

TUTORIAL ROBÔ BÍPEDE

Autores: Carlos Magno Feitosa da Silva, Leandro Borini Lone, Luís Fernando Patsko
Nível: Avançado
Criação: 26/10/2006
Última versão: 19/12/2006



Maxwell Bohr
Instrumentação Eletrônica

<http://www.maxwellbohr.com.br>
contato@maxwellbohr.com.br

PdP

Pesquisa e Desenvolvimento de Produtos

<http://www.automato.com.br>
atendimento@automato.com.br

1 – Introdução

A robótica é uma área cujo desenvolvimento ao longo dos próximos anos será relevante. O uso de robôs em trabalhos que sejam perigosos ou repetitivos (onde a indústria automobilística é um exemplo da presença da robótica na economia global), na área de entretenimento (com os robôs japoneses que imitam animais) ou até mesmo em serviços domésticos (o aspirador de pó autônomo é um grande exemplo) já é cada vez mais intenso.



Figura 1: Humanóide Asimo, desenvolvido pela Honda.

O projeto a ser desenvolvido ao longo desse tutorial é um bípede, ou seja, um pequeno robô que possui duas “pernas”, que é conectado ao Kit podendo ser controlado através de um programa rodando num computador. Através da criação de um robô bípede, é possível aplicar os conhecimentos de Eletrônica, Mecânica e Programação adquiridos ao longo do curso num projeto prático. Além disso, esse tutorial serve como uma ótima base para futuros projetos.

Um fator importante que não pode ser negligenciado durante a montagem é a segurança. É indispensável o uso de equipamentos de proteção individual (EPI's) tais como luvas, óculos de

proteção, jaleco, etc. Também é essencial tomar cuidados básicos para a prevenção de acidentes, especialmente durante a manipulação de produtos químicos e ferramentas. Por pura falta de atenção ou excesso de confiança, podem ser causados acidentes graves. E, além disso, os problemas decorrentes da postura inadequada e da repetição excessiva de movimentos são vários e é muito simples tomar providências para evitá-los. Não é difícil se conscientizar e realizar pequenas ações que podem prevenir muitos acidentes.

2 – Funcionamento

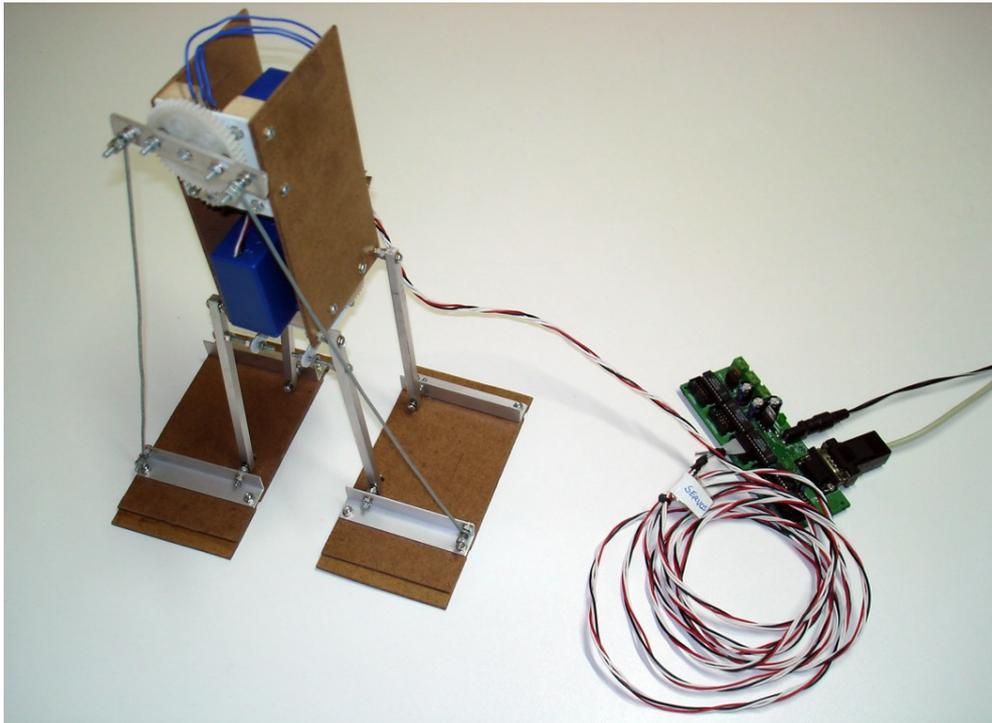


Figura 2: Visão do bípede pronto, ligado ao MEC1000.

O robô bípede é um pequeno robô que possui duas pernas, que pode ser controlado pelo usuário através de um programa rodando num computador. Ele possui dois servo-motores, que serão utilizados na sua locomoção, que se dará da seguinte forma: um dos servo-motores é destinado a inclinar o corpo do bípede para o lado. Desse modo, seu peso fica concentrado sobre uma perna só, possibilitando que a outra seja deslocada. Esse deslocamento é realizado pelo segundo servo-motor. Então o primeiro servo-motor entrará em ação novamente, jogando o peso do bípede para a perna que acabou de ser deslocada, permitindo a movimentação da outra. Para controlar o sentido do movimento do bípede, é necessário inverter a rotação do servo-motor responsável por deslocar as pernas do bípede, o que pode ser feito nas linhas de código do programa. Também podemos fazer com que o bípede vire, bastando que uma das pernas se desloque menos que a outra.

O nosso projeto é constituído por duas partes principais: a construção do robô e a programação necessária para seu funcionamento. A construção do robô é a elaboração da estrutura mecânica e eletrônica. A dedicação nessa etapa é indispensável, uma vez que peças ou circuitos mal

feitos podem comprometer o funcionamento de todo o sistema. Mas não basta apenas a estrutura estar perfeita, pois ela deve ser controlada adequadamente. A programação será responsável por proporcionar a conexão entre o Kit e o robô e pelo controle preciso da atuação dos servo-motores, de modo que o robô possa se movimentar corretamente, pois cada um desses componentes deve ser acionado num momento exato.

Seguindo corretamente as instruções a seguir, podemos ter em mãos um pequeno robô bípede. Mas nada o impede de usar a criatividade e realizar modificações e implementar o projeto com outras características, ou seja, construir o projeto ao seu gosto.

3 – Materiais

Os materiais utilizados nesta montagem são madeira tipo MDF, perfis de alumínio (barras e cantoneiras), madeirite, plástico (PVC), engrenagens de acrílico, dois servo-motores (adaptados para rotação contínua, mas com potenciômetros externos), parafusos de diversos tamanhos, porcas, arruelas, cola branca, cola de secagem instantânea e lubrificante. Além disso, serão necessárias ferramentas para a montagem tais como paquímetro, caneta de retroprojektor, serra de meia-esquadria. Tais ferramentas estarão à disposição nas bancadas de trabalho.

Optamos pelos materiais utilizados pela acessibilidade, baixo custo, resistência adequada, e também por serem fáceis de trabalhar, tentando sempre adequá-los às dificuldades surgidas no desenvolvimento, ou até substituindo-os por outros se necessário, lembrando-se sempre de que os materiais listados aqui, de forma alguma são padrão para as montagens, e que podem ser substituídos tranquilamente por outros semelhantes ou que desempenhem a mesma função. Deve-se apenas ter em mente que nem todos os materiais têm desempenho semelhante sob uma determinada situação. Por exemplo, a madeira MDF é uma boa escolha, pois é fácil de trabalhar e não trinca tão facilmente como uma chapa de compensado, por exemplo, quando é perfurada.

4 – Leituras prévias

Para a construção do projeto descrito nesse tutorial, é recomendada a leitura de outros tutoriais, que nos auxiliarão durante a construção da esteira. Os tutoriais recomendados são: *Montagem do Bloco de Servo-motor Padrão*, *Adaptação de Servo-motores* e *Criação de engrenagens em acrílico*. Esses tutoriais estão à disposição ao longo do curso de Mecatrônica e Robótica e serão utilizados ou darão suporte a alguns procedimentos ao longo deste tutorial.

5 – Montagem

O primeiro passo para a montagem de um projeto deve ser a elaboração de um esboço deste num papel, mesmo que se tenha todo projeto em mente. Isto facilitará muito a montagem de qualquer projeto, pois além de uma melhor visualização deste, não corremos o risco de esquecermos ou pularmos algum passo.

Com o “roteiro” do projeto em mãos, podemos começar a montagem escolhendo os materiais que mais se adequem as montagens de cada parte do projeto, enfatizando principalmente a acessibilidade dos materiais, o custo, para que o projeto não venha a ter um custo muito elevado, e maleabilidade. Por exemplo: perfis de alumínio podem ser encontrados em qualquer loja especializada em alumínio, são relativamente baratos e muito mais fáceis de trabalhar do que perfis de ferro, por exemplo.

O projeto descrito nesse tutorial visa a construção de um robô bípede, fazendo uso de servo-motores e estruturas mecânicas. A parte principal no que se refere ao bom funcionamento desse robô bípede é a correta construção dos blocos dos servo-motores. Esses servo-motores deverão ser adaptados para rotação contínua, mas devem utilizar um potenciômetro externo. Este potenciômetro tem exatamente a mesma função do potenciômetro original do servo-motor, só que agora o servo vai transferir seu movimento para uma outra engrenagem, e ela deverá ser a referência para o controle do servo-motor. Por isso, essa engrenagem será fixada ao potenciômetro, que será o eixo principal do bloco. Deste modo, conseguiremos controlar precisamente a posição da engrenagem. Esta engrenagem faz parte de um conjunto redutor, de modo que o torque disponível seja maior, para possibilitar uma correta movimentação do bípede.

Primeiramente, devemos confeccionar engrenagens de acrílico utilizando os moldes de silicone. As engrenagens utilizadas serão de tamanhos diferentes, sendo que a menor deve ser fixada no eixo do motor e a maior ao potenciômetro que foi adicionado ao bloco do servo-motor, pois a idéia é de reduzir ainda mais a velocidade do motor e conseqüentemente aumentar seu torque. Depois de confeccionadas, iremos tomar a distância entre seus centros. Esta medição nos informará qual a distância em que o potenciômetro deve ficar em relação ao eixo do servo-motor.

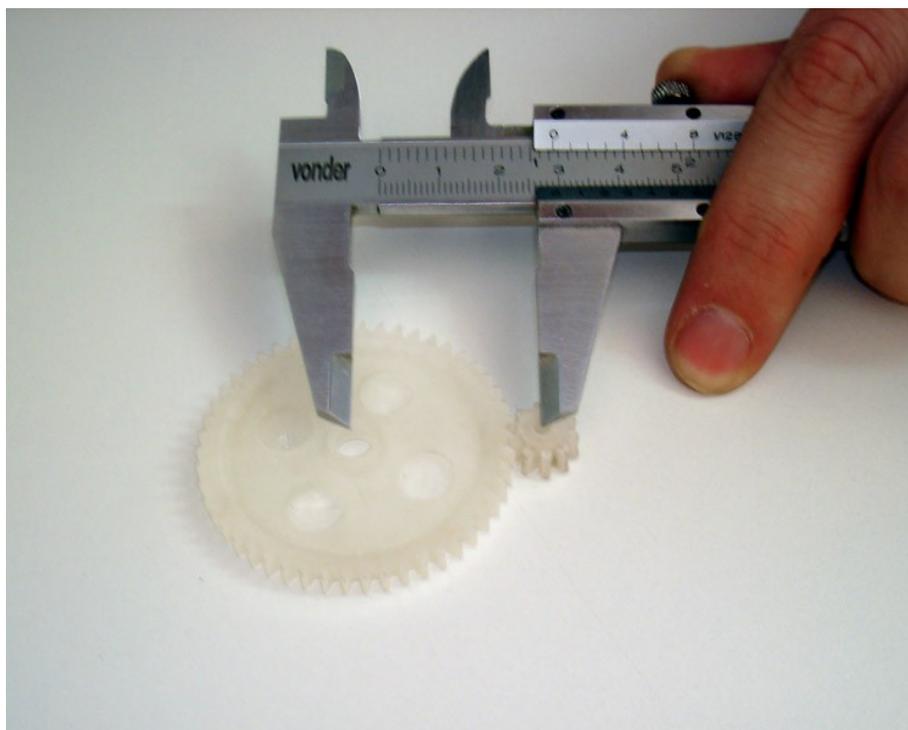


Figura 3: Medindo a distância entre os centros das engrenagens.

Nosso próximo passo é construir os blocos de fixação dos servo-motores de inclinação e de movimentação, utilizando madeira (MDF com espessura de 1,5 cm) e plástico (PVC com 2 mm de espessura). É importante que esses blocos tenham as mesmas dimensões. Em primeiro lugar,

marcaremos os furos a serem feitos para o encaixe do servo-motor e, em seguida, tomando a distância medida entre os centros das engrenagens, faremos a marcação dos furos para o encaixe do potenciômetro, além de dois furos extras para encaixe de parafusos. O bloco foi dimensionado aleatoriamente, mas dentro dos limites permitidos. Assumimos como padrão para nossas montagens tiras de MDF 1,5 cm de espessura cortadas com 5 cm de largura. Por fim, nosso bloco foi dimensionado em um retângulo de 6 cm x 5 cm.

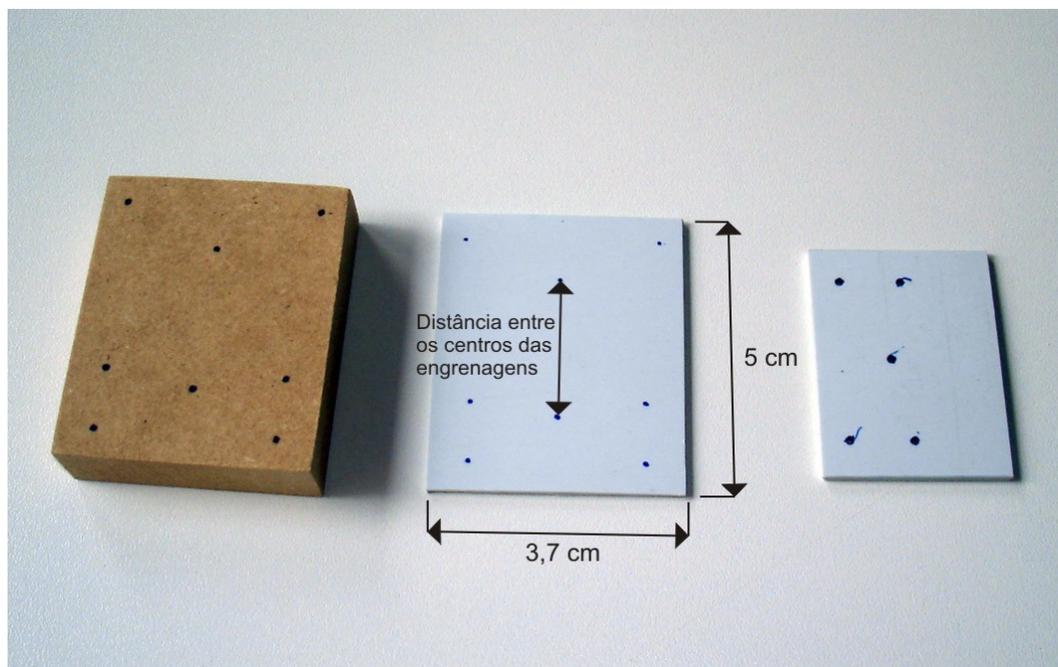


Figura 4: Marcações feitas no bloco de madeira e no plástico, utilizando a peça de plástico padrão (à direita) como referência.

