DES3

Swimming Pool Drown Detection and Rescue System

FYP Final Report

By:

LEUNG Lok Yan Loreen (CPEG & MAE), CHAN Tsz Ho (MAE),

CHENG Hoi Wai (COMP), CHONG Shing Tung (CPEG), LEE Ka Hei (MAE)

DES3

Advised By:

Prof. MA, Robin Lok Wang (MAE) & Prof. TSOI, Desmond Yau-Chat (CSE)

Submitted in partial fulfilment of the requirements of

COMP 4981 or CPEG 4901 or ENGG4901

in the

Department of Computer Science and Engineering

The Hong Kong University of Science and Technology

2021-2022

Date of Submission: 20 April 2022

Abstract

Drowning is one of the main reasons of the death of teenagers, and identifying victims is difficult that requires serious concentration. Therefore, we build a system to assist the lifeguard in rescue. The system consists of a wearable device to detect drowning, an image processing system to locate victims, a monitor system for lifeguards and a shooting robot to perform an active rescue. Although the system is in the proof-of-concept stage, the system is highly modular and adaptive to different existing systems, which can fit in the existing swimming pools.

This project participated in the HKUST President's Cup 2022 and became one of the finalists. Also, the project is nominated to join the ASM Technology Award 2022 by the MAE department.

Declaration of Project Nature

This is a joint department project specially endorsed by the School of Engineering.

This crossed department project is submitted to fulfil both Mechanical and Aerospace Engineering Final Year Design Project (MECH 4900) and Computer Engineering (COMP 4981 or CPEG 4901). In addition, this project is also used to fulfil the Engineering School requirement for Integrated Final Year Project - First and Second Major (ENGG4901 and ENGG4902).

Thus, the grouping included:

Students from MECH: CHAN Tsz Ho, LEE Ka Hei, LEUNG Lok Yan Loreen

Students from CPEG: CHONG Shing Tung, LEUNG Lok Yan Loreen

Co-advised by:

Prof. TSOI, Desmond Yau-Chat (Computer Science and Engineering) & Prof. MA, Robin Lok Wang (Mechanical and Aerospace Engineering)

Executive Summary

Due to the noticeable lifeguard shortage in Hong Kong, the current lifeguards are overworked, which may cause fatigue and the lifeguard's attention to drop [1] [2] [3]. This magnified the chance of drowning without being noticed immediately.

In response to drowning, some current systems are mainly developed to alert and support the lifeguard to observe the drowning people underwater. Poseidon is one of the examples that developed to the commercial level. However, these applications focus on the alert upon identifying victims who are already drowning.

On the other hand, our Swimming Pool Drown Detection and Rescue System focuses on the entire supporting architecture. In particular, we focus on identifying pre-drown [4] victims with an active rescue system. In our system, each swimmer will be given a wearable device to measure and detect any abnormal physical conditions by monitoring blood oxygen level and heart rate. When an abnormal signal is detected, the web application will alert the lifeguard and shoot a kickboard to the victim's location under the lifeguard's attention.

Our system would further focus on the indoor 25-meter training swimming pool setting. This venue selection not only enables a better focus of the rescue system on fundamental functionality but is also chosen for being the most common type of pool in Hong Kong’s public swimming pool selected to maximize the possible usage.

Table of Contents

1. Introduction

1.1 Background

In Hong Kong, the lifeguard shortage is a yearly problem. During summer, the peak season of swimming, the frequency of insufficient lifeguards leading to the closure of swimming pools and beaches increases significantly (Local lifeguard union, 2019). In 2019, some swimming pools and beaches were closed due to a shortage of nearly half of the required number of lifeguards needed during peak season [5].

Under the pandemic, the problem of lifeguard shortage had worsened. The keen competition between the public and private sector with the opening of Water World Ocean Park exacerbated the lack of lifeguards. According to a report by the Standard in August 2021, there were no lifeguards at half of the beaches resulting in temporary close for some of them. Due to the closures, the training for newly recruited lifeguards and revalidation for qualified lifeguards were cancelled [6]. This reflects the seriousness and the impact of the lifeguard shortage.

One of the lifeguard's main responsibilities is maintaining safety. However, without enough lifeguards at the swimming pool, there will be not enough or even no people monitoring the safety of the pool users. Among the safety issues, drowning is the most severe. As shown in Figure 1, it is noteworthy that poolside drowning took up 13% of all drowning cases, with 8% in the private sector and the other 5% in public swimming pools [7].

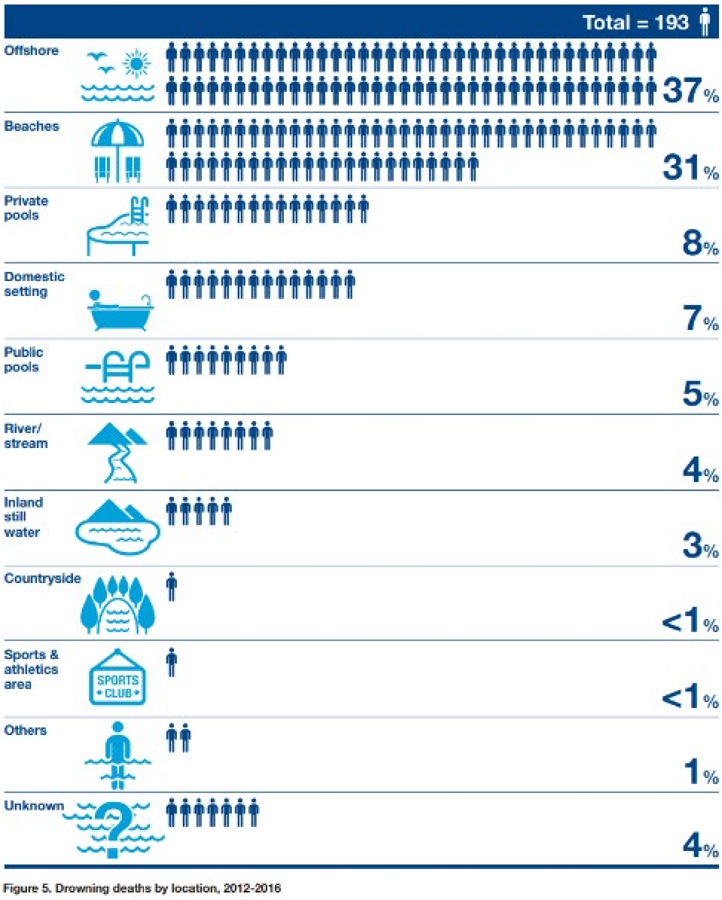


Figure 1 Drowning Death by location in 2012-2016 [7]

Furthermore, even for swimming classes with a coach, there are occasionally drowning cases for young swimmers. This increases the need for a better system. With the knowledge and concern mentioned earlier, improvement in drowning detection and rescue is imperative.

With the above considerations and case selection, to ease the lifeguard shortage pressure, a Swimming Pool Drown Detection and Rescue System is introduced to assist the lifeguard in efficiently identifying and saving the drowning victim.

Despite that there have been some developing drowning detecting systems, the current designs are mostly focused on drowning detection and warning. In our system, we would also include a follow-up action to shoot the kickboard to rescue the victim. With the drowning detection system, warning system, and kickboard throwing in one system, the signal transmission and reaction time can be further minimized. The rescue time can be shortened and become more automated. This aims to achieve a higher rescue rate since faster rescue means a higher chance of survival and fewer injuries experienced by the victim.

The drowning detection and warning sub-system is a combination of an electronic wearable device, a computer system for drowning alert trigger, a victim localization and a web application for the lifeguard to decide what to do when drowning is detected. Electronic components and computer programming will be utilized here.

The kickboard throwing sub-system consists of a robot that could move the mechanical parts such that it aims at the location of the drowning person and throws a kickboard for rescue. The throwing mechanism will be implemented by our Mechanical Engineering team [1] for the mechanical design and modification.

In the current development of our Final Year Project, we would focus our system on indoor swimming pool rescue for Olympic Pool Sized pools. In the future, the system may not be limited to indoor or outdoor swimming pools but also be used at beaches or at the seaside.

[1] MECH FYDP team consists of Lok Yan Loreen LEUNG, Tsz Ho CHAN, Ka Hei LEE and Prof. Lok Wang Robin MA (Supervisor).

1.2 Objective of the Project

The aim of our project is to detect and rescue drowning victims in the pre-drown stage in a swimming pool. Although an IoT wearable device detection system will be developed to identify the victim, the lifeguard will also then be alerted through the Web application and is given a choice to decide whether to shoot or not. Once the choice to shoot has been made, the kickboard will be shot to the drowning victim's location to aid the victim's self-rescue capability and shorten the time needed for lifeguard rescue.

Our system consists of three parts, namely (1) Wearable sub-system, (2) image processing sub-system, (3) server sub-system, and (4) shooting sub-system. A wearable device and image processing are done in the "drown detection sub-system". Through a combination of the three systems, the rescue speed might be able to be further minimized.

With this aim in mind, we define the following project objectives:

Objective 1: To build a wearable device that identifies potential drowning victims

The wearable device will be responsible for detecting the blood oxygen level and the heartbeat of the swimmer during swimming every 30 seconds. The device will then send the signal to the web application control. Within the wearable device, some key abilities include:

1. Enhanced water-resistant to 5 BARS [2] [8] of the wearable device. This is used to support the sensor to function well for swimming in a swimming pool.
2. Accurate collection of sensor's data underwater and during exercises.
3. Analysis of the data collected by the sensor to determine whether the situation is abnormal or has fallen into the drowning category
4. Transmitting the drowning message information from the swimming pool to the web application control system beside the swimming guard. The distance would be approximately 50m above water and 1m depth in water.

We will also explore the possibility of adopting a camera-based approach to locate the drowning swimmers who sunk with their heads in the water for 15 seconds as a hybrid approach to identify the stage 2 [3] drowning victim. In this approach, the camera would be mounted underwater inside the swimming pool to capture the swimmers' actions and moments.

[2] 5 BAR water resistance means the device is capable of withstanding splashes, showering and swimming. It is not suitable for pool-side diving, sea diving or water sports like snorkelling and water-skiing.

[3] Stage 2 means the victim chokes in water, and his/her vocal cord involuntary spasm to protect one's lung. More details are given in Section 1.3.2 Related Research..

Objective 2: To locate the drowning victim accurately with a sub-system

The location detection sub-system will be developed in a camera-based approach, where visible light detection by a camera system is used to identify the location of the possible drowning victim.

When the victim's wearable device sensor detects abnormal readings or has reported drowning, the Arduino LED device on the wearable device will be lit up for the location system to detect the light. In addition, a camera mounted on the top of the swimming pool will be used for checking the location of the emitted light from the wearable device. This camera selection and location further enhance the identification of the location. The computation of the camera-light location system would be done by the external computer system due to the large processing power.

Objective 3: To build a user-friendly application to alert and assist the lifeguard in making drown and rescue decision

The aim of building an application is to allow the lifeguard to make the final decision for the kickboard throwing mechanism operation. Since the lifeguard should be a professional judgment in rescue, the final decision would be to fall back on the lifeguard.

In order to allow the lifeguard to continue the regular routine checking while being aided by the system, there are three requirements. These requirements make the application easy and require minimal attention for the lifeguard to use.

The criteria include:

1) User-friendliness for first-time users. Since the lifeguard might not be professional in technology, it is suggested to have a simple and straightforward interface.
2) Responsiveness of the application. Since rescue is in a timely manner, the system should have a minimal delay, so the kickboard throwing system can be initiated once the confirmed signal is sent.
3) Clarity. The button and interface should be clear and easy to operate. Since rescue is a timely issue, it is not preferred to have the situation of wrong button pressing occur.

Node.js will be used to develop the web application to enhance compatibility. Different tablet computers and mobile devices may also use the application. This allows the flexibility of the management of the swimming pool. We will also simulate the swimming set in the application to make the situation more manageable for the lifeguard to use.

Objective 4: To Integrate the Detection and Warning System with the Mechanical Kickboard Throwing System

For this objective, due to the immediate rescue aim, the signal transmission between the Detection and Warning System and the Mechanical Kickboard system should be fast. Thus, various signal transmission approaches will be explored, including but not limited to Wi-Fi transmission and cable transmission.

Moreover, the data transmitted from the detection and warning system, particularly the location and the basic sensor information, should be merely essential and sufficient data. This enables the focused resources on the operation of the Mechanical Kickboard system.

There may also be some mechanical kickboard throwing system data, including the current shooting stage, possible error message, and power status, transmitted back to the web application for error reporting and advanced monitoring of the Mechanical System for the lifeguard.

Objective 5: (MECH) To calculate and test the shooting distance and shooting angle

This part is responsible by the students from the MAE department.

A three-dimension shooting machine is used. A spherical polar coordinate system, as illustrated in Figure 2, is used to control the shooting distance. The magnitudes of two angles and the shooting velocity are calculated by using the data collected by the sub-system. By controlling these parameters, the life ring can be shot in different positions.

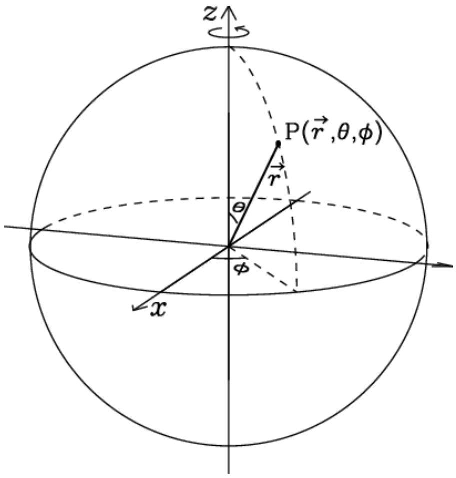


Figure 2 Figure illustrating the spherical polar coordinate system

Objective 6: (MECH) To shoot the kickboard to the potential drowning victim's location

This part is responsible by the students from the MAE department.

Air cannon and a wheel shooter are considered. For air cannon, air pressure is used to ignite the launching. For the wheel shooter, two wheels, which are driven by two motors, are used to accelerate the kickboard to create a projectile motion. The shooting time is targeted to be smaller than 20 seconds.

1.3 Literature Review

There are serval drowning alert systems and drowning rescue robots in commercial use or research. We will investigate their approaches and examine their advantages and disadvantages with different criteria.

1.3.1 Existing System

We have studied the existing system: EDDS [9], SwimTrack [10], Automated Underwater Lifeguard [11], Artificial Intelligence for Drowning Detection and Swimmer Performance Analysis System [12] [13], SwimEye [14], which uses different approaches and has different abilities. A comparison table is after the detailed description.

EDDS (Early Drowning Detection System) [9]

EDDS uses PPG (photoplethysmogram) sensor to monitor the user's abnormal change in heart rate. Before the victim drowns, he will experience a panic situation, where the heartbeat may increase drastically due to hyperventilating. EDDS monitor that drastically raises to detect victims before they drown. To continuously monitor the user's heartbeat, the sliding window technique is used. It counts the heartbeat samples for the first 10 seconds and multiplies by 6 to provide a heartbeat reading per minute (BPM) and continuously shifts for one second to give a new heartbeat reading. The accuracy of EDDS is good. In Figure 4, the area under the ROC curve is 0.94, where 1 is a perfect test and 0.5 is a useless test. However, the position of the victim is not detected in EDDS, which makes rescue less efficient.

Regarding communication method, 433MHz Universal Asynchronous Receiver/Transmitter (UART) is used in EDDS as the wireless communication protocol due to the ability to establish long-range wireless transmission. Even though 433MHz has the second maximum ability to penetrate water, loss of radiofrequency signal still occurs.

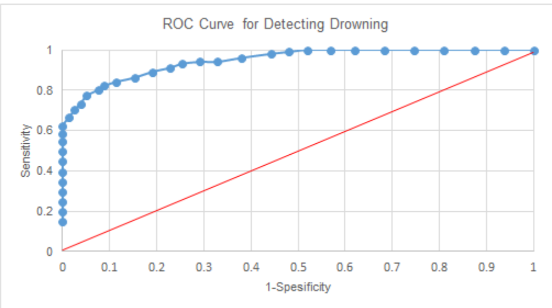


Figure 3 ROC Curve for Detecting Drowning [9]

Figure 4 Drowning Detection Device wear as a headband [9]

SwimTrack [10]

SwimTrack is drowning detection research utilizing the fact "water will block RFID signals.". In a swimming motion, the wearable RFID is detectable while the wrist is above water and indetectable while the wrist is underwater. By recording the "detectable/indetectable" frequency of swimming and drowning motions, they found that natural swimming strokes have a regular pattern, but drowning strokes has a random frequency. Therefore, drowning can differentiate from swimming motion by the frequency of the RFID tag detected.

Although the RFID approach is innovative, it fails to meet the need for real-world application. After nearly 2 meters, the RFID tag disappeared from the detection range of the antenna, which is unpractical to deploy in a 50m*20m swimming pool. Moreover, this research still incorrectly classified the drowning samples as it requires more empirical data to produce a decent error rate.

Automated Underwater Lifeguard [11]

Automated Underwater Lifeguard is an autonomous robot that uses computer vision with a haar cascade and convolution neural network to detect drowning victims in a swimming pool. The drown location is obtained by an affine transforming the location of the victim into a map grid, and then the robot travels toward the XY coordinate and deploys a float to keep the victim afloat. This robot is one of few examples that has an active rescue procedure.

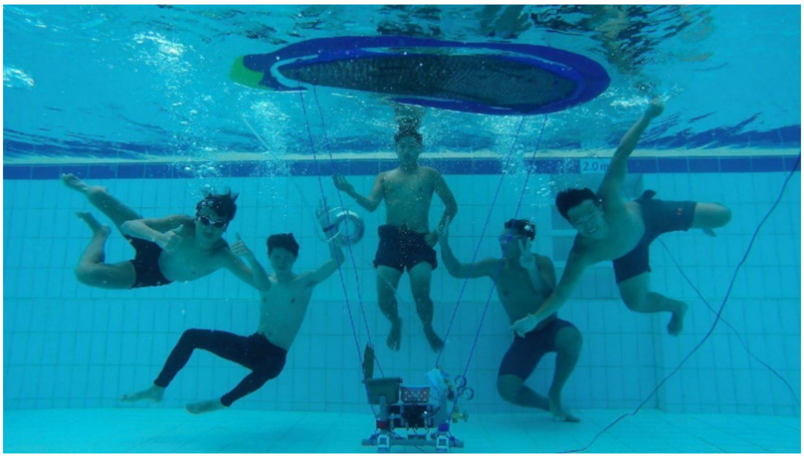


Figure 5 Photo of the Automated Underwater Lifeguard with the development team taken underwater [11]

Artificial Intelligence for Drowning Detection and Swimmer Performance Analysis System [12]

This project uses a computer vision approach to detect drowning. Sixteen cameras are placed around the pool, 12 at different points underwater and 4 on the ceiling. The footage captured is then labelled and fed into the computer to teach the AI about body parts and movements. Once the computer has accumulated enough data, it can monitor the movement in the pool.

When the threshold of drowning probability is reached, for instance, in Figure 7, an audio alarm is triggered, and the location of the victim will display on a computer for the lifeguard. This system can process the image into body skeleton movement to analyse drowning without recording the swimmers.

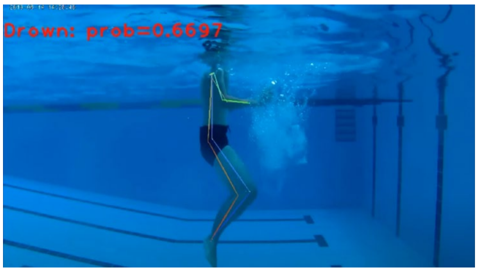


Figure 6 Drowning Detecting in Action [13]

SwimEye [14]

SwimEye has more than ten years of experience in the pool security system, currently used in 120+ swimming pools. This system places their underwater double camera houses within 9 meters to create monitoring overlap and additional cameras to cover the dead zone. This camera configuration allows this system to monitor 100% of the pool basin area in most pools.

Footage from those cameras is automatically monitored by an object recognition software called state-of-the-art. When it detects a swimmer in distress at the bottom of the pool, it will raise an alert. At first, an early warning yellow alert will rise, which implies that a swimmer might be in the first stage of drowning. A yellow box will catch the attention of lifeguards to that camera, and the lifeguard can click “reset” to stop the yellow alarm. If the alert lasts for more than 15 seconds, a red alert will raise with visuals and audio to alarm the lifeguard.

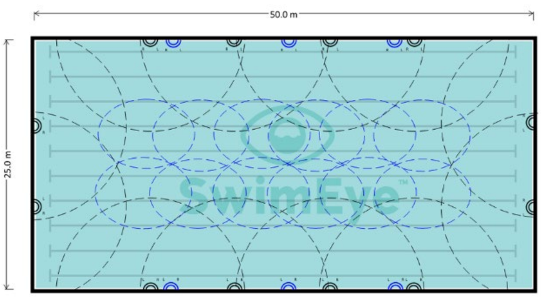


Figure 7 SwimEye’s camera placement [14]

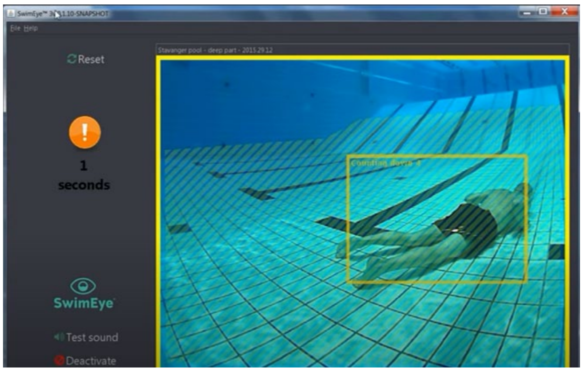


Figure 8 SwimEye detected a yellow alert situation [14]

Approaches	EDDS	SwimTrack	Automated Underwater Lifeguard	AI for Drowning Detection …	SwimEye	Our Design
Active Rescue Method	N/A	N/A	Deploy a float by underwater robot	N/A	N/A	Shooting a kickboard
Detect Method	Heart rate	RFID	Computer Vision	Computer Vision	Computer Vision	Heart rate
Positioning Method	No	No	Computer Vision	Computer Vision	Computer Vision	Computer Vision

Table 1 Comparison Table of approaches of existing system

Criteria	EDDS	SwimTrack	Automated Underwater Lifeguard	Drowning Detection with AI	SwimEye	Our Design
Cover Area	Within signal	Very small	Robot's sight	Within camera	Within camera	Within camera + anywhere (Blink function only)
Detect time	Pre-drown	Drown	Drown	Drown	Sunk	Pre-drown
Accuracy	Good	inaccurate	Depends on the amount of data collected	Depends on the amount of data collected	Commercial grade	Similar to EDDS
Cost	$$ Wearable +Server	$ RFID+ Antenna	$$$ Underwater Robot	$$$ Cameras+ Powerful computer	$$$$ Commercial	$$ Wearable +Server
Active Rescue	No	No	Yes	No	No	Yes
Active Rescue Time	N/A	N/A	Take time to travel to under the victim.	N/A	N/A	Fast, within seconds

Table 2 Comparison Table of pros and cons of existing system

From Table 2, those detecting systems can divide into the wearable approach and computer vision approach. While computer vision approaches can detecting drown victims, it is worth noting that their solutions rely on sufficient empirical drowning data. Moreover, computer vision approaches can only detect victims that are drowning or even sunk, while the wearable approach can detect the pre-drown period, which is important in the time-critical life-saving task. As the wearable approach does not rely on a large amount of empirical drowning data and has the advantage of detecting speed, EDDS is the model that our design is based on.

Our design inherits most of the features from EDDS, on top of which provides positioning function, resolves the problem of signal loss, and an active rescuing system faster than Automated Underwater Lifeguard.

Compared to the computer vision approach, position function and signal attenuation underwater are the weaknesses of the existing wearable approach. Our design can fill in those niches. As visible light has the highest penetrate-depth underwater, by making use of LED integrated into the wearable device, we can position the user by the light emitted from the wearable devicec and avoid electromagnetic wave attenuation. As an additional function, it can alert swimmers nearby.

Furthermore, an active rescuing system is rarely implemented in existing works. The Automated Underwater Lifeguard is the only indoor example we found. The underwater robot travelling in a 25m by 50m standard swimming pool takes a significant amount of time. It will not be very meaningful if it is slower than a lifeguard. Therefore, we will try a new shooting approach that takes less than a few seconds to reach the victim.

1.3.2 Related Research

Physiological Effect of Drowning

Many may think drowning victims will scream and have rapid movement when drowning, but that is incorrect. According to the World Health Organization (WHO), the definition of drowning is “Drowning is the process of experiencing respiratory impairment from submersion/immersion in liquid.”, “Any submersion or immersion incident without evidence of respiratory impairment should be considered a water rescue and not a drowning.” [4]

In stage 1, pre-drowning, the victim will panic and react with violent struggling, and the victim is still able to scream for help. Also, hyperventilating will occur, leading to a sudden rise in a heartbeat [9]. This stage will last for 20 to 60 seconds for adults and 10 to 60 seconds for kids. [15]

In stage 2, when the victim aspirates water, it will trigger laryngospasm, which means the vocal cord will spasm to block the airway and stop water from coming in to protect their lung. However, it also blocks the airway for breathing, which leads to brain hypoxia. In this stage, the victim can hardly differentiate from a beginner swimmer as the victim is fatigued from the violent struggle and cannot yell for help due to laryngospasm. [15] [16]

In stage 3, laryngospasm stops due to brain hypoxia, and aspiration of water continues. Hypoxemia quickly leads to loss of consciousness and apnea. The sequence of heart rhythm deterioration is usually tachycardia, then bradycardia, pulseless electrical activity, and finally cardiac arrest [4].

PPG (Photoplethysmogram)

PPG is an optical technique used to detect volume change and oxygen change of blood in the peripheral circulation. When infrared (IR) travel through human tissue, it gets absorbed by bones, skin pigments, and blood. To measure heart rate, PPG detects the reflected IR intensity change. Since blood absorbs more IR light, when blood volume gains, the IR intensity reflected will decrease, vices versa [17]. For blood oxygen, we detect the colour of light reflected. Blood with sufficient oxygen will absorb more infrared and reflect more red light. Less oxygen will absorb more red light, making reflected light appear bluer [18].

A field that makes use of PPG is air safety. According to Air Safety Institute Safety Alert, when a pilot measured blood oxygen saturation is below 90%, his mental function begins to deteriorate, and he should receive enough oxygen to function normally. Therefore, we may consider that 90% of blood oxygen saturation is in a danger zone [19].

Having 90% as the threshold is further supported by the medical study conducted by University and Medical Centre in the Netherlands [20]. Figure 13 shows the measured value of blood oxygen after swimming is still above 90% in a real-life scenario. Only a few cases lie between 80 and 90%, or even lower. The fact that there was no drowning case reported throughout the study, verified the validity of using 90% as our threshold value.

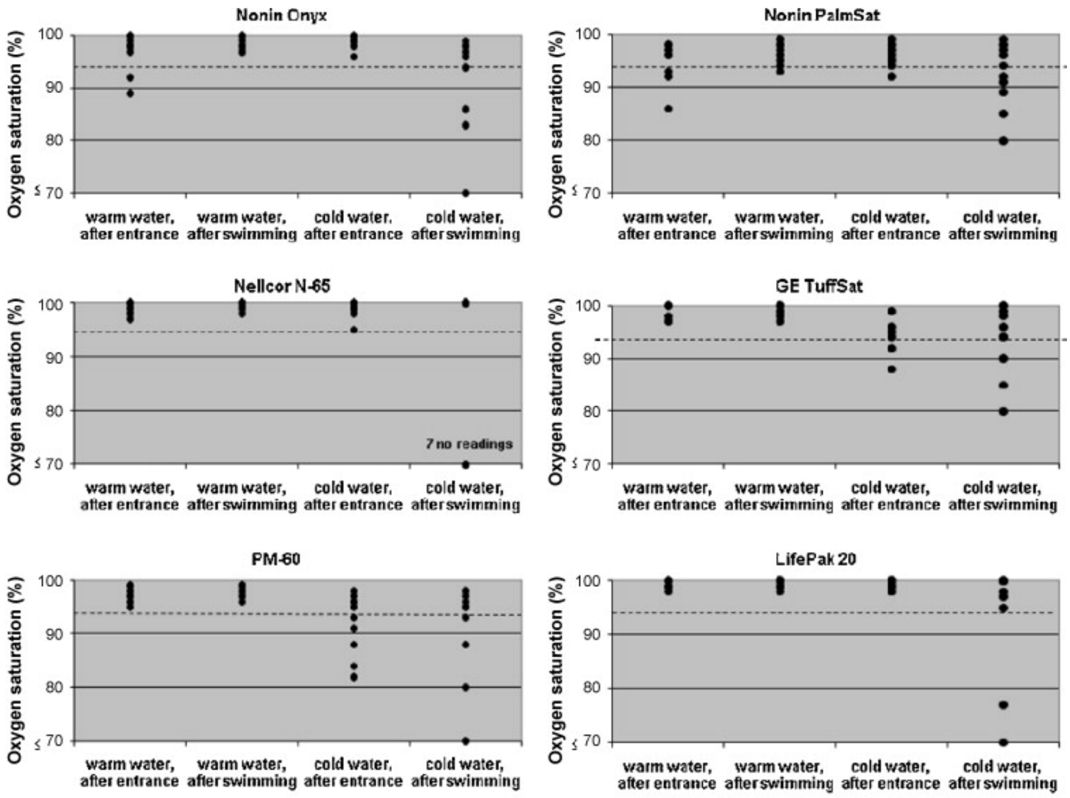


Figure 9 Line plots for oxygen saturation values (y-axis) per test situation (x-axis) from different pulse oximeters

Although PPG is a cheap and easy way to measure heart rate and oxygen levels, it has its limitation. PPG is inaccurate in tracking the PPG signals during daily routine activities and light physical exercises because of Motion Artifact (MA) caused by hand movement [21].

