

Bartosz_Galantowicz_1700994_ EIGP_2_ACCA5003_Assignment

by Bartosz Galantowicz

Submission date: 22-Mar-2020 08:56PM (UTC+0000)

Submission ID: 122527582

File name:

2275_Bartosz_Galantowicz_Bartosz_Galantowicz_1700994_EIGP_2_ACCA5003_Assignment_210055_1258359789.docx
(8.11M)

Word count: 4775

Character count: 26379

Bartosz Galantowicz

1700994

Enterprise & Innovation Group Project

ACDA 5003

James Williams

“EIGP 2 – Individual Report”

26/02/2019

Table of Contents

Introduction	3
Development	3
Software and Hardware Integrator	3
The Market and Competitors – Legal Aspects	17
References	20
Appendix	22
Reflection	22
Networking	23
Team-working	33
ESP8266 Connection And Using AT Commands Document	39

Introduction

The report is a part of a new product business innovation plan and documentation of the contribution in the project for tasks assigned by Tsunami Soft. Furthermore, the evidence of the developed skills and gathered knowledge is included in this document. However, the described tasks and activities are related to and focus on the particular performed roles in the team.

Development

Software and Hardware Integrator

The task of Software and Hardware Integrator was to research and develop a network system for providing a connection between used hardware and software. Accordingly, obtaining knowledge and skills to solve this issue was demanded.

The idea of the project was to use the Arduino board as a remotely connected scale. Namely, the weighting scale must be linked with the Unity application through Wi-Fi and accordingly sending data from the board to the game.

Firstly, internet technology was studied to understand the aspects of networking and how the Internet works.

The Internet is a wide world network of computers, which is used to send packets of digital data [1]. Thus, computer networking is a collection of computers, servers or other digital devices connected to each other to exchange the sharing data [2] [3].

In the case of the project, the information from the scale is transferred to the application. This is possible by sharing the same network by the scale and a device running the game. The concept was that the Arduino board is registered in local (home) network and if the application is opened and connected to the same network, a user is able to get the data (weight of the measured product) from the scale.

However, to accomplish the given task, it was a priority to design and build a prototype of the scale. Basing on research about Arduino boards and the fact that the final weighing scale prototype had not been done yet it was decided to build a sub prototype to start programming practise and conducting experiments with hardware and software.

Consequently, it was a simple architecture of Arduino and HX711 module with a load cell. HX11 is specially made for amplifying the signal from load cell circuit board, while the load

cell is a piece of metal with attached strain gauges. The strains gauges are resistors, and their resistance are modified by bending a load cell [4] [5].

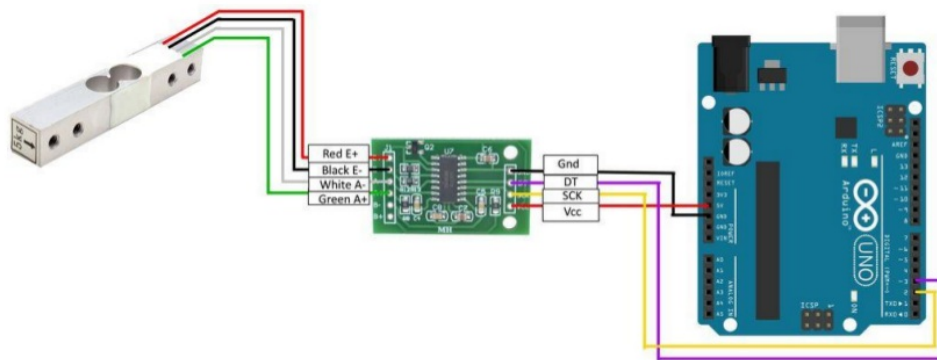


Fig.1 The wiring diagram for the load cells and HX711 with Arduino board (<https://cdn.instructables.com/FH7R58EJ7UR7KPT/FH7R58EJ7UR7KPT.LARGE.jpg?auto=webp&width=1024&fit=bounds>).

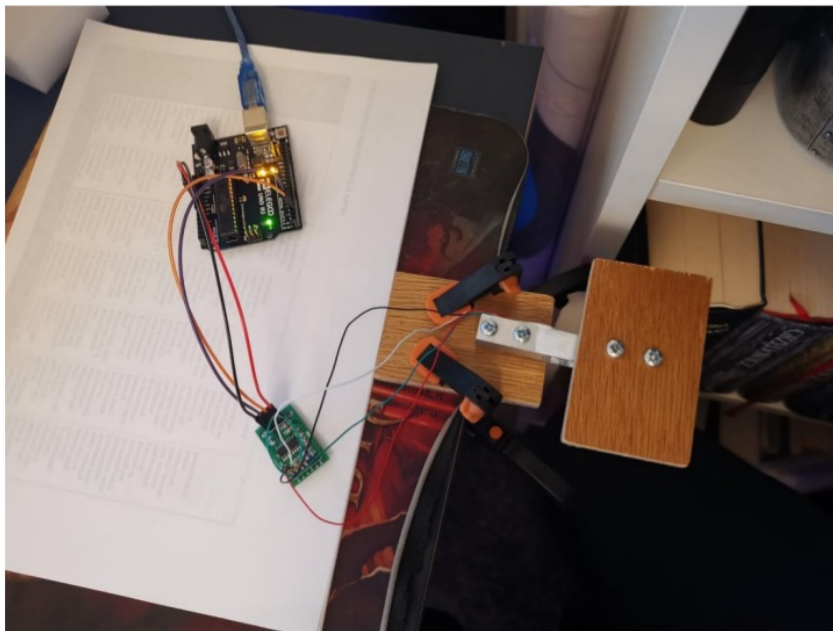


Fig.2 The photo is showing the constructed prototype for a scale, which was used to test code.

When the environment for work had been prepared, the rest of the job was depended from the software. Therefore, research for additional skills about microcontrollers and Arduino language had to be obtained.

Fortunately, many useful tips and tutorials are provided on the official website of Arduino enterprise. Hence documentation of various Arduino boards, as well as basic tutorials, are readily available to get and follow. Moreover, various Arduino projects are shared by enthusiasts of the technology on the well-developed Arduino project hub community website [6]. It is not only a source of inspiration but also a mine of information about Arduino development opportunities.

Programming Arduino is managed by Arduino Software, which is an integrated development environment for Arduino boards. When the software has been installed the IDE is ready for implementation of the code. Furthermore, the Arduino language is based on C language, and it is applied to creating “sketches” – scripts of Arduino language code. Some of the functions of Arduino language are strictly specific due to microcontrollers features. Additionally, it is allowed to upload a created program to the device, which is connected to the computer through USB cable. Since the moment of accomplishment of the uploading “sketch”, the board is programmed, and it can work independently from a cable connection [7].

Consequently, when the research for programming Arduino devices had been finished, the program for working weighing scale was coded and implemented. After setting a calibration value by using a known-weight item, the weighing scale was able to start reading weight.



```

HX711Scale | Arduino 1.8.9
File Edit Sketch Tools Help

HX711Scale $
/*

This example demonstrates basic scale output. See the calibration sketch to get the calibration_factor for your
specific load cell setup.

Arduino pin 6 -> HX711 CLK
Arduino pin 5 -> HX711 DOUT
Arduino pin 5V -> HX711 VCC
Arduino pin GND -> HX711 GND

*/

#include "HX711.h"

#define DOUT 5
#define CLK 6

HX711 scale;
float units;

float calibration_factor = 446; // this calibration factor is adjusted according to my load cell.

void setup() {
  Serial.begin(9600);
  Serial.println("HX711 scale demo");

  scale.begin(DOUT, CLK);
  scale.set_scale(calibration_factor); //This value is obtained by using the HX711Calibration sketch.
  scale.tare(); //Assuming there is no weight on the scale at start up, reset the scale to 0.

  Serial.println("Readings:");
}

void loop() {
  scale.set_scale(calibration_factor);

  Serial.print("Reading: ");
  units = scale.get_units(), 10;
  if (units < 0)
  {
    units = 0.00;
  }
  Serial.print(units);
  Serial.print(" g");
  Serial.println();
}

```

Fig.3 The screenshot of the code in the Arduino IDE, which was used to program a simple scale.



Fig.4 The example of using the scale. The statuette is placed on the load cell, and that information is registered by resistors, which send registered data to the Arduino board. Thereby, the result of the operation is printed in the console window.

