Documentação do script de automatização do 7GHz

Edison Neto

18 de Fevereiro de 2019

Conteúdo

1 Introdução

O script está escrito em Javascript, só é possível executá-lo usando o interpretador do SkyX.

2 Funcionamento geral

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 Funções do script

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

3.1 Funções de utilidade geral

3.1.1 Logging

Para fazer o logging existem três funções diferentes

```
1  function WriteLog(level, text)
2  {
3      var filename = setFileName();
4      try {
5          TextFile.openForAppend(filename);
6          var formattedTime = getFormattedTime();
7          var header = "[" + level + " - " + formattedTime  + "] ";
8          TextFile.write(header + text + "\n");
9          print(header + text);
10         TextFile.close();
11     } catch (texterr) {
12         PrintAndOut("Erro ao editar o log. " + texterr.message + "
               (WriteLog)");
```

```
13      }
14  }
```

3.1.2 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      var level = "WARNING";
4      var formattedTime = getFormattedTime();
5      var header = "[" + level + " - " + formattedTime  + "] ";
6      print(header + text);
7      RunJavaScriptOutput.writeLine(header + text);
8  }
```

3.1.3 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          WriteLogError("Erro de conexao (GetRaDec)");
5          return false;
6      }
7
8      try {
9          sky6StarChart.Find(object);
10     } catch (finderr) {
11         WriteLogError("Erro durante o find. " + finderr.message + "
               (GetRaDec)");
12         return false;
13     }
14
15     sky6ObjectInformation.Property(54);
16     var targetRA = sky6ObjectInformation.ObjInfoPropOut;
17     sky6ObjectInformation.Property(55);
18     var targetDec = sky6ObjectInformation.ObjInfoPropOut;
19
20     return {"ra": targetRA, "dec": targetDec};
21 }
```

3.2 Funções de controle

3.2.1 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.

```
function Connect_c()
{
    var time = getTimeNow();
    var formattedTime = getFormattedTime();

    ConnectTelescope();

    var filename = setFileName();
    TextFile.createNew(filename);
    TextFile.write(String(time.day) + "/" + String(time.month) + "/
        " + String(time.year) + "\n");
    var header = "[INFO - " + formattedTime + "] ";
    TextFile.write(header + "Conectado\n");
    print(header + "Conectado\n");
    TextFile.close();
}
```

3.2.2 Inicialização

Inicia o rastreamento do sol. Essa função dever 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.

```
function Initialize_c()
{
    if (sky6RASCOMTele.IsParked != 0) {
        sky6RASCOMTele.Unpark();
    }
    sky6RASCOMTele.FindHome();
    var props = GetRADec("Sun");

    WriteLogInfo("Iniciou o slew (Initialize_c)");
    SlewTelescopeToRaDec(props.ra, props.dec, "Sun");

    WriteLogInfo("Iniciou o rastreamento (Initialize_c)");
}
```

3.2.3 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).

```
function Flip_c()
{
    var props = GetRADec("Sun");
    WriteLogInfo("Iniciou o slew (Flip_c)");
    SlewTelescopeToRaDec(props.ra, props.dec, "Sun");

    WriteLogInfo("Completou o flip (Flip_c)");
}
```

3.2.4 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      WriteLogInfo("Desligou o rastreamento (TurnOff_c)");
5
6      ParkTelescope();
7      WriteLogInfo("Desconectado (TurnOff_c)");
8  }
```

3.2.5 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      WriteLogInfo("(Re)conectado (Reconnect_c)");
4      ConnectTelescope();
5      SetTelescopeTracking(0, 1, 0, 0);
6      sky6RASCOMTele.FindHome();
7      RestartTracking_c();
8  }
```

3.2.6 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 props = GetRADec("Sun");
4
5      WriteLogInfo("Iniciou o slew (RestartTracking_c)");
6      SlewTelescopeToRaDec(props.ra, props.dec, "Sun");
7
8      WriteLogInfo("Reiniciou o rastreamento (RestartTracking_c)");
9  }
```

3.2.7 Calibração

Aponta para o céu somando 20 graus na altitude da observação atual. A calibração é feita duas vezes ao dia, uma hora antes do flip e uma hora depois.

