



**ISEL**

**ADEETC**

Área Departamental de  
Engenharia Electrónica e  
Telecomunicações e  
de Computadores

## CONTROLADOR DE PAINÉIS DE LED (RGB)

**Paulo António da Costa Torres Neves**  
**(Número 38714)**

LICENCIATURA DE ENGENHARIA ELETRÓNICA DE TELECOMUNICAÇÕES E DE COMPUTADORES

(LEETC)

PROJETO FINAL DE CURSO

ORIENTADORES: ANDRÉ RIBEIRO LOURENÇO

PEDRO MIGUEL FLORINDO MIGUENS MATUTINO

Setembro 2019

# Índice de conteúdos

|   |            |
|---|------------|
| <i>Lista de Figuras .....</i>                   | <i>i</i>   |
| <i>Lista de Tabelas.....</i>                    | <i>ii</i>  |
| <i>Acrónimos .....</i>                          | <i>iii</i> |
| <i>1. Introdução.....</i>                       | <i>1</i>   |
| 1.1    Objetivos .....                          | 2          |
| <i>2. Arquitetura Proposta.....</i>             | <i>4</i>   |
| 2.1    Arquitetura do Painel de LED's.....      | 4          |
| 2.2    Funcionamento Painel de LED's RGB .....  | 6          |
| 2.3    Arquitetura do Microcontrolador .....    | 9          |
| 2.4 Ambiente de Desenvolvimento Integrado ..... | 11         |
| <i>3. Implementação.....</i>                    | <i>12</i>  |
| 3.1    Camada Aplicacional .....                | 13         |
| 3.2    Ligação Bluetooth.....                   | 14         |
| 3.3    Ligação Wifi .....                       | 17         |
| 3.4    EEPROM Interface .....                   | 18         |
| 3.5    Interface de Texto .....                 | 18         |
| 3.6    Controlador de matriz de LED's.....      | 20         |
| <i>4. Resultados Experimentais.....</i>         | <i>22</i>  |
| 4.1    Ligação Bluetooth .....                  | 22         |
| 4.2    Ligação Wifi via Bluetooth.....          | 25         |
| <i>5. Conclusão.....</i>                        | <i>29</i>  |
| 5.1    Trabalho Futuro.....                     | 29         |
| <i>6. Bibliografia.....</i>                     | <i>30</i>  |
| <i>7. Anexos.....</i>                           | <i>31</i>  |
| Anexo A – Esquema ESP32 Adafruit Feather.....   | 31         |
| Anexo B – Esquema Eléctrico .....               | 32         |

## **Lista de Figuras**

|  |    |
|--|----|
| Figura 1 - Exemplos de Aplicação Paineis de LED's RGB.....                                   | 1  |
| Figura 2 - Diagrama de Blocos do Sistema .....   | 2  |
| Figura 3 - Diagrama de Blocos Microcontrolador e Matriz de LED's .....                       | 4  |
| Figura 4 - Matriz de LED's 4x4 com 16 LED's .....  | 5  |
| Figura 5 – Seleção de LED da matriz.....   | 5  |
| Figura 6 – Seleção de 2 linhas e 2 colunas da matriz .....                                   | 6  |
| Figura 7 - Esquema Eléctrico Shift Register .....  | 7  |
| Figura 8 - Timing Waveform e Circuito Equivalente.....                                       | 8  |
| Figura 9 - Distribuição de saídas Shift Register com correspondência de ativação.....        | 9  |
| Figura 10 - Adafruit Huzzah32 - ESP32 Feather .....  | 10 |
| Figura 11 - Módulo ESP32-WROOM-32 .....  | 10 |
| Figura 12 - Sketch Arduino .....   | 11 |
| Figura 13 - Diagrama Lógico da aplicação e hardware.....                                     | 12 |
| Figura 14 - Fluxograma Função Setup .....  | 13 |
| Figura 15 – Função Loop.....   | 13 |
| Figura 16 – Fluxograma da Função Connection Init.....  | 14 |
| Figura 17 - Fluxograma Função Connect Using BT .....   | 15 |
| Figura 18 - Fluxograma da Função Process BT .....  | 16 |
| Figura 19 - Fluxograma Wifi Process.....   | 17 |
| Figura 20 - Mapa de Memória EEPROM.....  | 18 |
| Figura 25 - Bitmap Caracter 'A' .....  | 18 |
| Figura 21 - Exemplo de Janela com Deslocamento de Texto .....                                | 19 |
| Figura 22 - Fluxograma Função Scroll Display .....   | 19 |
| Figura 23 - Fluxograma Matrix Driver.....  | 20 |
| Figura 24 – Representação Conversão de Matriz displayRGB para matriz arrayTX .....           | 21 |
| Figura 26 - Emparelhamento ESP32 com Dispositivo Externo utilizando a Aplicação BluTerm..... | 22 |
| Figura 27 - Aguarda inserção do tipo de ligação.....   | 23 |
| Figura 28 - Ligação por Bluetooth ativa.....   | 24 |
| Figura 29 - Mensagem enviada para painel via Bluetooth .....                                 | 25 |
| Figura 30 - Ligação Wifi via Bluetooth com indicação endereço IP do LED do painel.....       | 26 |
| Figura 31 - Página Web Server ESP32.....   | 27 |
| Figura 32 - Output pedido HTTP Painel de LED's .....   | 27 |
| Figura 33 - Comando seleção ligação Wifi .....   | 28 |
| Figura 34 - Esquema Eléctrico ESP32 Adafruit Feather.....                                    | 31 |
| Figura 35 - Pinout ESP32 Adafruit.....   | 31 |
| Figura 36 - Esquema Eléctrico .....  | 32 |

## Lista de Tabelas

|   |    |
|---|----|
| Tabela 1 - Características a ter em consideração na seleção de painel de LED's..... | 6  |
| Tabela 2 - Características Técnicas Painel SDM P10RGB .....                         | 7  |
| Tabela 3 - Diferentes Módulos ESP32 .....   | 10 |
| Tabela 4 - LED's de Estado .....  | 16 |

## Acrónimos

|        |   |
|--------|---|
| LED    | - Light Emissor Diode                                 |
| LAN    | - Local Area Network                                  |
| WLAN   | - Wireless Local Area Network                         |
| USB    | - Universal Serial Bus                                |
| IDE    | - Integrated Development Environment                  |
| RGB    | - Red, Green, Blue                                    |
| SMD    | - Surface Mounted Device                              |
| FIFO   | - First in First Out                                  |
| RAM    | - Random Access Memory                                |
| HTTP   | - Hypertext Transfer Protocol                         |
| SMD    | - Surface Mounted Device                              |
| SMT    | - Surface Mounted Technology                          |
| GPIO   | - General Purpose Input/ Output                       |
| SOC    | - System on Chip                                      |
| EEPROM | - Electrically-Erasable Programmable Read-Only Memory |



## 1. Introdução

Hoje em dia vulgarizou-se a utilização de painéis de divulgação de informação não estáticos, habitualmente estes baseiam-se em painéis de LED's, existem soluções no mercado, no entanto estas habitualmente têm interfaces de controlo proprietárias. A interoperabilidade entre estas dificulta a sua gestão e manutenção. Em várias aplicações é necessário frequentemente a necessidade de flexibilizar a sua utilização.

Assim surge a necessidade de criar uma interface que seja comum entre diferentes painéis, que seja dimensionável para tantos painéis quantos sejam necessários formando um agregado de painéis e que seja escalável.

Atualmente grande parte dos painéis de LED's disponíveis são modulares, com montagem de LED's, estes têm aplicações em ecrãs interiores e exteriores, com diferentes densidades de LED's por  $cm^2$ . Um pixel SMD é composto por um conjunto de três LED's, um de cor vermelha, um verde e um azul todos juntos no mesmo encapsulamento, permitindo deste modo criar oito níveis de cor.



Figura 1 - Exemplos de Aplicação Paineis de LED's RGB

Se for considerada a criação de uma *LED Wall*, conjunto de vários painéis de LED's RGB, existem fatores importantes na escolha da melhor solução como:

- o número de interfaces necessárias para controlar a estrutura;
- o facto de ser uma opção modular;
- o seu baixo consumo;
- uma fácil instalação e manutenção;
- custos reduzidos;

A implementação de um sistema de controlo com um microcontrolador permite desenvolver sistemas de baixo custo e versáteis na sua adaptação.

A interface a desenvolver permitirá a centralização e uniformização da adaptação das várias aplicações dos painéis para os utilizadores, permitindo que a comunicação entre operador e interface seja realizada recorrendo a comunicação por cabo USB ou Wireless, como o *Wifi* ou o *Bluetooth*.

### 1.1 Objetivos

O objetivo deste projeto é o desenvolvimento de uma estrutura modular com módulos básicos de 16x32 conjuntos de LED's RGB, controlados através de um microcontrolador (1). Para atingir o objetivo proposto será desenvolvido Software para o microcontrolador e interface de utilizador e o diagrama de ligações entre o microcontrolador e o painel de LED's. Na Figura 2 está representado o diagrama de blocos.

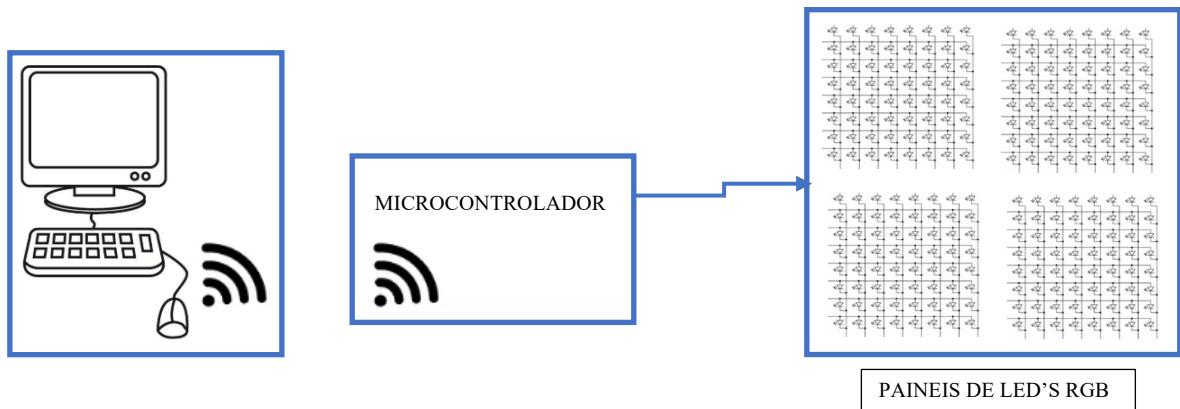


Figura 2 - Diagrama de Blocos do Sistema

Será desenvolvido software que permitirá as seguintes funcionalidades:

- Implementação de interface de controlo do painel;
- Comunicação com a aplicação utilizando interface Wifi;
- Comunicação com a aplicação utilizando interface Bluetooth;
- Acesso a uma EEPROM virtual;
- Implementação de biblioteca de texto;
- Desenho de interface de utilizador;

## 2. Arquitetura Proposta

O objetivo deste capítulo é apresentar a arquitetura utilizada na comunicação entre os diversos componentes, assim como a seleção dos mais adequados para este projeto.