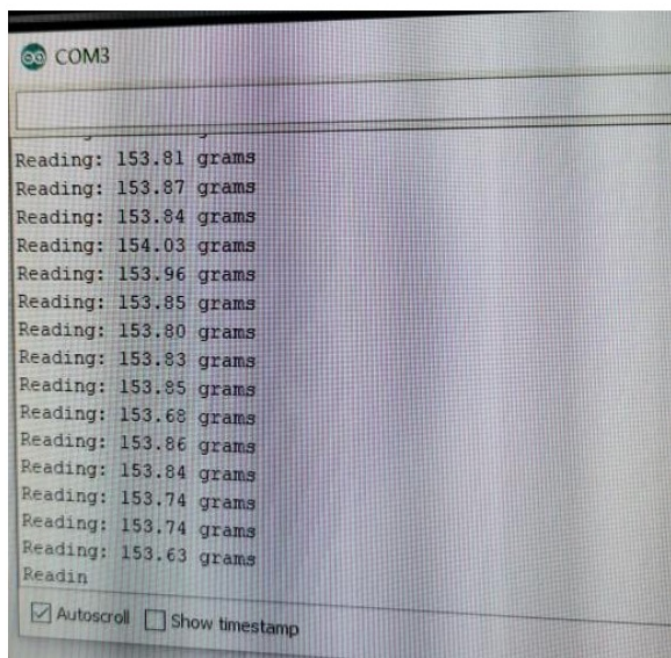


Fig.5 The figure is presenting the results of the scale measurement.

Process of constructing and testing of the simple scale was a field for gathering experience about coding Arduino and microcontrollers. The next step assumed to merge all past collected information and knowledge to connect the weighing scale to the Internet.

Therefore, research for different methods of connecting Arduino with a network was carried out. It had been determined that an additional module for the circuit board was required because this model of Arduino is not equipped in a Wi-Fi shield, which is used for network operations. Accordingly, the ESP8266 module was chosen as a network adapter for the device [8]. What is more, it is the cheapest and most efficient solution in comparison with other considered options [9].

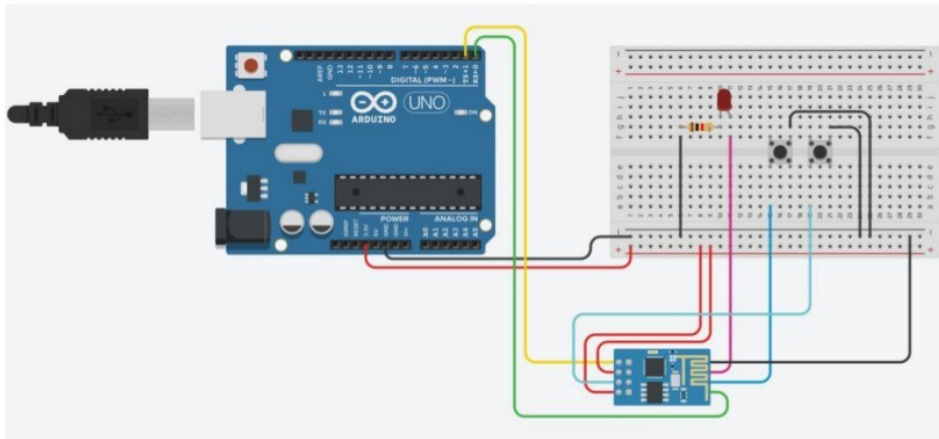


Fig.6 The wiring diagram for the ESP8266 module connection with Arduino board (<https://cdn.instructables.com/FFK/7HYA/JB3ZSEUE/FFK7HYAJB3ZSEUE.LARGE.jpg?auto=webp&width=1024&fit=bounds>). The most left button is used to enable flashing the ESP8266 module. In contrast, the reset connection button is the most right button.

Further, when the ESP8266 had been connected to Arduino, the device was ready to flash the memory of module. However, firstly the Arduino Software IDE had to be set up to cooperate with the connected boards. Because of setting up the IDE, the particular manager for programming microcontrollers was required to download [10].



Fig.7 The link relating to the ESP8266 board manager installation files must be set in the Arduino Settings window to download the package.

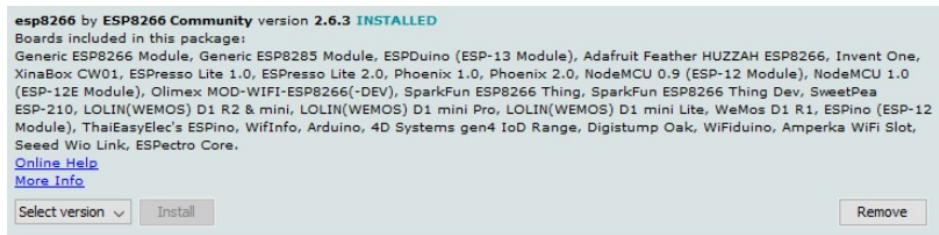


Fig.8 If the link has been set the installation of the new package can be carried out by the IDE. Since that change, the selected board in the menu should be “ESP8266 Generic” [9].

Nevertheless, setting the IDE was not the end of the task because of a requirement of coding the internet connection handler.

The very first step was to understand how the new built Arduino setup can communicate across the network. AT commands were analysed and used for this reason. Furthermore, it is allowed to apply them in the serial window of an Arduino board. Namely, AT commands are instructions for controlling modems and enabling client-server relationship [11].

Given the above, a short report about AT commands was documented to set up a step by step instruction of using them. In result, it was possible to register the Arduino device to the local network and linked it with the HTTP server. Accordingly to provided instructions, the data was transferred and then displayed in the browser windows at the address settled for the device.

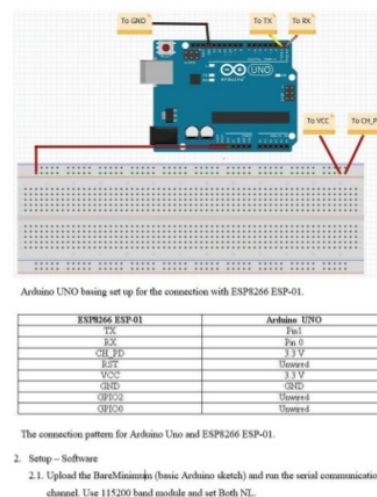
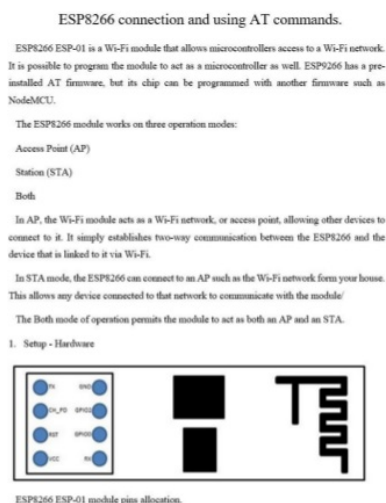


Fig.9 The fragment of a written document about using AT commands and setting up ESP8266 module (The full version of the document has been attached to Appendix section).



```

COM3
AT
OK
AT+WMODE=3
OK
AT+CWJAP
OK
AT+RST
OK
Connecting Wifi....
AT+CWJAP="VodafoneConnect81074253","ukis965jvlja6fe"
AT+CWJAP="VodafoneConnect81074253","ukis965jvlja6fe"
OK
Wifi Connected
IP Address:"192.168.1.14"
Port:80
AT+CWMUX=1
OK
AT+CIPSERVER=1,80
OK
System Ready..
Remove all weight from scale
After readings begin, place known weight on scale
☐ Autoscroll ☐ Show timestamp Both NL & CR 9600 baud Clear output
  
```

Fig.9 The figure is showing a process of connecting Arduino to the network and the implementation of AT commands in the serial monitor.

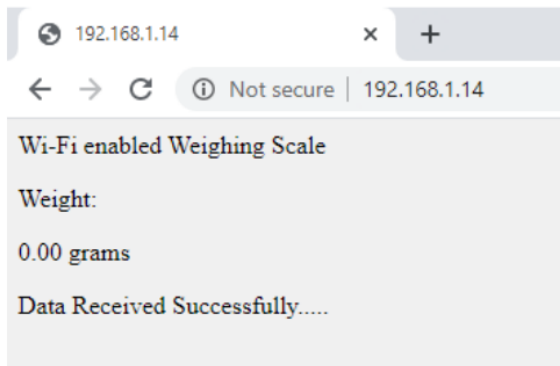


Fig.10 Typing the accessed IP address of the board in a web browser is applied to send the HTTP request to fetch data. For instance, if the website is going to be refreshed, the new actualised data will be displayed. Namely, changes in weight by putting a mass on the load cell are continuously updated by the scale. Nevertheless, the updated result can be displayed on the website only after reloading the page.

