

Docs for Funcis-Sharp / kod-test

Funcis-sharp laddas ned ifrån:

<https://github.com/danielwerthen/Funcis-Sharp/archive/master.zip>

kod-test laddas ned ifrån:

<https://github.com/danielwerthen/kod-central/archive/master.zip>

Funcis-sharp

.sln behöver kompileras, resultatet blir /bin/Release mappen innehållande de relevanta exekverbara filerna.

EtsProjektTranslator.exe – Används för översättning av KNX projekt filer (ifrån ETS)

KnxDefaultFalcon.exe – Testverktyg för att bestämma default koppling för falcon samt för att testa om KNX kopplingen fungerar lokalt

KnxNode /ProcStarter – KnxNode är knx-kopplingen som antingen kan exekveras stand-alone, eller via ProcStarter (som startar om KnxNode processen vid oväntade fel)

Kod-test

Projektet är endast en tunn wrapper kring Flux som huseras på

<https://github.com/danielwerthen/flux>

Kod-test har en dependency mot detta bibliotek, vilket kan vara säkrast att lyfta ut för framtida drift.

Driftsättning:

Kompilera Funcis-sharp, och flytta binärerna till den maskin som ska exekvera KNX kopplingen.

Publicera kod-test till en Azure website, företrädesvis med GIT.

Starta KnxDefaultFalcon.exe på KNX maskinen för att bestämma knx kopplingen och testa att den fungerar korrekt.

Starta därefter ProcStarter som håller KnxNode rullande.

KnxNode behöver konfigureras med en url till kod-test servern. Detta kan göras under drift via ett anrop till ProcStarter, ex:

Curl -d'{"url": "http://kod-test.azurewebsites.net"}' http://localhost:3001

Detta görs företrädelsevis med curl som i exemplet men vilket HTTP-POST anrop går bra, så länge den innehåller motsvarande JSON, med utbyten url.

3001 är default porten för ProcStarter och localhost kan bytas mot lämplig IP om man vill skicka anropet ifrån en annan maskin.

Signaler

För att konfigurera vilka signaler som kommer att användas editerar man dessa i signals mappen i kod-test. Det är fritt fram att lägga till fler .is filer så länge man låter signalExtender.is ligga kvar.

Dessa signaler kan använda de funktioner som har definiterat i kod-test och KnxNode, default följer följande med:

Kod-test (Central):

Listen (url) (value, time) =>

Lyssnar efter ett HTTP anrop till "url" där ett "value" i anropets body läses och returneras tillsammans med en tidsstämpel.

Send (url, value)

Tar emot ett "value" och publicerar det via den "url" som angivits.

KnxNode (KNX):

Listen(address) (value, time) =>

Lyssnar efter en angiven gruppadress och returnerar ett "value" tillsammans med en tidsstämpel när ett meddelande dyker in.

Send(address, value)

Skickar iväg ett KNX meddelande med angiven gruppadress och "value".

Syntax i signalfiler

Selector.Function(param, param...)

"\t" (arg, arg...) =>

"\t""\t" Selector.Function (arg, arg...)

Som funktionsparameter kan man definiera konstanta värden enligt JSON syntax. Ex: "string", 123-numeric, { "objekt": "värde" }, ["eller", "arrayer"]

Om en funktion inte returnerar några värden kan man hoppa över callback syntaxen:

Selector.Function(param, param...)

"\t" Selector.Function (arg, arg...)

"\t" motsvarar ett tabb i början av raden och kan inte ersättas av ett soft tab, med mellanslag.

Fler funktionsanrop kan kedjas ihop än två, man kan även ange flera funktionsanrop på samma tabbnivå vilket innebär att de kommer exekveras parallellt.

Kika in Pelare.is i kod-test för en referense.