyone, the Jetson Nano will be in sleep mode. Once there is someone in front of the machine, the ultrasonic sensor will send an interrupt signal to the microcontroller. The microcontroller will send a message to the Jetson Nano via UART, subsequently causing an interrupt signal to occur in the Jetson Nano, waking it from sleep.

Next is the temperature sensor. With a lack of diversity in the thermal camera market, our only option is the FLIR Lepton 3.5. The thermal camera does not really enhance the user experience from using the machine, but rather, it expands on the machine's toolkit. Since our goal is preventing the spread of the coronavirus, aside from wearing masks, another effective measure is to stay away from people who look ill. This is not discrimination;

rather, it is just a mean to keep yourself safe. Even if everyone is wearing a mask, if you get into close contact with a sick person, the chance of the disease spreading is still high. What the temperature sensor will do is to scan if the person standing in front of the robot has high body temperature. If that person is hotter than the allowable threshold, the Jetson Nano will send an interrupt signal to the microcontroller via UART. Then, the MSP430G2553 will sound the alarm for 3 times to caution everyone around that there is a sick person in the room. This is secondary. Our goal of finding faces and detecting mask is still the primary objective. If we have trouble with the mask detection, the chance that we will have to scrap the temperature detection is high.

As stated previously, since our goal is facial recognition, we will need to be able to capture high resolution images of the face of a person. To achieve that goal, for our photo camera, we have gone with the IMX219 camera module, with the resolution of 3280 x 2464. To put that into perspective, the resolution of a 4k camera is 3840 x 2160. The horizontal length of our picture is a bit shorter than 4k, but in exchange, our image is taller. The reason that we have gone with a development camera module rather than a full-blown camera is the cost. A full-blown camera is around 20x more expensive than a comparable camera module. The difference in cost is stemmed from the inclusion of an onboard image preprocessor. For our application, we do not need the image to be sharpened before being processed by the AI, so a camera module alone is more than enough.

Lastly, let us talk about the use of a push button in the system. The button is just the typical single push, single throw button. This is added as a failsafe option in case someone or something is blocking the ultrasonic sensor, preventing it from detecting an approaching object. This also helps in case the robot is mounted in high places and the person is too short to be detected by the ultrasonic sensor. Whatever the case may be, all the user has to do is to push the button to override the ultrasonic sensor detection. After pressing the button, the Jetson Nano will be turned on to do the AI processing sequence like usual. The reason we use the button to override the ultrasonic sensor and not the jetson nano is because the IMX219 camera module has a much wider field of view than the ultrasonic sensor, so the chance of it not being able to capture an image that does not contain a person is low.

3.5.4 Microcontroller

Previously, we compared two of the most popular microcontrollers against each other. These were the Microchip's ATmega328 and the Texas Instruments' MSP430G2553. Ultimately, the microcontroller we have chosen for basic peripheral control in our system is the MSP430G2553. Although the ATmega328 supports for GPIO lanes, as well as having double the amount of flash memory as compared to the MSP430G2553, its 8-bit processor is the primary factor that held us back from choosing it. For our system, we only need the microcontroller to control 1 proximity sensor, 1 buzzer, 1 active low push button and 1 to 2 servo motors. This simply means that our code will not take up that much space, so the 16KB of flash memory should be more than enough. Also, because we only need at most 4 peripheral connections with 1 full UART connection with the Jetson Nano, we do not need that many GPIO lanes. Because we need to calculate a large range of distance with the proximity sensor, we need the computation limit to be higher than just 8-bit numbers. With an 8-bit number, the upper limit is 255, while with the 16-bit number, the upper limit 65535, 256 times larger. Through testing, I have seen the proximity sensor returning a number as large as 8700 for a 1.5 meters measurement. This means that the 8-bit processor simply will not be able to do efficient computations on the number. Sure, it is entirely possible for an 8-bit processor to do computations on a 16-bit numbers by cascading 2 registers together and use 1 to hold the 8 most significant bits and use the other to hold the 8 least significant bits. However, such operation comes at a cost. Computations that span across multiple registers take many clock cycles to complete as compare to 1 clock cycle computation on a single register. This slow down does not seem like much, but over a period of time, the small-time difference adds up. If the ATmega328 had had a 16-bit processor like the MSP430G2553, we would likely have chosen it instead. The excess amount of GPIO lines and the large number of PWM-enabled pins come in handy when you do debug.

Rather than using a G2ET development kit for the MSP, we minimized the footprint by putting the MSP430G2553 microcontroller chip on the same PCB as the power delivery system. We do not need the fancy features provided by the development board like easy computer connection with the micro-USB port, nor do we need the 8-segments LCD screen. We do not even need a reset button for our current project. We only need direct access to the pins for the peripheral control and to set up the buttons. Also, since the interior of the robot is relatively small, putting another board into the body would take up too much space. We believe doing so is the right choice as the size of the microcontroller itself is only about 1/100th of the size of the entire board. Currently, since we do not have the PCB configured and ordered yet, we have set up the MSP430G2553 on a breadboard and is testing it from there.

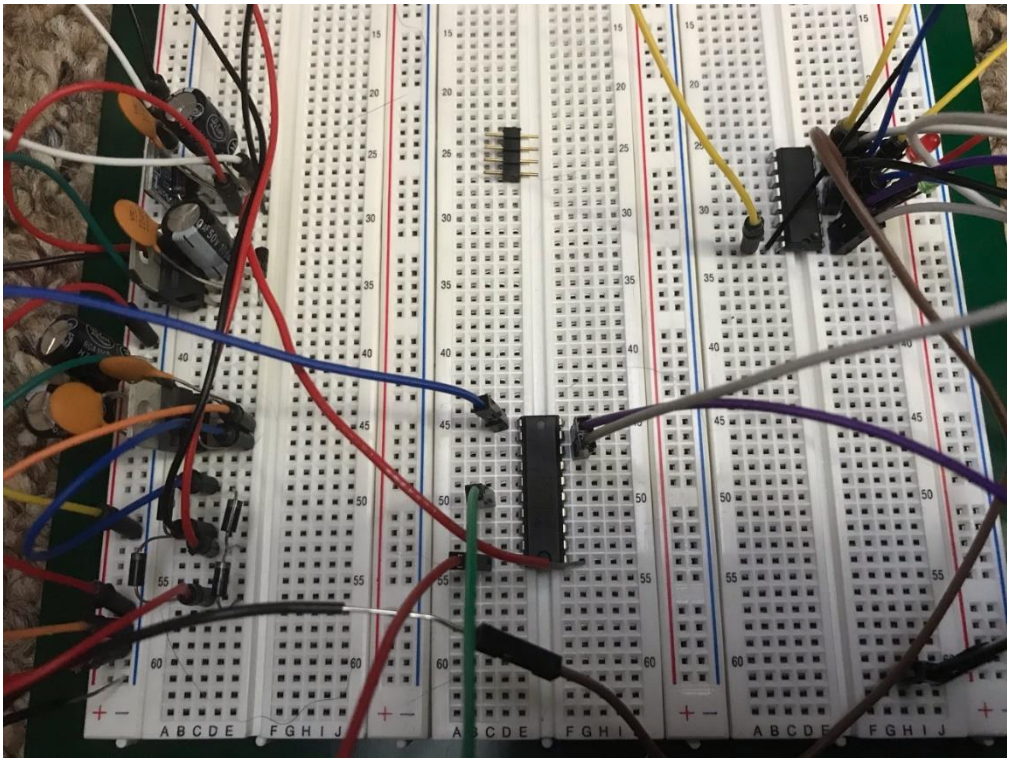


Figure 3-7 MSP430G2553 Setup on the Breadboard

As for the port and pin connectivity to and from the MSP430G2553 microcontroller, here is how it works. DVCC is the power source of the chip, so it is connected to the 3.3V rail of the power supply. Obviously, another terminal is needed to complete the circuit; so, we then connect the DVSS to the ground terminal of the power supply. Because the reset pin is active low, we must, at all time, bridge the reset pin with DVCC via a 47k resistor to hold it high. Without hooking it up the voltage source, the reset pin will float between high and low and constantly reset the microcontroller. This phenomenon is similar to a boot loop many people have experienced with the personal computer. The push button will be configured to bit 3 of port 1 of the microcontroller. As stated in the motor section, the motor will be hooked up to the motor driver, then the motor driver will be linked with the microcontroller via P2.0, P2.1, and P2.3. Finally, the buzzer will be linked to P2.4 and P2.5.

According to Texas Instruments' datasheet, the MSP430G2553 consumes no more than 1.4mW even under heavy load. MSP430G2553 takes in 3.3V as the operating voltage. Under heavy load, the input current is measured to be roughly 400uA, which is within expectation. What more, under low-power mode, the power consumption dropped drastically. With low-power mode 3, the amount of current it needs dropped to roughly 2uA, making the required power decreased to 6.6uW. Since the MSP430G2553 does not need that much power to operate, it can be powered very easily. Since our 3.3V rail of the power delivery can deliver up to 5W of power, it should be more than enough to power not just one, but multiple MSP430G2553 at once.

3.5.5 Image Processor

The image processor used in this design is ultimately chosen to be the NVIDIA Jetson Nano. The decision behind choosing a graphics/image processor development board in this case was simple due to both the specifications of the Nano being far above its competitors in the same price range and abundant support from NVIDIA. Among the abundant support for the Jetson platform is a face mask detection flow that has already been fleshed out officially by NVIDIA as an example design and the methodology can be easily found online.

The two other graphics processors we researched were the Raspberry Pi 4 and the ASUS Tinker Board S. Among these there were few others in the market at our price range and these came out on top. The reasons for choosing the Jetson vs. the other two products are given in the following sub-paragraphs. An image of the used Jetson Nano is shown below in figure 38.

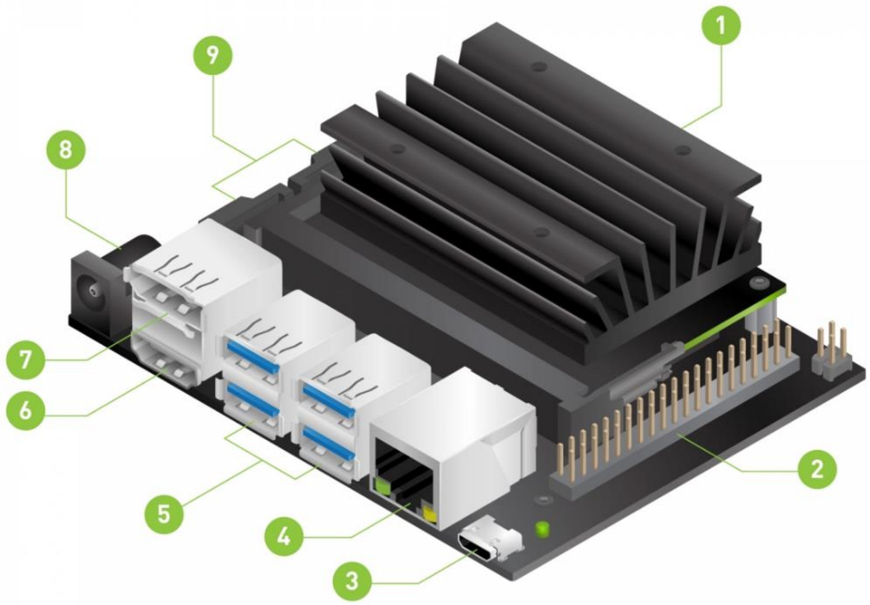


Figure 3-8 Jetson Nano Image

3.5.5.1 Processing Power

In summary: the Jetson Nano is chosen as the Image Processor due to its GPU power.

The main reason for choosing the Jetson Nano is that it has the strongest GPU – a 128 CUDA core NVIDIA Maxwell-series. Essentially, since we have an artificial neural network at the heart of our processing that consists of many simple multiplication and addition operations needed to be done in parallel, a CPU would not be very efficient at doing this since it is both inherently sequential and has a low number of cores, between these systems four at most. What a neural network needs for good performance are many “dumb” cores since we don’t care about concepts such as cache or a complex ISA provided by CPUs since what we’re doing is just simple multiplications and additions.

The element in a system used to realize this function is a GPU, where the Nano excels. At a low level a GPU consists of a large array of simple cores primarily specialized in these simple multiplication and addition problems – exactly what is required for acceleration. Additionally, since we are operating in the image processing realm this requirement is even more stressed due to it already being computationally intensive to even stream video.

In this category the Raspberry Pi is already out of the running due to the fact that it has no GPU. Between the Asus Tinker S and the Nano, the Nano greatly outperforms in this category with 128 CUDA cores vs. the ASUS’s ARM Mali GPU with a whole 16 GPU cores. Something that needs to be realized as well is that we need to actually be able to sufficiently utilize the GPU available. In terms of support, the Nano has a massive platform provided by NVIDIA for CUDA core drivers, making sure we get the maximum performance possible out of the GPU, so this is not a problem since it is built into their deployment SDK.

3.5.5.2 Power Delivery/Consumption

In summary: The Nano requires 15W of operating power. The PCB will provide a 5V DC power source with up to 3A of current possible to the Nano, with a minimum voltage sag of no lower than 4.75V, through two 5V VDC pins to the GPIO header.

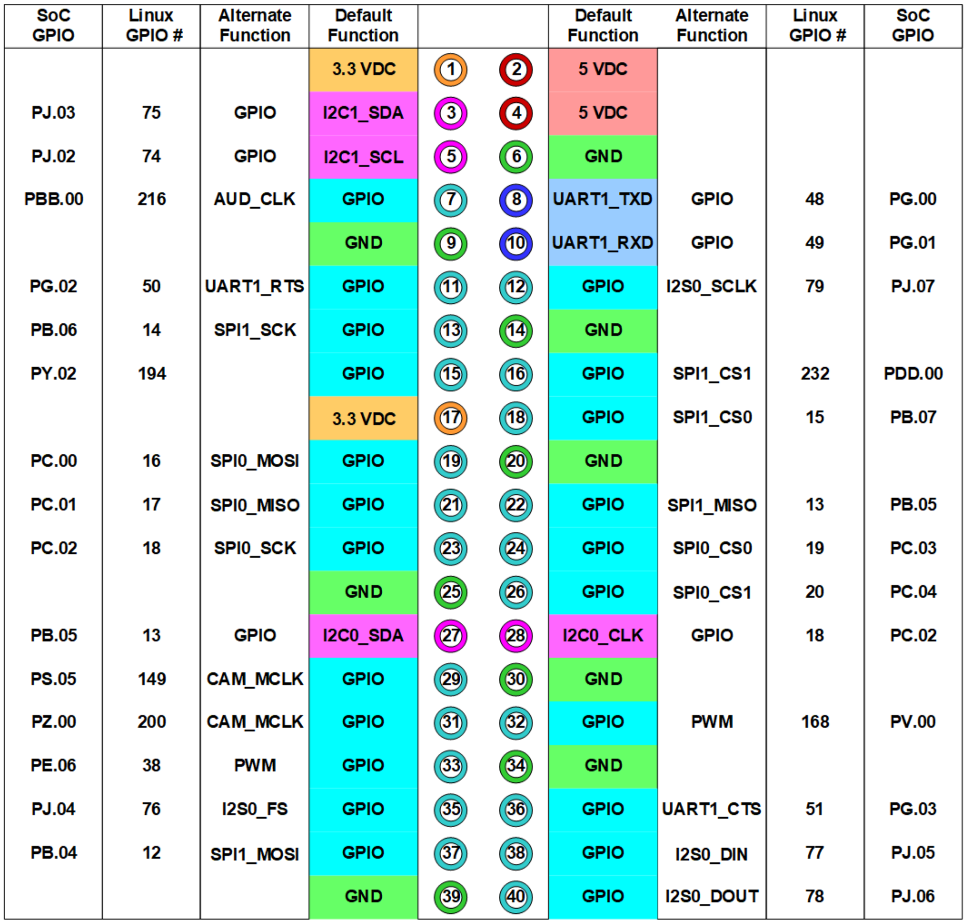
Power consumption was not a major reason for differentiating between any of the boards, however we must still take it into account when designing the hardware. The chosen Jetson Nano requires a steady 5v/2a DC power source in order to run at its peak 10W power usage for optimal performance. This is not taking into consideration the carrier board or attached peripherals, which may end up taking more power. NVIDIA's recommendation is a 5v/4a DC source to provide a stable voltage rail with no droops below 4.75V to prevent brownouts. This is a hard requirement due to the fact that we need to be able to run the Nano as hard as possible since we cannot afford any performance losses in our AI algorithm.

There are three different ways to power the board. The micro-USB route has been seen to be very unreliable in testing and will not be used. The 2.1mm barrel jack connector can supply up to 5v/4a and is a possible choice, however the simplest way to power the nano is through the 5v VDC headers (pins 2, 4) shown in Figure 38. They may take up to 2.5A per pin, however we will not require the full 5A supply since the Nano itself takes up 2A, and the only peripheral connected is the Camera module which should in no way take up more than an entire amp of power. Therefore, a loose requirement is 5V/3A connected to pins 2, 4 on the header.

3.5.5.3 Microcontroller Interface

In summary: the Jetson Nano provides one GPIO pin at 3.3V to the microcontroller and a UART interface for receiving serial data to the microcontroller. These pins are located on the 40-pin J6 header in Figure 3-9 Jetson Nano Pin Configuration. The UART is located on pin 10 (UART1_RXD). The GPIO used to signal to the MCU is located on pin 12. GND is used between boards on pin 6.

The Jetson needs to support an interface to/from the microcontroller to allow for signaling from the distance sensor and to the mask dispenser motors to open the mask case. There are many ways to proceed in this case however the simplest is to have a UART connection between the MCU and the Jetson Nano since it's an extremely simple protocol with a lot of support. In testing, we found that the MCU was having issues receiving UART data from the Jetson, so we ultimately used a single GPIO connection for Jetson Tx, MCU Rx, and a UART connection for vice versa.



SoC GPIO	Linux GPIO #	Alternate Function	Default Function			Default Function	Alternate Function	Linux GPIO #	SoC GPIO
			3.3 VDC	1	2	5 VDC			
PJ.03	75	GPIO	I2C1_SDA	3	4	5 VDC			
PJ.02	74	GPIO	I2C1_SCL	5	6	GND			
PBB.00	216	AUD_CLK	GPIO	7	8	UART1_TXD	GPIO	48	PG.00
			GND	9	10	UART1_RXD	GPIO	49	PG.01
PG.02	50	UART1_RTS	GPIO	11	12	GPIO	I2S0_SCLK	79	PJ.07
PB.06	14	SPI1_SCK	GPIO	13	14	GND			
PY.02	194		GPIO	15	16	GPIO	SPI1_CS1	232	PDD.00
			3.3 VDC	17	18	GPIO	SPI1_CS0	15	PB.07
PC.00	16	SPI0_MOSI	GPIO	19	20	GND			
PC.01	17	SPI0_MISO	GPIO	21	22	GPIO	SPI1_MISO	13	PB.05
PC.02	18	SPI0_SCK	GPIO	23	24	GPIO	SPI0_CS0	19	PC.03
			GND	25	26	GPIO	SPI0_CS1	20	PC.04
PB.05	13	GPIO	I2C0_SDA	27	28	I2C0_CLK	GPIO	18	PC.02
PS.05	149	CAM_MCLK	GPIO	29	30	GND			
PZ.00	200	CAM_MCLK	GPIO	31	32	GPIO	PWM	168	PV.00
PE.06	38	PWM	GPIO	33	34	GND			
PJ.04	76	I2S0_FS	GPIO	35	36	GPIO	UART1_CTS	51	PG.03
PB.04	12	SPI1_MOSI	GPIO	37	38	GPIO	I2S0_DIN	77	PJ.05
			GND	39	40	GPIO	I2S0_DOUT	78	PJ.06

Figure 3-9 Jetson Nano Pin Configuration

Hence the Nano will be connected to the MSP430G2553 via GPIO for Tx, and UART for Rx at a baud rate of 9600. The states of the algorithm in hardware will be:

1. Nano is in sleep and MCU distance sensor does not indicate any objects within vicinity.
2. MCU sees an object in vicinity using its distance sensor, or the button is pressed. It then signals to the Jetson Nano over UART with an ‘S’ to signify a start condition.
3. The Nano receives the ‘S’ over UART, and begins the mask detection algorithm. Once the algorithm finishes, we update the database based on the metadata from the AI and determine if someone has been seen without a mask. If true, then we signal to the MCU by toggling the GPIO pin.

3.5.6 Power Delivery

For power delivery for this project, we are going to use an external power brick (AC-DC converter) which converts from 120VAC to 12VDC, then, our PCB will implement 2 rails with voltage regulators that turn the input 12VDC to 5VDC and 3.3VDC respectively in order to run motor driver, MSP430G2553, and the Jetson Nano. As we know that AC-DC converter are electrical circuits that use to transform alternating current (AC) input into direct current input (DC). Now, let us talk a little bit about AC and DC. Direct current (DC) is a constant and flows in one direction. However, alternating current (AC) varies with time. It flows one way for a while, and the other way for a while repetitively. Moreover, a device that is used to convert from AC voltage to DC voltage is called a rectifier. A rectifier works by using diodes because the diode has non-linear characteristics and allows current to flow in only a single direction. Moreover, a DC voltage is required to use in most of the electronic devices such as laptop, television, stereo systems.

Figure 3-10 below serves as both a schematic, as well as a software prototype of the power supply that we are going to build. Below that, in Figure 3-11 is our equivalent breadboard setup of the same circuit. In our circuit, we have 3 voltage rails as the power outputs to required components. The outputs are connected to motor driver, MSP430G2553, and Jetson Nano. Now, let us walk through the entire circuit to see how it operates. First, the input signal is generally in the range of 120V(rms), 60Hz AC signal that comes out from United States' wall outlets. The input signal is connected directly to the primary side of step-down the transformer. The step-down transformer that we used in this circuit that converts from 120VAC to 14.5VAC. Based on what we have discussed in the previous sections, we are now able to calculate the number of turns a transformer has by using the ratio equation $v_i/v_s = N_1/N_2$ which means the number of turn on the primary side (input) is 240 and the secondary side is 29 (output).

Next, the output of the transformer is connected to a full bridge rectifier which is made by using four diodes in the arrangement of the picture below. What is a diode? The diode is like an electrical check valve that allows current flows in one direction but blocks it from going back. The bridge rectifier circuit is also used to isolate the input ac with the rectifier output in order to prevent the risk of electrical shock. All the diodes used are 1N4001 which has forward bias voltage of 1.1V and maximum current support of 1A, and we chose 1N4001 because they were cheap. By using full bridge rectifier, it converts both positive and negative peaks into positive bumps at the output. In other words, for the positive part of the sine wave, we pass the same voltage, and for the negative part of the sine wave, we pass the inverted voltage. The conventional current will always flow from high potential to a low potential. So, during the positive half of the input voltage cycle, the top part of the right side of the transformer is going to have a positive potential which makes the diode D2 and D3 acting as forward biased (ON), D1 and D4 will be reverse biased (OFF). Similarly, during the other half of the input voltage, the bottom part of transformer will have positive potential which makes the diode D1 and D4 are forward biased (ON), D2 and D3 are reverse biased (OFF).

At the output of the full bridge rectifier, we now have a DC voltage. However, the output is not a flat line DC signal that you would think of, but rather, it is a pulsating DC signal because the negative part of the AC sine wave gets flipped over the x-axis. The rectifier only flips the negative voltage value and makes it positive, but it does not change the positive voltage range of the input. For example, if the input's AC signal goes from -5V to 5V, then the rectifier will convert all negative values to positive, but it will not remove the voltage fluctuation. So, the output voltage is now 0V to 5V. The is not an ideal situation as it acts like a PWM signal and will only supply the effective voltage of ½ of the necessary output. If the period difference is too long, or if the effective voltage is too low, the device that we are powering may not be able to stay on, let alone function correctly. Our desired DC output signal is one that is as consistent as possible and as smooth as possible. To convert the pulsating DC signal effectively and efficiently into a smoother DC signal, we need to add an array of capacitors or inductors into the output of the rectifier. As we know, the capacitors are widely used in all kind of electronics as low-pass, band-pass, and high-pass filters. They are favored over inductors for their small size and easy replaceability. We chose to use the capacitors to basically filter out variations in voltages and achieve our ideal flat and smooth DC voltage, and the capacitors that we used in the circuit have its voltage greater or equal to the DC output voltage so that we will get pure DC output voltage without any ripples.

As we can see in the circuit, we used 470uF polarized capacitors. Since the output comes out from the full bridge rectifier is DC, we must use polarized capacitors to filter out because polarized capacitors can only be connected one way in a circuit and they also provide large values of capacitance If we connect it wrongly, we will not get pure DC output and sometimes it blasts. Although in the schematic we showed 470uF polarized capacitor, we actually used the 100uF polarized capacitor for our build instead and it worked perfectly for us. Now, we can start connecting voltage regulators to drop out the voltages that we want it to be. We used LM7812, LM7805, and AMS1117 that can drop to 12V, 5V, and 3.3V respectively. In each of the voltage regulators, it has three pins which is known as INPUT, GROUND, OUTPUT respectively. The input is a positive unregulated voltage, ground is common to both input and output, and output is the output regulated is taken at this pin. Especially, a reverse-biased diode is needed over the regulator because it's used to protect device against input short circuit or there is a heavy load on the input supply voltage is switched OFF while the output supply remains ON. Otherwise, the output capacitor will discharge through the parasitic diode and transistors in the regulator and the current might be high enough to damage the regulator. That is how the circuit works. Lastly, the circuit has been built and tested, and we got the desire output voltages from the circuit.

