

www.lusorobotica.com

Muitos de vocês perguntam-se:

"Como devo de começar na robótica?"

"Que irei eu comprar para começar a fazer uma coisa simples, sem gastar muito dinheiro?"

"Será que o que vou comprar será fácil o suficiente para começar, mas com potencial para depois continuar evoluir e não ficando estas peças por fim totalmente inúteis e colocadas de lado?"

Pois bem, para ir em conta às vossas questões decidi então criar o Smalluino.

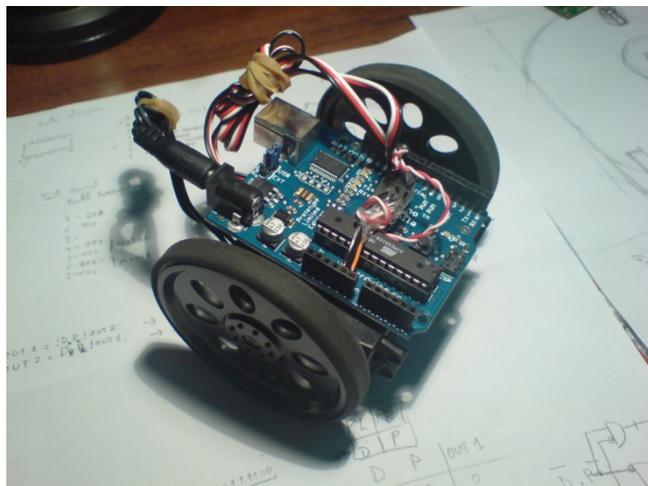


Fig.1 – Versão Final do Smalluino Montado

O Smalluino, é baseado no Arduino (dai o seu nome) e basicamente é o mínimo dos mínimos para se fazer algo que se pareça com um robô. Digo isto porque no fundo ele não é bem um robô, mas sim um autómato programável, visto que o Smalluino não recebe inputs do mundo que o rodeia para poder tomar decisões, logo não é autónomo, mas já é um passo para um futuro robot.

No entanto, este projecto faz uma boa iniciação do Arduino para a robótica, visto com este pequeno tutorial ficarem a saber controlar um enorme número de componentes que vos irão ser necessários um dia, tal como os motores servos, fazer código no Arduino recorrendo a métodos para otimizar este, e ter um primeiro contacto com os dispositivos autónomos.

Uma das coisas mais importantes quando desenvolvemos robôs, e que deixo aqui como uma recomendação, é termos feito um plano, um desenho, algo que nos ajude a visualizar os possíveis problemas com que nos vamos deparar para podermos corrigi-los ou evita-los antes de surgirem.

Tendo em conta isto, passemos à prática!

Smalluino:

O Smalluino em si, é composto por 4 componentes de que vão precisar em qualquer projecto de robótica:

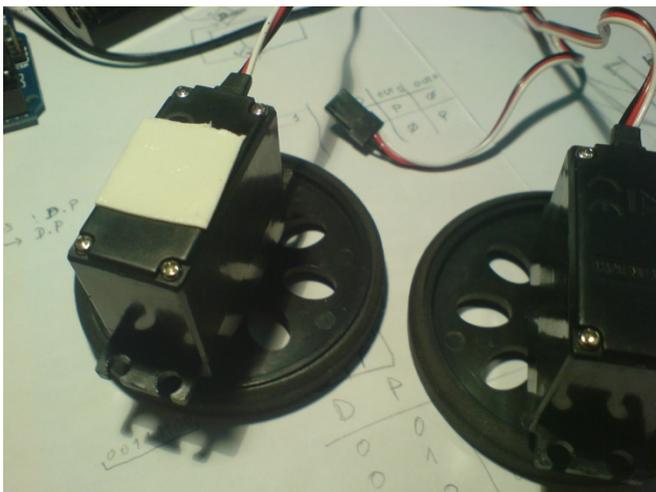
1xMicrocontrolado (neste caso o Arduino);

1xSuporte de Bateria (utilizamos 4 pilhas recarregáveis num suporte bastante comum;

2xMotores (recorremos aos servos por serem bastante fáceis de utilizar, e úteis em qualquer projecto de robótica);

