GMIT

INSTITIÚID TEICNEOLAÍOCHTA NA GAILLIMHE-MAIGH EO

GALWAY- MAYO INSTITUTE OF TECHNOLOGY

TackleBac

“A proximity and cleanse detection system”

Project Engineering

Year 4

Ricky Small

Bachelor of Engineering (Honours) in Software and Electronic Engineering

Galway-Mayo Institute of Technology

2020/2021

GMIT
INSTITIÚID TEICNEOLAÍOCHTA NA GAILLIMHE-MAIGH EO
GALWAY-MAYO INSTITUTE OF TECHNOLOGY
TACKLE BAC+
" A Proximity and Cleanse Detection System "
TB+
TACKLEBAC CLEANER
ROOM : 207
REQUIRES CLEANSING

Figure 1 - Project Graphic

DECLARATION

This project is presented in partial fulfilment of the requirements for the degree of Bachelor of Engineering (Honours) in Software and Electronic Engineering at Galway-Mayo Institute of Technology.

This project is my work, except where otherwise accredited. Where the work of others has been used or incorporated during this project, this is acknowledged and referenced.

ACKNOWLEDGEMENTS

I would like to take this time to thank Galway Mayo Institute of Technology for providing a new course on Software and Electronic Engineering. I would like to thank my project supervisor Michelle Lynch for her support and advice throughout the project. Also, I would like to send my gratitude to all the lecturers that I had the honour of being part of their modules. All their work has allowed me to get to where I am today.

All my classmates, I thank you all for the four years of friendship, advice, and support. We have finally come to the end we were all working towards.

I would like to extend my gratitude to my mother Martina Small, as she has always provided me with support and help when the path ahead started to fade. My brothers for the support and courage when needed. Words cannot describe how grateful I am to have had the backing. I look forward to overcoming all your expectations upon finishing my degree.

Finally, I would like to thank my partner Ciara for all her support through returning to education as a mature student. Always supporting me to keep me on track to success.

Thank you all very much!!

"The value of life is not based on how long we live, but how much we contribute to others in our society." -Buddha

"The world is moved along, not only by the mighty shoves of its heroes, but also by the aggregate of tiny pushes of each honest worker" -Helen Keller

TABLE OF CONTENTS

SUMMARY

The world, as we know, has changed rapidly since the end of 2019. The deadly virus known as Covid-19 has engulfed the world throughout. Health regulations and guidelines have resulted in many premises being closed with much uncertainty of when a normal will return. Not only is it devastating for communities, but also business owners lack income which will result in many closures.

“*TackleBac – A proximity and cleanse detection system*”

Aims at preventing bacteria’s and viruses from spreading due to close contact and/or unsanitised locations such as classrooms.

The TackleBac project has the main goal of providing a safe space for people during pandemics such as Covid-19, allowing locations to reopen safely. i.e.: educational sites and places of work. This means locations are sanitized regularly. The device must be affordable and simple to set up, so implementation can be possible throughout all communities across the nation and/or globe.

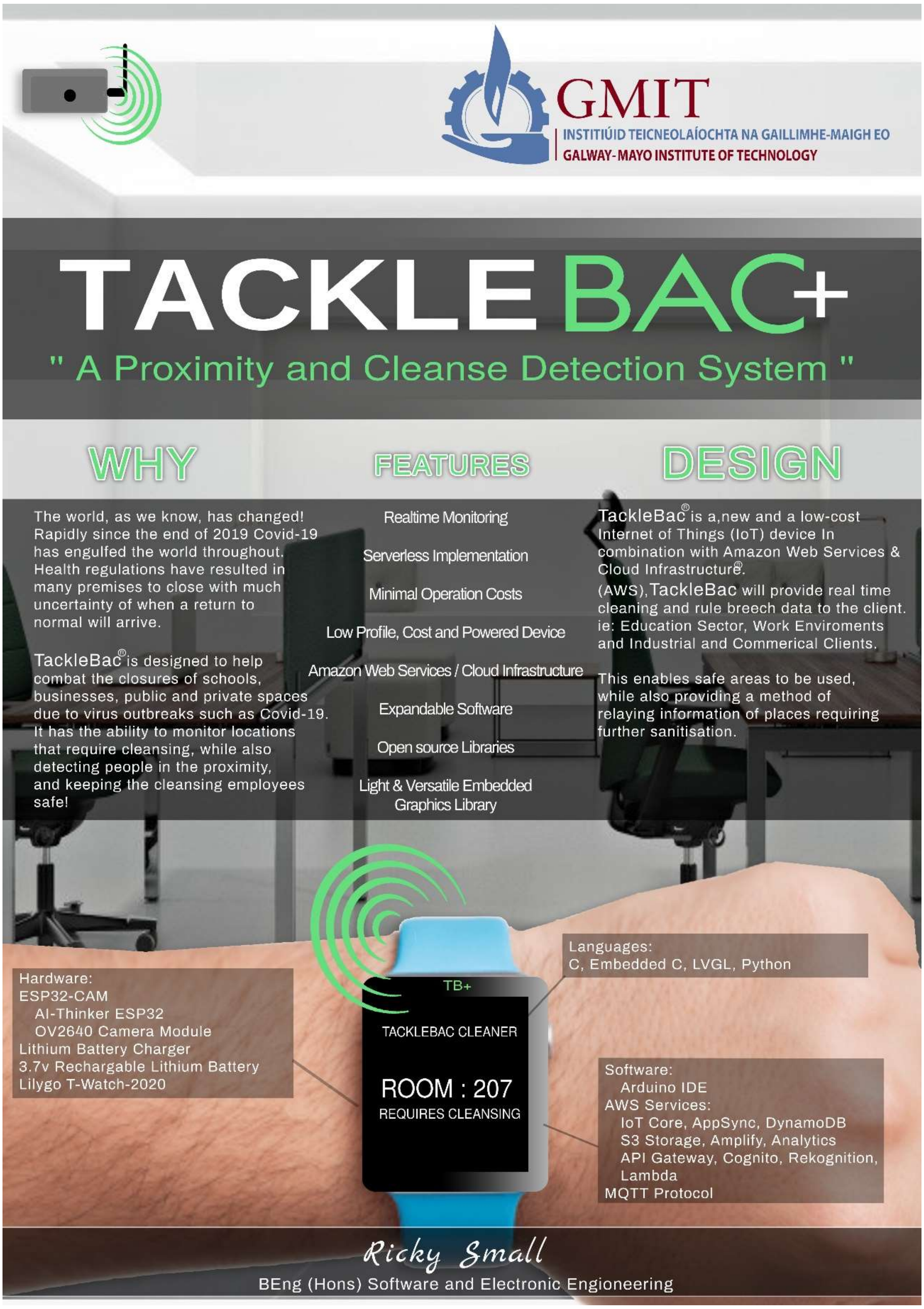
Features:

- Monitoring locations cleanliness
- Providing notifications to cleaners and assigned staff
- Method of updating the status of the location once cleansed
- Motion detection checking room use
- Object recognition to confirm people are present
- Serverless design
- Lambda functions
- Direct communication with the device
- Amazon Web Services (AWS)
  - Internet of Things (IoT Core)
  - DynamoDB
  - Rekognition
  - Amplify

To meet the goals of the project it requires a low powered and low costing device to provide communication with the cloud. Functions embedded, to detect activity in a location and confirming humans are present. Messaging to inform employees of current information and statuses of locations.

This is a final year project for a BEng (Hons) in Software and Electronic Engineering at Galway Mayo Institute of Technology. Learnings via modules completed throughout the course will be implemented into the project, with new technologies and methods researched externally.

POSTER



GMIT

INSTITIÚID TEICNEOLAÍOCHTA NA GAILLIMHE-MAIGH EO

GALWAY- MAYO INSTITUTE OF TECHNOLOGY

TACKLE BAC+

" A Proximity and Cleanse Detection System "

WHY

The world, as we know, has changed! Rapidly since the end of 2019 Covid-19 has engulfed the world throughout. Health regulations have resulted in many premises to close with much uncertainty of when a return to normal will arrive.

TackleBac® is designed to help combat the closures of schools, businesses, public and private spaces due to virus outbreaks such as Covid-19. It has the ability to monitor locations that require cleansing, while also detecting people in the proximity, and keeping the cleansing employees safe!

FEATURES

Realtime Monitoring

Serverless Implementation

Minimal Operation Costs

Low Profile, Cost and Powered Device

Amazon Web Services / Cloud Infrastructure

Expandable Software

Open source Libraries

Light & Versatile Embedded Graphics Library

DESIGN

TackleBac® is a,new and a low-cost Internet of Things (IoT) device In combination with Amazon Web Services & Cloud Infrastructure®. (AWS),TackleBac will provide real time cleaning and rule breech data to the client. ie: Education Sector, Work Enviroments and Industrial and Commerical Clients.

This enables safe areas to be used, while also providing a method of relaying information of places requiring further sanitisation.

Hardware:
ESP32-CAM
  AI-Thinker ESP32
  OV2640 Camera Module
Lithium Battery Charger
3.7v Rechargable Lithium Battery
Lilygo T-Watch-2020

Languages:
C, Embedded C, LVGL, Python

Software:
  Arduino IDE
AWS Services:
  IoT Core, AppSync, DynamoDB
  S3 Storage, Amplify, Analytics
  API Gateway, Cognito, Rekognition,
  Lambda
MQTT Protocol

Ricky Small

BEng (Hons) Software and Electronic Engioneering

Figure 2 - Project poster

INTRODUCTION

This report will bring us through the various aspects of the TackleBac project. The project report will introduce the reader to some technologies used in the completion, with the aid of diagrams and images. TackleBac is being developed by one person, to revive the livelihood of all people throughout the world.

PROJECT GOALS

The goals of the project are as follows:

- Provide a service to help the reopening of society during the pandemic's lockdown
- Ensure cleaners are safe throughout their work hours
- Monitor and report rooms that require cleaning
- Ensure guidelines are adhered to throughout the day
- Provide logging of information in a database, for future reference
- Develop and produce a functional TackleBac Sensor and Cleaners device.

PROJECT SCOPE

The TackleBac sensor will be affixed in each room of use, notifications for cleansing will be sent to the cloud when a human is detected using the room.

The TackleBac cleaners watch will be provided to all cleaning employees, notifying them of rooms requiring cleaning. This will prevent the cleaners from unnecessary cleaning, reducing durations present in potentially contaminated buildings.

PROJECT MOTIVATION

The motivation behind the TackleBac project derives from my altruistic personality. Being brought up in a way to help others if you can, without looking for a reward. That the rewards will follow in time to come.

Returning to college as a mature student was a huge hurdle in my life, as previous courses and educational routes were unsuccessful. The main drive was the one-to-one contact being in college, in a lab with little distractions, and the overall atmosphere of people learning all around.

When Covid-19 being a pandemic and doors were shutting all over the nation and world, worrying about how the final year would pan out was on my shoulders throughout the summer.

The college remained closed for all. A full final year of remote learning with no labs, no one-to-one contact, no learning atmosphere, distractions everywhere and many external pressures noticeable had my drive and grades deteriorate in comparison to the previous three years.

A main key reason for keeping places closed was the inability to control the virus with no way of monitoring activity in public spaces.

So, sitting down with a final year project to envision, how can I help to have the college reopen safely? But also needed to keep the cleaners safe, who would play an important role in the sanitisation of rooms and labs allowing the college to reopen.

If I could help have the college open for people similar to me with learning difficulties, do it safely, to prevent anybody experiencing what it was like for me was a big enough reward for me.

After long brainstorming sessions on my own, TackleBac was born!

TACKLEBAC IN SOCIETY

PANDEMIC LOCKDOWNS

Many people have now experienced the effects of long-term lockdowns in society. The Covid-19 pandemic created amass of job losses, depression, isolation, learning challenges and a new way of confined living. Lockdowns were the thought of a solution to stopping the spread of the virus. As time goes by, slowly guidelines have been refined to bring society back into operation. This however is a slow process with many indoor facilities still behind a closed door. A system that can monitor areas that require cleansing would allow locations to reopen while still provide a safe clean place of interaction.

