
SISTEMA EMBARCADO SIMPLES PARA LOCALIZAÇÃO E MONITORAMENTO MÓVEL BASEADO EM GPS

Mário E. S. Júnior

Wilian Soares Lacerda

Universidade Federal de Lavras (UFLA)

Resumo

Este trabalho apresenta a construção de um sistema embarcado simples que utiliza a tecnologia GPS para obter a posição do dispositivo no globo terrestre e monitorar dados como velocidade e hora. Essas informações são mostradas em um *display* e podem ser armazenadas na memória do dispositivo. Para descarregar os dados, o dispositivo é conectado via porta serial RS232 com um computador onde um *software* gerencia essa conexão e salva os dados para serem visualizados futuramente. Os dados podem ser exportados no formato KML para o *software Google Earth* em que os pontos são plotados demonstrando o caminho realizado pelo usuário.

Palavras-chave: Sistema embarcado. GPS. KML.

1 INTRODUÇÃO

O GPS (*Global Positioning System*) é um sofisticado sistema de navegação, baseado numa rede de satélites, desenvolvido pelo Departamento de Defesa dos Estados Uni-

dos, que permite a localização de uma área de interesse, em qualquer ponto da terra, com alta precisão. O GPS começou a ser desenvolvido na década de 1970, mas se tornou totalmente funcional somente em 1995. A aplicação inicial do projeto era somente para fins militares, mas foi aberto ao uso civil, assim como a internet (ALBUQUERQUE; SANTOS, 2003).

Nos dias de hoje, a tecnologia GPS é aplicada em várias áreas, como sistema de navegação aéreo e terrestre, cartografia, geodésia, topografia, georreferenciamento, agricultura de precisão, controle do movimento de placas tectônicas, entre outras. Com a popularização do GPS, novas aplicações que fazem uso dessa tecnologia e oferecem integração com ferramentas atuais para edição e visualização dos dados, como é a proposta deste trabalho, são bem aceitas pela sociedade.

Este trabalho teve como objetivo a construção de um sistema embarcado (WILSHURST, 2007) de localização e monitoramento baseado na tecnologia GPS, mas de montagem simples e com componentes de fácil aquisição e baixo custo. As informações obtidas por ele são armazenadas no mesmo sistema, para que possam ser utilizadas posteriormente de acordo com as necessidades do usuário. Nas seções que se seguem são descritas as informações necessárias para desenvolvimento do sistema embarcado proposto, bem como os resultados obtidos.

2 GLOBAL POSITIONING SYSTEM

O GPS utiliza 24 satélites artificiais no total para assegurar uma cobertura global. Para poder ter acesso à tecnologia é necessário ter um módulo receptor GPS. A função de um receptor GPS é receber o sinal de quatro ou mais desses satélites, determinar a distância para cada um e utilizar essa informação para deduzir sua própria posição.

O GPS utiliza o conceito de Tempo de Chegada (Time Of Arrival - TOA) para determinar a posição do receptor. Esse conceito consiste em medir o tempo que um sinal emitido por um emissor, em uma posição conhecida, demora a alcançar o receptor (GPS) que se encontra de posse do usuário. Esse intervalo de tempo, denominado tempo de propagação do sinal, é então multiplicado pela velocidade de propagação do sinal obtendo-se a distância emissor-receptor. Através do tempo de propagação do sinal emitido por vários satélites em posições conhecidas, o receptor pode determinar a sua própria posição.

São necessários um conjunto de quatro satélites no mínimo para que o sistema funcione corretamente. Os três primeiros satélites são responsáveis por fazer a triangulação do sinal, e assim encontrar sua posição no globo. O quarto satélite é utilizado como auxiliar para determinar o tempo preciso em que ocorrem as emissões, evitando assim que o receptor use um relógio atômico para determinação do tempo. Desse modo, um sistema de posicionamento global pode averiguar, de forma segura, a posição absoluta de um certo usuário.

O sistema GPS pode fornecer precisa capacidade de navegação tridimensional, em qualquer parte da Terra, mesmo para usuários submetidos a alta dinâmica. Velocidade e altitude também podem ser obtidas utilizando técnicas diferenciais e com minimização de erros.

2.1 Sistema de coordenadas geográficas

Os sistemas de coordenadas são necessários para expressar a posição de pontos sobre uma superfície, seja ela um elipsoide, seja uma esfera ou um plano (IBGE, 1998). É com base em determinados sistemas de coordenadas que é descrito geometricamente a superfície terrestre. Para elipsoides e esferas, usualmente é empregado um sistema de coordenadas cartesiano curvilíneo baseado em paralelos e meridianos (PERONI, 2004).

