

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA

LUIS GUILHERME MACHADO CAMARGO  
MARCELO TEIDER LOPES  
MATHEUS SILVA ARAÚJO

**ROBÔ EXPLORADOR DE AMBIENTES**

MONOGRAFIA

CURITIBA

2011

**LUIS GUILHERME MACHADO CAMARGO  
MARCELO TEIDER LOPES  
MATHEUS SILVA ARAÚJO**

## **ROBÔ EXPLORADOR DE AMBIENTES**

Monografia apresentada ao Departamento Acadêmico de Eletrônica da Universidade Tecnológica Federal do Paraná como requisito parcial para aprovação na Disciplina de Oficina de Integração 2.

Orientadora: Profa. Dra. Myriam Regattieri De Biase da Silva Delgado

**CURITIBA**

**2011**

## **AGRADECIMENTOS**

Este trabalho não teria sido possível sem o projeto anteriormente apresentado por Bruno Meneguele, Fernando Padilha e Vinicius Arcanjo. Por emprestar o robô e pelos diversos esclarecimentos (muitas vezes sobre assuntos que não os envolviam) nosso muito obrigado.

À Professora Myriam nosso agradecimento por aceitar o desafio de nos orientar e a atenção dispensada.

Aos Professores Hugo Vieira e Mário Sérgio pela oportunidade sem par de aprendizado.

Aos Professores João Fabro e Celso Kaestner por permitirem o uso das dependências e recursos do Laboratório de Arquitetura de Computadores para a realização do projeto.

Aos colegas Lucas Campiolo Paiva e Cláudio Akio pela colaboração ocasional com o projeto.

Aos marceneiros do Almoxarifado da UTFPR pela ajuda com a construção do suporte da câmera.

- **1ª lei:** Um robô não pode ferir um ser humano ou, por inação, permitir que um ser humano sofra algum mal.
- **2ª lei:** Um robô deve obedecer às ordens que lhe sejam dadas por seres humanos, exceto nos casos em que tais ordens contrariem a Primeira Lei.
- **3ª lei:** Um robô deve proteger sua própria existência, desde que tal proteção não entre em conflito com a Primeira e Segunda Leis.
- **Lei Zero:** Um robô não pode fazer mal à humanidade e nem, por inação, permitir que ela sofra algum mal.

*Isaac Asimov*

## RESUMO

CAMARGO, Luis Guilherme M. ; LOPES, Marcelo Teider; ARAÚJO, Matheus Silva. ROBÔ EXPLORADOR DE AMBIENTES. 35 f. Monografia – Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2011.

Neste projeto foi realizada uma tentativa de desenvolver um robô explorador ambientes capaz de encontrar um objeto pré-definido em um ambiente controlado, ou explorar todo o ambiente caso não seja capaz de encontrá-lo. Posteriormente, o escopo do projeto foi alterado para um robô capaz de localizar um objeto em seu campo de visão e segui-lo, como passo inicial para a construção de um robô explorador. Para tal usamos como sensor principal uma câmera, a *CMUCam3*, desenvolvida pela *Carmegie Mellon University*. Uma bússola também foi estudada como sensor auxiliar, para obter a direção do robô. A versão inicial do robô foi desenvolvido em (MENEGUELE et al., 2011).

**Palavras-chave:** Robótica, Exploração, Reconhecimento de Imagens, Sensores ...

## ABSTRACT

CAMARGO, Luis Guilherme M. ; LOPES, Marcelo Teider; ARAÚJO, Matheus Silva. AMBIENCE EXPLORER ROBOT. 35 f. Monografia – Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2011.

An attempt to develop an explorer robot is made in this project, which would be capable of both finding a predefined object on a small, controlled, environment, and exploring the whole environment if it can't find it. Later the scope of the project was changed to a robot capable of finding an object at his line of vision and following it, as a inicial step in order to build an explorer robot. A camera was used as the main sensor, the *CMUCam3*, developed by *Carmegie Mellon University*. A compass would also be used as an auxiliary sensor, in order to obtain the direction of the robot. The initial version of the Robot is developed in (MENEGUELE et al., 2011).

**Keywords:** Robotics, Exploration, Image Recognition, Sensors ...

## LISTA DE FIGURAS

FIGURA 1	– Foto do robô .....	9
FIGURA 2	– Diagrama do Robô .....	12
FIGURA 3	– Processo de Comunicação - Arduino .....	15
FIGURA 4	– Diagrama - Biblioteca <i>Robot</i> .....	19
FIGURA 5	– Circuito Eletrônico - Bússola .....	21
FIGURA 6	– Histograma e Detecção de Bordas de algumas imagens exemplo .....	24

## LISTA DE TABELAS

TABELA 1	– Pinos da placa <i>Arduino Duemilanove</i> .....	16
TABELA 2	– Sistema de Indicação .....	16
TABELA 3	– Acionamento dos Motores .....	18
TABELA 4	– Comunicação Arduino-Câmera .....	19
TABELA 5	– Acionamento dos Motores .....	22



