

2015•2016  
FACULTEIT INDUSTRIËLE INGENIEURSWETENSCHAPPEN  
*master in de industriële wetenschappen: energie*

## Masterproef

Datalogging en analyse door netwerkintegratie van PLC en PC

Promotor :  
ing. Geert LEEN

Promotor :  
ir. GEERT DE SMEDT

Copromotor :  
Mevr. WENDY SLECHTEN

Gezamenlijke opleiding Universiteit Hasselt en KU Leuven

Bas Janssen , Maarten Sciot

*Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: energie*

2015•2016  
Faculteit Industriële  
ingenieurswetenschappen  
*master in de industriële wetenschappen: energie*

## Masterproef

Datalogging en analyse door netwerkimtegratie van PLC en  
PC

Promotor :  
ing. Geert LEEN

Promotor :  
ir. GEERT DE SMEDT

Copromotor :  
Mevr. WENDY SLECHTEN

Bas Janssen , Maarten Sciot

*Scriptie ingediend tot het behalen van de graad van master in de industriële  
wetenschappen: energie*

## Voorwoord

In juni 2015 kregen we de mogelijkheid om een onderzoeksonderwerp voor onze masterproef te kiezen. Na een gesprek met één van onze docenten, bood hij ons een onderwerp aan dat vrijwel meteen onze interesse wekte omdat het PLC-technieken koppelt aan PC-programmering, namelijk *Datalogging en analyse door netwerkindegratie van PLC en pc.*

Wij hebben de kans gehad om onze masterproef te volbrengen bij Heraeus Electro-Nite te Houthalen. Hier hebben wij de kans gekregen om enerzijds onze kennis van PLC-programmatie in de praktijk toe te passen en anderzijds onze beperkte kennis van .NET programmatie uit te breiden en onszelf te testen in de voor ons toenmalige onbekende wereld van Csharp (C#) WPF binnen het .NET Framework. Wij zijn ervan overtuigd dat dit heel zeker van pas gaat komen binnen onze toekomstige carrières. Bovendien hebben wij dit allemaal mogen ervaren in de toffe sfeer van de afdeling Research and Development Instrumentatie.

Nu is de tijd gekomen dat wij onze opleiding als industrieel ingenieur afsluiten met als hoofdelement deze masterproef. Hierin tonen wij aan dat wij de kennis, die ons gedurende onze studiejaren bijgebracht is, beheersen en kunnen toepassen in de praktijk. Hiermee komt er een einde aan een belangrijk hoofdstuk van onze prille levensloop en zijn wij klaar om de stap te zetten in het industriële bedrijfsleven. Vandaar willen wij nog enkele mensen bedanken..

Op de eerste plaats willen we Geert De Smedt en het volledige team van R&D Instrumentatie van Heraeus Electro-Nite bedanken voor de kans om bij hen onze masterproef te mogen verwezenlijken. Ze hebben ons meermaals geholpen en bijgestuurd om ervoor te zorgen dat wij deze masterproef tot een goed einde zouden brengen. Hiervoor zijn wij hen zeer dankbaar.

Ook willen wij onze docent en promotor Geert Leen bedanken voor de nodige ondersteuning en bijsturing gedurende deze masterproef.

Tot slot willen wij onze ouders en iedereen die ons gesteund heeft gedurende onze schoolcarrière bedanken.

Bedankt!

Janssen Bas en Sciot Maarten  
Houthalen, maart 2016



## Inhoudsopgave

<b>Voorwoord .....</b>	<b>1</b>
<b>Inhoudsopgave.....</b>	<b>3</b>
<b>Lijst van tabellen .....</b>	<b>7</b>
<b>Lijst van figuren .....</b>	<b>9</b>
<b>Verklarende woordenlijst .....</b>	<b>13</b>
<b>Abstract.....</b>	<b>15</b>
<b>Abstract in English.....</b>	<b>17</b>
<b>1 Inleiding.....</b>	<b>19</b>
1.1    Situering .....	19
1.2    Probleemstelling.....	19
1.3    Doelstellingen.....	21
1.4    Materiaal en methode .....	22
1.5    Vooruitblik .....	23
<b>2 Literatuurstudie van gebruikte protocollen.....</b>	<b>25</b>
2.1    Profibus .....	25
2.1.1    Datatransmissie .....	26
2.1.2    Topologie.....	27
2.1.3    Redundantie.....	27
2.1.4    Profibus DP protocol.....	27
2.1.5    Communicatie .....	27
2.1.6    Telegram opbouw.....	29
2.2    Ethernet .....	30
2.2.1    Introductie Ethernet.....	30
2.2.2    Netwerk Eigenschappen .....	30
2.2.3    TCP/IP .....	33
2.2.4    EtherNet/IP.....	34
2.2.5    Topologie.....	35
2.2.6    Redundantie.....	36
2.2.7    Opbouw van het Telegram .....	37
2.3    Profinet .....	38
2.3.1    Introductie Profinet.....	38
2.3.2    Profinet IO .....	39
2.3.3    Profinet vs. Profibus .....	40

2.3.4	Datatransmissie .....	41
2.3.5	Topologie.....	41
2.3.6	Redundantie.....	42
2.3.7	Communicatie .....	42
2.3.8	Telegram opbouw.....	43
2.4	Modbus.....	44
2.4.1	Data transmissie.....	44
2.4.2	Topologie.....	45
2.4.3	MODBUS RTU .....	45
2.4.4	MODBUS ASCII .....	46
2.4.5	MODBUS TCP/IP.....	46
2.4.6	MODBUS PLUS .....	47
2.4.7	Telegram.....	47
<b>3</b>	<b>Uitwerking.....</b>	<b>49</b>
3.1	Siemens .....	49
3.1.1	Configuratie.....	50
3.1.2	Telegram inlezen .....	54
3.1.3	Connectie controle.....	55
3.1.4	Verbinding met pc .....	57
3.2	Allen Bradley .....	58
3.2.1	Configuratie.....	59
3.2.2	Telegram inlezen .....	60
3.2.3	Connectie controle.....	62
3.2.4	Verbinding met pc .....	62
3.3	Schneider .....	64
3.3.1	Configuratie.....	65
3.3.2	Telegram inlezen .....	70
3.3.3	Connectie controle.....	73
3.3.4	Verbinding met pc .....	74
3.4	Csharp Applicatie .....	75
3.4.1	Introductie .NET.....	75
3.4.2	Gevolgde stappen .....	79
3.4.3	Handleiding Userinterface.....	83
3.4.4	Werking programma.....	86
<b>4</b>	<b>Testen en resultaten.....</b>	<b>89</b>

4.1	Siemens .....	91
4.2	Allen Bradley .....	98
4.3	Schneider Electric.....	100
4.4	C# programma.....	103
<b>5</b>	<b>Besluit .....</b>	<b>115</b>
	<b>LiteratuurLijst.....</b>	<b>117</b>
	<b>Bijlagen.....</b>	<b>119</b>





## Lijst van tabellen

Tabel 1: Ondersteunde protocollen met bijhorende PLC .....	21
Tabel 2: Overzicht van transmissie mediums .....	26
Tabel 3: Evolutie van de Ethernet netwerken met hun snelheden en soorten kabels .....	31
Tabel 4: Verschillen tussen Ethernet en industrieel Ethernet.....	35
Tabel 5: MODBUS data.....	48
Tabel 6: MODBUS RTU registers E-Line Digitemp .....	68
Tabel 7: Communication errorcodes .....	73
Tabel 8: Errorcodes voor operatie error wanneer communicatie error 00 is .....	73
Tabel 9: Tabel errorcodes operation error als communicatie error FF is .....	74
Tabel 10: Siemens Drivers .....	80
Tabel 11: Allen Bradley Drivers .....	80



## Lijst van figuren

Figuur 1: OSI model versus PROFIBUS.....	25
Figuur 2: Cyclische en acyclische communicatie.....	28
Figuur 3: Opbouw van het Profibus protocol.....	29
Figuur 4: Ontmantelde weergave van de verschillende soorten kabels.....	31
Figuur 5: Vergelijking van het OSI-model met het TCP/IP-protocolstack.....	33
Figuur 6: Protocollen die deel uitmaken van de TCP/IP-protocolsuite.....	34
Figuur 7: EtherNet ring topologie.....	35
Figuur 8: EtherNet ster topologie.....	36
Figuur 9: EtherNet lijn of bus topologie.....	36
Figuur 10: Standaard Ethernet frame.....	37
Figuur 11: Ethernet jumbo frame.....	37
Figuur 12: Profinet combinatie van Profibus met Ethernet.....	38
Figuur 13: Profinet vs. OSI-model.....	38
Figuur 14: Opbouw van een PROFINET IO – systeem.....	39
Figuur 15: Profibus versus Profinet.....	40
Figuur 16: RJ45 en M12 connectoren.....	41
Figuur 17: Profinet topologieën.....	41
Figuur 18: Profinet communicatiekanalen.....	42
Figuur 19: Snelheid en nauwkeurigheid van IRT, RT en TCP/IP communicatie.....	43
Figuur 20: Profinet RT frame.....	43
Figuur 21: Mogelijke topologieën.....	45
Figuur 22: Master slave communicatie in MODBUS RTU.....	46
Figuur 23: MODBUS protocol frame.....	47
Figuur 24: Testmodule CPU314C-2PN/DP.....	49
Figuur 25: Hardware Configuratie.....	50
Figuur 26: Profibus communicatie instellingen.....	50
Figuur 27: Profinet communicatie instellingen.....	50
Figuur 28: Anybus Optie modules.....	51
Figuur 29: Voorbeeld HW configuratie Castemp Wireless.....	52
Figuur 30: Voorbeeld HW Configuratie Digitemp E-Line Profinet.....	53
Figuur 31: FC1 "WRITE_TO_BUFFER".....	54
Figuur 32: FB126 PNIOdiag.....	56
Figuur 33: Adres formule Siemens PNIOdiag.....	56
Figuur 34: Allen-Bradley L30ER.....	58
Figuur 35: Controller Organizer testopstelling.....	59
Figuur 36: JSR naar read_inputs.....	60
Figuur 37: GSV functie voor de systeem tijd.....	62
Figuur 38: Testmodule M258.....	64
Figuur 39: Ethernet instellingen Schneider.....	65
Figuur 40: Aansluitschema RJ45 naar DB9 kabel.....	66
Figuur 41: Prosoft DB9 connector.....	66
Figuur 42: Instellingen MODBUS RTU communicatie met E-Line Digitemp.....	67
Figuur 43: Instellingen MODBUS I/O scanner.....	67

Figuur 44: Instellingen MODBUS RTU kanaal .....	68
Figuur 45: Interface voor het MODBUS_RTU programma .....	70
Figuur 46: MODBUSTCPCONNECT interface .....	71
Figuur 47: Architectuur van het .NET Framework .....	76
Figuur 48: Databinding .....	77
Figuur 49: Schematische weergave van het MVVM-patroon .....	78
Figuur 50: De Userinterface van het C#-programma .....	83
Figuur 51: Het PLC Connection tabblad .....	83
Figuur 52: Het Telegram Sequence tabblad .....	84
Figuur 53: De actie knoppen en meldingen.....	84
Figuur 54: De data weergave.....	85
Figuur 55: Bestand bestaat al .....	86
Figuur 56: Geen PLC connectie .....	86
Figuur 57: Correct telegram .....	88
Figuur 58: .CSV bestand .....	88
Figuur 59: Quick Check modules.....	89
Figuur 60: Checkmate IV .....	89
Figuur 61: CasTemp Wireless instrument met sensormodule.....	90
Figuur 62: Hardware configuratie Siemens met Sensor Lab.....	91
Figuur 63: FC1 interface met instellingen voor Sensor Lab en Profibus .....	92
Figuur 64: Pcconnect DB tijdens de Sensor Lab test over Profibus.....	93
Figuur 65: Hardware configuratie Sensor Lab met Profinet .....	94
Figuur 66: FC1 interface met instellingen voor Sensor Lab en Profinet.....	94
Figuur 67: Pcconnect DB tijdens de test. ....	95
Figuur 68: Error buffer tijdens een goede connectie tussen PLC en meetinstrument. ....	96
Figuur 69: Error buffer tijdens onderbroken connectie .....	97
Figuur 70: Who Active functie tijdens Ethernet IP test.....	98
Figuur 71: Jump SubRoutine functie voor het read_inputs programma.....	98
Figuur 72: Pc.telegram tag tijdens de CasTemp Wireless test .....	99
Figuur 73: Controle voor verbinding tussen Allen-Bradley en meetinstrument. ....	99
Figuur 74: Instellingen voor Modbus RTU programma.....	100
Figuur 75: I/O mapping van de I/O scanner.....	100
Figuur 76: Buffer in de globale variabele lijst.....	101
Figuur 77: Interface voor Modbus TCP programma .....	101
Figuur 78: Globale buffer tijdens de test via Modbus TCP.....	102
Figuur 79: RTU interface tijdens verbroken verbinding .....	102
Figuur 80: PLC connectie tabblad.....	103
Figuur 81: Geen connectie mogelijk .....	103
Figuur 82: Verbonden met PLC maar ontvangt geen data .....	103
Figuur 83: Correct telegram .....	104
Figuur 84: Ingesteld telegram is te klein .....	104
Figuur 85: Ingesteld telegram is te groot.....	104
Figuur 86: Ontvangt geen of te weinig data .....	104
Figuur 87: Userinterface onderbreken van de communicatie tussen sensor en meettoestel.....	105
Figuur 88: Temperatuur en error grafiek communicatietest sensor en meettoestel.....	105
Figuur 89: Opgeslagen CSV-bestand communicatietest tussen sensor en meettoestel.....	105

Figuur 90: Userinterface uitschakelen van het meettoestel.....	106
Figuur 91: Temperatuur grafiek uitschakelen meettoestel.....	106
Figuur 92: CSV-bestand uitschakelen van het meettoestel.....	106
Figuur 93: Userinterface onderbreken van de communicatie tussen meetinstrument en PLC.....	107
Figuur 94: De temperatuur grafiek communicatietest meetinstrument en PLC.....	107
Figuur 95: Userinterface uitschakelen van de PLC.....	108
Figuur 96: Userinterface onderbreken van de communicatie tussen PLC en pc.....	108
Figuur 97: Userinterface duurttest met Siemens PLC en Castemp.....	109
Figuur 98: CSV-bestand duurttest met Siemens PLC en Castemp.....	109
Figuur 99: Userinterface duurttest met Allen Bradley en CasTemp.....	110
Figuur 100: CSV-bestand duurttest met Allen Bradley en CasTemp.....	110
Figuur 101: Userinterface E-Line test met Allen Bradley.....	111
Figuur 102: Temperatuur curve E-Line test met Allen Bradley.....	111
Figuur 103: Korte test Sensor Lab test met Siemens.....	112
Figuur 104: Userinterface Sensor Lab test met Siemens.....	112
Figuur 105: CSV-bestand Sensor Lab test met Siemens.....	112



## Verklarende woordenlijst

<b>CSV-file</b>	Comma Separated Value bestand
<b>Data binding</b>	Een controle naar een data object verbinden
<b>De smelt</b>	Een zekere hoeveelheid gesmolten metaal tijdens productieproces van metaal.
<b>Delimiter</b>	Begrenzing
<b>Determinisme</b>	Alles wordt bepaald door eerdere gebeurtenissen volgens vaststaande oorzaken
<b>Diagnostics</b>	De techniek of handeling om een oorzaak te vinden van een gevolg aan de hand van optredende verschijnselen.
<b>Framework</b>	Raamwerk
<b>HMI</b>	Human Machine Interface
<b>Inkapseling</b>	Een strategie waarbij stukjes software niet meer informatie over zichzelf laten blijken dan strikt noodzakelijk
<b>IP</b>	Internet Protocol
<b>Level 2</b>	Netwerk tussen PLC en meetinstrumenten
<b>MPI</b>	Multi Point Interface
<b>MVVM</b>	Model-View -ViewModel
<b>Namespace</b>	Een mechanisme dat de mogelijkheid biedt om namen van variabelen, constanten en functies zodanig te groeperen dat ze binnen verschillende contexten een afzonderlijke betekenis en werking krijgen
<b>OSI model</b>	Open System Interconnection reference model
<b>Overerving</b>	Een klasse erft variabelen, functies en procedures van een superklasse
<b>Polymorfisme</b>	Het gelijkvormig zijn in gebruik en naamgeving van operaties (of methodes)
<b>Preamble</b>	Inleiding tot het telegram
<b>Redundantie</b>	Meervoudig aanwezige delen, zodat het geheel goed blijft functioneren
<b>TCP</b>	Transmission Control Protocol
<b>Topologie</b>	Beschrijving van de samenstelling van een netwerk
<b>WPF</b>	Windows Presentation Foundation





## Abstract

Heraeus Electro-nite n.v. te Houthalen ontwikkelt sensoren en meetinstrumenten voor de controle en aansturing van metaalproductie. Aangezien de klanten van Heraeus vaak beschikken over een oudere installatie ondervinden zij weleens problemen met de integratie van nieuwe meetinstrumenten. Deze masterproef bestaat uit het realiseren van een testopstelling waarmee de integratie gesimuleerd en getest kan worden.

De testopstelling moet voldoen aan verschillende eisen: ten eerste moet het te integreren meetinstrument kunnen communiceren met drie verschillende PLC-merken (Siemens, Allen-Bradley en Schneider). Een tweede eis is dat een C#-testprogramma ontworpen moet worden die de communicatie tussen de PLC en het meetinstrument controleert. Het programma moet in staat zijn om minimaal 48 uur de data te loggen en te exporteren.

Voor de communicatie van het meetinstrument en de PLC wordt er gebruik gemaakt van databuffers en de meest courante protocollen: Profibus, Profinet, Ethernet en Modbus. In deze masterproef is een literatuurstudie uitgevoerd naar voorgaande protocollen en andere populaire communicatie protocollen. Voor elke PLC is ook een voorbeeldprogramma gemaakt. Het C#.NET- programma is ontworpen in WPF volgens de MVVM-structuur. De communicatie wordt gecontroleerd op volledigheid en als de data vernieuwd wordt. De data kan over een periode van meer dan 48 uur gelogd worden naar een .CSV-file.



## Abstract in English

Heraeus Electro-Nite n.v., located in Houthalen, develops sensors and measuring instruments for monitoring and managing metal production processes. As customers of Heraeus often have older installations, they sometimes encounter problems with the integration of new measuring instruments. This master's thesis consists of the realization of a test set-up, which allows Heraeus to test and simulate the integration beforehand.

The test set-up should meet the following requirements. Firstly, the integrated instrument must be able to communicate with three different PLC brands (Siemens, Allen-Bradley and Schneider). A second requirement is that a C#-program must be designed to control the communication between the PLC and measuring instrument. Thirdly, the program should be able to log and export data for at least 48 hours.