Utilizando os paralelos e meridianos, foi criado o sistema de latitude e longitude, em que a latitude é o afastamento, medido em graus, da linha do Equador, a um ponto qualquer da superfície terrestre. Ela vai de 0° a 90° e em direção ao norte ou ao sul. E a longitude é o afastamento, medido em graus, do meridiano de Greenwich, a um ponto qualquer da superfície terrestre. Ela vai de 0° a 180° e pode ser na direção leste ou oeste. Assim, qualquer ponto no globo pode ser indicado através dessas duas coordenadas. A Figura 1 mostra a disposição da latitude e da longitude ao longo do globo.

As distâncias são indicadas em graus e não em quilômetros, porque se trata de medições sobre uma esfera, que é a forma aproximada da Terra.

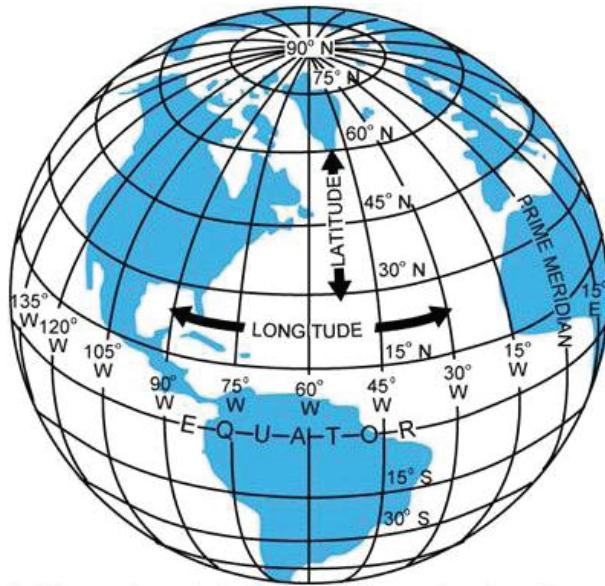


Figura 1 Latitude e longitude ao longo do globo.

Fonte: <<http://jeuriibeirogeografia.blogspot.com.br>>

2.2 Visualização de coordenadas

Para a visualização de coordenadas geográficas, podem ser utilizados programas como o Google Earth (2015), que possui fotos de satélite e mapas de todo o globo terrestre, permitindo visualizar pontos, criar caminhos e realizar passeios virtuais baseado em coordenadas geográficas inseridas pelo usuário. Para tornar a ferramenta mais flexível, a Google disponibiliza ao usuário uma linguagem de marcação de texto, o KML (Keyhole Markup Language) (GOOGLE, 2015), que permite escrever arquivos compatíveis com o Google Earth.

2.3 Módulo receptor GPS

O ME-1000RW é um módulo receptor GPS com antena acoplada (ME, 2012). A antena é conectada ao receptor por meio de um LNA (Amplificador de Baixo Ruído). O receptor tem 51 canais de aquisição e 14 canais de rastreamento que são capazes de receber sinais de até 65 satélites GPS e informar a posição e o tempo

precisos para ser lido na porta UART (*Universal Asynchronous Receiver/Transmitter*). O equipamento tem baixo consumo e a faixa de tensão de alimentação suportada vai de 3,3 a 6,0V. O conector possibilita a saída tanto em nível LVTTL quanto em nível RS232. A Figura 2 apresenta a face frontal do módulo ME-1000RW.

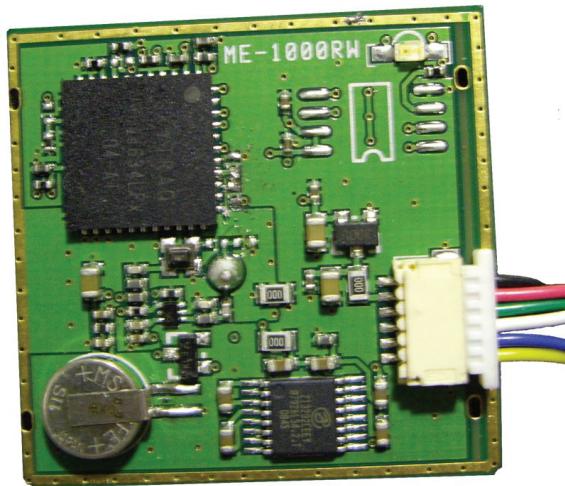


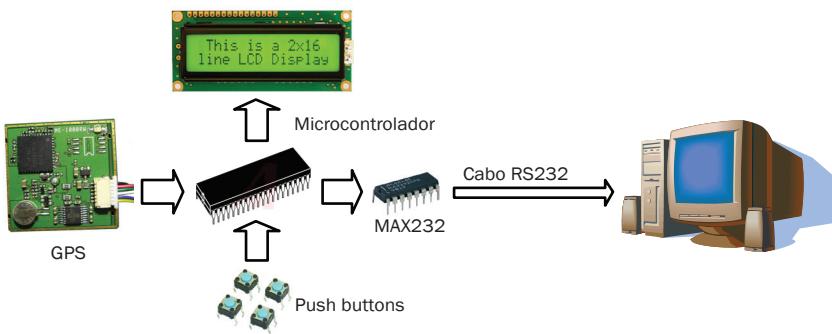
Figura 2 Face frontal do módulo ME-1000RW.

