

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA/INFORMÁTICA
CURSO DE ENGENHARIA DA COMPUTAÇÃO

LUCAS LONGEN GIOPPO
MARCELO MASSAO KATAOKA HIGASKINO
RICARDO FANTIN DA COSTA
WILLIAM HITOSHI TSUNODA MEIRA

ROBÔ SEGUIDOR DE LINHA

MONOGRAFIA

CURITIBA

2009

LUCAS LONGEN GIOPPO
MARCELO MASSAO KATAOKA HIGASKINO
RICARDO FANTIN DA COSTA
WILLIAM HITOSHI TSUNODA MEIRA

ROBÔ SEGUIDOR DE LINHA

Monografia apresentado à Unidade Curricular de
Oficina de Integração II do Curso de Engenharia da
Computação da Universidade Tecnológica Federal
do Paraná como requisito parcial para aprovação.

Orientador: Prof. Dr. João Alberto Fabro

CURITIBA

2009

TERMO DE APROVAÇÃO

LUCAS LONGEN GIOPPO
MARCELO MASSAO KATAOKA HIGASKINO
RICARDO FANTIN DA COSTA
WILLIAM HITOSHI TSUNODA MEIRA

MONOGRAFIA RELACIONADA AO PROJETO DE OFICINAS DE INTEGRAÇÃO II
SOB O TÍTULO “ROBÔ SEGUIDOR DE LINHA”. APROVADA EM: __/__/____.

BANCA EXAMINADORA

João Alberto Fabro, Universidade Tecnológica Federal do
Paraná

Mário Sérgio Teixeira de Freitas, Universidade Tecnológica
Federal do Paraná

Leandro Piekarski do Nascimento, Universidade Tecnológica
Federal do Paraná

RESUMO

GIOPPO, Lucas; HIGASKINO, Marcelo; COSTA, Ricardo; MEIRA, William. Robô Seguidor de Linha. 34 f. Monografia – Curso de Engenharia da Computação, Universidade Tecnológica Federal do Paraná. Curitiba, 2009.

Este projeto tem como objetivo a montagem de um robô que consiga seguir a trajetória de uma linha de cor branca ou preta em um fundo de cor oposta à da linha. O seu sistema para detecção da linha contará com o uso de sensores de luz infravermelha para detectar o sinal refletido a partir de um diodo emissor apropriado. As superfícies pretas têm a propriedade de absorver radiações de todos os comprimentos de onda na faixa visível.

Palavras-chave: Robô Seguidor de Linha, Arduino, Ponte H, Motor CC, Fotodiodo

ABSTRACT

GIOPPO, Lucas; HIGASKINO, Marcelo; COSTA, Ricardo; MEIRA, William. Line Following Robot. 34 f. Monografia – Curso de Engenharia da Computação, Universidade Tecnológica Federal do Paraná. Curitiba, 2009.

This project aims at assembling a robot which is capable of following a white or black colored line having its opposite color as background. The system for the line detection uses infrared light sensors to detect the reflected signal, which is emitted from an infrared LED. The robot will be able to take the right way due to the fact of black being a color that absorbs light and white being a color that reflects light.

Keywords: Line Following Robot, Arduino, H-Bridge, Motor DC, Photodiode

LISTA DE FIGURAS

FIGURA 1 – ARDUÍNO.	8
FIGURA 2 – DIAGRAMA DE BLOCOS.	9
FIGURA 3 – ESTRUTURA DE UM MOTOR DE CORRENTE CONTÍNUA.	10
FIGURA 4 – MOTOR AK2800/5-R330.	11
FIGURA 5 – ESQUEMA DE PONTE H.	12
FIGURA 6 – FUNCIONAMENTO DA PONTE H.	12
FIGURA 7 – PONTE H FEITA APARTIR DE COMPONENTES DISCRETOS. ...	13
FIGURA 8 – PONTE H L298N.	14
FIGURA 9 – PINOS DE CONEXÃO - L298N	15
FIGURA 10 – SÍMBOLO FOTODIODO.	15
FIGURA 11 – CIRCUITO DO SENSOR	16
FIGURA 12 – VERIFICAÇÃO PELO SENSOR DA PRESENÇA DA LINHA.	16
FIGURA 13 – DISPOSIÇÃO DOS SENSORES.	17
FIGURA 14 – PWM VARIANDO A TENSÃO MÉDIA.	18
FIGURA 15 – DEFINIÇÃO DE CICLO ATIVO.	18
FIGURA 16 – GRÁFICOS DE PWM COM COMANDOS DO ARDUÍNO.	19
FIGURA 17 – DISPOSIÇÃO DOS COMPONENTES DO ROBÔ SOBRE O CHASSI.	21
FIGURA 18 – DIAGRAMA DE ESTADOS.	22

SUMÁRIO

1	INTRODUÇÃO	6
1.1	OBJETIVOS	6
1.1.1	Objetivo Geral	6
1.1.2	Objetivos Específicos	6
2	FUNDAMENTAÇÃO TEÓRICA.....	8
2.1	MICROCONTROLADOR	8
2.2	MOTOR DE CORRENTE CONTÍNUA	10
2.3	PONTE H	12
2.4	SENSOR	14
2.5	PWM	17
3	DESENVOLVIMENTO PRÁTICO.....	20
3.1	ESTRUTURA.....	20
3.2	SOFTWARE.....	20
4	CONCLUSÃO.....	23
	REFERÊNCIAS	24
	APÊNDICE A – ORÇAMENTO.....	25
	APÊNDICE B – REUNIÕES (APS).....	26
	APÊNDICE C – CÓDIGO DO PROGRAMA DESENVOLVIDO.....	31

1 INTRODUÇÃO

Este trabalho apresenta os passos para construir um robô capaz de seguir uma linha no chão. A monografia começa com uma visão geral dos objetivos do trabalho e em seguida apresenta detalhes do planejamento, da montagem do hardware e da implementação do software que serão usados no robô. Na locomoção serão utilizados motores de corrente contínua alimentados por uma bateria, além de caixas de redução para aumentar o torque dos motores e circuitos que permitem inverter o sentido ou variar velocidade de rotação dos motores. Para detectar a presença da linha preta o robô estará equipado com emissores e receptores de luz na faixa infravermelho baseados na reflexão da luz pela cor branca e absorção pela luz preta. Para controlar tudo será utilizado o microcontrolador Arduíno.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

Desenvolver um robô que se movimente sobre uma linha preta ou branca em uma superfície plana e de cor oposta à da linha, guiado por sensores à base de fotodiodos que recebem a reflexão de um sinal emitido por um diodo emissor de infravermelho, atendendo assim ao objetivo principal da matéria de Oficinas de Integração 2, do curso de Engenharia de Computação da Universidade Tecnológica Federal do Paraná – Campus Curitiba, que é a de montar um aparato que utilize sensores de qualquer natureza, trate os dados gerados por eles e responda a eles de alguma maneira.

1.1.2 Objetivos Específicos

- Entender a teoria por trás dos componentes utilizados em nosso projeto, tais como fotodiodos, transistores, capacitores, etc.
- Realizar um estudo sobre dimensionamento de um motor para determinada tarefa ou, em nosso caso, escolher um motor de corrente contínua adequado para mover o robô,

entendendo o funcionamento deste.