THE INTERNET OF THINGS

Internet of Things or IoT as it is commonly abbreviated is a term given to devices of any type that connect to the Internet or other IoT capable devices. It allows anything that can be turned on and off to be controlled locally or remotely once set up correctly. It is still a new method of connectivity compared to other options. It has been gathering a lot more interest from people around the world within the last ten years and even more since lockdowns, with schools teaching classes to as young as four and five-year-olds. Hardware with IoT capabilities is in abundance with many variations of the same chips with different features available.

CLOUD COMPUTING

When we look at computers generally, we think of a central processing unit, random-access memory, motherboard, solid-state drive / hard disk for data storage, and a power supply all place inside a computer case. The computer case is connected to a monitor and can be interacted via a keyboard and/or mouse. This is commonly known as a home computer, which many homes have today. In recent times, laptops, and mobile devices eg: tablets, phones, netbooks are more common within the home. However, all mentioned above are limited to power and capabilities based on the hardware specification. Cloud computing allows end-users to use almost unlimited hardware resources remotely, to process instructions. It is possible to extend and upscale cloud computing based on your needs, while also making physical server maintenance non-existent. The results and goals possible with the use of cloud computing is a complete game-changer in the Information of Technology industry. It has allowed TackleBac to communicate and process images at lightning speeds, whereas the low powered low costing device would previously severely struggle to complete such tasks.

PROJECT ARCHITECTURE

Below is the current architectural diagram at the time of writing, for the TackleBac project. TackleBac will remain an ongoing project to enhance, improve and implement new features to required areas. I am expecting elements to change as input is received externally concerning the project. The current working state is reflected in the architecture below.

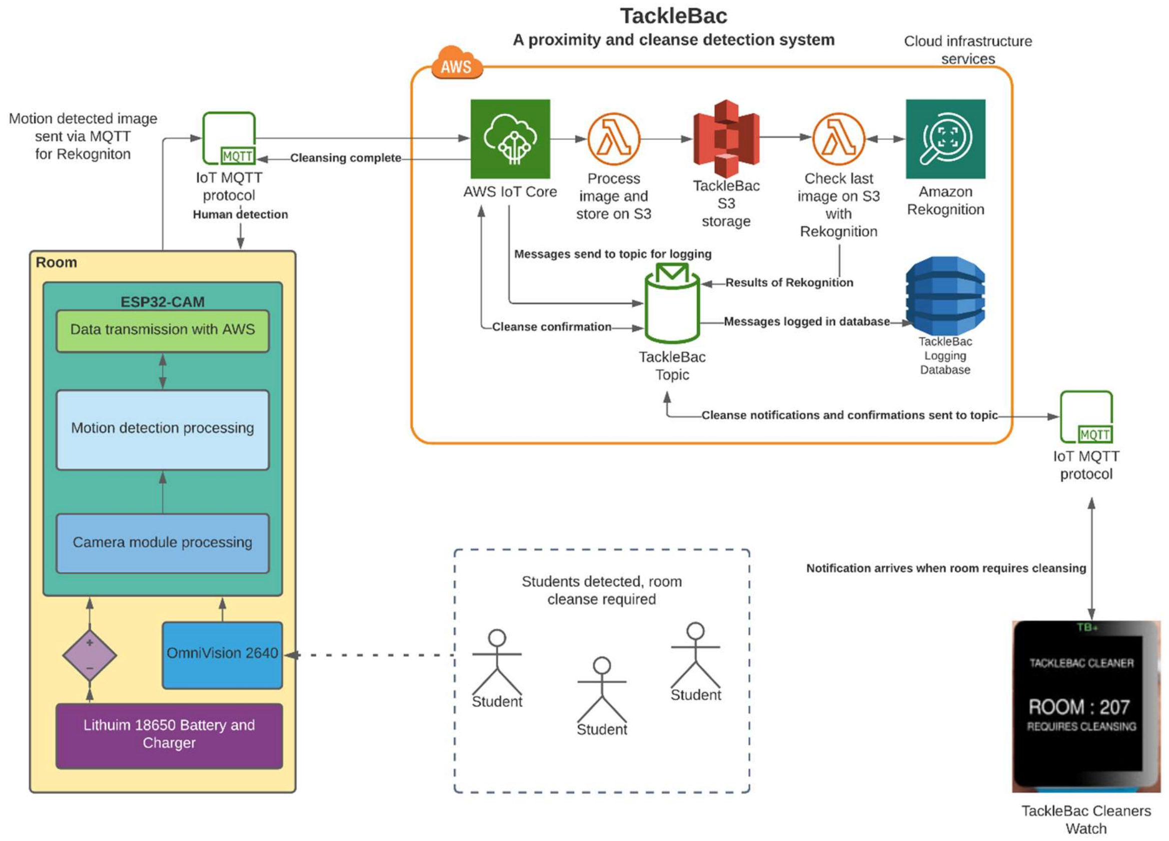
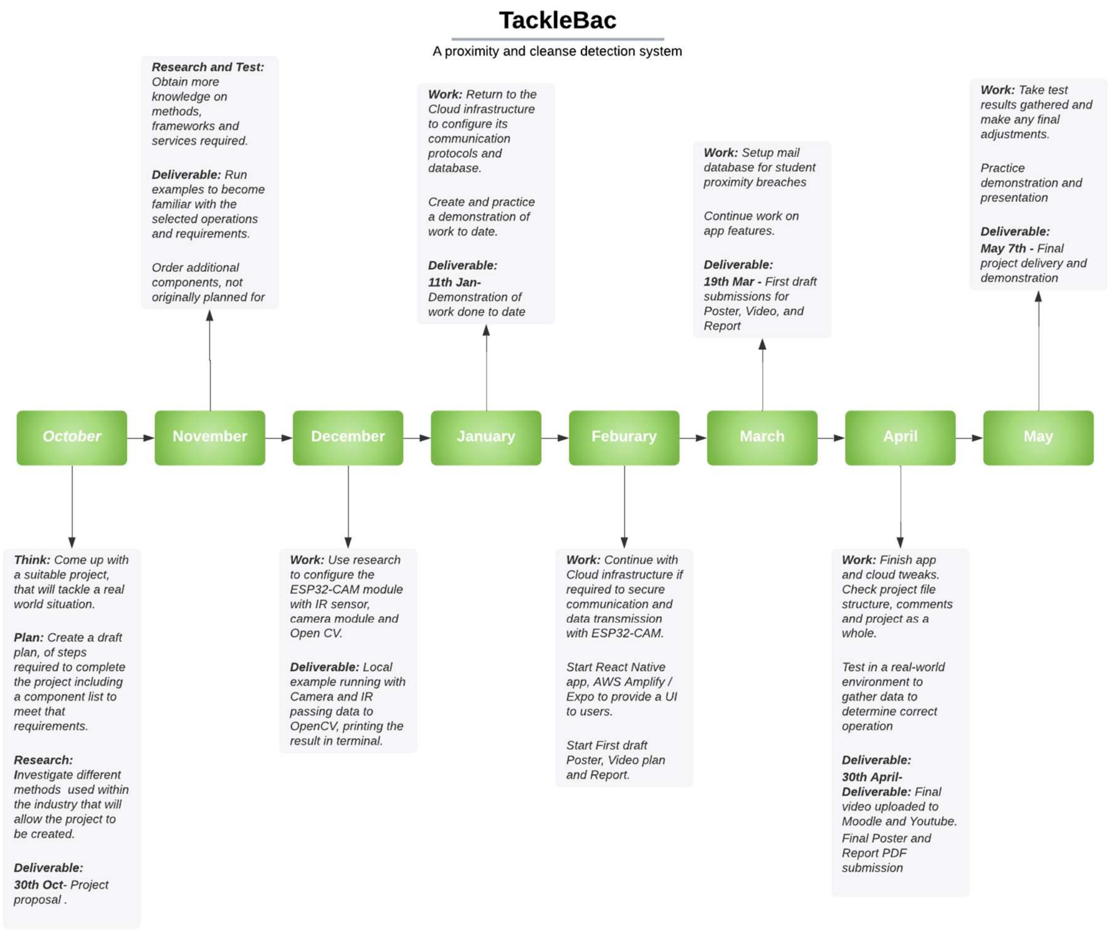

TackleBac
A proximity and cleanse detection system
Cloud infrastructure services
AWS
Motion detected image sent via MQTT for Rekogniton
MQTT
IoT MQTT protocol
Human detection
Cleansing complete
AWS IoT Core
Process image and store on S3
TackleBac S3 storage
Check last image on S3 with Rekognition
Amazon Rekognition
Messages send to topic for logging
Results of Rekognition
Cleanse confirmation
Messages logged in database
TackleBac Topic
TackleBac Logging Database
Cleanse notifications and confirmations sent to topic
MQTT
IoT MQTT protocol
Notification arrives when room requires cleansing
Room
ESP32-CAM
Data transmission with AWS
Motion detection processing
Camera module processing
+
-
OmniVision 2640
Lithuim 18650 Battery and Charger
Students detected, room cleanse required
Student
Student
Student
TB+
TACKLEBAC CLEANER
ROOM : 207
REQUIRES CLEANSING
TackleBac Cleaners Watch

Figure 3 - Architectural Diagram

PROJECT PLAN



TackleBac

A proximity and cleanse detection system

October

Think: *Come up with a suitable project, that will tackle a real world situation.*

Plan: *Create a draft plan, of steps required to complete the project including a component list to meet that requirements.*

Research: *Investigate different methods used within the industry that will allow the project to be created.*

Deliverable:

30th Oct*- Project proposal .*

November

Research and Test: *Obtain more knowledge on methods, frameworks and services required.*

Deliverable: *Run examples to become familiar with the selected operations and requirements.*

Order additional components, not originally planned for

December

Work: *Use research to configure the ESP32-CAM module with IR sensor, camera module and Open CV.*

Deliverable: *Local example running with Camera and IR passing data to OpenCV, printing the result in terminal.*

January

Work: *Return to the Cloud infrastructure to configure its communication protocols and database.*

Create and practice a demonstration of work to date.

Deliverable:

11th Jan- *Demonstration of work done to date*

Feburary

Work: *Continue with Cloud infrastructure if required to secure communication and data transmission with ESP32-CAM.*

Start React Native app, AWS Amplify / Expo to provide a UI to users.

Start First draft Poster, Video plan and Report.

March

Work: *Setup mail database for student proximity breaches*

Continue work on app features.

Deliverable:

19th Mar - *First draft submissions for Poster, Video, and Report*

April

Work: *Finish app and cloud tweaks. Check project file structure, comments and project as a whole.*

Test in a real-world environment to gather data to determine correct operation

Deliverable:

30th April- Deliverable: *Final video uploaded to Moodle and Youtube.*

Final Poster and Report PDF submission

May

Work: *Take test results gathered and make any final adjustments.*

Practice demonstration and presentation

Deliverable:

May 7th - *Final project delivery and demonstration*

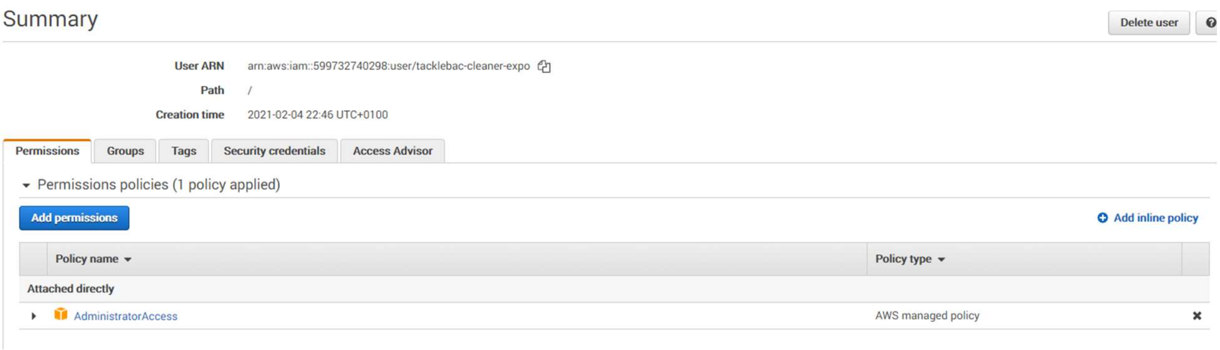
AMAZON WEB SERVICES (AWS)

Amazon Web Services (AWS) is a cloud infrastructure that hosts many services to clients around the world. They are one of the leading companies in the industry with an ever-growing catalogue of available services. I took two courses on AWS Architecture and Serverless design. I was intrigued with all that was possible, I plan to complete my AWS certification in serverless design once graduated from G.M.I.T.

