Ollscoil
Teicneolaíochta
an Atlantaigh

Atlantic
Technological
University

Internet of Things Project

ATU Attendance

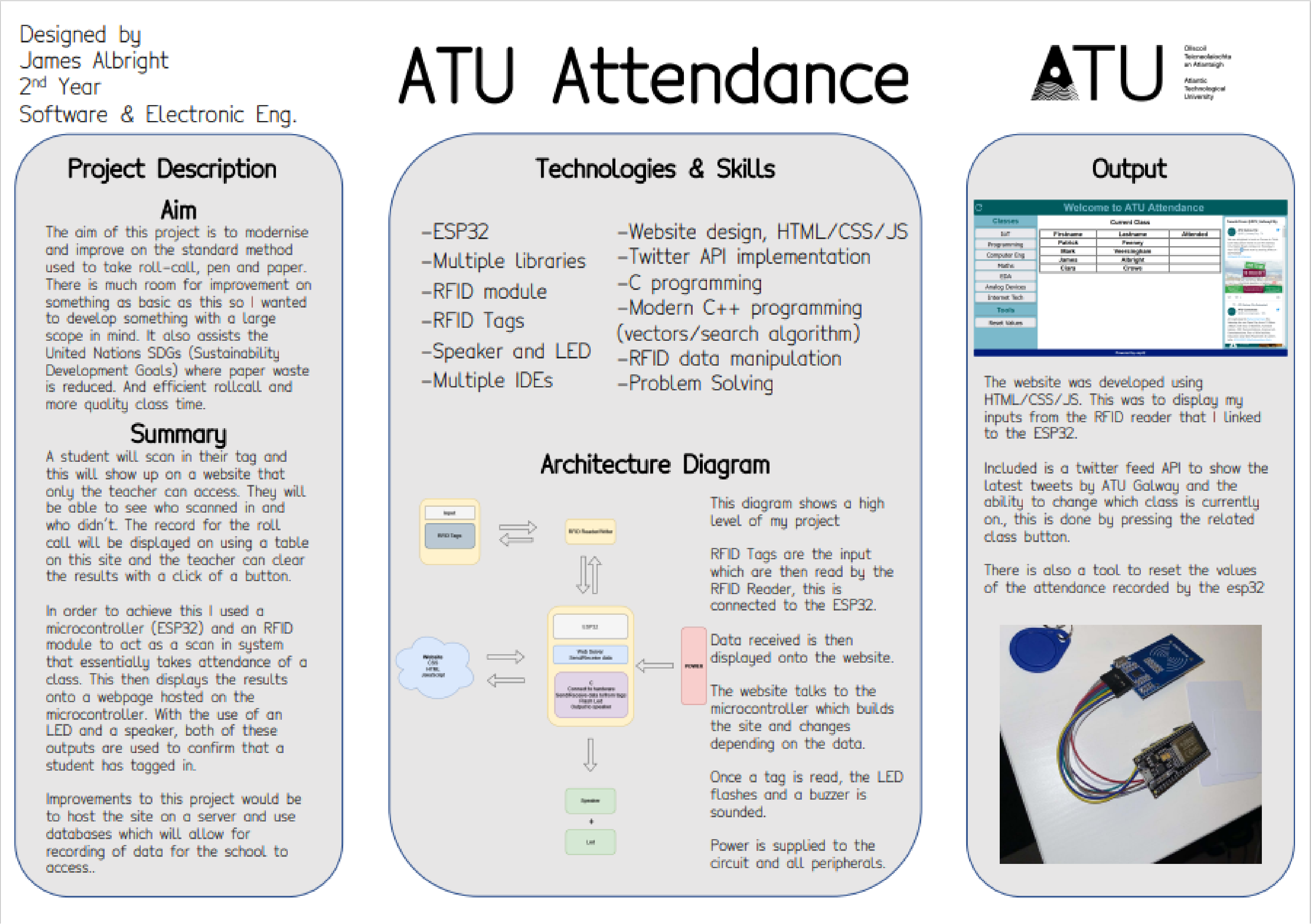
James Albright

Bachelor of Software & Electronic Engineering

Atlantic Technical University

2022/2023

Poster



Designed by
James Albright
2nd Year
Software & Electronic Eng.

ATU Attendance

ATU
Ollscoil Teicneolaíochta an Atlantaigh
Atlantic Technological University

Project Description

Aim

The aim of this project is to modernise and improve on the standard method used to take roll-call, pen and paper. There is much room for improvement on something as basic as this so I wanted to develop something with a large scope in mind. It also assists the United Nations SDGs (Sustainability Development Goals) where paper waste is reduced. And efficient rollcall and more quality class time.

Summary

A student will scan in their tag and this will show up on a website that only the teacher can access. They will be able to see who scanned in and who didn't. The record for the roll call will be displayed on using a table on this site and the teacher can clear the results with a click of a button.

In order to achieve this I used a microcontroller (ESP32) and an RFID module to act as a scan in system that essentially takes attendance of a class. This then displays the results onto a webpage hosted on the microcontroller. With the use of an LED and a speaker, both of these outputs are used to confirm that a student has tagged in.

Improvements to this project would be to host the site on a server and use databases which will allow for recording of data for the school to access..

Technologies & Skills

-ESP32
-Multiple libraries
-RFID module
-RFID Tags
-Speaker and LED
-Multiple IDEs

-Website design, HTML/CSS/JS
-Twitter API implementation
-C programming
-Modern C++ programming (vectors/search algorithm)
-RFID data manipulation
-Problem Solving

Architecture Diagram

This diagram shows a high level of my project

RFID Tags are the input which are then read by the RFID Reader, this is connected to the ESP32.

Data received is then displayed onto the website.

The website talks to the microcontroller which builds the site and changes depending on the data.

Once a tag is read, the LED flashes and a buzzer is sounded.

Power is supplied to the circuit and all peripherals.

