

Proposta de emprego de giroscópio e acelerômetro na perícia de acidentes automobilísticos

O.L. Vinícius ^{a,*}, Z. Ricardo ^b

^a Instituto de Criminalística, Polícia Civil do Distrito Federal, Brasília (DF), Brasil

^b Universidade de Brasília, Departamento de Engenharia Elétrica, Brasília (DF), Brasil

*Endereço de e-mail para correspondência: peritoviniciuslima@gmail.com. Tel.: +55-61-984119464.

Recebido em 08/11/2016 Revisado em 13/12/2016; Aceito em 14/12/2016

Resumo

Os módulos de *airbag* que equipam os atuais veículos são responsáveis por acionar a deflagração de suas bolsas, quando certo limiar de desaceleração é percebido pelo seu sensor. Quando este limiar é atingido, dados de colisão como aceleração, velocidade, situação do pedal de freio (pressionado ou não) e estado dos cintos de segurança (tensionados ou não) são gravados na memória do módulo. Em países desenvolvidos existem *scanners* capazes de extrair essas informações e gerar relatórios extremamente úteis para análise forense de acidentes de trânsito. Porém, na grande maioria dos casos, esses *scanners* não são compatíveis com os veículos fabricados no Brasil ou são de uso exclusivo das montadoras, por isso, a análise pericial fica prejudicada. Esse trabalho pretende propor uma solução de baixo custo que possa ser embarcada nos veículos, como por exemplo em viaturas policiais, e que grave, de maneira simples e confiável os dados de velocidade angular e aceleração de uma eventual colisão. O equipamento proposto seria útil, ainda, no estudo do funcionamento dos módulos de *airbag* dos veículos nacionais, na medida em que geraria resultados para servir de comparação com os módulos das montadoras.

Palavras-Chave: *Airbag*; Módulo; Acelerômetro; Giroscópio; Pré-tensionador; Caixa preta; Acidente de trânsito; EDR.

Abstract

Nowadays, *airbag* modules are the devices used to fire its safety bags when certain threshold deceleration is sensed. When this threshold is reached, data like acceleration, velocity, brake pedal status and sit belt status are written in the module's memory. In development country, there are scanners able to extract these informations and with them fill a report extremely useful in forensics science. Nevertheless, many of these scanners are not compatible with vehicles manufactured in Brazil or are exclusive of car brands, impairing forensics analysis. This paper intention is to build a simple, trustable, low-cost solution that can be installed inside a vehicle (for example: police cars) and able to write acceleration and angular velocity values in a eventual collision. This device would also be useful for understanding the operation of Brazilian modules, giving results to compare with convectional ones.

Keywords: *Airbag*; Module; Accelerometer; Gyroscope; Sit belt; Black box; Traffic accident; EDR.

1. INTRODUÇÃO

O termo técnico EDR, *Event Data Recorder*, é usado para designar as caixas pretas no mundo dos veículos automotores. Eles estão diretamente relacionados com os sistemas de *airbag* e pré-tensionadores dos cintos, na medida em que armazenam as informações nos instantes que antecedem e sucedem a uma colisão. Essas informações variam conforme o fabricante, mas em geral são dados como aceleração, variação de velocidade, estado do pedal de freio, das bolsas de *airbag* e dos cintos de segurança momento da colisão. Elas são armazenadas na

memória EEPROM do módulo de *airbag*, a qual por ser não volátil, mantém os dados intactos mesmo após um eventual corte de energia [1].

Todavia, o acesso a essas informações não é simples. Não há um padrão para a forma como elas são armazenadas e nem se sabe se os dados são de fato gravados na memória, após o veículo se envolver em um acidente que acionou as bolsas de *airbag*. No Brasil não há qualquer tipo de legislação que obrigue as montadoras a disponibilizarem um sistema de caixa preta em seus veículos, como ocorre, por exemplo, nos EUA, onde o órgão responsável pelo trânsito no país, *NHTSA (National Highway Traffic Safety*

Administration), regulou os parâmetros mínimos exigidos nas EDRs [2].

Atualmente a leitura das informações de colisão (conhecida na área pelo termo em inglês *crash data*) é restrita basicamente aos *scanners* fabricados pelas próprias montadoras e a um equipamento criado pela empresa alemã *Bosch* [3]. *Tais equipamentos são* conhecidos pelo termo inglês *CDR, Crash Data Retrieval*. Este equipamento da Bosch é o mais utilizado na área forense de países mais desenvolvidos, porém ele não é compatível com a maioria dos modelos de veículos que trafegam no Brasil, haja vista que o equipamento necessita do NIV (Número de Identificação Veicular) para gerar o relatório com os dados existentes no módulo.

