



CENTRO UNIVERSITÁRIO DE BRASÍLIA - UniCEUB
CURSO DE ENGENHARIA DE COMPUTAÇÃO

MURILO CABRAL RODRIGUES

SISTEMA DE MONITORAMENTO DE ALARME AUTOMOTIVO
UTILIZANDO SMS

Orientador: Prof. MsC Francisco Javier De Obaldía Díaz

Brasília
Novembro, 2011

MURILO CABRAL RODRIGUES

**SISTEMA DE MONITORAMENTO DE ALARME AUTOMOTIVO
UTILIZANDO SMS**

Trabalho apresentado ao Centro Universitário de Brasília (UniCEUB) como pré-requisito para a obtenção de Certificado de Conclusão de Curso de Engenharia de Computação.

Orientador: Prof. MsC.
Francisco Javier De Obaldía
Díaz

Brasília
Novembro, 2011

MURILO CABRAL RODRIGUES

**SISTEMA DE MONITORAMENTO DE ALARME AUTOMOTIVO
UTILIZANDO SMS**

Trabalho apresentado ao Centro Universitário de Brasília (UniCEUB) como pré-requisito para a obtenção de Certificado de Conclusão de Curso de Engenharia de Computação.

Orientador: Prof. MsC.
Francisco Javier De Obaldía
Díaz

Este Trabalho foi julgado adequado para a obtenção do Título de Engenheiro de Computação, e aprovado em sua forma final pela Faculdade de Tecnologia e Ciências Sociais Aplicadas - FATECS.

Prof. Abiezer Amarilia Fernandez
Coordenador do Curso

Banca Examinadora:

Prof. Francisco Javier De Obaldía Díaz, Mestrado.
Orientador

Prof. Cleber da Silva Pinheiro, Doutorado.
UniCEUB

Prof. Marco Antônio de Oliveira Araujo, Mestrado.
UniCEUB

Prof. Sidney Cerqueira B. dos Santos, Doutorado.
UniCEUB

DEDICATÓRIA**Dedico**

*Aos meus pais, Antonio Paulo e Josefina,
Às minhas irmãs Susana e Samara
Pela força, apoio, incentivo, carinho e atenção,
Que foram importantes para o meu desenvolvimento acadêmico,
E para a conclusão desta etapa tão importante em minha vida.*

AGRADECIMENTOS

A Deus todo poderoso, pela vida.

À minha namorada Alessandra, pela força e pelo carinho em todas as situações.

Ao meu primo Izaías Cabral, pela ajuda e pela atenção que me ajudaram no desenvolvimento deste projeto.

Ao professor MS Francisco Javier De Obaldía, da disciplina Projeto Final, pela orientação e correções que foram necessárias em meu projeto.

Ao professor MS Abiezer Amarília Fernandes, Coordenador Acadêmico do Curso de Engenharia de Computação da FATECS.

Aos professores do Curso de Engenharia de Computação da FATECS.

Ao meu amigo Felipe Souto, pela grande ajuda e suporte e em todo o desenvolvimento do projeto.

Aos meus colegas de Curso pelas grandes amizades que fiz ao longo do curso e que levarei para o resto de minha vida.

Aos funcionários da secretaria da FATECS.

E por fim, àqueles que, de alguma forma, me ajudaram em meu desenvolvimento acadêmico e na conclusão deste projeto.

SUMÁRIO

LISTA DE FIGURAS	VIII
LISTA DE TABELAS	IX
LISTA DE SIGLAS	X
RESUMO	XII
ABSTRACT	XIII
CAPÍTULO 1 – INTRODUÇÃO	14
1.1– Apresentação do Problema	14
1.2 – Objetivos do Trabalho	14
1.3 – Justificativa e Importância do Trabalho	15
1.4 – Escopo do Trabalho	15
1.5 – Resultados Esperados	16
1.6 – Estrutura do Trabalho	16
CAPÍTULO 2 - APRESENTAÇÃO DO PROBLEMA	18
2.1 – Identificação do Problema	18
2.2 – Proposta de Solução do Problema	21
CAPÍTULO 3 – BASES METODOLÓGICAS PARA RESOLUÇÃO DO PROBLEMA	24
3.1 – Sensores Infravermelhos.....	24
3.2 – Buzzer.....	25
3.3 – Microcontroladores.....	28
3.3.1 – Arduino	29
3.4 – Comunicação RS-232	33
3.5 – Comunicador GSM/GPRS.....	35
3.5.1 – Modem G24	35
3.5.2 – SIM Card	36
3.6 – Comandos AT.....	37
3.7 – Rede de Telefonia Celular	40
3.7.1 – SMS	43
3.8 – Linguagem C	44
3.8.1 – Desenvolvimento de uma Aplicação	45
3.8.2 – Linguagem aplicada ao Arduino – DSL.....	45
CAPÍTULO 4 – MODELO PROPOSTO	47
4.1 – Apresentação do Modelo Proposto.....	47
4.2 – Descrição das Etapas do Modelo.....	48
4.2.1 – Montagem do Sensor Infravermelho	49
4.2.2 – Integração dos Dispositivos.....	52

4.3 – Descrição da Implementação.....	55
CAPÍTULO 5 - APLICAÇÃO PRÁTICA DO MODELO PROPOSTO.....	58
5.1 - Apresentação da área de Aplicação do modelo.....	59
5.2 – Descrição da Aplicação do Modelo.....	59
5.2.1 – Teste dos componentes.....	60
5.3 – Resultados da Aplicação do Modelo.....	63
5.4 – Custos do modelo proposto.....	66
5.5 – Avaliação Global do Modelo.....	67
CAPÍTULO 6 – CONCLUSÃO.....	68
6.1 – Conclusões.....	68
6.2 - Sugestões para Trabalhos Futuros.....	69
REFERÊNCIAS BIBLIOGRÁFICAS.....	71
APÊNDICE A – CÓDIGO DO SISTEMA DE MONITORAMENTO.....	74
ANEXOS.....	79

LISTA DE FIGURAS

Figura 2.1 - Esquemático do Projeto.....	23
Figura 3.1 – Detecção por reflexão.....	25
Figura 3.2 – Detecção por interrupção de feixe.....	25
Figura 3.3 – Joy Buzzer.....	26
Figura 3.4 – Modelo do circuito magnético de um buzzer.....	26
Figura 3.5 – Estrutura do buzzer magnético.....	28
Figura 3.6 – Arduino Uno.....	30
Figura 3.7 – Microcontrolador Atmega328p.....	31
Figura 3.8 – Pinagem do microcontrolador Atmega328p	32
Figura 3.9 – Pinagem dos conectores DB9 e DB25, respectivamente.....	34
Figura 3.10 – Local do primeiro SIM Card.....	36
Figura 3.11 – SIM Card da Vivo.....	37
Figura 3.12 – Desenvolvimento das redes celulares.....	41
Figura 4.1 – Fluxograma do Projeto.....	48
Figura 4.2 – Circuito emissor infravermelho.....	49
Figura 4.3 – Variação da frequência em relação à resistência e capacitância.....	50
Figura 4.4 – Pinagem do oscilador NE 555N.....	50
Figura 4.5 – Circuito receptor do Sensor Infravermelho.....	51
Figura 4.6 – Circuito do Sensor de Presença.....	51
Figura 4.7 – Comunicação dos dispositivos através do MAX232.....	53
Figura 4.8 – Ligação do MAX232, buzzer, botão e sensor infravermelho.....	54
Figura 4.9 – Ligações na plataforma Arduino.....	55
Figura 4.10 – Upload do Código no Arduino.....	56
Figura 4.11 – Datagrama do código do Sistema de Monitoramento.....	57
Figura 5.1 – Protótipo do Sistema de Monitoramento.....	58
Figura 5.2 – Seleção da Placa e da porta a ser utilizada.....	61
Figura 5.3 – Exemplo de teste: Blink.....	61
Figura 5.4 – Teste do modem G24.....	62
Figura 5.5 – Resposta do teste do modem G24.....	63
Figura 5.6 – Detecção de presença.....	64
Figura 5.7 – Mensagem de Alerta.....	64
Figura 5.8 – Circuito do Sistema de Monitoramento.....	65

LISTA DE TABELAS

Tabela 1 – Regiões Administrativas líderes em número de roubos e furtos no Distrito Federal.....	18
Tabela 2 – Áreas metropolitanas dos EUA com o maior índice de roubo de veículos...20	
Tabela 3 – Especificações do Buzzer magnético (HDX).....	27
Tabela 4 – Função dos pinos do Conector DB9.....	34
Tabela 5 – Comandos AT comuns.....	37
Tabela 6 – Estimativa de custos do Dispositivo de Monitoramento.....	66

LISTA DE SIGLAS

SMS	Short Message Service
GSM	Global System for Mobile communications
TTL	Transistor-Transistor Logic
RS-232	Recommended Standard 232
SIM	Subscriber Identity Moduler
GPRS	General Packet Radio Service
DSL	Domain Specific Language
GPS	Global Positioning System
MMS	Multimedia Messaging Service
AT	Attention
IR	Infrared
CPU	Central Processing Unit
RAM	Random-Access Memory
ROM	Read-Only Memory
I/O	In/Out (Entrada e Saída)
LED	Light-emitting Diode
USB	Universal Serial Bus
FTDI	Future Technology Devices International
ICSP	In-circuit Serial Programming
PWM	Pulse-width Modulation
GND	Ground
AREF	Tensão de Referência
RISC	Reduced Instruction Set Computing
EEPROM	Electrically-Erasable Programmable Read-Only Memory
MIPS	Milhões de Instruções Por Segundo
IDE	Integrated Development Enviroment
EIA	Electronic Industries Association
DTE	Data Terminal Equipment
DCE	Data Circuit-terminating Equipment
CMOS	Complementary-metal-oxide-semiconductor
CI	Circuito Integrado
EDGE	Enhanced Data rates for Global Evolution
AVL	Automatic Vehicle Location
CSD	Circuit Switched Data
IMSI	International Mobile Subscriber Identity

ANATEL	Agência Nacional de Telecomunicações
1G	1ª Geração
NMT	Nordic Mobile Telephone
AMPS	Advanced Mobile Phone Device
AT&T	American Telephone and Telegraph
2G	2ª Geração
TDMA	Time Division Multiple Access
CDMA	Code Division Multiple Access
2,5G	Geração Dois e meio
1xRTT	1xRadio Transmission Technology
3G	3ª Geração
UMTS	Universal Mobile Telecommunications System
EVDO	Evolution Data Optimized
HSDPA	High-Speed Downlink Packet Access
HSUPA	High-Speed Uplink Packet Access
ANSI	American National Standards Institute
TX	Transmissor
RX	Receptor
ASCII	American Standard Code for Information Interchange
PIC	Peripheral Interface Controller

RESUMO

Este projeto apresenta um dispositivo de monitoramento de alarme automotivo por meio de uma mensagem de alerta via SMS. O projeto tem como principal objetivo desenvolver um sistema agregado ao alarme automotivo que funcione como um adicional na segurança do veículo. Será apresentado no desenvolvimento do projeto um circuito responsável pela simulação do alarme do carro, correspondente ao sensor infravermelho que, ao detectar presença, acionará o buzzer. Em seguida, com este circuito acionado, será feita a comunicação com microcontrolador Atmega328p presente na plataforma Arduino. Esta plataforma fará a comunicação com o modem G24GSM, através de um conversor de sinais TTL para comunicação serial RS-232, o MAX232. Posteriormente, o usuário receberá um SMS em seu celular com uma mensagem de alerta: “Alarme acionado: Favor Verificar!”. Este protótipo mostra uma das inúmeras possibilidades que se pode agregar a um sistema de alarme automotivo, tendo como principal foco a segurança do proprietário do veículo. Além disso, contribui para o desenvolvimento de um projeto utilizando a plataforma Arduino, que têm sido bastante utilizada em projetos acadêmicos devido à facilidade de desenvolvimento e implementação de projetos com esta plataforma.

Palavras Chave: Arduino, Atmega328p, Sensor Infravermelho, Buzzer, MAX232, SMS, Sistema de monitoramento.

ABSTRACT

This project presents an alarm monitoring device for vehicles through an alert message via a SMS. The project's main objective is the development of a system added to the car alarm that works as an improvement on the vehicle safety. It will be presented in this project a simulation of the circuit responsible for the car alarm, which corresponds to the infrared sensor that detects the presence, turning the buzzer on. Then, with this circuit on, there will be communication with the microcontroller Atmega328p, presented in the Arduino platform. This platform will communicate with the modem G24GSM through a TTL signal converter for RS-232 serial communication, the MAX232. After, the user receives a text message on your phone with an alert message: "Alarme acionado: Favor Verificar!" This prototype shows one of the many possibilities which can be added to a car alarm system, with a main focus on security of the vehicle owner. It also contributes to the development and implementation of a project using the Arduino platform, which has been widely used in academic projects due to the facility on developing projects with this platform.

Keywords: Arduino, Atmega328p, Infrared Sensor, Buzzer, MAX232, SMS, monitoring system.

CAPÍTULO 1 – INTRODUÇÃO

1.1 – Apresentação do Problema

A segurança é uma das principais preocupações atuais. Quando se trata de bens com valor agregado, esta questão passa a ter uma importância ainda maior. Os veículos automotivos possuem diversas funcionalidades, tais como computador de bordo, som, GPS, entre outros. Estes acessórios resultam em um objeto de valor bastante elevado, o que desperta a atenção de pessoas mal intencionadas.

O roubo e furto de veículos ou objetos no seu interior podem estar associados a diversos fatores. Algum objeto de valor pode estar à vista no banco do carro ou até mesmo outros acessórios do próprio veículo, como um navegador GPS ou o próprio som do veículo chamam a atenção.

O crescente número de roubos e furtos levou várias empresas a desenvolverem circuitos de alarme com o intuito de evitar este tipo de ação ou, ao menos, criar um obstáculo para o ladrão. Os alarmes automotivos possuem diversas formas de acionamento, podendo ser por sensores infravermelhos (de presença), sensores de pressão ou até mesmo sensores de janela, os chamados detectores de quebra vidro. Estes sensores acionam uma sirene. Contudo, os ladrões continuam achando formas para burlar estes sistemas e obter sucesso no furto do veículo.

Levando em consideração a evolução dos sistemas de telecomunicações e com a tecnologia atual, é possível criar um sistema de alarme mais eficaz?

1.2 – Objetivos do Trabalho

Objetivo Geral – Desenvolver um sistema agregado ao alarme automotivo que exerça a função de monitoramento do veículo através de mensagens SMS.

Objetivo Específico

- Desenvolver o acionamento do buzzer pela plataforma microcontrolador através da detecção de presença do sensor infravermelho;

- Fazer a ligação do modem G24GSM com o microcontrolador através do MAX-232;
- Implementar a programação necessária para a comunicação do microcontrolador com o modem GSM;
- Realizar a comunicação do modem G24GSM com o celular do usuário, através de um SMS de alerta.

1.3 – Justificativa e Importância do Trabalho

Os frequentes roubos e furtos acarretam um sério problema para a população, que é o sentimento de falta de segurança e que gera muita desconfiança. Os sistemas de alarme atuais supriram as necessidades da comunidade por muito tempo. Contudo, é normal que as tecnologias que ficam por um longo tempo no mercado sofram invasões no próprio sistema. Assim, atualmente os ladrões conhecem uma boa parte dos sistemas de alarme, bem como desativá-los.

A evolução da área de telecomunicações aliada ao desenvolvimento tecnológico trouxe muitas inovações nas mais diversas áreas. Lazer, ciência, saúde, entre outras áreas foram amplamente desenvolvidas. Assim como estas áreas, outras questões estão sendo bastante estudadas e implementadas, tais como automação, controle e a área de segurança.

Assim, o desenvolvimento de um novo sistema de segurança automotiva traria benefícios ao usuário. Além disso, o sistema funciona quase que em tempo real, visto que o tempo entre o acionamento do alarme e o recebimento do SMS é mínimo.

1.4 – Escopo do Trabalho

O projeto desenvolvido abrangerá o uso do hardware Arduino, que possui o microcontrolador Atmega328p, e fará a comunicação com o modem GSM/GPRS através de uma rede de telefonia celular. O acionamento do buzzer (que será detalhado no capítulo 3) será feito pelo microcontrolador, que está ligado a um sensor infravermelho que, ao detectar presença, será ativado.

Este sistema de monitoramento possui ainda diversas possibilidades de otimização, como por exemplo, a utilização de uma câmera fotográfica agregada à

plataforma arduino, que acionaria a mesma para que fossem capturadas imagens do interior do veículo no momento em que o alarme é acionado. Além disso, a desativação do alarme por meio de um recebimento de SMS também pode ser implementada.

O projeto não contemplará outros tipos de sensores, tais como sensores de pressão ou sensor de quebra vidro. Também não será desenvolvido o acionamento de uma câmera fotográfica conforme citado anteriormente, visto que o arduino não possui capacidade de processar e armazenar as imagens. Por fim, o projeto não abrangerá a desativação do alarme, no caso do projeto, o buzzer. O desligamento do buzzer será executado por um botão no próprio sistema.

1.5 – Resultados Esperados

O projeto deverá informar ao usuário através de seu celular que o buzzer está disparado e o alerta para verificar. O buzzer será disparado se o sensor infravermelho detectar algum tipo de movimento. Ao detectar a presença, a plataforma Arduino enviará os comandos ao modem, que enviará o SMS.

1.6 – Estrutura do Trabalho

Esta monografia está dividida em seis capítulos que irão abordar sobre o sistema de monitoramento desenvolvido neste projeto.

O primeiro capítulo é a Introdução, onde são definidas as ideias básicas que serão desenvolvidas ao longo do trabalho, bem como o escopo do projeto, o que ele aborda e o que ele não irá tratar. Neste capítulo também são apresentados os resultados esperados para o projeto.

O Capítulo 2 apresentará os problemas que motivaram o desenvolvimento do protótipo através de análises estatísticas da Polícia Civil do Distrito Federal e da *National Insurance Crime Bureau*. Também será apresentada a evolução da tecnologia dentro dos veículos, que são fatores predominantes para sua valorização. Por fim, um breve histórico acerca dos dispositivos de segurança veiculares.

O capítulo 3 irá tratar dos conceitos teóricos dos equipamentos e tecnologias que foram necessárias para o desenvolvimento deste projeto. Este capítulo irá demonstrar os conceitos de um sensor infravermelho, do buzzer, a evolução proporcionada pelo

desenvolvimento de microcontroladores, especificando a plataforma arduino, que possui o microcontrolador Atmega328p. Além disto, também será explicada a comunicação serial RS-232 e as comunicações da rede de telefonia celular, bem como suas tecnologias, tais como o modem G24, que utiliza a rede GSM para a comunicação. Por fim, será apresentado um breve conceito e histórico acerca da linguagem C e o DSL do arduino, que é baseado em C/C++.

O capítulo 4 apresentará um fluxograma com o desenvolvimento do projeto em quatro etapas essenciais. A primeira etapa é definida através da montagem do circuito emissor e do circuito receptor do sensor infravermelho. Posteriormente, temos os testes dos componentes (que serão tratados no capítulo 5). Em seguida, o próximo processo definido neste capítulo trata da integração dos circuitos, definindo a pinagem de cada componente, tais como o sensor de presença, o buzzer e os pinos que farão a comunicação com o modem através do conversor de sinais TTL em comunicação serial RS-232. A última etapa define o modo de gravação do código no Arduino e apresenta um fluxograma da programação implementada no sketch do arduino, finalizando o protótipo desenvolvido no projeto.

O Capítulo 5 demonstra as etapas do funcionamento do dispositivo de segurança, bem como a sua aplicabilidade em algum sistema real. Além disso, pode-se observar neste capítulo a elaboração final do circuito do sistema. Também é apresentado neste capítulo a etapa de testes de componentes, que envolve a utilização de um exemplo de teste da própria IDE do Arduino para verificar o funcionamento da plataforma Arduino, seguido da utilização de comandos AT no terminal Tera Term Web 3.1, para o teste do modem G24. Posteriormente, é realizado um levantamento de custos do projeto. Por fim, são apresentadas as vantagens do uso da plataforma Arduino e a função do sistema, suas limitações e as vantagens e desvantagens do mesmo.

