****

# RELATÓRIO DAS ATIVIDADES DESENVOLVIDAS

Arthur Bezerra Dantas Saraiva (UFRN, Bolsista PIBITI/CNPq)  
E-mail: arthur.saraiva@crn.inpe.br

Dr. José Marcelo Lima Duarte (INPE, Orientador)  
E-mail: jmarcelo@crn.inpe.br

Abril, 2017

# Sumário

[1- Resumo do plano de trabalho proposto 3](#_Toc479081859)

[2- Resumo do que foi realizado 3](#_Toc479081860)

[3- Detalhamento dos progressos e mudanças necessárias 4](#_Toc479081861)

[a) Descrição do algoritmo do PC 4](#_Toc479081862)

[b) Descrição do algoritmo do arduino 7](#_Toc479081863)

[4- Considerações finais 9](#_Toc479081864)

[REFERÊNCIAS 10](#_Toc479081865)

# Resumo do plano de trabalho proposto

O INPE CRN possui em Natal uma estação de Telemetria, Rastreio e Comando (Telemetry, Tracking and Commanding - TT&C) dedicada ao satélite SACI. Infelizmente, essa estação está sem uso desde da falha no lançamento deste satélite. Recentemente, o INPE vem desenvolvendo uma estação solo que possa atender múltiplas missões utilizando componentes da estação de TT&C do SACI, num esforço para reaproveitar parte de sua estrutura. Nesse contexto, foi proposto como plano um novo software para rastreio de satélites para comandar o sistema de posicionamento da antena da estação de TT&C. Esse software, para PC com sistema operacional Windows, deve receber como entradas a posição geográfica da estação solo e a lista de satélites que se deseja rastrear. De posse desses dados, o software deve baixar automaticamente da internet os dados orbitais dos satélites selecionados, no formato Two-Line Element Set (TLE), e gerar as efemérides, coordenadas para rastreio dos satélites a partir da posição geográfica fornecida. As efemérides devem ser transmitidas via cabo USB para o sistema de posicionamento da antena da estação, a fim de permitir o apontamento. Além disso, o software deve apresentar para o usuário o estado do sistema de posicionamento da antena e a previsão de horário para as próximas passagens de satélite, com dados adicionais como elevação máxima durante a passagem.

# Resumo do que foi realizado

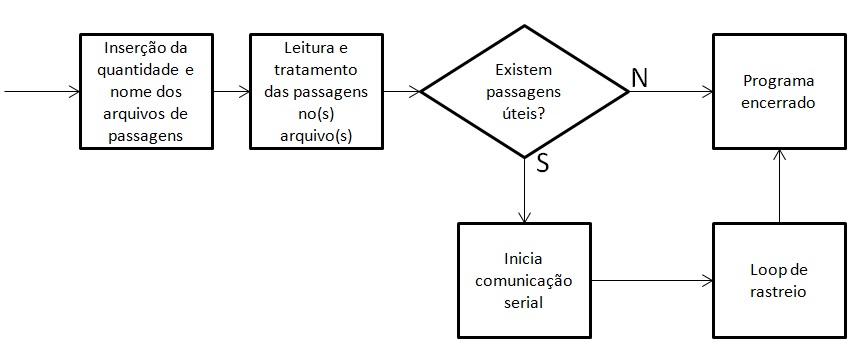
O novo software de controle e rastreio está sendo desenvolvido, uma parte já estando em funcionamento. O programa funciona como um sistema de mestre e escravo, onde o mestre é o PC e o escravo é um Arduino, que faz interface com os subsistemas de potência e de comando da estação. A comunicação entre o Arduino e o PC se dá através de uma porta USB, recebendo os comandos para ligar potência, enviar status da estação ao PC ou posicionar a antena. Para esta comunicação, foi desenvolvido e documentado um protocolo de comunicação serial. O código desenvolvido para o PC foi implementado na linguagem C/C++, e tem como objetivos carregar até três arquivos de texto com várias passagens, ordená-las cronologicamente e enviar serialmente as posições de referência das respectivas passagens no exato momento em que a antena deve apontar para posição enviada. O arquivo com as passagens utilizado neste programa é gerado no programa System Tool Kit (STK) da empresa AGI. O código desenvolvido para o Arduino realiza a comunicação serial com o PC, interpreta os comandos recebidos e os executa. Para o controle de posição, é utilizado um algoritmo de um controlador Proporcional Integrativo, sendo também possível o controle manual da posição da antena através de um *joystick*.

