



Universidade Federal de Minas Gerais  
Departamento de Ciência da Computação

## Introdução à Robótica

Trabalho Prático 1

Programação utilizando a HandyBoard

**Grupo 7:**

Celina Gomes  
Lucas Frossard  
Ludmila Villela

**Professor:**

Mário Fernando Montenegro Campos

Belo Horizonte, 23 de setembro de 2008

# 1 Objetivos

Desenvolver um dispositivo mecânico com LEGO que translate e rotacione e seja controlado apenas pela HandyBoard. O robô deve ser capaz de executar uma seqüência para 4 leds e se mover livremente durante um período exato e realizar autonomamente uma trajetória quadrada de 30cm de lado e uma trajetória circular de 25 cm de raio, ambos por 3 vezes consecutivas.

## 2 Etapas do desenvolvimento

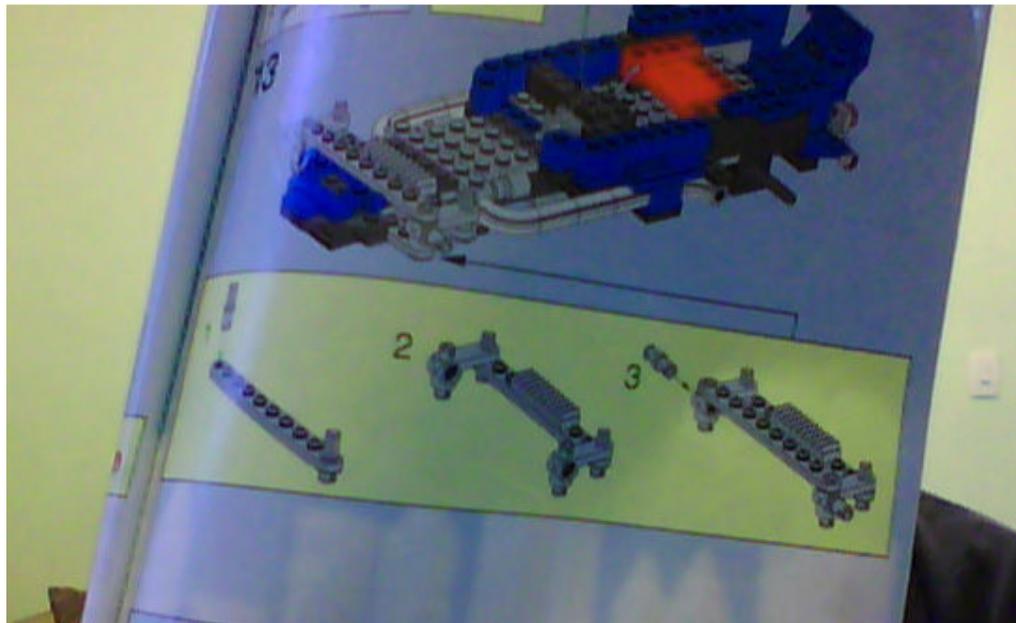
- 7 de setembro

Primeiro encontro para familiarização com a HandyBoard e o InteractiveC.

Foi realizado um ajuste fino da altura dos motores para que o mesmo se encaixasse de forma harmônica com as outras peças do lego.

Sobre a estrutura física:

Decidimos que a estrutura física do robô seria da seguinte forma: O robô teria um total de quatro rodas em dois eixos, como num carro. O eixo da frente seria responsável por direcionar o robô e o traseiro responsável pelo movimento de translate.



**Figura 2.1: O modelo do eixo frontal foi retirado do manual de um lego.**

Colocamos então o primeiro motor que seria responsável pelo movimento de translate.

### Sobre a estrutura lógica:

Nesta etapa foi desenvolvida boa parte do código da HandyBoard incluindo o menu e o acionamento e desligamento dos motores.

- **11 de setembro**

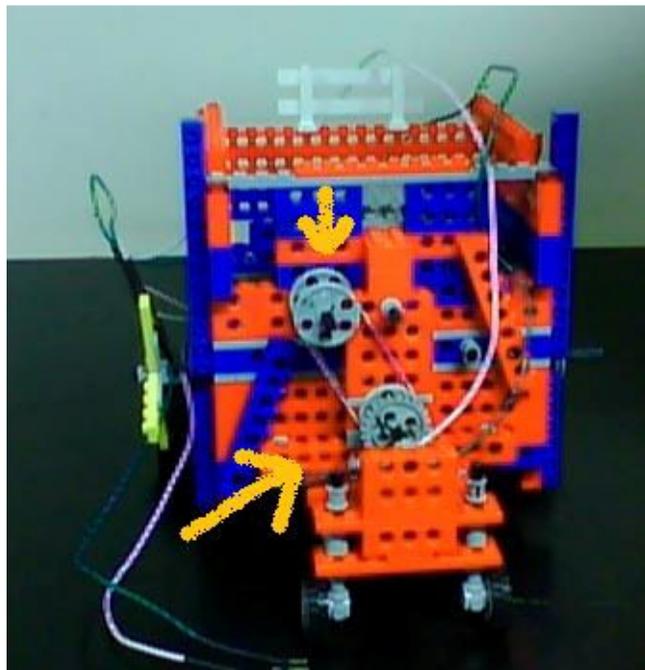
Nesta fase começamos a implementar a tarefa relacionada aos leads e continuamos a trabalhar na estrutura física do *Johnnie Walker*. A tarefa de translação linear começou a ser implementada também.

