

Ponte H com MOSFET

Por Eng. Roberto Bairros dos Santos

www.bairrospd.com

Data: 28/08/2017

Sumário

Prefácio.....	3
Introdução.....	4
O MOSFET IRF540.....	6
O circuito de acionamento.....	10
A interface com o Arduino UNO.....	12
O circuito.....	14
O programa do Arduino.....	16
Conclusão.....	22
Créditos.....	23

Prefácio.

Este tutorial irá mostrar um circuito em ponte H para controlar o sentido de rotação de um motor de corrente contínua.

Este projeto usa o CI IR2102, se você não conhece este CI pode dar uma olhada no tutorial “Como Usar o driver IR2102 para acionar MOSFET” no link da figura.

http://docs.wixstatic.com/ugd/83ad1e_bd8612f2b89548e4a4d1b7748f735c45.pdf

Para simular o funcionamento do circuito foi usado o programa ISIS do Proteus!

Introdução.

Você inverte o sentido de rotação de um motor de corrente contínua invertendo a polaridade de alimentação deste motor.

Quando você usa somente uma fonte de alimentação a forma mais comum para inverter a polaridade é usando um circuito de comutação montado em ponte, conhecido como ponte H.

O circuito da ponte H é composto por quatro chaves como mostra o circuito abaixo.

As chaves ligadas a alimentação +Vcc serão chamadas de chaves do canal de alta tensão ou simplesmente “chaves da alta”.

As chaves ligadas ao sinal de terra da alimentação serão chamadas de chaves do canal de baixa tensão ou simplesmente “chaves de baixa”.

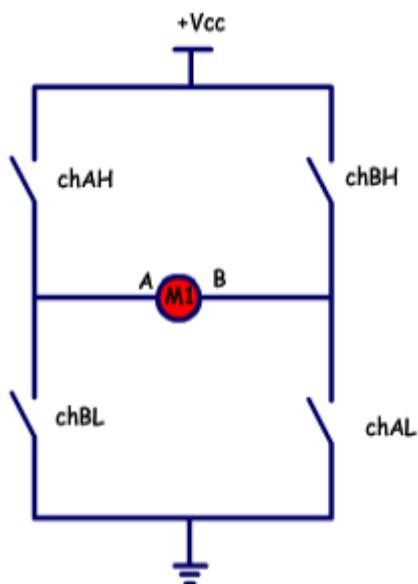
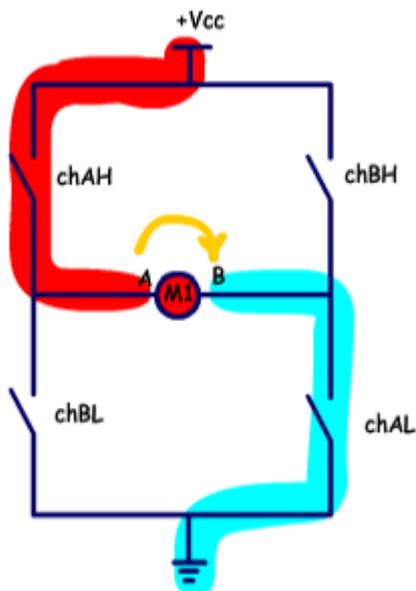
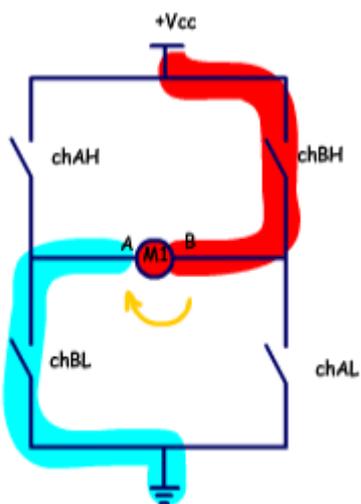


Figura Circuito ponte H.

Quando a chave “chAH” é ligada a alimentação positiva é aplicada ao lado “A” do motor, e a chave “chAL” é ligada o sinal de terra é ligado ao lado “B” do motor”. O motor irá girar em um sentido, por exemplo, sentido horário!



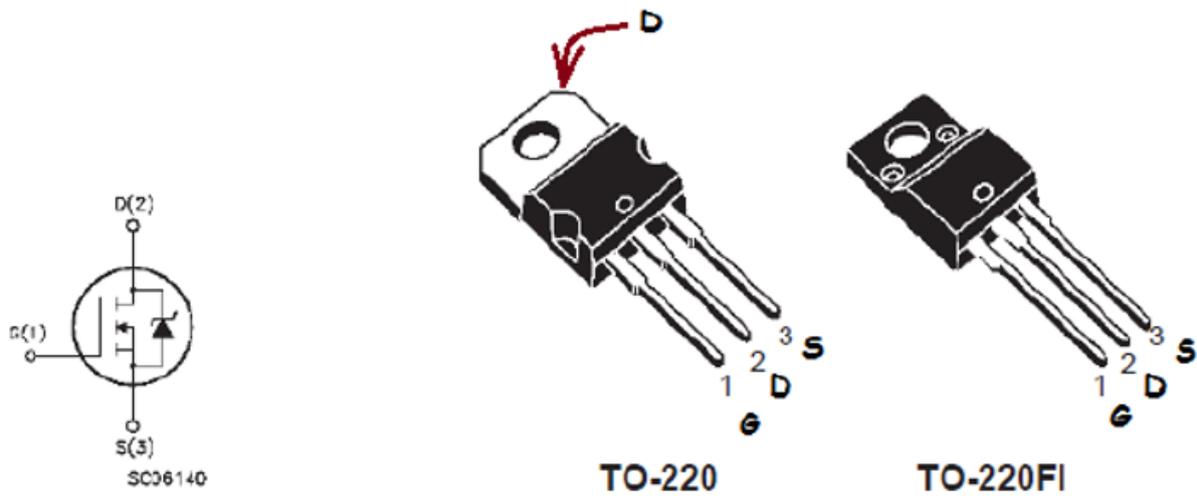
Quando a chave “chBH” é ligada a alimentação positiva é aplicada ao lado “B” do motor, e a chave “chBL” é ligada o sinal de terra é ligado ao lado “A” do motor”.



