

**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

Maio 2019

# Índice de conteúdos

[Lista de Figuras i](#_Toc9181412)

[Lista de Tabelas ii](#_Toc9181413)

[Acrónimos iii](#_Toc9181414)

[1. Introdução 1](#_Toc9181415)

[1.1 Enquadramento 2](#_Toc9181416)

[1.2 Descrição do projeto 3](#_Toc9181417)

[1.3 Objetivos 4](#_Toc9181418)

[2. Análise Requisitos 5](#_Toc9181419)

[2.1 Requisitos Técnicos 5](#_Toc9181420)

[2.2 Requisitos Funcionais 5](#_Toc9181421)

[3. Atividade Desenvolvida 6](#_Toc9181422)

[3.1 Funcionalidades implementadas 7](#_Toc9181423)

[4. Bibliografia 8](#_Toc9181424)

[5. Anexos 9](#_Toc9181425)

[Anexo A - Calendarização 9](#_Toc9181426)

[Anexo B – Esquema ESP32 Adafruit Feather 10](#_Toc9181427)

[Anexo C – Imagem Painel 11](#_Toc9181428)

[ANEXO D – Esquema Eléctrico 12](#_Toc9181429)

# Lista de Figuras

[Figura 1 - Diagrama de Blocos 3](file:////Users/pcostatorres/Documents/GitHub/PFC/410_v1.docx#_Toc17151006)

[Figura 2 - Diagrama de Blocos Completo 4](file:////Users/pcostatorres/Documents/GitHub/PFC/410_v1.docx#_Toc17151007)

[Figura 3 - Matriz de LED's 5](#_Toc17151008)

[Figura 4 - Esquema Eléctrico Shift Register 7](#_Toc17151009)

[Figura 5 - Distribuição de saídas Shift Register com correspondência de ativação 8](#_Toc17151010)

[Figura 6 - Adafruit Huzzah32 - ESP32 Feather 9](file:////Users/pcostatorres/Documents/GitHub/PFC/410_v1.docx#_Toc17151011)

[Figura 7 - Módulo ESP32-WROOM-32U 9](file:////Users/pcostatorres/Documents/GitHub/PFC/410_v1.docx#_Toc17151012)

[Figura 8 - Diagrama Lógico da aplicação 12](file:////Users/pcostatorres/Documents/GitHub/PFC/410_v1.docx#_Toc17151013)

[Figura 9 - Esquema Eléctrico ESP32 Adafruit Feather 18](#_Toc17151014)

[Figura 10 - Pinout ESP32 Adafruit 18](#_Toc17151015)

[Figura 11 - Esquema Eléctrico 19](file:////Users/pcostatorres/Documents/GitHub/PFC/410_v1.docx#_Toc17151016)

[Figura 12 - Esquema Eléctrico 19](file:////Users/pcostatorres/Documents/GitHub/PFC/410_v1.docx#_Toc17151017)

# Lista de Tabelas

# Acrónimos

LED - Light Emissor Diode

LAN - Local Area Network

WLAN - Wireless Local Area Network

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

# Introdução

Hoje em dia vulgarizou-se a utilização de painéis de divulgação de informação não estáticos, habitualmente estes baseiam-se em painéis de LED[[1]](#footnote-1), existem soluções no mercado, no entanto estas habitualmente têm interfaces de controlo proprietárias. A interoperabilidade entre estas dificulta a sua gestão e manutenção.

Se for considerada a criação de uma LED WALL[[2]](#footnote-2) 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; e custos reduzidos.

Com a implementação de um sistema de controlo independente é possível integrar os vários tipos de painéis de LED com esta interface. Quando deparados com uma LED WALL de grandes dimensões poderá haver constrangimentos quanto ao seu sincronismo e à comunicação com os vários painéis de LED, quando se recorre a LAN ou WLAN.

## Enquadramento

Hoje em dia com a grande presença de painéis de LED em várias aplicações encontramos com frequência a necessidade de flexibilizar a sua utilização.

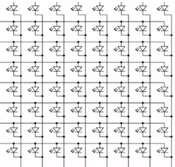
Assim surge a necessidade de criar uma interface que seja comum entre diferentes painéis, que seja dimensionável para tantos painéis quantos sejam necessários formando um agregado de painéis e que seja escalável.

