



CENTRO UNIVERSITÁRIO DE BRASÍLIA -UniCEUB
CURSO DE ENGENHARIA DE COMPUTAÇÃO

GUSTAVO MOURA FÉ MAIA

ACIONAMENTO REMOTO DE PORTÕES ELÉTRICOS VIA CELULAR
ATRAVÉS DE MICROCONTROLADOR

Orientadora: Profa. M. C. Maria Marony Sousa Farias

Brasília
Junho, 2012

GUSTAVO MOURA FÉ MAIA

**ACIONAMENTO REMOTO DE PORTÕES ELÉTRICOS VIA CELULAR
ATRAVÉS DE MICROCONTROLADOR**

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: Profa. M. C.

Maria Marony Sousa

Farias

Brasília
Junho, 2012

GUSTAVO MOURA FÉ MAIA

**ACIONAMENTO REMOTO DE PORTÕES ELÉTRICOS VIA CELULAR
ATRAVÉS DE MICROCONTROLADOR**

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: Profa. 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 Amarilia Fernandes
Coordenador do Curso

Banca Examinadora:

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

Prof. Luciano Henrique Duque,
mestre em Engenharia Elétrica.

Prof^a. Vera Lúcia Farini Alves
Duarte, mestre em Matemática.

AGRADECIMENTOS

À minha família, por todo o carinho e dedicação, sempre buscando o melhor para minha formação pessoal e acadêmica. Especialmente ao meu pai, sempre presente e solícito durante todo o desenvolvimento deste projeto.

Aos meus amigos e colegas que sempre me apoiaram nos momentos em que precisei.

À minha orientadora que forneceu todo o suporte e incentivo necessário para a elaboração e concretização do projeto final.

E à todos que colaboraram de alguma forma com o desenvolvimento e a realização deste trabalho.

RESUMO

Este projeto apresenta um dispositivo de acionamento remoto de portões elétricos via celular utilizando microcontrolador. O projeto integra a placa Arduino UNO, cujo microcontrolador é ATmega328, uma placa com módulo GSM SIM900 e uma placa com módulos relés num único protótipo. O protótipo realiza cadastro de usuários, adicionando novos ou removendo cadastros já existentes, através de mensagem de texto enviada por um administrador do sistema – o primeiro usuário a mandar um SMS ao protótipo. O protótipo, ao receber uma chamada, faz a identificação da ligação e a compara aos telefones cadastrados em seu SIM Card; e envia um pulso elétrico ao módulo relé acionando o portão elétrico caso o usuário seja cadastrado. A linguagem de programação utilizada para o desenvolvimento foi a Arduino que implementa as linguagens *Processing* e *C*.

Palavras-chave: Automação, Acionamento Remoto, GSM, Arduino, ATmega328, *Processing*, *C*, SMS, SIM Card.

ABSTRACT

This project presents a device to remotely actuate electric gates through a microcontroller using a mobile phone. The project integrates the Arduino board, UNO, which uses the microcontroller ATmega328, a board with GSM module SIM900 and a relay board within a single prototype. The prototype registers users, adding new or removing existing entries, by text message sent by a system administrator - the first user to send an SMS to the prototype. When the prototype receives an incoming call, it makes the identification of the incoming call and compares it with the phones registered on its SIM Card, and sends an electrical pulse to the relay module powering the electric gate if the user is registered. The programming language used for the development was the Arduino language that implements the Processing language and C language.

Keywords: Automation, Remote Actuation, GSM, Arduino, ATmega328, Processing, C, SMS, SIM Card.

SUMÁRIO

LISTA DE FIGURAS.....	IX
LISTA DE TABELAS.....	XI
LISTA DE ABREVIATURAS E SIGLAS.....	XII
CAPÍTULO 1 – INTRODUÇÃO.....	13
1.1 Apresentação do problema.....	13
1.2 Objetivos do Trabalho.....	13
1.3 Metodologias.....	14
1.4 Estrutura da Monografia.....	15
CAPÍTULO 2 – APRESENTAÇÃO DO PROBLEMA.....	16
2.1 Segurança Residencial.....	16
2.2 Comodidade.....	16
2.3 Soluções Existentes.....	17
2.4 Benefícios do dispositivo proposto e suas restrições.....	18
CAPÍTULO 3 – REFERENCIAL TEÓRICO.....	19
3.1 Microcontroladores.....	19
3.1.1 Arduino.....	19
3.2 Rede GSM.....	20
3.3 SMS.....	21
3.4 SIM Card.....	22
CAPÍTULO 4 – DESCRIÇÃO DE HARDWARE E SOFTWARE....	23
4.1 Arduino UNO.....	23
4.1.1 Especificações.....	23
4.1.2 Pinagem utilizada.....	24
4.2 Arduino IDE.....	25
4.3 Placa com Módulo GSM SIM900.....	27
4.4 Comandos AT.....	28
4.5 Placa com Módulos Relés.....	30
CAPÍTULO 5 – IMPLEMENTAÇÃO.....	33
5.1 Modelagem do sistema.....	33
5.2 Elaboração do código fonte.....	35

5.2.1	Escrita do código fonte.....	36
5.3	Montagem do protótipo.....	41
CAPÍTULO 6 – RESULTADOS OBTIDOS.....		45
6.1	Simulações.....	45
6.1.1	Cadastrar o primeiro usuário que enviar mensagem de texto como administrador do sistema.....	47
6.1.2	Adicionar usuários através de SMS do administrador.....	50
6.1.3	Remover usuários através de SMS do administrador.....	51
6.1.4	Acionar remotamente o portão elétrico quando usuário cadastrado efetuar uma chamada ao módulo GSM.....	53
6.2	Problemas encontrados.....	55
CAPÍTULO 7 – CONSIDERAÇÕES FINAIS.....		57
7.1	Conclusões.....	57
7.2	Propostas para trabalhos futuros.....	57
REFERÊNCIAS BIBLIOGRÁFICAS.....		59
APÊNDICE A – LINKS DE VIDEOS DO AUTOR.....		61
APÊNDICE B – CÓDIGO FONTE DE TESTE.....		62
APÊNDICE C – CÓDIGO FONTE FINAL.....		68
ANEXO A – ESQUEMÁTICO: ARDUINO UNO.....		72

LISTA DE FIGURAS

Figura 2.1 – Dispositivo (2), TiDiGino.....	17
Figura 2.2 – Ilustração do funcionamento do dispositivo (1).....	18
Figura 3.1 – Placa Arduino UNO.....	20
Figura 3.2 – Logo GSM desde 2000.....	20
Figura 3.3 – SIM Card usado no projeto.....	22
Figura 4.1 – Modelo Arduino UNO.....	23
Figura 4.2 – Interface do Arduino IDE versão 0022 (AUTOR).....	26
Figura 4.3 – Circuito esquemático da placa com modulo GSM SIM900.....	27
Figura 4.4 – Módulo GSM SIM900.....	28
Figura 4.5 – Placa do portão (à esquerda) e botoeira em destaque (à direita).....	31
Figura 4.6 – Tensão em volts (à esquerda) e a corrente em μA (à direita) do portão elétrico utilizado no projeto.....	31
Figura 4.7 – Placa com 8 módulos relés.....	32
Figura 4.8 – Circuito do modulo relé.....	32
Figura 5.1 – Fluxograma geral.....	34
Figura 5.2 – Fluxograma ‘Trata SMS’.....	34
Figura 5.3 – Fluxograma ‘Trata Chamada’.....	35
Figura 5.4 – Código fonte: Diretivas e Variáveis.....	36
Figura 5.5 – Código fonte: Função setup.....	37
Figura 5.6 – Código fonte: Função loop.....	39
Figura 5.7 – Código fonte: Função trataChamada.....	40
Figura 5.8 – Protótipo fase 1.....	41
Figura 5.9 – Protótipo fase 2.....	42
Figura 5.10 – Protótipo fase 3.....	43
Figura 5.11 – Protótipo fase final.....	44
Figura 6.1 – Placa Aruino UNO em funcionamento.....	46
Figura 6.2 – Sistema ativado.....	46
Figura 6.3 – Tela do monitor serial: Iniciando sistema.....	47
Figura 6.4 – Tela do monitor serial: Número não autorizado.....	48

Figura 6.5 – Tela do monitor serial: Definição do administrador e cadastro do primeiro usuário.....	48
Figura 6.6 – Tela do monitor serial: Confirmação do cadastro do administrador e número enviado.....	49
Figura 6.7 – Tela do monitor serial: Cadastro não autorizado e confirmação dos números do SIM Card.....	50
Figura 6.8 – Tela do monitor serial: Cadastro autorizado e confirmação dos números no SIM Card.....	51
Figura 6.9 – Tela do monitor serial: Remoção não autorizada e confirmação dos números no SIM Card.....	52
Figura 6.10 – Tela do monitor serial: Remoção autorizada e confirmação dos números do SIM Card.....	53
Figura 6.11 – Tela do monitor serial: Acionamento do portão elétrico autorizado.....	54
Figura 6.12 – Tela do monitor serial: Acionamento do portão elétrico não autorizado.....	55

LISTA DE TABELAS

Tabela 4.1 – Principais características da placa Arduino UNO.....	24
Tabela 4.2 – Pinagem da placa Arduino UNO utilizada neste projeto.....	25
Tabela 4.3 – Descrição da interface do Arduino IDE.....	26
Tabela 4.4 – Sintaxe dos comandos AT.....	29
Tabela 4.5 – Comandos AT utilizados.....	30
Tabela 5.1 – Comandos da função setup.....	38

LISTA DE SIGLAS E ABREVIATURAS

AT	Hayes AT Commands
cm	Centímetros
EEPROM	Erasable Electronically Programmable Read Only Memory
FTDI	Future Technology Devices International
GSM	Global System for Mobile Communications
IDE	Integrated Development Enviroment
KB	Kilo Bytes
μA	Micro Ampere
MHz	Mega Hertz
PWM	Pulse Width Modulation
SRAM	Static Random Access Memory
USB	Universal Serial Bus
V	Volts

CAPÍTULO 1 – INTRODUÇÃO

1.1 – Apresentação do Problema

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

Dentre as diversas tecnologias que têm sido trazidas às residências, portões elétricos acionados por controle remoto é uma destas tecnologias mais simples e presentes em casas e condomínios. Essa tecnologia permite um acesso mais ágil, seja à residência, ao condomínio, etc., tornando esse acesso mais seguro, evitando que o usuário tenha que sair do veículo para abrir o portão. No entanto, o controle remoto responsável pelo acionamento do portão elétrico não faz distinção de quem o aciona, tornando possível o acesso de qualquer indivíduo que tenha este controle em mãos.

Outro problema gerado pelo acionamento remoto é o fato do usuário ter que carregar consigo até mais de um controle, como no caso de condomínios em que é necessário um controle para o acesso ao condomínio e outro para a própria residência. Como solucionar ou evitar estes problemas?

1.2 – Objetivos do Trabalho

Este projeto tem como objetivo geral apresentar uma solução alternativa para os problemas apresentados na seção anterior, com a criação de um dispositivo de acionamento remoto de portões elétricos via celular através de microcontrolador que substitua o atual controle remoto.

Como objetivo específico, neste projeto é desenvolvido um protótipo que tem como funções cadastrar e remover usuários de seu SIM Card, através do envio de mensagens de texto, e enviar um pulso elétrico para o acionamento remoto de um portão elétrico quando este dispositivo receber uma ligação de um usuário cadastrado. Utilizando a placa microcontrolada, Arduino UNO,

juntamente com o módulo GSM SIM900, ao receber a primeira mensagem de texto o protótipo cadastra o cliente que a enviou como administrador do sistema – o único capaz de cadastrar e remover usuários. Após a definição do administrador, é possível realizar o cadastro de novos usuários através do envio de um SMS com o número de telefone do cliente. Caso seja necessária a remoção de um usuário, basta que o administrador envie outro SMS com o número deste cliente para que o sistema o apague da memória. Assim, ao receber uma ligação, o protótipo verifica o cadastro e aciona o portão elétrico para os usuários registrados.

1.3 – Metodologias

Para o desenvolvimento deste projeto foram realizadas pesquisas bibliográficas como ponto de partida para o desenvolvimento da ideia do protótipo. Diversas pesquisas foram feitas 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 utilizados no projeto. Todos os componentes eletrônicos foram estudados através de manuais e sites oficiais para o desenvolvimento do protótipo. Estes componentes foram testados através de medições com multímetro, bem como suas ligações.

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. Outros dois fluxogramas foram desenvolvidos, um para o tratamento de SMS e outro para tratamento de chamadas, para auxiliar tanto na compreensão quanto na elaboração de estas duas funções. Após a escrita do código fonte, este foi compilado e carregado no microcontrolador para que então fosse realizada a montagem do protótipo. O dispositivo teve como base um quadrado

de madeira onde foram fixados os componentes eletrônicos e, depois, colocados dentro de uma caixa de papelão adaptada.

Os testes realizados foram executados através de celulares comuns com chips de três operadoras diferentes. Estes testes têm como objetivo avaliar o desempenho do sistema nas seguintes tarefas: definição do administrador, cadastro de novos usuários, remoção de usuários e acionamento remoto do portão elétrico.

1.4 – Estrutura da Monografia

Este documento é dividido em sete capítulos. O primeiro, INTRODUÇÃO, trata da apresentação do problema, dos objetivos deste trabalho, as metodologias utilizadas para seu desenvolvimento e a estrutura da monografia.

O segundo capítulo, APRESENTAÇÃO DO PROBLEMA, trata em detalhes as motivações que levaram ao desenvolvimento deste projeto, apresentando também soluções existentes ao problema e os benefícios do dispositivo proposto e suas restrições.

O terceiro capítulo, REFERENCIAL TEÓRICO, aborda o embasamento teórico necessário para melhor compreensão e desenvolvimento do projeto, como introduções a microcontroladores, rede GSM e SIM Card.