O capítulo 6 irá tratar da conclusão do projeto, demonstrando as vantagens da utilização deste e suas limitações, bem como as maneiras que podem fazer com que o projeto possua um custo reduzido. Além disso, serão apresentadas propostas futuras para o desenvolvimento de sistemas mais robustos, que podem aumentar a segurança do veículo. Além disso, será proposta a utilização do modem no recebimento de SMS, que podem servir como um mecanismo de automatização e controle, tanto em ambientes residenciais, como em ambientes empresariais.

CAPÍTULO 2 - APRESENTAÇÃO DO PROBLEMA

2.1 – Identificação do Problema

Segundo dados de 2011 da Divisão de Cadastro de Roubos e Furtos de Veículos da Polícia Civil do Distrito Federal, a média diária de furtos analisadas no primeiro semestre é de 23 veículos. Ou seja, a cada hora, um carro é roubado. Contudo, cerca de 70% dos veículos são recuperados, sendo que na grande maioria, os veículos estão em boas condições. A razão disso é que os furtos não ocorrem mais com o intuito de desmontar o veículo e revender as peças do mesmo. Atualmente, os bandidos preferem revender o próprio veículo, geralmente na região do entorno, fazendo algumas adulterações (placas, vidros, chassis, motores).

Mesmo com uma média tão alta, este número diminuiu em relação a 2010, quando a média de furtos no mesmo período foi de 25 veículos e em 2009, 29 veículos. Um grande motivo que deve ser levado em consideração para este tipo de crime é a própria topologia do Plano Piloto, região administrativa que lidera o ranking de furtos. A disposição dos prédios, onde há poucas vagas de garagem e até mesmo os prédios públicos da esplanada colaboram para isto. (<http://g1.globo.com/distrito-federal/noticia/2011/09/df-tem-23-roubos-e-furtos-de-carro-por-dia-mas-70-sao-recuperados.html>, 2011)

Podemos observar ainda na tabela 1 a relação das regiões administrativas do Distrito Federal que lideram o ranking de crimes ligados a veículos, seja roubo ou furto. O levantamento dos dados foram realizados no período de janeiro a agosto de 2011.

Tabela 1 – Regiões Administrativas líderes em número de roubos e furtos no Distrito Federal

Ranking	Região Administrativa	Quantidade de roubos	Quantidade de Furtos
1	Plano Piloto	22	145
2	Taguatinga	37	97
3	Ceilândia	39	92
4	Samambaia	23	45

Fonte: (Delegacia de Roubos e Furtos de Veículos, 2011)

Dentro deste contexto, a sociedade desenvolveu uma série de dispositivos de segurança, com o intuito de criar cada vez mais obstáculos contra o bandido. Foram criadas diversas travas, que eram colocadas no volante, na embreagem ou no câmbio. A trava que mais se difundiu foi a Mult-T-Lock. Esta empresa foi criada em 1973 por dois jovens israelenses, Avraham e Moshe Dolev.

Além dos diversos tipos de travas desenvolvidos pela empresa, a divisão automotiva foi a que chamou mais atenção. A trava (de aço de alta resistência) era colocada no câmbio do veículo, fixada no túnel do chassi e com chaves codificadas. (<http://www.mul-t-lock.com>, 2011)

Este tipo de dispositivo de segurança foi muito utilizado. Contudo, era necessário mexer no interior do carro e inseri-lo no chassi do veículo, solução que não agradou a todos.

Com a evolução tecnológica, diversos componentes de um carro foram implementados. A direção deixou de ser mecânica e passou a ser hidráulica (hoje, temos, ainda, câmbios elétricos), que deixou o volante mais leve e melhorou a dirigibilidade; diversas opções de computador de bordo foram desenvolvidas, que informavam a quilometragem do carro, a relação de quilômetros por litro, a distância que poderia ser percorrida com a quantidade de combustível disponível no veículo, hora, calibragem dos pneus, dentre vários outros atributos.

A principal consequência desta série de evoluções foi uma enorme valorização deste bem móvel. Assim, tornou-se ainda mais desejado pelos ladrões. Segundo o *National Insurance Crime Bureau*, em 2004, a cada 26 segundos, um veículo era roubado nos Estados Unidos. A chance de um veículo ser roubado era de um a cada 190. Estes números diminuíram, assim com os do Distrito Federal. De acordo com dados preliminares do *Federal Bureau of Investigation*, o número de furtos de veículos dos Estados Unidos caiu em 7,2% em 2010, marcando o sétimo decréscimo consecutivo anual. Em 2009, o número de roubos de veículos caiu 17% a partir de 2008. Em 2009 um carro era roubado a cada quarenta segundos. Em 2008, as chances de um carro ser roubado tinham uma proporção de um a cada 270 veículos.

A tabela 2 a seguir mostra as cidades dos Estados Unidos com o maior índice de furto de veículos. Este estudo foi realizado pela NIBC em junho de 2011, onde foram realizados os levantamentos estatísticos relacionados ao ano de 2010 nos Estados Unidos. Esta análise também demonstrou que houve um aumento no número de roubos e furtos em relação a 2009 das cinco primeiras áreas metropolitanas do ranking da tabela, enquanto as demais cidades do top 10 apresentaram uma redução do número de incidentes. (<http://www.iii.org/media/hottopics/insurance/test4>, 2011)

Tabela 2 – Áreas metropolitanas dos EUA com o maior índice de roubo de veículos em 2010.

Rank	Estatística da área metropolitana	Veículos Roubados	Taxa (1)
1	Fresno, CA	7,559	812.4
2	Modesto, CA	3,878	753.81
3	Bakersfield, CA	5,623	669.70
4	Spokane, WA	2,673	586.35
5	Vallejo-Fairfield, CA	2,392	578.69
6	Sacramento-Arden-Arcade-Roseville, CA	11,881	552.83
7	Stockton, CA	3,779	551.43
8	Visalia-Porterville, CA	2,409	544.80
9	San Francisco-Oakland-Fremont, CA	22,617	521.68
10	Yakima, WA	1,266	520.49

(1) Taxa de veículos roubados a cada cem mil pessoas.

Fonte: (National Insurance Crime Bureau, 2011)

Uma das maiores contribuições para esta melhora no quadro foi a implementação de alarmes automotivos. A alta diversidade destes dispositivos foi um dos fatores que inibiram os ladrões, visto que o risco de ser pego tornou-se muito maior. Um alarme de carro consiste em um conjunto de sensores unidos a algum tipo de sirene. Os primeiros alarmes eram extremamente simples, e eram acionados somente quando alguma porta era aberta.

Os sistemas de alarme mais atuais possuem uma maior complexidade, que englobam uma série de sensores, que incluem interruptores, sensores de pressão,

sensores quebra vidro, sensores de temperatura e detectores de movimentos. Além disso, possuem uma sirene, um receptor de rádio, uma bateria auxiliar e uma unidade de controle.

No contexto dos sistemas de alarmes automotivos, surgiu um problema no acionamento do alarme. O consumo de energia do alarme ligado por muito tempo leva ao descarregamento da bateria. É comum nos depararmos com alarmes de carros acionados por meios em que não há um furto ou roubo envolvido como, por exemplo, um som alto de um carro que passa por perto, uma pessoa encostada no capô do carro, dentre outras razões. E nem sempre o proprietário do veículo se encontra próximo a ele, deixando o alarme acionado por muito tempo.

Além disso, outro problema que deve ser levado em consideração é a falta de algum sistema de segurança interno, visto que uma vez que o bandido entra no veículo, ele pode desativar o alarme em questão de segundos. Assim, não há um registro sobre o que ocorreu dentro do veículo e nem como ocorreu. Por fim, a identificação da pessoa que invadiu o carro não é revelada também.

2.2 – Proposta de Solução do Problema

A ideia do sistema desenvolvido é a de adicionar um item de segurança ao veículo, sendo assim, mais um obstáculo para o bandido. Através dele, as chances de evitar furtos são maiores. Além disso, caso ocorra algum tipo de roubo ou furto, contribui para resolver o problema com informações sobre o veículo, através da hora do recebimento do SMS. O sistema funcionaria como uma caixa preta do veículo.

A utilização de um sistema auxiliar no interior de um veículo trará muitos benefícios ao usuário, visto que o controle sobre o seu veículo é muito maior. Há diversas alternativas que podem ser acrescentadas a este projeto, tais como o uso de um GPS para definir a localização do veículo a partir do acionamento do alarme, o controle dos eixos do veículo a partir do celular, entre outros.

O uso de um GPS integrado ao sistema poderia trazer a localização exata do veículo após o acionamento do alarme, podendo verificar se o veículo se encontra no lugar correto ou se está em um lugar em que não deveria estar. O controle dos eixos

poderia ser realizado através da inserção de outro celular que captasse os comandos enviados pelo teclado alfanumérico do celular do proprietário, travando as rodas do carro. Já existem projetos desenvolvidos para este tipo de controle com robôs. (www.robotica10.com.br, 2011).

Além disso, outro dispositivo auxiliar que pode ser implementado é o uso de máquinas fotográficas a partir do acionamento do alarme. Posterior ao disparo da máquina ter-se-ia o envio desta imagem via MMS. Contudo, este tipo de componente apresenta um problema de sincronismo, visto que o acesso à imagem não é em tempo real, não caracterizando a situação real do que está acontecendo no veículo no instante da observação. Além disso, é necessária uma câmera fotográfica específica que permita o acionamento através dos mais diversos sensores. Assim, o envio desta imagem por MMS exige um dispositivo eletrônico de maior capacidade de processamento e desempenho que a do arduino ou com processador de sinais específico, o que representa um preço elevado. Sem este componente, não é possível o envio da imagem, visto que o microcontrolador utilizado no projeto não consegue processar imagens.

Este projeto consiste em um sistema de monitoramento acionado a partir do disparo de um buzzer, que é acionado a partir do momento que o sensor infravermelho detecta presença, fechando um circuito que permite que o microcontrolador ligado se comunique com um modem, que por sua vez fará o contato com o celular do destinatário (proprietário do veículo) através de uma mensagem SMS.

A comunicação do Arduino com o modem G24 ocorrerá através do uso de comandos AT, que serão enviados dentro da programação do sketch do Arduino e interpretados pelo modem.

É importante ainda ressaltar que a comunicação entre o microcontrolador e modem ocorrerá através do uso de um conversor de sinais, o MAX232. Este componente é responsável pela conversão de sinais TTL, que é a comunicação utilizada pela plataforma Arduino, em comunicação serial RS232, que é utilizada pelo modem. Com a comunicação implementada, os comandos podem ser enviados normalmente, visto que os transmissores e receptores dos dois componentes irão interpretar a informação. A seguir, a figura 2.1 apresenta como a solução funcionará:



Figura 2.1 - Esquemático do Projeto - Fonte: (Autor do Projeto, 2011)

Por fim, podemos verificar que o sistema desenvolvido irá funcionar como um monitoramento acerca da segurança do veículo. É bom ressaltar que a utilização de um sistema como esse não garante 100% de segurança ao veículo. Porém, serve como um obstáculo a mais para que seja possível impedir qualquer tipo de incidente. Com o auxílio das propostas futuras, será possível criar um sistema muito mais robusto, que ofereça mais opções ao usuário. No capítulo três iremos observar os conceitos que serão necessários para o desenvolvimento do projeto, tanto conceitos de equipamentos como conceitos de comunicação. Também serão apresentados as tecnologias e linguagens utilizadas na implementação do mesmo.

CAPÍTULO 3 – BASES METODOLÓGICAS PARA RESOLUÇÃO DO PROBLEMA

Neste capítulo serão apresentados os conceitos e tecnologias necessárias para o desenvolvimento do projeto. Estes foram apresentados em uma ordem lógica que trata primeiro dos componentes que irão simular o alarme automotivo, depois a plataforma arduino, que é responsável pela comunicação entre todos os dispositivos. Posteriormente, serão tratados os conceitos da comunicação GSM/GPRS e o modem utilizados no projeto, bem como os comandos AT utilizados no modem. Além disso, apresentaremos a evolução da telefonia celular, com um breve histórico de desenvolvimento tecnológico e a tecnologia SMS. Por fim, apresentaremos os conceitos básicos da linguagem C, que é a base do DSL do arduino, utilizado na programação do projeto.

3.1 – Sensores Infravermelhos

Os sensores infravermelhos possuem uma aplicabilidade muito grande, visto que podem ser utilizados de diversas formas, têm baixo custo, fazendo com que qualquer pessoa possa ter acesso a eles. Funcionam, basicamente, através da detecção de presença de pessoas ou objetos em movimento. O sensor possui um conjunto de emissor e receptor. O emissor é responsável pelo envio de sinais infravermelhos, e o receptor detecta estes sinais, se comunicando posteriormente com a central do alarme. No caso deste projeto, o sensor fará a comunicação direta com a sirene e com o microcontrolador. (Fundamentos e Aplicações - Daniel Thomazini & Pedro Urbano Braga De Albuquerque, 2005)

Podemos destacar dois tipos de detecção de sinal: detecção por reflexão e detecção por interrupção do feixe, conforme mostrado nas figuras 3.1 e 3.2, respectivamente. No primeiro, o sinal infravermelho emitido colide com determinado objeto e é refletido, sendo captado pelo receptor. A figura a seguir demonstra esta reflexão.

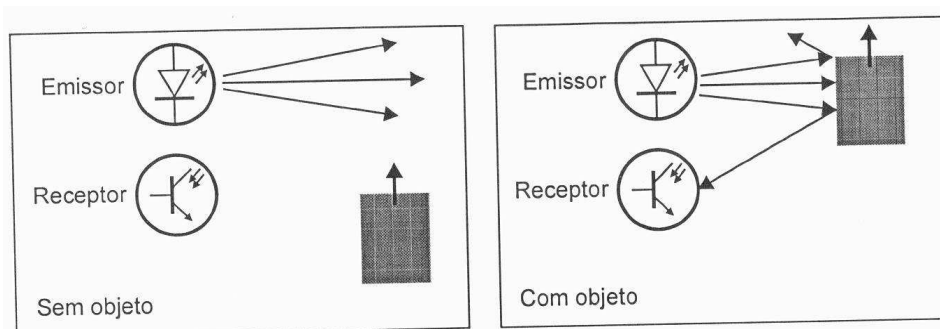


Figura 3.1 – Detecção por reflexão. – Fonte: (Thomazini, Urbano Braga, 2005)

Outro tipo de detecção existente é por interrupção de feixe, em que o emissor envia sinais infravermelhos ao receptor constantemente. O que aciona o sensor é uma interrupção neste fluxo. Este tipo de detecção pode ser observado na figura a seguir.

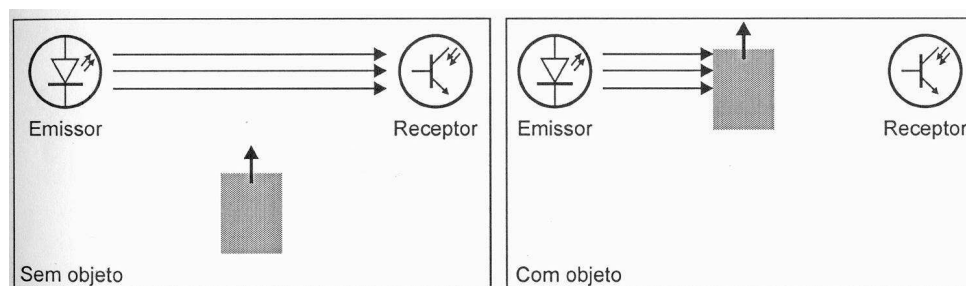


Figura 3.2 – Detecção por interrupção de feixe – Fonte: (Thomazini, Urbano Braga, 2005)

O sensor IR utilizado no projeto utilizou um o método de detecção por reflexão.

3.2 – Buzzer

O buzzer é um dispositivo de sinalização sonora, que normalmente é utilizado em sistema de alarme, timers ou confirmação da entrada do usuário com um clique do mouse ou pressionamento de teclas. O buzzer (ou beeper) pode ser mecânico, o qual ficou popularmente conhecido através do hand buzzer, como mostra a figura 3.3. Este consiste em uma mola dentro de um disco usado na palma da mão. Quando uma pessoa pressiona o botão do buzzer, desencadeia uma vibração inesperada para as pessoas que são interpretadas como um choque.

(<http://en.wikipedia.org/wiki/Buzzer>, 2011)



Figura 3.3 – Joy Buzzer – Fonte: (Ningbo Jojoy International Co. Ltd., 2011)

Outros dois tipos de buzzers produzidos são os magnéticos e os de piezo. Os elementos piezelétricos podem ser conduzidos por um circuito oscilante eletrônico ou fonte de sinal de áudio, impulsionado com um amplificador de áudio piezoelétrico. Em elementos piezoelétricos, a energia elétrica acumulada nos materiais é liberada a partir do stress mecânico, ou seja, é liberada a partir de uma deformação no material.

O buzzer magnético, que é o buzzer utilizado neste projeto, combina o uso da mecânica com o eletromagnetismo. A diferença básica entre um buzzer magnético e um piezelétrico é a ausência do componente piezo, que é responsável pela interação linear entre a mecânica e o estado elétrico dos materiais cristalinos, além de detectar vibrações ou toques.

(<http://www.buzzer-speaker.com/manufacturer/magnetic%20buzzer.htm,2011>).

Na figura 3.4 a seguir podemos observar o circuito de um buzzer magnético:

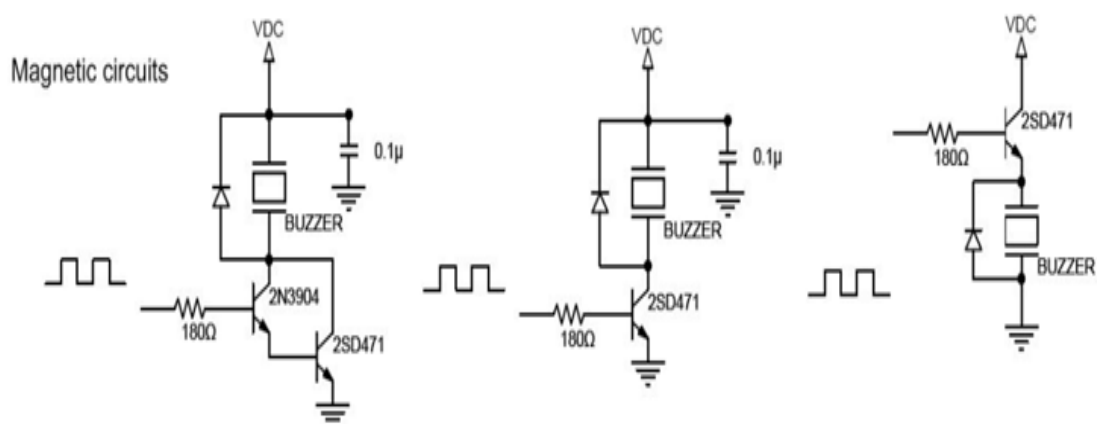


Figura 3.4 – Modelo do circuito magnético de um buzzer – Fonte: (Advanced Acoustic Technology, 2011)

O buzzer magnético possui uma bobina que oscila dentro de uma placa de metal contra outra, que emite som a partir da diferença de tensão com uma frequência pré-definida. Como o buzzer magnético utiliza uma bobina, é gerada uma carga indutiva. Para isso, é necessário proteger o circuito de movimentação, colocando os diodos em paralelo com o buzzer. Podemos observar três tipos de buzzers magnéticos na figura 3.4. variando entre eles a utilização dos transistores. O primeiro utiliza um transistor 2N3904, que possui uma alta velocidade de comutação, uma variação de sinal muito rápida. Além disso, os três modelos ainda utilizam um transistor 2SD471, que se trata de um transistor de silício responsável pela amplificação da frequência do áudio na saída. Podemos observar na tabela 3 as especificações técnicas do buzzer magnético. Além disso, na figura 3.5, temos a estrutura do mesmo.

