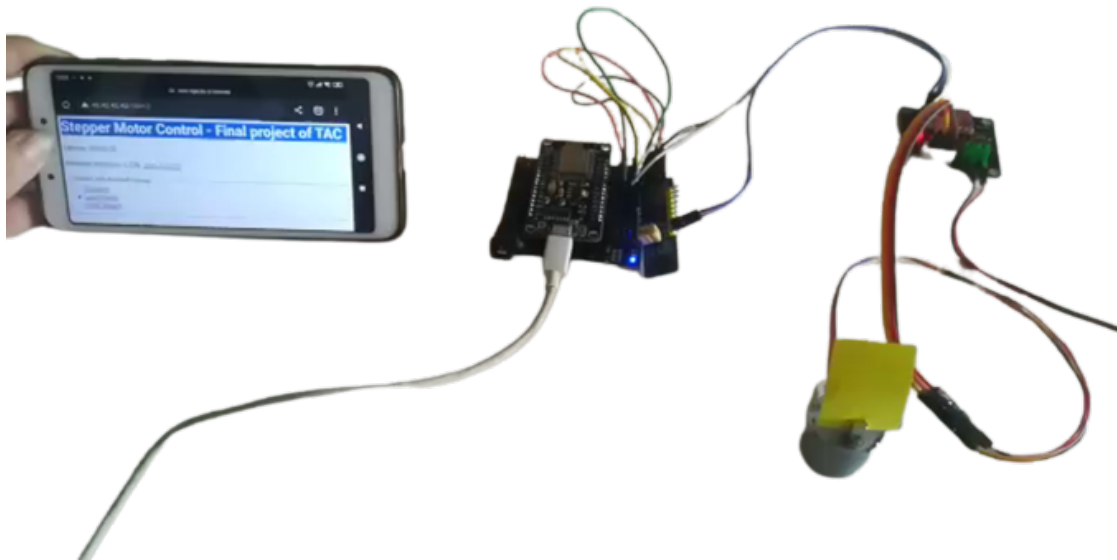


UNIVERSIDADE DE AVEIRO

TECNOLOGIAS DE ACIONAMENTO E COMANDO

Controlo de um motor de passo - Projeto final de TAC



93038 - Tatiana Pinho Resende

10 de janeiro de 2023



universidade
de aveiro

Palavras Chave

Protocolo de comunicação I2C	ESP8266	Web Server
Protocolo de comunicação WiFi	Motor de passo	Módulo de extensão I/O

Resumo

Este projeto, desenvolvido para a cadeira de Tecnologias de Acionamento e Comando, visa o controlo de um motor de passo. Trata-se na criação de uma interface para controlar um motor de passo. Isto inclui a definição do sistema elétrico, a criação de um programa para comunicação e controlo do motor e, finalmente, a criação de uma *Web Page* que permite a qualquer utilizador que se conecte ao ESP, o controlo do motor.

A *Web Page* criada permite definir o sentido de rotação, a velocidade e o ângulo de movimento relativo que se pretende que o motor de passo execute.

Conteúdo

1. Objetivo	3
2. Esquemático das ligações	3
3. Componentes	4
3.1 ESP8266	4
3.2 Módulo expensor I/O PCF8574	5
3.3 Stepper Drive HR4988sq	5
3.4 Stepper Motor 24byj48s	5
4. Comunicações	6
4.1 I2C	6
4.2 WiFi	7
5. Descrição do funcionamento	7
6. Demonstração	8

1. Objetivo

Este projeto, desenvolvido para a cadeira de Tecnologias de Acionamento e Comando, visa o controlo de um motor de passo. Trata-se na criação de uma interface para controlar um motor de passo. Isto inclui a definição do sistema elétrico, a criação de um programa em *Arduíno IDE (Arduino Integrated Development Environment)* para comunicação e controlo do motor e, finalmente, a escrita de um código em *HTML (Hypertext Mark-up Language)* que permite a qualquer utilizador que se conecte ao ESP, o controlo do motor.