Os furos poderão então ser feitos, sendo que os furos assinalados na figura abaixo poderão ser feitos com certa folga. Como neste caso os parafusos são de 1/8", recomendamos broca de 4 mm. O furo para o eixo do servo-motor pode ser feito com broca de 8 mm na madeira. O eixo do motor deve girar sem atrito com a madeira, por isso é feito com um diâmetro maior do que o do eixo a ser usado, neste caso 1/4". Mas deve-se ter em vista que não poderá ser muito maior do que isto, pois a trava utilizada no eixo deve apoiar-se na madeira como mostrado na figura 8-2). Para o plástico, o furo foi feito com broca de 1/4", o qual deve proporcionar ao eixo um giro livre, sendo necessário espaçá-lo um pouco mais, quando o mesmo não o proporcionar. Isso pode ser feito com o auxílio de uma furadeira de bancada, realizando movimentos leves e circulares, sempre tomando cuidado para manter o plástico perpendicular à furadeira, pois não desejamos que o furo fique com bordas arredondadas, como explicado no tutorial sobre bloco padrão para servo-motor.

O furo para o potenciômetro a ser feito na madeira deve passar por duas etapas: primeiro deve-se fazer um furo guia, e depois utilizar a broca chata para madeira para aumentá-lo, lembrando-se sempre de utilizar os equipamentos de segurança. No plástico, esse furo será do diâmetro da rosca do potenciômetro, podendo ser alargado futuramente se necessário para um ajuste de relação entre as engrenagens.

Os furos de fixação para o servo-motor na madeira podem ser de qualquer diâmetro que comporte o parafuso a ser utilizado, que no nosso exemplo é de 2,9 mm. O furo foi feito com broca de 2,5 mm e para o plástico utilizamos o mesmo tamanho do parafuso.

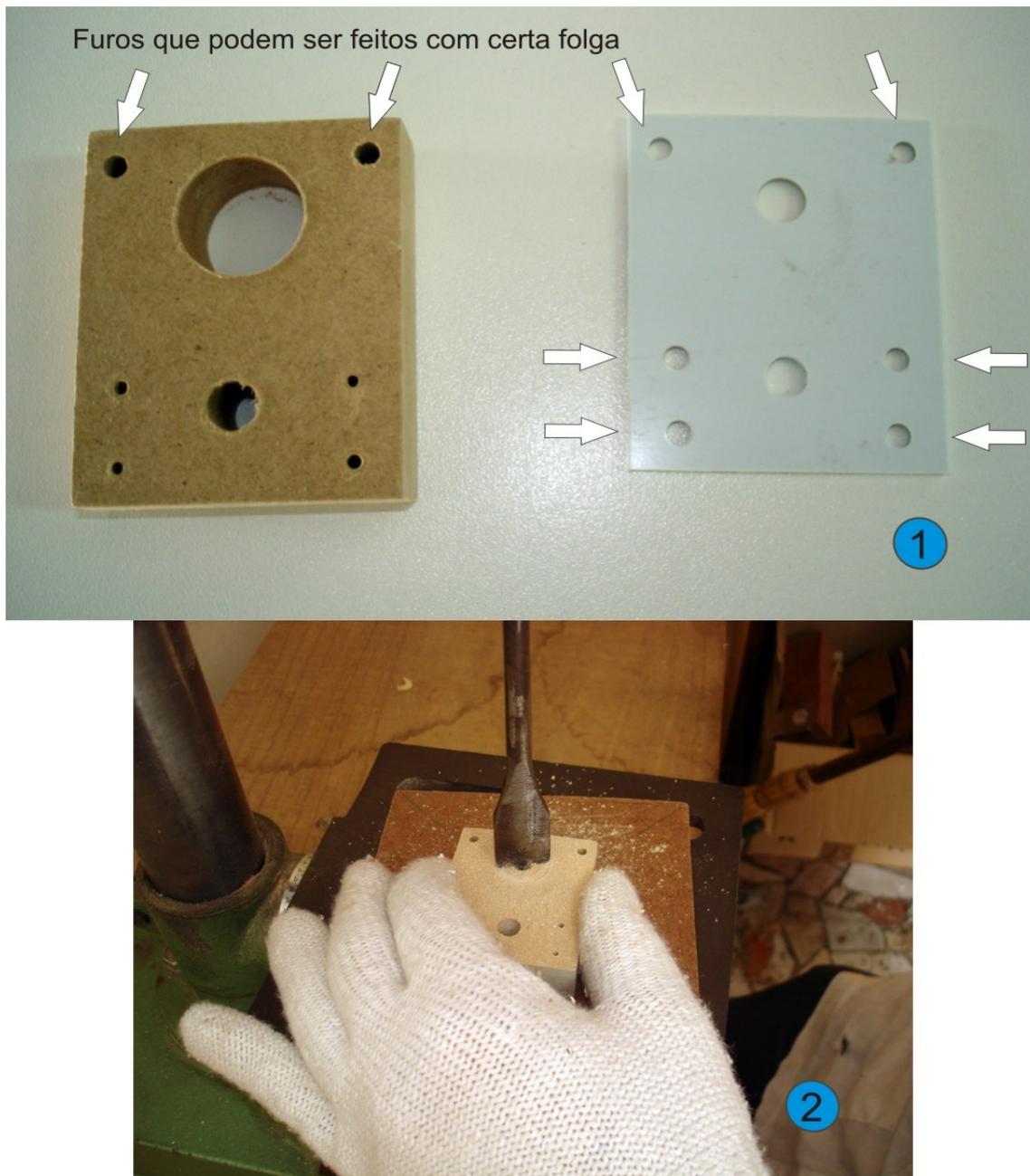


Figura 5: 1-Peças de madeira e de plástico perfuradas, com indicação dos furos que podem ser feitos com certa folga. 2-Furo para o potenciômetro sendo feito com a broca de madeira.

Em seguida, iremos utilizar a serra para cortar a madeira onde será encaixado o potenciômetro, já que, devido ao seu formato, não podemos encaixá-lo diretamente no furo. Antes de cortar, devemos colocar parafusos nos furos próximos ao local a ser serrado, com o objetivo de evitar que a madeira se abra.



Figura 6: Finalizando as peças do bloco do servo-motor. 1-Corte do bloco de madeira para o encaixe do potenciômetro. 2-Bloco de madeira e de plástico prontos.

O próximo passo consiste na criação de um pino-eixo para o servo-motor. Seu tamanho varia de acordo com o potenciômetro usado como eixo externo, sendo o tamanho mínimo mencionado no tutorial sobre bloco padrão para servo motor, mais a área não útil do eixo do potenciômetro (onde há a rosca para o encaixe de uma porca). No nosso caso utilizamos um pino-eixo de 2,7 cm (os 2,3 mm recomendados mais 4 mm). Este pino tem uma pequena diferença do modelo padrão: como a engrenagem utilizada é muito pequena para ser perfurada e encaixada no pino-eixo, iremos utilizar um pequeno parafuso para fixá-la. Primeiramente, devemos fazer um furo (da mesma dimensão do parafuso a ser usado, no caso 1/8") na engrenagem para o encaixe de um parafuso, com os cuidados necessários para que ele não saia torto.

Para fazer isso, precisamos, primeiro, lixar as engrenagens para retirar as rebarbas de acrílico que ficam em volta delas, evitando que fiquem desniveladas quando as apoiarmos para furá-las. Então, com muito cuidado, devemos fazer os furos nas engrenagens. Uma dica é utilizar broca de espessura menor para evitar que o furo saia torto e também para melhor direcionar a furação (recomendamos a broca de 1,5 mm para este fim). Depois é só furar as engrenagens utilizando a broca adequada. Por motivos de segurança recomendamos que não se segure a engrenagem diretamente com a mão. Sugerimos o uso de um alicate, por exemplo.

Depois, devemos fazer um furo no centro do pino-eixo, com tamanho suficiente para o encaixe do parafuso (em nossa montagem esse furo foi feito com broca de 1/8"). O parafuso neste caso não precisa entrar rosqueando, pois ele será colado ao pino-eixo. Esse procedimento é um pouco mais complicado, pois precisamos de um suporte para o pino-eixo. Utilizamos pedaço de madeira qualquer. Foi feito um furo para o encaixe do pino-eixo, servindo assim de base para que possamos segurar o pino com segurança. O pino-eixo deve entrar com relativa pressão na madeira para que fique bem firme, podendo ser necessário utilizar um martelo para que o pino entre na madeira, tomando cuidado para não amassá-lo. Como o pino-eixo utilizado possui um diâmetro de 1/4", o furo na madeira também foi feito com broca de 1/4". Após introduzir o pino-eixo na madeira, basta então furá-lo. É importante tomar cuidado para que o furo não seja torto nem muito profundo. Para retirar o pino-eixo do suporte, utilize um martelo, batendo no suporte para fazer a madeira abrir, tomando cuidado para não acertar o pino-eixo.

Com o furo pronto, parte-se para o processo convencional que adotamos para a criação de um pino para o motor, como o corte para o encaixe no servo-motor e o vinco para a colocação da

trava. Com o pino pronto para ser encaixado no servo-motor, colamos o parafuso à engrenagem e depois o parafuso ao pino, ambos com a cola de secagem instantânea.

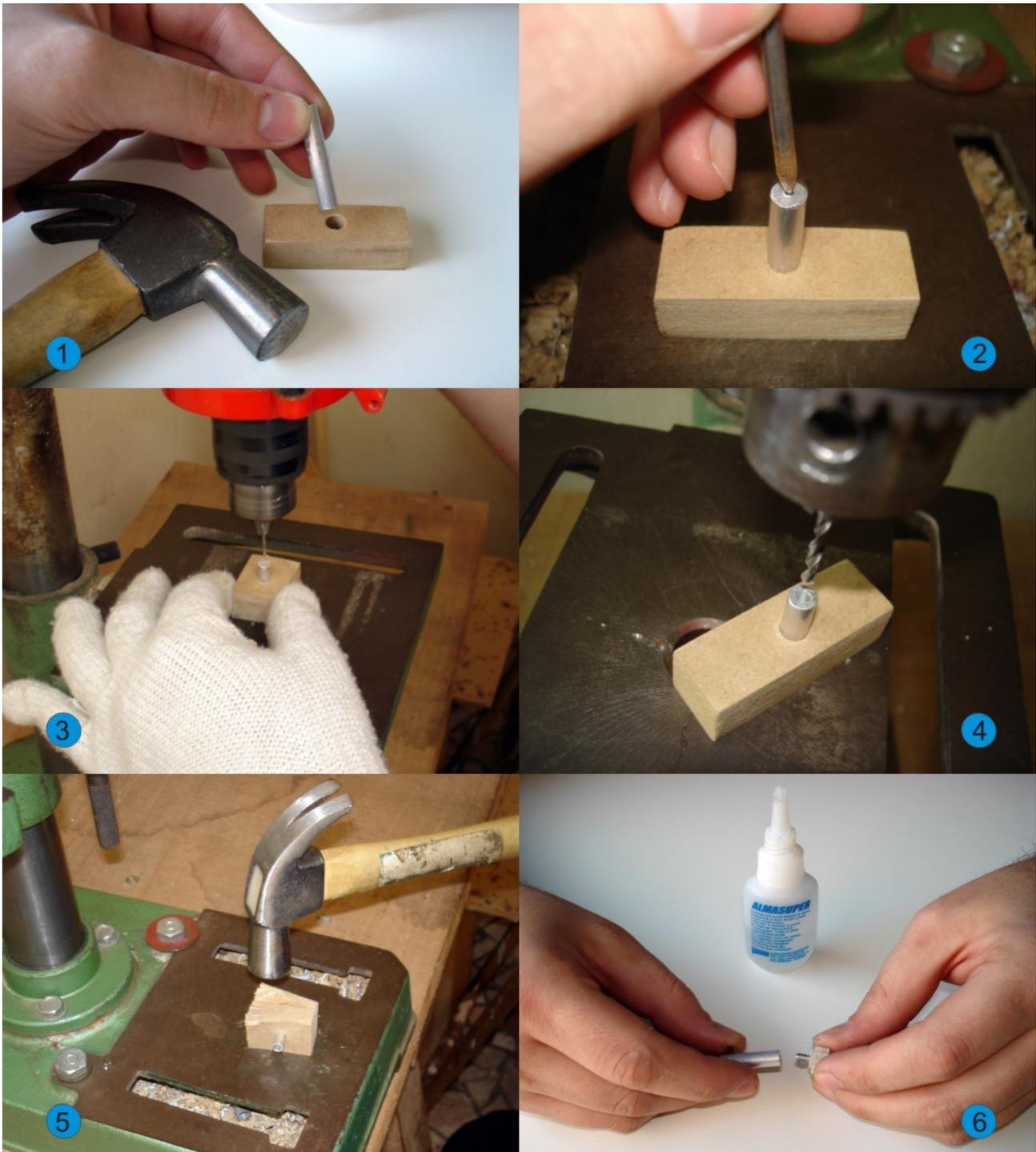


Figura 7: Criação do pino-eixo 1-Pequeno bloco de madeira para o encaixe do pino-eixo. Ele deve se manter firmemente preso à esse bloco. 2-Marcação no pino-eixo do furo para o encaixe do parafuso. 3-Pino-eixo sendo perfurado com a broca de 1/8". É importante que o furo não seja torto nem muito profundo. 4-Pino-eixo já perfurado. 5-Para retirar o pino-eixo do bloco de madeira, deve-se martelar esse bloco até que ele se abra. 6-O parafuso com a engrenagem pode então ser colado ao furo do pino-eixo com a cola de secagem instantânea.

É interessante também lubrificar o eixo, na região que ficará apoiada no plástico do bloco do servo-motor, preferencialmente com graxa. O servo-motor, então, deverá ser fixado no bloco por meio de parafusos para madeira e a parte de plástico que servirá para sustentação do pino-eixo e do potenciômetro deverá ser parafusada à madeira por meio de parafusos e porcas.

