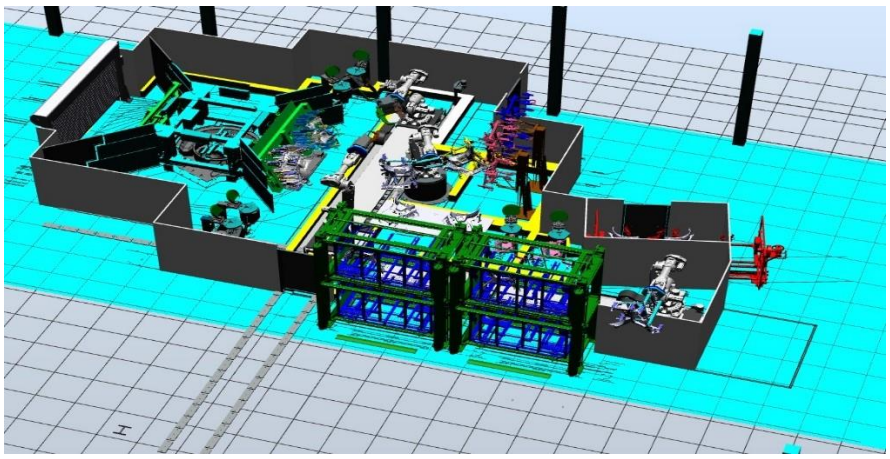
Professionele Bachelor Elektromechanica

Onderhoudstechnologie



Ciratec & C° - ABB ROBOTSTUDIO

Glenn Wauters

Promotoren:

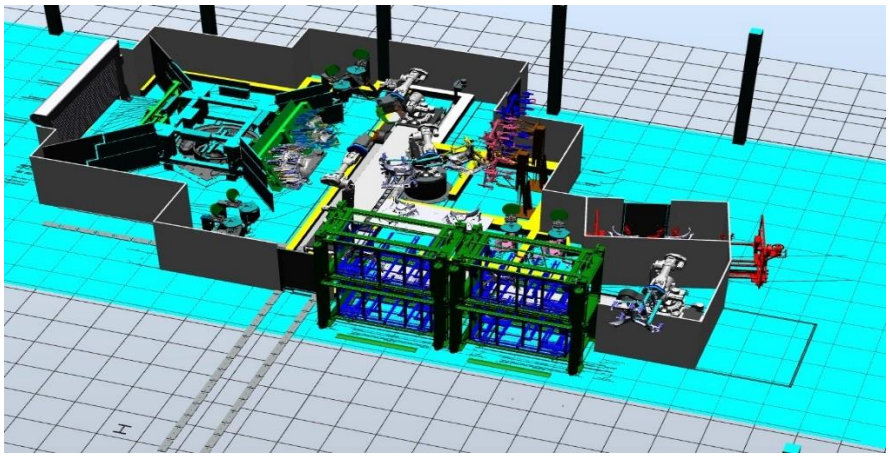
Robby Cloesen — BVBA Ciratec & C°

Ilona Stouten — PXL Hogeschool

Bachelorproef academiejaar 2013-2014

Professionele Bachelor Elektromechanica

Onderhoudstechnologie



Ciratec & C° - ABB ROBOTSTUDIO

Glenn Wauters

Promotoren:

Robby Cloesen — BVBA Ciratec & C°

Ilona Stouten — PXL Hogeschool

Bachelorproef academiejaar 2013-2014

Dankwoord

Met dit dankwoord wil ik me richten tot de mensen die geholpen hebben bij het tot stand brengen van deze bachelorproef, voor hun steun, hulp en vertrouwen.

Eerst en vooral zou ik graag mijn bedrijfspromotor dhr. Robby Cloesen willen bedanken voor de uitstekende begeleiding gedurende mijn stage. Zijn steun, advies en deskundige kennis hebben voor mij een duidelijke meerwaarde betekend.

Ten tweede richt ik mijn dankwoord aan mijn PXL-promotor, mevr. Ilona Stouten die mij met raad en daad bijstond doorheen het gehele proces. Haar raadgeving en adviezen zijn een grote hulp geweest. Ook haar lessen waren een bron van informatie.

Daarnaast wil ik ook mevr. Hannelore Dierickx en dhr. Patrick Pilat bedanken voor de praktische regeling van de stage en het controleren van de bachelorproef op taalvaardig niveau.

Tenslotte bedank ik ook de collega's van Ciratec & C° voor de fijne samenwerking, de steun en uitleg gedurende mijn stage. Ook wil ik mijn ouders en broers bedanken voor het nalezen van de bacherlorproef en voor de steun. Een speciaal dankwoord is voor dhr. Pascal Droogmans voor het logement dicht bij het bedrijf gedurende de 9 weken stage.

Inhoudsopgave Bacherlorproef

Abstract

Gedurende 9 weken dien ik in het bedrijf Ciratec & C° mijn stage uit te voeren. Dit bedrijf is gespecialiseerd in het programmeren en simuleren van industriële robots en PLC (Programmable Logic Controller) in Europa. Hierdoor is robotica het onderwerp van mijn stage.

De doelstellingen die mij worden opgelegd tijdens mijn stage, zijn het bereiken van kennis en vaardigheden voor de opbouw, het programmeren en simuleren van een bestaande robotcel met 4 ABB Robots. Het uitvoeren van deze doelstellingen worden volbracht via het programma ABB RobotStudio en via praktische toepassingen.

De methode voor het uitvoeren van deze doelstellingen geeft een beeld weer van hoe ik te werk ga, om het gewenste resultaat te komen. Het programmeren van de robotcel gebeurt m.b.v. het programma ABB RobotStudio. Gedurende de eerste 2 weken wordt de 3D-opbouw gemaakt van de robotcel. Hierbij worden robots, grijpers, lastangen, een draaitafel en veel andere onderdelen en toestellen via ingegeven coördinaten op de lay-out geplaatst.

Vervolgens ga ik de bewegingen van de 4 robots, die voorzien zijn van een grijper of lastang, programmeren. Na het programmeren van deze bewegingen, neem ik de BMW standard door zodat ik voor elke robot de juiste logica kan schrijven. Deze logica houdt in dat ik een overzicht krijg van de acties die de robots moeten ondergaan en waarbij elke robot in communicatie staat met de PLC. Hierdoor wordt in de logica heel het programma overlopen, dat volgens de BMW standard wordt voorgedragen.

Als resultaat zal het ontwikkelde programma gesimuleerd worden. Vervolgens wordt het desbetreffende programma ingeladen op het moment dat de werkelijke robotcel is opgebouwd in Oxford. Nadien gaat het omgezet worden in de praktijk en kan het in werkelijkheid toegepast worden. Dit zorgt voor een mooi resultaat en afsluiter van de stage.

Uit mijn stage kan ik concluderen of Ciratec & C° in de toekomst meer kan gaan werken met het programma ABB RobotStudio en of dit geschikt is voor de diensten die het bedrijf levert.
De verschillende robots en programma's om dit te verwezenlijken zijn heel interessant en we kunnen hier veel mee bereiken. ABB RobotStudio, het programma dat gebruikt wordt tijdens de stage, heeft zijn voor- en nadelen. Hierdoor is ABB RobotStudio op sommige vlakken een uitstekend programma, maar op andere vlakken is het minder goed. Door dit totaalpakket van doelstellingen en de vraag naar deze mensen zal de stap in de wereld van robots een uitstekende keuze zijn voor een werkzekere job.

1. Lijst gebruikte afkortingen

ABB: Asea Brown Boveri

PLC: Programmable Logic Controller

BMW: Bayerische Motoren Werke

PLM: Product Lifecycle Management

ISO: International Organization for Standardization

CAD: Computer-Aided Design

UCS: User Coordinate Sytem

TCP: Tool Center Point

2. Het stagebedrijf: Ciratec & C°

Het bedrijf Ciratec & C° is gevestigd in Diepenbeek en werd opgericht door dhr. Jean Cloesen in 1995 onder de naam Robot Automation.

In de beginjaren van Robot Automation richtte het bedrijf zich vooral op het programmeren van industriële robots. Door de toenemende vraag naar grotere projecten met betrekking tot automatisering, werd in november 1998 een geheel nieuwe bedrijfsstructuur ontwikkeld. Hierdoor werd een nieuwe bedrijfsnaam gekozen, met name Ciratec.

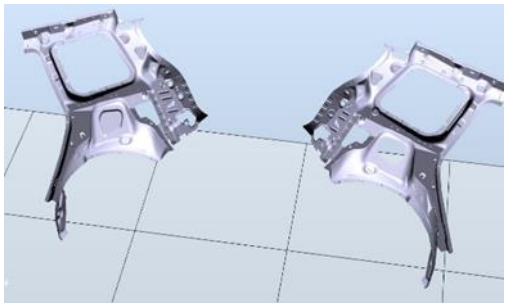
Sinds januari 2008 werd het beheer van Ciratec uitgebreid en begon het bedrijf zijn activiteiten te verwezenlijken onder de naam Ciratec & C °.

Ciratec & C ° is op heden een groeiende onderneming en biedt verscheidene diensten aan, namelijk de inbedrijfstelling en programmering van industriële robots, maar ook de volledige simulaties van geautomatiseerde cellen. Bovendien is Ciratec & C° gespecialiseerd in het optimaliseren van de bestaande productielijnen, het schrijven van klantspecifieke softwareprogramma's en het verstrekken van technische opleiding. [1]

3. Onderwerp en doelstellingen stage

Gedurende 9 weken stage word ik ondergedompeld in de wereld van robots . Hierdoor wordt robotica het onderwerp en de doelstelling van de stageperiode en de bachelorproef.

Het onderwerp is een totaalpakket van de 3D-opbouw, het programmeren en simuleren van een robotcel waardoor de belangrijkste stappen van het hele proces overlopen worden. Afhankelijk van de omstandigheden en de tijd zou deze robotcel opgebouwd worden in de body shop van de MINI fabriek in Oxford. Deze robotcel wordt gebruikt voor het produceren van het achterste carrosserie onderdeel, dat op onderstaande figuur (figuur 1) weergeven is. Dit deel zal dan worden geproduceerd voor de nieuwe MINI Cooper, die aangetoond wordt in figuur 2. Waardoor het resultaat in werkelijkheid tot zijn recht komt.



Figuur 1: Het carrosserie-onderdeel

Figuur 2: De MINI Cooper

De belangrijkste doelstelling van de stage is, het achterhalen of Ciratec & C° in de toekomst meer kan gaan werken met het programma ABB RobotStudio en of dit geschikt is voor de diensten die het bedrijf levert.

Om dit te onderzoeken, wordt heel het proces opgesteld van 3D-opbouw tot het simuleren van het resultaat. Het volledig proces voor het maken van de robotcel zal via ABB RobotStudio gemaakt worden. Door de hoofddoelstelling onder te verdelen in aparte doelstellingen, ontdek en leer je werken met het programma ABB RobotStudio en wordt je kennis van robotica uitgebreid.

De eerste doelstelling van de stage is het opmaken van de 3D-opbouw van de robotcel. Deze bestaat uit vier ABB robots waarvan 2 lasrobots en 2 robots met grijper.

De 2de doelstelling die van groot belang is in het proces, is het doornemen van de BMW standard. Deze beschrijft alles wat verwacht wordt om bewegingen aan te leren en om de logica van de robots te schrijven.

Na het bereiken van deze doelstellingen worden de bewegingen en logica van de robotcel gemaakt, met als controle een simulatie. Deze simulatie is het resultaat van het opbouwproces en zorgt ervoor dat er een overzicht wordt weergeven van hoe alles in de robotcel gaat bewegen. Indien er eventuele fouten tijdens de simulatie optreden, kunnen die worden aangepast.

Tot slot zouden we naar de MINI fabriek in Oxford afreizen voor het effectief inladen van het gemaakt programma via ABB RobotStudio. Door omstandigheden is de opbouw van de robotcel met een paar weken uitgesteld, waardoor het programma niet ingeladen kan worden. Om toch praktijk en kennis op te doen, zijn we een andere robotcel in het fabriek VDL Nedcar in Nederland gaan ontdekken. Het praktisch werken met de controller van een robot wordt aangeleerd en belangrijke informatie wordt aangehaald.

4. Programma’s stage

4.1 ABB RobotStudio

ABB RobotStudio 5.61 is een programma dat aangeboden wordt door ABB Robotics en dat gedurende de gehele stageperiode gebruikt wordt. Via dit programma zal heel het proces van de robotcel gemaakt worden. Op onderstaande figuur (figuur 3) is het symbool van het programma ABB RobotStudio 5.61 weergegeven.

Figuur 3: Symbool ABB RobotStudio 5.61

Dit programma is geschikt voor ABB simulatie en het offline programmeren van ABB robots. ABB RobotStudio is een goede investering en de beste manier om ABB robotsystemen te maximaliseren. Via dit programma moet u niet meer op de werkplaats aanwezig zijn, maar kan u vanop kantoor via het programma op uw computer werken zonder dat u hiervoor de productie afsluit.

Indien de kennis van het programma ondermaats is, biedt RobotStudio hulpmiddelen aan. Zo zorgt ABB voor opleiding, programmering en optimalisatie zonder de productie te storen. Deze software biedt tal van voordelen, waaronder: risicobeperking, snellere start-up, kortere change-over een verhoogde productiviteit.

RobotStudio is opgebouwd uit de ABB Virtual Controller, een exacte kopie van de echte software die uw robots laat draaien in de productie. Dit zorgt er voor dat realistische simulaties uitgevoerd kunnen worden, zoals met echte robotprogramma’s en configuratiebestanden zoals gebruikt wordt op de werkvloer.

De software voor dit programma kan via de website van ABB Robtics of via een aangekochte Cd-rom geïnstalleerd worden. Doordat ABB RobotStudio een gratis licentie heeft van slechts 30 dagen, heb ik het programma twee keer moeten downloaden tijdens de stageperiode. [2]

Bijlage 1: weergeeft het programma ABB RobotStudio 5.61

4.2 JT2Go

JT2Go is het meest gebruikte 3D-bestandsformaat voor PLM, (Product Lifecycle Management) ter wereld. Op onderstaande figuur (figuur 4) is het symbool van het programma JT2Go weergegeven.

Figuur 4: Symbool JT2Go

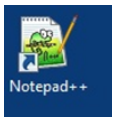
JT is de gemeenschappelijke taal van PLM. Hierdoor is dit de eerste ISO internationale standaard (IS 14306 - 1) voor 3D-visualisatie en samenwerking. Dit bestandsformaat is compact, nauwkeurig en wordt gebruikt gedurende de volledige productlevenscyclus van de ontwikkeling in alle belangrijke industrieën om te communiceren met de kritische ontwerpinformatie, die meestal vergrendeld is in een CAD-bestand.

JT2Go werd ontwikkeld op verzoek van de JT open lidmaatschap en is de beste keuze voor het gratis bekijken van JT bestanden. Hierdoor is JT2Go geschikt voor de visualisatie van JT-bestanden. [3]

Bijlage 2: Illustreert het programma JT2Go

4.3 Notepad++

Notepad ++ is een gratis broncode-editor die verschillende talen herkent en ondersteunt. [4]
Op onderstaande figuur (figuur 5) is het symbool van het programma Notepad++ weergegeven.



Figuur 5: Symbool Notepad++

Het programma Notepad++ zoals weergegeven in figuur 8, zorgt ervoor dat we de logica van de verschillende robotsystemen kunnen gaan bekijken en eventueel aanpassing aanbrengen.

Notepad++ heeft twee belangrijke voordelen. In het eerste geval zorgt het programma ervoor dat het bestand onder de juiste naam wordt opgeslagen, zodat ABB RobotStudio dit correct kan inladen. Als tweede voordeel kan Notepad++ ook bestanden openen die door ABB RobotStudio niet ingeladen kunnen worden, waardoor het programma geschikt is om fouten te achterhalen. Via aanpassingen in het programma kan ervoor gezorgd worden dat RobotStudio de logica correct en zonder fouten opnieuw kan inladen.

Bijlage 3: Illustreert het programma Notepad++

5. De functie van de robotcel

De robotcel die ontworpen en geprogrammeerd is zoals weergegeven in bijlage 4, zal geplaatst worden in de MINI fabriek in Oxford. De functie van deze robotcel is het puntlassen van het achterste auto onderdeel van de nieuwe MINI Cooper (figuur 1 en 2) Deze robotcel is ontworpen voor het puntlassen van het linkse en rechtse carrosserie onderdeel.

Hoe deze robotcel zijn functie gaat uitvoeren voor het proces, wordt in volgende stappen verduidelijkt.

Eerst worden de te maken carrosserie onderdelen naar de robotcel gebracht. Wanneer alles in veiligheid is, wordt de procedure gestart. Via de ingangspoort worden twee onderdelen op de draaitafel geplaatst die door middel van klemmen en pinnen de onderdelen stevig klemt.

De draaitafel draait zich in de richting van de twee lasrobots. Wanneer alles in zijn positie staat, wordt het eerste programma gestart. De twee lasrobots en 7de as starten de procedure van het puntlassen voor het linkse en rechtse auto-onderdeel.

Tijdens het puntlassen wordt eerst gebruikt gemaakt van de X-tang. Na deze toepassing verplaatst de robot zich naar de docking om te verwisselen naar een C-tang. Het proces wordt verdergezet en de andere puntlassen worden aangebracht.

Doordat de twee robots op hetzelfde moment gaan bewegen, is het noodzakelijk dat deze robots niet tegen elkaar gaan botsen of dat ze stoppen tijdens het proces. Hierdoor wordt een beweging aangeleerd die ervoor zorgt dat elk onderdeel voorzien wordt van een puntlas, zonder dat de robots gaan raken of dat de een moet wachten op de ander.

Na het puntlassen bewegen de twee lasrobots naar hun homepositie, zodat de robot met dubbele grijper de onderdelen vanuit de draaitafel kan opnemen. Wanneer de dubbele grijper deze onderdelen vastheeft, zullen de klemmen en pinnen van de draaitafel opgaan zodat de dubbele grijper de onderdelen zelf kan vastnemen en meebewegen naar de volgende positie.

De positie waar het proces zich nu bevindt, is de aflegtafel. Op deze plaats heeft de dubbele grijper de onderdelen neergezet en terug bewogen naar zijn homepositie.