Sem o uso de tais equipamentos, é possível até se extrair os dados binário das *EEPROMs* retiradas dos módulos de carros acidentados, porém, sua interpretação é muito difícil, uma vez que se necessita de informações sobre a formatação dos dados, das unidades métricas empregadas (distância e tempo) e da escala de operação do acelerômetro do *airbag* [4]. Muitos dos componentes eletrônicos presentes no circuito dos módulos são exclusivos e seus *datasheets* não são públicos (as montadoras contratam empresas terceirizadas para produzir componentes específicos e não divulgam a forma como funcionam). Informações sobre o limiar de aceleração para disparo das bolsas de *airbag*, sobre o tamanho da palavra e a resolução do conversor analógico-digital, sobre a taxa de amostragem, dentre vários outros parâmetros, se estivessem disponíveis ajudariam na decifração e interpretação dos bits dos dados coletados nas *EEPROM's* de carro acidentados. A Figura 1 apresenta a listagem hexadecimal de um *crash data* coletado de uma *EEPROM* modelo 93LC66 originária de um *Renault/Scenic*.

Por outro lado, alguns equipamentos de segurança, como o sistema ABS (do alemão *Antiblockier-Bremssystem*), dificultam o trabalho da perícia. Atualmente, o principal vestígio para a perícia determinar a velocidade do veículo no instante da colisão é a marca de frenagem deixada na superfície em que ele trafegava. Com ela é possível, estimar a velocidade a partir de fórmulas da Física. Com o advento dos sistemas de *ABS*, essas marcas estão cada vez mais raras, pois o *ABS* trabalha com sistema de travamento e liberação dos freios em frações de segundos, de forma a evitar o bloqueio das rodas, o que resulta em pouco ou quase nenhum rastro. Muitas vezes, os peritos deixam de concluir sobre a velocidade do veículo por não ter evidências físicas para calculá-la. Este é mais um fato a demonstrar a necessidade alternativa do uso das informações da “caixa preta” dos veículos modernos.

2. OBJETIVOS E MOTIVAÇÃO

A proposta aqui apresentada surgiu de um trabalho de mestrado onde se buscava ler e interpretar o *crash data* de veículos acidentados. Constatou-se uma grande dificuldade para se obter informações sobre a formatação dos dados gravados na memória dos módulos de *airbag*. Para contornar este problema, foi construído um circuito com acelerômetro e giroscópio, que pudesse ser fixado mecanicamente ao módulo de *airbag*, e assim gravasse informações precisas para posterior comparação com o *crash data* do módulo em análise.

Por outro lado, a ausência de regulamentação nacional na área permite que cada fabricante adote sua própria solução, ou então, que simplesmente não grave os dados por ocasião do acionamento do *airbag*. Isso trouxe grande preocupação, pois seria necessário fazer uma análise particular para cada modelo.

Diante desta dificuldade e levando em conta a ausência de regulamentação no país, decidiu-se ampliar o circuito cuja finalidade inicial era auxiliar na análise dos módulos de *airbag*, para um dispositivo que pudesse ser embarcado nos veículos e que fornecesse dados para a análise forense em caso de acidentes.

Assim, construiu-se um pequeno equipamento, de baixo custo, para ser acoplado a um veículo, e que forneça as informações de aceleração (variação da velocidade) e de velocidade angular no instante de uma colisão. Além de fornecer um resultado extremamente fiel à realidade, esse dispositivo pode servir como base de comparação na busca dos parâmetros “escondidos” no *crash data* do módulo de *airbag* que equipou o veículo acidentado.

O dispositivo proposto grava continuamente as informações de aceleração e velocidade angular. Assim, se o veículo sofrer um acidente que infle as bolsas de seu *airbag*, esses dados registrados poderão ser comparados com os que foram gravados na memória do módulo de

[illegible]

Figura 1. Listagem em hexadecimal do *crash data* do módulo de *airbag* de um carro acidentado (coletado em um Renault/Scenic)

airbag para busca de alguma relação. Seria um bom ponto de partida no entendimento do funcionamento dos equipamentos de *EDR* usados no país, já que a literatura brasileira nessa área é muito escassa.

Como a área forense está direta e indiretamente ligada à atividade policial, sugere-se que o equipamento projetado seja utilizado em viaturas policiais, pois são comuns os acidentes envolvendo esses carros oficiais.

Esse dispositivo proposto promete ser extremamente útil para uma análise pericial de reconstrução de acidentes de trânsito. De posse dessas informações é possível ter detalhado o comportamento do veículo nos instantes anteriores e posteriores à colisão. Nesse estudo retratou-se ainda, em uma simulação 3D, a dinâmica do veículo colidido, com os dados gravados no equipamento projetado. Com ajuda do software Unity3D é possível visualizar em um modelo virtual os movimentos de giro e os valores de aceleração dos últimos segundos gravados na memória do equipamento, inovação essa não existente nos aparelhos de *CDR* atuais.

