Controle Wifi – Instalação

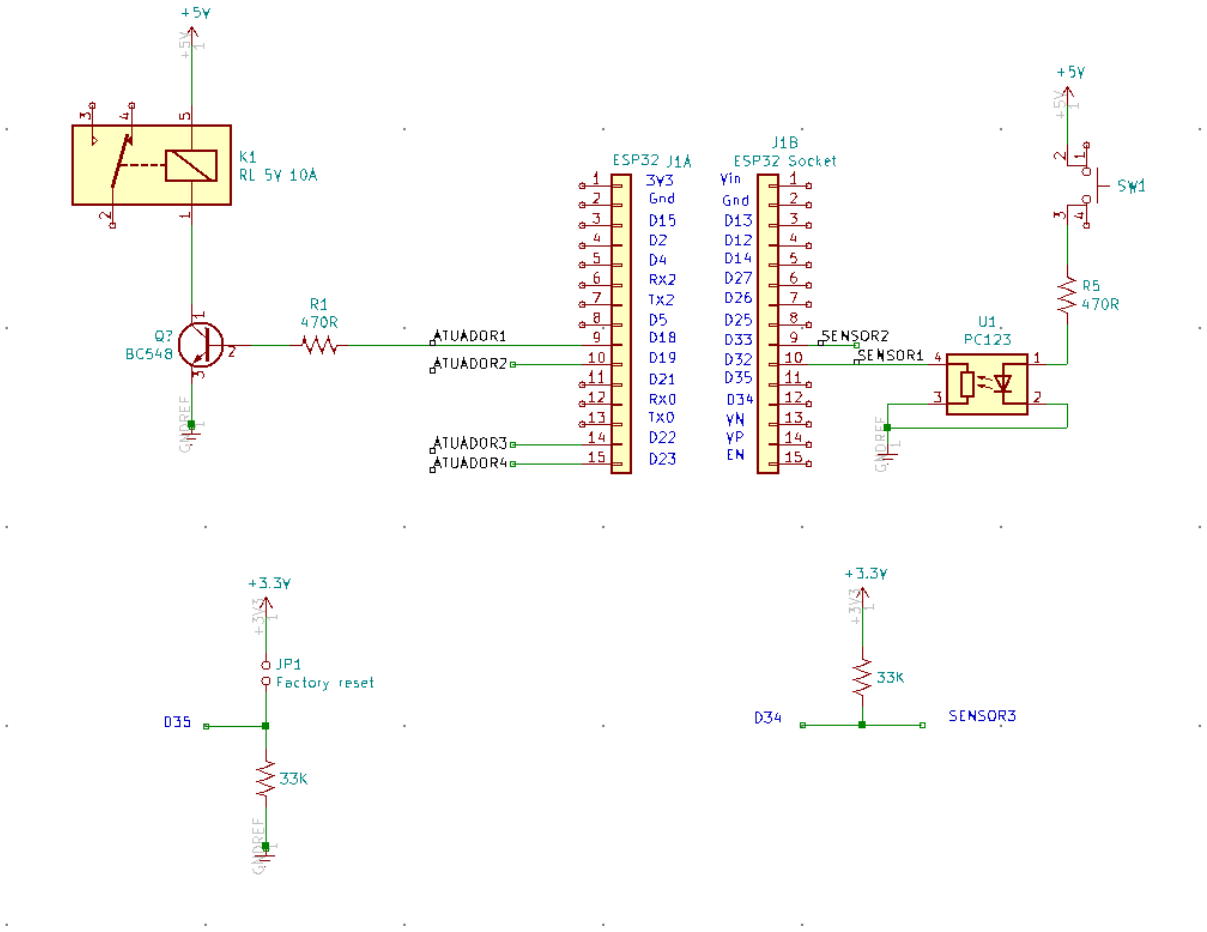
Apresentação

Este projeto tem por objetivo auxiliar os programadores do ESP-32 de nível iniciante ou intermediário com um exemplo bem completo de recursos, que podem ser modificados para criar seus próprios projetos. Também é útil para programadores Javascript que desejem utilizar o micro-controlador em seus projetos sem se aprofundar na programação do micro-controlador em linguagem C. Isto porque foi criado um nível de abstração que permite controlar alguns recursos do micro-controlador (que chamamos de atuadores e sensores) através de um protocolo HTTP, que pode ser implementado do lado do cliente em Javascript, Python ou outra linguagem de programação capaz de criar clientes HTTP.