Tabela 3 – Especificações do Buzzer magnético (HDX)

Especificação	Valores
Tensão	1.5 ~ 12VDC
Tensão Operacional	1.2 ~ 16VDC
Corrente Nominal	≤ 40mA
Corrente nominal	6.5 ± 1.0 45 ± 5 140 ± 14 (Ω)
Impedância da bobina	16 ~ 240Ω
Saída de som	≥ 85DB
Frequência Ressonante	2400HZ
Temperatura de operação	-20 ° C ~ 65 ° C
Temperatura de armazenamento	-20 ° C ~ +70 ° C
Peso	2g
Número do modelo	TMB12C

Fonte: (<http://www.made-in-china.com/showroom/ccc1314cn/product-detailDMXEHpLoJnYe/China-Buzzer-TMB12C-.html>, 2011)

Estrutura

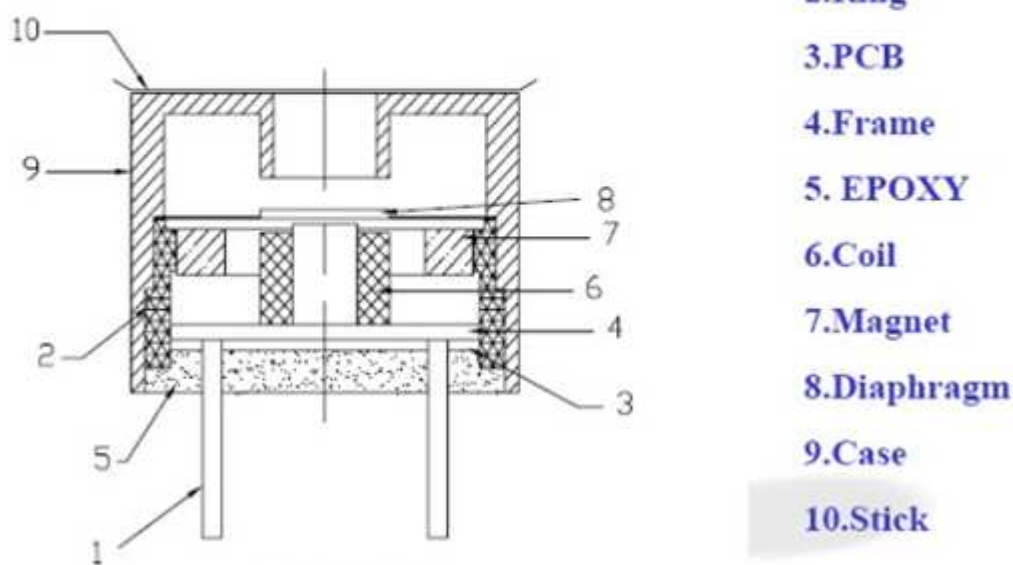


Figura 3.5 – Estrutura do buzzer magnético – Fonte: (Advanced Acoustic Technology, 2011)

3.3 – Microcontroladores

O microcontrolador é um hardware extremamente disseminado no mercado atual. Ele possui uma enorme versatilidade de hardware e um poderoso software. É comum ser denominado de “computador de um chip”. Reúnem neste único chip vários sistemas independentes, como contadores, CPU, memórias RAM, memórias ROM, entre outros. Este tipo de dispositivo é ideal para sistemas compactos e com um fim específico. Além disso, a linha de código deste tipo de dispositivo aliada à sua finalidade é mais difícil de ser violada.

Antes dos microcontroladores, é importante citar e descrever o que é um microprocessador. Este é basicamente um componente eletrônico que realiza inúmeras operações lógicas e aritméticas, sob o controle de um software externo a ele, ponto em que se difere do microcontrolador. Em suma, o microcontrolador é um microprocessador com memória RAM, ROM, entre outros circuitos no mesmo núcleo. É um microprocessador mais completo. O microcontrolador que será utilizado no projeto é o Arduino. (Silva Junior, Vidal Pereira da– Microcontrolares, 1988).

3.3.1 – Arduino

O arduino é uma plataforma de hardware livre. Na verdade, ele não é o microcontrolador em si. A sua arquitetura é a que inclui um microcontrolador, o Atmega328p. O projeto da plataforma arduino foi desenvolvido em 2005 em Ivrea, Itália, com o intuito acadêmico. Em outras palavras, o arduino é um pequeno computador, capaz de ser programado para processar entradas e saídas (através de seu sistema de suporte I/O) entre ele e outros dispositivos externos conectados a ele.

Este hardware pode atuar de diversas formas. Primeiramente, o dispositivo pode ser conectado a outros hardwares, que podem ser implementados com diversos propósitos. O projeto vai englobar o sensor infravermelho, o buzzer e o modem G24GSM da Motorola. Por outro lado, também é possível a comunicação do arduino com um computador, atuando em conjunto com o mesmo, através de troca de dados. Por fim, é possível fazer a comunicação entre todos os sistemas. Neste caso, por exemplo, o arduino recebe dados dos sensores ligados a ele e enviá-los para algum software específico ou até mesmo para algum site.

Os possíveis componentes que podem ser conectados ao arduino englobam sensores de pressão, sensores de presença, sensores de distância, sensores de temperatura, interruptores, motores, módulos Ethernet, LEDs, displays de matriz de pontos ou qualquer outro dispositivo que possa emitir dados ou possa ser controlado.

A arquitetura do arduino possui um suporte a dispositivos de entrada e saída, um cristal ou oscilador, um regulador linear de cinco volts e uma saída USB, que pode ser conectada ao PC. Por fim, os dois componentes mais importantes do arduino são o microprocessador e o microcontrolador. O microprocessador utilizado pelo arduino Uno é o Atmel AVR, modelo Atmega8U2, programado como um conversor USB para serial. A diferença básica da versão Uno para as anteriores é que era utilizado outro chip, como pode-se observar no modelo arduino Duemilanove, que utiliza o chip FTDI. A grande diferença, além do preço do chip (o Atmega8U2 é muito mais barato do que o chip FTDI), é que o Atmega8U2 permite que o chip USB tenha o seu firmware atualizado, o que abre uma série de novas possibilidades para o Arduino. (Michael Roberts - Arduino Básico, 2011).

Podemos observar na figura 3.6 o Arduino Uno com seus respectivos componentes:

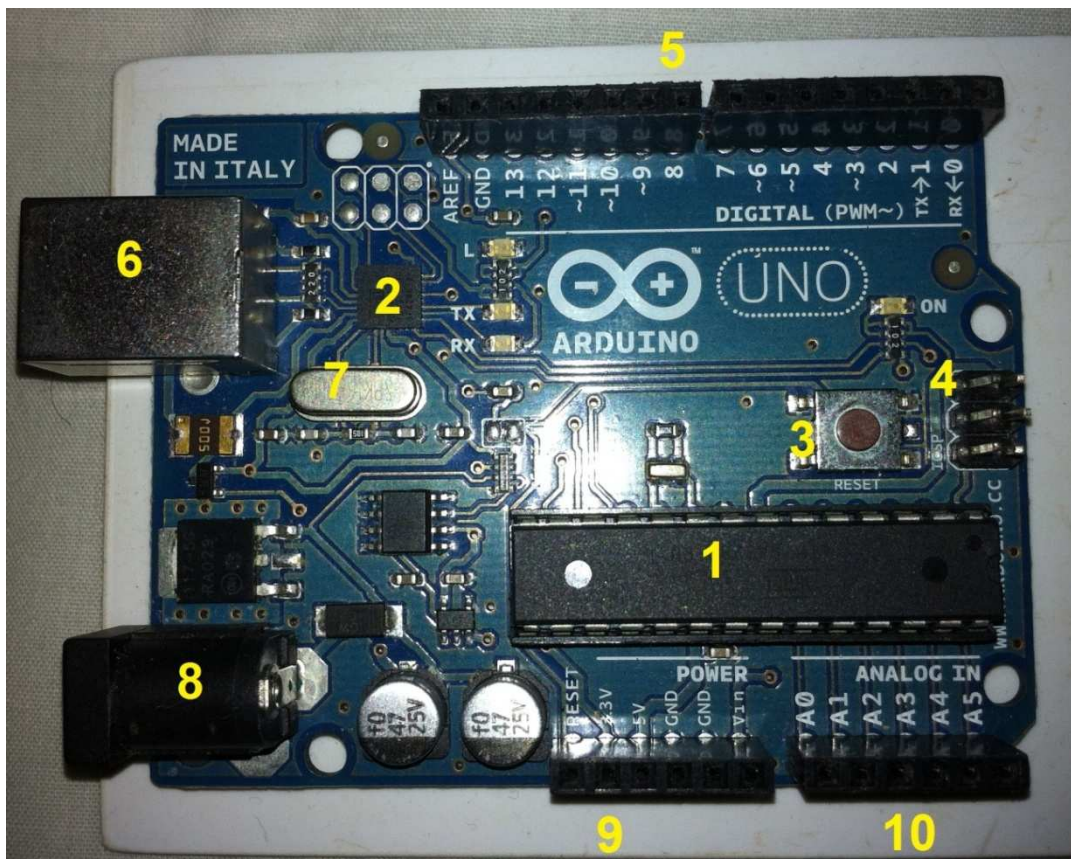


Figura 3.6 – Arduino Uno– Fonte: (Autor do Projeto, 2011)

Os itens presentes na figura 3.6 correspondem:

- 1 – Microcontrolador Atmega328p;
- 2 – Microprocessador Atmega8U2;
- 3 – Botão Reset, responsável por reiniciar o microcontrolador;
- 4 – Conectores ICSP, utilizados em uma programação de um circuito serial;
- 5 – Pinos Digitais, onde os pinos 0 e 1 são os pinos de recepção e transmissão, respectivamente; os pinos que são antecidos por um “~” fornecem 8 bits de saída PWM com a função analogWrite(); Os pinos 2 e 3 ainda podem ser utilizados para ativar interrupções de valor baixo; Os pinos 10, 11, 12 e 13 servem de apoio para uma comunicação utilizando uma biblioteca de SPI; Os pinos “GND” e “AREF” correspondem respectivamente, ao terra e à tensão de referência;
- 6 – Conector USB;

- 7 – Cristal, que oscila a 16MHz e é caracterizada pela estabilidade da frequência e pureza de fase;
- 8 – Conector de alimentação;
- 9 – Pinos de Energia;
- 10 – Pinos Analógicos.

Além do microprocessador Atmega8U2, o microcontrolador utilizado no arduino Uno é o Atmega328p. O alto desempenho deste dispositivo baseado em arquitetura RISC combina 32KB de memória flash com a capacidade leitura e escrita simultâneas, 1KB EEPROM, 2 KB de memória RAM, 23 linhas de I/O, entre outras características. Executando as instruções poderosas em um único ciclo de clock, o dispositivo atinge throughputs, ou seja, uma taxa de transferência efetiva que aproxima-se de 1 MIPS por MHz, equilibrando poder de consumo e velocidade de processamento. Podemos observar na figura 3.7 o Atmega328p. Em seguida, na figura 3.8, a pinagem do mesmo microcontrolador.

(http://www.atmel.com/dyn/products/product_card.asp?part_id=4198, 2011).



Figura 3.7 – Microcontrolador Atmega328p – Fonte: (John Boxall, www.tronixstuff.com, 2010)

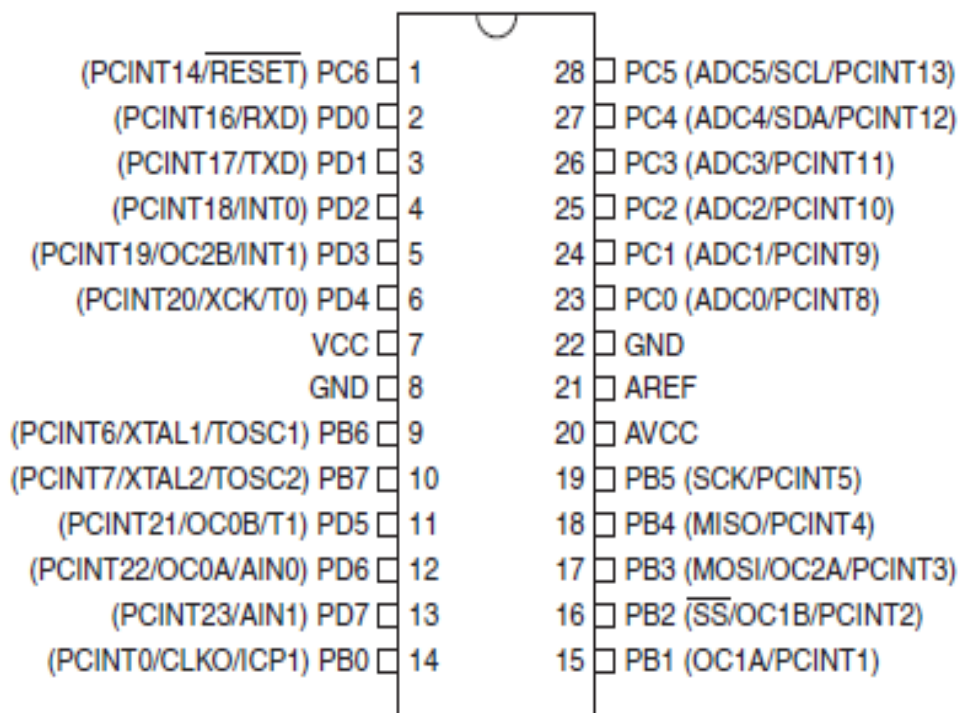


Figura 3.8 – Pinagem do microcontrolador Atmega328p – Fonte: (Atmel, 2011)

Os pinos mostrados acima representam as possíveis entradas e saídas do microcontrolador. Além dos pinos que podem ser utilizados como entradas e saídas, o pino VCC é o responsável pela alimentação do circuito, os pinos GND representam o terra (do inglês *ground*), e os pinos 0 e 1 são o receptor e o transmissor, respectivamente. Estes serão os pinos que serão utilizados no projeto. Os pinos 0 e 1 que serão ligados ao conversor de sinais TTL para comunicação serial RS232, o MAX232. Este último é utilizado para o contato serial com o comunicador GSM/GPRS. (Atmega328p Datasheet, 2010).

Por fim, o Arduino trabalha com uma IDE própria para a sua programação. Esta IDE é um software livre, de fonte aberta, o que significa que os códigos, esquemas e projetos podem ser utilizados por qualquer um. Ela é responsável por receber as linhas de códigos baseados na linguagem C, compilar e fazer o upload para que o Arduino possa interpretar os comandos, interagindo com o que estiver conectado a ele. Estas linhas de código da IDE do Arduino são conhecidos como *sketches*. (www.arduino.cc, 2011).

3.4 – Comunicação RS-232

A comunicação entre o Arduino Uno e o Comunicador GSM/GPRS encontra um obstáculo. Não há um padrão de comunicação entre os dispositivos. O Arduino utiliza uma comunicação TTL, enquanto o modem interpreta sinais de uma comunicação RS-232.

A comunicação RS-232 é um padrão para troca serial de informações. Antes de falar propriamente da interface RS-232 (que atualmente é denominada EIA232), é importante definir o que é a comunicação de dados.

A comunicação de dados é basicamente a transmissão de dados entre os mais variados dispositivos. Desde uma troca de informações internas, entre componentes eletrônicos de uma placa de um circuito, até trocas externas, entre dispositivos, podendo ser entre computadores, entre computadores e outros equipamentos eletrônicos ou entre os mais variados hardwares. A *Electronic Industries Association* (EIA) desenvolveu no início dos anos 60 uma padronização na comunicação de dados via serial: a RS-232 (RS é uma abreviação para *Recommended Standard*). Na época, a comunicação entre os equipamentos para a troca de dados digitais necessitava normalmente de um modem em ambos os dispositivos, justamente para modular ou demodular o sinal. Este padrão criado trouxe especificações padrões, tais como tensões, temporizações, funções de sinais, um protocolo para troca de informações e as conexões mecânicas necessárias. É importante ressaltar que as especificações foram sofrendo algumas alterações ao longo do tempo. (Edmur Canzian - Mini-Curso Comunicação Serial RS-232, CNZ Engenharia e Informática Ltda., 2004)

O RS-232 atua na troca de informações entre dois dispositivos, o primeiro denominado de *Data Terminal Equipment* (DTE), é responsável pelo processamento dos sinais, sendo normalmente um computador. O segundo é o *Data Circuit-terminating Equipment* (DCE), que é o terminal que faz a conexão com o DTE, como por exemplo um modem. (<http://www.lammertbies.nl/comm/cable/RS-232.html>, 2010).

Existem dois tipos de conectores utilizados na interface RS-232: o conector de 25 pinos (DB25) e o conector de nove pinos (DB9). O DB25 foi o conector inicialmente desenvolvido para a interface, possuindo dois canais de comunicação serial RS-232, sendo que um canal era secundário. Na prática, apenas um canal de comunicação serial com handshaking (processo que verifica a disponibilidade de comunicação) é utilizada. São raros os casos em que se utiliza os dois canais, como por exemplo as SparcStation

Sun modelos 10 e 20 e os Dec Alpha Multia. Em computadores pessoais, a menor versão de conector, o DB9, é mais comumente usado atualmente. Podemos observar na figura 3.9 a pinagem dos dois conectores:

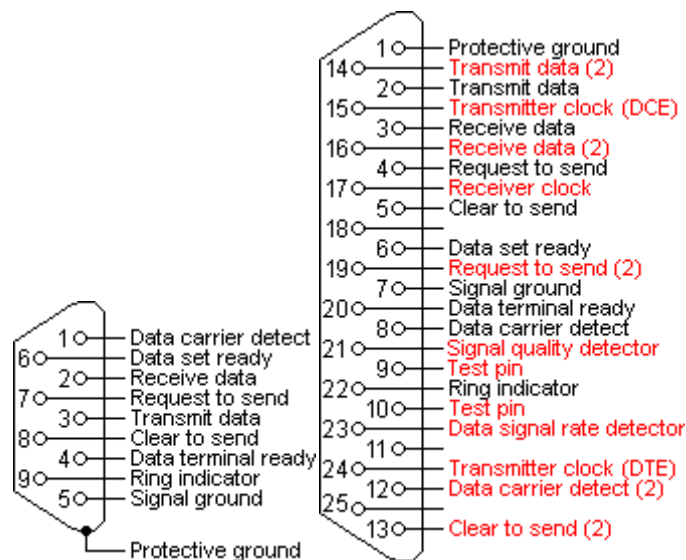


Figura 3.9 – Pinagem dos conectores DB9 e DB25, respectivamente

Fonte: (Lammert Bies, Agosto/2011)

A função dos pinos do conector DB9 podem ser observados na tabela 4:

Tabela 4 – Função dos pinos do Conector DB9

Pino	Sinal	Significado
1	DCD	Portadora do sinal detectada
2	RX	Receptor do sinal
3	TX	Transmissor do sinal
4	DTR	Terminal de dados pronto
5	GND	Sinal da Terra utilizado como referência para outros sinais
6	DSR	Conjunto de dados pronto
7	RTS	Requisição para enviar dados
8	CTS	Enviar os dados
9	RI	Indicador de chamada
-	Shield (capa)	Malha de aterramento do cabo e carcaça do conector

Fonte: (Edmur Canzian, 2011)

A grande maioria dos equipamentos digitais hoje em dia utiliza níveis TTL ou CMOS para a comunicação. Portanto, para comunicar com interfaces RS-232, são necessários os chamados conversores de nível, que fazem a conversão de TTL para RS-232 e vice-versa. Existem diversos circuitos integrados que executam esta conversão, sendo que o mais utilizado é o MAX232 da Maxim. Este CI é capaz de gerar tensões que variam de -10V a 10V a partir de uma fonte de alimentação de 5V. O MAX232 exige o uso de quatro capacitores de 1 μ F cada e possui dois receivers e dois drivers no mesmo encapsulamento. (<http://rogercom.com>, 2011).

3.5 – Comunicador GSM/GPRS

A comunicação do microcontrolador Atmega 328p com o celular do proprietário do veículo somente é possível através de um modem GSM/GPRS, que possui um chip inserido no interior do modem e será responsável pelo envio das mensagens de alerta e pelo recebimento da mensagem de texto que executará o desligamento do buzzer.

