

Tempo e execução de múltiplas tarefas

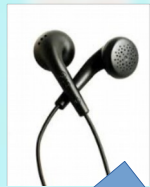
Prof. Dr. Roberto Kenji Hiramatsu
Prof. Dr. João Henrique Correia Pimentel

Agenda

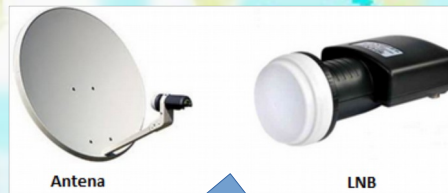
UPPER

Sistemas Analógicos e Digitais

- Sistemas analógicos
 - Manipula quantidades físicas de forma analógica ao longo de uma faixa de operação



Fones de ouvido
Caixas de som



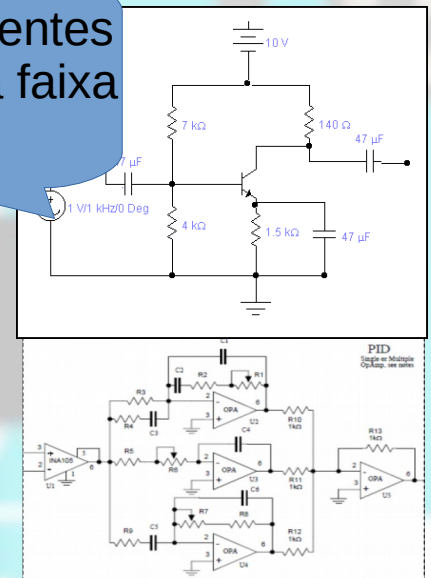
Antena
LNB

Amplificação de
Sinal de antenas



Reguladores de tensão
Fontes reguladas

Uso de componentes
Operando numa faixa
contínua

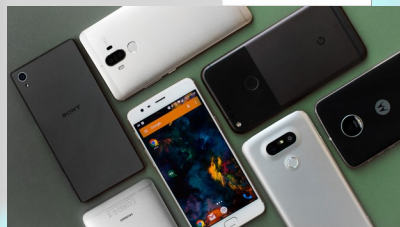


Sistema digital

- Manipula somente representações no formato digital
- São na maioria das vezes eletrônicos mas podem ser mecânicos, magnéticos ou pneumáticos.



Computadores, smartphones



Antes dos computadores
Calculadores mecânicos
Caixa registradora.



Quanto precisamos de uma saída analógica?



Velocidade
Motores
DC/AC



Converter áudio



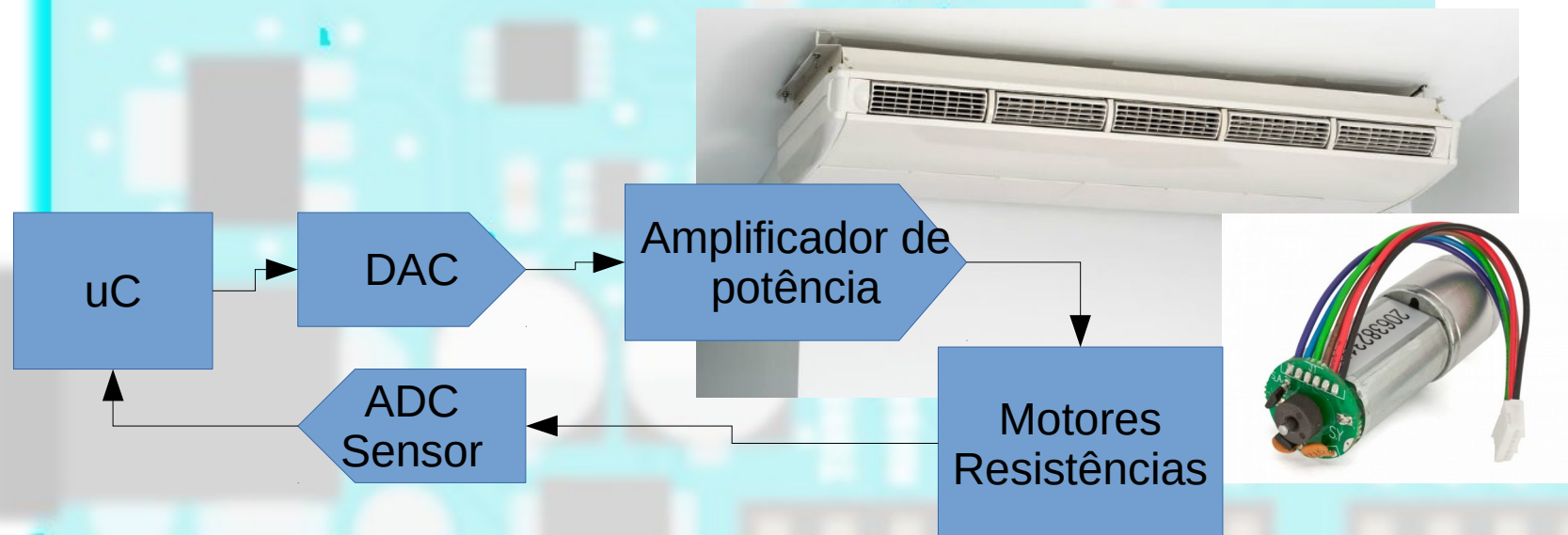
Intensidade de iluminação

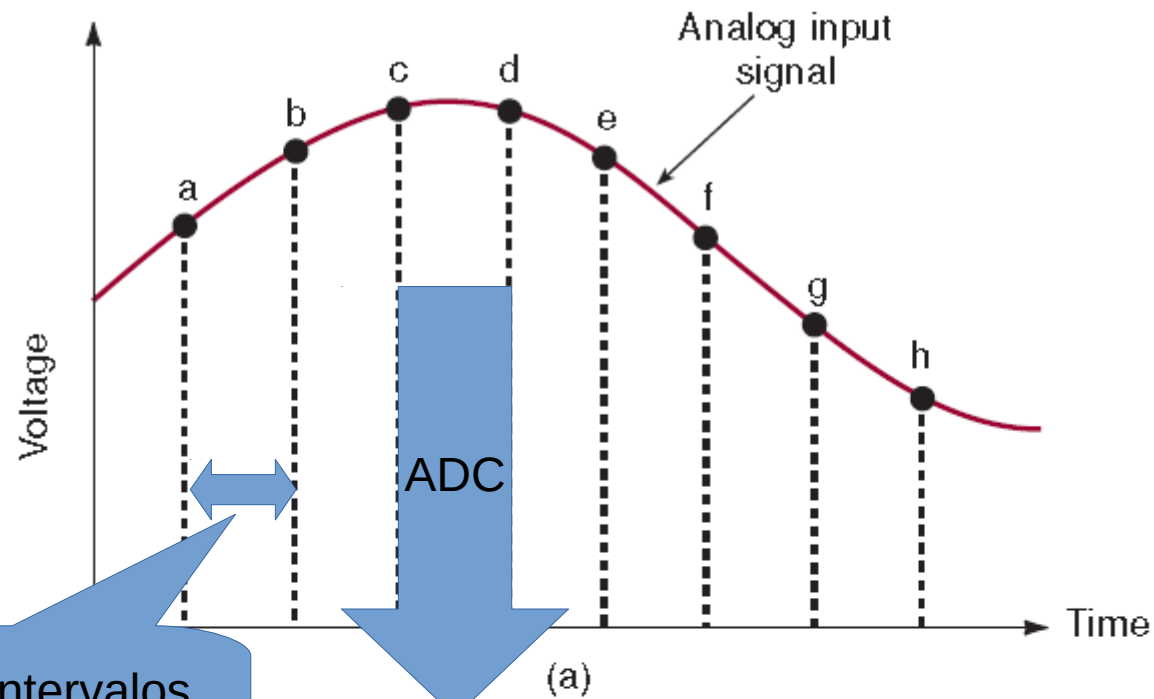
A saídas podem ser geradas por conversores
digital analógico – DAC ou por técnica de
PWM