O quarto capítulo, DESCRIÇÃO DE HARDWARE E SOFTWARE, detalha tecnicamente os componentes físicos e lógicos com ênfase no que será utilizado.

No quinto capítulo, IMPLEMENTAÇÃO, são apresentados os passos necessários para o desenvolvimento e compreensão do projeto.

O sexto capítulo, RESULTADOS OBTIDOS, contém simulações que têm como finalidade observar o desempenho de todas as funcionalidades do projeto num ambiente real, assim como os problemas encontrados.

O sétimo capítulo, CONSIDERAÇÕES FINAIS, contém as conclusões do projeto e sugestões para trabalhos futuros.

CAPÍTULO 2 – APRESENTAÇÃO DO PROBLEMA

Este capítulo visa detalhar as motivações para o desenvolvimento deste projeto, previamente descritas na seção – Apresentação do Problema – do capítulo anterior. Algumas soluções semelhantes ao proposto por este documento são apresentadas também e, no final deste capítulo, são apresentados os benefícios do protótipo obtido e suas restrições.

2.1 – Segurança Residencial

“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)

Não é de hoje que se observa o aumento dos furtos à residência. Por todos os veículos de comunicação tem-se relatos e números crescentes de assaltos todos os dias. Segundo o site de notícias, G1 (2011), “o número de furtos em casas do Lago Norte, bairro nobre de Brasília, quase dobrou no mês 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 formas de melhorar a segurança do lar. A automação pode ser um meio de prover este aumento de segurança, como através de câmeras de vigilância, sensores de proximidade e alarmes residenciais. O meio proposto por este projeto é o controle do acesso à residência através do acionamento do portão elétrico.

2.2 – Comodidade

O avanço tecnológico contínuo é observado nas mais diversas áreas, como na medicina, construção civil, engenharia da computação e, com isso, tem adentrado hospitais, escritórios e residências.

A tecnologia surge também como uma fonte geradora de conforto. Desde um simples controle remoto da televisão até geladeiras que informam quando os alimentos estão acabando são possibilidades.

O protótipo resultante deste projeto tem como objetivo trazer mais comodidade ao usuário, tendo o controle de acionamento remoto do portão elétrico ao alcance de uma ligação.

2.3 – Soluções Existentes

Depois de pesquisar dispositivos similares que tivessem o mesmo propósito do protótipo deste projeto, encontrou-se alguns próximos que são brevemente descritos.



Figura 2.1 – Dispositivo (2), TiDiGino. Fonte: TIDIGINO

Os dispositivos são *GSM Remote Control – 2 IN and 2 OUT*(1) e *TiDiGino, the Arduino-based GSM Remote Control*(2). Ambos possuem um *design* e funcionalidade muito próximos. No entanto, o dispositivo (1) possui o PIC18F46K20-I/PT como microcontrolador, enquanto o outro dispositivo (2) utiliza o microcontrolador ATmega 2560.

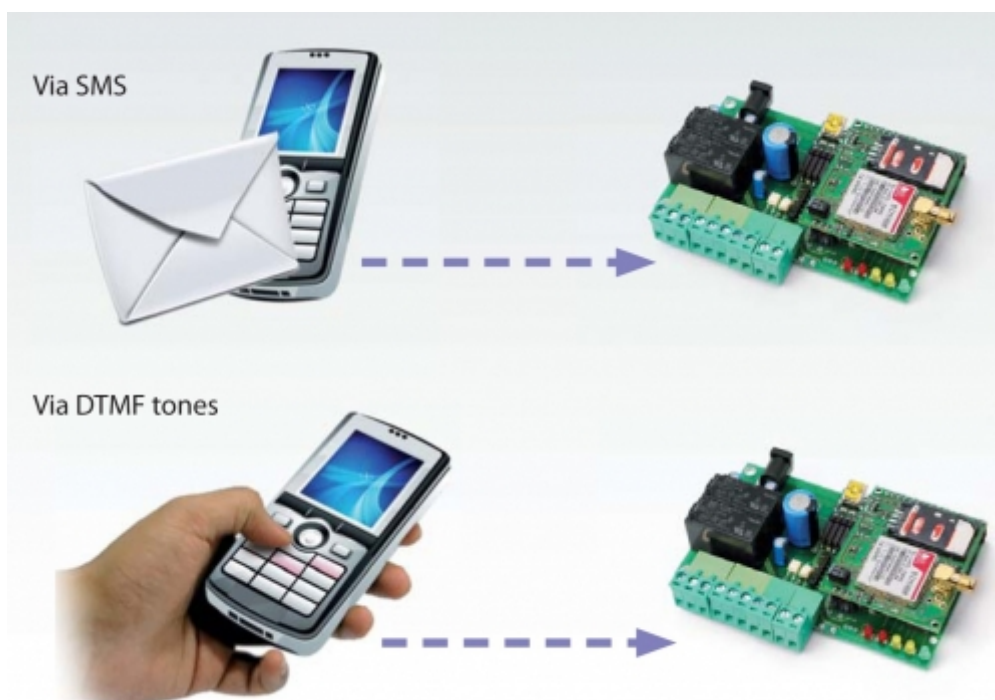


Figura 2.2 – Ilustração do funcionamento do dispositivo (1). Fonte: GSM REMOTE CONTROL

Ambos os dispositivos (1) e (2) citados funcionam como um controle remoto através do envio de mensagens de texto ou de ligações telefônicas seguidas de códigos numéricos pré-estabelecidos pelos fabricantes.

2.4 – Benefícios do Dispositivo Proposto e suas Restrições

O protótipo obtido com o desenvolvimento deste projeto tem como finalidade promover o acionamento remoto de um portão elétrico através de uma ligação. Os elementos eletrônicos são de fácil acesso e o desenvolvimento do protótipo simplificado.

Este dispositivo também deverá identificar o número da ligação recebida, verificar se este está no cadastro para, em caso afirmativo, acionar o portão elétrico.

O protótipo não tem como funcionalidade a leitura do estado de abertura do portão elétrico (aberto, fechado, abrindo ou fechando), ou seja, funciona de modo similar a um controle remoto. A leitura do estado do portão elétrico é feita pelo usuário, decidindo acionar remotamente seja para abertura como para o fechamento do portão.

CAPÍTULO 3 – REFERENCIAL TEÓRICO

Para compreensão e desenvolvimento deste trabalho são necessários alguns conceitos que se relacionam diretamente ao projeto. Esses conceitos, apesar de muito amplos, serão citados com o foco neste projeto.

3.1 – Microcontroladores

Segundo Gimenez (2005), “o microcontrolador é um dispositivo semicondutor 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”.

Microcontroladores são usados com frequência em tarefas específicas que não exijam grandes quantidades de dados, como em automação residencial – mais especificamente, placas de portões elétricos –, pois geralmente possuem pouca memória de armazenamento.

3.1.1 – Arduino

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

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



Figura 3.1 – Placa Arduino UNO. Fonte: ARDUINO

O arduino foi escolhido por se tratar de um ambiente multiplataforma, ou seja, pode ser executado no Linux, Mac OS e Windows. Além dessa vantagem, o arduino tem como base o IDE de programação *Processing*, um ambiente de desenvolvimento amigável e de fácil utilização. Finalmente, o arduino UNO pode ser programado através do cabo USB, sem a necessidade de se utilizar uma porta serial que vem sendo cada vez menos presente nos computadores modernos.

3.2 – Rede GSM

Originalmente sigla para *Groupe Special Mobile* (Grupo Especial de telefonia móvel, tradução livre), GSM agora é a abreviação de *Global System for Mobile Communications* ou Sistema Global para Comunicações Móveis.



Figura 3.2 – Logo GSM desde 2000. Fonte: WIKIPEDIA

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 do mesmo número de celular em até 219 países. As redes GSM terrestres cobrem mais de 90% da população mundial e a tecnologia *roaming*¹ GSM de satélites estende essa cobertura para áreas que a rede terrestre não alcança. Hoje, as conexões móveis globais superam 6 bilhões (GSM,2012).

3.3 – SMS

De acordo com Hillebrand (2010), o conceito do Serviço de Mensagens Curtas (do inglês, *Short Message Service*) foi desenvolvido em 1984 numa cooperação franco-germânica. A idéia seria aproveitar o sistema telefônico da época, que mandava mensagens de sinalização através de canais telefônicos para controle de tráfego durante certos períodos do dia, para enviar mensagens de texto a um baixo custo quando este sistema estivesse ocioso. No entanto, essas mensagens seriam limitadas a 128 *bytes* (futuramente aumentadas para 140 *bytes* ou 160 dígitos de 7 *bits*), o tamanho padrão das mensagens de sinalização utilizadas.

Atualmente, as mensagens de texto estão presentes no cotidiano dos usuários de telefones celulares como uma forma barata e rápida de se comunicar. Segundo o site de estudos estatísticos, Statista² (2012), durante o ano de 2010 mais de 192 mil mensagens de textos foram enviadas a cada segundo em todo o globo.

¹ *Roaming*, também chamado de itinerância, é um termo empregado na telefonia móvel que designa a capacidade de um usuário de uma rede obter conectividade em áreas fora da localidade geográfica onde está registrado.

² Statista é um site de pesquisas e estudos estatísticos parceiro de DOWJONES entre outras. Site da empresa: <http://www.statista.com/>

3.4 – SIM Card

O *Subscriber Identification Module* (módulo de identificação do assinante), ou SIM Card – também chamado de Cartão SIM – é um circuito integrado do tipo chip que é utilizado para identificar um usuário na rede GSM e também para controlar e armazenar dados de telefones celulares.

Na Figura 3.3 é ilustrado o SIM Card utilizado neste projeto. Este possui 128 KB de memória e dimensões padrões de 25 milímetros comprimento e 15 milímetros de largura.

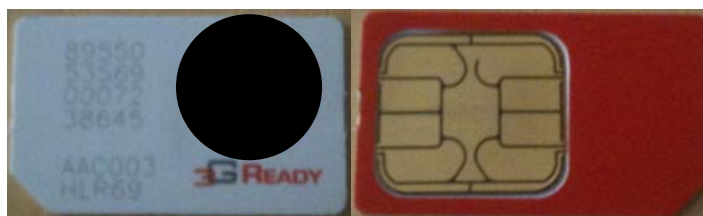


Figura 3.3 – SIM Card usado no projeto. Fonte: AUTOR

O SIM Card possui uma memória do tipo EEPROM, uma memória não volátil, que armazena, não só os números de telefones e o código de identificação do chip, mas as configurações e dados das outras funcionalidades. Algumas dessas funcionalidades são:

- SIM *Tool Kit* (STK)
- Autenticação do usuário
- Serviços OTA (*Over The Air*)

CAPÍTULO 4 – DESCRIÇÃO DE HARDWARE E SOFTWARE

4.1 – Arduino UNO

A placa Arduino UNO é a mais recente dentre as séries de placas Arduino USB. Essa placa difere das outras – 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, produzido pela Atmel. A figura 4.1 ilustra o modelo da placa utilizada neste projeto.

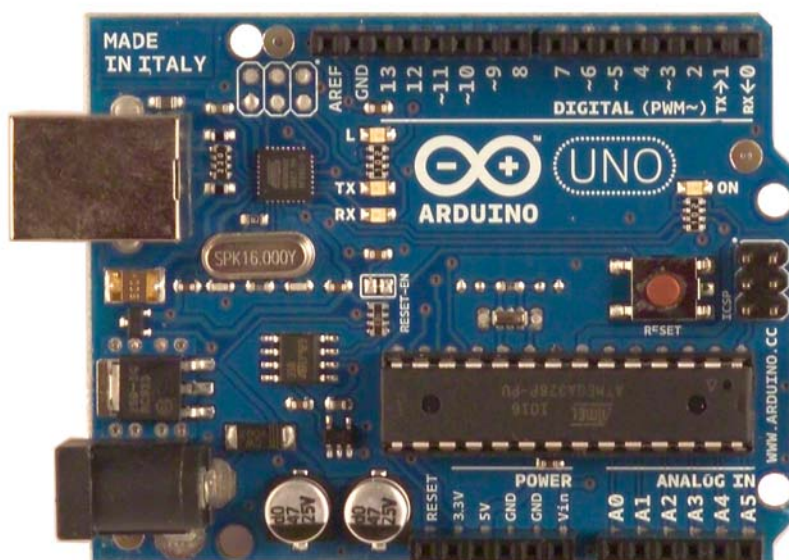


Figura 4.1 – Modelo Arduino UNO. Fonte: ARDUINO UNO

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

As principais características desta placa são apresentadas na tabela 4.1 abaixo:

Tabela 4.1 – Principais características da placa Arduino UNO

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

Fonte: AUTOR

Esta placa possui 3 pinos GND (do inglês *Ground*, ou Terra), dois pinos que fornecem uma voltagem regulada de 3,3 V e 5 V cada, um pino Vin que fornece a voltagem advinda diretamente da alimentação – seja por USB ou por fonte externa – e um pino de entrada para referência analógica (AREF).

4.1.2 – Pinagem Utilizada

A tabela 4.2 detalha a pinagem utilizada descrevendo o pino e sua funcionalidade específica neste projeto.

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

Tabela 4.2 – Pinagem da placa Arduino UNO utilizada neste projeto

Pino	Utilização
Vin	Alimentação do Módulo GSM
5 V	Alimentação da placa de Relés
GND	Aterramento da placa de Relés
4	Porta RX Virtual para controle do Módulo GSM
5	Porta TX Virtual para controle do Módulo GSM
2	Acionamento de um dos relés da placa de Relés

Fonte: AUTOR

4.2 – Arduino IDE

O Arduino IDE é um programa que permite a criação de novos programas – chamados de *sketches*¹ – para a placa Arduino numa linguagem simples, modelada a partir da linguagem *Processing*. Esse ambiente de desenvolvimento é responsável pelo envio do código fonte para o compilador *avr-gcc*² que converte o código para a linguagem de programação C para finalmente traduzir o resultado em linguagem de máquina, compreendida pelo microcontrolador. A figura 4.2 traz a interface do Arduino IDE numerada pelo autor para melhor descrição na tabela 4.3.

