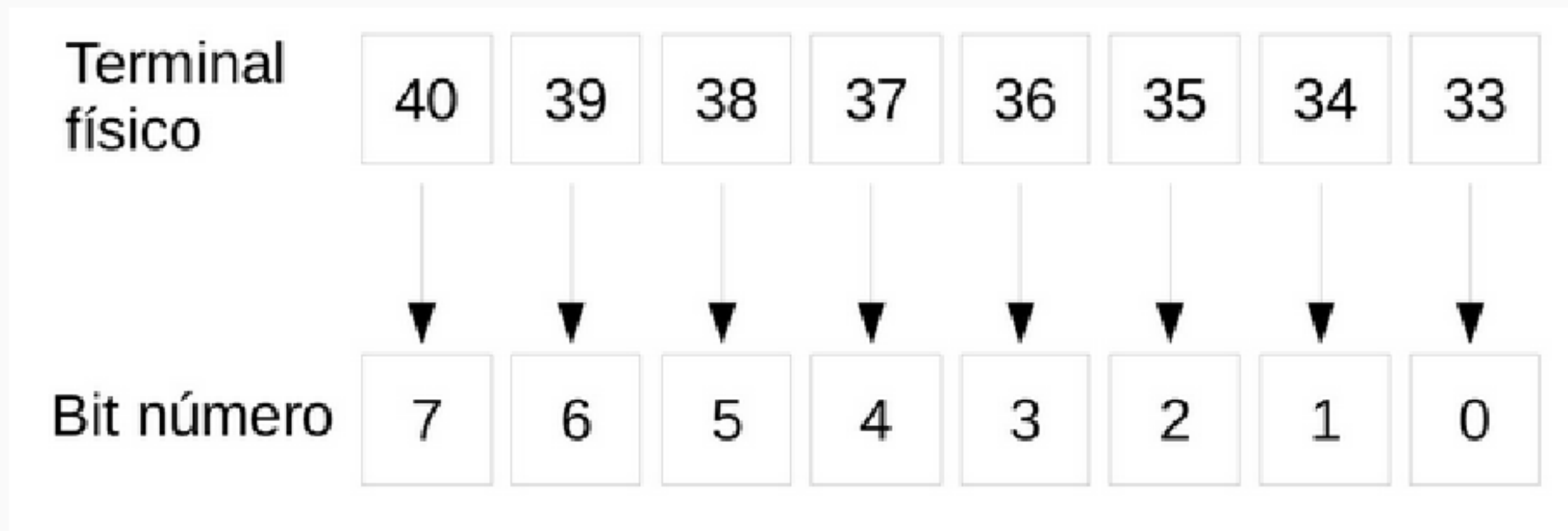


Microcontrolador

Sinais -> analógico e digital

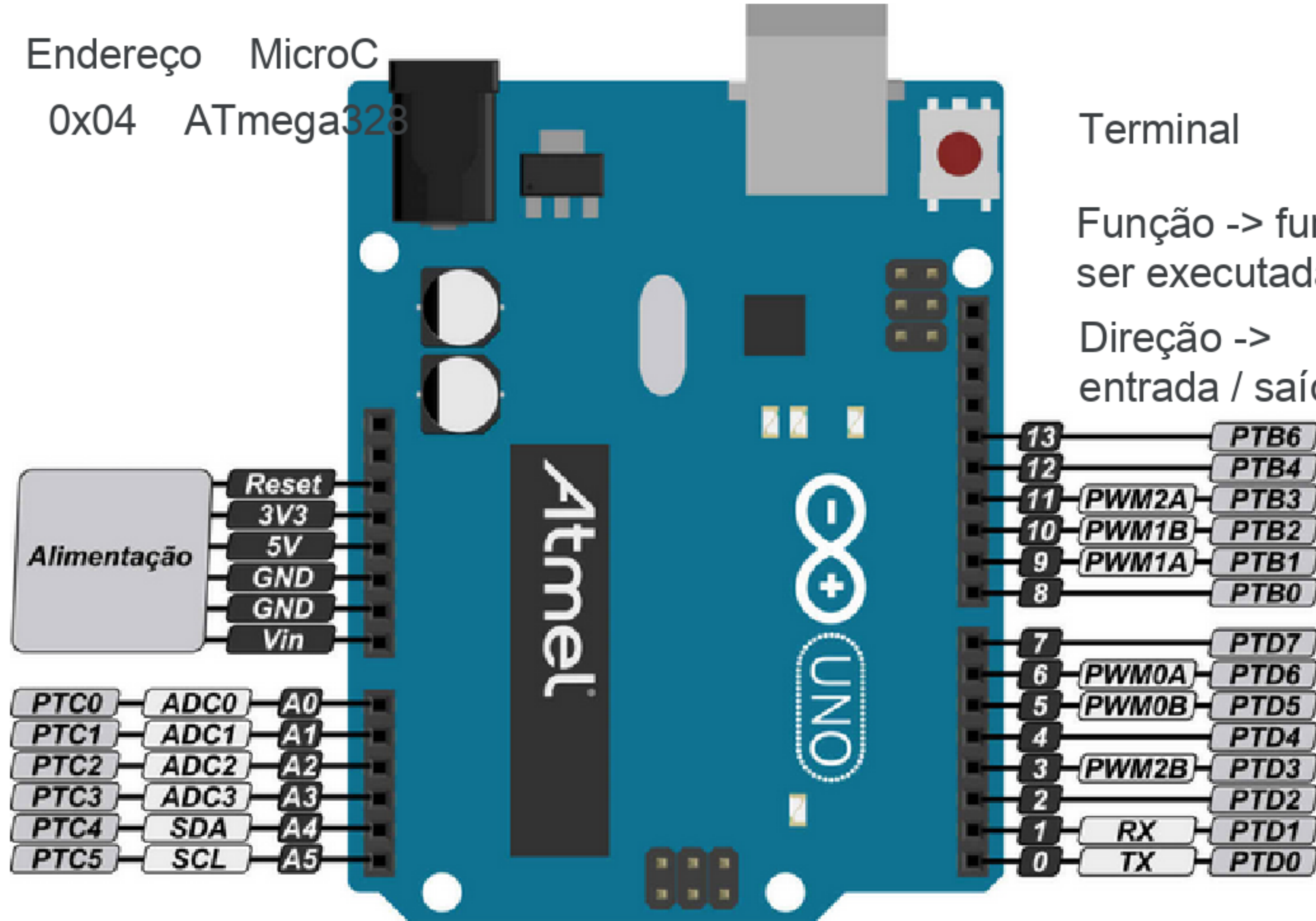
Analógico -> digital -> realizar contagem de tempo cpu, enviar mensagem nas portas seriais -> analógico



