

# Onderzoek naar open source HMI's

***EINDVERSLAG***

**V. 2.0.0**

*Lars Wicherson*

*Saxion Deventer*



**MA-IT B.V.**

085 130 25 26  
[info@ma-it.nl](mailto:info@ma-it.nl)

Keulenstraat 21 | 7418ET DEVENTER  
Science Park Eindhoven 5220 | 5692EG SON  
Einsteinstraat 9 | 4207HZ GORINCHEM

[www.my-automation.nl](http://www.my-automation.nl)  
[linkedin.com/company/960387](https://www.linkedin.com/company/960387)

## Communicatie

### Klantinformatie

Contactgegevens:

MA-IT

Keulenstraat 21

7418ET Deventer

Tel 085 130 2526

URL <https://my-automation.nl/>

Contactpersoon	Functie	Telefoon	e-mail
Rudy de Bock	Projectleiding oen software Engineer ICT	085-1 302526	bock@ma-it.nl

### Schoolinformatie

Contactgegevens:

Saxion Deventer

Handelskade 75

7417DH Deventer

Tel 088 019 8888

URL <https://my-automation.nl/>

Contactpersoon	Functie	Telefoon	e-mail
Etto Salomons	Hoofddocent HBO-ICT	0622490677	e.l.salomons@saxion.nl

### Persoonsinformatie

Contactpersoon	Functie	Telefoon	e-mail
Lars Olav Wicherson	Afstudeerder	0648471145	349038@student.saxion.nl

## Voorwoord

---

In het kader van de afstudeerstage, onderdeel van de academie Creatieve Technologie van Saxion Hogeschool te Deventer, heb ik dit eindverslag geschreven. De afstudeerstage werd gelopen bij My Automation-Information Technology (MA-IT).

Graag wil ik meneer de Bock bedanken voor zijn begeleiding tijdens het schrijven van dit eindverslag. Daarnaast wil ik meneer Bulsink bedanken voor de afstudeerstage en de opdracht zelf en wil ik meneer Waarle en meneer Keizer bedanken voor de hulp met de mini-factory. Verder wil ik de verschillende personen bedanken die ik heb mogen interviewen. Tenslotte wil ik graag meneer Salomons bedanken voor het begeleiden en nakijken van dit afstudeerverslag.

Lars Wicherson,  
Vierdejaars HBO-ICT student

Deventer, 15-06-2020

## Versie beheer

In dit hoofdstuk worden de verschillende versies genoteerd van het verslag. Zie tabel 1 hieronder.

Tabel 1: Versies van het eindverslag

Versie	Datum	Beschrijving
1.0.0	03-04-2020	Eerste versie eindverslag
1.0.1	10-04-2020	Afkortinglijst bijgewerkt, begrippenlijst bijgewerkt, extra informatie theoretische achtergrond VisiWin, Ignition, ICONICS, Zenon theoretische achtergrond gemaakt, kopje Interviews ingevuld, kopje Informatievergaring ingevuld, kopje beoordeling hmi's matrix ingevuld.
1.0.2	16-04-2020	Voorwoord, Samenvatting, Discussie, Conclusie, Aanbevelingen, Spellingcontrole, laatste controle check.
1.0.3	28-05-2020	Methoden ontwerpen, implementatie en testscenario's toegevoegd , bij resultaten ontwerpen en implementatie toegevoegd.
1.0.4	01-06-2020	Hoofdstuk indeling aangepast, Extra info hoofdstukken
2.0.0	15-06-2020	Definitieve versie eindverslag

## Gebruikte afkortingen

In dit hoofdstuk worden de gebruikte afkortingen beschreven met de betekenis.

Tabel 2: Gebruikte afkortingen

<b>HMI</b>	Human Machine Interface
<b>PLC</b>	Programmable Logic Controller
<b>OPC UA</b>	Open Platform Communications Unified Architecture
<b>C#</b>	C Sharp
<b>VB.NET</b>	Visual basic .NET
<b>API</b>	Application programming interface
<b>SSL</b>	Secure Sockets Layer
<b>TLS</b>	Transport Layer Security
<b>XAML</b>	Extensible Application Markup Language
<b>SCADA</b>	Supervisory Control And Data Acquisition

## Begrippenlijst

In dit hoofdstuk wordt de begrippenlijst beschreven.

Tabel 3: begrippenlijst

<b>HMI</b>	Human Machine Interface is een onderdeel van bepaalde typen apparaten die in staat zijn om interacties tussen mensen en machines te verwerken (AT-Automation, 2019).
<b>PLC</b>	PLC-systemen worden onder andere gebruikt voor het automatiseren van machine of productielijnen van fabrieken etc (Geertsma, 2013).
<b>OPC UA</b>	OPC is een protocol om te communiceren binnen en tussen de verschillende automatiseringslagen in de industrie. UA staat voor unified architecture wat betekent dat het op elk platform kan worden geïmplementeerd (IOS, Linux, windows) (Novotek, 2015).
<b>Zero install</b>	Zero Install geeft je de mogelijkheid om eenvoudig draagbare software binnen te halen en uit te voeren. Er is dan geen installatie vereist. Het programma heeft een bibliotheek van ongeveer 80 programma's die je met een druk op de knop kunt starten (Goede, 2016).
<b>SSL encryptie</b>	Een beveiligingscertificaat dat ingezet wordt om de data tussen browser en server te versleutelen (KPN.com, 2020).
<b>SCADA</b>	SCADA, afkorting van <i>Supervisory Control And Data Acquisition</i> , is het verzamelen, doorsturen, verwerken en visualiseren van meet- en regelsignalen van verschillende machines in grote industriële systemen.

---

## Samenvatting

---

Het bedrijf MA-IT levert software aan andere bedrijven. Zij programmeren onder andere *Programmable Logic Controllers* (PLC's) bij het desbetreffende bedrijf. Tevens kan een *Human Machine Interface* (HMI) gebruikt worden bij een PLC. Het doel van dit onderzoek was om te onderzoeken welk soort open source HMI met de programmeertaal C# gebruikt kan worden bij MA-IT, zodat de programmeurs één opleiding hoeven te doen voor één soort HMI.

Uit de probleem- en doelstelling was de volgende hoofdvraag opgesteld: "Welke open source HMI is het meest geschikt voor MA-IT om te implementeren in haar mini-factory, zodat deze gebruikt kan worden als nieuwe HMI-tool naast de bestaande HMI-oplossingen die al bekend zijn en gebruikt worden?".

Om het doel te bereiken en antwoord te krijgen op de hoofdvraag en de daarbij behorende deelvragen zijn verschillende methodes toegepast. Ten eerste is er een onderzoek uitgevoerd waaruit kwam dat VisiWin het meest geschikt zou zijn omdat deze wel voldeed aan de harde C# eis. Ten tweede is er een analyse geweest waarin alle requirements van de HMI naar voren kwamen. Ten derde zijn op basis van de requirements ontwerpen gemaakt. Ten vierde is er in VisiWin een HMI gerealiseerd voor de mini-factory. Ten slotte zijn er testen geschreven om de kwaliteit van het product te waarborgen.

Tijdens het gebruik van VisiWin bleek echter bij de import van gegevens richting de mini-factory dat VisiWin niet compatibel was met de structs die worden gebruikt in de Allen-Bradley PLC. Omdat het niet zo mag zijn dat de software in de PLC omgeschreven moet worden omdat VisiWin het niet kan begrijpen is de conclusie getrokken dat VisiWin niet bruikbaar is binnen MA-IT als nieuwe open source HMI.

Dus, om antwoord te geven op de hoofdvraag zou VisiWin het meest geschikt zijn vanwege de harde eisen vanuit MA-IT. Echter is gebleken dat VisiWin niet bruikbaar is voor de mini-factory en dus ook niet bruikbaar is binnen MA-IT.

## Inhoud

1	Inleiding.....	9
1.1	Aanleiding .....	9
1.2	Probleemstelling .....	9
1.3	Doelstelling .....	10
1.4	Onderzoeksvragen .....	10
1.5	Methode .....	11
1.6	Afbakening .....	11
1.7	Leeswijzer .....	11
2	Theoretische achtergrond.....	12
2.1	Bedrijfsomschrijving .....	12
2.2	Soorten HMI's .....	13
3	Onderzoek .....	15
3.1	Methoden .....	15
3.2	Onderzochte HMI's .....	18
3.3	VisiWin .....	19
3.4	Ignition .....	22
3.5	Iconics .....	24
3.6	Zenon.....	27
3.7	Samenvatting onderzoeksresultaten .....	29
3.8	Beoordeling HMI's matrix.....	31
4	Analyseren .....	33
4.1	Matrix .....	33
4.2	Programmeertalen PLC .....	33
4.3	Mini-factory .....	34
5	Ontwerpen .....	35
5.1	Methode .....	35
5.2	Functioneel ontwerp .....	35



5.3	Technisch ontwerp.....	37
6	Realiseren.....	38
6.1	Methode.....	38
6.2	Implementatie HMI.....	38
6.3	Opbouw en gebruik PLC code.....	39
6.4	Functionaliteiten in VisiWin inbouwen .....	45
6.5	Problemen realisatie met VisiWin.....	48
7	Testen.....	50
7.1	Methode.....	50
8	Conclusie .....	52
9	Aanbevelingen .....	53
9.1	Aanbeveling vervolgonderzoek .....	53
9.2	Aanbevelingen voor MA-IT .....	53
10	Behaalde HBO-I competenties .....	54
11	Reflectie.....	56
	Bibliografie .....	58
	Bijlagen .....	61
	Bijlage I: Interviewvragen .....	61
	Bijlage II: Uitwerking interviews .....	62
	Bijlage III: Informatievergaring bedrijven HMI's .....	81
	Bijlage IV: Uitwerking informatievergaring HMI's.....	89
	Bijlage V: Beoordelingsmatrix HMI's.....	96
	Contactinformatie.....	97
	Bedrijfsgegevens .....	97

# 1 Inleiding

In dit hoofdstuk wordt de aanleiding beschreven voor deze afstudeeropdracht. Daarna worden de daaruit volgende probleemstelling en doel beschreven. Vervolgens worden de verschillende onderzoeksvragen beschreven. Daarna wordt de methode beschreven van het onderzoek. Verder wordt de afbakening van de afstudeeropdracht beschreven en ten slotte wordt een leeswijzer beschreven.

## 1.1 Aanleiding

Het bedrijf MA-IT levert software aan andere bedrijven. Zij programmeren onder andere *Programmable Logic Controllers* (PLC's) bij het desbetreffende bedrijf. Een PLC is een programmeerbare logische eenheid. Een PLC is een elektronisch apparaat in de vorm van een digitale computer. In deze computer is een microprocessor aanwezig die op basis van informatie, die hij via zijn ingangen ontvangt, een aantal uitgangen aanstuurt. PLC-systemen worden onder andere gebruikt voor het automatiseren van machines (Geertsma, 2013). Tevens kan een *Human Machine Interface* (HMI) gebruikt worden bij een PLC. Een HMI is een onderdeel van bepaalde typen apparaten die in staat zijn om interacties tussen mens en machines te verwerken. De interface bestaat uit hardware en software die het mogelijk maakt dat gebruikersinput vertaald wordt naar signalen voor machines, die op hun beurt de gebruiker weer voorzien van het gewenste resultaat (AT-Automation, 2019). Een HMI zorgt ervoor dat alles visueel gemaakt kan worden van wat er in de PLC gebeurt. Er zijn veel verschillende soorten PLC's met hun eigen HMI. Deze hebben allen een verschillende werkwijze en dienen dus anders geprogrammeerd te worden.

## 1.2 Probleemstelling

Op dit moment zijn er veel verschillende PLC's en bijbehorende traditionele HMI's. Echter werken deze HMI's alleen op hetzelfde merk PLC. Een merk A HMI kan niet worden gebruikt op een merk B PLC. Omdat er veel verschillende type PLC's en HMI's zijn, zou een programmeur van MA-IT een opleiding moeten krijgen voor al deze verschillende soorten. Dit kost veel geld en tijd voor MA-IT. Tevens maakt MA-IT gebruik van de programmeertaal C# in de ontwikkelomgeving van Visual Studio.

De volgende probleemstelling kan geformuleerd worden: "MA-IT dient hun programmeurs opleidingen te verschaffen voor elk soort PLC en HMI die aanwezig is

bij het bedrijf waar zij gaan programmeren". Het gevolg hiervan is dat dit niet efficiënt werkt en dat mogelijkheden op extra rendement verloren gaan.

### 1.3 Doelstelling

Het doel van dit onderzoek was om te onderzoeken welk soort open source HMI met de programmeertaal C# gebruikt kan worden bij MA-IT, zodat zij één opleiding hoeven te doen voor één soort HMI. Deze open source HMI kan worden aangesloten op elk soort PLC. Het onderzoek is gedaan aan de hand van literatuuronderzoek, interviews met personen die werkzaam zijn met de soorten open source HMI's, informatievergaring bij bedrijven en aan de hand van de beoordeling van de verschillende open source HMI's met een matrix die is opgezet met MA-IT.

Na het onderzoek wordt er één HMI geïmplementeerd op de mini-factory.

### 1.4 Onderzoeksvragen

#### 1.4.1 Hoofdvraag

Uit de probleem- en doelstelling was de volgende hoofdvraag opgesteld: "Welke open source HMI is het meest geschikt voor MA-IT om te implementeren in haar mini-factory, zodat deze gebruikt kan worden als nieuwe HMI-tool naast de bestaande HMI-oplossingen die al bekend zijn en gebruikt worden?".

#### 1.4.2 Deelvragen

Om de hoofdvraag te kunnen beantwoorden zijn onderstaande deelvragen opgesteld.

1. Wat is een HMI en waar wordt deze voor gebruikt?
  - a. Welke HMI's worden nu toegepast en waarvoor?
2. Hoe werkt een HMI in combinatie met een Allen-Bradley PLC en hoe ziet dit er in de praktijk uit?
  - a. Welke PLC's zijn nog meer compatible met de HMI en hoe werkt dit?
  - b. Welke communicatieprotocollen kent het HMI-systeem?
  - c. Wat zijn de voor-en nadelen van de gekozen HMI ten opzichte van bestaande HMI-systemen?
3. Welke HMI's maken gebruik van de programmeertaal C# in Visual Studio?
4. Hoe kan de HMI gebruikt worden in de mini-factory van MA-IT?

## 1.5 Methode

Om het doel te bereiken en antwoord te krijgen op de hoofdvraag en de daarbij behorende deelvragen zijn verschillende methodes toegepast. Ten eerste is er literatuuronderzoek verricht naar de verschillende te onderzoeken open source HMI's. Ten tweede zijn er interviews gehouden met gebruikers van deze HMI's. Ten derde zijn de verschillende bedrijven benaderd via e-mail met aanvullende vragen met betrekking tot de werking van de open source HMI's. Ten vierde zijn de verschillende open source HMI's beoordeeld aan de hand van een matrix die is opgesteld met MA-IT.

Na het onderzoek zijn er requirements opgesteld voor de HMI. Aan de hand van de requirements zijn er ontwerpen gemaakt die vervolgens gerealiseerd zijn in de gekozen HMI. Tenslotte zijn deze functionaliteiten nog getest.

## 1.6 Afbakening

Het onderzoek is gedaan in twintig weken. Het onderzoek liep van 10 februari 2020 tot en met 12 juni 2020. De volgende activiteiten hebben plaatsgevonden tijdens het afstuderen:

- Onderzoek naar oudere HMI-frameworks;
- Onderzoek naar open source HMI-frameworks;
- Beschrijven van documentatie;
- Realiseren van software met behulp van het gekozen HMI-framework.

## 1.7 Leeswijzer

In hoofdstuk twee wordt de theoretische achtergrond beschreven. In hoofdstuk drie wordt het onderzoek beschreven. In hoofdstuk vier worden de analyses beschreven. In hoofdstuk vijf worden de ontwerpen beschreven. In hoofdstuk zes wordt het realiseren van de HMI beschreven. In hoofdstuk zeven wordt beschreven hoe de functionaliteiten zijn getest. In hoofdstuk acht wordt de conclusie beschreven. In hoofdstuk negen worden de aanbevelingen gedaan. Tenslotte worden in hoofdstuk tien en elf respectievelijk de behaalde competenties van de onderzoeker en de reflectie beschreven.

## 2 Theoretische achtergrond

---

In dit hoofdstuk wordt de theoretische achtergrond toegelicht van het onderzoek naar de verschillende HMI's die onderzocht zijn. Allereerst wordt er informatie beschreven over het bedrijf. Verder wordt er informatie gegeven over de mini-factory van MA-IT. Daarna wordt er informatie beschreven over de huidige en voorgaande werkwijzen met betrekking tot de HMI. Ten slotte wordt er informatie beschreven over de gewenste HMI's.

### 2.1 Bedrijfsomschrijving

MA-IT (My Automation-Information Technology) is een bedrijf dat drie vestigingen heeft in Nederland. Een van hun kantoren is gevestigd in Deventer waar de afstudeeropdracht zal worden gerealiseerd. MA-IT helpt bedrijven en organisaties bij het succesvol uitvoeren van projecten op het gebied van engineering, industriële automatisering of kantoorautomatisering. Hierin kan zij onderscheidend zijn door de kennis en kunde van inmiddels meer dan 70 medewerkers. Dit zijn zonder uitzondering hoog opgeleide engineers, programmeurs en consultants, die actief zijn in een grote verscheidenheid aan branches (MA-IT, 2020). De meeste mensen die werken bij MA-IT zijn programmeurs. Hiervan is 80% PLC programmeurs en ongeveer 20% software engineers.

### 2.2 Mini-factory

De mini-factory (afbeelding 1) bestaat uit twaalf units die bestuurd moeten worden met behulp van een HMI. Het doel van de mini-factory is om een product volledig rond te laten gaan waardoor het blokje zogenaamd wordt bewerkt. De warehouse begint bij unit 1 "de warehouse". Dit is de opslag van de mini-factory. Er kan een blokje opgehaald worden met unit 2 "de elevator". Deze brengt het opgehaalde blokje naar unit 3 "de conveyor" die het blokje vervolgens naar unit 4 brengt "de kraan". Unit 4 pakt het blokje op met behulp van een zuignap en brengt deze naar unit 5 "de vacuüm transport". Nadat de kraan het blokje heeft opgepakt uit unit 3 zal unit 3 het lege krat terug brengen naar de elevator die het lege krat vervolgens weer terug zet in de warehouse. Vanaf unit 5 zal het blokje eerst unit 6 "de oven" in gaan. Vervolgens brengt de vacuüm transport het blokje naar unit 7 "de turn table". Deze pakt het blokje en brengt het blokje naar unit 8 "de zaag". Als het blokje zogenaamd is gezaagd zal de turn table het blokje naar unit 9 en 10 brengen (allebei banden). Deze banden zullen het blokje naar unit 11 brengen waar de kleur van het blokje wordt geanalyseerd.

Vervolgens zal het blokje met behulp van de band bij unit 12 belanden waar hij aan de hand van de kleur in het juiste sorteer station wordt geduwd. Vervolgens zal unit 4 de kraan weer het blokje ophalen en naar unit 3 de conveyor brengen. Ondertussen zal de elevator een lege krat ophalen waar unit 4 het blokje weer in kan leggen. Als het blokje in het lege krat ligt zal de elevator het blokje weer terug plaatsen in de warehouse. Het blokje is dan klaar met productie.



Afbeelding 1: Mini-factory van MA-IT

## 2.3 Soorten HMI's

MA-IT maakt gebruik van traditionele HMI's. Traditionele HMI's zijn HMI's die alleen op hun eigen merk PLC geprogrammeerd kunnen worden. Zo heb je bijvoorbeeld een merk A PLC. Dit zou betekenen dat als je op een merk A PLC een HMI wilt toevoegen dat deze ook van het merk A moet zijn. Het kan dus niet zo zijn dat op merk A PLC een merk B HMI gekoppeld kan worden. Dit geldt ook andersom. Als je een merk B PLC hebt kun je geen merk A HMI er aan koppelen. Het komt wel eens voor dat MA-IT een klant heeft waar de klant werkt met een PLC waar MA-IT nog niet mee heeft gewerkt.



Dit zou betekenen dat eerst iemand in MA-IT opgeleid moet worden om ook de nieuwe HMI te leren van het nieuwe merk PLC. Dit kan zonde zijn van tijd en geld omdat er een grote kans bestaat dat de werknemer alleen voor dat project gaat werken met de nieuwe HMI. De Supervisory Control And Data Acquisition (SCADA) pakketten die gebruikt worden bij MA-IT zijn onder andere Siemens WinCC, WonderWare Intouch of Orchestra, Citect, IFIX, Ciimplicity en Proleit.

Naast traditionele HMI's bestaan er tevens open source HMI's. Dit zijn HMI's die bijna alle soorten PLC's ondersteunen. Daarnaast kan de gebruiker van de HMI ook zelf software schrijven voor de HMI. Zo zit de gebruiker niet vast aan de standaard functionaliteiten van een HMI. Een paar van deze namen zijn VisiWin, Ignition, ICONICS en Zenon.

## 2.4 Oude HMI voor mini-factory

De mini-factory had een HMI die gemaakt was met FactoryTalk View Studio. FactoryTalk is een traditionele HMI speciaal voor Allen-Bradley PLC's. Voor het maken van de nieuwe HMI zal er rekening gehouden moeten worden met de oude HMI van de mini-factory. Zo worden bepaalde functionaliteiten meegenomen die ook gebouwd worden in de nieuwe HMI. Een voorbeeld hiervan is het rond brengen van de blokjes door de mini-factory. Daarnaast is het doel om in de nieuwe HMI problemen zoals navigatie en overzichtelijkheid te verbeteren. In afbeelding 2 is een foto te zien van het hoofdscherm van het oude HMI. Hier is goed te zien dat het scherm erg rommelig is.



Afbeelding 2: Oude HMI van de Mini-factory

## 3 Onderzoek

In dit hoofdstuk worden de verschillende onderdelen van het onderzoek beschreven. Ten eerste worden de verschillende methoden beschreven die zijn gebruikt. Daarna worden per HMI de resultaten beschreven van de verschillende methoden. Tenslotte wordt een samenvatting gegeven van de onderzoeksresultaten die overeenkomen tussen de verschillende HMI's.

### 3.1 Methoden

In deze paragraaf worden de gebruikte methodes toegelicht die nodig waren om het onderzoek naar meest geschikte open source HMI voor MA-IT uit te voeren. Er wordt uitgelegd hoe het literatuuronderzoek is verricht en hoe de interviews zijn afgenomen. Verder wordt beschreven hoe verschillende informatie is vergaard bij de bedrijven van de verschillende onderzochte HMI's. Daarna wordt beschreven hoe de verschillende HMI's zijn beoordeeld aan de hand van een matrix.

#### 3.1.1 Literatuuronderzoek

Voor het onderzoek naar het vinden van een geschikte open source HMI voor MA-IT diende er eerst literatuuronderzoek gedaan te worden. Aan de hand van literatuuronderzoek kon er meer informatie worden vergaard over de werking van de huidige HMI. Daarnaast werd er meer informatie vergaard over de werking van de onderzochte eventuele geschikte HMI's die geïmplementeerd konden worden. De informatie over de HMI's is gevonden op de originele sites van het bedrijf dat de specifieke HMI aanbiedt.

