

AGV – VEÍCULO AUTOMATICAMENTE GUIADO

FERNANDO ITAMAR DO NASCIMENTO SILVEIRA¹

ADILSON MASSA²

CAIO BELTANI DA SILVA FERNANDES³

EDUARDO SOUZA SIMS⁴

Resumo: O presente artigo apresenta o desenvolvimento de um protótipo microcontrolado de um AGV, sigla inglesa referente a Veículos Automaticamente Guiados. Atualmente os AGV's podem ser utilizados nas mais diversas áreas de produção de uma fábrica, tais como logística, linhas de produção contínua. O estudo desse tema tem como objetivo entender e demonstrar a lógica de funcionamento dos AGV's proporcionando ao leitor e interessados a oportunidade de vislumbrar novas aplicabilidades.

Palavras-chaves: agv; automação; logística; microcontrolador.

¹ Graduando do curso de Engenharia Elétrica da Universidade de Araraquara – UNIARA. E-mail: fernandosilveira1990@hotmail.com

² Professor Especialista do curso de Engenharia Elétrica da Universidade de Araraquara – UNIARA. E-mail: adilsonmassa12@gmail.com

³ Graduando do curso de Engenharia Elétrica da Universidade de Araraquara – UNIARA. E-mail: cbs.belt@hotmail.com

⁴ Graduando do curso de Engenharia Elétrica da Universidade de Araraquara – UNIARA. E-mail: eduardo.sims@hotmail.com

AGV - AUTOMATICALLY GUIDED VEHICLE

Abstract: This article presents the development of a microcontrollable prototype of an AGV (Automated Guided Vehicles). Currently the AGV's can be used in the most diverse production areas of a factory, such as logistics, continuous production lines. The study of this theme aims to understand and demonstrate the logic of the operation of AGV's giving the reader and interested the opportunity to glimpse new applicability.

Keywords: agv; automation; logistics; microcontroller.

Lista de ilustrações

Figura 1 – Arquitetura da família MCS-51	5
Figura 2 – Protótipo do AGV	8
Figura 3 – Aspecto do software Proteus	9
Figura 4 – Interface MCU 8051 IDE	9
Figura 5 – Formatos de onda	11
Figura 6 – Conjunto LED e LDR	12
Figura 7 – Circuito divisor de tensão e comparador	13
Figura 8 – Hardware do motor	14
Figura 9 – Desenvolvimento da PCB	14
Figura 10 – Esboço 3D da placa eletrônica	15
Figura 11 – Resultados da placa	15
Figura 12 – Primeiros testes	16
Figura 13 – Montagem final	17

Lista de tabelas

Tabela 1 – Características da família MCS-51	6
Tabela 2 – Combinações Lógicas	13

SUMÁRIO

1 INTRODUÇÃO	4
1.1 Introdução	4
1.2 Objetivo	4
2 REVISÃO BIBLIOGRÁFICAS	5
2.1 Evolução do 8051, seu histórico e características	5
2.2 O mercado da família 8051 e suas aplicações	6
2.3 Os AGV's e suas características	6
3 DESENVOLVIMENTO	8
3.1 Softwares Utilizados	8
3.1.1 Inventor Professional	8
3.1.2 Proteus Professional	8
3.1.3 MCU 8051 IDE	9
3.2 Desenvolvimento da programação	10
3.3 Desenvolvimento do sensor	12
3.4 Desenvolvimento do hardware	13
4 RESULTADOS OBTIDOS	16
4.1 Solução final	16
5 CONCLUSÃO	18
REFERÊNCIAS BIBLIOGRÁFICAS	19

1 INTRODUÇÃO

1.1 Introdução

Atualmente sabe-se que há uma competitividade entre as empresas e a globalização, sabendo disso a busca por sistemas ágeis, inteligentes e de baixo custo de manutenção em todos os departamentos de uma empresa vem se desenvolvendo de forma rápida, buscando sempre o melhor custo benefício.

Sabendo disso, as empresas atualmente buscam modos automatizados para economizar tanto em custos quanto em tempo.

Sistemas automatizados possuem pontos que viabilizam a sua implementação sendo os principais; menores as chances de erros, maior escala de trabalho, sistemas mais precisos e eficazes.

Sendo assim, o projeto que será apresentado nesse relatório se trata de um veículo automaticamente guiado, muito utilizado em sistemas logísticos dentro de uma empresa. Durante esse relatório, será demonstrado um protótipo que utiliza a locomoção do veículo através de um circuito pré-determinado.

1.2 Objetivo

Entre vários modos de AGV's o escolhido por esse grupo foi o seguidor de linha, nesse modo o veículo apenas se moverá quando identificar um circuito.

Existem diversos modelos, porém esse projeto terá como objetivo diferenciar cores em um circuito e a partir de uma programação desenvolvida pelo grupo, tomar "decisões" para a sequência no circuito.

Isso é, ao passar por uma cor, um sensor em conjunto com uma lógica eletrônica mandará um sinal ao microcontrolador, dispositivo responsável por analisar e processar os dados, e então dependendo da cor realizará um comando específico. Por exemplo, caso o AGV passe pela cor vermelha ele deve aumentar a sua velocidade, ou então, passou pela cor verde deve diminuir sua velocidade.

2 REVISÃO BIBLIOGRÁFICA

2.1 Evolução do 8051, seu histórico e características

Um microcontrolador é basicamente um (CI) circuito integrado, porém programável que possui internamente unidades de memórias e interfaces com o mundo externo, porta serial, porta paralela e temporizadores.

