



## Relatório Final

Título do projeto de pesquisa: Desenvolvimento de Aplicativo para Aquisição e Visualização de Dados de Telemetria  
Bolsista: Victor Hugo Fernandes Breder  
Orientador(a): Alison de Oliveira Moraes  
Período a que se refere o relatório: Agosto de 2015 a Julho de 2016

### Resumo

O projeto tem como objetivo o desenvolvimento de um aplicativo para o Laboratório de Aquisição e Processamento de Dados (APDT) do Instituto de Aeronáutica e Espaço (IAE). Com esse aplicativo pretende-se decodificar e visualizar dados de telemetria em campo recebidos através de um sinal transmitido por cabo coaxial. O projeto faz uso de um decodificador e sincronizador de bits Apollotek, controlado via USB, que capta dados do cabo coaxial com sinais de telemetria recebidos em solo. Assim, o projeto permite que um notebook em solo, fazendo uso do dispositivo USB e do aplicativo desenvolvido, forneça visualização em tempo real das medidas transmitidas por rádio. O aplicativo foi utilizado com sucesso em ensaios de teste no Laboratório APDT e na Operação São Lourenço em novembro de 2015.

### 1. Introdução

Telemetria engloba o processo no qual medições realizadas por sensores embarcados são transmitidas para uma estação distante em solo, aonde os dados obtidos podem ser mostrados, gravados e analisados. A aplicação dessa técnica é essencial para o desenvolvimento de foguetes no IAE, já que é necessário obter dados críticos do foguete em tempo real e a uma distância segura durante ensaios e em operações de lançamento.

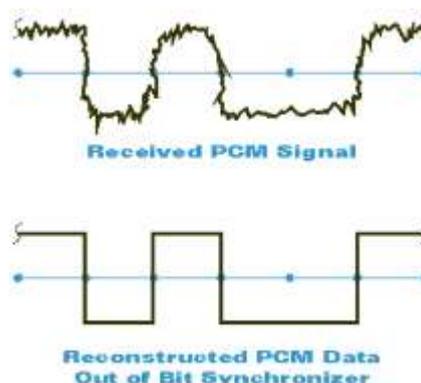
Devido à necessidade de enviar várias medidas em um único meio transmissor, os dados devem ser enviados em sequência, caracterizando a técnica de comutação. Para isso, no Plano de Medição é definido um quadro que descreve, entre outros parâmetros, a ordem na qual o sistema de telemetria embarcado deve enviar os dados.



**Figura 1** – Medições realizadas pelo sistema embarcado são enviadas em série de acordo com quadro. O quadro ilustrado é constituído por 6 subquadros.

Um quadro é constituído por vários subquadros. Cada subquadro inicia com o SFID, que é o número correspondente ao subquadro, e termina com o *Frame Sync* (Palavra de Sincronia), que é um código reservado para indicar o final de um subquadro. As demais palavras correspondem a medidas, que são representadas como números inteiros. Um quadro é configurado com uma quantidade de subquadros, uma quantidade de palavras por subquadro, uma quantidade de bits por palavra e uma certa disposição das medidas.

Durante o processo de transmissão, o sinal PCM é inevitavelmente distorcido devido às condições ambientais. Assim, o papel do Sincronizador de Bits é reconstruir o sinal PCM, produzindo um pulso com amplitude binária e distribuído em intervalos discretos para uso em sistemas digitais.



**Figura 2** – Distorção do sinal PCM pelo ambiente e reconstrução do sinal pelo Sincronizador de Bit.

Em seguida, o sistema de telemetria em solo, configurado com o mesmo quadro, é responsável por realizar a decomutação, isto é, receber cada dado em série e reconstruir o quadro associando as medidas de volta às grandezas correspondentes. O software em solo também deve oferecer uma interface amigável para visualização dos dados através de gráficos e tabelas.

Dessa maneira, um problema encontrado em operações como testes e lançamentos de foguetes no IAE é baixa acessibilidade às medidas dos sensores obtidas via telemetria. Normalmente, é necessário requisitar por voz o valor do parâmetro desejado a um operador com acesso a um software de telemetria através de rádio.

Portanto, o aplicativo objetiva solucionar esse problema oferecendo visualização dos dados de telemetria diretamente em campo utilizando apenas um notebook e um dispositivo USB, oferecendo, assim, respostas em tempo real para ações tomadas pelos técnicos que operam o ensaio ou o foguete.

