

---

MINISTÉRIO DA EDUCAÇÃO – MEC

SECRETARIA DE EDUCAÇÃO SUPERIOR – SESU

PROGRAMA DE EDUCAÇÃO TUTORIAL – PET

UNIVERSIDADE FEDERAL FLUMINENSE – UFF

ESCOLA DE ENGENHARIA – TCE

GRUPO PET DO CURSO DE ENG. DE TELECOMUNICAÇÕES – PET-TELE

## Tutorial PET-Tele

Grupo de estudos e experimentação  
sobre os dispositivos ESP e suas aplicações  
(Versão: A2024M01D18)

Autores: Pedro Henrique Barbosa da Silva e Kriz

Tutor: Alexandre Santos de la Vega

Niterói – RJ

Janeiro / 2024

---

# Sumário

<b>1</b>	<b>Introdução</b>	<b>2</b>
1.1	PET e grupo PET-Tele . . . . .	2
1.2	Motivações . . . . .	2
1.3	Objetivo . . . . .	3
1.4	Metodologia . . . . .	3
1.5	Resultados esperados . . . . .	3
1.6	Organização do documento . . . . .	3
<b>2</b>	<b>Conceitos básicos</b>	<b>4</b>
2.1	Dispositivo ESP32 . . . . .	4
2.2	Placas de prototipagem . . . . .	4
2.3	<i>Internet of Things</i> (IoT) . . . . .	4
<b>3</b>	<b>ESP32</b>	<b>5</b>
3.1	Tipos de ESP32 . . . . .	5
3.2	Arquitetura e funcionamento do ESP32 . . . . .	6
3.2.1	Recursos periféricos . . . . .	7
3.2.2	Pinos GPIO . . . . .	7
3.2.3	Pinos de entrada analógica . . . . .	7
3.3	GPIOs de toque capacitivo . . . . .	7
3.4	Conversor Digital para Analógico (DAC) . . . . .	7
3.5	RTC – Relógio de Tempo Real . . . . .	8
3.6	PWM – Modulação por Largura de Pulso . . . . .	8
3.7	Sensor de efeito Hall . . . . .	8
3.8	Portas seriais . . . . .	8
3.9	Comunicação I2C . . . . .	8
3.10	Comunicação SPI . . . . .	9
3.11	Interrupções . . . . .	9
3.12	Habilitação (EN) . . . . .	9
3.13	Consumo de corrente . . . . .	9
<b>4</b>	<b>Diferença entre o ESP32 e outras placas similares</b>	<b>10</b>
<b>5</b>	<b>Exemplos</b>	<b>11</b>
5.1	Acionamento de um LED e comunicação serial . . . . .	11
5.2	Exemplo de uma conexão bluetooth serial . . . . .	12
	<b>Referências bibliográficas</b>	<b>14</b>

# Capítulo 1

## Introdução

Este capítulo trata da introdução do presente documento. Inicialmente, o PET e o grupo PET-Tele são brevemente descritos. Em seguida, são apresentados as motivações, o objetivo e os resultados esperados, deste trabalho. Por fim, a organização do documento é definida.

### 1.1 PET e grupo PET-Tele

O Programa de Educação Tutorial (PET) [Min] exige que os bolsistas dos seus grupos, ao serem submetidos a uma formação complementar, desenvolvam atividades que possuam, conjuntamente, itens relativos às áreas de Pesquisa, Ensino e Extensão, que consigam algum tipo de penetração no curso ao qual pertencem e que realizem trabalhos de cooperação com outros grupos, ligados ou não ao seu curso de origem. Logo, o PET busca atitudes inovadoras em Educação. Procurando atender aos requisitos do Programa, o PET-Tele [Gru], grupo PET do Curso de Engenharia de Telecomunicações da Universidade Federal Fluminense, realiza atividades em diversas linhas do conhecimento, de acordo com o interesse e as competências de seus integrantes.