- Aprender a teoria, a montar e utilizar uma ponte H, que em nosso projeto será utilizada para inverter o sentido de rotação do motor CC.
- Aprender a utilizar o microcontrolador Arduíno e a linguagem de programação utilizada nele.
- Estudar sobre PWM (modulação por largura de pulso) e como gerar este pulso utilizando o Arduíno. Este estudo será utilizado para controlar a velocidade dos motores e assim conseguir realizar uma curva.
- Aprender a montar e utilizar sensores de infravermelho.
- Realizar um bom ajuste entre o relacionamento sensores, motores e microcontrolador.

2 FUNDAMENTAÇÃO TEÓRICA

Para montar um robô é necessário entender alguns conceitos e o funcionamento dos componentes e peças envolvidos no trabalho, a fundamentação teórica diz respeito a este ponto do projeto onde é explicado as etapas do processo de aprendizagem para se atingir o objetivo.

2.1 MICROCONTROLADOR

Optou-se por escolher o Arduíno, figura 1, como microcontrolador para o projeto, este dispositivo será responsável pela ligação entre o sensor e o motor. As informações geradas pelos sensores são analisadas pelo microcontrolador e então são passados comandos – através da ponte H – para o motor executar o movimento apropriado para se manter sobre a linha, o diagrama de blocos dos componentes é a figura 2.

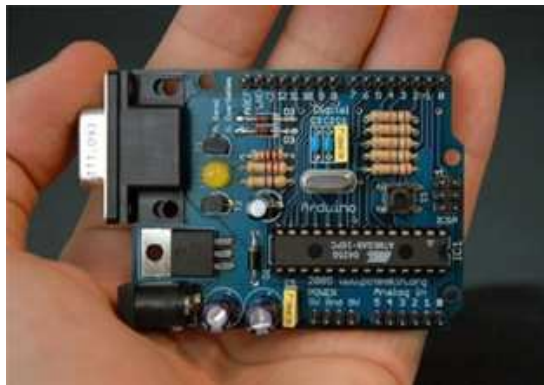


Figura 1: Arduíno.

Fonte: (BANZI et al., 2006)

O Arduíno é uma plataforma de prototipagem eletrônica *open-source*, baseado em um microcontrolador que se conecta ao computador pessoal através de uma porta serial ou USB, dependendo do modelo utilizado. Ele possui uma linguagem de programação própria, baseada em Wiring, que é implementada em um ambiente de desenvolvimento (IDE), também próprio, baseado em Processing, e que pode ser utilizada em vários sistemas operacionais. Através desta

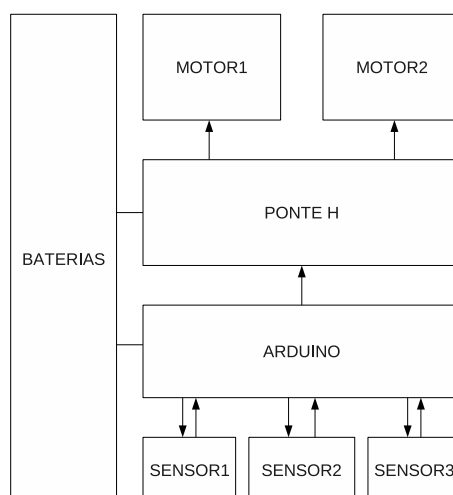


Figura 2: Diagrama de Blocos.

Fonte: Autoria própria.

plataforma pode-se montar uma variedade de circuitos de forma fácil e ágil, como, por exemplo, um sistema de sensores, cujos dados podem ser interpretados e utilizados pelo próprio Arduino ou repassados para um PC(*personal computer*, ou computador pessoal). A sua alimentação se dá pela porta USB conectada ao PC ou por uma fonte externa de até 25V (INTERFACING. . . , 2009).

A linguagem de programação do Arduino é simples e sua sintaxe se assemelha muito ao C++ e ao Java. Um programa típico possui duas funções básicas, o “void setup()”, que é executada logo no início do programa, e o “void loop()”, que é a função executada repetidamente pelo microcontrolador. Fora elas existem diversas outras funções como por exemplo as que fazem com que um pino emita e receba sinais digitais ou analógicos.

O modelo do Arduino escolhido para ser utilizado neste projeto foi o Freeduino BR v1.0 que utiliza um microcontrolador ATmega328P com interface USB e também um conector de alimentação externa.

Uma série de vantagens fazem do Arduino a escolha para ser utilizada neste projeto, dentre eles estão seu preço atrativo, programação descomplicada e fácil manuseio. Os comandos simples simplificam diversos passos do projeto, dando espaço para a equipe focar a atenção em outras partes, como na montagem dos sensores e no funcionamento dos motores.

2.2 MOTOR DE CORRENTE CONTÍNUA

Motores elétricos convertem energia elétrica em mecânica. Existem motores de corrente contínua (CC) e de corrente alternada (CA), cada um com diversas variações. Motores de corrente alternada são geralmente usados para máquinas grandes e recebem energia diretamente da rede de distribuição de energia. Segundo (JONES; FLYNN; SEIGER, 1999), robôs móveis usam tipicamente corrente contínua, pois sua fonte de energia é uma bateria.

Para as necessidades de locomoção de um robô, o motor gira em uma velocidade muito alta e com um torque muito baixo. Para inverter essa relação, o motor deve ser ligado a uma caixa de redução que produz uma nova saída que gira mais devagar, porém com um torque maior. Muitos motores CC são vendidos com a caixa de redução já instalada, como é o caso do adquirido para este projeto.

Os motores de corrente contínua têm pelo menos dois terminais que ao se aplicar uma tensão nestes terminais o motor gira em um sentido e invertendo a polaridade da entrada o motor gira no sentido contrário.

Alguns motores CC, como motores de passo, tem mais que dois terminais. Os sinais aplicados nestes terminais energizam partes do motor organizadamente de modo a controlar o ciclo interno do motor. Este controle permite saber a posição do motor em cada instante e obter uma precisão maior da velocidade de rotação do motor. Neste projeto a trajetória do robô pode ser constantemente corrigida a partir da linha, então não será necessário o uso desse tipo de motor.

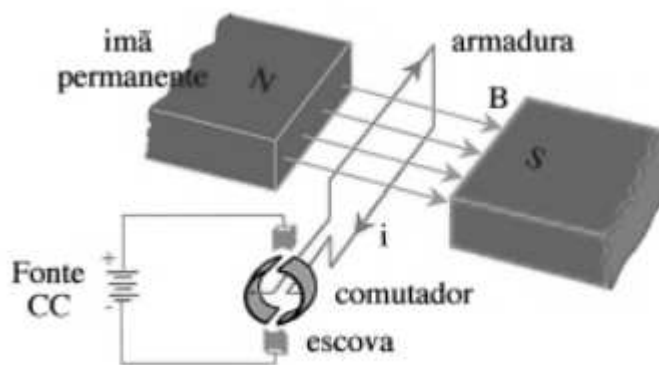


Figura 3: Estrutura de um motor de corrente contínua.

Fonte: (ANTUNES, 2009)

A figura 3 mostra uma simplificação de um motor CC com ímã permanente. Quando um fio mergulhado em um campo magnético é atravessado por uma corrente, surge uma força. O

sentido e a intensidade dessa força podem ser calculados pela equação 1.

$$\vec{F} = i\vec{L} \times \vec{B} \quad (1)$$

Duas vezes por ciclo os comutadores mudam de contato, invertendo o sentido da corrente na armadura. Esta mudança do sentido da corrente é essencial para que o torque sobre o comutador esteja atuando sempre no mesmo sentido.

