



CHALMERS

Virtuell driftsättning med 3DEXPERIENCE och ControlBuild

Referensrapport för vägledning och stöd inom
arbete med virtuell driftsättning

Examensarbete inom högskoleingenjörsprogrammet
Maskinteknik

Kevin Huang
Philip Malmqvist

INSTITUTIONEN FÖR INDUSTRI- OCH MATERIALVETENSKAP

CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige 2023
www.chalmers.se

Virtuell Driftsättning med 3DEXPERIENCE och ControlBuild
Referensrapport för vägledning och stöd inom arbete med virtuell driftsättning

Kevin Huang
Philip Malmqvist

© Kevin Huang, 2023
© Philip Malmqvist, 2023

Examinator och handledare: Henrik Kihlman, Institutionen för industri- och materialvetenskap.

Examensarbete för högskoleingenjörsprogrammet Maskinteknik 2023
Institutionen för industri- och materialvetenskap
Chalmers Tekniska Högskola
SE-412 96 Göteborg
Telefon +46 31 772 1000

Abstract

In manufacturing today, there are many tools that can be used in order to create a more efficient production system. One of the tools that is being used is software. With the help of different programs, it is possible to build and simulate a production system. The last step before the simulation can be transferred to the real world is called virtual commissioning. The knowledge about virtual commissioning is however still limited among many companies, which is the reason for carrying out this thesis. This paper works towards creating a fundamental understanding about the basic functions on how to work with virtual commissioning, which will furthermore give the opportunity for further development by students in a later phase. The software programs that are used in this work are called 3DEXPERIENCE and ControlBuild. Using these two programs, there will firstly be a simple connection created between them, to make sure that they can “communicate” with each other. After that is complete, the work with different use-cases begins. In order for this paper to be useful for others, the work surrounding the creation of the use-cases are being documented in detail with the purpose of giving others the chance to replicate our work.

Sammandrag

I dagens industri med inriktning mot produktion, finns det många sätt att förbättra sitt produktionssystem. Ett sätt är att arbeta med programvaror. Med hjälp av olika programvaror, blir det möjligt att bygga och simulera produktionssystem. Det sista steget innan simuleringen överförs till verkligheten kallas för virtuell driftsättning. Kunskapen kring virtuell driftsättning är dock begränsad inom stora delar av industrin, vilket är grunden för utförandet av detta examensarbete. Denna rapport kommer att behandla grundläggande information och kunskap om virtuell driftsättning, för att göra det möjligt att utveckla ytterligare färdigheter inom området. Programmen som används i detta arbete är 3DEXPERIENCE och ControlBuild. Med dessa två program, kommer det först att skapas en koppling mellan dem, för att möjliggöra "kommunikation" sinsemellan. Därefter kommer arbetet fortsätta med att behandla olika use-cases. Syftet med rapporten är att underlätta för framtida studenter som arbetar med virtuell driftsättning och därför är arbetet kring use-cases beskrivna i detalj vilket gör det möjligt för andra att replikera innehållet i rapporten.

Förord

Detta examensarbete gjordes på Chalmers Tekniska Högskola på programmet maskinteknik. Arbetet har en omfattning på 15 högskolepoäng och utfördes under vinterhalvåret 2022/2023.

Under projektets gång har vi fått vägledning av vår handledare och examinator Henrik Kihlman och därmed riktar vi ett stort tack till honom. Vi vill också passa på att tacka Prodtex AB som har bidragit med resurser vilket gjorde det möjligt för oss att utföra examensarbetet.

Kevin Huang & Philip Malmqvist

Akronym

HMI – Human machine interface

PLC – Programmable logic controller

SFC – Sequential function chart

IEC - International Electrotechnical Commission

FMI – Functional Mockup Interface

FMU – Functional Mockup unit

RFQ – Request for quote

UI – User interface

XML – Extensible Markup Language

AGV – Automated guided vehicle

Delmia – Digital Enterprise Lean Manufacturing Interactive Application

Innehållsförteckning

Innehållsförteckning

1.	Introduktion	1
1.1	Bakgrund	1
1.2	Syfte	1
1.3	Beskrivning	2
1.4	Avgränsning	2
2.	Teori	3
2.1	3DEXPERIENCE	3
2.1.1	Delmia	3
2.2	ControlBuild	4
2.3	Relevanta begrepp	4
2.3.1	HMI	4
2.3.2	PLC	5
2.3.3	FMU	5
2.3.4	Virtuell driftsättning	6
2.3.5	RFQ	6
3.	Metod	7
3.1	Kunskapsinsamling	7
3.1.1	Dassault Systemes kurs	7
3.1.2	EDU SPACE	7
3.1.3	Övningar	8
3.2	HelloWorld	8
3.3	Use cases	8
4.	Arbetsgång	9
4.1	Förarbete	9
4.1.1	Grundinformation	9
4.1.2	HelloWorld	10
4.2	AGV Move-In	11
4.2.1	ControlBuild	12
4.2.2	3DEXPERIENCE	15
4.3	Pick and Place	19
4.3.1	Profiler och kopplingar	19
4.3.2	Simulering och test	24

5.	Resultat.....	25
5.1	Förstudie till projektet.....	25
5.2	Resultat av use-cases.....	25
5.3	Resultat för vidareutveckling.....	25
6.	Diskussion och slutsats	26
6.1	Programvaror	26
6.2	Utmaningar.....	26
6.3	Vidareutveckling	27

Referenser

Bilaga

1. Introduktion

I det här kapitlet beskrivs arbetets bakgrund, syfte, beskrivning och avgränsning. Målet är att ge en bra överblick kring innehållet av denna rapport, genom att förklara vad som behandlas och inte behandlas i rapporten.

1.1 Bakgrund

I dagens samhälle är det mycket som kräver effektivisering. I och med att olika verktyg blir mer och mer sofistikerade kommer företag att behöva använda sig av dessa verktyg för att inte hamna efter i utvecklingen. Gör de inte det kommer de småningom att riskera att gå i konkurs, eftersom konkurrenter kommer få ett övertag på marknaden. Från ett hållbarhetsperspektiv är det också viktigt att minska miljöpåverkan, vilket dessa verktyg kan hjälpa till med genom att effektivisera produktionsprocesser. Exempel på ett verktyg, kan vara simuleringsprogram för produktion i fabriker. Förr när simuleringsprogram inte fanns, var produktionsingenjörer tvungna att planera allt och testköra i verkliga fabriken. Detta är en mycket tidskrävande och dyr metod som var nödvändig att genomföras eftersom det inte fanns några möjliga alternativ. Nu finns möjligheten att istället arbeta i en virtuell miljö och bygga upp fabriken där, för att sedan kunna jobba med att skapa en effektiv fabrik som möjligt. En av fördelarna med att arbeta i en virtuell miljö är att det går att genomföra flera olika iterationer av samma steg i produktionen, för att sedan kunna välja den mest lämpliga för ändamålet. Virtuell driftsättning är en viktig del i arbetet med att förbättra produktionssystem och kommer att behandlas i denna rapport. Mer om arbetet följer här nedan.

1.2 Syfte

Denna rapport kommer att utgöra en förstudie, som behandlar hur arbete med virtuell driftsättning fungerar. Syftet med projektet är att göra det möjligt för framtida studenter/personer att utföra fortsatt arbete med att skapa kunskap inom området, eftersom det fortfarande finns mycket vidareutveckling att göra. Rapporten är utformad som ett underlag för framtida projekt som har i uppgift att utveckla kunskapen genom att arbeta med mer komplexa aspekter i förhållande till virtuell driftsättning.

1.3 Beskrivning

Arbetet består av att jobba med två olika program för att kolla på möjligheter kring virtuell driftsättning. Programmen som används är ControlBuild och 3DEXPERIENCE.

3DEXPERIENCE är plattformen som har applikationen Delmia för att illustrera och simulera den virtuella fabriken och produktionssystemet. Det är i detta program som det går att se de olika robotarna genomföra sina uppgifter för att verifiera att allting fungerar enligt plan.

ControlBuild är programmet där bland annat HMI:t och koden för själva kopplingen mellan de två programmen skapas. Det är två olika delmål i jobbet som ska genomföras. Det första handlar om att skapa en grundläggande förståelse för ämnet och de program som ska användas. Sedan ska det arbetas med virtuell driftsättning från en RFQ med olika use-cases.

Utifrån dessa use-cases är det tänkt att skapa förutsättningar för att ytterligare kunna fördjupa sig inom området med den kunskap som kommer från rapporten.

1.4 Avgränsning

Projektets genomförande och resultat kommer inte att kunna överföra virtuella driftsättningen i en fysisk miljö, då både tidsåtgång och komplexitet skulle överskridas. Beskrivningar och förklaringar kommer endast redogöras till funktioner och utrustning som används eller tycks nödvändiga i 3DEXPERIENCE och ControlBuild. Detta på grund av att programmen är fantastiskt invecklade, således kommer det inte vara möjligt att förklara samtliga detaljer och funktioner i programmen. De olika utrustningar som till exempel robotarmen och produkter under bearbetning, kommer inte att programmeras eller skapas från grunden. Dessa utrustningar kommer istället vara tillgängliga för arbetet som färdigt arbetsmaterial, försedda av examinator och handledare Henrik Kihlman.

2. Teori

I det här avsnittet kommer viktiga teoretiska bakgrundsdetaljer att genomgå. Det kommer att presenteras begrepp och kunskap som är nödvändiga för att uppnå resultatet som projektet har i målsättning. Vissa aspekter av det som presenteras är av stor karaktär och kommer därför inte att förklaras allt för detaljerat, utan de kommer förklaras utifrån det som krävs för att uppnå rapportens mål.

2.1 3DEXPERIENCE

3DEXPERIENCE är en molnbaserad plattform som har i sitt syfte att möjliggöra en ny generation av design- och ingenjör-verktyg. Det öppnar upp tillgängligheten för produktutveckling, informationsflöde och öppnar upp samarbete som normalt kräver externa tekniska tillgångar. Grundutförandet av 3DEXPERIENCE är ganska simpel men kraftfull nog för att binda samman minnen och information i plattformen som också kan förvaras och återskapas i olika applikationer inom plattformen. En analogi som Glenn Whyte använder sig av i *What Is The 3DEXPERIENCE? Let's Find Out!* (Whyte, 2021), plattformen är som ett operativsystem. Plattformen verkar i sig inte göra mycket men under ytan koordineras de olika applikationerna och tjänsternas kommunikation, för att utföra uppgifter.

2.1.1 Delmia

Delmia (Digital Enterprise Lean Manufacturing Interactive Application) är en av applikationerna som återfinns på plattformen 3DEXPERIENCE. Applikationen är ämnat till virtuell tillverkning och simulering där fokus ska ligga på ett mer användarvänligt UI (user interface). En stark fördel med att använda Delmia är att kunna simulera driftsättning av en process för att förhindra osäkerheter och begränsningar som kan uppkomma under en tillverkningsprocess. Det går exempelvis att iscensätta utrymmesbegränsningar eller andra komplexa tillverkningsprocesser, där det är svårt att förutse komplikationer. Morey (2017) menar att, utöver simuleringen, är Delmia anpassat för ingenjörer som inte har en omfattande kunskap inom programmering, vilket underlättar arbetsgången med applikationen. Fokus skiftas därmed till det bättre för ändamålet i virtuell tillverkning.

2.2 ControlBuild

Funktioner som finns i Dassault Systemes ControlBuild, används för den som ska designa ett mjukvaru-kontrollsystem för industrier med PLC och DCS baserade kontrollsystem. Utöver designfunktionen kan ControlBuild utföra simulering, test, validering och utveckling av kontrollsystemets applikationer med språk standarden IEC 61131-3 (Dassault Systèmes®, 2023). Programmeringsspråken Sequential Function Charts och Structured Text är några av språken som innefattas av standarden IEC 61131-3 (International Electrotechnical Commission [IEC]. 2003). Programmet har också möjlighet att kommunicera och samverka med Dassault Systemes andra program i plattformen 3DEXPERIENCE. ControlBuild fungerar huvudsakligen som en virtuell programmeringsmjukvara för PLC och skapandet av HMI som ska återspegla en verklig PLC och operatörspanel.

2.3 Relevanta begrepp

När arbete inom området sker, finns det en del viktiga begrepp och information som är nödvändiga att förstå vad de innefattar. Nedan beskrivs det som är mest relevant för detta projekt.

2.3.1 HMI

HMI är en förkortning för *human machine interface* och är en viktig del av en fabrik. Det är med hjälp av denna, operatörer kan kommunicera med maskinerna i en fabrik, via en instrumentpanel. Det är möjligt att både styra och läsa av data från maskinerna.

Informationen ger möjligheter för till exempel att förhindra driftstopp, eftersom det kan finnas notifikationer på HMI:t när något håller på att bli fel (COPADATA, 2022). När det jobbas med HMI i simulering, skapas en exakt modell av det verkliga HMI:t och den kopplas samman med den virtuella produktionslinjen för att se att allt funkar som det ska. Här gäller det att man skapar *inputs* och *outputs* som korrelerar med robotarna/maskinerna. Ett exempel kan vara att en robotrörelse ska utföras när det trycks på en knapp på HMI:t, då ska rörelsen ske och det kan finnas en lampa eller indikator som signalerar för operatören att signalen gått fram och att allt fungerar som planerat. Ett annat exempel kan innebära att det finns en nödstoppsknapp för fabriken robotar. Det ska alltså vara möjligt att stoppa produktionen utifall det skulle ske en olycka.

2.3.2 PLC

Programmable logic controller (PLC) är en slags dator som ofta används inom industrin, för att kontrollera olika funktioner automatiskt i fabriken. För att en PLC ska fungera krävs det att olika *inputs* läses av från sensorer i fabriken, alternativt manuellt inmatade data. Dessa *inputs* kan därefter processeras i PLC:n och utföra kommandon för till exempel en robotrörelse. Ett annat exempel kan vara ett nödstopp-kommando som aktiveras när en sensor läser av att en människa har korsat en gräns där robotar utför arbete, som kan vara skadligt för människor. Det finns tre huvudsaker som utmärker en PLC. Dessa är I/O, kommunikation och HMI. De tre egenskaperna gör att PLC:n kan utföra sin uppgift på korrekt vis (Unitronics, 2022). För att en PLC ska veta hur den ska agera vid olika situationer krävs det att ett PLC program skapas. Det finns ett antal olika programspråk som kan användas, exempelvis “C” eller “sequential function chart” (SFC).

