

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
Campus DIVINÓPOLIS
GRADUAÇÃO EM ENGENHARIA MECATRÔNICA

DESENVOLVIMENTO DE UM SISTEMA DE RASTREAMENTO E
ATUAÇÃO À DISTÂNCIA PARA AUTOMÓVEIS.

Divinópolis
2019

Paulo Ivan Vieira Marinho

DESENVOLVIMENTO DE UM SISTEMA DE RASTREAMENTO E
ATUAÇÃO À DISTÂNCIA PARA AUTOMÓVEIS.

Monografia de Trabalho de Conclusão de Curso
apresentado ao Colegiado do Curso de Graduação
em Engenharia Mecatrônica como parte dos requi-
sitos exigidos para a obtenção do título de Enge-
nheiro Mecatrônico.

Áreas de integração: Eletrônica e Computação.

Orientador: Prof. Doutor Christian Gonçalves
Herrera.

Divinópolis
2019

Paulo Ivan Vieira Marinho

DESENVOLVIMENTO DE UM SISTEMA DE RASTREAMENTO E
ATUAÇÃO À DISTÂNCIA PARA AUTOMÓVEIS.

Monografia de Trabalho de Conclusão de Curso
apresentado ao Colegiado do Curso de Graduação
em Engenharia Mecatrônica como parte dos requi-
sitos exigidos para a obtenção do título de Enge-
nheiro Mecatrônico.

Eixos de formação: Eletrônica e Computação.

Comissão Avaliadora:

Prof. Dr. João Carlos de Oliveira

Eng. Mecatrônica / CEFET/MG *Campus V*

Prof. Me. Nestor Dias de Oliveira Volpini

Eng. Mecatrônica / CEFET/MG *Campus V*

Prof. Dr. Christian Gonçalves Herrera

Eng. Mecatrônica / CEFET/MG *Campus V*

Divinópolis
2019

Resumo

O projeto proposto visou o desenvolvimento de um sistema de rastreamento e atuação em automóveis à distância. Tais tecnologias vêm sendo amplamente usadas, tentando-se mitigar a quantidade de furto de veículos existente. Contudo, os sistemas de rastreamento atuais normalmente são disponibilizados por empresas seguradoras, havendo um custo adicional para o usuário utilizar esta tecnologia. Além disto, os dispositivos antifurto vêm se tornando cada vez mais ineficazes devido aos criminosos estarem aprendendo como burlar as medidas de segurança. Assim, teve-se como intenção criar um sistema onde o usuário pudesse requisitar informações de seu veículo e também atuar sobre o mesmo através de um celular qualquer via SMS, podendo desligar o automóvel ou realizar outras operações similares. Objetivou-se também que não fosse necessário se arcar com custos de manter um servidor ou ter que contar com uma empresa prestadora de serviços. Além disto, foi desenvolvido um circuito receptor GSM com conexão USB, onde o mesmo se comunica com um computador. Tendo sido também desenvolvido um programa que trata as informações dadas por este circuito e permite o usuário verificar a localização do automóvel em um mapa global. Para isso, além dos módulos GPS e GSM, foi utilizado microcontroladores, sensores e diversos componentes eletrônicos. Almejou-se assim, ao se construir o protótipo, testá-lo em um carro. Por fim, desejou-se com este projeto desenvolver um sistema de rastreamento e atuação veicular mais eficiente e com melhor custo-benefício do que os dispositivos existentes no mercado.

Palavras-chave: Rastreamento, GPS, GSM, microcontrolador.

Sumário

Lista de Figuras	viii
Lista de Tabelas	ix
Lista de Acrônimos e Notação	x
1 Introdução	1
1.1 Definição do Problema	3
1.2 Motivação	3
1.3 Objetivos do Trabalho	3
1.3.1 Objetivo Geral	3
1.3.2 Objetivos Específicos	3
1.4 Organização do Documento	4
2 Fundamentos	5
2.1 Revisão de Literatura	5
2.2 Estado da Arte	6
2.3 Fundamentação Teórica	7
2.3.1 Protocolos de Comunicação	7
2.3.2 UART	7
2.3.3 Padrão TTL	7
2.3.4 Comunicação USB	8
2.3.5 Sistema de Posicionamento Global (GPS)	8
2.3.6 Padrão NMEA	9
2.3.7 Módulo Receptor GPS	11
2.3.8 Módulo GSM	12
2.3.9 Conversores DC-DC	13
2.3.10 CIs Reguladores de Tensão	13
2.3.11 Microcontroladores	15
2.3.12 Linguagem de Programação JAVA	17
2.3.13 Linguagem de Programação C#	17
2.3.14 Linguagem de Programação C	17
2.3.15 Proteus	18
2.3.16 MikroC PRO for PIC	18

3	Metodologia	19
3.1	Projeto Comportamental	19
3.1.1	Diagramas de Estados	20
3.1.2	Diagramas de Processos	23
3.1.3	Análise de Falhas	26
3.2	Projeto Lógico	27
3.2.1	Escolha dos Microcontroladores	27
3.2.2	Escolha dos Módulos GSM	28
3.2.3	Escolha do Módulo GPS	29
3.2.4	Projeto das Fontes de Alimentação	30
3.2.5	Projeto do Circuito de Sensores e Atuadores	31
3.2.6	Esquemáticos dos Dispositivos de Hardware	32
3.3	Projeto Físico	33
3.3.1	Dispositivo de Monitoramento	33
3.3.2	Programa de Rastreamento no Computador	39
3.3.3	Dispositivo Embarcado	40
4	Resultados e Análise de Resultados	43
5	Considerações Finais	51
5.1	Conclusões	51
A	Circuitos Esquemáticos e Tabelas	52
	Referências	58

Lista de Figuras

2.1	Segmentos do Sistema de Posicionamento Global (GUARD, 1996).	9
2.2	Circuito integrado NEO-7M.	11
2.3	Circuito integrado SIM800.	12
2.4	Circuito integrado LM7805.	14
2.5	Circuito integrado LM2596T.	14
2.6	Circuito recomendado para o regulador LM2596T.	15
2.7	Módulo regulador de tensão, baseado no LM2596T.	15
2.8	Microcontrolador PIC16F84A.	16
3.1	Diagrama geral do sistema.	20
3.2	Diagrama de estados do cliente através de um celular via SMS.	21
3.3	Diagrama de estados do MCU do dispositivo embarcado ao automóvel.	22
3.4	Diagrama de estados do MCU do dispositivo de monitoramento.	22
3.5	Diagrama de estados do programa no computador.	23
3.6	Fluxograma para o MCU do dispositivo embarcado ao automóvel.	24
3.7	Fluxograma para o programa no computador.	25
3.8	Fluxograma para o dispositivo de monitoramento.	26
3.9	Módulo GSM escolhido, SIM800L V2 5V.	29
3.10	Módulo receptor GPS escolhido, GY-GPS6MV2.	29
3.11	Circuito de alimentação para o dispositivo embarcado.	30
3.12	Fonte de alimentação de 5V e 3A, com conector USB.	31
3.13	<i>Pull-down</i> para conectar os sensores do veículo no dispositivo embarcado.	31
3.14	Circuito do dispositivo embarcado para acionar os atuadores no automóvel.	32
3.15	Dispositivo de Monitoramento montado em uma <i>protoboard</i>	33
3.16	Ferramenta do MikroC para se gerar o programa descritor.	34
3.17	Dispositivo de Monitoramento sendo reconhecido pelo PC após conexão USB.	35
3.18	Filtro <i>anti-spam</i> da operadora Vivo bloqueando o envio de SMSs.	36
3.19	Pintura da placa de fenolite com tinta fotossensível.	37
3.20	Transferência do <i>Layout</i> para a placa pelo método fotográfico.	37
3.21	Remoção da tinta não sensibilizada pela luz ultravioleta.	37
3.22	<i>Layout</i> transferido para a placa com sucesso.	38
3.23	Corrosão da placa de fenolite.	38
3.24	Placa de circuito impresso finalizada.	38
3.25	Dispositivo Embarcado na <i>protoboard</i> , utilizando o PIC16F648A.	40
3.26	Dispositivo Embarcado na <i>protoboard</i> , utilizando o PIC18F4550.	40

4.1	<i>Design</i> do dispositivo embarcado simulado no Proteus.	43
4.2	<i>Design</i> do dispositivo de monitoramento simulado no Proteus.	43
4.3	Dispositivo embarcado prototipado.	44
4.4	Dispositivo de monitoramento prototipado.	44
4.5	Enviando comandos ao dispositivo embarcado e recebendo os retornos. . .	45
4.6	Programa indicando que o dispositivo de monitoramento não foi conectado.	46
4.7	Dispositivo de monitoramento foi conectado, mas não atualizou a senha. . .	47
4.8	Programa permitindo tentativa de <i>login</i>	47
4.9	PIN ou Senha incorretos, não efetuando o <i>login</i>	48
4.10	<i>Login</i> efetuado, tendo-se início o rastreamento.	48
4.11	Dispositivos desenvolvidos comportados por uma estrutura de MDF. . . .	49
4.12	Dispositivos desenvolvidos comportados por uma estrutura de MDF. . . .	49
A.1	Tabela de análise de riscos FMEA.	53
A.2	Esquemático do Dispositivo Embarcado ao Automóvel no Projeto Lógico. .	54
A.3	Esquemático do Dispositivo de Monitoramento no Projeto Lógico.	55
A.4	Esquemático do Dispositivo Embarcado ao Automóvel alterado após o Projeto Físico.	56
A.5	Esquemático do Dispositivo de Monitoramento alterado após o Projeto Físico.	57

Lista de Tabelas

2.1	Sentença “\$GPGLL,5221.34,N,12144.32,W,113444,A,*1D”.	10
2.2	Comandos AT para módulo SIM800.	13
3.1	Comparativo das especificações de microcontroladores PIC.	28
3.2	SMS recebido: “123455221.145322134.47291”.	36

Lista de Acrônimos e Notação

GPS	Global Positioning System
3G	Terceira geração de padrões e tecnologias da telefonia móvel
GSM	Global System for Mobile Communications
SMS	Short Message Service
SIM	Subscriber Identity Module
USB	Universal Serial Bus
MCU	Microcontrolador
UART	Universal Asynchronous Receiver-Transmitter
TTL	Transistor-Transistor Logic
NMEA	National Marine Electronics Association
ASCII	American Standard Code for Information Interchange
CI	Circuito integrado
SMD	Surface Mount Device
PWM	Pulse Width Modulation
EEPROM	Electrically-Erasable Programmable Read-Only Memory
RAM	Random Access Memory
PC	Personal computer
FMEA	Failure Mode and Effect Analysis
PCB	Printed Circuit Board

Introdução

Segundo o jornal FOLHA (2017), o Brasil possui um roubo de veículo a cada minuto. Diante dessa conjuntura, cresce drasticamente a utilização de sistemas de rastreamento automotivo e dispositivos antifurto. Os sistemas de rastreamento normalmente são utilizados por seguradoras de veículos. As mesmas possuem um servidor próprio, ou alugam servidores dedicados, estabelecendo uma comunicação entre uma central com os automóveis segurados, recuperando a localização de cada veículo. Os dispositivos antifurto existentes, em sua maioria, funcionam cortando a injeção de combustível do carro, a menos que o motorista aperte um botão que fica escondido.

Estas tecnologias, embora amplamente difundidas e utilizadas, possuem algumas desvantagens. A começar pelos sistemas de rastreamento; como carecem de uma comunicação à distância, praticamente todos estes utilizam uma comunicação cliente/servidor onde no veículo se encontra um dispositivo com um GPS e com acesso à Internet pelo padrão 3G; o qual se comunica com o servidor, enviando a sua localização. Uma vez necessário a existência de tal servidor, o usuário final fica condicionado à poder rastrear seu veículo pelo serviço de uma seguradora ou, arcando com os custos de comprar um servidor, ou ainda alugando um servidor dedicado. Contudo, como a maioria dos motoristas não possui formação na área de computação, utilizar um servidor próprio para rastreamento se torna uma tarefa demasiadamente complexa. Uma vez que o usuário esteja restrito ao serviço de monitoramento de uma seguradora, além do custo adicional gerado, surge um problema de confidencialidade da localização do automóvel. Já que esta informação não permanece restrita ao dono do veículo, mas fica disponível para um conjunto de funcionários da empresa seguradora. Sendo estas portanto as desvantagens dos atuais sistemas de rastreamento: a dificuldade do usuário se desvencilhar de uma empresa prestadora de serviços, o que aumenta o custo da tecnologia e compromete a confidencialidade.

Os dispositivos antifurtos existentes apresentam a desvantagem de que o conhecimento da tecnologia é de fácil acesso e aprendizado; assim, um bandido que saiba como desins-

talar o aparelho irá contornar a medida de segurança.

Com a difusão da tecnologia GSM (*Global System for Mobile Communications*), que é um padrão de telefonia celular de segunda geração, hoje a grande maioria dos celulares, *smartphones* ou não, possuem a mesma (CAVALCANTI *et al.*, 2018). Este padrão permite aos usuários a transmissão de sinais digitais de áudio, a troca de mensagens de texto, dentre outras aplicações. Atualmente, todos os modelos de celulares, que usam a tecnologia, permitem a realização de chamadas e o envio de mensagens SMS. O padrão exige a utilização de um cartão SIM (*Subscriber Identity Module*), o qual identifica o cadastro do assinante e permite a cobrança pelos serviços utilizados (CAVALCANTI *et al.*, 2018). A tecnologia GPS (*Global Positioning System*) é um sistema que permite um aparelho receptor móvel consultar sua posição por satélite (EL-RABBANY, 2002); a consulta de posição é gratuita e não exige nenhuma identificação do usuário - como o cartão SIM no padrão GSM, por exemplo - contudo, para que um aparelho móvel consulte e envie sua localização a um outro dispositivo pelo sistema GPS, há um alto custo; de forma que normalmente as consultas de posição são feitas pelo GPS, e, o envio desta localização por outros padrões.

A motivação para este projeto se baseia em, utilizando as tecnologias descritas, conciliar em um único sistema a rastreabilidade de um veículo e atuação sobre o mesmo; de forma que se possa oferecer este serviço ao usuário final sem que o mesmo tenha que contar com um prestador de serviços ou dispendiosos e recorrentes gastos. Assim, propõe-se desenvolver um dispositivo integrado ao veículo do usuário, o qual irá recuperar sua localização global através de um módulo GPS. O mesmo contará com módulo GSM e um microcontrolador para realizar a comunicação e atuação à distância com o cliente. De forma que este possa requisitar informações e enviar comandos ao dispositivo de qualquer celular via SMS, bastando apenas ter memorizado uma senha pré-estabelecida. O usuário será capaz de trancar ou destrancar o automóvel, abrir ou fechar janelas, desligá-lo, requisitar a localização e estado lógico dos sensores; tudo isto à distância. Também será capaz, caso queira, de monitorar seu veículo por seu computador pessoal, vendo-o em um mapa. Sendo que para isto, deseja-se desenvolver um receptor GSM com conexão USB e um programa para ser instalado no computador do cliente.

Pode-se questionar a utilização do GSM, um padrão de segunda geração, em detrimento da existência de padrões mais novos e bem desenvolvidos como o 3G e o 4G. Contudo, caso a comunicação do usuário pelo celular com o automóvel fosse feita via internet móvel, seria necessário se desenvolver um aplicativo para o celular. Considerando a hipótese de que o indivíduo tenha seu celular roubado juntamente com seu veículo; o mesmo estaria incapacitado de utilizar o sistema. Entretanto, uma vez que a comunicação é feita via SMS, não sendo necessário baixar nenhum aplicativo, na mesma conjectura,

o indivíduo poderia pedir emprestado um celular à primeira pessoa que encontrasse e, conseguiria utilizar o sistema proposto.

1.1 Definição do Problema

Como descrito, as atuais tecnologias antifurto automotivas vêm se tornando cada vez mais ineficazes quanto ao que se propõem. Simultaneamente, os sistemas de rastreamento, em sua maioria, utilizam-se de servidores de Internet para permitir a comunicação entre o veículo e o usuário. Este fator, além de acarretar um custo adicional, normalmente implica que os sistemas de rastreamento serão disponibilizados por uma empresa terceirizada, como argumentado anteriormente. O que além de aumentar ainda mais o custo, compromete em parte a confidencialidade de informações que deveriam ser restritas ao usuário.

1.2 Motivação

O trabalho em questão, além de abordar conhecimentos úteis e pertinentes à engenharia, apresenta também um grande potencial mercadológico. Uma vez que se confere tantas faltas nos dispositivos antifurto e de rastreamento como problematizado anteriormente.

1.3 Objetivos do Trabalho

Nestes termos, são objetivos deste projeto:

1.3.1 Objetivo Geral

Desenvolver um sistema de monitoramento e atuação em automóveis, capaz de fornecer ao celular do usuário, via GSM, informações sobre o veículo - como localização, nível lógico dos sensores nas portas e janelas - e, que permita ao cliente atuar à distância no veículo - trancar portas, fechar ou abrir janelas, e, desligar o carro. Desenvolver ainda um programa para que o usuário possa monitorar seu veículo em um computador pessoal, vendo a localização do automóvel sobreposta a um mapa global.

1.3.2 Objetivos Específicos

- Projetar e construir o circuito eletrônico que integre o microcontrolador com os módulos GSM e GPS.
- Projetar e construir o circuito bloqueador de corrente da injeção eletrônica do carro.
- Projetar e construir o circuito para trancar, destrancar e monitorar as portas.

- Projetar e construir o circuito para abrir, fechar e monitorar vidros elétricos.
- Projetar e construir circuito receptor de dados GSM que se conecte a um computador por uma porta USB.
- Projetar uma estrutura para comportar os circuitos eletrônicos.
- Desenvolver o programa do microcontrolador, estabelecendo uma comunicação e atuação via GSM do usuário no sistema.
- Desenvolver o programa do computador que represente a localização do automóvel sobreposta a um mapa.
- Obter sistema funcional em um carro.

1.4 Organização do Documento

Este trabalho foi organizado em 5 capítulos. No presente capítulo apresenta-se uma introdução, a definição do problema, a motivação do projeto, os objetivos e a organização do documento.

No segundo capítulo encontram-se a revisão de literatura, o estado da arte, e, os fundamentos teóricos para a realização do trabalho.

No terceiro capítulo foi tratado o desenvolvimento do trabalho.

No quarto capítulo foram apresentados os resultados obtidos com o trabalho.

Por fim, no quinto capítulo foram apresentadas as considerações finais e conclusão.

Fundamentos

Seguem-se dispostos neste capítulo os tópicos: revisão de literatura, estado da arte, e, fundamentação teórica.

2.1 Revisão de Literatura

Com o avanço da eletrônica foram desenvolvidos sistemas de navegação cada vez mais precisos. Os primeiros sistemas se baseavam em ondas de rádio, como o LORAN (*Long-Range Navigation System*) e o DECCA (*Low Frequency Continuous Wave Phase Comparison Navigation*) (MONICO, 2000). Os mesmos contudo, funcionavam somente na faixa costeira, onde se comunicavam com uma rede de estações para suporte de posicionamento.

Em 1957 foi lançado o primeiro satélite pela União Soviética, o *Sputnik 1*. Imediatamente um grupo do MIT (*Massachusetts Institute of Technology*) começou a estudar a relação do Efeito *Doppler* nas transmissões de rádio de tal satélite. Nesse momento começaram a surgir os primeiros sistemas de navegação baseados em satélites artificiais, como o NNSS (*Navy Navigation Satellite System*) (MONICO, 2000).

As tecnologias de transmissão da época eram contudo limitadas ao sinal analógico. Sabendo-se que na década de 70 surgiram os primeiros microprocessadores e microcontroladores comerciais (SILVA, 2006), tem-se esta data como marco do início do desenvolvimento de sistemas, tecnologias e padrões digitais eletrônicos.