Os microcontroladores tiveram sua origem no início dos anos 70, em 1971 foi criado o primeiro microcontrolador o TMS 1000 (TEXAS INSTRUMENTS) basicamente sua função primordial era a utilização em calculadoras e brinquedos eletrônicos da época.

Ainda no início da década de 70 a INTEL fabricante de componentes eletrônicos lançou o microcontrolador 8048 de 4 bits, posteriormente surgindo o seu herdeiro o 8051 de 8 bits já no início dos anos 80. A partir dessa década o microcontrolador de maior sucesso lançado pela INTEL foi a família MCS-51, que são microcontroladores de uso geral com capacidade de memória, e entradas e saídas diferenciados. Na figura 1, mostra-se por meio de um esquema a parte externa de um microcontrolador da família MCS-51.

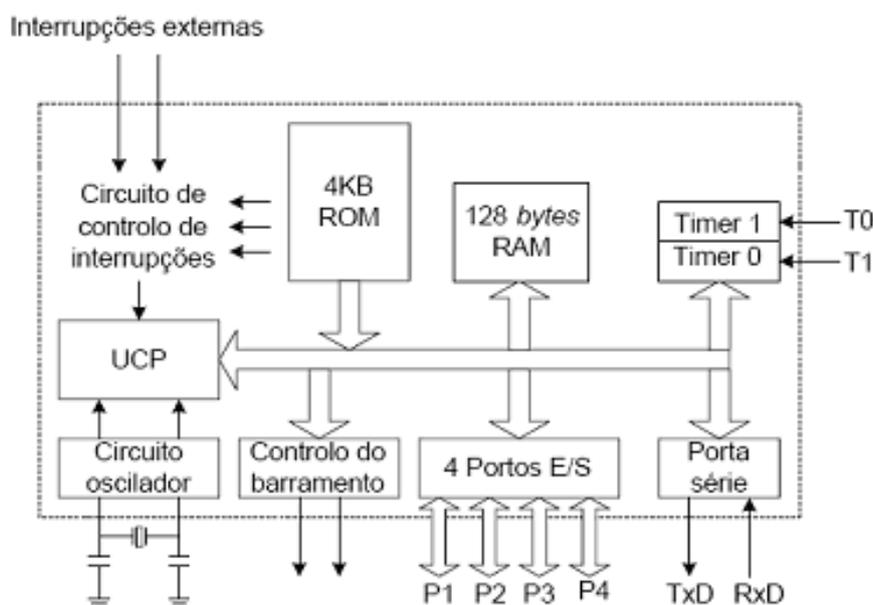


Figura 1 – Arquitetura da família MCS-51
fonte: Imagem Própria

Os microcontroladores dessa família possuem memória de programa e dados internamente com possibilidade de expansão para até 64 Kbytes de programa e mais 64 Kbytes de dados. Permitem o acesso a portas internas de I/O, canal de comunicação serial UART full duplex, interrupções com estrutura “nesting” com 5 fontes mascaráveis de dois níveis de prioridade, timers / counters de 16 bits, oscilador interno, frequência de clock de 12 a 60 MHz. Permitem facilidades de software que possibilita a execução de complexas operações aritméticas e lógicas (multiplicação, divisão, permuta e deslocamento de bits, etc.)

A fabricante INTEL lançou o 8051, com uma memória ROM interna programável de

fabrica, posteriormente lançou o 8751 com EPROM interna programável pelo usuário e o 8031 com necessidade de chips de EPROM externo. Após alguns anos, a família se expandiu para 8052, que tem um timer a mais que o 8051, esta é a família chamada MCS-51. Outros 50 fabricantes tem autorização de fabricados, como Siemens, AMD, Fujitsu, Philips e Oki, entre outros.

A família MCS-51 possui um conjunto de 111 instruções sendo elas: 24 aritméticas; 25 lógicas; 28 transferências de dados; 17 booleanas; 17 desvios. Cerca de 98% destas instruções são de 1 ou 2 ciclos de máquina, oferecendo maior velocidade para seu processamento.

Suas principais características são:

Tabela 1 – Características da família MCS-51

Características da família MCS-51
CPU de 8 bits
4 portas bidirecionais de I/O cada uma de 8 bits,(P0, P1, P2, P3)
32 linhas de E/S
128 bytes de memória RAM interna
4 Kbytes de memória programável ROM
Endereçamento de até 64 Kbytes de memória de dados externa
Endereçamento de até 64 Kbytes de memória de programa externa
Oscilador de clock interno
2 timers/counters (T/C) de 16 bits
1 porta serial
5 fontes de interrupção (2 externas, 2 T/C, 1 serial)

2.2 O mercado da família 8051 e suas aplicações

Após a concessão de fabricação dada pela INTEL a outros fabricantes para reprodução e atualização desse chip, as autorizadas desenvolveram uma grande quantidade de chips derivados do 8051, com significativos aumento de velocidade e significativa redução de consumo de potência.

Com a liberação dada pela INTEL, tem-se varias aplicações do uso dos microcontroladores como em: indústrias, eletrodoméstico, celular, agricultura, carros, segurança.

2.3 Os AGV's e suas características

Os AGV's da sigla inglesa Automated Guided Vehicle (Veículo Guiado Automaticamente), são máquinas autônomas que executam tarefas sem a necessidade de um condutor humano, são projetados para receber e executar instruções e seguir um caminho predefinido.

As principais características de um AGV são: manutenção simples, são flexíveis e possuem capacidade de carga bastante variável. São empregados normalmente em indústrias, armazéns, portos, linhas em loop com paradas para adição ou remoção de cargas e distribuidoras.

