

# 201383 - Microprocessadores e Microcontroladores

## Turma A – Semestre 2018/1

### Relatório Final – GPS rastreador de veículos

Lucas Machado de Moura e Silva

Faculdade do Gama  
Universidade de Brasília - UnB  
Brasília-DF, Brasil  
lucasmmsilva@gmail.com

Gustavo Luiz Monteiro de Oliveira

Faculdade do Gama  
Universidade de Brasília - UnB  
Brasília-DF, Brasil  
gustavomont97@gmail.com

**Resumo**— Um sistema rastreador de veículos é um sistema que permite que os proprietários do veículo consigam saber a localização precisa do mesmo. O projeto em questão visa desenvolver e implementar este sistema de modo que seja uma solução fácil e com ambiente virtual acessível a qualquer usuário.

**Palavras-chave**— rastreador; GPS; coordenadas; veículo.

#### I. INTRODUÇÃO

De acordo com o levantamento realizado pelo Fórum Brasileiro de Segurança no ano de 2016, a cada 1 minuto um brasileiro tem seu carro furtado ou roubado no país. [1] Essa informação que está presente no 10º Anuário Brasileiro de Segurança Pública explica a crescente procura da população por serviços de rastreamento de veículos nos últimos anos. Como comentado anteriormente, o número de furtos e roubos de carros no Brasil é muito elevado. Deixando claro a vulnerabilidade das pessoas à ação dos criminosos. A presença de um sistema de rastreamento, torna as chances de recuperação do veículo em caso de roubo ou furto muito maiores.

O simples fato de saber que no veículo há um rastreador e a localização do mesmo poderá ser verificada em poucos instantes após um possível crime e que poderá auxiliar as

autoridades a efetuar a localização com maior rapidez e segurança, tranquiliza e garante a segurança do proprietário do veículo. [2]

O sistema de rastreamento será composto por dispositivos, relativamente pequenos, que facilitará a instalação no veículo em locais não visíveis. O recebimento das coordenadas para a localização (latitude e longitude) se dará por meio de mensagem SMS para um número de preferência do proprietário do veículo.

#### II. DESENVOLVIMENTO

##### A. REVISÃO BIBLIOGRÁFICA

Para que o sistema de rastreamento funcione efetivamente, é necessário que seja feita a leitura dos dados de latitude e longitude, posteriormente a concatenação com o link do *Google Maps* e o envio de dados pelo serial monitor do MSP430.

Como esses dados serão enviados via SMS, é necessário também que o usuário possua um aparelho celular, desse modo podendo rastrear qualquer objeto. Para isso, também torna-se imprescindível a configuração do modem GSM (*Global System for Mobile communication*), uma tecnologia que tornou-se popular pela facilidade na transferência e recebimento de dados com um alto nível de

confiança e que pode ser utilizado para o envio de SMS de um computador ou celular para rastrear um veículo integrado com GPS.

No projeto, será utilizado o microcontrolador MSP430, cuja arquitetura, combinada com cinco modos de baixa potência, é otimizada para alcançar a extensa vida útil da bateria em aplicações de medição portáteis. O dispositivo possui uma poderosa CPU RISC de 16 bits, registradores de 16 bits e geradores que contribuem para a máxima eficiência do código. [3]

### B. VISÃO GERAL DO SISTEMA

O projeto é dividido basicamente em duas partes principais, que são a parte de rastreamento e uma parte de exibição. A parte de rastreamento é responsável por obter o local do usuário enquanto a parte de controle e exibição é para exibir o local detectado no *Google Maps*. A parte de rastreamento consiste no GPS, microcontrolador e GSM, enquanto a parte de exibição é um aplicativo baseado na web que mostra a localização no *Google Maps*. [4]

Uma terceira parte será adicionada ao projeto, que é a parte do envio de dados via SMS, para um aparelho celular, a fim de poder rastrear qualquer objeto. Para essa etapa, será realizada a configuração do modem GSM de acordo com as suas respectivas características baud rate (taxa de transmissão de dados por segundo) bem como os pinos de configuração TX/RX que variam de acordo com o módulo a ser utilizado.