De vierde robot met de enkele grijper neemt elk onderdeel apart van de aflegtafel en plaatst deze in één van de rekken die voorzien zijn van containers. In deze containers worden, tijdens het proces, de gemaakte carrosserie onderdelen geplaatst. Indien de containers vol zijn, zal een heftruck deze komen halen en verplaatsen naar de volgende robotcel en proces.

Tot slot worden nieuwe containers geplaatst. Wanneer alles terug in zijn homepositie staat, wordt de procescyclus opnieuw herhaald waardoor we continue productie creëren.

6. 3D-opbouw van de robotcel

De eerste stap in het proces van de stage, is de 3D-opbouw van de robotcel. Via het programma ABB RobotStudio wordt de volledige robotcel opgebouwd zoals het er in werkelijkheid zal gaan uitzien.

Binnen het proces heeft iedereen zijn taak. De toestellen en onderdelen die in de robotcel aanwezig zijn gemaakt door de ontwerpers. Via deze ontwerpen, die we ontvangen in een bibliotheek, kan door de programmeurs de 3D-opbouw perfect worden nagemaakt.

6.1 Inladen onderdelen en toestellen

Via de aangemaakte bibliotheek van de ontwerpers en de bibliotheek van ABB kunnen we elk toestel of onderdeel in het programma inladen en vervolgens op de gewenste coördinaten plaatsen.

Doordat ABB RobotStudio over een eigen bibliotheek beschikt, kunnen alle onderdelen en toestellen waarvan ABB de kennis over heeft, ingeladen worden. Tijdens de 3D-opbouw wordt hierdoor gebruik gemaakt van de ABB robots en de 7de as.
Het probleem dat we ondervinden in de bibliotheek van ABB RobotStudio, is dat het niet beschikt over alle onderdelen en toestellen die aanwezig moeten zijn in de 3D-opbouw. Omwille van dit probleem maken we gebruik van de aangemaakte bibliotheek die ons gegeven wordt door de ontwerpers. Via deze weg kunnen we onderdelen zoals draaitafel, aflegtafel, tangen, grijpers en veel andere onderdelen in het programma gaan importeren.

6.2 Bestandstype voor ABB RobotStudio

Het bestandstype is een belangrijk item voor het correct inladen van toestellen en onderdelen. Bij gebruik van het gewenste type zal hierdoor alles ingeladen worden met een hoge kwaliteit.

De bibliotheek van ABB RobotStudio wordt standaard geïnstalleerd bij de installatie van de software, waardoor de onderdelen en toestellen over het juiste bestandstype, namelijk RSGFX, beschikken.

Het probleem met de opgemaakte bibliotheek is dat de onderdelen die ons gegeven worden, opgeslagen zijn als een JT-bestand. Dit kan ABB RobotStudio niet lezen of inladen. Het is dus noodzakelijk na te gaan welke bestandstypes het programma kan inladen. De verschillende bestandtypes die door ABB RobotStudio gelezen kunnen worden zijn weergegeven in tabel 1.

• ACIS • IGES • STEP • VDAFS • PRO/ENGINEER • INVENTOR • CATIA V4	• CATIA V5 • VRML • STL • COLLADA • OBJ • 3DS • RSGFX

Tabel 1: Bestandtypes ABB RobotStudio

De beste oplossing, om dit probleem op te lossen, is het JT bestand via de CAD Converter van RobotStudio om te zetten naar één van de bestanden die RobotStudio wel kan inlezen.

Door het uittesten van de verschillende bestandtypes, is de keuze uitgegaan naar het RSGFX-type. Het type biedt twee voordelen: hoge kwaliteit en laag aantal KB.

Eerst en vooral wordt door de kwaliteit een duidelijk beeld van het onderdeel of toestel weergegeven.

Het tweede voordeel heeft betrekking tot de bestandsgrootte. Wanneer deze over een weinig aantal KB beschikt, zal het programma minder gaan vastlopen. Dit is een belangrijk voordeel tijdens de opbouw en het programmeren, want het vastlopen van het programma is voor iedereen een pijnlijke situatie.

6.3 Het plaatsen van de onderdelen

Het plaatsen van de onderdelen is de volgende belangrijke stap en heeft als doel een duidelijke visualisatie van het station te verkrijgen.

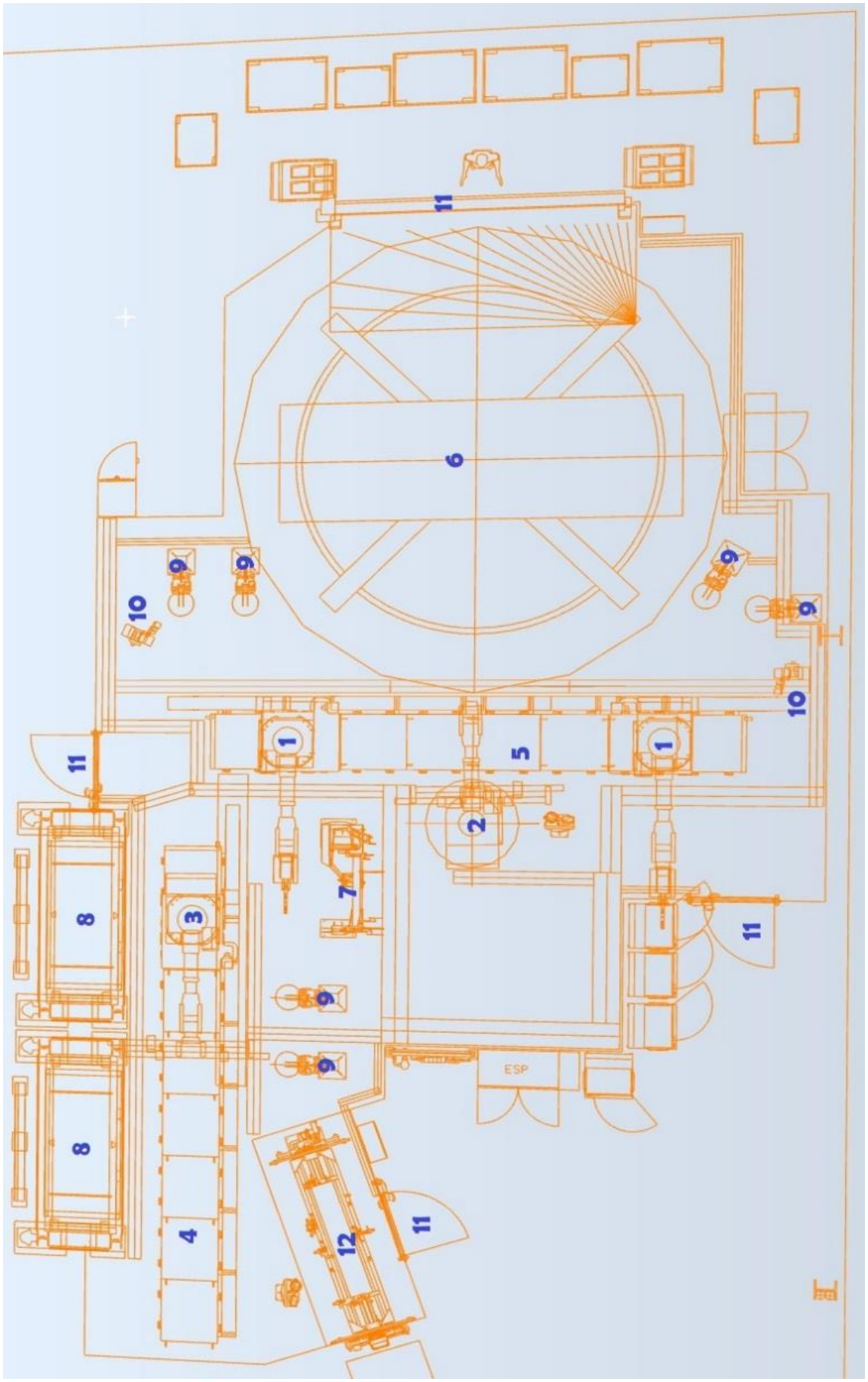
De onderdelen en toestellen die we hebben ingeladen worden weergeven in het tabblad layout. Dit is een lijst met een duidelijk overzicht van wat aanwezig is in het station.

Het voordeel van dit tabblad is dat we elk onderdeel of toestel apart op zijn gewenste positie in de robotcel kunnen plaatsen. Om van deze posities een duidelijk beeld te krijgen, beschikken we over een lay-out die we in het programma gaan inladen. Via deze manier kunnen we visueel bekijken waar de onderdelen of toestel in de robotcel moeten staan.

De onderdelen en toestellen die in de robotcel op de lay-out geplaats worden zijn in tabel 2 weergegeven en worden verduidelijkt In figuur 6.

1.	De twee lasrobots
2.	De robot met dubbele grijper
3.	De robot met enkele grijper
4.	De 7de as
5.	De dual 7de as
6.	De draaitafel
7.	De aflegtafel
8.	De rekken die voorzien zijn van containers
9.	De dockings
10.	De tipdress en tipchanger
11.	De veiligheidspoort en deuren
12.	De controle tafel
Rondom de lay-out	De veiligheidshekken

Tabel 2: Lijst onderdelen en toestellen die op de lay-out staan



Figuur 6: De lay-out van de robotcel

Om het onderdeel of toestel op de gewenste positie te plaatsen, maken we gebruik van gegeven coördinaten en de functie ‘set position’. Belangrijk bij deze programmering is rekening houden met de twee volgende punten.

Ten eerste moeten we de gewenste referentiekeuze maken (World, Parent, Local of UCS), zodat een onderdeel of toestel verplaatst kan worden t.o.v. het gekozen referentiepunt. Indien deze keuze niet correct is, zal het onderdeel of het toestel op een verkeerde positie gezet worden.

Het tweede belangrijke punt bij het programmeren van de positie, zijn het invoeren van de zes coördinaten.

- ➔ Position: X,Y,Z (mm)
- ➔ Orientation, RX,RY,RZ (deg)

Na het plaatsen van de onderdelen en toestellen op de gewenste positie in de lay-out, wordt de volgende stap belangrijk om de robot samen met de 7de as te laten bewegen.

Deze stap gebeurt via de functie ‘attach to’. Het wordt gebruikt om de onderdelen aan elkaar te zetten, zodat dit één geheel gaat vormen en ook als één geheel gaat bewegen. Indien een fout onderdeel werd vastgezet, is het mogelijk gebruik te maken van de functie ‘detach’. Deze zorgt ervoor dat de onderdelen van terug van elkaar gescheiden worden.

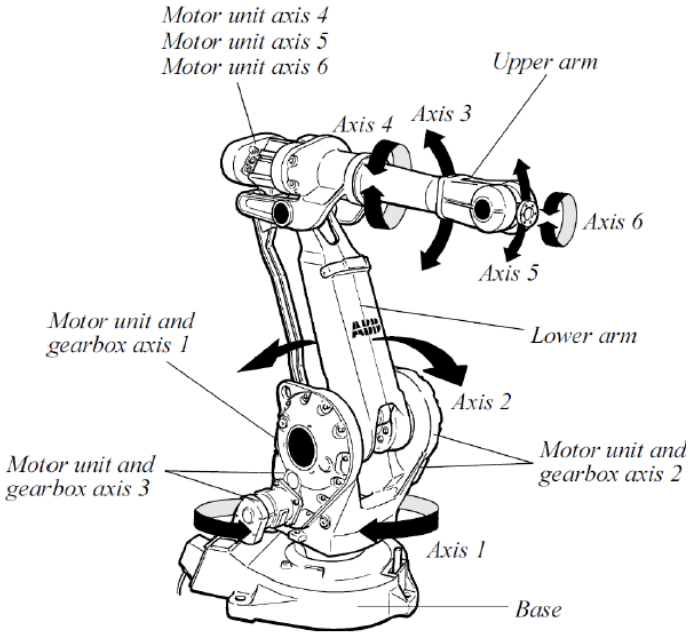
6.4 De onderdelen waaruit de robotcel is opgebouwd

De opbouw van de robotcel bestaat, zoals eerder vermeld, uit verscheidene onderdelen en toestellen. Deze zijn noodzakelijk binnen het proces en zijn daardoor ook van groot belang. In de volgende paragrafen worden de verschillende onderdelen in het kort besproken en weergegeven in een figuur

6.4.1 De robots

De ABB robots zijn de belangrijkste toestellen die in het programma ABB RobotStudio geplaatst worden en die betrekking hebben tot het hele proces.

De robot die we in het station toepassen, is opgebouwd uit 6 assen die een roterende of translerende beweging maken. De eerste 3 assen van de robot, die de robotarm voorstelt neemt de positie in de ruimte aan. De 3 andere assen of de robotpols bepaalt de oriëntatie van het gereedschap in de ruimte. Onderstaande figuur (figuur 7) verduidelijkt deze alinea.



Figuur 7: De robot met zijn 6 assen

De robots die gebruikt worden tijdens het proces, worden geïmporteerd via de bibliotheek van ABB RobotStudio. Via de aanwezige documentatie wordt voor elke toepassing de correcte robotkeuze gemaakt. Hierdoor zal het proces beschikken over de gewenste robots.

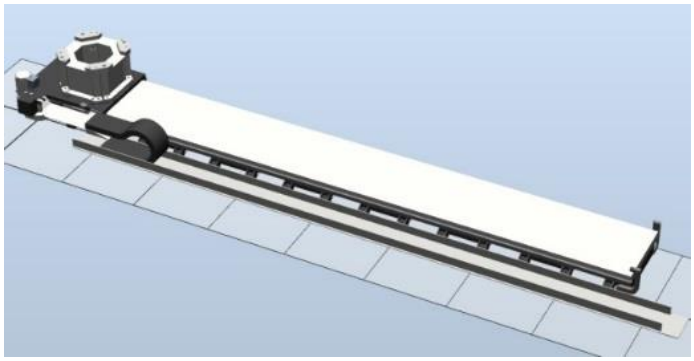
De robot, zoals weergegeven in figuur 8 en 9, zijn twee van de vier robots die worden gebruikt in het proces. Deze robots zijn van het type IRB6640 en zijn terug te vinden in de bibliotheek van ABB. Wanneer we deze robots gaan inladen, dienen we de hoogte (205 of 235) en het gewicht (275 of 255) van de robot aan te duiden. Deze keuze is afhankelijk van de soort toepassing, zoals het plaatsen van een grijper of lastang aan de robot.

Figuur 8: Robot IRB6640_205_275

Figuur 9: Robot IRB6640_235_255

6.4.2 De 7de as

De 7de as, zoals weergegeven in figuur 10, is een extra as waaraan de robot vastgezet kan worden. Via deze extra as kan de robot in horizontale of verticale richting bewegen, afhankelijk van de positie van de 7de as.



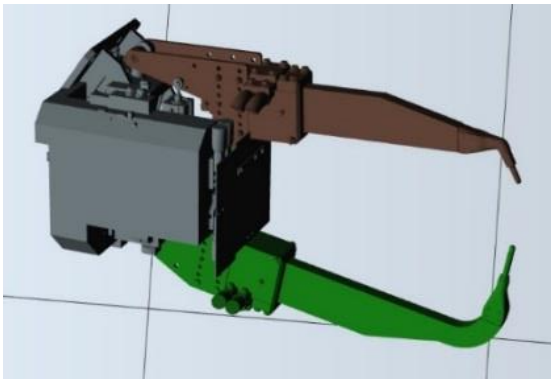
Figuur 10: De 7de as

Een probleem waarbij ABB RobotStudio te maken heeft, is dat in de bibliotheek geen dubbele 7de as te verkrijgen is. Dit is een nadeel en zorgt voor extra werk in het programma en het proces. De oplossing die hiervoor gebuikt wordt, is een tweede as te gaan inladen. We plaatsen deze op de gewenst coördinaten, maar gaan nu een rotatie van 180° maken. Door deze toepassing wordt een dubbele 7de as gevisualiseerd. Het gebruik van twee assen om de dual 7de as te maken, zorgt ervoor dat we beschikken over twee home posities.

6.4.3 De lastangen

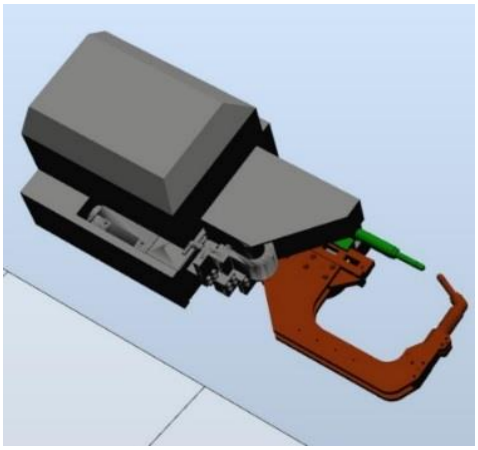
In het proces maken we gebruik van 2 lastangen die op één van de robots geplaats wordt. Deze tangen worden gebruikt voor het aanbrengen van puntlassen in het carrosserie onderdeel.

De eerste tang die we gebruiken voor het puntlassen is de X-tang. Waarom deze naam gekozen, is wordt geïllustreerd in figuur 11. Zowel de groene als de bruine tang gaan bewegen tijdens het lassen.



Figuur 11: X-tang

De tweede tang die we gebruiken in het proces is de C-tang, zoals weergegeven in figuur 12. In tegenstelling tot de X-tang zal slechts de groene tang gaan bewegen omdat het bruine gedeelte een vaste tang is.



Figuur 12: C-tang

De lastangen beschikken over 3 posities, namelijk: open, close en de home positie. Deze dienen geprogrammeerd te worden via de functie ‘create mechanisme’, zodat tijdens het lassen de tang de juiste bewegingen maakt.

Belangrijk om te weten, is dat tijdens het bewegen van de tangen, de opening tussen beide minimum 25 mm en maximum 150 mm mag zijn.