¹ Do inglês, esboços ou rascunhos.

² *Software* de código aberto com compilador C (encontrado na página web: winavr.sourceforge.net/)

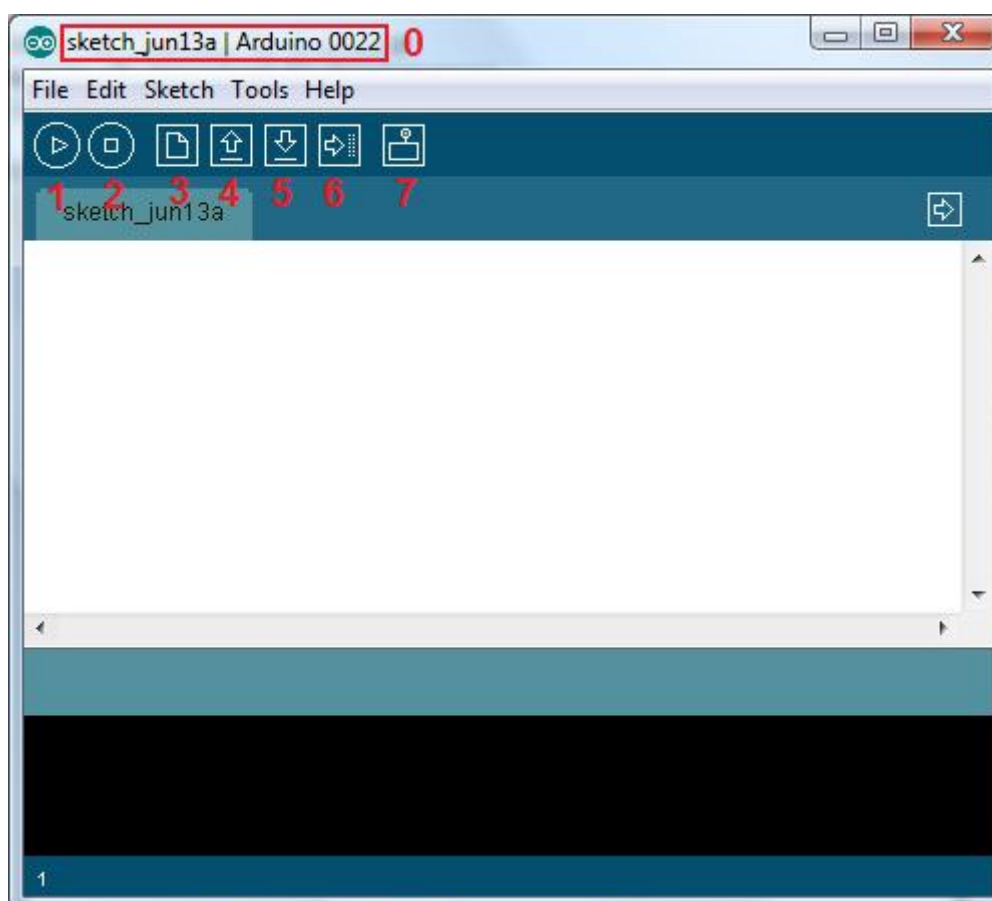


Figura 4.2 – Interface do Arduino IDE versão 0022. Fonte: AUTOR

Tabela 4.3 – Descrição da interface do Arduino IDE

Número	Nome	Descrição
0	Título Versão	Item composto do nome da <i>sketch</i> e da versão do Arduino IDE
1	<i>Verify</i>	Compila o código fonte
2	<i>Stop</i>	Para a execução ou compilação do programa
3	<i>New</i>	Gera uma nova <i>sketch</i>
4	<i>Open</i>	Abre <i>sketches</i> salvas
5	<i>Save</i>	Salva a <i>sketch</i> atual
6	<i>Upload</i>	Efetua a compilação do código e transmite para a placa
7	<i>Serial Monitor</i>	Abre um monitor serial que simula a comunicação do computador com a placa

Fonte: AUTOR

4.3 – Placa com Módulo GSM SIM900

A placa com módulo GSM SIM900, também chamada de *shield*¹ GSM, é uma placa que pode ser conectada facilmente sobre a placa Arduino UNO estendendo as funcionalidades do módulo GSM ao conjunto resultante. Como esta placa é empilhada sobre a placa Arduino UNO, através de todos seus pinos, a placa com módulo GSM SIM900 utiliza esta estrutura para sua alimentação e funcionamento. A figura 4.3 ilustra o circuito desta placa.

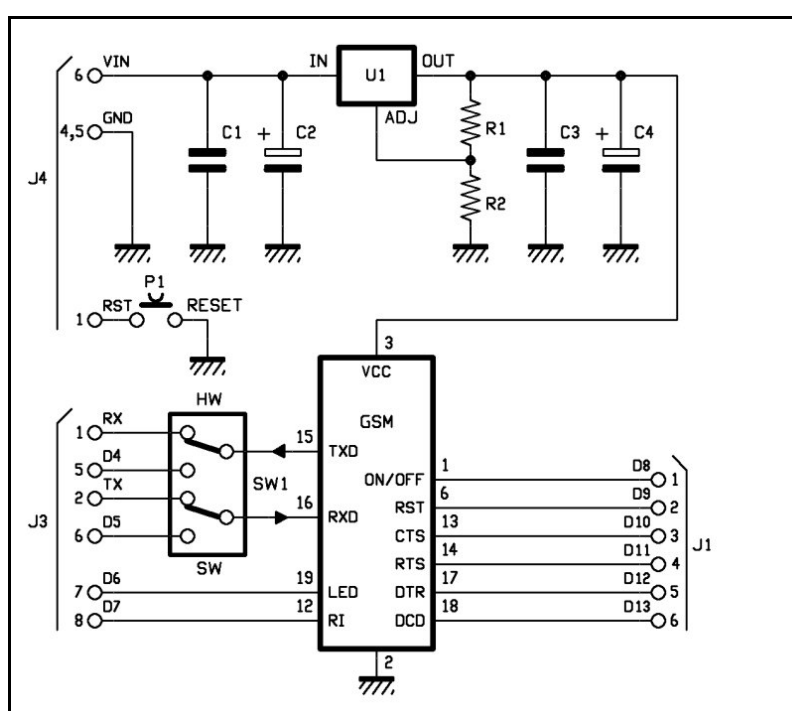


Figura 4.3 – Circuito esquemático da placa com módulo GSM SIM900.
Fonte: ARDUINO GSM SHIELD

O módulo GSM SIM900, produzido pela SIM Com, é o responsável pela comunicação com a rede GSM tornando possível a efetuação e o recebimento de ligações telefônicas e mensagens de texto. O modelo utilizado neste projeto é ilustrado na figura 4.4.

¹ *Shield* do inglês, escudo. Utilizada neste contexto como placa empilhável em formato compatível com a placa Arduino UNO.



Figura 4.4 – Módulo GSM SIM900.
Fonte: AUTOR

O funcionamento deste módulo é controlado através do programa carregado no microcontrolador por meio de comandos AT que serão discutidos a seguir.

4.4 – Comandos AT

Os comandos AT são linhas de código com solicitações de serviços reconhecidas pelo módulo GSM SIM900. Esses comandos são fornecidos pelo fabricante e seguem um padrão de sintaxe descrito brevemente a seguir na tabela 4.4.

Tabela 4.4 – 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 por usuário
Comando de Execução	AT+<comando>	Executa parâmetros não variáveis afetados por processos internos do módulo

Fonte: SIMCom, SIM900_AT Command Manual_V1.03, p.13

Apesar do módulo GSM SIM900 reconhecer uma vasta lista de comandos AT, apenas alguns desses foram utilizados direta ou indiretamente no desenvolvimento deste projeto. A tabela 4.5 traz esses comandos e suas respectivas descrições.

Tabela 4.5 – Comandos AT utilizados

Comando	Descrição
AT	Verifica comunicação com o módulo
AT+IPR=9600	Define a taxa de comunicação do módulo (bauds)
AT&F	Define as configurações de fábrica do módulo
AT+CLIP=1	Ativa a identificação de chamada
AT+CMEE=0	Desabilita mensagem com código de erro e, ao invés, utiliza mensagem ERROR
AT+CMGF=1	Define modo texto para mensagens de texto
AT+CPBS="SM"	Define o tipo de armazenamento da agenda para armazenamento em SIMCard
ATE0	Desabilita o eco das informações recebidas no módulo
AT+CPBR=i	Retorna o registro na posição i do SIM Card
AT+CPBW=i,"06199998888"	Escreve o número de telefone 06199998888 na posição i do SIM Card
AT+CPAS	Retorna o status da atividade do módulo
AT+CLCC	Retorna a lista atual de chamadas
AT+CMGL="ALL"	Lista todas mensagens de texto presentes
AT+CMGR=i	Retorna o número de telefone que enviou a mensagem de texto e o conteúdo do SMS
AT+CMGD=1,4	Apaga todas mensagens de texto do local de armazenamento atual
ATH	Desliga quaisquer chamadas e/ou ligações

Fonte: SIMCom, SIM900_AT Command Manual_V1.03

4.5 – Placa com Módulos Relés

Relés são dispositivos eletromecânicos capazes de controlar circuitos de grandes tensões e correntes (1) a partir de pequenas tensões e correntes de outro circuito (2). O funcionamento de um relé se dá pela seguinte forma: quando o circuito de controle (2) – com pequenas tensões e correntes – envia uma corrente para a bobina do relé, este gera um campo magnético que atrai o contato, fechando o circuito de carga (1).

Para a definição dos relés a serem utilizados no protótipo foi necessária a análise do circuito de carga do projeto, neste caso, o portão elétrico. A placa do portão é ilustrada na figura 4.5, com um destaque para a botoeira.



Figura 4.5 – Placa do portão (à esquerda) e botoeira em destaque (à direita). Fonte: AUTOR

Em seguida, foram medidas a tensão e a corrente da botoeira com um multímetro como ilustra a figura 4.6. A tensão obtida foi de 4,78 V e a corrente de 295 μ A.



Figura 4.6 – Tensão em volts (à esquerda) e a corrente em μ A (à direita) do portão elétrico utilizado no projeto. Fonte: AUTOR

Como o protótipo visa abranger um grande número de portões elétricos – e suas botoeiras – o relé escolhido foi o comum SRD-05VDC-SL-C da SONGLE que requer a alimentação de 5 V que a placa Arduino UNO pode fornecer.

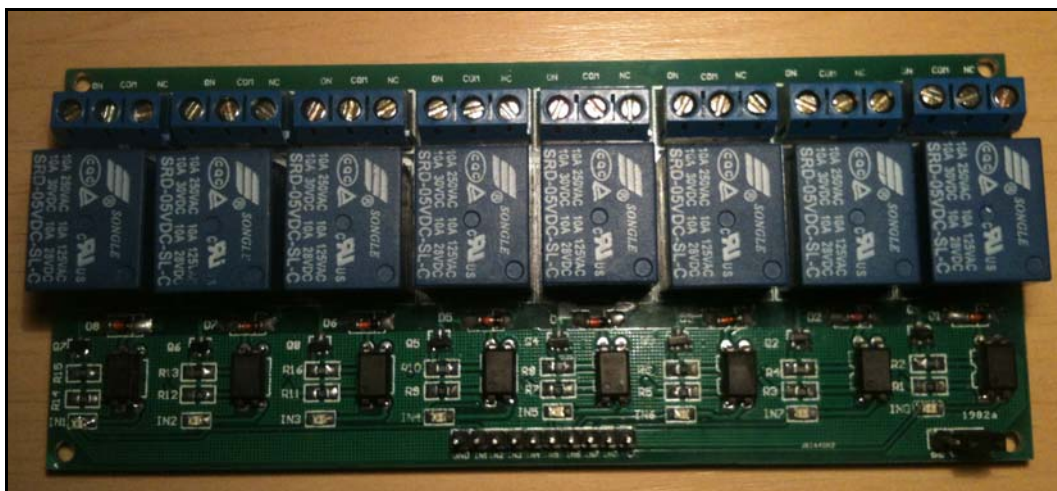


Figura 4.7 – Placa com 8 módulos relés. Fonte: AUTOR

A placa com módulos relés utilizada neste projeto se caracteriza pela associação de um opto-acoplador (U1) – utilizado para proteção das placas Arduino UNO e *shield* GSM –, um transistor (Q1) que controla o fluxo da corrente no relé, um diodo (D1) e um relé, para cada de módulo relé. Apesar da placa utilizada possuir 8 módulos relés, apenas um é utilizado para o acionamento do portão elétrico.

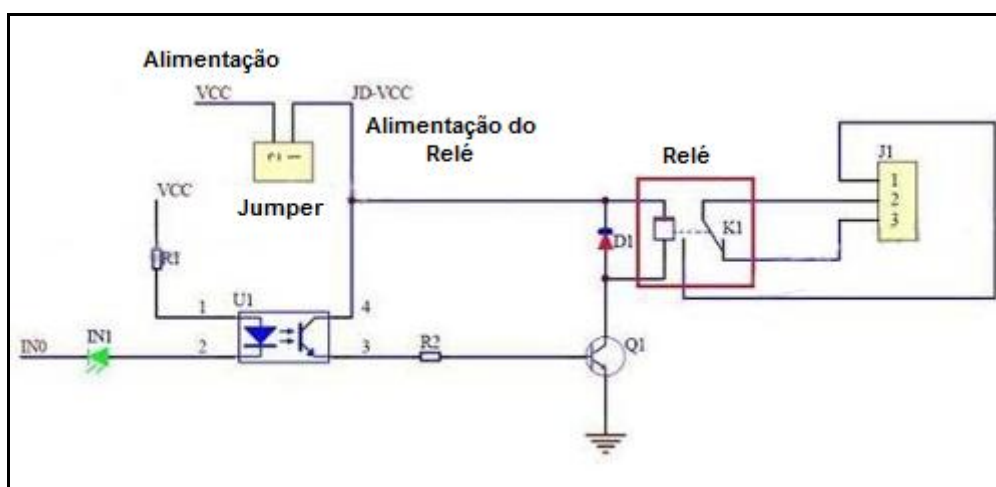


Figura 4.8 – Circuito do módulo relé. Fonte: MODULO RELE

CAPÍTULO 5 – IMPLEMENTAÇÃO

Neste capítulo são descritos, detalhadamente, os procedimentos necessários para o desenvolvimento deste projeto. Esses procedimentos foram divididos da seguinte forma:

- Modelagem do sistema;
- Elaboração do código fonte;
- Montagem do protótipo;

5.1 – Modelagem do sistema

Na primeira etapa da implementação foi realizada a modelagem do sistema proposto e, para tal, foram desenvolvidos três fluxogramas para melhor ilustrar o funcionamento desejado do protótipo.