3. PROJETO PROPOSTO

Buscou-se simplicidade e funcionalidade na construção do protótipo para os ensaios. Por isso foi selecionado uma placa microcontroladora *Arduino Mega 2560* [5], um sensor *MPU-6050* (acelerômetro e giroscópio) [6] e uma memória externa *EEPROM*, modelo 24LC1025, de 128 KB [7] e um Relógio de Tempo Real (*Real Time Clock*, *RTC*) que usa o *chip* DS-1307 [8]. O preço ficou baixo, pois o *Arduino Mega* custa na faixa de 70 reais, enquanto o *MPU*, por volta de 20 reais, a memória 20 reais e o *RTC* em torno de 10 reais (quando da publicação desse artigo). Assim, o custo estimado do protótipo é de R\$ 120,00.

O sistema faz uso do barramento *I2C* (*Inter-Integrated Circuit*), sendo que o *Arduino Mega* é o mestre deste barramento e os demais elementos, a saber, o *MPU*, a *EEPROM* e o Relógio são os escravos. O projeto prevê alimentação de 12 V (mesma existente em veículos) e todo o dispositivo foi alocado em um caixa com as dimensões 3 x 12 x 12 cm, como mostrado na Figura 2. É importante frisar que, assim como nos módulos de *airbag* convencionais, a disposição do equipamento no veículo deve ser referenciada com o sentido dos eixos do *MPU*. A sugestão é que o eixo X do *MPU* seja colocado sobre o eixo longitudinal do veículo e apontado para a direção de deslocamento, ou seja, para sua frente.

Ao equipamento ainda poderia se conectar outros dispositivos e como sugestão se propõe um módulo *GPS* serial capaz de enviar dados de geoposicionamento do veículo, no momento da colisão e ainda uma estimativa de velocidade. Isto enriqueceria ainda mais as possibilidades para a análise forense.

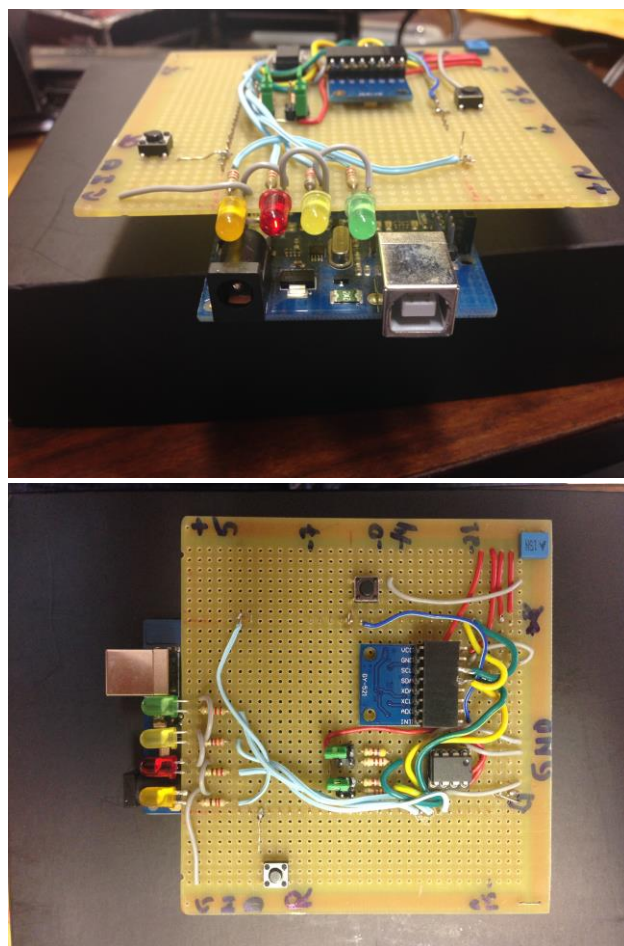


Figura 2. Equipamento construído.

O esquema elétrico do protótipo é apresentado na Figura 3. É fácil a identificação dos três periféricos básicos:

- *MPU-6050*: acelerômetro e giroscópio;
- *RTC DS1307*: relógio de tempo real e
- *24LC1025*: *EEPROM* de 128 Kbytes

É preciso notar que o *Arduino Mega* se comunica com todos os dispositivos por meio de um barramento serial a dois fios (*SDA* e *SCL*) que segue o protocolo *I²C* (que na literatura do *Arduino* é referenciado como *TWI – Two Wire Interface*). Nessas linhas *SDA* e *SCL*, há a presença de resistores que funcionam como *pull-ups*, exigidos no protocolo *I2C* [8]. Existem duas opções de resistores de *pull-up* que devem ser selecionadas pelos jumpers *JP1* e *JP2*, de acordo com a velocidade do barramento.