A corrente de partida de motores CC é muito grande, as vezes até maior que o valor em regime permanente. Isto ocorre pois quando o motor é ligado, a armadura está parada e portanto a força contra-eletromotriz E é nula. Logo, toda a tensão sobre o motor é aplicada na resistência da armadura, que é pequena, resultando em uma corrente alta. Isto pode ser melhor compreendido através da equação 2.

$$V = RI + E \quad (2)$$

Por outro lado, essa mesma equação nos mostra que quando a tensão no motor é cessada bruscamente, a energia armazenada na armadura transforma-se em uma corrente reversa. Para prevenir que esta corrente danifique o circuito são colocados diodos em paralelo com os transistores.

Os motores comprados para o robô são do modelo AK280/5-R330, (TECNOLÓGICAS,), e foram comprados com caixa de redução de 1 para 70 embutidos. O motor com a caixa de redução está representado na figura 4. A eficiência máxima do motor é atingida em 1,44 A, o motor aplica um torque de $0,63 \text{ kgf} \cdot \text{cm}$ e atinge a velocidade angular de 280 rpm. Com as rodas e a caixa de redução isto resulta em um torque de 17,3 N e uma velocidade de 1,7 cm/s.



Figura 4: Motor AK2800/5-R330.

Fonte: Autoria Própria.

2.3 PONTE H

Dentro do projeto, os motores CC precisam ser acionados nos dois sentidos, portanto há a necessidade de controlar o sentido da corrente fornecida para que o motor inverta o sentido de rotação. Esse controle é feito pela ponte H, figura 5, que controla o sentido da corrente para o motor através da disposição de quatro chaves eletrônicas em que o motor é posicionado entre elas, de modo que formem uma letra “H”, daí o seu nome.

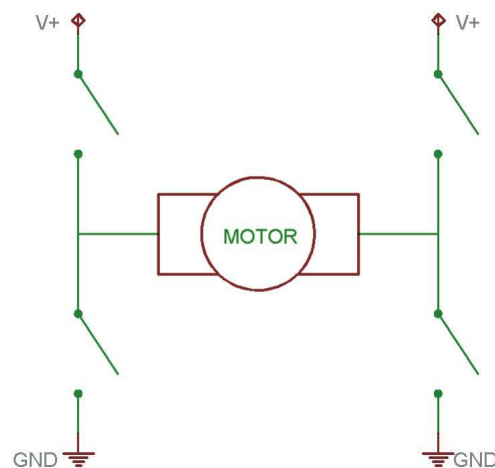


Figura 5: Esquema de Ponte H.

Fonte: (PATSKO, 2006)

O fornecimento de corrente ocorre pelo fechamento das chaves duas a duas, de tal forma que a fonte e o terra estejam em posições alternadas dos pólos do motor como indicado na imagem 6.

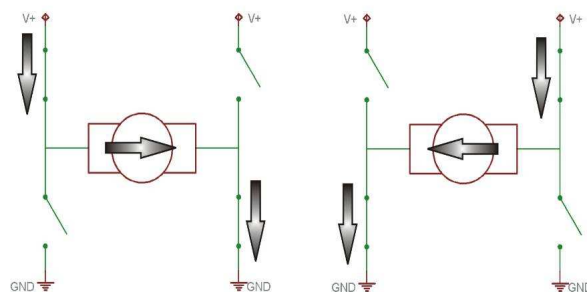


Figura 6: Funcionamento da Ponte H.

Fonte: (PATSKO, 2006)

Caso seja necessário parar o motor CC, a ponte H propicia duas formas distintas para fazê-lo. A primeira consiste em abrir as quatro chaves da ponte, fazendo com que a corrente gerada

pela fonte não circule pelo motor, então o movimento nele é apenas inercial e cessa suavemente devido apenas pela força do atrito. A segunda maneira é feita fechando as duas chaves superiores, que estão ligadas a fonte, ou fechando as duas chaves inferiores, ligadas ao terra. Desta forma produz-se um “freio eletrônico”, parando o motor quase instantaneamente. Isso ocorre pois as chaves fazem com que o motor entre num “curto-circuito” e seja freado justamente pela corrente inversa gerada pela natureza indutiva do próprio motor, que o obriga a girar em sentido contrário.

Um caso especial que se deve tomar cuidado é a de não fechar as quatro chaves ao mesmo tempo ou duas chaves de um mesmo lado, pois isso faz com que “o fluxo da corrente vá direto do pólo positivo para o negativo, causando um curto-circuito fatal para a fonte de alimentação e para os componentes eletrônicos envolvidos no circuito” (PATSKO, 2006).

No caso deste projeto, a equipe envolvida pode optar por duas formas distintas da ponte H, montar uma com componentes discretos, como transistores e diodos ou utilizar o circuito integrado L298N (MICROELECTRONICS, 2000).

Na ponte H com componentes discretos, figura 7, foram utilizados transistores MOSFETs como chaves que para fechá-las basta colocar uma tensão na base deles. Os resistores são adicionados para evitar que haja sobrecarga da corrente e os diodos para evitar que uma possível corrente reversa gerada ao acionar o freio dos motores possa danificar o circuito.

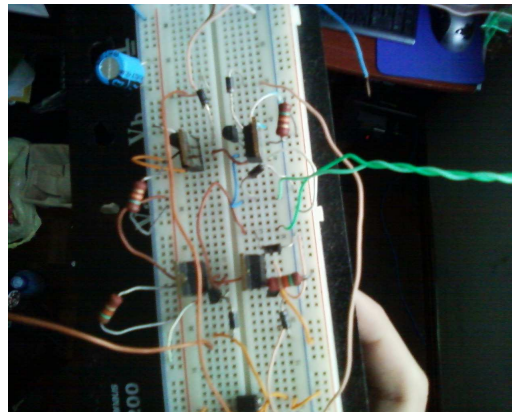


Figura 7: Ponte H feita apartir de componentes discretos.

Fonte: Autoria própria.

No L298N, figura 8, são implementadas duas pontes H. A sua utilização é diferente da ponte H anterior e possui algumas peculiaridades como podem ser observadas no *datasheet*. Dentre as diferenças estão: a existência do “sense” que deve ser conectada a um resistor para o terra com o intuito de controlar a corrente de carga de cada ponte; a existência de quatro pinos

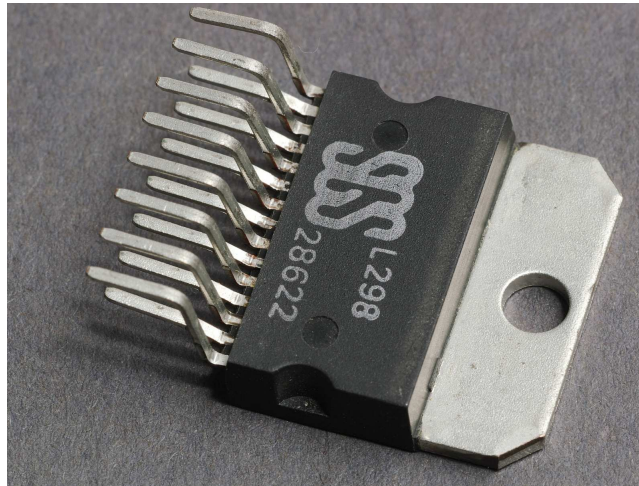


Figura 8: Ponte H L298N.

Fonte: Autoria própria.

