

Imbrax Lab - Eletrônica com Diversão

Blog destinado a apresentar notícias, projetos, experimentos relacionados a robótica, eletrônica e programação. Tudo isso com uma pitada de diversão.

quinta-feira, 22 de julho de 2010

Controle da interface Cyberbox com SuperLogo

Apresentação:

Olá amigos, hoje teremos um post diferente dos anteriores. Apresentaremos exemplos de como controlar a interface Cyberbox utilizando o software educacional SuperLogo. Serão dois exemplos que ajudarão a compreender a utilização das portas de entrada analógicas e digitais. Com base nos valores dos sensores das portas de entrada uma ação será executada e assim uma porta de saída será acionada.

Estes exemplos foram desenvolvidos durante o rápido mini curso que ministramos no Centro Social Marista no bairro Sabará em Curitiba onde a Cyberbox será utilizada com adolescentes do bairro.

Seguidores

Arquivo do blog

- ▼ [2010](#) (5)
 - ▼ [Julho](#) (1)
 - [Controle da interface Cyberbox com SuperLogo](#)
 - ▶ [Junho](#) (2)
 - ▶ [Maio](#) (2)

Quem sou eu

[IMBRAX - LAB](#)

[Visualizar meu perfil completo](#)



**Material necessário:**

Para este experimento precisaremos de alguns materiais:

- 1 Interface Cyberbox, seu cabo serial e fonte de alimentação
- 1 Versão do SuperLogo que pode ser baixado gratuitamente do site da [Unicamp](#)
- 1 Biblioteca de controle da Cyberbox para Superlogo que pode ser baixado [Imbrax](#)
- 1 Potenciômetro de 10k Ω
- 1 Motor DC de 12V
- 1 Sensor Reed Switch
- 1 Buzzer 12V
- 1 Alguns centímetros de fio AWG 22 pode ser necessário
- 1 Alicates e chave de fenda

Para saber mais sobre o sensor Reed Switch acesse [aqui](#), para saber mais sobre o Buzzer acesse o [link](#).

O que faremos?

Dois experimentos que farão a leitura das portas de entrada e com base nos sinais adquiridos tomarão uma decisão. O primeiro exemplo fará a leitura de um potenciômetro conectado a porta de entrada analógica, com base no valor do potenciômetro um motor DC será ligado ou desligado. O segundo exemplo fará o acionamento de um buzzer conectado na porta de saída baseado no sinal da entrada onde temos o sensor reed switch conectado.

Como ligar os acessórios?

Agora começa a parte divertida e prática, a montagem dos acessórios dos exemplos. Siga o texto que a montagem é simples e rápida.

Exemplo 1

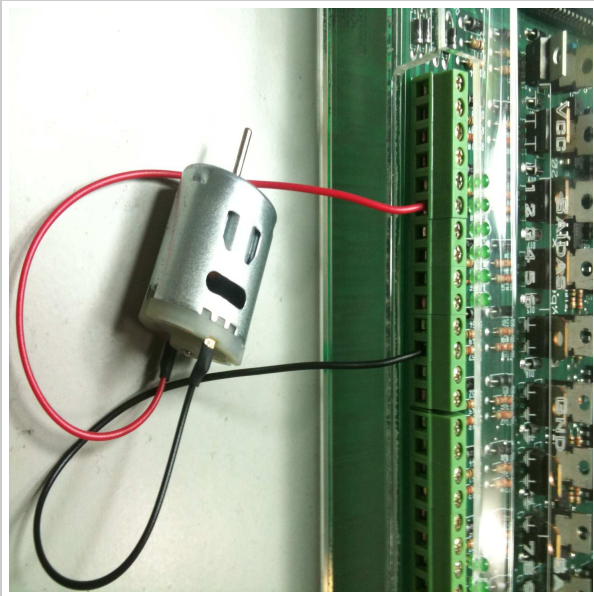
Utilizaremos neste exemplo um motor DC de 12V, a seguir a imagem do motor que utilizamos.



Também usaremos um potenciômetro, que neste caso foi um de 10k Ω .



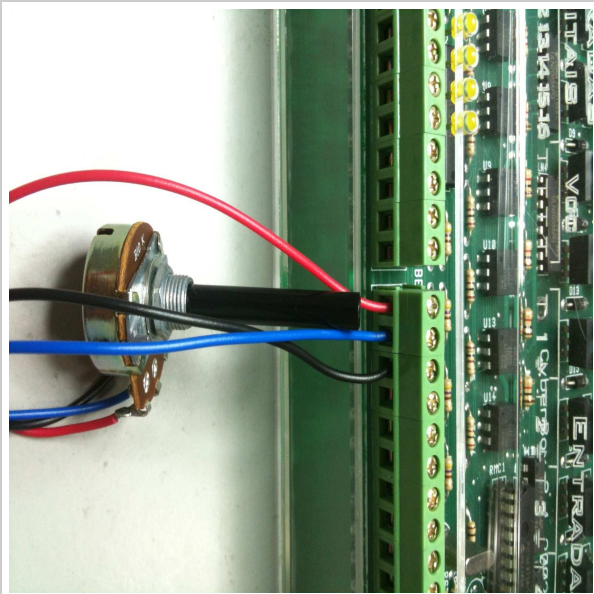
O motor foi conectado na porta de saída número 2. Ligue um dos fios do motor na porta 2 e o outro fio na porta auxiliar GND.



