

## INDICE

- 1 - Descrição geral da telemetria
- 2 - Descrição Esp32
  - 2.1 - Funcionamento esp32 na telemetria
  - 2.2 - Pacote de dados
- 3 - Descrição esp32 cam
  - 3.1 - Funcionamento esp32 cam na telemetria
- 4 - Descrição módulo sd
- 5 - Descrição tablet usado como painel do barco
- 6 - Raspberry pi 3
  - 6.1 - Funcionamento Raspberry Pi instalado no barco
  - 6.2 - Funcionamento Raspberry Pi instalado em terra
- 7 - Antenas wifi 5ghz
- 8 - Aplicativo android "display solaris"
- 9 - Aplicativo android "ip web cam"
- 10 - Mit App Inventor
- 11 - Servidor Web e Servidor Http

## 1 – Descrição da Telemetria.

### Partes de Armazenamento, Exibição e Monitoramento por Câmeras.

O objetivo do grupo responsável por esta parte do projeto da telemetria é:

- Desenvolver o firmware para o microcontrolador esp32 que irá auxiliar no armazenamento e transmissão dos dados coletados pelo sistema de monitoramento do barco da equipe Solaris.
- Desenvolver um aplicativo android que irá exibir informações relevantes ao piloto da embarcação. Esse APP será instalado em um tablet presente no painel do barco e que servirá como display de dados.
- Instalar, configurar e disponibilizar câmeras de monitoramento pela embarcação, para que possam ser salvos vídeos das provas realizadas e permitir o stream desses vídeos em tempo real com a equipe em terra.
- Criar e armazenar localmente um arquivo de Log com os dados captados pelo barco.
- Configurar uma rede local que permita a comunicação entre os dispositivos presentes na telemetria do barco com a equipe em terra, bem como a transmissão dos dados para a equipe em tempo real, com fins de monitoramento e aplicação de estratégias durante as provas do DSB.
- Desenvolver uma aplicação que permita a equipe em terra acompanhar o desempenho do barco e seus componentes, de forma autônoma ao piloto da embarcação.

## 2 – Descrição Esp32

ESP32 é um módulo microcontrolador de alto desempenho com baixíssimo consumo de energia. É uma evolução do já conhecido ESP8266, com maior poder de processamento e bluetooth BLE 4.2 embutido.

No ESP32 temos Wifi 2.4Ghz com antena embutida, uma interface usb-serial e regulador de tensão 3.3V. A programação pode ser feita usando a IDE do Arduino através de um cabo micro-usb. Com 4 MB de memória flash, o ESP32 permite criar variadas aplicações para projetos de IoT, acesso remoto, webserver, dataloggers, entre outros.

### 2.1 – Funcionamento Esp32 na telemetria

O microcontrolador Esp32 fará as seguintes tarefas:

- **WebServer** – Irá responder as solicitações HTTP do aplicativo android instalado no painel;
- **Banco de Dados** – Irá salvar os dados obtidos pelos sensores no barco em um arquivo de LOG num cartão SD, através de um módulo SD Card;
- **Publicador MQTT** – Irá publicar no broker MQTT, instalado no raspberry em terra, os dados coletados, para que a aplicação que exibe os dados para equipe seja alimentada.
- Irá receber os dados dos sensores dos barcos através da interface Serial de outro ESP32 onde estão conectados os sensores de leitura.

## 2.2 – Pacote de dados

O pacote de informações disponibilizado pelo esp32 responsável pelo funcionamento dos sensores, possui o tamanho fixo de 48 caracteres, no seguinte formato:

**30.3|10.1|20.2|40.4|50.5|-26.242370|-48.642423|2806|182100**

Contendo as seguintes informações: Tensão das baterias principais, Tensão das baterias auxiliares, Tensão das placas solares, Corrente do motor, Latitude, Longitude, data (DDMM), hora (HHMMSS).

## 3 – Descrição Esp32 Cam

Módulo desenvolvido a partir do microcontrolador Esp32, equipado com uma câmera de 2mp com ângulo de captura de 66º (padrão) ou 160º (adquirida separadamente), conexão wifi 2.4Ghz, entrada para cartão Micro Sd e pinos GPIO.

Este módulo é capaz de gerar vídeos e armazena-los automaticamente no cartão Sd.



### 3.1 - Funcionamento Esp32 Cam na telemetria

Utilizando-se a biblioteca “ESP32 CAM Video Recorder junior”, o módulo ao se ligar gera arquivos de vídeo no formato avi, que são armazenados automaticamente no cartão micro Sd inserido na placa. O cartão deve estar no formato FAT32 obrigatoriamente.

A gravação inicia ao se ligar o módulo e será encerrada após atingir determinado tempo limite, definido em seg (que pode ser ajustado na biblioteca), gerando assim um arquivo de vídeo. Logo após se inicia um novo arquivo de vídeo com a mesma duração do anterior, seguindo assim até que o cartão micro Sd fique cheio ou o módulo seja desligado.