3.5.1 – Modem G24

O G24 Serial Data Modem é um dispositivo para transmissão de voz e dados através da tecnologia celular GSM/GPRS-EDGE. Sua cobertura de sinal coincide com o raio de cobertura de uma operadora de telefonia celular. O hardware pode ser usado em várias aplicações como telemetria, segurança AVL, etc. No projeto desenvolvido, o G24 será utilizado para monitoramento do alarme automotivo. Operando por GPRS, é possível transferir dados a uma taxa de até 85,6 Kbps. Operando por EDGE a taxa máxima de transferência é de até 236,8 Kbps. Ambas as opções também permitem transmissão de dados CSD, chamadas de voz, SMS e todas as funções básicas de um telefone móvel, como agenda, transferência de chamadas, etc. (Itech, 2010)

O modem G24 da Motorola tem suporte para comunicação serial RS232 e uma saída USB, não sendo possível a utilização de ambas as portas simultaneamente. Além disso, sua fonte de alimentação varia entre 5VDC e 35VDC, com uma corrente mínima de 500mA. O dispositivo ainda possui uma antena, que deve ser parafusada no conector SMA e entrada para dois SIM Cards, sendo que o primeiro pode ser inserido como

mostra a figura 3.10, e o segundo é inserido depois de abrir a tampa do modem. (Manual Itech do modem G24 Motorola, 2011).



Figura 3.10 – Local do primeiro SIM Card – Fonte: (Manual Itech, 2010)

Por fim, o dispositivo recebe comandos AT, que podem ser testados através do hyperterminal com as seguintes configurações da porta serial:

- 1 start bit
- 1 stop bit
- 8 data bits
- Sem paridade.
- A taxa de transferência da porta serial pode ser configurada entre os valores 300 bps a 115200 bps. O valor padrão é de 9600 bps.

3.5.2 – SIM Card

O SIM Card (ou chip, como é popularmente conhecido) é um circuito integrado que atua como uma memória portátil e que geralmente é utilizado em celulares. O cartão SIM (*Subscriber Identity Module*) foi primeiramente desenvolvido em 1991 através da empresa alemã *Giesecke & Devrient*.

Este circuito integrado possui uma chave denominada IMSI (*International Mobile Subscriber Identity*), que é responsável pela identificação da linha do usuário. Por fim, o SIM possui capacidade para armazenar números, mensagens de texto e qualquer outro tipo de dado (imagem, áudio, etc.). (<http://www.wisegeek.com>, 2003) Podemos observar um chip na figura 3.11:



Figura 3.11 – SIM Card da Vivo – Fonte: (Autor do Projeto, 2011)

3.6 – Comandos AT

Os comandos AT, também conhecidos como *Hayes AT Commands*, foram desenvolvidos em 1977 por Dennis Hayes e implementado no Hayes Smartmodem em 1981. O objetivo destes comandos era criar uma interface para configuração e fornecimento de instruções, como por exemplo, discar um número de telefone.

O objetivo principal destes comandos era fornecer comunicação entre os modems. Contudo, a telefonia celular, através do GSM, passou a adotar este padrão de comunicação, permitindo assim que ações como realização de chamada, transmissão de dados, voz, SMS, entre outros serviços fossem possível. O “AT” é uma abreviatura da palavra *Attention*. (http://en.wikipedia.org/wiki/Hayes_command_set, 2011).

Os serviços que podem ser acionados no modem através dos comandos AT são serviços de chamada, enviar e receber SMS, auto-resposta e verificação de qualidade do sinal. Podemos observar na tabela 5 os comandos AT mais utilizados:

Tabela 5 – Comandos AT comuns

Comandos AT	Resposta/Ação	Comentário
AT	OK/ ERROR	
AT + CPIN?	+CPIN: SIM PIN	Estado do código PIN:

	+CPIN: READY +CPIN: ERROR	-Se a resposta for SIM PIN o código PIN deve ser inserido. -Se a resposta for PIN READY o Modem já esta operacional. -Se a resposta for ERROR não há SIM Card na gaveta ou não está inserido corretamente.
AT+CPIN="xxxx"	OK ERROR	Inserindo o código PIN: -se a resposta for OK o PIN foi inserido corretamente. -se a resposta for ERROR o código PIN foi inserido incorretamente, ele já foi inserido antes ou o SIM Card não esta inserido Corretamente.
AT+CSQ	+CSQ: XXX, 099 (+CSQ: 000, 099 ... +CSQ: 031, 099)	Verifica a qualidade do sinal: o valor XXX indica a intensidade do sinal GSM: -000 indica sem serviço; -031 indicam melhor qualidade de sinal; Se a resposta for 000, verifique se a antena está conectada corretamente, se o SIM Card está inserido ou se o código PIN está correto.
AT+CLCK="SC",0,"xxxx"	OK	Desbloqueia o código PIN. O valor XXXX deve ser o código PIN.
AT+CGPRS	+CGPRS: 0 +CGPRS: 1 +CGPRS: 2	Indica se a cobertura GPRS ou EDGE esta disponível. 0 = nenhum serviço disponível 1 = GPRS disponível. 2 = EDGE disponível.
AT+CGATT?	+CGATT: 0 +CGATT: 1	Indica se o Modem está anexado ou não à rede GPRS.

		<p>PS: não significa que o modem já tenha iniciado uma sessão GPRS.</p>
<p>AT+IPR= "x"</p>	<p>OK</p>	<p>Configura a taxa de transferência na porta serial:</p> <p>x = 0: auto-baud rate</p> <p>x = 1: 600 bps</p> <p>x = 2: 1200 bps</p> <p>x = 3: 2400 bps</p> <p>x = 4: 4800 bps</p> <p>x = 5: 9600 bps</p> <p>x = 6: 19200 bps</p> <p>x = 7: 38400 bps</p> <p>x = 8: 57600 bps</p> <p>x = 9: auto-baud rate.</p> <p>x=10: 115200 bps</p> <p>x=11: 300 bps</p> <p>x=12: 230400 bps</p> <p>x=13: 460800 bps</p> <p>A opção auto-baud rate identifica taxas de até 115200 bps.</p> <p>PS: esse comando configura e salva a taxa de transferência. Se desejar que a taxa não seja salva para a próxima vez que o modem for ligado use o comando: AT+CBAUD= "X" usando os valores acima para X.</p>
<p>ATDxxxx</p>	<p>OK CONNECT NO CARRIER</p>	<p>Faz uma chamada de dados CSD:</p> <p>XXXX é o número de telefone a ser chamado.</p> <p>OK é mostrado se o comando foi aceito;</p>

		CONNECT é mostrado se a chamada foi atendida e o outro lado pôde sincronizar com o G24.
		NO CARRIER é mostrado quando a chamada não pode ser completada ou a chamada caiu.
ATDxxxx;	OK	Faz uma chamada de voz onde XXXX é o número de telefone a ser chamado.
	NO CARRIER	OK é mostrado quando o comando é aceito e novamente quando a chamada foi atendida.
		NO CARRIER é mostrado quando a chamada não pode ser completada ou a chamada caiu.

Fonte: (Manual Itech, 2010)

3.7 – Rede de Telefonia Celular

A rede de telefonia teve início em 1890 e tinha como principal foco a comunicação de sinais de voz. Contudo, com a evolução da informática, a rede passou a exigir a comunicação de dados, o que se tornava inviável com uma rede de sinal analógico. Assim surgiram modificações na rede, que hoje suporta tanto dados quanto sinais analógicos (Behrouz A. Forouzan – Comunicação de dados e redes de computadores, Quarta Edição, 2008).

O laboratório Bell foi o pioneiro em comunicações satélites. Em 1962 ele construiu e lançou com sucesso o primeiro satélite de comunicações em órbita (Telstar I). O Telstar foi o único satélite que tinha a capacidade de receber um sinal, amplificá-lo, e então transmitir de volta para outro lugar na Terra, que é basicamente o que um núcleo de um satélite de comunicações faz. Esta tecnologia permitiu que as chamadas telefônicas fossem realizadas para qualquer lugar do mundo.

A rede de telefonia celular começou a ser desenvolvida em 1947, através dos laboratórios Bell, que foi a primeira empresa que propôs uma rede celular. A inovação principal foi o desenvolvimento de uma rede de pequenas células sobrepostas apoiada

por uma infra-estrutura de comutação de chamadas que rastreia os usuários na medida em que eles se moviam através de uma rede e passar a sua chamada a partir de um local para outro sem deixar cair a ligação. Em 1970, os laboratórios Bell instalaram em Chicago a primeira rede comercial de celular. (<http://www.alcatel-lucent.com>, 2011)

No Brasil, podemos observar a densidade do serviço móvel celular através do gráfico apresentado na figura 3.12:

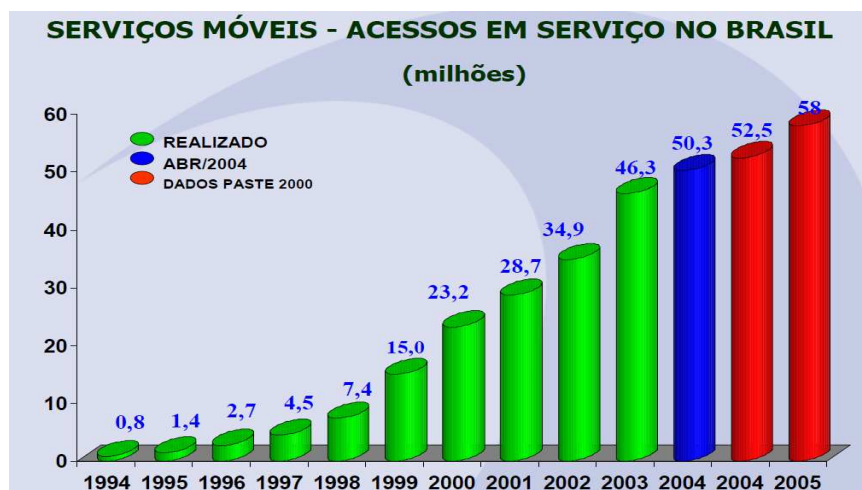


Figura 3.12 – Desenvolvimento das redes celulares – Fonte: (Anatel, 2005)

A rede de telefonia celular se encontra hoje em sua quarta geração, sendo que no Brasil ainda é a terceira geração. Estas fases de evolução são definidas pelos meios de transmissão das ondas eletromagnéticas para a troca de dados.

A primeira geração (1G) contava com transmissão de dados analógica, baseada na compressão ou na distribuição das suas informações. Esta contava com dois sistemas: o NMT e o AMPS. O AMPS (*Advanced Mobile Phone Device*) era o sistema mais utilizado e foi desenvolvido pelos Laboratórios Bell da AT&T no início dos anos 80. A transmissão ocorria em uma faixa de 800MHz e só era capaz de transmitir voz. Em junho de 2006 esta rede foi desativada, para que se pudessem utilizar outros sistemas mais avançados.

Os sistemas da segunda geração (2G) contavam com uma inovação em relação à transmissão anterior: a transmissão digital. Esta transmissão permitia codificação digital de voz mais poderosa, maior eficiência espectral, melhor qualidade de voz, facilidade a comunicação de dados e a criptografia. Nessa geração, começam a se formar três grupos evolutivos: o TDMA, o CDMA e o GSM. O TDMA (*Time Division Multiple Access*) opera em 850MHz e divide os canais de frequência em até seis intervalos de tempo diferentes, onde cada usuário utiliza um espaço específico, evitando problemas de

interferências. Já o CDMA (*Code Division Multiple Access*) permite múltiplos acessos simultâneos em um único canal de estação rádio-base. Este sistema opera com frequências que variam de 850 a 1900MHz. A transmissão GSM (*Global System for Mobile Communication*) possui uma diferença básica em relação aos outros sistemas: a utilização de um SIM Card (chip), que possibilita a portabilidade das características da linha. O GSM opera nas faixas de 400, 450, 850, 900, 1800 e 1900 MHz.

A geração 2,5G conta com melhorias significativas em capacidade de transmissão de dados e na adoção da tecnologia de pacotes e não mais comutação de circuitos. O GPRS (*General Packet Radio Service*) é a evolução da tecnologia GSM. Essa tecnologia oferece velocidades máximas de dados de 115 kbps e um throughput médio de 30 a 40 kbps. Os dados são divididos em pacotes para transmissão, o que favorece os usuários, já que fornece uma conexão permanente de dados, fazendo com que os usuários não necessitem entrar no sistema cada vez que desejarem ter acesso a serviços de dados. Outra vantagem é que os usuários só pagam pelos dados e não pagam pelo tempo de permanência no ar em que se faz a conexão e nem pelo tempo de carregamento. É o GPRS que permite a conexão da maior parte dos smartphones e celulares à internet. Atualmente, o GPRS é o padrão que oferece a maior cobertura móvel para aparelhos de mão com acesso à internet. A Oi, TIM, Claro, Brasil Telecom GSM, CTBC e Vivo, já operam no Brasil com sistemas GSM/GPRS.

Além disso, a geração 2,5G também opera com o sistema EDGE (*Enhanced Data Rates for Global Evolution*), que é uma tecnologia de transmissão de dados e acesso à Internet de alta velocidade. Por último, temos o sistema 1xRTT (*1xRadio Transmission Technology*). Ele possui taxas de transmissão superiores a 144kbps, que oferecem aos consumidores e profissionais total conectividade sem fio.

Os sistemas 3G começaram a ser desenvolvidos no final dos anos 90, e a grande vantagem em relação às outras tecnologias é a sua velocidade de transmissão de dados. O padrão 3G especifica, mais exatamente, 144kbps em ambientes móveis, 384 kbps em ambientes de pedestres e 2 Mbps em ambientes fixos. As principais tecnologias desta geração são o UMTS (*Universal Mobile Telecommunications Service*) e o EVDO (*Evolution Data-Optimized*). O primeiro é a evolução do GSM, mas que ainda se baseia nessa tecnologia, embora o seu acesso por rádio seja diferente. Opera principalmente em 2100MHZ, mas em algumas regiões opera em 850MHz ou 1900MHZ e mais recentemente em 1700mhz. Além disso, é compatível com a EDGE e a GPRS permitindo ao usuário sair de uma área de cobertura UMTS e ser automaticamente

transferido para uma rede EDGE ou GPRS, dependendo de fatores como disponibilidade de rede e o consumo de banda do seu aplicativo. Já o EVDO é a tecnologia 3G do CDMA, que possui alto desempenho para transmissão de dados com picos de até 2,4 Mbps. São necessárias portadoras de sinal distintas para dados e voz neste sistema. A taxa de transmissão de dados teórica é de 2.4mbps e taxa de transmissão média de 300 a 500 kbit/s. Por fim, o EVDO opera em 800 e 1900MHz.

Ainda temos a 3,5G, que embarcam os sistemas HSDPA (*High Speed Downlink Packet Access*) e o HSUPA (*High Speed Uplink Packet Access*). Estes dois sistemas são responsáveis por todas as atividades realizadas pelos smartphones hoje em dia, tais como e-mail, jogos online, acesso a internet e redes sociais, download de arquivos, etc. Ambas tecnologias possuem taxas de transferência de até 14,4Mbps/s em uma banda de 5MHz. (<http://www.fundacaotelefonica.org.br/Arte-e-Tecnologia/Memoria-Telefonica-Pesquisa.aspx>, 2011).

3.7.1 – SMS

O serviço de mensagem curta (*Short Message Service*) surgiu por volta dos anos 80. A *CEPT group GSM* aprovou em 1982 o serviço de SMS para a utilização em dispositivos móveis. Já em 1984, o conceito de SMS foi definido pela *Franco-German GSM cooperation através de Friedhelm Hillebrand e Bernard Ghillebaert*. O SMS foi definido como um padrão GSM móvel para troca de mensagens curtas. Esta mensagem possui um tamanho limitado a 160 caracteres, que incluem letras, números e combinações alfanuméricas. As informações destas mensagens passam por uma central SMS da operadora, que é responsável pelo gerenciamento das mesmas. (<http://wataniya.yahoo.com/smshistory.html,2003>).

Os serviços de SMS ganharam uma importância muito grande no mundo das telecomunicações. Hoje em dia, há diferentes formas de envio que ampliam o tamanho das mensagens. Podemos destacar os SMS múltiplos, que basicamente são uma concatenação de mensagens curtas, permitindo o uso de mais de 160 caracteres. Além disso, é possível utilizar os SMS de compressão, que também permitem mais de 160 caracteres, comprimindo a mensagem em um único SMS.

Neste trabalho é utilizado o modem G24 Serial Data Modem, que é um dispositivo para transmissão de voz e dados através da tecnologia celular GSM/GPRS-

EDGE, abrangendo a segunda geração e a geração 2,5. Operando por GPRS é possível transferir dados a uma taxa de até 85,6 Kbps. Operando por EDGE a taxa máxima de transferência é de até 236,8 Kbps. Ambas as opções também permitem transmissão de dados CSD, chamadas de voz, SMS e todas as funções básicas de um telefone móvel.

3.8 – Linguagem C

A linguagem C foi desenvolvida em 1972 nos *Bell Telephone Laboratories* por *Dennis Ritchie*, com o intuito de permitir a escrita de um sistema operacional (Unix), utilizando uma linguagem mais alto nível do que era o Assembly. O nome da linguagem deriva de uma evolução da linguagem B, também desenvolvida nos laboratórios Bell por Ken Thompson. Apesar de ser uma linguagem da terceira geração, a linguagem C possui um nome destoante em relação às outras linguagens, tais como o FORTRAN, PASCAL ou COBOL.

Esta linguagem, por ter uma capacidade muito grande e ser a linguagem do sistema Unix, disseminou-se por todos os tipos de programadores, o que levou a problemas de portabilidade. Assim, em 1983 o *American National Standards Institute* (ANSI) formou um comitê para criar padrões para a linguagem C.

As oportunidades oferecidas pela linguagem C são o que fazem dela extremamente utilizada até hoje, justamente pode ser extremamente potente e flexível.

Diferentemente do PASCAL e do COBOL, a linguagem C consegue performances semelhantes à do Assembly, visto que utiliza instruções de alto nível para usar mecanismos de nível mais baixo. Além disso, sua sintaxe é bastante simples, e o número de palavras reservadas, tipos de dados básicos e de operadores é menor.

A modulação permite que o C faça uma separação de projetos em módulos distintos e independentes, recorrendo à utilização de funções específicas dentro de cada módulo. Outra característica que torna esta linguagem muito utilizada é o fato de possuir muitas bibliotecas que possuem as mais diversas capacidades de processamento. A maior parte das funcionalidades desta linguagem é adicionada pela utilização de funções que existem em bibliotecas adicionais e realizam todo tipo de tarefas.

Por último, podemos destacar a utilização de macros, que reduz a necessidade de escrita de funções distintas para a realização do mesmo processamento para tipos de dados diferentes. Elas aumentam a velocidade de execução sem ter que aumentar a

complexidade de escrita do código. Assim, a linguagem C se tornou referência para as mais diversas linguagens desenvolvidas posteriormente, tais como C++ e até o próprio JAVA.

A estruturação da linguagem C possui uma filosofia inspirada no Unix, ou seja, o sistema é subdividido em vários códigos que executam uma função específica, o que diminui a probabilidade de uma deficiência no sistema. Assim, um projeto deve ser dividido em vários módulos, no qual cada um é implementado de forma independente do outro. (Dama Luis – Linguagem C, 2007).

3.8.1 – Desenvolvimento de uma Aplicação

Podemos distinguir quatro fases distintas na implementação de um código da linguagem C: Edição do Código-Fonte, Compilação do Programa, “Linkagem” dos objetos e Execução do Programa.

A primeira fase trata do trabalho do programador em criar o código-fonte em si, salvando o arquivo em extensões **.c**. Em seguida, a compilação do programa é responsável pela verificação do código, procurando por erros de escrita ou até mesmo erros de sintaxe. Se não houver nenhum tipo de erro, o compilador cria um arquivo de mesmo nome do código com a extensão **.obj**. Após a compilação, o *linker* é responsável por criar o arquivo executável (.exe), que engloba o arquivo .obj e os códigos compilados das bibliotecas. Por fim, após a linkagem do código, a última fase do ciclo do desenvolvimento de uma aplicação é a sua própria execução do arquivo executável. Em geral, os ciclos de todas as linguagens se assemelham. Contudo, o grande diferencial da linguagem C e de seus descendentes é justamente esta fase de compilação, que tem por missão a expansão das macros e a execução de todas as linhas de código para o pré-processador. (Dama Luis – Linguagem C, 2007).