```
function CalibrateTelescope_c()
{
    WriteLogInfo("Calibracao iniciada (CalibrateTelescope_c)")

    var delta = 20;
    var props = GetAzAlt();
    WriteLogInfo("Azimute atual: " + props.az + " | Altitude atual:
         " + props.alt);
    var newAz = props.az + delta;
    WriteLogInfo("Azimute futuro: " + newAz + " | Altitude futura:
        " + props.alt);

    SlewTelescopeToAzAlt(newAz, props.alt, "");
}
```

4 Guia de estilo do código

4.1 Indentação

O código usa espaços com 4 caracteres de largura.

4.2 Posicionamento das chaves

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

```
if (something === true) {
  print(something);
}
```

Para funções coloque a chave embaixo da próxima linha.

```
function myFunction()
{
  return 0;
}
```

4.3 Nomeando funções

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

5 Biblioteca do SkyX

5.1 Principais classes

- sky6ObjectInformation - Informações dos objetos

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

5.2 Exemplos

5.2.1 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

5.2.2 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():

```
// Procura pelo sol.
sky6StarChart.Find("Sun");
// Prepara no ObjInfoPropOut o valor da declinacao.
sky6ObjectInformation.Property(55);

print(sky6ObjectInfomation.ObjInfoPropOut + "\n")
```

5.2.3 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.

```
sky6RASCOMTele.Connect();

var targetObject = "Sun";
sky6StarChart.Find(targetObject);

sky6ObjectInformation.Property(54);
var targetRA = sky6ObjectInformation.ObjInfoPropOut;

sky6ObjectInformation.Property(55);
var targetDec = sky6ObjectInformation.ObjInfoPropOut;
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) + " | ");
5 print(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.

5.2.4 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.

5.2.5 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.