Neste projeto propõe-se o desenvolvimento de uma interface para os painéis de LED, através de um microcontrolador, que faz a ligação entre o operador e os painéis de LED. O desenvolvimento desta interface será feito de forma modular e genérica, permitindo desta forma construir um sistema escalável.

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 LAN ou WLAN.

## Descrição do projeto

Tal como descrito no diagrama de blocos representado na Figura 1, este projeto será dividido em 3 blocos principais. A aplicação desenvolvida a partir do IDE para enviar informação para o microcontrolador. O microcontrolador que irá realizar a integração com o Painel de LED’s, controlando a informação visível nos painéis de LED’s. Por último temos o painel de LED’s que será uma matriz de LED que recebe a informação proveniente da aplicação e replicando-a nos seus LED’s.



MICROCONTROLADOR

APLICAÇÃO / IDE



USB

USB

WLAN

WLAN



PAINEIS DE LED RGB

Figura - Diagrama de Blocos

## Objetivos

A existência de um LED WALL é comum nos dias de hoje devido aos seus custos reduzidos de produção fazem com que sejam uma das principais opções quando se trata de reprodução de textos ou imagens.

O objetivo deste projeto é o desenvolvimento de uma estrutura modular com módulos básicos de 16x32 conjuntos de LED RGB, controlados através de um microcontrolador. Atualmente grande parte dos painéis de LED disponíveis são do tipo surface mounted device (SMD) e estes têm aplicações em ecrãs interiores e exteriores, diferentes densidades de LED por . Um pixel SMD é composto por um conjunto de LED’s com a côr vermelha, verde e azul todos juntos no mesmo encapsulamento, sendo montados estes na placa de circuito impresso, permitindo deste modo criar 8 níveis de cor.

O sistema a desenvolver irá utilizar hardware e software para permitir a transmissão de dados em blocos de 16x32 pixel, usando microcontroladores. A comunicação entre os vários módulos é feita através de um microcontrolador que possui interfaces de comunicação USB e sem fios para a receção de dados de um PC. Na Figura 2 está representado o diagrama de blocos de todo o sistema.



SHIFT REGISTER







SHIFT REGISTER

Figura - Diagrama de Blocos Completo

## Abordagem

Este projeto tem como base o controlo de um painel de LED’s RGB através de um micrcocontrolador, onde são analisados e escolhidos, os componentes de hardware tais como a fonte de alimentação, o painel de LED’s RGB e o microcontrolador a utilizar no projeto, e relativamente ao software qual a linguagem mais adequada a utilizar na programação do microcontrolador, que neste caso foi selecionada a linguagem de programação em C devido ao seu comprovado uso nos mais variados sistemas embebidos.

A abordagem para o projeto está distribuída em 3 fases, onde na primeira fase é analisada a arquitetura do painel de LED’s e a sua integração com o microcontrolador, com a seleção de comunicação e bibliotecas indicadas.

Na fase seguinte com os dados recolhidos e definidos nas fases anteriores, é feita uma prova de conceito na qual se demonstra o funcionamento do sistema.

Na última fase são corrigidos problemas detetados no desenvolvimento do projeto, melhorada a estabilidade e robustez do programa desenvolvido. Nesta fase será possível realizar uma demonstração funcional, onde o sistema deve apresentar-se com o comportamento definido nos requisitos.

# Análise de Requisitos

Neste capítulo é apresentada uma visão geral teórica acerca dos vários componentes e as técnicas de controlo que compõem o projeto.

Por forma a identificar as funcionalidades do sistema, definição dos requisitos, analisa-se e descreve-se detalhadamente cada uma das funcionalidades, de modo a que as mesmas possam ser validadas e desse modo se poder optar pelos módulos mais adequados ao seu funcionamento.

## Funcionamento de uma Matriz de LED

Os LED’s são frequentemente 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 \*.



Figura - Matriz de 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.

Este tipo de montagem pode levantar um problema caso seja pretendido ativar um determinado LED, ou seja no caso de se ativar o LED da linha dois e coluna dois, iria ligar o LED conforme está na Figura \* .



Agora, caso se opte por ativar o LED da linha quatro e coluna três, iria ativar outros dois LED’s que não há interesse em activar, conforme demonstrado na Figura \*.



Esta situação ocorre, pois, está a ser imposta uma tensão nas linhas dois e quatro, e nas colunas dois e três. Para resolver estes casos foi desenvolvida a multiplexagem.

A multiplexagem 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.

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.

