

# Projeto Integrado Segundo Trimestre

**Turma: 31-TE**

## **Integrantes:**

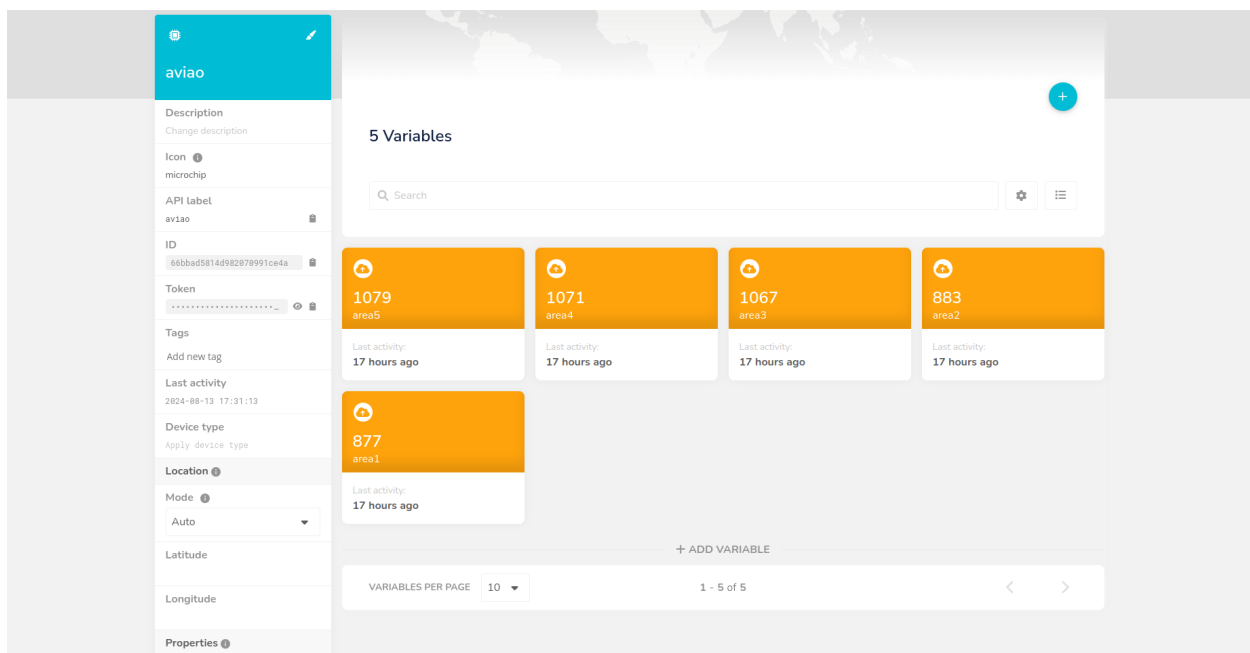
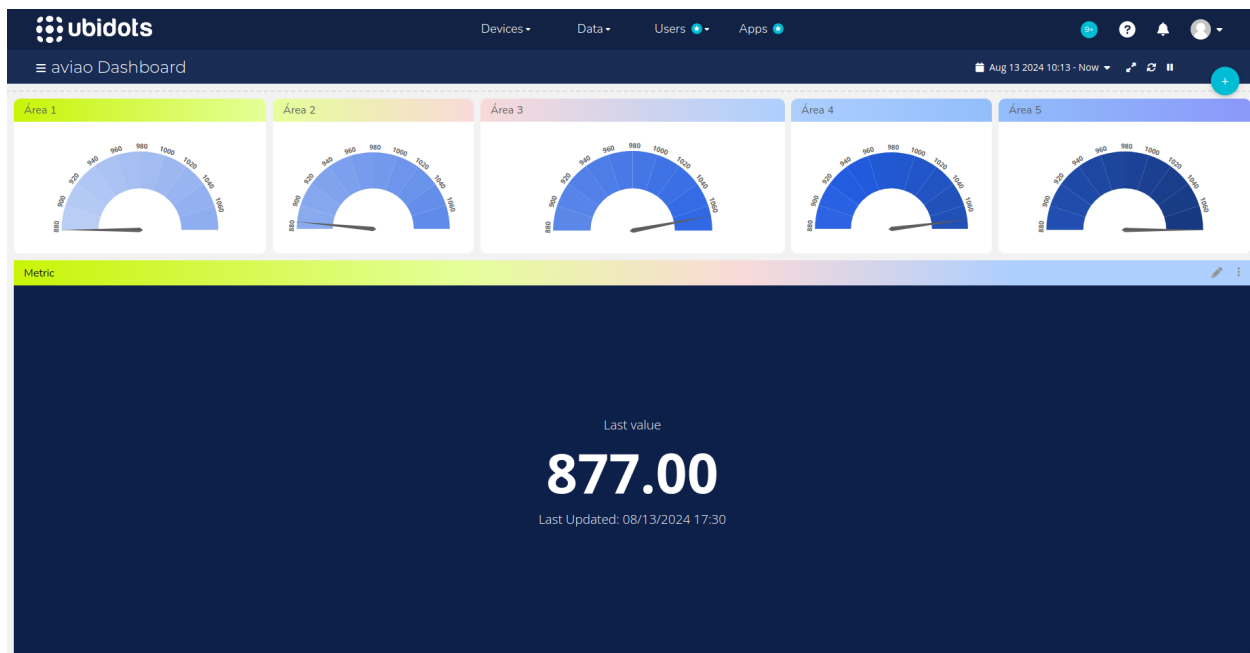
- **Kauã de Oliveira Ribeiro**
- **Júlio César Magalhães Bueno**
- **Samuel Farias Santos**
- **Victor de Jesus Alberti**