6 O código

6.1 Versão 1.5

```
1  /*
2   * Version: 1.5
3   */
4
5  /**
6   * Confirma se o SkyX tem conexao com a montagem.
7   *
8   * @returns {boolean} false se nao estiver conectado.
9   *                    true se estiver conectado.
10  */
11 function Sky6IsConnected()
12 {
13     if (sky6RASCOMTele.IsConnected == 0) {
14         return false;
15     }
16     return true;
17 }
18
19 /**
20  * Estabiliza a conex o com a montagem.
21  *
22  * @returns {boolean} false caso algum erro aconte a.
23  */
24 function ConnectTelescope()
25 {
26     try {
27         sky6RASCOMTele.Connect();
28     } catch (connerr) {
29         WriteLogError("Erro de conexao com a montagem. " + connerr.
               message + " (ConnectTelescope)");
30         return false;
31     }
32
```

```
33      return true;
34  }
35
36  /**
37   * Liga/Desliga o tracking.
38   *
39   * @param {number} IOn - O n mero que desliga ou liga o tracking.
40   *                               0 - desliga
41   *                               1 - liga
42   *
43   * @param {number} IIgnoreRates - O n mero que especifica se
       para o
44   *                                 telesc pio usar a taxa de
       tracking atual.
45   *                               0 - Ignora os valores de dRaRate e
       dDecRate
46   *                               1 - Usa os valores de dRaRate e
       dDecRate
47   *
48   * @param {number} dRaRate - Especifica a ascens o reta a ser
       usada.
49   *                           S      utilizada se IIgnoreRates for
       igual    1.
50   *
51   * @param {number} dDecRate - Especifica a declina  o a ser usada
       .
52   *                           S      utilizada se IIgnoreRates for
       igual    1.
53   *
54   * @returns {boolean} false se a montagem n o estiver conectada.
55   */
56  function SetTelescopeTracking(IOn, IIgnoreRates, dRaRate, dDecRate)
57  {
58      if (!Sky6IsConnected()) {
59          return false;
60      }
61
62      sky6RASCOMTele.SetTracking(IOn, IIgnoreRates, dRaRate, dDecRate
            );
63      return true;
64  }
65
66  /**
67   * Faz o slew para um determinado objeto dados sua ascens o reta e
         declina  o.
68   *
69   * @param {number} dRa - ascens o reta.
70   * @param {number} dDec - declina  o.
71   * @param {string} targetObject - Objeto em quest o.
72   *
73   * @returns {boolean} true se tudo tiver ocorrido corretamente.
74   */
75  function SlewTelescopeToRaDec(dRa, dDec, targetObject)
76  {
77      if (!Sky6IsConnected()) {
78          WriteLogError("Telescopio nao conectado (
                SlewTelescopeToRaDec)");
```

```
79          return false;
80      }
81
82      try {
83          sky6RASCOMTele.SlewToRaDec(dRa, dDec, targetObject);
84          return true;
85      } catch (slewerr) {
86          WriteLogError("Falha durante o slew. " + slewerr.message +
                " (SlewTelescopeToRaDec)");
87          return false;
88      }
89  }
90
91  /**
92   * Faz o slew para um determinado objeto dados azimute e altitude.
93   *
94   * @param {number} az - Azimute.
95   * @param {number} alt - Altitude.
96   * @param {string} targetObject - Objeto em quest o.
97   *
98   * @returns {boolean} true se tudo tiver ocorrido corretamente.
99   */
100 function SlewTelescopeToAzAlt(az, alt, targetObject)
101 {
102     if (!Sky6IsConnected()) {
103         WriteLogError("Telescopio nao conectado. (
                SlewTelescopeToAzAlt)");
104         return false;
105     }
106
107     try {
108         sky6RASCOMTele.SlewToAzAlt(az, alt, targetObject);
109         return true;
110     } catch (slewerr) {
111         WriteLogError("Falha durante o slew. " + slewerr.message +
                " (SlewTelescopeToAzAlt)");
112         return false;
113     }
114 }
115
116 /**
117  * Leva o telesc pio para a posi   o de parking.
118  *
119  * @returns {boolean} true se tudo tiver ocorrido corretamente.
120  */
121 function ParkTelescope()
122 {
123     if (!Sky6IsConnected()) {
124         return false;
125     }
126
127     if (sky6RASCOMTele.IsParked != 0) {
128         sky6RASCOMTele.Park();
129         WriteLogInfo("Parking finalizado (ParkTelescope)");
130         return true;
131     }
132 }
```

```
133
134 /**
135  * Procura pelo objeto dado e pega a ascens o reta e a
        declina  o dele
136  * no momento.
137  *
138  * @param {string} object - Nome do objeto a ser encontrado.
