class.eFHT.php

Vorwort:

Dieses Projekt stellt zusätzliche Funktionen der FHT80b Thermostate von ELV in IP-SYMCON zur Verfügung. Wie:

- den Party-/Urlaubsmodus
- das Wochenprogramm
- Komfort-, Absenk- und Fenstertemperatur
- sowie div. Methoden um sie in eigenen Skripten zu verwenden

Es wurde anhand eines IPS – Bricks aufgebaut. Dafür benötigt man auch das eFHZ – Brick und die entsprechende **Core.Main.dll**. Sie sind im Paket enthalten und muss mit der existierenden ausgetauscht werden, falls man die Bricks noch nicht verwendet.

Ich gebe keine Garantie auf die Funktion aller Methoden der eFHT - Klasse. Getestet wurden die Kommandos über mehrere Wochen, jedoch habe ich festgestellt dass es öfters zu Timingproblemen zwischen dem Empfang und der Verarbeitung der Thermostatwerte kommt.
Das betrifft hautsächlich den Empfang des Wochenprogrammes. Durch die Verwendung des Bricks wird das Funkprotokoll leider auch nicht besser.

Ich verwende aber auch nur einen ThinClient - PC mit 500MHz und habe zusätzlich noch etliche Dienste parallel im Betrieb. Deswegen auch der komplizierte Aufbau. In dieser Konfiguration konnte ich immerhin eine gewisse Zuverlässigkeit und Konstanz erreichen. Falls es zu Unklarheiten mit dieser Anleitung kommen sollte, bitte diese zu verzeihen – das Erstellen von Bedienungsanleitungen ist nicht gerade meine Lieblingsbeschäftigung.

Die eFHT Klasse

Installation

Für jedes eingesetzte, zu verwendende FHT muss ein Script mit dem eFHT – Brick erstellt. Dieses Script ist die Grundlage für die Zuweisung der Instanz und die neu eingeführten Statusvariablen.

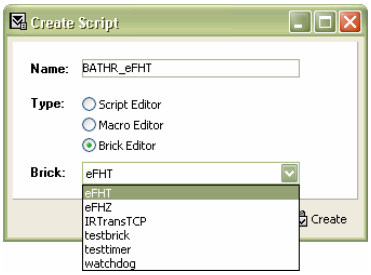

Abb. 5: Erstellung des/der eFHT Skript(e)

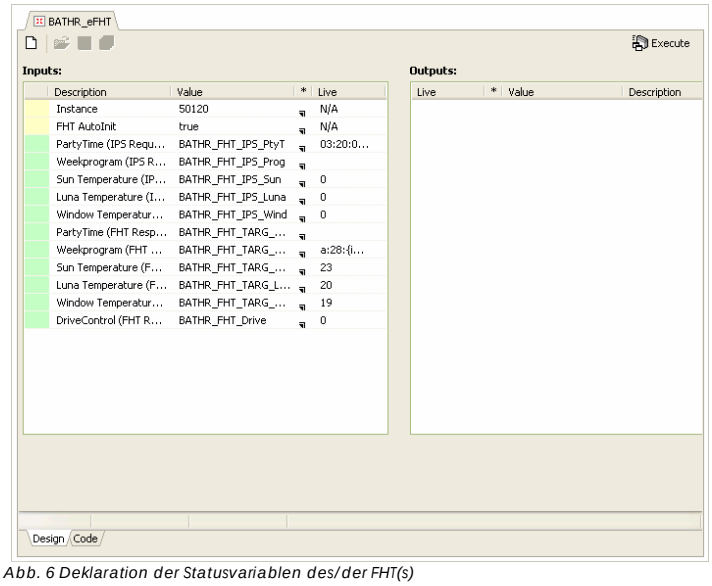

Abb. 6 Deklaration der Statusvariablen des/der FHT(s)

Update

Bei einem Update müssen nur die entsprechenden Ordner im Bricksverzeichnis und die enthaltenen Dateien angepasst werden.
Bei Änderungen an den Brickseinstellungen muss nur das jeweilige Brick im Script Editor aufgerufen und die Einstellungen hinzugefügt oder gerändert werden. Eine Neuerstellung der vorhandenen Bricks ist nicht nötig.

Einbinden

Das Einbinden der Klasse ist ganz einfach mit include, include_once, require oder require_once durchzuführen. Die Klasse wurde selbstdeklarierend programmiert und kann somit sofort verwendet werden.