“out”, duas para cada ponte, que é por onde sai a corrente controlada; a existência de um pino “ V_s ” para a entrada da tensão das duas pontes; “ V_{ss} ” para a alimentação dos blocos lógicos; “GND” para o terra; dois “Enable”, um para cada ponte, que funciona justamente para ativar as pontes; e quatro pinos “Input”, dois para cada ponte, que determinam por qual “out” sairá a tensão fornecida em “ V_s ”. No datasheet consta ainda quatro estados para a ponte H: no caso do Input1 estar recebendo sinal e do Input2 não, o motor funciona para “frente”, no caso contrário o motor funciona para “trás”. Caso os dois Input estejam iguais, ou seja, ambos recebendo sinal ou ambos não recebendo sinal, temos o freio do motor, e caso enable seja desligado o motor roda apenas com a inércia.

Por fim, a equipe decidiu pela utilização do L298N, pois o seu encapsulamento, figura 9, já possui duas pontes H integradas, ideal para o caso, pois necessitamos controlar dois motores CC de forma independente através do microcontrolador. Além disso outra vantagem é o seu dimensionamento, pois ocupa um espaço muito menor do que duas pontes H feitas com componentes discretos.

2.4 SENSOR

Os sensores utilizados no projeto são fototransistores e fotodiodos, funcionando como receptores e emissores respectivamente. Basicamente eles verificam se o protótipo está ou não em cima do seu trajeto e se ele precisa fazer alguma correção na rota.

Todo o funcionamento desses sensores está baseado numa propriedade fundamental dos fo-

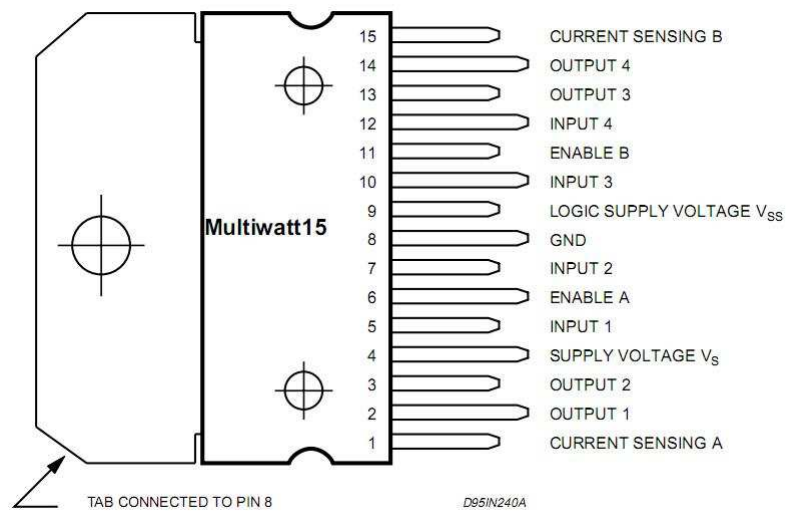


Figura 9: Pinos de Conexão - L298N

Fonte: (MICROELECTRONICS, 2000)

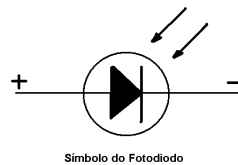


Figura 10: Símbolo Fotodiodo.

Fonte: Autoria própria.

todiodos, cujo símbolo está representado na figura 10. Esses componentes eletrônicos quando instalados em polaridade reversa possuem uma resistência elétrica praticamente infinita assim como qualquer diodo em polaridade reversa. No entanto quando um feixe de luz infravermelha é direcionado para o fotodiodo, sua resistência elétrica cai drasticamente e é através dessa propriedade que é possível verificar se o robô móvel está ou não sobre a faixa de cor escura utilizada para orientar o percurso que deverá ser percorrido. O funcionamento do fototransistor é basicamente o mesmo do fotodiodo, entretanto ele é mais sensível ao sinal luminoso pois basta uma pequena corrente na base para que ele altere seu estado verificando então a presença da luz.

O diodo emissor (LED) emite um feixe luminoso que reflete sobre uma superfície de cor clara e incide no fototransistor, "receptor", diminuindo sua resistência e permitindo a passagem de corrente que é verificada pelo Arduino. Como a faixa é de cor escura, o feixe luminoso emitido pelo LED é absorvido fazendo com que não haja passagem de corrente pelo fotodiodo. Com isso é verificado se o "carro" está ou não sobre o percurso indicado. Como escolhemos criar um percurso de cor escura o Arduino interpreta a "ausência de luz" como caminho e a

presença, no caso a reflexão da luz emitida, como sendo o fundo fora do caminho. A figura 11 mostra o esquema do circuito individual dos sensores:

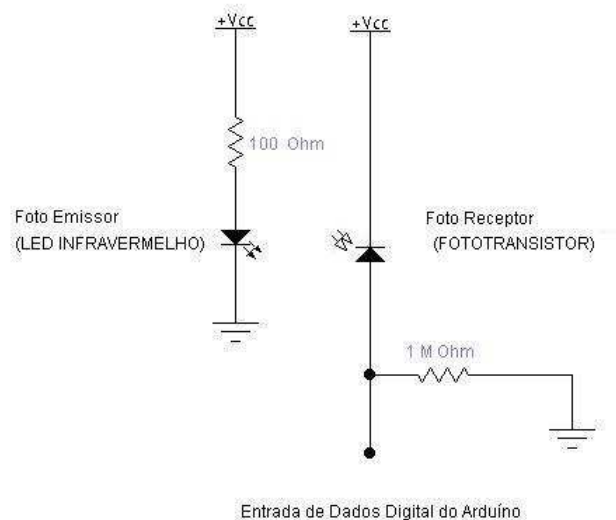


Figura 11: Circuito do Sensor

Fonte: Autoria própria.

A figura 12 mostra como funciona a verificação da presença ou não do percurso e a disposição do par de sensores.

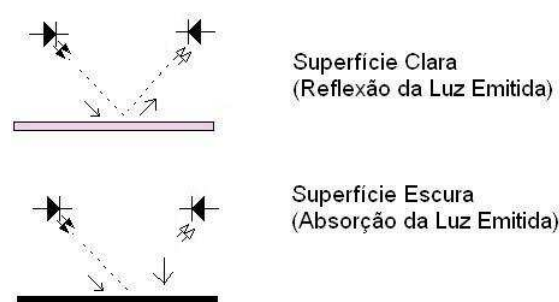


Figura 12: Verificação pelo sensor da presença da linha.

Fonte: Autoria própria.

Os sensores estão dispostos em pares (receptor e emissor) próximos um do outro que são distribuídos de tal forma que um fica na faixa central do robô e os outros dois ficam adiantados e mais para as laterais, com uma distância maior que a da faixa preta utilizada como caminho. Inicialmente, foi previsto que seria necessário haver um isolamento lateral entre o emissor e

o receptor para que não houvesse interferência na captação do sinal, entretanto após feitos alguns testes verificou-se que tal isolamento era desnecessário pois não gerava tal interferência, pelo contrario, ele atrapalhava na reflexão da luz. Eles estão levemente inclinados na diagonal para haver uma reflexão mais precisa e dispostos a uma altura próxima de dois centímetros do solo. Quanto à quantidade de sensores são três pares (receptor e emissor) dispostos de maneira semelhante à figura 13:

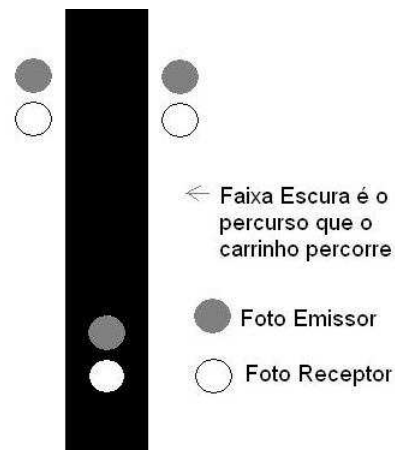


Figura 13: Disposição dos Sensores.