Na década de 70, o projeto NAVSTAR (*Navigation System with Timing and Ranging*), iniciado na década de 60, toma maiores proporções. Com o auxílio da empresa *Rockwell*, que construiu 24 satélites para navegação, o sistema GPS (*Global Positioning System*) tem origem (MONICO, 2000). Porém, só se tornou completamente funcional na década de 90.

Os sistemas de telefonia móvel começaram a surgir a partir da década de 80 (SVERZUT, 2005). A primeira geração de telefonia móvel, AMPS (*Advanced Mobile Phone Service*), foi desenvolvida pelos Laboratórios Bell da AT&T, só permitindo a transmissão

de voz. Não tardou muito para surgirem as tecnologias de segunda geração, sendo que estas trabalham com sinais digitais, podendo transmitir sinais de áudio codificados e também informações de texto, como o SMS (REDL; WEBER; OLIPHANT, 1995). O padrão mais conhecido e popularizado da segunda geração foi o GSM, tendo surgido também na década de 80 (REDL; WEBER; OLIPHANT, 1995).

Os padrões de segunda geração foram fundamentais para o desenvolvimento dos atuais sistemas de rastreamento, já que como discutido anteriormente, oferecem uma comunicação de dados digitais à distância por um baixo custo.

2.2 Estado da Arte

Nas últimas décadas inúmeros trabalhos sobre rastreamento automotivo e dispositivos antifurto foram realizados.

No artigo de MANGLA *et al.* (2017) é possível constatar o desenvolvimento de um sistema de rastreamento automotivo, utilizando módulos GPS e GSM, e, um *smartphone* para que o usuário possa monitorar o sistema. O autor desenvolve um programa para um microcontrolador e um aplicativo para o *smartphone*, estabelecendo uma comunicação *full-duplex* entre o celular e o dispositivo instalado no automóvel. Embora a parte de códigos seja muito enriquecedora, apresentando diversas informações instrucionais quanto ao uso de alguns módulo; o *hardware* desenvolvido é superficial e não segue nenhuma metodologia de projeto. O autor simplesmente monta seu dispositivo conectando os módulos utilizados a um arduino.

No trabalho de R.RAMANI *et al.* (2013) é discutido não somente um sistema de rastreamento, mas também um sistema de travamento automotivo. Neste artigo foi desenvolvido um sistema onde um dispositivo instalado em um automóvel se comunica com um celular via SMS. O mesmo aborda com profundidade a questão do projeto físico do dispositivo, apresentando um circuito eletrônico de todo o *hardware* desenvolvido e simulando no *software Proteus*. Contudo, o artigo em questão não implementa seu projeto físico, permanecendo apenas na simulação.

No artigo de MUKHTAR (2015) é desenvolvido um sistema de rastreamento e de controle sobre um automóvel. O autor também utiliza as tecnologias GPS e GSM, criando um dispositivo que é instalado no veículo e pode se comunicar com um celular, ou, computador através de uma página *Web*. São abordados com profundidade tanto o *hardware* quanto o *software* projetados. Nos resultados obtidos, o autor realiza a montagem do dispositivo físico bem como os devidos testes no mesmo.

2.3 Fundamentação Teórica

Na presente seção são explanados conceitos e tecnologias de potencial uso no trabalho; sendo que os mesmos foram organizados na forma de tópicos. A compreensão dos temas que se seguem são necessários para se entender o fundamento das escolhas tomadas no desenvolvimento do trabalho, ainda que nem todas as tecnologias a seguir tenham por fim sido utilizadas no trabalho.

2.3.1 Protocolos de Comunicação

Um protocolo de comunicação é um padrão que define o formato e a ordem para a transmissão e recepção de mensagens entre duas ou mais máquinas. Define também ações realizadas pelas máquinas a partir de tais mensagens (KUROSE; ROSS, 2010).

Deve-se destacar que a comunicação entre dispositivos microprocessados pode ser: unidirecional, ou *simplex*, permitindo o fluxo de informação em apenas um único sentido; *half-duplex*, possibilitando a troca de informações em dois sentidos, contudo utilizando um mesmo barramento, devendo se intercalar comunicações em sentidos contrários; e, *full-duplex*, que permite a troca de informações cruzadas simultaneamente, tendo-se barramentos diferentes para a transmissão e recepção em cada dispositivo comunicante (TANENBAUM, 2002).

2.3.2 UART

O UART (*Universal Asynchronous Receiver/Transmitter*) é um padrão para comunicação serial muito utilizado em computadores e microcontroladores, implementado a nível de *hardware*. O mesmo especifica regras para o envio de um pacote de dados; definindo-se que um pacote deve iniciar com um *bit* de *start*, seguido do *byte* contendo a informação de interesse, um *bit* de paridade, e, um ou dois *bits* de parada (SILVA, 2006). Sempre que na documentação de um dispositivo for dito que o mesmo possui um UART, então entende-se que existem pinos específicos para esta comunicação e que no *hardware* do componente existe um circuito lógico que opera conforme as normas citadas.

2.3.3 Padrão TTL

O acrônimo TTL significa *Transistor-Transistor Logic*, ou, Lógica Transistor-Transistor. Esta designação passou a ser utilizada para se denominar os níveis lógicos de trabalho dos dispositivos, mesmo quando não se trata diretamente de circuitos integrados TTL. Quando se diz que um elemento opera em níveis TTL, entende-se que o binário 0 (falso) é o sinal de tensão em 0V, e, o binário 1 (verdadeiro) é o sinal de tensão de 5V (SILVA, 2006).

2.3.4 Comunicação USB

A interface USB (*Universal Serial Bus*) é um padrão de comunicação muito comum em computadores. Foi desenvolvido na década de 90, sofrendo várias revisões, sendo a mais recente a USB 3.0 - comercializada desde 2010. A taxa de transmissão máxima é de 5000Mbits/s.

A transferência de dados nesta interface ocorre através do envio de vetores, ou pilhas, de *bytes*. O dispositivo que se comunica a um respectivo computador, podendo ser até mesmo outro computador, deve portanto conter um programa denominado “Descriptor”. Este programa contém diversas informações, como a dimensão dos vetores de *bytes* que serão enviados. Dentre outras informações importantes, vale-se destacar também o *Vendor ID* e o *Product ID*, que são sequências de caracteres armazenadas no descriptor e utilizadas para que o dispositivo comunicante possa ser identificado independentemente de em qual porta COM tenha sido conectado.

2.3.5 Sistema de Posicionamento Global (GPS)

O Sistema de Posicionamento Global (GPS) é um sistema desenvolvido e mantido pelo Departamento de Defesa do Governo dos Estados Unidos da América (EUA). O mesmo se baseia na navegação por satélite para permitir que usuários na crosta terrestre e atmosfera determinem sua posição tridimensional, velocidade e horário. Sendo possível requisitar tais informações em qualquer horário de qualquer lugar do mundo (GUARD, 1996).

O GPS possui três grandes segmentos: O segmento espacial, o segmento de controle, e, o segmento de usuários, conforme pode ser visto na figura 2.1. O segmento espacial consiste em 24 satélites que orbitam ao redor da terra de tal forma que pelo menos 4 podem ser vistos de qualquer lugar no mundo. Tais satélites transmitem ininterruptamente mensagens contendo várias informações, como por exemplo dados de tempo e suas órbitas. O segmento de controle consiste em estações de controle e monitoramento dos satélites, atualizando as mensagens que os mesmos transmitem e ajustando os relógios destes. Por fim, o segmento de usuários, e o de mais importante compreensão para este trabalho, consiste em receptores dos dados transmitidos pelos satélites. Tais receptores, com as informações provenientes dos satélites e através de um processo denominado trilateração, são capazes de obter informações sobre si próprios como a posição e a velocidade (GUARD, 1996). Estes receptores são dispositivos microprocessados que se comunicam fisicamente com outros dispositivos, normalmente via comunicação serial, através do padrão NMEA 0183 ou de padrões similares.

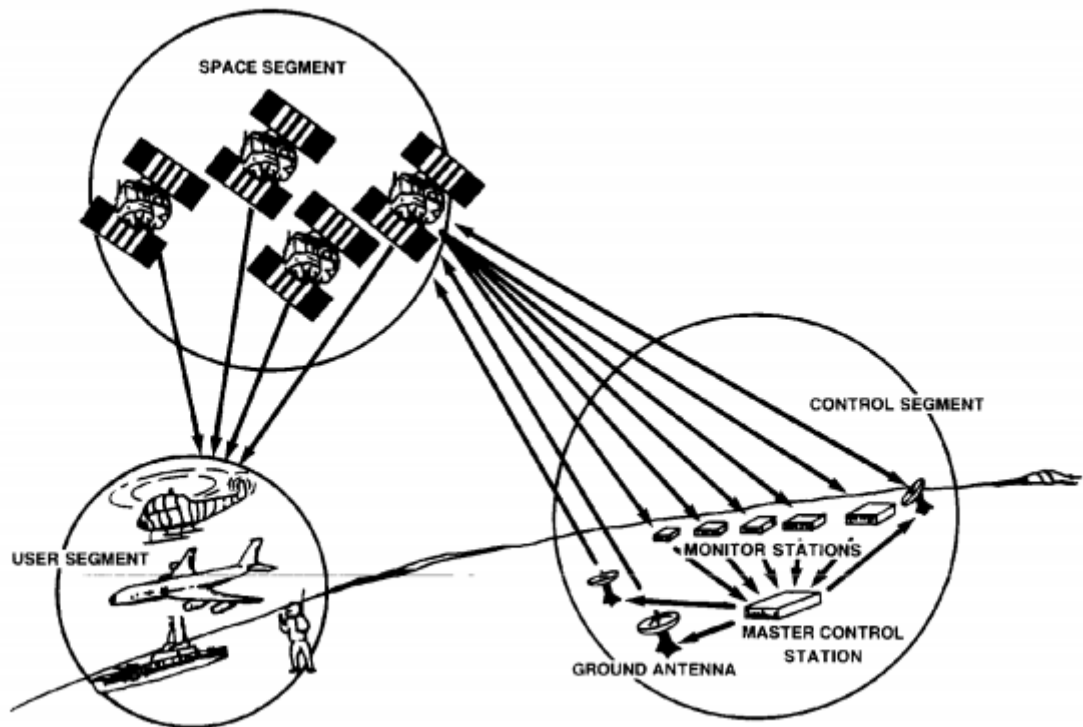


Figura 2.1: Segmentos do Sistema de Posicionamento Global (GUARD, 1996).

2.3.6 Padrão NMEA

O padrão NMEA 0183 (*National Marine Electronics Association 0183*) contém especificações dos níveis elétricos, *baud rate*, paridade e o formato dos dados a serem transmitidos por módulos receptores GPS (HOFMANN-WELLENHOF; LICHTENEGGER; WASLE, 2007). Contudo, não é raro se encontrar dispositivos que não sigam todo o protocolo. Por exemplo, normalmente os fabricantes diferem do que é definido pelo NMEA para os níveis de tensão dos terminais de comunicação, mas adotam a mesma lógica de envio e recepção de dados. Assim, focaremos no que os receptores GPS geralmente adotam do padrão NMEA, que consiste basicamente no envio de dados, através do código ASCII. Estes dados são um conjunto de caracteres que denominamos sentenças.

Uma sentença consiste em um conjunto de caracteres que contém as informações obtidas pelo receptor GPS através do processo de trilateração. Por definição do padrão, toda sentença se inicia com o caractere “\$”; os próximos caracteres identificam a fonte da sentença, por exemplo “GP” indica que o módulo receptor GPS é a fonte da sentença em questão. O padrão define que os próximos 3 caracteres irão indicar o tipo de informação que a sentença irá transmitir; os próximos caracteres contém a informação de interesse e são dispostos em conjuntos de campos delimitados por vírgulas. O último campo é

preenchido com o caractere “*”, seguido por um caractere numérico hexadecimal de dois dígitos utilizado para se verificar a integridade da informação enviada através de uma soma (*checksum*) (EL-RABBANY, 2002).

Um exemplo de uma sentença seria: “\$GPGLL,5221.34,N,12144.32,W,113444,A,*1D”; segue-se na tabela 2.1 a explicação de cada termo.

Tabela 2.1: Sentença “\$GPGLL,5221.34,N,12144.32,W,113444,A,*1D”.

\$	Sentença é iniciada com o “\$”
GP	Indica que a fonte da sentença é o módulo GPS
GLL,	Indica o tipo de sentença
5221.34,N,	Indica uma latitude de 52 graus e 21.34 minutos
12144.32,W,	Indica uma longitude de 121 graus e 44.32 minutos
113444,	Indica o horário de 11h34min44s
A,	Indica que a informação é válida
*1D	Hexadecimal da soma de verificação

Existem várias sentenças GP distintas que o módulo receptor GPS envia constantemente aos dispositivos conectados a ele, como por exemplo o GGA, GLL, GSA, dentre outras. É importante se destacar que as siglas das sentenças nem sempre são acrônimos de seus significados. GGA significa “*System Fix Data*”, GLL “*Geographic Latitude and Longitude*”, e, GSA significa “*Satellite Status*”.

Este trabalho foca apenas a sentença GPGLL, pois como exemplificado na tabela 2.1, esta sentença fornece a informação da latitude e longitude que é o suficiente para a realização do projeto.

A sentença GPGLL conforme a tabela 2.1, possui em seus primeiros 4 campos a informação da longitude e latitude, seguida de um campo com o horário em que tal posição foi obtida. O próximo campo indica se a informação é válida pois, para o processo de trilateração ocorrer corretamente é necessário se captar o sinal de 4 satélites distintos. Caso o receptor obtenha o sinal de um número inferior, o campo será preenchido com o caractere “V” indicando que a informação não é confiável. O último campo contém o número hexadecimal para se realizar a soma de verificação; este processo consiste em se somar o valor binário da representação ASCII de cada caractere entre o “\$” e o “*” da sentença, para que possa se verificar a integridade da mesma (EL-RABBANY, 2002).

É importante se ressaltar a existência de várias representações de coordenadas. A sentença GPGLL nos dá a representação em um formato conhecido como geodésico, em graus e minutos. Outras representações comuns são o geodésico decimal, representado pela sigla DD (Decimal Degrees), e, o UTM (Universal Transversa de Mercator), também podendo ser representado pela sigla DMS (Degrees Minutes Seconds). A conversão destes

formados é simples, basta saber que 1 grau equivale a 60 minutos, e, 1 minuto equivale a 60 segundos. Assim, para se converter de DMS para geodésico, por exemplo, basta dividir os segundos do DMS por 60 e adicioná-los aos minutos da coordenada geodésica.

2.3.7 Módulo Receptor GPS

Os módulos receptores GPS existentes no mercado consistem em sistemas microprocessados que realizam todo o processo de trilateração. Se comunicam com um segundo dispositivo enviando as informações, geradas pela trilateração, através do protocolo NMEA, ou ainda um protocolo derivado ou similar deste.



Figura 2.2: Circuito integrado NEO-7M.

Fonte: <<https://emlid.com.br/navio-ublox-tutorial-cpp/>>. Acesso em: 14/04/2019.

Quanto à comunicação com o segundo dispositivo mencionado, a grande maioria dos módulos possuem uma comunicação serial com níveis TTL e protocolo UART na camada física. Contudo, como mencionado anteriormente, os fabricantes não seguem rigorosamente um padrão para esta camada. Existindo dispositivos que trabalham com níveis de tensão RS-232 ao invés do TTL, por exemplo.

No mercado existem tais módulos com tamanho bastante reduzido, todos com configurações muito similares e preços bastante acessíveis, como o SkyNav SKM53 e o GY-NEO6MV2. Também existem receptores GPS com especificações como sensibilidade e tempo de aquisição com valores muito reduzidos. Normalmente sendo dispositivos de preços extremamente altos (em torno de milhares de reais no mercado nacional), como o GPS Furuno Gp-39, sendo utilizados em embarcações e aplicações similares.

Quanto aos primeiros tipos de módulos, a grande maioria necessita de uma alimentação de 3,3V, não requisitando uma corrente muito elevada, normalmente inferior a 700mA. No caso do módulo GPS NEO-6M, por exemplo, o consumo varia entre 10mA a 100mA.

Um exemplo de um módulo receptor GPS de baixo custo é o NEO-7M da fabricante U-BLOX, conforme pode ser visto na figura 2.2.

2.3.8 Módulo GSM

Similares aos receptores GPS, os módulos GSM também são dispositivos microprocessados que se comunicam fisicamente com um segundo dispositivo, normalmente via comunicação serial e se comunicam ainda à distância com outros módulos GSM. A maioria destes módulos possuem terminais para a comunicação serial, entradas e saídas de áudio, terminal para *reset* e também para alimentação.

É importante enfatizar a alimentação destes módulos. Mediante uma pesquisa das especificações de vários modelos disponíveis, pôde-se ver que todos exigem uma elevada corrente de operação - podendo alcançar picos de até 2A.



Figura 2.3: Circuito integrado SIM800.

Fonte: <<https://simcom.ee/modules/gsm-gprs/sim800/>>. Acesso em: 14/04/2019.

Também se faz necessário ressaltar que estes dispositivos não funcionam sem um cartão SIM, como mencionado no capítulo 1.

A tecnologia GSM permite que estes módulos troquem entre si mensagens de texto, sinais digitais de áudio, dentre outros tipos de dados. Contudo, focaremos apenas em como são enviados SMS através de tais módulos.

Através dos pinos de comunicação serial, os módulos GSM recebem comandos e enviam dados. Os comandos que podem ser passados aos módulos normalmente são definidos pelo fabricante do mesmo. Por exemplo, no caso do SIM800, um CI SMD (*Surface Mount Device*) como disposto na figura 2.3, a SIMCOM, fabricante do componente, disponibiliza tais comandos em um documento (SIMCOM, 2016).

Estes comandos são enviados segundo o protocolo UART em níveis TTL. Os mesmos sempre começam com os caracteres “AT”, e, são necessários para se utilizar as funcionalidades do módulo; como por exemplo ler e enviar SMS. A tabela 2.2 apresenta alguns de seus comandos básicos.

Tabela 2.2: Comandos AT para módulo SIM800.

Comando recebido pelo módulo	Ação realizada pelo módulo
AT+IPR=9600	Configura o baud rate em 9600 bps
AT+CMGS="+5511974254932"	Envia uma mensagem para o SIM 5511974254932
ATD+5511974254932	Realiza uma ligação para o SIM 5511974254932
AT+CMGL=?	Retorna o número de SMSs na memória do módulo

2.3.9 Conversores DC-DC

Os conversores, ou reguladores, de corrente contínua para corrente contínua se dividem em dois grandes grupos: os lineares e os chaveados.

As fontes de alimentação lineares normalmente são de pouca complexidade, contudo, pouco eficientes. Isto ocorre pois os semicondutores operam como amplificadores. Desta forma, apresentam uma resistência considerável para a corrente direcionada à carga - dissipando potência e gerando muito calor.

Já as fontes chaveadas utilizam os semicondutores para realizar o chaveamento. Desta forma, a resistência do semicondutor é quase nula quando o mesmo permite a passagem de corrente para a carga, e praticamente infinita quando o mesmo bloqueia a corrente para a carga. Por esta razão, a potência dissipada nos semicondutores em fontes chaveadas é muito pequena, sendo então muito mais eficientes que as fontes lineares (AHMED, 2000).

Os conversores chaveados DC são classificados como *step-down* quando fornecem uma tensão de saída menor que a tensão de entrada, e *step-up* quando fornecem uma tensão de saída maior que a de entrada (AHMED, 2000).

2.3.10 CIs Reguladores de Tensão

Existem no mercado circuitos integrados que operam como conversores DC-DC. As vantagens de tais dispositivos são o baixo custo e a simplicidade de uso.