Syntax

Die Methoden werden verwendet indem man sie mit dem Kürzel „eFHT::“ am Beginn einleitet.
Die Klassenvariablen wurden mit dieser Version entfernt und durch Methoden entfernt. (siehe Variablenmethoden)

z.B: `include('../bricks/eFHT/class.eFHT.php');`

z.B:
```
$erfolg=eFHT::bFHT_SetTemerature(24);
eFHT::bFHT_SetTemerature(24);
```

Beim Einbinden der Klasse wird zusätzlich die class.eFHZ.php eingebunden. Innerhalb der eFHT – Klasse kann nun auch auf die eFHZ – Klasse zugegriffen werden.

z.B: `include('../bricks/eFHT/class.eFHT.php');`

z.B:
```
$erfolg=eFHT::sFHZ_FreeMem();
eFHT:: sFHZ_FreeMem();
```

Das Kürzel „eFHT::“ gilt nun auch für die eFHZ – Klasse.
ACHTUNG: Hier ist das Kürzel „eFHZ::“ ***ungültig***.

Methoden

Allgemeine Methdoden

Begin($IPS-Brickscript)	Setzt die Klasse mit den Statusvariablen für den jeweiligen FHT in Kraft ***$IPS - Brickscript***: Name des Skriptes das erzeugt wurde Pro FHT muss ein Script vorhanden sein.

Puffermethoden

Kennzeichen: ein kleines „b“ am Beginn des Methodennamens
z.B: `eFHT::bFHT_SendTo();`

Diese Methoden speichern die Kommandos in einem Puffer, der anschließend auf einmal zum FHT gesandt werden kann.

bFHT_ClrBuffer()	Löscht den Puffer	
bFHT_SetTemperature($value)	Setzt die Solltemperatur auf ***$value***	Integer & Float
bFHT_SetSunTemperature($value)	Setzt die Komforttemperatur auf ***$value***	Integer & Float
bFHT_SetLunaTemperature($value)	Setzt die Absenktemperatur auf ***$value***	Integer & Float
bFHT_SetWindTemperature($value)	Setzt die Fenstertemperatur auf ***$value***	Integer & Float
bFHT_SetMode ($Mode, $Leap=false)	Setzt den Betriebsmodus. ***$Mode*** = 0 → Automatikmodus ***$Mode*** = 1 → Manueller Modus ***$Mode*** > 1 → UNIX Zeitstempel ***$Leap*** = false → KEINE Anpassung des Schaltjahrproblems (=Voreinstellung) ***$Leap*** = true → Anpassung des Schaltjahres [+8 Jahre] (z.B.: aus 2008 wird 2016) Anhand der Zeit wird der Party- oder Urlaubsmodus eingestellt. Zeitstempel <= akt. Zeit → akt. Zeit + 30 min Zeitstempel > Maximum → = Maximum	$ Mode =Integer $ Leap =Boolean
bFHT_RclActualState()	Setzt eine Aufforderung, dass der FHT alle seine aktuellen Werte neu übertragen soll. (Position, Ist-Temperatur, Fensterzustand, Batteriezustand)	
bFHT_RclTargetState()	Setzt eine Aufforderung, dass der FHT alle seine Sollwerte neu übertragen soll. (Solltemperatur, Komforttemperatur, Absenktemperatur, Fenstertemperatur, Betriebsmodus)	
bFHT_RclWeekProgram($Day=7)	Setzt eine Aufforderung, dass der FHT das Wochenprogramm für den angegebenen Tag schicken soll. ***$Day*** >= 7 → alle Tage ***$Day*** = 0 → Sonntag ***$Day*** = 1 bis 6 → Montag bis Samstag	Integer
bFHT_SendTo($ClearBuffer)	Überträgt den Puffer zum FHT ***$ClearBuffer*** = true (=Voreinstellung) löscht den Befehlspuffer ***$ClearBuffer*** = false → Inhalt des Puffers bleibt erhalten	Boolean

Die Kommandos werden nach der Rangordnung ihrer Register sortiert. Beim Bestätigen des Empfanges durch das FHT wird dann aber nur das erste Register mit ihrem neuen Wert zurückgegeben.
Wie sonst auch üblich wird hier das Zeitfenster (116 Sekunden) eingehalten. Zusätzlich wird mit den Kommandos eine Aufforderung zum Übertragen aller veränderten Werte hinzugefügt. Diese werden aber erst mit dem nächsten Zeitfenster (weitere 116 Sekunden) empfangen.