Fonte: ME (2012).

As informações enviadas via comunicação serial pelo módulo GPS estão definidas em cinco tipos de sentença de dados: GPGGA, GPGSA, GPRMC, GPVTG, GPGSV. De acordo com ME Componentes e Equipamentos Eletrônicos Ltda., o formato dos dados transmitidos pelo módulo GPS via comunicação serial é baseado na especificação da interface da National Marine Electronics Association's (NMEA 0183 ASCII) e pode ser obtido em <<http://www.nmea.org>>.

3 SISTEMA EMBARCADO PROPOSTO

Um diagrama geral simplificado do sistema proposto pode ser visto na Figura 3.

**Figura 3** Diagrama geral do sistema proposto.

Fonte: Elaborado pelo autor

O sistema contém um microcontrolador, um módulo receptor de GPS, um grupo de *push buttons*, um *display* tipo LCD, e uma interface para padrão RS232. O LCD mostra os dados obtidos do módulo GPS em tempo real. Esses dados oferecidos pelo sistema ao usuário são: longitude, latitude, altitude, velocidade e hora. O sistema também apresenta, via *display*, a quantidade de dados salvos na memória. Os *push buttons* permitem ao usuário escolher as ações fornecidas pelo sistema.

O armazenamento dos dados na memória é feito de forma manual, acionando um dos *push buttons*, ou de forma automática, configurando o sistema para realizar o armazenamento em um período de tempo específico. Um sistema de rotas é oferecido, para possibilitar a separação lógica dos dados, quando eles são descarregados no computador.

Um *software* também foi desenvolvido (plataforma Windows) para receber em um computador (tipo PC) os dados armazenados pelo sistema embarcado por meio da interface RS232. Esse *software* permite o salvamento dos dados tanto em formato texto, para que sejam analisados e trabalhados futuramente, quanto no formato compatível com o programa Google Earth, em que os dados são tratados como um caminho realizado pelo usuário, e cada ponto do caminho oferece as informações coletadas.

3.1 Desenvolvimento do *hardware*

O diagrama eletrônico completo do sistema embarcado proposto pode ser visto na Figura 4. Para o módulo GPS foi adotado o modelo ME-1000RW, descrito anteriormente, pelo fato de atender as necessidades do sistema, oferecendo uma boa quantidade de dados relevantes, como: latitude, longitude, altitude, velocidade e hora.

Para a comunicação serial, o módulo MAX232N (TEXAS INSTRUMENTS MAX232, 2004) foi utilizado, pois a maioria dos computadores possuem uma interface compatível com o padrão RS232. Embora sua taxa de transmissão de dados não seja alta, ela atende plenamente a necessidade do projeto já que a quantidade de dados a ser transmitida não é muito grande.