## **2. Materiais**

Foram utilizados os seguintes equipamentos disponibilizados pelo Laboratório APDT, equivalentes aos que serão usados em campo:

- Apollotek APK8763 – USB PCM Decommutator and Bit Synchroniser
- Notebook Toshiba i7 4 GB de RAM com Windows 7

A escolha do Decomutador e Sincronizador de Bits Apollotek se deveu ao fato do dispositivo ser compacto e leve, permitindo o uso em campo, e ser capaz ler o sinal PCM de um cabo coaxial operando unicamente através do cabo USB. Assim, como a alimentação do dispositivo é unicamente via USB, evita-se problemas relacionados com aterramento.

Além disso, o equipamento já havia sido adquirido pelo IAE, mas não era utilizável, já que os notebooks disponíveis para uso em campo possuem Windows 7, enquanto o software fornecido pelo fabricante é compatível apenas com Windows XP.

O Laboratório também forneceu os seguintes equipamentos para gerar sinais PCM, com a finalidade de testar o aplicativo:

- Apollotek APK8764-2 – USB PCM Simulator
- Notebook Dell Inspiron XPSM1210 com Windows XP (para operação do simulador de sinal PCM)
- Gravador de Dados de Telemetria em Estado Sólido (para reprodução de dados PCM gravados durante ensaios)

Além disso, foram utilizados os seguintes programas disponíveis na Internet:

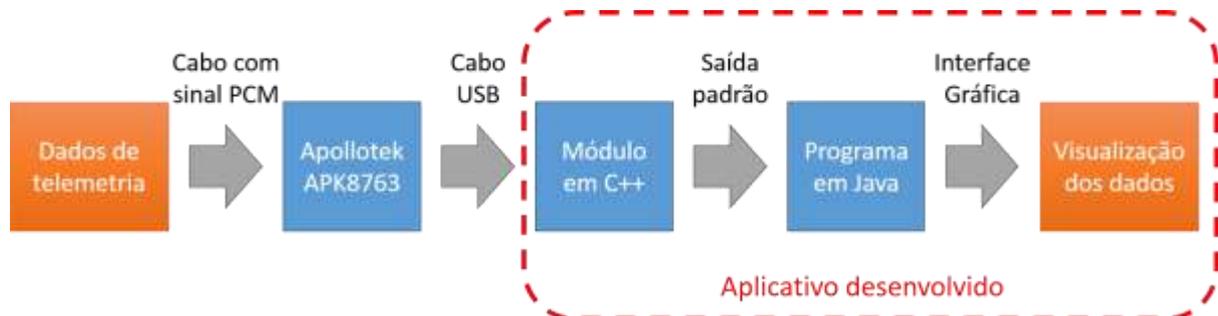
- Eclipse – IDE para desenvolvimento em Java
- Sublime Text 3 – Editor para desenvolvimento em C++
- GCC – Compilador para C++

A linguagem Java foi escolhida devido ao seu suporte nativo a interface gráfica e sua portabilidade, que confere ao software possibilidade de uso com versões posteriores do Windows. Além disso, a linguagem C++ foi escolhida para mais simples integração com DDLs fornecidas pelo fabricante, utilizadas para controlar o dispositivo USB, através do padrão Microsoft STDCALL.



**Figura 3** – Decomutador de Sincronizador de Bits USB Apollotek APK8763

### 3. Metodologia



**Figura 4** – Diagrama da visão geral do funcionamento do aplicativo.

O aplicativo deverá ser desenvolvido para atuar em conjunto com o decomutador e sincronizador de bits USB, Apollotek APK8763. Este dispositivo receberá um cabo com sinal de telemetria PCM e será controlado pelo aplicativo através de um cabo USB.

Para controlar o dispositivo USB, o fabricante Apollotek forneceu drivers e DLLs. Para lidar com essas instruções de baixo nível, será desenvolvido um módulo em C++ responsável por fazer chamadas à DLL e retornar os quadros obtidos em formato de texto pela saída padrão.

Finalmente, para lidar com a visualização de dados com uma interface gráfica, será utilizada a linguagem Java. Assim, o programa desenvolvido em Java receberá a saída em texto do módulo em C++ e mostrará os dados na tela do computador na forma de gráficos e tabelas em uma interface configurável.

O programa em C++ será desenvolvido com o Sublime Text 3 e compilado com o GCC como executável em Windows. O programa em Java será desenvolvido no Eclipse, com o uso do módulo WindowBuilder para a elaboração de janelas e da biblioteca gráfica AWT para geração de gráficos e tabelas.