### Painel RGB

Para o desenvolvimento do projeto optou-se por um painel com a correspondência de ativação dos LED’s na matriz, não linear e com uma refresh rate de ¼.

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 tem 6 shift registers de 64 bits que têm a responsabilidade sobre cada côr da 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. O shift register tem um latch (LAT), active high, e um output enable (OE), active low assíncronos.

Uma imagem com mapa

Descrição gerada automaticamente

Figura - Esquema Eléctrico Shift Register

Ao colocar o LAT com o valor ‘0’ e o OE com o valor ‘1’ irá permitir 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.

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 verifica na figura 4.



Figura - Distribuição de saídas Shift Register com correspondência de ativação

## Microcontroladores

Um microcontrolador é um computador dentro de um circuito integrado. Com a capacidade de ser programado, permite ao projeto onde se utiliza uma grande adaptabilidade.

Na escolha de um microcontrolador há determinados aspetos a ter em consideração e que influenciam a sua escolha:

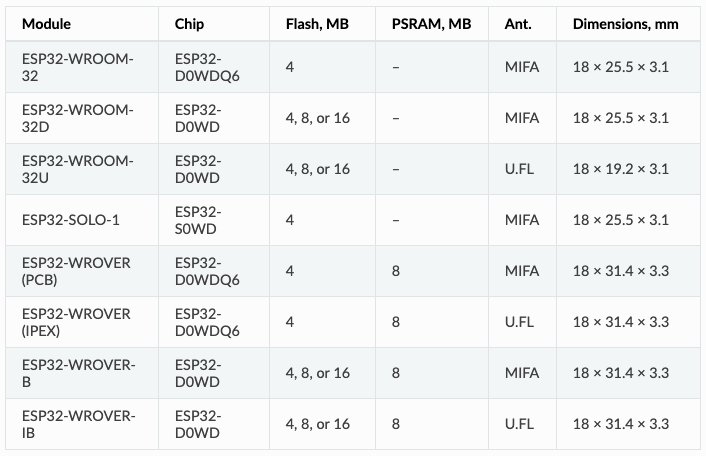
* Dimensão da memória RAM: é utilizada pelo programa principal para executar as suas rotinas. Quanto maior a memória RAM maior vai ser a sua capacidade de trabalho.
* Dimensão da memória FLASH: é utilizada para armazenar o programa e não necessita de fonte de alimentação para guardar o seu conteúdo.
* Número de Pinos de Entrada e Saída (GPIO): para se poder receber dados, processá-los e enviar para uma saída. O número de pinos pode ser importante quando se está a desenvolver um projeto.
* Conectividade: para se poder estabelecer comunicação com outros instrumentos é importante verificar os tipos de conexão que são possíveis, além da comunicação por cabo, também é importante para alguns projetos a conexão sem fios.

### Microcontrolador

O microcontrolador da Espressif ESP32 tem num chip integrado o microntrolador, 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 Figura \*.

Tabela - Diferentes Módulos ESP32



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

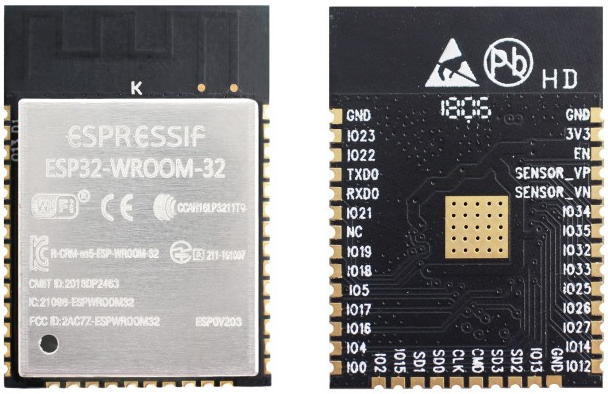
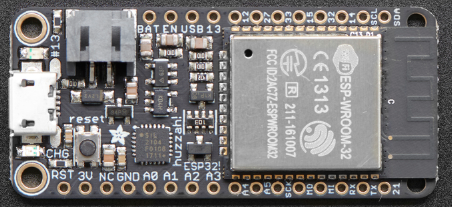


Figura - Adafruit Huzzah32 - ESP32 Feather

Figura - Módulo ESP32-WROOM-32

As características e as interfaces para periféricos deste módulo da Espressif são descritas:

* Processadores:
  + CPU: Xtensa dual-core (ou single-core) de 32 bits LX6 microprocessador, operando em 160 ou 240 MHz e realizar até 600 DMIPS
  + Ultrabaixa potência (ULP) co-processador
* Memória: 520 KB SRAM
* Conectividade sem fio:
  + Wi-Fi: [802.11](https://pt.wikipedia.org/wiki/IEEE_802.11) b/g/n
  + Bluetooth: v4.2 BR/EDR e BLE
* Interfaces para periféricos:
  + 12-bit SAR ADC até 18 canais
  + 2 × 8-bit [DACs](https://pt.wikipedia.org/wiki/Conversor_digital-anal%C3%B3gico" \o "Conversor digital-analógico)
  + 10 × Sensores de toque (GPIOs de [detecção capacitiva](https://pt.wikipedia.org/wiki/Sensores_capacitivos" \o "Sensores capacitivos))
  + Sensor de temperatura
  + 4 × [SPI](https://pt.wikipedia.org/wiki/Serial_Peripheral_Interface)
  + 2 × Interfaces I²S
  + 2 × Interfaces [I²C](https://pt.wikipedia.org/wiki/I%C2%B2C)
  + 3 × [UART](https://pt.wikipedia.org/wiki/USART)
  + Controladores de host [SD](https://pt.wikipedia.org/wiki/Secure_Digital_Card)/[SDIO](https://pt.wikipedia.org/wiki/Secure_Digital_Card)/CE-ATA/[MMC](https://pt.wikipedia.org/wiki/MultiMediaCard)/eMMC
  + Controlador escravo SDIO/SPI
  + Interface [Ethernet](https://pt.wikipedia.org/wiki/Ethernet) MAC da com DMA dedicado e Suporte [Protocolo IEEE 1588 com Tempo Preciso](https://pt.wikipedia.org/wiki/IEEE_1588)
  + [Barramento CAN](https://pt.wikipedia.org/wiki/Controller_Area_Network) 2.0
  + Controlador remoto infravermelho (TX/RX, até 8 canais)
  + Motor [PWM](https://pt.wikipedia.org/wiki/Modula%C3%A7%C3%A3o_por_largura_de_pulso)
  + LED [PWM](https://pt.wikipedia.org/wiki/Modula%C3%A7%C3%A3o_por_largura_de_pulso) (até 16 canais)
  + [Sensor de efeito Hall](https://pt.wikipedia.org/wiki/Sensor_de_efeito_Hall)
  + Pré-amplificador analógico de ultrabaixa potência

## Requisitos Técnicos

RT01: A alimentação dos componentes do painel de LED RGB deve ser garantido com uma fonte de alimentação externa adequada ao painel.

O consumo de corrente caso não venha explicitado no manual de utilização do painel poderá ser dado pela expressão em baixo:

RT02: A alimentação do microcontrolador deverá ser feita por USB 2.0 no mínimo.

RT03: Painel de RGB modular com montagem de LED SMD.

RT04: O microcontrolador deverá ter disponível os GPIO necessários \*\*\*, deverá ter conexão WIFI a um router e Bluetooth local.

# Estado da Arte

Este capítulo tem como objetivo a abordagem e apresentação de soluções comerciais existentes no mercado.

Painel xpto e o micro xpto são os adequados para projeto

Falar sobre o Wifi

2.1Tipos de painéis

2.2 Fontes de alimentação

2.3 Tipos de microcontrolador

2.4 Tipo de IDE (Setup e Loop)

# Implementação

Neste capítulo descreve-se a implementação no microcontrolador os algoritmos para o processamento de dados, o controlo do painel RGB, analisando assim as várias camadas que compõem a aplicação, estruturas de dados desenvolvidas e a interação entre as várias peças desenvolvidas, usando o estilo top-down.



Figura - Diagrama Lógico da aplicação e hardware

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 3.

* Camada Application – Onde é aplicada toda a lógica da aplicação e o processamento de eventos.
* Camada Middlewares – Nesta camada estão presentes as bibliotecas 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 Hardware – Esta camada é constituída pelo microcontrolador ESP32 e pelo painel de LED’s RGB.

A aplicação começa por inicializar o hardware dentro da função *Setup* como demonstrado no fluxograma presente na figura \*, no qual se efetua uma leitura da eeprom com o propósito de se obter o ssid e a password da última ligação 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 procede à inicialização de uma tarefa (task), que é responsável pelo refrescamento continuo da matriz do painel.