The first attempt at networking experiments was successful. Hence it was a pivot point in the development process of the project. However, follow the research about networking, it was decided to send data between the scale and the application by settling a TCP connection. To establish such a network, a server, and client are determined. The server is bound to a port, which is opened for exchanging a stream of data. In this example, the scale is a server which is waiting for the application response (client). If the app is run the execution of the code measuring weight is started, and the input data from the scale is sent to the application. Additionally, the guarantee of delivering packets over the network in the same order in which they were sent is arranged due to used TCP protocol [12] [13].

At this stage of the development process, the prototype was assembled with inclusion both additional modules (HX711 and ESP8266).

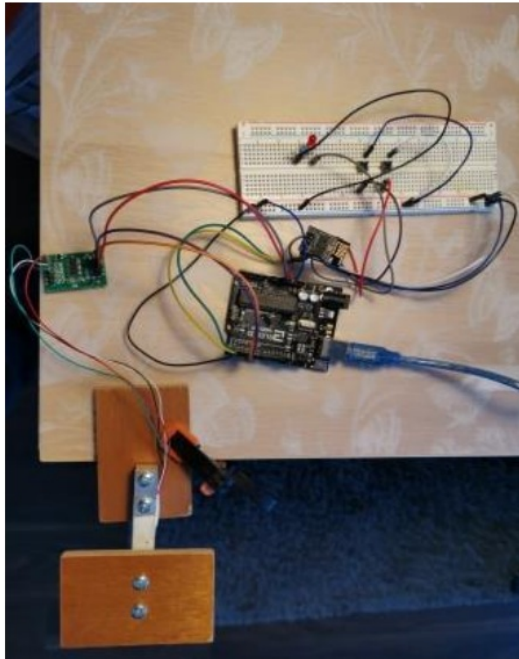


Fig.11 The figure is showing the final version of the prototype, which is made of Arduino board, HX711, ESP8266 modules and load cell.

Moreover, the new program was coded in Arduino Software for the weighing scale model. Its primary role is to initialise the server on the Arduino device and control the input from the loading cell.

```
Unity-Arduino_Client
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include "HX711.h"

const char* ssid = "VodafoneConnect81074253";
const char* password = "ukis965jvlja6fe";

// Initialise server and client
WiFiServer server(26); // 26 - the server port
WiFiClient client;

#define DOUT 2
#define CLK 3
HX711 scale;
float calibration_factor = 446; // This calibration factor is adjusted according to the load cell.
float units; // Stores data retrieved by the scale
bool isScaleConnected = false;

void setup() {
  Serial.begin(9600);

  // Set pins
  pinMode(DOUT, OUTPUT);
  pinMode(CLK, INPUT);

  // Start Wi-Fi connection
  WiFi.begin(ssid,password);
  Serial.println("Connecting");

  while(WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }

  Serial.print("Connected to ");
  Serial.println(ssid);

  Serial.print("IP Address: ");
  Serial.println(WiFi.localIP());

  // Start the TCP server
  server.begin();
}
```

```

void loop() {
  // Listen for connecting clients
  client = server.available();
  if (client){
    Serial.println("Client connected");

    while (client.connected()){

      // Set the scale if the connection is established for the first time
      if(isScaleConnected == false)
      {
        Serial.println("Remove all weight from scale");
        Serial.println("After readings begin, place known weight on scale");
        scale.begin(DOUT, CLK);
        scale.set_scale();
        scale.tare(); //Reset the scale to 0
        isScaleConnected = true;
      }

      // Start start weighing
      scale.set_scale(calibration_factor);
      units = scale.get_units(), 10;
      if (units < 0)
      {
        units = 0.00;
      }
      client.print(units);
      client.print('\r'); // In C this is the carriage return

      // Delay before the next reading
      delay(10);
    }
  }
}

```

Fig.12 The screenshot of the code responsible for handling the server connection and data from the loading cell.

Subsequently, it was necessary to make up the code for the client (the application). The app was developed in Unity game engine with accompany of C# language scripting system. Therefore, additional time was planned to find out how network technology is associated with Unity and to learn advanced C# language aspects.

The solution was socked programming in C#. It is a one-way Server and Client setup where the messages are sent by a Client to Server, and they are shown by the server. Consequently, the communication between devices is successful, while the IP of the other node is recognised. However, the IP address to connect with must be known, before creating a client [14].

In Unity, it is allowed to use C# socked programming techniques together with Unity technology. Thus, the input controller script was created, which the main task is to apply the TCP protocol procedure to link the game with the scale [12] [13]. Once, the client (the application) is recognised by a server (the scale), the measurement of the weight is begun, and input is sent to the related node.



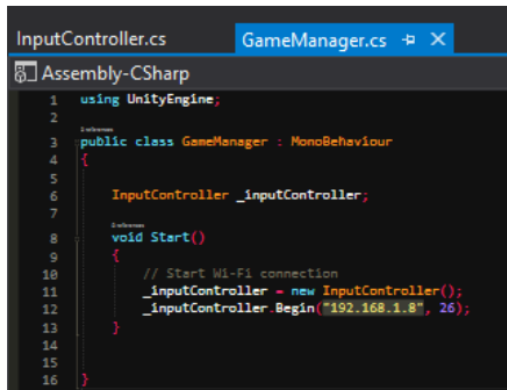
```

1 using UnityEngine;
2 using System.Collections.Generic;
3 using System.IO;
4 using System.Net.Sockets;
5 using System.Text;
6 using System.Threading;
7
8 public class InputController
9 {
10     public float CurrentValue;
11
12     public void Begin(string ipAddress, int port)
13     {
14         // The network is synchronised by using operations on thread
15         var thread = new Thread(() =>
16         {
17             // Initialise client connections for TCP network services
18             var client = new TcpClient();
19
20             // Connect with Arduino
21             client.Connect(ipAddress, port);
22             var stream = new StreamReader(client.GetStream());
23             Debug.Log("Connected");
24
25             // Read values and buffer them up
26             var buffer = new List<byte>();
27
28             while (client.Connected)
29             {
30                 // Read the next byte
31                 var read = stream.Read();
32
33                 // We split readings with a carriage return
34                 if (read == 13) // Regarding ASCII "13" is a carriage return
35                 {
36                     // Convert buffer to a string since the values are coming as strings
37                     var str = Encoding.ASCII.GetString(buffer.ToArray());
38
39                     Debug.Log(str);
40
41                     // Clear the buffer ready for another reading
42                     buffer.Clear();
43                 }
44                 else
45                 {
46                     // If this wasn't the end of a reading, then just add this new byte to our buffer
47                     buffer.Add((byte)read);
48                 }
49             }
50
51         });
52     }
53
54     thread.Start();
55 }
56

```

Fig.13 The screenshot of the InputController class, which is managed to keep TCP connection between the Arduino device and the client.

Nevertheless, the game is managed by a game manager object, which is the centrum of program control in Unity. For instance, the game manager is typically responsible for instantiating a game scene, keeping track the user progress or cameras movement. In this case, it was used to set up the input controller object while the application is run by a user. Hence, the input controller is initialised, and the Wi-Fi connection can be settled.



```
InputController.cs  GameManager.cs
Assembly-CSharp
1  using UnityEngine;
2
3  public class GameManager : MonoBehaviour
4  {
5
6      InputController _inputController;
7
8      void Start()
9      {
10         // Start Wi-Fi connection
11         _inputController = new InputController();
12         _inputController.Begin("192.168.1.8", 26);
13     }
14
15
16 }
```

Fig.14 The example of initialising the input controller object in the game manager.

Summarising, when the development of hardware and software had been finished, the bundle could be tested. Firstly, setting the IP address for the scale was processed.



Fig.15 The figure is showing the acquired IP address of the scale.

Then, when the application was run, the information about the connection was displayed in the console window in Unity.

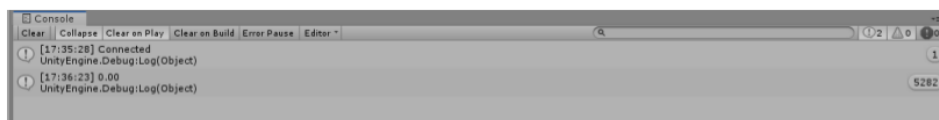


Fig.16 The figure is indicating that the network between the app and the Arduino board was established successfully.