For the communication between the PLC and measuring instrument data buffers are used and the most common protocols: Profibus, Profinet, Ethernet and Modbus. In this master's thesis a literature study is done on the above-mentioned protocols and other popular communications protocols. For each PLC, a sample program is created. The C#.net program is designed in WPF in accordance with the MVVM structure. In the communication control the data is checked whether it is complete, and if it is renewed. The data is logged to a CSV file over a period of more than 48 hours.



# 1 Inleiding

Dit hoofdstuk licht het onderwerp van deze masterproef toe. Allereerst geeft sectie 1.1 de situering van het onderzoek en het bedrijf weer. Vervolgens formuleert sectie 1.2 de probleemstelling. Daarna beschrijft sectie 1.3 de doelstellingen van deze masterproef. Sectie 1.4 beschrijft de gebruikte materialen en methodes.

## 1.1 Situering

Heraeus Electro-Nite nv. ontwikkelt en produceert sensoren, samplers en meetinstrumenten die gebruikt worden voor procescontroles bij metaalproductie (staal, gietijzer, aluminium, zink en koper). De sensoren (o.a. thermokoppels en chemische cellen) worden ondergedompeld in een vloeibaar metaal. Afhankelijk van de spanning gegenereerd door het thermokoppel of de geleiding van de chemische cel, meet het instrument een spanning. Vervolgens zet het toestel deze waarden d.m.v. thermokoppeltabellen om naar de gemeten temperatuur. Het meetinstrument gebruikt de temperatuur, afkomstig van het thermokoppel, en het gemeten voltage van de chemische cel om calculaties uit te voeren. Zo kan bv. het koolstofgehalte van staal binnen 10 seconden bepaald worden.

Afhankelijk van deze gemeten waarden kan het proces dan onmiddellijk worden bijgestuurd. Dit zorgt ervoor dat nabehandelingen van de smelt overbodig, of toch minimaal worden. Er is ook minder afkeur door monstername, die typisch langer duurt, en waarvan het resultaat te laat beschikbaar is om het proces nog tijdig bij te sturen.

Het energiegebruik, wat uiteraard ook een belangrijke kost is voor de metaalindustrie, wordt ook geoptimaliseerd doordat er sneller afgegoten kan worden, op het moment waarop de smelt net helemaal optimaal is. Er moet niet onnodig gewacht worden en er kan bijgevolg ook meer productie plaatsvinden. Ook het aantal additieven wordt verminderd, en enkel de nodige toevoegingen moeten gedaan worden doordat men de exacte samenstelling van het staal onmiddellijk, en niet na monstername, kent.

De snelheid waarmee dat gebeurt, is dus duidelijk van cruciaal belang om voor de klant een meerwaarde te bieden in kwaliteit en een vermindering in energiekost.

Bepaalde gemeten of berekende eigenschappen worden van het instrument gestuurd naar de PLC, die het productieproces aanstuurt via het machinenetwerk van de klant, het zogenaamde Level 2 netwerk. Voor de communicatie op dit netwerk wordt gebruik gemaakt van een communicatieprotocol met zijn specifieke dataformaten.. De gebruikte PLC, en dus ook het protocol en dataformaat, is eigen aan de installatie van de klant. Producenten van vaak gebruikte PLC's zijn onder andere Siemens, Allen-Bradley en Schneider. Deze drie grote spelers maken elk gebruik van specifieke communicatieprotocollen, die elk hun eigen voordelen hebben. Deze protocollen worden in detail besproken in de literatuurstudie.

## 1.2 Probleemstelling

Voordat de meetinstrumenten op de markt gebracht worden, wordt de communicatie tussen de PLC en de meetinstrumenten door middel van een PLC-simulatiemodule getest. Deze tool neemt de rol van de PLC over en toont de data, die door het instrument worden uitgestuurd, op de pc. Deze data is ook telkens configureerbaar, want niet iedere klant of meetstand is geïnteresseerd in dezelfde waarden.

Niettegenstaande deze testing moet Heraeus vaststellen dat zo nu en dan klanten problemen rapporteren over de integratie van het meetinstrument in hun Level 2 netwerk. De bestaande testopstelling, zoals hierboven beschreven, voldoet blijkbaar niet. De oorzaak is wellicht dat de testopstelling werkt met een PLC-simulatiemodule en niet met een echte PLC. Immers, de simulatiemodule heeft vaste datagroottes en maar één protocol, terwijl PLC's in de praktijk verschillende protocollen en dataformaten ondersteunen, die eigen zijn aan de gebruikte PLC. Ook de timings kunnen verschillen van een echte PLC, omdat deze meerdere processen controleert. Een tweede beperking van de bestaande testopstelling is dat ze geen duurtest kan uitvoeren. Dit is een test waarbij de communicatie uren of wekenlang moet blijven werken, of herstellen, onder allerlei omstandigheden.

Het nadeel van simulaties is dat ze niet altijd representatief zijn voor een PLC. Een echte PLC geeft hierbij een duidelijke meerwaarde. Door een echte PLC te gebruiken bij het testen, kan er al verzekerd worden dat de communicatie theoretisch gezien kan werken bij een klant. Voor de hand liggende fouten kunnen er dus zo al uitgehaald worden, nog voor het toestel bij de klant staat.

De gevolgen van deze ontoereikende testopstelling zijn erg nadelig voor de klant. Indien de klant daadwerkelijk problemen ondervindt met het integreren van de meetinstrumenten, gaat er te veel tijd verloren. Dit brengt op zijn beurt veel kosten met zich mee.

Het komt nu en dan voor dat producten gelanceerd worden zonder dat deze grondig getest zijn op het Level 2 netwerk van de klant. Dit kan resulteren in het verlies van data, waardoor de processturing verstoord wordt. Aangezien iedere klant beschikt over een eigen geprogrammeerd Level 2 netwerk is de vraag naar verschillende dataformaten (int, float, ASCII, ...) en protocollen (Profibus, Modbus, ...) zeer uitgebreid. Deze combinatie van dataformaten en protocollen is zeer klantspecifiek, aangezien de installatie van de klant niet zomaar aangepast kan worden vanwege andere toestellen die ook op dit netwerk zitten. Vaak is het ook verouderde infrastructuur die niet wordt geüpdatet om budgettaire redenen. Ook is het zo dat er vaak een specialist (extern) wordt ingehuurd om dit netwerk op te zetten, waardoor er niet zomaar een aanpassing, of verandering in setup kan gebeuren bij het gebruik van een nieuw toestel, dat slechts een schakel is in een volledig geoptimaliseerd netwerk.

Doordat deze combinatie van dataformaat en protocol zo specifiek is, is het wenselijk om de testen bij Heraeus uit te voeren alvorens naar de klant te gaan. Omdat de klanten wereldwijd verspreid zitten, en niet zomaar toegang geven tot hun level 2 netwerk, is het niet zo evident om grondige testen bij de klant uit te voeren. Dit aangezien er dan een technicus/ingenieur ter plaatse moet zijn in de niet ideale omstandigheden van de productieomgeving van een staalfabriek.

Daarenboven zou daarvoor de productie onderbroken moeten worden, en dat is vaak geen aanvaardbare situatie.

Het realiseren van een wel toereikende testopstelling wordt bemoeilijkt door volgende punten:

- beperkte PLC kennis bij eigen personeel;
- tijdsgebrek door nieuwe projecten;
- geen kant en klare opstelling verkrijgbaar die aan alle voorwaarden voldoet;
- een specifieke opstelling ontwikkeld door derden beperkt flexibiliteit en toekomstige aanpasbaarheid.

### 1.3 Doelstellingen

Door middel van een degelijke testopstelling kunnen de verschillende combinaties van protocollen en dataformaten getest worden alvorens de meetinstrumenten vrij te geven. Op die manier kan aan de klant een voorbeeld van een PLC-programma gepresenteerd worden om zo hun implementatie te vergemakkelijken. Hierdoor kan lange-afstand-communicatie met de wereldwijd verspreide klanten vermeden worden wanneer er toch iets misloopt en kan de installatie en integratie vanaf de eerste keer foutloos in gebeuren.

Deze masterproef bestaat enerzijds uit het maken van een testopstelling voor de meest courante communicatieprotocollen en PLC-types. De ontwikkeling betreft vier verschillende protocollen en drie PLC-types welke weergegeven worden in Tabel 1. Hiermee kunnen dan alle meettoestellen getest worden.

Tabel 1: Ondersteunde protocollen met bijhorende PLC

Level 2 protocollen	Ondersteunende PLC
Profibus	Siemens PLC
Profinet	Siemens PLC
Ethernet IP	Allen Bradley PLC
Modbus RTU/TCP	Schneider PLC

Anderzijds dient er een programma in C# (Csharp) ontwikkeld te worden waarmee de data, die verzonden wordt van meetinstrument naar PLC, via ethernetconnectie gelogd wordt op een pc. Deze data kan vervolgens gebruikt worden om de connectie van meetinstrument met PLC te controleren en het log te analyseren en communicatiefouten op te sporen.

De testopstelling zal bestaan uit een sensorsimulatiemodule, die bv. een temperatuurcurve van verschillende samples doorstuurt naar het meetinstrument. Hierin kan het dataformaat en protocol gedefinieerd worden waarover het meetinstrument zal communiceren met één van de drie PLC-types. De gecommuniceerde data wordt vervolgens door het C#-programma opgevraagd uit de PLC, en nadien gelogd in een Comma Separated Value (CSV) bestand. Deze file kan dan geopend worden met behulp van Excel om verdere bewerkingen uit te voeren op deze data. Belangrijk is dat de testopstelling eenvoudig uit te breiden en flexibel is, met minimale configuratietijd, om een nieuw instrument en protocol te kunnen testen.

De PLC controleert zelf de cyclustijd, als stabiliteitstest, op de verbinding en als controle zodat er geen data verloren gaat. Deze informatie wordt door de PLC in een aparte datablok gestoken.

De applicatie moet in staat zijn om de ontvangen data kritisch te evalueren aan de hand van het ingegeven telegram. Dit telegram moet identiek zijn aan het opgesteld telegram in het meetinstrument. Bij deze evaluatie worden ook volgende elementen nagekeken:

- ✓ de voltaligheid van de datapakketten;
- ✓ de updatetijd van de datapakketten.

Verder moet de testopstelling ook nog voldoen aan de volgende criteria:

- ✓ de testopstelling moet gestart kunnen worden door een individu die geen kennis heeft van PLC- of C#- programmatie;
- ✓ data moet binnen drie seconden in de PLC gebufferd worden;
- ✓ de PLC-buffer moet ook binnen drie seconden uitgelezen worden door de applicatie;
- ✓ de testopstelling moet in staat zijn minimaal 24uur data te loggen;
- ✓ setup moet eenvoudig uit te breiden zijn voor toekomstige instrumenten waarvan de configuratiefiles (GSD-files) beschikbaar zijn;
- ✓ de communicatie tussen PLC en meetinstrument verloopt in één richting;
- ✓ de userinterface is een eenvoudige weergave van de data met de mogelijkheid tot het exporteren van de data;

Mogelijke uitbreidingen zijn:

- ✓ de communicatie tussen PLC en meetinstrument half tot full duplex laten verlopen;
- ✓ de userinterface uitbreiden met een handleiding om de verschillende PLC's te configureren, GSD-files in te lezen en te downloaden.

## 1.4 Materiaal en methode

Voor de configuratie van de drie PLC-types waarmee de meetinstrumenten in de testopstelling communiceren, zijn verschillende softwarepakketten nodig. De configuratie bestaat uit het definiëren van de hardware waarmee de PLC verbonden is en uit het selecteren van het communicatieprotocol. De Siemens-PLC wordt ondersteund door het softwarepakket Step7. Voor de Allen Bradley PLC dient gebruik gemaakt te worden van het softwarepakket Studio 5000 en RSLinx. Het derde en laatste PLC-type is Schneider Electric en die wordt ondersteund door het softwarepakket SoMachine.

Momenteel zijn de volgende meetinstrumenten ter beschikking om mee te communiceren:

- ✓ CasTemp Wireless,
- ✓ E-Line,
- ✓ Sensor Lab.

De CasTemp is enkel in staat om temperaturen te meten. Andere meetinstrumenten kunnen buiten de temperatuur ook andere of meerdere eigenschappen meten en/of berekenen.

De E-Line is een reeks van instrumenten waaronder de DigiTemp-E, Celox-Lab E, etc. Deze instrumenten zijn telkens specifiek voor één bepaalde applicatie. De toestellen meten met één of twee specifieke sensoren. Zo kan met Celox-Lab E het zuurstofgehalte bepaald worden en met de Carbon-Lab E het koolstofgehalte.

Van alle instrumenten is de Sensor Lab het meest uitgebreide (high-end) toestel omdat de Sensor Lab in staat is om automatisch te detecteren welke sensoren aangesloten zijn en zo dus weet welke eigenschappen gemeten gaan worden en automatisch overschakelt tussen verschillende sensoren. Een groot voordeel van deze meetinstrumenten is dat ze allemaal in staat zijn om draadloos te communiceren met hun gekoppelde sensoren. Op deze manier wordt een speciale en dure bekabeling, die bovendien een zwakke schakel vormt in de ruwe omgeving van de staalproductie, vermeden. Dit wegens de hoge temperaturen en de vaak hardhandige manier van werken. Als resultaat geeft dit een verhoogde betrouwbaarheid en veiligheid.

Het programma waarmee de data van het meetinstrument naar de PLC gelogd wordt, is geschreven in C#. Dit is een objectgeoriënteerde programmeertaal van Microsoft in het .NET-Framework.



Hiervoor wordt gebruik gemaakt van het softwarepakket Visual Studio 2015. Volgens de wensen van het bedrijf gebeurt het programmeren met het Windows Presentation Foundation (WPF) en volgens de Model-View-ViewModel (MVVM) programmeerstructuur.

## 1.5 Vooruitblik

Het volgende hoofdstuk bespreekt de literatuurstudie van de vier hoofdprotocollen die zijn toegepast tijdens de masterproef. Deze protocollen zijn Profibus, Ethernet, Profinet en Modbus. Er is vooral de nadruk gelegd op hun werking en gebruik in de industriële automatisering.

Vervolgens wordt de uitwerking van de testopstelling besproken en hoe de beoogde doelstellingen bereikt zijn. Als eerste worden de drie PLC types Siemens, Allen Bradley en Schneider besproken: hoe de connectie met de meetinstrumenten in elkaar zit, de werking van de geschreven programma's voor het inlezen van het telegram en de controle van de connectie tussen PLC en meetinstrument. Ook wordt er bij elk PLC type gesproken over de connectie met de pc. Na de PLC's volgt de Csharp applicatie, waarin eerst een kleine introductie staat over het hele .NET gebeuren. Vervolgens worden de gevolgde stappen uitgelegd, waarna een kleine handleiding de werking van de userinterface beschrijft. Tot slot zal dit hoofdstuk de werking van de C# applicatie uitleggen.

Na de uitwerking volgen de testen en resultaten. Hierin staan enkele testen die tijdens de masterproef zijn uitgewerkt. De testen zijn opgedeeld volgens PLC type en een apart onderdeel, specifiek voor het C# programma.

Tot slot zal het laatste hoofdstuk een besluit geven over het verloop van deze masterproef en de behaalde resultaten. De literatuurlijst, met alle gebruikte bronnen, vindt men na het besluit. In de bijlagen die op de CD-ROM staan, die zich achteraan in de thesis bevindt, staat de overige literatuurstudie van de andere populaire protocollen die niet zijn toegepast. Daarnaast vindt men op deze CD-ROM ook de testopstelling zoals CC-Link, BACNet, ControlNet, RS232 en CAN bus. Op de CD staan ook de geschreven programma's voor de drie PLC types en de C# applicatie, zowel als PDF alsook de software versies van de projecten.



## 2 Literatuurstudie van gebruikte protocollen

In dit hoofdstuk worden de gebruikte industriële protocollen in detail besproken, namelijk:

- Profibus,
- Ethernet,
- Profinet,
- Modbus.

In de bijlage worden volgende industriële protocollen vergeleken als bijkomend onderzoek:

- ControlNet,
- RS232,
- BACnet,
- CANbus,
- CC-link.

### 2.1 Profibus

Profibus is een fieldbus systeem dat zeer applicatie georiënteerd is en dit met zijn eigen standaard protocol. Het ontwerp van de meeste Profibus modules steunt vooral op het OSI (Open Systems Interconnection Reference Model) 7 lagen model, waar enkel gebruik gemaakt wordt van de 1ste, 2de en 7de laag:

- de eerste laag bepaalt de fysieke transmissie, zoals koperdraad (RS485 en MBP zie deel datatransmissie) en optische vezel;
- de tweede laag bepaalt hoe de connectie tot stand komt, bijvoorbeeld de master slave methode of de token ring methode;
- de zevende laag vormt de interface met de toepassing. In het geval van Profibus wordt het protocol Profibus DP gebruikt.

De overige lagen worden niet gebruikt.

In onderstaande Figuur 1 staat een grafische weergave van het OSI 7 lagen model en het Profibus profiel.

	User program		Application profiles
7	Application Layer		PROFIBUS DP Protocol (DP-V0, DP-V1, DP-V2)
6	Presentation Layer.		Not used
5	Session Layer		
4	Transport Layer		
3	Network Layer		
2	Data link Layer		Fieldbus Data Link (FDL): Master Slave principle Token principle
1	Physical Layer		Transmission technology
	OSI Layer Model		OSI implementation at PROFIBUS

*Figuur 1: OSI model versus PROFIBUS [1, p. 3]*

Profibus volgt de richtlijnen die zijn vastgelegd in de IEC 61158 (“Digital data communication for measurement and control – fieldbus for use in industrial control systems”) en IEC 61784 (“Profile sets for continuous and discrete manufacturing relative to fieldbus use in industrial control systems”)

### 2.1.1 Datatransmissie

Voor Profibus zijn er een aantal verschillende manieren beschikbaar om de gegevens te verzenden:

- RS485,
- MBP,
- optical fiber.

#### RS485

RS485 wordt voornamelijk gebruikt in de productie industrie en bij procesindustrie voor toepassingen die gebruikt worden in explosie vrije omgevingen en waar een hoge snelheid vereist is. Voor gebieden waar wel explosie gevaar is kan echter wel het “Intrinsically Safe” RS 485-IS gebruikt worden. In de meeste gevallen wordt er gebruik gemaakt van een afgeschermd “twisted pair” kabel waarbij de gegevens over twee draden worden verzonden.

#### MBP

MBP (Manchester coded Bus Power) heeft als extra voordeel dat de voeding van de modules ook via de bus gebeurt en heeft bovendien ook een explosie beveiligde versie, genaamd MBP-IS.

MBP is uitermate geschikt voor het gebruik in risicovolle gebieden. Hierdoor vindt het zijn toepassingen vooral in de chemische en petrochemische industrie.

#### Optical fiber

Deze vorm van datatransport wordt vooral gebruikt in gebieden waar veel storingsmogelijk zijn en waar grote afstanden overbrugd moeten worden.

