Chapter 1 Abstract

The report is based on the work that was explored during the 6 month's internship at Maya global. 3 different projects were assigned including the smart golf ball, text detection for water analogue meters and soil data extraction. The report will explain the methodologies taken, the component's and software application's used and the results achieved.

Chapter 2 Introduction

Maya global has led the way in transforming golf course administration using cutting-edge invention and technology. The business created Maya, a ground-breaking platform for golf course management and illness prediction, initially launched through a university thesis on water management and resilience in golf courses. The company achieved illness prediction and effective implementation across many pesticide-free golf courses in Belgium [1].

The 3 research and development projects I was involved with (the smart golf ball, text detection for water analogue meters and soil data extraction) will be discussed thoroughly through out the rest of the report.

2.1 Smart golf ball

The aim of this project is to develop a better alternative to test if the field is suitable for golf sporting events. The important measurements include:

1. How far the ball will travel across the field after getting off the ramp as a function of it's speed (usually the test involves rolling 3 balls of the ramp and calculating the average distance rolled after leaving the ramp).

2. The lateral and vertical deviation of the golf ball when rolling on the field based on the STRI metric.

3. Firmness of the field (The most common tool for measurement is the Clegg impact tester)

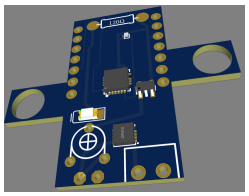
All these variables can be calculated with the main component the MPU6050 accelerometer through the Adafruit ESP32-C3 QT-PY which was coded on Arduino IDE. Fig. 2.1 shows the components needed for the ball.

The PCB in Fig. 2.1a includes:

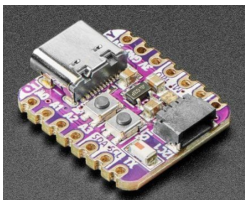
1. An MCP73833 (Battery Charger controller).
2. An MPU6050 (accelerometer).
3. An MMC5603 (Magnetometer). To stop the Z-axis acceleration from rising due to gravitational acceleration.
4. A LM367 (Voltage regulator), to convert 3.7 V to 3.3 V.
5. A Variable resistor, to manually turn off the PCB when unused.
6. A 120-ohm Resistor because the LED requires a lower voltage.

7. An LED light to show the PCB is ON or OFF.
8. A Battery Pad for the 3.7V lithium Battery.
9. A power source Pad for charging for the receiver of the wireless Charger.

The PCB Trace wiring were made in accordance to the data sheets [2] and [3].

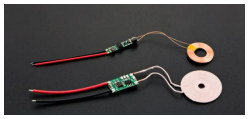


(a) The PCB



(b) Adafruit ESP32-C3 QT-PY [4]

(c) lithium battery 3.7 V [5]



(d) wireless receiver coil and module [6]

(e) 4 x (M3 x 3mm Philips pan Head Screws DIN 7985H A2 Stainless Steel) [7]

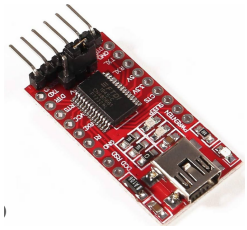
(f) 4 x (Brass Heat inserts for plastic) [8]

Figure 2.1: components for the smart golf ball

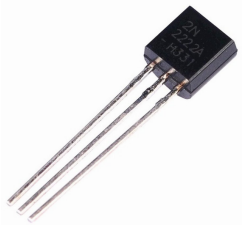
2.2 Text detection for analogue water consumption meter

Analogue water consumption meters measure the amount of water consumed on a regular basis; unfortunately, analogue water meter have to be observed manually in order to record the amount of water consumed. The aim of this project is to install a camera that will read the number presented on the water consumption meter once every minute and will update our database of the water consumed. The main coding packages that will be used are OpenCV, YOLO V7 and PaddleOCR. The required components to complete this project are presented in Fig. 2.2.

(a) ESP32 CAM



(b) HiLetgo FT232RL Mini USB to TTL Serial Converter Adapter Module 3.3V/5.5V [10]



(c) NPN transistor [11]

(d) 10 ohm resistor and 600 ohm resistor [12]

(e) Cool White LED Ring Light PCB - 70mm Diameter - 5V Power [13]

(f) 2 x (Brass Heat inserts for plastic) [8]

(g) 2 x (M3 x 3mm Philips pan Head Screws DIN 7985H A2 Stainless Steel) [7]

Figure 2.2: components for the water meter detection project

2.3 Soil data extraction from PDF

YOLO V7 provided impressive results in detection for the Text detection for analogue water consumption meter. So the team wanted to implement YOLO V7 to extract data from PDF files rather than enter them manually in SQL. This outlined a whole new project, where the consumer can extract the values they want from PDFs, use EasyOCR to read the values, convert the results into a Json format send the Json to the API where it places the results into different SQL databases. This project did not include any physical components, so it was much more convenient to accomplish.The software and packages used for this project were:

1. Ubuntu 20.04
2. YOLO V7
3. EasyOCR
4. OpenCV
5. Apache NiFi
6. PHP Storm
7. Data GRIP (SQL database)
8. Docker

The codes written were Python on VS code for YOLO V7 detection and EasyOCR

Detection and PHP on PHP Storm for API development.

Chapter 3 Main Work

3.1 Smart golf ball

3.1.1 RPS method

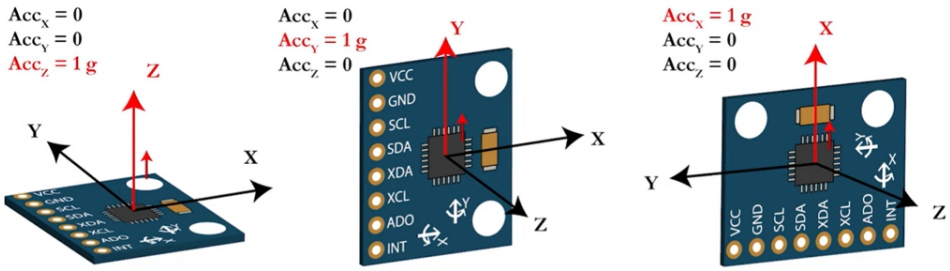


Figure 3.1: MPU6050 Acceleration Axis [14]

The idea of the RPS (Revolution per second) method is to use the circumference of the ball as an indication to the distance travelled. The MPU6050 received it's commands through register allocations. The advantage of communicating through registers is that it saves a lot of sketch memory on Arduino which can be advantages when used with NodeJs.The register allocation was inspired by [14].