Similar information was notified by the scale. The process of reading the measurement and sending the outcome was begun.

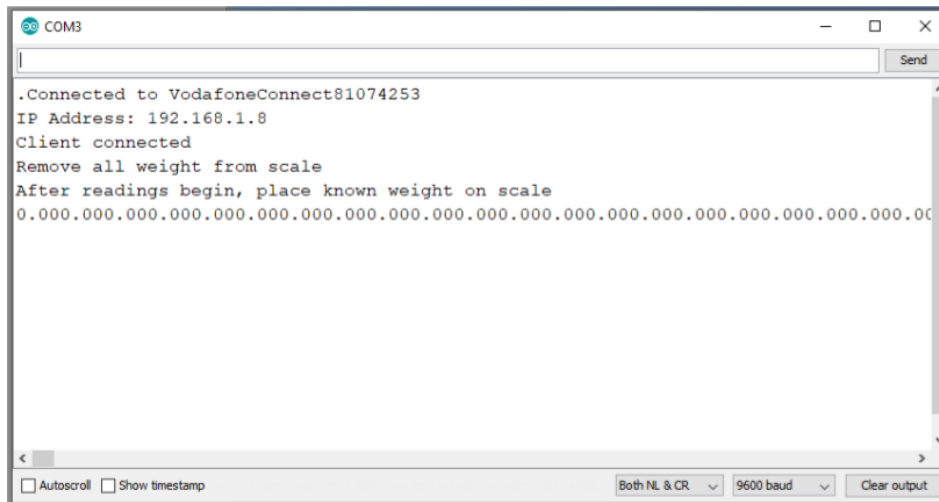


Fig.16 The screenshot is representing the outcome of the successful connection.

In conclusion, the given task for Software and Hardware Integrator has been accomplished. Due to research and experiments, the working prototype has been created. It is able to connect with a client remotely by using the internet connection. Furthermore, the new skills and experience have been gained during the development process what was the additional part of the assignment.

The Market and Competitors – Legal Aspects

The primary role of the legal system researcher was to analyse the legal aspects of application development and functionality of an enterprise. Accordingly, the understanding of the enterprise law and the structure of various types of agreements was crucial for the further appliance of the start-up.

If something new is created, patent and copyrights are the very first things which are come to mind. However, in the case of software development, it is not common to patent a program as the procedure of it is rather complicated and time expensive. The world is changing every day, especially the growth in technology development is rapid [15]. Hence creators of new software are focused on making a profit from the product, rather than struggling to secure the source code and ideas.

On the other hand, the work of the Tsunami Soft is protected by copyright. Basically, the right to reproduce the invented product is given automatically to the inventor in the form of copyrights. Thus, any piece of work created like reports, drawings or piece of code are covered by copyrights [16].

Notwithstanding the rights of inventors, there were also other topics about the legal system to investigate– the agreements. An agreement is a negotiated and typically legally connected arrangement between parties. Namely, it is a set of rules, prohibitions and privileges describing the relationship and responsibilities. For Tsunami Soft, the essential agreements may be related to the Non-Disclosure Agreement and End-User Licence Agreement (EULA). The first one is signed between at least two parties which outline confidential work and prevents parties from sharing the topic of conversation to or by third parties. It is a guaranty of the protection of the idea [17]. While the EULA is a contract between software developers and end-user, it is about the terms and conditions which are set for using the product. Furthermore, the agreement must be displayed to the user in a conspicuous position of an application window. The decision of acceptance is depended from the response of a user [18].

The effect of the research on the legal side of enterprises was preparing the structure of Non-Disclosure, and EULA agreements for Tsunami Soft. The aim was to make the start-up project more valuable, trustful and reliable product for the potential co-workers and investors. Accordingly, the structures of both agreements have been attached to the appendix section of this report.

Further studies were focused on the releasing of the application to the market and potential legal issues and arrangements which might be required to solve or initiate. Furthermore, the will of Tsunami Soft is to distribute the application over the most popular digital distribution service, which is Google Play. Therefore, the research was focused on this particular platform.

Nevertheless, placing the application on the Google Play Store is related to establishing the agreement between the producer of the application and the distribution platform. Hence, the developer distribution agreement has been prepared by Google to regulate the distribution aspect of applications [19].

Google Play Developer Distribution Agreement includes 16 main topics:

1. Definitions
2. Accepting this Agreement
3. Commercial Relationship, Pricing, Payments and Taxes
4. Use of Google Play by Ourselves
5. Authorisations
6. Brands Features and Publicity
7. Promotional activities
8. Product Takedowns
9. Privacy and Information
10. Terminating this Agreement
11. Representations and Warranties
12. DISCLAIMER OF WARRANTIES
13. LIMITATION OF LIABILITY
14. Indemnification
15. Changes to the Agreement.
16. General Legal Terms

If the Google Play developer distribution agreement is not accepted by both sides, then uploading the application on the Google Play store is not possible [19]. Therefore it might be necessary to find another source of distribution.

Moreover, in 2017 the prominent disclosure requirement was introduced by Google. This disclosure is another contribution of the potential product distributor, which is applied in concern of users good. The role of the requirement is to inform users, which data is collected by the application and how it will be used within the app.

It is the responsibility of developers to notify a user of data collected by the app. For instance, a user might be asked about allowing access to the content of its gallery or using a camera in a smartphone. If the prominent disclosure requirement is not attached to the app, it might be removed from the Google Play store [20].

Additionally, political and other legal aspects have been taken into consideration. The impact of those two features might be crucial for the performance and development of Tsunami Soft. Thus, the possibility of coming opportunities, events or actual political and legal situation was analysed and presented in the form of the table.

Political & Legal
The election is coming in December – Labours have pledged free broadband for all homes and businesses, thereby the number of the potential users/clients would be increased.
No-deal Brexit would affect the economy: • 28% of food is imported from the EU – the prices of food may increase. • 45% of gas supplies comes is imported via European pipelines – electricity and gas prices may go up, so preparing food using kettle or stove might be more expensive. • The government has warned that the cost of using a bank card to pay for goods and services may become more expensive outside the UK.
Video Games Tax Relief – is a form of financial help provided by the government. Game studios can claim a cash repayment or tax relief from the government after they have spent money on developing a video game.
General Data Protection Regulation (GDPR) – is a legal framework that sets guidelines for the collection and processing of personal data. The document provides greater control over personal data and improves privacy. These regulations must be fulfilled.
Copyright protection for Software – it is given automatically; however, there is not a register of copyright in the UK. You can mark your work with the copyright symbol (©), your name and the year of creation. Whether you mark the work or not doesn't affect the level of protection you have. Copyright prevents people from: • copying your work • distributing copies of it, whether free of charge or for sale • renting or lending copies of your work • performing, showing or playing your work in public • adapting your work • putting it on the Internet
Non-disclosure agreement – This agreement is a legal contract signed between two or more people which states that the information will not be shared with others. NDA can be beneficial between members of the team, so prevent them from pocket the work for themselves. Moreover, it can be helpful to sign NDA if the group find sponsors, or company happy to buy the idea, as it will prevent the theft of an idea.

Fig.17 The table is showing the Political and Legal outcomes of PESTLE analyses about the development of the project of Tsunami Soft. The research was carried out in December 2019.