Fonte: Autoria própria.

Tal disposição foi escolhida para que os sensores funcionem da seguinte maneira: o sensor mais recuado verifica sempre se o carrinho está ou não no percurso, funcionando como uma espécie de “estado presente” enquanto os dois mais a frente verificarão em qual direção o carro deverá seguir caso haja uma mudança de trajetória (uma curva), funcionando como verificador de “estado futuro” do protótipo.

2.5 PWM

Um dos objetivos da equipe é poder controlar a velocidade com que os motores do robô giram para melhorar a sua resposta em uma curva. Para isso é necessário que a tensão fornecida para o motor seja variável, entretanto isso não seria possível, à primeira vista, utilizando puramente sinais digitais, que funcionam somente com zero ou tensão máxima. A *pulse width modulation* (PWM), ou modulação por largura de pulso, serve justamente como técnica para obter um sinal analógico a partir de um digital.

Tendo um circuito que possui apenas uma chave e uma carga qualquer, quando a chave é ligada passará a potência máxima sobre a carga e quando ela está desligada a potência será nula.

A pergunta é como fazer para obter um valor diferente de potência sobre a carga entre este valor nulo e a potência máxima, aí está o ponto em que entra a idéia do PWM (GHIRARDELLO, 2008).

O PWM consiste em um padrão de rápida alternância entre liga e desliga de um período T , na qual, parte do tempo ela estará em estado ativo e parte do tempo em estado desativado. Vemos assim que a potência média aplicada a carga pode ser diferente dos valores digitais e que a tensão sobre a carga varia. Veja figura 14.

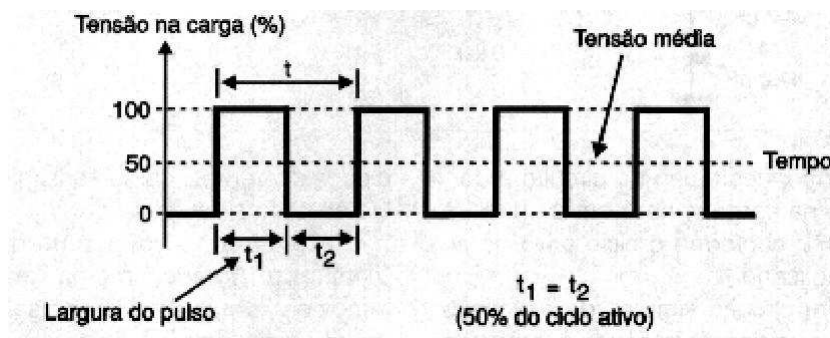


Figura 14: PWM variando a tensão média.

Fonte: (GHIRARDELLO, 2008)

A relação $(t_1 \div t) \times 100\%$ é chamada de ciclo ativo (*duty-cycle*) do PWM, sendo o tempo em que ele fica ativo igual a t_1 e o período igual a t . Caso o ciclo ativo seja igual a 0% o circuito está desligado e caso o ciclo ativo seja 100% o circuito estará sempre ligado. A figura 15 demonstra o que foi explicado.

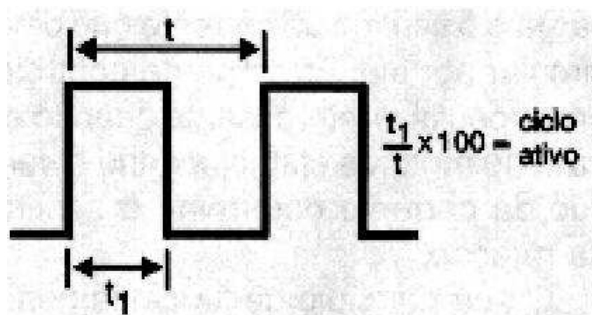


Figura 15: Definição de Ciclo Ativo.

Fonte: (GHIRARDELLO, 2008)

Utilizando o microcontrolador Arduíno é possível fornecer este tipo de sinal utilizando as portas adequadas e o comando em código “`analogWrite(valor)`”, em que valor varia de 0 a 255, sendo 255 a tensão máxima.

Na figura 16 pode-se ver um gráfico representando o funcionamento do PWM para valores variados da função “analogWrite”.

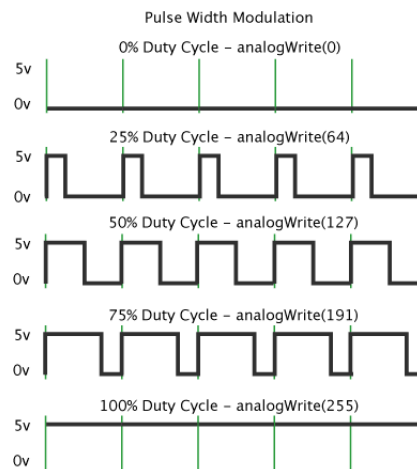


Figura 16: Gráficos de PWM com comandos do Arduino.

Fonte: (NASCIMENTO; DIAS; ZORZO, 2008)

3 DESENVOLVIMENTO PRÁTICO

3.1 ESTRUTURA

O chassi do robô foi feito utilizando papel foam que é um material que é rígido e leve ao mesmo tempo além de poder ser facilmente modelado, características ideais para um protótipo. Os motores com redução e os demais itens para a montagem dos circuitos foram adquiridos em lojas de eletrônica e as rodas foram compradas através da *Internet*. O preço de cada componente pode ser encontrado no apêndice A.

Os sensores ficam sob a parte dianteira do robô e sobre o foam estão os motores seguidos pelo Arduino, pela bateria e pela ponte H. Além das duas rodas do motor o robô é sustentado por mais duas rodinhas menores na traseira. Isso pode ser melhor compreendido através da figura 17.

A alimentação dos sensores e a da ponte H é feita pelo próprio Arduino, que é por sua vez alimentado por uma bateria de 9 V. Os motores são alimentados por um casulo de 6 pilhas AA em série de 1,5 V, somando cerca de 9 V. Para desligar o carrinho foi adicionado uma chave que pode desconectar a bateria do circuito.

Sob o comando do Arduino a ponte H pode controlar a velocidade e o sentido da rotação dos motores. O controle da velocidade é permitida através do comando de PWM do Arduino que varia a tensão de entrada na ponte H.