As a counterargument, commercial wearable health devices such as Garmin Venu and Huawei Band 6 do the all-day monitor. Take ZOLL's R Series Pulse Oximeter (SpO-2) as an example. During the no-motion condition, its accuracy is within ±2%, and during motion conditions, it is within ±3%. The error range is acceptable, especially in extreme cases like hypoxia [22].

As suggested in the "Feasibility of pulse oximetry after water immersion", the value taken from the finger is appreciated and more accurate [23]. This meant the possibility of using it on the wrist with further support from the wide usage of healthcare watches.

2. Methodology

2.1 Design Philosophy

To maximise the ability of the system to rescue, there are a few design principles to be taken into account.

- Responsiveness
  - As an active rescue system, the response time has to be short to save time to rescue. Thus, the response and signal transmission between different systems have to be minimal.
- Modularity
  - Having the system divided into sub-systems, they can contribute individually to alert and thus support the drowned rescue decision making of the lifeguard. This also facilitates the debugging and easy maintenance of the system in the long run.
- Optimised for the Swimming Pool Environment
  - Being a computer system operated in a swimming pool, durability is very important. Since electronic components will be easily damaged upon water contact, the material selection should be carefully made.
- Minimal disturbance to the routine
  - Being a rescue system supplementary to the lifeguard observation, the system is not designed to replace lifeguard professionals. Thus, lifeguards should be not disturbed or are required to spend additional attention observing the report from the system. The system should be mainly automated and have its own ability to alert the lifeguard instead. Additionally, the location and the installation of the system should not affect the regular operation of the swimming pool.
- Safety
  - Being used by the poolside and as a rescue system, the system aims to rescue instead of causing casualty. Thus, the insulation of electricity becomes crucial. Moreover, the system setup should not induce possible injury.

2.2 System Overview

The system consists of 4 sub-systems, the wearable device, the server, the image processing algorithm, and the launching robot. Figure 9 shows how the sub-system is integrated and how the data transfers in between. First, the wearable device will detect if the user is drowning. If so, it will emit an LED signal, and the server's camera will capture that signal. Then the server will send the webcam input to the front end and the image processing algorithm to determine the position of the LED. Once the location is known, it will send to the front end to alert the lifeguard. The lifeguard will give a shooting command or trigger automatically after the timer runs out. The command is then sent to the kickboard launch robot to shoot the kickboard to the destination. In the following, we describe the design and implementation of each sub-system in detail.

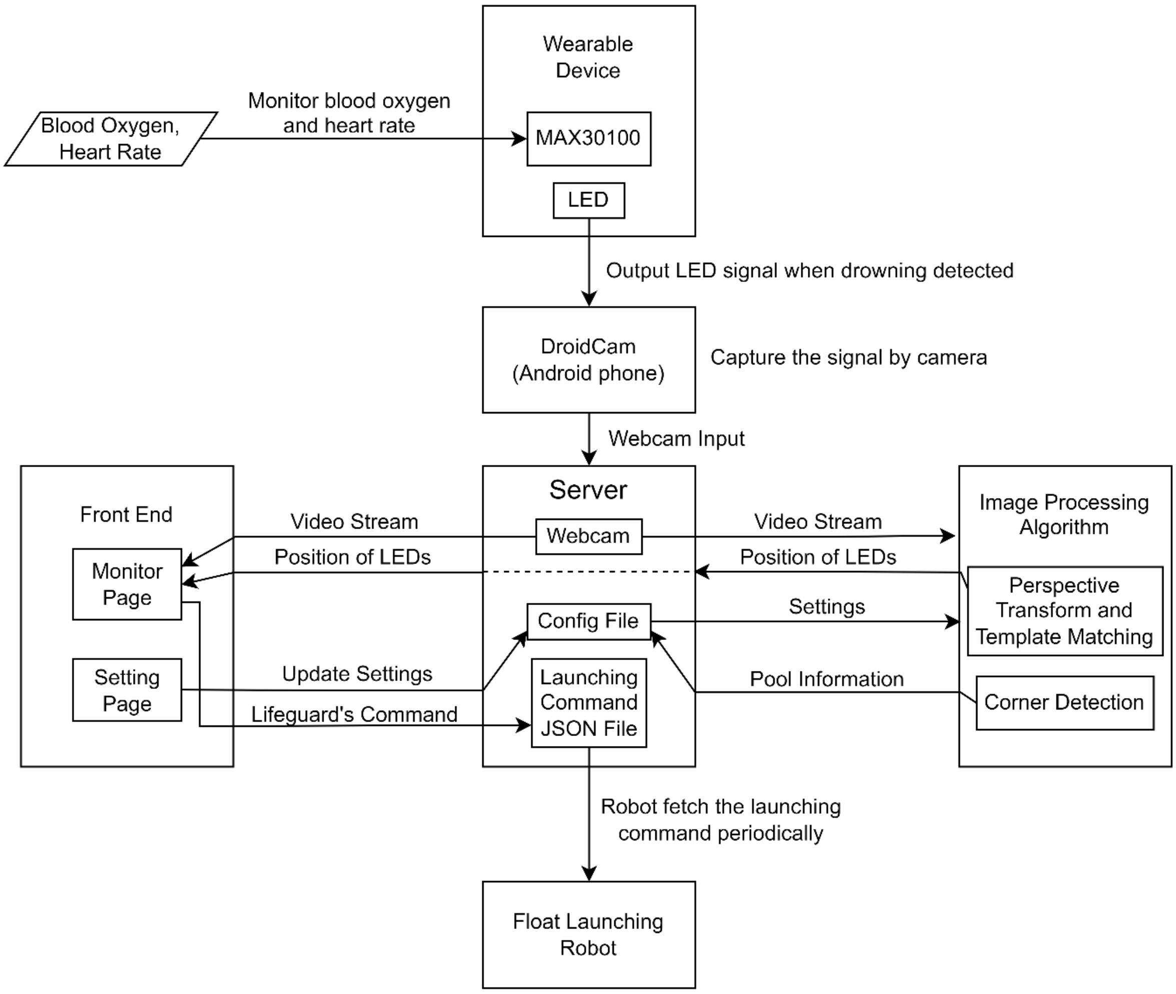


Figure 10 System architecture

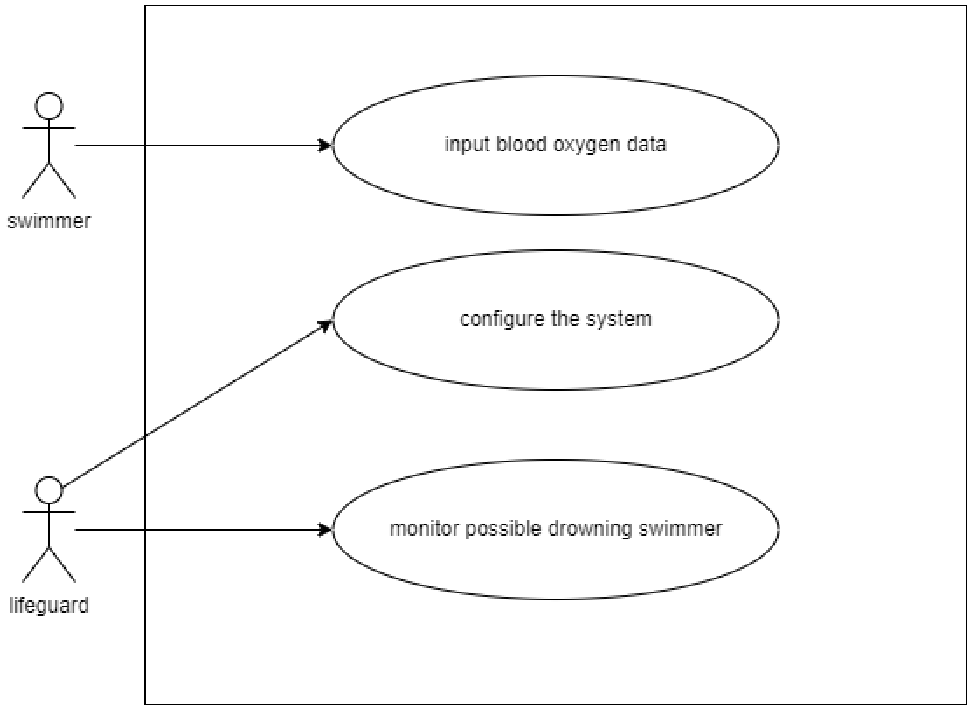


Figure 11 Use Case Diagram

2.2.1 Use Case: Input Blood Oxygen Data

This use case describes how a swimmer inputs blood oxygen data into the wearable device when he/she is in the swimming pool.

Basic Flow

1. The use case begins when the swimmer wears the device before going to the pool. To wear the device, the swimmer should fasten the strap and fit it onto the wrist.
2. The swimmer turns on the device, and data will be collected automatically.
3. The swimmer turns off the device and takes it off after he/she leaves the swimming pool.
4. The use case ends.

2.2.2 Use Case: Configure the System

This use case describes how a lifeguard configures the pool size, pool corners and bot position.

Basic Flow

1. The use case begins when the lifeguard chooses to configure the system.
2. The system displays the configuration page.

   {Input pool size, bot position, and count down time}

3. The lifeguard inputs the pool size, bot position and countdown time and clicks the confirm button.

4. The system displays a message to notify the lifeguard that the inputs are received.
5. If the lifeguard chooses to perform automatic corner detection
    5.1. The lifeguard clicks the detect button, and the four corners are detected automatically.
6. The system displays a screen for manual adjustment of the corners.
7. The lifeguard adjusts the corners manually.
8. The lifeguard confirms the configuration.
9. The system displays a message to notify the lifeguard that the inputs are received.
10. The use case ends.

Alternative Flow

A1: Invalid pool size/bot position/countdown time

At {Input pool size, bot position, and countdown time} if any of the inputs is not a positive integer,

1. The system displays a message to notify the lifeguard of invalid input.
2. The flow of events is resumed at {Input pool size, bot position and countdown time}

2.2.3 Use Case: Monitor Possible Drowning Swimmer

This use case describes how a lifeguard monitors the swimming pool and shoots a kickboard at the drowning swimmer.

Basic Flow

1. The use case begins when the lifeguard chooses to monitor the pool.
2. The system displays the swimming pool.

    {Monitor the swimming pool}

3. While the lifeguard is monitoring the pool,

    {Detect drowning swimmer}

    3.1. If the system detects a drowning swimmer,
        3.1.1. The system marks the drowning swimmer with a yellow box.
    3.2. The lifeguard clicks on a position of the pool.
    3.3. The system displays a prompt window with “Confirm rescue? Yes/No”.
    3.4. If the lifeguard clicks “Yes”,
        3.4.1. The system shoots a kickboard to the target

3.4.2. The prompt window closes.

3.5. If the lifeguard clicks “No”,

3.5.1. The prompt window closes without any action taken.

Alternative Flow

A1: No response from the lifeguard

At {Detect drowning swimmer} if no response is received from the lifeguard after some time,

1. If the prompt window has opened,
    1.1. The prompt window turns red.
2. Else,
    2.1. The prompt window pops out in red.
3. The system displays the countdown timer in the prompt window
4. If still no response is received from the lifeguard after the countdown,
    4.1. The system shoots a kickboard to the target automatically.
5. The prompt window closes.
6. The flow of events is resumed at {Monitor the swimming pool}.

2.3 Wearable Sub-system

2.3.1 Design

The wearable system is based on the concept of the internet of things (IoT). This system consists of data analysis, wearable hardware design and waterproof design. In detail, the software required is C language for Arduino; the hardware counterparts include Arduino Nano, GY-MAX30100, WS2812 5050 RGB Round Panel, and a 7.4V power supply.

As shown in Figure 11, Figure 12, this sub-system targets to alert the imaging processing sub-system with a victim detected with a high chance of being drowned. With the user's blood oxygen level measured, the system would decide if the swimmer was drowning and then respond with a warning light signal.

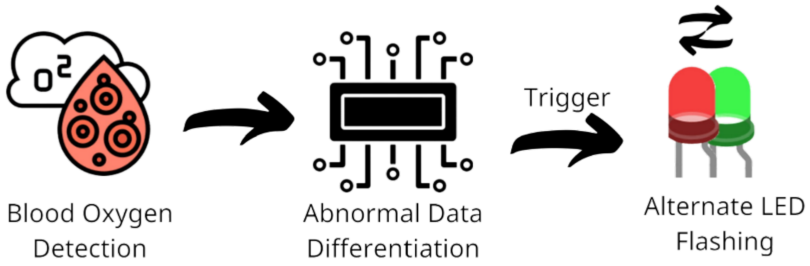


Figure 12 Image diagram illustrating the flow of the wearable sub-system

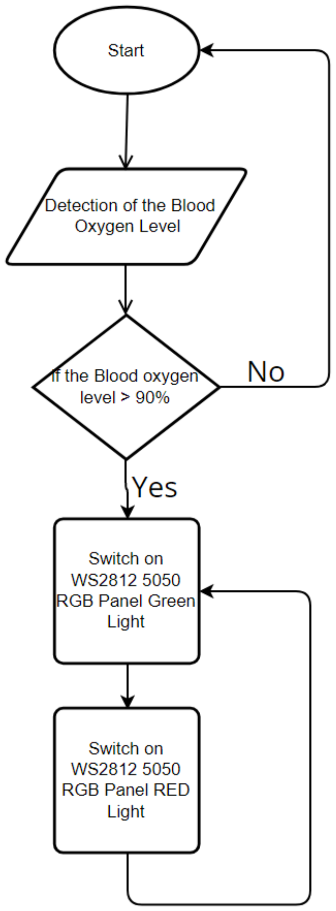


Figure 13 Flowchart representation of the drown-alarming sub-system

2.3.1.1 Software Design

In order to measure the blood oxygen level and give out a warning as soon as possible, once the lower than threshold blood oxygen level is detected consecutively, the blinking mechanism is triggered to start the whole active rescue system.

By the working principle of the MAX30100, as illustrated in Figure 14. The infrared is emitted from the diode and reflected from the fingertip back to the photodetectors. With thin skin between the blood vessel at the wrist, setting the detection location at the wrist can reduce signal noise and infrared loss by travelling through to other body tissue.

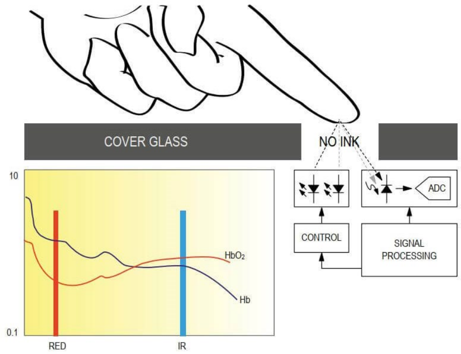


Figure 14 MAX30100 sensor system [24]

2.3.1.2 Hardware Component

Mainly five components are used. To minimize the dimension for ease of wearing, smaller devices are used. These include one Arduino Nano, one GY-MAX30100, four pieces of 7 Bit WS2812 5050 RGB Round Panel, a 7.4 V power supply by two 18650 Lithium-ion Battery and wrist band.

Before selecting the hardware, two factors were considered: small dimension and ease of usage. Therefore, Arduino Nano is chosen due to the minimal dimension and sufficient connectivity for the input and output devices. This includes both the MAX30100 sensor and the WS2812 5050 RGE LED pieces. In our original design, we have also considered adding a waterproof speaker. However, due to the user experience and vulnerability added to the design, the design was later discarded.

Arduino Nano

Arduino Nano is chosen with the consideration of minimal disturbance to the routine for the swimmer, compatibility, and support to other electronic components. Comparing STM32 and Arduino Nano, Arduino Nano is smaller with fewer but sufficient functions. The smaller size helps build the waterproof layer since this size is easier to control. Although both are written in C language, Arduino's platform consists of more libraries and support. The support further facilitates us to collect data from the sensor ahead for more accurate calibration.

Comparing Micro-bit, a smaller microcontroller, with Arduino Nano, Micro-bit is a more basic micro-controller. The customization and programming are comparatively limited due to the fact it was designed for academic purposes for the general public.

GY-MAX30100

We have made some comparisons with another model of MAX30100. As shown in Figure 15, the green board is with resistors building deflect. The cheap cost suggested the testing application purpose for the water test, for the destructive testing.

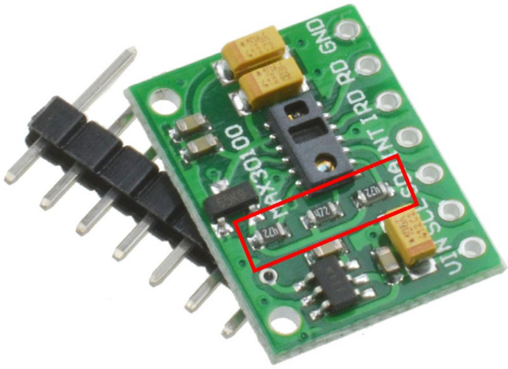


Figure 15 Green MAX30100 Circuit with design Deflect

WS2812 5050 RGB Round Panel

The RGB Round Panels (known as LED) are used as the output, signal emission medium for the camera to capture. However, considering the warning LED being blocked by the waving of the drowning swimmer's hands, four LEDs are installed in different directions, namely, top, bottom, left and right, placed around the wrist.

Also, we have tested with different colours, including but not limited to red, green and blue. There are some specific concerns about the colour. With the colour filter used by the image processing, various type of blue colour or colours that appears to be similar to the pool has to be excluded. With further consideration of the possibility of the image processing unit mistaking the swimmer's suit as an LED, dual colours are introduced. The LED will be blinking between red to green to enhance accuracy.

Waterproof Speaker with 4 Ohm and 3 W

Specifically, the user experience is limited by the dimension of the hardware diameter of 50mm. Compared with the normally worn locker key bracelet straps suggested in Figure 16, the longest key accepted would be 50mm, with a pointed design [25]. However, the waterproof speaker we would suggest is 57mm in diameter, with the additional circuit board, PAM8403, needing to adapt to the Arduino. It will increase the wearable device to be a larger size and increase in weight. Thus, the idea of including such is later removed.

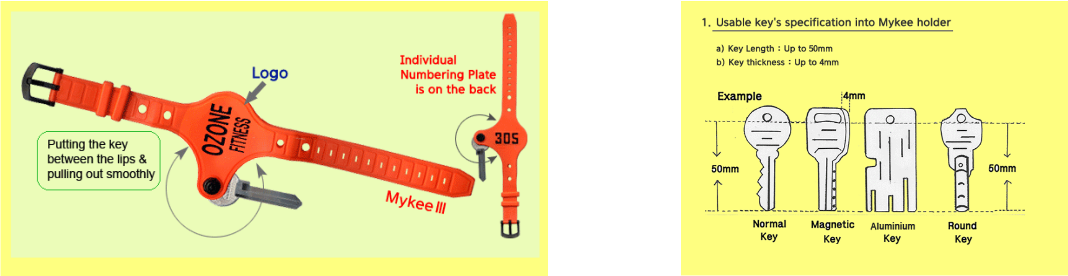


Figure 16 Advertisement figure shows the swimming wrist band which holds the locker key

18650 Lithium-ion battery

To power up the pulse oximeter, the 4 LEDs and the Arduino Nano, 240mA and 5V are required for a functioning prototype. With this consideration, 2 3.7V 18650 batteries are used in series to provide a sufficient current of 3.8A and voltage of above 6V for the power supply.

The battery is swapped from 3V coin batteries (CR1220 / CR2032) to 2 18650 batteries due to rechargeable ability and battery capacity. Coin batteries are not sufficient to provide the required current to the system. It is also worth noticing that two batteries are needed to work in series in order to provide sufficient current for the LED to lit up (240mA) while powering the Arduino (6-9V).

Component	Arduino Nano	GY-MAX30100	WS2812 5050 RGB Round Panel	Waterproof Speaker with 4 Ohm, 3W	3.7V 18650	Wrist band
Function	Processor	Blood Oxygen Sensor (also known as Pulse Oximeter)	Warning & Signal for Location detection	Warning	Power supply	Link up all the component
Amount (pieces)	1	1	4	1	2	1
Dimension						
Width(mm)	18	14	17	57	36	15
Length(mm)	45	14	17	57	65	230
Height (mm)	5	3	3	13+2	18	1
Pin Number	32	5	3	2	2	-
Input Voltage						
Max. Voltage	12	5	7.5	4.8989	7.4	-
Min. Voltage	7	3.3	-0.5	3.4641	4	-

Table 3 Summary of the hardware for the wearable device on functionality and specifications

Table 3 concludes the specification for all the used hardware. This specifies the exact version and functionality of the hardware on the wearable device.

2.3.1.3 Waterproof Design

A 3D model will be designed to contain the electronic elements with potting by silicone rubber, as shown in Figure 17. We will design two versions of the drawing and model to achieve an optimal waterproof of 5 bar. The first version aims to provide fundamental skills and experience for potting with silicone rubber. The second design would focus more on the user experience and accuracy.

Figure 17 Electronic component potting with silicone rubber

2.3.1.4 Wearable Design

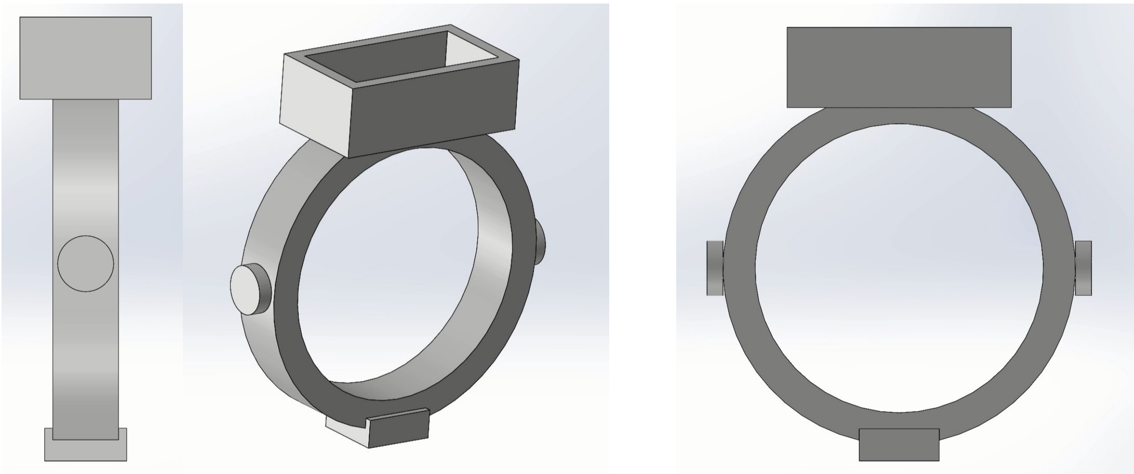


Figure 18 SolidWorks drawing for the mould of the potting container

Figure 19 Waterproof bracelet for attaching the electronic parts [26]

As illustrated in Figure 18, the wearable device would be wrapped around the circular ring. The circular ring in the middle would be the bracelet. The detailed appearance of the bracelet can be referred to the Figure 19. Small cylinders are the left and right LEDs that go around the bracelet to wrap around to face a different direction. The top part is the potting area for the Arduino Nano, while the bottom part is mainly potting for the MAX30100.

2.3.2 Implementation

2.3.2.1 Software Implementation

The software mainly includes the input signal from the MAX30100 (pulse oximeter) and the output signal to the LED. Among the two parts, libraries were used for enhancing their functionalities respectively.

For the pulse oximeter, the library named MAX30100_milan is used. This library is produced by Gabriel Gazola Milan, which aims to provide a modular approach to calculating pulse rate and SpO2 for the Maxim-IC MAX30100. The main functions used in our programme are pox.update() of pulse oximeter and pox.getSpO2(). The code segment Figure 20 below shows the update function and the decision-making component.

To determine if the swimmer is really drowned, a few consecutive abnormal data have to be collected to ensure the lower than the threshold value is not just a loss of connection. This is taken to minimise the effect from some extreme value due to detection error between the skin surface and the sensor. If the data are consecutively detected, the state of the Boolean flag “ledtrigger” will be toggled, and the LED will start blinking.

```
PulseOximeter pox;

  // Setting up the pulse oximeter
  if (!pox.begin()) {
      Serial.println("FAILED");
    for (;;);
  } else {
    Serial.println("SUCCESS");
  }
    // Detection started looping
    pox.update();

    int hrVal;
    int spO2Val;

    // Asynchronously dump heart rate and oxidation levels to the serial
    // For both, a value of 0 means "invalid"
    if (millis() - tsLastReport > REPORTING_PERIOD_MS) {
      hrVal = pox.getHeartRate();
      spO2Val = pox.getSpO2();

      Serial.print("Heart rate:");
      Serial.print(hrVal);
      Serial.print("bpm / SpO2:");
      Serial.print(spO2Val);
      Serial.println("%");

      tsLastReport = millis();

      // Determine if the swimmer is drown
      if (spO2Val < 90) {
      Serial.println("No Oxygen!!");
      numOfZero++;

      // ensure the zero value is detected for a while
      if (numOfZero % 4 == 0) {
            ledTrigger = true;
            numOfSafe = 0;
      }
      // if there are indeed data detected, the
      } else if (spO2Val > 90) {
            numOfZero = 0;
      }
```

Figure 20 Code Segment for the MAX30100

For the output signal, the LED programme has used the WS2812FX library (Version 1.4.0). The library is written by Harm Aldick. This library mainly provides some functions for selecting a specific colour for RGB LED while providing the signal control for customised speed. The main function used is listed in Figure 21.

```
// Set up
  ws2812fx.init();
  ws2812fx.setBrightness(10); // to be 255
  ws2812fx.setSpeed(10);
  ws2812fx.setColor(BLACK);
  ws2812fx.setMode(FX_MODE_STATIC);
  ws2812fx.start();

// main compilation
if (ledTrigger) {

    // if the time period is long enough, change the time
    // and swap the LED
    if (now - last_change > TIMER_MS) {

     // swap from red to green
     if (swap % 2) {
        ws2812fx.setColor(RED);
      } else {
        ws2812fx.setColor(GREEN);
      }
      swap++;
      last_change = now;
}
```

Figure 21 Block included the commonly used code for the WS2812 library.

With the use of a timer, the swapping of LED signal performs faster than the function call of delay(). Since delay() causes programme stall and stop everything running, only utilising timer for the signal control ensures the smooth operation for pulse oximetry and LED. This means that a more accurate frequency swapping can be implemented for image detection. Moreover, since the pulse oximeter uses timers, if delay() is used for the LED part, this will disturb the running of the blood oxygen detection. With blood oxygen detection being more important, control of LED follows the approach of the pulse oximeter.

The colours of the LED are also specially designed to avoid the missed up with the pool colour. In our design, we have chosen Red and Green to minimise the usage of BLUE. Also, the change of LED colour and the high frequency of colour change is added for improved recognition of the image processing component. The LED changing colours contrast with the swimmer's suit; even the swimmer might have worn a red or green suit. Since the swimmer's swimming suit colour cannot change, this further reduces the chance of mistaking the suit as one of the possible values.

A customised high frequency of 7.4 blinks per second (135ms) for the LED colour change aids the image processing unit. Since the swimmer would not move as fast as the frame, say 3m per second, having two different colours changing at about the nearby spot within 1 second suggests a higher possibility that the image reading is the output from the wearable device instead of the swimsuit.