O que muda de um circuito para outro é o tipo de dispositivo usado como chave, neste trabalho será mostrado como usar um transistor IRF540 do tipo MOSFET!

O MOSFET IRF540.

Este transistor é usado para acionamentos de motores com corrente acima de 3A, sendo recomendado o uso de dissipador de calor para evitar sobre aquecimento!



IRF 540

As principais características do IRF540 são mostradas na tabela.

Nesta tabela você pode observar que a tensão de trabalho máxima é de 100V, e a corrente de trabalho máxima é de 30A. A máxima tensão no circuito de comutação V_{gs} é de 20V!

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value		Unit
		IRF 540	IRF540 FI	
V_{DS}	Drain-source Voltage ($V_{GS} = 0$)	100		V
V_{DGR}	Drain- gate Voltage ($R_{GS} = 20\text{ k}\Omega$)	100		V
V_{GS}	Gate-source Voltage	± 20		V
I_D	Drain Current (continuous) at $T_c = 25^\circ\text{C}$	30	17	A
I_D	Drain Current (continuous) at $T_c = 100^\circ\text{C}$	21	12	A
$I_{DM}(\bullet)$	Drain Current (pulsed)	120	120	A
P_{tot}	Total Dissipation at $T_c = 25^\circ\text{C}$	150	45	W
	Derating Factor	1	0.3	W/°C
Viso	Insulation Withstand Voltage (DC)	-	2000	V
T_{stg}	Storage Temperature	-65 to 175		°C
T_j	Max. Operating Junction Temperature	175		°C

(•) Pulse width limited by safe operating area

(1) $I_{SD} \leq 30\text{ A}$, $di/dt \leq 200\text{ A}/\mu\text{s}$, $V_{DD} \leq V_{(BR)DSS}$, $T_j \leq T_{jMAX}$

Nesta outra tabela você observa que a tensão V_{gs} para ligar o transistor deve ser maior do que 3V, no circuito com o CI IR2102 esta tensão é garantida pelo CI!

ELECTRICAL CHARACTERISTICS ($T_{case} = 25\text{ }^{\circ}\text{C}$ unless otherwise specified)

OFF

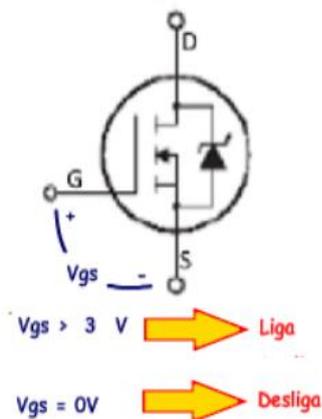
Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
$V_{(BR)DSS}$	Drain-source Breakdown Voltage	$I_D = 250\text{ }\mu\text{A}$ $V_{GS} = 0$	100			V
I_{DSS}	Zero Gate Voltage Drain Current ($V_{GS} = 0$)	$V_{DS} = \text{Max Rating}$ $V_{DS} = \text{Max Rating}$ $T_c = 125\text{ }^{\circ}\text{C}$			1 10	μA μA
I_{GSS}	Gate-body Leakage Current ($V_{DS} = 0$)	$V_{GS} = \pm 20\text{ V}$			± 100	nA

ON (*)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
$V_{GS(th)}$	Gate Threshold Voltage	$V_{DS} = V_{GS}$ $I_D = 250\text{ }\mu\text{A}$	2	3	4	V
$R_{DS(on)}$	Static Drain-source On Resistance	$V_{GS} = 10\text{ V}$ $I_D = 15\text{ A}$		0.05	0.077	Ω
$I_{D(on)}$	On State Drain Current	$V_{DS} > I_{D(on)} \times R_{DS(on)max}$ $V_{GS} = 10\text{ V}$	30			A

Para ligar o MOSFET o circuito deve aplicar entre os pinos de gate e source uma tensão maior do que 3V, esta tensão é chamada de V_{gs} .

Para desligar o MOSFET a tensão de V_{gs} deve ser igual a zero Volt!