### C. DESENVOLVIMENTO DO SISTEMA

O esquemático abaixo mostra o processo de design do sistema de rastreamento, que será iniciado escrevendo o código-fonte para a parte de rastreamento e exibindo o módulo. Depois disso, o código será baixado para o microcontrolador. Tanto a codificação para rastreamento e exibição de peças serão testados para garantir que atendam às especificações. Quando a codificação atender às especificações, todo o sistema será testado novamente para garantir que funcione corretamente.

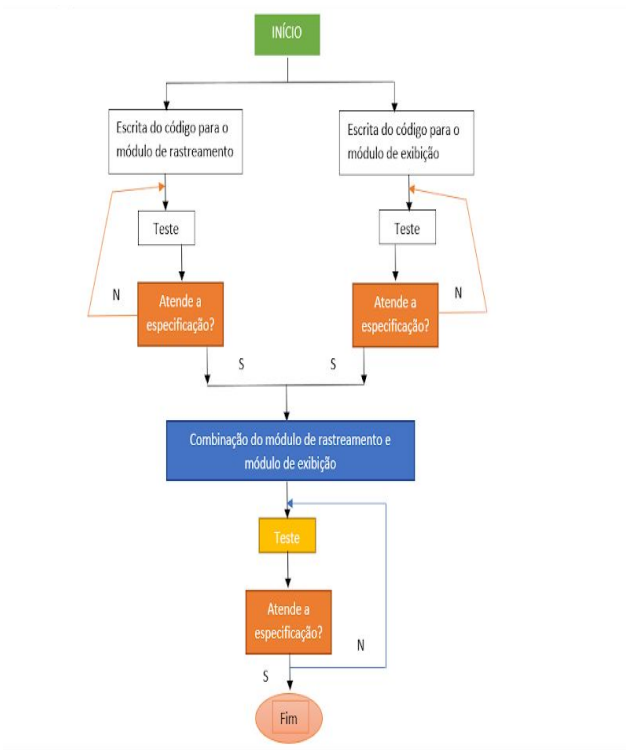


Fig.1: Diagrama do processo de design do sistema rastreador

### D. MÓDULOS UTILIZADOS

- GSM modelo sim800L

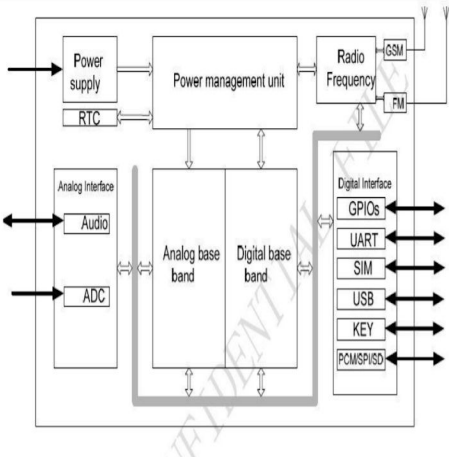


Fig. 2: Diagrama de função do sim800L



Fig. 3: Estrutura do módulo GSM SIM800L

- GPS modelo GY-GPS6MV2 NEO6M



Fig. 4: Estrutura do módulo GPS a ser utilizado

#### E. DESCRIÇÃO DO HARDWARE

##### LISTA DE MATERIAIS:

UNIDADES	COMPONENTE	FABRICANTE
01	MSP430G2553IN20	TEXAS INSTRUMENTS
01	MÓDULO GPS GY-GPS6MV2 NEO6M	UBLOX
01	MÓDULO SIM800L	UBLOX
---	JUMPERS	----

Para a montagem do circuito que se apresenta na Figura 5 a seguir, os únicos materiais necessários além dos módulos externos ao microcontrolador, foram os jumpers

macho-fêmea para ligar a MSP430 ao módulo GPS. O pino TX do módulo GPS foi conectado ao pino P1.1 do MSP e o RX ao pino P1.2 para alimentação do módulo foi utilizado o pino VCC do MSP.

