ELECTRIC AIRPLANE CONTROLLED WITH THE MOBILE APPLICATION

A graduation project submitted to
the Faculty of Engineering and Natural Sciences of
International University of Sarajevo in partial
fulfillment of the requirements for the degree of
Bachelor of Science
in Computer Sciences and Engineering

by
Omar Hassan

June, 2019

Graduation project written by

Omar Hassan

Graduation Committee

Prof. Dr. Izudin Džafić	IUS, B&H, Supervisor
Prof. Dr. Sencer Yeralan	IUS, B&H

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Omar Hassan

INTERNATIONAL UNIVERSITY OF SARAJEVO

DECLARATION OF COPYRIGHT AND AFFIRMATION

OF FAIR USE OF UNPUBLISHED WORK

ELECTRIC AIRPLANE CONTROLLED WITH THE MOBILE APPLICATION

Affirmed by Omar Hassan

Signature Date

Abstract

This project is a result of more than 3 months of hardworking and learning process to different engineering fields. a lot of electrical engineering and computer science was needed to complete this project. The final product is an airplane and a mobile application to control the airplane. The biggest problem we faced is time, we just needed a little more time to make things more perfect.

Contents

List of Figures

List of Tables

List of Abbreviations

A	Ampere
Ah	Ampere-hour
BEC	Battery Eliminatory Circuit
BLDC	Brushless DC
DC	Direct Current
ESC	Electronic Speed Controller
LED	Light Emitting Diode
S	Cells in series
V	Volt

Acknowledgements

This project was done with the help of my colleagues Himzo Hasak & Bakir Brkić.

Omar Hassan

June 2019, Sarajevo, Bosnia and Herzegovina

Part I
Introduction

Our idea is to build a medium size airplane and a control software. Plane brain will be a microcontroller which will be programed to control output devices (mainly motors) on the airplane based on the input signal from the remote-control mobile application and different information from the sensors attached to the airplane itself. the project is a big project on which we did not have much prior knowledge. We did not have any guideline to follow instead we had to think good and research before taking any decision. We have faced many problems throwout the process of building the plane but we managed to overcome. it was really joy-full trip full of knowledge and challenges. I am very proud of the job we finished and we are going to continue with project even after we graduate.

Part II
Components

Name	Operating DC Voltage [V]	Current [A]	Description
Active Buzzer	5	-	
Antenna	-	-	
Arduino Mega	5	0.1	Secondary microcontroller
Battery	11.1	100	LiPo, Turnigy, 3S, 5 Ah
BME280 Sensor	5	-	Temperature, pressure, humidity
Boost Converter	5-24	0.1	
Capacitor	-	-	
Charger	220 AC, 3.7	3x0.8	Balancing LiPo
ACS712 Current meter	3.3-5	5, 20, 30	
Diode	-	-	
DS18B20 Temperature sensor	5	-	1-wire protocol
ESC	7.4-11.1	30	5 V BEC
ESP Camera	3.3	0.1	
Fan	5	0.1	BLDC
NEO-6M GPS	5	0.1	
MPU5060	5	-	6-axis MPU
Inductor	-	-	
IRF830 MOSFET	5	0.2	
LED	3 (24)	0.02	
LED Flash	12	0.2	
Motor	0-11.1	10	Out-runner BLDC
NodeMCU	3.3	0.1	
Power Wire	400	50	
Propeller	-	-	10x4.5
Resistor	-	-	
Servo Motor	5	1	
HC-SR04 Ultrasound sensor	5	-	Proximity: 2 cm - 400 cm
Switch	400	40	
Wire	400	5	
XT60 Connector	400	100	

Part III
Construction

1 Hardware

1.1 Body

First step in realization of our project was making the airplane body out of Styrofoam. The body is 1.8 m long and wing span is 1.6 m. It is built on principles that are used while building RC model airplanes. The design and proportions of the body loosely resemble Boing 747 airliner. Base is made of thicker Styrofoam

reinforced by bamboo sticks and had chassis made out of wood with letter “V” shape placed below the wings. The body is 18 cm in diameter made by 1 cm thick Styrofoam slices 1 cm and 2 cm wide glued by hot glue at some spots and held by arc base every 30 cm. Wings are 2 cm thick attached to the body with the angle of 4° with respect to body axis. Wings are also angled with respect to ground.

