

**INSTITUTO  
FEDERAL**  
Paraíba

Sistema de Automação Residencial com  
interface de comunicação via Telegram.

**Disciplina:** Sistemas Embarcados

**Docente:** Alexandre Sales Vasconcelos

**Discentes:** Áquila Samuel Azevedo Dias

João Gabriel de Oliveira Ponciano

João Messias da Silva Medeiros

## **Resumo**

Nesse relatório está o desenvolvimento de um protótipo que foi planejado para automatizar residências por meio de um microcontrolador (Esp32), integrando funcionalidades que agregam o dia a dia do utilizador. Como por exemplo, acender ou apagar luz, por meio de sensor de presença, sensor de luminosidade e etc. Essas funcionalidades estarão integradas a uma comunicação com o utilizador via telegram.

**Palavras chaves:** Esp32, telegram, microcontrolador.

## **1. Introdução**

Automação residencial apresenta melhorias ao ambiente doméstico que aumenta a qualidade de vida do residente, provendo um ambiente flexível, confortável, saudável e seguro. Recentemente, sistemas de automação baseados na Internet tornaram-se mais populares em mercados internacionais (ELKAMCHOUCHI; ELSHAFEE, 2012).

Para evitar o uso excessivo de fios na instalação de sistemas de automação residencial, a maioria dos sistemas disponíveis usa diferentes padrões de comunicação sem fio para trocar dados e sinais entre seus componentes, como Bluetooth, Rádio Frequência e WiFi (AZID; KUMAR, 2011).

A utilização de todos esses padrões e tecnologias é facilitada com o uso de um microcontrolador (Esp32), que é uma plataforma de prototipagem eletrônica de hardware livre, cujo objetivo é criar ferramentas acessíveis, de baixo custo, flexíveis e fáceis de se usar. O projeto fornece uma forma acessível para que seja possível criar dispositivos que interagem com seu ambiente usando sensores e atuadores.

## **2. Objetivos**

O objetivo desse projeto é agregar conhecimento sobre o microcontrolador em uso, desenvolver habilidades de programação e integração do mesmo com outros dispositivos, componentes e periféricos já descritos acima. Além do conhecimento, estaremos desenvolvendo um protótipo de baixo custo que pode ser utilizado diariamente em qualquer residência.

### 3. Materiais principais

#### 1. ESP32-D0WDQ6



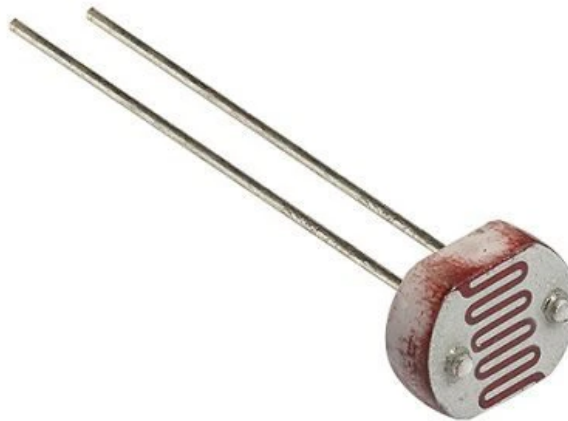
O ESP32 é um microcontrolador de baixa potência e baixo custos. Ele vem com Bluetooth e Wi-Fi já integrados. Destina-se especialmente a fornecer versatilidade, robustez e confiabilidade em um grande número de aplicações.

#### Características principais 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, uma touch button capacitivo, um amplificador analógico de baixo sinal e um quartzo de cristal de 32 kHz.

