

Guia para experimentos de controle de temperatura

Por: Henrique Assis
IMPA Tech-2025

1 Montando o circuito

Guia para montar o circuito.

1.1 Pinagens

Antes de iniciar, confira as pinagens de módulos e Arduino Uno:



Figura 1: Pinagem do Arduino

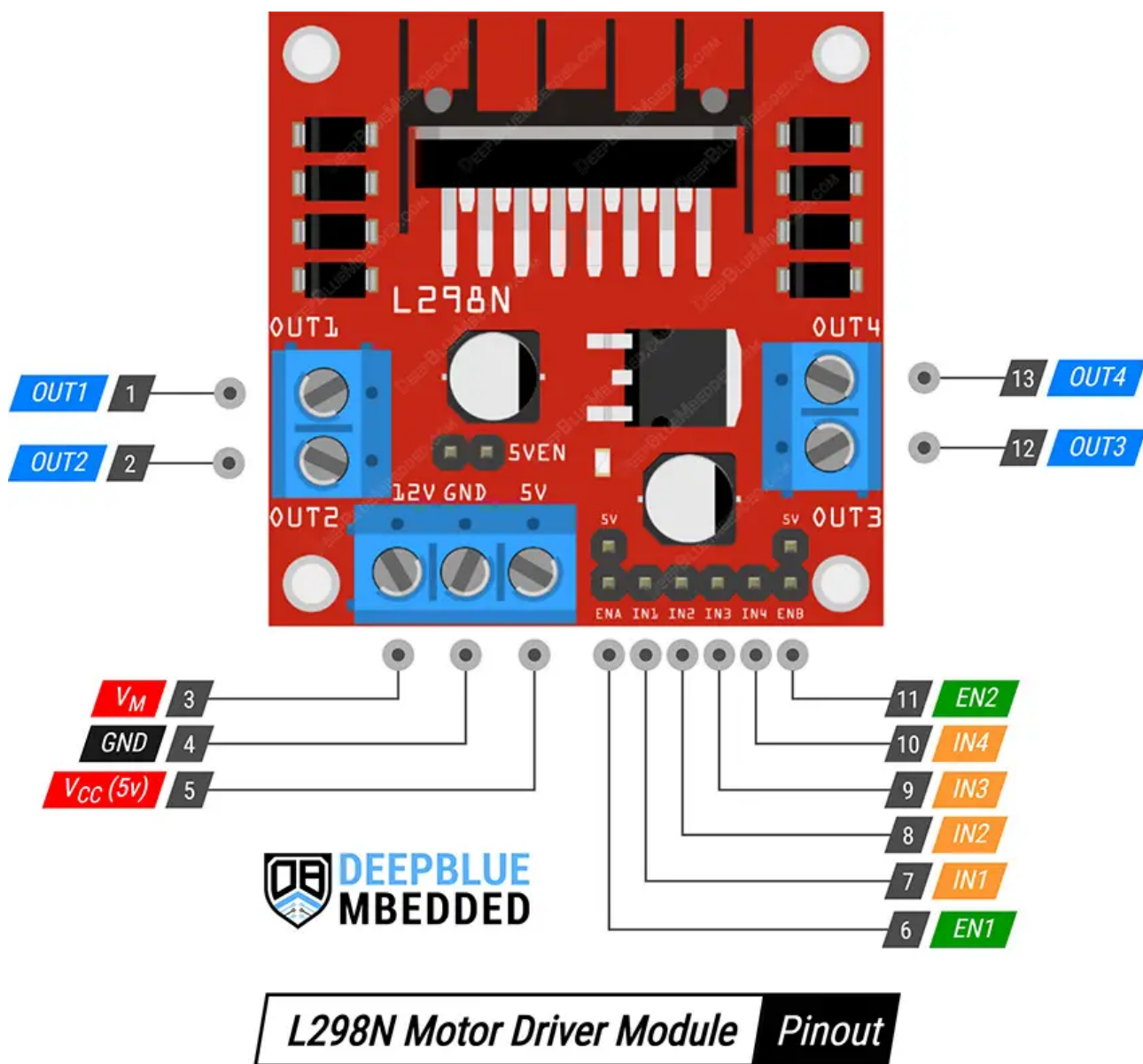


Figura 2: Pinagem do L298N

1.2 Montando geral

Com a fonte e o Arduino desligados.

Conecte o positivo da fonte em VM do L298N.

Conecte o negativo da fonte em GND do L298N.

Conecte o GND do L298N ao GND do arduino UNO.

Conecte o Vcc(5V) do L298N ao Vin do Arduino (opcional, e toda vez que conectar o Arduino ao computador terá que desfazer essa conexão, pois pode queimar o Arduino se ligar os dois ao mesmo tempo, essa conexão é utilizada para permitir que o Arduino seja ligado sem precisar do computador).

Ligue o 5V do Arduino a um dos terminais do resistor de 10K.

Ligue o outro terminal do resistor de 10K ao A0 do Arduino e também a um dos terminais do termistor.