### 1.2 Motivações

O kit de desenvolvimento Arduino é uma placa de circuito eletro-eletrônico, cujo elemento principal é um dispositivo eletrônico denominado de microcontrolador. Ela pode ser usada como um simples dispositivo de prototipação rápida ou como um elemento dentro de um sistema mais complexo. Desde 2009, o grupo realiza pesquisas, estudos, palestras e oficinas/minicursos/cursos, sobre o kit Arduino. Tais atividades culminaram, por iniciativa do grupo, na criação da disciplina optativa hoje identificada como TET-00.319 – Introdução ao kit de desenvolvimento Arduino I, com aulas de caráter prático e executadas em laboratório. Nessa disciplina, o grupo desenvolve uma iniciação à docência e uma monitoria informal (ver Atividade CC3 do Planejamento de Atividades 2023 do grupo). Em adição, o grupo já realizou projetos, em colaboração com outros setores na IES, culminando até na produção de TCCs. Na mesma linha de atuação, porém ocupando um outro nicho, placas de desenvolvimento usando os dispositivos ESP8266 e ESP32 têm ganhado a atenção dos projetistas, pela maior complexidade de tais dispositivos em relação aos microcontroladores comuns. O ESP é considerado um dispositivo do tipo SoC (System on a Chip ou Sistema em um chip), contendo antena integrada, amplificador de potência, receptor de baixo ruído amplificado, filtros, gerenciamento de energia, comunicação por Bluetooth e Wi-Fi. No passado, o grupo já teve uma rápida experiência com o ESP8266. Nesse momento, o grupo visualizou uma oportunidade de aprendizado, similar ao

que já foi realizado com o Arduino e brevemente com o ESP8266, porém buscando expandir suas fronteiras de conhecimentos e aplicações, agora com o ESP32.

## 1.3 Objetivo

O objetivo desta atividade é desenvolver um trabalho com as placas de desenvolvimento que usam os dispositivos ESP e documentar o trabalho realizado, de modo a facilitar o entendimento sobre os computadores de placa única em geral e entender suas funcionalidades e suas aplicações em projetos.

## 1.4 Metodologia

Deverão ser cumpridos os seguintes itens:

- Realizar pesquisa sobre os dispositivos ESP.
- Organizar e estudar os dados obtidos.
- Realizar experimentos com os dispositivos que o grupo possui.
- Produzir um material autoral textual atualizado, resumindo a pesquisa, o estudo e os experimentos realizados.
- Disponibilizar o material autoral no *website* do grupo.

## 1.5 Resultados esperados

Espera-se que, de posse deste documento, o leitor consiga identificar as diferenças entre as placas de desenvolvimento mais comuns, entender o funcionamento dos dispositivos ESP e saber utilizá-los nos seus projetos.

## 1.6 Organização do documento

Além deste primeiro capítulo, que trata da introdução do presente documento, o restante do texto possui a seguinte organização. No Capítulo 2, são brevemente apresentados alguns conceitos básicos. O dispositivo ESP32 é abordado no Capítulo 3. No Capítulo 4, são comparadas algumas das placas de desenvolvimento mais comuns. Alguns exemplos de experimentos são apresentados no Capítulo 5.

# Capítulo 2

## Conceitos básicos

### 2.1 Dispositivo ESP32

Criado pela empresa Espressif, o ESP32 é uma evolução do ESP8266, criado em 2016. Ele tem o intuito de ser uma placa de desenvolvimento de baixo custo e com baixo consumo de energia, voltado para IoT, sendo popularizado por ter acesso direto a Wi-Fi e Bluetooth [Cur].

### 2.2 Placas de prototipagem

Placas de prototipagem são *hardwares* previamente montados, com intuito de facilitar a realização de projetos. Essas placas representam um sistema padronizado, que torna mais fácil o trabalho de desenvolvimento, tratando-se de uma ferramenta que auxilia a solução de problemas que surgem no cotidiano das pessoas que estão desenvolvendo projetos [Vic].

Exemplos de placas de prototipagem são os seguintes:

- Arduino.
- ESP8266.
- ESP32.

### 2.3 *Internet of Things* (IoT)

A Internet das Coisas (ou *Internet of Things* ou IoT) [Ora] descreve a rede de objetos físicos incorporados a sensores, *software* e outras tecnologias, com o objetivo de conectar dispositivos e trocar dados entre dispositivos, via Internet. Esses dispositivos variam de objetos domésticos comuns a ferramentas industriais sofisticadas.

Seguindo tal ideia, as placas de prototipagem participam desse contexto, sendo utilizadas como interfaces, que controlam e captam os dados dos sensores, podendo enviá-los utilizando algum dispositivo de comunicação, tais como: LoRa, Wi-Fi e o Bluetooth.