É possível se encontrar tanto CIs reguladores lineares quanto chaveados. Por exemplo, o linear LM7805, conforme a figura 2.4, que tendo uma entrada de 7,5V a 35V fornece uma saída constante de 5V. Há também CIs chaveados como o LM2596T, disposto na figura 2.5, que é um encapsulamento de um conversor *step-down*. A vantagem deste dispositivo é que o projetista pode utilizá-lo sem ter que desenvolver todo um circuito regulador chaveado.

O uso dos reguladores da família LM78XX é simples; tem-se um terminal para a tensão de entrada, um terminal para o GND, e, um para a tensão de saída.

O LM2596T já necessita de uma quantidade maior de componentes para funcionar. Contudo, como o próprio fabricante disponibiliza no *datasheet* do dispositivo um circuito

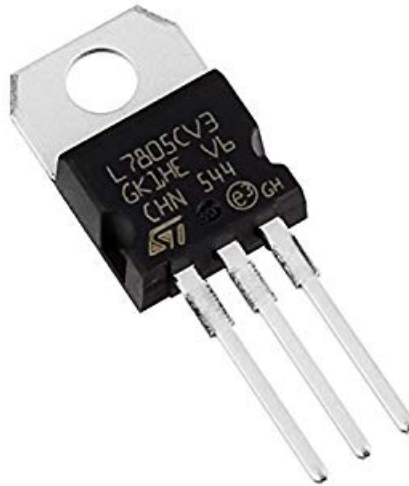


Figura 2.4: Circuito integrado LM7805.

Fonte: <<http://www.baudaeletronica.com.br/regulador-de-tensao-l7805.html>>. Acesso em: 12/04/2019.



Figura 2.5: Circuito integrado LM2596T.

Fonte: <<http://www.baudaeletronica.com.br/regulador-de-tensao-lm2596.html>>. Acesso em: 12/04/2019.

recomendado, sua utilização pode ser considerada simples. Segue na figura 2.6 o circuito de aplicação recomendado para o LM2596T.

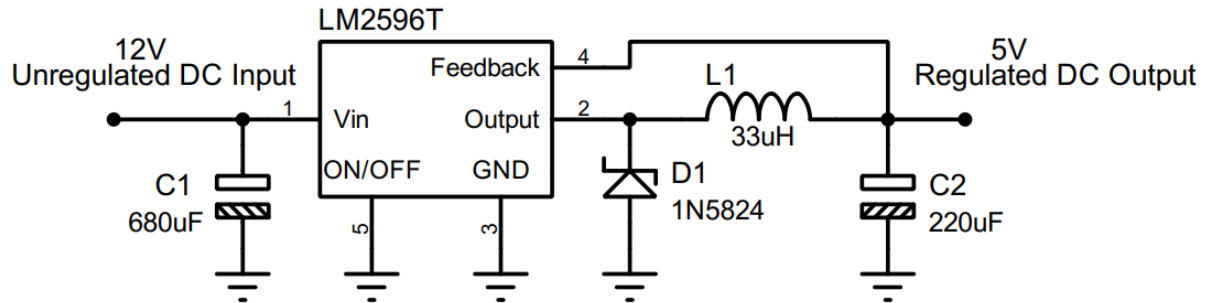


Figura 2.6: Circuito recomendado para o regulador LM2596T.

Na figura 2.7 pode-se ver um circuito baseado no LM2596T, que é vendido conforme à imagem e pronto para uso.



Figura 2.7: Módulo regulador de tensão, baseado no LM2596T.

Fonte: <<https://www.eletrogate.com/modulo-regulador-de-tensao-step-down-lm2596/>>.

Acesso em: 02/09/2019.

2.3.11 Microcontroladores

Um microcontrolador é um CI programável que contém internamente: uma unidade central de processamento; memória EEPROM ou FLASH para armazenar programas; memória RAM; portas de entrada e saída de dados; conversores analógico/digital, dentre outras funcionalidades que variam de acordo com o modelo de microcontrolador, por exemplo a existência ou não de UART (SILVA, 2006).

Em contrapartida aos microprocessadores, a grande vantagem de se usar microcontroladores é que necessitam de poucos circuitos externos para funcionar, normalmente apenas um cristal oscilador e uma fonte de alimentação.

A maioria dos microcontroladores podem ser programados em *Assembly* ou C. Para se carregar seu programa, pode-se utilizar um gravador ou, para alguns modelos que são

vendidos em uma placa, como o Arduino, pode-se utilizar um simples cabo USB. Segue na figura 2.8 um exemplo de um microcontrolador, o PIC16F84A.



Figura 2.8: Microcontrolador PIC16F84A.

Fonte: <<http://www.baudaeletronica.com.br/microcontrolador-pic16f84a.html>>. Acesso em: 04/05/2019.

O consumo de energia dos microcontroladores é baixo. Um PIC alimentado com 5V e trabalhando a 4MHz, considerando os terminais de alimentação, consome apenas 2mA, por exemplo (SILVA, 2006). Contudo, o gasto do MCU (*Micro Controller Unit*) aumenta conforme a quantidade de terminais de saída digital utilizados; normalmente cada porta consome cerca 10mA, não devendo se exceder 30mA. Assim, por exemplo um PIC funcionando com 18 terminais de saída digital consumiria apenas cerca de 180mA das portas lógicas, com mais 2mA de consumo fixo, totalizando cerca de 182mA. As portas de entrada digital praticamente não interferem no consumo do PIC, pois permanecem em um estado de alta impedância.

Muitos microcontroladores possuem temporizadores, que vêm a ser uma ferramenta de bastante interesse. Um temporizador, ou *timer*, é um contador que opera paralelamente ao processamento das linhas de código. Nele uma contagem é feita se incrementando *bit* por *bit*, na frequência do sinal de *clock* interno ou externo, tendo-se a possibilidade da divisão deste sinal (*prescaler*). A maioria dos *timers* são de 8 bits, assim é esperado que, realizando esta contagem, em um dado momento o contador tenha um *overflow*. Ou seja, quando o contador atinge a contagem posterior ao hexa FF no caso do *timer* de 8

bits. Assim, o *timer* reinicia sua contagem, retornando ao hexa 00 e uma interrupção é gerada. Neste momento, o código é desviado para a função de interrupção, na qual podem ser acrescentadas instruções pelo programador. As interrupções são de extrema valia nos processadores para que não se sobrecarregue o dispositivo com varreduras no código principal, além de permitir o desvio do código principal segundo uma contagem independente, gerando um efeito similar a um processamento paralelo. Além disto, existem outros tipos de interrupções, como a interrupção externa; muito útil para substituir a varredura de entradas no microcontrolador (SILVA, 2006).

Os microcontroladores possuem ainda uma série de configurações que podem ser definidas por linha de código, alterando-se os FUSES BITS. Nestas configurações é onde se determina o *clock*, se o terminal de *reset* estará habilitado ou não, dentre outras.

2.3.12 Linguagem de Programação JAVA

A linguagem Java surgiu na década de 90, desenvolvida pela empresa *Sun Microsystems*. É uma linguagem orientada a objetos, mas que se diferencia das demais principalmente pela portabilidade entre vários tipos de processadores e sistemas operacionais. Tal linguagem também é bastante amigável para o desenvolvimento de interfaces gráficas se comparada ao C++ e similares (SCHILDT, 2014).

2.3.13 Linguagem de Programação C#

A linguagem de programação C# surgiu no ano de 2002. A mesma, em termos de uso, é bastante similar ao Java. Sendo bastante amigável para o desenvolvimento de aplicações em geral. A IDE *Visual Studio*, da *Microsoft*, contempla esta linguagem.

2.3.14 Linguagem de Programação C

A linguagem de programação C é uma linguagem estruturada que surgiu na década de 70 na *Bell Labs*, por Dennis Ritchie. Desde então sofreu melhorias e muitas outras linguagens surgiram se baseando nesta, como o C++ e o Java. Apesar de a linguagem C estar caindo em desuso para o desenvolvimento de programas em computadores pessoais, a mesma ainda é muito utilizada para a programação de microcontroladores (SILVA, 2006).

Algumas características da linguagem C são: poucos comandos e operadores, recursos de baixo e alto nível, geração de códigos eficiente, confiabilidade, regularidade e portabilidade (SILVA, 2006).

2.3.15 Proteus

O *software* Proteus é um programa de simulação de circuito elétricos e eletrônicos. O mesmo se diferencia dos demais por dispor de inúmeros microcontroladores e outros sistemas embarcados disponíveis para simulações.

2.3.16 MikroC PRO for PIC

O MikroC PRO for PIC é um ambiente de desenvolvimento de programas específico para MCUs da fabricante *Microchip*. Tal IDE conta com diversas bibliotecas, facilitando o uso dos microcontroladores PIC.

Metodologia

O desenvolvimento deste trabalho ocorreu mediante três tipos de projeto, a saber: Projeto Comportamental, Projeto Lógico, e, Projeto Físico.

No projeto comportamental deseja-se tratar a parte abstrata do sistema; com o auxílio de diversos diagramas, descrever o funcionamento básico do projeto. Discute-se também fatores de risco, possíveis falhas e a prevenção para as mesmas.

No projeto lógico deseja-se iniciar o desenvolvimento dos circuitos eletrônicos, especificação de alguns componentes, e testes de *software*. Nesta etapa trabalha-se com simulações, e, se obtém um circuito eletrônico esquemático do produto.

No projeto físico, tendo-se o circuito esquemático, parte-se para o desenvolvimento das PCBs; seguindo-se com a prototipação do produto.

3.1 Projeto Comportamental

Para a execução do projeto comportamental, temos como especificações:

- Garantir ao usuário acesso de atuação e monitoramento veicular via celular.
- Garantir ao usuário acesso de monitoramento veicular via computador.
- Garantir ao usuário confiabilidade contra defeitos.

Foi elaborado um diagrama geral de funcionamento do sistema proposto, disposto na figura 3.1. Onde se apresentam as principais funcionalidades de *hardware* e o esquema de comunicação dos dois dispositivos a serem desenvolvidos: o “Dispositivo Embarcado ao Automóvel” e o receptor GSM com conexão USB, nomeado de “Dispositivo de Monitoramento”.

Pela figura 3.1, vê-se que o usuário pode ter acesso ao dispositivo embarcado através do celular, ou, através de um computador. Com um telefone, via SMS, o cliente tem uma comunicação do tipo *full-duplex*, onde é capaz de alterar os níveis lógicos dos atuadores e

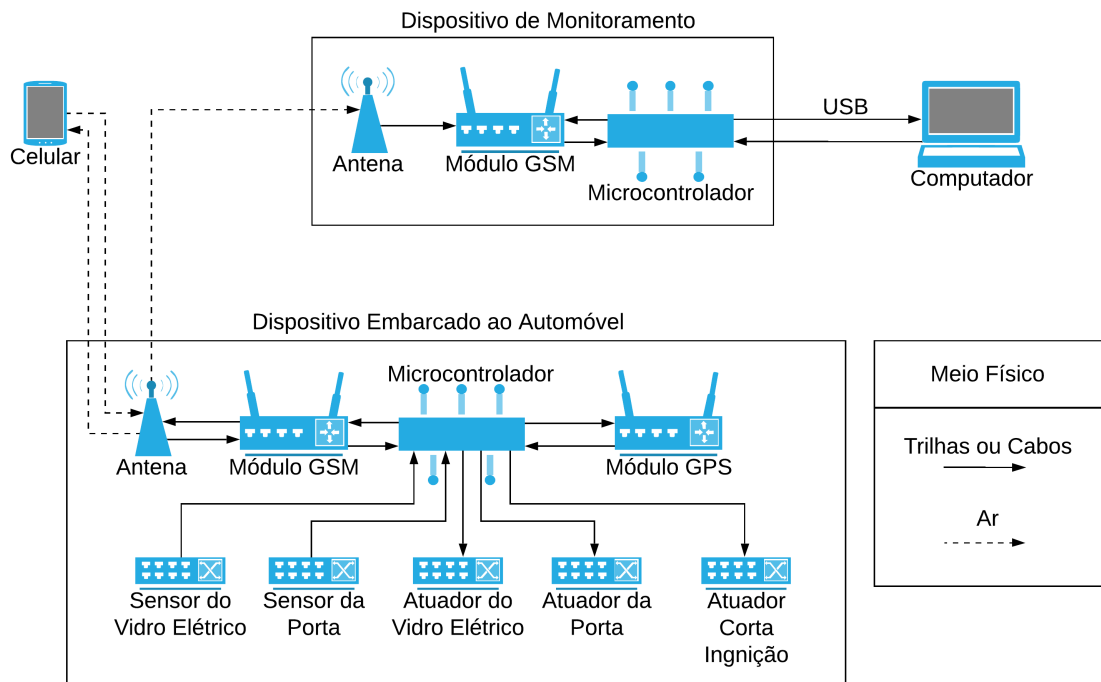


Figura 3.1: Diagrama geral do sistema.

receber informações sobre o automóvel, graças ao modem GSM no dispositivo embarcado. Os dados recebidos via SMS, contudo, são dispostos apenas no formato de texto.

Para se verificar a posição do veículo em um mapa global, o usuário deve utilizar um computador conectado ao dispositivo receptor GSM. Sendo que no computador em questão é contido o programa para monitoramento do automóvel. Pode-se observar na figura 3.1 que, neste caso, o cliente dispõe de uma comunicação unidirecional com o dispositivo embarcado ao veículo, já que o intuito é realizar-se apenas o monitoramento, e não atuação sobre o mesmo.

3.1.1 Diagramas de Estados

Na presente subseção são demonstrados os diagramas de estados do sistema. Este tipo de diagrama visa facilitar a observação dos possíveis estados que os dispositivos de *hardware*, o programa desenvolvido para o PC, e, o usuário no celular podem se encontrar. Em cada diagrama o objeto parte de um estado inicial e troca para outro estado conforme a existência de um evento. Os estados são representados pelas circunferências, enquanto os eventos são representados pelas setas curvas.

Segue-se na figura 3.2 a máquina de estados para o usuário quando o mesmo utiliza o sistema através de um celular via GSM. Definiu-se que em uma única mensagem SMS o usuário enviará uma senha e um comando, sendo que ao se desenvolver os programas do

projeto será normatizado um padrão para tais mensagens. Após enviar o SMS, o usuário entra em um estado em que aguarda um SMS do dispositivo embarcado ao veículo. Este retorno poderá ser uma mensagem com as informações do veículo atualizadas ou uma mensagem de erro. Uma possível falha levantada é que o usuário permaneça no estado de aguardar o SMS indefinidamente - este evento pode ocorrer caso um dos módulos tenha uma má cobertura de sinal ou problema semelhante.

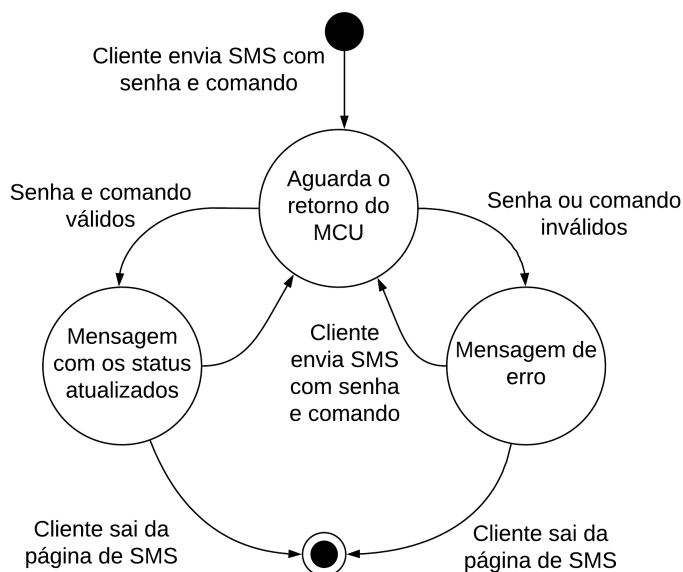


Figura 3.2: Diagrama de estados do cliente através de um celular via SMS.

Na figura 3.3 é mostrado o diagrama de estados do MCU do dispositivo embarcado. Percebe-se que o microcontrolador ao ser ligado, permanece simultaneamente em dois estados: um onde mantém a comunicação GSM com o dispositivo de monitoramento, enviando a localização para o mesmo; outro estado onde aguarda um SMS com um comando válido do usuário. Tendo o usuário enviado uma mensagem, o MCU poderá alterar de estado conforme a validação do SMS recebido.

Na figura 3.4 é disposto o diagrama de estados do microcontrolador do dispositivo de monitoramento. O mesmo inicia em um estado onde o MCU recebe do computador a senha e o *username* que o usuário digitou no programa do PC. Assim, o microcontrolador irá validar a informação passada pelo usuário com as informações armazenadas em sua memória EEPROM. A maneira como este MCU atualiza a senha na EEPROM é tratada mais especificamente na seção de diagramas de processos; basicamente, o microcontrolador recebe a senha atualizada do dispositivo embarcado via SMS. Uma vez que o MCU receba a senha correta pelo computador, é iniciado um estado onde se aguarda a localização ser requisitada.

Na figura 3.5 o diagrama de estados do programa no computador é representado. Pode-

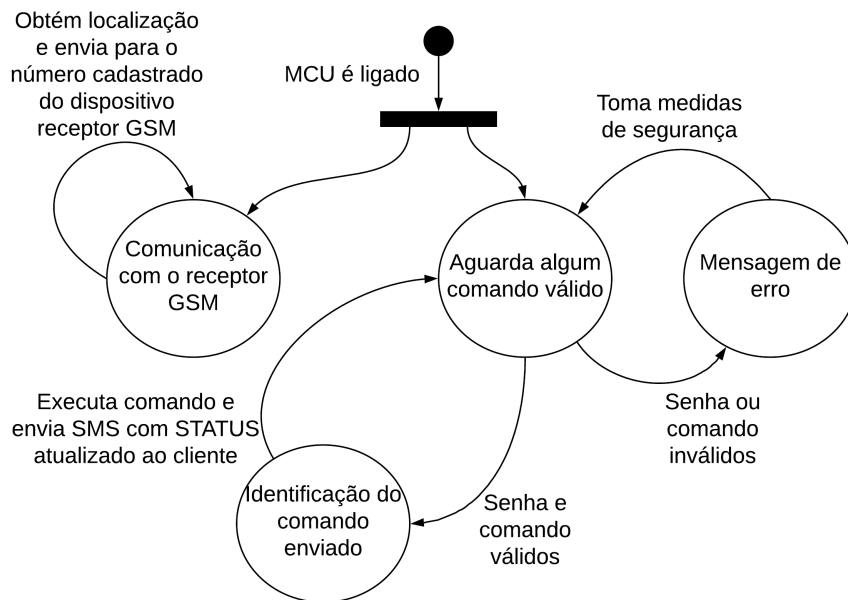


Figura 3.3: Diagrama de estados do MCU do dispositivo embarcado ao automóvel.

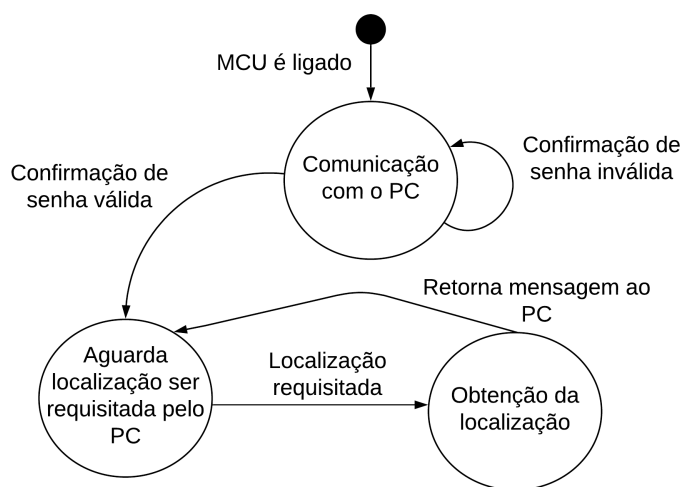


Figura 3.4: Diagrama de estados do MCU do dispositivo de monitoramento.

se observar que o programa possui dois estados, que também serão os *layouts* visíveis ao cliente, sendo uma página com a opção de *login*, e, um segundo estado onde o veículo é mapeado.