Inicialmente será abordada a arquitetura do hardware dos componentes e posteriormente a seleção de ferramentas adequadas para o desenvolvimento do software.

Para o projeto houve a necessidade de realizar uma pesquisa sobre os produtos similares existentes no mercado que incluiu a pesquisa de fonte de painéis de LED's RGB, microcontroladores e o produto acabado, que consiste num painel de LED's RGB controlado por microcontrolador, como demonstrado no diagrama da **Erro! A origem da referência não foi encontrada..**

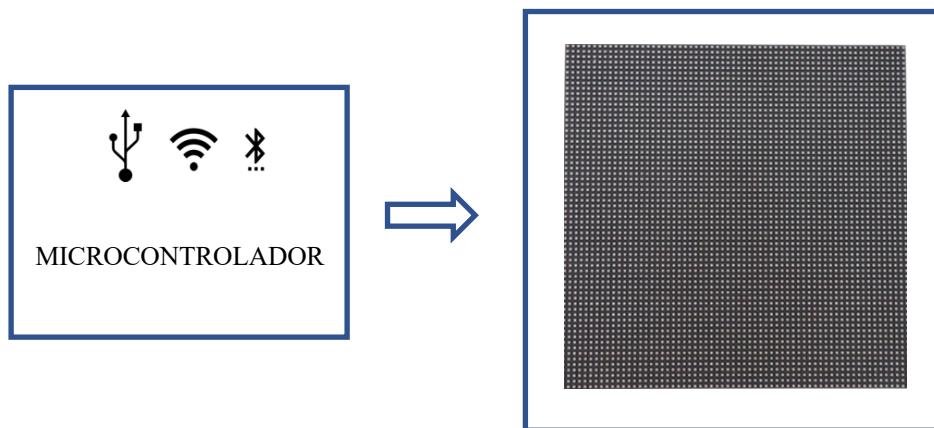


Figura 3 - Diagrama de Blocos Microcontrolador e Matriz de LED's

### 2.1 Arquitetura do Painel de LED's

Os LED's são organizados em forma de matriz e desse modo ficam dispostos em linhas e colunas. Com esta organização é possível controlar um maior número de LED's utilizando menos pinos de seleção, como por exemplo um painel com uma matriz de 4 linhas e 4 colunas, caso não estivessem dispostos de forma matricial seriam necessárias 16 saídas para ativar cada um dos LED, mas com a sua construção matricial apenas são necessárias ativar 8 saídas, como é ilustrado na Figura 4.

Este tipo de montagem pode levantar um problema caso seja pretendido ativar um determinado LED, ou seja, no caso de se querer ativar o LED da linha dois e coluna dois, seria ligado o LED conforme está apresentado na Figura 5.

Para facilitar trabalhar com forma matricial, foi utilizada a técnica de multiplexagem que consiste em ativar uma linha de cada vez. Ao selecionar a coluna com o LED que se pretende ligar, os LED's da linha ativada serão ligados. A linha é então desligada, ligando a linha seguinte repetindo o processo. Este processo torna-se num ciclo, repetido da primeira à última linha. Como esta solução requer multiplexagem no tempo, é necessário um maior controlo por parte do software.

A frequência com que cada linha é ativada é chamada de *refresh rate* e caso esta tenha uma frequência suficientemente elevada, não permite que o olho humano se aperceba da transição, o que origina que quando se observa se verifiquem um conjunto de LED ligados.

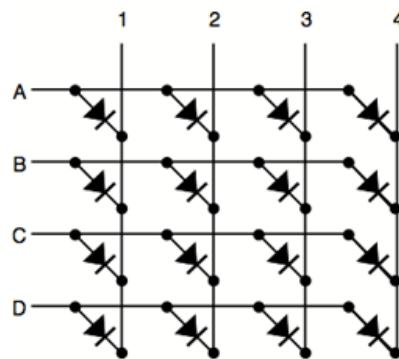


Figura 4 - Matriz de LED's 4x4 com 16 LED's

Utilizando esta técnica todos os LED's de cada linha e coluna estão interligados e quando se liga um LED ativa-se a respetiva linha e coluna.

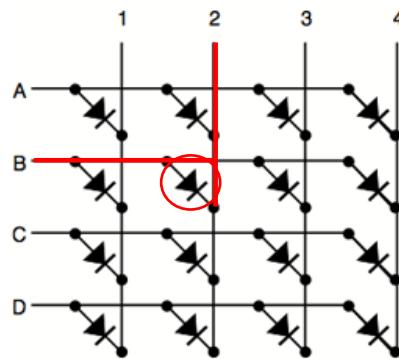


Figura 5 – Seleção de LED da matriz

Agora, caso se opte também por ativar o LED da linha quatro e coluna três, iria ativar outros dois LED's que não há interesse em ativar, conforme demonstrado na Figura 6. Esta situação ocorre, pois, está a ser imposta uma tensão nas linhas dois e quatro, e nas colunas dois e três.

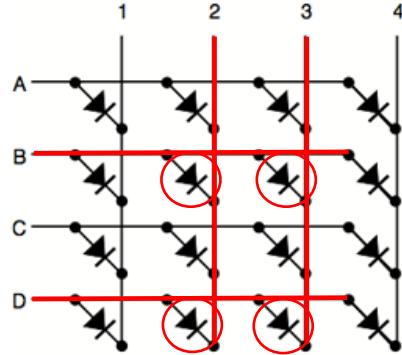


Figura 6 – Seleção de 2 linhas e 2 colunas da matriz

Com a pesquisa foi possível identificar tipos de painel que podem ter diversas características, as quais estão identificadas na Tabela 1, sendo que se optou por um painel (2), onde o seu funcionamento está explicado no capítulo 2.2 com a dimensão de 16x32 conjuntos de LED's RGB devido à sua dimensão possibilitar uma modularidade do painel, permite a visibilidade mais clara da informação projetada pelo painel, permitindo assim uma maior portabilidade durante o desenvolvimento do projeto.

Tabela 1 - Características a ter em consideração na seleção de painel de LED's

|                  |                                  |
|------------------|----------------------------------|
| Module Type      | Outdoor / Indoor                 |
| LED Type         | R/ G/ B/ W - SMD/ SMT            |
| Display Color    | Red, Green,Blue White            |
| Module Dimension | Largura x Altura                 |
| Module Pixel     | Número de pixels / Modulo        |
| Pixel Pitch      | Densidade Pixel / mm             |
| Resolution       | Número de LED's/ m <sup>2</sup>  |
| Gray Scale       | Escala de cinzentos por cada côn |
| Wheight          | Peso                             |

## 2.2 Funcionamento Painel de LED's RGB

O painel escolhido tem como característica principal a correspondência de ativação dos LED's na matriz não linear e com uma *refresh rate* de ¼. Na Tabela 2 estão apesentadas as características técnicas do painel escolhido.

Tabela 2 - Características Técnicas Painel SDM P10RGB

|                  |                            |
|------------------|----------------------------|
| Module Type      | Outdoor LED Display Module |
| LED Type         | 3-in1 SMD                  |
| Display Color    | Red, Green, Blue           |
| Module Dimension | 320mm x 160mm              |
| Module Pixel     | 512 pixel / module         |
| Pixel Pitch      | 10mm                       |
| Resolution       | 10000dot / m2              |
| Brightness       | 8000nit                    |
| Gray Scale       | 4096 Level for each color  |
| Weight           | 1000g / Module             |

Para a alimentação do painel é necessária uma fonte de alimentação que respeite a Equação 1. Para o painel escolhido foi selecionada uma fonte de alimentação de 5 Volts e 7 Amperes, que apesar de não cumprir os requisitos que permita a utilização do painel na sua plenitude, adequa-se para o desenvolvimento do projeto, mantendo uma boa relação custo *versus* performance.

Equação 1 - Cálculo de corrente Máxima Painel de LED's RGB

$$I(A) = ILed * n^{\circ} Led$$

O painel utilizado é composto por 512 LEDs RGB distribuídos numa matriz de 16 linhas e 32 colunas. Para ativar os LED's da matriz o painel possui 8 *shift registers* de 64 bits que têm os dados de cada cõda matriz de LED's e estão divididos entre a metade superior do painel e a metade inferior, cada uma é composta por 8 linhas e 32 colunas. Os *shift register's* são controlados por um sinal de *latch* (LAT), *active high*, e um sinal de *output enable* (OE), *active low*.