- **12 de setembro**

Nesta etapa terminamos a estrutura física do *Johnnie Walker* e continuamos a investigar como seriam feitas as ligações dos leds e a programação da tarefa.

### Sobre a estrutura física

Na Figura 2.2 pode-se observar a estrutura finalizada, muito embora ela tenha sido destruída dando lugar a uma estrutura completamente nova utilizando o roll-on.



**Figura 2.2: Estrutura inicial**

Nesta estrutura havia dois motores, um responsável pelo movimento de translação e outro responsável por controlar o eixo frontal, que direcionava o robô.

O motor de translação estava acoplado ao eixo traseiro, em que as rodas traseiras eram ligadas, de modo que o lado esquerdo e o lado direito sempre teriam a mesma velocidade angular. Este método garantia que o Johnnie conseguiria andar reto, mas este projeto apresentou uma falha que será abordada mais adiante.

O motor de rotação funcionava da seguinte maneira: O motor, que estava com duas reduções utilizando o sem-fim, girava uma estrutura que virava uma engrenagem e que por fim estava associada a uma cremalheira. A cremalheira estava ligada num eixo móvel, como em um carro, que direcionava as rodas frontais. As roldanas eram responsáveis pela transmissão do movimento, como está indicado na Figura 2.2.

- **14 de setembro**

Nesta fase começamos os testes com o robô.

Primeiramente terminamos a programação dos leads, corrigimos alguns problemas encontrados no código e começamos a testar a estrutura do robô. A programação para calibração do motor também foi terminada.

Começamos então a testar todo o conjunto que completava o Johnnie, isto é, ligamos a HandyBoard na estrutura lego e o robô movimentou-se para a frente e para trás. O motor do eixo traseiro não necessitaria de uma calibração de sincronismo dado que as rodas deste eixo estavam diretamente ligadas. Pequenas modificações na estrutura do robô mostraram-se necessárias.

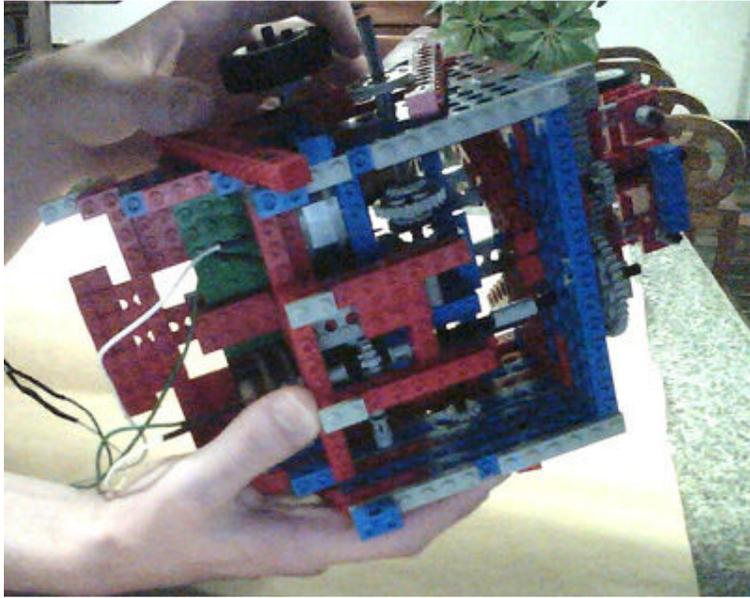
A tarefa dos leads foi testada e funcionou corretamente.