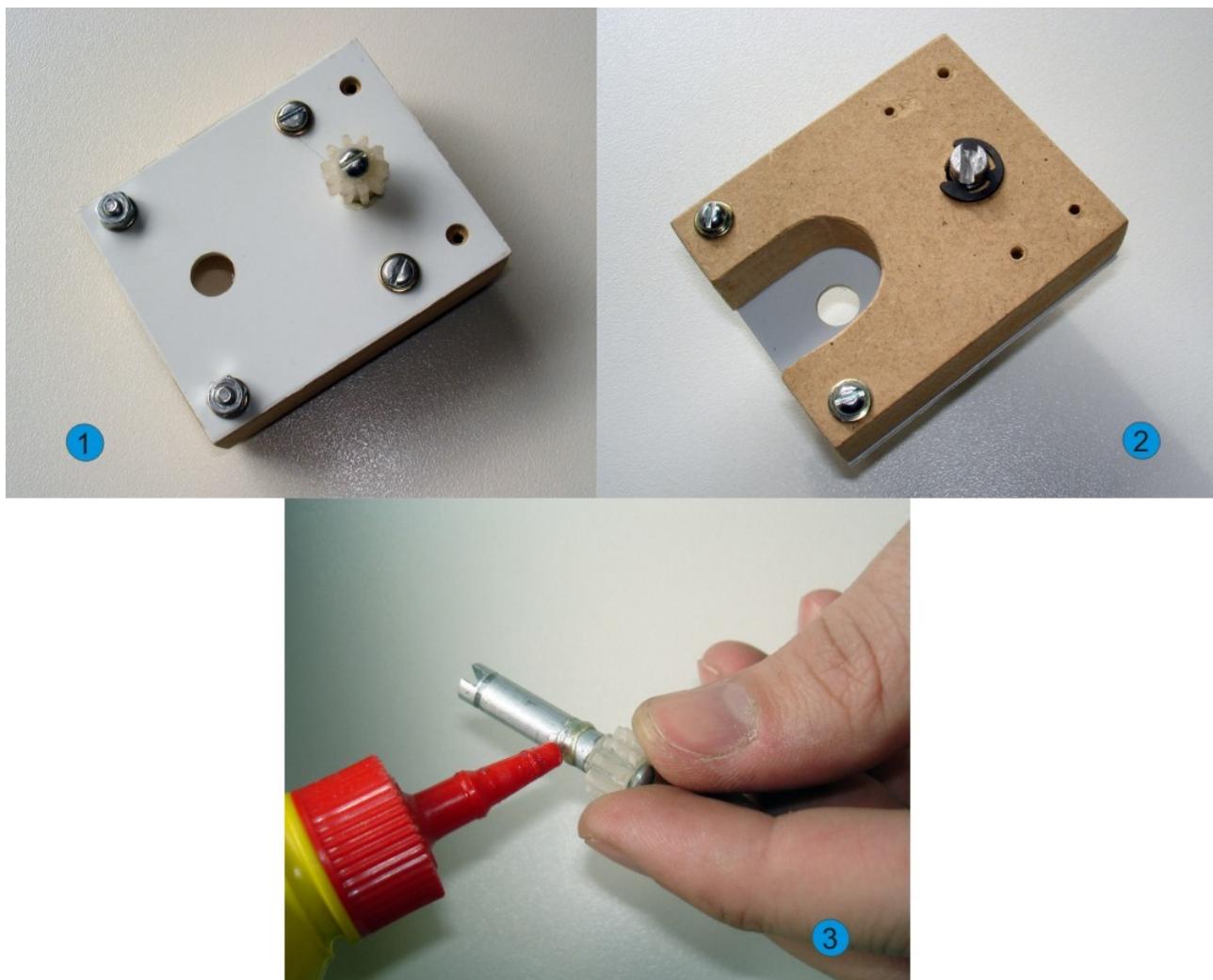


Figura 8: Bloco do servo-motor, já com o pino-eixo . 1- Visão superior. Observe que os dois furos que sobram serão destinados à fixação do servo-motor 2- Visão inferior. 3-Lubrificando o pino-eixo na região que estará em contato com o plástico.

O potenciômetro externo servirá para informar a posição da engrenagem maior. Isso será essencial, pois o Kit controlará sua posição de modo a fazer com que o bípode se incline e desloque. Devemos observar bem a forma com que os fios do potenciômetro devem ser soldados aos terminais onde estava o potenciômetro original, já que estamos utilizando um servo-motor adaptado para rotação contínua. Seguindo o exemplo, com os terminais do potenciômetro voltados para o lado de fora do bloco, os fios são soldados de maneira espelhada ao potenciômetro original, ou seja, com os fios das extremidades invertidos. Isso deve ser feito pois a engrenagem a qual está preso o potenciômetro gira no sentido contrário da engrenagem presa ao eixo original do servo-motor.

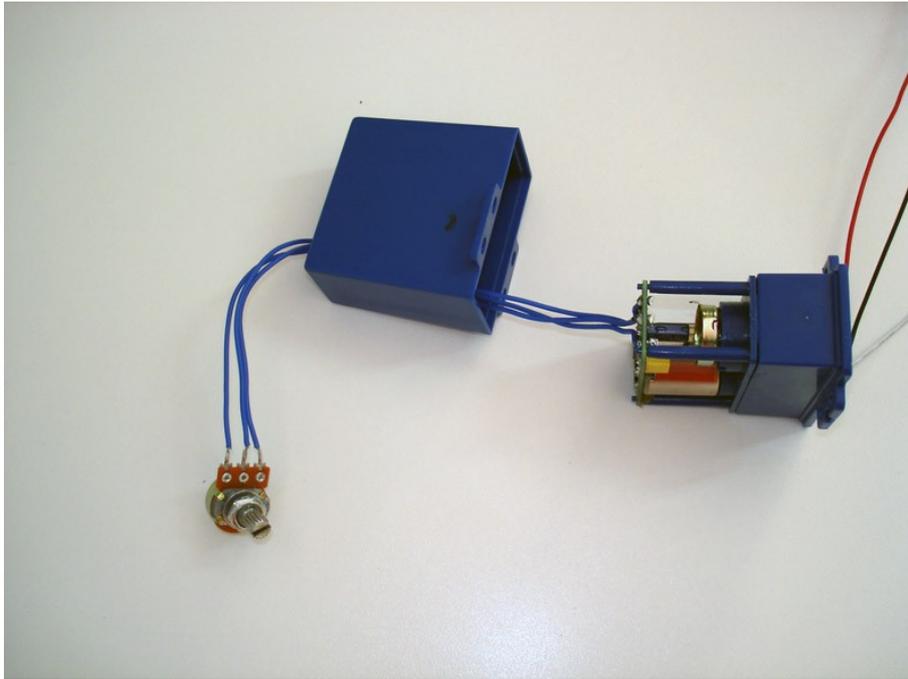


Figura 9: Servo-motor com o potenciômetro externo.

O próximo passo será preparar a engrenagem grande antes de encaixá-la ao potenciômetro. Deve ser feito um furo no seu centro cujo tamanho seja suficiente para o encaixe do eixo do potenciômetro. Além disso, devemos fazer furos em dois pontos da extremidade, alinhados com o centro no pino do potenciômetro. Esses furos devem ser da espessura do parafuso a ser utilizado como futura base para a transferência de movimento do servo-motor para as pernas do bípode. Utilizamos arruelas dentadas para que o parafuso fique travado na engrenagem. Se necessário, reforce com cola de secagem instantânea. Esses procedimentos deverão ser feitos nas duas engrenagens. No servo-motor de inclinação, encaixaremos uma barra de alumínio, enquanto que no servo-motor de deslocamento, cada parafuso será preso a uma barra de plástico.



Figura 10: Engrenagem responsável pela redução e transferência do movimento para as pernas.

Iremos agora encaixar o potenciômetro por meio da sua própria porca e colocar a engrenagem grande de modo que os seus parafusos tenham suas cabeças voltadas para o bloco do servo-motor. Para o correto encaixe da engrenagem, devemos primeiro definir a posição central do potenciômetro. Ligaremos o servo-motor na saída do Kit destinada a servo-motores, através do programa de controle de servo-motores faremos com que o que o motor vá para a posição central, e depois, girando o potenciômetro com a mão, podemos achar a posição central de referência do potenciômetro externo. Isso se dá quando o motor pára de girar por completo. Então, poderemos colar a engrenagem, utilizando cola de secagem instantânea, tomando o cuidado para que ela não escorra para dentro do potenciômetro.

Sempre que possível deixe para colar as peças no final das montagens, às vezes pode ser necessário ter de desmontar algumas peças. Ao colar a engrenagem no potenciômetro, por exemplo, deve-se ter em mente que não será mais necessário alterar o bloco, pois, ao tentar descolar a engrenagem do potenciômetro, podemos vir a danificar alguma dessas peças.

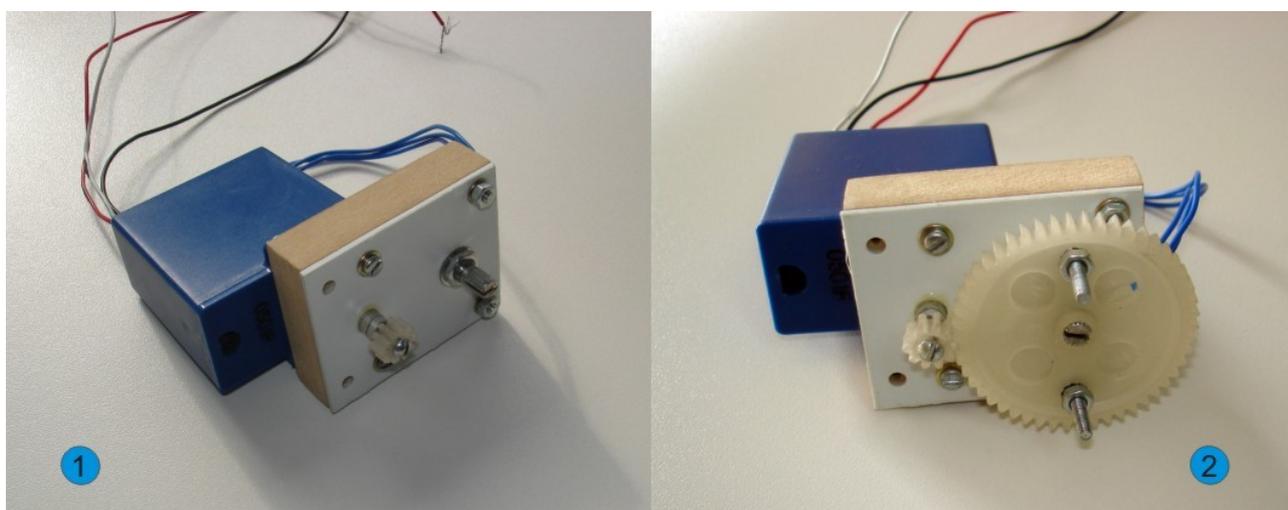


Figura 11: Finalizando o bloco do servo-motor. 1- Servo-motor fixado ao bloco. 2- Engrenagem grande fixada ao eixo do potenciômetro.

Com os dois blocos dos servo-motores prontos, passaremos a confecção do corpo. Faremos o corpo do bípede utilizando madeirite, pois é um material leve e com resistência adequada, mas outros materiais também podem ser utilizados, como plásticos, acrílicos, até mesmo papéis mais resistentes. Devemos, primeiramente, definir qual será o tamanho destas peças, que nesse exemplo foram feitas com dimensões de 16 cm x 8 cm. Em seguida, temos apenas de transferir essas medidas para o madeirite e cortá-lo com a serra. Para que o corte saia reto, deve-se traçar uma linha sobre o madeirite e segui-la corretamente, apoiando a peça sobre uma mesa para que fique bem firme, já que ele é bem maleável. Desse modo teremos certeza de que cada peça terá as dimensões corretas. Em seguida, devemos dimensionar nestas peças as furações a serem feitas para o encaixe dos blocos dos servo-motores.

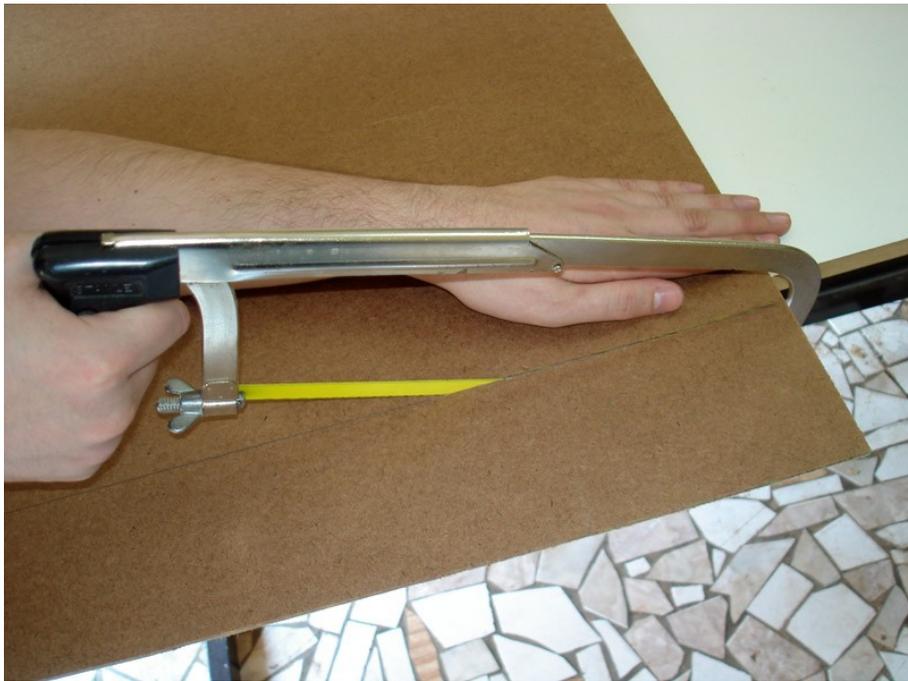


Figura 12: Madeirite sendo cortado com o arco de serra.

Fizemos os furos nestas peças de modo que os furos nos blocos dos servo-motores ficassem à 1 cm da borda, procurando atingir o centro da madeira do bloco do servo-motor, para assim evitar possíveis rachaduras no mesmo. O bloco responsável pela movimentação deverá estar com a face de suas engrenagens voltadas para baixo, enquanto que o outro bloco, responsável pela inclinação, deve ter sua face com engrenagens voltada para frente, como pode ser observado na figura 11. Próximo aos furos para encaixar o bloco do servo-motor responsável pela movimentação, faremos furos para colocar as pernas do bípede, bem nos cantos das peças de madeirite. Todos os furos devem estar perfeitamente alinhados, para evitar uma possível angulação inadequada nas peças móveis, prejudicando assim o movimento. Neste caso os furos para a fixação das pernas foram feitos a 5 mm das laterais da madeira, estando cada um a 7 cm de distância do outro. Todos os furos foram feitos com broca 1/8".

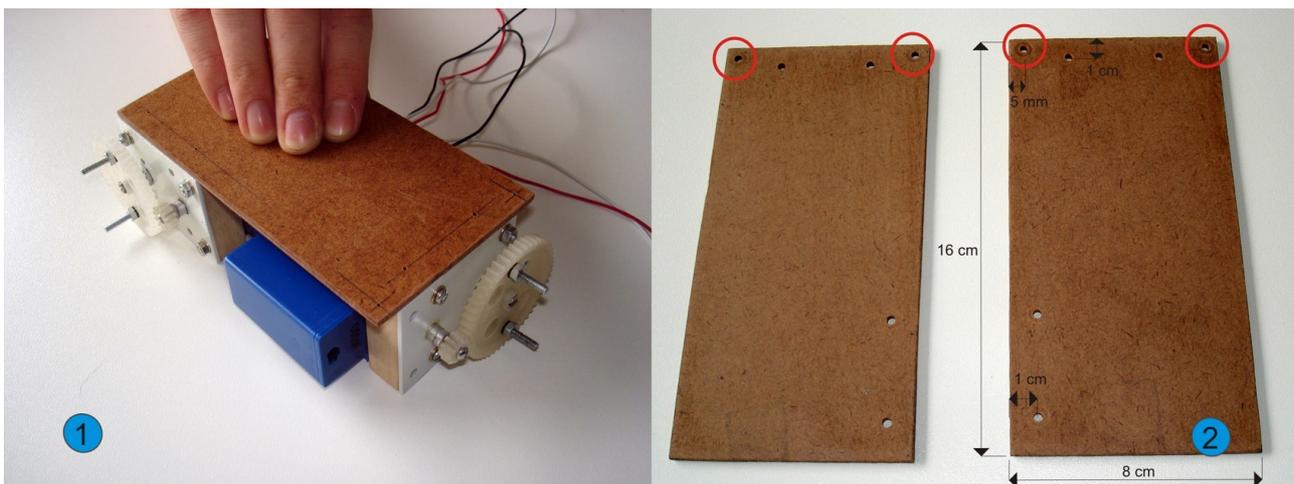


Figura 13: Confeção do corpo do bípede. 1- Definindo o tamanho. 2- Peças de madeirite já perfuradas. Observe os furos extras feitos para a fixação das pernas.

