



CENTRO UNIVERSITÁRIO DE BRASÍLIA
FACULDADE DE TECNOLOGIA E CIÊNCIAS SOCIAIS APLICADAS
CURSO DE ENGENHARIA DA COMPUTAÇÃO

ARMANDO VITOR DE OLIVEIRA SILVA

ALARME COM ATIVAÇÃO POR SENSOR PRESENCIAL E ALERTA VIA SMS

Orientadora: Prof^a. M.C. Maria Marony de Sousa Farias

BRASÍLIA

1º SEMESTRE DE 2013

ARMANDO VITOR DE OLIVEIRA SILVA

ALARME COM ATIVAÇÃO POR SENSOR PRESENCIAL E ALERTA VIA SMS

**Trabalho apresentado ao UniCEUB – Centro
Universitário de Brasília como pré-requisito pa-
ra obtenção de Certificação de Conclusão do
Curso de Engenharia de Computação.
Orientadora: Prof^a. M.C. Maria Marony Sousa
Farias.**

BRASÍLIA

1º SEMESTRE DE 2013

ARMANDO VITOR DE OLIVEIRA SILVA

ALARME COM ATIVAÇÃO POR SENSOR PRESENCIAL E ALERTA VIA SMS

Trabalho apresentado ao UniCEUB – Centro Universitário de Brasília como pré-requisito para obtenção de Certificação de Conclusão do Curso de Engenharia de Computação. Orientadora: Prof^a. M.C. Maria Marony Sousa Farias.

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 Amarília Fernandes
Coordenador do Curso

Banca Examinadora:

Prof^a. Maria Marony Sousa Farias, mestre em Engenharia Elétrica – UFPB – PB.
Orientadora

Prof.

Prof.

Prof.

*Dedico a todos que influenciaram e
contribuíram para a conclusão desse pro-
jeto e grande etapa da minha vida.*

AGRADECIMENTOS

Aos meus amigos e colegas que estudaram comigo ao longo do curso, que sempre estiveram ao meu lado nos meus momentos de dificuldade, me ajudaram a crescer academicamente e me ensinaram tanto ao longo desses anos.

A Débora Mackert, minha namorada, pelo incentivo, companheirismo e por não me deixar desistir de concluir essa etapa.

A minha família, pelo suporte prestado, e por sempre acreditar no meu sucesso.

E a todos que participaram direta e indiretamente para que eu conseguisse concluir meu projeto.

*“Um passo à frente, e você não está mais
no mesmo lugar.”*

Chico Science.

RESUMO

Neste trabalho é apresentado um dispositivo de alarme com detector de presença e aviso de alerta por SMS. O sensor utilizado é o sensor PIR (Passive Infrared), que detecta movimentação ambiente. O projeto integra um módulo GSM, microcontrolador Arduino e o sensor PIR. No projeto, a detecção de movimento é feita pelo sensor PIR, que aciona um Led e envia um SMS a um celular previamente cadastrado, alertando sobre a invasão de perímetro.

Palavras-chave: Sensor PIR, alarme, microcontrolador Arduino, GSM, SMS.

ABSTRACT

This work presents an alarm device with presence detector and alert notification by SMS. The sensor used is a PIR sensor (Passive Infrared) which detects movement environment. The project integrates a GSM module, microcontroller Arduino and PIR sensor. In the project, the motion detection is done by the PIR sensor, which activates a LED and sends an SMS to a mobile phone previously registered, warning of the invasion perimeter.

Keywords: PIR sensor, alarm, microcontroller Arduino, GSM, text messaging.

SUMÁRIO

LISTA DE FIGURAS	XII
LISTA DE TABELAS E QUADROS.....	XIV
LISTA DE ABREVIATURAS E SIGLAS	XV
CAPÍTULO 1 – INTRODUÇÃO	16
1.1 Apresentação do Problema.....	16
1.2 Objetivos do Trabalho	16
1.3 Metodologia.....	17
1.4 Visão Geral do Projeto.....	18
1.5 Estrutura da Monografia.....	19
CAPÍTULO 2 – MOTIVAÇÃO	221
2.1 Segurança	221
2.2 Conforto.....	221
2.3 Soluções Existentes	22
2.4 Benefícios do Dispositivo Proposto e Suas Restrições.....	23
CAPÍTULO 3 – REFERENCIAL TEÓRICO	25
3.1 Domótica	25
3.2 Microcontroladores.....	26
3.2.1 Arduino.....	26
3.3 Sensor	27

3.3.1 Sensor PIR.....	28
3.4 Radiação Infravermelha.....	28
3.5 Rede GSM	29
3.6 SMS.....	30
3.7 Cartão SIM	30

CAPÍTULO 4 – DESCRIÇÃO DO HARDWARE E SOFTWARE 32

4.1 Arduino UNO.....	32
4.1.1 Especificações.....	332
4.1.2 Pinagem Utilizada.....	34
4.2 Sensor PIR	34
4.2.1 Pinagem.....	36
4.3 Módulo GSM SIM900.....	37
4.3.1 Especificações.....	37
4.3.2 Pinagem.....	40
4.4 Comandos AT	42
4.5 Arduino IDE.....	4 Erro! Indicador não definido.

CAPÍTULO 5 – IMPLEMENTAÇÃO 45

5.1 Modelagem do Sistema	45
5.2 Elaboração do Código Fonte	46
5.2.1 Código Teste do Sensor PIR.....	47
5.2.2 Código Fonte Final.....	49
5.3 Montagem do Protótipo	52

CAPÍTULO 6 – RESULTADOS OBTIDOS.....	54
6.1 Simulações.....	54
6.2 Problemas Encontrados.....	57
6.3 Orçamento do Projeto	58
CAPÍTULO 7 – CONSIDERAÇÕES FINAIS	59
7.1 Conclusões.....	59
7.2 Propostas para Trabalhos Futuros.....	59
REFERÊNCIAS BIBLIOGRÁFICAS	61
APÊNDICE A – LINK DE VÍDEO DO AUTOR	63
APÊNDICE B – CÓDIGO FONTE TESTE DO SENSOR PIR.....	63
APÊNDICE C – CÓDIGO FONTE FINAL.....	65
APÊNDICE D – ESQUEMÁTICO ARDUINO UNO.....	67

LISTA DE FIGURAS

Figura 1.1 – Diagrama Esquemático do Projeto	19
Figura 2.1- Módulo do Alarme Disc Cell 5 Ultra	22
Figura 2.2– Circuito do Disc Cell 5 Ultra.....	23
Figura 3.1 – Domótica.....	25
Figura 3.2 – Placa Arduino Uno	27
Figura 3.3– Cartão SIM Utilizado no Projeto	31
Figura 4.1 – Placa Modelo Arduino UNO	32
Figura 4.2– Sensor PIR frente	35
Figura 4.3 – Sensor PIR trás.....	35
Figura 4.4 – Circuito Eletrônico do Sensor PIR	36
Figura 4.5 – Indicação de pinos do sensor PIR.....	36
Figura 4.6 – Módulo GSM SIM900	37
Figura 4.7 – Pinagem e Ligações Utilizadas	40

Figura 4.8 – Interface do Arduino IDE.....	44
Figura 5.1 – Protótipo Concluído.....	45
Figura 5.2 – Fluxograma do Projeto.....	46
Figura 5.3 – Código Fonte para o teste do sensor PIR.....	48
Figura 5.4 – Código Fonte: Diretrizes e Variáveis	50
Figura 5.5 – Código Fonte: Função do Sensor de Presença	51
Figura 5.6 – Código Fonte: Função Setup.....	51
Figura 5.7 – Código Fonte: Função Loop	52
Figura 6.1 – Monitor Serial: Simulação com o Protótipo Final.....	54
Figura 6.2 – Entrega das Mensagens SMS.....	56

LISTA DE TABELAS E QUADROS

Quadro 4.1 – Principais Características da placa Arduino UNO	33
Quadro 4.2 – Pinagem da placa Arduino UNO utilizada nesse projeto	34
Tabela 4.1 – Pinagem Módulo GSM SIM900	41
Tabela 4.2– Comandos AT utilizados	42
Quadro 4.3– Sintaxe dos comandos AT	43
Quadro 6.1 – Custo do Projeto	58

LISTA DE ABREVIATURAS E SIGLAS

AT	Hayes AT Commands
bps	bit por Segundo
cm	centímetros
CDMA	Code Division Multiple Access
GHz	Gigahertz
GSM	Global System for Mobile Communications
IDE	Integrated Development Enviroment
LED	Diodo Emissor de Luz
MHz	Megahertz
PIR	Passive Infrared
SIM	Subscriber Identification Module
SMS	Short Message Service
THz	Terahertz
USB	Universal Serial Bus
V	Volts

CAPÍTULO 1 – INTRODUÇÃO

1.1 Apresentação do Problema

Nos últimos anos, tem sido uma tendência a utilização de tecnologia no dia a dia do ser humano, em diversos setores e áreas de sua vida. No que diz respeito à tecnologia no controle e automação residencial, há diversas ofertas de produtos para um mercado amplo e crescente, para quem pode pagar por mais comodidade e segurança.

Dentre as diversas tecnologias que vêm sendo trazidas às residências, o alarme sensorial é uma dessas tecnologias mais simples e presentes em casas e apartamentos. Essa tecnologia permite identificar qualquer invasão de área de sua residência, trazendo mais segurança ao usuário, que sabe que será alertado. No entanto, quando o alarme for acionado e acender luzes ou tocar sirenes, de nada vai adiantar se o usuário não estiver presente no momento do alerta.

O usuário tem que estar ciente que o alarme foi disparado, tanto quando estiver presente, ou ausente de sua casa. Assim, ele poderá tomar as providências necessárias para proteger sua residência. Este projeto propõe uma solução para este tipo de problema, contribuindo com um alerta SMS para o celular do usuário, assim que o alarme for disparado. Além disso, o baixo custo e a mobilidade do alarme proposto nesse trabalho oferece também, comodidade ao usuário.