IDENTITY AND ACCESS MANAGEMENT (IAM)

IAM is a User management system for security purposes. It allows the user to create different users for different tasks, in the image below for prototyping a user with admin privileges was created. However, this would not be recommended for long term use. A developer would create a user just giving access to the specific task required. This prevents people from gaining full access if details were to be compromised.

USERS



User ARN arn:aws:iam::599732740298:user/tacklebac-cleaner-expo

Path /

Creation time 2021-02-04 22:46 UTC+0100

Permissions policies (1 policy applied)

Policy name	Policy type
Attached directly	
AdministratorAccess	AWS managed policy

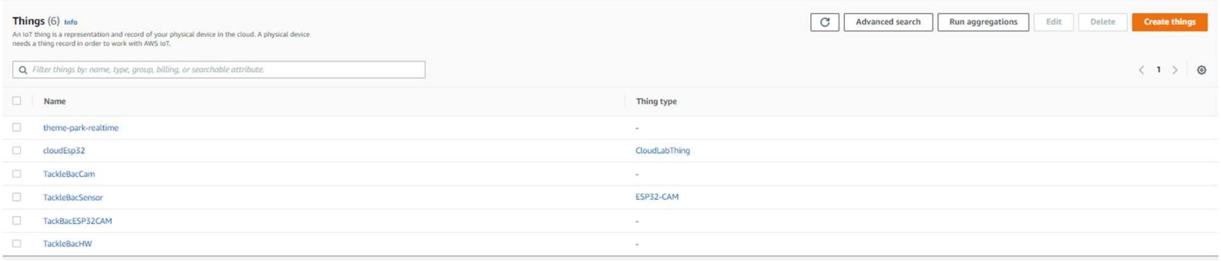
Figure 4 - IAM users

IOT CORE

AWS IoT Core is a managed cloud platform that lets connected devices - cars, light bulbs, sensor grids, and more - easily and securely interact with cloud applications and other devices.[1]

THINGS

Each device used for connecting to the IoT core is identified as a Thing. All hardware communicating for outside AWS is required to be registered.[2]

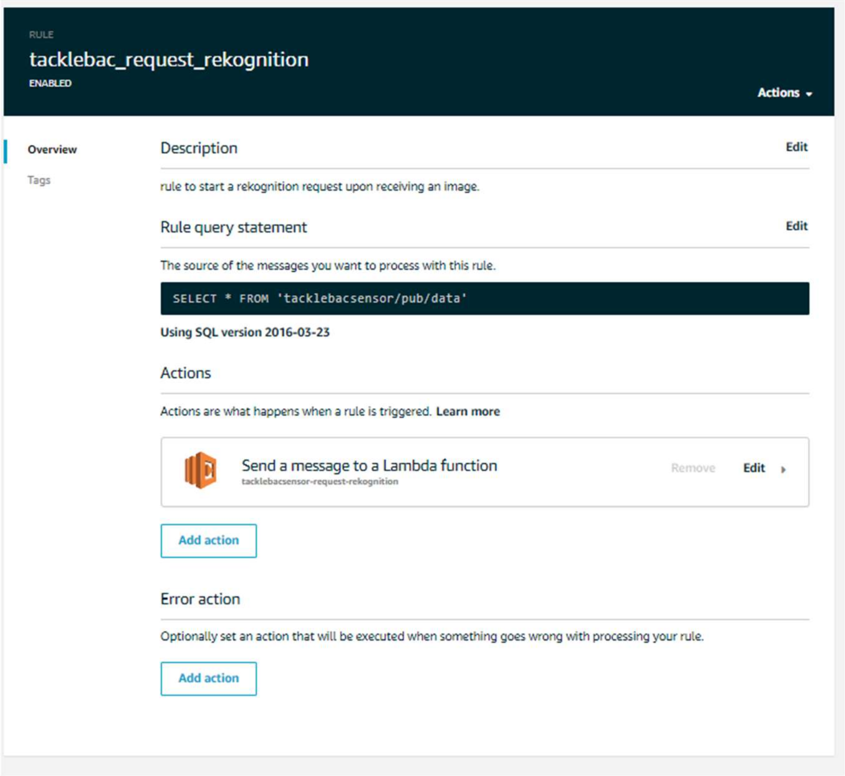

Things (6) Info
Advanced search
Run aggregations
Edit
Delete
Create things
Filter things by: name, type, group, billing, or searchable attribute.
1

An IoT thing is a representation and record of your physical device in the cloud. A physical device needs a thing record in order to work with AWS IoT.

Name	Thing type
theme-park-realtime	-
cloudEsp32	CloudLabThing
TackleBacCam	-
TackleBacSensor	ESP32-CAM
TackBacESP32CAM	-
TackleBacHW	-

Figure 5 - Amazon Web Services - Things

RULES



tacklebac_request_rekognition

Description

rule to start a rekognition request upon receiving an image.

Rule query statement

The source of the messages you want to process with this rule.

```
SELECT * FROM 'tacklebacsensor/pub/data'
```

Using SQL version 2016-03-23

Actions

Actions are what happens when a rule is triggered. Learn more

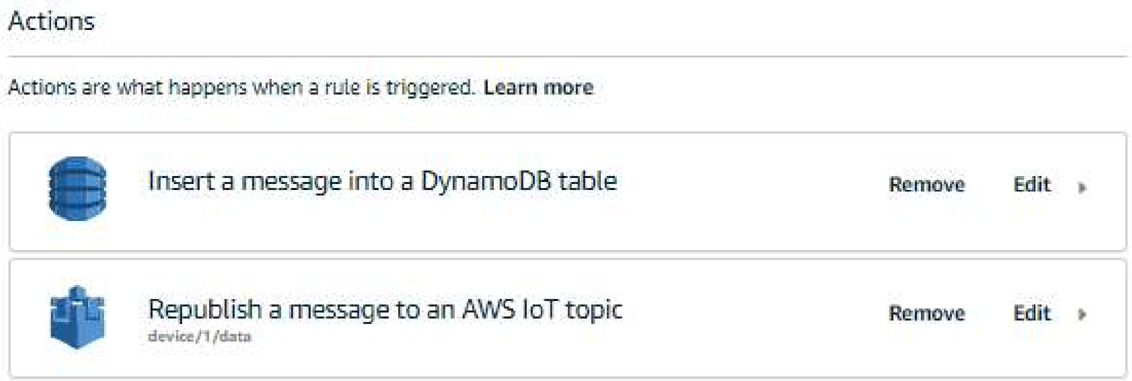
Send a message to a Lambda function
tacklebacsensor-request-rekognition

Error action

Optionally set an action that will be executed when something goes wrong with processing your rule.

In IoT, there is messaging using the MQTT protocol, but that would leave the services pretty limited if all was possible was messaging. Rules come into play as we can build rules to initiate when a certain parameter is met. The image above (Select * From '………/data) is the parameter we want the rule to use. The rule will select all from the …/data topic and send it to the lambda function.

Figure 6 - IoT rules



Actions

Actions are what happens when a rule is triggered. Learn more

Insert a message into a DynamoDB table

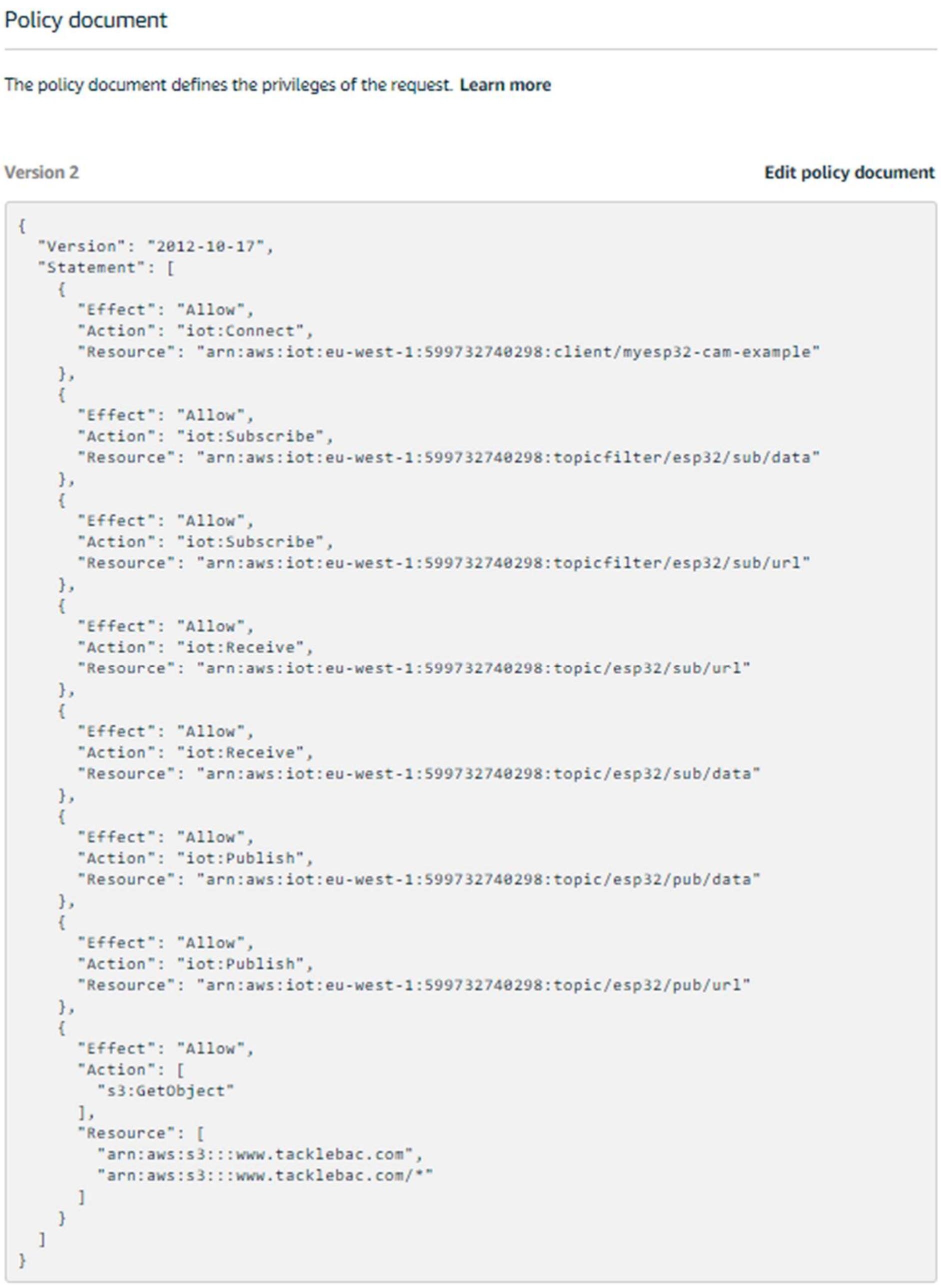
Republish a message to an AWS IoT topic
device/1/data

Figure 7 - Actions assigned to the above rule

We can see two actions one telling the rule to insert the topic's message to a database previous setup, while it also can republish the message initiated separate rules or communication.

