

Projeto automação residencial

Ruan Flaneto Cartier

August 2, 2021

Contents

1	Motivação	1
2	Objetivos	1
3	Detalhamento do projeto	2
3.1	Lâmpadas	2
3.1.1	Descrição do circuito	2
3.1.2	Componentes utilizados (por lâmpada)	3
3.2	Descrição do software	3
3.2.1	Conectividade e gerenciamento de ações	3
3.2.2	GUI	4
3.2.3	Geração de relatórios	4

1 Motivação

Um projeto de automação residencial foi demandado. Primeira coisa que vem em mente é poder controlar as lâmdas de casa individualmente como meio de gerenciar o uso de cargas residenciais, viabilizando a economia de energia elétrica. Assim, pretende-se usar um módulo de ESP01 com relé (vide figura 2) para cada ponto de interruptor de lâmpada para poder ter conexão com o computador central (raspberry pi).

2 Objetivos

Gerenciar o funcionamento das lâmpadas de casa, cujo funcionamento deve ser por comando de voz ou de forma manual. Este gerenciamento também

inclui a formação de relatórios sobre consumo elétrico (estimado) em cada dispositivo, apresentando as informações em histogramas.

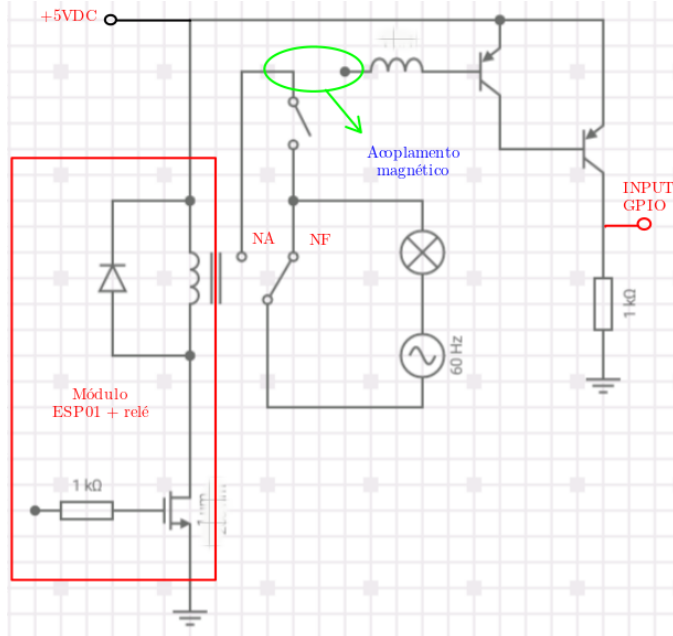
3 Detalhamento do projeto

3.1 Lâmpadas

3.1.1 Descrição do circuito

Um pequeno trafo recebe a energia da tomada, é retificada por uma ponte retificadora e então o módulo relé com o esp8266 controla o chaveamento da lâmpada. Para fazer o controle da lâmpada ser manual torna-se necessário detectar a existência de fase no pino Normalmente Aberto (NA) do relé, como na figura 1.

Figure 1: Circuito a ser implementado para detecção de fase



Não é intenção deste projeto confeccionar placa de circuito impresso para simplificar o projeto e também no momento é impossível para mim imprimir sem uma impressora adequada.

3.1.2 Componentes utilizados (por lâmpada)

- ☒ 1 Trafo de carregador;
- ☒ 4 Diodos 1n4007;
- ☒ 1 Capacitor eletrolítico (47uF);
- ☒ 1 Capacitor cerâmico (100nF);
- ☒ 1 Sensor piroelétrico
- ☒ 1 Módulo de acionamento de relé por ESP8266 (figura 2;
- ☒ 2 transistores de uso geral para para detecção de fase;
- ☒ Resistores diversos

O módulo de relé possui o esquemático como na figura ??

3.2 Descrição do software

O projeto de software é dividido em 3 partes: Conectividade e gerenciamento de ações; GUI; geração de relatórios

3.2.1 Conectividade e gerenciamento de ações

Esta parte consiste em fazer os ESP8266 se conectarem com o raspberryPI por rede para estabelecer comunicação (vide figura 4) e também consiste nas tomadas de decisão para o raspberryPI, determinando o comportamento de cada lâmpada e dando prioridade aos comandos. Os esp8266 das tomadas devem entrar em um ponto de acesso central e então ficar à espera de comandos. Ele age como escravo para responder aos comandos do computador central.

- Atividades de pesquisa e implementação:
 - Protocolo de comunicação (http)
 - * Usar os esps como servidores, de modo que o raspberry consiga solicitar informações.
 - Secure shell (ssh) para compartilhar tela
 - Programação dos ESP8266
 - * frameworks: Arduino, micropython, RTOS?