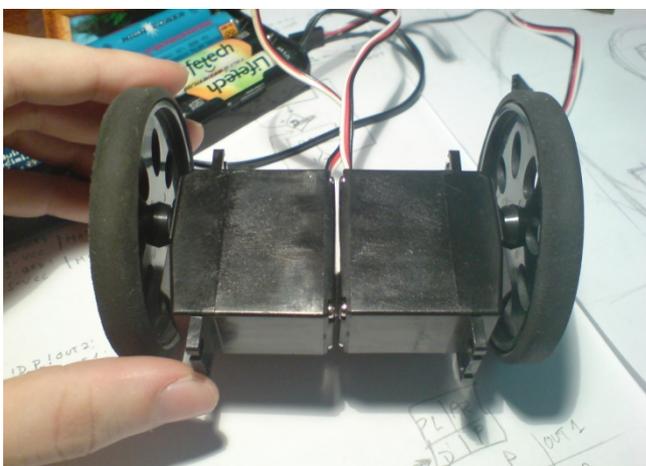
2xRodas (rodas acopladas aos motores que permitam que este se desloque.

Passemos então à montagem, passo por passo:



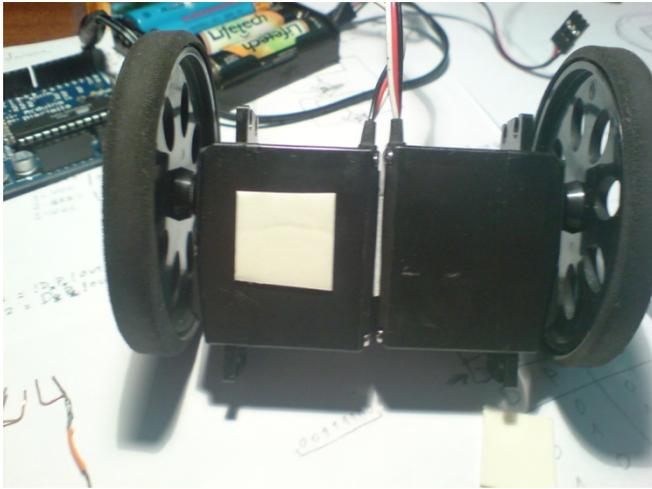
Passo 1:

Começamos por colocar um pouco de fita dupla (cola dos dois lados) em cada um dos motores com as rodas previamente acopladas montadas.



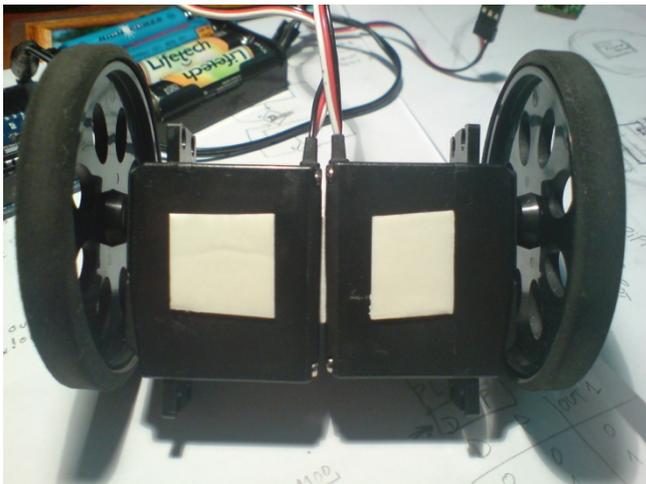
Passo 2:

Após colocar a fita, vamos uni-los de forma a ficar uma pequena plataforma.



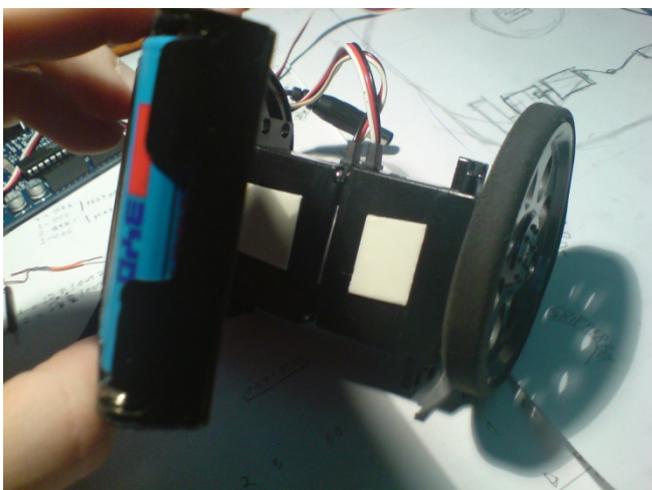
Passo 3:

Após bem unidos, iremos colocar um pouco da mesma fita no motor esquerdo.