DAC – Digital Analogic Converter – Conversor Digital-Analógico
PWM – Pulse-Width Modulation

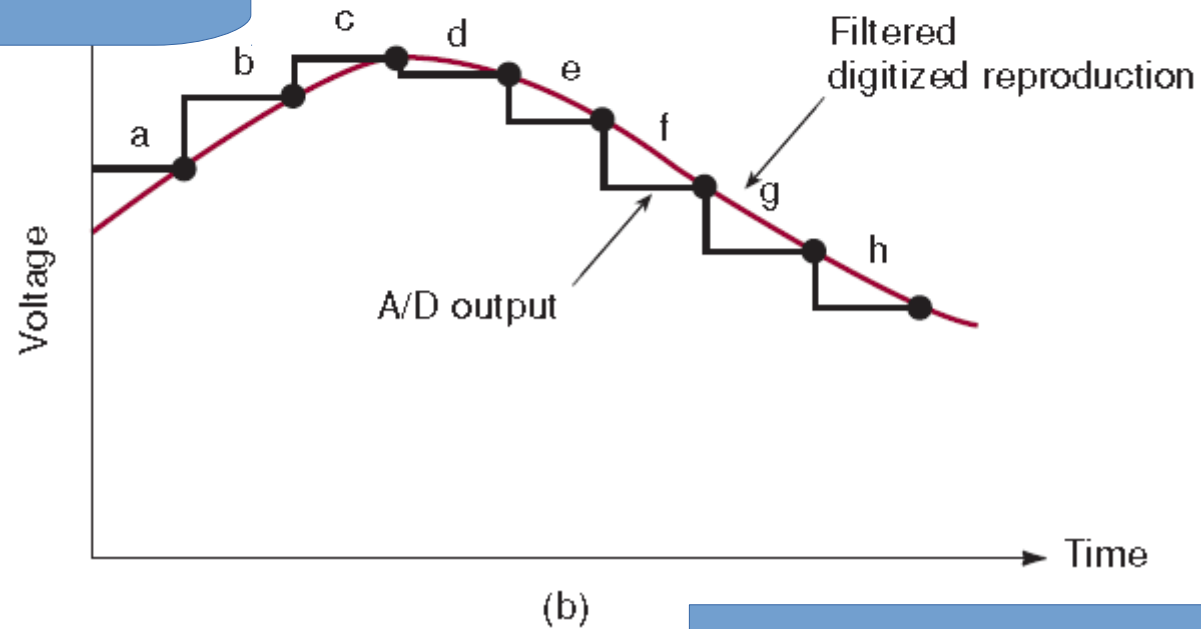
Quando precisamos de uma entrada analógica?

- Medida de uma grandeza do mundo real
- Controle





Captura em intervalos regulares

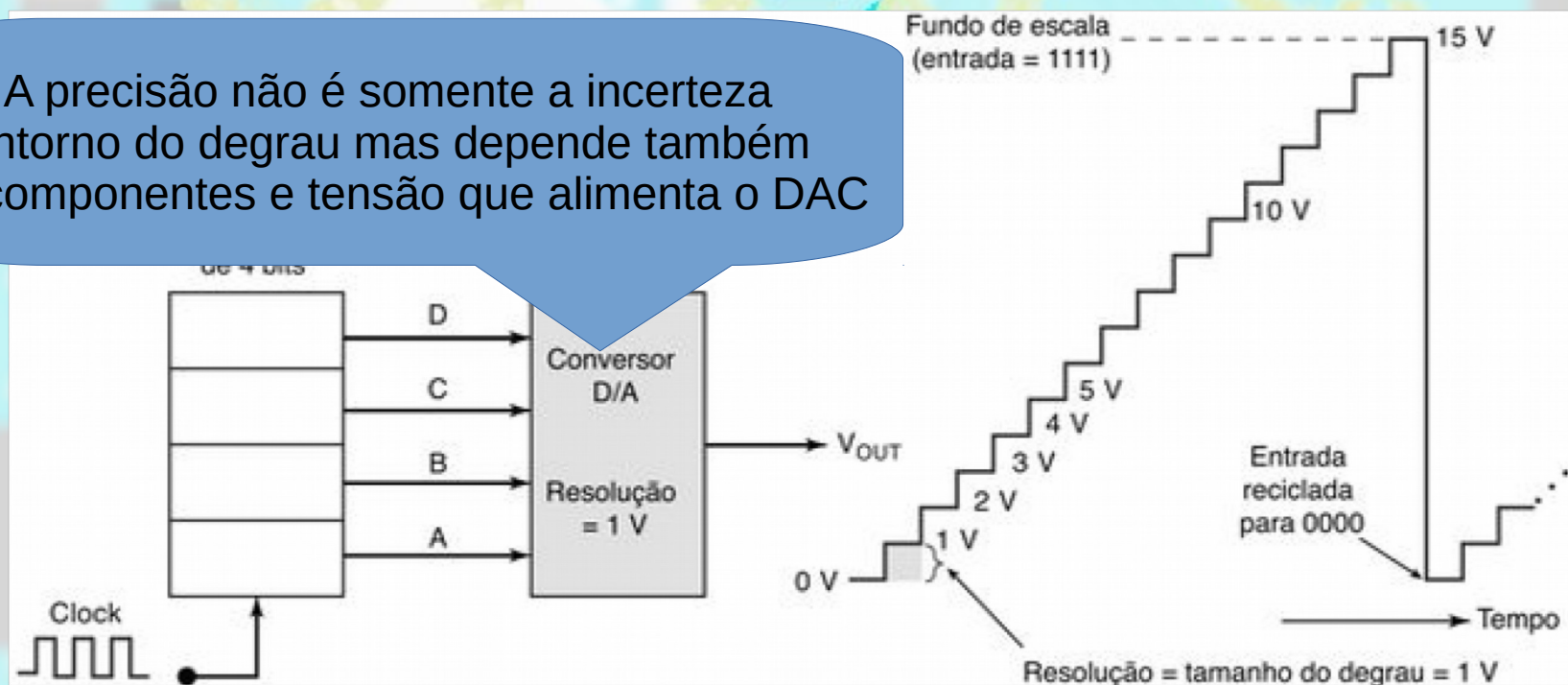


Teoria:

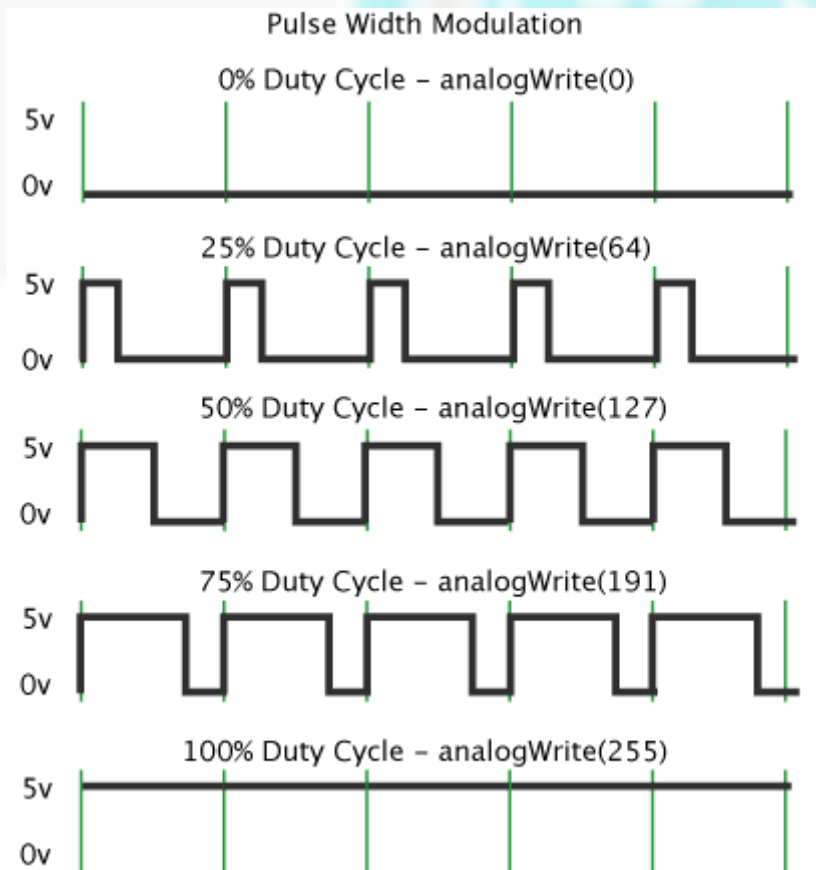
Fundo de escala, Resolução e precisão

- Fundo de escala (fe) 15V em conversão de 4 bits com resolução (K) de 1V
 - Saída analógica = $K \times$ Saída Digital
 -

A precisão não é somente a incerteza entorno do degrau mas depende também dos componentes e tensão que alimenta o DAC



Modulação por largura de pulso – (Pulse Width Modulation - PWM)



- No Arduino UNO R3 não temos o DAC
 - No entanto existem uC que tem saídas DAC como o Arduino DUE
- O PWM, que é a modulação da largura de pulso digital, é “enxergada” como sinal analógico dependendo da frequência de operação
 - Controle de motores e aquecedores por exemplo.
 - Considere o resultado como media
 - Na biblioteca para o UNO R3 opera em 500Hz aproximadamente.
 - Não é adequada para reprodução de áudio.

Pinos no Arduino UNO R3

Timer output	Arduino output	Chip pin	Pin name
OC0A	6	12	PD6
OC0B	5	11	PD5
OC1A	9	15	PB1
OC1B	10	16	PB2
OC2A	11	17	PB3
OC2B	3	5	PD3

Observe que número aqui representa o temporizador usado

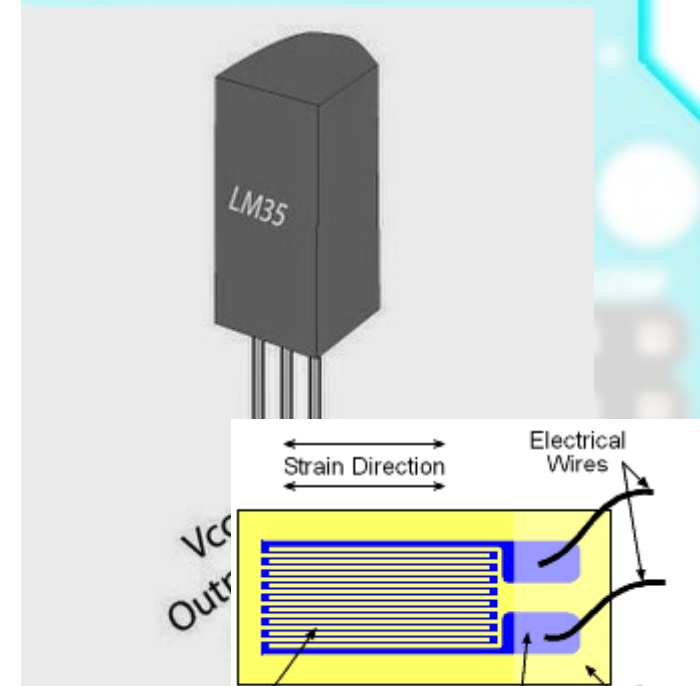
Nem todos os pinos tem saída PWM
Em outros uC tem que verificar disponibilidade

Leitura de sinal analógica

- Uso de ADC – quase todos os uC tem este periférico
 - No arduino UNO R3:
 - Escala de fundo do ADC é 5V.
 - tem ADC de 10bits com resolução de $5V/1024=4,9mV$
 - Converte em 0,1 milissegundos
- O Arduino UNO R3 tem somente 1 periférico ADC acessível dos pinos A0 a A6.

Como é o tratamento de um sinal de sensor para ser lida no ADC

T	B	R	S	Cominação metálica		Código de Cores ANSI MC-96.1	Internacional IEC 584-3	BS 1843	DIN 43710	JIS C1610-1981	Limite de erro	
				LIGAS	+ FIO - FIO						Padrão	Especial
K	J	W	X	Fe (magnético)	Constantan Cu-Ni						2.2°C ou 0.75%	1.1°C ou 0.4%
				Niquel-Cromo Ni-Cr	Niquel-Alum Ni-Al (magnético)						2.2°C ou 0.75%	1.1°C ou 0.4%
				Cobre Cu	Cu-Ni							
T	E	N	S	Cobre Cu	constantan Cobre-Niquel Cu-Ni						1.0°C ou 0.75%	0.5°C ou 0.4%
				Niquel-Cromo Ni-Cr	constantan Cobre-Niquel Cu-Ni						1.7°C ou 0.5%	1.0°C ou 0.4%
				Nicrosil Ni-Cr-Si	Nisil Ni-Si-Mg						2.2°C ou 0.75%	1.1°C ou 0.4%
E	J	R	B	Platina 13% rhodio Pt-10% Rh	Platina Pt						1.5°C ou 0.25%	0.6°C ou 0.1%
				Platina 10% rhodio Pt-10% Rh	Platina Pt						1.5°C ou 0.25%	0.6°C ou 0.1%
				Platina 30% rhodio Pt-30% Rh	Platina 6% rhodio Pt-6% Rh							