Singlemethoden

Kennzeichen: ein kleines „s“ am Beginn des Methodennamens
z.B.: eFHT::**s**FHT_Init();

Diese Methoden übertragen die gewählten Kommandos und Parameter sofort beim Aufruf.
Sie beeinflussen aber nicht den Puffer für die Puffermethoden. (Das heißt sie können willkürlich miteinander vermischt werden.)

sFHT_Init()	Initialisiert den FHT neu. Zusätzlich sendet der FHT im Anschluss alle Sollzustände und das gesamte Wochenprogramm.	
sFHT_SetDateAndTime ($Timestamp=0, $Leap=false)	Überträgt den gesamten UNIX Zeitstempel (H:i d.m.Y) an den FHT auf einmal. Das geschieht als ein einziges Kommando, ist keine Schleifenfunktion. ***$Timestamp*** → UNIX Zeitstempel ***$Leap*** = false → KEINE Anpassung des Schaltjahrproblems (=Voreinstellung) ***$Leap*** = true → Anpassung des Schaltjahres [+8 Jahre] (z.B.: aus 2008 wird 2016)	$ Timestamp =Integer $ Leap =Boolean
sFHT_SetWeekProgram ($Day, $Von1=false, $Bis1=false, $Von2=false, $Bis2=false)	Setzen des Wochenprogrammes für den angeführten Tag. (siehe oben) Schaltpaar 1 : ***$Von1*** bis ***$Bis1*** Schaltpaar 2 : ***$Von2*** bis ***$Bis2*** Es muss immer ein ganzes Paar verwendet werden. Ist eine Eingabe false wird das ganze Paar im FHT gelöscht. Übergabe erfolgt als String. z.B.: „08:20“	$Day =Integer $Von1=String $ Bis1=String $Von2=String $ Bis2=String

Der Empfang von Statusmeldungen von den FHT 's ist analog wie bei den Puffermethoden und natürlich abhängig von ihrer Funktion, außer bei der Funktion sFHT_SetDateAndTime($timestamp=0, $Leap=false) hier wird keine Rückmeldung vom FHT gesendet.

Variablenmethoden

Kennzeichen: ein kleines „v“ am Beginn des Methodennamens
z.B.: eFHT::**v**FHT_GetName();

Der Aufruf dieser Methoden liefern interne Werte des Bricks. (= vormals die Klassenvariablen)

vFHT_GetInstance()	Liefert die Instanz des betreffenden FHT	
vFHT_GetHousecode()	Liefert den Hauscode des betreffenden FHT (wie im Konfigurationsdialog)	
vFHT_GetSettings()	Array aller Einstellungen des FHT	
vFHT_GetName()	der vergebene Name für den FHT	
vFHT_GetTargetWPMode($Timestamp)	Liefert den programmierten Temperaturmodus aus dem Wochenprogramm (Request Variable TARGET) ***$Timestamp*** = UNIX Zeitstempel 0 = Fehler 1 = kein Prog. für diesen Tag vorhanden 2 = Absenktemperatur 3 = Komforttemperatur	Integer
vFHT_GetTargetWPTemp($Timestamp)	Liefert die Temperatur anhand des programmierten Temperaturmodus aus dem Wochenprogramm (Request Variable TARGET siehe oben) ***$Timestamp*** = UNIX Zeitstempel 0 = Fehler 1 = akt. Solltemperatur 2 = Absenktemperatur 3 = Komforttemperatur	Integer

vFHT_DecodeWP($Day, $Target=**true**)	Liefert als Ergebnis ein dekodiertes Array eines Wochenprogrammes. ***$Day*** = Wochentag (0=So,1=Mo,2=Di…) ***$Target*** = true (=Voreinstellung) FHT Response – Wochenprogramm Wenn ***$Target*** = false: IPS Request – Wochenprogramm	$Day=Integer $Target=Boolean
vFHT_GetActualPosition()	Prozentwert der Stellantriebsposition	
vFHT_GetActualTemperature()	Isttemperatur	
vFHT_ GetTargetTemperature()	Solltemperatur	
vFHT_ GetTargetSunTemperature()	Komforttemperatureinstellung	
vFHT_ GetTargetLunaTemperature()	Absenktemperatureinstellung	
vFHT_ GetTargetWindTemperature()	Fenstertemperatureinstellung	
vFHT_ GetTargetMode()	Betriebsmodus des FHT	

Sonderfunktionen

Die Sonderfunktionen sind ein Teil der eFHZ – Klasse die mit der Installation der eFHT – Klasse genützt werden.

Stellantrieb: In der Statusvariable „Drive Control (FHT Response)“ werden unterschiedliche Betriebszustände festgehalten.