Seus modelos variam de acordo com as necessidades da empresa, embora não exista um modelo padrão de AGV, existe uma classificação que é definida de acordo com o tipo de aplicação. Sendo essa classificação: AGV Reboque, AGV Arraste, AGV empilhadeira, AGV para transferência de carga.

3 DESENVOLVIMENTO

3.1 Softwares Utilizados

No desenvolvimento de projetos é comum a utilização de softwares para a simulação, desenho do protótipo e o mais essencial para a programação. Sabendo disso sempre é válido descrever e listar os softwares utilizados.

3.1.1 Inventor Professional

O inventor é um software de CAD/CAM, isso é, é um software específico para desenhos em 3D, com ele o grupo conseguiu definir um esboço do agv, conforme pode ser visto na figura 2.

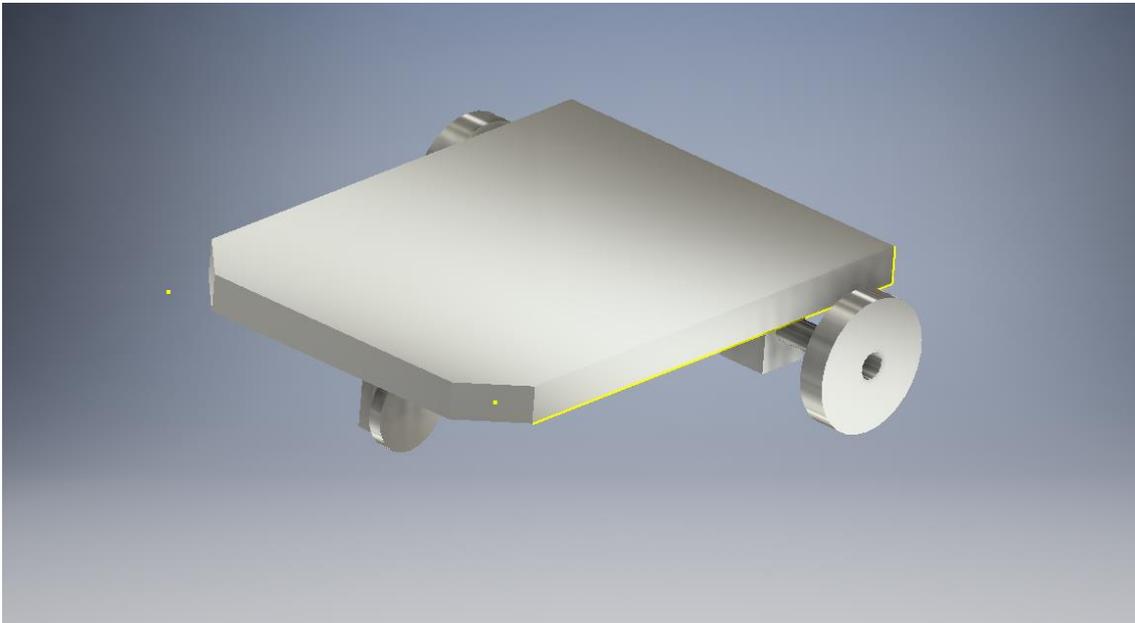


Figura 2 – Protótipo do AGV
fonte: Imagem Própria

3.1.2 Proteus Professional

O proteus é um software muito utilizado para a simulação de circuitos eletrônicos, nesse projeto ele foi muito utilizado principalmente pela capacidade de simular o microcontrolador com o código que estava sendo desenvolvido, na figura 3, pode-se ver a montagem realizada para simulação no software proteus.

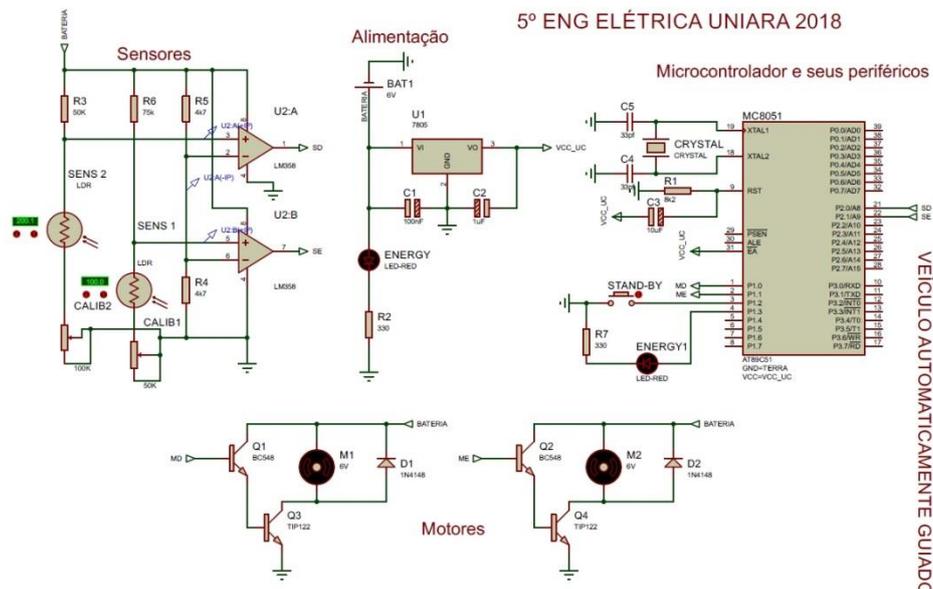


Figura 3 – Aspecto do software Proteus
 fonte: Imagem Própria

3.1.3 MCU 8051 IDE

O MCU 8051 IDE foi o software utilizado para a realização da programação do microcontrolador, ele foi escolhido pela sua interface simples e pela facilidade de identificar erros de códigos, na figura 4, pode-se ver a interface.