#### 3.1.2 Interviews

Alle interviews zijn met toestemming van de geïnterviewde opgenomen. Tijdens het interview stelde de onderzoeker de vooraf opgestelde vragen aan de geïnterviewde. Deze vragen zijn opgenomen in bijlage I. Er werd zowel gebruik gemaakt van open als gesloten vragen. Het waren semigestructureerde interviews. Dat houdt in dat tijdens het interview de vooraf opgestelde vragen leidend waren. Echter konden nieuwe vragen gesteld worden en kon er dieper op een vraag ingegaan worden.

Er zijn drie mensen geïnterviewd die zelf gebruik maken van deze HMI's of die in het verleden gebruik hebben gemaakt van de HMI's tijdens hun werkzaamheden.



Voor VisiWin is Stefan van Dijken geïnterviewd. Hij werkt bij Mpac in vestiging Wijchen. Zij gebruiken VisiWin voor de fijnere afstelling en besturing van de inpakmachines. Dit was echter geen mondeling interview. De geïnterviewde heeft de vragen toegestuurd gekregen en vervolgens beantwoord.

Bij Ignition is er niemand geïnterviewd omdat er geen directe relatie is tussen iemand van MA-IT en iemand die Ignition gebruikt. Dit heeft er voor kunnen zorgen dat er minder informatie is verkregen dan andere HMI's. Echter was de site van Ignition zo duidelijk dat bijna alle informatie goed te vinden was door de onderzoeker zelf.

Voor het interview met ICONICS is Wim Janssen geïnterviewd. Hij werkt Etronics en werkt hier al 42 jaar. Er wordt bij Etronics sinds 2016 met ICONICS gewerkt.

Voor het interview met Zenon is Rob Bastens geïnterviewd. Hij heeft gewerkt bij NedCar in Born. Hij heeft Zenon gebruikt van 2015 tot 2018.

De uitwerking van de interviews is te vinden in bijlage II.

### 3.1.3 Informatievergaring bij bedrijven HMI's

Om extra expertise te ontvangen zijn er mails opgesteld naar alle HMI support e-mails. Er zal hier kort samengevat worden wat de vragen waren. Uit de antwoorden van de bedrijven is gebleken dat sommige informatie al bekend was bij de onderzoeker. Dit is al beschreven in de theoretische achtergrond. De extra informatie zal worden beschreven bij de paragraaf van de desbetreffende HMI. Zie bijlage III voor de vragen die zijn gesteld en voor de antwoorden die zij hebben gegeven.

De vragen waren gebaseerd op de matrix waarop de HMI's beoordeeld worden. Deze wordt in het volgende kopje uitgebreid uitgelegd. De vragen waren:

- Hoe makkelijk is het om de HMI te installeren en hoe is het mogelijk om met een Allen-Bradley PLC te connecten?
- Is er documentatie voor programmeurs? Is er een actieve community waar je vragen kan stellen over je werk en zijn er code snippets beschikbaar?
- Wat zijn de kosten voor een licentie van de HMI.
- Kan de gebruiker zijn eigen componenten maken?
- Hoe is de performance van de HMI en werkt het op verschillende PLC's?
- Welke functies ondersteunt de HMI denkend aan dat de HMI in ieder geval alarmen, rollen, mobiel, screen resoluties en dashboards ondersteunt?

Zenon heeft niet geantwoord op de vragenlijst. VisiWin heeft wel geantwoord maar heeft in plaats van de vragen per mail te beantwoorden een demo gegeven over VisiWin. Ignition reageerde met hele uitgebreide antwoorden op de vragenlijst. ICONICS heeft geantwoord op de mail maar de antwoorden waren relatief kort.

De antwoorden van de bedrijven op de e-mails is opgenomen in bijlage IV.

### 3.1.4 Beoordeling HMI's

De HMI's zijn beoordeeld met behulp van een matrix. Deze matrix is opgezet met behulp van MA-IT. Zij hebben aangegeven wat belangrijk is voor een HMI binnen hun bedrijf. Hierop is de matrix tevens gerangschikt. De matrix is opgenomen in een Excelbestand, maar een voorbeeld schermopname is opgenomen in bijlage VI.

De HMI's zijn beoordeeld met behulp van een HMI matrix. Deze matrix en wegingen zijn opgezet samen met MA-IT. De matrix eisen zijn als volgt:

1. Installatie en koppeling
  - 1.1. Is de HMI makkelijk te installeren?
  - 1.2. Is de HMI makkelijk om te koppelen met een PLC?
2. Documentatie
  - 2.1. Hoe is de documentatie van de HMI?
  - 2.2. Is de documentatie makkelijk te begrijpen voor nieuwe developers?
  - 2.3. Is er een actieve community waar je vragen aan kan stellen?
  - 2.4. Zijn er code snippets beschikbaar?
3. Development
  - 3.1. Kun je makkelijk met het HMI schermen maken die er goed uitzien?
  - 3.2. Kun je snel schermen maken in weinig tijd?
  - 3.3. Hoe is de learning curve?
4. Kosten
  - 4.1. Wat zijn de kosten van de HMI?
  - 4.2. Hoeveel tags zijn er beschikbaar?
5. Programming paradigma

- 5.1. Is het goed te begrijpen hoe je een technisch design kan maken van je applicatie.
- 5.2. Kun je componenten aanmaken die je opnieuw kan gebruiken?
- 5.3. Bestaan er al componenten die je kan gebruiken?
  
6. Performance en platforms
  - 6.1. Hoe is de performance van de HMI?
  - 6.2. Is de HMI snel of duurt het lang om data te versturen en te visualiseren?
  - 6.3. Kan de HMI op meerdere PLC's gebruikt worden?
  
7. Ondersteuning
  - 7.1. Ondersteunt de HMI functies zoals alarmen, rollen voor gebruikers, dashboards, verschillende resoluties en mobiel?

Ten tijde van het opstellen van de matrix was de harde eis van C# en Visual studio nog niet bekend bij de onderzoeker. Hierdoor was deze eis nog niet meegenomen in de matrix die op het begin is opgesteld van het onderzoek. Er is voor gekozen om de harde eis niet mee te nemen in de matrix om zo te kunnen zien of het misschien gunstig is om deze harde eis later te laten vallen.

Bij de resultaten van de verschillende HMI's staan de verschillende tabellen per HMI beschreven en met daarin de scores op de verschillende kritiekpunten. % in de tabel betekent hoeveel % dit onderdeel meeweegt. Criterion betekent de criteria van het onderdeel, zoals hierboven is beschreven. Grade betekent het cijfer wat het onderdeel heeft gekregen. Weighted grade betekent het gewogen gemiddelde van het onderdeel en deze is berekend door % keer Grade te doen en dan gedeeld 100 ( $\% \times \text{Grade} / 100$ ). Tevens wordt de totaalscore weergegeven. Deze is berekend door de som te nemen van alle gewogen gemiddelde van alle afzonderlijke onderdelen.

Voor het volledige bestand over hoe de HMI's beoordeeld zijn op elk kritiekpunt is bestand Beoordeling\_HMI.xlsx bijgevoegd.

### 3.2 Onderzochte HMI's

De onderzochte open source HMI's zijn:

- VisiWin;

- Ignition;
- ICONICS;
- Zenon.

### 3.3 VisiWin

#### 3.3.1 Resultaten literatuuronderzoek

VisiWin van Inosoft is een gemaakt door een Duits bedrijf. VisiWin biedt een HMI-software aan waarmee je relatief eenvoudig schermen kan aanmaken en aanpassen voor je PLC's. Omdat deze software open source is kun je eenvoudig eigen code toevoegen om componenten te maken. Dit gebeurt in de eigen ontwikkelomgeving die VisiWin heeft. VisiWin werkt het beste als het gaat om meer specifieke taken van procesvisualisatie. VisiWin maakt gebruik van de programmeertalen C sharp (C#) en Visual basic .NET (Vb.NET) en biedt een complete *Application Programming Interface* (API) voor het implementeren van functies. Met de koppeling tussen Blend, Visual studio en de ontwikkelomgeving van VisiWin is het maken van schermen nog eenvoudiger (INOSOFT a, 2020).

Voordat het proces in gebruik kan worden genomen met VisiWin, dient het eerst in de ontwikkelomgeving te worden geïmplementeerd en getest. Dit vindt plaats in de VisiWin-ontwikkelomgeving. De VisiWin-ontwikkelomgeving en de projecten daaruit kunnen worden gebruikt door de Microsoft-ontwikkeltools Visual Studio en Blend voor Visual Studio (INOSOFT b, 2020).

Het engineeringconcept is opgesplitst in twee afzonderlijke fasen:

- Fase 1:
  - o De software-engineer ontwikkelt afzonderlijke componenten in Visual Studio. Deze componenten worden vervolgens in de VisiWin-componentenbibliotheek geladen.
- Fase 2:
  - o De software-engineer heeft toegang tot de aangepaste componenten in de componenten-bibliotheek en kan deze samen met de standaard VisiWin-componenten gebruiken om een gebruikersinterface te maken (INOSOFT b, 2020).

Aangezien VisiWin een open source systeem is dat niet wordt gekoppeld aan een specifiek paneel, geldt dit ook voor de communicatie met de automatisering hardware zoals besturings- en aandrijfsystemen. Door middel van een Open Platform

Communications- (OPC)-Interface of een eigen stuurprogramma wordt de communicatie in VisiWin losgekoppeld van het runtime-systeem. De soorten OPC's zijn Classic en/ of Unified Architecture (UA). De communicatie met de PLC loopt voornamelijk via Ethernet (INOSOFT c, 2020).

De communicatie tussen de PLC en VisiWin is met een TCP/IP protocol. Om te communiceren met een Allen-Bradley PLC kan er gebruikt gemaakt worden van CompactLogix en ControlLogix (INOSOFT c, 2020).

### 3.3.2 Interviews

Uit het interview met een gebruiker van VisiWin is gebleken dat VisiWin de volgende voordelen heeft:

- Het framework is open source;
- Er kan gebruik gemaakt worden van dynamische afbeeldingen;
- Er kan gebruik gemaakt worden van aliasing;
- Het kan gebruikt worden op meerdere besturingssystemen;
- Er kan relatief simpel een scherm in elkaar gezet worden;
- Eenvoudige manier van communicatie met de PLC;
- Framework ondersteunt multi-touch.

Uit het interview zijn tevens enkele nadelen naar voren gekomen van Visiwin:

- Geen directe koppeling met de cloud;
- Het framework is geen gedistribueerd platform;
- Microsoft omgevingen zijn niet direct te importeren;
- Weinig duidelijke documentatie beschikbaar;
- Lastige aanpassing van de huisstijl van VisiWin naar eigen stijl;
- Problemen in het framework kunnen niet snel worden opgelost.

Bij VisiWin is het niet mogelijk om gebruik te maken van een cloud. Een paar voordelen van het werken met een cloud infrastructuur is dat het een uitgebreide beveiliging heeft dat voldoet aan de AVG eisen, het is altijd overal toegankelijk en het is efficiënter en effectiever om samen te werken.

Met behulp van Blend kan VisiWin plaatjes inladen om het vervolgens te dynamiseren in de VisiWin ontwikkelomgeving.

De documentatie van VisiWin is voornamelijk in het Duits. Er is geringe documentatie in het Engels. Dit bemoeilijkt ook het oplossen van problemen. Wanneer een probleem zich voordoet bij VisiWin, dan is dit niet direct op te lossen. Deze problemen dienen doorgegeven te worden aan Lenze en dan wordt het doorgegeven aan INOSOFT. Dit leidt tot lange wachttijden.

### 3.3.3 Informatievergaring bedrijven HMI's

VisiWin reageerde niet met antwoorden op de vragen maar bood een privé gesprek aan via Microsoft teams. In dit gesprek werd een presentatie geven over wat VisiWin allemaal kan. Het meeste van deze informatie is al verwerkt in het verslag bij theoretische achtergrond.

Daarnaast werd er een demo gegeven om te laten zien hoe VisiWin te gebruiken is. Dit zodat er een beter beeld is van hoe VisiWin gebruikt kan worden in de praktijk.

Er werd vervolgens gevraagd of er nog overige vragen zijn. Hier kwam alleen naar voren wat de prijzen zijn van VisiWin. De prijzen van VisiWin beginnen bij de engineering licentie waar €3400,- voor betaald wordt. Hier zitten de software en updates bij in. Deze licentie is voor een computer. Voor een runtime licentie wordt €690,- euro betaald en geeft toegang tot 250 tags. Deze kan uitgebreid worden maar zal dus meer kosten.

### 3.3.4 Beoordeling HMI

In tabel 4 is te zien welke scores VisiWin kreeg aan de hand van de opgestelde matrix.

Tabel 4: Beoordeling VisiWin

%	Criterion	Grade	Weighted grade
5	Installatie en koppeling	6	0,3
15	Documentatie	4	0,6
30	Development	8	2,4
15	Kosten	8	1,2
15	Programming paradigma	8	1,2
10	Performance and platforms	8	0,8
10	Ondersteuning	8	0,8
Totaal:			7,3

## 3.4 Ignition

### 3.4.1 Resultaten literatuuronderzoek

Ignition van Inductive Automation is een open source SCADA (Inductive Automation a, 2020). Ignition noemt zich zelf de nieuwe SCADA omdat het oude knel punten oplost van oude scada programma's (Inductive Automotion b, 2020). Ignition is binnen drie minuten te installeren waardoor het snel te gebruiken is (Inductive University a, 2020). Ignition heeft een Inductive University omgeving waarop de werking van Ignition wordt uitgelegd aan de hand van een tutorial (Inductive University b, 2020).

Met Ignition kan er direct op vrijwel elk apparaat een onbeperkt aantal zero-install, full runtime-clients gestart worden. Updates werken realtime. Dit betekent dat wanneer een wijziging wordt aangebracht, dan dient deze door gedrukt te worden naar iedereen. Dit kan gedaan worden zonder enige downtime of dat het systeem offline gehaald moet worden (Inductive Automation c, 2020).

De centrale Ignition Gateway kan zich op verschillende plekken bevinden binnen het bedrijf, te weten: één centrale server, over meerdere servers verdeeld, on-premise, in de cloud, een combinatie van beide of aan de rand van een netwerk (Inductive Automation c, 2020).

Wanneer een extra Web Development Module wordt aangeschaft voor Ignition, kan er direct tegen de Ignition Gateway aan geprogrammeerd worden. Dit wordt gedaan in de programmeertaal Python (Inductive Automation c, 2020).

Ignition servers kunnen gehost worden in de cloud. Ignition zelf bied niet zelf een cloud service aan. In plaats hiervan kan er gebruikt gemaakt worden van Amazon EC2 of Windows Azure (Ignition a, 2020).

Ignition maakt gebruik van een verbindingspad. Het eerste verbindingspunt tussen Ignition en het apparaat is een ControlLogix Ethernet-module. De verbinding met een Allen-Bradley kan uitsluitend gedaan worden met een Ethernet driver. Om te verbinden naar de Allen-Bradley is er alleen een IP-adres nodig (Ignition b, 2020).

### 3.4.2 Interviews

Uit het interview met een gebruiker van Ignition is gebleken dat Ignition de volgende voordelen heeft:

- Het framework is open source;
- Directe koppeling met de cloud;
- Er kan gebruik gemaakt worden van dynamische afbeeldingen;
- Er kan gebruik gemaakt worden van aliasing;
- Het kan gebruikt worden op meerdere besturingssystemen.
- Er kan relatief simpel een scherm in elkaar gezet worden;
- Eenvoudige manier van communicatie met de PLC;
- Framework ondersteunt multi-touch;
- Duidelijke documentatie beschikbaar;
- Eenvoudige aanpassing van de huisstijl van VisiWin naar eigen stijl;
- Problemen in het framework kunnen snel worden opgelost.

Uit het interview zijn tevens enkele nadelen naar voren gekomen van Ignition:

- Het framework is geen gedistribueerd platform;
- Microsoft omgevingen zijn niet direct te importeren;

Ignition laat gebruikers afbeeldingen importeren zodat deze daarna in het programma dynamisch gemaakt kunnen worden,

### 3.4.3 Informatievergaring bedrijven HMI's

Ignition heeft om programmeurs te helpen meer dan 650 gratis video's die bekeken kunnen worden. Deze zijn te vinden in de Inductive University. Dit platform is ontworpen om Ignition te leren, de gebruiker te testen of om een organisatie te trainen. Er is een forum waar je vragen kan stellen aan andere Ignition gebruikers. Dit forum is zeer actief. Er zijn code snippets beschikbaar voor gebruikers. Deze zijn te vinden onder de scripting functies. Hier kun je kiezen wat je wilt gaan implementeren en krijgt hier vervolgens een code snippet voor. Er wordt gebruik gemaakt van Python scripting in Ignition.

De kosten van een de licentie pakketten van Ignition lopen tussen de 10.000 en 30.000 dollar. Bekijk bijlage V om de functies en prijzen per pakket te bekijken. Er kan ook gekozen worden om losse modules aan te schaffen in Ignition. Dit kan handig zijn als een bedrijf een basis pakket heeft aangeschaft maar toch enkele drivers of functies mist.

Ignition heeft al veel zelf ingebouwde componenten die de gebruiker kan gebruiken. De gebruiker kan ook zelf zijn eigen componenten maken met de taal Python. De gebruiker



kan zijn eigen templates en dynamische embedded views maken. Naast de ingebouwde componenten en templates van Ignition is er ook een community driven repository. Deze resources zijn gebouwd door andere Ignition gebruikers en zijn te downloaden en te gebruiken door iedereen.

Wanneer Ignition geïnstalleerd wordt op goede hardware kan het ongeveer 100-500 PLC's, 100.000-500.000 tags en 100-300 gebruikers aan. Door meerdere servers te gebruiken kunnen deze getallen nog hoger oplopen.

### 3.4.4 Beoordeling HMI

In tabel 5 worden de verschillende scores van Ignition beschreven aan de hand van de opgestelde matrix.

Tabel 5: Beoordeling Ignition

%	Criterion	Grade	Weighted grade
5	Installatie en koppeling	9	0,45
15	Documentatie	10	1,5
30	Development	9	2,7
15	Kosten	7	1,05
15	Programming paradigma	8	1,2
10	Performance and platforms	8	0,8
10	Ondersteuning	9	0,9
Totaal:			8,6

## 3.5 Iconics

### 3.5.1 Resultaten literatuuronderzoek

ICONICS is een HMI/ Scada om data van een PLC te visualiseren en te besturen. ICONICS is te gebruiken met een webapplicatie of een desktop applicatie. Met behulp van ICONICS kan een gebruiker zijn hele operatie systeem in kaart brengen in 2D of 3D. Naast het in kaart brengen van zijn operatie systeem kan hij deze ook besturen (ICONICS b, 2020).

ICONICS biedt OPC UA- en BACnet-gecertificeerde visualisatie-, analyse- en mobiele softwareoplossingen met webfunctionaliteit (ICONICS a, 2020).

ICONICS is gebouwd op zijn onderliggende producten namelijk:

- GENESIS64;
- Hyper Historian;
- MobileHMI;

- AnalytiX (ICONICS c, 2020).

Bij het installeren van ICONICS via de website wordt alle documentatie geleverd. Tussen de “App notes” zijn verschillende tutorials die gevolgd kunnen worden.

Met behulp van GraphworX64 wat in het pakket GENESIS64 zit kan er eigen logica geprogrammeerd. Dit wordt met behulp van Jscript.NET gedaan (ICONICS d, 2020).

Er kan gecommuniceerd worden met de cloud door middel van een door ICONICS aangeboden cloud connector, Hiervoor dient er een Azure account zijn, wat eventuele extra kosten met zich meebrengt als het bedrijf dit nog niet heeft. In het standaard Genesis64 pakket zit FrameWorX. FrameWorX kan geconfigureerd worden om met een server data uit te wisselen. De grootste verschillen tussen de cloud connector en FrameWorX is dat FrameWorX met zich zelf kan connecten waar cloud connector dit niet kan. Daarnaast alle handelingen die cloud connector niet zelf kan afhandelen wordt doorverwezen naar FrameWorX (ICONICS e, 2020).

De PLC kan gekoppeld worden met ICONICS door middel van OPC (ICONICS f, 2020).

### 3.5.2 Interviews

Uit het interview met een gebruiker van ICONICS is gebleken dat ICONICS de volgende voordelen heeft:

- Het framework is open source;
- Directe koppeling met de cloud;
- Het framework is een gedistribueerd platform;
- Er kan gebruik gemaakt worden van dynamische afbeeldingen;
- Microsoft omgevingen zijn direct te importeren;
- Er kan gebruik gemaakt worden van aliasing;
- Er kan relatief simpel een scherm in elkaar gezet worden;
- Eenvoudige manier van communicatie met de PLC;
- Framework ondersteunt multi-touch;
- Duidelijke documentatie beschikbaar;
- Eenvoudige aanpassing van de huisstijl van VisiWin naar eigen stijl;
- Problemen in het framework kunnen snel worden opgelost.

Uit het interview is tevens een nadelen naar voren gekomen van ICONICS:

- Er kunnen geen meerdere besturingssystemen aangestuurd worden.

ICONICS is volledig gebaseerd op Microsoft waardoor deze een directe koppeling heeft met Microsoft producten, Een voordeel hiervan is dat er direct geïmporteerd kan worden uit Excel, Word, Paint, enzovoort. VisiWin, Ignition en Zenon hebben deze directe koppeling niet. Een groot nadeel van ICONICS is dan wel dat het op het moment alleen op Windows gebruikt kan worden, waar VisiWin, Ignition en Zenon wel draaien op Linux, Windows en IOS.

### 3.5.3 Informatievergaring bedrijven HMI's

Omdat ICONICS in GENESIS64 scripting afraadt zijn er geen code snippets te vinden. De documentatie vind je onder App notes nadat je ICONICS hebt geïnstalleerd. De licentie kosten zijn schaalbaar. Er wordt alleen betaald voor de functionaliteit die je gebruikt. Het is dus handig als gebruiker om eerst de user requirements op te stellen om dit vervolgens in een ICONICS oplossing om te zetten.

In ICONICS is het niet nodig om je eigen componenten te maken. Het is wel mogelijk om je eigen interface te maken met eigen gemaakte componenten zolang deze voldoen aan de open industrie standaard. Dit heeft te maken met veiligheid.

De performance van ICONICS ligt aan de OPC server die gebruikt wordt door de gebruiker. Het platform heeft een refresh tijd van 50 milliseconde per keer. In sommige industrieën worden er soms meer dan 1000 PLC's aan elkaar geknoopt.

Om ICONICS te gebruiken voor mobiel heeft de gebruiker de "Modilehmi" app nodig. Deze app is te downloaden in de App store. De GraphWorX64 bewerker maakt een dashboard waar de HMI responsive in is. Er hoeft dus geen rekening gehouden te worden met responsiveness.