Este projeto é relevante para outros do tipo “Máquina de lançamento de bolas de futebol”, desenvolvido no departamento de engenharia mecânica da universidade de Aveiro. O objetivo deste é comunicar e controlar os seis motores existentes no sistema. Em situações como essa, para controlar cada motor, requer-se duas ou quatro saídas digitais, conforme o drive empregado. Em alguns casos, um dispositivo ESP, como o ESP8266 usado nesta pesquisa, não tem pinos de i/o (*inputs/outputs*) digitais suficientes, sendo necessário um extensor de i/o digitais.

2. Esquemático das ligações

A figura 1 representa o esquemático das ligações.

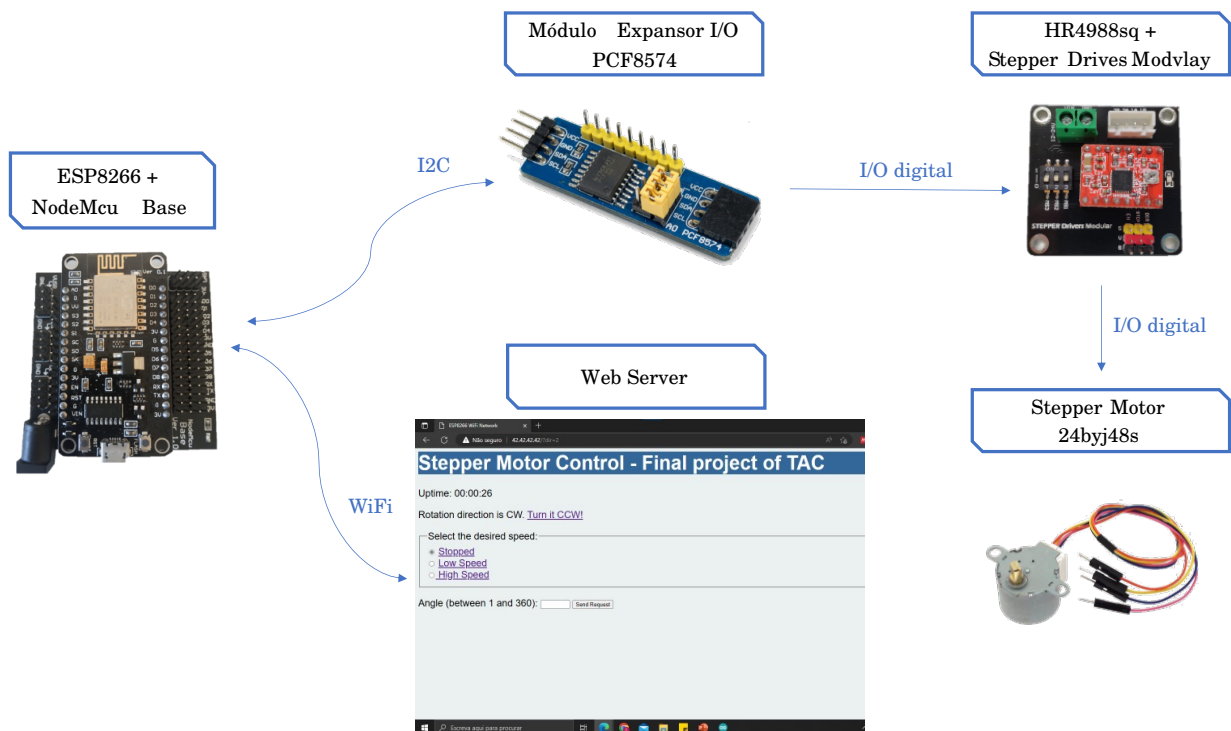


Figura 1: Esquemático das ligações

3. Componentes

Na presente secção, são enumerados os vários elementos envolvidos no projeto, bem como uma sucinta explicação de cada um deles.

3.1 ESP8266

A placa de desenvolvimento IOT (*Internet of Things*) NodeMCU ESP8266 CP2102 trabalha em conjunto com o WiFi ESP8266 para se conectar a redes WiFi (*Wireless Fidelity*). O NodeMCU é pequeno e possui o seu próprio microcontrolador de 32 bits. Possui uma entrada USB Micro que serve tanto para conexão com um computador como para a alimentação, um conversor USB-SERIAL CP 2102 para lidar com a comunicação entre o computador e o microcontrolador durante a programação e um regulador de tensão AMS1117 para reduzir a tensão de entrada de 5V para 3,3V [1].