Figura - Fluxograma Função Setup

Após o Setup entra em execução a função Loop a qual como é referido anteriormente é chamada continuamente, onde são executadas as ações do fluxograma presente na figura \* a cada iteração.

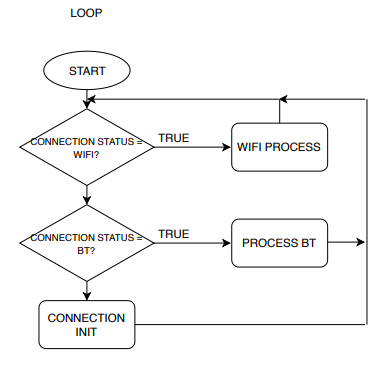


Figura – Função Loop

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 \*.



Figura – Fluxograma da Função Connection Init(Alterar e acrescenta)

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

Caso sejam realizadas as vinte iterações e não tiver sido estabelecida uma ligação por *Wifi* é chamada a função *Connection Using BT* cujo retorno afeta a variável de estado com o valor correspondente ao tipo de ligação estabelecida.

A função *Connetion Using BT* é responsável pela ativação do *Bluetooth* e o seu emparelhamento com um dispositivo que irá permitir uma ligação série e onde se configura o tipo de ligação, *Bluetooth* ou *Wifi*.

Como ilustrado na figura \*, começa-se por inicializar o Bluetooth e aguarda uma ligação. Quando o emparelhamento é estabelecido, é criada uma ligação série que permite a comunicação bidirecional entre o dispositivo e o microcontrolador.

A aplicação aguarda que seja inserida uma linha de texto no dispositivo com o comando a executar. Ao inserir um comando, verifica se o comando inserido é “BLUE” e neste caso, mantem a ligação Bluetooth ativa e retorna o valor a indicar que está estabelecida uma ligação Bluetooth, caso seja inserido outro comando, verifica se os primeiros quatro caracteres são “SSID” e no caso de se verificar esta condição os caracteres inseridos após o quarto caracter são copiados para a variável global denominada “ssid”, após este passo pode ser inserida a password, escrevendo “PASS” seguido da *password* onde os caracteres inseridos após o quarto caracter são copiados para a variável global denominada “password”.

Validada a inserção do “SSID” e “PASS” é retornado o valor a indicar que se irá realizar a tentativa de ligação por *Wifi.*



Figura - Fluxograma Função Connect Using BT

Após o retorno da função Connection Init, é executado o processamento correspondente com o valor da variável de estado.

No caso de a variável de estado ter o valor Wifi, é efetuado o processamento da função Wifi Process, cujo fluxograma se apresenta na figura\*.

A função Wifi Process tem como responsabilidade a gestão dos pedidos realizados pelo utilizador através do protocolo http.



Figura - Fluxograma Wifi Process

Ao entrar na função é verificada se a ligação Wifi está ativa, no caso da ligação se ter perdido é retornado o valor a indicar que não existe ligação. Se a ligação estiver ativa é criado um cliente, após a criação do mesmo é realizada uma tentativa de ligação a um determinado endereço de IP e a um porto definido. Com a ligação estabelecida é feita uma verificação se existem carateres disponíveis para serem recebidos, caso hajam carateres disponíveis é feita uma leitura dos mesmos, onde se realiza um parse sobre a string lida, para ser convertida de modo a poderem ser processados os dados para a matriz. Após a leitura de todos os bytes disponíveis ou se não existirem bytes disponíveis para receber entra em execução a função Scroll Display, que tem como função tratar o deslocamento horizontal, quando existem mais caracteres que colunas disponíveis e vertical, quando existem menos linhas disponíveis que as que são pretendidas. Quando a ligação com o cliente deixar de estar disponível, o cliente é desligado e a função retorna o valor a indicar o valor Wifi.

No caso de a variável ter o valor BT é efetuado o processamento da função Process BT cujo fluxograma é apresentado na figura \*.



Figura - Fluxograma da Função Process BT

A função Process BT tem como responsabilidade gerir a informação enviada pelo utilizador para ser transmitida no painel através da ligação por Bluetooth.

Após entrar na função é verificado se existem carateres disponíveis no buffer de receção, caso existam é realizada a leitura dos mesmos e é feita uma verificação se a String lida é igual à String “WIFI”. Se a verificação for falsa a função retorna o valor de NCON a indicar que a ligação por Bluetooth será interrompida, caso contrário a String lida é processada de modo a poder ser transmitida no painel. Após o processamento de todos os dados entra em execução a função Scroll Display.