### 3.5.4 Beoordeling HMI

In tabel 6 wordt beschreven welke scores ICONICS heeft gekregen op basis van de opgestelde matrix.

Tabel 6: Beoordeling ICONICS

%	Criterion	Grade	Weighted grade
5	Installatie en koppeling	8	0,4
15	Documentatie	7	1,05
30	Development	8	2,4
15	Kosten	9	1,35
15	Programming paradigma	7	1,05
10	Performance and platforms	8	0,8
10	Ondersteuning	10	1,0
Totaal:			8,05

## 3.6 Zenon

### 3.6.1 Resultaten literatuuronderzoek

Zenon is een SCADA pakket dat gemaakt is door het bedrijf COPA-DATA (COPA-DATA a, 2020).

Zenon visualiseert in realtime uiteenlopende processen om gebruikers volledig overzicht en controle te geven over de apparatuur, van een individuele machine tot aan het hele bedrijf (COPA-DATA b, 2020).

Zenon heeft een forum waar eventuele vragen gesteld kunnen worden bij problemen. De documentatie van Zenon is te vinden op de website.

Ook biedt Zenon een eigen programmeer interface aan. Deze is beschikbaar vanaf versie 7.10 (nieuwste versie beschikbaar is 8.10). Zenon geeft de optie om eigen add-ins te schrijven in Microsoft Visual Studio of SharpDevelop. Add-ins kunnen gemaakt worden met .NET programmeer talen. De IDE geeft support voor programmeer talen C# en Visual Basics.NET (COPA-DATA c, 2019).

Er is in Zenon een optie waar je met een cloud oplossing werkt. Er kan dan met Microsoft Azure een infrastructuur gecreëerd worden. Op deze manier kan er historische data naar de cloud geschreven worden. Ook kan er data gecombineerd worden uit de cloud om deze op een dashboard te laten zien. Data wordt geëncrypt met Secure Sockets Layer (SSL) encryptie. Om deze cloud oplossing te gebruiken is er licentie nodig van Microsoft Azure (COPA-DATA d, 2018).

De connectie met Zenon verloopt via een driver. Er is een speciale Allen-Bradley driver die geïnstalleerd kan worden. De software requirements die hiervoor staat is dat er Rockwell RSLinx is geïnstalleerd op de computer waar de driver op geïnstalleerd wordt. RSLinx is van Rockwell Automation. De communicatie met de Allen-Bradley verloopt via het TCP/IP protocol (COPA-DATA e, 2020).

### 3.6.2 Interviews

Uit het interview met een gebruiker van Zenon is gebleken dat Zenon de volgende voordelen heeft:

- Het framework is open source;
- Directe koppeling met de cloud;
- Er kan gebruik gemaakt worden van dynamische afbeeldingen;
- Er kan gebruik gemaakt worden van aliasing;
- Er kan gebruik gemaakt worden om meerdere besturingssystemen aan te sturen;
- Er kan relatief simpel een scherm in elkaar gezet worden;
- Eenvoudige manier van communicatie met de PLC;
- Eenvoudige aanpassing van de huisstijl van VisiWin naar eigen stijl;
- Framework ondersteunt multi-touch.

Uit het interview zijn tevens enkele nadelen naar voren gekomen van Zenon:

- Het framework is geen gedistribueerd platform;
- Microsoft omgevingen zijn niet direct te importeren;
- Weinig duidelijke documentatie beschikbaar;
- Problemen in het framework kunnen niet snel worden opgelost.

Zenon laat gebruikers afbeeldingen importeren zodat deze daarna in het programma dynamisch gemaakt kunnen worden,

De documentatie van Zenon is niet op een logische manier ingericht. Doordat de documentatie van Zenon niet duidelijk is, is het lastig om problemen binnen Zenon op te lossen wanneer deze zich voordoen. Er kunnen vragen gesteld worden via een forum, maar dan is het onduidelijk hoe lang het duurt voordat er een antwoord komt met de oplossing.

### 3.6.3 Informatievergaring bedrijven HMI's

Helaas na het sturen van drie mails naar de Zenon support is er geen reactie terug gekomen vanuit Zenon. Hier wordt rekening mee gehouden in de beoordeling van de HMI's.

### 3.6.4 Beoordeling HMI

In tabel 7 worden de verschillende scores van Zenon beschreven aan de hand van de opgestelde matrix.

Tabel 7: Beoordeling Zenon

%	Criterion	Grade	Weighted grade
5	Installatie en koppeling	6	0,3
15	Documentatie	3	0,45
30	Development	7	2,1
15	Kosten	1	0,15
15	Programming paradigma	8	1,2
10	Performance and platforms	8	0,8
10	Ondersteuning	8	0,8
Totaal:			5,8

## 3.7 Samenvatting onderzoeksresultaten

In deze paragraaf wordt een globale samenvatting beschreven van de onderzoeksresultaten om een overkoepelend beeld te geven.

### 3.7.1 Interviews

Hieronder is een tabel te vinden met verschillende functionaliteiten van HMI's. Een groen balkje betekent dat het van toepassing is bij de HMI en rood betekent dat de HMI deze functie niet heeft. De tabel is opgesteld uit de voordelen en nadelen uit de interviews. Vervolgens is onderzocht of het voordeel of nadeel ook op de andere drie frameworks van toepassing zijn. Onder de tabel is enkele toelichting beschreven.

Tabel 8: Uitkomsten informatie over de HMI's van de interviews

	VisiWin	Ignition	ICONICS	Zenon
Het framework is open source				
Bevat een directe koppeling met het web (cloud)				
Het framework is een gedistribueerd platform				
Er kunnen plaatjes gedynamiseerd worden				
Microsoft omgevingen zijn direct te importeren				
In het framework kan gebruik gemaakt worden van aliasing				
Het framework kan gebruikt worden op meerdere besturingssystemen				
Er kan relatief simpel een scherm in elkaar gezet worden				
Er is een eenvoudige manier van communicatie met de PLC mogelijk				
Er is veel en duidelijke documentatie beschikbaar				
Het aanpassen van de standaard stijl naar een eigen stijl is makkelijk				
Problemen kunnen snel opgelost worden in het framework				
Het framework ondersteunt multi-touch				

Zoals in de tabel te zien zijn alle frameworks open source. Dit was een van de harde eisen die MA-IT stelden aan het HMI.

Alle frameworks zijn in staat om plaatjes te dynamiseren. Dat wil zeggen dat plaatjes kunnen bewegen. Dit kan handig zijn bijvoorbeeld bij ovens waar de temperatuur constant verandert. Alle frameworks laten je ook je eigen ontwerpen en plaatjes inladen.

Alle HMI's zijn voorzien van de functie aliasing. Dit houdt in dat er een component aangemaakt kan worden en gekopieerd kan worden. Omdat niet altijd alle componenten hetzelfde moeten zijn kunnen er aparte variabelen gegeven worden aan de gekopieerde component. Het plaatje is dan wel hetzelfde maar de component kan andere gegevens bijhouden dan zijn mede componenten.

Elk HMI heeft beschikking over een drag en drop systeem om snel een scherm in elkaar te zetten. Er kunnen extra componenten gemaakt worden met behulp van de

programmeertalen die de HMI's ondersteunen. Dit kan de gebruiker gebruiken om een meer ingewikkeld naar zijn eisen systeem te bouwen. Elke HMI heeft zijn eigen manier om te connecten met een Allen-Bradley PLC. Dit is of via een TCP/IP protocol (Ethernet) of via een OPC UA protocol. Dit zijn de meest gebruikte protocollen in de PLC wereld. Mochten er meerdere PLC's aanbod komen die tegelijkertijd met dezelfde HMI moeten communiceren is dit ook mogelijk. Ook dit gebeurt bijna altijd via TCP/IP of OPC. Er zijn enkele uitzonderingen met als voorbeeld bij een Beckhoff ADS welke bij VisiWin met een TwinCat protocol benaderd moet worden.

De documentatie van Zenon is niet overzichtelijk. Vaak dient er eerst een forum vraag gemaakt te worden om te vragen waar de plek staat waar het stuk documentatie te vinden is. Ignition heeft zijn documentatie goed op orde. Ignition heeft een aparte website voor documentatie. ICONICS levert zijn documentatie bij het installeren van de HMI en heeft daarnaast een documentatie website waar de verschillende producten en functionaliteiten worden uitlegt.

Bij VisiWin verandert de standaardstijl van VisiWin lastig. De geleverde elementen hebben een eigen VisiWin-stijl, kleine aanpassingen aan deze stijl zorgen ervoor dat de VisiWin-stijl genegeerd wordt en de aanpassingen worden op de basis WPF stijl aangebracht. De VisiWin stijl moet lokaal gekopieerd worden en vervolgens de aanpassing in de 'code' krijgen.

Door de duidelijke documentatie van Ignition en ICONICS kunnen er snel problemen opgelost worden. Mocht de gebruiker niet uit zijn vraag komen dan is de life support die Ignition en ICONICS bieden zeer hoog.

Alle HMI's ondersteunen multi-touch. Dit geeft de gebruiker de gelegenheid om te swipen en te zoomen op zijn HMI scherm.

### 3.8 Beoordeling HMI's matrix

De verschillende HMI's zijn beoordeeld aan de hand van een vooraf opgestelde matrix. Hierin waren verschillende eisen van MA-IT meegenomen. Echter is hierin niet C# meegenomen als harde eis, aangezien dit pas later in het onderzoek een harde eis bleek te zijn.



Uit de opgestelde matrix kwamen de volgende eindresultaten:

VisiWin kreeg een score van 7,3.

Ignition kreeg een score van 8,6.

ICONICS kreeg een score van 8,05.

Zenon kreeg een score van 5,8.

Uit het onderzoek is naar voren gekomen dat de implementatie van de HMI gedaan zal worden in VisiWin ondanks dat ICONICS en Ignition een betere score behaalde.

VisiWin is gekozen omdat de harde eis die MA-IT stelde aan de HMI was dat de HMI ondersteuning moet bieden voor Visual Studio en C#.

## 4 Analyseren

In dit hoofdstuk wordt de analyse van de HMI VisiWin besproken. Ten eerste wordt er een terugkoppeling gemaakt naar de matrix die gebruikt is in het onderzoek. Daarna wordt er beschreven welke programmeertalen er zijn gebruikt bij de PLC. Tenslotte zijn de requirements opgesteld voor het realiseren van de mini-factory.

### 4.1 Matrix

Na het onderzoek is er met behulp van de matrix een conclusie getrokken om de proof of concept te bouwen voor de mini-factory met VisiWin. Met het opstellen van de matrix zijn alle belangrijke functies meegenomen wat het HMI moet ondersteunen. Deze functionaliteiten moeten ook worden meegenomen in de bouw van de applicatie. De twee belangrijkste functionaliteiten waar de opdrachtgever zelf mee kwam waren alarmen en grafieken.

### 4.2 Programmeertalen PLC

Omdat VisiWin gekoppeld moet worden aan de Allen-Bradley van de mini-factory is er onderzocht hoe de koppeling gemaakt kan worden. Er was al onderzocht dat dit mogelijk was via een TCP/IP verbinding maar de vraag was nog hoe. Naast de koppeling met de PLC moeten variabelen ingeladen worden zodat er gecommuniceerd kan worden met de PLC. Omdat het voor de onderzoeker volledig onbekend was hoe een PLC geprogrammeerd wordt, is er eerst gekeken hoe de opbouw van de PLC was. Zo is de PLC opgebouwd in twee programmeertalen die de onderzoeker nog niet kende namelijk ladder en structured text.

#### 4.2.1 Ladder

Ladder ook wel LAD genoemd is één van de programmeertalen dat gebruikt wordt om PLC's te programmeren. De reden dat ladder vaak gebruikt wordt bij het programmeren bij PLC's is omdat het eenvoudig te begrijpen is voor andere programmeurs. Met de structuur van ladder is het PLC programma overzichtelijk. Daarnaast is het overzichtelijk als er een storing is opgetreden en waar deze storing zich bevindt.

#### 4.2.2 Structured text

Structured text wordt samen gecombineerd met ladder om een PLC commando's te geven. Ladder zorgt voor de bit manipulatie waarin structured text vooral dient voor

coördinatie, wiskundige berekening en dataverwerking. Structured text lijkt meer op programmeertalen zoals Java en C#.

### 4.3 Mini-factory

Nadat er duidelijk was hoe de PLC werkt kon er een analyse uitgevoerd worden op de mini-factory zelf. Dit is gedaan door de oude HMI van de mini-factory te analyseren. Zo konden functionaliteiten die in de oude HMI zaten meegenomen worden als eisen voor de nieuwe HMI. Voor de nieuwe HMI was het belangrijk om zoveel mogelijk functionaliteiten van VisiWin te onderzoeken. Het was hierin belangrijk dat het HMI de blokjes van de warehouse rond de mini-factory kon brengen, er alarmen ondersteund werden, er trend grafieken gemaakt konden worden en losse componenten van de mini-factory aangestuurd konden worden zoals een compressor. Daarnaast moesten alle schermen van de mini-factory zichtbaar zijn en alle units moesten handmatig gestopt kunnen worden. Dit moest in zijn geheel kunnen, maar ook per unit.

Omdat er in de oude HMI gebruikt werd gemaakt van dialogs is er voor gekozen om dit ook te doen in de nieuwe HMI. Dit omdat het meteen duidelijk is hoe dialogs kunnen worden toegepast in VisiWin. Dialogs kunnen worden gebruikt om losse componenten van de mini-factory aan te sturen.

## 5 Ontwerpen

In dit hoofdstuk wordt het ontwerpen van de HMI besproken. Ten eerste wordt de methode beschreven. Ten tweede wordt het functioneel ontwerp beschreven en ten slotte het technisch ontwerp.

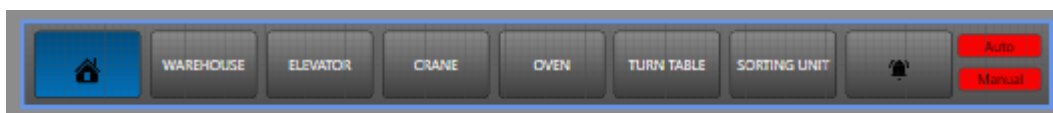
### 5.1 Methode

Om de HMI te ontwerpen is er gebruik gemaakt van Adobe XD. De reden dat er gekozen is voor Adobe XD is omdat de gebruiker makkelijk een echte applicatie kan nabootsen denkende aan de flow en style van de applicatie. Daarnaast is het programma speciaal gebouwd om UI/UX ontwerpen te maken. Voor het ontwerpen van het bestand in Adobe XD zijn er use-cases opgezet en is er een flow diagram gemaakt. Dit zodat er duidelijk naar voren kwam welke eisen belangrijk waren voor de applicatie. Hier kon dan rekening mee gehouden worden in de ontwerpen.

### 5.2 Functioneel ontwerp

Tijdens het realiseren van de HMI is er een functioneel ontwerp gemaakt. Deze dient als een blauwdruk voor de applicatie. Zo kunnen gebruikers die de applicatie niet kennen eenvoudig hun weg vinden over wat de applicatie allemaal kan en welke schermen er zijn.

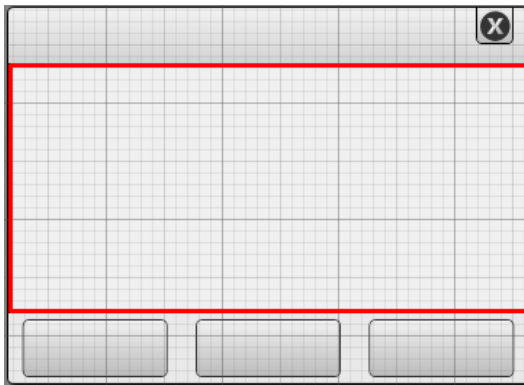
Elke pagina van de applicatie heeft een header en een footer. De header heeft als functie om het meest recente alarm te laten zien (maximaal 1). Hierdoor kan de gebruiker snel zien of er wat mis gaat in de PLC, bijvoorbeeld als er geen signaal wordt ontvangen door een unit. De gebruiker kan door op het alarm te klikken eenvoudig navigeren naar het alarm scherm. De footer heeft vooral als functie om navigatie eenvoudig te maken voor de gebruiker. Doordat deze altijd in beeld staat kan de gebruiker overzichtelijk en snel van scherm wisselen. Tevens is het actieve scherm waar de gebruiker op is blauwgedrukt (afbeelding 3). Naast de navigatie geeft de footer de mogelijkheid om de gehele HMI in auto of manual te zetten. De optie die aan staat zal als groen worden weergegeven waarin automatisch de andere rood zal worden weergegeven.



Afbeelding 3: Footer van applicatie

### Dialog scherm

Het dialog scherm is een pop up scherm dat gebruikt wordt voor het aansturen van specifieke componenten in de mini-factory. Omdat deze componenten altijd hun eigen knoppen nodig hebben die gebonden zijn aan hun eigen variabelen is er voor gekozen om de randen van de dialog hetzelfde te laten (de header en footer). Er kunnen schermen gemaakt worden die in de dialog geladen kunnen worden. Er hoeft bij het ontwerpen van een dialog scherm dus geen rekening gehouden te worden met functionaliteiten zoals het kruisje in de rechterhoek en de close button onderaan. In afbeelding 4 is de rode rand de dialog die de gebruiker zelf maakt. Een voorbeeld van een gemaakte dialog scherm is weergegeven in afbeelding 5.



Afbeelding 4: Dialog scherm



Afbeelding 5: Dialog dat wordt ingeladen

De volgende schermen zijn gemaakt in de applicatie:

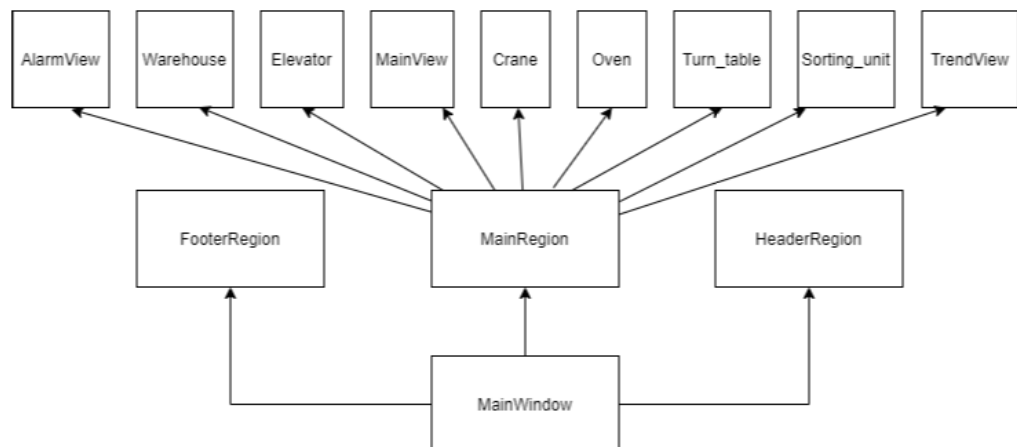
- Hoofdscherm;
- Warehouse scherm;
- Elevator scherm;
- Crane scherm;
- Oven scherm;
- Turn table scherm;
- Sorting unit scherm;
- Alarm scherm;
- Grafiek scherm;
- Dialog scherm.

Voor de volledige beschrijving van alle schermen kan minifactory\_FO.pdf worden geraadpleegd.

### 5.3 Technisch ontwerp

Naast het functioneel ontwerp dat vooral bedoeld is voor gebruikers van de applicatie is er ook een technisch ontwerp opgesteld. Het technisch ontwerp heeft als doel om andere ontwikkelaars een idee te geven hoe de applicatie is opgebouwd. Hierin worden vooral de technische aspecten benoemd zoals welke talen er gebruikt zijn en welke projectstructuur er gebruikt is. Ook worden stukjes code of complexere handelingen zoals het importeren van data vanuit een PLC beschreven. Door het technisch ontwerp door te nemen kunnen andere ontwikkelaars eenvoudig met de bestaande software aan de gang om de software eventueel verder uit te breiden of aan te passen.

Een scherm in VisiWin bestaat uit de MainWindow. De MainWindow gebruikt een footerRegion, headerRegion en een mainRegion. De footerRegion en headerRegion zijn in de mini-factory altijd hetzelfde. Op basis van welk scherm de gebruiker is wordt de MainRegion aangepast. In afbeelding 6 is de architectuurplaat van de applicatie binnen VisiWin te zien.



Afbeelding 6: Architectuurplaat

## 6 Realiseren

In dit hoofdstuk wordt de realisatie van de HMI besproken. Ten eerste wordt de methode besproken. Ten tweede wordt de implementatie van de HMI besproken. Ten derde wordt de opbouw en gebruik van de PLC code beschreven. Ten vierde worden verschillende gerealiseerde functionaliteiten in VisiWin besproken. Ten slotte worden er problemen tijdens de realisatie besproken.

### 6.1 Methode

Met behulp van de ontwerpen zijn de schermen van de HMI gerealiseerd. De implementatie heeft plaats gevonden in VisiWin die als meest geschikt naar voren kwam uit het onderzoek. De helft van de implementatie is vanuit huis gedaan en de andere helft is gedaan op kantoor met de mini-factory ernaast. Dit heeft te maken met het coronavirus dat zich afspeelde in de afstudeerperiode. Daarnaast was er al bestaande code voor de PLC die gebruikt is bij de implementatie van de HMI.

### 6.2 Implementatie HMI

De eerste fase van de implementatie was voornamelijk het bekijken van de functionaliteiten die VisiWin ondersteund, zoals alarmen en grafiek ondersteuning. Ook zijn in deze eerste fase alle schermen gemaakt en de navigatie tussen alle schermen. Omdat de coronatijd afspeelde in de implementatie fase is er voor gekozen om de functionaliteiten zoveel mogelijk thuis uit te vinden en te proberen. Met het gebruik van een simulatie PLC zijn functionaliteiten zoals grafieken met behulp van data vanuit de simulatie PLC getest.

Uit het onderzoek is naar voren gekomen dat de implementatie van de HMI gedaan zal worden in VisiWin ondanks dat ICONICS en Ignition een betere score behaalde. VisiWin is gekozen omdat de harde eis die MA-IT stelde aan de HMI was dat de HMI ondersteuning moet bieden voor Visual Studio en C#. Voor de implementatie is er gebruik gemaakt van een demo versie van VisiWin. De demo versie stelt de gebruiker in staat om maximaal één project te maken met VisiWin. Naast dat er maar maximaal één project gemaakt kan worden kon de HMI maar maximaal één uur runnen achter elkaar. Het project kon na het uur wel afgesloten worden en weer opnieuw in run modus gezet worden.

