

ALARMS MANAGER PROGRAMMING EVENTS HANDLER

5.0. ALARMS MANAGER – PROGRAMMING – EVENTS HANDLER

Events Handler er en funktion i HSYCO systemet, hvor individuel programmering vil blive tilbudt på en enkelt måde gennem en indbygget Event Handler

INDHOLDSFORTEGNELSE

Indhold

Hvad er Events Handler	2
Introduktion	2
Events programmering	2
Status Browser	3
Keywords	3
Event Keyword	4
Action Keywords	5
Programmering	6
Eksempler	6
Temperatur sensor styrer et relæ	6
Kontakt tænder og slukker for et kamera	7
Sender SMS alarm ved tabt internetforbindelse	8

COPYRIGHT

©2017 Mobbis Mobile Solutions

Redigering af dette dokument er afsluttet 04.05.2017

OM DETTE DOKUMENT

Dette dokument giver brugere af Alarms Manager, en vejledning i at foretage tilføjelser til de leverede standard løsninger i form af forskellige typer scripts der ønskes afviklet af systemets Events Handler

VAREMÆRKER

HSYCO er et en varemærke ejet af virksomheden Home Systems Consulting
Alle andre varemærker der måtte være nævnt i dette materiale respekteres

Alarms Manager er udviklet af Team Mobbis i samarbejde med HSYCO

VERSIONERING

Dette dokument er baseret på HSYCO software version 3.6.x
Dette dokument er baseret på Alarms Manager Projekt version 2.6.x
Dette dokument har version 1.1 DK

Hvad er Events Handler

INTRODUKTION

Alarms Manager er et ekstremt omfattende programkompleks. Næsten enhver form for digital og analog måling vil kunne bearbejdes og præsenteres, relæer og kontakter kan overvåges og styres, kameraer kan vise live streaming og opståede hændelser kan automatisk påbegynde en optagelse af valgfri varighed og interne regler og processer kan afvikles og igen initiere andre handlinger og prosesser. Alle former for input kan logisk kobles sammen med alle former for output og dermed skabe et fuldautomatisk kontrolsystem.

Med Alarms Manager har vi skabt et fleksibelt standard system, der udnytter en lang række af de nævnte funktioner. Men et standard system er og bliver jo som navnet siger; et standard system. Hvis Alarms Manager er blevet præsenteret af én af vores konsulenter, vil det være tydeligt at Alarms Manager er yderst omfattende. Ind imellem opstår der dog ønsker og krav fra kunder, der mangler en funktion der ikke er tænkt ind i standard systemet. I næsten alle tilfælde har vi kunnet tilbyde at udvikle disse funktioner, ved at programmere dem som specialudvikling. Dette vil vi fortsætte med at tilbyde.

Imidlertid er nogle funktioner så simple at programmere, at vi har valgt løbende at samle nogle eksempler i denne "manual".

NB: DETTE ER IKKE EN PROGRAMMERINGSMANUAL. HVIS VIRKSOMHEDEN HAR ADGANG TIL KOMPETENTE UDVIKLERE, STILLER VI GERNE OG GRATIS PROGRAMMERINGSANVISNINGER TIL RÅDIGHED PÅ ENGELSK. SYSTEMET ER RIMELIGT ÅBENT FOR EGEN VIDEREUDVIKLING.

DENNE SKRIVELSE PÅ DANSK ER IMIDLERTID BASERET PÅ AT FREMVISE EKSEMPLER PÅ ALLEREDE GENNEMFØRT PROGRAMMERING AF SMÅ RUTINER OG DET ER UDELUKKENDE DE MEST ENKLE EKSEMPLER VI MEDTAGER HER. DET ER VORES HENSIGT AT AL DEN INFORMATION VI GENGIVER I DENNE SKRIVELSE, SKAL KUNNE GENNEMFØRES AF KUNDER HELT UDEN PROGRAMMERINGSERFARING OG AT ALLE EKSEMPLER SKAL KUNNE IMPLEMENTERES HELT UDEN RISIKO FOR AT ØDELÆGGE NOGET SOM HELST I DE LEVEREDE STANDARD PROGRAMMER.