# Detalhamento dos progressos e mudanças necessárias

Algumas mudanças foram realizadas em relação ao plano de trabalho. Depois de algumas discussões, viu-se que não havia necessidade de desenvolver um programa que calcule as efemérides, devido a disponibilidade de diversos programas comerciais que realizam essa tarefa. Um deles é o STK, um software que o INPE já possui, que permite customizar um arquivo com várias passagens, incluindo informações adicionais como distância entre satélite e estação. Por isso, com o objetivo de simplificar a implementação, a ideia de calcular as efemérides foi substituída pela de gerar um arquivo texto com esses dados utilizando o STK e codificar um método para carregar os dados do arquivo texto gerado. Maiores detalhes sobre a codificação e descrição do algoritmo do arduino e do PC estão disponíveis em [3] e [4] respectivamente.

## Descrição do algoritmo do PC

O algoritmo do PC utiliza uma biblioteca que foi desenvolvida para criar objetos das passagens e armazenar em cada um as informações contidas nos arquivos carregados. Essa biblioteca é chamada Efem. Quando chegar a hora inicial de cada uma dessas passagens, as informações poderão ser recuperadas e tratadas como desejado. O Diagrama 1 ilustra a sequência geral de etapas que o programa do PC executa.

 **Diagrama 1 – Visão geral do software de controle e rastreio do PC.**

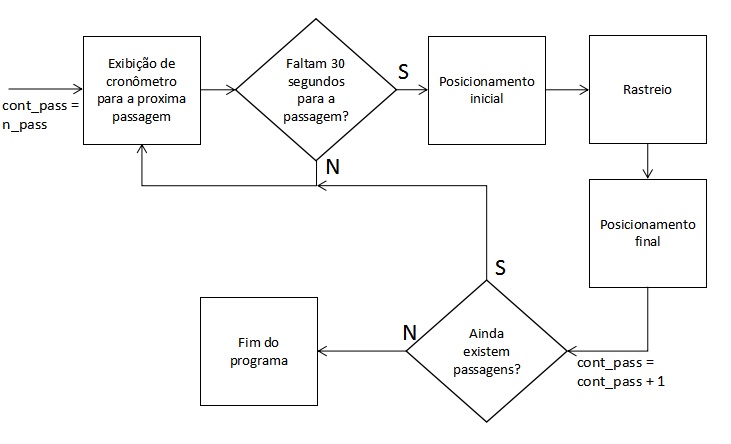
O programa inicia solicitando ao usuário a quantidade de arquivos e logo após os nomes desses arquivos a serem carregados. Essas informações devem ser inseridas pelo usuário do software, e inseridas através do teclado. Após a correta inserção, o PC faz a leitura e o armazenamento das informações contidas nos arquivos em objetos da classe Efem. Caso o usuário tenha solicitado que o PC carregue apenas um arquivo, após ser feita a leitura e o armazenamento das informações, as passagens já estarão em ordem cronológica. Caso contrário, as passagens não estarão em ordem, por isso o PC faz essa organização.

O passo seguinte é a verificação de passagens úteis em cada arquivo, e - caso existam – será executado o loop de rastreio. Os critérios que foram definidos para decidir se uma passagem é útil foram: a passagem não pode ter acabado; a passagem tem que ter elevação máxima de pelo menos 10°; e a passagem tem que ter pelo menos 3 minutos restantes. Tais critérios foram elaborados pensando também em passagens pegas depois do inicio (após um rastreio executado, por exemplo, quando o programa for para a próxima passagem e ela já estiver ocorrendo). Se durante a verificação uma passagem for considerada não útil, o PC irá pular para a seguinte.

Caso o arquivo inserido tenha passagens úteis, o PC irá iniciar a comunicação serial com o arduino, e esperar até a hora da primeira passagem. Para realizar a comunicação serial está sendo utilizada uma bilbioteca chamada “serial”, que foi obtida em [8]. Essa biblioteca foi testada e funcionou perfeitamente para os objetivos desse software.

A comunicação entre o arduino e o PC é feita via porta serial. O PC envia para o arduino os comandos SET, STATE, POWER. Além disso, existem três tipos de respostas: ACKNOWLEDGED e NOT ACKNOWLEDGED e a resposta ao comando STATE. O comando SET envia para o arduino uma string com as referências de azimute e elevação, codificadas em 16 bits cada um, além de um código de verificação de erros, o checksum. O comando STATE envia um byte específico ao arduino e o PC espera a resposta, que deve conter o azimute e elevação da antena, codificado em 16 bits, um byte codificado com o status dos relés da antena, e o checksum. O comando POWER envia um byte ao arduino, que faz com que ele ligue a potência do sistema, ou caso já esteja ligada, que ele a desligue.

