

Matheus de Melo Machado

Hardware integrado de baixo custo para automação veicular

Brasil

2016

Matheus de Melo Machado

Hardware integrado de baixo custo para automação veicular

Este exemplar corresponde à redação final da monografia da disciplina Trabalho de Conclusão de Curso II, devidamente corrigida e defendida por Matheus de Melo Machado e aprovada pela Banca Examinadora, como parte dos requisitos para a obtenção do título de Bacharel em Sistemas de Informação.

Universidade Federal de Mato Grosso do Sul

Campus do Pantanal

Curso de Sistemas de Informação

Orientador: Me. Murilo Oliveira Machado

Brasil

2016

Matheus de Melo Machado

Hardware integrado de baixo custo para automação veicular

Este exemplar corresponde à redação final da monografia da disciplina Trabalho de Conclusão de Curso II, devidamente corrigida e defendida por Matheus de Melo Machado e aprovada pela Banca Examinadora, como parte dos requisitos para a obtenção do título de Bacharel em Sistemas de Informação.

Trabalho aprovado. Brasil, 31 de Março de 2016:

Me. Murilo Oliveira Machado
Orientador

Me. Lucineide Rodrigues da Silva

Me. Anderson Pereira das Neves

Brasil
2016

Agradecimentos

É difícil agradecer todas as pessoas que de algum modo, nos momentos serenos e ou apreensivos, fizeram ou fazem parte da minha vida, por isso primeiramente agradeço à todos de coração.

Agradeço aos todos professores que desempenharam com dedicação as aulas ministradas, em particular o professor Felipe Jimenez pelo apoio nas etapas finais do curso.

Dedico este trabalho "in memoriam" de uma amiga, Camila Panovitch, ou Mila como era mais conhecida, esta pessoa que recebeu um estranho de braços abertos oferecendo todo amparo possível e um ombro amigo durante risos e lágrimas.

Agradeço aos meus amigos e companheiros Guilherme Alves, Rafaella Amanda, Kauê Murakami, Eliana Silva, José Cassimiro, Renner Antônio, Leonardo Scaloni, Lucas Mendes, Maria Clara Bizinotto, Isabela Silveira e Tarcisio Florencio que eu trouxe no coração de Sacramento-MG para Corumbá-MS, e sempre que podiam me ampararam em momentos difíceis mesmo a distância. Agradeço aos meus novos amigos e colegas que fiz durante minha jornada de 5 anos na cidade de Corumbá-MS, agradeço a Gustavo Alef, Alana Casagrande, Rayssa Maressa, Helene Miguéis, Cassiano Panovitch, Laerte Guerreiro, Ivan Claudino, Anne Caroline, Suzy Paiva, Karina Caetano e Stephine Okimasu pessoas que marcaram minha vida nesta trajetória de graduação que levarei para o resto da vida comigo em minhas lembranças. Agradeço em particular a família de Divaldo Fonseca e Ronilce Fonseca pelo amparo em vários momentos difíceis e também felizes durante esta jornada tendo eles como uma segunda família.

Por fim agradeço aos meus pais, Pedro Jasmar e Helena Beatriz, pela determinação e luta na minha formação e dos meus irmãos Vinicius Machado, Daniel Machado e Murilo Machado que por coincidência do destino meu professor e orientador desta monografia, fazendo amparar os ensinamentos a mim passados.

Resumo

Esta monografia apresenta a implementação de uma plataforma de controle, utilizando módulo bluetooth e sensores ultra-som, chuva e buzzer. O objetivo principal do trabalho, é adicionar um hardware de tecnologia Open Source, apto a realizar a leitura e controle das funções básicas de um automóvel de passeio, utilizando a ferramenta microcontrolador arduino. Por fim, com este trabalho é possível a criação de protótipos de atuais tecnologias que possam ser acopladas a conduções populares, como sistema de apoio ao estacionamento, automatização do para-brisa e um produto final para manuseio de algumas funções via smartphone.

Abstract

The description of the project is the implementation of a control platform using bluetooth module and ultrasound sensors, rain and buzzer. The main objective of the work, will add a technology hardware Open Source, able to perform reading and control of basic functions of a passenger car used the tool microcontroller Arduino. At the end of this work, it is expected to have the ability to create prototypes of current technologies that can be attached to popular conveyances, such as supporting the parking system, automation of the windshield and a final product for handling some functions via smartphone.

Keywords: arduino. autonomos. hardware. microcontrolador. sistema embarcado. sistema ativo. sistema passivo.

Lista de ilustrações

Figura 1 – Arduino Duemilanove ATmega328; 1- Plug P4, 2- Porta VIN	19
Figura 2 – Sensor de Chuva YL-83 e Placa de Controle	20
Figura 3 – Sensor Ultrassonico HC-SR04	21
Figura 4 – Buzzer Ativo 5V Bip Contínuo	22
Figura 5 – Módulo Bluetooth RS232 HC-05	22
Figura 6 – Carrinho Arduino, Carro Robô 80RPM	23
Figura 7 – IDE Arduino versão 1.6.4; 1-Verificar, compilar, corte, colar e busca; 2-Serial Monitor; 3-Editor de código; 4-Área de feedback	24
Figura 8 – IDE do programa PLX-DAQ	25
Figura 9 – Aplicativo ATC Lite; 1- Connect, 2- Show All, 3- Disconnect ou Connect; 4- Circuito “on” e “off”	25
Figura 10 – Protótipo do sensor de chuva YL-83 utilizando os leds como simulador de velocidade do temporizador do para-brisa; 1-potenciômetro	28
Figura 11 – Funcionamento do sensor ultrassônico com um obstáculo no caminho	29
Figura 12 – Protótipo do sensor de distancia	29
Figura 13 – Protótipo do sensor de distancia, 1- sensor, 2- Arduino, 3- Baterias	30
Figura 14 – Esquema da configuração de bluetooth RS232 HC-05 em modo mestre	30
Figura 15 – Diagrama da Conexão do sensor de chuva; 1- Arduíno Duemilanove, 2- Módulo, 3- Placa do Sensor, 4- Chave de Ignição, 5- Módulo de Comando	31
Figura 16 – Diagrama da Conexão do sensor de estacionamento, 1- Arduino Duemi- lanove, 2- Buzzer, 3- Sensor ultrassônico, 4- Chave de ignição	32
Figura 17 – Protótipo enviado a empresa, 1- Arduíno Duemilanove, 2- Módulo Bluetooth, 3- Chave de ignição, 4- Relê, 5- Buzina, 6- Farol, 7- Trava, 8- Vidros	33
Figura 18 – Configuração entre código e aplicativo ATC Lite	33
Figura 19 – Gráfico I-Leitura do Sensor durante o dia 09/04/2016; Gráfico II Taxa de variação do volume de chuva, Gráfico III- Taxa de Variação da velocidade do volume de chuva; Gráfico IV- Mudança de velocidade do para-brisa de acordo com os parâmetros	35
Figura 20 – Gráfico I-Leitura do Sensor durante o dia 10/04/2016; Gráfico II Taxa de variação do volume de chuva, Gráfico III- Taxa de Variação da velocidade do volume de chuva; Gráfico IV- Mudança de velocidade do para-brisa de acordo com os parâmetros	37

Figura 21 – Medidas da distância entre o ultrassom e um obstáculo em movimento, calculo da velocidade e aceleração do obstáculo, e simulações da velocidade de alimentação do atuador de resposta ao obstáculo. A figura mostra respectivamente Gráfico I- Distancia em em função do Tempo, Gráfico II- Velocidade em função do Tempo, Gráfico III- Aceleração em função do tempo, Gráfico IV- Velocidade em função do Tempo com parâmetros	38
Figura 22 – Medidas da distância entre o ultrassom e um obstáculo em movimento, calculo da velocidade e aceleração do obstáculo, e simulações da velocidade de alimentação do atuador de resposta ao obstáculo. A figura mostra respectivamente Gráfico I- Distancia em em função do Tempo, Gráfico II- Velocidade em função do Tempo, Gráfico III- Aceleração em função do tempo, Gráfico IV- Velocidade em função do Tempo com parâmetros	39
Figura 23 – Reconhecimento do bluetooth por parte do aplicativo	40
Figura 24 – Imagem do produto finalizado	40
Figura 25 – Legenda do chicote de instalação	41
Figura 26 – O diagrama final elaborado de acordo com a instalação, nomenclatura das cores na figura 25; 1-Trava, 2-Farol, 3-Relê, 4-Buzina, 5-Vidros . . .	42
Figura 27 – Placa ARM Cortex	42
Figura 28 – Layout do aplicativo modificado	43
Figura 29 – Condições do tempo na data 09-04-2016	67
Figura 30 – Informações sobre o clima no estado do Mato Grosso do sul	68
Figura 31 – Gráfico I-Preciptação de chuva, Gráfico II- Temperatura, Gráfico III- Umidade Relativa, Gráfico IV- Vento, Gráfico V-Pressão do nivel do mar, Gráfico VI-Nebulosidade	69
Figura 32 – Condições do tempo na data 09-04-2016	70
Figura 33 – Informações sobre o clima no estado do Mato Grosso do sul	70
Figura 34 – Gráfico I-Preciptação de chuva, Gráfico II- Temperatura, Gráfico III- Umidade Relativa, Gráfico IV- Vento, Gráfico V-Pressão do nivel do mar, Gráfico VI-Nebulosidade	71

Sumário

1	INTRODUÇÃO	10
	Introdução	10
1.1	Objetivos	10
1.1.1	Objetivos Gerais	11
1.1.2	Objetivos Específicos	11
1.2	Justificativa	11
2	ESTADO DA ARTE	12
2.1	Desenvolvimento na área veicular	13
2.2	Tipos de tecnologias e suas aplicabilidades	14
3	TRABALHOS RELACIONADOS	16
4	MATERIAIS E FERRAMENTAS	18
4.1	Materiais	18
4.1.1	Arduíno Hardware	18
4.1.2	Sensor de chuva YL-83	20
4.1.3	Sensor ultrassonico HC-SR04	21
4.1.4	Buzzer 5V	21
4.1.5	Módulo bluetooth RS232 HC-05	22
4.1.6	Carro robô 80RPM	23
4.2	Ferramentas Computacionais	23
4.2.1	Arduino software 1.6.4	24
4.2.2	Excel e PLX-DAQ	24
4.2.3	Aplicativo ATC Lite	25
5	METODOLOGIA	27
5.1	Implementações dos materiais	27
5.1.1	Sensor de chuva YL-83	27
5.1.2	Sensor ultrassônico HC-SR04	28
5.1.3	Configuração bluetooth	30
5.2	Integração dos materiais no veículo real	31
5.2.1	Sensor de chuva	31
5.2.2	Sensor de estacionamento	32
5.2.3	Sistema de portas e alerta	32

6	RESULTADOS	34
6.1	Sensor autônomo de chuva	34
6.2	Sensor de apoio ao estacionamento	37
6.3	Hardware integrado automotivo	39
6.3.1	Configuração do módulo bluetooth	39
6.3.2	Hardware integrado automotivo	40
6.4	Aplicativo modificado	42
6.5	Mercado para o produto	44
7	LIMITAÇÕES DO TRABALHO	46
8	CONCLUSÃO	47
9	CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS	48
9.1	Considerações Finais	48
9.2	Trabalhos futuros	48
	REFERÊNCIAS	49
	APÊNDICES	52
	APÊNDICE A – APENDICE	53
A.1	Protótipo Funções	53
	ANEXOS	56
	ANEXO A – ANEXOS	57
A.1	Sensor de Chuva	57
A.2	Sensor Ultrassônico	58
A.3	Configuração Bluetooth	60
A.4	Comandos da Configuração Bluetooth	61
A.5	Ascender Led	62
A.6	Links Youtube	62
A.7	Leitura do sensor de chuva pelo PLX-DAQ e Excel	63
A.8	Leitura do sensor de chuva pelo PLX-DAQ e Excel	63
A.9	Dados do CPTEC/INPE Dia 09/04/2016	67
A.10	Dados do CPTEC/INPE Dia 10/04/2016	70

1 Introdução

Há décadas, os sistemas embarcados ¹ têm sido aproveitados em distintas áreas da ciência, possibilitando várias aplicações, tendo a competência de cumprir trabalho para os quais foram criados. Recentemente, o relacionamento entre computadores e a indústria de veículos, tem oferecido aos condutores e passageiros distintos benefícios, um exemplo importante é a segurança obtida com os limitadores e controladores de velocidade (ANJOS, 2011).

Os sistemas embarcados têm ganhado espaço, ampliando aplicações ao redor do mundo, fazendo com que os aparelhos móveis ganhem destaque pela sua facilidade de uso essa relação vem permitindo a criação de novos dispositivos cada vez mais sofisticados (BARROS, 2012). Juntamente com a geração dos computadores portáteis como os smartphones vêm abrindo portas para projetos com foco em desenvolver sistemas embarcados (BARROS, 2012).

Apontando este aumento dos engenhos automotivos e do desenvolvimento de sistemas eletrônicos para aplicações veiculares, as pesquisas para o controle dos sistemas veiculares tem se tornado um tema interessante. Os protótipos apresentados nesta monografia se diferenciam das demais aplicações já desenvolvidas, pois oferece um sistema de baixo custo, com possibilidades de acrescentar novas funções que não foram previstas inicialmente.

Após esta breve introdução serão enunciados os objetivos e justificativa da monografia, descrito o estado da arte em que se encontra o tema abordado, e a revisão bibliográfica dos trabalhos que mais se assemelham com este trabalho, em seguida serão apresentados os materiais e a metodologia utilizados no desenvolvimento do protótipo, e os resultados obtidos, por fim, a conclusão e as considerações finais

1.1 Objetivos

Este trabalho descreve um sistema embarcado, capaz de auxiliar condutores de veículos desprovidos de tecnologias recentes. Foi utilizando um hardware Open Source para implementar sensores, e controlar alguns atuadores contidos no carro, possibilitando assim a construção de uma base de controle do sistema elétrico do veículo.

¹ É um sistema microprocessado que é completamente encapsulado ou dedicado ao dispositivo ou sistema que ele controla (ANJOS, 2011)

1.1.1 Objetivos Gerais

O principal objetivo foi desenvolver uma plataforma integrada para automóveis desprovidos das recentes tecnologias de automação, capaz de acionar automaticamente (sistema ativo) o para-brisa mediante chuvas e auxiliar o condutor ao estacionar o carro (sistema passivo), e proporcionando uma interação via celular com as travas, luzes, vidros e buzina.

1.1.2 Objetivos Específicos

Aqui são apresentados os objetivos específicos do trabalho, quais funções serão controladas, os protótipos construídos, mostrar a limitação de algum dispositivo utilizado no desenvolvimento do projeto.

- Controlar as funções do veículo como: buzina, farol, vidros e trava através do bluetooth
- Construir um protótipo de apoio ao estacionamento, utilizando o sensor ultrassônico para mensurar a distância de um obstáculo, e ter um buzzer como alerta.
- Verificar as limitações do dispositivo de apoio ao estacionamento.
- Construir um protótipo do sensor autônomo para o para-brisa, utilizando um sensor de chuva e mensurando suas limitações.
- Construir um protótipo que contemple as funções do veículo de buzina, vidros, trava e faróis.
- Fazer uma pesquisa dos produtos disponíveis no mercado que são similares aos protótipos desenvolvidos neste trabalho para uma avaliação de custo e benefício.

1.2 Justificativa

Devido ao aumento do consumo de dispositivos eletrônicos focados na indústria automobilística (CONSONI, 2004) juntamente com a quantidade de pesquisas abrangendo este tema, que mostram as melhorias proporcionadas por tecnologias desta natureza, principalmente no quesito segurança, conforto, e praticidade (ANJOS, 2011) justifica-se o trabalho desenvolvido por esta monografia, pois vem despertando um grande interesse das montadoras e laboratórios de desenvolvimento e pesquisa.

2 Estado da Arte

Sistemas de automação surgiram na Grécia antiga com o aparecimento dos primeiros autômatos¹, que serviam inicialmente como brinquedos, ícones religiosos ou como ferramentas para demonstrar princípios científicos e são considerados ancestrais dos atuais robôs (DUDEK; JENKIN, 2000). Numerosos autômatos surgiram como a construção do primeiro aparelho de calcular inventado pelo matemático Pascal em 1642², mas foi apenas a partir de 1923 que a palavra “robot” começou a ser usada, sendo aplicada pela primeira vez por Capek Dudek (KELBER C. R. JUNG, 2005).

A inserção da eletrônica nos automóveis revolucionou a aplicabilidade de muitos dispositivos, principalmente do câmbio e da injeção eletrônica, proporcionando melhorias significantes no desempenho, conforto e confiabilidade do carro (ANJOS, 2011). Desde então cresceu o número de pesquisas focadas na área veicular contribuindo pelo surgimento de sistemas capazes de oferecer melhores assistências para os motoristas, como: monitorar a temperatura do veículo, a quantidade de combustível, sistemas de abandono de pista e sinalização de luminosidade (BARROS, 2012).

Pesquisas voltadas para a área veicular surgiram e tiveram apoio de diversas agências dos governos mundiais, destaca-se a Agência de Projetos de Pesquisa Avançada conhecida como DARPA, que foi uma criação americana logo após o lançamento do satélite sputnik³, que tinha como objetivo superar a tecnologia dos principais centros de pesquisa do mundo (SILVA, 2014).

Em 2005, houve um desafio realizado pela DARPA, agência comandada pelas forças aéreas americanas e pelo pentágono, eles organizam eventos e incentivam o desenvolvimento de veículos autônomos, oferecendo 2 milhões de dólares para o vencedor. Algumas universidades participaram como: *Stanford University* e *Carnegie Mellon University* (THRUN, 2006).

O veículo vencedor do evento foi desenvolvido em uma parceria entre algumas empresas, a *Volkswagen of America*, *Mohr Davidow Ventures* e *Intel Research*. Este automóvel possui um sistema embarcado conectado a uma rede de sensores que “scaneavam” o percurso por 30 metros à frente montando o mapa do local, as informações então eram combinadas com as imagens do sistema de visão procurando padrões similares ao chão com uma passagem segura, fazendo com que o campo de “visão” do veículo aumentasse em

¹ Autômatos são modelos abstratos de máquinas que executam cálculos de uma entrada movendo através de uma série de estados ou configurações (AZIZ, 2005)

² Uma máquina que realiza as quatro operações matemáticas básicas, conseguia fazer diretamente operações de adição e subtração, as operações de multiplicação e divisão eram feitas por repetições.