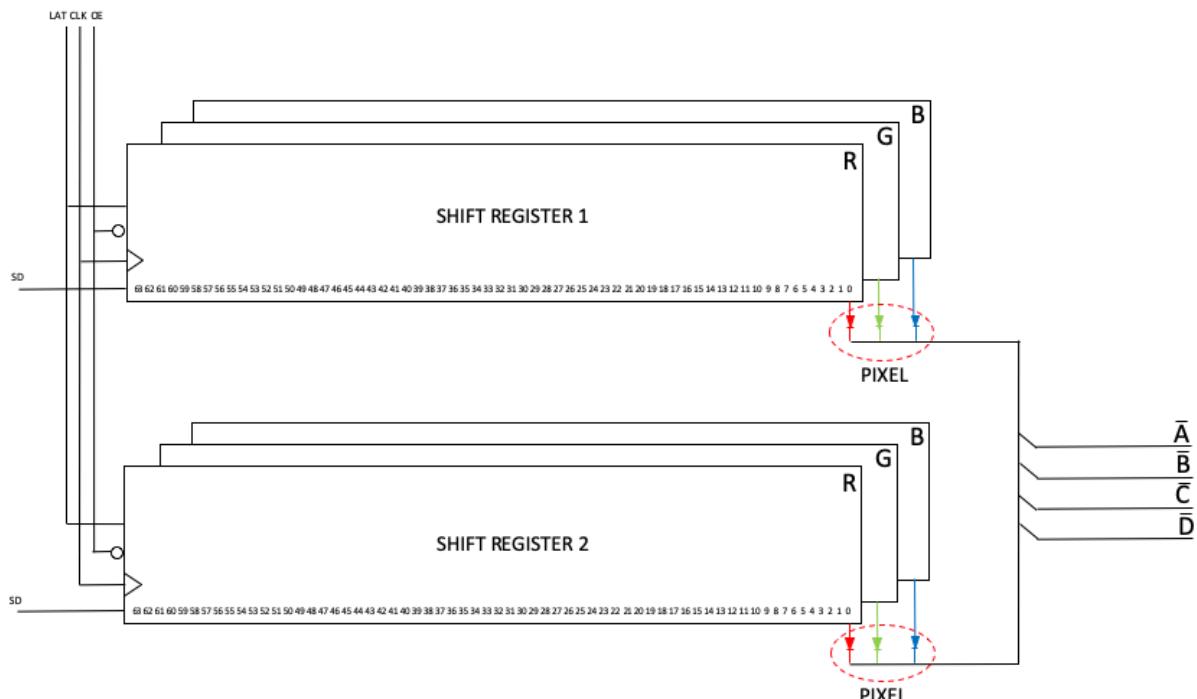


Figura 7 - Esquema Eléctrico Shift Register

Ao colocar o sinal LAT com o valor lógico '0' e o sinal OE com o valor lógico '1' permite que os dados fiquem à saída do *shift register* e ao mesmo tempo desativar a saída de modo a que nenhum LED fique ativo, este estado permite que o painel seja atualizado sem que seja visualizado a deslocação dos bits e consequente ativação dos LEDs. Os *shift register* presentes no painel seguem a filosofia FIFO, o primeiro bit a entrar é o primeiro a sair.

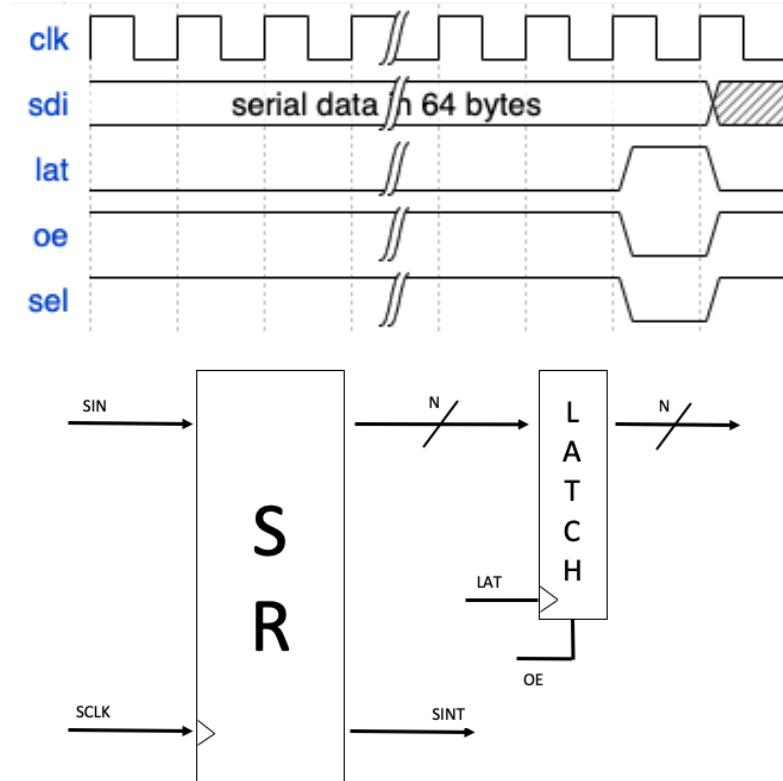


Figura 8 - Timing Waveform e Circuito Equivalente

As linhas são ativas com a utilização de 4 seletores, do tipo *active low*, identificados com A, B, C e D, onde cada um é responsável pela ativação de 4 linhas do painel. A distribuição das linhas é feita com 2 linhas na metade superior e 2 linhas na metade inferior do painel, que correspondem às 64 saídas do *shift register* de cada metade do painel, com a respetiva cor selecionada. São selecionadas 2 linhas de cada metade do painel devido à distribuição das 64 saídas do *shift register* no painel que têm de ser repartidas por 2 linhas de 32 colunas, estas saídas estão distribuídas no painel de uma forma não linear como se ilustra na

Figura 9.

|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |    |    |    |    |    |    |   |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|----|----|----|----|----|----|---|
| 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | A |
| 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | B |
| 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | C |
| 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | D |
| 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 7 | 6 | 5  | 4  | 3  | 2  | 1  | 0  | A |
| 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 7 | 6 | 5  | 4  | 3  | 2  | 1  | 0  | B |
| 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 7 | 6 | 5  | 4  | 3  | 2  | 1  | 0  | C |
| 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 7 | 6 | 5  | 4  | 3  | 2  | 1  | 0  | D |
| 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | A |
| 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | B |
| 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | C |
| 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | D |
| 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 7 | 6 | 5  | 4  | 3  | 2  | 1  | 0  | A |
| 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 7 | 6 | 5  | 4  | 3  | 2  | 1  | 0  | B |
| 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 7 | 6 | 5  | 4  | 3  | 2  | 1  | 0  | C |
| 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 7 | 6 | 5  | 4  | 3  | 2  | 1  | 0  | D |

Figura 9 - Distribuição de saídas Shift Register com correspondência de ativação

## 2.3 Arquitetura do Microcontrolador

Para a seleção do microcontrolador efetuou-se uma pesquisa e verificou-se a existência de uma grande diversidade no mercado, mas esta lista pode ser restringida devido ao projeto ter como requisitos a utilização de um microcontrolador de baixo custo, com ligação *Wifi* e *Bluetooth*. Perante estas restrições destaca-se duas famílias de microcontroladores que são:

- i. família Raspberry Pi (3);
- ii. família Espressif (4);

Opta-se pela utilização da família Espressif, pois a família Raspberry Pi apresenta um custo superior e um maior consumo de energia, quando comparadas. Dentro da família Espressif pode-se optar por:

- i. ESP 8266 (5);
- ii. ESP32 (6);

Opta-se pela opção ii) pois é o único com suporte de ligação por *Bluetooth*.

O microcontrolador da Espressif ESP32 tem num circuito integrado o microcontrolador, Wifi e Bluetooth.

A família de microcontroladores ESP32 tem diferentes módulos como o WROOM, SOLO, WROVER e PICO. As principais características destes módulos são descritas na Tabela 3.

Tabela 3 - Diferentes Módulos ESP32

| Module              | Chip         | Flash, MB   | PSRAM, MB | Ant. | Dimensions, mm  |
|---------------------|--------------|-------------|-----------|------|-----------------|
| ESP32-WROOM-32      | ESP32-D0WDQ6 | 4           | -         | MIFA | 18 × 25.5 × 3.1 |
| ESP32-WROOM-32D     | ESP32-D0WD   | 4, 8, or 16 | -         | MIFA | 18 × 25.5 × 3.1 |
| ESP32-WROOM-32U     | ESP32-D0WD   | 4, 8, or 16 | -         | U.FL | 18 × 19.2 × 3.1 |
| ESP32-SOLO-1        | ESP32-S0WD   | 4           | -         | MIFA | 18 × 25.5 × 3.1 |
| ESP32-WROVER (PCB)  | ESP32-D0WDQ6 | 4           | 8         | MIFA | 18 × 31.4 × 3.3 |
| ESP32-WROVER (IPEX) | ESP32-D0WDQ6 | 4           | 8         | U.FL | 18 × 31.4 × 3.3 |
| ESP32-WROVER-B      | ESP32-D0WD   | 4, 8, or 16 | 8         | MIFA | 18 × 31.4 × 3.3 |
| ESP32-WROVER-IB     | ESP32-D0WD   | 4, 8, or 16 | 8         | U.FL | 18 × 31.4 × 3.3 |

Para o desenvolvimento do projeto o módulo utilizado foi o ESP32-WROOM-32, integrado numa placa desenvolvida pela Adafruit Huzzah Feather (7). Esta placa tem incorporado um conversor USB-to-Serial, *bootloader reset* automático, conexão para bateria *Lithium Ion/Polymer* e todos os GPIO disponíveis no microcontrolador.



Figura 11 - Módulo ESP32-WROOM-32



Figura 10 - Adafruit Huzzah32 - ESP32 Feather

Do módulo da Espressif escolhido para o projeto utilizam-se as funcionalidades:

- Memória Flash 4MB;
- Memória RAM 512KB;
- 14 GPIO;
- Conectividade sem fio:
  - Wifi: 802.11 b/g/n
  - Bluetooth: v4.2 BR/EDR

## 2.4 Ambiente de Desenvolvimento Integrado

Para o desenvolvimento do software existem vários tipos de ambientes, tais como o ESP-IDF, MicroPython, Arduino e o NodeMCU. Opta-se pela plataforma Arduino (8), por possuir maior número de documentação, uma maior comunidade de suporte e disponibilizar um maior número de bibliotecas.

Esta plataforma é caracterizada pelo seu conceito de *sketch*, que é o *upload* da unidade de código para ser processado no microcontrolador.

A função de entrada *Main* é substituída por uma função de inicialização de periféricos e variáveis com o nome *Setup* que é executada no arranque do *sketch* e uma função *Loop*, que como o seu nome sugere repete-se consecutivamente.