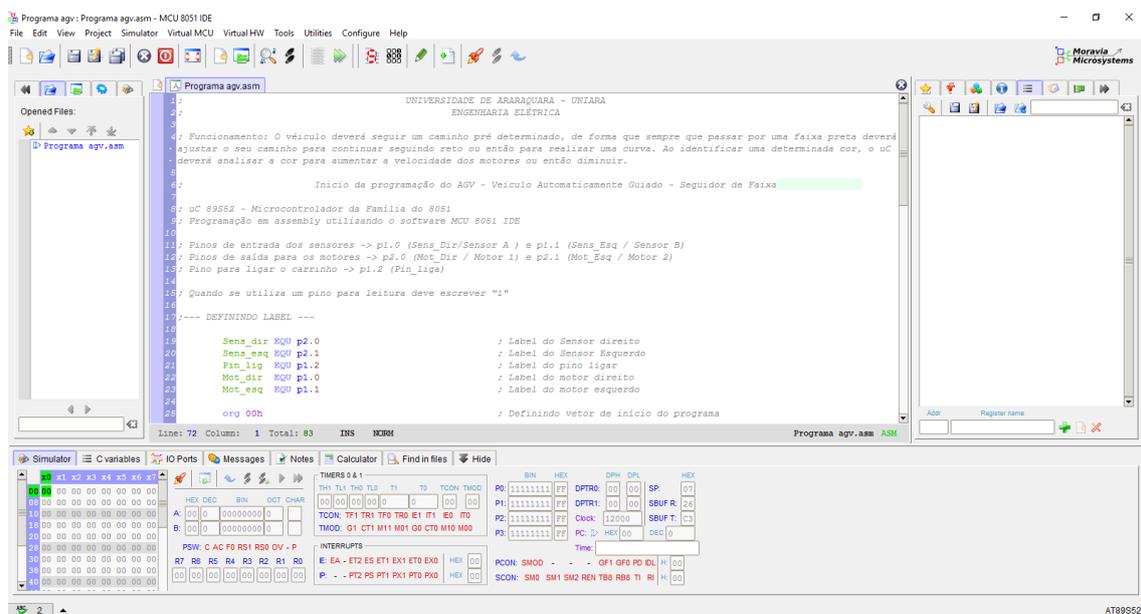


Figura 4 – Interface MCU 8051 IDE
 fonte: Imagem Própria

3.2 Desenvolvimento da programação

Utilizando o software MCU 8051 IDE, o grupo teve que desenvolver uma lógica de programação que fosse capaz de realizar a leitura de dois sensores, sensor direito e sensor esquerdo, e feita a leitura desses sensores realizar o controle de dois motores c.c (corrente contínua).

Basicamente o controle dos motores é dado pela lógica PWM (Pulse Width Modulation). O PWM é uma lógica que modula a largura de pulso de uma onda quadrada, fazendo com que possa se controlar a potência ou a velocidade, conforme a necessidade e essa lógica é baseada em relação ao tempo que eu quero a onda quadrada ligada e desligada, é dada pela fórmula 1.

$$Duty_{Cycle} = \frac{T_{on}}{T_{on}+T_{off}} * 100\% \quad (1)$$

Onde:

Duty_Cycle: significa o tempo em que a onda ficará em estado alto em Porcentos

Ton: Tempo total em estado alto

Tof: Tempo total em estado baixo

Sabendo que o projeto precisava de dois Duty Cycles, um para ele andar reto (50% de Duty) e outro para realizar as curvas (80% de Duty), a primeira etapa era determinar os valores de duty cycle, adotando um ciclo de 25 Hz que é equivalente a um período de 0,04 segundos ou 40 mS definiu-se o valor de Ton.

Para atingir o duty cycle de 50% seria necessário que ele ficasse 40 mS ligados e 40 mS desligados, sem a necessidade de realizar cálculo, enquanto que para atingir os 80%, utilizou-se a fórmula 1 para determinar o Toff.

$$80\% = \frac{40 * 10^{-3}}{40 * 10^{-3} + T_{off}} * 100\% \rightarrow T_{off} = 10 * 10^{-3}$$

Tendo como o resultado 10 mS. Assim tinha-se determinado todas as incógnitas para o PWM, faltando apenas calcular o valor para carregar os registradores do microcontrolador.

Para carregar os registradores de tempo do microcontrolador deve-se escolher o modo de operação e carregar seus registradores. O modo escolhido para esse processo foi o TMOD 1 (Modo de timer 1) e os valores carregados nos registradores TH0 e TL0 variavam de acordo com o tempo necessário para o pwm, os cálculos para os registradores estão sendo mostrados abaixo.

Ciclo de 100 Hz = 0,01 seg = 10 mS -> 65536 - 10000 = 55536 = D8F0h

Ciclo de 25 Hz = 0,04 seg = 40 mS -> 65536 - 40000 = 30536 = 63C0h

Sabendo os valores que deveriam ser carregados nos registradores, faltava apenas definir uma rotina que fizesse a leitura dos sensores da direita e da esquerda e então dependendo da combinação, entrar em uma rotina do pwm, isso é, caso os dois sensores estivessem sobre uma superfície branca os motores estariam ativos proporcionalmente, caso o sensor esquerdo estivesse sobre uma faixa preta o motor esquerdo deveria diminuir sua velocidade e o direito aumentar sua velocidade para que o curso fosse corrigido e ao contrário caso o sensor direito estivesse sobre a faixa.