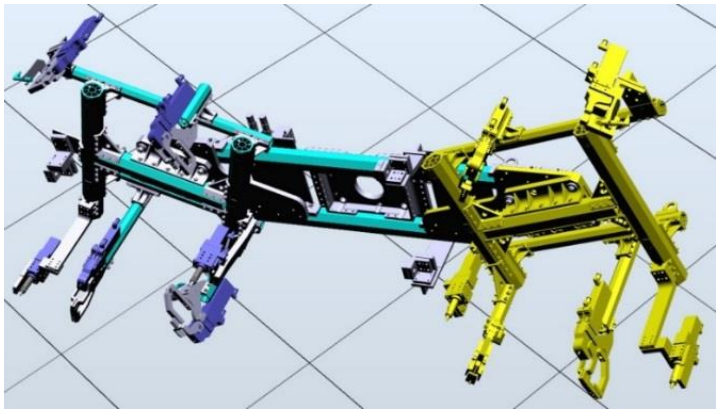
Doordat deze tangen in het proces veel laspunten op het carrosserie onderdeel gaan aanbrengen, zullen de kappen, zoals weergegeven in figuur 13, na een zekere tijd vervangen worden. Verdere uitleg wordt besproken in het hoofdstuk praktische toepassingen.

Figuur 13: Kap lastang

6.4.4 De grijpers

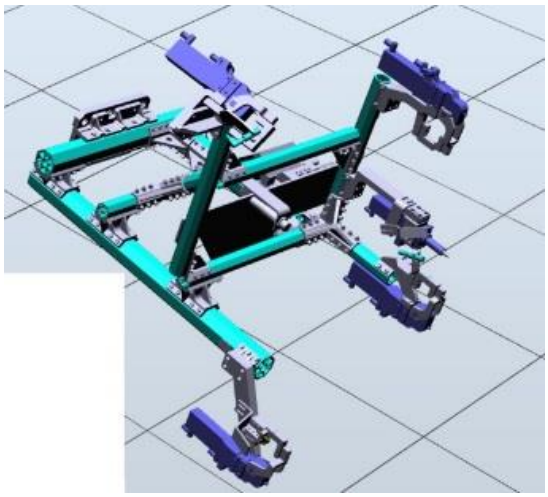
De grijpers worden in het proces toegepast om het carrosserie onderdeel vast te grijpen en doorheen de robotcel te bewegen naar de volgende aflegpositie.

De eerste grijper waarover we beschikken, is de dubbele grijper en wordt weergegeven in figuur 14. De grijper is voorzien van 2 grijpersystemen voor zowel het linkse als het rechtse auto onderdeel van de draaitafel op te pakken. Vervolgens zal deze grijper het onderdeel neerleggen op de volgende aflegpositie, namelijk de aflegtafel.



Figuur 14: De dubbele grijper

De 2de grijper, zoals weergeven in figuur 15, wordt gebruikt om slechts 1 auto-onderdeel op te pakken van de aflegtafel. Vervolgens zal het onderdeel in de juiste container geplaatst worden. De keuze van de container, hangt af van het feit of de grijper een links of een rechts auto onderdeel vast heeft. Indien we een andere auto onderdeel gaan oppakken, dienen we de grijper te verwisselen in de docking die aanwezig is in het station.

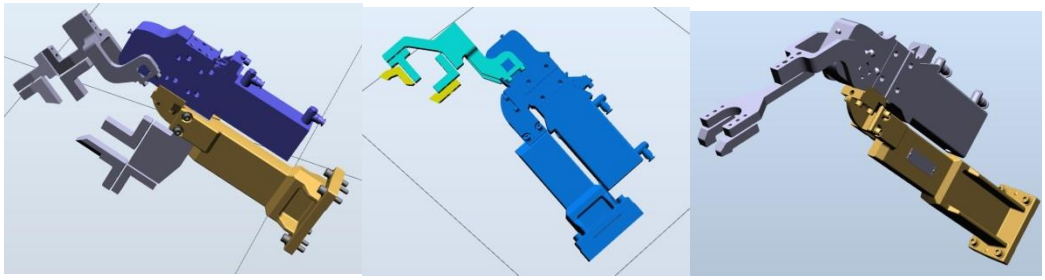


Figuur 15: De enkele grijper

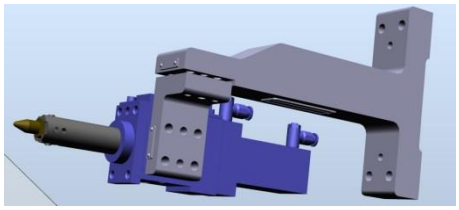
6.4.5 De pinnen en klemmen

De klemmen en pinnen zijn belangrijke onderdelen die ervoor zorgen dat het auto onderdeel wordt vastgehouden.

Deze zijn aanwezig op de draaitafel en grijpers. Ze zullen ervoor zorgen dat het deel stevig vastgezet wordt, zonder dat het gaat verschuiven tijdens een beweging. Voor het openen en sluiten van deze onderdelen maken we opnieuw gebruik van de functie ‘create mechanisme’, zodat deze kunnen bewegen tijdens het proces. Onderstaande figuren geven (figuur 16 en 17) een gedeelte van de klemmen en de pin weer.



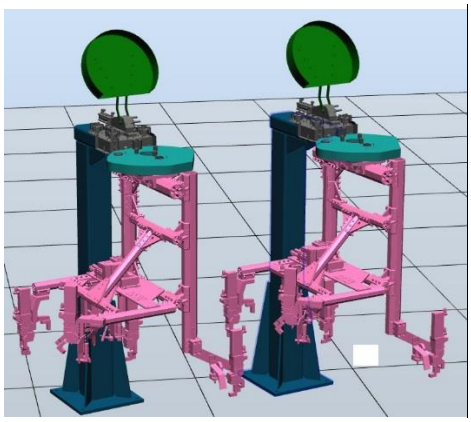
Figuur 16: De klemmen



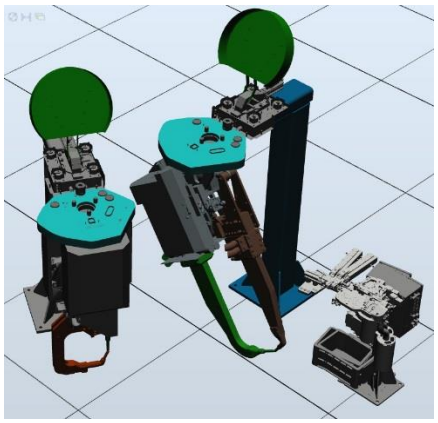
Figuur 17: De pin

6.4.6 De dockings

Deze toestellen, zoals weergegeven in figuur 18 en 19 zijn in het station voorzien om ervoor te zorgen dat we kunnen verwisselen van gereedschap. Zo zal gedurende het proces gewisseld worden van de X-tang naar de C-tang. Ook de enkele grijper zal gewisseld worden en zowel het linkse als het rechtse carrosserie-onderdeel op te pikken en neer te zetten.



Figuur 18: Docking voor enkele grijper

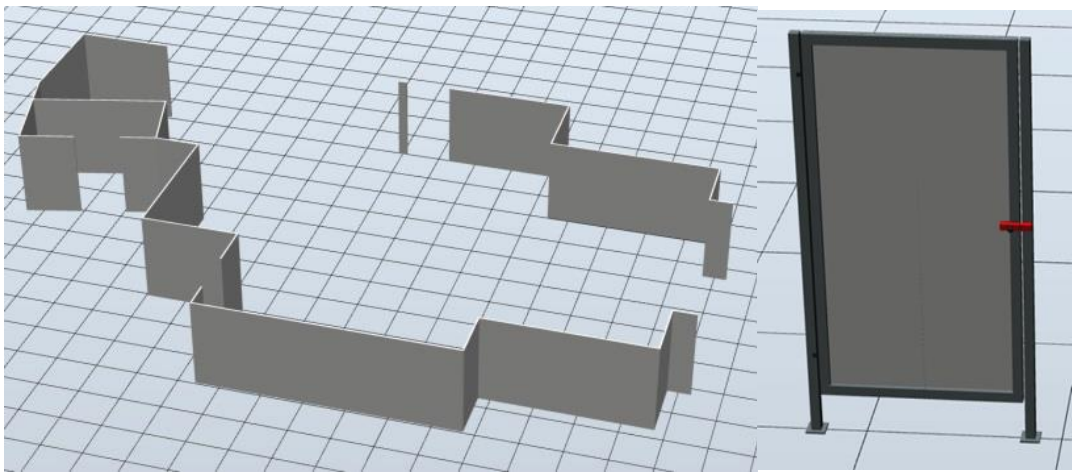


Figuur 19: Docking lastangen

6.4.7 Veiligheidstoebehoren

Veiligheid is een zeer belangrijk item binnen elk bedrijf. Doordat de robotcel zoals in de werkelijk word opgebouwd, is het ook noodzakelijk dat in het station de veiligheidshekken en deuren aanwezig zijn.

Onderstaande figuur (figuur 20) toont de veiligheidstoebehoren aan.

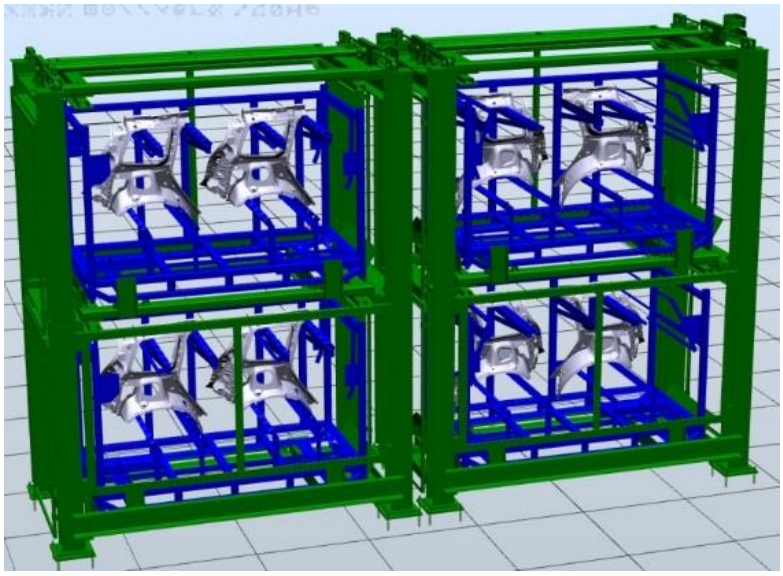


Figuur 20: Veiligheidstoebehoren

Indien er problemen zijn, kan de arbeider steeds de robotcel betreden. Deze veiligheidsdeuren worden open gedaan via een speciale sleutel en er zal een rode lamp gaan branden, zodat iedereen weet dat één van de deuren open is. Voor veiligheidsreden moet je bij het betreden van de robotcel een veiligheidsbril, -pet en -schoenen aanhebben. Ook moet je met een slot de deur vergrendelen, zodat niemand anders de deur kan sluiten en opdat de collega's weten dat er iemand aanwezig is in de cel. Ingeval van afwezigheid is het verplicht naam en gsm-nummer op het slot te plaatsen opdat de arbeider gecontacteerd kan worden.

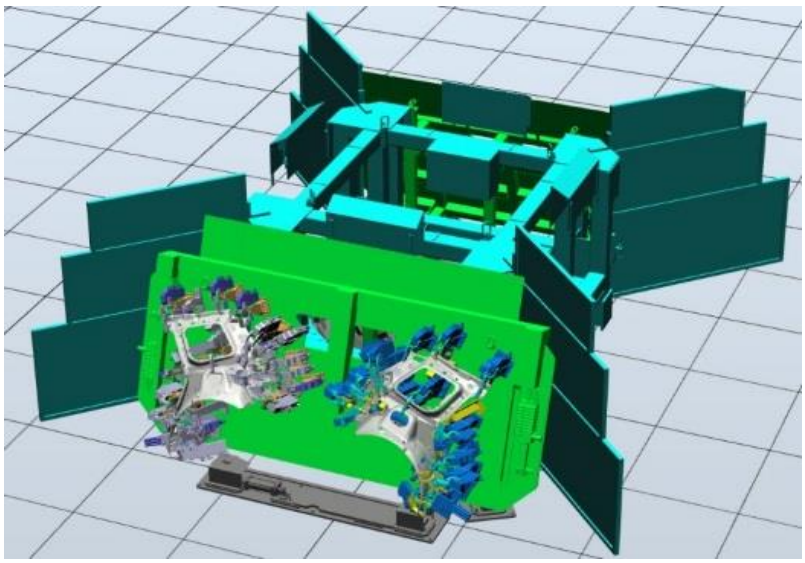
6.4.8 Containers en rekken

Dit is de plaats waar de enkele grijper elke onderdeel apart in het juiste rek gaat plaatsen. Het rek, zoals weergegeven in figuur 21, bestaat uit 2 containers. Deze zijn elk voorzien van twee rijen. Aan de rechtse kant worden alle rechtste auto-onderdelen geplaatst en in het linker rek, de linkse auto-onderdelen. Wanneer de rekken vol zijn, worden deze met een heftruck uitgehaald en verplaatst naar het volgende proces. Hierdoor kunnen opnieuw lege rekken geplaatst worden en zal de cyclus van het proces opnieuw beginnen.



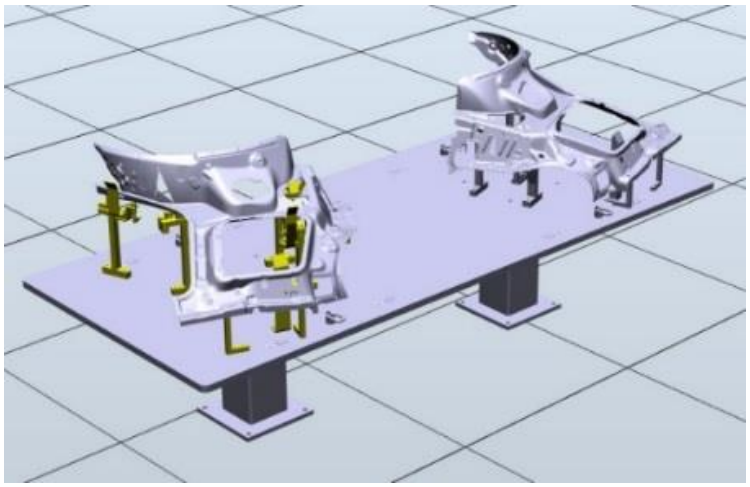
6.4.9 De tafels

De eerste tafel die aangetoond wordt, is de draaitafel (fig 22). Dit is het begin van het proces. Via de toegangspoort worden zowel de linkse en rechtse auto onderdelen op de draaitafel geplaatst, weg van de robots. Vervolgens draait de draaitafel 180°, waardoor de draaitafel in de juiste positie staat. Vervolgens kan het programma van de lasrobots stap voor stap doorlopen worden en worden de laspunten op het onderdeel aangebracht.



Figuur 22: Draaitafel met klemmen en auto-onderdeel

De 2de tafel die we gebruiken, is de aflegtafel. Deze tafel wordt gebruikt als tussenpositie omwille van het feit dat de robot met dubbele grijper niet tot aan de containers raakt. Ook is de aflegtafel noodzakelijk omdat elke auto-onderdeel apart in het rek geplaatst gaat worden. Onderstaande figuur (figuur 23) illustreert de aflegtafel.



Figuur 23: De aflegtafel met de auto-onderdelen

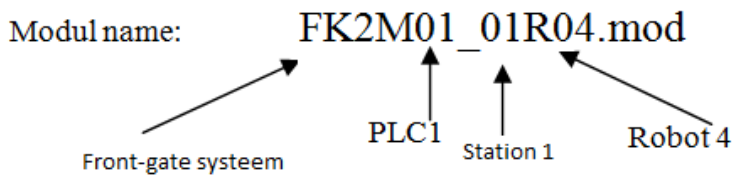
7. De BMW standard

Dit document dat beschikbaar is in het Engels en Duits, weergeeft de standaardprocedure voor het programmeren en maken van de logica en de bewegingen.

Het is de belangrijkste stap binnen het proces waardoor het noodzakelijk is om de standaard documenten door te nemen. Het beschrijft hoe BMW de programmatie wil. Hierdoor moet er rekening gehouden worden met de volgende hoofdstukken, opdat het proces en het programma voldoen aan de eisen van BMW.

7.1 De Modulenaam

De modulenaam is noodzakelijk om ervoor te zorgen dat elke robot zijn unieke naam toegewezen krijgt. We krijgen bijgevolg een duidelijk overzicht en alles is eenvoudig terug te vinden. Een modulenaam moet bestaan uit de celnaam, productiecel, het station nummer en robotnummer. Onderstaande figuur (figuur 24) toont een modulenaam.

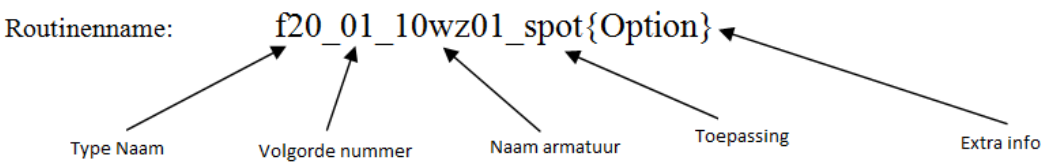


Figuur 24: De modulenaam

7.2 De Routinenaam

De routinenaam zoals weergegeven in figuur 25 bevat het typenaam, het volgnummer, de gereedschapsnaam en een afkorting voor de beschrijving van de activiteit.

- De typenaam moet als eerste weergegeven zijn. (F20, F30, enz.)
- Het volgnummer wordt weergegeven door: (01, 02, 03, ...)
- De armatuurnaam is de positie waar de robot werkt.
- De toepassing is de afkorting voor de beschrijving van de activiteit
- Indien vereist, kan er extra informatie weergegeven worden



Figuur 25: De routinenaam

7.3 De toolnaam

De toolnaam is de gereedschapsnaam die toegewezen wordt aan de grijpers en lastang die in het proces gebruikt worden. Deze namen zijn toegewezen aan een TCP (Tool Center Point) die weergegeven staan in een lijst of zelf opgemaakt kunnen worden indien deze niet in de lijst staan. Een voorbeeld van hoe een toolnaam is opgesteld, ziet u hier onder.