## **2.Sensor de Luminosidade**



LDR (do inglês Light Dependent Resistor ou em português Resistor Variável Conforme Incidência De Luz) é um tipo de resistor cuja resistência varia conforme a intensidade de radiação eletromagnética do espectro visível que incide sobre ele

### 3.Sensor de Presença



Os sensores de presença mais comuns (utilizados em portas e lâmpadas, por exemplo) funcionam por detecção de calor (radiação infravermelha), então, ao entrarmos em ambientes com sensor, o calor emitido pelo nosso corpo é interpretado como um sinal elétrico, acionando o que estiver conectado ao dispositivo

#### 3.1 Material secundário utilizado

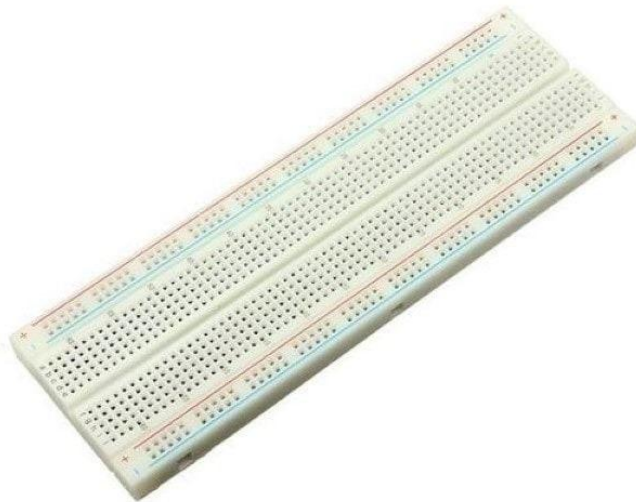
- Leds



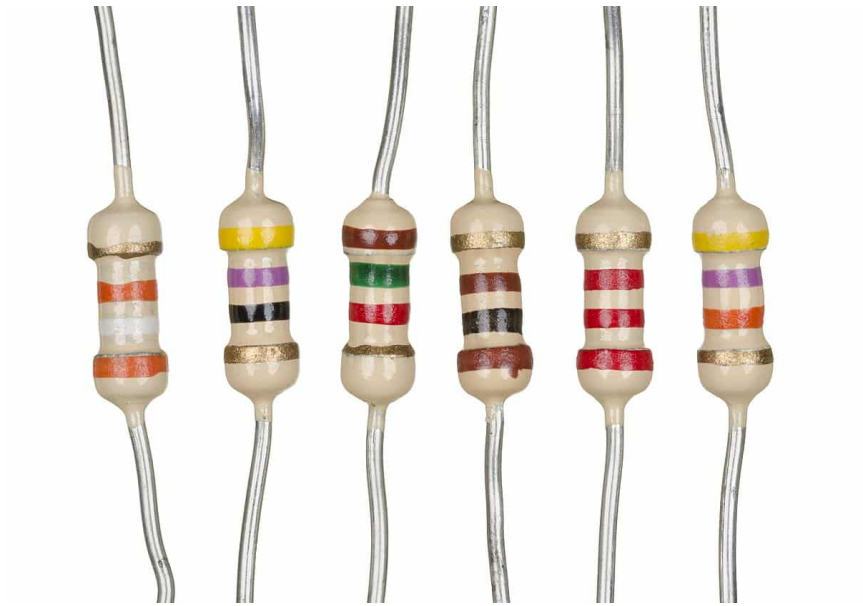
- **Cabos**



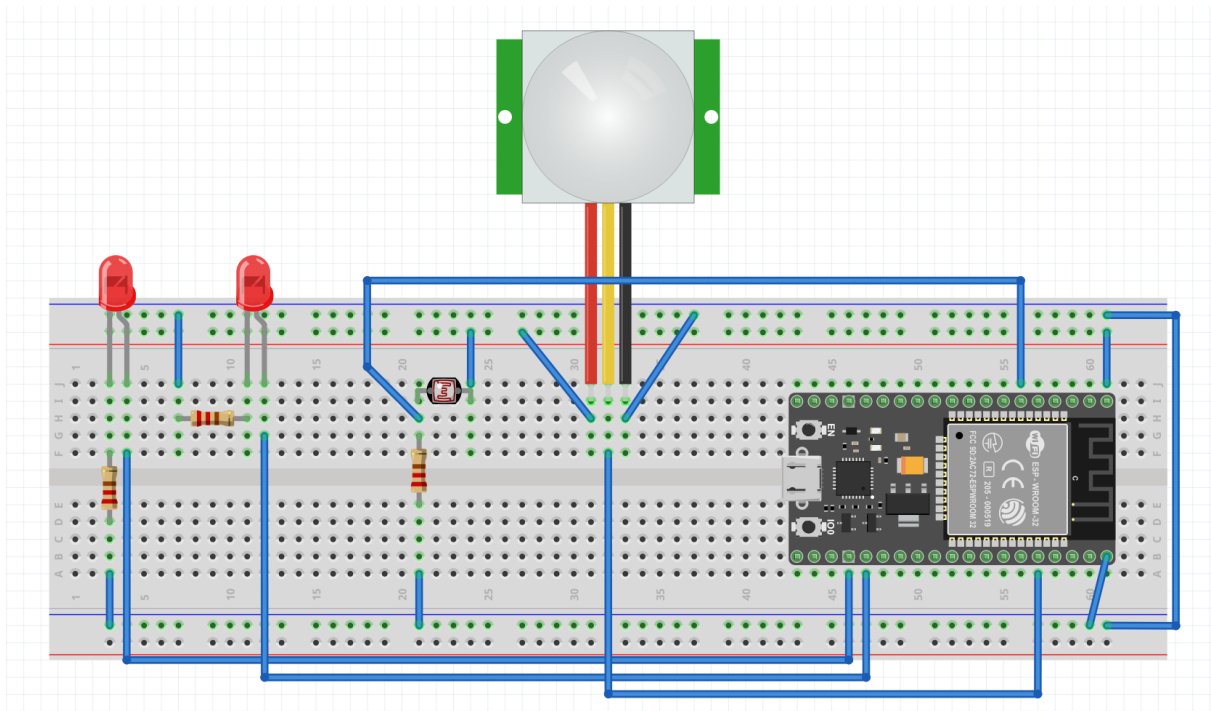
- **Protoboard**



- Resistores



- Diagrama do protótipo montado no protoboard



## **4. Resultados**

Com o final do nosso projeto conseguimos concluir com sua entrega parcial. Conseguimos realizar todas as funções apresentadas no projeto, como utilizar os sensores de luminosidade e de presença para avisar quando a luz estava apagada ou ligada, também ligar e desligar as luzes para uma casa automatizada.

Junto com essa atividade conseguimos implementar algo que foi novidade para todo o grupo que foi a ligação do ESP com o Telegram, com isso nós saímos do ponto de vista acadêmico alunos bem mais preparados com a parte de sistemas embarcados.