1.2 Objetivos do Trabalho

Este trabalho tem como objetivo geral apresentar uma proposta de solução para o problema apresentado na seção anterior, em relação ao aviso de alerta ao usuário, mesmo ausente, com a criação de um dispositivo microcontrolado que funcionará como alarme presencial e avisará o usuário onde ele estiver por mensagem SMS. Esse dispositivo microcontrolado, também será de pequeno porte, facilitando assim, sua mobilidade.

Como objetivo específico, neste trabalho é desenvolvido um protótipo simulando um alarme presencial, que tem como função detectar movimento em um limite de perímetro através de um sensor PIR. Utilizando a placa microcontrolada, Arduino UNO, juntamente com o módulo GSM IComsat SIM900, o sensor PIR, além de acionar um LED indicador de que houve detecção de movimento, também acionará uma sirene e através do envio de uma mensagem SMS, o microcontrolador avisará ao usuário sobre a detecção do alarme.

O módulo GSM terá a função de se comunicar com o microcontrolador Arduino UNO através de um código fonte, e assim atuará como dispositivo de envio de mensagem SMS. A mensagem SMS enviada pelo microcontrolador, será realizada a partir de um cartão GSM com linha telefônica e permissão da operadora para envio de mensagens SMS.

1.3 Metodologia

Para o desenvolvimento desse projeto, foram realizadas pesquisas bibliográficas como ponto de partida para o desenvolvimento da ideia do protótipo. Foram feitas pesquisas no site oficial do microcontrolador escolhido para a determinação dos componentes de *hardware* - como a placa Arduino UNO e o módulo GSM SIM900 - e *software* - o ambiente de desenvolvimento Arduino (IDE) – que foram utilizados ao longo do desenvolvimento do projeto.

Foram realizados estudos sobre a tecnologia da telefonia móvel, como a rede GSM e o envio de mensagens de texto, e também sobre as características físicas dos componentes usados no projeto. Além disso, todos os componentes eletrônicos foram estudados através de manuais e sites oficiais para o desenvolvimento do projeto. Esses componentes foram testados através de um multímetro, localizado no laboratório do bloco V do UniCEUB com a ajuda de um monitor.

O desenvolvimento do sistema teve como início a elaboração de um fluxograma geral do programa, que serviu como base para a escrita do código fonte, posteriormente, compilado e carregado para o microcontrolador já montado com seus componentes.

Os testes foram realizados com chips de duas operadoras diferentes com o objetivo verificar e otimizar o código fonte.

Após a etapa inicial da elaboração de um fluxograma, que serviu como diretrizes para o desenvolvimento do projeto, como segunda etapa, foram feitos os primeiros testes com a placa Arduino UNO. Estes testes consistiram em escrever códigos básicos para o entendimento das funções usadas na interface Arduino IDE. Na sequência, foi testado um código de ativação do sensor PIR, onde este detectava movimento e mandava a resposta para o microcontrolador.

Na última etapa do processo, quando já havia familiarização com a placa microcontroladora e sua interface de *software*, começaram a serem feitos os testes utilizando o módulo GSM acoplado no microcontrolador Arduino UNO. Estes testes já visavam o código fonte final, com a função de envio de SMS por parte do microcontrolador, testando bibliotecas diferentes, até ser encontrada uma biblioteca que atendesse a necessidade do projeto.

1.4 Visão Geral do Projeto

O projeto basicamente é constituído por módulo GSM (IComsat SIM900), microcontrolador (Arduino UNO), sensor de presença (PIR), e demais componentes eletrônicos, tais como LEDs e *buzzers*.

Além dos componentes citados, foi escrito um código fonte para a comunicação entre as placas.

O diagrama esquemático do projeto proposto pode ser observado na Figura 3.3, na qual representa de forma objetiva a composição geral do projeto.

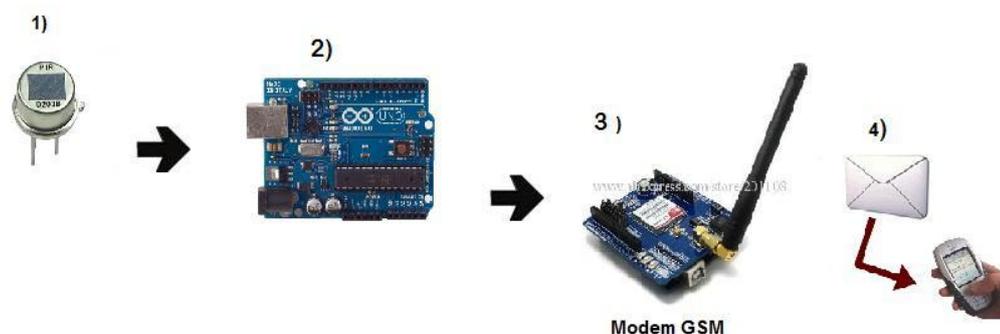


Figura 3.3 – Diagrama Esquemático do Projeto (FONTE: Autor)

- 1) O sensor PIR detecta o movimento;
- 2) O Arduino UNO processa a informação de detecção;
- 3) O modem GSM é ativado e envia o SMS;
- 4) A mensagem SMS chega ao celular.

1.5 Estrutura da Monografia

Esta monografia é dividida em sete capítulos, incluindo a INTRODUÇÃO, que trata da introdução ao tema proposto, a motivação do projeto, os principais objetivos, e essa seção que descreve toda a estrutura da monografia.

No segundo capítulo, APRESENTAÇÃO DO PROBLEMA, é apresentada uma descrição aprofundada sobre as questões motivacionais do trabalho, as soluções existentes atualmente sobre o tema proposto. A última seção deste capítulo aborda de forma resumida os benefícios do dispositivo proposto e suas restrições.

O terceiro capítulo, REFERENCIAL TEÓRICO, trata de assuntos como sensor, microcontroladores, GSM de forma conceitual. Nesse capítulo é apresentada também uma visão geral do projeto.

O quarto capítulo, DESCRIÇÃO DO HARDWARE E SOFTWARE, aborda as especificações dos dispositivos utilizados, onde é detalhada a especificação dos componentes de controle de comunicação serial, do microcontrolador PIC utilizado, detalhamento do sensor, bem como a configuração do módulo GSM, dentre outros dispositivos e *softwares*.

No quinto capítulo, IMPLEMENTAÇÃO, é apresentado o desenvolvimento do projeto, como a integração entre o *hardware* e *software*, desenvolvimento do código na linguagem C e outras informações.

No sexto capítulo, RESULTADOS OBTIDOS abordam toda a parte de teste do protótipo, ajustes de problemas encontrados e suas dificuldades.

No sétimo capítulo, CONSIDERAÇÕES FINAIS, são apresentadas a conclusão e as sugestões para trabalhos futuros.

CAPÍTULO 2 – MOTIVAÇÃO

Este capítulo tem o objetivo de detalhar as questões que motivaram o trabalho, como: segurança e comodidade, que foi apresentado resumidamente na seção — Apresentação do Problema — do capítulo anterior.

Serão apresentadas também algumas soluções existentes para estas questões, no que diz respeito à alarmes sensoriais para residência. E por fim, são apresentados os benefícios do dispositivo proposto pelo trabalho e suas restrições.

2.1 Segurança

Não é de hoje que se observa o aumento de furtos e invasões a residências. Os veículos de comunicação mostram relatos e números crescentes de infrações todos os dias.

“O número de assaltos a residências e homicídios no DF cresceu em 2009. Segundo dados da Secretaria de Segurança Pública, os roubos a casas subiram 40% de 2008 para 2009...” (CORREIO BRAZILIENSE, 2010)

Segundo o site de notícias G1 (2011), “o número de furtos em casas do Lago Norte, bairro nobre de Brasília, quase dobrou no mês de Julho. Foram 14 casas furtadas no mês, seis a mais do que no mesmo período do ano passado.”.

Por esses motivos, muitos cidadãos têm procurado melhorar a segurança de seus lares. O meio proposto por este projeto, é através de um alarme sensorial.

2.2 Comodidade

O avanço tecnológico contínuo é observado nas mais diversas áreas: medicina, construção civil, engenharia de computação; e com isso, a tecnologia também tem ocupado lugar cada vez mais proeminente em nossas residências.

O protótipo resultante deste projeto tem como objetivo também trazer mais comodidade ao usuário avisando-o de um possível perigo em sua casa, mesmo ele estando ausente.

2.3 Soluções Existentes

Em uma breve pesquisa, não é difícil encontrar no mercado alarmes residenciais com um módulo GSM que permita a opção de envio de mensagem SMS quando o alarme for ativado. A figura 2.1 abaixo, ilustra um modelo de alarme:



Figura 2.1 – Módulo do Alarme Disc Cell 5 Ultra.

Fonte: (http://img2.mlstatic.com/kit-alarme-residencial-discadora-gsm-liga-e-envia-sms_MLB-O-3181425062_092012.jpg)

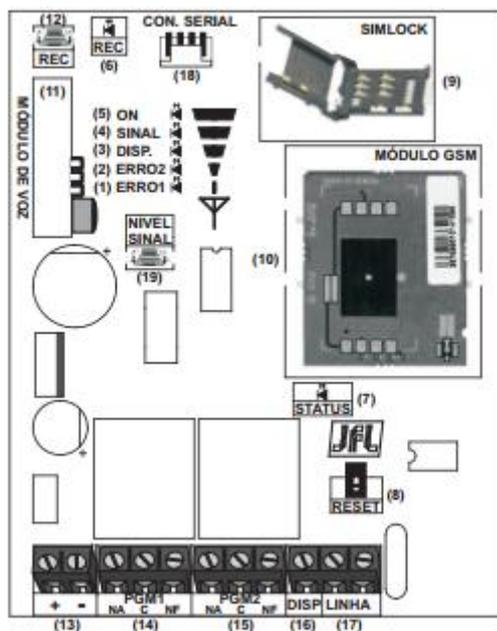


Figura 2.2 – Circuito do Disc Cell 5 Ultra.