Pelas figuras anteriores ainda observamos que no dispositivo proposto há a presença de 4 leds e 2 botões. Os leds servem para sinalização ao usuário. Por enquanto, a proposta é a de que o led verde (*D3*) fique aceso enquanto o sistema está operando, o amarelo (*D2*), quando algum erro foi detectado e o vermelho (*D1*) para indicar que já ocorreu uma colisão. O led *D4* é usado para sinalização extra e teve grande emprego durante a fase de desenvolvimento. O botão *S1* é o de *Reset* e o *S2* também é voltado para o desenvolvimento do protótipo.

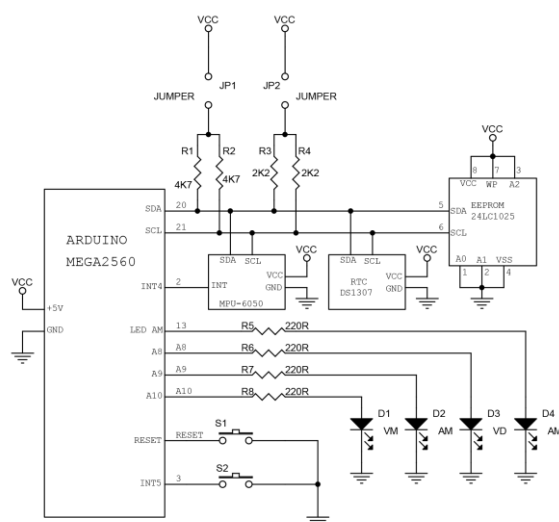


Figura 3. Esquema de comunicação do projeto.

O MPU-6050 é um sensor do tipo MEMS (*Micro Electro Mechanical System*) capaz de realizar leituras de aceleração e giroscópio em cada um dos 3 eixos (X, Y e Z). Possui conversor analógico-digital de 16 bits para cada eixo, possibilitando assim grande precisão nos resultados de cada canal. Cada leitura completa do MPU possui 12 bytes, ou seja, 2 bytes (*HIGH* e *LOW*) para cada eixo de aceleração e giro. As quatro escalas em que ele trabalha são de ± 2 g, ± 4 g, ± 8 g e ± 16 g para a aceleração e ± 250 °/s, ± 500 °/s, ± 1000 °/s e ± 2000 °/s para o giroscópio.

3.1. Lógica de registro dos dados na memória

Para construir tal sistema de registro, era necessária uma memória não volátil onde se pudessem escrever continuamente os dados de aceleração e rotação. É claro que os dados mais recentes vão sobreescrevendo os mais antigos. Por ocasião de um eventual acidente, grava-se ainda o correspondente a metade da memória e se encerra a gravação. Assim, o sistema proposto é capaz de fornecer dados antes e depois do acidente.

Foi selecionada a memória 24LC1025 que é uma EEPROM com capacidade de 128 Kbytes (131.072 bytes). Ela trabalha com páginas de 128 bytes, sendo que gasta, no máximo, 5 ms para gravar cada página. Assim, o total de páginas é de 1024. A primeira página, denominada de página 0, como será explicado mais adiante, foi reservada para gravar as informações de configuração do sistema.

É importante esclarecer a questão da temporização, pois se trabalha com uma taxa de dados relativamente alta. A porta I²C foi configurada para operar na taxa limite do Arduino que é de 400 kHz (SCL). Nessa velocidade, gastam-se, aproximadamente 3,3 ms para se transmitir 128 bytes para a memória. Terminada a transmissão, a memória inicia a fase de gravação da página que dura, no máximo,

5 ms. Esta fase não demanda atenção do processador, por isso ele pode se voltar para as leituras de dados do MPU. Tem-se então, sob a ótica da memória 3,3 ms (transmissão) mais 5,0 ms (gravação). Isto resulta em 8,3 ms. Podemos arredondar para 10 ms, para assim garantir uma boa margem de segurança. Em 10 ms, o MPU-6050, que opera na taxa de 1.000 leituras por segundo, gera apenas 120 bytes. Assim, se conclui que o sistema proposto atende à taxa especificada.

Excluindo-se a página 0, que é a de configuração, sobram 130.944 bytes para gravar as leituras do MPU (12 bytes por leitura). O interessante é que 130.944 é múltiplo de 12, o que simplifica muito ao projeto. Portanto, temos que a memória é capaz de guardar 10.912 (130.944/12) leituras de aceleração e giro.

Para favorecer a resolução, o MPU foi colocado para trabalhar com a máxima taxa de amostragem, que é de uma leitura completa (3 eixos de aceleração e 3 eixos de rotação) a cada 1 ms. Como visto no parágrafo anterior, a memória consegue armazenar 10.912 leituras completas, o que corresponde a um intervalo de 10,912 segundos.