3.8.2 – Linguagem aplicada ao Arduino – DSL

O hardware Arduino trabalha com uma linguagem que é baseada na linguagem C/C++, tratando-se de um DSL (*Domain Specific Language*), que trabalha com diversos atalhos e conseqüentemente gera facilidade na programação das portas digitais, analógicas e outros recursos do microcontrolador. Os DSL são linguagens de

programação baseadas em outras linguagens, mas que possuem uma característica específica, que podem trazer benefícios, tais como a otimização da resolução de um problema e o tempo de aprendizado da linguagem mais rápido (www.arduino.cc, 2011).

O compilador utilizado para a programação desta DSL é a própria IDE do arduino, um software livre que é disponibilizado no site. O sketch (compilador) utilizado permite a programação do DSL e o seu upload, gravando a informação desenvolvida na plataforma arduino escolhida no sketch.

Assim, definiram-se todos os conceitos e tecnologias necessárias para o desenvolvimento do dispositivo de monitoramento utilizando SMS. Observou-se no mesmo o funcionamento dos dispositivos utilizados no projeto, bem como os componentes que os compõem. Assim, será abordado no próximo capítulo o desenvolvimento do projeto, a partir de sua implementação para a obtenção do produto final de estudo.

CAPÍTULO 4 – MODELO PROPOSTO

O modelo deste projeto propõe um sistema de monitoramento de um veículo através do acionamento do alarme (neste caso, utilizou-se um sensor infravermelho e um buzzer para representar o circuito do alarme). A partir do alarme ativado, será emitido um SMS para o celular do proprietário do veículo, informando que o alarme está tocando.

4.1 – Apresentação do Modelo Proposto

O sistema de monitoramento proposto funcionará a partir de uma detecção de presença do sensor infravermelho, através de uma reflexão da luz infravermelha em uma frequência de 33 KHz, que será emitida por um LED infravermelho através de uma oscilação no pino 3 do circuito integrado NE-555. Ao se deparar com algum objeto, a luz infravermelha será refletida e captada pelo circuito receptor, através do TSOP 2233. Com isso, o circuito enviará o sinal de detecção para o arduino, que acionará o buzzer através da função “digitalWrite”, que irá setar o buzzer como HIGH e LOW em um loop, com intervalos de tempos definidos entre eles, simulando assim um alarme.

A partir do acionamento do alarme simulado, a plataforma Arduino enviará os comandos AT por meio de uma função criada “envia_sms” (esta função será detalhada na programação apresentada neste capítulo e no Apêndice A) para o modem, que irá receber os comandos e enviar o SMS para o celular do usuário. Através do sketch do Arduino, será realizada a programação que será responsável pela comunicação entre os componentes. Além disso, será necessário o uso de um SIM Card no modem utilizado.

A primeira fase do projeto contempla a montagem do circuito do sensor infravermelho. Posteriormente, serão feitos os testes para verificar o funcionamento individual dos componentes (esta etapa de testes será detalhada no Capítulo 5). O Arduino será testado através de seu compilador próprio, *sketch*. Em seguida, será realizado o teste com o modem, utilizando o terminal Tera Term Web 3.1 para interpretar os comandos AT utilizados. A seguir, testaremos o funcionamento do sensor infravermelho e do buzzer conectado ao arduino, que estará ligado ao computador para alimentação. Esta etapa de testes coincide com a montagem do circuito em si.

Com os circuitos testados, os mesmos serão implementados em conjunto para o resultado final do sistema de monitoramento. Com o auxílio de um protoboard, serão feitas as ligações entre o sensor infravermelho, o buzzer, o arduino e o modem G24GSM. Por fim, a programação definirá o funcionamento do sistema. Podemos observar na figura 4.1 o fluxograma do desenvolvimento do projeto.

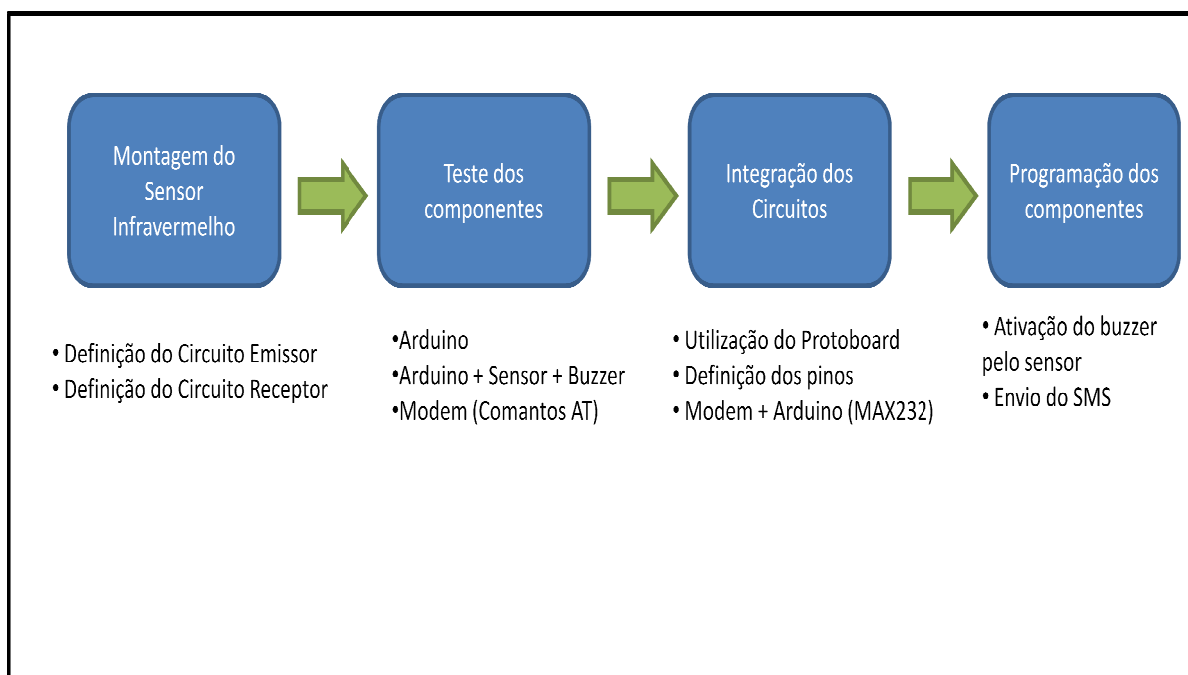


Figura 4.1 – Fluxograma do Projeto – Fonte: (Autor do Projeto, 2011)

4.2 – Descrição das Etapas do Modelo

O projeto do sistema de monitoramento foi dividido em quatro etapas principais: a montagem do sensor infravermelho, o teste dos componentes utilizados no sistema (detalhada no Capítulo 5), a integração deles e, por fim, a programação do sistema de monitoramento. Os testes foram importantes para que fosse possível verificar o funcionamento de cada dispositivo separadamente. O desenvolvimento de cada etapa somente foi possível após a conclusão dos testes de cada componente.

4.2.1 – Montagem do Sensor Infravermelho

O projeto utiliza um circuito emissor e um circuito receptor infravermelho que trabalha com detecção por reflexão. Assim que um objeto entrar no campo de emissão de raios infravermelhos e, conseqüentemente, interromper o fluxo, o buzzer é acionado. Podemos observar na figura 4.2 o circuito do emissor infravermelho.

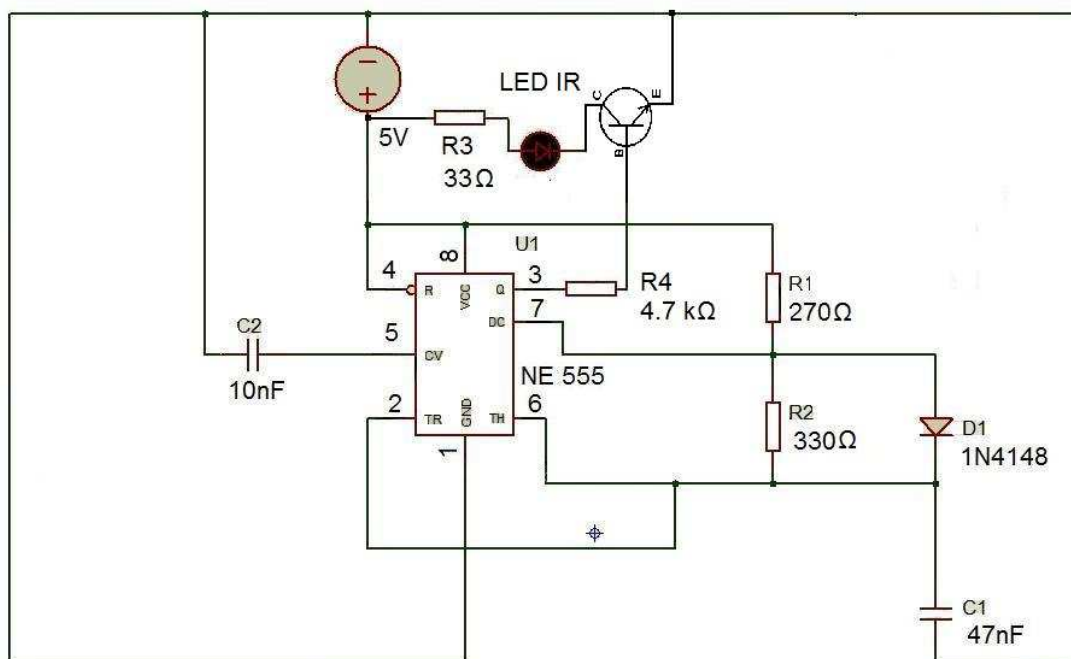


Figura 4.2 – Circuito emissor infravermelho – (Fonte: Luciana Mello, 2011)

O circuito infravermelho emissor utilizou um circuito integrado temporizador NE555N, configurado em modo estável e que gera uma oscilação no pino três de acordo com os resistores R1, R2 e capacitor C1 dado pela fórmula:

$$F = \frac{1}{T} = \frac{1.44}{[(R1 + 2 * R2) * C1]}$$

Onde F é representada pela frequência que será levada em consideração e os resistores associados ao capacitor serão responsáveis pelo cálculo da frequência, que será de 33 KHz para que o receptor possa detectar. Podemos observar a variação da frequência em relação à resistência e capacitância no gráfico da figura 4.3 abaixo:

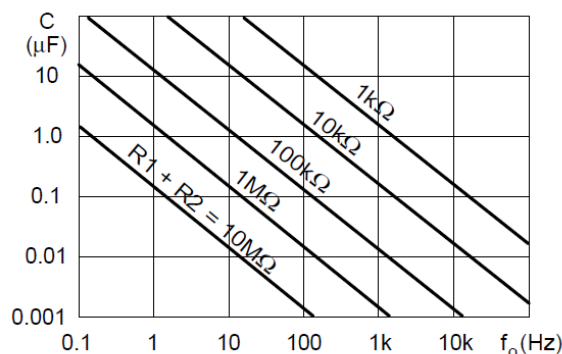


Figura 4.3 – Variação da frequência em relação à resistência e capacitância – Fonte: (Datasheet NE555N, 1998)

Pode-se observar ainda que o capacitor C2 tem como função o controle de tensão. O resistor R3 ajusta a potência do emissor infravermelho definindo a distância que o sensor detecta o objeto. Por fim, a saída do pino três permite a passagem de corrente através do transistor fazendo com que o LED infravermelho oscile na frequência de 33 KHz. A figura 4.4 ilustra a pinagem do oscilador.

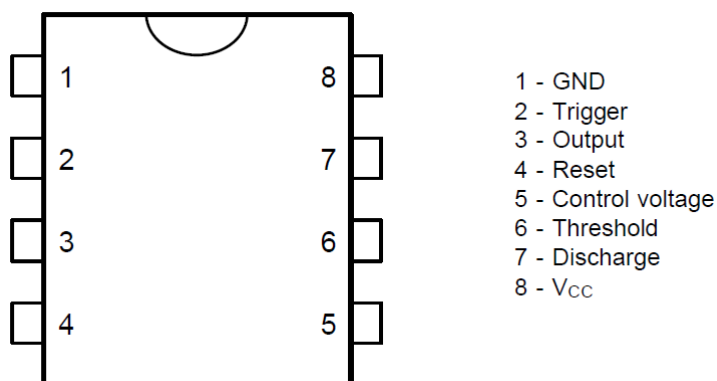


Figura 4.4 – Pinagem do oscilador NE 555N - Fonte: (Datasheet NE 555N, 1998)

Definido os parâmetros do oscilador, outro componente utilizado no sensor infravermelho é o TSOP 2233, que se trata de um receptor de luz infravermelha. O circuito receptor pode ser observado na figura 4.5, que é composto pelo TSOP 2233, que detecta luz infravermelha na faixa de 33 KHz e um led para verificação da presença. O capacitor utilizado no circuito tem o intuito de diminuir ruídos da fonte. O resistor de 220 Ohms é necessário para que o LED se acenda a uma voltagem de 3.3 volts.

Quando não há objeto detectado pelo TSOP 2233, a tensão de saída no pino 3 é de 3.3v. Assim, o catodo do LED está alimentado com 3,3 volts. O resistor 220 Ohms

ocasiona uma queda de tensão de 1.7 volts dos 5 volts que são alimentados pela ligação com o Arduino (que é alimentado por uma fonte de 5v) e fornece 3,3 volts para o anodo do LED. Assim não há diferença de potencial, não acendendo o LED. Quando a frequência definida é detectada, o TSOP 2233 abaixa o nível para 0 volts no catodo e o anodo permanece com o mesmo valor de 3,3 volts, resultando em uma diferença de potencial que acende o LED.

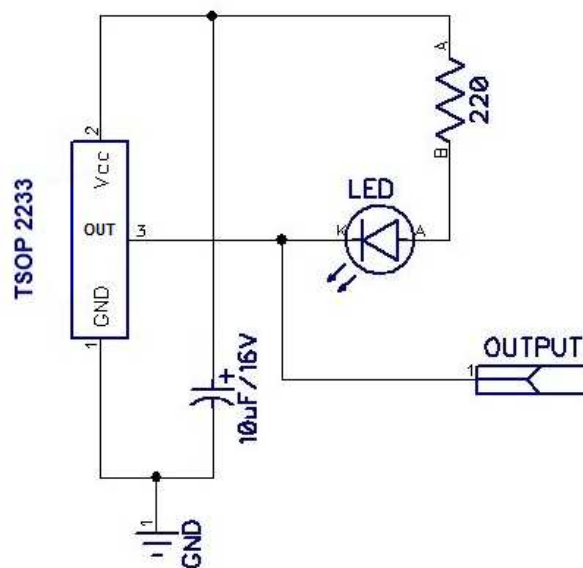


Figura 4.5 – Circuito receptor do Sensor Infravermelho – Fonte (Luciana Mello, 2011)

Por fim, é importante ressaltar que a alimentação dos dois circuitos seja a mesma. Tanto o circuito emissor quanto o circuito receptor serão alimentados pela plataforma Arduino. Na figura 4.6 podemos observar o dois circuitos montados.

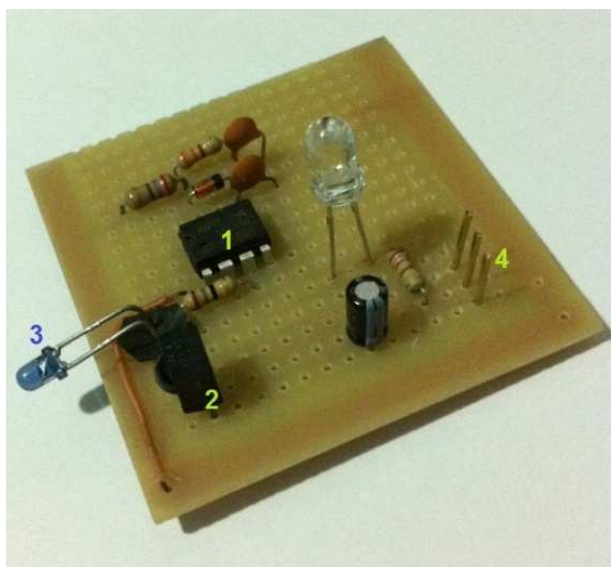


Figura 4.6 – Circuito do Sensor de Presença – Fonte (Autor do Projeto, 2011)

Podemos observar no circuito da figura 4.6 os seguintes componentes:

- 1 – Circuito Integrado Oscilador NE555N;
- 2 – TSOP 2233, responsável pela leitura da luz infravermelha;
- 3 – LED infravermelho, que emite luz infravermelha a uma frequência de 33KHZ;
- 4 – Saídas do Circuito, correspondente ao pino de saída, positivo e o terra.

4.2.2 – Integração dos Dispositivos

Nesta etapa ocorreu a definição dos pinos do Arduino utilizados no projeto e a montagem do mesmo, com o auxílio de um protoboard. Com os testes dos componentes realizados, esta etapa do projeto foi iniciada com a implementação do circuito simulado do alarme, correspondente ao sensor de presença e ao buzzer.

O sensor de presença está ligado ao pino 8 do arduino, que foi programado para fazer constantemente a leitura deste pino, detectando presença ou não. Por outro lado, o buzzer está conectado ao pino 9 do microcontrolador e seu negativo a um botão, que se encontra ligado ao pino 7.

O arduino foi programado primeiramente para fazer a leitura do pino 8, ou seja, do sensor infravermelho. Assim, foi necessário declarar uma variável e setar em zero a mesma. Caso o valor deste sensor seja diferente de zero, o programa chamará a função alarme, que será responsável por ativar e desativar o buzzer, com intervalos de tempo entre o HIGH e o LOW. O botão instalado, junto a um resistor de 10k Ω (que serve para evitar interferências), é responsável pelo desligamento do buzzer. Ele fará a leitura contínua do pino sete, que está ligado ao buzzer. Assim que o buzzer disparar, o botão fará a leitura até que o mesmo seja acionado, setando a ativação do alarme novamente em zero, fazendo com que o buzzer desligue. Por fim, o circuito sensor/buzzer/arduino é implementado, restando a integração com o modem.

A última etapa da montagem do projeto envolve a plataforma arduino e o modem G24GSM, através do MAX232, que é responsável pela conversão da comunicação serial RS-232 para a comunicação nos níveis TTL (os níveis de tensão variam do 0 para o Vcc, onde o Vcc é geralmente entre 4.75V e 5.25V. O intervalo de tensão de 0V a 0.8V cria o nível lógico “0”. O intervalo de tensão de 2V a Vcc cria o nível lógico “1”). Os pinos utilizados neste projeto no cabo DB9 foram os pinos 2, 3 e 5. O pino 2 representa o receptor de sinal, o pino 3 o transmissor e o pino 5 representa o

terra. Também foram utilizados quatro capacitores eletrolíticos de capacitância de $1\mu\text{F}$ e voltagem de 50V. Na figura 4.7 podemos observar tanto a disposição dos capacitores como a ligação de seus pólos positivos e negativos e os transmissores e receptores de sinais (TX e RX do microcontrolador e do modem).

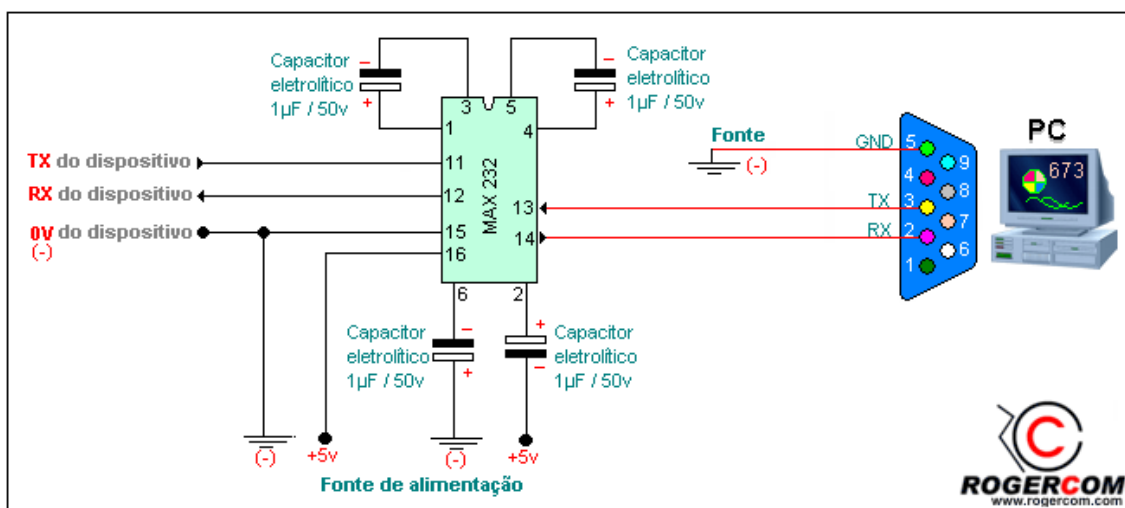


Figura 4.7 – Comunicação dos dispositivos através do MAX232 – Fonte: (RogerCom, 2006)

Após a integração do modem e do microcontrolador com sua devida comunicação, a etapa final da montagem do projeto envolveu a conexão do sensor de presença, o buzzer e o botão para desligar o buzzer. Primeiramente, ligou-se o buzzer com o negativo no botão (ligado ao pino 7) e o positivo no pino 9 da plataforma arduino. Em seguida, o sensor infravermelho que possui três ligações, sendo que uma é o terra, a outra é o nó positivo do arduino e a terceira ligada ao pino 8 do microcontrolador.