Ligue o outro terminal do termistor que sobrou ao GND que sobrou do Arduino (o Arduino possui dos GND que estão ligados entre só).

Considerando que cores iguais estão conectadas entre si, faça a seguinte ligação:

| Arduino | L298N |
|-----------|-------|
| Digital 6 | EN1 |
| Digital 7 | IN1 |
| Digital 8 | IN2 |

Tabela 1: Pinos a se ligar

1.2.1 Montando com o peltier

Após fazer tudo que foi feito em 1.2

Ligue um dos fios do peltier em OUT1 do L298N e o outro fio do peltier em OUT2 do L298N.

ATENÇÃO: Caso o peltier esteja esquentando onde deveria esfriar e esfriando onde deveria esquentar, inverta essa conexão, o fio que era em OUT1 fica em OUT2, e onde era em OUT2 fica em OUT1, isso deve resolver o problema.

Agora o circuito foi montado.

Selecione um canal da fonte, deixe desligado, defina a tensão para 5V até 6V (de preferência 5.5V), defina a corrente para 2A, ligue o canal.

Caso esteja usando o Arduino ao computador, desfaça a ligação do Vcc(5V) do L298N com o Vin do Arduino, e ligue o Arduino no computador.

Agora o sistema já está atuando (se você fez o *upload* previamente do código).

1.2.2 Montando com resistor

Após fazer tudo que foi feito em 1.2.

Ligue um dos fios do resistor de potência em OUT1 do L298N e o outro fio do resistor de potência em OUT2 do L298N.

1.2.2.1 Montando com um resistor

Nesse caso o circuito foi montado, não precisa fazer mais nada

Selecione um canal da fonte, deixe desligado, defina a tensão para 10 V até 11V (de preferência 10 V), defina a corrente para 2A, ligue o canal.

Caso esteja usando o Arduino ao computador, desfaça a ligação do Vcc(5V) do L298N com o Vin do Arduino, e ligue o Arduino no computador.

Agora o sistema já está atuando (se você fez o *upload* previamente do código).

1.2.2.2 Montando com dois resistores

Além de tudo feito em 1.2 e 1.2.2 Faça :

| Arduino | L298N |
|-----------|-------|
| Digital 5 | EN2 |
| Digital 3 | IN3 |
| Digital 4 | IN4 |

Tabela 2: Pinos a se ligar para dois resistores

No outro resistor de potência, ligue um terminal ao OUT4 do L298N e o outro terminal ao OUT3 do L298N.

Selecione um canal da fonte, deixe desligado, defina a tensão para 10 V até 11V (de preferência 10 V), defina a corrente para 2A, ligue o canal.

Caso esteja usando o Arduino ao computador, desfaça a ligação do Vcc(5V) do L298N com o Vin do Arduino, e ligue o Arduino no computador.

Agora o sistema já está atuando (se você fez o *upload* previamente do código).

1.3 Montando resistores com TIP122

Com o Arduino e a fonte desligados

Conecte o GND do Arduino no GND da fonte

Conecte o terminal do resistor no positivo fonte (caso esteja usando dois resistores, coloque um terminal do outro do resistor no positivo da fonte também, fazendo uma ligação em paralelo).

Ao apontar a parte preta do TIP122 para você, temos da esquerda para direita a seguinte pinagem: Base, Coletor, Emissor.

Conecte o pino 6 do Arduino a um terminal resistor de resistência entre 100 ohms e 1000ohm (de preferência 300 ohms). Conecte o outro terminal desse resistor à base do transistor TIP122.

Conecte os terminais que sobraram dos resistores de potência ao coletor do TIP122.

Conecte o Emissor do TIP122 ao GND.

Ligue o Arduino e a fonte

Agora o sistema já está atuando (se você fez o *upload* previamente do código).

2 Fazendo o *upload* do código

Selecione o código na pasta do *zip* de acordo com sua necessidade e abra no Arduino IDE.

| Projeto | Pasta |
|------------|--------------------|
| Peltier | PID_para_peltier |
| Resistores | PID_para_resisores |

Tabela 3: pastas para o código

Com o Arduino desconectado, vai em:

Ferramentas→ Placa→ Arduino AVR Boards→ Arduino Uno

Após isso faça:

Ferramentas→ Porta

Preste atenção nas portas, saia dessa janela.

Conecte o Arduino ao computador a faça:

Ferramentas→ Porta

Aparecerá uma nova porta, selecione ela.

Clique no símbolo de seta para direita (canto superior esquerdo), e o *upload* do código será feito.

3 Calibrando o termistor (opcional)

Com um multímetro meça o valor do resistor de 10K, e coloque no código esse valor em ohms em **RESISTOR_REF**

Com uma temperatura conhecida, coloque essa temperatura em Kelvin em **T_ZERO**

Meça a resistência do termistor nessa temperatura, a resistência em ohm será o **R_ZERO**, e coloque no código esse valor.

Coloque o termistor em outra temperatura conhecida, essa será T em Kelvin, meça a resistência com um multímetro e essa será o R.