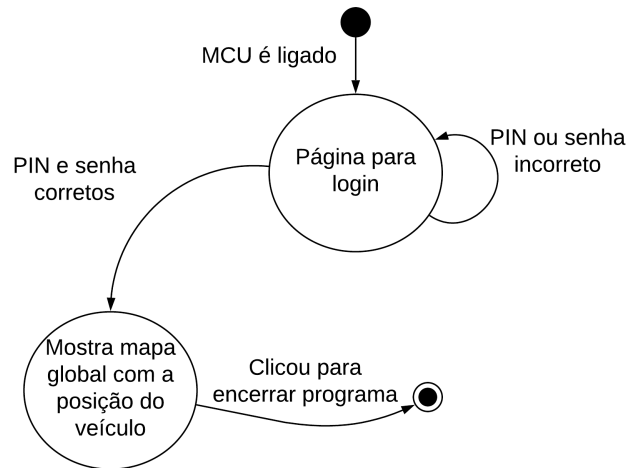


Figura 3.5: Diagrama de estados do programa no computador.

3.1.2 Diagramas de Processos

Os diagramas de processos são utilizados para se definir todas as ações que um dispositivo, ou programa, podem tomar. Seguem-se nesta subseção os diagramas de processos desenvolvidos para os dois MCUs dos dispositivos de *hardware*, e, para o programa do computador pessoal.

Na figura 3.6 segue-se o diagrama de processos do MCU do dispositivo embarcado ao automóvel. Pode-se observar que o microcontrolador, após ser ligado, configura os FUSES BITS. Como tratado na fundamentação teórica, estes *bits* definem algumas configurações do MCU, como a presença ou não de *watchdog*, a frequência de oscilação, dentre outras. Segue-se declarando uma variável denominada “tentativas”, e, inicializando-se uma comunicação serial com o módulo GSM. Têm-se então dois fluxos de processos que ocorrem paralelamente; mostrando que será preciso fazer uso de interrupções.

O MCU inicia um processo de *loop* infinito onde envia ao número cadastrado do dispositivo de monitoramento as coordenadas lidas pelo módulo GPS juntamente com a senha atualizada. A senha permanece salva na memória EEPROM do microcontrolador do dispositivo embarcado, para que o cliente não precise defini-la sempre que o mesmo for ligado. Além disso, o dispositivo irá conter uma senha padrão inicial, a qual poderá ser alterada via SMS.

Simultaneamente, o MCU entra em um *loop* para verificação de novas mensagens SMS. Neste ponto, a variável “tentativas” é utilizada para se implementar uma medida

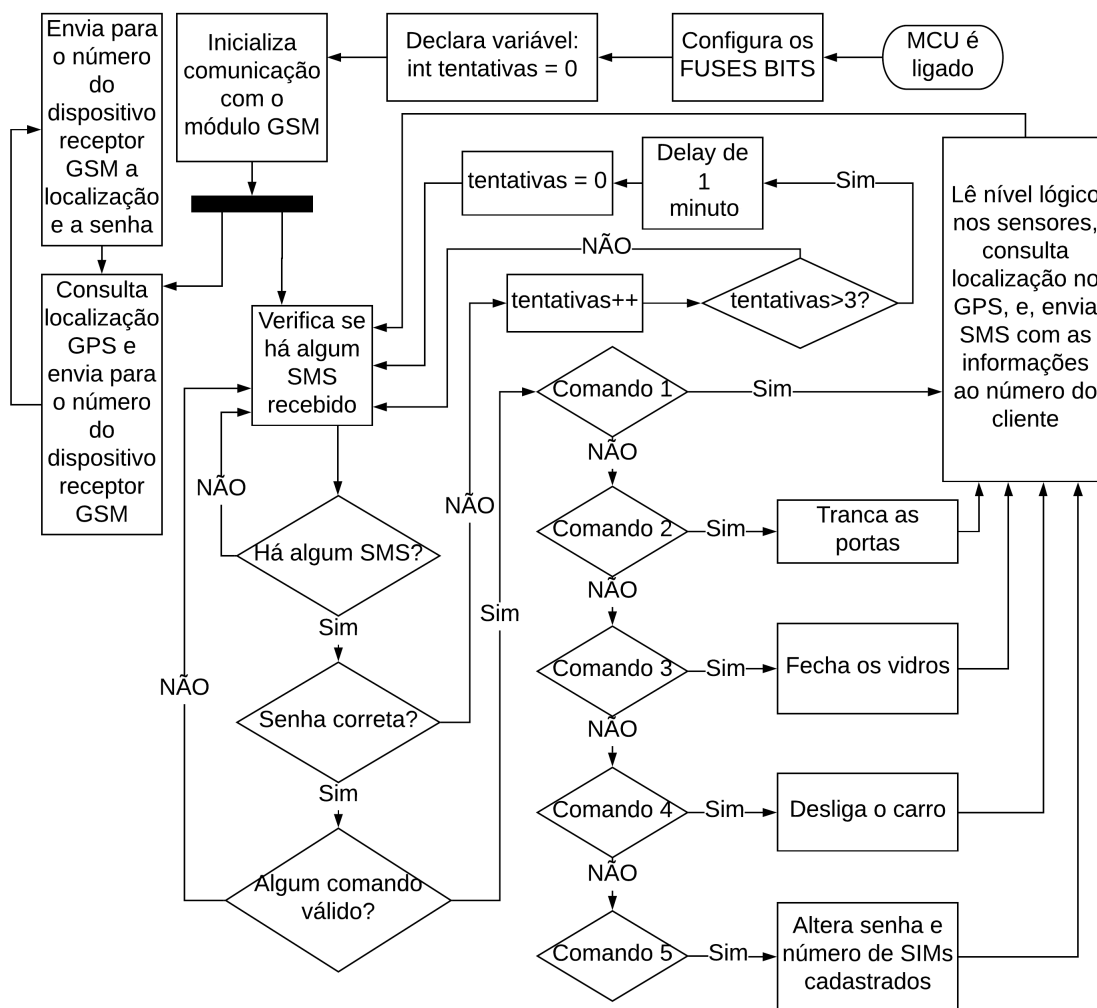


Figura 3.6: Fluxograma para o MCU do dispositivo embarcado ao automóvel.

de segurança, a qual impede que usuários não autorizados tentem descobrir a senha por programas de múltiplas tentativas de acesso.

Havendo uma mensagem com senha e comando válidos, o MCU iniciará um novo processo conforme o comando enviado. Independentemente de qual comando tenha sido enviado, o microcontrolador retornará uma mensagem com os status do automóvel ao número cadastrado do cliente.

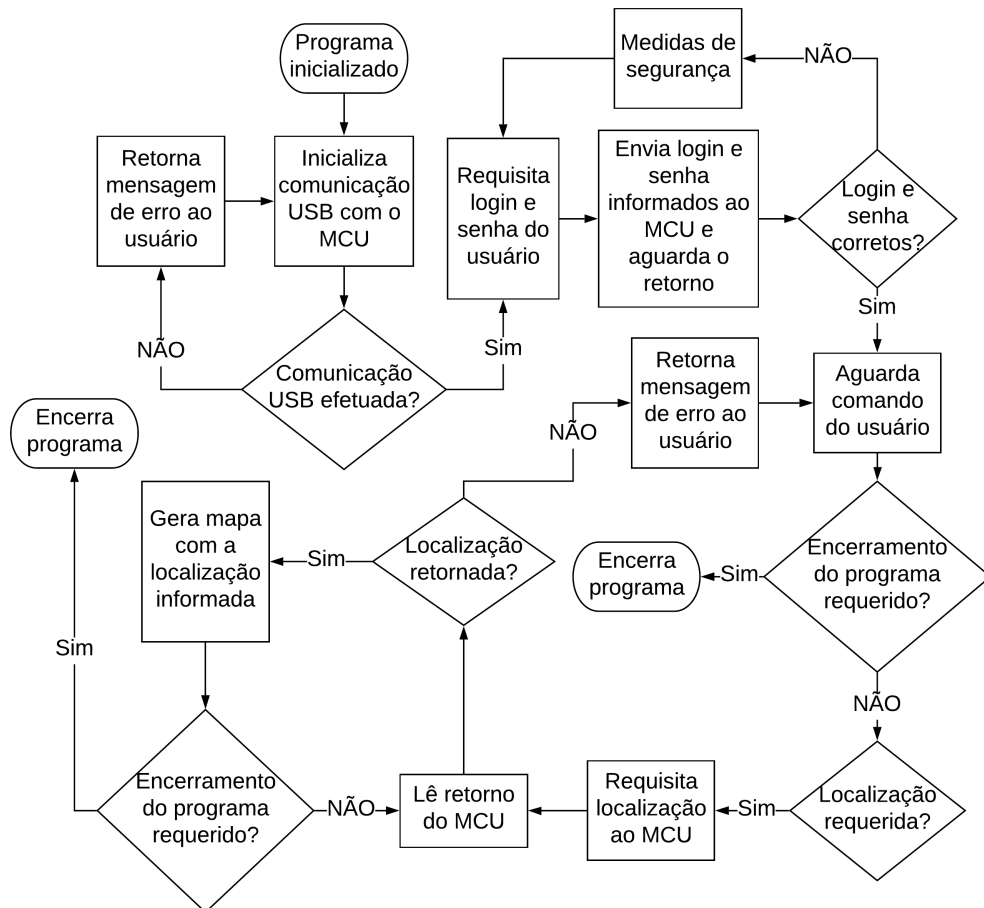


Figura 3.7: Fluxograma para o programa no computador.

Para o programa do computador pessoal, tem-se o diagrama de processos da figura 3.7. Enfatiza-se que o MCU no diagrama é o que encontra-se no dispositivo de monitoramento, e não o do dispositivo embarcado ao automóvel. O programa se inicia em uma página de *login*, onde é requisitado do usuário um PIN e uma senha. Neste processo, há uma etapa de conferência; como a senha não é armazenada no PC, mas sim nos dois microcontroladores, o aplicativo envia os dados passados pelo usuário e aguarda o retorno do dispositivo de monitoramento. Uma vez que o programa no PC tenha recebido o retorno, ele enfim informa ao usuário se o *login* foi efetuado ou não. Tendo sido efetuado, o programa automaticamente inicia a localização do veículo, exibindo um mapa.

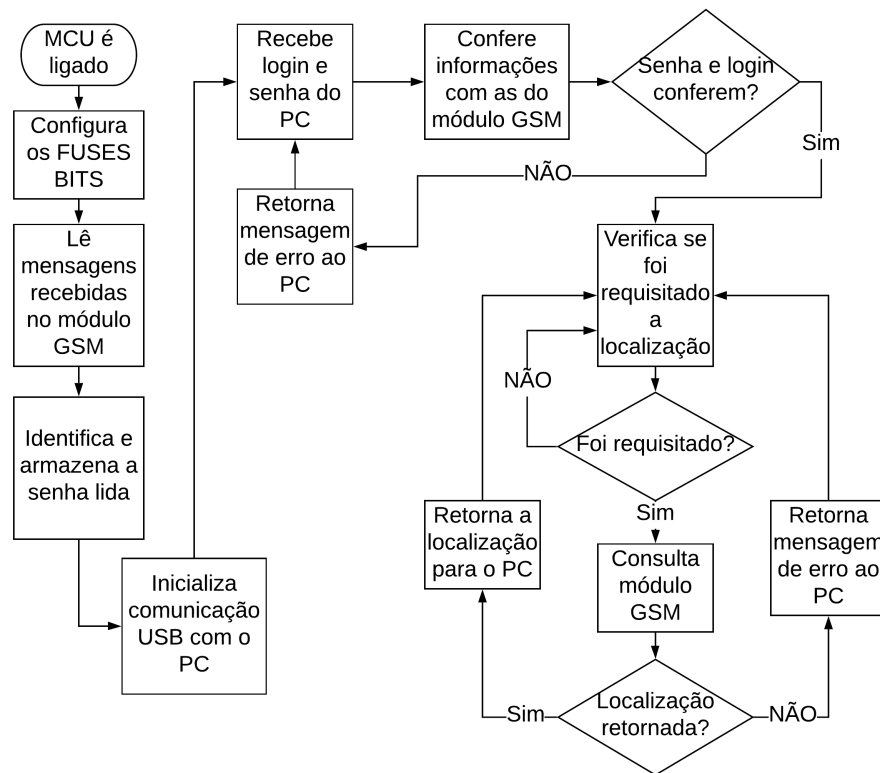


Figura 3.8: Fluxograma para o dispositivo de monitoramento.

Segue-se na figura 3.8 o diagrama desenvolvido para o microcontrolador do dispositivo de monitoramento. Neste, o MCU é ligado, passando para as configurações dos FUSES BITS. Em seguida, o mesmo lê as mensagens SMS enviadas pelo dispositivo embarcado. Nestas mensagens é contida a senha atualizada definida pelo cliente, de forma que o MCU armazena e atualiza na RAM o valor lido toda vez que é ligado. Em seguida, o MCU recebe o PIN e a senha passados pelo programa no PC; inicia-se então um processo de conferência das senhas e PIN - definindo-se que o PIN será o número do cartão SIM do dispositivo embarcado. Uma vez que o MCU receba o *login* do programa do computador, e, confira com sucesso com as informações armazenadas no próprio microcontrolador, inicia-se um processo de espera de novas requisições do PC. Caso o computador requira a localização, o MCU irá buscar tal informação nos SMSs do módulo GSM, e, havendo encontrado, retornará ao computador.

3.1.3 Análise de Falhas

Para o projeto comportamental foi feito, por fim, uma tabela de análise de falhas do projeto, disposta no Apêndice A; conhecida como FMEA (*Failure Mode and Effect Analysis*). Esta é utilizada para se prever possíveis falhas no projeto, e, tentar evitá-las (SAE, 2018). Na mesma, foi calculado um índice de risco, adimensional, para se comparar

as falhas citadas.

O caractere “G”, na tabela A.1, representa a gravidade - variando de 1 a 10, onde 10 é o nível mais grave. O caractere “O” indica a ocorrência, variando também de 1 a 10, onde 10 é a maior ocorrência. O caractere “D” representa a dificuldade de detecção antes da falha ocorrer, variando de 1 a 10, onde 10 é impossível de se detectar. A partir destes três valores, se calculou o índice de risco de cada falha, que é simplesmente a multiplicação dos três valores.

3.2 Projeto Lógico

Para o projeto lógico se definiu as seguintes especificações:

- O dispositivo embarcado deve ser alimentado com uma tensão de 12V, fornecida pela bateria do automóvel.
- O dispositivo embarcado deve conter módulos GPS e GSM.
- O MCU do dispositivo embarcado deverá ter uma memória EEPROM ou FLASH de aproximadamente 5KB; um mínimo de 16 pinos de entrada e saída de dados, e não muito mais de que isto. protocolo UART; e, *timers* de interrupção.
- O dispositivo embarcado deve ter o menor custo possível.
- O dispositivo de monitoramento deve ser alimentado por por uma fonte conectada à rede elétrica.
- O dispositivo de monitoramento deve dispor de uma comunicação USB e um módulo GSM.
- O MCU do dispositivo de monitoramento deve dispor de uma UART e comunicação USB.

Inicialmente, para o projeto lógico, fez-se necessário escolher os componentes eletrônicos a serem utilizados.

3.2.1 Escolha dos Microcontroladores

Para o dispositivo embarcado, deseja-se um MCU com as especificações estabelecidas. O número de 16 pinos I/O (entrada/saída) foi escolhido pois serão utilizados apenas dois PORTS de pinos digitais; sendo interessante que o microcontrolador não tenha uma quantidade de pinos muito superior ao indicado, para se evitar gastos desnecessários. Os *timers* também são necessários pois, como visto no projeto comportamental, será preciso

se fazer uso de interrupções. Assim, foi feita uma lista com alguns microcontroladores da *Microchip Technology Inc.* que estão próximos das especificações indicadas; conforme pode ser visto na tabela 3.1. Os valores de custo dos MCUs foram obtidos no site oficial da *Microchip*, e, convertidos de dólar para a moeda brasileira; as especificações dos mesmos foram obtidas nos *datasheets* respectivos de cada um, que também podem ser encontrados no site da *Microchip*.

Tabela 3.1: Comparativo das especificações de microcontroladores PIC.

MCU	Flash	RAM	EPROM	I/O	UART	Timers	Preço
16F627A	1024B	224B	128B	16	Sim	3	R\$6,00
16F628A	2048B	224B	128B	16	Sim	3	R\$6,80
16F648A	4096B	256B	256B	16	Sim	3	R\$8,00
16F87	7168B	368B	256B	16	Sim	3	R\$9,60
16F687	2048B	128B	256B	18	Sim	1	R\$6,20
16F886	8192B	368B	256B	24	Sim	3	R\$8,20

Em um primeiro momento, pensou-se no PIC16F87 e PIC16F886 como as melhores opções; pois ambos têm ótimas configurações de memória, e, um custo não elevado. Contudo, tais dispositivos não puderam ser encontrados à venda no mercado nacional, e, verificado o altíssimo valor da taxa de importação, se tornaram inviáveis. Assim, foi escolhido o PIC16F648A como o MCU do dispositivo embarcado.

Quanto ao MCU do dispositivo de monitoramento, os microcontroladores da família 18F atendem às especificações. Foi decidido se utilizar o PIC18F2550; pois o mesmo foi o de menor custo encontrado à venda no Brasil.

Decidiu-se também utilizar em ambos os microcontroladores um cristal de 8MHz.

3.2.2 Escolha dos Módulos GSM

Os módulos GSM de baixo custo disponíveis no mercado são, em geral, bastante similares na relação de preço. Praticamente todos se baseiam no CI SIM800L, ou, no SIM900. Escolheu-se o módulo GSM disposto na figura 3.9 devido à tensão de alimentação do mesmo, que é de 5V; diferentemente da maioria dos outros módulos GSM - que operam com uma tensão de alimentação de 3,8V a 4,4V. Esta característica se tornou interessante pois, tendo o módulo a mesma alimentação que o MCU, será necessário uma única fonte para os dois dispositivos.

O módulo escolhido possui terminais Rx/Tx para comunicação serial, via UART. Sendo que, esta comunicação é feita através de comandos AT, como descrito na fundamentação teórica. Além destes terminais, também possui um terminal de *reset*.



Figura 3.9: Módulo GSM escolhido, SIM800L V2 5V.

Fonte: <<https://www.makerlab-electronics.com/product/sim800l-v2-5v-wireless-gsm-gprs-module/>>. Acesso em: 17/05/2019.

3.2.3 Escolha do Módulo GPS

O módulo GPS escolhido encontra-se disposto na figura 3.10. O critério para a escolha do mesmo se baseou simplesmente no de menor custo possível.

Este módulo possui terminais de alimentação, sendo esta de 3,3V. Possuindo também terminais Rx/Tx para comunicação serial, via UART, implementando o protocolo NMEA; conforme discutido na fundamentação teórica.



Figura 3.10: Módulo receptor GPS escolhido, GY-GPS6MV2.

Fonte: <<https://www.caldeiratech.com/sensores/modulo-gps-para-arduino-e-raspberry-ublox-neo-6m-gy-neo6mv2/>>. Acesso em: 17/05/2019.