2.3.2.2 Hardware Component

Figure 23 and Figure 24 show the combined circuit. As illustrated below, the 4 LEDs are placed around the wristband. This design ensures the LED can be seen from a different angle. For instance, when the swimmer's hand may be waving, the back of the hand (i.e. top of the device) will face toward the water, as shown in Figure 22. As a result, the top part of the device might not face the camera mount on the pool top and thus cannot be detected. With the tilted angle, the side LED will be shown instead. Then the camera can still catch the signal from the side of the wearable device.

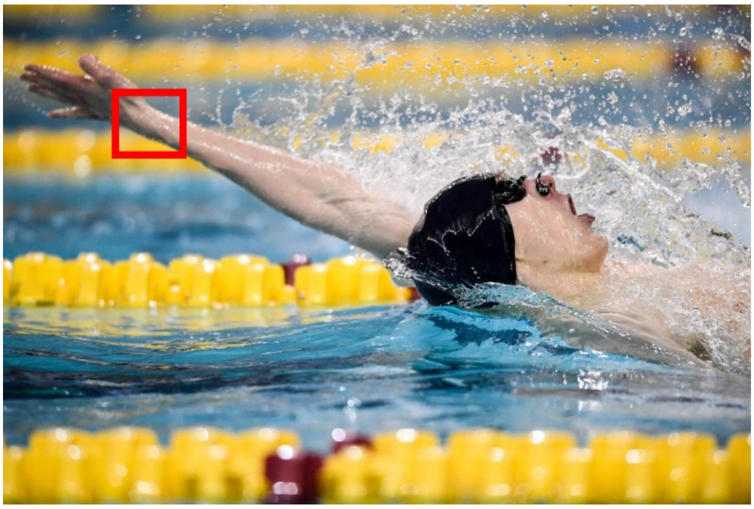


Figure 22 Swimmer having his/her back of the hand facing the pool [27]

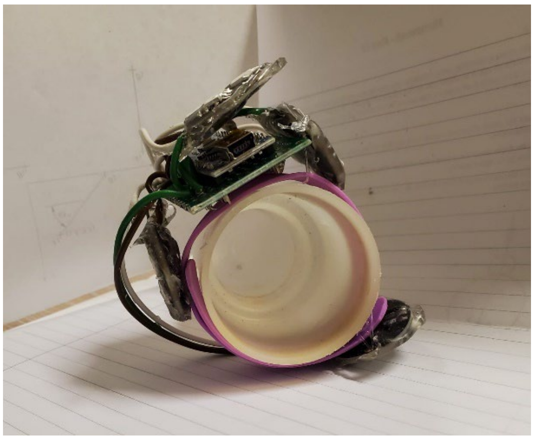


Figure 23 Prototype with the LED and the pulse oximeter

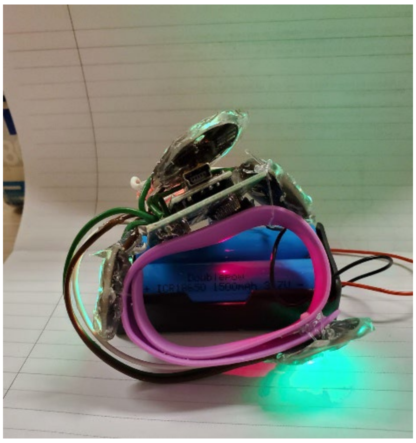


Figure 24 Prototype with the LED and the pulse oximeter with the light lit

According to the testing result from Section 2.3.3.2 MAX30100 Detection Location Accuracy Testing, the blood oxygen level detected directly below the palm gives a more accurate result. Thus, in the last version, the MAX30100 sensor is placed at the bottom of the bracelet. This allows the sensor to have closer contact with the wrist to achieve a more accurate value.

Circuit Diagram

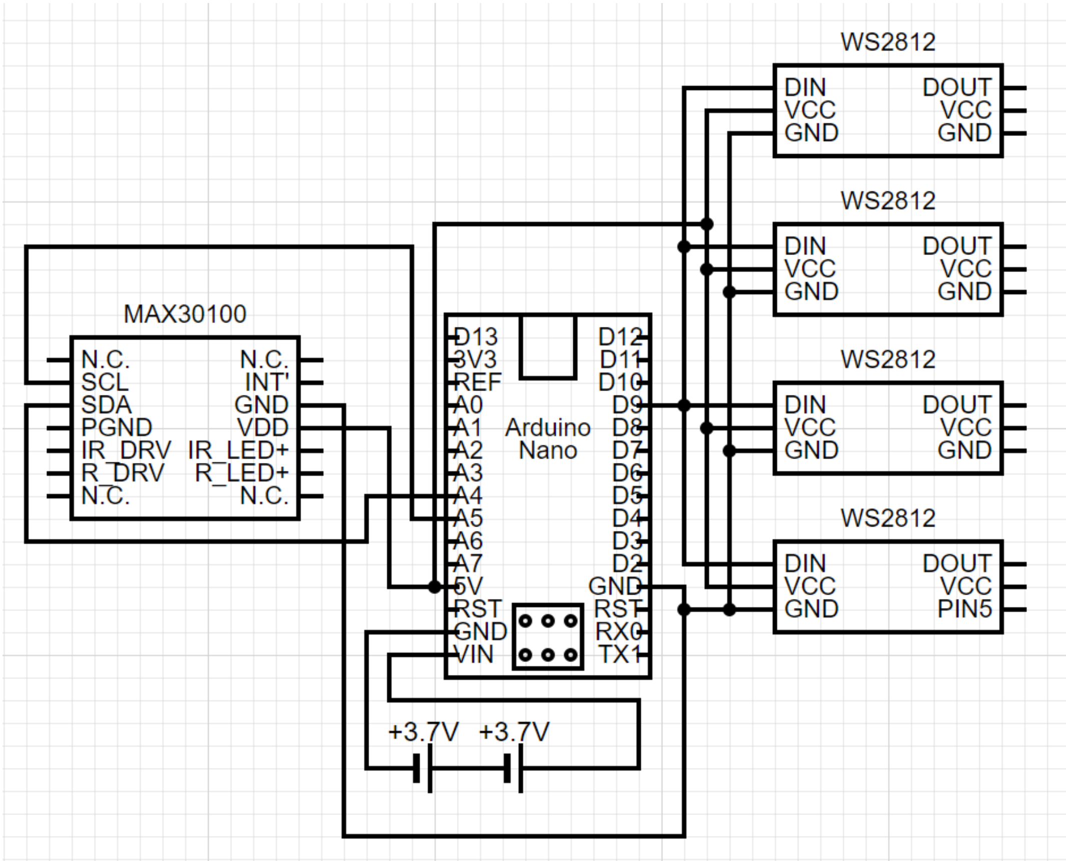


Figure 25 Circuit diagram for the hardware components

Figure 25 shows the detailed circuit diagram with the 4 hardware components. In particular, pin D9 of Arduino Nano is used as signal control for all four WS2812s. Pin A4 of the Arduino Nano is the SDA, and A5 is the SCL for the corresponding pins in MAX30100.

There are also some problems with the programming for the Arduino Nano.

Although Arduino Nano is ideal in size, the debugging progress is commonly known with the communication problem when reloading the programme to the Arduino Nano board. The problem, known as “avrdude”, happens almost every time. Thus, it is highly recommended to test the functionality of the programme on Arduino Uno and the hardware before transferring the circuit design to Arduino Nano. This supports modularity and speeds up the debugging progress.

The main problem we encountered was the avrdude issue. After multiple testing and research, there are some possible solutions for the avrdude: stk500_getsync(). Firstly, the processor should be using “ATmega328P (Old Bootloader)”. Secondly, different programmers have also been tested to bypass such an error. “Arduino ISP” is tested to be functioning for this Arduino Nano Chip.

Furthermore, to avoid any error when loading the programme, it is essential to remove all wires connected to the Arduino Nano Board and then load the programme to the board.

Considering the current requirement, to support the WS2812, more current is needed. Thus, the power supply is updated to the use of a 18650 Li-ion battery to supply 3800mAh. For future development, a battery with a smaller size but similar current and 6-12V will be selected.

2.3.2.3 Waterproof Implementation

To achieve a waterproof of 5 bar, different approaches were tested. The first approach was applying a dual-layer of protection for the electronic circuit to ensure its' functionality even underwater. The second approach optimises the single plastic layer to achieve mechanical waterproofness by potting with silicone rubber.

The first approach was placing the model inside a small lock-n-lock box, as illustrated in Figure 26. However, after water testing, the container we used was not waterproof enough, so another box was chosen to try. The second rectangular box had better performance in waterproofing. Additional details are discussed in Section 2.3.3.4 Water testing. However, this device has become too bulky for the user to wear.

Figure 26 Wearable device placed in the first lock-n-lock box

With the limitation of the wearable device size, the second approach focus on providing a waterproof coating to each component. However, with one layer less protection, the water test for each component is further tested individually and as a whole to ensure they are able to achieve waterproofness. Additional details are discussed in Section 2.3.3.4 Water testing and 2.3.3.5 Waterproofness Testing.

To further reduce the device size, instead of using a mini breadboard with a certain thickness, a though hole PCB, as shown in our prototype from Figure 27 is used to connect the micro-controller, wires and other hardware. Furthermore, by soldering, the electronic conductivity between the component can be ensured with minimised usage of wires.

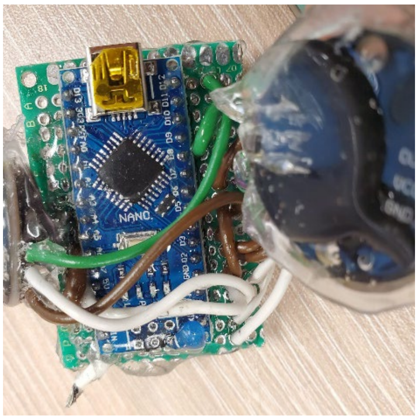


Figure 27 Wearable Device from top-down view illustrating the circuit board

To better mould different components, the components are pot with silicone rubber K-705 individually. The connection ports are further pot and covered with silicone rubber as well. Figure 28 shows the LED above the Arduino Nano is surrounded by a layer of plastic. There is also a layer of plastic cover on top of the Arduino Nano to prevent water from entering the micro-controller. This provides mechanical coverage over the electronic component to avoid the oxidation of components and electric shock. As the paper tape in Figure 28 demonstrates, the USB serial port remains unpotted directly to enhance the modularity and maintained to allow further updates of the program to the board.

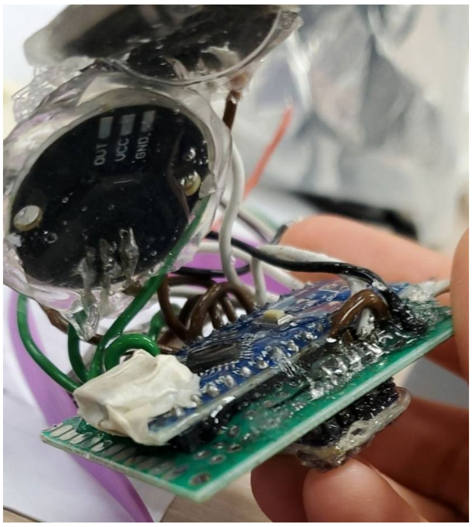


Figure 28 Photo illustrating the potting for the PCB board and the LED component

To achieve minimal size, a custom-made, bendable PCB board will be further improved to reduce the wiring being soldered.

2.3.2.4 Wearable Implementation

In order to achieve higher comfortability while upholding waterproof, a wrist band for mosquito repellent is used. The exact model and type are selected as shown in Figure 29Figure 30. This is chosen for the well-known method of wearing. Using this model, try to avoid the user being allergic to it since the people might have worn it before. People have a better chance to know if they are allergic to this type of plastic. Due to the scope of this project, only a mosquito repellent band is used to illustrate the idea. However, in the production stage, other types of plastic wrist bands allowing adjustable wrist length, such as the watch wristband illustrated in Figure 30, are possible solutions to provide a more extensive range of options for the user to choose from minimal deviation of hardware installation.

Figure 29 A range of colourful mosquito repellent bands [28]

Figure 30 Image illustration of watch wristband [29]

Wires are soldered to the PCB board for connection. The wires are also specially chosen for their flexibility in bending. To fit people having different sizes of wrists, the LED should be not fixed at one location, but it should be located at approximately the top, two sides and the bottom. Having a relatively hard but bendable wire allows the small location adjustment automatically while the user is wearing the device. Some fishing lines are used to loosely tie the LED and the Arduino Nano to the wristband with the same ideology behind it. The fishing line has sufficient tension to hold in place but allows small adjustments for a better user experience.

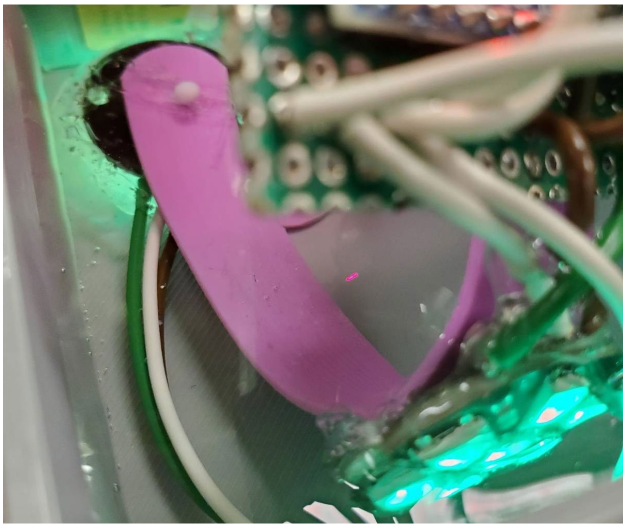


Figure 31 Wrist band with the LED hold in placed with fishing line tied at the start of the wrist (top part in the image)

2.3.3 Testing

2.3.3.1 MAX30100 Accuracy Testing

Testing Purpose:

To better understand the accuracy of the pulse oximeter, MAX30100, testing is done to compare the MAX30100 and the external device.

Apparatus:

Pulse oximeter: MAX30100 are used to detect the blood oxygen data. Arduino Nano and the Micro-B USB cable are used to process and display the data. An external fingertip pulse oximeter is used to take an external benchmarking value.

Testing Procedure:

Place the middle finger on the MAX30100 and clip the index finger to the external pulse oximeter. Then, started the MAX30100 at the same time as the external device.

Observation & Result:

Sample	MAX30100	External Device	Percentage Error (%)
1	97	98	1.020408163
2	97		1.020408163
3	97		1.020408163
4	97		1.020408163
5	97		1.020408163
6	97		1.020408163
7	97	98	1.020408163
8	96		2.040816327
9	96		2.040816327
10	96		2.040816327
11	94		4.081632653
12	94		4.081632653

Table 4 Comparison between the measured value from the MAX30100 and external device

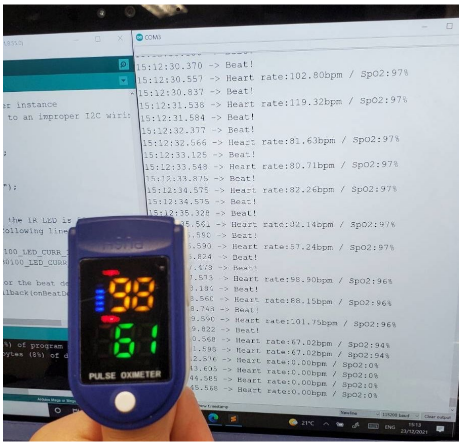


Figure 32 Record of the external pulse oximeter and MAX30100

Compared to the value with the external device, in Table 4 and Figure 32, the external record data was 98% and the MAX30100 was also arriving at 94-97%. This shows that the values are about ± 5% difference. This concludes that the result from the MAX30100 is quite consistent.

2.3.3.2 MAX30100 Detection Location Accuracy Testing

Testing Purpose:

By understanding which location of the hand and wrist the sensor can detect the most accurate value, the wearable device design will set pulse oximetry at that location.

Apparatus:

Pulse oximeter: MAX30100 are used to detect the blood oxygen data. Arduino Nano and the Micro-B USB cable are used to process and display the data.

Testing Procedure:

In the testing, there are six targeted positions to be tested. They include nothing, such as a benchmarking, fingertip (test 1), wrist on the back of the hand, i.e., palm facing down (test 2), wrist vertically below the palm (test 4). The detailed locations of the 5 test points are shown in Figure 33.

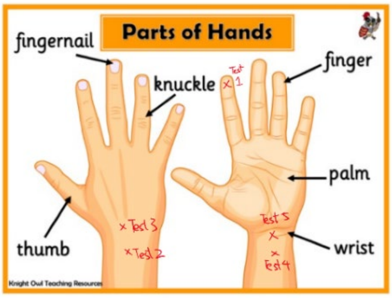


Figure 33 Figure of hand illustrating the testing location [30]

The actual steps are stated as follows:

1. Place the location to be tested on the sensor and record the data of the pulse oximeter.

Observation and Result:

Test Attempt Number	1	2	3	4	5	6	7	8	9	10	Detected Rate
Nothing Placed on the sensor	0	0	0	0	0	0	0	0	0	0	100%
Test 1: Finger	93	98	96	95	95	0	96	95	95	95	90%
Test 2: Wrist (Back of hand)	0	96	0	0	0	0	0	93	0	0	20%
Test 3: On the vessels	0	0	94	0	96	94	0	0	95	93	50%
Test 4: Wrist (Vertically below the palm)	95	96	0	95	0	0	0	0	94	95	50%
Test 5: on the vessel	95	97	0	94	96	96	0	94	0	93	70%

Table 5 Comparison between data collected from different parts of the hand

Conclusion

To conclude, as suggested in Table 5, the wearable device works the best and more value is detected from the fingertip with 90% success. It is followed by a vessel at Wrist (vertically below the palm) of 70% and 50% for both the Wrist (vertically below the palm) and the vessel for wrist (back of the hand). With this consideration, the MAX30100 is placed at the wrist with optimal of facing the vessel. For further development, using a wrist specialised pulse oximeter, such as MAXREFDES103, can improve the accuracy.

2.3.3.3 MAX30100 Detection Underwater Testing

Testing Purpose:

To understand can the device work underwater, the key component for blood oxygen detection underwater has to be tested for functionality. This ensures the abnormal signal can be detected even when the swimmer is swimming.

Apparatus:

About 500ml of water is used to place the droplet and wet the user’s hand. Pulse oximeter: MAX30100 are used to detect the blood oxygen data. Arduino Nano and the Micro-B USB cable are used to process and display the data. A Huawei Band 6 are used to take an external benchmarking value.

Testing Procedure:

Two tests for different levels of water disturbance are conducted. The first test is the water droplet test as illustrated in Figure 34. The second test is placing a wetted hand on the sensor to check the detected result.

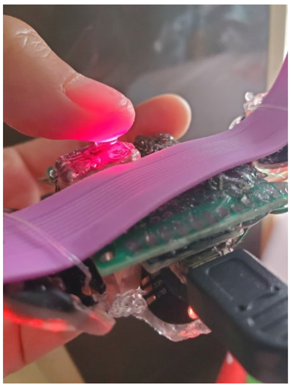


Figure 34 Testing setup for water droplet test

The actual steps are stated as follows:

1. Take the control data comparing the accuracy and successful /rate
2. Adding one droplet of water and place the finger/ wrist on top
3. Place a wetted finger/ wrist on the sensor

Observation and Result:

	MAX30100 (on finger)	MAX30100 (on wrist)	Huawei Band 6 Result
Water droplet Test	0%	0%	No result (tightness is required)
Wetted Hand Test	98%	95%	99%

Table 6 Detection result for MAX30100 with water disturbance on wrist

The testing for the finger is used to eliminate and reduce the error value from the wrist. Even the value detected from the finger is supposed to be more accurate than the wrist. The water droplet test for the finger still fails. This suggested that the failure of the water droplet test was not due to insensitive for the wrist detection.

Table 6 reflects that tight contact between the skin and the sensor is crucial for a successfully detected value. With the water droplet, all three approaches of the finger, wrist and external testing, fail. But with tightly measuring on even wetted skin, the blood oxygen measurement is still a success. Moreover, this test shows that MAX30100 can work even with water. This suggests the further advancement of the project.

Conclusion

To conclude, the wearable device can be placed underwater, but a tightening strip has to be included to ensure the data can be successfully measured.

2.3.3.4 Water Testing

Testing Purpose:

In order to understand the behaviour of the wearable device underwater and evaluate the waterproof capability of different shielding approaches, two water testing were conducted.

Apparatus:

In this experiment, 1 circular lock-n-lock and one rectangular lock-n-lock were used as the container. The unpotted wearable device was the main testing subject to check if it could be waterproof. Some screws and nuts were used as weight to simulate the force of the people wearing.

Testing Procedure:

The first stage of the test venue was conducted within a water basin. The second stage of the test moved to the outdoor pool.

The testing method was mainly as follows:

1. Placing two pieces of dry tissue into the lock-n-lock.
2. Place the lock-n-lock into a large container, i.e., water basin, with sufficient water to cover the entire lock-n-lock.
3. Place the lock-n-lock under the water.
4. Rotate the box for 3 rounds to ensure no side is slipping the water in.

5. Check if the paper gets wet or not.
6. If not, the test is repeated by putting a plastic bag with two pieces of dry tissue inside.
7. If the box content is still dry, some screw and nuts will be added to increase the weight. Then repeated steps 2 – 6 at water depth of 50 cm at the water pool outside LT-J.
8. If the tissues is still dry, the box will be passed to the LED testing.

After passing the two attempts of water-proof testing, the wearable device will be used for further testing with the image processing sub-system.

Observation:

1st Water Testing

Figure 35 Wearable device in the circular lock-n-lock

The first test is conducted by the circular lock-n-lock shown in Figure 35. However, after the first attempt, bubbles are coming out from the top of the lock-n-lock. This indicates that it is not an air seal and not a water seal. With this observation, the tissue got wet, as shown in Figure 36.

Figure 36 Test result for the circular lock-n-lock

2nd Water testing

The second water testing was also done by a lock-n-lock box. This time, the tissue was dry for both the first and the re-test. Thus, the box was further brought into the water pool outside LT-J for water testing. As shown in Figure 36, some metal screws and nuts are placed to increase the weight to sink the lock-n-lock to create some pressure to counteract the buoyancy. Placing the lock-n-lock under 50cm water with weight and rotation, the tissue inside was still maintained in a dry condition. Thus, the test was passed.

Further Testing:

The entire electronic component was then placed into the box and started our image testing. Further testing details are included in Section 2.4.2.2 LED detection algorithm Test.

Figure 37 2nd Water test for the wearable device

2.3.3.5 Waterproofness Testing

Testing Purpose:

To ensure each of the component can withstand water around itself before doing a full water testing for the entire component. The test is also done to check if there might be any hole missing during the potting or any soldering that are not tightly seal.

Apparatus:

A 3D printed box and a water basin that are deep enough to place the whole wearable device in. The pulse oximeter: MAX30100, LED chip: WS2812, 2 18650 batteries, PCB board, Arduino nano.

Testing Procedure:

Each component will conduct two tests. The first is function underwater. The second is whether the water will slip into the component after prolonged contact with water.

1. Power on the testing component to check the functionality of the component
2. Place the test component underwater to see if the component could still function.
3. After 1 hour, get the component out of the water to check is there any hint of water slipping in.

4. Test the functionality of the component by powering the component.
5. Observe the expected behaviour from the tested component.

Observation and Result:

The table below illustrate the testing result for each component. ✓ means the component works as expected, ☑ it works sometimes, ✗ means no response / not working as expected.

	Expected Behaviour	Underwater Functional Test	Off water Functional Test After prolong water contact	Additional Underwater Functional Test (if needed)	Additional Off water Functional Test After prolong water contact (if needed)
Pulse Oximeter: MAX30100	Detect the blood oxygen	☑	✓	✓	
LED chip: WS2812 - top	Alternating Red and Green Light	✔	✔		
LED chip: WS2812 - left	Alternating Red and Green Light	✘	✓ Some part was not tightly soldered.	✔	✔
LED chip: WS2812 – right	Alternating Red and Green Light	✔	✔		
LED chip: WS2812 - bottom	Alternating Red and Green Light	✔	✔		
2 18650 batteries	Power Up the Arduino Board, the LED and MAX30100	✔	✘ Water slipped into the paper part. Some part was not tightly pot, some gaps appear.	✔	✔

Table 7 Waterproofness testing result for each component

Table 7 shows the behaviour of the component underwater and after prolong exposure to water. The experiment shows some of the soldering and potting deflect. However, when the holes are filled, the electronic components are working as expected and perform the ideal behaviour. Due to the multiple wires soldered to the PCB board, the waterproofness of the PCB board and Arduino required special care and effort. Thus, additional to the above test, additional trail of tests are conducted to ensure the waterproofness.

Conclusion

To conclude, during the process, it shows the importance of soldering correctly with the potting. Since water is destructive to electronic component, it is crucial to have additional spare parts as back up and replacement during the design and implementation stage.

2.3.3.6 Blinking Testing

To choose the optimal blinking frequency, three criteria need to be considered:

1. Fast colour change: The quicker the colour change, the more times the algorithm decides if it is our wearable.
2. Fewer yellow frames, of red and green exposure in the same frame, will result in yellow light, therefore not recognized by our algorithm.
3. Distributed yellow frames, a period of clustered yellow frames might result in not being able to detect the LED for a moment.

To test it, we have picked 33ms,67ms,70ms, and 133ms blinking duration, recording in 30fps to test the best result in our setting. To ensure consistency, we have done the followings:

1. All videos are recorded in the same settings
2. The position of the LED and camera is unchanged in all tests
3. The black background is used to reduce reflection

Figure 38 The testing environment

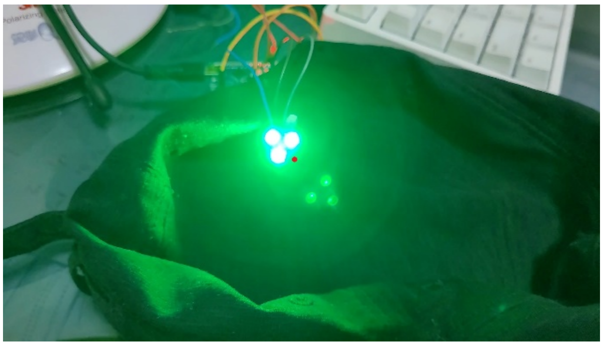


Figure 39 The captured frame and the analysing pixel

Figure 38 shows the environment of testing, and Figure 39 is the resulting frame. The red dot indicates the pixel (x:937, y:394) that we are analysing.

Hue Value

Before further processing, we obtain a result that looks like Figure 39. Since the hue ring is a cycle, we wrap around the data (minus 360 for all data larger than 300) to produce a more readable result like Figure 40. The two bars on the two ends indicate red and green frames (Figure 42, Figure 43), and the bar in between indicates yellow (Figure 44).

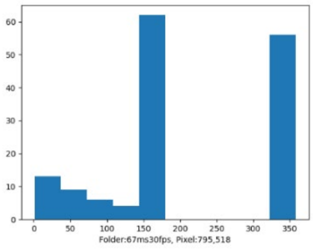


Figure 40 Histogram before postprocessing

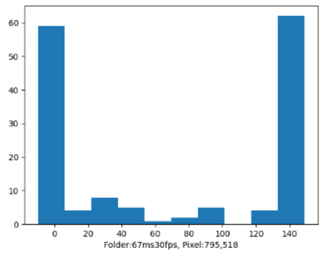


Figure 41 Histogram after postprocessing

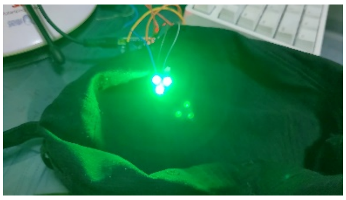


Figure 42 Green frame

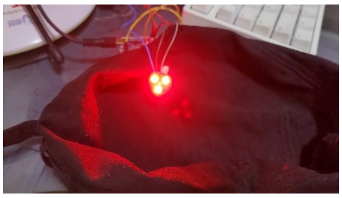


Figure 43 Red frame

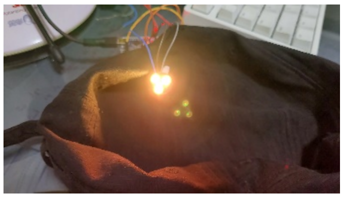


Figure 44 Yellow frame

The results:

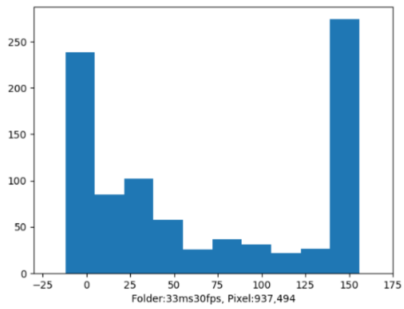


Figure 45 Hue histogram of 33ms

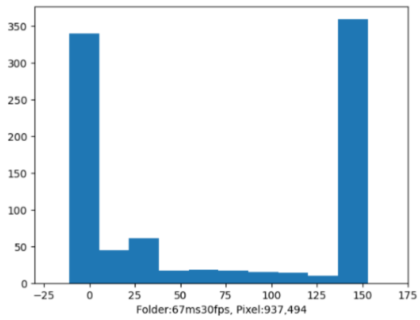


Figure 46 Hue histogram of 67ms

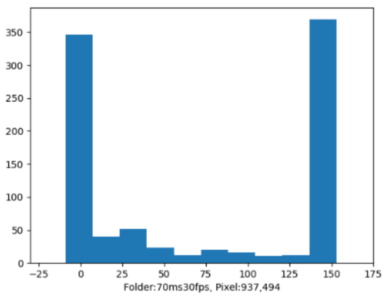


Figure 47 Hue histogram of 70ms

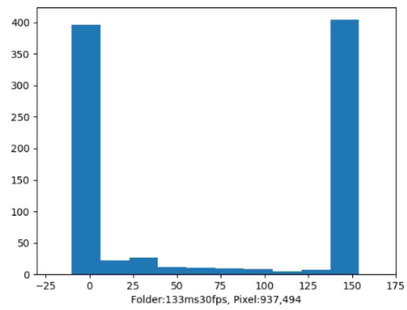


Figure 48 Hue histogram of 133ms

The result shows that using a frequency below Nyquist frequency (Figure 33ms) performs badly, and the lower the frequency, the better.