Onderstaande Tabel 2 geeft een overzicht van de verschillende transmissiewaarden.

Tabel 2: Overzicht van transmissie mediums [1, p. 4]

	RS485	RS485-IS	MBP	MBP-IS	Fiber Optic
<b>Data transmission</b>	Digital; differential signals acc. to RS485, NRZ (no return to zero)	Digital; differential signals acc. to RS485, NRZ	Digital, bit-synchronous, Manchester coding	Digital, bit-synchronous, Manchester coding	Optical, digital, NRZ
<b>Transmission rate</b>	9.6 to 12000 Kbit/s	9.6 to 1500 Kbit/s	31.25 Kbit/s	31.25 Kbit/s	9.6 to 12000 Kbit/s
<b>Data security</b>	HD=4; parity bit; start/end delimiter	HD=4; parity bit; start/end delimiter	Preamble; fail-safe start/end delimiter	Preamble; fail-safe start/end delimiter	HD=4; parity bit; start/end delimiter
<b>Cable</b>	Twisted, shielded two-wire cable, cable type A	Twisted, shielded two-wire cable, cable type A	Twisted, shielded two-wire cable, cable type A	Twisted, shielded two-wire cable, cable type A	Multi- and single mode glass fiber; PCF; plastic fiber
<b>Remote power supply</b>	Possible using additional cores	Possible using additional cores	Optional using signal cores	Optional using signal cores	Possible using hybrid cable
<b>Ignition protection types</b>	None	Intrinsic safety Ex ib	None	Intrinsic safety Ex ia/ib	None
<b>Topology</b>	Line topology with termination	Line topology with termination	Line topology with termination	Line and tree topology with termination; also combined	Star and ring topology typical; line topology possible
<b>Number of nodes</b>	Up to 32 nodes per segment. Max. total 126 per network	Up to 32 nodes per segment. Max. total 126 per network	Up to 32 nodes per segment. Max. total 126 per network	Up to 32 nodes per segment. Max. total 126 per network	Up to 126 nodes per network
<b>Number of repeaters</b>	Max. 9 with signal refreshing	Max. 9 with signal refreshing	Max. 4 with signal refreshing	Max. 4 with signal refreshing	Unlimited with signal refreshing; note signal propagation delay

### 2.1.2 Topologie

Bij het gebruik van de RS485 technologie wordt voornamelijk de lijn of bus topologie toegepast, waar men maximaal 32 stations (master of slaves) in één segment kan plaatsen. Indien er meerdere stations nodig zijn, dient er een repeater gebruikt te worden. Op het einde van elk segment moet ook een afsluitweerstand voorzien worden.

Als men daarentegen gebruik maakt van de MBP technologie is bijna elke vorm van topologie toegestaan. De totale lengte van een segment mag niet groter zijn dan 1900 meter en voor MBP-IS is deze maximum lengte slechts 60 meter.

### 2.1.3 Redundantie

Voor systemen waarbij uitval van één of meerdere componenten geen problemen mag geven, biedt Profibus de optie om bepaalde kritische onderdelen dubbel uit te voeren zodat bij uitval van het eerste component, de tweede de werking probleemloos overpakt.

Dit biedt Profibus aan op drie niveaus:

- Master redundancy:  
hierbij is de controller van het systeem redundant;
- Media redundancy:  
dit betekent dat er meerdere verbindingen zijn tussen de componenten zodat bij draadbreek de data langs een andere weg wordt verstuurd;
- Coupler/link/gateway redundancy:  
hierbij worden de koppelingen in het netwerk redundant uitgevoerd.

### 2.1.4 Profibus DP protocol

Het Profibus DP protocol is opgebouwd in drie levels: DP-V0, DP-V1 en DP-V2.

DP-V0 zorgt hoofdzakelijk voor de basisfunctionaliteiten van het protocol. Dit bevat de cyclische communicatie en de toestel-, module- en kanaal specifieke diagnose. Deze helpen in het zoeken naar fouten.

DP-V1 draagt bij tot DP-V0 met de functionaliteit om acyclische communicatie toe te laten. Dit is nodig voor functies zoals: parametrisatie, monitoring en alarm handling.

DP-V2 heeft extra functies voor drive controle die cyclussynchronisatie en time stamping bevat.

### 2.1.5 Communicatie

Profibus laat toe om communicatie cyclisch of acyclisch te laten verlopen. De kern van dit proces is het "Master-Slave" principe, waarbij de master cyclisch connectie maakt met elke slave. Die verstuurt op zijn beurt een antwoord naar de master. Een verzoekbericht bevat de output data (bv. afstand dat een motor moet verplaatsen), en het geassocieerde antwoord bevat input informatie (bv. de laatst gemeten temperatuur).

Een cyclus eindigt pas als alle verbonden slaves in de juiste volgorde zijn aangesproken.

### Cyclische communicatie

Tijdens het opstarten van cyclische communicatie zal de master verbinding maken met de toegewezen slave via het MS0 kanaal. Tijdens deze fase zal de slave de configuratie data van de master controleren in twee stappen.

In de eerste stap worden de configuratiegegevens zoals ID nummers, master adres en watchdog tijd, verzonden naar de slave. De connectie kan alleen tot stand komen als het ID nummer van de configuratie overeen komt met de ID nummer van de slave.

De verbinding wordt daarna gecontroleerd door het opvragen van de diagnostics data. De slave meldt hierin of er ongeldige parameters of configuraties aanwezig zijn. Wanneer alle data correct zijn, zal de master in staat zijn om cyclisch met de slave te communiceren.

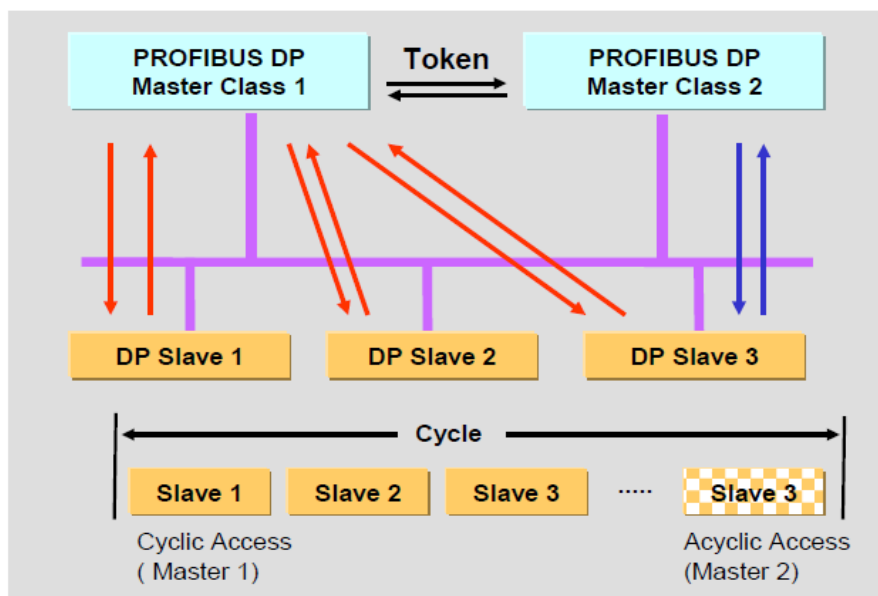
De bovengenoemde diagnostics data kan bestaan uit Profibus-specifieke informatie maar ook data die eigen is aan het slave toestel zelf. De slave zal deze informatie doorsturen naar de master welke op zijn beurt pas de volgende cyclus kan reageren op de informatie.

### Acyclische communicatie

Acyclische communicatie maakt gebruik van twee verschillende kanalen MS1 en MS2. Het MS1 kanaal kan enkel gebruikt worden als er tegelijkertijd een cyclische verbinding is via het MS0 kanaal tussen master en slave.

Het MS2 kanaal moet expliciet door de master geïnitieerd worden. Dit kanaal heeft een aparte tijdscontrolemechanisme waardoor het kanaal zichzelf sluit wanneer er een bepaalde tijd geen dataverkeer is. Acyclische data verbindingen hebben als voordeel dat er geen nood is aan complexe configuratie. Meestal is enkel het adres van het toestel voldoende om te communiceren.

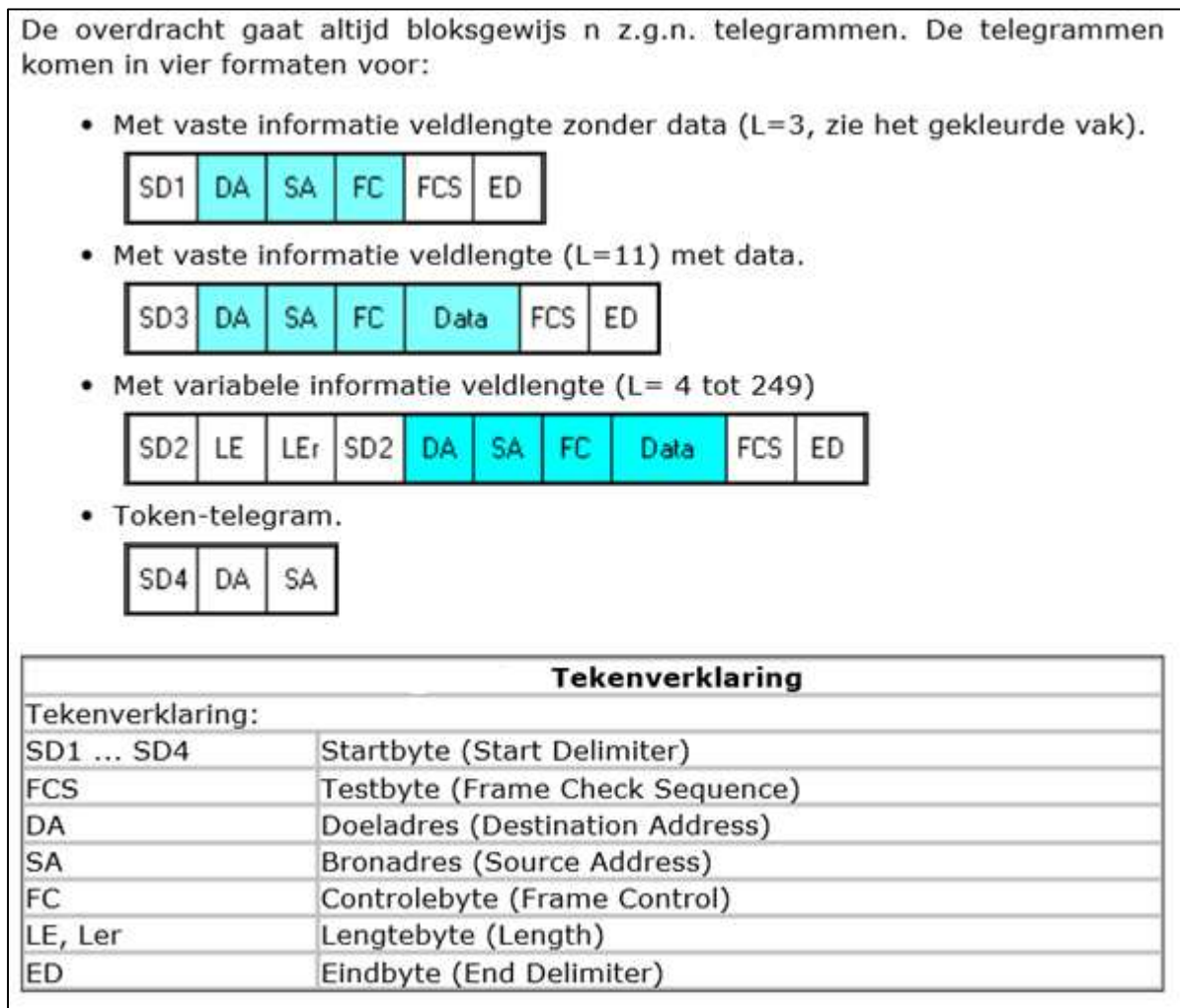
Onderstaande Figuur 2 geeft een grafische voorstelling van cyclische en acyclische communicatie in Profibus DP.



Figuur 2: Cyclische en acyclische communicatie [1, p. 9]

## 2.1.6 Telegram opbouw

Figuur 3 geeft de opbouw van de telegrammen weer.



Figuur 3: Opbouw van het Profibus protocol [2]

## 2.2 Ethernet

### 2.2.1 Introductie Ethernet

Het klassieke Ethernet is het meest bekende en wereldwijd toegepaste bussysteem, waarop stations met elkaar verbonden worden via netwerk componenten. De stations, zoals pc's, worden verbonden via hun Network Interface Card (NIC), Communication processors (CP) in het geval van Automatisatie systemen, met switches en routers om de data botsingvrij te kunnen switchen tussen deze stations. De datapakketten worden verstuurd naar alle andere stations op het netwerk. Elk station filtert de datapakketten eruit die voor hun bedoeld zijn en negeert al de andere data. In het geval van het gebruik van switches zorgt de switch ervoor dat de datapakketten rechtstreeks naar de juiste ontvanger verstuurd worden. Dit door middel van het destination address van de datapakketten te lezen.

De toegankelijkheid wordt gecontroleerd volgens de Carrier Sense Multiple Access with Collision Detection (CSMA/CD) procedure. Dit houdt in dat als een station data wilt verzenden deze eerst gaat controleren dat het netwerk vrij is (Carrier Sense). Indien het netwerk vrij is, kan de data verzonden worden. Vervolgens wordt er gecontroleerd of de andere stations niet begonnen zijn met data te zenden (Collision Detection). Gebeurt dit wel, dan komt een databotsing voor. Indien dit voorkomt, stoppen alle stations met zenden en wachten ze een willekeurige tijd vooraleer opnieuw voorgaande stappen te overlopen. Indien het aantal botsingen van data toeneemt, zal het netwerk trager worden.

Ethernet volgt de richtlijnen die vastgelegd zijn in IEEE 802.3 door IEEE (Institute of Electrical and Electronics Engineers) met betrekking tot de fysische laag en het MAC(Media Access Control) van de datalinklaag.

### 2.2.2 Netwerk Eigenschappen

#### **Ethernet evolutie**

Het oorspronkelijke 10BASE5 Ethernet maakte gebruik van een coaxiale kabel. Nieuwe varianten van ethernet maken gebruik van twisted pair en Optische vezel(fiber) kabels.

Bij het oorspronkelijke Ethernet (Legacy Ethernet) was de transmissiesnelheid gelimiteerd tot 10Mbps. Het daarop volgende Fast Ethernet verhoogde de transmissie snelheid tot 100Mbps. Door de grote vraag naar een sneller Ethernet ontstond GigaBit Ethernet met een transmissie snelheid tot 1000Mbps. Optische vezelkabels kunnen gebruikt worden voor nog hogere snelheden tot 10Gbps.



Onderstaande Tabel 3 geeft enkele voorbeelden van de Ethernet-netwerken in de evolutie van verschillende transmissie snelheden van het Ethernet.

Tabel 3: Evolutie van de Ethernet netwerken met hun snelheden en soorten kabels

Ethernet	Transmissie Snelheid	Netwerken	Cable Type
Legacy Ethernet	10Mbps	10BASE5 - "Thicknet"	Coaxiaal
		10BASE2 - "Thinnet"	Coaxiaal
		10BASE - T	Twisted pair
Fast Ethernet	100Mbps	100BASE - TX	Twisted pair
		100BASE - FX	Multi-Mode Fiber Optic
Gigabit Ethernet	1000Mbps	1000BASE - T	Twisted pair
		1000BASE - SX	Multi-Mode Fiber Optic
		1000BASE - LX	Multi-Mode Fiber Optic

De

benaming van de netwerken wordt afgeleid uit de fysische eigenschappen:

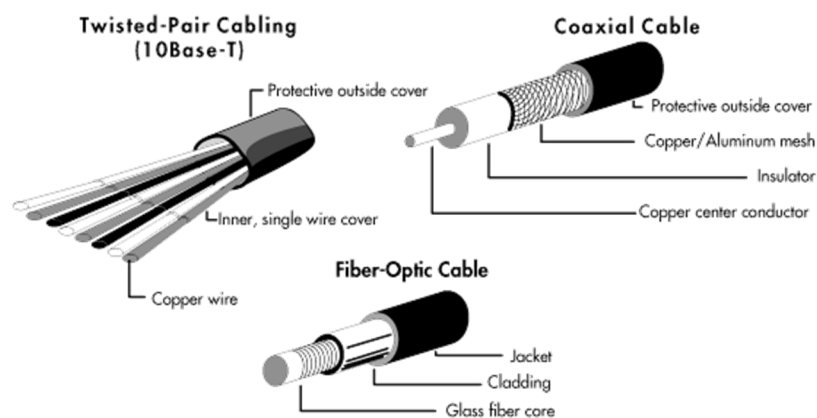
- Het kopnummer staat voor de transmissie snelheid
- BASE staat voor baseband transmissie
- Het laatste deel staat voor het type kabel

### Kabeltypes

De kabels beschikken over een RJ-45 connector op de uiteinden. RJ-45 (Registered Jack – 45) is een 8-polige connector die gebruikt wordt voor twisted pair ethernetverbindingen.

Voor het gebruik van optische vezelkabels dient er gebruik gemaakt te worden van "Ethernet to Fiber" convertoren.

Figuur 4 geeft de ontmantelde kabels weer.



Figuur 4: Ontmantelde weergave van de verschillende soorten kabels [11]

De hedendaagse Ethernetnetwerken maken gebruik van:

- Categorie 5 kabel (UTP-Cat5 of STP-Cat5)
- Categorie 6 kabel (UTP-Cat6 or STP-Cat6).

Beide kabeltypes kunnen gebruikt worden voor zowel het Legacy Ethernet als het Fast en het Gigabit Ethernet. Ze zijn ook compatibel met alle voorafgaande kabels (Cat1, Cat2, Cat3 en Cat4). Zowel Cat5 en Cat6 zijn verkrijgbaar als UTP en STP. UTP staat voor Unshielded Twisted Pair. STP staat voor Shielded Twisted Pair en biedt meer weerstand tegen externe storingen. Categorie 6 werkt beter met snellere signalen zoals 1GBit of 10GBit, in tegenstelling tot Categorie 5. Het biedt ook het voordeel dat deze beter bestand is tegen Crosstalk (elektrische interferentie in de kabel zelf).