1g is equal to $9.81\,\mathrm{m/s^2}$ in Fig. 3.1. The logic can be used for the smart ball to assume the current angle of its microprocessor. The angle can be calculated using the atan2 function, so that the microprocessor can distinguish the angles between 2 sides. otherwise the microprocessor will be measuring 180 degrees on one side and the same 180 degrees on the other side. The angle in radians is converted to degrees.The equations below shows angles in the Roll, Pitch and Yaw being calculated.

$$\text{roll\_angle\_raw} = \text{atan2}(AccY, AccZ) \times \frac{180}{\pi},$$
$$\text{pitch\_angle\_raw} = \text{atan2}(AccX, AccZ) \times \frac{180}{\pi},$$
$$\text{yaw\_angle\_raw} = \text{atan2}(AccX, AccY) \times \frac{180}{\pi}.$$

Another issue was raised on how when it reaches 0 degrees it goes straight to -180 degree. For better accuracy, the microprocessor will need to have a full 360 degrees. The issue was solved by including a command, If any of the raw angle values are lower than or equal to zero then the angle should stay the same as the raw angle value otherwise the angle would be 360 +(the negative value), this will assign a full 360 degrees to a full rotation.

The idea of the RPS method is to use the current angle and subtract by the previous angle as way to measure the distance the ball travelled within this rotation. Another concern was raised when the ball reaches 360 degrees it will go to 0 after, if we calculate the angle difference between around 360 and around 0, it will assume there has been a

huge rotation. Another conditional statement was placed. The elapsed time for every loop is 0.002 to 0.004 s, so a rotation of 180 degrees within 0.004s is impossible, so if the angle difference is higher than 180 degrees during this time step, subtract the difference by 360 degrees if its lower than -180, increase the difference by 360 degrees, this will in return stop giving an unrealistic rotation size. Since the angle measurements are now accurate and consider the full rotation of the ball, The RPS method can now be applied. First of all, The angular speed (degrees/s), using the angle difference and the elapsed time.

$$\begin{aligned}
\text{angular_speed_roll} &= \frac{\text{angle_difference_roll}}{\text{deltaTimeSeconds}}, \\
\text{angular_speed_pitch} &= \frac{\text{angle_difference_pitch}}{\text{deltaTimeSeconds}}, \\
\text{angular_speed_yaw} &= \frac{\text{angle_difference_yaw}}{\text{deltaTimeSeconds}}.
\end{aligned}$$

The revolution per second is calculated by dividing the speed with 360 degrees. how many revolutions are made every second from this angular speed.

$$\begin{aligned}
\text{RPS_roll} &= \frac{\text{angular_speed_roll}}{360}, \\
\text{RPS_pitch} &= \frac{\text{angular_speed_pitch}}{360}, \\
\text{RPS_yaw} &= \frac{\text{angular_speed_yaw}}{360}.
\end{aligned}$$

The displacement is calculated using the diameter of the golf ball which is 0.04268 m, the elapsed time, the RPS value and PI (3.14).

$$\begin{aligned}
\text{distance_roll} += (\text{RPS_roll} \times \pi \times 0.04268) \times \text{deltaTimeSeconds}, \\
\text{distance_pitch} += (\text{RPS_pitch} \times \pi \times 0.04268) \times \text{deltaTimeSeconds}, \\
\text{distance_yaw} += (\text{RPS_yaw} \times \pi \times 0.04268) \times \text{deltaTimeSeconds}.
\end{aligned}$$

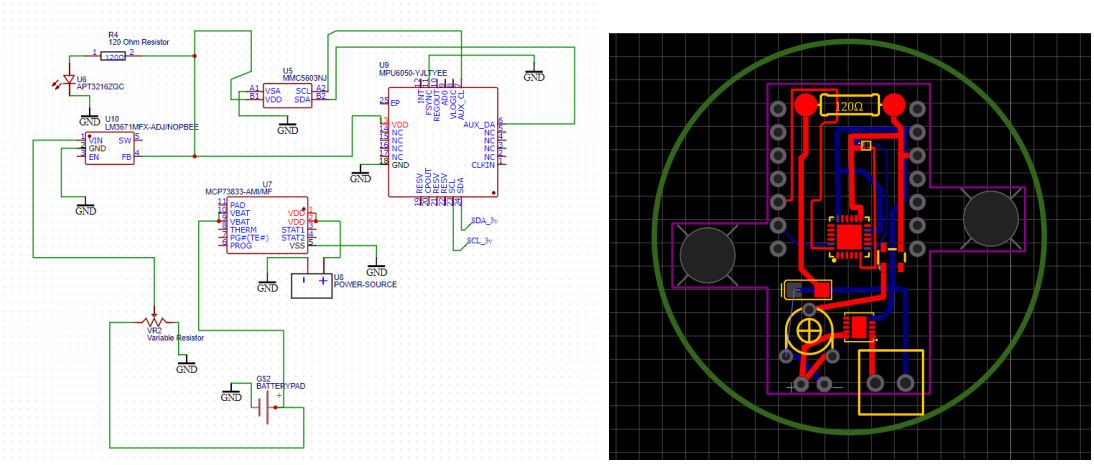
3.1.2 The smart golf ball for the firmness test

How the Clegg impact tester works is by using the maximum deceleration of accelerometer as an indication of the firmness of the ground.The difference of acceleration between 2 points, this will run in the code every 0.002 - 0.004 second. The process is done for X,Y and Z

A conditional statement is then included that if there is a deceleration indicated by delta_accel, go to the next condition. The next condition records the maximum deceleration if there is a new one. At the end the next acceleration is recorded as the previous acceleration for the next loop

A Web Socket approach is used to monitor the results collected wirelessly when inside the ball. .

3.1.3 EasyEDA design



(a) Smart golf ball PCB Schematic (b) Smart golf ball PCB footnote

Figure 3.2: Easyeda design smart golf ball

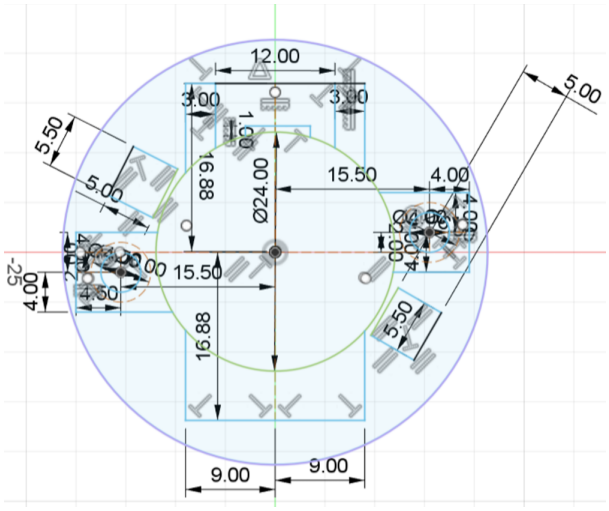
The schematic in Fig. 3.2a shows the wiring made between each of the components. The Battery charger controller (MCP73833) is connected to the power source (Receiver Coil) at pins 1 and 2. To charge the lithium battery connected at pins 9 and 10. The battery charger controller is connected to the ground at pin 5.

