

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.

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

RESUMO

CAMARGO, Luis Guilherme M. ; LOPES, Marcelo Teider; ARAÚJO, Matheus Silva. ROBÔ EXPLORADOR DE AMBIENTES. 29 f. Monografia – Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2011.

Texto do resumo (máximo de 500 palavras).

Palavras-chave: Palavra-chave 1, Palavra-chave 2, ...

ABSTRACT

CAMARGO, Luis Guilherme M. ; LOPES, Marcelo Teider; ARAÚJO, Matheus Silva. AM-BIANCE EXPLORER ROBOT. 29 f. Monografia – Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2011.

Abstract text (maximum of 500 words).

Keywords: Keyword 1, Keyword 2, ...

LISTA DE FIGURAS

FIGURA 1	– Diagrama do Robô	10
FIGURA 2	– Processo de Comunicação - Arduino	13
FIGURA 3	– Diagrama - Biblioteca <i>Robot</i>	17
FIGURA 4	– Circuito Eletrônico - Bússola	19

LISTA DE TABELAS

TABELA 1	– Pinos da placa <i>Arduíno Duemilanove</i>	14
TABELA 2	– Sistema de Indicação	14
TABELA 3	– Acionamento dos Motores	16
TABELA 4	– Comunicação Arduíno-Câmera	17
TABELA 5	– Acionamento dos Motores	20

SUMÁRIO

1	INTRODUÇÃO	9
1.1	MOTIVAÇÃO	9
1.2	OBJETIVO	9
1.2.1	Objetivo Geral	9
1.2.2	Objetivos Específicos	9
1.3	VISÃO GERAL DO PROJETO	9
2	SISTEMA MECÂNICO	10
2.1	PROJETO MECÂNICO	10
2.1.1	Sistema de Alimentação	11
2.1.1.1	Bateria de 9V	11
2.1.1.2	Bateria de 4,2V	11
2.1.1.3	Bateria de 6V	11
2.1.2	Sistema de Indicação	11
2.1.2.1	LEDs	11
2.1.2.2	Speaker	11
2.1.3	Sistema de Tração	11
2.1.3.1	Motores	11
2.1.3.2	Caixas de redução	12
2.1.3.3	Rodas	12
2.1.4	Sistema de Controle	12
2.1.4.1	Ponte H	12
2.1.4.2	Arduíno Duemilanove	12
2.1.4.3	Microcontrolador ATMEGA328P	12
2.1.5	Sensores	12
2.1.5.1	Câmera	13
2.1.5.2	Bússola	13
2.2	PLATAFORMA ARDUÍNO	13
2.2.1	Comunicação com Indicadores	14
2.2.2	Comunicação com Bússola	15
2.2.3	Acionamento dos Motores	15
2.2.4	Comunicação com CMUCam	16
2.3	SOFTWARE DE CONTROLE	17
2.3.1	Compass	18
2.3.2	Motor	18
2.3.3	CommSerial	18
2.3.4	Indicator	18
2.3.5	Robot	18
2.3.6	Constants	18
3	SENSORES	19
3.1	BÚSSOLA	19
3.2	CÂMERA	20

4	VISÃO	21
4.1	RECONHECIMENTO DE IMAGENS	21
5	NAVEGAÇÃO	22
5.1	CONSTRUÇÃO DO MAPA	22
5.2	ROTEAMENTO	23
6	CONCLUSÃO	24
	REFERÊNCIAS	25
	Apêndice A – CADERNO DE BORDO	26

1 INTRODUÇÃO

1.1 MOTIVAÇÃO

1.2 OBJETIVO

O objetivo do projeto é a construção de um robô explorador, utilizando uma câmera e uma bússola como sensores. Ele deve 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

1.3 VISÃO GERAL DO PROJETO

2 SISTEMA MECÂNICO

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 *Arduíno 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 1.