References

- [1] F. Halsall, Computer Networking and The Internet, 5th ed., Harlow: Pearson Education Limited, 2005, p. 155.
- [2] F. Halsall, Computer Networking and The Internet, 5th ed., Harlow: Pearson Education Limited, 2005, pp. 4-14.
- [3] GeeksforGeeks, "Basics of Computer Networking," [Online]. Available: <https://www.geeksforgeeks.org/basics-computer-networking/> . [Accessed 02 February 2020].
- [4] DegrawSt, "Arduino Scale With 5kg Load Cell and HX711 Amplifier," [Online]. Available: <https://www.instructables.com/id/Arduino-Scale-With-5kg-Load-Cell-and-HX711-Amplifi/>. [Accessed 6 January 2020].
- [5] ARDUINO PROJECT HUB, "Digital Force Gauge & Weight Scale w/ Loadcell & Arduino," Arduino, [Online]. Available: <https://create.arduino.cc/projecthub/electropeak/digital-force-gauge-weight-scale-w-loadcell-arduino-7a7fd5>. [Accessed 6 January 2020].
- [6] Arduino, "ARDUINO PROJECT HUB," Arduino, [Online]. Available: <https://create.arduino.cc/projecthub>. [Accessed 3 January 2020].
- [7] Arduino, "Arduino Software (IDE)," [Online]. Available: <https://www.arduino.cc/en/Guide/Environment>. [Accessed 29 January 2020].
- [8] Expressif Systems, "ESPRESSIF SMART CONNECTIVITY PLATFORM: ESP8266," Expressif Systems, 12th October 2013. [Online]. Available: https://nurdspace.nl/images/e/e0/ESP8266_Specifications_English.pdf. [Accessed 29 January 2020].
- [9] squix78, "ESP8266 module comparison: ESP-01, ESP-05, ESP-12, ESP-201, Test Board and NodeMCU," 16th March 2015. [Online]. Available: <https://blog.squix.org/2015/03/esp8266-module-comparison-esp-01-esp-05.html>. [Accessed 1 February 2020].
- [10] JaiS45, "Arduino UNO + ESP8266 WiFi Module," [Online]. Available: https://www.instructables.com/id/Arduino-UNO-ESP8266-WiFi-Module/?fbclid=IwAR0qiX2irywU37d0uC2NXhNXW--rvTZBNaEK5YMVN_DdunErXUaLPtCgiEQ. [Accessed 3 February 2020].
- [11] Espressif Systems, "ESP8266 AT Instruction Set," Espressif Systems, 2019. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/4a-esp8266_at_instruction_set_en.pdf. [Accessed 3 February 2020].

- [12] F. Halsall, Computer Networking and The Internet, 5th ed., Harlow: Pearson Education Limited, 2005, pp. 438-487, 493-494.
- [13] P. L. Dordal, "An Introduction to Computer Networks," Chicago, 2020.
- [14] GeeksforGeeks, "Socket Programming in C#," [Online]. Available: <https://www.geeksforgeeks.org/socket-programming-in-c-sharp/>. [Accessed 7 February 2020].
- [15] H. R. Max Roser, "Technological Progress," 2020.
- [16] GOV.UK, "How copyright protects your work," [Online]. Available: <https://www.gov.uk/copyright>. [Accessed 22 December 2019].
- [17] GOV.UK, "Non-disclosure agreements," [Online]. Available: <https://www.gov.uk/government/publications/non-disclosure-agreements>. [Accessed 20 December 2019].
- [18] TermsFeed, "Sample EULA Template," 18 January 2020. [Online]. Available: <https://www.termsfeed.com/blog/sample-eula-template/>.
- [19] Google Play, "Google Play Developer Distribution Agreement," 5 November 2019. [Online]. Available: <https://play.google.com/about/developer-distribution-agreement.html>.
- [20] Google Play, "Privacy, Security, and Deception," [Online]. Available: https://play.google.com/about/privacy-security-deception/user-data/#!?zippy_activeEl=personal-sensitive#personal-sensitive. [Accessed 20 February 2020].

Appendix

Reflection

Taking part in enterprise and innovation project was an excellent opportunity to develop practical skills and gain an insight into the IT industry as well as a structure of a business. My group decided to create an application, which would mix the functionality of a social-media app and a game. What is more, the project aims to serve a general good – promoting a healthy lifestyle by making it fun. As a game developer student, I was intrigued about making a game. However, I wondered about my role in a team. It was apparent that I did not want to be in charge of something that had been already known for me because this would not make progress in my personal development. Therefore, the assigned role was about achieving new skills in IT and electronics and gathering knowledge about the legal aspects of the functionality of enterprises. These fields of science were rather new to me, except for software engineering, hence the given tasks were found out both fascinating and challengeable.

The project allowed me to study microcontrollers and Arduino programming, which I realised is something interesting. I had to learn the essentials for microcontrollers and electricity. Moreover, I have soldered for the first time and discovered the opportunities Arduino devices give.

When I had understood the Arduino environment, my next aim was to study the Internet technology with the application to Unity and Arduino. The theory was not as challenging as the practical part, because there are not many documentation and project of using Arduino device with Unity over the Wi-Fi connection, so I was forced to perform various experiments. Finally least the link has been established.

Consequently, I figured out that it is possible to create various Arduino devices which will be compatible with games, or even part of the gameplay. Moreover, I learnt the basics of networking structures of games and this knowledge might happen to be useful in other projects of mine. Finally, the Arduino programming language is a new programming language I could attach to my CV.

Due to carried out research making games is not only about coding and designing but also about doing business, using available tools and being aware of rights and privileges. Thanks to the research, I found out what is the process of software distribution combined with and what is essential to know when you develop a fully legal application. I was managed to get a cleared view on how to get into business and how law can protect you and your ideas. I see the gathered

knowledge useful as the traditional Roman principle states “ignorantia iuris nocet” (not knowing the law is harmful).

Networking

To develop my personal skills, I decided to experience a few different ways – an industry event, internet class courses and YouTube webinars.

The event I attended was a talk on virtual production by Shaun Varney, which took place at the University. Thereby, the lecture was about applying computer technology such as games engines, for video production. I found that interesting that games engines are used not only for games but also to make visual effects in movies. Thanks to the shared experience of Shaun Varney, my point of view into the games industry got much broader in the aspect of using the technology. Namely, it was noticed that games developers may find a job in the film industry and use own skills innovatively.

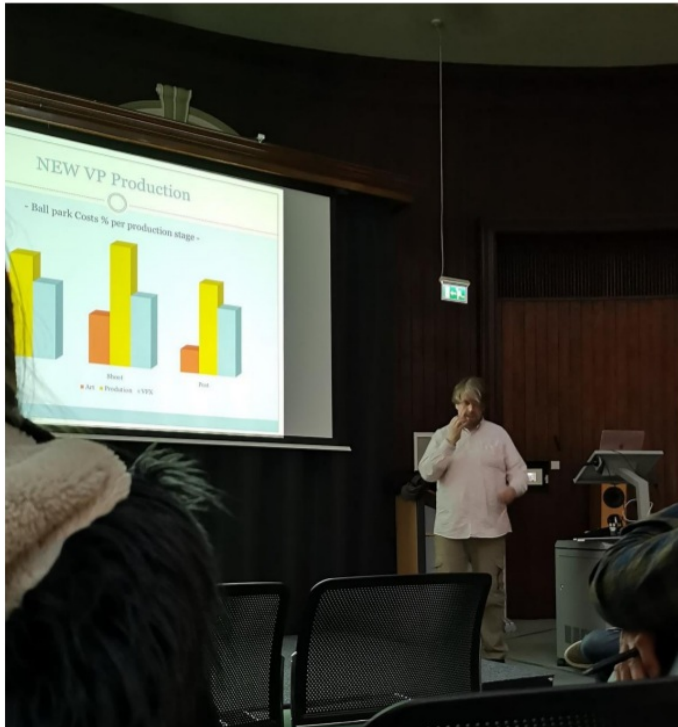




Fig.18 The photos were taken during the lecture of Shaun Varney.

The other form of developing myself was joining the community of “GameDev.tv”. It is a passionate group of instructors helping to begin a game development, with online courses. Furthermore, the classes are on a high level of standard and they offer high-quality materials to learn. Thus, I bought Unity 2D course on Udemy making by Rick Davidson and Ben Tristem as a “GameDev.tv” team. It was the right choice because the course teaches by practice and its construction is thought out. The tutors give many examples, attachments and inspirations for students. In case of any issue with the study, it is possible to contact with the tutor or ask the broad community of students. I treat this experience as additional hours spend on games programming out of the related university course.

Complete C# Unity Developer 2D: Learn to Code Making Games

Game development & design. Learn Unity 2018 in C#. Your first 7 2D Unity games for web, Mac & PC. Includes Tilemap

BESTSELLER ★★★★★ 4.7 (70,188 ratings) 336,651 students enrolled

Created by Ben Tristem, GameDev.tv Team, Rick Davidson Last updated 2/2020

English English, French [Auto-generated], 7 more

What you'll learn

- ✓ Learn C#, a powerful modern language, from scratch. No prior programming experience is necessary.
- ✓ Build a solid foundation for game design and game development that will help you build your own games.
- ✓ Create playable game projects - good for your portfolio, or just for your own sense of achievement.
- ✓ Become excellent at using the Unity game engine.
- ✓ Learn how object oriented programming works in practice.
- ✓ Transfer your knowledge from this course to .NET, other languages, and more.