## SUMÁRIO

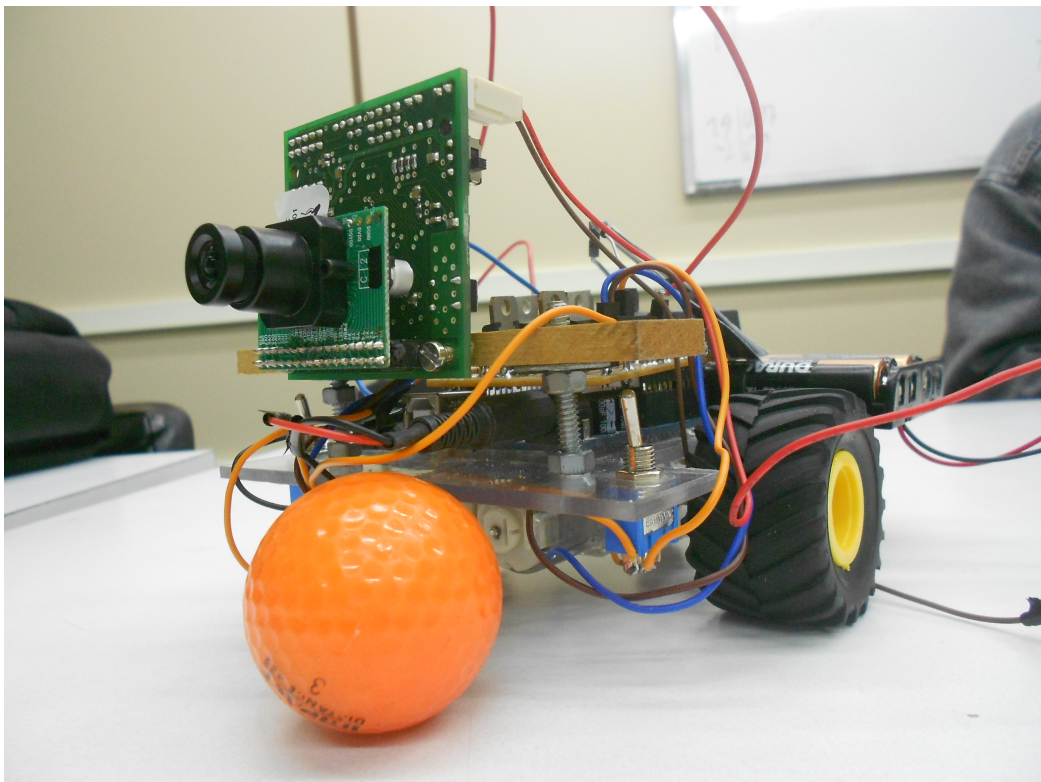
<b>1</b>	<b>INTRODUÇÃO</b>	<b>9</b>
1.1	MOTIVAÇÃO	10
1.2	OBJETIVO	10
1.2.1	Objetivo Geral	10
1.2.2	Objetivos Específicos	10
<b>2</b>	<b>DESCRIÇÃO DO ROBÔ</b>	<b>12</b>
2.1	PROJETO MECÂNICO	12
2.1.1	Sistema de Alimentação	13
2.1.2	Sistema de Indicação	13
2.1.3	Sistema de Tração	13
2.1.4	Sistema de Controle	14
2.1.5	Sensores	14
2.2	PLATAFORMA ARDUINO	15
2.2.1	Comunicação com Indicadores	15
2.2.2	Comunicação com Bússola	16
2.2.3	Acionamento dos Motores	17
2.2.4	Comunicação com a CMUCam	18
2.3	SOFTWARE DE CONTROLE	18
2.3.1	Compass	19
2.3.2	Motor	20
2.3.3	CommSerial	20
2.3.4	Indicator	20
2.3.5	Robot	20
2.3.6	Constants	20
<b>3</b>	<b>SENSORES</b>	<b>21</b>
3.1	BÚSSOLA	21
3.2	CÂMERA	22
<b>4</b>	<b>VISÃO</b>	<b>23</b>
4.1	IDENTIFICAÇÃO DE OBJETOS DE INTERESSE	23
4.2	IDENTIFICAÇÃO DE OBSTÁCULOS	23
4.3	CORREÇÃO DE DESVIOS	24
<b>5</b>	<b>NAVEGAÇÃO</b>	<b>25</b>
5.1	CONSTRUÇÃO DO MAPA	25
5.2	ROTEAMENTO	26
<b>6</b>	<b>CONCLUSÃO</b>	<b>27</b>
	<b>REFERÊNCIAS</b>	<b>28</b>
	<b>Apêndice A – CADERNO DE BORDO</b>	<b>29</b>
	<b>Apêndice B – DESCRITIVO</b>	<b>34</b>

## 1 INTRODUÇÃO

Robôs autônomos são aqueles capazes de realizar um objetivo desejado em um determinado ambiente sem a intervenção humana. Dentre os robôs autônomos, robôs exploradores são aqueles que têm como objetivo explorar um ambiente até atingir uma finalidade específica, um ponto ou um objeto dentro do espaço a ser explorado, por exemplo, sem se perder ou colidir com obstáculos.

Para atingir seus objetivos, esses robôs utilizam sensores para percepção do ambiente e algoritmos inteligentes para tomada de decisões.

O robô construído pela equipe pode ser visto na Figura 1.



**Figura 1: Foto do robô**

## 1.1 MOTIVAÇÃO

Robôs autônomos exploradores têm diversas áreas de aplicação, desde atividades em hospitais a explorações planetárias.

A possibilidade de aplicar diversas áreas de conhecimento do curso, de eletrônica à inteligência artificial, motivou a equipe a escolher esse projeto. O subsídio de um robô já desenvolvido e a possibilidade de utilizar um sensor inteligente (*CMUCam3*) viabilizaram o projeto.

## 1.2 OBJETIVO

O objetivo inicial do projeto foi a construção de um robô explorador, utilizando uma câmera e uma bússola como sensores. Ele deveria ser capaz de identificar um objeto específico e posteriormente explorar o ambiente onde se encontra a procura do objeto, caso este saia de seu campo de visão.

### 1.2.1 OBJETIVO GERAL

Para o projeto atual, o ambiente de exploração será limitado a um retângulo de 168,2 cm por 118,9 cm, dimensões somadas de duas folhas tamanho A0, que serão utilizadas lado a lado para composição da arena de exploração. A cor branca das folhas irá determinar o chão para o robô. Eventualmente, podem ser colocados referenciais de fácil identificação (círculos de determinadas cores, por exemplo) nas extremidades da arena para melhor auto-localização do robô.

### 1.2.2 OBJETIVOS ESPECÍFICOS

Foram definidas inicialmente as seguintes etapas para alcançar o objetivo final:

- Sistema de Controle do Robô;
- Obter leituras da Bússola com o *Arduino*;
- Reconhecimento do objeto a ser encontrado;
- Comunicação entre *Arduino* e *CMUCam*;
- Reconhecimento de Pontos de Interesse no ambiente;

- Integração da Bússola com *CMUCam*;
- Algoritmo de Navegação;
- Detecção de Obstáculos.

## 2 DESCRIÇÃO DO ROBÔ

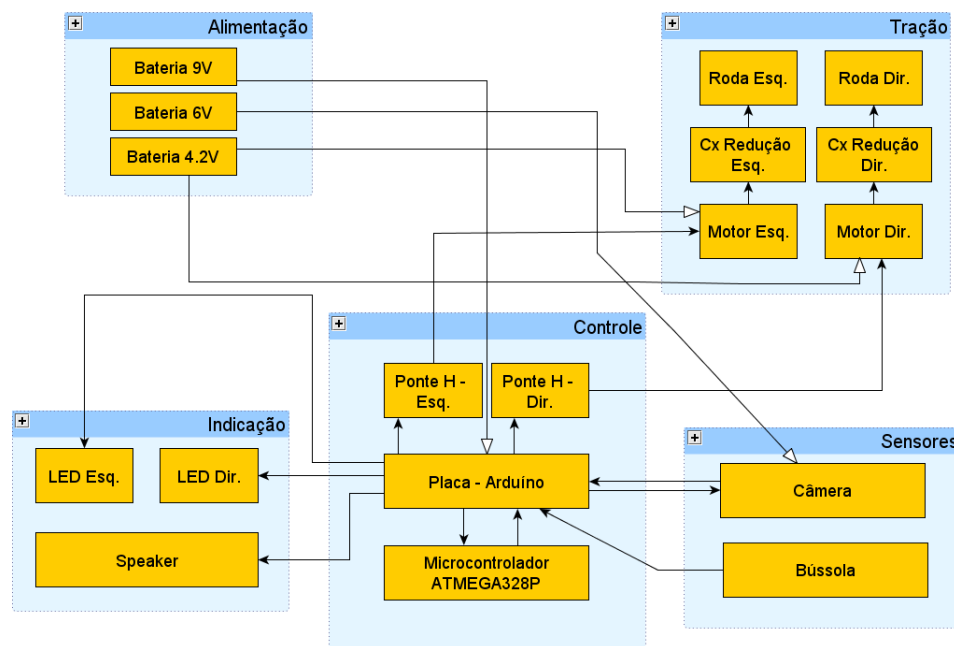
O robô utilizado no projeto é o mesmo robô construído durante o projeto **Robô Explorador de Labirintos 2D** (MENEGUELE et al., 2011) desta mesma disciplina.