POLICIES

A Policy is created to assign permission of blocked or granted to each Thing. Without a policy set with the configurations, you could result in thinking there is no communication with the AWS services. However, it is in a way waiting at the door to be allowed entry. The policy would act like a key allowing the communication to walk right in through the door.



Policy document

The policy document defines the privileges of the request. **Learn more**

Version 2

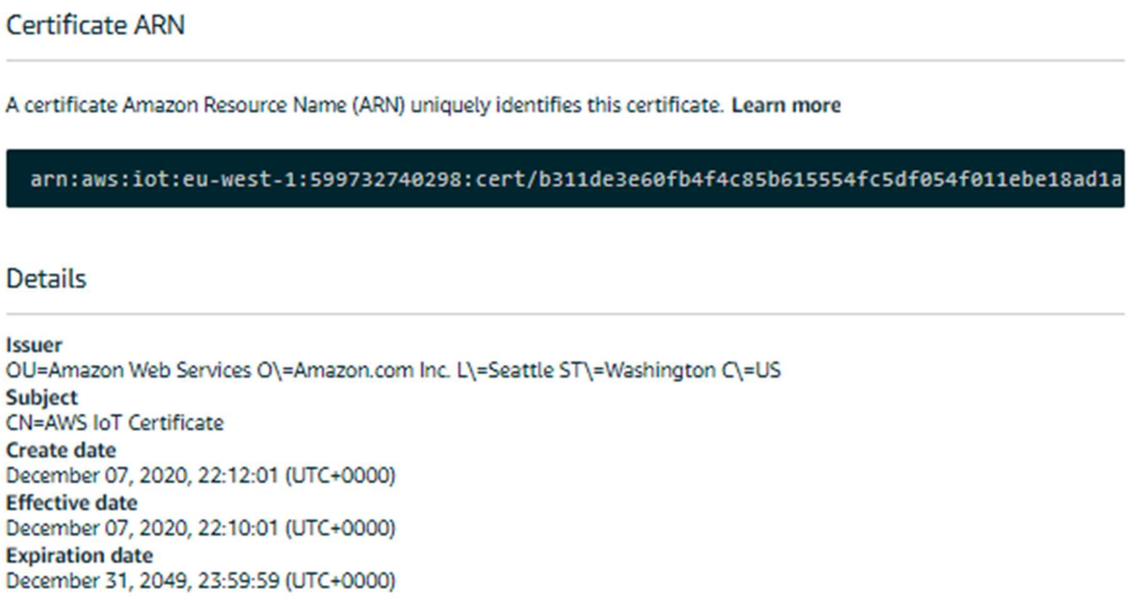
Edit policy document

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iot:Connect",
      "Resource": "arn:aws:iot:eu-west-1:599732740298:client/myesp32-cam-example"
    },
    {
      "Effect": "Allow",
      "Action": "iot:Subscribe",
      "Resource": "arn:aws:iot:eu-west-1:599732740298:topicfilter/esp32/sub/data"
    },
    {
      "Effect": "Allow",
      "Action": "iot:Subscribe",
      "Resource": "arn:aws:iot:eu-west-1:599732740298:topicfilter/esp32/sub/url"
    },
    {
      "Effect": "Allow",
      "Action": "iot:Receive",
      "Resource": "arn:aws:iot:eu-west-1:599732740298:topic/esp32/sub/url"
    },
    {
      "Effect": "Allow",
      "Action": "iot:Receive",
      "Resource": "arn:aws:iot:eu-west-1:599732740298:topic/esp32/sub/data"
    },
    {
      "Effect": "Allow",
      "Action": "iot:Publish",
      "Resource": "arn:aws:iot:eu-west-1:599732740298:topic/esp32/pub/data"
    },
    {
      "Effect": "Allow",
      "Action": "iot:Publish",
      "Resource": "arn:aws:iot:eu-west-1:599732740298:topic/esp32/pub/url"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::www.tacklebac.com",
        "arn:aws:s3:::www.tacklebac.com/*"
      ]
    }
  ]
}
```

Figure 8 - Policies used to allow communication

CERTIFICATES

Certificates are used to secure each Thing connected to IoT Core. Each thing has a set of certificates used to identify it. Amazon holds the root CA which the public and private key certificates are compared against. Secure Things have a special address called an ARN address. It is used to identify the certificates generate as an extra layer of security.



Certificate ARN

A certificate Amazon Resource Name (ARN) uniquely identifies this certificate. **Learn more**

```
arn:aws:iot:eu-west-1:599732740298:cert/b311de3e60fb4f4c85b615554fc5df054f011ebe18ad1a
```

Details

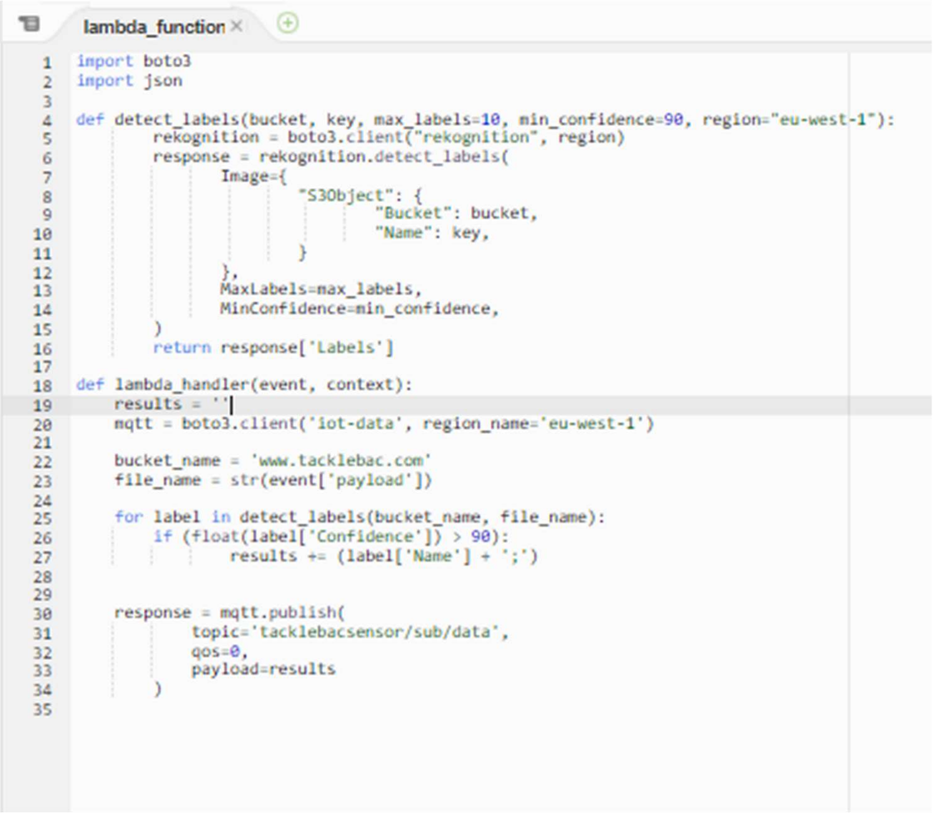
Issuer
OU=Amazon Web Services O\=Amazon.com Inc. L\=Seattle ST\=Washington C\=US
Subject
CN=AWS IoT Certificate
Create date
December 07, 2020, 22:12:01 (UTC+0000)
Effective date
December 07, 2020, 22:10:01 (UTC+0000)
Expiration date
December 31, 2049, 23:59:59 (UTC+0000)

Figure 9 - Unique ARN address for Thing

LAMBDA FUNCTIONS

Lambda functions allow the project to run serverless. The code required only runs when required therefore eliminating the need for constant servers running enduring high costs.

Below is a lambda function coded in Python although, other languages are available. As mentioned before we can initiate lambda functions using a deterministic setup in the rules.

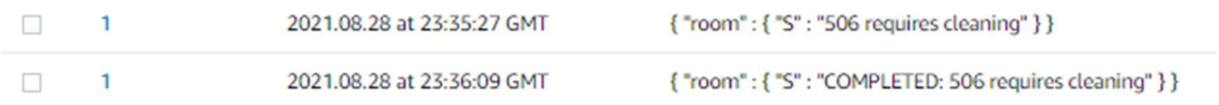


```python
import boto3
import json

def detect_labels(bucket, key, max_labels=10, min_confidence=90, region="eu-west-1"):
        rekognition = boto3.client("rekognition", region)
        response = rekognition.detect_labels(
                Image={
                        "S3Object": {
                                "Bucket": bucket,
                                "Name": key,
                        }
                },
                MaxLabels=max_labels,
                MinConfidence=min_confidence,
        )
        return response['Labels']

def lambda_handler(event, context):
    results = ''
    mqtt = boto3.client('iot-data', region_name='eu-west-1')

    bucket_name = 'www.tacklebac.com'
    file_name = str(event['payload'])

    for label in detect_labels(bucket_name, file_name):
        if (float(label['Confidence']) > 90):
                results += (label['Name'] + ';')


    response = mqtt.publish(
            topic='tacklebacsensor/sub/data',
            qos=0,
            payload=results
        )
```

Figure 10 - Lambda function used to detect objects

DYNAMODB

DynamoDB is AWS answer to a NO SQL database. It is very easily incorporated with other AWS services eliminating the need for external database hosts in the cloud.



□	1	2021.08.28 at 23:35:27 GMT	{ "room" : { "S" : "506 requires cleaning" } }
□	1	2021.08.28 at 23:36:09 GMT	{ "room" : { "S" : "COMPLETED: 506 requires cleaning" } }

Figure 11 - TackleBac database entries

REKOGNITION

Amazons object recognition service calls AWS Rekognition, it comes with 100,000 checks per month for free, which makes it great for use with each sensor. It has many options available to create custom labels based on your needs, however, identifying a human is standard in today's computer vision.

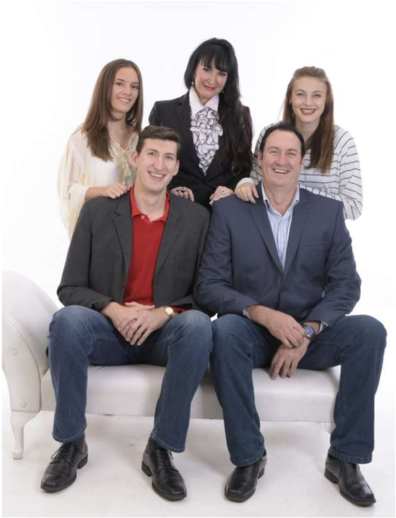
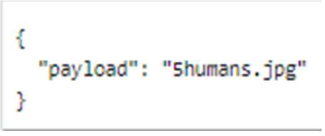
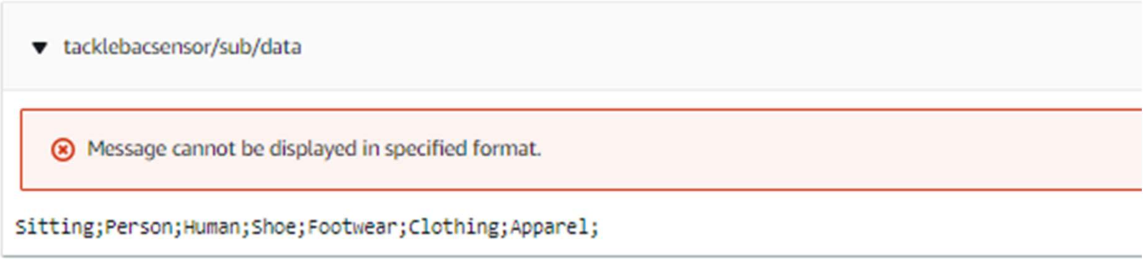


Figure 12 - 5humans image used in the example



```
{
  "payload": "5humans.jpg"
}
```

Figure 13 - message contains a payload of the image just added to S3



▼ tacklebacsensor/sub/data

Message cannot be displayed in specified format.