- t_cGun1 = Spot weld gun 1 (C-gun)
- t_xGun1 = Spot weld gun 1 (X-gun)
- t_Grp1 = gripper 1

7.4 Het workobject

De carrosserie onderdelen die in het proces aanwezig zijn, worden gedefinieerd als een werkobject. Via deze werkobjecten kunnen we targets aanmaken die de bewegingen van de robot zullen overlopen tijdens het proces. Op onderstaande figuur (figuur 26) is een werkobject gedefinieerd.

wobj_10fx01 Work object for station 10 fixture 01

Figuur 26: het werkobject

7.5 Toewijzing programmanummer

Deze nummers zijn toegewezen door de standaardprocedure en zijn niet beschikbaar voor de programmering door een gebruiker. De programma's zijn noodzakelijk in het proces en worden in de logica van de systemen weergeven. Ze zorgen ervoor dat volgende toepassingen uitgevoerd kunnen worden.

➔ Program_45 > Tip dressing

Dit programma wordt toegepast om de kappen op de lastangen af te frezen.

➔ Program_46 > Automatic tip change gun 1

Via dit programma wordt een automatische kappenwissel uitgevoerd.

➔ Program_51 > Manual tip changing - gun 1

Via dit programma wordt een manuele kappenwissel uitgevoerd.

➔ Program_56 > Maintenance program - gun / handling 1

Dit programma geeft het onderhoud van de lastang of grijper weer.

➔ Program_61 > Maintenance program-Tool changer head only

Dit programma zorgt voor het onderhoud van het gereedschap.

➔ Program_62 > Brake test

Dit programma dient om de robot naar de remtestpositie te bewegen. De punten van de beweging naar en vanuit deze positie moeten worden gemaakt door de programmeur. In de remtestpositie, worden alle remmen van de robot gecontroleerd, beginnend met as 1.

7.6 De home positie

De homepositie is de thuispositie van een robot. Het is de positie die als eerst wordt geprogrammeerd en waar alles bij begint. De robot zal eerst van hieruit uit gaan bewegen en zal als laatst op deze plaats terugkomen.

Bij het programmeren van de homeposities is het belangrijk rekening te houden met de verschillende robots. Zo beschikt de lasrobot slechts over 1 homepositie krijgen. Voor de robot met grijper kan men tot 5 homeposities hebben. Dit om ervoor te zorgen dat het hoofdprogramma MAIN vanuit iedere homepositie, na iedere bewerking, terug kan starten.

De weergave van de homepositie wordt in de logcia weergegeven door MoveAbsJ.

7.7 Move

De functie move is noodzakelijk voor de bewegingen in de logica aan te geven. Hierbij kan men de keuze maken om de robot of het gereedschap lineair of als joint te laten bewegen. De weergave van de schillende moves worden hieronder getoond.

➔ MoveJ (Joint)

➔ MoveL (Lineair)

➔ MoveJG (Joint Gun)

7.8 Job processing

Via job processing wordt een aanvraag gedaan om verschillende jobs te mogen starten. Voor de programmatie in het proces, maken we gebruik van job 1 en 25. Hierbij staat 1 voor het sluiten van de klemmen en 25 voor het niet draaien van de draaitafel. Zolang de job niet wordt vrijgegeven, zal de robot in deze instructie blijven.

De aanvragen van deze jobs worden aangegeven door Move(J/L)_JobRequest. Wanneer de job afgelopen is, geef je de PLC een teken via de instructie JobFinished. Deze aanvraag wordt in de logica aangegeven als Move(J/L)_JobFinished.

7.9 De Collision protection

De collision protection bestaat uit 62 systeem specifieke zones, waardoor de robot toegang kan vragen tot elke zone. Maar enkel zones kunnen vrijgegeven worden die de robot zelf bezit. Hierdoor zal een robot moeten wachten als een andere robot in zijn collision zit. Hoe dit in zijn werk gaat, wordt in volgende punten beschreven.

1. Robot vraagt PLC of een zone onbezet is,
2. Zone is onbezet,
3. Hierdoor wordt deze zone voorbehouden voor de verzoekende robot,
4. De robot ontvangt een inschakeling,
5. Wanner de robot de zone verlaat,
6. Tekent hij af,
7. Zone beschikbaar voor een andere robot.

De aanvragen van deze zones worden aangegeven door Move(J/L)_CollReq. Wanneer de zone vrijgeven wordt, geef je de PLC een teken via de instructie CollClr. Deze aanvraag wordt in de logica aangegeven als Move(J/L)_CollClr. Indien alles gereset moeten worden, maken we gebruik van de instructie CollClearAll.

8. De bewegingen en logica van het programma.

8.1 De bewegingen

De volgende stap in het proces is het programmeren van bewegingen. Deze dienen voor elke robot aangeleerd te worden. Om te beginnen starten we in de homepositie van de robot. Vervolgens gaan we voor de lasrobot de laspunten aanspringen, zodat we een verloop krijgen van hoe de robot tijdens het lassen gaat bewegen. Wanneer dit in orde is, worden tussenpunten aangeleerd zodat de robot de laspunten kan bereiken.

Tijdens het overlopen van de bewegingen wordt de configuratie van de robots bepaald, zodat de assen op de correcte manier naar volgende positie gaat draaien. Hierdoor wordt gekeken dat we met de robot geen helikoptereffect creëren. Het is dus noodzakelijk correcte en eenvoudige bewegingen te maken in een kort tijdsverloop.

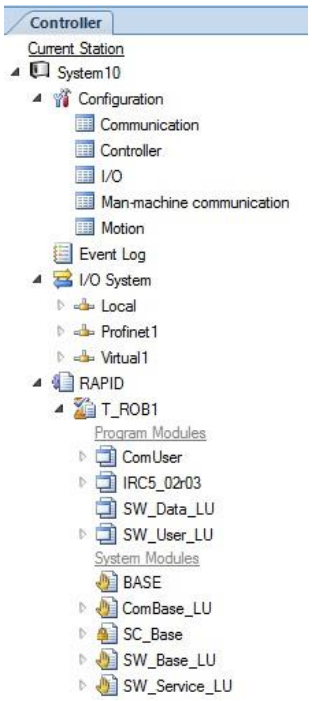
Na het aanleren van de bewegingen voor elke robot kunnen we overgaan naar het maken van de logica.

8.2 De logica

De logica zorgt ervoor dat we een soort van instructielijst opstellen van het hele proces dat de robot moet aflopen. De logica beschikt hierdoor over de bewegingen van de robots, parameters en modules zoals geschreven in de BMW standaard.

8.2.1 Het aanmaken van de robotsystemen

Om de logica van de robots op te stellen, dienen we robotsystemen te maken. Deze bevatten inhoudelijk de robot, 7de as en alle onderdelen of toestellen die tot een systeem gaan behoren. Doordat we beschikken over vier robots, dienen we ook vier systemen te maken. Wanneer dit systeem voor elke robot gemaakt is, kunnen we hierin de logica schrijven en parameters, modules gaan inladen. Onderstaande figuur (figuur 27) geeft een gemaakt robotsysteem weer.



Figuur 27: RobotSysteem

8.2.3 Inladen bewegingen, modules en parameters

De bewegingen die we aangeleerd hebben, zijn het belangrijkste en dienen ook in de logica weergegeven te worden. Hierdoor maken we gebruik van de functie ‘synchronisatie’. Via deze methode wordt alles van de bewegingen naar de logica gesynchroniseerd. De logica bevat de coördinaten van het laspunt, de positie van de assen, met welk gereedschap dit gebeurd en op welk werkobject de laspunten worden aangebracht.

Het tweede belangrijke punt voor de logica is het inladen van parameters, zodat ABB RobotStudio de logica kan lezen en kan gaan toepassen tijdens de simulatie. Deze worden gegeven door ABB. In onderstaande lijst worden de parameters opgesomd.

- Inladen van I/O
    - ➔ Staat voor Inputs /outputs en worden aan de PLC gelinkt via profinet.
    - ➔ I/O logica voor de grijpers
    - ➔ I/O logica voor de lastangen
- Inladen van de communication
- Inladen van de controller
- Inladen van de man-machine communication
- Inladen van de Motion

Vervolgens worden de modules ingeladen zodat de logica alles verstaat wat in het geschreven programma staat. De modules beschikken over verschillende programma's zoals in paragraaf 7.5 aangehaald. Hierdoor zal de logica opgesteld zijn volgens de BMW standaard. Doordat de robot met lastang en grijper van elkaar verschillen, dienen er ook andere modules ingeladen te worden zodat er geen fouten weergeven worden. De modules die nodig zijn voor het programmeren zijn weergeven in tabel 3.

Robot met X- en C-tang	**Robot met grijper**
Program Modules: ComUser SW_Data_LU SW_User_LU	Program Modules ComUser GrpData GrpUser TchUser_LU
Syteem Modules ComeBase_LU SC_Base SW_Base_LU SW_Base_LU	Program Modules ComUser GrpData GrpUser TchUser_LU

Tabel 3: Modules

8.2.4 Gemaakte programma's

Deze kunnen teruggevonden worden in bijlage 5 en 6. Het beschrijft de logica van de lasrobot en de robot met grijper. In deze bijlage wordt de logica + extra uitleg beschreven. Het geeft een beeld weer van hoe een logica eruit ziet en waaruit het bestaat.

8.3 De simulatie

Voor het controleren en bekijken van het gemaakt programma met de gewenste logica, maken we binnen het programma van ABB RobotStudio gebruik van een simulatie. Via deze methode zal heel het programma overlopen worden zoals besproken werd in hoofdstuk 5, de functie van de robotcel. Na het uittesten van simulatie is de programmatie klaar en kan alles in de werkelijke robotcel worden ingeladen. Sommige bewegingen en toepassingen dienen bijgestuurd te worden omdat er eventueel verschillen kunnen zijn tussen de theoretisch en praktische robotcel.

9. Praktische toepassingen

De bedoeling was om eveneens de gemaakte programma's in te laden in de werkelijke robotcel die opgebouwd wordt in de MINI fabriek in Oxford. Echter door omstandigheden wordt de robotcel met enkele weken vertraging opgebouwd. Dit valt buiten mijn stage en daardoor konden er geen praktische uitvoeringen gebeuren. Om toch kennis te krijgen in het praktisch gebeuren van een proces werd dit mij uitgelegd en aangeleerd in een andere robotcel. Hierdoor heb ik tijdens mijn stage de praktische kant van robotica aangeleerd en de productie van de MINI Cooper kunnen meevolgen.

9.1 De IRC5 – Flexpedant

Dit is de controller die we gaan gebruiken voor het bedienen van de ABB Robots. Wanneer we het programma dat gemaakt werd in ABB RobotStudio gaan linken aan de controller, kunnen we de gemaakte programma's en de programma's die voorgeschreven zijn door ABB actief gaan maken. We kunnen het proces praktisch overlopen, testen en effectief uitvoeren.

Het uitvoeren van de programma's kunnen we slechts gaan uitvoeren wanneer de controller aangezet wordt en de bedrijfsmodus gekozen wordt waarin we willen gaan werken. Op onderstaande figuur (Figuur 28) wordt de keuze van bedrijfsmodus weergeven.

Voor de modus hebben we 3 keuzes:

1. De automatische methode. Dit is de productiemodus.
2. De manuele modus. Deze is de programmeer en testmodus waarbij de dodemansknop actief is en de robotsnelheid beperkt is tot 250mm/s.
3. Manueel 100%. Dit is een optie waardoor een robotprogramma kan worden uitgevoerd zonder snelheidslimiet.

Figuur 28: Keuze bedrijfsmodus

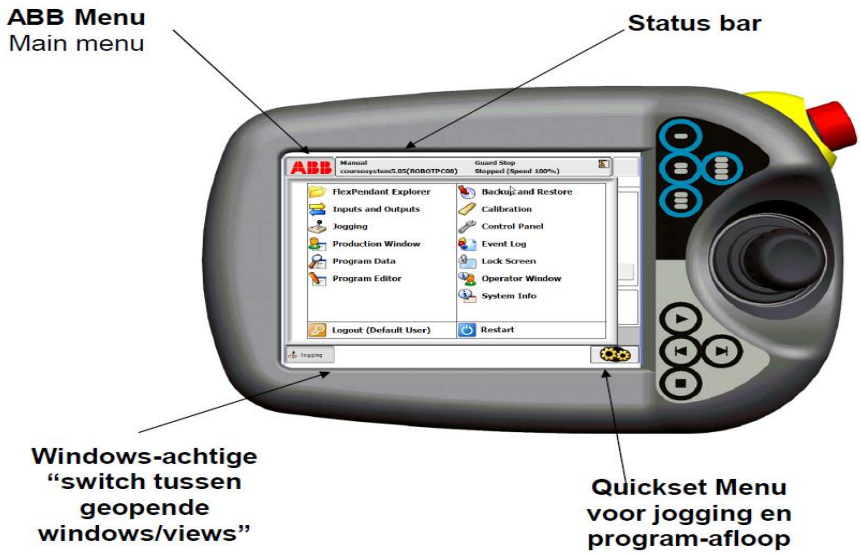
Na het inschakelen van de bedrijfsmodus kunnen we gebruik maken van de controller. De belangrijkste knoppen waardoor de controller bediend kan worden, is aangetoond in figuur 29.

De controller beschikt over een aanrakingscherm om de functie te bedienen zoals weergegeven in figuur 30. Via deze weg kunnen we alles gaan instellen en programma's gaan overlopen en bedienen. Wanneer we de functies of programma's willen uittesten, maken we gebruik van de programma-afloop toetsen. Hierbij kunnen we via 'play' het volledige proces aan één stuk door laten lopen. Wanneer dit een eerste keer gebeurt, worden de 'Backward/Forward' toets gebruikt zodat het proces stap per stap overlopen wordt.

Belangrijk bij het bedienen van de controller is dat de dodemansknop ingedrukt moet worden, anders zullen de motoren niet actief worden in de manuele bedrijfsmodus. De dodemansknop bestaat uit 3 posities, namelijk de middenpositie voor de werking, loslaten of volledig indrukken. Dit laatste zorgt voor het onmiddellijk stoppen van alle robotbewegingen.

Indien posities aangepast moeten worden, kunnen we gebruik maken van de joystick, die gelinkt is aan de assen en bewegingsrichting. Tot slot, ingeval van nood, beschikt de controller over een noodstop.

Figuur 29: Knoppen controller



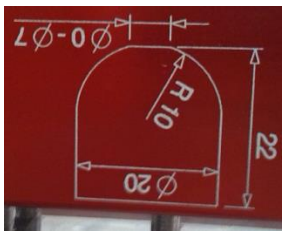
Figuur 30: Aanrakingsscherm met zijn functies

9.2 Tipdress en kappenwissel

Dit is een programma en belangrijk onderdeel dat voorgeschreven wordt door BMW. Bij het puntlassen van het auto-onderdeel gaan de kappen van de lastang na x aantal tijd bijgefreesd of vervangen worden. Waarom dit nodig is, wordt aangetoond in figuur 31 waar een nieuwe kap en een gebruikte kap weergegeven is.


Figuur 31: Nieuwe en gebruikte kap

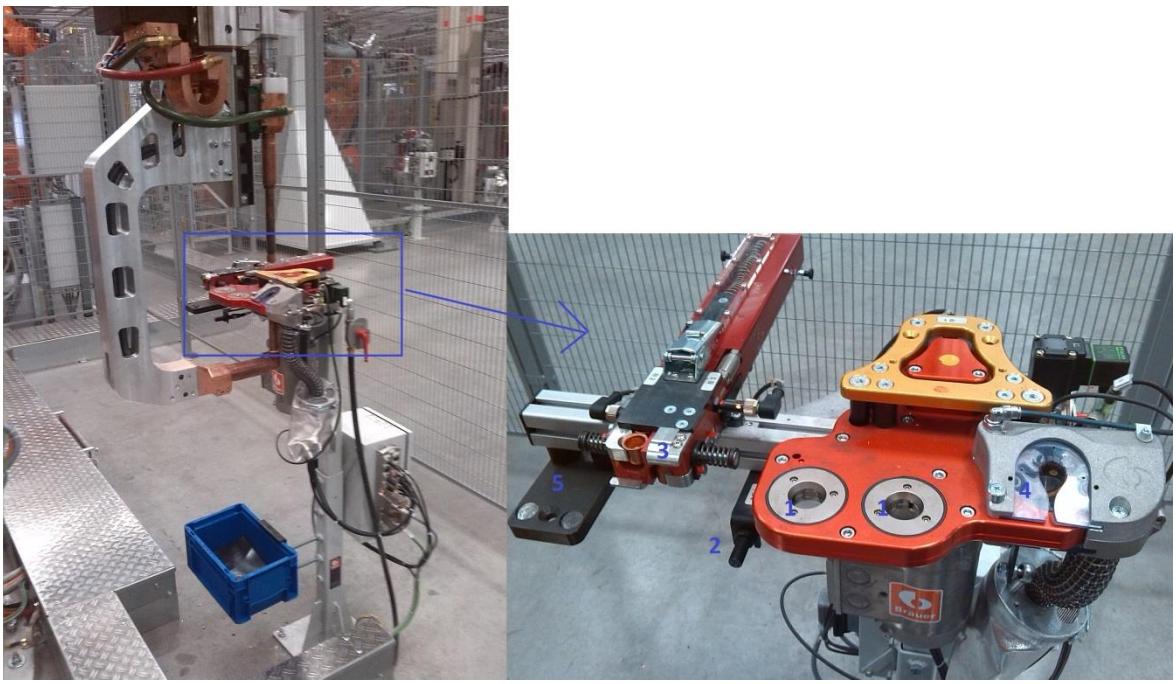
Doordat de laspunten zeer cruciaal zijn tijdens het proces is het noodzakelijk gebruik te maken van de juiste laskappen. Om dit duidelijk aan te halen worden de maten van de kap weergegeven zoals aangetoond wordt in figuur 32.



Figuur 32: Afmeting kap

Het toestel om de kappen van de lastangen af te frezen en te vervangen, is de tipdress en het kappenwisseltoestel (figuur 33). Dit toestel is zo ontworpen dat het automatisch de actie gaat uitvoeren.

De tang beweegt naar het toestel en de kappen worden vervangen, dit zowel voor de bovenste als de onderste tang (1). Vervolgens wordt via de sensor (2) gekeken of de kappen verwijderd zijn. Daarna gaat het programma verder richting (3). Op deze plaats wordt de kap van de bovenste en onderste tang geplaatst. Wanneer dit gebeurd is, zullen de kappen zesmaal afgefreesd worden, zodat ze de gewenste afmetingen en vormen krijgen (4). Tot slot worden de tangen tegen elkaar gedrukt om ervoor te zorgen dat de nieuwe kappen stevig vast staan (5).