Na figura 4.8, é possível observar o desenvolvimento do projeto no protoboard, contendo o conversor MAX232, o buzzer, o botão e as ligações do sensor de presença, do modem e do Arduino.

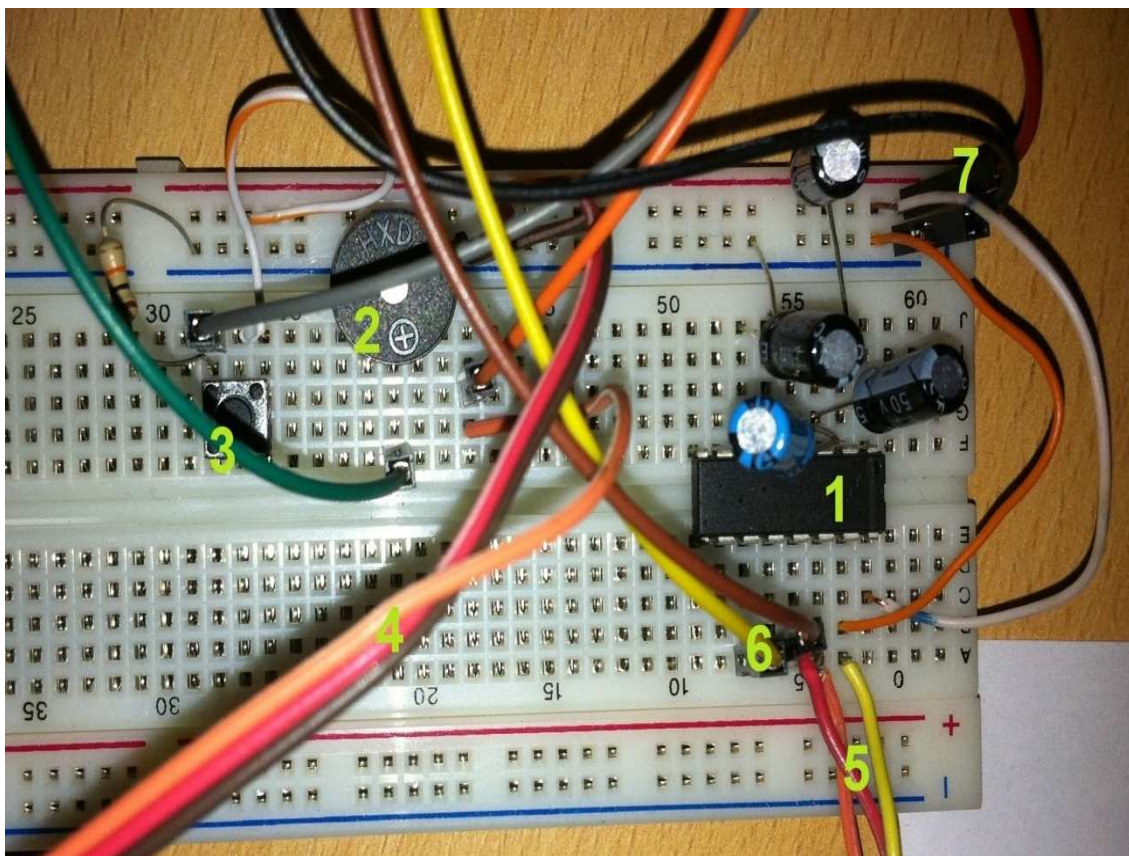


Figura 4.8 – Ligação do MAX232, buzzer, botão e sensor infravermelho – Fonte: (Autor do Projeto, 2011)

Podemos observar na figura acima os seguintes componentes:

- 1 Conversor MAX232;
- 2 Buzzer;
- 3 Botão;
- 4 Ligações do sensor de presença, sendo que o vermelho é ligado no positivo, o marrom no negativo e o laranja é o pino de saída;
- 5 Ligações do Modem a partir do conversor. Vermelho corresponde ao TX, Laranja ao RX e o Amarelo liga-se ao terra;
- 6 Ligações do Arduino. Amarelo liga-se ao TX da plataforma e o Marrom, ao RX;
- 7 Ligações do Arduino para Alimentação e Terra;

As ligações à plataforma Arduino podem ser observadas na figura 4.9 a seguir:

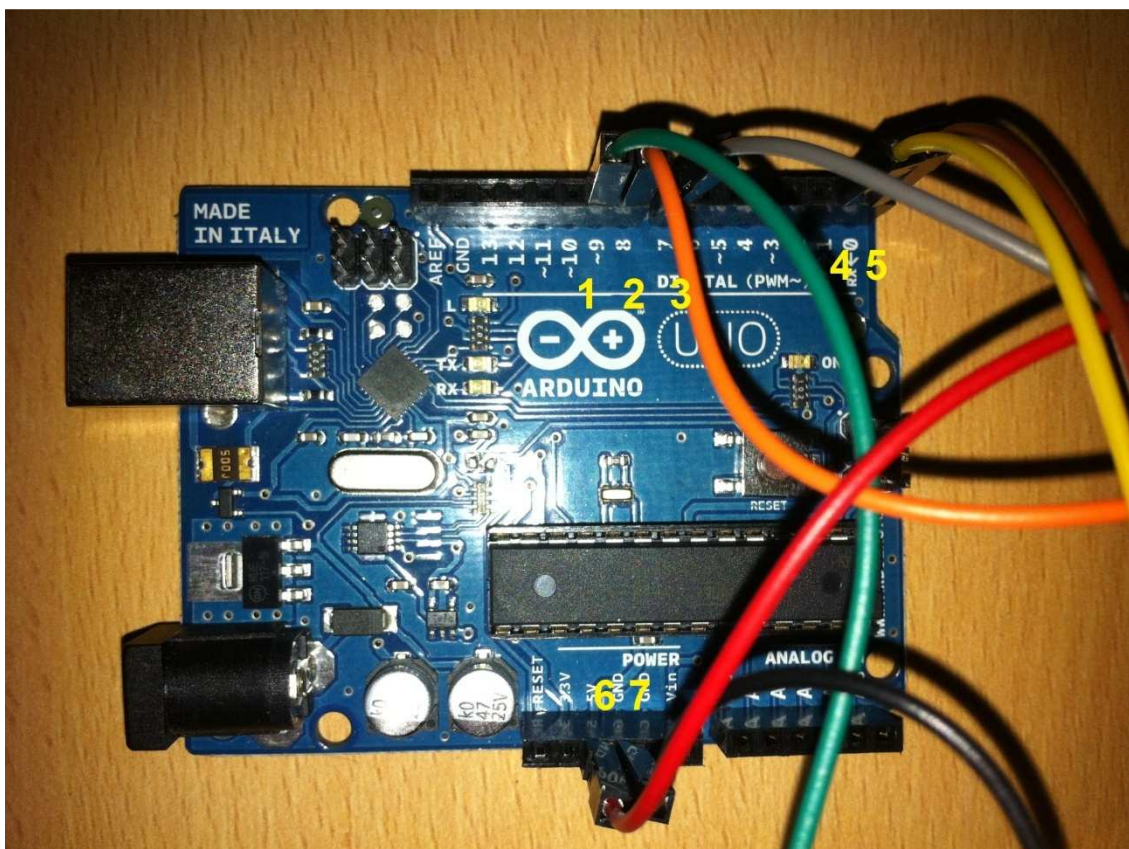


Figura 4.9 – Ligações na plataforma Arduino – Fonte: (Autor do Projeto, 2011)

1. Pino 9 – Ligação do Arduino com o Buzzer;
2. Pino 8 – Ligação do Arduino cmo o Sensor de Presença;
3. Pino 7 – Ligação do Arduino com o botão;
4. Pino 1 (TX) – Ligação do pino transmissor com o MAX232;
5. Pino 0 (RX) – Ligação do pino receptor com o MAX232;
6. Pino de alimentação (5V);
7. Pino de aterramento (GND).

4.3 – Descrição da Implementação

Pode-se observar nesta etapa após a montagem do protótipo a implementação da lógica do sistema, utilizando o sketch do arduino, que é o compilador da plataforma. Com o código definido, é feito o upload do programa na própria IDE do Arduino, que grava automaticamente o código na plataforma. O upload do código pode ser observado na figura 4.10.

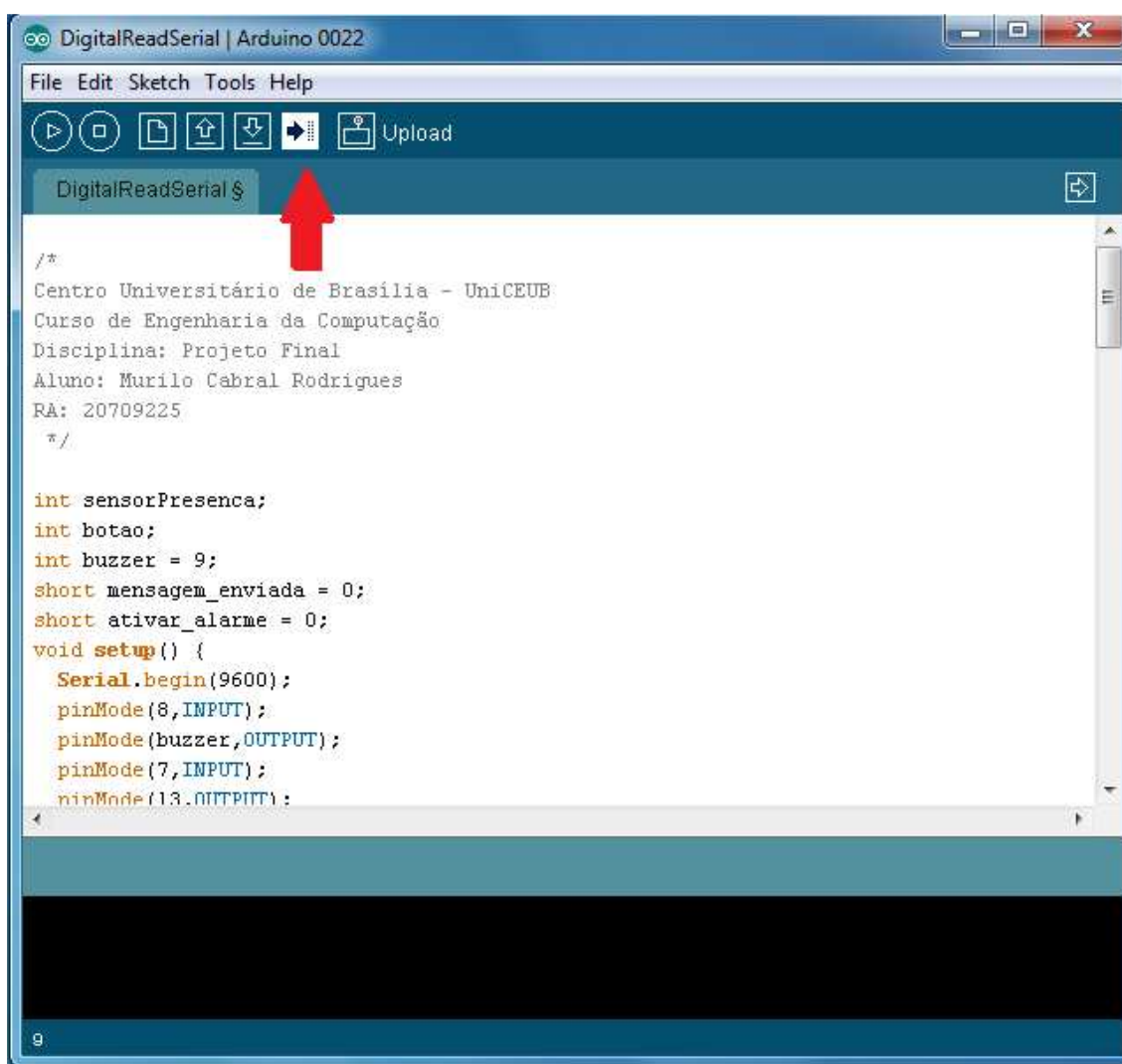


Figura 4.10 – Upload do código no Arduino – Fonte: (Autor do Projeto, 2011)

A programação desenvolvida no projeto pode ser observada no datagrama da figura 4.11. Neste processo de implementação, foram definidas as variáveis e funções necessárias para o desenvolvimento do projeto. Primeiramente, definiu-se a função “setup ()”, com os parâmetros iniciais necessários para que o programa compile e a função “loop ()”, que se trata da principal função do programa, com toda a lógica do sistema de monitoramento.

A partir da função principal, são utilizados dois componentes de entrada de dados: o sensor infravermelho e o botão. Nesta função também foi implementada uma função de verificação do código, `alive ()`. Esta serve basicamente para verificar se o código está rodando no Arduino, fazendo com que o led do pino 13 pisque constantemente. Em seguida, ao detectar presença, o circuito receptor do sensor

infravermelho se comunica com a plataforma Arduino, que chamará duas funções: “alarme ()” e “ envia_sms ()”. A primeira é responsável pela sinalização sonora do buzzer e o segundo, pelo envio dos comandos AT ao modem, que fará a comunicação com o celular. Após o a execução de envio, duas variáveis são setadas com o valor lógico 1: “ativar_alarme” e “mensagem_enviada”.

Por fim, a variável definida para o botão fará leitura do pino 7. Assim que receber dados a partir do acionamento do mesmo, as variáveis “ativar_alarme” e “mensagem_enviada” serão alteradas para valor lógico 0. O código do sistema de monitoramento será apresentado no Apêndice A.

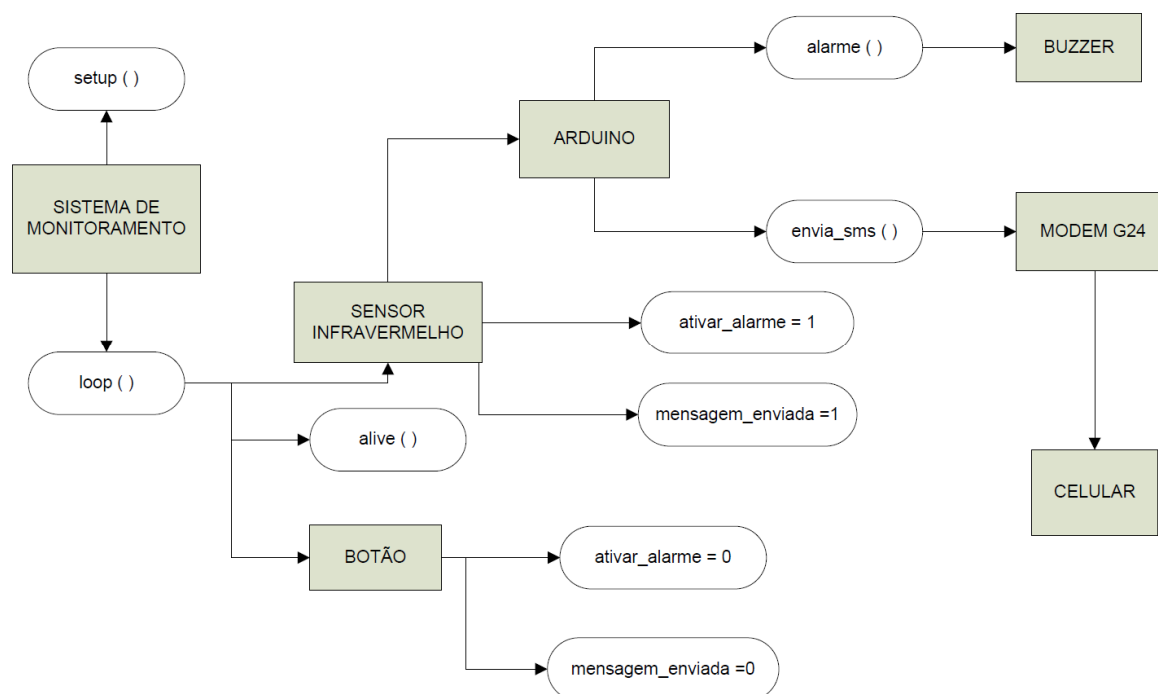


Figura 4.11 – Datagrama do código do Sistema de Monitoramento – Fonte (Autor do Projeto, 2011)

Definida toda a parte de implementação e programação, o protótipo define um monitoramento a partir do acionamento do sensor de presença, que aciona o buzzer e realiza a comunicação com o Arduino. Este por sua vez, enviará os dados com os comandos AT para o modem, por meio do conversor de sinais TTL em comunicação serial RS-232, o MAX232. Assim que o modem receber os comandos, ele enviará o SMS para o celular do proprietário.

CAPÍTULO 5 - APLICAÇÃO PRÁTICA DO MODELO PROPOSTO

O desenvolvimento do projeto demonstrou o funcionamento do sistema de monitoramento proposto. Assim, a partir do acionamento do circuito simulado do alarme, foi realizado o envio do SMS para o celular do usuário. Na figura 5.1, podemos observar a implementação do protótipo do sistema de monitoramento.

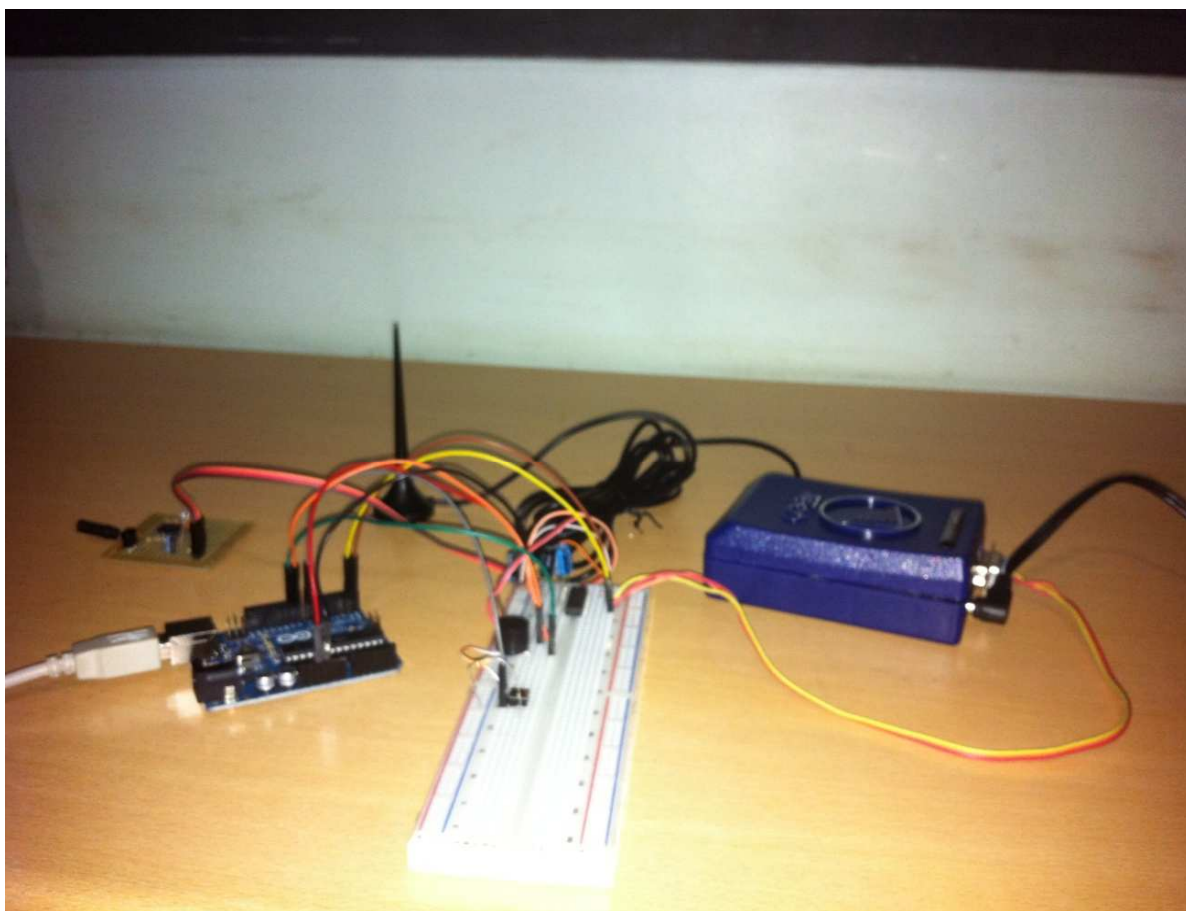


Figura 5.1 – Protótipo do Sistema de Monitoramento – Fonte: (Autor do Projeto, 2011)

Em um sistema real, o circuito simulado que é constituído pelo sensor infravermelho de presença e o buzzer para emitir o sinal sonoro seria substituído pelo próprio alarme automotivo, que envolvem outros tipos de sensores, tais como sensor de pressão, sensor de temperatura, sensor quebra vidro, entre outros. No lugar do buzzer, o circuito real de um alarme de carro possui uma sirene. Assim, a ligação entre a plataforma arduino e o modem se ligaria diretamente ao alarme de um carro.