- **16 de setembro**

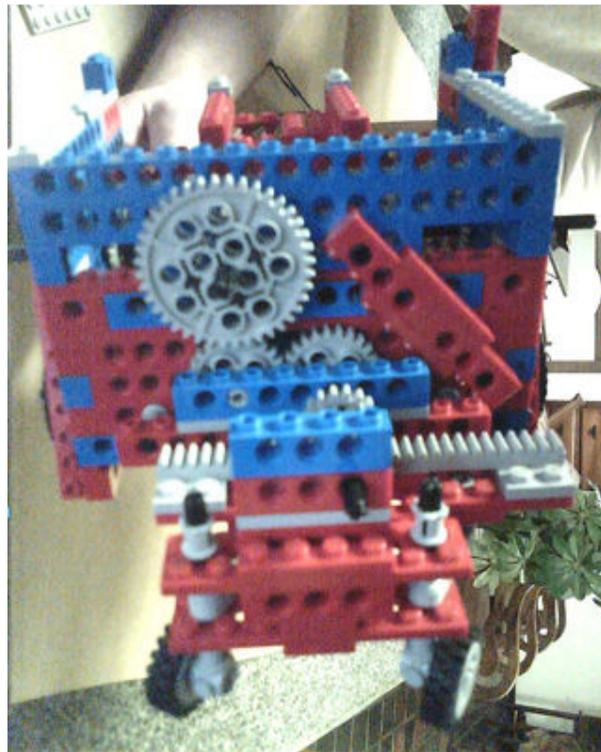
Nesta etapa a idéia era calibrar o motor responsável por direcionar o Johnnie Walker. Porém ao começar os testes uma falha que passou despercebida tornou-se evidente. Independente da direção que colocávamos o eixo o frontal, o robô insistia em andar somente reto. Isto ocorreu devido ao fato das duas rodas do eixo traseiro estarem completamente ligadas. Desta forma, ambas sempre tinham a mesma velocidade angular, o que gerava um atrito forte quando o robô tentava fazer uma curva. Este atrito impedia que o robô fizesse a curva corretamente e o mesmo ia “quicando” para frente.

Tentamos colocar mais atrito no eixo frontal, mas foi insuficiente para que o robô fizesse a curva. Ainda que funcionasse, o atrito nas rodas traseiras era grande e culminava numa enorme resistência que desgastaria demais a roda do lego. Mostrou-se evidente a necessidade de mudar a mecânica do robô (Figura 2.3).

Nesta etapa também descobrimos alguns cuidados que não tomamos quanto a ligação dos leads. Não estávamos usando resistores, muito embora isto não tenha se mostrado um problema. Porém, fomos convencidos de que o uso do mesmo era necessário para que limitássemos a corrente que passava nos leads (diodos).



(a)



(b)



(c)

Figura 2.3 (a) (b) (c): Estrutura que foi refeita

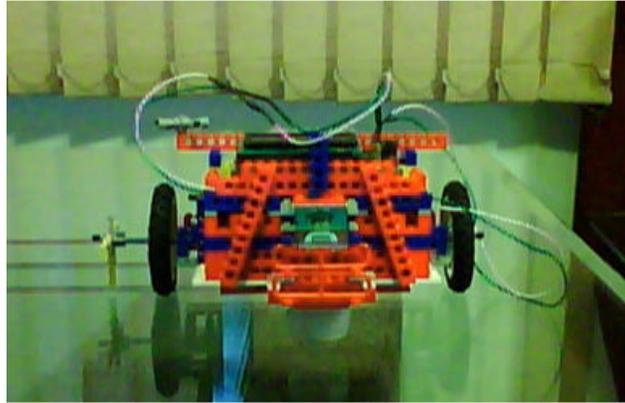
- **17 de setembro**

Nesta fase mudamos completamente a estrutura do robô e fizemos pequenos testes com a nova estrutura, que foram bem sucedidos. Também começamos a reprogramar a parte de calibragem e de multi-tarefas do robô.

Sobre a estrutura física

Tínhamos duas opções para a estrutura física do Johnnie: Poderíamos mudar a tração do eixo traseiro para o eixo da frente ou colocar cada motor controlando uma roda do próprio eixo traseiro.

Estávamos convencidos de que a primeira opção resolveria o nosso problema, porém a mecânica seria complicada e mais trabalhosa do que a exigida para a segunda opção. Optamos, portanto, pela segunda opção, que demandaria uma calibragem para a sincronia dos dois motores (Figura 2.4).