Então finalizando a parte da programação, para simplificar, o programa possui 3 rotinas que se auto analisam e que servem para ajustar o caminho do veículo, sendo que a primeira rotina é destinada para o agv seguir reto com carregamento do pwm em 50% (40 mSeg ligados e 40 mSeg desligados) sendo assim os dois motores estão sincronizados em relação a sua velocidade, a segunda rotina é para realizar curvas para a direita carregando o pwm em 80% (40% mSeg Ligados e 10 mSeg% Desligados) isso é, para o motor esquerdo seria carregado com 10 mSeg enquanto que o motor direito com 40 mSeg, fazendo com que corrigisse o curso e assim vice e verso para curvas a esquerda.

Para demonstrar melhor, a figura 5 estará mostrando o formato de onda sobre os pinos dos motores, e assim analisar os modos dele seguindo reto e realizando uma curva.

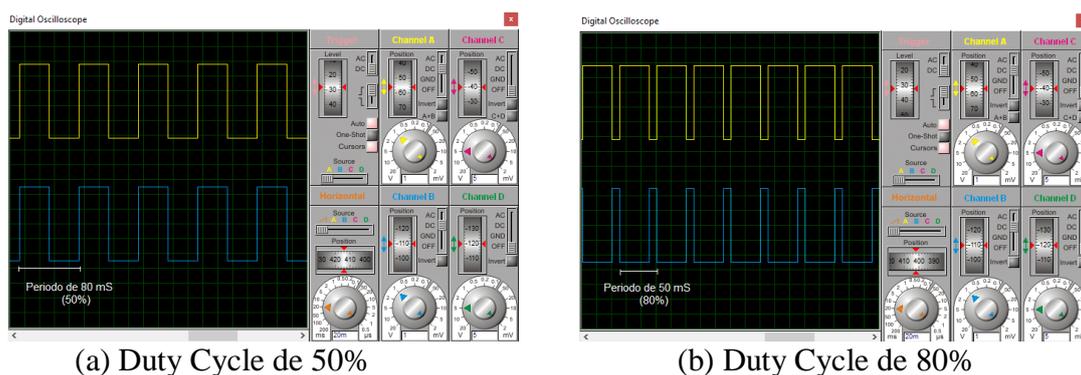


Figura 5 – Formatos de onda

OBS: O formato de onda de cor amarela é equivalente ao pino 1 da porta 0 enquanto que o azul pino 2 da porta 0, são referentes aos pinos dos motores.

Como pode se ver na figura 5 letra a) o motor esta configurado para andar reto, os dois pinos estão respondendo da mesma forma sendo meia onda ligada e meia onda desligada (40 mSeg cada um, totalizando um ciclo total em 80mS) enquanto que na b) o pino esta respondendo em uma frequência maior em relação ao pino 2 isso para que o agv ajuste sua rota nesse caso, o pino 1 fica em estado alto 40 mSeg e em baixo 10 mSeg e o pino 2 é o inverso, 10 mSeg em estado alto e 40 mSeg em estado baixo, totalizando um ciclo total de 50 mSeg.

No apêndice I, estará a programação completa comentada, para futuras análises e melhoramentos.

3.3 Desenvolvimento do sensor

O sensor nesse projeto é um dos pontos fundamentais para o funcionamento do carrinho, ele é o responsável por captar o ambiente externo e mandar informações para que o microcontrolador possa processar esses dados e então realize algo programado. No caso desse projeto, utilizamos dois sensores que são responsáveis por diferenciar a cor branca da preta através do quanto cada uma reflete da luz emitida sobre elas, afim do fotoresistor variar a resistência.

A lógica que foi desenvolvida utiliza um modelo de comparação de tensão, isso é, um circuito eletrônico foi desenvolvido utilizando LDR (Fotoresistor) em conjunto com um Led (Diodo emissor de luz) que através de um circuito divisor de tensão com o comparador de tensão, Lm 358, é capaz de enviar pulso alto ou baixo dependendo da cor que estiver sob ele.

Na figura 6, pode-se ver o conjunto do LDR com o Led. O Led é responsável por emitir uma luz e dependendo a cor da faixa que estiver sob ele o LDR irá variar sua resistência.

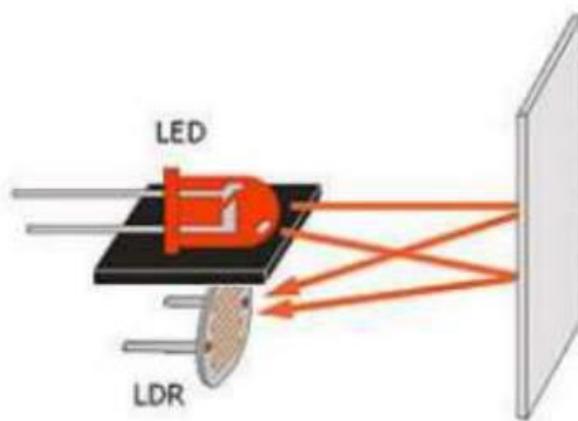


Figura 6 – Conjunto LED e LDR
fonte: Imagem Própria

A figura 6 demonstra como é emitido e captado o sinal que utilizamos, porém, ainda foi desenvolvido um circuito elétrico em conjunto com um comparador para garantir que apenas as cores brancas e pretas mudassem o estado lógico baixo e alto consecutivamente.