```
Sitting;Person;Human;Shoe;Footwear;Clothing;Apparel;
```

Figure 14 - Result from after passing image to Rekognition

S3 STORAGE

S3 Storage is the equivalent of a hard drive on a local machine, it allows storage of data or installations of full operating systems to run instances of Linux for example.

TACKLEBAC SENSOR FOR LOCATIONS

The TackleBac sensor plays a key role in the project by detecting motion present in a location, capturing an image, sending the image to AWS IoT Core as a message, then via IoT rules set and lambda functions created making use of Amazon Rekognition, it can confirm the motion was by a human. Once the human activity is confirmed a message is published to the TackleBac topic which is passed to a TackleBac cleaner as a cleanse required location notification. The device is low powered and low cost while maintaining adequate abilities to carry out the required operations for the project.

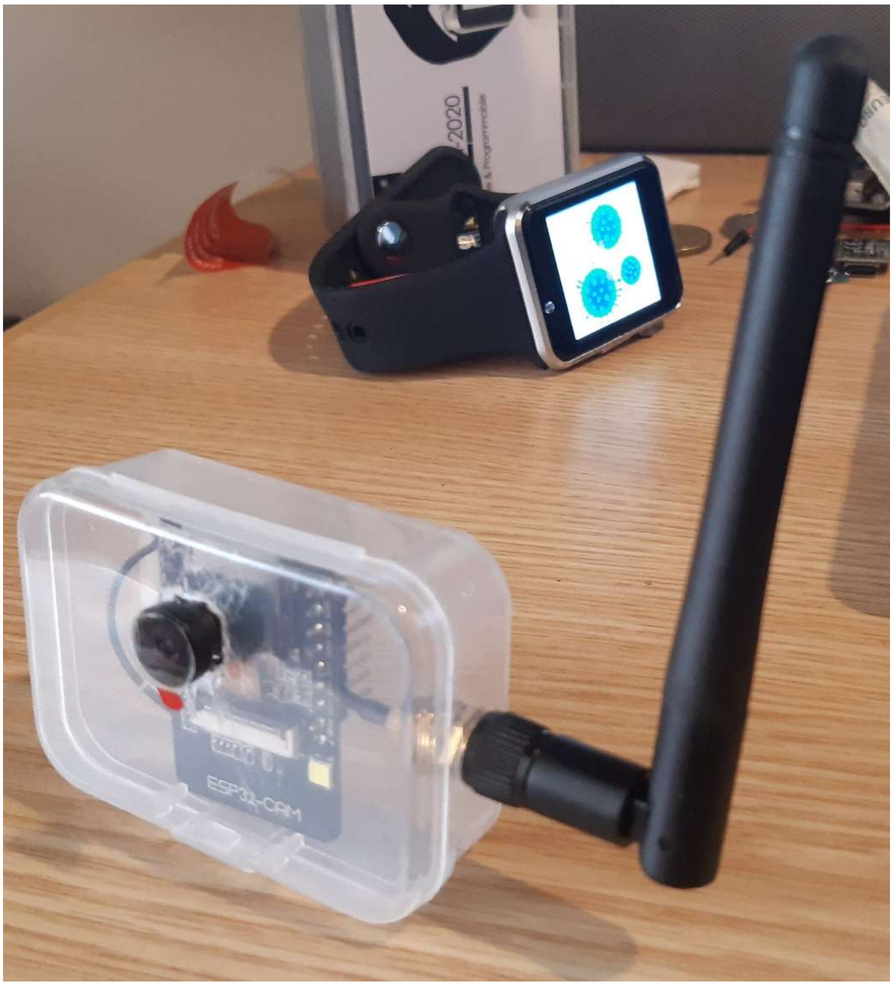


Figure 15 - TackleBac sensor (front), TackleBac cleaners watch (background)

ESP32-CAM

ESP32-CAM is a low-cost ESP32-based development board with an onboard camera connector using the Serial Camera Control Bus. It has a low profile which makes it ideal for IoT projects. The board features WiFi with an external antenna connection, Bluetooth, a dual-core 32-bit LX6 CPU (ESP32-S). It adopts 7-stage pipeline architecture, on-chip sensor, Hall sensor, temperature sensor and so on, and its main frequency adjustment ranges from 80MHz to 240MHz. [3]

OMNI VISION 2640 IMAGE SENSOR

The included camera is an OmniVision 2640, which is a 2MP camera. A higher quality version at 5MP with a wide-angle is available, although the pinout has a different layout, further configuration to the camera pins of the ESP32-CAM header is required for stable operation.

OmniVision Technologies, Inc. has defined and deployed the Serial Camera Control Bus (SCCB), a 3-wire serial bus, for control of most of the functions in OmniVision's family of CAMERACHIPSTM. In reduced pin package parts, the SCCB operates in a modified 2-wire serial mode. [4] The protocol is an I2C clone, which ignores the 9th bit which is an ACK bit of I2C.

FT232RL FUTURE TECHNOLOGY DEVICES INTERNATIONAL (FTDI) SERIAL PROGRAMMER

The ESP32-CAM unlike other variations of ESP32 boards has not had a built-in serial programmer. A USB to TTL serial adapter is required to update the device's software. Purchased was an FT232RL FTDI serial programmer. Again this is a cost-effective piece of hardware for the TackleBac project coming in at only €1.75.

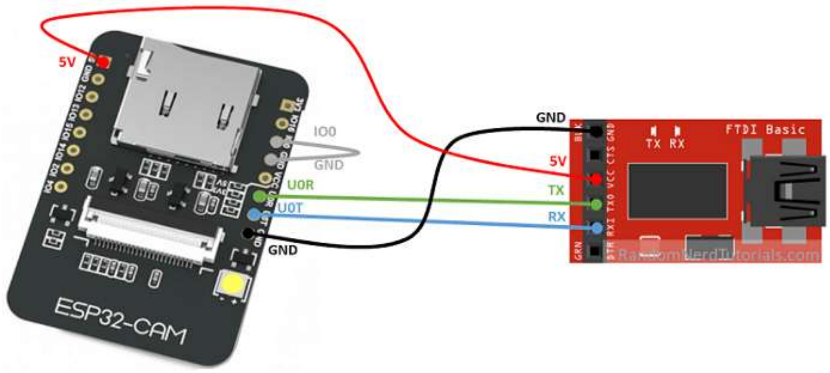


Figure 16 – A wiring diagram for connecting the FTDI to ESP32-CAM from www.randomnerdstutorial.com

POWER DISTRIBUTION

The TackleBac sensor requires constant power between 3.3volts to 5volts for operation. This solely depends on the application you are running. With the case of TackleBac, constant WiFi connectivity is required, with also frequent use of the image sensor. A micro USB cable is sufficient for powering the device. But what happens if the power is disconnected?

LITHIUM BATTERY CHARGER

To prevent data loss if there was an interruption to the mains power supply, a rechargeable lithium battery is connected. In the instance where mains power cannot be provided. The device will identify the power being removed/no present, and switch to battery power. A timestamped message informing power status will then be published to the TackleBac log table within DynamoDB. For this purpose, a €1 lithium battery charger/power distribution board has been installed. This has a TP4056 IC which is a lithium battery charging processor and for safety a DW01A IC which is used for lithium battery circuit protection. The complete circuit is produced in high volumes with a generic model number of 03962a. This allows the device to be powered with mains connected, while also allowing the device to continue operating on battery power if required.

LITHIUM BATTERY

The battery chosen to provide backup power is the lithium-ion 18650. It is like an AA battery in size. Although, unlike the 1.5v AA battery the 18650 boasts a 3.7v 2.6Ah rating. The life expectancy of the battery is between 300-500 cycles. [5]This does not seem very high, yet the TackleBac device will only be operating on battery power in severe conditions that result in the loss of power, with this in mind this battery is a perfect solution for the project.

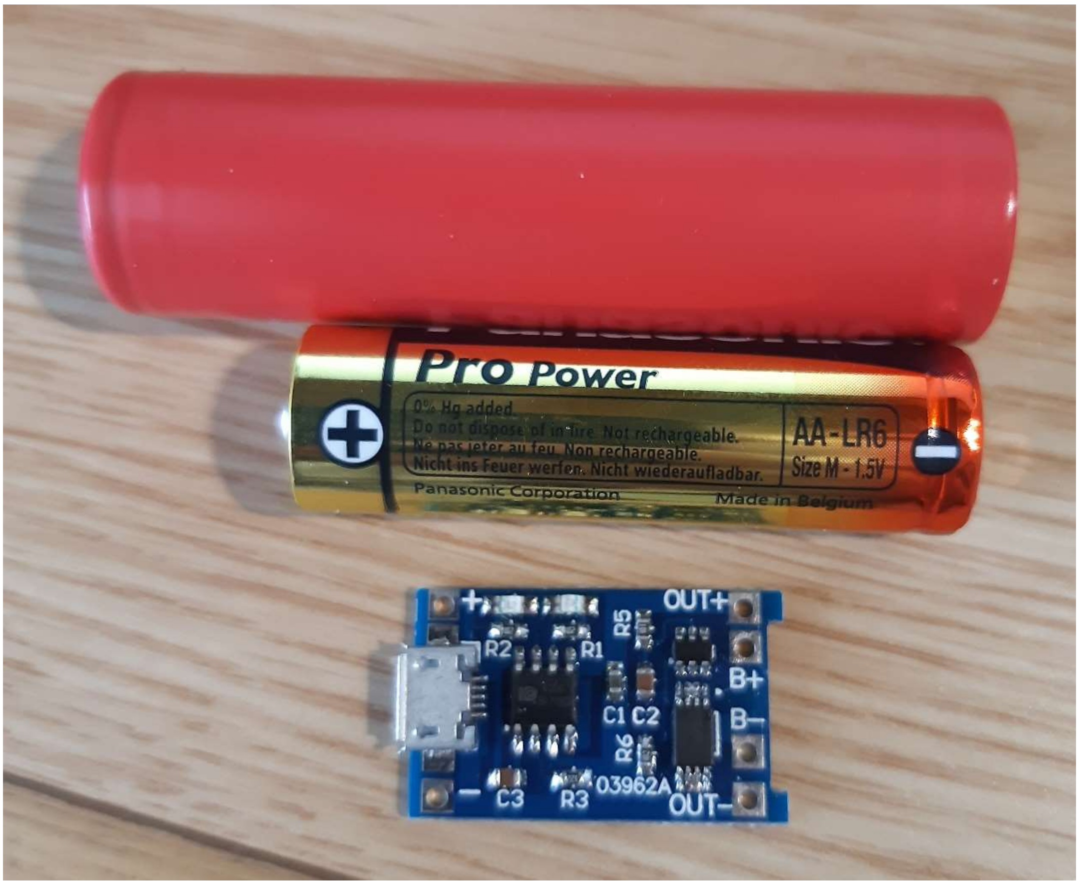

Pro Power
0% Hg added.
Do not dispose of in fire. Not rechargeable.
Ne pas jeter au feu. Non rechargeable.
Nicht ins Feuer werfen. Nicht wiederaufladbar.
AA-LR6
Size M - 1.5V
Panasonic Corporation
Made in Belgium
OUT+
B+
B-
OUT-
R1
R2
R3
R5
R6
C1
C2
C3
03962A

Figure 17 - 3.7v 18650 Lithium battery (Top), 1.5v AA Battery (Middle) for size comparison and a lithium battery charger (Bottom)

DETECTING POWER CUT / BATTERY POWER

Although not yet implemented into the code of the project, research has been carried out to provide a working solution to inform the system that a TackleBac sensor has switched to battery power. Using a voltage divider with two 10k resistors across the battery connections of the ESP32-CAM, we can tap out between the resistors and connect a wire to one of the IO pins on the ESP32-CAM. A reading of cycles will be counted and stored as a threshold variable. When the TackleBac sensor has a loss of power, the battery will be in use therefore discharging. The discharge will slowly reduce the counted cycles, therefore, operating below the threshold so a power status message will then be published to the cloud. It is a simple yet effective method of detecting the difference between mains and battery power.

CODING THE SENSOR

Coding the sensor required elements and

AWS CONNECT

A secure connection is required to safely transmit information between sensor, cleaner and AWS Cloud infrastructure. To ensure and provide the required measures, certificates were used for authentication and verification from point to point. Having no previous knowledge of certificates I spent some time learning about their uses. Fortunately, AWS now has incorporated a certificate generator that uses their root ca certificate to confirm the device has the permissions for access.