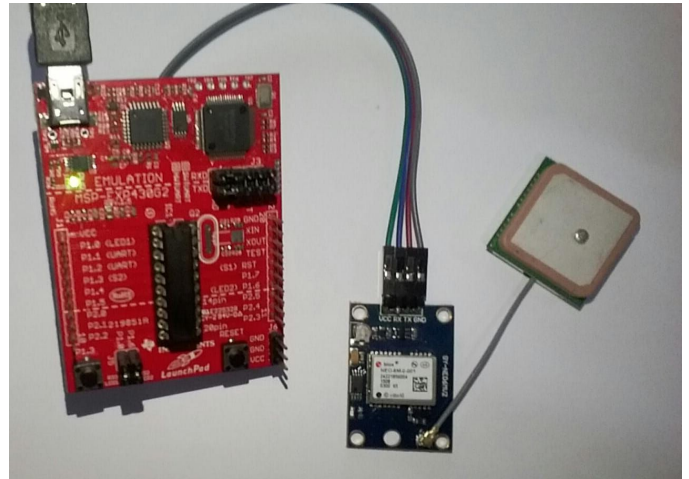


Fig. 5: Estrutura do circuito

#### F. DESCRIÇÃO DO SOFTWARE

O código foi realizado na IDE do Energia, uma plataforma de prototipagem rápida da Texas Instruments. Para uma melhor comunicação e aquisição dos dados (latitude e longitude) do módulo GPS, foi utilizado a biblioteca TinyGPS++. Devido à baixa memória do MSP430, para que fosse possível a utilização da biblioteca algumas alterações foram feitas como, a retirada de métodos que forneciam informações como altitude e velocidade que para a proposta do projeto era desnecessário.

O funcionamento do código ocorre da seguinte forma: um método da biblioteca TinyGPS++ que retorna se há ou não um sinal válido de latitude e longitude é utilizado na estrutura condicional if, enquanto o sinal não for válido a mensagem "Aguardando pelo sinal dos satélites..." será exibida no serial monitor caso contrário os dados de latitude e longitude são concatenados ao link ([maps.google.com/maps?q=LATITUDE, LONGITUDE](https://maps.google.com/maps?q=LATITUDE, LONGITUDE)) do GoogleMaps para que a localização seja visualizada.

#### III. RESULTADOS

Os testes foram realizados no laboratório SS e no segundo andar do prédio UED, ambos na faculdade do Gama. Os primeiros testes não foram bem sucedidos quanto à conexão do sinal GPS dentro do laboratório. Fora dele, o sinal foi encontrando bem mais rapidamente e o resultado mostrado no monitor, como representado na figura a seguir:

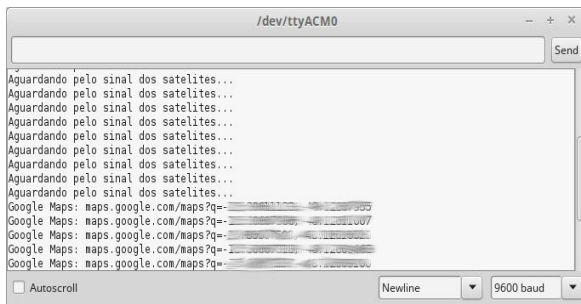


Fig. 6: Resultado apresentado da leitura dos dados e o posterior envio do link da localização precisa.

Após os resultados obtidos com a execução do código da Figura 6, que cobriu a parte do sinal GPS basicamente, realizou-se a codificação da conexão do microcontrolador com o módulo GSM a ser utilizado. Algumas alterações foram feitas no escopo do código, de modo a estar de acordo com os objetivos almejados, que seria a concatenação do link do Google Maps, com as coordenadas de latitude e longitude, com o número de celular o qual seria utilizado para solicitar a localização exata do veículo, via SMS. O código se encontra no apêndice deste relatório.