Om stukken van de schermen te realiseren was er eerst gebruik gemaakt van draw.io. Draw.io is een teken programma waarin verschillende tekeningen gemaakt kunnen worden van flow-chart diagrammen tot klasse diagrammen. Echter kunnen er ook normale tekeningen gemaakt worden zoals voor de mini-factory. Nadat de tekeningen gemaakt zijn in draw.io, zijn deze met een picturebox neer te zetten in VisiWin. Dit bleek al snel problemen op te leveren omdat de gebruiker niet specifieke blokken kan laten verdwijnen zoals dat wenselijk zou zijn bij de warehouse. Op het moment dat een blokje is opgehaald uit de warehouse is het wenselijk dat de gebruiker die de HMI gebruikt het gevoel krijgt dat het blokje ook daadwerkelijk is opgehaald. Op het moment dat de tekeningen in draw.io zijn gemaakt zou de hele warehouse verdwijnen in plaats van alleen het opgehaalde blokje. Omdat VisiWin zelf ook de mogelijkheden geeft om rechthoeken en cirkels te tekenen met behulp van *extensible Application Markup Language* (XAML) is er voor gekozen om dit te gebruiken met als hoofdreden dat er via code aanpassingen gemaakt kunnen worden aan de zichtbaarheid van de rechthoeken. Daarnaast kunnen er methoden gekoppeld worden aan de XAML-elementen. Met behulp van variabelen kan er een onChange methode gemaakt worden om variabelen te laten verdwijnen. Om variabelen beter te begrijpen en er mee te testen moest er eerst meer ervaring opgedaan worden met de bestaande PLC code.

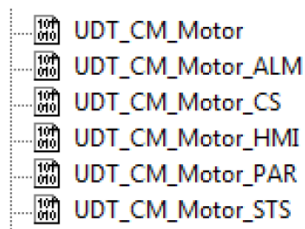
## 6.3 Opbouw en gebruik PLC code

De software in Allen-Bradley is opgebouwd uit verschillende onderdelen. Elk onderdeel wordt apart besproken. De onderdelen zijn Control module (CM), Human machine interface (HMI), unit en process cell.

### 6.3.1 Opbouw Control module

Een Control module is opgebouwd in een Add-on instructie. Deze functioneert als een functieblok. De instructies bevatten netwerken die vervolgens de PLC van boven naar beneden uitleest. De tijd om alle netwerken uit te lezen heet cyclustijd. Deze tijd wordt gemeten in milliseconden en is afhankelijk van het aantal instructies die de PLC moet uitvoeren. Een CM bestaat uit meerdere tags ook wel UDT genoemd. Voordat een PLC programma tot stand komt, moeten eerst alle UDT's zijn aangemaakt. De "UDT\_CM\_Motor" is een verzameling van alle UDT's die onder de motor zijn beschreven (afbeelding 7). In afbeelding 8 is een voorbeeld te vinden van een Add-on instructie. Hierin wordt de motor weergegeven met zijn toegewezen in- en uitgangen die ook in de mini-factory gebruikt worden.





Afbeelding 7: UDT van de CM motor

Name	Usage	Alias For	Data Type
EnableIn	Input		BOOL
EnableOut	Output		BOOL
l_Isolated	Input		BOOL
i_MotorRunning	Input		BOOL
i_Thermal	Input		BOOL
io	InOut		UDT_CM_Motor
mem_CS_SimOff_ons	Local		BOOL
mem_CS_SimOn_ons	Local		BOOL
mem_ToggleSim	Local		BOOL
o_MotorCCW	Output		BOOL
o_MotorCW	Output		BOOL

Afbeelding 8: In- en uitgangen Add-on instructie CM motor

In de instructie (afbeelding 8) is aangegeven waar de UDT aangekoppeld moet worden. De "UDT\_CM\_Motor" zorgt voor communicatie tussen de verschillende lagen. Door een control signal (CS) of HMI signaal kan een CM aangestuurd worden. Met behulp van statussen weet de laag waarmee het CM communiceert of de actie is uitgevoerd.

De UDT's die gemaakt worden in de PLC worden ingeladen in VisiWin als structures ook wel structs genoemd en dienen als een template die hergebruikt kan worden. De struct die is opgebouwd in de PLC is in VisiWin terug te vinden zoals te zien in afbeelding 9.

Search		[Structure elements] Variables\Structures\UDT_CM_Motor		
Name	Order number	Data type	Field Size	Text - English (United States) (103
CS	0	UDT_CM_Motor_CS	0	
HMI	1	UDT_CM_Motor_HMI	0	
STS	2	UDT_CM_Motor_STS	0	
ALM	3	UDT_CM_Motor_ALM	0	
PAR	4	UDT_CM_Motor_PAR	0	

Afbeelding 9: Struct van UDT\_CM\_Motor

De structs kunnen vervolgens aan units gehangen worden. Een voorbeeld is te vinden in afbeelding 10. Hier is te zien dat de UDT\_CM\_Motor hergebruikt wordt voor meerdere units.

Alias	ItemID	Access Rights	Data type	Field Size	Group
PLC.io_U02_M_Hor_ArmElevator	L1.io_U02_M_Hor_ArmElevator	R/W	UDT_CM_Motor	0	Standard
PLC.io_U03_M_Conveyor_In_Outfeed	L1.io_U03_M_Conveyor_In_Outfe	R/W	UDT_CM_Motor	0	Standard
PLC.io_U10_M_Conveyor_CW	L1.io_U10_M_Conveyor_CW	R/W	UDT_CM_Motor	0	Standard
PLC.U5_Transport_Motor01	L1.U5_Transport_Motor01	R/W	UDT_CM_Motor	0	Standard
PLC.U6_Oven_Motor02	L1.U6_Oven_Motor02	R/W	UDT_CM_Motor	0	Standard
PLC.U7_TurnTable_Motor03	L1.U7_TurnTable_Motor03	R/W	UDT_CM_Motor	0	Standard
PLC.U8_Saw_M01	L1.U8_Saw_M01	R/W	UDT_CM_Motor	0	Standard
PLC.U9_OutfeedConveyor_M01	L1.U9_OutfeedConveyor_M01	R/W	UDT_CM_Motor	0	Standard

Afbeelding 10: VisiWin unit structuur

### 6.3.2 Opbouw HMI

Zoals bij opbouw CM beschreven staat kan doormiddel van een HMI signaal een CM aangestuurd worden. Door het inladen van alle variabelen kan VisiWin communiceren met de CM. Variabelen die opgehaald kunnen worden zijn bijvoorbeeld de statussen van de motor. Deze zullen aangeven of het bitje hoog of laag is. De schermen in VisiWin zijn opgebouwd in XAML. Er kunnen aan bepaalde objecten variabelennamen mee gegeven worden. Zoals in afbeelding 11 weergegeven is heeft de checkbox het variabelen aan zich gekoppeld die de status van de elevator kan ophalen (bitje is hoog of laag). Daarnaast heeft deze checkbox een ValueChanged event. Deze kijkt of de gebonden variabele veranderd. Mocht dit gebeuren dan roept hij de functie ElevatorStatusChanged aan.

```
<Rectangle Fill="#FF19FF00" Width="156" Height="78" HorizontalAlignment="Left"
    VerticalAlignment="Top" Name="RectangleElevatorNotReady" Margin="66,419,0,0"/>
<vw:CheckBox Visibility="Hidden" VariableName="Ch2.PLC.io_U01{.STS.ElevatorNotReady}"
    Margin="1072,8,-1072,-8" Background="#FF388CA1" ValueChanged="ElevatorStatus_Change"/>
<vw:Label Width="144" Height="48" HorizontalAlignment="Left" VerticalAlignment="Top"
    Margin="82,437,0,0" Name="labelElevatorReady" Text="Elevator ready"/>
```

Afbeelding 11: XAML VisiWin met checkbox en variabelen

De ElevatorStatusChanged is een functie die vervolgens geschreven kan worden in C#. Zoals in afbeelding 12 is te zien wordt de variabele omgezet naar een string die vervolgens wordt omgezet naar een boolean. Aan de hand van de boolean zal dan de kleur van de rechthoek en de label gemanipuleerd worden zodat de gebruiker van de HMI de juiste informatie krijgt, namelijk de elevator is klaar voor een commando of juist niet.

```
// Method called when the elevator status is changed
private void ElevatorStatus_Change(object sender, VariableEventArgs e)
{
    // gets the new value of the variable
    object objectValue = e.Value;
    // converts the value to a string
    string stringValue = objectValue != null ? objectValue.ToString() : string.Empty;
    bool boolValue = false;
    // Parse the string to a boolean
    bool elevatorNotReady = bool.TryParse(stringValue, out boolValue) && boolValue;
    // if elevator is not ready for new command
    if (elevatorNotReady)
    {
        // Change the color of the rectangle to red and set label that the elevator is not ready
        RectangleElevatorNotReady.Fill = new SolidColorBrush(Color.FromRgb(255, 0, 0));
        labelElevatorReady.Text = "Elevator not ready";
    }
    else
    {
        // Change the color of the rectangle to green and set label that the elevator is ready
        RectangleElevatorNotReady.Fill = new SolidColorBrush(Color.FromRgb(0, 255, 0));
        labelElevatorReady.Text = "Elevator ready";
    }
}
```

Afbeelding 12: Manipulatie van objecten in VisiWin met behulp van C#

Het vorige resultaat kan ook gerealiseerd worden met een PictureBox in VisiWin. Doormiddel van een variabele en een ValueChanged methode aan de PictureBox te koppelen kan er met behulp van een SymbolList een andere achtergrond gezet worden. Het nadeel is dat er als nog code geschreven moet worden om de label te veranderen. Dit kan uiteraard ook voorkomen worden als de label al in het plaatje verwerkt is. Een voorbeeld van een PictureBox en SymbolList is te vinden in afbeelding 13.

```
<vw.PictureBox Width="155" Height="72" HorizontalAlignment="Left" VerticalAlignment="Top" Margin="81, 485, 0, 0"
    VariableName="CH2.PLC.io_U01{.STS.ElevatorNotReady}" ValueChanged="ElevatorstatusChanged">
    <vw.PictureBox.SymbolList>
        <vw.SymbolState Symbol="{DynamicResource on}" Value="0"/>
        <vw.SymbolState Symbol="{DynamicResource off}" Value="1"/>
    </vw.PictureBox.SymbolList>
</vw.PictureBox>
```

Afbeelding 13: PictureBox met symbolList

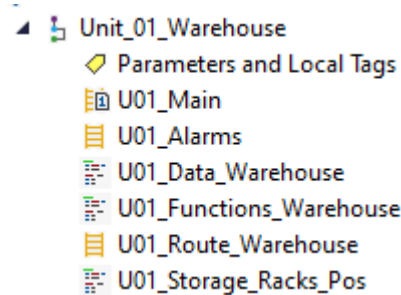
In de vorige voorbeelden werd er alleen gekeken naar een status die veranderd werd met behulp van een variabele. In afbeelding 14 is een voorbeeld te zien hoe een variabele is gebonden aan een ToggleButton. Deze variabele is speciaal gemaakt om aan te roepen vanuit een HMI. Deze geeft namelijk een signaal aan de CM om een commando uit te voeren.

```
<vw:ToggleButton Width="114" Height="84" HorizontalAlignment="Left"
    VerticalAlignment="Top" Margin="438,46,0,0" Text="Pick tray 1"
    Name="tray1Button" VariableName="Ch2.PLC.io_U01{.HMI.Get_Rack_1}"/>
<vw:CheckBox Visibility="Hidden" VariableName="Ch2.PLC.U01_Warehouse_Data{[1].Rack_Empty}"
    ValueChanged="VerBlock1_Change" Margin="1076,-10,-1076,10"/>
```

Afbeelding 14: Togglebutton met een HMI variabele

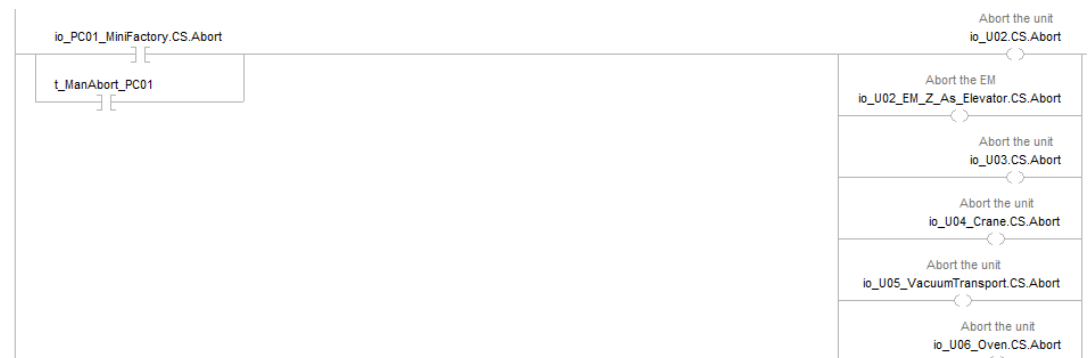
### 6.3.3 Opbouw unit en process cell

Elke unit heeft zijn eigen programma en aan deze programma's kan een prioriteit gegeven worden. Omdat alle units vallen onder de MainTask hebben alle unit de zelfde prioriteit en cyclustijd. De opbouw van unit 1 (warehouse) is te zien in afbeelding 15. Door de structuur van de units is het vinden van problemen tijdens het testen gemakkelijk aangezien je weet in welke unit je moet zoeken. Er zijn totaal 12 units in de mini-factory.



Afbeelding 15: Unit warehouse

De units communiceren via een process cell. De route van de process cell is zo geprogrammeerd dat één commando alle units kan aansturen. Één van deze commando's is het afsluiten van alle taken. In afbeelding 16 is te zien hoe deze code in de process cell is opgebouwd.



Afbeelding 16: process cell commando abort all

### 6.3.4 Hoe controleren of iets werkt

Om snel de HMI te testen is het mogelijk om zelf bitjes te verhogen in de PLC software. Een voorbeeld hiervan is als een krat is opgepakt door de elevator uit de warehouse. De gebruiker wil feedback op de HMI dat het krat verdwijnt. Nu wil de gebruiker ook dat het blokje uiteraard verdwijnt want anders blijft er alleen een zwevend blokje op het scherm staan. In de PLC software kun de gebruiker zelf het scenario naspelen. Zo wordt er bij een krat ophalen eerst de Rack.empty en Tray\_Empty hoog gezet (1). Als de code in VisiWin goed is geschreven zou dit met een Value\_Changed methode de knop en het blokje moeten verdwijnen van het HMI scherm. In afbeelding 17 is te zien hoe de waarden van de warehouse handmatig gemanipuleerd kunnen worden.

▲ U01_Warehouse_Data[1]	{...}
▶ U01_Warehouse_Data[1].Button	1
U01_Warehouse_Data[1].Rack_Empty	0
U01_Warehouse_Data[1].Tray_Unproduced_Product	1
U01_Warehouse_Data[1].Tray_Empty	0
U01_Warehouse_Data[1].Tray_Produced_Product	0
▶ U01_Warehouse_Data[1].Color	1
▶ U01_Warehouse_Data[1].Value_Color	2460
U01_Warehouse_Data[1].RestTray7	0

Afbeelding 17: Warehouse data rack 1

### 6.3.5 Aanpassen PLC code

Naast het begrijpen en kunnen manipuleren van de code is het ook noodzakelijk geweest om kleine stukjes code van de PLC aan te passen. Zo werkte het weghalen van de trays in de HMI niet goed. Dit bleek te komen door de PLC code die de bitjes op de juiste waarde moet zetten bij het resetten van de data in de warehouse (de HMI bepaald aan de hand van het bitje die op 0 of 1 staat of iets wel of niet getoond moet worden op het scherm). Omdat de code het wel deed in de oude HMI was dit vreemd. Dit bleek te komen omdat er in de oude HMI plaatjes over elkaar heen gezet waren wat een snelle oplossing was. Om deze reden is de code herschreven naar de code in afbeelding 18. De code is ook getest voor het oude HMI zodat er daar niks kapot zou gaan.

```

IF NOT io_U01.STS.FirstRun THEN

    // Loop to all the Warehouse data racks
    FOR Reset_Warehouse :=0 TO Reset_Warehouse_Racks DO // loop to all the warehouse data

        U01_Warehouse_Data[Reset_Warehouse].Tray_Unproduced_Product := 1;
        U01_Warehouse_Data[Reset_Warehouse].Tray_Produced_Product := 0;
        U01_Warehouse_Data[Reset_Warehouse].Tray_Empty := 0;
        U01_Warehouse_Data[Reset_Warehouse].Rack_Empty := 0;

    END_FOR;

ELSE

    Reset_Warehouse:= 0;

END_IF;

```

Afbeelding 18: resetten van bits met structured tex)

## 6.4 Functionaliteiten in VisiWin inbouwen

Er zijn verschillende functionaliteiten gebouwd in VisiWin in deze paragraaf worden een paar van deze functionaliteiten uitgelegd.

### 6.4.1 Alarmen

Alarmen dienen als signalen dat er iets mis gaat in de PLC. Alarmen zijn onderverdeeld in verschillende alarm classes namelijk alarm, hint, message en systemalarm. Een normaal alarm is bijvoorbeeld als er een signaal niet goed wordt doorgegeven. Zo kan unit 3 “de conveyor” al een tray klaar hebben staan maar beweegt unit 4 “de kraan” nog niet naar de de tray toe. Er wordt dan een alarm gegeven aan de gebruiker dat de kraan geen signaal heeft ontvangen en daarom dus ook niks doet. Hints alarmen worden gebruikt om de gebruiker een hint te geven wat hij kan doen zoals het aborten van een unit. Een message alarm geeft de gebruiker simpelweg een bericht wat er gaande is en waar niet iets mee gedaan hoeft te worden maar wel handig is om terug te lezen. Systemalarmen zijn alarmen die aangegeven dat er een fout is die een grote impact heeft op het werken van de HMI. Een voorbeeld hiervan is op het moment dat de gebruiker de turn table dialog opent. Omdat VisiWin de structs niet snapt zal er een systemalarm komen die aangeeft dat de struct niet goed is opgebouwd. Dit resulteert in het fllikkeren van de knoppen dat er iets niet klopt.

De alarmen in de applicatie zijn onderverdeeld onder de 12 units. Zo kan een alarm eenvoudig terug gevonden worden. Voorbeelden van alarmen die in unit 3 kunnen voorkomen zijn te zien in afbeelding 19. Hier zijn twee alarmen te zien en kan gelezen

worden als: alarm1 gaat af als er een krat ligt maar er geen signaal is gestuurd naar de kraan en alarm2 gaat af als de conveyor de band wel heeft laten rollen maar geen krat ziet.

Alarm1	(Bit)	Ch2.PLC.io_U03{ALM.NO_switched_but_no_Incoming_Signal}
Alarm2	(Bit)	Ch2.PLC.io_U03{ALM.NO_switched_but_no_Tray}

Afbeelding 19: (Alarmeren unit 3

Om de alarmeren te laten zien in een lijst is wordt er gebruik gemaakt van de AlarmAdapter (Afbelding 20). In de constructor wordt de alarmService aangemaakt die vervolgens blijft kijken of er nieuwe alarmeren zijn en of er verandering zijn aan een de alarmeren.

```
public CurrentAlarmsAdapter()
{
    // Alarm service
    this.alarmService = ApplicationService.GetService<IAlarmService>();

    // Check if alarmService is not null
    if (this.alarmService == null)
    {
        return;
    }

    // Check if alarms have changes if yes call method OnChangeAlarm to update the alarm
    this.currentAlarms = this.alarmService.GetCurrentAlarms2();
    this.currentAlarms.ChangeAlarm += this.OnChangeAlarm;
    this.currentAlarms.ClearAlarm += this.OnChangeAlarm;
    this.currentAlarms.InitAlarm += this.OnChangeAlarm;
    this.currentAlarms.NewAlarm += this.OnChangeAlarm;
}
```

Afbeelding 20: Constructor alarmAdapter

Met behulp van setters en getters worden de alarmeren weergegeven. In afbeelding 20 is ook te zien hoe de methode OnChangeAlarm eruit ziet die aangeroepen wordt vanuit de constructor. Ook is in afbeelding 21 te zien hoe de getter en setter eruit ziet van de alarmeren.

```
private void OnChangeAlarm(object sender, AlarmEventArgs args)
{
    this.SetCurrentValue(AlarmsProperty, new ObservableCollection<IAlarmItem>(this.currentAlarms.Alarms));
}

/// <summary>
/// Gets or sets the alarms to be displayed
/// </summary>
public ObservableCollection<IAlarmItem> Alarms
{
    get { return (ObservableCollection<IAlarmItem>)GetValue(AlarmsProperty); }
    set { SetCurrentValue(AlarmsProperty, value); }
}
```

Afbeelding 21: Setters en getters alarm

### 6.4.2 Trend grafieken

Trend grafieken maken gebruik van de trendAdapter en trendCurveInformation. De trendCurveInformation bevat vooral logica over wat de x en y coördinaten van de lijn moeten zijn waarin de adapter het tekenen van de lijn regelt. Trendgrafieken zijn altijd gebonden aan een PLC variabele. Omdat er voor week 19 niet getest kon worden op de laatste unit is er voor gekozen om random getallen (die gegenereerd worden door een variabele) te laten weergeven in de grafiek. In week 19 is er pas een onderdeel binnen gekomen waardoor unit 12 weer getest kan worden (een relay was kapot die de aansturing doet van de vacuümtransport. Hierdoor kon het blokje nooit unit 12 bereiken). Toen het onderdeel vervangen werd kon de waarde van de Allen-Bradley PLC gebruikt worden om de trendgrafiek te tekenen en is het random getal weggehaald. Met behulp van een ValueChanged methode wordt er gekeken of de grafiek bijgewerkt moet worden. In afbeelding 22 is te zien hoe deze code van het updaten van de lijn in elkaar zit.

```
// On change method that checks if a marker has changed value
private void Marker_MarkerPointsChanged(object sender, RoutedEventArgs e)
{
    this.UpdateCurveInformation(sender as ChartMarker);
}

private void UpdateCurveInformation(ChartMarker marker)
{
    // Check if the marker is not null
    if (marker == null)
        return;

    // Only display the values for the selected marker.
    if (!marker.IsSelected)
        return;

    // Update each markerpoint
    foreach (var markerPoints in marker.MarkerPoints.GroupBy(mp => mp.Curve))
    {
        // Get the key of the current marker
        ICurve actualCurve = markerPoints.Key;
        // Get first markerpoint
        var firstMarkerPoint = markerPoints.First();
        // Check if screen layout is horizontal
        if (marker.Orientation == Orientation.Horizontal)
        {
            // Set x value of the marker
            this.CurveInformations[actualCurve].MarkedXValues = firstMarkerPoint.XValue.ValueFormatted;
            // Set y value of the marker in ascending order
            this.CurveInformations[actualCurve].MarkedYValues = string.Join(
                "; ", markerPoints.OrderBy(mp => mp.YValue.RawValue).Select(mp => mp.YValue.ValueFormatted));
        }
        // If screen layout is vertical
        else
        {
            // Set x value of the marker in ascending order
            this.CurveInformations[actualCurve].MarkedXValues = string.Join(
                "; ", markerPoints.OrderBy(mp => mp.XValue.RawValue).Select(mp => mp.XValue.ValueFormatted));
            // Set y value of the marker
            this.CurveInformations[actualCurve].MarkedYValues = firstMarkerPoint.YValue.ValueFormatted;
        }
    }
}
```

Afbeelding 22: updaten van de trendgrafiek

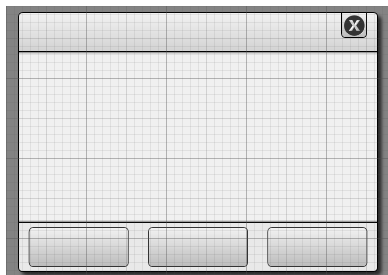


### 6.4.3 Dialogs

Omdat elke dialog een apart scherm heeft met verschillende functionaliteiten is er gekozen om de standaard functionaliteiten in de dialogView te zetten. Vervolgens kan de inhoud van de dialog gezet worden met de show methode. In afbeelding 23 is te zien hoe de compressorView wordt aangeroepen vanuit de crane.cs. Hierin is de eerste parameter de View die ingeladen moet worden, de tweede parameter de titel van de dialog, de derde en vierde parameter zijn om de knop te zetten onder in de dialog. Er kunnen nog meer parameters mee gegeven worden zodat er drie knoppen onder de dialog staan zoals aangegeven in afbeelding 24.

```
private void OpenCompressorDialog(object sender, System.Windows.RoutedEventArgs e)
{
    DialogView.Show("DialogViewCompressor", "Compressor", DialogButton.Custom, leftButtonText: "Close");
}
```

Afbeelding 23: Laten zien van dialogs



Afbeelding 24: standaard layout dialog

## 6.5 Problemen realisatie met VisiWin

Tijdens het realiseren van het HMI kwam er een probleem naar voren. VisiWin ondersteunt niet alle structs van de Control Module waardoor er herhaalde alarmen gegeven worden die invloed hebben op de HMI. Omdat VisiWin bepaalde structs niet snapt kunnen niet alle variabelen gebruikt worden. Dit resulteert automatisch in dat sommige functionaliteiten niet werken omdat de variabelen niet benaderd kunnen worden. Een voorbeeld hiervan is als de turn table dialog geopend wordt. Er wordt aangegeven dat de struct niet leeg mag zijn terwijl deze dat overduidelijk niet is in het VisiWin programma. Omdat de structs zoals ze zijn opgebouwd wel werken in het oude HMI is er een hypothese gesteld dat VisiWin deze structs niet ondersteunt. Er is nog geprobeerd de variabelen te importeren via verschillende manieren maar zonder succes. Hieruit is geconcludeerd dat VisiWin de structs zeker niet ondersteunt. Omdat de koppeling met de HMI pas rond week 11 is gebeurd kwam de onderzoeker hier pas

laat achter (vanwege corona werd er eerst thuis gewerkt en had de koppeling met de HMI geen prioriteit vanuit MA-IT). Met de eerste variabelen waren er echter geen problemen waardoor het de onderzoeker nog niet opviel. Vooral bij het handmatig aansturen van de mini-factory waren er problemen met de structs.