139  *
140  * @returns {object} Um objeto com a ascens o reta e a
        declina  o.
141  */
142 function GetRADec(object)
143 {
144     if (!Sky6IsConnected()) {
145         WriteLogError("Erro de conexao (GetRaDec)");
146         return false;
147     }
148
149     try {
150         sky6StarChart.Find(object);
151     } catch (finderr) {
152         WriteLogError("Erro durante o find. " + finderr.message + "
                (GetRaDec)");
153         return false;
154     }
155
156     sky6ObjectInformation.Property(54);
157     var targetRA = sky6ObjectInformation.ObjInfoPropOut;
158     sky6ObjectInformation.Property(55);
159     var targetDec = sky6ObjectInformation.ObjInfoPropOut;
160
161     return {"ra": targetRA, "dec": targetDec};
162 }
163
164 /**
165  * Pega o azimute e a altitude do objeto sendo observado no momento
        .
166  *
167  * @returns {object} Um objeto com o azimute e a altitude.
168  */
169 function GetAzAlt()
170 {
171     if (!Sky6IsConnected()) {
172         WriteLogError("Erro de conexao (GetAzAlt)");
173         return false;
174     }
175
176     sky6RASCOMTele.GetAzAlt();
177     var az = sky6RASCOMTele.dAz;
178     var alt = sky6RASCOMTele.dAlt;
179
180     return {"az": az, "alt": alt};
181 }
182
183 /**
184  * Pega a data e o hor rio do momento que a fun   o    chamada.
185  *
```

```
186  * @returns {object} Um objeto com os dados.
187  */
188 function getTimeNow()
189 {
190     var time = new Date();
191     var day = time.getDate();
192     var month = time.getMonth();
193     var year = time.getFullYear();
194     var hour = time.getHours();
195     var minutes = time.getMinutes();
196     var seconds = time.getSeconds();
197 
198     return {
199         "day": day,
200         "month": month,
201         "year": year,
202         "hour": hour,
203         "minutes": minutes,
204         "seconds": seconds
205     };
206 }
207 
208 /**
209  * Cria o nome do arquivo para o dia atual.
210  *
211  * @returns {string} O nome do arquivo do dia atual.
212  */
213 function setFileName()
214 {
215     var time = getTimeNow();
216 
217     if (time.day < 10) {
218         var day = "0" + String(time.day);
219     } else {
220         var day = String(time.day);
221     }
222 
223     if (time.month < 10) {
224         var month = "0" + String(time.month);
225     } else {
226         var month = String(time.month);
227     }
228 
229     var year = String(time.year);
230 
231     var filename = year + month + day;
232 
233     return filename;
234 }
235 
236 /**
237  * Pega o hor rio atual do computador formatado.
238  *
239  * @returns {string} O hor rio no formato %H:%M:%S.
240  */
241 function getFormattedTime()
242 {
```

```
243     var time = getTimeNow();
244     var formattedTime = String(time.hour) + ":" + String(time.
            minutes) + ":" + String(time.seconds);
245     return formattedTime;
246 }
247 
248 /**
249  * Escreve no log uma mensagem, junto com o hor rio do momento.
250  * Tamb m define o tipo(n vel) de mensagem(Info, Warning ou Error
        ).
251  * Formato da mensagem: [LEVEL - 00:00:00] Text
252  *
253  * @param {string} text -  A mensagem a ser escrita.
254  * @param {string} level - N vel da mensage.
255  */
256 function WriteLog(level, text)
257 {
258     var filename = setFileName();
259     try {
260         TextFile.openForAppend(filename);
261         var formattedTime = getFormattedTime();
262         var header = "[" + level + " - " + formattedTime  + "] ";
263         TextFile.write(header + text + "\n");
264         print(header + text);
265         TextFile.close();
266     } catch (texterr) {
267         PrintAndOut("Erro ao editar o log. " + texterr.message + "
                (WriteLog)");
268     }
269 }
270 
271 /**
272  * Escreve a mensage de warning no log.
273  *
274  * @param {string} text -  A mensagem a ser escrita.
275  */
276 function WriteLogWarning(text)
277 {
278     WriteLog("WARNING", text);
279 }
280 
281 /**
282  * Escreve a mensagem de erro no log.
283  *
284  * @param {string} text - A mensagem a ser escrita.
285  */
286 function WriteLogError(text)
287 {
288     WriteLog("ERROR", text);
289 }
290 
291 /**
292  * Escreve a mensagem de informa  o no log.
293  *
294  * @param {string} text - A mensagem a ser escrita.
295  */
296 function WriteLogInfo(text)
```

```
297 {
298     WriteLog("INFO", text);