IOT CERTIFICATES, ENDPOINT

To obtain a secure connection between the TackleBac devices and AWS IoT core, certificate-based authentication is required. A public key, private key and root ca certificate is required and can be obtained from AWS upon initialisation of a thing. They are each set using the functions from within the WiFiClientSecure header. The endpoint is a unique URL designated per thing, to allow communication directly and securely when combined with the provided certificates. It is very important to not upload your certificates to public spaces such as GitHub. Hackers have been known to scan repositories for certificates to gain access to peoples AWS consoles, set up cloud servers for mining, where huge costs have occurred to the account holder.

MQTT

MQTT is an OASIS standard messaging protocol for the Internet of Things (IoT). It is designed as an extremely lightweight publish/subscribe messaging transport that is ideal for connecting remote devices with a small code footprint and minimal network bandwidth. [6] Topics must be subscribed to receive messages published to it. We can think of these as television channels, if you are not subscribed to the channel you will be unable to view the program.

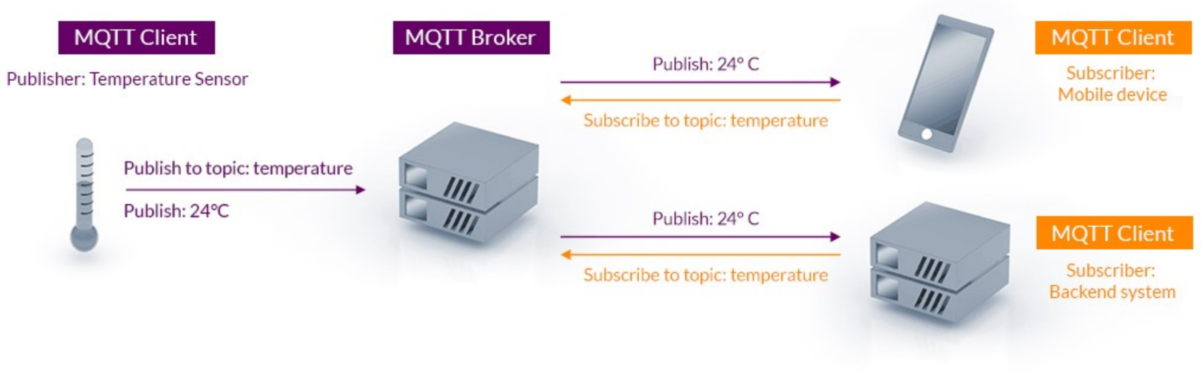


Figure 18 – MQTT protocol diagram from mqtt.org

As the diagram above from mqtt.org, TackleBac sensor and TackleBac cleaners watch both act as independent MQTT clients, AWS IoT Core is the acting MQTT broker. Each device subscribes to the TackleBac topic to receive messages ie: the cleaner watch subscribes to receive cleansing notifications. When sending messages, they are published to a topic. It is not necessary to be subscribed to the topic messages that are being published.

MOTION DETECTION

"Motion detection is the process of detecting a change in the position of an object relative to its surroundings or a change in the surroundings relative to an object"[7]

The method being used is known as naive motion detection, as a specific object is not detected a change in the captured image. The motion detection element of the project is built making use of the eloquent library, which is focused on computer vision and machine learning algorithms. [8], [9]

A frame is captured at start-up and stored as the current frame, it is checked against the previous frame which has a NULL value; therefore, no motion has been detected. The image is stored as the previous frame and the motion detection repeats. On the next capture, we have a reference frame now stored as the previous frame, each pixel within the capture has the potential to change. If the value of the previous frame's pixel does not match the value of the current frames' pixel of the same location, this will be counted as a change. Once all pixels have been compared, If the change amount is above a set threshold then the current frame is encoded as base64 and published to the subscribed topic. A rule is then set in place to send the image to AWS Rekognition which returns the objects found in the picture. A response including the label Human is required to publish a cleanse required notification.[10]

Checking each pixel in an image is time-consuming and with higher resolution more pixels are present. To overcome this, downsampling is performed on the captured image. The image is split into blocks with the value of each block being compared to the last frames. This reduced the processing time of the operation.

TACKLEBAC DEVICE FOR CLEANERS

Cleaners play a vital role when it comes to pandemics, providing adequate sanitation of used locations such as lectures, halls, labs, rooms etc. Limiting the duration spent in potentially contaminated areas reduces the risk of contracting the bacteria's and/or viruses for cleaning employees. For the safety of all cleaning employees, and to reduce costs of implementation a low-cost programmable smartwatch is used. This also protects the employee further as they will not be required to carry their mobile device with them in high-risk areas, as their device could harbour the bacteria's and/or viruses potentially bringing them home creating unnecessary contamination and risks.

When a human has been confirmed to have occupied a location. A message is published to an AWS IoT Core topic. This message informs of the location that requires cleansing. This displays the location that requires cleansing to the cleaner and deletes the notification once the cleaner has marked cleansing of the location has been complete.

LILYGO T-WATCH-2020

The Lilygo T-Watch-2020 V1 is a low costing programmable smartwatch for only €28, while also being an ESP32 System On Chip (SOC) embedded device. This makes it very useful for cleaners and developers of code for the device. It contains many features expected from a smartwatch such as IPS display, vibration motor, speaker, infrared signal sensor, reliable power distribution, triaxial accelerometer, RTC Clock Module, capacitive touch screen, low costing replaceable Lithium-Ion battery, and most importantly being an embedded ESP32 SoC gives access to built-in Wi-Fi and Bluetooth. [11]-[13]

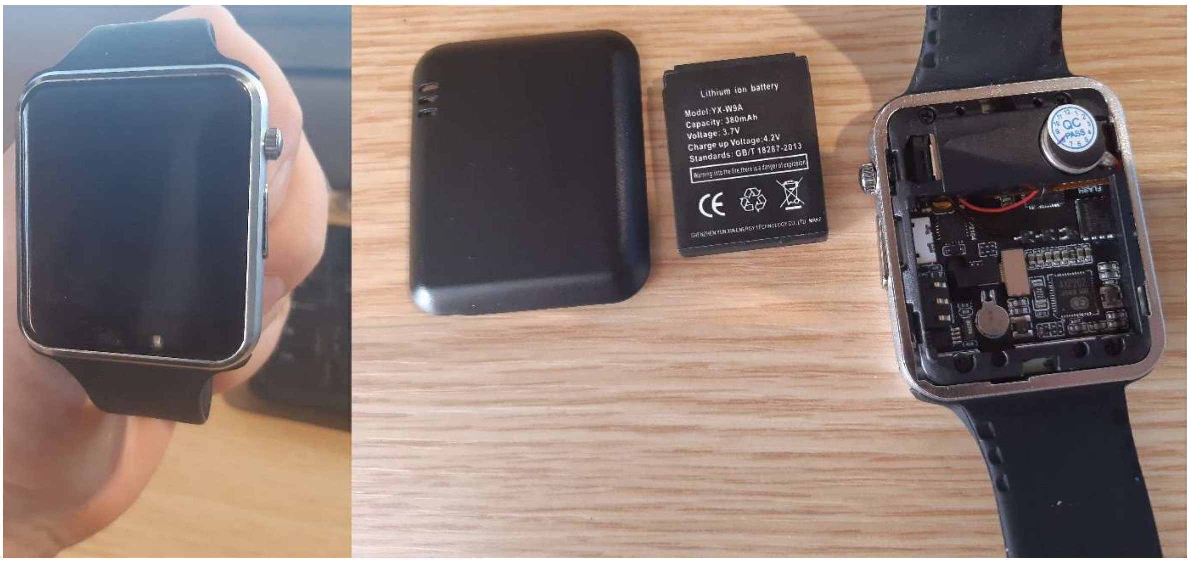


Figure 19 - Lilygo T-Watch-2020 Front and Back with the cover removed to see the internals

LIGHT AND VERSATILE GRAPHIC LIBRARY

LVGL (Light and Versatile Graphics Library) is a free and open-source graphics library providing everything you need to create an embedded GUI with easy-to-use graphical elements, beautiful visual effects, and a low memory footprint.[14], [15]

The library was a learning curve although documentation is available with extensive explanations and demos. The end goal allowed a GUI to be provided for the cleaner wearing the smartwatch, notifying them of rooms that required cleaning while allowing them to selected the room that was cleansed which was then removed from the list.

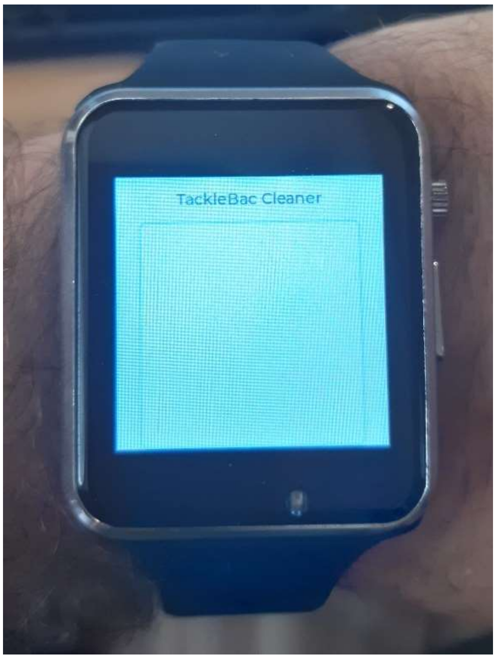
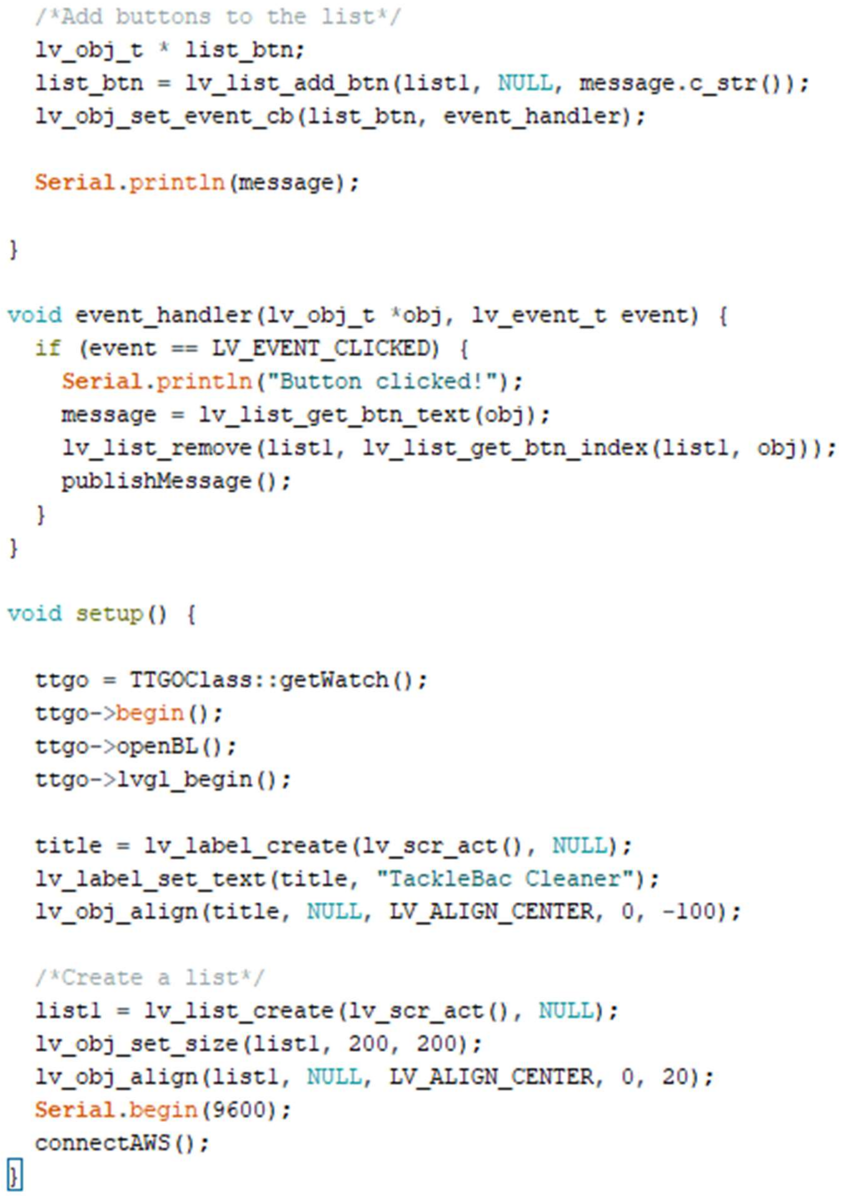


Figure 20 - TackleBac Cleaner app on startup

CODING THE WATCH

The core of the watch code for connecting to AWS IoT core is the same as the TackleBac sensor mentioned previously. This allows the watch to subscribe securely to the TackleBac topic, while also publishing a message to the topic once a room has been cleansed.