A lógica proposta é que a EEPROM seja gravada constantemente, sobreescrevendo a porção mais antiga quando seu limite é atingido. Porém, quando for detectado um acidente, prossegue-se a gravação por mais 5,456 segundos e depois se trava o sistema, permitindo apenas sua leitura ou o comando de apagar a memória. Assim, ficam gravados os dados correspondentes aos 5,456 segundos que antecederam ao acidente e aos 5,456 segundos que sucederam o acidente.

A caracterização de um acidente é dada pela ultrapassagem de valores limites para aceleração e rotação. Durante a fase de configuração, é possível especificar um limite de aceleração para cada um dos eixos do acelerômetro e também um limite de rotação para cada um dos eixos do giroscópio. Quando pelo menos um desses limites é ultrapassado, o sistema reconhece um acidente e inicia a gravação por mais 5,456 segundos. Em seguida atualiza a página zero, para indicar qual sensor identificou o acidente e o ponteiro para os dados do instante do acidente.

3.2. Dados de configuração do sistema

Como já afirmado, a página zero (de tamanho 128 bytes) foi reservada para armazenar as configurações do sistema e os dados que caracterizaram a batida. Por enquanto, foram empregadas apenas 45 posições, listadas na Tabela 1. Existe, portanto, espaço para ampliações futuras. O byte de endereço zero usa as letras 'T' ou 'F' para indicar se o dispositivo já foi acidentado. O byte seguinte, da mesma forma, indica se o MPU passou no auto-teste (*self-test*). As posições de 2 a 15 trazem os parâmetros da calibração. As posições 16 e 17 indicam as

escalas usadas durante a operação do dispositivo. Os limiares a serem usados para caracterizar um acidente estão indicados nas posições de 18 a 29. O ponteiro para a leitura que detectou o acidente está nas posições 30, 31 e 32. As 6

posições seguintes (33 a 38) indicam qual sensor detectou o acidente. Finalmente, as próximas 6 posições trazem as informações da data e hora do acidente. As demais posições estão vazias e são preenchidas com o byte 0xFF.

Tabela 1. Lista de alguns comandos

Bytes	Tipo	Descrição
0	Char	(T) TRUE → pronta para uso, (F) FALSE → acidentada
1	Char	(T) TRUE → passou no <i>self test</i> , (F) FALSE → falhou no <i>self test</i>
2	Byte	Acelerômetro, escala usada na calibração
3	Byte	Giroscópio, escala usada na calibração
4, 5	Int	Eixo X, resultado da calibração do acelerômetro
6, 7	Int	Eixo Y, resultado da calibração do acelerômetro
8, 9	Int	Eixo Z, resultado da calibração do acelerômetro
10, 11	Int	Eixo X, resultado da calibração do giroscópio
12, 13	Int	Eixo Y, resultado da calibração do giroscópio
14, 15	Int	Eixo Z, resultado da calibração do giroscópio
16	Byte	Acelerômetro, escala usada na operação
17	Byte	Giroscópio, escala usada na operação
18, 19	Int	Limiar de disparo, eixo X, acelerômetro (valor absoluto)
20, 21	Int	Limiar de disparo, eixo Y, acelerômetro (valor absoluto)
22, 23	Int	Limiar de disparo, eixo Z, acelerômetro (valor absoluto)
24, 25	Int	Limiar de disparo, eixo X, giroscópio (valor absoluto)
26, 27	Int	Limiar de disparo, eixo Y, giroscópio (valor absoluto)
28, 29	Int	Limiar de disparo, eixo Z, giroscópio (valor absoluto)
30, 31, 32	Long	Ponteiro para o bloco de 6 leituras onde ocorreu o disparo
33	Char	Acel Eixo X: (T) TRUE → disparou, (F) FALSE → não disparou
34	Char	Acel Eixo Y: (T) TRUE → disparou, (F) FALSE → não disparou
35	Char	Acel Eixo Z: (T) TRUE → disparou, (F) FALSE → não disparou
36	Char	Giro Eixo X: (T) TRUE → disparou, (F) FALSE → não disparou
37	Char	Giro Eixo Y: (T) TRUE → disparou, (F) FALSE → não disparou
38	Char	Giro Eixo Z: (T) TRUE → disparou, (F) FALSE → não disparou
39	Char	Ano (0 → 99)
40	Char	Mês (1 → 12)
41	Char	Dia (1 → 31)
42	Char	Hora (0 → 23)
43	Char	Minuto (0 → 59)
44	Char	Segundo (0 → 59)
45	Char	0xFF
...	Char	0xFF
127	Char	0xFF

3.3. Comunicação com o dispositivo proposto

Para facilitar seu acesso, o sistema proposto é acessado via porta USB. Ele se comporta como um terminal serial que responde a comandos enviados pelo computador. Inicialmente, alguns comandos simples foram propostos. É claro que o conjunto de comandos será expandido por ocasião de seu emprego profissional. Cada comando é caracterizado por uma letra, que pode ser ou não seguida por parâmetros. A [Tabela 2](#) apresenta apenas alguns dos comandos básicos.