Entretanto, tivemos algumas dificuldades na questão da logística, pois iríamos utilizar o sensor de luminosidade (TSL2561), mas ele não veio soldado e como não tínhamos a máquina de solda, resolvemos trocar e usar o sensor descrito no relatório acima. Outra dificuldade que encontramos foi em programar o projeto em esp-idf, devido a falta de material existente na internet em relação a todos os sensores, atuadores e funções que iremos usar na nossa aplicação, devido a este fato decidimos fazer a programação do nosso projeto em arquivos .ino.

## **5. Conclusão**

Com todo o processo do nosso projeto, nosso grupo aprimorou bastante o conhecimento em relação a sistemas embarcados, principalmente na parte da ligação de sistemas embarcados com o Telegram, apesar de não ter sido possível entregar o trabalho em esp-idf, também foi importante para nós pois foi algo que nenhum dos integrantes tinha conhecimento e adquirimos mesmo que só o básico, conhecimento de algo novo.

Como sugestão, pedimos que sejam feitas mais aulas práticas de programação em sistema embarcados, que acredito que seja o principal ponto de dificuldades da maioria dos alunos, apesar de que tivemos aulas teóricas, acreditamos que somente a teoria não é suficiente para atingirmos um conhecimento necessário para implementar projetos como esse em ESP-IDF.



## 6. Referências

<https://www.espressif.com/en/products/socs/esp32>

<https://deinfo.uepg.br/~alunoso/2019/SO/ESP32/HARDWARE/>

<https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-guides/tools/idf-tools.html>

<https://blog.smartkits.com.br/>

<https://docs.espressif.com/projects/esp-idf/en/stable/esp32/get-started/vscode-setup.html>