EVENTS PROGRAMMERING

HSYCO og Alarms Manager er baseret på en hændelsesorienteret arkitektur. En lang række tilsluttede teknologier, kan sammen med et stort antal interne services knyttes sammen og dermed skabe et yderst avanceret systemkompleks. Alene udtrykket kan skræmme selv de mest hardcore teknikere væk, men sagen er at det er lykket for HSYCO at skabe nogle yderst intuitive metoder til at skabe sammenhæng mellem de mange tekniske elementer.

Events programmering er et tilbud, til de der har behov for at foretage yderligere beregninger på data, eller sammenhænge, der ikke er tænkt ind i de leverede standardløsninger.

HSYCO understøtter Java programmering og det står kunden frit for at anvende dette. Men Java kræver kompilering og det kræver genstart af systemet og det er måske ikke altid lige praktisk. Derfor understøtter HSYCO desuden JavaScript og et eget sprog, kaldet EVENTS.

EVENTS programmeringsmiljøet er baseret på et simpelt sprog, der forbinder opståede hændelser (events) med efterfølgende aktiviteter (actions).

EVENTS regler skrives og gemmes i en tekstfil med navnet events.txt i roden på HSYCO Server.

Som sagt skal EVENTS definitioner ikke compileres. Når en programmering er afsluttet og gemt, genindlæses filen automatisk og de nye eller ændrede funktioner træder straks i kraft.

HSYCO Server analyserer konstant om der er regler fra EVENTS Handler der skal udføres.

Alle de viste opgaveløsninger i dette dokument, er baseret på EVENTS.

Status Browser

En HSYCO applikation, uanset om den er udviklet i Java, JavaScript, eller EVENTS, interagerer med eksterne systemer gennem I/O datapunkter. HSYCO benytter desuden variabler til at gemme mellemresultater til eventuelle videre beregninger. Når egne applikationer skal testes, er det yderst brugbart at have adgang til denne information. HSYCO giver adgang til en såkaldt Status Browser, med live opdaterede registre med denne information.

KEYWORDS

Keyword benyttes i EVENTS til at identificere hændelser og give ordre om at udføre aktiviteter. Derfor skelnes mellem hændelses keywords (Events keywords) og aktivitets keywords (Action keywords).

Nedenfor listes alle disse keywords. I dette dokument vil parametre til keywords udelukkende blive beskrevet for de viste programmeringseksempler. Antallet af beskrevne keywords vil derfor stige efterhånden som flere beskrivelser indgår i dokumentet.

Event Keyword

En Event (hændelse) defineres med et keyword der identificerer event typen, typisk efterfulgt af én eller flere parametre. Status på hændelserne benyttes til at foretage én eller flere aktiviteter (Actions). Event keywords der er beskrevet i eksemplerne, er angivet med grøn skrift.

System Events	
DAY	Angiver dagtimerne (baseret på data angivet i HSYCO settings)
HAACTIVE	HSYCO er master eller slave i High Availability setting
HSYCOSTART	Angiver HSYCO opstart
INIT	Angiver reload af EVENTS filen
NIGHT	Er <i>true</i> i nattetimerne (baseret på data angivet i HSYCO settings)
POWER	Angiver niveau af strømforbrugsmåler
PROGRAMTIMER	Systemets Program Timer
SUNAZIMUTH	Ændring af solens azimuth vinkel (data i HSYCO)
SUNELEVATION	Ændring af solens breddegrad
TIME	Aktuel tid
Cameras	
CAMERA	Kamera optager
CAMERACOMMAND	Viser kamera indstillinger (ved PTZ kameraer)
CAMERAVIEW	Live kamera view vises
DMX	
DMX	Viser DMX hændelser
DMXSTART	Status ved Start af hver DMX bus
Infrared Control	
IR	Status på Infrarød kommunikation
I/O Servers	
<i>IO (benyttet i eksempel)</i>	Status ved hver I/O server
IOSTART	Ved positive opstart af en I/O server
Leak Detector	
LEAK	Viser status på lækage sensorer
Network Services	
HTTP	Viser når HSYCO modtager en GET forespørgsel
LOCATION	Tilhørsforhold til et WiFi access Point
PING	Viser PING status
URL	Viser status på en HTTP eller HTTPS forespørgsel
PBX	
PBX	Viser når en forespørgsel kommer fra et PBX system
Squeezebox	
MUSIC	Viser status på en SqueezeBox
Timers and Schedulers	
TIMER	Viser når der er overensstemmelse med en Timer indstilling
User Interface	
PAGE	Viser hvilken side i et HSYCO Project er aktivt
USER	Viser når en bruger aktiverer et element i web interface