O primeiro fluxograma – mostrado na figura 5.1 – ilustra o fluxo geral do sistema, numa visão macroscópica de suas funcionalidades como a verificação se o módulo GSM recebeu alguma mensagem de texto ou alguma chamada.

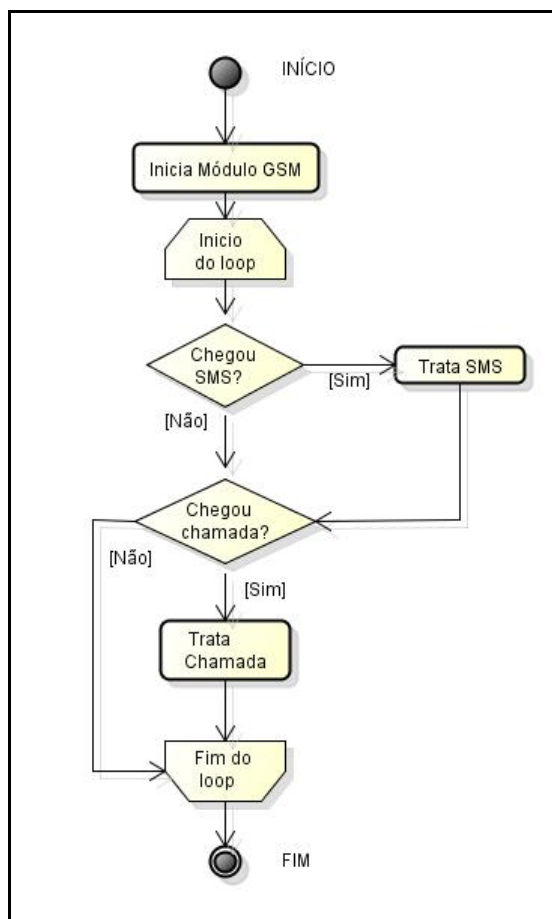


Figura 5.1 – Fluxograma geral. Fonte: AUTOR

No segundo fluxograma – mostrado na figura 5.2 –, é ilustrado o fluxo do processo ‘Trata SMS’, detalhando as funções contidas nele. Este fluxograma demonstra a lógica de registro ou remoção de usuário através de uma mensagem de texto, assim como o registro do ‘Administrador’ que será o responsável pelo registro de novos usuários ou remoção de usuários que não poderão mais acionar o portão elétrico.

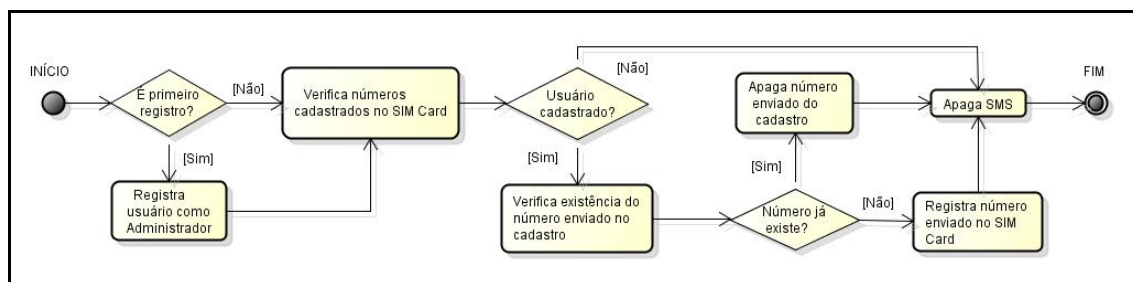


Figura 5.2 – Fluxograma ‘Trata SMS’. Fonte: AUTOR

No último fluxograma – figura 5.3 –, é ilustrado o fluxo do processo ‘Trata Chamada’. Neste processo é verificado se o usuário que está efetuando a chamada está ou não cadastrado no SIM Card e libera a linha do módulo GSM para a não obstrução do seu canal de comunicação. É neste processo que o usuário aciona o portão elétrico, caso esteja cadastrado.

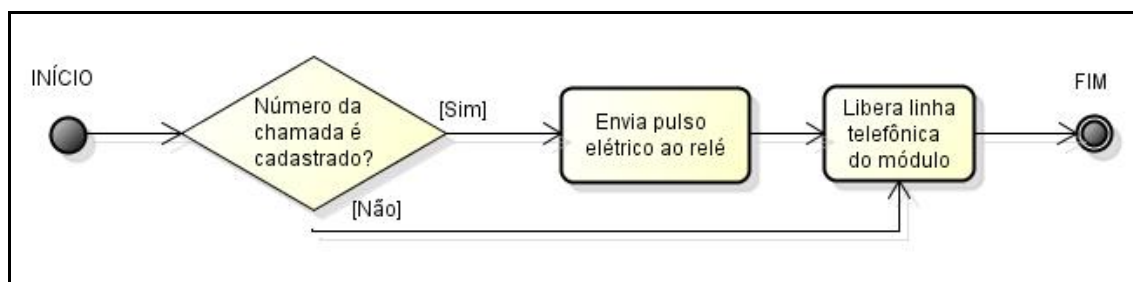


Figura 5.3 – Fluxograma ‘Trata Chamada’. Fonte: AUTOR

5.2 – Elaboração do código fonte

Após a modelagem do sistema, já é possível a elaboração do código fonte que será salvo no microcontrolador. Para melhor observação do comportamento do sistema, foram desenvolvidos dois códigos fonte similares: o código fonte para fase de teste e o código fonte para protótipo final. O código de teste possui apenas duas diferenças funcionais:

- Comunicação serial:
O código fonte de teste faz com que o protótipo envie ao monitor serial do Arduino IDE todas etapas ocorridas em forma de texto, facilitando a visualização de toda a atividade realizada pelo protótipo.
- Relatório periódico:
O código fonte envia a cada período de tempo (pré-definido pelo autor) um relatório com todos os telefones cadastrados no SIM Card para o monitor serial, possibilitando o acompanhamento da atualização dos registros.

Apesar dessas duas funções adicionais do código fonte de teste, ambos os códigos executam o que é esperado do modelo apresentado na seção 5.1 –

Modelagem do sistema. Ambos os códigos foram desenvolvidos, compilados e testados no Arduino IDE.

5.2.1 – Escrita do código fonte

Todo o código foi escrito na linguagem de programação do Arduino, que implementa *Processing* e C. De acordo com o fluxograma apresentado na seção 5.1 – Modelagem do sistema – e a pinagem utilizada definida na tabela 4.2 – Pinagem da placa Arduino UNO utilizada neste projeto – é possível definir as diretivas de pré-compilação e as variáveis.

```

/*****
*****      DIRETIVAS DE PRÉ-COMPILAÇÃO      *****
*****
*****/
#include <GSM_Shield.h>          //Biblioteca do Módulo GSM
#define tamanhoTelefone 20      //Número de dígitos do telefone
#define tamanhoSMS 25          //Tamanho do texto do SMS
#define capacidadeTelefones 20 //Capacidade de armazenamento do SIM Card
#define pinoRele 2             //Pino para onde será enviado o pulso elétrico
/*****
*****      VARIÁVEIS      *****
*****
*****/
GSM gsm;
char telefone[tamanhoTelefone];
char sms[tamanhoSMS];
int erro;
int posicao_sms;
int i;
boolean deletarNumero;
unsigned long tempo;

```

Figura 5.4 – Código fonte: Diretivas e Variáveis. Fonte: AUTOR

Após definir as diretivas de pré-compilação e as variáveis, é necessário definir que parte do código fonte deve executar apenas uma única vez. Na linguagem de programação Arduino, utiliza-se a função *setup* para o que se executará apenas na inicialização do sistema.

```

void setup()
{
  pinMode(pinoRele,OUTPUT);
  Serial.begin(9600);
  gsm.TurnOn(9600);
  gsm.InitParam(1);
  gsm.Echo(0);
  Serial.println("Verificacao inicial de cadastro...");
  for (i=1;i!=capacidadeTelefones;i++)
  {
    erro=gsm.GetPhoneNumber(i,telefone);
    if(erro==1)
    {
      if (i==1) Serial.println("Telefones cadastrados:");
      Serial.print(i);
      Serial.print(" ");
      Serial.println(telefone);
    }
    else if (i==1)
    {
      Serial.println("Nenhum administrador cadastrado.");
      break;
    }
  }
  tempo = millis();
}

```

Figura 5.5 – Código fonte: Função *setup*. Fonte: AUTOR

Essa função de inicialização basicamente define qual pino será utilizado para enviar o pulso elétrico (pinoRele), inicia a comunicação serial para o envio de informações ao monitor serial, liga e configura o módulo GSM de acordo com a necessidade do protótipo. A tabela 5.1 mostra cada função utilizada e uma breve descrição dessas.

Tabela 5.1 – Comandos da função *setup*

Nome	Descrição
<code>pinMode(i,OUTPUT)</code>	Define o pino 'i' como saída
<code>Serial.begin(9600)</code>	Inicia comunicação serial na taxa escolhida
<code>gsm.TurnOn(9600)</code>	Inicia módulo GSM na taxa escolhida e verifica comunicação com o módulo através do comando "AT"
<code>gsm.InitParam(1)</code>	Configura módulo GSM de acordo com os comandos pré-determinados: <ul style="list-style-type: none"> - AT&F - AT+CLIP=1 - AT+CMEE=0 - AT+CMGF=1 - AT+CPBS="SM"
<code>gsm.Echo(0)</code>	Desabilita eco das informações no módulo GSM através do comando: ATE0
<code>Serial.println("texto")</code> ou <code>Serial.print("texto")</code>	Envia "texto" para o monitor serial
<code>gsm.GetPhoneNumber(i,telefone)</code>	Verifica se existe algum número de telefone na posição 'i' e carrega este número em "telefone" através do comando AT+CPBR=i. Retorna 1 caso encontre registro e 0 se não encontrar nada na posição 'i'.

Fonte: AUTOR

Depois de inicializado, o sistema entra em laço de repetição infinito, definido na linguagem de programação Arduino coma a função *loop*. A figura 5.6 ilustra o laço de repetição de acordo com a modelagem proposta.

```

void loop()
{
  trataSMS();
  trataChamada();
  if (millis()-tempo>60000)
  {
    tempo = millis();
    Serial.println();
    Serial.println("=====");
    Serial.println("Relatorio de Numeros Cadastrados:");
    for (i=1;i!=capacidadeTelefones;i++)
    {
      erro=gsm.GetPhoneNumber(i,telefone);
      if(erro==1)
      {
        Serial.print(i);
        Serial.print(" ");
        Serial.println(telefone);
      }
    }
    Serial.println("=====");
    Serial.println();
  }
}

```

Figura 5.6 – Código fonte: Função *loop*. Fonte: AUTOR

Neste laço de repetição infinito tem-se o corpo principal do sistema, onde se irá tratar o SMS recebido – para cadastro ou remoção de usuários –, a chamada recebida – acionando o portão elétrico para o usuário cadastrado – e, para a fase de teste, gerar um relatório a cada minuto (60000 milissegundos) dos telefones cadastrados.

Quando o módulo GSM receber uma mensagem de texto, a função “trataSMS” irá seguir os seguintes passos:

1. Verificar se este é o primeiro SMS recebido, no caso positivo, registra o usuário que enviou o SMS como Administrado do sistema – o único capaz de cadastrar ou remover usuários – e avança para o próximo passo. Caso já exista algum cadastro pula este passo.
2. Verifica se o corpo do SMS contem apenas números (de 0 até 9), em caso positivo, avança para o próximo passo. Em caso negativo, pula para passo 6.

3. Verifica se o número no corpo do SMS já está cadastrado. Em caso positivo avança para o próximo passo e em caso negativo pula para o passo 5.
4. Apaga número já existente no cadastro, ou seja, remove o usuário. Avança para passo 6.
5. Verifica se existe espaço para cadastrar um novo número. Em caso positivo, cadastra o número do corpo do SMS na primeira posição disponível. Em caso negativo, avança para próximo passo.
6. Apaga SMS recebido para economizar espaço de memória.

Depois da função “trataSMS” ser executada, a função “trataChamada” é executada. A figura 5.7 ilustra o código fonte desta função.

```
void trataChamada()
{
  erro = gsm.CallStatusWithAuth(telefone,1,capacidadeTelefones);
  switch(erro)
  {
    case CALL_INCOM_VOICE_AUTH:
      Serial.println("Usuario registrado.");
      Serial.print("Numero: ");
      Serial.println(telefone);
      Serial.println("Acionando portao...");
      digitalWrite(2,HIGH);
      delay(500);
      digitalWrite(2,LOW);
      gsm.HangUp();
      break;
    case CALL_INCOM_VOICE_NOT_AUTH:
      Serial.println("Usuario nao registrado.");
      Serial.print("Numero: ");
      Serial.println(telefone);
      gsm.HangUp();
      break;
    case CALL_ACTIVE_VOICE:
    case CALL_INCOM_DATA_AUTH:
    case CALL_INCOM_DATA_NOT_AUTH:
    case CALL_ACTIVE_DATA:
      gsm.HangUp();
      break;
    default:
      break;
  }
}
```

Figura 5.7 – Código fonte: Função trataChamada. Fonte: AUTOR

5.3 – Montagem do protótipo

Para a montagem do protótipo foi cortado um quadrado de madeira de 13,5cm de lado e aparafusados tanto a placa Arduino UNO quanto a placa com os 8 módulos relés, como mostra a figura 5.8.