Apresentamos alguns exemplos: Quando se recebe o comando “?”, o dispositivo envia os dados da página zero,

aquela que armazena todas as configurações. O comando “R” indica para o sistema apagar toda a EEPROM (gravar 0xFF). O comando “I” indica para inverter o estado de acidentado (T) para não acidentado (F) ou ao contrário. Como resposta ao comando “L”, o dispositivo envia todo o conteúdo da memória. Os outros 6 comandos permitem especificar os limites para caracterizar acidentes.

Apenas este pequeno conjunto de comandos já permite o emprego do dispositivo proposto e a leitura dos dados para posterior análise.

Tabela 2. Lista de alguns comandos

Cmdo.	Param.	Descrição
?	-	Enviar configuração. (página 0)
I	-	Inverter estado acidentado (T → F ou de F→T)
R	-	Zerar toda a EEPROM
L	-	Ler toda a memória
X	n	Limiar acelerômetro eixo X
Y	n	Limiar acelerômetro eixo Y
Z	n	Limiar acelerômetro eixo Z
x	n	Limiar giroscópio eixo X
y	n	Limiar giroscópio eixo Y
z	n	Limiar giroscópio eixo Z

3.4. Segurança no armazenamento dos dados

Os dados de aceleração e velocidade são gravados sequencialmente na memória EEPROM. Cada valor de leitura tem 16 bits, ou seja, é composta por 2 bytes, denominados *High* e *Low*. Assim, uma leitura completa que é composta por 3 eixos do acelerômetro e 3 eixos do giroscópio, precisa de 12 bytes, como mostrado na Tabela 3. Todavia, a gravação em sequência é preocupante, pois se a gravação na EEPROM perder 1 único byte, todos os demais dados ficarão deslocados e serão interpretados de forma errada.

Tabela 3. Forma como os 12 bytes do MPU são armazenados.

Acelerômetro						Giroscópio					
Eixo X		Eixo Y		Eixo Z		Eixo X		Eixo Y		Eixo Z	
H	L	H	L	H	L	H	L	H	L	H	L

Para fornecer segurança no armazenamento é proposta o seguinte método de marcação dos dados. Reduzir a precisão de 16 bits para 14 bits, e usar esses dois bits extras para marcar leituras do acelerômetro e do giroscópio. A palavra que agora é de 14 bits é quebrada em duas palavras de 7 bits, e para completar o byte, bit mais à direita é feito igual a 1 no caso dos dados do acelerômetro e igual a 0, no caso dos dados do giroscópio.

Desta forma, na memória deverão estar grupos de 12 bytes, onde os 6 primeiros são ímpares (último bit igual a 1) e os 6 últimos são pares (último bit igual a 0). Enquanto este padrão for repetido, os dados são confiáveis. Caso haja algum erro, deve-se recusar os dados errados e avançar para o próximo bloco confiável. A sugestão é interpolar os dados que forma perdidos.

3.5. Lógica para leitura e gravação dos dados

Como já foi citado, a memória EEPROM gasta 5 ms para gravar um byte. Porém ela oferece um modo de gravação de página que gasta, no máximo, os mesmos 5 ms para gravar 128 bytes. Assim, por premência de tempo, decidiu-se realizar a gravação por páginas. É preciso comentar que o número 128 não é múltiplo de 12, com isso, leituras consecutivas poderão ficar em páginas diferentes. Outro ponto importante é que a leitura dos dados do MPU e a gravação na EEPROM não podem ter uma sincronização fina. O tempo de gravação varia muito. O manual especifica apenas o limite máximo que é de 5 ms.

Por isso, a solução foi trabalhar com dois *buffers* de 128 bytes (tamanho da página) cada um. Assim, enquanto um *buffer* está sendo preenchido com as leituras, o outro está sendo gravado. O programa trabalha com dois estados: Estado 1 e Estado 2. No Estado 1, grava o primeiro *buffer* enquanto preenche o segundo. No Estado 2, que é o oposto, grava o segundo *buffer* enquanto preenche o primeiro. Isto está mostrado na Figura 4. Para facilitar a partida do programa, foi criado o Estado 0, que preenche o primeiro *buffer* com os parâmetros de configuração e desvia para o Estado 1. O ponteiro do endereço a ser gravado vai de 128 a 130.944 e volta a 128. Assim, preserva a primeira página da memória, que é a de configuração. Ao final da gravação, após um acidente, a página 0 é atualizada com os dados relativos ao acidente.