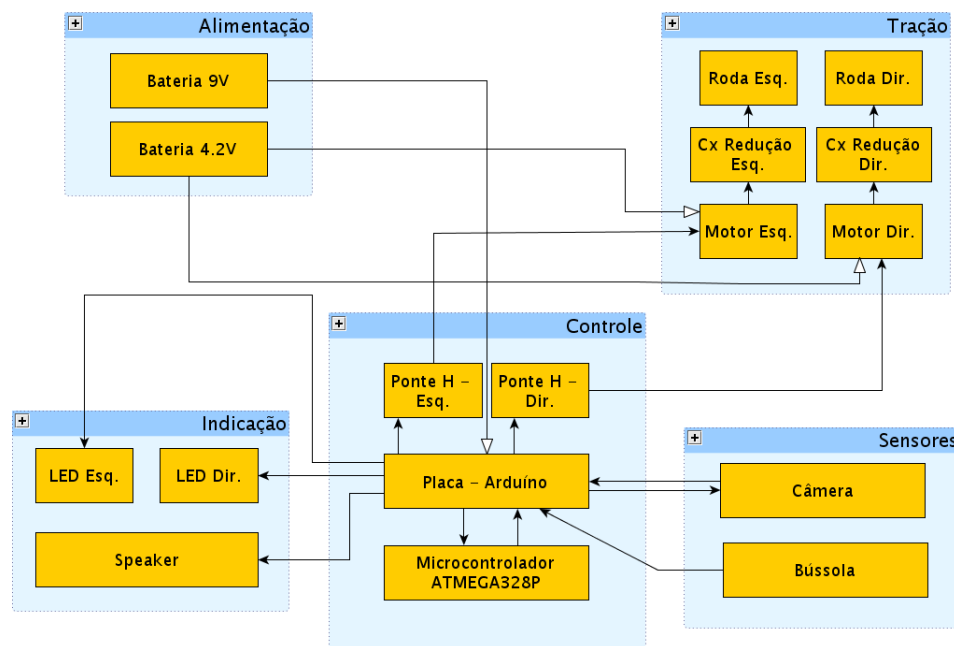


Figura 1: Diagrama do Robô

2.1.1 SISTEMA DE ALIMENTAÇÃO

Conjunto de baterias utilizadas como alimentação do robô.

2.1.1.1 BATERIA DE 9V

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

2.1.1.2 BATERIA DE 4,2V

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

2.1.1.3 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.

2.1.2.1 LEDS

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

2.1.2.2 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 um motor elétrico e duas rodas centrais.

2.1.3.1 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)

2.1.3.2 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).

2.1.3.3 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.

2.1.4.1 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.

2.1.4.2 ARDUÍNO DUEMILANOVE

Placa *Arduíno* 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.

2.1.4.3 MICROCONTROLADOR ATMEGA328P

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

2.1.5 SENSORES

São apresentados na seção 3.

2.1.5.1 CÂMERA

Ver seção 3.2.

2.1.5.2 BÚSSOLA

Ver seção 3.1.

2.2 PLATAFORMA ARDUÍNO

Arduíno 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 utilizado uma placa *Arduíno Duemilanove* que foi reutilizada no trabalho atual.

Dentro do projeto, o *Arduíno* 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 da Ponte H. Além disso, ele recebe o sinal lido através da bússola e aciona os dispositivos de indicação. Esse processo é representado na Figura 2.

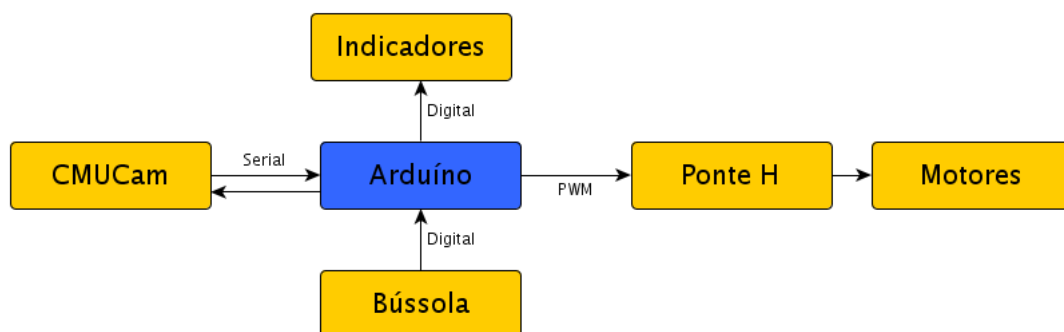


Figura 2: Processo de Comunicação - Arduíno

O *Arduíno 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 *Arduíno*, 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 *Arduíno Duemilanove*

possui 14 pinos de comunicação digital (Arduino Team, 2011b), a configuração e utilização desses pinos está resumida na Tabela 1.

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 *Arduíno Duemilanove*

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.

Estado	LED Esquerdo	LED Direito
Parado	Apagado	Apagado
Andando para frente	Aceso	Apagado
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 *Arduíno*.

O *Speaker* é ligado ao *Arduíno* 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 *Arduíno*.

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 será explicitado, maiores 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 ALTO (*duty cycle*) é possível obter níveis médios de tensão diferentes. Assim, por exemplo, uma onda sempre em BAIXO, produzirá 0 V de nível médio de tensão, uma onda sempre em ALTO, 5 V; e uma onda 50% do tempo em ALTO, 2,5 V. Esses diferentes níveis de tensão produzirão diferentes velocidades nos motores.

O *Arduíno* 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 ALTO.