Overlap Timing

Here we plot the Hue error across time, to investigate the period that yellow frames occur frequently. Comparing Figure 50 and Figure 51, by introducing a 3ms longer blinking duration, the yellow frame occurrence is less concentrated.

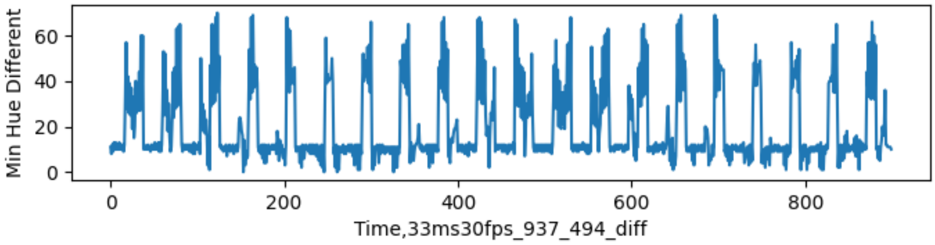


Figure 49 Hue difference along with frames of 33ms

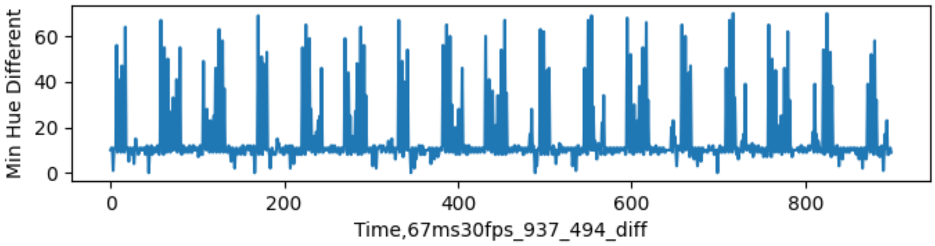


Figure 50 Hue difference along with frames of 67ms

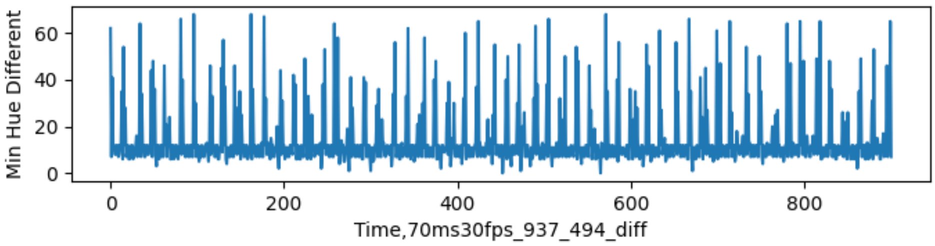


Figure 51 Hue difference along with frames of 70ms

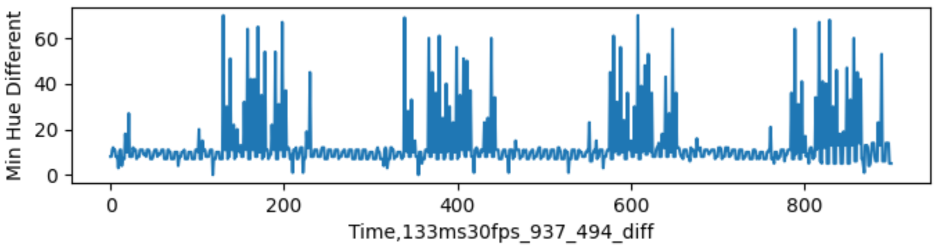


Figure 52 Hue difference along with frames of 133ms

Finally, considering the three criteria mentioned above, we picked 135ms as our blinking duration since it is fast-changing, less overlapping (88% distinct correct colour), and more distributed yellow frames, and here is the testing result (Figure 52, Figure 53).

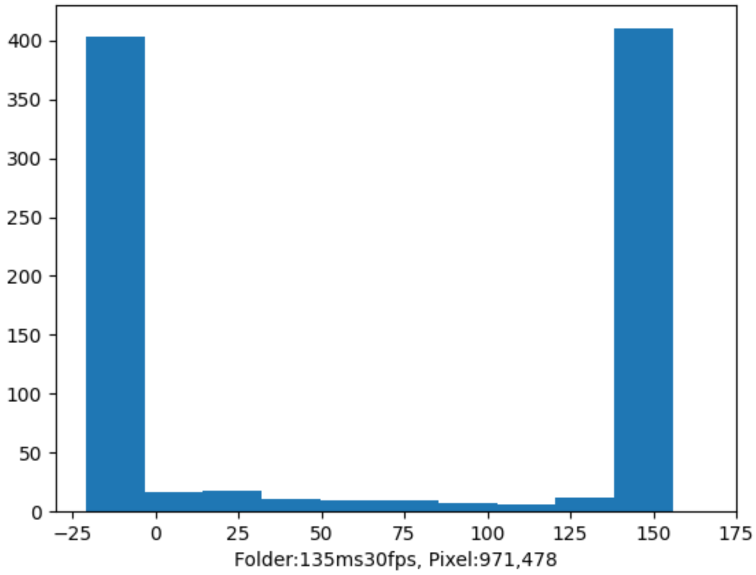


Figure 53 Hue histogram of 135ms

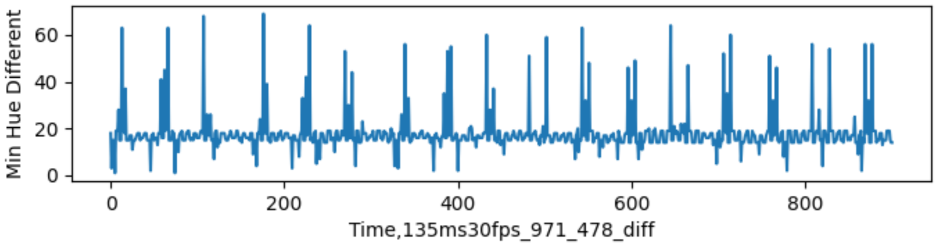


Figure 54 Hue difference along with frames of 135ms

2.4 Image Processing Sub-system

2.4.1 Design

The sub-system consists of 3 algorithms, namely pool corner detection, perspective transformation and LED detection algorithm.

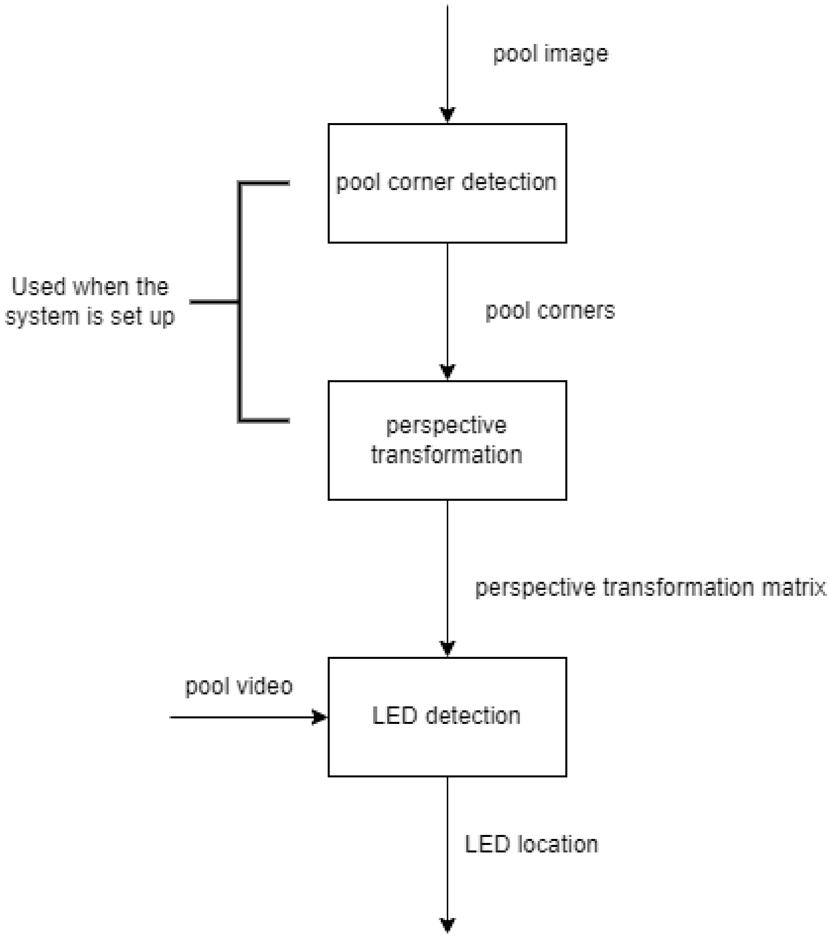


Figure 55 Flow of the three algorithms

The pool corner detection algorithm takes the pool image as input and outputs the 4 corners of the pool. The output is a 4 * 2 list of lists of 32-bit float numbers. The four lists are the four corners of the pool in the order of top left, top right, bottom right and bottom left. Each list has two elements: the x and y coordinates, respectively.

The perspective transformation algorithm takes the four corners of the pool as input and outputs a perspective transformation matrix. The matrix will then be applied to the pool video, as shown in Figure 54, to perform a perspective transform to locate accurate coordinates.

The LED detection algorithm takes the perspective transformation matrix and the pool video as inputs and outputs the coordinates of the LED light as a tuple of 32-bit float numbers. The coordinates will be relative to the top left corner of the pool, i.e. (0, 0) is top left, (width, height) is bottom right.

The corner detection and the perspective transformation algorithms are used once to set up the system.

2.4.2 Implementation

In the following section, we will talk about implementing the algorithms with Python, OpenCV (cv) and Numpy (np) libraries.

2.4.2.1 Pool Corner Detection

To detect the corners of the pool, we need to find out a large blue area in the image and find out four intersection points of the four linear equations of the four pool sides. We believe the pool has a blue colour within a range of hue, saturation and brightness values. Therefore, we would filter the image within a certain range of hue and saturation to extract the blue area out. The large blue area in the image is believed to be the pool. Since the pool is a rectangle, it should look like a quadrilateral in the image. Then we use Hough Transformation to find out the four sides of the pool. The four intersection points, which are the four corners, will then be calculated using the four lines.

pool_corner_detection_with_hough_transform(pool_image):

```
1   denoised_image <- gaussianBlur(pool_image)
2   hsv <- ConvertToHSV(pool_image)
3   upper_limit <- [160, 240, 160]
4   lower_limit <- [100, 100, 60]
5   masked_image <- filter(hsv, lower_limit, upper_limit)
6   masked_image <- erode(masked_image)
7   edge <- canny(masked_image)
8   lines <- hough(edge)
9   strong_lines <- findStrongLines(lines)
10  corners <- findIntersects(strong_lines)
11  final_corners <- Order(final_corners)
12  Return corners
```

Figure 56 Pseudo code of the pool corner detection algorithm using Hough Transformation

As shown in Figure 55, it starts by denoising the image and converting it to an HSV image. Then it masks out the blue pixels with a hue value between 100 and 160, saturation value between 100 and 240, and brightness value between 60 and 160 in Line 5. The values (100 and 160, 100 and 240, 60 and 160) are obtained by investigating the values of several swimming pool pictures with Paint. After that, the masked image is eroded using cv.erode with a kernel size of (5, 5) and 10 iterations to remove the tiny blue clusters in the image. In Line 7, an edged image is obtained using cv.Canny with 300 and 500 as the two thresholds and aperture size of (5, 5).

The edged image is then passed to the cv.HoughLines function to obtain the lines. The lines are expressed in the Polar system [31], i.e.

$$r = x\cos\theta + y\sin\theta$$

The result of the function will be a list of r and θ. However, not every line will be taken into the result. The function only considers those lines which have intersections larger than a threshold. For our algorithm, we set the distance resolution to 1 pixel, angle resolution to 1 degree and threshold to 70.

The algorithm obtains a lot of similar lines, i.e., lines that have similar r and θ. We want to find out the four most distinct lines from these lines. In Line 9, it loops over all the lines until it finds the four lines with distinct r and θ and stores them in a list.

In Line 10, it loops over the strong lines list and calculates the intersection points of each pair of lines. It is done by solving a linear matrix equation. We use np.linalg.solve to solve the equation. Then, it returns the result after ordering them.

This approach finds out the 4 corners of the pool successfully. However, if there is another big blue object in the image, the sky, for example, the result will probably be disturbed. Therefore, when the user uses the corner detection function, he/she should avoid large blue objects in the scene.

2.4.2.2 Perspective Transformation

This algorithm takes advantage of functions given by OpenCV to perform a perspective transformation. Given the four points of the pool in the image (src), our algorithm finds out the four corresponding corners in the warped image (dst). A perspective transformation matrix is then calculated based on these eight points [32]. The matrix should suffice the relation:

$$\left(t_i x_i' \quad t_i y_i' \quad t_i\right)^T = matrix \cdot \left(x_i \quad y_i \quad 1\right)^T$$

where

$$dst(i) = \left(x_i' \,,\; y_i'\right),\; src(i) = (x_i \,,\; y_i),\; for\, i = 0,\; 1,\; 2,\; 3$$

perspective_transform(tl, tr, br, bl):

```
1   width_1 <- sqrt((bl[0] - br[0]) ^ 2 + (bl[1] - br[1]) ^ 2)
2   width_2 <- sqrt((tl[0] - tr[0]) ^ 2 + (tl[1] - tr[1]) ^ 2)
3   width <- max(width_1, width_2)
4
5   height_1 <- sqrt((tl[0] - bl[0]) ^ 2 + (tl[1] - bl[1]) ^ 2)
6   height_2 <- sqrt((tr[0] - br[0]) ^ 2 + (tr[1] - br[1]) ^ 2)
7   height <- max(height_1, height_2)
8
9   pts <- [tl, tr, br, bl]
10  dst <- [[0, 0], [width – 1, 0], [width –1 , height – 1], [0, height – 1]]
11  transform_matrix <- getPerspectiveTransform(pts, dst)
12  Return transform_matrix
```

Figure 57 Pseudo code of the perspective transformation algorithm

In Line 1 and 2 of Figure 56, the distances between the bottom left and bottom right corner, and the top left, and the top right corner is calculated. The longer one will be taken as the width of the image after perspective transformation, as shown in Line 3. Line 5 to 7 serve a similar purpose, which calculates the height of the perspective transformed image.

Then, two lists that store the four corners of the input image and the perspective transformed image are declared in Line 9 and 10.

In Line 11, we use cv.getPerspectiveTransform to obtain the perspective transformation matrix. The matrix is solved using Gaussian elimination with the optimal pivot chosen. Then, it returns the matrix.

2.4.2.3 LED Detection

We locate the LED with template matching. Template matching is a technique to locate an object in an image like a given template. The template is slid over the whole image to calculate a similarity score at each pixel. The location with the best similarity score will be recognized as the object location [33].

To locate the drowning swimmer, we need to detect a blinking LED signal instead of a steady LED signal. The pattern of the blinking signal is red-green-red-green. To detect the blinking signal, we need to repeat the multiscale-template matching process for each light signal. That means we need to detect the red LED for some frames, then the green LED for some frames, then red, then green. If such a pattern is detected at a certain location within some time, that location will be returned by the algorithm.

Multiscale-Template Matching

To detect the LED at different distances, i.e., different sizes on the image, we used a variation of template matching – multiscale-template matching. It allows the program to detect objects of different sizes on the image by scaling the template as different sizes and performing template matching for each scale.

multiscale_template_matching(image, template, matrix):

```
transformed_image <- warpPerspective(image, matrix)
scale_factor <- 0.2, step <- (1.0 - 0.2)/ 10
sum_x <- sum_y <- num_locs <- 0
avg_loc <- (-1, -1)
While scale_factor <= 1.0
   scaled_template <- resize(template, scale_factor)
   score <- matchTemplate(transformed_image, scaled_template)
   best_loc <- getBestScoreLoc(score)
   best_score <- getBestScore(score)
   if best_score < THRESHOLD
   then sum_x += best_loc[0]
          sum_y += best_loc[1]
          num_locs += 1
   endif
```

```
15        scale_factor <- scale_factor + step
16        Endwhile
17     If num_locs > 0
18     then avg_loc <- (sum_x / num_locs, sum_y / num_locs)
19     endif
20     Return avg_loc
```

Figure 58 Pseudo code of the LED detection algorithm with multiscale-template matching

Line 1 of Figure 57 performs a perspective transformation on the image using the matrix obtained from the previous perspective_transform algorithm. It is done by cv.warpPerspective. Line 2 initializes the scale_factor and the step to increment the scale_factor for resizing the template. Line 3 and 4 initialize the variables to 0. These variables will then be used to find out the average location.

Line 5 starts the loop. In lines 6 to 9, it first scales the template with the scale_factor. Then it calculates the similarity score at each location on the image and finds out the best location and the corresponding best score. Line 7 uses cv.matchTemplate to calculate a similarity score at each pixel of the image. We use the normalized square difference to calculate the similarity score. Thus, the location with the best score should have the smallest similarity score. Line 8 and 9 use cv.minMaxLoc to find the best location and the corresponding best score.

Line 10 filters the match by checking if the similarity score is lower than the threshold. We did this since we would like to ignore the matches with a high similarity score. The threshold was set as 0.007. This was set according to empirical observation. We observed that if it is a mismatch, usually it has a score that is higher than 0.007. If the location has a score lower than the threshold, the x and y coordinates will be added to sum_x and sum_y, respectively. num_locs will also increment by one, as stated in lines 11 to 13.

After the loop ends, line 17 checks if something is detected in the loop. If something is detected, the num_locs should be larger than 0. Then the average location will be calculated using sum_x, sum_y and num_locs. An average location was adopted in this approach since we want the result to be more robust. If we only take one of the locations calculated in the loop, it could be a mismatch with a location that is way off the LED light. We observed that there might be one or two mismatches located in the loop, but most of the matches were accurate. Therefore, by calculating the average location, we can obtain a more accurate result. The function returns the average location eventually.

Blinking LED Detection

To detect the blinking LED light, the program first searches for a LED light. Once it locates a LED, it starts looking for the red-green-red-green pattern at that location for a period. If it finds a pattern there, it reports the location.

blinking_led_detection(video, temp_red, temp_green, pool_size, matrix):

```
detect_red <- true
finding_pattern <- false
frame_count <- num_iterations <- switch_count <- 0

for each frame in video do
   if detect_red
   then loc <- multiscale_template_matching(frame, temp_red, matrix)
   else  loc <- multiscale_template_matching(frame, temp_green, matrix)
   endif

   if loc is not (-1, -1)
   then
          current_real_loc <- to_real_loc(loc, pool_size, frame.shape)
          if num_iterations == 0
          then finding_pattern <- true
                 initial_loc <- current_real_loc
          endif

          if initial_loc is close to current_real_loc by 0.6
          then
                 frame_count += 1
                 if frame_count > REQUIRED_NUM_OF_FRAME
                 then frame_count <- 0
                        switch_count += 1
                        detect_red <- not detect_red
                 endif
         endif
   endif

   if finding_pattern
   then num_iterations += 1
   endif

   if num_iterations == FRAME_LIMIT or switch_count == 4
   then
          if switch_count == 4
          then output initial_loc
          endif
          frame_count <- num_iterations <- switch_count <- 0
          finding_pattern <- false
   endif
```

```
42      endfor
```

Figure 59 Pseudo code of the blinking LED detection algorithm

Line 1 to 3 of Figure 59 initialize the variables. The detect_red variable is a boolean to check if the program is tracking for the red LED light. The finding_pattern variable is a boolean to check if the program is finding the red-green-red-green pattern at a particular location. The frame_count variable keeps track of the number of frames in which the program finds a red or green LED light. The num_iterations variable keeps track of the total time lapse of looking for the pattern in terms of the number of frames (one iteration equals work done for one frame).

Line 5 starts the loop to work on each frame of the video input. Line 6 to 8 detects the red LED or green LED using the multiscale_template_matching algorithm introduced previously. Line 11 checks if something is detected. If true, the program finds out the corresponding actual location of the pool, as stated in line 13. The pseudo-code is as follows:

to_real_loc(loc, pool_size, picture_size):

```
1       real_x <- loc[0] / picture_size[1] * pool_size[1]
2       real_y <- loc[1] / picture_size[0] * pool_size[0]
3       Return (real_x, real_y)
```

Figure 60 Pseudo code of image to real location conversion

Both the perspective transformed image and the pool are rectangular. It is not difficult to observe the following ratio relation:

$$\frac{x_p}{w_p} = \frac{x_i}{w_i}$$

where x_p is the actual x coordinate on the pool,

x_i is the x coordinate on the image,

w_p is the actual pool width,

w_i is the image width.

By rearranging the terms, we obtain

$$x_p = \frac{x_i}{w_i} w_p$$

The equation is similar for the y coordinate. Line 1 and 2 of Figure 59 make use of this concept.

After getting the real location, line 14 in Figure 58 checks if this is the first LED signal detected. If true, it starts finding pattern by setting find_pattern to true and stores the real location in initial_loc.

Line 19 checks if the newly detected location is close to the initial_loc. If true, it increments the frame_count. Then it checks if frame_count has reached the required number of frames (REQUIRED_NUM_OF_FRAME) to detect the LED light. If true, it resets frame_count to 0, increments switch_count and sets detect_red to the opposite value so that the program can switch to detect another colour of the LED light. We assumed the LED light switched to another colour every 0.135 seconds and fps was 30. Thus, the LED light switches to another colour every 4.05 frames. Therefore, we set the required number of frames to be 3(1 frame as a buffer).

Line 30 increments num_iterations if the program is finding the pattern. Line 34 checks if the number of iterations has reached the limit (It was set as 17 since 4.05 frames for each light) or if a pattern is found. If true, it resets the variables to 0 and resets finding_pattern to false, as stated in lines 39 and 40. If a pattern is found, it outputs the first location it found. Then it starts over again to look for a LED light.

2.4.2 Testing

Because of the pandemic, we could not access a real pool. Thus, we used a small pool (2m * 3m) to test our algorithms.

2.4.2.1 Corner Detection and Perspective Transformation Algorithms Test

For corner detection and perspective transformation algorithms, we tested them with swimming pool images. The pool is rectangular, and the four corners are visible in the image. The pool is filled with water which appears blue in the image. Figure 60 is an example of the testing data.

Figure 61 Example of testing data for corner detection and perspective transformation

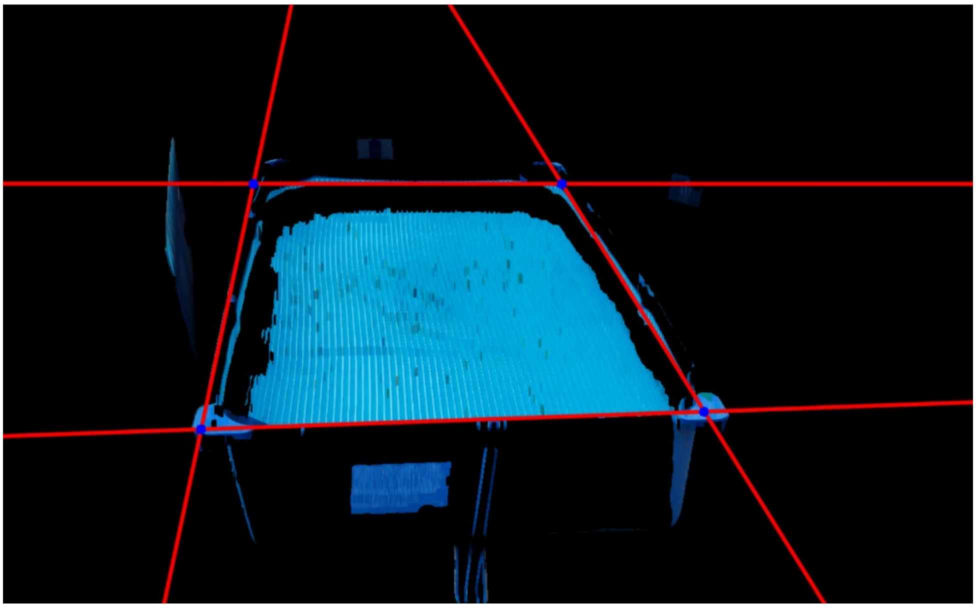


Figure 62 Image after corner detection

Figure 61 is the image after corner detection. The four red lines are the Hough lines calculated by the Hough transformation. The four blue dots are the intersections of the four lines.

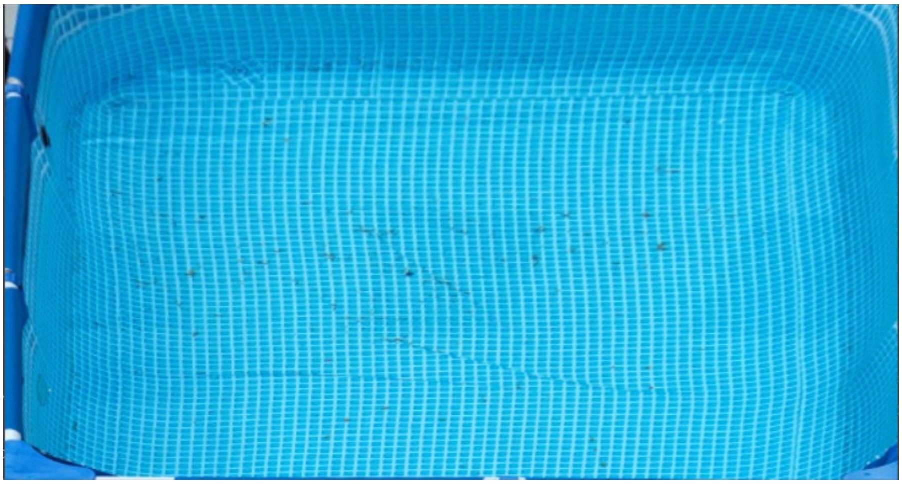


Figure 63 Image after perspective transformation

Figure 62 is the image after perspective transformation. The obtained image is a “top-down” view of the pool in the original image.

The algorithms work fine with this pool. The corner detection algorithm obtains the four corners of the pool, and the perspective transformation algorithm transforms the image into a “top-down” view image successfully. Because of the pandemic, we could not access a real pool. Thus, we used a small pool to test our algorithms. Since there are not so many valid (rectangular swimming pool with four corners visible, high resolution) images online, we couldn’t find more valid images of different pools to test our algorithms. Fortunately, a standard swimming pool has a standardized appearance, and it does not look so different from this small pool. Therefore, we thought it would be fine to test our algorithms with this small pool.

2.4.2.2 LED Detection Algorithm Test

Multiscale-Template Matching Test

For the LED detection algorithm, we tested it with images of the LED light in the 2m * 3m pool. Both red and green LED light tests were performed. Both above and beneath water surface tests were performed. The LED light was placed above or beneath the water surface at around 5 cm. The LED light was placed at different locations, and we labelled the locations with (x, y) coordinates, where the upper left corner of the pool is (0, 0) and the lower right corner of the pool is (2, 3). For each case, an error value was calculated using the Euclidean distance between the real location and the detected location. If the LED light is detected and the error of detection is within one arm length(~0.6m), then we will consider it a successful detection. Otherwise, it fails the detection. Since using images is easier to quantize the results, we tested the algorithm using images instead of videos. The following tables are the summary of the results.