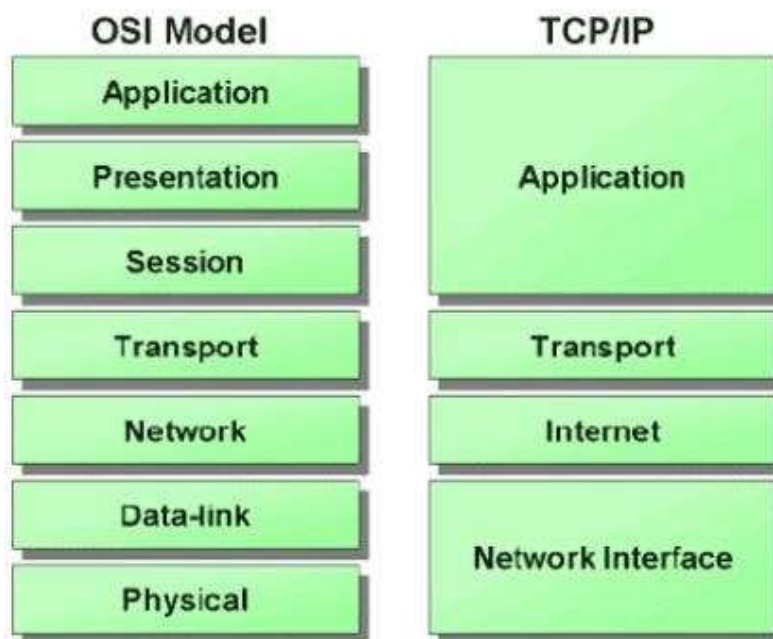
### **Netwerkkomponenten**

De kabels lopen van de pc's Network Interface Card (NIC) naar een boxen zoals hubs, switches en routers. Een hub is een soort van repeater die de ontvangen data naar alle andere poorten verzendt zonder naar het IP adres te kijken. Switches, daarentegen, versturen de data rechtstreeks naar de poort met het gewenste IP adres, en vermindert zo het aantal databotsingen. Een router verbindt verschillende computernetwerken met elkaar.

### 2.2.3 TCP/IP

TCP/IP is een pakketgeschakeld protocol waarbij de gegevens in kleine pakketjes, onafhankelijk van elkaar, worden verzonden. De communicatiesoftware plaatst de pakketten weer in de juiste volgorde, detecteert eventuele fouten in de ontvangst om indien nodig bepaalde pakketten opnieuw te vragen totdat alle pakketten ontvangen zijn. Het is een samentrekking van de twee bekendste protocollen die deel uit maken van de TCP/IP-protocolstack (= protocolstapel): het Transmission Control Protocol (TCP) en het internetprotocol (IP). Het specificeert hoe de gegevens verpakt, geadresseerd, verzonden, geretourneerd en ontvangen moeten worden. Het grootste voorbeeld van een TCP/IP-netwerk is het internet.

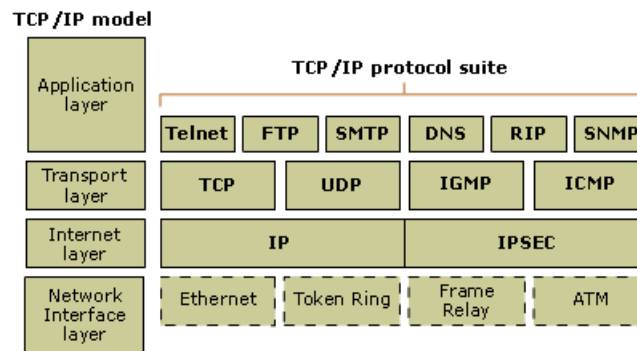
Deze functionaliteit omvat vier abstractielagen die gebruikt worden om alle protocollen te sorteren volgens het bereik van de betrokken netwerken. Figuur 5 geeft de vergelijking van het OSI-model en het TCP/IP-protocolstack.



Figuur 5: Vergelijking van het OSI-model met het TCP/IP-protocolstack [12]

In de applicatielaag wordt bepaald welk protocol gebruikt wordt en op welke manier hostprogramma's gegevens uitwisselen met de transportlaag, zodat er gebruik gemaakt kan worden van het netwerk. De transportlaag behandelt de host- tot- host communicatie. Deze definieert het serviceniveau en de status van de gegevensverbinding. De internetlaag verpakt de gegevens in IP-datapakketten. Deze pakketten bevatten het bron- en doeladres. De retournering wordt ook behandeld in deze laag. De Network Interface of Link Layer bevat communicatie methodes voor het fysiek verzenden van de gegevens, zoals de manier waarop de data geconverteerd wordt tot elektrische signalen.

In Figuur 6 worden een aantal van de gebruikte protocollen weergegeven.



Figuur 6: Protocollen die deel uitmaken van de TCP/IP-protocolsuite

De protocollen die zich in de bovenste drie lagen bevinden van het TCP/IP-model behoren tot de TCP/IP-protocolsuite. Elke laag heeft zijn eigen protocol om zijn bijhorende taken uit te voeren.

#### 2.2.4 EtherNet/IP

EtherNet/IP is een protocol gebruikt in de applicatie laag van een TCP/IP pakket. Dit brengt ons, in tegenstelling met het voorgaande, bij een industrieel Ethernet netwerk. Dit is een netwerk dat het standaard Ethernet combineert met het CIP(Common Industrial Protocol). Het maakt gebruik van de meest gebruikte Ethernet richtlijnen zowel het Internet Protocol Suite (TCP/IP) als het IEEE 802 om de kenmerken en functies voor de transport-, netwerk-, datalink- en fysieke laag te definiëren. Dankzij het objectgeoriënteerde ontwerp van CIP heeft EtherNet/IP de benodigheden voor realtime controle applicaties en implementatie van automatisatie functies.

Data is georganiseerd als een verzameling objecten. Deze bestaan uit twee soorten:

- benodigde objecten,
- applicatie objecten.

Benodigde objecten zijn nodig voor de specificatie van alle CIP-apparaten. Dit is onder andere het identiteits-, een messengerouter- en een netwerkobject. Het identiteitsobject bevat attributen zoals het ID, datum van productie, serienummer, etc. Het messengerouterobject behandelt boodschappen van object tot object. Een netwerkobject bevat de fysieke connectie data, zoals het MacID voor CIP-apparaten en IP-adres voor EtherNet/IP-apparaten. De organisatie van data in de applicatie objecten zijn apparaat specifiek.

Een belangrijk voorbeeld is dat EtherNet/IP gebruik maakt van commerciële Ethernet componenten. Dit maakt EtherNet/IP een zeer kostenefficiënte oplossing.

Een EtherNet/IP-netwerk :

- ondersteunt de configuratie, het verzamelen van gegevens, en de controle op realtime netwerk;
- biedt integratie op de fabrieksvloer;
- ondersteunt tijdkritische applicaties;
- ondersteunt toepassingen voor veiligheid, motie, aandrijvingen, I/O, en tijdsynchronisatie.

Tabel 4 geeft een aantal verschillen tussen het standaard “Enterprise” Ethernet en het industrieel Ethernet.

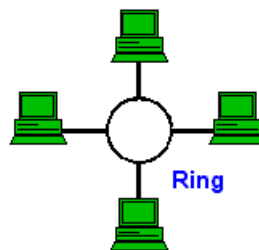
Tabel 4: Verschillen tussen Ethernet en industrieel Ethernet [3]

	<b>Enterprise</b>	<b>Industrial</b>
<b>Installation</b>	Fixed basic installation in the building, Typically use RJ45 connectors kept "in the clear", Network is normally centrally located in wiring closets.	Plant-dependent cabling and cable ducting, Field attachable connectors with ratings of up to IP67, Network is distributed throughout the plant in panels and small cabinets.
<b>Data</b>	Medium-to-large data packets, Medium-to-high network availability, Predominantly acyclic transmission- file downloads, printing, Internet, etc, Real-time behavior not necessary.	Small data packets, Very high network availability, Predominantly cyclic transmission- known data rate with programmed request/reply intervals, Real-time behaviour necessary, especially where coordinated motion is used (IEEE 1588).
<b>Environment</b>	Normal temperature range, Little dust, moisture and vibration, Hardly any mechanical loads or problems with chemicals, Low EMI/RFI/EMC requirements.	Extended temperature ranges, very low to very high, Dust, moisture and vibrations, Risk of mechanical damage or problems with chemicals in hazardous environments, High EMI/RFI/EMC requirements.

## 2.2.5 Topologie

### Ringtopologie

Figuur 7 geeft een ringtopologie weer.

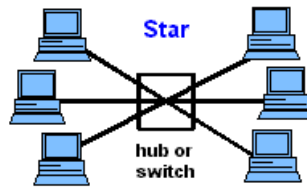


Figuur 7: EtherNet ring topologie [15]

Een ringtopologie kan uitgevoerd worden met centrale Ethernet switches of door de apparaten rechtstreeks aan elkaar te koppelen. Data passeert elk apparaat of elke switch, afhankelijk van hoe de ring opgesteld is. Deze topologie zorgt voor een redundant systeem dat indien een verbinding verbroken wordt, het netwerk blijft werken. Dit omdat het systeem bidirectioneel is en het via een andere richting nog altijd bij het gewenste apparaat uitkomt.

## Stertopologie

Figuur 8 geeft een stertopologie weer.

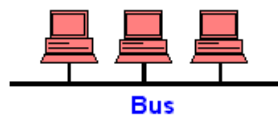


Figuur 8: EtherNet ster topologie [15]

Een stertopologie is een van de meest voorkomende topologieën en bestaat uit één centrale switch die verbonden wordt met omliggende apparaten volgens de point- to- point methode. Het is een betrouwbare topologie want als één van de apparaten uitvalt dan wordt de werking van de andere apparaten niet beïnvloed. Maar indien de centrale switch uitvalt, zal het netwerk niet meer functioneel zijn.

## Lijntopologie

Figuur 9 geeft een lijntopologie weer.



Figuur 9: EtherNet lijn of bus topologie [15]

Hier worden de apparaten direct aan elkaar gekoppeld zonder gebruik te maken van een centrale Ethernet-switch. Dit wordt ook wel een bustopologie genoemd.

### 2.2.6 Redundantie

Er zijn verschillende soorten/ niveaus van redundantie voor een industrieel Ethernet netwerk:

- voedingsbronredundantie,
- mediaredundantie,
- netwerkstationredundantie,
- netwerkredundantie,
- volledig redundant systeem.

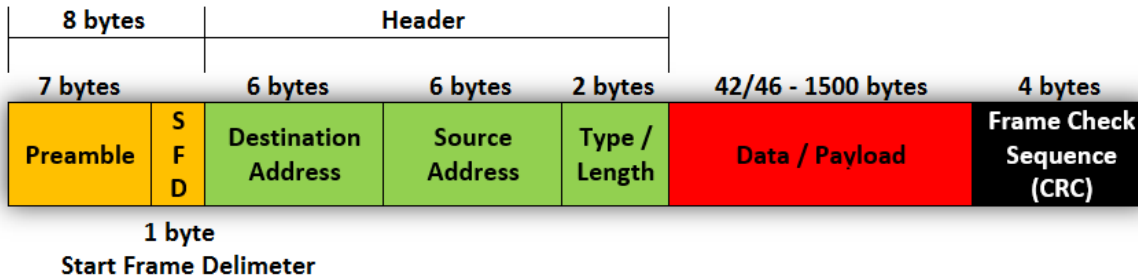
Bij voedingsbronredundantie wordt er gezorgd dat elk deel van het communicatienetwerk voorzien wordt van een back-up voedingsbron, voor het geval dat de eerste voedingsbron wegvalt.

Mediaredundantie houdt in dat er een switch ingesteld wordt als back-up communicatiepad voor het geval dat een deel van het netwerk niet meer bereikbaar is. Om netwerkstationredundantie te bekomen dient er, voor risicovolle apparaten, een dubbel netwerkknooppunt opgezet te worden met switches.

Bij netwerkredundantie wordt er gewerkt met een tweede parallel netwerk. Dit houdt in dat alle apparaten verbonden zijn met twee switches, elk op een ander netwerk. Een volledig redundant systeem bestaat uit redundante componenten zoals switches, communicatiepoorten en apparaten. Alle apparaten en stations zijn verbonden met twee netwerken. Het combineert alle bovenstaande soorten van redundantie.

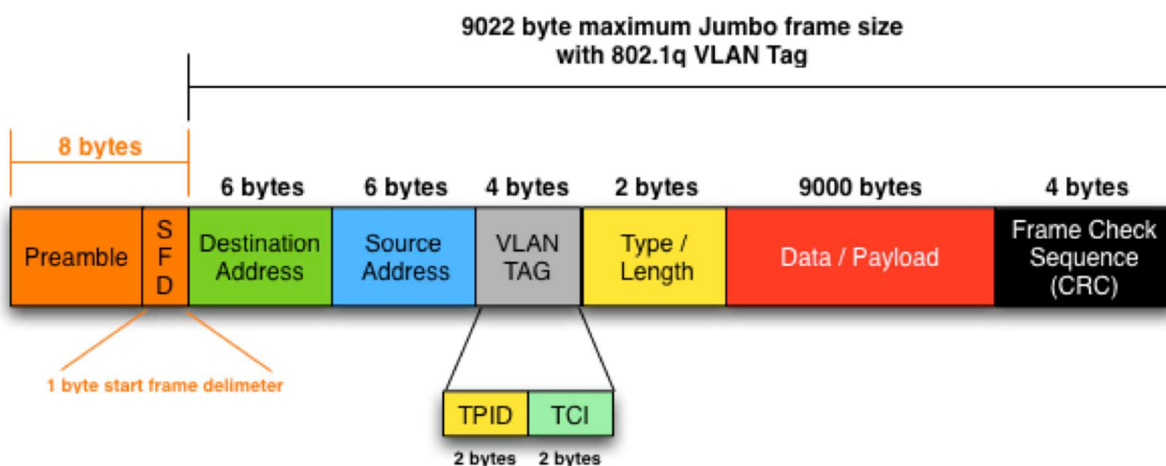
## 2.2.7 Opbouw van het Telegram

Figuur 10 geeft een Ethernet frame weer.



Figuur 10: Standaard Ethernet frame

Een Ethernet frame wordt voorafgegaan door een preamble en een Start Frame Delimiter (SFD), welke tot de fysieke laag behoren. De preamble zorgt ervoor dat de ontvangers van de apparaten op het netwerk gesynchroniseerd worden. De SFD geeft aan dat er nieuwe frame binnenkomt. Het frame begint met een header, welke de MAC-adressen van bestemming en bron bevat, gevolgd door het EtherType en optioneel een IEEE 802.1Q label. Het EtherType kan op twee manieren gebruikt worden, voor waarden van 1500 of lager geeft het de grootte van de data of Payload weer. In het geval van waarden van 1536 of groter geeft het weer welk protocol mee gegeven wordt met de data. Het optionele IEEE 802.1Q label indiceert een virtual LAN en IEEE 802.1p prioriteit. Dit wordt gevolgd door de data die een minimum van 46 bytes (42 bytes indien IEEE 802.1Q aanwezig is) en een maximum van 1500bytes. Jumbo frames laten, zoals hieronder weergegeven in Figuur 11, een grotere payload (tot 9000 bytes) toe. Het frame wordt beëindigd met een Frame Check Sequence (FCS). Dit is een cyclic redundancy check (CRC) welke in staat is onvolledige data te detecteren aan de kant van de ontvanger.

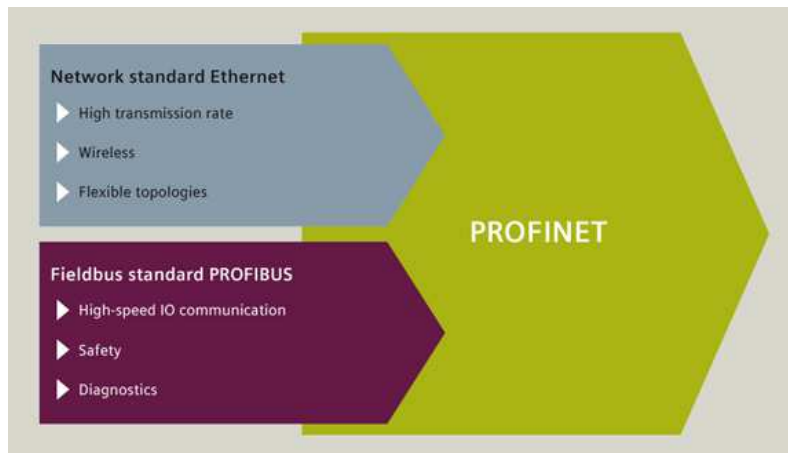


Figuur 11: Ethernet jumbo frame [4]

## 2.3 Profinet

### 2.3.1 Introductie Profinet

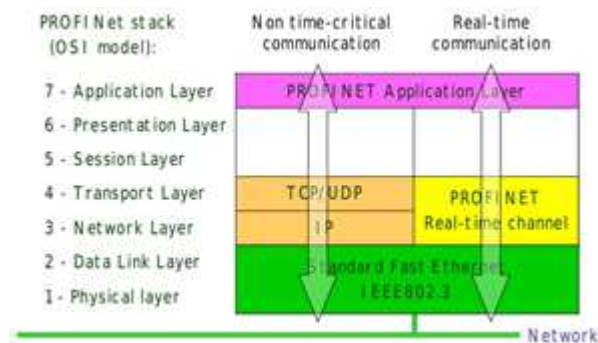
Profinet (PROcess Field NET) is een standaard, gebaseerd op Industrieel Ethernet volgens IEEE 802.xx. Zoals Figuur 12 weergeeft is Profinet een samentrekking van Profibus en Ethernet.



Figuur 12: Profinet combinatie van Profibus met Ethernet [17]

Het maakt gebruik van traditionele Ethernet-hardware en software om een netwerk te definiëren, dat in staat is om taken gestructureerd uit te voeren zoals het uitwisselen van gegevens, alarmen en diagnostiek met programmeerbare controllers.

Figuur 13 geeft de structuur van Profinet weer ten opzichte van het OSI-model.



Figuur 13: Profinet vs. OSI-model [16]

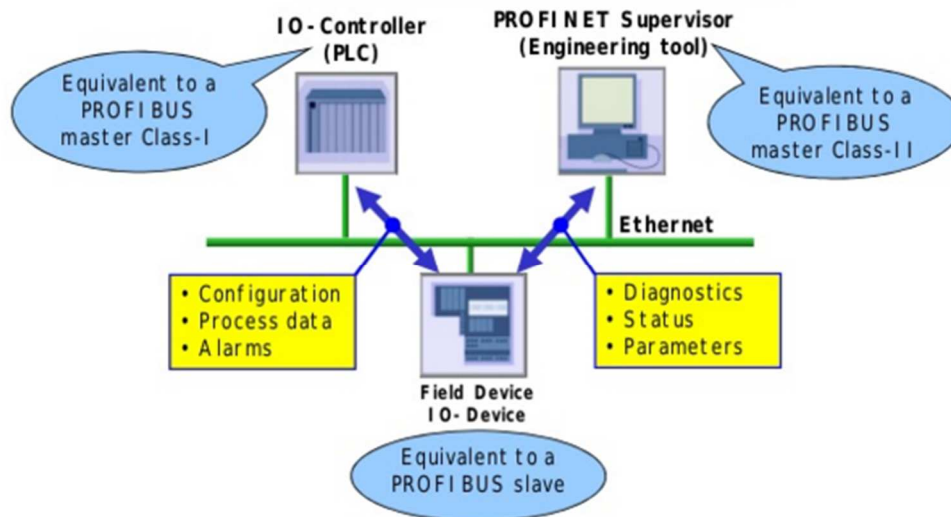
Het TCP/IP kanaal wordt gebruikt voor niet-tijdkritische taken, zoals configuratie downloaden, diagnose, etc. Voor tijdkritische data wordt er gebruik gemaakt van het realtime kanaal, dit is voor cyclische procesdata, alarmen, etc.