O código se divide no programa a ser instalado no micro-controlador e um exemplo de cliente em Javascript, que pode ser modificado conforme a aplicação final desejada.

Hardware

O hardware pode ser montado a partir de uma placa de desenvolvimento para o ESP-32, conforme o diagrama abaixo:



Circuito do módulo de controle

No código e circuito fornecido, os atuadores são do tipo binário, assim como os sensores.

Com alguma habilidade, é possível ampliar o protocolo e acrescentar outros periféricos, como DAC, PWM, ADC, Displays, …

O projeto foi testado em uma placa ESP-Doit. São necessários no mínimo 2Mb de memória FLASH.

O pino GPIO35 tem uma função especial: Quando mantido em nível alto durante o processo de ‘boot’, a configuração do módulo mantida na memória FLASH é ignorada, e é carregada a configuração ‘de fábrica’ escolhida nas opções de compilação do sistema, como será visto adiante.

Preparação do código

Baixe os arquivos em uma pasta onde você possa compilar o programa e instalar no micro-controlador. Conforme recomendado no site da Espressif, esta pasta deveria estar no mesmo nível de pasta onde foi instalado o ambiente de desenvolvimento esp-idf (Veja o Guia de Programação do ESP-IDF).

Siga as instruções sobre como criar seu projeto, e, ao executar:

```
idf.py menuconfig
```

Escolha os parâmetros como nome e porta do servidor, ssid e senha do Wifi, etc. Alguns desses parâmetros valem como ‘configuração de fábrica’, mas, em operação normal, são substituídos pelos valores gravados no arquivo de configuração em memória FLASH.

Veja também na pasta flash_data um arquivo de configuração que será gravado em memória FLASH. Você pode alterar alguns parâmetros, que serão utilizados quando o módulo for iniciado.

Após criar o arquivo binário, instale o sofware no módulo utilizando o comando:

```
idf.py -p <PORTA> flash
```

Se desejar visualizar mensagens emitidas pelo módulo, acrescente a opção ‘monitor’ ao comando acima.

Para testar o módulo, é necessário instalar o software do cliente em um computador capaz de se conectar por Wifi ao módulo. Copie a pasta com o exemplo de cliente para este computador.

Se a opção softap=1 do arquivo de configuração for mantida, ou se o pino de ‘configuração de fábrica’ for mantido em nível alto, o módulo inicia no modo Wifi Soft AP, criando um ponto de acesso independente com o nome configurado. Se a configuração utilizar a opção softap = 0, o módulo irá se conectar ao ponto de acesso Wifi com nome e senha configurados.

Conecte-se com o módulo pelo Wifi: Escolha o ssid e forneça a senha. Com o computador conectado, abra o arquivo index.html utilizando um navegador de internet. Neste ponto, será possível testar a placa utilizando os botões disponíveis no aplicativo cliente.

Na tela principal, existe um botão configurar. Utilize este botão para alterar as opções de configuração, como nome do servidor, ssid e senha do Wifi, e modo de operação AP ou Station. Se a configuração for Salva, os valores serão armazenados na memória FLASH e serão utilizadas na próxima vez que o módulo for iniciado (ver observação sobre configuração ‘de fábrica’.