3.2.4 Projeto das Fontes de Alimentação

Para o dispositivo embarcado, foi necessário se desenvolver uma fonte que regule a tensão de 12V (valor fornecido pela bateria do automóvel) para 5V e 3,3V. Como visto anteriormente, o módulo GSM pode exigir da fonte até 2A; somado à corrente que o MCU também demandará, decidiu-se projetar uma fonte de 5V e 3A para estes dois componentes.

Assim, escolheu-se elaborar, para o módulo GSM e o MCU do dispositivo embarcado, uma fonte com o circuito integrado LM2596T. Como tais componentes consumirão muita corrente, é vital que neste caso a fonte tenha uma boa eficiência, para diminuir a chance de descarregar a bateria do veículo quando o mesmo estiver desligado e o dispositivo embarcado funcionando.

Além disto, decidiu-se utilizar o CI regulador de tensão linear LM1117T para fornecer os 3,3V ao módulo GPS. Já neste caso, como o módulo GPS consome pouca corrente, o uso de um regulador com má eficiência não irá impactar tanto sobre a bateria do automóvel; escolhendo-se tal CI pela facilidade de uso do mesmo. O CI LM1117T tem o funcionamento similar ao LM7805 descrito anteriormente; mudando apenas a tensão de saída, que é de 3,3V.

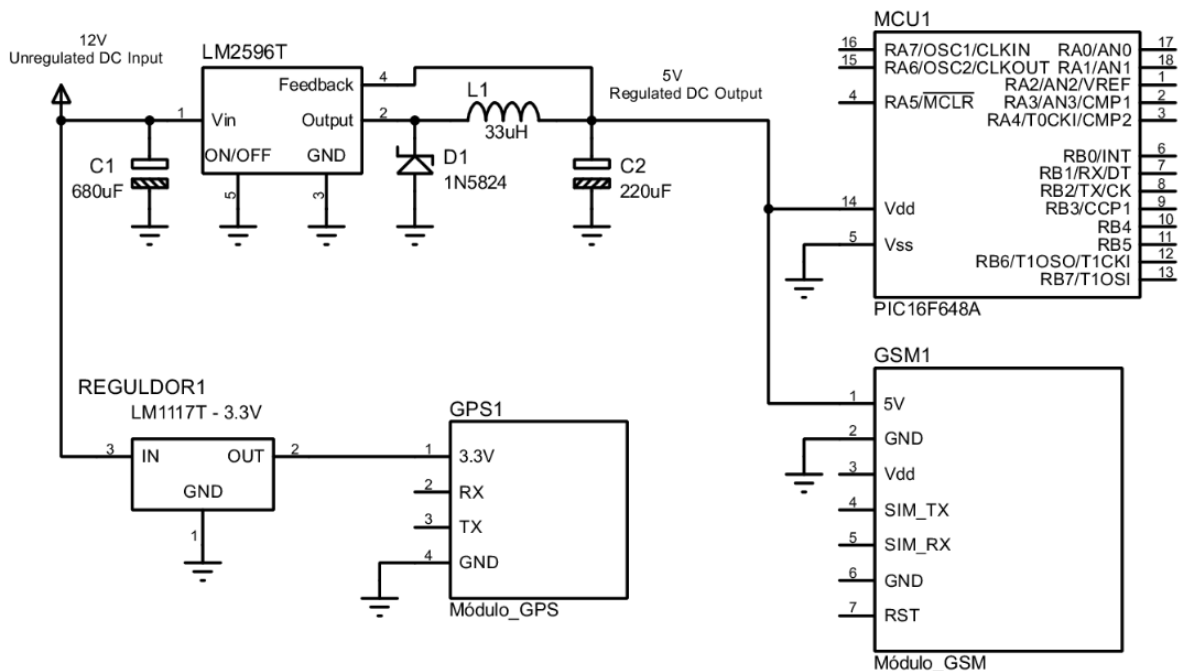


Figura 3.11: Circuito de alimentação para o dispositivo embarcado.

Para o dispositivo de monitoramento, será necessário uma fonte que também possa fornecer até 3A. Contudo, a conversão dos valores de tensão deverá ser de 127Vrms para 5Vdc, já que o cliente utilizará uma tomada com a tensão da rede elétrica. Assim, para este dispositivo, decidiu-se utilizar uma fonte conforme a figura 3.12, de 5V e 3A.



Figura 3.12: Fonte de alimentação de 5V e 3A, com conector USB.

Fonte: <https://produto.mercadolivre.com.br/MLB-1074970603-fonte-raspberry-5v3a-com-interruptor-_JM?quantity=1>. Acesso em: 23/05/2019.

3.2.5 Projeto do Circuito de Sensores e Atuadores

Pôde-se observar que, atualmente, a maioria dos automóveis vêm de fábrica com os sensores fim-de-curso das portas e dos vidros. Assim, para o sensoriamento no dispositivo embarcado, foi projetado apenas um resistor *pull-down* para cada sensor do veículo com uma conexão entre o microcontrolador e um *borne*; e, no momento da instalação deverá se conectar o *borne* aos sensores do veículo. A figura 3.13 representa o o circuito de *pull-down* mencionado.

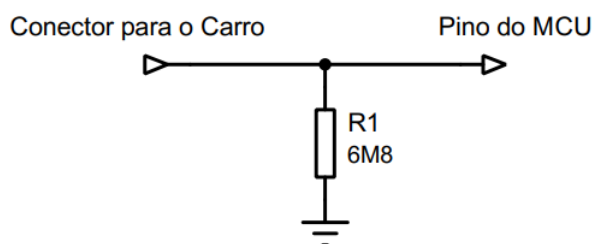


Figura 3.13: *Pull-down* para conectar os sensores do veículo no dispositivo embarcado.

Quanto aos atuadores, decidiu-se utilizar relés que terão os contatos em paralelo ou em série com os atuadores do carro. Como a bobina do relé é um componente de baixíssima resistividade, acioná-la diretamente pelo MCU pode vir a queimar a saída deste. Logo, utilizando-se um transistor e um diodo de proteção contra tensão reversa, foi desenvolvido um circuito para que o microcontrolador possa energizar a bobina do relé de maneira isolada. Segue-se na figura 3.14 o circuito de atuação desenvolvido.

O relé a ser conectado à ignição deve ser instalado em série com a mesma. Isto pois

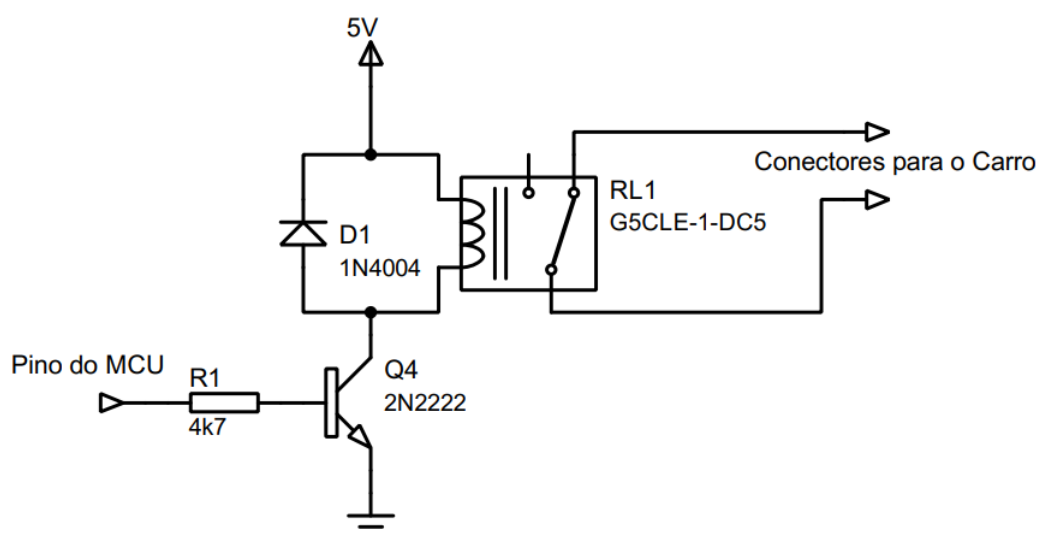


Figura 3.14: Circuito do dispositivo embarcado para acionar os atuadores no automóvel.

deseja-se que o relé possa desligar o carro independentemente da situação (lógica OR). Este relé também deve ser, como uma medida de segurança, normalmente fechado; isto para que, no caso de uma falha onde o dispositivo embarcado não ligue, ou, desligue sozinho, o circuito não venha a cortar a ignição indesejadamente.

Os relés para o vidro elétrico e a trava elétrica devem ser instalados em paralelo com tais circuitos. Isto para que se possa acioná-los independentemente dos botões do carro não estarem pressionados (lógica AND). Além disto, como uma medida de segurança, estes relés devem ser normalmente abertos; para que não venham a acionar os atuadores indesejadamente no caso do dispositivo embarcado ter sido desligado.

3.2.6 Esquemáticos dos Dispositivos de Hardware

Com base no que foi desenvolvido até então, pôde-se gerar o circuito esquemático do dispositivo embarcado e do dispositivo de monitoramento; sendo que ambos seguem em anexo no Apêndice A - o dispositivo embarcado na figura A.2 e o dispositivo de monitoramento na figura A.3.

O esquemático desenvolvido do dispositivo embarcado consiste na junção do circuito de alimentação, os módulos, o microcontrolador, os circuitos de sensores e atuadores, com um conector, um cristal oscilador e dois diodos de proteção na alimentação de 12V.

O esquemático desenvolvido para o dispositivo de monitoramento consiste simplesmente em um microcontrolador conectado a um módulo GSM, um cristal, e, dois conectores USB - um para alimentação e outro para conectar ao computador.

Destaca-se que, até este ponto, os circuitos não foram simulados. Assim, existindo a possibilidade dos mesmos serem alterados no Projeto Físico, caso surja algum problema.

3.3 Projeto Físico

Para o projeto físico, definiu-se as seguintes especificações:

- Obter-se os circuitos finais, tanto esquemáticos quanto os PCBs, dos dispositivos de *hardware* de forma funcional e validada.
- Obter-se os programas dos microcontroladores e do computador pessoal, de forma que atendam os limites de memórias de cada respectivo *hardware*.
- Obter-se os protótipos do projeto funcionais.

Em cada tópico a seguir, é contido as etapas de desenvolvimento físico e construção dos respectivos dispositivos. Todos os programas elaborados estão sendo enviados separadamente do texto deste trabalho, uma vez que ficaram muito extensos.

3.3.1 Dispositivo de Monitoramento

Depois de algumas simulações no *software* Proteus, iniciou-se a programação e testes com o dispositivo de monitoramento montado em uma *protoboard*.

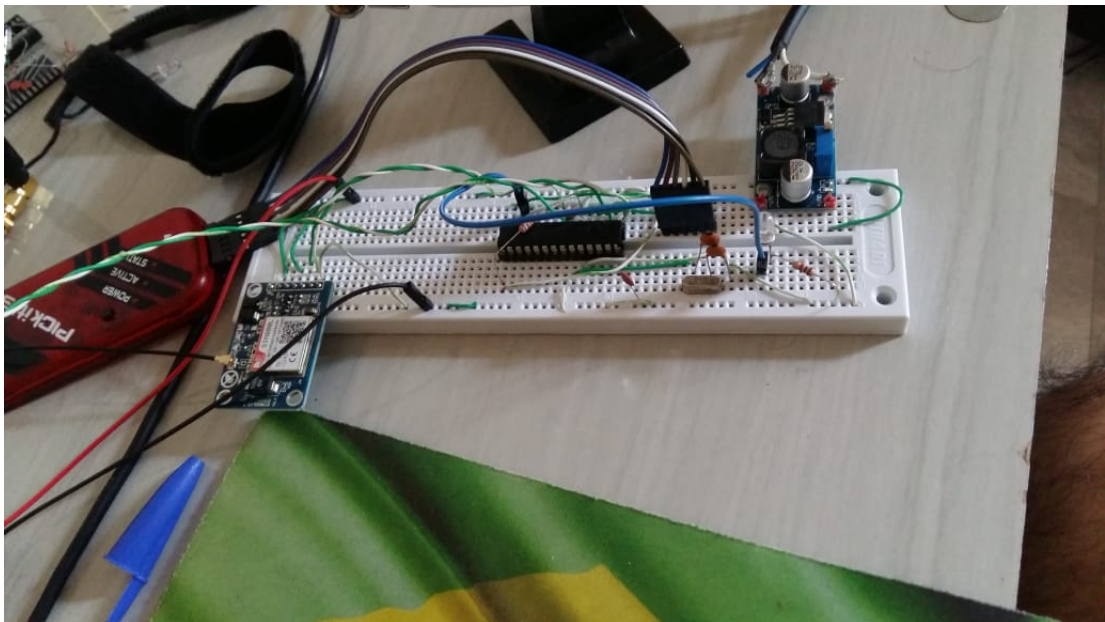


Figura 3.15: Dispositivo de Monitoramento montado em uma *protoboard*.

Inicialmente, trabalhou-se em estabelecer uma comunicação USB entre o dispositivo de monitoramento e o computador. Para isso, foi necessário se criar um programa descritor, como abordado na fundamentação teórica.

Graças a um recurso da IDE MikroC Pro for PIC, este programa pôde ser gerado sem dificuldades. Na IDE, na aba “Tools” e depois clicando em “HID Terminal”, surgirá uma

janela conforme a figura 3.16. Nesta janela, devemos definir o *Vendor Name* e o *Product Name*. Esses nomes serão informados ao usuário pelo computador sempre que a conexão USB for estabelecida, podendo ser vistos em “Dispositivos e Impressoras”. Deve-se definir também o *Vendor ID* (VID) e o *Product ID* (PID). Essas informações são necessárias para que, no programa a ser desenvolvido no computador, seja possível se encontrar o dispositivo conectado via USB, independentemente de em qual porta COM for conectado.

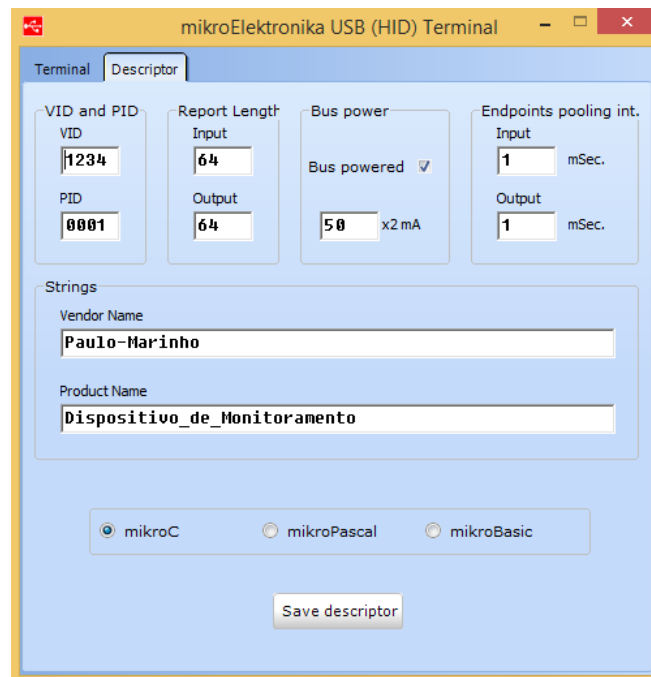


Figura 3.16: Ferramenta do MikroC para se gerar o programa descritor.

Também foi necessário se definir o *Report Length* de entrada e saída, que é a dimensão dos vetores de *bytes* que serão trocados entre o MCU e o computador. Escolhendo-se 64, definimos que na comunicação USB poderá ser enviado vetores de até 64 posições, onde em cada posição armazena-se um *byte*.

Clicando-se em *Save descriptor*, a IDE irá gerar um arquivo “USBdsc.c”. Este arquivo deverá ser adicionado ao projeto do microcontrolador, e, uma vez gerado o arquivo de extensão “.hex”, o descritor estará contido neste arquivo “.hex” (arquivo a ser gravado no MCU).

Após alguns testes, pôde-se observar que o computador estava identificando o MCU do dispositivo de monitoramento corretamente, conforme a figura 3.17.

A biblioteca de comunicação USB do MikroC Pro for PIC conta com as funções *HID_Read()* e *HID_Write()*, pelas quais foi possível enviar e receber dados do lado do microcontrolador. Tais funções são rotinas de bloqueio, ou seja, uma vez chamadas, o MCU não sai destas rotinas até que tenham sido realizadas com sucesso.

Rotinas de bloqueio devem ser utilizadas com muito cuidado, pois podem gerar trava-

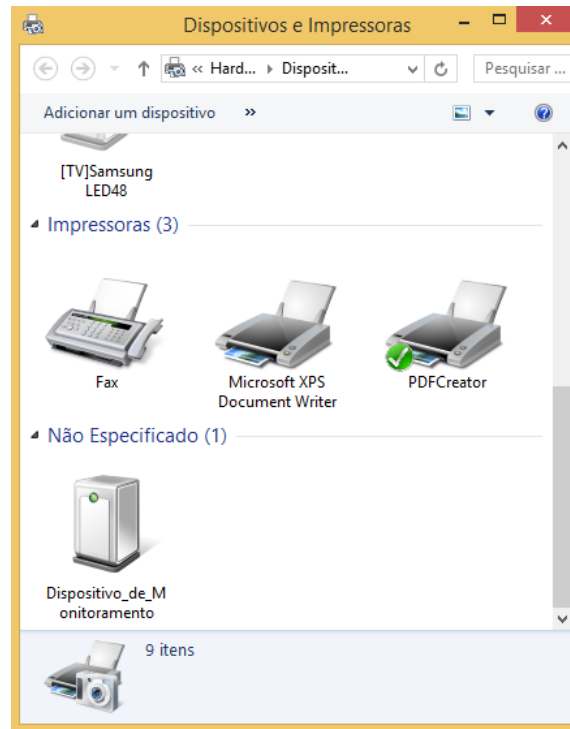


Figura 3.17: Dispositivo de Monitoramento sendo reconhecido pelo PC após conexão USB.

mentos permanentes no código. Para se evitar este tipo de *bug*, várias medidas podem ser tomadas, como o WDT ou a utilização de *timers()* para se gerar interrupções e a quebra dessas rotinas.

No caso da comunicação USB, estas rotinas de bloqueio não geraram nenhum tipo de travamento no código, pois no próprio programa do computador implementou-se algoritmos que conferem o formato e dimensão dos datagramas a serem enviados e recebidos.

Ao se tentar estabelecer a comunicação entre o MCU e o módulo GSM via UART, foi preciso se utilizar uma série de funções disponíveis no MikroC PRO for PIC e que também são rotinas de bloqueio. Neste caso, considerando a possibilidade do módulo GSM receber SMSs indesejados, como ofertas de operadoras ou algo similar, viu-se que seria necessário se evitar um possível *bug*. Isto pois, já que as funções para leitura de SMS são rotinas de bloqueio, caso o MCU tente ler uma mensagem com uma quantidade de caracteres inferior ao esperado, o processador irá travar, aguardando indefinidamente o término da mensagem.

Desta forma, se utilizou o *timer0* do MCU. Este realiza uma contagem de alguns poucos segundos até ter o estouro. Na função de interrupção, foi chamado uma função, também proporcionada pelo MikroC PRO for PIC, que pula a última função de leitura da UART utilizada. Assim, o MCU tem alguns poucos segundos para ler um possível SMS, e, se não conseguir concluir, entender-se-á que o formato da mensagem é inválido, retirando o processador do ponto de travamento.

Definiu-se que, os SMSs a serem recebidos pelo dispositivo de monitoramento possuem um formato do tipo: “SSSSSxxxx.xxxxxYYYY.YYYYY”. No lugar dos caracteres “S”, serão contidos cinco caracteres referentes à senha do usuário. Esta senha permanece salva na EEPROM do dispositivo embarcado, e, o modo de alteração da mesma ainda será tratada posteriormente. No lugar dos caracteres “x”, estará contido a latitude no formato geodésico, que é o formato utilizado na sentença GPGLL. E, no lugar dos caracteres “Y”, estará contido a longitude, também no formato geodésico.