Voor optimale ondersteuning bij verschillende applicatietypes, had Profinet twee mogelijkheden: Profinet IO voor de integratie van verspreide periferie (I/O, drives, etc) en Profinet CBA(Component Based Automation) richt zich op communicatie van autonoom werkende automatiseringssystemen. Maar Profinet CBA wordt niet meer toegepast.



### 2.3.2 Profinet IO

Profinet IO lijkt veel op Profibus. Dit wordt duidelijk in onderstaande figuur. Figuur 14 geeft een PROFINET IO – systeem weer.



Figuur 14: Opbouw van een PROFINET IO – systeem [16]

De IO Controller, coördineert de buscommunicatie en de parametring van de IO Devices. Deze ontvangt de procesdata en de alarmen van de IO Devices en stuurt deze door naar een controlesysteem (PLC of PC). De IO Devices, gedecentraliseerde Sensoren/Actuatoren, zijn verbonden via Ethernet met een Controller. Een Device kan bestaan uit verschillende modules en sub modules. De Supervisors zijn bewakende of diagnostische analyse apparaten zoals HMI's of PC's.

IO-Controllers moeten de volgende soorten diensten ondersteunen:

- Cyclic Data Exchange - De uitwisseling van gegevens tussen de IO-Controller en IO-Devices.
- Acyclic Data Exchange - De uitwisseling van configuratie- en diagnostische gegevens
- Alarmen - uitwisseling van alarmgegevens uit een IO-Device naar een IO-Controller
- Context management - Aansluiting verwerking

IO - Supervisors kunnen interne diagnostische gegevens in verband met de PROFINET IO- stack of diagnostische gegevens van het applicatieprogramma van een apparaat lezen en schrijven. Ze zijn ook in staat om configuratiegegevens te lezen en schrijven met behulp van speciale, niet- cyclische geregistreerde dataobject diensten.

Profinet IO is een unieke industriële Ethernet applicatielaag, dat vele voordelen biedt ten opzichte van concurrerende applicatielagen:

- hoge snelheden voor het uitwisselen van berichten - Door het omzeilen van de tijd nodig om de TCP / IP-stack te verwerken;
- bijna identieke Siemens S7 PLC integratie met Profibus;
- ondersteuning voor tijd kritische motion control toepassingen;
- korte opstart tijd;
- verspreide intelligentie;
- eenvoudige installatie.

### 2.3.3 Profinet vs. Profibus

De belangrijkste verschillen van Profinet ten opzichte van Profibus worden in Figuur 15 weergegeven.

	PROFIBUS	PROFINET
Transmission Speed	12MBit/s	100MBit/s 1000MBit/s <i>Backbone</i>
Cycle time	Min. 300µs	Min. 31,25µs
Number of devices on ONE Controller	125 <i>limited</i>	256 (with CPUs) 128 (with CPs)
IO Data	244 Byte	1440 Byte

*Figuur 15: Profibus versus Profinet [18]*

Op het fysieke niveau zijn deze twee communicatienetwerken incompatibel. De basis van Profibus is een RS485-gebaseerde communicatie standaard met een maximale overdrachtssnelheid van 12Mbps. De basis van PROFINET is Ethernet, een wereldwijde standaard met een overdrachtssnelheid van meer dan 100Mbps.

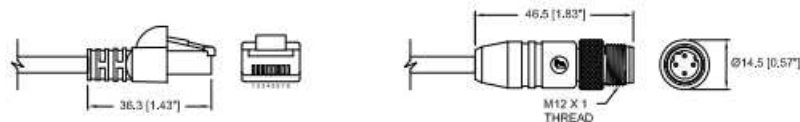
Op het protocolniveau zijn ze wel compatibel. Zowel Profibus en Profinet IO zijn I/O gedreven, hier zit het verschil erin dat Profinet grotere datapakketten ondersteunt en sneller afhandelt dan Profibus. IO-Controllers slaan IO-gegevens van Profinet IO-Devices op in het “periferie proces image” van de controller. Dit wordt vrijwel op dezelfde manier gedaan als bij Profibus-apparaten.

In de opbouw van beide systemen zijn er weinig verschillen, uitgezonderd dat een profinet-netwerk meer devices per controller ondersteunt. De IO-Controllers komen overeen met Profibus Master van klasse 1. De IO-Devices kunnen vergeleken worden met de Profibus slaves. De IO-Supervisors komen overeen met Profibus Master van klasse 2. Dit wordt weergegeven in Figuur 14.

### 2.3.4 Datatransmissie

De communicatie of datatransmissie bij Profinet kan draadloos verlopen of via kabels. Voor de kabels wordt gebruik gemaakt van Industrial Twisted Pair (ITP), zowel Shielded als Unshielded, en optic fiber. Hiervoor dient gebruik gemaakt te worden van conversie modules. Voor draadloze communicatie wordt gebruik gemaakt van Industrial WLAN(Wireless Local Area Network)

Figuur 16 geeft twee verschillende Profinet connectoren weer.

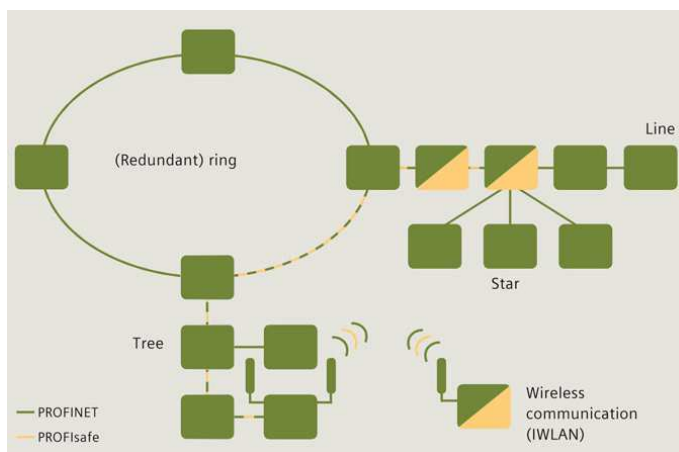


Figuur 16: RJ45 en M12 connectoren [19]

Voor Profinet bestaat een uitgebreid gamma aan kabels van verschillende fabrikanten, maar maakt in het algemeen gebruik van een RJ45 connector. Dit is dezelfde connector als die van Ethernet. Verder wordt, op plaatsen die onderhevig zijn aan frequente of continue beweging, gebruik gemaakt van M12 connectoren. Dit aangezien deze connectoren robuuster, maar nog steeds flexibel zijn.

### 2.3.5 Topologie

Zoals in Figuur 17 weergegeven wordt, zijn er verschillende topologieën mogelijk.



Figuur 17: Profinet topologieën [8]

De weergegeven topologieën zijn dezelfde als die bij Ethernet gebruikt worden:

- Bus/lijn,
- Boom,
- Ster,
- Ring.

### 2.3.6 Redundantie

Stelsystemredundantie: synchronisatie tussen twee onafhankelijke CPU's

Draadloze redundantie: hierbij verloopt de dataoverdracht over twee verschillende bandbreedtes.

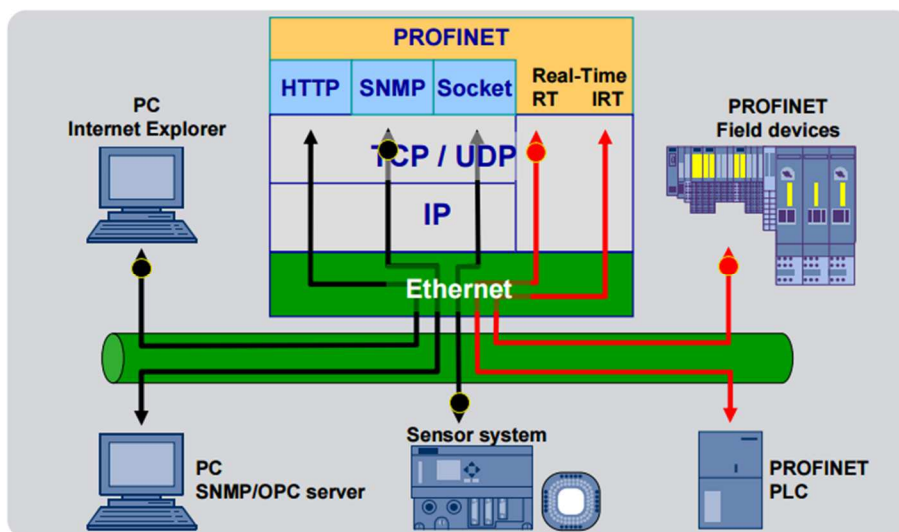
Media Redundancy Protocol(MRP) werkt volgens het principe van de redundante ring bij ethernet en kan bij Profinet geïmplementeerd worden met extra switches of rechtstreeks via de Profinet interfaces.

### 2.3.7 Communicatie

Net zoals Profibus maakt Profinet IO ook gebruik van cyclische communicatie voor het uitwisselen van data. De uitwisseling van configuratie- en diagnostische gegevens gebeurt acyclisch door IO-Controllers.

In Profinet CBA kan data op verschillende manieren uitgewisseld worden, zowel cyclisch als door triggers of andere mechanismen.

Figuur 18 toont een overzicht van de verschillende communicatiekanalen.



Figuur 18: Profinet communicatiekanalen [18]

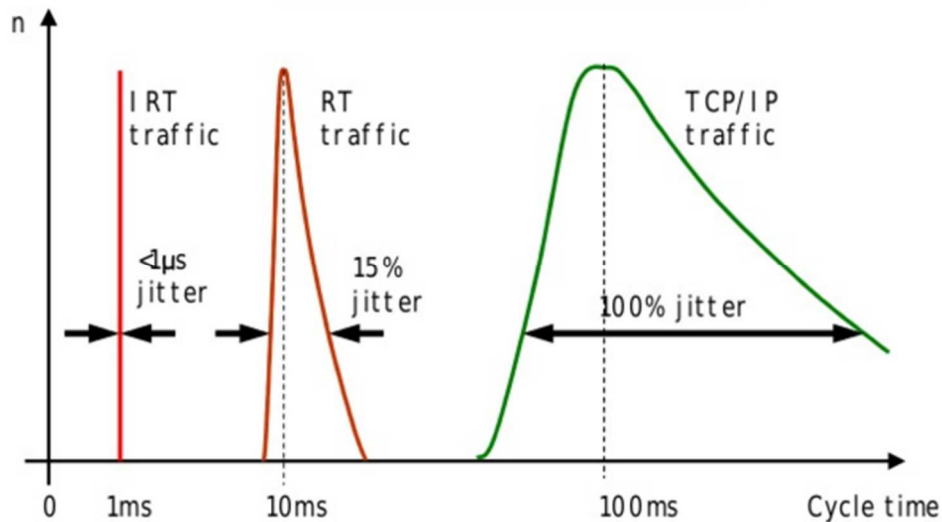
Profinet IO maakt gebruik van drie verschillende communicatiekanalen om gegevens uit te wisselen met programmeerbare controllers en andere apparaten.

- Non-Real-Time (NRT) gebruikmakend van TCP/IP en UDP/IP,
- Real Time (RT),
- Isochrone Real Time (IRT).

Het standaard Non-Real-Time kanaal wordt gebruikt voor niet kritische data zoals parametrisering, configuratie en acyclische lees- en schrijfbewerkingen. Het Real Time kanaal wordt gebruikt voor cyclische dataoverdracht en alarmen. Het derde kanaal, Isochrone Real Time, is het hoge snelheidskanaal voor motion control en safety toepassingen.

Isochrone Real Time (IRT) gegevensuitwisseling met Profinet wordt gedefinieerd in het IRT-concept. Gegevensuitwisselingscycli zijn gewoonlijk in het bereik van enkele honderden microseconden tot enkele milliseconden.

Figuur 19 geeft de snelheid en nauwkeurigheid weer tussen TCP/IP, RT en IRT communicatie.

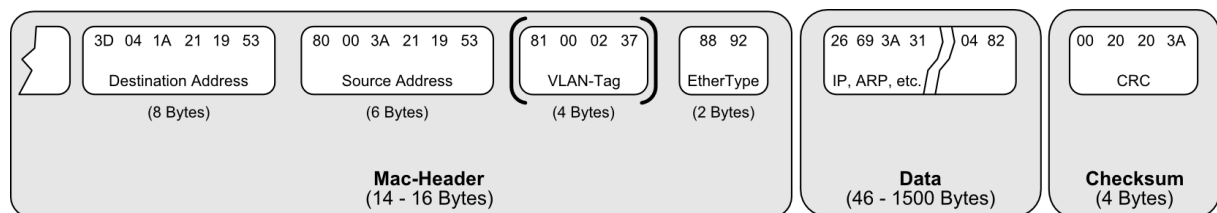


Figuur 19: Snelheid en nauwkeurigheid van IRT, RT en TCP/IP communicatie [16]

Het verschil met real-time communicatie is het determinisme, zodat het begin van een buscyclus met hoge precisie gehouden kan worden. Het begin van een buscyclus kan afwijken tot 1 microseconde (jitter). IRT wordt toegepast voor motion control toepassingen.

### 2.3.8 Telegram opbouw

Zoals Figuur 20 weergeeft, bestaat een Profinet Frame uit een MAC-header, procesdata en een Checksum.



Figuur 20: Profinet RT frame [20]

Het Destination address (aankomstadres) bevat het MAC-adres van de ontvanger. De Source address (bronadres) bevat dan het MAC-adres van de zender. De VLAN-Tag houdt bij welke dataoverdracht prioriteit krijgt ten opzichte van de andere. Het EtherType geeft de identiteit van de daarop volgende data, bijvoorbeeld "0x8892" voor Profinet RT. De data is de procesdata welke doorgegevens dient te worden. De Checksum controleert de verstuurde data.

## 2.4 Modbus

MODBUS is een telegram protocol dat zich voornamelijk bevindt in de 7de laag van het OSI model. MODBUS zorgt voor een “client/server” communicatie tussen toestellen, bussen of netwerken.

Momenteel wordt MODBUS gebruikt onder volgende vormen:

- MODBUS TCP/IP,
- MODBUS RTU,
- MODBUS PLUS.

### 2.4.1 Data transmissie

Vervolgens is er een onderscheid tussen de verschillende transmissie wijzen waarvoor Modbus kan gebruikt worden. Er wordt voornamelijk gesproken over:

- Seriële communicatie met MODBUS RTU en ASCII,
- Ethernet communicatie met MODBUS TCP/IP,
- P2P communicatie met MODBUS PLUS.

#### **Serieel**

Om het MODBUS protocol te gebruiken via seriële communicatie kan men, als fysieke transmissie laag, kiezen voor RS232 alsook RS485. De eenvoud van de implementatie is de reden dat seriële communicatie nog steeds gebruikt wordt in de industrie. Ook heeft seriële communicatie aanzienlijk minder geheugen nodig. Waar MODBUS RTU genoeg heeft aan 2Kb geheugen op een 8-bit CPU of PIC processor, heeft BACnet en Ethernet IP 30 tot 100Kb nodig.

#### **Ethernet**

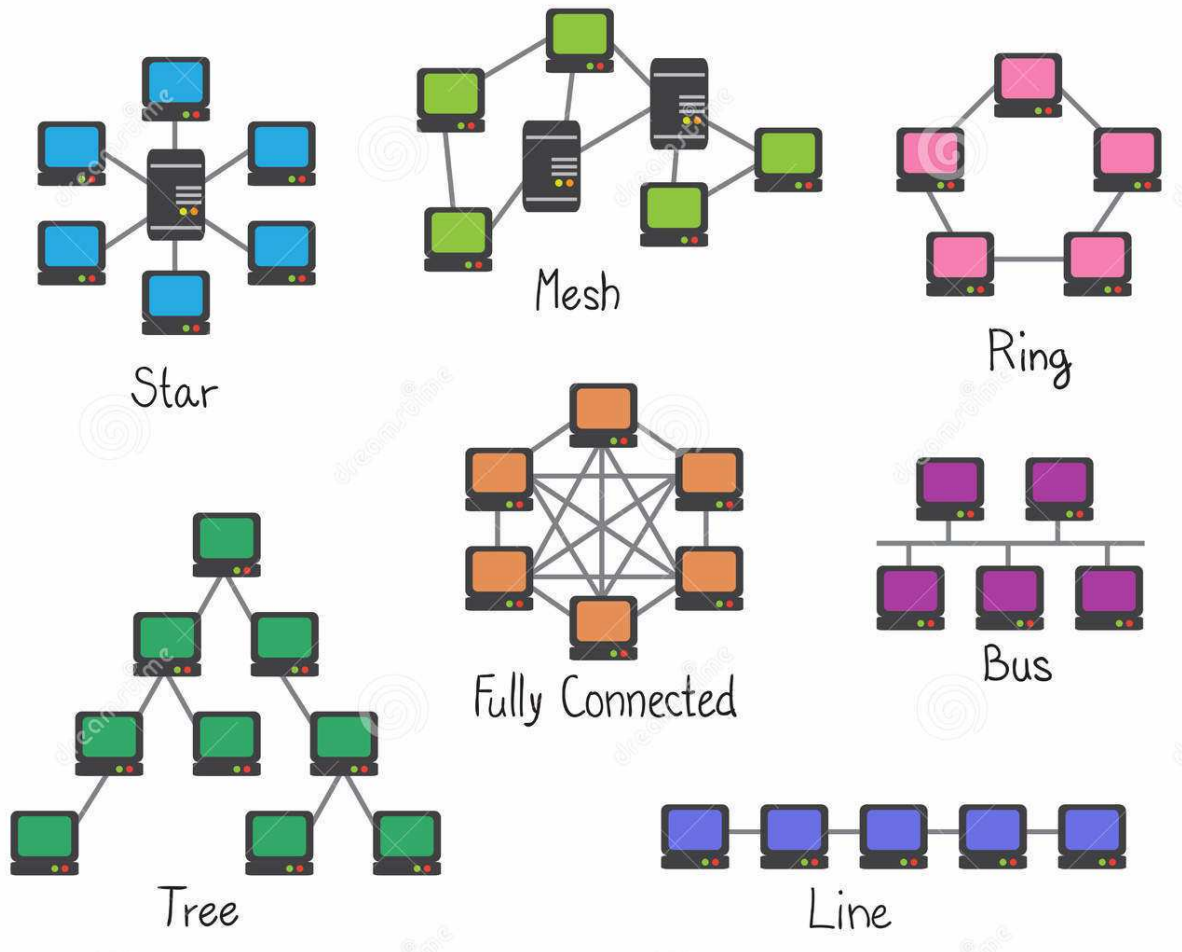
De meest gebruikte vorm van transmissie met MODBUS is Ethernet. Voor meer informatie over de werking zie 2.2 Ethernet.