O potenciômetro que chamaremos de pot é um resistor variável, ou seja, sua resistência varia de acordo com o movimento do seu eixo. Girando no sentido horário ou anti-horário variamos a resistência e assim a tensão no seu pino de saída varia. Assim, sendo temos para cada movimento do eixo diferentes níveis de tensão. Esse comportamento o caracteriza como um elemento analógico e por isso ele vai ligado nas portas analógicas da Cyberbox.

O pot foi conectado na porta de entrada analógica 1. Seus três fio foram ligados da seguinte forma:

Fio vermelho, alimentação foi conectado na porta auxiliar VCC (onde temos 5Volts). O fio azul, de sinal foi conectado na porta de entrada 1. E o fio preto, terra foi ligado na porta auxiliar GND. Veja a imagem para fica bem claro.



Pronto, foi bem rápido hein?!

Exemplo 2

No segundo exemplo utilizaremos o sensor reed switch, visualize abaixo o que utilizamos. Note que ao centro temos o ímã que quando esta próximo do sensor fecha seu contato e quando esta longe os

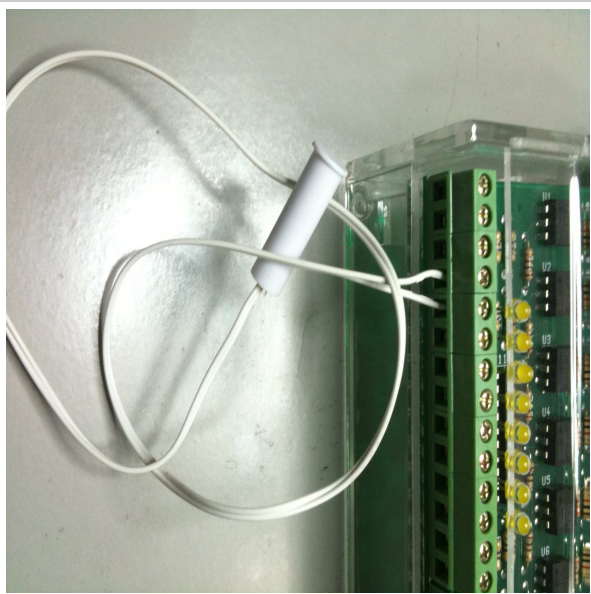
contatos permanecem abertos.



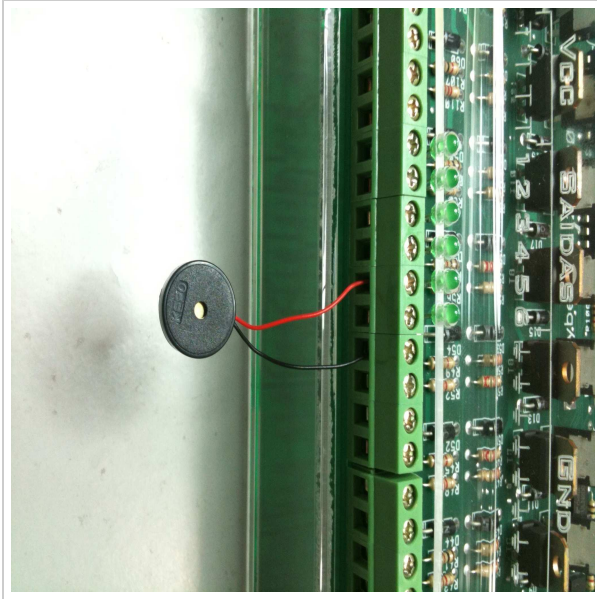
Também temos o buzzer que será visualizado diretamente na imagem de conexão.

Como montar?

O reed switch é um elemento digital pois ele admite duas opções. 0 ou 1, ligado ou desligado ou ainda uma tensão de 0V ou 5Volts como em nosso exemplo. No nosso caso, um dos terminais do reed switch que chamaremos de RS foi ligado na porta auxiliar VCC a qual fornece uma tensão de 5V. O outro terminal foi ligado na porta digital 1. Assim, quando o ímã se aproxima do sensor a tensão de 5V da porta auxiliar é transmitida para a porta digital. Internamente a Cyberbox considera essa tensão de 5V como um sinal verdadeiro ou ligado.