³ Primeira missão do Programa Sputnik, que enviou o primeiro satélite artificial da Terra realizado pela URSS

até 80 metros à frente garantindo segurança para acelerar, caso não conseguisse “enxergar” um percurso por onde pudesse realizar o trajeto, o veículo diminuía sua velocidade e os sensores de distância procuravam outro caminho seguro em um raio mais curto, tinha atuadores que são equipamentos capazes de converter energia hidráulica em movimento, este veículo foi batizado de Stanley. Com o desenvolvimento deste projeto a tecnologia nesta área foi expandida e ajudou a aperfeiçoar os sistemas eletromecânicos (SABBAGH, 2009).

Ainda no desafio da DARPA o veículo Stanley, foi modificado para *Drive-by-Wire*⁴ contribuindo para os condutores que portam alguma deficiência física. Foi integrado ao automóvel um GPS e um IMU⁵, permitindo a correção da posição do veículo a partir de dados locais que eram obtidos (SABBAGH, 2009). O hardware utilizado integra sensores e atuadores, juntamente com um algoritmo de síntese sensorial que planeja o caminho e calcula a melhor opção em torno dos obstáculos (COSTA, 2013).

Com o aparecimento de novas tecnologias, tornou-se possível o desenvolvimento de sistemas cada vez mais seguros. A abundância de sistemas embarcados nos veículos, proporcionaram a construção de dispositivos como: air bags, freio ABS, teto solar, controle de tração, alarme entre outros (BARROS, 2012).

Já circulam notícias de que a Ford vai fazer o lançamento no Brasil da terceira geração do sistema de conectividade SYNC⁶, trazendo funcionalidades dos smartphones para dentro dos carros com uma performance altamente intuitiva, o SYNC 3 vai permitir que o usuário acesse por comando de voz aplicativos de uso do dia a dia, como: entretenimento e navegação (LAB, 2016).

Este Capítulo apresenta o desenvolvimento na área veicular, a divisão das tecnologias mostrando a diferença entre sistema de apoio ao motorista e o sistemas veiculares robóticos e como essas tecnologias foram aplicadas na automação veicular.

2.1 Desenvolvimento na área veicular

As tecnologias desenvolvidas nesta área têm o objetivo de auxiliar o motorista no cumprimento de determinadas tarefas, melhorando o desempenho do usuário e a segurança. O *Hands-free Vehicle Driving*⁷, permite utilizar o celular no veículo sem as mãos, com isso se estuda um sistema para abranger todos os controles parciais do automóvel. O sistema chamado de AHS (Sistema de Estrada Automatizada)(CHEON, 1997), tem sido

⁴ Sistemas drive-by-wire permitem o comando eletrônico dos diversos atuadores de um veículo por meio de um sistema de software e uma rede de comunicação.

⁵ Uma unidade de medição inercial, é um dispositivo eletrônico que mede e relata força específica do corpo

⁶ Sistema de navegação dos veículos da Ford

⁷ Sistemas que deixa as mãos livres para conduzir o veículo

aplicado em países do continente europeu, como: a Alemanha, Itália e França onde existem rodovias com um alto grau de automação operando de forma inteligente, estas rodovias são equipadas com diversos sensores, como: câmeras e radares trabalhando conectados e auxiliando o condutor no controle do veículo, fazendo com que se estabeleça o controle de velocidade eletronicamente, desta forma prevenindo o motorista nas condições de emergência(MACEDO, 2015).

O foco das pesquisas desta área de veículos inteligentes, revolucionaram os conceitos no que se entende de transporte de pessoas (BARROS, 2012). As linhas de pesquisas surgem nas mais diferentes e possíveis áreas como: segurança, meio ambiente, conforto, e adaptabilidade voltadas para deficientes.(GALILEU, 2014)

2.2 Tipos de tecnologias e suas aplicabilidades

Os desenvolvimentos atuais buscam fundir as soluções mecânicas e elétricas por soluções eletromecânicas(KELBER, 2003). Existem diferentes sistemas de automação veicular apontando os objetivos e as técnicas aplicadas, o sistema de automação veicular é dividido em duas partes, os sistemas veiculares robóticos que tem como características a sua mobilidade semi-autônomos ⁸ e os sistemas completamente autônomos ⁹, e os sistemas de apoio ao motorista que tem o foco principal o mercado automotivo, proporcionando conforto e segurança ao condutor.

Recentemente surgiram robôs móveis semi-autônomos e autônomos aplicados a exploração de ambientes hostis, e com decorrer do tempo são inseridos em automóveis de passeios(BARROS, 2012). Anualmente acontece o evento *Grand DARPA Challenge Autonomous Ground Vehicles* ¹⁰, com finalidade de gerar percursos autônomos de carros de passeio. (CORREIA, 2007). Outros projetos envolvem a autonomia de empilhadeiras e carregadeiras como os desenvolvidos pelas empresas, FOX GmbH e Goetting KG (BARROS, 2012).

O sistema de apoio ao motorista se classifica em sistemas passivos e sistemas ativos. No sistema passivo, a eletrônica auxilia no alerta ao condutor com o objetivo de contornar a situações, no caso do sistema ativo, as funções eletromecânicas agem em situações críticas como um “co-piloto eletrônico” (BARROS, 2012).

Segundo (BARROS, 2012) um sistema ativo é focado na segurança dos passageiros e motoristas, um exemplo é alertando quando o veículo abandona a pista. Um sistema passivo muito aproveitado hoje em dia é o sistema de GPS *Global Positioning System* juntamente com mapas digitais das regiões que permitem sugestões na navegação do

⁸ É a opção que junta o controle manual com a opção automática

⁹ É aquele que age de forma pré-estabelecida para determinado objetivo quando solicitado

¹⁰ Grande Desafio DARPA de veículos autônomos

motorista, construindo rotas a serem cursadas.

Estes sistemas estão sendo acoplados a sistemas mais robustos, permitindo ao motorista obter mais informações sobre condições externas do carro como: clima, tráfego de veículos, obstáculos e estradas, e as condições internas do automóvel como: nível de combustível, situação elétrica e temperatura ambiente (KELBER, 2003).

Quando um veículo aciona o modo autônomo, os olhos do condutor são “trocados” por câmeras e lasers que tem o objetivo de averiguar o espaço em 360 graus, um exemplo é o sistema usado no *Google Street View* (OLIVEIRA, 2013). Além da automação veicular proporcionar ao condutor e passageiros mais segurança em circunstâncias críticas, podem automatizar algumas tarefas do dia a dia como realizar baliza.

A indústria automotiva tem investido muito nesta área, visando principalmente acrescentar mais segurança, confiabilidade, conforto. Este investimento proporcionaram a frenagem ABS *Anti-lock-Breaking-System*, estabilidade do carro ESP *Eletronic Stability Program*, controle da autonomia, controle da velocidade, controle de temperatura dentro do veículo e controle do combustível (CORREIA, 2007).

Em 1997, o *Institut National de Recherche en Informatique et en Automatique* (I.N.R.I.A)¹¹, construiu um automóvel denominado de CyCab, com diversos estilos de movimentação automatizados como a execução do planejamento e movimento em meio a obstáculos utilizando o programa Scilab que foi uma criação do próprio instituto (BARROS, 2012). O CyCab já está sendo criado pela agência Robosoft que oferece outros exemplares, como: RoBUCAB que é um modelo de transporte de médio porte no máximo 4 pessoas que não tem motorista, o RobuRIDE um modelo de porte grande parecido com um vagão de trem e não necessita de condutor e o RoBUCAR um veículo de pequeno porte que é usado para andar em ambientes hostis. (C.A, 2003).

¹¹ É uma organização pública francesa de carácter científico e tecnológico criada em janeiro de 1967. O seu objetivo é de reunir pesquisadores e incentivar a pesquisa nas áreas de informática e automação

3 Trabalhos Relacionados

Este Capítulo, foi elaborado a partir de uma revisão bibliográfica dos trabalhos com temas comuns ao desenvolvido desta monografia. Serão apresentados pontos importantes, como: a utilização de Arduino para sistema de automação veicular, evolução dos sistemas embarcados na indústria automobilística brasileira, uma plataforma robótica controlada remotamente utilizando Arduino, automação veicular utilizando interação entre o Arduino e o sistema android.

A monografia de Sistema de Automação Veicular com Arduino, publicada em 2012 pelo Centro Universitário Adventista de São Paulo Engenheiro Coelho, desenvolvida por (BARROS, 2012), tem como objetivo principal fazer a elaboração de um regulamentador de automação para automóveis acionando as funções como : manter a velocidade do veículo, manter a distância do veículo a sua frente, controle do capô do veículo e o monitoramento da temperatura no interior do veículo de acordo com as necessidades do condutor, sendo implementado um controlador de sistema embarcado, que possa ser configurável para o carro a receber comandos, para isso foram utilizados um hardware e software open source (Arduino e Android) juntamente com um smartphone.

O artigo escrito por (ARROYO; RUFINO, 2015) intitulado como Automação Veicular através do elo entre Arduino e Sistema Android, publicado no congresso SEINPAR 2015, apresenta uma interação entre os mesmos. Este trabalho apresenta um protótipo integrando duas tecnologias open source Arduino e Android, no qual a comunicação pelo shield bluetooth(RL232Ttl) possibilita ativar dispositivos do veículo, como: farol, lanterna traseira, lanterna de teto, trava elétrica, motor de partida, vidro elétrico além de fornecer informações sobre o ventilador de arrefecimento e bomba elétrica de combustível. O protótipo foi construído com um Arduino Leonardo pois tem conexão micro usb, foi utilizado um smartphone contendo o sistema operacional Android, alguns fusíveis para a proteção do sistema e uma bateria de 12 volts.

O artigo de (KELBER, 2003), intitulado Tecnologia para Automação Veicular, soluções em mecatrônica e sistemas de apoio ao motorista foi publicado na Feira de Telecomunicações e Tecnologia da Informação(TELEXPO) no ano de 2003, apresenta uma visão ampla das novas tecnologias em desenvolvimento e implementadas na área da automação veicular. O projeto desenvolvido por um grupo de pesquisadores de veículos autônomos da UNISINOS, apresenta a importância do sistema de apoio ao condutor, com base nas informações internas e externas do automóvel através de sensores específicos, como: chuva, leitores de faixa e obstáculos fazendo com que tenha uma troca de dados entre o veículo e a sua parte externa de forma similar a monografia aqui apresentada oferecendo

apoio ao motorista. Utilizando um sensor ultrassônico que o auxilia no estacionamento e um sensor para detectar chuva dando autonomia para o limpador para brisa.

A monografia intitulada como *Evolução Eletrônica Embarcada na Indústria Automobilística Brasileira*, escrito por (ANJOS, 2011) e publicado pela Escola de Engenharia de Mauá no ano de 2011, apresenta as atuais mudanças ocorridas na área de automação veicular. Este trabalho realiza um estudo técnico e teórico, com foco nas partes elétricas e peças automotivas sobre a evolução dos componentes, como: travas elétricas, alarmes e vidros elétricos. Este projeto ofereceu orientações para realizar medições na parte elétrica do carro, fornecendo informações sobre voltagens de peças como a bateria e ignição.

A monografia de (SMIDT, 2013), intitulada *Implementação de uma Plataforma Robótica Controlada Remotamente* utiliza o Arduíno publicado pela Universidade de São Paulo no Campus de São Carlos, que faz implementação usando o Arduíno software e hardware com o foco no baixo custo do produto. A plataforma realiza comunicação via bluetooth, para ordenar os movimentos do robô móvel de porte pequeno que será controlado remotamente com o emprego do microcontrolador ATMega328. Este trabalho teve utilidade de agregar um valor teórico, técnico e científico para o conhecimento em diversas tecnologias como: demonstrando que a robótica é uma das melhores maneiras de se introduzir o conhecimento de uma forma prática funcional e totalmente didática, e os conhecimentos teóricos aplicados para a escolha do sensor ultrassônico e módulo bluetooth.

4 Materiais e Ferramentas

Este Capítulo mostra os recursos disponíveis e compatíveis com o desenvolvimento desta monografia. Abaixo são descritas as características de microcontroladores, sensores e softwares que possibilitam a construção de sistemas passivos e ativos, bem como os materiais e ferramentas computacionais.

4.1 Materiais

Com prioridades de materiais com baixo custo e compatíveis aos objetivos da monografia. São apresentados quatro modelos de Arduíno, três sensores capazes de mensurar a quantidade de chuva e três para medir distância de obstáculos, três módulo bluetooth capazes de realizar interação via ondas de frequência.

4.1.1 Arduíno Hardware

O Arduino é uma plataforma de prototipagem contendo um microcontrolador reprogramável, inserido no contexto open source. Esta plataforma é utilizado em diversos projetos acadêmicos, empresariais e informais, contribuindo pela vasta literatura disponível na web e nas bibliotecas, colaborando de forma significativa para esta monografia.

Atualmente existem no mercado vários modelos de placas Arduíno, onde os mais comuns são o Uno, Duemilanove, Leonardo e o Mega. Cada um deles possui uma característica particular, onde a escolha depende da necessidade que o projeto requer.

O Arduíno Mega é adequado para projetos que requerem muitas portas, pois este contém quinze portas analógicas¹ e cinquenta e quatro digitais², pode ser alimentado entre 7V e 12V pela entrada P4, por 5V pelo cabo USB ou pela porta VIN (Voltagem de entrada), podendo alimentar o circuito com duas portas de 5V, uma de 3.3V, e três GND. Este Arduíno contém um microcontrolador Atmega 2560, e portando uma memória Ram de 8 KB e uma Flash 256 KB, possibilitando um processamento de 16 Mhz³.

O Arduíno Leonardo é apropriado para projetos de médio porte, pois este contém doze portas analógicas e vinte digitais, pode ser alimentar entre 7V e 12V pela entrada externa P2, ou por 5V via cabo microUSB e uma porta VIN, e pode alimentar o circuito

¹ Portas analógicas fazem leitura de sinais que carrega uma informação numa variação continua de tensão

² As portas digitais só podem assumir dois estados, HIGH e LOW, ou seja, 0 V ou 5 V. Dessa forma só é possível ler apenas dois estados. Por exemplo, verificar se uma porta está aberta ou fechada, identificar se um botão está pressionado ou solto.

³ O clock é a frequência com que o processador consegue executar as tarefas, ou seja, quanto maior a frequência (o clock), menor será o tempo de execução e portanto mais rápido será o processador.

projetado por uma porta 5V, uma 3.3V e três GND. Este Arduino contém um microcontrolador Atmega32u4, contendo uma memória Ram de 2.5 KB e uma Flash 32 KB, possibilitando um processamento de 16 Mhz .

O Arduino Uno é utilizado para projetos de médio e pequeno porte, pois este contém seis portas analógicas e quatorze digitais, pode alimentar o circuito entre 7V e 12V pela entrada P4, pois contém uma porta 5V, uma 3.3V e três portas GND capazes de alimentar o circuito projetado. Este Arduino contém um microcontrolador ATmega328, e apresentando uma memória Ram de 2 KB e uma Flash 32 KB possibilitando um processamento de 16 Mhz.

O Arduino Duemilanove, Figura 1 é adequado para projetos de médio e pequeno porte, pois este contém seis portas analógicas e quatorze portas digitais, pode alimentar o circuito entre 7V e 12V, pois contém uma porta 5V, uma 3.3V e três portas GND capazes de alimentar o circuito projetado. Este Arduino contém um microcontrolador ATmega328, e contendo uma memória Ram de 2 KB e uma Flash 32 KB possibilitando um processamento de 16 Mhz, sua maior diferença com o Uno é que ele é uma versão mais antiga e por isso sendo adquirido por um preço menor. O Arduino Duemilanove pode ser alimentado de três formas diferentes, sendo que via cabo USB suporta uma faixa de tensão entre 7V e 12V, pelo plug P4 com os limites entre 7V e 12V, ou pela porta VIN que fornece em média 5V.

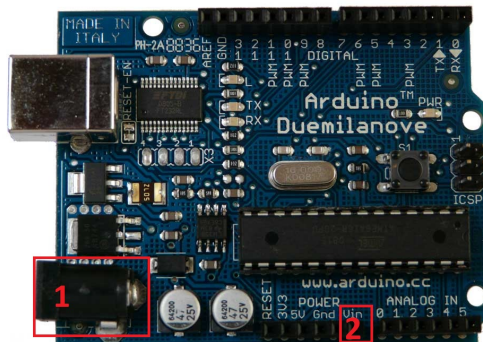


Figura 1 – Arduino Duemilanove ATmega328; 1- Plug P4, 2- Porta VIN

Fonte: www.arduino.cc/en/Main/ArduinoBoardDuemilanove

É importante salientar que existe uma faixa de alimentação recomendada pelos fabricantes 7V e 12V, entretanto se alimentado com limite inferior 6V, para um projeto que requer utilizar muitas portas, ou os dispositivos conectados as portas exigirem demasiada corrente, a alimentação pode não ser suficiente. Também deve-se evitar a alimentação do Arduíno com o limite superior a 12V, pois pode superaquecer a placa, ocasionando o mal funcionamento ou ate mesmo danificar componentes da placa(ARDUINO.CC, 2016).

4.1.2 Sensor de chuva YL-83

Foram encontrados três tipos de sensores de chuva, sensor PIC, sensor P2-3A e o sensor YL-83. O sensor PIC têm as dimensões 2,0x3,0cm de área de placa, sendo pequeno em vista dos outros modelos. O sensor do modelo P2-3A tinha suas dimensões de 5,5x4,5cm, adequadas a esta monografia porém não acompanhava um módulo de regulação para diferentes condições climáticas. Em contra partida o modelo YL-83 tem as dimensões 5x4cm e possui um módulo de regulação climática.



Figura 2 – Sensor de Chuva YL-83 e Placa de Controle

Fonte: www.arduinoecia.com.br/2014/06/sensor-de-chuva-arduino.html

O sensor YL-83 mostrado na Figura 2 é usado no monitoramento do volume de líquido em contato com a placa do sensor. A sensibilidade entre o tempo seco e úmido é detectado pela saída digital D0. Este módulo possui dois leds, um vermelho que mostra o sensor ligado e um verde que ascende quando identificado a leitura de algum líquido. É possível realizar leituras pela saída analógica A0.

O módulo do sensor recebe o sinal elétrico emitido pela placa, e através de um potenciômetro ⁴ permite a calibração da sensibilidade para diferentes climas.

A placa do sensor de chuva, é forrada em ambos os lados, com circuitos de Ni ⁵, resistente a oxidação permitindo a condutividade elétrica provocada pelo encontro do

⁴ é um resistor de três terminais onde a conexão central é deslizante e manipulável

⁵ Elemento químico Níquel

líquido com a placa.