0 → normale Aktivität der/des Stellantriebe/s
1 → eine Entkalkungsfahrt wurde eingeleitet
2 → der Testmodus (im FHT) wurde gestartet
3 → eine (Neu-) Synchronisation findet gerade statt

FHT AutoInit: Mit der Brick - Konstanten „FHT AutoInit“ kann bei Empfangsausfall (*Einschlafen eines FHT*) eine automatische Neuinitialisierung des FHTs durchgeführt werden.
Ich habe derzeit nur einen konkreten Zustand analysieren können bei dem das Einschlafen auftritt. Sollte ein anderer Fall auftreten, kann entweder die FHT AutoInit Funktion hier nicht helfend eingreifen, oder sie blockiert unter Umständen den restlichen Funkverkehr.

false → FHT AutoInit = deaktiviert
true → FHT AutoInit = aktiviert

Die Konstante muss unbedingt mit einem Wert (true od. false) gesetzt werden, ansonsten kommt es zu einem Ablauffehler.
Ungefähre Dauer bis das FHT wieder vollständig betriebsbereit ist: 3*116 = 348 Sek.

Beispiele

```
include(“../bricks/eFHT/class.eFHT.php“);
eFHT::Begin(“OFFIC_eFHT”);                    //FHT im Büro (Statusvariablen)
eFHT::bFHT_SetMode(mktime(16,50,0,2,18,2008); //Setzt den Partymodus (=Endzeit)
eFHT::bFHT_SetTemperature(23.5);              //Solltemperatur = 23.5 °C
eFHT::bFHT_SendTo();                          //an FHT übertragen & Puffer löschen
echo eFHT::sFHZ_FreeMem();                    //zeige Restkapazität des FHZ-Puffers
                                              //(verwendet das eFHZ Brick)
```

```
include(“../bricks/eFHT/class.eFHT.php“);
eFHT::Begin(“OFFIC_eFHT”);            //FHT im Büro (Statusvariablen)
eFHT::bFHT_SetMode(time()+7200);      //Setzt den Partymodus (Laufzeit=2 h)
eFHT::bFHT_SetTemperature(25);        //Solltemperatur = 25.0 °C
```

class.eFHT.php

```
eFHT::bFHT_SendTo(false);              //an FHT übertragen & Puffer NICHT löschen
eFHT::Begin("KITCH_eFHT");             //FHT in der Küche (Statusvariablen)
eFHT::bFHT_SendTo(true);               //an FHT übertragen & Puffer löschen


include("../bricks/eFHT/class.eFHT.php");
eFHT::Begin("FHT_Kinderz");                    //FHT (Statusvariablen)
eFHT::sFHT_Init();                             //FHT initialisieren


include("../bricks/eFHT/class.eFHT.php");
eFHT::Begin("FHT_Kueche");                        //FHT (Statusvariablen)
eFHT::sFHT_SetWeekProgram(0,"08:50","15:00");     //Wochenprogramm :
                                                  //Sonntag: Komforttemperatur
                                                  //von 8h50-15h & Rest löschen


include("../bricks/eFHT/class.eFHT.php");
eFHT::Begin("FHT_Kueche");                     //FHT (Statusvariablen)
print_r(eFHT::vFHT_DecodeWP(0));               //Wochenprogramm (TARGET) für Sonntag:
                                               //(= eFHT::sFHT_DecodeWP(0,true)
print_r(eFHT::vFHT_DecodeWP(7,false));         //Wochenprogramm (IPS)für alle Tage:
```

Enthaltene Dateien

- class.eFHT.php
- config.xml
- run.brx.php

History

Bugfix →	24. Mai 2008 Bugfix beim Schaltjahrproblem bei bFHT_SetMode beim $LEAP Parameter – Danke an ***bmwm3***
zusätzl. Variablenmethoden →	Abruf von Target und Actual Variablen (IPS und eFHT - Brick)
Anpassung FHT Firmwarefehler →	5. März 2008 (Spawn) Anpassung der UNIX Zeitstempel wegen Problemen in der Firmware veralteter FHTs (=Schaltjahrproblem) – Tipp von ***DHOKAI***
Release: Version 0.2 →	4. März 2008 (Spawn) Trennung vom eFHZ – Brick Codeoptimierung - Tipps von ***tommi*** Grundsätzliche Überarbeitung mit zusätzlichen Funktionen.
Neue Singlefunktion: →	25. Februar 2008 bFHT_DecodeWP hinzugefügt – Tipp von ***wgreipl***
Bugfix: →	24. Februar 2008 Aktualisierung der Statusvariablen (falls nicht vorhanden) Auf nicht vorhandene IPS- und eFHT Statusvariablen und deren Typ wird nun ab sofort geprüft. Bei nicht existierenden eFHT Statusvariablen kann es bei Ausführung (include) trotzdem zu einem Fehler führen, der Grund liegt im grundlegenden Aufbau der IPS - Bricks und nicht bei den eFHT – Bricks. Fehler: bFHT_RclWeekProgram behoben – Tipps von ***wgreipl***
Release: Version 0.1 →	16. Februar 2008 (Spawn)

Und jetzt erst mal gutes Gelingen.

Gruß
Günter.

„Was nicht passt, wird passend gemacht !“