Figuur 33: Tipdress en kappenwissel toestel

De logica waarbij dit toestel werkt, wordt beschreven in bijlage 7 waar de stappen overlopen worden van het programma en met enkele foto’s ter verduidelijking.

10. Implementatie

Het resultaat dat bereikt werd tijdens de stage is de 3D-opbouw en de programmatie van de robotcel zoals weergeven staat in bijlage 4. Dit resultaat toont het beeld van de werkelijke robotcel die in de MINI fabriek in Oxford opgebouwd zal worden. Hierdoor is mijn resultaat voorzien van een simulatie, die via video's opgenomen werd. Deze video's tonen de bewegingen en programmaties van de robots aan die afgelopen worden.

Voor het bereiken van dit resultaat via ABB RobotStudio ondervond ik enkele problemen. Gevolg hiervan: enkele afwijkingen in het tijdsschema. Deze problemen werden uiteindelijk opgelost waardoor het resultaat bereikt werd.

De problemen en de oplossingen worden in volgende punten weergeven:

Probleem 1: De juiste CAD-bestanden importeren via ABB RobotStudio voor de 3D-opbouw. Het probleem is dat sommige bestanden van slechte kwaliteit zijn of een te groot aantal KB bezitten. Daardoor is de kwaliteit van de onderdelen en toestellen slecht en loopt het programma ABB RobotStudio vast. Dit laatste wordt veroorzaakt door het te veel aantal KB.

Oplossing 1: De CAD-bestanden via JT-bestand als monolithic of per part opslaan, waardoor we via de CAD-converter van RobotStudio kunnen gaan converteren naar het ideale bestandstype. Hierdoor zal de beste keuze een RSGFX-bestand zijn.

Probleem 2: De bibliotheek van ABB beschikt niet over een dubbele 7de as.

Oplossing 2: De 7de as gaan spiegelen.

Probleem 3: De functie Mirror werkt niet op het toestel en hierdoor kunnen we de 7de as niet gaan spiegelen.

Oplossing 3: De 7de as gaan opslaan via de functie export geometry. Vervolgens zal dit bestand via een CAD-programma ingeladen worden. Hierbij kunnen we een spiegeling toepassen, waardoor achteraf een gespiegelde 7de as ingeladen kan worden en we beschikken over een dubbele 7de as.

Probleem 4: Grijper roteert rond het verkeerde nulpunt.

Oplossing 4: Om dit probleem op te lossen wordt een nieuwe frame gemaakt. Door deze gemaakte frame vast te zetten als UCS, zal de grijper rond het juiste nulpunt roteren.

Probleem 5: Robot verschuift niet mee met de gespiegelde 7de as van oplossing 3.

Oplossing 5: We maken gebruik van Edit System. Dit zorgt ervoor dat we de Mirror niet meer nodig hebben. Via Edit System kunnen we de 7de as + robot 180° laten draaien. Echter om de dubbele 7de as te verkrijgen, dienen we deze over elkaar te leggen.

Probleem 6: Robotsystemen in ABB 5.61 kunnen niet gemaakt worden met robots van het programma ABB RobotStudio 5.15.

Oplossing 6: De robots worden vervangen door robots van de 5.61.

Probleem 7: Probleem met het plaatsen en vastzetten van frames en targets op juiste laspunten van het onderdeel.

Oplossing 7: We plaatsen de frames op de laspunten via de gewenste coördinaten. Via attach to maken we de frames vast aan het onderdeel, waardoor het onderdeel met zijn frames op de juiste positie in de robotcel geplaatst kan worden. Hierna worden de targets geplaatst op coördinaten van de frames en leggen we onze laspunten vast.

Probleem 8: Het plaatsen van TCP op juiste plaats (C-tang, grijpers,...)

Oplossing 8: UCS plaatsen op de robotflens, waardoor we een frame maken met de TCP-coördinaten. Vervolgens wordt de TCP via de functie tooldata vastgelegd.

Probleem 9: Het plaatsen van onderdelen in de grijper naar de juiste positie op de aflegtafel.

Oplossing 9: We plaatsen een TCP op de flens van de grijperrobot. Het onderdeel dat op de aflegtafel ligt wordt aangeklikt en vastgelegd als UCS. Wanneer de UCS is vastgelegd, maken we een frame die als referentie UCS heeft. Hierna wordt de frame geconverteerd naar het werkobject. We maken nu een path aan en selecteren insert move instruction waarbij we het werkobject gaan vastzetten. Wanneer we nu via de functie jump to move instruction toepassen, zal de robot perfect de onderdelen in de grijper, op de gewenste positie, op de aflegtafel zetten.

Probleem 10: ABB RobotStudio kan de BMW standard niet lezen waardoor sommige bewegingen niet gesynchroniseerd worden naar het station.

Oplossing 10: Inladen van de juiste modules, I/O en het invullen van de path parameters.

Probleem 11: Systemen waarbij de 7de as niet meer tot het systeem behoort. Hierdoor beweegt robot niet meer mee met de 7de as.

Oplossing 11: Systeem opnieuw maken of de back-up inladen van het gemaakte systeem

Probleem 12: Tijdens het draaien van station verdwijnt het station volledig in een bepaalde positie.

Oplossing 12: Een ingeladen path zorgt voor dit probleem. We moeten dit path onzichtbaar maken waardoor het station 360° kan draaien zonder dat het van het scherm verdwijnt.

11. Conclusie

Uit mijn stage kan ik concluderen dat bij het ontwikkelen van de robotcel heel wat komt kijken. De verschillende robots en programma's die gebruikt worden om dit te verwezenlijken zijn heel interessant en hebben voor het resultaat van mijn stage gezorgd.

ABB RobotStudio, het programma dat gebruikt werd tijdens de stage, heeft zijn voor- en nadelen. Hierdoor is ABB RobotStudio op sommige vlakken een uitstekend programma, maar op andere vlakken is dit programma minder goed. Waarom dit programma niet uitstekend is, werd aangehaald in hoofdstuk 12, Implementatie.

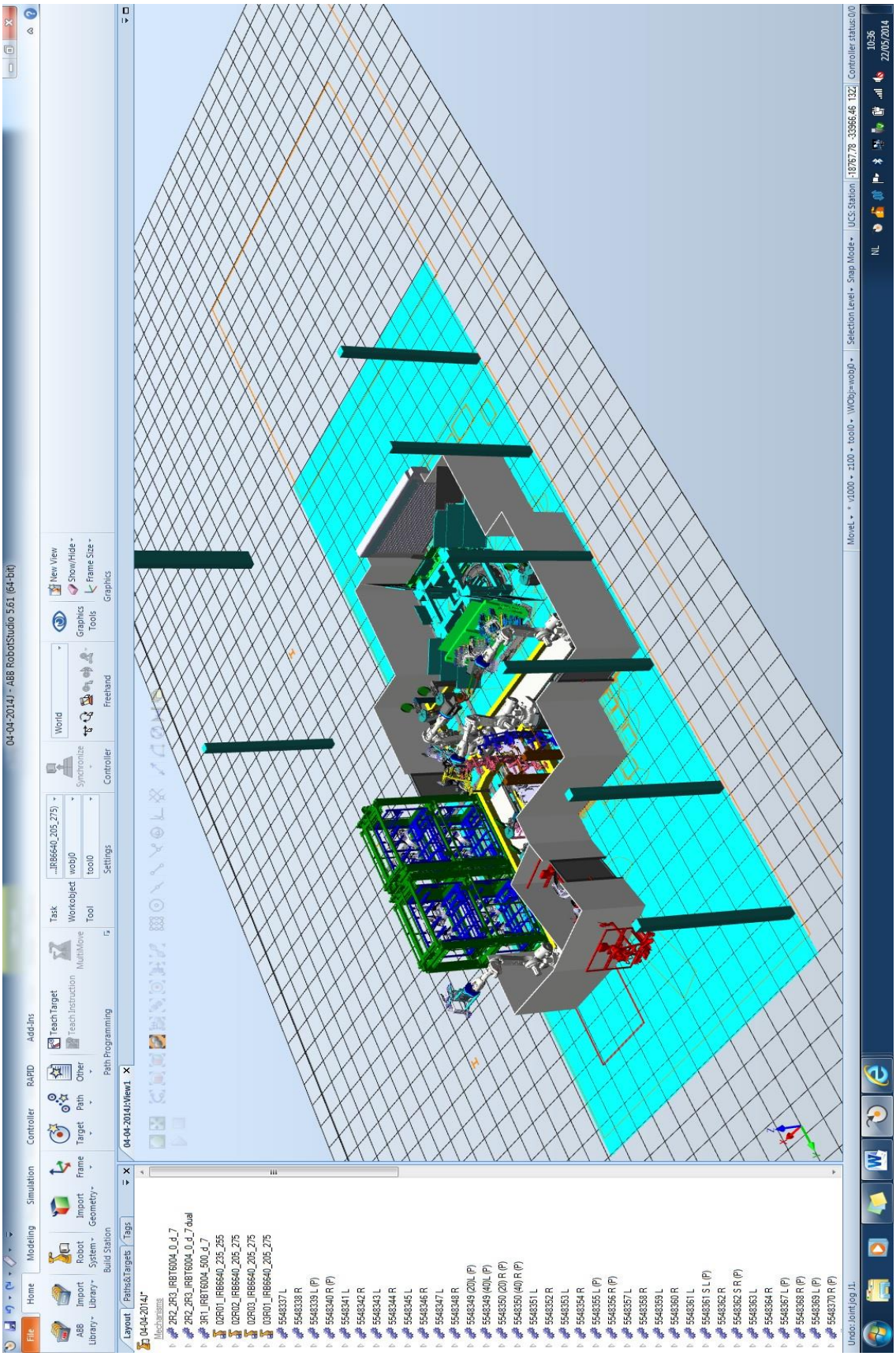
Voor bepaalde doeleinden zal Ciratec & C° in de toekomst meer gaan werken met het programma ABB RobotStudio voor het schrijven van de logica. Hierdoor is het programma geschikt voor de diensten die het bedrijf levert.

Mijn resultaat is klaar. Van de 3D-opbouw tot het programmeren en simuleren. Doordat de opbouw van de robotcel in Oxford met enkele weken uitgesteld is, kon mijn programma niet ingeladen worden tijdens mijn stage. Wanneer de opbouw klaar is, zal het programma in de werkelijke robotcel ingeladen worden. De robotcel wordt zo klaargemaakt om in productie te treden voor het maken van de nieuwe MINI Cooper.

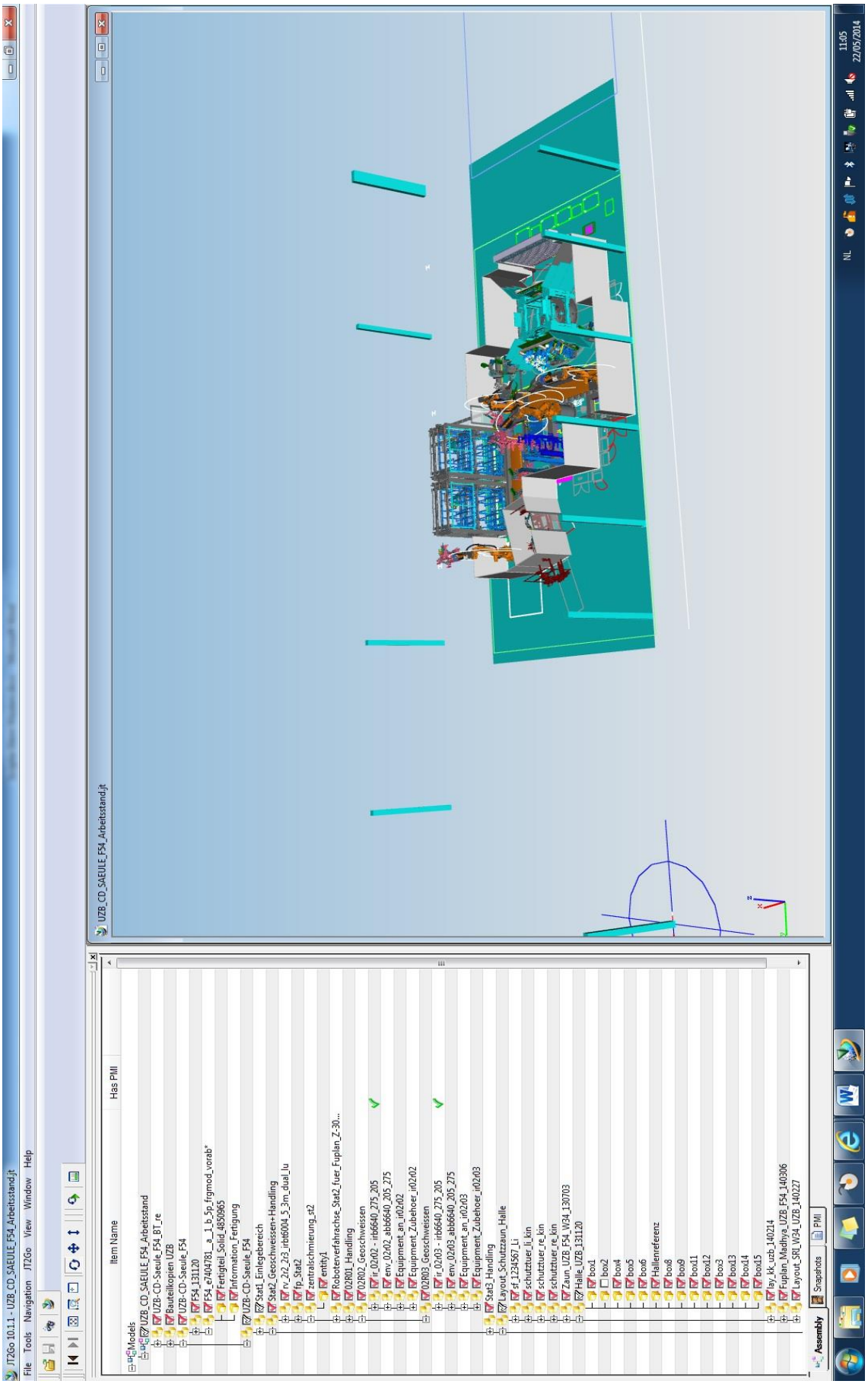
Door dit totaalpakket van doelstellingen is mijn kennis en interesse zeer groot geworden. Het was een zeer leerrijke stage en een mooie ervaring. De stap in de wereld van robots zal een uitstekende keuze zijn voor een werkzekere en interessante job.