#### **Peer to peer**

MODBUS PLUS maakt gebruik van het P2P (Peer to Peer) netwerk. In tegenstelling van het client/server model is iedere aansluiting gelijkwaardig en kan elk toestel functioneren als server of als workstation.

## 2.4.2 Topologie

De toegelaten topologie is sterk afhankelijk van de gebruikte transmissiemedia. Voor seriële communicatie wordt meestal een lijnstructuur toegepast. Voor Ethernet en P2P toepassingen zijn er meerdere vormen mogelijk door gebruik te maken van switches, zoals ster-, boom- of ringtopologie. Onderstaande Figuur 21 geeft een overzicht van de verschillende vormen van topologie.

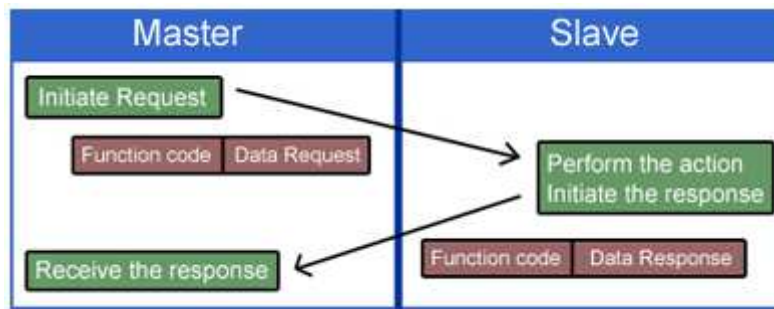


Figuur 21: Mogelijke topologieën [5]

## 2.4.3 MODBUS RTU

MODBUS RTU maakt gebruik van een open seriële communicatie (RS232 of RS485) en maakt gebruik van het Master/Slave principe. RTU staat voor “Remote Terminal Unit” waarbij bedoeld wordt dat een systeem telemetrische data doorstuurt naar een master systeem.

In onderstaande Figuur 22 wordt weergegeven hoe de communicatie tot stand wordt gebracht.



Figuur 22: Master slave communicatie in MODBUS RTU [6]

De Master zal eerst een aanvraag sturen naar de Slave waarbij er vermeld wordt welke actie uitgevoerd moet worden. Hierop zal de Slave de gewenste actie uitvoeren en het antwoord naar de Master zenden met informatie over de actie die de Master moet ondernemen. Pas als de Master dit antwoord heeft ontvangen kan de cyclus opnieuw beginnen.

Het standaard aantal adressen dat MODBUS RTU toelaat gaan van 1 tot en met 254. Waarbij 0 gereserveerd wordt voor “Write only” berichten waarbij er geen mogelijkheid is op bevestiging dat het bericht juist is toegekomen. Dit is echter niet belangrijk als de transmissie gebeurt via RS232 waar er telkens maar 1 Master en 1 Slave kan zijn. Bij RS485 is het aantal adressen gelimiteerd tot 32, maar kan er gebruik gemaakt worden van een repeater om meerdere adressen toe te laten.

#### 2.4.4 MODBUS ASCII

Net zoals MODBUS RTU maakt MODBUS ASCII gebruik van de seriële transmissie laag. Het grootste verschil met MODBUS RTU is dat de waarden ASCII gecodeerd zijn. Dit heeft als nadeel dat ASCII mode telkens twee data woorden (twee 7-bit characters) nodig heeft om een equivalent 8-bit byte in RTU mode te verzenden. Als men bijvoorbeeld, het getal 6A (HEX) wilt, kan dit in RTU mode als één 8-bit byte (0110 1010), maar in ASCII mode heeft ditzelfde getal 2 bytes nodig. Eén voor de ASCII gecodeerde “6” (“6” = 36 = 0011 0110) en één voor de ASCII “A” (“A” = 41 = 0100 0001). Dit betekent dat ASCII mode minder efficiënt is en de voorkeur meestal uitgaat naar RTU.

#### 2.4.5 MODBUS TCP/IP

MODBUS TCP/IP maakt de combinatie tussen het MODBUS protocol en Ethernet TCP/IP transmissie (voor verdere informatie over de werking zie 2.2 Ethernet). Het grootste verschil tussen MODBUS RTU, wat seriële transmissie gebruikt, en MODBUS TCP/IP, wat de fysieke Ethernet laag gebruikt, is een extra 6 bytes header in het telegram. Dit zorgt voor de routing van het signaal.

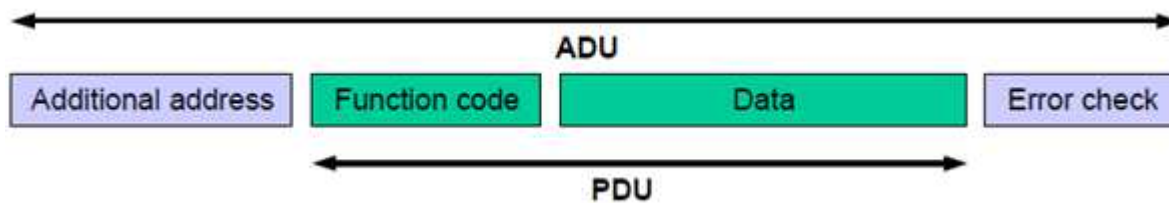


## 2.4.6 MODBUS PLUS

MODBUS PLUS is ontworpen door Modicon en Schneider Electric om een fieldbus te creëren met een P2P netwerk structuur. Dit gebeurt door het “Token Passing” principe toe te passen op een RS485 connectie.

## 2.4.7 Telegram

De onderstaande Figuur 23 geeft een grafische weergave van de opbouw van het MODBUS protocol.



Figuur 23: MODBUS protocol frame [7, p. 3]

Het protocol bestaat uit een simpele datablok (Protocol Data Unit PDU) welke onafhankelijk is van de gebruikte communicatielaag. Buiten deze data wordt er nog extra informatie doorgestuurd. Dit bevat voornamelijk de adresinformatie van de bus of het netwerk in de applicatie data unit (ADU).

De applicatiedata wordt samengesteld door de cliënt die de transactie start. De functiecode vertelt de server welke actie ondernomen moet worden. Dit gebeurt in een 1 byte code die varieert tussen de decimale waarden 1 tot en met 255 (128 – 255 zijn gereserveerd voor het oplossen van uitzonderingen).

Het dataveld bevat extra informatie die de server gebruikt om de actie te ondernemen die aangegeven is in de functiecode. Dit kan informatie bevatten over de hoeveelheid gegevens die behandeld moeten worden. Het dataveld kan leeg zijn wanneer de server geen extra informatie nodig heeft om de functie uit te voeren. Wanneer er zich een probleem voordoet met de MODBUS-communicatie geeft het dataveld de errorcode weer. Deze kan door de server gebruikt worden om de gepaste actie te kunnen ondernemen.

MODBUS maakt gebruik van ‘big-Endian’ representatie voor adressen en data. Dit betekent als er een numerieke waarde groter dan 1 byte wordt verzonden dat de meest beduidende byte eerst verzonden wordt. Als er bijvoorbeeld de waarde 0x1234 wordt verzonden, zal de eerst verzonden byte 0x12 zijn daarna 0x34.

Een overzicht van MODBUS data wordt weer gegeven in Tabel 5.

*Tabel 5: MODBUS data [7, p. 6]*

Primaire tabellen	Object type	type	opmerking
Discrete input	enkele bit	Read-Only	dit type kan door het IO systeem worden aangeboden
Coils	enkele bit	Read-Write	dit type is aanpasbaar door een applicatie programma
Input Registers	16-bit word	Read-Only	dit type kan door het IO systeem worden aangeboden
Holding Registers	16-bit word	Read-Write	dit type is aanpasbaar door een applicatie programma

### 3 Uitwerking

In dit hoofdstuk worden alle verschillende onderdelen besproken, beginnende bij de programmatie. Hier wordt de configuratie van de PLC uitgelegd, gevolgd door de verwerking van de gegevens die afkomstig zijn van het meetinstrument. Hierna komt het controleren van de connectie en de verbinding met de pc aan bod. Dit voor de drie gebruikte PLC merken namelijk:

- Siemens,
- Allen-Bradley,
- Schneider Electric.

De geschreven programma's voor elk van deze PLC's zijn terug te vinden in de bijlage.

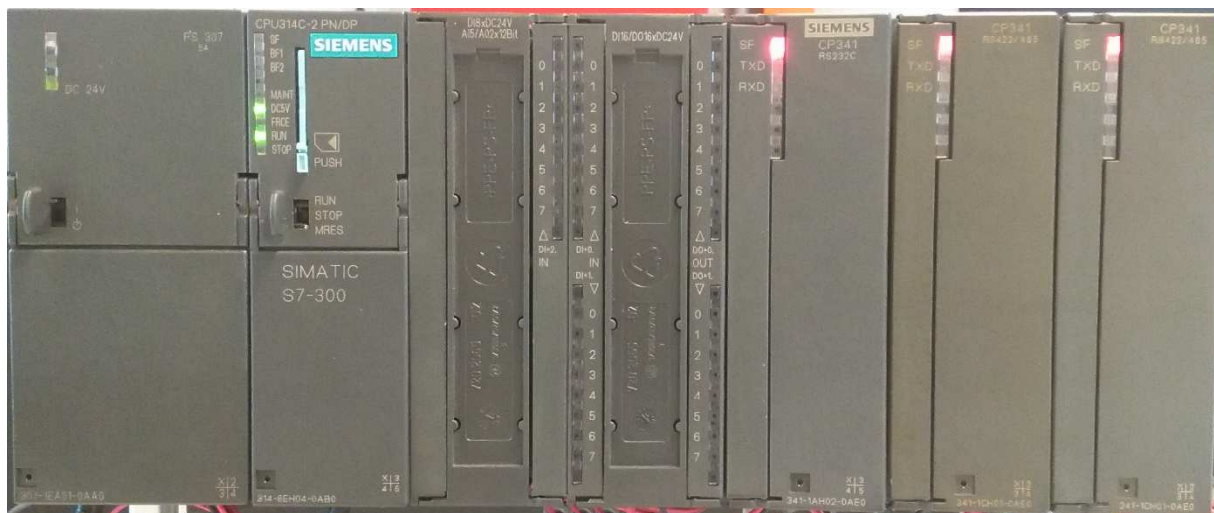
Als laatste onderdeel van dit hoofdstuk komt de C#-applicatie aan bod. Dit begint met een introductie omtrent het .NET Framework en bijhorende definities. Gevolgd door de evolutie van het programma. Ten slotte wordt de werking van de applicatie besproken.

#### 3.1 Siemens

De programmatie van de Siemens PLC gebeurt door middel van het Step 7 V5.5 software pakket van Siemens AG. De testmodule bestaat uit een 24V 4A voeding (PS 307 5A) en een 314C-2 PN/DP CPU met ingebouwde Profinet, Profibus en MPI poorten. De CPU heeft ook al standaard 24/16 digitale in-/uitgangspoorten.

De Siemens PLC kan met de meetinstrumenten communiceren d.m.v. Profinet of Profibus.

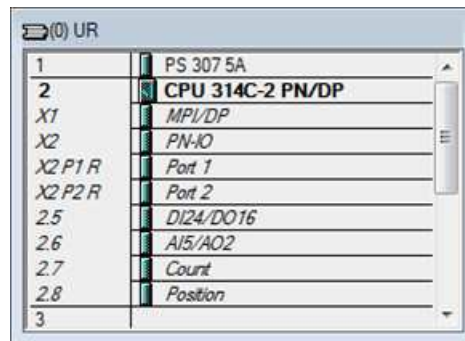
Onderstaande Figuur 24 geeft de huidige testmodule weer.



Figuur 24: Testmodule CPU314C-2PN/DP

### 3.1.1 Configuratie

De configuratie van de PLC en het meetinstrument gebeurt in de HW Configuratie tool van Step 7. Hierbij wordt een onderscheid gemaakt tussen twee kaarten MPI/DP voor Profibus en PN-IO voor Profinet. Figuur 25 geeft het overzicht van de hardware in de PLC weer.

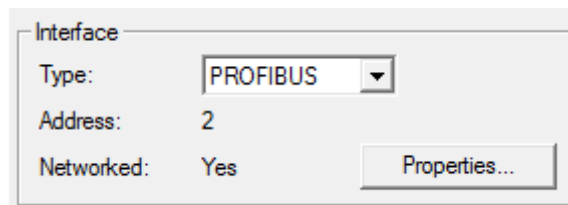


Figuur 25: Hardware Configuratie

#### **MPI/DP (Multi-Point Interface/Decentrale Periferie)**

Hierin wordt de PLC ingesteld en krijgt die een Profibus adres. Het Profibus adres van de PLC is zijn master adres voor de communicatie met de slaves.

De gekozen instellingen voor Profibus communicatie zijn zichtbaar in onderstaande Figuur 26.

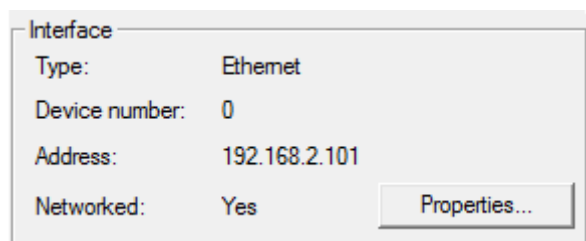


Figuur 26: Profibus communicatie instellingen

#### **PN-IO (Profinet-IO)**

Hierin wordt de PLC ingesteld en krijgt die een IP adres. Het IP adres van de PLC dient voor de ethernet communicatie tussen PLC en ander toestellen binnen hetzelfde netwerk.

De gekozen instellingen voor Profinet communicatie zijn zichtbaar in onderstaande Figuur 27.



Figuur 27: Profinet communicatie instellingen

Figuur 28 geeft de Anybus optie modules weer.



*Figuur 28: Anybus Optie modules*

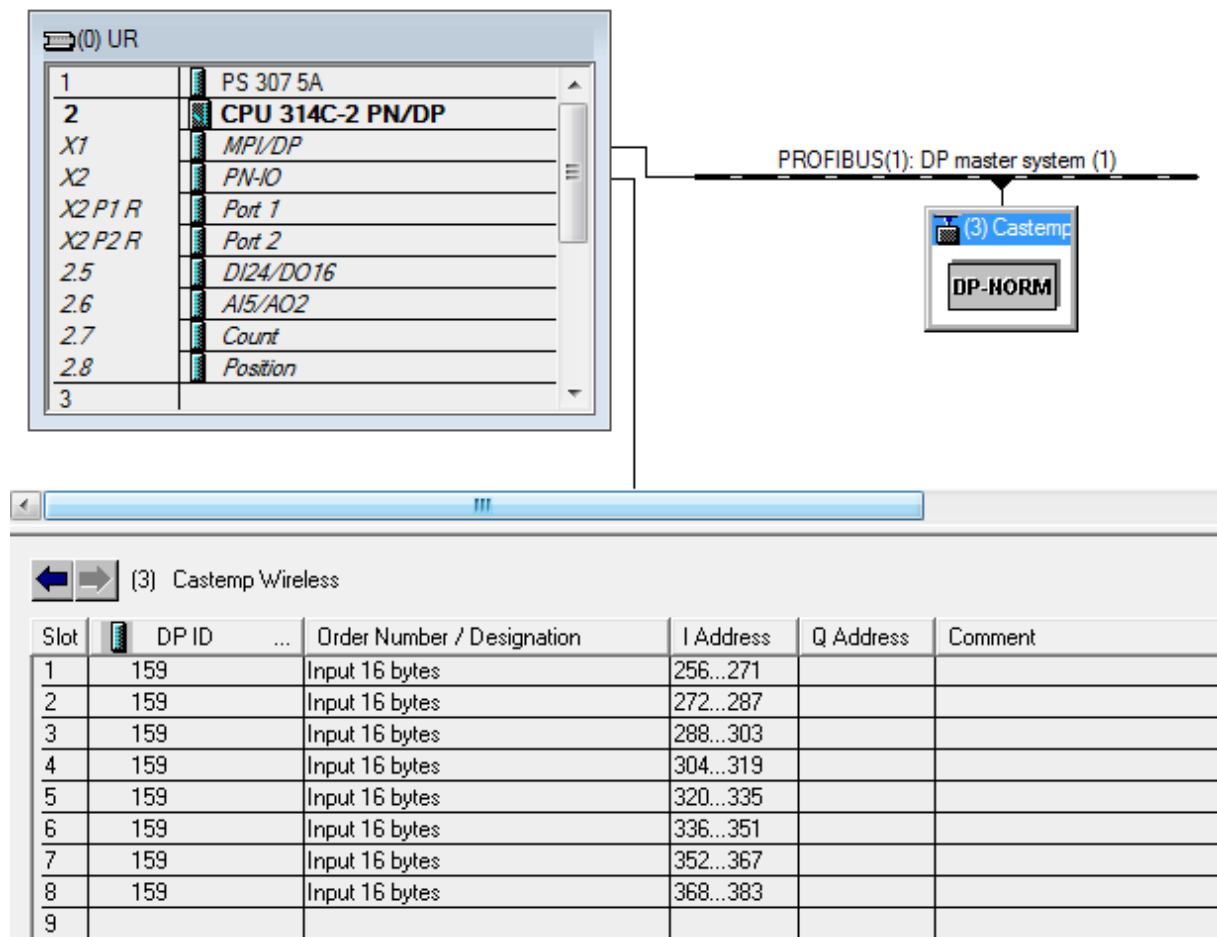
De meetinstrumenten maken gebruik van Anybus optie modules. Dit laat toe om een groot aanbod van communicatieprotocollen aan te bieden door het gebruik van één platform. Voor het gebruik van de Profibus optie module kan het GSD bestand (General Station Description) ingeladen worden in de hardware bibliotheek.

Voor Profinet is er een gelijkaardige GSDML bestand beschikbaar.

Een uitzondering is de iM<sup>2</sup> Sensor Lab waarbij de verschillende protocollen mogelijk gemaakt worden door Beckhoff modules in de plaats van Anybus, maar voor elk van deze modules is er ook weer een GSD/GSDML bestand beschikbaar.

## Profibus

Na het inladen van het GSD bestand kan het meetinstrument aan het netwerk worden toegevoegd. Elk toestel krijgt een uniek Profibus adres dat verschillend is van de PLC. Afhankelijk van het type van meetinstrument moeten er een aantal input en output bytes worden ingevoerd. Onderstaande Figuur 29 geeft een voorbeeld van de hardware configuratie voor een Castemp Wireless met het ABCC-DPV1 Profibus module.