Download high-quality resources for this course *The path of an aspiring and successful dev*

Preview this course

You purchased this course on Oct. 28, 2018

Go to course

Share this course

30-Day Money-Back Guarantee

This course includes

- 35 hours on-demand video
- 3 articles
- 27 downloadable resources
- Full lifetime access
- Access on mobile and TV
- Certificate of Completion

Fig.19 The main page of the course (<https://www.udemy.com/course/unitycourse/>).

 Your progress ▾

Share

i

Course content

×

Section 1: Introduction & Setup

8 / 8 | 32min

▾

Section 2: Number Wizard - Basic C# Coding

10 / 10 | 1hr 16min

▾

Section 3: Text101

18 / 18 | 2hr 53min

▾

Section 4: Number Wizard UI

13 / 13 | 1hr 41min

▾

Section 5: Block Breaker

45 / 45 | 6hr 15min

▾

Section 6: Laser Defender

41 / 41 | 5hr 58min

▾

Section 7: Glitch Garden

68 / 68 | 10hr 24min

▾

Section 8: TileVania - 2D Tilemap Platformer

11 / 38 | 5hr 44min

▴

☒ 177. TileVania - A Bonus Section For You!

1min

☒ 178. Welcome To TileVania

3min

Resources ▾

☒ 179. TileVania Game Design

6min

Resources ▾

☒ 180. Slicing Sprite Sheets

6min

Resources ▾

☒ 181. Intro To Unity Tilemap

15min

Resources ▾

☒ 182. Unity Rule Tiles

12min

Resources ▾

☐ 183. Create Player Idle Animation

10min

Resources ▾

☐ 184. Animation States & Transitions

Fig.20 The figure is showing the sections already covered in the course and actual stage of learning.

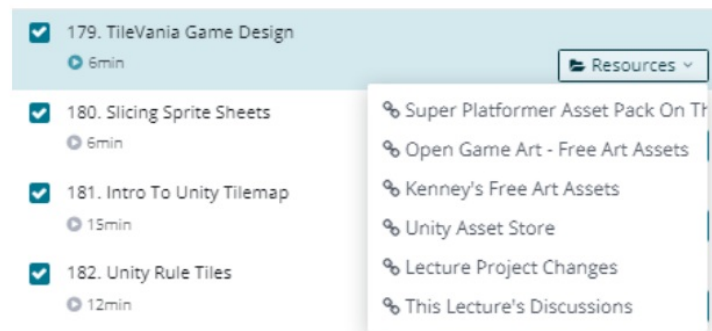


Fig.21 The screenshot of available additional resources attached by the tutor for each lesson.

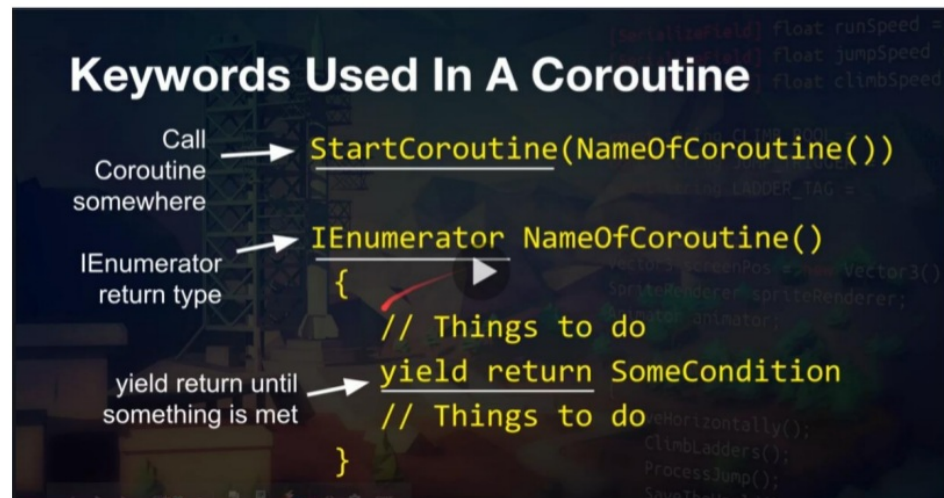


Fig.22 The example of material explaining an issue in a course video lesson.

Next form of gathering additional information and developing myself as a future games programmer was watching videos and reading magazines regarding my field of study. I started watching video tutorials on YouTube, especially “Brackeys” and “Code Monkey” channels inspired me to start doing my own mini-games. However, I was getting more interested in advanced aspects of programming and games development, so I subscribed “Ask Gamedev” and Comp-3 Interactive” channels, which professionally explain potentiality of Unity games engine and games industry.

Furthermore, I collect “MCV Develop” and “Wireframe” magazines. They contain useful information about the games industry and games development, such as articles about the upcoming commercial and business events, practical tutorials or job offers.

Unity Basics for Beginners ▶ PLAY ALL
Learn the complete basics of Unity to get you up and running fast.

GAME OBJECT?

What is a GameObject (Unity Tutorial for Beginners)

Code Monkey

25K views • 1 year ago

TRANSFORM?

What is a Transform (Unity Tutorial for Beginners)

Code Monkey

15K views • 1 year ago

TRANSFORM?

What is a Transform: Parents and Children (Unity Tutorial...)

Code Monkey

14K views • 1 year ago

SPRITE RENDERER?

What is a SpriteRenderer? (Unity Tutorial for Beginners)

Code Monkey

12K views • 1 year ago

COMPONENT?

What is a Component? (Unity Tutorial for Beginners)

Code Monkey

4.2K views • 1 year ago

Platform Dependent Compilation

Platform Dependent Compilation Statements...

93 views • 2 days ago

Pixel Art Speed Paint

Pixel Art Speed Painting in Aseprite - Pastel Islands

163 views • 4 days ago

[Attributes]

Class Based Attributes [Unity Tutorial]

241 views • 6 days ago

Beginners Guide To Data Encryption

A Beginners Guide to Data Encryption [Unity Tutorial]

238 views • 1 week ago

Saving Data Using JSON

Saving Data using JSON Serialization [Unity Tutorial]

350 views • 1 week ago

FIRST PERSON MOVEMENT in Unity - FPS Controller

374K views • 4 months ago

EVERYTHING COMING TO UNITY IN 2020!

196K views • 5 months ago

UNITE KEYNOTE

Unite Copenhagen 2019 Keynote on all the new...

36K views • 5 months ago

MAKING YOUR FIRST LEVEL in Unity with ProBuilder!

144K views • 5 months ago

TOP DOWN SHOOTING in Unity

173K views • 6 months ago

GAME DEV REACTS

Game Developer Breaks Down Graphics in Video...

114K views • 7 months ago

TOP DOWN MOVEMENT in Unity!

200K views • 7 months ago

10 YEARS OF MAKING VIDEO GAMES

What I learned after 10 Years of Game Development!

563K views • 7 months ago

How to make Terrain in Unity!

333K views • 8 months ago

2D Lights in Unity!

185K views • 9 months ago

WILL UNITY BREAK?

HOW MANY PARTICLES CAN UNITY HANDLE?

90K views • 9 months ago

ENEMY IN 2D

2D PATHFINDING - Enemy AI in Unity

174K views • 9 months ago

Fig.23 The examples of watched youtube channels and videos
(<https://www.youtube.com/user/Brackeys/videos>,
<https://www.youtube.com/channel/UC26kmK523wCy9RziFrvhp7g>,
https://www.youtube.com/channel/UCFK6NCbuCIVzA6Yj1G_ZqCg).



Fig.24 The current collection of games development magazines.

The last form of development my skills is webinars. The recent webinar, which I watched was about mathematics for games developments by Freya Holmers. She has been an experienced worker in the games industry for ten years and she commenced to make educational videos about mathematics, programming and Unity tools on various platforms. In the video called “Math for Game Dev - An Improvised Live Course,” she explains the most essential mathematics used by games such as scaling and ratios, vectors and dot product, trigonometry or local and world space transformation. I noticed this webinar practical because it would be applied to the improvement of my own projects in the future.