Em seguida, iremos usar estas peças para marcar os furos a serem feitos nos blocos dos servo-motores, para podermos parafusá-los. Devemos, então, furar os blocos dos servo-motores utilizando uma broca com espessura adequada para o parafuso utilizado. Como utilizamos parafusos autoatarraxantes de 2,9 mm, o bloco do motor foi perfurado com a broca de 2,5 mm. Tome muito cuidado para que o potenciômetro não seja perfurado.

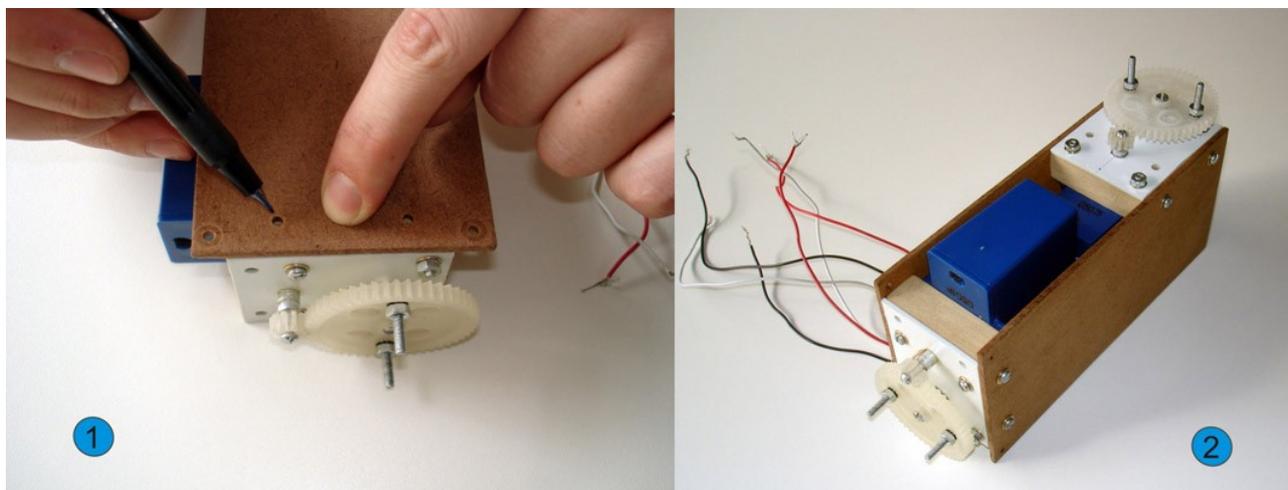


Figura 14: Finalização do corpo do bípede. 1- Marcando o bloco do servo-motor para furá-lo. 2- Corpo do bípede pronto.

Em seguida iremos confeccionar as pernas para o bípede, com 4 barras (perfis) de alumínio quadradas de 1/4". Optamos pela quadrada por ser mais fácil de ser perfurada, mas a barra circular pode ser considerada esteticamente melhor. O comprimento destas pernas dependem do tamanho do corpo, para que não tenhamos uma estrutura desproporcional. No exemplo, as pernas do bípede têm 15 cm de comprimento. Após cortá-las, devemos lixar as rebarbas para um melhor acabamento e também para evitar possíveis ferimentos.

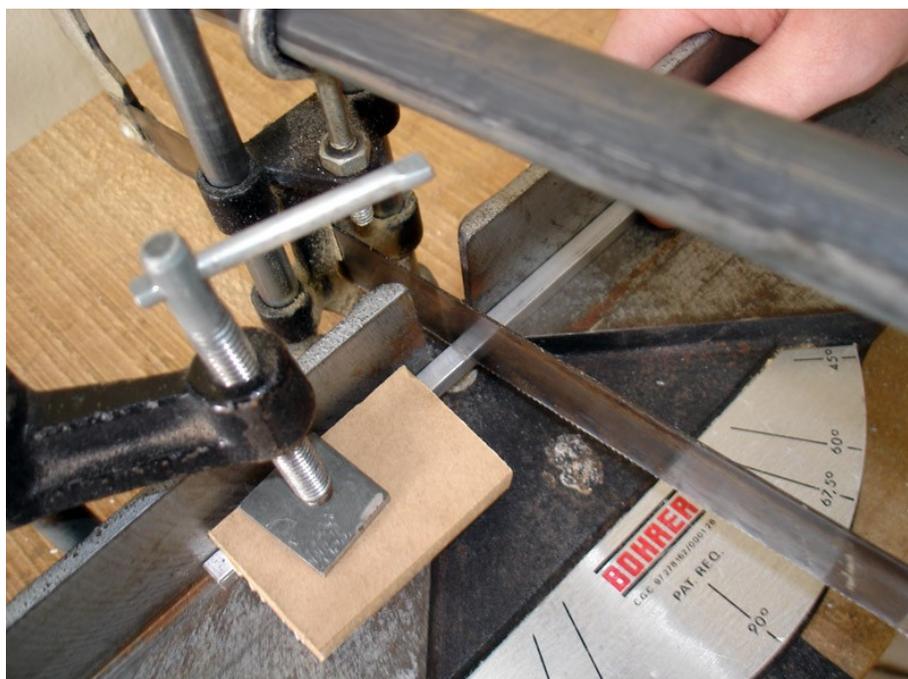


Figura 15: Cortando a barra de alumínio quadrada na serra de meia-esquadria.

Em seguida, em cada barra, faremos a marcação de um furo a ser feito a 5 mm de uma das extremidades. Recomendamos caneta para retroprojektor, sempre que forem feitas marcações em alumínio. Estes furos servirão para fixar as pernas ao corpo do bípede. Devemos fazer mais um furo em cada perna para podermos, depois, encaixar os pés. Este furo será feito na outra extremidade e do lado perpendicular ao furo feito para o encaixe do corpo. Além disso, este furo deve ser um pouco modificado: encaixaremos a perna na broca e a moveremos para cima e para baixo para que a abertura do furo seja espaçada externamente, com o intuito de formar junto com o parafuso, um pé articulado. É necessário fazer esse procedimento nos dois lados do furo, para que as pernas do bípede possam se movimentar para frente e para trás. Sem isso, os pés do bípede ficariam travados, impedindo que o movimento ocorresse. Porém deve-se tomar cuidado para não aumentar o diâmetro do furo, e tentar manter uma angulação igual para as quatro pernas.



Figura 16: Procedimentos para a criação das pernas do bípede. 1-Deve ser feito um furo a 5 mm da extremidade. 2-Furo sendo feito. 3 e 4-Para espaçar a abertura do furo, a barra de alumínio deverá ser movida para cima e para baixo.

Vale lembrar que a maioria dos metais quando perfurados, irão ficar com rebarbas, e isso, além de prejudicar os movimentos, pode nos ferir. Isso pode ser resolvido com o auxílio de uma broca um pouco maior do que o furo que foi feito, girando de leve como na imagem abaixo, tomando cuidado para não arredondar demais as beiradas dos furos.



Figura 17: Para retirar as rebarbas, basta utilizar uma broca maior que o furo feito e girá-la levemente.

As pernas receberão parafusos, que se tornarão uma espécie de eixo, para prendê-las aos pés e ao corpo do bípode. Isso se dá da seguinte forma: com os furos preparados, já sem rebarbas, pegase uma arruela, coloca-se ela no parafuso, coloca-se o parafuso na perna, adiciona-se outra arruela e depois mais duas porcas. As arruelas normalmente possuem dois lados diferentes, um liso e o outro com rebarba. O lado liso neste caso deve ficar voltado para a barra de alumínio, pois a idéia é gerar o menor atrito possível. Rosqueie as porcas até o final e quando encostar no alumínio, solte-as um pouco (mas não muito pois é necessário que haja uma folga suficiente para que o parafuso gire com o menor atrito possível, porém, se houver uma folga muito grande o movimento será prejudicado). Verifique se não há atrito entre o parafuso e a perna e então aperte uma porca contra a outra. Essa é uma forma simples de se obter um tipo de eixo. Pode-se também, utilizar apenas uma porca, colando-a ao parafuso. No entanto recomendamos que seja utilizado o processo que envolve o uso das duas porcas, para evitar problemas futuros.



Figura 18: Colocando parafusos e porcas na perna do bípode. 1-Enquanto uma das porcas é presa com uma chave de boca, outra é regulada através de um alicate de bico. 2-Porcas e arruelas colocadas.

No furo articulado a diferença é que a folga mantida entre a perna e as arruelas e porcas deverá ser igual à angulação máxima obtida com o movimento. O tamanho dos parafusos utilizados pode variar de projeto para projeto e deve-se ter em mente o que será feito depois que se concluiu o eixo. Neste caso, esta perna será fixada no corpo do bípede, ou seja, utilizaremos mais duas arruelas, uma porca, e ainda devemos levar em consideração a largura do material utilizado para estrutura do corpo (que no caso é de 3 mm). Para o eixo que vai preso ao corpo utilizamos um parafuso de 7/8" de comprimento, e para o eixo articulado um com 1" de comprimento, sendo ambos de 1/8" de diâmetro.

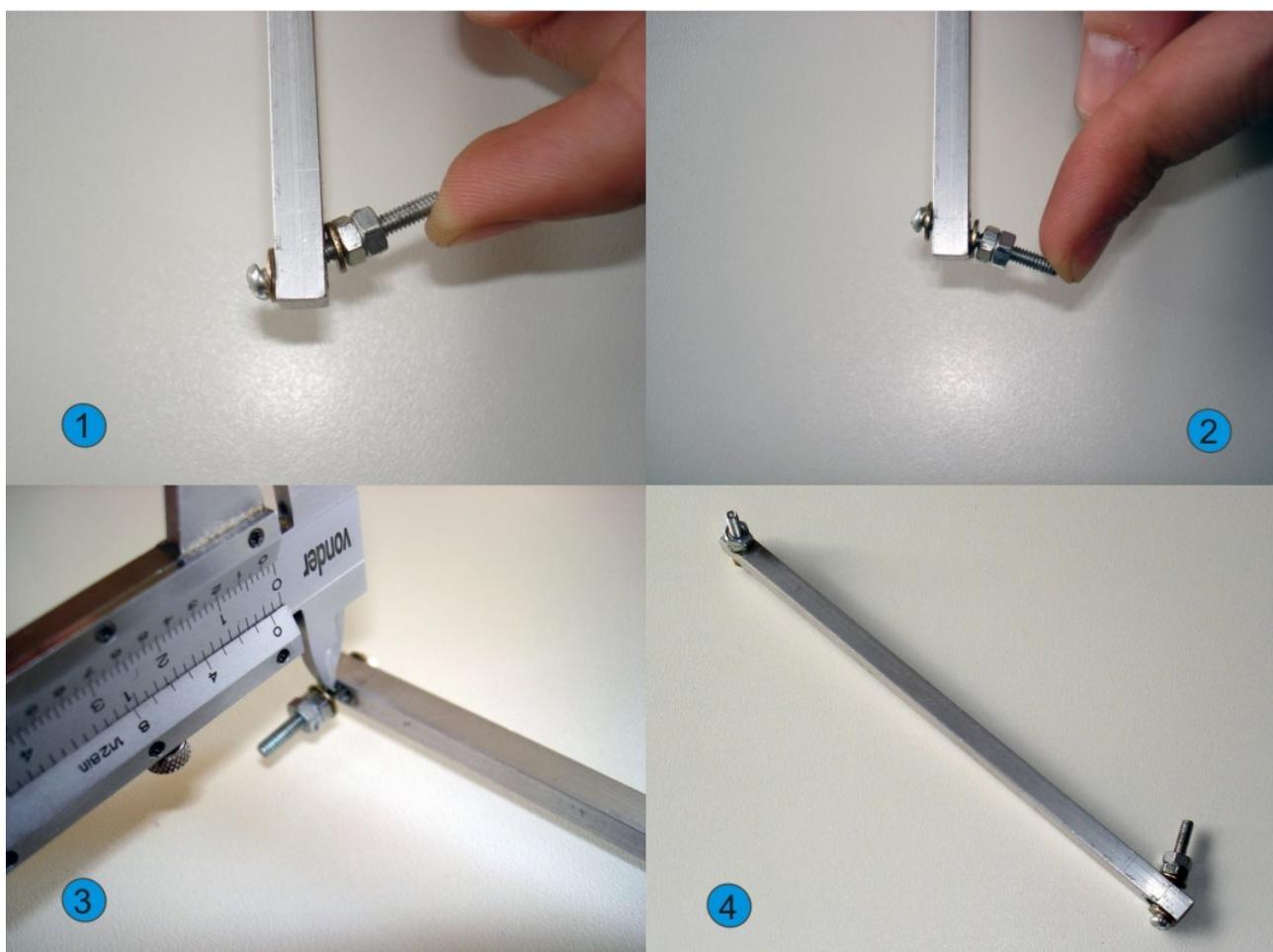


Figura 19: 1 e 2-Exemplificando o movimento a ser realizado pela perna através dos eixos articulados. 3-Medindo o espaço disponível para a movimentação. 4-Perna pronta.

Estes são pontos que devem ser lubrificados com óleo, mas precisamos ter cuidado para não exagerar na quantidade de lubrificante. É necessário tomar bastante cuidado para não deixar que o óleo atinja pontos indesejáveis, como as duas porcas que foram travadas umas nas outras, pois isso facilitará com que as porcas se soltem com o esforço realizado com a movimentação. Assim como nas colagens (como no caso das engrenagens em potenciômetros), recomendamos que esta etapa fique para o final do processo de montagem.

Após as pernas estarem prontas, criaremos pequenas hastes para transformar o movimento giratório do eixo do servo-motor de deslocamento em movimento retilíneo, para que o bípede possa “caminhar”. Estas hastes servirão para transferir o movimento do motor para as pernas do bípede.

Elas foram feitas com plástico de PVC com 2 mm de espessura da seguinte maneira: cortamos quatro tiras do PVC com cerca de 5 cm x 1 cm cada uma, e fizemos um vinco em uma extremidade de cada peça, encaixando uma na outra, formando dois pares. Utilizamos cola de secagem instantânea, para fixar uma à outra. Este vinco pode ser feito com um estilete ou até mesmo com o disco de corte de microrretífica.