2.3.3 FMU

Functional Mockup Interface (FMI) är ett standardgränssnitt som kan möjliggöra data-utbyten mellan applikationer. I praktiken används FMI som ett verktyg för att samköra simuleringar, men gränssnittet skapar en modul av simuleringen i form av en ZIP fil med tillägget “.fmu”. Modulen, *Functional Mockup Unit* (FMU), som skapas används för att exportera och importera komponenter till andra applikationer. Filen kommer att innehålla bland annat en XML fil, funktioner och ekvationer som används för sammanflätning av simuleringen (Modelica, 2010). Filerna kan göras om och exporteras i form av en FMU fil, vilket kan användas som en sammanlänkning för plattformen 3DEXPERIENCE (Dassault Systemes, 2022).

2.3.4 Virtuellt driftsättning

Virtuellt driftsättning är det sista steget i produktionsplanering, innan man kan starta produktion i en verklig fabrik. Detta sker i en digital tvilling, som gör det möjligt att identifiera problem som eventuellt uppstår. Därmed ges utrymme för att lösa dessa problem innan systemet sätts i drift i den fysiska fabriken (Siemens, 2022). Detta har inte alltid varit självklart, i och med att tekniken inte alltid funnits tillgänglig. Det har gjort att företag är mer intresserade av virtuellt driftsättning, då de kan få en mängd olika information från simuleringen. Den tänkta orsaken för intresset av virtuellt driftsättning är den ekonomiska faktorn, eftersom det handlar om vinstdrivande företag som vill öka marginalerna och bli mer konkurrenskraftiga på marknaden.

2.3.5 RFQ

Request for quote (RFQ) är något som används när till exempel ett företag vill få reda på hur mycket en produkt eller tjänst kostar. Detta sker genom att en förfrågan skickas med önskade produkter eller tjänster till ett företag. Sedan processas innehållet och det förfrågade företaget skickar tillbaka ett kostnadsförslag. En RFQ är inte ett bindande kontrakt eftersom det oftast skickas till flera olika företag. Därav går det att välja det bästa förslaget som erbjuds (Kenton, 2022).

3. Metod

Det här avsnittet presenterar metoden som har använts under projektets gång. Först beskrivs hur kunskapen införskaffats för att utföra arbetet. Därefter kommer det förklaras hur arbetet kring HelloWorld och de olika use-cases har genomförts.

3.1 Kunskapsinsamling

För att genomföra arbetet var det nödvändigt att samla ihop information och kunskap kring ämnet och de olika programvaror som kommer att användas. Det som behövdes var dels att skapa en överblick kring ämnet och dels att lära sig hur programvaran fungerar med hjälp av webbsökningar och dokumentation som finns i 3DEXPERIENCE. Som en introduktion till ämnet deltog det på ett event där det diskuterades om simulering och hur viktigt det är att företag och universitet ska samarbeta med varandra för att bland annat ge studenter bättre förutsättningar att bli mer förberedda på hur arbetet ute i industrin fungerar.

3.1.1 Dassault Systemes kurs

Ett av de första stegen för att få en överblick i 3DEXPERIENCE och ControlBuild var att delta på en kurs som Dassault Systemes erbjöd via Henrik Kihlman. Det var en två dagars kurs, där gick det att se hur olika funktioner används och det var genomgångar på hur det går att applicera dessa på ett smidigt sätt.

3.1.2 EDU SPACE

Ett annat sätt som utnyttjades för att lära sig mer om 3DEXPERIENCE var att ta del av kurser på EDU SPACE. Läroplattformen EDU SPACE är utgiven av Dassault Systemes. På denna plattform kan förinspelade videos ses och det är även möjligt att läsa om hur 3DEXPERIENCE kan användas med dess olika applikationer och funktioner.

3.1.3 Övningar

Efter att kursen och information från EDU SPACE var insamlat, kunde det utföras övningar för att testa kunskapen. För att utföra övningarna och arbetet framöver försedde Prodtex AB licenser till 3DEXPERIENCE och ControlBuild. Övningarna som gjordes var bland annat att efterlikna det som genomfördes på kursen. Ett exempel på det är att skapa en kapacitiv givare som innebär att en sensor placeras på ett sådant sätt att den skickar en signal när något bryter den, vilket gör att produktionen stannar. Det blir som att ett nödstopp aktiveras, för att till exempel stoppa produktionen om någon har gått in i ett område där det inte är tillåtet att vistas.

3.2 HelloWorld

För att börja arbetet med bägge programmen, måste en grundläggande koppling skapas mellan ControlBuild och 3DEXPERIENCE, detta kallas i detta fall för en *HelloWorld*. Här gäller det att de två programmen ska kunna kommunicera med varandra för att kunna utföra de instruktioner som sedan ska skapas. Det är en del steg som måste följas för att lyckas med detta. Ett delat minne måste skapas för att programmen ska kunna använda sig av samma data, vilket åstadkommer att det inte finns fördröjningar som kan skapa problem.

3.3 Use cases

När en *HelloWorld* har skapats börjar arbetet med use-cases. Beskrivning på use-cases är vilka kommandon som ska åstadkommas, till exempel för robotar i en produktionslinje. Det kan vara ett tillvägagångsätt inom industrin fungerar när ett företag vill få en funktion för sin robot. Instruktionerna för use-cases kommer oftast från en så kallad RFQ (request for quote), där ett företag, till exempel ett konsultföretag, arbetar med att ta fram en offert för vad de kan bidra med för att uppfylla deras krav. I arbetet utgår det från olika use-cases, som kommer att byggas upp med programmen 3DEXPERIENCE och ControlBuild.

4. Arbetsgång

I avsnitt 4 beskrivs det mer i detalj hur arbetsgången har varit. De olika stegen som nämnts i metoddelen kommer att brytas ner i mindre delar för att få en större inblick i hur arbetet har genomförts. Det kommer även att finnas en bilaga till rapporten, där det går att följa steg för steg hur det är möjligt att gå tillväga för att återskapa en grundkoppling som presenteras.

4.1 Förarbete

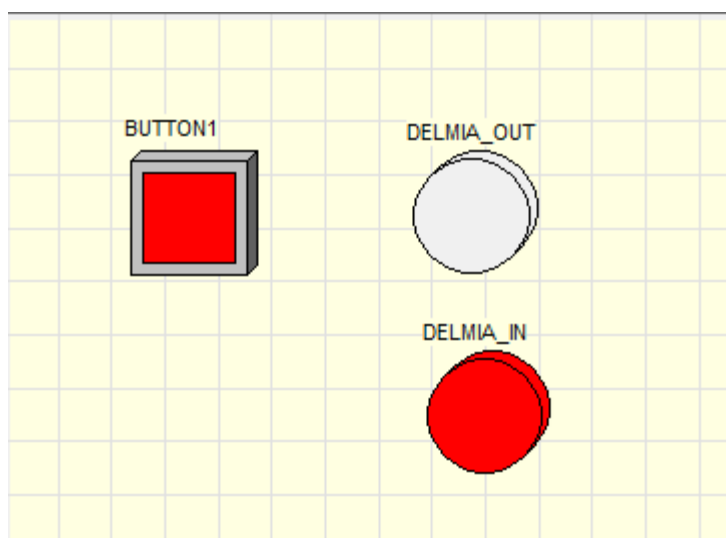
Här arbetades det med att införskaffa sig kunskap för att få en stabil grund att utgå ifrån när arbetet med programvaran sker. Det krävs både teoretisk och praktisk kunskap för att arbeta med 3DEXPERIENCE och ControlBuild.

4.1.1 Grundinformation

I början av projektet jobbades det med att skaffa kunskaper som gör att det går att förstå grundprinciperna av programmen som används i projektet. Här togs det del av ett event där ett antal personer med olika yrken presenterade intressanta aspekter att fundera över när det kommer till ämnet. En annan introduktion till ämnet var att delta på en kurs från Dassault Systemes där det delgavs kunskaper på hur olika funktioner i 3DEXPERIENCE och ControlBuild går att använda. Efter kursen användes även EDU SPACE för att få mer insyn i 3DEXPERIENCE och de olika applikationer som det går att använda sig av för att nå sina önskade mål.

4.1.2 HelloWorld

Det här projektet handlar om att arbeta med olika use-cases med hjälp av ControlBuild och 3DEXPERIENCE. Då måste en grundkoppling skapas. Detta har gjorts med hjälp av Henrik Kihlman. För att möjliggöra det att skapa en liknande grundkoppling har det byggts upp en step-by-step guide som går att hitta som bilaga till rapporten. Denna guide är en version av guiden som Henrik delade med sig av, men en del förändringar har skett vilket ska underlätta för replikering av use-casen i ett senare stadie. Det går alltså att kolla på denna guide för att enkelt kunna återskapa en grundkoppling som är väsentlig under detta arbete, samt att guiden också går att referera till för att smidigare kunna återskapa de use-cases som beskrivs i avsnitt 4.2 och 4.3. I figur 1 visas det HMI som skapats för "HelloWorld".



Figur 1. Skapat HMI från HelloWorld

I figur 2 och 3 visas de *inputs* respektive den *output* som används för denna grundläggande koppling.

Input	Output	Input/Output	Local	Parameter	Physical
Name	Mnemonic Pattern	Type	Valu		
DELMIA_OUT		BOOL	false		
BUTTON1		BOOL	false		

Figur 2. *Inputs* från HelloWorld

Input	Output	Input/Output	Local	Parameter	Physical
Name		Mnemonic Pattern		Type	Value
-DELMIA_IN				BOOL	false

Figur 3. *Outputs* från HelloWorld

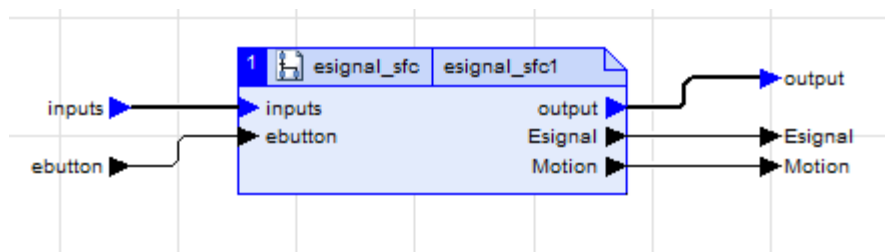
Den grundläggande koppling som skapats, har ett HMI med en knapp och en lampa som gör att man kan se i 3DEXPERIENCE att *in/output* blir *true* respektive *false*. Lampan på HMI:t ska även få en signal tillbaka från 3DEXPERIENCE, vilket gör det möjligt att se att det fungerar bägge hållen.

4.2 AGV Move-In

AGV Move-In är ett use-case där en AGV (automated guided vehicle) ska bära en komponent och flytta den till en arbetsstation för att vidare behandlas. Innehållet av detta use-case är att en operatör ska kunna trycket på en knapp för att få AGV:n att börja förflytta sig till arbetsstationen. Där finns en kapacitiv givare som skapades tidigare. När kapacitiva givaren får en signal att AGV:n är framme, är detta use-case klart. Inbyggt är även en nödstoppsknapp som operatören ska kunna trycka på om det skulle behövas. Då stannar hela processen av för att det ska vara möjligt att åtgärda eventuella problem som uppstått.

4.2.1 ControlBuild

För detta use-case krävs det tre stycken *inputs* och två stycken *outputs*. De tre *inputsen* är Start_button, agv_in_place och ebutton. Start_button är *inputen* som får AGV:n att starta igång och åka framåt. agv_in_place är en signal som kommer från en kapacitiv givare som stannar AGV:n. ebutton kommer från nödstopp knappen, vilket stannar hela processen. Esignal är en output som visar att processen har avstannat på grund av att ebutton har aktiverats. Motion indikerar att AGV:n är i rörelse. I figur 4, 5 och 6 visas de beskrivna *inputs* och *outputs*.



Figur 4. *Inputs* och *Outputs* sammankopplade

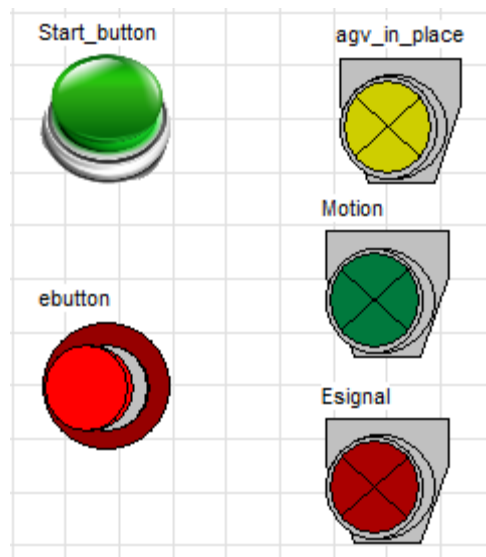
Input	Output	Input/Output	Local	Parameter	Physical
Name	Mnemonic Pattern	Type	Value		
inputs					
Start_button		BOOL	false		
agv_in_place		BOOL	false		
ebutton		BOOL	false		

Figur 5. *Inputs* för AGV Move In

Input	Output	Input/Output	Local	Parameter	Physical
Name	Mnemonic Pattern	Type	Value		
output					
Esignal		BOOL	false		
Motion		BOOL	false		

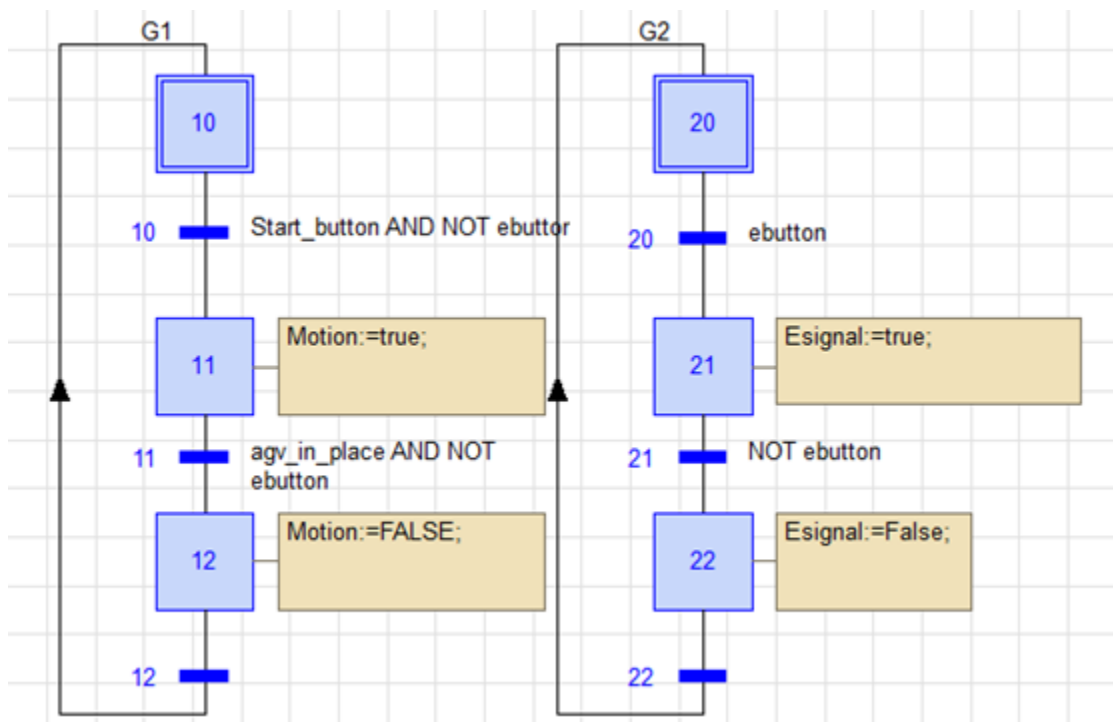
Figur 6. *Outputs* för AGV Move In

Här nedan, i figur 7, visas HMI:t som skapades för detta use-case. Det är två knappar till vänster och tre lampor till höger.