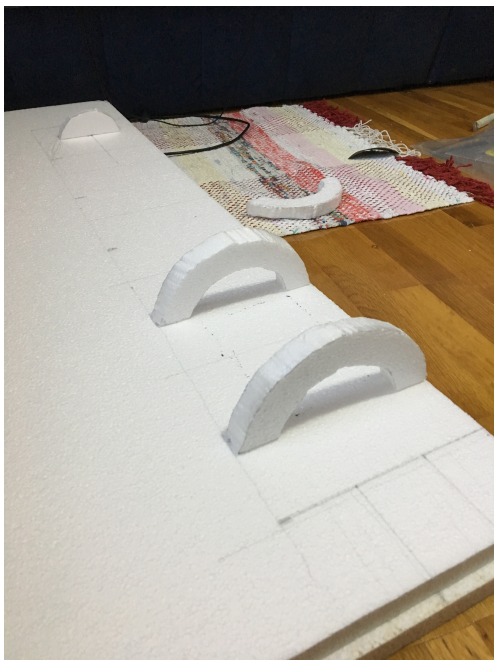

Figure 1: Making of the body

Figure 2: Making of the wings

1.2 Mechanical parts

Next, we inserted mechanical parts of the wings (ailerons, rudder, elevator). Each of them is attached to servo motor which will be controlled by the microcontroller. They rotate either on bamboo sticks or on the screws and nuts. After that, we attached motors to the wings by thicker bamboo sticks and hot glue. There are two of them on each wing separated 25 cm. We used 1000 kv brushless outrunner motors, 120 W each. Each motor has to have Electronic Speed Controller to drive it. It converts DC from battery to three- phase AC going into the motors.

Figure 3: Chases

Figure 4: “Tail”

1.3 Power

Every device needs power. Our airplane is electric-powered.

1.3.1 Battery

The main power of the airplane is of course – battery. It is three cell, 11.1 V, 5 Ah, Lithium-Polymer that can output huge amount of current (100 A continuous). We choose that one because it has just enough voltage to power all our devices which are 3S compatible. With full charge and with full load constantly (even though full load is happening just with quick taking off maneuvers) it can provide, in theory, 7 minutes and 30 seconds of autonomy. In practice, it is 8 minutes and 30 seconds due to battery manufacturer capacity label policy. Battery is connected via master switch to splitter. It is also planned to put current meter here but there is not cheap solution for that yet since current meters can handle up to 30 A and even that can burn them easily.

Figure 5: Battery

1.3.2 Cables and Connectors

Wire from battery is very thick, for large currents but even with that, we have voltage drop on maximum consumption of 40 A. Splitter splits power from battery to four motors. Here are current meters since maximum motor current is around 10 A. For connectors, we used XT60 because they are robust, easy-to-handle, popular and can handle a lot of current. Wire from splitter via current meter to motors is not as thick as battery one.

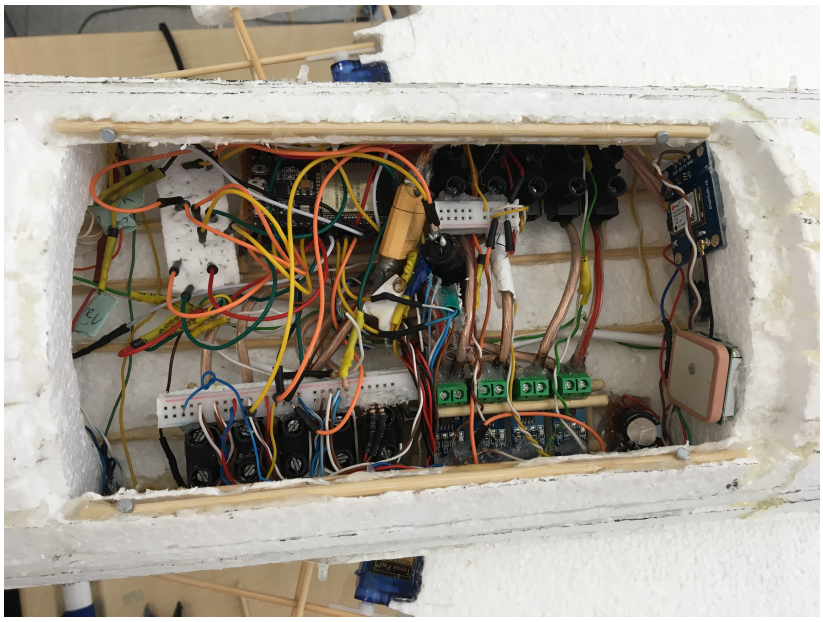

Figure 6: Main cabin

1.3.3 Electronic Speed Controller

Before motors and after splitters and current meter, there are Electronic Speed Controllers. ESCs are powerful electronic devices that have main two features:

1. Converting DC from battery to three-phase quasi AC for motors
2. Providing 5 V to all circuitry

First feature is simple: take PWM signal from microcontroller and proportionally let current from battery to motors. Period of the PWM signal is 20 ms and the duty cycle is usually from 5% to 10% corresponding to 0%-100% power. It uses high-power MOSFETs to produce three-phase AC by, again, PWM.
Second feature is very helpful. It provides 5 V with 3 A each for whatever you need to get rid of another circuitry just for lowering the voltage (buck converter, for example). This one is called Battery Eliminatory Circuit (BEC) and not all ECSs have it.

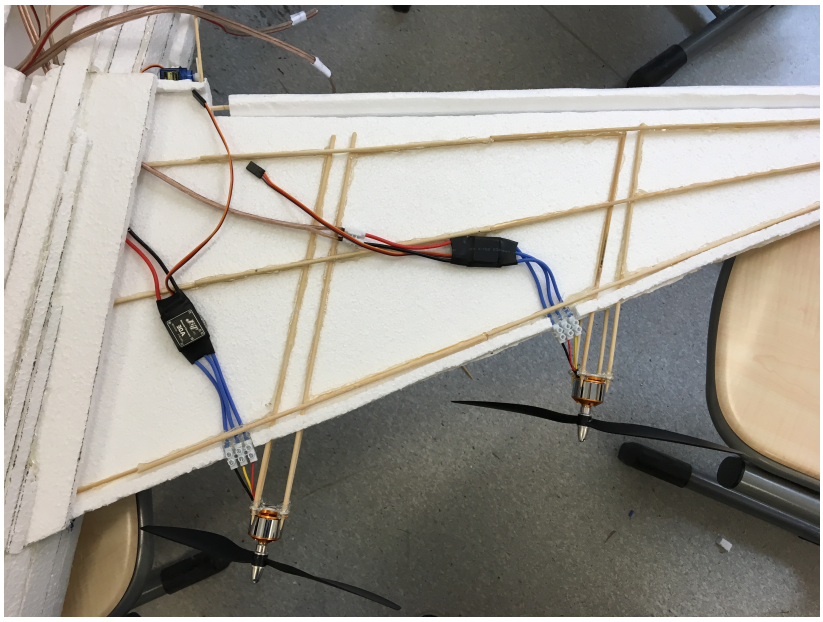

Figure 7: Electronic Speed Controllers

1.3.4 5 V & 3.3 V Supplies

Many sensors and devices in the airplane are 5 V powered. To ensure that every of them has enough power for itself, we make buses for 5 V and for 3.3 V. That bus is powered primarily from ECSs, but of master switch is not on, it can be powered with alternative sources such as USB either from PC or travel charger. Buses are not capable of handling large currents so we needed to put extra cables for current-demanding devices (more than 1 A) to keep other sensors voltage constant at 5 V.

1.3.5 Boost Converter

Making long series of LEDs we are making construction much easier and more power efficient. But, the trade-off is: we have to increase the voltage. To increase voltage we need boost converter which is basically device that takes low voltage and more current and converts it to higher voltage and lower current. Similar like a transformer for AC. Our boost converter takes 5 V DC and boosts it to 24 V for driving 8 LEDs in series together with resistor. It is not feedback controlled since it is made specially for this project.