The battery is also connected to a variable resistor to turn on and off the PCB to save battery charge. An LED connected as well to indicate if the PCB is on or off, it is connected to a 120 ohms resistor since the LED will require less power. The variable resistor is connected to a voltage regulator since the accelerometer operates on 3.3 volts and the lithium battery sends a 3.7 voltage, it is better to minimise the amount of voltage that passes through. The accelerometer (MPU6050)is connected to the power at pin 13. Pin 18 is connected to the ground. Pin 23 and 24 which are the SCL and SDA will be connected to AUX_CL and AUX_DA of the ESP32. According to the MPU6050 data sheet [2] these are connected to the magnetometer pins (MMC5603).

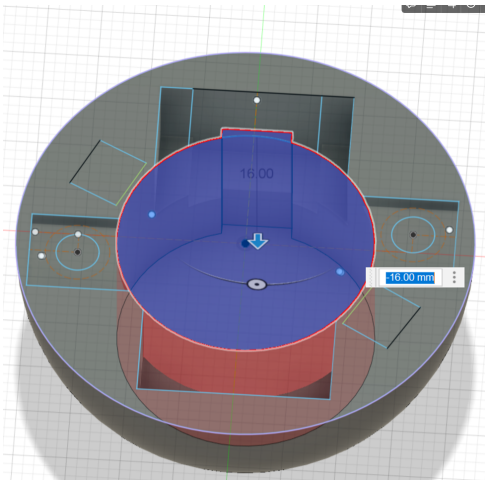
The idea of the PCB is for it to be soldiered under the ESP32, using the pins provided on the right and left of the foot print in Fig. 3.2. The Pin holes are 1 mm each identical to the the pin holes for the Esp32-C3. The GND, the SCL_3v and SCL_EXIT traces are drawn at the bottom of the PCB while Everything else is drawn at the top of the PCB. The 3.4 mm diameter screw holes on the far right and left are made to hold the PCB in place when rolling, they are insured to be on the same height as the accelerometer to make sure that the accelerometer is at the centre of the ball. Both screw holes are 15.5 mm away from the accelerometer.

3.1.4 Fusion360 design

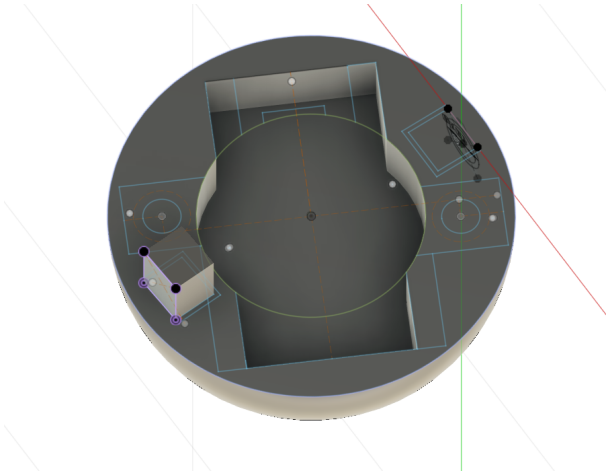
The ball created is a sphere that is 42.67 mm in diameter, The ball was split into 2 parts. Fig. 3.3 shows the extrusion.



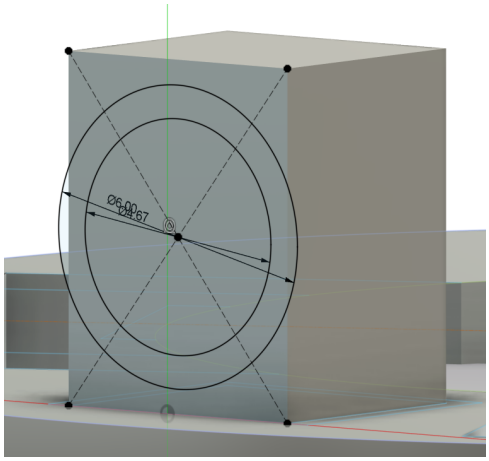
(a) Smart golf ball Sketch



(b) Smart golf ball bottom part



(c) Smart golf ball top part



(d) Smart golf ball connector block

Figure 3.3: Fusion360 design of smart golf ball

The entire sketch was extruded by 1.6 mm to accommodate the PCB on the ball. Two 4 mm deep holes were added for screwing the PCB into the ball while taking into consideration the heat inserts.The bottom part of the ball was further extruded by 8.1 mm to accommodate the maximum thickness (6.5 mm) of the battery. The wireless receiver coil has an outer diameter of approximately 21 mm and 0.79 mm thickness, an additional 2 mm extrusion was made to create space for the coil and it's micro controller. The coil was positioned 6 mm closer to the bottom for optimal operation within it's 2-10 mm operating distance range. The design consists of two parts, each with an extruded block featuring screw holes for assembly using 1-2 screws. Fig. 3.3d shows one of the connector blocks. Each block will house a 4 mm long heat threaded insert. The screws, 3 mm in length, will have heads larger than the block holes for secure attachment.

The smart golf ball codes, Fusion 360 designs and EasyEDA designs are provided in the git hub link below. [1]

[1]github link: https://github.com/YoussefOkab-HWU/LOG_Book/tree/main/smart%20golf%20ball%20Arduino

3.2 Soil data extraction from PDF

3.2.1 Labelling and training

The project was inspired by [15] and [16].The values from the PDF files will need to be detected using YOLO V7, the problem is the values will always change. So, it's hard to find any similarities in changing values. In order to fix the issue, two labels were created for each value. The first label is the area which will contain the variable name and the result. the second label has the result which will be a rectangle around the result while inside the first label. The area label will use the variable name as an indication it's in the right place, while the result label will use the area label as indication that it is in the correct place. An example of one of the PDF files in the first Patch that are labelled using CVAT.ai shown in Fig.3.4.