Por fim, tendo em vista o uso a longo prazo, o software desenvolvido será validado para funcionar bem em computadores com Windows 7 e Windows 8.

## 4. Resultados

### 4.1. Funcionalidades do aplicativo



Figura 5 – Interface configurável de visualização de dados.

A Figura 5 ilustra a interface principal do aplicativo, que é utilizada para a visualização dos dados de telemetria. Essa interface é facilmente configurável para atender às necessidades do ensaio de teste ou operação de lançamento. São oferecidos os seguintes elementos para visualização dos valores de parâmetros na tela:

- Gráfico – Mostra um gráfico do valor do parâmetro ao longo do tempo. Também exibe numericamente o valor mínimo, valor máximo e média móvel das últimas amostras.
- Tabela – Mostra o valor numérico atual de vários parâmetros, podendo mostrar uma barra horizontal representando o valor do parâmetro.
- Botão – Representa o estado de um relê. Caso o relê esteja ativado, apresenta um fundo verde, e caso esteja desativado, apresenta um fundo vermelho.
- Registro de eventos – Armazena mensagens de texto com o tempo que eventos ocorreram. Os eventos são indicados por mudança do estado de relês.
- GPS – Representa longitude, latitude e altitudes obtidos via GPS.

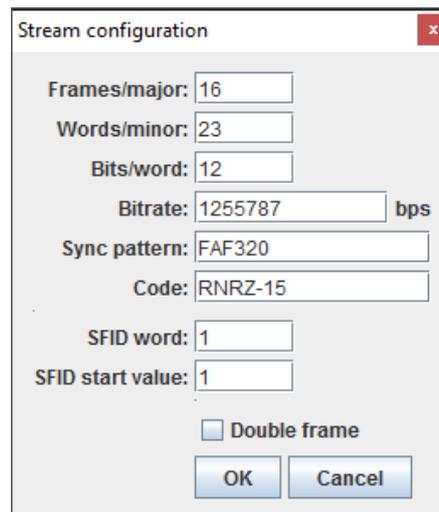


Figura 6 – Janela de configuração do sinal de telemetria

Através da janela mostrada na Figura 6, é possível configurar os parâmetros do sinal de telemetria, informando informações como tamanho do quadro, bits por palavra e taxa de bits. Esses parâmetros permitem que o dispositivo USB Apollotek adquira corretamente os dados do sinal PCM. O sucesso na configuração do steam é indicado no próprio dispositivo USB por um LED verde indicando “Lock”.

