

U.C. de Projeto Integrado de

Telecomunicações

Ano Letivo: **2021/2022**

**Relatório da Fase B**

**Grupo 2**

* Catarina Neves, a93088
* Eduardo Cardoso, a89627
* José Gomes, a93083
* Luís Oliveira, a89380

27/03/2022

Universidade do Minho

Mestrado Integrado em Engenharia de Telecomunicações e Informática

Índice

[1. Introdução 5](#_Toc101470801)

[2. Trabalho Relacionado 6](#_Toc101470802)

[2.1 ESP32 WebSocket Server: Control Outputs 6](#_Toc101470803)

[2.2 ESP32 WebSocket Server with Multiple Sliders: Control LEDs Brightness 7](#_Toc101470804)

[3. Etapas do trabalho desenvolvido 8](#_Toc101470805)

[3.0. Montagem Eletrónica 8](#_Toc101470806)

[3.1. Aquisição das amostras dos sensores para o Sistema Sensor 9](#_Toc101470807)

[3.2. Conversão e processamento dos valores obtidos nos sensores 11](#_Toc101470808)

[3.3. Apresentação dos dados recolhidos em tempo real 12](#_Toc101470809)

[3.4. Transmissão dos dados via BLE para o Gateway 12](#_Toc101470810)

[3.5. Envio dos dados via Wi-Fi para visualização e armazenamento online 13](#_Toc101470811)

[4. Análise de resultados e testes efetuados 15](#_Toc101470812)

[5. Conclusão 18](#_Toc101470813)

[5.1. Contribuição de cada aluno 18](#_Toc101470814)

[6. Lista de referências 19](#_Toc101470815)

Índice de figuras

Figura 1 - Tarefas propostas pela fase B. 5

Figura 2 – Aplicação e circuito implementados pelo autor. 6

Figura 3 – Página web com 3 sliders. 7

Figura 4 - Circuito eletrónico implementado pelo autor. 7

Figura 5 - Diagrama eletrónico dos componentes. 8

Figura 6 - Sistema Sensor: sensores e variáveis associadas. 9

Figura 7 – Fluxograma da função responsável pela aquisição dos dados meteorológicos. 10

Figura 8 - Processamento dos dados obtidos na etapa anterior. 11

Figura 9 - Acesso à humidade e temperatura através da variável event. 11

Figura 10 - Apresentação dos dados no terminal Serial Monitor do Arduino. 12

Figura 11 - Comunicação via BLE entre Sensor e Gateway. 12

Figura 12 - Configuração das propriedades do BLE. 12

Figura 13 - Envio de dados pelo Gateway para o servidor online ThingSpeak. 13

Figura 14 - Fluxograma da função responsável pelo envio de dados para o ThingSpeak. 14

Figura 15 - Serial Monitor do Sistema Sensor. 15

Figura 16 - Serial Monitor do Gateway. 15

Figura 17 - Variação da temperatura. 16

Figura 18 - Variação da humidade. 16

Figura 19 - Variação da pressão atmosférica. 17

Figura 20 – Campos e detalhes da base de dados. 17

Figura 21 – Detalhes e composição de uma determinada amostra. 17

Lista de siglas e acrónimos

**LED** *Light Emitting Diode*

**PWM** *Pulse Width Modulation*

**TCP** *Transmission Control Protocol*

**IP** *Internet Protocol*

# Introdução

Serve o presente relatório como síntese do trabalho desenvolvido e implementado no decorrer desta segunda fase. Este contém a descrição das estratégias e algoritmos adotados pelo grupo, tal como os respetivos testes realizados. Além dos tópicos acima citados, faz-se referência a trabalhos ou projetos similares, que se enquadram na ótica deste projeto.

O relatório abordará cada etapa desta fase de forma detalhada, ou seja, será apresentada a resposta ou proposta de solução empregue pelo grupo em junção com as ferramentas que foram necessárias para a sua construção.

A figura 1 ilustra o sumário das diferentes tarefas propostas nesta fase.

Uma imagem com texto, ecrã, escuro

Descrição gerada automaticamente

Figura 1 - Tarefas propostas pela fase B.

<

Nesta fase, o objetivo é a implementação de um sistema central e do protocolo de comunicação entre este sistema e os *gateways* (*Gateway* e Sistemas Simulados), que deve estar assente numa comunicação por mensagens aplicacionais sobre TCP (*Transmission Control Protocol*) /IP (*Internet Protocol*).

O sistema central recolhe a informação enviada por todos os *gateways*, que por sua vez, será armazenada na base de dados. Sempre que o serviço *web* queira atualizar os dados apresentados, este faz um pedido ao gestor de serviço que, de forma encadeada, irá buscar os dados à base de dados e enviará os mesmos para o serviço *web*.