Passo 4:

O mesmo se passará com o motor direito.



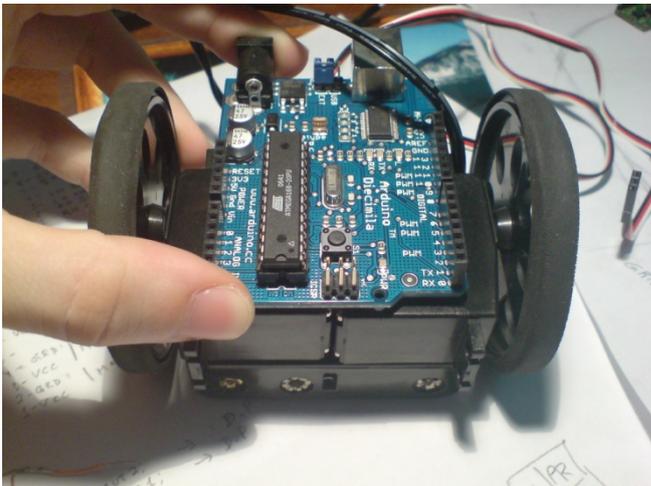
Passo 5:

Após colocarmos a fita, iremos colocar nos motores o suporte de bateria (neste caso alimentado por 4 pilhas AA)



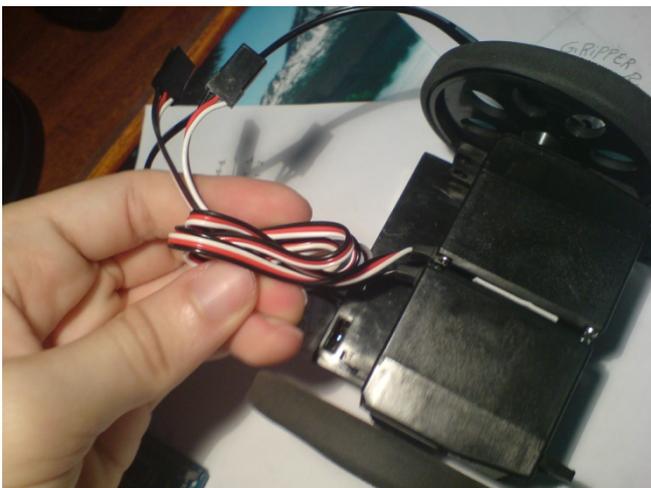
Passo 6:

Certifiquem-se que na parte frontal o suporte da bateria fica à face dos motores, este passo é importante para o próximo projecto que será um upgrade ao smalluino. Este será o aspecto que o vosso Smalluino deverá ter após executarem os passos anteriores.



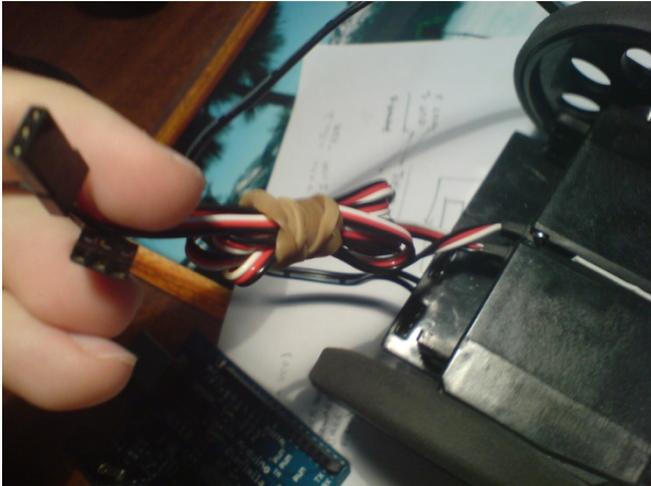
Passo 7:

Tempo agora de colocar o Microcontrolador, vamos colocá-lo da mesma maneira que fizemos para montar o suporte da bateria. (reparem, os conectores da bateria e da porta USB estão no mesmo lado em que se encontram os fios dos motores, e da bateria)



Passo 8:

Para facilitar a futura ligação dos cabos dos motores, vamos enrolá-los de maneira a ficarem mais curtos.