Assim o ESP32 se destaca, sabendo-se que, em suas versões mais básicas, ele já possui uma placa de rede e um dispositivo Bluetooth integrado.

# Capítulo 3

## ESP32

O ESP32 [Usi] é um microcontrolador de baixo custo e baixo consumo de energia desenvolvido pela empresa chinesa Espressif Systems. É considerado um dos microcontroladores mais poderosos e versáteis disponíveis no mercado, especialmente em termos de conectividade, pois possui suporte a Wi-Fi, Bluetooth, Bluetooth LE e outras tecnologias de comunicação sem fio. O ESP32 é baseado em um processador *dual-core Tensilica LX6* com *clock* de até 240 MHz e tem suporte a 520 KB de RAM e 4 MB de memória *flash* interna. Ele também possui uma grande variedade de periféricos, incluindo interfaces para câmera, UART, SPI, I2C, I2S, ADC, DAC, PWM e muito mais. Assim como o Arduino, o ESP32 também é programado usando uma linguagem de programação de alto nível, como C ou C++, e pode ser programado usando uma variedade de ferramentas de desenvolvimento, incluindo a Arduino IDE, a plataforma *Espressif IoT Development Framework* (ESP-IDF) e outras ferramentas de terceiros. O ESP32 tem sido usado em muitos projetos de IoT, robótica, automação residencial e outros projetos eletrônicos.

### 3.1 Tipos de ESP32

Os tipos de ESP32 [Esp] são variados, apresentando modelos simples com o chip unicamente, modelos em módulos com conexão USB e pinagem extra integrada, modelos com *display oled* para visualização de informações, modelos com comunicação LoRa, além de modelos com câmera e suportes para alimentação com bateria já inclusos. O que todos eles tem em comum é o LX6 da *Tensilica*, de 2 *Core*. Na Figura 3.1, é mostrado o diagrama de um ESP32 referente ao modelo ESP32-WROOM-32.

Alguns tipos de ESP32 mais comuns são os seguintes:

- ESP32-WROOM-32: Este é o módulo original da Espressif, que é um dos mais populares e amplamente utilizados módulos ESP32. Possui 4 MB de memória *flash* e 4 MB de memória RAM, além de recursos Wi-Fi e Bluetooth.
- ESP32-PICO: É um módulo de menor tamanho, com um conjunto completo de recursos, como o ESP32-WROOM-32, porém com um encapsulamento menor.
- ESP32-SOLO-1: É um módulo compacto com apenas um núcleo e 4 MB de memória *flash*.
- ESP32-S2: Este é um módulo com menor consumo de energia e menor custo, com suporte para Wi-Fi e Bluetooth. Ele vem com 128 KB de RAM e 4 MB de memória *flash*.
- ESP32-CAM: É um módulo com uma câmera embutida, permitindo a criação de projetos de vigilância e monitoramento com facilidade.