Figur 7. HMI skapad för AGV Move In

I ControlBuild skapas även programmet som styr automationen för AGV:n och detta gjordes genom att göra en SFC (sequential function chart). Nedanför i figur 8, ses utformandet av SFC:n. Det skapades två separata loopar, där den ena loopen (G1), genomför rörelsen för AGV:n och den andra loopen (G2), reagerar på nödstoppsknappen. I en SFC, finns det övergångsvillkor och tillstånd. Det som står mellan de stora blåa kvadraterna är övergångsvillkor och det som står i de gula rektanglarna är tillstånd. För loop G1, börjar programmet att kolla om Start_button är nedtryckt och att inte ebutton är nedtryckt. Om båda dessa stämmer går Motion igång genom att ändras från *false* till *true*. Därefter väntar programmet på att agv_in_place ska bli *true* och att ebutton fortfarande är *false*. Om detta stämmer ändras Motion till *false* för att indikera att sekvensen är genomförd.



Figur 8. SFC för AGV Move In

I ControlBuild skapas det även en FMU fil som sedan överförs till 3DEXPERIENCE för att synkronisera *inputs* och *outputs* för programmen. Nedan kan det ses i figur 9 hur denna FMU fil skapas.

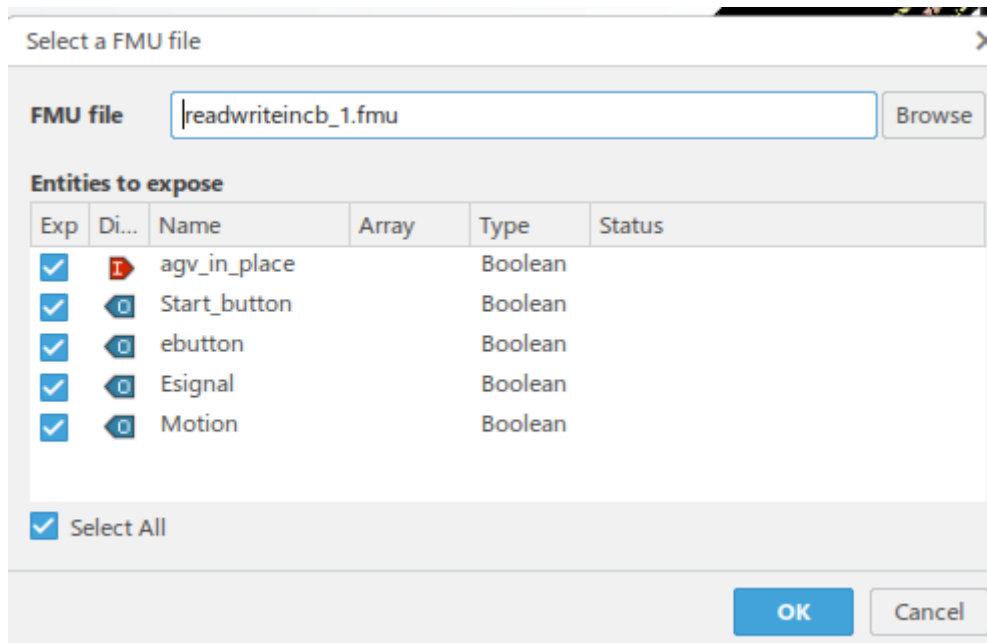
CB name	Read	Write	Fmu Name	Type	Path
Start_button	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Start_button	BOOL	
agv_in_place	<input checked="" type="checkbox"/>	<input type="checkbox"/>	agv_in_place	BOOL	
ebutton	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ebutton	BOOL	
Esignal	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Esignal	BOOL	
Motion	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Motion	BOOL	

XML generation
Fmu Name
Cb Application path

Figur 9. XML generation med *Read* och *Write* definierat.

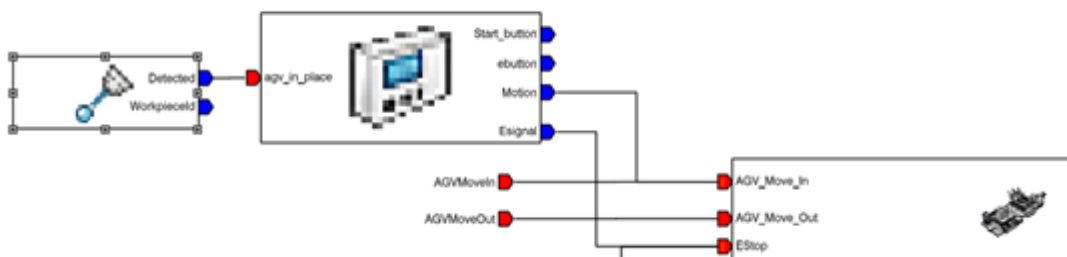
4.2.2 3DEXPERIENCE

Efter att alla sekvenser i ControlBuild skapats, som beskrivs i kapitel 4.2.1, här ovan, börjar arbetet i 3DEXPERIENCE. Det första steget här är att skapa en PLC för att kunna överföra den skapade programkoden från ControlBuild. Efter PLC:n är skapad, väljs den fmu-fil som skapats med tillhörande *inputs* och *outputs*. I figur 10 syns den valda fmu filen med dess *inputs* och *outputs*.



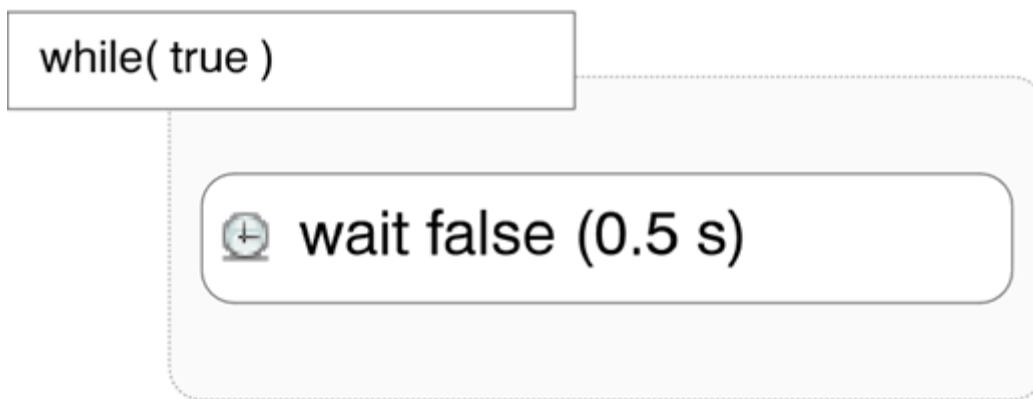
Figur 10. FMU fil importerad till PLC i 3DEXPERIENCE

För att allt ska fungera, måste de olika signalerna kopplas på ett korrekt sätt i 3DEXPERIENCE. Enligt figur 11, går det att se dessa kopplingar som utförts. De blåa markeringarna representerar *outputs* och de röda markeringarna motsvarar *inputs*. I SFC:n som visas i tidigare avsnitt aktiverar Start_button och ebutton, Motion respektive Esignal. Motion aktiverar AGV_Move_in som ändrar värdet på ingången till *true* och förflyttning sker. Nödstoppsignalen Esignal aktiverar Estop i komponent AGV som gör att AGV:n stannar. När AGV:n har förflyttats till sin slutdestination, skickas en signal från kapacitiva givaren *Detected* till agv_in_place, då ändras Motion till *false* och AGV:n stannar.



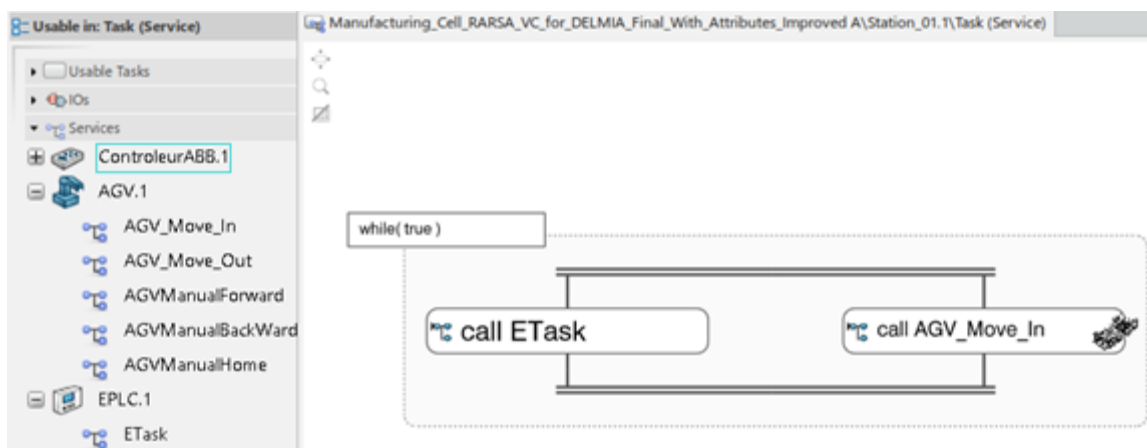
Figur 11. PLC kopplad till AGV och kapacitiv givare i 3DEXPERIENCE

När de olika kopplingarna är klara, behövs det en *Heartbeat*. En *Heartbeat* har funktionen att synkronisera programmen för att kunna simulera use-cases. Här skapas en while loop som sätts till *true*. Inuti denna loop, finns ett *wait* kommando som sätts till *false* med en 0.5s intervall, som visas i figur 12. *Heartbeat* namnges till ETask.



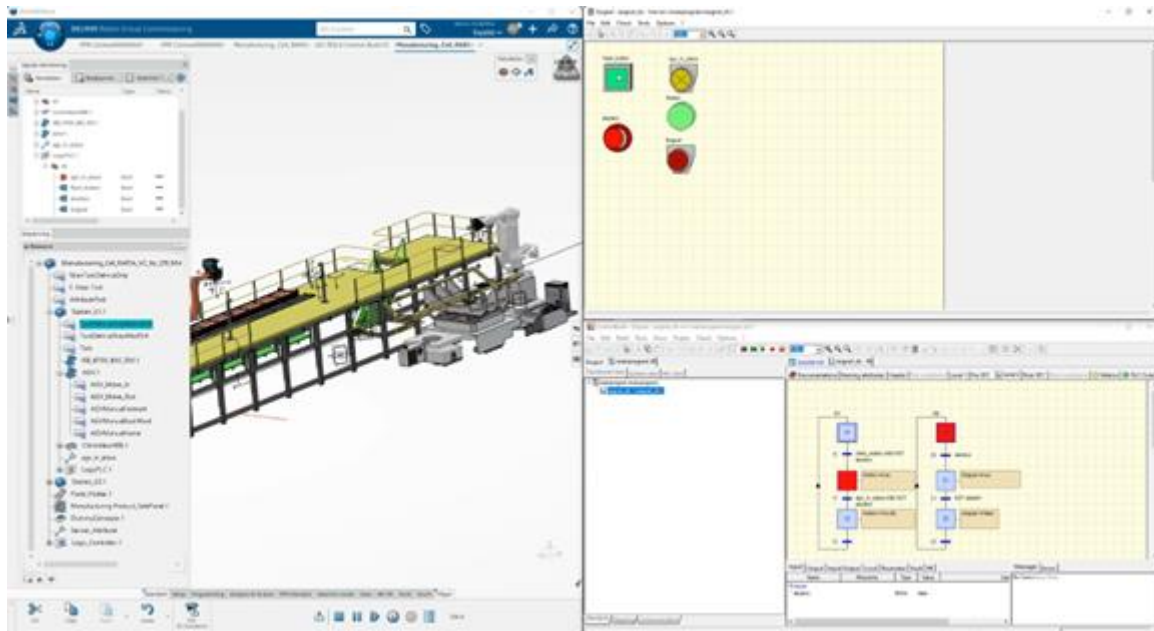
Figur 12. HeartBeat Task i 3DEXPERIENCE

Utöver en *Heartbeat* skapas även en *task* i *Station 1* i programmet. Denna *task* har i uppgift att starta use-case funktionen. Här finns en while-loop som sätts till *true*. I denna loop finns två *tasks* som är parallella med varandra. Den ena kallar på ETask och den andra kallar på AGV_Move_In.



Figur 13. Parallell task i Station 1.

När det ovanstående är genomfört går det att testa sitt use-case och detta sker via en simulation. Här nedan, i figur 14, finns en överblick från simuleringen av AGV Move-in. Det går att se ControlBuild till höger i figur 14 och till vänster syns 3DEXPERIENCE. När startknappen är intryckt, "hoppas" SFC:n till nästa steg och lampan *move* lyser i HMI:t. Eftersom nödknappen inte är intryckt, står G2 sekvensen i SFC:n kvar i sin startposition och inväntar ett eventuellt nödstopp.