Um exemplo de um possível SMS recebido pelo dispositivo de monitoramento segue-se na tabela 3.2.

Tabela 3.2: SMS recebido: “123455221.145322134.47291”.

12345	Senha previamente definida pelo usuário
5221.14532	Latitude no formato geodésico (graus e minutos)
2134.47291	Longitude no formato geodésico (graus e minutos)

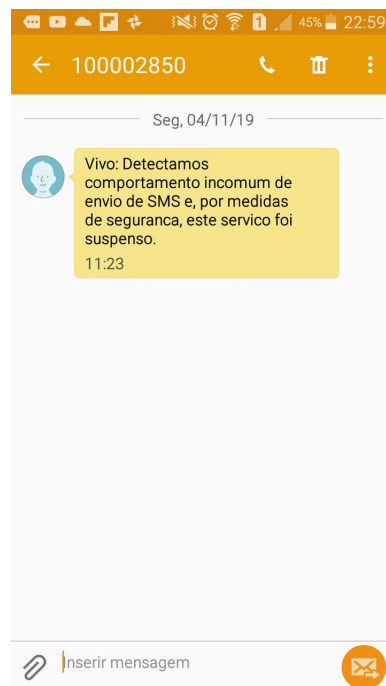


Figura 3.18: Filtro *anti-spam* da operadora Vivo bloqueando o envio de SMSs.

Realizando-se mais alguns testes com o protótipo, percebeu-se que as operadoras possuem filtros *anti-spam* SMS. Foi possível se perceber isto pois o cartão SIM que estava sendo usado nos testes fora bloqueado durante uma semana, após o envio de muitas mensagens, conforme a figura 3.18. Assim, alterou-se a proposta nos fluxogramas onde o dispositivo embarcado enviaria SMSs ininterruptamente ao dispositivo de monitoramento.

Sendo definido que o dispositivo embarcado envia um SMS ao dispositivo de monitoramento apenas quando o usuário requisita a localização pelo seu celular.

Quanto ao dispositivo de monitoramento, fez-se, por fim, um novo circuito esquemático, disposto em A.5. Fez-se também o PCB (*Printed Circuit Board*) do mesmo. Para isso foi utilizado o método de transferência fotográfico, que se baseia em utilizar uma luz ultravioleta para gravar com tinta o circuito desejado em uma placa fenolite, seguido da corrosão da placa. Segue-se nas figuras 3.19, 3.20, 3.21, 3.22, 3.23 e 3.24 fotos tiradas durante a confecção do dispositivo.

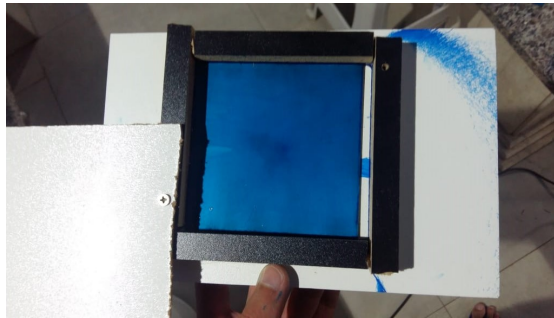


Figura 3.19: Pintura da placa de fenolite com tinta fotossensível.

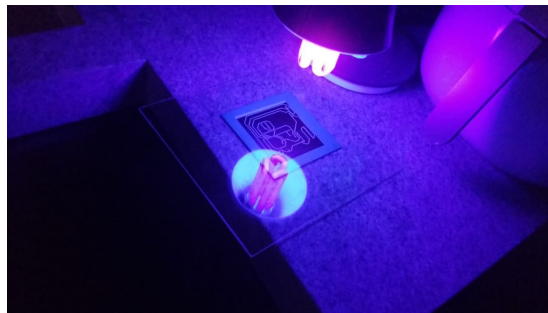


Figura 3.20: Transferência do *Layout* para a placa pelo método fotográfico.

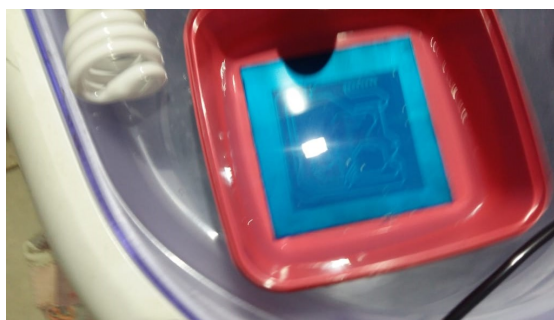


Figura 3.21: Remoção da tinta não sensibilizada pela luz ultravioleta.

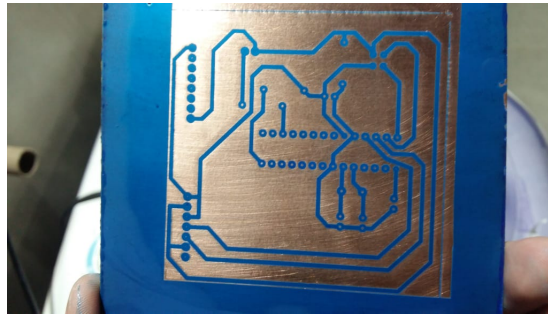


Figura 3.22: *Layout* transferido para a placa com sucesso.

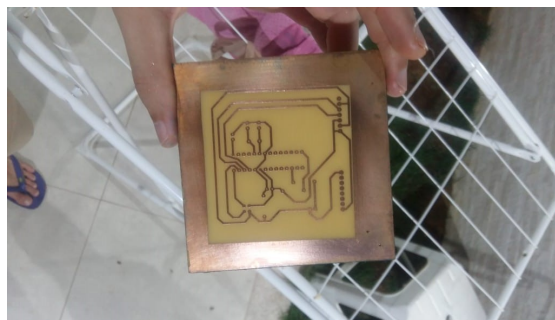


Figura 3.23: Corrosão da placa de fenolite.



Figura 3.24: Placa de circuito impresso finalizada.

3.3.2 Programa de Rastreamento no Computador

Paralelamente ao Dispositivo de Monitoramento, iniciou-se a realização do programa para o computador.

Para que o programa no computador possa receber e enviar informações ao microcontrolador via USB, é necessário que o mesmo conte com uma DLL (*Dynamic Link Library*). Buscou-se na internet algumas DLLs prontas, e, foi possível se encontrar várias opções. Contudo, encontrou-se uma enorme dificuldade de utilizá-las na linguagem JAVA. Pesquisando-se um pouco mais, descobriu-se que, apesar de a linguagem JAVA ser excelente para o desenvolvimento de aplicações, a mesma não se comunica bem com bibliotecas de código nativo já existentes.

Assim, percebeu-se que a linguagem C# seria mais conveniente para a realização do programa do computador. Alterando-se então para tal linguagem o desenvolvimento do programa no computador.

Para estabelecer a comunicação do dispositivo de monitoramento com o computador, utilizou-se uma biblioteca proporcionada pelo *Visual Studio*.

É sabido que, a senha do usuário permanece salva no dispositivo embarcado, mas não no dispositivo de monitoramento. Na realidade, o dispositivo de monitoramento obtém a senha sempre que é ligado ao realizar a leitura de um dos SMSs enviado pelo dispositivo embarcado. Desta forma, para se evitar que o usuário tente logar, pelo computador, antes que o dispositivo de monitoramento tenha conseguido ler e atualizar a senha através de um SMS recebido, os botões para login permanecem inabilitados. Quando o dispositivo de monitoramento armazena a senha, segue enviando uma mensagem via USB para o programa no computador, que só então permite a tentativa de login.

Também foi feito uma caixa de diálogo onde o programa no computador informa ao usuário em que estado se encontra o MCU do dispositivo de monitoramento, para que se facilite identificar possíveis falhas.

Para transformar a latitude e a longitude recebida em um mapa, fez-se, dentro do programa desenvolvido, um *browser* de Internet. Assim, ao iniciar o rastreamento, o programa abre uma página com o endereço do *Google Maps*, e, atualiza automaticamente o local de busca tendo como base a localização recebida do dispositivo de monitoramento. Para que isso fosse possível, foi preciso se criar algoritmos que gerassem o DNS (*Domain Name System*) do *Google Maps* na localização desejada. Para só então atualizar o *browser* com o endereço correto.

Definiu-se que, para realizar o *login*, o usuário deve digitar o número SIM do dispositivo embarcado no campo escrito “PIN”. Em seguida deve preencher o campo “Senha” com a mesma senha utilizada no dispositivo embarcado, e, clicar no botão de *login*.

As imagens com o programa em execução seguem no capítulo de resultados.

3.3.3 Dispositivo Embarcado

Foram feitos testes em uma *protoboard* com o circuito do dispositivo embarcado, utilizando-se o PIC16F648A, escolhido no projeto lógico. Contudo, de imediato encontrou-se um problema intransponível.

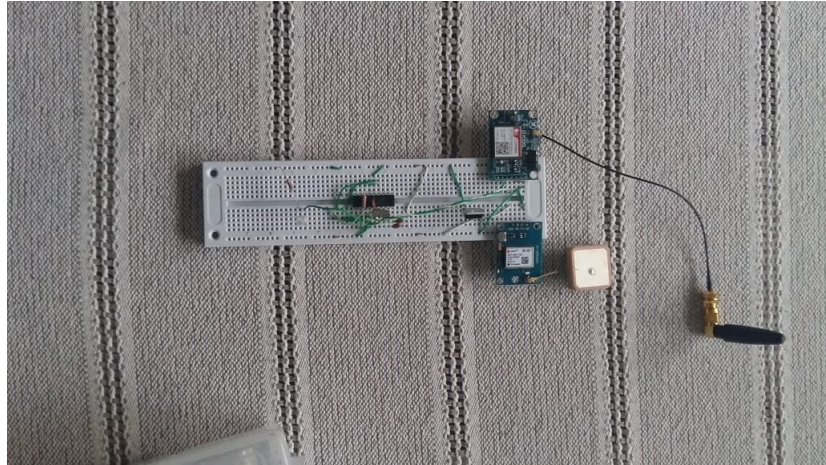


Figura 3.25: Dispositivo Embarcado na *protoboard*, utilizando o PIC16F648A.

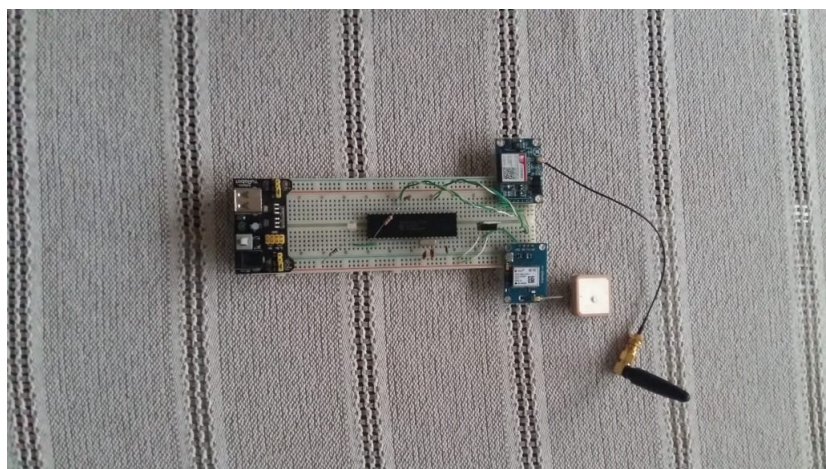


Figura 3.26: Dispositivo Embarcado na *protoboard*, utilizando o PIC18F4550.

Basicamente, a memória RAM do PIC 16F648A não permite a criação de vetores com mais de 90 posições. A princípio, pensou-se que esta dimensão de vetores seria suficiente para a elaboração do programa do MCU.

Contudo, notou-se que, ao se fazer a leitura via UART dos caracteres recebidos pelo módulo GPS ou GSM, não era possível tratar a informação nos mesmos laços em que ocorria a leitura. Isto pois, no padrão UART, o dispositivo transmissor não verifica se o receptor de fato recebe a informação. E, se o MCU realizava alguma operação lógica entre a leitura de caracteres, no momento em que realizasse tal operação, o módulo continuaria a enviar caracteres que seriam completamente ignorados pelo MCU.

Assim, foi preciso se armazenar integralmente o conjunto de caracteres enviados pelos módulos em uma variável vetorial. Para só depois tratar a informação e retirar o conteúdo de interesse.

Diante de tal constatação, vendo-se que as sequências de caracteres enviadas pelo módulo GPS ultrapassavam 700 caracteres, concluiu-se que com a memória RAM do PIC 16F648A seria de fato impossível a realização deste trabalho.

Assim, tendo em mãos um PIC18F4550, fez-se uso do mesmo para a montagem do dispositivo embarcado, embora as especificações deste ultrapassem e muito o necessário.

Além dessa alteração, também foi acrescentado um LED ao circuito, para facilitar a análise do funcionamento do dispositivo. E, simplificou-se o uso do regulador LM2596T, de forma que se comprou um circuito impresso já contendo os componentes recomendados para o uso deste regulador, semelhante ao da figura 2.7.

Uma vez que este microcontrolador se comunica com dois dispositivos distintos via UART, fez-se uso da UART física do MCU, e, de uma UART implementada por código.

A UART contida no *hardware* do MCU foi utilizada para ler os caracteres enviados pelo módulo GPS. Desta forma, utilizou-se registradores do PIC para gerar uma interrupção sempre que o *bit* de *Start* da UART fosse identificado. Seguindo-se acrescentando o caractere lido a um vetor de caracteres. Este vetor, uma vez preenchido completamente, é tratado, onde busca-se a sentença GPGLL e extrai-se dela as coordenadas desejadas.

A comunicação com o módulo GSM foi feita com uma UART implementada por *software*. Isto foi possível graças a mais uma biblioteca do MikroC PRO for PIC.

No código do MCU, definiu-se um padrão para o envio de SMS ao dispositivo embarcado, sendo: “SSSSS-xx”. Onde, no lugar dos caracteres “S”, o usuário deve informar sua senha. No lugar dos caracteres “x”, o usuário terá um código para acionar os relés. O primeiro caractere “x” faz referência a um dos três relés, podendo ser preenchido com os caracteres “1”, “2”, ou “3”. Já o segundo caractere “x” faz referência ao estado do relé; desejando-se nível alto, deve-se preencher com “1”, e para nível baixo, com “0”. Assim, supondo que a senha do usuário seja “12345”, e o usuário queira deixar o relé 1 em nível alto, deve-se portanto enviar o SMS “12345-11” ao dispositivo embarcado.

Caso o usuário queira apenas visualizar a localização do carro e verificar os níveis dos sensores, deverá enviar sua senha seguida de “40”.

Foi estabelecido ainda mais um comando para a alteração da senha. O dispositivo vem, por padrão, com a senha “12345”, e o usuário deverá mudá-la conforme seu gosto. Para isso, deve-se enviar um SMS com a seguinte estrutura: “SSSSSxxYYYYY”. Onde nos caracteres “S” consta a antiga senha; nos caracteres “x” deve-se ter “50”, que é o comando para alteração de senha; e, nos caracteres “Y” deve-se constar a nova senha.

A nova senha enviada pelo usuário é armazenada na memória EEPROM do MCU.

Desta forma, mesmo desligando-se o dispositivo, a senha permanecerá atualizada.

O dispositivo embarcado foi programado para identificar o número que o enviou um SMS, e, assim retornar o SMS ao mesmo número. Não havendo a necessidade de armazenar o número do usuário e permitindo que o mesmo faça consultas de qualquer aparelho.

Tendo-se em vista evitar o demasiado consumo de energia do dispositivo embarcado, pensou-se em utilizar algum modo de suspensão ou economia de energia. O módulo SIM800L pode ser configurado em três modos no que diz respeito ao consumo do mesmo, isto através do comando “AT+CSCLK=n”. No lugar do caractere “n”, deve-se preencher com “0”, “1” ou “2”. Sendo preenchido com “0”, o módulo nunca irá entrar no modo de suspensão; caso preenchido com “1”, um parâmetro do SIM800 denominado DTR determinará se o módulo entrará ou não no modo de suspensão; caso preenchido com “2”, o módulo entrará no modo de suspensão automática, de forma que, sempre que há interrupções na UART ou algum SMS recebido, o módulo sai do modo de suspensão. Desta forma, optou-se pelo modo de suspensão automática, utilizando o comando “AT+CSCLK=2”. No modo de suspensão, o consumo médio do módulo é em torno de 1mA.

Elaborou-se então um novo circuito esquemático para o dispositivo embarcado, com as alterações discutidas. Este pode ser conferido na figura A.4 do Apêndice A.

Fez-se o PCB do mesmo, também através do método fotográfico. Neste caso, utilizando-se uma placa dupla face, já que havia muitas trilhas no circuito impresso. A imagem do dispositivo prototipado segue no capítulo de resultados.

Resultados e Análise de Resultados

Segue-se nas figuras 4.1 e 4.2 os *designs* dos dispositivos de *hardware* gerados no Proteus.

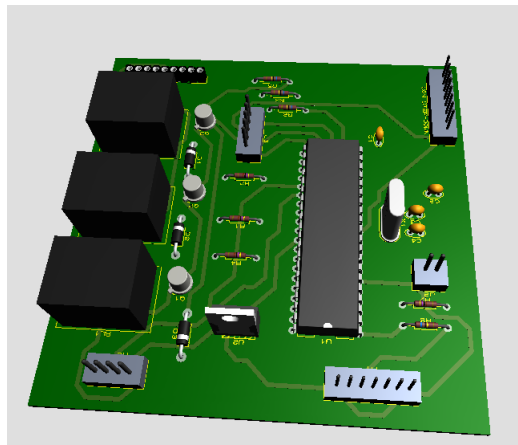


Figura 4.1: *Design* do dispositivo embarcado simulado no Proteus.

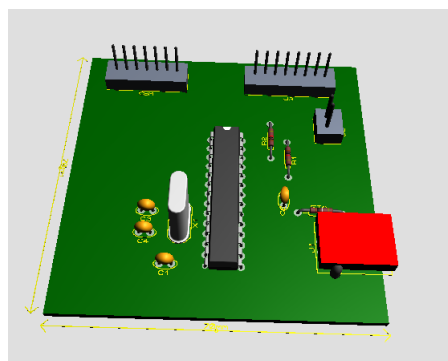


Figura 4.2: *Design* do dispositivo de monitoramento simulado no Proteus.

Segue-se nas figuras 4.3 e 4.4 os dispositivos de *hardware* prototipados.

Com relação a tais dispositivos, embora tenham funcionado adequadamente, durante os testes pôde-se perceber que o tempo de resposta dos mesmos deixa a desejar. O módulo

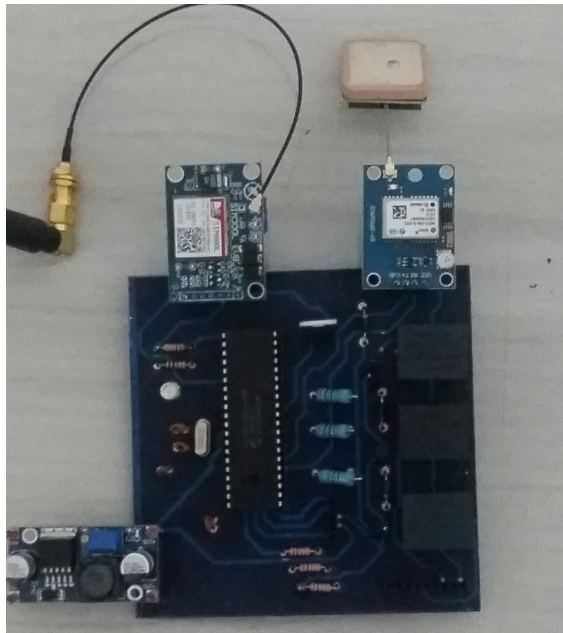


Figura 4.3: Dispositivo embarcado prototipado.