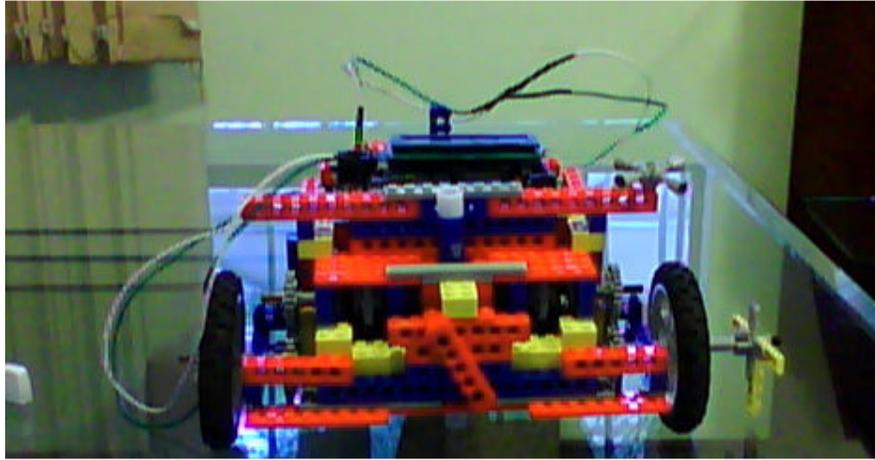
(a)



(b)



(c)



(d)

Figura 2.4 (a) (b) (c) (d): Nova Estrutura

- 21 de setembro

Com a estrutura do robô completamente definida, nos dedicamos nesta fase somente ao código do programa e à calibração.

Foram feitos todos os testes necessários à calibração dos motores, tanto para movimento linear quanto rotacional (Tabela 2.1).

Tabela 2.1: Testes de calibração

Potências		Tempo	Distancia1	Distancia2	Distancia3	Distancia4	Distancia5	Distancia6	Distancia7	Distancia8	Distancia9	Distancia10
Direita	Esquerda		cm									
49	59	1.092	34.3	31	32.3	31.5	34.1	33.2	34.1	31.5	32.9	32.5
49	59	1.95	61	53.3	67	61.6	55.7	55.3	58.8	60.3	59.4	62
49	59	0.858	30	30.5	31.5	30.2	31.6	31.1	30.4	30.5	31.6	31
49	59	2.964	95.2	95.5	90.5	90.9	92.5	93.5	91	91.5	92.8	93.8
49	59	3.9	122	120	128	130	124	133	127	126	123	125
38	50	2.28	38.75	36	37.5	36.5	38.2	36.4	38.1	36.5	36.3	37
38	50	1.014	17.5	18.2	16.2	16.7	17	16.8	17.1	18.1	17.8	16
38	50	1.07	19	18.7	19.8	20	18.8	19.2	19.8	18.3	19.4	19.6
38	50	1.248	20	20.4	19.8	19	22.5	21.4	21.2	20.5	19.7	21.1
38	50	0.78	13	13	12.5	12.7	13.1	12.7	12.9	13	12.6	12.7
34	46	1.14	17.6	18.7	15	16.6	18.5	17.8	16.8	17.9	15.2	14.9
34	46	0.78	8.2	11	10.4	8.5	9	10.2	11.1	10.6	8.5	9.2
34	46	2.28	30	27.9	32.9	21.7	28.5	31.2	32.6	28.1	29.2	30
34	46	3.42	47	45.5	46	45.9	48.4	48.2	46.5	47.8	48	47.9
34	46	3	36.2	36.8	37.2	38	36.4	37.5	38	37.8	37.1	36.7