## **Funcionamento do Projeto:**

Inicialmente, o operador da torre de comando precisa definir as frequências de cada uma das cinco áreas por meio da utilização do código de controle de voo no computador, responsável por enviar esses valores ao broker MQTT no Ubidots. A partir dessa ação, as informações podem ser acessadas posteriormente pela dashboard, também hospedada na plataforma, e pelo sistema de comunicação do avião.

```
Insira o valor de frequência da área 1: 877
Insira o valor de frequência da área 2: 883
Insira o valor de frequência da área 3: 1067
Insira o valor de frequência da área 4: 1071
Insira o valor de frequência da área 5: 1079
[INFO] Attempting to connect...
[INFO] Connected to broker
[INFO] Attempting to publish payload:
{'area1': {'value': 877, 'context': {'Info': 'AreaA'}}}
[INFO] Into Topic:
/v1.6/devices/aviao
[INFO] Published!
[INFO] Attempting to publish payload:
{'area2': {'value': 883, 'context': {'Info': 'AreaB'}}}
[INFO] Into Topic:
/v1.6/devices/aviao
[INFO] Published!
[INFO] Attempting to publish payload:
{'area3': {'value': 1067, 'context': {'Info': 'AreaC'}}}
[INFO] Into Topic:
/v1.6/devices/aviao
[INFO] Published!
[INFO] Attempting to publish payload:
{'area4': {'value': 1071, 'context': {'Info': 'AreaD'}}}
[INFO] Into Topic:
/v1.6/devices/aviao
[INFO] Published!
[INFO] Attempting to publish payload:
{'area5': {'value': 1079, 'context': {'Info': 'AreaE'}}}
[INFO] Into Topic:
/v1.6/devices/aviao
[INFO] Published!
Insira o valor de frequência da área 1: █
```

Após isso, o piloto do avião precisa então selecionar a área correspondente à posição em que seu avião se encontra localizado por meio da utilização de cinco botões do “painel elétrico” da aeronave (A,B,C,D e E).