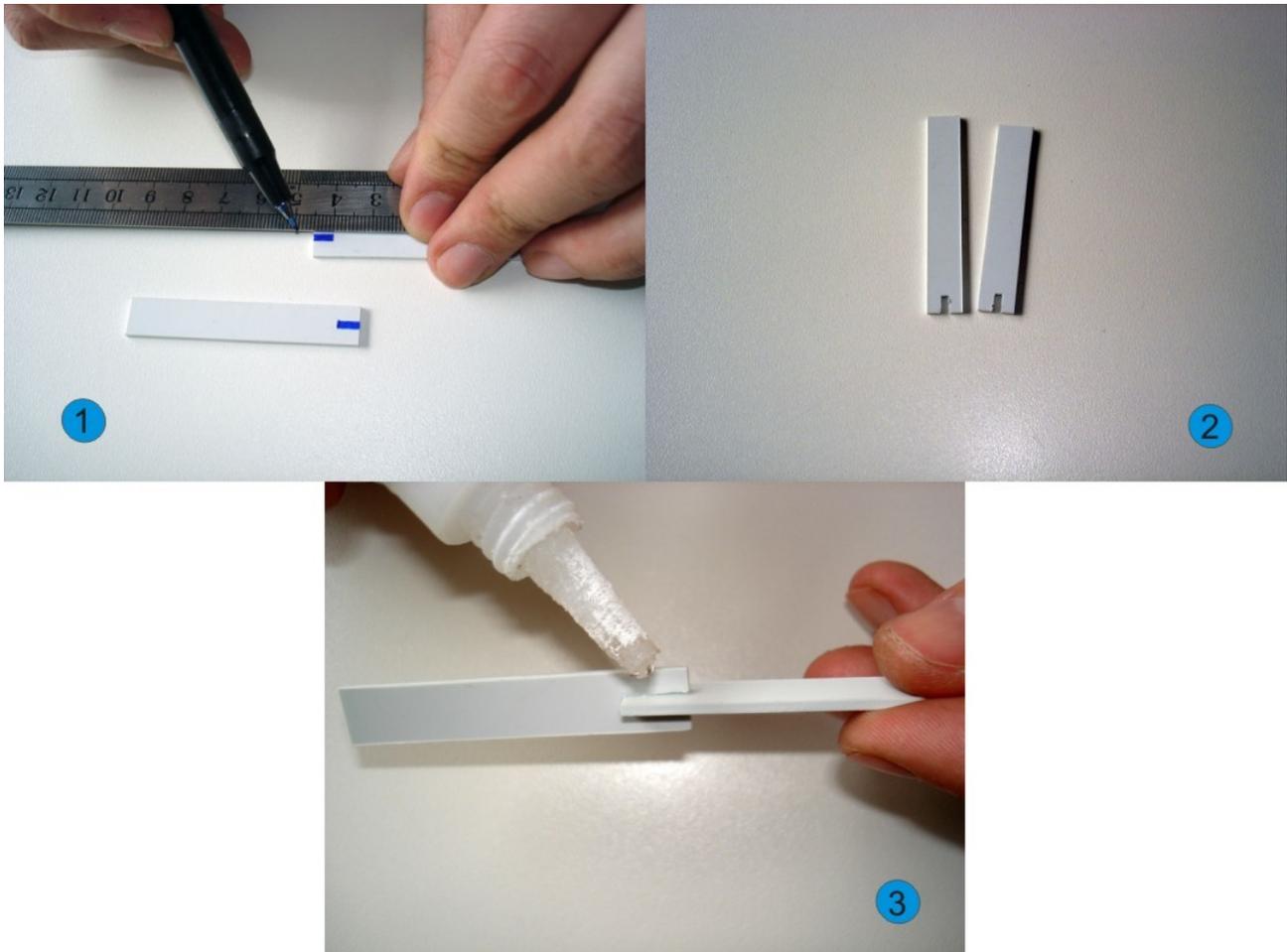


Figura 20: Criação das tiras de plástico, destinadas a deslocar as pernas do bípode para frente. 1-Marcando os vincos a serem feitos. 2-Vincos feitos nas peças. 3-Colando as peças com cola de secagem instantânea.

Em seguida, fizemos um pequeno furo numa das extremidades das tiras de plástico para podermos encaixá-las nos parafusos que foram fixados anteriormente na engrenagem. Não fixe estas hastes agora, pois elas ainda receberão mais um furo. Assim podemos medir, com a ajuda de um paquímetro, por exemplo, o ponto em que a perna do bípode receberá um novo furo. Nesse furo, será encaixado um parafuso que interligará a perna com a engrenagem através das peças de plástico. Agora também é possível determinar a localização do outro furo que receberá a peça de plástico responsável pela transmissão do movimento. Todos os furos foram feitos com broca 1/8", pois foram utilizados parafusos de 1/8". Os furos nas hastes plásticas foram levemente espaçados para não gerarem atrito durante a movimentação, o que pode vir a prejudicar a movimentação.

Depois de furados, encaixaremos as tiras nos parafusos e ajustaremos o espaço móvel para elas por meio de duas porcas. Essas porcas poderão ser coladas com cola de secagem instantânea, para evitar que se movimentem. Na fixação das hastes nos parafusos da engrenagem precisamos

levar em conta que elas estarão se movimentando, portanto nunca se deve fixá-las diretamente com uma porca. Mantêm-se uma pequena folga para que as hastes possam se movimentar livremente. O processo realizado foi o seguinte: colocamos primeiramente uma arruela com seu lado liso voltado para a haste que será encaixada, colocamos outra arruela (também com o lado liso voltado para o plástico), e depois duas porcas que serão apertadas uma contra a outra de forma semelhante a já realizada com os eixos das pernas do bípede.

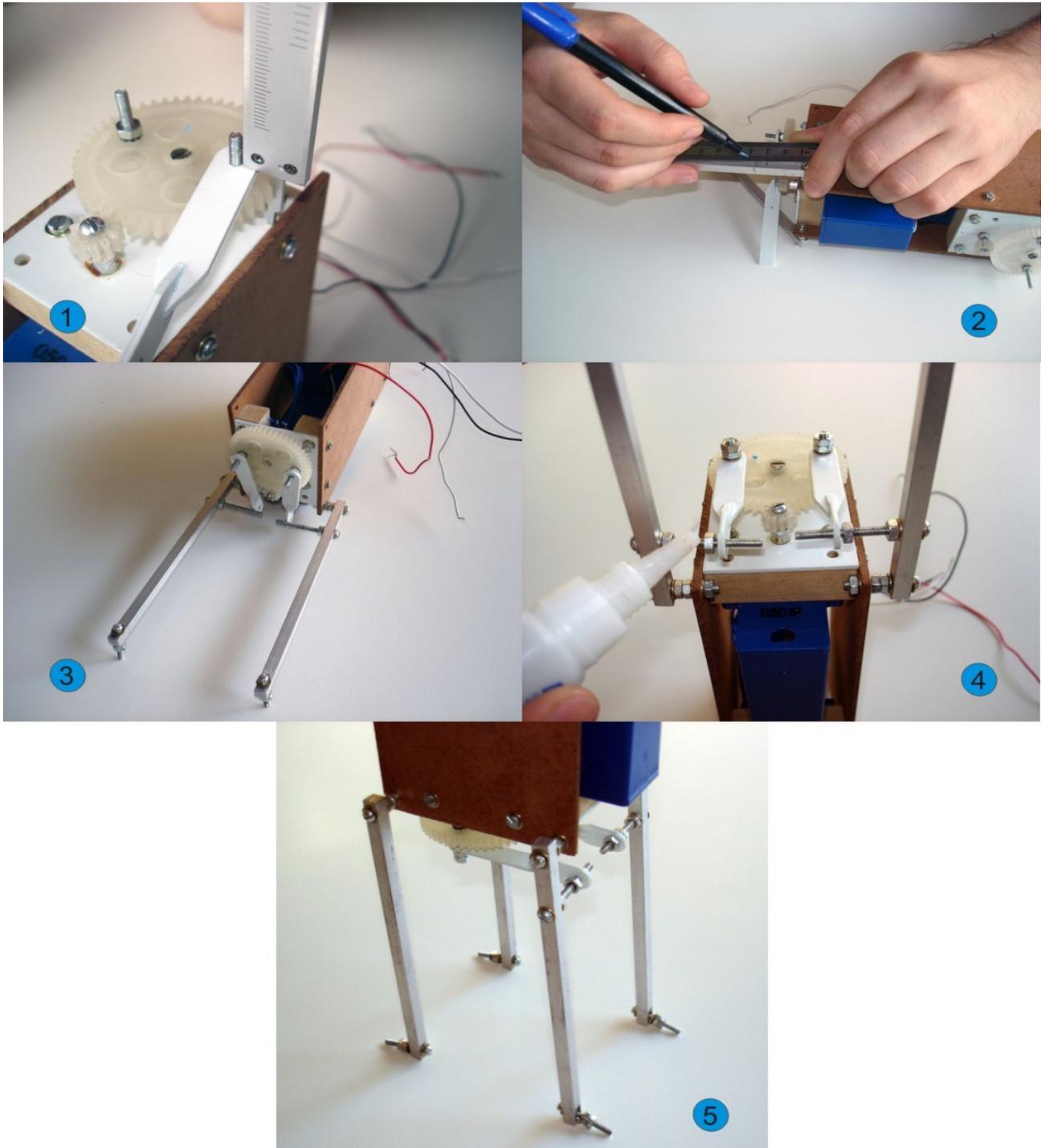


Figura 21: Colocando as pernas no corpo do bípede. 1 e 2-Encontrando a posição onde será feito o furo na perna do bípede, para o encaixe das hastes de plástico. 3-Pernas já com os parafusos para o encaixe das hastes. 4-Colando as porcas que ajustam o espaço livre para o movimento da haste. 5-Pernas presas ao corpo do bípede.

O passo seguinte se refere a confecção de pés e “sapatos” para o bípede. Serão utilizados para isso quatro retângulos de madeirite e cantoneiras de alumínio. As dimensões do exemplo são de 15 cm x 8,5 cm para os pés e 17 cm x 8,5 cm para os sapatos. Utilizamos cantoneiras de alumínio de 5/8” para que sirvam de relação entre a fixação das pernas e dos a pés. Estas cantoneiras (quatro no total) foram cortadas com 8,5 cm, ou seja, a mesma largura dos pés. Cada uma delas recebeu três furos, dois deles para a sua fixação nos pés e um terceiro onde serão fixados os eixos das pernas. As cantoneiras que ficarão voltadas para frente ainda receberão um quarto furo feito na extremidade externa da cantoneira. Explicaremos este furo mais adiante. Os furos que prendem as cantoneiras aos pés podem ser escolhidos aleatoriamente, dando preferência para deixá-los mais nas extremidades. Os furos que servirão para fixar os pés às pernas devem manter uma certa distância da base do pé para que a perna não entre em contato com a estrutura do pé quando o bípede estiver em movimento. Este furo não deve ficar muito distante da parte interna do pé do bípede por motivos de equilíbrio do robô. No exemplo, o furo foi feito com 0,8 cm de altura em relação a base do pé, e 1 cm para dentro.

Fixaremos primeiramente as cantoneiras da parte de trás do bípede, utilizando o seguinte procedimento: colocaremos a cantoneira bem na borda do pé de madeirite, marcaremos os furos a serem feitos e, depois de furá-los, utilizaremos parafusos, arruelas e porcas para fixar esta cantoneira ao pé.

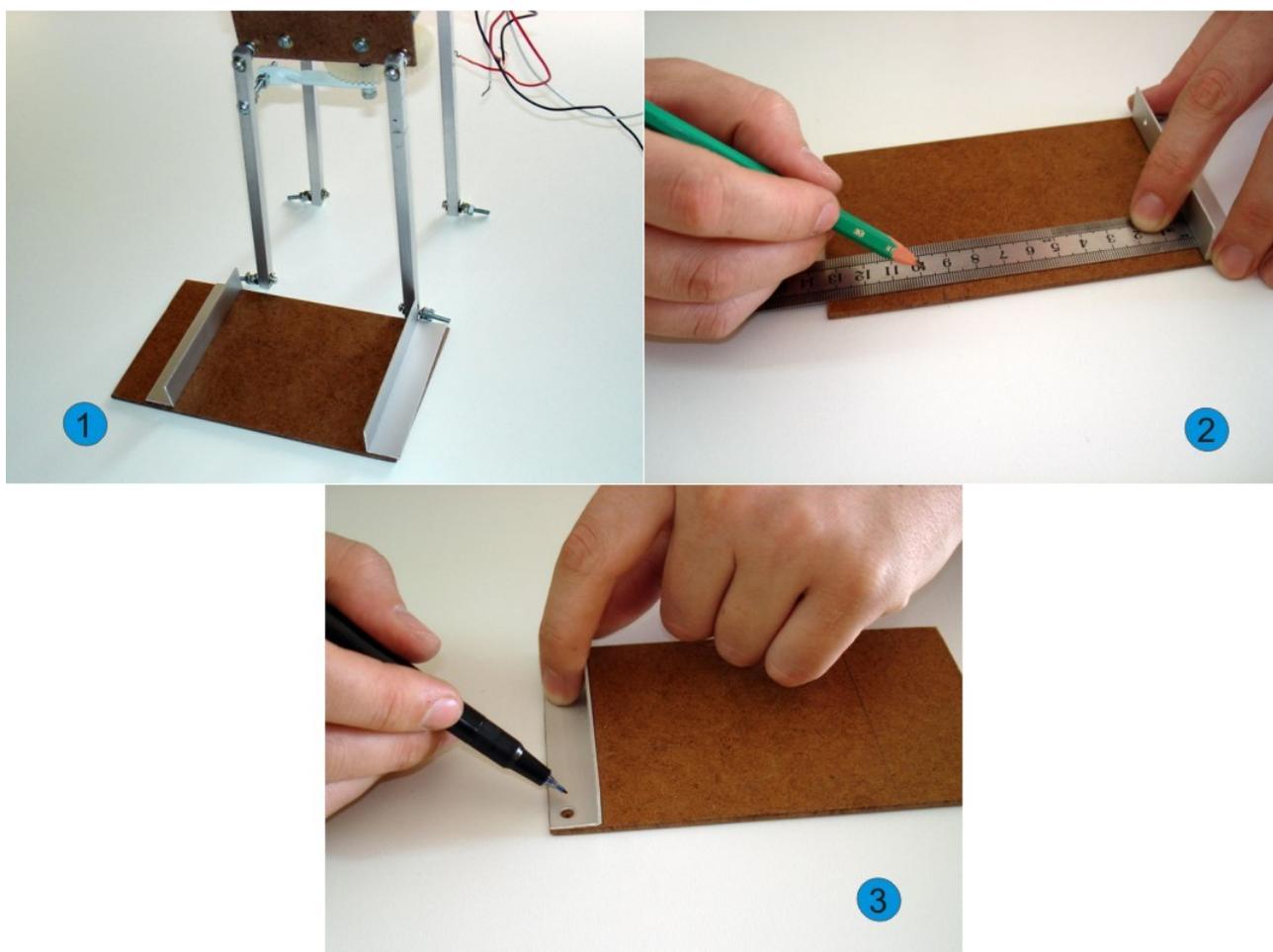


Figura 22: Construindo os pés do bípede. 1-Definindo a localização das cantoneiras. 2-Cantoneira sendo posicionada no pé. 3-Marcando o furo a ser feito no pé, para o encaixe da cantoneira.