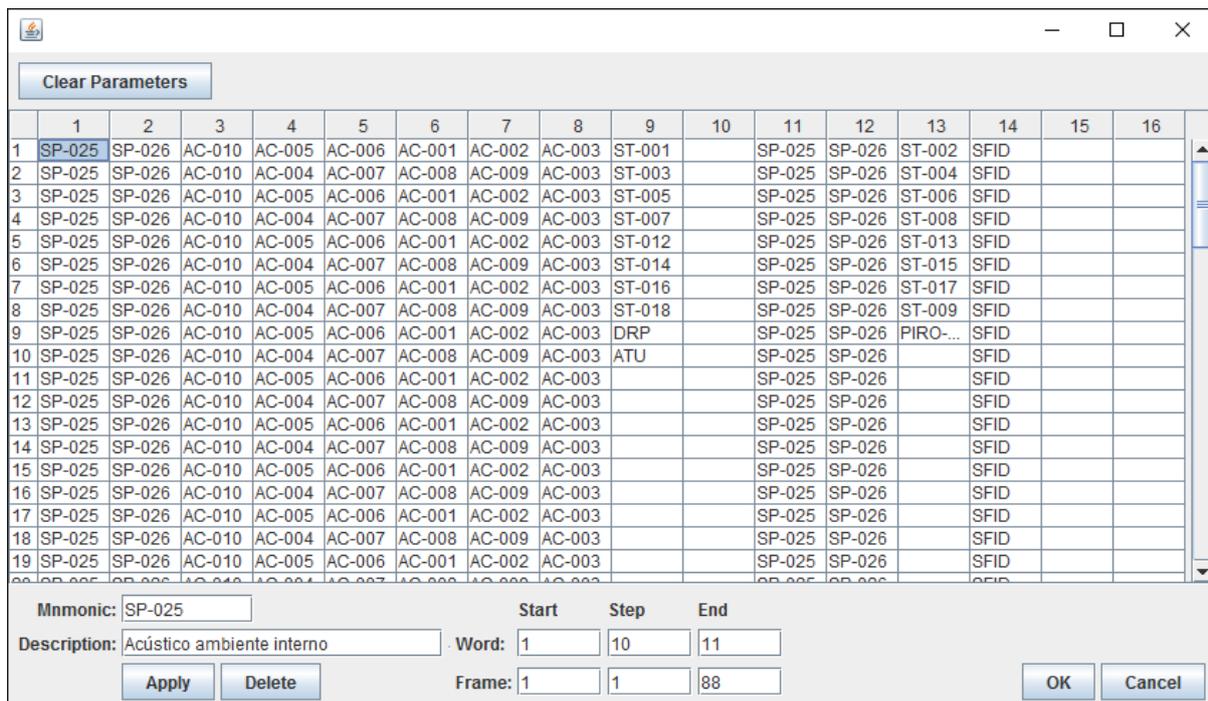
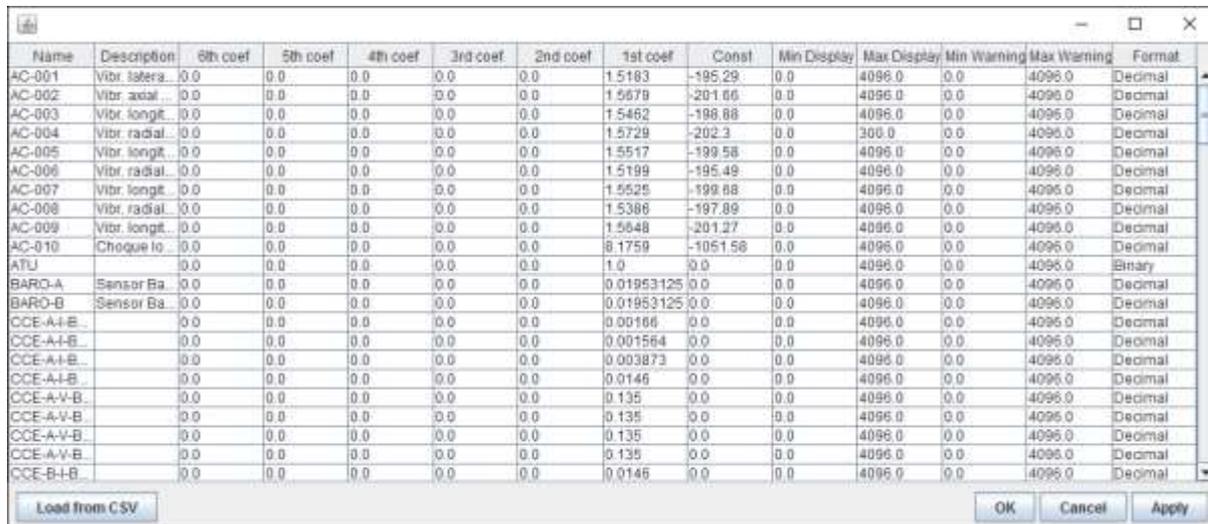


Figura 7 – Janela de configuração do quadro

A janela indicada pela Figura 7 permite informar ao aplicativo como o quadro que está sendo recebido através do stream se encontra configurado. Isso possibilita que cada medição isolada obtida seja associado com o parâmetro correto para posterior visualização em tela.