Registro
DDRB

Endereço
0x04

MicroC
ATmega328

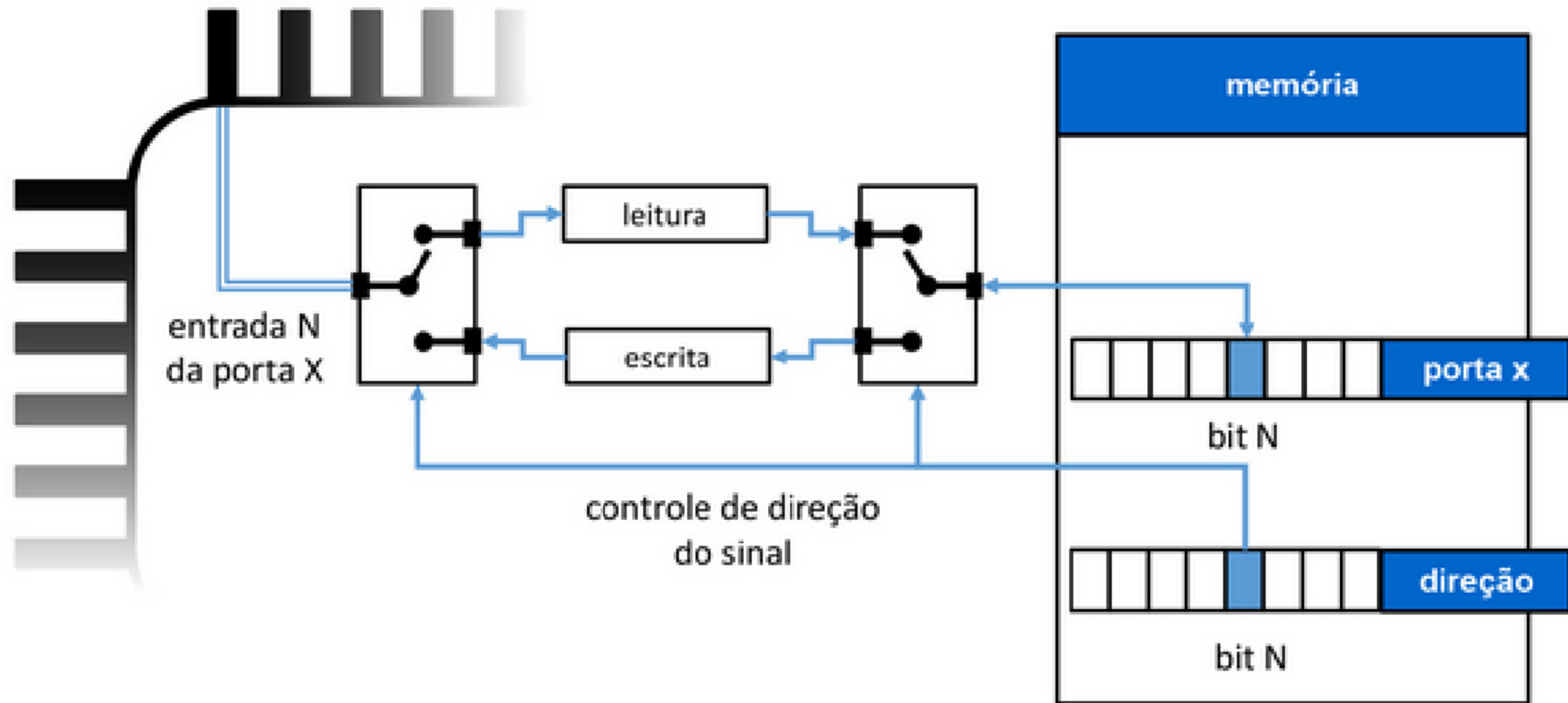


Terminal

Função -> função a
ser executada

Direção ->
entrada / saída

Processo de leitura e escrita de terminal a partir de um bit na memória