Em seguida, para podermos colocar a cantoneira da parte da frente, mediremos a distância entre as duas pernas e utilizaremos estas medidas para podermos marcar a distância que a cantoneira da frente ficará da cantoneira anterior. Antes de fazer os furos nos pés de madeirite, podemos apenas apoiar a segunda cantoneira e verificar se com isso as pernas ficarão exatamente paralelas. Assim, podemos fazer os furos nos pés do bípode e parafusar as cantoneiras, colocando os parafusos da parte de baixo para cima e rosqueando-o com a porca. O passo seguinte se refere à confecção dos “sapatos” para evitar que os parafusos toquem diretamente no chão. Retiraremos os pés do bípode para podermos marcar os furos a serem feitos nos sapatos, da seguinte maneira: apoiaremos o pé sobre o sapato deixando 1 cm de borda, então faremos as marcações. Em seguida, é só furá-lo e colar este sapato ao pé do bípode, utilizando cola branca.



Figura 23: Finalizando a construção dos pés do bípede. 1-Cantoneiras presas aos pés através de parafusos. 2-Dimensões dos pés e dos sapatos do bípede. 3-Os sapatos podem ser colados aos pés com cola branca. 4-Fixando os pés às pernas do bípede. 5-Bípede com os pés completos.

Para podermos levantar os pés, inclinando o bípede, utilizaremos um sistema composto por uma barra e duas hastes. Neste exemplo, a barra foi feita com um pequeno pedaço de alumínio, que deve ser fixada à engrenagem do servo-motor de inclinação. Esta barra foi criada no intuito de aumentar o ângulo de inclinação do bípede quando o servo-motor estiver em movimento. Foram feitos dois furos para encaixá-la aos parafusos da engrenagem e mais um furo em cada lateral para podermos colocar as hastes que erguerão os pés.

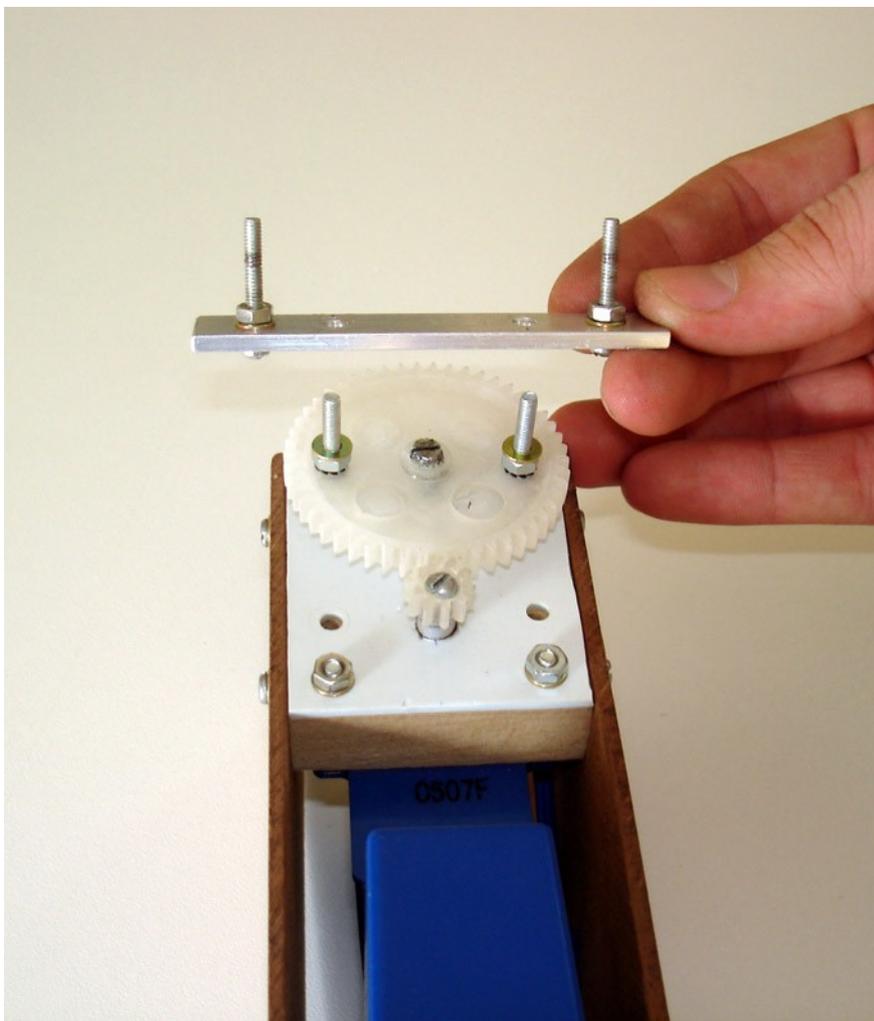


Figura 24: Barra destinada a fixar as hastes que erguerão os pés.

Estas hastes foram feitas com arame, mas podem ser substituídas por algum outro material, porém, note que devem ser rígidas o suficiente para aguentarem o peso do corpo do bípede, pois elas trabalharão de duas formas diferentes: ou empurrando o bípede, ou segurando o peso do corpo. Estas hastes ficarão presas em parafusos, que deverão ser colocados nos furos feitos na barra que foi fixada na engrenagem, e nos dois furos feitos nas extremidades das cantoneiras dos pés, como mostrado na figura.

O comprimento destas hastes foi medido sobre o próprio arame. É necessário desentortar os arames antes de encaixá-los nos furos, para que a distância entre as tiras seja uniforme. Isso pode ser feito com o auxílio de um martelo e uma base resistente, como uma morsa, ou a base da furadeira de bancada. Em suas extremidades fizemos dois ganchos, os quais deverão ser fechados, e devem possuir um diâmetro que permita com que o parafuso passe sem gerar atrito, porém, não

pode haver uma folga muito grande. Fazer estes ganchos pode vir a ser um pouco trabalhoso. Por usarmos um arame relativamente grosso (2 mm), fizemos uso de um alicate de bico para tal. Estas hastes devem possuir o mesmo comprimento. Elas deverão ser encaixadas nos parafusos apropriados e deverão ter uma folga para poderem trabalhar, pois o movimento realizado pelo bípede irá gerar uma angulação e esta angulação é quem vai nos informar qual será a folga necessária. Esta folga foi mantida com a utilização de duas porcas apertadas uma contra outra (como já visto anteriormente). Pode-se utilizar cola de secagem instantânea como opção para travar na posição ideal se for utilizada apenas uma porca.

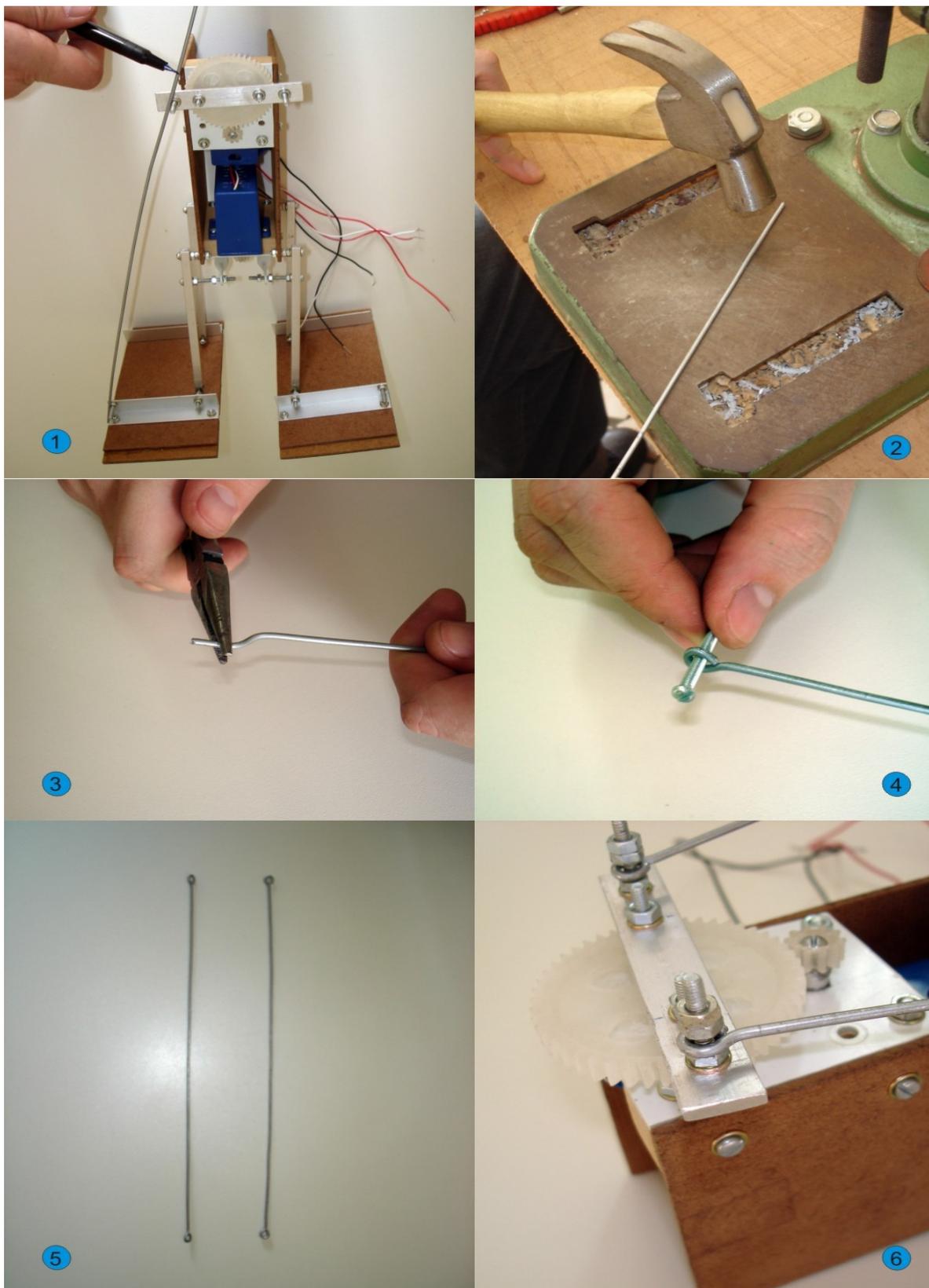


Figura 25: Colocando as hastes que erguerão os pés. 1-Definindo o tamanho. 2-As hastes deverão estar perfeitamente retas, o que pode ser feito com um martelo. 3-Fazendo o gancho com alicate de bico. 4-Verificando se o gancho está correto. 5-Hastes prontas. 6-Detalhe da fixação da haste à barra de alumínio.

Desta forma, se encerra por aqui o projeto mecânico do robô bípede. Passamos, neste instante, ao desenvolvimento da parte eletrônica e programação.

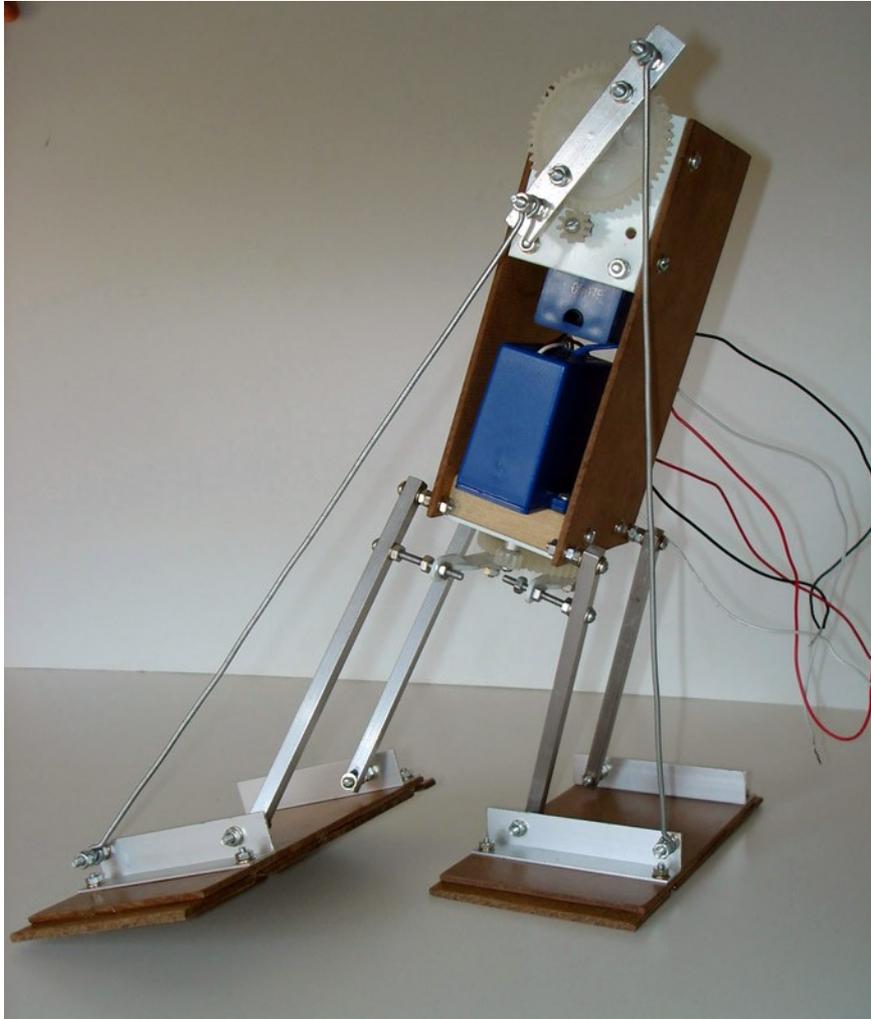


Figura 26: Bípede pronto.

6 – Eletrônica

A eletrônica envolvida com o bípede é simples. Serão descritos apenas alguns detalhes em relação à conexão do bípede, sua alimentação e o potenciômetro externo dos servo-motores.

Para conectar o bípede ao KDR5000, deverá ser utilizado o *Módulo de Entradas, Saídas e Servo-Motores*. É recomendado utilizar o primeiro conector de servo-motores disponível, tendo em vista que o MEC1000 possui uma quantidade de portas menor e o programa deverá ser compatível com os dois.

Para conectar o servo-motor de inclinação, é recomendado utilizar a porta Servo 0, enquanto que para o servo-motor de deslocamento, é recomendado a porta Servo 1. É possível ligar os servo-motores em outras portas, mas a conexão descrita acima foi padronizada para facilitar a programação. Caso haja dúvidas sobre a conexão, basta consultar os manuais desses equipamentos.

Não é necessária uma fonte de alimentação externa para o bípede, pois ele possui apenas dois servo-motores, consumindo uma corrente que pode ser suportada pela fonte própria do MEC1000 ou do KDR5000. Os servos podem ser ligados através das saídas de alimentação disponíveis nos conectores destinados ao seu controle.

Para conectar o bípede ao Kit, é recomendado utilizar um cabo de grande extensão, para que ele possa andar livremente. Também é interessante usar um conector (DB-9, por exemplo) para ligar esse cabo ao bípede, para facilitar o transporte do robô. Além disso, esse cabo pode ser utilizado para conectar o robô inseto ao Kit.

O potenciômetro externo dos servo-motores deverá ter seus terminais ligados de forma invertida em relação ao potenciômetro original, tendo em vista que a engrenagem a qual ele está ligado gira no sentido inverso ao eixo do servo-motor.