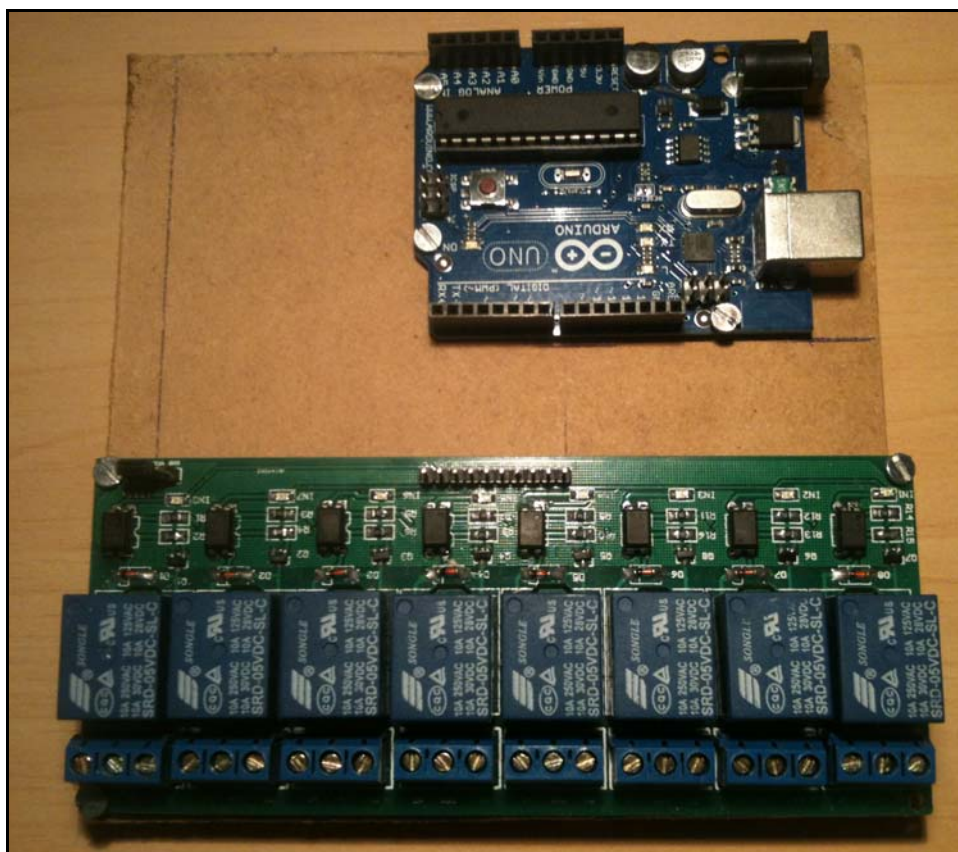


Figura 5.8 – Protótipo fase 1. Fonte: AUTOR

Após a fixação dos itens ao quadrado de madeira, conectou-se o módulo GSM sobre a placa Arduino UNO e os fios foram conectados de acordo com o destacado na tabela 4.2.

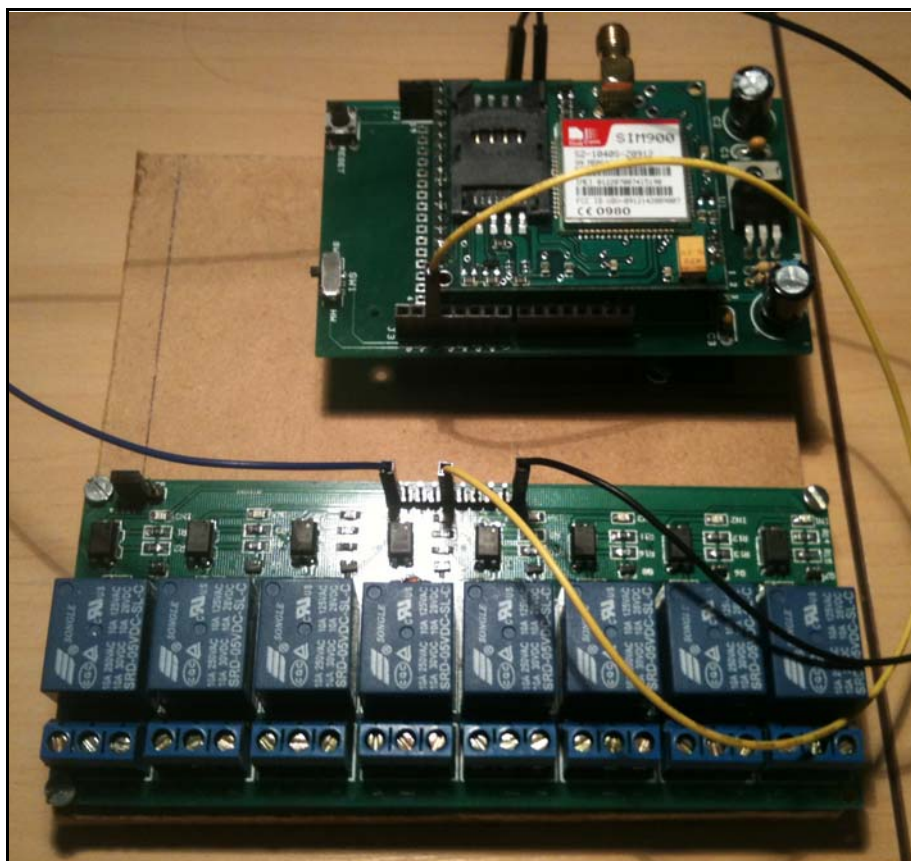


Figura 5.9 – Protótipo fase 2. Fonte: AUTOR

Depois de conectar os circuitos, foi inserido o SIM Card no módulo GSM. O quadrado de madeira com os componentes foi colocado dentro de uma caixa branca de papelão, com base quadrada com 14 cm de lado e altura de 7 cm, cortada de acordo com as necessidades do protótipo. Foram também conectados a antena para recepção de sinal telefônico e a alimentação do protótipo – um transformador de 12 V comum.



Figura 5.10 – Protótipo fase 3. Fonte: AUTOR

Finalmente, após todas as etapas, a caixa de papelão foi fechada para a visão esteticamente mais agradável do protótipo, deixando apenas a antena e as conexões para os módulos relés – onde serão ligados os fios da botoeira do portão elétrico – expostos.

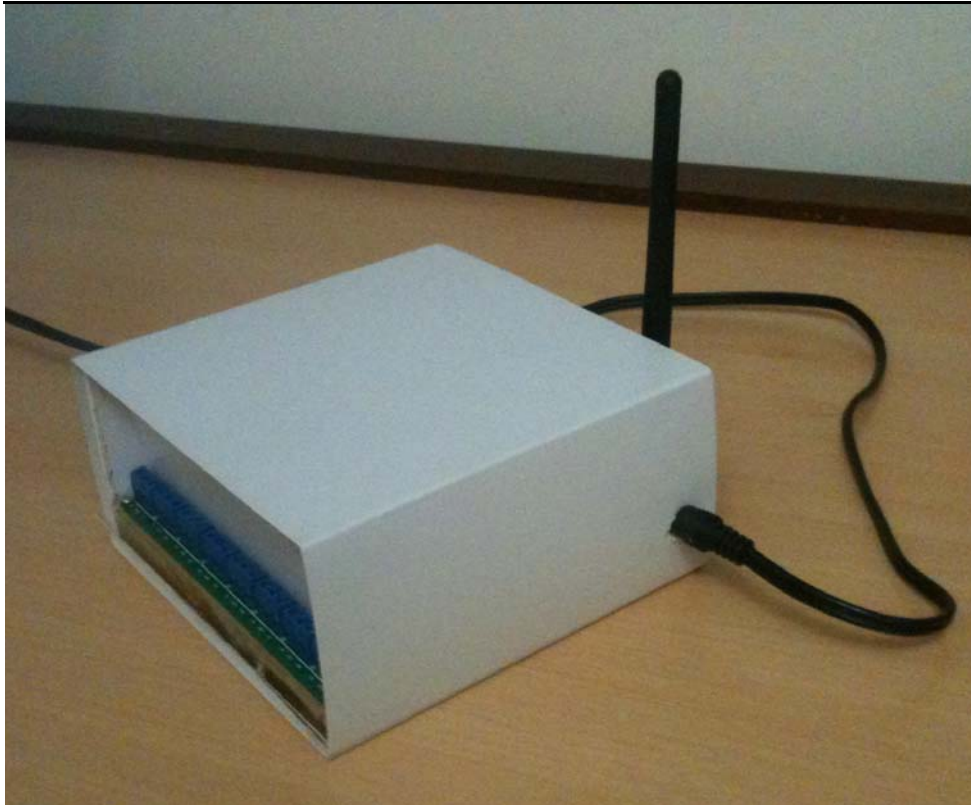


Figura 5.11 – Protótipo fase final. Fonte: AUTOR

CAPÍTULO 6 – RESULTADOS OBTIDOS

Neste capítulo são apresentados os resultados obtidos através de testes e simulações para chegar à solução do problema apresentado, atingindo os objetivos propostos. Serão apresentados nesse capítulo, também, alguns problemas encontrados no decorrer do desenvolvimento deste projeto.

Devido à dificuldade de documentação por meio de fotografias, do acionamento do portão elétrico via celular através de microcontrolador, alguns vídeos com demonstrações e testes foram feitos e disponibilizados no site YouTube. Estes *links* estão disponíveis na apêndice A.

6.1 – Simulações

Nessa seção, são testadas todas as funcionalidades propostas ao sistema, no intuito de simular um ambiente real. As funcionalidades simuladas tiveram como objetivo atingir as seguintes metas:

- Cadastrar o primeiro usuário que enviar mensagem de texto como administrador do sistema;
- Adicionar usuários através de SMS do administrador;
- Remover usuários através de SMS do administrador;
- Acionar remotamente o portão elétrico quando usuário cadastrado efetuar uma chamada ao módulo GSM;

Antes de realizar as simulações, todos os componentes foram testados. A figura 6.1 ilustra a verificação da placa Arduino UNO através de seu acionamento.



Figura 6.1 – Placa Aruino UNO em funcionamento. Fonte: AUTOR

Na figura 6.2, observa-se o módulo GSM conectado sobre a placa Arduino UNO associado à placa com 8 módulos relé. Destacados nos retângulos vermelhos estão os LEDs indicando que o sistema está ativado.

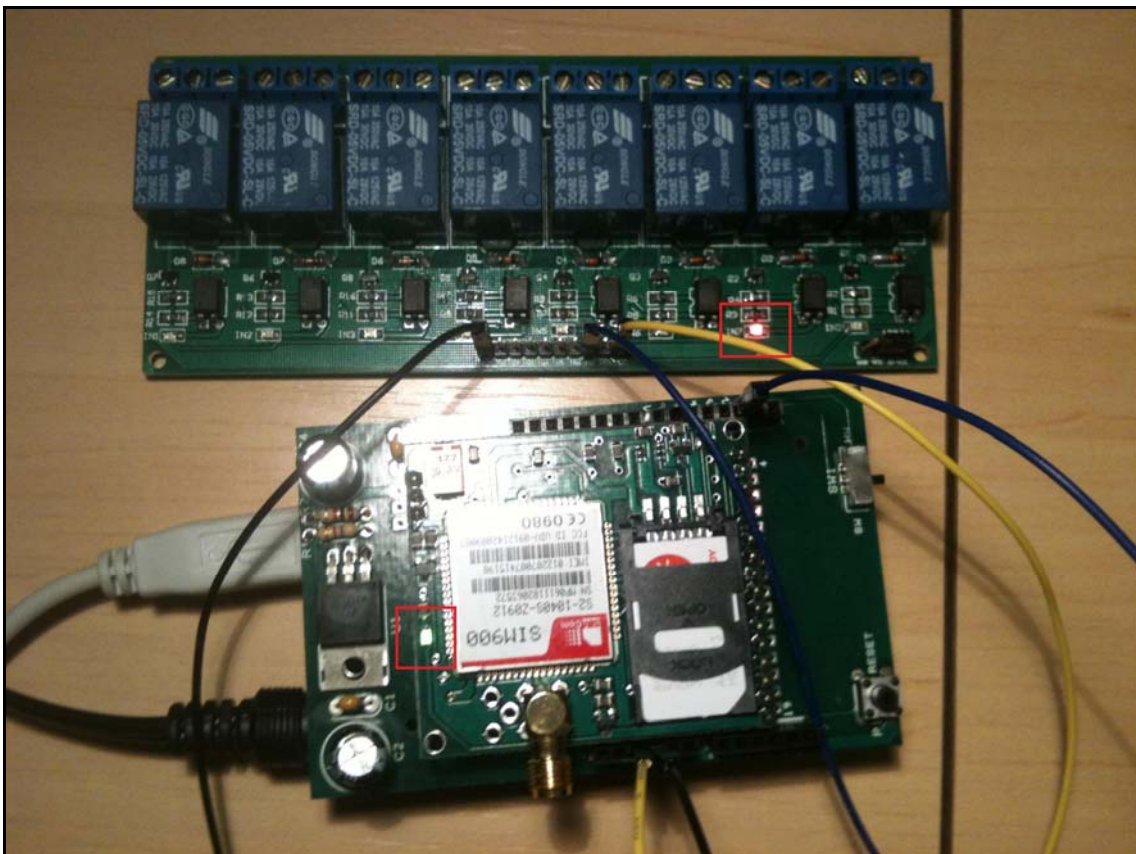


Figura 6.2 – Sistema ativado. Fonte: AUTOR

6.1.1 – Cadastrar o primeiro usuário que enviar mensagem de texto como administrador do sistema

Esta simulação tem como finalidade observar o funcionamento do dispositivo desde o momento em que é ligado até o cadastro do primeiro usuário que enviar um SMS como administrador do sistema. Também foi testado, nesta simulação, o caso em que um usuário efetua uma chamada antes do cadastramento do administrador. Este teste foi executado para cada uma das operadoras a seguir: Claro®, Oi® e TIM®. O autor, que envia o SMS e efetua a chamada, repetiu a simulação 10 vezes para cada operadora citada, totalizando 30 simulações. As etapas realizadas nesta simulação foram:

1. Ligar o protótipo utilizando um transformador de 12V, conectar o cabo USB na placa Arduino UNO e no computador do autor e, finalmente, iniciar a execução do programa com a inicialização do monitor serial.