Figur 14. Simulering av use-case med 3DEXPERIENCE till vänster och ControlBuild till höger.

4.3 Pick and Place

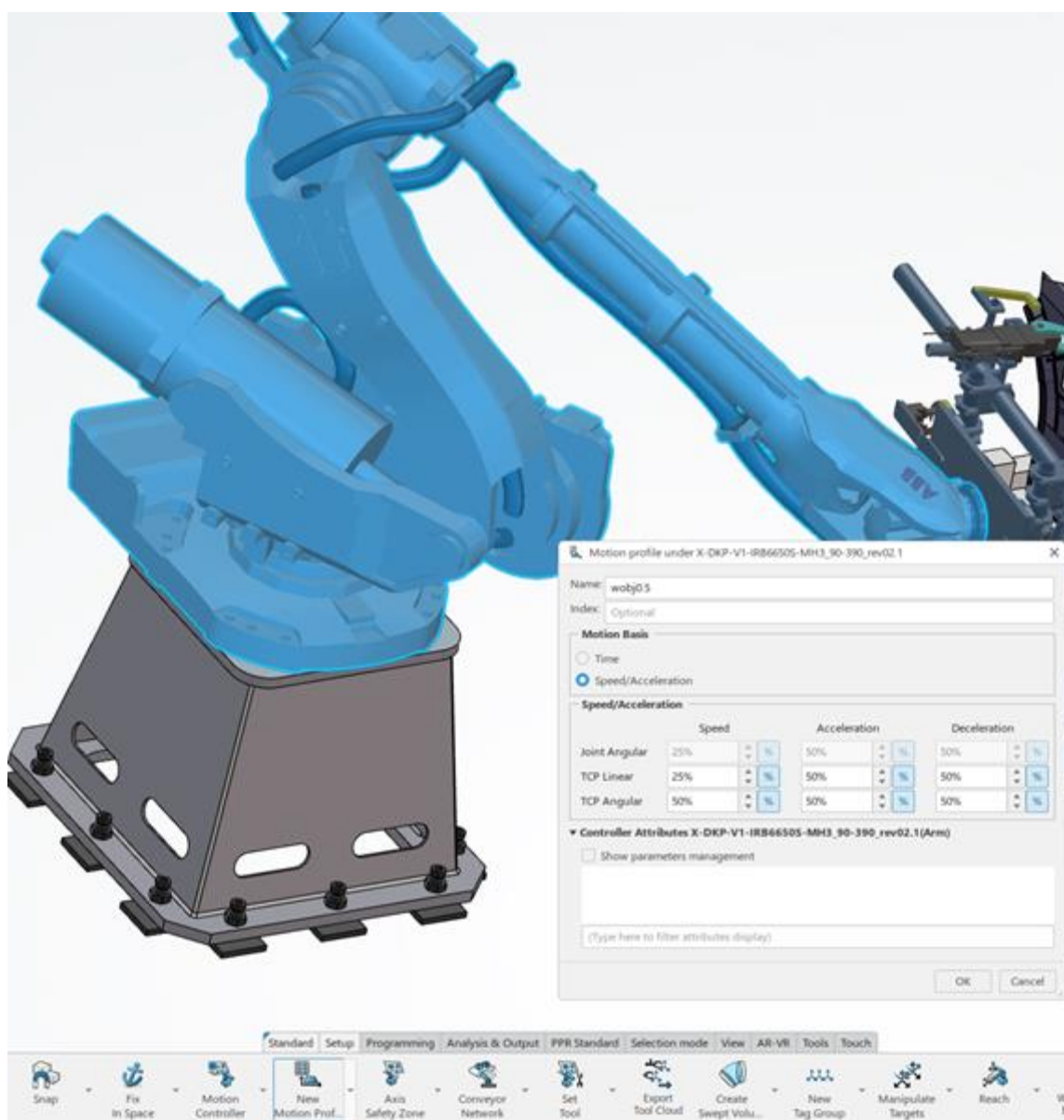
Use-case Pick and Place har i uppgift att utnyttja en robotarmkoppling för att ansluta sig mot ett gripverktyg X-DKP-V1-GRIPPER_Creux_R.1 som är gripdon för ett dörrchassi. Vid lyckad sammankoppling ska roboten förflyttas till en ställning som håller upp dörrchassit och hastigheten bör minska när roboten närmar sig ställningen. Hastighetssänkningen är till för att undvika eventuella risker som kan uppstå vid felplacering av dörrchassit. Om dörrchassit finns på plats ska sensorn som finns på gripdonet kunna detektera det och tar tag i arbetsstycket. Det lyfts sedan bort till en utsedd avlastningsplats, för att sedan möjliggöra att roboten kan starta om processen igen.

4.3.1 Profiler och kopplingar

För att uppnå uppgiften som beskrivs i Pick and Place, behöver det först definieras och läggas till några inställningar i 3DEXPERIENCE. Det som behöver ställas in för att justera precision, säkerhet och slitage kommer att presentera i följande stycken.

4.3.1.1 Hastighetsprofil

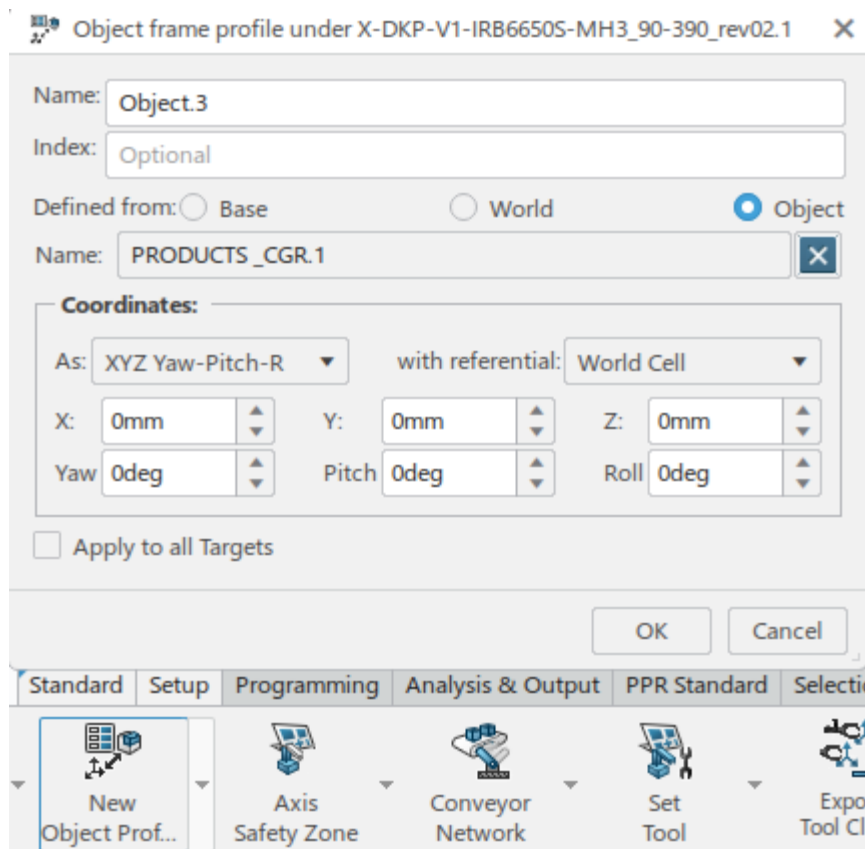
Hastighetsprofil kan skapas enligt figur 15. Inställningen för hastighetsprofilen ställer in hastigheten, accelerationen och inbromsning som roboten kommer att upprätthålla för den bestämda sträckan. Det är en viktig detalj för säkerheten, slitage och minimering av risker. Snabba accelerationer och inbromsningar orsakar större slitage för armen, vilket kan medföra mer underhållningsarbete. Riskerna ska minimeras när roboten närmare sig ett objekt, då ska hastigheten vara lägre för att undvika eventuella skador som kan uppstå när den griper fast objektet. När objektet är fast, är det möjligt att öka hastigheten på robotrörelser igen, för att minska tiden det tar att transportera objektet från ett ställe till ett annat.



Figur 15. Skapad hastighetsprofil.

4.3.1.2 Objektprofil

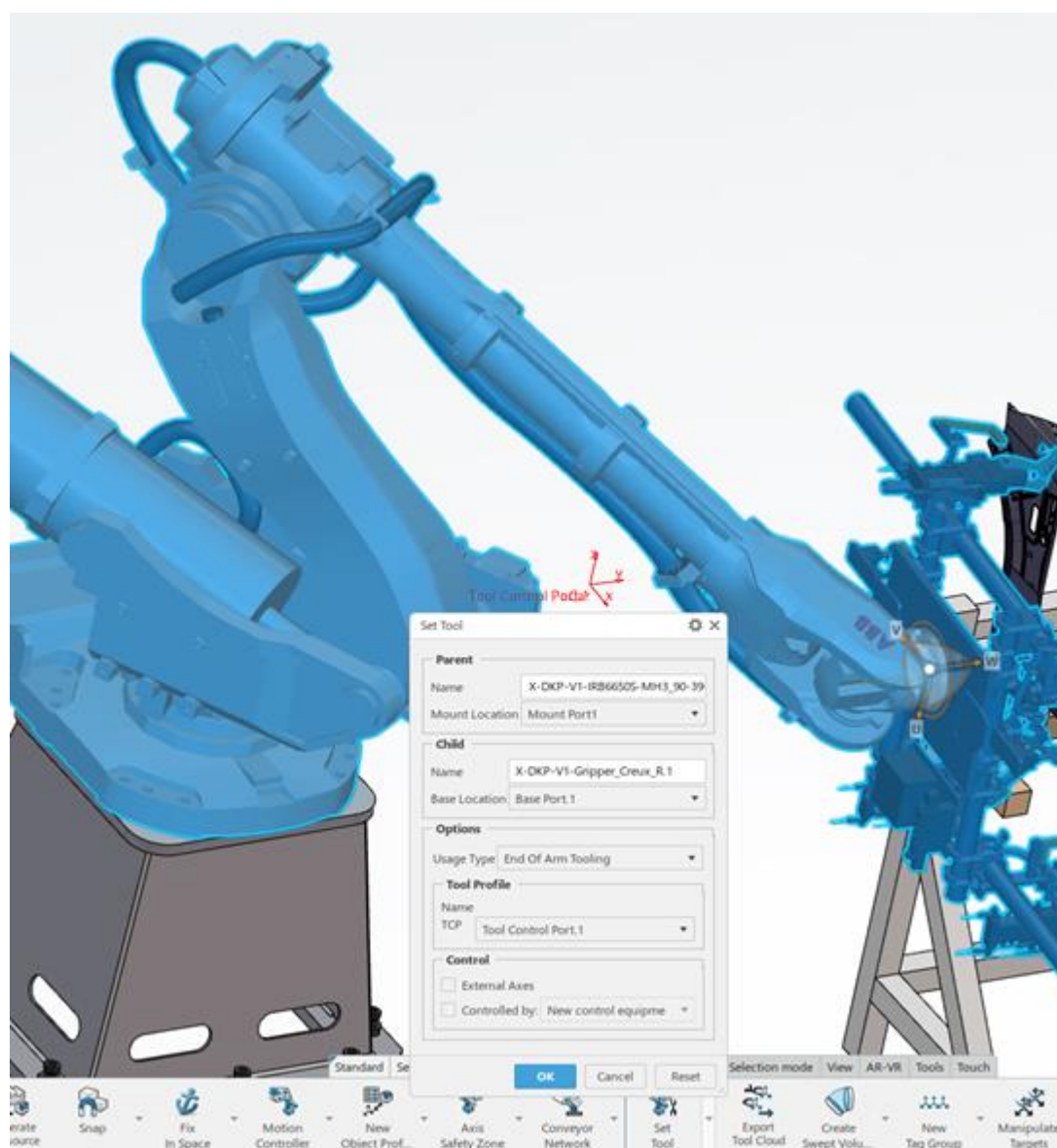
Arbetsstycket behöver definieras av *Object Profile* för att den ska kunna interagera med verktygen. Skapandet av en ny objektprofil används även som en referenspunkt för *tool center point* när robotarmen förflyttas. Det kan till exempel vara att arbetsytan förflyttas, då kan koordinaterna bli inkorrekta för robotarmen när arbetscellen förflyttas, men om objektprofilen finns på plats, kan koordinaterna korrigeras då objektet väljs som den nya referenspunkten (Dassault Systemes, 2022).