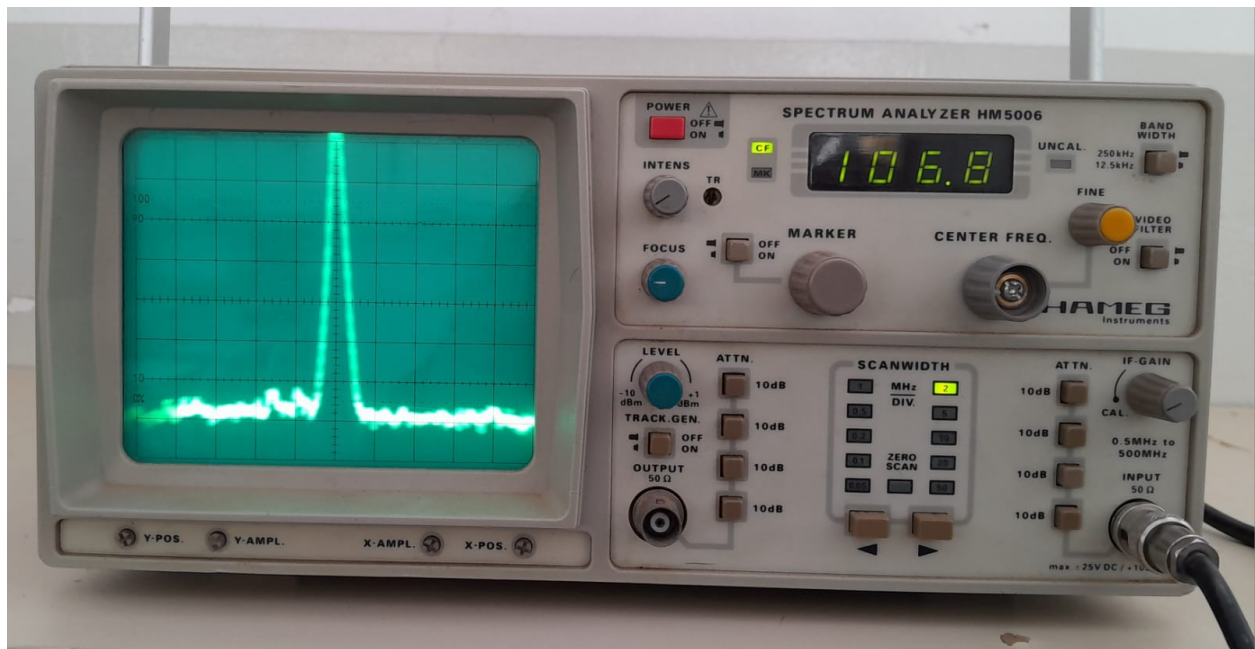
Após essa seleção, a informação da área é então enviada ao Raspberry Pi pelo Arduino Nano via comunicação serial ("A", "B", "C", "D" ou "E"). Ao chegar ao Raspberry Pi, essa informação é então utilizada pelo microcontrolador para determinar a frequência de comunicação respectiva à área escolhida, descoberta anteriormente por meio da inscrição do microcontrolador nos tópicos do broker

MQTT do projeto, na qual as frequências de cada uma das áreas, definidas pelo código hospedado no computador da "Torre de Comando", podem ser acessadas.

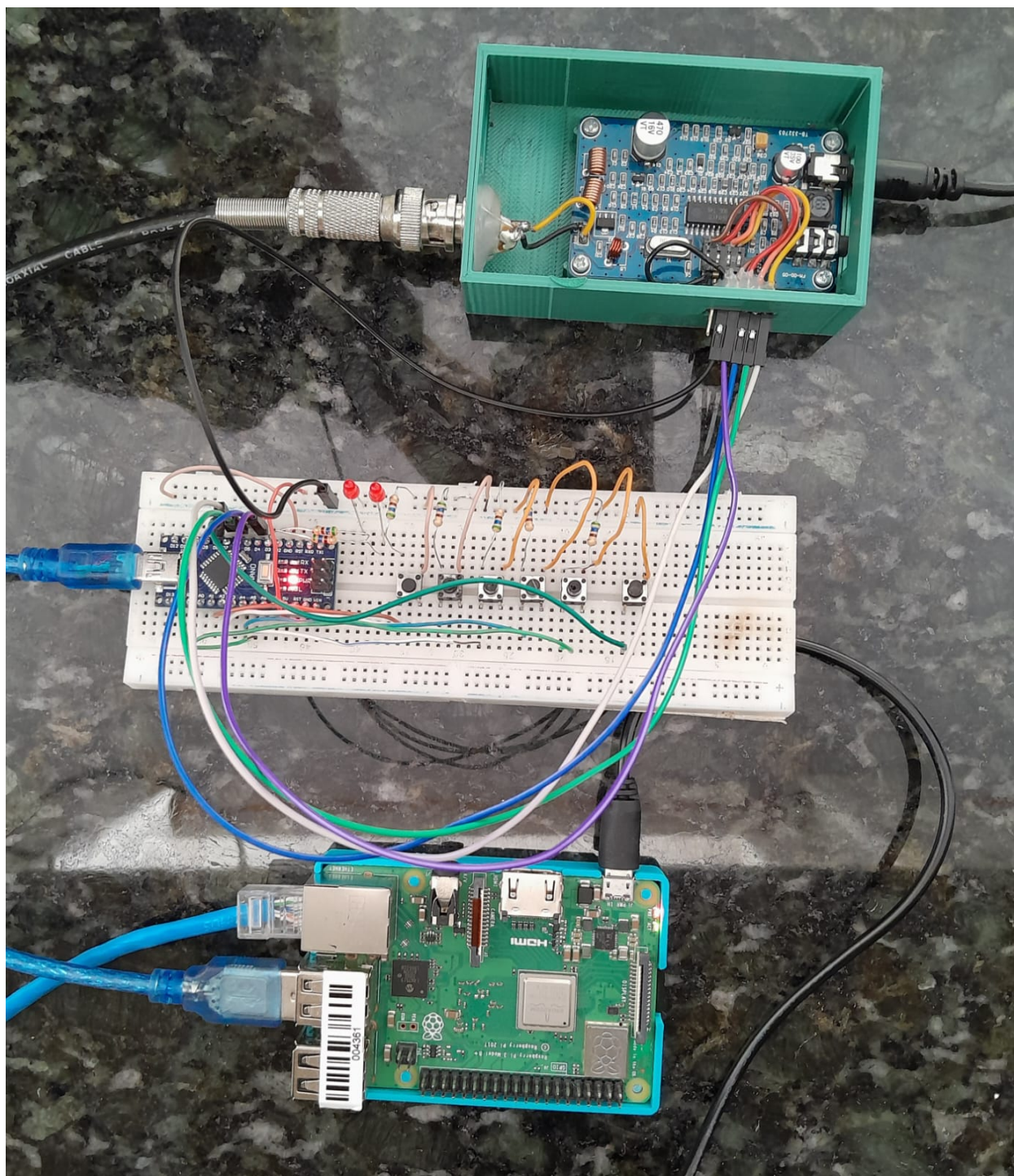
```
Shell
Valores de frequencias: 877, 883, 1067, 1071, 1079
Serial lida: A
b'877'
[INFO]Frequência do avião publicada no Ubidots.
Valores de frequencias: 877, 883, 1067, 1071, 1079
Serial lida: D
b'1071'
[INFO]Frequência do avião publicada no Ubidots.
Valores de frequencias: 877, 883, 1067, 1071, 1079
Serial lida: D
b'1071'
[INFO]Frequência do avião publicada no Ubidots.
```