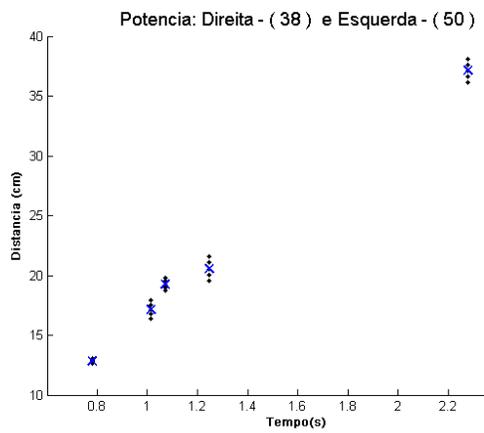
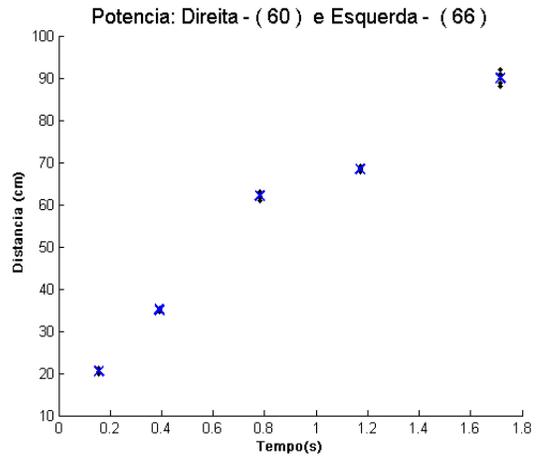
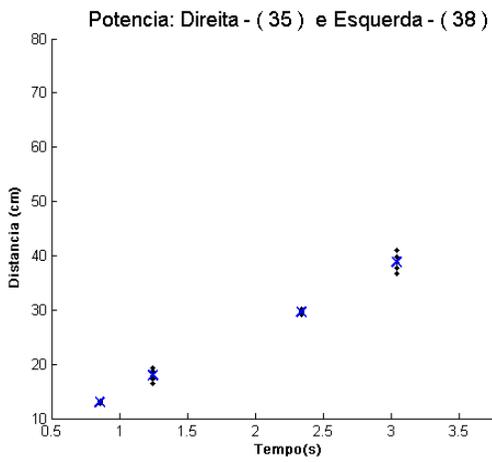
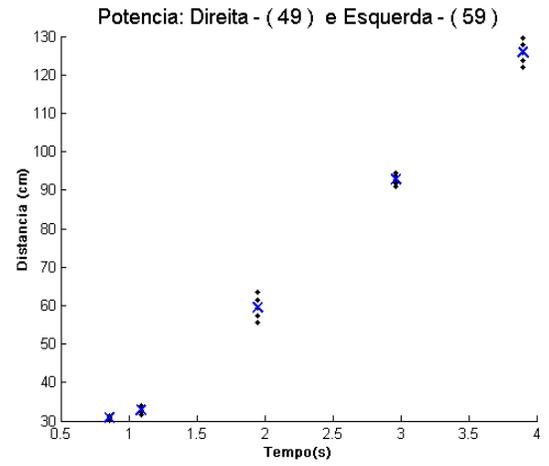
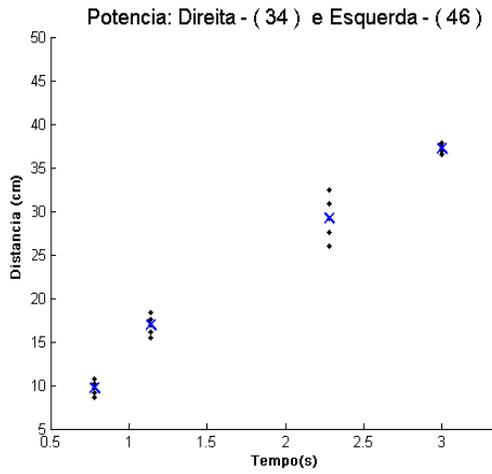
60	66	1.716	92	87.5	91.3	88.5	91.8	88.1	92.3	88.2	87.9	91.1
60	66	1.17	68.7	67.6	69.1	69	67.4	68.3	68.2	67.9	68	69.2
60	66	0.78	60.3	62	64	61.2	63	62.3	61.5	61.4	62	63.1
60	66	0.39	35	35.3	34.5	34.7	35.1	35.8	34.9	35.7	34.8	35.1
60	66	0.156	21.2	20.3	19.4	21.1	20.6	20.1	21.3	19.7	19.9	21
35	38	3.042	40	35.4	41	36	39.3	40.2	41.9	38	37.2	39
35	38	2.34	29.2	29	30	29.5	30.1	29.7	29.3	30	29.5	30.3
35	38	1.248	16.2	20	18	17.2	19.2	16.7	20.1	18.2	16.3	17.1
35	38	0.858	12.7	13	13.2	13.1	12.8	12.9	13.3	12.6	12.8	13
35	38	3.9	70.2	71.7	78	75.1	72.3	76.9	72.7	74.1	70.3	75.1
<b>Rotação</b>												
<b>Direita</b>	<b>Esquerda</b>	<b>Tempo</b>	<b>Angulo</b>									
40	-40	1.95	121	133	124	130	127	130	125	132	128	126
40	-40	1.4	110	110	105	109	111	107	108	106	110	104
40	-40	1.014	84.2	93	83	87.1	82	88	92.4	87	88	91
35	-35	0.546	24	27	24	24	23.8	26.2	26	25	28	25
35	-35	1.014	50	47.5	49	48	46.2	49	46.2	48	51	46
35	-35	0.7	26	29	34	33.6	28	27	31	32	27	30
50	-50	0.7	72	79	86	75	82	73	77	85	83	71
50	-50	0.39	52	49	51	48	50	47	49	49.2	50	50.2
50	-50	1.014	141	142	133	140	137	138	139	143	135	138