# Trabalho Relacionado

Relativamente aos trabalhos relacionados, enquadram-se os projetos seguintes, que serviram de base à construção deste projeto.

## 2.1 ESP32 WebSocket Server: Control Outputs

Este projeto, da autoria de Rui Santos [1], consiste numa aplicação que varia o estado de um determinado LED (*Light Emitting Diode*)*,* sendo baseada no modelo cliente-servidor: a placa ESP32 desempenha o papel de servidor e o *browser*, que contém uma página *web* (configurada no código do servidor), atua como cliente, capaz de controlar a placa referida de forma remota.

A página *web* apresenta o estado do LED e possui um botão, que é responsável pela mudança de estado do mesmo (ligar ou desligar). É de salientar, que o LED é conectado à placa referida, através de fios de ligação e com o auxílio de uma *breadboard*.

O cliente estabelece uma ligação via *websocket* com o servidor; quando esta é estabelecida, o cliente e o servidor podem enviar dados com recurso a *sockets*, de forma *full-duplex*.

A utilização do protocolo baseado em *websockets,* permite que o servidor envie informação para o(s) seu(s) cliente(s) sem necessitar de ser requisitado pelos mesmos, logo, sempre que o estado do LED varie, essa informação é enviada para o *browser*, tal como é visível na figura 2.

Uma imagem com texto, interior

Descrição gerada automaticamente

Figura 2 – Aplicação e circuito implementados pelo autor.

A aplicação funciona de acordo com os passos seguintes:

1. Clicar no botão “*Toggle*”;
2. O cliente (*browser*) envia a mensagem “*toggle*” via *socket*;
3. O servidor (placa ESP32) recebe essa mensagem e muda o estado do LED;
4. O servidor envia o novo estado do LED para o(s) seu(s) clientes;
5. O(s) cliente(s) recebe(m) a mensagem e atualiza(m) o estado do LED, na webpage associada;

## 2.2 ESP32 WebSocket Server with Multiple Sliders: Control LEDs Brightness

Este projeto, da autoria de Rui Santos [2], consiste numa aplicação que permite variar a luminosidade de vários LEDs, sendo baseada no modelo cliente-servidor: a placa ESP32 desempenha o papel de servidor e o browser, que contém uma página web (configurada no código do servidor), atua como cliente, capaz de variar a luminosidade dos respetivos LEDs. A comunicação entre cliente(s) e servidor é realizada através de um protocolo assente em websockets.

A variação da luminosidade dos LEDs é feita através de um conjunto de *sliders,* que controlam o *duty cycle* dos respetivos sinais PWM (*Pulse Width Modulation*).

A figura seguinte ilustra a página web que contém os *sliders* referidos.

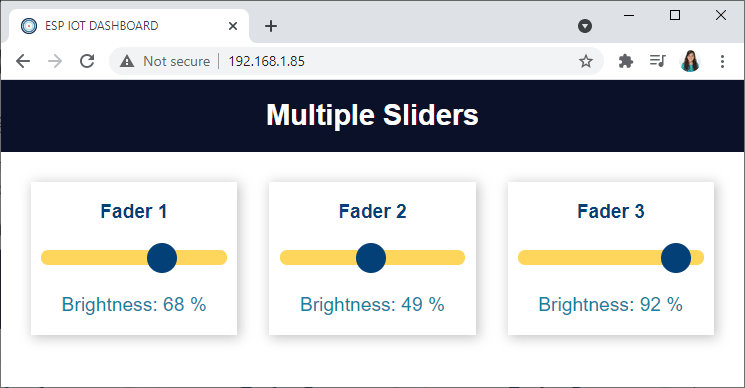


Figura 3 – Página web com 3 sliders.

A página web encontra-se organizada da seguinte forma:

* Um *fader* por LED, num total de 3 LEDs;
* Cada *fader* possui um *slider,* que é responsável pela variação da luminosidade do LED associado;
* A luminosidade do LED (em percentagem) varia entre 0 e 100;
* Sempre que um *slider* varie a luminosidade, esta é atualizada em todos os clientes de forma simultânea;

O servidor coloca na rede local uma página web, que é constituída por 3 *sliders*. Sempre que um *slider* é alterado (luminosidade alterada), o cliente envia o número relativo desse *slider* para o servidor, de acordo com o protocolo referido anteriormente. Por exemplo, se um utilizador colocar a posição do *slider* 3 em 50%, o cliente enviará a mensagem “**3s50**” para o servidor. Após o servidor ter recebido a mensagem mencionada, ajusta o *duty cycle* do sinal PWM, de acordo com os valores incorporados na mensagem e notifica os restantes clientes, com os novos valores.