Output

Welcome to ATU Attendance

Current Class

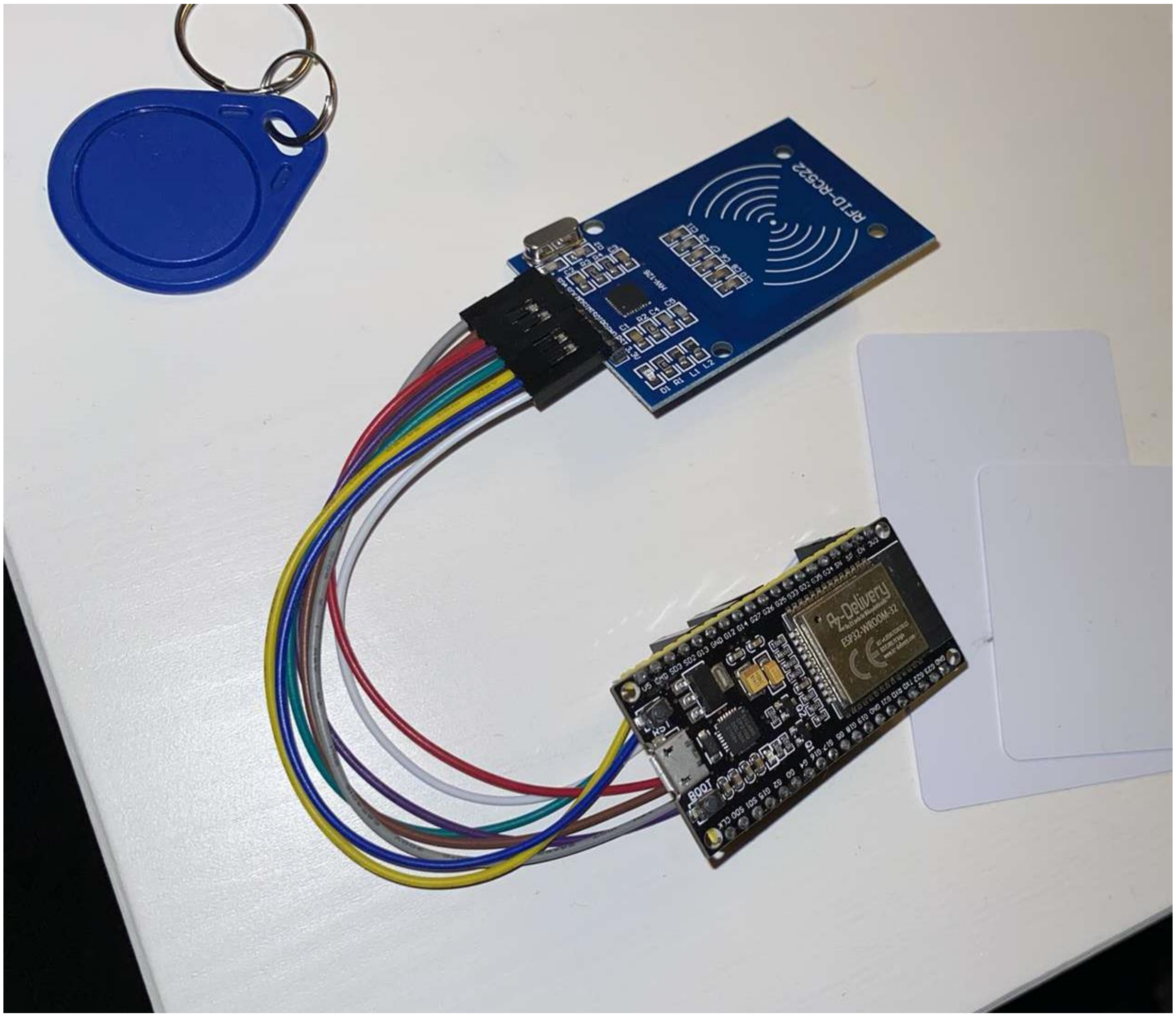
Firstname	Lastname	Attended
Patrick	Feeney	
Mark	Werrington	
James	Albright	
Clara	Crowe	

The website was developed using HTML/CSS/JS. This was to display my inputs from the RFID reader that I linked to the ESP32.

Included is a twitter feed API to show the latest tweets by ATU Galway and the ability to change which class is currently on., this is done by pressing the related class button.

There is also a tool to reset the values of the attendance recorded by the esp32

Project



Declaration

This project is presented in partial fulfilment of the requirements for the degree of Bachelor of Engineering in Software & Electronic Engineering at the Atlantic Technical University, Galway campus.

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

James Albright.

Acknowledgements

Use this section to acknowledge anyone, if you wish to, who might have helped during your project.

Michelle Lynch

Brian O’Shea

Natasha Rohan

Summary

ATU Attendance was designed to modernize a method used in so many educational institutions. I created an attendance system using RFID (Radio Field Identification) technology to record attendance of students and display it on a website to the teacher.

The scope of project is to record data and display data, using RFID and a webpage.

My project is not inherently aimed at being more sustainable and a better choice for the environment, but it still assists in the United Nations 17 goal by doing the following:

Quality Education (4): It allows the teacher to efficiently take roll and allows for more total teaching time and focused learning.

Responsible Consumption and Production (12): The project I am making is made of items that are reusable and cheap. The school wastes paper on ticking a box for every class, every day. This reduces paper waste.

I approached this project with many ideas because there are many ways to do it. I decided on having a unique identification for each user so it would be easy to track and record. It was also a new technology and I thought it would be interesting.

The methods I implemented in this project were troubleshooting for any issues I came across regarding hardware/software, C to write the bulk of the hardware code, HTML/CSS/JavaScript to develop the webpage to display data on to.

Technology used: Esp32, Buzzer, LED, Libraries, RFID Module/Tags

I accomplished an integrated system that combines software/hardware/networking. The project uses an RFID module and ESP32 to improve the efficiency of roll taking, the students scan in and their tag data is recorded and compared with an accepted list of tags and is then signed in. The result is displayed on a webpage then.

The project definitely had too large of a scope initially but was happy with how much I learned in the process, inclusion of database and a properly hosted webpage would make it possible to do the way I intended. Proof of concept.