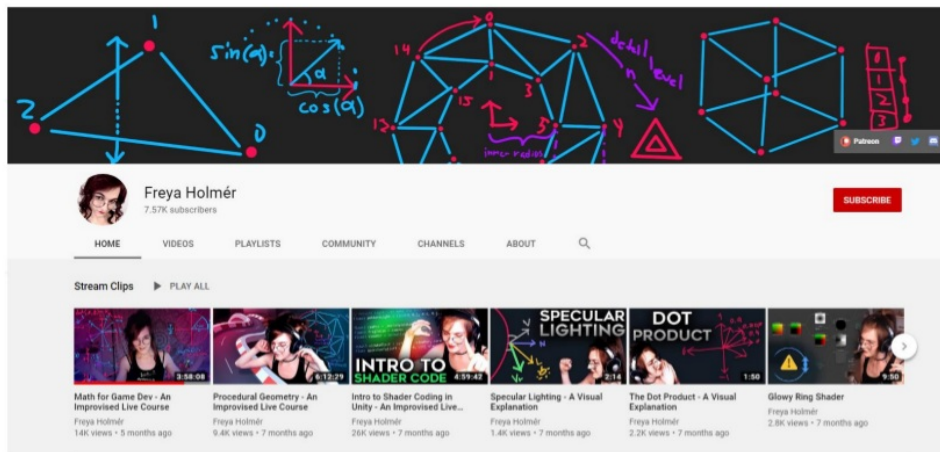


Fig.25 The YouTube channel of Freya Holmer (<https://www.youtube.com/channel/UC7M-Wz4zK8oikt6ATcoTwBA>).

pxWidthBar

4 m/s = 0.25 s/m

pxWidthBar * pctHealth

Scale =

pxWidthBar * pctHealth * pxPerSprite

healthCurrent

healthMax

pxSpriteWidth

$\frac{px}{spx} = \frac{px}{spx}$

$\frac{1}{x}$

pxPerSprite

0.25 spx/px

pctHealth = $\frac{Current}{Total} = 0.4 = 40\%$

try to find the sprite?

pxPerSprite

0.25 spx/px

$\sqrt{x^2 + y^2} = Length$

$\sqrt{(A_x - B_x)^2 + (A_y - B_y)^2}$

$\frac{A - B}{length(A - B)} = direction(B \text{ to } A)$

I've always had trouble figuring how to easily track an object in another game object (like a player and a floating object)

17:24 / 3:58:07

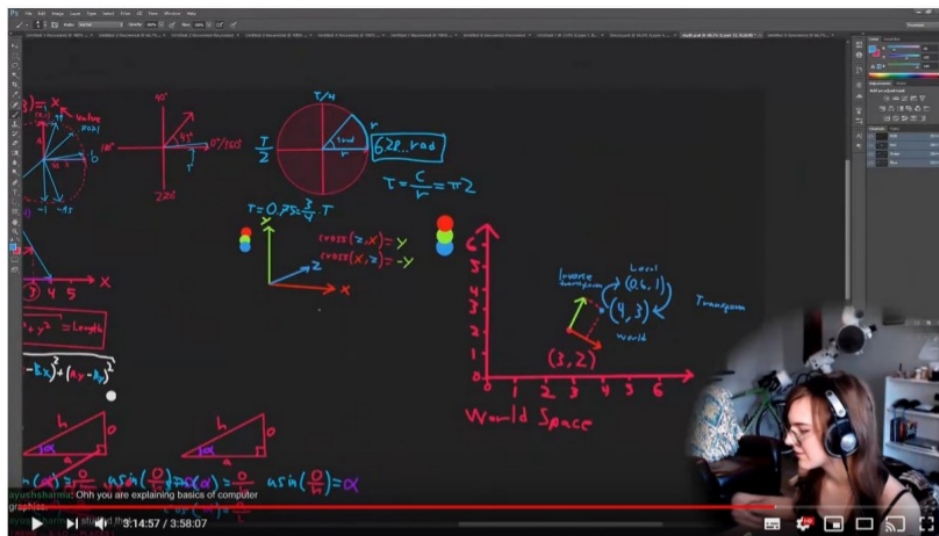


Fig.26 The screenshots of the webinar by Freya Holmers (<https://www.youtube.com/watch?v=yg6h4XQqPNQ>).

----- **TEAM PROGRESS SHEET** -----

Team Name:

Team Member Names & Roles:

RL - Rhys Lewis - Team Leader / Audio Technician / Actor

DD - Daniel O'Dwyer - Graphics / Animations Designer

KL - Kyle Lovell - Chief Coder/Programmer

AH - Adam Horton - Weighing Scale Designer / Builder

MK - Mariusz Kalasz - Weighing Scale Designer / Builder

BK - Ben Kiddie - Social Network Integration

BG - Bartosz Galantowicz - Weighing Scale to Mobile App Connection

Weekly Progress report:

Week 3

Date: 13/11/19

In Attendance: RL, DD, KL, AH, MK, BK, BG.

Apologies: N/A.

Absent: N/A.

Team Goals:

Complete PEST Analysis, Competitor analysis, email potential partners and finalise equipment needed for scales.

> Individual Updates:

RL - Completed survey

DD - Designed game menus

KL - Completed a Minimal Viable Product document

AH - Began writing component list for scale

MK - Began writing component list for scale

BK - Completed an Initial Requirements Specification document

BG - Completed the networking technology system research

> Team discussion:

Discussing the beginning of product development and app features.

> Individual Actions For Upcoming Week:

RL - Email potential partners, discuss recipes with Dan.

DD - Complete Competitor analysis, discuss recipes with Rhys.

KL - T+Ec of PESTLEc analysis

AH - Complete component list

MK - Complete component list

BK - Work on environmental and social research for PESTLE analysis

BG - Political + Legal - PESTLE analysis

11th December 2019

TEAM PROGRESS SHEET

Team Name:

Team Member Names & Roles:

RL - Rhys Lewis - Team Leader / Audio Technician / Actor

DD - Daniel O'Dwyer - Graphics / Animations Designer

KL - Kyle Lovell - Chief Coder/Programmer

AH - Adam Horton - Weighing Scale Designer / Builder

MK - Mariusz Kalasz - Weighing Scale Designer / Builder

BK - Ben Kiddie - Social Network Integration

BG - Bartosz Galantowicz - Weighing Scale to Mobile App Connection

Weekly Progress report:

Week 7:

Date: 11/12/2019

In Attendance: BK, RL, MK, AH.

Apologies: N/A

Absent: BG, DD, KL.

Team Goals:

> Individual Updates:

RL - Planned for beginning of production

DD - Supplied kyle with assets for game menus

KL - *No information given*

AH - reviewed the progression of the parts being delivered and began forming an initial start-up of the build of the weighing scale.

MK - Ordering of the final part

BK - Outlined steps to research on social media integration of Facebook

BG - Research on Arduino

> Individual Actions For Upcoming Week:

RL - Write script with Dan and record Voiceover for character voice

DD - Write script with Rhys

KL - *Requires further discussion*

AH - Begin the build of the weighing scale and ensure we as a team meet the correct targets at the given times or before hand for additional testing and earlier completion.

MK - Begin the build and coding of the scale

BK - Outline steps for implementation of other social media

BG - Creating a prototype for network connection tests

26th February 2020

----- **TEAM PROGRESS SHEET** -----

Team Name:

Team Member Names & Roles:

RL - Rhys Lewis - Team Leader / Audio Technician / Actor

DD - Daniel O'Dwyer - Graphics / Animations Designer

KL - Kyle Lovell - Chief Coder/Programmer

AH - Adam Horton - Weighing Scale Designer / Builder

MK - Mariusz Kalasz - Weighing Scale Designer / Builder

BK - Ben Kiddie - Social Network Integration

BG - Bartosz Galantowicz - Weighing Scale to Mobile App Connection

Weekly Progress report:

Week 19:

Date: 26/2/2020

In Attendance: BK, MK

Apologies: RL, DD, KL, AH, BG

Absent: N/A

Team Goals:

> Individual Updates:

RL - Global Assistance

DD - Worked on Implementing Camera function

KL - Continued app development

AH - Continued construction

MK - Continued construction

BK - Continued development of app

BG - Continued Research methods

> Individual Actions For Upcoming Week:

RL - Global Assistance

DD - Implement Camera function

KL - Continue app development

AH - Continue construction

MK - Continue construction

BK - Continue app dev

BG - Continue Research methods

ESP8266 connection and using AT commands.

ESP8266 ESP-01 is a Wi-Fi module that allows microcontrollers access to a Wi-Fi network. It is possible to program the module to act as a microcontroller as well. ESP9266 has a pre-installed AT firmware, but its chip can be programmed with another firmware such as NodeMCU.