Fonte: (<http://jflalarmes.tecnologia.ws/uploads/download-seguranca-eletronica-discadoras-disc-cell-5-ultra.pdf>)

O modelo representado acima é o alarme residencial Disc Cell 5 Ultra, da empresa JF Alarmes. Este modelo possui um módulo de discagem para até 5 números de telefone, com cada chamada durando aproximadamente 1 minuto, onde uma vez discado o número, uma mensagem de voz será repetida continuamente alertando sobre o disparo do alarme. Além disso, há uma opção para enviar mensagens SMS para até 5 números de celular, indicando que houve disparo do alarme.

2.4 Benefícios do Dispositivo Proposto e Suas Restrições

A finalidade deste projeto é apresentar um dispositivo de detecção de movimento através de um sensor PIR com aviso por SMS, utilizando para isso equipamentos computacionais simples, de fácil acesso e com custo baixo.

O projeto não tem como funcionalidade de tratar erros do sensor ou até armazenar os dados recebidos pelo mesmo. O protótipo somente fará a detecção do movimento, reagindo de forma a enviar a mensagem SMS.

O principal benefício do dispositivo será a possibilidade de ser avisado de uma invasão de perímetro de sua residência por um telefone e ter o *feedback*¹, informado através de uma mensagem SMS se a segurança de sua casa está comprometida, por um custo bem mais baixo do que o encontrado no mercado, e pela mobilidade do protótipo devido ao pequeno porte.

¹ *Feedback* é o procedimento que consiste no provimento de informação à uma pessoa sobre uma ação executada por esta, objetivando informar sobre as ações executadas anteriormente.

Neste trabalho, o foco da domótica é no âmbito da segurança residencial. A domótica atuar em diversos níveis de segurança, sendo o mais comum, o sistema de sensores. Neste projeto, o sistema será auxiliado por um sensor de presença, visualizando a detecção de intrusos.

3.2 Microcontroladores

Microcontroladores são usados frequentemente em tarefas específicas que não exijam grandes quantidades de dados, como em automação residencial, pois geralmente possuem pouca capacidade de armazenamento.

O microcontrolador é um dispositivo semiconductor em forma de circuito integrado, que integra as partes básicas de um microcomputador - microprocessador, memórias não-voláteis e voláteis e portas de entrada e saída. Geralmente, é limitado em termos de quantidade de memória, principalmente no que diz respeito à memória de dados, é utilizada em aplicações específicas, ou seja, naquelas que não necessitam armazenar grandes quantidades de dados. Apresenta um custo bastante baixo e possui vários fornecedores. (GIMENEZ, 2002, p.4).

Alguns dos fornecedores mencionados acima são: Hitachi, Intel, Microchip, Mitsubishi, Motorola, Philips, Toshiba etc.

3.2.1 Arduino

O Arduino é uma plataforma de computação física de fonte aberta, com base em uma entrada simples de entrada/saída (I/O, ou input/output), assim como em um ambiente de desenvolvimento que implementa a linguagem *Processing*¹. O Arduino pode ser utilizado para desenvolver objetos interativos independentes, ou conectados a *softwares* de seu computador (BANZI, 2011).

¹ *Processing* é uma linguagem de programação de código aberto que foi inicialmente desenvolvida para ensinar fundamentos na programação computacional num contexto visual.



Figura 3.2 – Placa Arduino Uno. Fonte: ARDUINO

O Arduino foi escolhido por se tratar de um ambiente multiplataforma, ou seja, pode ser executado em Windows, MAC OS ou Linux. Além dessa vantagem, o Arduino tem como base o IDE de programação *Processing*, um ambiente de programação amigável e de fácil utilização. Finalmente, o Arduino UNO pode ser programado através de cabo USB, sem a necessidade de se utilizar porta serial.

3.3 Sensor

É um termo utilizado para indicar dispositivos sensíveis a alguma forma de energia do ambiente que pode ser luminosa, térmica, cinética, relacionando informações sobre uma grandeza que pode ser medida, como: temperatura, pressão, velocidade, corrente, aceleração, posição, etc.(THOMAZINI e ALBUQUERQUE, 2005, p.17).

Sensor é um termo empregado para designar dispositivos sensíveis a alguma forma de energia do ambiente. Geralmente o sinal de saída é manipulado antes de sua leitura no sistema de controle. (THOMAZINI e ALBUQUERQUE, 2005, p.17).

O sensor utilizado no projeto, trata-se de um sensor de presença PIR (*Passive Infrared*), que detecta objetos que geram calor e radiação infra-vermelha cujo comprimento de onda for igual ou maior que $0.9 \mu\text{M}$. Essa onda ainda passa através de um polietileno, material utilizado na lente do sensor, para concentrar a radiação sobre o sensor.

3.3.1 Sensor PIR

O sensor PIR é um sensor infravermelho passivo, e é um importante elemento na detecção de invasão e intrusões em um ambiente, pois não se trata de um sensor de movimento, mas sim, um sensor de variação de temperatura.

Assim como outros sensores infravermelhos, o sensor PIR é calibrado para a temperatura do corpo humano.

A denominação de infravermelho passivo deve-se ao fato do sensor apenas captar variações de irradiação de luz infravermelha, ou seja, variações de temperatura. Todos os objetos emitem certa quantidade de luz infravermelha, e uma mudança repentina de quantidade de luz infravermelha num ambiente, como por exemplo, a entrada de uma pessoa no ambiente, causará detecção por parte do sensor.

Por ser um sensor infravermelho, o sensor PIR possui um recurso de compensação de temperatura. Este recurso permite que o sensor tenha sua sensibilidade ajustada de acordo com a temperatura ambiente, portanto, quanto mais alta for a temperatura ambiente, maior será a sensibilidade do sensor.

A área de atuação do sensor PIR é dimensionada por dois fatores. O primeiro fator é a abertura horizontal de sua lente, e o segundo fator é a distância máxima de sua atuação. O modelo de sensor PIR utilizado neste projeto, o BISS0001, possui um ângulo de abertura horizontal de 120° e alcance máximo de 5m.

Ainda, existe no mercado, tipos de sensores infravermelho especiais, com recursos de imunidade a animais de pequeno porte, sendo assim, não detectam presença quando houver movimentação desses animais.

3.4 Radiação Infravermelha

A radiação conhecida como infravermelha é uma radiação não ionizante, ou seja, uma radiação com frequência menor ou igual à frequência da luz, e não possui efeitos danosos ao corpo humano.

Dentro do espectro de luz visível, o infravermelho localiza-se ao final da cor vermelha, adjacente aos comprimentos de onda longos, por isto recebe o nome de infravermelho.

A radiação transmitida pelo infravermelho pode ser percebida através do calor do corpo humano, por terminações nervosas na pele conhecidas como termorreceptores.

Outra característica marcante da radiação infravermelha é sua alta frequência periódica, de 300 GHz a 400 THz, que a caracteriza como alta emissora de oscilações eletromagnéticas por segundo.

3.5 Rede GSM¹

Criada em 1982, a rede GSM é uma tecnologia digital para celulares usada para transmissão móvel de voz e dados. A rede GSM permite que indivíduos sejam contatados através no mesmo número de celular em até 219 países. As redes GSM terrestres cobrem mais de 90% da população mundial, e a tecnologia *roaming*² GSM de satélites estende essa cobertura para áreas em que a rede terrestre não alcança. Hoje, as conexões móveis globais superam 6 bilhões (GSM, 2012).

A rede GSM foi criada com o objetivo de padronizar o sistema de telefonia móvel acabando com a incompatibilidade de redes. Que ocorreu devido ao crescimento desenfreado da produção dos celulares analógicos durante os anos 80 onde cada país tinha seu próprio sistema de envio de dados, protocolos e frequência de comunicação. Um grupo de estudos Europeu foi criado para desenvolver um sistema móvel padrão o *Group Special Mobile* (GSM).

A rede GSM é uma tecnologia digital para celulares usada para transmissão de voz e serviços de dados móveis. Suporta chamadas de voz e dados como o

¹ GSM é a sigla em inglês de *Global System for Mobile Communications* (Sistema Global para Comunicações Móveis), antiga *Group Special Mobile*.

² *Roaming* é um termo empregado na tecnologia móvel que designa a capacidade de um usuário de uma rede obter conectividade em áreas fora da localidade geográfica em que está registrado.

envio de SMS, com velocidade de transferência de até 9,6 Kbps. (GSM WORD, 2012, traduzido).

O sistema GSM 900 utiliza dois conjuntos de frequência na banda dos 900 MHz: o primeiro nos 890-915 MHz, utilizado para transmissões de terminal, e o segundo nos 935-960 MHz, utilizado para as transmissões da rede, e utilizado neste projeto.

3.6 SMS

Atualmente, as mensagens de texto (*Short Message Service*) estão presentes no cotidiano do usuário de celular como uma forma barata e rápida de se comunicar. Segundo o site de estudos estatísticos Statista, durante o ano de 2010 mais de 192 mil mensagens de texto foram enviadas a cada segundo em todo o planeta.

“O Serviço de Mensagens Curtas (*Short Message Service*) permite aos usuários enviarem e receberem mensagens a partir de um aparelho de telefone celular. Cada mensagem pode ser de até 160 caracteres e ser enviada e recebida por usuários de diferentes redes de operadoras.” (CAMPOS, 2010).

Este serviço foi criado para funcionar com a tecnologia digital GSM, seu principal objetivo era um sistema simples de mensagens que funcionasse mesmo quando os aparelhos móveis dos usuários estivessem fora da área de cobertura ou desligados.

3.7 Cartão SIM¹

O cartão SIM é um chip de memória portátil usado principalmente em telefones celulares que operam em redes GSM. Estes cartões SIM possuem memória do tipo EEPROM², onde são guardados informações pessoais do titular da conta de

¹ SIM é a sigla de *Subscriber Identity Module* (Módulo de Identificação do Assinante).

²EEPROM (*Electrically-Erasable Programmable Read-Only Memory*) – é um *chip* de armazenamento não-volátil, podendo ser programada e apagada várias vezes, eletricamente.

telefone, incluindo o número de telefone, contatos, mensagens de texto e outros dados. Quando um usuário quer mudar de aparelho, ele pode facilmente remover o chip de um aparelho e inseri-lo em outro.