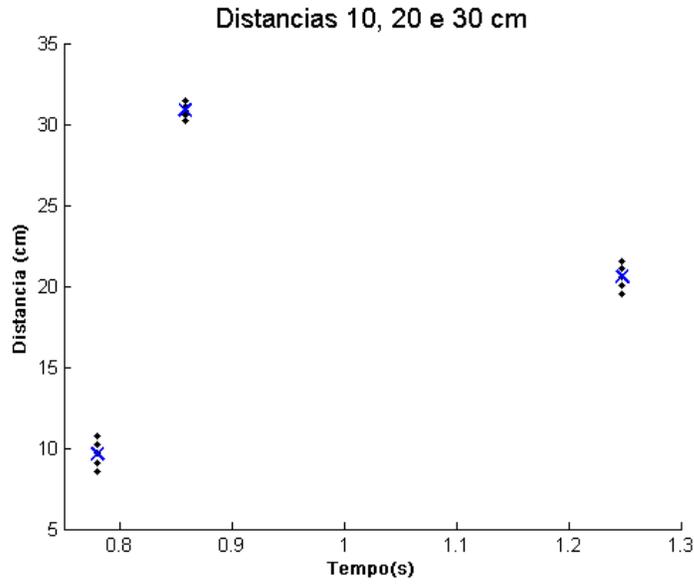
Tabela 2.2: Valores para as distâncias 10, 20 e 30 e os ângulos 30, 60 e 90

Angulo	Direita	Esquerda	Tempo
90	35	-35	1.326
60	35	-35	1.42
30	35	-35	0.7
Distancia	Direita	Esquerda	Tempo
30 cm	35	38	2.34
20cm	60	66	0.156
10cm	34	46	0.78

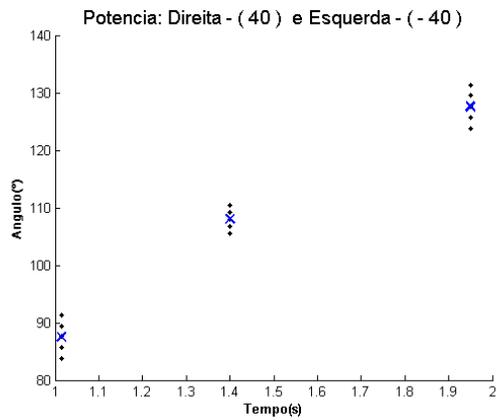
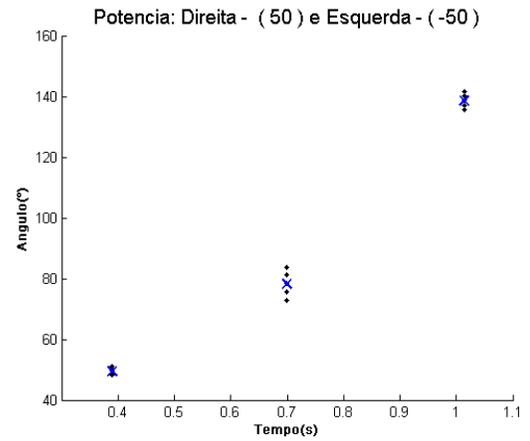
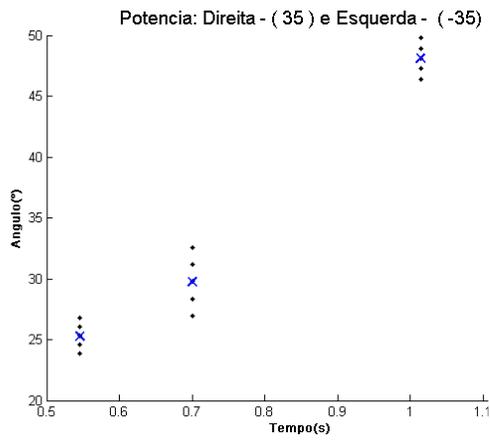
Com os valores obtidos foram plotados gráficos *distancia x tempo* e *ângulo x tempo* para cada par de potências escolhida. Para cada distância/ângulo e tempo foi plotado no gráfico a média das medições com um (X) e uma reta de pontos (...) em torno da média correspondendo ao desvio padrão.

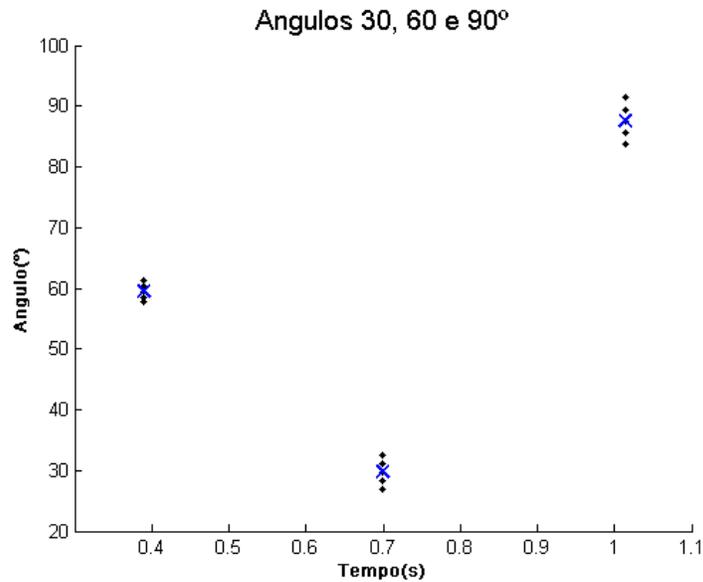
## Calibração linear:





### Calibração angular:





Realizada a calibração salvamos os valores da Tabela 2.2 no programa da HandyBoard que possa realizar o movimento novamente.

- **23 de setembro**

Nesta última etapa foram realizados os testes para os desenhos do quadrado e do círculo e os ajustes finais.

### 3 Estrutura Física

A versão final do robô contém 2 motores, cada um ligado a uma roda do eixo traseiro de forma independente. A parte da frente continha um roll on.

Foi necessária uma calibragem para determinar qual potência que cada motor deveria receber para que o mesmo andasse reto e fizesse uma rotação de 30, 60 e 90 graus em torno do próprio do eixo.

A Handy Board foi colocada numa estrutura feita para acomodá-la no topo do Johnnie. Uma estrutura apropriada foi construída para os leds e colocada ao lado do robô.

## 4 Estrutura Lógica

### 4.1 Estrutura de Dados

Para melhor organização do código, assim como para facilitar o controle de certos dados, foram criadas duas estruturas.

A primeira, Motor, é a abstração de um motor, a qual armazena a porta a qual ele se encontra ligado, bem como sua potência atual. O interesse de se armazenar a potência atual é o de facilitar a mudança de potência de forma gradativa. Além disso, evita-se o

uso de variáveis globais, além de possibilitar o reuso de certas funções de forma mais eficiente.

A segunda estrutura criada, Config, possui a configuração de uma trajetória qualquer. Ela armazena a potência desejada de cada motor, além do tempo em que essas potências devem ser mantidas. Isso permite que toda trajetória seja realizada por uma só função, pois depende somente das variáveis armazenadas.

Para a calibragem e execução das tarefas, foram criadas 9 variáveis persistentes do tipo Config:

- *linear10*, *linear20* e *linear30*: Armazenam as configurações para trajetórias lineares de comprimento 10, 20 e 30cm.
- *angle30*, *angle60* e *angle90*: Armazenam as configurações para o robô girar em 30, 60 e 90 graus.
- *circle25*: Configuração para realizar um círculo de 25cm de raio.

## 4.2 Módulos

O código do trabalho foi dividido em vários módulos, cada um relacionado a uma tarefa ou funcionalidade específica. O método main está contido no arquivo hello.ic, bem como todos os "includes".

**hello.ic:** Além de conter o main, esse módulo é responsável pela impressão do menu principal, bem como pela interação com o usuário. Assim que uma opção do menu é escolhida, uma função do módulo correspondente é chamada. Essa função é sempre no formato "startX", sendo X a tarefa escolhida.

**leds.ic:** O módulo gerencia os estados das saídas digitais para os quatro leds embutidos no robô. Além disso, a função relacionada a execução da tarefa de leds também se encontra nesse módulo. A função é responsável por realizar uma sequência aleatória de funcionamento dos leds.

**linear.ic:** Dentro deste, encontra-se a função de execução do quadrado, a qual nada mais é do que chamar alternativamente a execução das tarefas de translação linear de 30cm e de rotação de 90 graus. Além de possibilitar ao usuário abortar a tarefa a qualquer momento, a partir do acionamento do botão "stop".

**rotation.ic:** Novamente, o módulo é responsável pela realização de uma tarefa, realizar o círculo de raio de 25cm. Também é feito o controle de parada da tarefa via o botão "stop".

**thread.ic:** O seguinte módulo dispara dois processos, um para funcionamento aleatório dos leds e outro para a realização de uma tarefa qualquer. São duas as condições de parada, o botão "stop" e um marcador de tempo. O marcador é setado para executar a tarefa no máximo por 30s, caso não abortada pelo usuário.

**motor.ic:** Possui métodos associados aos motores. Foram criadas funções para ajustar uma nova potência de um motor e de dois de forma gradativa, assim como para desligar o mesmo. Dessa forma, evitam-se mudanças abruptas de velocidade. Além disso, foi criada uma função `execute()`, a qual é responsável por executar uma tarefa com uma potência "x" para o motor direito, uma potência "y" para o esquerdo em um intervalo "t" de tempo.