Como temos a fórmula aproximada em:

$$R = R_0 e^{B(\frac{1}{T} - \frac{1}{T_0})}$$

$$\text{Temos: } R = R_0 e^{B(\frac{1}{T} - \frac{1}{T_0})} \rightarrow \frac{R}{R_0} = e^{B(\frac{1}{T} - \frac{1}{T_0})} \rightarrow \ln\left(\frac{R}{R_0}\right) = B\left(\frac{1}{T} - \frac{1}{T_0}\right) \rightarrow B = \frac{\ln\left(\frac{R}{R_0}\right)}{\frac{1}{T} - \frac{1}{T_0}}$$

$$B = \frac{\ln\left(\frac{R}{R_0}\right)}{\frac{1}{T} - \frac{1}{T_0}}$$

Coloque no código o valor **BETA** o valor encontrado em B

4 Lendo e salvando os valores da atuação

Abra o **codigo_para_ler.py** em algum editor e compilador de código de sua preferencia (não use *jupyter* ou *colab*, eles não interagem com o Arduino).

Caso seja necessário, abra o terminal e instale tudo, usando um por um os seguintes comandos:

pip install pyserial

pip install matplotlib

pip install numpy

pip install json

As bibliotecas *re* e *time* são padrões do *python* (eu acho), caso seja necessário instale elas também usando o mesmo tipo de comandos.

Em **arduino = serial.Serial(port='COM6', baudrate=baudrate)** troque “COM6” pela sua porta.

Em **exp_time = 1000.0** coloque seu tempo de duração do experimento.

Rode o código, e espere a conclusão do projeto, quando concluir será apresentado um gráfico do *matplotlib*, ao fechar ele outro aparecerá e ao fechar esse outro aparecerá, e ao fechar esse, aparecerá uma mensagem perguntando se deseja salvar o arquivo, “n” e “N” para não salvar e “s” e “S” para salvar, aperte enter. Caso deseje salvar aparecerá a opção de nomear seu arquivo, coloque o nome e aperte enter, seu arquivo será salvo em .json na mesma pasta de onde estava o código .py.

5 Regulando os parâmetros do PID

Dois métodos para regular os parâmetros do PID.

5.1 Regulando os três parâmetros

Inicie com K_p , K_d , K_i todos nulos.

5.1.1 Regulando o K_p

Vá aumentando apenas o K_p , o K_p ideal será alcançado quando ele quase corrigir o erro, mas não corrigirá, ficando assintoticamente perto do valor alvo.

K_p muito baixo ficará longe de corrigir e não corrigirá.

K_p muito alto, ficará subindo e descendo em torno do intervalo

Não deixe o K_p ficar muito alto ou baixo.

5.1.2 regulando o K_i

Vá aumentando o K_i , até a temperatura medida ficar oscilando em torno da temperatura ideal.

K_i ideal a temperatura medida oscilará em torno da temperatura alvo, mas a amplitude cairá cada vez mais até ficar estabilizado na temperatura alvo.

K_i muito alto, pode ter efeito semelhante a um K_p alto, e nunca estabilizar.

K_i muito baixo, pode ter um efeito assintótico, e a temperatura medida nunca chegar totalmente na temperatura alvo.

5.1.3 Regulando o K_d

Vá aumentando o K_d até os “Calombos” do K_i e K_p irem sumindo, ou seja, o sistema passa a regular a temperatura muito mais rapidamente.

K_d ideal faz com que a regulação fique mais rápida, “amortecendo” as oscilações em torno da temperatura alvo.

K_d alto, tem efeito semelhante a um K_p e K_i altos, oscilando e nunca estabilizando na temperatura alvo.

K_p baixo tem pouco impacto, mas o sistema demorará a se regular.

5.2 Método Ziegler-Nichols

Inicie com K_p , K_d , K_i todos nulos.

Vá aumentando o K_p até ter uma oscilação em torno da temperatura alvo que nunca estabiliza, esse será o K_n , meça o tempo da oscilação de um vale a outro vale, ou de um pico a outro pico (período completo), esse será o T_n

Regule os parâmetros de acordo com o tipo de sistema que você utilizará.

| Sistema | K_p | K_i | K_d |
|---------|-----------------|-------|-------|
| P | $\frac{K_n}{2}$ | 0 | 0 |

| | | | |
|-----|------------|------------------------|-----------------|
| PI | $0.45 K_n$ | $0.54 \frac{K_n}{T_n}$ | 0 |
| PID | $0.6 K_n$ | $1.2 \frac{K_n}{T_n}$ | $0.075 K_n T_n$ |

Tabela 4: Tabela de regulagem do PID

6 Agradecimentos e avisos

Muito obrigado por ler esse guia, e escolher esse código em seu experimento.

Se possível, considere incluir um agradecimento ao Henrique Assis nas apresentações.

Essa ainda é uma versão preliminar, logo os códigos podem apresentar erros falhas, caso encontre alguma, avise, para que a correção do erro seja feita. Uma nova versão deve sair dia 10/02/2025 ou 11/02/2025.