Os cartões SIM são uma parte fundamental do desenvolvimento da tecnologia dos celulares, pois antigamente somente existia a rede CDMA onde o aparelho que guardava as informações do usuário e para realizar a troca de aparelho, o usuário tem que ir à loja da operadora para transferir o número para outro aparelho. (Wise Geek, 2012, traduzido e modificado).

Na Figura 3.3 ilustra o cartão SIM utilizado no projeto. Cartão SIM da operadora Claro, com 128 KB de memória e 25 mm por 15 mm de tamanho.



Figura 3.3 - Cartão SIM utilizado no projeto (FONTE: Autor)

CAPÍTULO 4 – DESCRIÇÃO DO HARDWARE E SOFTWARE

Este capítulo explica as especificações e pinagens dos dispositivos usados e seu devido funcionamento nesse projeto, tanto a parte física, ou seja, o *hardware*, como a parte lógica, os *softwares* utilizados.

4.1 Arduino UNO

O microcontrolador Arduino UNO, é a mais recente dentre as séries de placas Arduino utilizando USB. Essa placa difere das outras placas da família Arduino, por não utilizar o chip conversor FTDI USB para Serial, facilitando o *upload*¹ do código fonte do computador para a placa. O esquemático da placa pode ser encontrado no Anexo A – Esquemático: Arduino UNO.

4.1.1 Especificações

A placa Arduino UNO é baseada no microcontrolador ATmega328, com produção pela Atmel. A figura 4.1 ilustra a placa Arduino UNO, mesmo modelo utilizado nesse projeto.

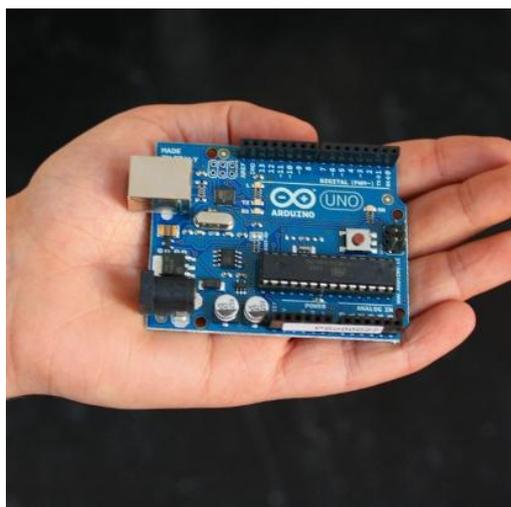


Figura 4.1 – Placa Modelo Arduino UNO. Fonte: ARDUINO UNO

¹ *Upload*, ou carregamento, é a transferência de dados de um computador local para outro, com a placa Arduino.

As principais características do Arduino UNO são representadas no quadro 4.1 abaixo:

Quadro 4.1 – Principais características da placa Arduino Uno.

Nome	Descrição
Microcontrolador	ATmega328
Voltagem de operação	5V
Alimentação (recomendada)	7 – 12V
Alimentação (limite)	6 – 20V
Pinos de E/S digitais	14 (de 0 a 13, dos quais 6 podem prover saída PWM)
Pinos de entrada analógica	6 (de A0 até A5)
Corrente contínua por pino de E/S	40mA
Corrente contínua para pino 3,3V	50mA
Memória flash	32kb (ATmega328)
SRAM	2kb (ATmega328)
EEPROM	1kb (ATmega328)
Frequência de <i>clock</i> ¹	16MHz

Fonte: MAIA, 2012

Essa placa possui 3 pinos GND (terra), dois pinos que fornecem voltagem regulada de 3,3V e 5V cada, um pino Vin que fornece voltagem advinda diretamente da alimentação – seja por USB ou por fonte externa – e um pino de entrada para referência analógica (AREF).

¹ Frequência de *clock* representa o número de ciclos realizados pelo microcontrolador no período de um segundo.

4.1.2 Pinagem Utilizada

O quadro 4.2 detalha a pinagem utilizada, descrevendo o pino e sua funcionalidade específica neste projeto.

Quadro 4.2 – Pinagem da placa Arduino UNO utilizada nesse projeto.

Pino	Utilização
5V	Alimentação do sensor PIR
GND	Aterramento do sensor PIR
7	Entrada do sensor PIR
13	Entrada do LED
GND	Aterramento do LED
10	Entrada do <i>buzzer</i>
GND	Aterramento do <i>buzzer</i>
Vin	Alimentação do módulo GSM

Fonte: AUTOR

4.2 Sensor PIR

O sensor utilizado para esse projeto, é o sensor de presença PIR (Passive Infrared). O sensor PIR tem um circuito eletrônico capaz de amplificar os sinais e assim poder modular um sinal de saída em nível digital. São circuitos formados com o componente BISS0001 que em sua saída geram um sinal "on" "off" para pode acionar ou desligar um circuito externo de alta potência.

As figuras 4.2 e 4.3 ilustram o circuito do sensor PIR modelo BISS0001, utilizado no projeto.



Figura 4.2 – Sensor PIR frente. Fonte: AUTOR

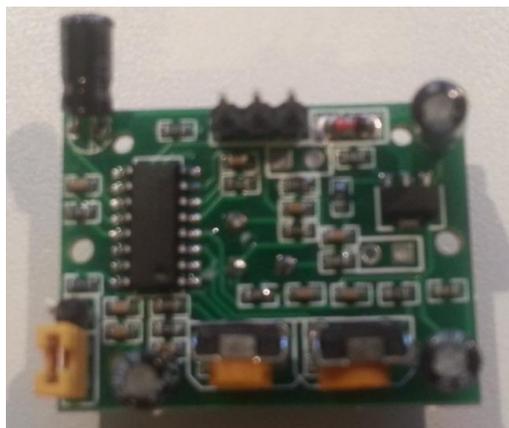


Figura 4.3 – Sensor Pir trás. Fonte: AUTOR

4.2.1 Pinagem

O sensor PIR já vem com o circuito eletrônico necessário para detecção de movimento e acionamento de cargas, com o chip BISS0001, regulador, diodo e três pinos: um para aterramento, outro para saída digital, e outro para alimentação, como mostrado nas figuras 4.4 e 4.5 abaixo.

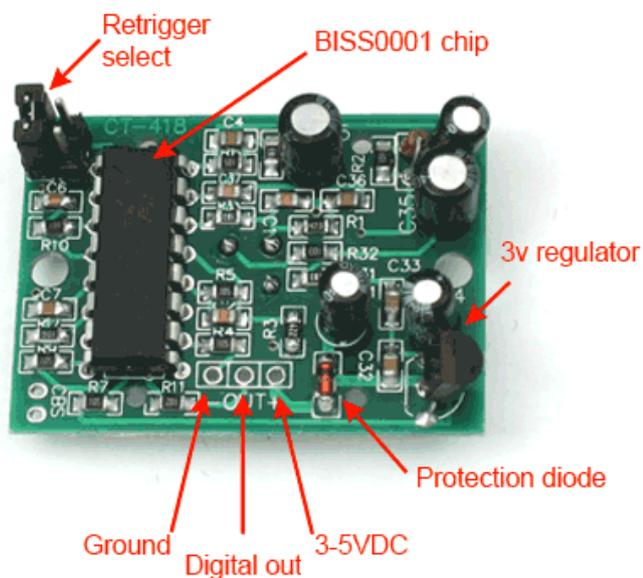


Figura 4.4 – Circuito Eletrônico do sensor PIR. (Fonte: BISS0001 Datasheet)

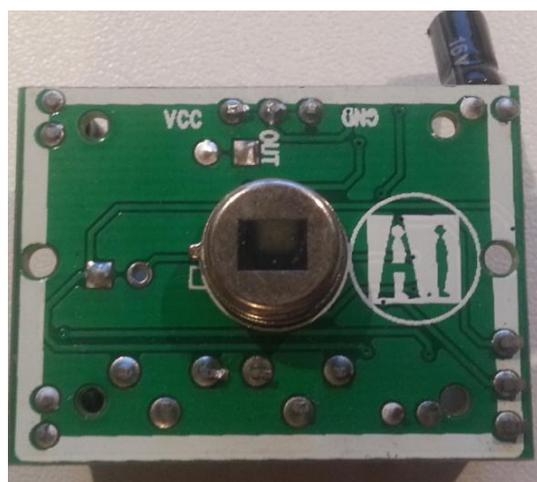


Figura 4.5 – Indicação de pinos do Sensor PIR. Fonte: AUTOR

4.3 Módulo GSM SIM900

O módulo GSM IComsat SIM900, também chamado de *shield*¹ GSM, é uma placa que pode ser conectada facilmente a placa Arduino UNO, estendendo a esta, as funcionalidades do módulo GSM.

O IComSat é um shield GSM/GPRS baseado no modulo Quad-band GSM/GPRS SIM900. Ele é controlado via comando AT (GSM 07.07, 07.05 e comando AT melhorado SIMCOM²), é um módulo totalmente compatível com vários microcontroladores, porém foi produzido para uso fácil com Arduino. (SIM 900 Data Sheet, IComSat SIM 900). A Figura 4.6 ilustra o modem GSM SIM900.



Figura 4.6 – Módulo GSM SIM900 (Fonte: SIM 900 Data Sheet, IComSat SIM 900)

4.3.1 Especificações

O SIM900 é um módulo GSM/GPRS quadband com pilha TCP/IP e pode ser usado em aplicações onde a transmissão via tecnologia GSM/GPRS se faça ne-

¹ Shield provém do inglês, escudo. Neste contexto, é uma placa empilhável compatível com a placa Arduino UNO.

² SIMCOM *Wireless Solutions* – é líder na indústria fornecendo módulos sem fio de alta qualidade para diferentes plataformas de tecnologias em redes. (Página na Web: <http://wm.sim.com>).

cessária; seja, transmissão de voz, SMS - mesmo assim, consome pouca energia. Seu reduzido tamanho o torna ideal para os mais exigentes requisitos das aplicações industriais, como M2M¹, telemetria ou qualquer outra forma de comunicação móvel. O módulo é fabricado pela SIMCOM e distribuído no Brasil pela ME Componentes². As principais características do módulo SIM900 são: (SIM 900 Data Sheet, IComSat SIM 900).