299 }
300
301 /**
302  * Escreve no debugger e na janela Run Java Script.
303  * Deve ser usado somente quando o log estiver inacess vel.
304  *
305  * @param {string} text - O conte do a ser escrito.
306  */
307 function PrintAndOut(text)
308 {
309     var level = "WARNING";
310     var formattedTime = getFormattedTime();
311     var header = "[" + level + " - " + formattedTime  + "] ";
312     print(header + text);
313     RunJavaScriptOutput.writeLine(header + text);
314 }
315
316 /**
317  * Conecta o telesc pio e cria o arquivo de log do dia.
318  */
319 function Connect_c()
320 {
321     var time = getTimeNow();
322     var formattedTime = getFormattedTime();
323
324     ConnectTelescope();
325
326     var filename = setFileName();
327     TextFile.createNew(filename);
328     TextFile.write(String(time.day) + "/" + String(time.month) + "/
        " + String(time.year) + "\n");
329     var header = "[INFO - " + formattedTime + "] ";
330     TextFile.write(header + "Conectado\n");
331     print(header + "Conectado\n");
332     TextFile.close();
333 }
334
335 /**
336  * Processo de inicializa  o.
337  */
338 function Initialize_c()
339 {
340     if (sky6RASCOMTele.IsParked != 0) {
341         sky6RASCOMTele.Unpark();
342     }
343     sky6RASCOMTele.FindHome();
344     var props = GetRADec("Sun");
345
346     WriteLogInfo("Iniciou o slew (Initialize_c)");
347     SlewTelescopeToRaDec(props.ra, props.dec, "Sun");
348
349     WriteLogInfo("Iniciou o rastreamento (Initialize_c)");
350 }
351
352 /**
```

```
353  * Processo do Flip.
354  */
355 function Flip_c()
356 {
357     var props = GetRADec("Sun");
358     WriteLogInfo("Iniciou o slew (Flip_c)");
359     SlewTelescopeToRaDec(props.ra, props.dec, "Sun");
360 
361     WriteLogInfo("Completou o flip (Flip_c)");
362 }
363 
364 /**
365  * Desliga o tracking e faz o parking.
366  */
367 function TurnOff_c()
368 {
369     SetTelescopeTracking(0, 1, 0, 0);
370     WriteLogInfo("Desligou o rastreamento (TurnOff_c)");
371 
372     ParkTelescope();
373     WriteLogInfo("Desconectado (TurnOff_c)");
374 }
375 
376 /**
377  * Reconecta o telesc pio e reinicia o tracking.
378  */
379 function Reconnect_c()
380 {
381     WriteLogInfo("(Re)conectado (Reconnect_c)");
382     ConnectTelescope();
383     SetTelescopeTracking(0, 1, 0, 0);
384     sky6RASCOMTele.FindHome();
385     RestartTracking_c();
386 }
387 
388 /**
389  * Reinicia o rastreamento.
390  */
391 function RestartTracking_c()
392 {
393     var props = GetRADec("Sun");
394 
395     WriteLogInfo("Iniciou o slew (RestartTracking_c)");
396     SlewTelescopeToRaDec(props.ra, props.dec, "Sun");
397 
398     WriteLogInfo("Reiniciou o rastreamento (RestartTracking_c)");
399 }
400 
401 /**
402  * Aponta para o c u baseando-se no azimute.
403  */
404 function CalibrateTelescope_c()
405 {
406     WriteLogInfo("Calibracao iniciada (CalibrateTelescope_c)")
407 
408     var delta = 20;
409     var props = GetAzAlt();
```

```
410     WriteLogInfo("Azimute atual: " + props.az + " | Altitude atual:
            " + props.alt);
411     var newAz = props.az + delta;
412     WriteLogInfo("Azimute futuro: " + newAz + " | Altitude futura:
            " + props.alt);
413
414     SlewTelescopeToAzAlt(newAz, props.alt, "");
415 }
416
417 /*
418  * Configura os hor rio para inicializar, fazer o flip e desligar.
419  */
420 var work_time = {
421     start_hour: 11,
422     start_minutes: 00,
423     start_seconds: 30,
424     flip_hour: 16,
425     flip_minutes: 00,
426     turn_off_hour: 20,
427     turn_off_minutes: 00,
428     first_calibration_hour: 15,
429     first_calibration_minutes: 00,
430     first_calibration_seconds: 00,
431     second_calibration_hour: 17,
432     second_calibration_minutes: 00,
433     second_calibration_seconds: 00,
434     finish_first_calibration_hour: 15,
435     finish_first_calibration_minutes: 00,
436     finish_first_calibration_seconds: 30,
437     finish_second_calibration_hour: 17,
438     finish_second_calibration_minutes: 00,
439     finish_second_calibration_seconds: 30,
440 };
441
442 /**
443  * Verifica se    a hora da primeira calibra  o.
444  *
445  * @param {object} time - Hor rio atual.
446  * @returns {boolean}
447  */
448 function timeToFirstCalibration(time)
449 {
450     return time.hour == work_time.first_calibration_hour &&
451                 time.minutes == work_time.first_calibration_minutes