Table of Contents

1 Introduction

This project is an attendance system which will improve a day-to-day process that every school uses, roll call. My goal was to make this task as automated as possible by making roll call done by students just swiping a tag. I decided on this idea because I see in both school and University, we still use the basic method of pen and paper. A chance at modernizing this was a good project idea to me.

To achieve this, I relied heavily on RFID Technology (Radio Frequency Identification) and the versatility of the ESP32. RFID works by transmitting electromagnetic energy which hold information (the ‘ID’) from the tag to the reader. Although on a different end of the electromagnetic spectrum, it works the same as light does to our eyes, data sent to a reader which is then processed. [1]

A basic explanation is this - a student will scan in their tag; this will show up on a website that only the teacher can access. They will be able to see who scanned in and who didn’t. After class ends the teacher clicks a button on the site and refreshes the currently shown attendance record.

There are also two other components which are for the end user’s sake, an LED and a Buzzer, these are just methods to tell them that their action has gone through and has been received.

2 Project Architecture

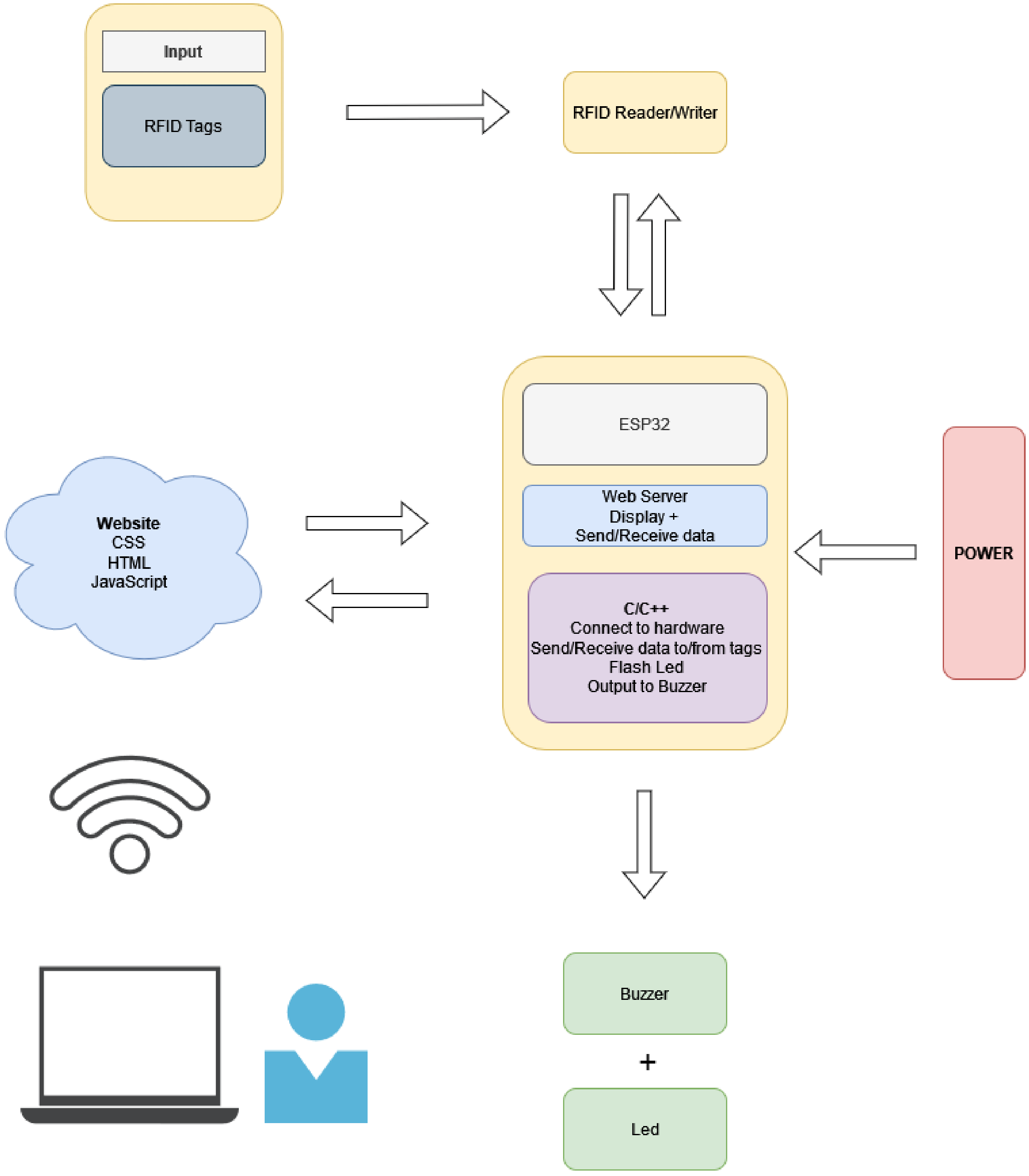


Figure 2.1 Architecture Block Diagram

Development platforms and IDEs

ESP32 [2]

The ESP32 is a microcontroller with Wi-Fi and Bluetooth connectivity. It is designed for low power consumption and has a high level of integration. The ESP32 is powerful enough to function as a standalone system, which is what I did. The entire project is run off of this board.

Arduino IDE [3]

The Arduino IDE is a software application that provides a platform for writing, compiling, and uploading code to specific microcontroller boards. The IDE has a text editor for writing code, a message area for displaying error messages and a console for displaying the output of the code. The IDE also includes tools for managing libraries and boards, as well as a serial monitor for interacting with the microcontroller. I used this to code in C and C++ and even parts of my website, mainly to integrate the site and the ESP32.

Notepad++ [4]

Notepad++ is a code editor and I made use of this software because it has features such as tabbed editing, this allowed me to work on many different files at once and syntax highlighting, this made code easier to read. I mainly used the program for testing snippets of my code and writing the bulk of the website.

Multiple browsers [5] [6] [7]

I used Firefox, Microsoft Edge and Google Chrome. They are all browsers, and it is what will be running the webpage I have built. So I tested on all three of these sites to check what different issues that may arise with each individual program.