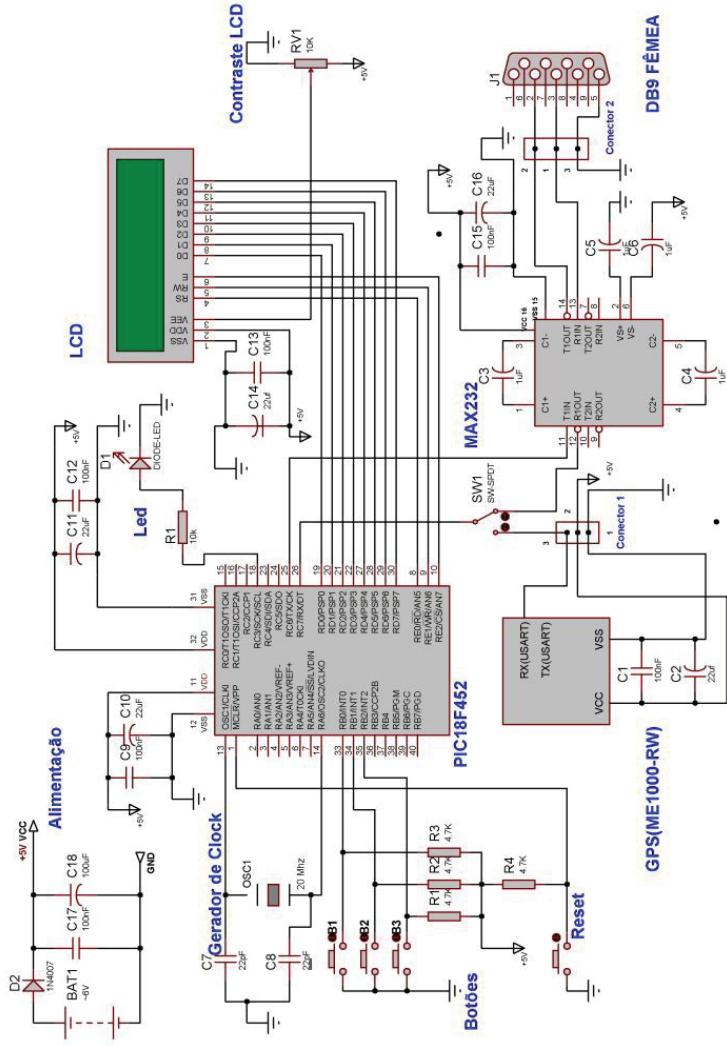


Figura 4 Esquema eletrônico do sistema embarcado.

Fonte: Elaborado pelo autor

Como *display* foi utilizado o modelo de LCD MGD1602B (POWERTIP, 2005). Embora seja um *display* de 16 colunas por duas linhas, ele atende bem as exigências do projeto, pois a quantidade de dados a ser apresentada ao usuário é baixa. Um *trim-pot* é utilizado para ajustar o contraste.

Para o microcontrolador foi escolhido o PIC18F452 (MICROCHIP, 2006) do fabricante Microchip, pelo fato de possuir comunicação serial, *timer*, suporte à interrupção, quantidade de memória adequada para armazenamento, um número de pinos suficiente para o controle de todos os dispositivos do sistema, além do baixo custo e facilidade de aquisição.

Levou-se em consideração que o sistema é móvel para especificar a alimentação do circuito, por isso são utilizadas quatro pilhas de 1,5 V totalizando 6 Volts. Como a alimentação de todos os dispositivos deve ser próxima a 5 V, um circuito com um diodo em série com a fonte é necessário para realizar uma queda de potencial de aproximadamente 0,7 V, deixando a tensão mais próxima de 5 V. Para evitar ruídos na alimentação dos dispositivos do sistema, capacitores eletrolíticos e de disco são utilizados para filtrar os mesmos.

Uma pequena chave HH é utilizada para que o pino de recepção serial do microcontrolador (RX) seja compartilhado tanto pelo módulo GPS, como pelo *chip* MAX232. A chave serve para alternar a conexão do microcontrolador com os dois módulos. Também um pequeno LED é utilizado para mostrar ao usuário que o sistema está executando certas ações que não devem ser interrompidas.

Para o desenvolvimento do *software*, que é gravado no microcontrolador, foi utilizado o compilador MikroC (MIKRO, 2006) que é gratuito. Por utilizar C como linguagem nativa, ele retira grande parte do trabalho das mãos do programador, facilitando assim o desenvolvimento do projeto.

Os dados obtidos do GPS são apresentados no *display* para o usuário, assim que esses são obtidos. Como o *display* utilizado é do tipo 2 linhas por 16 colunas, o usuário pode alternar as informações que são visualizadas por meio de um *push button*.

Como o sistema desenvolvido é móvel e alimentado por baterias, o consumo de energia é uma característica que deve ser levada em conta, por isso o *software* desenvolvido foi construído baseando-se principalmente na geração de interrupções para que o microcontrolador execute as tarefas somente quando for necessário.

Ainda com ênfase no custo energético, as rotinas para ler dados do GPS e as rotinas para armazenar na memória são executadas por interrupção. O período de tempo para armazenamento dos dados na memória pode ser configurado pelo usuário quando o sistema é iniciado, e o tempo para a leitura do GPS é definido diretamente no código fonte.

Para armazenamento dos dados lidos do GPS é utilizada a memória *flash* interna do microcontrolador que no caso do PIC18F452 é de 32.768 *bytes*. Como essa me-

mória também é utilizada para o armazenamento do programa, ela foi dividida em dois blocos, um para o programa e um para os dados.

A memória EEPROM é utilizada para armazenar algumas variáveis de controle como o número de registros e o número de rotas existentes, pois essas precisam estar disponíveis mesmo que o sistema seja desligado. Como as rotinas do compilador MikroC para salvar na EEPROM só permitem salvar 1 byte de cada vez, isso limita o valor máximo dessas variáveis em 255. Levando isso em conta, o número máximo de pontos que o sistema pode salvar é de 254, pois o número 255 (FF em hexadecimal) é o valor interpretado no código como posição de memória vazia.