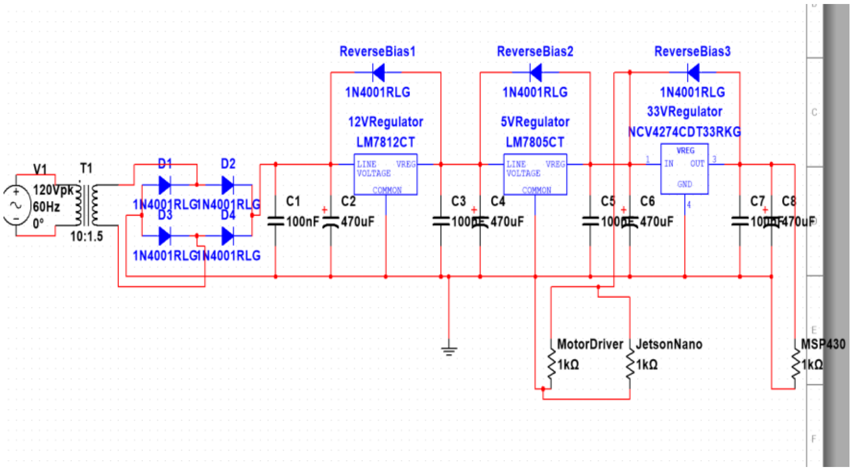


Figure 3-10 Initial Power Delivery System Schematic (Scrapped)

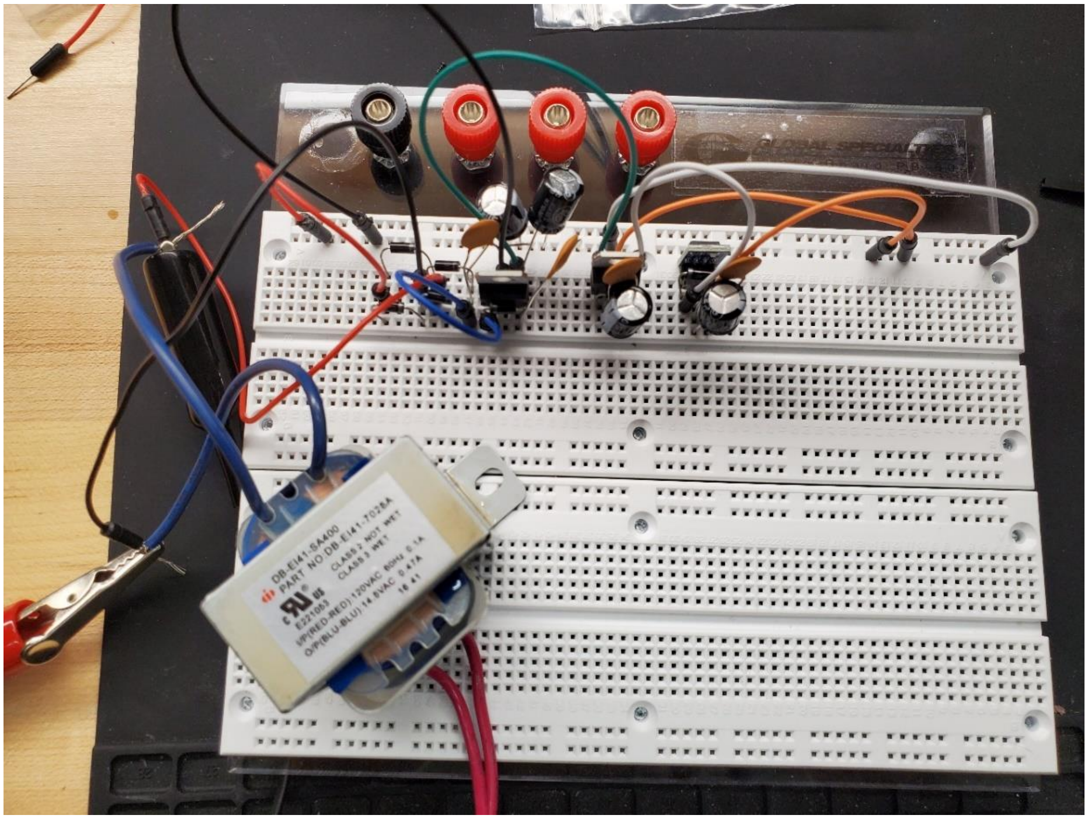


Figure 3-11 Initial Power Delivery System Mockup (Scrapped)

We later found out from our professor that the LM78XX parts are extremely inefficient in voltage conversion. The efficiency rating of the 78XX parts is only around 50%. If we have all the energy in the world, such low efficiency would not be a problem. However, the world’s electrical energy is limited, so we do not have the leisure to waste energy. The following design’s efficiency is 80%+, which is a huge up lift from the 50% efficiency. The new design consists of 3 stages. In the first stage, the 120Vac will be converted to DC voltage with the 4 diodes rectifier just like the old design. Then, the converted current will run through an array of resisters, diodes, and capacitors to get to the UCC28742 chips. Here, the voltage will be step down and output as 12Vdc. In the second stage, the 12V output will be taken in by the LM25085A regulator to be converted to 5V. Parallel to the second stage is the third stage. In the third stage, we will run the 12V output voltage from the first stage into the TPS563249 regulator to convert it into 3.3V. The use of each of the output voltage is the same as mentioned before. The only difference between the old set up and the new one is that the new one is more stable and more efficient. The below figures show the schematic of the 3 stages obtained using TI’s Webench power designer tool as well as the BOM of the design.

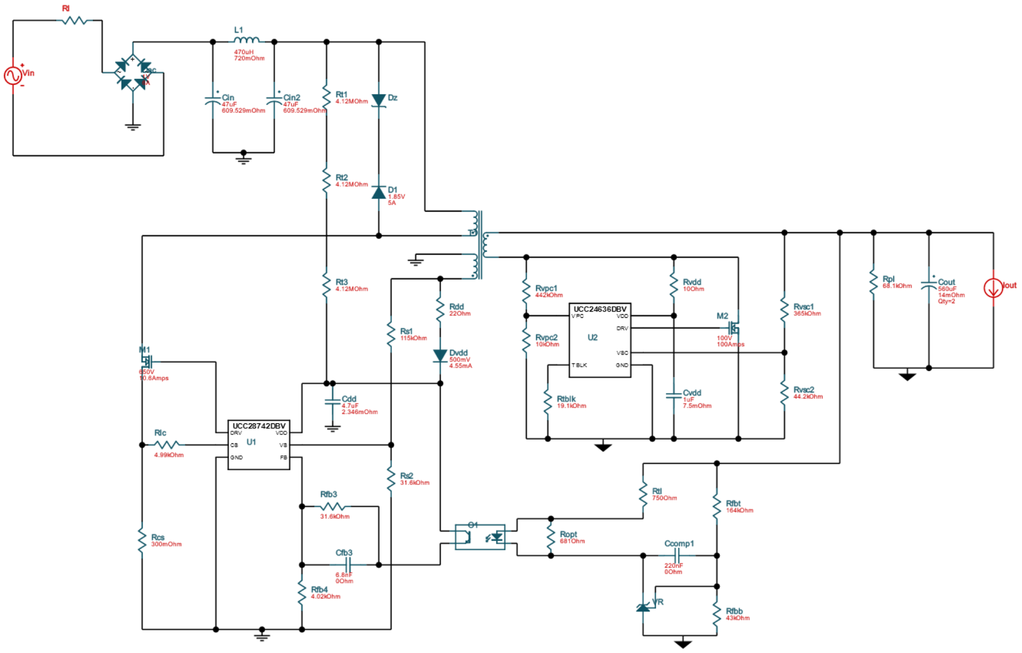


Figure 3-12 Final Power Delivery System Stage 1: 120Vac to 12Vdc circuit

Stage 1 includes quite a lot of components, which makes the mounting process extremely demanding. To simplify the process, we will use the conventional 12Vdc power brick to replace the stage 1 circuit entirely. This will give more space on the PCB for the microcontroller and the components of other stages. And since the transformer is taken out of the question, the PCB footprint will be smaller as well, giving us more space in the robot's body to do other things.

Figure 3-13 Conventional 12Vdc Power Brick

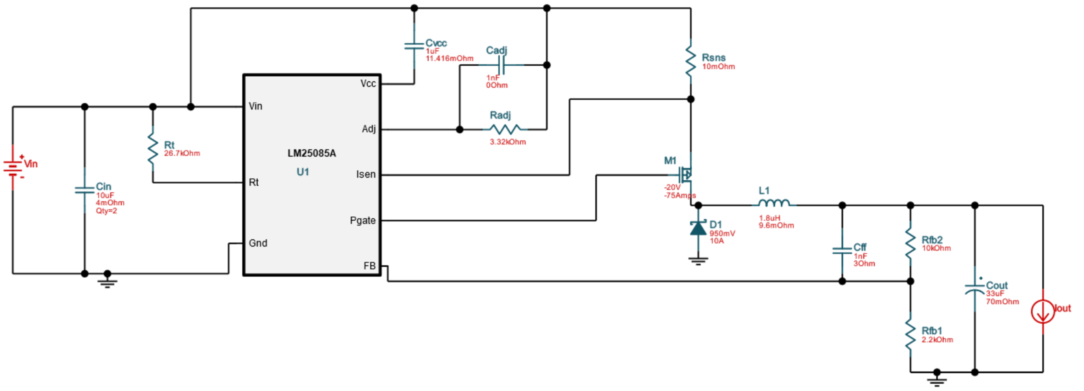


Figure 3-14 Final Power Delivery System Stage 2: 12Vdc to 5Vdc circuit

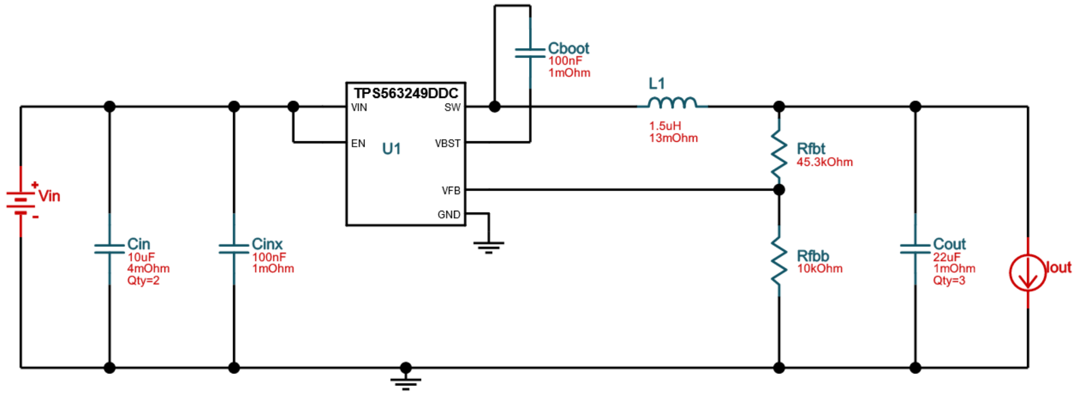


Figure 3-15 Final Power Delivery System Stage 3: 12Vdc to 3.3Vdc circuit

The bill of material (BOM) is presented in the table below:

Table 3-4 Power Deliver BOM

Part	Manufacturer	Part Number	Quantity	Price ($)	Footprint (mm²)	Description
T1	Core=TDK CoilFormer=TDK	Core=B65847A0000R087 CoilFormer=B65848D1010D001	1	3.2	714.24	Core Type: EP20 Core Material: N87 Turns Ratio(Nps): 35:4

Part	**Manufacturer**	**Part Number**	**Quantity**	**Price ($)**	**Footprint (mm^2)**	**Description**
Cvdd	TDK	C3216X7R2A105M160AA	1	0.12	10.92	Cap: 1 μF Total Derated Cap: 1 μF VDC: 100 V ESR: 7.5 mΩ Package: 1206
Rtl	Vishay-Dale	CRCW0402750RFKED	1	0.01	3	Resistance: 750 Ω Tolerance: 1.0% Power: 63 mW
Rvsc2	Vishay-Dale	CRCW020144K2FNED	1	0.01	2.08	Resistance: 44.2 kΩ Tolerance: 1.0% Power: 50 mW
Rlc	Vishay-Dale	CRCW04024K99FKED	1	0.01	3	Resistance: 4.99 kΩ Tolerance: 1.0% Power: 63 mW
O1	California Eastern Laboratories	PS2811-1	1	0.406	110.7	CTR_Min: 100.0% CTR_Max: 400.0% IcMax: 40 mmA
D1	STMicroelectronics	STTH506B-TR	1	0.63	101.71	Type: Ultrafast VRRM: 600 V Io: 5 A
Rs1	Vishay-Dale	CRCW0402115KFKED	1	0.01	3	Resistance: 115 kΩ Tolerance: 1.0% Power: 63 mW
Dac	Fairchild Semiconductor	GBU6J	1	0.43	130.59	Type: Switching-Bridge VRRM: 600 V Io: 6 A
Rt1	Vishay-Dale	CRCW08054M12FKEA	1	0.01	6.75	Resistance: 4.12 MΩ Tolerance: 1.0% Power: 125 mW
Cfb3	MuRata	GRM188R72A682KA01D	1	0.01	4.68	Cap: 6.8 nF Total Derated Cap: 6.8 nF VDC: 100 V ESR: 400 mΩ Package: 0603
Rfb4	Vishay-Dale	CRCW04024K02FKED	1	0.01	3	Resistance: 4.02 kΩ Tolerance: 1.0% Power: 63 mW
Rvpc1	Yageo	RC0201FR-07442KL	1	0.01	2.08	Resistance: 442 kΩ Tolerance: 1.0% Power: 50 mW
Dz	Diodes Inc.	SMBJ120A-13-F	1	0.09	44.16	Type: SMT VRWM: 120 V

Part	Manufacturer	Part Number	Quantity	Price ($)	Footprint (mm²)	Description
Rt2	Vishay-Dale	CRCW08054M12FKEA	1	0.01	6.75	Resistance: 4.12 MΩ Tolerance: 1.0% Power: 125 mW
Rvsc1	Yageo	RC0201FR-07365KL	1	0.01	2.08	Resistance: 365 kΩ Tolerance: 1.0% Power: 50 mW
M2	Texas Instruments	CSD19531Q5A	1	0.58	55.2	VdsMax: 100 V IdsMax: 100 Amps
Cdd	TDK	C2012X7R1V475K125AC	1	0.18	6.75	Cap: 4.7 µF Total Derated Cap: 4.7 µF VDC: 35 V ESR: 2.35 mΩ Package: 0805
Rvpc2	Yageo	RC0201FR-0710KL	1	0.01	2.08	Resistance: 10 kΩ Tolerance: 1.0% Power: 50 mW
Rfbb	Yageo	RC0603FR-0743KL	1	0.01	4.68	Resistance: 43 kΩ Tolerance: 1.0% Power: 100 mW
Rfb3	Vishay-Dale	CRCW040231K6FKED	1	0.01	3	Resistance: 31.6 kΩ Tolerance: 1.0% Power: 63 mW
Rdd	Yageo	RC0603FR-0722RL	1	0.01	4.68	Resistance: 22 Ω Tolerance: 1.0% Power: 100 mW
Rvdd	Vishay-Dale	CRCW040210R0FKED	1	0.01	3	Resistance: 10 Ω Tolerance: 1.0% Power: 63 mW
L1	Bourns	SDR1307-471KL	1	0.42	226.5	L: 470 µH DCR: 720 mΩ IDC: 900 mA
Rfbt	Yageo	RT0805BRD07164KL	1	0.06	6.75	Resistance: 164 kΩ Tolerance: 0.1% Power: 125 mW
Rcs	Stackpole Electronics Inc	CSRN2512FKR300	1	0.13	43.16	Resistance: 300 mΩ Tolerance: 1.0% Power: 2 W
U1	Texas Instruments	UCC28742DBVR	1	0.22	9.61	
VR	Texas Instruments	TL431IDBVR	1	0.061	16.39	IkMin: 400 mA InitialAccuracy: 1.5% ReferenceOutput: 2.5 V

Part	**Manufacturer**	**Part Number**	**Quantity**	**Price ($)**	**Footprint (mm²)**	**Description**
Cin	Panasonic	EEUED2G470S	1	0.73	400	Cap: 47 µF Total Derated Cap: 47 µF VDC: 400 V ESR: 609.53 mΩ Package: 18x20
Ccomp1	MuRata	GRM188R71E224KA88D	1	0.03	4.68	Cap: 220 nF Total Derated Cap: 190 nF VDC: 25 V ESR: 220 mΩ Package: 0603
Dvdd	CUSTOM	CUSTOM	1	NA	NA	Type: ? VRRM: 132.48 V Io: 4.55 mA
Rpl	Yageo	RC0201FR-7D68K1L	1	0.01	2.08	Resistance: 68.1 kΩ Tolerance: 1.0% Power: 50 mW
U2	Texas Instruments	UCC24636DBVR	1	0.352	14.82	
Rs2	Vishay-Dale	CRCW040231K6FKED	1	0.01	3	Resistance: 31.6 kΩ Tolerance: 1.0% Power: 63 mW
Rt3	Vishay-Dale	CRCW08054M12FKEA	1	0.01	6.75	Resistance: 4.12 MΩ Tolerance: 1.0% Power: 125 mW
Rtblk	Vishay-Dale	CRCW040219K1FKED	1	0.01	3	Resistance: 19.1 kΩ Tolerance: 1.0% Power: 63 mW
Cout	Panasonic	16SVPF560M	2	0.63	106.09	Cap: 560 µF Total Derated Cap: 1.1 mF VDC: 16 V ESR: 14 mΩ Package: 8x11.9
M1	Infineon Technologies	IPD65R380C6	1	0.78	101.71	VdsMax: 650 V IdsMax: 10.6 Amps
Cin2	Panasonic	EEUED2G470S	1	0.73	400	Cap: 47 µF Total Derated Cap: 47 µF VDC: 400 V ESR: 609.53 mΩ Package: 18x20
Ropt	Vishay-Dale	CRCW0402681RFKED	1	0.01	3	Resistance: 681 Ω Tolerance: 1.0% Power: 63 mW

Part	**Manufacturer**	**Part Number**	**Quantity**	**Price ($)**	**Footprint (mm²)**	**Description**
Cadj	MuRata	GRM1555C1H102JA01J	1	0.01	3	Cap: 1 nF Total Derated Cap: 1 nF VDC: 50 V ESR: 0 Ω Package: 0402
Cin	MuRata	GRM21BR61E106MA73L	2	0.05	6.75	Cap: 10 µF Total Derated Cap: 3 µF VDC: 25 V ESR: 4 mΩ Package: 0805
Cout	Panasonic	10TPB33M	1	0.31	17.1	Cap: 33 µF Total Derated Cap: 33 µF VDC: 10 V ESR: 70 mΩ Package: 3528-20
Cvcc	TDK	C1005X6S1C105K050BC	1	0.02	3	Cap: 1 µF Total Derated Cap: 1 µF VDC: 16 V ESR: 11.42 mΩ Package: 0402
D1	SMC Diode Solutions	SBRD10200TR	1	0.12	101.71	Type: Schottky VRRM: 200 V Io: 10 A
L1	Coilcraft	XAL6030-182MEB	1	0.65	71.56	L: 1.8 µH DCR: 9.6 mΩ IDC: 14 A
M1	Texas Instruments	CSD25402Q3A	1	0.26	18.49	VdsMax: -20 V IdsMax: -75 Amps
Radj	Vishay-Dale	CRCW04023K32FKED	1	0.01	3	Resistance: 3.32 kΩ Tolerance: 1.0% Power: 63 mW
Rfb1	Yageo	RC0603FR-072K2L	1	0.01	4.68	Resistance: 2.2 kΩ Tolerance: 1.0% Power: 100 mW
Rfb2	Vishay-Dale	CRCW040210K0FKED	1	0.01	3	Resistance: 10 kΩ Tolerance: 1.0% Power: 63 mW
Rsns	Stackpole Electronics Inc	CSR1206FK10L0	1	0.12	10.92	Resistance: 10 mΩ Tolerance: 1.0% Power: 500 mW
Rt	Vishay-Dale	CRCW040226K7FKED	1	0.01	3	Resistance: 26.7 kΩ Tolerance: 1.0% Power: 63 mW
U1	Texas Instruments	LM25085AMY/NOPB	1	0.66	23.6	

Part	**Manufacturer**	**Part Number**	**Quantity**	**Price ($)**	**Footprint (mm^2)**	**Description**
Cff	MuRata	GRM188R71E102KA01D	1	0.01	4.68	Cap: 1 nF Total Derated Cap: 1 nF VDC: 25 V ESR: 3 Ω Package: 0603
U1	Texas Instruments	TPS563249DDCR	1	0.25	10.47	
L1	TDK	VLP8040T-1R5N	1	0.22	113.42	L: 1.5 µH DCR: 13 mΩ IDC: 7.8 A
Cin	MuRata	GRM21BR61E106MA73L	2	0.05	6.75	Cap: 10 µF Total Derated Cap: 3 µF VDC: 25 V ESR: 4 mΩ Package: 0805
Cboot	Taiyo Yuden	EMK107B7104KA-T	1	0.01	4.68	Cap: 100 nF Total Derated Cap: 100 nF VDC: 16 V ESR: 1 mΩ Package: 0603
Cout	MuRata	GRM188R60J226MEA0D	3	0.05	4.68	Cap: 22 µF Total Derated Cap: 24 µF VDC: 6.3 V ESR: 1 mΩ Package: 0603
Cinx	Yageo	CC0805KRX7R9BB104	1	0.02	6.75	Cap: 100 nF Total Derated Cap: 99 nF VDC: 50 V ESR: 1 mΩ Package: 0805
Rfbt	Vishay-Dale	CRCW040245K3FKED	1	0.01	3	Resistance: 45.3 kΩ Tolerance: 1.0% Power: 63 mW
Rfbb	Vishay-Dale	CRCW040210K0FKED	1	0.01	3	Resistance: 10 kΩ Tolerance: 1.0% Power: 63 mW

We are quite tight on the budget side, so to save on cost, we will order the majority of parts from the BOM from eBay. We will also make use of parts that are available in the Senior Design lab. For major components like regulators, we will order them directly from the manufacturer’s website.

3.5.7 PCB Design

The obvious step immediately after designing the multi-rail power delivery system is to design the PCB. We will start designing the PCB immediately after the finish setting up and testing the new power delivery design. We expect to have the PCB’s design sent to the manufacturer by the end of January. Because of the complexity and the small size of traces, they are typically manufactured by a specialized vendor. To utilize a PCB manufacturing service, we will need to first design the PCB and obtain and Gerber file to send to manufacture. Gerber file will contain all the information that a manufacture may need to produce the board like the copper layers, solder mask, legend, and distance between traces, etc. There are many free and paid options that can be used to design a PCB. Due to our inexperience and lack of expertise, we will be using Autodesk Eagle to design our PCB. We chose Eagle because it is the software that we have some experience with, and it is also that the software that is widely used by our professors at the University of Central Florida.

PCB could be very simple, or it could be very complex. The complexity of a PCB is highly dependent on the number of layers, number of components, and the number of traces on a single board. As expected, the higher the number of traces on the board, the higher the cost will be. This is because the high number of traces require extreme precision. If the trace laying is not precise, the traces could cross and create a short in the system. Some advice we received from people who have already graduated is that to cut cost on a circuit with many components, it may be smart to split the circuit into different PCBs with each PCB handles a separate stage of the circuit. We will not really know the size of the PCB until we have finished prototyping the circuit, so we will not be able to decide the number of PCB we will need.

There are many PCB manufactures that we can use, from those that are based in the US to those that are based in Taiwan. To save on cost, the obvious option is to go with a Taiwan manufacture. However, due to the current pandemic, overseas travel and overseas shipping are restricted. If we order from Taiwan, we will need to wait 2-3 months before we can get the board. By the time we get the PCB, we may have already passed the deadline for the project. The safer option for now is to go with a domestic manufacture. In the United States, two of the more popular PCB manufactures are JLCPCB and OSHPARK. From a quick scan, JLCPCB seems to be cheaper for smaller boards. Though as the board gets larger, OSHPARK's pricing makes more sense. For a 25 square inch board, OSHPARK charges $42 per board, and you must buy 3 boards at a time. JLCPCB charges $10 per board, with no order limit. Currently, JLCPCB is offering free soldering with their boards, though we do not know if this offer will last until we finish our PCB design. The current JLCPCB price is the discounted holiday price, so we do not know who is more expensive until the promotion is over. By the time we finish designing our PCB, we will use whichever is the cheapest between the two options.

At the end of the day, we have decided to go with JLCPCB for the manufacturing of our PCB and with Quality Manufacturing Services at Lake Mary for the assembly of our PCB. Initially, we had wanted to go with PCBWAY simply because our quote there was cheaper. However, after almost 2 months of delay, the board came in dead-on-arrival. We had to reorder on JLCPCB. Our experience with JLCPCB was very nice. Their staffs were really responsive to questions and they even gave us a small discount on our order. The board came in quickly (1 week after ordering), even though the company is located in mainland China. The International shipping was very expensive though. The below images show our final PCB, as well as its schematic and board layout from Eagle.