Figur 16. Skapad objektprofil

4.3.1.3 Verktyg

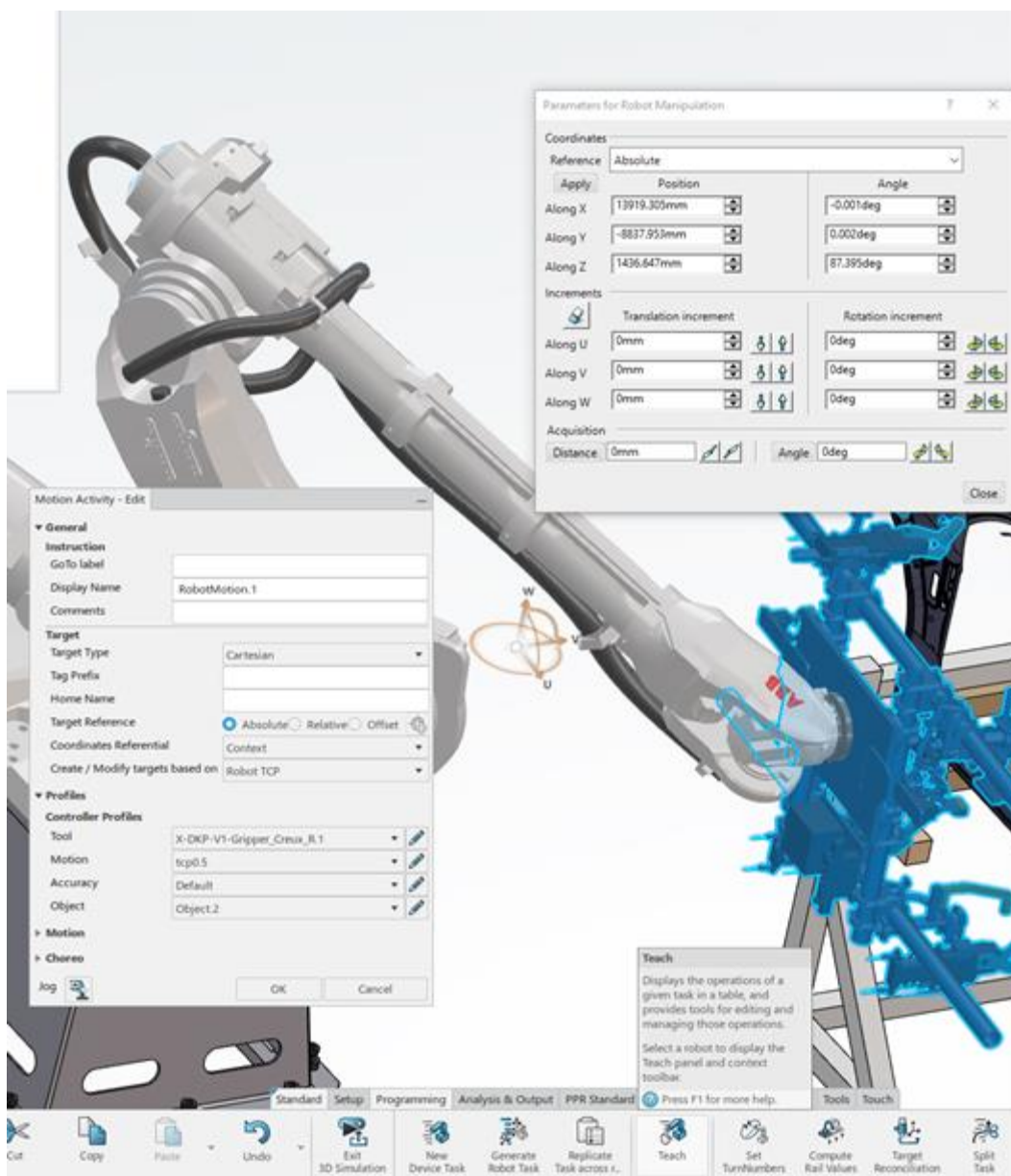
I figur 17 visas hur verktygsprofilen konfigureras. Det väljs en *parent tool* och därefter en *child tool*. *Parent* blir robotarmen med kopplingen som ska sätta ihop verktygsstycket. *Mount port1* är centrum för robotarmen som ska greppa fast verktyget. *Child* är verktyget som ska greppa tag i arbetsstycket, det är ett gripdon för dörrarna. Det väljs även en punkt som gripdonet ska fästas mot robotarmens koppling, *base port* har då valts. Detta görs för att kopplingen mellan robotarmen och objektet ska ske på ett framgångsrikt vis. *Mount location* och *base location* specificerar var kopplingarna är placerade. Det måste även definieras vilken typ av verktyg som används. I detta fall är det *End Of Arm Tooling*.



Figur 17. Skapad *tool profile*.

4.3.1.4 Manipulering

När en robotarm ska manipuleras används *teach*. Funktionen *teach* skapar en *motion activity* med hjälp av hastighetsprofil, objektprofil och verktyg som har skapats. Robotarmen valdes som ska utföra rörelsen enligt *motion activity* och dess profiler. Rörelsemönster valdes till *Cartesian* med referens till absolut-punkten, vilket är en referens för den totala arbetsytan i Delmia och är därmed inte en relativ punkt till robotarmen. Ändring av position som robotarmen utför bör vara bestämt med hänsyn till *tool center point* som i det här fallet är det definierade i *motion activity* fönstret i rutan *tool X-DKP-V1-GRIPPER_Creux_R.1* som visas på figur 18.

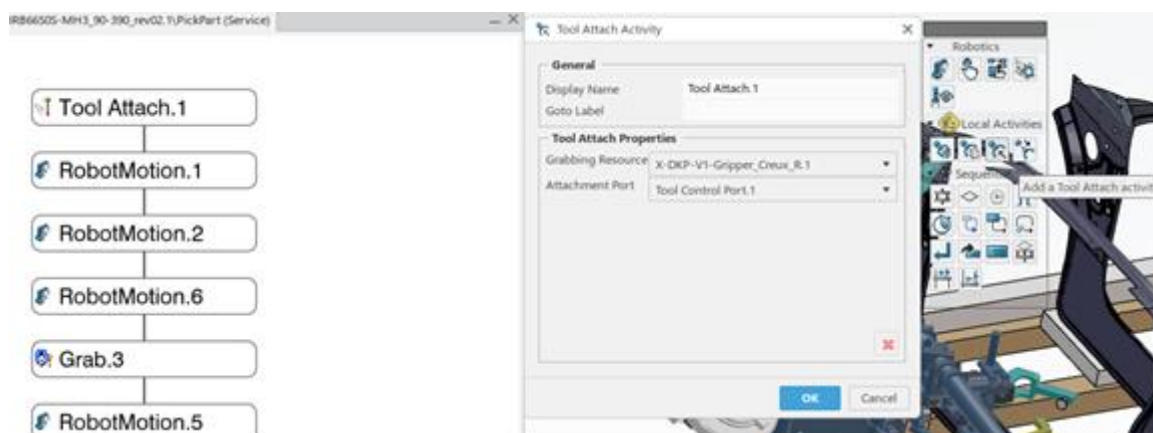


Figur 18. Robotmotion 1 skapad för Pick and Place

Det går även att välja samtliga tidigare skapade profiler som definierar hastighet och objekt som ska bearbetas. Manipulering kan ske på några olika sätt men det som genomfördes är med hjälp av *snap* funktion på *free rotation handle* som ser ut som en *disk* med W, V och U riktningspilarna som visas i figur 18. För att uppnå en högre precision kan *edit* användas, vilket kan uppnå positioner i högre upplösning.

4.3.1.5 Service task

För att kombinera alla de skapade *robotmotions* till en helhet, byggs en *task* upp med dessa inuti. En viktig detalj är att denna *task* ska vara av sorten *service task*. Det beror på att de övriga alternativen inte nödvändigtvis fungerar för ändamålet. De är mer specificerade för andra sorters *tasks*, medan *service task* är mer övergripande. I figur 19 visas hur den *task* som skapades är uppbyggd. Det är först en *Tool Attach*, följt av 3 olika *robotmotions* som är avsedda för sträckan och hastigheten beroende på avståndet till dörrchassit. Därefter finns en *grab* funktion, följt av ytterligare en *robotmotion* som tar dörrchassit vidare till avsedd plats.



Figur 19. Servicetask för Pick and Place

4.3.2 Simulering och test

Till sist genomfördes en simulering, med målet att verifiera att det önskade use-caset är uppnått. Sammanlänkning återskapas på samma metod som från det tidigare nämnda use case *agv move in*. Functional mockup unit är länken mellan ControlBuild och 3DEXPERIENCE. Hur en liknande länk skapas kan tas del av i bilagan. Simulation enligt angivelsen som beskrivits, ska robotarmen förflyttas mot arbetsstycket med en viss hastighet beroende på avståndet som den närmar sig arbetsstycket. Därefter griper munstycket tag i objektet vid korrekt punkt. Arbetsstycket lyfts sedan av robotarmen till önskad slutdestination.

5. Resultat

I det här avsnittet presenteras resultatet av arbetet som genomgått. Det kommer beskrivas vad som har skapats, resultatet från de use-cases som har genomförts, samt beskrivning av vad rapporten bidrar med.

5.1 Förstudie till projektet

Utifrån de förberedande studierna som genomfördes, skapades en bra grund för arbetet med 3DEXPERIENCE och ControlBuild. Genom att delta på kursen som erbjuds av Dassault systemes och möjligheten att använda EDU SPACE för att få en introduktion till 3DEXPERIENCE kunde detta projekt inledas med rätt förutsättningar.

5.2 Resultat av use-cases

Som ses i kapitel 4, går det att konstatera att det har skapats två stycken use-cases som har genomförts. Det första handlar om AGV move in och det andra handlar om *pick and place*

I första use-caset har det på ett framgångsrikt sätt lyckats att få AGV:n att förflytta sig till och stanna på rätt ställe. Detta har gjorts genom att dels skapa ett HMI och SFC i ControlBuild och dels att rätt kopplingar och *tasks* har genomförts i 3DEXPERIENCE.

På liknande vis går det att konstatera att *pick and place* use-caset har applicerats. Här har en *robotmotion task* bildats som gör att det blir möjligt att plocka upp dörrar från en ställning. Sedan förflyttas dessa dörrar till angiven plats som planerat. Ett problem som uppstod var att när gripdonet skulle greppa tag i en enskild dörr, kopplades alla dörrar i ställningen samman, istället för en i taget. Use-caset var utformad för att plocka upp en dörr i taget. Eftersom alla dörrar greppas samtidigt är detta use-case inte ännu helt färdigställt.

5.3 Resultat för vidareutveckling

Syftet med denna kandidatuppsats var att bidra med en rapport som kan användas för att skapa bättre förutsättningar för studenter som är intresserade av att utvecklas inom området virtuell driftsättning. I och med att rapporten beskrivit relevant teori, hur en grundkoppling mellan 3DEXPERIENCE och ControlBuild skapas, samt visat arbetet för ett par olika use-cases, har resultatet av rapport bidragit till det planerade ändamålet.

6. Diskussion och slutsats

Kapitel 6 behandlar diskussionen kring projektet. Har syftet/frågeställningen uppnåtts? Vilka utmaningar har uppstått under arbetet? Vad det finns för möjligheter med vidareutveckling av resultatet.

6.1 Programvaror

Arbetet med 3DEXPERIENCE och ControlBuild har fungerat på ett bra sätt under arbetets gång. Generellt har både inläring och sedan möjligheten att applicera denna kunskap varit genomförbar. Syftet med detta projekt var att skapa en grundförståelse för dessa två program, för att möjliggöra enklare fortsatta studier inom området. Denna rapport har gått igenom både relevant teori samt metod för att underlätta framtida arbeten kring ämnet virtuell driftsättning. Utöver detta har även en utförlig förklaring presenterats kring arbetsgången, där det går att få en större inblick i hur och vad som har gjorts i de olika use-case som arbetet kretsat kring. I och med detta ska det vara möjligt att replikera arbetet som har gjorts och därmed har syftet med rapporten uppfyllts.

6.2 Utmaningar

Under arbetets gång har det uppstått en del utmaningar. Eftersom detta är en förstudie till fortsatt arbete har en stor del av arbetet tillbringats för att lösa olika problem som uppstått. Det har både uppkommit stora och små problem. Den större mängden av problem som uppstått är de små detaljerna som är lättare att göra fel på. Därför är det mycket viktigt att arbeta på ett systematiskt sätt för att undvika att ett steg missas. Det är något som har förbättrats under arbetet, för att minimera risken för misstag. Ett exempel på något som har orsakat problem är de olika kopplingarna som krävs vid PLC:n i 3DEXPERIENCE. Det är lätt att man förväxlar *input* med *output* vid exporteringen till 3DEXPERIENCE, vilket självklart skapar problem vid simulering. När ett sådant misstag uppstår, behöver en *debug* genomföras för att hitta orsaken till att det inte fungerar som det är tänkt och sedan kan misstaget rättas till. Det hade varit intressant att arbeta med ännu fler use-cases i detta projekt, men eftersom det förekommit en del jobb med att fixa olika problem har tiden inte funnits för mer use-cases. Däremot finns det tillräckligt med underlag i denna rapport för att utveckla sina kunskaper inom området, vilket var syftet med projektet.

6.3 Vidareutveckling

Ett nästa steg för vidareutveckling av rapporten kan vara att bland annat skapa mer komplexitet i de olika robot-programmen, kombinera flera olika moment med varandra. Arbete med att skapa ännu mer effektiva processer kan minska miljöpåverkan, genom att till exempel bättre anpassa hastigheten på robotarna för att minska slitage och energiförbrukning. En annan intressant tanke som diskuterats med Henrik Kihlman är att undersöka möjligheten med att göra 3DEXPERIENCE mer adaptiv med att knyta sig till en Siemens *TIA Portal*. *TIA Portalen* är ett simuleringsverktyg från Siemens. Skulle en sådan koppling lyckas, ökar praktikaliteten för 3DEXPERIENCE genom att vidga dess kompatibilitet med andra användbara program.

REFERENSER

Copadata. (2022) Vad är HMI? Hämtad från

<https://www.copadata.com/sv/produkter/zenon-software-platform/visualisering-kontroll/vad-aer-hmi-human-machine-interface-maenniska-maskingraenssnitt-copadata/>

Dassault Systèmes®. (2022) FUNCTIONAL MOCK-UP INTERFACE. Hämtad från <https://www.3ds.com/products-services/catia/products/dymola/fmi/>

Dassault Systèmes®. (2022) Creating Device Profiles. Hämtad från <https://help.3ds.com/2022x/English/DSDoc/DbgUserMap/dbg-t-ObjectProfile-CreateEdit.htm?contextscope=onpremise&id=d8e0d07ffa874a98b240c5b4fc8c720b>

Dassault Systèmes®. Controlbuild. Hämtad från

<https://www.3ds.com/products-services/catia/products/controlbuild/>

International Electro technical Commission. (2003) International Standard IEC (61131-3) Programmable Controllers-Part 3: Programming Languages, 3rd ed.

Kenton, W. (2022) Request for Quote: Definition and How Businesses Use RFQs.

Investopedia. Hämtad från

<https://www.investopedia.com/terms/r/request-for-quote.asp>

Modelica. (2010) Functional Mockup Interface (FMI). hämtad från

https://modelica.org/publications/newsletters/2010-1/index_html#item8

Morey, B. (2017) Building simulation tools for non-programmer types

hämtad från <https://www.sme.org/building-simulation-tools-non-programmer-types>

Siemens. (2022) What is robotics virtual commissioning? hämtad från

<https://www.plm.automation.siemens.com/global/en/our-story/glossary/what-is-robotics-virtual-commissioning/102446>

Unitronics. (2022) What is the definition of “PLC”? hämtad från

<https://www.unitronicsplc.com/what-is-plc-programmable-logic-controller/>

Whyte, G. (2021) WHAT IS THE 3DEXPERIENCE? LET’S FIND OUT!