Configuração dos periféricos

`pinMode(pino, modo)`

`pinMode(13, OUTPUT)`

`pinMode(8, INPUT)`

Saídas Digitais

Leds light emitting diode

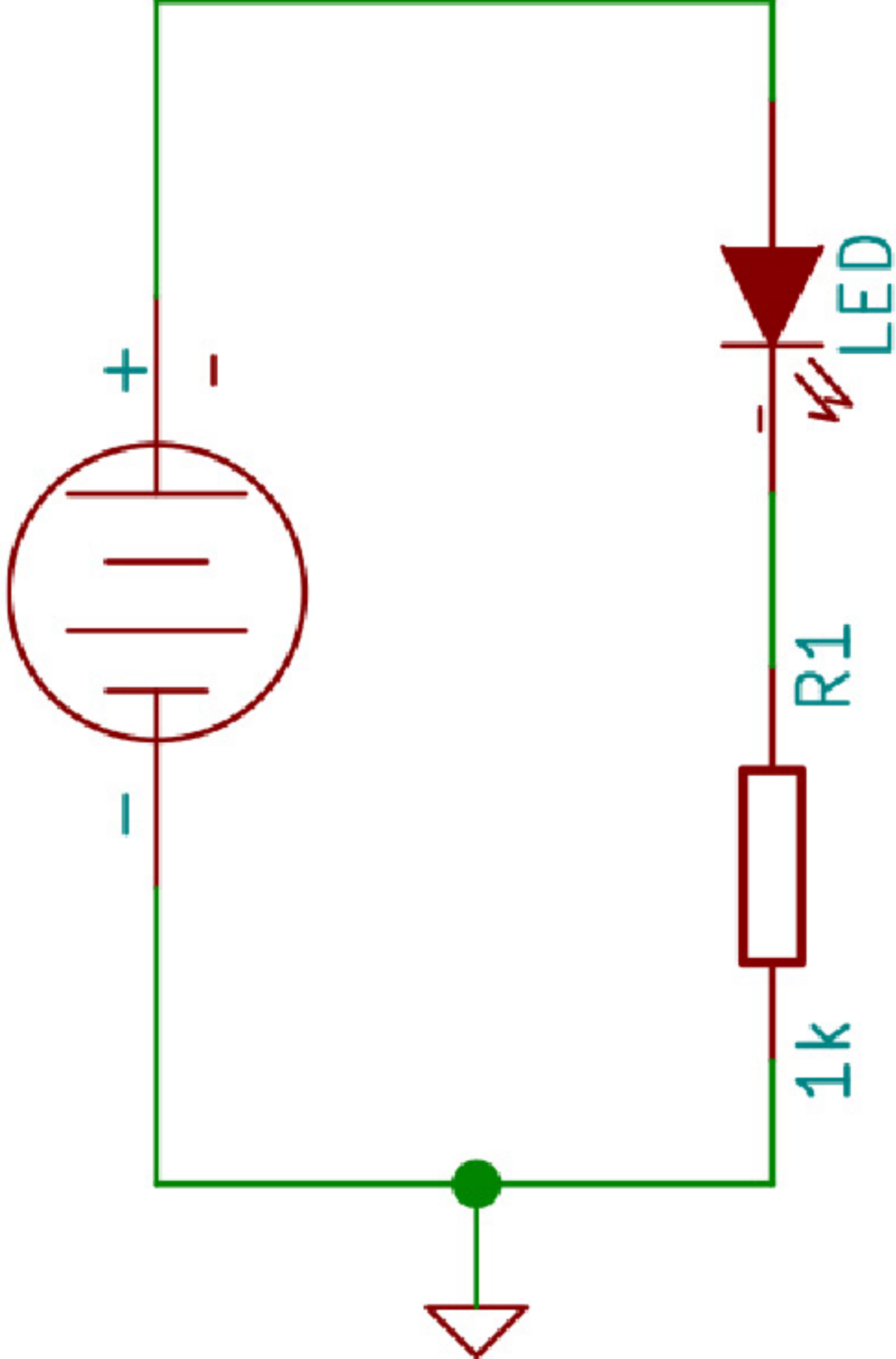
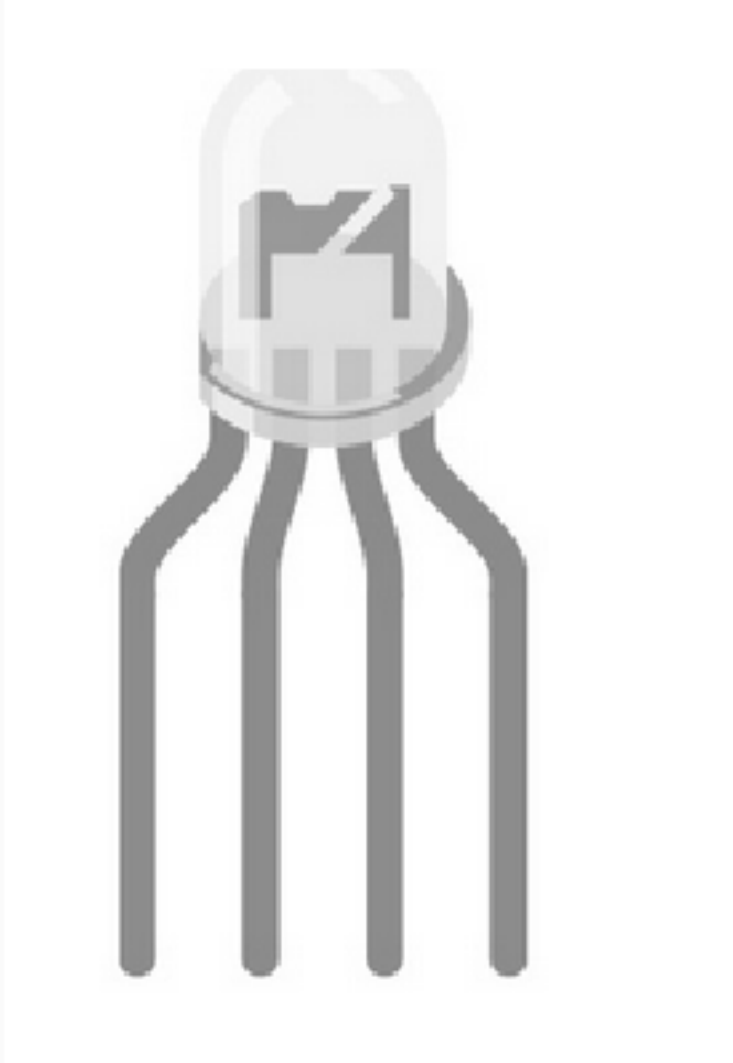
Transistor