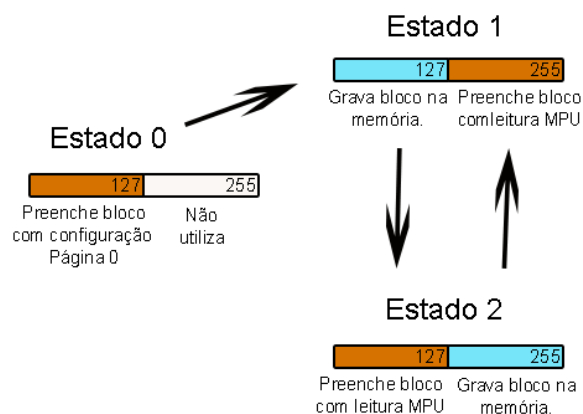


Figura 4. Diagrama de estados para leitura do MPU e escrita na memória EEPROM.

3.6. Garantia de alimentação

O sistema foi previsto para empregar os 12 V da alimentação veicular. Porém, é comum que a alimentação do veículo seja interrompida durante um acidente. Para solucionar este problema, foi prevista uma bateria de *back-up* de 9 V. Essa bateria só alimenta o sistema na falta da alimentação principal. Ela é capaz de sustentar o sistema

por algumas horas, o que é mais que suficiente para finalizar a gravação dos dados do acidente.

Esse recurso de *back-up* de alimentação pode ser mais uma informação útil para a perícia, pois pode indicar o instante exato em que o sistema de alimentação do veículo entrou em colapso.

4. MANIPULAÇÃO DOS RESULTADOS

O sistema proposto se comunica através de comandos e respostas enviados pela via porta USB. O comando “L” permite que se obtenha um mapa hexadecimal de toda a memória. De posse deste mapa, é possível empregar diversos programas para analisar os dados do acidente. Por exemplo, com o *Matlab*, é possível traçar gráficos com os valores instantâneos de aceleração e de velocidade angular. Com emprego da integral discreta, traçam-se gráficos de velocidade e posição angular.

Considerando que a colisão é exatamente o ponto de disparo do sistema e, ainda, considerando que o veículo não mais se movimentou após atingir sua posição de repouso final, teremos precisamente as velocidades do veículo segundo os três eixos, nos instantes anteriores e posteriores à colisão.

Como a quantidade de valores adquiridos é considerável (por volta de 10.900 amostras) e detalhada, é possível montar uma simulação 3D do acidente, a partir da plataforma *Processing*. Para tanto, é necessário um pré-processamento para gerar um arquivo texto (.txt) com a tabela de dados de aceleração e giro segundo cada eixo. Nessa simulação podemos visualizar o movimento de giro do veículo nos momentos que antecedem e sucedem a colisão e ainda os valores de aceleração alterando-se conforme esse movimento.

5. UTILIDADE PERICIAL

A velocidade é um dos principais elementos a ser desvendado nos laudos de exames em perícias de trânsito. Esse trabalho apresenta um dispositivo de baixo custo, que teria grande utilidade para a área pericial de acidente de trânsito, e pretende contornar a dificuldade de se extrair este parâmetro variável dos atuais módulos de *airbag*. O equipamento proposto, quando acoplado a um veículo, grava dados de aceleração e velocidade angular no momento de uma colisão e ainda serve como auditoria da velocidade dos últimos segundos armazenados memória.

O valor do giro coletado também seria importante para, além de poder projetar a dinâmica de movimentação do veículo, servir de comparação para verificar se o ponto de repouso final encontrada pelos peritos no local do acidente, por exemplo, é compatível com aquela encontrada nos valores de giroscópio.

Sugere-se o acréscimo de um módulo GPS serial. Assim além da velocidade no momento da colisão, o sistema forneceria, ainda, as coordenadas do veículo no instante do acidente. Esse tipo de sistema é simples, de baixo custo e, mais uma vez, com enorme valor pericial.

O sistema proposto possui uma bateria de *back-up* para suprir energia na eventualidade de falha no sistema de 12 V do veículo. É possível gravar no bloco de configuração em que instante do acidente ocorreu a falha de alimentação.