- Quad-Band 850/900/1800/1900MHz;
- GPRS classe multi-slot 10/8;
- GPRS estação móvel classe B;
- Compatível com o GSM fase 2/2+;
- Classe 4 (2W@850/900MHz);
- Classe 1 (1W@1800/1900MHz);
- Controle via comandos (GSM 07.07, 07.05 e SIMCOM Comandos AT melhorado);
- SMS(Short Message Service);
- Porta serial livre para seleção;
- Todos os módulos SIM900 conexões para pinos de saídas;
- RTC³ suportado com Super Capacitor;
- Power on/off e reset de funções suportadas pela interface Arduino;
- Indicadores de PWR, LED de estado e LED de estado de rede;

¹ Máquina a máquina (M2M) – é a utilização um dispositivo para capturar um evento que é retransmitido para um aplicativo que traduz o evento capturado em informações significativas.

² ME Componentes e Equipamentos – é uma empresa fundada em 1997, que trabalha com todos os tipos e com as melhores marcas de componentes eletrônicos, tais como módulos GPS, módulos GSM/GPRS, circuitos integrados. (Página na Web: <http://www.mecomp.com.br>).

³ RTC (*Real-time clock*) – Relógio de Tempo Real é um circuito integrado que mantém o controle do tempo presente, utilizado para evitar confusão com um outro “clock”.

- Tamanho da PCB: 77,2mm X 66 mm X 1,6mm;
- Alimentação Externa: 9 a 20 volts DC(Compatível com Arduino);
- Protocolo de comunicação: UART¹;
- RoHS: Sim.

¹ UART (*Universal Asynchronous Receiver/Transmitter*) – tem a finalidade de converter dados paralelos para a forma serial e vice versa.

4.3.2 Pinagem

Na Figura 4.7 é mostrada a pinagem do modem GSM/GPRS SIM900 e a troca de “jumpers” para realizar a configuração do modem. A Tabela 4.1 descreve todos os pinos do modem.

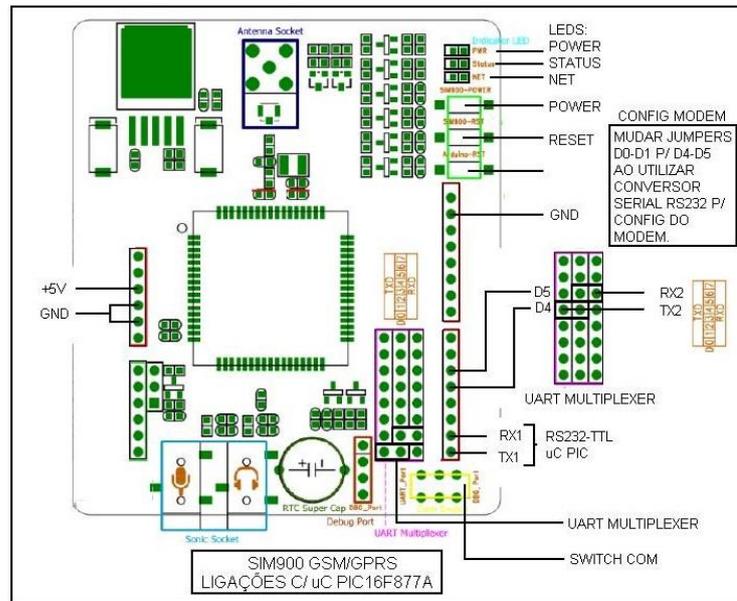


Figura 4.7 – Pinagem e Ligações Utilizadas (Fonte: SIM 900 Data Sheet, adaptado)

Tabela 4.1 –Pinagem módulo GSM SIM900

Interface	Pino	Descrição
GPIOs	1	VDD*
	2	GND
	3	GPIO1
	4	GPIO2
	5	GPIO3
	6	GPIO4
	7	GPIO5
	8	GPIO6
	9	GPIO7
	10	GPIO8
	11	GPIO9
	12	GPIO10
	13	GPIO11
	14	GPIO12
	15	GND
	16	VDD*
UART	1	GND
	2	DTR
	3	RI
	4	DCD
	5	CTS
	6	RTS
	7	TXD
	8	RXD
IIC&PWM	1	GND
	2	GND
	3	IIC_SCL
	4	IIC_SDA
	5	PWM2
	6	PWM1
	7	GND
	8	GND
Debug_Port	1	GND
	2	PERKEY
	3	DBG_RXD
	4	DBG_TXD

VDD* = 3.0 V

Nota: O nível de operação da porta? É de 3.0V.

FONTE: Datasheet Módulo GSM/GPRS

4.4 Comandos AT

O comando AT é um conjunto de linhas de comando que se comunica com o módulo GSM e é utilizado para configurar este módulo. Um dos serviços do módulo GSM que foram configurados neste projeto é o envio de SMS através da rede de celular com tecnologia GSM. Os comandos AT utilizados para a configuração do modem GSM SIM 900 e suas descrições são apresentados na Tabela 4.2.

Tabela 4.2 – Comando AT Utilizados

Comando	Descrição
AT <ENTER>	Verifica se o modem está conectado corretamente. Receberá a resposta OK na tela.
AT+IPR=<RATE>	Define a taxa de transferência de dados. No lugar do <RATE> é colocado a taxa de bauds por segundo.
AT+CPIN="xxxx" <ENTER>	Libera o uso do SIM CARD no modem. Usado somente caso o chip possua PIN, O número "xxxx" deverá ser substituído pelo PIN referente ao SIM CARD.
AT+CMGF=1 <ENTER>	Estabelece a condição de enviar mensagens de texto simples SMS. Receberá resposta OK.
AT+CSCS="+55XXXXXXXXXX"	Cadastra o número de telefone celular em que se deseja enviar a mensagem de texto. O número "+55XXXXXXXXXX" deverá ser substituído.
TESTE DO MODEM <CTRL-Z>	Deve-se digitar o texto da mensagem que deseja enviar e em seguida digitar o comando <CTRL-Z> para finalizar e concluir o envio.

FONTE: SIMCOM, 2010b, p. 6.

Ainda sobre os comandos AT, essas linhas de código seguem um padrão de sintaxe, descritos brevemente no quadro 4.3 a seguir.

Quadro 4.3 – Sintaxe dos comandos AT

Nome	Sintaxe	Descrição
Comando de Teste	AT+<comando>=?	O módulo GSM retorna uma lista de parâmetros e valores possíveis para o comando.
Comando de Leitura	AT+<comando>?	Retorna o valor do parâmetro ou o parâmetro atual
Comando de Escrita	AT+<comando>=<...>	Escreve os valores de parâmetros definidos pelo usuário.
Comando de Execução	AT+<comando>	Executa parâmetros não variáveis afetados por processos internos do módulo.

4.5 Arduino IDE

O Arduino IDE é um programa que permite a criação de *sketches*¹ para a placa Arduino, em uma linguagem simples, modelada a partir da linguagem *Processing*. Esse ambiente de desenvolvimento é responsável pelo envio do código fonte para o compilador, que converte o código para a linguagem de programação C, para então converter para linguagem de máquina, compreendida pelo microcontrolador.

¹ *Sketch*, do inglês, rascunho.

A figura 4.8 ilustra a interface do Arduino IDE.

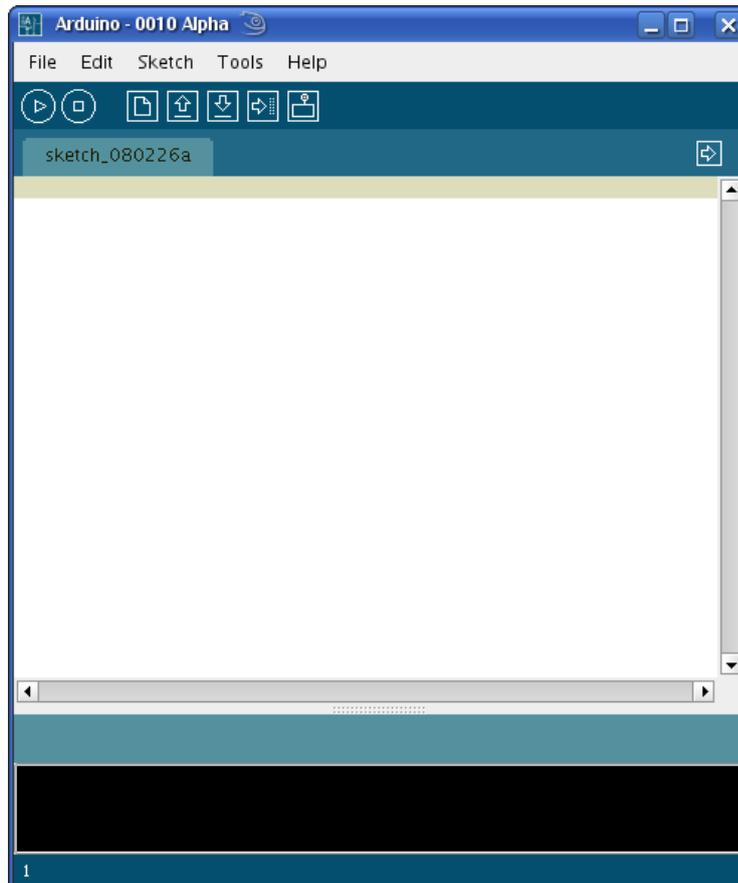


Figura 4.8 – Interface do Arduino IDE. Fonte: ARDUINO

Os botões próprios da interface têm como função:

- Compilar o código;
- Parar a compilação do código;
- Abrir um novo *sketch*;
- Carregar um *sketch* já salvo;
- Salvar o *sketch* atual;
- Fazer o *upload*² do código;
- Abrir o monitor serial, que simula a comunicação com a placa.

² *Upload*, do inglês carregar, é a função para carregar o código compilado no microcontrolador.

CAPÍTULO 5 – IMPLEMENTAÇÃO

Neste capítulo são apresentados tópicos fundamentais para compreensão geral da implementação do projeto, e definidas as etapas necessárias para isso, que são:

- Modelagem do sistema;
- Elaboração do código fonte para o microcontrolador Arduino UNO;
- Montagem do protótipo;

Na Figura 5.1 é mostrado o protótipo em sua fase final.