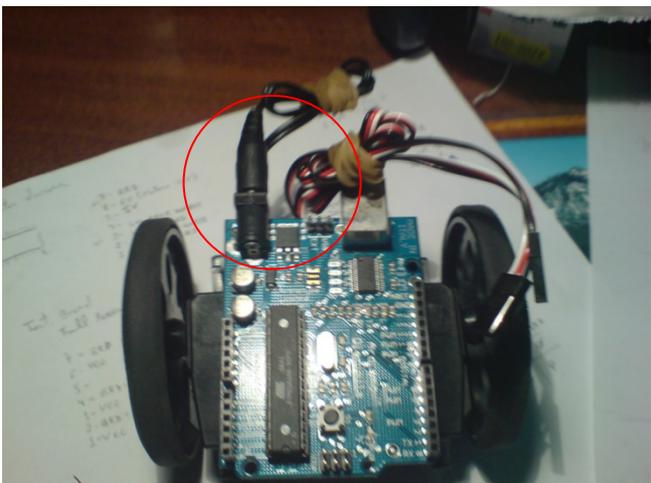
Passo 9:

Após enrolarmos os cabos, vamos colocar um elástico para que se mantenham fixos e seja mais fácil manobrar.



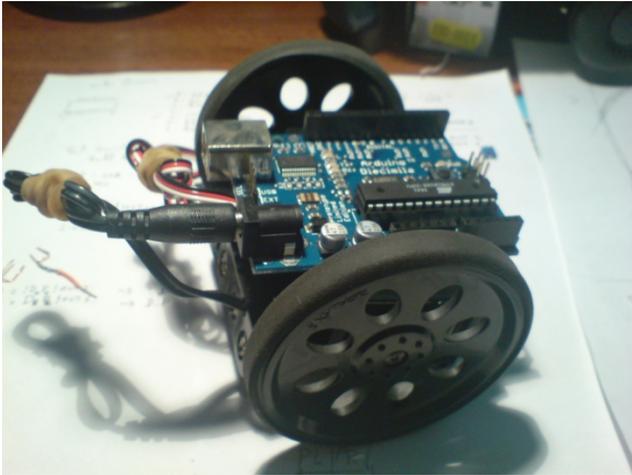
Passo 10:

Faremos o mesmo de seguida para o cabo que vêm do suporte para as baterias.



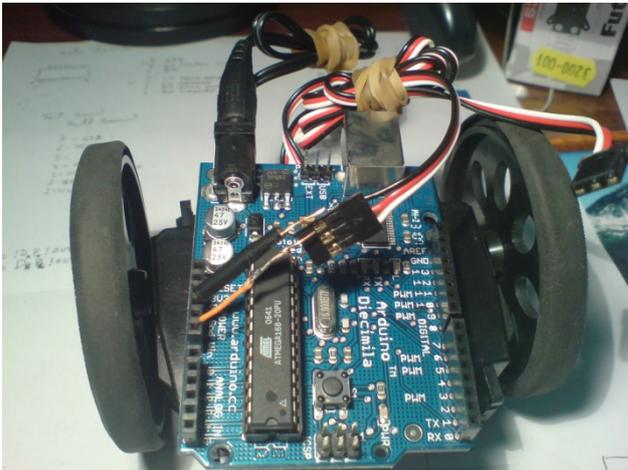
Passo 11:

Ligaremos agora ao Smalluino o cabo que futuramente lhe irá dar a energia fornecida pelas baterias, para este funcionar.