Para melhor entendimento, a figura 7 está demonstrada o circuito.

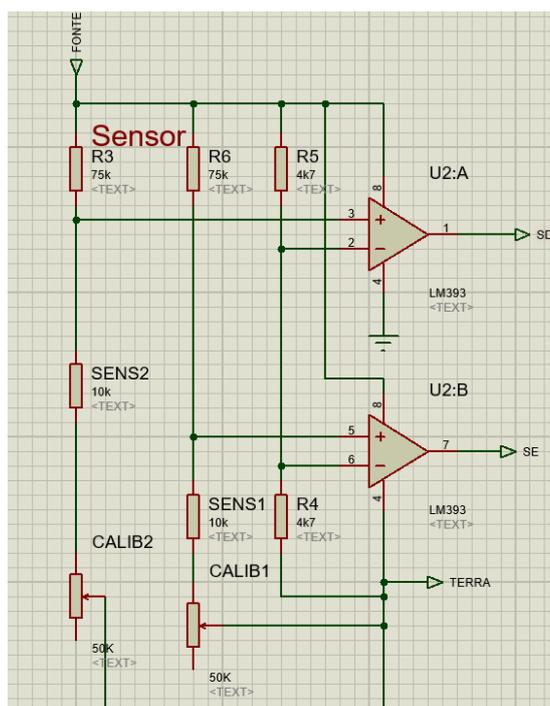


Figura 7 – Circuito divisor de tensão e comparador
fonte: Imagem Própria

Explicando a montagem do proteus, CALIB2 e CALIB1 são os LDR's e U2:A e U2:B são os comparadores lógicos. O comparador funciona como uma chave liga e desliga, sendo que a saída do comparador estará em estado alto quando o LDR estiver com sua resistência máxima, que ocorre quando o LDR está sobre uma superfície escura.

Para finalizar esse tópico e deixar o mais explicado possível, foram desenvolvidos dois sensores, que estão posicionados na frente do carrinho e estão conectados aos pinos 0 e 1 da porta 0 do microcontrolador e então dependendo da combinação lógica entre eles o carrinho andarà reto, realizarà curva para esquerda ou direita.

Tabela 2 – Combinações Lógicas

Combinação Lógica		Saída do Microcontrolador
P1.0	P1.1	
0	0	Reto
1	0	Curva para Esquerda
0	1	Curva para Direita

3.4 Desenvolvimento do hardware

Nessa parte será demonstrado o desenvolvimento do hardware do motor e das placas eletrônicas.

Os hardwares dos motores são acionados pelos pinos 0 e 1 da porta 2 do microcontrolador, e são chaveados pelo transistores BC548 e cada um possui um diodo de proteção, para não haver risco de corrente reversa sobre o transistor.

Seu circuito é relativamente simples, como é demonstrado na figura 8.

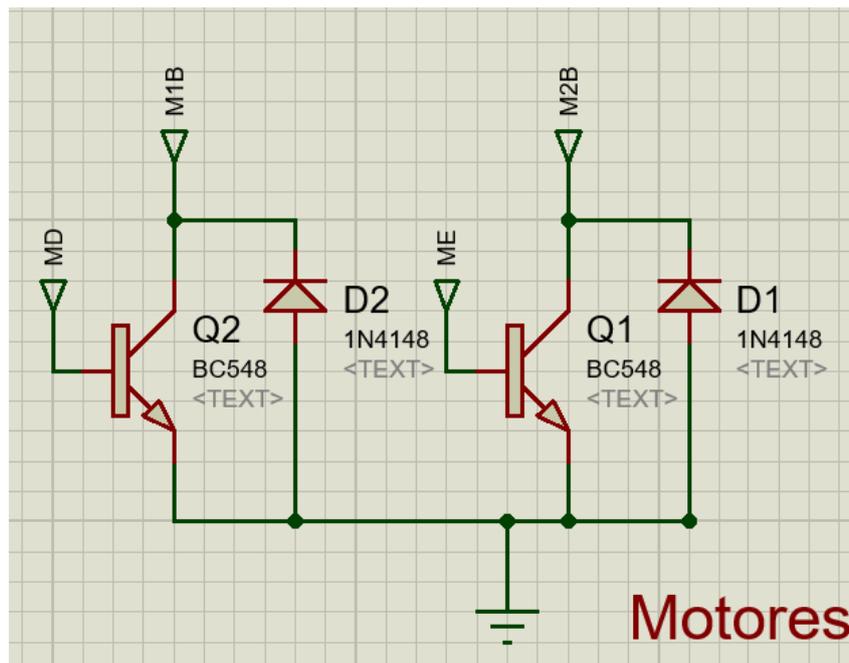


Figura 8 – Hardware do motor
fonte: Imagem Própria

A confecção da placa foi feita com o auxílio do Ares, programa de criação de PCB do Isis (Proteus), conforme pode ser visto na figura 9.