- Procedimentos a serem utilizados na cpu principal:
 - `get state()` # Retorna o estado atual lâmpada;
 - `turn(boolean state)` # Pede para ligar/desligar a lâmpada
 - `get switch()` # Retorna a posição do interruptor;

3.2.2 GUI

Uma interface gráfica para o usuário como a da figura 6 é tida como meio de centralizar as informações de forma que fique acessível ao usuário. Esta será feita no raspberryPI IOs com a biblioteca Qt for python, que é uma versão alternativa ao PyQt com licença LGPL, para caso o projeto futuramente se torne comercial.

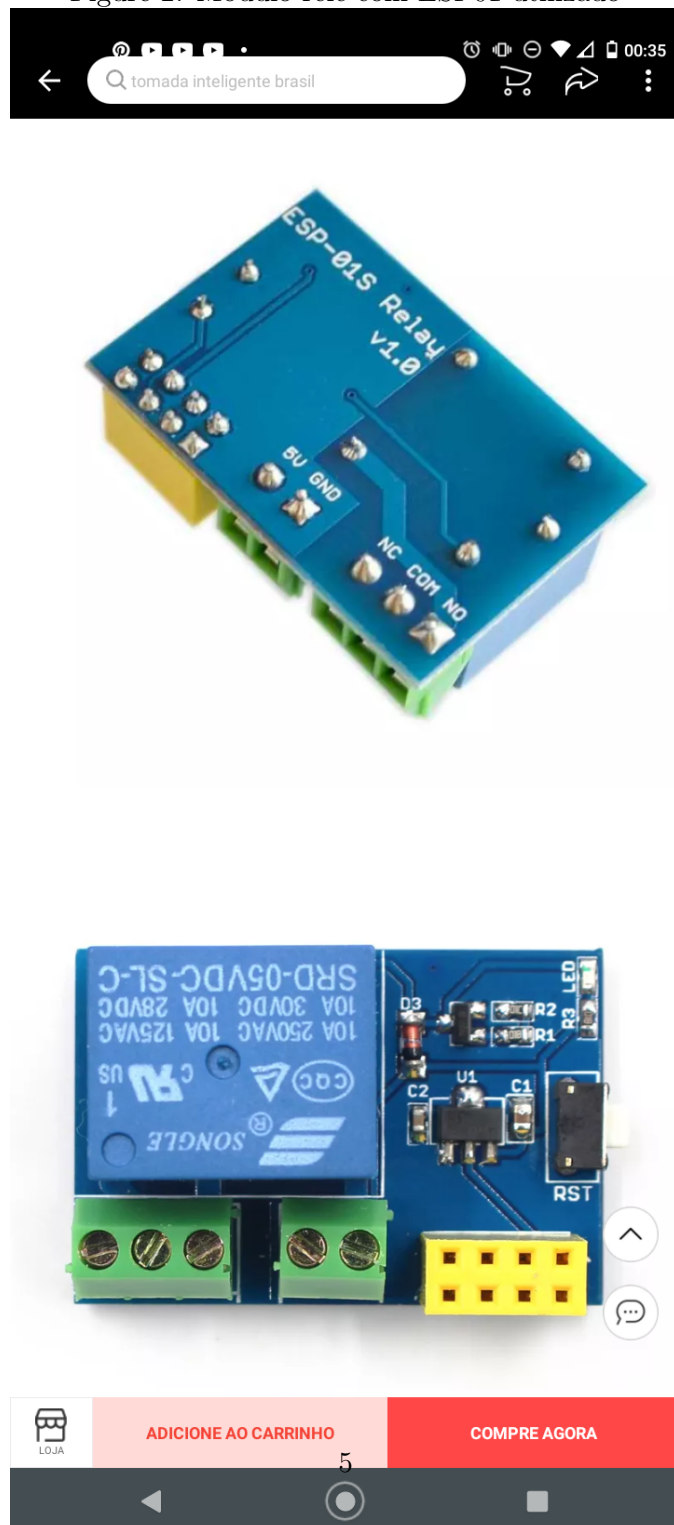
- Atividades de pesquisa e implementação TODO!!!

3.2.3 Geração de relatórios

Esta parte do projeto consiste em trabalhar com as informações obtidas com as lâmpadas, visa calcular consumos e gerar um histograma para o consumo de energia dos dispositivos.

- Atividades de pesquisa e implementação TODO!!!