O buzzer foi ligado na porta de saída 5. Ligue o fio vermelho na porta de saída 5 e o fio preto na porta auxiliar GND. Pronto, visualize abaixo a imagem do buzzer.



Programação:

Agora que conectamos os acessórios vamos ligar os cabos da Cyberbox, ligue primeiro o cabo serial na Cyberbox e depois no Computador. Em seguida ligue o plugue de alimentação na Cyberbox e depois na tomada. Pronto, os 2 leds vermelhos devem acender.

Agora podemos mergulhar a programação com a linguagem LOGO que é bem simples, como mencionado anteriormente usaremos o Superlogo fornecido pela Unicamp.

Clique no ícone do Superlogo na sua área de trabalho ou selecione o link onde ele foi instalado. Normalmente seu arquivo executável esta em C:\Program Files\SuperLogo\SLogo.exe. Para facilitar chamaremos o Superlogo de SL.

Depois de aberto o SL é necessário carregar a biblioteca de comandos de controle da Cyberbox para o SL. Para isso clique no menu Arquivo e em seguida clique Abrir. Agora selecione o arquivo cblogo22.lgo e clique no botão Abrir.

Na linha de comando do SL digite: "cyberbinit" (sem as aspas duplas) e pressione ENTER. A mensagem: [Biblioteca carregada com êxito] deverá aparecer na sua tela.

Agora é necessário mais um comando, digite "SR" (sem as aspas duplas) e pressione ENTER.

Ótimo agora já podemos controlar a Cyberbox.

Exemplo 1

Abra o menu Procedimento e crie um novo. Copie o código abaixo dentro da tela que foi aberta.

A explicação do código está inserida na forma de comentários.

```
; Procedimento para controle de um motor usando um potenciômetro.
aprenda controle_pot_motor
```

```
; Este comando aciona a porta analógica 1
apa 1
```

```
; ativetemporizador cria um bloco de código que será executado em
background pelo SL a cada
```

```
; 500 milissegundos. Esse bloco de código é identificado pelo número
```

```

13
ativetemporizador 13 500 [

; le porta analogica 1
lpa 1

; mostra para usuario valor da variável da porta analogica 1
escreva :VAL_ENTA1

; Aqui temos a lógica, se o valor do pot for maior que 500
executamos o comando
; ligue 2 senão executamos o comando desligue 2
senao :VAL_ENTA1 > 500 [
ligue 2
] [
desligue 2
]
]
fim
---
```

Depois de inserir o código na tela "Editor de Procedimentos" selecione o menu Área de Trabalho e clique em Atualizar. Pronto, agora você pode executar o procedimento criado. Na janela de comandos digite: controle_pot_motor.

Deve ser exibido na tela um número indicando o valor do pot, gire o pot no sentido horário e anti-horário e verifique se o motor DC liga ou desliga conforme o valor fica maior que 500.

Exemplo 2

Copie o código abaixo para o SL.

```

---
```

; Procedimento para controle de um buzzer a partir de um reed switch.
aprenda rs_buzzer

```

; habilita porta digital 1
apd 1

; ativetemporizador cria um bloco de código que será executado em
background pelo SL a cada
; 500 milissegundos. Esse bloco de código é identificado pelo número
13
ativetemporizador 13 500 [

; le o valor da porta digital 1
lpd 1

; escreve na tela se o valor é 0 ou 1
escreva :VAL_ENTD1

; se o valor da entrada 1 for igual a 1 então executamos o comando
ligue 5 que
; aciona a porta de saída 5 onde o buzzer esta conectado. Caso
contrário executamos
; o comando para desligar a porta 5
senao :VAL_ENTD1 = 1 [
ligue 5
] [
desligue 5
]
]
```

```
]
fim
---
```

Depois de inserir o código na tela "Editor de Procedimentos" selecione o menu Área de Trabalho e clique em Atualizar. Pronto, agora você pode executar o procedimento criado. Na janela de comandos digite:rs_buzzer.

Deve ser exibido na tela um número indicando o valor do sensor rs, 0 ou 1. Aproxime o ímã ao sensor reed switch e veja que o valor deve mudar para 1, afastando volta para 0. Assim, quando o valor é 1 ou verdadeiro o buzzer é ligado e então emite o som.

Legal não? Agora você pode explorar outros acessórios como botões e quem sabe desenvolver um aparelho de som. Ou pode usar o sensor de luz para controlar o acionamento de uma lâmpada quando esta escuro ou até mesmo usar um sensor de temperatura de quando esquentar acionar um motor com um pá para esfriar. Tudo depende da sua imaginação.

O SL possui uma sistema de ajuda embutido dentro do aplicativo, para acessá-lo selecione o menu Ajuda e depois clique em Índice. Lá você encontrará informações sobre a programação do SL.