7 – Programação

Para que o bípede funcione perfeitamente, ele deverá ser controlado de modo adequado. Sendo assim, devemos criar um programa que faça com que o Kit acione os servo-motores de modo que o bípede possa se deslocar.

Através do programa a ser criado aqui, o Kit acionará os servo-motores de inclinação do corpo e deslocamento das pernas do bípede. De acordo com o modo que os servo-motores forem acionados, o bípede poderá ir para frente ou para trás ou virar para a direita ou esquerda. Também adicionaremos um comando extra para que ele mantenha-se na posição de repouso, o que é útil antes de desligá-lo.

O nosso programa será criado através do Delphi, utilizando o Projeto Base, que nos permite a comunicação com o Kit através de uma porta serial, como ponto de partida para a elaboração do programa, que deverá apresentar a seguinte interface.



Figura 27: Programa de controle do bípede.

AllowAllUp	=	True
GroupIndex	=	1
Name	=	SpeedButtonEsquerda
Caption	=	Esquerda
Width	=	57
AllowAllUp	=	True
GroupIndex	=	1
Name	=	SpeedButtonRepouso
Caption	=	Repouso
Width	=	57
AllowAllUp	=	True
GroupIndex	=	1

Desse modo, nosso Form ficará da seguinte maneira. Observe o modo como os SpeedButton foram colocados.



Figura 30: Form após a adição dos SpeedButton para controle do bípede.

Nosso próximo passo será adicionar os componentes para iniciar e parar a movimentação do bípede. Isso será feito através de componentes Button, que se encontram na aba "Standard" da barra de componentes.



Figura 31: Aba "Standard" da Barra de componentes

O componente Button possui o seguinte ícone.



Figura 32: Ícone do componente Button.

Vamos adicionar dois componentes desses e modificar as seguintes propriedades do botão para iniciar o bípede.

Name = ButtonIniciar
Caption = Iniciar
Font/Style/fsBold = true

Em seguida as propriedades do botão para parar o bípede.

Name = ButtonParar
Caption = Parar
Font/Style/fsBold = true

Com isso, o nosso Form apresentará o seguinte aspecto.



Figura 33: Form após a adição dos componentes para ligar e desligar o bípede.

Sendo assim, finalizamos a interface gráfica do nosso programa e podemos partir para a implementação de suas linhas de código. Aqui, desenvolveremos o código responsável pelo controle dos servo-motores, possibilitando a movimentação do bípede. Para facilitar a criação do programa, é recomendado seguir as instruções de ligação dos servo-motores descritas no trecho referente à Eletrônica.

Devido às características da própria estrutura do bípede, os parâmetros utilizados nesse tutorial para a movimentação dos servo-motores não serão definitivos, mas devem ser estabelecidos de acordo com a sua construção. Deve-se tomar cuidado para que eles não girem demais, o que pode danificá-los. Como eles estão presos às estruturas de movimentação do bípede, não será possível que eles façam o movimento completo, o que forçaria o servo-motor.

O programa funcionará da seguinte forma: criaremos uma função chamada SleepAndTest e uma variável chamada Stop, sendo que os dois poderão ter apenas os valores True (verdadeiro) e False (falso). Quando a variável Stop for True, a função SleepAndTest será falsa e vice-versa. A função SleepAndTest terá dois objetivos: gerar um “delay” (pausa na execução do programa). Essa pausa será muito clara durante o funcionamento do bípode, pois enquanto ele desloca o seu corpo para o lado, o programa estará nessa pausa, e só depois ele desloca sua perna.

O outro objetivo dessa função será verificar se o botão Parar foi pressionado. Isso é feito verificando o estado da variável Stop. Quando apertamos o botão Iniciar, o valor dessa variável é definido como False. Sendo assim, a função SleepAndTest retornará um valor verdadeiro, o que é um requisito para a execução do programa. Ao apertar o botão Parar, a variável Stop é definida como verdadeira, o que acarretará na interrupção da execução do programa.

O nosso primeiro passo então será criar a variável Stop. Isso será realizado da seguinte forma: apertamos F12, mudando do Form para a Unit Main. Na parte superior dessa Unit, antes do código que foi criado no Tutorial Base, temos a declaração das variáveis globais da Unit. A única variável global presente no momento é a FormMain. Declararemos então a variável Stop, que será do tipo Boolean (True/False). Desse modo, a linha de código do programa ficará assim:

```
...  
var  
    FormMain: TFormMain;  
    Stop: Boolean;  
implementation  
...
```

Devemos agora criar a função SleepAndTest. Vamos nos dirigir ao final da Unit, nas linhas de código criadas no Tutorial Base. Antes do end final do programa (**end.**) criaremos a função SleepAndTest. Ela será do tipo Boolean, ou seja, retorna um valor verdadeiro ou falso. Inicialmente, criaremos a função desse modo:

```
...  
function SleepAndTest (tempo: integer): Boolean;  
begin  
    end;  
  
end.
```

Essa função depende do parâmetro Tempo, que será do tipo Integer, cujo objetivo será explicado a seguir. Devemos declarar uma variável e uma constante locais, ou seja, que serão utilizadas apenas nessa função, chamadas i, que será do tipo Integer, e Tempobase, cujo valor será 10.

```
...
```

```

function SleepAndTest (tempo: integer): Boolean;
var
    i: integer;
const
    Tempobase = 10;
begin

end;

end.

```

Vamos agora implementar nossa função SleepAndTest, fazendo com que ela gere o “delay” e verifique o valor da variável Stop. Para isso, utilizaremos uma estrutura de repetição **for** e uma estruturas de comparação **if-else**, que comparará o valor de Stop com True e False. Deste modo, nossa linha de código ficará assim:

```

...
function SleepAndTest (tempo: integer): Boolean;
var
    i: integer;
const
    Tempobase = 10;
begin
    for i:=1 to tempo do
        begin
            Application.ProcessMessages;
            if Stop = true then
                begin
                    // Se o botão Parar foi pressionado, a variável Stop
                    // será verdadeira e a função retornará um valor falso.
                    SleepAndTest := false;
                end
            else
                begin
                    // A função Sleep pausa o programa durante um tempo
                    Sleep(Tempobase);
                    // Se o botão Parar não foi pressionado, a função re-

```

```

        // tornará um valor verdadeiro
        SleepAndTest := true;
    end;
end;
end;

end.

```

Após criar a função, vamos analisá-la. Primeiramente, adicionamos uma estrutura de repetição **for**. As instruções contidas nessa estrutura serão executadas até que o valor da variável *i*, que foi iniciada com o valor 1, seja igual ao valor da variável *tempo*, lembrando-se que a cada repetição o valor da variável *i* será incrementada em uma unidade. Temos então a função `Application.ProcessMessages`. Essa função processa todas as tarefas pendentes no programa, sendo que nesse caso o principal objetivo é atualizar a interface e verificar se algum botão foi pressionado na interface gráfica do programa e, caso isso tenha ocorrido, as linhas de código referentes a esse botão serão executadas.

Temos depois uma estrutura de comparação **if-else**, que verificará o valor da variável `Stop`. Se `Stop` for verdadeiro, o que acontecerá ao pressionar o botão `Parar`, então a função `SleepAndTest` retornará um valor falso. Mas se a variável `Stop` for falsa, então `SleepAndTest` será verdadeiro, o que é uma condição essencial para a execução do programa, como veremos a seguir. Além disso temos a função `Sleep`. Essa função pausa o programa durante um tempo em milisegundos. Como nossa constante `Tempobase` foi definida como 10, a cada vez que essa função `Sleep` for acionada, haverá uma pausa de 10 ms.

Mas lembre-se de que essa função `Sleep` está dentro de um laço **for**, cujo número de repetições é definido pelo parâmetro `tempo`. Se declararmos nossa função dessa maneira: `SleepAndTest (100)`; então `tempo` será 100 e, conseqüentemente, haverão 100 repetições no laço **for**. Se a variável `Stop` for falsa, a função `Sleep` será executada 100 vezes, gerando então um “delay” de 1 segundo.

O próximo passo será então implementar o código que será executado ao pressionarmos o botão `Iniciar`. Para isso, basta dar um duplo clique sobre ele no nosso `Form`, o que abrirá o seguinte manipulador na nossa `Unit`:

```

...
procedure TFormMain.ButtonIniciarClick(Sender: TObject);
begin
end;

end.

```

Nesse programa utilizaremos uma única variável local, chamada de `BipedeRepouso` que será utilizada mais adiante para manter o bípede em posição de repouso. Vamos então declará-la.

```

...

```

```

procedure TFormMain.ButtonIniciarClick(Sender: TObject);
var
    BipedeRepouso: integer;
begin

end;

end.

```

Também devemos declarar as constantes locais, que nesse caso serão apenas duas, usadas para identificar os servo-motores. Será atribuído a elas um valor correspondente a sua posição dos respectivos servo-motores nas suas portas. Sendo assim, para um correto funcionamento do programa, é recomendado seguir as instruções de conexão descritas no trecho referente à Eletrônica. Caso haja alguma dúvida, basta consultar os manuais do MEC1000 ou do KDR5000, onde estão presentes as pinagens das portas de Servo-motores.

```

...
procedure TFormMain.ButtonIniciarClick(Sender: TObject);
var
    BipedeRepouso: integer;
const
    ServoInclina = 0;
    ServoDesloca = 1;
begin

end;

end.

```

Agora vamos implementar o código a ser executado quando o botão Iniciar for pressionado. Como utilizaremos a função SleepAndTest frequentemente, e esta depende da variável Stop, nosso primeiro passo deverá ser atribuir um valor a essa variável, que deverá ser False.

Também devemos inicializar a variável BipedeRepouso, atribuindo a ela o valor zero. O código então ficará assim:

```

...
procedure TFormMain.ButtonIniciarClick(Sender: TObject);
var
    Repouso: integer;

```

```

const
    ServoInclina = 0;
    ServoDesloca = 1;
begin
    // Ao pressionar o botão iniciar, a variável Stop será
    // falsa
    Stop:= false;
    // Inicialização da variável BipedeRepouso
    BipedeRepouso:= 0;
end;

end.

```

Devemos agora nos dedicar ao trecho principal da linha de código do programa, onde serão executados os comandos para a movimentação do bípede de acordo com o SpeedButton selecionado. As ações relacionadas a sua movimentação devem se repetir até que o botão Parar seja pressionado, sendo assim, precisamos de uma estrutura de repetição, que no nosso caso será um **while**. Ele dependerá do valor da função SleepAndTest, ou seja, o nosso programa estará preso nesse laço **while** enquanto a função SleepAndTest retornar um valor verdadeiro. Caso o botão Parar seja pressionado, o que fará com que SleepAndTest retorne um valor falso, esse laço **while** será encerrado e o programa se encaminhará ao código de finalização do bípede.

Criando esse laço **while**, nosso programa ficará assim:

```

...
begin
    // Ao pressionar o botão iniciar, a variável Stop será
    // falsa
    Stop:= false;
    // Inicialização da variável BipedeRepouso
    BipedeRepouso:= 0;

    while SleepAndTest(1) = true do
        begin
            end;
        end;

end.

```

Agora que criamos o laço **while**, devemos implementar o código a ser executado enquanto nosso programa estiver preso nele. O programa deverá verificar qual SpeedButton foi pressionado e executar as linhas de código referentes a ele, para realizar a movimentação do bípede. Devemos então utilizar estruturas de comparação **if**, sendo cada uma relacionada a um SpeedButton. Vamos criar os blocos **if** referente aos SpeedButton de direção. O bloco **if** do SpeedButton Repouso é ligeiramente diferente e será feito depois.

```
...
begin
    // Ao pressionar o botão iniciar, a variável Stop será
    // falsa
    Stop:= false;
    // Inicialização da variável BipedeRepouso
    BipedeRepouso:= 0;

    while SleepAndTest(1) = true do
    begin
        // Verifica se o botão "Frente" foi pressionado
        if (SpeedButtonFrente.Down = true) then
        begin
            end;

        // Verifica se o botão "Trás" foi pressionado
        if (SpeedButtonTras.Down = true) then
        begin
            end;

        // Verifica se o botão "Esquerda" foi pressionado
        if (SpeedButtonEsquerda.Down = true) then
        begin
            end;

        // Verifica se o botão "Direita" foi pressionado
        if (SpeedButtonDireita.Down = true) then
        begin
            end;
        end;
    end;
end;
```

end.

Após criarmos os bloco acima, vamos nos dedicar a elaborar o código a ser executado quando o SpeedButton Frente estiver pressionado, fazendo com que o bípede desloque-se para a frente. Mas, antes de tudo, devemos analisar como movimentar corretamente os servo-motores.

Como foi comentado anteriormente, o funcionamento do bípede depende tanto da sua estrutura mecânica como da programação envolvida no seu controle. Para entender o problema a ser descrito aqui, vamos analisar o modo como o bípede foi construído. Os servo-motores movimentarão as duas engrenagens grandes, sendo que uma é responsável por inclinar o seu corpo e outra por deslocar as pernas, lembrando-se de cada uma está presa a um potenciômetro.

Ambos os servo-motores não podem fazer o movimento total que é possível, sob o risco de danificar tanto a estrutura quanto o próprio motor. Os limites máximos de movimentação dependem muito de como o bípede foi construído. O servo-motor destinado a inclinar o corpo do bípede deverá levantar cada perna de modo que ela fique apenas levemente acima da superfície. Caso o servo-motor incline demais o corpo, ele pode forçar as hastes de arame, danificando tanto elas quanto o próprio motor.

Já o outro servo-motor deverá deslocar a perna um pouco para frente. O seu deslocamento máximo depende basicamente da estrutura formada pela engrenagem e pernas dianteiras. Não podemos forçar essa estrutura, sob o risco de danificar o motor e a estrutura.

Devemos primeiramente estabelecer os parâmetros a serem enviados para o deslocamento dos eixos dos servo-motores. Através da função ServoMotorOn, enviamos um valor que varia entre 0 e 255 que, no caso dos servo-motores originais, serve para controlar a posição do seu eixo. Como estamos utilizando um servo-motor adaptado, cujo potenciômetro é externo, esse valor tem a função de controlar a posição da engrenagem presa ao potenciômetro.

Não podemos utilizar todos os valores possíveis entre 0 e 255 como parâmetro para o posicionamento da engrenagem. Como sua movimentação será limitada, valores muito altos ou muito baixos podem fazer com que o servo-motor force a estrutura. Como os valores limites dependem de como o bípede foi construído, os valores máximos e mínimos que podemos utilizar sem danificar o servo-motor variam.

Os valores definidos aqui nesse tutorial devem servir apenas como base na implementação de seu programa. Por isso, esteja atento ao funcionamento do bípede. Caso o movimento de algum servo-motor seja insuficiente ou seja forçado (o que pode-se perceber através do seu barulho), basta mudar o valor enviado através da função ServoMotorOn.