                        &&
452                 time.seconds == work_time.first_calibration_seconds
                        ;
453 }
454
455 /**
456  * Verifica se    a hora de voltar para o sol.
457  *
458  * @param {object} time - Hor rio atual.
459  * @return {boolean}
460  */
461 function timeToFinishFirstCalibration(time)
462 {
```

```
463     return time.hour == work_time.finish_first_calibration_hour &&
464                 time.minutes == work_time.
                        finish_first_calibration_minutes &&
465                 time.seconds == work_time.
                        finish_first_calibration_seconds;
466 }
467
468 /**
469  * Verifica se    a hora da segunda calibra  o.
470  *
471  * @param {object} time - Hor rio atual.
472  * @returns {boolean}
473  */
474 function timeToSecondCalibration(time)
475 {
476     return time.hour == work_time.second_calibration_hour &&
477                 time.minutes == work_time.
                        second_calibration_minutes &&
478                 time.seconds == work_time.
                        second_calibration_seconds;
479 }
480
481 /**
482  * Verifica se    a hora de voltar para o sol.
483  *
484  * @param {object} time - Hor rio atual.
485  * @return {boolean}
486  */
487 function timeToFinishSecondCalibration(time)
488 {
489     return time.hour == work_time.finish_second_calibration_hour &&
490                 time.minutes == work_time.
                        finish_second_calibration_minutes &&
491                 time.seconds == work_time.
                        finish_second_calibration_seconds;
492 }
493
494 /**
495  * Verifica se    a hora de inicializar.
496  *
497  * @param {object} time - Hor rio atual.
498  * @returns {boolean}
499  */
500 function timeToInitialize(time)
501 {
502     return time.hour == work_time.start_hour &&
503                 time.minutes == work_time.start_minutes &&
504                 time.seconds == work_time.start_seconds;
505 }
506
507 /**
508  * Verifica se    a hora do flip.
509  *
510  * @param {object} time - Hor rio atual.
511  * @returns {boolean}
512  */
513 function timeToFlip(time)
```

```
514 {
515     return time.hour == work_time.flip_hour &&
516                 time.minutes == work_time.flip_minutes;
517 }
518
519 /**
520  * Verifica se   (ou j   passou) (d)a hora de desligar o tracking e
        se ele
521  * est   ocorrendo.
522  *
523  * @param {object} time - Hor rio atual.
524  * @returns {boolean}
525  */
526 function timeToTurnOff(time)
527 {
528     return time.hour >= work_time.turn_off_hour &&
529                 sky6RASCOMTele.IsTracking != 0;
530 }
531
532 /**
533  * Verifica se    a hora de iniciar a conex o.
534  *
535  * @param {object} time - Hor rio atual.
536  * @returns {boolean}
537  */
538 function timeToConnect(time)
539 {
540     return time.hour == work_time.start_hour &&
541                 time.minutes == work_time.start_minutes;
542 }
543
544 /**
545  * Verifica se o telesc pio est  no hor rio de opera  o.
546  * Procura prever um eventual problema de simples desconex o do
       SkyX.
547  *
548  * @param {object} time - Hor rio atual.
549  * @returns {boolean}
550  */
551 function inOperatingTime(time)
552 {
553     return time.hour >= work_time.start_hour &&
554                 time.hour < work_time.turn_off_hour;
555 }
556
557 while (true)
558 {
559     var time = getTimeNow();
560
561     if (Sky6IsConnected()) {
562         if (timeToInitialize(time)) {
563             Initialize_c();
564         }
565         else if (timeToFirstCalibration(time)) {
566             CalibrateTelescope_c();
567         }
568         else if (timeToFinishFirstCalibration(time)) {
```

```
569             RestartTracking_c();
570         }
571         else if (timeToFlip(time)) {
572             Flip_c();
573         }
574         else if (timeToSecondCalibration(time)) {
575             CalibrateTelescope_c();
576         }
577         else if (timeToFinishSecondCalibration(time)) {
578             RestartTracking_c();
579         }
580         else if (timeToTurnOff(time)) {
581             TurnOff_c();
582         }
583     }
584     else if (timeToConnect(time)) {
585         Connect_c();
586     }
587     else if (inOperatingTime(time)) {
588         Reconnect_c();
589     }
590 }
```

6.2 Versão mais recente

https://github.com/3ldr0n/Automatiza-o-7ghz/