No projeto original o robô era utilizado para explorar e solucionar labirintos em duas dimensões feitos através de trilhas pretas em um chão branco, utilizado emissores e sensores de luz infravermelha para identificar a pista.

Todo o projeto mecânico foi reutilizado neste trabalho, incluindo rodas, caixa de redução e chassi. Foram reutilizados também o sistema de alimentação e a uma placa *Arduino Duemilanova*; o conjunto de sensores do robô original foi substituído por uma câmera *CMUCam3*.

### 2.1 PROJETO MECÂNICO

O diagrama do projeto físico do robô é apresentado na Figura 2.



**Figura 2: Diagrama do Robô**

### 2.1.1 SISTEMA DE ALIMENTAÇÃO

Conjunto de baterias utilizadas como alimentação do robô.

- **Bateria de 9V**

Utilizada para alimentação do *Arduino Duemilanove*, uma bateria PP3.

- **Bateria de 4,2V**

Usada na alimentação dos motores, bateria de câmera fotográfica digital.

- **Bateria de 6V**

Usada na alimentação da CMUCam3, quatro pilhas AA em série.

### 2.1.2 SISTEMA DE INDICAÇÃO

LEDs e *Speaker* usados para indicar as ações do robô.

- **LEDs**

Usados para indicação dos estados do robô, detalhados na Tabela 2

- **Speaker**

Utilizado como indicador sonoro de estados específicos do sistema, como reconhecimento do objeto e início e fim da busca do mesmo.

### 2.1.3 SISTEMA DE TRAÇÃO

Para tração do robô foi construído um sistema baseado em dois motores elétricos, cada um acionando uma roda.

- **Motores**

Motor elétrico de corrente contínua *Mabuchi FA-130RA* de 3V com rotação de 12300 rpm, ou 205 voltas por segundo (MENEGUELE et al., 2011)

- **Caixas de redução**

As caixas de redução, acopladas ao motor e às rodas, reduzem a rotação do motor para que seja possível acionar as rodas. Na configuração usada, a redução é de 344:1 (MENEGUELE et al., 2011).

- **Rodas**

O robô utiliza duas rodas *off-road* em seu centro e uma esfera com giro livre atrás para manter o equilíbrio.

Com a rotação de 205 voltas/segundo e a redução de 344:1, a roda completa 0,6 voltas por segundo.

#### 2.1.4 SISTEMA DE CONTROLE

Sistema para controle da movimentação do robô.

- **Ponte H**

A Ponte H é um circuito que permite a um microcontrolador acionar um motor de corrente contínua. Por questões eletrônicas (MENEGUELE et al., 2011), o circuito utilizado no projeto foi construído a partir de componentes discretos.

- **Arduino Duemilanove**

Placa *Arduino* utilizada no projeto anterior e reutilizada no projeto atual. Faz o interfaceamento dos diversos sistemas do projeto, *i.e.*, recebe as decisões tomadas pelo Sistema de Navegação, embarcado na câmera, e aciona os motores para que o robô as execute. Seu funcionamento é detalhado na seção 2.2.

- **Microcontrolador ATMEGA328P**

Microcontrolador presente na *Arduino Duemilanove*, os códigos construídos para controle do robô (Seção 2.3) serão executados por ele.

#### 2.1.5 SENSORES

São apresentados com maiores detalhes na seção 3.

- **Câmera**

No projeto foi utilizado como principal sensor a *smart-cam CMUCam3* - ver seção 3.2.

- **Bússola**

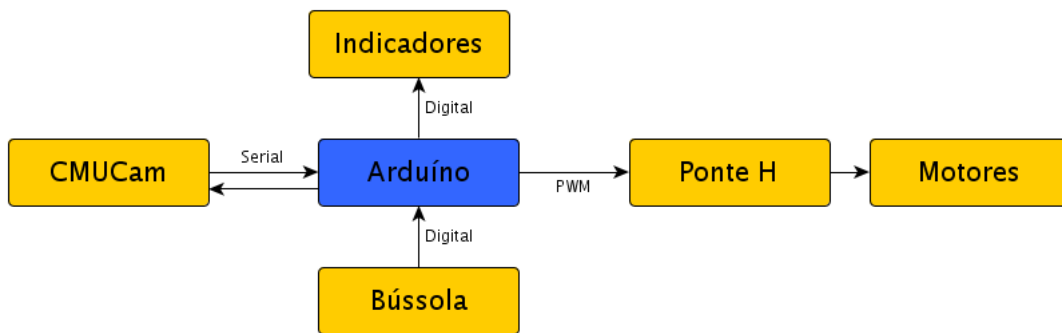
A bússola *Dinsmore* #1490 foi estudada para obter informação de orientação do robô - ver seção 3.1.

## 2.2 PLATAFORMA ARDUINO

*Arduino* é uma plataforma *open-source* para prototipagem eletrônica que busca facilitar a construção de sistemas onde haja interação com objetos e o ambiente. (Arduino Team, 2011a)

Para o projeto **Robô Explorador de Labirintos 2D** foi utilizada uma placa *Arduino Duemilanove* que foi reutilizada no trabalho atual.

Dentro do projeto, o *Arduino* exerce um papel central de receber, via comunicação serial, as decisões tomadas pelo Sistema de Exploração, embarcado na câmera, e atuar sobre os motores através das Pontes H. Além disso, ele recebe o sinal lido da bússola e aciona os dispositivos de indicação. Esse processo é representado na Figura 3.



**Figura 3: Processo de Comunicação - Arduino**

O *Arduino Duemilanove* possui um microprocessador *ATMEGA328P*. Sua programação pode ser feita utilizando a linguagem de programação própria, baseada em *Wiring* (Arduino Team, 2011a). No entanto, é possível utilizar *C++* para construir bibliotecas próprias utilizando também as bibliotecas já prontas do *Arduino*, e essa foi a opção feita pela equipe. O software de controle é detalhado na Seção 2.3.

Para o interfaceamento com os diversos módulos do sistema, o *Arduino Duemilanove* possui 14 pinos de comunicação digital (Arduino Team, 2011b). A configuração e utilização desses pinos está resumida na Tabela 1.

### 2.2.1 COMUNICAÇÃO COM INDICADORES

Os indicadores do robô são os LEDs, esquerdo e direito, e o *Speaker*.

Os estados possíveis dos LEDs são aceso ou apagado, como visto na Tabela 2. Para produzir os dois estados, é utilizada uma comunicação digital TTL, em que um sinal baixo (*LOW*) produzirá o estado apagado e o sinal alto (*HIGH*) irá deixar o LED aceso.