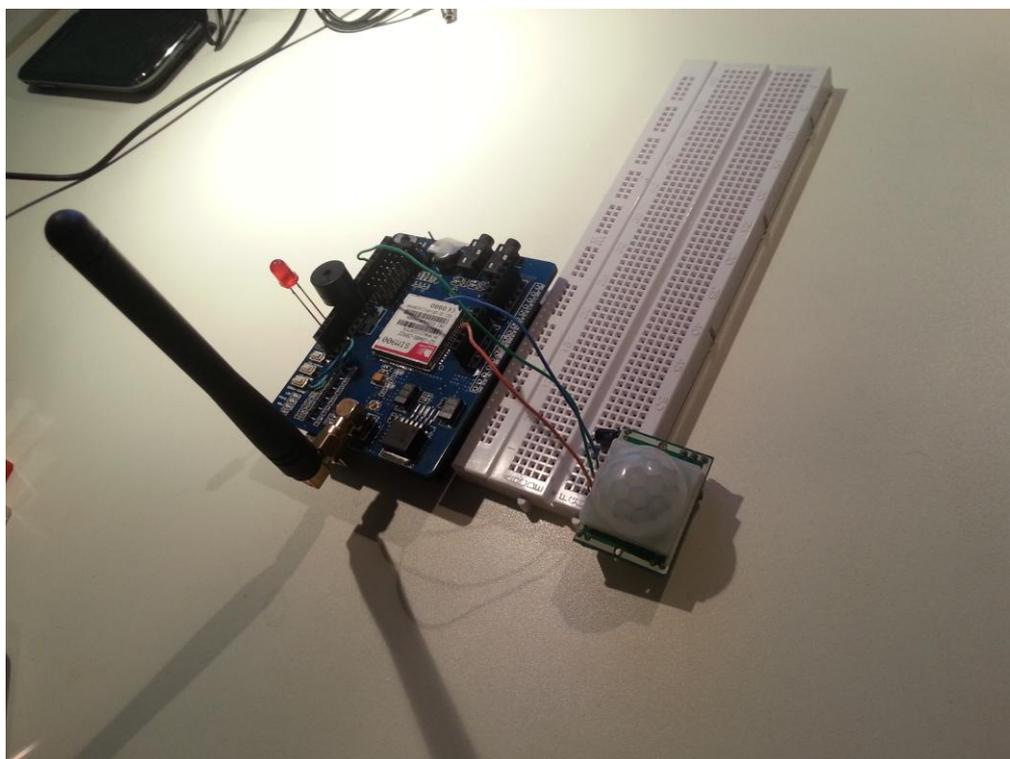


Figura 1.1 – Protótipo Concluído (Fonte: Autor)

5.1 Modelagem do Sistema

Primeiramente foi definido o que o projeto deveria ter de forma geral e quais os componentes necessários para fazer sua integração.

Após a decisão de qual microcontrolador seria utilizado, bem como qual sensor funcionaria melhor com o sistema, e quais componentes eletrônicos este

sensor ativaria junto com o modem GSM, foi feito um fluxograma geral do sistema, com suas funcionalidades. O projeto conta com um circuito impresso universal com um terminal de 640 contatos e dois distribuidores de 200 contatos, para a ligação do sensor PIR, e aterramento do *buzzer*. O fluxograma da figura 5.2 ilustra o sistema.

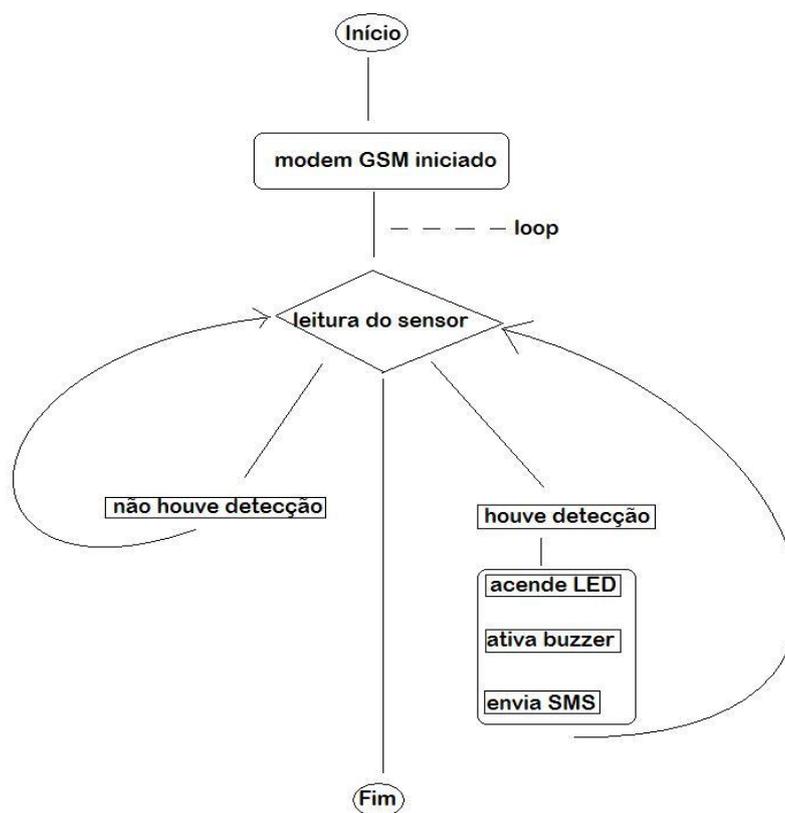


Figura 5.2 – Fluxograma do Projeto (Fonte: Autor)

5.2 Elaboração do Código Fonte

Após a modelagem do sistema, e a concepção de um fluxograma geral para ser seguido, foi possível a elaboração do código fonte compilado no microcontrolador. Nesta etapa foi feito um teste com o sensor PIR, compilando um código

apenas para testar seu funcionamento, justamente com acionamento de outros componentes eletrônicos. Só após estes testes terem sido feitos com sucesso, foi feita a elaboração do código fonte para o protótipo final, com interação entre o microcontrolador e o *shield* GSM através do sensor previamente testado. Nesta etapa será apresentado tanto o código teste do sensor PIR, como o código utilizado na compilação com o protótipo final.

5.2.1 Código Teste do Sensor PIR

O código de ativação do sensor PIR foi escrito na linguagem de programação do Arduino, que abrange *Processing* e C. O quadro 4.2, apresentado na sessão 4.1.2, que define a pinagem utilizada neste projeto, somente pode ser compreendida para o código fonte utilizado no protótipo final. Neste código, ainda, também não são usadas diretrizes ou bibliotecas específicas.

```

int ledPin = 13;
int inputPin = 2;
int pirState = LOW;
int val = 0;
int pinSpeaker = 10;

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(inputPin, INPUT);
  pinMode(pinSpeaker, OUTPUT);
  Serial.begin(9600);
}

void loop(){
  val = digitalRead(inputPin);
  if (val == HIGH) {
    digitalWrite(ledPin, HIGH);
    playTone(300, 160);
    delay(150);

    if (pirState == LOW) {

      Serial.println("Motion detected!");

      pirState = HIGH;
    }
  } else {
    digitalWrite(ledPin, LOW);
    playTone(0, 0);
    delay(300);
    if (pirState == HIGH){

      Serial.println("Motion ended!");

      pirState = LOW;
    }
  }
}
// duration in mSecs, frequency in hertz
void playTone(long duration, int freq) {
  duration *= 1000;
  int period = (1.0 / freq) * 1000000;
  long elapsed_time = 0;
  while (elapsed_time < duration) {
    digitalWrite(pinSpeaker, HIGH);
    delayMicroseconds(period / 2);
    digitalWrite(pinSpeaker, LOW);
    delayMicroseconds(period / 2);
    elapsed_time += (period);
  }
}

```

Figura 5.3 – Código Fonte para o teste do Sensor PIR (Fonte: Autor)

A figura 5.3 mostra primeiramente as definições das variáveis representantes do sensor PIR, do LED e do *buzzer*, e suas respectivas pinagens. Aqui, é

assumido que o sistema é inicializado sem detecção de movimento, e a variável de leitura do status do sensor, é assumida com o número zero.

Após as definições de pinagem e dos componentes de entrada e saída, utiliza-se a função *setup*, própria do Arduino, que define a parte do código que será executada somente na inicialização do sistema. Na função *setup* também é declarado se as variáveis são de entrada ou de saída.

Depois de o sistema ser inicializado, ele entra em repetição infinita. Essa função, no Arduino, é chamada de *loop*. No *loop* é escrito a parte do código que faz a leitura do sensor, e caso necessário, aciona seus componentes. Após a leitura do sensor, se o valor lido for alto, o LED e o *buzzer* são acionados. Caso não seja mais detectado movimento pelo sensor, a variável de leitura volta a ser assumida como baixa.

Por último, é mostrada a função de acionamento do *buzzer*. Esta função tem como único objetivo a configuração da duração em segundos do toque da buzina.

5.2.2 Código Fonte Final

Com o código teste estando funcional, foi elaborado e escrito o código fonte de integração do modem GSM.



```

sketch_ProjetoFinal2
/*=====
*****
PROJETO FINAL - Engenharia da Computação - UniCEUB
1o. Semestre de 2013

ARMANDO VITOR DE OLIVEIRA SILVA

ALARME COM ATIVAÇÃO POR SENSOR PRESENCIAL E ALERTA VIA SMS
*****
=====*/

#include "SIM900.h"
#include "sms.h"
#include <SoftwareSerial.h>
SMMSGSM sms;

boolean started=false;

char telefones[][14] = { // Matriz para o número de dígitos do telefone
    "+556199719338", 0}; // o 0 é utilizado como condicional de stop

int ledVermelho = 13; // Pino para o LED vermelho
int PIRPin = 7;      // Pino OUT do sensor PIR
int PIR = 0;        // representa o valor fornecido pelo PIR

```

Figura 5.4 – Código Fonte: Diretrizes e Variáveis (Fonte: Autor)

Na figura 5.4 são definidas as diretrizes e variáveis que são usadas no código. Entre as diretrizes, estão as bibliotecas necessárias para o funcionamento do programa. Foi declarada a biblioteca para inicialização do modem GSM, e para a ativação de sua função de envio de SMS.