Action Keywords

Ved EVENTS programmering kan såkaldte Action Keywords benyttes. Hvert keyword henviser til en Action (aktivitet og alle keywords har ét eller flere tilhørende parametre. Action keywords der er beskrevet i eksemplerne, er angivet med grøn skrift.

System Keywords	
DATESET	Indstil Dato/tid
HAACTIVE	Gør en Master eller Slave aktiv
POWER	Sætter værdi i Watt til strømforbrugsmåler
PROGRAMTIMER	Opretter eller sletter en program timer
WAIT	Vent antal seconder før næste Action
Cameras	
<i>CAMERA (benyttet i eksempel)</i>	<i>Kan midlertidigt afbryde kamera</i>
CAMERAREC	Starter en kamera optagelse
CAMERARECFULL	Starter en optagelse uden "dropped frames"
Data Logger	
DATALOGGER	Styring af datalogger
DMX	
DMX	Styrer værdier i DMX-512 kanaler
Infrared Control	
IR	Sender infrarød signaler
I/O Servers	
<i>IO (benyttet i eksempel)</i>	<i>Skriver værdier i datapunkter</i>
Leak Detector	
LEAK	Beregning af potential vandlækage
Log	
FILELOG	Tilføjer en log linje i enden af en log
LOG	Generer en informationslinje i den daglige log
Mail	
MAIL	Sender en email meddelelse
Network Services	
PING	Sender PING forespørgsel til IP adresse eller URL
URL	Sender GET eller POST request til URL
Public Announcement	
AUDIO	Afspiller en "Text to Speak" meddelelse
Serial Communication Ports	
COMM	Sender en bytes sekvens til en specifik port
Squeezebox	
MUSIC	Styrer SqueezeBox music afspillere
User Interface	
UISESSIONSET	Styrer specifikke dynamiske felter i user interface
UISET	Styrer specifikke dynamiske felter i user interface
USER	Styrer specifikke dynamiske felter i user interface

Programmering

Nedenfor vises nogle eksempler på små funktioner, der ikke som standard er implementeret i Alarms Manager, men som nogle kunde har efterspurgt. Funktionerne har været lette at implementere og har kunne glæde disse kunder. Derfor stille vi dem til rådighed, så andre kunder ligeledes kan gøre brug af dem.

Listen er endnu meget begrænset, men dokumentet vil løbende blive opdateret, når nye små opgaver finder en enkelt løsning, så download dokumentet af og til. Der kan måske findes et lille guldkorn.

EKSEMPLER

Temperatur sensor styrer et relæ

Event keyword: I/O (I/O navnet består af "sensorkontrolenhednavn.sensornavn")

Event	State	Description
IO name	transient	any change of an I/O variable
IO name = value IO name > value IO name < value IO name >= value IO name <= value	stable	the value is equal, greater, greater or equal, less, less or equal to the given value. The names and values of the I/O variables are specifically related to the type of I/O server. Values can be numeric or strings

Action keyword: I/O

Action	Description
IO name = value	sets the data point to a value. Values are specific to the type of I/O server
IO name = FLIP	inverts the output of the I/O data point. If the status of the output is 0 the new status will be 1, if different from 0 the new status will be 0
IO name1 = IO name2	sets the I/O interface output name1 to the current value of the I/O interface name2