12. Bijlagen

Bijlage 1: Programma ABB RobotStudio 5.61



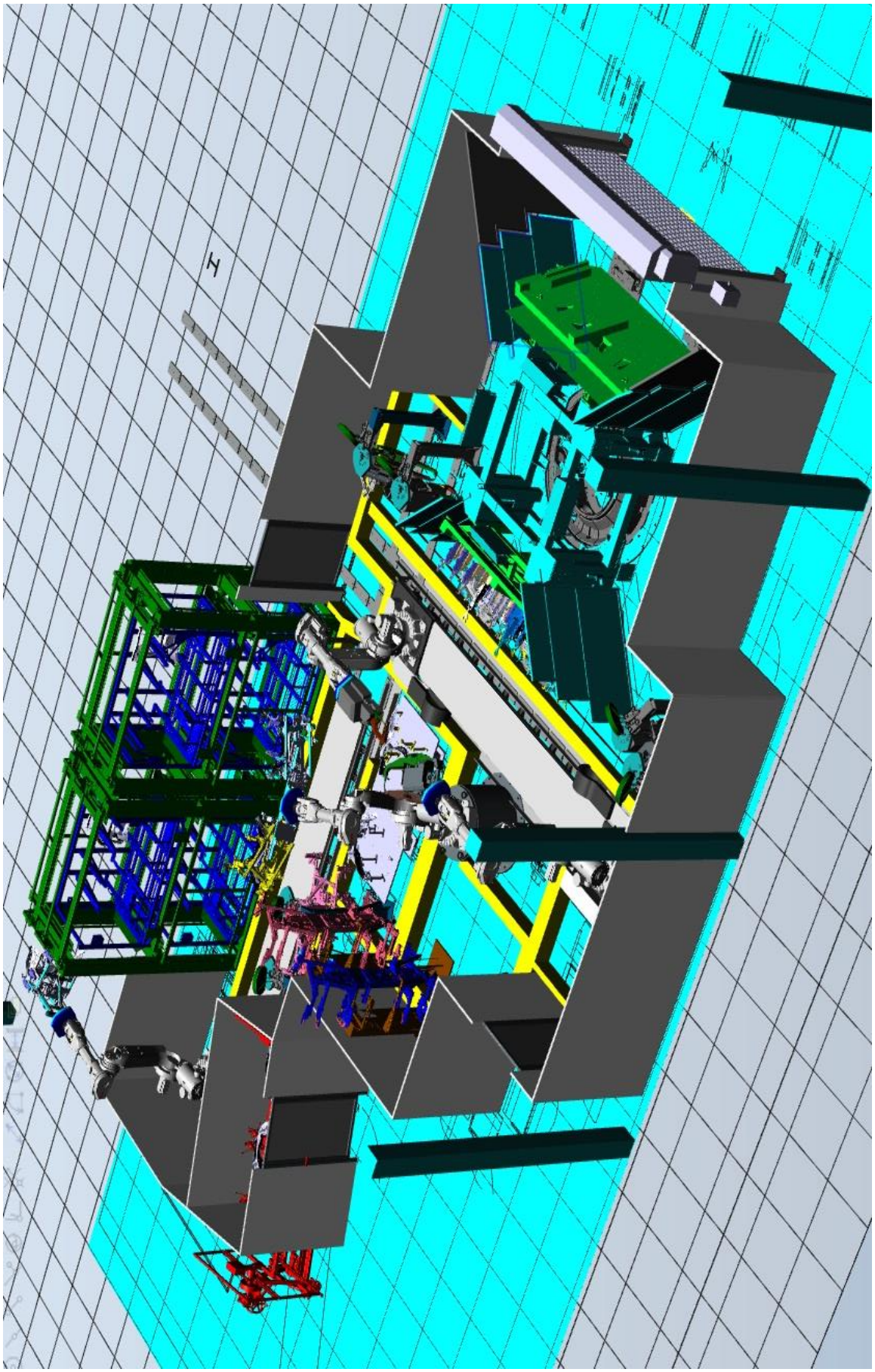
Bijlage 2: Programma JT2Go

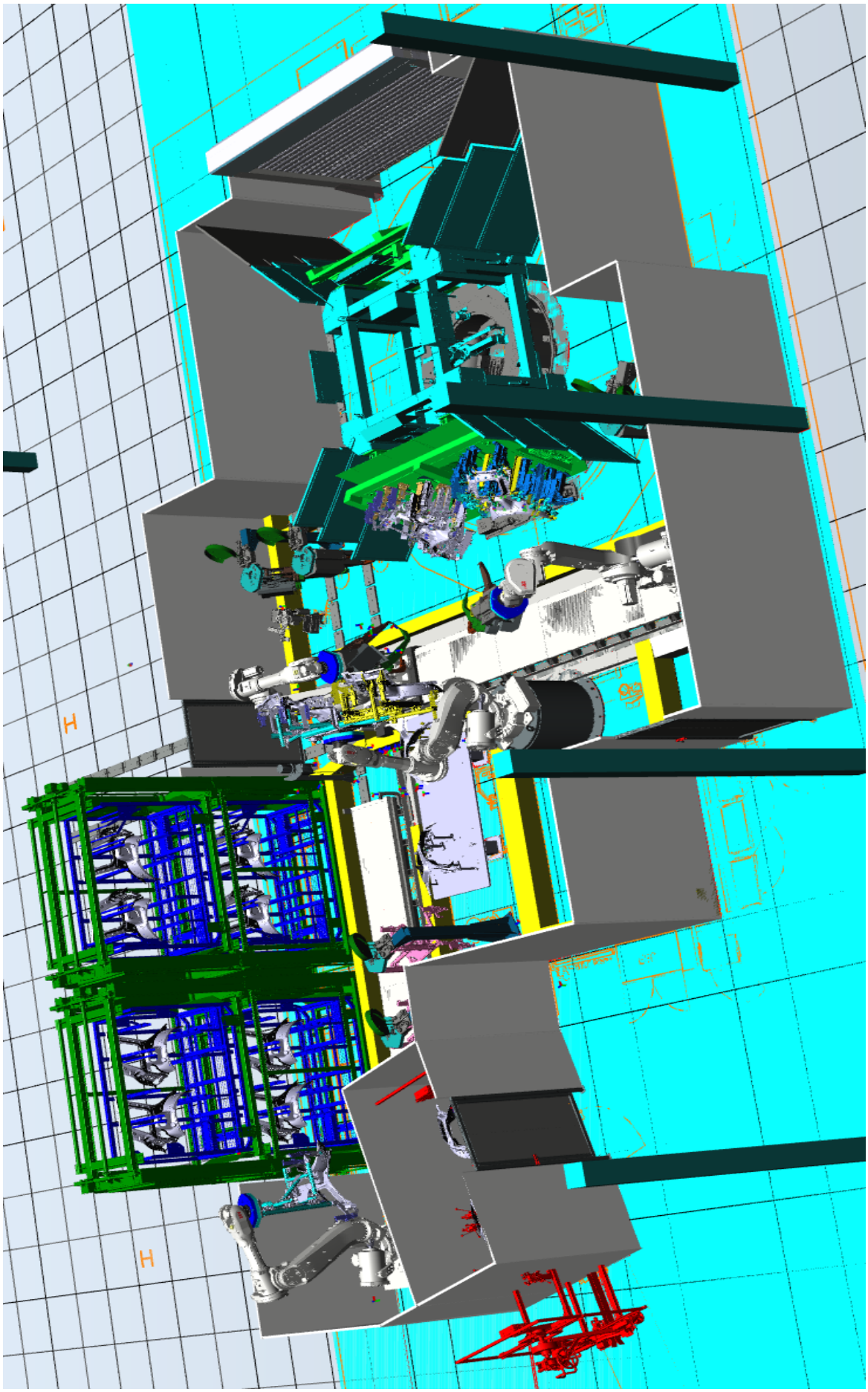


Bijlage 3: Programma Notepad++


```
MODULE IRC5_02r03
!
! Homeposition Wird vom User benutzt/Home positions is used by the user
PERS jointtarget jpHome1:=[[0,-37,47,-173,-78,163],[0,9E9,9E9,9E9,9E9,9E9]];
!#-------------------------------------------------------------------------------------------------------------------#!

!# ------------------------------
!# ------ ROBTARGET
!# ------------------------------
CONST robtarget lsp54089_F54:=[[2259.64,-726.51,304.43],[0.360282,-0.641127,0.603446,-0.308231],[0,-1,0,1],[100,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget lsp135401_F54:=[[2207.9,-678.05,231.67],[0.067523,-0.703876,-0.703875],[0,-2,-1,0],[100,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget mad_lsp021:=[[2246.95,-676.75,313.3],[0.221486,0.215079,-0.676876,0.668225],[0,-1,2,1],[600,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget lsp54093_F54:=[[2269.51,-795.21,297.54],[0.649389,0.389859,-0.249,-0.603576],[0,-1,2,1],[600,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget lsp175285_F54:=[[2273.54,-530.43,1018.5],[0.856295,0.413464,-0.11543,0.287197],[-1,-2,1,1],[1500,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget lsp175283_F54:=[[2405.61,-532.49,1016.06],[0.650764,0.358794,-0.300798,0.5977741],[-1,-2,1,1],[1500,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget lsp56275_F54:=[[2494.94,-674.95,654.81],[0.49579,-0.504174,0.504175],[0,-2,0,1],[1500,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget lsp142739_F54:=[[2559.94,-674.95,654.81],[0.429746,0.429746,-0.561532,0.561532],[-1,-2,1,1],[1500,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget lsp56279_F54_nm_BG49_1__O_:=[[2894.94,-674.25,654.81],[0.313714,0.313714,-0.633706,0.633706],[0,-2,3,1],[1500,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget lsp174675_F54:=[[2625.7,-770.85,510.36],[0.583583,0.343117,0.428262,-0.598576],[-1,-2,1,1],[1500,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget lsp174673_F54:=[[2698.51,-756.57,526.84],[0.714866,0.439416,0.328711,-0.433393],[0,-2,1,1],[1500,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget lsp55821_F54_nm_BG69:=[[2928.55,-693.36,539.79],[0.411395,0.347084,0.54346,-0.6441571],[-1,-2,1,1],[1500,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget lsp172159_F54:=[[2956.73,-718.13,394.79],[0.609618,0.514321,0.388961,-0.461031],[0,-2,1,1],[1850,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget lsp135379_F54:=[[3036.03,-552.92,655.15],[0.641115,0.765312,0.002457,0.057113],[0,-1,-1,1],[1850,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget lsp135381_F54:=[[2996.75,-558.42,655.4],[0.600479,0.793695,0.077218,0.059254],[0,-1,-1,1],[1850,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget lsp54097_F54:=[[2289.85,-852.97,283.29],[0.717683,0.574238,-0.002703,-0.393923],[0,-1,-1,0],[600,9E9,9E9,9E9,9E9,9E9]];

!# ------------------------------
!# ------ TOOL DATA
!# ------------------------------
PERS tooldata t_xGun1:=[TRUE,[[-144.94,540.92,1234.45],[0.703714,0.281096,0.366332,-0.539975]],[0.001,[0,0,0],[1,0,0,0],0,0,0]];
PERS tooldata t_cGun1:=[TRUE,[[-29.89,111.57,1096.98],[0.000001,0.793355,-0.608759,-0.0000002]],[0.001,[0,0,0],[1,0,0,0],0,0,0]];
!# ------------------------------
!# ------ WOBJ DATA
!# ------------------------------
PERS wobjdata wobj_01fx01_LI:=[FALSE,TRUE,"",[[42.1,1990.3,-851.136],[0.969846,-0.17101,-0.17101,0.03015931],[[0,0,0],[1,0,0,0]]];

PROC Program_1()

WaitUntil GetTypeCode()<>0 OR PLC_di_ProgSkip = 1;
!**********************************************************************
! Wait for Typbit Value or Program Skip  1=(F56) / 2=(F57)       *
!**********************************************************************

IF PLC_di_ProgSkip = 1 RETURN; ! Program Cancel

!**********************
!* Store Typbit Code *
!**********************
n_Type_Mem:=GetTypeCode();
```