Entre as variáveis, está a matriz que guarda o número de telefone a ser enviado o SMS, a pinagem do LED e do sensor PIR, bem como sua variável de inicialização de leitura.

```

void sensor_presenca() {
  if(started){
    for (int i = 0; (int)*telefonos[i] != 0; i++) { // o 0 detecta o final
      // função de envio do SMS
      if (sms.SendSMS(telefonos[i], "Alerta! Perimetro Invadido!")) {
        Serial.print("\nSMS enviado para o numero: ");
        Serial.println(telefonos[i]);
        Serial.println("Perimetro Invadido!");
        Serial.println("SMS OK");
      }
    }
  }
};

```

Figura 5.5 – Código Fonte: Função Sensor de Presença (Fonte: Autor)

Na figura 5.5, é mostrada a função do sensor de presença. Esta função será chamada, quando o sensor PIR detectar movimento, e passar uma leitura alta para o microcontrolador. A função, após ser iniciada, faz a leitura dos números de telefone cadastrados. Posteriormente, ativa a diretriz sms, já inclusa na biblioteca, enviando a mensagem de alerta para o número cadastrado. Além disso, ela escreve no monitor serial a mensagem de alerta, escreve o número para qual a mensagem foi enviada, e conclui sinalizando positivamente que a mensagem foi enviada pelo modem.

```

void setup()
{
  pinMode(ledVermelho, OUTPUT);
  pinMode(PIRPin, INPUT);

  //função da Conexão Serial
  Serial.begin(9600);// bitrate - Taxa de dados 9600 bps
  Serial.println("Conectando Shield IComSat GSM/GPRS SIM900 v1.1");//teste de conexão com a rede
  //configuração do Shield com a baudrate.
  if (gsm.begin(2400)){
    Serial.println("\nSTATUS = SISTEMA CARREGADO!");
    started=true;
  }
  else Serial.println("\nSTATUS = SISTEMA COM PROBLEMA, FALHA NA REDE GSM!");
};

```

Figura 5.6 – Código Fonte: Função *setup* (Fonte: Autor)

A figura 5.6 mostra a função *setup* do código. Como já explicado anteriormente, a função *setup* é executada apenas no início do programa. Aqui é definida a caracterização de entrada ou saída das variáveis.

Também é observado, que a função traz a inicialização do modem GSM, bem como a taxa de dados a qual o modem está funcionando, e escreve no monitor serial o status de sucesso ou falha na conexão.

```
void loop()
{
  PIR = digitalRead(PIRPin);    // Faz leitura do OUT do sensor PIR e atribui o valor lido em PIR
  if (PIR == HIGH) {           // Testa se a variável PIR é alta
    digitalWrite(ledVermelho, HIGH); // acende led vermelho
    Serial.println("Perimento Invadido!");
    sensor_presenca();
    delay(10000);
  }
  else{
    digitalWrite(ledVermelho, LOW); // mantém o led vermelho apagado
  }
};
```

Figura 5.7 – Código Fonte: Função *loop* (Fonte: Autor)

Na figura 5.7 é exibida a função de laço do código fonte, ou seja, a função que se repetirá enquanto o sistema estiver em funcionamento. Nesta função é feita a leitura recebida pelo sensor, e atribuída seu valor à sua variável. Caso este valor seja alto, o LED é aceso, e no monitor serial, é escrita a mensagem de alerta. Após isso, é chamada a função do sensor de presença, para mandar a mensagem SMS ao número cadastrado. Caso o valor retornado pelo sensor seja baixo, o LED continua apagado, e nenhuma função é chamada.

No *loop* também é declarado um *delay*³ de dez segundos. Nesse tempo, o LED continua aceso, e não há leitura de valor retornado por parte do sensor. Isso impede leituras altas repetidamente em um curto espaço de tempo, e portanto, excesso de mensagens SMS mandadas.

5.3 Montagem do Protótipo

A montagem do protótipo foi feita em uma caixa de alumínio com a parte de cima aberta, para fácil manuseio do protótipo. Tendo em vista que o protótipo pode ser alimentado por uma fonte de 9V, ou por um cabo USB conectado a um computador, mostrou-se necessário que o protótipo tivesse uma folga dentro da cai-

³ *Delay*, do inglês atraso, é o tempo de espera da função para que possa ser ativada novamente.

xa, para fácil acoplagem dos meios de energia, sem o manuseio contínuo do protótipo.

A caixa utilizada tem 20cm x 12cm x 3cm, e foi utilizado um tubo de papelão envolvendo o sensor, para diminuir sua sensibilidade de abrangência.

Ainda, o microcontrolador Arduino UNO possui encaixe perfeito para o *shield* GSM iComSat SIM900, e para demais componentes foi utilizada uma placa *protoboard*.

CAPÍTULO 6 – RESULTADOS OBTIDOS

Neste capítulo é apresentado o desenvolvimento das simulações feitas com o projeto, suas funções e os problemas ocorridos durante o período de desenvolvimento para chegar ao objetivo proposto.

6.1 Simulações

Como explicado no item 5.2.1 do capítulo 5, a partir da figura 5.3, a primeira simulação feita, foi apenas com um acionamento do sensor PIR. Essa simulação serviu principalmente para a verificação se os componentes estavam funcionando. Além disso, a programação foi testada várias vezes, e o código fonte foi alterado em alguns momentos até conseguir o funcionamento correto do modem, e do acionamento dos LED no momento correto com o *delay*.

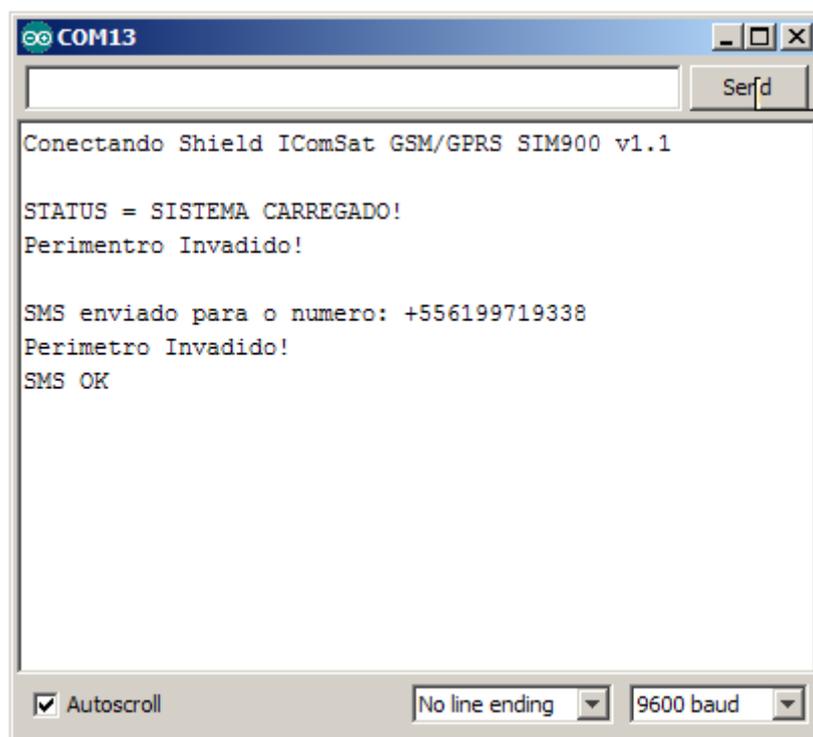


Figura 6.1 Monitor Serial: Simulação com o Protótipo Final (Fonte: Autor)

A figura 6.1 mostra a resposta do microcontrolador, através do monitor serial da interface Arduino IDE. Na figura, é mostrado o status de conexão do modem GSM, e a mensagem de alerta escrita, após o sensor ter detectado movimento. O

monitor serial, também mostra que a mensagem foi enviada corretamente para o número previamente cadastrado.

Como simulação real, o protótipo foi deixado em cima de um móvel da sala de estar, a 1,40m do chão, e a 2m a frente da porta de entrada da casa, para detectar a entrada e saída de pessoas na residência. Como o esperado, a medida que alguém entrava ou saía da casa, o sensor conseguia detectar o movimento de passagem pela porta, e foram recebidas algumas mensagens SMS – ilustrado na figura 6.2.



Figura 6.2 Entrega das mensagens SMS (Fonte: Autor)

6.2 Problemas Encontrados

No início dos testes do modem, foi obtido sempre a mesma mensagem de erro, e após visitar vários fóruns na internet, foi encontrado um caso com o mesmo erro do projeto, o qual relatava que o problema estava na antena do modem danificada. Da procura pela solução da mensagem de erro, até a obtenção da nova antena, transcorreu um tempo considerável, que levou a um atraso no cronograma de montagem e teste de hardware. Adicionalmente ao problema da antena, a maior dificuldade encontrada foi com uma biblioteca para o código, que atendesse aos requerimentos do projeto. Recorreu-se primeiramente a biblioteca do site oficial do Arduino, mas não foi possível chamar a função de envio de SMS. Então, foram testadas outras três bibliotecas, até chegar a solução encontrada.

Paralelamente, também foram enfrentados problemas com o cartão GSM. No momento em que o cartão fica sem créditos para o envio de SMS, o modem não tem função que avise sobre esse imprevisto. Então, levou-se um tempo até descobrir o motivo que as mensagens não estavam sendo enviadas.

6.3 Orçamento do Projeto

A aquisição dos componentes eletrônicos para o projeto foi realizada e foi fundamental para o projeto. Abaixo o quadro 6.1 detalha os componentes e o seus devidos custos.

Quadro 6.1 – Custo do Projeto

Item	Custo
Microcontrolador Arduino UNO	R\$ 75,00
Shield GSM iComSat SIM900	R\$ 140,00
Sensor PIR	R\$ 20,00
Buzzer 5v	R\$ 1,25
Led Vermelho	R\$ 0,25
Protoboard	R\$ 21,00
Antena p/ Shield SIM900	R\$ 50,00
Fonte de Alimentação	R\$ 16,00
TOTAL	R\$ 323,50

FONTE: Autor.