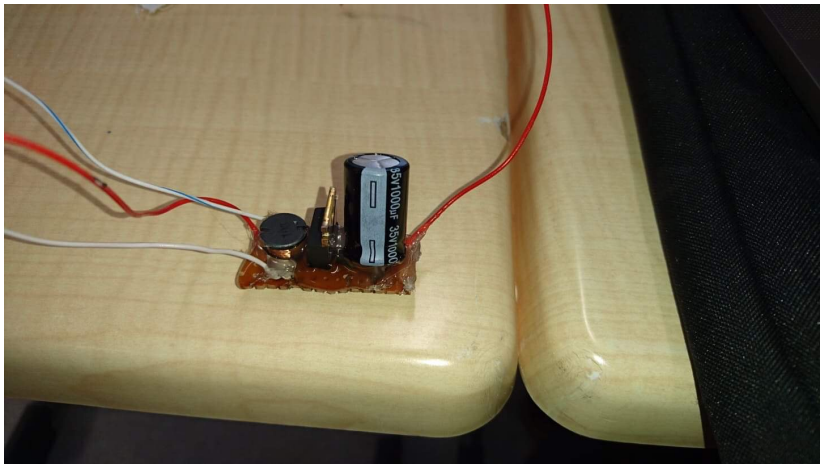

Figure 8: Boost converter

1.3.6 Charger

Battery is rechargeable and to charge it we need to connect it via another plug (not the main one for powering the airplane) to special charger. Charger has to be dedicated for LiPo batteries since there is procedure to charge those batteries. LiPo batteries should be charged to 4.2 V each cell and they have to be balanced all the time. Balancing means to keep their voltage as close to each other as possible. Maximum battery charging current is 5 A but our charger is 0.8 A. LEDs on the charger indicate status of each cell in the battery. Red is for charging and green is charged. To keep it as simple to use as possible, we put charger in the airplane and connect the battery permanently while putting socket for 220 V on the right side of the plane. 220 V wires are as short as possible and secured for safety.

In future versions of this airplane it is planned to have solar cells for prolonging flight time and inductive charging for even more simplicity.

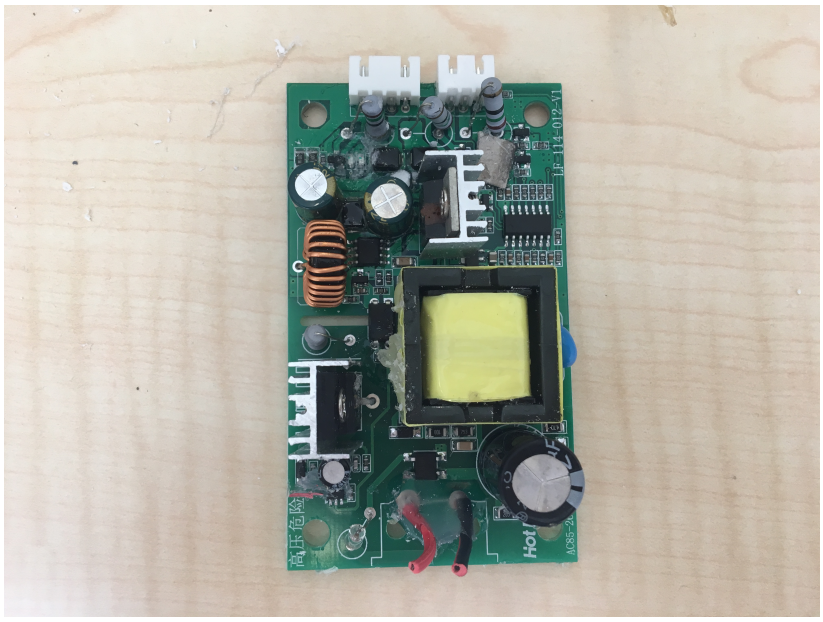

Figure 9: Charger

Figure 10: Charging port and LED indicator

1.4 Motors

1.4.1 Main Power Motors

The main power motors are outrunner brushless DC motors working on 2S-3S (7.4 V - 11.1 V) with maximum

current of 10 A. They work with three-phase AC current. As they are current-hungry, they get very hot so proper installation and cooling is required. Cooling is achieved by letting the airplane fly but installation on a heat insulation material, for not letting styrofoam melt, is mandatory.

For motors to do their job, they need propellers. We used 1045 for our airplane. 1045 means 10 inch in diameter with 4.5 pitch angle. Pitch angle is, in theory, how far would the propeller move forward with one revolution in a completely incompressible fluid.