Een ander probleem bij het realiseren van VisiWin was dat er bijna geen documentatie te vinden is op internet. Alle documentatie die er is staat in VisiWin zelf.

Zoekopdrachten op internet hadden dus geen zin. De meeste documentatie die in VisiWin staat is helaas ook nog in het Duits. Bijna alle documentatie die er is gaat over de functionaliteiten van VisiWin zelf. Er zijn simpelweg geen code voorbeelden om VisiWin te gebruiken met behulp van C#. Hierdoor heeft de onderzoeker veel functionaliteiten zelf moeten uitvinden.

Nog een nadeel van VisiWin is dat het altijd herstart moet worden na aanpassingen in de PLC. Omdat de PLC even offline gaat bij het opnieuw downloaden van de code kan VisiWin niet bij de PLC code. Dit resulteert in foutmeldingen dat VisiWin niet bij de variabelen kan. Tevens als de PLC zijn code opnieuw heeft gedownload kan VisiWin nog steeds niet bij de variabelen. VisiWin moet dan eerst opnieuw opgestart worden. Dit is zeer ongebruiksvriendelijk. Vele HMI's hoeven namelijk niet eerst herstart te worden.

Een ander probleem dat naar voren kwam tijdens de realisatie was de ondersteuning van variabelen met REAL datatypes. REAL datatypes zijn zoals software engineers dat kennen floats. Het opvallende hierbij is dat VisiWin de variabelen wel inleest en dus ook in de lijst heeft staan maar vervolgens niet benaderd kunnen worden. Er zijn verschillende oplossingen geprobeerd zoals de bit grote van 32 bits naar 64 bits te veranderen maar zonder succes. Om de elevator positie toch te kunnen bepalen is de REAL omgezet naar een int. Dit is gedaan in ladder.

## 7 Testen

---

In dit hoofdstuk wordt het testen van de applicatie beschreven. Ten eerste wordt de methode uitgelegd. Ten tweede is uitgelegd hoe de gebruikerstesten zijn opgesteld.

### 7.1 Methode

Om de kwaliteit van het product te waarborgen zijn er gebruikerstesten opgesteld. Deze zijn opgesteld om te kijken of het product naar de verwachting van de gebruiker voldoet. Tijdens de implementatie fase zijn verschillende functionaliteiten al getest maar nog niet grondig. Ook om deze reden zijn er gebruikerstesten opgesteld.

De testen zijn in de laatste twee weken geschreven om zo volledig de hele applicatie nogmaals te testen.

### 7.2 Gebruikerstesten

De gebruikerstesten zijn afgenomen door een ander persoon dan de ontwikkelaar zelf. De ontwikkelaar heeft de testen zo geschreven dat de tester precies weet wat hij moet doen. De testen bestaan uit de volgende onderdelen. Als eerst wordt de requirement gegeven die getest gaat worden. In de testgegevens staat wat er van de requirement wordt getest. Zo kunnen er good en bad weather testen geschreven worden voor dezelfde requirement. Een good weather test is wat de gebruiker zou moeten doen om de requirement te laten voldoen. Een bad weather test is als de gebruiker iets probeert te doen terwijl dit eigenlijk niet mag. Het programma zou dit dan ook moeten verbieden. Een voorbeeld hiervan is op het moment dat de compressor van de kraan op auto staat mag de gebruiker deze niet handmatig uit of aan zetten. Deze actie moet afgevangen zijn in de code die geschreven is voor deze requirement.

Sommige testen moeten voldoen aan een pre-conditie. Dit is een conditie die waar moet zijn voordat de requirement getest kan worden. Een voorbeeld hiervan is als de gebruiker een krat wilt ophalen uit de warehouse moet er wel een krat staan in de warehouse.

Taken zijn de stappen die de gebruiker moet uitvoeren om de requirement te testen. Er bestaat een kans dat de gebruiker nog nooit de applicatie heeft gezien en dus niet weet hoe de applicatie in elkaar zit. Door de stappen te volgen zou in principe iedereen de applicatie moeten kunnen testen.

De verwachte resultaten is het resultaat dat de programmeur verwacht op het moment dat een willekeurige persoon de requirement gaat testen. Vervolgens staat in de daadwerkelijke resultaat wat de tester heeft waargenomen. De uitslag zal groen of rood zijn aan de hand van of de test is gelukt of is mislukt.

Als de uitslag rood is van een test worden er aanbevelingen gegeven hoe de test toch succesvol gemaakt kan worden. Een voorbeeld hiervan is dat de kleur sensor soms lichtinval heeft waardoor een verkeerde waarde van de kleur gemeten wordt. Dit resulteert in een verkeerde kleur die gescand wordt. Een oplossing hiervoor zou kunnen zijn dat de kleur sensor wordt vervangen door een betere kleur sensor die wel licht van buitenaf filtert waardoor er nauwkeuriger gemeten kan worden.

Een voorbeeld van een gebruikerstest is te zien in afbeelding 25. Voor het volledige overzicht van alle testen wordt verwezen naar het functioneel ontwerp.

8.0	Als gebruiker wil ik de compressor van de kraan handmatig aan en uit kunnen zetten	Compressor aan of uit zetten	Niet van toepassing	1) De gebruiker is op het scherm van de kraan 2) De gebruiker zet de compressor op manual 3) De gebruiker zet de compressor aan of uit	De compressor gaat aan of uit aan de hand van welke knop de gebruiker heeft aangeklikt	De compressor gaat aan of uit aan de hand van welke knop de gebruiker heeft aangeklikt	
8.1	Als gebruiker wil ik de compressor van de kraan handmatig aan en uit kunnen zetten	Compressor aan of uit zetten terwijl auto nog aan staat	Niet van toepassing	1) De gebruiker is op het scherm van de kraan 2) De gebruiker zet de compressor op auto 3) De gebruiker zet de compressor aan of uit	De compressor verandert niet van status	De compressor verandert niet van status	

Afbeelding 25: Gebruikerstesten

## 8 Conclusie

De hoofdvraag van dit onderzoek was: “Welke open source HMI is het meest geschikt voor MA-IT om te implementeren in haar mini-factory, zodat deze gebruikt kan worden als nieuwe HMI-tool naast de bestaande HMI-oplossingen die al bekend zijn en gebruikt worden?”.

De informatie die uit de interviews en het literatuuronderzoek over de verschillende open source HMI's is verwerkt in een matrix om zo te beoordelen welke HMI het meest geschikt is voor de mini-factory van MA-IT. Deze matrix is gebaseerd op verschillende onderdelen die voor MA-IT van toepassing zijn. De open source HMI's hebben allen een totaalscore gekregen die ligt tussen 1 (het laagste) en 10 (het hoogste). De scores van de verschillende open source HMI's zijn als volgt:

- VisiWin: 7,3;
- Ignition: 8,6;
- ICONICS: 8,05;
- Zenon: 5,8.

Uit de matrix blijkt dat Ignition het meest geschikt is voor MA-IT. Als tweede HMI komt ICONICS het hoogste uit de beoordelingsmatrix. Echter ondersteunen Ignition en ICONICS de programmeertaal C# en het bewerkingsprogramma Visual Studio niet. Omdat de onderzoeker pas later achter de harde eis van C# en Visual studio kwam is hier geen rekening mee gehouden in de wegingen van de HMI's. Omdat dit wel een vereiste vanuit MA-IT is wordt VisiWin het meest geschikte open source HMI.

Voor de realisatie is dus VisiWin gekozen. Tijdens het gebruiken van VisiWin bleek echter bij de import van gegevens richting de mini-factory dat VisiWin niet compatibel was met de structs die worden gebruikt in de Allen-Bradley PLC. Omdat het niet zo moet zijn dat de software in de PLC omgeschreven moet worden omdat VisiWin het niet kan begrijpen is de conclusie getrokken dat VisiWin niet bruikbaar is binnen MA-IT als nieuwe open source HMI.

Dus, om antwoord te geven op de hoofdvraag zou VisiWin het meest geschikt zijn vanwege de harde eisen vanuit MA-IT. Echter is gebleken dat VisiWin niet bruikbaar is voor de mini-factory en dus ook niet bruikbaar is binnen MA-IT.

## 9 Aanbevelingen

In dit hoofdstuk worden als eerste aanbevelingen beschreven voor eventueel vervolgonderzoek. Daarna worden aanbevelingen beschreven voor MA-IT met betrekking tot de gekozen HMI.

### 9.1 Aanbeveling vervolgonderzoek

Tijdens dit onderzoek mist er nog een interview met iemand van Ignition. Voor de betrouwbaarheid van het onderzoek zou er nog een interview gehouden kunnen worden met iemand uit het werkveld van Ignition. In een vervolgonderzoek zou MA-IT kunnen onderzoeken of er meer open source HMI's beschikbaar zijn die werken met C# en Visual studio naast VisiWin en Zenon. Dit onderzoek zou als doel hebben of er een beter platform is met C# en Visual studio dan VisiWin. Er zou ook een onderzoek gedaan kunnen worden of het nut heeft om C# en Visual studio te laten voor wat het is en over te stappen naar een andere programmeertaal en bewerkingsprogramma. Dit onderzoek zou als doel hebben of het qua tijd en geld gunstig is om een andere programmeertaal te leren aan alle werknemers binnen MA-IT.

### 9.2 Aanbevelingen voor MA-IT

Mocht MA-IT zijn eisen aanpassen zou er één aanbeveling zijn. Als de eis van C# en Visual studio vervallen zou MA-IT kunnen overwegen om Ignition of ICONICS te gebruiken. Dit omdat deze HMI's significant hoger scoorden dan VisiWin. Het grootste voordeel van deze systemen is dat er goede documentatie beschikbaar is waardoor problemen in het framework snel opgelost kunnen worden. Een ander voordeel is dat beide frameworks ondersteuning bieden voor een cloud omgeving. VisiWin ondersteunt dit niet. Het enige nadeel van ICONICS kan zijn dat het niet werkt op verschillende besturingssystemen. ICONICS ondersteunt tot heden alleen Windows en dus geen Linux en IOS.

## 10 Behaalde HBO-I competenties

---

In dit hoofdstuk worden de behaalde HBO-I competenties besproken die over de gehele opleiding software engineering aan de orde zijn geweest.

De competenties van software uit de HBO-I richtlijn zijn beheren, analyseren, adviseren, ontwerpen en realiseren. Deze competenties lopen van niveau 1 tot en met niveau 3. Het doel is om aan het eind van het 4<sup>e</sup> jaar minstens drie van deze competenties op niveau 3 te hebben.

Ik ben van mening dat ik de volgende drie competenties op niveau 3 heb behaald:

- Analyseren;
- Ontwerpen;
- Realiseren.

### **Analyseren**

Om de competentie analyseren op niveau 3 te behalen heb ik een groot onderzoek gedaan naar verschillende HMI software pakketten waarmee ik alle eisen van de klant in acht heb genomen. Deze HMI software pakketten zijn naast elkaar gezet om te analyseren welke het meest gunstig zou zijn voor MA-IT. Doormiddel van een uitgebreide matrix is er een keuze naar voren gekomen.

Een ander onderdeel om de competentie analyseren op niveau 3 te behalen is er met personen van het bedrijf MA-IT overleg geweest over de requirements van de HMI over wat deze zou moeten kunnen. Dit is onder andere welke codes gebouwd moesten worden zodat de HMI de juiste onderdelen van de mini-factory kan aansturen.

### **Ontwerpen**

Om de competentie ontwerpen op niveau 3 te behalen heb ik zelf ontwerpen gemaakt voor VisiWin die aansluiten aan de requirements van MA-IT. De requirements waren onder het ondersteunen van grafieken, alarmen en de standaard functionaliteiten die de HMI moet kunnen uitvoeren. Deze ontwerpen heb ik gemaakt in AdobeXD. Om de software te realiseren voor de HMI in VisiWin heb ik gebruik gemaakt van al bestaande componenten die geprogrammeerd waren in de PLC. Hiervoor heb ik een connectie opgezet om met de code van de PLC te kunnen communiceren. Hierdoor konden alle functies in VisiWin geprogrammeerd en getest worden.

**Realiseren**

Om de competentie realiseren op niveau 3 te behalen heb ik zelf een software systeem gebouwd met behulp van VisiWin. Er is rekening gehouden met de ontwerpen die vooraf gemaakt zijn en die de eisen van MA-IT bevatten. Het systeem dat is ontwikkeld in VisiWin sluit aan op de bestaande systemen die al geprogrammeerd waren in de PLC. Er is geprobeerd om zoveel mogelijk functionaliteiten in VisiWin te ontdekken om er achter te komen of dit software pakket wat voor MA-IT is. Daarnaast zijn er een aantal wijzigingen in de PLC code gemaakt.



## 11 Reflectie

In dit hoofdstuk reflecteer ik terug op de afstudeerperiode en stage. Hierin beschrijf ik wat ik heb geleerd en wat nog ontwikkelpunten zijn.

Tijdens mijn afstudeerperiode heb ik veel nieuwe technologieën mogen leren. Ik heb speciaal voor deze afstudeerstage gekozen omdat ik mijn kennis wilde uitbreiden over PLC's. Op het begin was dit wat spannend want ik had nog nooit zelf wat met PLC's gedaan. Om deze reden leek het mij dan ook een goede uitdaging.

Tijdens de stage moest er veel van huis uit gewerkt worden vanwege het Corona virus. Dit vereiste nog meer zelfstandigheid dan dat er normaal wordt verwacht tijdens een afstudeerstage. Bij problemen kon ik niet makkelijk even naar iemand toe lopen wat er toe heeft geleiden dat ik zelf langer door ging om zelf tot een oplossing te komen voor vragen en problemen. De reden hiervan is dat je niet constant iemand wilt lastig vallen via de mail. Dit heeft er een paar keer voor gezorgd dat ik toch zelf tot een oplossing kwam in plaats van dat iemand mij het antwoord gaf. Een voorbeeld hiervan is het uitzoeken van de code leren begrijpen die voor de PLC al geschreven was. Met behulp van internet heb ik stukjes ladder leren lezen. Door de goede basis die gelegd was in school was het begrijpen van structured text makkelijk te doen.

Een verbeterpunt dat ik anders had willen doen was de communicatie. Omdat er veel van huis uit gewerkt werd deden we de daily standups via e-mail. Op een gegeven moment moesten we Trello gebruiken om de voortgang aan te geven. Op het moment dat we Trello gingen gebruiken ben ik gestopt met de daily standups sturen. Niet omdat dit niet meer hoefde maar omdat ik hier zelf vanuit was gegaan. Ik dacht namelijk dat het een vervanging was voor de daily standups. Ik had hier beter kunnen vragen of het nog nodig was om de daily standups te sturen in plaats van zelf deze conclusie trekken.

Een andere reden dat de communicatie beter kon was dat ik tijdens het onderzoek pas er achter kwam dat C# en Visual studio een harde eis was voor het HMI. Ik had beter deze requirements moeten uitzoeken voordat ik aan mijn onderzoek begon. Dit heeft er toe geleidt dat ik eigenlijk twee HMI's heb onderzocht die toch nooit gekozen konden worden.

Als ik terug kijk qua het niveau en hoe ik heb gewerkt tijdens mijn afstudeerperiode ben ik erg trots op mijzelf. Ik heb leren werken met een PLC en een HMI. Dit is iets wat niet

op de opleiding software engineering zelf geleerd wordt. Hiervoor ben ik dus echt in het diepe gedoken. Daarnaast heb ik methodes en structuur gebruikt die ik heb geleerd van school zoals het gebruik van scrum. Hierdoor heb ik mijn voortgang kunnen bewaren over mijn project en kon er bijgestuurd worden waar nodig was.

Vanuit school heb ik een goede basis meegekregen om in de praktijk een project uit te kunnen voeren. Aan school wil ik meegeven dat ik veel heb geleerd in de praktijk. Om deze reden denk ik dat het goed is dat studenten stage lopen bij een bedrijf. Je bent echt op jezelf aangewezen en ziet hoe het er aan toe gaat in het werkveld. De schoolbegeleider heeft mij goed geholpen met zijn adviezen over mijn eindverslag. Met de feedback heb ik uiteindelijk een compleet eindverslag opgeleverd.

## Bibliografie

- AT-Automation. (2019, september 25). *HMI - Wat is een Human Machine Interface?*  
Opgehaald van AT-Automation: <https://www.at-automation.nl/knowledge-base/knowledge-base-article/>
- COPA-DATA a. (2020). *There is always an easier way*. Opgehaald van COPA-DATA: <https://www.copadata.com/en/about-copadata/>
- COPA-DATA b. (2020). *Zenon Software Platform*. Opgehaald van COPA-DATA: <https://www.copadata.com/en/product/zenon-software-platform-for-industrial-automation-energy-automation/>
- COPA-DATA c. (2019). *Zenon Manual Programming Interfaces*. Opgehaald van COPA-DATA: [http://download.copadata.com/fileadmin/user\\_upload/Downloads/Dokumentation/810SP0/ENGLISH/Manual/Programming\\_Interfaces.pdf](http://download.copadata.com/fileadmin/user_upload/Downloads/Dokumentation/810SP0/ENGLISH/Manual/Programming_Interfaces.pdf)
- COPA-DATA d. (2018). *faqs: zenon in the cloud (part 1/2)*. Opgehaald van COPA-DATA: <https://blog.copadata.at/wp/faqs-zenon-in-the-cloud-part-12/>
- COPA-DATA e. (2020). *Zenon driver manual ALLANBNT*. Opgehaald van COPA-DATA: [http://download.copadata.com/fileadmin/user\\_upload/Downloads/Dokumentation/810SP0/ENGLISH/Driver/ALLANBNT.pdf](http://download.copadata.com/fileadmin/user_upload/Downloads/Dokumentation/810SP0/ENGLISH/Driver/ALLANBNT.pdf)
- Geertsma, P. (2013, oktober 11). *Wat is PLC en waarvoor wordt PLC-techniek gebruikt?* Opgehaald van Technisch Werken: <https://www.technischwerken.nl/kennisbank/techniek-kennis/wat-is-plc-en-waarvoor-wordt-plc-techniek-gebruikt/>
- Goede, J. d. (2016, november 11). *Niets meer installeren met Zero Install*. Opgehaald van PCM: <https://pcmweb.nl/artikelen/software/niets-meer-installeren-met-zero-install/>
- ICONICS a. (2020). *Automation for Anything*. Opgeroepen op april 7, 2020, van ICONICS: <https://iconics.com/>
- ICONICS b. (2020). *HMI/SCADA*. Opgeroepen op april 7, 2020, van ICONICS: <https://iconics.com/Solutions/HMI-SCADA>
- ICONICS c. (2020). *Download ICONICS Suite Trial*. Opgeroepen op april 7, 2020, van ICONICS: <https://iconics.com/Downloads/Download-ICONICS-Suite>
- ICONICS d. (2020). *Scripting Basics*. Opgeroepen op april 7, 2020, van Docs ICONICS: [https://docs.iconics.com/V10.96/GENESIS64/Help/Apps/GWX/Scripting\\_Basics.htm](https://docs.iconics.com/V10.96/GENESIS64/Help/Apps/GWX/Scripting_Basics.htm)

- ICONICS e. (2020). *How to Set Up Cloud Connector*. Opgeroepen op april 7, 2020, van Docs ICONICS:  
[https://docs.iconics.com/V10.96/GENESIS64/Help/Com/Arch/Cloud/How\\_to\\_Set\\_Up\\_Cloud\\_Connector.htm#How\\_Cloud\\_Connector\\_Works](https://docs.iconics.com/V10.96/GENESIS64/Help/Com/Arch/Cloud/How_to_Set_Up_Cloud_Connector.htm#How_Cloud_Connector_Works)
- ICONICS f. (2020). *Cyberlogic*. Opgeroepen op april 7, 2020, van ICONICS:  
<https://iconics.com/Resources/Success-Stories/Cyberlogic>
- Ignition a. (2020). *Cloud Based Architecture*. Opgeroepen op april 9, 2020, van Ignition:  
<https://docs.inductiveautomation.com/display/DOC79/Cloud+Based+Architecture>
- Ignition b. (2020). *Allen Bradley Ethernet*. Opgeroepen op april 9, 2020, van Ignition:  
<https://docs.inductiveautomation.com/display/DOC79/Allen+Bradley+Ethernet>
- Inductive Automation a. (2020). *About*. Opgeroepen op april 9, 2020, van Inductive Automation: <https://inductiveautomation.com/about/>
- Inductive Automation c. (2020). *Architectures*. Opgeroepen op april 9, 2020, van Inductive Automation: <https://inductiveautomation.com/ignition/architectures>
- Inductive Automotion b. (2020). *Scada*. Opgeroepen op april 9, 2020, van Inductive Automotion: <https://inductiveautomation.com/scada-software/>
- Inductive University a. (2020). *Installing Ignition on Windows*. Opgeroepen op april 9, 2020, van Inductive University: <https://inductiveuniversity.com/videos/installing-ignition-on-windows/8.0>
- Inductive University b. (2020). *Ignition Overview*. Opgehaald van Inductive University:  
<https://www.inductiveuniversity.com/courses/ignition-overview/8.0>
- INOSOFT a. (2020). *The VisiWin principle*. Opgehaald van INOSOFT:  
<https://www.inosoft.com/en/product/visiwin-principle/>
- INOSOFT b. (2020). *Architecture/ Engineering*. Opgehaald van INOSOFT:  
<https://www.inosoft.com/en/product/architectureengineering/>
- INOSOFT c. (2020). *Communication Drivers*. Opgeroepen op april 9, 2020, van INOSOFT - HMI Unlimited: <https://www.inosoft.com/en/product/process-communication/>
- INOSOFT d. (2020). *Mobile Process Visualization for Smartphones and Tablets with VisiWin*. Opgeroepen op april 9, 2020, van INOSOFT - HMI Unlimited:  
<https://www.inosoft.com/en/product/mobileweb-hmi/>
- KPN.com. (2020, januari 1). *SSL, TLS, HTTPS. Wat is het en hoe werkt het?*  
 Opgehaald van KPN.com: <https://www.kpn.com/zakelijk/blog/ssl-tls-https.htm>
- MA-IT. (2020). *MA-IT: KENNIS & KUNDE CENTRAAL IN EEN TRANSPARANTE ORGANISATIE*. Opgehaald van MA-IT: <https://my-automation.nl/over-ma-it/>

Novotek. (2015, juni 23). *OPC UA uitgelegd*. Opgehaald van Novotek:  
<https://www.novotek.com/nl/oplossingen/opc/opc-ua/>

## Bijlagen

### Bijlage I: Interviewvragen

#### Algemene interviewvragen

- Bij welke bedrijf werkt u en waarvoor gebruiken jullie de huidige HMI?
- Waarom hebben jullie specifiek voor deze HMI gekozen?
- Hebben jullie ook nog andere HMI's overwogen om te gebruiken?
- Hebben jullie nog andere HMI's gebruikt zo ja welke en waarom zijn jullie overgestapt?
- Wat zijn volgens jou de grootste voordelen van het gebruik van deze HMI?
- Wat zijn volgens jou de grootste nadelen van het gebruik van deze HMI?