Para um ambiente seco eleva-se a tensão aplicada na porta de saída em um dos periféricos do sensor, e na presença de ambientes úmidos provocados pelo contato com a água diminui-se a tensão. Os dados transmitidos pelos pinos A0 ou D0 informam o volume de água em contato com a placa, e limitam a magnitude do volume entre 0 e 1023.

4.1.3 Sensor ultrassônico HC-SR04

Não foram encontrados muitos sensores ultrassônicos no mercado, sendo apresentados os seguintes sensores LV-MaxSonar-EZ, HC-SR04 e SRF05. O sensor SRF05 é capaz de detectar objetos de 0 a 1,5m com precisão de 1cm, pode ser alimentado por 5V, tem dimensões de 20x15x10mm, e custa em média 60,00 reais. O Sensor ultrassônico HC-SR04 é capaz de medir distâncias de 2cm a 4m com precisão 2cm, com dimensões de 45x20x15mm, custa 20,00 reais. O sensor LV-MaxSonar-EZ é capaz de detectar objetos de 0 a 6,45m, diferentemente dos outros dois sensores ultrassônicos, sua alimentação é entre 2.5V e 5.5V, possui dimensões de 2.2x2.0x1.6cm, e pode detectar desde pequenos objetos até um toque na superfície do sensor, tem o custo de 200,00 reais.

O alcance e o custo do sensor foram as principais características na escolha, como a finalidade é auxiliar o motorista no estacionamento, o limite de 4 metros de alcance satisfaz as necessidades, e com a diferença de custo entre os sensores são de 180,00 reais, a escolha foi pelo sensor ultrassônico HC-SR04, que custa 20,00 reais.



Figura 3 – Sensor Ultrassônico HC-SR04

Fonte: <http://buildbot.com.br/blog/como-utilizar-o-sensor-ultrasonico-hc-sr04/>

O sensor mostrado na Figura 3 contém 4 pinos, onde o VCC deve ser alimentado com uma tensão de 5V e o GND utilizado para fechar o circuito, ele tem a corrente de operação de 2 miliamperes sendo bastante preciso.

Os pinos trigger e echo são responsáveis respectivamente pela emissão e recepção das ondas ultrassônicas, sendo que a diferença de tempo entre estes eventos, permitem calcular a distância entre o sensor e o obstáculo que refletiu a onda.

4.1.4 Buzzer 5V

O mercado apresenta uma variedade de buzzers, sendo que eles se diferenciam pela alimentação, frequência e intensidade da onda sonora. O buzzer escolhido deve ser

alimentado com 5V, e possui um pino de controle do oscilador interno que pode emitir ondas na faixa de frequência 1500 mhz.



Figura 4 – Buzzer Ativo 5V Bip Contínuo

Fonte: www.webtronico.com/buzzer-dc-5v.html

O buzzer mostrado na Figura 4 possui uma estrutura cilíndrica de diâmetro 12mm e altura 10mm. O interessante do buzzer é que seja integrado ao sensor de ultrassônico, esta integração entre sensor, buzzer e Arduino é importante para avisar o condutor de modo sonoro que existe algum obstáculo, para que seja efetuada a manobra correta e de forma segura.

4.1.5 Módulo bluetooth RS232 HC-05

O mercado apresenta uma variedade de dispositivo bluetooth, sendo que os principais encontrados foram os módulos bluetooth Zucchi, RS232 HC-05 e HC-06.

O módulo bluetooth Zucchi tem tensão de alimentação de 5V, suporta somente modo escravo ou seja só aceita conexão de outros dispositivos e não possui autenticação ⁶ e encriptação ⁷ e trabalha em uma frequência de 1.4GHz. O módulo HC-05 trabalha em uma tensão de 3.3V, suporta modo escravo, possui autenticação, encriptação e trabalha na frequência 2.4GHz a principal diferença deste modelo é que ele atua tanto no modo escravo quanto no modo mestre, ou seja ele aceita conexões de outros dispositivos e permite que se conectem à outros dispositivos bluetooth.



Figura 5 – Módulo Bluetooth RS232 HC-05

Fonte: www.filipeflop.com/pd-b4742-modulo-bluetooth-rs232-hc-05.html

⁶ Como é que o dispositivo se identifica na rede

⁷ Como é que os dados são encriptados ao serem enviados para a placa

O bluetooth escolhido RS232 HC-05 Figura 5 por ser programável podendo realizar as configurações como a autenticação e encriptação. Nele também é contido LEDs responsáveis por informar se o módulo está pareado com outro aparelho, caso contrário o led vermelho fica piscando, informando a não conectividade.

4.1.6 Carro robô 80RPM

Para a realização dos testes do sensor ultrassônico foi utilizado um carro robô de 80RPM⁸ e voltagem do motor de 3V à 6V. Para a escolha do carrinho não foi feito uma pesquisa detalhada de outros carros robôs pois o preço deles são elevados, com o custo entre 150,00 a 250,00 reais que muda de acordo com sua estrutura(tamanho) e RPM, portanto para o teste foi disponibilizado um carrinho da Figura 6 do laboratório de robótica da Universidade Federal do Mato Grosso do Sul campus Pantanal.



Figura 6 – Carrinho Arduino, Carro Robô 80RPM

Fonte:<https://www.usinainfo.com.br/robotica-e-motores/carrinho-arduino-carro-robo-4wd-80rpm-acrilico-46mm-kit-chassi-2871.html>

4.2 Ferramentas Computacionais

Para construção de algoritmos compatíveis com a plataforma Arduino existem diferentes softwares, sendo o mais comum e prático o Arduino software versão 1.6.4 que pode ser obtido na página oficial do Arduino. Este software foi programado utilizando a linguagem Java, pode ser instalado nos sistemas operacionais Windows, Mac OS X, e Linux, e permite utilizar qualquer modelo de placa Arduino. Para criação de gráficos em tempo real foi escolhido o Excel juntamente com o programa PLX-DAQ que é necessário para intermediar a conexão Arduino (IDE) e Excel. Em contrapartida o aplicativo ATC lite disponível de forma gratuita na play store, tem compatibilidade somente com sistema operacional android.

⁸ Rotação por minuto

4.2.1 Arduino software 1.6.4

Programas escritos usando Arduino Software (IDE) são chamados de sketches. Estas sketches são escritos no editor de código e são salvos com a extensão de arquivo “.ino”. O editor tem recursos para corte, colar, buscas e substituições de textos. A área de mensagem dá um feedback ao salvar e exportar e também exibe erros, Figura 7. O canto superior direito se localiza o monitor serial onde pode visualizar os valores lidos em tempo real das informações instanciadas no código. (ARDUINO.CC, 2016)

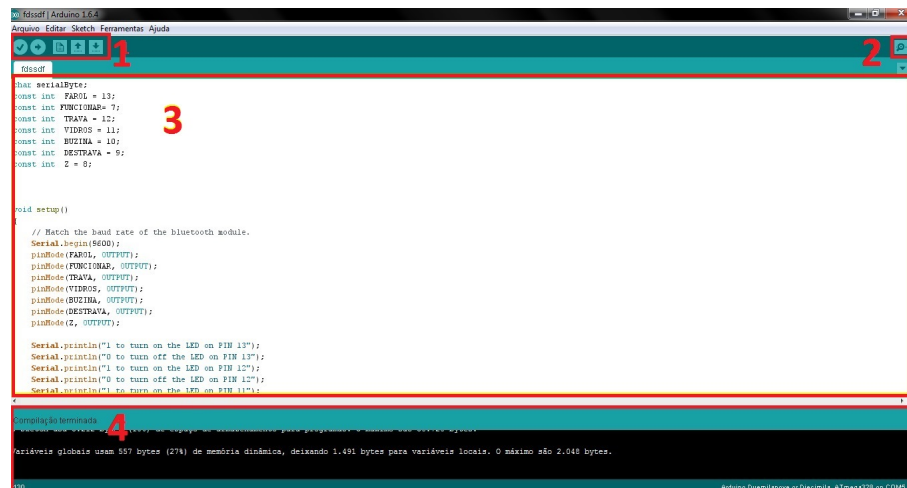


Figura 7 – IDE Arduino versão 1.6.4; 1-Verificar, compilar, corte, colar e busca; 2-Serial Monitor; 3-Editor de código; 4-Área de feedback

4.2.2 Excel e PLX-DAQ

Foi realizada uma pesquisa por programas que a criassem gráficos em tempo real e pudessem ser conectado a interface do Arduino IDE para realizarem os testes do sensor de chuva e ultrassônico. Alguns programas encontrados foram o Matlab, Scilab e o Excel, entretanto o Matlab é um software pago e caro, o Scilab apresentou problemas na conexão com o Arduino IDE, sendo assim foi utilizado o Excel junto ao programa PLX-DAQ Figura 8 que lê a porta serial da IDE do Arduino e faz a transmissão em tempo real das informações lidas pelos sensores. O programa apresenta uma interface com as opções de configurações, se o está conectado e sendo transmitido dados. Na área de configuração é localizado o suporte para selecionar a linha de transmissão e a porta de entrada USB para conectar no Arduino.

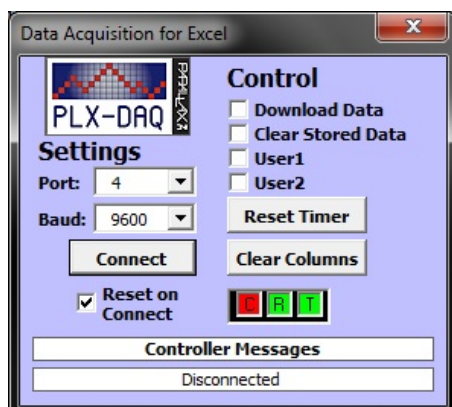


Figura 8 – IDE do programa PLX-DAQ

Fonte: imgsimon.blogspot.com.br/

4.2.3 Aplicativo ATC Lite

Foi feita uma vasta procura por aplicativos que conectasse ao módulo bluetooth HC-05, e tivesse resposta rápida para a função solicitada e principalmente que fosse gratuito, foram encontrados vários como: Arduino bluetooth, Arduino bluetooth Control Device, Arduino Bluetooth Kawalan, Arduino Bluetooth Controler e o ATC Lite, a maioria não tinha como adequar seu layout ao projeto ou a conexão não permanecia estável entre bluetooth e aplicativo.

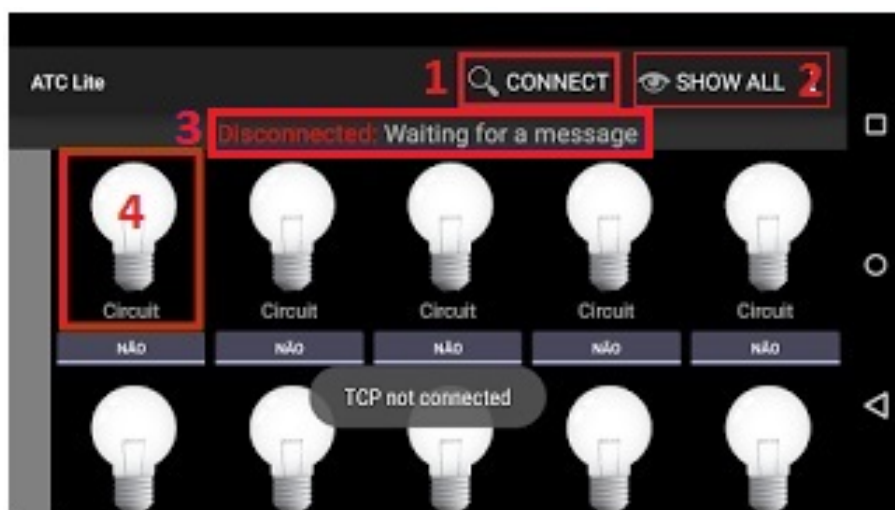


Figura 9 – Aplicativo ATC Lite; 1- Connect, 2- Show All, 3- Disconnect ou Connect; 4- Circuito “on” e “off”

Fonte: <http://eletrodex.blogspot.com.br/2015/06/controlando-led-via-bluetooth.html>

O aplicativo escolhido, chama-se ATC Lite Figura 9, disponível gratuitamente na play store, a escolha deste aplicativo foram pelos motivos de criar e modificar o seu layout e adequa-lós conforme o objetivo, podendo fazer conexões simultâneas com até sete outros dispositivos, seja bluetooth ou wi-fi. Neste aplicativo pode-se tirar fotos momentâneas e

adequando suas funções para a automação. O seu principal objetivo é auxiliar em projetos de automação doméstica avançada, veicular e projetos escolares.

O ATC Lite apresenta em sua interface o menu “CONNECT” onde é realizado a procura dos dispositivos bluetooth ou Wi-fi, mostra em vermelho ou verde se o aplicativo está conectado em algum dispositivo exibindo a mensagem “disconnect” ou “connect”. Os layout das lâmpadas podem ser adequados as imagens de acordo com o objetivo desejado, as opções de “on” e “off” onde é enviado os bytes fazendo ligar ou desligar a função que ali foi programada, o menu “SHOWALL” onde mostra todas as configurações disponíveis do aplicativo, onde também é notado a opção de configuração do código implementado na placa Arduíno.

5 Metodologia

Inicialmente foi realizado uma revisão bibliográfica no repositório de periódicos da Capes de artigos científicos relacionados ao tema automação veicular, sendo que foi pouco utilizado a biblioteca pública. Feita a seleção dos materiais compatíveis e adequados aos objetivos da monografia, utilizou-se do buscador Google para realizar pesquisas em sites da web e dos manuais e tutoriais que auxiliam nas criações de projetos. O passo seguinte foi a implementação dos materiais escolhidos e simulação com os protótipos reais, permitindo registrar medidas quantitativas de seu alcance e sua sensibilidade em relação a variação dessas medidas. Por fim foi adaptado alguns protótipos em um veículo de passeio corsa sedan 2001 para realizar as funções propostas neste trabalho.

Este Capítulo apresenta as implementações dos materiais e ferramentas computacionais utilizadas, testes e aplicações de protótipos construído com estes materiais, bem com a integração em um veículo real.

5.1 Implementações dos materiais

As implementações iniciaram-se pelo sensor de chuva YL-83, seguidos pelo ultrassônico HC-SR04, e em seguida configuração do módulo bluetooth RS232 HC-05. Sendo que após a implementação de cada um dos sensores, foram realizados testes para verificar suas respectivas sensibilidades.

5.1.1 Sensor de chuva YL-83

Na implementação deste sensor, foi utilizando uma protoboard para interligar o circuito composto pelo sensor YL-83, leds e Arduíno Duemilanove, foi utilizado uma chave para a regulagem da sensibilidade do potenciômetro do módulo como mostra a figura abaixo.

Na alimentação do circuito, foi utilizado a porta de 5V do Arduíno e distribuído na protoboard, possibilitando alimentar em paralelo todos os leds mostrados na Figura 10, a alimentação da placa Arduíno e a leitura do sensor foi realizada via cabo USB. O algoritmo no Anexo A.1 foi responsável por controlar as tensões das portas das portas digitais 5,6 e 7, em função das leituras realizadas pelas portas digital 2 e analógica A0, ambas conectadas ao módulo do sensor.

Após montar todo o circuito, foram feitos testes com uma pequena quantidade água provocada por um conta-gotas, seguido da variação do volume de líquido inserido sobre a placa. O software PLX-DAQ descrito na seção materiais realizou a leitura da

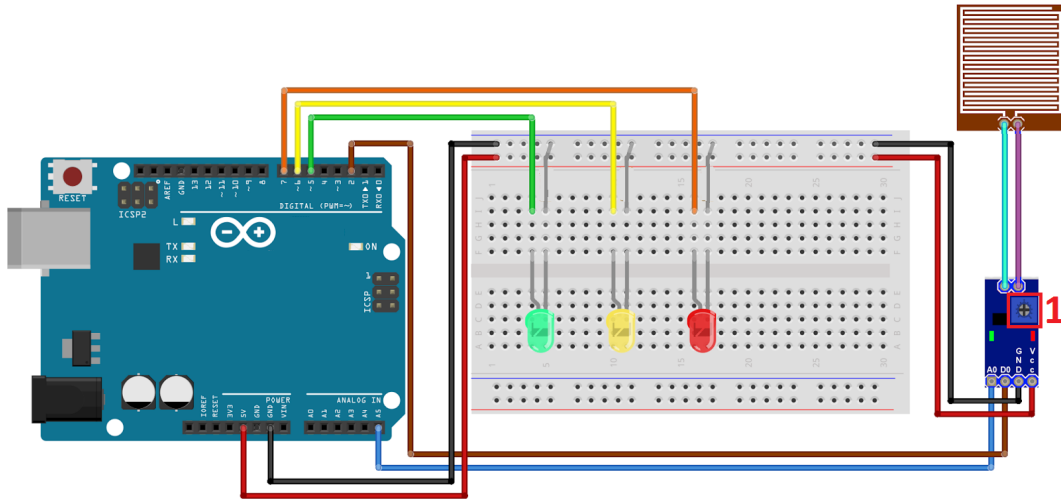


Figura 10 – Protótipo do sensor de chuva YL-83 utilizando os leds como simulador de velocidade do temporizador do para-brisa; 1-potenciômetro

porta serial e transmitiu os dados para o Excel. Em seguida, simula-se a velocidade dos atuadores de resposta a variação desse volume, como descrito no algoritmo do Anexo A.8, sendo utilizado um conjunto de três proposições para comparar os parâmetros das variáveis midade e atribuir valores para variáveis velocidades. Por fim recorre-se a uma função do primeiro grau $y = -0.2443 * x + 250$, para simular a função da velocidade dos atuadores em relação a umidade do sensor, permitindo respostas contínuas para o atuador.

5.1.2 Sensor ultrassônico HC-SR04

Na implementação do sensor ultrassônico HC-SR04, foi montado o circuito ligado diretamente no Arduino. O circuito é alimentado pela porta 5V, sendo a alimentação do Arduino e a leitura do sensor realizada via cabo USB. O código no Anexo A.2 juntamente com a biblioteca ultrasonic controlam as tensões das portas digitais 10 e 13, que são responsáveis respectivamente pela alimentação do pino emissor e leitura do sinal provocado pela onda recebida após ser refletida por um obstáculo como mostra a Figura 11.