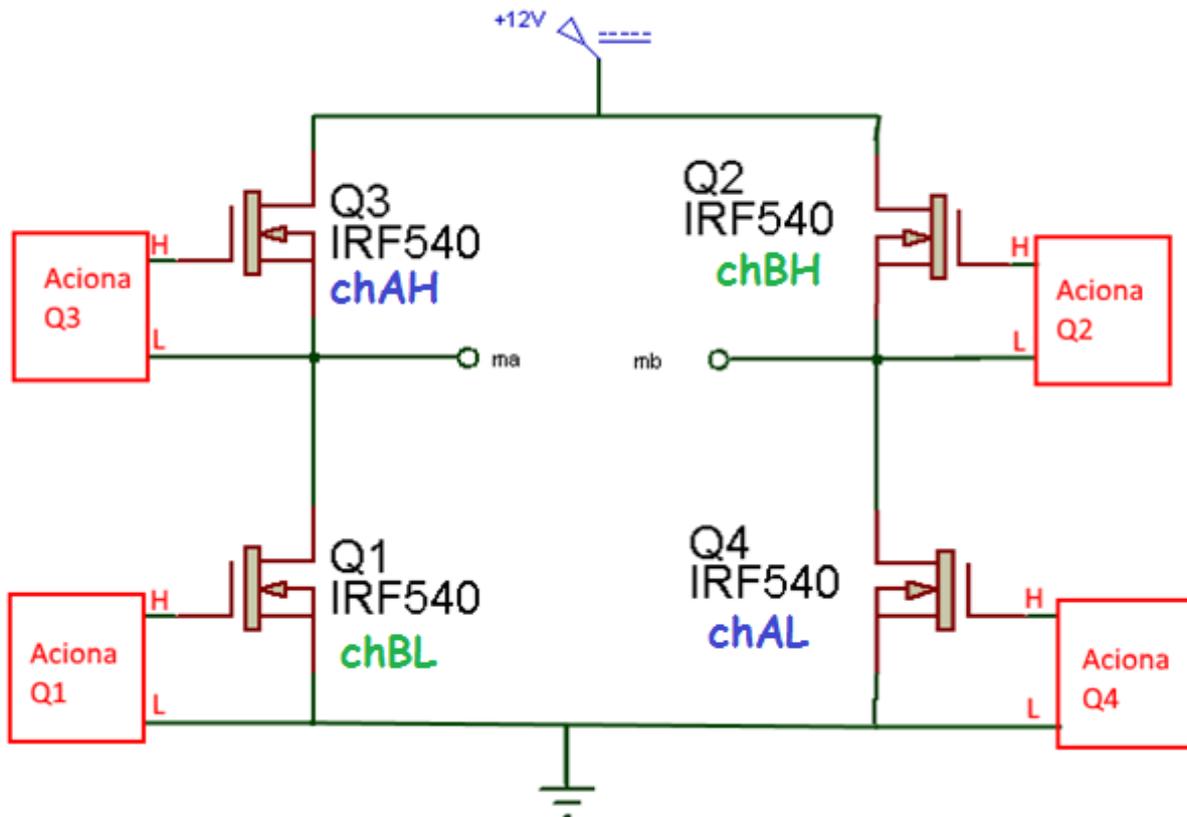


O circuito da figura mostra os MOSFETs no lugar das chaves.

Cada MOSFET precisa um circuito de acionamento porque a entrada “L” (LOW) do circuito dos MOSFETs de alta não são aterrados.

Neste circuito os MOSFETs foram alimentados com uma tensão de 12V porque neste exemplo está sendo usado um motor de 12V, assim Vcc pode ser uma bateria!

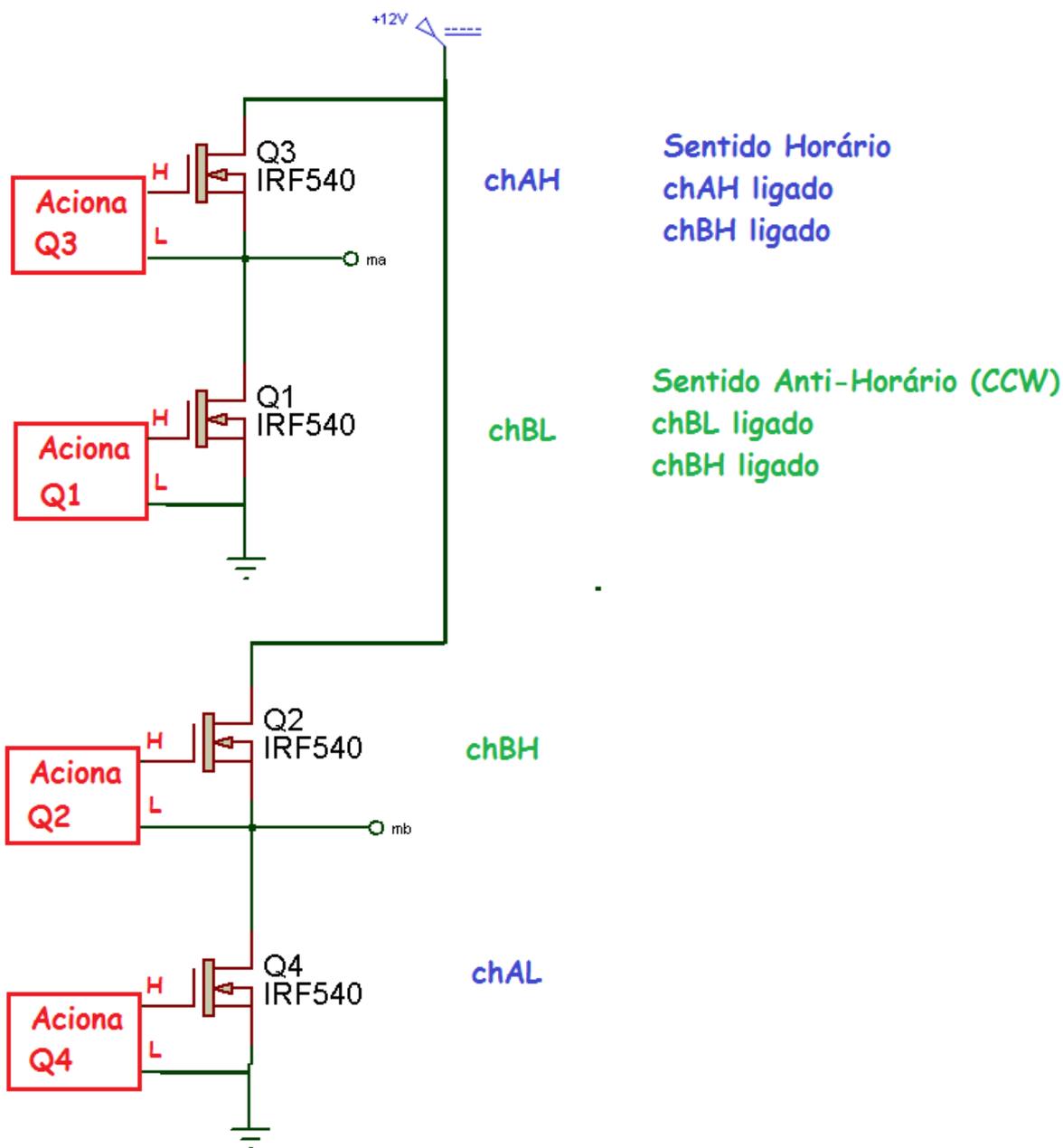
Entre os terminais “ma” e “mb” é ligado o motor!