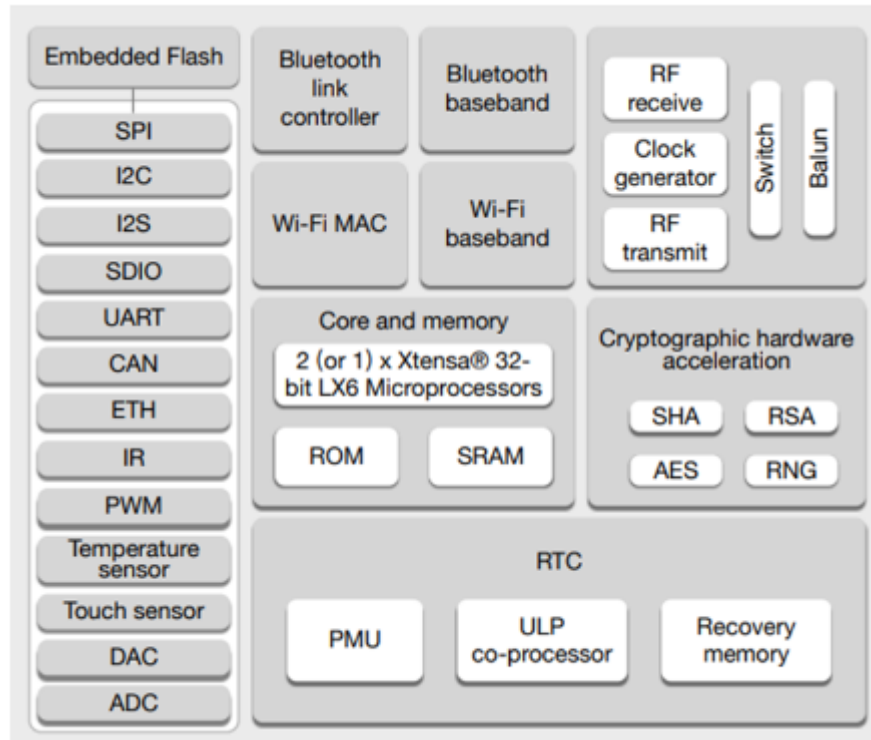


Figura 3.1: Diagrama de blocos do ESP32.

## 3.2 Arquitetura e funcionamento do ESP32

- Microprocessador *dual-core Tensilica LX6 240 MHz* com um desempenho de 600 DMIPS.
- Possui 520 KBy SRAM, há 16 MB de memória *flash* na placa.
- O *chip* é alimentado por uma tensão de 2,2 a 3,6 V.
- Temperatura ambiente possível de -40 ° C a 125 ° C.
- Em sono profundo, o ESP32 consome 2,5  $\mu A$ , de acordo com o fabricante.
- Baixo consumo de energia é fornecido por um processador ULP (*Ultra-Low-Power*).
- Até 8 Kbytes de dados e programas podem ser armazenados na SRAM do RTC (relógio de tempo real) para que o acesso a temporizadores, interrupções e periféricos seja possível, mesmo no modo de hibernação profunda.
- Para se comunicar com o mundo externo, o SoC (Sistema em um *Chip*) inclui o componente HT40 802.11b / g / n WiFi e a funcionalidade Bluetooth.
- Além da antena PCB integrada, um componente externo pode ser conectado por meio de um conector IPEX
- Como sensores, o ESP32 possui um sensor Hall, um *touch button* capacitivo, um amplificador analógico de baixo sinal e um quartzo de cristal de 32 kHz

### 3.2.1 Recursos periféricos

- 18 canais conversor analógico (ADC)
- 10 GPIOs (*General Purpose Input/Output*) de detecção capacitiva
- 3 interfaces UART (Receptor/Transmissor universal assíncrono)
- 3 interfaces SPI (Serial protocolo de interface periférica)
- 2 interfaces I2C (Circuito inter-integrado para dados)
- 16 canais de saída PWM
- 2 conversores digital para analógico
- 2 interfaces I2S (Circuito inter-integrado para áudio)

### 3.2.2 Pinos GPIO

Possui 25 GPIOs, alguns desses pinos podem ser usados apenas para entrada, são eles: GPIOs 34, 35, 36 e 39.

Ainda assim, nem todos os pinos têm *pull up* de entrada, é necessário de um *pull up* externo nesses casos. Dessa maneira, os pinos com *pull up* interno são os GPIOs: 14, 16, 17, 18, 19, 21, 22 e 23. Enquanto os pinos sem *pull up* interno são os GPIOs: 13, 25, 26, 27, 32 e 33.

### 3.2.3 Pinos de entrada analógica

ESP32 possui 18 canais analógicos de 12 *bits*, que são pinos que podem converter esse sinal para digital.

Portanto, uma tensão de 0 *volts* produzirá um valor digital de 0, logo, a tensão máxima produzirá um valor digital de 4095. De maneira proporcional, as faixas de tensão entre eles produzirão um valor digital correspondente.

## 3.3 GPIOs de toque capacitivo

ESP32 possui 10 sensores de toque capacitivos, cuja função é detectar variações em qualquer coisa que contenha uma carga elétrica, como a pele humana.

Assim, eles funcionam como o *touch* da tela de seu celular, se formos comparar. Dessa maneira, tais pinos são facilmente integrados em blocos capacitivos e substituem os botões mecânicos. O pino de toque capacitivo também podem ser usados para despertar o ESP32 do estado *deep sleep*.

## 3.4 Conversor Digital para Analógico (DAC)

Para fazer a conversão digital-analógico, existem dois canais de 8 *bits* que convertem sinais digitais em saídas de sinal analógico: DAC1 (GPIO25) e DAC2 (GPIO26).