Hämtad från

<https://hawkridgesys.com/blog/what-is-the-3dexperience-lets-find-out>

Bilaga

The screenshot displays a simulation environment with a watchlist on the left and a 3D model on the right. The watchlist table is as follows:

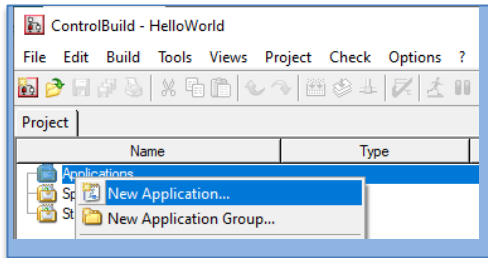
Name	Type	Status	Value
IO			
Start	Bool	—	false
Stop	Bool	—	false
GreenLight	Bool	—	false
RedLight	Bool	—	false

The 3D model on the right shows a control panel with four buttons: 'Start' (green), 'Stop' (red), 'GreenLight' (grey), and 'RedLight' (grey). The 'Start' button is currently lit green, while the others are grey.

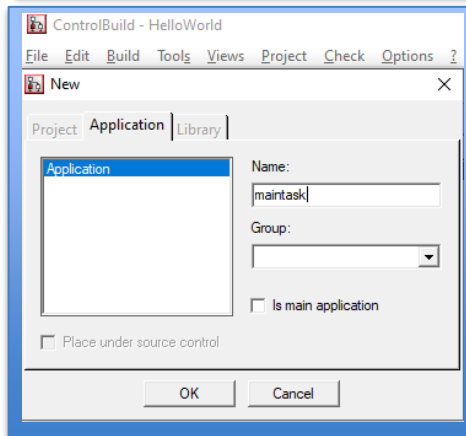
Step by step guide för grundkoppling i 3DEXPERIENCE och ControlBuild

Skapa projektet i ControlBuild

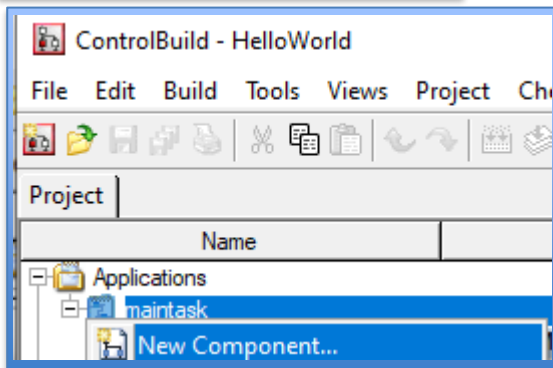
1. Öppna ControlBuild och skapa projekt, namnge det HelloWorld.
2. Högerklicka Applications och tryck på New Application och namnge det till maintask.
3. I maintask, tryck på New Component
4. Namnge Component till sfc_intro och välj Sequential Function Chart som Language.



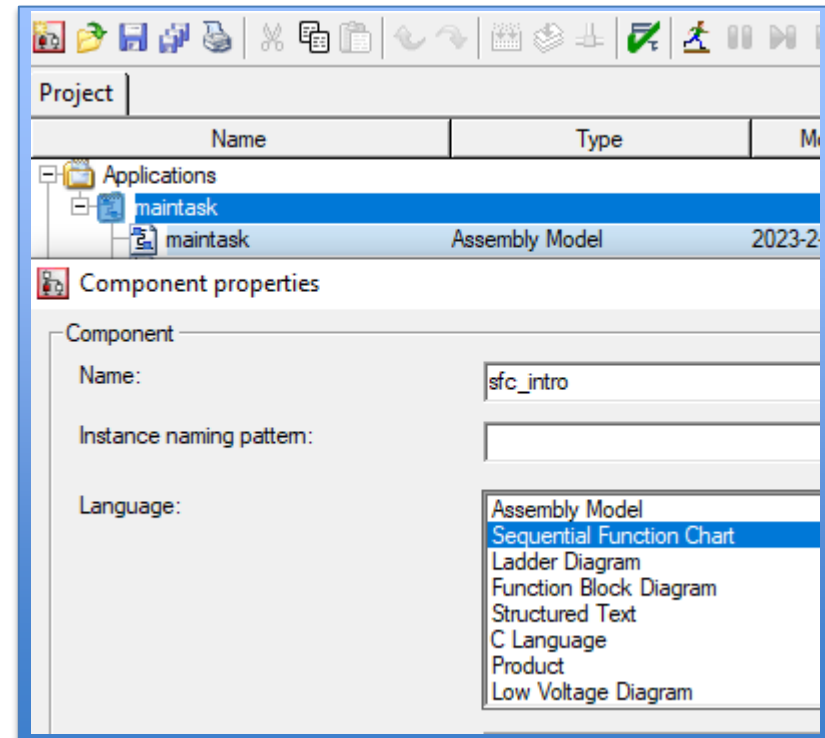
1



2



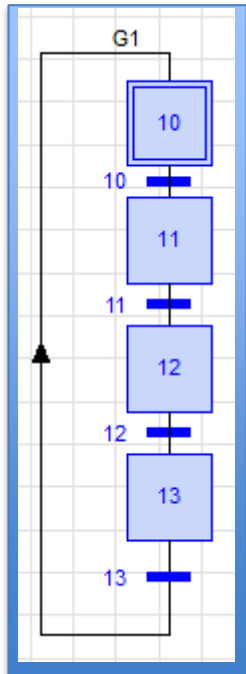
3



4

Skapa Logik

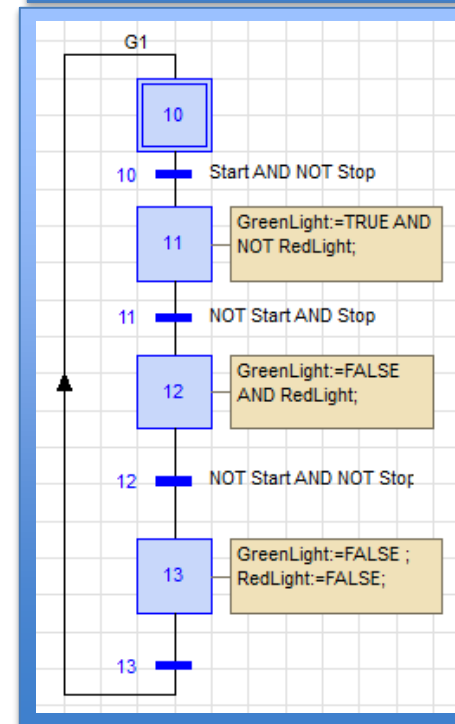
1. Öppna sfc_intro och dra ut lådorna enligt bilden. Den första lådan ska vara Initial step, övriga add steps. Sammankoppla lådorna med Link transition och tryck på number sequential function chart.
2. Lägg först till variable group Inputs och outputs. Markera dem och skapa new variable och alla ska ha type BOOL.
3. Lägg in syntax på SFC som visas enligt bilden.



1

Input	Output	Input/Output	Local	Parameter	Fault	FB
Name	Mnemonic Pattern	Type	Value			
[-] Inputs						
Stop		BOOL	false			
Start		BOOL	false			
[-] Outputs						
GreenLight		BOOL	false			
RedLight		BOOL	false			

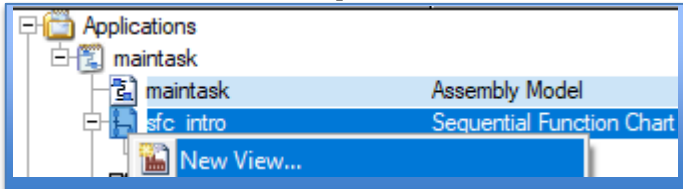
2



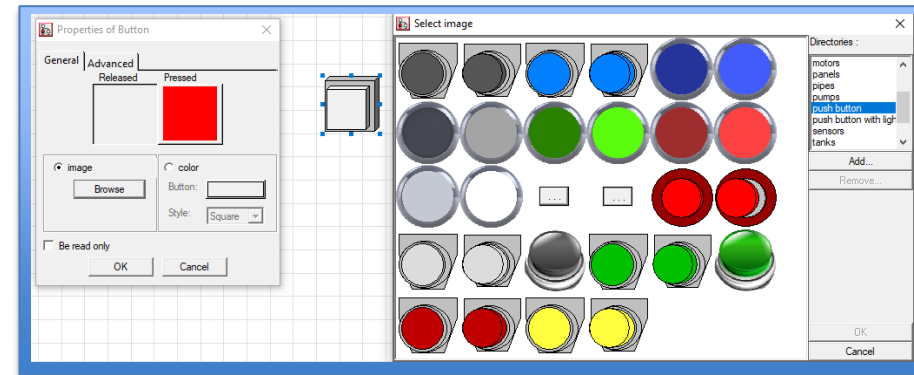
3

Skapa HMI för SFC

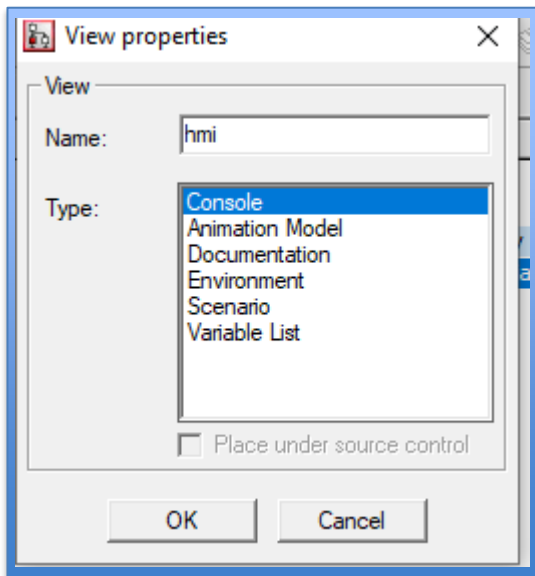
1. Öppna new view på sfc_intro för att skapa ett HMI.
2. Välj console och namnge till hmi.
3. Skapa 2 stycken lights och 2 stycken buttons. Markera released och tryck på browse för att hitta rätt ikon, gör sedan samma sak med pressed.



1



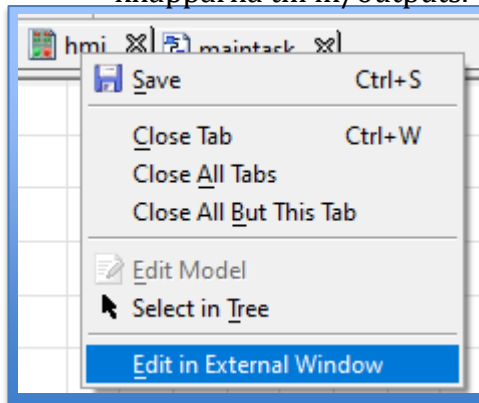
3



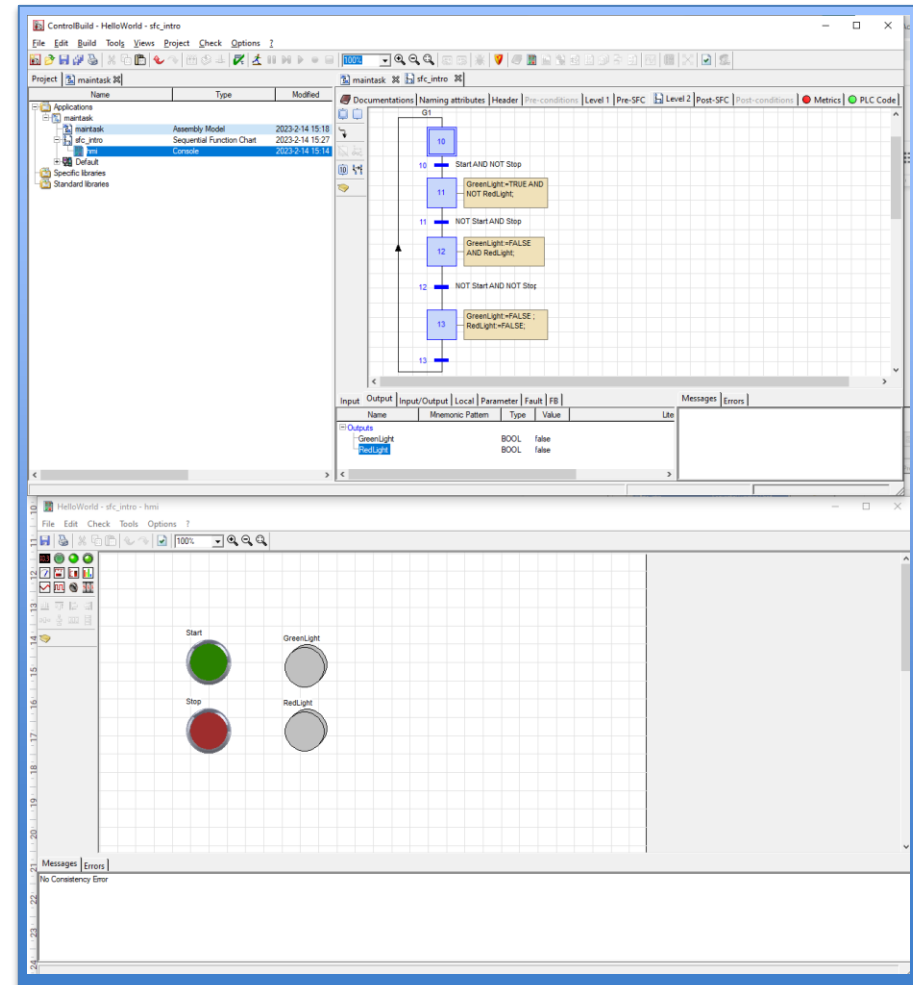
2

4. Dra ut fliken genom att trycka på Edit in External Window.

5. Det är drag and drop när det kommer till att associera knapparna till in/outputs.



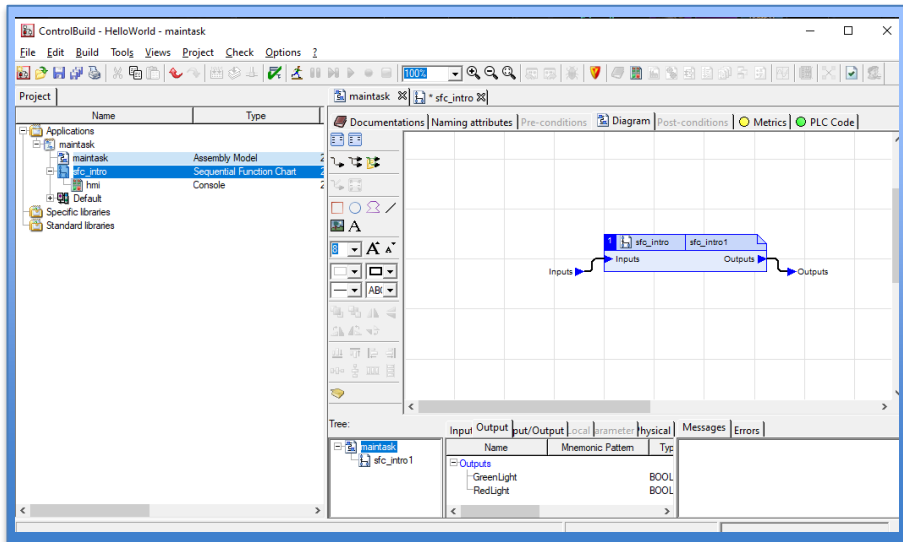
4



5

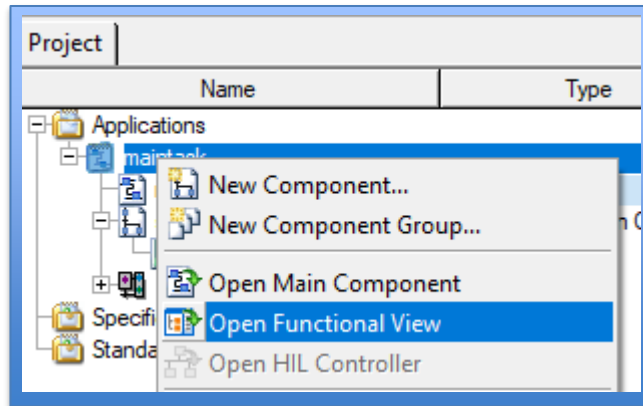
6. Ha upp fliken maintask(assembly model) och dra sedan in sfc_intro i rutan