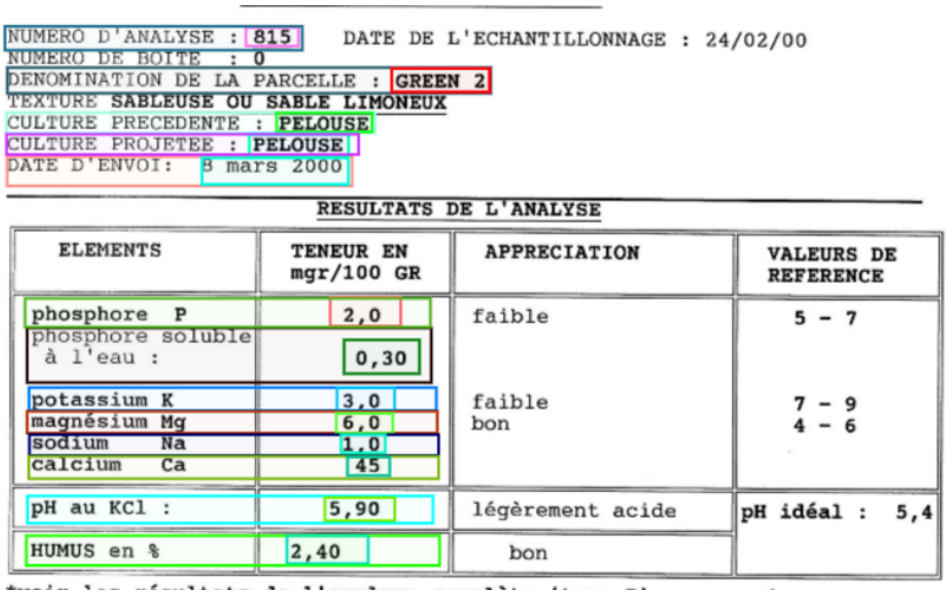


Figure 3.4: Document label example

In the initial training batch, 3 different paper structures were labelled using CVAT.ai across more than 100 PDF pages. Text files containing label coordinates were downloaded for training. Two folders were set up: "train" and "val", each containing "images" and "labels" sub folders. YOLO training iterates between learning the "train" set and testing it's accuracy on the "val" set to refine its detection accuracy. The process yields a "best.pt" file with the optimal model weights for subsequent detection tasks. The dataset is trained with an Epoch of 1000, indicating how many times it will test itself on the Val folder. More epochs will conclude a better chance of detection. We will then use the best.pt folder to detect the values we need.

3.2.2 Object detection and EasyOCR Integration

The process is simplified as shown in Fig. 3.5.

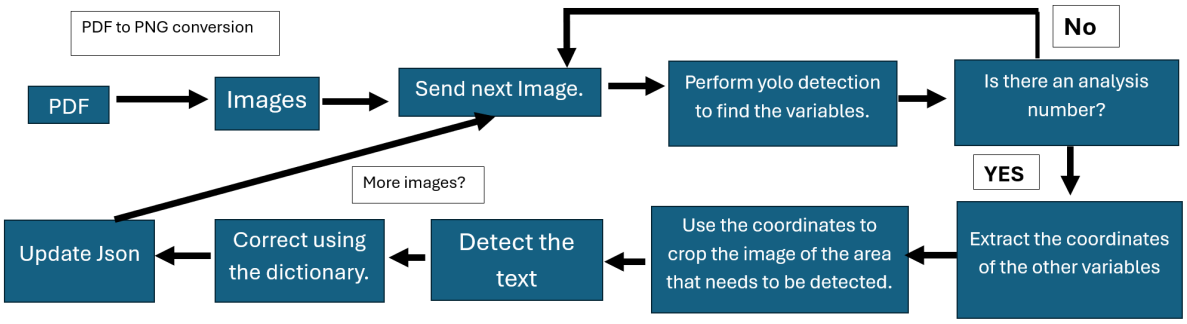


Figure 3.5: code flow chart

The packages YOLO V7 [18] and EasyOCR [19] were downloaded from GitHub. The PDF is split into a number of images then deletes the PDF to avoid causing confusion. YOLO detection is performed on the first image. The PDF that should be detected has an analysis number, so a command was given, if analysis number detected continue the process. If not, go to the next image. The extracted coordinates are used to crop the image of the value that needs to be detected. The text is then detected and corrected using a dictionary made to correct text detection errors. The values are then uploaded as a Json file and regularly updated after every PDF images. The code is called "nifi_yolo_soil_extraction_version_3.py" and provided in the footnote below [2]. The code can be found in the folder called "VScode for water meter and soil Data extraction"

3.2.3 Apache NiFi

NiFi is a potent enterprise-grade data flow management solution that is scalable and dependable in its ability to gather, route, enhance, convert, and process data [17]. The NiFi flow diagram produced for the project is shown in Fig. 3.6.

In Fig. 3.6, A PDF file is moved from the "pdf_file" folder to "pdf_file_output".The execute stream command block starts by splitting the PDF into images and assigning Json file with a UUID that is the same as the file name. After image creation, the original PDF file is deleted from "pdf_file_output". YOLO V7 detects data, while EasyOCR extracts text, producing a Json file. Once processing completes, "fetchfile" retrieves the Json file, renaming it from "(filename).pdf" to "(filename).Json" for compatibility before sending it to the API server.

[2]github link:https://github.com/YoussefOkab-HWU/LOG_Book/tree/main/

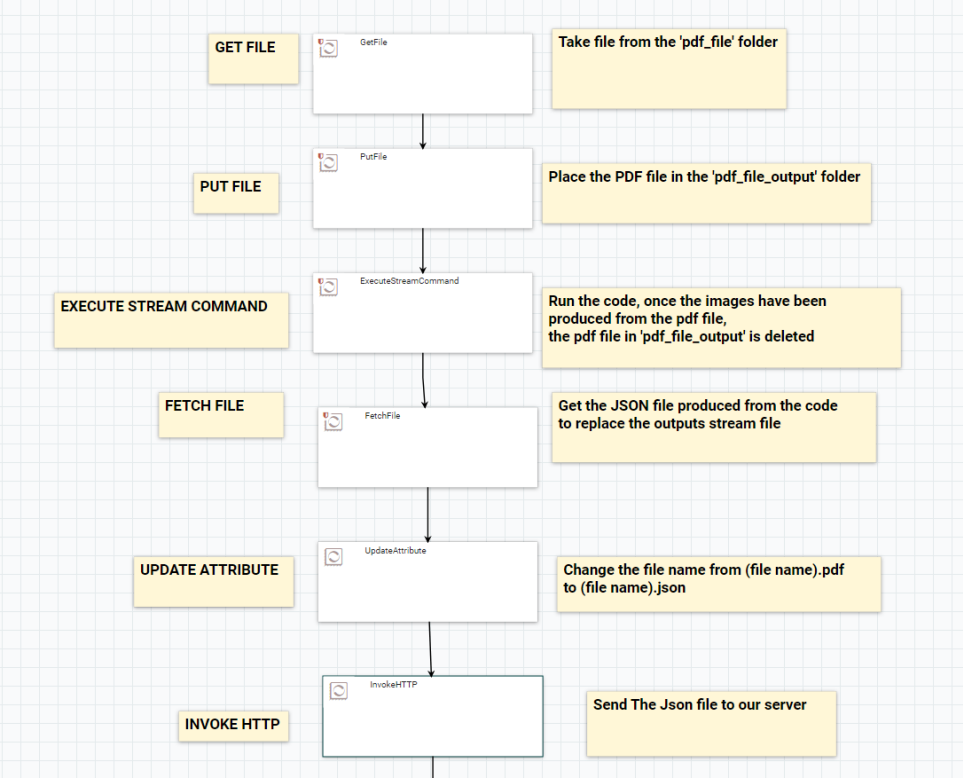


Figure 3.6: NiFi flow chart

3.2.4 PHP API development and SQL Database

The goal was to develop an API endpoint for the Json data provided by NiFi to be uploaded to an SQL database which in return will update the Maya Dev website. Due to the size of the work, This section will only present the key points of the project. There are 3 PHP files (api.php , odata.php , ActionController.php, MayaActions.php) were edited in this process. For the api.php file a single line is added to define the route that will be accessed by the API, Odata.php sets up the parameters that should be expected from the Json format and MayaAction.php is the most important file in the 4 PHP files that were edited. The code follows the following steps

1. Create the function for the postSoilDataAnalysis.
2. Set up a variable of extracting the ”Date” and the ”iddocument” from the Json file.
3. Set up a databyanalysis array, since each data section of the Json received does not have the complete information. They all have an analysis number which can be used to assign the correct data values to the right “name of the plot”/”date”.
4. For each input data extract analysis number, if analysis number is not in the array initialise the data array with it’s information.
5. Set up a loop for all the substance received from each data entry to extract the substance name and value.
6. Re-define the analysis numbers in the array as other_analysis_number, if there is a match for the current analysis number the data merging will begin replacing all the N/A values with the actual values

7. We then access the array we have databyanylysis for each analysis number for the SQL data base, for each substance aswell if the value is not 'N/A', add it to the data base.