```
  /*Add buttons to the list*/
  lv_obj_t * list_btn;
  list_btn = lv_list_add_btn(list1, NULL, message.c_str());
  lv_obj_set_event_cb(list_btn, event_handler);

  Serial.println(message);

}

void event_handler(lv_obj_t *obj, lv_event_t event) {
  if (event == LV_EVENT_CLICKED) {
    Serial.println("Button clicked!");
    message = lv_list_get_btn_text(obj);
    lv_list_remove(list1, lv_list_get_btn_index(list1, obj));
    publishMessage();
  }
}

void setup() {

  ttgo = TTGOClass::getWatch();
  ttgo->begin();
  ttgo->openBL();
  ttgo->lvgl_begin();

  title = lv_label_create(lv_scr_act(), NULL);
  lv_label_set_text(title, "TackleBac Cleaner");
  lv_obj_align(title, NULL, LV_ALIGN_CENTER, 0, -100);

  /*Create a list*/
  list1 = lv_list_create(lv_scr_act(), NULL);
  lv_obj_set_size(list1, 200, 200);
  lv_obj_align(list1, NULL, LV_ALIGN_CENTER, 0, 20);
  Serial.begin(9600);
  connectAWS();
}
```

Using LGVL a list was created at start-up with the ability to store buttons. When a location notification was received a button was created and added to the list, with the location being the button text. This button was added to the list and as more locations are added, the list becomes scrollable automatically.

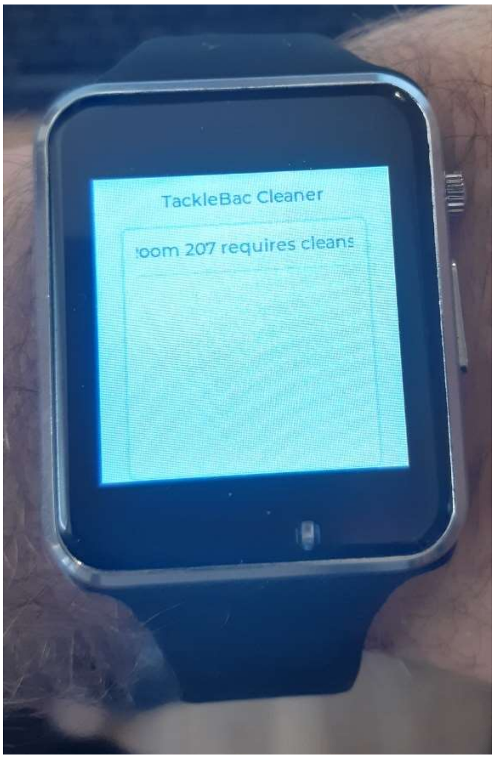


Figure 21 - A Notification of a room requiring cleaning

When a button is pressed on the list an event handler is called which checks the button text object and uses it to obtain the button index. The button index is then used to delete the correct button from the list. At the same time as confirming the cleansing and deleting the notification, an entry is made to the DynamoDB table for logging purposes.

CONCLUSION

To conclude the report, we have found out about the Why, What and How of the project TackleBac. Learning the use of the low powered and low costing device within the project has advantages while disadvantages to its use, which are understandable due to its designed architecture. Using the serverless clouds design that was implemented allowed the running cost to be minimal, while almost eliminating any maintenance required to the system on-site. Using the concept of the project together with the cloud implementation functions and methods, the project could be upscaled to larger devices with the capabilities of processing all information locally if internet connectivity was limited. Having completed the project has thought that underestimating time to complete aspects of a project is common in engineering, where research can sometimes be more time consuming than originally planned.

The overall project contains many aspects and technologies from the course, which have been enhanced and developed throughout its duration. Modules completed within the course have been vital in the research and development of the TackleBac project. It has provided great insight into cloud technologies such as Amazon Web Services which will be a great addition to include in the expanding skillset.

PROBLEM-SOLVING

TackleBac has created problems during the project. Although, solutions were found to eliminate the issues allowing progress to continue. As with many projects, the original plans and architecture may not match or be remotely the same as the finished product. Some projects in the industry can take years to complete, and a complex project like TackleBac with many elements opens the doors for such problems to occur.

MICROCONTROLLER