Figura 4.4: Dispositivo de monitoramento prototipado.

GSM normalmente leva cerca de um minuto para conectar-se à rede GSM; contudo, em uma má cobertura, pode levar mais tempo. Além do que, a conexão GSM se comporta de uma maneira bastante instável, muitas vezes caindo e demorando muito tempo para retomar a conexão. O módulo GPS possui uma conexão mais estável, contudo mostrou demorar mais tempo para estabelecer a conexão, podendo levar até cerca de 5 minutos.

Segue-se na figura 4.5 a foto de um teste realizado com o dispositivo embarcado, onde se enviou mensagens SMS com comandos para se alterar o estado dos relés e se obteve a resposta do módulo. Neste caso, o teste foi feito de modo que os sensores eram chaves em uma *protoboard*, de forma que não eram alterados com a mudança de estado dos relés. Contudo, foram ligados LEDs aos relés, e pôde-se observar o correto funcionamento do dispositivo.

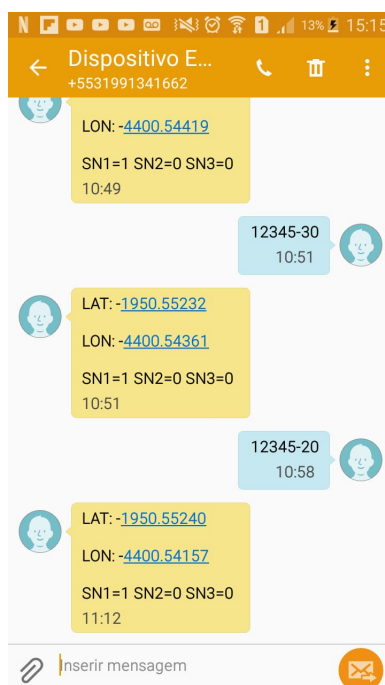


Figura 4.5: Enviando comandos ao dispositivo embarcado e recebendo os retornos.

Com relação ao consumo do dispositivo embarcado, podemos fazer a seguinte conta: considerando os três relés que consomem cerca de 50mA cada; estimando para o MCU se gasta cerca de 42mA, pois possui 4 terminais de saída consumindo em média 10mA cada, mais 2mA referente aos terminais de alimentação; pensando-se que o módulo GPS esteja consumindo 100mA, o que seria a pior das hipóteses, já que seu consumo varia de 10mA a 100mA; considerando que o módulo GSM esteja em modo de suspensão, com apenas 1mA; e, que a eficiência dos reguladores de tensão seja uma média de 80%. Assim, teríamos um consumo de corrente de aproximadamente:

$$P = (50mA * 3 * 5V + 42mA * 5V + 100mA * 3.3V + 1mA * 5V) * \frac{100}{80} = 1.619W \quad (4.1)$$

Para 1,619W, considerando a bateria do automóvel de 12V, teríamos uma corrente média saindo da bateria por volta de 135mA. Este valor, para uma bateria automotiva de 50 Ampéres/Hora, por exemplo, é um consumo razoavelmente baixo. Contudo, ainda assim não se deve deixar o módulo ligado a um carro por vários dias sem utilizar o veículo, estando sob o risco de descarregar a bateria.

Quanto ao programa desenvolvido no computador, segue-se nas figuras 4.6, 4.7, 4.8, 4.9 e 4.10 todas as possíveis telas que o mesmo pode se encontrar.

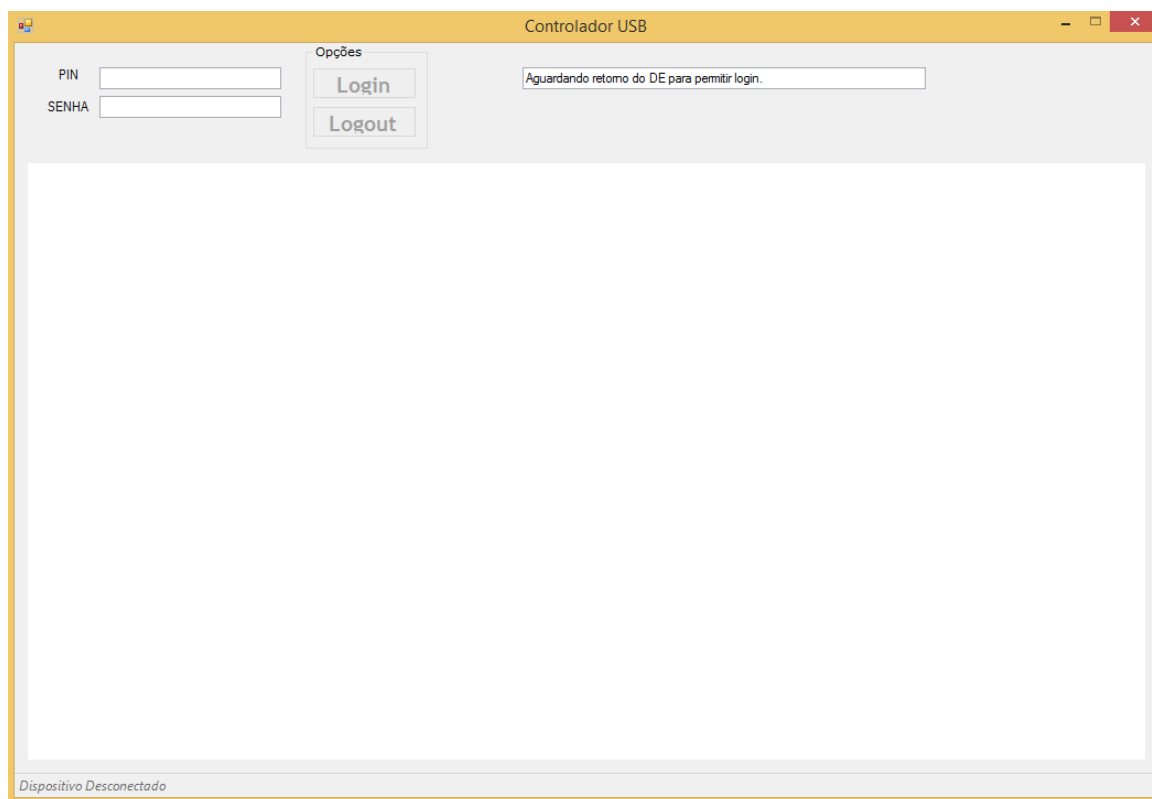


Figura 4.6: Programa indicando que o dispositivo de monitoramento não foi conectado.

Na figura 4.6, o programa identifica que o dispositivo de monitoramento não foi conectado ao computador, e, indica esta informação no canto inferior esquerdo da tela.

Na figura 4.7, o programa identifica que o dispositivo de monitoramento foi conectado ao computador. Contudo, ainda não habilita o botão de *login*, pois o dispositivo de monitoramento ainda não atualizou a senha.

Na figura 4.8, o dispositivo de monitoramento já conseguiu ler a senha em um SMS vindo do dispositivo embarcado. Assim, o programa no computador recebe uma notificação do dispositivo de monitoramento e habilita o botão de *login*. Uma vez que o usuário digite o PIN e a senha, o programa irá enviar estas informações ao MCU e, estando corretas, o MCU retornará a localização. Na figura 4.10, pode-se observar a localização sendo efetuada com sucesso.

O programa para o computador funcionou bem como o esperado. E, utilizando-se o

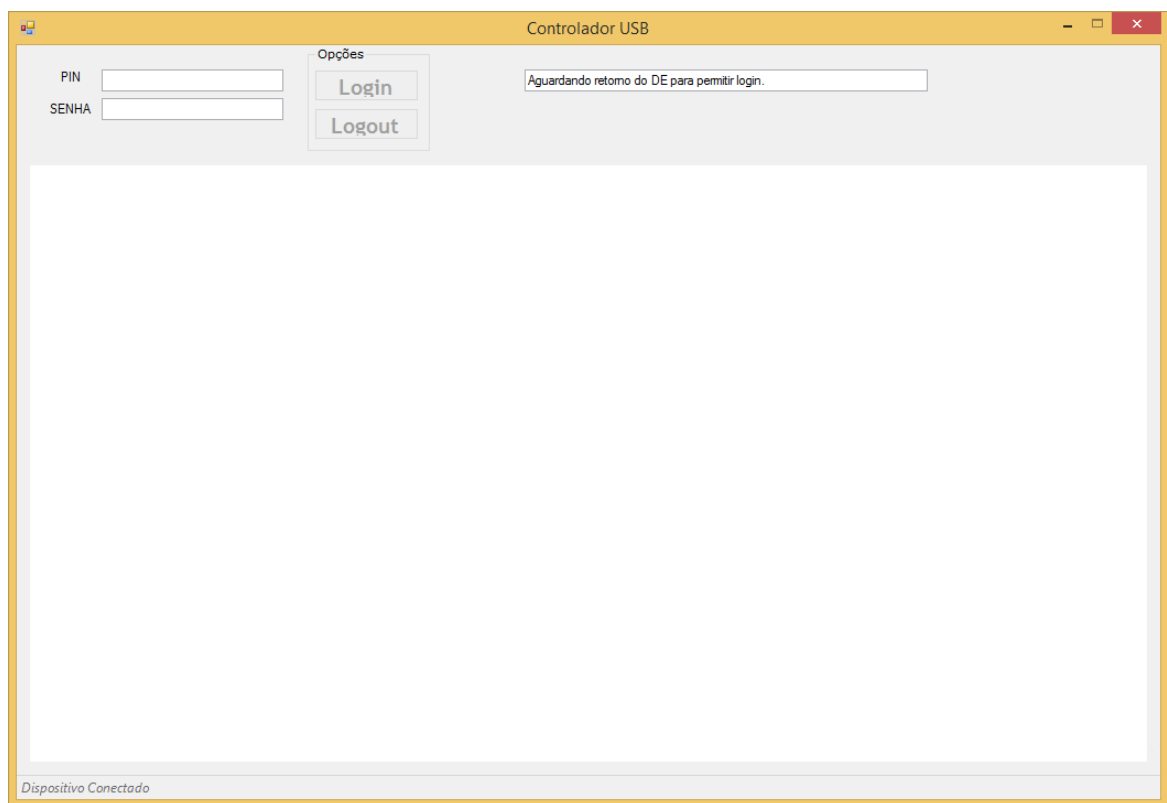


Figura 4.7: Dispositivo de monitoramento foi conectado, mas não atualizou a senha.

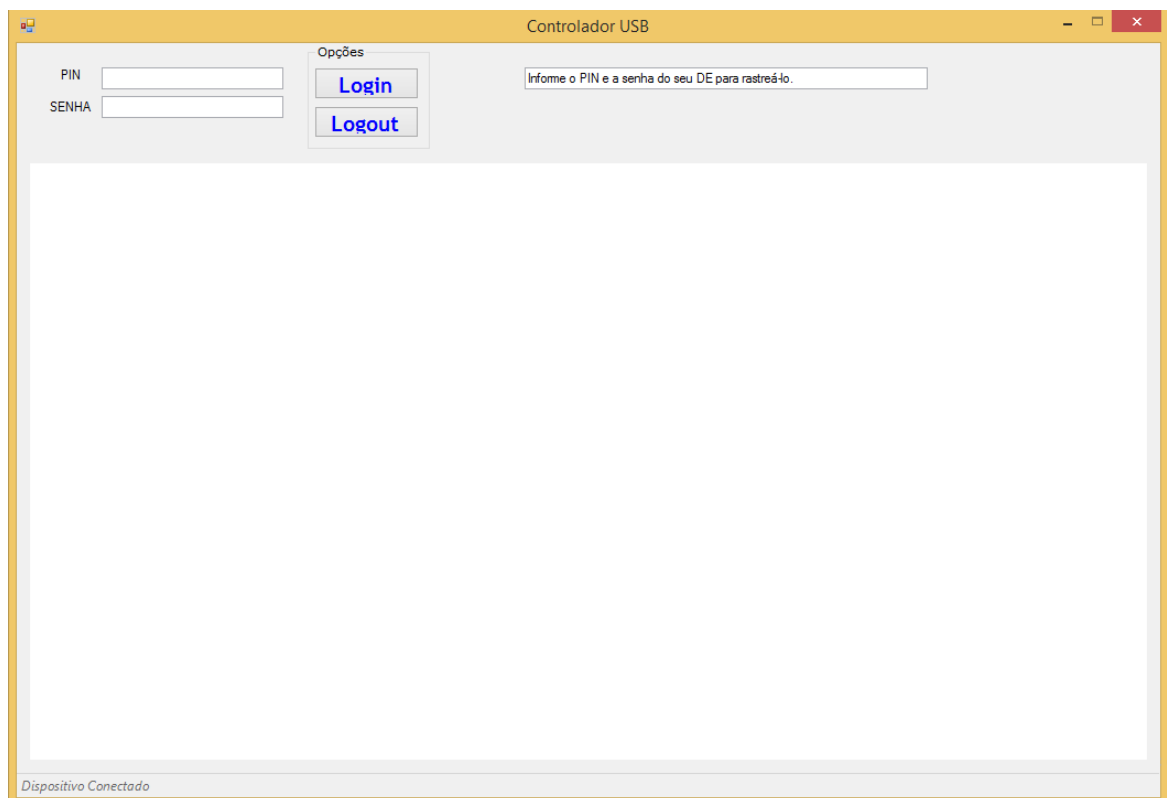


Figura 4.8: Programa permitindo tentativa de *login*.

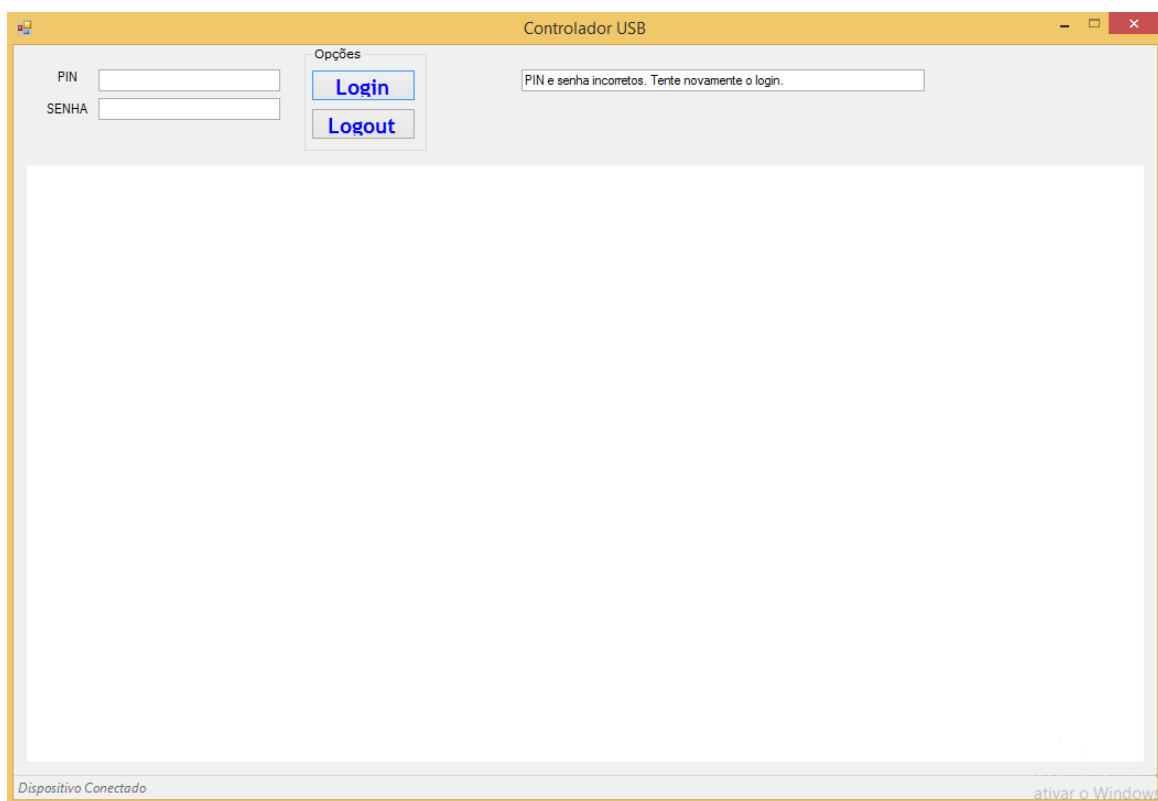


Figura 4.9: PIN ou Senha incorretos, não efetuando o *login*.

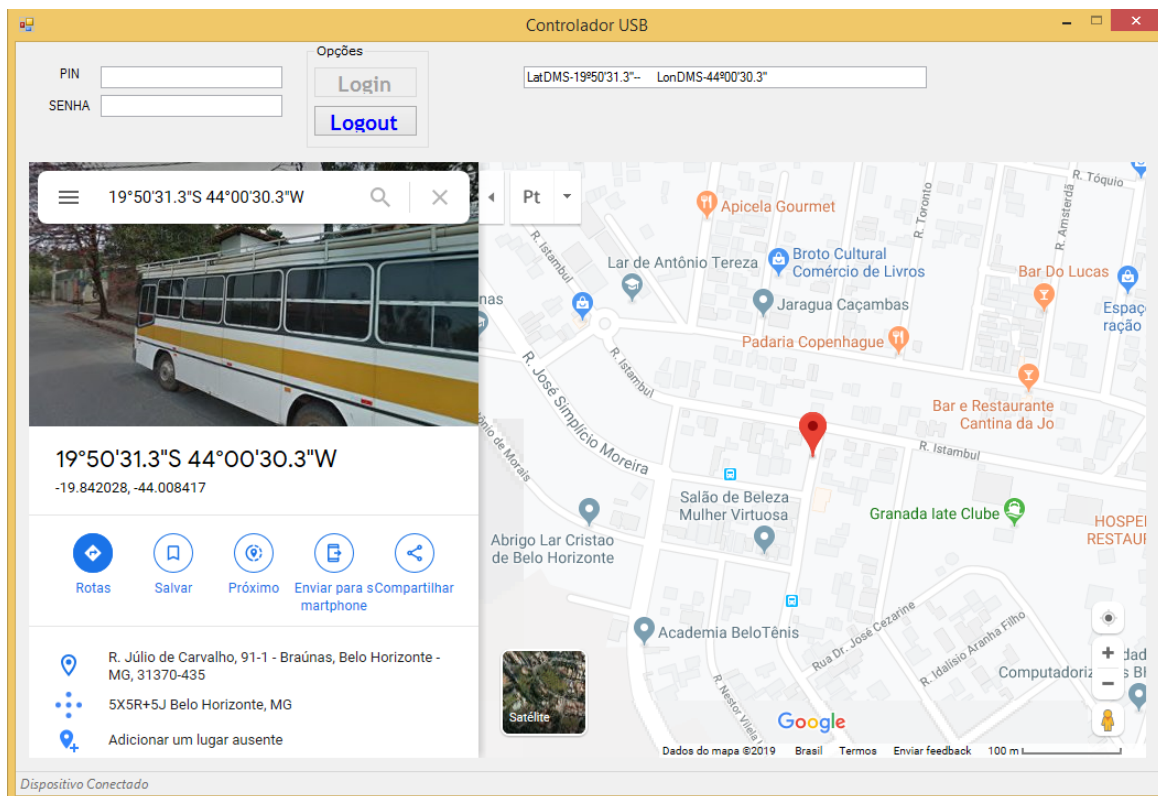


Figura 4.10: *Login* efetuado, tendo-se início o rastreamento.

