

Projeto automação residencial

Ruan Flaneto Cartier

July 27, 2021

Contents

1	Motivação	1
2	Objetivos	2
3	Detalhamento do projeto	2
3.1	Lâmpadas	2
3.1.1	Descrição do circuito	2
3.1.2	Componentes utilizados (por lâmpada)	2
3.2	Controle do ar condicionado	3
3.2.1	Descrição do hardware	3
3.2.2	I2C revisão	3
3.3	Descrição do software	3
3.3.1	Criar servidor local/Aberto	4
4	Caso de uso	4

1 Motivação

Um projeto de automação residencial foi demandado. Primeira coisa que vem em mente é poder controlar as lâmpadas de casa individualmente. Assim, pretende-se usar um ESP8266 para cada ponto de interruptor de lâmpada para poder ter conexão com o computador central (raspberry pi). Temos disponíveis sensores piroelétricos, que são úteis para desligar as lâmpadas automaticamente na ausência de pessoas no cômodo.

2 Objetivos

Gerenciar o funcionamento das lâmpadas de casa, cujas ações de desligar ou ligar possam ser configuradas como automáticas (depende do sensor de presença) ou manuais (depende de comandos por smartphone, sejam botões e/ou comandos de voz). Este gerenciamento também inclui a formação de relatórios sobre consumo elétrico (estimado) em cada dispositivo, apresentando as informações em histogramas e suas respectivas conclusões. Um objetivo secundário seria implementar o controle de ar condicionado, em que a diferença para a lâmpada seria o controle (malha aberta) de temperatura e o registro da mesma no relatório.

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. Não menos importante, o interruptor da tomada dever ser alimentado por um resistor, cujo estado é lido por uma porta digital. Quase esqueci dos sensores de presença. Devido ao espaço ocupado, novos interruptores devem ser comprados. Sendo assim, o μC precisará de 3 portas digitais para controlar os periféricos e mais talvez duas para poder programar em ISP. Pretendo não fazer placa de circuito impresso para simplificar o projeto e tb 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 Conjunto de interruptor; (Precisa comprar)
- ☒ 1 Módulo de acionamento de relé por ESP8266 (figura 1)

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

3.2 Controle do ar condicionado

3.2.1 Descrição do hardware

O computador principal se conecta ao controle do ar condicionado através dos barramentos de I2C do display e do microcontrolador. O primeiro barramento seria usado para o computador central identificar as configurações atuais do ar condicionado e o segundo serve para fazer alterações nas configurações. Para tanto é preciso fazer uma revisão sobre o protocolo e interpretar os dados lidos.

3.2.2 I2C revisão

Ao abrir o controle do ar condicionado foram encontrados os pinos 36,37,38 e 44 acessíveis ao usuário. Como mostrado na figura 3. Claramente eles servem para estabelecer comunicação i2c entre o chip e um computador externo.

Fui procurar circuitos de controle remoto que aplicam este microcontrolador e achei um resultado interessante, como na figura 4.

Logo fiquei na dúvida o que seria o VPP do pino 36, pesquisei no datasheet e achei este resultado (vide figura 5).

Depois disso notei que precisava rever um pouco sobre i2c e achei as seguintes figuras chave: TODO ...

3.3 Descrição do software

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 e também irá enviar mensagens durante a comutação do sensor piroelétrico (descobrir se não vai haver realimentação positiva com a lâmpada) Protocolo de comunicação: Tem que descobrir uma forma de protocolar as mensagens. O receptor vai ler a mensagem e vai decodificá-la. Após decodificar, vai executar a ação de desligar/ligar.

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;