CAPÍTULO 7 – CONSIDERAÇÕES FINAIS

Neste capítulo são apresentadas as conclusões e as propostas para trabalhos futuros, que podem utilizar este projeto como base.

7.1 Conclusões

Foi desenvolvido, neste trabalho, um protótipo de alarme com o objetivo de apresentar uma solução para os problemas descritos no início deste documento, como fim de um usuário ser avisado sobre possível invasão de sua residência, mesmo este estando ausente.

Apesar dos problemas descritos e apresentados na sessão anterior, o objetivo foi alcançado, e o sensor de presença escolhido mostrou-se eficaz para a detecção de movimento necessária ao envio de mensagem SMS como alerta. Porém, esse sistema está sujeito a falhas, principalmente por parte da rede GSM, ou por parte da operadora de telefone. Não há um método que exponha se a linha telefônica do cartão está funcionando, enquanto eles está acoplado ao modem. Portanto, uma vez com o usuário ausente, o cartão GSM falhar, nada poderá ser feito para contornar esse problema.

Contudo, os resultados obtidos e apresentados por esse projeto, cumprem com o proposto e planejado desde o início, e o protótipo desde finalizado, encontra-se em funcionamento.

7.2 Propostas para Trabalhos Futuros

Com a crescente tecnologia na área de controle e automação residencial, tornam-se muito interessantes novos projetos nessa área. Uma boa modificação no projeto apresentado, seria adicionar outros tipos de sensores, como sensor de luz ou sensor de gás, onde cada novo sensor, quando ativado, enviaria uma mensagem SMS diferente ao usuário, aumentando o nível de segurança da residência.

Além disso, o modem GSM SIM900 possui várias outras funcionalidades inteligentes que poderiam ser utilizadas, como o acionamento do alarme via mensagem SMS do usuário, e até mesmo ligações provindas do Arduino, realocando os *jumpers* do módulo GSM.

REFERÊNCIAS BIBLIOGRÁFICAS

ARDUINO, Site do Projeto Arduino. **Arduino**. Disponível em: <arduino.cc/>. Acessado em 21 de mai 2013.

ARDUINO GSM SHIELD. **Open Eletronics, Open Source Eletronic Projects**. Disponível em: <http://www.open-eletronics.org/arduino-gsm-shield/>. Acessado em 7 de mai 2013.

BANZI, M. **Primeiros passos com o Arduino**. 1ª. Ed. Novatec, 2011.

CAMPOS, D. A. L. **Sistema de Alarme de Intrusão utilizando o SMS**. Brasília: Faculdade de Tecnologia e Ciências Sociais Aplicada, UniCEUB, 2010.

GIMENEZ, S. P. **Microcontroladores 8051**. 1ª ed. São Paulo: Pearson, 2002.

GSM World. **GSM**. 2012. Disponível em: < http://www.gsma.com/aboutus/gsm-technology/gsm/ > Acesso em: 1 de mai. 2013.

ICOMSAT SIM 900. **SIM 900 DataSheet**. Disponível em: < http://iteadstudio.com/store/images/produce/Shield/IComSat/icomsat_DS1.2.pdf > Acessado me: 26 abr. 2013.

INFRARED WAVES. **National Aeronautics and Space**. Disponível em: <http://science.hq.nasa.gov/kids/imagers/ems/infrared.html/>. Acessado em 11 de jul 2013

ME COMPONENTES. **Módulo GSM/GPRS, SIM900D**. Disponível em: <http://www.mecomp.com.br/sim900d.html> Acesso em 26 abr 2013.

MCROBERTS, M **Arduino Básico** 1ª Ed. Novatec, 2011.

PIR SENSOR. **BISS00001 DataSheet.** Disponível em: < <http://www.ladyada.net/media/sensors/BISS0001.pdf>>. Acessado em 10 de abr 2013.

THOMAZINI, D. ; ALBURQUERQUE, P. U. B. de. **Sensores Industriais:** Fundamentos e Aplicações. 4ª Ed. Érica, 2005.

What Is a SIM Card. Wise Geek. Disponível em: < <http://www.wisegeek.com/what-is-a-sim-card.htm> > Acessado em 3 de mai 2013.

WIKIPEDIA. A enciclopédia livre. **Wikipédia.** Disponível em: <<http://pt.wikipedia.org/wiki/gsm>>. Acessado em 2 de mai 2013.

APÊNDICE A – LINK DE VÍDEO DO AUTOR

Vídeo: “Alarme Arduino + Shield GSM”, disponível em:

<http://www.youtube.com/watch?v=xczthm9vc4>

APÊNDICE B – CÓDIGO FONTE TESTE DO SENSOR PIR

/******Teste Sensor PIR*****

Descrição:

Nome do arquivo: sketchPIR;

Autor: John Park – modificado pelo autor;

Microcontrolador utilizado: Arduino UNO;

Compilador: Arduino IDE v1.0.4;

Data de criação: Março 2013;

*****/

```
int ledPin = 13;           // pino para o LED
int inputPin = 7;         // pino output para o sensor PIR
int pirState = LOW;       // assume que não há detecção de movimento no início
int val = 0;              // variável de leitura do sensor PIR
int pinSpeaker = 10;      // pino para o buzzer
```

```
void setup() {
  pinMode(ledPin, OUTPUT); // declara o LED como output
  pinMode(inputPin, INPUT); // declara o sensor PIR como input
  pinMode(pinSpeaker, OUTPUT); // declara o buzzer como output
  Serial.begin(9600);      // declara a taxa de bits
}
```

```
void loop(){
  val = digitalRead(inputPin); // lê o valor de input do PIR
  if (val == HIGH) {          // verifica se o valor é alto, ou seja, se houve detecção
    digitalWrite(ledPin, HIGH); // acende o LED
    playTone(300, 160);       // ativa o buzzer
    delay(150);               // o delay baixo faz o ED apenas piscar e o buzzer tocar a
    sirene rapidamente
  }
```

```

    if (pirState == LOW) { /           // se o valor da leitura voltar a ser baixo
        Serial.println("Motion detected!"); // o monitor serial escreve que houve mo-
vimento
        pirState = HIGH;
    }
    } else {
        digitalWrite(ledPin, LOW); // apaga o LED
        playTone(0, 0); // sem buzzer
        delay(300);
        if (pirState == HIGH){           //se o valor da leitura voltar a ser alto
            Serial.println("Motion ended!");
            pirState = LOW; // aqui só quero escrever a mudança de valores, e não o
estado do PIR
        }
    }
}

void playTone(long duration, int freq) {
    duration *= 1000;
    int period = (1.0 / freq) * 1000000;
    long elapsed_time = 0;
    while (elapsed_time < duration) {
        digitalWrite(pinSpeaker, HIGH);
        delayMicroseconds(period / 2);
        digitalWrite(pinSpeaker, LOW);
        delayMicroseconds(period / 2);
        elapsed_time += (period);
    }
}
/* a função preset playTone é responsável por controlar a altura e duração da sirene
do buzzer*/

```

APÊNDICE C – CÓDIGO FONTE FINAL

```

/*=====
=====
*****
PROJETO FINAL - Engenharia da Computação - UniCEUB
1o. Semestre de 2013

ARMANDO VITOR DE OLIVEIRA SILVA

ALARME COM ATIVAÇÃO POR SENSOR PRESENCIAL E ALERTA VIA SMS
*****
=====
=====*/

#include "SIM900.h"                //bibliotecas do módulo GSM
#include "sms.h"
#include <SoftwareSerial.h>
SMSGSM sms;

boolean started=false;

char telefones[][14] = {          // Matriz para o número de dígitos do telefone
  "+556199719338", 0};           // o 0 é utilizado como condicional de stop

int ledVermelho = 13;           // Pino para o LED vermelho
int PIRPin = 7;                 // Pino OUT do sensor PIR
int PIR = 0;                    // representa o valor fornecido pelo PIR

/*****
*****FUNÇÃO ENVIO DE SMS*****/
void sensor_presenca() {
  if(started){
    for (int i = 0; (int)*telefones[i] != 0; i++) { // o 0 detecta o final
      if (sms.SendSMS(telefones[i], "Alerta! Perimetro Invadido!")) {
        Serial.print("\nSMS enviado para o numero: ");
        Serial.println(telefones[i]);
        Serial.println("Perimetro Invadido!");
        Serial.println("SMS OK");
      }
    }
  }
};
/*****
*****/

```

```

void setup()
{
  pinMode(ledVermelho, OUTPUT);
  pinMode(PIRPin, INPUT);

  //função da Conexão Serial
  Serial.begin(9600);           // bitrate - Taxa de dados 9600 bps
  Serial.println("Conectando Shield IComSat GSM/GPRS SIM900 v1.1"); //teste de
  conexão com a rede
  //configuração do Shield com a baudrate.
  if (gsm.begin(2400)){
    Serial.println("\nSTATUS = SISTEMA CARREGADO!");
    started=true;
  }
  else Serial.println("\nSTATUS = SISTEMA COM PROBLEMA, FALHA NA REDE
  GSM!");

};

void loop()
{
  PIR = digitalRead(PIRPin); // Faz leitura do OUT do sensor PIR e atribui o valor
  lido em PIR
  if (PIR == HIGH) {          // Testa se a variável PIR é alta
    digitalWrite(ledVermelho, HIGH); // acende led vermelho
    Serial.println("Perimetro Invadido!");
    sensor_presenca();
    delay(10000);
  }
  else{
    digitalWrite(ledVermelho, LOW); // mantem o led vermelho apagado
  }
};

```

APÊNDICE D – ESQUEMÁTICO ARDUINO UNO

Arduino™ UNO Reference Design

Reference Designs ARE PROVIDED 'AS IS' AND 'WITH ALL FAULTS'. Arduino DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Arduino may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Arduino reserves the right to change specifications and instructions at any time without notice, and is not liable for errors or for any consequences arising from the use of the information on the product instructions or the Web Site of Arduino in respect to design and the information.

