

Documentação do script de automatização do 7GHz

Edison Neto

October 4, 2018

Contents

1	Introdução	4
2	Funcionamento	4
3	Biblioteca do SkyX	4
3.1	Principais classes	4
3.2	Exemplos	4
3.2.1	sky6ObjectInformation	4
	Property()	5
3.2.2	sky6StarChart	5
	Find()	5
3.2.3	sky6RASCOMTele	5
	Connect(void)	5
	Disconnect(void)	6
	Abort(void)	6
	SlewToRaDec()	6
	GetRaDec(void)	6
	Park(void)	6
	ParkAndDoNotDisconnect(void)	7
	Unpark(void)	7
	IsConnected	7
	IsParked	7
	IsTracking	7
	dRa	7
	dDec	7
3.2.4	TextFile	7
	createNew()	7
	write()	8
	openForAppend()	8
	close(void)	8
3.2.5	Não relacionadas com as classes	8
	String()	8
	print()	8
	A variável Out	8
4	Funções do script	9
4.1	Funções de utilidade	9
4.1.1	Escrevendo no log e no debugger a mesma mensagem	9
4.1.2	Escrevendo no debugger e no RunJavaScript	9
4.1.3	Encontrando um objeto e pegando informações sobre ele	9
4.1.4	Pegando a ascensão reta e a declinação do objeto	10
4.2	Funções de controle	10
4.2.1	Conexão	10

4.2.2	Inicialização	11
4.2.3	Flip	11
4.2.4	Desligamento	11
4.2.5	Reconexão	12
4.2.6	Reinicialização do tracking	12
5	Guia de estilo do código	12
5.1	Indentação	12
5.2	Posicionamento das chaves	13
5.3	Nomeando funções	13

1 Introdução

O script está escrito em Javascript(ECMAScript), usando a biblioteca do TheSkyX. Características mais recentes do Javascript, como programação funcional, ou definição de classes, não estão presentes no SkyX.

2 Funcionamento

A rotina está dentro de um loop infinito que fica pegando o horário atual do computador e comparando com os horários pré-determinados para iniciar algum dos processo.

Antes do início do loop são definidos os horários (UT) para ligar, fazer o flip e desligar. Para o início, é verificada se a hora é exatamente a do horário de inicialização. Antes de fazer o Slew, é necessário usar a função FindHome. Como não há uma forma de saber se o telescópio já fez o home ou não, a função FindHome deve ser executada sempre na inicialização. O flip, como a inicialização, é realizado precisamente no horário determinado. O desligamento ocorre se o tracking estiver sendo realizado e se a hora atual for maior ou igual a hora de desligamento.

Se a conexão for perdida há a possibilidade dela ser recuperada e que o telescópio volte a sua rotina normal. Entretanto, o problema causado pela perda de conexão pode não ser resolvido, e há a possibilidade de que seja necessária um reconexão manual.

A documentação das funções usa o padrão JSDoc

3 Biblioteca do SkyX

3.1 Principais classes

- sky6ObjectInformation - Informações dos objetos
- sky6StarChart - Acesso aos aspectos visuais do SkyX
- sky6RASCOMTele - Controle físico da montagem
- TextFile - Manipulação de arquivos

3.2 Exemplos

3.2.1 sky6ObjectInformation

Dá o acesso ao banco de dados do SkyX. Podendo pegar diversas informações sobre o objeto sendo observado.

Property()

Argumentos:

- *number* number - Representa um certa informação do objeto.

Há um total de 189 informações separadas nessa função. A função em si não retorna nada, o valor fica armazenado na propriedade *ObjInfoPropOut*, como escrito no exemplo.

Exemplo:

```
1 sky6ObjectInformation.Property(55);
2 print(sky6ObjectInformation.ObjInfoPropOut);
```

As duas propriedades que são usadas no script, são as seguintes:

- 54: Ascensão reta
- 55: Declinação

3.2.2 sky6StarChart

Controle da parte visual do SkyX. Basicamente com essa classe é possível fazer o que se faria clicando no TheSkyX.

