Instituto nacional de pesquisas espaciais

Estação Multi-Missão de Natal

Guia do programador

Estação Multi-Missão de Natal

Guia do programador

Escrito por Felipe Cortez de Sá <[felipe.cortez@crn.inpe.b](mailto:felipe.cortez@crn.inpe.b)r>

Revisado por José Marcelo <jmarcelo@crn.inpe.br>

**Sumário**

[Diretórios 4](#_Toc517449265)

[Bibliotecas utilizadas 4](#_Toc517449266)

[Lendo e gerando documentação 4](#_Toc517449267)

[Estilo de código 5](#_Toc517449268)

[Andamento do projeto 5](#_Toc517449269)

[Funcionalidades implementadas 5](#_Toc517449270)

[Funcionalidades a serem implementadas 5](#_Toc517449271)

[Possíveis melhorias 5](#_Toc517449272)

[Organização do código 6](#_Toc517449273)

[Interface gráfica 6](#_Toc517449274)

[Classes 8](#_Toc517449275)

[Arquivos de configuração 9](#_Toc517449276)

[emmn-gui.pro 9](#_Toc517449277)

[emmn.pri 9](#_Toc517449278)

[Miscelânea 9](#_Toc517449279)

# Diretórios

|  |  |
| --- | --- |
| antena-arduino | Contém o código Arduino |
| build | Contém o executável gerado na compilação |
| docs | Especificações e relatórios relacionados à EMMN |
| doxygen | Documentação gerada automaticamente a partir de anotações no código |
| spg4 | Biblioteca para cálculo de efemérides com classe DateTime modificada |
| src | |  |  | | --- | --- | | emmn-gui | Código-fonte do projeto | | emmn-tests | Testes unitários | | includes | Arquivos .h (cabeçalhos) das bibliotecas | | libs | Arquivos .a (bibliotecas estáticas) | |

# Bibliotecas utilizadas

* SGP4: recebe TLE e gera posições de azimute e elevação
* SolTrack: permite rastreios do sol
* CSerial: comunicação com porta serial

# Lendo e gerando documentação

O projeto é documentado utilizando a ferramenta [Doxygen](http://www.stack.nl/~dimitri/doxygen/) através de comentários iniciados por sinais de exclamação (e.g.: //! Caixa de diálogo de configurações). Esses comentários permitem especificação de parâmetros como descrições curtas e longas, descrição de argumentos de um método etc. É possível gerar uma documentação interativa HTML a partir desses comentários através do programa Doxywizard:

* Utilizando o *prompt* de comando (doxywizard.exe ou doxywizard.exe Doxyfile)
* Através de o Menu Iniciar: Digite Doxywizard e localize o Doxyfile na pasta principal do projeto

A documentação gerada pode ser acessada através do arquivo doxygen/html/index.html.

Para gerar um PDF da documentação — útil para colocar na biblioteca do INPE —, vá para a pasta doxygen/latex e execute make.bat no explorador de arquivos do Windows ou através do programa make utilizando o *prompt* de comando ou Cmder.

# Estilo de código

Você é livre para utilizar o estilo que preferir, mas caso queira continuar usando a formatação atual, siga estes padrões:

* Quatro espaços para indentação
* Espaço entre if, while, for etc. e (
* { na mesma linha: if (ele > 180) {, exceto em construtores com [*member* *initializer lists*](http://en.cppreference.com/w/cpp/language/initializer_list)
* *switches* são formatados de acordo com a [convenção do Java](http://www.oracle.com/technetwork/java/javase/documentation/codeconventions-142311.html#468):

switch (var) {  
case 1:  
 // alguma coisa  
case 3:  
 // outra coisa  
default:  
 // mais uma coisa  
}

* Código em inglês, comentários em português
* Mensagens de *commit* do Git em português

# Andamento do projeto

## Funcionalidades implementadas

* Adicionar e remover satélites para rastreio
* Definir ordem de prioridade de satélites para rastreio
* Atualizar TLE automaticamente todo dia
* Enviar comandos de apontamento (azimute e elevação) para o Arduino
* Controlar manualmente a antena (azimute e elevação determinados pelo usuário)
* Gravar histórico de posicionamento da antena em arquivos .csv

## Funcionalidades a serem implementadas

* Definição de parâmetros adicionais para cada satélite rastreado

## Possíveis melhorias

Na documentação Doxygen é possível ver uma lista de tarefas extraídas dos comentários @todo. Aqui, destacam-se algumas delas por ordem de relevância:

* Atualizar lista de TLE obtidas do *SpaceTrack* numa *thread* diferente. A maneira atual bloqueia a thread da interface
* Remover userGeo da classe Tracker
* Instanciar Settings apenas na inicialização do programa
* Colocar velocidade máxima e aceleração da antena nas configurações
* A classe Control utiliza a biblioteca CSerial, mas o Qt possui funções para comunicação com porta serial no cabeçalho QSerialPort. A mudança para a biblioteca do Qt eliminaria uma dependência e possivelmente aumentaria a portabilidade
* Interação com o mouse no widget gráfico de próximas passagens
* Colocar localização geográfica nas configurações