Um ponto importante a se considerar é o valor do limiar para caracterizar um acidente. A literatura nacional não é específica sobre este ponto. Ao que parece, o valor de 8 g é o que tem maior aceitação internacional [4]. Entretanto, cada fabricante usa um determinado valor. O sistema proposto se mostra, mais uma vez, de grande utilidade para a atividade forense, uma vez que possibilita encontrar qual é o valor de desaceleração que um determinado fabricante usa para ativar as bolsas e os pré-tensionadores. Por exemplo, consideremos um ensaio usando o módulo proposto configurado para disparar com uma desaceleração de 8 g instalado em um carro cujo módulo de *airbag* original esteja configurado para 4 g. Caso esse veículo envolva-se em colisão, que o faça disparar suas bolsas e ao mesmo tempo haja um corte de energia no sistema, teremos gravado em nosso equipamento o momento que houve o chaveamento da alimentação para a bateria (que provavelmente será algo em torno dos - 4 g). Se não houver o corte de energia, teremos no módulo projetado um valor de pico, onde ocorre a mudança de sentido do veículo (mudança de aceleração para desaceleração), saberemos assim, que o valor de disparo das bolsas no módulo original é algo abaixo desse pico.

Vale reforçar que, dada a escassez de informações sobre os módulos de *airbag* e seu conteúdo (*hardware* e *software*), além da limitada literatura no assunto, o equipamento proposto serve como porta de entrada para o entendimento e estudo do funcionamento desses sistemas nos veículos fabricados no Brasil. É um ótimo aparelho para servir como base para desvendar o significado dos códigos coletados na memória dos módulos.

6. CONCLUSÕES

No Brasil existe uma grande dificuldade para se extrair informações de relevância pericial dos módulos de *airbag*. A literatura internacional afirma que esses dispositivos funcionam como caixa preta de veículos, armazenando informações como aceleração e velocidade de uma colisão que tenha acionado seu código de deflagração das bolsas e pré-tensionadores dos cintos de segurança [4]. Porém, nada há de concreto sobre os veículos que são fabricados no Brasil: não se sabe se realmente gravam esse tipo de

informação e, em caso positivo, como interpretar os bits coletados a partir de sua memória.

Tendo este problema como ponto de partida, o presente trabalho obteve êxito na construção de um equipamento que apresente funcionamento semelhante, mas com custo baixo e que possa gerar facilmente informações úteis na reconstrução de uma cena de acidente de trânsito. O sistema proposto faz uso de um processador Arduino e de um módulo *MPU-6050* (acelerômetro e giroscópio) e de uma memória não volátil para armazenar dados dos 3 eixos de desaceleração e rotação durante um acidente.

O sistema é capaz de armazenar 10,9 s de informação o que representa 10.912 leituras de cada eixo X, Y e Z de aceleração e de giro. A grosso modo, são armazenados dados sobre os 5 s que antecederam e sobre os 5 s que sucederam o acidente. Em posse dos valores de aceleração no eixo X (eixo de direção do veículo) é possível facilmente obter os valores de diferença de velocidade, parâmetro esse extremamente valioso nas perícias de trânsito.

O sistema proposto permite o transporte dos dados coletados para um simulador 3D, o que permite a reconstituição virtual do acidente.

Com isso reforça-se a grande aplicabilidade do dispositivo proposto para o mundo forense de acidentes de trânsito e como ele pode ajudar os peritos e *experts* do assunto no entendimento dos sistemas *EDR* dos veículos.

AGRADECIMENTOS

Ao professor Zelenovisky pela atenção e empenho; à Fundação de Peritos Criminais (FPCIAA) e à Fundação de Apoio à Pesquisa do Distrito Federal (FAP/DF) pelo investimento e incentivo para com o projeto.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Wikipedia – Event Data Recorder. Disponível em: <https://en.wikipedia.org/wiki/Event_data_recorder>. Acesso em 5 de novembro de 2016.
- [2] Rosalyn G. Millman. 49 CFR Parts 552, 571, 585 and 595. National Highway Traffic Safety Administration, Departamento f Transportation.
- [3] Wikipedia – Bosch Diagnostics. Disponível em: <<https://www.boschdiagnostics.com/cdr/>>. Acesso em 21 de outubro de 2016.
- [4] W. Rosenbluth. Black Box data From Accident Vehicles, Methods of Retrival, Translation, and Interpretation. Ed. ASTM International. 2009.
- [5] Atmel ATmega640/V-1280/V-1281/V-2560/V-2561/V. 2549Q–AVR–02/2014. Datasheet.
- [6] MPU-6000 and MPU-6050 Product Specification, Revision 3.4, Release Date: 08/19/2013, Document Number: PS-MPU-6000A-00. InvenSense.
- [7] Michroship 24AA1025/24LC1025/24FC1025. DS21941B. Datasheet.
- [8] Maxim Integrated DS1307 64 x 8, Serial, I2 C Real-Time Clock.
- [9] Maxim Integrated I2C-bus specification and user manual. UM10204. Rev. 6 — 4 April 2014. NXP Semiconductors.
- [10] C.Y. Chan. Fundamentals of Crash Sensing in Automotive Air Bag Systems. Ed. SAE International. 2000.