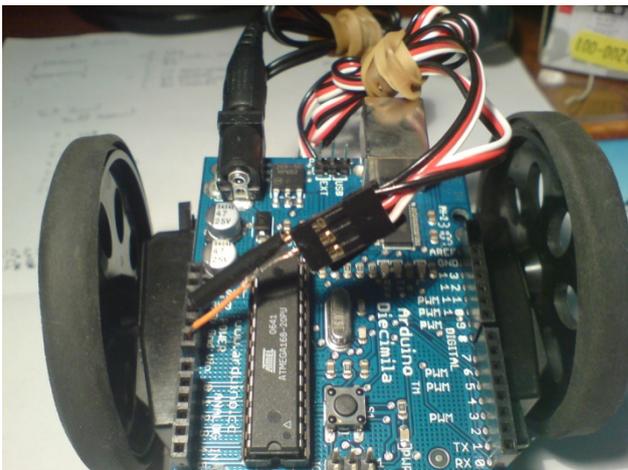
Passo 12:

Assim será o aspecto final do vosso Smalluino, após enrolarem os cabos e colocarem os respectivos elásticos.



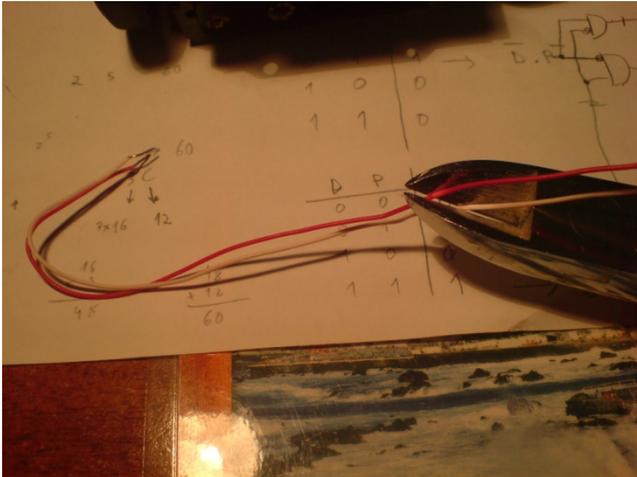
Passo 13:

Passamos então às ligações, iremos fazer a ligação dos motores com o Arduino, ligando assim ambos os fios pretos ao GRD, e os fios vermelhos ao VIN, ficará com o aspecto da imagem a baixo.



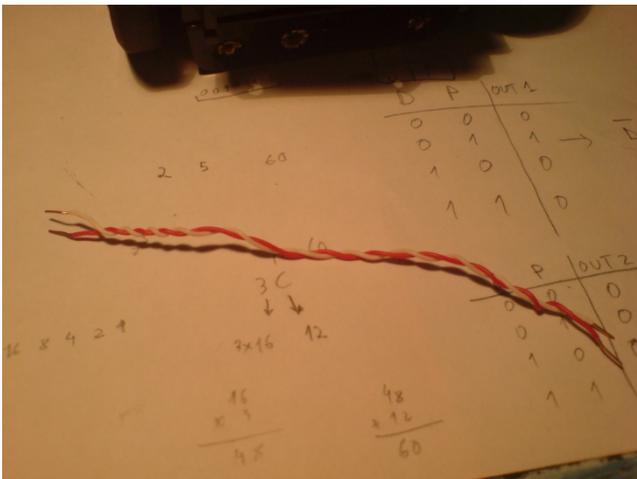
Passo 14:

Depois dos dois motores ligados à alimentação, falta-nos apenas ligá-los aos fios de sinal, afim de receberem do controlador as ordens durante o decorrer do código.



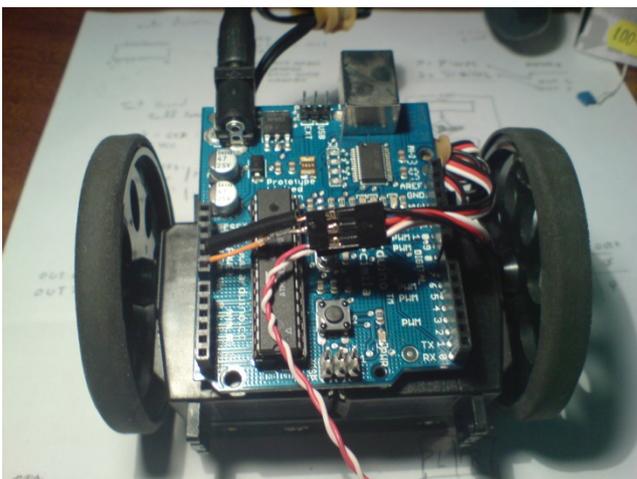
Passo 15:

Vamos agora cortar os fios que iremos usar futuramente para ligar os motores entre os fios de sinal (brancos), e o Arduino.