Figure 11: Motor with propeller

Figure 12: Sideview of the motor installed on the main frame

1.4.2 Servo Motors

On this airplane, servo motors are used for tilting ailerons, elevators and rudder. Aileron is small section on rear end of a wing, used to control roll of the airplane. Roll is movement of the airplane on the y-axis. Elevators are small moving parts of rear part of an airplane. They are used for controlling the pitch of an airplane (x-axis movement). Lastly, rudder is vertical part of the "tail" of an airplane used to control yaw (z-axis).

Our servo motors have three wires: V_{cc}, Signal and Ground. V_{cc} is usually 5 V and Signal wire takes PWM input which respects to angle of shaft. 0°-180° corresponds to 5%-10% of duty cycle of the PWM signal. Signal is usually independent of the voltage.

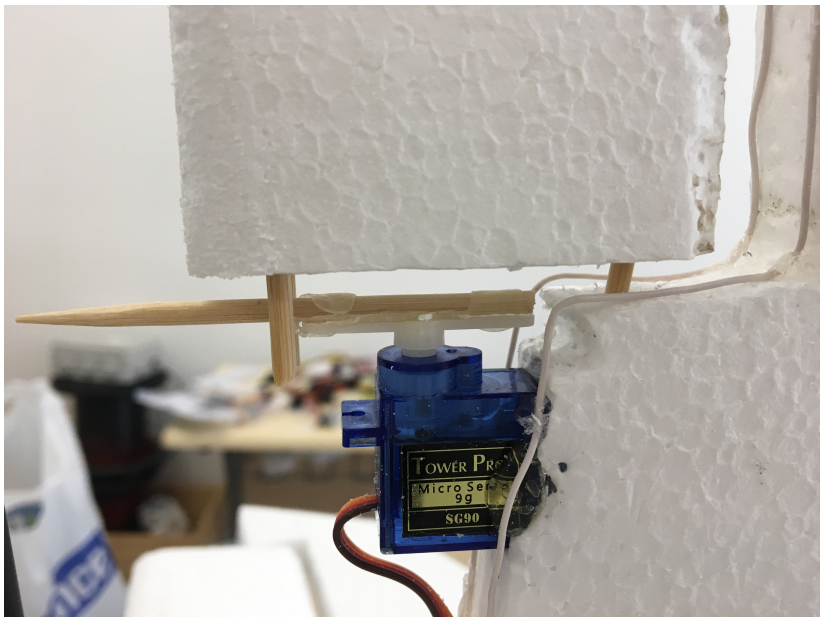

Figure 13: Servo motor controlling the rudder

1.5 Microcontrollers

1.5.1 NodeMCU-32

NodeMCU is a microcontroller based on Espressif's popular ESP32 chip featuring Wi-Fi and Bluetooth connection. Very powerful, dual core processor capable of many good super-arduino projects. Some of the benefits of this microcontroller is 16 PWM channels, 12-bit analog to digital conversion resolution, touch inputs, three serial communication ports. It is our main controller that receives commands from application and performing operations for the airplane. Interesting fact is that it has to be powered with 3.3 V so all inputs should be 3.3 V or less. All unnecessary parts are not connected to this microcontroller.