Red LED Detection Test					
Case	Above/Beneath Water Surface	Real Location	Detected Location	Error	Successful✔ /Failed ✘ Detection
***1**	/	/	/	/	✔
2	Above	(0.5m, 2.5m)	(0.535m, 2.561m)	0.070m	✔
3	Above	(0.5m, 2m)	(0.490m, 2.079m)	0.080m	✔
4	Above	(0.5m, 1.5m)	(0.513m, 1.521m)	0.025m	✔
5	Above	(0.5m, 1m)	(0.435m, 1.023m)	0.069m	✔
6	Above	(0.5m, 0.5m)	(0.445m, 0.651m)	0.161m	✔
7	Above	(0.5m, 0m)	(0.471m, 0.025m)	0.038m	✔
8	Beneath	(0.5m, 2.5m)	(0.481m, 2.713m)	0.214m	✔
9	Beneath	(0.5m, 2m)	(0.516m, 2.358m)	0.358m	✔
10	Beneath	(0.5m, 1.5m)	(0.461m, 1.817m)	0.319m	✔
11	Beneath	(0.5m, 1m)	(0.445m, 1.428m)	0.432m	✔
12	Beneath	(0.5m, 0.5m)	(0.384m, 0.946m)	0.461m	✔
13	Beneath	(0.5m, 0m)	(0.474m, 0.321m)	0.322m	✔
Total Above Water Surface Average Error					0.074m
Total Beneath Water Surface Average Error					0.351m
Total Average Error					0.212m
***No LED in the image of case 1, succeeds if it detects nothing.**					

Table 8 Summary of results of testing red LED

Green LED Detection Test					
Case	Above/Beneath Water Surface	Real Location	Detected Location	Error	Successful✔ /Failed✘ Detection
***1**	/	/	/	/	✔
2	Above	(0.5m, 2.5m)	(0.477m, 2.552m)	0.057m	✔
3	Above	(0.5m, 2m)	(0.484m, 2.113m)	0.114m	✔
4	Above	(0.5m, 1.5m)	(0.452m, 1.682m)	0.188m	✔
5	Above	(0.5m, 1m)	(0.458m, 1.132m)	0.140m	✔
6	Above	(0.5m, 0.5m)	(0.503m, 0.532m)	0.032m	✔
7	Above	(0.5m, 0m)	(0.523m, 0.076m)	0.079m	✔
8	Beneath	(0.5m, 2.5m)	(0.519m, 2.687m)	0.188m	✔
9	Beneath	(0.5m, 2m)	(0.523m, 2.349m)	0.350m	✔
10	Beneath	(0.5m, 1.5m)	(0.523m, 1.885m)	0.386m	✔
11	Beneath	(0.5m, 1m)	(0.458m, 1.369m)	0.371m	✔
12	Beneath	(0.5m, 0.5m)	(0.490m, 0.744m)	0.244m	✔
13	Beneath	(0.5m, 0m)	(0.494m, 0.406m)	0.406m	✔
Total Above Water Surface Average Error					0.102m
Total Beneath Water Surface Average Error					0.324m
Total Average Error					0.213m
***No LED in the image of case 1, succeeds if it detects nothing.**					

Table 9 Summary of results of testing green LED

From Table 8 and Table 9, we can observe that the average error above the water surface is smaller than that beneath the water surface. The errors above the water surface are usually below 0.1m while those beneath the water surface are around 0.35m. The larger errors beneath the water surface may be due to the refraction effect of light. Yet the errors are still acceptable. Since all errors are lower than 0.6m. In other words, if a kickboard is shot to the detected location, it is reachable within one arm's length(~0.6m). In conclusion, the multiscale-template matching algorithm works fine.

Blinking LED Detection Test

We tested the blinking LED detection algorithm with a video shooting at a blinking LED light. We expected the program would output the coordinates when it detected a pattern.

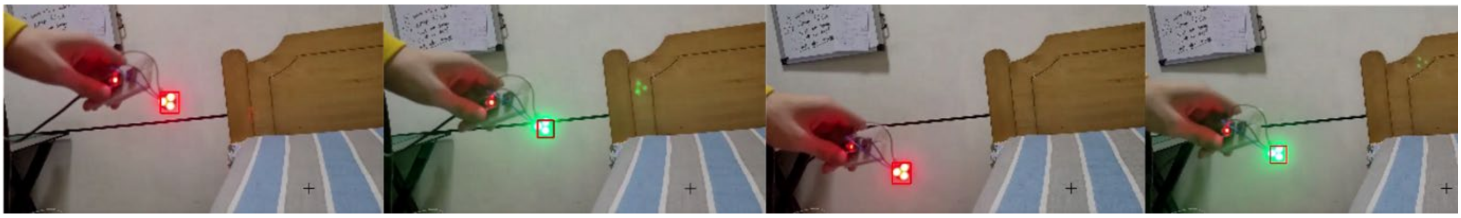


Figure 64 screenshots from the blinking test video

Figure shows the blinking pattern of the LED light. From left to right, it changes between red and green. We can see that the program marks the LED light with a red square, which means it detected the LED light successfully. It reports the coordinates after the detection of the pattern. In conclusion, the blinking LED detection algorithm works fine.

2.5 Server Sub-system

This system is responsible for integrating sub-systems and providing an interface for the lifeguard.

2.5.1 Design

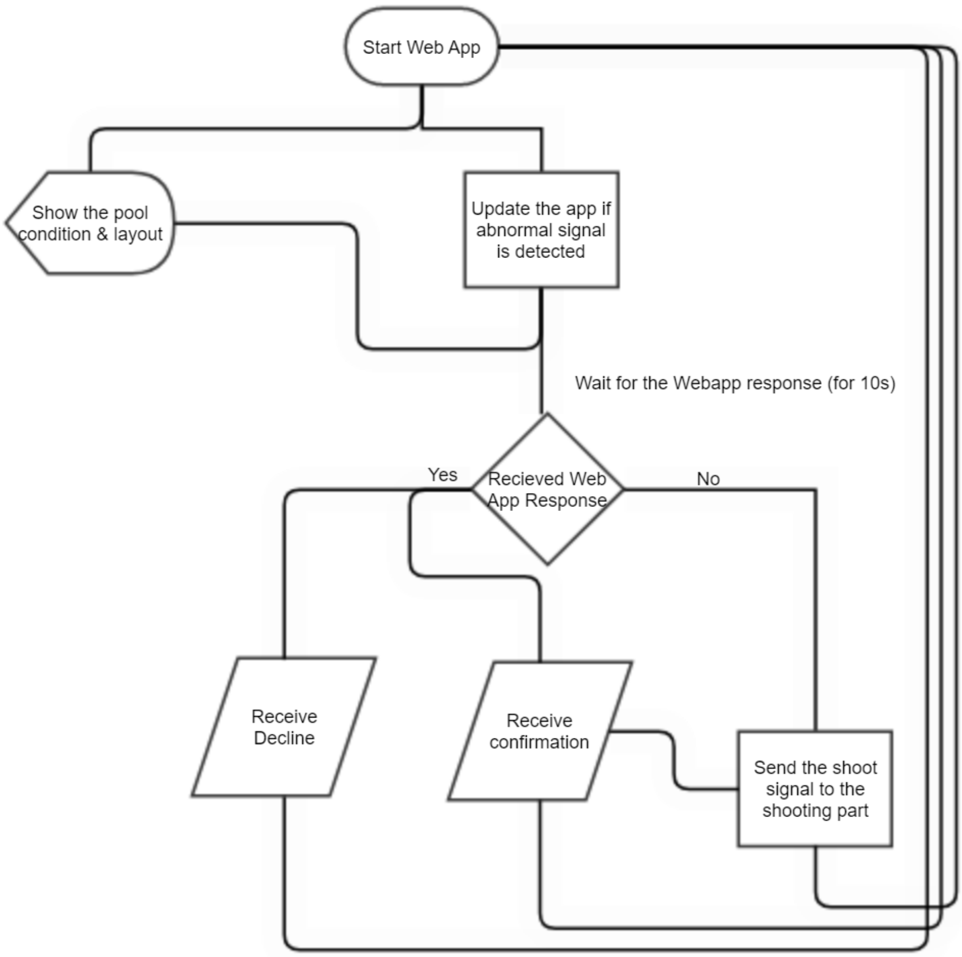


Figure 65 Flowchart representation of the drown-alarming sub-system

Figure illustrates the flow of how the web application will function. Upon receiving the warning signal, the application will be updated with the victim's location and displayed on the screen. An alarm signal will also be sent out to catch the lifeguard's attention.

Then, there will be a 10-second countdown for the lifeguard to confirm whether there is a drowning detection. This would allow the lifeguard to make a professional decision. If the lifeguard has confirmed the drowning condition or no response is received within 10 seconds, the Kickboard throwing sub-system will follow up to throw the kickboard.

Moreover, there would also be an input system for the lifeguard to enter the location of the drowning case. This allows the lifeguard sitting on the lifeguard tower to be able to control the Kickboard Mechanism in case the detection system misses any drowned victim, but the lifeguard is not able to reach the victim due to not being able to leave the lifeguard tower.

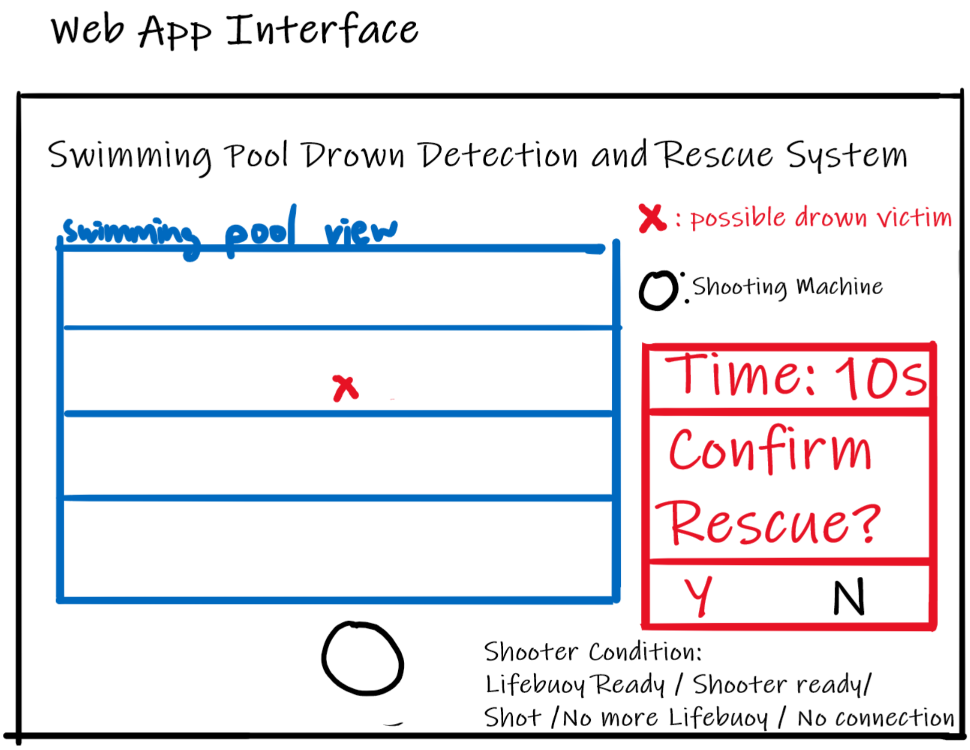


Figure 66 Preliminary UI Interface for the web application

For the web application, due to the design factor of being easy to use and not causing an additional burden for the lifeguard, a minimalistic approach is taken. As shown in Figure 65 only the pool view, the location of the victim, and the shooter stage (prepared, no kickboard, shoot) will be shown.

The monitoring webpage can be accessed by smartphone, tablet, or laptop, but touch screen devices will be the main service platform. The user interface will be designed as touch-screen-friendly as it has a low learning slope and is intuitive to interact with.

The website will contain two web pages to serve different purposes:

- Monitor Page: The footage captured will display here with a notation of alert. The lifeguard can answer the prompt to stop the false alarm (depending on the setting) and touch the screen to do override shooting.
- Setting Page: Set the pool size, pool edge, and config the alarm procedure.

2.5.2 Implementation

In the implementation, the system can further break apart into the server itself and the connection between different sub-systems, which are 1) the server, 2) webcam, 3) adaptation to image processing algorithm, 4) web App, 5) communication to the robot as shown in Figure 66. The following will explain their design and implementation in detail.

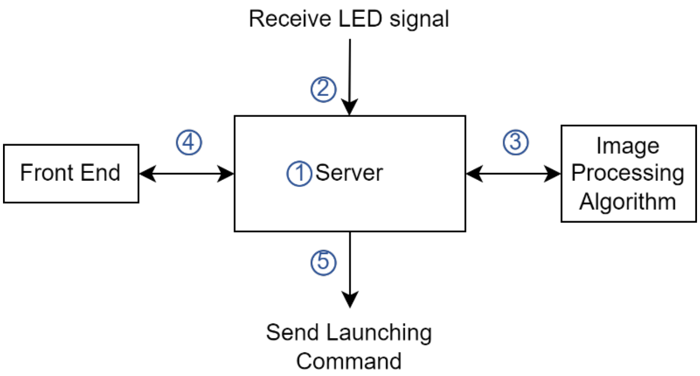


Figure 67 Simplified server architecture

1. <u>The Server</u>

The server is running on a VM from HKUST. The operating system is Ubuntu 20.04.1 LTS. Since the software currently used is available on a multi-platform, we also set up an identical server on the Windows 10 platform for testing purposes. There is a minor issue running on Windows 10. The npm package used for capturing the webcam into an image file is slow. It can be fixed by using another package, but we keep the package for consistency of the code between the Linux platform and the Windows platform for development. Therefore, it will take not much further development to run the server on different platforms to adapt to already present swimming pool equipment.

The backend logic runs by nodeJS, and we use different packages to implement functions and adapt to sub-systems. Mainly we use the following packages: “node-webcam” for webcam capturing, “child_process” for Python integration, “express” for web-related logic, and “socket.io” for streaming webcam to the client.

2. <u>Webcam</u>

We use an Android app, DroidCam, for wireless webcam input to the server. It provides 640x480 resolution for the free version and 1920×1080 resolution for the paid version DroidCamX Pro, and it is available on both Linux and Windows. We choose this method as our webcam input because it is portable, easy to set up, low cost, and easier to find suitable mounts for a smartphone. With that said, the whole system also supports any webcam input, such as an IP camera, a wired DSRL camera for high-quality video, or OBS virtual camera as the adhesive for older systems.

3. Adaptation to the image processing algorithm

Since the server uses node.js, the image processing algorithm uses Python. We need to bridge the two languages to allow the server to use the algorithm. The implementation is by using the "child_process" package. It provides node.js a way to call a Python program with arguments and receive the output by flushing system stdout. Figure 67 shows the basic use of “child_process”, and based on that, we can run image processing algorithms on nodeJS.

```
//code on node js
const spawn = require("child_process").spawn;
const pythonProcess = spawn('python',["./public/test.py"]);
//stdout event handler
pythonProcess.stdout.on('data', (data) =>
{  console.log(data.toString());
});
```

```
#code on python, in test.py
print("Hello Python, Hello JS.")
sys.stdout.flush()
```

Output from terminal

Figure 68 Code of Adaptation to the image processing algorithm

The image processing sub-system provides several functions, including 1) Corner Detection, 2) Perspective Transform 3) LED Detection, 4) Coordinate Mapping and they are utilized in different places.

1. Corner Detection: This function is called when using the "Auto" function of the calibration settings. It returns the detected four corners of the pool, and the server forwards it to the user.

2. Perspective Transform: This function is called when the user submits the four corners. It updates a config file containing the transformation matrix for later use.

3. LED Detection: This function is called when the server starts. It detects the LED's location and returns to the server. A helper function will read the configs files periodically to update the transformation matrix, pool size and sensitivity settings.

4. Coordinate Mapping: This function is called when a shoot/aim command is sent. It converts the shooting position (image coordinate) to real-world coordinates based on the saved transformation matrix. The kickboard will then be shoot using a real-world location.

4. Web App

Back end

The backend is handled by nodeJS package "Express" and uses "EJS" as the template engine to provide website services. Express is a web backend framework of node.js that features routing function, logic handling, and generating front-end HTML by EJS. EJS is like PHP, EJS is a template that can take in parameters and generate a customized webpage when a request is received.

The webcam streaming functionality is implemented by the "socket.io" package. First, the client and server will establish a socket.io connection. Socket.io is an instance of WebSocket used for real-time data transfer in the HTTP service. Then, "node-webcam" captures a frame from webcam input and returns base64 data. When node-webcam captures a frame, it will send it out to the client by socket.io.

Front End

For the front end, we use HTML, CSS, JavaScript, SVG, and jQuery to develop. Here we will describe the streaming function, stream display function, UI design, and iOS optimization.

Webcam streaming function

Before the data comes from the server, it displays an animated gif indicating that the stream is loading. Figure 68 shows the implementation of webcam streaming on the front end. When new image data comes in from the socket connection, JavaScript will update the current image with the new image data. We also obtain the size of the image as it will be used when the user inputs the shooting position, but the size is usually constant in our use case. Therefore, we comment out the function to reduce calculation.

```
const socket = io();//Start socket connection
const image = document.getElementById("webcam");//The image node
socket.on("image",(data)=>{//When new data comes in, this will run
                  if(data){
                  //getImgSize(data);
                  image.setAttribute("href",data)
      }
});
```

Figure 69 Code of webcam update on client-side

Stream Display

Figure 70 Monitor page

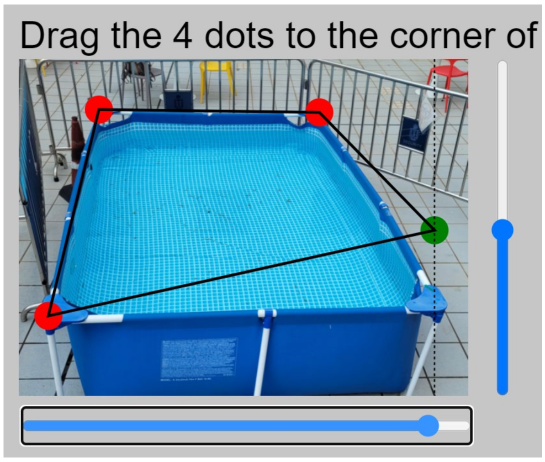


Figure 71 Cropped setting page

This part presents the image data received from the backend. We use an SVG image node to hold the data. Every time the client receives valid image data from socket.io, it will update the "href" attribute of the image node, therefore updating the latest image from the webcam. The reason for using SVG image node instead of HTML "<img>" tag, is because SVG allows us to draw different shapes on top of the image without obstacles to the position input from the user. In Figure 69, the monitor page demonstrates drawing a detection box and kickboard throwing SVG animation. Another advantage of using SVG is the animation of SVG is handled by the browser. Therefore, it is smoother and will not consume the resource of the single-threaded JavaScript engine. In Figure 70, the setting page demonstrates the combined use of SVG and JavaScript. JavaScript draws auxiliary lines in SVG to assist the user positions the dots precisely.

UI design

The UI is optimized for 4:3 ratio large screen devices, for example, most iPad devices. It is because most old cameras are captured in a 4:3 ratio. So using a 4:3 screen can maximize the use of both camera and the screen. As shown in Figure 71, the screen is fully utilized on the iPad mini. But since we applied responsive web design. Figure 72 and Figure 73 demonstrate it is also flexible to other screen and webcam input aspect ratios, and the gap will fill with the background colour.

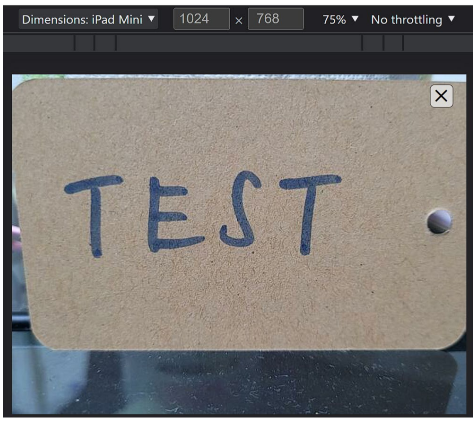


Figure 73 Display of monitor page on Galaxy Fold using Chrome DevTools

Figure 74 Display of monitor page using 16:9 webcam input on iPad mini using Chrome DevTools

Figure 72 Display of monitor page on iPad mini using Chrome DevTools

Monitor Page

The main screen is minimalistic that only includes the video stream and a universal button that can navigate between the monitoring page and the setting page. This helps the lifeguard focus on monitoring the swimming pool. In Figure 74, Figure 75, when drowning is detected, a colour-changing rectangle from yellow to red will show on the position indicating how long the victim has been drowning. In Figure 76, if the timer is nearly running out, an alert will pop up and ask the lifeguard for a shooting command. In Figure 77, if auto shooing is triggered, an alert message and the shooting location will be shown until the lifeguard responds.

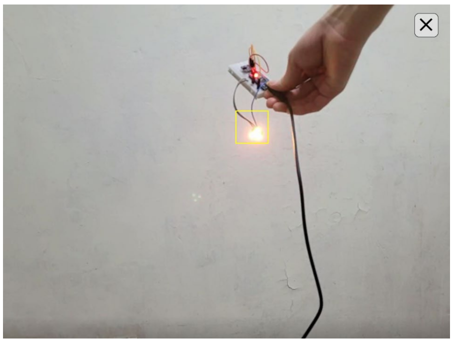


Figure 75 Monitor page starts alerting drowning

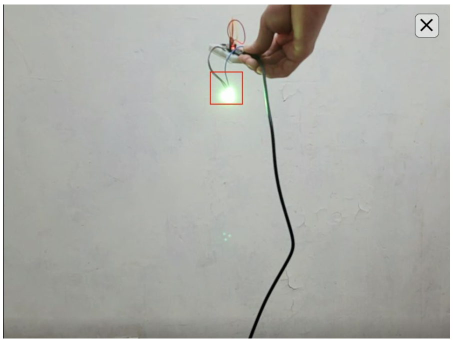


Figure 76 Alert gradually becomes red after a duration

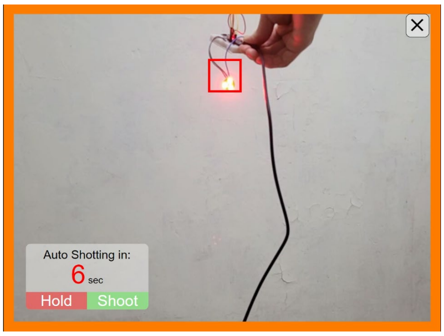


Figure 77 Auto shooting alert pops up

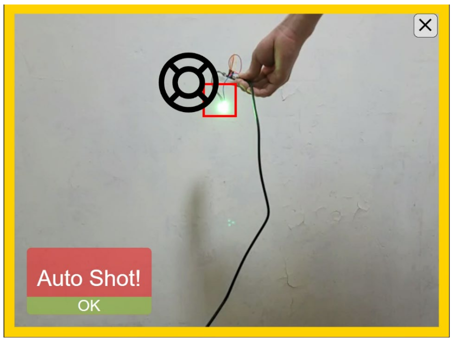


Figure 78 Alert will be kept until the lifeguard respond

Other than auto shooting, the lifeguard can also perform two actions: 1) Select shooting, and 2) Manual shooting.

1) Select shooting: Lifeguards can click on the alert box to perform rescue decisions before the auto alert occurs. If lifeguards open this tab and idle, auto shooting will still trigger when the timer counts to zero.
2) Manual shooting: If the victim is not detected, a lifeguard can click on the victim and perform manual shooting. A green box will indicate the targeted area, and a prompt will be shown for giving the command. The green box will hide after 15 seconds.

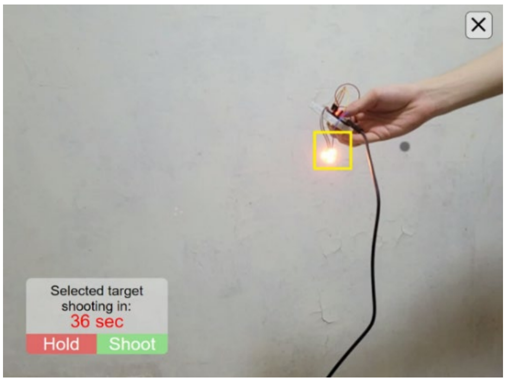


Figure 79 Select shooting

Figure 80 Manual shooting

If multiple notifications pop up, non-urgent notifications (notifications that can reopen) will be hidden to show urgent notifications (i.e., automatic shooting alerts). If a lifeguard decides to perform a manual shooting during an automatic fire alert, we consider them equally important. So, the manual shooting notification will overlap on the top of the automatic shooting alert.

The alert system can take in multiple alerts at the same time. It will perform tracking by moving distance, direction, and magnitude. Figure 80 shows the system classifies three generated detections with different speeds and moving directions.

If multiple auto shooting alerts raise simultaneously, the execution order is managed by a semaphore. In Figure 81, the system will warn the first one (show the alert box is in bold), and the rest will wait for the previous alert to clear. For the upcoming alerts, the system will give the lifeguard another 10 seconds to decide.

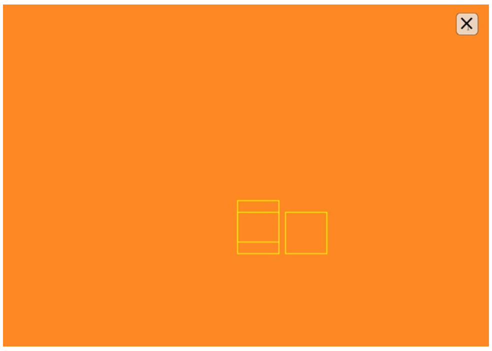


Figure 81 Multiple alert testing

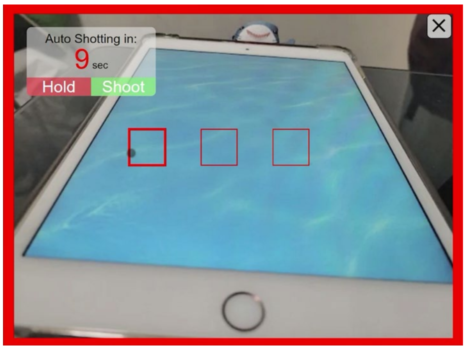


Figure 82 Multiple auto shooting alert ("Shotting" is updated to "shooting" for consistency")

Setting Page

The setting page is optimized for large-screen mobile devices and designed to be touch screen friendly, interactive, and intuitive. On the settings page, the user can configure several different settings, including 1) Alarm Timer, 2) Calibration, 3) Deployment and 4) Monitoring.

1) Alert Timer (Figure 82)

   This defines the total time it will wait until the auto shooting. For instance, if it is set to 30, the alert box will turn red in 20 seconds and prompt an auto shooting alert, and after another 10 seconds, the auto shooing procedure will run. The UI uses coloured buttons to indicate the options, which are easy to click on and intuitive.

2) Calibration (Figure 83)

   This defines the four corners of the swimming pool. Users can click on the "AUTO" button, to auto-detect the four corners of the swimming pool. Then further adjust it with the two sliders. Here we use horizontal and vertical range input for users to enter the X-Y of the position to avoid fingers blocking their view. When the users adjust the input, JavaScript draws the auxiliary lines to help users align to the pool.

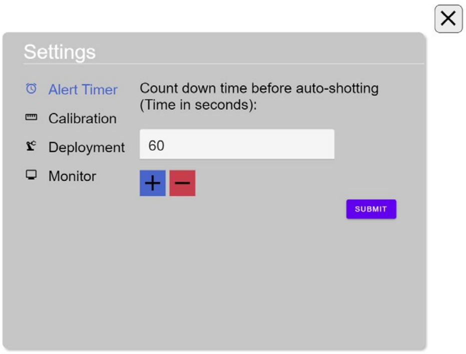


Figure 83 Setting page – Alert Timer

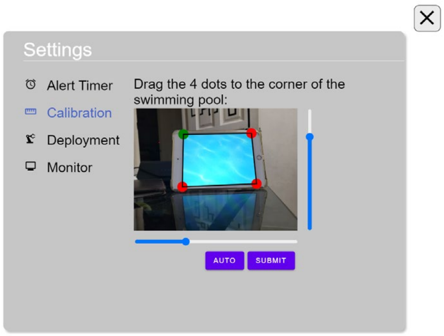


Figure 84 Setting page – Calibration

3) Deployment (Figure 84, Figure 85)

   This defines the environment settings, including pool size and where the robot is located. Changing the pool size ratio will change the 'pool' shape, as shown.