Para saber mais sobre a interface Cyberbox, suas características, compatibilidade, recursos clique [Cyberbox](#). Mais detalhes sobre SuperLogo acesse [aqui](#).

Siga nosso twitter para novidades, twitter.com/imbraxlab.

Dúvidas, comentários e sugestões são bem vindas.

Abraço, Diego
twitter.com/imbraxlab

Postado por Imbrax - Lab às [13:52](#) [0 comentários](#)

Marcadores: [buzzer](#) [motor](#) [potenciômetro](#) [imbrax](#) [cyberbox](#) [sensor](#) [reed switch](#) [robótica](#) [curitiba](#) [marista](#) [sabarã](#) [educação](#) [programação](#) [eletrônica](#) [superlogo](#)

sexta-feira, 25 de junho de 2010

Controlando Display de 7 Segmentos com CD4511 e Arduino

Apresentação:

Olá amigos hobistas, fãs do Arduino, programadores ou apenas curiosos. Depois de algumas semanas sem postar estamos de volta com um projeto bacana. Hoje faremos um contador utilizando um display de 7 segmentos.

O display de 7 segmentos é normalmente utilizado para indicar números e podem ser vistos normalmente em elevadores, alguns rádio relógios.

Aqui está um exemplo de um display de elevador.

**Material necessário:**

Para este experimento precisaremos de alguns materiais:

- 1 Placa Arduino Duemilanove e seu respectivo cabo
- 1 Fonte de alimentação de 5V
- 1 Protoboard
- 7 resistores de 150Ω
- 1 CI CD4511- Decodificador BCD - 7 Segmentos
- 1 Display de 7 Segmentos Catodo Comum
- Alguns fios para conectar os componentes

O que faremos?

Criaremos um código para a placa Arduino, este enviará para um decodificador BCD-7-Segmentos 4 bits, esses serão decodificados e então serão convertidos em 7 bits então usados para controlar um display de 7 segmentos.

Para saber mais sobre códigos BCD visite o link para a [Wikipedia](http://pt.wikipedia.org/wiki/C%C3%B3digo_BCD).

Como ligar os acessórios?

Neste post testaremos uma nova forma de mostrar como os acessórios foram ligados, apresentarei o circuito na forma de um desenho feito com o software [Fritzing](http://fritzing.org/) que pode ser baixado gratuitamente do seu site. Veja na imagem a seguir como ficou o circuito ou clique [aqui](#) para ver em tamanho real.

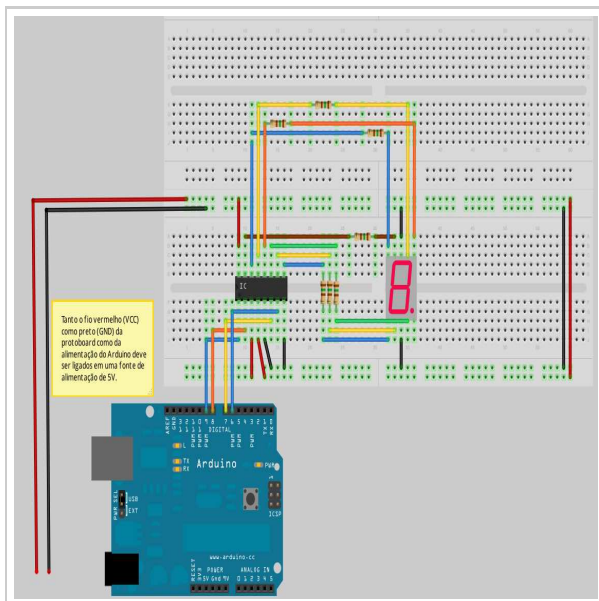
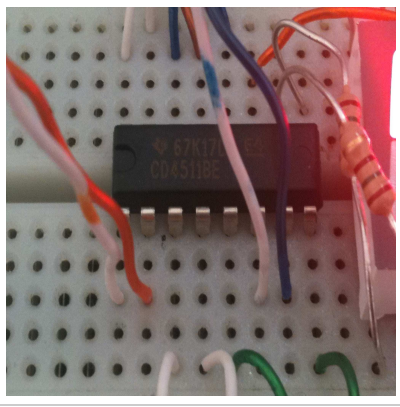
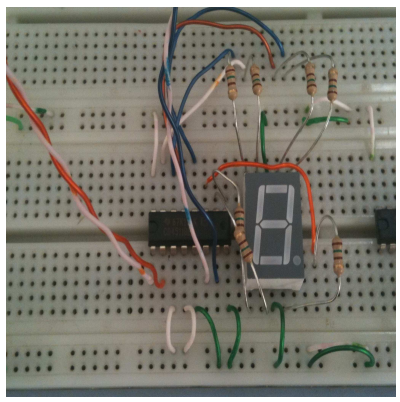
**Montagem:**