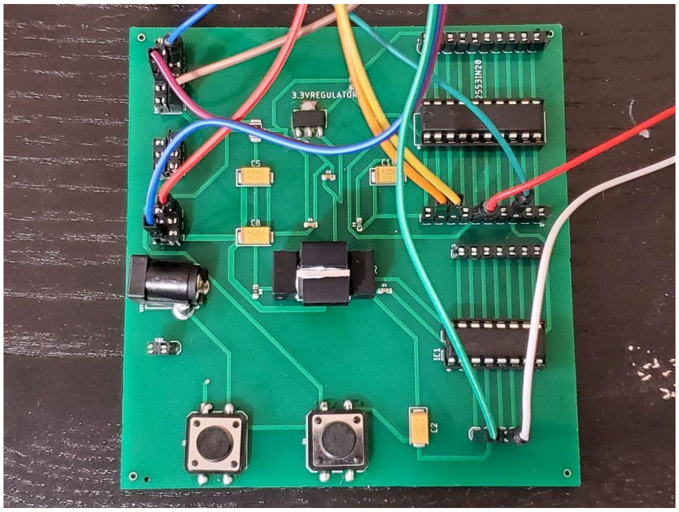


Figure 3-16 Final PCB

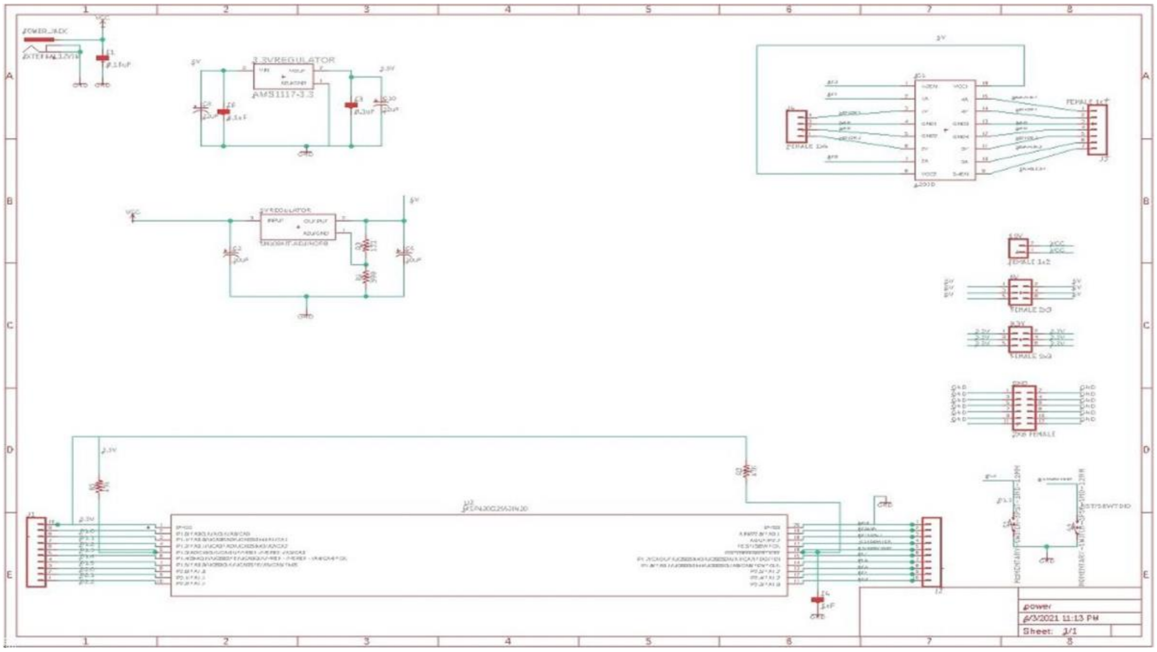


Figure 3-17 Final PCB Schematic

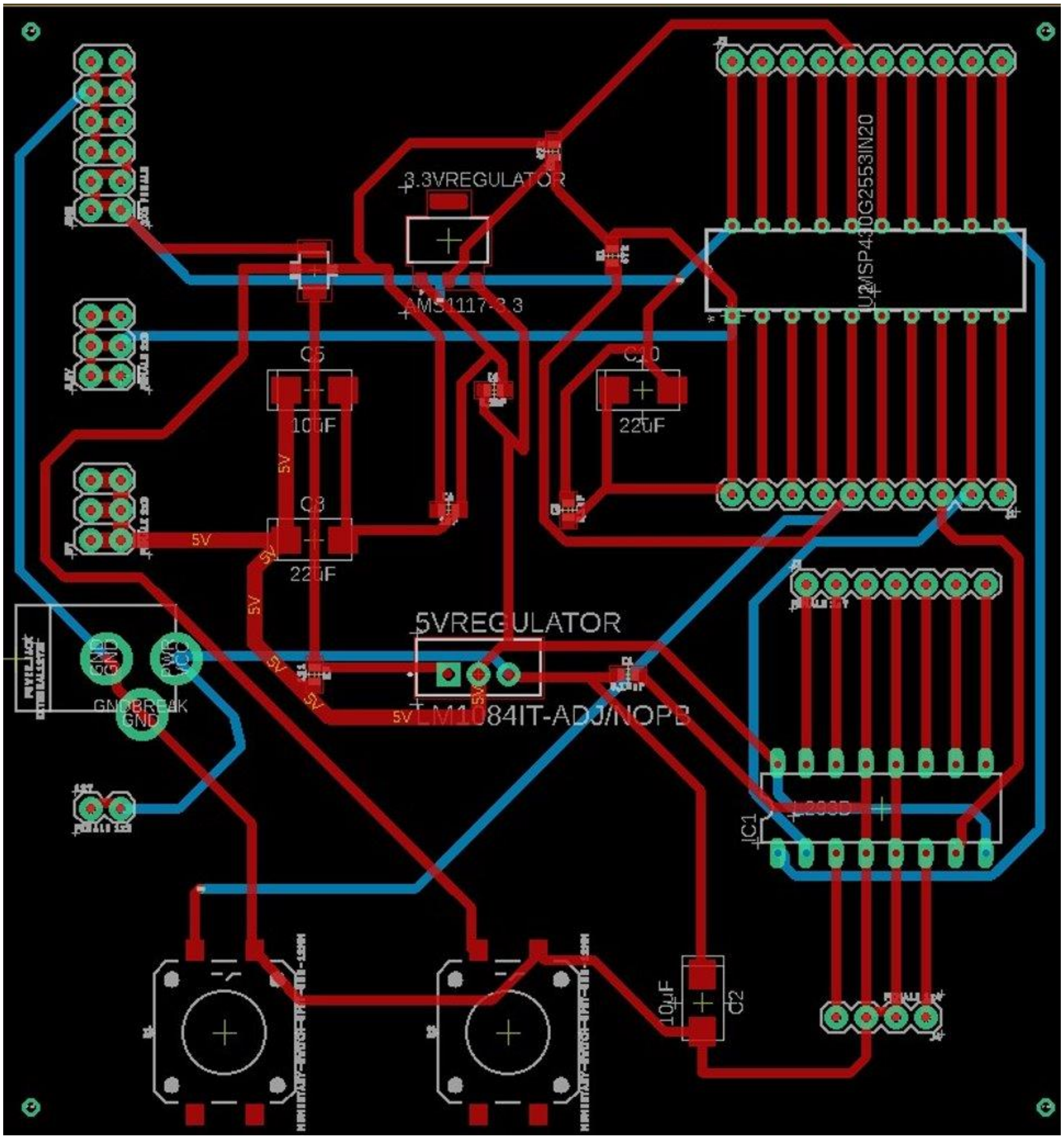


Figure 3-18 Final PCB Board Layout

3.6 Software Design

The below subsections detail our design of the software components of the robot. For obvious reasons, our project cannot be solely hardware-based. We need to have a seamless combination between the hardware and software to make the user's experience as stress-free as possible. Aside from listing the software components, we will also go over the purpose of the part in the grand scheme of functionality of the system.

3.6.1 Jetson and MCU Communication Protocol

Based on the requirements we must have a communication protocol between our two main data processors for successful operation. There are two things necessary in this design that require correct information exchange. The simplest solution then is to have one GPIO pin from the MCU to the Nano, and vice-versa in order to have what is essentially a Tx/Rx interface on both ends. Although this is possible, in testing we found that UART is a much simpler standard to follow by both devices and ultimately chose to go with a UART interface for Jetson Tx, MCU Rx, and a GPIO interface for the other way around. This is explained further below.

When beginning integration we first used one full duplex UART connection, though ran into problems with the MCU receiving data over the serial port. Regardless of how much we tried to fix the problem we couldn't find a solution, so we ended up using a GPIO connection to signal from the Jetson to the MCU to meet our requirements, which worked just as well. A UART connection was kept from the MCU to the Jetson.

3.6.1.1 Clock Domain Crossing

Since the two units will be running off of completely separate clocks with no phase relation at almost certainly different operating frequencies, we must devise a way to cross these clock domains to/from the MCU/Jetson. Luckily when using UART this isn't an issue since it is meant to be asynchronous on both ends. Since our Jetson Tx, MCU Rx is a GPIO connection, we kind of worry about this. Thankfully though, we don't care whether the pin is metastable since once it is high we run the ISR on the MCU, and wait for ~15 seconds until polling again. This is plenty of time for the data to settle.

3.6.2 Jetson Nano

Training a machine learning model is a complex process. While it is possible to incorporate every step of the process into one big part, doing so would make the troubleshooting and debugging process a living nightmare. To make our lives easier, we have decided to split the model training portion of our project into multiple small parts. The sub-subsections below detail the process that we go through to train our model.

3.6.2.1 NVIDIA Transfer Learning Toolkit

"Transfer Learning" is the method of machine learning which uses the knowledge in an already trained model and applies it onto a different, but related problem. The idea of transfer learning is reusing the model developed for a certain task as the initial model for a related task which does not have sufficient data. Instead of starting the process by collecting data, the users can begin with the model that "solves" a related task. Transfer learning has become popular in computer vision because of the requirements for large amount of computational power when training a network from scratch. Transfer learning is extremely beneficial in different ways such as saving a huge amount of training time (it normally takes days to weeks for a complex task) due to the fact that it uses a previously trained and specialized model.

Transfer learning is surprisingly useful when we do not have enough data for a new domain, while there is a huge pre-existing data pool that can be reused in the new problem. The NVIDIA Transfer Learning Tool Kit provides GPU accelerated pre-trained models and functions that we can make little adjustments to in order to achieve the desired performance in mask detection. NVIDIA TLT not only helps minimize the AI training time but also provides an efficient and accurate AI model that we are using for this project by transferring the NVIDIA pre-trained model to our mask detection model. Also, TLT does not require high-level understanding in deep learning, so it is easy to use that make our own custom model for our project.

Figure 13 shows the NVIDIA TLT process. Using TLT itself consists of choosing a pre-trained model and architecture relevant to our requirements, finding a viable dataset in order to re-train the model, run training, prune the model, and then finally implement it and run inference at the end point on the Jetson Nano. The model we use for TLT face mask detection will be the DetectNet_v2 with ResNet18 architecture due to its relative simplicity (vs. more complex and deeper backbones such as ResNet-34, 50, DarkNet, etc.). The model will be re-trained for face mask detection, pruned, then deployed. Figure 2-1 shows the general flow of training a model, and then deploying the actual model using Deepstream SDK.

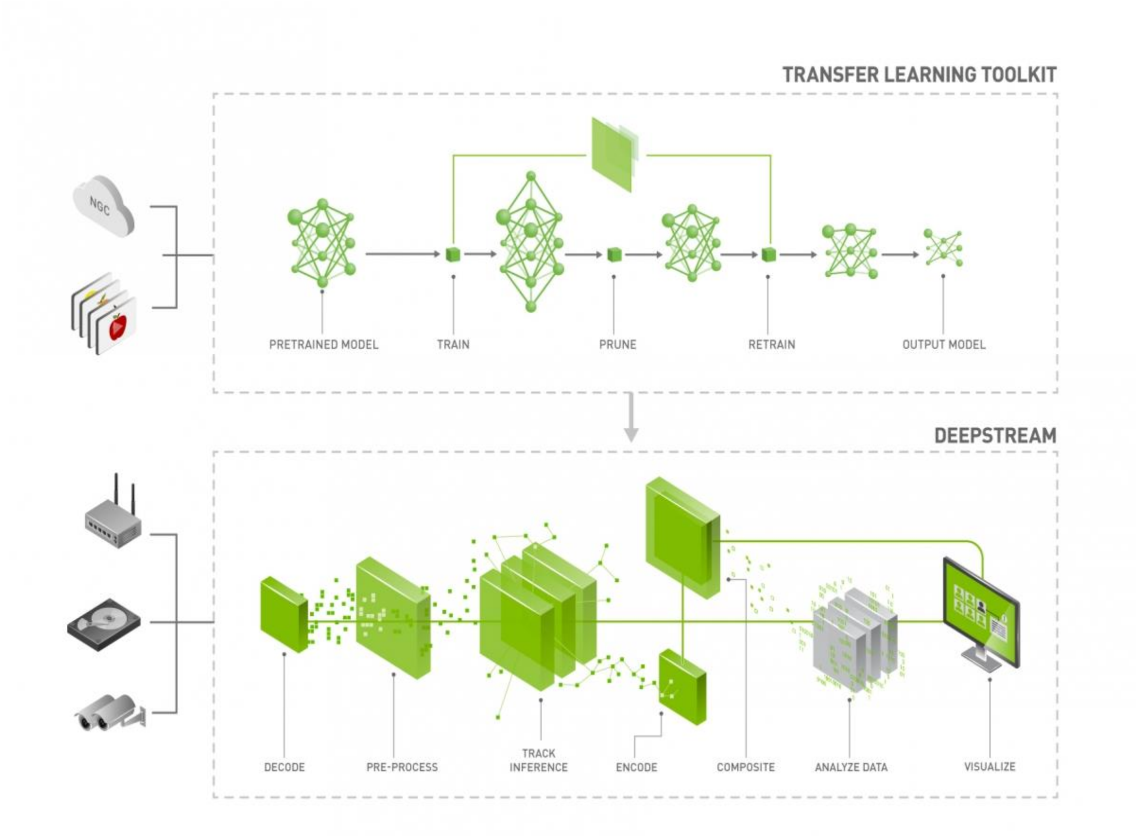
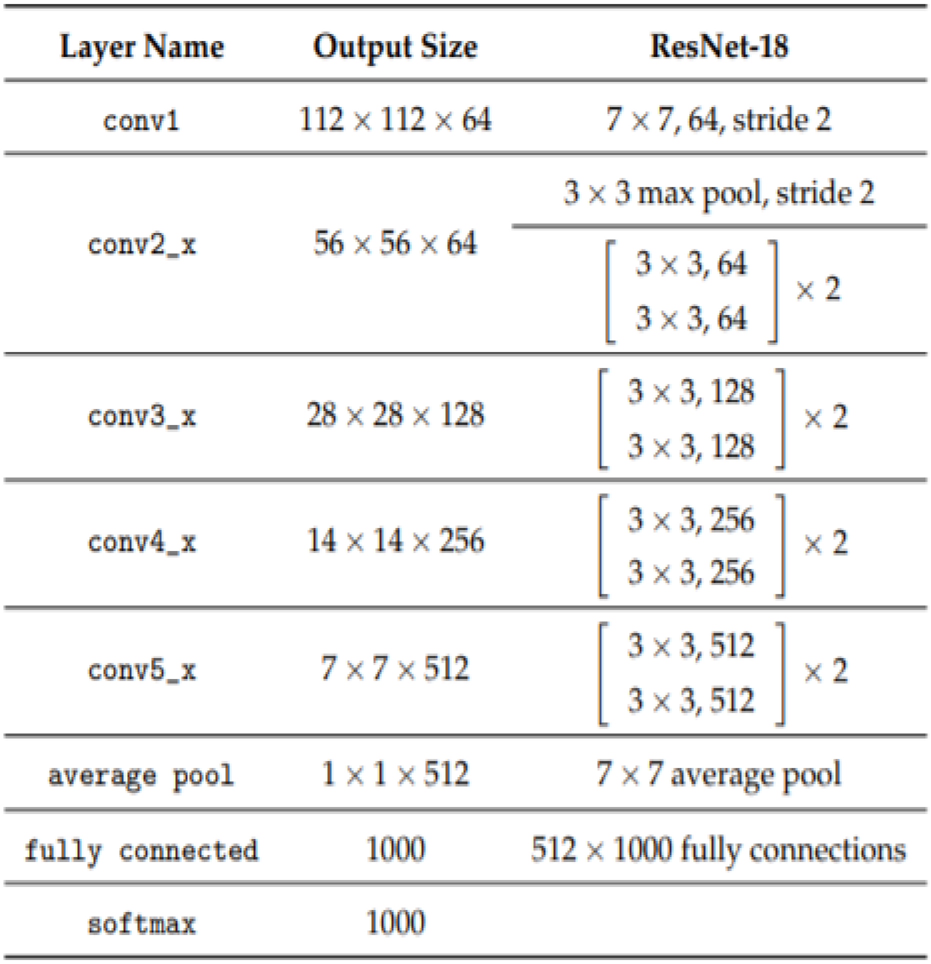


Figure 3-19 Transfer Learning Toolkit

ResNet-18

ResNet has become one of the most powerful architecture for deep neural network that achieve excellent performance in recognition tasks after winning the first place in ILSVRC 2015 classification competition.



Layer Name	Output Size	ResNet-18
conv1	$112 \times 112 \times 64$	7×7, 64, stride 2
conv2_x	$56 \times 56 \times 64$	3×3 max pool, stride 2
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$
conv3_x	$28 \times 28 \times 128$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$
conv4_x	$14 \times 14 \times 256$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$
conv5_x	$7 \times 7 \times 512$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$
average pool	$1 \times 1 \times 512$	7×7 average pool
fully connected	1000	512×1000 fully connections
softmax	1000	

Figure 3-20 ResNet 18 Architecture

ResNet-18 is a convolutional neural network that uses 18 layers that has been widely used in many computer applications including face recognition because of its powerful ability. The problem in the previous network architectures is that with the complex problems, feedforward network could be massive that lead to overfitting the data; therefore, the convolutional neural network architecture has to be deeper. However, deep network is the serious problem for vanishing gradient, which means the gradients of the loss function become infinitively small (close to zero), making the network hard to train, and it suffers from the degradation problem. The primary concept of ResNet is using shortcut connections which skips layers in order to obtain better performance compared to plain networks and handle the degradation issue. The following table will demonstrate more details about the ResNet-18 architecture.

Assume the image input size is 224x224, this implementation will be based on the original paper “Deep Residual Learning for Image Recognition”:

Table 3-5 Implementation of ResNet-18

Layer Name		**Filter Size**	**Stride**	**Padding**	**Output layer**	**Output size**
Input image						224*224
Conv1	Conv	7*7	2	3	64	112*112*64
	BN					
	MaxPooling	3*3	2	1		56*56*64
Conv2	Conv	3*3	1	1	64	56*56*64
	Conv	3*3	1	1	64	
	Conv	3*3	1	1	64	56*56*64
	Conv	3*3	1	1	64	
Conv3	Conv	3*3	2	1	128	28*28*128
	Conv	3*3	1	1	128	
	Conv	3*3	1	1	128	28*28*128
	Conv	3*3	1	1	128	
Conv4	Conv	3*3	2	1	256	14*14*256
	Conv	3*3	1	1	256	
	Conv	3*3	1	1	256	14*14*256
	Conv	3*3	1	1	256	
Conv5	Conv	3*3	2	1	512	7*7*512
	Conv	3*3	1	1	512	
	Conv	3*3	1	1	512	7*7*512
	Conv	3*3	1	1	512	
	AvgPooling	7*7		0		1*1*512
FC	FC	512*1000				1000
	Softmax					1000

3.6.2.1.1 Training The Model

Alongside the abundant support NVIDIA has given the community, an open-source walkthrough of face mask detection is supplied, with the link available in the references section. This Github comprises the TLT process from the very beginning of choosing a dataset up until the deployment in DeepStream SDK, with detailed walkthroughs. This will be the process followed for creating an AI algorithm to detect users with/without face masks.

The pre-requisites for following these steps in the Github repository are that we need a current up to date Linux build, and an NVIDIA GPU in order to actually train the model. For the Linux distribution, we will use Ubuntu 20.04 running in WSL2 inside of Windows 10. The correct CUDA drivers must be installed for the GPU to be recognized inside of WSL2 and subsequently the underlying Docker service later on for training, pruning, and re-training.

The Github provides a link to a Dockerfile that will be ran to save time of setting up the environment ourselves. All of the dependencies and pre-requisites are provided by the Dockerfile. One important thing to note is we need to mount a volume for the Docker image to our Ubuntu directory such that we can provide the datasets for training and export the model once we are done. Once this Dockerfile is ran with the mounted volume, we clone the face-mask-detection Github repository and continue following the provided steps.

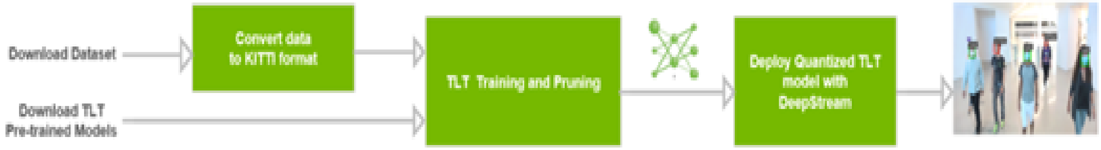

Download Dataset
Download TLT Pre-trained Models
Convert data to KITTI format
TLT Training and Pruning
Deploy Quantized TLT model with DeepStream

Figure 3-21 Workflow of TLT and DeepStream SDK

This figure above shows the general idea of the implementation steps behind deploying our model for real-time face mask detection. First, we download the dataset of faces with mask and without mask and convert them into KITTI format using the scripts given, and then into TFRecords format for the underlying TensorFlow API that TLT runs on top of. We also download the pretrained model from NGC (NVIDIA GPU Cloud). Specifically, we will use the DetectNet_v2 with the ResNet-18 for accuracy and efficiency. After that, the quantized TLT deep learning model is deployed with DeepStream SDK to identify faces with mask or no mask. By using NVIDIA TLT and NVIDIA DeepStream SDK, it saves a significant amount of time in training as well as allows us to achieve high accuracy performance in real-time.

3.6.2.1.2 Download Dataset

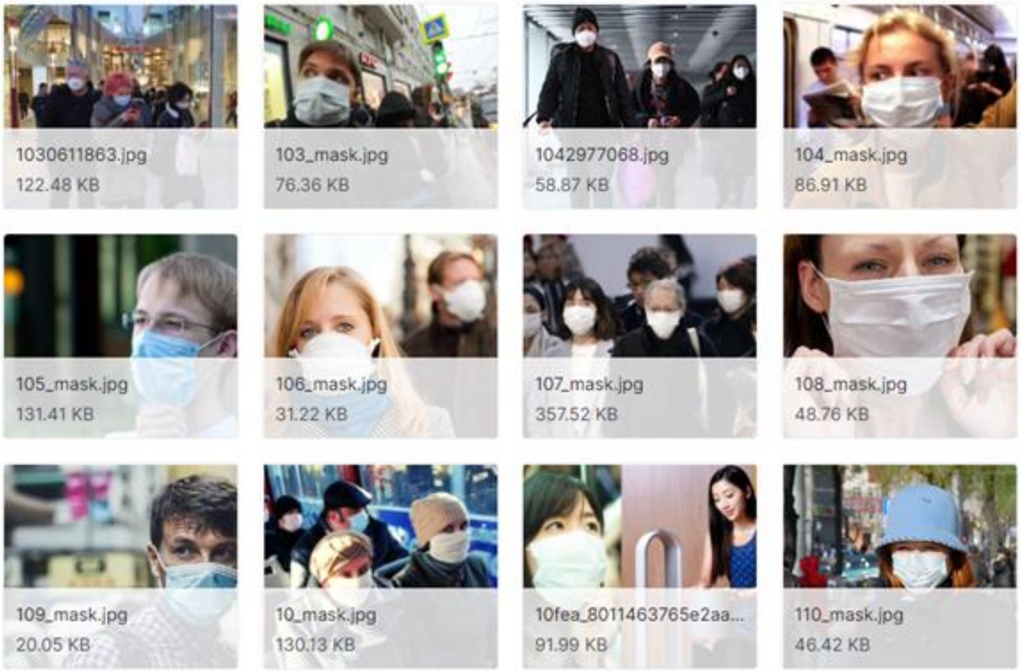


Figure 3-22 Mask Dataset examples

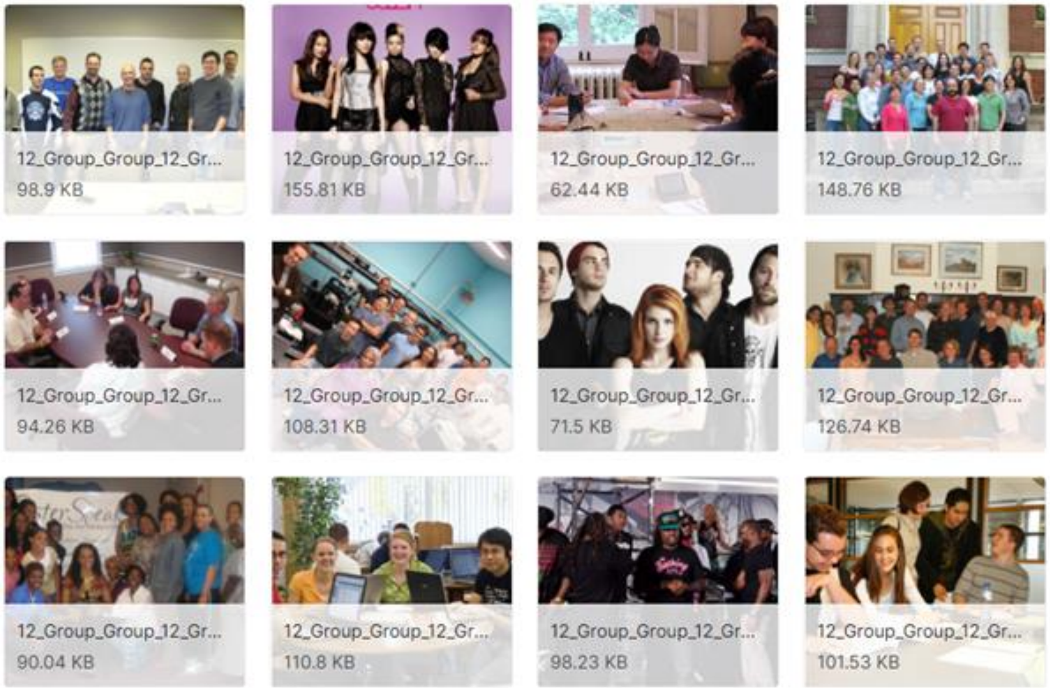
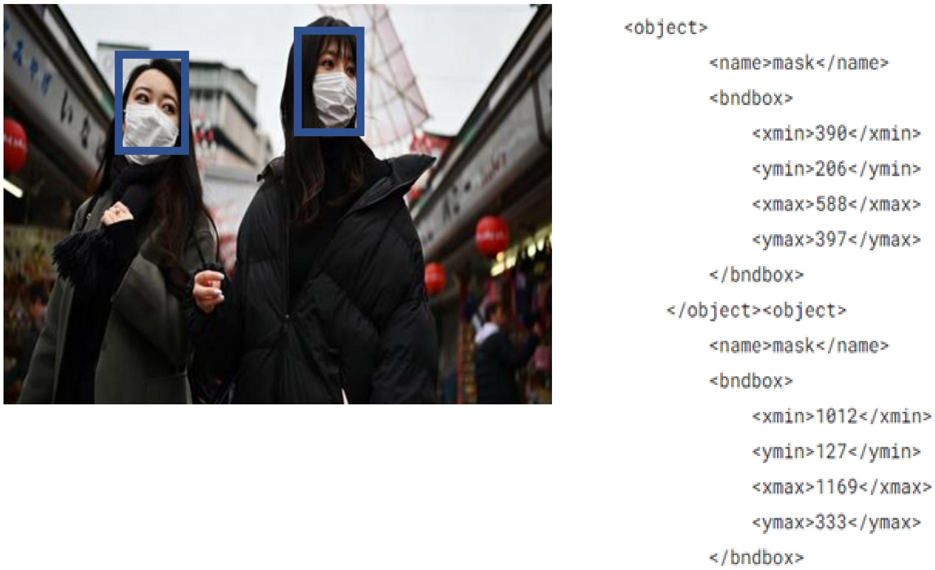


Figure 3-23 No mask Dataset examples



```
<object>
        <name>mask</name>
        <bndbox>
            <xmin>390</xmin>
            <ymin>206</ymin>
            <xmax>588</xmax>
            <ymax>397</ymax>
        </bndbox>
    </object><object>
        <name>mask</name>
        <bndbox>
            <xmin>1012</xmin>
            <ymin>127</ymin>
            <xmax>1169</xmax>
            <ymax>333</ymax>
        </bndbox>
```

Figure 3-24 Sample XML file

Data labelling is important step in supervised machine learning, which means the result of the model depends on the labels that we used in the initial dataset. There are several techniques for labeling the images, in this data set, we used bounding box as our image annotation since it is the most common type of labeling in computer vision. Bounding boxes are defined as rectangular boxes to determine the location of the faces. The bounding boxes use x and y axis coordinates from upper left corner to the right corner (x axis), and from lower right corner to upper corner (y axis).