Entretanto, o resultado não foi como o esperado, pois ao mandar o SMS com o comando ">Local<" como especificado no código para o número cadastrado no chip do módulo GSM, nenhuma mensagem com a localização exata foi recebida. A figura abaixo mostra o que apareceu no compilador:

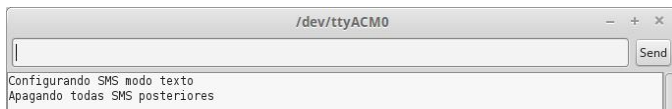


Fig. 7: Resultado com o novo código realizado.

Uma possível causa apurada foi o fato do código possivelmente não entrar no laço da condição if do seguinte trecho:

```
if(sms==">Local" && smsAnterior!=sms)
{
    Serial.println("Enviando localizacao atual");
    envia_sms_gps();
}
```

Com a adição do módulo GSM no projeto, o circuito ficou com a seguinte apresentação:

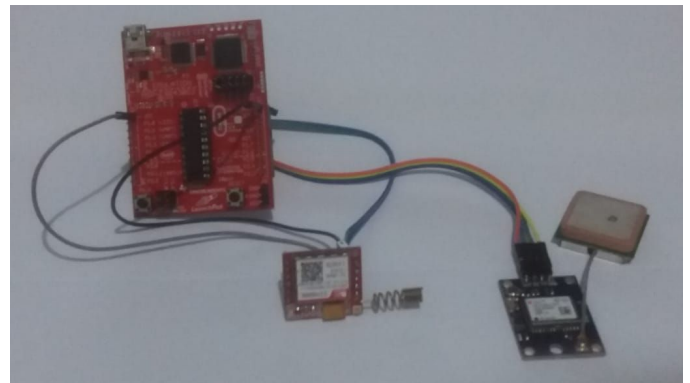


Fig. 8: Circuito montado com o módulo GSM SIM800L

#### IV. CONCLUSÃO

Obteve-se, com o projeto realizado, alguns sucessos e algumas falhas. A partir da ideia inicial e ao longo da execução, o grupo, depois de inúmeras pesquisas realizadas na internet, teve condições suficientes para fazer o código, tendo juntado todas as informações. O código foi realizado, embora não completamente na linguagem C e infelizmente o resultado final desejado, que era o de conectar os módulos GPS e GSM e fazer com que fosse enviada a localização exata do veículo via SMS ao usuário após o envio do comando adequado, não foi atingido. Como já comentado ao longo do relatório, algumas falhas relacionados ao sinal da antena do módulo GSM podem ter sido a causa desse problema e não o código.

No entanto, além do aprendizado, com a execução deste projeto pôde-se observar também que a necessidade de instalação de sistemas localizadores de veículos vem crescendo à medida que também cresce o número de veículos nas ruas. Alguns dos benefícios estão listados a seguir:

- Um sistema rastreador de veículo pode ajudar uma pessoa a determinar se o seu veículo está sendo utilizado para fins indevidos.
- Com o Sistema de Rastreamento de Veículos, não haverá necessidade de um chefe continuar ligando para o funcionário para saber sua localização. Assim, reduzindo as despesas da empresa em termos de contas telefônicas.
- Em grandes empresas, o sistema de rastreamento de veículos pode ser usado como um sistema de gerenciamento de frota. Assim, tornando-se essencial para resolver quaisquer alegações falsas ou reclamações contra a empresa.
- No Brasil, casos de roubos de carros raramente são resolvidos pela polícia. Ter um sistema de rastreamento próprio aumenta as chances de alguém encontrar o veículo perdido, aconselhando-se ajuda policial para maior segurança.