6

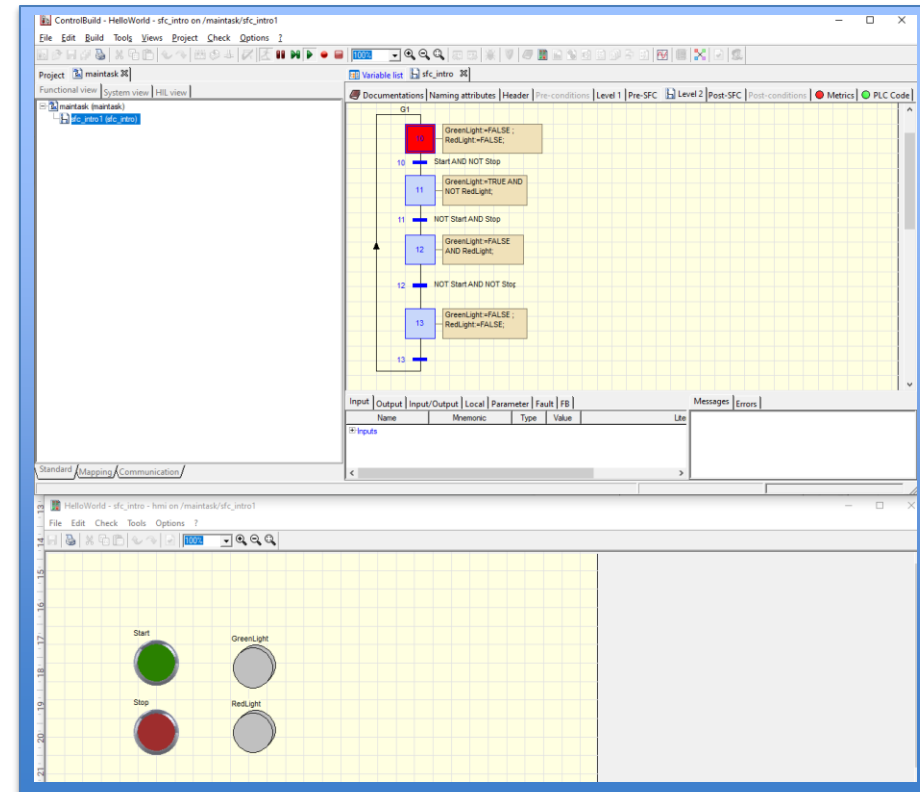


Testkör

1. Markera maintask och öppna upp functional view.
2. Öppna den nya fliken och öppna samtidigt upp hmi och sfc.
3. Tryck på den "springande gubben" och sedan på play.
4. Kontrollera att allt funkar som det skall.



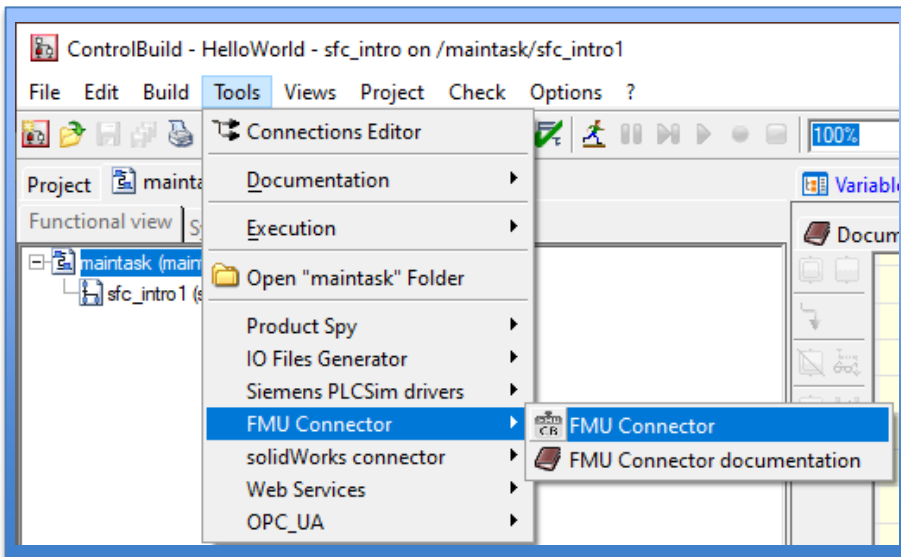
1



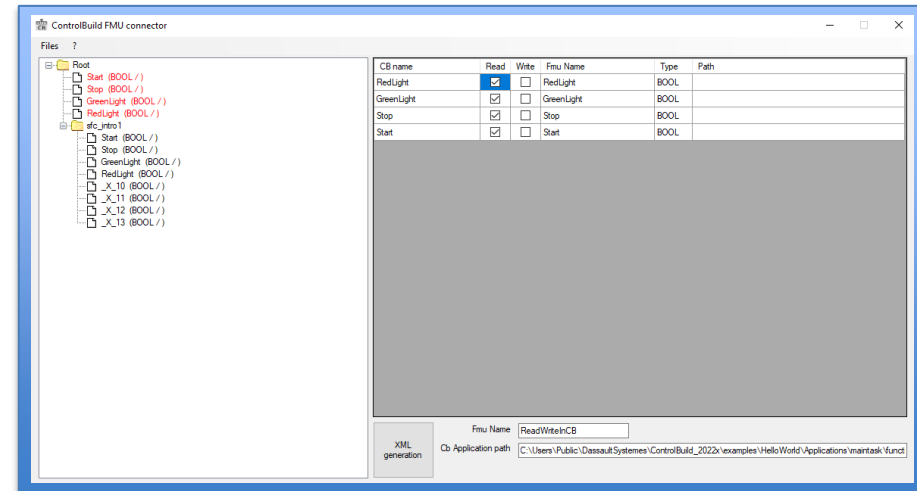
2 / 3

Importera och exportera FMU connector

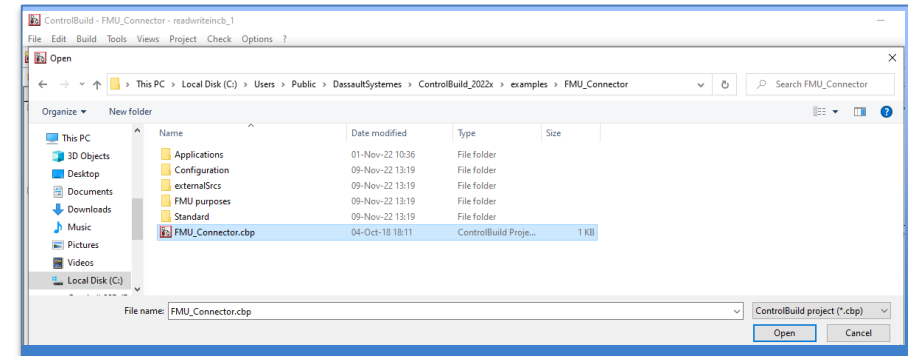
1. Öppna FMU Connector inne i Functional View via Tools.
2. Drag and drop de fyra komponenter under Root.
3. Se till att alla är markerade som Read och klicka sedan på XML generation.
4. Spara filen i
C:\.....DassaultSystemes\ControlBuild_2022x\examples\HelloWorld\Applications\maintask\functional\maintask
5. Öppna projektet FMU_connector som finns bland examples.



1



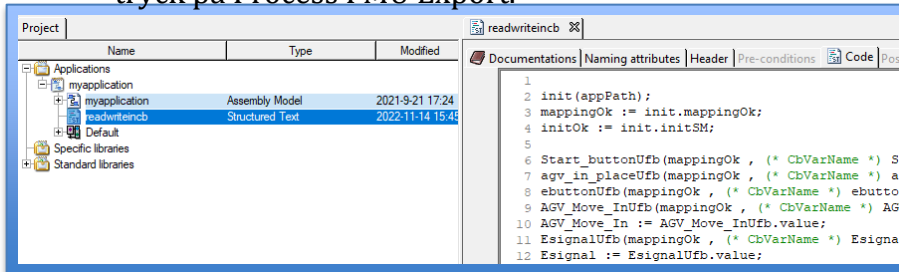
2 / 3



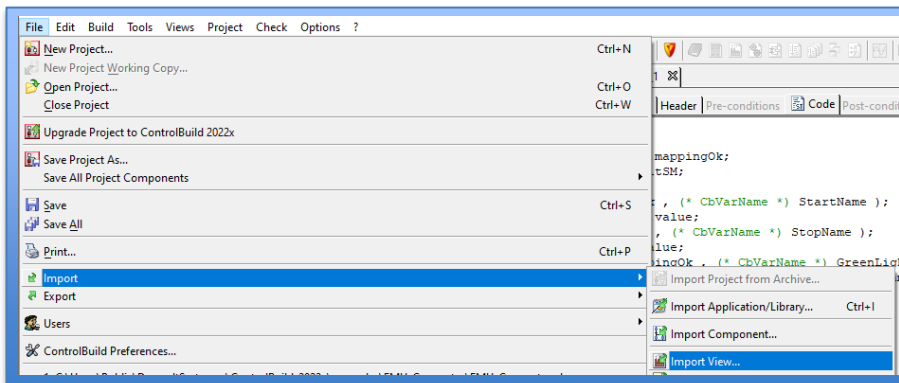
4

Write=Kan skrivas över på din SFC kod (Input på 3DX)
Read=Skriver över från till Controlbuild (Output på 3DX)

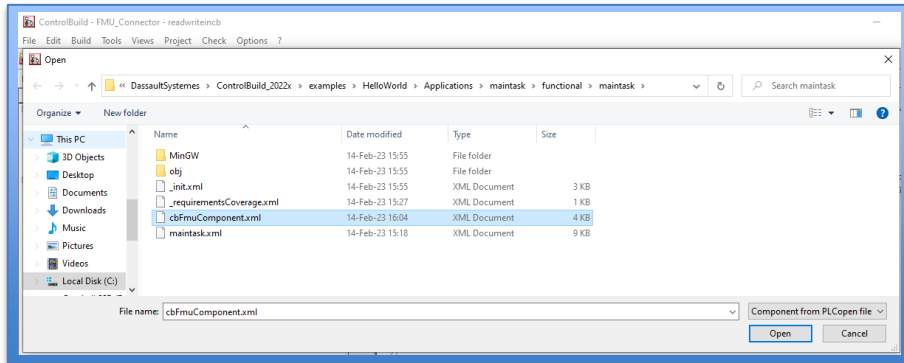
6. Öppna readwriteincb filen för att sedan trycka på Import Component.
7. Importera xml filen som sparades i steg 4.
8. Öppna den nya readwriteincb_1 och tryck på Export Component.
9. Ändra filtyp till 2.0 fmu och spara det på samma ställe som xml filen i HelloWorld
10. Ta bort allt förutom Start, Stop, GreenLight, RedLight och tryck på Process FMU Export.