Para tornar a ligação mais legível, o circuito foi desenhado um pouco diferente, a ponte foi retorcida, como mostra a figura!

Para ligar o motor no sentido horário os transistores Q3 e Q4 devem ser ligados.

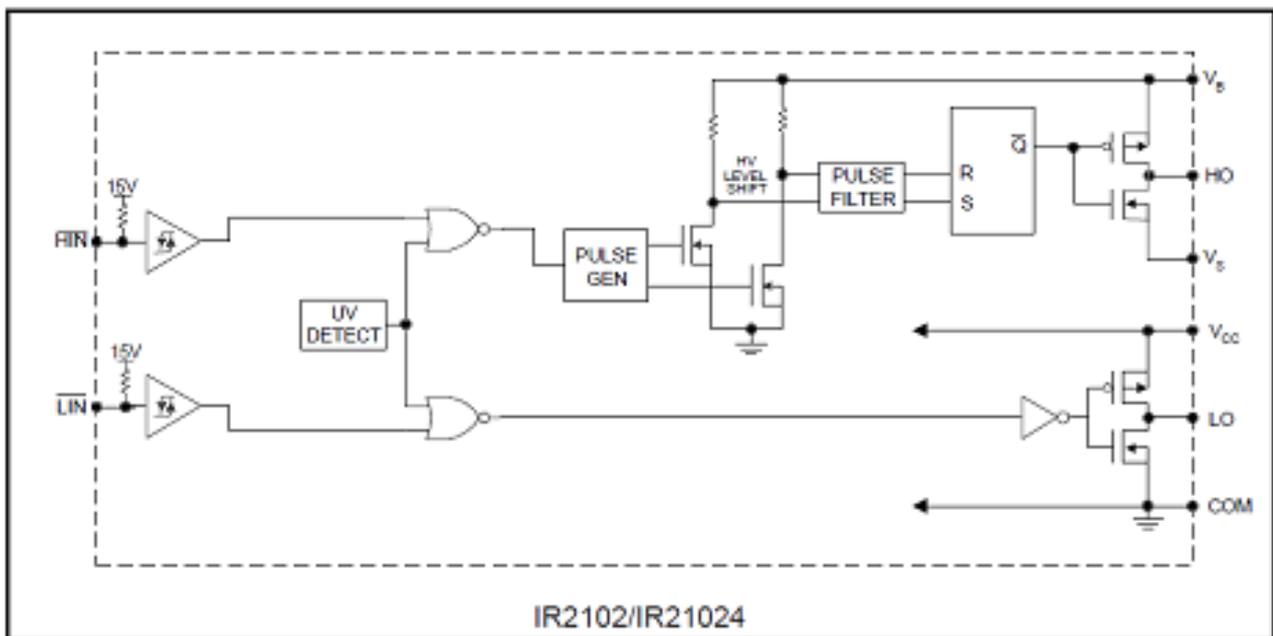
Para ligar o motor no sentido anti-horário os transistores Q1 e Q2 devem ser ligados.



O circuito de acionamento.

O problema do circuito de acionamento é que o pino “L” dos MOSFETs de alta não está ligado ao sinal de terra, assim o circuito de alimentação do CI deve ser isolado do circuito dos MOSFETs de baixa ligados ao sinal de terra. Como fazer isto?

Usando o CI IR2102!



Cada CI aciona um par de MOSFETs, assim você vai precisar de dois Cis!

A alimentação pode ser de até 15Vcc, sendo o positivo ligado no pino Vcc e o negativo ao pino COM!

No simulador o pino de Vcc é chamado de V+.

Os dois pinos de saídas para o acionamento do MOSFET de alta são os pinos HO e Vs.