A placa de desenvolvimento ESP8266 possui um total de 30 pinos que a conectam ao mundo externo. Para simplificar, os pinos com funcionalidade semelhante são agrupados. O *pinout* é representado na Figura 2.

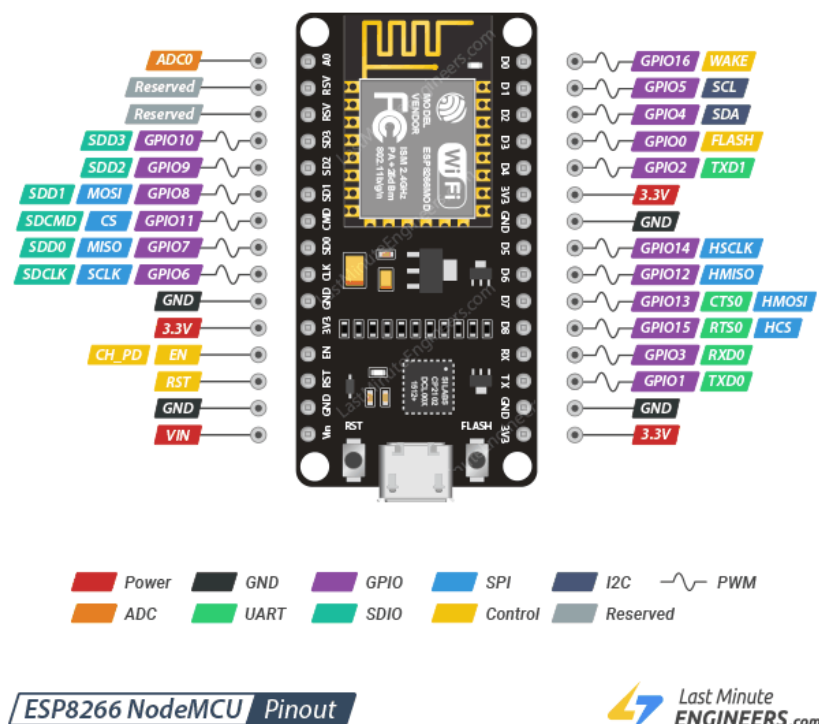


Figura 2: Pinout ESP8266 NodeMCU [2]

Também foi utilizada a placa NodeMcu Base Ver 1.0 para simplificar o acesso aos pinos do ESP8266.

3.2 Módulo expensor I/O PCF8574

Este módulo é utilizado para expandir as portas do ESP8266 utilizando apenas dois pinos do ESP através da comunicação I2C (*Inter-Integrated Circuit*). Para isso o módulo inclui quatro pinos que são o VCC (alimentação 3,3V), GND (*ground*), SDA (*serial data*) e SCL (*serial clock*). Permite o controle de até oito pinos de I/O (P0 a P7), mais um pino de interrupção (INT). A configuração do endereço pode ser definida através dos *jumers* nos pinos A0, A1 e A2.

3.3 Stepper Drive HR4988sq

O driver do motor de passo tem uma capacidade de saída de até 35V e $\pm 2A$, o que permite o controle do motor de passo bipolar. Além disso, a corrente de saída é regulada, permitindo a operação silenciosa do motor de passo e a eliminação de ressonância ou zumbido.

O driver contém um tradutor integrado para simplificar a operação, isso diminui o número de pinos de controle para somente dois, um para regular os passos e outro para alterar a direção de rotação. O driver apresenta cinco variações de resolução de passo: passo completo, meio passo, quarto de passo, oitavo passo e décimo sexto passo. Os recursos adicionais do drive, como a subtensão, o disparo, o curto-circuito, a sobrecorrente e a proteção térmica, garantem uma operação segura [3].

O driver possui um total de 16 pinos que o conectam ao mundo exterior. O *pinout* é representado na Figura 3.