XML files stores the details of the faces in individual images, each image will have its own XML file that include the image size, and the coordinates x and y of around the face with mask or without mask. For example, in this picture above, the size of original image is 1920 * 1277 * 3 (width * height * C), and there are two faces are detected within the bounding box in this image are listed with specific *xmin*, *ymin*, *xmax*, *ymax*, with “mask” label in the XML file.

Thankfully, labeled datasets are already provided by NVIDIA from the Kaggle Medical Mask, MAFA, FDDB, and WiderFace datasets to use for training.

3.6.2.1.3 Download TLT Pre-Trained Model

In order to use Transfer Learning Toolkit, there are several hardware and software requirements that need to be met by the PC running the actual training:

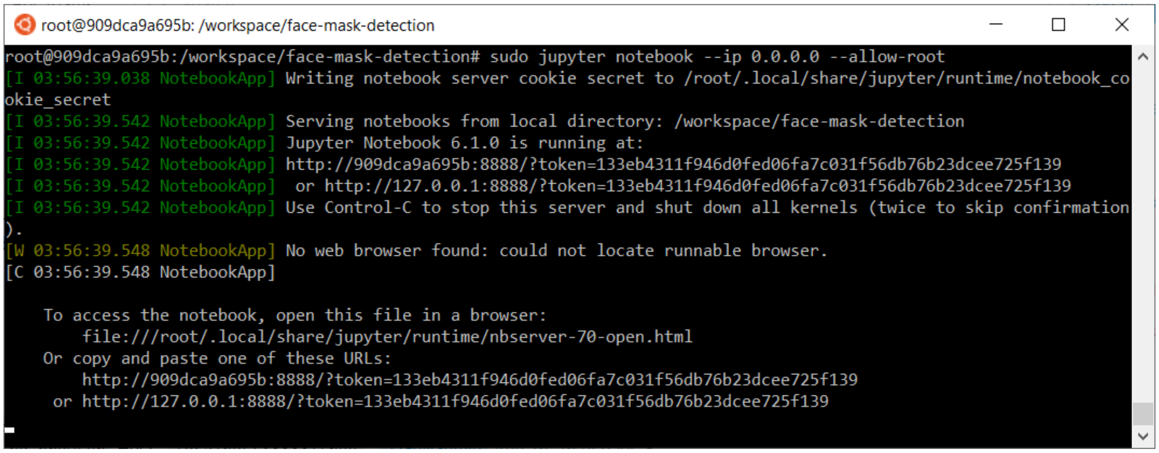
Table 3-6 Requirements Table

Hardware requirements	Software requirements
1 NVIDIA GPU >4GB of GPU RAM >4GB of system RAM 4 core CPU >50GB of HHD space	Ubuntu 18.04 LTS NVIDIA docker installed Docker installed NVIDIA GPU driver

3.6.2.1.4 Convert the dataset

For training we require our dataset to be ultimately converted into a binary TFRecords format that the TensorFlow API requires which TLT runs on. To do this we first convert our labeled datasets into KITTI format using the data2kitti.py script given by NVIDIA, along with providing it the path to the datasets and an output directory. Ultimately, we end up using 12,000 images – half with mask, half without, for an even split of training that eventually get resized to a resolution of 960x544. After converting to the KITTI dataset, we continue with our flow onto the Jupyter Notebook provided in the Github repo.

The Jupyter Notebook is set up inside of the Docker image from, and then accessed from Windows outside of WSL2 by using the WSL2 IP on port 8888 with the token given when Jupyter is ran. Jupyter notebook is seen running in the figure below. Once Jupyter is running in Docker, we follow the steps inside of the given notebook face-mask-detection.ipynb to convert our KITTI formatted images/labels into TFRecords, and continue on to training.



root@909dca9a695b: /workspace/face-mask-detection

```
root@909dca9a695b:/workspace/face-mask-detection# sudo jupyter notebook --ip 0.0.0.0 --allow-root
[I 03:56:39.038 NotebookApp] Writing notebook server cookie secret to /root/.local/share/jupyter/runtime/notebook_co
okie_secret
[I 03:56:39.542 NotebookApp] Serving notebooks from local directory: /workspace/face-mask-detection
[I 03:56:39.542 NotebookApp] Jupyter Notebook 6.1.0 is running at:
[I 03:56:39.542 NotebookApp] http://909dca9a695b:8888/?token=133eb4311f946d0fed06fa7c031f56db76b23dcee725f139
[I 03:56:39.542 NotebookApp]  or http://127.0.0.1:8888/?token=133eb4311f946d0fed06fa7c031f56db76b23dcee725f139
[I 03:56:39.542 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation
).
[W 03:56:39.548 NotebookApp] No web browser found: could not locate runnable browser.
[C 03:56:39.548 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///root/.local/share/jupyter/runtime/nbserver-70-open.html
    Or copy and paste one of these URLs:
        http://909dca9a695b:8888/?token=133eb4311f946d0fed06fa7c031f56db76b23dcee725f139
     or http://127.0.0.1:8888/?token=133eb4311f946d0fed06fa7c031f56db76b23dcee725f139
```

Figure 3-25 Jupyter Running in Docker

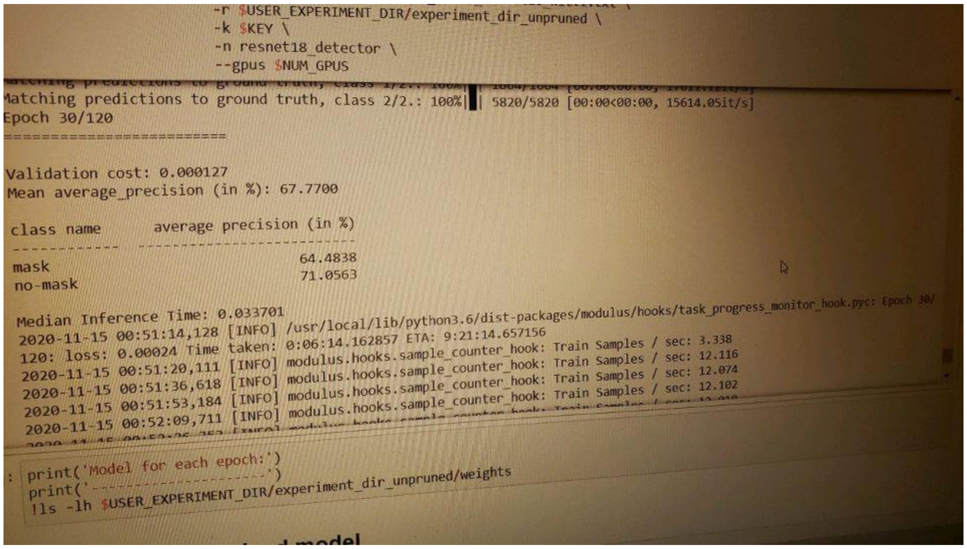
3.6.2.1.5 TLT Training

The main step in this software flow is the actual TLT training, done inside of the provided Jupyter Notebook. Once our labeled images are converted to TFRecords format and we have the pre-trained DetectNet_v2 model from NGC, we now can now continue to training.

The first step in training is re-configuring the detectnet_v2_train_resnet18_kitti.txt config file in the specs dir which controls many parameters for the training such as paths to files, validation split, and many others that have been pre-configured by NVIDIA. The ones that must be reconfigured are the image width, height, fold number (for validation split of images), and other training parameters such as number of epochs, batch size, etc. along with the paths to the training TFRecords.

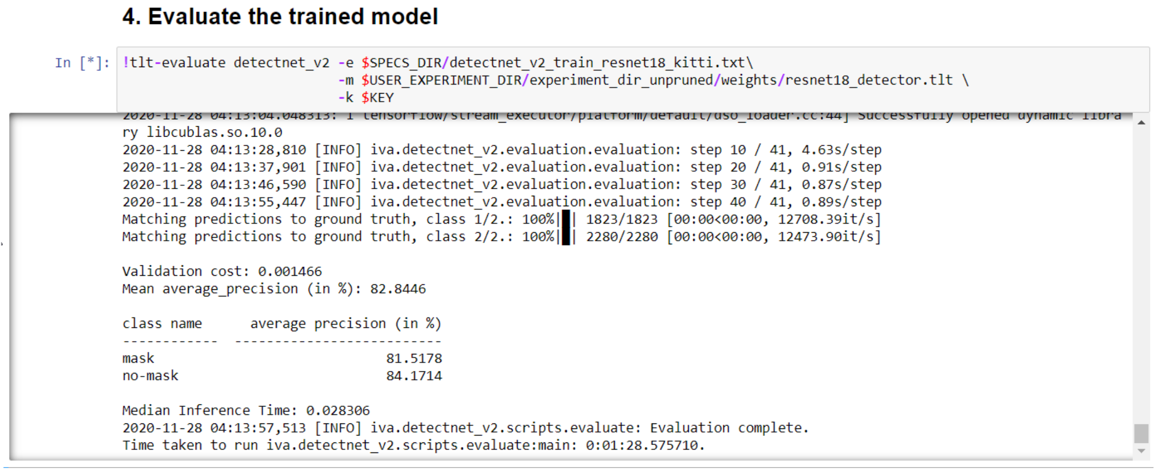
We set our parameters to 960x544, 120 epochs, and a batch size of 12. Depending on the amount of both system RAM and GPU DRAM is available we must lower the batch size parameter. Essentially the batch size defines how many samples we work through during the current epoch before we update the model weights/parameters. In other words, we go through all of the images in batches until we have looped over all the training images in the dataset, and only update our parameters at the end of each batch. A higher batch size implies better training for the model at the cost of taking more time and more memory since we are performing operations on a larger subset of data. In training, we found that a batch size of 12 was the most optimal setting – the highest we could go with acceptable training speed and not running into memory starvation.

After the parameters are set, we begin the training by using the tlt-train. A mid-training screenshot is shown in the figure below, along with the inference evaluation on our finished model using tlt-evaluate. We see that our final accuracy for the unpruned model is 82% and 84% for mask or no mask inference, respectively. This is close to NVIDIA's numbers of ~85% for each, though less likely due to a different split of training data, less epochs, smaller batch size, etc. Had we run training for more epochs on better hardware with larger batch size this accuracy would likely increase, although non-deterministically since training varies on every run.



```
                        -r $USER_EXPERIMENT_DIR/experiment_dir_unpruned \
                        -k $KEY \
                        -n resnet18_detector \
                        --gpus $NUM_GPUS
Matching predictions to ground truth, class 2/2.: 100%|█| 5820/5820 [00:00<00:00, 15614.05it/s]
Epoch 30/120
=========================

Validation cost: 0.000127
Mean average_precision (in %): 67.7700

class name      average precision (in %)
------------  --------------------------
mask                             64.4838
no-mask                          71.0563

Median Inference Time: 0.033701
2020-11-15 00:51:14,128 [INFO] /usr/local/lib/python3.6/dist-packages/modulus/hooks/task_progress_monitor_hook.pyc: Epoch 30/
120: loss: 0.00024 Time taken: 0:06:14.162857 ETA: 9:21:14.657156
2020-11-15 00:51:20,111 [INFO] modulus.hooks.sample_counter_hook: Train Samples / sec: 3.338
2020-11-15 00:51:36,618 [INFO] modulus.hooks.sample_counter_hook: Train Samples / sec: 12.116
2020-11-15 00:51:53,184 [INFO] modulus.hooks.sample_counter_hook: Train Samples / sec: 12.074
2020-11-15 00:52:09,711 [INFO] modulus.hooks.sample_counter_hook: Train Samples / sec: 12.102

: print('Model for each epoch:')
  print('---------------------')
  !ls -lh $USER_EXPERIMENT_DIR/experiment_dir_unpruned/weights
```

Figure 3-26 Model Mid-Training



4. Evaluate the trained model

```
In [*]: !tlt-evaluate detectnet_v2 -e $SPECS_DIR/detectnet_v2_train_resnet18_kitti.txt\
                                   -m $USER_EXPERIMENT_DIR/experiment_dir_unpruned/weights/resnet18_detector.tlt \
                                   -k $KEY
```

```
2020-11-28 04:13:04.048313: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic libra
ry libcublas.so.10.0
2020-11-28 04:13:28,810 [INFO] iva.detectnet_v2.evaluation.evaluation: step 10 / 41, 4.63s/step
2020-11-28 04:13:37,901 [INFO] iva.detectnet_v2.evaluation.evaluation: step 20 / 41, 0.91s/step
2020-11-28 04:13:46,590 [INFO] iva.detectnet_v2.evaluation.evaluation: step 30 / 41, 0.87s/step
2020-11-28 04:13:55,447 [INFO] iva.detectnet_v2.evaluation.evaluation: step 40 / 41, 0.89s/step
Matching predictions to ground truth, class 1/2.: 100%|█| 1823/1823 [00:00<00:00, 12708.39it/s]
Matching predictions to ground truth, class 2/2.: 100%|█| 2280/2280 [00:00<00:00, 12473.90it/s]

Validation cost: 0.001466
Mean average_precision (in %): 82.8446

class name      average precision (in %)
------------  --------------------------
mask                             81.5178
no-mask                          84.1714

Median Inference Time: 0.028306
2020-11-28 04:13:57,513 [INFO] iva.detectnet_v2.scripts.evaluate: Evaluation complete.
Time taken to run iva.detectnet_v2.scripts.evaluate:main: 0:01:28.575710.
```

Figure 3-27 Unpruned Model Inference Accuracy

3.6.2.1.6 TLT Pruning & Re-Train

The next stage in the TLT flow is the pruning stage, followed by re-training. From a high-level pruning is known as "Data compression" technique that takes nodes in the network that have a very small impact on the overall decision of the output and strips them away from the model. This method helps the model to achieve better solution, reduces the complexity of the model as well as improving the accuracy by decreasing overfitting the model. Also, this greatly reduces inference time and is extremely valuable to do in our case since we will be running on relatively weak hardware (Jetson Nano), and can use whatever performance increases we can get within reason. Theoretically, this technique will only reduce the size of decision trees without increasing errors. However, in this specific case, removing nodes from our neural network, we lose the accuracy since some may have a ripple effect through the network. An acceptable loss in pruning for our application is an arbitrary 5% loss. A high-level diagram is shown below:

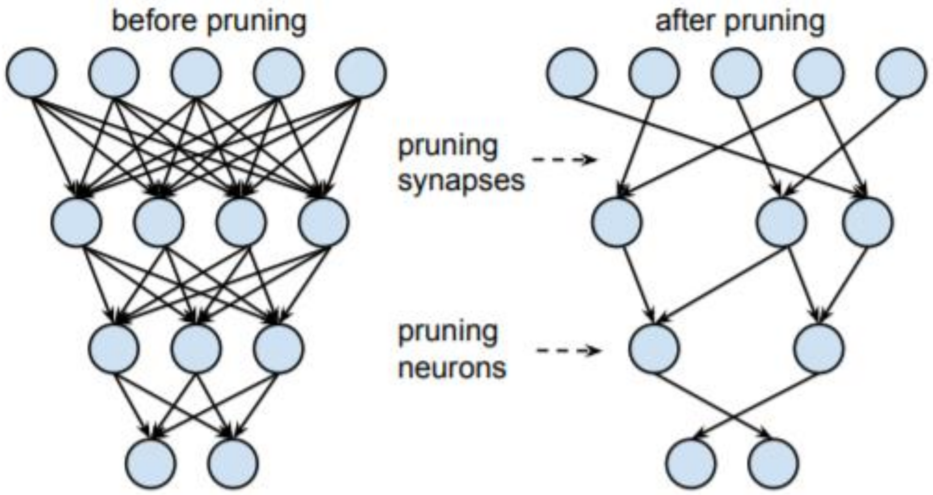
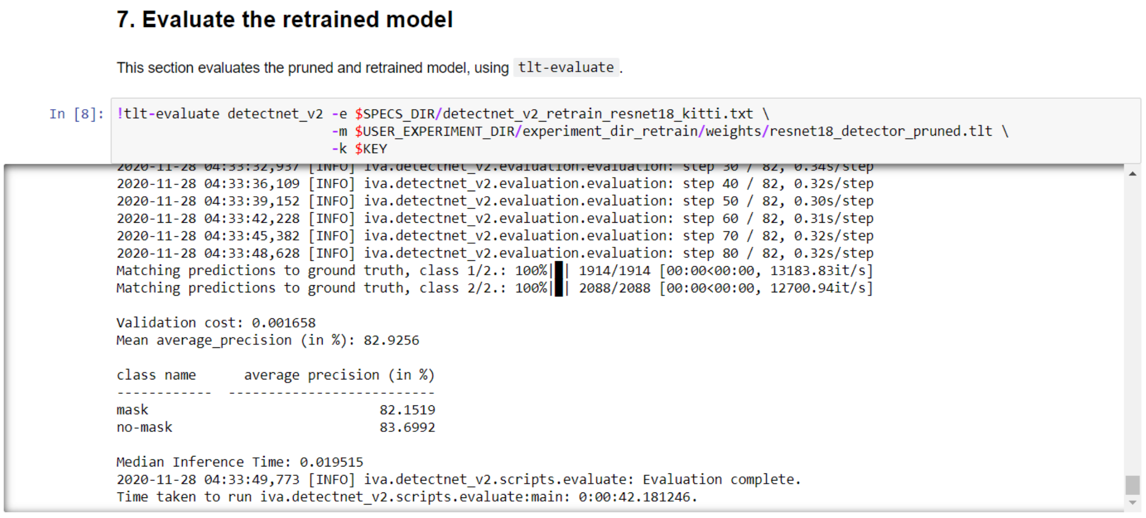


Figure 3-28 Pruning a Neural Network

To actually prune the model, we run the tlt-prune with a “-pth #” option setting that acts as a threshold value and is a floating-point number between 0 and 1. A larger number closer to 1 prune more of the network while a smaller number near 0 leaves more intact. It’s obvious that the higher the value is, the more accuracy we will lose since more of the network will be stripped away, so it ends up being a balancing act between performance gain and accuracy loss.

Additionally, due to the fact that we will be losing accuracy upon doing this it is necessary that we re-train the network once pruning finishes, and that we re-evaluate the network to see if we pruned too much. If our accuracy suffers too much (more than the arbitrary 5% stated before), we will have to go back and prune less, and re-train once again. Thus, next we prune the model with a starting value of 0.4 and re-evaluate. The re-evaluated model accuracy is shown in the figure below and is satisfactory at ~83% for both mask and no mask. Later on, we will deploy our model to the Jetson Nano, and if inference performance (frames per second) is satisfactory, we are done. However, if either accuracy or inference performance is lacking too much, we may re-visit the pruning and modify the threshold value to modify our tradeoff in this area.



7. Evaluate the retrained model

This section evaluates the pruned and retrained model, using `tlt-evaluate`.

```
In [8]: !tlt-evaluate detectnet_v2 -e $SPECS_DIR/detectnet_v2_retrain_resnet18_kitti.txt \
                                   -m $USER_EXPERIMENT_DIR/experiment_dir_retrain/weights/resnet18_detector_pruned.tlt \
                                   -k $KEY
```

```
2020-11-28 04:33:32,937 [INFO] iva.detectnet_v2.evaluation.evaluation: step 30 / 82, 0.34s/step
2020-11-28 04:33:36,109 [INFO] iva.detectnet_v2.evaluation.evaluation: step 40 / 82, 0.32s/step
2020-11-28 04:33:39,152 [INFO] iva.detectnet_v2.evaluation.evaluation: step 50 / 82, 0.30s/step
2020-11-28 04:33:42,228 [INFO] iva.detectnet_v2.evaluation.evaluation: step 60 / 82, 0.31s/step
2020-11-28 04:33:45,382 [INFO] iva.detectnet_v2.evaluation.evaluation: step 70 / 82, 0.32s/step
2020-11-28 04:33:48,628 [INFO] iva.detectnet_v2.evaluation.evaluation: step 80 / 82, 0.32s/step
Matching predictions to ground truth, class 1/2.: 100%|█| 1914/1914 [00:00<00:00, 13183.83it/s]
Matching predictions to ground truth, class 2/2.: 100%|█| 2088/2088 [00:00<00:00, 12700.94it/s]

Validation cost: 0.001658
Mean average_precision (in %): 82.9256

class name      average precision (in %)
------------  --------------------------
mask                             82.1519
no-mask                          83.6992

Median Inference Time: 0.019515
2020-11-28 04:33:49,773 [INFO] iva.detectnet_v2.scripts.evaluate: Evaluation complete.
Time taken to run iva.detectnet_v2.scripts.evaluate:main: 0:00:42.181246.
```

Figure 3-29 Pruned, Re-Trained Model Accuracy

3.6.2.1.7 Qualitative Verification of Model

The final step we would like to do in this section is to input some real-world images into our model and verify that the output is what we expect. Our intended use case is to determine whether there is a person in the image, and whether they have a mask on or no mask, thus we will input images to test both cases. This is of course running on our PC and not the Nano, but the model is still the same, so we expect the same behavior on the Nano. The figure below is one of many test images used to verify the model after the TLT flow and shows how our model is successful in categorizing those with mask vs. those without. A green box indicates a face with a mask was found, and a red box indicates face with no mask. This is a simple qualitative gauge of our inference being successful.

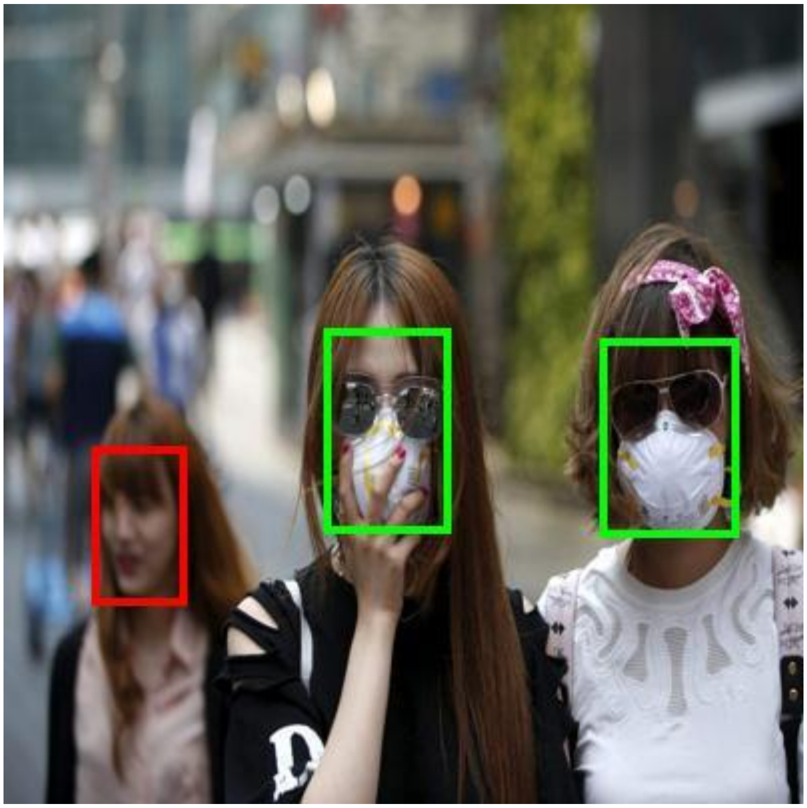


Figure 3-30 Test Image with Bounding Boxes

3.6.2.2 Deploying the model using DeepStream SDK

The last step in the AI application is finally deploying to the Jetson Nano using the Deepstream SDK. With the models that are trained with TLT, they will be compatible with DeepStream SDK because of its specific design of TLT. DeepStream SDK provides streaming analytics toolkit for AI-based videos and images understanding and allows everyone is able to create AI-based, GPU accelerated analytic application easily and efficiently. It is also able to deliver high throughput with low latency streaming processing framework. In details, DeepStream takes streaming data from camera as input and provides actionable insights that transforms pixel to deeper understanding using AI and computer vision. The SDK helps users achieve core deep learning network instead of creating end-to-end solution from scratch by using AI to analyze the metadata. DeepStream SDK is a critical step for video analytic solutions and it could vary widely in what it is able to do including detecting components, understand the retails analytics, and healthcare monitoring.

Operating system	**Dependencies**
Ubuntu 18.04	NVIDIA driver 450.51 CUDA toolkit: 10.2 TensorRT: 7.1.0

3.6.2.3 Database Communication

Our database system is based on the LAMP (Linux, Apache, MySQL, Python) stack. With many different options of web applications, building a powerful database system may become a tedious task. However, Linux, MySQL, Python are such powerful platform that contributes essential capabilities to the stack as well as providing the sufficient requirements for database system for our mask detection task.

Linux (The operating system): Serve as the first layer for this stack model. Linux is free and open-source operating system, which provide flexibilities and configuration options to run our server.

Apache (The web server): Since we are planning to host the database locally, we do not use the Apache to deliver the website on the internet.

MySQL (The database): MySQL is known as open-source relational database management system for storing information. This will be the second layer for our model, that will store the data, and deliver the information as requested.

Python (The programming language): We use Python as our primary scripting language to create dynamic database server.

This database system will have three main layers as mentioned above, Linux sets the foundation for the stack model, following by MySQL and Python. Python communicates with MySQL for fetching or storing data referenced in the code. We will manage database directly using SQL commands instead of using a web server. This stack model is efficient enough to handle the changing of data depends on date, time and user IDs.

The database will be able to automatically generate a unique number each time the person come through the mask detection system with a unique identifier to ensure the data recorded is not duplicatable. After is primary key is created (null values will be rejected), the related information about the person from DeepStream SDK analytic tool is added. The new data associated with a particular number guarantees two persons never have same unique value and allows us to parse data within the table quickly. In our database system, we will generate a simple table providing sufficient information for analytic purpose if time allows.