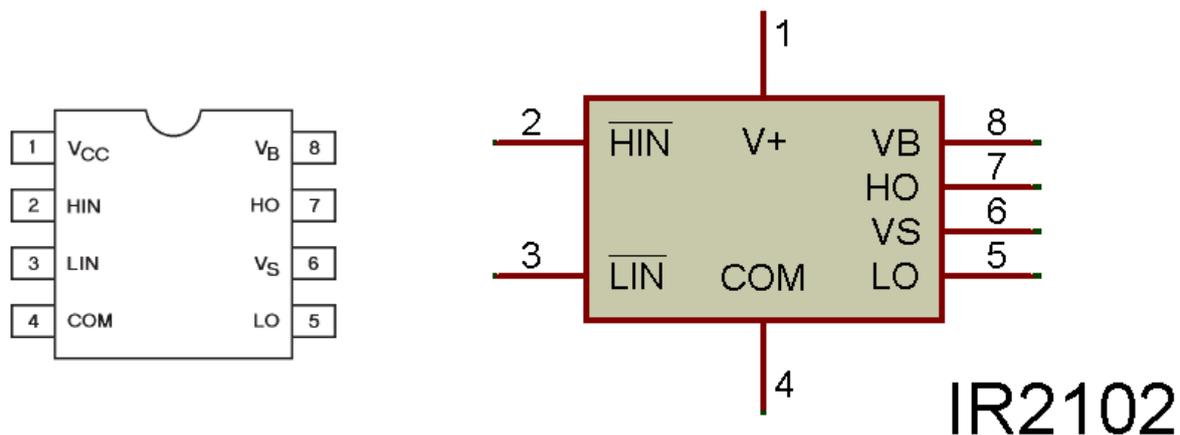
Os dois pinos de saídas para o acionamento do MOSFET de baixa são os pinos LO e COM.

Para ligar o MOSFET de alta o circuito externo deverá aterrar a entrada HIN.

Para ligar o MOSFET de baixa o circuito externo deverá aterrar a entrada LIN.

Notar que cada entrada já possuem um resistor de 15kOhm interno (pullup) ligado ao Vcc.

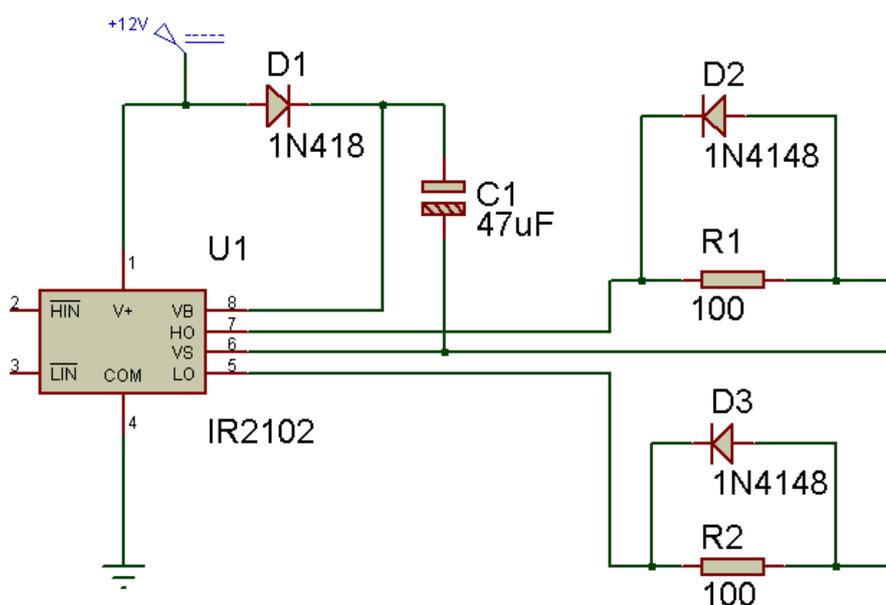
A pinagem do CI é mostrada na figura.



O diodo D1 e o capacitor C1 são usados para gerar a tensão que alimenta o MOSFET de alta.

Os resistores R1 e R2 não devem ser ignorados e servem para garantir que o capacitor consiga ligar e manter ligado o MOSFET pelo tempo desejado.

Os diodos D2 e D3 garantem um desligamento rápido do MOSFET!