8. There can never be 2 of the same substance in the "substance" database that why we set up the variable $matchingsubstance to look and see all the substances in "substance" database

9. If there is a matching substances, we take the idsubstance for the existing substance and use it for the other 2 databases, if the substance is a new idsubstance is generated.

10. After the whole process is finished, it is then added to 3 different SQL tables, which are soil analysis, substance and soil substance.

Data grip is used to view the SQL database, the SQL 8.3 database is initialised on Docker. Since NiFi is running on Linux and PHP is running on windows, Port connection between NiFi and the local computer was created through windows defender firewall.The process needs to be accessible to the average consumer. So,it was necessary to create an API that adds the URL document to the SQL Database, takes the URL document from the SQL Database if it's a pdf document then downloads the file to the Ubuntu 20.04 directory.

3 files(api.php,RecordController.php and MayaActions.php) were edited to have a successful postDocument API. For api.php we route the process to the record controller. Record controller will use the already existing API handlepost and added to the "document" SQL database.In Maya actions the following procedure is as follows:

1. When the document is posted all the information are added to the "document" SQL database, however there are two key information needed from the post request and that is name and type.

2. The iddocument value is extracted from the SQL database by looking for the name value, in addition it is told to look for the latest entry in case there are 2 or more entries with the same name.

3. If the type is not soil_analysis_excel get all the information from the $Iddocument row, since it is a PDF.

4. If urlacquired is not empty define the $url variable as the storage value (the url link).

5. The PDF is downloaded on the pdf_file directory on Ubuntu 20.04 where the NiFi process will start,the file name will be named as the [Iddocument].pdf.

The php codes is provided on the footnote below as a GitHub link [3]

[3]github link: https://github.com/YoussefOkab-HWU/LOG_Book/tree/main/PHP%20CODE

3.3 Text detection for analogue water consumption meter.

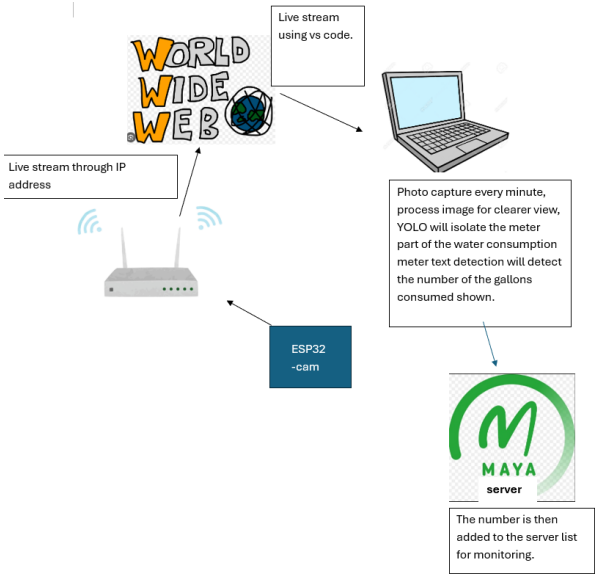
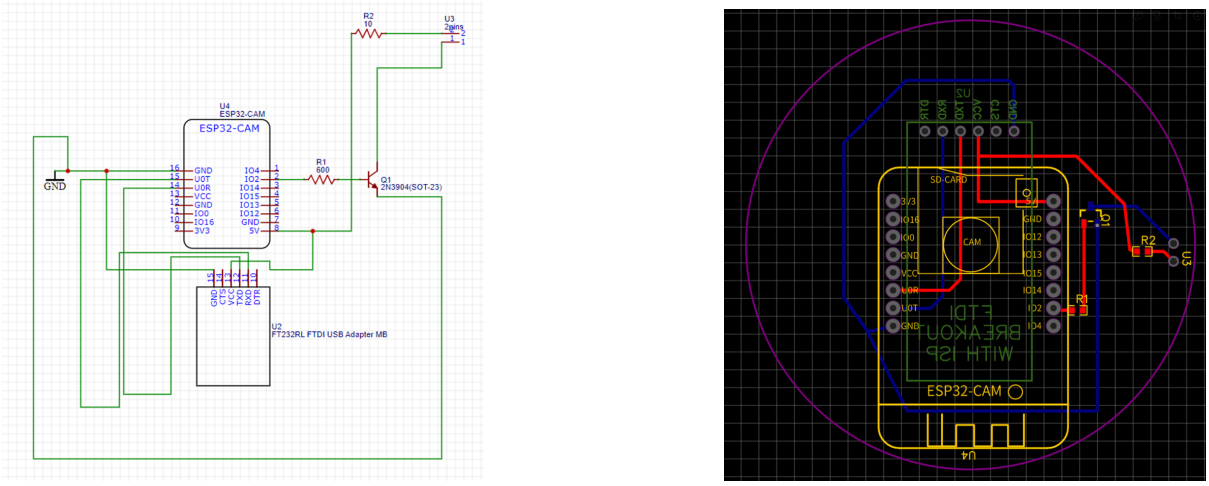


Figure 3.7: Concept plan for Text detection for analogue water consumption meter

According to Fig 3.7 the camera will live stream the water meter it will take a screenshot once every minute and send it for YOLO V7 and paddle OCR detection it will then update the SQL database with the new value presented on the water meter.