Relé

Relé de estado sólido

Ponte H

Led



Transistor

Trabalha em corrente contínua

Amplificador

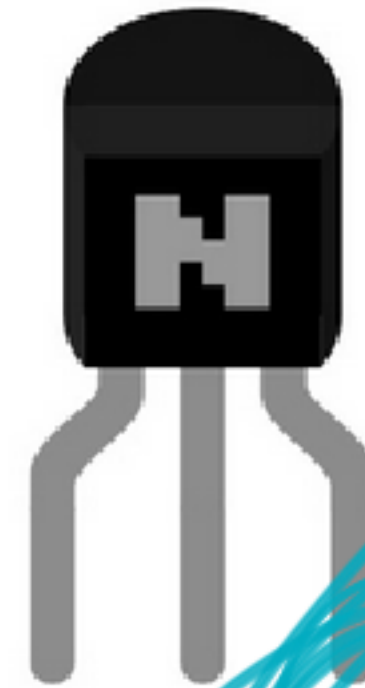
{ amplia o nível do sinal

Chave

{ usar um sinal pequeno de entrada para ligar cargas maiores de saída

o terminal tem que estar ligado ao terra

deve ser fornecido tensão maior que 1 volt



(a) To-18



(b) To-202

Relé dispositivos eletromecânicos

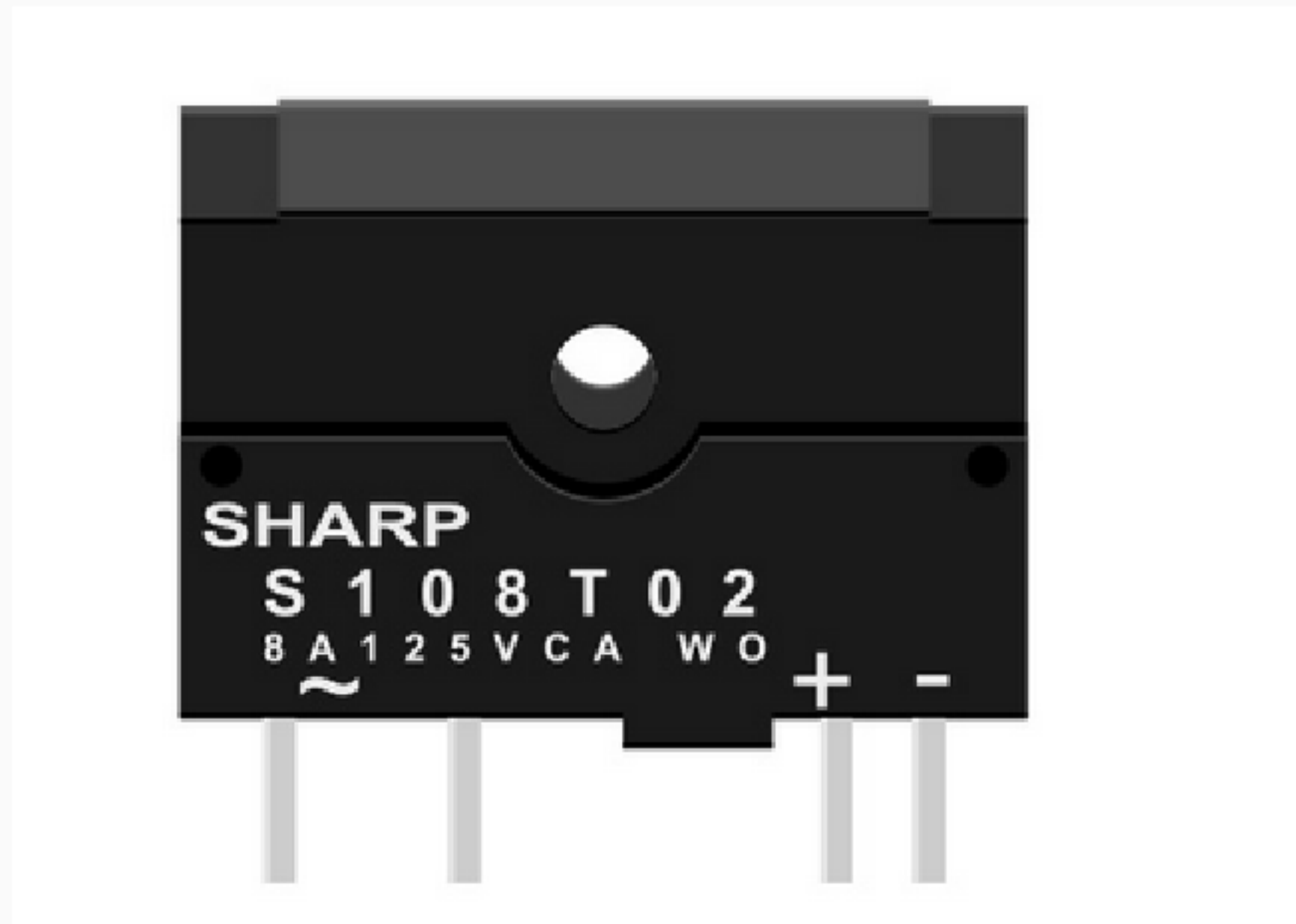
Corrente alternada

Corrente muito alta

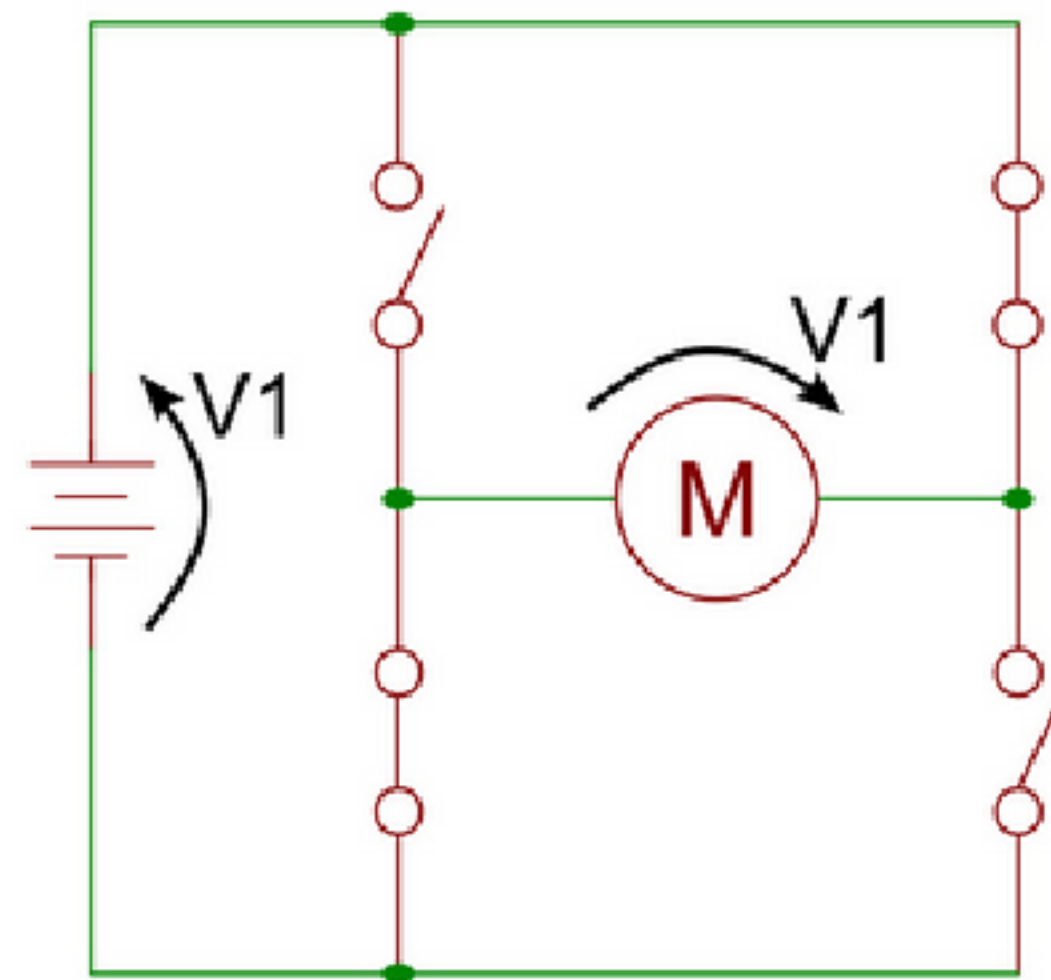
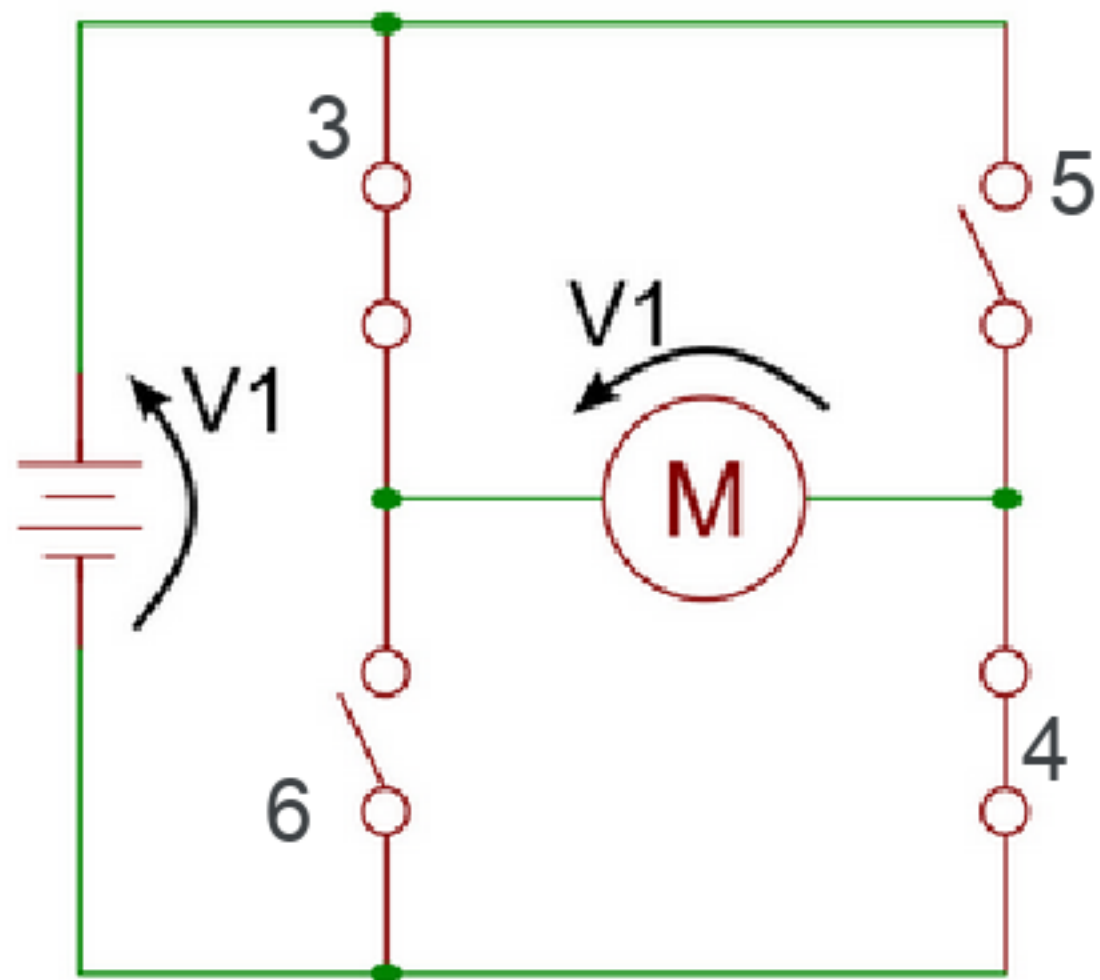