3 Input

In this project I have only one input, this is the RFID module 'RFID-RC522'. Where the user will touch an RFID tag to the component which will act as the users only interaction to the project.

3.1 RFID Module

The RFID-RC522 is a module that can read and write data to RFID tags. The RC522 module contains a reader that can send and receive signals to and from RFID tags. When an RFID tag is placed close to the module, the reader sends out a radio signal that powers the tag and allows it to transmit the data back to the reader. This data can then be read by the ESP32. The RC522 module can also write data to certain types of RFID tags. This allows you to store information on the tag that can be read by other RFID readers.

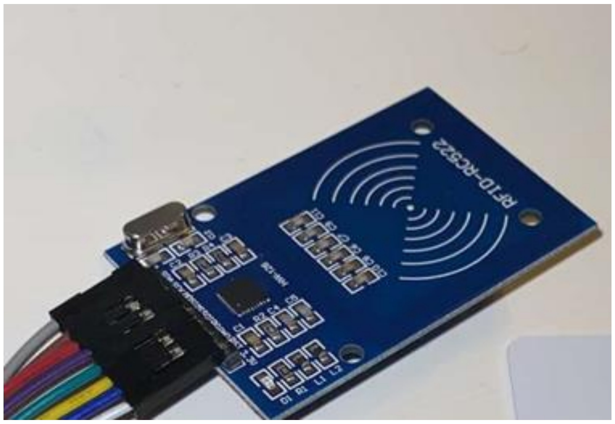


Figure 3.1.1 RFID Reader image

The purpose of the RFID module here is to read the RFID tag and pass on its associated number to the ESP32 which will then display that to the website for the user to see.

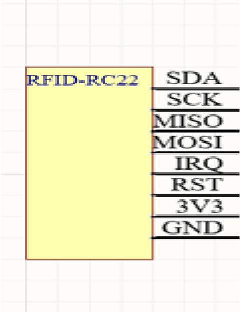


Figure 3.1.2 RFID Reader circuit

3.1.1 RFID Reader Sample Code

To use this module, I decided on using a library that already existed to help me use the RFID reader. After a short time researching which library to use, I discovered the MFRC522 Esp32 Library. This contains a lot of useful information on how to use and setup this component which I implemented into my project.

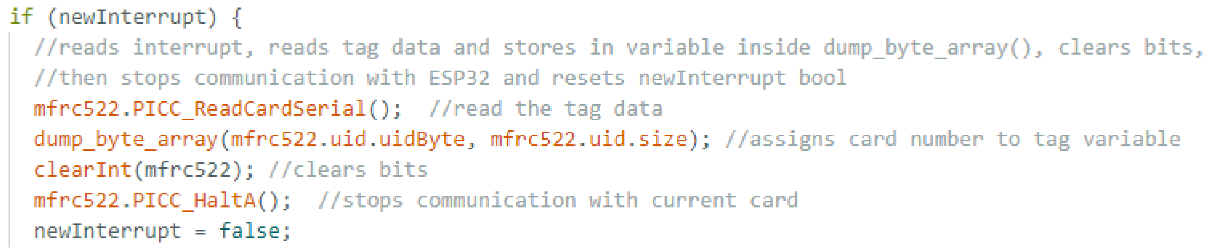
How the tag is read

This is done by calling multiple functions from the MFRC522 library that was linked above.

The code below checks the interrupt flag to see if it is true. If so, it reads the RFID tag data and then stores this into a string using the dump_byte_array function which iterates through a loop and builds up the string by converting the original hex value into a string and concatenating each character into the variable.

After this is done the register bits on the RC522 module are cleared and communication is stopped by the mfrc.PICC_HaltA function.

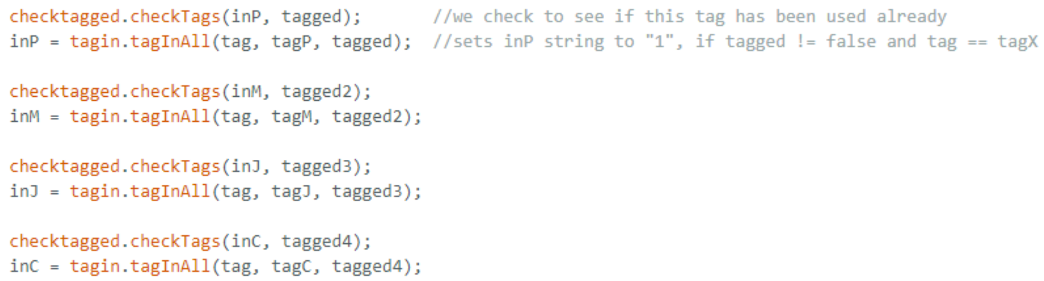
The interrupt flag is then set as false.



```
if (newInterrupt) {
  //reads interrupt, reads tag data and stores in variable inside dump_byte_array(), clears bits,
  //then stops communication with ESP32 and resets newInterrupt bool
  mfrc522.PICC_ReadCardSerial();  //read the tag data
  dump_byte_array(mfrc522.uid.uidByte, mfrc522.uid.size); //assigns card number to tag variable
  clearInt(mfrc522); //clears bits
  mfrc522.PICC_HaltA();  //stops communication with current card
  newInterrupt = false;
```

Figure 3.1.1.1 How an interrupt is called

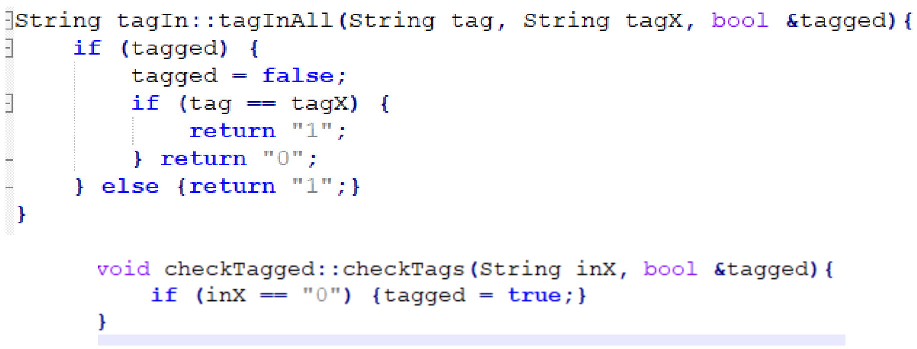
How we compare tag to known values



```
checktagged.checkTags(inP, tagged);       //we check to see if this tag has been used already
inP = tagin.tagInAll(tag, tagP, tagged);  //sets inP string to "1", if tagged != false and tag == tagX

checktagged.checkTags(inM, tagged2);
inM = tagin.tagInAll(tag, tagM, tagged2);

checktagged.checkTags(inJ, tagged3);
inJ = tagin.tagInAll(tag, tagJ, tagged3);

checktagged.checkTags(inC, tagged4);
inC = tagin.tagInAll(tag, tagC, tagged4);
```