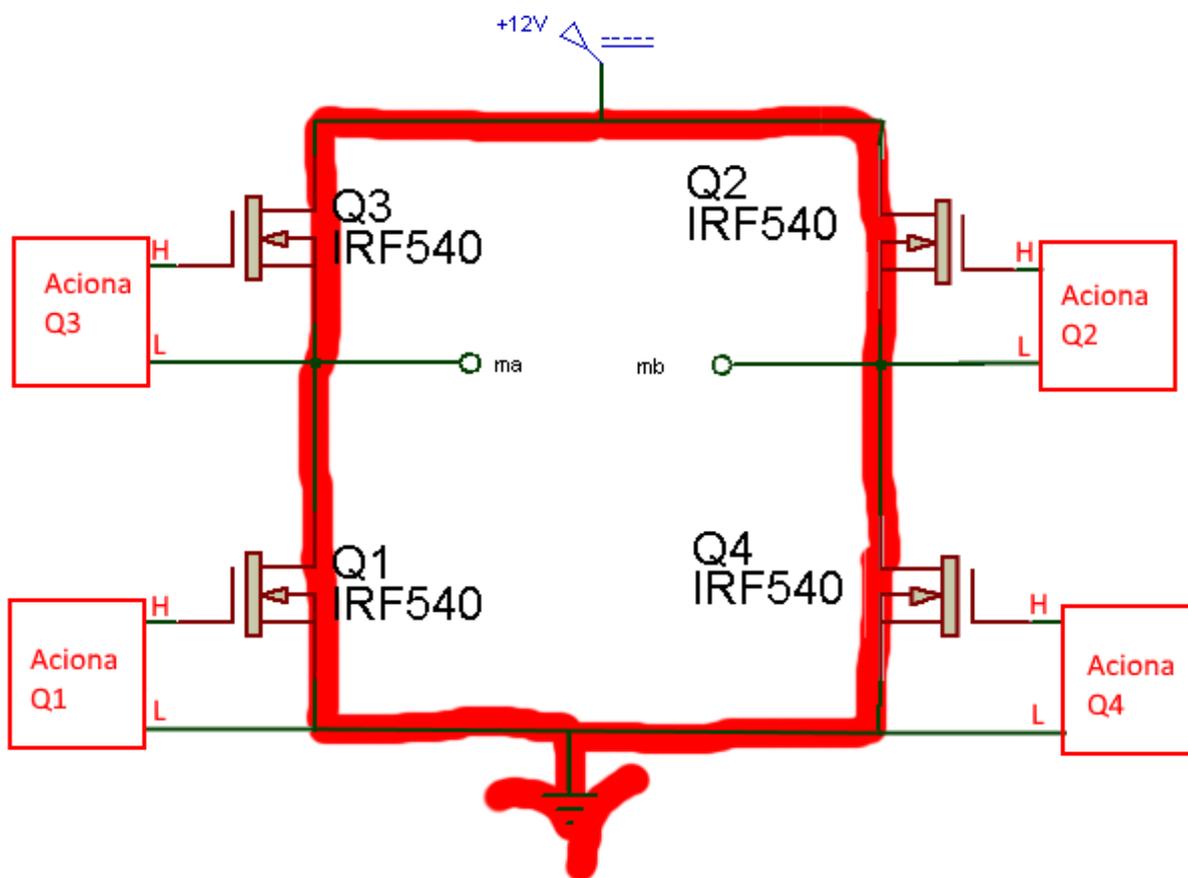


O circuito de acionamento deve garantir que o sinal que Liga a chave A e o sinal que liga a chave B não sejam acionados ao mesmo tempo, pois neste caso, ocorrerá um curto circuito.

Você deve estar prevenido que muitos transistores poderão ser queimados durante os testes, qualquer descuido é fatal!

Um fusível foi colocado em série com a alimentação dos transistores por segurança.

A figura mostra o caminho do curto circuito!