3.3.1 EasyEDA design



(a) Camera PCB Schematic

(b) Camera PCB footnote

Figure 3.8: Easyeda design camera

The PCB holder integrates the ESP32-CAM and FT232RL Mini USB to TTL Serial Converter Adapter on opposite sides. LED is connected to a transistor via data pin 2 of

the ESP32-CAM which will light up once every minute during live stream for a screenshot. The transistor manages current to the LED ring by adjusting for the ESP32-CAM is 310 mA output (less than the LED's 500 mA requirement)to ensure a proper operation with a 600-ohm resistor between the transistor and IO2 Pin. The schematics and the foot print are shown in Fig 3.8. The PCB design are provided in the GitHub link below [4]

3.3.2 Fusion 360 design

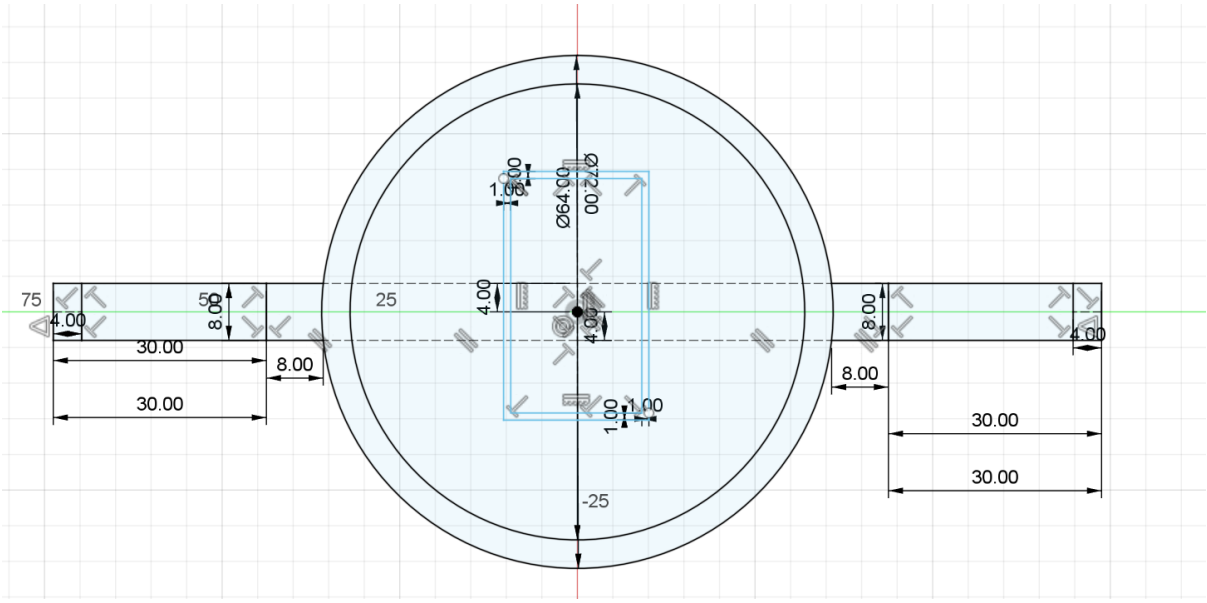


Figure 3.9: Cam mount fusion360 Sketch

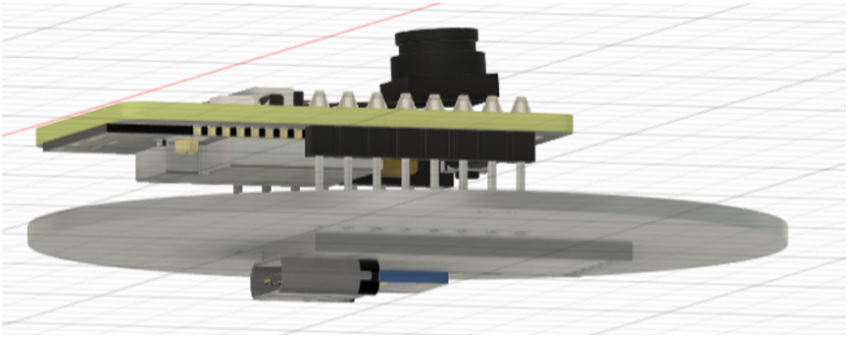


Figure 3.10: PCB for cam

The esp32Cam will be soldered top and the ftdi component will be soldered on the other side the PCB as shown in Fig. 3.10. The PCB is 64 mm in diameter. The LED light is represented as 70 mm outer diameter with 65 mm inner diameter. The component is designed as two parts the bottom part will hold the PCB and will have a port to power the PCB. Then there is the second part will work as a cover while leaving a hole for the camera and a hole for the LED wires to go through to the ports on the PCB. The case design are provided in the GitHub link below [5]

[4]github link: https://github.com/YoussefOkab-HWU/LOG_Book/tree/main/Camera%20case%20easyeda
[5]GitHub link: https://github.com/YoussefOkab-HWU/LOG_Book/tree/main/camera%20case%20designs

3.3.3 Training and VS code

Text detection for water meters was a main concern since reading the water meter was not as straightforward as reading a document, water meters are unclear and there was the main concern if image processing is required.

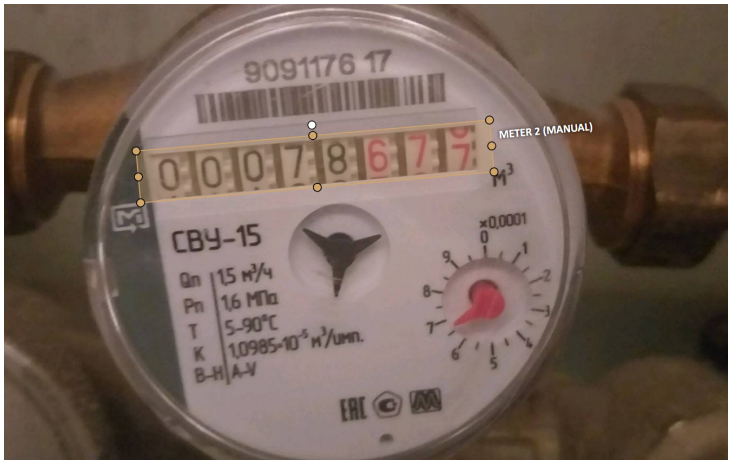


Figure 3.11: Water meter label

Same as before the water meter is labelled and trained. A dataset of 1243 images [20] was used and was labelled for each image like Fig. 3.11, the dataset was then trained same as last time. PaddleOCR was downloaded from the GitHub link [21]. The coordinates will be extracted, however the cropped image will be saved in the meter_scan_folder as cropped_image.png. The PaddleOCR will then be used to detect the text. The code will then extract the number from the printed text in the terminal. The code is called "paddle_water_meter.py" and provided in the footnote below [6]. The code can be found in the folder called "VScode for water meter and soil Data extraction"

Chapter 4 Results

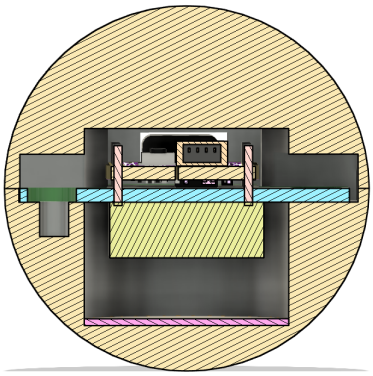
The work that was done on all 3 projects was extensive, however the smart golf ball and water meter was not completed. Due to Inexperience in PCB design and soldering. The PCB couldn't be ordered without getting an expert to check the traces and the components. Regardless experiments were made with the codes written and concluded valuable outcomes.