Pino	Sentido do Sinal	Tipo de Comunicação	Função
00	(RX)	Serial	Comunicação Câmera
01	(TX)	Serial	Comunicação Câmera
02	Entrada	Digital TTL	Bússola - Pino 1
03	Entrada	Digital TTL	Bússola - Pino 4
04	Saída	Digital TTL	LED Direito
05	Saída	Sinal PWM	Motor Esquerdo - Pino 2
06	Saída	Sinal PWM	Motor Esquerdo - Pino 1
07	Entrada	Digital TTL	Bússola - Pino 3
08	Entrada	Digital TTL	Bússola - Pino 2
09	Saída	Digital TTL	<i>Speaker</i>
10	Saída	Sinal PWM	Motor Direito - Pino 2
11	Saída	Sinal PWM	Motor Direito - Pino 1
12	Saída	Digital TTL	LED Esquerdo
13	-	-	<i>Não utilizado</i>

**Tabela 1: Pinos da placa *Arduino Duemilanove***

Estado	LED Esquerdo	LED Direito
Parado	Apagado	Apagado
Andando para frente	Aceso	Aceso
Virando para esquerda	Aceso	Apagado
Virando para direita	Apagado	Aceso

**Tabela 2: Sistema de Indicação**

O LED Direito é ligado no pino 04 e o Esquerdo no 12.

Para o *Speaker* é gerado um sinal com *duty-cycle* de 50% numa frequência especificada através da função *tone* presente na biblioteca básica do *Arduino*.

O *Speaker* é ligado ao *Arduino* através do pino 09.

## 2.2.2 COMUNICAÇÃO COM BÚSSOLA

A bússola utilizada pela equipe é a *Dinsmore Sensor Modelo #1490*, uma bússola digital com precisão de 45 graus, ou oito estados (Dinsmore Sensor Division, 1993).

A *Dinsmore #1490* gera quatro sinais digitais TTL, a combinação desses sinais fornece a orientação lida pela bússola.

Os quatro sinais são ligados nos pinos 02, 03, 07 e 08 do *Arduino*.

Para funcionamento detalhado da bússola, ver seção 3.1.

### 2.2.3 ACIONAMENTO DOS MOTORES

Para acionamento dos motores, são utilizadas Pontes H, circuitos eletrônicos que permitem o acionamento por parte de um microcontrolador de um motor de corrente contínua em qualquer sentido de rotação. Por não ser objeto de estudo deste trabalho, seu funcionamento e a construção do circuito utilizado no robô não serão explicitados, mais informações podem ser encontradas em (MENEGUELE et al., 2011) e (PATSKO, 2006).

A Ponte H, por sua vez, é acionada através de um sinal PWM - *Pulse Width Modulation*. A modulação por largura de pulso é uma técnica usada para obter resultados analógicos com um sinal digital (Arduino Team, 2011c). Uma onda quadrada é gerada, mudando o período em que essa onda está em nível ALTO (*duty cycle*) é possível obter níveis médios de tensão diferentes. Assim, por exemplo, uma onda sempre em nível BAIXO, produzirá 0 V de nível médio de tensão, uma onda sempre em ALTO, 5 V; e uma onda 50% do tempo em nível ALTO, 2,5 V. Esses diferentes níveis de tensão produzirão diferentes velocidades nos motores.

O *Arduino* possui portas que implementam o PWM, para utilizá-las é necessário usar a função *analogWrite* que recebe como parâmetro um inteiro entre 0 e 255. Esse valor irá definir o tempo em que o sinal gerado ficará em nível ALTO.

Neste projeto, não há necessidade de diferentes velocidades para o robô, por isso ela é mantida constante, foi definido experimentalmente o valor 191 para parâmetro da função *analogWrite*, que produzirá um *duty-cycle* de 75%, três quartos da velocidade máxima do robô. Nos movimentos de correção do alinhamento do robô, esse valor é alterado para 40 ou 60%, de acordo com o lado que precisa ser corrigido<sup>1</sup>.

Para cada Ponte H são gerados dois sinais que irão acionar os motores para frente ou para trás. Quando o sinal 1 é ALTO e o 2 é BAIXO, o motor gira num sentido, para frente, por exemplo; sinal 1 em BAIXO e sinal 2 em ALTO, gira no sentido inverso, para trás; esses sinais em função da ação executada pela robô são apresentados na tabela 3. Nos níveis médios de tensão gerados pelo PWM, o sentido do movimento se mantém, alterando apenas a velocidade do motor.

---

<sup>1</sup>A correção do movimento do robô consiste em fazer girar com maior velocidade o motor do lado para o qual ele precisa ir, isso é, se o movimento do robô foi irregular para a esquerda, a correção será na direção contrária, o motor direito irá girar com maior velocidade para alinhar novamente o robô.

<b>Ação</b>	<b>Motor Esquerdo</b>	<b>Motor Direito</b>
Parado	Desligado	Desligado
Para frente	Frente	Frente
Para trás	Trás	Trás
Virando esquerda	Frente	Desligado
Virando direita	Desligado	Frente
Rotacionando esquerda	Frente	Trás
Rotacionando direita	Trás	Frente

**Tabela 3: Acionamento dos Motores**

#### 2.2.4 COMUNICAÇÃO COM A CMUCAM

As decisões de movimentação são tomadas na aplicação que roda no microprocessador da câmera, mas quem aciona os motores e de fato produz o movimento é a aplicação executada no *Arduino*, por isso a necessidade de comunicação entre os dois dispositivos.

Esse interfaceamento entre o *Arduino* e a *CMUCam* é feito através de comunicação serial assíncrona *full-duplex*.

Por comunicação serial, entende-se aquela em que os *bits* são enviados em *fila* por um mesmo canal de comunicação. Por ser assíncrona, não há necessidade sincronismo de tempo entre os dois dispositivos envolvidos na comunicação, a própria mensagem irá definir o sinal de sincronismo. Como há necessidade de envio de mensagens da câmera para o *Arduino* e do *Arduino* para a câmera, a comunicação precisa ser *full-duplex*, isso é, trafegar nos dois sentidos.

Para cada troca de mensagem será enviado um byte entre os dispositivos.

As mensagens que trafegarão da *CMUCam* para o *Arduino* conterão comandos de ação para o robô, andar para frente ou para trás, rotacionar para a esquerda ou para a direita, parar o robô ou ainda corrigir o movimento do robô para determinado lado <sup>2</sup>. E a câmera pode ainda requisitar ao *Arduino* a leitura de informação da bússola, nesse momento o *Arduino* envia para ela a informação do sensor.

Foram definidas constantes que serão enviadas de um dispositivo para o outro, apresentadas na Tabela 4.

### 2.3 SOFTWARE DE CONTROLE

O Sistema de Controle executado no *Arduino* é responsável por receber e interpretar os comandos enviados pela *CMUCam3* e atuar sobre os motores do robô produzindo o movimento

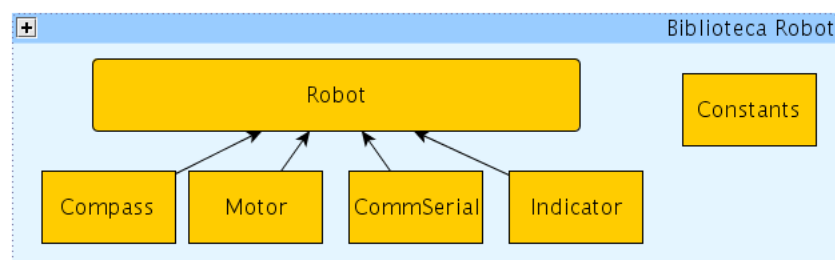
<sup>2</sup>Quando o movimento do robô for muito irregular e ele não conseguir seguir uma linha reta.