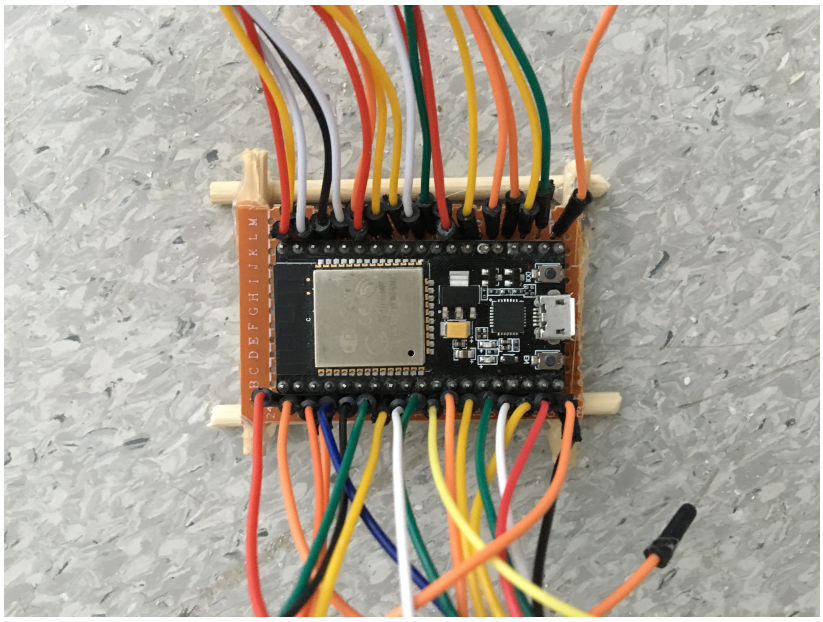

Figure 14: NodeMCU with soldered wires

1.5.2 Arduino Mega

Arduino Mega is very famous microcontroller which does not have to be explained too much. In our project it is used for all unnecessary things such as controlling LEDs or measuring air pressure. It is connected to our main microcontroller via serial communication protocol. It is powered by 5 V bus.

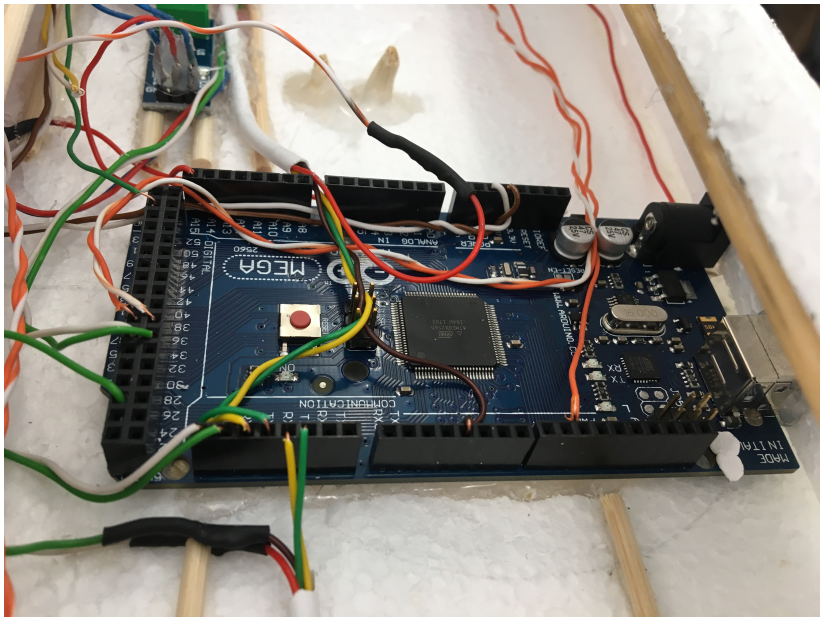

Figure 15: Arduino Mega with its connections

1.6 Sensors

1.6.1 Angle

To keep the airplane stabilized and to know in which direction is the airplane going, we have to use gyroscopic sensor which calculates everything for us. We used MPU6050 featuring 3-axis accelerometer, gyroscope and even temperature sensor. For now, we just use current angle as inputs to our airplane. Communication is achieved via I^2C protocol

1.6.2 Position

Measuring position with accelerometer is a nightmare, and almost impossible, so we decided to move on to the GPS sensor called NEO-6M.