After taking picture of the person coming toward the camera within distance, every person in the picture will be labeled as “Mask” or “No Mask”. “PersonID” will be the primary key to distinguish with the preexisting data. “PersonID” is “AUTO_INCREMENT”, which allow us to generate unique number automatically when a new data record is added into the table, following by “MaskDetection”, will be stored in Boolean type: without mask is False while with mask is True. To store date and time information, we use SQL server timestamp type in order to define specific time a person has experience mask detection system. The MySQL timestamp is temporal data type, able to hold time and date in format *YYYY-MM-DD HH:MM:SS*, its range from “1970-01-01 00:00:01” UTC to “2038-01-19 03:14:07” UTC.

Table 3-7 Database Schema

PersonID	MaskDetection	Time
UInt	Boolean	Datetime

3.6.2.4 GPIO Driver, Monitor, Interrupt Service

For implementation of the Jetson and MCU Communication Protocol, we require some way to implement monitoring and driving of the GPIOs on the MCU in the Software. Any of the pins of Figure 3-9 Jetson Nano Pin Configuration are available, and we choose pin 10 for Nano UART Rx, and pin 12 for Nano GPIO Tx. Pin 6 is also used as GND between the two boards.

The simplest way to interface with the GPIOs on the Jetson is to use the Jetson.GPIO python library. The Tx GPIO will be driven according to the wrapper software using this library, and a blocking serial read from the UART is done using the python serial library. This is explained in more detail in the wrapper software section.

3.6.2.5 Wrapper Software

This section serves as a high-level perspective of the purpose of the image processor in terms of its function in the final design. We have previously identified and shown the purpose of the individual elements under the sections of Jetson Nano that will be running during operation. The final step in software design for the Jetson is to put everything together and decide when to run certain steps of our algorithm, along with other decisions such as driving GPIO pins, reading the UART, etc.

For the Jetson Nano to meet its requirements we must have a way to tie everything together from a high-level. To do this, we incorporate a wrapper Python script that controls and monitors the inference portion from Deepstream SDK, the database insertion function, and the monitoring/driving of GPIO/UART pins to/from the microcontroller. Pseudocode for the wrapper software is given in the appendix, Wrapper Software Pseudocode.

3.6.3 Microcontroller Software Algorithm

The software algorithm for the MSP430G2553 is nowhere near as complex as the Jetson Nano, but rather, it is quite simple and straight forward. The primary reason for the simplicity is due to the lack of a dedicated Operating System. To get to the GPIOs for configuration, the Jetson Nano has to first obtain OS permission, which makes the code rather lengthy and takes a few instructions cycles to execute. With the MSP430G2553, the programming code directly influences the behavior of the pins, so it is quite snappy. There are 3 main parts to the software algorithm of the MSP430G2553, and all of which are interrupt-based.

The first part is the distance detection. Every 0.5 second, the MSP430G2553 will enable the proximity sensor to detect if an object has gotten within the range of 1 meter. If the answer is yes, the sensor’s interrupt will send the “START” signal to the Jetson Nano for it to take the picture of the subject and starts computation. Then, it will enable the status interrupt of the microcontroller to start listening to the updates from the Jetson Nano. The Jetson Nano will have been set up with interrupt enabled such that once it receives the "START" signal from the MSP430, the interrupt service routine will wake the Jetson Nano up from deep sleep and commence operation.

Follow closely the first part is the status interrupt when information is received from the Jetson Nano. These interrupts are hardwired into the 5th bit of port 1. The bit is setup as active high, so there will be a resistor that initially pull the bit down to low. Because most other operations are already set to happen on rising edge, the status interrupt is also set to happen on rising edge of the clock to keep everything in sync.

Finally, the final main part of the MSP430 software algorithm is the override button interrupt. This was specifically configured such that if the Jetson Nano stalls or if the user just wants quick access to masks without having to wait for the computation to finish, he can just press the button to override everything else that is happening. First the ISR will check for the status of the door. If the door is already up, it will do nothing. If the door is not up, then it will enable to timer that controls the action of the door to open, wait, then close the door. This interrupt shares the same ISR as the status interrupt from the Jetson Nano.

The door's timer interrupt will raise the door, hold it there for 10 seconds, then close the door. After the door is closed, the timer interrupt will also clear all the signals and flags that were triggered in the process and retain the MSP430 to the initial stage of continuously fetching distance.

The code snippets for this section are presented in the appendix.

3.7 User Experience and Usage

Here is the short sequence of operation for the machine. First, the user will need to stand within 1 meter of the machine to be detected by the proximity sensor. Once the proximity sensor has picked up that there is an object in front of the machine, the microcontroller will send a "START" signal to the Jetson Nano to wake it from deep sleep. Immediately after being woken up, the Jetson Nano will invoke the camera to take an array of images and chooses the best one among those to analyze. The Jetson Nano will run through the machine learning model to determine if the person in front of the machine is wearing a mask or not. If the person is wearing a mask, nothing will happen, and the Jetson Nano will go back to sleep. However, if the person is not wearing a mask, then the Jetson Nano will send a signal to the microcontroller, telling it to open up the mask robot's door for the user to reach in a get a mask. While the door is opening, the microcontroller will ring the buzzer 3 times to alert the user to fetch a mask. After signaling the microcontroller, the Jetson Nano will go back to sleep. After 10 seconds, the microcontroller will automatically close the door. Now, let us say that the user is too short, and that he is out of range of the proximity sensor and of the camera, then the door will never open. If the user still wants a mask, then there will be an override button located next to the door so that the user can manually force the door to open without having to first being scanned by the Jetson Nano. This entire process is further summarized in the flow chart below.

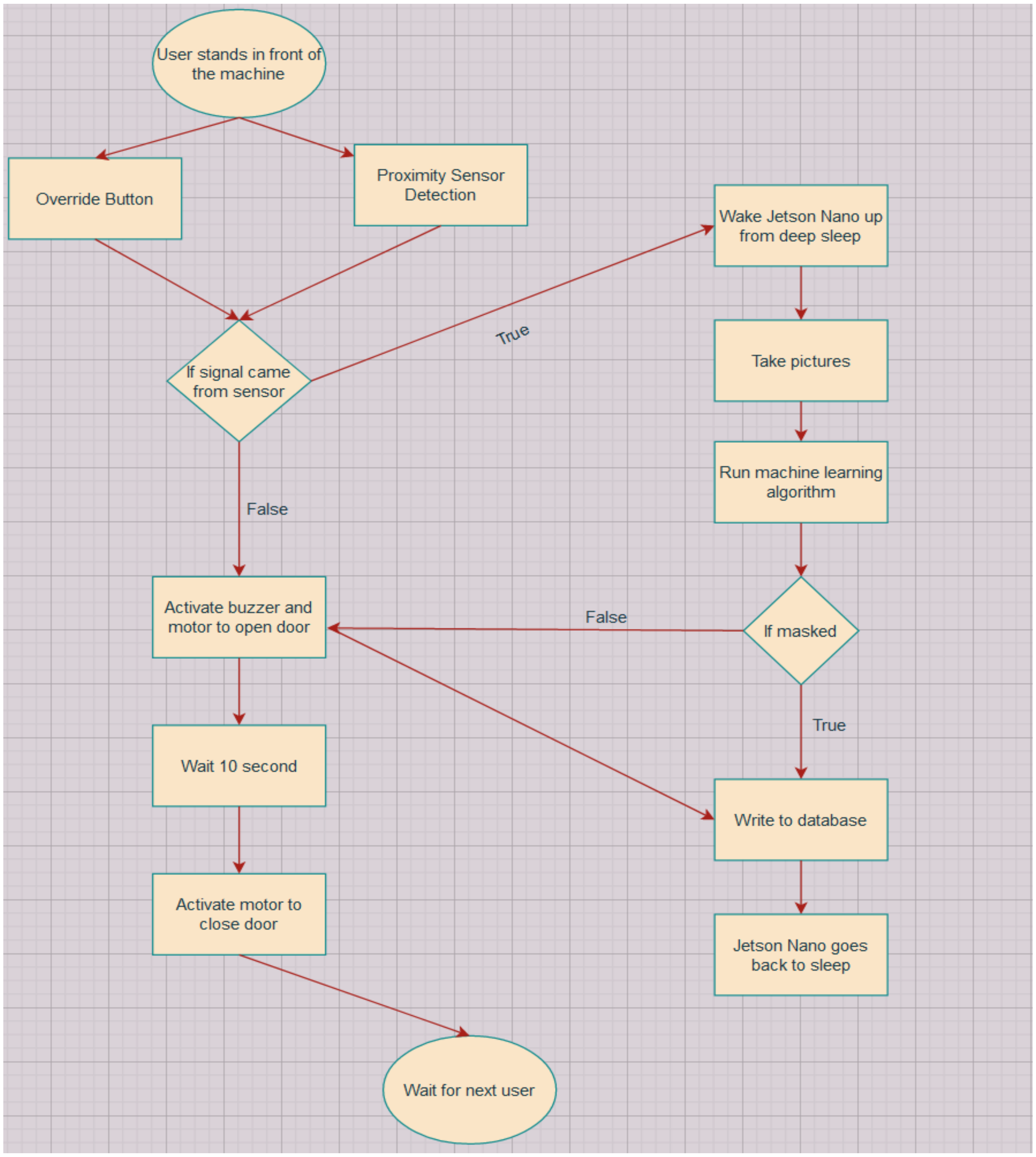

User stands in front of the machine
Override Button
Proximity Sensor Detection
Wake Jetson Nano up from deep sleep
True
If signal came from sensor
Take pictures
Run machine learning algorithm
False
Activate buzzer and motor to open door
False
If masked
True
Wait 10 second
Write to database
Activate motor to close door
Jetson Nano goes back to sleep
Wait for next user

Figure 3-31 Order of Operation Flowchart

4 DESIGN SUMMARY

This section basically sums up all the decisions that we have made for our robot. First and foremost, we are budget conscious. All of our components are chosen such that the final cost is as small as possible. Wherever we could, we would opt for used/refurbished parts, or parts that we have bought and used for a previous project.

For our exterior body design, we have chosen to go with the basic acrylic sheets. They are easier than most materials to work with and are relatively cheap for their sophisticated look and durability. What more, they are fairly easy to cut for amateurs without dedicated tools as well. For the front sheet, we will cut out a door latch such that a motor can open and close so that users can easily reach into the body to fetch a mask.

With small systems like this, precise voltage and current control is crucial for the smooth operation. Thus, we have chosen to go with 3 separate voltage rails to make sure that each component has enough power to operate at its peak performance. The 3 voltage rails are 12V, 5V, and 3.3V. The voltage is supplied from a 120Vac source, which is later stepped down via a transformer. Since we are working directly with source voltage, rather than depending on third-party AC-DC converters, we can expand the system to include many more voltage rails in case we want to include more components in the future.

Next is the microcontroller. We have chosen the MSP430G2553 for peripheral controls. Its processor is fast for its class, up to 16MHz, and its number of GPIOs are plentiful enough for our purpose. The microprocessor will control the motor to open and close the door of the robot. It will also be in charge of periodically fetching information from the proximity sensor to detect whether a person is within range. Finally, it will quickly and efficiently deliver command to and from the Jetson Nano to process.

Finally, we have the NVIDIA Jetson Nano. Among all of NVIDIA’s current development systems line up, it is the weakest. However, compare to competitors from other brands, it does not lack behind at all. Although it is weak when compare to NVIDIA’s other products, it has enough processing power to do Machine Learning and image processing quick enough for our use. The Jetson Nano’s primary mission is to capture the image of the person standing in front of the machine, then it will process the image to determine if the person is wearing a mask. After it is done processing, it will notify the MSP430G2553 of the result and store the processed result in a local MySQL database. The software we have chosen for the Jetson Nano are also free and open-sourced. They are either obtained from Github, like with Jupyter Notebook and OpenCV or directly through NVIDIA, like with the DeepStream SDK.

5 TESTING AND EVALUATION

In this section, we will discuss the testing plans that was and will be implemented to make sure that each of the components of the robot functions as desired. This will help us discover flaws in our design and hardware selection, as well as give us ample time to rectify the problems. This initial planned was made early in the design process, so as the project matures, the testing will change to accommodate new parameters.

5.1 Hardware

This is one of the parts which is really important for our project. Before building the power delivery system prototype, we need to test all the components that we ordered to make sure that it works properly. The major components that we got such as microcontroller, voltage regulators, resistors, capacitors, transformer, wires, and some of the components will be taken from laboratories. Other components will be ordered at the beginning of Senior

Design 2 after we finalize the layout of the PCB. The tests are summarized in the table 5-1 below.

Table 5-1 Electrical Components Testing

Component	Test	Outcome Determination
DC Power Brick (12V rail)	Use the multimeter to check the output voltage and the maximum current output	The output voltage should constantly be at 12VDC, and the maximum current should be no less than 5A if the power brick is functioning correctly
5V rail	Use the multimeter to check the output voltage and the maximum current output	The output voltage should constantly be at 5VDC, and the maximum current should be no less than 6A if the power brick is functioning correctly
3.3V rail	Use the multimeter to check the output voltage and the maximum current output	The output voltage should constantly be at 3.3VDC, and the maximum current should be no less than 2A if the power brick is functioning correctly
Push button	Use the multimeter to measure the resistance of the switch as the button is pressed	If the resistance drop from high to close to 0, then the button is working correctly.
Motor	Connect the terminals of the motor with a voltage source and a PWM source	If the motor turns on and rotates, then it works
MSP430 pins	Set the pins to high and low. Use multimeter to measure the voltage at each pin with respect to ground	If the high voltage is around 3.5V and the low voltage is around 0V, then the pins of the MSP430 works
MSP430 timer	Configure PWM for a PWM-enable pin on the MSP430. Use the oscilloscope to measure period and the active percentage within the period	If the period and the active percentage matches what was set in the software, then the internal timer works perfectly
MSP430 UART	Configure UART. Set up a serial communication with a computer. Send data using UCA0TXD	If the computer is connected to the MSP430 and the correct data is printed onto the serial console, then UART works

MSP430 Interrupt	Configure timer interrupt every second. Toggle an LED whenever there is an interrupt	If the LED toggles regularly, the Interrupt routine works
Proximity sensor	Configure UART on MSP430. Place an object in front of the sensor. Print out sensor’s status to console continuously	If the sensor’s readings are close to the actual distance of the object to the sensor, then the sensor works
Buzzer	Run a PWM signal through the buzzer	If the buzzer makes sound and the sound changes as the PWM signal changes, then the buzzer works
Image camera	Connect the camera to the Jetson Nano	If there is an image, then the camera works
Jetson Nano	Connect to a monitor	If there is an output image, then the Jetson Nano is working

Based on the table above, we will test the components that we have received. First off, we set the digital multimeter from the laboratory to measure the microfarad, and we have to make sure to discharge them so as not to damage the device itself. After using probes to measure between two terminals, we got the desired values. So, the capacitors are working perfectly. Second, we start testing the diodes. Since the diode only allows the current to flow in one direction from anode to cathode, we must determine the cathode and anode before measuring. Most of the diodes have a band on one side which is known as the cathode side, and the other side is anode. By using two probes to measure the DC voltage through them, the red lead is connected to the anode and the black one is connected to the cathode side. We got the voltage around 0.5V which matches with the provided datasheet. On the other hand, in order to check the reverse biased diodes, we just need to reverse the probes which means the positive (red) test lead is on the cathode and the negative (black) test lead is on the anode. Lastly, we test the transformer. To test the transformer, we need to set the digital multimeter to AC voltage. After plugging the input side into the wall outlet, we carefully measure the voltage at the input to make sure that we get 120VAC then start measuring on the output side, and we got 14.5VAC which means the transformer is working correctly.

After all the test is passed, the circuit is built on the breadboard. We start testing the output at each stage. At the first stage where the first voltage regulator is placed, we measure the DC voltage between output terminal of the voltage regulator versus ground, we got 12V which desired our requirement. Doing the same test as the first voltage regulator for the second and third voltage regulator, and we finally got the 5V and 3.3V at the second and third voltage regulators, respectively. With all the test that we just performed, that demonstrated that our components were working correctly. The power delivery voltage testing was done successfully as presented in the figures below. The 3.3V rail is right on point. However, the 5V rail overshot our desired voltage by a little bit. This is not really a concern as the motor and motor driver that we used are tolerable toward a range of voltages.

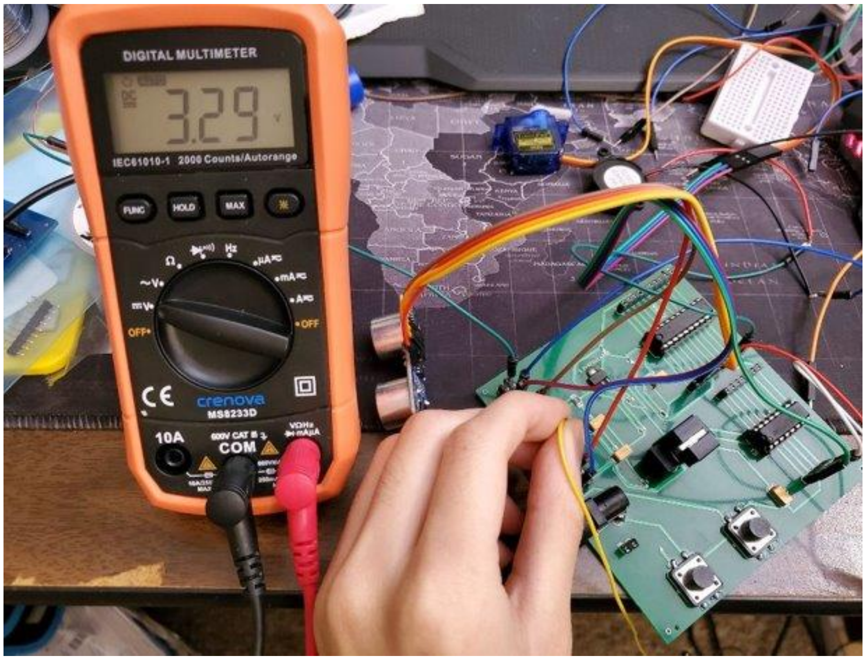


Figure 5-1 3.3V-rail Testing

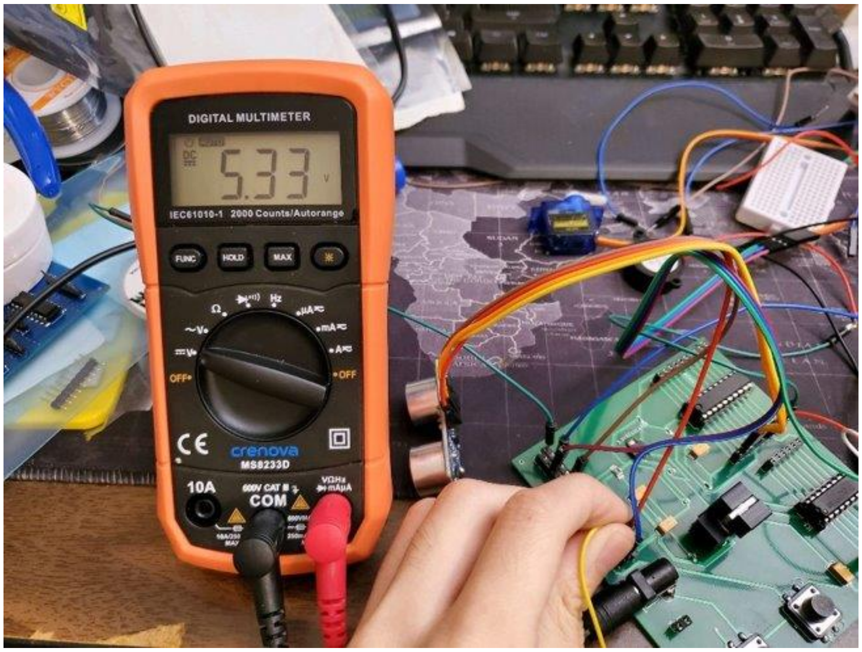


Figure 5-2 5V-rail Testing

After finished testing the power delivery, we will also need to test the MSP430G2553's connectivity and its peripherals. Setting up the peripheral testing for the MSP430G2553 is quite simply and straight forward. All we need to do is to connect the peripherals to the desired pins and turn on/off those pins to see the behavior changes of the peripherals. If the timer and interrupts are set up correctly, the proximity sensor will send back information to the MSP430G2553 on a regular interval. If the proximity sensor works without problems, then the value received should be within 1 meter of the actual distance. As for the motor, by changing the PWM configuration, the direction of the servo motor will change with the supplied voltage being constant.

For testing and debugging purpose, we will also need the UART functionality of the MSP430G2553 to be working correctly. To know if we get the correct value from the proximity sensor or to know what we get from the Jetson Nano, an easy way to do so without having to probe the voltages of the IO pins is to print out the result to the console. Normally, we would result in using "printf" function to print. However, the "printf" function requires more memory than the onboard memory of the microcontroller. To print out the information, we need to use UART. After setting up the UART, we can then transmit all information received to the console so that we can ascertain if we have gotten the expected value, or where the program may have stalled at.

5.2 Software

The MSP430G2553's software ties directly to its hardware interfacing, so to test its software, the simplest way to do so is to test the hardware. Thus, the MSP430G2553's testing will not be listed in this section.

Table 5-2 Camera Testing

Test Name	Test camera feed
Purpose	Verify Jetson can utilize Camera image Verify Camera functionality
Test Materials	Jetson Nano Jetson Nano power supply Camera Module Keyboard, Mouse, HDMI Monitor
Prerequisites	Power supply, keyboard, mouse, HDMI monitor and camera plugged into Nano Jetson JetPack installed on Nano
Procedure	Pull the NVIDIA "jetson-inference" Docker container Follow instructions from NVIDIA's "Running the Docker container". Run application, follow the script and use command line *./video-viewer /dev/video0*
Expected results	This test should return the video stream in real time using Jetson Nano

Table 5-3 Mask Detection AI Testing

Test Name	Mask Detection

Purpose	Verify the AI model qualitatively, independent of the Jetson before deployment.
Test Materials	face-mask-detection.ipynb Jupyter Notebook (face-mask-detection github) Trained and Pruned Face Mask Detection AI model (resnet18_detector_pruned.tlt)
Prerequisites	Windows 10 PC with WSL2 installed along with Ubuntu 20.04 Docker installed and started resnet18_detector_pruned.tlt trained and pruned AI model Environment set up for face-mask-detection docker container Jupyter Notebook Testing images consisting of people with mask, no mask in the image
Procedure	1. Modify the detectnet_v2_inference_kitti_tlt.txt file to point to the resnet18_detector_pruned.tlt model. 1. Drop testing images inside DATA_DOWNLOAD_DIR in the Docker container. 2. Run step 8 of the Jupyter Notebook with the pruned AI model. 3. Visually confirm the output images.
Expected results	The output images are expected to have been labeled with the persons wearing masks/no-masks in the image correctly. Red bounding boxes signify a person with no mask, and green signifies a person with a mask.

Table 5-4 Jetson Nano Deployment Testing

Test Name	Jetson Nano Deployed Model
Purpose	Qualitatively determine that the Jetson can run our AI model using DeepStream in real-time from the Camera.
Test Materials	Jetson Nano Jetson Nano power supply Camera Module Keyboard, Mouse, HDMI Monitor
Prerequisites	resnet18_detector_pruned.etlt pruned and deployed mask detector model Jetson Nano Camera connected, verified Jetson Nano DeepStream SDK set up
Procedure	Launch the Deepstream SDK program pointing to the .etlt model. The video should be viewed on the screen with the AI results present.
Expected results	The program should output the video stream for mask detection with green box outlining if a person wearing a mask, and a person without a mask, the bounding box is red.

	This bounding box will appear around human's face that been detected and will follow the face until the person move out of the camera's view.

Table 5-5 Jetson Nano GPIO Interface Testing

Test Name	Jetson Nano GPIO
Purpose	Quantitatively determine that we are able to read from the GPIOs and output to the GPIOs used for the communication between the Jetson and the MCU.
Test Materials	Jetson Nano Jetson Nano power supply Camera Module Keyboard, Mouse, HDMI Monitor Multimeter
Prerequisites	Python GPIO script
Procedure	1. Set the Tx pin to "out", Rx pin to "in" using the GPIO script. 2. Read the Tx pin with multimeter set to voltage mode with positive connected to Tx GPIO pin, negative connected to the GND on the GPIO. 3. Use Python GPIO script to set Tx pin HIGH. 4. Read voltage from Tx pin. Verify it is at 3.3v. 5. Read Rx pin using GPIO script. Verify it is at logic level LOW. 6. Connect Nano 3.3v DC to Rx GPIO pin. 7. Read Rx pin using GPIO script. Verify it is at logic level HIGH.
Expected results	The program verifies functionality for the Nano's GPIO pins for Rx and Tx.

Table 5-6 Jetson Nano Full System Test

Test Name	Jetson Nano Full System Test
Purpose	Determine that our entire system works as expected on the Nano's end.
Test Materials	Entire system connected.
Prerequisites	Complete assembly, connection from MCU to Jetson Nano GPIO pins
Procedure	User steps in front of distance sensor to activate the system with no mask. The system should initiate and trigger the Nano to begin inference.
Expected results	Upon no mask being seen, the mask case should open along with the buzzer ringing.

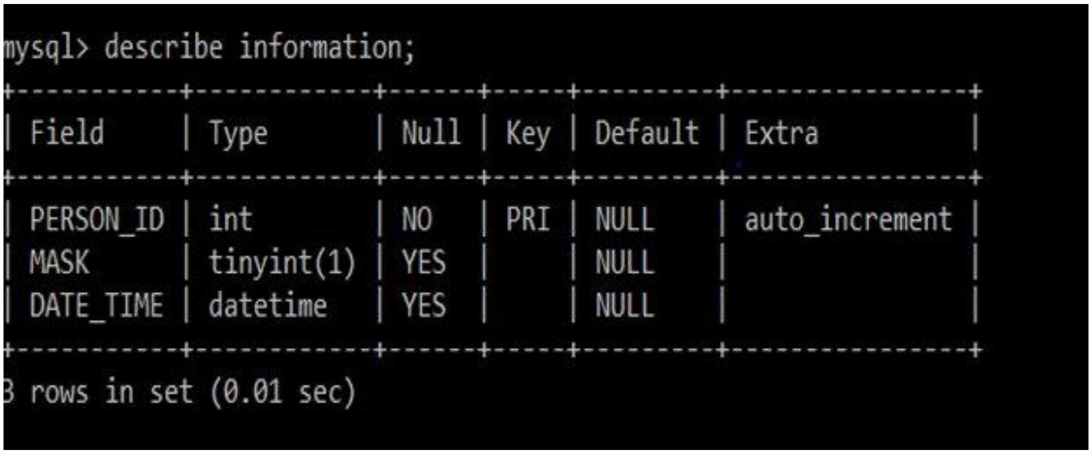
5.2.1 Results

In this face mask detection model, we use Detectnet_v2 as the head architecture with ResNet18 as the backbone feature extractor. Detectnet_v2 is one of the most popular model architectures that NVIDIA provides for object detection, and produces an output frame with attached metadata that includes bounding boxes of objects detected, frame number, timestamp, etc. ResNet-18 is a convolutional neural network backbone that uses 18 layers and has been widely used in many computer applications because of its balance between performance and accuracy.