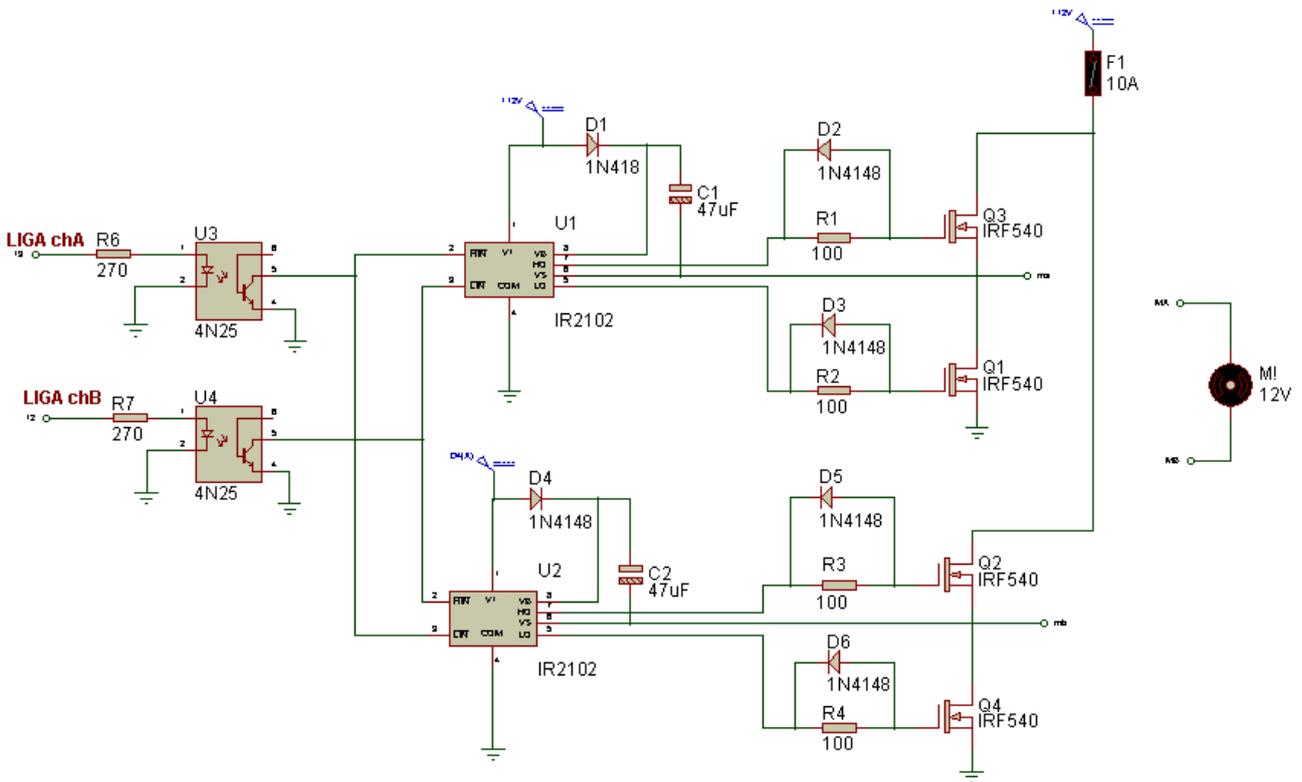


Ponte H com MOSFET

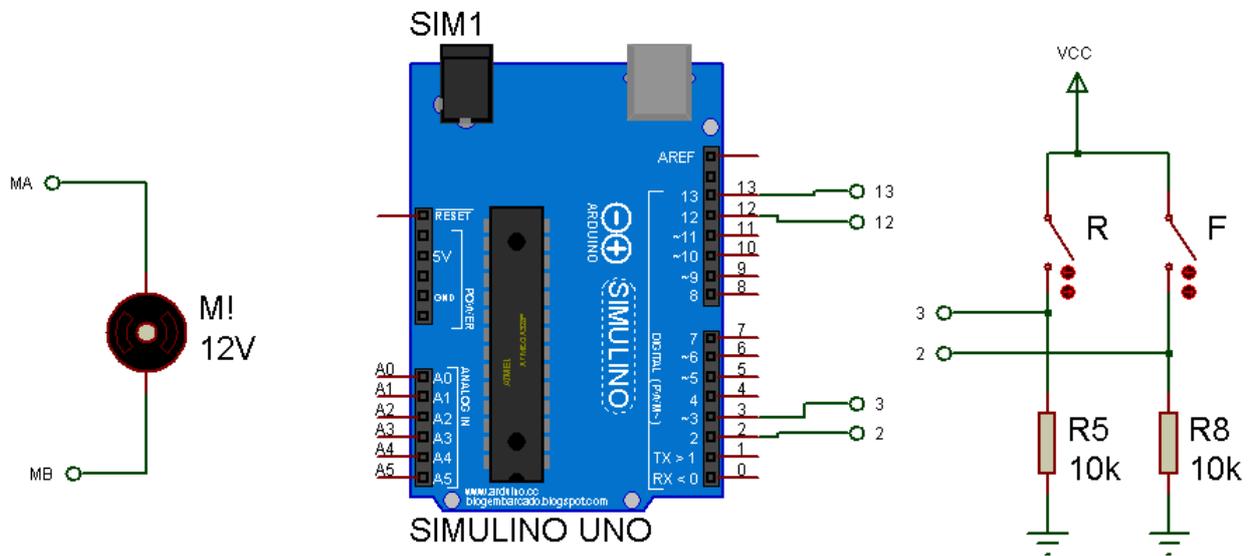
O circuito.

O circuito completo da potência da chave H é mostrado abaixo.

Para organizar crie uma pasta para o seu projeto com o nome que você achar mais interessante, por exemplo, "ponte h". Abra o programa do simulador ISIS do Proteus e monte o circuito desenhado abaixo. Notar que o desenho foi separado em dois para deixar o circuito mais claro na figura. O circuito do Arduino deverá ser desenhado do lado do motor de forma a permitir que você enxergue as chaves e o motor na hora de simular.



O circuito do Arduino com as duas chaves de liga o motor para frente e liga motor para trás é mostrado abaixo.



O programa do Arduino.

Você deverá criar um sketch novo no programa do Arduino e salvar na mesma pasta criada para guardar o desenho do diagrama, com o nome que você preferir, por exemplo “ponte h”.

Leia os detalhes do programa descrito abaixo para entender como funciona, depois copie e cole o programa sobre todo o sketch novo do Arduino.

Depois de montado o programa gere o arquivo binário usando o menu “Sketch” clicando na opção “Exportar Arquivo Binário”.

Para simular carregue o arquivo do programa no Arduino Uno do simulador clicando na placa UNO e selecionando a propriedade “Program File” procure e selecione o programa ponto hex compilado antes!

Rode o programa e clique nas chaves para ver o motor rodar e inverter a rotação!

Veja os detalhes do programa.

No início são definidas as variáveis que dão nomes aos pinos de saída.

```
int motorafrente=13;  
int motoratras=12;
```

No início também são definidas duas variáveis auxiliares do tipo booleanas (ligado=High e desligado=LOW), estas variáveis servirão como flags indicando ao programa o momento certo de ligar os motores.

```
char motorafrenteligado;//booleana  
char motoratrasligado;//booleana
```

No setup são configurados os pinos 12 e 13 estes pinos devem ser configurados como saídas, e serão ligados ao circuito de entrada da ponte H.

Os pinos 2 e 3 devem ser configurados como entradas, nestes pinos serão ligadas as chaves de direção do movimento, neste exemplo o motor foi usado para movimentar um robô para frente e para trás.

```
void setup() {  
  // put your setup code here, to run once:  
  pinMode(13,OUTPUT);  
  pinMode(12,OUTPUT);  
  pinMode(2,INPUT);  
  pinMode(3,INPUT);  
}
```

O programa de controle é colocado na função loop().

As primeiras instruções são usadas para ler o estado das chaves.

```
int chafrente=digitalRead(2); // pegar o valor das chave anda para frente
int chatras=digitalRead(3); // pegar o valor das chave anda para frente
```

A lógica para ligar o motor para frente e para trás é a mesma.

Será detalhado agora somente a lógica usada para ligar o motor para frente.

Para testar a condição das chaves é usada uma função if() que faz a seguinte pergunta.

Se a chave “liga para frente” está ligada “E” a chave “liga para trás” NÃO está ligada, a rotina dentro dos colchetes após o if() será processada para ligar o motor.

Observar que a função if() é verdadeira se o resultado da pergunta não é zero, por isto, não precisa escrever “chafrente== HIGH”!

O ponto de interrogação na frente da variável indica a função lógica “NÃO” que pode ser descrita como “chave NÃO ligada”!

```
if (chafrente && !chatras){
```

A rotina dentro do colchete após a função if() é processada quando a pergunta dentro dos parênteses tiver uma resposta afirmativa.

Dentro do colchete da função if() existe outra função if() que pergunta se o flag que indica que o motor já pode ser ligado está com o valor HIGH (ligado).

No início este flag estará com o valor LOW (desligado) e o resultado deste if() levará o programa para os colchetes após a instrução else{}

A rotina dentro do colchete da instrução else{} primeiro desliga os dois sentidos de acionamento do motor, depois o programa gera um atraso de 100ms para garantir que o motor pare realmente, e então, liga o flag que indica que o motor pode ser ligado.