Find()

Argumentos:

- *string* objectName - O nome do objeto a ser procurado.

Procura pelo objeto dado.

Exemplo:

```
1 sky6StarChart.Find("Sun");
```

Exemplo usando o sky6ObjectInformation.Property():

```
1 // Procura pelo sol.
2 sky6StarChart.Find("Sun");
3 // Prepara no ObjInfoPropOut o valor da declinacao.
4 sky6ObjectInformation.Property(55);
5
6 print(sky6ObjectInformation.ObjInfoPropOut + "\n")
```

3.2.3 sky6RASCOMTele

Dá o controle físico da montagem. Para coisas como o slew ou parking.

Connect(void)

Inicia a comunicação entre a montagem e o SkyX.

Disconnect(void)

Termina a conexão entre a montagem e o SkyX.

Abort(void)

Aborta a operação que estiver sendo realizada.

SlewToRaDec()

Argumentos:

- *number* TargetRa - A ascensão reta;
- *number* TargetDec - A declinação;
- *string* targetObject - O nome do objeto.

Aponta o telescópio para a coordenada dada.

```
1 sky6RASCOMTele.Connect();
2
3 var targetObject = "Sun";
4 sky6StarChart.Find(targetObject);
5
6 sky6ObjectInformation.Property(54);
7 var targetRA = sky6ObjectInformation.ObjInfoPropOut;
8
9 sky6ObjectInformation.Property(55);
10 var targetDec = sky6ObjectInformation.ObjInfoPropOut;
11 sky6RASCOMTele.SlewToRaDec(targetRa, TargetDec, targetObject);
```

GetRaDec(void)

Pega a declinação e a ascensão reta atual, e prepara os valores nas variáveis dRa e dDec.

Exemplo:

```
1 sky6RASCOMTele.Connect();
2 sky6RASCOMTele.GetRaDec();
3
4 print(String(sky6RASCOMTele.dRa) + " | " + String(sky6RASCOMTele.
    dDec));
```

Park(void)

Faz o slew para a posição de parking, e finaliza a conexão com o TheSky6.

ParkAndDoNotDisconnect(void)

Tem quase o mesmo funcionamento que a função ‘Park’. A diferença é que essa função não finaliza a conexão entre o telescópio e o TheSky6. Para fazer outro Slew depois de usar esta função é necessário utilizar a função ‘Unpark’ antes.

Unpark(void)

Tira o telescópio do estado de parked.

IsConnected

Tem o valor zero se o telescópio não estiver conectado.

IsParked

Tem o valor zero se o telescópio não estiver na posição de parking.

IsTracking

Tem o valor zero se o telescópio não estiver fazendo o tracking.

dRa

A ascensão reta atual.

dDec

A declinação atual.

3.2.4 TextFile

Classe usada para manipulação básica de arquivos.

createNew()

Cria um arquivo txt. Os arquivos são obrigatoriamente na pasta */Meus Documentos/Software Bisque/TheSkyX Professional Edition/ScriptFiles*, o local dos não pode ser modificado. O nome do arquivo só pode conter letras e números.

Argumentos:

- *string* filename - O nome do arquivo (sem a extensão).

write()

Escreve uma string no arquivo.

Argumentos:

- *string* text - String a ser escrita no arquivo.

openForAppend()

Abre o arquivo de forma a anexar novos conteúdos depois do que já está escrito. Caso o arquivo não tiver sido criado e essa função for usada, o arquivo será criado.

Argumentos:

- *string* filename - O nome do arquivo

close(void)

Fecha o arquivo salvando as modificações feitas.

3.2.5 Não relacionadas com as classes

String()

Argumentos:

- *number* int - Uma variável numérica qualquer.

Transforma um número em uma string. Essa função funciona como o método toString() do javascript(que curiosamente não funciona no SkyX).

Exemplo:

```
1 var int = 2;  
2 print(String(int));
```

print()

Argumentos:

- (*string, number*) text - Uma variável qualquer.