**calibrate.ic:** Esse módulo é responsável pelas tarefas de calibragem do motor, bem como do tempo. Utiliza-se uma variável temporária para armazenar as configurações setadas pelo usuário e permite que o mesmo faça testes com as mesmas. Caso seja de interesse do usuário, ele pode salvar tais configurações em alguma das estruturas persistentes.

**time.ic:** Possui uma função de sleep diferente da disponível pelo IC, a qual permite que o usuário aborte sua execução a qualquer momento. Para isso, realiza-se uma espera ocupada, ou seja, utiliza-se um `while` e, a cada interação, são feitos cálculos de quanto tempo já se passou desde o início da chamada da função. Caso o tempo seja igual ou maior o passado, a função termina e o fluxo de controle volta para o método anterior. O mesmo ocorre caso o botão "stop" seja pressionado.

Além disso, os menus de cada tarefa são gerenciados pelos módulos associados às mesmas.

## 4.3 Usabilidade

A interação do usuário com o robô foi ajustada para ser feita por meio de um menu, do potenciômetro e dos botões "stop" e "start". Cada botão tem as seguintes funções:

**.start:** Sempre utilizado para confirmar/executar a opção impressa no display.

**.stop:** Utilizado para alternar entre as opções quando no menu, para cancelar o ajuste de alguma variável durante a fase de calibragem e para abortar as tarefas a qualquer momento.

**.potenciômetro:** Utilizado para setar os valores associados à variáveis no menu de configuração, tempo e potência.

O menu inicial possui 6 opções:

**0- Quadrado:** Inicia a tarefa de realizar 3 quadrados.

**1- Rotação:** Execução da trajetória circular de 25 cm de raio.

**2- Leds:** Executa uma seqüência de funcionamento aleatório dos leds.

**3- Multi-tarefa:** Inicia tanto uma trajetória qualquer, quanto o funcionamento dos leds.

**4- Calibrar:** Chama o menu de calibragem.

**5- Tarefas:** Chama o menu de execução de trajetórias armazenadas nas variáveis persistentes.

Além do menu principal, a calibragem possui um menu auxiliar, com 6 opções:

**0- Potencia Motor Direito:** Permite o ajuste da potência a ser usada no motor direito.

**1- Potencia Motor Esquero:** Permite o mesmo ajuste mas para o motor esquerdo.

**2- Tempo:** Ajuste do tempo de execução da trajetória.

**3- Teste:** Executa a trajetória com a configuração escolhida.

**4- Salvar Config:** Salva a configuração atual em uma das variáveis persistentes.

**5- Voltar:** Volta para o menu anterior.

Tanto a opção "Tarefas" do menu principal, quanto a opção "Salvar Config", chamam um terceiro menu, o qual permite selecionar a variável persistente alvo. As opções são as variáveis citadas anteriormente, bem como a opção de voltar ao menu anterior.

## 5 Desafios

Encontramos os seguintes desafios nesta etapa do trabalho:

- A calibragem não era exata e a variação da carga da bateria contribuía para que esta variação se tornasse maior de maneira que uma calibragem que servia num determinado momento não serviria para um momento posterior.

- Para a primeira estrutura do robô foi desafiador encontrar uma maneira de fazê-lo virar. Problema este que foi contornado alterando toda a estrutura do robô.

- Para este trabalho fizemos um ajuste fino da altura do motor. No primeiro TP isto foi um grande problema, que resolvemos neste trabalho. Além disso, travamos bem o motor e utilizamos o sem-fim para obter uma redução maior.

- Sincronizar os dois motores. Para a estrutura final fizemos duas tentativas. Na primeira utilizamos o recurso multitarefa do robô. Porém o overhead gerado pela troca de processos fazia com que o robô andasse uma trajetória em S. A melhor opção foi ligar os dois motores como duas instruções seguidas, cada uma ligando um motor.

## 6 Conclusão

A tarefa de calibrar os motores é difícil e trabalhosa. Várias medições devem ser feitas até ter certeza de que determinados valores são os adequados para uma determinada tarefa. Ainda assim, estes valores podem, quando usados diversas vezes, podem apresentar diversos resultados diferentes, com uma variação significativa entre

os resultados extremos. Estes resultados também variam de acordo com a carga da bateria o que torna a tarefa ainda mais complicada.

Tal calibragem, apesar disto, contribuiu para compreender o funcionamento do motor e da HandyBoard e obter valores aproximados para que o robô executasse uma determinada tarefa, como a tarefa de desenhar um quadrado e um círculo.

Também percebemos que quanto menor fosse a potência e conseqüentemente menor fosse a velocidade do robô, mais reta era a linha que obtínhamos e menor era a variação nas diversas medições para uma dada potência e um tempo.