Figure 3.1.1.2 Compare scanned tag to known tags

This requires two functions to be called from a small library I made – this was done to prevent writing repetitive code.



```
String tagIn::tagInAll(String tag, String tagX, bool &tagged){
    if (tagged) {
        tagged = false;
        if (tag == tagX) {
            return "1";
        } return "0";
    } else {return "1";}
}

void checkTagged::checkTags(String inX, bool &tagged){
    if (inX == "0") {tagged = true;}
}
```

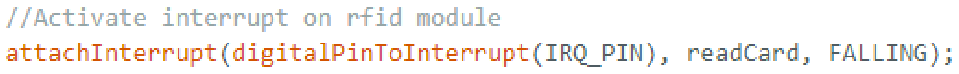
Figure 3.1.1.3 The code behind the function in Fig 3.1.1.2

In the first function tagInAll, we check I the Boolean tagged has been used before. If it hasn't then we proceed to change its state to false and compare the known tag to the one scanned in by the user. If there is a match, we return "1". Otherwise, we return "0". Outside of this if statement we return a "1" anyway when false because for it to be false it must have already been used.

The second function checks the users status, if it is a "0" then we reset the tagged value to true in order for it to be able to be compared to again. If this wasn't done then the tag wouldn't get checked again, preventing the program from working properly.

3.1.2 Interrupts

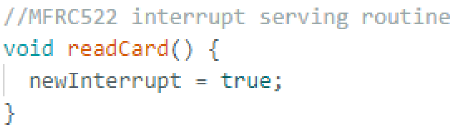
I used the modules built in interrupt call for this project to make the process to scan in faster. I was only able to do this with help from the library I had already used (found here).



```
//Activate interrupt on rfid module
attachInterrupt(digitalPinToInterrupt(IRQ_PIN), readCard, FALLING);
```

Figure 3.1.2.1 Activates the interrupt on the module

The interrupt is set up to run when the IRQ_PIN (the pin on the ESP32 that's associated with the IRQ pin on the RFID module) goes from high to low. When the interrupt is triggered, the readCard function is called.



```
//MFRC522 interrupt serving routine
void readCard() {
  newInterrupt = true;
}
```

Figure 3.1.2.2 Reset interrupt flag

The readCard function simply resets the interrupt flag to true so it can run the interrupt code from the void loop method.

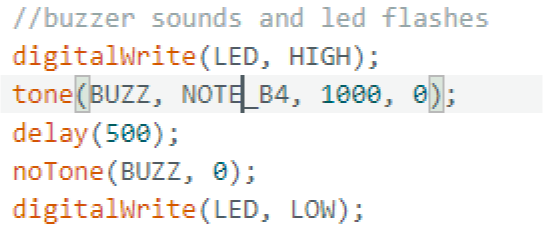
4 Output Devices

I have decided to employ the use of two components to be outputs in this project. They act as visual and auditory feedback for the user to show that they have tagged into class. The led flashes at 2Hz or 500ms. The buzzer also produces noise for this amount of time.

4.1 Buzzer & LED

4.1.1 Setup

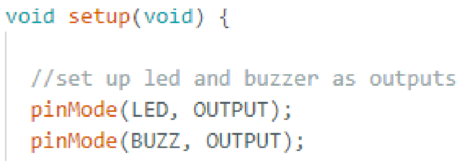
To use this buzzer, I found a library (Tone32) that implements the same functions that an Arduino would have to interact with a buzzer. It minimalised the code needed and made it far more readable than the alternative which is sending PWM (pulse width modulation) signal to it.



```
//buzzer sounds and led flashes
digitalWrite(LED, HIGH);
tone(BUZZ, NOTE_B4, 1000, 0);
delay(500);
noTone(BUZZ, 0);
digitalWrite(LED, LOW);
```

Figure 4.1.1.1 How buzzer and led are setup

The LED and Buzzer are set up as outputs using pinMode() in the void setup method.



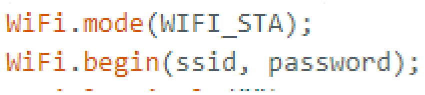
```
void setup(void) {

  //set up led and buzzer as outputs
  pinMode(LED, OUTPUT);
  pinMode(BUZZ, OUTPUT);
```

Figure 4.1.1.2 Sets components as outputs

5 Website

I wanted a website in this project to display results but to do that I needed to use two libraries. WiFi allows the ESP32 to connect to the internet with its onboard chip. WebServer lets the ESP32 handle HTTP requests.

5.1 WiFi



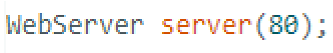
```
WiFi.mode(WIFI_STA);
WiFi.begin(ssid, password);
```

Figure 5.1.1 WiFi setup

This code connects the ESP32 to a WiFi network (the hotspot on my phone). It sets the WiFi mode to station mode and allows it to connect to a router and access the internet. After this we give it the credentials (ssid, password) to connect to the hotspot on my phone.