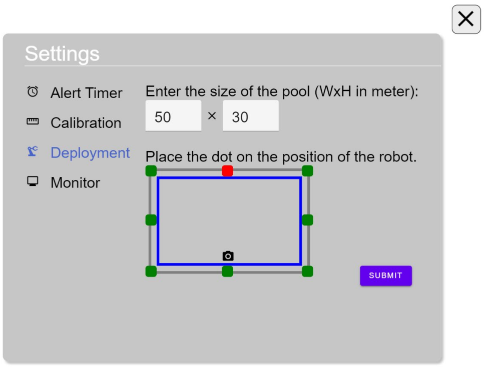


Figure 85 Setting page – Deployment (Pool shape in 5:3)

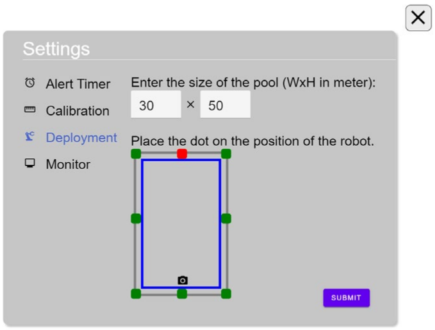


Figure 86 Setting page – Deployment (Pool shape in 3:5)

4) Monitor (Figure 86, Figure 87)

This defines where the notification pops up to suit the different environments and user preferences.

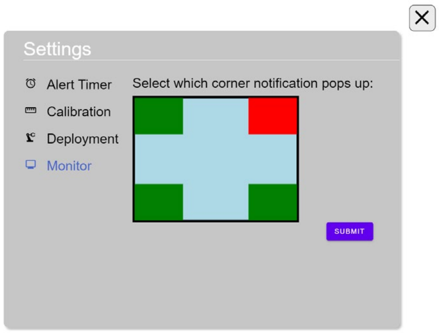


Figure 87 Setting page – Monitor

Figure 88 Setting page –Monitor (Result)

iOS optimization

First, we maximize the usage of the screen. Usually, browsers on iOS like Safari and Chrome have a navigation bar at the top that cannot be removed by the webpage, which results in a black bar around the screen. Therefore, we use a specific browser, "Full-Screen Web Browser App", to avoid the navigation bar.

Second, we eliminate the 300ms button delay on iOS. To avoid delay, we identified the user-agent right after the page loaded. If it is a mobile device, we insert the "ontouchstart" event handler, and else we insert the "onclick" event handler.

Third, we disable the double-click zoom feature on mobile devices by setting the meta element. Since we will click multiple times on a button on the setting page, it will trigger the double-click zoom function, which is inconvenient for the user.

Finally, we add periodic updates for the settings GUI. Because iOS is lazy in "onchange" event updates, it makes UI less responsive.

Different with design

1. The monitor page is further simplified to provide maxim usage of the screen, and minimum distraction to the lifeguard.
2. The front-end design was intended to use React for UI development. After further findings on React, we find that the benefit of React is adding program logic to generating UI, therefore reducing repetitive codes. But our application only consists of a few pages. It is excessive to use React and will cost more loading time to load React library.

5. <u>Communication with robot</u>

In the communication with the robot, we use the internet as the channel and JSON as the data transfer format. After the user clicks on the swimming pool and sends a shooting command, nodeJS will run the first part of the code in Figure 88. It will calculate the projectile motion and obtain the shooting angle and velocity, then those into a JSON file accessible through the internet. The second part of Figure 88 demonstrates how the robot receives the command. The robot will fetch the JSON file through the network, decode the JSON file and turn the JSON into usable variables. And the robot can send back data using an HTTP post request.

```
//code on nodejs
var shootingPara = projectileMotion(req.query.x-200,req.query.y);
var shootingData = {
      angleH: shootingPara[0]*57.2958,
      angleV: shootingPara[1]*57.2958,
      velocity: shootingPara[2],
      shoot: req.query.shoot,
      note: req.query.note
};
fs.writeFileSync(path.resolve('./public', 'shootingData.json'),
JSON.stringify(shootingData));
```

```
//code on the robot
//Getting JSON
HTTPClient http;
http.begin("http://theAddressOfTheServer/test.json");
http.GET();
String JSON = http.getString();
http.end();
//Parse JSON
StaticJsonDocument<192> doc;
DeserializationError error = deserializeJson(doc, JSON);
//the error handling is cut out for better demonstration.
//Got the variables
int x = doc["x"];
int y = doc["y"];
bool shoot = doc["shoot"];
const char* note = doc["note"];
```

Figure 89 Code for robot communication

2.5.3 Testing

The user test concerns responsiveness, user-friendliness, and clarity. We will time the webcam to client delay and the whole process from the guard clicking on the screen to the bot shooting the kickboard to test the responsiveness. And we will conduct a user test by inviting a lifeguard to evaluate the user-friendliness and clarity.

Responsiveness

To test the responsiveness, we video-recorded the following events. Then we measure the time it takes to finish the event by video editing software.

1) Webcam responsiveness (166ms - 433ms)

   From real-world event to local monitor: 166ms

   From real-world event to monitor in worse case (from home to HKUST and back to home): 433ms

   Therefore, real-world events to lifeguard's monitor would take 166ms~433ms, depending on the network.

2) LED Detect responsiveness(67ms)

   From LED start blinking to shown on Monitor (in a local network): 233ms

   Since its analysis is based on webcam input, the delay of the portion is about 67ms (233ms-166ms)

3) Robot Responsiveness (66ms)

   From shooting command to receive by the robot (in a local network): 66ms

In conclusion, the responsiveness is enough for time-critical use. The system will introduce a delay optimally, 0.3 seconds and under 0.5 seconds.

User-friendliness and Clarity

We consulted a lifeguard to evaluate the UI design, we have asked for his point of view[4] in terms of ease of use and clearness, here are his comments:

<u>Is the UI (i.e., the position of buttons, display position, etc) easy to use?</u>

Lifeguard: In terms of user-friendliness, it seems good, but we need to deploy it to know its real ability, especially in emergencies.

<u>Is the instruction(including text and colour) clear?</u>

Lifeguard: The icons and instructions are clear, but lifeguards should learn how to use them beforehand.

[4] The question here is simplified, multiple choice is given in the actual questions to avoid preconceived answers.

<u>Will using the system introduce a negative effect or unreasonable design in the field?</u>

Lifeguard: We need to apply it to see. If the pool size is an irregular shape, the software may not be able to recognize it.

In conclusion, the UI works well in theory, is easy to use, and is clear, but the actual result requires on-field testing. And for rebuttal of the irregular shape pool problem, the user can treat the pool as a rectangle(box the irregular pool), and manually set the "four corners" for the pool.

2.6 Shooting Sub-system

This sub-system is responsible for shooting the kickboard to the destination. This section only contains a brief introduction to the shooting robot. Please refer to mech's FYP report for further details.

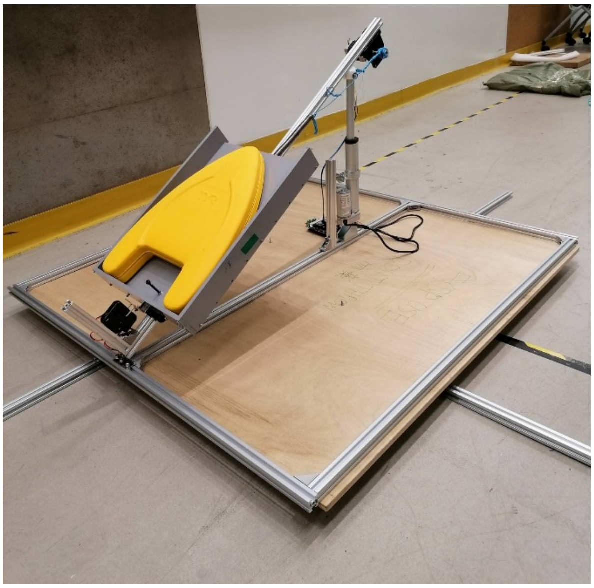


Figure 90 Shooting robot - Side view

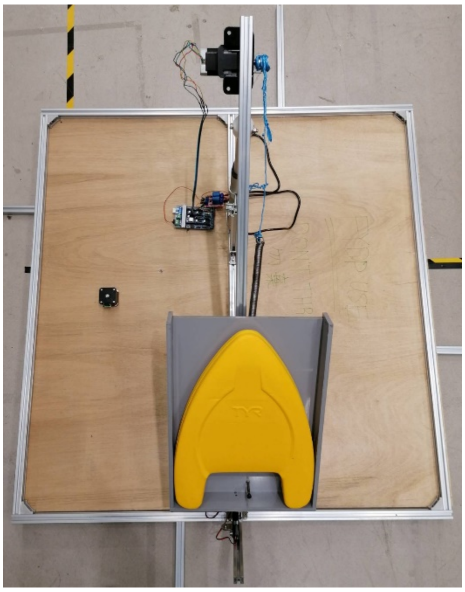


Figure 91 Shooting robot – Top View

Pan and Tilt sub-system

The purpose of the pan and tilt sub-system is to pan the robot horizontally to the correct shooting direction and tilt the robot to the correct shooting angle. The Pan mechanism consists of a motor, spur gears, and a bearing. In Figure 92, the small gear is installed on the shaft of the motor. When the motor turns, the small gear will use the large gear to turn the wooden board that supports the components on top. In Figure 91, it is shown that the tilt mechanism consists of a linear actuator to push the aluminium extrusion vertically to adjust the shooting angle.

Shooting sub-system

The purpose of the shooting mechanism is to store elastic potential energy in the form of a spring and use it to launch the kickboard to the location of the drowning victim. The shooting sub-system consists of a spring, a rail made with a v-slot aluminium extrusion, a PVC plastic container for the kickboard, two gantry plates to support the container, the spring, and the motor to pull the spring, and the kickboard release mechanism. During the shooting process, the kickboard release mechanism will hold the gantry plate in place as shown in Figure 93, then the motor will begin to pull the spring to store elastic potential energy as shown in Figure 91. Once the desired length has been pulled, the kickboard release mechanism will release the gantry plate and the plates will be launched forward along the rail. The forward momentum will launch the kickboard out of its container.

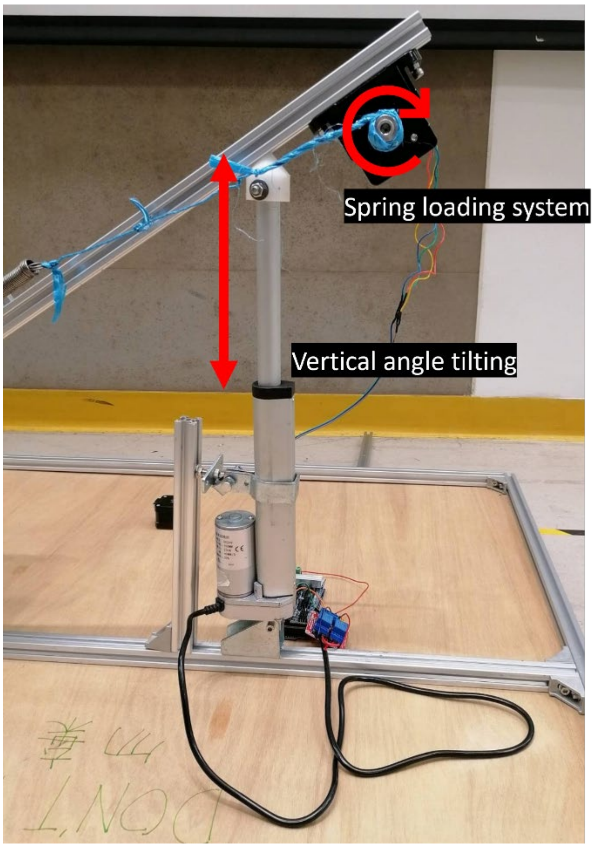


Figure 92 Tilting and spring loading mechanism

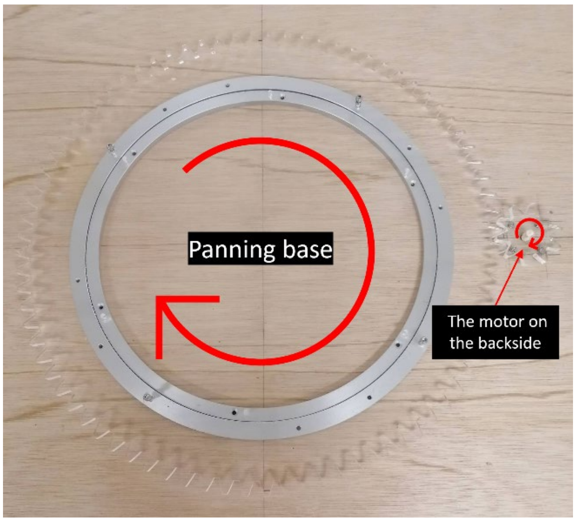


Figure 93 Panning mechanism

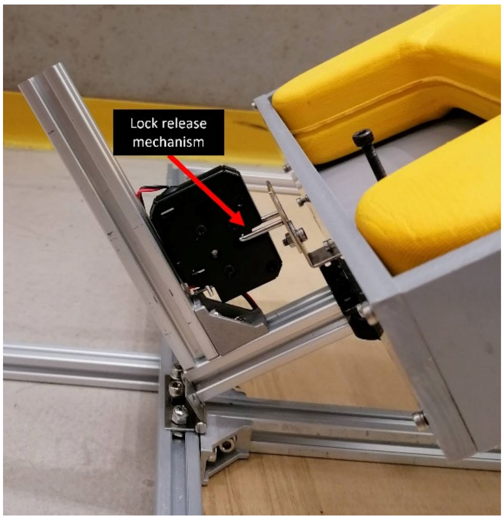


Figure 94 Kickboard release mechanism

Control Mechanism

In the mechanical system, Arduino Mega 2560 is responsible for receiving the pre-Aim command, i.e. the first part of the JSON, the coordinate of the victim with required tilting angle from the server. This information is used to initiate the shooting signal, control the rotation of the shooting platform and the shooting rail and the shooting motor. As suggested in Figure 94, the Arduino Mega is waiting for the signal from the network by ESP01. Then, with the information of the pre-shooting signal, the robot will start to set up according to the parameter received from the JSON. In the panning mechanism, the robot will first set up the small gear for the shooting direction, i.e. the small gear. In the spring load mechanism, the robot will also set up the shooting motor to tighten the spring. Then, in the tilting mechanism, the robot will set up the linear actuator to set the shooting angle. Upon receival of the shooting command, the lock release mechanism will be triggered with a low-level activation on the 12V relay. The will then be released. In order to facilitate system maintenance and future development, for instance adding additional functions in the future. In the Arduino programme, different control of the components are written in class.

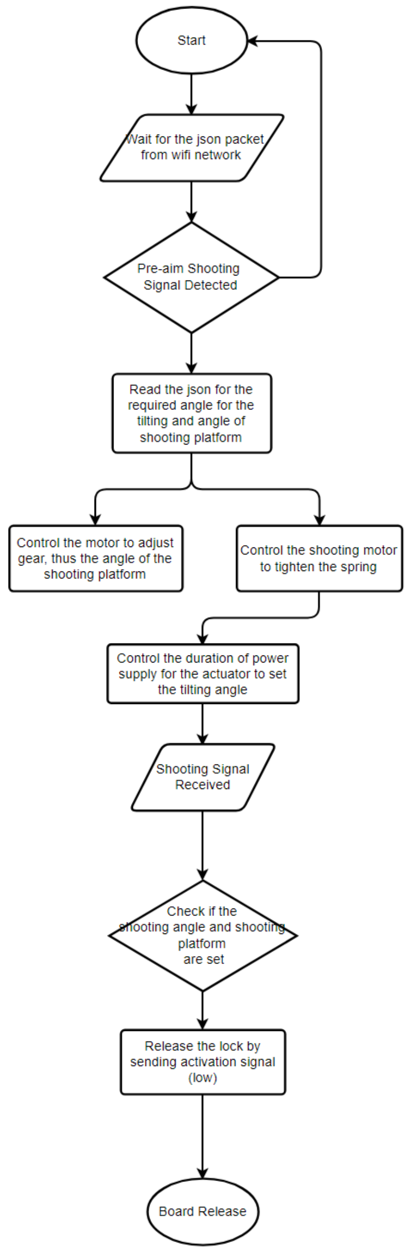


Figure 95 Flowchart illustration for the Mechanical System

2.7 Evaluation

Here we will compare our goal (objectives) and the result to evaluate the system.

Objective 1: To build a wearable device that identifies potential drowning victims

To identify potential drowning victims, with the high accuracy in blood oxygen detection, and the stable result measurement at the wrist, our system could detect the potential drowning victim with a low error rate of about 5% for every second. Upon the detection, the abnormal detection will trigger the blinking LED within 10 seconds to trigger the image processing system. There are some limitations to the wrist detection mechanism of the pulse oximeter MAX30100.

For the wearable device, considering the user conformability and the waterproof ability, the device is able to stay underwater for a period of time while successfully detecting the blood oxygen. This suggests the ability of the wearable device to be a workable device in swimming is accomplished.

Objective 2: To locate the drowning victim accurately with a sub-system

For corner detection and perspective transformation, our system could detect the corners and obtain a “top-down” view image after perspective transformation successfully.

For LED detection, our system could locate the LED light with an average error of around 0.35m, which is within an arm length(~0.6m). That means a kickboard could be shot to the location where that is reachable by the drowning swimmer. Our system could also detect blinking LEDs to avoid misdetection of a swimmer dressing in a swimsuit with a similar colour.

Objective 3: To build a user-friendly application to alert and assist the lifeguard in making drown and rescue decision

1) Responsiveness: The responsiveness is good, as the UI always respond instantly, and the data transmission time takes under 0.5 seconds in total.
2) User-friendliness: From an interviewed lifeguard's point of view, it seems to be easy to use, and further testing in the field is preferred.
3) Clarity: From an interviewed lifeguard's point of view, the instructions are clear.
4) Compatibility: The system has good Compatibility. The system can support the different sizes of screen and webcam inputs, and it can run on Windows and Linux.

Objective 4: To Integrate the Detection and Warning System with the Mechanical Kickboard Throwing System

1) Shooting accuracy: The kickboard should be launched accurately to the location of the drowning victim

2) Speed: The entire shooting process starting from once the location coordinates have been received should be three seconds or less

3) Safety: The shooting process and the components used in the system should take into account the safety of the others in the swimming pool. For example, no spinning parts or explosions that have the potential of injuring nearby swimmers and the lifeguard.

3. Discussion

3.1 Result

The result has largely accomplished the goals we set.

For the wearable sub-system, we successfully collected user data for drowning detection, built a water resistance prototype, and conquered signal attenuation problems by using the visible spectrum. The method of detecting drowning may not be accurate, since the detection requires biological knowledge. But we believe it proved the concept of using a wearable device to detect the biological parameter and warn the other sub-system with the detected value accordingly. With the design, we believe the data collected is sufficient to perform accurate drowning detection.

The image processing sub-system is also successful. It can detect a pool's location, locate LEDs, and return the LEDs to a real-world location with good accuracy. Although due to the pandemic, we could not test the algorithm in a standard size pool, the algorithm could successfully detect swimming pools from online images and the concept of detecting LED would work the same on larger pools.

And finally, the server sub-system has met our goal, it integrates all components and can execute rescue in under 0.5 seconds after drowning is detected, and the pre-aim command helps reduce aiming time. Therefore, the server satisfies the time-critical requirement. The UI components look good and reasonable in theory, but practical testing is needed to truly evaluate its performance in real-world situations.

3.2 Suggested Camera Settings

Although we didn't implement the system on a standard size pool, we would like to suggest some camera setting that optimizes the system. In order for the LED detection to work best, the following setting should be considered:

1) Shutter speed: The shutter speed should be relatively fast. It should fall below the LED blinking speed and can capture a fast-moving LED.

2) Aperture: A small aperture should be used to increase the depth of view to include the whole swimming pool, but at the same time, it should be as large as possible to allow more light in.

3) Polarizing filter (CPL): CPL can reduce the reflection from the water surface to give a clearer view of the pool, but it halves the light intensity, so it will be a trade-off. If the environment has good lighting, a CPL is suggested.

4) Hyperfocal distance: The camera should place at a hyperfocal distance to get maximum depth of field.

5) Good lighting: Image quality largely relies on good lighting. Since we need a faster shutter speed, smaller aperture, and a CPL filter, good lighting in the swimming pool is essential to obtain the best result.

3.3 Limitation

There are some limitations in the current design of the system.

For the blood oxygen detection in the wearable device, despite the MAX30100 pulse oximeter can achieve high accuracy at the fingertip, when it comes to the wrist, the sensor has to be placed at the vessel to obtain accurate measurements. Obtaining accurate blood oxygen data from the wrist is the main issue in the pulse oximeter detection field. But with new technology like MAXREFDES103, the newer MAX version specialises for wrist pulse oximeter detection. By using MAXREFDES103, the accuracy of blood oxygen detection can be improved significantly in future development.

The second limitation of the system lies in the current design of the wearable device is a bit bulky. With a smaller sized battery and a custom flexible PCB, the size can be further minimised and enhance comfortability.

For corner detection, the program might be distracted by large blue objects. Since the program might consider those blue objects as parts of the pool since they are also extracted by the program. The results might not be accurate then. When the user uses it, he/she should avoid large blue objects other than the pool.

For LED detection, there will be larger errors when detecting the LED underwater. The error is due to the refraction effect of light. However, the error is acceptable. The error is around 0.35m which is within one arm's length(~0.6m). The kickboard is reachable within this range.

For the server sub-system, when running the server, webcam video streaming is laggy on windows because of the npm package's problem. Since our main development platform is on Linux, we decided to keep it in the development stage for code consistency. It could be fixed by producing another variant dedicated to the Windows platform.

3.4 Future Development

There is three possible further development that we think it is most worth doing, they are:

Customize flexible PCB

The wearable is still a prototype, but it has great potential to shrink in size and make it more comfortable for swimmers. By customizing PCB for the wearable, as in Figure 95, the wearable can be tinier and have better water resistance. Also, it can wrap around the user's wrist, which makes it more comfortable.

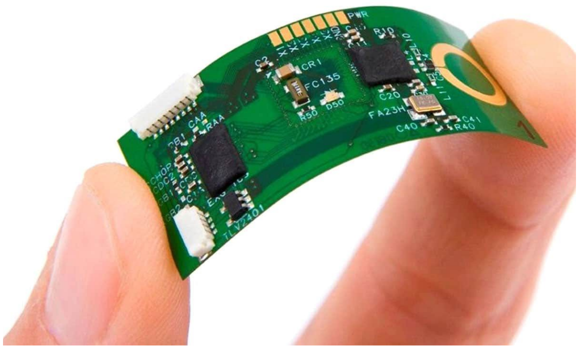


Figure 96 Flexible printed circuit board [34]

Drowning Detection

Drowning detection in the field of biology, we are not professionals in this discipline. By working with biologists, we can build a more scientific and accurate drowning detection.

Visible Light Communication (VLC)

Currently, the blinking LED can only tell us the victim is drowning and where is he. By using VLC technologies, we can transmit the heart rate, blood oxygen, and user identity to the server. These are useful in understanding situations, prioritizing rescue orders, and calling emergencies.

4. Conclusion

As a closure of this FYP report, this swimming pool drowns detection and rescue system targets to address and alleviate the pressure on the lifeguards from the lifeguard shortage issue by providing a supportive drown detection and rescue system. This system aims to identify the drowning victim by the Internet of Thing, i.e., the wearable device, locate the drowned swimmer by image processing and feedback to the lifeguard by the server. This system further provides a user-friendly interface for the lifeguard to control and shoot the kickboard to actively rescue the drowning victim on the computer tablet. To sum up, the system was realized by the use of the Internet of Thing, image processing, server design, user interface design and robotics technology.

Despite the aforementioned limitation, there are some feasible designs, if not, ready-made implementations that are able to support future improvement. With consideration and comparison to other existing research and commercial solutions, our system inherits most of the positive features from the five systems we compared. This includes but is not limited to detecting the swimmer in the pre-drown stage, low implementation cost, detecting the swimmer with computer vision for a swimming pool size coverage and most importantly providing an active rescue system.

In comparison with other current solutions, there are some further improvements we introduced. Firstly, using the visible light as the warning signal allows accurate positioning and solve signal attenuation problem by using the characteristic of visible light and image processing system. Secondly, the quick-release mechanism for the shooting robot takes less than a few seconds to shoot the kickboard to reach the victim.

In conclusion, this project aims to introduce a complete architecture that includes 1)drowning detection, 2)localization, 3)monitoring, and 4)active rescue to support the lifeguard drowning rescue operation inside a swimming pool. Last but not least, the project is with no intention to replace lifeguards or to be used to represent medical usage as the project developed without explicit medical guidance.

4.1 Acknowledgement

We would like to express our gratitude to our supervisors Prof. Ma and Prof. Tsoi, who gave numerous guidance and advice and supported us throughout the project. We also would like to thank Kris, who gave us a lot of ideas and guidance while defining the topic, Miss Noor who gave us a lot of advice in writing our report and Mr Wong, who gave us precious advice as a lifeguard.

5. Project Planning

5.1 Project Work Plan

Year	Date	Task	Personal In Charge
2021	12 September	Internal Proposal Deadline	Loreen
	30 September	Collection for Detection Sensor Data	Don
	15 October	Functionality Testing for the Detection System	Loreen
	22 October	Deadline for Individual Ethics Essay	Loreen, Don, Edwood
	29 October	Deadline for Monthly Report	Don
	30 October	Water Testing for the Detection System	Loreen
	30 October	Testing for the Positioning System	Edwood
	15 November	Functionality Testing for Web App	Don
	26 November	Deadline for Monthly Report	Edwood
	30 November	Functionality Testing for the Positioning System	Edwood
	30 December	Full Testing in Swimming Pool for Detection System	Loreen
2022	12 January	Deadline for Monthly Report	Loreen
	30 January	Seek User Opinion from Lifeguards for Web Application	Don
	15 February	Combined Testing for the Detection System & Positioning System	Edwood
	28 February	Combined Testing for the Positioning System & Kickboard Throwing System	Loreen
	1 February	Internal Deadline for Progress Report	Loreen
	14 February	Deadline for Progress Report	Edwood
	6 April	Internal Deadline for Final Report	Loreen
	20 April	Deadline for Final Report & Self-Assessment Report	Don
	1 May	Internal Deadline for Video	Edwood
	11 May	Deadline for Video	Don

Table 10 Project Work Plan

With a new groupmate CHENG Hoi Wai joined at the start of October 2021, the person in charge section is updated.

5.2 Human Resources

Name	Skills / Ability
LEUNG, Lok Yan Loreen	Stronger in have an overview in a problem, Have communication skills for different parties, Understand both skills in Computer Engineering and Mechanical Engineering, Interested in testing data collection and programming ☐ Can help in project management ☐ Can coordinate between different departments and resources ☐ Can integrate the mechanical and computer system ☐ Responsible for the drown detection system
CHAN Tsz Ho	Stronger in language skills, Rich in hands on experience on Mechanical parts ☐ Can proofread and become second reader for Computer Engineering Documents ☐ Can direct and guide the progress of prototype building and testing
CHENG Hoi Wai	Stronger in programming skills and software documentation Experienced in learning new programming platform, especially image processing ☐ Responsible for the image processing sub-system ☐ Can help in software architecture documentation
CHONG Shing Tung	Rich experience in electronic components and web programming, Swimming ☐ Can initiate the electronic control and design the web application ☐ Can support on-scene testing
LEE Ka Hei	Experienced at design and drawing, Thinks with a careful and detailed mind Patient ☐ Responsible for the robot design and solid work drawing for Mechanical and gives suggestion for Computer UI design ☐ Can give second option for the user experience ☐ Responsible for the calibration and Mechanical Debugging

Table 11 Our skills and ability

With a new groupmate CHENG Hoi Wai joined at the start of October 2021, the Human Resources Section is updated.

5.3 Project Monitoring

We will be monitoring the project using the RACI method (Responsible, Accountable, Consulted, Informed). With a new groupmate CHENG Hoi Wai (Edwood) joined at the start of October 2021, the responsibility is slightly modified and redistributed. Since some of the design and research are done before October, some entries for CHENG Hoi Wai are labelled as X.