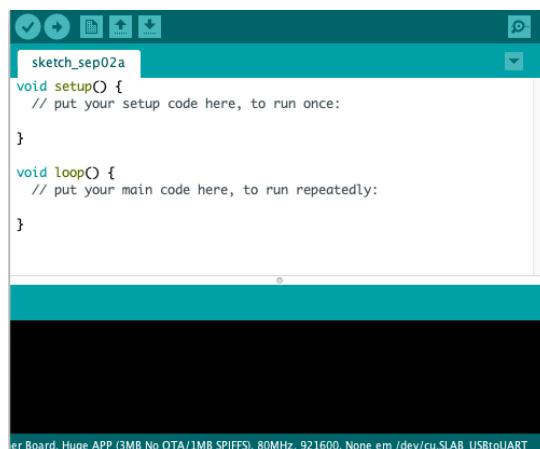


Figura 12 - Sketch Arduino

### 3. Implementação

Neste capítulo descreve-se a implementação no microcontrolador do sistema proposto. Para o desenvolvimento do *software* foi dividido o código em camadas lógicas, sendo esta uma forma de dividir responsabilidades, integrar bibliotecas nativas e tornar a aplicação portável para outras plataformas. O código desenvolvido neste projeto segue a orientação representada na

Figura 13.

- Camada Aplicacional – Onde é aplicada toda a lógica da aplicação e o processamento de eventos. Esta camada é responsável por fazer a seleção do tipo de ligação, *Wifi* ou *Bluetooth*, e o respetivo processamento mediante a ligação escolhida.
- Camada Intermédia – Nesta camada estão presentes as bibliotecas que foram desenvolvidas e bibliotecas que pertencem à *Framework* integrada no IDE, que permitem a comunicação da aplicação com o painel LED, comunicação com uma rede *Wifi* usando protocolo 802.11 b/g/n, comunicação *Bluetooth* usando protocolo v4.2 BR/EDR e acesso à EEPROM.
- Camada Física – Esta camada é constituída pelo microcontrolador ESP32 e por um painel de LED's RGB.

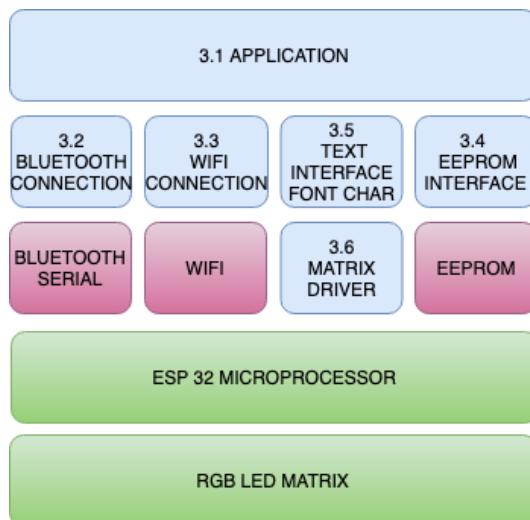


Figura 13 - Diagrama Lógico da aplicação e hardware

### 3.1 Camada Aplicacional

O sistema tem dois modos de funcionamento, o modo *Bluetooth* e o modo *Wifi*. O modo *Bluetooth* utiliza uma ligação série para realizar a comunicação entre o utilizador e o microcontrolador. O modo *Wifi* permite a utilização de um *Browser* para comunicação entre o utilizador e o microcontrolador. A aplicação começa por inicializar o hardware dentro da função *Setup*, na qual se efetua uma leitura da EEPROM com o propósito de se obter o ssid e a password da última ligação *Wifi* estabelecida. De seguida é efetuada a inicialização do painel, onde são definidos os pinos de output do microcontrolador que irão fazer a conexão ao painel e inicializa os *shift registers* do painel com valor '0'. Por fim de modo a permitir o refreshamento contínuo da matriz do painel juntamente com a execução da aplicação é criada uma tarefa (task), como demonstrado no fluxograma presente na Figura 14.

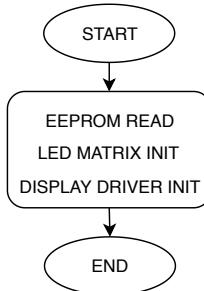


Figura 14 - Fluxograma Função *Setup*

Após o *Setup* entra em execução a função *Loop*, a qual como é referido no capítulo 2.4 é chamada continuamente, onde são executadas as ações do fluxograma presente na Figura 15 a cada iteração.

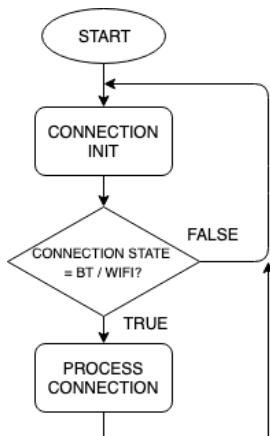


Figura 15 – Função *Loop*

Em cada iteração é analisado o estado de uma variável e executada a ação correspondente. Quando não encontra nenhuma ligação válida, por defeito executa a função *Connection Init*. Esta função é responsável por estabelecer uma tentativa de ligação por *Wifi* ou por *Bluetooth*, como representado na

Figura 16.

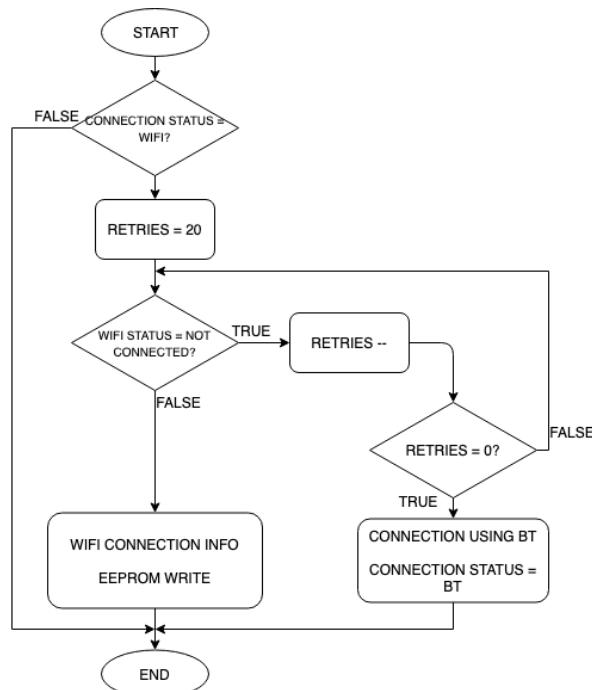


Figura 16 – Fluxograma da Função *Connection Init*

Esta função começa por verificar se existem condições para realizar uma ligação *Wifi*, a verificação é feita durante vinte iterações. Durante as tentativas se for estabelecida uma ligação com sucesso, apresenta os dados da ligação estabelecida e guarda na EEPROM os valores do *ssid* e da *password* correspondentes à ligação.

Caso sejam realizadas as vinte iterações e não tiver sido estabelecida uma ligação por *Wifi*, estabelece-se uma ligação *Bluetooth* de modo a permitir a configuração de uma ligação *Wifi*, cujo retorno afeta a variável de estado com o valor correspondente ao tipo de ligação estabelecida.

### 3.2 Ligação Bluetooth

A comunicação *Bluetooth* tem início com o emparelhamento com um dispositivo que irá permitir uma comunicação série, onde se configura o tipo de comunicação com o painel, esta poderá ser feita por uma ligação *Bluetooth* ou uma ligação *Wifi*, cujo fluxograma é ilustrado na

Figura 17.

Após o emparelhamento com o sucesso, é estabelecida uma comunicação série que permite a introdução de comandos entre o dispositivo e o microcontrolador.

Neste estado a aplicação aguarda pela inserção de um dos seguintes comandos.

- Comando “SSID<ssid-da-rede>”;
- Comando “PASS<password-da-rede>”;
- Comando “BLUE”;

Para inicializar uma configuração da ligação *Wifi*, começa-se por inserir o comando “SSID” seguido do nome da rede, é verificado se os primeiros quatro caracteres são “SSID” e no caso de se verificar esta condição os caracteres seguintes são guardados localmente e passa para o estado seguinte onde aguarda a inserção do comando “PASS” seguido da password da rede. Neste estado é realizado o mesmo procedimento que no estado anterior, mas é feita a verificação para a combinação de “PASS”, mais uma vez se esta condição se verificar os caracteres seguintes são guardados na localmente para posterior processamento.

Caso se pretenda realizar uma ligação por *Bluetooth* deve ser inserido o comando “BLUE” e é passada a informação à aplicação que a comunicação com o painel será realizada usando a comunicação série disponibilizada pela ligação *Bluetooth*.

Durante o processo de seleção da comunicação a utilizar o painel apresenta uma indicação visual através de dois LED's de estado e durante este processo os LED's estão no estado “Aguarda Ligação”, cujas cores são descritas na Tabela 4.

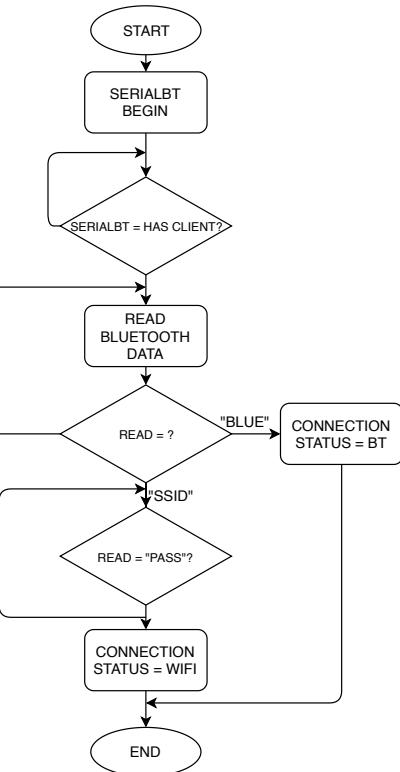


Figura 17 - Fluxograma Função Connect Using BT

Tabela 4 - LED's de Estado

| CÔR LED STATE | CÔR LED BLUETOOTH | DESCRIÇÃO               |
|---------------|-------------------|-------------------------|
| VERMELHO      | VERMELHO          | SISTEMA A INICIALIZAR   |
| VERMELHO      | ROxo              | AGUARDA LIGAÇÃO         |
| Azul          | roxo              | LIGAÇÃO BLUETOOTH ATIVA |

No caso de se optar pela ligação Bluetooth como forma de comunicação com o painel a aplicação fica a aguardar a inserção de um dos seguintes comandos:

- Formato query string para apresentação de conteúdos no painel:
  - "Linha1=<mensagemlinha1>&Linha2=<mensagemlinha2>&Cor1=<numerocor>&Cor2=<numerocor>";
- Comando “WIFISTART”, com a inserção deste comando volta ao estado de configuração descrito no início do capítulo 3.2;

Após o processamento de todos os dados, entra em execução a função *Scroll Display*, que tem como função tratar o deslocamento dos caracteres horizontalmente e verticalmente e que é descrita no capítulo 3.5.