5.2 Web Server

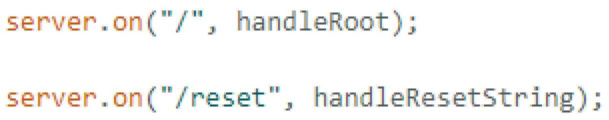
This is used to build and create the website by checking the connection to the internet and compiling the html/CSS/JavaScript into one site.



```
WebServer server(80);
```

Figure 5.2.1 Server object created

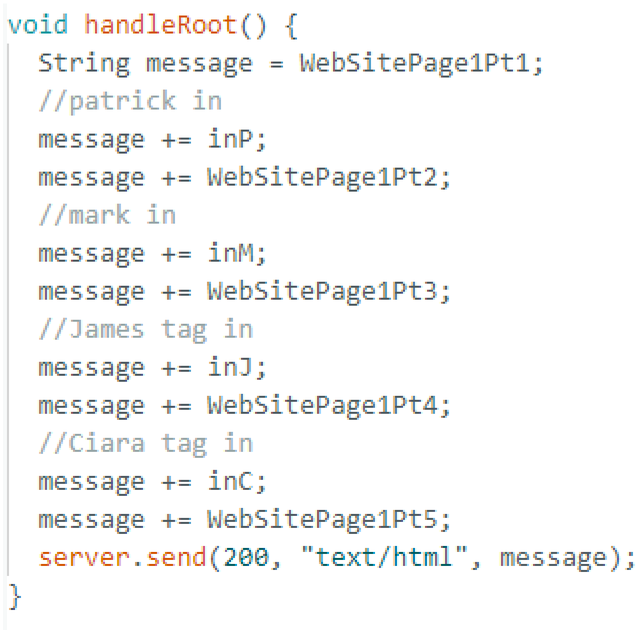
This creates the server on port 80.



```
server.on("/", handleRoot);

server.on("/reset", handleResetString);
```

Figure 5.2.2 Server.on functions

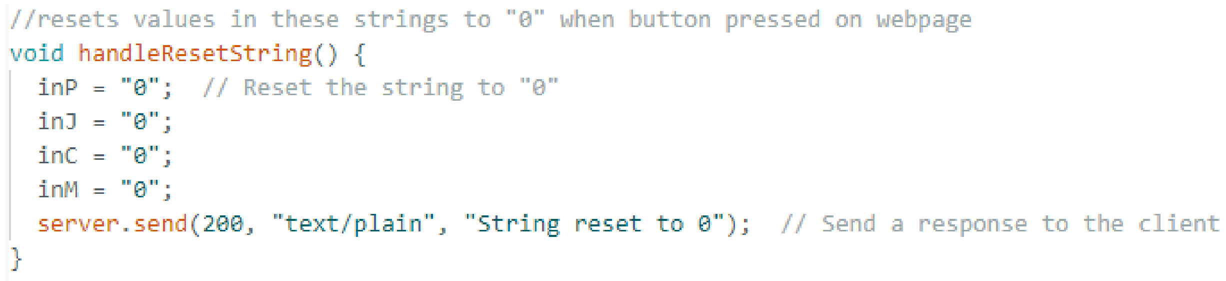
The server.on function allows a method to be executed whenever the specified paths (“/reset” or “/”) are reached.



```
void handleRoot() {
  String message = WebSitePage1Pt1;
  //patrick in
  message += inP;
  message += WebSitePage1Pt2;
  //mark in
  message += inM;
  message += WebSitePage1Pt3;
  //James tag in
  message += inJ;
  message += WebSitePage1Pt4;
  //Ciara tag in
  message += inC;
  message += WebSitePage1Pt5;
  server.send(200, "text/html", message);
}
```

Figure 5.2.3 handleRoot method, builds webpage

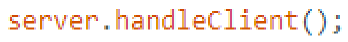
This code builds the webpage by breaking it down into strings and then concatenating them into one string called “message” at the end. This particular way was used so I could insert values to the website at certain points.



```
//resets values in these strings to "0" when button pressed on webpage
void handleResetString() {
  inP = "0";  // Reset the string to "0"
  inJ = "0";
  inC = "0";
  inM = "0";
  server.send(200, "text/plain", "String reset to 0");  // Send a response to the client
}
```

Figure 5.2.4 handleResetString function

When this function is called it resets the strings above to “0”.



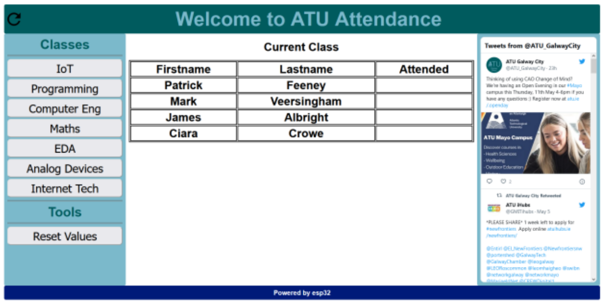
```
server.handleClient();
```

Figure 5.2.5 starts website

This is ran in the void loop method and calls all the server.on functions when requested.

5.3 HTML & CSS Code

5.3.1 Responsive Design



Firstname	Lastname	Attended
Patrick	Feeney	
Mark	Veersingham	
James	Albright	
Ciara	Crowe	

Tweets from @ATU_GalwayCity

ATU Galway City @ATU_GalwayCity · 23h

Thinking of using CAO Change of Mind? We're having an Open Evening in our #Mayo campus this Thursday, 11th May 4-6pm if you have any questions :) Register now at atu.ie/.openday

ATU Mayo Campus
Discover courses in
- Health Sciences
- Wellbeing
- Outdoor Education

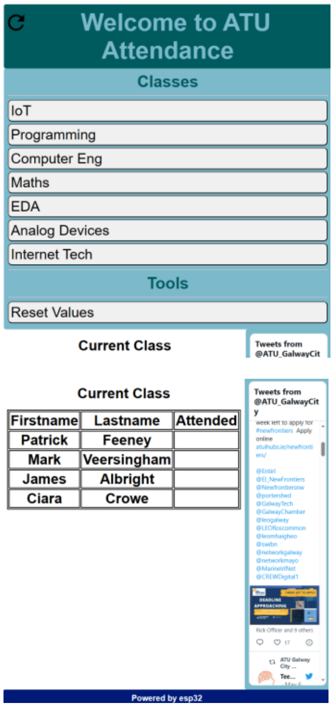
ATU Galway City Retweeted

ATU iHubs @GMITihubs · May 5

PLEASE SHARE 1 week left to apply for #newfrontiers. Apply online atuihubs.ie/newfrontiers/