No final da rotina o flag que indica que o motor já pode ser ligado é colocado no estado HIGH (ligado).

Na próxima vez que o programa passar pelo if() o flag estará ligado e o motor será ligado.

Esta sequência é necessária para evitar que os dois sentidos sejam acionados ao mesmo tempo o que acarretaria um curto circuito nos MOSFETs.

A função loop() é mostrada na figura.

```
// Se chafrente ligada E chatras não ligada então liga motor a frente
if (chafrente && !chatras){
  if (motorafrenteligado){ //espera o flag com delay para ligar motor
    digitalWrite(motorafrente, HIGH);
  }
  else{
    digitalWrite(motorafrente, LOW); //desliga tudo por segurança
    digitalWrite(motoratras, LOW);
    delay(100); //atraso para desligar motor
    motorafrenteligado=HIGH; //liga flag motor ligado
  }
}
else{
  digitalWrite(motorafrente, LOW);
  motorafrenteligado=LOW; //desliga flag motor ligado
}
```

O programa completo é mostrado a seguir e logo depois você verá o programa para copiar e colar.

```
1 // ponte H com movimento para frente e para trás
2
3 int motorafrente=13;
4 int motoratras=12;
5 char motorafrenteligado;//booleana
6 char motoratrasligado;//booleana
7
8 void setup() {
9   // put your setup code here, to run once:
10  pinMode(13,OUTPUT);
11  pinMode(12,OUTPUT);
12  pinMode(2,INPUT);
13  pinMode(3,INPUT);
14 }
15
16 void loop() {
17   int chafrente=digitalRead(2);// pegar o valor das chave anda para frente
18   int chatras=digitalRead(3);// pegar o valor das chave anda para frente
19
20   // Se chafrente ligada E chatras não ligada então liga motor a frente
21   if (chafrente && !chatras){
22     if (motorafrenteligado){ //espera o flag com delay para ligar motor
23       digitalWrite(motorafrente, HIGH);
24     }
25     else{
26       digitalWrite(motorafrente, LOW); //desliga tudo por segurança
27       digitalWrite(motoratras, LOW);
28       delay(100);//atraso para desligar motor
29       motorafrenteligado=HIGH;//liga flag motor ligado
30     }
31   }
32   else{
33     digitalWrite(motorafrente, LOW);
34     motorafrenteligado=LOW;//desliga flag motor ligado
35   }
36
37   // Se chatras ligada E chafrente não ligada então liga motor a frente
38   if (chatras && !chafrente){
39     if (motoratrasligado){ //espera o flag com delay para ligar motor
40       digitalWrite(motoratras, HIGH);
41     }
42     else{
43       digitalWrite(motorafrente, LOW); //desliga tudo por segurança
44       digitalWrite(motoratras, LOW);
45       delay(100);//atraso para ligar motor
46       motoratrasligado=HIGH;//liga flag motor ligado
47     }
48   }
49   else{
50     digitalWrite(motoratras, LOW);
51     motoratrasligado=LOW;//desliga flag motor ligado
52   }
53 }
```

O programa para copiar e colar é mostrado a seguir.

```
// ponte H com movimento para frente e para trás
int motorafrente=13;
int motoratras=12;
char motorafrenteligado;//booleana
char motoratrasligado;//booleana

void setup() {
  // put your setup code here, to run once:
  pinMode(13,OUTPUT);
  pinMode(12,OUTPUT);
  pinMode(2,INPUT);
  pinMode(3,INPUT);
}

void loop() {
  int chafrente=digitalRead(2);// pegar o valor das chave anda para frente
  int chatras=digitalRead(3);// pegar o valor das chave anda para frente

  // Se chafrente ligada E chatras não ligada então liga motor a frente
  if (chafrente && !chatras){
    if (motorafrenteligado){ //espera o flag com delay para ligar motor
      digitalWrite(motorafrente, HIGH);
    }
    else{
      digitalWrite(motorafrente, LOW); //desliga tudo por segurança
      digitalWrite(motoratras, LOW);
      delay(100);//atraso para desligar motor
      motorafrenteligado=HIGH;//liga flag motor ligado
    }
  }
  else{
    digitalWrite(motorafrente, LOW);
    motorafrenteligado=LOW;//desliga flag motor ligado
  }

  // Se chatras ligada E chafrente não ligada então liga motor a frente
  if (chatras && !chafrente){
    if (motoratrasligado){ //espera o flag com delay para ligar motor
      digitalWrite(motoratras, HIGH);
    }
    else{
      digitalWrite(motorafrente, LOW); //desliga tudo por segurança
      digitalWrite(motoratras, LOW);
      delay(100);//atraso para ligar motor
      motoratrasligado=HIGH;//liga flag motor ligado
    }
  }
  else{
    digitalWrite(motoratras, LOW);
    motoratrasligado=LOW;//desliga flag motor ligado
  }
}
```

Para testar o programa no simulador vá no menu Sketch e clique na opção “Exportar Binário Compilado” para gerar o arquivo compilado ponto hex.

Vá no simulador ISIS do Proteus clique na propriedade do Arduino UNO e na opção “Program Files” selecione o arquivo ponto hex compilado antes.

Conclusão.

Você viu neste tutorial como montar um circuito para inversão da rotação de um motor de corrente contínua usando MOSFET.

O projeto inclui um programa para Arduino usando a placa UNO, mas você poderia usar qualquer outro tipo de controle.

Créditos.

Sites: www.bairrospd.com

SEO: www.bairrospd.com, professor bairros, eletrônica, tutorial, Ponte H, Acionamento motor corrente contínua.