Modificando o Aplicativo Cliente

Para um programador Javascript, não deve ser difícil modificar o código do cliente para adequá-lo à aplicação desejada. Sugere-se que, a menos que o programa seja utilizado em um ambiente de testes e desenvolvimento, o botão e o código que permite configurar o módulo seja removido, pois uma configuração descuidada realizada por um usuário do software pode afetar o funcionamento do sistema (por exemplo, impedindo que o módulo faça login no Wifi por conta de uma senha errada), o que iria tornar necessário que se retorne à configuração de fábrica e reconfigure o módulo para corrigir o problema.

Em contrapartida, o integrador/instalador do módulo deve possuir à mão uma cópia do sofrware cliente que permita a configuração e teste do módulo. Uma possibilidade é ter o cliente Javascript dentro do celular e um App de Servidor HTTP instalado, que disponibilize os arquivos como recursos navegáveis para um navegador de Internet instalado. Um exemplo possível é o App: Simple HTTP Server.

Modelo de Abstração e Protocolo HTTP

No projeto, foi utilizada uma camada de abstração de hardware no código C que é instalado no módulo, implementada no arquivo controle_gpio.c, que esconde dos demais componentes os detalhes de alocação e configuração dos diversos pinos do micro-controlador. O modelo que é exposto consiste de classes de atuadores e sensores, sendo cada classe numerada sequencialmente de 1 até o máximo de dispositivos disponíveis na classe. Nesta implementação inicial, os dispositivos são binários, de forma que se pode alterar o estado dos atuadores entre ligado (1) e desligado (0). Da mesma forma, pode-se ler um sensor, com o mesmo tipo de resultado.

Protocolo HTTP

O modelo tem uma camada de mais alto nível usando protocolo HTTP, onde as solicitações seguem o protocolo:

GET /status

Para obter um texto indicando o status do controlador.

GET /sensor?id=<identificador>

Para ler o estado de um sensor (porta de entrada do módulo), Onde id indica o sensor para o qual se deseja obter o valor.

POST /atuador<identificador>

action=(‘off’, ‘on’, ‘toggle’ ou ‘pulse’)

[duration=(tempo em ms)]

Para alterar o estado de um atuador (porta de saída do módulo), Onde ‘off’ desliga, ‘on’ liga, ‘toggle’ alterna o estado e ‘pulse’ liga e desliga após duração de tempo determinada.

POST /config

ssid=(ssid do Wifi)

password=(senha do Wifi)

wifi_mode=(STA/AP)

hostname=(nome do servidor HTTP)

Para alterar a configuração do módulo.

O modelo pode ser estendido para outras classes de dispositivos.

Domínio da Aplicação

No Aplicativo cliente, a interface com o usuário traduz os identificadores numéricos de cada dispositivo em nomes pertencentes ao domínio da aplicação. Os níveis elétricos lidos ou gerados pelos dispositivos passam a ter um sentido concreto, como a existência de movimento em uma sala, uma porta aberta, uma campainha pressionada, ligar uma lâmpada, soar uma sirene ou abrir um portão.

No exemplo em Javascript, o comportamento de atuadores e sensores está encapsulado nas classes Sensor e Atuador, respectivamente.

Limitações

A ênfase do projeto foi oferecer uma ferramenta simples para estudar e criar protótipos de dispositivos de automação. Fatores como confiabilidade, segurança e conformidade com protocolos existentes foram deixados de lado.

No caso de aplicações em projetos IOT, há protocolos mais apropriados do que HTML combinado com Wifi, mas pesou na escolha a simplicidade de desenvolvimento do cliente, dada a grande disponibilidade de ferramentas próprias para isto. Desde a versão 0.82, existe uma tentativa de aproximar o protocolo de APIs REST. Uma das diferenças é que, por serem ainda muito simples, com apenas uma camada de informações, o conteúdo das solicitações POST ainda é em texto (text/plain), e não em JSON.

Também há oportunidade para expor outras classes de dispositivos, como conversores de valores digitais para analógicos e vice-versa. Também é possível implementar contadores utilizando interrupções geradas por alterações no nível do sinal de um sensor.

Sumário