Caso ocorra alguma interrupção de energia e o módulo se desligue, ou algum erro impeça o normal funcionamento do mesmo, o arquivo de vídeo que estava sendo criado no momento será perdido. Para evitar essa situação, foi definido que cada arquivo de vídeo terá duração de 90seg, gerando diversos arquivos pequenos e reduzindo a chance de perda de vídeos.

O módulo possui conexão wifi que permite acessar stream da câmera, o que auxilia no posicionamento e fixação do ângulo de gravação. Para configurar uma rede wifi é necessário inserir o SSID e a senha diretamente no firmware do módulo (consulte o

repositório da biblioteca “ESP32 CAM Video Recorder junior” no github para mais informações).

Feito isso, deve-se iniciar o módulo conectado via USB no computador e através do serial monitor verificar qual ip foi atribuído na rede para o módulo, e através deste IP acessar a câmera via browser.

Biblioteca Esp32 Cam Video Recorder Junior

<https://github.com/jameszah/ESP32-CAM-Video-Recorder-junior>

#### 4 – Descrição módulo SD

Este módulo irá permitir a criação de um Datalogger para salvar os dados obtidos pelos sensores instalados no barco.

O módulo se comunica através da interface SPI com sinal de 3.3V, mas internamente o módulo possui um regulador permitindo a ligação de placas e controladores que trabalham com 5V. Suporta cartões micro SD e micro SDHC.



#### 5 – Descrição Tablet Utilizado como Painel do barco

Será utilizado um tablet de baixo custo, modelo DL HD7 Plus, instalado dentro do painel do barco, para que seu display mostre informações ao piloto, se aproveitando também de sua câmera integrada para fazer streaming do piloto para equipe em terra.

Especificações do Tablet:

Tela	7 polegadas
Resolução	800 x 480 pixels
Sistema operacional	Android 4.0 Ice Cream Sandwich
Processamento	1,2GHz; 1 GB de RAM
Rede	Não
Armazenamento	4 GB
Câmera	2 megapixels
Conectividade	Wi-Fi 802.11 b/g/n, USB, HDMI
Dimensões	23,5 x 16 x 6,5 cm
Peso	570g

## **6 - Raspberry Pi 3**

O Raspberry Pi é um computador montado em uma placa única do tamanho aproximado de um cartão de crédito, de baixo consumo de energia e que roda sistema Linux ou alguma de suas variáveis desenvolvidas pela comunidade Open Source.

Para o projeto da telemetria serão utilizados 2 Raspberry Pi 3 Model B, como descritos abaixo.

### **6.1 – Funcionamento Raspberry Pi instalado no barco**

Este Rasp irá funcionar como hotspot wi-fi, disponibilizando uma rede Wireless Lan e permitindo assim a comunicação dos dispositivos do barco entre si. Irá se conectar via ethernet com a antena Wifi 5Ghz instalada no barco, a qual se comunica com a estação em terra.

A rede Wifi deste aparelho está em bridge com a rede lan, operando dessa forma na faixa de ip 192.168.0.0/24 (conferir mapa de rede no fim deste documento).

### **6.2 – Funcionamento Raspberry Pi instalado em terra**

Este raspberry irá servir como hotspot wifi e gerenciador da rede em terra. Sua interface ethernet funciona na faixa de IP 192.168.0.0/24, enquanto o hotspot Wifi distribui IP na faixa 192.168.5.0/24.

## **7 – Antenas WiFi 5Ghz**

Foram utilizadas quatro antenas WiFi 5.8 Ghz, sendo duas em uso e duas como reserva, da marca Proeletronic, modelo PQWS 5820A, de 20dbi, para comunicação entre o barco e a estação em terra, elas são alimentadas via adaptador POE, através de baterias 12v 7Ah. As antenas utilizam a configuração WDS, sendo que apenas a antena em terra está configurada como DHCP. Tanto a antena da estação em terra quanto a antena instalada no barco, possuem uma sobressalente de igual modelo e prontamente configurada para eventual necessidade de substituição.

## 8 – Aplicativo Android “Display Solaris”

Desenvolvido na plataforma Mit App Inventor, este é o aplicativo que irá exibir os dados coletados pelos sensores do barco ao piloto.



## 9 – Aplicativo Android “IpWebCam”

Transforma as câmeras do dispositivo android em uma espécie de WebCam, que pode ser acessada através do IP pela rede

## 10 – Mit App Inventor

Plataforma gratuita, disponibilizada pelo MIT (Massachusetts Institute of Technology), permite a criação de aplicativos para plataforma Android de forma simples, por meio do uso de blocos para facilitar a programação.

(<http://appinventor.mit.edu/>)

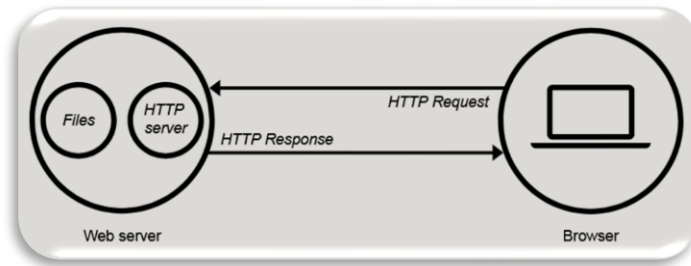
## 11 – Servidor Web e Servidor HTTP

Estando o tablet conectado a rede Wifi fornecida pelo raspberry instalado no barco, os dados que serão exibidos ao piloto através do app irão vir do ESP32 via protocolo HTTP disponibilizados por um WebServer. Mas o que é um WebServer?