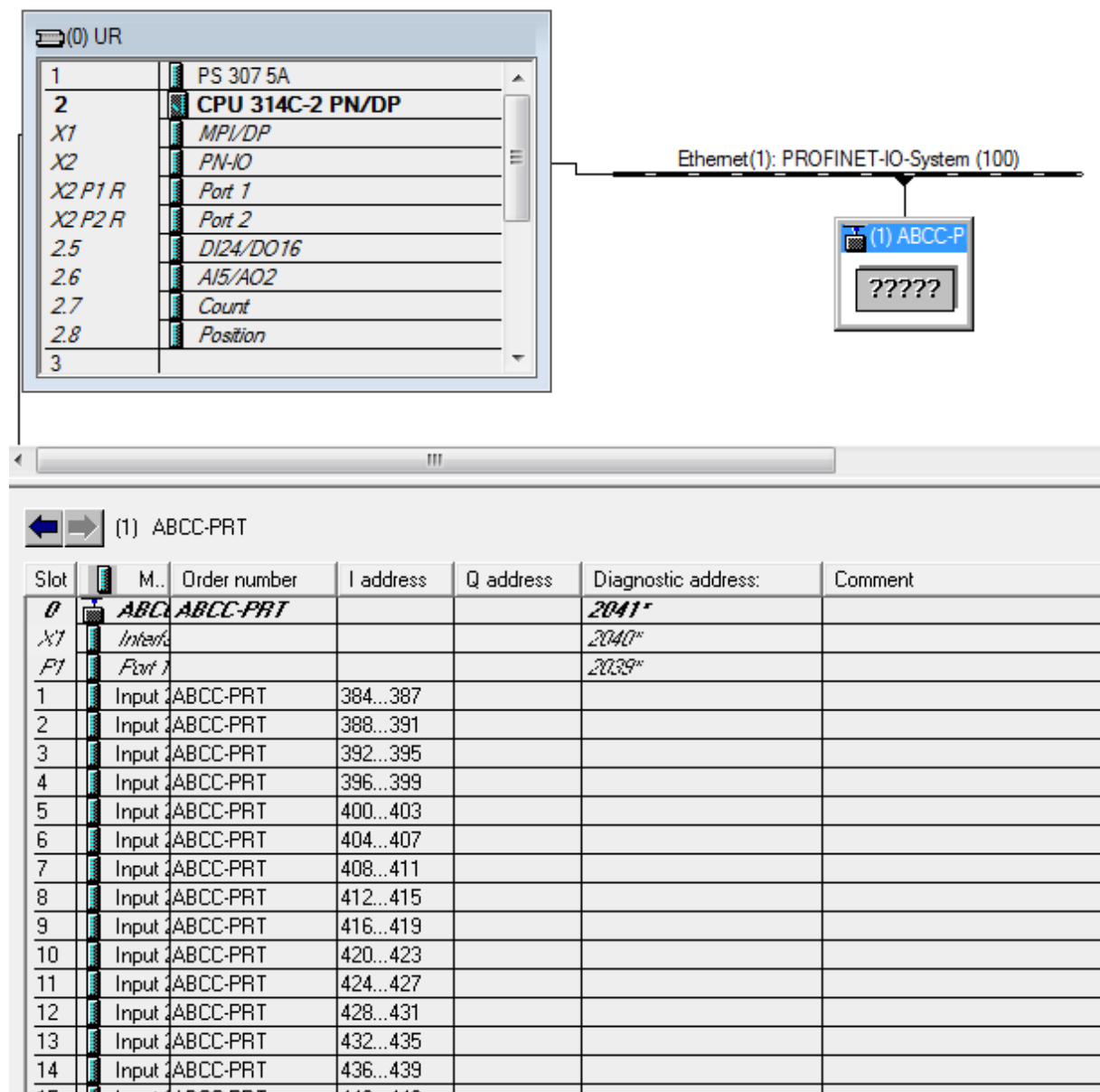


Figuur 29: Voorbeeld HW configuratie Castemp Wireless

## Profinet

Zoals bij Profibus kan na het inladen van het GSDML-bestand het meetinstrument toegevoegd worden aan het Profinet netwerk. Softwarematig zal er in de hardwareconfiguratie een IP-adres ingesteld worden en krijgt het instrument een naam en een MAC-adres. Met deze laatste twee gegevens kunnen via het “Assign device name” menu de instellingen doorgestuurd worden naar het toestel. Dit is voor Profibus niet vereist aangezien deze instellingen hardwarematig op het instrument zelf gebeuren.

Afhankelijk van het type van meetinstrument moeten er een aantal input en output bytes worden ingevoegd. Deze hoeveelheid input en output bytes is terug te vinden in de handleiding van de meetinstrumenten. Onderstaande Figuur 30 geeft een voorbeeld van de hardware configuratie voor een Digitemp E-Line met het ABCC-PRT Profinet module.



Figuur 30: Voorbeeld HW Configuratie Digitemp E-Line Profinet

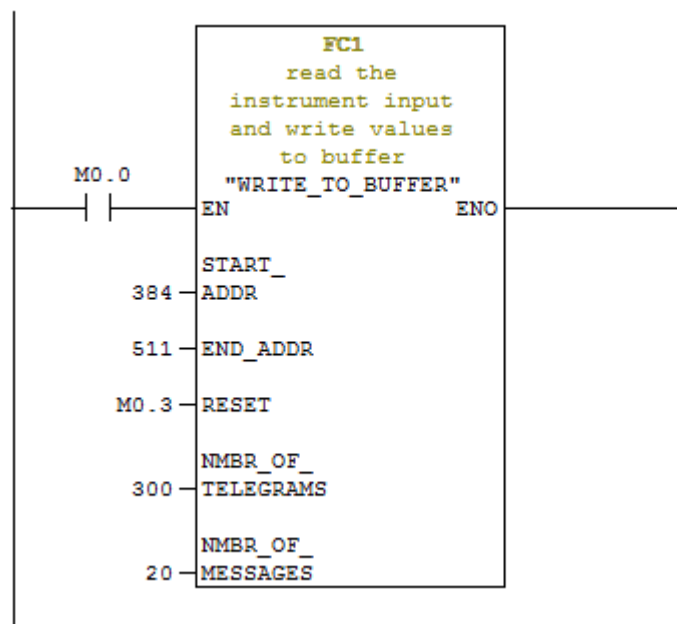
### 3.1.2 Telegram inlezen

Om ervoor te zorgen dat de data volledig wordt verstuurd naar de pc applicatie, wordt de data eerst in het intern geheugen van de PLC gebufferd om te voorkomen dat de waarde verandert tijdens het verzenden.

Voor de buffer wordt DB2 "PB\_Receive" gebruikt. Hierin worden tot twintig boodschappen (meetinstrument telegrammen) opgeslagen waarbij elke boodschap 300 bytes kan bevatten. Eén byte is één PLC-telegram, aangezien de PLC de data byte per byte inleest. De buffer kan in totaal dus 6000 Bytes aan data opslagen. Dit is zo groot gekozen zodat uitbreiding voor nieuwe instrumenten vlot en gemakkelijk kan gebeuren.

Om de waarden in te lezen en weg te schrijven naar de buffer maakt de PLC gebruik van FC1 "WRITE\_TO\_BUFFER" deze functie zal met een instelbaar interval rechtstreeks de gegevens uit het PII (Periferie Input Image) geheugen lezen en deze op de juiste plaats in de buffer schrijven. Dit wordt later besproken.

Onderstaande Figuur 31 geeft FC1 weer en de nodige instelling.



Figuur 31: FC1 "WRITE\_TO\_BUFFER"

Om deze functie te gebruiken moeten het startadres en het eindadres gekend zijn. Het startadres is de eerste ingangsbyte die ingesteld is in de hardwareconfiguratie. Het eindadres is de laatste ingangsbyte ingesteld in de hardware configuratie. De reset zal ervoor zorgen dat de buffer weer vanaf de eerste waarde wordt ingevuld. De grootte van de buffer wordt bepaald met het aantal bytes (PLC-telegrammen) en het aantal boodschappen (meetinstrument telegrammen).



**Werking:**

Het volledige programma is te vinden in de bijlagen. Het programma zal eerst bepalen hoeveel waarden er ingelezen moeten worden, dit door aantal waarden = eindadres – startadres.

Hierna zal een “pulstimer” telkens om de 20 ms een signaal geven om telkens één waarde in te lezen. De benodigde tijd om een volledig telegram in te lezen kan dus berekend worden door.

$$T_{boodschap} = T_{puls} \times \text{aantalwaarden}$$

Na iedere puls zal de telegramteller met 1 verhogen. Bij elke cyclus van het programma moet ook de diagnose van het netwerk gebeuren. Dit wordt verder uitgelegd in 3.1.3 Connectie controle. De input byte wordt bepaald door de waarde van de telegramteller op te tellen bij het startadres.

$$IB = \#Telegram + StartAdres$$

Telkens de telegramteller gelijk is aan het aantal te tellen waarden, wordt de messageteller met 1 verhoogd. De plaats in de buffer DB is afhankelijk van de telegramteller en de messageteller. De plaats kan berekend worden via volgende formule:

$$DB_{plaats} = (\text{aantal telegrams} + 2)$$

Na het berekenen van de DB plaats wordt het adres omgezet naar een pointerwaarde. Pointers worden gebruikt bij indirecte adressering. Om een woord om te zetten naar een pointer moet het eerst omgezet worden naar een dubbel woord (DW). In een pointer worden de 3drie minst beduidende bits gebruikt om naar één enkele bit in een byte te verwijzen. Aangezien enkel volledige ingangsbytes ingelezen worden moeten deze drie bits een 0 bevatten. Dit gebeurt door het dubbelwoord drie plaatsen naar links te verschuiven.

Nadat de pointers zijn aangemaakt, worden deze gebruikt om de ingangsbyte te transfereren naar de juiste datablokbyte. Op het einde van elke cyclus zal FC3 het voorgaande telegram uit de buffer kopiëren naar de pc datablok waar het C# programma de gegevens kan opvragen en verwerken.

### 3.1.3 Connectie controle

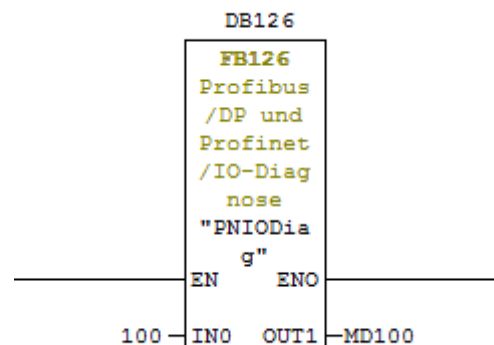
Voor de controle van de communicatie tussen meetinstrument en PLC maakt de Siemens PLC gebruik van de functie FB126 “PNIODiag” (Profibus/DP und Profinet/IO-Diagnose) die in staat is om zowel Profibus alsook Profinet netwerken te controleren. Deze functie is ontworpen door Siemens zelf en kan gedownload worden van de “industry online support international” website (<https://support.industry.siemens.com/cs/document/98278878/diagnostic-package-pniodiag-for-diagnostic-evaluation-for-profibus-dp-and-profinet-io-%28step-7-v5-x%29?dti=0&lc=en-WW>).

Deze functie moet opgeroepen worden in volgende organisatieblokken:

- OB1;
- OB82 I/O FLT1;
- OB83 I/O FLT2;
- OB86 RACK FLT;
- OB100 COMPLETE RESTART.

Bij het oproepen van FB126 moet er een instantie DB126 aangemaakt worden.

De functie zelf heeft een extra DB nodig waar de diagnosegegevens worden opgeslagen. In Figuur 32 is dit DB100. De DB100 is enkel zichtbaar in het geheugen van de PLC door gebruik te maken van de accessible nodes functie. Hij zal dus niet te zien zijn in de offline symatic manager.



Figuur 32: FB126 PNIOdiag

Om de connectie met een specifiek toestel in een bepaald netwerk te controleren, moet eerst het adres van dit toestel berekend worden. Dit kan met de formule weergegeven in Figuur 33.

$$\text{Address [Byte]} = (\text{system number}-1) * 256 * 2\text{Byte} + (\text{station number}-1) * 2\text{Byte}$$

Figuur 33: Adres formule Siemens PNIOdiag [8, p. 14]

Elk van deze bytes bestaat uit de volgende gegevens in onderstaande volgorde van LSB. Tot MSB:

- Communication OK;
- Communication maintenance;
- Communication faulty;
- Communication failed;
- Communication was deactivated
- Communication DPV1;
- Communication was maintenance;
- Communication was faulty
- Communication had failed
- Communication deactivated
- Communication reserved;
- Communication reserved;
- Communication reserved;
- Communication reserved;
- Communication reserved.

Deze waarden worden in de PLC opgeslagen in DB1 samen met de cyclustijden van het volledige programma en de waarden van de systeemklok. Deze waarden kunnen door het C#-programma uitgelezen worden.

### 3.1.4 Verbinding met pc

De verbinding met de pc gaat via ethernetverbinding tussen de CPU van de Siemens PLC en de netwerkkaart van de pc. De functie FC3 "WRITE\_TO\_PC" gaat ervoor zorgen dat iedere keer er een nieuwe boodschap in de buffer wordt geschreven, de vorige boodschap gekopieerd wordt naar DB3. Op de plaats waar FC3 wordt opgeroepen, kan ook bepaald worden hoeveel waarden er tussen de nieuwe boodschap en die voor de pc mogen zijn (vb. als de offset = 1 zal de pc waarde 1 boodschap achterliggen op het meetinstrument).

Het C# programma kan verbinding maken met de PLC door middel van de DotNetSiemensPLCToolBoxLibrary.dll wat beschikbaar is op de GitHub website en vrij te gebruiken is.

### 3.2 Allen Bradley

De programmatie van de Allen-Bradley PLC gebeurt door middel van het studio 5000 softwarepakket. De connectie tussen de PLC, pc en andere externe toestellen gebeurt met RSLinx software.

De testmodule bestaat uit een compactlogix L30ER die over twee geïntegreerde RJ45 poorten beschikt voor Ethernet IP communicatie. Verder is de testmodule uitgerust met een 24V DC voedingsmodule 1769-PB2. Voor het inlezen van digitale signalen zijn er twee digitale input kaarten 1769-IQ16 en 1769-IQ16F waarbij de F staat voor hoge snelheid. Als laatste beschikt de testmodule over een 1769-OB23 kaart waarmee de testmodule drieëntwintig digitale uitgangen kan aansturen.

Onderstaande Figuur 34 geeft de gebruikte Allen Bradley PLC weer.



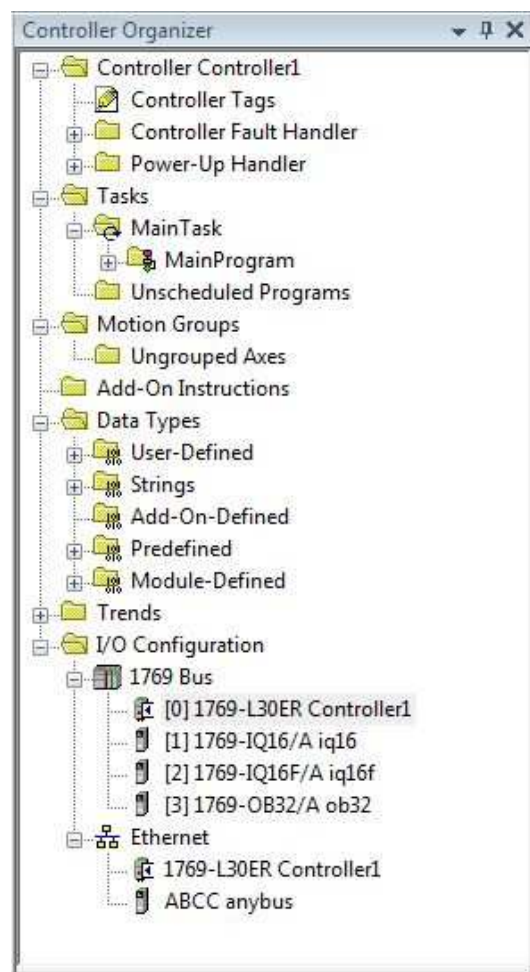
Figuur 34: Allen-Bradley L30ER

### 3.2.1 Configuratie

Allereerst moet het RSLinx programma een driver aanmaken die de pc met de juiste poort of netwerkkaart laat zoeken naar beschikbare toestellen. In de testopstelling is er gebruik gemaakt van de interne netwerkkaart (Intel® 82579LM Gigabit Network Connection).

Deze netwerkkaart zal enkel naar beschikbare toestellen zoeken in het lokale subnet.

Wanneer de driver is aangemaakt, kan die als “Communications Path” geselecteerd worden in de Studio 5000 software. Eenmaal dit gebeurd is, kan de gebruiker de modules van de PLC toevoegen in de map I/O configuratie van de “Controller Organizer”. Ook kunnen de Ethernet IP toestellen bijgevoegd worden. De I/O configuratie voor de testopstelling is te zien in Figuur 35.



Figuur 35: Controller Organizer testopstelling

De I/O configuratie bestaat uit een 1769 bus, wat vergelijkbaar is met de “RACK” in de Siemens configuratie, en een Ethernetverbinding. De 1769 bus is onderverdeeld in de verschillende modules waaruit de PLC is opgebouwd. In het Ethernetnetwerk bevinden zich de PLC (1769- L30ER Controller1) en het meetinstrument (ABCC anybus).

In de eigenschappen van de controller kan het IP adres ingesteld worden. Voor de testopstelling is dit ingesteld op 192.168.2.100.

### 3.2.2 Telegram inlezen

Net zoals bij de Siemens zal de Allen-Bradley de ontvangen data eerst bufferen alvorens het volledige telegram naar de pc te verzenden. Dit om onvolledige data in de pc te voorkomen.

De buffer is opgebouwd uit twintig “User Defined dataTypes” of UDT’s genaamd “messages”.

Elke message bestaat uit 300 SINT (Signed Integers) telegrams. Er is in de buffer dus plaats voor 20 \* 300 SINT waardes.

Om de waarden in te lezen, moet er in het hoofdprogramma “MainRoutine” een sprong gemaakt worden naar het subroutine “read\_inputs” met de JSR functie. Deze functie moet een paar waarden meekrijgen:

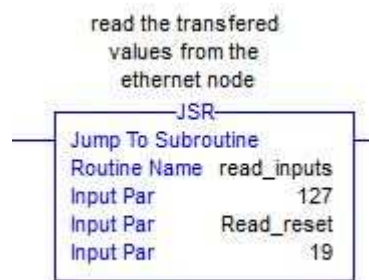
- Routine name = read\_inputs;
- Input Par = aantal waardes;
- Input Par = reset;
- Input Par = aantal messages in de buffer.

De buffer is aangemaakt in de programmatags. De tagnaam is als volgt opgesteld:

buffer[nummer van de boodschap 0 tot 20].telegram[nummer van het telegram 0 tot 300]. De Allen-Bradley PLC telt vanaf 0, dus de eerste waarde bevindt zich op plaats 0.