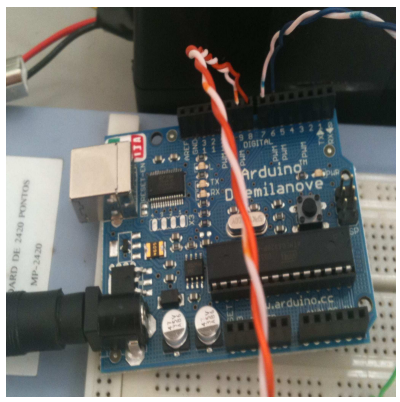
Image do CI CD4511BE.



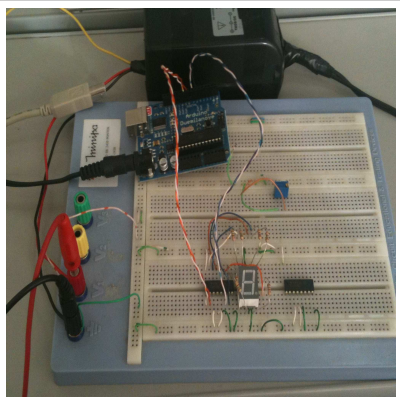
Conexão do CI, resistores e display de 7 segmentos. Ficou uma aranha. :)



Placa Arduino, responsável por enviar 4 bits em formato BCD.



Visão geral do projeto, protoboard, fonte, Arduino e componentes.



Programação:

A programação para este projeto é bem simples, defino 4 variáveis e atribuo a elas o endereço das portas de IO do Arduino. Em seguida na função Setup faço mais algumas configurações e então defino a função number que recebe um número como parâmetro. Baseado neste número as portas de saída são configuradas para enviar um valor BDC adequado.

Dentro da função loop crio 2 laços para fazer a contagem de forma crescente e depois decrescente. Bem simples, veja abaixo o código.

```
/*
7 Segments

This project uses the CD4511 BCD to 7segments Latch Decoder
Driver
to count the numbers from 0 - 9 and after 1 second from 9 - 0.

Created 25 June 2010
By Diego Antunes
*/

// Atribui a cada variável um endereço.
int pinA = 6, pinD = 7, pinC = 8, pinB = 9;

void setup()
{
  // Configura a velocidade da porta serial
  Serial.begin(9600);

  // Diz ao controlador que os pinos são de saída
  pinMode(pinA, OUTPUT);
  pinMode(pinB, OUTPUT);
  pinMode(pinC, OUTPUT);
  pinMode(pinD, OUTPUT);
}

// Recebe um inteiro como parâmetro
// Baseado no número configura os pinos de saída conforme da tabela
BCD
// do CI CD4511
void number(int num)
{
  switch (num) {
    case 0:
      digitalWrite(pinA, LOW);
      digitalWrite(pinB, LOW);
      digitalWrite(pinC, LOW);
      digitalWrite(pinD, LOW);
      break;
    case 1:
      digitalWrite(pinA, HIGH);
      digitalWrite(pinB, LOW);
      digitalWrite(pinC, LOW);
      digitalWrite(pinD, LOW);
      break;
    case 2:
      digitalWrite(pinA, LOW);
      digitalWrite(pinB, HIGH);
      digitalWrite(pinC, LOW);
      digitalWrite(pinD, LOW);
      break;
```

```
case 3:
digitalWrite(pinA, HIGH);
digitalWrite(pinB, HIGH);
digitalWrite(pinC, LOW);
digitalWrite(pinD, LOW);
break;
case 4:
digitalWrite(pinA, LOW);
digitalWrite(pinB, LOW);
digitalWrite(pinC, HIGH);
digitalWrite(pinD, LOW);
break;
case 5:
digitalWrite(pinA, HIGH);
digitalWrite(pinB, LOW);
digitalWrite(pinC, HIGH);
digitalWrite(pinD, LOW);
break;
case 6:
digitalWrite(pinA, LOW);
digitalWrite(pinB, HIGH);
digitalWrite(pinC, HIGH);
digitalWrite(pinD, LOW);
break;
case 7:
digitalWrite(pinA, HIGH);
digitalWrite(pinB, HIGH);
digitalWrite(pinC, HIGH);
digitalWrite(pinD, LOW);
break;
case 8:
digitalWrite(pinA, LOW);
digitalWrite(pinB, LOW);
digitalWrite(pinC, LOW);
digitalWrite(pinD, HIGH);
break;
case 9:
digitalWrite(pinA, HIGH);
digitalWrite(pinB, LOW);
digitalWrite(pinC, LOW);
digitalWrite(pinD, HIGH);
break;
default:
digitalWrite(pinA, HIGH);
digitalWrite(pinB, HIGH);
digitalWrite(pinC, HIGH);
digitalWrite(pinD, HIGH);
}
}

void loop()
{
// Faz um loop de 0 a 9
for (int i = 0; i < 10; i++) {
// Imprime o valor desejado
number(i);
// Aguarda meio segundo
delay(500);
}

// Apaga o número, percebe que quando chamo a função number
// com um valor <0 ou >9 a função simplesmente apaga todos
// os leds do contador. Por isso, se chamo a função com o
```

```
// valor 10 simplesmente o display se apagará.  
number(10);  
// Aguarda 1 segundo  
delay(1000);  
  
// Faz um loop de 9 a 0  
for (int i = 10; i >= 0; i--) {  
  // Imprime o valor desejado  
  number(i);  
  // Aguarda meio segundo  
  delay(500);  
}  
}
```