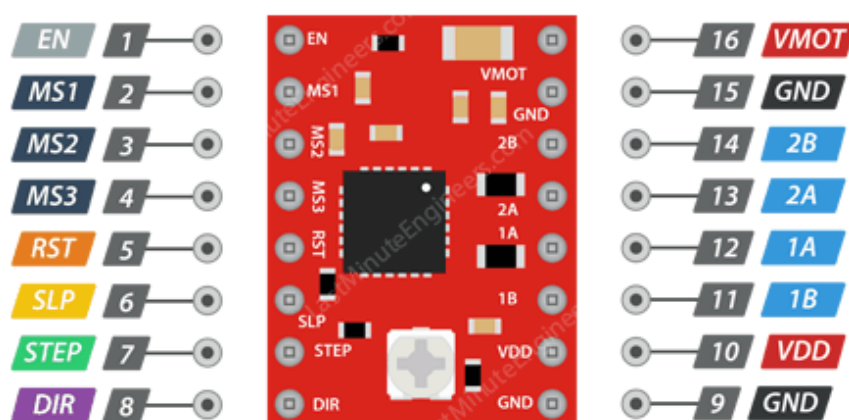


Figura 3: *Pinout* Stepper Drive HR498sq [3]

3.4 Stepper Motor 24byj48s

O motor de passo 24byj48s apresenta um tamanho pequeno e alto torque. Segundo a documentação, presente no Anexo A na figura 5, é composto por quatro fases e apresenta

2048 passos por revolução e um ângulo de passo $5.625^\circ/32$. É um motor de baixo custo voltado para mercados de *hobby* e geralmente têm um padrão inferior ao de produtos destinados ao uso industrial.

Ao longo do projeto notou-se que os passos por revolução não eram os 2048 descritos na documentação, mas sim de 1051,8 passos/revolução. Essa conclusão é baseada num fórum, [4], em que descrevem o processo para chegar a esse valor. O processo requer a desmontagem do motor, o que relevou que este possui duas bobinas, cada uma com 16 posições, perfazendo 32 passos para uma volta completa. A caixa redutora resulta numa relação de 1:32,86914, o que significa $32 \times 32,86914$ que dá origem a 1051,8 passos/revolução.

4. Comunicações

De forma a comunicar entre os diversos componentes são usados protocolos de comunicação como I2C e WiFi. Entre o ESP8266 e o módulo expensor i/o PCF8574 a comunicação dá-se por I2C. Entre o ESP8266 e o *Web Page* a comunicação é por WiFi.

4.1 I2C

O protocolo foi desenvolvido para permitir uma comunicação rápida e ter controlo sobre os registos nos dispositivos, bem como os dados que podem ser salvos. O barramento I2C consiste fisicamente em dois fios ativos (SCL e SDA) e uma massa. Esses fios, SCL e SDA, são bidirecionais do tipo *half duplex* [5].

Para a escrita da mensagem pode-se utilizar a biblioteca *Wire.h*. A mensagem começa com uma condição de início, *START*, que funciona como um aviso para todos os circuitos integrados conectados de que algo está para ser enviado.

```
1 Wire.begin();
```

Após o envio da condição de início, um byte pode ser transmitido a um *slave* pelo *master*. Este primeiro byte após uma condição de início identificará o *slave* no barramento (endereço) e selecionará o modo de operação. O significado de todos os bytes seguintes depende do *slave*.

```
1 #define expander B00100000 //expander address 0x20
2
3 Wire.beginTransmission(expander);
```

Uma vez que o *slave* foi endereçado e o *slave* reconheceu isso, um byte pode ser recebido do *slave* se o bit R/W (*read/write*) no endereço foi definido como *READ* (definido como '1'). Ao receber um byte de um *slave*, o *master* deve confirmar isso para o dispositivo *slave*. Se não houver mais dados para receber, o *master* enviará um sinal de não reconhecimento (NACK) e interromperá a transação de dados. Se o bit R/W for definido como *WRITE* (como '0') um byte pode ser enviado para o *slave*.

```
1 Wire.write(_data);
```

Depois que uma mensagem é concluída, uma condição de *STOP* é enviada.

```
1 Wire.endTransmission();
```

4.2 WiFi

O WiFi (que, em geral, tem largura de banda de 20 MHz) é usado por dispositivos móveis para obter alta taxa de dados. A operação sem fios apresenta diversas vantagens, tais como: possibilitar o uso da rede em qualquer ponto no alcance da transmissão; facilitar a inserção de dispositivos à rede; suportar uma grande quantidade de dispositivos conectados simultaneamente; evitar que paredes ou estruturas prediais sejam danificadas para a passagem de fios [6].