Figure 2: Módulo relé com ESP01 utilizado



LOJA

ADICIONE AO CARRINHO

COMPRE AGORA

Figure 3: Esquema do circuito do módulo com relé

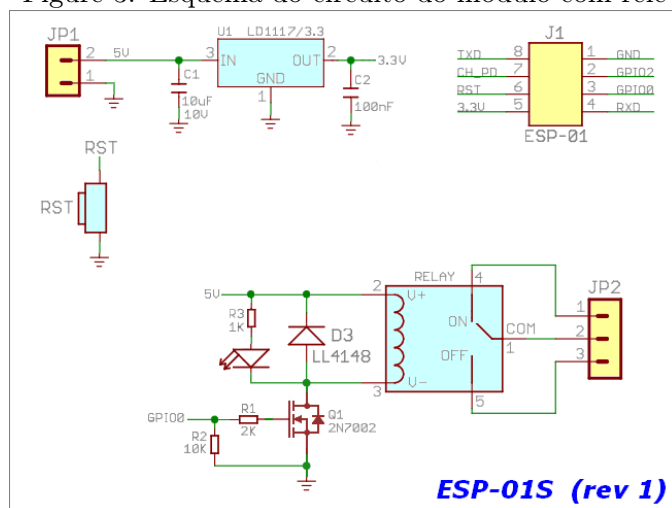


Figure 4: Visão conceitual para conectividade LAN dos dispositivos

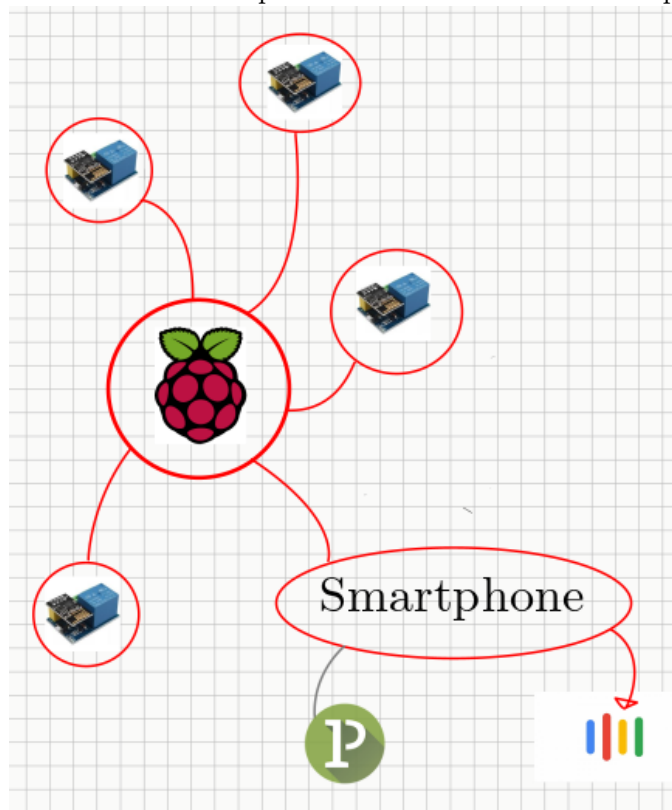


Figure 5: Diagrama de caso de uso

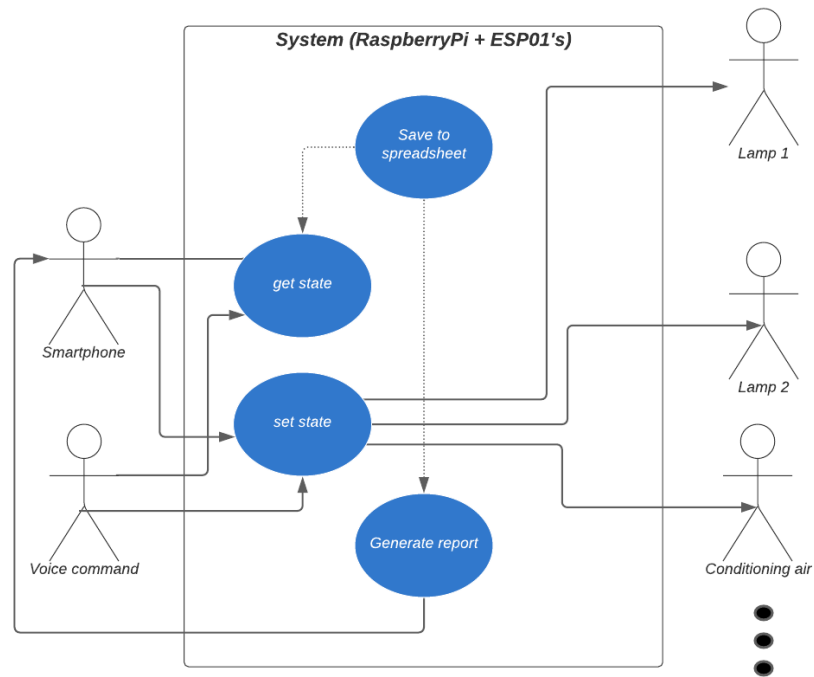


Figure 6: GUI a ser implementada no RaspberryPI OS