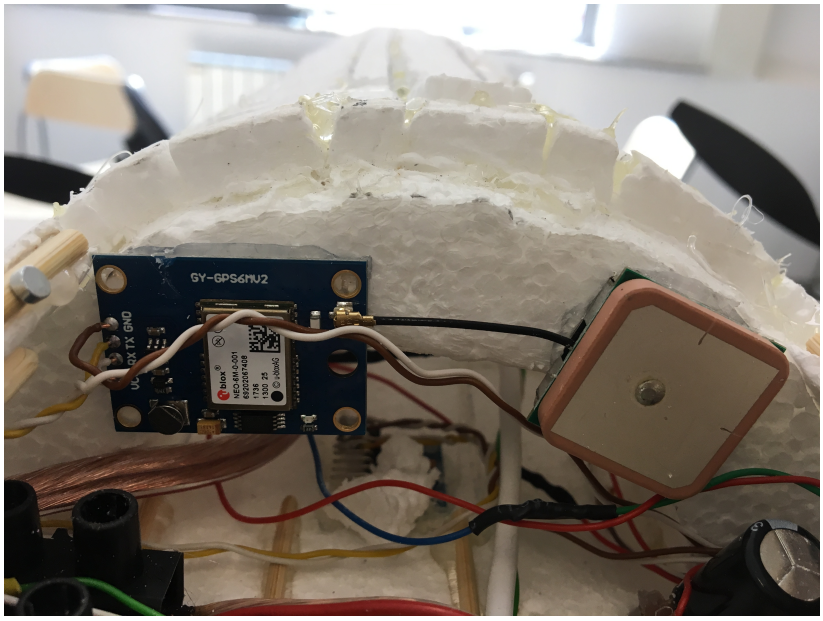

Figure 16: GPS and Gyro sensor

1.6.3 Voltage

For measuring the voltage we use NodeMCU's analog to digital converter with voltage divider. NodeMCU has 12-bit resolution so it sorts it to very accurate devices. The voltage is primary measured on the battery to know state of charge of the battery.

1.6.4 Temperature

1.6.5 Pressure

1.6.6 Humidity

1.6.7 Distance

1.6.8 Current

1.6.9 Light

1.6.10 Camera

2 Communication

In this project communication protocols play a very important role since there is a need of reliable and fast connection between several points of the system. First and most important connection is the connection between the main microcontroller in the airplane (ESP32) and the user operated web / mobile application, this connection has to be wireless and as fast as possible. The other connection is the connection between the ESP32 and the other microcontroller (Arduino Mega) that handles the parts and sensors that are not critical for the flight of the airplane. This second connection can be wired and it is not as time critical as the first one.

2.1 Web Socket

2.1.1 WebSocket Protocol introduction

Selection of the first connection's protocol and type was relatively easy because of our previous experience with the issues of connecting the embedded systems with remote servers or clients. We examined several options and two most obvious options were TCP Socket protocol and WebSocket protocol. TCP sockets are the golden standard for embedded systems communication when it needs to be real time, but their robustness also comes with a relatively complex way of using them. WebSockets on the other side are much newer protocol that got really popular in last several years due to its relative ease of use, while maintaining all the reliability and speed of traditional socket connection. In the end we settled down for the WebSocket protocol since it is much more suitable to use with JavaScript based web and mobile applications.

WebSocket protocol is an advanced technology that makes it possible to open a full duplex interactive communication session between the user's browser (web page) or mobile application (native or web view)

and a server. With the API provided and maintained by Mozilla foundation and w3c, you can send messages to a server and receive event-driven responses without having to poll the server for a reply.

2.1.2 WebSocket Protocol implementation

As we already mentioned the main parts that are connected through the webSocket protocol are Mobile / Web application and micro controller located in the airplane itself. The applications are developed using web technologies JavaScript and HTML and later repacked into the mobile application using Apache Cordova framework. That helps in rapid changes while testing and also it turned out to be a really good choice since the HTML5 and JavaScript support the webSocket protocol natively as part of [WebAPI]. The micro controller utilizes webSocket protocol through library for Arduino framework called WebSockets and developed and maintained by Marcus Sattler.

Microcontroller runs in WiFi Access point mode, and hosts the webSocket server, applications try to connect to it as soon as they are launched. When the connection is established and the socket is open the microcontroller starts streaming the sensor data to the connected applications. The sensor data is emitted every 200ms. It is helpful because it serves both as the real time diagnostics and telemetrics of the airplane but also from the practical case of keeping the socket open and alive even if there is no incoming data to the microcontroller.

Applications test the status of the socket every 200ms (to detect possible loss of connection) and process the data that they receive in real time. If the connection loss is detected they immediately try to reconnect to the server. Since the fallback code and connection loss handling is very robustly written, the end user can not notice any lag in communication, because reconnecting takes place in just a few milliseconds. Flight commands that are being sent from Application to microcontroller are not constantly streamed through the socket since there is is possibility to send too much unnecessary data that would slow down connection and provoke the microcontroller to flush the socket. To address this issue the application detects the difference between last sent flight command and new flight command in intervals of 100ms, and sends it to the microcontroller if needed. This ensures the realtime, robust and reliable connection between users application and airplanes microcontroller.