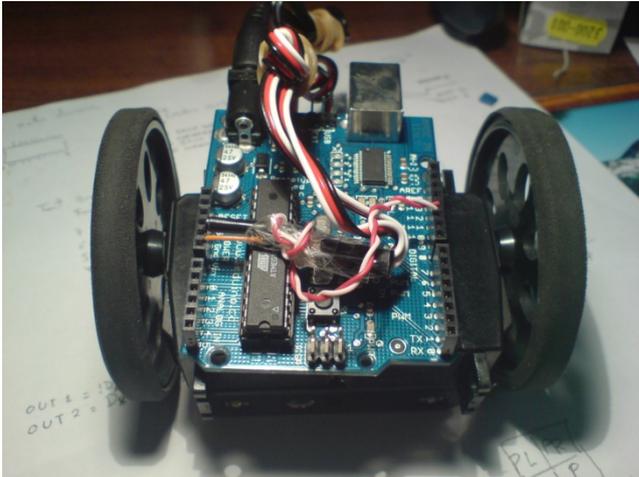
Passo 16:

Pode enrolar os fios em torno deles próprios, permitindo assim uma mais fácil organização dos fios, visto que estes tem um início e um destino muito próximos um do outro.



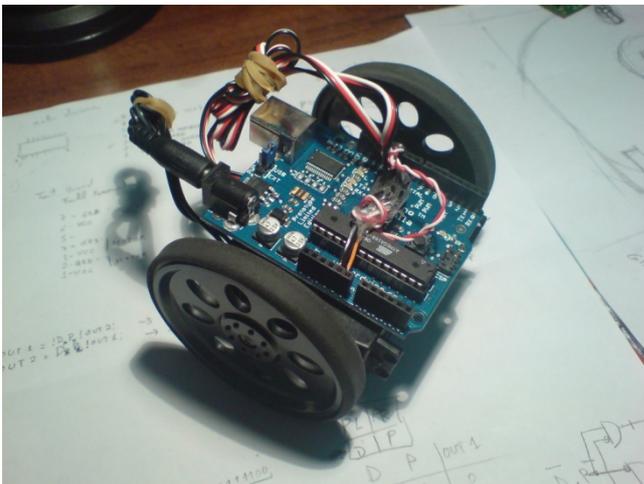
Passo 17:

Ligando os fios feitos anteriormente aos motores (fio branco) ficará com este aspecto.



Passo 18:

Ligamos a seguir estes dois fios ao pin 10 e 11 do Arduino, e o vosso Smalluino terá este aspecto visto de cima. (foi colocada fita-cola para prender os fios, utilizem o que vos apetecer, mas lembrem-se de evitar cola ou outro tipo de contactos permanentes)



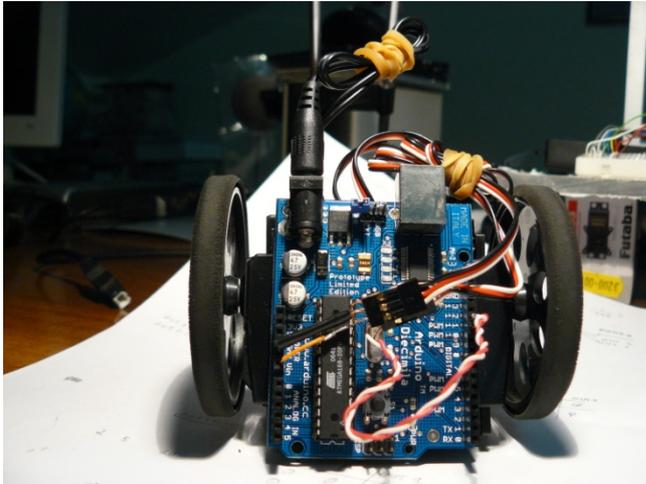
Passo 19:

O vosso Smalluino visto de perfil.



Passo 20:

O vosso Smalluino visto de frente.



Passo 21:

Aqui está o nosso Smalluino, montado e pronto a ser programado.