In total, we used 12,000 images for our mask/no-mask dataset– half with mask, half without, and resized the images each to a resolution of 960x544. After training, pruning, and retraining, we were able to end with a validation accuracy of 84%.

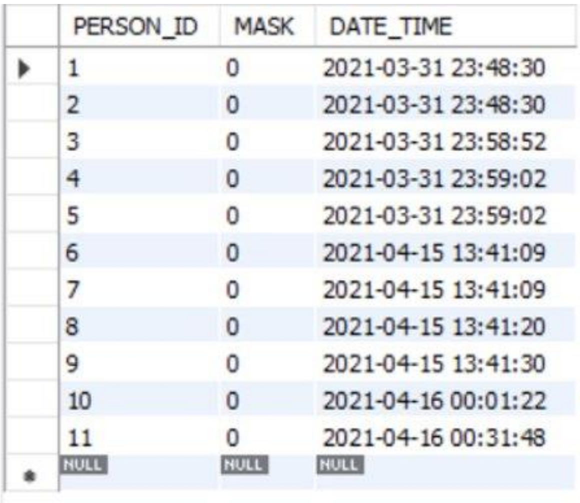
The machine learning model itself was deployed on the Jetson Nano using NVIDIA Deepstream SDK. We vaguely followed an example design given by NVIDIA to outline our software stack and added extra code on top for extraction of metadata from each frame. We used this metadata to detect how many objects of interest we found in the image: those wearing masks and those not. This code was then wrapped with the rest of our code to communicate with the MCU, and with the database.

Our database system is based on the LAMP (Linux, Apache, MySQL, Python) stack with 4 major components which meets our requirements for the mask detection database. After recording a video of 128 frames once the MCU signals a start condition, every person detected in the image will be labeled as “Mask” or “No Mask”. “PersonID” will be the primary key to distinguish with the preexisting data (people previously seen). “PersonID” is “AUTO_INCREMENT”, which allows us to generate a unique number automatically when a new data record is added into the table, following by “Mask”, will be stored in Boolean type: without mask is 0 and with mask is 1. To store date and time information, we use SQL server DATETIME data type to define the specific time a person has been detected by the system.



```
mysql> describe information;
+-----------+------------+------+-----+---------+----------------+
| Field     | Type       | Null | Key | Default | Extra          |
+-----------+------------+------+-----+---------+----------------+
| PERSON_ID | int        | NO   | PRI | NULL    | auto_increment |
| MASK      | tinyint(1) | YES  |     | NULL    |                |
| DATE_TIME | datetime   | YES  |     | NULL    |                |
+-----------+------------+------+-----+---------+----------------+
3 rows in set (0.01 sec)
```

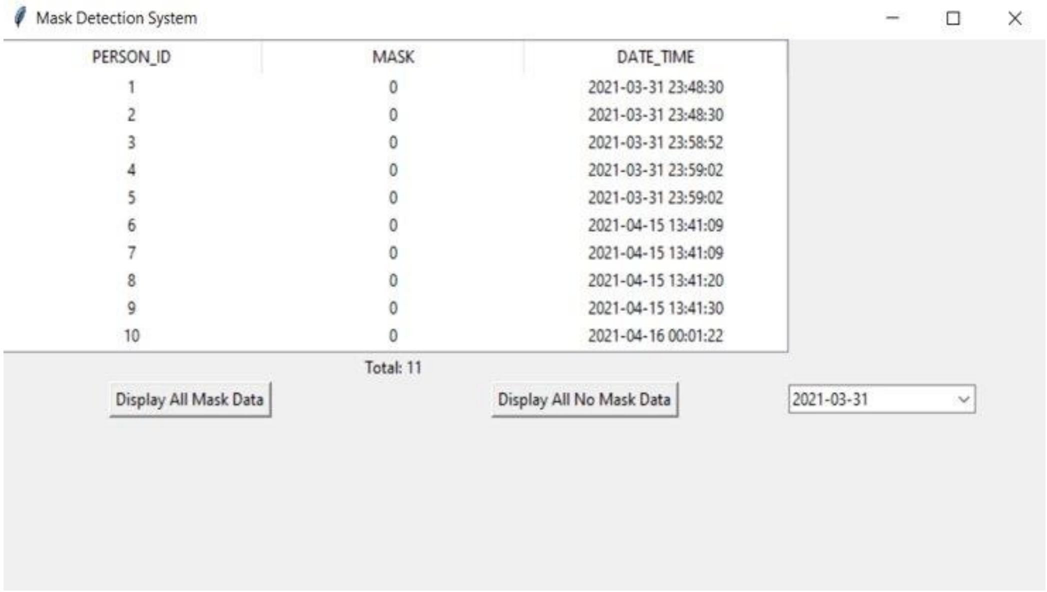
Figure 5-3 Database system



PERSON_ID	MASK	DATE_TIME
1	0	2021-03-31 23:48:30
2	0	2021-03-31 23:48:30
3	0	2021-03-31 23:58:52
4	0	2021-03-31 23:59:02
5	0	2021-03-31 23:59:02
6	0	2021-04-15 13:41:09
7	0	2021-04-15 13:41:09
8	0	2021-04-15 13:41:20
9	0	2021-04-15 13:41:30
10	0	2021-04-16 00:01:22
11	0	2021-04-16 00:31:48
NULL	NULL	NULL

Figure 5-4 MySQL database

We created a graphical user interface, specifically using python's Tkinter module and MySQL database server. This application is user friendly and allows users to see all data from the database along with display mask/no mask data on different dates. The user is not able to change data, like inserting or deleting because the data is read only. This database could potentially help researchers to collect data or statistics of individuals wearing mask vs no mask in public places.



PERSON_ID	MASK	DATE_TIME
1	0	2021-03-31 23:48:30
2	0	2021-03-31 23:48:30
3	0	2021-03-31 23:58:52
4	0	2021-03-31 23:59:02
5	0	2021-03-31 23:59:02
6	0	2021-04-15 13:41:09
7	0	2021-04-15 13:41:09
8	0	2021-04-15 13:41:20
9	0	2021-04-15 13:41:30
10	0	2021-04-16 00:01:22

Total: 11

Figure 5-5 GUI application

6 ADMINISTRATIVE CONTENT

In order to succeed, we should have well-planned and managed carefully and concisely. Administrative is the key to help the project succeed. In this section, the administrative aspects will be including our backgrounds, skills, budgeting, project milestones, and the final product. The timeframe that from the beginning of the project to the presentation time will be shown in the table below, and this will help our project on schedule and budget. Budget is important in the development of every project because it will show the detail of each expense and make sure that there is no over budget for the project.

6.1 Personnel

Our group consists of 4 members. Below are short introductions of our members, and each of the members has their responsibilities for the project. They are sorted in alphabetical order of last name.

Emanuel Cicortas is a Computer Engineering student whose interest is in VLSI. He will be working on the Jetson Nano to help design the mask detection software, train the AI model, and deploy to the Jetson. He interned for Lockheed Martin in FPGA design for 2 years and has accepted a full time offer with NVIDIA as an ASIC Verification Engineer.

Dat Nguyen is a Computer Engineering student who specializes in microcontrollers. For this project, he will be in charge of programming the microcontroller as well as testing the peripherals. After graduation, he will continue his academic endeavor at UCF.

Ha Tran is the only Electrical Engineering student in our group, and his primary interest is in Power Systems. Ha has been working as an intern at Jacobs for 1.5 years, and he has experience in the renovation of electrical engineering design services for a variety of projects encompassing building power, power distribution, lighting, controls, etc. Additionally, Ha has extensive experience working with MicroStation, AutoCAD and AGi32. In this project, he specializes in signal processing, so he is in charge of designing and testing the power delivery of our system. Upon graduating, he will transition from an intern to full time employee at Jacobs as an Electrical Designer.

Phuong Anh Vu is a Computer Engineering student who specializes in software design. In this project, she will be in charge of building database server and helps with software design including facial recognition and machine learning algorithms. She plans to pursue working career in computer engineering profession in short-term future and gain a Master in Machine Learning in long-term future.

6.2 Budget and Financial Details

6.2.1 Group Budget

Our group’s project is self-fund and is subjected to our personal earnings and willingness to spend. After talking with each other, we have determined that we can spend no more than $1000 on this project. However, if necessary, we will be able to fork out another $50 to spend on the project. We are all students with only low-pay part time jobs to sustain ourselves, so we cannot afford to spend more than that.

6.2.2 Component Costs

The cost of each component is listed in table 6-1 below. Roughly $500 of the $1000 of the allocated budget have been used up. This leaves us with roughly $200 for the PCB manufacturing. In addition to the PCB's BOM, we also have to account for other components that were bought. Because we made a change in the power delivery systems, not all the components listed here will be used. The table below lists all the components that we needed aside from the components listed in the BOM.

Table 6-1 Current Costs

Component	Count	Total Cost	Acquired	Tested
1N4001 Diode	20	$4.80	Yes	Yes
Keurig B31 Transformer	1	$7.78	Yes	Yes
Solenoid	2	$12.25	Yes	Yes
DC Motor	2	$11.18	Yes	Yes
LM7812 12V Regulator	5	$4.98	Yes	Yes
Push Buttons	3	$2.95	Yes	No
LM7805 5V Regulator	5	$4.50	Yes	Yes
AMS1117 3.3V Regulator	5	$3.39	Yes	Yes
MT 5012 Mini Speaker	1	$5.79	Yes	Yes
L293D Motor Driver	5	$6.50	Yes	Yes
Micro-USB Male Terminal Jack	10	$5.24	Yes	No
128GB MicroSD Card	1	$23.39	Yes	Yes
NVIDIA Jetson Nano Dev. Kit	1	$108.95	Yes	Yes
IMX219 Camera Module	1	$21.77	Yes	Yes
MSP430G2553IN20	3	$5.52	Yes	Yes
100nF Capacitor	5	Stolen from lab 4/5	Yes	Yes
10uF Polarized Capacitor	1	To be stolen from lab	Yes	Yes
47K Ohm Resister	2	To be stolen from lab	Yes	Yes
100uF Polarized Capacitor	4	Stolen from lab	Yes	Yes

Component	**Count**	**Total Cost**	**Acquired**	**Tested**
HC-SR04 Ultrasonic Sensor	1	Don't remember (bought for JrD)	Yes	Yes
FLIR Lepton 3.5 Thermal Camera	1	~~$200~~ (returned)	Yes	Yes
Servo Motor	10	$15.99	Yes	Yes
Piezo Buzzer	2	$5.99	Yes	Yes
Acrylic Sheets	8	$43.98	Yes	Yes
PCBs	3	$252.64	Yes	Yes
Cables	40	$42.40	Yes	Yes
Subtotal	$589.99			

6.3 Milestones

6.3.1 Senior Design 1

For Senior Design 1, initially, we had wanted to have at least finished all the testing and have started on designing the PCB. However, due to having only limited access to the lab, we had some trouble booking lab spots to test our set up. Firstly, for this semester, the lab is only opened during weekdays. This means during the days when we have the most time available, we were unable to access the lab. Second, the school closes before the semester even ended. The semester ends the second week of December; however, school closes the week of Thanksgiving. Originally, we had wanted to go to lab at least 3 times to test components, build circuit, and load test the power delivery system. This plan quickly went down the drain from the time constraint and we were only able to go to lab once. I really hope that it will be easier to have lab access for the next semester so that we can finish our circuit load testing quickly and move on to PCB design.

As for the software side, we never expected the machine learning model training process to take so long and be so involved. At first, we had wanted to just set up the training and have the AI train itself. Then, we would spend time designing and setting up the database to store analyzed data. However, the model training process was more involved than expected. It took almost 2 months just to get the model to the 85% accuracy mark. Even to the final day of Senior Design 1, the machine learning model has still not reached the accuracy that we desired, which further delays the database implementation even more.

6.3.2 Senior Design 2

Our first priority for Senior Design 2 was to finish load testing the power delivery and adjust it if any adverse electrical reaction happens. Power delivery is the most important part of any system. To make sure that our system is reliable and operates within specifications, we need to do our due diligent to make sure that the power delivery can deliver the expected voltage at the appropriate current value, as well as making sure that the power deliver is able to sustain a heavy load without overheating. The breadboard testing went smoothly without any problems. Next step after that is to design the PCB and send the schematic to a manufacturer to get the PCB made. Then while waiting for the PCB, we will measure and cut the acrylic sheets into 10in x 10in sheets and glue them together using hot glue gun. After receiving the PCB, we tested the voltage and signal leads and mounted the components onto the box. And finally, we have to test the robot to make sure that everything still functions even after withstanding abuses from amateurs.

In parallel with our hardware designing process, we will continue to train the machine learning model such that the accuracy is as close to 100% (80+% success rate achieved) as possible before the end of February. Once the machine learning model is to our satisfaction, we will implement a MySQL database to store the information that the machine collected after analyzing images. We have implemented a database before for other classes, so we do not think this part will take too long. Though saying that, accidents can and will happen. To account for unexpected events like another pandemic, we tried to plan such that we have everything finished by mid-March. And Murphy's law held true. We had trouble with the PCB and had to get it remanufactured. Thankfully, everything came in on time before the final deadline.

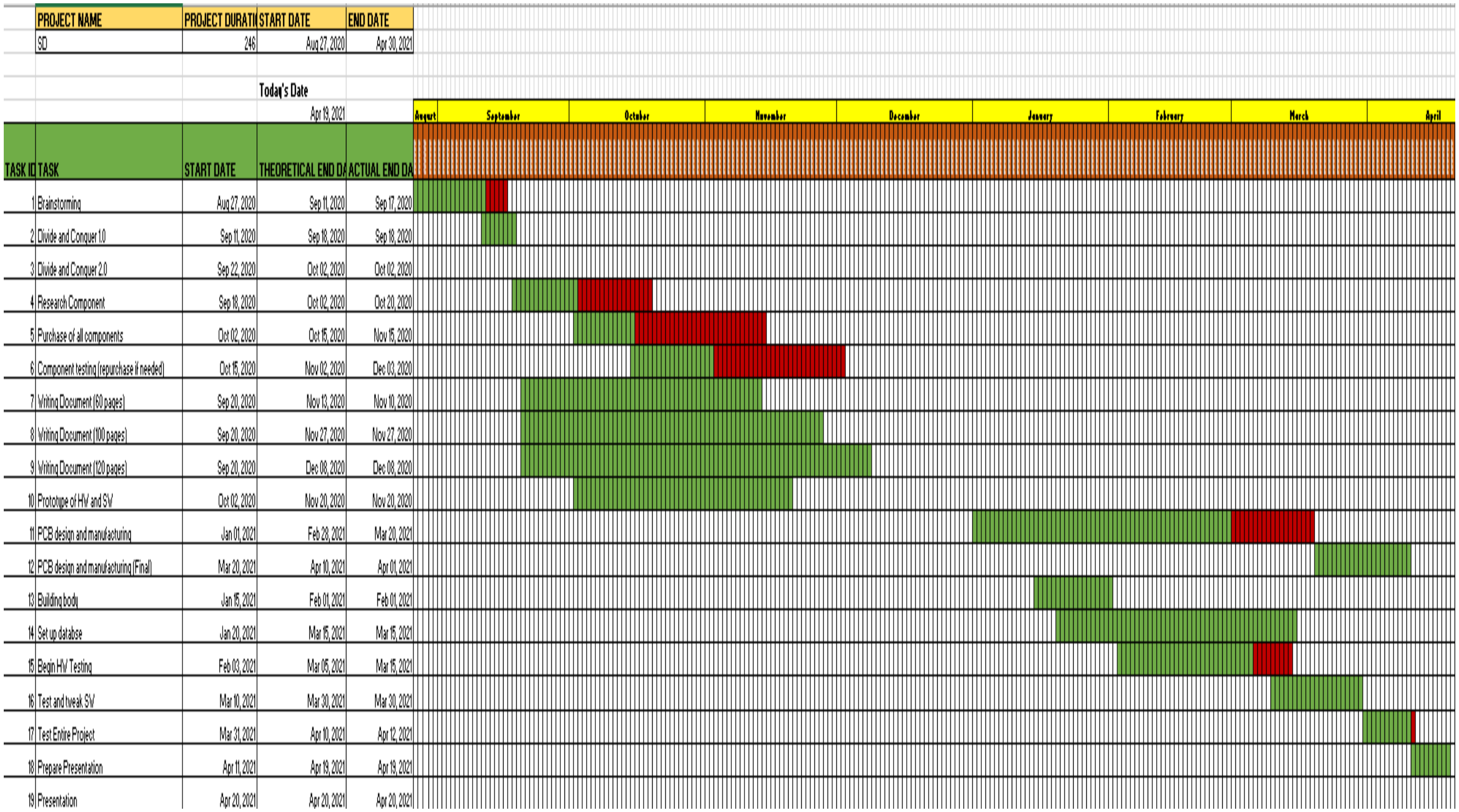
6.3.3 Machine Built and Functioning

We have built the prototype for both the hardware and software aspects of the system. However, we have yet to fully test the set up. We have built the final body for the system to be 10x10x10in. The cardboard body that we built made us realize that the brushless DC motor is not suitable as a door opening and closing mechanism. Sometimes it over spins, some other times, it under spins. While the variability is minor, it still affects the functionality of the machine. For our final build, we will go with a servomotor, as a servomotor has high torque and can be specified in term of direction and position. Currently, our power delivery prototype is built in series such that the lower voltage rail is 100% dependent on the higher voltage rail. This works for now, but for the reliability of the system, we will change the power deliver so that the voltage rails will be in parallel so that if one fail, the others will not be affected.

As for software prototype, our facial recognition can detect faces, however, to detect if the face is wearing a mask, our algorithm only works around 86% of the time. We will continue to train the machine learning model so that by the time of the final build, the machine learning model can detect a mask wearing face with the accuracy of at least 95%.

6.3.4 Gantt Chart (Project Timeline)

For a complex project, we need to create a plan. From there we can select who will be assigned to do the work based on their skills and availability. In our project, we chose the Gantt Chart as a project management tool that assists us in planning and keeps track of our progress. Figure 6-1 below is the Gantt Chart where it shows the project progress timeline for the deadline of each task that needs to be done to accomplish our project in Senior Design 1 and milestone for Senior Design 2.



PROJECT NAME	PROJECT DURATI	START DATE	END DATE
SD	246	Aug 27, 2020	Apr 30, 2021

Today's Date

Apr 19, 2021

TASK ID	TASK	START DATE	THEORETICAL END DA	ACTUAL END DA
1	Brainstorming	Aug 27, 2020	Sep 11, 2020	Sep 17, 2020
2	Divide and Conquer 1.0	Sep 11, 2020	Sep 18, 2020	Sep 18, 2020
3	Divide and Conquer 2.0	Sep 22, 2020	Oct 02, 2020	Oct 02, 2020
4	Research Component	Sep 18, 2020	Oct 02, 2020	Oct 20, 2020
5	Purchase of all components	Oct 02, 2020	Oct 15, 2020	Nov 15, 2020
6	Component testing (repurchase if needed)	Oct 15, 2020	Nov 02, 2020	Dec 03, 2020
7	Writing Document (60 pages)	Sep 20, 2020	Nov 13, 2020	Nov 10, 2020
8	Writing Document (100 pages)	Sep 20, 2020	Nov 27, 2020	Nov 27, 2020
9	Writing Document (120 pages)	Sep 20, 2020	Dec 08, 2020	Dec 08, 2020
10	Prototype of HW and SW	Oct 02, 2020	Nov 20, 2020	Nov 20, 2020
11	PCB design and manufacturing	Jan 01, 2021	Feb 28, 2021	Mar 20, 2021
12	PCB design and manufacturing (Final)	Mar 20, 2021	Apr 10, 2021	Apr 01, 2021
13	Building body	Jan 15, 2021	Feb 01, 2021	Feb 01, 2021
14	Set up databse	Jan 20, 2021	Mar 15, 2021	Mar 15, 2021
15	Begin HW Testing	Feb 03, 2021	Mar 05, 2021	Mar 15, 2021
16	Test and tweak SW	Mar 10, 2021	Mar 30, 2021	Mar 30, 2021
17	Test Entire Project	Mar 31, 2021	Apr 10, 2021	Apr 12, 2021
18	Prepare Presentation	Apr 11, 2021	Apr 19, 2021	Apr 19, 2021
19	Presentation	Apr 20, 2021	Apr 20, 2021	Apr 20, 2021

Figure 6-1 Schedule Gantt Chart

7 PROJECT SUMMARY AND CONCLUSIONS

This document was reviewed by all team member to ensure the research and design is original and meet all requirements for this course. This paper sets the foundation for the next step, building and testing the system in the Senior Design 2. We believe after all effort on researching and finalizing the design on Senior Design 1, we will complete the design successfully and deliver the best outcome with time given. Coronavirus Prevention System is AI based, face mask detection, that uses streaming data from the camera, combined with machine learning techniques to detect a person has not wearing a mask. The customer will experience with user-friendly set up of the system, does not require human intervention to activate the system, since it operates automatically. Once a person has been detected violating not wearing a mask, Jetson Nano will send signal to microcontroller, which allows the robot open face mask tray, so the user could take a mask from it. This system can be deployed in hospitals, schools, public places, etc.

With the team of three computer engineering students and one electrical engineering student, we believe this project would be a great way to demonstrate our skills and knowledge that we have learn from coursework at UCF including hardware design and software design. Everyone in the team has opportunities to work on parts that they are interested in or has experience with. This project is more difficult than expected. The hard part is not the design, but rather, the implementation and testing. For hardware, both time and location are major constraints that we did not account for. Initially, we thought that we will be able to build and test our circuit at home. However, that is not that case. At home, we lack equipment and materials needed to solder, build, and test the set up. We were only able to create a schematic on Multisim and tested that at home. We then planned on going to the lab 3 times to test the components, build, and do load test on the circuit. We were only able to accomplish roughly 2/5 of the objectives as we were only granted access once.

As for the hardware side, the model training part was the most gruesome. We thought that it would only take 2 weeks maximum to design and train the model. This was not the case as by now, it had been almost 2 months and the model has not finished training.

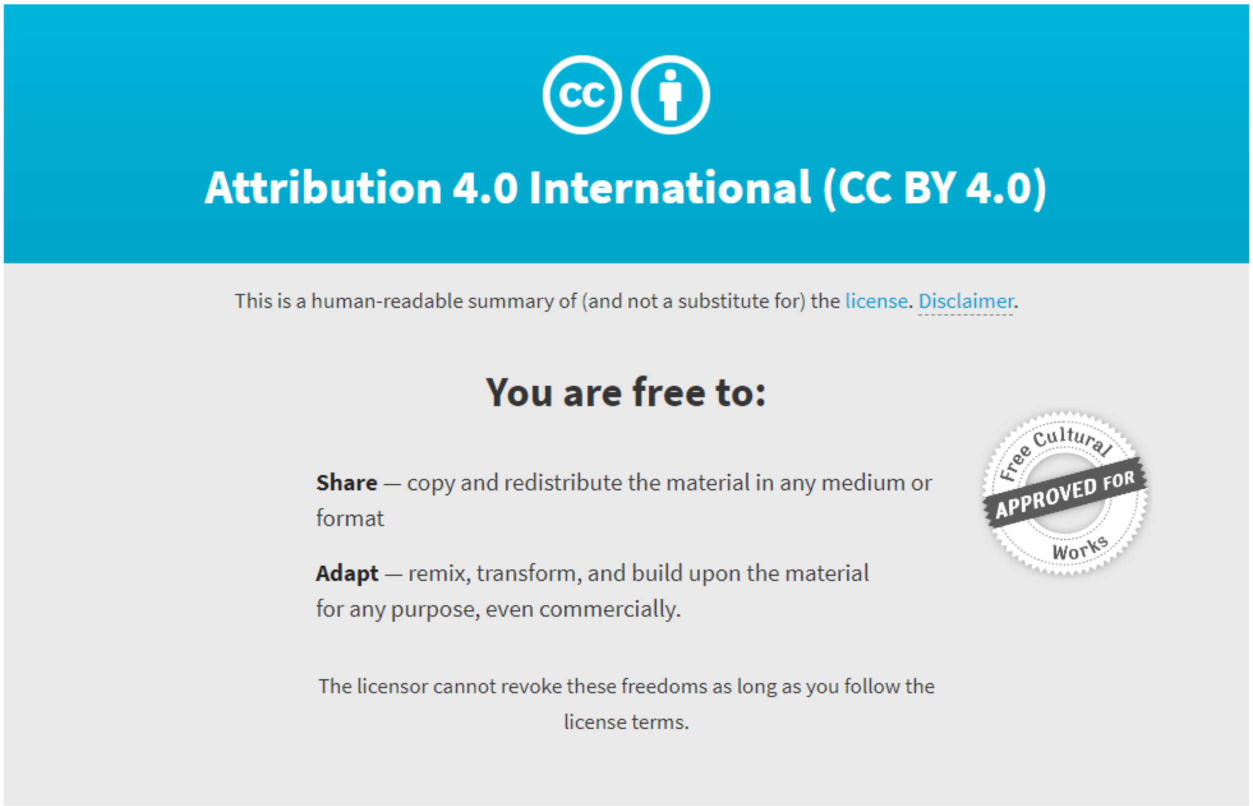
Aside from the ML model, the delay in the shipment of the first PCB for almost 2 months and that the PCB came in defective was the most devastating thing for us. We thought we would not be able to finish the project on time and would need to ask for an extension. Luckily, the manufacturing process from JLCPCB was much faster, so we were able to get our final PCB on time for the presentation.

Overall, we hope to learn valuable skills that not only for individual growth but also where everyone’s contribution can lead to something that is socially beneficial, during the current pandemic COVID-19.

8 APPENDICES

8.1 Copyrights Permissions

Permission the use of CDC materials



Attribution 4.0 International (CC BY 4.0)

This is a human-readable summary of (and not a substitute for) the license. Disclaimer.