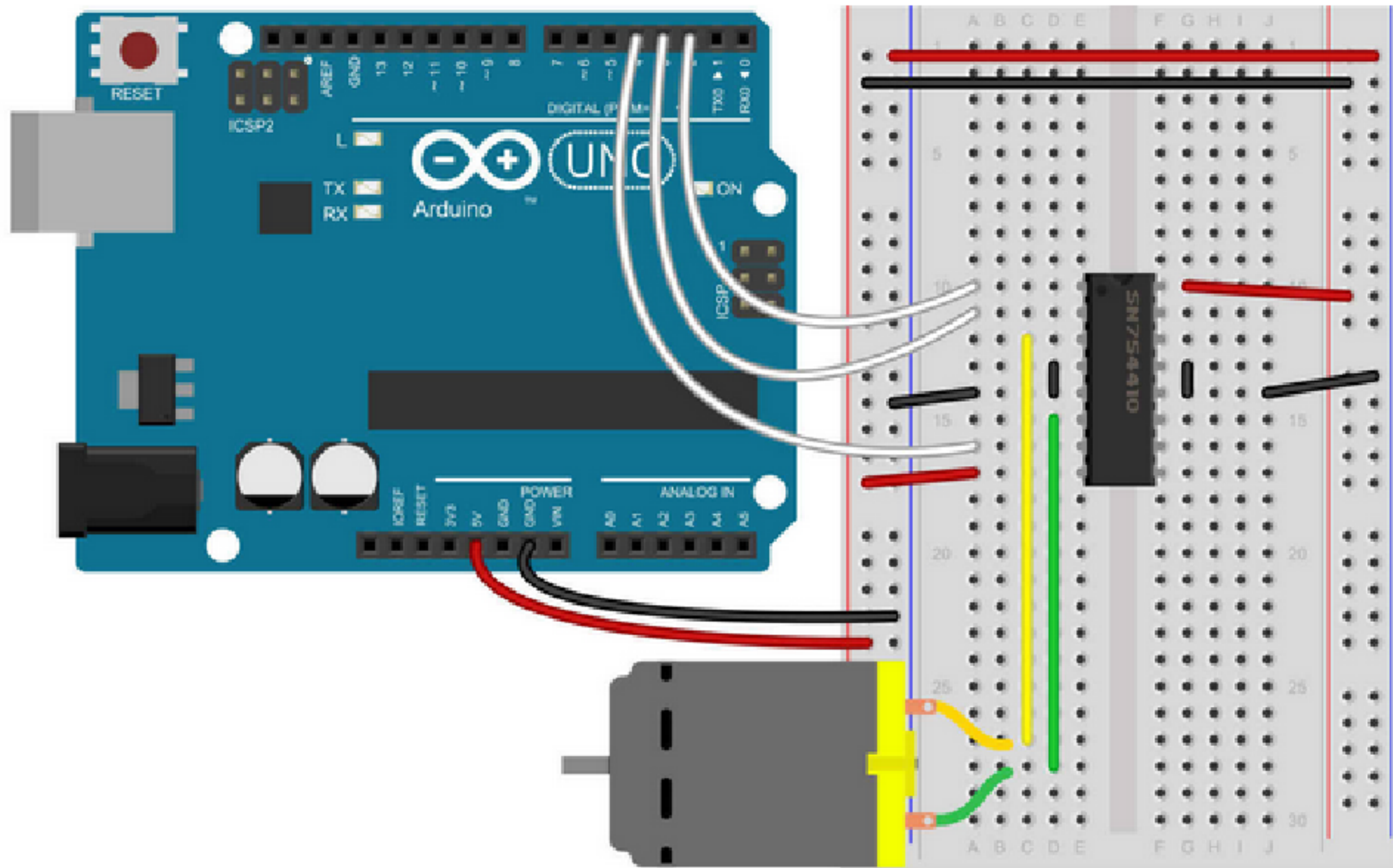
ATI assistência técnica improvisada => gambiarra