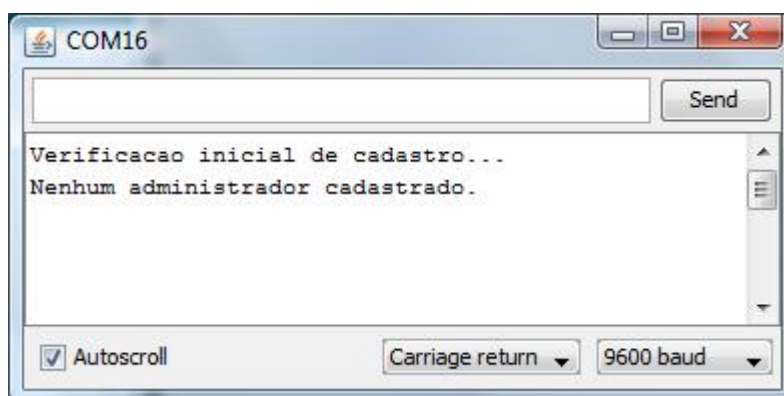


Figura 6.3 – Tela do monitor serial: Iniciando sistema.
Fonte: AUTOR

2. Efetuar chamada ao sistema antes de cadastrar o administrador.

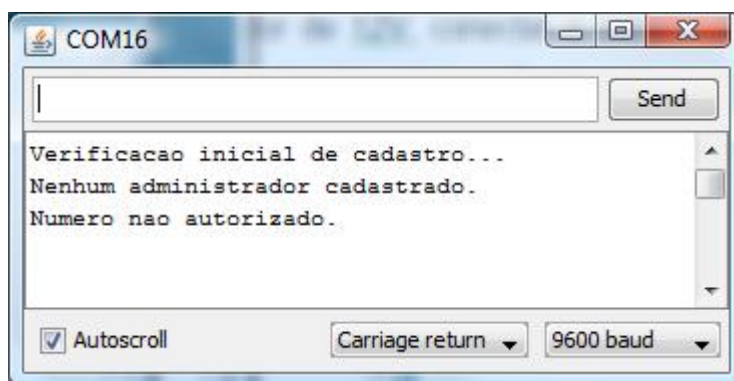


Figura 6.4 – Tela do monitor serial: Número não autorizado. Fonte: AUTOR

3. Enviar primeiro SMS ao sistema para a definição do administrador e o cadastrado do número enviado no corpo do SMS.

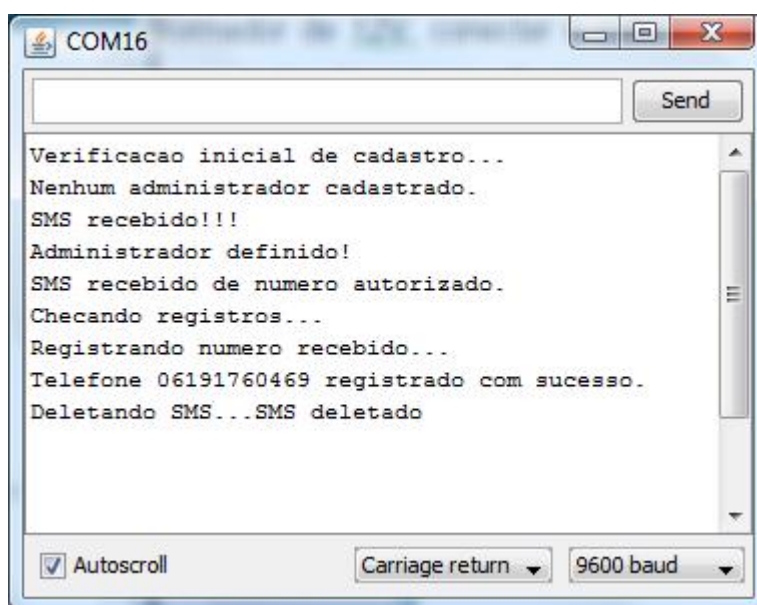


Figura 6.5 – Tela do monitor serial: Definição do administrador e cadastro do primeiro usuário. Fonte: AUTOR

4. Confirmar cadastro do administrador e do número enviado por SMS através de relatório periódico do sistema.

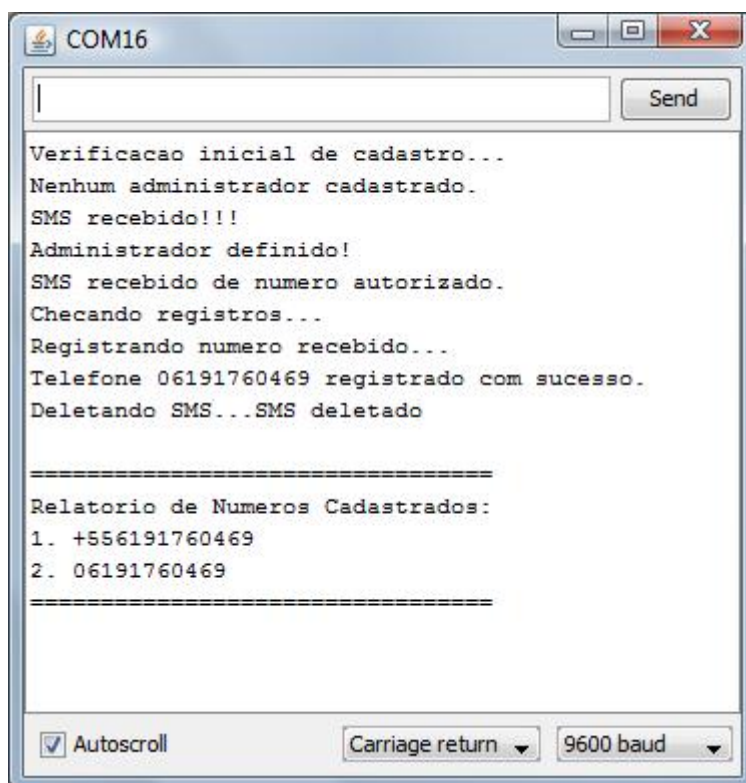


Figura 6.6 – Tela do monitor serial: Confirmação do cadastro do administrador e número enviado. Fonte: AUTOR

Os resultados obtidos nessas simulações do sistema demonstraram funcionamento do protótipo da forma esperada. Em todos os testes o sistema não demonstrou falhas na fase de inicialização do protótipo até a execução da etapa 1. Em todos os testes realizados, na etapa 2, o sistema funcionou como esperado, não autorizando o usuário não cadastrado. Na etapa 3, o tempo para o recebimento da mensagem de texto pelo protótipo oscilou entre 10 e 30 segundos para 27 das 30 simulações. Das 3 mensagens de texto que extrapolaram o tempo padrão, duas foram da operadora Claro® e uma da operadora TIM®. Uma da operadora Claro® chegou 2 minutos após o envio enquanto a outra não chegou após 10 minutos do envio. O SMS da operadora enviado da operadora TIM® que passou do tempo padrão demorou 5 minutos para ser recebido após seu envio.

6.1.2 – Adicionar usuários através de SMS do administrador

Esta simulação teve como finalidade observar o funcionamento do sistema já acionado, após a definição do administrador, quando este tenta cadastrar um novo usuário do sistema. Também foi testado nesta simulação o caso em que outro indivíduo, outro que o administrador, tenta realizar o cadastro de um novo usuário. Este teste foi executado para cada uma das operadoras a seguir: Claro®, Oi® e TIM®. O autor, que envia o SMS com a solicitação de cadastro, repetiu a simulação 10 vezes para cada operadora citada, totalizando 30 simulações. As etapas realizadas nesta simulação foram:

1. Enviar solicitação de cadastro via SMS por um telefone, que não o administrador, e aguardar confirmação do NÃO cadastramento via relatório periódico.

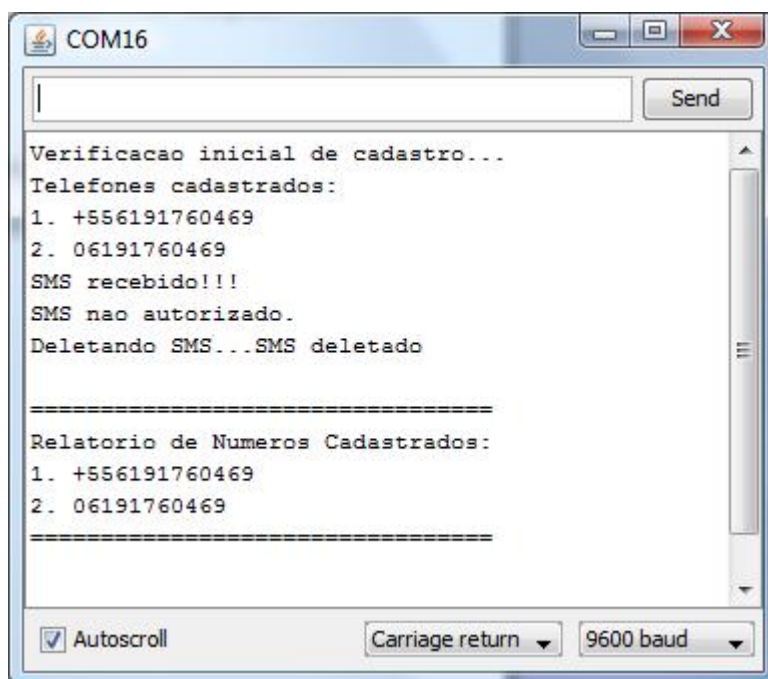


Figura 6.7 – Tela do monitor serial: Cadastro não autorizado e confirmação dos números do SIM Card. Fonte: AUTOR

2. Enviar solicitação de cadastro via SMS através do telefone administrador e aguardar confirmação do cadastro via relatório periódico.

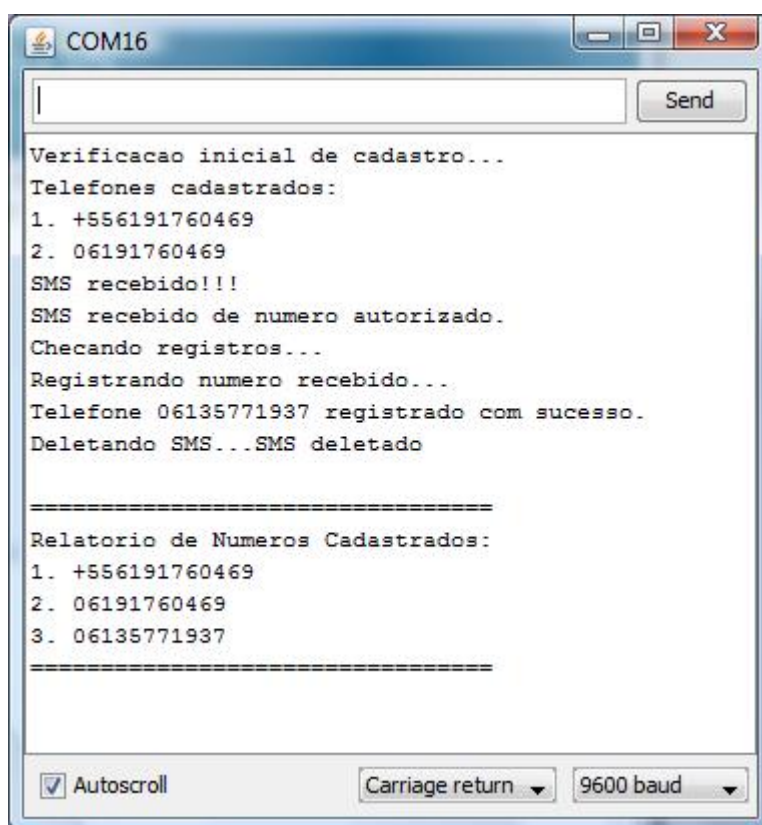


Figura 6.8 – Tela do monitor serial: Cadastro autorizado e confirmação dos números no SIM Card. Fonte: AUTOR

Os resultados obtidos nessas simulações do sistema demonstraram funcionamento esperado do protótipo. Em todas as simulações da etapa 1, onde o usuário que não é administrador tenta cadastrar um novo usuário, o sistema não autorizou tal operação e, através o relatório, foi confirmado o não cadastramento. Em todas as simulações da etapa 2, onde o administrador tenta cadastrar um novo usuário, a operação foi autorizada e confirmada no relatório de números cadastrados.

6.1.3 – Remover usuários através de SMS do administrador

Esta simulação teve como finalidade observar o funcionamento do sistema já acionado, após a definição do administrador e do cadastro de algum usuário, quando o administrador tenta remover um usuário já existente no sistema. Também foi testado nesta simulação o caso em que outro indivíduo,

outro que o administrador, tenta realizar a remoção de um usuário cadastrado. Este teste foi executado para cada uma das operadoras a seguir: Claro®, Oi® e TIM®. O autor, que envia o SMS com a solicitação de remoção, repetiu a simulação 10 vezes para cada operadora citada, totalizando 30 simulações. As etapas realizadas nesta simulação foram:

1. Enviar solicitação de remoção via SMS por um telefone, que não o administrador, e aguardar confirmação de que NÃO houve remoção do cadastro via relatório periódico.