Características:

Resistencia 350 Ohms

O valor médio da tolerância de resistência 0,1%

Fator de sensibilidade 2,00-2,20

Coeficiente de dispersão sensível 1%

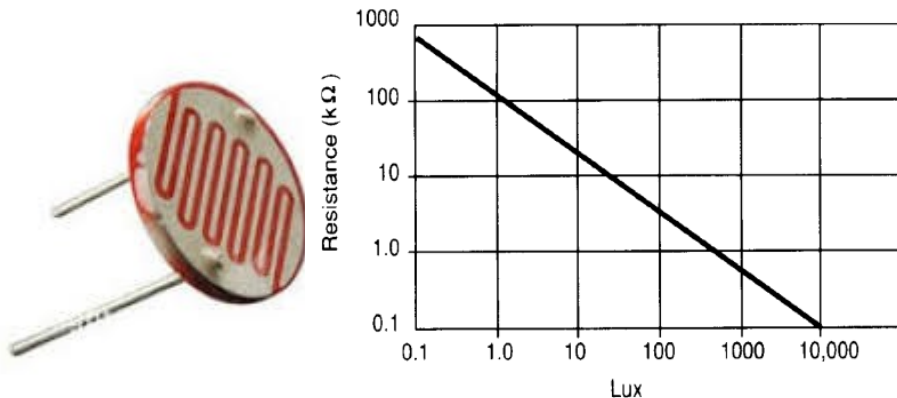
Limite de deformação 2,0%

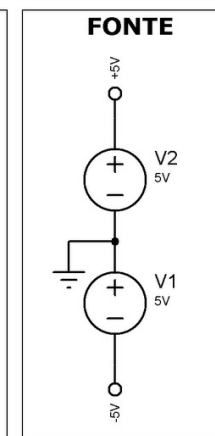
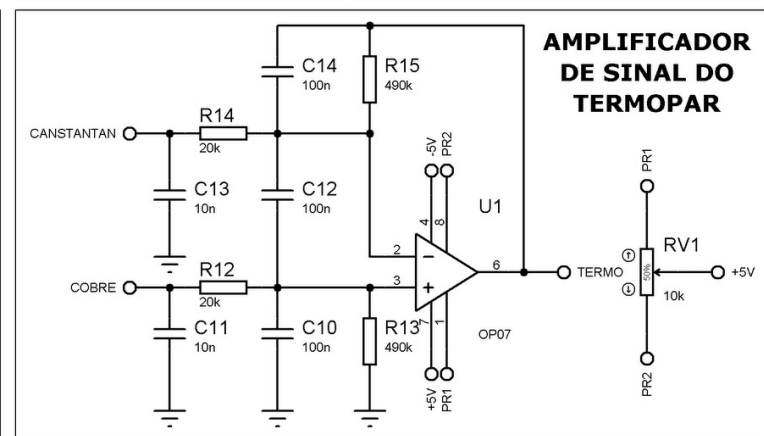
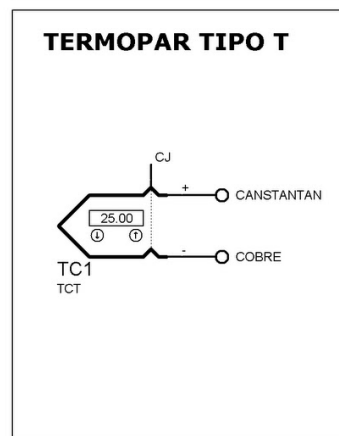
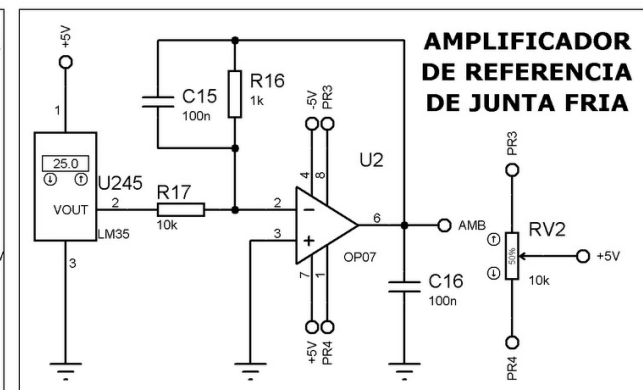
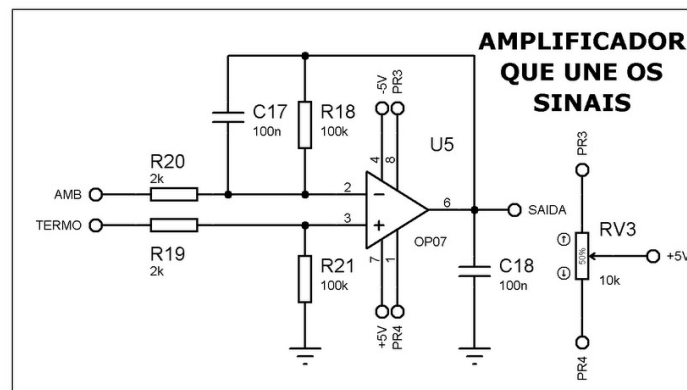
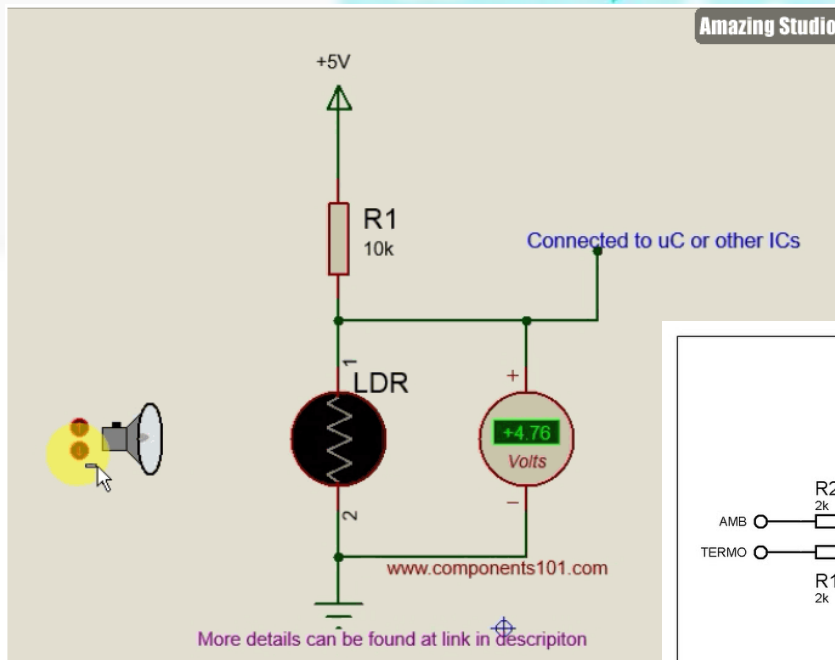
Ciclos de Vida: 1M

Coeficiente de compensação de temperatura: 9,11,16,23,27

Temperatura de trabalho: -30- + 80

FIGURE 2 RESISTANCE AS FUNCTION OF ILLUMINATION





Exemplos de termopares para arduino



MAX6675 tipo K-módulo sensor de temperatura termopar MAX6675

R\$ 6,74 de AliExpress.com

em de no AliExpress.com | Alibaba Group



Modulo Max6675 + Termopar Tipo K Arduino Excelente!