@EntIrl @EI_NewFrontiers @Newfrontiersnw @portershed @GalwayTech @GalwayChamber @leogalway @LEORoscommon @leomhaigheo @swibn @networkgalway @networkmayo @MarineIrlNet @CREWDigital1

Full size



Tweets from @ATU_GalwayCity

Firstname	Lastname	Attended
Patrick	Feeney	
Mark	Veersingham	
James	Albright	
Ciara	Crowe	

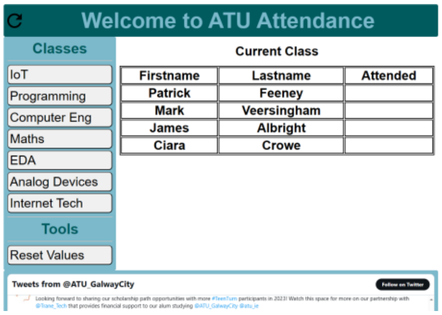
week left to apply for #newfrontiers Apply online atuihubs.ie/newfrontiers/

@EntIrl @EI_NewFrontiers @Newfrontiersnw @portershed @GalwayTech @GalwayChamber @leogalway @LEORoscommon @leomhaigheo @swibn @networkgalway @networkmayo @MarineIrlNet @CREWDigital1

Rick Officer and 9 others

ATU Galway City ...

Half size



Firstname	Lastname	Attended
Patrick	Feeney	
Mark	Veersingham	
James	Albright	
Ciara	Crowe	

Tweets from @ATU_GalwayCity

Looking forward to sharing our scholarship path opportunities with more #TeenTurn participants in 2023! Watch this space for more on our partnership with @Trane_Tech that provides financial support to our alum studying @ATU_GalwayCity @atu_ie

¾ size

5.4 Web Design for Accessibility

I have done the following to make my website as accessible as possible to all kinds of readers.

Perceivable

Web content is made available to the senses. Sight and hearing if ran through a text reader.

Operable

There is only one page for this website and only buttons which make the website easy to use for anyone.

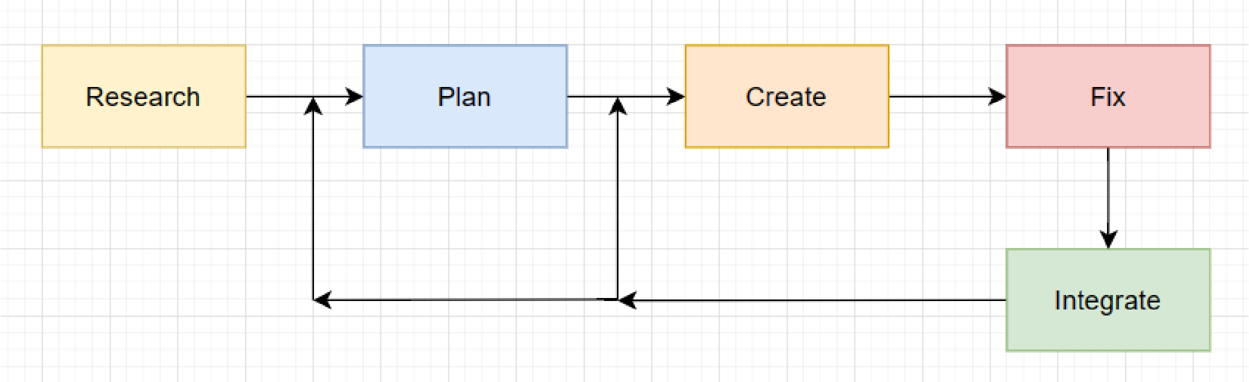
Understandable

The information on the webpage is easy to read as it is all within tables or laid out under related headings.

Robust

Content is mostly text, tables and buttons. There is only one use of an image, and it is a refresh button that could be avoided if the user wishes to.

6 System Integration



This was my design process for making ATU Attendance.

Research a section of the project and immerse myself in it however I could.

Plan out how I would use and benefit from the new addition.

Create the code or circuitry needed to get it working to just meet base requirements.

Fix this until it does exactly what I want.

Integrate into the project.

Sometimes I would need to repeat parts of this process for any given section if it was not working entirely after integration as this can sometimes break projects.

For example, my process for the website was this:

Researched HTML/CSS until I came across The Odin Project [8] which is a brilliant source for web development courses.

Planned out what I wanted the webpage to look like and what it’s function would be.

Created the skeleton of the page with barely any CSS and basic HTML.

Fix the site until it looks and acts close to how I planned.

Integrate this into the project, it definitely took a few loops from here to add/remove code from the website to fix it and get it to function as intended.

7 Problem Solving

RFID interrupts:

When I connected the RFID module and wrote the setup code, I assumed wrongly that it would read the RFID tag in a reasonable time. However, it was slow and very inconsistent. This was resolved after digging about an ESP32 library made for RFID module I had [9]. There was an interrupt pin on the module I was ignoring. Once setup, with help from the library's example, it was faster, consistent and accurate.

Buzzer not being the same as in Arduino

Buzzers being interacted with by an Arduino use simple functions like Tone(pin, frequency) and noTone(pin). Tone() turns the buzzer on and noTone() stops it. I tried to set up the buzzer using these functions on the ESP32 however there were errors. After some research I discovered that Espressif did not implement these, and I did not want to set up PWM (Pulse Width Modulation) and attach it to a pin to output to the buzzer. Instead, I found a library that did this and I was able to call the same functions as an Arduino.

No RFID reader until January, replaced with switches

A component I ordered was not available to me until January of 2023, this was known to me around October of 2022. Instead of stopping progress or waiting, I decided to replace the RFID section of the project with switches. It helped me further along with my website and how certain functions would need to work when I finally did get my RFID components.