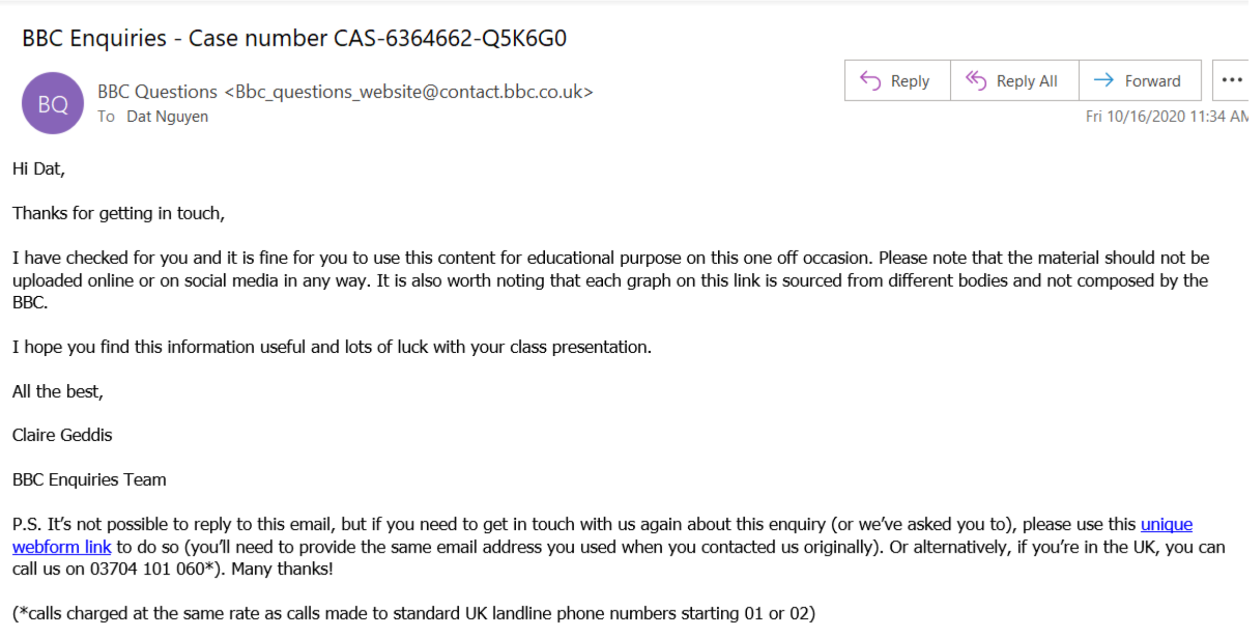
You are free to:

Share — copy and redistribute the material in any medium or format

Adapt — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Permission for BBC contents



BBC Enquiries - Case number CAS-6364662-Q5K6G0

BBC Questions <Bbc_questions_website@contact.bbc.co.uk>
To Dat Nguyen

Fri 10/16/2020 11:34 AM

Hi Dat,

Thanks for getting in touch,

I have checked for you and it is fine for you to use this content for educational purpose on this one off occasion. Please note that the material should not be uploaded online or on social media in any way. It is also worth noting that each graph on this link is sourced from different bodies and not composed by the BBC.

I hope you find this information useful and lots of luck with your class presentation.

All the best,

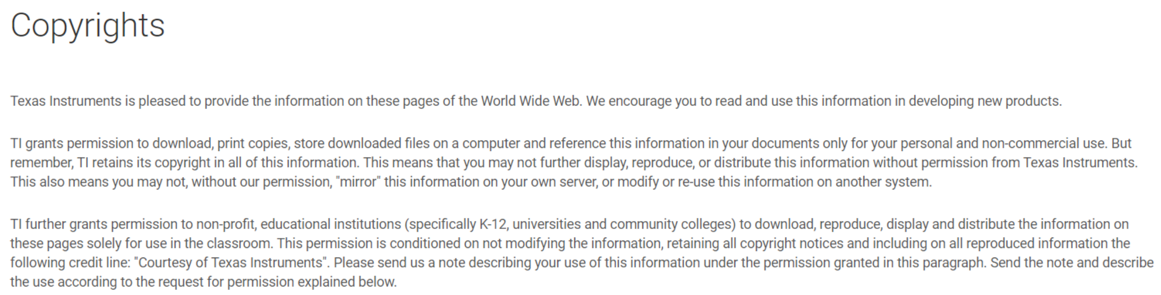
Claire Geddis

BBC Enquiries Team

P.S. It's not possible to reply to this email, but if you need to get in touch with us again about this enquiry (or we've asked you to), please use this unique webform link to do so (you'll need to provide the same email address you used when you contacted us originally). Or alternatively, if you're in the UK, you can call us on 03704 101 060*). Many thanks!

(*calls charged at the same rate as calls made to standard UK landline phone numbers starting 01 or 02)

Permission for Texas Instrument images



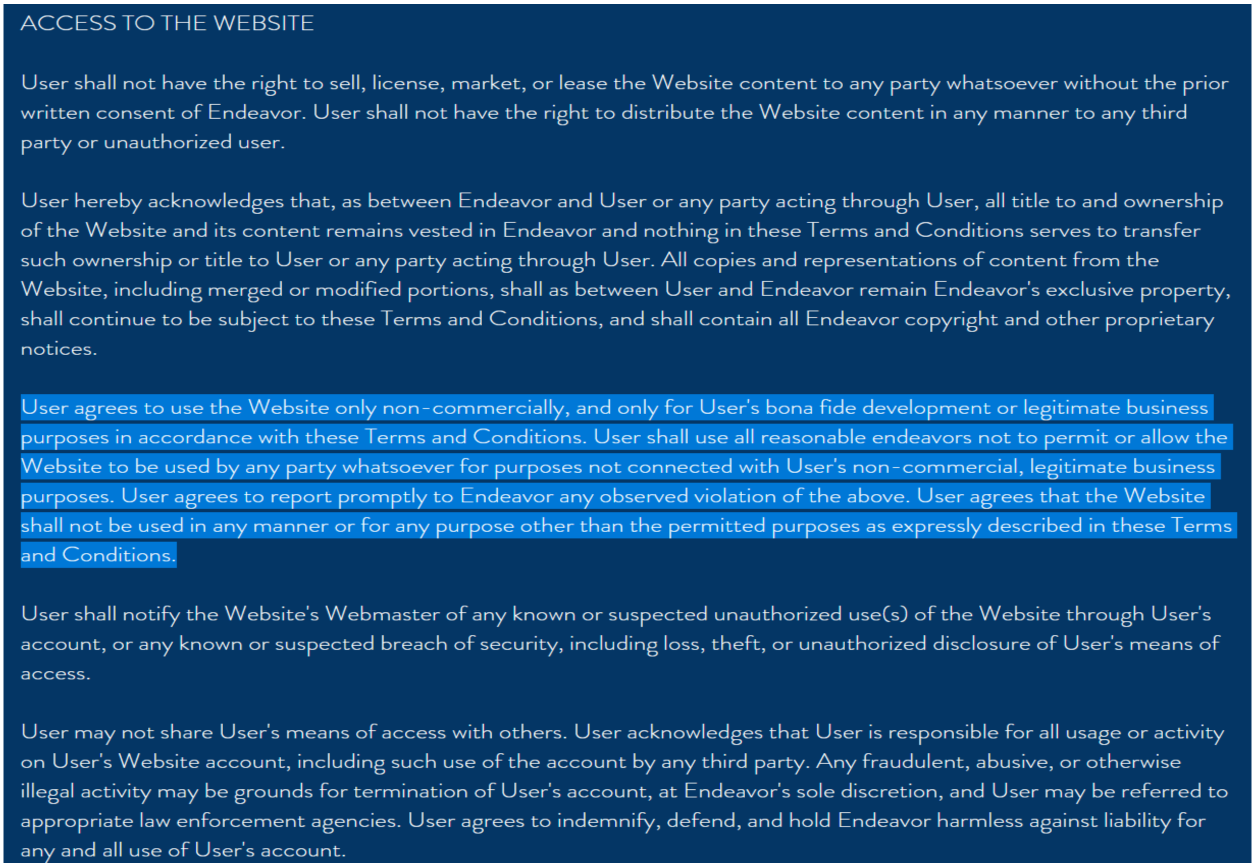
Copyrights

Texas Instruments is pleased to provide the information on these pages of the World Wide Web. We encourage you to read and use this information in developing new products.

TI grants permission to download, print copies, store downloaded files on a computer and reference this information in your documents only for your personal and non-commercial use. But remember, TI retains its copyright in all of this information. This means that you may not further display, reproduce, or distribute this information without permission from Texas Instruments. This also means you may not, without our permission, "mirror" this information on your own server, or modify or re-use this information on another system.

TI further grants permission to non-profit, educational institutions (specifically K-12, universities and community colleges) to download, reproduce, display and distribute the information on these pages solely for use in the classroom. This permission is conditioned on not modifying the information, retaining all copyright notices and including on all reproduced information the following credit line: "Courtesy of Texas Instruments". Please send us a note describing your use of this information under the permission granted in this paragraph. Send the note and describe the use according to the request for permission explained below.

Electronic design permission:



ACCESS TO THE WEBSITE

User shall not have the right to sell, license, market, or lease the Website content to any party whatsoever without the prior written consent of Endeavor. User shall not have the right to distribute the Website content in any manner to any third party or unauthorized user.

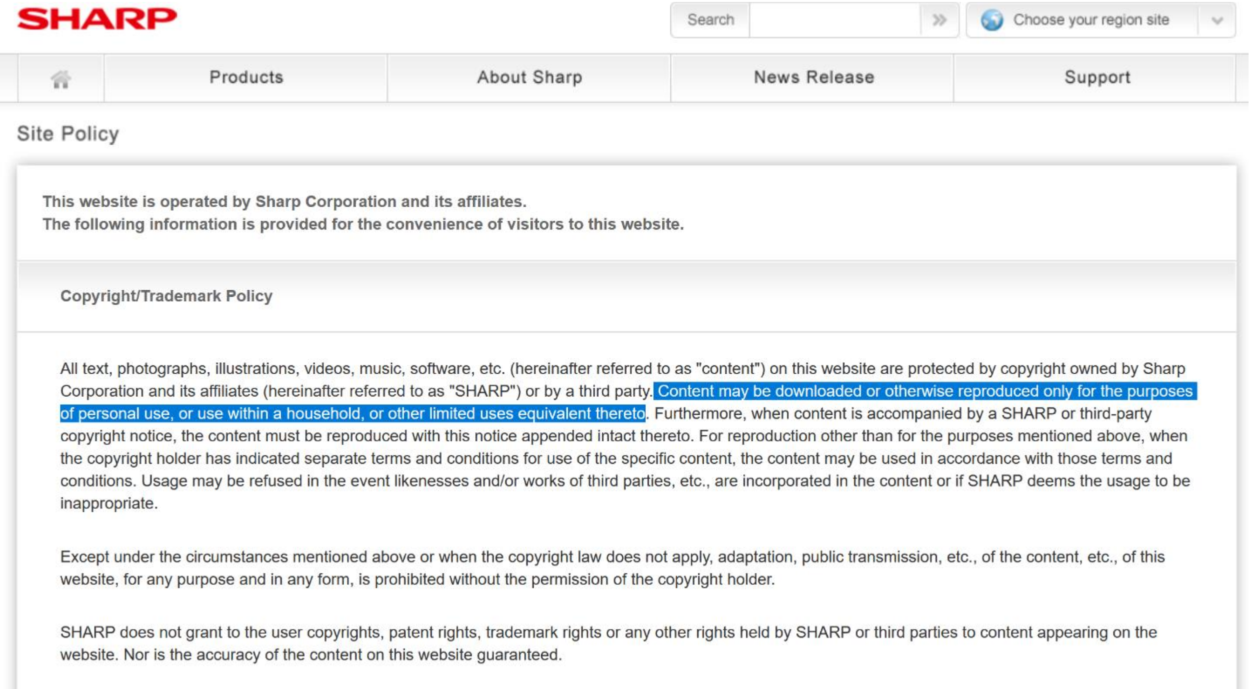
User hereby acknowledges that, as between Endeavor and User or any party acting through User, all title to and ownership of the Website and its content remains vested in Endeavor and nothing in these Terms and Conditions serves to transfer such ownership or title to User or any party acting through User. All copies and representations of content from the Website, including merged or modified portions, shall as between User and Endeavor remain Endeavor's exclusive property, shall continue to be subject to these Terms and Conditions, and shall contain all Endeavor copyright and other proprietary notices.

User agrees to use the Website only non-commercially, and only for User's bona fide development or legitimate business purposes in accordance with these Terms and Conditions. User shall use all reasonable endeavors not to permit or allow the Website to be used by any party whatsoever for purposes not connected with User's non-commercial, legitimate business purposes. User agrees to report promptly to Endeavor any observed violation of the above. User agrees that the Website shall not be used in any manner or for any purpose other than the permitted purposes as expressly described in these Terms and Conditions.

User shall notify the Website's Webmaster of any known or suspected unauthorized use(s) of the Website through User's account, or any known or suspected breach of security, including loss, theft, or unauthorized disclosure of User's means of access.

User may not share User's means of access with others. User acknowledges that User is responsible for all usage or activity on User's Website account, including such use of the account by any third party. Any fraudulent, abusive, or otherwise illegal activity may be grounds for termination of User's account, at Endeavor's sole discretion, and User may be referred to appropriate law enforcement agencies. User agrees to indemnify, defend, and hold Endeavor harmless against liability for any and all use of User's account.

SHARP permission:



Site Policy

This website is operated by Sharp Corporation and its affiliates.
The following information is provided for the convenience of visitors to this website.

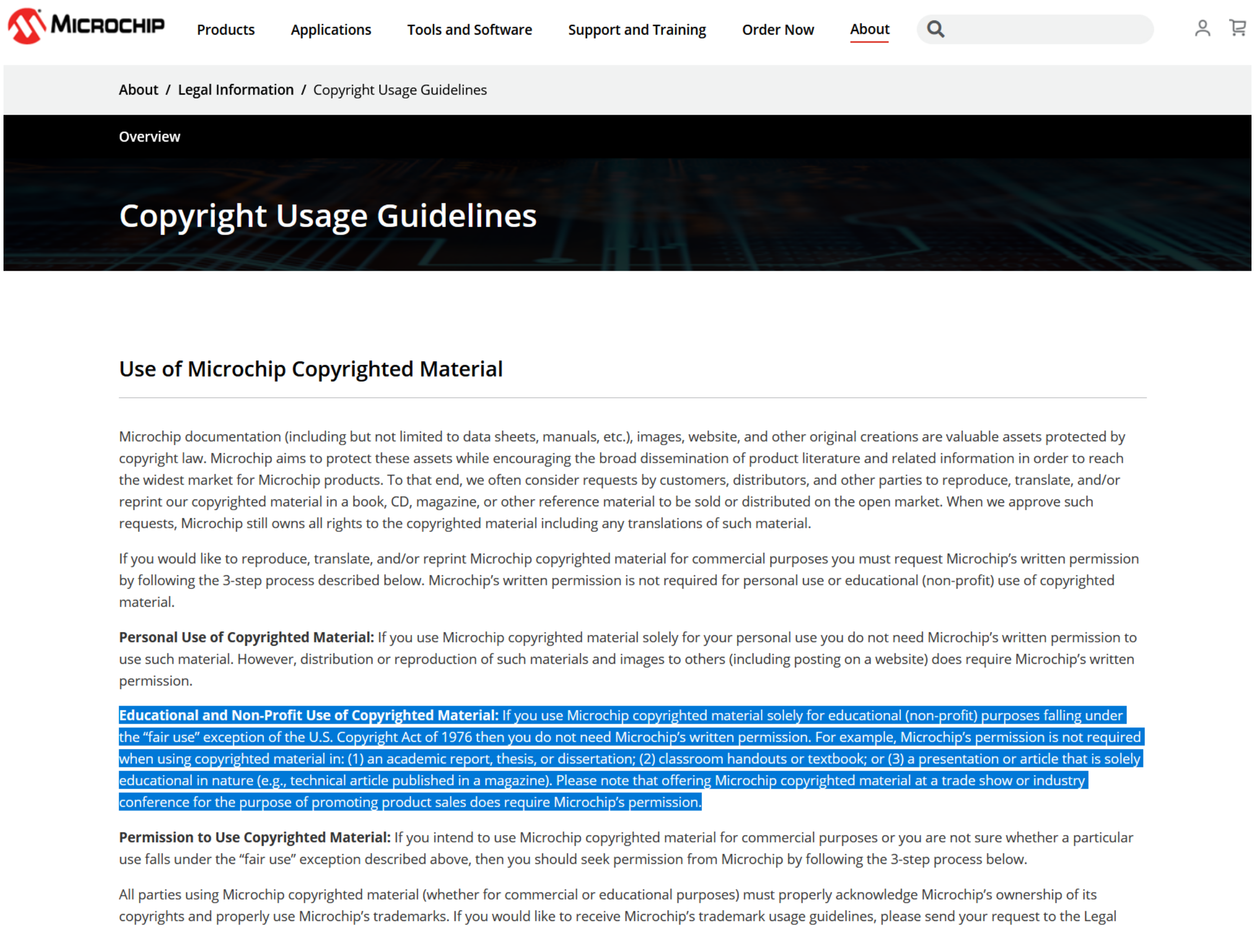
Copyright/Trademark Policy

All text, photographs, illustrations, videos, music, software, etc. (hereinafter referred to as "content") on this website are protected by copyright owned by Sharp Corporation and its affiliates (hereinafter referred to as "SHARP") or by a third party. Content may be downloaded or otherwise reproduced only for the purposes of personal use, or use within a household, or other limited uses equivalent thereto. Furthermore, when content is accompanied by a SHARP or third-party copyright notice, the content must be reproduced with this notice appended intact thereto. For reproduction other than for the purposes mentioned above, when the copyright holder has indicated separate terms and conditions for use of the specific content, the content may be used in accordance with those terms and conditions. Usage may be refused in the event likenesses and/or works of third parties, etc., are incorporated in the content or if SHARP deems the usage to be inappropriate.

Except under the circumstances mentioned above or when the copyright law does not apply, adaptation, public transmission, etc., of the content, etc., of this website, for any purpose and in any form, is prohibited without the permission of the copyright holder.

SHARP does not grant to the user copyrights, patent rights, trademark rights or any other rights held by SHARP or third parties to content appearing on the website. Nor is the accuracy of the content on this website guaranteed.

Microchip permission:



Copyright Usage Guidelines

Use of Microchip Copyrighted Material

Microchip documentation (including but not limited to data sheets, manuals, etc.), images, website, and other original creations are valuable assets protected by copyright law. Microchip aims to protect these assets while encouraging the broad dissemination of product literature and related information in order to reach the widest market for Microchip products. To that end, we often consider requests by customers, distributors, and other parties to reproduce, translate, and/or reprint our copyrighted material in a book, CD, magazine, or other reference material to be sold or distributed on the open market. When we approve such requests, Microchip still owns all rights to the copyrighted material including any translations of such material.

If you would like to reproduce, translate, and/or reprint Microchip copyrighted material for commercial purposes you must request Microchip's written permission by following the 3-step process described below. Microchip's written permission is not required for personal use or educational (non-profit) use of copyrighted material.

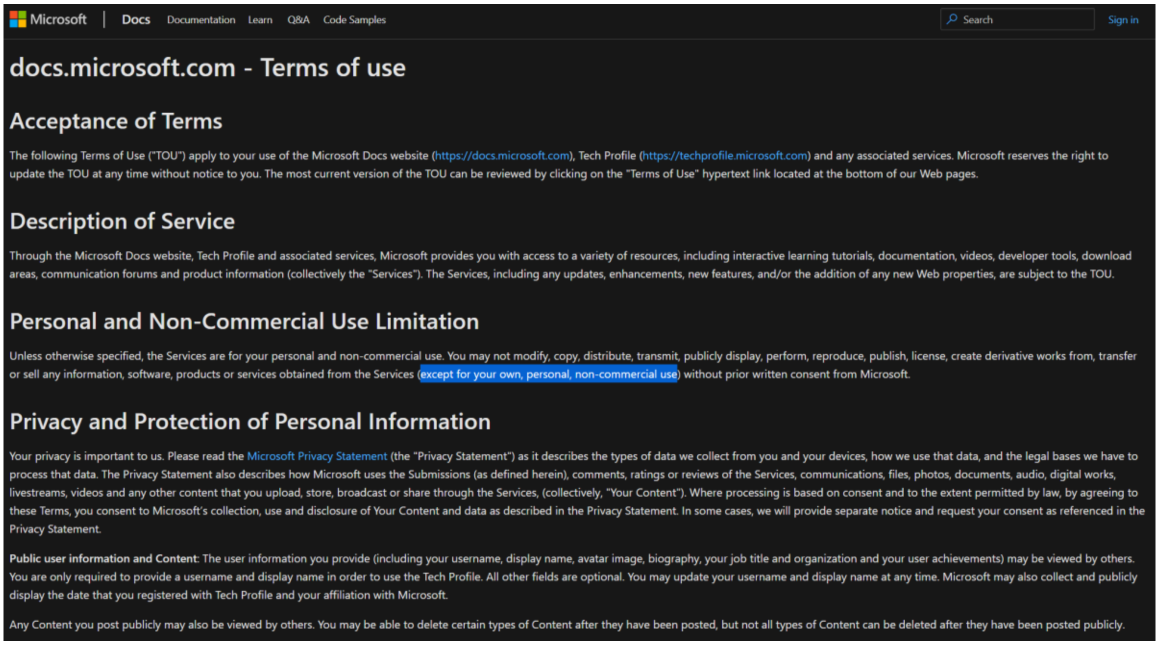
Personal Use of Copyrighted Material: If you use Microchip copyrighted material solely for your personal use you do not need Microchip's written permission to use such material. However, distribution or reproduction of such materials and images to others (including posting on a website) does require Microchip's written permission.

Educational and Non-Profit Use of Copyrighted Material: If you use Microchip copyrighted material solely for educational (non-profit) purposes falling under the "fair use" exception of the U.S. Copyright Act of 1976 then you do not need Microchip's written permission. For example, Microchip's permission is not required when using copyrighted material in: (1) an academic report, thesis, or dissertation; (2) classroom handouts or textbook; or (3) a presentation or article that is solely educational in nature (e.g., technical article published in a magazine). Please note that offering Microchip copyrighted material at a trade show or industry conference for the purpose of promoting product sales does require Microchip's permission.

Permission to Use Copyrighted Material: If you intend to use Microchip copyrighted material for commercial purposes or you are not sure whether a particular use falls under the "fair use" exception described above, then you should seek permission from Microchip by following the 3-step process below.

All parties using Microchip copyrighted material (whether for commercial or educational purposes) must properly acknowledge Microchip's ownership of its copyrights and properly use Microchip's trademarks. If you would like to receive Microchip's trademark usage guidelines, please send your request to the Legal

Microsoft terms of use:



docs.microsoft.com - Terms of use

Acceptance of Terms

The following Terms of Use ("TOU") apply to your use of the Microsoft Docs website (https://docs.microsoft.com), Tech Profile (https://techprofile.microsoft.com) and any associated services. Microsoft reserves the right to update the TOU at any time without notice to you. The most current version of the TOU can be reviewed by clicking on the "Terms of Use" hypertext link located at the bottom of our Web pages.

Description of Service

Through the Microsoft Docs website, Tech Profile and associated services, Microsoft provides you with access to a variety of resources, including interactive learning tutorials, documentation, videos, developer tools, download areas, communication forums and product information (collectively the "Services"). The Services, including any updates, enhancements, new features, and/or the addition of any new Web properties, are subject to the TOU.

Personal and Non-Commercial Use Limitation

Unless otherwise specified, the Services are for your personal and non-commercial use. You may not modify, copy, distribute, transmit, publicly display, perform, reproduce, publish, license, create derivative works from, transfer or sell any information, software, products or services obtained from the Services (except for your own, personal, non-commercial use) without prior written consent from Microsoft.

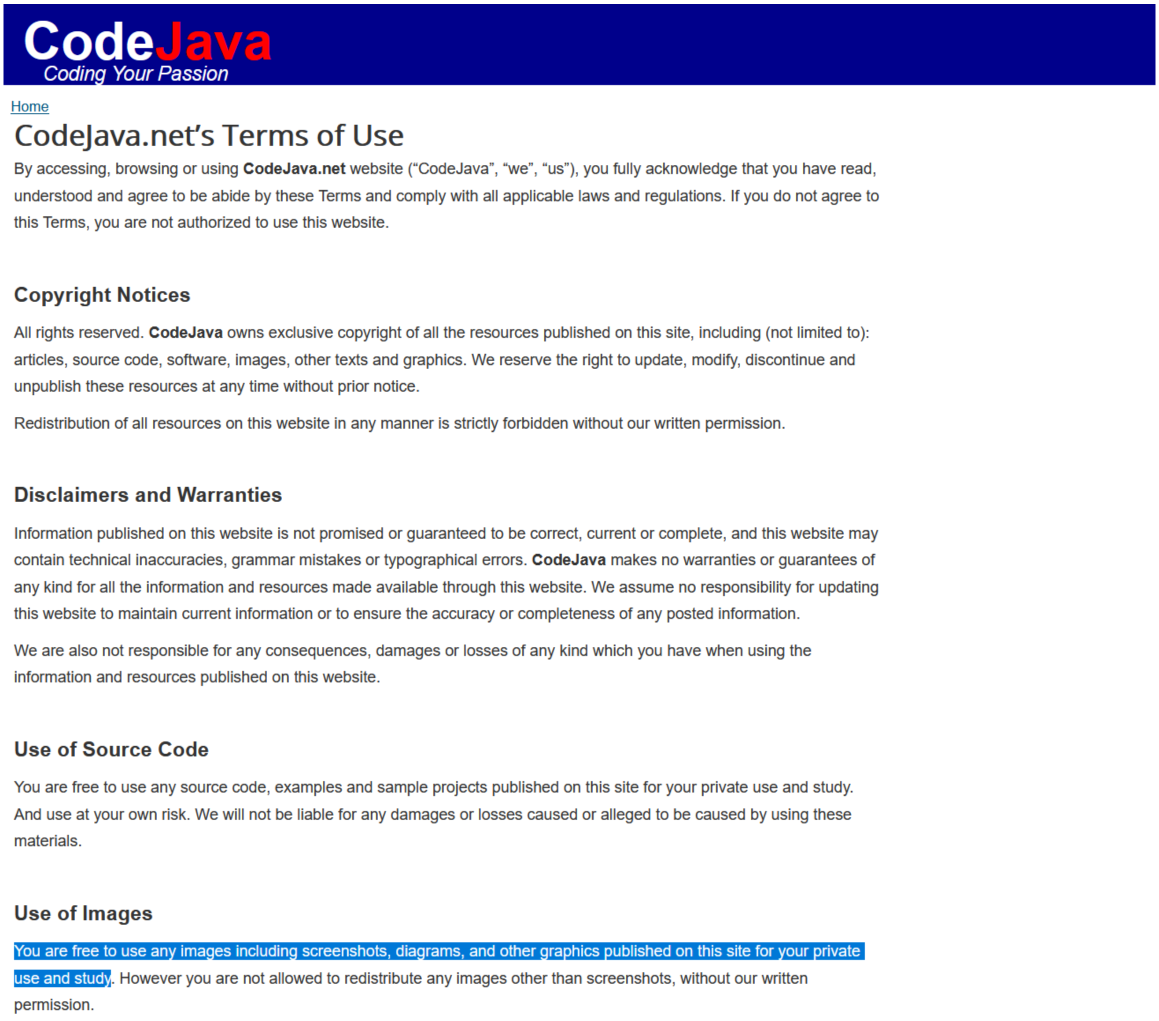
Privacy and Protection of Personal Information

Your privacy is important to us. Please read the Microsoft Privacy Statement (the "Privacy Statement") as it describes the types of data we collect from you and your devices, how we use that data, and the legal bases we have to process that data. The Privacy Statement also describes how Microsoft uses the Submissions (as defined herein), comments, ratings or reviews of the Services, communications, files, photos, documents, audio, digital works, livestreams, videos and any other content that you upload, store, broadcast or share through the Services, (collectively, "Your Content"). Where processing is based on consent and to the extent permitted by law, by agreeing to these Terms, you consent to Microsoft's collection, use and disclosure of Your Content and data as described in the Privacy Statement. In some cases, we will provide separate notice and request your consent as referenced in the Privacy Statement.