R\$ 26,40 de Mercado Livre - Absolem Eletrons

MAX6675 + Termopar K 1/2m + jumpers Descrição: MAX6675 é um conversor AD com compensação de junta fria , correção de ...



Sensor Termopar Tipo K 600oc Modulo Leit.max6675 P/ Arduino

R\$ 35,00 de Mercado Livre - Argfbig

Termopar tipo K com medição até 600oC.Modulo de leitura utiliza o chip MAX6675 para fazer a conexão do termopar ao arduino ...

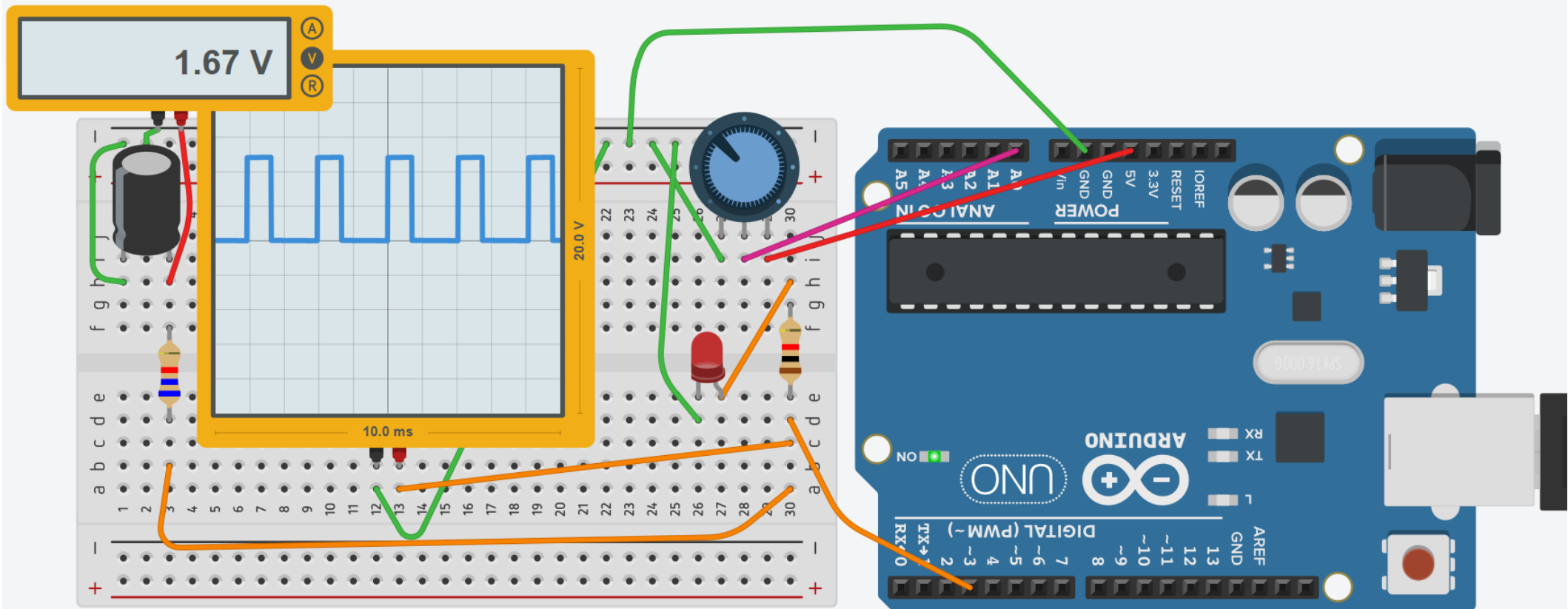


Termopar Tipo K 0 a 800°C + Módulo de Leitura Max6675

R\$ 37,90 de Eletrogate ★★★★★ (362)

Este Kit contém um Termopar Tipo K e um Módulo de Leitura Max6675 que são capazes de realizar medições de temperatura de 0°C à ...

Montar o circuito abaixo para entender o PWM



Código teste

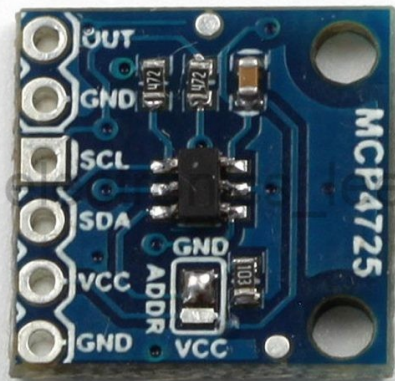
```
1  /** Autor: Roberto Kenji Hiramatsu
2  Demostracao de leitura de potenciometro para controle de
3  intensidade de led
4  */
5  int pinAnalog=3;           //Definindo PWM no pino 3
6  void setup(){
7      pinMode(pinAnalog, OUTPUT); // Precisa dizer que e uma saida di
8      Serial.begin(9600);      // Para ver o valor lido
9  }
10 int anteriorV;
11 void loop(){
12     int v=analogRead(A0);     // Faz a leitura do valor analogico
13     if(v!=anteriorV){         // So imprime se for diferente
14         Serial.println(v);
15         anteriorV=v;
16     }
17     analogWrite(pinAnalog,v>>2); // v>>2 e o mesmo que v/4
18     delay(10);
19 }
```

No Arduino os pinos analógicos são acessados com prefixo A seguindo do Número do pino analógico