Importer XML filen som skapades tidigare

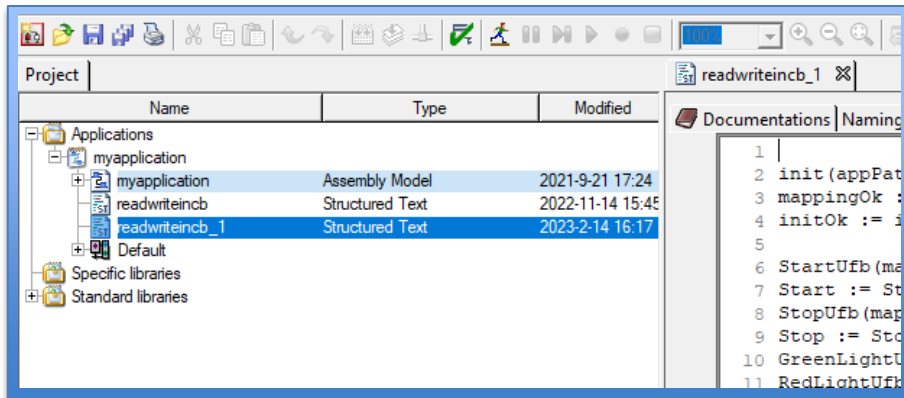


5



7

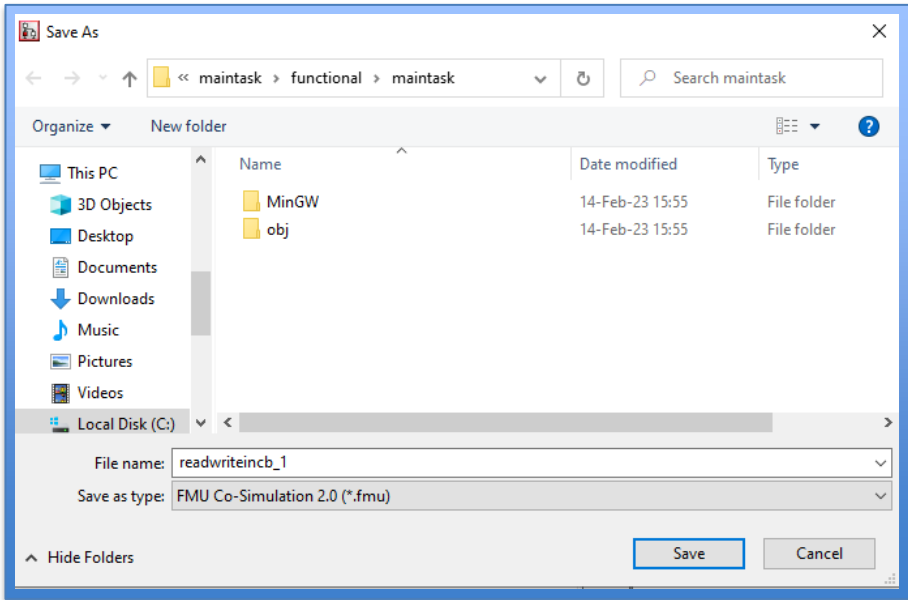
Dubbelklicka på den nya Structured Text och exportera



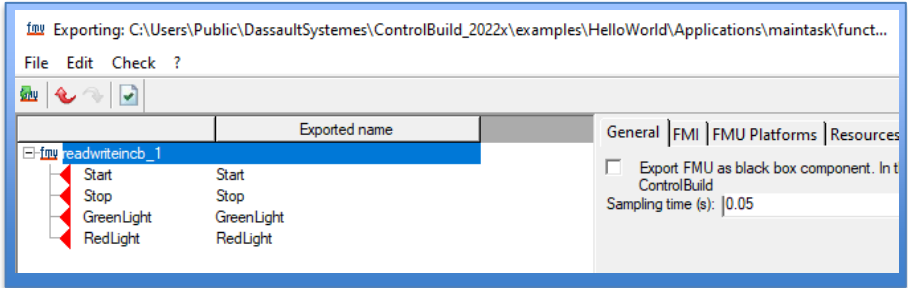
8

6

9

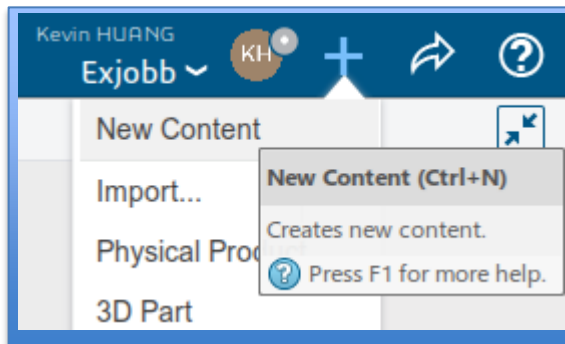


10

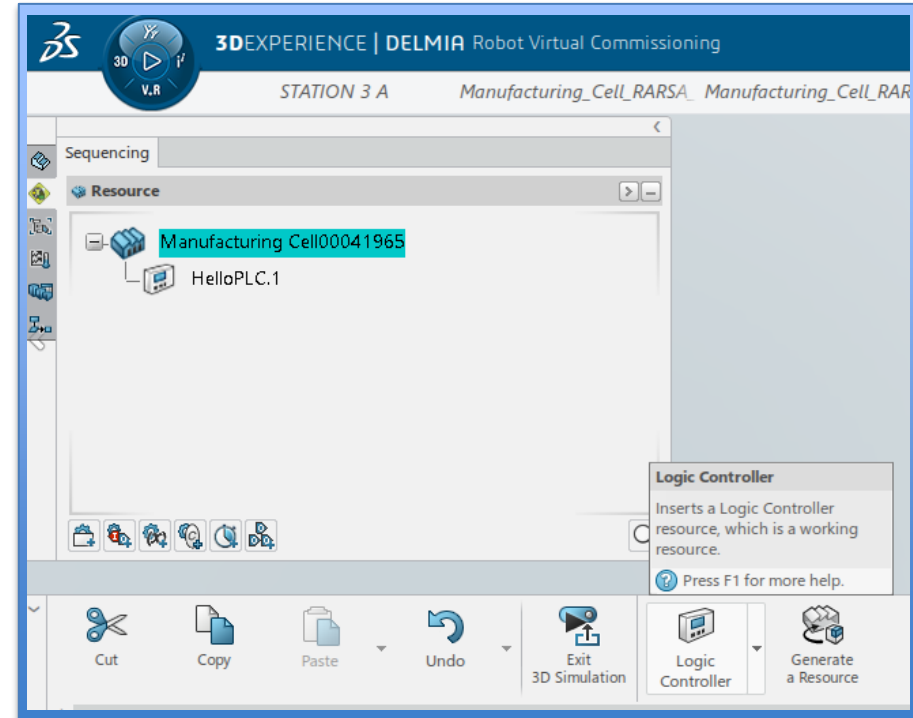


Skapa komponenter i 3DEXPERIENCE

1. Öppna upp Robot Virtual Commissioning och tryck på New Content för att skapa en ny manufacturing cell och lägg till Logic Controller (PLC) i manufacturing cell.
2. Högerklicka på PLC och tryck Associate FMU. Öppna nu upp "readwriteincb_1.fmu" som sparades från ControlBuild inne i HelloWorld mappen.
3. Komponenter bör vara outputs.
4. Skapa en Heartbeat-task genom att högerklicka på PLC och välj type Service.
5. Tasken ska vara uppbyggd som en while loop med wait false 0.5s.

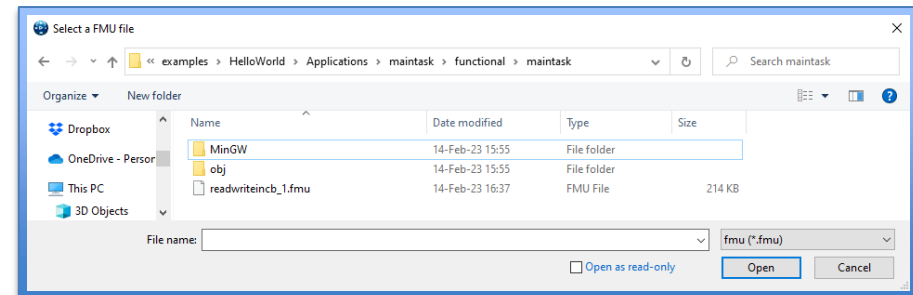


1

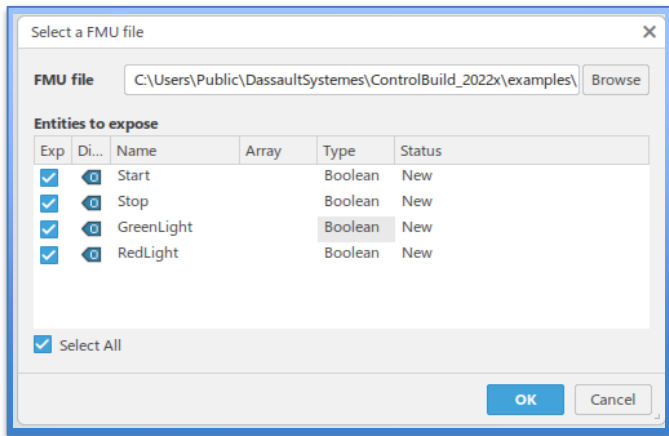


1

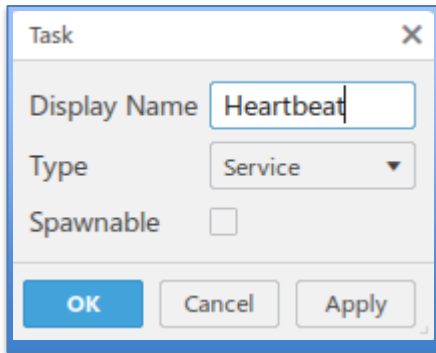
Högerklicka på PLC och associate FMU outputs



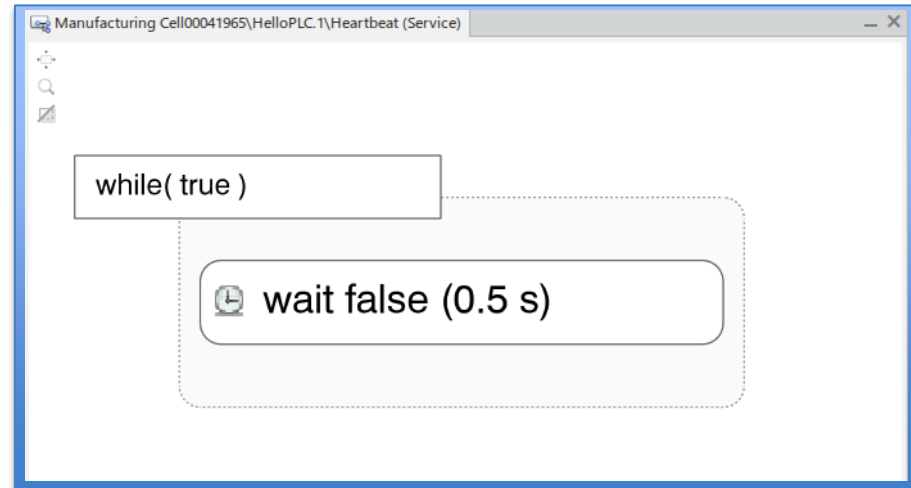
2



3



4



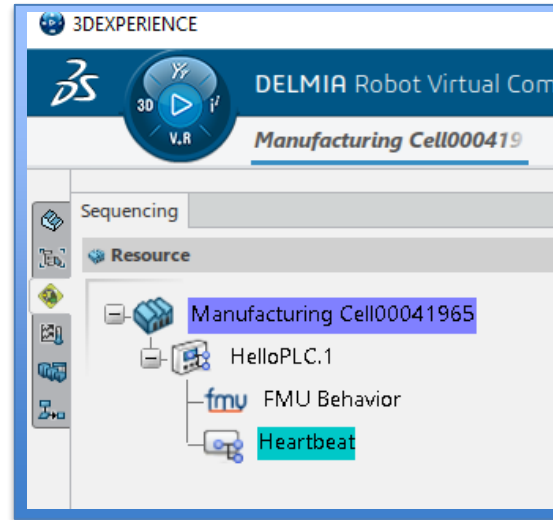
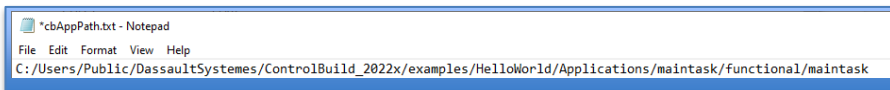
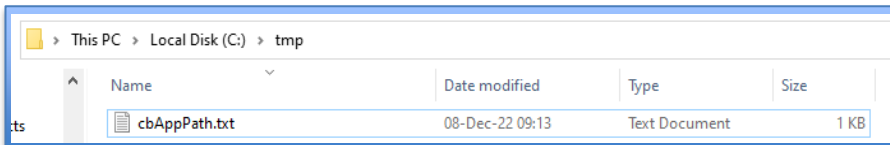
5

Sammanflätning och testkör

1. Skapa en notepad i "C:\tmp" och i den ska det finnas sökvägen till "readwriteincb_1.fmu".

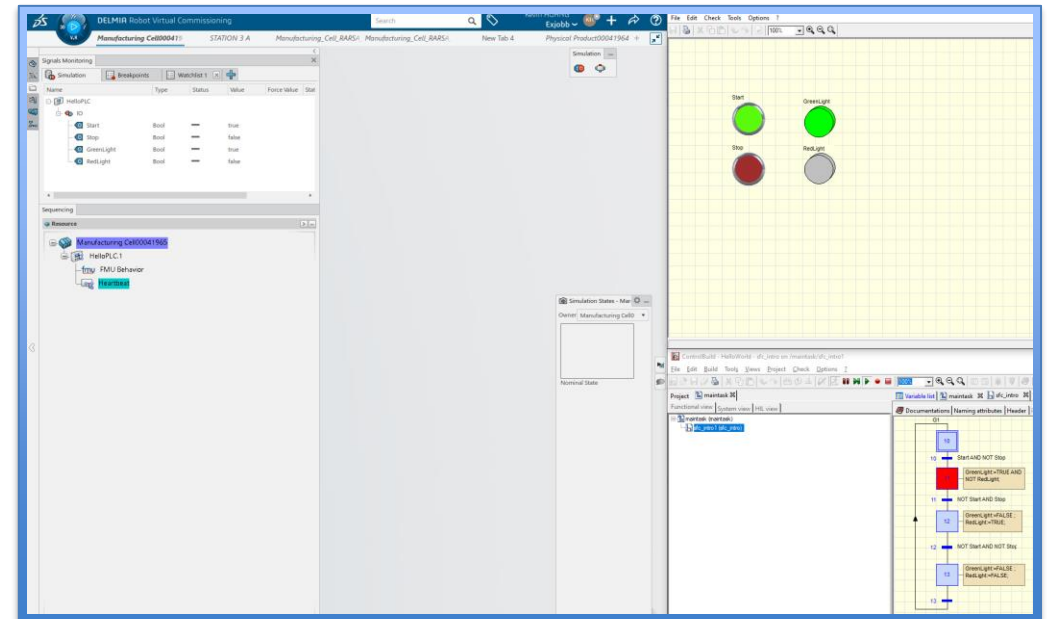
OBS!! det ska vara forward slash [/]

2. Starta simuleringen i 3DEXPERIENCE och ControlBuild.
Därefter tryck på play i ControlBuild först.
3. Öppna fliken Signal Monitoring för att se Value.
4. Kontrollera att sammankopplingen funkar.



2

1



3

INSTITUTIONEN FÖR INDUSTRI- OCH
MATERIALVETENSKAP
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige 2023
www.chalmers.se



CHALMERS