Used a search/find algorithm to iterate through a vector

I was stuck at a point in my project on how to approach storing strings that would be added to and be referenced to. I didn't consider using C++ until a Lecturer mentioned it might be worth looking at after discussing this problem. They pointed me to vectors which solved half of the problem and left me to figure the rest out. This is when I discovered iterators and an algorithm to search through a vector for a particular value. I needed to compare the scanned in RFID tag to a known list of tags and this worked perfectly. With some fiddling with a very common find algorithm [10] I was able to get the location and the corresponding user's name of the tag.

8 Impact of Project on Sustainability

My project is not inherently aimed at being more sustainable and a better choice for the environment, but it still assists in the United Nations 17 goal by doing the following:

Quality Education (4): It allows the teacher to efficiently take roll and allows for more total teaching time and focused learning.

Responsible Consumption and Production (12): The project I am making is made of items that are reusable and cheap. The school wastes paper on ticking a box for every class, every day. This reduces paper waste

Health and Safety is not really an issue with this project as it will be contained in a cardboard box and has no high voltages or current going through any component.

Plagiarism/referencing: Any code used for my project that isn’t original has been referenced above or is part of a library that I am using to make my program run more efficiently.

9 Conclusion

After completion of this project, I have a prototype for an attendance system that could be implemented into a place of education or a workplace.

The system uses RFID technology to read (RFID Reader) and store data taken from RFID tags. This is then translated to the website showing the user which members have clocked in for the class/workday.

The buzzer and LED give visual and audible signals to tell the person they have successfully tagged in.

Potential improvements would be to include a database and a website hosted online in order to progress this project more.

10 References

[1] [Online]. Available: https://www.wireless-technology-advisor.com/what-is-rfid.html.

[2] [Online]. Available: https://www.espressif.com/en/products/socs/esp32.

[3] [Online]. Available: https://www.arduino.cc/en/software.

[4] [Online]. Available: https://notepad-plus-plus.org/.

[5] [Online]. Available: https://www.microsoft.com/en-us/edge?form=MA13FJ&exp=e00.

[6] [Online]. Available: https://www.mozilla.org/en-US/firefox/new/.

[7] [Online]. Available: https://www.google.com/chrome/.

[8] [Online]. Available: https://www.theodinproject.com/.

[9] [Online]. Available: https://github.com/miguelbalboa/rfid.

[10] [Online]. Available: https://www.geeksforgeeks.org/std-find-in-cpp/.

Appendix 1: Code

The file below contains all code required to run the project. Its is packed in a zip file as there are multiple files including the small library wrote by me.

ATU_Attendance.zip

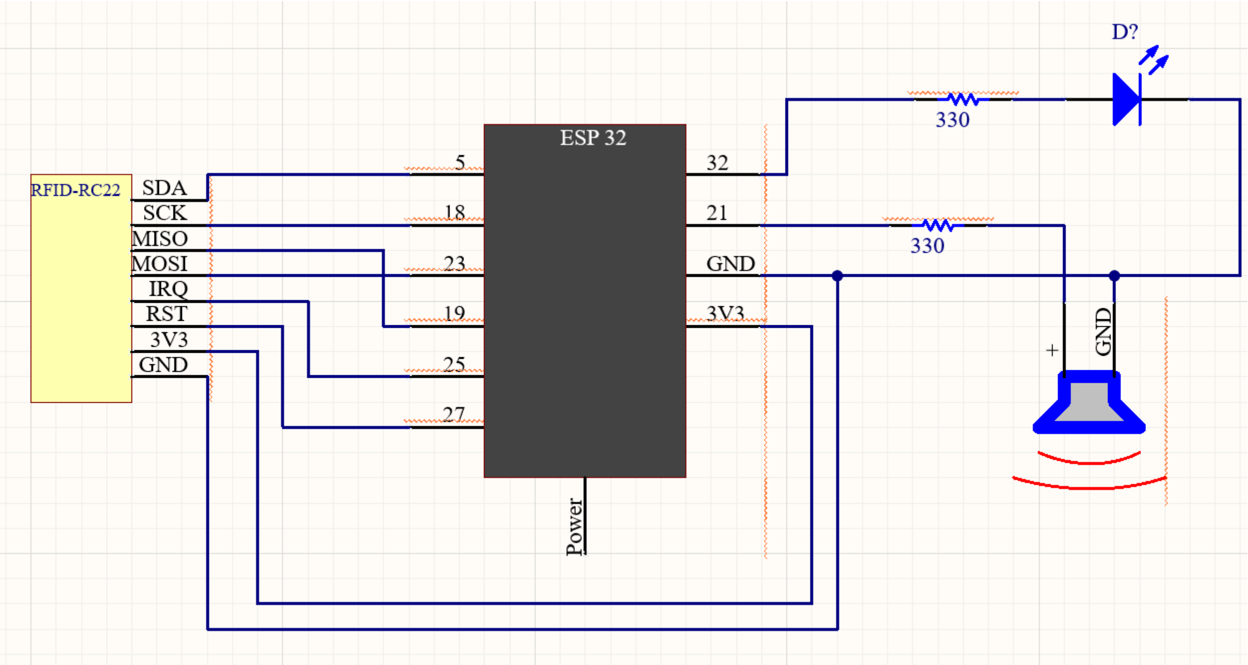
Appendix 2: Bill of Materials

Item	**Quantity**	**ATU Stores**	**Sourced from**	**Cost Euros**
ESP32*	**1**	n	www.amazon.co.uk	**3.30**
RFID Module*	**1**	n	www.amazon.co.uk	**0.70**
RFID Tag*	**6**	n	www.amazon.co.uk	**0.60**
LED*	**1**	n	www.amazon.co.uk	**0.03**
Buzzer*	**1**	n	www.amazon.co.uk	**0.17**
Total Cost				**4.80**

Components marked with an asterisk indicate that the I have sourced this component myself.

Total cost does not cover resistors or wires.

Appendix 3: Schematic


D?
330
ESP 32
5
32
RFID-RC22
SDA
SCK
MISO
MOSI
IRQ
RST
3V3
GND
18
21
330
23
GND
19
3V3
25
27
+
GND
Power