Inicialmente foi necessário definir o *access point* (AP) chamado TacEsp8266, com a palavra-passe Tac12345, através do código apresentado em baixo.

```
1 // NETWORK CREDENTIALS
2 const char* ssid = "TacEsp8266";
3 const char* password = "Tac12345";
4 // Define a web server at port 80 for HTTP
5 ESP8266WebServer server(80);
```

A seguir definiu-se o IP, 42.42.42.42, da página WEB que será a interface. Noutro instante, para informar o utilizador do IP definido, efetua-se a impressão da mensagem “AP IP address” e o IP definido.

```
1 IPAddress apIP(42, 42, 42, 42); // Defining a static IP address:
   local & gateway
2
3 /* set-up the custom IP address */
4 WiFi.mode(WIFI_AP_STA);
5 WiFi.softAPConfig(apIP, apIP, IPAddress(255, 255, 255, 0)); // subnet
   FF FF FF 00
6
7 IPAddress myIP = WiFi.softAPIP();
8 Serial.print("AP IP address: ");
9 Serial.println(myIP);
```

5. Descrição do funcionamento

O ESP8266 recebe informação de uma *Web Page* através de WiFi. Por I2C comunica com o módulo de expansão fazendo ativar/desativar os pinos "P0" e "P1", que no que lhe concerne, estão conectados aos pinos “dir” e “step” do drive do motor. Com a ativação/desativação destes pinos consegue-se controlar o movimento de rotação do motor. As ligações estão apresentadas no esquema elétrico presente no Anexo B, Figura 6.

A *Web Page*, Figura 4, criada permite definir o sentido de rotação, no sentido horário (*Clockwise*) ou no sentido anti-horário (*Counterclockwise*). Também é possível selecionar uma velocidade baixa ou alta. Notando que com a velocidade “stopped” o motor não roda. De forma a definir uma posição, pode-se dar o ângulo de movimento relativo que se pretende que o motor faça.

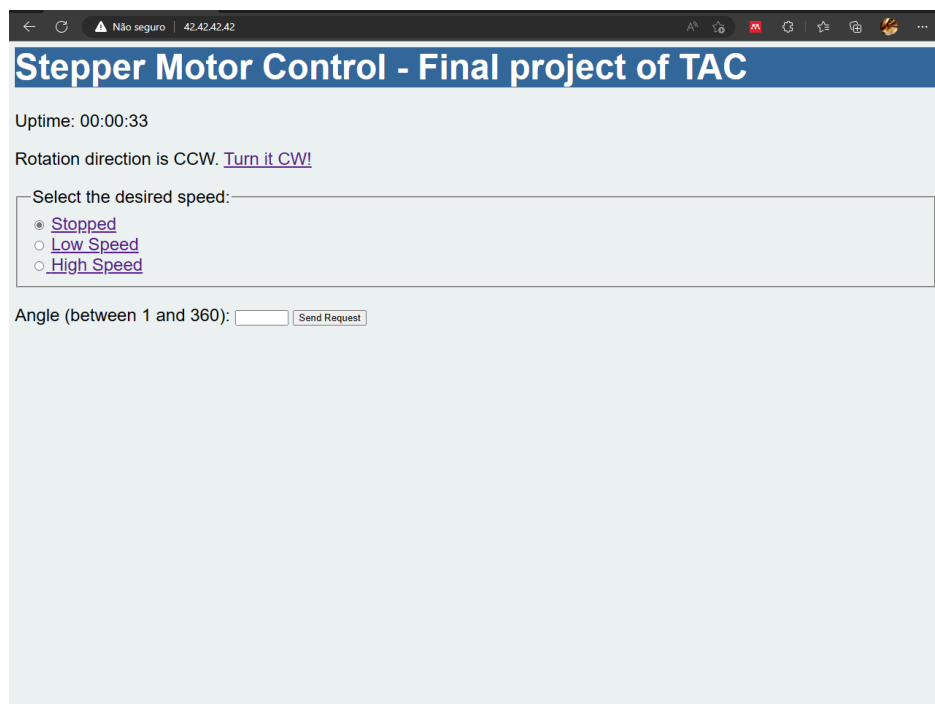


Figura 4: *WebPage: Stepper Motor Control - Final project of TAC*

6. Demonstração

Um vídeo demonstrando todo o funcionamento do controlo do motor de passo pode ser encontrado em URL: <https://youtu.be/ooWkEYa3iWE>