Para uma maior flexibilidade do sistema, um recurso de rotas é implementado para que o usuário possa armazenar caminhos distintos caso seja necessário, portanto, toda vez que o sistema é iniciado, uma nova rota é criada. Outra medida, para aumentar a flexibilidade, é a possibilidade de o usuário desabilitar o recurso de salvamento automático dos pontos, podendo fazer isso manualmente quando julgar necessário.

3.2 Desenvolvimento do software do sistema embarcado

Para facilitar o desenvolvimento e o reuso do código fonte, ele foi modularizado. Portanto, as principais tarefas que o microcontrolador precisa realizar foram divididas em funções.

Ao ligar o sistema, uma mensagem de boas-vindas é mostrada no *display*, e a rotina para configurar o tempo de armazenamento na memória é chamada. Após a configuração, caso o número de registros na memória não seja máximo, uma nova rota é criada. Na próxima etapa, há a execução da rotina para a leitura de dados do GPS e armazenamento dos mesmos na memória.

Logo depois de salvar o primeiro registro na memória, as interrupções externas (as interrupções que ocorrem ao apertar um *push button*) e de tempo são configuradas e ativadas. A interrupção de tempo é configurada para gerar uma interrupção a cada segundo, o que torna necessário o uso de variáveis “contadoras de tempo”, para que o sistema execute ações em um período determinado.

Após a configuração das interrupções, o sistema entra em um laço de repetição infinito, que é responsável por deixar o microcontrolador em execução contínua. Dentro desse laço estão implementadas as seguintes ações:

- Mostrar a latitude e a longitude no *display*;
- Mostrar a altitude e a velocidade no *display*;
- Mostrar o número de registros na memória e a hora no *display*;
- Descarregar os dados via porta Serial;

- Ler dados do GPS, ajustar o horário, converter a velocidade e salvar os mesmos na memória;
- Ler dados do GPS, ajustar o horário e converter a velocidade.

Essas ações são controladas por uma variável condicional, ou seja, quando a variável assumir determinado valor, a ação é executada. O valor dessa variável é atribuído de acordo com o tipo de interrupção que ocorrer. Os tipos de interrupções possíveis e as ações que o sistema realiza ao recebê-la podem ser vistos a seguir:

1. Apertar o *push button* “um” gera uma interrupção que altera os dados a serem mostrados no *display*;
2. Apertar o *push button* “dois” gera uma interrupção que lê os dados do GPS, trata os mesmos, e salva na memória do microcontrolador;
3. Apertar o *push button* “três” gera uma interrupção que descarrega os dados via Serial.

Se acontecer uma interrupção de tempo e o contador de tempo para armazenamento da memória “estourar”, os dados são lidos do GPS, tratados, e salvos na memória. Se acontecer uma interrupção de tempo e o contador de tempo para ler do GPS “estourar”, os dados são lidos do GPS e apenas tratados.

Após realizar qualquer ação, a variável condicional recebe um valor com o qual o sistema não consegue executar nenhuma ação, deixando o sistema em estado de “espera”. Vale a pena ressaltar que no caso de o usuário atribuir o valor zero ao contador de tempo para armazenamento da memória, a opção de salvar os dados via interrupção de tempo é desativada, e o usuário poderá somente salvar os dados manualmente, por meio do *push button* “dois”.

3.3 Software para o computador

O *software* desenvolvido para o computador tem como finalidade receber os dados enviados serialmente pelo sistema embarcado e apresentá-los de uma forma legível ao usuário, além de salvar os mesmos em arquivo para uso futuro.

No desenvolvimento do *software* para o computador foi utilizada a linguagem Microsoft C# (MICROSOFT, 2013) por ser uma linguagem de alto nível, e com bom suporte a manipulação de porta serial e criação de interface gráfica. Como ambiente de desenvolvimento foi utilizado o Microsoft Visual C# 2010 Express. O *software* oferece uma interface intuitiva, com botões para realizar as ações, assim como informações sobre a ação que está sendo executada.

Para uma visualização intuitiva dos dados, esses podem ser salvos no formato de arquivo KML compatível com o *software Google Earth*, porém, é importante salvar os dados de uma maneira que o usuário possa trabalhar facilmente com eles. Assim, há a opção de os dados também serem salvos na forma de texto.

O arquivo KML é construído para que cada rota recebida do sistema tenha seus pontos interligados indicando o caminho percorrido, e cada ponto contém como descrição a altitude, velocidade e a hora naquele ponto.