4.1 Smart Golf Ball

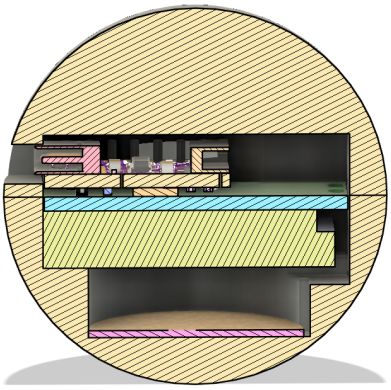
4.1.1 EasyEDA & Fusion360 designs

Fig. 4.1 shows the different perspectives of the design. The wireless coil is placed at the bottom of the ball above it is the battery which will power the PCB. There is also a port opening made for compiling code when changes need to be made. The design of the screw holes to close both parts of the sphere is a creative approach to see if it's possible

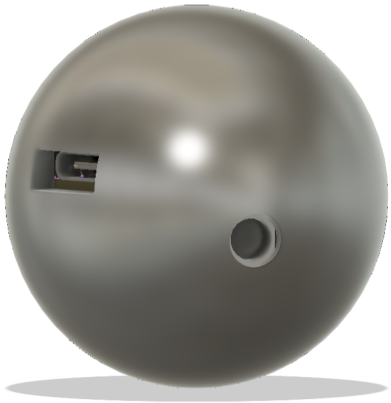
[6] github link:https://github.com/YoussefOkab-HWU/LOG_Book/tree/main/



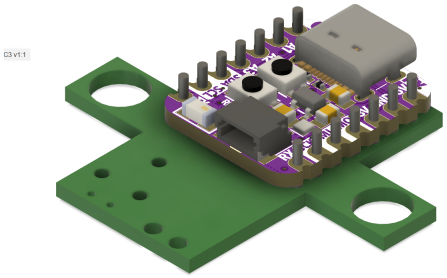
(a) Front Analysis section of the smart golf ball



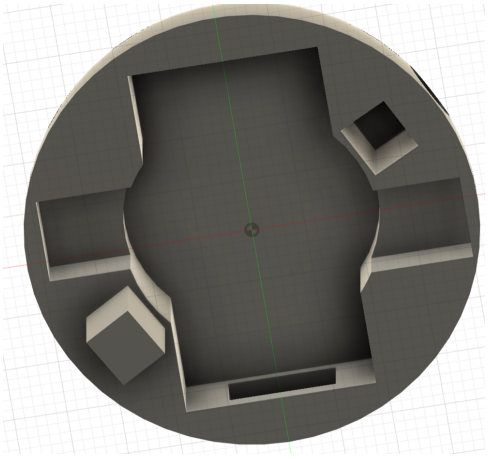
(b) Side Analysis section of the smart golf ball



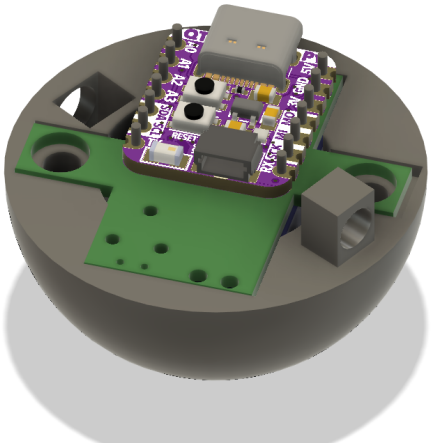
(c) Full Smart golf ball design



(d) ESP32 + PCB



(e) Smart golf ball top design



(f) Smart golf ball bottom design

Figure 4.1: Smart golf ball fusion 360 design

to close the ball with only one screw. Two screw holes were installed as a precaution to make sure the ball is fully closed without impacting the results.

4.1.2 Arduino code tests

The test were run on a bread board using Arduino IDE. For the RPS method and the Clegg impact test while also testing the Web socket approach. Video links are also provided as a means for illustration

The RPS method

The test involved rotating the breadboard clockwise and anti-clockwise around the pitch axis direction and observe the change in displacement in the "distance_pitch" direction Providing the lateral and vertical deviation of the ball.

Experiment video: https://youtu.be/oNpVEPpafUY

Clegg impact test using the smart golf ball

The Clegg Impact test is used to calculate the firmness of the ground. This is done through obtaining the maximum deceleration. The bread board is moved around and dropped on the table causing a high maximum deceleration.

Experiment video : https://youtube.com/shorts/t778uwvnH6M?feature=share

The RPS method through the web socket approach

It was assumed that tests will run in an actual ball. A web socket connection was made to test the components working and provide information regarding displacement without connecting to the laptop.

Experiment video : https://youtu.be/vP2om0YUTZ0

4.2 Soil data extraction from PDF

The API was tested using postman were

1. The document was sent via postmen to the SQL Database called "document"
2. The document was extracted from the SQL database and downloaded on Ubuntu 20.04 directory starting the NiFi process.
3. The code ran on the document and produced a Json format
4. The Json format was fetched and sent to the postsoilDataAnalysis, the Json was read and arranged into 3 SQL Databases (Soil_analysis, substance and soil_substance).

The video of the entire experiment is shown below

Experiment video : https://youtu.be/Jn2xEmodZBg

A number of PDFs were tested on the code giving impressive results. Some documents were not successfully trained since there was not enough dataset provided for the new document,It was found that 30 pages of a Dataset are required to run a perfect training. Regardless The result provided were satisfactory. The code can analyse a 100 pages in under 18 minutes saving the consumer a great amount of time.

4.3 Text detection for analogue water consumption meter.

4.3.1 Fusion 360 design

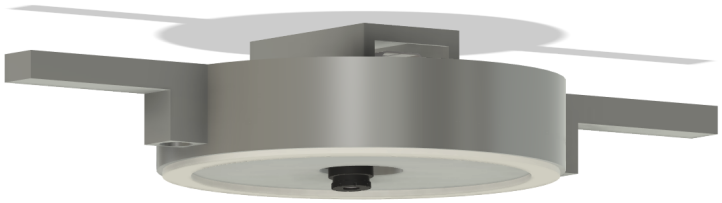


Figure 4.2: Full camera mount with LED

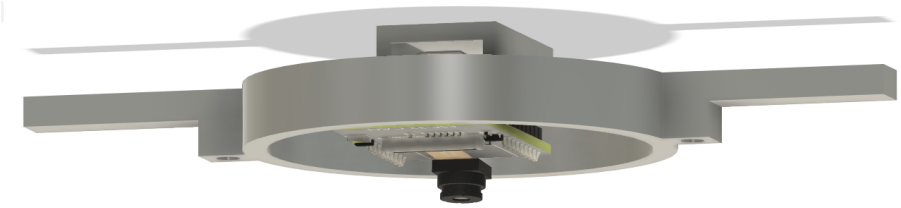


Figure 4.3: Bottom Camera mount

The camera mount design in Fig.4.2 shows the entire design of the camera case with two extension sides that will be attached to 2 clamps attached to both sides of the pipes of the water consumption meter.The holes on the bottom part are 4 mm in diameter and height to hold the heat insert, and for the holes on top part are 6 mm in diameter to hold in the head of the screw. The clamp will be held on both handle while the other side of the clamp will hold on to the pipes. The LED will flash for every time the Esp32 CAM is about to take a screenshot of the live stream. On the back of the case there will be a power port since the camera will live stream 24/7. The components can not run with a chargeable battery without wasting the consumer's time.