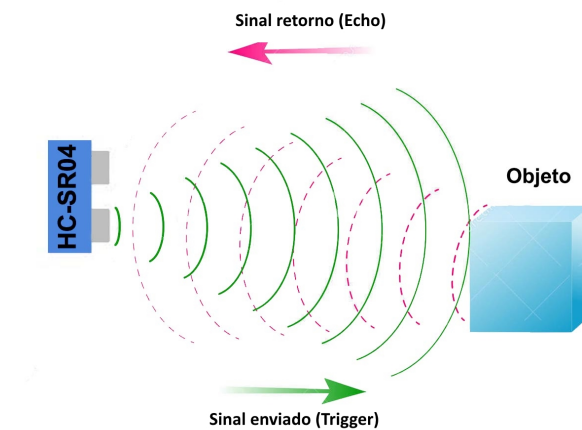


Figura 11 – Funcionamento do sensor ultrassônico com um obstáculo no caminho

Fonte: <http://buildbot.com.br/blog/como-utilizar-o-sensor-ultrasonico-hc-sr04/>

Após o circuito montado conforme a figura abaixo iniciaram-se os testes com os obstáculos parados para verificar a precisão do sensor, em seguida sua sensibilidade mediante aos obstáculos em movimento. Foi utilizado o software PLX-DAQ para realizar a leitura da porta serial mensurando a distância em função do tempo para o obstáculo em movimento, e sua sensibilidade com a taxa de variação da distância em relação ao obstáculo em movimento. Por fim, realiza-se o cálculo da velocidade e aceleração do obstáculo derivando duas vezes a relação da distância em função do tempo.

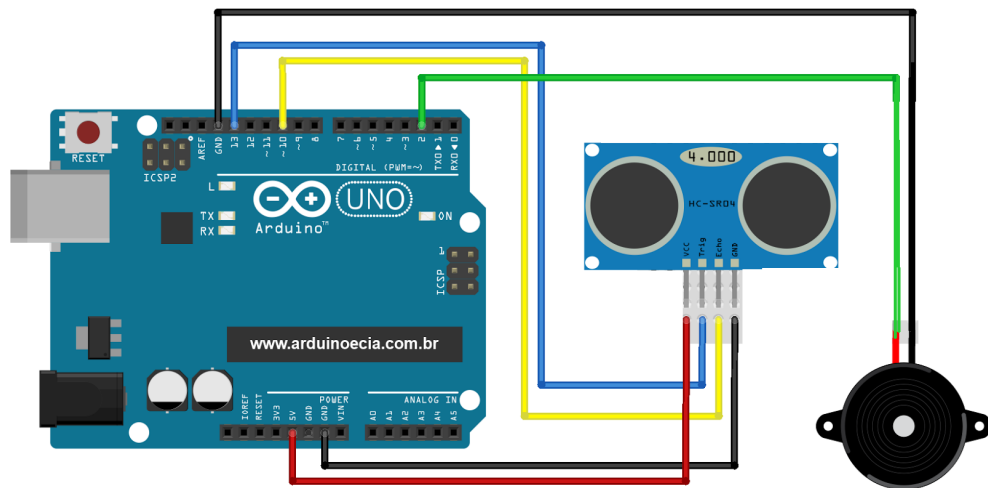


Figura 12 – Protótipo do sensor de distancia

O passo seguinte foi testar o protótipo em um carro robô para simular a frenagem do carro em função de obstáculos em movimento, que posteriormente foi substituído por um buzzer servindo de alerta em um carro real permitindo realizar a resposta da aceleração em função da distância de obstáculos fixos e obstáculos em movimentos como em ambientes contendo paredes, árvores, canteiros e animais. Foram simulados obstáculos com as próprias

mãos, ao realizar interrupções esporádicas da onda emitida pelo sensor ultrassom. O sensor ultrassônico foi inserido no carrinho para realizar a leitura dos obstáculos, sua alimentação era feita por uma bateria externa acoplada junto ao carrinho conforme a Figura 13.

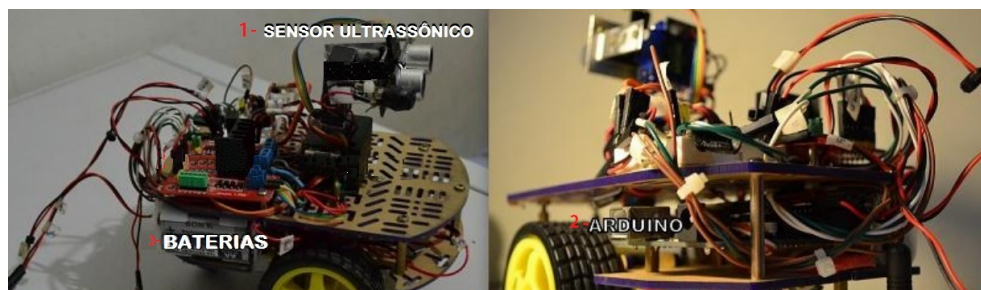


Figura 13 – Protótipo do sensor de distancia, 1- sensor, 2- Arduino, 3- Baterias

5.1.3 Configuração bluetooth

A configuração do módulo bluetooth RS232 HC-05 em modo mestre foi necessária para inserir a criptografia e alterar seu nome, para ser feito essas modificações o pino Key do módulo deve estar conectada a uma porta digital da placa como na figura abaixo, foi utilizado a protoboard para que fizesse a distribuição do GND da placa do Arduíno Duemilanove para alimentar o módulo bluetooth, não foi necessário a utilização de nenhum resistor na conexão entre a porta 5V da placa com o pino VCC do módulo, as configuração das portas de transmissão(TX) e recepção(RX) foram conectadas e realizado a implementação do código no Anexo A.3 junto a biblioteca softwareserial para permitir a comunicação serial e os comandos do Anexo A.4.

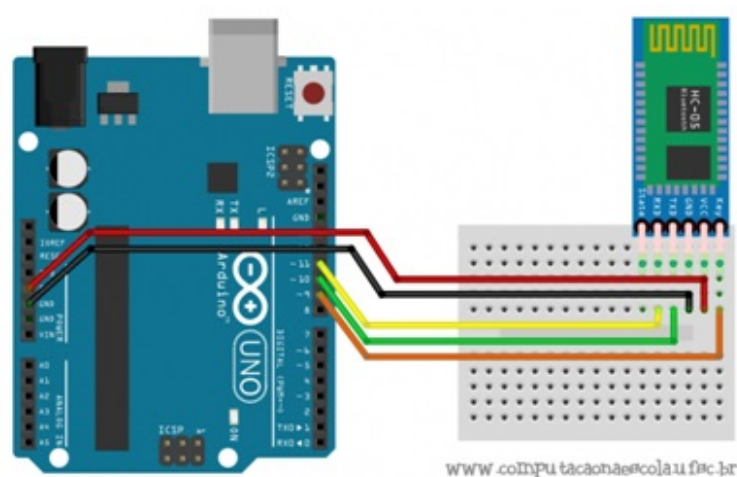


Figura 14 – Esquema da configuração de bluetooth RS232 HC-05 em modo mestre

Fonte: www.computacaonaescola.ufsc.br/?pageid=43

5.2 Integração dos materiais no veículo real

A adaptação dos sensores e módulos ao veículo teve-se início pela integração do sensor de chuva YL-83 ao para-brisa, seguido pela alimentação do sensor ultrassônico HC-SR04 com a bateria do carro, e finalizando pela conexão do bluetooth para acionar as opções de buzina, abrir e fechar os vidros, travar e destravar as portas, ligar e desligar os faróis.

5.2.1 Sensor de chuva

Na integração do sensor de chuva YL-83 no veículo, obteve-se o auxílio de um eletricista da área para retirar o módulo de comando localizado atrás do volante, permitindo assim coletar as medidas de tensões nos terminais de saídas deste módulo nos três tempos diferentes, velocidades normal, média e rápida. Estes valores de tensão serviram para parametrizar a simulação dos atuadores em resposta ao nível de umidade detectado pela placa do sensor.

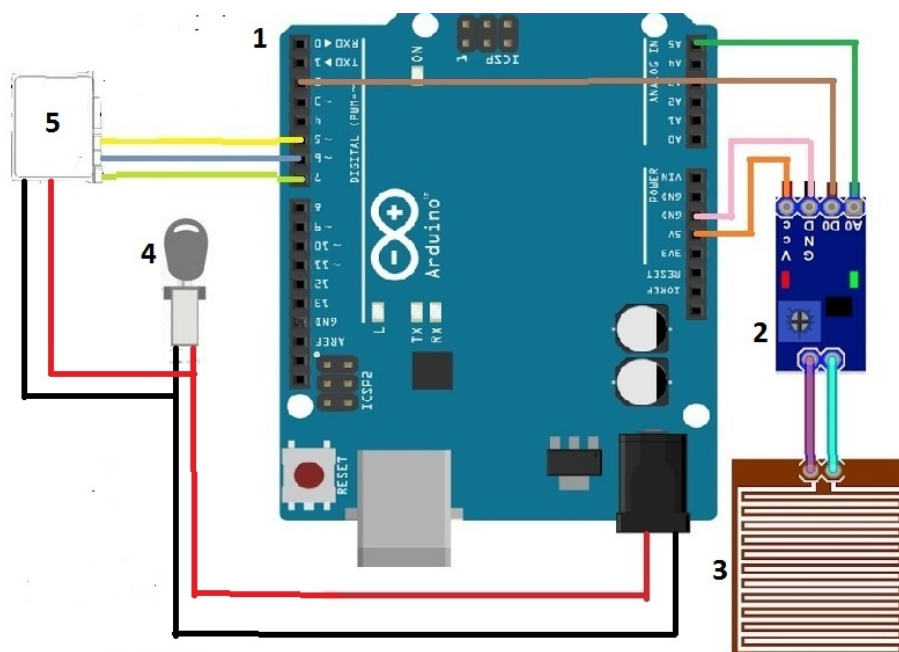


Figura 15 – Diagrama da Conexão do sensor de chuva; 1- Arduino Duemilanove, 2- Módulo, 3- Placa do Sensor, 4- Chave de Ignição, 5- Módulo de Comando

Na conexão do sensor de chuva ao módulo de comando do para-brisa como mostra a Figura 15 foi utilizado uma garrafa de água para o teste de acionamento da função junto a implementação do algoritmo da Seção A.1 na qual o para-brisa é acionado de maneira correta correspondendo a quantidade de água lida, porém não foi possível realizar o teste do sensor instalado no veículo com chuva real, pois era necessário a retirada do módulo de comando para o lado de fora e sempre com supervisão de um profissional, e nos dias de supervisão não havia chuva, entretanto foi simulado com chuva real o sensor de chuva na

superfície do para-brisa do veículo parado para medir em tempo real seu comportamento diante a chuva, sendo assim utilizando o programa PLX-DAQ para realizar a leitura das informações para o Excel plotar os gráficos.

5.2.2 Sensor de estacionamento

Na integração do sensor ultrassônico HC-SR04, como mostra a figura abaixo, foi utilizado um buzzer emitir a vibração sonora, foi mensurado a tensão elétrica entre os fios negativo e positivo responsáveis pela partida do carro, verificando a compatibilidade com os materiais integrados ao veículo, sendo utilizado 2m de fios para conectar o sensor até a parte traseira do veículo, o microcontrolador e o buzzer ficaram situados próximo a caixa de fusível.

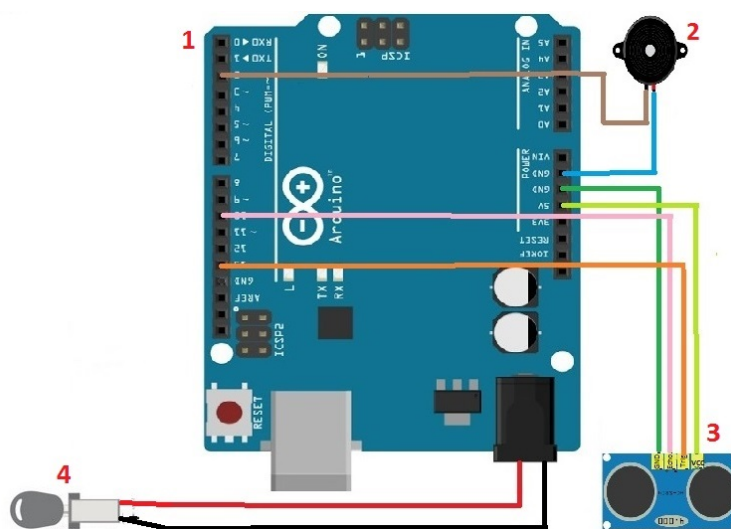


Figura 16 – Diagrama da Conexão do sensor de estacionamento, 1- Arduino Duemilanove, 2- Buzzer, 3- Sensor ultrassônico, 4- Chave de ignição

5.2.3 Sistema de portas e alerta

Na integração do módulo bluetooth HC-05 ao veículo foi construído o circuito mostrado na figura abaixo, composto pelo sistema de portas e alerta. Esta figura mostra a alimentação do circuito pela bateria do carro, sendo utilizado os portas RX e TX do Arduino para receber e enviar dados aos respectivos pinos RX e TX do módulo bluetooth.

A Figura 17 mostra as conexões das portas digitais 13,12,11,10,9 e 8 do Arduino ao sistema elétrico do carro, podendo controlar as funções de subir e descer dos vidros, travar e destravar portas, acionar a buzina e os faróis. A interação é realizada pelo módulo de bluetooth que após receber um sinal emitido por outro dispositivo, no presente caso esse sinal foi emitido por smartphone com sistema operacional android utilizando o aplicativo ATC Lite.

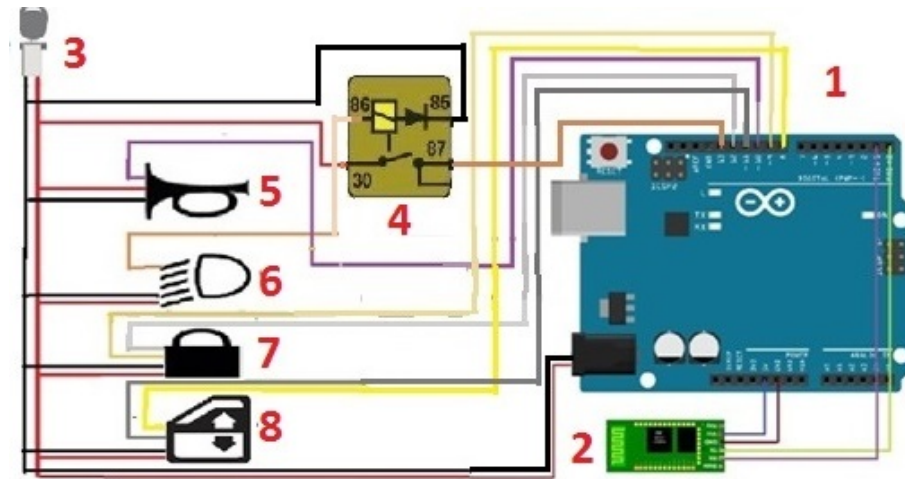


Figura 17 – Protótipo enviado a empresa, 1- Arduíno Duemilanove, 2- Módulo Bluetooth, 3- Chave de ignição, 4- Relê, 5- Buzina, 6- Farol, 7- Trava, 8- Vidros

No aplicativo foi realizado as configurações de acordo com a programação do código e modificado seu layout para figuras que representassem as funções veiculares. Utiliza-se do envio de bytes do celular para a placa conectada no veículo clicando sobre os botões de “on” e “off”, na opção *data sent on touch* é enviado o byte desejado quando clicado e *data sent on untouch* envia-se o byte 0 para que desligue a função, conforme a figura abaixo.

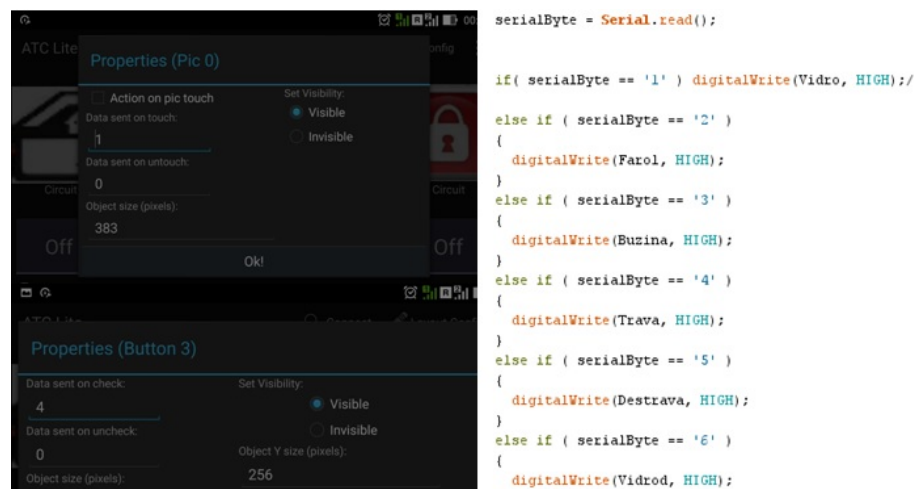


Figura 18 – Configuração entre código e aplicativo ATC Lite

6 Resultados

Foi solicitado a montagem de uma placa para que pudesse ser conectada no veículo. A plataforma Arduíno autoriza a venda e a comercialização de produtos porém com algumas condições como o material ser modificado sob a licença da *Creative Commons*¹ *Attribution*² *Share-Alike*³, podendo fabricar e vender o produto resultante (ARDUINO.CC, 2016), sendo assim a empresa Sacrasom, a qual se localiza no estado de Minas Gerais na cidade de Sacramento, empresa essa que trabalha com todo tipo de equipamento automotivo, na qual foi enviado o código do Anexo A.1, e o esquema do protótipo representado pela Figura 17. Este Capítulo apresenta as implementações dos materiais e ferramentas computacionais utilizados, testes e aplicações de protótipos construído com estes materiais, bem com a integração em um veículo real.

6.1 Sensor autônomo de chuva

Com a supervisão de um eletricista da área veicular foi possível a localização e conhecimento sobre o módulo de comando do para brisa que localizado em um lugar de difícil acesso (atrás do volante) e passível de problemas como curtos circuitos devido a conexões erradas. Foi constatado que para o veículo utilizado a instalação do sensor de chuva deve ser colocado no meio do para-brisa. Essa posição pode mudar de carro para carro dependendo de quantos limpadores que o veículo possui e a posição destes limpadores, evitando o choque do limpador com o sensor.

Durante o procedimento sob supervisão do profissional foi utilizado o multímetro para evitar qualquer danificação elétrica ao Arduíno, placa ou veículo, sendo assim foram medidos as conexões do módulo com Arduíno em modo médio⁴ detectando de 0,02A a 0,11A e standby⁵ 0,02A porém o multímetro detectou uma variação ocasionada na mudança das velocidades do para-brisa, na passagem do modo standby para velocidade baixa ouve uma mudança na amperagem de 0,02A para 0,08A depois mantendo uma

¹ Uma entidade sem fins lucrativos criada para permitir maior flexibilidade na utilização de obras protegidas por direitos autorais. A ideia é fazer com que um autor/criador possa permitir o uso mais amplo de seus materiais por terceiros, sem que estes o façam infringindo as leis de proteção à propriedade intelectual

² Direito sob a licença da Creative Commons de adaptar para qualquer fim mesmo comercial, utilizando os seguintes termos, você deve dar o crédito apropriado, prover um link para a licença e indicar se mudanças foram feitas, você deve fazê-lo em qualquer circunstância razoável, mas de maneira alguma que sugira ao licenciante a apoiar você ou o seu uso

³ Direito sob a licença da Creative Commons de compartilhar igual. Se você remixar, transformar, ou criar a partir do material, tem de distribuir as suas contribuições sob a mesma licença que o original.