Figura - Fluxograma Função Scroll Display

# Resultados Experimentais

# Conclusões

Ponto de melhoria

# Bibliografia

1. Programming ESP32 Board with Arduino IDE: <https://circuitdigest.com/microcontroller-projects/programming-esp32-with-arduino-ide>
2. Texas Instruments: 16-CHANNEL CONSTANT-CURRENT LED SINK DRIVERS
3. Sanders Full Color LED Module: P10 RGB Module
4. ESP32: <https://pt.wikipedia.org/wiki/ESP32>

# Anexos

## Anexo A – Esquema ESP32 Adafruit Feather

Uma imagem com texto

Descrição gerada automaticamente

Figura - Esquema Eléctrico ESP32 Adafruit Feather

Uma imagem com captura de ecrã, mapa

Descrição gerada automaticamente

Figura - Pinout ESP32 Adafruit

## Anexo B – Esquema Eléctrico

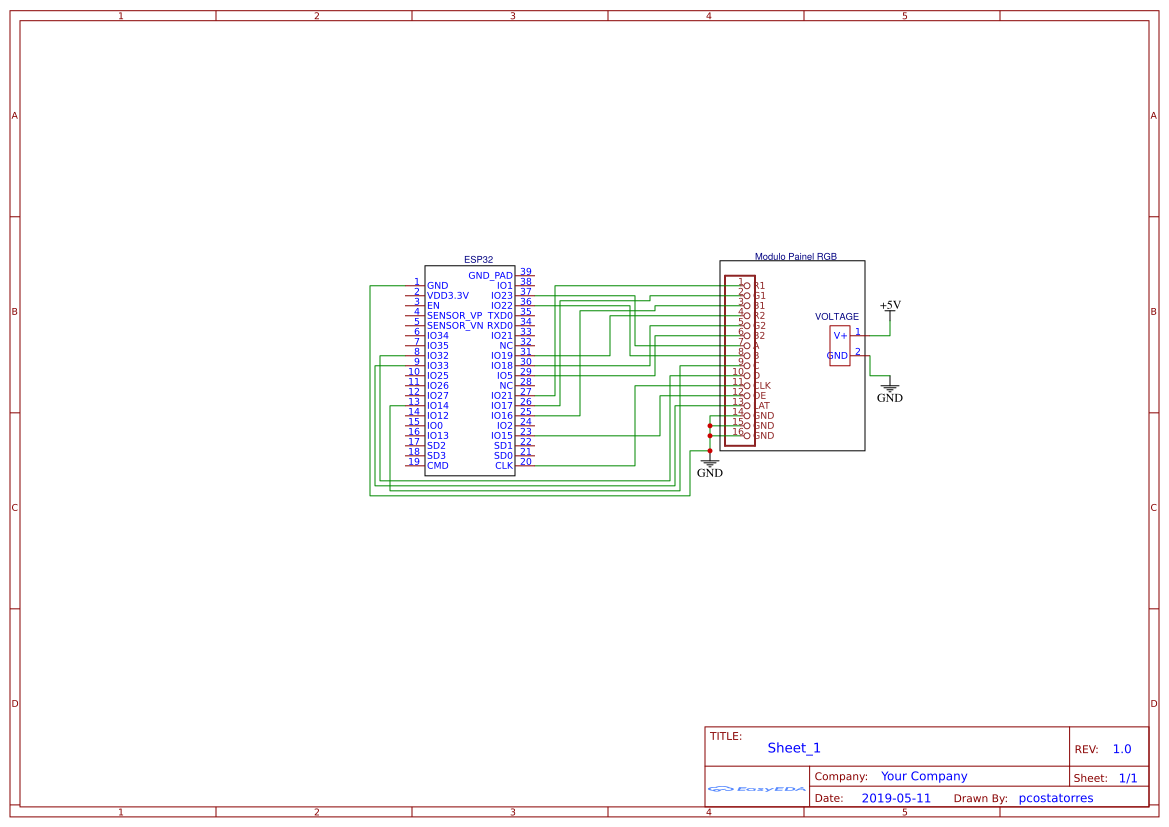


Figura - Esquema Eléctrico

Figura - Esquema Eléctrico

1. LED – Light Emissor Diode [↑](#footnote-ref-1)
2. LED WALL – Ecrã gigante de LED [↑](#footnote-ref-2)