3.2 SOFTWARE

O software para controlar o robô foi implementado na linguagem própria do Arduino. O programa possui duas funções básicas, o `setup()` e o `loop()` ambas não possuem retorno. A função `setup()` é chamada pelo Arduino uma única vez logo no começo do programa, ou seja, ocorrendo a cada vez que o Arduino é ligado ou resetado. O propósito da função é a de encapsular todas as linhas de código que são utilizadas para inicialização, como a definição dos pinos ou as bibliotecas a serem utilizadas. A função `loop()`, por sua vez, encapsula as linhas

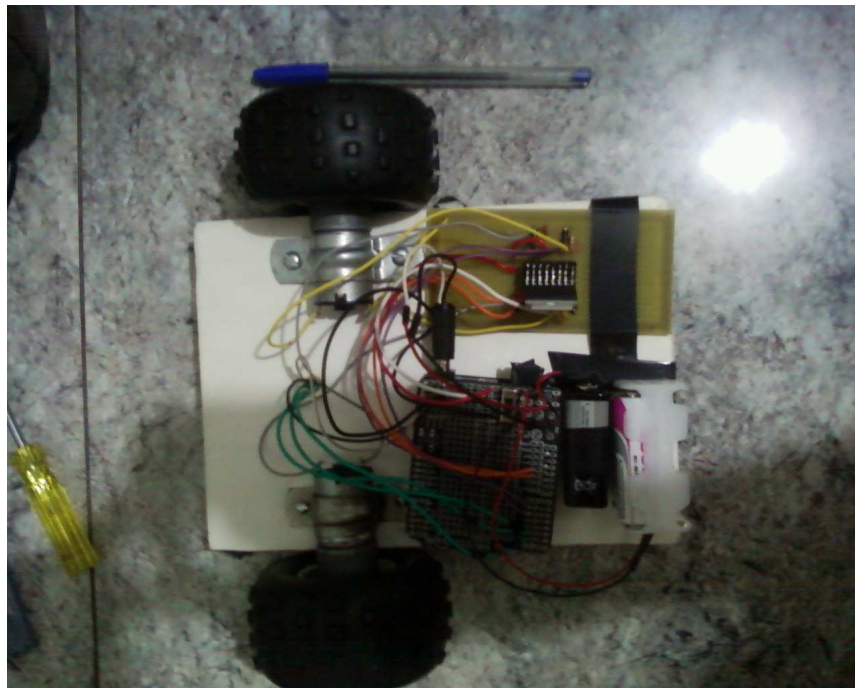


Figura 17: Disposição dos componentes do robô sobre o chassi.

Fonte: Autoria Própria.

de código que serão rodadas em um laço infinito logo após o término do `setup()`, permitindo que o Arduino receba dados, mude de estado e as responda conforme programado. A seguir será explicado brevemente o funcionamento do programa, cujo código pode ser consultado no apêndice C desta monografia.

No programa utilizado pelo robô, o `setup()` inicializa sete terminais ao todo, três para lerem cada um dos três sensores do robô e os outros quatro para controlar os motores através da ponte H, sendo dois terminais para cada motor, um deles estará polarizado em nível lógico baixo e o outro em nível lógico alto, que dependendo da ordem o motor rotacionará para sentidos contrários. Terminado a inicialização dos terminais, o Arduino chama o `loop()` e começa a rodá-lo repetidamente. Dentro dele há quatro condicionais básicas. A primeira é para o caso da linha ser detectada pelo sensor frontal direito, fazendo com que o motor esquerdo tenha uma velocidade maior que o direito. A segunda é para o caso da linha ser detectada pelo sensor frontal esquerdo, fazendo com que o motor direito tenha uma velocidade maior que o esquerdo. A terceira é para o caso da linha ser detectada somente pelo sensor central, fazendo com que os dois motores tenham a mesma velocidade. A última é para o caso da linha não ser detectada por nenhum dos sensores, então avalia-se qual dos três casos anteriores foi o último a ser rodado. Essa informação é guardada por uma variável global chamada “ultimoEstado” que é alterada conforme o estado atual, por padrão ele é inicializado como se o último estado tivesse sido uma

reta. Caso o último estado tenha sido uma curva para a direita, o programa para o motor da direita e faz o motor da esquerda funcionar numa velocidade mais baixa. Caso tenha sido para a esquerda, o programa faz o mesmo, mas para os motores trocados. Caso tenha sido uma reta, então os motores são ligados para fazerem o robô andar de ré.

Cada uma das quatro condicionais possui um filtro para evitar que leituras errôneas e momentâneas afetem o robô de forma perceptível. Esse filtro funciona de tal maneira que faz contagens de quantas vezes dada leitura ocorre, somente depois de ocorrer por 5 vezes o código da condicional é realmente rodado. Esse filtro foi implementado pois já havia sido constatado que o Arduíno acaba lendo erroneamente os dados dos sensores por algumas raras vezes, mas que poderiam tornar a trajetória do robô caótica por alguns instantes.

Na figura 18 temos o diagrama de estados do programa utilizado.

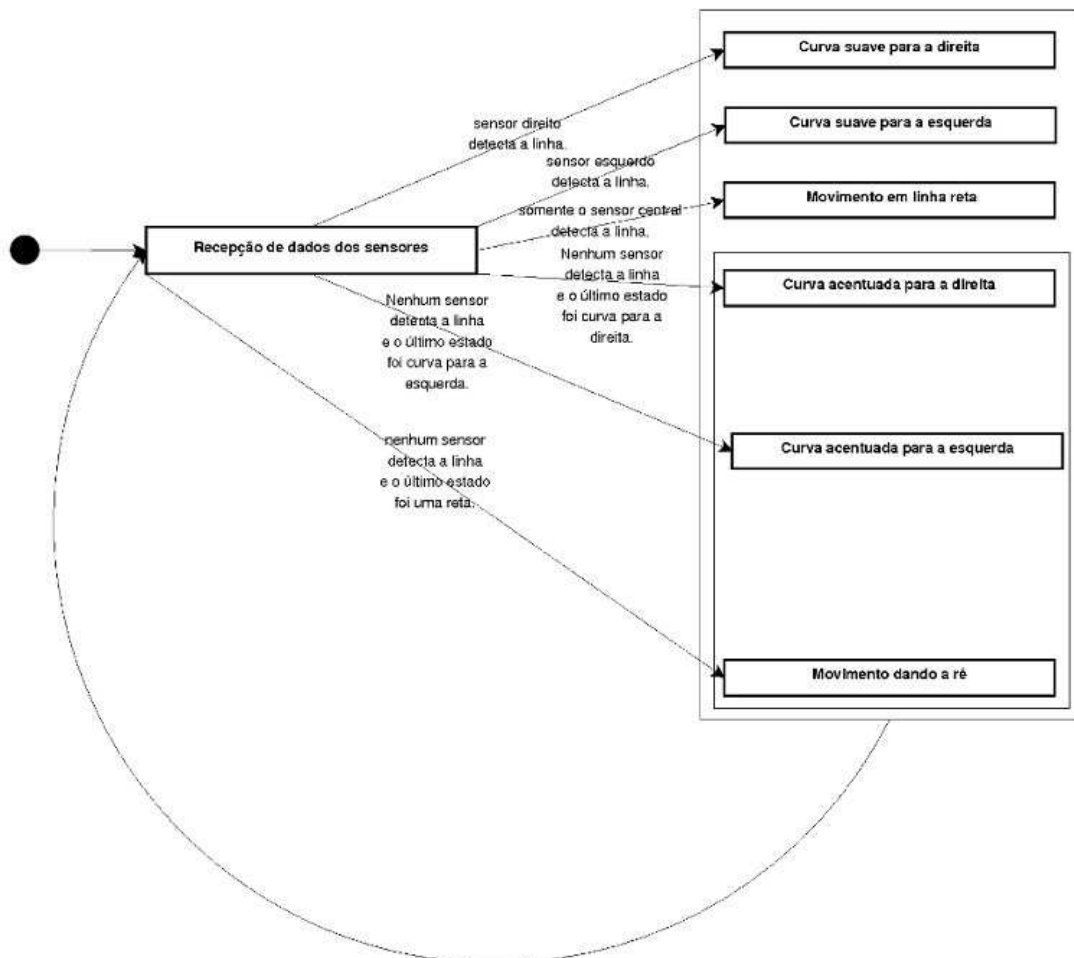


Figura 18: Diagrama de Estados.

