Communication

***intro to comunication:***

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 secont connection can be wired and it is not as time critical as the first one.

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 commes 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.

***Websocketi u dušu:***

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.

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 incomming 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 contstantly 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.

**Serial**

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 establich 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.