The ESP8266 module works on three operation modes:

Access Point (AP)

Station (STA)

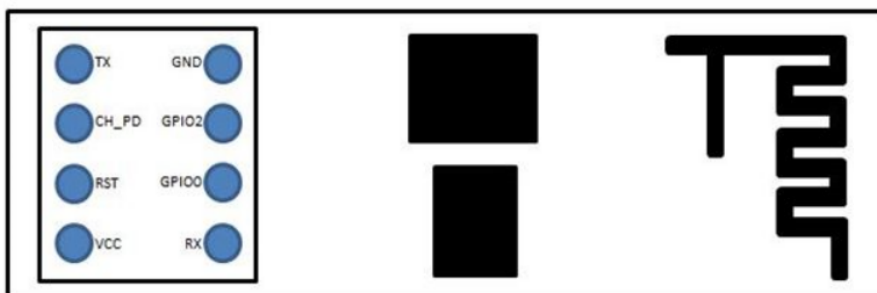
Both

In AP, the Wi-Fi module acts as a Wi-Fi network, or access point, allowing other devices to connect to it. It simply establishes two-way communication between the ESP8266 and the device that is linked to it via Wi-Fi.

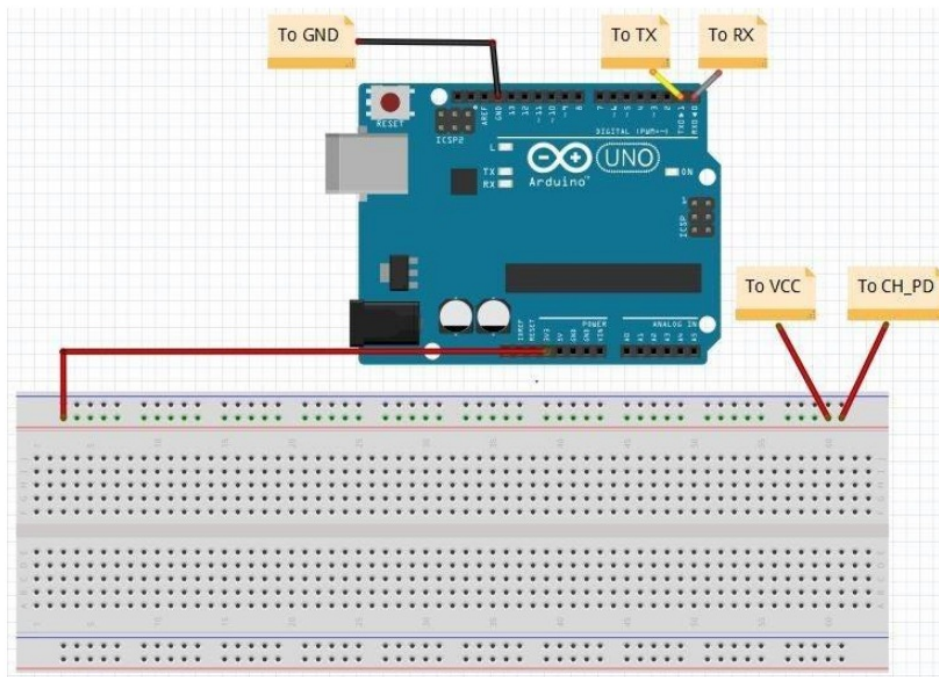
In STA mode, the ESP8266 can connect to an AP such as the Wi-Fi network form your house. This allows any device connected to that network to communicate with the module/

The Both mode of operation permits the module to act as both an AP and an STA.

1. Setup - Hardware



ESP8266 ESP-01 module pins allocation.



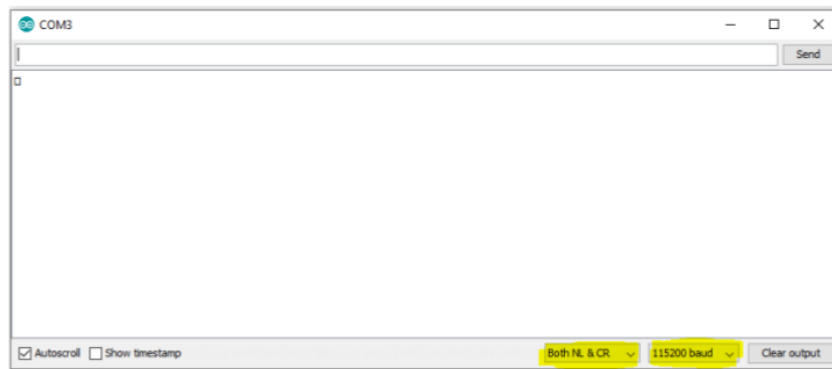
Arduino UNO basing set up for the connection with ESP8266 ESP-01.

ESP8266 ESP-01	Arduino UNO
TX	Pin1
RX	Pin 0
CH_PD	3.3 V
RST	Unwired
VCC	3.3 V
GND	GND
GPIO2	Unwired
GPIO0	Unwired

The connection pattern for Arduino Uno and ESP8266 ESP-01.

2. Setup – Software

- 2.1. Upload the BareMinimum (basic Arduino sketch) and run the serial communication channel. Use 115200 baud module and set Both NL.



2.2. Type the following command in the serial monitor:



The “ok” states that the connection can be established.

2.3. The following command is used to set the operation mode:

```
AT+CWMODE=1
```

The corresponding number for each mode of operation is:

STA = 1

AP = 2

Both = 3

Use the following command to check the current module of operation:

```
AT+CWMODE?
+CWMODE:1
```

2.4. If the ESP8266 works in STA mode, it is possible to connect to a Wi-Fi network.

Firstly, the current connection is checked by the following command:

```
AT+CIFSR
+CIFSR:STAIP,"192.168.1.9"
+CIFSR:STAMAC,"cc:50:e3:7d:17:60"
```

This command displays the station IP address of the module. To establish an IP address of the device, use the following command:

```
AT+CWJAP= "Wi-FiNetwork", "Password"
```

Type the name of the network and the correct password in quotations marks.

2.5. Enabling connection is required before the configuration of the module as a server.

Use this command to allow the connection:

```
AT+CIPMUX=1
```

Once again, each number is associated with a type of connection:

Single = 0

Multiple = 1

2.6. Now, the next step is to start the server at port 80:

```
AT+CIPSERVER=1,80
```

The first number is used to indicate whether to close server mode (0), or open server mode (1). The second number indicates the port that the client uses to connect to a server. Port 80 is chosen because this is the default port for HTTP protocol.

2.7. Typing the IP address of the module into the web browser causes a response. This is HTTP request that the computer sends to the server to fetch a file, it contains some data about the operation.

2.8. To send and display data it in the web browser's window use:

```
AT+CIPSEND=0,5
```

The "0" indicates the channel through which the data is going to be transferred, while "5" represents the number of characters, which are going to be sent. However, nothing appears on the web browser's window until the channel is going to be closed. Closing the channel is set by the command:

```
AT+CIPCLOSE=0
```

"0" indicates the channel that is being closed.

The entered message is displayed on the browser's window now.

3. Telnet

Checking the connection works is made by using the android application “Mobile Telnet”. In the app, go to telnet settings and type the IP address and the port number used by the ESP8266 and then click connect.

Send a message through the application to check if the module receives the data. The output should be printed on the serial monitor.

<https://www.instructables.com/id/Getting-Started-With-the-ESP8266-ESP-01/>

FINAL GRADE

GENERAL COMMENTS

Instructor

97 /100

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

PAGE 37

PAGE 38

PAGE 39

PAGE 40

PAGE 41

PAGE 42

PAGE 43

BARTOSZ
GALANTOWICZ

97

RESEARCH

Engagement with Literature Skills.



Citations: 20. Excellent. References: 20. Well-formatted relevant and up-to-date. Excellent.

20

KNOWLEDGE

Knowledge and Understanding Skills.



+ Hardware and software integration. + Legal considerations. + PESTLE Analysis. + Technical skills learned for project and additional skills learned during and as a result of the process.

20

ANALYSIS

Cognitive and Intellectual Skills.



+ Analysis of development process and problem solving. + Professional legal and ethical considerations.

18

PRACTICAL

Practical Application Skills.



+ Development process. Very Well documented.
Excellent.

20

TRANSFERABLE SKILLS

Transferable Skills for Life and Professional Practice.



+ Report format structure and presentation. Very good. – section numbers could be used. - Over the word count. + Reflection. Excellent. + Networking Excellent. + Team progress sheets and contribution.

19