Acionamento de motores

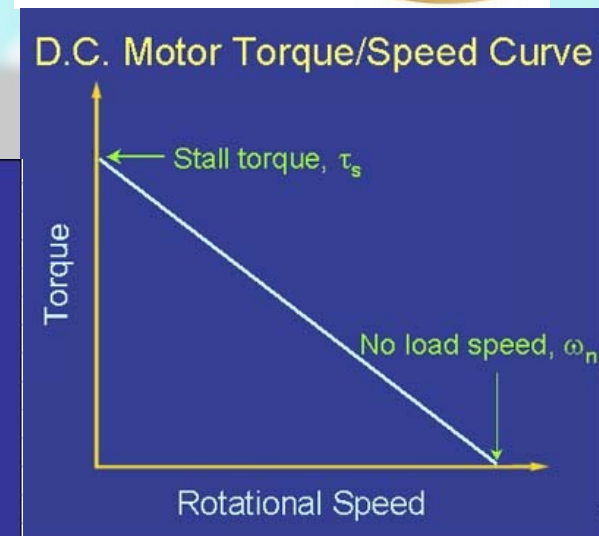
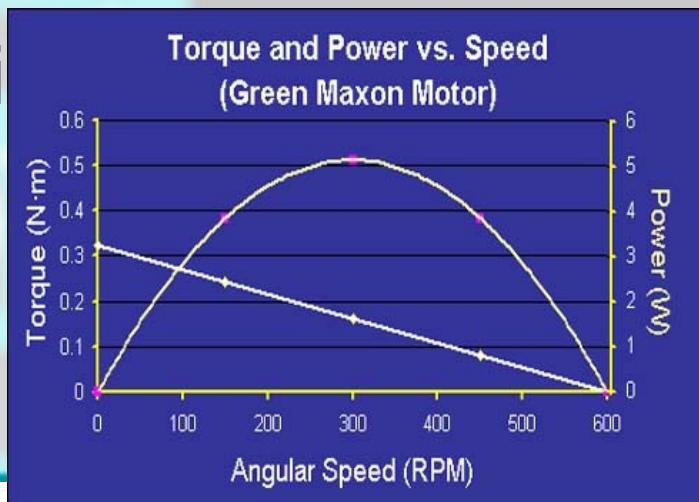
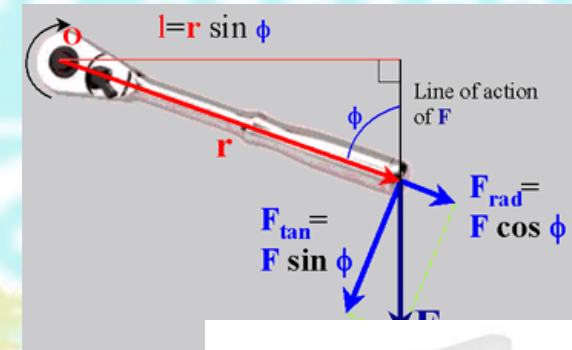
- Motores DC
 - Acionamento usando PWM para gerar uma tensão média
 - Normalmente aceita rotação nos dois sentidos por meio de mudança de polaridade
- Motor DC tipo Servomotor
 - Usando o PWM para estabelecer ângulo de giro – possui internamente controle PID para manter a posição
 - Usando em Robos
- Motor DC tipo Stepper (motor de passo)
 - Usa sequência de pulsos para mudar a posição
 - Não permite controle fino dos passos e não permite verificação de posição
- Motores AC
 - Criando uma forma de onda com frequência controlada por uC
 -

DACs



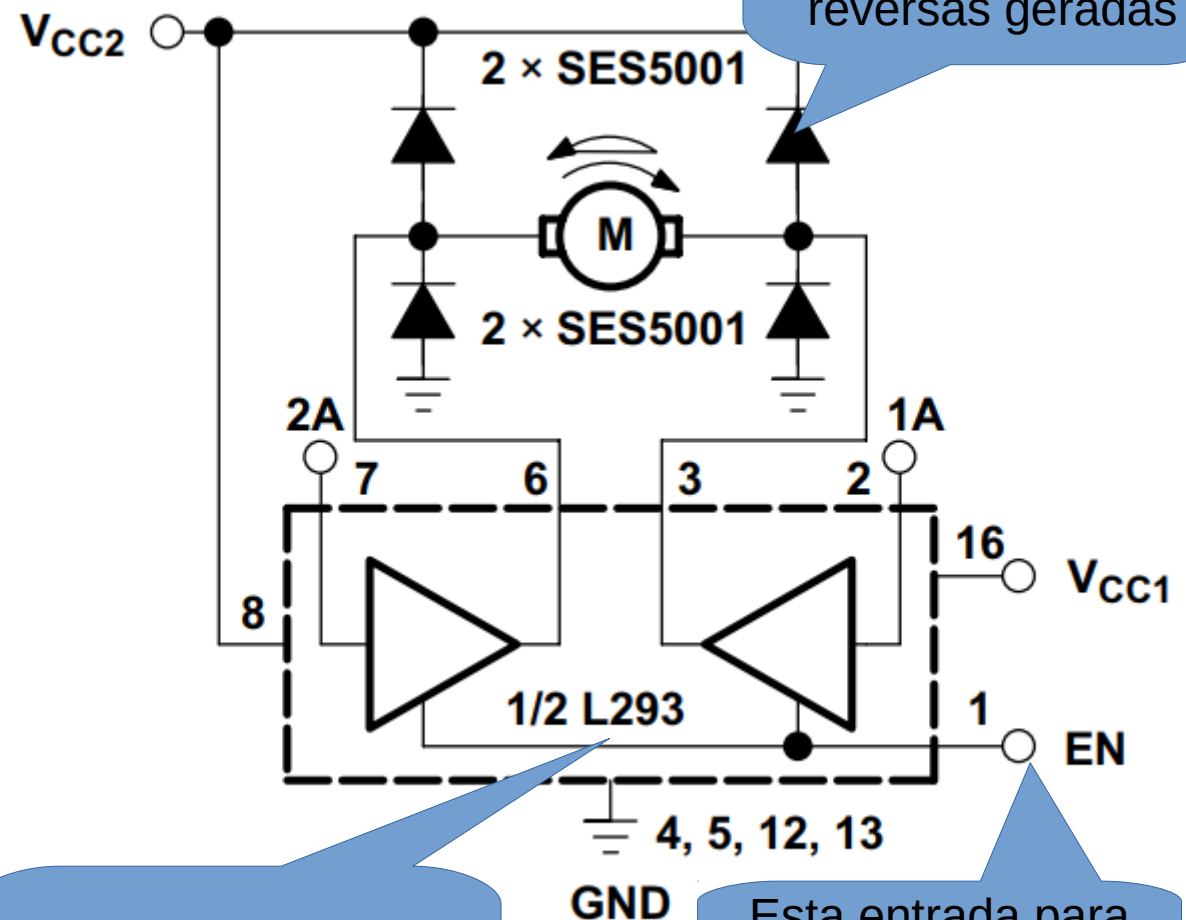
Algumas características a pensar no motor

- Torque
 - Sg90 → 1.8Kg/cm em máxima carga (stall)
- Curva de velocidade/torque
- Curva de Potência/torque
 - Considere usar do ponto de carga livre até o ponto de potencia máxima.
- Corrente de parti



Amplificadores de potencia Motores Drivers e ponte H

Diodos são colocados para proteger o circuito eletrônico das tensões reversas geradas