# Organização do código

Uma descrição mais detalhada de cada classe pode ser vista na documentação Doxygen, localizada na pasta doxygen/html/índex.html

## Interface gráfica

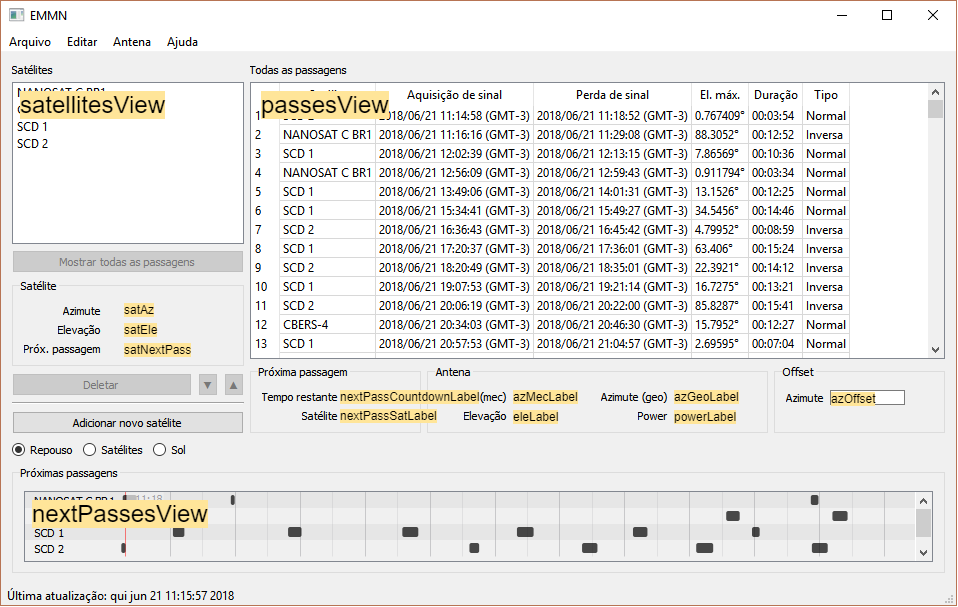


Figura 1 - Objetos da visão principal



Figura 2 - Hierarquia de classes de interface gráfica

Quatro janelas da interface gráfica são construídas a partir de *forms* do Qt nos arquivos mainwindow.ui, manualcontroldialog.ui, settingsdialog.ui e trackerdialog.ui. Para chamar estes *forms* através de códigos-fonte, os includes estão na forma #include "ui\_{form}.h", como #include "ui\_settingsdialog.h".

## Classes

|  |  |
| --- | --- |
| MainWindow | Janela principal do programa. A classe faz a inicialização de todo o sistema e contém timers para leitura de estado da antena. |
| AddTrackerDialog | Janela de diálogo para inserção de um satélite |
| JoystickWidget | Widget para controle da antena através de um joystick virtual. |
| ManualControlDialog | Contém o JoystickWidget e campos para posicionar a antena no azimute e elevação desejados. |
| Control | Contém toda a lógica de manipulação física da antena. Envia e recebe comandos de *hardware*. O envio periódico de mensagens ao *hardware* é feito através de um *timer*. |
| Helpers | Contém funções auxiliares divididas em dois *namespaces*. Settings contém funções para carregar e salvar configurações. Helpers contém funções de conversões matemáticas, formatação de data e horários, leituras de arquivo etc. |
| Logger | Classe responsável pelo registro periódico de informações contendo posição da antena em azimute e elevação, velocidade etc. |
| Network | Classe responsável pela conexão com a Internet para atualizar diariamente dados de TLE. |
| NextPassesView | Visão gráfica de próximas passagens localizada na parte inferior do programa. |
| Serial | Biblioteca CSerial para comunicação com *hardware*. |
| SettingsDialog | Janela de diálogo para configurações do programa, incluindo porta do Arduino e especificação de fuso horário. |
| SolTrack | Informa localização do sol permitindo rastreio automático. |
| Tracker |  |
| TrackerListModel |  |

## Arquivos de configuração

### emmn-gui.pro

Arquivo atualizado automaticamente pelo Qt toda vez que uma classe é criada. Informa ao compilador quais são os *includes,* bibliotecas do Qt, tipo de projeto, cabeçalhos e arquivos de código-fonte utilizados pelo projeto.

### emmn.pri

Arquivo de includes do Qt contendo configurações que podem ser adicionadas a múltiplos arquivos .pro.

# Miscelânea

* Ao mudar o diretório de build pelo Qt, é preciso copiar os arquivos .dll do ssh (libeay32.dll e ssleay32.dll) para o diretório que contém o executável.
* É utilizado o sistema de controle de versão Git.