Os dados de latitude e longitude obtidos do GPS estão no formato ddd.ddddd (decimal de graus), que é incompatível com o formato de dados ddmm.mmmm (graus com minutos decimais) utilizado pelo Google Maps. Então, os mesmos devem ser convertidos. Para isso, deve-se dividir a parte dos minutos por 60 e após isso somar ao resultado a parte dos graus. Além disso, o resultado deve ser multiplicado por -1 caso o hemisfério da coordenada seja sul ou leste.

A implementação do *software* pode ser dividida em três partes: interface com o usuário, comunicação serial e exportação dos dados. A comunicação serial é feita por meio da classe PortaSerial, que utiliza a classe nativa da linguagem *SerialPort*, que contém métodos e atributos para configuração e controle de comunicação serial. Para a exportação dos dados em formato de arquivo foi implementada a classe Exportar.

Para o desenvolvimento da interface gráfica com o usuário foi utilizado o recurso *Windows Forms*, presente no *Microsoft Visual Studio C# 2010 Express*. Esse recurso permite a construção de interface por meio de um *toolbox*, no qual se pode selecionar e posicionar o recurso que deseja. A interface projetada conta com uma área para impressão de texto, responsável por apresentar ao usuário todos os *feedbacks* durante a execução do programa. A interface ainda disponibiliza quatro botões virtuais para interação com o usuário. Os botões e suas respectivas ações realizadas são listados a seguir:

1. Botão “Conectar”: realiza a conexão com a porta serial, e recebe os dados enviados pelo sistema, imprimindo os mesmos na tela ao recebê-los.
2. Botão “Limpar”: limpa as mensagens impressas na tela do programa.
3. Botão “Exportar para Txt”: reorganiza os dados recebidos de uma forma mais legível ao usuário, além de salvá-los em um arquivo no formato texto, com o nome e local escolhidos pelo usuário.
4. Botão “Exportar para Kml”: reorganiza e formata os dados de forma que eles sejam compatíveis com o formato KML; constrói o arquivo KML e salva os mesmos com o nome e local escolhido pelo usuário.

4 RESULTADOS

Após o sistema embarcado ser montado, e o *software* para o PIC18F452 ser gravado nele, foi obtida a foto do sistema completo que pode ser vista na Figura 5.

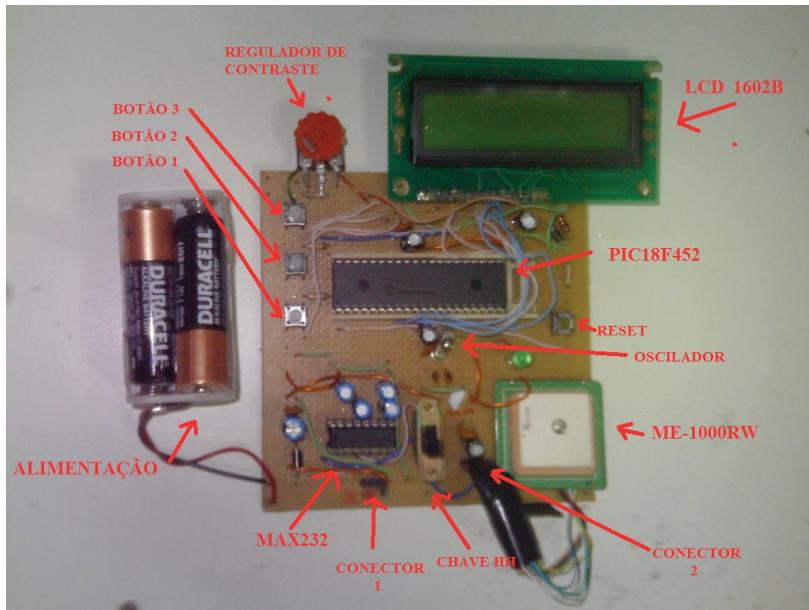
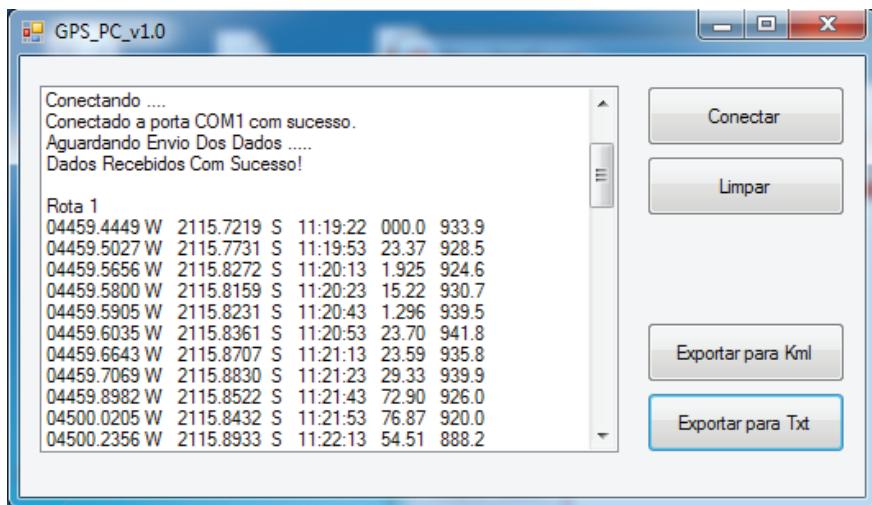


Figura 5 Sistema proposto montado.