4.3.2 VS code results

The code perfectly analysed the the number on the water meter. However the code needs to go through a practical run in real life rather than read through pictures to explore the need of using image processing to improve detection. Fig. 4.4 shows the number extracted at the end of the terminal.

Figure 4.4: Paddle OCR detection test

Chapter 5 Conclusion

All work done in the internship was documented as a step by step tutorial for the next interns to continue the work that was done, PHP codes were committed and merged on GitLab. The codes written on Vs Code and Arduino IDE were downloaded as zip files on GitLab and SharePoint. Fusion 360 and EasyEDA designs were also downloaded to share point in case the next interns would like to make some changes to the design.

The smart golf ball project was unfortunately not tested on a golf ball due to lack of experience in PCB construction, however the code was tested effectively on a bread board and I am sure the necessary calibrations can be made once the design is created.

Text detection for analogue water consumption meter provided an incredible start in detecting the numbers on the analogue water meter, The next interns will have to test the process on a real life analogue water consumption meter, to see if image processing is necessary and develop an API to take in the number and add it to the SQL Data base.

Soil data extraction from PDF has had a successful run, The YOLO V7 has also trained 2 more different types of documents with no issues. Some documents were not able to be trained because in order for a successful training to occur atleast 30 different dataset images are needed and some of the documents received were only 5 pages. Regardless the fact that the project was successful shows the method can work if provided with proper training.

Bibliography

[1] About Us (2023) Maya Global. Available at: https://mayaglobal.io/about-us/ (Accessed: 25 July 2024).

[2] InvenSense Inc., 2013. MPU-6000 and MPU-6050 Product Specification. Document Number: PS-MPU-6000A-00, Revision 3.4. Sunnyvale, CA: InvenSense Inc. Avail-

able at: https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf [Accessed 2 July 2024].

[3] Microchip, (2009). MCP73833-AMI/un Microchip, Battery Charger. Available at: https://uk.farnell.com/microchip/mcp73833-ami-un/li-ion-li-poly-charge-controller/dp/1332160 [Accessed 2 July 2024].

[4] Adafruit Industries, (no date) Adafruit QT Py ESP32-C3 WIFI Dev Board with Stemma QT. Adafruit Industries Blog RSS. Available at:https://www.adafruit.com/product/5405 [Accessed 1 July 2024].

[5] Ufine Battery, (no date) 3.7 V 1000mAh lithium ion battery 803040. Available at: https://www.ufinebattery.com/products/3-7-v-1000mah-lithium-ion-battery-803040/ [Accessed 1 July 2024].

[6] The Pi Hut, (no date) Wireless charging module 5V/300mA. Available at: https://thepihut.com/products/wireless-charging-module-5v-300ma?variant=27740714833&currency=GBP&utm_medium=product_sync&utm_source=google&utm_content=sag_organic&utm_campaign=sag_organic&gad_source=1&gclid=CjwKCAiA8YyuBhBSEiwA5R3-E2gmMQLaUrp8lD8LtOeUhGfbUN0yLqJLS9Oag5wm1abDS6PimEFwsBoCpywQAvD_BwE [Accessed 1 July 2024].

[7] Accu.co.uk, (no date) M3 x 3mm Phillips Pan Head Machine Screws (DIN 7985H) - 18-8 / 304 Stainless Steel. Available at: https://accu-components.com/us/phillips-pan-head-screws/65400-SIP-M3-3-A2 [Accessed 1 July 2024].

[8] Adafruit Industries, (no date) Brass heat-set inserts for plastic - M3 X 4mm - 50 pack. Adafruit Industries Blog RSS. Available at: https://www.adafruit.com/product/4255 [Accessed 1 July 2024].

[9] Admin, (2023). ESP32 cam object detection identification with OpenCV. How To Electronics. Available at: https://how2electronics.com/esp32-cam-based-object-detection-identification-with-opencv/ [Accessed 2 July 2024].

[10] Amazon.com, (no date) Atnsinc 3PCS FT232RL mini USB to TTL FTDI Adapter Module, 3.3V 5.5V FT232R breakout FT232RL USB to Serial Converter Adapter Board : Electronics. Available at: https://www.amazon.com/FT232RL-Adapter-Breakout-Converter-Arduino/dp/B08B878T7T [Accessed 2 July 2024].

[11] Indiamart.com, (no date) Bipolar (BJT) transistor NPN 40 V 600 mA 300MHz 500 mW through hole TO-18. Available at: https://www.indiamart.com/proddetail/bipolar-bjt-transistor-npn-40-v-600-ma-300mhz-500-mw-through-hole-to-18-2850339896562.html [Accessed 2 July 2024].

[12] RS, (no date) TE connectivity 100Ω carbon film resistor 1W ±5% CFR100J100R. Available at: https://uk.rs-online.com/web/p/through-hole-resistors/0131772 [Accessed 2 July 2024].

[13] Adafruit Industries, (no date) Cool white LED ring light PCB - 70mm diameter - 5V power. Adafruit Industries Blog RSS. Available at: https://www.adafruit.com/product/5138 [Accessed 2 July 2024].

[14] Carbon Aeronautics (2022) 14 | Measure angles with the MPU6050 accelerometer. Available at: https://www.youtube.com/watch?v=7VW_XVbtu9k [Accessed 2 July 2024].

[15] Computer vision Engineer (2023) Automatic number plate recognition with Python, YOLOv8 and EasyOCR | Computer vision tutorial. Available at: https://www.youtube.com/watch?v=fyJB1t0o0ms[Accessed 2 July 2024].

[16] The Coding Bug (2021) Official YOLO v7 Custom Object Detection Tutorial | Windows Linux Available at: https://www.youtube.com/watch?v=-QWxJ0j9EY8[Accessed 2 July 2024].

[17] Pandey, P., Kumar, S. and Chaudhuri, R. (no date) Apache NiFi, What is Apache NiFi - Learning Journal. Available at: https://www.learningjournal.guru/article/apache-NiFi/what-is-NiFi/ [Accessed: 02 July 2024].

[18] WongKinYiu. (no date) YOLOv7. Available at: https://github.com/WongKinYiu/YOLOv7 [Accessed: 2 July 2024].

[19] JaidedAI. (no date) EasyOCR. Available at: https://github.com/JaidedAI/EasyOCR [Accessed: 2 July 2024].

[20] Roman, K. (2020) Water meters dataset, 1244 Photos Masks, Kaggle. Available at: https://www.kaggle.com/datasets/tapakah68/yandextoloka-water-meters-dataset [Accessed: 02 July 2024].

[21] PaddlePaddle. (no date) PaddleOCR. Available at: https://github.com/PaddlePaddle/PaddleOCR [Accessed: 2 July 2024].