#### Theoretische interviewvragen (rubrics)

- Hoe lang duurt het installeren van de HMI?
- Hoe is de HMI te koppelen met een Allen-Bradley plc?
- Waar is de documentatie te vinden om aan de slag te kunnen met de HMI? Is er veel documentatie beschikbaar?
- Is er een online forum waar je vragen kan stellen? Waar?
- Hoe werkt de HMI? Moet er code geschreven worden (Zijn er code snippets te vinden)? Kun je componenten slepen?
- Tussen welke bedragen schaalde de HMI bedrijf?
- Is het mogelijk om je eigen componenten te maken? Is dit nodig of kun je het ook makkelijk af met de componenten die de HMI levert?
- Hoe is de performance van de HMI? Heeft het vaak een downtime? Is de HMI snel?
- Hoe regelmatig zijn er updates voor de HMI?
- Is de HMI makkelijk te switchen naar een andere PLC? Allen-bradley->Siemens (is er dan veel omzet werk nodig)?
- Wat zijn de meest belangrijke functionaliteiten van de HMI? (Alarming, Rollen, code generating, scherm resoluties, mobiel, dashboards)?
- Hoe is de learning curve?
- Zou je het platform aanraden?
- Waarom ja/ waarom nee?

## **Bijlage II: Uitwerking interviews**

### **Interview met iemand die werkt met of heeft gewerkt met VisiWin**

*Dit interview is niet telefonisch afgenomen, maar de geïnterviewde heeft de vragenlijst per e-mail beantwoord.*

#### **Bij welk bedrijf werkt u en waarvoor gebruiken jullie de huidige HMI.**

Ik werk bij Mpac, vestiging Wijchen. Wij gebruiken onze HMI voor de fijnere afstelling en besturing van onze inpakmachines.

#### **Waarom heb je specifiek voor deze HMI gekozen?**

Mpac was al een tijdje bezig met een vernieuwde HMI. De eerste iteraties waren van de grond opgebouwd door een PLC-programmeur die er erg geïnteresseerd in was. Hiervoor had hij nog geen ervaring met C# of WPF, maar wel Visual Basic. Tijdens dit project heeft hij zichzelf C# geleerd. Logischerwijs kwam hij na verloop van tijd problemen tegen. Door hun goede band met Lenze kon hij hun om advies vragen. Hierop gaven zij aan dat ze nauw samenwerken met het bedrijf INOSOFT en hun product VisiWin, en dat dit product aan veel van de eisen voldeed van wat Mpac zocht. Na een uitleg/demonstratie/cursus was de PLC-programmeur het er mee eens en sindsdien gebruiken wij het VisiWin framework.

#### **Heb je ook nog andere HMI's overwogen om te gebruiken?**

Voor de nieuwe versie is dus in eerste instantie een zelfgemaakte variatie mbv WPF gemaakt. Voor zover ik weet zijn er geen andere frameworks overwogen.

#### **Hebben jullie nog andere HMI's gebruikt zo ja welke en waarom ben je overgestapt?**

In het verleden gebruikte Mpac HMI's van de gebruikte PLCs, voornamelijk Lenze en AllenBradley. De reden dat we zijn overgestapt is omdat we een algemeen platform wilden hebben voor de HMI. Als bonus heeft VisiWin meer mogelijkheden en een groter scherm.

### **Wat zijn volgens jou de grootste voordelen van het gebruik van deze HMI?**

Na wat instructie kunnen de PLC-programmeurs zelf (relatief simpele) schermen aanmaken of aanpassen. Door onze setup hoeven zij alleen maar het design te maken en de PLC en op te geven.

Voor ons als HMI programmeurs is de hoeveelheid beschikbare onderdelen en de vereenvoudigde communicatie met de PLC een groot voordeel.

### **Wat zijn volgens jou de grootste nadelen van het gebruik van deze HMI?**

Er is niet of nauwelijks documentatie/uitleg van de code, het beetje wat er is is in het Duits.

De geleverde elementen hebben een eigen VisiWin-stijl, kleine aanpassingen aan deze stijl zorgen ervoor dat de VisiWin-stijl genegeerd wordt en de aanpassingen worden op de basis WPF stijl aangebracht. De VisiWin stijl moet lokaal gekopieerd worden en vervolgens de aanpassing in de 'code' krijgen.

Wanneer wij tegen problemen aanlopen die in het VisiWin-gedeelte zitten moeten wij contact zoeken met Lenze, die het probleem kunnen opschalen naar INOSOFT. Dit kan echter lange tijd duren. Een probleem wat wij begin vorig jaar gemeld hebben is nog steeds niet volledig opgelost, ondertussen hebben wij een nieuwe oplossing gekregen alleen dit nog niet volledig getest ivm Corona en andere deadlines.

Overige grote nadelen liggen meer aan de mensen die er mee werken.

### **Theoretische vragen**

#### **Hoe lang duurt het installeren van de HMI?**

Onze installatie bestaat uit 4 grote stappen. Omdat mij niet helemaal duidelijk is welke stap voor jou interessant is zal ik de schatting van iedere stap melden.

#### **Mechanisch bevestigen van het scherm**

Dit is mij niet bekend, tegen de tijd dat het team waarmee ik werk er mee in aanraking komt is dit al klaar.

#### **Installatie software**

De schermen komen standaard geleverd met een Windows Server-versie die zo min mogelijk update. Er moeten nog wat randzaken aangepast worden (firewall ed), kwa installatie hoeft alleen de VisiWin-omgeving en een extra programma voor het



tonen/maken van rapporten hoeft geïnstalleerd te worden. Dit (randzaken & installatie) duurt ongeveer 2-3 uur. We zijn bezig met een backup-image maken die alleen over de bestaande Windows installatie geïnstalleerd hoeft te worden. Als het goed is verkort dit de tijd tot minder dan 1 uur.

#### **Vorbereiden HMI**

Dit stuk kan door de PLC programmeur opgepakt worden, mits de machine geen vreemde speciale onderdelen nodig heeft/gebruikt. Dit kan hij tijdens de opbouw van de machine en het maken van het PLC programma doen. Deze tijdsduur is erg afhankelijk van complexiteit van de machine. Omdat de PLC programmeur dit zelf doet, tijdens de opbouw van zijn PLC programma kan ik moeilijk inschatten hoe lang dit duurt.

#### **Overzetten HMI programma van PLC programmeur naar HMI scherm**

De VisiWin ontwikkelomgeving die wij gekregen hebben heeft hier handige opties voor. Ik weet even niet meer of die omgeving bij INOSOFT of Lenze. Dit kost ongeveer 10-15 minuten.

#### **Hoe is de HMI te koppelen met een Allen-Bradley plc?**

Wij gebruiken een VisiWin ontwikkelomgeving voor de PLC-programmeurs en een plugin voor VisualStudio om hetzelfde effect te bereiken. Deze plugin is beschikbaar in de Visual Studio NuGet-plugin manager.

Daarnaast moet je een driver installeren die de communicatie aankan, wij gebruiken een CIP driver.

In de VisiWin omgeving kan je dan je project koppelen aan de PLC via de driver en een IP-adres. De HMI en PLC moeten dus in hetzelfde netwerk zitten.

**Waar is de documentatie te vinden om aan de slag te kunnen met de HMI? Is er veel documentatie beschikbaar?**

Er is minimale documentatie beschikbaar. Deze staat lokaal op je schijf, maar makkelijker toegankelijk via de VisiWin omgeving. Hierin staan algemene beschrijvingen en ruwe uitleg.

Ook is er een soort API hier beschikbaar, maar het commentaar bij methodes/properties legt niets uit. Het is een beschrijving in de vorm van "deze methode/property heet X en returned Y".

In sommige blokken code staat wat commentaar, maar dit is in het Duits geschreven. Niet persoonlijk mijn sterkste kant, via google translate krijg je een idee, en dan nog mis je de fijne kneepjes.

**Is er een online forum waar je vragen kan stellen? Waar?**

Niet dat ik weet, als wij dringende vragen hebben stellen wij deze aan Lenze. Ook hebben wij een wekelijks overleg met hun, om lopende zaken te bespreken.

**Hoe werkt de HMI? Moet er code geschreven worden (Zijn er code snippets te vinden)? Kun je componenten slepen?**

Het framework is een uitbreiding op WPF en is ingesteld op het MVVM systeem. Het Model is een data-klasse, meestal gebaseerd op PLC informatie. De View is logischerwijs het scherm. Het ViewModel wordt in VisiWin een Adapter genoemd. Het framework werkt op 2 niveaus. Een instap niveau en een (kantoor software) programmeur niveau. In beide gevallen is een basis MVVM in WPF kennis nodig. Voor het instap niveau is een korte cursus met 'trucs', technieken genoeg. De geïnteresseerden kunnen een extra uitleg krijgen voor de structuur erachter. Als programmeur heb je een diepgaande kennis van MVVM in WPF (met name Bindings) nodig.

Het instap niveau kan in de VisiWin editor makkelijk schermen aanmaken via drag&drop van elementen. Via properties en eventueel code-behind kan je (relatief) makkelijk het scherm aanpassen naar jouw wensen. De editor zal zelf de xaml opbouwen/aanpassen obv de acties van de gebruiker. In de VisiWin editor kan je wel gebruik maken van Adapters, maar deze niet maken of aanpassen.

Het kantoor-software programmeur niveau gebruikt meestal VisualStudio met de VisiWin plugin. Hierbij zijn dezelfde onderdelen ook beschikbaar, maar niet meer in drag&drop formaat. Over het algemeen zijn kantoor-software programmeurs

vertrouwer met de designcode en kunnen makkelijker direct de xaml aanpassen voor het gewenste resultaat.

De VisiWin editor past wel zelf de xaml aan, maar niet op de manier die een programmeur zou doen. De editor maakt bijvoorbeeld veel gebruik van margins, ondanks een voorbereid Grid en handige Rows&Columns. Dit kan dan wel weer via properties aangepast worden, maar het is makkelijker om het zelf in 1 keer 'goed' te doen. Daarnaast kan je in VisualStudio wel Adapters aanpassen of maken. Ook kan (moet) je het 'Model' beheren.

Bij ons hebben wij volgende groep verdeling:

PLC programmeurs maken de project specifieke HMI's op basis van de geleverde standaard HMI. Indien nodig maken zij extra schermen voor hun specifieke project en geven aan of dit scherm na verwachting vaker gebruikt zou kunnen worden.

Het HMI Framework Team (kantoor software programmeurs, waar ik bij zit) die de standaard code beheren en waar nodig uitbreiden. Ook leveren wij ondersteuning aan de andere groep.

#### **Tussen welke bedragen schaal de HMI?**

Ons scherm kost ongeveer 2000€, verder kost het ongeveer 500€ om het scherm goed in de machine in te bouwen (ondersteuning/elektra/netwerk). Wij zijn ruwweg 1000€ kwijt per licentie, dit is dus per HMI op een machine en per ontwikkelaar.

Binnenkort stapt Lenze/INOSOFT over op een nieuw licentie systeem, ik weet niet of deze getallen de oude of de nieuwe data is.

#### **Is het mogelijk om je eigen componenten te maken? Is dit nodig of kun je het ook makkelijk af met de componenten die de HMI levert?**

Het is mogelijk en goed te doen. Maar over het algemeen niet nodig. De bestaande componenten dekken ruwweg 85-90% van de functionaliteit. De overige functionaliteit kan vaak iets verbogen worden om alsnog met basis componenten af te vangen.

Wanneer dit niet het geval is kan een programmeur een nieuw onderdeel maken en indien nodig ook presenteren in de VisiWin editor.

#### **Hoe is de performance van de HMI? Heeft het vaak een downtime? Is de HMI snel?**

Dit is erg afhankelijk van het specifieke project. Gemiddeld heeft de HMI een erg goede performance. Maar als je een project krijgt met veel communicatie kanalen vertraagd dit het opstarten, algemene productie (lijkt) hier niet onder te lijden. Een 'probleem' tijdens

de algemene productie is wanneer de gebruiker te veel alarmen uit de geschiedenis wil bekijken (denk aan 100.000-den). De overige performance problemen die wij tot nu toe tegengekomen zijn komen door fouten tijdens de configuratie. Bijvoorbeeld actuele posities toevoegen aan het recept. Dit zorgt ervoor dat iedere keer dat het bijbehorende PLC veld aangepast wordt, het recept bestand opnieuw gemaakt moet worden.

**Hoe regelmatig zijn er updates voor de HMI?**

In de periode die ik nu bij Mpac zit zijn wij van ServicePack 3 naar 9 gegaan, al zijn er een paar overgeslagen. INOSOFT zelf is bezig met de ontwikkeling van de nieuwe versie (7.3), maar daar wil MPac niet naar toe overstappen. Wij zijn de grootste gebruikers van VisiWin (in ieder geval via Lenze) en krijgen op die manier nog wel support van Lenze/INOSOFT voor onze versie.

**Is de HMI makkelijk te switchen naar een andere PLC? Allen-Bradley->Siemens (is er dan veel omzet werk nodig)?**

In theorie is de omzetting erg makkelijk. Installeer de juiste driver, pas het kanaal aan (configuratie richting de PLC) en het geheel zou weer moeten werken.

Maar afgelopen jaar hebben we voor het eerst een Lenze PLC gebruikt ipv AllenBradley. De Lenze PLC is significant anders gestructureerd, hierdoor zijn de paden compleet anders. Dit zorgt ervoor dat veel schermen gecontroleerd moeten worden. Daarnaast kregen wij ook problemen met de communicatie tussen de PLC en de HMI. Het blijkt dat de Lenze PLC anders opstart t.o.v. AllenBradley. Hiervoor hebben wij uiteindelijk 2 communicatie check ingebouwd om op de juiste manier te controleren of er communicatie is tussen de PLC en HMI.

Vermoedelijk geldt hetzelfde voor andere PLC-types.

### **Wat zijn de meest belangrijke functionaliteiten van de HMI? (Alarming, Rolen, code generating, scherm resoluties, mobiel, dashboards)?**

Zoals ik het inschat zijn de Alarmen en PLC-controle het belangrijkste. Onder PLC controle versta ik:

- Het beheren van de Settings, parameters van de machine.
  - o Deze zijn verder onderverdeeld in machine en recept(product)-parameters.
- Tonen/besturen van Actuatoren
  - o Deze mogen alleen aangestuurd worden tijdens handmatige productie/onderhoud
- Diagnostieke informatie
- Tonen van huidige waarden van PLC-bitjes
- In de toekomst komt hier een gedeelte van PredictiveMaintenance bij, het verzamelen van informatie.

### **Hoe is de learning curve?**

Voor instappers zonder ObjectOriented programmeer kennis is de curve redelijk steil. Maar volgens mij goed te doen tot het niveau waar wij ze op willen hebben. De meeste PLC programmeurs waarmee wij werken kunnen het goed oppakken en ook makkelijk nieuwe schermen maken naar hun wensen. Al zitten er ook anderen tussen die dit minder kunnen en graag terug willen naar het oude vertrouwde scherm.

Voor kantoor software programmeurs ligt de curve voornamelijk op het WPF MVVM vlak, werken met Bindings. Het VisiWin gedeelte is goed te doen, de fijne kneepjes komen met ervaring.

Verder is het goed omzetten van de potentieel vage omschrijvingen van medewerkers/klanten voor nieuwe schermen/functionaliiteit, maar dit geldt volgens mij voor elke programmeur.

### **Zou je het platform aanraden? Waarom ja waarom nee?**

Ja, op voorwaarde dat je een soortgelijke structuur opzet zoals Mpac dat heeft. Het systeem is te ingewikkeld voor alleen 'instappers'. Waarschijnlijk krijg je het wel werkend, maar alle problemen die ontdekt worden zullen veel langer duren om op te lossen en lang niet alles zal uit het systeem gehaald kunnen worden. Hierbij is wel een goede communicatie tussen de 2 groepen nodig. Wel zou ik aangeven dat gebruik van meerdere PLCs afgeraden wordt. Focus (indien mogelijk) op 1 type, dan zijn er minder problemen.

## **Interview met iemand die werkt met of heeft gewerkt met Iconics**

### **Bij wat voor bedrijf heeft u gewerkt en waarvoor gebruikte jullie ICONICS?**

Daar kan ik makkelijk over zijn. Ik heb altijd bij Etronics gewerkt en koning willem hartman is een management van Etronics, dus ik heb eigenlijk altijd bij dezelfde baas gewerkt.

### **Hoelang heeft u bij Etronics gewerkt?**

42 jaar.

### **Hoelang gebruiken jullie ICONICS zelf al?**

Sinds 2016. Daarvoor gebruikte wij ICONICS eigenlijk alleen als OEM (Original equipment manufacturer) product. Dus bijvoorbeeld voor ASML (abstract state machine language). Alle machines voor asml worden met ICONICS uitgevoerd die onze controls gebruikt en dat is heel specifiek met het idee van een OEM product. Dat is eigenlijk de basis van ICONICS want ICONICS is al 30 jaar oud. ICONICS komt zelf uit 1986.

### **Was er een specifieke reden dat jullie gebruik zijn gaan maken van ICONICS?**

Psytech mochten we niet meer gebruiken. Wij hebben 20 jaar psytech gebruikt en psytech is gekocht door group schneider en die verbood ons om verder te gaan. Wij doen nog wel services aan psytech want wij hebben natuurlijk servers runnen met psytech maar wij mogen dit product niet meer verkopen. Het is ook wel een verouderd product psytech. Niks negatief over psytech want wij hebben er onze boterham mee kunnen verdienen maar het is wel een hele ouderwetse SCADA. Er zitten niet veel nieuwe dingen in en het wordt een beetje weggedrongen door wonderware. Psytech was trouwens zelf ook een open source SCADA pakket maar dat is het nu niet meer. Het werkt nu met ouderwetse drivers dus dat gaat nu allemaal niet meer werken.

### **Hebben jullie nog getwijfeld tussen andere SCADA pakketten?**

Wij hebben voor ICONICS gekozen omdat wij ICONICS al kenden als HMI dus als tekenpakket en de workbench dat het ons zoveel tijd scheelde in het maken van beeldplaten dat is een heel andere manier van werken. Vandaar dat wij voor ICONICS hebben gekozen.

### **Jullie hebben zoals je al zei eerst een andere HMI gebruikt voordat jullie overstapten naar ICONICS.**

Ja wij hebben er zat gebruikt. Eerst voor mitsubishi daar hadden van bayer een HMI. Heel dedicated. We hebben WisCon gedaan, nou ja je kan het zo gek niet noemen. We

zitten natuurlijk al zolang in die wereld dat wij eigenlijk alles al gebruikt hebben en dan komt gewoon dat boven drijven van wat je het makkelijkste en het beste kan implementeren. Dat is gewoon een feit.

### **Wat vind jij de grootste voordelen van Iconics?**

De grootste voordelen van ICONICS is dat het een open platform is. Dat het een gedistribueerde platform is. Dat het tegenwoordig direct een koppeling heeft met het web. Hij is direct herkenbaar in cloud en alle andere dingen die daarbij nodig zijn. De productiviteit die we hebben. De snelheid waarmee we plaatjes kunnen maken die we kunnen dynamiseren. Alle microsoft omgevingen zijn direct te importeren zonder daar hele rare dingen voor hoeven te doen. Ik kan je bijvoorbeeld een voorbeeld geven: Ik heb nu bezig met een project bij een luchthaven. Daar hebben we luchtbehandelingskasten. Dat is een plaatje van de luchtbehandelingskast in paint en wij kunnen dat plaatje direct importeren en dynamiseren in ICONICS. Je moet het eigenlijk downloaden dat pakket om te zien hoe eenvoudig de opbouw is van ICONICS. Als je gewend bent om met windows om te gaan of met microsoft dan hebben we een workbench en een asset tree en we werken tegenwoordig op basis van asset. Er zijn genoeg voordelen. Wat je veel ziet is aliasing dat wil zeggen dat je iets maakt en dat je het heel eenvoudig kan kopiëren en dat die alles meeneemt op basis van een asset. Dus je maakt 1 asset aan en je kan dan zeggen ik heb 20 van dit soort en dan kun je later misschien nog bepalen daar zit ergens een uitzondering op. Het is echt heel erg gebruiksvriendelijk.