Fonte: Elaborado pelo autor

A Figura 6 mostra a interface do *software* desenvolvida para o computador, para receber os dados recebidos pelo sistema. Dois testes foram realizados para verificar o funcionamento do sistema. O primeiro teste teve como objetivo verificar o comportamento para cada requisição do usuário. Já o segundo testou o sistema em uma situação para que o mesmo armazenasse vários pontos automaticamente, que pudessem ser utilizados posteriormente.

**Figura 6** Interface do software para o computador recebendo os dados.

Fonte: Elaborado pelo autor

No teste em ambiente real o sistema foi verificado em uma situação em que o usuário desejava armazenar os pontos coletados automaticamente para uso futuro. O teste foi realizado na cidade de Lavras, em Minas Gerais, no qual o usuário, utilizando um veículo automotor com sistema proposto, percorreu algumas ruas da cidade e voltou ao ponto inicial. Para o teste foi configurado o período de armazenamento na memória em 10 segundos. Após o término do teste, os dados armazenados no sistema foram descarregados no computador e os pontos salvos foram recebidos pelo *software* como mostra também a Figura 6.

Ao receber os dados, esses podem ser exportados tanto para o formato txt quanto para o formato KML. A estrutura interna do arquivo salvo no formato txt pode ser vista na Figura 7. Após a criação do arquivo no formato KML, ele foi aberto no programa *Google Earth* e o resultado pode ser visto na Figura 8.

The screenshot shows a Windows Notepad window with the title bar 'teste1 - Notepad'. The menu bar includes 'File', 'Edit', 'Format', 'View', and 'Help'. The main content area displays a list of data rows, each starting with 'Rota 1' followed by a series of numbers separated by spaces. The data columns represent coordinates and time.

Rota 1						
04459.4449	W	2115.7219	S	11:19:22	000.0	933.9
04459.5027	W	2115.7731	S	11:19:53	23.37	928.5
04459.5656	W	2115.8272	S	11:20:13	1.925	924.6
04459.5800	W	2115.8159	S	11:20:23	15.22	930.7
04459.5905	W	2115.8231	S	11:20:43	1.296	939.5
04459.6035	W	2115.8361	S	11:20:53	23.70	941.8
04459.6643	W	2115.8707	S	11:21:13	23.59	935.8
04459.7069	W	2115.8830	S	11:21:23	29.33	939.9
04459.8982	W	2115.8522	S	11:21:43	72.90	926.0
04500.0205	W	2115.8432	S	11:21:53	76.87	920.0
04500.2356	W	2115.8933	S	11:22:13	54.51	888.2
04500.2696	W	2115.9027	S	11:22:23	3.074	886.6
04500.2953	W	2115.9092	S	11:22:43	25.94	891.4
04500.3537	W	2115.9229	S	11:22:53	34.98	897.5
04500.4685	W	2115.8684	S	11:23:13	51.72	898.4
04500.4991	W	2115.7977	S	11:23:23	49.12	887.0
04500.5229	W	2115.7483	S	11:23:43	6.333	877.5
04500.5271	W	2115.7392	S	11:23:53	9.518	880.8
04500.5704	W	2115.6461	S	11:24:13	36.81	885.9

Figura 7 Estrutura interna do arquivo formato txt.

Fonte: Elaborado pelo autor.

O sistema, em alguns casos, demorou bastante tempo após a sua inicialização para produzir dados confiáveis devido à calibração do módulo GPS. Embora o fabricante do modelo alerte que ele demore até 30 segundos para realizar essa etapa, muitas vezes o tempo esperado foi de até 5 minutos. Outro problema encontrado durante os testes foi a estabilização dos dados de altitude e velocidade que, mesmo quando o sistema estava imóvel, demonstrava grande variação.

5 CONCLUSÃO

Este trabalho detalhou todo o processo de construção de um sistema embarcado de rastreamento móvel. Por meio dele, o usuário pode armazenar pontos coletados para que os mesmos possam ser utilizados posteriormente, para traçar rotas ou para qualquer outra atividade, em que as coordenadas geográficas são necessárias. O sistema

proposto é simples, composto por peças de fácil aquisição no mercado e de montagem sem complicações, além de baixo custo. A operação do sistema também é simples, inclusive para baixar os dados coletados e mostrar os resultados em formato gráfico.