Relé de estado sólido



Ponte H





Expansão as saídas

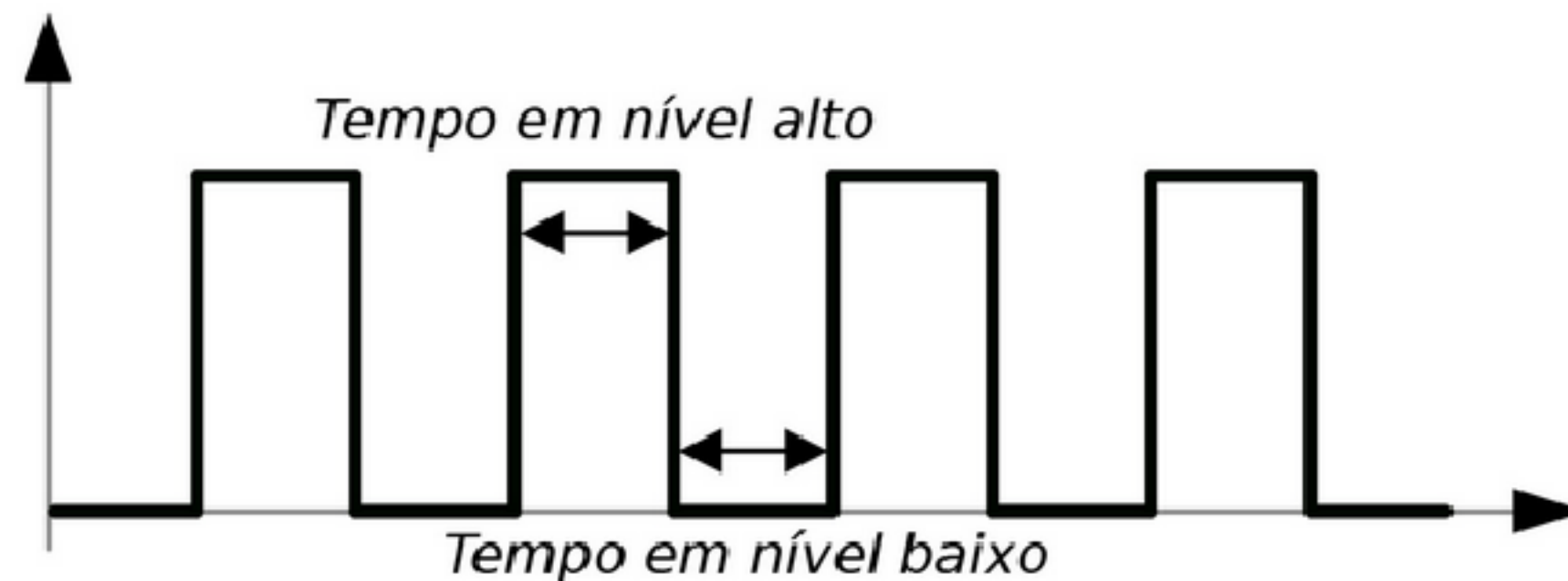
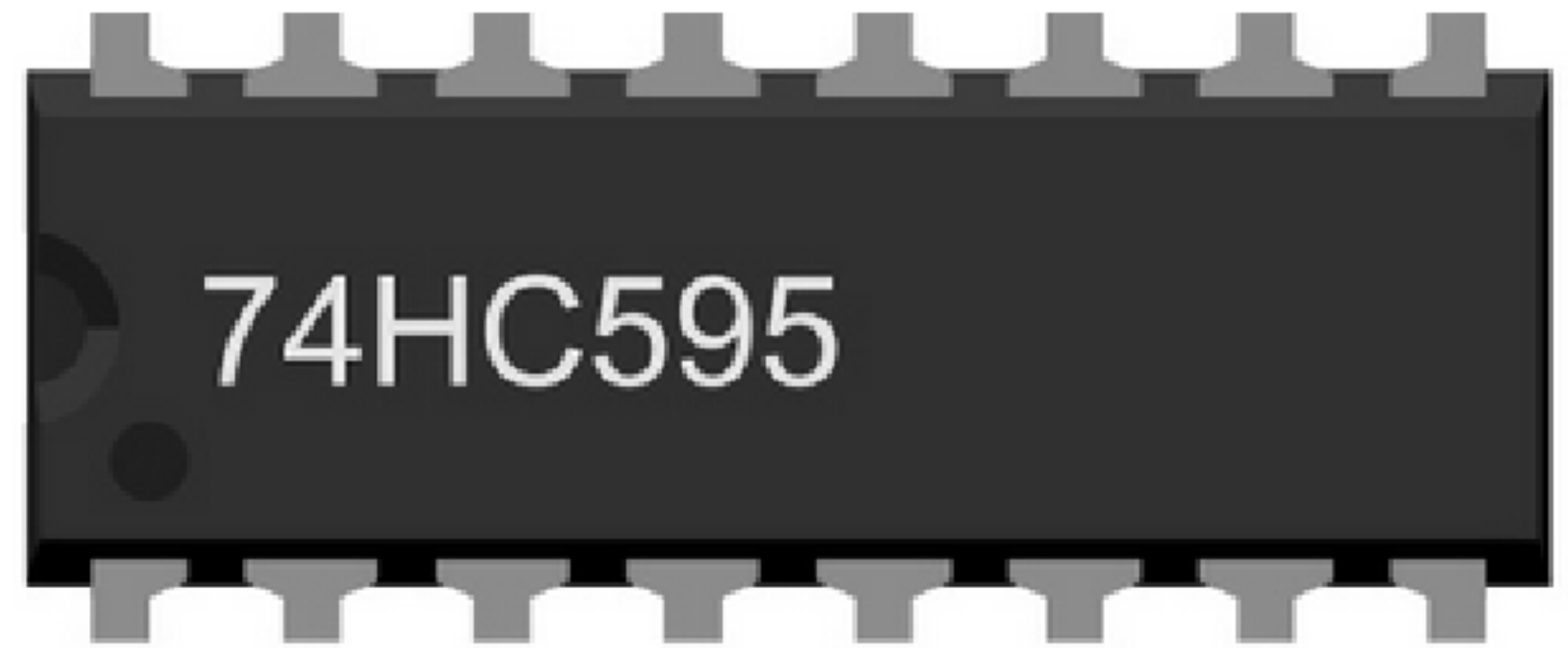
Conversor serial-paralelo