Eksempel

Først skrives Event keyword med tilhørende værdi. Der skrives et kolon som separator tegn og herefter skrives Action keyword med tilhørende værdi (1 eller 0):

```
IO enhed1.o1 >= 25.0 : IO enhed2.o3 = 1
```


I nedenstående aktuelle eksempel fra vores eget demosystem, måles temperaturen af analog sensor "a1" på sensor kontrolenhed "ste2". Er temperaturen over eller lig med 25°C sættes relæ output "o3" på sensor kontrolenhed "er84" til 1 (ON). I linje 2 ændres relæet til 0 (OFF) hvis temperaturen falder til under 25°C.

```
#***** AUTO-START RELAY ON HIGH TEMP *****
# Temperatur sensor ste2a1 på sensor kontrolenhed ste2 tænder relæ o3 (ventilator) på sensor kontrolenhed
# HWg-ER84 når temperaturen er større end, eller lig med 25C og slukker igen når temperaturen er mindre end 25,0C
IO ste2.a1 >= 25.0 : IO er84.o3 = 1
IO ste2.a1 < 25.0 : IO er84.o3 = 0
#***** END *****
```

Kontakt tænder og slukker for et kamera

Event keyword: I/O (I/O navnet består af "sensorkontrolenhednavn.sensornavn")

Event	State	Description
IO name	transient	any change of an I/O variable
IO name = value IO name > value IO name < value IO name >= value IO name <= value	stable	the value is equal, greater, greater or equal, less, less or equal to the given value. The names and values of the I/O variables are specifically related to the type of I/O server. Values can be numeric or strings

Action keyword: I/O

Action	Description
CAMERA name = ON	Live view enabled
CAMERA name = OFF	Live view and recording disabled
CAMERA name = RECON	Recording enabled
CAMERA name = RECOFF	Recording disabled

Eksempel

Først skrives Event keyword med tilhørende værdi. Der skrives et kolon som separator tegn og herefter skrives Action keyword med tilhørende værdi:

IO enhed1.o1 = 1 : CAMERA cam1 =ON

I nedenstående aktuelle eksempel fra vores eget demosystem, anvendes en ubenyttet relæ-udgang som kontakt til at tænde og slukke for et kamera. Der er ikke tilsluttet noget til relæet. Vi benytter blot det datapunkt som relæet afgiver til systemet.

Hvis relæ udgang o8 på sensor kontrolenhed "er84" er 1 (ON) tændes kamera c1. I linje 2 slukkes kamera c1 hvis relæet skiftes til 0 (OFF). Relæet vises som en betjeningsknap i Alarms Manager, hvor kameraet i øvrigt også er syning.

```
#***** STOP CAMERA LIVE VIEW AND RECORDING *****
```

```
# Relæudgang o8 på sensor kontrolenhed HWg-ER84 bruges som virtuel kontakt til at tænde og slukke for kamera c1
```

```
IO er84.o8 = 0 : CAMERA c1 = OFF
```

```
IO er84.o8 = 1 : CAMERA c1 = ON
```

```
#***** END *****
```

Sender SMS alarm ved tabt internetforbindelse

Benytter I/O Event keyword ligesom foregående eksempel. Der benyttes et system Event datapunkt der hedder "system.internet.connection" til at læse status og et system Action datapunkt der hedder "gsm.sms".

NB: Alternativt kan Alarms Managers PING funktion benyttes til at pinge en internet server der typisk aldrig er nede.

```
#***** SEND ALERT WHEN INTERNET LOST *****
```

```
Sender SMS besked til administrator ved tabt internetforbindelse
```

```
IO system.internet.connection = offline : IO gsm.sms.12345678 = "Ingen internet forbindelse"
```

```
IO system.internet.connection = online : IO gsm.sms.12345678 = " internet forbindelse OK"
```

```
#***** END *****
```

>>>>>>>> Blank side <<<<<<<<