Google Maps, é possível se acessar todas as funcionalidades deste, como navegar pelas redondezas ou até mesmo obter a visão do *Google Earth*.

Segue-se nas figuras 4.11 e 4.12 as estruturas feitas para comportar os dispositivos desenvolvidos. Estas estruturas foram feitas em MDF (*Medium-Density Fiberboard*).

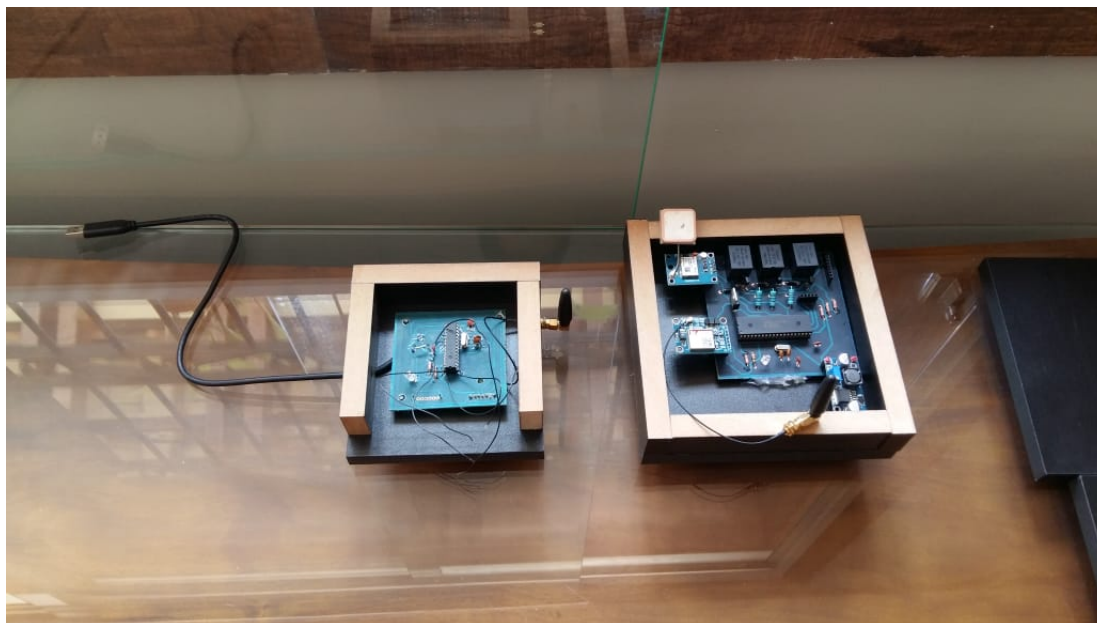


Figura 4.11: Dispositivos desenvolvidos comportados por uma estrutura de MDF.

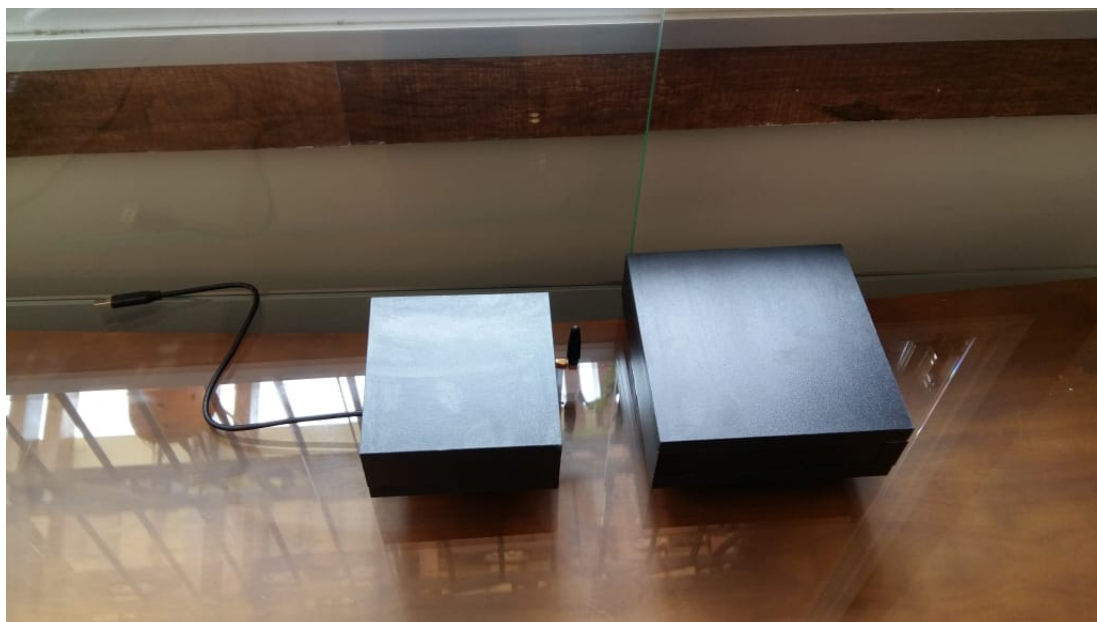


Figura 4.12: Dispositivos desenvolvidos comportados por uma estrutura de MDF.

Por fim, pôde-se realizar um teste com o dispositivo embarcado instalado em um veículo, onde o mesmo atua diretamente no transformador da ignição de um veículo. Para se visualizar o funcionamento dos protótipos, foram gravados vídeos com os testes

feitos; sendo dispostos no sítio *Youtube*. Segue-se os endereços para os vídeos:

- “<https://www.youtube.com/watch?v=ZjSAm4aZ1k8t=133s>”.
- “<https://www.youtube.com/watch?v=DhWWO9lHnQA>”.

Considerações Finais

No presente capítulo foram tratadas as conclusões do trabalho.

5.1 Conclusões

Neste trabalho pôde-se tratar de assuntos concernentes à eletrônica digital, um pouco de eletrônica analógica, e, programação.

Diante de alguns problemas que surgiram, pôde-se perceber que é um desafio se elaborar um dispositivo eficiente com um baixo custo. Apesar do sistema projetado ter se mostrado funcional, ele deixou a desejar em alguns pontos, como o longo tempo para estabelecer as conexões à rede GSM e GPS, por exemplo.

Acredita-se, por fim, que o conteúdo deste trabalho seja útil para o desenvolvimento de projetos similares.

Apêndice **A**

Circuitos Esquemáticos e Tabelas

Item/Processo	Efeito de Falha	G	Causa da Falha	O	Meio de Detecção	D	Índice de Risco	Medida Preventiva do Desenvolvedor
Módulo GSM	Falha no envio ou recepção de dados	8	Área com má cobertura GSM	4	Ausência de retorno ao se enviar um comando SMS	10	320	Escolher antenas com ganho adequado
		8	Danificação do hardware	4		4	128	Projetar hardware robusto
Atuação no veículo	Atuadores trocam de estado lógico sem requisição do usuário	10	Falha de software	2	Observação do comportamento inesperado do dispositivo	2	40	Investir em qualidade de programação
		10	Danificação do hardware	4		4	160	Projetar hardware robusto
Atuação no veículo	Atuadores não trocam de estado lógico tendo sido requisitado pelo usuário	8	Falha de software	5	Observação do comportamento indevido do dispositivo	2	80	Investir em qualidade de programação
		8	Danificação do hardware	4		4	128	Projetar hardware robusto
Comunicação USB	Dispositivo receptor GSM não identificado pelo computador	7	Falha de software	3	Observação através da mensagem de erro no computador	4	84	Investir em qualidade de programação
		7	Danificação do hardware	5		3	105	Investir em conectores de qualidade
Leitura dos sensores no veículo	Leitura incorreta dos sensores binários no veículo	8	Falha de software	5	Testes dos sensores comparando-se a medição com o estado esperado	2	80	Investir em qualidade de programação
		8	Danificação do hardware	4		4	128	Projetar hardware robusto
Programa no computador pessoal	Travamento ou comportamento inesperado	7	Falha de software	4	Observação de erros na execução do programa	3	84	Investir em qualidade de programação
		10		2	Observação do comportamento indevido do dispositivo	9	180	Estudar estratégias de proteção em segurança da informação
Proteção contra acesso não autorizado	Permitir acesso de pessoas não autorizadas	10	Erro de software	2	Identificação de comandos recebidos e que não foram enviados pelo cliente	9	180	Estudar estratégias de proteção em segurança da informação
		10		2				
Proteção contra acesso não autorizado	Não permitir acesso de pessoas autorizadas	8	Perda da senha pelo usuário	5	Observação do usuário	10	400	Estabelecer um meio de recuperação de senha
Instalação do dispositivo embarcado	Conectar circuito à bateria reversamente	7	Desatensão do instalador	5	Observação do dispositivo danificado	10	350	Utilizar diodos de proteção tensão reversa

Figura A.1: Tabela de análise de riscos FMEA.

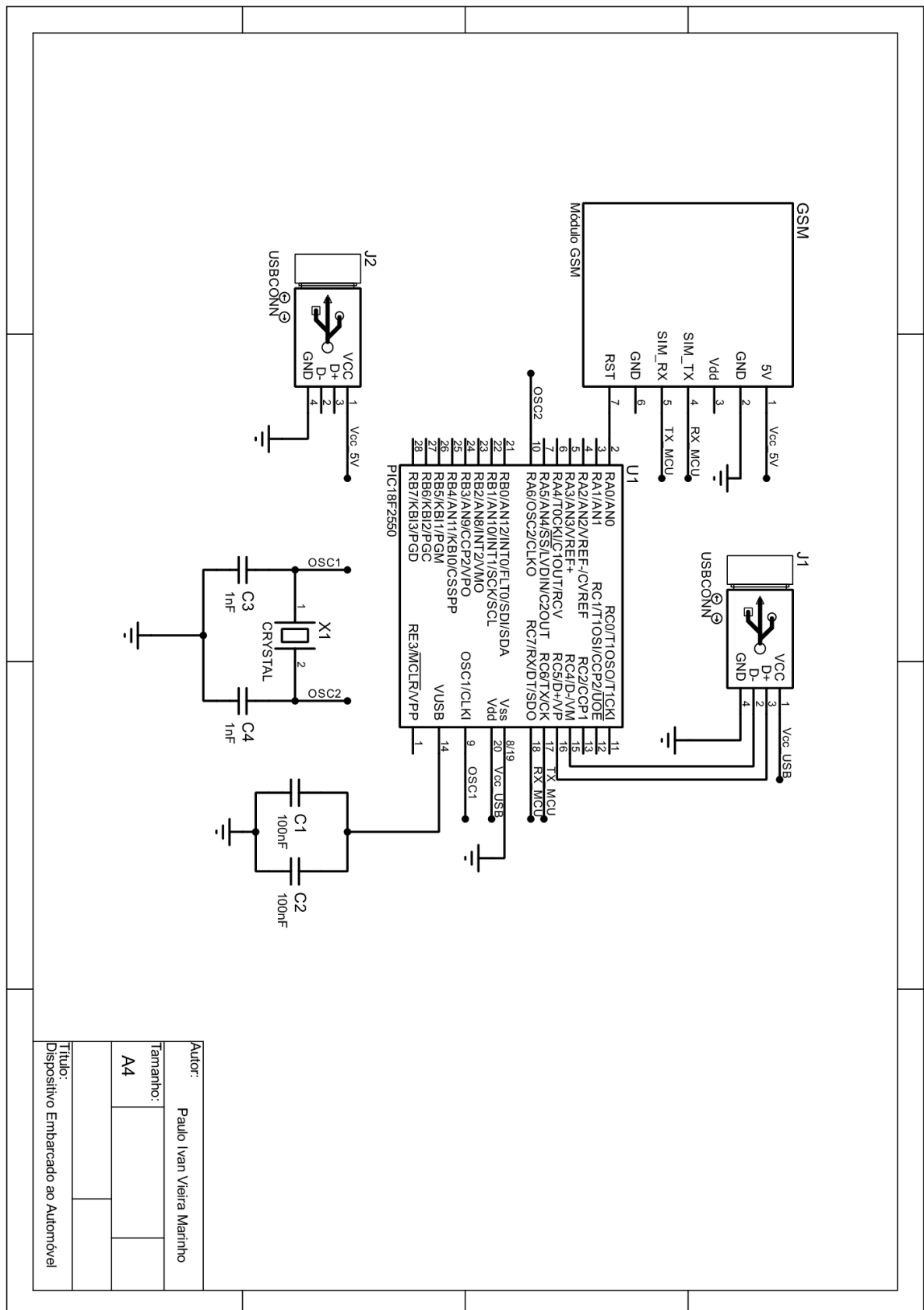


Figura A.3: Esquemático do Dispositivo de Monitoramento no Projeto Lógico.

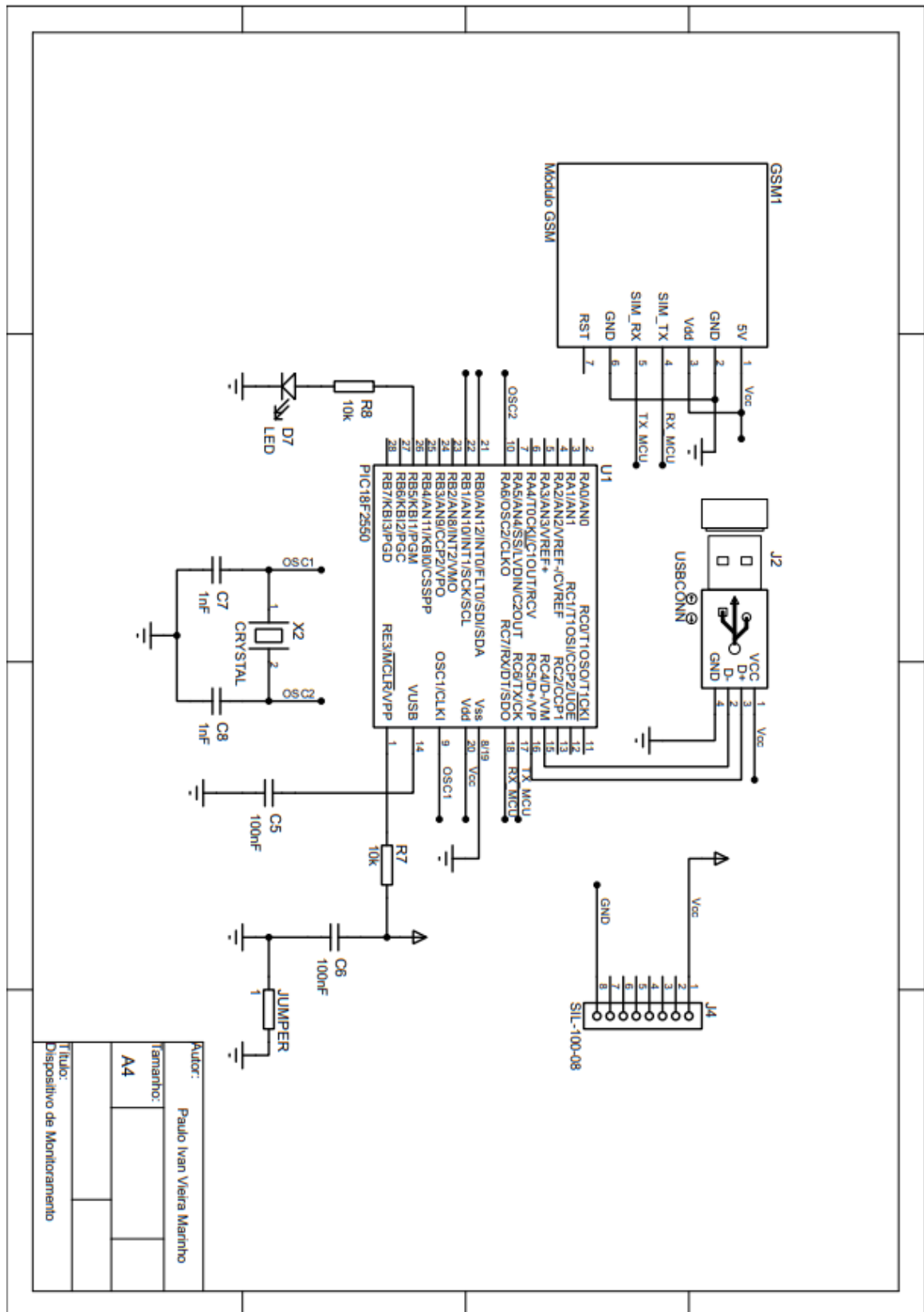


Figura A.5: Esquemático do Dispositivo de Monitoramento alterado após o Projeto Físico.

Referências

- AHMED, A. *Eletrônica de Potência*. 1th.ed. [S.l.]: Prentice Hall, 2000.
- CAVALCANTI, F. R. P. *et al. Comunicação Móvel Celular*. 1th.ed. [S.l.]: Elsevier, 2018.
- EL-RABBANY, A. *Introduction to GPS: the global positioning system*. [S.l.]: Artech House, 2002.
- FOLHA. *Brasil tem 1 roubo ou furto de veículo a cada minuto; Rio lidera o ranking*. Disponível em: <<https://www1.folha.uol.com.br/cotidiano/2017/10/1931061-brasil-tem-1-roubo-ou-furto-de-veiculo-a-cada-minuto-rio-lidera-o-ranking.shtml>>. Acessado em Março 2019.
- GUARD, U. C. *NAVSTAR GPS USER EQUIPMENT INTRODUCTION*. Disponível em: <<http://www.navcen.uscg.gov/pubs/gps/gpsuser/gpsuser.pdf>>. Acessado em Maio de 2019.
- HOFMANN-WELLENHOF, B.; LICHTENEGGER, H.; WASLE, E. *GNSS Global Navigation Satellite Systems: gps, glonass, galileo, and more*. [S.l.]: Springer Science Business Media, 2007.
- KUROSE, J. F.; ROSS, K. W. *Redes de Computadores e a Internet*. 5th.ed. [S.l.]: Pearson, 2010.
- MANGLA, N. *et al.* A GPS-GSM Predicated Vehicle Tracking System, Monitored in A Mobile App based on Google Maps. *2017 International Conference on Energy Communication Data Analytics and Soft Computing (ICECDS)*, [S.l.], 2017.
- MONICO, J. F. G. Posicionamento pelo NAVSTAR-GPS Descrição, fundamentos e aplicações. , [S.l.], 2000. Presidente Prudente.
- MUKHTAR, M. GPS based Advanced Vehicle Tracking and Vehicle Control System. *International Journal of Intelligent Systems and Applications*, [S.l.], v.03, p.1–12, 02 2015.

-
- REDL, S. M.; WEBER, M. K.; OLIPHANT, M. W. *An Introduction to GSM*. [S.l.]: Artech House, 1995.
- R.RAMANI *et al.* Vehicle Tracking and Locking System Based on GSM and GPS. *International Journal of Intelligent Systems and Applications*, [S.l.], 09 2013.
- SAE. *Fault/Failure Analysis Procedure (STABILIZED Feb 2018) ARP926C*. Disponível em:<<https://www.sae.org/standards/content/arp926c/>>. Acessado em Junho de 2019.
- SCHILD, H. *Java - A Referência Completa*. 8th.ed. [S.l.]: Editora Alta Books, 2014.
- SILVA, R. A. *Programando Microcontroladores PIC Linguagem C*. [S.l.]: Ensino Profissional, 2006.
- SIMCOM. *SIM800 Series AT Command Manual*. Disponível em:<<https://simcom.ee/documents/?dir=sim800>>. Acessado em Maio de 2019.
- SVERZUT, J. U. *Redes GSM, GPRS, EDGE e UMTS Uma evolução a caminho da terceira geração (3G)*. [S.l.]: Érica, 2005.
- TANENBAUM, A. S. *Computer Networks*. 4th.ed. [S.l.]: Prentice Hall, 2002. Vrije Universiteit, Amsterdam, Holanda.