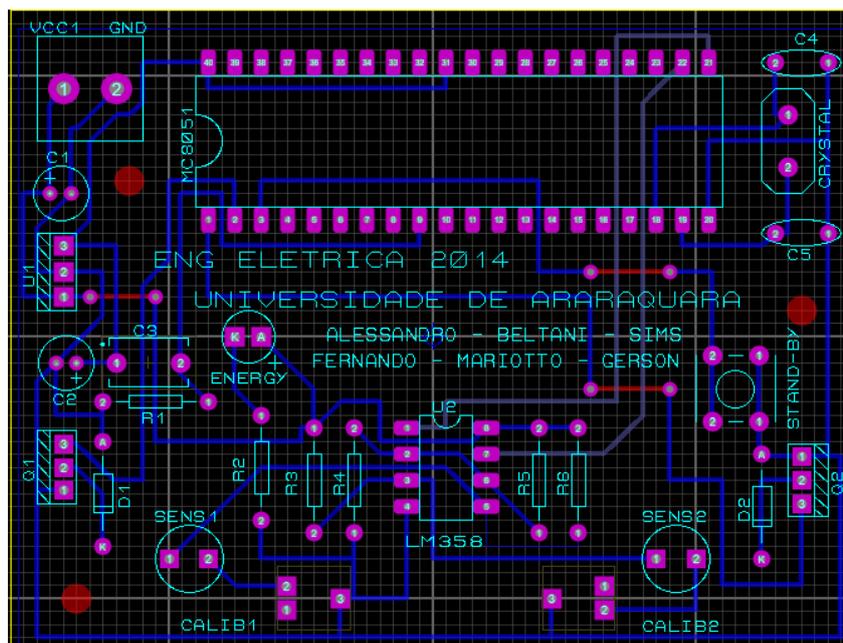


Figura 9 – Desenvolvimento da PCB
fonte: Imagem Própria

Ainda com o auxílio do Ares, foi possível gerar um esboço 3D da placa desenvolvida, como pode ser visto na figura 10.

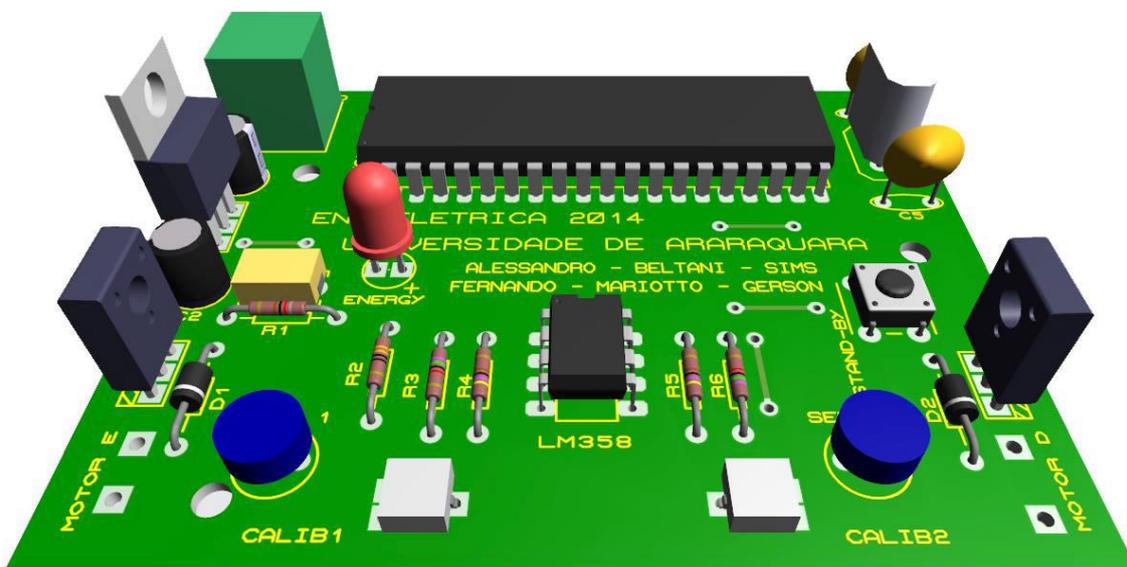


Figura 10 – Esboço 3D da placa eletrônica
fonte: Imagem Própria

Para a confecção da placa, foi utilizado a técnica de transferência térmica utilizando papel transparência.

Essa técnica basicamente é necessário imprimir o silk superior e inferior da PCB separadamente, para então passar na placa (fenolite) por transferência térmica, primeiramente é feita a transferência na parte inferior da fenolite onde ocorre a corrosão das trilhas, após isso é feita a transferência superior onde se encontra a descrição dos componentes.

Na figura 11 pode se ver a parte inferior da placa pronta com as trilhas já corroídas e a parte sendo feita a montagem da placa com os componentes.



(a) Parte inferior da placa

(b) Parte superior

Figura 11 – Resultados da placa

4 RESULTADOS OBTIDOS

Todos os resultados obtidos através do Proteus ficaram totalmente dentro do que foi calculado e responderam da forma como planejado.

Partindo para a parte prática, primeiramente montamos o circuito em uma protoboard e realizamos as medições e visualizamos o formato de ondas do PWM utilizando um osciloscópio, nessa etapa não houve nenhum problema, todos os resultados novamente bateram com o esperado, tanto o formato de onda do PWM dos motores quanto o circuito dos sensores, , nessa etapa não testamos o motor, conforme pode ser visto na figura 12.