Fonte: Autoria Própria.

4 CONCLUSÃO

O projeto desenvolveu um robô seguidor de linhas personalizado ao utilizar uma ponte H e propor uma disposição diferente dos sensores. Os robôs mais simples podem usar um transistor no lugar da ponte H e apenas dois sensores. Essas personalizações no projeto permitiram desenvolver um algoritmo para voltar em busca da linha quando nenhum sensor está a detectando. O software armazena o estado anterior devido o caso da perda da linha e faz o carrinho voltar em marcha ré pelo mesmo caminho percorrido antes de perdê-la ou acentuar a curva, assim espera-se que os sensores voltem a detectar a linha.

O microcontrolador possui entradas e saídas de dados não utilizadas permitindo a expansão do número de sensores ou outros dispositivos. O software também pode ser melhorado, incluindo algoritmos para outros casos, quando a pista for mais complexa. Já o Arduíno possui memória disponível para expansão. Os motores também estão superdimensionados e oferecem uma força e velocidade acima do necessário.

A equipe investiu tempo demasiado para a escolha e compra dos motores, afetando profundamente as previsões iniciais do cronograma do projeto. Faltou montar o robô em uma estrutura mais confiável que o foam, como madeira ou acrílico. No entanto, os objetivos da disciplina que eram utilização de sensores fornecidos por eles foram plenamente atingidos pelo grupo.

REFERÊNCIAS

- ANTUNES, F. L. M. **Motores de Corrente Contínua**. 2009. <http://www.dee.ufc.br/fantunes/MaquinasEletricasI/Aulas/MotoresCorrenteContinua-Impressao.pdf>. Acesso em: 20 nov. 2009.
- BANZI, M. et al. **ARDUÍNO**. 2006. <http://www.arduino.cc>.
- GHIRARDELLO, A. **Apostila sobre Modulação PWM**. 2008. http://www.eletronica.org/arq_apostilas/apostila_pwm.pdf. Acesso em: 20 nov. 2009.
- INTERFACING with Hardware. [S.l.]: Arduino, 2009. <http://www.arduino.cc/playground/Main/InterfacingWithHardware>. Acesso em: 12 nov. 2009.
- JONES, J. L.; FLYNN, A. M.; SEIGER, A. B. **Mobile Robots: Inspiration to Implementation**. 2. ed. [S.l.]: Natick, Massachusetts: A K Peters, 1999.
- MICROELECTRONICS, S. T. **L298 datasheet**. 2000. <http://www.datasheetcatalog.org/datasheet/SGSThompsonMicroelectronics/mXrqxz.pdf>.
- NASCIMENTO, L. P.; DIAS, L. del C.; ZORZO, R. de R. **Cadeira de Rodas Motorizada** — Universidade Tecnológica Federal do Paraná, Curitiba, 2008.
- PATSKO, L. F. **Tutorial Montagem da Ponte H**. 2006. [http://www.maxwellbohr.com.br/downloads/Tutorial Eletronica - Montagem de uma Ponte H.pdf](http://www.maxwellbohr.com.br/downloads/TutorialEletronica-MontagemdeumaPonteH.pdf).
- TECNOLÓGICAS, A. S. **AK280 com redução**. [http://www.motores.akiyama.com.br/pdf/AK280 - Red.pdf](http://www.motores.akiyama.com.br/pdf/AK280-Red.pdf).

APÊNDICE A – ORÇAMENTO

Aqui estão listados todas os itens utilizados em nosso projeto.

Tabela 1: Itens utilizados para a montagem do robô

Nome	Quantidade	Preço por unidade (R\$)
Motor DC com caixa de Redução AK280/5-R330	2x	37,00
Microcontrolador Freeduino BR v1.0 ATMEGA328 e Shield	1x	110,00
Roda com 80 mm de diâmetro	2x	10,00
LED Emissor INFRA-RED	3x	0,50
Fotodiodo	3x	0,50
Circuito Integrado L298N	1x	15,00
Resistores de 1 M Ω	3x	0,20
Resistores de 100 Ω	3x	0,20
Resistores de 1 Ω	2x	0,20
Capacitor 100nF	2x	0,30
Diodo 1N4004	4x	0,20
Placa de Circuito 10x5 cm	2x	3,00
Suporte para 6 Pilhas AA	1x	5,00
Pilha AA	6x	1,00
Roda Auxiliar	1x	0,00
Placa de Foam 30x15 cm	1x	0,00
Total	1x	242,00

APÊNDICE B – REUNIÕES (APS)

Para atender os pedidos de execução dos APS por parte da Universidade Tecnológica Federal do Paraná os professores da matéria de Oficinas de Integração 2 decidiram pela realização de registros acerca da evolução do projeto. Esses registros deveriam ser simples, constando os passos mais importantes ou as reuniões feitas pela equipe e os temas tratados durante o período em questão.

Segue abaixo os registros feitos desde o pedido do APS feito pelos professores até a conclusão da monografia, antes de sua impressão.

Reunião pela internet (16/09/2009)

Integrantes: Marcelo e William

Tópicos abordados: motores a serem utilizados, PWM do Arduino e explicação teórica dos tópicos.

Tópicos resolvidos: utilização do PWM no Arduino através de uma ponte H.

Tópicos pendentes: escolha do motor ideal.

Reunião na UTFPR (25/09/2009)

Integrantes: William

Tópicos abordados: Ponte H (mosfets)

Tópicos concluídos: funciona, mas uma única ponte H

Reunião pela internet (27/09/2009)

Integrantes: Marcelo e William

Tópicos abordados: Composição do robô, união de sensores + ponte H + Arduino.

Tópicos concluídos: reunião na terça (29/09/2009) para resolver os esquemas de sensores

+ ponte H + Arduino.

Reunião casa do William (29/09/2009)

Integrantes: Lucas, Marcelo, William, Ricardo

Tópicos abordados: Montagem da ponte H (L298N)

Tópicos concluídos: Montagem feita na protoboard, utilizando o CI, bateria 9V. Funciona com dois num único CI. Mas o L298N funcionou parcialmente, mandado para o Lucas Longen para fazer reparos.

Reunião em aula (07/10/2009)

Integrantes: William, Lucas, Ricardo, Marcelo

Tópicos abordados: L298N

Tópicos concluídos: Mal contatos resolvidos, ponte H funcionando, mas a bateria de 9V não fornece corrente suficiente para partida dos motores através da ponte H, necessário uma fonte diferente. Desenvolvimento do layout dos sensores.

Tópicos pendentes: Próximo passo é o desenvolvimento da carcaça do robô.

Reunião em aula (14/10/2009)

Integrantes: William, Lucas, Ricardo, Marcelo

Tópicos abordados: Relatório de Qualificação.

Tópicos concluídos: Ajustes finais do relatório que foi feito em OpenOffice.

Reunião em aula (21/10/2009)

Integrantes: William, Lucas, Ricardo, Marcelo

Tópicos abordados: Redução da ponte H, programas controladores do robô e rodinhas para o robô.

Tópicos resolvidos: Ponte H pronta para ser passada para a placa de circuito universal. Programa semi-pronto.

Tópicos pendentes: Rodinhas para o robô.

Reunião pela internet (24/10/2009)

Integrantes: William, Lucas, Ricardo

Tópicos abordados: Rodinhas do robô

Tópicos resolvidos: Compra decidida e efetuada.