O código elaborado para este projeto pode ser consultado no GitHub: <https://github.com/TatianaResend/Controlo-de-um-motor-de-passo>

Referências

- [1] Espressif. *Docs Espressif – esp8266*. Acedido em: 09-01-2023. URL: <https://docs.espressif.com/projects/esp8266-rtos-sdk/en/latest/get-started/index.html>.
- [2] lastminuteengineers. *esp8266 - pinout reference*. Acedido em: 09-01-2023. URL: <https://lastminuteengineers.com/esp8266-pinout-reference/>.
- [3] lastminuteengineers. *stepper motor driver arduino tutorial*. Acedido em: 09-01-2023. URL: <https://lastminuteengineers.com/a4988-stepper-motor-driver-arduino-tutorial/>.
- [4] digikey. *how many steps does 24byj48s actually have per revolution*. Acedido em: 09-01-2023. URL: <https://forum.digikey.com/t/how-many-steps-does-24byj48s-actually-have-per-revolution/6202>.
- [5] P. Łukasz et al. *I 2 C Interface Design for Hardware Master Devices*. URL: <http://www.onsemi.com>.
- [6] infowester. *O que é WiFi? (conceito e versões)*. Acedido em: 10-01-2023. URL: <https://www.infowester.com/wifi.php>.

Small Size and High Torque
Stepper Motor - 24BY48

SKU: 108990003



(images/product/Motor 24BY48.jpg)

 **WIO LINK is LIVE at Kickstarter!**
(https://www.kickstarter.com/projects/seed/wio-link-3-steps-5-minutes-build-your-iot-applicat/description?ref=banner_depot)

Description

Small size and high torque, Mainly used in air conditioner louver, small cooling heating fan and Automatic curtains etc.

Specification:

- Phase: 4
- Step Angle: 5.625°/32
- Ratio: 1/32
- Voltage: 5V DC
- Resistance: 25ohm
- Idle In-traction Frequency: ≥500Hz
- Idle Out-traction Frequency: ≥800Hz
- In-traction Torque: ≥39.2mN/m
- Detent Torque: ≥20mN/m
- Insulation Class: A
- Noise: <40dB
- Steering: CCW
- Weight: 35g

Figura 5: Stepper Motor Datasheet

Anexo B

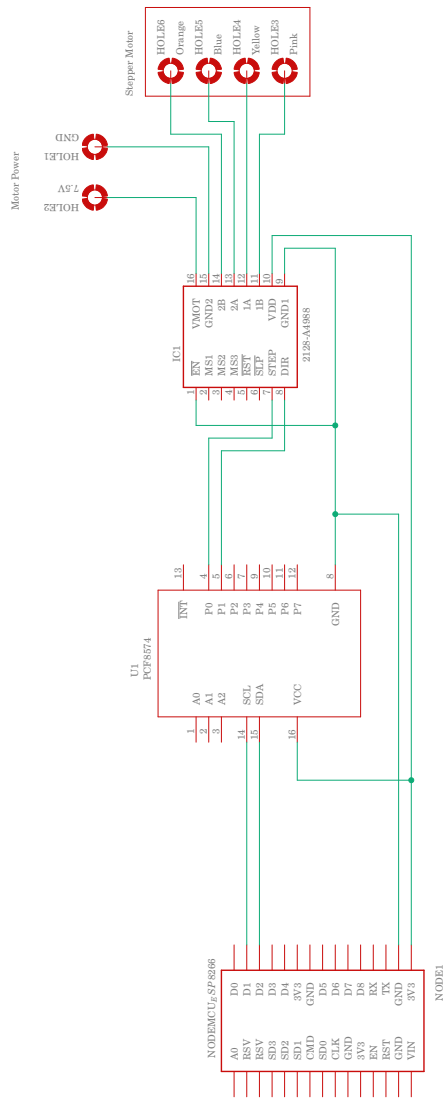


Figura 6: Esquema elétrico