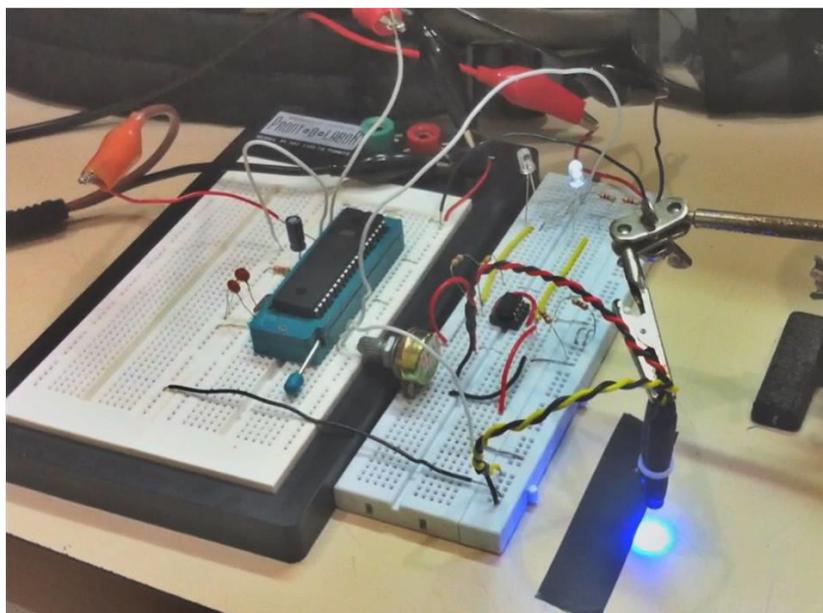


Figura 12 – Primeiros testes
fonte: Imagem Própria

Partindo para a parte prática com a placa e com os motores, os resultados não ocorreram conforme o planejado, porém, após um período de análise e testes conseguiu-se definir que o problema se encontra nos transistores que chaveiam os motores.

Esse problema estava acontecendo devido que a corrente fornecida pelo microprocessador não estava sendo capaz de chavear os transistores, porém, de forma precária e apenas como teste conseguimos forçar uma corrente de saturação na base do transistor fazendo com que o carrinho funcionasse, podendo assim ver onde se encontrava o problema e ver que tanto a parte lógica como todo o hardware da placa estavam corretos.

4.1 Solução final

Sabendo que o problema era o transistor inicial substituímos ele por um transistor do tipo Darlington NPN BC517 para condução da corrente do mesmo.

Pela corrente de saída do microcontrolador utilizado (Atmel AT89s52-24PU) ser muito baixa, na ordem de 50 micro amperes (50uA), justifica-se a utilização do darlington.

Também foi introduzido o uso do diodo de silício 1n4148 em paralelo com a carga para que quando o transistor esteja conduzindo, a corrente restante gerada pela indução da bobina do motor decresça mais rapidamente circulando pelo diodo, não inserindo corrente reversa no transistor e por assim não sobrecarrega-lo na condução.

Com as modificações descritas acima, conseguiu-se fazer os acionamentos dos motores, conseguindo dessa forma finalizar a ideia do projeto.

Na figura 13 temos a montagem final.

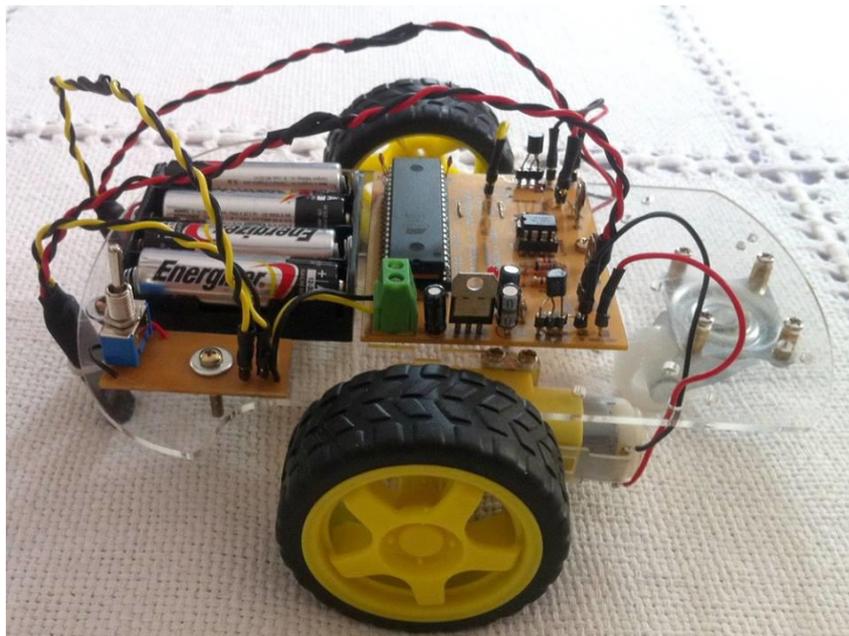


Figura 13 – Montagem final
fonte: Imagem Própria

5 CONCLUSÃO

Com o desenvolvimento conseguiu-se aplicar conhecimentos diversos, como o desenvolvimento de um circuito eletroeletrônico para aquisição de sinal dos sensores e acionamento de motores, a utilização de comparadores de tensão, confecção de placas eletrônicas, desenvolvimento de programação utilizando a linguagem assembly, além de evidenciar as pequenas diferenças da parte prática e teórica.

Referências Bibliográficas

Boylestad, A. R. Nashelsky L; Dispositivos eletrônicos e teoria de circuitos - 8 edi., 2013. p. 672.

Denys E. C. Nicolosi; Laboratorio de microcontroladores Família 8051 / 5° edição remodelada / São Paulo 2008- editora Erica LTDA.

Normas da ABNT, Citações e Referências Bibliográficas

Disponível em:<<http://www.leffa.pro.br/textos/abnt.htm>>. Acesso em: 24 de Outubro. 2018.

Apostila didática microcontroladores família MCS-51

Disponível em:<<http://pt.slideshare.net/danielrodriguesdesousa90/unicsul-aplicao-dosmicrocontroladores-no-dia-a-dia>> Acesso em: 24 de Outubro. 2018.