⁴ Medida com o produto funcionando

⁵ modo de espera é o termo usado para designar o consumo de energia elétrica em modo de espera

estabilidade de 0,07A. Na transição da velocidade baixa para média ouve uma pequena variação entre 0,07A e 0,09A, se estabilizando nos 0,07A e na passagem da velocidade média para rápida mudando de 0,07A para 0,11A e se estabilizando nos 0,07. Foram medidos a amperagem do carro enquanto estava desligado e depois de ligado sem o uso das velocidades, a amperagem reagiu de forma normal de 0,00A para 0,02A dado essas informações demonstra sem necessidade a utilização de resistores ou relês nas conexões dos fios sem perigo de danificações ao carro ou ao Arduíno. Após a instalação verificou-se que os movimentos realizado pelas paletas do para-brisa corresponderam aos respectivos comando das portas do Arduíno, sendo acionadas com o aumento do volume de água mensurado pela placa. Foi possível alterar entre as três velocidades disponíveis no carro.

Foram realizados os testes com o sensor de chuva no veículo parado, para obter a sua leitura nos dias 09/04/2016 as 15:00:00 horas (horário de Brasília) e 10/04/2016 as 15:22:00, no Anexo A.9 e A.10 encontra-se todos os dados sobre o clima no momento que foi realizado o teste, esses dados são fornecidos pelo CPTEC/⁶INPE⁷. Os gráficos que serão apresentados abaixo demonstram as leituras realizada pelo sensor em referência ao tempo. A leitura é feita por parâmetros pré-estabelecidos dentro do algoritmo de 1024 até 900 para velocidade baixa, 900 até 400 velocidade média e 400 até 0 velocidade rápida.

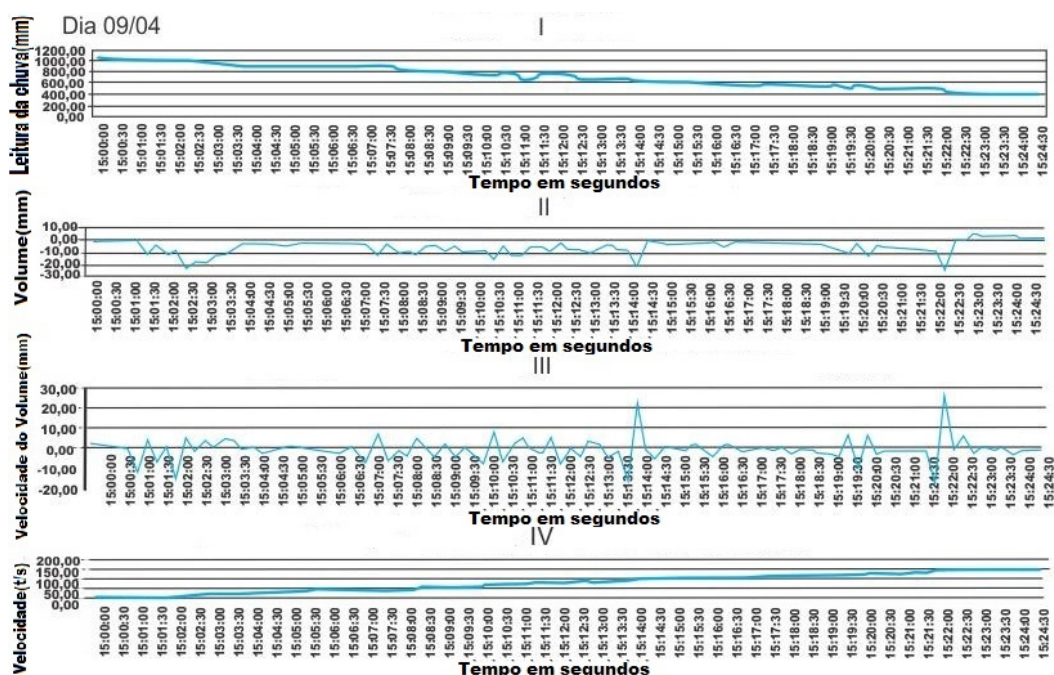


Figura 19 – Gráfico I-Leitura do Sensor durante o dia 09/04/2016; Gráfico II Taxa de variação do volume de chuva, Gráfico III- Taxa de Variação da velocidade do volume de chuva; Gráfico IV- Mudança de velocidade do para-brisa de acordo com os parâmetros

⁶ Centro de Previsão de Tempo e Estudos Climáticos

⁷ instituto nacional de pesquisa espaciais

Os gráficos I e IV do dia 09/04/2016 da Figura 19 mostram a relação entre a velocidade simulada da paleta e a quantidade de volume de água mensurado pelo sensor e pode ser observado no gráfico IV que no momento em que os parâmetros da velocidade começam a aumentar no instante 15:01:30, indicam que a placa começa a sair do estado seco para úmido, como mostram as leituras do gráfico I, entretanto os parâmetro para velocidade simulada não representa a velocidade real da paleta. A velocidade simulada está parametrizada entre os valores de 0 e 255 que correspondem ao intervalo de valores utilizados pelas portas analógicas do Arduíno, esse valor é diretamente proporcional a tensão aplicada pela porta, variando entre 0V e 5V e possibilita o controle de forma gradual e contínua da velocidade de um atuador, como um motor que tem a rotação proporcional a magnitude da tensão aplicada.

Os gráficos II e III mostram os instantes que ocorrem os aumentos no volume da chuva entre dois instantes seguidos, os três picos em descensão visto no gráfico II, instante 15:02:30, 15:14:30, e 15:22:30 representam o aumento relativamente grande do volume da chuva, durante um intervalo de tempo curto. Os valores negativos da velocidade da chuva, gráfico II, deve-se ao fato da leitura decrescente do sensor de chuva, que é inversamente proporcional ao valor do volume de chuva. O gráfico III mostra que nos instantes 15:14:15 e 15:22:15 ocorreu uma rápida estiagem, ocasionando uma pequena queda na curva do gráfico I, diferente do que ocorre no instante 15:02:30, que não apresenta uma estiagem significativa para evitar a queda da curva da Figura I nesse instante.

Como no primeiro teste realizado no dia 09/04/2016 não foi suficiente para ler todas as velocidades do sensor por não ter caído uma grande quantidade de chuva, foi realizado outro teste, no dia 10/04/2016 as 15:22:00 horas (horário de Brasília).

Os gráficos I e IV do dia 10/04/2016 da Figura 20 apresenta a coorelação entre a velocidade simulada da paleta e a quantidade de volume de água lido pelo sensor, e pode ser visto no gráfico IV que no momento em que os parâmetros da velocidade começam a aumentar no instante 15:28:00 a placa começa a sair do estado seco para úmido como nas leituras do gráfico I.

Nos gráficos II e III ocorrem aumentos no volume da chuva entre dois instantes seguidos, as duas descensão presentes no gráfico II nos instantes 15:36:00 e 15:40:40 representam o aumento do volume da chuva durante um intervalo de tempo curto. Os valores negativos da velocidade da chuva no gráfico II é ocorrido pelo motivo da leitura decrescente do sensor de chuva. O gráfico III mostra que nos instantes 15:35:00, 15:39:40 e 15:40:00 ocorre uma rápida estiagem que resulta uma leve queda na curva do gráfico I.

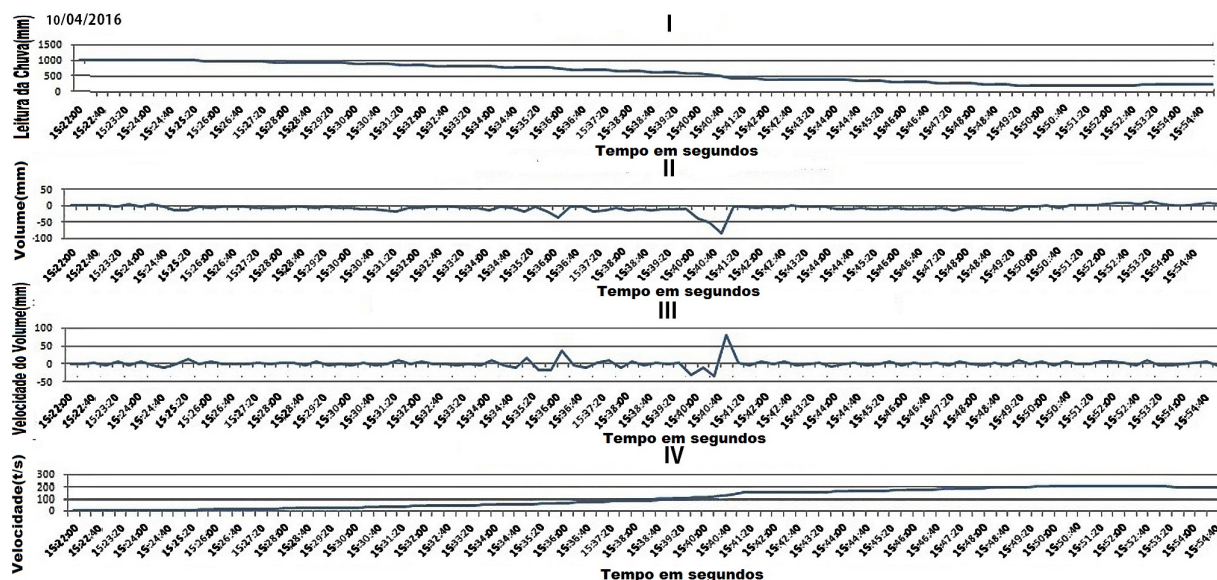


Figura 20 – Gráfico I-Leitura do Sensor durante o dia 10/04/2016; Gráfico II Taxa de variação do volume de chuva, Gráfico III- Taxa de Variação da velocidade do volume de chuva; Gráfico IV- Mudança de velocidade do para-brisa de acordo com os parâmetros

6.2 Sensor de apoio ao estacionamento

O processo de instalação do sensor ultrassônico HC-SR04, buzzer e Arduino foi realizado com os conhecimentos passados anteriormente pelo auxílio do profissional, conforme as orientações é necessário a utilização do multímetro para evitar qualquer danificação elétrica ao Arduino, placa, sensor ou veículo foram realizadas as medidas nas conexões do positivo e negativo da ignição com Arduino em modo médio detectando 0,11A ou seja quando encontrou um obstáculo se mantendo estável e sem alterações, ao passar para o modo standby foi detectado 0,02A, foi medido a amperagem do carro enquanto estava desligado e depois de ligado, 0,00A para 0,02A dado essas informações demonstra uma boa estabilidade e sem necessidade de utilização de resistores ou relês na conexão dos fios sem perigo de danificação ao carro ou material.

O teste realizado com o carrinho robô descrito na subseção 5.1.2, teve como objetivo gerar gráficos em tempo real simulando a distância em função do tempo, velocidade em função do tempo, aceleração em função do tempo demonstrando a reação do buzzer que foi implementado no veículo e assim a eficiência da leitura do sensor ultrassônico conectado ao buzzer como um sensor de estacionamento, no Anexo A.6 encontra-se o link do vídeo sensor integrado ao veículo.

O primeiro teste realizado com o ultrassom apresentou os resultados mostrados na figura abaixo. Essa figura mostra a medida da distância em função do tempo referente a um obstáculo em movimento, bem como o cálculo de sua velocidade e aceleração. O algoritmo do Anexo A.8 foi responsável em simular a resposta para possíveis atuadores, e

pode ser vista no gráfico IV da figura abaixo.

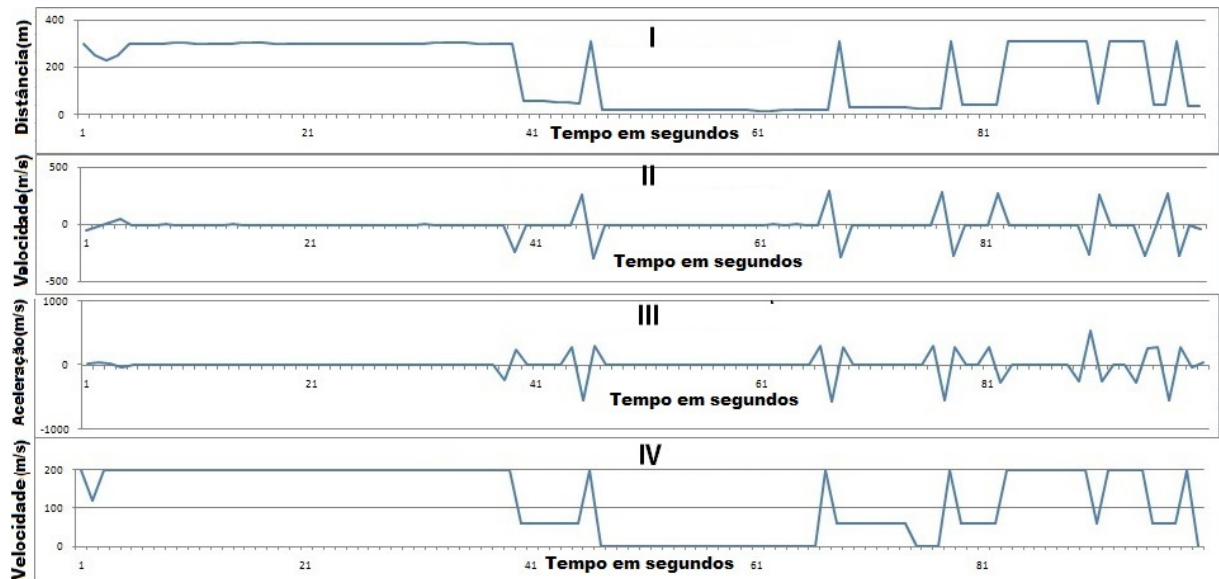


Figura 21 – Medidas da distância entre o ultrassom e um obstáculo em movimento, cálculo da velocidade e aceleração do obstáculo, e simulações da velocidade de alimentação do atuador de resposta ao obstáculo. A figura mostra respectivamente Gráfico I- Distancia em em função do Tempo, Gráfico II- Velocidade em função do Tempo, Gráfico III- Aceleração em função do tempo, Gráfico IV- Velocidade em função do Tempo com parâmetros

Os resultados mostrados pela figura acima, apresentam a resposta prevista que pode se visto comparando o Gráfico I com o Gráfico IV, que mostra a relação diretamente proporcional entre a velocidade do motor e distância do obstáculo. O gráfico IV mostra que a resposta do atuador são constituídos de três valores ($V=0$, $V=90$, $V=200$) que foram pré-estabelecidos no algoritmo utilizado, que possui uma programação limitada, composta por quatro posições responsáveis por atribuir a velocidade aos atuadores.

No experimento realizado com o protótipo mostrado na Figura 13, foi mensurado a distância em função do tempo, durante o percurso real em um espaço aberto descrito na Subsecção 5.2.2. Os resultados apresentado nos gráficos abaixo, mostram as medidas obtidas da distância em função do tempo para obstáculos reais, bem como as respostas obtidas após a realimentação do motores do protótipo.

Como pode ser visto, o algoritmo do Anexo A.8 corrigiu a velocidade dos motores como era esperado, nota-se no gráfico I que no instante 15 segundos a distância do obstáculo era 4,5m, e aproximava-se a uma velocidade de 255 que corresponde a velocidade máxima do carrinho de 80Rpm, Gráfico II, quando apareceu um obstáculo reduzindo a distância para 0.4m, ocasionando uma diminuição na aceleração visto no gráfico III, e como pode ser visto nos instantes seguintes, resultou na estabilização do protótipo, que segue se aproximando do próximo obstáculo a uma velocidade constante.

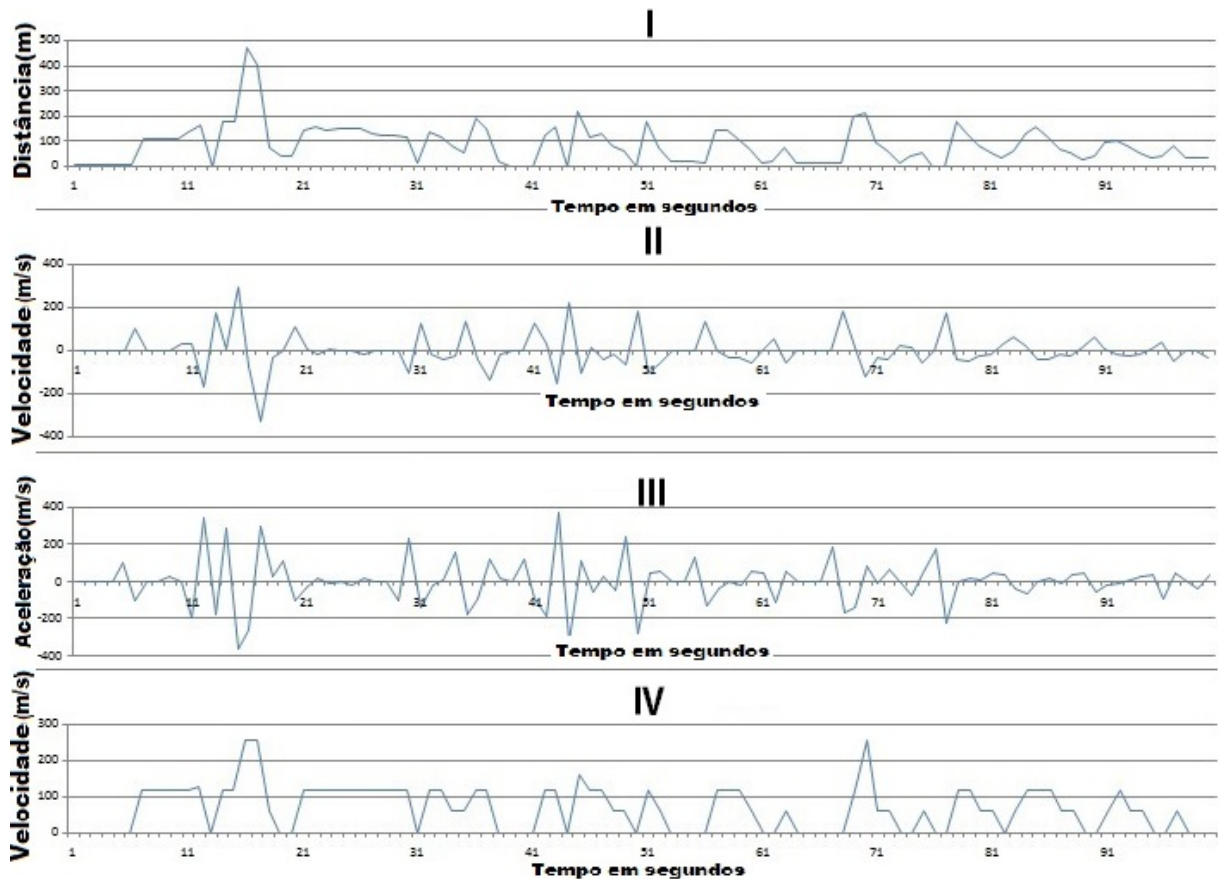


Figura 22 – Medidas da distância entre o ultrassom e um obstáculo em movimento, cálculo da velocidade e aceleração do obstáculo, e simulações da velocidade de alimentação do atuador de resposta ao obstáculo. A figura mostra respectivamente Gráfico I- Distancia em em função do Tempo, Gráfico II- Velocidade em função do Tempo, Gráfico III- Aceleração em função do tempo, Gráfico IV- Velocidade em função do Tempo com parâmetros

6.3 Hardware integrado automotivo

Para construção do hardware integrado automotivo foi utilizando a plataforma Arduino, foi necessário a configuração do módulo bluetooth para a conexão com o veículo, e enviado o protótipo para a empresa que realizou a montagem do hardware integrado de acordo com o algoritmo implementado para o funcionamento junto ao aplicativo utilizado.