5.1 - Apresentação da área de Aplicação do modelo

O modelo do sistema de monitoramento proposto pode ser implementado na área de segurança automotiva, podendo ser desenvolvido em empresas que fabricam alarme de carros, tais como a Positron, H-Buster, Steel Bull, entre outras.

Estas empresas podem desenvolver o dispositivo de monitoramento integrado ao sistema de alarme da empresa, ou comercializar o mesmo como um item adicional ao alarme veicular.

É importante ressaltar que este projeto também pode ser desenvolvido com diversos outros intuitos, sendo que o principal objetivo do sistema de monitoramento é a manutenção da segurança. Assim, este dispositivo pode ser implementado em residências, prezando a segurança do lar através do uso de qualquer tipo de sensor que possa detectar algum tipo de invasão. Além disso, o projeto também pode abranger para área de segurança de empresas, em áreas de sigilo que necessitam de manutenção ininterrupta, entre outros.

Através de uma implementação de recebimento de SMS no modem, a área de automação também pode-se beneficiar com aplicação em diversas áreas. O envio de comandos para executar tarefas, tais como acender ou apagar uma luz, abrir ou fechar um portão, dentre diversas outras funções que podem ser implementadas a partir do recebimento do SMS pelo Modem e pela leitura e interpretação do SMS pelo microcontrolador.

5.2 – Descrição da Aplicação do Modelo

A emissão de luz infravermelha pelo LED infravermelho presente no circuito emissor do sensor de presença alimentado pela plataforma Arduino é o primeiro passo do projeto. Ao se deparar com um objeto, esta radiação é refletida e detectada pelo TSOP 2233 presente no circuito receptor deste mesmo sensor. Assim que a presença é detectada, o microcontrolador aciona o buzzer através do pino 9.

Paralelo ao disparo do alarme simulado, o Arduino envia os comandos AT que serão interpretados pelo modem G24GSM. Este, que possui um chip da Vivo inserido

(pode ser chip de qualquer outra operadora), irá enviar o SMS de alerta “Alarme Acionado. Favor Verificar” para o celular do usuário utilizando a telefonia celular.

O projeto desenvolvido obteve os resultados esperados. Assim, o monitoramento a partir da detecção de presença funcionou conforme o planejado. Existe um atraso mínimo de cerca de 4 segundos entre o acionamento do alarme simulado e o envio da mensagem. Porém, este atraso é tolerável. A maior dificuldade encontrada no desenvolvimento deste projeto foi a alta sensibilidade do sensor de presença, visto que à medida que o circuito é alimentado, a emissão da luz infravermelha é cada vez mais captada pelo circuito receptor, fazendo com que o circuito se confunda com a presença ou não de algum obstáculo. Para isso, utilizou-se um termo retrátil, que se trata de um cabo que se retrai ao ser aquecido. Assim, o mesmo foi integrado com o intuito de isolamento, a fim de reduzir o raio de emissão da luz infravermelha.

A vantagem da utilização do Arduino em relação às mudanças no modem é a facilidade na alteração do número do usuário e da mensagem, que pode ser implementada no sketch do Arduino apenas alterando para o número e mensagem desejados e posteriormente, carregando diretamente no Arduino, sem a necessidade de algum tipo de gravador, como no caso do PIC onde isto é necessário.

Por fim, a realização dos testes dos componentes foi fundamental para o desenvolvimento do projeto, visto que o mesmo poderia ser comprometido com a falha de algum componente. Assim, com todos componentes funcionando, as outras etapas podem ser desenvolvidas.

5.2.1 – Teste dos componentes

A etapa dos testes mencionada no Capítulo 4 foi importante para verificar o funcionamento individual dos componentes. O primeiro componente que foi testado neste projeto foi o Arduino. Nesta etapa, verificamos o funcionamento do mesmo através do uso de seu respectivo compilador. Para isso, ligamos o Arduino ao computador através do cabo USB e instalamos o seu driver, que encontra-se na pasta do compilador, baixado no site www.arduino.cc. Em seguida, foi necessário configurar a placa do Arduino utilizada (no caso deste projeto, o Arduino UNO) e a porta de comunicação, que depende da porta USB utilizada em seu computador. Em seguida, um exemplo do próprio compilador foi aberto para testar o funcionamento do microcontrolador. O exemplo “*blink*” fez com que os leds do TX e RX piscassem

rapidamente e o led do pino 13 ficasse piscando continuamente. Na figura 5.2 podemos observar a escolha da placa utilizada, bem como a opção “Serial Port”, para a seleção da porta USB em uso. Já na figura 5.3, podemos observar a programação para o exemplo utilizado no teste do microcontrolador.

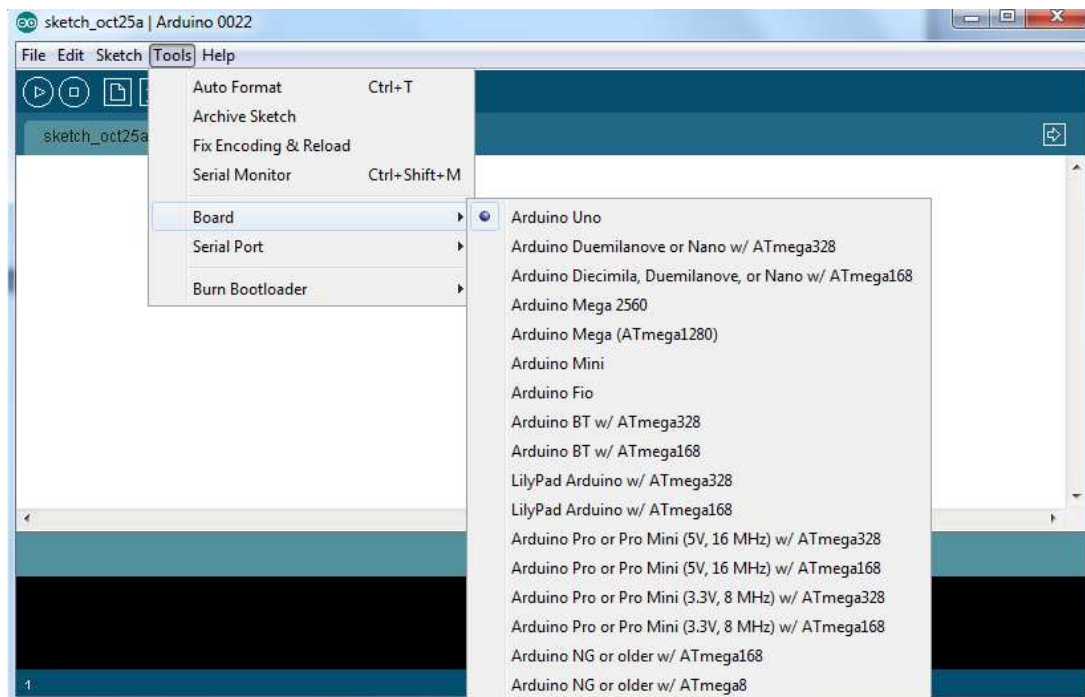


Figura 5.2 – Seleção da Placa e da porta a ser utilizada – Fonte: (Autor do Projeto, 2011)

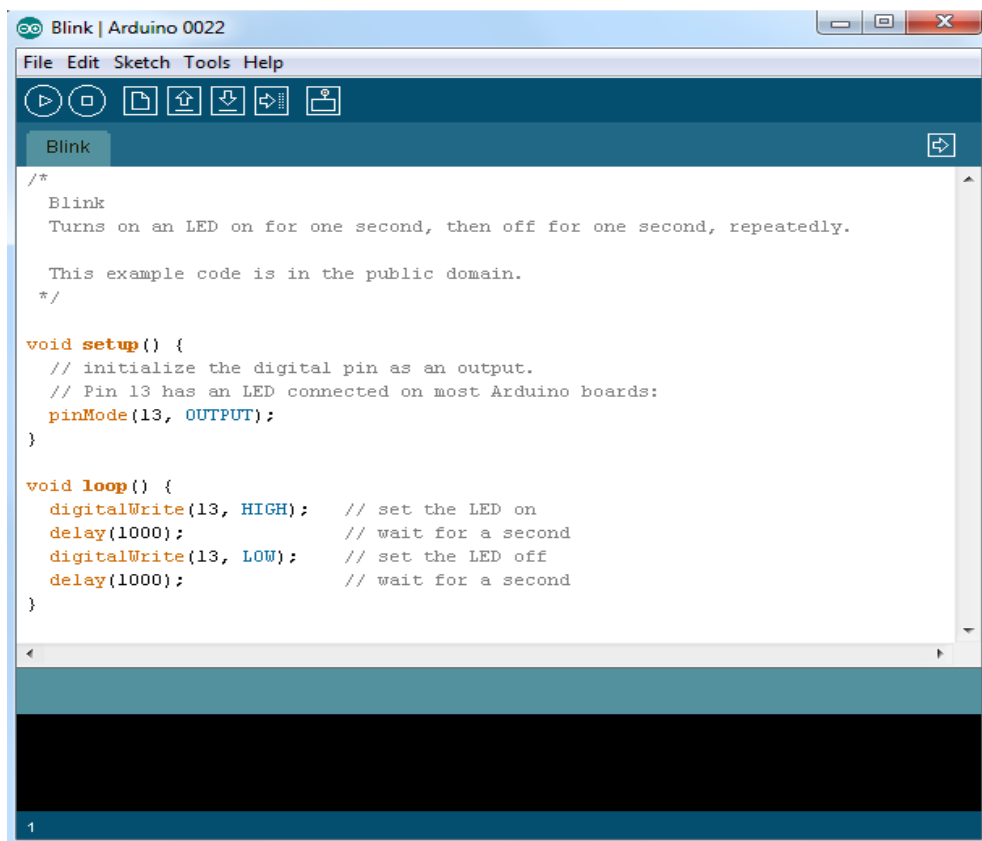
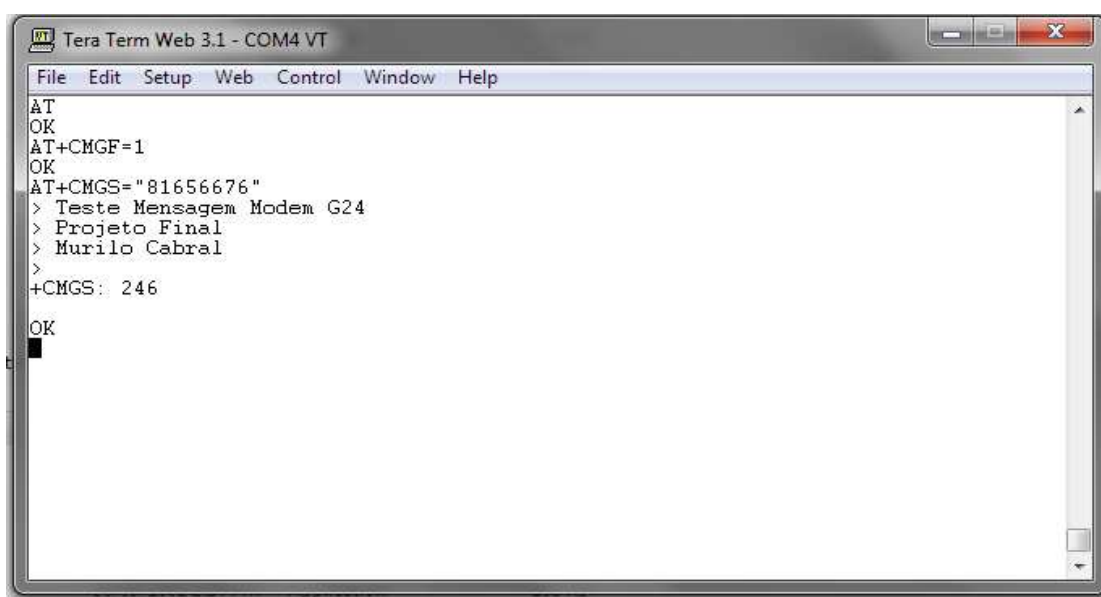


Figura 5.3 – Exemplo de teste: Blink – Fonte: (Autor do Projeto, 2011)

O modem G24GSM é o dispositivo responsável pela comunicação do microcontrolador com a rede de telefonia celular e conseqüentemente, com o celular do usuário. Assim, utilizou-se o terminal *Tera Term Web 3.1* para testar a comunicação do modem por meio de comandos AT. Ligou-se o modem, com um SIM Card da Vivo instalado, através de um cabo Serial/USB e iniciou-se a comunicação. Antes de testar a comunicação em si, o controle modem exige uma configuração padrão para a sua porta serial, que é especificada no manual do produto. É necessário 1 start bit, 1 stop bit, 8 data bits, sem paridade e com uma taxa de transferência de 9600bps. Após a configuração, é testada a comunicação em si através dos comandos AT.

O primeiro comando é justamente o “AT”, que verifica se há alguma comunicação entre o computador e o modem. Caso exista a comunicação, o terminal responde com um “OK”. Caso contrário, o terminal responderia com um “ERROR”. Em seguida, definimos o tipo da mensagem no modo texto, através do “AT+CMGF=1”. Este comando permite definir o tipo de mensagem, modo PDU (aonde todas as mensagens SMS são representados como strings binários codificados em caracteres hexadecimais) ou modo texto, sendo que o primeiro é definido por zero e o segundo por um. O retorno do comando também é um “OK”. Por fim, o comando “AT+CMGS=<numero>” define o número para o qual a mensagem será enviada, bem como o conteúdo da mensagem. A resposta ocorre através de uma mensagem de referência, “+CMGS: num” seguido por um “OK”. Podemos observar os comandos utilizados no teste na figura 5.4:



```
Tera Term Web 3.1 - COM4 VT
File Edit Setup Web Control Window Help
AT
OK
AT+CMGF=1
OK
AT+CMGS="81656676"
> Teste Mensagem Modem G24
> Projeto Final
> Murilo Cabral
>
+CMGS: 246
OK
```

Figura 5.4 – Teste do modem G24 – Fonte: (Autor do Projeto, 2011)

Depois do desenvolvimento dos comandos AT, a mensagem chegou ao celular do usuário, como podemos observar na figura 5.5. O chip inserido no modem possui o número “9664-5280”.



Figura 5.5 – Resposta do teste do modem G24 – Fonte: (Autor do projeto, 2011)

A análise do acionamento do buzzer através da detecção de presença foi a segunda etapa dos testes. Com o auxílio de um protoboard, realizamos a conexão entre os componentes buzzer, sensor infravermelho montado e o arduino, este último ligado ao computador, para que fosse feita a programação de teste do circuito. As etapas de teste deste componente coincidem com o início da integração dos dispositivos, que foram tratados no Capítulo 4.

5.3 – Resultados da Aplicação do Modelo

As etapas realizadas neste projeto podem ser descritas em três partes principais:

1º - Detecção do objeto, representado pelo circuito emissor e circuito receptor do sensor infravermelho, que pode ser observado na figura 5.6. Pode-se observar que ao detectar presença, o LED do circuito receptor acende, devido a diferença de potencial entre o catodo e o anodo do mesmo;

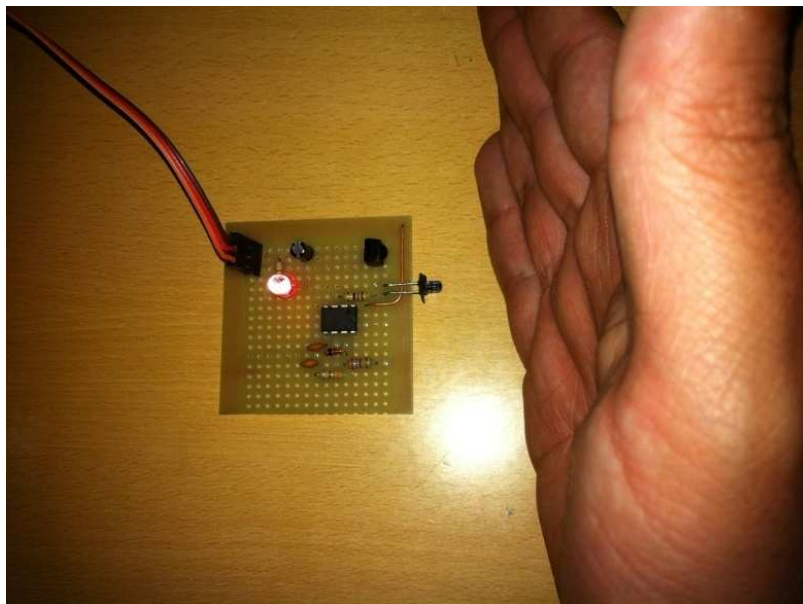


Figura 5.6 – Detecção de presença – Fonte: (Autor do Projeto, 2011)

2º - Acionamento do buzzer, que serve para simulação do alarme por meio de um sinal sonoro com intervalos de HIGH e LOW definidos no programa. A figura 5.7 demonstra o acionamento do buzzer;



Figura 5.7 – Mensagem de Alerta – Fonte: (Autor do Projeto, 2011)

3º - Envio do SMS através da plataforma Arduino, que envia os comandos AT específicos de envio da mensagem;

Assim, com todas as etapas definidas e implementadas, o projeto do sistema de monitoramento funcionou como esperado. O circuito completo do projeto do dispositivo pode ser observado na figura 5.8, que agrega o circuito emissor e receptor do sensor infravermelho. Ambos os circuitos são alimentados pelo Arduino. O TSOP 2233 têm o seu pino de saída ligado ao pino 8 do arduino, que será o alerta para o acionamento do buzzer no pino 9 e o envio do SMS, através de comandos enviados ao modem, o qual será ligado ao conector DB9 presente no circuito. A ligação entre o modem e o Arduino pode ser observada através do conversor MAX232.