Valor	Constante	Significado
<i>Operações para o robô - Câmera-Arduino</i>		
1	FORWARD	Movimenta robô para frente
2	BACKWARD	Movimenta robô para trás
3	SPINLEFT	Rotaciona robô para esquerda
4	SPINRIGHT	Rotaciona robô para direita
5	STOP	Pára robô
6	COMPASS	Executa leitura da bússola
7	ADJUSTLEFT	Corrige movimento do robô para esquerda
8	ADJUSTRIGHT	Corrige movimento do robô para direita
<i>Leituras da bússola - Arduino-Câmera</i>		
1	NORTH	Norte
2	NORTHEAST	Nordeste
3	EAST	Leste
4	SOUTHEAST	Sudeste
5	SOUTH	Sul
6	SOUTHWEST	Sudoeste
7	WEST	Oeste
8	NORTHWEST	Noroeste

**Tabela 4: Comunicação Arduino-Câmera**

necessário. Para executar essa tarefa, a equipe desenvolveu uma biblioteca orientada a objetos em C++, cujo diagrama básico é apresentado na figura 4.

A ideia de construir a classe se baseia em representar em código a maneira como a equipe enxerga a função do *Arduino* e tentar encapsular as funções em classes que devem então ser apenas instanciadas, facilitando a programação, já que operações por vezes complexas são convertidas em operações elementares.



**Figura 4: Diagrama - Biblioteca Robot**

### 2.3.1 COMPASS

Classe responsável por fazer a leitura da bússola e decodificar o sinal lido, tendo como resposta apenas a constante que representa a direção cardinal lida.

### 2.3.2 MOTOR

Classe responsável por acionar os motores, recebe da função *main* executada no *Arduino* como parâmetro qual movimento deve ser feito e atua sobre os motores produzindo-o.

### 2.3.3 COMMSERIAL

Classe para a comunicação com a *CMUCam3*, possui um método de escrita e um de leitura apenas.

### 2.3.4 INDICATOR

Classe para produzir a representação necessária nos indicadores do robô.

### 2.3.5 ROBOT

Uma vez que todas as ações do robô estão definidas nas classes anteriormente citadas, a classe *Robot* possui objetos das mesmas e sua tarefa é chamar as funções nos momentos adequados.

No programa principal do *Arduino* há um objeto dessa classe que possui um método principal que monitora a comunicação serial esperando os comandos da *CMUCam* através da classe *CommSerial*, quando ele recebe um comando, dispara as funções das classes imediatamente abaixo.

### 2.3.6 CONSTANTS

Armazena as constantes definidas ao longo da biblioteca e não possui nenhum método. Armazena tanto as constantes de comunicação, Tabela 4, quanto as constantes para acionamento dos motores, por exemplo.

### 3 SENSORES

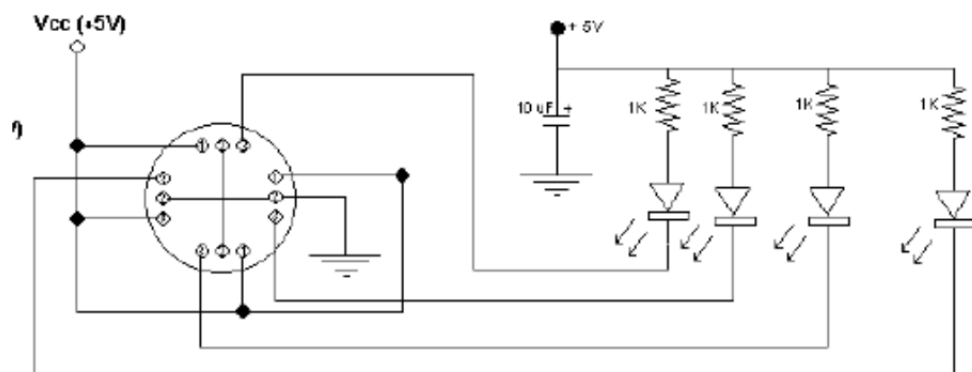
A equipe utiliza dois sensores no projeto: um eletromagnético digital, Bússola; e um ótico, Câmera. O primeiro fornece informação de orientação do robô e o segundo informações visuais do ambiente.

#### 3.1 BÚSSOLA

A bússola utilizada no projeto é a *Dinsmore Sensor Modelo # 1490* (Dinsmore Sensor Division, 1993), emprestada à equipe pelo professor Hugo Vieira.

Foi necessário construir um *shield*, placa de suporte para a bússola conectada ao *Arduino*, que possibilitasse a conexão da Bússola com o restante do sistema. O circuito utilizado é apresentado na figura 5 e foi baseado no trabalho do *Autonomous Vehicle Team* do *College of New Jersey* (BUECHEL et al., 2005).

Internamente a bússola funciona como um transistor coletor aberto NPN e fornece sinais 0 ou 5V, TTL, (Dinsmore Sensor Division, 1993), captados nos cátodos dos diodos, para as entradas digitais do *Arduino*, sendo alimentada com 5V através do *Arduino* também.



**Figura 5: Circuito Eletrônico - Bússola**

A combinação dos quatro sinais fornecidos pela bússola fornece sua orientação com

uma precisão de 45 graus, *i.e.*, há oito estados possíveis para a bússola.

A combinação entre esses estados e sua representação são apresentados na Tabela 5. As combinações de *bits* que não representam um estado válido não são apresentadas.

Sinal 1	Sinal 2	Sinal 3	Sinal 4	Direção	
0	0	0	1	W	Oeste
0	0	1	0	S	Sul
0	0	1	1	SW	Sudoeste
0	1	0	0	E	Leste
0	1	1	0	SE	Sudeste
1	0	0	0	N	Norte
1	0	0	1	NW	Noroeste
1	1	0	0	NE	Nordeste

**Tabela 5: Acionamento dos Motores**

### 3.2 CÂMERA

Para o projeto foi adquirida uma *CMUcam3*, desenvolvida pela *Carmegie Mellon University*, que se propõe a criar um sistema de visão simples em sistemas embarcados através de um sensor inteligente (Carmegie Mellon University, 2011).

A *CMUcam3* utiliza um sistema baseado na arquitetura *ARM7TDMI* e tem como principal microprocessador um *Philips LPC2106* conectado a uma câmera CMOS (Carmegie Mellon University, 2006).

A justificativa da escolha da *CMUcam3* se baseia na existência de diversos exemplos de códigos e bibliotecas prontas para o processamento de imagens para essa plataforma, como obtenção de mapa de cores, histograma e detecção de bordas.

Por se tratar de um microprocessador com maior capacidade de processamento, o sistema de controle e tomada de decisões foi desenvolvido no *LPC2106*.

A alimentação da *CMUcam3* é feita por quatro pilhas AA em série, totalizando 6V.