6.3.1 Configuração do módulo bluetooth

Utilizando o diagrama apresentado na Figura 14 foram feitas as alterações no módulo bluetooth, utilizando os comandos disponíveis no Anexo A.4 e os parâmetros *NL* (*New Line*) e *CR* (*Carriage Return*), visando a segurança do veículo foram modificados o nome do módulo bluetooth HC-05 para TCC e a senha para 1234 pois o mesmo vinha com o nome HC-05 e sem autenticação alguma dando acesso ao hardware a qualquer pessoa com um smartphone. No código do Anexo A.3 foi necessário usar a biblioteca

softwareserial para permitir a comunicação serial em outros pinos digitais do Arduino, além de fazer com que o software disponibilize as velocidades em Bps necessárias, com velocidades de até 115.200 Bps a usada foi 9.600 Bps pois é a frequência compatível ao bluetooth HC-05, depois do código descarregado é necessário a desconexão do pino Key para carregar outro código senão o próximo código será descarregado por cima do módulo bluetooth ocasionando a perda de todas as configurações. Depois foi testado seu reconhecimento na rede junto ao aplicativo conforme mostra a figura abaixo.

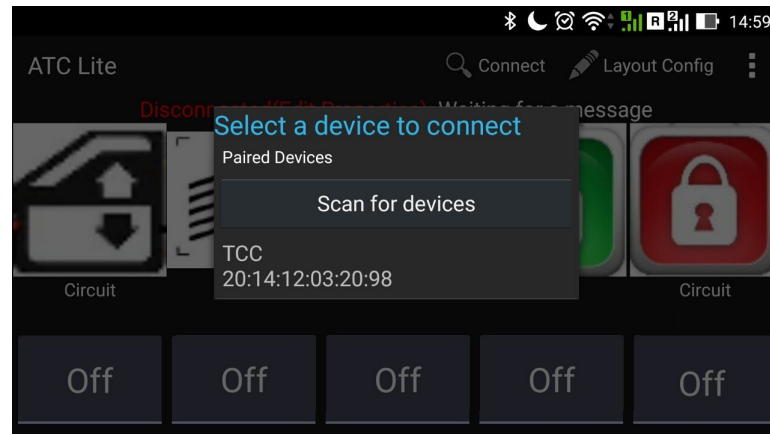


Figura 23 – Reconhecimento do bluetooth por parte do aplicativo

6.3.2 Hardware integrado automotivo

Primeiramente foi feito o orçamento na empresa Sacrasom de todos os dispositivos ligados em uma placa só, sensor de chuva, sensor de apoio ao estacionamento e as funções de sistema de portas e alertas que ficou no valor de 570,00 reais, com a inviabilidade econômica para o projeto à esse preço, foi feito o orçamento somente dos sistema de porta e alerta no valor de 270,00 porém o valor final chegou a 290,00 reais pois a empresa alegou a necessidade da compra de uma placa maior com proteção anti-aquecimento. Ao passar algumas semanas o produto foi entregue conforme a figura abaixo.

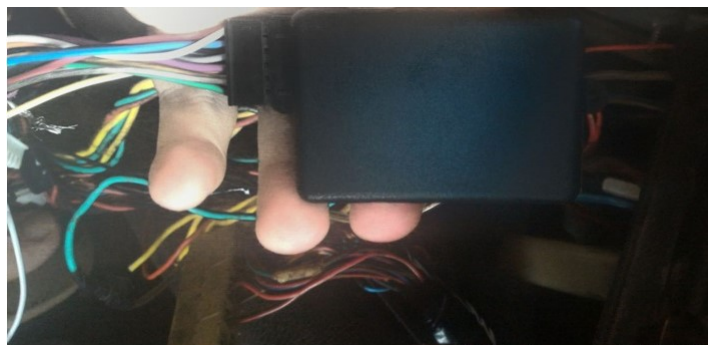


Figura 24 – Imagem do produto finalizado

Depois do produto em mãos ele foi analisado para a instalação, contendo a largura de 3,5cm, comprimento de 5cm, 54 gramas, apresentava 3 led's e 12 portas, para ser feito a integração do hardware no automóvel corsa sedan chevrolet 2001, foi necessário um conector chamado de chicote que contém 12 fios pelo valor de 15,00 reais, é o mesmo utilizado na instalação de sons automotivos.

Foi observado que cada fio do chicote é de uma cor diferente, para que seja simbolizado as funções do sistema veicular e facilitar as conexões entre o produto e o veículo, houve a necessidade da criação de uma legenda para representatividade de cada função do automóvel por uma cor do chicote. As definições das cores dos fios conectados no hardware ficaram assim: verde, marrom e azul são entradas auxiliares, branca é saída auxiliar, os fios prata e ouro realiza as funções de trava e destrava, roxo aciona buzina, laranja liga e desliga farol, cinza e amarelo sobe e desce vidros, preto e vermelho é o positivo/negativo.

Saida1(Branco)	Positivo+(Vermelho)
Entrada 3(Azul)	Vidros Descer(Amarelo)
Vidros Subir(Cinza)	Negativo-(Preto)
Buzina(Roxo)	Farol(Laranja)
Trava(Prata)	Destrava(Ouro)
Entrada1(Verde)	Entrada2(Marrom)

Figura 25 – Legenda do chicote de instalação

Na integração do produto com o carro, foi conectado utilizando a Figura 25 como referência, é notado entradas auxiliares que normalmente são encontradas em alarmes e som automotivo, elas têm como função auxiliar alguma porta que esteja com defeito é um padrão para produtos automotivos, e uma branca que é saída adicionada a mais, de acordo com a empresa a placa é padronizada com 12 portas, sendo 6 saídas, 3 auxiliares e 1 sendo positivo e 1 negativo e como não foi desenvolvido na programação algum impulso de byte para essa saída branca ela ficou como inválida. Nas conexões dos fios no caso do farol foi constatado um relê automotivo de 30A de 4 pinos onde o terminal 30 simboliza a entrada de 12V ligada no positivo da bateria, o terminal 86 ligado no negativo da bateria, o terminal 87 têm duas saídas que seguem para os faróis e o terminal 85 para interruptor local da conexão do fio laranja. Utilizando o multímetro foi realizado a medida das amperagens do produto nas suas respectivas conexões positiva e negativa do produto, se manteve estável

nos 0,02A sem a necessidade do uso de relê ou resistores. Com o produto já instalado foi construído um diagrama do produto conforme a figura abaixo.

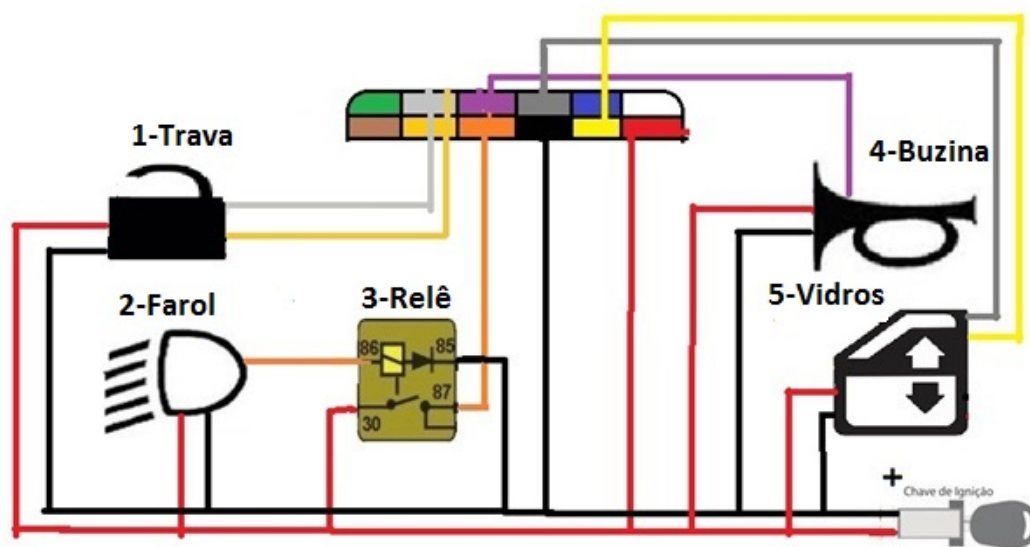


Figura 26 – O diagrama final elaborado de acordo com a instalação, nomenclatura das cores na figura 25; 1-Trava, 2-Farol, 3-Relê, 4-Buzina, 5-Vidros

Após a instalação foi aberto a caixa de anti-aquecimento e foi notado o microprocessador utilizado no sistema embarcado é um ARM Cortex M3. A ARM⁸ trabalha com o Cortex-M voltado para dispositivos embarcados, Cortex-R voltado para mediacenters⁹ e controladoras de HDs, e o Cortex-A voltado para dispositivos móveis como tablets e smartphones, onde pode ser notado também o dispositivo de bluetooth HC-05 acoplado, junto a resistores e um processador de 16 Mhz.



Figura 27 – Placa ARM Cortex

6.4 Aplicativo modificado

Para a utilização do aplicativo é necessário sua configuração, foi modificado o layout conforme a necessidade das funções, foram utilizadas imagens para representar as funções

⁸ Empresa de tecnologia britânica, conhecida por seus microprocessadores

⁹ O Media Center (Centro de Mídias) é basicamente um PC configurado para ser uma central de entretenimento.

de descer e subir vidros, travar e destravar portas, buzina e farol, para deixar o aplicativo com uma representação de aspecto veicular, conforme a figura abaixo.

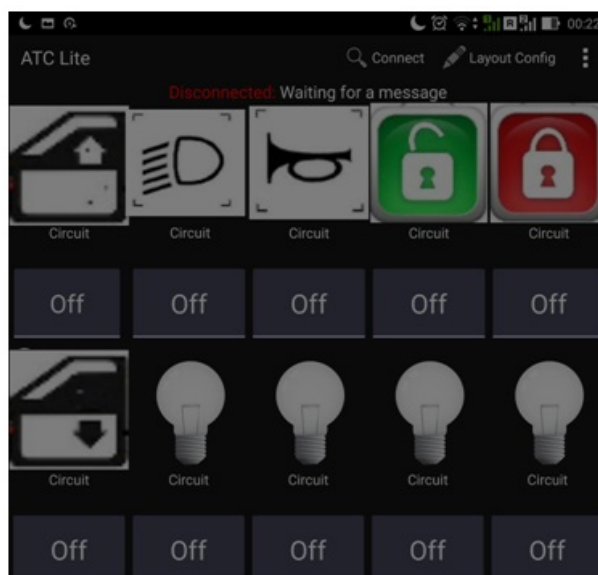


Figura 28 – Layout do aplicativo modificado

Para o envio dos serialbyte do aplicativo para o hardware integrado automotivo foram feitas as configurações descrita na subsecção 5.2.3 junto a Figura 18, depois do reconhecimento do módulo bluetooth do hardware conforme demonstrado na Figura 23, o status do aplicativo que está em vermelho "disconnect"apresentado na Figura 7 muda para verde "connect". A partir da conexão no hardware é notado que os três leds têm representatividades, um dos leds se mantém ligado para auxiliar visualmente que o produto está ligado(positivo/negativo) na chave de ignição, o outro led se mantém piscando quando se encontra conectado ao aplicativo e o ultimo led pisca somente quando é enviado um serialbyte do smartphone para o hardware, foram realizados os testes dos dispositivos de buzina, farol, trava e vidros. No teste utilizando a buzina, quando é enviado o serialbyte é ativado a função que mantém o seu barulho contínuo até ser desligado pelo celular, se for ligado e desligado rapidamente é notado o barulho intercalado sendo assim a função agiu da mesma forma como se fosse acionada manualmente. O teste com o farol foi realizado ligando e desligando, em todos os testes o farol ascendeu e apagou como o esperado porém foi notado um barulho no momento do acionamento da função, esse barulho é percebido no relê de 4 pinos que se encontra conectado aos fios do farol, entretanto o barulho não interferiu negativamente no acionamento dos faróis. Os testes das funções de trava e destrava foram feitos das seguintes formas; o veículo foi trancado pelo lado de fora e realizado o teste para destravar, obtendo êxito, depois do lado de fora foi realizado o teste para travar o veículo e novamente obtendo o resultado esperado, a trava funcionou da mesma forma como o controle do alarme do carro. Nos testes dos vidros, foram testados primeiramente subindo até o final e descendo até o final, obtendo êxito, depois feito o teste

de parar o movimento dos vidros até o meio do caminho; subir e descer, o resultado foi conforme o desejado, funcionou como se estivesse utilizando o interruptor de acionamento manual.

Uma observação muito pertinente a ser acrescentada na utilização do sistema de portas, trava e vidros, quando é utilizado estas funções é necessário estar atento a ativação do aplicativo “on” e “off”, por exemplo sendo acionado a função descer vidro mudando para “on” é enviando o serialbyte 1, e é extremamente importante, que depois do vidro abaixado volte de “on” para a opção “off”, caso isso não for feito quando se utilizar a opção subir vidros consequentemente as duas opções ficariam “on” ocasionando uma confusão no hardware, fazendo com que o aplicativo envie serialbyte de 1 para subir e 1 para descer, manualmente iso é representado como se o usuário estivesse apertando a opção de subir e descer o vidro simultâneamente. O mesmo procedimento acontece para a opção de travar e destravar portas.

6.5 Mercado para o produto

Diante a pouca disponibilidade de produtos no mercado de acordo com a pesquisa detalhada dos produtos que mais se assemelham com o produto proposto por este projeto. Foram encontrados quatro produtos similares, somente dois com disponibilidade em sites brasileiro e outros dois encontrados somente em sites de importados, nos sites brasileiros(Mercado Livre, Walmart.br) foram encontrados o SmartX que custa no valor de 450,00 reais que é um hardware que é integrado no veículo e realiza os acionamentos de buzina, subir e descer vidros, trava e destravar veículo a empresa fornece o aplicativo próprio. O Keydiy que tem como objetivo o gerencimaneto das funções de travar e destravar portas e abrir o porta mala no valor de 400,00 reais. No mercado estrangeiro foi encontrado o Viper SmartStart que ascende e apaga farol, trava e destrava o veículo e tem GPS de rastreio do automóvel incluso no seu aplicativo com o custo de 1.600,00 reais. O JFA Smart Control Bluetooth que seu objetivo principal não são as funções veiculares, ele controla as principais funções do som automotivo através de um controle, os sinais são enviados via bluetooth e os comandos são captados pela central em distâncias de até 60 metros, monitora e apresenta o nível da carga de bateria do sistema de som, isso no valor de 350,00 porém para obter o controle das funções do veículo é necessário comprar um hardware adicional que disponibiliza o controle de seis funções do veículo, este módulo adicional para ser adquirido é cobrado mais 145,00 reais totalizando 495,00 reais, lembrando que o módulo só pode ser adquirido junto ao produto que gerencia o som automotivo pois é um controle que auxilia nos comandos das funções e não um smartphone. O hardware integrado automotivo deste projeto visando o baixo custo saiu por 290,00 reais, se fosse integrado todas as funções em conjuntos como o sensor de chuva e estacionamento ficaria no valor de 570,00 reais acima dos produtos apresentados entretando o usuário seria conteplados com

mais funções e essas funções que não são disponibilizados pelos produtos similares, e no mercado o sensor de chuva é vendido somente integrado no veículo ou seja veículos novos de fábrica, e o sensor de estacionamento pode ser achado separado no valor de 70,00 reais ou nos novos modelos de carros, ou seja o protótipo continuaria com um custo/benefício interessante.

7 Limitações do trabalho

As limitações a serem destacadas deste trabalho são: o sensor de apoio ao estacionamento que não é aprova d'água foi feito o teste colocando de baixo da chuva, resultando na queima do dispositivo, e a instalação fixa do protótipo dos sensor de estacionamento no carro pois teria que ser realizadas modificações internas e externas no veículo, para fazer as passagens dos fios até a parte traseira do automóvel necessitando de mão de obra especializada e financeiro.

Não foi previsto chuvas reais para lidar com sensor de para-brisa integrado no veículo em movimento, pois havia necessidade de auxílio de um profissional constante na retirada do módulo de comando do veículo que ficava em um local de difícil acesso na parte posterior do volante e a espera da chuva não seria viável pois colocaria a segurança do condutor em risco por andar com o dispositivo “pendurado” no carro, além da necessidade da retirada do para-brisa para passar os fios por dentro do veículo demandando mão de obra e dinheiro, entretando foi realizado os testes com chuva real para a realização da leitura do sensor e simulação das mudanças das velocidades do para-brisa. Outras limitações foram as faltas de dados exatos por parte do CPTEC/INPE referente a quantidade de chuva em milímetros para os dias dos testes, pois os gráficos de precipitações de chuva em mm apresentado no anexo do trabalho mostra 0mm para os dois dias e ao mesmo tempo mostra que está chovendo durante o dia.