## 3.5 RTC – Relógio de Tempo Real

RTC nada mais é do que *real time clock* (relógio em tempo real). Sua função é manter as datas e horários corretos mesmo com quedas de energia. Entretanto, ele precisa de uma bateria para casos de queda de energia.

A ESP32 mantém, ainda, a data atualizada em *deep sleep*. Assim, pode-se vê em qualquer momento que ela estará correta.

## 3.6 PWM – Modulação por Largura de Pulso

Assim como o Arduino, a ESP32 também possui pinos PWM. A placa contém 16 canais independentes para configurar e gerar sinais com propriedades diferentes.

Dessa forma, todos os pinos que podem atuar como saídas podem ser usados como pinos PWM (apenas os pinos GPIOs 34 a 39 de entrada não podem gerar PWM).

Para definir um sinal PWM, é necessário colocar esses parâmetros no código: frequência do sinal, ciclo de trabalho, Canal PWM, GPIO onde você deseja enviar o sinal.

## 3.7 Sensor de efeito Hall

A placa também possui um transdutor, que é um sensor que quando está sob influência de um campo magnético sofre variação da tensão de saída. O efeito *hall* acontece quando a corrente elétrica desvia sua trajetória por conta do campo magnético. Assim, esse sensor fica inserido no interior do chip do microcontrolador. Para gerar o campo magnético, pode-se colocar um ímã sobre ele.

## 3.8 Portas seriais

A ESP32 tem três portas seriais. A primeira (RX0, TX0) é usado para programação:

- GPIO3 (U0RXD)
- GPIO1 (U0TXD)

Outra porta serial está disponível em:

- GPIO16 (U2RXD)
- GPIO17 (U2TXD)

## 3.9 Comunicação I2C

Ao usar a ESP32 com o Arduino IDE, você deve usar os pinos padrões ESP32 I2C (compatíveis com a biblioteca *Wire* e mostrados na tabela 3.1):

- GPIO 22 (SCL)
- GPIO 21 (SDA)

SPI	MOSI	MISSÔ	CLK	CS
VSPI	GPIO 23	GPIO 19	GPIO 18	GPIO 5
HSPI	GPIO 13	GPIO 12	GPIO 14	GPIO 15

Tabela 3.1: Tabela SPI

### 3.10 Comunicação SPI

Por padrão, o mapeamento de pinos para SPI são mostrados na tabela 3.1:

### 3.11 Interrupções

Todos os GPIOs podem ser configurados como interrupções.

### 3.12 Habilitação (EN)

Ativar ou habilitar (EN) é o pino de ativação do regulador de 3,3 V. Como ele é um pino *pull up*, então conecte ao terra para desativar o regulador. Isso significa que, pode utilizar esse pino conectado a um botão para reiniciar o ESP32.

### 3.13 Consumo de corrente

A corrente máxima absoluta por GPIO é de 40 mA fornecidos e de 28 mA que podem ser consumidos, de acordo com a seção intitulada “Condições de operação recomendadas”, no *datasheet* da placa.

A Figura 3.2 mostra toda a relação dos pinos de uma placa ESP32.