Nome	Description	5th coef	4th coef	3rd coef	2nd coef	1st coef	Const	Min Display	Max Display	Min Warning	Max Warning	Format
AC-001	Vibr. latera.	0.0	0.0	0.0	0.0	1.5183	-195.29	0.0	4096.0	0.0	4096.0	Decimal
AC-002	Vibr. axial	0.0	0.0	0.0	0.0	1.5679	-201.86	0.0	4096.0	0.0	4096.0	Decimal
AC-003	Vibr. longit.	0.0	0.0	0.0	0.0	1.5462	-198.88	0.0	4096.0	0.0	4096.0	Decimal
AC-004	Vibr. radial	0.0	0.0	0.0	0.0	1.5729	-202.3	0.0	300.0	0.0	4096.0	Decimal
AC-005	Vibr. longit.	0.0	0.0	0.0	0.0	1.5517	-199.58	0.0	4096.0	0.0	4096.0	Decimal
AC-006	Vibr. radial	0.0	0.0	0.0	0.0	1.5199	-195.49	0.0	4096.0	0.0	4096.0	Decimal
AC-007	Vibr. longit.	0.0	0.0	0.0	0.0	1.5525	-199.88	0.0	4096.0	0.0	4096.0	Decimal
AC-008	Vibr. radial	0.0	0.0	0.0	0.0	1.5386	-197.89	0.0	4096.0	0.0	4096.0	Decimal
AC-009	Vibr. longit.	0.0	0.0	0.0	0.0	1.5848	-201.27	0.0	4096.0	0.0	4096.0	Decimal
AC-010	Choque io	0.0	0.0	0.0	0.0	8.1759	-1051.58	0.0	4096.0	0.0	4096.0	Decimal
ATU		0.0	0.0	0.0	0.0	1.0	0.0	0.0	4096.0	0.0	4096.0	Binary
BARO-A	Sensor Ba.	0.0	0.0	0.0	0.0	0.01953125	0.0	0.0	4096.0	0.0	4096.0	Decimal
BARO-B	Sensor Ba.	0.0	0.0	0.0	0.0	0.01953125	0.0	0.0	4096.0	0.0	4096.0	Decimal
CCE-A-I-B		0.0	0.0	0.0	0.0	0.00166	0.0	0.0	4096.0	0.0	4096.0	Decimal
CCE-A-I-B		0.0	0.0	0.0	0.0	0.001564	0.0	0.0	4096.0	0.0	4096.0	Decimal
CCE-A-I-B		0.0	0.0	0.0	0.0	0.003873	0.0	0.0	4096.0	0.0	4096.0	Decimal
CCE-A-I-B		0.0	0.0	0.0	0.0	0.0146	0.0	0.0	4096.0	0.0	4096.0	Decimal
CCE-A-V-B		0.0	0.0	0.0	0.0	0.135	0.0	0.0	4096.0	0.0	4096.0	Decimal
CCE-A-V-B		0.0	0.0	0.0	0.0	0.135	0.0	0.0	4096.0	0.0	4096.0	Decimal
CCE-A-V-B		0.0	0.0	0.0	0.0	0.135	0.0	0.0	4096.0	0.0	4096.0	Decimal
CCE-A-V-B		0.0	0.0	0.0	0.0	0.135	0.0	0.0	4096.0	0.0	4096.0	Decimal
CCE-B-I-B		0.0	0.0	0.0	0.0	0.0146	0.0	0.0	4096.0	0.0	4096.0	Decimal

Figura 8 – Janela de configuração de parâmetros

Cada medição recebida através do sinal de telemetria é um número inteiro. Assim, a janela ilustrada pela Figura 8 permite a configuração dos coeficientes do polinômio que realiza a transformação do número inteiro recebido para um número decimal com significado físico. Também é possível selecionar o formato do parâmetro como número binário, que é uma maneira melhor de representar estados digitais de relés.

#### 4.2. Uso em campo

O aplicativo foi utilizado com sucesso em ensaios de teste no Laboratório APDT. O programa ofereceu resultados imediatos para as ações de técnicos enquanto realizavam alterações no componente do foguete que estava sendo testado. Além disso, na operação São Lourenço, o aplicativo foi utilizado nas preparações de lançamento.



## 5. Conclusões

O aplicativo desenvolvido atende com sucesso às necessidades do Laboratório APDT e se mostrou muito útil em ensaios de teste e no lançamento da operação São Lourenço. Assim, foi desenvolvido um aplicativo capaz de visualizar dados de telemetria em tempo real e de maneira portátil para uso em campo. Tendo tudo isso em vista, conclui-se que o projeto atingiu plenamente os objetivos.



## Referências

Telemetry Tutorial - L-3 Telemetry-West. Disponível em:

<[www.storm.com/telemetry\\_tutorial/telemetry\\_tutorial.pdf](http://www.storm.com/telemetry_tutorial/telemetry_tutorial.pdf)>. Acessado em 19 de fevereiro de 2016.

ApolloTek APK8763 USB Bit Synchroniser and Decommutator. Disponível em:

<[http://www.apollotek.com/Bit%20Sync&DecomData%20Sheet241108\\_1.pdf](http://www.apollotek.com/Bit%20Sync&DecomData%20Sheet241108_1.pdf)>. Acessado em 19 de fevereiro de 2016.

Gamma, E. Helm, R. Johnson, R. Vlissides, J. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Reading (1994).