Essa função escreve nos logs do debugger. Ela escreve enquanto o programa roda, diferentemente da variável Out.

A variável Out

Essa variável armazena tudo que será escrito na tela do "Run Java Script" no SkyX. Ela só é escrita quando o script acaba de rodar. É possível escrever nessa tela em tempo de execução com a classe RunJavaScriptOutput.

4 Funções do script

Algumas funções foram reescritas com tratamento de erro, e com finalidade mais relacionada à automatização do 7GHz.

4.1 Funções de utilidade

4.1.1 Escrevendo no log e no debugger a mesma mensagem

Por diversas vezes há a necessidade de escrevermos no log e no debugger a mesma mensagem, com o horário. Devido a essa necessidade há uma função implementada que abre o arquivo do log do dia e escreve, tanto do arquivo quanto no debugger, a mensagem dada seguida pelo horário daquele exato momento.

```
1 function WriteFileAndPrint(text)
2 {
3     // Gera o nome do arquivo.
4     var filename = setFileName();
5     try {
6         TextFile.openForAppend(filename);
7         var horario = getHorario();
8         TextFile.write(text + " " + horario + "\n");
9         print(text + " " + horario);
10        TextFile.close();
11    } catch (texterr) {
12        print("Erro ao editar o log");
13    }
14 }
```

4.1.2 Escrevendo no debugger e no RunJavaScript

Para escrever no debugger e na janela RunJavaScript a mesma mensagem em tempo de execução.

```
1 function PrintAndOut(text)
2 {
3     print(text);
4     RunJavaScriptOutput.writeLine(text);
5 }
```

4.1.3 Encontrando um objeto e pegando informações sobre ele

Se é preciso pegar algum tipo de informação sobre um objeto x, a função Find(x) vai procurar esse objeto e escrever todas as informações que podem ser adquiridas sobre o mesmo.

```
1 function Find(objectName)
2 {
3     // Numero de propriedades que um objeto tem.
4     var props = 189;
5
6     try {
7         sky6StarChart.Find(objectName);
```

```

8   } catch (finderr) {
9       PrintAndOut("Objeto nao encontrado");
10      return false;
11  }
12
13  for (var prop = 0; prop < props; ++prop) {
14      if (sky60ObjectInformation.PropertyApplies(prop) != 0) {
15          sky60ObjectInformation.Property(prop);
16
17          PrintAndOut(sky60ObjectInformation.ObjInfoPropOut + "\n");
18      }
19  }
20 }

```

4.1.4 Pegando a ascensão reta e a declinação do objeto

Pega a ascensão reta e a declinação de algum objeto qualquer dentro do limite preestabelecido.

```

1  function GetRADec(object)
2  {
3      if (!Sky6IsConnected()) {
4          WriteFileAndPrint("Erro de conexao tentando executar a funcao
5              GetRADec ");
6          return false;
7      }
8      try {
9          sky6StarChart.Find(object);
10     } catch (finderr) {
11         WriteFileAndPrint("Erro durante o find.\n" + finderr.message +
12             " ");
13         return false;
14     }
15     sky60ObjectInformation.Property(54);
16     var targetRA = sky60ObjectInformation.ObjInfoPropOut;
17     sky60ObjectInformation.Property(55);
18     var targetDec = sky60ObjectInformation.ObjInfoPropOut;
19
20     return {"ra": targetRA, "dec": targetDec};
21 }

```

4.2 Funções de controle

4.2.1 Conexão

Inicia a conexão entre o SkyX e a montagem e cria o arquivo de log do dia. Essa função deve ser executada quando o SkyX não está conectado e for exatamente 11:00(UT), ou o horário escolhido para o início da execução. Esse processo é muito rápido comparado com outras operações de controle, demorando não mais de 1 segundo.

```

1  function Connect_c()