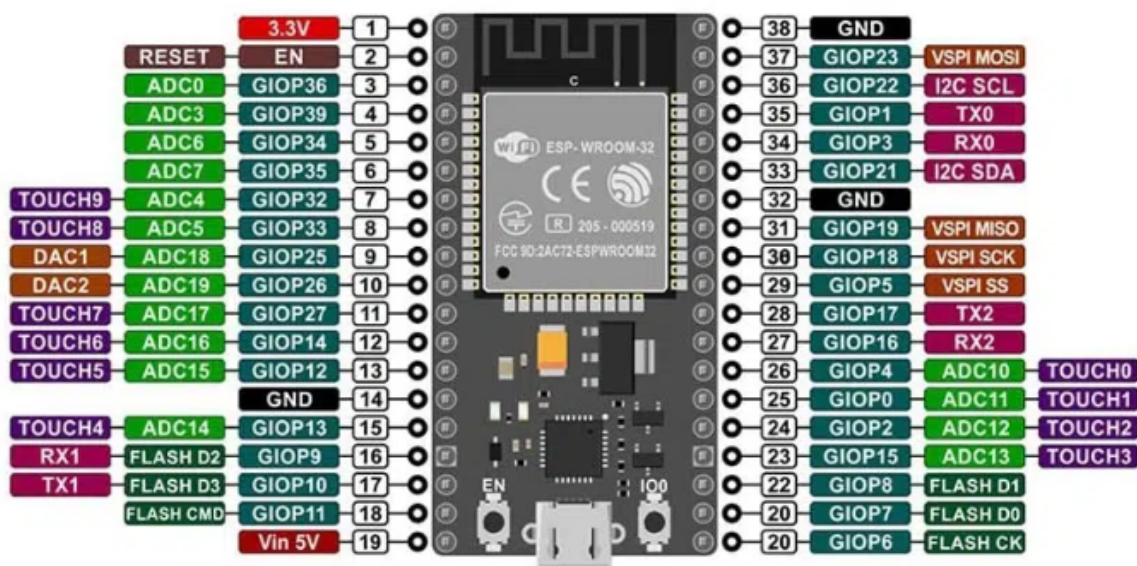


Figura 3.2: Pinagem do ESP32.

# Capítulo 4

## Diferença entre o ESP32 e outras placas similares

O Arduino [Mar] é formado pelo microcontrolador ATmega, que pode vir no formato SMD ou PID. O *chip* ATmega é o cérebro da placa, controlando as portas de entrada e saída. Além disso, é possível montar sua própria placa somente a partir desse *chip*, é o que chamamos de Arduino *standalone*, ou seja, você pode criar o seu Arduino de forma caseira com alguns componentes. Podemos dizer que, o Arduino é o ponto de partida para todos aqueles que querem começar a aprender sobre o assunto. Através dele, fica mais fácil entender como funciona a programação em C e C++, as lógicas e funções, além das conexões físicas, facilitadas pelo uso da *proto-board* e do Arduino.

Porém, com Wi-Fi e Bluetooth integrados, o ESP trouxe uma alternativa de mais baixo custo para projetos IoT, tendo em vista que para conectar o Arduino a *internet* se é necessário um *shield* que se acopla a placa.

Segue, na Tabela 4.1, uma comparação entre as principais placas de prototipagem.

	<b>ESP8266</b>	<b>ESP32</b>	<b>Arduino UNO</b>
Corrente	197 mA	220 mA	40 mA
Núcleo	1	2	1
Arquitetura	32 <i>bits</i>	32 <i>bits</i>	8 <i>bits</i>
<i>Clock</i>	80 - 160 MHz	160 - 240 MHz	16 MHz
Bluetooth	Não possui	Clássico e BLE (baixa energia)	Não possui
Wi-Fi	Sim	Sim	Não
RAM	160 KB	52 KB	2 KB
FLASH	16 Mb	16 Mb	32 KB
GPIO	13	34	14
DAC	0	2	0
ADC	1	18	6
Interfaces	SPI, I2C, UART, I2S	SPI, I2C, UART, I2S, CAM	SPI, I2C, UART

Tabela 4.1: Tabela de comparação entre ESP8266, ESP32 e Arduino UNO.

# Capítulo 5

## Exemplos

### 5.1 Acionamento de um LED e comunicação serial

```
void setup() {  
  Serial.begin(115200);  
  Serial.println("Hello, ESP32!");  
  pinMode(23, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(23, HIGH);  
  delay(1000);  
  digitalWrite(23, LOW);  
  delay((500));  
}
```