- CÓDIGO NO ENERGIA

```
#include <SoftwareSerial.h>
#include "TinyGPS.h"

TinyGPSPlus gps;

SoftwareSerial mySerial(P1_2, P1_1); //RX,TX

void setup()
{
  Serial.begin(9600);
  mySerial.begin(9600);
}

void loop()
{
  smartDelay(1000);

  if (!gps.location.isValid())
  {
    Serial.println("Aguardando pelo sinal dos satelites...");
  }
  else
  {
    Serial.print("GoogleMaps:maps.google.com/maps?q=");
    delay(300);
    Serial.print(gps.location.lat(), 8);
    Serial.print(",");
    delay(300);
    Serial.print(gps.location.lng(), 8);
    delay(300);
    Serial.print("\n");
    delay(300);
  }
}

static void smartDelay(unsigned long ms)
{
  {
    unsigned long start = millis();
    do
    {
      while (Serial.available())
        gps.encode(Serial.read());
    } while (millis() - start < ms);
  }
}
```

- CÓDIGO EM C NO CODE COMPOSER STUDIO

O CÓDIGO A SEGUIR REPRESENTA APENAS O MODO DE CONFIGURAÇÃO UART UTILIZANDO O MODO DE LOOP BACK, EM QUE UM CANAL TRANSMITE E RECEBE A MESMA VARIÁVEL.

```
//#include <SoftwareSerial.h>
#include "msp430g2553.h"
#include <stdlib.h>
#include "TinyGPS.h"

#define MAX_ITER 10

//TinyGPSPlus gps;

void main(void)
{
  WDTCTL = WDTPW + WDTHOLD; //parar o watchdog timer

  if (CALBC1_1MHZ==0xFF)
  {
    while(1);
  }

  //Configurando os clocks

  DCOCTL = 0;
  BCSCCTL1 = CALBC1_1MHZ;
  DCOCTL = CALDCO_1MHZ;

  //Configurando os LEDs

  P1DIR |= BIT0 + BIT6; // P1.0 and P1.6 output
  P1OUT &= ~BIT0 + BIT6; // P1.0 and P1.6 = 0

  //Setando a função UART para P1.1 & P1.2

  P1SEL |= BIT1 + BIT2; // P1.1 UCA0RXD input
  P1SEL2 |= BIT1 + BIT2; // P1.2 UCA0TXD output

  //Configurando a UART(USCI_A0)

  UCA0CTL1 |= UCSSEL_2 + UCSWRST;
  UCA0BR0 = 104;
  UCA0BR1 = 0;
  UCA0MCTL = UCBRS_1;
  // UCA0STAT |= UCLISTEN;
  UCA0CTL1 &= ~UCSWRST;

  //Acionando as interrupções

  IE2 |= UCA0TXIE;
  IE2 |= UCA0RXIE;
```

```

_BIS_SR(GIE);
}

UCA0TXBUF = 'A';           // Trasmittir um byte

_BIS_SR(LPM0_bits + GIE);
}

//Transmitir e receber interrupções

#pragma vector = USCIAB0TX_VECTOR
__interrupt void TransmitInterrupt(void)
{
    P1OUT ^= BIT0; //ligar o led P1.0 em Tx
}

#pragma vector = USCIAB0RX_VECTOR
__interrupt void ReceiveInterrupt(void)
{
    P1OUT ^= BIT6; // Ligar o led P1.6 em RX
    IFG2 &= ~UCA0RXIFG;
}

```

## VI. REFERÊNCIAS

- [1] <http://www1.folha.uol.com.br/cotidiano/2017/10/1931061-brasil-tem-1-roubo-ou-furto-de-veiculo-a-cada-minuto-ri-o-lidera-o-ranking.shtml>
- [2] <https://www.cooking-hacks.com/projects/arduino-realtime-gps-gprs-vehicle-tracking>
- [3] MSP430G2553- Datasheet by Texas Instruments.
- [4] <http://eie.uonbi.ac.ke/sites/default/files/cae/engineering/eie/VEHICLE%20TRACKING%20SYSTEM%20USING%20MSP430.pdf>