```

```

2 {
3   var time = getNow();
4   var horario = getHorario();
5
6   print("Conectado as " + horario);
7   ConnectTelescope();
8
9   var filename = setFileName();
10  TextFile.createNew(filename);
11  TextFile.write(String(time.day) + "/" + String(time.month) + "/"
12    + String(time.year) + "\n");
13  TextFile.write("Conectado as " + horario + "\n");
14  TextFile.close();
15 }

```

4.2.2 Inicialização

Inicia o rastreamento do sol. Essa função deve ser executada quando o SkyX está conectado e for exatamente 11:00(UT). Pelo fato do processo de conexão ser muito rápido, não há necessidade de iniciar 1 ou 2 minuto(s) depois da conexão.

```

1 function Initialize_c()
2 {
3   sky6GRASCOMTele.FindHome();
4   var propriedade = GetRADec("Sun");
5
6   WriteFileAndPrint("Iniciou o slew as");
7   SlewTelescopeTo(propriedade.ra, propriedade.dec, "Sun");
8
9   WriteFileAndPrint("Iniciou o rastreamento as");
10 }

```

4.2.3 Flip

Faz o flip, basicamente reiniciando o rastreamento. A única diferença de código entre a função Initialize_c, é a inutilidade da função FindHome, visto que sua execução somente é necessária uma única vez por dia (desconsiderando problemas de conexão).

```

1 function Flip_c()
2 {
3   var propriedade = GetRADec("Sun");
4   WriteFileAndPrint("Iniciou o slew as");
5   SlewTelescopeTo(propriedade.ra, propriedade.dec, "Sun");
6
7   WriteFileAndPrint("Completo o flip as");
8 }

```

4.2.4 Desligamento

Desliga o rastreamento primeiro e depois vai para a posição de parking, desconectando logo em seguida. É executada quando o SkyX está conectado e já passou das 20:00(UT).

```

1 function TurnOff_c()
2 {
3     SetTelescopeTracking(0, 1, 0, 0);
4     WriteFileAndPrint("Desligou o rastreamento as");
5
6     ParkTelescope();
7     WriteFileAndPrint("Parking finalizado as");
8
9     WriteFileAndPrint("Desconectado as");
10 }

```

4.2.5 Reconexão

Reconecta o telescópio e reinicia o rastreamento. Também executa a função FindHome, já que se o script for (re)iniciado depois das 11:00, este processo pode não ter sido realizado. É executada quando o SkyX não está conectado e a hora atual está entre o horário de execução.

```

1 function Reconnect_c()
2 {
3     WriteFileAndPrint("(Re)conectando as");
4     ConnectTelescope();
5     sky6RASCOMTele.FindHome();
6     // Verifica se o Tracking nao esta ocorrendo.
7     if (sky6RASCOMTele.IsTracking == 0) {
8         RestartTracking_c();
9     }
10 }

```

4.2.6 Reinicialização do tracking

Reinicia o rastreamento. É executada na função de reconexão e quando o SkyX está conectado, não está fazendo o tracking e a hora atual está entre o horário de execução.

```

1 function RestartTracking_c()
2 {
3     var propriedade = GetRADec("Sun");
4
5     WriteFileAndPrint("Iniciou o slew as");
6     SlewTelescopeTo(propriedade.ra, propriedade.dec, "Sun");
7
8     WriteFileAndPrint("Reiniciou o rastreamento as");
9 }

```

5 Guia de estilo do código

5.1 Indentação

O código usa espaços com 2 caracteres de largura.

5.2 Posicionamento das chaves

A forma correta é colocar a chave de abertura por último na linha, e colocar a chave de fechamento primeiro.

```
1 if (something === true) {  
2     print(something);  
3 }
```

Para funções coloque a chave embaixo da próxima linha.

```
1 function myFunction()  
2 {  
3     return 0;  
4 }
```

5.3 Nomeando funções

As funções que usam alguma classe do SkyX são nomeadas usando UpperCamel-Case, já as que não usam são nomeadas usando lowerCamelCase. As função principais de controle, são nomeadas usando UpperCamelCase e com o sufixo ‘_c’.