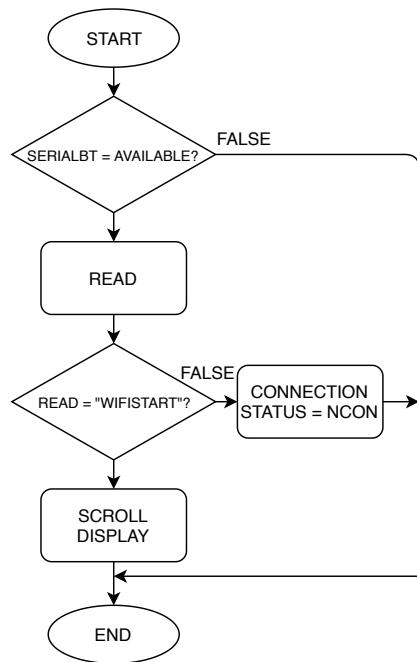


Figura 18 - Fluxograma da Função Process BT

### 3.3 Ligação Wifi

A ligação Wifi permite a comunicação entre o utilizador e o microcontrolador na mesma rede usando um *browser* com o qual se pode alterar o conteúdo a apresentar no painel.

Esta ligação pode ser inicializada de forma automática durante a inicialização da aplicação, através da leitura das credenciais da EEPROM ou através da configuração realizada com uma ligação *Bluetooth*. Inicialmente é verificada se a ligação *Wifi* está ativa, no caso da ligação se ter perdido retorna à configuração da ligação descrito no capítulo 3.2.

Com o estabelecimento da ligação *Wifi* à rede local é obtido o endereço IP, que é indicado no painel e as credenciais da rede são escritas na EEPROM.

Após a ligação *Wifi* à rede, o servidor aguarda pedidos no porto 80, ao receber pedidos através deste porto é criado um cliente no qual se procede à leitura dos dados caracter a caracter enviados pelo pedido, como ilustrado na Figura 19. Após a leitura dos dados é realizado um *parse* sobre a *string* lida, de modo a extrair o conteúdo para linha um e linha dois e respetivas cores para ser apresentado no painel. Nesta ligação o *Scroll Display* está sempre ativo, pois desta forma garante-se que a apresentação continua do texto no painel. A ligação tem a duração da ligação estabelecida pelo cliente.

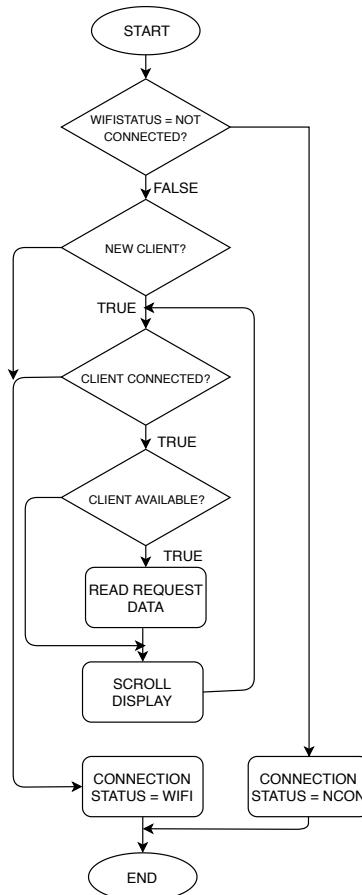


Figura 19 - Fluxograma *Wifi Process*

### 3.4 EEPROM Interface

De modo a garantir a persistência das credenciais da ligação *Wifi* foi desenvolvido uma interface que permite guardar as credenciais da memória flash. Esta funcionalidade permite ler as credenciais da última ligação *Wifi* estabelecida e escrever na memória novas credenciais da ligação *Wifi*.

A EEPROM tem definida uma dimensão de 128 *bytes*, onde são guardadas as credenciais da rede *Wifi*, dos quais 64 *bytes* são atribuídos ao SSID e os restantes à *Password*. O primeiro *byte* de cada bloco contém a dimensão em *bytes* do valor armazenado, como apresentado na Figura 20.

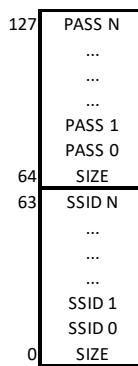


Figura 20 - Mapa de Memória EEPROM

### 3.5 Interface de Texto

A conversão dos caracteres em *bitmaps* é da responsabilidade do módulo *Text Interface*. Neste módulo está presente a fonte dos caracteres que é convertida para um bitmap com base na tabela ASCII do carácter, como representado na

Figura 21. Sendo que existem várias bibliotecas de fonte disponíveis na internet, optou-se por utilizar uma já desenvolvida, a qual possui os caracteres necessários para a representação de texto no painel. A escrita de um carácter no *buffer* de texto consiste na cópia do *bitmap* do carácter ponto a ponto.

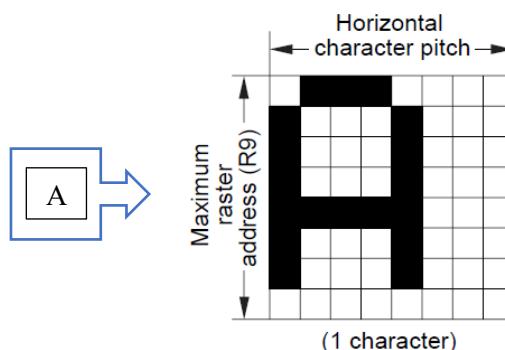


Figura 21 - Bitmap Caracter 'A'

Nesta interface é realizada a gestão do deslocamento horizontal, quando existem mais caracteres que colunas disponíveis e vertical, quando existem menos linhas disponíveis que as que são pretendidas pelo utilizador. Na Figura 22 está ilustrado o como é feito o deslocamento, onde a dimensão da janela de visualização é sempre igual, deslocando-se em toda a dimensão da matriz.



Figura 22 - Exemplo de Janela com Deslocamento de Texto

Como se verifica no fluxograma da Figura 23, quando o deslocamento é efetuado é verificado se decorreu o tempo definido após a última iteração. Com este processo é possível manter o ritmo de deslocação do texto no painel de LED's, deste modo é atualizada a posição da matriz horizontalmente e verticalmente. Após a atualização das variáveis de posição, inicia-se o processo de conversão de matrizes.

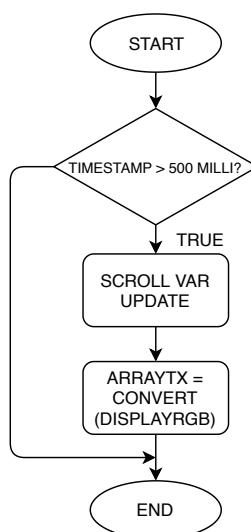


Figura 23 - Fluxograma Função Scroll Display

### 3.6 Controlador de matriz de LED's

Como existem diversos tipos de painéis, com diferentes tipos de matrizes e diferentes distribuições de LED's no painel, tal como demonstrado na Figura 9, onde a distribuição dos LED's na matriz não segue uma distribuição linear, foi necessário criar um módulo Matrix Driver, cujo fluxograma está representado na Figura 24 que tem como função a conversão de matrizes e o envio da matriz final para o painel de LED's.