Neste projeto, na maior parte dos movimentos, a velocidade do robô é constante, então foi definido experimentalmente o valor 127 para parâmetro da função *analogWrite*, que produzirá um *duty-cycle* de 50%, e metade da velocidade do robô. Nos movimentos de correção do alinhamento do robô, esse valor é alterado para 40 ou 60%.

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, ele gira no sentido invertido, para trás; esses

sinais em função da ação executada pela robô são apresentados na tabela 3.

Ação	Motor Esquerdo	Motor Direito
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 CMUCAM

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

Esse interfaceamento entre o *Arduíno* 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 *Arduíno* e do *Arduíno* 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 *Arduíno* 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 ¹. E a câmera pode ainda requisitar ao *Arduíno* a leitura de informação da bússola, nesse momento o *Arduíno* envia para ela a informação do sensor.

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

¹Quando o movimento do robô for por demais 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

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 necessário. Para executar essa tarefa, a equipe desenvolveu uma biblioteca orientada a objetos em C++, cujo diagrama básico é apresentado na figura 3.

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.

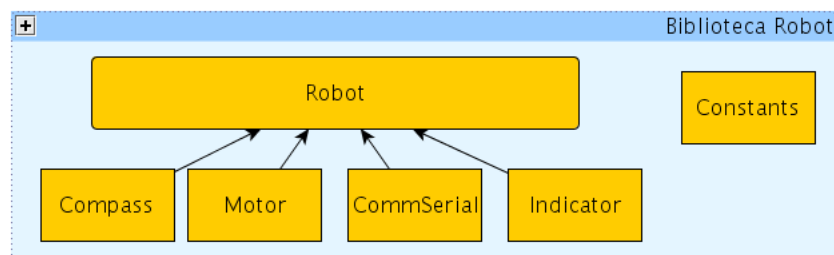


Figura 3: Diagrama - Biblioteca Robot

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 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 *Arduíno* há um objeto dessa classe que possui um método principal que funciona como um *watchdog* para 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* para conectá-la ao *Arduíno*. O circuito utilizado é apresentado na figura 4 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 catodos dos diodos, para as entradas digitais do *Arduíno*, sendo alimentada com 5V através do *Arduíno* também.

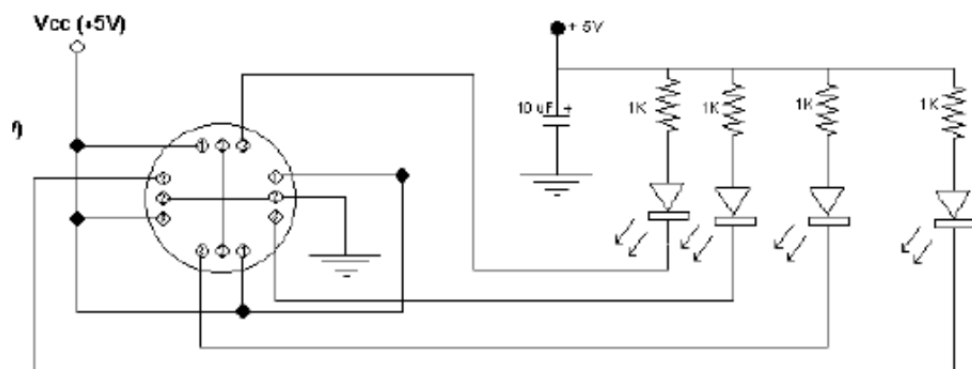


Figura 4: Circuito Eletrônico - Bússola

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.

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

4.1 RECONHECIMENTO DE IMAGENS

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 [5].

De acordo com Jorg [5], 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.

Na representação local os nodos (pontos de referência, denominados 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 nodo já presente no mapa ou não.

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

5.1 CONSTRUÇÃO DO MAPA

Para a criação do mapa utilizamos 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 deve ser 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, denominados 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

Analisando a estrutura do robô desde a parte mecânica, elétrica e as estratégias para exploração e localização do objeto encontrado, esperamos 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.

Esperamos conseguir implementar corretamente um código de exploração inteligente baseado em representações locais para locomoção do robô dentro do espaço determinado. Além disso, esperamos 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, gostaríamos de ver o robô se locomovendo em espaços maiores e mais dinâmicos e também gostaríamos que o robô possa identificar objetos mais familiares ao dia a dia como algum equipamento ou molho de chaves. Se possível, também gostaríamos de ver o robô interagir com o ambiente a sua volta, como tentar buscar e pegar algum objeto ou simplesmente locomovê-lo.

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>.

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 *Arduíno* 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 *Arduíno 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 *Arduíno*.

30 de Setembro

Foram encontrados problemas com o interfaceamento entre a bússola e o arduíno 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 arduíno.

12 de Outubro

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

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 Arduíno; 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 *Arduíno*.

9 de Novembro

Concentramos nossas atenções na finalização da primeira versão da monografia.