2.2 Serial communication

To solve the task of connecting two microcontrollers located in the airplane we utilized the well known serial communication that is available on both the boards. This simple way of connecting serial in/out pins of one microcontroller to out/in serial pins of other microcontroller allowed us to use already enabled features of Arduino framework to establish the connection between them. The communication itself is not very complex the Mega microcontroller reads the data from the sensors attached to it and sends it as a string through serial to ESP32 microcontroller on which there is a dedicated task running on it's dedicated core of microprocessor for receiving and processing the incoming sensor data. Multitasking and dual core capabilities are explained in more details in their dedicated section. It is used only to avoid possible lockdown or slowdown of ESP's microprocessor while waiting for the data from Arduino.

3 Software

This project contains not only hardware engineering part but also a big part of software engineering and programming. Software development related to this project can be separated in several parts. First and most obvious is programming of microcontrollers ESP-32 and Arduino Mega. But there are also other important parts of software mainly web and mobile applications used for telemetry, testing of individual components, testing of different scenarios and finally application that is meant to be used as the final tool of controlling and flying the airplane.

3.1 Mobile and Web Applications

Since all the data exchanged between the pilot (on the ground) and the airplane (in the air) goes through Wi-Fi connection and uses WebSocket protocol the most natural form of visualizing and generating that data was through creation of web applications. Later it was enhanced, and all these web applications got their mobile application counterparts. It was logical step forward since not all testing takes place in the laboratory and in front of the computer. Also piloting application needed to be portable from the very beginning since it is much more suitable to control the aircraft by gliding the fingers over the tablet or phone screen than by holding the bulky laptop.

Web application were developed by using JavaScript and HTML5 with several frameworks and libraries like

jQuery and Bootstrap, and all the networking features were utilized by using the existing Web API. Mobile applications are developed by reusing majority of the code needed for the web applications and combining it with Apache Cordova framework and its plug-ins.

3.2 Coding Telemetrics & Diagnostics Web Application

First application that we developed was for testing purposes. It went through several iterations since every new component added to the aircraft needed to be represented and tested through this application. In the first version it was used to test the stability of the connection between the user and the aircraft (this is addressed again in communication section). Next, we added the user interface needed to test and calibrate the motors, and servo motors.

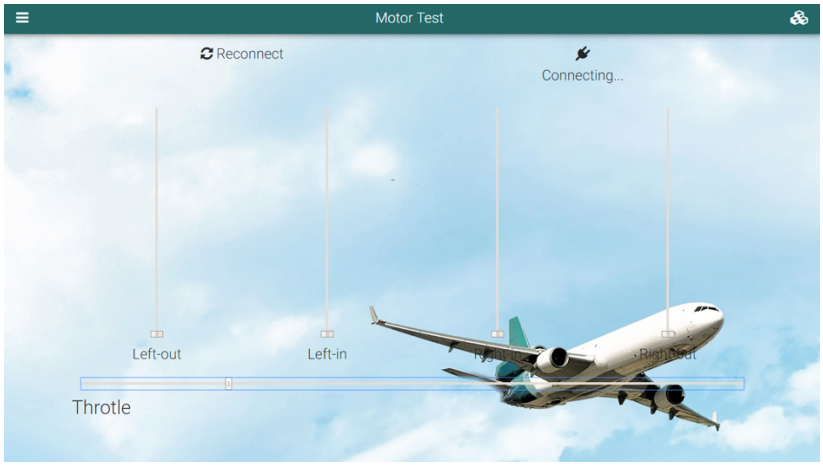

Figure 17: Initial verison of testing application

As we added more sensors, we also needed the way of monitoring the data coming in from the sensors, so we improved the application once more to be able to interpret the strings of data received from the airplane and visualize it in understandable way.

Figure 18: Incoming data string

Finally, we worked on improving the speed and reliability of the connection and optimized the amount of data that must be sent from the application to the aircraft. It also brought slight redesign to the application.

Figure 19: Final testing application

3.3 Pilot Application

Pilot application inherits all the features from the testing application, but some data is hidden from the user since it is not flight critical and it would only obstruct the screen real estate. The main difference is in the GUI design since the pilot application resembles the common design of RC Airplane joysticks.

Figure 20: Pilot application

Part IV

Conclusion

Working on a big project is a whole new experience, beside the knowledge you also need a good team and project management plan on order to have the best possible outcome. The realization of this project summarized a lot of things we learned during our university studies, and I think it is the first step to the real-word. I hope the project had fulfilled the requirements set by the university and the supervisor and that will be just the beginning to bigger steps in our lives.