Bijlage 4: De robotcel





Bijlage 5: Logica lasrobot met X- en C-tang

```
MODULE IRC5_02r02 (De module naam)
!
! Homeposition wird vom User benutzt/Home positions is used by the user
PERS jointtarget jpHome1:=[[0,-30,40,180,-75,-11],[0,9E9,9E9,9E9,9E9,9E9]];
(De homepositie)              (stand van de assen)  (Positie 7de as)
!#----------------------------------------------------------------------------------------------------------------------------------------------------------#!

!# ---------------------------------------------
!# ------ ROBTARGET
!# ---------------------------------
CONST robtarget lsp54090_F54:=[[2259.71,726.57,304.51],[0.354846985,0.651560973,0.592165976,0.314469987],
[-1,0,1,1],[870,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget lsp135402_F54:=[[2207.96,678.1,231.76],[0.038651,0.038652,-0.70605,0.70605],
[-1,-1,0,1],[50,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget lsp135382_F54:=[[2996.81,558.47,655.48],[0.600973375,-0.792990495,0.084142053,-0.054011034],
[-1,-4,1,1],[1650,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget lsp135380_F54:=[[3036.1,552.98,655.24],[0.642717772,-0.764760728,0.02916499,-0.034702988],
[-1,-4,1,1],[1650,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget lsp172160_F54:=[[2956.79,718.19,394.88],[0.538482047,-0.45430504,0.45762804,0.542421047],
[-1,-4,0,1],[1650,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget lsp55822_F54_nm_mit_BG70:=[[2928.61,693.42,539.88],
[0.393653949,-0.332116957,0.552733928,0.655148915], [-1,-3,0,1],[1300,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget mad_lsp004:=[[2732.65,777.27,501.64],[0.663819944,-0.381805968,0.372754968,0.524042955],
[-1,-3,0,1],[1300,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget mad_lsp002:=[[2591.88,793.58,498.61],[0.4319289,-0.222863948,0.484823888,0.727127832],
[-1,-3,-1,1],[1300,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget lsp56280_F54_nm_mit_BG50_i__O_:=[[2895,674.3,654.89],
[0.301492194,-0.301491194,-0.639611412,-0.639611412],[-1,-4,-2,1],[1300,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget lsp142740_F54:=[[2560,675,654.89],[0.458481133,-0.458481133,-0.538326156,-0.538326156],
[-1,-2,-4,0],[1300,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget lsp56276_F54:=[[2495,675,654.89],[0.545775855,-0.545774855,-0.44958788,-0.44958788],
[-1,-2,-4,0],[1300,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget lsp175284_F54:=[[2405.68,532.55,1016.14],[0.629508843,-0.348075913,-0.313131922,-0.620088845],
[-1,-1,-1,0],[1300,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget lsp175286_F54:=[[2273.61,530.48,1018.6],[0.858770121,-0.414453058,-0.111815016,-0.279714039],
[-1,-1,-2,0],[1300,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget lsp54098_F54:=[[2289.91,853.02,283.37],[0.717681042,-0.574239034,-0.002705,0.393924023],
[-1,-2,-1,0],[400,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget lsp54094_F54:=[[2269.58,795.27,297.62],[0.632869378,-0.396433237,-0.238392142,0.620875371],
[-1,-3,0,0],[400,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget mad_lsp022:=[[2247.01,676.8,313.38],[0.217442928,-0.21098393,-0.678163777,-0.669550779],
[-1,-2,-3,0],[400,9E9,9E9,9E9,9E9,9E9]];
Declaratie van de laspunten , Coördinaten van het laspunt , positie van de assen

!# ---------------------------------------------
!# ------ TOOL DATA → Data gereedschap
!# ---------------------------------------------
PERS tooldata t_xGun:=[TRUE,[[-144.96,540.91,1234.45],
[0.703714,0.281096,0.366332,-0.539975]],[0.001,[0,0,0],[1,0,0,0],0,0,0]]; → X-tang met coördinaten
CONST tooldata t_cGun1:=[TRUE,[[-29.91,111.55,1096.98],
[-0.000002,0.793355,-0.608759,-0.000002]],[0.001,[0,0,0],[1,0,0,0],0,0,0]]; → C-tang met coördinaten

!# ---------------------------------------------
!# ------ WOBJ DATA → Date werkobject
!# ---------------------------------------------
PERS wobjdata wobj_01fx01_RE:=[FALSE,TRUE,"",[[42.1,-1990.3,-851.136],
[0.969865832,0.171271053,-0.170624978,-0.030226011]],[[0,0,0],[1,0,0,0]]];
```

```
PROC Program_1() → PLC programma

WaitUntil GetTypeCode()<>0 OR PLC_di_ProgSkip = 1;
!*********************************************************************
! Wait for Typbit Value or Program Skip  1=(F56) / 2=(F57)         *
!*********************************************************************

IF PLC_di_ProgSkip = 1 RETURN; ! Program Cancel

!*********************
!* Store Typbit Code *
!*********************
n_Type_Mem:=GetTypeCode();

!****************************
!* Echo TypeBit Code to PLC *
!****************************
SetTypeCode(n_Type_Mem);

TEST n_Type_Mem

CASE 1:
        !=====================================================
        !               TYPE  F54
        !=====================================================
        !=====================================================
        ! JOB 1 Weld F54 in Station 01fx01
        !=====================================================

        WaitUntil PLC_di_Job01_Enable = 1 XOR PLC_di_ProgSkip = 1;
        !*****************************************
        !* Wait for Job1 or Program Skip         *
        !*****************************************

        IF PLC_di_ProgSkip = 1 RETURN; ! Program Cancel

        !Weld F54 in 01fx01
        f54_01_01fx01_RE_spot;

        WaitUntil PLC_di_Job01_Enable = 0;
        !*****************************************
        !* Wait for Job1  = 0                *
        !*****************************************

DEFAULT:
        WHILE TRUE DO
                Stop;
                !*************************
                !* Incorrect Job Enable   *
                !*************************
        ENDWHILE
ENDTEST
RETURN;
ENDPROC !(Program_1)

!#-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------#!
```

```
PROC Program_62()

   !SafeMove Bremsentest
IF DOutput(SM1_do_BrakeOk) = 0       VelSet 100, 150;

SM1_BrakeTest;

RETURN;
ENDPROC !(Program_62)

!#----------------------------------------------------------------------------------------------------------------------------------------------------------#!
PROC f54_01_01fx01_RE_spot() → Routine naam, begin van gemaakt programma voor de lasrobot

MoveAbsJ jpHome1,v5000,fine,t_xGun;
Vertrek homepositie , Snelheid en beweging, gebruikt gereedschap
MoveJ_JobRequest 25, [[896.81445937,1800.58060986,486.658709027],
[0.287578503,-0.707362038,-0.632067476,-0.132016123],[0,0,-1,1],[0,9E9,9E9,9E9,9E9,9E9]],
v5000, fine, t_xGun\WObj:=wobj_01fx01_RE;
MoveJ_JobRequest 1, [[896.81445937,1800.58060986,486.658709027],
[0.287578503,-0.707362038,-0.632067476,-0.132016123],[0,0,-1,1],[0,9E9,9E9,9E9,9E9,9E9]],
v5000, fine, t_xGun\WObj:=wobj_01fx01_RE;

        → Wordt aan de PLC een aanvraag gedaan om job 1 en 25 te mogen starten.

!Col Req 20 with robot 02R01;
!Col Req 21 with robot 02R03;
MoveJ_CollReq 20\Coll_B:=21, [[896.81445937,1800.58060986,486.658709027],
[0.287578503,-0.707362038,-0.632067476,-0.132016123],[0,0,-1,1],[0,9E9,9E9,9E9,9E9,9E9]],
 v5000, fine, t_xGun\WObj:=wobj_01fx01_RE\Coll_Desc:="02R01"\Coll_Desc_B:="02R03";
        → Collision request, er een collision  aangevraagd met een andere robot

MoveJG [[2179.953877028,938.713757796,251.18251592],[0.354847425,0.651560904,0.592165445,0.314470633],
[-1,-2,-1,0],[870,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_xGun\Wobj:=wobj_01fx01_RE;

        → Beweging robot, coördinaten, positie assen, snelheid, gereedschap dat gebruikt wordt ,werkobject waar
          gewerkt wordt

SpotL lsp54090_F54,Gun1,Id_02,100,v5000,t_xGun\Wobj:=wobj_01fx01_RE;

        → Weergave laspunt

MoveJG [[2179.953660976,938.713690579,251.182892648],[0.354847377,0.651560957,0.592165562,0.314470356],
[-1,-2,-1,0],[870,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_xGun\Wobj:=wobj_01fx01_RE;
MoveJG [[2847.876604206,2286.827200149,1463.340929654],[0.052908746,-0.0284127,-0.617474912,0.784294661],
[0,-2,-3,0],[253.470987082,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_xGun\Wobj:=wobj_01fx01_RE;
MoveJG [[1838.67105339,685.032431789,261.10737563],[0.251789054,0.09278978,-0.647516356,0.713242524],
[-1,-1,-1,1],[50,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_xGun\Wobj:=wobj_01fx01_RE;

SpotL lsp135402_F54,Gun1,Id_02,100,v5000,t_xGun\Wobj:=wobj_01fx01_RE;

MoveJG [[1838.67105339,685.032431789,261.10737563],[0.251789054,0.09278978,-0.647516356,0.713242524],
[-1,-1,0,1],[50,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_xGun\Wobj:=wobj_01fx01_RE;
MoveJG [[2847.876604206,2286.827200149,1463.340929654],[0.052908746,-0.0284127,-0.617474912,0.784294661],
[-1,-1,-1,1],[253.470987082,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_xGun\Wobj:=wobj_01fx01_RE;
MoveJG [[896.81445937,1800.58060986,486.658709027],[0.287578503,-0.707362038,-0.632067476,-0.132016123],
[0,0,-1,1],[0,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_xGun\Wobj:=wobj_01fx01_RE;

MoveJG [[3186.805368323,1218.46651197,665.47874467],[0.601069057,-0.792927702,0.08408877,-0.053951154],
[-1,-4,1,1],[1650,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;
MoveJG [[3186.808969241,558.466802423,665.478280365],[0.601055791,-0.79293645,0.084095756,-0.053959489],
[-1,-4,1,1],[1650,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;
```

```
SpotL lsp135382_F54,Gun1,Id_02,100,v5000,t_cGun1\Wobj:=wobj_01fx01_RE;

MoveJG [[3186.808969241,558.466802423,665.478280365],[0.601055791,-0.79293645,0.084095756,-0.053959489],
[-1,-4,1,1],[1650,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;

SpotL lsp135380_F54,Gun1,Id_02,100,v5000,t_cGun1\Wobj:=wobj_01fx01_RE;

MoveJG [[3186.808969241,558.466802423,665.478280365],[0.601055791,-0.79293645,0.084095756,-0.053959489],
[-1,-4,1,1],[1650,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;
MoveJG [[3186.805368323,1218.46651197,665.47874467],[0.601069057,-0.792927702,0.08408877,-0.053951154],
[-1,-4,1,1],[1650,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;
MoveJG [[2956.79169801,1048.190510486,224.878748789],[0.538489545,-0.454295878,0.4576292,0.542420299],
[-1,-3,1,1],[1650,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;
MoveJG [[2956.791716252,718.190127721,224.879542807],[0.53848695,-0.454299012,0.457628758,0.542420623],
[-1,-3,0,1],[1650,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;

SpotL lsp172160_F54,Gun1,Id_02,100,v5000,t_cGun1\Wobj:=wobj_01fx01_RE;

MoveJG [[2956.791716252,718.190127721,224.879542807],[0.53848695,-0.454299012,0.457628758,0.542420623],
[-1,-4,0,1],[1650,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;
MoveJG [[2786.626873105,637.535589617,148.832344571],[0.393739107,-0.332017455,0.552772136,0.655115937],
[-1,-4,0,1],[1300,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;

SpotL lsp55822_F54_nm_mit_BG70,Gun1,Id_02,100,v5000,t_cGun1\Wobj:=wobj_01fx01_RE;

MoveJG [[2786.626873105,637.535589617,148.832344571],[0.393739107,-0.332017455,0.552772136,0.655115937],
[-1,-3,0,1],[1300,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;
MoveJG [[2732.651602384,777.27180859,231.641893286],[0.663847325,-0.381762472,0.372754897,0.524040009],
[-1,-4,0,1],[1300,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;

SpotL mad_lsp004,Gun1,Id_02,100,v5000,t_cGun1\Wobj:=wobj_01fx01_RE;

MoveJG [[2732.651602384,777.27180859,231.641893286],[0.663847325,-0.381762472,0.372754897,0.524040009],
[-1,-3,0,1],[1300,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;
MoveJG [[2591.881106598,793.58243956,288.608778265],[0.431965321,-0.222801309,0.484849047,0.727108617],
[-1,-3,0,1],[1300,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;

SpotL mad_lsp002,Gun1,Id_02,100,v5000,t_cGun1\Wobj:=wobj_01fx01_RE;

MoveJG [[2591.881106598,793.58243956,288.608778265],[0.431965321,-0.222801309,0.484849047,0.727108617],
[-1,-3,-1,1],[1300,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;
MoveJG [[2591.879515155,1133.580374609,288.609756393],[0.431985779,-0.222769898,0.484862355,0.727097212],
[-1,-3,-1,1],[1300,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;
MoveJG [[2708.708786736,1138.979964998,617.621843517],[0.35628952,-0.450482736,-0.695081536,-0.432417322],
[-1,-3,-1,1],[1300,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;
MoveJG [[2743.914753005,856.461813944,714.420525375],[0.356290636,-0.450482752,-0.695076907,-0.432423826],
[-1,-4,-1,1],[1300,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;
MoveJG [[2762.940818737,703.779919489,766.734049656],[0.356291191,-0.45048272,-0.695072618,-0.432430298],
[-1,-4,-1,1],[1300,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;
MoveJG [[2806.593267921,662.999474426,631.842721015],[0.35629308,-0.450482616,-0.695051171,-0.432463321],
[-1,-4,-1,1],[1300,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;
MoveJG [[2836.278163989,671.981898003,645.676426723],[0.288839803,-0.343399396,-0.753726701,-0.48015048],
[-1,-4,-1,1],[1300,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;
MoveJG [[2836.275605849,671.980778097,645.674773946],[0.362051003,-0.377177848,-0.603531011,-0.602001878],
[-1,-3,-1,1],[1300,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;

SpotL lsp56280_F54_nm_mit_BG50_i__O_,Gun1,Id_02,100,v5000,t_cGun1\Wobj:=wobj_01fx01_RE;

MoveJG [[2836.276112426,671.980707301,645.67524341],[0.362051327,-0.377178662,-0.603533571,-0.601998607],
[-1,-4,-2,1],[1300,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;
MoveJG [[2836.278084273,671.981539026,645.676223797],[0.288839185,-0.343399742,-0.753728791,-0.480147324],
[-1,-4,-2,1],[1300,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;
MoveJG [[2806.593627501,663.000123391,631.841425408],[0.356293412,-0.450482825,-0.695053104,-0.432459723],
```

```
[-1,-4,-1,1],[1300,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;
MoveJG [[2762.94110163,703.779231231,766.734128217],[0.356290775,-0.450482584,-0.695074931,-0.432427064],
[-1,-4,-1,1],[1300,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;
MoveJG [[2743.914558885,856.46108808,714.421296848],[0.356290269,-0.450483112,-0.695078937,-0.432420492],
[-1,-4,-1,1],[1300,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;

MoveJG [[2538.351134713,874.312942698,671.403703582],[0.395581888,-0.500463479,-0.666665764,-0.385497127],
[-1,-4,-1,1],[1300,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;
MoveJG [[2564.785354096,662.193090958,744.083834909],[0.395581755,-0.500464017,-0.666659611,-0.385507207],
[-1,-2,-3,0],[1300,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;

SpotL lsp142740_F54,Gun1,Id_02,100,v5000,t_cGun1\Wobj:=wobj_01fx01_RE;

MoveJG [[2564.78502828,662.193069009,744.083781298],[0.39558189,-0.500463718,-0.66666168,-0.385503878],
[-1,-2,-4,0],[1300,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;

SpotL lsp56276_F54,Gun1,Id_02,100,v5000,t_cGun1\Wobj:=wobj_01fx01_RE;

MoveJG [[2564.78502828,662.193069009,744.083781298],[0.39558189,-0.500463718,-0.66666168,-0.385503878],
[-1,-2,-4,0],[1300,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;
MoveJG [[2538.350987872,874.313126686,671.402884],[0.395582141,-0.500463433,-0.666667648,-0.38549367],
[-1,-2,-3,0],[1300,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;
MoveJG [[2555.291502022,1054.300161159,1428.425556814],[0.525560493,-0.491275581,-0.543787134,-0.432122697],
[-1,-2,-3,0],[1299.999952316,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;
MoveJG [[2603.692720415,665.894606764,1561.506019684],[0.60559142,-0.33731421,-0.28497098,-0.662019408],
[-1,-2,-3,0],[1300,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;
MoveJG [[2414.282762384,463.534036611,1039.786237448],[0.605588477,-0.337319494,-0.284965203,-0.662021895],
[-1,-1,-1,0],[1300,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;

SpotL lsp175284_F54,Gun1,Id_02,100,v5000,t_cGun1\Wobj:=wobj_01fx01_RE;

MoveJG [[2414.281437256,463.532499766,1039.788447006],[0.60559142,-0.33731421,-0.28497098,-0.662019408],
[-1,-1,-1,0],[1300,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;

SpotL lsp175286_F54,Gun1,Id_02,100,v5000,t_cGun1\Wobj:=wobj_01fx01_RE;

MoveJG [[2414.281437256,463.532499766,1039.788447006],[0.60559142,-0.33731421,-0.28497098,-0.662019408],
[-1,-1,-2,0],[1300,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;
MoveJG [[2603.692101081,665.894949114,1561.507244111],[0.605590721,-0.337320461,-0.28497518,-0.662015054],
[-1,-1,-1,0],[1300,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;
MoveJG [[2555.291822635,1054.300503694,1428.42643992],[0.52556478,-0.491270994,-0.543790905,-0.432117951],
[-1,-1,-1,0],[1300,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;
MoveJG [[2307.929071653,1082.245863657,187.916431284],[0.703683331,-0.545455408,-0.031371696,0.454229],
[-1,-2,-1,0],[400,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;
MoveJG [[2329.934795166,905.661460729,248.420599787],[0.70368143,-0.54545894,-0.031371414,0.454227722],
[-1,-2,-1,0],[400,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;

SpotL lsp54098_F54,Gun1,Id_02,100,v5000,t_cGun1\Wobj:=wobj_01fx01_RE;

MoveJG [[2329.934992068,905.661796184,248.420102126],[0.703683109,-0.54545557,-0.031371761,0.454229145],
[-1,-2,-1,0],[400,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;

SpotL lsp54094_F54,Gun1,Id_02,100,v5000,t_cGun1\Wobj:=wobj_01fx01_RE;

MoveJG [[2329.934992068,905.661796184,248.420102126],[0.703683109,-0.54545557,-0.031371761,0.454229145],
[-1,-3,0,0],[400,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;
MoveJG [[2307.931347037,1082.247712506,187.914876428],[0.703684734,-0.545451623,-0.031372143,0.45423134],
[-1,-2,-1,0],[400,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;
MoveJG [[1783.919099141,1292.019972109,242.880034258],[0.217438954,-0.210986205,-0.678172308,-0.669542712],
[-1,-2,-1,0],[400,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;
MoveJG [[1860.595068637,676.724528747,453.699340719],[0.217440761,-0.210985258,-0.678169465,-0.669545304],
[0,-1,-3,0],[400,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;
```

```
SpotL mad_lsp022,Gun1,Id_02,100,v5000,t_cGun1\Wobj:=wobj_01fx01_RE;

MoveJG [[1860.595068637,676.724528747,453.699340719],[0.217440761,-0.210985258,-0.678169465,-0.669545304],
[-1,-2,-3,0],[400,9E9,9E9,9E9,9E9,9E9]],Gun1,100\GunZone:=50,v5000,z200,t_cGun1\Wobj:=wobj_01fx01_RE;

MoveJ_JobFinished 1, [[1860.595068637,676.724528747,453.699340719],
[0.217440761,-0.210985258,-0.678169465,-0.669545304],[-1,-2,-3,0],[400,9E9,9E9,9E9,9E9,9E9]],
v5000, z200, t_xGun\WObj:=wobj_01fx01_RE;
```

➔ Met de instructie JobFinished geef je de PLC een tekenen dat de job afgelopen is.

```
! Col Clr 20 robot 02R01;
! Col Clr 21 robot 02R03;
MoveJ_CollClr 20\Coll_B:=21, [[896.81445937,1800.58060986,486.658709027],
[0.287578503,-0.707362038,-0.632067476,-0.132016123],[0,0,-1,1],[0,9E9,9E9,9E9,9E9,9E9]],
v5000, fine, t_xGun\WObj:=wobj_01fx01_RE;
```

➔ Met de instructie Collision clear wordt de collision zone vrijgegeven.

```
MoveJ_JobFinished 25, [[896.81445937,1800.58060986,486.658709027],
[0.287578503,-0.707362038,-0.632067476,-0.132016123],[0,0,-1,1],[0,9E9,9E9,9E9,9E9,9E9]],
v5000, fine, t_xGun\WObj:=wobj_01fx01_RE;

MoveAbsJ jpHome1,v5000,fine,t_xGun;
RETURN;
ENDPROC → einde programma
ENDMODULE → einde module
```

Bijlage 6: Logica robot met grijper

```
MODULE IRC5_02r01
!
! Homeposition wird vom User benutzt/Home positions is used by the user
PERS jointtarget jpHome1:=[[-118,-55.97,-24.91,0,76.67,165],[9E9,9E9,9E9,9E9,9E9,9E9]];
PERS jointtarget jpHome2:=[[-4.22,-55.97,-15.65,0,81.21,165],[9E9,9E9,9E9,9E9,9E9,9E9]];

!# ---------------------------------------------
!# ------ ROBTARGET
!# ---------------------------------------------
CONST robtarget F54_Pick_01fx01:=[[0,0,0],[1,0,0,0],[-2,-1,1,1],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget F54_Drop1_02fx01:=[[-1.762382824,0.000074434,13.150425401],
[0.999998217,0.000000187,0.001888591,0.000000006],
[-1,0,2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget Target_10:=[[339.064816784,-1871.908698921,927.344154274],
[0.99424424,0.100314901,-0.000063828,-0.037620574],[-2,-1,1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget Target_20:=[[-0.001442277,-534.985019755,-0.016320854],
[1,-0.000000564,-0.000000907,-0.000000656],[-2,-1,1,1],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget Target_30:=[[-0.000490732,-534.984653964,-0.016982558],
[1,-0.000000631,-0.000000939,-0.000000666],[-2,-1,1,1],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget Target_40:=[[339.064158425,-1871.907849856,927.344388899],
[0.994244237,0.100314955,-0.000063865,-0.037620515],[-2,-1,1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget Target_50:=[[814.779187977,-6702.771271548,-738.498995073],
[0.682064409,0.384822854,-0.093492801,0.614783383],[-1,-1,1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget Target_60:=[[10.343178559,-1859.495421129,-375.664516479],
[0.945753462,0.324703196,0.001786119,0.010725378],[-1,-1,2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget Target_70:=[[-1.764806865,-314.993273272,13.149464592],[0.999998217,-
0.000000379,0.001888564,-0.000000013],[-1,0,2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget Target_80:=[[-1.764844417,-314.992886599,13.150670453],
[0.999998217,0.000000187,0.001888564,0.000000006],[-1,-1,2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget Target_90:=[[107.148172625,-2119.388784514,361.31887708],
[0.743312614,0.664613442,-0.06579657,-0.038028174],[-1,0,1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget Target_100:=[[6190.088481128,-2119.388784514,1252.435771289],
[0.550902983,0.421192222,0.503337594,-0.51551361],[-2,-1,1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

!# ------ TOOL DATA
!# ---------------------------------------------
TASK PERS tooldata t_Grp1_F54:=[TRUE,[[-774.615,-3426.312426033,1296.183392865], → Weergave grijper
[0.521333804,0.521333804,0.477714417,0.477714417]],[1,[0,0,1],[1,0,0,0],0,0,0]];

!# ---------------------------------------------
!# ------ WOBJ DATA
!# ---------------------------------------------
PERS wobjdata wobj_01fx01:=[FALSE,TRUE,"",[[1431.329535577,-4286.078146813,-1366.563],
[0.180407853,0.129510644,-0.20414125,0.953418245]],[[0,0,0],[1,0,0,0]]];
PERS wobjdata wobj_02fx01:=[FALSE,TRUE,"",[[2343.513248125,3330.53356474,-590.033],
[0.473146789,-0.473146789,0.525482745,-0.525482745]],[[0,0,0],[1,0,0,0]]];

PROC Program_1()

WaitUntil GetTypeCode()<>0 OR PLC_di_ProgSkip=1;
!*********************************************************************
! Wait for Typbit Value or Program Skip  1=(F56) / 2=(F55) / 3=(F54) *
!*********************************************************************




IF PLC_di_ProgSkip=1 RETURN ;
! Program Cancel

!*********************
```

```
!* Store Typbit Code *
!*********************
n_Type_Mem:=GetTypeCode();

!****************************
!* Echo TypeBit Code to PLC *
!****************************
SetTypeCode(n_Type_Mem);

TEST n_Type_Mem

CASE 1:
!=========================================================
!                TYPE  F54
!=========================================================
!Tool_Changer 1;
!=========================================================
! JOB 1 Pick F54 From 01fx01
!=========================================================

!Check correct tool on Robot
Tool_Changer\nNextTool:=1;

WaitUntil PLC_di_Job01_Enable=1 XOR PLC_di_ProgSkip=1;
!*****************************************
!* Wait for Job01 or Program Skip        *
!*****************************************

IF PLC_di_ProgSkip=1 RETURN ;
! Program Cancel

!Pick F54 From 01fx01
f54_01_01fx01_pick;

WaitUntil PLC_di_Job01_Enable=0;
!*****************************************
!* Wait for Job01 = 0                    *
!*****************************************

!=========================================================
! JOB 2 Drop F54 Part To Station 02fx01
!=========================================================


WaitUntil PLC_di_Job02_Enable=1 OR PLC_di_ProgSkip=1;
!*****************************************
!* Wait for Job02 or Program Skip        *
!*****************************************

IF PLC_di_ProgSkip=1 RETURN ;
! Program Cancel

!Drop F54 Part To Station 02fx01
f54_02_02fx01_Drop;




WaitUntil PLC_di_Job02_Enable=0;
!*****************************************
!* Wait for Job02 = 0                    *
!*****************************************
```

```
DEFAULT:
WHILE TRUE DO
Stop;
!*************************
!* Incorrect Type Info    *
!*      From PLC          *
!*************************
ENDWHILE
ENDTEST
RETURN ;
ENDPROC

! (End Program_1)

!#-------------------------------------------------------------------------------------------------------------------------------------------------------
------#!

PROC Program_2()

WaitUntil GetTypeCode()<>0 OR PLC_di_ProgSkip=1;
!*********************************************************************
! Wait for Typbit Value or Program Skip  1=(F56) / 2=(F55) / 3=(F54) *
!*********************************************************************

IF PLC_di_ProgSkip=1 RETURN ;
! Program Cancel

!**********************
!* Store Typbit Code *
!**********************
n_Type_Mem:=GetTypeCode();

!****************************
!* Echo TypeBit Code to PLC *
!****************************
SetTypeCode(n_Type_Mem);

TEST n_Type_Mem

CASE 1:
!=======================================================
!               TYPE  F54
!=======================================================
! JOB 2 Drop F54 Part To Station 02fx01
                      !=======================================================


WaitUntil PLC_di_Job02_Enable=1 XOR PLC_di_ProgSkip=1;
!*****************************************
!* Wait for Job02 or Program Skip         *
!*****************************************

IF PLC_di_ProgSkip=1 RETURN ;
! Program Cancel




!Drop F54 Part To Station 02fx01
f54_02_02fx01_Drop;
```

```
WaitUntil PLC_di_Job02_Enable=0;
!*****************************************
!* Wait for Job2 = 0                     *
!*****************************************

DEFAULT:
WHILE TRUE DO
Stop;
!*************************
!* Incorrect Type Info    *
!*      From PLC          *
!*************************
ENDWHILE
ENDTEST
RETURN ;
ENDPROC

! (End Program_2)

!#-------------------------------------------------------------------------------------------------------------------------------------------------------
------#!

PROC Program_62()

!*************************************************************

!SafeMove Bremsentest
IF DOutput(SM1_do_BrakeOk)=0 VelSet 100,150;

SM1_BrakeTest;

RETURN ;
ENDPROC

! (End Program_62)

!#-------------------------------------------------------------------------------------------------------------------------------------------------------
------#!


PROC f54_01_01fx01_pick() → start programma grijper

MoveAbsJ jpHome1,v5000,fine,t_Grp1_F54; → start in homepositie 1

GrpPartChk 0,GR01_B01PP\Part2:=GR01_B02PP\Part3:=GR01_B03PP\Part4:=GR01_B04PP;
GrpChk Opn,Grp_11\Grp2:=Grp_12\Grp3:=Grp_13; → controle grijper of het geen onderdeel bevat

MoveJ_JobRequest 25,[[339.064816784,-1871.908698921,927.344154274],
[0.99424424,0.100314901,-0.000063828,-0.037620574],[-2,-1,1,0],[9E9,9E9,9E9,9E9,9E9,9E9]],
v3000,fine,t_Grp1_f54\WObj:=wobj_01fx01;
MoveJ_JobRequest 1,[[339.064816784,-1871.908698921,927.344154274],
[0.99424424,0.100314901,-0.000063828,-0.037620574],[-2,-1,1,0],[9E9,9E9,9E9,9E9,9E9,9E9]],
v3000,fine,t_Grp1_f54\WObj:=wobj_01fx01;

! Col Req 5 with Robot 02r02;
! Col Req 10 with Robot 02r03;
MoveJ_CollReq 5\Coll_B:=10,[[339.064816784,-1871.908698921,927.344154274],
[0.99424424,0.100314901,-0.000063828,-0.037620574],[-2,-1,1,0],[9E9,9E9,9E9,9E9,9E9,9E9]],
v3000,fine,t_Grp1_f54\WObj:=wobj_01fx01\Coll_Desc:="02R02"\Coll_Desc_B:="02R03";

MoveJ Target_10,v5000,z10,t_Grp1_f54\WObj:=wobj_01fx01;
MoveJ Target_20,v5000,z10,t_Grp1_f54\WObj:=wobj_01fx01;
```