Explicação:

“Os servos motores, têm 3 fios, 2 de alimentação e um de sinal. Normalmente Vermelho, Preto e Branco, para positivo, negativo e sinal, respectivamente.

Assim, ligamos o Vermelho dos dois servos ao pino Vin (Voltage IN) que é a voltagem que vem directamente das alimentação exterior, neste caso as pilhas que fazem um total de cerca de 5 ~ 6V (atenção nunca ligar um servo a mais de 6V a menos que venha indicado no datasheet do mesmo que este suporta tensões superiores).

O pin branco de cada um dos servos foram ligados ao pin 10 e 11, estes pins porque permitem melhor utilização do sinal PWM do Arduino, como pode ser lido na documentação do mesmo em www.arduino.cc, permitindo assim que sejam programados individualmente para seguirem num dado sentido a uma velocidade.”

Após a nossa montagem e de estar tudo ligado, está na altura de metermos o Smalluino a andar. Para tal temos de abrir o compilador do Arduino (disponível para download em www.arduino.cc).

O Arduino como também já foi explicado em outros tutoriais, é programado através da linguagem C/C++, sendo linguagens de comum utilização e conhecimento. De seguida será apresentado um algoritmo de exemplo e a referida explicação do mesmo.

Visitem o nosso site, lá terão também um exemplo filme elucidativo ao código que a seguir irá ser discutido e a toda a montagem que anteriormente foi explicada.

Um abraço e até ao próximo Tutorial.

O Teu Primeiro Programa:

```
int motorEsq = 11;
int motorDrt = 10;
void setup()
{
  pinMode(motorEsq, OUTPUT);
  pinMode(motorDrt, OUTPUT);
}
void loop()
{
  parado(4);
  frente(2);
  esquerda(6);
  tras(2);
  esquerda(6);
  direita(6);
}
void parado(int tempo){ // tempo = segundos/2
  for(int i = 0; i<tempo;i++){
    analogWrite(motorEsq, 127);
    analogWrite(motorDrt, 127);
    delay(500);
  }
}
void tras(int tempo){ // tempo = segundos/2
  for(int i = 0; i<tempo;i++){
    analogWrite(motorEsq, 152);
    analogWrite(motorDrt, 102);
    delay(500);
  }
}
void frente(int tempo){ // tempo = segundos/2
  for(int i = 0; i<tempo;i++){
    analogWrite(motorEsq, 102);
    analogWrite(motorDrt, 152);
    delay(500);
  }
}
void esquerda(int tempo){ // tempo = segundos/2
  for(int i = 0; i<tempo;i++){
    analogWrite(motorEsq, 102);
    analogWrite(motorDrt, 102);
    delay(500);
  }
}
void direita(int tempo){ // tempo = segundos/2
  for(int i = 0; i<tempo;i++){
    analogWrite(motorEsq, 152);
    analogWrite(motorDrt, 152);
    delay(500);
  }
}
```

Explicando o código:

Depois de tudo ligado, abrimos o compilador do Arduino e fizemos um simples código em C, este define os métodos que vão ser chamados no código inicial, e executaram a devida instrução.

O tempo de paragem dentro dos ciclos foi definido para 500ms, o que faz com que 2 ciclos sejam equivalentes a 1 segundo.

Se quisermos que o smalluino fique a andar 3 segundos em frente temos que chamar o método passando o parâmetro 6 (3s x 2 (500ms cada execução)).

Se no código frente o vosso smalluino se encontrar a andar para traz, troquem no inicio do código o pin 11 com o pin 10, evitando estar a trocar estes fios no robô.

Bem, só me resta desejar-vos um bom projecto, e que façam modificações no código para irem aprendendo a utiliza-lo.

Não Percam o Próximo Tutorial:

Brevemente iremos agarrar no mesmo projecto e adicionar-lhe um sensor na parte frontal, foi por isso que deixamos a frente sem fios, para podermos depois colocar lá sensores de proximidade com obstáculos, e dando assim capacidade ao smalluino de se deslocar de modo autónomo, sendo ele próprio a tomar decisões de por onde seguir evitando assim os obstáculos que lhe vão surgindo pelo caminho.

Agradecimentos:

Ao AngellS pela ajuda na edição gráfica do projecto, e ao meu gato pelos testes de resistência do Smalluino.