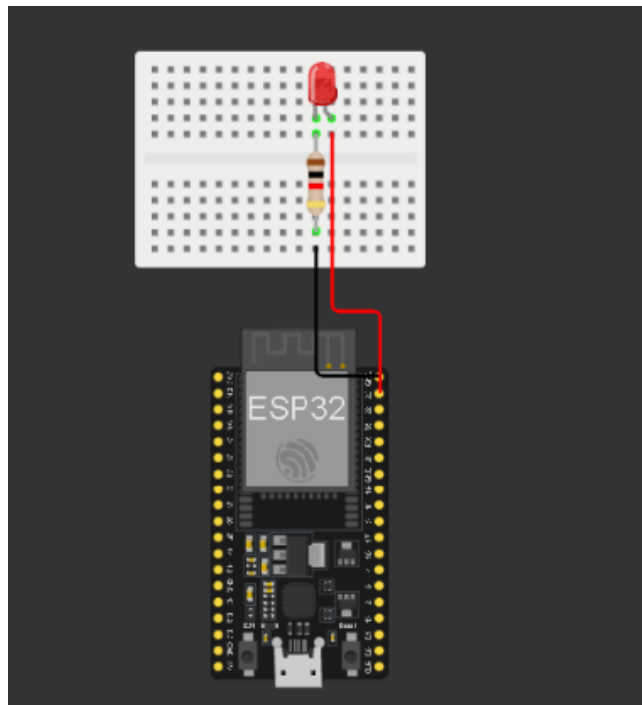


Figura 5.1: Circuito de acionamento de um LED.

```

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:2
load:0x3fff0030,len:1156
load:0x40078000,len:11456
ho 0 tail 12 room 4
load:0x40080400,len:2972
entry 0x400805dc
Hello, ESP32!

```

Figura 5.2: Imagem da tela da ferramenta Monitor Serial.

## 5.2 Exemplo de uma conexão bluetooth serial

O objetivo desse exemplo é mostrar o funcionamento da conexão bluetooth com o ESP32.

Para iniciar, no preâmbulo, inclui-se a biblioteca “BluetoothSerial.h”. Em seguida, a variável nome\_do\_dispositivo serve para inserir o nome que vai aparecer na hora de parear os dispositivos. Em seguida, os se encontram condicionais de pareamento as quais vão identificar se tiver algum erro na hora de ligar o bluetooth do ESP. No final, inicia-se o objeto que será utilizado “SerialBT”.

```

#include "BluetoothSerial.h"

String nome_do_dispositivo = "ESP32-BT";

#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled!
  Please run 'make menuconfig' to and enable it
#endif

#if !defined(CONFIG_BT_SPP_ENABLED)
#error Serial Bluetooth not available or not enabled.
  It is only available for the ESP32 chip.
#endif

BluetoothSerial SerialBT;

```

No *setup*, apenas inicia-se a comunicação serial e o nosso objeto.

```

void setup() {
  Serial.begin(115200);
  SerialBT.begin(device_name);
}

```

No *loop*, é onde será feita a verificação de recebimento e envio de mensagens via bluetooth e comunicação serial.

```
void loop() {
  if(Serial.available()){
    SerialBT.write(Serial.read());
  }
  if(SerialBT.available()){
    char recebido = SerialBT.read();
    Serial.write(recebido);
    Serial.print(recebido);
    if(recebido == 'c'){
      Serial.println('c');
    }
  }
  delay(20);
}
```

Para realizar os testes, foi utilizado o Arduino IDE, junto ao aplicativo Serial Bluetooth Terminal.

# Referências bibliográficas

- [Cur] Curto Circuito. *Curto Circuito*. URL: <https://curtocircuito.com.br/blog/Categoria%20IoT/conhecendo-esp32>. Acesso em: 18/01/2024.
- [Esp] Espressif. *Espressif*. URL: <https://www.espressif.com/en/home>. Acesso em: 18/01/2024.
- [Gru] Grupo PET-Tele. *PET-Tele / UFF*. URL: <http://www.telecom.uff.br/pet/>. Acesso em: 18/01/2024.
- [Mar] Maria Torela. *Lobo da Robótica*. URL: <https://lobodarobotica.com/blog/arduino-ou-esp-descubra-a-melhor-opcao//>. Acesso em: 18/01/2024.
- [Min] Ministério da Educação (MEC). *Programa de Educação Tutorial (PET)*. URL: [http://portal.mec.gov.br/index.php?option=com\\_content&view=article&id=12223&ativo=481&Itemid=480](http://portal.mec.gov.br/index.php?option=com_content&view=article&id=12223&ativo=481&Itemid=480). Acesso em: 18/01/2024.
- [Ora] Oracle team. *Oracle*. URL: <https://www.oracle.com/br/internet-of-things/what-is-iot/>. Acesso em: 18/01/2024.
- [Usi] Usinainfo. *Usinainfo*. URL: <https://www.usinainfo.com.br/esp32-611#:~:text=Os%20tipos%20de%20ESP32%20s%C3%A3o,alimenta%C3%A7%C3%A3o%20com%20bateria%20j%C3%A1%20inclusos>. Acesso em: 18/01/2024.
- [Vic] Victor Vision. *Victor Vision*. URL: <https://victorvision.com.br/blog/prototipagem-de-placas-de-circuito-impresso/>. Acesso em: 18/01/2024.