MoveL F54_Pick_01fx01,v500,fine,t_Grp1_F54\Wobj:=wobj_01fx01; → opnemen van onderdeel

GrpPartChk 1,GR01_B01PP\Part2:=GR01_B02PP\Part3:=GR01_B03PP\Part4:=GR01_B04PP;
GrpSet Cls,Grp_12\Grp3:=Grp_13\PartLoad:=ld_Pt_F54; → controle of grijper onderdeel vast heeft

SetRobotTypeCode\GripperSystem:=1\PartOnGripper:=TRUE\InPos;

PLC_Release "Pick_F54_Part"\Set_:=PLC_do_user_09\Wait_:=PLC_di_user_09,\InPos;

➔ Als het deel in de grijper zit zal de robot een userbit (PLC_do_user_09) hoog maken om tegen de plc te zeggen dat de robot in pick positie staat met het deel in de grijper. De robot zal wachten tot de plc een userbit (PLC_di_user_09) terug stuurt.

MoveL Target_30,v500,fine,t_Grp1_F54\WObj:=wobj_01fx01;

GrpPartChk 1,GR01_B01PP\Part2:=GR01_B02PP\Part3:=GR01_B03PP\Part4:=GR01_B04PP;

PLC_Release "Check_F54_Part"\Set_:=PLC_do_user_10\Wait_:=PLC_di_user_10;

MoveJ Target_40,v5000,z200,t_Grp1_f54\WObj:=wobj_01fx01;

MoveJ_JobFinished 1,[[339.064158425,-1871.907849856,927.344388899],
[0.994244237,0.100314955,-0.000063865,-0.037620515],[-2,-1,1,0],[9E9,9E9,9E9,9E9,9E9,9E9]],
v5000, z200, t_Grp1_f54\Wobj:=wobj_01fx01;

MoveJ Target_50,v5000,z200,t_Grp1_f54\WObj:=wobj_01fx01;

!Col Rel 5 with Robot 02r02;
!Col Rel 10 with Robot 02r03;

MoveJ_CollClr 5\Coll_B:=10,[[3148.048543697,-6777.010665241,-1468.28092364],
[0.565626048,0.320400565,-0.210605088,0.730106943],[-1,0,1,4],[9E9,9E9,9E9,9E9,9E9,9E9]],
v5000,fine,t_Grp1_f54\WObj:=wobj_01fx01;

MoveJ_JobFinished 25,[[3148.048543697,-6777.010665241,-1468.28092364],
[0.565626048,0.320400565,-0.210605088,0.730106943],[-1,0,1,4],[9E9,9E9,9E9,9E9,9E9,9E9]],
v5000, z100, t_Grp1_f54\Wobj:=wobj_01fx01;
MoveAbsJ jpHome2,v5000,fine,t_Grp1_F54; → programma pick stopt in homepositie 2

RETURN;
ENDPROC !(f54_01_01fx01_pick)

PROC f54_02_02fx01_drop() → start programma drop in homepositie 2

MoveAbsJ jpHome2,v5000,fine,t_Grp1_F54;

GrpPartChk 1,GR01_B01PP\Part2:=GR01_B02PP\Part3:=GR01_B03PP\Part4:=GR01_B04PP;
GrpChk Cls,Grp_12\Grp3:=Grp_13;
GripLoad ld_Pt_F54;

MoveJ_JobRequest 26,[[10.343178559,-1859.495421129,-375.664516479],
[0.945753462,0.324703196,0.001786119,0.010725378],[-1,-1,2,0],[9E9,9E9,9E9,9E9,9E9,9E9]],
v3000,fine,t_Grp1_f54\WObj:=wobj_02fx01;
MoveJ_JobRequest 2,[[10.343178559,-1859.495421129,-375.664516479],
[0.945753462,0.324703196,0.001786119,0.010725378],[-1,-1,2,0],[9E9,9E9,9E9,9E9,9E9,9E9]],
v5000,fine,t_Grp1_f54\WObj:=wobj_02fx01;

! Col Req 15 with Robot 02r02;
! Col Req 20 with Robot 02r03;
! Col Req 25 with Robot 03r01;
MoveJ_CollReq 15\Coll_B:=20\Coll_C:=25,[[10.343178559,-1859.495421129,-375.664516479],
[0.945753462,0.324703196,0.001786119,0.010725378],[-1,-1,2,0],[9E9,9E9,9E9,9E9,9E9,9E9]],
v5000,fine,t_Grp1_f54\WObj:=wobj_02fx01\Coll_Desc:="02R02"\Coll_Desc_B:="02R03"\Coll_Desc_C:="03R01";

```
MoveJ Target_60,v5000,z100,t_Grp1_f54\WObj:=wobj_02fx01;
MoveJ Target_70,v1000,z100,t_Grp1_f54\WObj:=wobj_02fx01;

MoveL F54_Drop1_02fx01,v1000,z100,t_Grp1_f54\WObj:=wobj_02fx01; → neerzetten van ondereel

GrpSet Opn,Grp_12\Grp3:=Grp_13\noGripChk;
GrpSet Cls,Grp_11\noGripChk;
GripLoad load0;

SetRobotTypeCode\GripperSystem:=1\PartOnGripper:=FALSE\InPos;

PLC_Release "Drop_F54_Part_1"\Set_:=PLC_do_user_11\Wait_:=PLC_di_user_11;

GrpChk Opn,Grp_12\Grp3:=Grp_13;
GrpChk Cls,Grp_11;

MoveL Target_80,v1000,z100,t_Grp1_f54\WObj:=wobj_02fx01;

GrpPartChk 0,GR01_B03PP\Part4:=GR01_B04PP;
GrpPartChk 1,GR01_B01PP\Part2:=GR01_B02PP;

SetRobotTypeCode\GripperSystem:=1\PartOnGripper:=FALSE\InPos;

PLC_Release "Drop_F54_Part_b"\Set_:=PLC_do_user_11\Wait_:=PLC_di_user_11;

MoveJ Target_90,v1000,z100,t_Grp1_f54\WObj:=wobj_02fx01;

MoveL_JobFinished 2,[[107.148172625,-2119.388784514,361.31887708],
[0.743312614,0.664613442,-0.06579657,-0.038028174],[-1,0,1,0],[9E9,9E9,9E9,9E9,9E9,9E9]],
 v500, fine, t_Grp1_f54\Wobj:=wobj_02fx01;

MoveJ Target_100,v1000,z100,t_Grp1_f54\WObj:=wobj_02fx01;

!Col Rel 15 with Robot 02r01;
!Col Rel 20 with Robot 02r02;
!Col Rel 25 with Robot 03r01;
MoveJ_CollReq 15\Coll_B:=20\Coll_C:=25,[[6190.088481128,-2119.388784514,1252.435771289],
[0.550902983,0.421192222,0.503337594,-0.51551361],[-2,-1,1,0],[9E9,9E9,9E9,9E9,9E9,9E9]],
 v5000, z100, t_Grp1_f54\Wobj:=wobj_02fx01;

MoveJ_JobFinished 26,[[574.237934718,-2784.448241462,1414.638032308],
[0.977371343,0.188378581,0.00321952,-0.096168615],[-2,0,1,4],[9E9,9E9,9E9,9E9,9E9,9E9]],
v5000,z200,t_Grp1_f54\Wobj:=wobj_02fx01;

MoveAbsJ jpHome1,v5000,fine,t_Grp1_F54; → einde programma, hierdoor beweegt robot naar homepositie 1, zodat
cyclus opniew begint.

RETURN;
ENDPROC !(f54_02_02fx01_drop)
ENDMODULE
```

Bijlage 7 Tipdress en kappenwissel

```
! Open dress station
TipChange_CMD Gun1, TipDr1\Motor_Right\TipVacuum_Fct;

!----------------------------------------
! Unlock - Check - Mount first tip (lower)
!----------------------------------------
MoveJ0 Clr_UnlockTip1_Gun1, Gun1, 100\GunZone:=10, vmax, z100, t_cGun1\Wobj:=wobj_Tipdresser1;
MoveL0 Pre_UnlockTip1_Gun1, Gun1, 100,vmax, z10, t_cGun1\WObj:=wobj_Tipdresser1;
MoveL0 UnlockTip1_Gun1, Gun1, 100, v200, fine, t_cGun1\Wobj:=wobj_Tipdresser1;

! Clamp tip 1 / vacuum on
TipChange_CMD Gun1, TipDr1\Motor_Left\TipVacuum_Fct;

MoveL0 Post_UnlockTip1_Gun1, Gun1, 100, vmax, z20, t_cGun1\Wobj:=wobj_Tipdresser1;

! Moves from Post_UnlockTip1_Gun1 position to CheckTip1_Gun1 position
MoveL0 [[183.8993,-90.89024,-111.8311],[0.976509,0.02114834,0.2113361,-0.03633035],[0,0,-2,0],
MoveL0 CheckTip1_Gun1, Gun1, 100, v200, fine, t_cGun1\Wobj:=wobj_Tipdresser1;

! Check tip 1 removed
TipChange_CMD Gun1, TipDr1\Tip_Check_Low;

! Open-clean dressing station
TipChange_CMD Gun1, TipDr1\Motor_Right;

! Moves from CheckTip1_Gun1 position to Pre_MountTip1_Gun1 position
MoveL0 [[179.7704,-194.5824,-123.5453],[0.9765101,0.02114703,0.211332,-0.03632558],[0,0,-2,0],[9E+09,
MoveL0 Pre_MountTip1_Gun1, Gun1, 100, vmax, z10, t_cGun1\WObj:=wobj_Tipdresser1;
MoveL0 MountTip1_Gun1, Gun1, 100\GunNoConc:=TRUE, v200, fine, t_cGun1\Wobj:=wobj_Tipdresser1;
```

Tang beweegt in de richting van het toestel

Kap wordt losgemaakt door de de tipchanger, (figuur 32). Hierna beweegt de tang naar beneden en zal de oude kap opgevangen worden in een afvalbak

Detectiesysteem kijkt of tip effectief verwijderd is

De tang beweegt naar het aanvultoestel om nieuwe kappen op de tangen te plaatsen

```
! Gun reference drive
MoveLG [[184.1993,-229.5585,-171.4476],[0.9999993,-0.0007175775,0.0003645778,-0.0008721625],[0,0,-2,(
SW_GunReference 1, 100;

MoveLG Pre_PressTip_Gun1, Gun1, 100, v300, z10, t_cGun1\WObj:=wobj_Tipdresser1;
MoveLG PressTip_Gun1, Gun1, 100, v200, fine, t_cGun1\WObj:=wobj_Tipdresser1;

! Press station
TipPress Gun1, 3120, 1.0\GunPos:=100, 6;

! Activate cooling water
TipChange_CMD Gun1, TipDr1\WaterOn;

! Reset counter
TipChange_CMD Gun1, TipDr1\CounterReset;
```



In de volgende stappen, zal een detectiesysteem controleren of de kap op de tang staat.

Vervolgens wordt de tang tegen elkaar bewogen om de GunReference vast te stellen.

Hierna wordt de tang samen geperst op een blok om ervoor te zorgen dat de kap stevig vast staat

```
! Moves from Post_PressTip_Gun1 position to Pre_Tipdress_Gun1 position
MoveLG [[224.4297,-146.549,-163.029],[0.9999984,-0.0006871795,0.0003776962,-0.001(
MoveLG [[222.0961,52.29181,-103.0505],[0.9999984,-0.0006896366,0.0003753061,-0.001
MoveLG Pre_Tipdress_Gun1, Gun1, 100, vmax, z20, t_cGun1\Wobj:=wobj_Tipdresser1;
MoveLG Tipdress_Gun1, Gun1, 100, v200, fine, t_cGun1\Wobj:=wobj_Tipdresser1;

TipDress_CMD Gun1_Par, 100\TripleHead;

MoveLG Post_Tipdress_Gun1, Gun1, 100, vmax, z40, t_cGun1\Wobj:=wobj_Tipdresser1;

! Gun calibration
MoveLG pReference1, Gun1, 100, vmax, fine, t_cGun1\Wobj:=wobj_Tipdresser1;
```


Na het vast zetten van de nieuwe kappen op de tang, wordt de tang naar de frees gebracht om ervoor te zorgen dat de kappen de gewenste afmeting hebben en klaar zijn voor het puntlassen

Bibliografie

Internet:

[1] Ciratec & C°. (2008). *Company*. Opgehaald van Ciratec & C°: http://www.ciratec.be/company.html

[2] ABB. (2014). RobotStudio. Opgehaald van ABB: http://new.abb.com/products/robotics/robotstudio

[3] Siemens Industry Software B.V. (2014). JT2Go. Opgehaald van Siemens: http://www.plm.automation.siemens.com/nl_nl/products/teamcenter/lifecycle-visualization/jt2go/

[4] Notepad++. (2011). About. Opgehaald van Notepad++: http://notepad-plus-plus.org/

ABB. (2013, 4 16). ABB Robotics - Product Documentation. Opgehaald van ABB: http://developercenter.robotstudio.com/Index.aspx?DevCenter=DevCenterManualStore&OpenDocument&Url=../RapidIFDTechRefManual/doc164.html

ABB. (2014). *RobotStudio download page*. Opgehaald van ABB: http://new.abb.com/products/robotics/robotstudio/downloads

Jens Van Laer. (2007, 11 5). *Dankwoord.* Opgehaald van ethesis: http://www.ethesis.net/nmbs/nmbs.pdf

Karen Vekeman . (2007, 6 13). *Dankwoord .* Opgehaald van dev.ulb.ac: http://dev.ulb.ac.be/ceese/ABC_Impacts/documents_abc/Vekeman.pdf

Tutorials:

ABB. (2009, 11 12). *RobotStudio Tutorials*. Opgehaald van Youtube : http://www.youtube.com/user/RobotStudio

Documenten:

BMW Group. (2013). 1MAN_12_01_ROB_Allgemeinteil_130108_de_en, Standard Software - ABB Robot IRC5., (p. 78).

BMW Group. (2013). 1MAN_12_02_ROB_SafeMove_100623_en, Standard Software - ABB Robot IRC5., (p. 12).

BMW Group. (2013). 1MAN_12_18_StackingComponent_130221_en, Standard Software - ABB Robot IRC5., (p. 14).

BMW Group. (2013). 1MAN_12_99_ROB_LU_Nameing_conventntion_2012-07-02_en, Standard Software - ABB Robot IRC5., (p. 10).

BMW Group. (2013). 1Weld_Type_ID_Definitions _LU_W34_Applications_260612, Standard Software - ABB Robot IRC5., (p. 2).

BMW Group. (2013). MAN_12_01_Rob_LU Additional_Guidelines_090213_EN_Update, Standard Software - ABB Robot IRC5., (p. 73).

BMW Group. (2013). MAN_12_03_ROB_SpotWelding_130110_en, Standard Software - ABB Robot IRC5., (p. 25).

BMW Group. (2013). MAN_12_04_ROB_Handling_130220_de_en, Standard Software - ABB Robot IRC5., (p. 35).

BMW Group. (2013). MAN_12_08_ LU_ROB_ToolChanger _130311_en, Standard Software - ABB Robot IRC5., (p. 28).

Figuren:

Figuur2: Mini. (2014). *Home*. Opgehaald van MINI Plant Oxford : http://www.mini-production-triangle.com/

Figuur 7: VDL Steelweld . (2009, 03 19). *Training Robottechniek.* Opgehaald van Induteq: http://www.induteq.nl/induteq/bestanden/Training%20Robottechniek.pdf

Figuur 28: ABB university (28.06.2013). IRC5 Veiligbedienen revA-NL, (p. 18).

Figuur 29: ABB university (28.06.2013). IRC5 Veiligbedienen revA-NL, (p. 18).

Figuur 30: ABB university (28.06.2013). System Introductie_nl, (p. 26).