The Microcontroller the project used was the ESP32-CAM by AI Thinker. It is Low Costing and has a Low-Performance rating which fundamentally brought along some issues to the project design. Despite its built-in camera being a 2Mega Pixels, the resolution and quality of the images were below par. This itself caused issues in image processing. This is due to many factories' manufacturing the OmniVision 2640 Image sensor module, so quality can vary across the board. There are various iterations of the ESP32-CAM with some performing better than others, so ordering from a reputable source is a must to obtain the highest and stable results.

IMAGE/VIDEO TRANSMISSION

As mentioned, the low performance of the microcontroller has limitations. Transmitting images are a key part of the TackleBac project, multiple routes were considered to provide a cost-effective and simple setup process to be enabled. With the concept of decision transmitting on a frame by frame basis for analytics.

REFERENCES

[1] “AWS IoT.” https://eu-west-1.console.aws.amazon.com/iot/home?region=eu-west-1#/home (accessed Jun. 29, 2021).

[2] “AWS IoT - Manage - Things.” https://eu-west-1.console.aws.amazon.com/iot/home?region=eu-west-1#/thinghub (accessed Jun. 29, 2021).

[3] “What I Learned About ESP32-CAM and Everything You Need to Know | by Nishan Chathuranga Wickramarathna | Medium.” https://nishanc.medium.com/what-i-learned-about-esp32-cam-and-everything-you-need-to-know-12a9e520a0da (accessed Jun. 28, 2021).

[4] “APPLICATION NOTE OmniVision® OmniVision Serial Camera Control Bus (SCCB) Functional Specification Revision Number Date Revision”.

[5] “18650 Battery Life Span.” https://www.18650batterystore.com/blogs/news/18650-battery-life-span (accessed Jun. 28, 2021).

[6] “MQTT - The Standard for IoT Messaging.” https://mqtt.org/ (accessed Jun. 28, 2021).

[7] “Motion detection - Wikipedia.” https://en.wikipedia.org/wiki/Motion_detection (accessed Jun. 28, 2021).

[8] “Motion detection with ESP32 cam only (Arduino version).” https://eloquentarduino.github.io/2020/01/motion-detection-with-esp32-cam-only-arduino-version/#tocblocks-difference-threshold (accessed Jun. 28, 2021).

[9] “Easier, faster pure video ESP32 cam motion detection.” https://eloquentarduino.github.io/2020/05/easier-faster-pure-video-esp32-cam-motion-detection/ (accessed Jul. 28, 2021).

[10] “Creating Object Recognition with Espressif ESP32 | The Internet of Things on AWS – Official Blog.” https://aws.amazon.com/blogs/iot/creating-object-recognition-with-espressif-esp32/ (accessed Jun. 29, 2021).

[11] “14. T-Watch 2020 — T-Watch Docs.” https://t-watch-document-en.readthedocs.io/en/latest/introduction/product/2020.html (accessed Jun. 27, 2021).

[12] “Basics — T-Watch Docs.” https://t-watch-document-en.readthedocs.io/en/latest/basics/index.html (accessed Jun. 27, 2021).

[13] “LILYGO® TTGO T-Watch-2020 ESP32 Main Chip 1.54 Inch Touch Display Programmable Wearable Environmental Interaction (1) - T-WATCH Series - Shenzhen Xin Yuan Electronic Technology Co., Ltd.” http://www.lilygo.cn/prod_view.aspx?TypeId=50053&Id=1290&FId=t3:50053:3 (accessed Jun. 27, 2021).

[14] “Welcome to the documentation of LVGL! — LVGL documentation.” https://docs.lvgl.io/latest/en/html/index.html (accessed Jun. 27, 2021).

[15] “Chart (lv_chart) — LVGL documentation.” https://docs.lvgl.io/latest/en/html/widgets/chart.html (accessed Jun. 27, 2021).

PROTOTYPES

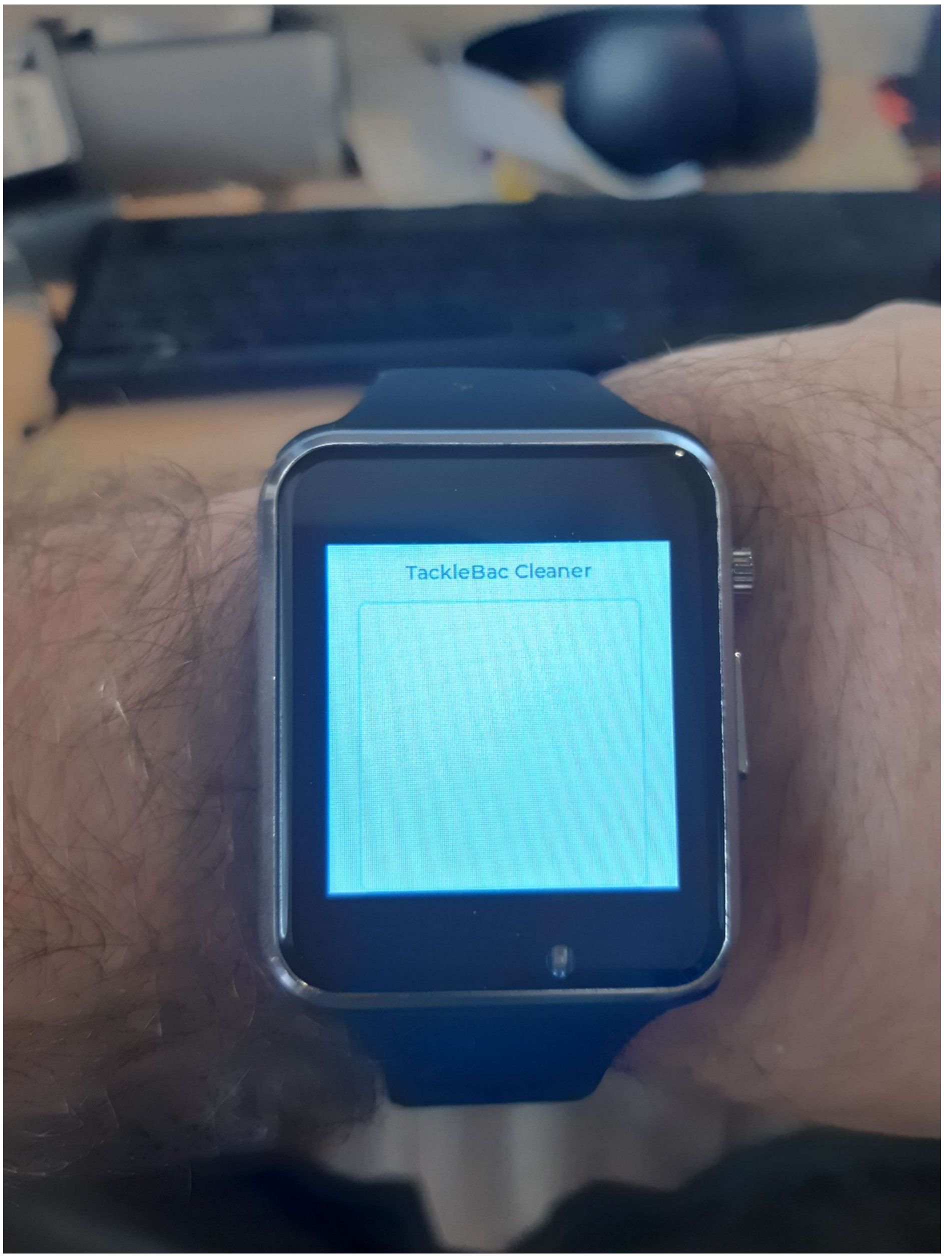


Figure 22 - TackleBac cleaner's watch

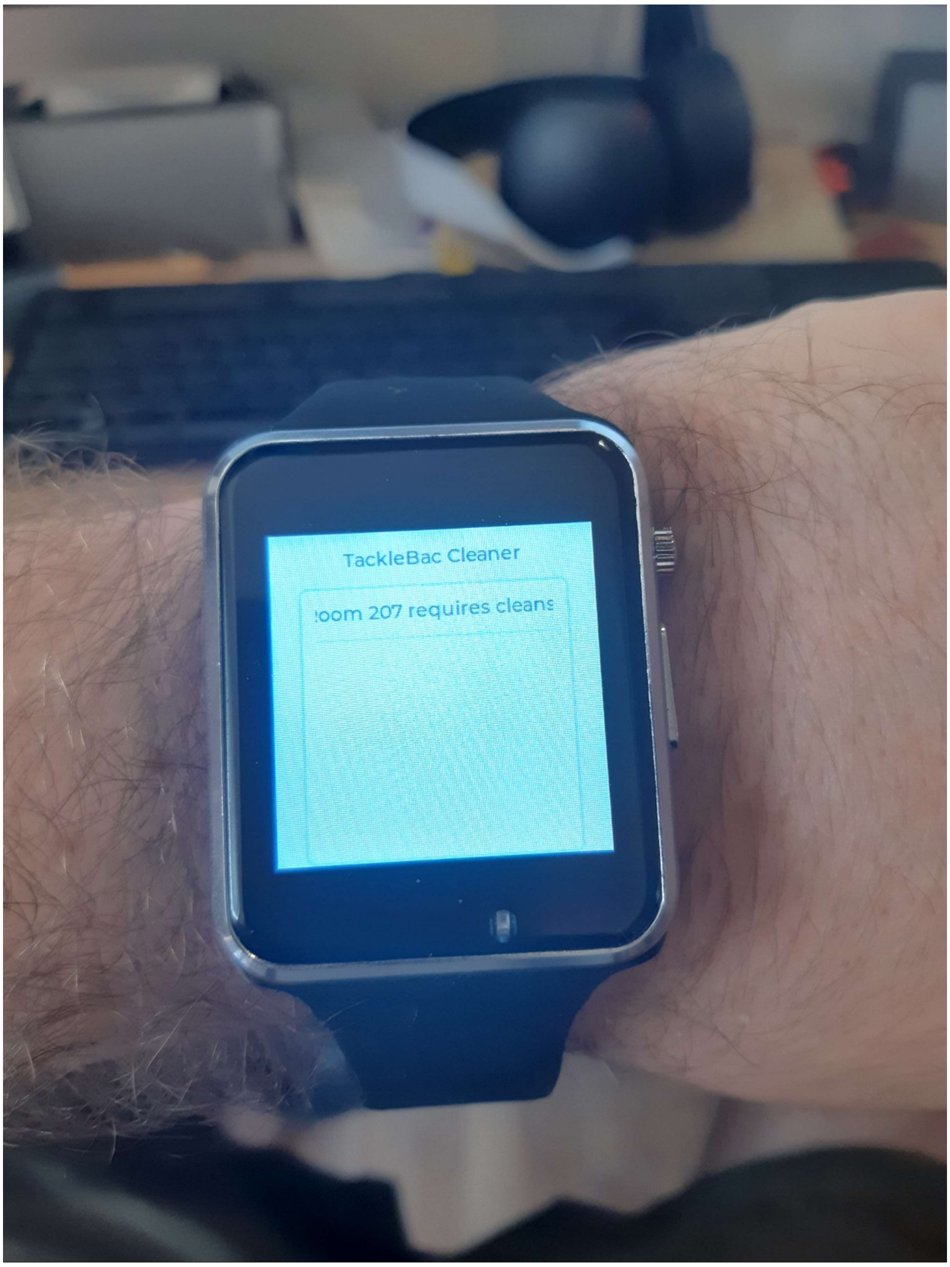


Figure 23 - Cleanse notification on the watch

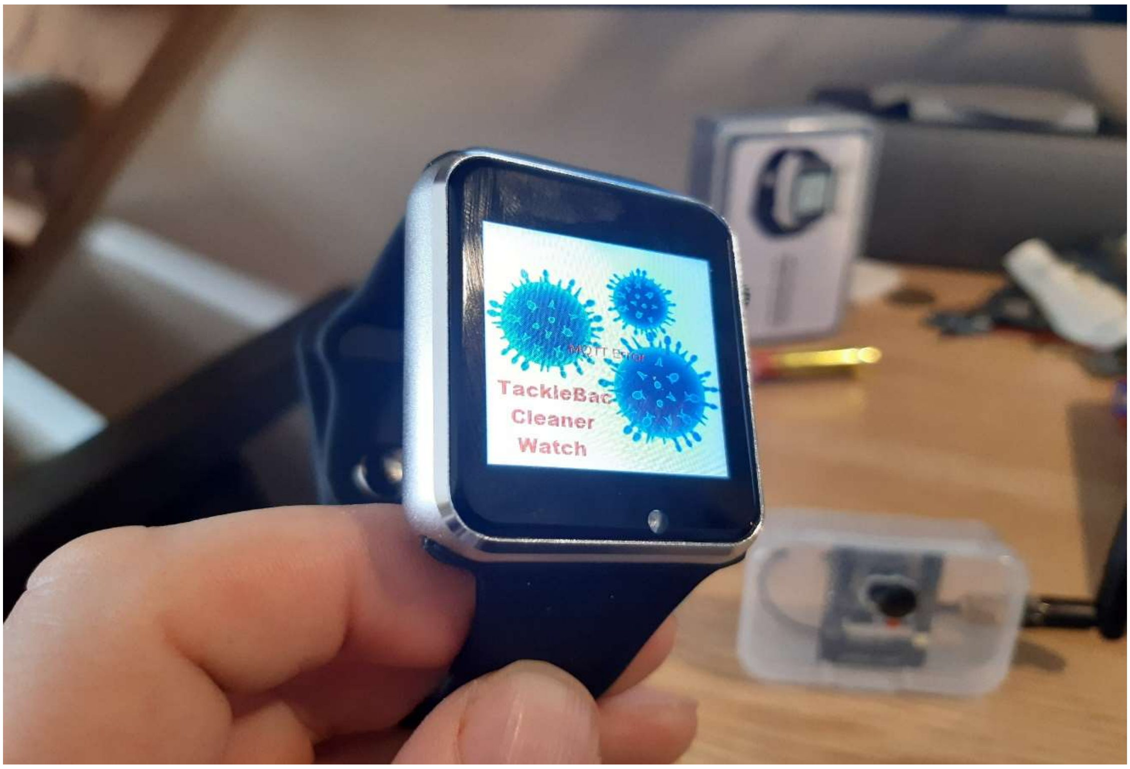


Figure 24 - A prototype splash screen for cleaners watch

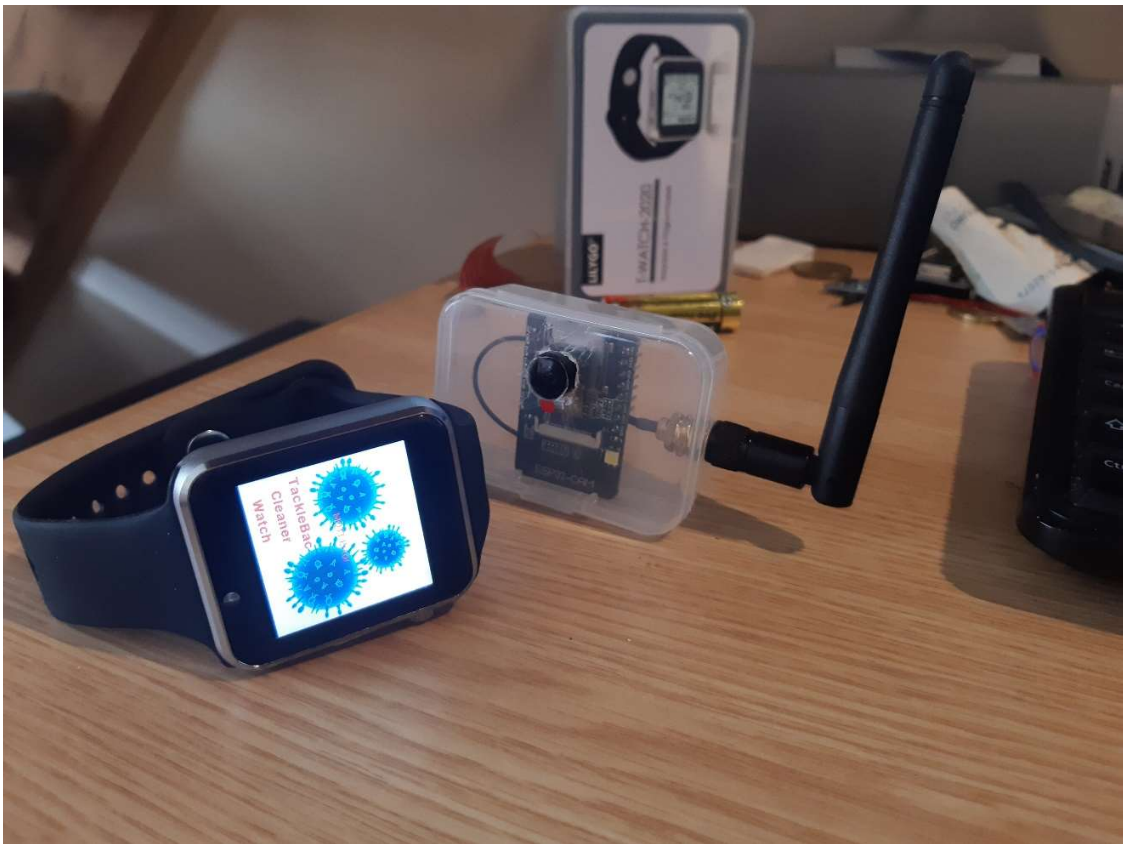


Figure 25 - TackleBac package V1

APPENDIX

For more information and updates, I have developed a webpage for the project which I plan to use as the project is expanded and developed further.

www.tacklebac.com

Repositories and files can also be found on www.github.com/TackleBac