Public user information and Content: The user information you provide (including your username, display name, avatar image, biography, your job title and organization and your user achievements) may be viewed by others. You are only required to provide a username and display name in order to use the Tech Profile. All other fields are optional. You may update your username and display name at any time. Microsoft may also collect and publicly display the date that you registered with Tech Profile and your affiliation with Microsoft.

Any Content you post publicly may also be viewed by others. You may be able to delete certain types of Content after they have been posted, but not all types of Content can be deleted after they have been posted publicly.

Permission from CodeJava to use Relational Database graphics



CodeJava.net’s Terms of Use

By accessing, browsing or using **CodeJava.net** website (“CodeJava”, “we”, “us”), you fully acknowledge that you have read, understood and agree to be abide by these Terms and comply with all applicable laws and regulations. If you do not agree to this Terms, you are not authorized to use this website.

Copyright Notices

All rights reserved. **CodeJava** owns exclusive copyright of all the resources published on this site, including (not limited to): articles, source code, software, images, other texts and graphics. We reserve the right to update, modify, discontinue and unpublish these resources at any time without prior notice.

Redistribution of all resources on this website in any manner is strictly forbidden without our written permission.

Disclaimers and Warranties

Information published on this website is not promised or guaranteed to be correct, current or complete, and this website may contain technical inaccuracies, grammar mistakes or typographical errors. **CodeJava** makes no warranties or guarantees of any kind for all the information and resources made available through this website. We assume no responsibility for updating this website to maintain current information or to ensure the accuracy or completeness of any posted information.

We are also not responsible for any consequences, damages or losses of any kind which you have when using the information and resources published on this website.

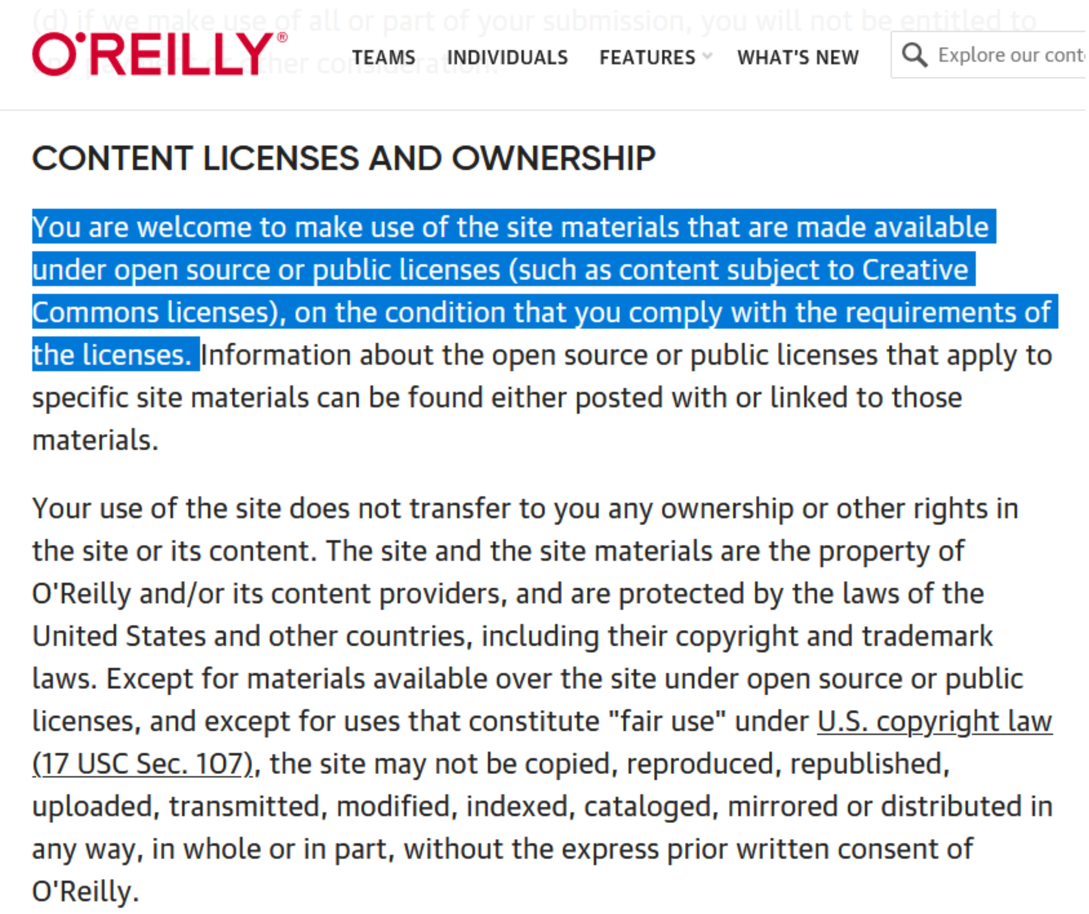
Use of Source Code

You are free to use any source code, examples and sample projects published on this site for your private use and study. And use at your own risk. We will not be liable for any damages or losses caused or alleged to be caused by using these materials.

Use of Images

You are free to use any images including screenshots, diagrams, and other graphics published on this site for your private use and study. However you are not allowed to redistribute any images other than screenshots, without our written permission.

O’Reilly term of service for the use of MySQL Module chart



(d) if we make use of all or part of your submission, you will not be entitled to

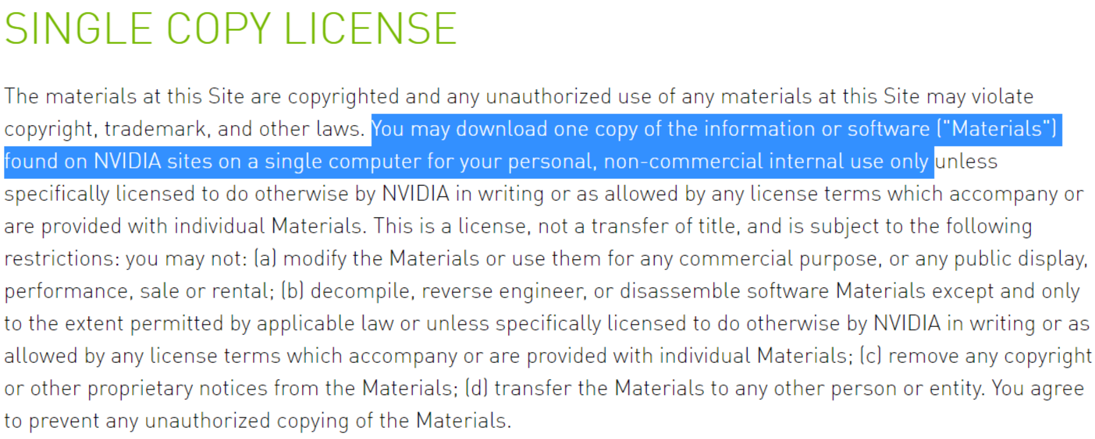
ther consideration.

CONTENT LICENSES AND OWNERSHIP

You are welcome to make use of the site materials that are made available under open source or public licenses (such as content subject to Creative Commons licenses), on the condition that you comply with the requirements of the licenses. Information about the open source or public licenses that apply to specific site materials can be found either posted with or linked to those materials.

Your use of the site does not transfer to you any ownership or other rights in the site or its content. The site and the site materials are the property of O'Reilly and/or its content providers, and are protected by the laws of the United States and other countries, including their copyright and trademark laws. Except for materials available over the site under open source or public licenses, and except for uses that constitute "fair use" under U.S. copyright law (17 USC Sec. 107), the site may not be copied, reproduced, republished, uploaded, transmitted, modified, indexed, cataloged, mirrored or distributed in any way, in whole or in part, without the express prior written consent of O'Reilly.

NVIDIA terms of use:



SINGLE COPY LICENSE

The materials at this Site are copyrighted and any unauthorized use of any materials at this Site may violate copyright, trademark, and other laws. You may download one copy of the information or software ("Materials") found on NVIDIA sites on a single computer for your personal, non-commercial internal use only unless specifically licensed to do otherwise by NVIDIA in writing or as allowed by any license terms which accompany or are provided with individual Materials. This is a license, not a transfer of title, and is subject to the following restrictions: you may not: (a) modify the Materials or use them for any commercial purpose, or any public display, performance, sale or rental; (b) decompile, reverse engineer, or disassemble software Materials except and only to the extent permitted by applicable law or unless specifically licensed to do otherwise by NVIDIA in writing or as allowed by any license terms which accompany or are provided with individual Materials; (c) remove any copyright or other proprietary notices from the Materials; (d) transfer the Materials to any other person or entity. You agree to prevent any unauthorized copying of the Materials.

8.2 References

[1] "About COVID-19." *Centers for Disease Control and Prevention*, Centers for Disease Control and Prevention, www.cdc.gov/coronavirus/2019-ncov/cdcresponse/about-COVID-19.html.

[2] Analog Devices, Inc. “Understanding How a Voltage Regulator Works.” *Understanding How a Voltage Regulator Works | Analog Devices*, 2009, www.analog.com/en/technical-articles/how-voltage-regulator-works.html.

[3] Archiveddocs. “Features (SQL Server Compact).” Microsoft Docs, docs.microsoft.com/en-us/previous-versions/sql/compact/sql-server-compact-4.0/ms172037(v=sql.110).

[4] Bellis, Mary. “The Incredible Reason the Vending Machine Was Created.” *ThoughtCo*, 2019, www.thoughtco.com/the-history-of-vending-machines-1992599.

[5] Bhargava, Hansa D. “Coronavirus History: How Did Coronavirus Start?” *WebMD*, WebMD, 15 Apr. 2020, www.webmd.com/lung/coronavirus-history.

[6] Burke, Kathy, et al. “Wavelength.” *Amazon*, Longman, 2000, aws.amazon.com/wavelength/.

[7] Choudhury, Ambika. “Top Databases Used In Machine Learning Projects.” *Analytics India Magazine*, 21 Nov. 2020, analyticsindiamag.com/top-databases-used-in-machine-learning-projects/.

[8] Choudhury, Ambika. “Top Databases Used In Machine Learning Projects.” *Analytics India Magazine*, 21 Nov. 2020, analyticsindiamag.com/top-databases-used-in-machine-learning-projects/.

[9] “Clinical Trial Phases.” *GSK*, www.gsk.com/en-gb/research-and-development/trials-in-people/clinical-trial-phases/.

[10] Costan, Victor, and Srinivas Devadas. “Intel SGX Explained.” 2016, eprint.iacr.org/2016/086.pdf.

[11] “Coronavirus Cases:” *Worldometer*, www.worldometers.info/coronavirus/.

[12] “Cost-Optimized and High-Performance Database.” *Oracle*, www.oracle.com/database/.

[13] Daware, Kiran. “Please Enable Cookies.” *StackPath*, 2020, www.electronicdesign.com/power-management/whitepaper/21140276/what-is-a-stepdown-transformer.

[14] Dries Buytaert. The History of MySQL AB, Dries Buytaert, 31 Oct. 2017, dri.es/the-history-of-mysql-ab.

[15] Engdahl, Sylvia. “Blogs.” *Amazon*, Greenhaven Press/Gale, 2008, aws.amazon.com/blogs/machine-learning/find-distinct-people-in-a-video-with-amazon-rekognition/.

[16] Gabler, Ellen, and Michael H. Keller. “Prescriptions Surged as Trump Praised Drugs in Coronavirus Fight.” *The New York Times*, The New York Times, 25 Apr. 2020, www.nytimes.com/2020/04/25/us/coronavirus-trump-chloroquine-hydroxychloroquine.html.

[17] Geitgey, Adam. “Build a Hardware-Based Face Recognition System for $150 with the Nvidia Jetson Nano and Python.” *Medium*, Medium, 25 June 2019, medium.com/@ageitgey/build-a-hardware-based-face-recognition-system-for-150-with-the-nvidia-jetson-nano-and-python-a25cb8c891fd/.

[18] Gupta, Sourav. “AC to DC Converter Circuit.” *Circuit Digest*, 19 Feb. 2019, circuitdigest.com/electronic-circuits/ac-to-dc-converter-circuit-diagram/.

[19] Han, Song, et al. “Learning Both Weights and Connections for Efficient Neural Networks.” *ArXiv.org*, 30 Oct. 2015, arxiv.org/abs/1506.02626.

[20] “How Do Thermal Cameras Work?” *FLIR Systems*, www.flir.com/discover/rd-science/how-do-thermal-cameras-work/.

[21] “JetPack SDK.” *NVIDIA Developer*, 21 Oct. 2020, developer.nvidia.com/embedded/jetpack.

[22] “Jetson Nano Developer Kit.” *NVIDIA Developer*, 9 Oct. 2020, developer.nvidia.com/embedded/jetson-nano-developer-kit.

[23] Kulkarni, Amey, et al. “Implementing a Real-Time, AI-Based, Face Mask Detector Application for COVID-19.” *NVIDIA Developer Blog*, 13 Oct. 2020, developer.nvidia.com/blog/implementing-a-real-time-ai-based-face-mask-detector-application-for-covid-19/.

[24] Kwong, Andrew, et al. “RAMBleed: Reading Bits in Memory Without Accessing Them.” Annual IEEE Symposium on Security &amp; Privacy, no. 41, May 2020, rambleed.com/docs/20190603-rambleed-web.pdf.

[25] Lora Jones, Daniele Palumbo & David Brown. “Coronavirus: A Visual Guide to the Economic Impact.” *BBC News*, BBC, 29 June 2020, www.bbc.com/news/business-51706225.

[26] Liu, Fangfei, et al. “Last-Level Cache Side-Channel Attacks Are Practical.” IEEE Symposium on Security and Privacy, 2015, palms.ee.princeton.edu/system/files/SP_vfinal.pdf.

[27] Microchip Technology Inc. MegaAVR® Data Sheet. 2020, ww1.microchip.com/downloads/en/DeviceDoc/ATmega48A-PA-88A-PA-168A-PA-328-P-DS-DS40002061B.pdf.

[28] “NVIDIA DeepStream SDK.” *NVIDIA Developer*, 13 Nov. 2020, developer.nvidia.com/deepstream-sdk.

[29] “NVIDIA Transfer Learning Toolkit.” *NVIDIA Developer*, 13 Nov. 2020, developer.nvidia.com/transfer-learning-toolkit.

[30] “NVIDIA Transfer Learning Toolkit.” *NVIDIA Developer*, 13 Nov. 2020, developer.nvidia.com/transfer-learning-toolkit.

[31] Pachev, Sasha. “Understanding MySQL Internals.” *O'Reilly Online Learning*, O'Reilly Media, Inc., www.oreilly.com/library/view/understanding-mysql-internals/0596009577/ch01.html.

[32] Raspberry Pi. “Raspberry Pi 4 Model B Specifications.” *Raspberry Pi*, www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/?resellerType=home.

[33] Rouse, Margaret. “What Is a Microcontroller and How Does It Work?” *IoT Agenda*, TechTarget, 7 Nov. 2019, internetofthingsagenda.techtarget.com/definition/microcontroller.

[34] Sattel, Sam. “What Is a Voltage Regulator?: EAGLE: Blog.” *Eagle Blog*, 25 Aug. 2017, www.autodesk.com/products/eagle/blog/what-is-a-voltage-regulator/.

[35] Schimelpfening, Nancy. “How Long Does Immunity Last After COVID-19? What We Know.” *Healthline*, Healthline Media, 14 Oct. 2020, www.healthline.com/health-news/how-long-does-immunity-last-after-covid-19-what-we-know.

[36] Schroff, Florian, et al. “FaceNet: A Unified Embedding for Face Recognition and Clustering.” *ArXiv.org*, 17 June 2015, arxiv.org/abs/1503.03832.

[37] “Scientific Brief: SARS-CoV-2 and Potential Airborne Transmission.” *Centers for Disease Control and Prevention*, Centers for Disease Control and Prevention, www.cdc.gov/coronavirus/2019-ncov/more/scientific-brief-sars-cov-2.html.

[38] Simic, Sofija. “What Is a LAMP Stack?” Knowledge Base by PhoenixNAP, 22 Sept. 2020, phoenixnap.com/kb/what-is-a-lamp-stack.

[39] Switching Regulator Fundamentals. Texas Instruments Incorporated, 2012, www.ti.com/lit/snva559.

[40] “Types of Distance Sensor and How to Select One?” *Latest Open Tech from Seeed Studio*, 25 Dec. 2019, www.seeedstudio.com/blog/2019/12/23/distance-sensors-types-and-selection-guide/.

[41] “Temperature Sensor Types for Temperature Measurement.” *Basic Electronics Tutorials*, 11 Feb. 2018, www.electronics-tutorials.ws/io/io_3.html.

[42] Texas Instruments Incorporated. MSP430G2553. 2011, www.ti.com/product/MSP430G2553.

[43] “Tinker Board S: Tinker Board.” ASUS USA, 2018, www.asus.com/us/Single-Board-Computer/Tinker-Board-S/overview/.

[44] Trujillo, Marco, and Ingrid Melander. “Europe's Worst Infection Hotspot Madrid Heads for Lockdown.” *Reuters*, Thomson Reuters, 1 Oct. 2020, www.reuters.com/article/us-health-coronavirus-spain-madrid/europes-worst-infection-hotspot-madrid-heads-for-lockdown-idUSKBN26M5CK.

[45] *Types of Databases*, www.tutorialspoint.com/Types-of-databases.

[46] “What Is a GPU? Graphics Processing Units Defined.” *Intel*, www.intel.com/content/www/us/en/products/docs/processors/what-is-a-gpu.html.

[47] William, Juan, et al. “Face Recognition Using FaceNet (Survey, Performance Test, and Comparison).” IEEE, 2015.

[48] “Working Age Population: Aged 15-64: All for the United States.” *Working Age Population: Aged 15-64: All for the United States - 1977-2020 Data*, tradingeconomics.com/united-states/working-age-population-aged-15-64-all-persons-for-the-united-states-fed-data.html.

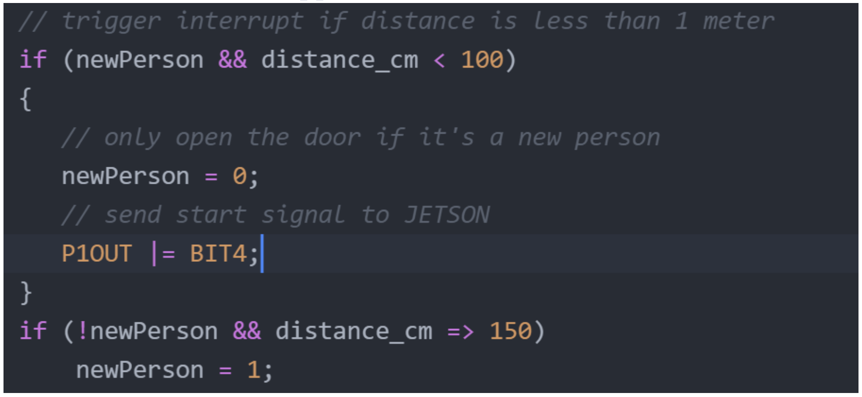
8.3 Software

8.3.1 Jetson Nano

8.3.1.1 Wrapper Software Pseudocode

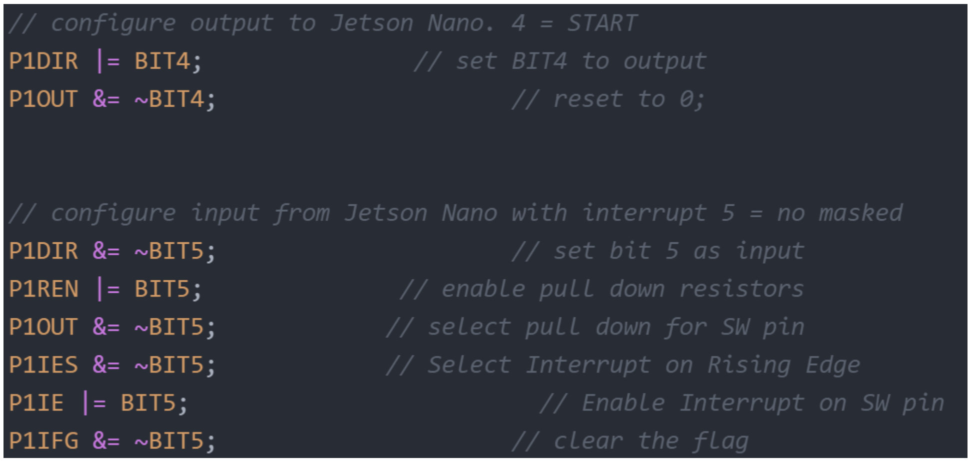
```
Boot
Begin in sleep
Loop forever
    One-byte blocking read from the serial port in “sleep”
    Once byte is read, check if it equals ‘S’. If not, restart loop. Else, proceed
    Run DeepStream inference on 128 images from the camera
    Collect analytics on images, highest # of masks/no-masks seen in frames
    If (person with no mask detected in inference)
        Toggle GPIO to MCU
    Write analytics to SQL DB along with timestamp
    Go back to sleep
End Loop
```

8.3.2 MSP430G2553’s Code Snippets



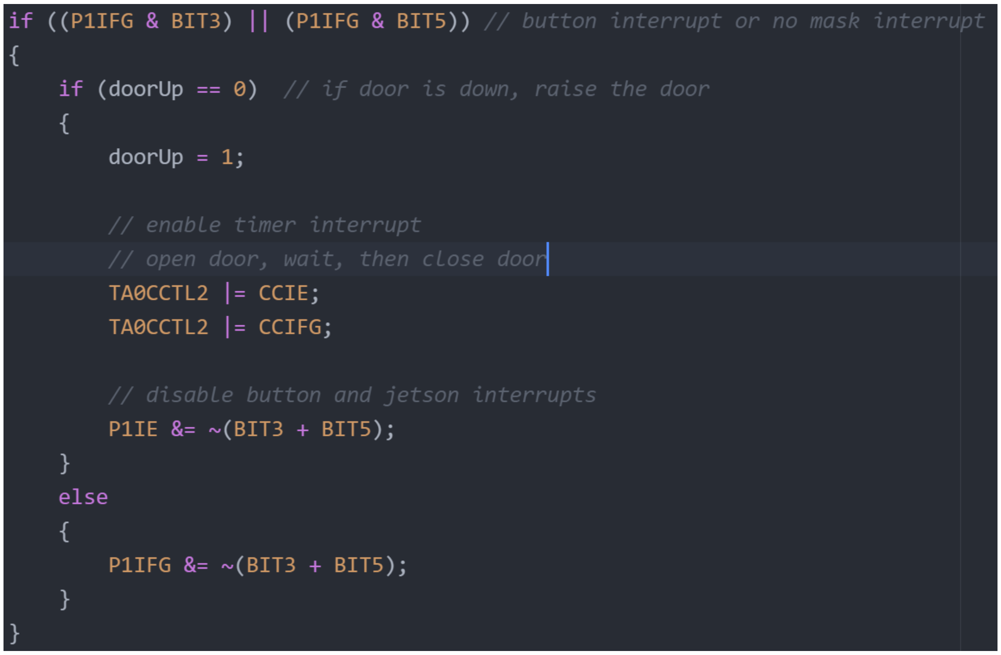
```
// trigger interrupt if distance is less than 1 meter
if (newPerson && distance_cm < 100)
{
   // only open the door if it's a new person
   newPerson = 0;
   // send start signal to JETSON
   P1OUT |= BIT4;
}
if (!newPerson && distance_cm => 150)
    newPerson = 1;
```

Figure 8-1 MSP430G2553 Distance Interrupt



```
// configure output to Jetson Nano. 4 = START
P1DIR |= BIT4;                // set BIT4 to output
P1OUT &= ~BIT4;                      // reset to 0;


// configure input from Jetson Nano with interrupt 5 = no masked
P1DIR &= ~BIT5;                      // set bit 5 as input
P1REN |= BIT5;               // enable pull down resistors
P1OUT &= ~BIT5;             // select pull down for SW pin
P1IES &= ~BIT5;             // Select Interrupt on Rising Edge
P1IE |= BIT5;                          // Enable Interrupt on SW pin
P1IFG &= ~BIT5;                      // clear the flag
```

Figure 8-2 MSP430G2553 Computation Status Interrupts Configuration



```
if ((P1IFG & BIT3) || (P1IFG & BIT5)) // button interrupt or no mask interrupt
{
    if (doorUp == 0)  // if door is down, raise the door
    {
        doorUp = 1;

        // enable timer interrupt
        // open door, wait, then close door
        TA0CCTL2 |= CCIE;
        TA0CCTL2 |= CCIFG;

        // disable button and jetson interrupts
        P1IE &= ~(BIT3 + BIT5);
    }
    else
    {
        P1IFG &= ~(BIT3 + BIT5);
    }
}
```

Figure 8-3 MSP430G2553 No-mask ISR