Algumas limitações com o hardware integrado automotivo foi a não integração da parte mecânica ao controle veicular, a ideia originalmente era conectar sensor de apoio ao estacionamento e sensor de chuva ao hardware mas por questões financeiras não foi realizado, entretando foi contruído os protótipos de forma separadas e realizados os testes de funcionamento dos mesmos. No caso do aplicativo ATC Lite não foi desenvolvido pelo autor podendo sofrer alguma modificação ou atualização que talvez impeça o manuseio das funções porém foi testados outros aplicativos como: Arduino bluetooth, Arduino bluetooth Control Device, Arduino Bluetooth Kawan, Arduino Bluetooth Controller, para caso isso ocorra, esses aplicativos conseguem substituir o ATC Lite.

8 Conclusão

Foi concluído que este trabalho, conseguiu desenvolver um hardware de controle fundamentado em técnicas de arquitetura de hardware Open Source, implementando diferentes atuadores, sensores e integrando com interface do smartphone em um sistema operacional android possibilitando a construção de uma base de controle do sistema elétrico do veículo.

O sistema de apoio ao estacionamento que se encaixa no sistema passivo foi integrado ao carro, seus testes foram suficientes para a demonstração da resposta do sensor diante a um obstáculo, e o sistema autônomo de chuva que se encaixa no sistema ativo foi integrado ao carro, o teste da leitura do sensor foi satisfatórios apresentando uma simulação com chuva real para o acionamento das paletas do para-brisa de acordo com a quantidade de chuva recebida.

A plataforma foi integrada em um veículo que não tenha tecnologia recente, no caso o veículo de passeio da marca chevrolet modelo corsa sedan 2001, obtendo com êxito com a comunicação entre automóvel e o smartphone, foi construído um hardware como produto que realiza as funções do sistema de alerta do veículo como buzina e farol e do sistema de portas como trava e vidros com um custo bem abaixo do mercado em comparação a outros similares.

9 Considerações finais e Trabalhos Futuros

9.1 Considerações Finais

Dentre as principais contribuições, destaca-se que este trabalho pode servir como modelo para a implementação de outras automações como residencial, comercial e tecnologia assistiva, bastando apenas alterar o roteiro das programações e circuitos.

9.2 Trabalhos futuros

O Hardware integrado automotivo, possui funcionalidades que podem ser incorporadas em trabalhos futuros, tais funcionalidade podem ser úteis para o aperfeiçoamento das funções e da interação entre o usuário e o aplicativo, e estão descritos a seguir:

- Desenvolvimento de um aplicativo próprio para o projeto;
- Integração de mais funções do veículos ao hardware;
- Utilização do produto em automação residencial,comercial e assistiva;

Referências

- ANJOS, E. G. P. dos. Evolucao da eletrônica embarcada na industria automobilística brasileira. *Escola de Engenharia Maua do Centro Universitario do instituto Maua de Tecnologia*, 2011. Disponível em: <<<http://maua.br/files/monografias/a-evolucao-da-eletronica-embarcada-na-industria-automobilistica-brasileira.pdf>>> Acessado em 23/11/2015. Citado 4 vezes nas páginas 10, 11, 12 e 17.
- ARDUINO.CC. *The WebKit Open Source Project*. 2016. Disponível em: <<<https://www.arduino.cc/>>>. Acessado em 11/10/2015. Citado 3 vezes nas páginas 20, 24 e 34.
- ARROYO, A. C.; RUFINO, R. R. Automacao veicular atraves de uma interaracao entre arduino e sistema android. *UNIPAR*, 2015. Disponível em: <<http://web.unipar.br/~seinpar/2015/_include/artigos/Ademir_Conessa_Arroyo.pdf>> Acessado em 20/01/2016. Citado na página 16.
- AZIZ, A. D. Automata theory. *Stanford University*, 2005. Disponível em: <<<https://cs.stanford.edu/people/eroberts/courses/soco/projects/2004-05/automata-theory/index.html>>> Acessado em 11/12/2015. Citado na página 12.
- BARROS, W. R. Sistema de automacao veicular com arduino e android. *UNASP*, 2012. Disponível em: <<<http://www.unasp-ec.com/sistemas/admin/upload/1411505628tcc.pdf>>> Acessado em 15/10/2015. Citado 6 vezes nas páginas 10, 12, 13, 14, 15 e 16.
- C.A, R. Applications, technologies and perspectives. *Rbsoft-CA*, 2003. Disponível em: <<http://www.robosoft.fr/Brochures/wp-transportes_EN.pdf>> Acessado em 03/06/2015. Citado na página 15.
- CHEON, S. *An overview of automated highway systems(AHS) and the social and institutional challenges they face*. 1997. Disponível em: <<<http://uctc.net/research/papers/624.pdf>>>. Acessado em 11/11/2015. Citado na página 13.
- CONSONI, F. L. Da tropicalização ao projeto de veículos: um estudo das competências em desenvolvimento de produtos nas montadoras de automóveis no brasil. *UNICAMP*, 2004. Disponível em: <<<http://www.bv.fapesp.br/en/publicacao/6215/da-tropicalizacao-ao-projeto-de-veiculos-um-estudo-das-compe/>>> Acessado em 15/10/2015. Citado na página 11.
- CORREIA, A. P. Um projeto de controle de movimentacao veicular projetado em um processador embarcado em fpga com ambiente de simulacao usando instrumentacao virtual. *Universidade de Brasilia(UNB), departamento de tecnologia*, 2007. Disponível em: <<<http://www.ppmec.unb.br/downloads/dissertacoes/14-07-2007.pdf>>> Acessado em 11/12/2015. Citado 2 vezes nas páginas 14 e 15.
- COSTA, J. H. R. Implementação de fusão sensorial para localização de um veículo autônomo. *UFMG*, 2013. Disponível em: <<http://coro.cpdee.ufmg.br/attachments/128_PFC20Joao0Henrique.pdf>> Acessado em 1/12/2015. Citado na página 13.
- DUDEK, G.; JENKIN, M. Computational principles of mobile robotics. *Cambridge University Press, Cambridge, UK.*, 2000. Disponível em:

<<https://books.google.com.br/books?hl=pt-BR&lr=&id=db5pr2275i0C&oi=fndpg=PR7&dq=Computational++Principles++of++Mobile++Robotics&ots=_B1N21kY0T&sig=lmllTVlibwTKqHPqVziszAqXwpwv=onepageq=Computational2020Principles2020of2020Mobile2020Roboticsf=false>> Acessado em 11/09/2015. Citado na página 12.

GALILEU, R. *Carros Inteligentes*. 2014. Disponível em: <<<http://revistagalileu.globo.com/Revista/noticia/2014/01/carros-inteligentes.html>>>. Acessado em 19/12/2015. Citado na página 14.

KELBER, C. R. Tecnologias para automacao veicular solucoes em mecatronica e sistemas de apoio ao motorista. *UNISINOS*, 2003. Disponível em: <<<http://osorio.wait4.org/publications/Kelber-Estudos-Tecnol2003.pdf>>> Acessado em 25/10/2015. Citado 3 vezes nas páginas 14, 15 e 16.

KELBER C. R. JUNG, F. H. C. R. Computacao embarcada; projeto e implementacao de veiculos autonomos inteligentes. *XXV Congressoda SBC*, 2005. Disponível em: <<https://www.researchgate.net/publication/242092725_Computacao_Embarcada_Projeto_e_Implementacao_de_Veiculos_Autonomos_Inteligentes>> Acessado em 28/10/2015. Citado na página 12.

LAB, F. G. Ford anuncia a terceira geracao do sync no brasil, que abre uma nova era de conectividade nos veiculos. *Valor Agregado*, 2016. Disponível em: <<<http://valoragregado.com/tag/ford-garage-lab/>>> Acessado em 03/2/2016. Citado na página 13.

MACEDO, J. *O desafio enfrentado pelos veiculos automos*. 2015. Disponível em: <<<http://canaltech.com.br/materia/veiculos/os-desafios-enfrentados-pelos-carros-autonomos-52651/>>>. Acessado em 15/11/2015. Citado na página 14.

OLIVEIRA, M. de. Tecnologia e automação. *Universidade Federal do Espírito Santo(UFES)*, 2013. Disponível em: <<<http://www.lcad.inf.ufes.br/carros.pdf>>> Acessado em 5/1/2016. Citado na página 15.

SABBAGH, V. B. Desenvolvimento de um sistema de controle para um veículo autonomo. *UNIVERSIDADE FEDERAL DE MINAS GERAIS (UFMG)*, 2009. Disponível em: <<http://coro.cpdee.ufmg.br/attachments061_Monografia_Vitor_Brandao_Sabbagh_Versao_Final.pdf>> Acessado em 28/11/2015. Citado na página 13.

SILVA, F. P. P. D. Novas missoes e novas tecnologias: o papel do governo federal e a criacao da darpa na construcao da estrategia de supremacia em ciencia e tecnologia e defesa dos estados unidos na guerra fria. *UNICAMP*, 2014. Disponível em: <<<http://repositorio.unesp.br/bitstream/handle/11449/134095/000857107.pdf?sequence=1&isAllowed=y>>> Acessado em 03/10/2015. Citado na página 12.

SMIDT, A. C. G. Implementacao de uma plataforma robotica controlada remotamente utilizando o arduino. *Universidade de Sao Paulo(USP), Campus Sao Carlos*, 2013. Disponível em: <<[file:///C:/Users/user/Downloads/Smidt_Andre_Crepaldi_Geiger20\(2\).pdf](file:///C:/Users/user/Downloads/Smidt_Andre_Crepaldi_Geiger20(2).pdf)>> Acessado em 05/12/2015. Citado na página 17.

THRUN, S. The robot that won the darpa grand challenge. *Journal of Field Robotics*, 2006. Disponível em: <<<http://isl.ecst.csuchico.edu/DOCS/darpa2005/DARPA20200520Stanley.pdf>>>Acessado em 28/11/2015. Citado na página 12.

Apêndices

APÊNDICE A – Apêndice

A.1 Protótipo Funções

```

1 char serialByte;
2 const int FAROL = 13;
3 const int FUNCIONAR= 7;
4 const int TRAVA = 12;
5 const int VIDROS = 11;
6 const int BUZINA = 10;
7 const int DESTRAVA = 9;
8 const int Z = 8;
9 void setup() {
10 //Coincidir com a taxa de transmiss o do m dulo Bluetooth
11     Serial.begin(9600);
12     pinMode(FAROL, OUTPUT);
13     pinMode(FUNCIONAR, OUTPUT);
14     pinMode(TRAVA, OUTPUT);
15     pinMode(VIDROS, OUTPUT);
16     pinMode(BUZINA, OUTPUT);
17     pinMode(DESTRAVA, OUTPUT);
18     pinMode(Z, OUTPUT);
19
20         Serial.println("1 to turn on the LED on PIN 13");
21     Serial.println("0 to turn off the LED on PIN 13");
22     Serial.println("1 to turn on the LED on PIN 12");
23     Serial.println("0 to turn off the LED on PIN 12");
24     Serial.println("1 to turn on the LED on PIN 11");
25     Serial.println("0 to turn off the LED on PIN 11");
26     Serial.println("1 to turn on the LED on PIN 10");
27     Serial.println("0 to turn off the LED on PIN 10");
28     Serial.println("1 to turn on the LED on PIN 9");
29     Serial.println("0 to turn off the LED on PIN 9");
30     Serial.println("1 to turn on the LED on PIN 8");
31     Serial.println("0 to turn off the LED on PIN 8");
32 }
33 void loop() { // esperar at que os dados est o dispon veis
34     while (!Serial.available());
35     // leia o pr ximo byte dispon vel

```

```
36     serialByte = Serial.read();
37     // Se o byte      um 1 Ligue o FAROL
38     if( serialByte == '1' ) digitalWrite(FAROL, HIGH);
39     else if ( serialByte == '2' )
40     {
41         digitalWrite(FUNCIONAR, HIGH);
42     }
43     else if ( serialByte == '0' )
44     {
45         digitalWrite(FAROL, LOW);
46         digitalWrite(FUNCIONAR, LOW);
47     } // Se o byte      um 0 desligar o FAROL
48 // Se o byte      um 1 Ligue A TRAVA
49     if( serialByte == '1' ) digitalWrite(TRAVA, HIGH);
50     else if ( serialByte == '2' )
51     {
52         digitalWrite(FUNCIONAR, HIGH);
53     }
54     else if ( serialByte == '0' )
55     {
56         digitalWrite(TRAVA, LOW);
57         digitalWrite(FUNCIONAR, LOW);
58     } // Se o byte      um 0 desligar a op    o Trava
59     // Se o byte      um 1 suba os VIDROS
60     if( serialByte == '1' ) digitalWrite(VIDROS, HIGH);
61
62     else if ( serialByte == '2' )
63     {
64         digitalWrite(FUNCIONAR, HIGH);
65     }
66     else if ( serialByte == '0' )
67     {
68         digitalWrite(VIDROS, LOW);
69         digitalWrite(FUNCIONAR, LOW);
70     } // Se o byte      um 0 desligar a op    o vidro
71         // Se o byte      um 1 des a os VIDROS
72     if( serialByte == '1' ) digitalWrite(Z, HIGH);
73
74     else if ( serialByte == '2' )
75     {
76         digitalWrite(FUNCIONAR, HIGH);
77     }
```

```
78     else if ( serialByte == '0' )
79     {
80         digitalWrite(Z, LOW);
81         digitalWrite(FUNCIONAR, LOW);
82     } // Se o byte      um 0 desligar a op      o des a vidros
83
84         // Se o byte      um 1 BUZINE
85         if( serialByte == '1' ) digitalWrite(BUZINA, HIGH);
86
87     else if ( serialByte == '2' )
88     {
89         digitalWrite(FUNCIONAR, HIGH);
90     }
91     else if ( serialByte == '0' )
92     {
93         digitalWrite(BUZINA, LOW);
94         digitalWrite(FUNCIONAR, LOW);
95     } // Se o byte      um 0 desligar a Buzina
96
97         // Se o byte      um 1 DESTRAVA
98         if( serialByte == '1' ) digitalWrite(DESTRAVA, HIGH);
99
100     else if ( serialByte == '2' )
101     {
102         digitalWrite(FUNCIONAR, HIGH);
103     }
104     else if ( serialByte == '0' )
105     {
106         digitalWrite(DESTRAVA, LOW);
107         digitalWrite(FUNCIONAR, LOW);
108     } // Se o byte      um 0 desligar a op      o Destrava
109
110     else
111     {
112         Serial.print(serialByte);
113         Serial.println();
114     }
115     delay(100);
116 }
```


Anexos

ANEXO A – Anexos

A.1 Sensor de Chuva

```

1  int pino_d = 2; //Pino ligado ao D0 do sensor
2  int pino_a = A5; //Pino ligado ao A0 do sensor
3  int val_d = 0; //Armazena o valor lido do pino digital
4  int val_a = 0; //Armazena o valor lido do pino analogico
5  // Portas ligadas aos leds(Velocidades)
6  int pino_led_verde = 5;
7  int pino_led_amarelo = 6;
8  int pino_led_vermelho = 7;
9  void setup()
10 {
11     // Define os pinos do sensor como entrada
12     pinMode(pino_d, INPUT);
13     pinMode(pino_a, INPUT);
14     // Define os pinos dos leds como saida
15     pinMode(pino_led_verde, OUTPUT);
16     pinMode(pino_led_amarelo, OUTPUT);
17     pinMode(pino_led_vermelho, OUTPUT);
18     Serial.begin(9600);
19 }
20 \end{lstlisting}
21 \label{alg:algoritmo}
22 \caption{Algoritmo sensor de chuva Parte 1}
23 \end{verbatim}
24 \end{algorithm}
25
26
27
28 void loop()
29 {
30     //Le e armazena o valor do pino digital
31     val_d = digitalRead(pino_d);
32     //Le e armazena o valor do pino analogico
33     val_a = analogRead(pino_a);
34     //Envia as informacoes para o serial monitor
35     Serial.print("Valor digital : ");

```

```
36 Serial.print(val_d);
37 Serial.print(" - Valor analogico : ");
38 Serial.println(val_a);
39 // Acende o led de acordo com a intensidade
40 if (val_a >900 && val_a <1024)
41 {
42     // Acende led verde - intensidade baixa
43     digitalWrite(pino_led_verde, HIGH);
44     digitalWrite(pino_led_amarelo, LOW);
45     digitalWrite(pino_led_vermelho, LOW);
46 }
47 if (val_a >400 && val_a <900)
48 {
49     // Acende led amarelo - intensidade moderada
50     digitalWrite(pino_led_verde, LOW);
51     digitalWrite(pino_led_amarelo, HIGH);
52     digitalWrite(pino_led_vermelho, LOW);
53 }
54 if (val_a > 0 && val_a <400)
55 {
56     // Acende led vermelho - intensidade alta
57     digitalWrite(pino_led_verde, LOW);
58     digitalWrite(pino_led_amarelo, LOW);
59     digitalWrite(pino_led_vermelho, HIGH);
60 }
61 delay(1000);
62 }
```

A.2 Sensor Ultrassônico

```
1 //Dados do buzzer
2 #define tempo 500
3 int frequencia = 2000;
4 int Pinofalante = 2;
5 int atraso = 1000;
6 //Define o pino do Arduino a ser utilizado com
7 //o pino Trigger do sensor
8 #define PINO_TRIGGER 13
9 //Define o pino do Arduino a ser utilizado com
10 //o pino Echo do sensor
11 #define PINO_ECHO 10
12
```

```
13 //Inicializa o sensor ultrasonico
14 Ultrasonic ultrasonic(PINO_TRIGGER, PINO_ECHO);
15
16 void setup()
17 {
18     pinMode(Pinofalante,OUTPUT); //Pino do buzzer
19     Serial.begin(9600); //Inicializa a serial
20 }
21
22 void loop()
23 {
24     float cmMsec, inMsec;
25
26     //Le os dados do sensor, com o tempo de retorno do sinal
27     long microsec = ultrasonic.timing();
28
29     //Calcula a distancia em centimetros
30     cmMsec = ultrasonic.convert(microsec, Ultrasonic::CM);
31
32     \end{lstlisting}
33 \label{alg:algoritmo1}
34 \caption{Algoritmo sensor de estacionamento Parte I}
35 \end{algorithm}
36
37
38 //Ajusta o atraso de acordo com a distancia
39 if (cmMsec > 80)
40 {
41     atraso = 2000;
42 }
43 else if (cmMsec >50 && cmMsec<80)
44 {
45     atraso = 1500;
46 }
47 else if (cmMsec >30 && cmMsec<50)
48 {
49     atraso = 1200;
50 }
51 else if (cmMsec > 10 && cmMsec < 30)
52 {
53     atraso = 700;
54 }
```

```
55     else if (cmMsec < 10)
56     {
57         atraso = 300;
58     }
59
60     //Apresenta os dados, em centímetros, na Serial
61     Serial.print("Cent: ");
62     Serial.print(cmMsec);
63     Serial.print(" atraso : ");
64     Serial.println(atraso);
65     //Emite o bip
66     NewTone(PinoFalante, frequencia, tempo);
67
68     delay(atraso);
69 }
70 \end{lstlisting}
71 \label{alg:algoritmoopp1}
72 \caption{Algoritmo sensor de estacionamento Parte II}
73 \end{algorithm}
```

A.3 Configuração Bluetooth

```
1 //Carrega a biblioteca SoftwareSerial
2 #include <SoftwareSerial.h>
3
4 //Define os pinos para a serial
5 SoftwareSerial mySerial(10, 11); // RX, TX
6 String command = ""; // Resposta do Dispositivo bluetooth
7         // que permite a entrada de respostas
8
9 void setup()
10 {
11     //Inicia a serial
12     Serial.begin(115200);
13     Serial.println("Digite os comandos AT :");
14     //Inicia a serial configurada nas portas 10 e 11
15     mySerial.begin(38400);
16 }
17
18 void loop()
19 {
20     // Le a sada do dispositivo, se est disponvel.
```

```
21  if (mySerial.available())
22  {
23      while(mySerial.available())
24      { \\Enquanto no h mais o que ser lido, matm a leitura.
25          command += (char)mySerial.read();
26      }
27      Serial.println(command);
28      command = ""; // no repete
29  }
30
31  // Le a entrada do usurio se est disponvel.
32  if (Serial.available())
33  {
34      delay(10); // o DELAY!
35      mySerial.write(Serial.read());
36  }
37 }
```

A.4 Comandos da Configuração Bluetooth

Entre no serial monitor e selecione a velocidade de 9600 na parte inferior (destaque em vermelho), assim como o comando de fim de linha para Ambos NL e CR (destaque em azul). A mensagem Digite os comandos AT :

- AT+ORGL (Reseta o módulo para a configuração padrão)
- AT+RMAAD (remove dispositivos anteriormente pareados)
- AT+ROLE=1 (define o modo de operação do módulo como MASTER)
- AT+RESET (Reset do módulo após a definição do modo de operação)
- AT+CMODE=1 (Permite a conexão a qualquer endereço)
- AT+INQM=0,5,10 (Modo de varredura : padrão, procura por 5 dispositivos ou pára a varredura após 10s)
- AT+PSWD=1234 (define a senha do módulo mestre, que deve ser a mesma do módulo slave/escravo)
- AT+INIT (inicializa o perfil para transmissão/recepção)
- AT+INQ (inicializa a varredura por dispositivos bluetooth)

Digite os seguintes comandos na sequência, cada um deles deve retornar ?OK?, indicando que o comando foi aceito pelo módulo.