Para visualizar o resultado final deste projeto clique [aqui](#).

Dúvidas, comentários e sugestões são bem vindos.

Abraço, Diego

Postado por Imbrax - Lab às [12:10](#) [0 comentários](#)

terça-feira, 1 de junho de 2010

Divirta-se vigiando os amigos.

Apresentação:

Depois que montei o projeto do potenciômetro controlando o servo motor na semana passada fiquei curioso para testar o servo com outros elementos. Quem nunca pensou em controlar o ponto de foco de uma webcam? Bom, neste projeto iremos montar uma base para colocar uma webcam e ligaremos a base ao motor e o motor a placa Arduino.

Como gosto de trabalhar com sucata (objetos que parecem já sem utilidade mas que podem ser retrabalhados e reutilizados) pensei neste projeto.

Para este projeto precisaremos de:

- 1 Servo motor;
- 1 WebCam (qq modelo simples serve);
- Alguns jumpers ou fios (neste projeto usei o fios de um cabo par trançado que não tinha mais utilidade);
- Pedacos de chapas de acrílico ou plástico;
- Alguns parafusos;
- Chave de fenda ou philips;
- Estilete;
- Pistola de cola quente;
- Alicates pequena;

O que faremos?

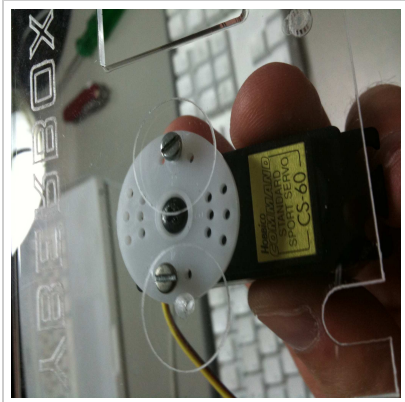
O objetivo do projeto é usar a placa Arduino para controlar o movimento de um servo motor que tem seu eixo acoplado a uma webcam.

Como ligar os acessórios?

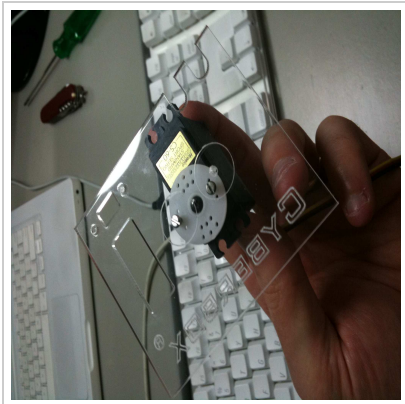
Esse projeto é mais simples do que o anterior, basta ligar o cabo USB na placa Arduino e depois no computador. O servo motor possui três fio, VCC, GND e Signal. Ligue cada um deles como foi feito no exemplo anterior.

Montagem:

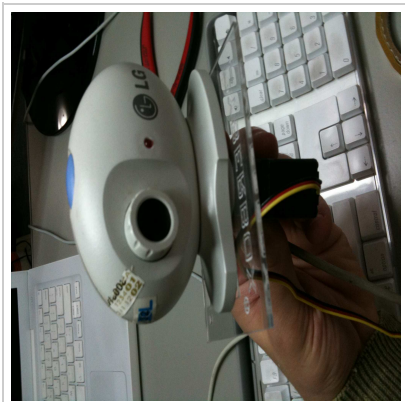
Para começar peguei um pedaço de acrílico que estava sobrando, fiz dois furos e parafusei na roda ligada ao eixo do servo.



Mais uma foto mostra como ficou a base.