Vamos voltar agora a implementação das linhas de código do nosso programa. Dentro do nosso bloco **if**, a primeira coisa a fazer será inclinar o corpo do bípede. Para isso, utilizaremos a função ServoMotorOn, que terá como primeiro parâmetro a constante referente ao motor de inclinação. Já o segundo parâmetro, como foi explicado acima, deve ter um valor suficiente para que a perna seja levantada um pouco acima do solo, sem movimentar demais a engrenagem. Utilizamos o valor 50 como parâmetro, mas ele poderá ser modificado, caso seja insuficiente ou seja demais.

```
...  
while SleepAndTest(1) = true do  
begin
```

```

// Verifica se o botão "Frente" foi pressionado
if (SpeedButtonFrente.Down = true) then
begin
    // Inclina o corpo do bípede para um lado
    kit.ServoMotorOn(ServoInclina,50);
end;
...

```

O próximo passo, após levantar a perna, será deslocá-la para a frente. Devemos então movimentar o servo-motor de deslocamento, utilizando novamente a função ServoMotorOn. O segundo parâmetro dessa função deverá ser um valor que suficiente para que a perna seja deslocada para a frente sem danificar o motor ou as estruturas. Aqui, foi utilizado o valor 180, mas convém lembrar que ele pode ser modificado caso não seja adequado.

Mas antes que essa função seja utilizada, devemos gerar uma pausa no programa, o que será feito através da função SleepAndTest. Utilizando o valor 100 como parâmetro, geramos um atraso de 1 segundo. Essa pausa é extremamente necessária, pois ela ocorrerá enquanto o servo-motor estiver inclinando o corpo do bípede. Caso não utilizarmos essa pausa, a inclinação do corpo e o deslocamento da perna ocorrerão simultaneamente, causando problemas.

```

...
while SleepAndTest(1) = true do
begin
    // Verifica se o botão "Frente" foi pressionado
    if (SpeedButtonFrente.Down = true) then
        begin
            // Inclina o corpo do bípede para um lado
            kit.ServoMotorOn(ServoInclina,50);
            // Gera um atraso de 1 segundo
            SleepAndTest(100);
            // Desloca a perna livre para a frente
            kit.ServoMotorOn(ServoDesloca,180);
        end;
end;
...

```

Agora devemos inclinar o corpo do bípede para o outro lado, concentrando seu peso na perna que acabou de ser deslocada. Antes, geramos uma outra pausa de um segundo, para que a perna possa ser deslocada. Utilizamos então a função ServoMotorOn tendo como primeiro parâmetro a constante ServoInclina. Só que desta vez, utilizamos outro valor como segundo parâmetro. Esse valor deve ser suficiente para inclinar o corpo do bípede para o outro lado, deixando uma perna levemente acima do chão. Nesse caso, utilizamos o valor 220.

Vamos também gerar uma outra pausa de 1 segundo no nosso programa, que ocorrerá enquanto o servo-motor de inclinação estiver atuando.

```
...
while SleepAndTest(1) = true do
begin
  // Verifica se o botão "Frente" foi pressionado
  if (SpeedButtonFrente.Down = true) then
  begin
    // Inclina o corpo do bípede para um lado
    kit.ServoMotorOn(ServoInclina,50);
    // Gera um atraso de 1 segundo
    SleepAndTest(100);
    // Desloca a perna livre para a frente
    kit.ServoMotorOn(ServoDesloca,180);
    // Gera um atraso de 1 segundo
    SleepAndTest(100);
    // Inclina o corpo do bípede para outro lado
    kit.ServoMotorOn(ServoInclina,220);
    // Gera um atraso de 1 segundo
    SleepAndTest(100);
  end;
end;
...
```

Sendo assim, temos uma perna livre para ser deslocada, o que será feito utilizando novamente a função `ServoMotorOn`. O primeiro parâmetro será `ServoDesloca`, indicando qual motor será acionado. Já o segundo parâmetro deverá ser suficiente para que a perna seja movimentada para a frente, lembrando-se novamente de que não podemos forçar o servo-motor, utilizando um valor muito alto.

Depois dessa função, também necessitamos gerar um atraso de um segundo, através da função `SleepAndTest`.

```
...
while SleepAndTest(1) = true do
begin
  // Verifica se o botão "Frente" foi pressionado
  if (SpeedButtonFrente.Down = true) then
  begin
    // Inclina o corpo do bípede para um lado
```

```

    kit.ServoMotorOn(ServoInclina,50);
    // Gera um atraso de 1 segundo
    SleepAndTest(100);
    // Desloca a perna livre para a frente
    kit.ServoMotorOn(ServoDesloca,180);
    // Gera um atraso de 1 segundo
    SleepAndTest(100);
    // Inclina o corpo do bípede para outro lado
    kit.ServoMotorOn(ServoInclina,220);
    // Gera um atraso de 1 segundo
    SleepAndTest(100);
    // Desloca a perna livre para a frente
    kit.ServoMotorOn(ServoDesloca,120);
    // Gera um atraso de 1 segundo
    SleepAndTest(100);
end;
...

```

Já temos pronta a programação referente ao movimento do bípede para frente, podendo nos dedicar ao movimento para trás. A sequência de movimentos é extremamente semelhante, tendo como única diferença o fato de que a perna que estiver livre será deslocada para trás.

Um modo muito simples de implementar o código do SpeedButtonTras é copiar o que foi feito para deslocar o bípede para a frente e apenas trocar os valores utilizados como segundo parâmetro para o controle do servo-motor de deslocamento. O bloco **if** desse SpeedButton ficará assim.

```

...
// Verifica se o botão "Trás" foi pressionado
if (SpeedButtonTras.Down = true) then
begin
    // Inclina o corpo do bípede para um lado
    kit.ServoMotorOn(ServoInclina,50);
    // Gera um atraso de 1 segundo
    SleepAndTest(100);
    // Desloca a perna livre para trás
    kit.ServoMotorOn(ServoDesloca,120);
    // Gera um atraso de 1 segundo
    SleepAndTest(100);

```

```

// Inclina o corpo do bípede para outro lado
kit.ServoMotorOn(ServoInclina,220);
// Gera um atraso de 1 segundo
SleepAndTest(100);
// Desloca a perna livre para trás
kit.ServoMotorOn(ServoDesloca,180);
// Gera um atraso de 1 segundo
SleepAndTest(100);
end;
...

```

Os próximos códigos a serem implementados são os comandos para virar à esquerda e à direita. Os procedimentos adotados para virar o bípede são um pouco diferentes. O primeiro passo será manter o bípede com as duas pernas no chão, utilizando como segundo parâmetro da função ServoMotorOn algum valor entre 50 e 220 (que foram utilizados para incliná-lo) suficiente para isso. No exemplo descrito nesse tutorial, o valor ideal foi 100.

Com as duas pernas do bípede no chão, quando acionarmos o servo-motor de deslocamento, ele virará para um lado. A direção para a qual o bípede vira depende do valor utilizado para acionar o servo-motor. Esses valores podem ser os mesmos usados anteriormente. Ao usarmos 120, ele virará para a esquerda e ao usarmos 180, ele virará para a direita.

Após realizar esses movimentos, podemos notar que uma das pernas ficou atrás da outra. Para que seja possível virar o bípede novamente, devemos levá-la para a frente, utilizando os mesmos procedimentos de inclinar e deslocar a perna para a frente.

A implementação completa do SpeedButtonEsquerda é mostrada a seguir.

```

...
// Verifica se o botão "Esquerda" foi pressionado
if (SpeedButtonEsquerda.Down = true) then
begin
// Deixa o bípede com as duas pernas no chão
kit.ServoMotorOn(ServoInclina,100);
// Gera um atraso de 1 segundo
SleepAndTest(100);
// Desloca o bípede para a esquerda
kit.ServoMotorOn(ServoDesloca,120);
// Gera um atraso de 1 segundo
SleepAndTest(100);
// Inclina o corpo do bípede o lado
kit.ServoMotorOn(ServoInclina,50);

```

```

// Gera um atraso de 1 segundo
SleepAndTest(100);
// Desloca a perna livre para a frente
kit.ServoMotorOn(ServoDesloca,180);
// Gera um atraso de 1 segundo
SleepAndTest(100);
end;
...

```

A implementação do SpeedButtonDireita é mostrada a seguir.

```

...
// Verifica se o botão "Direita" foi pressionado
if (SpeedButtonDireita.Down = true) then
begin
// Deixa o bípede com as duas pernas no chão
kit.ServoMotorOn(ServoInclina,100);
// Gera um atraso de 1 segundo
SleepAndTest(100);
// Desloca o bípede para a direita
kit.ServoMotorOn(ServoDesloca,180);
// Gera um atraso de 1 segundo
SleepAndTest(100);
// Inclina o corpo do bípede o lado
kit.ServoMotorOn(ServoInclina,220);
// Gera um atraso de 1 segundo
SleepAndTest(100);
// Desloca a perna livre para a frente
kit.ServoMotorOn(ServoDesloca,120);
// Gera um atraso de 1 segundo
SleepAndTest(100);
end;
end;
end;

end.

```

Com os comandos de movimentação feitos, podemos agora implementar o bloco **if** do SpeedButtonRepouso. Ele será utilizado para manter o bípede com as duas pernas no chão, sendo um procedimento útil para ser feito antes de desligá-lo.

Esse bloco **if** será um pouco diferente dos outros. Como ele só precisa ser feito uma única vez, devemos utilizar mais uma condição, além do fato do SpeedButtonRepouso estar pressionado. É aí que precisamos da variável BipedeRepouso.

A outra condição para que esse bloco **if** seja executado será de que a variável BipedeRepouso deverá ser diferente de 1, o que pode ser feito através do operador \neq . Ao pressionar o botão Iniciar, será atribuído o valor 0 a essa variável. Sendo assim, caso o SpeedButtonRepouso estiver pressionado, as linhas de código referentes a ele serão executadas. Só que no final do bloco **if**, vamos atribuir o valor 1 à variável. Sendo assim, teremos certeza de que esse bloco não será executado novamente, a não ser que o botão Iniciar seja pressionado de novo.

O código para o SpeedButtonRepouso começará assim:

```
...
// Verifica se o botão "Repouso foi pressionado e se
// BipedeRepouso é diferente de 1
if (SpeedButtonRepouso.Down = true) and (BipedeRepouso <> 1) then
begin
end;
end;
end;

end.
```

O primeiro passo será alinhar as pernas do bípede. Devemos então inclinar seu corpo para um lado e deslocar a perna livre de modo que ela fique ao lado da perna a qual o bípede se apóia. Como no nosso exemplo utilizamos os valores 180 e 120 para deslocá-la para frente e para trás, o valor adequado deverá estar aproximadamente entre eles. O valor 150 foi o mais adequado para a utilização nesse exemplo. Não podemos nos esquecer das pausas de 1 segundo no programa.

```
...
// Verifica se o botão "Repouso foi pressionado e se
// BipedeRepouso é diferente de 1
if (SpeedButtonRepouso.Down = true) and (BipedeRepouso <> 1) then
begin
// Inclina o corpo do bípede para o lado
kit.ServoMotorOn(ServoInclina,50);
// Gera um atraso de 1 segundo
SleepAndTest(100);
```

```

        // Alinha a perna livre com a qual o bípede se apóia
        kit.ServoMotorOn(ServoDesloca,150);
        // Gera um atraso de 1 segundo
        SleepAndTest(100);
    end;
end;
end;

end.

```

Desse modo, as pernas estarão alinhadas, mas o corpo do bípede ainda estará inclinado. Devemos então acionar o servo-motor de inclinação novamente, utilizando como parâmetro um valor adequado para deixá-lo com as duas pernas no chão. Esse valor deve ser o mesmo que foi utilizado para virar o bípede, que no exemplo foi 100.

Sendo assim, completamos os procedimentos para manter o bípede em posição de repouso. Agora podemos atribuir à variável BípedeRepouso o valor 1, evitando que esses procedimentos se repitam.

```

...
// Verifica se o botão "Repouso foi pressionado e se
// BípedeRepouso é diferente de 1
if (SpeedButtonRepouso.Down = true) and (BípedeRepouso <> 1) then
begin
    // Inclina o corpo do bípede para o lado
    kit.ServoMotorOn(ServoInclina,50);
    // Gera um atraso de 1 segundo
    SleepAndTest(100);
    // Alinha a perna livre com a qual o bípede se apóia
    kit.ServoMotorOn(ServoDesloca,150);
    // Gera um atraso de 1 segundo
    SleepAndTest(100);
    // Deixa o bípede com as duas pernas no chão
    kit.ServoMotorOn(ServoInclina,50);
    // Valor 1 atribuído à variável, evitando que os
    // procedimentos de repouso se repitam
    BípedeRepouso:= 1;
end;
end;

```

```
end;
```

```
end.
```

O nosso próximo passo agora será criar as rotinas para o desligamento do bípede. Essas linhas de código serão escritas depois do **end;** do laço **while** principal, aquele ao qual nosso programa estará preso enquanto a função `SleepAndTest` for verdadeira. Pode parecer estranho à primeira vista, mas elas serão mesmo escritas no procedure do botão Iniciar, mas somente serão executadas quando a função `SleepAndTest` for falsa, pois assim o nosso programa sairá do laço **while** principal. Devemos desligar todos os servo-motores através da função `ServoMotorOff`.

```
...
```

```
end;
```

```
// As funções a seguir desligarão o bípede e serão execu-  
// tadas quando o programa sair do loop while principal,  
// ou seja, quando SleepAndTest retornar um valor falso
```

```
// Desliga os motores
```

```
kit.ServoMotorOff(ServoInclina);
```

```
kit.ServoMotorOff(ServoDesloca);
```

```
end;
```

```
end.
```

Nosso programa está quase pronto. Basta agora implementar o código a ser executado quando o botão Parar for pressionado. Voltamos então ao nosso Form, apertando a tecla F12, e damos um duplo clique sobre esse botão, abrindo o seguinte manipulador.

```
...
```

```
procedure TFormMain.ButtonPararClick(Sender: TObject);
```

```
begin
```

```
end;
```

```
end.
```

A única coisa que precisamos fazer é atribuir à variável `Stop` o valor `True`, o que fará com que a função `SleepAndTest` retorne um valor falso e, conseqüentemente, o bípede seja desligada.

```
...
```

```
procedure TFormMain.ButtonPararClick(Sender: TObject);
```

```
begin
```

```
// Quando o botão for pressionado, a variável Stop será ver-  
// dadeira, conseqüentemente a função SleepAndTest será fal-  
// sa e o funcionamento do bípede será interrompido  
    Stop := true;  
end;  
  
end.
```

Chegamos então ao fim da programação para o funcionamento do bípede. Se todos esses passos forem seguidos corretamente, o programa poderá ser compilado e executado.

8 – Dicas

Podemos acrescentar algumas funcionalidades a mais ao do bípede, exercitando assim a criatividade e os conhecimentos de Robótica adquiridos. Em relação ao programa de controle do bípede, pode-se implementar o código para que o movimento do bípede seja modificado automaticamente quando algum SpeedButton for pressionado, sem a necessidade de se apertar o botão Iniciar.

Também é possível fazer com que o bípede fique em posição de repouso automaticamente quando o botão Parar for pressionado.

O bípede mesmo pode ser incrementado. Com alguns conhecimentos de Eletrônica e Programação, é possível fazer uma versão que siga ou fuja da luz, utilizando LDRs. Com um sensor de distância é possível fazer coisas bem interessantes também, como uma versão que não caia em buracos (caso o bípede esteja em cima de uma mesa), com o sensor apontado para a frente num ângulo de 45 graus.