Referente ao hardware, um WebServer é um dispositivo que armazena arquivos que compõem os sites (no caso deste projeto, esses arquivos estão contidos na biblioteca ESPAsyncWebServer) e os entrega para o dispositivo do usuário final (nesse caso o app instalado no tablet).

Referente ao software, um servidor web inclui diversos componentes que controlam como os usuários acessam os arquivos hospedados (armazenados para disponibilização), no caso desse projeto será fornecida a String contendo todos os dados coletados. Um servidor HTTP é um software que compreende URLs (endereço web) e HTTP (o protocolo que seu navegador utiliza para visualizar páginas web).

Simplificando, o app fará uma requisição utilizando o protocolo HTTP sempre que necessitar dos dados hospedados no WebServer. Quando a requisição alcançar o servidor web (hardware), o *servidor HTTP* (software) enviará a string de dados requerida, também via HTTP.



\* O uso do protocolo mqtt na telemetria foi substituído pelo uso do web server. Esta parte foi deixada apenas para referência e consulta.

## 12 – Descrição Protocolo MQTT

MQTT é um protocolo de mensagens leve para sensores e pequenos dispositivos móveis, otimizado para redes TCP/IP. O esquema de troca de mensagens é fundamentado no modelo Publicador-Subscritor, extremamente simples.

### 12.1 - Funcionamento MQTT na Telemetria

O módulo ESP32 irá fazer o papel de **publicador**, recebendo os dados dos sensores vindos de um segundo ESP32 via comunicação serial e “publicando” via Wifi no broker instalado no raspberry pi em terra. Uma aplicação web ficará responsável por ser o **subscritor**, lendo os dados publicados no broker e alimentando com eles um mapa com marcador, exibindo assim a posição do barco na prova e as informações de seus componentes como bateria, placas solares e consumo do motor.

No firmware do receptor são definidos:

ID do Esp32 que irá se comunicar com o broker: **Esp32**

Endereço do tópico para publicação: **/app/dados**

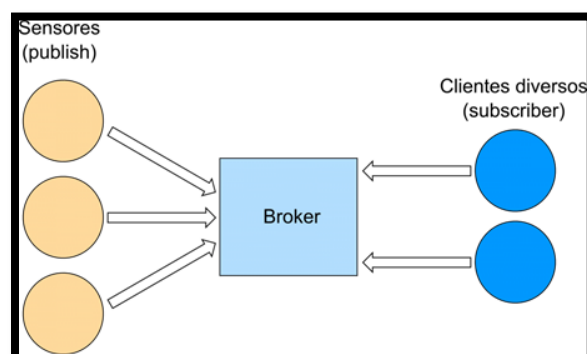
IP do broker/servidor MQTT: **192.168.0.14** (pode variar dependendo da rede).

Biblioteca PubSubClient:

<https://github.com/knolleary/pubsubclient>

Documentação PubSubClient:

<https://pubsubclient.knolleary.net/>



## MAPA REDE TELEMETRIA SOLARIS

### COMPONENTENENTE PRESENTES NO BARCO

Raspberry PI Barco  
Ip Lan – 192.168.0.10  
WiFi 2.4Ghz  
(Wifi em Bridge com a Lan)  
Ssid – Solaris Boat  
Senha – abcd1234

Ant. Wifi 5Ghz Barco  
Ip Lan – 192.168.0.3  
WiFi 5Ghz  
(Wifi em Bridge com a Lan)  
Sem DHCP  
Ssid – Proeletronic D

Tablet Display  
Ip Wifi – 192.168.0.150  
WiFi 2.4Ghz  
IpWebCam – 192.168.0.150:8032/video  
http request - <http://192.168.0.12/info>

Esp32 WebServer  
Ip Wifi – 192.168.0.12  
WiFi 2.4Ghz  
Ip WebServer – 192.168.0.12:80

Esp32 WebCam  
Ip Wifi – 192.168.0.54  
WiFi 2.4Ghz  
Ip Stream – 192.168.0.154/stream

### COMPONENTENENTE PRESENTES EM TERRA

Ant. Wifi 5Ghz Terra  
Ip Lan – 192.168.0.1  
WiFi 5Ghz  
(Wifi em Bridge com a Lan)  
DHCP ativo (100 / 199)  
Ssid – Proeletronic A

Raspberry PI Terra / Broker MQTT  
Ip Lan – 192.168.0.14  
WiFi 2.4Ghz em modo AP  
Ip WiFi – 192.168.5.1  
DHCP – 5.10 / 5.20  
Ssid – Solaris\_AP  
Senha – abcd1234  
Tópico MQTT /app1/dados/equipe00