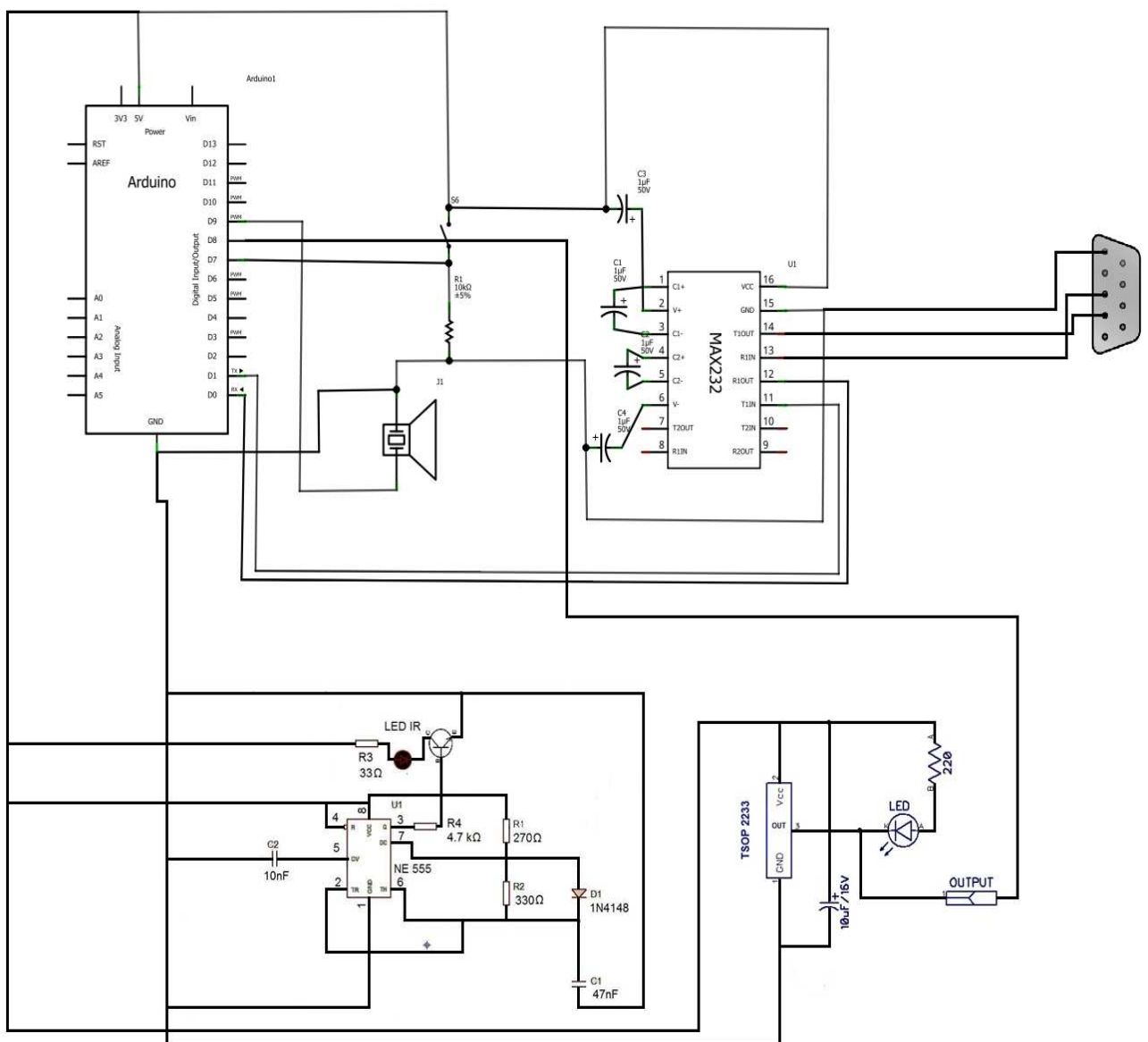


Figura 5.8 – Circuito do Sistema de Monitoramento – Fonte: (Autor do Projeto, 2011)

5.4 – Custos do modelo proposto

O desenvolvimento deste projeto teve um custo muito elevado, devido ao alto preço do módulo presente no modem G24GSM. Podemos observar a seguir na tabela 6 uma estimativa de custos do projeto e o seu valor total.

Tabela 6 – Estimativa de custos do Dispositivo de Monitoramento

Componente	Quantidade	Custo (por unidade)
Arduino Uno	1	R\$ 100,00
Buzzer	1	R\$ 1,49
MAX232	1	R\$ 1,96
Capacitor Eletrolítico 1µF 50V	4	R\$ 0,12
Capacitor Eletrolítico 10µF 16V	1	R\$ 0,15
Capacitor Cerâmico 10nF 50V	1	R\$ 0,09
Capacitor Cerâmico 47nF 50V	1	R\$ 0,15
Diodo 1N4148	1	R\$ 0,15
Resistores em geral	5	R\$ 0,10
Barra de pinos	1	R\$ 0,69
LED infravermelho	1	R\$ 0,48
LED 10mm	1	R\$ 0,99
Circuito Integrado Intersil NE555N	1	R\$ 0,65
Receptor Infravermelho TSOP2233	1	R\$ 3,74
Termoretrátil (30cm)	1	R\$ 1,00
Cabo de 10 dias (30cm)	1	R\$ 10,00
Botão	1	R\$ 0,50
Conector Macho DB9	1	R\$ 0,61
Modem G24GSM – iTech	1	R\$ 398,00
TOTAL		R\$ 520,87

Fonte: (Autor do Projeto, 2011)

O projeto, contudo, poderia ter um custo inferior se fosse utilizado alguma plataforma genérica do Arduino, que possuem as mesmas características e são compatíveis com o compilador do Arduino. Além disso, o circuito do modem poderia ter sido desenvolvido, necessitando apenas comprar o módulo para comunicação GSM/GPRS. Assim, o projeto poderia ficar muito mais barato. A utilização do PIC ao invés do Arduino a princípio pode parecer mais barato. Contudo, é necessário um

gravador de PIC para que as linhas de código desenvolvidas possam ser gravadas no microcontrolador. Este componente possui um valor elevado. Assim, os custos entre a utilização de um microcontrolador PIC e a plataforma Arduino são similares.

5.5 – Avaliação Global do Modelo

O desenvolvimento e implementação deste dispositivo de monitoramento serve como um item adicional no sistema de segurança veicular, possibilitando ao usuário uma maior noção acerca da situação de seu veículo. Contudo, o sistema desenvolvido não garante evitar algum tipo de incidente, visto que ele só tem como foco o monitoramento. A partir deste monitoramento, é possível ou não a prevenção de algum roubo ou furto.

A implementação do dispositivo utilizando a plataforma Arduino mostrou-se bastante eficiente, visto que esta plataforma, amplamente utilizada em meios acadêmicos na atualidade, demonstra uma grande facilidade tanto em sua implementação quanto em sua programação. Além disso, a gravação do código para o microcontrolador de forma direta, ou seja, fazendo o “upload” do computador direto para o Arduino demonstra sua versatilidade na hora do desenvolvimento.

CAPÍTULO 6 – CONCLUSÃO

6.1 – Conclusões

Através do desenvolvimento deste projeto, pôde-se observar a implementação de um sistema de monitoramento de alarme automotivo através do envio de uma mensagem SMS.

Além da proposta inicial do projeto como um item adicional ao alarme veicular, foi possível perceber a enorme abrangência com o desenvolvimento de um circuito desses. Ou seja, a utilização deste dispositivo pode ser utilizada não somente em um alarme automotivo, mas em qualquer lugar que necessite de algum nível de segurança; como uma casa, que possui valores agregados maiores do que um carro; em empresas, que detêm informações que exigem uma confidencialidade para a segurança da empresa, entre outros exemplos em que possam ser utilizados sensores, seja de presença, temperatura, pressão, entre outros.

O projeto envolve um sistema adicional de segurança, porém, não garante a prevenção de algum incidente. A utilização e funcionamento deste podem vir a prevenir algum roubo ou furto. Contudo, não há garantia nisto.

Os valores deste projeto foram elevados, custando cerca de R\$520. Este valor a princípio pode limitar o desenvolvimento deste projeto, visto que nem todos podem investir em um sistema de monitoramento tão caro. Porém, desmembrando os componentes utilizados, podemos perceber que o modem pode ser implementado, necessitando apenas de seu módulo GSM, que possui um valor muito abaixo do que o modem completo. Ou seja, é possível montar uma placa genérica de um modem por um preço inferior. Além disso, uma vez que se possui o Arduino, o mesmo pode ser utilizado também como um gravador de código, encaixando o microcontrolador Atmega328p à plataforma. O valor deste microcontrolador é inferior ao da própria plataforma. Assim, com o código gravado no microcontrolador, resta desenvolver uma placa com os componentes correspondentes no Arduino. Com isto, os valores deste projeto são reduzidos a menos da metade do valor inicial.

Em relação aos objetivos gerais e específicos do desenvolvimento do projeto, todos foram atendidos. O objetivo geral, o desenvolvimento do sistema de monitoramento veicular agregado ao alarme de um carro, funcionou conforme o

planejado. As etapas que contribuíram para este desenvolvimento, ou seja, os objetivos específicos do projeto também foram implementados. Assim, o alarme simulado, que corresponde ao buzzer e ao sensor infravermelho (composto por um circuito emissor infravermelho e um circuito receptor infravermelho), ao detectar presença, disparou o buzzer. Além disso, a utilização do conversor MAX232 permitiu a comunicação do modem com o Arduino, tornando possível a troca de informações entre os dois dispositivos, através de seus respectivos TX e RX. Assim, o último objetivo específico foi desenvolvido, o envio do SMS de alerta ao celular do usuário do sistema.

Por fim, a utilização da plataforma Arduino na implementação deste sistema pode demonstrar a enorme abrangência em relação à funcionalidade do mesmo. A facilidade em desenvolver o sistema e gravar o código no microcontrolador (através de um simples upload na IDE do Arduino) são características que fazem com que este componente seja amplamente utilizado em projetos acadêmicos.

6.2 - Sugestões para Trabalhos Futuros

O projeto do dispositivo de monitoramento possui inúmeras utilidades dentro da área de segurança, podendo ser implementada de várias formas. O dispositivo de segurança pode ser otimizado através da utilização de outros componentes que possam servir como outro adicional ao alarme veicular, sempre buscando um sistema cada vez mais robusto quando se trata de segurança. Assim, a utilização de um GPS poderia ser agregado ao sistema, que, ao acionar o alarme do carro, possa enviar o SMS com a mensagem de alerta do alarme acionado e as coordenadas da posição atual do veículo. Além disso, o envio de um SMS ao dispositivo poderia fazer com que esta informação de localização pudesse ser acessada na hora em que o usuário desejasse. Com isto, o sistema poderia servir até como uma ajuda em uma busca ao veículo no caso de roubo.

Aliado ao sistema de monitoramento ainda, o acionamento de uma câmera poderia ser utilizada no dispositivo, a fim de obter um quadro da situação atual do interior do veículo. Ou seja, com o acionamento do alarme, a câmera é acionada e a partir deste momento, qualquer ação ocorrida no interior do veículo é monitorada. Porém, o desenvolvimento deste item exige um processamento maior, que não é suportado pelo Atmega328 da plataforma Arduino. Em caso de utilização do Arduino, o

projeto necessitaria da utilização de um processador digital de sinais, que possui um valor bastante elevado.

Além disso, em uma situação real, um sistema de alarme automotivo não envolve somente incidentes relacionados a roubos e furtos. O disparo acidental dos alarmes de carro ocorre com frequência. Um dos grandes motivos disto pode ser a interferência provocada por um som automotivo com uma intensidade de som bastante elevada. Assim, o carro não está em perigo, porém, o alarme é disparado. Portanto, se o usuário e dono do veículo souber deste incidente, é possível implementar um sistema que envie um SMS para desativar o alarme, sem ter que se dirigir ao raio de cobertura do controle do alarme, visto que o raio de cobertura de um SMS coincide com o raio de cobertura de uma operadora de telefonia celular, sendo muito superior.

Por fim, o conhecimento adquirido no desenvolvimento do projeto através da troca de informações entre o modem e o microcontrolador por meio de mensagens de texto podem ser implementados em outras áreas, que não envolvam a segurança. Um exemplo de implementação que pode ser desenvolvida é a automação residencial por meio de SMS. A vantagem deste tipo de automação seria o seu enorme raio de cobertura. Além disso, é possível tratar não somente de automação residencial, mas também automação empresarial, como por exemplo, uma irrigação de uma determinada agricultura, gerando benefícios para o negócio.

REFERÊNCIAS BIBLIOGRÁFICAS

BEHROUZ A. FOROUZAN - COMUNICAÇÃO DE DADOS E REDES DE COMPUTADORES, EDITORA MCGRAW-HILL BRASIL, QUARTA EDIÇÃO, 2008, BRASIL

BOYLESTAD, ROBERT & NASHELSKY, LOUIS – DISPOSITIVOS ELETRÔNICOS E TEORIA DOS CIRCUITOS, EDITORA LTC, SEXTA EDIÇÃO, 1999, RIO DE JANEIRO

DAMA, LUIS - LINGUAGEM C, EDITORA LTC, 10ª EDIÇÃO

MALVINO, ALBERT PAUL & J. BATES, DAVID – ELETRONICA VOLUME 1, EDITORA MCGRAW-HILL BRASIL, SÉTIMA EDIÇÃO, 2008, RIO DE JANEIRO

MELLO, LUCIANA FERREIRA – RFID EM SISTEMAS DE SEGURANÇA EM PRÉDIOS, MONOGRAFIA DE GRADUAÇÃO UNICEUB, ENGENHARIA DA COMPUTAÇÃO, 01SEM/2010

PEREIRA DA SILVA JUNIOR ,VIDAL – MICROCONTROLADORES, ED. ERICA, SEXTA EDIÇÃO, 2008, BRASIL

ROBERTS, MICHAEL - ARDUINO BÁSICO , EDITORA NOVATEC, PRIMEIRA EDIÇÃO, 2011, SÃO PAULO

SOARES NUNES, JOSÉ RENATO - COMUNICAÇÃO DE DADOS: CONCEITOS BÁSICOS

THOMAZINI, DANIEL & URBANO BRAGA DE ALBUQUERER, DANIEL - SENSORES INDUSTRIAIS - FUNDAMENTOS E APLICAÇÕES

<http://g1.globo.com/distrito-federal/noticia/2011/09/df-tem-23-roubos-e-furtos-de-carro-por-dia-mas-70-sao-recuperados.html>, ACESSADO EM 09/09/2011.

<http://www.pcdf.df.gov.br>, ACESSADO EM 09/09/2011.

http://www.multlock.com.br/automotivo_home.html, ACESSADO EM 09/09/2011.

<http://www.iii.org/media/hottopics/insurance/test4/>, ACESSADO EM 09/09/2011.

<http://carros.hsw.uol.com.br/alarmes-dos-carros.htm>, ACESSADO EM 09/09/2011.

<http://www.lammertbies.nl/comm/cable/RS-232.html>, ACESSADO EM 10/09/2011.

<http://www.apostilando.com/download.php?cod=333&categoria=Outras%20Apostilas>, ACESSADO EM 10/09/2011.

<http://www.mc.gov.br>, ACESSADO EM 15/09/2011.

http://www.telebras.com.br/inst/?page_id=41, ACESSADO EM 15/09/2011.

<http://www.alcatel-lucent.com>, ACESSADO EM 15/09/2011.

<http://www.fundacaotelefonica.org.br/Arte-e-Tecnologia/Memoria-Telefonica-Pesquisa.aspx>, ACESSADO EM 15/09/2011.

<http://wataniya.yahoo.com/smshistory.html>, ACESSADO EM 16/09/2011.

<http://www.wisegeek.com/what-is-sms-marketing.htm>, ACESSADO EM 16/09/2011.

<http://www.wisegeek.com/what-is-a-sim-card.htm>, ACESSADO EM 16/09/2011.

http://www.electronica-pt.com/index.php/component/option.com_wrapper/Itemid,98/, ACESSADO EM 31/10/2011.

<http://www.diafaan.com/sms-tutorials/gsm-modem-tutorial/at-cmgf/>, ACESSADO EM 31/10/2011.

<http://www.datasheetcatalog.org/datasheet/MicroElectronics/mXuyxru.pdf>, ACESSADO EM 11/11/2011.

<http://mae.osu.edu/sites/mae.web.engadmin.ohio-state.edu/files/page/2010/09/2n3904.pdf>, ACESSADO EM 11/11/2011.

http://www.atmel.com/dyn/resources/prod_documents/doc8161.pdf, ACESSADO EM 11/11/2011.

<http://www.elboby.com/2011/10/esquema-arduino/>, ACESSADO EM 11/11/2011.

<http://www.arduino.cc/es/Main/ArduinoBoardDiecimila>, ACESSADO EM 11/11/2011.

http://www.datasheetcatalog.org/datasheets/2300/301343_DS.pdf, ACESSADO EM 12/11/2011.

<http://www.datasheetcatalog.org/datasheet2/1/03tsqojks833e6hozotadpz5yxfy.pdf>, ACESSADO EM 12/11/2011.

<http://digital.ni.com/public.nsf/allkb/33E285562B2C8A91862577AC005527F1>, ACESSADO EM 15/11/2011.

APÊNDICE A – CÓDIGO DO SISTEMA DE MONITORAMENTO

```
/*
```

```
Centro Universitário de Brasília - UniCEUB
```

```
Curso de Engenharia da Computação
```

```
Disciplina: Projeto Final
```

```
Aluno: Murilo Cabral Rodrigues
```

```
RA: 20709225
```

```
*/
```

```
int sensorPresenca; //Declara variável do Sensor Infravermelho
```

```
int botao; // Declara a variável do botão
```

```
int buzzer = 9; // Declara a variável do buzzer e seta no pino 9
```

```
short mensagem_enviada = 0; //Verificar se a mensagem foi enviada
```

```
short ativar_alarme = 0; // Verificar se o alarme está ativado
```

```
void setup() {
```

```
    Serial.begin(9600); // Define parâmetros do modem
```

```
    pinMode(8,INPUT); // Define o pino do sensor como entrada de dados
```

```
    pinMode(buzzer,OUTPUT); // Define o buzzer como saída de dados
```

```
    pinMode(7,INPUT); // Define o botão como entrada de dados
```

```
    pinMode(13,OUTPUT); // Define o led do pino 13 como saída de dados
```

```
    digitalWrite(buzzer,LOW); //Seta o buzzer como LOW, ou seja, desligado
```

```
Serial.print("AT\n\r"); //Envia comando de comunicação com o modem

delay(500);

Serial.print("ATE0\n\r"); //Desativa Echo Commands

delay(500);

Serial.print("AT+CMGF=1\n\r"); // Define mensagem em modo texto

}

void envia_sms() { //função para o envio da mensagem

    Serial.print("AT\n\r");

    delay(500);

    alarme(); //chama a função alarme

    Serial.print("ATE0\n\r");

    delay(500);

    alarme();

    Serial.print("AT+CMGF=1\n\r");

    delay(500);

    alarme();

    Serial.print("AT+CMGS=\"81656676\"\n\r"); //define o número do destinatário
```

```
delay(500);

alarme();

Serial.print("Alarme Acionado. Favor Verificar!"); //define a mensagem enviada

Serial.print(26, BYTE); //ENTER (ctrl+z utilizando a tabela ASCII)

delay(500);

alarme();

Serial.print(26, BYTE);

delay(500);

alarme();

Serial.print(26, BYTE);

delay(1000);

}

void alive() { //função para verificar se o código está funcionando

digitalWrite(13,HIGH); //liga o led do pino 13

delay(50);

digitalWrite(13,LOW); // desliga o led do pino 13

delay(350);

}
```

```
void alarme() { //função que aciona o buzzer

digitalWrite(buzzer,HIGH); //liga o buzzer

delay(80);

digitalWrite(buzzer,LOW); // desliga o buzzer

delay(80);
}

void loop() { //função principal

alive(); //função de teste

sensorPresenca = digitalRead(8); // leitura do pino 8 do Arduino

if(!sensorPresenca) { // se o sensor infravermelho detectar presença...

    if(!mensagem_enviada) { // se a mensagem não foi enviada ainda...

        envia_sms(); // chama a função para o envio do SMS

        mensagem_enviada = 1; // seta a mensagem_enviada para 1

        ativar_alarme=1; // seta o ativar_alarme para 1
    }

    botao = digitalRead(7); // faz a leitura do pino 7 do Arduino

    while(ativar_alarme && !botao) { // Alarme acionado e botão não acionado

        alarme();
```

```
    botao = digitalRead(7); //leitura do pino 7
}

ativar_alarme=0; //seta o ativar_alarme em zero

mensagem_enviada=0; //seta o mensagem_enviada em zero
}
}
```

ANEXOS

Manual G24 Data Modem - Português.pdf

AT_command_Reference_Manual-H.pdf

Datasheet TSOP2233

Datasheet NE555N