Após essa operação, o Raspberry então envia a informação da área em que o avião se encontra localizado, via comunicação serial, ao Arduino, que ao recebê-la decide qual frequência será utilizada como a frequência portadora do dispositivo de rádio do avião:

- Em caso de sucesso (Leitura de um número de 3 ou 4 dígitos, como 878), o LED que simboliza as trocas de frequências do rádio pisca no painel de controle do avião e a modulante é configurada de acordo com o número recebido, na qual o ultimo dígito corresponde ao decimal (Ex: 87.8 MHz).
- Em caso de fracasso (Leitura do caractere "X"), o LED de emergência começa a piscar, sinalizando ao piloto que o sistema não conseguiu obter as informações do Ubidots. Nesse caso, ele deve apertar o sexto botão do painel de controle para que o sistema de comunicação volte a funcionar, porém dessa vez na frequência de emergência, sendo ela a onda na qual o sinal apresenta a maior propagação (87.7MHz).







## Esquema Elétrico:



## Protocolos de Transmissão Utilizados:

- Comunicação serial, realizada por meio da importação da biblioteca "Serial" na Raspberry Pi, permitindo que os dois microcontroladores do projeto comuniquem-se enviando e recebendo informações como as frequências das cinco áreas e a área na qual encontra-se o avião.
- Protocolo MQTT, utilizado pela Raspberry Pi para extrair as informações de frequência do broker hospedado no Ubidots e pelo código da central de controle para inserir essas informações no broker.

## Problemas Encontrados:

- Ao customizarem a dashboard do projeto, os integrantes colocaram um gráfico muito pesado no servidor do Ubidots, o que fez com que ele travasse, tornando-se inacessível pelos membros. Dessa forma, foi necessário que a equipe criasse uma nova conta no Ubidots e refizesse tudo o que haviam feito da última vez, solucionando assim o problema.
- Ao receber as informações das frequências do Ubidots não era possível estabelecer uma relação entre cada uma delas e as cinco áreas de voo. Para solucionar o problema, a equipe decidiu inserir uma nova informação na chave "context" do broker, identificando tudo o que fosse recebido pela Raspberry Pi.
- Após todas as configurações do projeto realizadas, a Raspberry Pi e o Arduino Nano ainda não conseguiam se comunicar devido a alguns problemas na lógica de operação de ambos os microcontroladores. Após algumas mudanças no funcionamento dos microcontroladores, o empecilho foi rapidamente resolvido pela equipe.

## Códigos do Projeto:



**Código em Python da Raspberry Pi:**

<https://github.com/Scalifax/ProjetoIntegradoT2/blob/main/app.py>

**Código em Python da torre de controle:**

<https://github.com/Scalifax/ProjetoIntegradoT2/blob/main/tower.py>

**Código em Wiring do Arduino Nano:**

<https://github.com/Scalifax/ProjetoIntegradoT2/blob/main/ArduinoRadio.ino>