Overall Version

Task/Name	LEUNG, Lok Yan Loreen	CHAN Tsz Ho	CHENG Hoi Wai	CHONG Shing Tung	LEE Ka Hei
Design and Research					
Communication Between Different Parties	R	A	I	I	C
Analyse on Different Topic	I	R	X	A	C
Defining Topic	C	R	I	I	A
Feasibility Study	R	C	I	A	I
Integration with CPEG/Mech Portion	A	I	C	R	C
Finalization					
Testing, Debugging, and Tunning	I	A	R	C	C
UI Improvements	C	I	C	R	A
Documentation					
Writing Proposal	A	C	X	R	I
Monthly Report	R	I	A	C	I
Progress Report	C	C	R	A	I
Writing Final Report	R	C	A	C	C
Preparing Presentation	C	I	R	A	C
Making Video Trailer	I	C	R	A	I

Table 12 Project Monitoring Table for the overall project (Both CPEG and MECH)

CPEG Version

Task/Name	LEUNG, Lok Yan Loreen	CHAN Tsz Ho	CHENG Hoi Wai	CHONG Shing Tung	LEE Ka Hei
Design and Research					
Study on ROS	A	C	X	R	I
Analyze on Different Approaches	R	C	X	A	I
Feasibility Study	R	C	X	A	I
Literature Review	A	C	X	R	I
Implementation					
Testing Electronic Component	R	C	C	A	I
Decide Detection Process of Drowning	R	I	C	A	C
Implement Detection System	R	C	A	C	I
Testing Positioning System	A	I	R	C	C
Implement Positioning System	C	I	R	A	C
Integration with Mech Portion	A	I	C	R	C
Web App Development	C	I	A	R	C
Finalization					
Implement Calibration System	R	I	C	A	C
Testing, Debugging, and Tunning	A	I	R	C	C
UI Improvements	C	I	A	R	C
Documentation					

Writing Proposal	A	C	X	R	I
Monthly Report	R	I	A	C	I
Progress Report	C	C	R	A	I
Writing Final Report	R	C	A	C	I
Preparing Presentation	C	C	R	A	I
Making Video Trailer	I	C	R	A	C

Table 13 Project Monitoring Table for the CPEG component of the project

MECH Version

Task/Name	LEUNG, Lok Yan Loreen	CHAN Tsz Ho	CHENG Hoi Wai	CHONG Shing Tung	LEE Ka Hei
Design and Research					
Feasibility Study	C	A	I	I	R
Built varies robot models	C	R	I	I	A
Literature Review	C	R	I	I	A
Analysis of Different Launching Approaches	R	C	I	I	A
Implementation					
Launcher 3D Modelling	C	A	I	I	R
Acquiring Launcher Parts	C	R	I	I	A
Implement first Launcher	C	R	I	I	A
Testing Different Launching Approaches	R	C	I	I	A
Acquiring Pan Tilt Mechanism Parts	R	A	I	I	C
Testing Pan Tilt Mechanism	R	A	I	I	C
Integration with CPEG Portion	R	A	C	C	I
Finalization					
Testing, Debugging, and Tunning	A	R	C	I	C
Documentation					
Project Planning	R	C	I	I	A
Progress Report	A	R	I	I	C
Final Report	A	R	I	I	C
Making Video Trailer	C	R	I	I	A
Oral Presentation	R	C	I	I	A

Table 14 Project Monitoring Table for the MECH component of the project

5.4 Gannt Chart

Below are the three Gannt Charts for overall, Computer and the Mechanical Engineering part.

Overall version

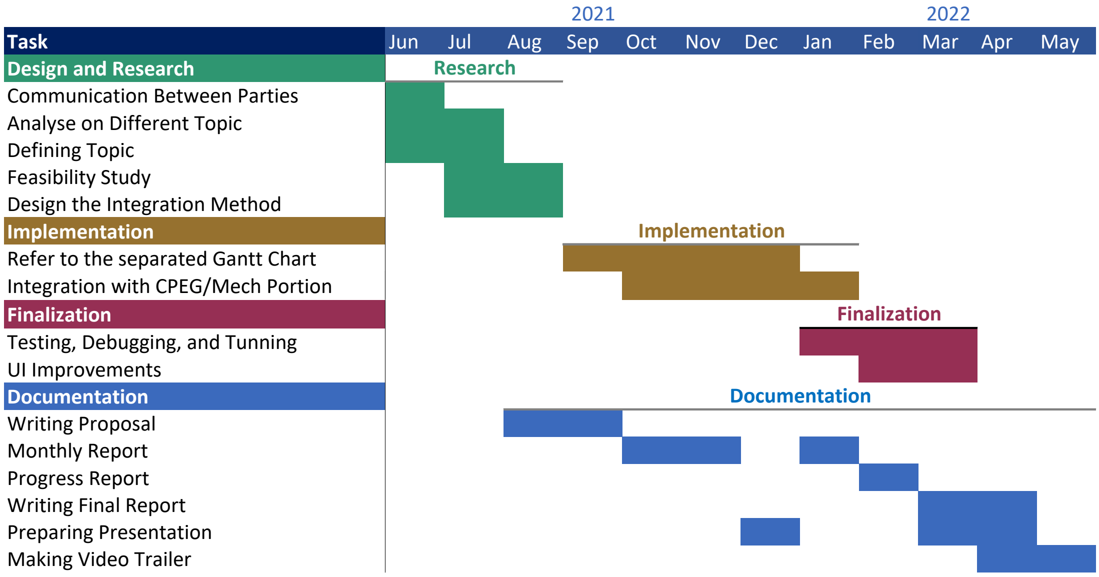


Table 15 Gannt Chart for Overall Progress

CPEG version

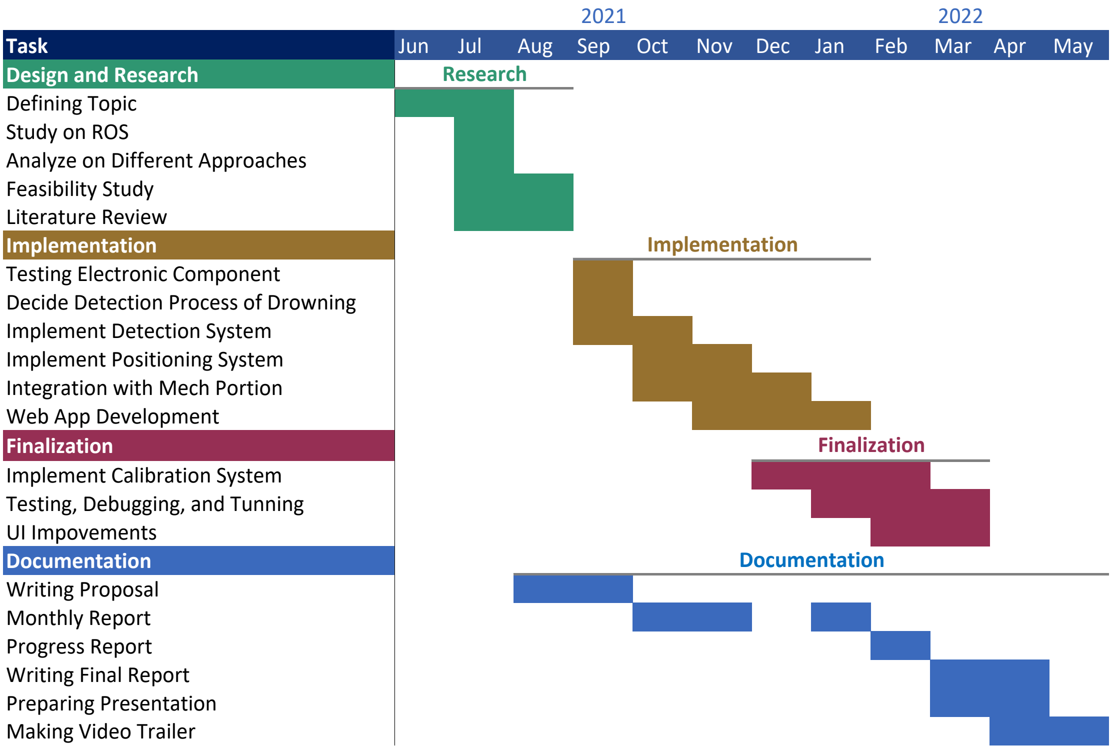


Table 16 Gannt Chart for the CPEG component of the project

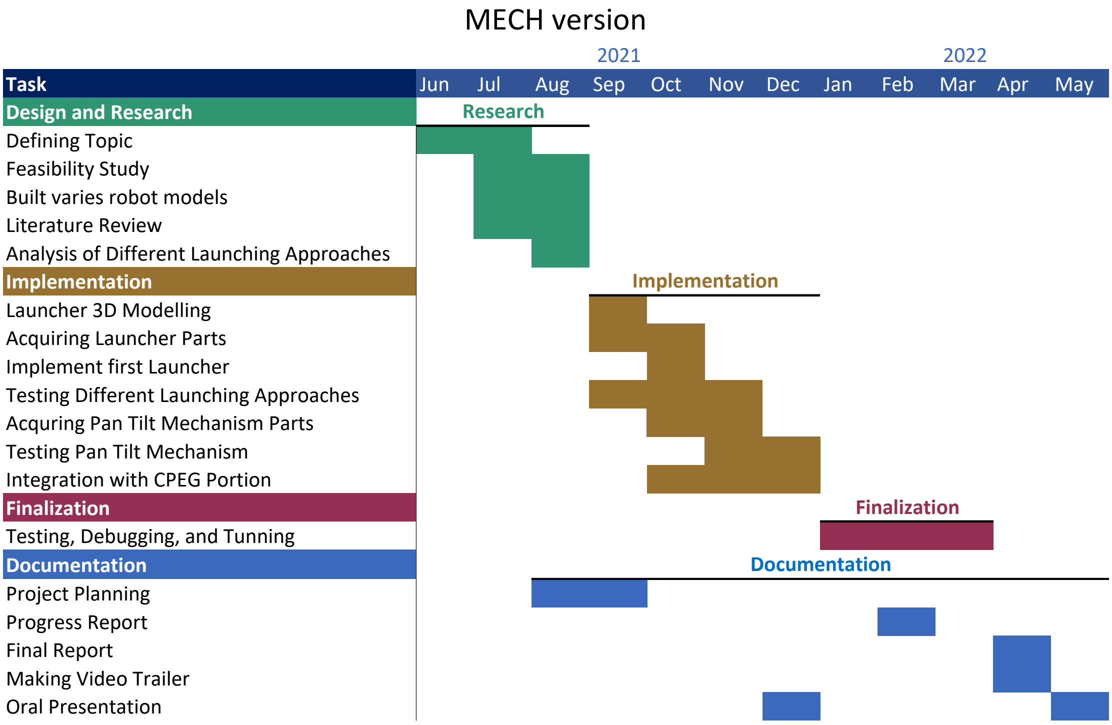


Table 17 Gannt Chart for the MECH component of the project

6. Required Hardware & Software

6.1 Hardware

Hardware	Usage	Price (HKD)	Quantity
Arduino Nano	Process data in the wearable	60	1
MAX30100	Monitor heartrate and blood oxygen level of user	19	1
Small Lithium ion Battery	Power the whole wearable device	80	1
LEDs	Positioning and alert nearby swimmers	7	4
Smart Phone	Webcam	1000	1
HKUST Server	Calculate position and host the webpage	Free	1

Table 18 List of details for the required hardware

6.2 Software

Software	Version	Usage	Price (HKD)
Arduino IDE	1.8.16	Program the Arduino	Free
Node.js	14.17.6 LTS	Server of Webpage	Free
DroidCam	6.15	Webcam Input	Free
Full Screen Web Browser App	2.1	Fullscreen web browser for iOS	Free
Python	3.10.0	Image processing algorithm	Free

Table 19 List of details for the required software

7. References

[1] “【泳灘遇溺】工會發聲明指救生員人手嚴重不足 憂泳灘遇溺事件陸續有來 ([Drowning at the Beach] The union issued a statement saying that lifeguards are seriously understaffed.It is worried more drowning incidents to be coming),” 香港經濟日報 HKET, 06 Jun 2021. [Online]. Available: https://topick.hket.com/article/2975671/%E3%80%90%E6%B3%B3%E7%81%98%E9%81%87%E6%BA%BA%E3%80%91%E5%B7%A5%E6%9C%83%E7%99%BC%E8%81%B2%E6%98%8E%E6%8C%87%E6%95%91%E7%94%9F%E5%93%A1%E4%BA%BA%E6%89%8B%E5%9A%B4%E9%87%8D%E4%B8%8D%E8%B6%B3%E3%80%80%E6%86%82%E6%B3%B. [Accessed 16 Sep 2021].

[2] “奪命泳池再揭救生員煲手機 (The life-threatening pool reveals that the lifeguard is phubbing.),” 東方日報 Oriental Daily News, 4 May 2010. [Online]. Available: http://orientaldaily.on.cc/cnt/news/20100504/00176_035.html. [Accessed 16 Sep 2021].

[3] “探射燈：救生員缺人康文署密冚 盲點睇漏遇溺難救(Spotlight: LCSD is troubled by the lack of lifeguard. It is difficult to rescue the victim in blind spots.),” 東方日報 Oriental Daily News, 10 Jun 2019. [Online]. Available: https://orientaldaily.on.cc/cnt/news/20190619/mobile/odn-20190619-0619_00176_091.html. [Accessed 16 Sep 2021].

[4] D. Szpilman and J. J. Bierens, “Drowning,” *New England Journal of Medicine,* vol. 366, no. 22, pp. 2102-2110, 2012.

[5] K. Ho, “Hong Kong's lifeguard shortage could seriously impact your summer plans,” SCMP, 16 Apr 2019. [Online]. Available: https://www.scmp.com/yp/discover/news/hong-kong/article/3062096/hong-kongs-lifeguard-shortage-could-seriously-impact. [Accessed 14 sep 2021].

[6] “No lifeguard at half of all public beaches,” The Standard, 25 Aug 2021. [Online]. Available: https://www.thestandard.com.hk/breaking-news-print/179370/No-lifeguard-at-half-of-all-public-beaches. [Accessed 14 9 2021].

[7] “HONG KONG DROWNING REPORT,” the Non-Communicable Disease Division, HKSAR, 2019.

[8] “Water Resistance On Watches Guide,” The Watch& Clock Shop, [Online]. Available: https://www.watcho.co.uk/watches/information/water-resistance-on-watches-guide.html. [Accessed 14 Sep 2021].

[9] M. R. MS, M. Ali, P. E. N, N. E. G, S. Ali and K. M.Y, “An early drowning detection system for internet of things (Iot) applications,” *TELKOMNIKA (Telecommunication Computing Electronics and Control),* vol. 16, no. 4, p. 1870, 2018.

[10] F. Dehbashi and N. Ahmed, “SwimTrack: Drowning Detection using RFID,” *Proceedings of the ACM SIGCOMM 2019 Conference Posters and Demos on - SIGCOMM Posters and Demos '19,* 2019.

[11] O. X. Qian, “Automated underwater lifeguard,” 2019. [Online]. Available: https://ongxiangqian.com/portfolio. [Accessed 14 Sep 2021].

[12] W. T. Fok, “Drowning detection with AI,” The University of Hong Kong - KE Newsletter, 2020. [Online]. Available: https://www.ke.hku.hk/story/drowning-detection-with-ai. [Accessed 14 Sep 2021].

[13] W. T. Fok, “HKU AI Drowning Detection System,” 15 Oct 2020. [Online]. Available: https://youtu.be/nFtpUgEQvvI. [Accessed 14 Sep 2021].

[14] “A drowning detection system for swimming pools,” SwimEye, 15 Nov 2020. [Online]. Available: https://swimeye.com/. [Accessed 14 Sep 2021].

[15] “What Does a Real Drowning Victim Look Like?,” FOX NASHVILLE, 29 7 2014. [Online]. Available: https://youtu.be/ErtQKtTTGxs. [Accessed 14 Sep 2021].

[16] L. Noggin, “What Really Happens To Your Body When You Drown?,” 9 Mon 2017. [Online]. Available: https://youtu.be/CMxHQ5B7dZ8. [Accessed 14 Sep 2021].

[17] S. Cheriyedath, “Photoplethysmography (PPG),” News Medical, 27 Feb 2019. [Online]. Available: https://www.news-medical.net/health/Photoplethysmography-(PPG).aspx. [Accessed 14 Sep 2021].

[18] “Tests to measure your oxygen levels,” British Lung Foundation, Jan 2020. [Online]. Available: https://www.blf.org.uk/support-for-you/breathing-tests/tests-measure-oxygen-levels. [Accessed 14 Sep 2021].

[19] “Air Safety Institute Safety Alert,” AOPA, [Online]. Available: https://download.aopa.org/asf/Air-Safety-Institute-Safety-Alert_Hypoxia.pdf. [Accessed 14 Sep 2021].

[20] L. J. Montenij, W. de Vries, L. Schwarte and J. J. Bierens, “Feasibility of pulse oximetry in the initial prehospital management of victims of drowning: A preliminary study,” *Resuscitation,* vol. 82, no. 9, pp. 1235-1238, 2011.

[21] M. Ghamari, “A review on wearable photoplethysmography sensors and their potential future applications in health care,” *International Journal of Biosensors & Bioelectronics,* vol. 4, no. 4, 2018.

[22] “ZOLL R Serires Pulse Oximetry (SpO2),” ZOLL, Jan 2018. [Online]. Available: https://www.zoll.com/-/media/public-site/products/r-series-defibrillators/9650-0901-01-sf_f.ashx. [Accessed 14 Sep 2021].

[23] L. Holbery-Morgan, J. Carew, C. Angel and N. Simpson, “Feasibility of pulse oximetry after water immersion,” *Resuscitation Plus,* vol. 7, p. 100147, 2021.

[24] “MAX30100 Pulse Oximeter and Heart-Rate Sensor IC for Wearable Health,” 24 Sep 2014. [Online]. Available: https://datasheets.maximintegrated.com/en/ds/MAX30100.pdf. [Accessed 2022 Feb 14].

[25] “Mykee Locker Key Holder,” [Online]. Available: http://www.mykee.com/english/html/emykee_main.html. [Accessed 14 Feb 2022].

[26] “Strap band for Xiaomi Mi Band 3/4/5/6,” AliExpress, [Online]. Available: https://tr.aliexpress.com/item/1005002460119063.html?pdp_ext_f=%7B%22ship_from%22:%22%22,%22sku_id%22:%2212000020731413412%22%7D&&scm=1007.37427.249190.0&scm_id=1007.37427.249190.0&scm-url=1007.37427.249190.0&pvid=f3b1da52-b527-4fff-bd4b-885895169986&utpa. [Accessed 13 2 2021].

[27] “Boys' swimming story lines, metro swimmers and divers to watch,” 11 Dec 2017. [Online]. Available: https://www.mnswimminghub.com/news_article/show/865090. [Accessed 14 Feb 2022].

[28] “Mosquito Repellent Wristband, Non-Toxic Travel Insect Repellent Bracelets Keeps Insects & Bugs Away - 6 Packs,” Walmart Canada, [Online]. Available: https://www.walmart.ca/en/ip/Mosquito-Repellent-Wristband-Non-Toxic-Travel-Insect-Repellent-Bracelets-Keeps-Insects-Bugs-Away-6-Packs/PRD5U0FG7ZO9CIQ. [Accessed 19 April 2022].

[29] “5pcs Silicone Wristband Bracelet Casual Bangles Wrist Band Sports Jogger Running Adjustable Jewelry Accessories Gift Wholesale,” Joom, [Online]. Available: https://www.joom.com/en/products/60d155ebd6f14b0111df7848. [Accessed 19 April 2022].

[30] “Knight Owl Teaching resources - Parts of the Hand Poster,” Knight Owl, [Online]. Available: http://knightowl.education/product/parts-of-the-hand-poster/. [Accessed 19 April 2022].

[31] “Hough Line Transform,” OpenCV, [Online]. Available: https://docs.opencv.org/3.4/d9/db0/tutorial_hough_lines.html. [Accessed 14 Feb 2022].

[32] "Geometric Image Transformations," OpenCV, [Online]. Available: https://docs.opencv.org/4.x/da/d54/group__imgproc__transform.html#ga20f62aa3235d869c9956436c870893ae. [Accessed 14 Feb 2022].

[33] "Template Matching," OpenCV, [Online]. Available: https://docs.opencv.org/4.x/d4/dc6/tutorial_py_template_matching.html. [Accessed 14 Feb 2022].

[34] EmmaAshely, "Flexible Printed Circuit Board, Introduction and importance.," DESIGNSPART, 7 Jul 2021. [Online]. Available: https://www.rs-online.com/designspark/flexible-printed-circuit-board-introduction-and-importance. [Accessed 19 April 2022].

[35] A. H. Marblestone and Z. B. M., "Physical principles for scalable neural recording," *Frontiers in Computational Neuroscience,* 2013.

[36] "How to ensure heart rate is accurate on Samsung Galaxy Watch Active?," Samsung, 12 Oct 2020. [Online]. Available: https://www.samsung.com/sg/support/mobile-devices/how-to-ensure-heart-rate-is-accurate-on-samsung-galaxy-watch-active/. [Accessed 14 Sep 2021].

[37] RichD, "Tennis Ball Thrower," 11 Feb 2019. [Online]. Available: https://grabcad.com/library/tennis-ball-thrower-1. [Accessed 14 Sep 2021].

[38] "Arduino Nano," Arduino.cc, [Online]. Available: https://store.arduino.cc/products/arduino-nano. [Accessed 14 Sep 2021].

[39] "small li-polymer battery 501015 3.7v 20mah 50mah lithium ion battery," Alibaba, [Online]. Available: https://www.alibaba.com/product-detail/small-li-polymer-battery-501015-3_62138465279.html. [Accessed 14 Sep 2021].

[40] "434MHz RF Transceiver Module (UART)-100m," Cytron, [Online]. Available: https://www.cytron.io/c-wireless-devices/c-rf-module/p-434mhz-rf-transceiver-module-uart-100m. [Accessed 14 Sep 2021].

[41] "Arduino MAX30100 | Pulse Oximeter Heart-Rate," Taiwan Intelligent Sensor Technology Co., Ltd., [Online]. Available: https://www.taiwansensor.com.tw/product/arduino-max30100-%E8%84%88%E6%90%8F%E8%A1%80%E6%B0%A7%E5%84%80-%E5%BF%83%E7%8E%87pulse-oximeter-heart-rate-%E6%84%9F%E6%B8%AC%E5%99%A8%E6%A8%A1%E7%B5%84/. [Accessed 14 Sep 2021].

[42] "NeoPixel LEDs," Compoents101, 3 Oct 2018. [Online]. Available: https://components101.com/displays/neopixel-led-strip. [Accessed 14 Sep 2021].

[43] “Dev47Apps,” Dev47Apps, [Online]. Available: https://www.dev47apps.com/. [Accessed 14 Sep 2021].

[44] K. Watson, “Drowning Facts and Safety Precautions,” Healthline Media, 24 Mar 2020. [Online]. Available: https://www.healthline.com/health/how-long-does-it-take-to-drown#drowning-stages. [Accessed 14 Sep 2021].

[45] “Double for loop performance | Speed Test | Java vs Javascript vs Python | Tech Tips,” Flabo Education, 17 4 2020. [Online]. Available: https://youtu.be/QrFq8d8ngzQ. [Accessed 14 9 2021].

[46] S. Izle, “Node.js advantages: What do you need to know before starting the project?,” ProCoders, 03 Aug 2021. [Online]. Available: https://procoders.tech/blog/advantages-of-using-node-js/. [Accessed 14 Sep 2021].

[47] H. D. “Best Frontend Frameworks of 2021 for Web Development,” SIMFORM, 5 Jan 2021. [Online]. Available: https://www.simform.com/blog/best-frontend-frameworks/. [Accessed 14 Sep 2021].

[48] “Optimizing Performance,” React, [Online]. Available: https://reactjs.org/docs/optimizing-performance.html. [Accessed 16 Sep 2021].

[49] “ISL will Donate bulkhead to city of Naples to divide pool into 25 Meter Course,” SwimSwam, 19 Sep 2019. [Online]. Available: https://swimswam.com/isl-will-donate-bulkhead-to-city-of-naples-to-divide-pool-into-25-meter-course/. [Accessed 15 Sep 2021].

[50] R. Pelayo, “How to use the Max30100 as Arduino heart rate sensor,” Microcontroller Tutorials, 23 Jul 2021. [Online]. Available: https://www.teachmemicro.com/max30100-arduino-heart-rate-sensor/#google_vignette. [Accessed 16 Sep 2021].

[51] “Harris corner detection,” OpenCV, [Online]. Available: https://docs.opencv.org/4.x/dc/d0d/tutorial_py_features_harris.html. [Accessed 14 Feb 2022].

8. Minutes

8.1 Minutes of 1st project Meeting

Date: 10 June 2021 (Thur)

Time: 17:30 – 19:00

Present: Dr. Desmond Tsoi, Loreen, Fionn, Jackal, Don

To-do:

- ☐ Research on SLAM, ROS, Printed PCB.
- ☐ Terrain the robot will operate on.
- ☐ Ask Mech Prof for advice on the terrain adaptability of the robot.
- ☐ Try all peripherals before summer break ends.

Discussion:

1. General stuff to be aware
    a. We have two reports to submit (MECH and CPEG)
    b. Think of how to test our robot
    c. Setup "complete stages" of the project, to ensure to Project is "complete"
    d. UI is a crucial element
    e. The project should Demo Oriented
2. Idea on robot
    a. The Robot should be overridable, as we cannot promise the bot to operate the way we want.
    b. Which Sensor we may use: Thermal Camera, Gyro, GPS, Mic, LTE module
    c. Form of Robot: Portable? 4-feet? Drone? Hybrid (4-feet + drone)?
    d. Control of Robot: autonomous/with a human assist/Human control with an electronic assist(advice)
    e. What to do when it encounters patient: set up a communication channel to outside, Bring patient back to the main route.
3. We set up 4 Temporary "complete stages" for our Project

8.2 Minutes of 2nd Project Meeting

Date: 1 July 2021 (Thur)

Time: 22:00 – 00:30

Present: Dr. Desmond Tsoi, Loreen, Fionn, Jackal, Don

To-do:

- ☐ Research on GPS accuracy
- ☐ Ask Robin for a sample of FYP

Discussion:

1. Focus: The patients who cannot call for help, mostly solo and without preparation
2. Idea:
   a. Keep Track of the people, send out when needed
   b. A wearable device, keep track of heart rate, etc
   c. Trigger from outside to obtain patient's GPS location (PLBS?)
   d. A drone that able to fly inside the forest
   e. Signal Approach: a drone that detects the common phone's communication frequency
3. Soft Robot This Unstoppable Robot Could Save Your Life
4. How to Find People who are not in range of signal
   a. 天災時代替基地台的無人機，可供方圓 10 公里手機同時使用 (A Drone that provides Mobile Phone Signal for phones within the range of 10KM)
   b. 中華電信與雷虎科技合作展出「空中基地台系統」(Chunghwa Telecom Unmanned Air Systems; CHTUAS)

Useful Links:

- ☐ MOBILE NETWORK COVERAGE OF HIKING TRAILS IN COUNTRY PARKS
- ☐ Use fall detection with Apple Watch
- ☐ Hong Kong Frequency Allocation Chart
- ☐ WHY DRONES ARE THE FUTURE OF SEARCH AND RESCUE

8.3 Minutes of 3rd Project Meeting

Date: 7 June 2021 (Wed)

Time: 22:00 – 00:30

Present: Dr. Desmond Tsoi, Kris, Loreen, Fionn, Jackal, Don

To-do:

1. Try out different Wheel
2. Try out ROS
3. Feasibility study on Drone
4. Assemble and try belt drive in different terrain /w ROS

Discussion:

1. Idea
    a. A drone saves people from heat stroke
        i. Scenario:
            1. Heatstroke blackspots usually do not have many trees that can block the sunlight, which is suitable for the drone to operate
            2. According to AFCD and hobbies website, there is a place surrounded by 3-4 blackspots within 10km, set up a charging station and keep petrol around those blackspots
        ii. Stage 1: use a thermal sensor to identify patients, take advantage of the drone's propeller to fan the patient, and use a mechanical part to spray water to help heat dissipation.
        iii. Use an AirTag-ish cheap BLE, distribute at the entrance of the hiking trail, if the Tag is not returned within 2-days, we send out the drone to search for the patient
            1. Advice: use the deposit to increase the return rate, add a stamp on the tag, the user can return my putting the tag post-box
    b. Land-Base robot
        i. Scenario: Following the rescue team, Petrol 24/7

ii. Usage:
    1. Rescuer uses a phone to order the bot to search for an area
    2. Petrol blackspot with schedule
    3. If it meets the patient, build a communication channel to the rescuer, block sunlight, spray water, measure body temperature, heart rate
iii. Appearance
    1. WALL-E like: https://youtu.be/ISznqY3kESI

c. Limitation in outdoor: signal usually weak as it will not bounce back

2. Kris gives us some detail of using InnoLab
3. Note on Project
    a. It is ok to change the project if it is justifiable
        i. e.g., What benefits that is better than original
    b. The report is a storytelling of the report: the journey of exploration, what option you encounter and why you did not choose them, compare their pros and cons

8.4 Minutes of 4th Project Meeting

Date: 31 June 2021 (Thur)

Time: 17:30 – 19:00

Present: Dr. Desmond Tsoi, Loreen, Fionn, Jackal, Don

To-do:

1. Ask Mech Prof for approval of the swimming pool robot
2. Research on how to detect drowning, and how to save a drowning person
3. Ready some questions for the swimming pool manager

Discussion:

1. Huge "human-eating" firefighting bot
   a. Hi-power Motor may be expensive, take advantage of gear ration
   b. Fire-proof material, we need to ensure the patient save inside the robot
   c. Can it substitute a firefighter? Maybe it can bring the patient out of the area, to free the firefighter to keep saving people.
   d. Scale is huge and difficulty is high, we might not have enough time, 1:10 scale demo?
2. Swimming Pool Robot:
   a. Privacy problem (If we use computer vision approach), CCTV does exist in public swimming, but it is confidential, we may use a low-resolution camera/Infrared, but does infrared work with water?
   b. Wearable Device approach, measure position, blood oxygen, and heart rate to detect drowning
   c. Underwater robot approach, it is easier to reach the patient. Corrosion problem? Does not necessary to solve
   d. Key Points: Is it fast enough? (30m to reach) Can it save a person?
3. Surveillance heat stroke
   a. We can have multi bot petrol together
4. If we encounter a situation where we do not have enough budget, we need to justify the user scenario

8.5 Minutes of 5th Project Meeting

Date: 5 Aug 2021 (Thur)

Time: 17:30 – 19:00

Present: Prof. Ma, Loreen, Fionn, Jackal, Don

To-do:

1. Gantt Chart
2. CS: Wearable Device, detect drawn, Locating
3. MECH: Release Mechanism
4. Ask Robin to get an FYP sample

Discussion:

1. Shooting approach is preferred than Spidercam approach, but we need to consider the safety issue as it might hit other swimmers.
2. Wrist band is feasible solution.
3. We need to consider the effect of water and movement toward the heart rate sensor.