A figura seguinte ilustra o circuito eletrónico efetuado pelo autor.

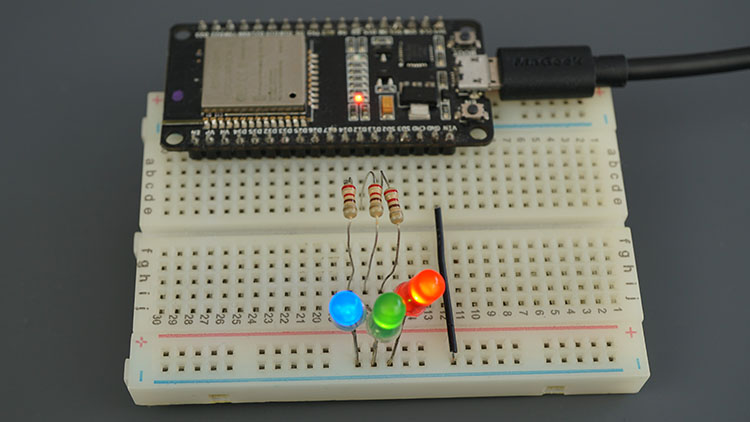


Figura 4 - Circuito eletrónico implementado pelo autor.

# Etapas do trabalho desenvolvido

Esta secção contém todas as etapas necessárias ao desenvolvimento desta fase, que se encontram descritas pelos mecanismos, ferramentas utilizadas e algoritmos implementados pelo grupo.

## 3.1. Implementação dos Sistemas Simulados

## 3.2. ETAPA 2

## 3.3. ETAPA 3

## 3.4. ETAPA 4

## 3.5. ETAPA 5 (Servidor Web maybe?)

# Análise de resultados e testes efetuados

# Conclusão

Concluída esta fase, na opinião do grupo, conseguiu-se atingir todos os objetivos propostos. Esta fase contribuiu para o conhecimento e estudo das tecnologias *Bluetooth Low Energy* e WiFi, tal como a interação com sensores e as suas respetivas bibliotecas.

O grupo pretende, através uma boa e forte base, seja possível atingir as metas definidas para as próximas etapas do projeto enunciado.

## 5.1. Contribuição de cada aluno

Catarina Neves

* Conceção dos algoritmos;
* Tratamento de dados, tabelas e imagens a estes relacionados;
* Desenho de esquemas e gráficos;
* Recolha e implementação de *timestamps;*
* Desenvolvimento do relatório;

Eduardo Cardoso

* Conceção dos algoritmos;
* Implementação e desenvolvimento dos algoritmos de comunicação BLE;
* Implementação da comunicação Wi-Fi;
* Implementação da comunicação entre *Gateway* e *Thingspeak;*
* Recolha e implementaçãode *timestamps;*
* Desenvolvimento do relatório;

José Gomes

* Conceção dos algoritmos;
* Tradução dos algoritmos implementados em fluxogramas;
* Implementação dos algoritmos de comunicação BLE;
* Desenvolvimento do relatório;

Luís Oliveira

* Conceção dos algoritmos;
* Criação e configuração da base de dados no *ThingSpeak*;
* Implementação da comunicação entre *Gateway* e *ThingSpeak;*
* Desenho de esquemas e gráficos;
* Desenvolvimento do relatório.

# Lista de referências

|  |  |
| --- | --- |
| [1] | R. Santos, “Random Nerd Tutorials,” 10 2020. [Online]. Available: https://randomnerdtutorials.com/esp32-websocket-server-arduino/. |
| [2] | R. Santos, “ESP32 Web Server (WebSocket) with Multiple Sliders: Control LEDs Brightness (PWM),” 5 2021. [Online]. Available: https://randomnerdtutorials.com/esp32-web-server-websocket-sliders/. |
| [3] | Aosong, “Temperature and humidity module DHT11 Product Manual,” 14 maio 2015. [Online]. Available: https://www.waveshare.com/w/upload/c/c7/DHT11\_datasheet.pdf. |
| [4] | Bosch, “BME280 Combined humidity and pressure sensor,” setembro 2018. [Online]. Available: https://www.mouser.com/datasheet/2/783/BST-BME280-DS002-1509607.pdf. |
| [5] | ioBridge, “ThingSpeak for IoT Projects,” 2022. [Online]. Available: https://thingspeak.com/. |
| [6] | NTP Pool Project, “Europe — europe.pool.ntp.org,” [Online]. Available: https://www.pool.ntp.org/zone/europe. |