### **Wat zijn de nadelen dan vooral?**

ICONICS is helemaal gebaseerd op microsoft. Dus linux kennen we niet. Dat hebben we wel voor de cloud natuurlijk de koppeling ernaar, maar we zijn gebaseerd op microsoft. Dat vinden sommige mensen wel wat lastig. Het is namelijk zo: alles wat microsoft maakt in de industrie is gemaakt door ICONICS en dat is eigenlijk ook zo met mitsubishi tegenwoordig. Mitsubishi heeft altijd ICONICS gebruikt als zij zelf iets niet konden ontwikkelen op HMI gebied.

### **Het installeren van ICONICS?**

Het is alleen via web en dat hangt af van de snelheid van je machine. Ik heb hier een pc staan die kan het binnen 20 minuten over het hele pakket. Maar dat is ook echt gigantisch met ontwikkel keys en weet ik wat allemaal. Een oude computer trekt er misschien wel een uur aan.



### **Wat houden de pakketten in van ICONICS (genesis64, liberian, mobile etc)?**

Dat is eigenlijk alleen maar marketing gebabbel. Je hebt genesis 64 nodig en daar zit alles in. Wij doen niet aan scripting of andere dingen, dat vinden wij helemaal niks. Dus wat doet ICONICS, die bouwt pakketjes die je kan gebruiken op genesis 64 zoals analytics. Maar dat is niets meer dan een dashboard gebouwd op genesis 64. Je hebt wel eens klanten die zeggen ik wil alleen wat nummertjes uitlezen en ik wil weten wat de kosten zijn, dan heeft ICONICS al wat templates uitgewerkt. Die templates hebben ze vermarkt als zijnde, ik heb hier een product wat je daar en daarvoor kan gebruiken. Het is eigenlijk net iets uitgebreider als een template want er zit soms een historian of een genesis. Maar de basis van het systeem is je hebt de software genesis 64, clients en je hebt dan een historian en daar is eigenlijk alles op gebouwd. Komt er eigenlijk op neer dat je voor jouw project genesis 64 nodig hebt. Over dat mobile, alles is altijd mobile. Wij noemen dat een any glass. Dus je kan het op je smartphone gebruiken hij past dit zelf aan. Je hoeft dus niet een aparte template te maken als je denkt ik ga op een andere pc werken. Hij doet het allemaal op basis van HTML5 zelf. Schermresoluties past die zelf aan.

### **Hoe is de documentatie van ICONICS?**

Je kan gewoon naar iconics.com gaan en daar kan je alles downloaden wat je wilt. En als je alleen maar een HMI wilt dat is dus de Human machine interfacing. Dat wil zeggen eigenlijk zonder bediening of dat soort dingen. Dan kun je graphworks gebruiken dat is voor dashboarding. Het systeem bestaat uit 3 delen: je hebt een workbench zoals wij dat noemen. Daarmee kan je hem besturen daarmee kan je onderliggende data aanpassen, je kan plaatjes maken en bewerken. Er zit ook een library in van 10.000 voorbereekte plaatjes. Wij helpen een klant met dit soort dingen als je helemaal niks weet en je hoeft alleen maar graphworks te gebruiken dan ben je snel running. Als je een beetje technisch onderlegd bent dan snap je dat vrij snel.

### **Zou ik graphworks nodig hebben?**

Dat ligt eraan of je de PLC vanuit de HMI wilt bedienen of wil je dat apart bedienen. Je kan ook zeggen ik wil ook nog vanuit de HMI want dat is de oude SCADA natuurlijk. Dat is gewoon een knopje met die je aan en uit kan zetten en ik wil een minimale en maximale kunnen instellen waarop het proces gaat reageren. Je kan het ook allemaal in de PLC programmeren. Als jij stukjes code moet aanroepen van de PLC dan heb je het volledige SCADA nodig. Dus genesis 64. De andere templates zijn voor mensen die altijd dezelfde fabrieken maken zoals een energie bedrijf. Die kunnen beter een energy

template nemen. Maar normaal als je zo fabriekje hebt dan pak je gewoon genesis 64 en dan kun je alles maken wat je wilt.

#### **Is er een online forum waar je vragen kan stellen?**

Wat wij doen, wij zijn een distributeur van ICONICS en wij helpen iedereen met de support desk. ICONICS zelf heeft een support desk en er zijn natuurlijk gebruikers van ICONICS maar er is niet iets als een modern forum.

#### **Werkt ICONICS zonder code?**

In ICONICS schrijf je geen code. Code is hierin verboden. Koppelingen en drivers bestaan allemaal niet. Wij werken op basis van OPC UA en als wij vreemde dingen tegenkomen dan gebruiken wij een open OPC UA server voor van takamatsu. En dat is omdat wij eigenlijk alleen maar standaard protocollen gebruiken. Als iemand zegt ik wil een dedicated iets wil voor siemens dan zeggen wij, de meeste siemens kletsen tegenwoordig ook cua. Mocht die het niet doen dan hebben wij een mogelijkheid om een oude S5 ook OPC te laten kletsen. Wij kennen dan OPC DA als gene de ingang.

#### **Wat voor OPC koppeling is er bij een allen-bradley nodig?**

Ja hoor wij noemen dat een systemal system. Wij hebben net een heel groot project gedaan waar wij op allerlei onderliggende SCADA's aan elkaar hebben geknoopt en wij daar bovenop gaan zitten. Wij kletsen dan met allerlei verschillende SCADA's zoals wonderware.

#### **Kan iedereen OPC gebruiken?**

OPC is een algemene afspraak. Een foundation. OPC is een afspraak wereldwijd. Je kan bijna wel zeggen dat bijna iedere nieuwe leverancier van software en hardware wel OPC kan communiceren. Alleen als het hele oude dingen zijn dan gebruiken we de koning hartman open server. Wij kennen natuurlijk tegenwoordig van allerlei rest protocollen zoals JSON. Het kan zijn dat je dat wel eens moet aanpassen. Maar dat doen we niet in ICONICS. ICONICS houden we schoon. ICONICS gebruik je als in wat je aangeleverd krijgt. Dus niet scripten.

#### **Hoeveel zou het kosten om ICONICS aan te schaffen?**

Je moet het eigenlijk zo zien. Als MA-IT zou beginnen dan zouden ze een zogenaamd system integrator contract afsluiten. Dat wil zeggen dat zij 3 ontwikkel keys krijgen en ze krijgen daarbij training. Die training is dan bij onze support in tsjechië. Dat kan ook in nederland maar in het contract staat support vanuit tsjechië. Ze moeten daar dan wel

zelf naar toe rijden. Die training is voor iedereen bij MA-IT kan dan gratis. Dat zijn drie ontwikkel keys. 1 hardware key nog ouderwets een met een soort dongeltje en dan 2 software keys. Het is allemaal te downloaden. Maar dat is in ieder geval de basis. De prijs is dan 1964 euro voor het eerste jaar het jaar daarop is het 1264 euro. Dat is wat MA-IT kwijt zou zijn om initieel te beginnen. Je krijgt hiermee alle pakketten van ICONICS alles wat er is. Alle templates enzovoort krijg je tot je beschikking. Dan heb je nog de licenties. Een licentie de goedkoopste licentie is een light versie. Dat zou zijn als MA-IT alleen ontwikkelt en niet voor het bedrijf waar ze aan leveren dan zou je beginnen bij ongeveer 1200 euro met 75 tags. Die 75 tags zijn de adressen die je kan benaderen vanuit de HMI. Die zitten zo in elkaar dat ze alleen tellen als ze ook echt zichtbaar zijn op het scherm. Je kunt alleen wat kopen bij ons als je al wat ontwikkelt heb. Want dan weet je hoeveel tags je nodig hebt. Een tag wordt dus pas geteld als deze zichtbaar is en natuurlijk de alarmen want die wil je altijd controleren. Het zit dus nu zo voor die 1964 krijg je 100.000 tags en dat zijn standaard 4 clients. 4 bedienstationen. Dat is de eerste aanschaf van die 1964 euro. Verder hoeven ze dan helemaal niks te doen. Ze hebben dan training ze hebben dan software. Ze kunnen dan tijdloos alles maken. De bedoeling van ICONICS is dat de licenties het geld opleveren voor ontwikkelkosten.

#### **Kun je zelf componenten maken in ICONICS?**

Ja, je kan een paint tekening dynamiseren en uit elkaar trekken. Dit deeltje A moet gedynamiseerd worden. Dat moet van links naar recht bewegen of van 0 naar 100 procent bewegen of 0 1 worden. Je kan alles dynamiseren dus laten bewegen. Mensen vinden dat mooi. Het kan ook in 3d. Maar of je dat wilt gebruiken is dan maar de vraag.

#### **Hoe zit de performance van de HMI (heeft het downtime, updates)?**

Nee dat heeft die niet. We stellen een bepaalde eis aan de machines die je gebruikt. Dus of je hem nou virtueel gebruikt of op een gewone machine met een gewone server. Daar worden wel wat eisen aan gesteld maar wij zijn natuurlijk gewoon een industrieel product wat betekent dat we hele snelle response tijden moeten hebben. Updates gaat die niet van down. Wij stoppen niet met updates de machine. De updates worden on the fly gedaan. Je hebt dus nooit downtime van updates.

#### **Wat zijn de functionaliteiten die ICONICS heeft?**

Nou als je een functionaliteit vind dat het niet kan dan moet je het even melden. Je zou jezelf echt een heel groot plezier doen door naar [iconics.com](http://iconics.com) te gaan en daar een stukje genesis te downloaden om het gevoel te krijgen. Als je dat wilt en je hebt wat

hulp nodig dan kan ik je daarmee nog wel wat opweg mee helpen. Je zal zien als je er een beetje goed doorheen fietst dan ben je zo op weg.

#### **Hoe is de learning curve?**

Het SCADA leren en de HMI leren is zo gepiept. Het gaat er meer om dat je alles op een verantwoordelijke manier. Je kan zeg maar iets maken en iets maken maar dat zijn nog wel 2 verschillende dingen. Het is handig als je de tips en tricks kent van het product. Dus als wij bijvoorbeeld zeggen begin met assets te maken, een asset is dan bijvoorbeeld een motor met een beveiliging. Als je dat 10 keer tegenkomt maak dan eerst de motor zo efficiënt mogelijk want dan kun je die asset weer opnieuw gebruiken. Zonder dat je iedere keer alles opnieuw hoeft te maken. Sommige mensen maken dat beeldje dan elke keer opnieuw. Je moet nadenken over wat je hebt en hoe je dat gaat gebruiken. Als je iets meerdere keren gebruikt kun je variabelen veranderen waardoor ze toch anders worden.

#### **Zou je het product aanraden? Waarom ja of waarom nee?**

Even heel simpel. Wij hebben heel veel ervaring met verschillende SCADA systemen. Hier heb je een hele moderne omgeving. Het is een Uitblinker voor zijn tijd. Hij is al gekoppeld aan microsoft en andere cloud omgevingen. Bij grote projecten gebruiken wij onder andere SCADA pakketten ICONICS als bovenste laag. Het is een heel flexibel ding. Het is heel makkelijk te koppelen met van alles.

#### **Heb jij toevallig ervaring met VisiWin, Ignition of Zenon?**

Nee dit heb ik niet. Ik ken alleen andere SCADA pakketten zoals wonderware omdat ik ermee moet kletsen. Uiteraard houden wij wel onze concurrenten in de gaten.

#### **Zijn er zelf nog dingen waarvan jij vind dat ik ze moet weten?**

Wees open minded en je kan me altijd alles vragen over ICONICS als je iets niet begrijpt ook uit dit interview. Ik wil er best tijd instoppen. Moet je gewoon doen. Als je nog iets extra moet doen qua programmeren dan verzinnen we daar wel wat voor. Bijvoorbeeld met Augmented reality.

### **Interview met iemand die werkt met of heeft gewerkt met Zenon**

#### **Bij welk bedrijf werkt u en waarvoor gebruiken jullie op dit moment Zenon voor?**

Ik heb het gebruikt. Dat bedrijf was Nedcar in Born voor twee jaar. Die bouwen auto's voor BMW en dat werd toen standaard opgesteld. Dat was toen Siemens Step 7 in combinatie met Zenon als visualisatie pakket. Toen heb ik het gebruikt. Dat was het enige bedrijf waar ik ooit Zenon heb gebruikt.

#### **Ja, ik vond bij Zenon de informatievoorziening op de site niet heel duidelijk.**

##### **Misschien is uw ervaring daar anders in?**

Ja, het is wat je gewend bent. Ik denk dat als je een scada pakketten van Siemens gebruikt hebt en daarna ga je met Zenon werken, dan denk je echt: 'wat is dit?'. In principe kan je er bijna hetzelfde mee, het is gewoon een ander naampje.

#### **Zenon is natuurlijk ook geen traditionele scada. Het is een open source scada, zodat je hem juist kan gebruiken op alle PLC's toch?**

Ja, dat klopt. Als je dan bijvoorbeeld Siemens gebruikt, dan wordt dat aan elkaar gekoppeld. Dat is met Zenon inderdaad niet zo. Stel je wilt variabelen gebruiken dan ben je die 1 voor 1 aan het maken. Dan kan je gewoon linken en dan zeg je tegen het programma 'kijk maar wat je ziet' en dan die gebruik ik. Daar zitten een hoop nadelen aan, maar ook voordelen.

#### **Was er een specifieke reden dat er gekozen is voor Zenon bij het bedrijf?**

Ja, ze maakten auto's voor BMW en BMW had een standaard opgesteld en dat was Zenon. Dus daar waren we afhankelijk van. Maar ik denk dat zij Zenon gebruiken, omdat ze niet afhankelijk willen zijn van dezelfde systemen. Als je alles van Siemens hebt moet je alles van Siemens gebruiken. Als jij iets anders wilt dan anders dan is dat niet mogelijk. Bij Zenon kan je alles maken wat je zelf wilt. Ik denk dat het te maken heeft met dat ze begonnen aan Zenon zijn wat Siemens te bieden had niet uitgebreid was voor wat ze wilden gebruiken en dat ze daarom Zenon zijn gebruiken. Maar dit weet ik niet zeker.

#### **Is er verder geen sprake geweest van scada's zoals Visiwin, Iconics en Ignition?**

Dat weet ik niet.

#### **Wat zijn volgens u de grootste voordelen van Zenon?**

Ik denk dat het grootste voordeel is dat je niet afhankelijk bent van je merk. Dan kan je doen wat je wilt, aangezien het aan open source is verbonden. Wat ik ook gezien heb is

dat Zenon toen wel vooruitstrevend waren met functies zoals multitouch. Volgens mij had Siemens dat toen nog niet eens. Je kan qua visualisatie heel veel, alleen kost het veel tijd. Ik denk dat voor de meeste applicaties die je nu gebruikt ze vrij straight forward zijn. Je hebt wat knopjes en je wilt wat dingen zien maar misschien gebruik je dat allemaal niet. Ik denk dat als je echt een mooi visualisatie pakket wil maken, dan is Zenon wel een mooi pakket. Ik praat wel over toen ik het zelf gebruikte. Daarna is er veel veranderd, dus ik weet niet of het nu nog steeds zo is.

#### **Wanneer was het dat u het gebruikte? Hoe lang geleden ongeveer?**

Ik ben begonnen in 2013 en ben weggegaan in 2018. Ik weet dat BMW toen al de nieuwe standaard hadden. Volgens mij zijn hun begonnen in 2015 met de nieuwe standaard en toen hebben ze Zenon eruit gehaald. Dus dat is wel een tijd geleden toen ze ermee begonnen zijn.

#### **Wat zijn de nadelen van Zenon?**

Een heel groot nadeel is dat je dus niet linkt aan PLC's. Alles wat je zelf wil hebben moet je zelf waar maken. Je moet zelf aangeven in de PLC op welke plekken de variabelen worden gebruikt zodat je het daarna kan mappen. Als je een groot project hebt is het heel veel werk.

Er is heel veel mogelijk in Zenon. Als je bijvoorbeeld een scada pakket hebt en knoppen dan snapt hij dit is een knop en die knop koppel je dan aan een event. Deze knop moet dan bijvoorbeeld ook een variable zetten of het moet naar een ander scherm gaan. Bij Zenon werkt dit niet zo. Je maakt daar een functie aan en die zegt wat hij gaat uitvoeren. Je roept die functie aan en die functie geeft dan aan wat hij moet uitvoeren. Je gaat dan bijvoorbeeld naar een ander scherm en dan verandert deze variable ook. En die functie koppel je dan weer aan een knop. Dit is dus eigenlijk een soort tussenstap die je moet uitvoeren.

Wat het allergrootste nadeel is dat er geen cross reference in zit. Dus je kan bijvoorbeeld niet zeggen: "ik zoek deze variabele en laat mij zien waar deze allemaal gebruikt wordt". Dit kan dus niet. Als je dus een specifieke variabele zoekt dan zoek je je zelf helemaal kapot.

#### **Er moet dus wel geprogrammeerd worden binnen Zenon?**

Er zit wel een drag en drop systeem in. Je kan het programmeren maar dit hoeft niet. Maar het meest gebruikelijke is om het drag en drop systeem te gebruiken.

#### **Het installeren van Zenon hoe gaat dat?**

Ik heb het zelf maar een keer geïnstalleerd maar volgens mij ging dat wel redelijk makkelijk.

#### **Hoe ging het koppelen met de PLC (Allen-bradley)?**

Het is eigenlijk heel simpel. Je koppelt het aan een ip-adres ongeacht het merk en je kan dan een driver aanmaken. Ik heb dit toen gedaan bij een hele grote hal en hier zaten iets van 80 PLC's in. Heel vaak heb je info nodig van een andere PLC bijvoorbeeld het installatie stuk. Je kan dan een driver aanmaken, tikt het ip-adres in en dan kun je variabelen uit die PLC halen. Dat is namelijk makkelijker dan een PLC door te sturen naar een scada waar de editor op draaide. Dat deed je dan gewoon vanuit een driver aanmaken, bijvoorbeeld PLC A naar PLC B naar PLC C en dan kun je vanaf deze drie alles visualiseren. Het protocol is tcp-ip.

#### **Is er veel documentatie over Zenon te vinden?**

Ik weet dat er op youtube wat basics te vinden zijn zoals hoe maak je een variabelen aan. Als je met gewone scada pakketten hebt gewerkt dan red je het hier ook wel. Het werkt alleen allemaal net iets anders. Ik heb het groote voordeel gehad dat wij werkten voor BMW dus wij kregen de cursus van BMW. BMW heeft een eigen Zenon cursus gemaakt hiervoor en die hebben hun eigen standaard. Je kreeg dan een stappenplan van hoe je alles moet uitvoeren. Misschien dat ik daar nog wat van heb dat ik dat met je kan delen.

#### **Is het gebruik van Zenon te doen voor iemand die nog nooit heeft gewerkt met een HMI?**

Dat zou wel te doen moeten zijn. Helemaal als je nog geen andere scada pakketten hebt gebruikt. Je hebt dan namelijk geen vergelijkingsmateriaal.

#### **Zijn er ook online forums waar je vragen kan stellen?**

Niet dat ik weet. Ik heb dat nooit hoeven te gebruiken. Er zit wel in de editor een helper. Die geeft suggesties van wat je kan doen. Deze staat constant open. Je ziet dan iets van een regeltje staan en die geeft dan aan van wat het is en daar kun je op doorklikken en dan ga je naar de volledige help.

#### **Wat zijn de kosten ongeveer voor een Zenon licentie?**

Dat zou ik niet durven zeggen. Voor ons werd alles geregeld door BMW dus daar heb ik niet echt mee te maken gehad.

### **Kun je zelf componenten aanmaken in Zenon?**

Je kan soort van je eigen templates maken. Je kan een eigen library aanmaken en je kunt dan zelf van alles tekenen. Je kunt symbolen dan erin slepen en linken. Als je het goed opzet kan je heel snel programmeren in Zenon. Ook qua schermen kun je snel wat neerzetten.

### **Hoe is de performance van Zenon? Heeft het vaak een downtime door bijvoorbeeld updates?**

Ik heb zelf nooit problemen gehad hiermee. Ja af en toe dat je de editor even opnieuw moest opstarten maar verder dan ook niet. Wij hadden zeg maar een hele grote hal met 80 PLC's. Elke PLC had dan weer zijn eigen computer met een HMI en daar draaide dan een editor op. Elke PLC had ook 1 project en die stonden dag en nacht open. Hierdoor crashde de editor wel eens. Ik denk dat ik tijdens het werk dat ik daar heb gedaan misschien 2 keer een update heb gezien en dat is dus in de tijd dat ik er 5 jaar werk.

### **Als ik Zenon van een Allen-bradley wil overzetten naar een Siemens is dit veel omzet werk?**

Ik durf niet met zekerheid te zeggen of je zo maar kan switchen tussen PLC's. Het lijkt me dat je wel iets moet omzetten aangezien niet alle PLC's hetzelfde zijn. Dat ligt er dus aan hoe een Allen-bradley werkt en hoe een Siemens werkt.

### **Zijn er functionaliteiten die er niet waren waarvan je dacht dat was wel handig dat ze het wel hadden (alarming, rollen, mobiel, , resoluties, dashboards)?**

Volgens mij moet dit allemaal kunnen. Je kan als het goed is ook alarming doorsturen naar een email adres. Dus als er dan wat fout gaat krijg je het meteen op de email. Er zullen natuurlijk ook wel dingen bijkomen. Dingen zoals resoluties kon je in ieder geval wel aanpassen. Het is makkelijk schaalbaar.

### **Hoe is de learning curve van Zenon?**

Ik denk als je al een keer met een scada pakket gewerkt hebt dat je dit wel redelijk makkelijk oppakt. Het werkt nou net allemaal iets anders. Ik was er snel doorheen. Ik denk dat het je wel redelijk makkelijk af gaat.

### **Zou je het platform aanraden? Waarom ja of waarom nee?**

Ik zou het pakket aanraden maar ik ken het pakket ook. Ook zal ik het jou aanraden omdat je natuurlijk naar een open source scada zoekt. Maar het is ook niet verkeerd



om een scada te nemen die bij de plc behoort. Je kan alles met dit scada pakket wat je volgens mij nodig hebt.

**Zijn er zelf nog dingen waarvan je denkt dit zou handig kunnen zijn voor mij?**

Ja, ik denk als je het eenmaal gaat gebruiken dan komen er sowieso wel vragen naar boven en dan kun je mij wel gewoon bereiken. Ik kan je vragen dan wel beantwoorden.

**Ik zal het laten weten als ik voor dit scada pakket gekozen heb en wil jou bedanken voor je tijd om dit interview te houden.**

Geen dank en ik hoor het wel.

## Bijlage III: Informatievergaring bedrijven HMI's

### Email VisiWin

Dear Sir/Madam,

For my graduation project I am researching four different HMI scada platforms for my company. The reason for the research is that they want to quit using traditional scada platforms. The final result of the research should be that they will use one of the HMI scada platforms in their own company. VisiWin is one of the scada's I am researching. To get the best results I wanted to get some answers from the companies themselves. I am pretty unfamiliar with HMI scada's so hopefully you can help me with those questions. Under every question I have listed what I have found in a small summary.