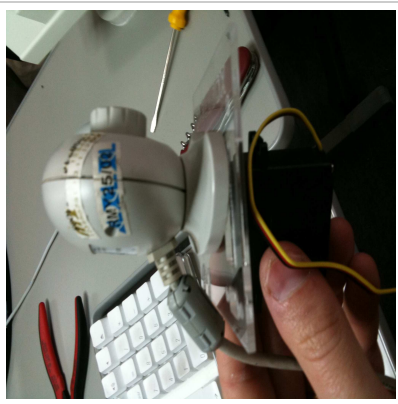
Fiz mais 2 novos furos e parafusei a base da webcam na base de acrílico, foi fácil pois havia dois parafusos na base da webcam que substitui por maiores.



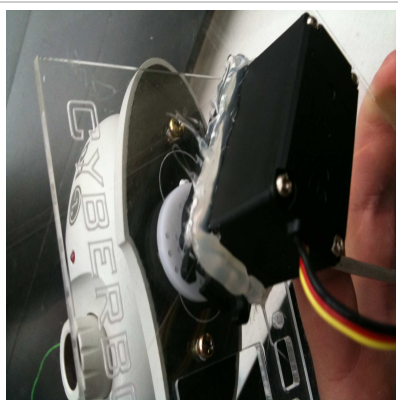
Visão da base da webcam conectada a base de acrílico.



Mais uma. :)



Agora preciso fixar o servo motor em alguma coisa pesada. Como estou trabalhando com sucata, usei um pedaço de uma caixa de acrílico que não seria mais utilizada. Removi um retângulo do mesmo tamanho do servo e coloquei o servo ali. Para fixar bem coloquei um pouco de cola quente. Pronto, ficou bem preso.



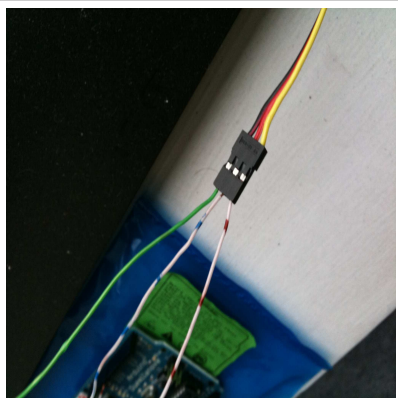
Aqui esta o resultado final!



Mais uma, agora lateralmente.



Agora uma imagem que mostra a conexão dos fios do servo na placa Arduino. Como não tinha um fio preto usei o verde para simbolizar o GND ou Terra. O branco com azul é o VCC ou Força e o branco com marrom é o fio que envia o sinal para o servo.



Na outra ponta, temos os fios conectados nos conectores GND (fio verde), VCC 5V (fio branco com azul) e I/O PWM 9 (fio branco com marrom)



É muito fácil mesmo.

Programação:

Para controlar o servo precisamos programar o Arduino, peguei como exemplo o código para controle de servo chamado Sweep. Neste código o servo vai para a posição 0 e depois vai se movemndo lentamente até 180. Depois volta para o 0 lentamente, isso num laço infinito. A única diferença no meu código foi que aumentei o tempo de delay, troquei para 40ms assim o eixo do servo motor se moverá um pouco mais devagar. No site do Arduino você pode encontrar o original <http://arduino.cc/en/Tutorial/Sweep>.

```
// Sweep
// by BARRAGAN

#include

Servo myservo; // cria um objeto servo
// um máximo de 8 objetos servo podem ser criados.

int pos = 0; // variável para guardar a posição do servo

void setup()
{
  myservo.attach(9); // anexa o servo ao pino 9. Isso diz para o
  objeto servo
  // onde o pino de sinal do servo esta conectado
}

void loop()
{
  for(pos = 0; pos < 180; pos += 1) // vai de 0 até 180 graus em
  passos de 1 grau.
  {
    myservo.write(pos); // envia para o servo a posição contida na
    // variável 'pos'
    delay(40); // espera 40ms para que o servo chegue na posição
  }
  for(pos = 180; pos >= 1; pos -= 1) // vai de 180 até 0 graus
  {
    myservo.write(pos); // envia para o servo a posição contida na
    // variável 'pos'
    delay(40); // espera 40ms para que o servo chegue na posição
  }
}
```

Baixe o código para a sua placa Arduino e pronto, o servo motor deve ir para a posição 0 e depois vai girando o eixo com intervalo de tempo de 40ms e incrementos de 1 grau.

Fiz um pequeno video que demonstra este exemplo clique [aqui](#).

Dúvidas, comentários e sugestões são bem vindos.

Abraço, Diego

Postado por Imbrax - Lab às [07:56](#) [2 comentários](#)

Marcadores: [imbrax arduino servo motor sucata programação eletrônica](#)

quinta-feira, 27 de maio de 2010

Servo Motor com Arduino

Apresentação:

Controlar um servo motor utilizando a placa Arduino 2009 é muito fácil! Neste post realizaremos um projeto muito simples mas que mostra a capacidade e as possibilidades do arduino.