Pós reunião: chegada prevista inicialmente para 28/10/2009. Chegada efetiva das rodinhas para o robô em 30/10/2009.

Reunião na casa do William (04/11/2009)

Integrantes: William, Lucas

Tópicos abordados: Encaixe das rodinhas.

Tópicos concluídos: Encaixe das rodinhas auxiliado por um pedaço de tecido.

Tópicos pendentes: Montagem do robô.

Reunião na casa do William (10/11/2009)

Integrantes: William, Lucas, Ricardo, Marcelo

Tópicos abordados: montagem do robô.

Tópicos concluídos: Montagem dos sensores na placa de circuito universal e chassi feito a partir de papel foam.

Tópicos pendentes: Montagem da ponte H, interruptor de energia no shield do Arduino, reunião das peças no chassi.

Reunião na UTFPR (17/11/2009)

Integrantes: William, Lucas, Ricardo, Marcelo

Tópicos abordados: Montagem do robô

Tópicos concluídos: Circuito dos sensores pronto e preso ao corpo do robô, motores com roda prontos e presos ao robô, placa de circuito universal da ponte H semi-pronto, Shield do Arduino preparado para receber os contatos, primeira versão do programa pronta para ser testada.

Tópicos pendentes: Esquema do interruptor, problema na fonte de alimentação → ponte H → motores, falta de corrente ou tensão.

Reunião pela internet (17/11/2009)

Integrantes: William, Lucas, Ricardo, Marcelo

Tópicos abordados: Monografia

Tópicos concluídos: Layout final, orçamento semi-pronto, estrutura semi-pronta.

Tópicos pendentes: Orçamento, resumo das reuniões, estrutura do robô, problemas e soluções, próximos passos.

Reunião na UTFPR (18/11/2009)

Integrantes: William, Lucas, Ricardo, Marcelo

Tópicos abordados: Montagem do robô

Tópicos concluídos: Placa de circuito universal da ponte H pronto, shield do Arduino com interruptor e plug de alimentação. 4 pilhas AA são suficientes para dar partida nos motores.

Tópicos pendentes: Fixar os circuitos no chassi no robô. Ligar ponte H, sensores e fonte de alimentação no Arduino. Ligar motores na ponte H. Conclusão da monografia.

Reunião pela internet (18/11/2009)

Integrantes: William, Lucas, Ricardo, Marcelo

Tópicos abordados: Monografia

Tópicos concluídos: Resumos das reuniões para APS de Oficinas de Integração 2. Passagem da monografia para a segunda versão do \LaTeX . Orçamento. Parte da explicação sobre estrutura.

Tópicos pendentes: Explicação sobre o software e a conclusão.

Reunião na UTFPR e na casa do William (19/11/2009)

Integrantes: William, Lucas, Ricardo, Marcelo

Tópicos abordados: Montagem do Robô

Tópicos concluídos: Soldagem dos componentes, problemas com alimentação e sensores.
Orçamento.

Tópicos pendentes: Consertar um receptor queimado. Fazer testes.

APÊNDICE C – CÓDIGO DO PROGRAMA DESENVOLVIDO

Listagem C.1: Código comentado do programa que controla o robô.

```

1  int s1 = 4; //Sensor 1, frente direito
2  int s2 = 2; //Sensor 2, frente esquerdo
3  int s3 = 3; //Sensor 3, tras
4
5  //MOTOR A, lado esquerdo
6  int mA1 = 11; //Motor A 1
7  int mA2 = 10; //Motor A 2
8
9  //MOTOR B, lado direito
10 int mB1 = 5; //Motor B 1
11 int mB2 = 6; //Motor B 2
12
13 int normalA = 255;
14 int normalB = 255;
15 int correcaoA = 220; //Diminuicao da rotacao de um motor para correcao da
    rota
16 int correcaoB = 220; //Diminuicao da rotacao de um motor para correcao da
    rota
17 int parado = 0;
18
19 int LINHA = LOW, FUNDO= HIGH;
20
21 //FILTRO
22 int filtroCurvaD = 0;
23 int filtroCurvaE = 0;
24 int filtroReta = 0;
25 int filtroNada = 0;
26 int filtroN = 5;
27
28 int RETA = 1;
29 int DIREITA = 2;
30 int ESQUERDA = 3;

```

```

31  int ultimoEstado = RETA;
32
33  void setup()
34  {
35      pinMode(s1, INPUT);
36      pinMode(s2, INPUT);
37      pinMode(s3, INPUT);
38
39      pinMode(mA1, OUTPUT);
40      pinMode(mA2, OUTPUT);
41
42      pinMode(mB1, OUTPUT);
43      pinMode(mB2, OUTPUT);
44
45      //Serial.begin(9600);
46  }
47
48  void loop()
49  {
50      if (digitalRead(s2)==LINHA)
51      {
52          //CASO: CURVA DIREITA
53          filtroCurvaD++;
54          if (filtroCurvaD >= filtroN)
55          {
56              analogWrite(mA1, normalA);
57              digitalWrite(mA2, LOW);
58
59              analogWrite(mB1, correcaoB);
60              digitalWrite(mB2, LOW);
61
62              filtroCurvaD = 0;
63              ultimoEstado = DIREITA;
64          }
65      }
66      else if (digitalRead(s1)==LINHA)
67      {
68          //CASO: CURVA ESQUERDA
69          filtroCurvaE++;
70          if (filtroCurvaE >= filtroN)
71          {
72              analogWrite(mA1, correcaoA);
73              digitalWrite(mA2, LOW);

```

```

74
75     analogWrite(mB1, normalB);
76     digitalWrite(mB2, LOW);
77
78     filtroCurvaE = 0;
79     ultimoEstado = ESQUERDA;
80 }
81 }
82 else if (digitalRead(s3)==LINHA && digitalRead(s2)==FUNDO && digitalRead(s1)
        )==FUNDO)
83 {
84     //CASO: LINHA RETA
85     filtroReta++;
86     if(filtroReta >= filtroN)
87     {
88         analogWrite(mA1, normalA);
89         digitalWrite(mA2, LOW);
90
91         analogWrite(mB1, normalB);
92         digitalWrite(mB2, LOW);
93
94         filtroReta = 0;
95         ultimoEstado = RETA;
96     }
97 }
98 else if (digitalRead(s3)==FUNDO && digitalRead(s2)==FUNDO && digitalRead(s1)
        )==FUNDO)
99 {
100     //CASO: LINHA SUMIU
101     filtroNada++;
102     if(filtroNada >= filtroN)
103     {
104         if(ultimoEstado == DIREITA)
105         {
106             analogWrite(mA1, correcaoA);
107             digitalWrite(mA2, LOW);
108
109             digitalWrite(mB1, LOW);
110             digitalWrite(mB2, LOW);
111         }
112         else if(ultimoEstado == ESQUERDA)
113         {
114             digitalWrite(mA1, LOW);

```

```
115         digitalWrite(mA2, LOW);
116
117         analogWrite(mB1, correcaoB);
118         digitalWrite(mB2, LOW);
119     }
120     else if(ultimoEstado == RETA)
121     {
122         digitalWrite(mA1,LOW );
123         analogWrite(mA2, correcaoA);
124
125         digitalWrite(mB1, LOW);
126         digitalWrite(mB2, correcaoB);
127     }
128     filtroNada = 0;
129 }
130 }
131
132
133 }
```