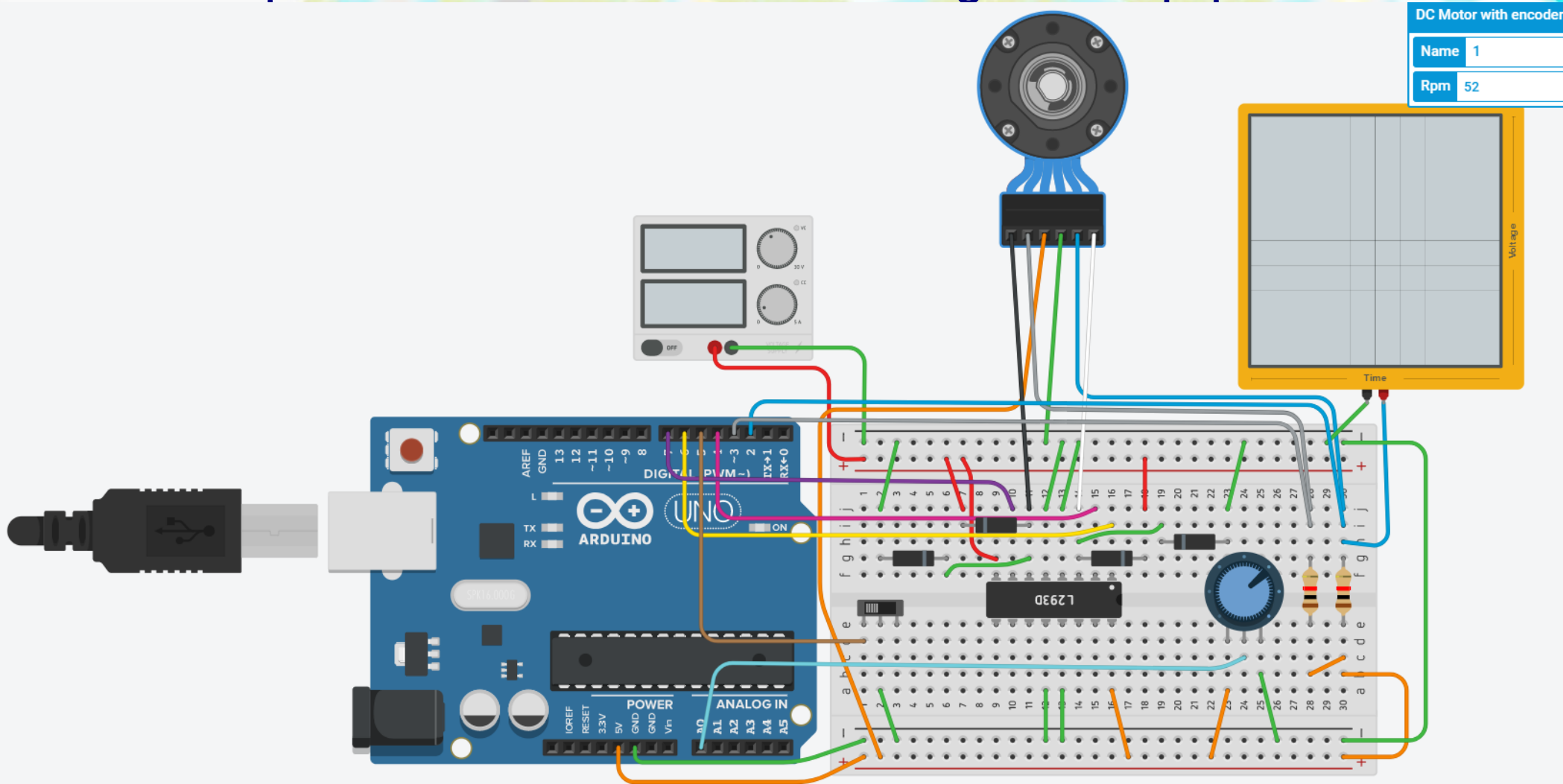


Meia ponte H disponível tinkercard

Esta entrada para PWM

Vamos analisar o acionamento de motor abaixo

- Duplique o projeto de:
 - <https://www.tinkercad.com/things/iNXi9pq43RE>



Eventos gerados por pinos externos tratamento do encoder

- Interrupções externas são recebidas nos pinos 2 e 3 do Arduino UNO R3 e servem para atender operações geradas por sinais externos
 - O encoder do motor gera pulso que precisam ser medidos

```
46 /*Conta pulsos de
47 volatile int cA=0;
48 void contaCHA(){
49     cA++;
50 }
```

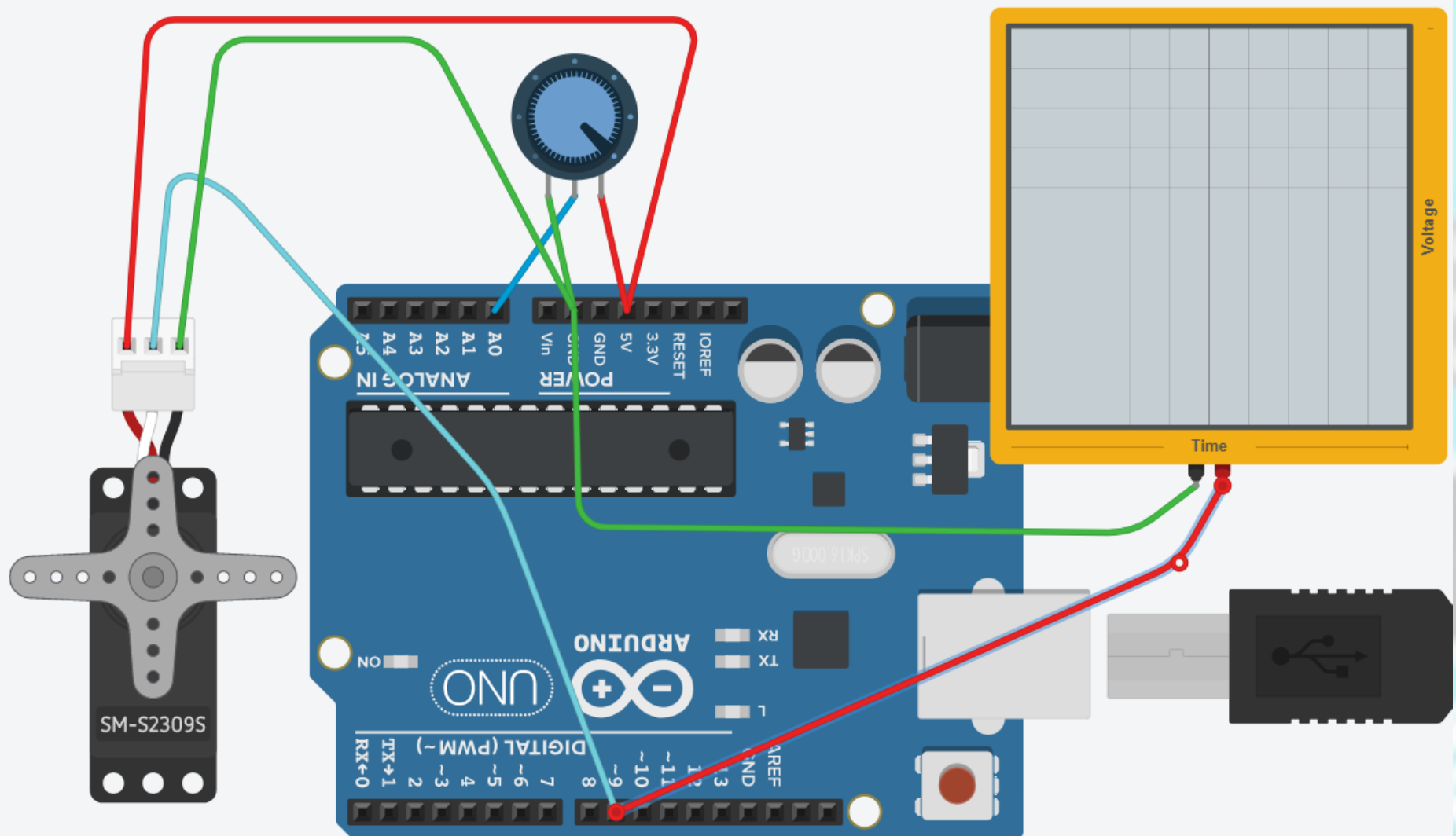
Por se tratar de evento gerado fora do loop normal o modificador volatile deve ser aplicado para a variável seja segura durante o acesso normal e por interrupção