Sinais importantes

SHCP - controla a velocidade do envio dos bits

STCP - repassa os dados recebidos até o momento para a saída paralela

OE - habilita a saída de dados



NÃO

Operação lógica

```
char A = 12;
```

```
char r;
```

```
r = !A // r = 0
```

Operação bitwise

```
char A = 12;
```

```
char r;
```

```
r = ~~A // r = 243
```

```
// A = 0b00001100
```

```
// A = 0b11110011
```



E

Operação lógica

```
char A = 8;
```

```
char B = 5;
```

```
char r;
```

```
r = A && B;
```

```
// r = 1
```

```
// A > 0
```

```
// B > 0
```

```
// B = 1
```

Operação binária

```
char A = 8;
```

```
char B = 5;
```

```
char r;
```

```
r = A & B;
```

```
// r = 0
```

```
// A = 0b000010000
```

```
// B = 0b000000101
```

```
// B = 0b000000000
```

Ou

Operação lógica

```
char A = 8;
```

```
char B = 5;
```

```
char r;
```

```
r = A || B;
```

```
// r = 1
```

```
// A > 0
```

```
// B > 0
```

```
// B = 1
```

Operação binária

```
char A = 8;
```

```
char B = 5;
```

```
char r;
```

```
r = A | B;
```

```
// r = 13
```

```
// A = 0b00001000
```

```
// B = 0b00000101
```

```
// B = 0b00001101
```

OU EXCLUSIVO

Operação lógica

char A = 8;

char B = 5;

char r;

r = (A && !B) || (!A && B);

r = (1 && !1) || (!1 && 1);

r = (1 && 0) || (0 && 1)

r = 0 || 0

// r = 0

// A > 0

// B > 0

// B = 1

Operação binaria

char A = 13;

char B = 5;

char r;

r = A ^ B;

// r = 9

// A = Ob00001100

// B = Ob00000101

// r = Ob00001001

Operação binária?

$A = 5, B = 0, C = 12$

$(A \& !B) \mid \sim(C \wedge A)$

$(0101 \& !0) \mid \sim(1100 \wedge 0101)$

$(0101 \& 1) \mid \sim(1001)$

$(0101 \& 0001) \mid 0110$

$0001 \mid 0110$

0111

$\begin{array}{r} 1100 \wedge \\ 0101 \end{array}$

$\begin{array}{r} 0001 \mid \\ 0110 \end{array}$

$\begin{array}{r} 0101 \& \\ 0001 \end{array}$