Após a comunicação ser estabelecida, o PC entrará no *loop* de rastreio. O Diagrama 2 ilustra as etapas do que o PC executa quando entra nesse *loop*.

**Diagrama 2 – Loop de rastreio.**

Ao entrar no loop de rastreio o PC irá exibir um cronômetro com o tempo restante para a próxima passagem. Quando faltar 30 segundos para a passagem começar será feito o posicionamento inicial. Esse posicionamento consiste calcular posições sucessivas que em 30 segundos leve a antena para a posição inicial do rastreio a partir de sua posição no inicio do posicionamento. Para isso é feito o calculo da inclinação de uma reta, e a partir dela o calculo de uma reta que gere essas posições. Ao final desse posicionamento a antena estará na posição do inicio do rastreio.

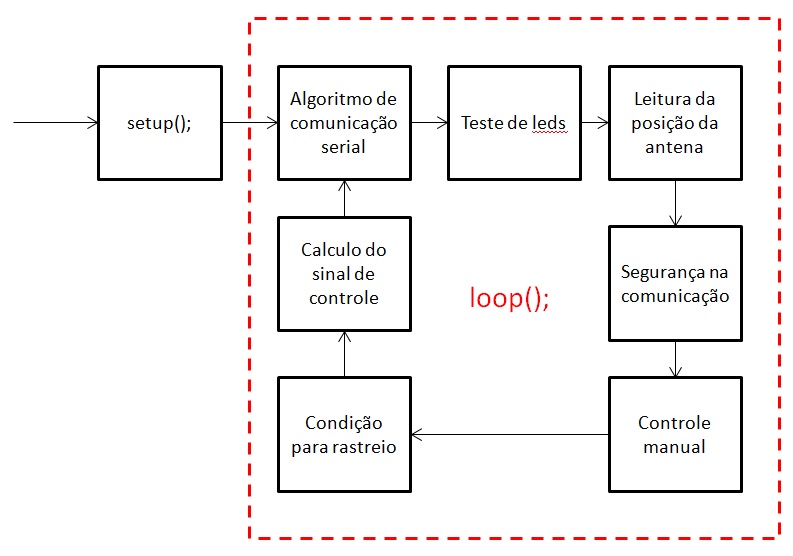
Durante o rastreio a antena estará de fato seguindo o satélite referente a passagem. O PC irá enviar para o arduino as posições de referência que foram obtidas no arquivo de passagens. Além disso, receberá do arduino o status do sistema (o status dos relés da gaveta de pilotage, do controle manual, da potência do sistema e o status dos *drivers* de azimute e elevação).

Ao final do rastreio o PC irá realizar o posicionamento final. Este posicionamento é feito de forma similar ao posicionamento inicial, entretanto o objetivo é levar a antena para a posição de 180° de azimute e 90° de elevação a partir de sua posição no final do rastreio. Este posicionamento também é feito durante 30 segundos.

Em todos esses processos, o ligamento da potência, envio das posições de referência e recebimento do status do sistema é feito através dos comandos descritos anteriormente.

## Descrição do algoritmo do arduino

O algoritmo foi estruturado de modo a realizar as várias tarefas necessárias para o funcionamento correto do sistema. O Diagrama 3 ilustra de modo global como funciona esse programa a partir do instante que o arduino é ligado.

**Diagrama 3 – Visão geral do código do arduino.**

O código geral do arduino é dividido em duas funções: a função *setup* e a função *loop.* A declaração de todas as variáveis e funções a serem utilizadas neste projeto foi feita antes da função setup. Essa função só é executada uma vez, no momento em que o arduino é inicializado. Ela geralmente é utilizada para configurar quais recursos do arduino serão utilizados, como: portas digitais, analógicas, porta serial, entre outras coisas. Neste projeto, na função *setup* é feita configuração da porta serial para uma taxa de 9600 bps, configuradas todas as portas digitais que são utilizadas e definido o valor estado inicial de algumas delas, como: porta digital que controla o ligamento da potência do sistema em “desligado”; todos os leds e displays iniciam apagados; e sinal de controle para os *drivers* inicial é 0V.

Após a função setup ser executada, a função *loop* é executada. A principal característica dessa função é que ela é executada infinitamente enquanto o arduino estiver ligado. Caso ele venha a ser desligado e ligado, o ultimo código enviado a ele é executado, primeiro executando a função *setup* e o que estiver declarado antes dela, e após a função *loop*. No Diagrama 1, a função loop é representada pela linha vermelha tracejada, mostrando todas as tarefas que são executadas nela. O código desse projeto como um todo foi elaborado tendo em mente essa funcionalidade da função loop de se repetir infinitamente.