Uma interrupção é definida como uma função que é sempre chamado quando ocorre o evento

```
57 void setup()
58 {
59     Serial.begin(9600);
60     iniciaLog();
61     attachInterrupt(digitalPinToInterrupt(pinEncCHA), contaCHA, RISING);
62     //attachInterrupt(digitalPinToInterrupt(pinEncCHB), contaCHB, RISING);
63     sentidoRot(false);
64 }
```

O Arduino define attachInterrupt para registrar a interrupção

Exemplo usando servomotor e biblioteca servo.h



Código exemplo com biblioteca servo.h

```
#include <Servo.h>
```

```
Servo myservo; // create servo object to control a servo
```

```
int potpin = 0; // analog pin used to connect the potentiometer
```

```
int val; // variable to read the value from the analog pin
```

```
void setup() {
```

```
  myservo.attach(9); // attaches the servo on pin 9 to the servo object
```

```
  Serial.begin(9600);
```

```
}
```

```
int antVal=0;
```

```
void loop() {
```

```
  val = analogRead(potpin); // reads the value of the potentiometer (value between 0 and 1023)
```

```
  if(antVal!=val){
```

```
    antVal=val;
```

```
    Serial.println(val);
```

```
}
```

```
val = map(val, 0, 1023, 0, 180); // scale it to use it with the servo (value between 0 and 180)
```

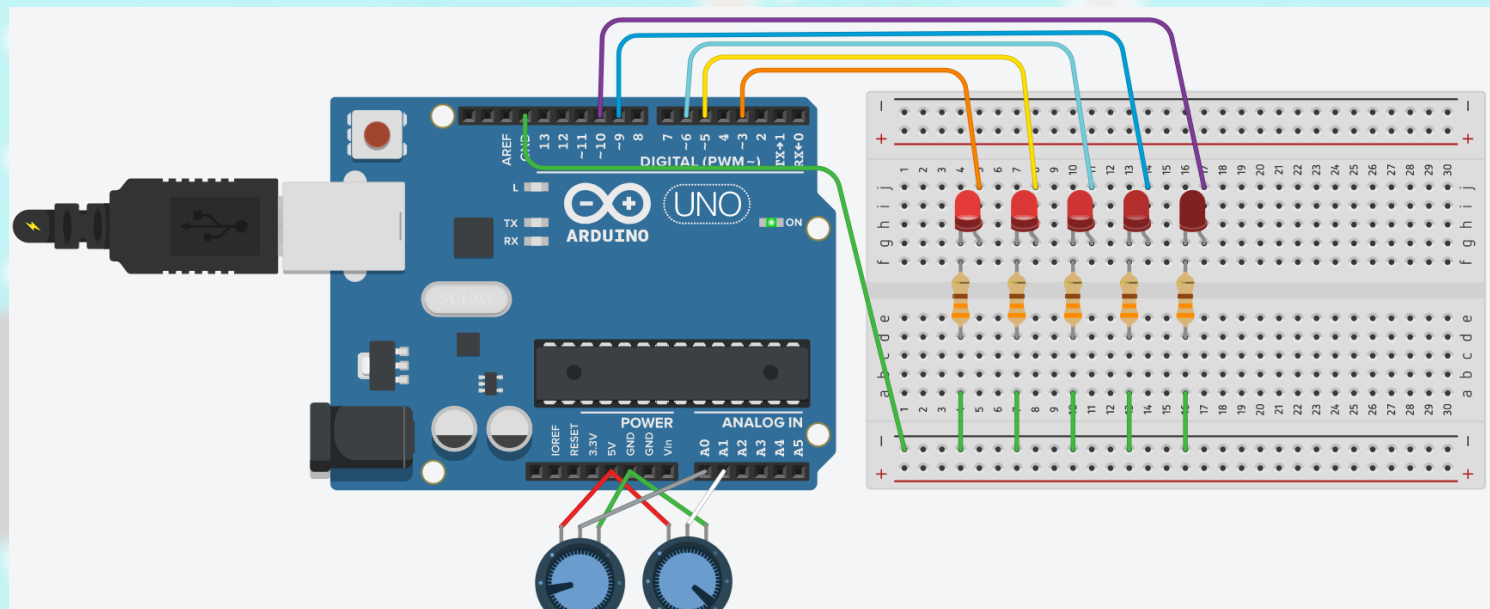
```
myservo.write(val); // sets the servo position according to the scaled value
```

```
delay(15); // waits for the servo to get there
```

```
}
```

Execício proposto 1

- Construa o circuito abaixo similar ao proposto na 2 aula no Tinkecar. Faça com que o potenciômetro ligado no analógico A0 controle a velocidade de acendimento dos Leds com valor variando de 250ms a 2 segundos e que a transição do led de maio brilho desloque da direita para a esquerda e vice-versa. O potenciômetro ligado ao Analógico A1 deve controlar o brilho em torno do led de maior brilho como mostra a figura abaixo.



Proposição 2

- Usando o circuito de controle do motor mostrado nesta aula, proponha um projeto com controle de 2 motores que permita realizar uma volta circular com raio de 50cm.
- Considere para isso uma roda de 65mm de diâmetro e um chassi entre eixos de 10 cm.
- Dica 1: Desenhe e determine a proporção que cada roda deve percorrer com base na velocidade da roda mais rápida.
- Dica 2: Considere que o valor do PWM gera velocidade proporcional para tensões nas faixas entorno de 6V.

Leituras e videos

- Veja as explicações sobre registro de interrupção
- <https://www.arduino.cc/reference/en/language/functions/external-interrupts/attachinterrupt/>
- <https://playground.arduino.cc/Code/Interrupts>
- PWM
- Amplificadores
 - <https://www.electronics-tutorials.ws/amplifier/amplifier-classes.html>

Um pouco de teoria de motores DC

<http://lancet.mit.edu/motors/motors3.html>

Determinando parâmetros de um motor

<https://www.youtube.com/watch?v=roINUVVpEbs>

<https://www.youtube.com/watch?v=gwFE-NkzR-k>

Motores tipo stepper e servo

<http://www.ni.com/white-paper/14863/en/>

Leitura adicional

- Termopares
 - <http://www.estatica-metrologia.com.br/termopares.php>
- Circuito
 - <http://diego-bittencourt.blogspot.com.br/2011/09/boa-tarde-conforme-combinado.html>
- LDR
 - <https://components101.com/ldr-datasheet>



roberto.hiramatsu@ufrpe.br