Bem, para este pequeno projeto precisaremos de:

- 1 Protoboard (pode ser qualquer uma mesmo, não importa a quantidade de pontos);
- 1 Servo motor;
- 1 Potenciômetro de 10K;
- Alguns jumpers ou fios (neste projeto usei o fios de um cabo par trancado que não tinha mais utilidade);

O que faremos?

Esse projeto esta descrito no site do arduino em <http://arduino.cc/en/Tutorial/Knob> e exemplifica como controlar a posição de um servo motor a partir da mudança de resistência de um potenciômetro.

Isso mesmo, você vai girar o eixo do potenciômetro (isso vai mudar a resistência interna) e o valor da resistência será lido pelo conversor analógico-digital do micro controlador na placa arduino. Com base no valor digital da resistência o código dentro do arduino irá mapear a posição do eixo do servo motor.

Como ligar os acessórios?

Como sempre, ligue a sua placa arduino ao seu computador utilizando o cabo USB e abra o programa de desenvolvimento/gravação do arduino. No menu (Tools ou Ferramentas) configure a sua placa (a minha é uma arduino 2009 com micro controlador ATmega328) e agora configure a porta serial de acordo com as portas disponíveis na sua máquina.

Ainda não codificaremos, aguarde um pouco.

Servo motores possuem 3 fios: Alimentação, terra e sinal. Consulte a folha de dados do seu servo para conhecer as tensões que ele suporta sendo que normalmente os servos trabalham com 5 ou 6V.

Potenciômetros também possuem normalmente 3 fios, os pinos externos serão ligados em VCC e GND e não se preocupe que não há uma orientação padrão tanto faz ligar um ou outro em VCC e GND.

Montagem:

Para ficar mais claro eu fiz assim:

Peguei um jumper e liguei uma das pontas do fio na placa arduino onde diz 5V, a outra ponta liguei em uma trilha da protoboard. Peguei outro fio e liguei uma das pontas na placa arduino onde diz GND e a outra ponta em outra trilha. Fiz isso para trazer os sinais de tensão de 5 e 0V para a protoboard.

Agora pegue o cabo do servo, utilizando outro jumper ligue uma das pontas do fio no cabo vermelho do servo, ele normalmente indica alimentação. Ligue a outra ponta do fio na protoboard na trilha onde temos o sinal de 5V.

Execute o mesmo processo para o fio preto do servo, utilize um jumper para ligar este fio na trilha GND ou 0V na protoboard.

Agora para finalizar com o servo ligue um jumper no conector onde temos o fio amarelo do servo, a outra ponta deverá ser conectada na porta 9 do arduino.

Se o seu potenciômetro não possui cabos soldados aos seus contatos pegue 3 pedaços de fio do cabo de rede e conecte nos contatos, não precisa soldar mas prenda bem firme. Use uma alicate de bico para fazer isso que é mais fácil.

Ligue um dos fios dos contatos externos do potenciômetro na trilha VCC da protoboard, ligue o outro fio do contato externo na trilha GND onde conectamos o fio preto do servo. Agora ligue o fio conectado ao contato do meio na porta de entrada 0 no seu arduino.

Pronto agora tudo esta conectado corretamente!

Programação:

No menu Arquivo->Examples->Servo selecione o exemplo Knob. Esse exemplo terá o código abaixo:

```
#include
```

```
Servo myservo; // cria um objeto servo para controlar um servo
```

```
int potpin = 0; // pino analógico utilizado para ler os dados do potenciômetro
```

```
int val; // variável que armazenará os valores digitais do pot.
```

```
void setup()
```

```
{  
myservo.attach(9); // diz para o objeto servo que um servo motor  
esta conectado no pino 9  
}
```

```
void loop()
```

```
{  
val = analogRead(potpin); // lê os valores do pot (entre 0 e 1023)  
val = map(val, 0, 1023, 0, 179); // mapeia os valores para controlar  
o servo  
// valores entre 0 e 180 (graus)  
myservo.write(val); // envia a posição para o servo  
delay(15); // aguarda o servo chegar na posição  
}
```

Baixe o código para a sua placa arduino e pronto.

Agora gire o eixo do potenciômetro para ver o servo motor sendo controlado.

Fiz um pequeno video que demonstra este exemplo clique [aqui](#).

Dúvidas, comentários e sugestões são bem vindos.

Abraço, Diego

Postado por Imbrax - Lab às [14:43](#) [2 comentários](#)

Marcadores: [arduino](#) [servo](#) [motor](#) [potenciômetro](#) [imbraxlab](#)

segunda-feira, 24 de maio de 2010

Testando 1 2 3... testando.

Primeiro post no blog, sejam todos bem vindos.

Postado por Imbrax - Lab às [10:37](#) [0 comentários](#)

[Início](#)

Assinar: [Postagens \(Atom\)](#)