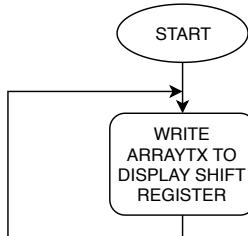


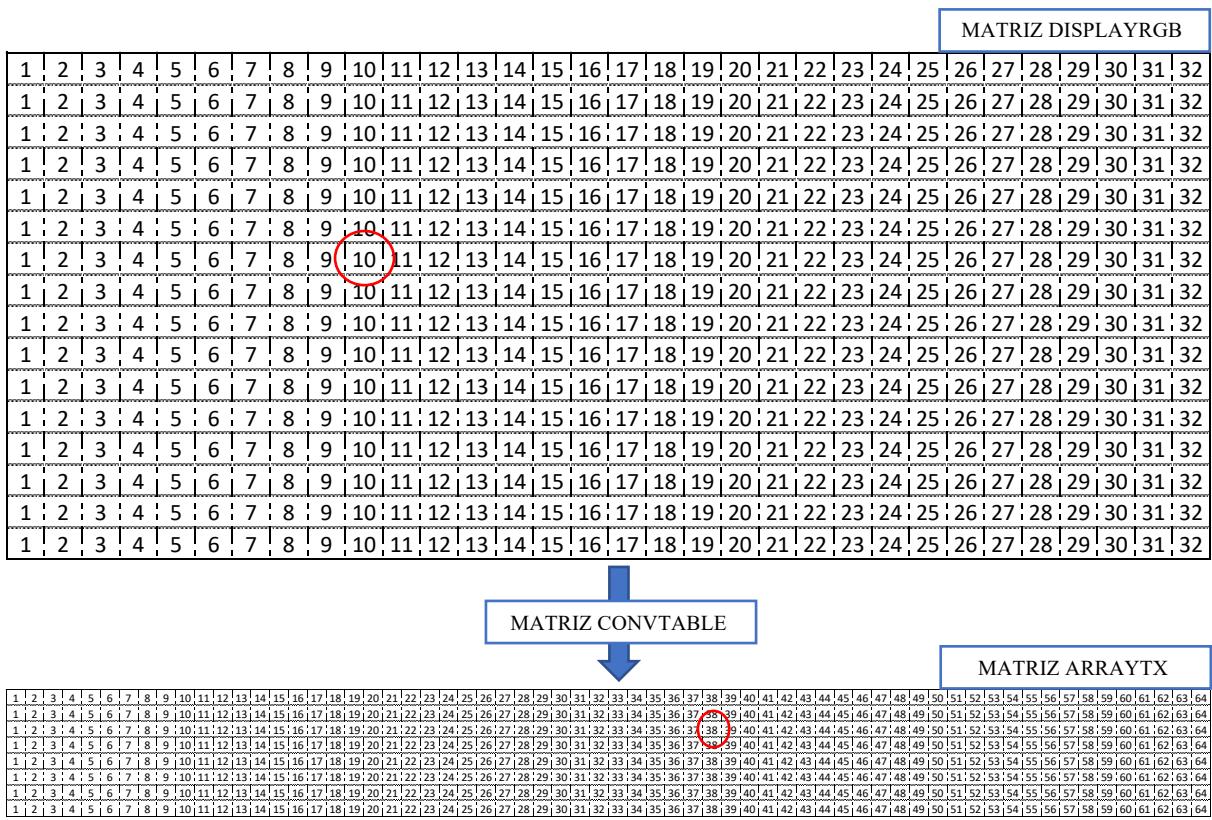
Figura 24 - Fluxograma Matrix Driver

Foi necessário desenvolver uma matriz (*displayRGB*) que guarde o bitmap correspondentes ao buffer de caracteres a exibir no painel. A dimensão desta matriz define o número de caracteres visíveis no painel com base na dimensão do *bitmap* de cada caracter. Caso a sequência de *bitmaps* for superior à dimensão do painel de LED's, o driver efetua um deslocamento sobre esta matriz de forma a mostrar os restantes *bitmaps* presentes nesta matriz.

Como referido no capítulo 2.2 devido à estrutura do painel de LED's, não é possível fazer um varrimento direto da matriz *displayRGB* para o painel, para esse feito é necessário realizar uma conversão para uma matriz intermédia (*arrayTX*) com base numa tabela de máscaras (*convTable*). A leitura da matriz *displayRGB* é feita a partir do índice linha / coluna.

Com a matriz *convTable* é possível mapear a matriz *displayRGB* na matriz *arrayTX*, como por exemplo o mapeamento da posição, na matriz *displayRGB*, linha 7 coluna 10, corresponde na matriz *ArrayTX*, à linha 3 coluna 38, como apresentado na Figura 25. Antes de se desenvolver a matriz intermédia de conversão, começou por se desenvolver um algoritmo que fizesse diretamente essa função, e verificou-se que o algoritmo encontrado obrigava o programa a fazer vários ciclos fazendo com o programa ficasse mais lento e pesado. Cada elemento da matriz *convTable* é do tipo inteiro (*Int*) no qual permite guardar 2 *bytes* em que cada um dos *bytes* corresponde a um par linha/ coluna da matriz *arrayTX*, onde na parte alta é guardado o valor da linha correspondente e na parte baixa a posição da coluna que está atribuída.

Após a matriz *arrayTX* estar completa estão reunidas as condições necessárias para o envio dos dados para o painel. Devido às características dos *shift registers* do painel, é enviado o conteúdo da matriz *arrayTX* em oito vectores de 64 *bytes* correspondentes a oito *shift registers*.



*Figura 25 – Representação Conversão de Matriz displayRGB para matriz arrayTX*

## 4. Resultados Experimentais

Neste capítulo é apresentado o resultado final do trabalho desenvolvido. O programa desenvolvido ocupa 1 356 807 bytes, divididos entre o código do programa e as variáveis globais. O rascunho do programa está inserido na memória de armazenamento do programa e tem uma ocupação de 41% o que corresponde a 1 291 235 bytes. As variáveis globais estão inseridas na memória dinâmica e usam 20% do espaço disponível, o que corresponde a uma ocupação de 65 572 bytes, este espaço de memória foi otimizado sempre possível, guardando as constantes em memória de programa em vez da memória dinâmica.

São abordadas duas formas de se poder apresentar texto no painel, usando uma ligação *Bluetooth* ou utilizando uma ligação *Wifi*. A ligação por *Wifi* pode ser configurada utilizando uma ligação por *Bluetooth*.

### 4.1 Ligação *Bluetooth*

Ao não existir nenhuma ligação *Wifi* ativa, o sistema inicializa a antena *Bluetooth* e permite o emparelhamento com um dispositivo externo permitindo uma comunicação série com o qual é possível trocar mensagens, na Figura 26 apresenta-se um exemplo de como é executado o emparelhamento. Para a realização deste projeto e para o emparelhamento e consequente ligação série, optou-se pela utilização da aplicação *BluTerm*, este *software* foi obtido gratuitamente na plataforma *Apple Store*.

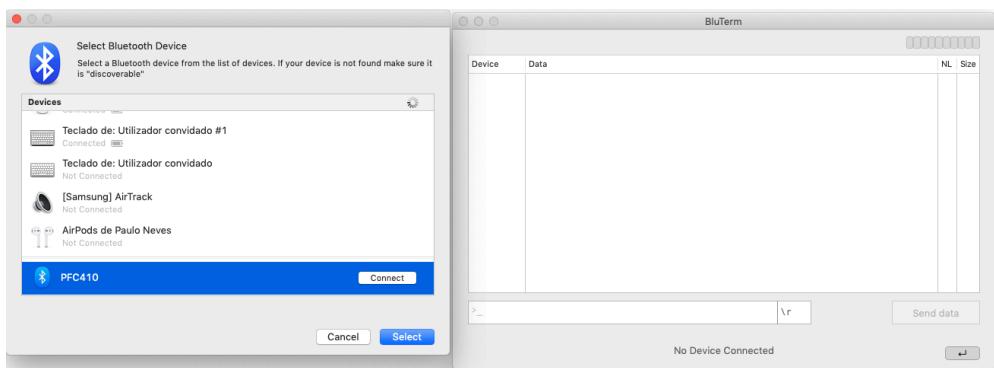


Figura 26 - Emparelhamento ESP32 com Dispositivo Externo utilizando a Aplicação *BluTerm*

O painel tem dois LED's ativos que durante o período de estabelecimento de ligações que dão o estado da ligação, de acordo com a Tabela 4, onde a posição linha 1 coluna 1 dá a indicação se a ligação por *Wifi* está estabelecida (*LED Bluetooth*) e a posição linha 1 coluna 2 (*LED State*) dá dois tipos de indicação, se aguarda o tipo de ligação ou se a ligação por *Bluetooth* está ativa. Verifica-se na Figura 27 que o painel muda a cor do *LED State* para rosa, o que significa que aguarda a inserção do tipo de ligação pretendida.

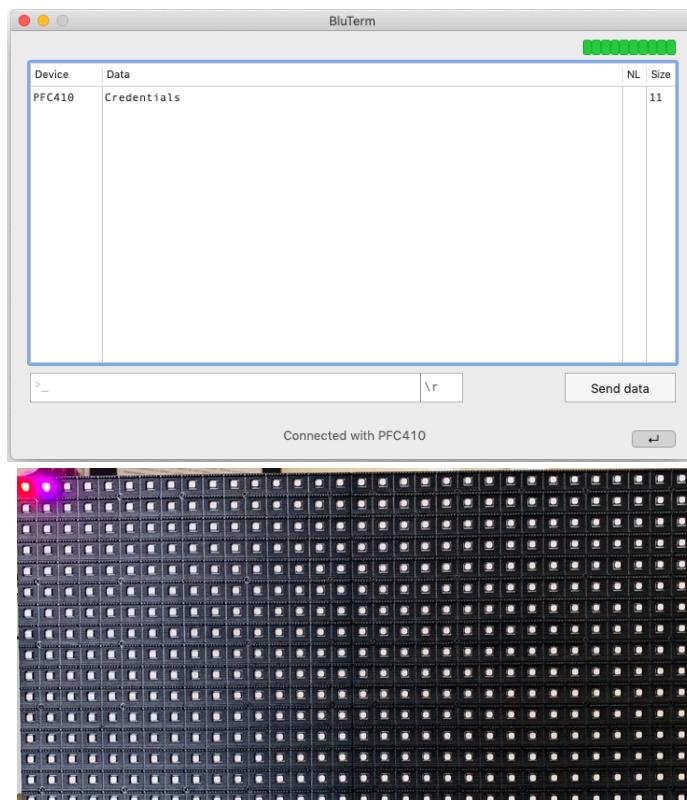
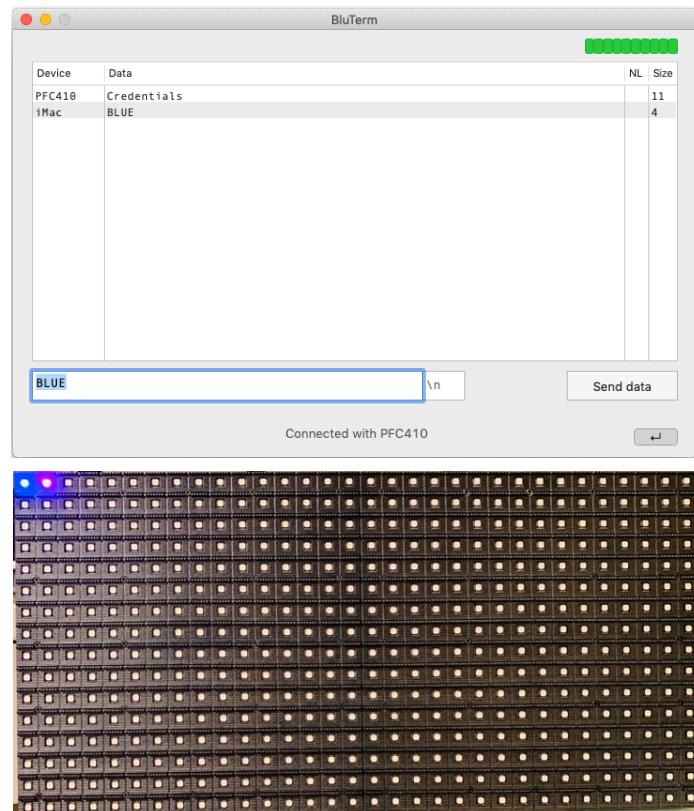


Figura 27 - Aguarda inserção do tipo de ligação

Com o emparelhamento concluído é enviada uma mensagem para o dispositivo a informar que aguarda a inserção dos comandos, que neste caso será o comando “BLUE” para a reprodução de texto no painel utilizando a ligação *Bluetooth*.