A primeira parte da função loop a ser executada é o algoritmo de comunicação serial. Esse algoritmo foi feito com base no protocolo de comunicação desenvolvido, e funciona baseado nos comandos descritos nele. Os comandos existentes são SET, STATE, POWER, ACK e NACK. O arduino receberá esses comandos do PC, irá interpretá-los e executar o referido comando. Maiores detalhes estão disponíveis em [5].

Em seguida é feita a verificação da solicitação do teste de leds. Esse teste é feito através do botão na gaveta de pilotage, e só pode ser feito fora de rastreio. Se o botão for apertado, os leds da gaveta permanecerão acessos até que o botão seja solto.

O algoritmo seguinte a ser executado é a leitura da posição atual da antena. Essa posição é obtida através da leitura do conversor analógico para digital dos sensores de posição angular da antena (*resolvers*). Esse sensor possui 16 bits de precisão, por isso são utilizadas 16 portas digitais do arduino para realizar a leitura.

Um simples algoritmo de segurança foi desenvolvido. Sempre que o arduino entra em rastreio, ou seja, quando ele recebe o primeiro comando POWER, um contador de tempo é atualizado em uma variável. A partir daí, todos os comandos recebidos do PC atualizam esse contador. E constantemente é feita uma verificação: caso esse contador de tempo não tenha sido atualizado a mais que 3 segundos, a potência do sistema será desligada. Isso é feito como medida de segurança para o caso de a comunicação entre o PC e o arduino venha a ser perdida durante um rastreio (caso alguém puxe o cabo USB do PC, por exemplo).

Em seguida é feito a verificação do controle manual. Caso o interruptor da gaveta de pilotage seja acionado, será possível controlar a antena através do joystick da gaveta. Caso o sistema esteja desligado e o interruptor seja acionado, a potência do sistema será ligada e será possível controlar a antena através do joystick.

Após o controle manual ser verificado, o PC irá verificar a condição de rastreio. Para o arduino entender que está em rastreio, ele deverá receber um comando POWER do PC. Ao receber esse comando, a potência do sistema será ligada e, se o arduino receber comandos SET, será feito o calculo do sinal de controle. Três condições são necessárias para o calculo do sinal de controle: o arduino deve estar em rastreio (deve ter recebido o comando POWER); o controle manual não pode estar ativo; e o arduino deve ter recebido um comando SET. Essa ultima condição é feita para garantir um controle a cada 1 segundo, já que os comandos SET são enviados ao arduino a cada 1 segundo. O algoritmo que controla a antena é um Proporcional Integrativo (PI). Após a execução do algoritmo de controle, o algoritmo de comunicação é executado, e o arduino permanecerá no loop enquanto estiver ligado.

# Considerações finais

O progresso das atividades está de acordo com a proposta do trabalho. Apesar de ainda não estar em um formato final, já possuímos um software que realiza as funções básicas que foram propostas, de carregar as passagens desejadas e realizar rastreio. Com isso, podemos dizer que estamos no período de testes de avaliação do software que foi implementado até agora. A medida que ele for sendo utilizado poderemos verificar se existe algum erro de funcionamento ou possíveis funcionalidades a mais para serem incorporadas.

Uma possível atualização é a adaptação do software de controle sem interface gráfica para um com interface gráfica, que é mais amigável ao usuário que venha a utilizar o programa.

# REFERÊNCIAS

[1] QUEIROZ, K. I. P. M. Descrição Funcional do Subsistema de Controle e Rastreio da Estação Multimissão de Natal (EMMN), Instituto Nacional de Pesquisas Espaciais, Natal. 2006.

[2] JOTHA, L. EMMN-DT-030-V01 - Displays de Ângulos de AZ e EL - Tabela de Cabeação

[3] SARAIVA, A. B. D. Descrição funcional do algoritmo de controle e rastreio do arduino, Instituto Nacional de Pesquisas Espaciais, Natal. 2017.

[4] SARAIVA, A. B. D. Descrição funcional do algoritmo de controle e rastreio do PC, Instituto Nacional de Pesquisas Espaciais, Natal. 2017.

[5] SARAIVA, A. B. D. Protocolo de comunicação entre o PC e o arduino - Descrição, Instituto Nacional de Pesquisas Espaciais, Natal. 2017.

[6] SARAIVA, A. B. D. Especificação do arquivo de efemérides a ser utilizado no novo programa de controle e rastreio, Instituto Nacional de Pesquisas Espaciais, Natal. 2017.

[7] <https://www.codeproject.com/Articles/992/Serial-library-for-C.> Acesso em: 03/04/2017