The questions are:

1. How easy is it to install VisiWin? Is it possible to link it with an Allen-Bradley PLC. If yes how?

I saw you can connect VisiWin with tcp/ip with CompactLogix.

2. Is there a lot of documentation for programmers? Also is there an active community where you can ask questions about your work? Are there code snippets available?

I found some documentation in the main manual of VisiWin on how to use things like variables etc. There is not an active forum that I found. Also the manual does not contain any code snippets.

3. What are the costs of an VisiWin license and what do you get from it?

I could not find anything about prices on the website.

4. There are components you can use from VisiWin itself, can you also make your own components? How easy is this?

In a video of VisiWin I saw that you could use components of VisiWin but also program your own components.

5. How is the performance of VisiWin? Does it work on multiple PLC's?

If I am correct VisiWin is an open scada so it should work on all PLC's.

6. What features does VisiWin support? Thinking about alarming, roles, code generating, screen resolutions, mobile, dashboards.

I found some information about alarming. In the promotion video of VisiWin all the other things like mobile etc were also mentioned.

Hopefully you can give me some more answers. The chosen HMI scada platform will be tested on an Allen-Bradley PLC to get a better picture of how the scada works.

If you need any information from my side please let me know!

I am looking forward to your reply!

Met vriendelijke groet / With kind regards,

Lars Wicherson

### Email Ignition

Dear Sir/Madam,

For my graduation project I am researching four different HMI scada platforms for my company. The reason for the research is that they want to quit using traditional scada platforms. The final result of the research should be that they will use one of the HMI scada platforms in their own company. Ignition is one of the scada's I am researching. To get the best results I wanted to get some answers from the companies themselves. I am pretty unfamiliar with HMI scada's so hopefully you can help me with those questions. Under every question I have listed what I have found in a small summary. The questions are:

**1. How easy is it to install Ignition? Is it possible to link it with an Allen-Bradley PLC. If yes how?**

I saw you can connect Ignition with Ethernet. Are there any other options?  
There are videos on youtube on how to use Ignition.

**2. Is there a lot of documentation for programmers? Also is there an active community where you can ask questions about your work? Are there code snippets available?**

There are some webinars on the website. Also there seems to be an active forum. Ignition uses Python if I am correct. Found some easy code snippets for different things like making components.

**3. What are the costs of an Ignition license and what do you get from it?**

Pretty clear, stated on the website. All the other HMI don't show this on their website.

**4. There are components you can use from Ignition itself, can you also make your own components? How easy is this?**

There are a lot of components you can use from Ignition and change them.

**5. How is the performance of Ignition? Does it work on multiple PLC's?**

If I am correct Ignition is an open scada so should work on all PLC's.

**6. What features does Ignition support? Thinking about alarming, roles, code generating, screen resolutions, mobile, dashboards.**

Most of these are listed on the website. What is the best way if I want to learn to use Ignition by myself?

Hopefully you can give me some more answers. The chosen HMI scada platform will be tested on an Allen-Bradley PLC to get a better picture of how the scada works.

Ignition is definitely my favorite at the moment since most of the info is on the website!  
If you need any information from my side please let me know!

I am looking forward to your reply!

Met vriendelijke groet / With kind regards,  
Lars Wicherson

## Email ICONICS

Dear Sir/Madam,

For my graduation project I am researching four different HMI scada platforms for my company. The reason for the research is that they want to quit using traditional scada platforms. The final result of the research should be that they will use one of the HMI scada platforms in their own company. ICONICS is one of the scada's I am researching. To get the best results I wanted to get some answers from the companies themselves. I am pretty unfamiliar with HMI scada's so hopefully you can help me with those questions. Under every question I have listed what I have found in a small summary. The questions are:

### **1. How easy is it to install ICONICS? Is it possible to link it with an Allen-Bradley PLC. If yes how?**

I saw success stories of ICONICS with Allen-Bradley users so probably yes. I could not find how to connect to an Allen-Bradley though.

### **2. Is there a lot of documentation for programmers? Also is there an active community where you can ask questions about your work? Are there code snippets available?**

I could only find some documents on brochures. Also I could not find a forum where you can ask questions. There is a support chapter where you can ask questions though. What is the best way to learn if you want to start using ICONICS?

### **3. What are the costs of an ICONICS license and what do you get from it?**

I Couldn't find anything on license prices. I am currently downloading the trial version.

### **4. There are probably some components you can use from ICONICS itself, can you also make your own components? How easy is this?**

I saw some youtube tutorials on components but could not find making your own components.

### **5. How is the performance of ICONICS? Does it work on multiple PLC's?**

If I am correct ICONICS should be an open scada. So it should work on multiple PLC's.

### **6. What features does ICONICS support? Thinking about alarming, roles, code generating, screen resolutions, mobile, dashboards.**

On the main site most of these above were named and explained pretty good. Are there tutorials you can use to get a better understanding on how to use each of those in your own project?

Hopefully you can give me some answers. The chosen HMI scada platform will be tested on an Allen-Bradley PLC to get a better picture of how the scada works.

If you need any information from my side please let me know!

I am looking forward to your reply!

Met vriendelijke groet / With kind regards,

Lars Wicherson

### Email Zenon

Dear Sir/Madam,

For my graduation project I am researching four different HMI scada platforms for my company. The reason for the research is that they want to quit using traditional scada platforms. The final result of the research should be that they will use one of the HMI scada platforms in their own company. Zenon is one of the scada's I am researching. To get the best results I wanted to get some answers from the companies themselves. I am pretty unfamiliar with HMI scada's so hopefully you can help me with those questions. Under every question I have listed what I have found in a small summary. The questions are:

**1. How easy is it to install Zenon? Is it possible to link it with an Allen-Bradley PLC. If yes how?**

I tried making an account to install Zenon but it didn't let me install after. So I had no clue on this topic.

**2. Is there a lot of documentation for programmers? Also is there an active community where you can ask questions about your work? Are there code snippets available?**

I saw there was a youtube channel with some tutorials. Also found a forum where you can ask questions. I couldn't really find any snippets.

**3. What are the costs of a Zenon license and what do you get from it?**

I Couldn't find anything on license

**4. There are probably some components you can use from Zenon itself, can you also make your own components? How easy is this?**

I Will watch some of the youtube tutorials to get a better understanding on this but if you have extra information please let me know.

**5. How is the performance of Zenon? Does it work on multiple PLC's?**

If I am correct Zenon should be an open scada. So it should work on multiple PLC's.

**6. What features does Zenon support. Thinking about alarming, roles, code generating, screen resolutions, mobile, dashboards.**



On the main site all these above were named, but I couldn't find any more information about these subjects. Is there documentation about these things to be found?

Hopefully you can give me some answers. The chosen HMI scada platform will be tested on an Allen-Bradley PLC to get a better picture of how the scada works. If you need any information from my side please let me know!

I am looking forward to your reply!

Met vriendelijke groet / With kind regards,

Lars Wicherson

## Bijlage IV: Uitwerking informatievergaring HMI's

Antwoord Ignition

Hi Lars,

Thanks again for reaching out with questions. One of our Sales Engineers was more than happy to answer them. Please find the responses, in blue, below:

1. How easy is it to install Ignition? Is it possible to link it with an Allen-Bradley PLC. If yes how?

I saw you can connect Ignition with Ethernet. Are there any other options?

There are videos on youtube on how to use Ignition.

Ignition is available to download from our website. Installation takes about 3 minutes.

<https://inductiveuniversity.com/videos/installing-ignition-on-windows/8.0>

We have an Allen-Bradley Ethernet driver. Connecting only requires an IP address. We do not support other options outside of ethernet.

<https://inductiveuniversity.com/videos/connecting-to-micrologix/8.0>

2. Is there a lot of documentation for programmers? Also is there an active community where you can ask questions about your work? Are there code snippets available?

There are some webinars on the website. Also there seems to be an active forum.

Ignition uses Python if I am correct. Found some easy code snippets for different things like making components.

We have an online user manual. <https://docs.inductiveautomation.com/>

Free Online Video Training with over 650 videos. <https://inductiveuniversity.com/>

And our online forum is a great place to ask questions.

<https://forum.inductiveautomation.com/>

Code snippets can be found in the documentation for each of our scripting functions.

<https://docs.inductiveautomation.com/display/DOC80/Scripting+Functions>

3. What are the costs of an Ignition license and what do you get from it?

Pretty clear, stated on the website. All the other HMI don't show this on their website.

Yes, we are very transparent about our pricing. Ignition is server based software so you buy a license per server you want to install it on. The cost of that server is determined by which modules you install on it. Each module provides different optional functionality.

Modules can be purchased at a carte or in bundles. All modules support unlimited device connections, tags, and users. Pricing can be found here:

<https://inductiveautomation.com/pricing/ignition>

The licenses are perpetual meaning you buy it once and you own it. Option support contracts are available for an annual cost. Support cost will be a percentage of the total cost of software you have purchased.

<https://inductiveautomation.com/pricing/ignition/choose-support>

4. There are components you can use from Ignition itself, can you also make your own components? How easy is this?

There are a lot of components you can use from Ignition and change them.

Yes, Ignition has many pre-built components. We also give the user the ability to create their own templates and dynamic embedded views.

<https://inductiveuniversity.com/videos/about-templates/8.0>

<https://inductiveuniversity.com/videos/embedded-view/8.0>

We also have a community driven repository of pre-built resources that you can download and utilize. <https://inductiveautomation.com/exchange/>

5. How is the performance of Ignition? Does it work on multiple PLC's?

If I am correct Ignition is an open scada so should work on all PLC's.

Ignition is hardware agnostic and supports connections to many different types of PLCs. When installed on proper hardware a single Ignition server can handle approximately 100-500 plcs, 100,000-500,000 tags, and 100-300 concurrent users. By adding additional servers into a scale-out architecture, Ignition can support thousands of PLCs, millions of tags, and thousands of concurrent users.

6. What features does Ignition support? Thinking about alarming, roles, code generating, screen resolutions, mobile, dashboards.

Most of these are listed on the website. What is the best way if I want to learn to use Ignition by myself?

Our free online training videos, Inductive University, are the best way to learn Ignition.

<https://inductiveuniversity.com/courses/whats-new-in-ignition-8/8.0>

You can also check out our online demo project. <https://demo.inductiveautomation.com/>

That project is also broken out by features to give a quick look at the kinds of things people do with Ignition.

The features are too many to list out here but this page gives a good overview.

<https://inductiveautomation.com/scada-software/>

2e antwoord

Hi Lars,

Here are customer resources that can help you with your project:

<https://www.inductiveuniversity.com/courses/whats-new-in-ignition-8/8.0> - Videos on what Ignition can do

<https://forum.inductiveautomation.com/> - Forum

<https://docs.inductiveautomation.com/display/DOC80/Welcome> - User Manual

We highly recommend to go through Inductive University along side with the User Manual. This will answer most of your questions.

If you are looking for a temp license to use for your project, we highly recommend that you reach out to our Sales Dept where they can provide you more details. I know that college students may have a practice license but I would inquire about that with the Sales Team at [accountservices@inductiveautomation.com](mailto:accountservices@inductiveautomation.com).

Good luck on your project!

Cheers,

## Antwoord ICONICS

Dear Lars Wicherson,

Thank you for contacting ICONICS.

See here after our answers marked with [ICONICS]

If you have any questions afterwards, please feel free to ask.

How easy is it to install ICONICS? Is it possible to link it with an Allen-Bradley PLC. If yes how?

[ICONICS]

Go to the [iconics.com](http://iconics.com) web site, download the package and select setup.exe on a windows 64 bit computer.

I saw success stories of ICONICS with Allen-Bradley users so probably yes. I could not find how to connect to an Allen-Bradley though.

[ICONICS]

AB devices are connected through OPC.

Is there a lot of documentation for programmers? Also is there an active community where you can ask questions about your work? Are there code snippets available?

[ICONICS]

In a professional platform like ICONICS GENESIS64 there are not "code snippets" you configure using editors and deploy to web, head mounted devices, etc. without having to code.

Hitting the "F1" function key opens up the documentation center in your ICONICS installation.

I could only find some documents on brochures. Also I could not find a forum where you can ask questions. There is a support chapter where you can ask questions though.

What is the best way to learn if you want to start using ICONICS?

[ICONICS]

In your ICONICS installation you will find an "App notes" section with hundreds of quick starts targeted to address your goals

What are the costs of an ICONICS license and what do you get from it?

[ICONICS]

Licensing is scalable. You only pay for the functionality you need to address your project goals. So first start with a user requirement spec. and then determine which and how you would like to turn these in an ICONICS solution.

I Couldn't find anything on license prices. I am currently downloading the trial version.

There are probably some components you can use from ICONICS itself, can you also make your own components? How easy is this?

[ICONICS]

There's no need to make your own components. However, you can interface with your own components if you base these on open industry standards.

Is saw some youtube tutorials on components but could not find making your own components.

How is the performance of ICONICS? Does it work on multiple PLC's?

[ICONICS]

Performance is dictated by your OPC server. The ICONICS platform handles up to 50 millisecond refresh times.

If I am correct ICONICS should be an open scada. So it should work on multiple PLC's.

[ICONICS]

Certainly. Sometimes we connect to thousands of these in the field, for instance in renewable energy industries, water treatment etc.

What features does ICONICS support? Thinking about alarming, roles, code generating, screen resolutions, mobile, dashboards.

[ICONICS]

Mobile – run your ICONICS dashboards on Hololens, tablets or phones. Download the “modilehm” app for your phone from the App stores.

screen resolutions – ICONICS software is responsive to the screen resolution (SVG, re-order on Phones, full size on large screens)

Alarming – Based on OPC A&E standard. Serving/Logging/presentation

Authentication, groups and policies can be maintained in active directory.

code generating – Let us know why you believe you want to generate code. Our solutions can be code-less and you may use the BridgeWorX workflow engine to interact with 3<sup>rd</sup> parties.

On the main site most of these above were named and explained pretty good. Are there tutorials you can use to get a better understanding on how to use each of those in your own project?

[ICONICS]

One ICONICS platform creates the dashboard for any target device. Simply open the “graphworx64” editor and create your dashboard and it works on phone or tablet.

Arthur Blom, Sales and Business Development

m: +31 617 767 403 • [iconics.com](http://iconics.com)

**Antwoord VisiWin**

Good morning Lars,

that's a whole lot of interesting questions.

Unfortunately, it will take some time to give you all the answers, so please be patient.

Alternatively, we could arrange a web conference and I can tell you / show you everything you need?


With kind regards

ppa. Stefan NIERMANN  
Registered Manager / Sales

INOSOFT GmbH  
Buender Straße 194  
D-32120 Hiddenhausen  
52°08.99'N 008°38.93'E

Op basis van deze mail een afspraak gemaakt om een web conference te houden.

## Bijlage V: Prijzen Ignition

	Ignition Basic	Ignition Pro	Ignition Ultimate
	Communicate with PLCs, log historical data, get alarm notifications, and build HMIs.	Level up your SCADA with powerful tools for data management, report creation, and mobile-first visualization.	Expand your horizons with everything you'll need for the ultimate enterprise-grade SCADA solution.
	<b>\$10,800 USD</b>	<b>\$20,850 USD</b>	<b>\$30,250 USD</b>
Ignition Platform	Unlimited	Unlimited	Unlimited
Tags & Designers	Unlimited	Unlimited	Unlimited
Vision Clients	Unlimited	Unlimited	Unlimited
Perspective Clients <small>NEW!</small>	Optional	Unlimited	Unlimited
OPC UA Server Module	✓	✓	✓
Tag Historian Module	✓	✓	✓
Symbol Factory	✓	✓	✓
Alarm Notification Module	✓	✓	✓
Allen-Bradley Drivers	✓	✓	✓
Siemens Driver	Optional	✓	✓
Reporting Module	Optional	✓	✓
SQL Bridge Module	Optional	✓	✓
Voice Notification Module	Optional	Optional	✓
SMS Notification Module	Optional	Optional	✓
Sequential Function Charts Module	Optional	Optional	✓
OPC COM DA+HDA Module	Optional	Optional	✓
DNP3 Realtime Driver	Optional	Optional	✓
Enterprise Administration Module	Optional	Optional	✓
Web Development Module	Optional	Optional	✓
SELECT PACKAGE:	<div>Customize</div>	<div>Customize</div>	<div>Customize</div>



## Bijlage VI: Beoordelingsmatrix HMI's

%	Criterion	1-5 (insufficient)	6 (sufficient)	8 (good)	10 (excellent)
5	<b>Installatie en koppeling</b> Is de HMI makkelijk om te installeren? Is de HMI makkelijk om te koppelen met een PLC?	De HMI is moeilijk om te installeren en het koppelen kost meer dan 1 uur aan tijd.	Het is goed te doen om de HMI te installeren en te koppelen met de PLC. Dit kost ongeveer 30 minuten aan tijd.	De HMI is makkelijk om te installeren en te koppelen met de PLC. Het koppelen kost niet meer dan 20 minuten.	De HMI is zeer makkelijk te installeren en te koppelen met de PLC. Het koppelen kost niet meer dan 10 minuten.
15	<b>Documentatie</b> Hoe is de documentatie? Is het makkelijk te begrijpen voor nieuwe developers? Is het up to date? Is er een actieve community waar je vragen aan kan stellen? Zijn er code snippets beschikbaar?	Er is weinig tot geen documentatie. Het werken met deze HMI voor nieuwe developers levert veel problemen op. Er is daarnaast geen actieve community waar je vragen aan kan stellen. Er zijn weinig tot geen code snippets die je kunt gebruiken.	Er is genoeg documentatie te vinden om de HMI goed te begrijpen en de documentatie is up to date. Het begrijpen van deze HMI is voor nieuwe developers goed te doen. Er is een community maar deze is niet altijd even actief. Er zijn weinig plekken waar je vragen kan stellen. Er zijn wat code snippets beschikbaar.	Er is veel documentatie te vinden en de documentatie is up to date. Het begrijpen van deze HMI is makkelijk voor nieuwe developers. Er is een actieve community. Er zijn aardig wat plekken waar je vragen kan stellen en er zijn aardig wat code snippets	Er is heel veel documentatie te vinden en de documentatie is up to date. Het begrijpen van de HMI is zeer makkelijk voor nieuwe developers. Er is een zeer actieve community waar je vragen aan kan stellen. Er zijn veel plekken waar je vragen kan stellen en er zijn veel code snippets beschikbaar.
30	<b>Development</b> Kun je makkelijk met het HMI schermen maken die er goed uitzien? Kun je snel schermen maken in weinig tijd? Hoe is de learning curve?	Het maken van schermen kost veel tijd en zien er over het algemeen standaard uit. De learning curve duurt lang. Het leren kost veel tijd.	Het maken van schermen kost niet veel tijd en de schermen zien er boven gemiddeld uit. De learning curve is prima. Het leren kost qua tijd niet heel lang maar ook niet kort.	Het maken van schermen kost weinig tijd en de schermen zien er goed uit. De learning curve is goed te doen. Het leren kost weinig tijd.	Het maken van schermen kost zeer weinig tijd en zien er professioneel uit. De learning curve is zeer goed. Het leren kost heel weinig tijd.
15	<b>Kosten</b> Wat zijn de kosten van de HMI? Zijn het maandelijkse kosten of betaal je eenmalig voor een licentie. Hoeveel apparaten kunnen op de licentie?	Er zijn hoge maandelijkse kosten (€200,- plus) die betaald moeten worden om de HMI running te houden.	Er zijn maandelijkse kosten om de HMI running te houden (€20,- ~ €50,-). Daarnaast kost een licentie aardig wat geld (€500,- max).	Er zijn weinig tot geen maandelijkse kosten om de HMI running te houden (€10,- ~ €20,-). Daarnaast zijn de prijzen voor een licentie goedkoop (€200,- max).	Er zijn geen maandelijkse kosten om de HMI running te houden. Daarnaast zijn de prijzen voor een licentie erg goedkoop (€100,- max).
15	<b>Programming paradigma</b> Is het goed te begrijpen hoe je een technisch design kan maken van je applicatie? Kun je componenten aanmaken die je opnieuw kan gebruiken? Bestaan er al componenten die je makkelijk kan toevoegen en gebruiken?	Het is zeer moeilijk om te begrijpen hoe het technisch design in elkaar zit van de applicatie. Je kan geen componenten aanmaken. Er bestaan weinig componenten die je kan gebruiken.	Het technisch ontwerp begrijpen is goed te doen. Je kan zelf componenten aanmaken die je later kan hergebruiken. Er bestaan componenten die je kan gebruiken in je project.	Het technisch ontwerp is makkelijk te begrijpen. Je kan zelf gemakkelijk componenten toevoegen en hergebruiken. Er bestaan al veel componenten die je toe kan voegen aan je project.	Het technisch ontwerp is zeer makkelijk te begrijpen in zeer weinig tijd. Je kan zelf heel makkelijk nieuwe componenten maken en hergebruiken. Er bestaan al veel componenten die je toe kan voegen aan je project.
10	<b>Performance and platforms</b> Hoe is de performance van de HMI? Is de HMI snel of duurt het lang om data te versturen en te visualiseren? Kan de HMI op meerdere plc's gebruikt worden?	De performance van de HMI is laag. De wachttijden om data op het scherm te krijgen duurt lang (3 seconde+). De HMI werkt enkel op 1 a 2 type plc's.	De performance van de HMI is gemiddeld. De wachttijden om data op het scherm te krijgen is niet snel maar ook niet langzaam (1-3 seconde). De HMI werkt op de meeste type plc's.	De performance van de HMI is hoog. De wachttijden om data op het scherm te krijgen is minimaal (1 seconde en lager). De HMI werkt op bijna alle type plc's.	De performance van de HMI is zeer hoog. De wachttijden om data op het scherm te krijgen is zeer minimaal (1 seconde en lager). De HMI werkt op alle type plc's.
10	<b>Ondersteuning</b> Ondersteunt de HMI dingen zoals een raamwerk, alarmen beheren, rollen van gebruikers, data opslag, code generatie, dashboards, verschillende resoluties?	De HMI ondersteunt bijna geen van de benoemde functies.	De HMI ondersteunt de meeste van de genoemde functies.	De HMI ondersteunt bijna alle genoemde functies.	De HMI ondersteunt alle genoemde functies.

## Contactinformatie

Indien u vragen heeft over deze rapportage, dan kunt u met onderstaande mensen contact opnemen.

**Aram Malas**

Algemeen directeur

[aram@ma-it.nl](mailto:aram@ma-it.nl)**Naam**

Projectleider

@ma-it.nl

**Naam**

Software engineer

@ma-it.nl

## Bedrijfsgegevens

**MA-IT B.V.**

Keulenstraat 21 | 7418ET DEVENTER  
Science Park Eindhoven 5220 | 5692EG SON  
Einsteinstraat 9 | 4207HZ GORINCHEM

**Tel** 085 130 25 26  
**eM** [info@ma-it.nl](mailto:info@ma-it.nl)  
**URL** [www.my-automation.nl](http://www.my-automation.nl)

**BTW** NL.822179830.B.01  
**KvK** 08222372  
**ABN** NL17ABNA0484418254  
**G** NL39ABNA0994028407