## 4 VISÃO

O sistema de visão, desenvolvido na *CMUCam3*, é responsável principalmente por identificar e localizar o objeto a ser procurado no seu campo de visão, identificar objetos de interesse para serem utilizados na navegação, e identificar obstáculos. Ele também tem a função auxiliar de corrigir desvios na movimentação do robô. De forma a resolver esses problemas, diversas funcionalidades apresentadas pela *CMUcam3* são utilizadas. Exemplos de projetos já existentes, como o *simple-track-color* em que uma determinada cor é identificada na imagem, ou o *serial-test* que utiliza a comunicação serial da câmera, também são aproveitados como referência.

### 4.1 IDENTIFICAÇÃO DE OBJETOS DE INTERESSE

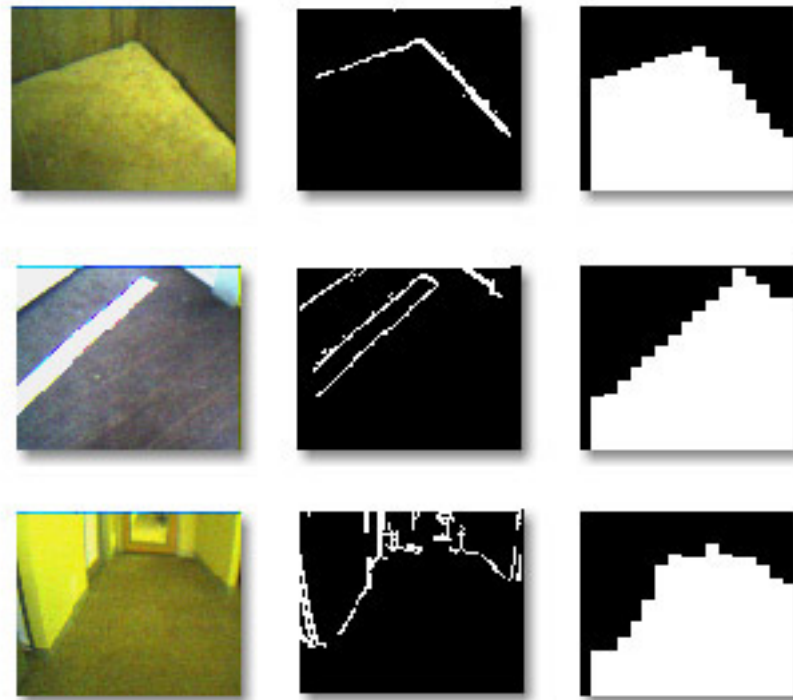
Uma das principais funções do sistema de visão é localizar, caso se encontre dentro de seu campo de visão, o objeto a ser procurado pelo robô. Como este é simples e possui uma única cor, de destaque, que não existe no ambiente proposto (e é improvável de se encontrar em um ambiente genérico) o mapa de cores obtido com a *CMUcam* pode ser utilizado para localizar o objeto na imagem, aliado a uma ferramenta de reconhecimento de forma.

O sistema também deve localizar, dentro de seu campo de visão, objetos que se destaquem, para serem utilizados pelo sistema de navegação, de forma a identiicar a posição atual. Para a identificação desses objetos, ou pontos de interesse, as características de cor e forma também podem ser utilizadas.

### 4.2 IDENTIFICAÇÃO DE OBSTÁCULOS

Obstáculos podem ser identificados, na imagem, através do histograma da imagem obtida, ou então por detecção de bordas. A primeira abordagem é preferível por auxiliar a identificar as áreas (chão) por onde o robô pode se locomover, ao invés dos limites entre o chão e os obstáculos. A Figura 6 apresenta algumas imagens exemplo obtidas no site da *CMUcam3*.





**Figura 6: Histograma e Detecção de Bordas de algumas imagens exemplo**

#### 4.3 CORREÇÃO DE DESVIOS

De forma a corrigir eventuais desvios na movimentação do robô, causados por uma diferença de rotação entre as duas rodas, o sistema também deve identificar, na imagem, através da detecção de bordas e do mapa de cores, alguns objetos ou bordas de destaque, e caso o movimento seja muito irregular, corrigir sua rota.

## 5 NAVEGAÇÃO

Para o robô encontrar o objeto anteriormente especificado ele deve saber se locomover dentro de um ambiente desconhecido. Para navegar no ambiente, o robô deve criar uma espécie de mapa que contenha informações que possibilitem sua locomoção no ambiente. Duas abordagens distintas são a representação de maneira global ou de maneira local (CONRADT, 2008).

De acordo com (CONRADT, 2008), a representação global procura descrever o ambiente da maneira mais fiel à realidade possível, indicando o mapa a partir de seus nodos e suas conexões, retendo o máximo de informação possível para conseguir manter uma consistência espacial global conforme o mapa vai sendo construído. A grande desvantagem da representação global é a necessidade de se reter e processar toda essa informação para criar-se o mapa, portanto conforme a quantidade de nodos aumenta, o mapa em si aumenta e a complexidade computacional também aumenta consideravelmente.

Para o projeto atual, a equipe optou por utilizar a representação local, pois a abordagem global se torna ineficiente devido a sua complexidade computacional envolvida para manter a consistência do mapa. A abordagem local possibilita o escalonamento do sistema para permitir a ele operar em ambientes maiores.

### 5.1 CONSTRUÇÃO DO MAPA

Para a criação do mapa é utilizada principalmente a câmera *CMUcam3* acoplada ao robô. A câmera é a principal responsável por adquirir informações do ambiente para a criação do mapa. Junto com a câmera, a bússola é utilizada para fornecer as direções entre um nodo e outro. Com as informações obtidas da câmera e da bússola, o robô é capaz de construir um mapa do ambiente e deve ser capaz de locomover-se dentro do ambiente, procurando o objeto desejado e desviando de obstáculos caso necessário.

Inicialmente, ao ser inicializado num novo ambiente, o robô deve criar um nodo A

que será demarcado como o primeiro nodo existente no mapa. A primeira informação a ser obtida será procurar características do ambiente ou objetos próximos ao nodo para caracterizar o nodo em questão. Após obter informações do ambiente atual, o robô deve observar seus arredores e definir outros nodos alcançáveis a partir do nodo atual. O robô então deve guardar informações pertinentes sobre como chegar a cada um dos nodos vizinhos, como direção e a existência de objetos de destaque. Tais informações sobre o caminho serão guardadas em ambos os nodos envolvidos (nodo A e nodo B, início e destino, respectivamente). Dentre todos os nodos vizinhos, o sistema escolhe um e se locomove até ele. O processo então é repetido diversas vezes até que o mapa seja completamente montado (não existe nenhum nodo alcançável não visitado) ou o objeto seja encontrado.

## 5.2 ROTEAMENTO

Na representação local os nodos (pontos de referência, ou *Place Agents*) possuem informações apenas do próprio local, de quais são os nodos vizinhos, possíveis pontos de interesse (objetos de destaque para identificação do nodo, por exemplo) e informações pertinentes sobre como alcançar o próximo nodo (direção, existência de obstáculos, etc.) seja este já presente no mapa ou não. Na representação local, para se alcançar um nodo já mapeado e que não seja vizinho ao atual será necessário um algoritmo de roteamento no sistema, onde cada *Place Agent* pergunta por essa informação aos seus vizinhos e assim sucessivamente até que o nodo requisitado seja encontrado.