Ao selecionar a ligação por *Bluetooth*, há uma mudança de cor do LED *Bluetooth* para azul, que está representado na Figura 28.



*Figura 28 - Ligação por Bluetooth ativa*

Após estabelecida a ligação por Bluetooth é possível enviar a mensagem que se pretende mostrar no display. A mensagem verificada na Figura 29 é composta pelos seguintes parâmetros:

“Linha1=38714&Linha2=PFC&Cor1=1&Cor2=6”

- Linha 1 = “38714”;
- Linha 2 = “PFC”;
- Cór 1 = “1”;
- Cór 2 = “6”;

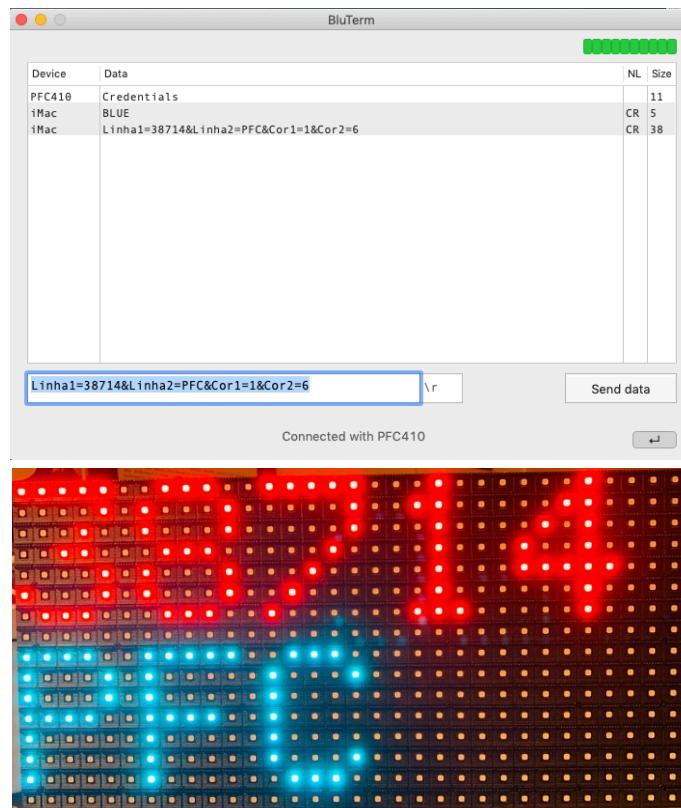


Figura 29 - Mensagem enviada para painel via Bluetooth

#### 4.2 Ligação Wifi via Bluetooth

No caso de se optar por uma ligação por *Wifi* deverão ser introduzidas as credenciais da rede *Wifi* que se pretende conectar após o emparelhamento do ESP32 com o dispositivo *Bluetooth* externo. Quando validada a ligação é indicado no display o IP que foi atribuído ao ESP32 pelo router, indicando assim que a ligação *Wifi* está ativa, como mostra a Figura 30. Para o exemplo foram inseridas as credenciais:

- SSID – “SSIDMEO-8E93A0”;
- PASS – “PASS0cb4dc7f2b”;

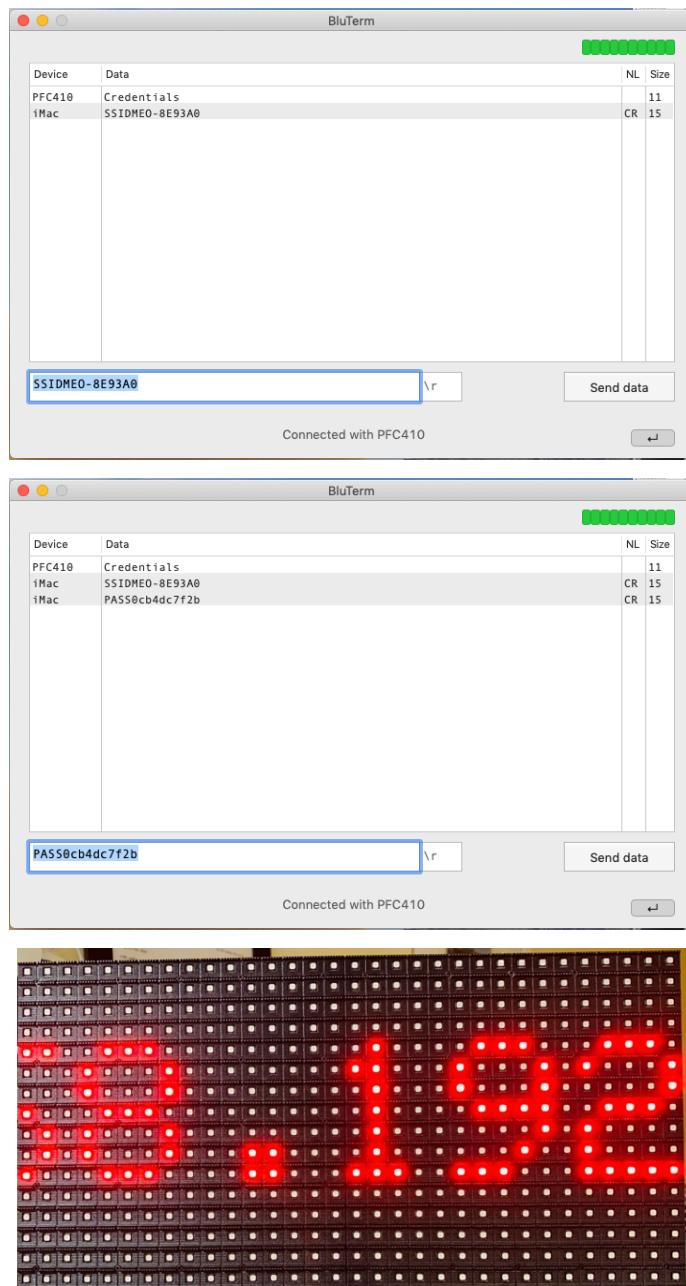


Figura 30 - Ligação Wifi via Bluetooth com indicação endereço IP do LED do painel

Estabelecida a ligação *Wifi* é então possível efetuar a comunicação por HTTP, onde serão registadas e enviadas as mensagens a exibir no painel. Para estabelecer uma ligação com o ESP32 é necessário inserir o IP indicado no painel, no *browser* e registar as mensagens nos campos disponíveis na página *Web*.

Após a ligação do ESP32 através da rede local é apresentada uma página no *browser*, idêntica à apresentada na Figura 31.

Nesta página *Web* deve ser inserido o texto na linha correspondente, selecionar a cor pretendida e fazer *submit*. Esta ação efetua um pedido HTTP, os dados são enviados para o ESP32 e são projetados no painel.

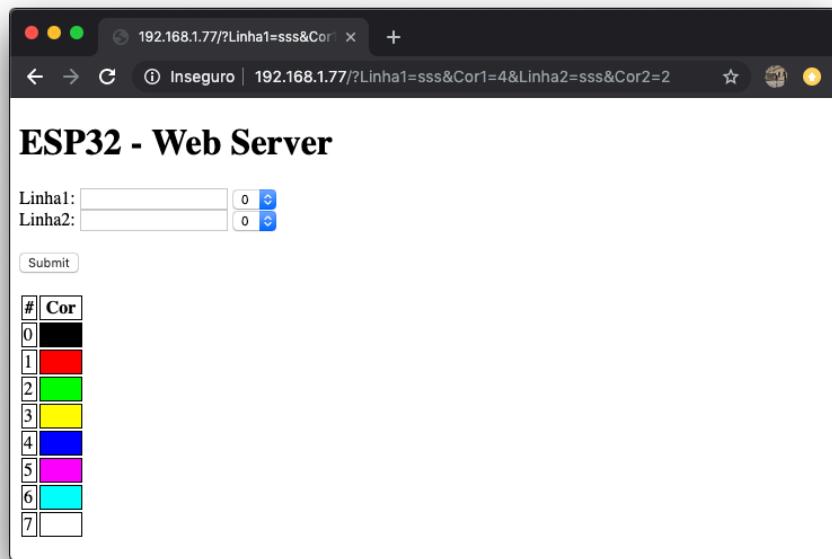


Figura 31 - Página Web Server ESP32

Para o exemplo apresentado na Figura 32 foram inseridos os dados:

- Linha1: “ISEL” com a côn “7”;
- Linha2: “2019” com a côn “3”;

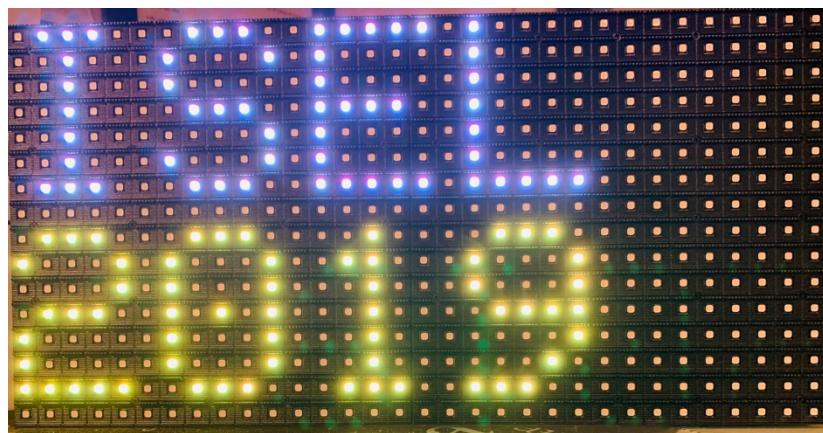


Figura 32 - Output pedido HTTP Painel de LED's

Se durante uma ligação por *Bluetooth* se optar por se estabelecer uma ligação *Wifi*, deve ser inserida a mensagem “WIFISTART”, com apresentado na Figura 33 e seguir os passos indicados anteriormente neste capítulo.