Figura 8 Rota gerada a partir do arquivo de formato KML.

Fonte: Elaborado pelo autor.

Embora alguns problemas tenham sido encontrados durante a fase de testes do sistema, os resultados obtidos podem ser considerados satisfatórios para aplicações que não precisem de dados de altíssima precisão, como no caso de rastreamento de veículos. Além disso, como o sistema também oferece informações como velocidade e hora, ele pode ser utilizado por empresas de logística para monitorar e otimizar a rota de veículos e o transporte de mercadorias. O sistema também pode ser utilizado para geração de curvas de nível já que é disponibilizada a informação de altitude.

Para dar continuidade a este trabalho, algumas melhorias são propostas, como: aumentar a capacidade de armazenamento de dados do sistema oferecendo suporte a cartões de memória ou *pendrives*; substituição da interface RS232 pela USB que é mais rápida e mais comum nos computadores atuais; realizar a montagem do sistema em circuito impresso menor; adição de diferentes tipos de sensores para coleta de dados: temperatura, umidade, pressão atmosférica etc.

SIMPLE EMBEDDED SYSTEM FOR LOCATION AND MOBILE MONITORING USING GPS

Abstract

This paper presents the construction of a simple embedded system that uses GPS technology to get the devices' location on the terrestrial globe and monitors data such as speed and time. This information is shown on a display and can be stored in the device memory. In order to download the data, the device is connected to serial port with a computer by RS232 where a software manages this connection and saves the data to be viewed in the future. Data can be exported in KML format for Google Earth software where the points are plotted showing the path taken by the user.

Keywords: Embedded System. GPS. KML.

REFERÊNCIAS

ALBUQUERQUE, P. C. G.; SANTOS, C. C. dos. *GPS para iniciantes*. INPE - Instituto Nacional de Pesquisas Espaciais, 2003. Disponível em: <<ftp://mtc-m12.sid.inpe.br/col/sid.inpe.br/jeferson/2003/06.02.09.16/doc/publicacao.pdf>>. Acesso em: 15 maio 2015.

GOOGLE. Google Earth. Disponível em: <<http://www.google.com/intl/ptPT/earth/learn/>>. Acesso em: 15 maio 2015.

GOOGLE. Introdução à documentação KML. Disponível em: <<https://developers.google.com/kml/documentation/>>. Acesso em: 15 maio 2015.

IBGE - Instituto Brasileiro de Geografia e Estatística. Noções básicas de cartografia. 1998. Disponível em: <ftp://geoftp.ibge.gov.br/documentos/cartografia/nocoes_basicas_cartografia.pdf>. Acesso em: 15 maio 2015.

ME COMPONENTES E EQUIPAMENTOS ELETRÔNICOS Ltda. Módulo GPS com antena acoplada ROM ME-1000RW. 2012. Disponível em: <<http://www.mecomp.com.br/me1000rw.html>>. Acesso em: 15 maio 2015.

MICROCHIP Technology Inc. PIC18FXX2 Datasheet. 2006. Disponível em: <<http://ww1.microchip.com/downloads/en/devicedoc/39564c.pdf>>. Acesso em: 15 maio 2015.

MICROSOFT. Recursos do Visual C#. 2013. Disponível em: <<http://msdn.microsoft.com/ptbr/vstudio/hh341490>>. Acesso em: 15 maio 2015.

MIKRO ELEKTRONIKA. MikroC User's Manual. 2006. Disponível em: <http://www.mikroe.com/pdf/mikroc/mikroc_manual.pdf>. Acesso em: 15 maio 2015.

PERONI, R. *Fundamentos do GPS*. 2004. Disponível em: <<http://goo.gl/844s2w>>. Acesso em: 15 maio 2015.

POWERTIP. PC 1602-Q. 2005. Disponível em: <http://www.datasheetcatalog.com/datasheets_pdf/P/C/1/6/PC1602-L.shtml>. Acesso em: 15 maio 2015.

TEXAS INSTRUMENTS. MAX232. 2004. Disponível em: <http://www.datasheetcatalog.com/datasheets_pdf/M/A/X/2/MAX232.shtml>. Acesso em: 15 maio 2015.

WILMSHURST, T. *Designing embedded systems with PIC microcontrollers: principles and applications*. London: Elsevier, 2007.

Contato

Wilian Soares Lacerda
lacerda@dcc.ufla.br

Tramitação

Recebido em maio de 2015.
Aprovado em fevereiro de 2017