Quando o nodo requisitado é encontrado o robô traça um caminho para chegar ao local desejado, passando por nodos já existentes.

## 6 CONCLUSÃO

No atual momento do projeto, a comunicação entre *CMUcam3* e *Arduino* ainda não foi implementada, por esse motivo ainda não foi possível fazer os testes de exploração e identificação do objeto.

Analisando a estrutura do robô desde a parte mecânica, eletrônica e as estratégias para exploração e localização do objeto encontrado, espera-se que o robô consiga encontrar um objeto específico e de destaque dentro de um ambiente pequeno, com as dimensões 168,2 cm por 118,9 cm.

Espera-se também conseguir implementar corretamente um código de exploração inteligente baseado em representações locais para locomoção do robô dentro do espaço determinado. Além disso, conseguir aplicar um código para reconhecer o objeto a ser encontrado utilizando características de destaque (cores fortes, forma, etc.) do objeto. Tais características do robô (locomoção e identificação) serão controladas pela câmera acoplada ao robô, a *CMUcam3*.

Como continuação deste projeto, seria interessante ver o robô se locomovendo em espaços maiores e mais dinâmicos e também que o robô possa identificar objetos mais familiares ao dia a dia como algum equipamento ou molho de chaves.

## REFERÊNCIAS

Arduino Team. **Arduino**. Outubro 2011. <http://arduino.cc>.

Arduino Team. **Arduino Duemilanove**. Outubro 2011.  
<http://arduino.cc/en/Main/arduinoBoardDuemilanove>.

Arduino Team. **PWM**. Outubro 2011. <http://www.arduino.cc/en/Tutorial/PWM>.

BUECHEL, M.; NISH, T.; VACIRCA, N. New jersey autonomous vehicle. **New Jersey: The College of New Jersey**, 2005.

Carmegie Mellon University. **CMUcam3 Datasheet**. 2006.

Carmegie Mellon University. **CMUcam3: Open Source Programmable Embedded Color Vision Plataform**. 2011. <http://cmucam.org>.

CONRADT, J. A distributed cognitive map for spatial navigation based on graphically organized place agents. **Institute of Neuroinformatics, UZH/ETH-Zürich**, 2008.

Dinsmore Sensor Division. **Dinsmore Sensing Systems Geral Information**. 1993.

MENEGUELE, B.; PADILHA, F.; ARCANJO, V. Robô explorador de labirintos 2d. **Universidade Tecnológica Federal do Paraná**, 2011.

PATSKO, L. F. **Tutorial de Montagem da Ponte H**. Dezembro 2006.

## **APÊNDICE A – CADERNO DE BORDO**

### **10 de Agosto**

Primeira aula da disciplina.

### **17 de Agosto**

Marcelo Teider e Matheus Araujo decidem formar a equipe, têm em mente o projeto de um robô explorador. A equipe ainda não tem os três integrantes, como sugerido para a Disciplina. Então iniciam-se negociações com outras equipes para a definição dos integrantes.

### **12 de Agosto**

Após algumas conversas, as equipes da disciplina são definidas. Luis Camargo é integrado à equipe.

### **13 de Agosto**

Após a definição da equipe, a professora Myriam Delgado é convidada para nos orientar, aceitando a proposta.

### **13 de Agosto**

O projeto é então definido como a construção de um robô explorador. A intenção é “mostrar” ao robô um objeto para reconhecimento, então esse mesmo objeto é escondido de seu campo de visão e o robô deve então explorar o ambiente procurando-o. Ele deve também evitar obstáculos durante o percurso.

### **24 de Agosto**

Apresentação da pré-proposta. A equipe apresentou aos professores da disciplina a pré-proposta de projeto, sendo aceita pelos professores. Como sugerido pelo professor Hugo durante a apresentação, decidimos comprar uma *SmartCam*. A primeira ideia seria utilizar a *CMUcam*, mas optamos por pesquisar outros modelos.

### **29/30 de Agosto**

Iniciamos as pesquisas das câmeras. Pesquisamos os modelos *AVRcam*, que não está sendo produzido no momento; e alguns modelos comerciais, que foram descartados devido ao elevado custo, acima de US\$ 2000,00. Dentre os modelos *CMUcam*, ficamos em dúvida entre dois modelos, *CMUcam1* e *CMUcam2*.

Fizemos um levantamento do material que será necessário para a construção do robô, como motor, chassi, bateria e microprocessador.

### **31 de Agosto**

Em conversa com o professor Hugo, optamos pela *CMUcam1*; uma vez que ela atende os requisitos do projeto e tem menor custo. Os procedimentos para adquiri-la foram tomados.

Conseguimos com a equipe que desenvolveu o Robô Explorador de Labirintos 2D nessa mesma disciplina, em 2011-01, o robô emprestado. Segundo a Equipe, uma das Pontes H do robô apresenta problemas, precisaremos solucioná-lo.

Precisamos agora decidir e conseguir o microcontrolador para o Robô. Os da plataforma *Arduino* apresentam possibilidade de comunicação com a câmera via software e pela familiaridade dos membros da equipe podem se tornar uma boa escolha.

### **5 de Setembro**

Pegamos o robô com a equipe do semestre passado. Sem problemas com a ponte H, o robô não apresenta nenhum problema mecânico ou elétrico-eletrônico. Estamos utilizando o mesmo microcontrolador da equipe, um *ATMEGA 328P* em uma placa *Arduino Duemilanove*.

### **7 de Setembro**

Desenvolvemos uma API para controle do hardware do Robô. Fizemos uma classe *Robot* em C++ com funções pré-definidas como *startrobot*, *forward*, *turnleft*. No código a ser desenvolvido não necessitaremos controlar o robô diretamente, apenas através dessa classe.

### **12 de Setembro**

Por indicação de nossa orientadora, conhecemos o trabalho *A Distributed Cognitive Map for Spatial Navigation Based on Graphically Organized Place Agents*, de Jörg Conradt e Rodney Douglas, e por orientação do professor Luiz Merkle com um algoritmo de busca recursiva em um espaço bidimensional pelas quatro direções cardeais através de pilha, no livro *Data Structures - An advanced approach using C*.

### **21 de Setembro**

Definida a data da banca final, 7 de Dezembro.

Por orientação do professor Hugo, entramos em contato com diversos artigos de Prestes, Lowe, Bay e Artolazabal.

### **23 a 28 de Setembro**

Escrita do relatório de Qualificação, entregue aos professores no dia 28. Percebemos a necessidade de usar uma bússola como sensor para definir a direção em que se encontra o robô.

### **28 de Setembro**

Entrega da Qualificação aos professores da Disciplina. Conseguimos emprestado com o professor Hugo duas bússolas *Dinsmore*, uma digital e outra analógica. Durante a próxima semana faremos os testes para definir qual iremos usar. A bússola será integrada ao sistema através da placa do *Arduino*.

### **30 de Setembro**

Foram encontrados problemas com o interfaceamento entre a bússola e o Arduino devido a diferentes níveis de tensão e corrente, será necessário construir um conversor DC-DC para fazer a integração.