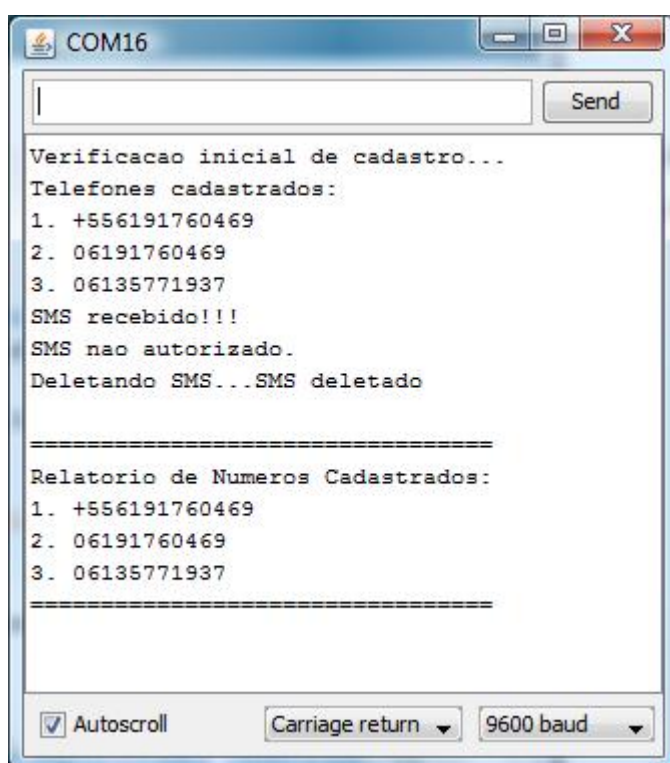


Figura 6.9 – Tela do monitor serial: Remoção não autorizada e confirmação dos números no SIM Card. Fonte: AUTOR

2. Enviar solicitação de remoção via SMS através do telefone administrador e aguardar confirmação da remoção via relatório periódico.

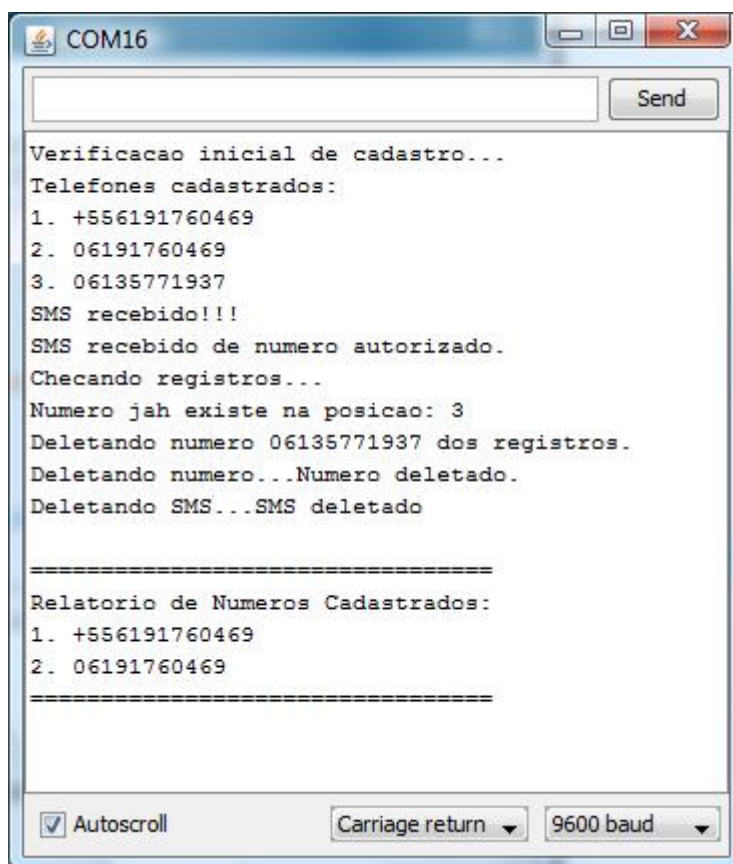


Figura 6.10 – Tela do monitor serial: Remoção autorizada e confirmação dos números do SIM Card. Fonte: AUTOR

Os resultados obtidos nessas simulações do sistema demonstraram funcionamento esperado do protótipo. Em todas as simulações da etapa 1, onde o usuário que não é administrador tenta remover um usuário, o sistema não autorizou tal operação e, através o relatório, foi confirmado que a remoção não foi efetuada. Em todas as simulações da etapa 2, onde o administrador tenta remover um usuário já cadastrado, a operação foi autorizada e confirmada no relatório de números cadastrados.

6.1.4 – Acionar remotamente o portão elétrico quando usuário cadastrado efetuar uma chamada ao módulo GSM

Esta simulação teve como finalidade observar o funcionamento do sistema já acionado, após a definição do administrador e do cadastro de algum usuário, quando algum usuário cadastrado efetua uma chamada ao protótipo e

aciona o portão elétrico. Também foi testado nesta simulação o caso em que outro indivíduo, não cadastrado, tenta acionar o portão elétrico através de uma chamada ao protótipo. Este teste foi executado para cada uma das operadoras a seguir: Claro®, Oi® e TIM®. O autor, que efetuou a chamada ao protótipo, repetiu a simulação 10 vezes para cada operadora citada, totalizando 30 simulações. As etapas realizadas nesta simulação foram:

1. Efetuar chamada de um usuário cadastrado e confirmar o acionamento do portão elétrico através de mensagem mostrada no monitor serial.

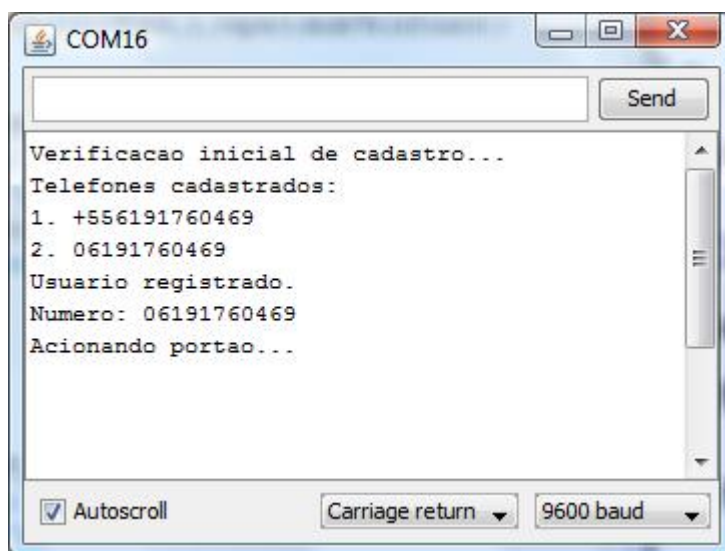


Figura 6.11 – Tela do monitor serial: Acionamento do portão elétrico autorizado. Fonte: AUTOR

2. Efetuar chamada de um usuário não cadastrado e confirmar o NÃO acionamento do portão elétrico através de mensagem mostrada no monitor serial.

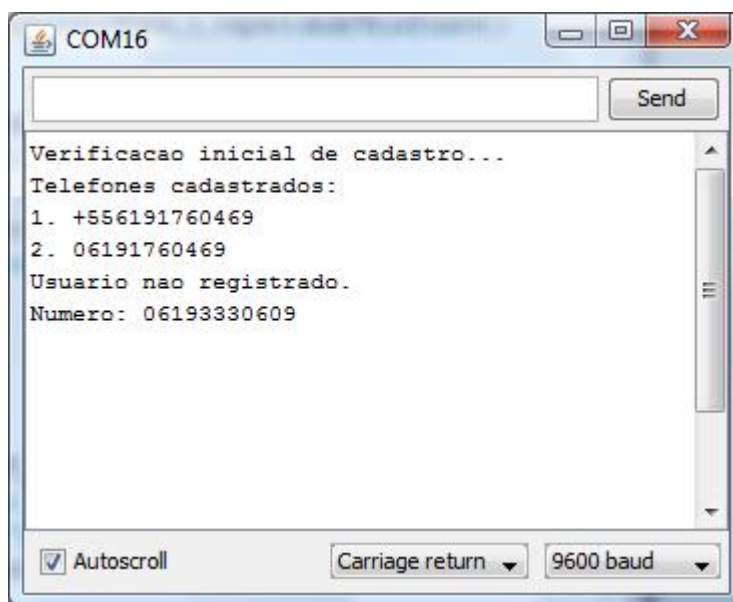


Figura 6.12 – Tela do monitor serial: Acionamento do portão elétrico não autorizado. Fonte: AUTOR

Os resultados obtidos nessas simulações do sistema demonstraram funcionamento esperado do protótipo. Em todas as simulações da etapa 1, onde o usuário cadastrado efetua chamada ao protótipo, o sistema autorizou tal operação e, através de mensagens no monitor serial, foi confirmado o acionamento do portão elétrico. Em todas as simulações da etapa 2, onde um usuário não cadastrado tenta acionar o portão elétrico através de chamada ao protótipo, a operação não foi autorizada e mostrada através de mensagens no monitor serial.

6.2 – Problemas encontrados

Durante o desenvolvimento do sistema, alguns problemas surgiram ao que se refere à autorização de usuários cadastrados. Inicialmente foi verificado que nem todos os usuários disponibilizam o celular para identificação de chamadas, ou seja, bloqueiam o número do telefone, com isso o protótipo não é capaz de identificar a chamada e, conseqüentemente, não é capaz de acionar o dispositivo.

Outro problema encontrado é referente ao cadastramento e remoção de usuários. Inicialmente foi sugerido que todos os usuários cadastrados pudessem adicionar ou remover cadastros do sistema, com isso surgiu um problema de controle de números que estão ou não cadastrados no sistema. Para isso, foi definido que apenas o primeiro cadastro, definido como administrador do sistema, é capaz de cadastrar novos usuários ou remover cadastros. Um dos problemas gerados com essa decisão surgiu na hipótese do administrador perder a linha telefônica (por perda/roubo do celular, por exemplo), com isso será necessária a aquisição de um novo SIM Card para o dispositivo caso os usuários desejem adicionar ou remover cadastros.

Finalmente, por se tratar de um dispositivo que não mostra ao usuário a confirmação de cadastramento ou remoção de outros usuários e que nem todas as mensagens de texto enviadas durante o desenvolvimento e simulação do protótipo foram recebidas, existe a possibilidade que tanto o cadastro quanto a remoção de usuários não ocorra, se fazendo necessária uma verificação adicional, como a tentativa de um novo usuário acionar o portão elétrico para garantir seu cadastro.

CAPÍTULO 7 – CONSIDERAÇÕES FINAIS

7.1 – Conclusões

Neste trabalho foi desenvolvido um protótipo com o objetivo de solucionar os problemas apresentados no início deste documento, como o controle do acionamento do portão elétrico apenas para usuários cadastrados e a substituição do tradicional controle remoto de portões elétricos por uma ligação que pode ser efetuada de um celular que já faz parte da lista de periféricos que a maioria das pessoas carrega diariamente.

Apesar dos problemas encontrados no desenvolvimento deste projeto, os objetivos foram alcançados. Através de todos os testes realizados foi possível avaliar o desempenho do protótipo como satisfatório para as funcionalidades propostas, como a definição do administrador do sistema, a adição e remoção de usuários no cadastro do SIM Card e o acionamento remoto do portão elétrico apenas para clientes registrados.

No entanto, existe a possibilidade das mensagens de texto não chegarem ao dispositivo e esta possibilidade foi atribuída às operadoras de telefonia. No caso de perda da linha telefônica do administrador do sistema, se fará necessária a troca do SIM Card do protótipo, que atualmente custa por volta de R\$10 à R\$20.

É possível, portanto, concluir que os resultados obtidos com o desenvolvimento deste projeto cumpriram as propostas e objetivos planejados e que o protótipo encontra-se em funcionamento e operando de acordo com o previsto e programado.

7.2 – Propostas para trabalhos futuros

Controle e automação residencial tem se provado uma área em grande crescimento nos últimos anos e, com isso, projetos como este tem ganhado espaço no mercado atual. Algumas propostas de desenvolvimento para este projeto são:

- Desenvolver uma placa apenas com os itens necessários para o protótipo, no intuito de baixar o custo final do produto.
- Desenvolver sistema de envio de mensagens de texto ao administrador confirmando o cadastro ou a remoção de um usuário.
- Aplicar o conceito de acionamento remoto via celular através de microcontrolador para utensílios domésticos para maior automação residencial.

REFERÊNCIAS BIBLIOGRÁFICAS

AMOUNT of SMS sent per second worldwide from 2007 to 2010, **Statista – the leading statistics portal**. Disponível em: <<http://www.statista.com/statistics/167048/number-of-sms-sent-per-second-worldwide-since-2007/>> Acesso em: 10 jun. 2012.

ARDUINO, Site do Projeto Arduino. **Arduino**. Disponível em: <arduino.cc/> Acesso: 10 jun. 2012.

ARDUINO GSM SHIELD. **Open Eletronics, Open source eletronic projects**. Disponível em: <<http://www.open-electronics.org/arduino-gsm-shield/>> Acesso: 10 jun. 2012.

ARDUINO UNO, Arduino UNO Front. **Arduino UNO Board**. Disponível em: <<http://arduino.cc/en/uploads/Main/ArduinoUnoFront.jpg>> Acesso: 10 jun. 2012.

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

GIMENEZ, S. P. **Microcontroladores 8051**. 1ª Ed. São Paulo: Pearson, 2005.

GSM REMOTE CONTROL – 2 IN and 2 OUT Part 2 – Schematic **Open Eletronics, Open source eletronic projects**. Publicado em: 7 jan. 2011. Disponível em: <<http://www.open-electronics.org/gsm-remote-control-2-in-and-2-out-part-2-schematic/>> Acesso em: 17 mar. 2012.

GSM World. **GSM**. 2012. Disponível em: <<http://www.gsm.org/>> Acesso em: 10 jun. 2012.