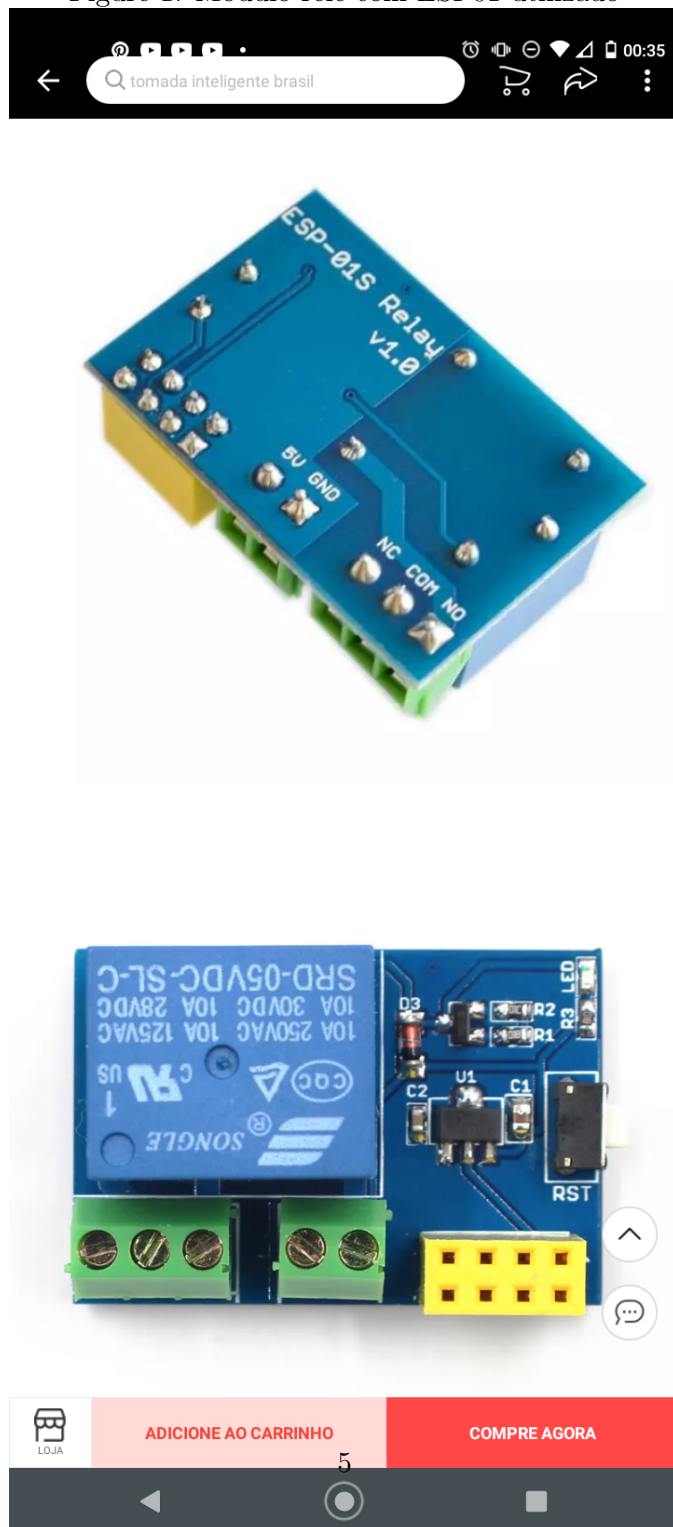
3.3.1 Criar servidor local/Aberto

- Conseguir externar o ip desse servidor

**

4 Caso de uso

Figure 1: Módulo relé com ESP01 utilizado



LOJA

ADICIONE AO CARRINHO

COMPRE AGORA

The diagram shows the PCB layout for the ESP-01S (rev 1) module. It includes the main board and a separate relay module.

Main Board Components:

- U1:** LD1117/3.3 voltage regulator.
- C1:** 10uF 10V electrolytic capacitor.
- C2:** 100nF ceramic capacitor.
- JP1:** Header for 5V and GND.
- RST:** Reset button.

Relay Module Components:

- RELAY:** Relay with U+ and U- terminals.
- D3:** LL4148 diode.
- Q1:** 2N7002 MOSFET.
- R1:** 2K resistor.
- R2:** 10K resistor.
- R3:** 1K resistor.
- JP2:** Header for 1, 2, 3, 4, 5 pins.

Pin Connections:

- 5V:** Connected to U1 pin 1, C1 positive, and RELAY U+.
- GND:** Connected to U1 pin 2, C1 negative, C2 negative, RST, and RELAY U-.
- GPIO0:** Connected to R2, which is connected to the gate of Q1.
- GPIO2:** Connected to the drain of Q1, which is connected to the coil of the RELAY.
- GPIO4:** Connected to the coil of the RELAY.
- GPIO5:** Connected to the coil of the RELAY.
- GPIO8:** Connected to the coil of the RELAY.
- GPIO10:** Connected to the coil of the RELAY.
- GPIO12:** Connected to the coil of the RELAY.
- GPIO14:** Connected to the coil of the RELAY.
- GPIO16:** Connected to the coil of the RELAY.
- GPIO18:** Connected to the coil of the RELAY.
- GPIO20:** Connected to the coil of the RELAY.
- GPIO22:** Connected to the coil of the RELAY.
- GPIO24:** Connected to the coil of the RELAY.
- GPIO26:** Connected to the coil of the RELAY.
- GPIO28:** Connected to the coil of the RELAY.
- GPIO30:** Connected to the coil of the RELAY.
- GPIO32:** Connected to the coil of the RELAY.
- GPIO34:** Connected to the coil of the RELAY.
- GPIO36:** Connected to the coil of the RELAY.
- GPIO38:** Connected to the coil of the RELAY.
- GPIO40:** Connected to the coil of the RELAY.
- GPIO42:** Connected to the coil of the RELAY.
- GPIO44:** Connected to the coil of the RELAY.
- GPIO46:** Connected to the coil of the RELAY.
- GPIO48:** Connected to the coil of the RELAY.
- GPIO50:** Connected to the coil of the RELAY.
- GPIO52:** Connected to the coil of the RELAY.
- GPIO54:** Connected to the coil of the RELAY.
- GPIO56:** Connected to the coil of the RELAY.
- GPIO58:** Connected to the coil of the RELAY.
- GPIO60:** Connected to the coil of the RELAY.
- GPIO62:** Connected to the coil of the RELAY.
- GPIO64:** Connected to the coil of the RELAY.
- GPIO66:** Connected to the coil of the RELAY.
- GPIO68:** Connected to the coil of the RELAY.
- GPIO70:** Connected to the coil of the RELAY.
- GPIO72:** Connected to the coil of the RELAY.
- GPIO74:** Connected to the coil of the RELAY.
- GPIO76:** Connected to the coil of the RELAY.
- GPIO78:** Connected to the coil of the RELAY.
- GPIO80:** Connected to the coil of the RELAY.
- GPIO82:** Connected to the coil of the RELAY.
- GPIO84:** Connected to the coil of the RELAY.
- GPIO86:** Connected to the coil of the RELAY.
- GPIO88:** Connected to the coil of the RELAY.
- GPIO90:** Connected to the coil of the RELAY.
- GPIO92:** Connected to the coil of the RELAY.
- GPIO94:** Connected to the coil of the RELAY.
- GPIO96:** Connected to the coil of the RELAY.
- GPIO98:** Connected to the coil of the RELAY.
- GPIO100:** Connected to the coil of the RELAY.

Pinout diagram of the SH77P1651 microcontroller. The chip is shown with pins 1 through 48. Pins 35, 37, 39, 41, 43, 45, 47, and 48 are highlighted with red circles. The pin functions are listed around the chip:

- P0.7/VPP
- P1.0/INT40/SEG1
- P1.1/INT41/SEG2
- P1.2/INT42/SEG3
- P1.3/INT43/SEG4
- P1.4/INT44/SEG5
- P1.5/INT45/SEG6
- P1.6/INT46/SEG7
- P1.7/INT47/SEG8
- P2.0/SEG9
- P2.1/RXD/SEG10
- P2.2/TXD/SEG11
- P2.3/SEG12
- P2.4/T2/SEG13
- P2.5/T2EX/SEG14
- P2.6/T3/SEG15
- P2.7/AN7/SEG16
- P3.0/SEG17
- P3.1/AN1/SEG18
- P3.2/AN2/SEG19
- P3.3/AN3/SEG20
- P3.4/AN4/SEG21
- P3.5/AN5/SEG22
- P3.6/AN6/SEG23
- XTAL1/P5.3
- VDD
- REM/PWM1/P5.4
- GND
- RESET/P5.5
- VP2/P0.2
- VP1/P0.3
- CUP2/P0.4
- SCK/P0.5
- SDA/P0.6

Figure 4: Pinos i2c disponíveis ao usuário.

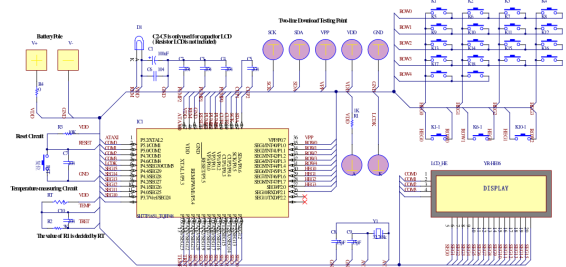


Figure 5: Pinos i2c disponíveis ao usuário.

Pin	Type	Description
Programming Port		
VDD	P	Programming Power (+3.3V)
VPP	P	Programming High Voltage Power (+7.5V)
GND	P	Ground
SCK	I	Programming Clock input Pin
SDA	I/O	Programming Data Pin
Note: When P0.5, P0.6 and P0.7 are used as test ports, I/O function is forbidden.		

Figure 6: Diagrama de caso de uso