### **6 de Outubro**

Após orientação do professor Hugo e pesquisa de outros projetos, fizemos o primeiro experimento com a bússola e obtivemos 5 [V] na saída, o que possibilita interfaceamento direto com o Arduino.

### **12 de Outubro**

Construímos um *shield* para a bússola, possibilitando conectá-la ao Arduino.

### **14 de Outubro**

A câmera foi finalmente enviada de Hong Kong. A previsão de entrega é de um mês, precisamos elaborar um plano B caso a câmera não chegue a tempo da finalização do projeto.

### **15 de Outubro**

Conectamos o *shield* da bússola ao Arduino; tivemos alguns problemas de mau contato com os cabos, mas o robô é capaz de interpretar as informações fornecidas pela bússola e segue os comandos que lhe foram dados.

### **18 de Outubro**

Tivemos uma reunião com nossa orientadora, Prof. Myriam, onde apresentamos o estado atual do projeto e elaboramos um segundo plano, caso a câmera não chegue. Iremos



utilizar os sensores do projeto passado, cinco pares de leds emissor/receptor, e construir uma pasta com marcas de diferentes cores perceptíveis pelo robô, onde ele deverá construir um mapa cognitivo.

### **19 de Outubro**

A câmera chegou! Iremos começar os testes com ela em breve. Precisaremos de um cabo serial ou um adaptador para comunicação com a placa e também de baterias para alimentação da mesa, 6V.

### **26 de Outubro**

Compramos um adaptador para o cabo de comunicação e um suporte para quatro pilhas AA que servirá como alimentação.

Conseguimos compilar e gravar os códigos de exemplo na câmera.

### **2 de Novembro**

Estamos estudando a comunicação da *CMUCam* com o *Arduino*.

### **9 de Novembro**

Concentramos nossas atenções na finalização da primeira versão da monografia.

### **16 de Novembro**

Entrega da primeira versão da monografia, os trabalhos com o robô serão retomados a partir de agora.

### **23 de Novembro**

Estamos testando o algoritmo de detecção de objetos, usamos como exemplo o projeto *simple-track-color* da biblioteca de projetos da *CMUCam*, ele é capaz de identificar dentro da imagem a presença de uma faixa de cores definida como parâmetro. Seu retorno é o retângulo e o centróide na imagem onde a faixa de cores definida foi encontrada.

Os testes com a *VirtualCam* estão funcionando, no entanto com a câmera na real ainda não.

### **25 de Novembro**

O código do *simple-track-color* parecia conter alguns erros, no fórum de discussão da *CMUCam* conseguimos uma correção e ele passou a funcionar também com a câmera real. Estamos levantando os mapas de cores da bola laranja.

**28 de Novembro**

Estamos fazendo os testes com a comunicação entre *Arduino* e *CMUCam*, estamos enviando comandos da câmera para o *Arduino*, mas ainda não conseguimos um método eficiente de identificar o que ele está recebendo.

**30 de Novembro**

A comunicação está funcionando! Estamos enviando caracteres 'a', 'b', 'c', 'd' e 'e' da câmera para o *Arduino*, mas não conseguimos identificar uma relação entre o que foi enviado e o que está recebido *Arduino*, 79, 39, 78, 19, 77, respectivamente.

**2 de Dezembro**

Com a identificação do objeto funcionando e a comunicação também, iremos começar os testes reais com o robô. Como suporte para a câmera no robô foi utilizado um pedaço de madeira.

**5 de Dezembro**

O robô funciona!!! Não é o que havíamos planejado no começo do projeto, mas conseguimos fazer com que ele identifique a bola laranja e se movimente até ela.

Começamos a preparar a apresentação.

**7 de Dezembro**

Apresentação da banca final.

## APÊNDICE B – DESCRITIVO

Esse apêndice tem como intuito descrever de maneira sucinta o que foi realizado durante o projeto, e pode ser utilizado em trabalhos futuros como referência rápida.

O objetivo inicial da equipe era construir um robô que pudesse se localizar em um ambiente controlado e conseguisse encontrar um objeto especificado; durante o projeto, a bola laranja foi definida como o objeto a ser pesquisado.

Houve algumas dificuldades na aquisição da câmera, inicialmente, a equipe optou pela *CMUCam1*, por atender os requisitos do projeto e ser mais barata. No entanto, nenhum fornecedor da câmera possuía o modelo mais simples da *CMUCam*, o mesmo aconteceu com a *CMUCam2*, sendo necessário então adquirir a *CMUCam3* junto a um fornecedor em Hong Kong.

A entrega da câmera atrasou devido a dos funcionários dos Correios no Brasil.

Enquanto a câmera não chegava, a equipe trabalhou com o robô que já havia sido construído, e implementou a biblioteca de controle do mesmo. Nesse período foram feitos os testes com a bússola também. A equipe conseguiu integrá-la ao *Arduino* e chegou a fazer alguns testes de movimento do robô a partir de informações que eram obtidas através dela.

Com a câmera em mãos, foram feitos os testes dos algoritmos de reconhecimento da *CMUCam3*. O programa exemplo usado para identificação foi o *simple-track-color*. Este programa é capaz de, a partir de uma faixa RGB, identificar dentro da imagem onde está a cor e retorna como parâmetro dois pontos de um retângulo onde a cor está contida, o centróide da cor, i.e., o ponto onde há maior concentração daquela cor, e a densidade da cor dentro daquele retângulo.

Com um algoritmo rudimentar de identificação funcionando, a equipe se concentrou em desenvolver o sistema de comunicação *Arduino-CMUCam*. Fisicamente, foi utilizada a porta serial da *CMUCam*, a mesma que ela utiliza para comunicar com o computador, e os pinos 0 e 1 do *Arduino*. Nos testes feitos pela equipe, aparentemente não havia uma relação direta entre o que era enviado pela câmera e o que era recebido pelo *Arduino*; no entanto, havia

a integridade da informação. Ou seja, o que era enviado pela câmera era sempre recebido da mesma forma pelo *Arduino*, então a equipe levantou uma tabela de relação e implementou o sistema de comunicação.

Quando os dois módulos descritos anteriormente já estavam funcionando, o prazo de entrega do projeto já estava muito próximo e a equipe então reduziu o escopo do projeto e o objetivo passou a ser identificar o objeto e fazer o robô se movimentar até ele. Para isso, foi elaborado um algoritmo simples de navegação.

Esse algoritmo movimenta o robô em direção à bola sem no entanto evitar obstáculos. O robô para quando a bola está suficientemente próxima. Os parâmetros de decisão de parada, virar para a esquerda ou direita, movimentar para a frente foram obtidos de maneira experimental. i Todo o código desenvolvido pela equipe está disponível no repositório:

*<http://github.com/matheusaraujo/oficina2-robocam>*

Como projetos futuros, a equipe sugere implementar um algoritmo inteligente de navegação e aprimorar a percepção, utilizando outras informações além da cor do objeto, como detecção de bordas e histograma também. Com relação ao projeto mecânico do robô, acredita-se que foi atingido um nível satisfatório, precisando apenas alguns ajustes pequenos principalmente com relação aos diversos fios do robô que eventualmente podem atrapalhar os movimentos. Outro ponto a ser explorado é a integração das informações da bússola ao algoritmo de exploração.