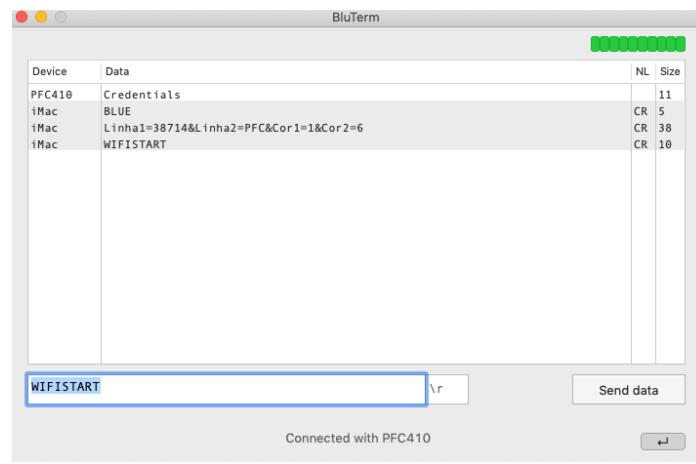


Figura 33 - Comando seleção ligação Wifi

## 5. Conclusão

Este trabalho consistiu no desenvolvimento de um sistema de controlo para painéis de LED's RGB, utilizando um microcontrolador ESP32.

Foram atingidos com sucesso os objetivos estabelecidos para este projeto com a implementação de uma interface de comunicação com o painel de LED's RGB.

Foram conseguidas ligações sem fios, ligação *Wifi* e ligação *Bluetooth*, através do desenvolvimento de interfaces capazes de sustentar este tipo de comunicações e o desenvolvimento de uma interface de comunicação com a EEPROM.

Para a exibição de texto no painel foram implementadas bibliotecas de texto com o propósito de criar *bitmaps* correspondentes aos caracteres da tabela ASCII.

Por fim foi desenvolvida uma página Web para apresentação no *browser* para desta forma servir de interface entre o utilizador e o painel quando estabelecida uma ligação *Wifi*.

### 5.1 Trabalho Futuro

Com a conclusão do trabalho, há ainda a possibilidade de fazer evoluir este trabalho. Pode ser multiplicado o número de cores disponíveis no painel, passando de 8 cores para uma solução *full-color*, utilizando a técnica de *Pulse Width Modulation*.

Pode ser implementada a funcionalidade de reprodução de imagens *bitmaps*.

Escalar a dimensão do projeto aumentando o número de painéis interligados num único microcontrolador e a sincronização entre os vários painéis. Também se pode sincronizar vários microcontroladores de modo a poder criar uma LED *Wall*.

## 6. Bibliografia

1. Microcontrolador. [Online] <https://pt.wikipedia.org/wiki/Microcontrolador>.
2. Module, Sanders Full Color LED. P10 RGB Module.
3. Raspberry Pi. [Online] <https://www.raspberrypi.org/>.
4. Espressif. [Online] <https://www.espressif.com/>.
5. ESP8266. [Online] <https://www.espressif.com/en/products/hardware/esp8266ex/overview>.
6. ESP32. [Online] <https://pt.wikipedia.org/wiki/ESP32>.
7. Adafruit Feather Huzzah. [Online] <https://www.adafruit.com/product/3405>.
8. Arduino Software. [Online] <https://www.arduino.cc/en/main/software>.
9. Programming ESP32 Board with Arduino IDE. [Online] <https://circuitdigest.com/microcontroller-projects/programming-esp32-with-arduino-ide>.
10. IDE. [Online] [https://pt.wikipedia.org/wiki/Ambiente\\_de\\_desenvolvimento\\_integrado](https://pt.wikipedia.org/wiki/Ambiente_de_desenvolvimento_integrado).
11. Macroblock. *16-Channel PWM-Embedded Led Driver*. 2006.

## 7. Anexos

### Anexo A – Esquema ESP32 Adafruit Feather

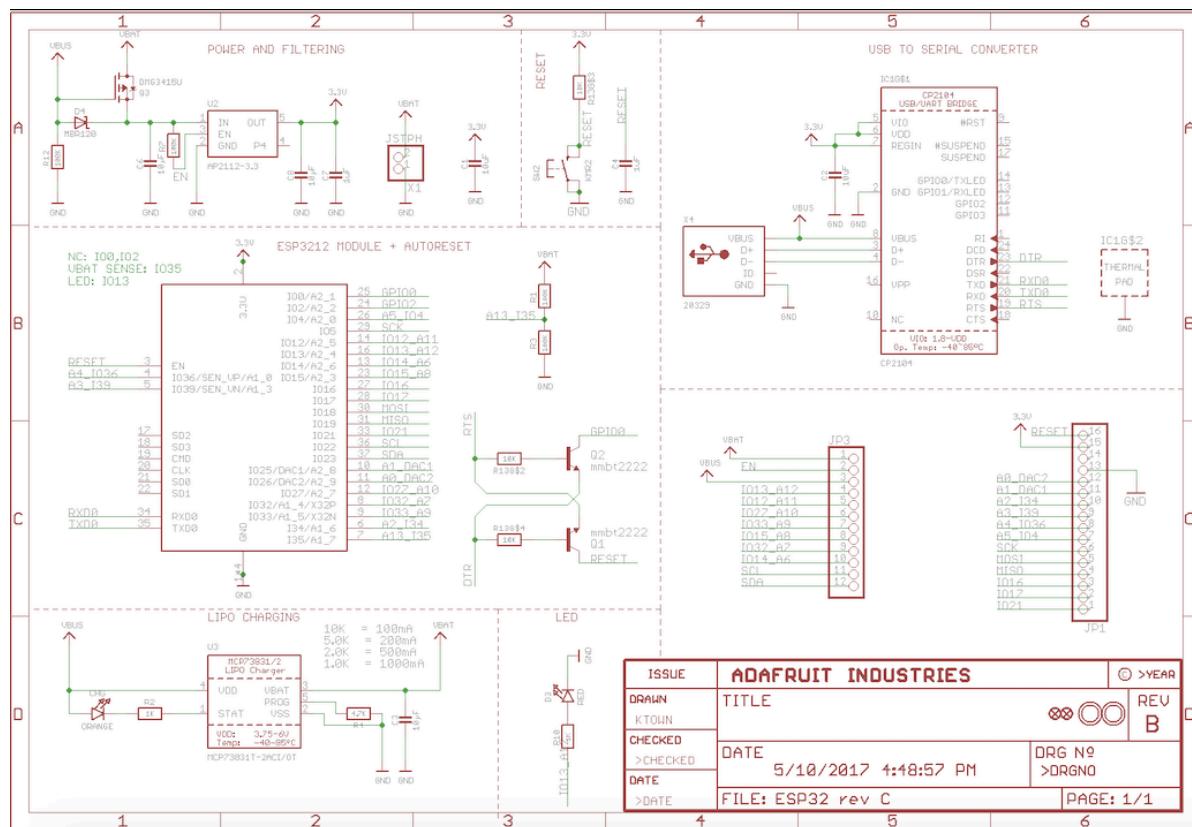


Figura 34 - Esquema Eléctrico ESP32 Adafruit Feather

**ZERYNTH**  
PINOUT DIAGRAM

DO NOT USE D6 TO D11

PWM IS ENABLED ON EVERY DIGITAL PIN

ICU NOT SUPPORTED

ADC ON PINS D4, D12, D13, D14, D15, D25, D26, D27  
CAN BE READ ONLY WITH WI-FI NOT STARTED

## Adafruit Huzzah32

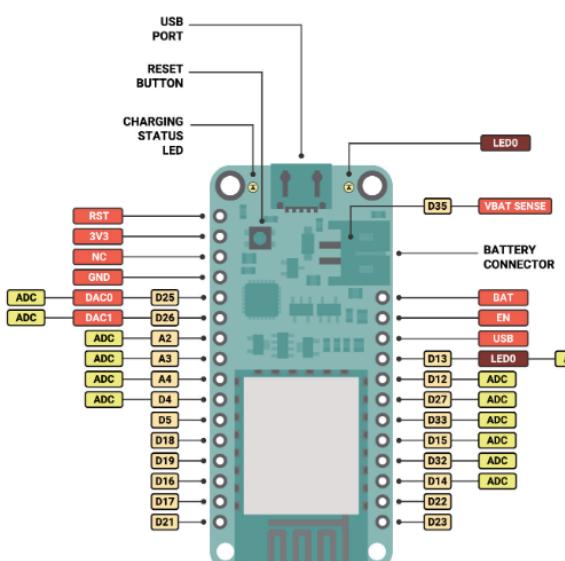


Figura 35 - Pinout ESP32 Adafruit

## Anexo B – Esquema Eléctrico

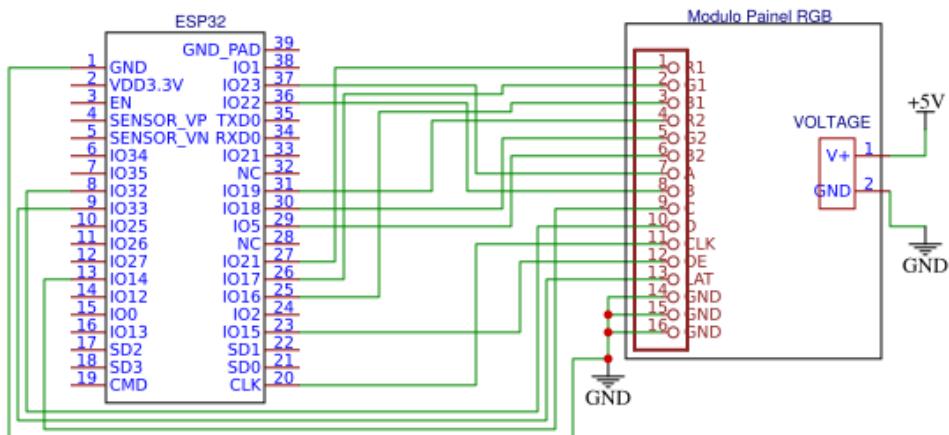


Figura 36 - Esquema Eléctrico