A.5 Ascender Led

```
1 char serialByte;
2 const int LEDPIN = 13;
3 const int yellow = 11;
4 void setup()
5 { // Coincidir com a taxa de transmissao do modulo bluetooth.
6     Serial.begin(9600);
7     pinMode(LEDPIN, OUTPUT);
8     pinMode(yellow, OUTPUT);
9     Serial.println("1 to turn on the LED on PIN 13");
10    Serial.println("0 to turn off the LED on PIN 13");
11 }
12 void loop()
13 { // esperar at que os dados estejam disponveis
14     while (!Serial.available());
15     // le proximo byte disponvel
16     serialByte = Serial.read();
17     // Se o byte um 1 Ligue o LED
18     if( serialByte == '1' ) digitalWrite(LEDPIN, HIGH);
19     else if ( serialByte == '2' )
20     {
21         digitalWrite(yellow, HIGH);
22     }
23     else if ( serialByte == '0' )
24     {
25         digitalWrite(LEDPIN, LOW);
26         digitalWrite(yellow, LOW);
27     } // Se o byte um 0 desigue o LED
28     else
29     {
30         Serial.print(serialByte);
31         Serial.println();
32     }
33     delay(100);
34 }
```

A.6 Links Youtube

1-Sensor de Estacionamento no veículo [https : //www.youtube.com/watch?v = DA2VRSw6QeQ](https://www.youtube.com/watch?v=DA2VRSw6QeQ)

2-Acionamento da Trava [https : //www.youtube.com/watch?v = 8Ck-qSu-3UI](https://www.youtube.com/watch?v=8Ck-qSu-3UI)

3-Acionamento do Vidros <https://www.youtube.com/watch?v=vCfPGgVf7M>

4-Acionamento do Farol <https://www.youtube.com/watch?v=dg4ctqbU5k4>

5-Acionamento da Buzina <https://www.youtube.com/watch?v=bhvpLKHXoVk>

A.7 Leitura do sensor de chuva pelo PLX-DAQ e Excel

```
1 int pino_a = A5; //Pino ligado ao A0 do sensor
2 int linha = 0;           // variavel que se refere as linhas do
   excel
3 int LABEL = 1;
4 int valor = 0;
5
6 void setup(){
7   pinMode(pino_a, INPUT);
8   Serial.begin(9600);
9   Serial.println("CLEARDATA");
10  Serial.println("LABEL,Hora,valor,linha");
11 }
12
13 void loop(){
14
15   valor = analogRead(pino_a);
16   linha++;
17
18   Serial.print("DATA,TIME,");
19   Serial.print(valor);
20   Serial.print(",");
21   Serial.println(linha);
22
23   if (linha > 100) //lao para limitar a quantidade de dados
24   {
25     linha = 0;
26     Serial.println("ROW,SET,2"); // alimenta as linhas com os
       dados sempre iniciando
27   }
28   delay(2000); // espera 200 milisegundos
29
30 }
```

A.8 Leitura do sensor de chuva pelo PLX-DAQ e Excel


```
1  /*Pinagem do arduino*/
2
3  //motor A
4  int IN1 = 2 ;
5  int IN2 = 4 ;
6  int velocidadeA = 3;
7  //motor B
8  int IN3 = 6 ;
9  int IN4 = 7 ;
10 int velocidadeB = 5;
11 //variavel auxiliar
12 int velocidade = 0;
13 int d;
14 //Inicializa Pinos
15 void setup(){
16   pinMode(IN1,OUTPUT);
17   pinMode(IN2,OUTPUT);
18   pinMode(IN3,OUTPUT);
19   pinMode(IN4,OUTPUT);
20   pinMode(velocidadeA,OUTPUT);
21   pinMode(velocidadeB,OUTPUT);
22   Serial.begin(9600); //inicia a porta serial
23   pinMode(A4, INPUT); // define o pino 13 como entrada (recebe)
24   pinMode(A2, OUTPUT); // define o pino 12 como saida (envia)
25
26 }
27
28 void loop(){
29
30 /*Exemplo de velocidades no motor A*/
31 //seta o pino 12 com um pulso baixo "LOW" ou desligado ou ainda
32 0
33   digitalWrite(A2, LOW);
34   // delay de 2 microssegundos
35   delayMicroseconds(2);
36   //seta o pino 12 com pulso alto "HIGH" ou ligado ou ainda 1
37   digitalWrite(A2, HIGH);
38   //delay de 10 microssegundos
39   delayMicroseconds(10);
40   //seta o pino 12 com pulso baixo novamente
41   digitalWrite(A2, LOW);
42   //pulseInt l o tempo entre a chamada e o pino entrar em high
```

```
42     long duration = pulseIn(A4,HIGH);
43     //Esse calculo baseado em  $s = v \cdot t$ , lembrando que o tempo
        vem dobrado
44     //porque o tempo de ida e volta do ultrassom
45     long distancia = duration /29 / 2 ;
46
47     Serial.print("Distancia em CM: ");
48     Serial.println(distancia);
49
50
51     if (distancia>300)
52     {
53
54         //Sentido Horario
55         digitalWrite(IN1,HIGH);
56         digitalWrite(IN2,LOW);
57         digitalWrite(IN3,HIGH);
58         digitalWrite(IN4,LOW);
59
60
61         analogWrite(velocidadeA,255);
62         analogWrite(velocidadeB,255);
63         delay(900);
64
65
66
67     }
68
69     else if (distancia<300 && distancia >200)
70     {
71
72         //Sentido Horario
73         digitalWrite(IN1,HIGH);
74         digitalWrite(IN2,LOW);
75         digitalWrite(IN3,HIGH);
76         digitalWrite(IN4,LOW);
77
78
79         analogWrite(velocidadeA,160);
80         analogWrite(velocidadeB,160);
81         delay(900);
82
```

```
83
84
85 }
86
87 else if (distancia<200 && distancia >100)
88 {
89
90 //Sentido Horario
91 digitalWrite(IN1,HIGH);
92 digitalWrite(IN2,LOW);
93 digitalWrite(IN3,HIGH);
94 digitalWrite(IN4,LOW);
95
96
97 analogWrite(velocidadeA,120);
98 analogWrite(velocidadeB,120);
99 delay(900);
100
101
102
103 }
104 else if (distancia<100 && distancia >50)
105 {
106
107 //Sentido Horario
108 digitalWrite(IN1,HIGH);
109 digitalWrite(IN2,LOW);
110 digitalWrite(IN3,HIGH);
111 digitalWrite(IN4,LOW);
112
113
114 analogWrite(velocidadeA,60);
115 analogWrite(velocidadeB,60);
116 delay(900);
117
118 }
119
120 else if (distancia<50 )
121 {
122
123 //Sentido Horario
124 digitalWrite(IN1,HIGH);
```

```

125 digitalWrite(IN2,LOW);
126   digitalWrite(IN3,HIGH);
127 digitalWrite(IN4,LOW);
128
129
130 analogWrite(velocidadeA,0);
131 analogWrite(velocidadeB,0);
132   delay(900);
133 }
134 }
135 \end{lstlisting}
136 \newpage

```

A.9 Dados do CPTEC/INPE Dia 09/04/2016

Apesar de ser do site do CPTEC/INPE mostrar parcialmente nublado, no momento estava chovendo com a umidade relativa do ar em 47 por cento com ventos a 7km. Também é mostrado o provável clima para o dia seguinte, no caso 10/04/2016.

Corumbá-MS



Figura 29 – Condições do tempo na data 09-04-2016

Fonte: <http://www.cptec.inpe.br/cidades/tempo/1645>

A imagem apresenta como está o clima em todo centro-oeste, podendo ser observado pancadas de chuva no sul e oeste do Mato Grosso do Sul.



Figura 30 – Informações sobre o clima no estado do Mato Grosso do sul

Fonte: <http://www.cptec.inpe.br/cidades/tempo/1645>

Meteograma mostra menos de 1mm de precipitação de chuva para a região do Mato Grosso do Sul, apresenta alternância nas temperaturas atingindo pico de 38 graus celsius, umidade relativa até 60 por cento com ventos até 4m(ms)

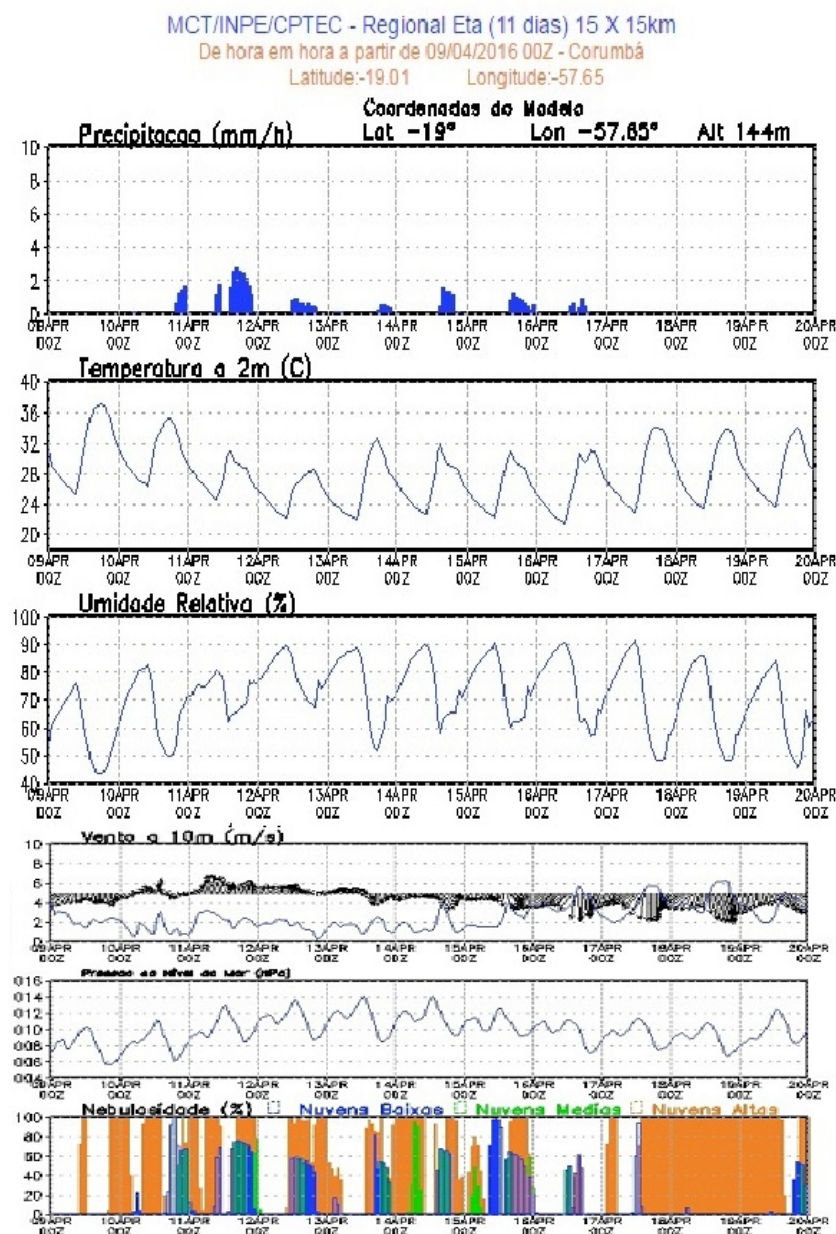


Figura 31 – Gráfico I-Precipitação de chuva, Gráfico II- Temperatura, Gráfico III-Umidade Relativa, Gráfico IV- Vento, Gráfico V-Pressão do nível do mar, Gráfico VI-Nebulosidade

Fonte:<http://www.cptec.inpe.br/cidades/tempo/1645>

A.10 Dados do CPTEC/INPE Dia 10/04/2016

O site do CPTEC/INPE mostra tempo com muitas nuvens e chuvas peródicas, com a umidade relativa do ar em 94 por cento com ventos a 5km.

Corumbá-MS



Figura 32 – Condições do tempo na data 09-04-2016

Fonte: <http://www.cptec.inpe.br/cidades/tempo/1645>

A imagem apresenta como está o clima em todo centro-oeste, podendo ser observado pancadas sol no leste de Goiás, e Mato Grosso do Sul e chuvas nas demais áreas.



Figura 33 – Informações sobre o clima no estado do Mato Grosso do sul

Fonte: <http://www.cptec.inpe.br/cidades/tempo/1645>

Meteograma mostra menos de 1mm de precipitação de chuva para a região do Mato Grosso do Sul, apresenta alternância nas temperaturas atingindo pico de 40 graus celsius,

umidade relativa até 94 por cento com ventos até 2m(ms).

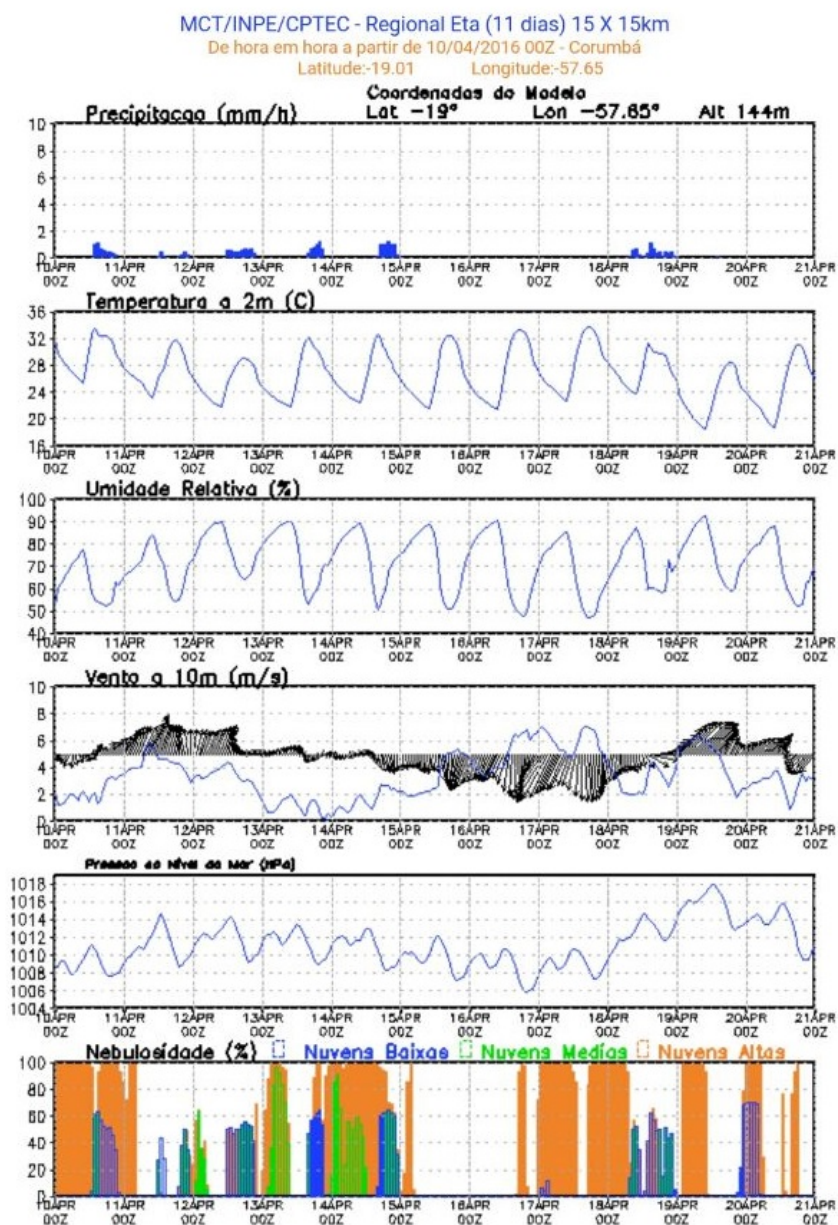


Figura 34 – Gráfico I-Precipitação de chuva, Gráfico II- Temperatura, Gráfico III-Umidade Relativa, Gráfico IV- Vento, Gráfico V-Pressão do nível do mar, Gráfico VI-Nebulosidade

Fonte:<http://www.cptec.inpe.br/cidades/tempo/1645>