8.6 Minutes of 6th Project Meeting

Date: 19 Aug 2021 (Thur)

Time: 17:30 – 19:00

Present: Dr. Desmond Tsoi, Loreen, Fionn, Jackal, Don

To-do:

1. Research and settle a detection method.
2. Draft report before/on 1/9, prove read till last mins.
3. Buy electronic parts and test.
4. Explore more papers about drowning detection, compare their feasibility, cost, and accuracy.
5. Ask Prof. Tim Woo for the question about the control system.

Discussion:

I. Resolution with Robin: 1. wrist band is recommended 2. A NO for spider cam approach, use shooting approach.

II. Backup System: We may give the decision part to the lifeguard to avoid error, Make it configurable for shooting

III. Calibration System: camera angle? Wind direction? we need a system for calibration to adapt to different swimming pools.

IV. IoT communication method: visible Light, Radio wave, we should test one of it first.

V. Desmond gives the detail of writing the FYP report and Presentation

8.7 Minutes of 7th Project Meeting

Date: 25 Aug 2021 (Thur)

Time: 12:00 – 15:00

Present: Loreen, Fionn, Jackal, Don

To-do:

I. Buy heartbeat and Blood oxygen sensor.

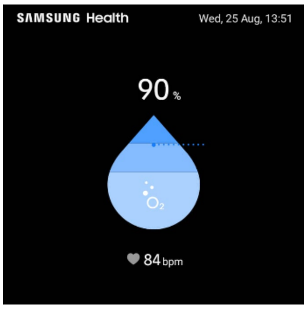
II. Try the exiting image approach on GitHub (Drown detection).

III. Begin in writing the proposal.

IV. Think of how our system communicates with the robot.

V. Ask for advice from Desmond to decide should we buy a commercial watch as a control data.

Discussion:



1. Tennis ball shooting approach and Air cannon shooting approach.
2. We can measure the drop of blood oxygen when we hold our breath, so our approach is promising.
3. We may make a complete seal to avoid water's effect on the blood Oxygen sensors. And use a watch clasp mechanism to ensure it fits.
4. We draft the flowchart and Gantt chart of our project.

8.8 Minutes of 8th Project Meeting

Date: 26 Aug 2021 (Thur)

Time: 11:00 – 01:00

Present: Dr. Desmond Tsoi, Loreen, Fionn, Jackal, Don

To-do:

1. Read the Indoor Light positioning paper.
2. Buy max30100.
3. Read FYP sample “Vision and Graphic”
4. Ask Robin for the progress report and if Mech need a proposal

Discussion:

1. Desmond gives us detailed advice on writing the proposal. (From structure, to topic, so, literially, a lot)
2. Calibration System: Make a noddle that is equipped with our positioning system that shape and weight similar to our kickboard, so it can measure position, making the calibration process easier.

8.9 Minutes of 9th Project Meeting

Date: 2 Sep 2021 (Thur)

Time: 22:30 – 00:00

Present: Dr. Desmond Tsoi, Loreen, Fionn, Jackal, Don

To-do:

1. Read the FYP sample “Vision and Graphic”
2. Ask if Prof. Robin have any Mech progress report sample, and if mech need a proposal report

Discussion:

1. The detail, requirement and advice in writing a proposal

8.10 Minutes of 10th Project Meeting

Date: 4 Oct 2021 (Tue)

Time: 20:00 – 22:30

Present: Dr. Desmond Tsoi, Loreen, Fionn, Jackal, Don

To-do:

1. Email to communication tutor to arrange a meeting for individual essay
2. Read sample of individual essay
3. Submit the 1st intermediate report
4. Inform mech FYP group of about the new FYP member

Discussion:

1. Update progress of both CPEG and MECH
   i. Request VM server from FYP
   ii. Tried setup the server using raspberry pi
      1. Advice: separate workload of server (run on client)
2. Distributed the task to do on the project.
   i. CPEG
      1. Take camera data, communicate with robot, UI, config interface (Don)
      2. Drowing detection (Loreen)
      3. Location system, image processing (TBC)
   ii. MECH
      1. Air compressing machine, Tilting
3. Arranged the meeting of mech and cpeg.

8.11 Minutes of 11th Project Meeting

Date: 7 Oct 2021 (Wed)

Time: 20:00 – 22:00

Present: Dr. Desmond Tsoi, Don, Edwood, Fionn, Jackal, Loreen

To-do:

1. Don: web app, design the appreance of the website
2. Edwood: try template matching, homography transformation and radio distortion.
3. Loreen: Testing on MAX30100
4. Mech: ongoing test on valve

Discussion:

i. Update progress

1. Borrowed VM server
2. Corner Detection
3. Test the Arduino, LED and buzzer
4. Brough rubber band and kickboard

8.12 Minutes of 12th Project Meeting

Date: 27 Oct 2021 (Wed)

Time: 21:30 – 23:00

Present: Dr. Desmond Tsoi, Don, Edwood, Fionn, Jackal, Loreen

To-do:

1. Setup VM server, and make a basic control interface for manipulating the robot
2. waterproof & program (data storage) [buy 2 cheap buzzers]
3. corner detection [harris corner detection]

Discussion:

1. Report Progress
    a. Web interface design by figma
    b. Planning of waterproof setting
    c. Leaning OpenCV

8.13 Minutes of 13th Project Meeting

Date: 3 Nov 2021 (Fir)

Time: 20:30 – 22:30

Present: Dr. Desmond Tsoi, Don, Edwood, Fionn, Jackal, Loreen

To-do:

Update progress:

1. Loreen:
   i. trying to verify data with a real SpO2 meter
   ii. export MAX30100 data record in spreadsheet for analysis
2. Don:
   i. made a simple demo for communication with robot by web app
3. Edwood:
   i. trail on corner detection (Harris corner detection) w/ blur, RGB channel etc

8.14 Minutes of 14th Project Meeting

Date: 17 Nov 2021 (Wed)

Time: 20:30 – 21:30

Present: Dr. Desmond Tsoi, Don, Edwood, Fionn, Jackal, Loreen

To-do:

- [ ] Waterproof design
- [ ] Purchase a waterproof speaker
- [ ] Filter out blue screen and research on edge approach
- [ ] MECH: find new parts from the trail

Update progres:

1. Bought a commercial SpO2 tseting for control testing
2. MECH: 1st trail on 3D printing, and find it's limitation

8.15 Minutes of 15th Project Meeting

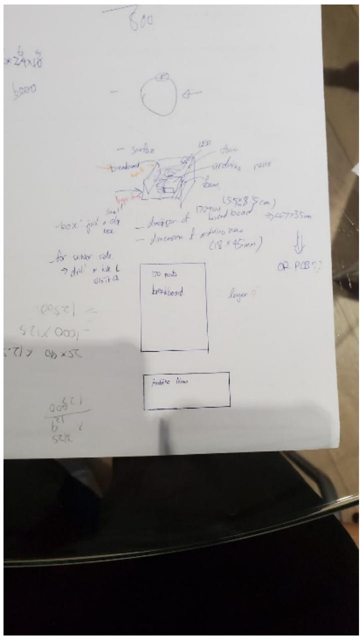
Date: 1 Dec 2021 (Wed)

Time: 20:00 – 22:00

Present: Dr. Desmond Tsoi, Don, Edwood, Fionn, Jackal, Loreen

To-do:

1. UI, stream the video input to client, origin setting (config) of the robot
2. Start assemble the waterproof box
3. Buy the water-proof speaker, arduino nano and 170point breadboard
4. Extract the edge and corner



Update progress:

1. Projectile motion algorithm
2. Made the whole working mechanic from input(touching the screen) to sending command to robot
3. Waterproof planning
4. More study on corner/edge detection solution
5. MECH: ref to 18_meeting

Discussion:

We will need more time, and will avaliable after 10/12 (due to exam)

8.16 Minutes of 16th Project Meeting

Date: 21 Jan 2022 (Fir)

Time: 14:00 – 16:00

Present: Dr. Desmond Tsoi, Don, Edwood, Fionn, Jackal, Loreen

To-do:

1. Test led refraction (Loreen)
2. build small sample (Loreen)
3. test different components (Loreen)
4. Create kickboard animation (Don)
5. Fix zoom in/out bug (Don)
6. Check latency (Edwood)
7. Zoom template (Edwood)
8. Test if object does not appear (Edwood)

Update progress:

1. Done 4 corner setting page
2. Done size and robot setting page
3. Done alert time setting page
4. Calculated refraction of the led light -> don't have to worry total internal refraction
5. Done multiscale template matching

Discussion:

issue about LED detection:

1) Refraction index too big -> difficult to detect the LED

Suggestion:

2) Underwater camera?
3) Change to when hear "help", shoot buoy -> no because can't locate that person

8.17 Minutes of 17th Project Meeting

Date: 10 Feb 2022 (Thur)

Time: 20:00 – 23:00

Present: Dr. Desmond Tsoi, Don, Edwood, Fionn, Jackal, Loreen

To-do:

- [ ] Change the old stuff (fix figure problem)
- [ ] Read fyp report (visually impair and guita)
- [ ] add stick man (use case diagram) , may ref zinc's final report page 20. usually start with a verb
- [ ] Finish draft2 on 12

Discussion:

We discuss the detail about progress report.

Advice:
1. Make a progress table, may put at the end (after explaination)
2. Consistent terminology
3. Address sec reader's comment
-Detailed division of labor.
-Individual sub-system testing is mentioned.
-Figures cannot express the clear purpose.
4. No need literature review, we can put it in appendix to play safe.
5. Add stick man (use case diagram)

Question to ask:
- [ ] Do we need the change the old stuff?

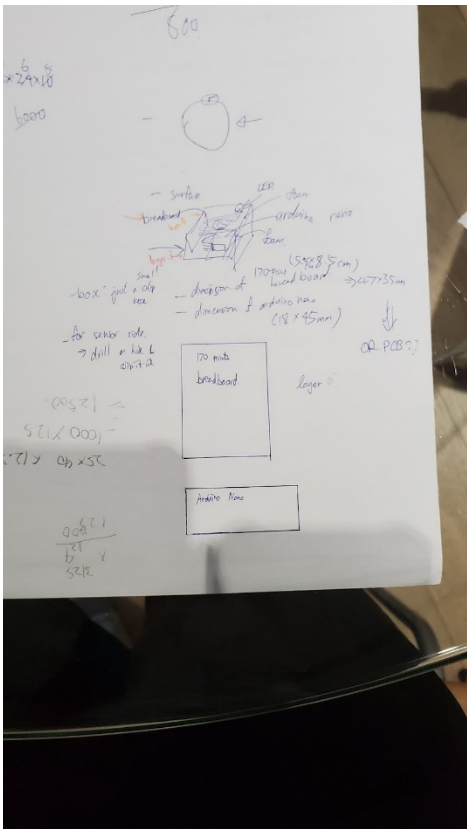
change the old stuff (address figures clearness)

- [ ] What is user case diag

How do different users interact with the system
it is a method to analyze the user requirement
We have two users: the client and the lifeguard, and we can also mention the mech part
note: Edwood have studied it b4

8.18 Minutes of 18th Project Meeting

Minutes
Date: 1 Dec, 8:00PM
2021
Attendance:
Desmond, Don, Edwood, Fionn, Jackal, Loreen



AGENDA

1. Progress Update for CPEG
2. Future Work for CPEG
3. Other issues for CPEG
4. Progress Update for MECH
5. Future update for MECH
6. Other issues for MECH
7. Gathering of Examination timetable
8. Upcoming planning for the winter semester
    1. Progress Update for CPEG
    2. Future Work for CPEG

Don:

1. Projectile motion algo,
2. connected from input to XML output

Future:

1. UI,
2. steam video input to client,
3. Origin setting of the robot

Loreen:

1. waterproof design planning.

Future:

1. test the water-proof method (expose the sensor only)
2. start to assemble the waterproof box
3. Buy the water-proof speaker, Arduino Nano, 170 point breadboard

Edward

1. more study on corner/edge detection solution

Future:

1. extracting the edge and the corner
2. need more time, available 10/12 (due to exam)

    1. Other issues for CPEG

- ☐ Winter Semester Room booking arrangements
- ☐ Holiday on PH, like Christmas, new year eve, lunar new year

Come: (Don), Edwood, Fionn, Jackal, Loreen

Don is not always available due to PC limitations.

1. Progress Update for MECH
2. Future update for MECH

Progress Update

1. Progress Report submission

Fionn:

1. Developed a new approach of shooting mechanism: 3D printing rail method(Elastic Band / Spring)
2. 1st version of the solidworks done

Loreen:

1. Angle Tuning Item: 1st version of the solidworks done

Jackal:

1. Update on the Taobao delivery status:

   stock has already reached the delivery store, so should arrive pretty soon

Future:

1. Presentation
2. Finalise and present PowerPoint Demo at 530pm tomorrow
3. Real presentation on Friday Morning

Fionn:

1. Finalise design on the v-slot kart motor control system (esp. the rubber band one)
2. Try to build the basic model for the V-slot shooting

Loreen:

1. finalise design for the angle tuning system on Solidworks
2. Try to build the basic model of the rotational table

Jackal:

1. purchasing items from Taobao, presentation preparation

Taobao:

Done:

1. bought the cart running on the rail, currently shopping
2. bought the bearing, solenoid, currently shipping

To Do item:

1. Buy the 1m v-slot (alum rails), screw jack,

Target： get all material before the next meeting after the exam

1. Other issues for MECH

    Seek Desmond help for being the 2nd reader for MECH Final Report

2. Gathering of Examination timetable

    Exam：

    Desmond: 8,16
    Don: 10,17,18
    Edwood: 8, 10,
    Fionn: 7,8, 16
    Jackal 11, 18
    Loreen 7,8,18

    1. Upcoming meeting:

    2/12 6:00 PM for the practice of presentation

8.19 Minutes of 19th Project Meeting

Date: 21/1/2022 14:00 – 16:00

CPEG:

Loreen:

issue about LED detection:

- ☐ Refraction index too big -> difficult to detect the LED

Suggestion:

- ☐ Underwater camera?
- ☐ Change to when hear "help", shoot buoy -> no because can't locate that person

Will do:

- ☐ Test led refraction
- ☐ build small sample
- ☐ test different components

Don:

- ☐ Done 4 corner setting page
- ☐ Done size and robot setting page
- ☐ Done alert time setting page
- ☐ Calculated refraction of the red light -> don't have to worry total internal refraction

Will do:

- ☐ Create kickboard animation
- ☐ Fix zoom in/out bug

Edwood:

- ☐ Done multiscale template matching

Will do:

- ☐ Check latency
- ☐ Zoom template
- ☐ Test if object does not appear

Prof. Desmond:

Issue:

- ☐ Camera is covered

Suggestions:

- ☐ More camera?

Justification of our system:

- ☐ Help improve but not replace the lifeguard
- ☐ Talk about the limitation
- ☐ Have to look for paper to support our approach

Progress report:

- ☐ list out what have done clearly and detailed
- ☐ list out what have done including those which are explored and discarded and explain why they are discarded
- ☐ start doing on 29 or 30 and then after new year

Mech:

Issue:

- ☐ can't do accurate stepping

Will do:

- ☐ laser cut the gear
- ☐ buy big support bearing
- ☐ buy wood plank
- ☐ set jaral
- ☐ stepping program (change library?)
- ☐ create release mechanism

6.17 Minutes of 17th Project Meeting

Minutes
10/2/22 8 pm, meeting with desmond (CPEG)

Advice:
1. make a progress table, may put at the end (after explaination)
2. consistent terminology
3. address sec reader's comment
-Detailed division of labor.
-Individual sub-system testing is mentioned.
-Figures cannot express the clear purpose. --charlesz
-He likes the impact and business style stuff
4.no need literature review, we can put it in appendix to play safe.
5. add stick man (use case diagram) to address Charles Zhang's need lol

Question to ask:
-Do we need the change the old stuff?
change the old stuff (address figures clearness)

-What is user case diag
How do different users interact with the system
it is a method to analyze the user requirement
We have two users: the client and the lifeguard, and we can also mention the mech part
node: Edwood have studied it b4

Todo:
-change the old stuff (fix figure problem)
-Read fyp report (visually impair and guita)
-add stick man (use case diagram) , may ref zinc's final report page 20. usually start with a verb
- draft2 done on 12

6.21 Minutes of 21st Project Meeting

Minutes
23/2/22 21:00, Desmond, Don, Edwood, Fionn, Jackal, Loreen

problem: because of covid can't go to the lab!
it is possible to postpone the deadline
Loreen: if Loreen wants to go to the lab, there is nothing that can stop her.
UST: We are going to lock all rooms, except the one approved.

About president cup:
-submit a final report + poster
they will pick some groups, and make a video presentation for 10-15mins
then pick one for live presentation
Loreen: maybe we can write something will implement in final report

they focus on the practical, and the fusion of hardware and software

Desmond:
-it is a good chance to push our progress
-how to present the value of our product
-I gonna give u a sample of the final report of the president club report
-no pressure to try or not try the president club report
-reuse the materials (slides, final report)
-the people who join the president club is few

meeting tmr night 24/2, Thur, 9 pm, decide president club or not.
-tell desmond we join or not!

6.22 Minutes of 22nd Project Meeting

Minutes
23/2/22 21:00, Don, Edwood, Jackal, Loreen
Minutes
Todo：sat2pm，大圍 交收 Arduino+線

MECH:
Updates:
1) Rotational stand: let's buy one hehe, we got more budget
Loreen: the one we are looking for can support the weight, but not sure the area (i.e. balance)
田 shape, the balance should be OK

2)We have already tested on a shooting
We haven't measured if the angle is correct or not, but it is mathematically correct
Anyway, it can shoot at specific params repetitively
3)We might burn a MEGA board, hopefully not QAQ
4) Release mechanism is not yet confirmed, will update with Fionn

What to do next week:
1) Will solve the problem of MEGA or RAMP1.6 (the one we have burnt)
2) Will make the large kickboard stand
Kickboard stand: will scale it up, will use PC plastic and steel rod to strengthen it.
3) Shooting test: on the highest and lowest angle. gather the information of Speed

CEPG:
Edwood's update
need some time to catch up
Target:
1) Will dev more on template matching, e.g get the location in terms of the 3m*2m pool

Question:
1) What are we going to use? Ai or template matching

Don's update:
none since president cup
Target:
1)integrating image processing system
2)config process

Loreen's update:
-gathered information on waterproof,
target:
-will plot a LED for testing next week
-will ensure the Arduino work before plotting
-will glue it into a swimming band
-If nano is not usable, we will use uno for demo
-haven't tried it yet,

Loreen's question
Why the photo after the perspective transform is so long?
Edwood: Currently the process does not match the ratio, will implement
concern1: the photo might a bit off(走樣) after rescaling
concern2: resolved underwater LED cannot detect by LED, sample underwater can detect it

Is it possible to use AI?
Edwood: Yes, given that there are enough data. By YOLO
The difficult part is the have to label it and get samples.
How many samples do we need?
Edwood: Hard to estimate
SchoolAssig: 2k, intern: 20k for recongnize 200 diff objs
Require a different camera angle, different location, rotation, depth, etc

6.23 Minutes of 23th Project Meeting

Minutes
16/3/22 21:00, Desmond, Don, Edwood, Fionn, Jackal, Loreen

Progress:
MECH:
1. Loreen went to ssp and brought the MEGA board (since the board seem broken)
2. Rotation base, stoled a rotational base from old FYP project, lmao.
3. alt option on rotation base, buy it. (~1k), concern about supporting and balance
4. Place the release mechanism latch on the shooting rail, so that it will not bother when changing angle.

Future:
1. Loreen will test the MEGA board tmr
2. Will test shooting within these 2 weeks.
3. Fionn is making the release mech, will be able to test it soon
4. A 1:1 scale kickboard holder in progress, which is concept approved by TAs.

CS/CPEG:
Edwood
1. improved corner detection
2. tested on LED detection on different depth

Loreen
1. waterproofed a LED for testing

Future:
1. solder it onto the Arduino and plot it

Don
1. Tested Edwood's code on VM
2. Config logic
3. a bit of help on Arduino

Future:
1. Combine corner detection algorithm

Note on the new comment from charlesz

1. Final report put back the lit review (elaborate more on it, discuss the related work, and make the *objective* stronger)
TODO: add a discussion section on the literature review, which address and reinforce our 5 objectives. (e.g. time-critical, difficulty in identifying the victim (i.e., warable))

2. Add a subsection on eval.
a. research questions=> i don't really get this.
b. evaluation methodology (dataset, measure)

e.g. what criteria, eval all parts individually and overall.
UAT to 2 groups of people: lifeguard, swimmer by questionary.

3. Get an overall Gantt Chart
Our Gantt chart is separated into CPEG and MECH.
TODO: Make an Overall version. (And update CPEG to CS/CPEG)

Note on the final report and presentation:

1. If we got president cup, we could write it on the page after the cover page.

1. Advice: Summerize all 範疇 that we need. (Image processing, web programming, electronic circuit, mech,)

Note on oral presentation

1. it will be conducted on zoom
2. 40mins oral presentation:
   32mins for {present,demo}, 8 mins for questions.
   usually, high-level question, even if the question is harsh, answer them professionally :D

Note on the final video

1. 3 mins only, fast-forward the boring parts.

6.24 Minutes of 24th Project Meeting

Minutes
24/3/22 20:00, Desmond, Don, Edwood, Fionn, Jackal, Loreen

Todo: rec a vid for Edwood, 0.5s red green shift;

Desmond:
Presentation (we likely be the second sat 10:00-night)
Time of MECH report deadline (afaik Loreen: May?)
Summerize all fields that we need. (Image processing, web programming, electronic circuit, mech)

Progress:
Loreen:
1. testing on plotting
2. bought nano and Uno board.
Todo:
1. do a water test
2. Merge LED and Oximeter
3. Buy soldering
4. Buy a smartwatch 錶帶, glue them tgt

Edwood:
1. improve mem space usage
2. bug fixed
3. Red-Green LED detection
TODO:
1. detect blinking light (require a blinking led)

Don:
-Adapted to corner detection
-UI update
Todo:
1. manage alerts
2. update click function
3. update click-rect function

MECH:
TODO:
1. 搵人度窿
2. Forwarded email from Jackal （about claiming），need a new form （Loreen will fill it up tonight）
3. Present vs Teaching time crash: Ask early, email Desmond tonight!

Fionn:
低盤大螺絲（怕齒輪之間 contact surface 太少）
1. Confirm the gear is shorter (原裝)
2. release mechanism has done printing, accuracy is weird.

Question:
啲窿有無計 tolerence
Jackel：result 會比想像中大
結論：無視 tolerence？

6.25 Minutes of 25th Project Meeting

Minutes
25/3/22 11:00, Noor, Don, Edwood, Loreen

Previous problem:
- messy in the way of construction
- got good potential

Solution:
- think in terms of subsystem
- Before writing: draw out the content of the work

Context: a certain size pool, (aka real stuff), Specific the distance of the subsystem in real space (setting the scene for the reader)
simulation of the scenario, i.e a pool (then we put our toy in that simulation)

Introduction (?)
- Perfect scenario (flow illustration):
- Drawing out the floor plan for the possible scene to give the reader an idea of how it will do in actual scenario
- bring the point to simulation

1st: [wearable] detect the drown, wearable determine to alert the point of drowning,
1.5th: alert the camera
2nd: [localisation] localise the person with a camera, alert activate the camera
3rd: Calculate the location (image processing, computer vision offers lo)
4th: Activation of the shooting system
Technology behind IoT -> Image processing IoT of the system

Inner story of the subsystem:
Preprocessing of data to get to the data analysis & {Drowning Algorithm}
Measurement intake to Arduino Nano & Data analysis camera is going to localise
ALOG & ALOG (????? Seems Edwood know) Image
Phase 1: start on finding a pool {searching for a pool}
Phase 2: starting to find the [searching, and get the pixel location]
Phase 3: Find the exact point [in the real life, by m and m]
Happens after the location
Phase 4: IoT [Shooting]
Technology: Data analysis (sensory x1, Computer vision, Perspective transform analytics, conner detection for localisation, exact location), IoT (connection from software system to hardware, Wearable by Arduino | Shooting robot by Arduino & server)

4 Data Analysis:
wearable (detect the drown symptom)),
corner detection (set up the pool),
find the people (in pixel),

find the exact coordinate (in m)

Writing Flow
Component for Explaining Data analysis:
1) Set of Data input {example dataset} ->
2) equation used, show the math
3) expected output
4) function for it

IoT hardware:
wearable & robot

The connection between the IoT and server/ software

Experiment & testing & Discussion: from the experiment, we find these….
How did we conduct the testing, 5W(?)
Further Progress:
- Work the ppt out first
- Diagram out the algorithm and maths

Note mark by don:

Our main Contribution
1. When is drowning
2. Where the person is
3. Shoot the kickboard to there

Flow
(1). set the scene
(2). the technology behind (4algo)
(3). Prove (testing and discussion)
(4). conclusion (we can save a life)

Advice:
Do a PowerPoint presentation ourselves (lets go president cup hehe)
Then do the video.
Find our way from that.

Forget the introduction, we should go straight into the project. (?)

we are writing documentation
data mine our report (diagram it out)
custom our report based on how we explain it

6.26 Minutes of 26th Project Meeting

Minutes
3/4/22 2100-2300, Desmond, Don, Edwood, Fionn, Jackal, Loreen

Progress report

Mech:
—big 浮板托 assembling
—Motor test（checked no problem）
—testing mega board （debugging）
Fionn: Got bearing data, will draw tonight and laser cut it later.
Aim:
—solve the motor problem

CPEG：

Loreen
—waterproof is good
—it is so bright that it illuminates the edge region
Will do:
-will keep doing testing
-aim: waterproof wearable before the president cup

Edwood:
- Red Green shift
- Finish template matching already

President cup：
give a storyline
immersive
understand what are you doing! (some groups cant l)
layman
-technology (all in graph)
innovativeness
uniqueness
importance

15mins
plan the demo, 1mins for 1 slide (7-8 slides) , 花巧, for laymen, aim finish in 13 mins
will release the result on the day
T-shirt, smart casual

6.27 Minutes of 27th Project Meeting

Minutes
8/4/22 2200-0100, Desmond, Don, Edwood, Fionn, Jackal, Loreen

11/4 start writing the final report
15/4 1st draft

Tackle the comments of sec reader,
1) Testing section
2) add lit review

TODO：
-borrow batteries

Progress
Loreen:
1. battery not working problem, current not enough
Solu: use a Li-ion battery
2. frequency is weird
3. nano working voltage 6-12V, will supply with 2 18650
4. waterproof -, last
5. blood oxygen detection will plot except for the battery

Future development: how to solve the battery charging problem, use wireless charging?

Don
discussion,
suggestion, how to set up the camera

Mech:
1) tmr try to shoot a big kickboard
2) tmr start the base, to do integration
3) what is the width of the small gear to push the big gear, big 600mm gear ratio is 1:8, it should follow the gear ratio, and the small gear should be 96mm.
    tip diameter
    will punch through the wood plant, and place the motor under it
4) Need to decide how to spin the bottom plate, Loreen prefers to buy one (they are 400mm)
5) Jackal, should we focus on the bottom plate rotation