HILLEBRAND, F.; TROSBY, F.; HOLLEY, K.; HARRIS, I. **Short Message Service (SMS): The Creation of Personal Global Text Messaging**. 1ª Ed. Wiley, 2010.

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

MODULO RELE. **Módulo de Relés com 8 canais**. Disponível em: <<http://produto.mercadolivre.com.br/MLB-427565012-modulo-de-reles-8-canais-5v-arduino-arm-avr-pic-automaco- JM>> Acesso: 10 jun. 2012.

NÚMERO de furtos em casas do Lago Norte quase dobra em julho. **G1, o portal de notícias da globo**. Publicado em: 1 ago. 2011. Disponível em: <<http://g1.globo.com/distrito-federal/noticia/2011/08/numero-de-furtos-em-casas-do-lago-norte-quase-dobra-em-julho.html>> Acesso em: 8 jun. 2012.

TIDIGINO, the Arduino-based GSM remote control. **Open Eletronics, Open source eletronic projects**. Publicado em: 5 jan. 2012. Disponível em: <<http://www.open-electronics.org/tidigino-the-arduino-based-gsm-remote-control/>> Acesso em: 17 mar. 2012.

TRÊS homens fazem moradores reféns em casa no Lago Sul e fogem. **Correio Braziliense, Brasília**. Publicado em: 7 jan. 2010. Disponível em: <http://www.correio braziliense.com.br/app/noticia/cidades/2010/01/07/interna_cidadesdf,165080/index.shtml> Acesso em: 8 jun. 2012.

WIKIPEDIA, a enciclopédia livre. **Wikipédia**. Disponível em: <<http://pt.wikipedia.org/wiki/GSM>> Acesso: 10 jun. 2012.

APÊNDICE A – LINKS DE VIDEOS DO AUTOR

Canal de vídeos do autor disponível em:
<<http://www.youtube.com/user/gmfmaia>>

Vídeo: “Apresentação: Dispositivo do Projeto Final”, disponível em
<<http://www.youtube.com/watch?v=p6EhQ5zuqKM&feature=plcp>>

APÊNDICE B – CÓDIGO FONTE DE TESTE

```

/*****
***** DIRETIVAS DE PRÉ-COMPILAÇÃO *****/
/*****
#include <GSM_Shield.h>           //Biblioteca do Módulo GSM
#define tamanhoTelefone 20       //Número de dígitos do telefone
#define tamanhoSMS 25           //Tamanho do texto do SMS
#define capacidadeTelefones 20   //Capacidade de armazenamento do SIM Card
#define pinoRele 2               //Pino para onde será enviado o pulso elétrico
/*****
***** VARIÁVEIS *****/
/*****
GSM gsm;
char telefone[tamanhoTelefone];
char sms[tamanhoSMS];
int erro;
int posicao_sms;
int i;
boolean deletarNumero;
unsigned long tempo;

void setup()
{
  pinMode(pinoRele,OUTPUT);
  Serial.begin(9600);
  gsm.TurnOn(9600);
  gsm.InitParam(1);
  gsm.Echo(0);
  Serial.println("Verificacao inicial de cadastro...");
  for (i=1;i!=capacidadeTelefones;i++)
  {
    erro=gsm.GetPhoneNumber(i,telefone);
    if(erro==1)
    {
      if (i==1) Serial.println("Telefones cadastrados:");
      Serial.print(i);
      Serial.print(" ");
      Serial.println(telefone);
    }
  }
}

```

```

    }
    else if (i==1)
    {
        Serial.println("Nenhum administrador cadastrado.");
        break;
    }
}
tempo = millis();
}

void loop()
{
    trataSMS();
    trataChamada();
    if (millis()-tempo>60000)
    {
        tempo = millis();
        Serial.println();
        Serial.println("=====");
        Serial.println("Relatorio de Numeros Cadastrados:");
        for (i=1;i!=capacidadeTelefones;i++)
        {
            erro=gsm.GetPhoneNumber(i,telefone);
            if(erro==1)
            {
                Serial.print(i);
                Serial.print(". ");
                Serial.println(telefone);
            }
        }
        Serial.println("=====");
        Serial.println();
    }
}

```

```

/*****

```

Nome da funcao: verificaSMS()

Descricao:

Essa funcao verifica se existe algum SMS nao lido no modulo e;
 ou registra o ADMINISTRADOR caso nao existam outros cadastros,
 ou registra NOVO USUARIO no SIM Card caso exista espaco.

*****/

```

void trataSMS()
{
  deletarNumero = false;
  posicao_sms = gsm.IsSMSPresent(SMS_UNREAD);
  //Se chegou SMS
  if (posicao_sms>0)
  {
    Serial.println("SMS recebido!!!");

    //Verifica se eh o primeiro SMS e registra ADM
    erro = gsm.GetPhoneNumber(1,telefone);
    //Nao existem cadastros
    if (erro == 0)
    {
      //Carrega numero e sms
      erro = gsm.GetSMS(posicao_sms,telefone,sms,tamanhoSMS);
      if (erro > 0)
      {
        //Tenta registrar administrador
        erro = gsm.WritePhoneNumber(1,telefone);
        if (erro == 1) Serial.println("Administrador definido!");
        else Serial.println("Erro no cadastro do Administrador!");
      }
    }
    //Verifica se SMS eh de usuario cadastrado
    erro = gsm.GetAuthorizedSMS(posicao_sms,telefone,sms,tamanhoSMS,1,2);
    if (erro == GETSMS_AUTH_SMS)
    {
      //Verifica se SMS contem apenas digitos (0..9)
      i=0;
      while(sms[i]!='\0')
      {
        if ((sms[i]<'0')||((sms[i]>'9'))
        {

```



```

Serial.println("Mensagem contem caracteres diferentes de (0..9)");
break;
}
i++;
}
if (sms[i]!='\0')
{
Serial.println("SMS recebido de numero autorizado.");
Serial.println("Checando registros...");
//Verifica se jah existe o NOVO numero no SIM Card
for (i=1;i!=capacidadeTelefones;i++)
{
erro=gsm.ComparePhoneNumber(i,sms);
//Se encontrar o NOVO numero, habilita flag de deletarNumero
if(erro==1)
{
Serial.print("Numero jah existe na posicao: ");
Serial.println(i);
deletarNumero = true;
break;
}
}
//Caso seja um numero jah cadastrado, deletar numero
if (deletarNumero)
{
Serial.print("Deletando numero ");
Serial.print(sms);
Serial.println(" dos registros.");
//Tenta deletar numero dos registros
erro = gsm.DelPhoneNumber(i);
Serial.print("Deletando numero...");
if (erro==1) Serial.println("Numero deletado.");
else Serial.println("Erro no momento de deletar o numero.");
//
}
//Caso numero nao esteja cadastrado, cadastrar NOVO numero
else
{
//Procura primeira posicao livre no SIM Card
for (i=2;i!=capacidadeTelefones;i++)

```

```

{
    erro=gsm.GetPhoneNumber(i,telefone);
    if(erro==0) break;
}
if (i==capacidadeTelefones)
{
    Serial.println("Capacidade maxima de telefones atingida!");
    Serial.println("Cadastro nao efetuado.");
}
else
{
    Serial.println("Registrando numero recebido...");
    //Tenta registrar numero na primera posicao livre
    erro = gsm.WritePhoneNumber(i,sms);
    if(erro==1)
    {
        Serial.print("Telefone ");
        Serial.print(sms);
        Serial.println(" registrado com sucesso.");
    }
    else
    {
        Serial.println("Problema no cadastro!");
        Serial.print("Erro: ");
        Serial.println(erro);
    }
}
}
}
}
}
//Fim do: if (sms[i]!='\0')
//Fim do: if (erro == GETSMS_AUTH_SMS)
else Serial.println("SMS nao autorizado.");
erro = gsm.DeleteSMS(posicao_sms);
Serial.print("Deletando SMS...");
if(erro==1) Serial.println("SMS deletado");
else Serial.println("Problema no momento de apagar o SMS");
//Fim do: if (posicao_sms>0)
}

/*****
Nome da funcao: verificaChamada()

```

Descricao:

Essa funcao verifica se existe alguma chamada no modulo, verifica se o numero da chamada esta registrado no SIM Card e manda um pulso eletrico para o pinoRele, acionando o portao.

*****/

```
void trataChamada()
{
  erro = gsm.CallStatusWithAuth(telefone,1,capacidadeTelefones);
  switch(erro)
  {
    case CALL_INCOM_VOICE_AUTH:
      Serial.println("Usuario registrado.");
      Serial.print("Numero: ");
      Serial.println(telefone);
      Serial.println("Acionando portao...");
      digitalWrite(2,HIGH);
      delay(500);
      digitalWrite(2,LOW);
      gsm.HangUp();
      break;
    case CALL_INCOM_VOICE_NOT_AUTH:
      Serial.println("Usuario nao registrado.");
      Serial.print("Numero: ");
      Serial.println(telefone);
      gsm.HangUp();
      break;
    case CALL_ACTIVE_VOICE:
    case CALL_INCOM_DATA_AUTH:
    case CALL_INCOM_DATA_NOT_AUTH:
    case CALL_ACTIVE_DATA:
      gsm.HangUp();
      break;
    default:
      break;
  }
}
```

APÊNDICE C – CÓDIGO FONTE FINAL

```

/*****
***** DIRETIVAS DE PRÉ-COMPILAÇÃO *****/
*****/

#include <GSM_Shield.h>           //Biblioteca do Módulo GSM
#define tamanhoTelefone 20       //Número de dígitos do telefone
#define tamanhoSMS 25           //Tamanho do texto do SMS
#define capacidadeTelefones 20   //Capacidade de armazenamento do SIM Card
#define pinoRele 2               //Pino para onde será enviado o pulso elétrico
/*****
***** VARIÁVEIS *****/
*****/

GSM gsm;
char telefone[tamanhoTelefone];
char sms[tamanhoSMS];
int erro;
int posicao_sms;
int i;
boolean deletarNumero;

void setup()
{
  pinMode(pinoRele,OUTPUT);
  Serial.begin(9600);
  gsm.TurnOn(9600);
  gsm.InitParam(1);
  gsm.Echo(0);
}

void loop()
{
  trataSMS();
  trataChamada();
}

/*****
Nome da funcao: verificaSMS()

```

Descricao:

Essa funcao verifica se existe algum SMS nao lido no modulo e;
ou registra o ADMINISTRADOR caso nao existam outros cadastros,
ou registra NOVO USUARIO no SIM Card caso exista espaco.

*****/

```
void trataSMS()
{
    deletarNumero = false;
    posicao_sms = gsm.IsSMSPresent(SMS_UNREAD);
    //Se chegou SMS
    if (posicao_sms>0)
    {
        //Verifica se eh o primeiro SMS e registra ADM
        erro = gsm.GetPhoneNumber(1,telefone);
        //Nao existem cadastros
        if (erro == 0)
        {
            //Carrega numero e sms
            erro = gsm.GetSMS(posicao_sms,telefone,sms,tamanhoSMS);
            if (erro > 0)
            {
                //Tenta registrar administrador
                erro = gsm.WritePhoneNumber(1,telefone);
            }
        }
        //Verifica se SMS eh de usuario cadastrado
        erro = gsm.GetAuthorizedSMS(posicao_sms,telefone,sms,tamanhoSMS,1,2);
        if (erro == GETSMS_AUTH_SMS)
        {
            //Verifica se SMS contem apenas digitos (0..9)
            i=0;
            while(sms[i]!='\0')
            {
                if ((sms[i]<'0')||((sms[i]>'9'))
                {
                    break;
                }
                i++;
            }
        }
    }
}
```

```
}
if (sms[i]!='\0')
{
    //Verifica se jah existe o NOVO numero no SIM Card
    for (i=1;i!=capacidadeTelefones;i++)
    {
        erro=gsm.ComparePhoneNumber(i,sms);
        //Se encontrar o NOVO numero, habilita flag de deletarNumero
        if(erro==1)
        {
            Serial.println(i);
            deletarNumero = true;
            break;
        }
    }
    //Caso seja um numero jah cadastrado, deletar numero
    if (deletarNumero)
    {
        //Tenta deletar numero dos registros
        erro = gsm.DelPhoneNumber(i);
    }
    //Caso numero nao esteja cadastrado, cadastrar NOVO numero
    else
    {
        //Procura primeira posicao livre no SIM Card
        for (i=2;i!=capacidadeTelefones;i++)
        {
            erro=gsm.GetPhoneNumber(i,telefone);
            if(erro==0) break;
        }
        if (i==capacidadeTelefones)
        {
            break;
        }
        else
        {
            //Tenta registrar numero na primera posicao livre
            erro = gsm.WritePhoneNumber(i,sms);
        }
    }
}
```

```

    }//Fim do: if (sms[i]!='\0')
  }//Fim do: if (erro == GETSMS_AUTH_SMS)
  erro = gsm.DeleteSMS(posicao_sms);
  }//Fim do: if (posicao_sms>0)
}

```

```

/*****
Nome da funcao: verificaChamada()

```

Descricao:

Essa funcao verifica se existe alguma chamada no modulo, verifica se o numero da chamada esta registrado no SIM Card e manda um pulso eletrico para o pinoRele, acionando o portao.

```

*****/

```

```

void trataChamada()
{
  erro = gsm.CallStatusWithAuth(telefone,1,capacidadeTelefones);
  switch(erro)
  {
    case CALL_INCOM_VOICE_AUTH:
      digitalWrite(2,HIGH);
      delay(500);
      digitalWrite(2,LOW);
      gsm.HangUp();
      break;
    case CALL_INCOM_VOICE_NOT_AUTH:
      gsm.HangUp();
      break;
    case CALL_ACTIVE_VOICE:
    case CALL_INCOM_DATA_AUTH:
    case CALL_INCOM_DATA_NOT_AUTH:
    case CALL_ACTIVE_DATA:
      gsm.HangUp();
      break;
    default:
      break;
  }
}

```

ANEXO A – ESQUEMÁTICO: ARDUINO UNO