Bijvoorbeeld de eerste waarde van de derde boodschap is te vinden in de tag buffer[2].telegram[0].

Onderstaande Figuur 36 geeft de JSR functie weer met de nodige instellingen.



Figuur 36: JSR naar read\_inputs

In bovenstaande figuur is een voorbeeld te zien voor het communiceren met een Castemp Wireless, welke gebruik maakt van een ABCC-EIP Ethernet module. De eerste input parameter bepaalt het aantal waarden die het meetinstrument doorstuurt. De tweede parameter biedt de mogelijkheid om het programma te resetten. Dit betekent dat van zodra het resetsignaal weer wegvalt, de eerste waarde van de boodschap wordt opgeslagen op de eerste plaats in de buffer.

De laatste parameter laat het programma weten hoeveel plaatsen er zijn om boodschappen op te slaan in de buffer.

## **Werking**

Het volledige programma is beschikbaar in de bijlagen. Allereerst worden de ingangsparemeters ingelezen. Hierna start de pulstimer, die zal elke 10 ms een signaal geven. Om een telegram van 127 bytes in te lezen zal het programma dus  $127 * 10 \text{ ms} = 1,27 \text{ s}$  nodig hebben.

Vervolgens start de eerste teller. Deze teller zal het aantal telegrammen bijhouden die vergeleken worden met het ingestelde aantal telegrammen per boodschap. Deze telegrammen zijn ingesteld door de gebruiker. Iedere keer de teller deze waarde bereikt heeft, zal de teller de boodschap index met één verhogen.

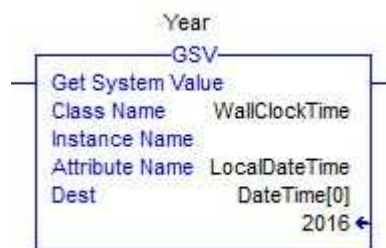
De waarden van het meetinstrument kunnen, met beide tellerwaarden in de buffer, ingelezen en gelogd worden. Dit kan door middel van de move instructie. Welke anybus:I.Data[tellegram teller] verplaatst naar de buffer[boodschap teller].telegram[telegram teller].

Tot slot zal het programma de tellers resetten als hun maximum waarde is bereikt, of als de ingangsparemeters van de reset hoog is. Hierna wordt het programma om de gegevens, bedoeld voor de pc, weg te schrijven, gestart.

### 3.2.3 Connectie controle

Om de connectie tussen het meetinstrument en de PLC te controleren, kan de pc, in het geval dat er een Anybus module is gebruikt, de controllertags raadplegen om na te gaan of er zich fouten bevinden op de connectie. Dit is mogelijk door de tag “anybus:I.ConnectionFaulted”. Welke de booleaanse waarde “true” geeft wanneer er problemen zijn met de connectie.

Als extra controle houdt de PLC ook de interne systeemtijd bij. Deze wordt opgevraagd via de GSV functie (Get System Value). De systeemtijd bevindt zich in de klasse “WallClockTime” met attribuut naam “LocalDateTime”. Deze waarden zijn opgeslagen in een array van zeven dubbele integers (DINT[7]). In de GSV functie moet enkel de eerste plaats ingesteld zijn. De rest is automatisch aangevuld. De interface van de functie is weergegeven in Figuur 38.



Figuur 37: GSV functie voor de systeem tijd

De gegevens worden dan automatisch als volgt opgeslagen.

- Date Time[0] = jaar;
- Date Time[1] = maand;
- Date Time[2] = dag;
- Date Time[3] = uur;
- Date Time[4] = minuten;
- Date Time[5] = seconden
- Date Time[6] = milliseconden.

### 3.2.4 Verbinding met pc

Net zoals bij de Siemens PLC zal tijdens het inlezen van de nieuwe boodschap, de voorgaande boodschap op een aparte plaats in het PLC geheugen worden opgeslagen. Hierdoor kan de C#.NET applicatie de data snel terugvinden door de “pc\_connect” routine.

Het subprogramma heeft de volgende drie parameters:

- de “telegram\_count” parameter,
- de “message\_nmbr” parameter,
- de offset.

Aan de “telegram\_count” parameter moet de telegramteller gekoppeld zijn. Zo weet het programma welk telegram er ingelezen is. De “message\_nmbr” bepaalt welke boodschap verplaatst moet worden. De laatste parameter is de offset die bepaalt hoeveel boodschappen er zich tussen de huidige en de pc - boodschap bevinden. De boodschap voor de pc wordt bepaald door de waarde van de boodschapteller af te trekken van de offset. Als deze waarde kleiner is dan 0, dan zal het maximale aantal boodschappen opgeteld worden met de waarde kleiner dan 0.

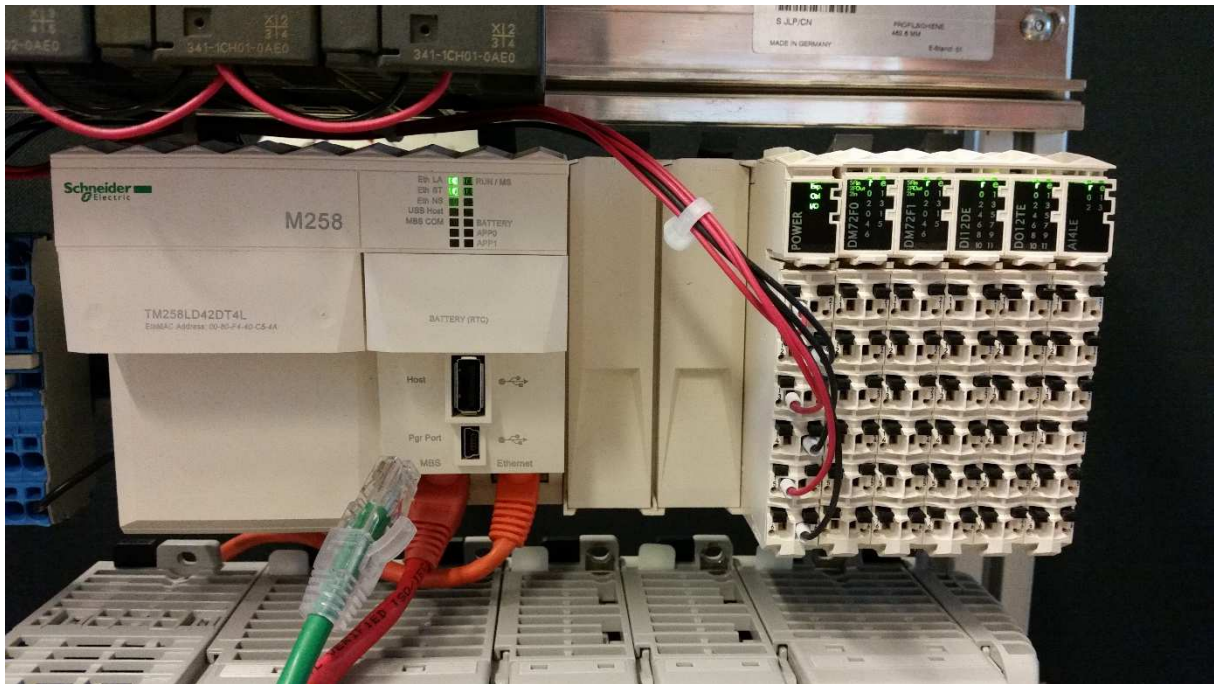


Hierna wordt het telegram, de vorige boodschap, verplaatst naar de geheugenplaats voor pc-waarden. Dit door de move functie met de volgende tags: buffer[vorige boodschap].telegram[tellegram teller] en pc.telegram[tellegram teller].

De verbinding tussen de Allen-Bradley en de C#.NET applicatie gebeurt door een Ethernetverbinding tussen PLC en pc. De applicatie kan deze connectie verwerken dankzij de drivers en dll's van Advanced HMI.

### 3.3 Schneider

De programmatie van de Schneider PLC gebeurt door middel van het software pakket SoMachine V3.1. De testmodule bestaat uit een M258 PLC, meer bepaald een TM258LD42DT4L. Deze PLC is standaard uitgerust met een 24V DC voeding, een CPU met 2 ingebouwde USB poorten (1 gewone USB poort en 1 mini USB). Ook beschikt de CPU over twee ingebouwde RJ45 poorten, waarvan er een gebruikt kan worden voor MODBUS RTU en MODBUS ASCII (MBS poort) en de andere voor Ethernetverbinding met de pc en MODBUS TCP. Onderstaande Figuur 38 geeft de huidige testmodule weer.



Figuur 38: Testmodule M258

Belangrijk om op te merken, is dat de RJ45 poort voor Ethernet enkel gebruikt kan worden om verbinding te maken met een pc, dus niet om te communiceren met Ethernet Slaves zoals de Allen Bradley PLC. Dit omdat de M258 niet ontworpen is als Ethernet Master. De PLC kan echter wel als Ethernet Slave ingesteld worden om zo verbinding te maken met een Ethernet Master.

### 3.3.1 Configuratie

Voor de configuratie onderscheiden we drie delen.

- configureren van de PLC;
- configureren van MODBUS RTU;
- configureren van MODBUS TCP.

#### PLC

Tijdens de eerste configuratie van de PLC heeft deze al standaard een IP-adres. Dit is afhankelijk van het MAC-adres van de PLC. Het standaard IP adres vormt men als volgt: 10.10.(voorlaatste getal van het MAC-adres omgevormd van hexadecimaal naar decimaal). .(laatste getal van het MAC-adres omgevormd van hexadecimaal naar decimaal).

Het MAC-adres van de testopstelling is 00-80-F4-40-C5-4A waarvan de twee laatste getallen, omgerekend naar decimale waarde, respectievelijk 197 en 74 bedragen. Het standaard IP adres is dan 10.10.197.74. Voor de test opstelling is dit veranderd naar 192.168.2.102.

Het aanpassen van deze instellingen kan via de SoMachine software in de configuratie tab.

Onderstaande Figuur 39 geeft de volledige ethernetinstelling van de testopstelling.

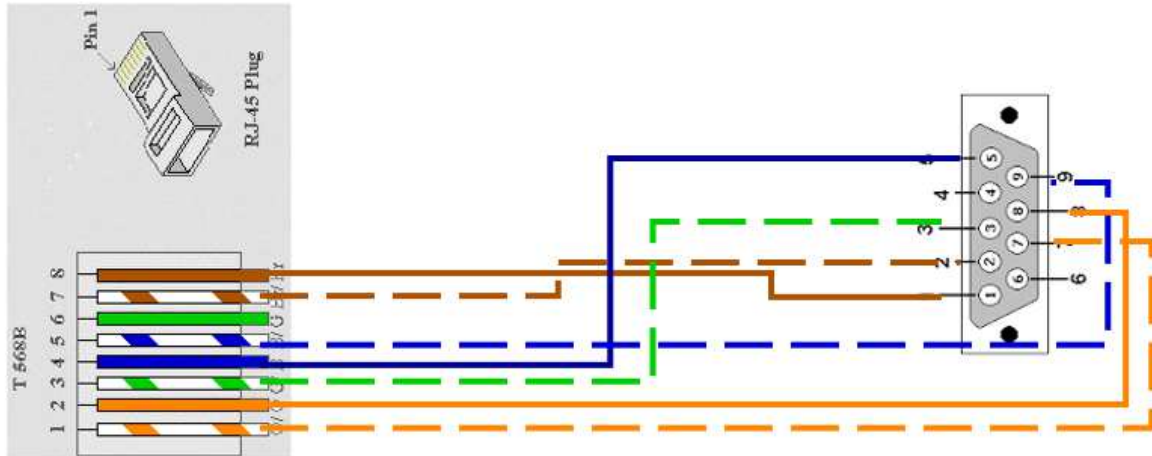
The screenshot shows a configuration window titled "Configured Parameters". It contains the following fields and options:

- Interface Name: ether\_0
- Network Name: my\_Device
- IP Address configuration options:
  - IP Address by DHCP
  - IP Address by BOOTP
  - fixed IP Address
- IP Address: 192 . 168 . 2 . 102
- Subnet Mask: 255 . 255 . 255 . 0
- Gateway Address: 0 . 0 . 0 . 0
- Transfer Rate: Auto
- Ethernet Protocol: Ethernet 2 (dropdown menu)
- Web Server active

Figuur 39: Ethernet instellingen Schneider

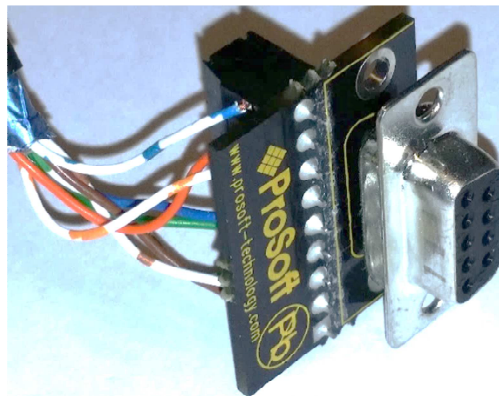
## MODBUS RTU

Door het feit dat de M258 PLC een RJ45 poort gebruikt, voor de seriële MODBUS communicatie en de meeste toestellen een 9-pins DB9 stekker, moest er tijdens de masterproef eerst een passende kabel gemaakt worden. Dit was noodzakelijk aangezien deze kabels niet door de fabrikant (Schneider) noch door de toeleverancier (Automotion nv.) geleverd kon worden. Onderstaande Figuur 40 geeft het aansluitschema van beide connectoren.



Figuur 40: Aansluitschema RJ45 naar DB9 kabel

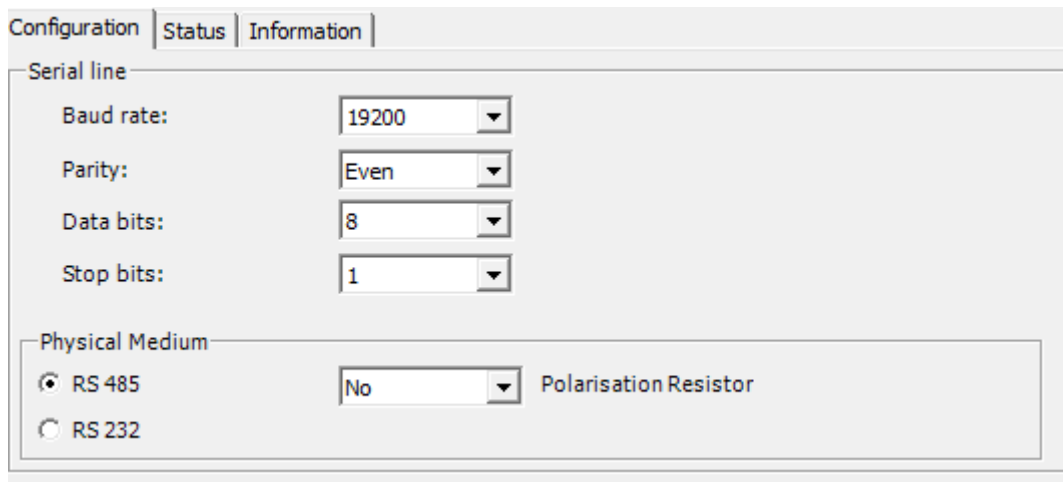
Tijdens de masterproef is dit eerst getest met een DB9 poort van Prosoft, die voorzien is van schroefconnectoren, wat het aanpassen van de verbinding vereenvoudigt. Figuur 41 geeft de test DB9 connector weer van Prosoft.



Figuur 41: Prosoft DB9 connector

Om de PLC in te stellen om MODBUS RTU transmissies te ontvangen, kan de PLC gebruik maken van de “MODBUS I/O scanner”, die gekoppeld is met een “Generic MODBUS Slave”. Om deze toe te voegen, moet men eerst de seriële lijn van de PLC instellen. De instellingen van de seriële lijn zijn afhankelijk van de instellingen van het meetinstrument en moeten overeenkomen.

Als voorbeeld geeft Figuur 42 de instelling weer voor MODBUS RTU communicatie met een DigiTemp-E.



Configuration | Status | Information

Serial line

Baud rate: 19200

Parity: Even

Data bits: 8

Stop bits: 1

Physical Medium

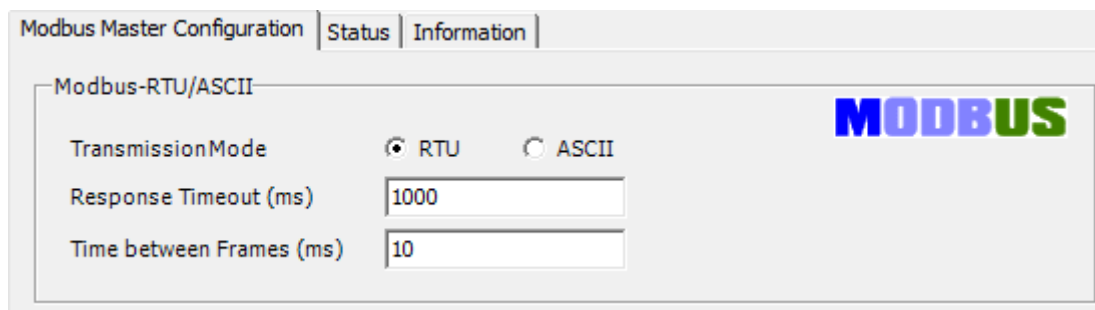
RS 485 No Polarisation Resistor

RS 232

Figuur 42: Instellingen MODBUS RTU communicatie met E-Line DigiTemp

De instellingen bestaan uit de transmissiesnelheid (Baud rate), de pariteitscontrole (even of oneven waarde), het aantal gebruikte databits en het aantal gebruikte stopbits. Ten slotte vraagt het programma om aan te geven als het om een RS458 verbinding of een RS232 verbinding gaat, al dan niet met afschakelweerstand.

Vervolgens wordt de MODBUS I/O scanner geconfigureerd. Deze scanner kan zowel in RTU als in ASCII modus worden ingesteld, waarbij ook de responstijd in ms en de tijd tussen de frames in ms wordt gevraagd. Figuur 43 toont de instelling voor de testopstelling met een DigiTemp-E.



Modbus Master Configuration | Status | Information

Modbus-RTU/ASCII

Transmission Mode  RTU  ASCII

Response Timeout (ms) 1000

Time between Frames (ms) 10

**MODBUS**

Figuur 43: Instellingen MODBUS I/O scanner

De configuratie voor de “Generic MODBUS Slave” moet een slave-adres krijgen wat identiek is aan het slave-adres in het instrument. De generic MODBUS Slave werkt met kanalen. In elk kanaal kan ingesteld worden wat het verbindingstype is, welke trigger het kanaal gebruikt, de offset om het eerste register te bepalen en de lengte van het register.

Het voorbeeld in Figuur 44 geeft de instelling voor MODBUS RTU connectie met een E-Line Digttemp weer.

Figuur 44: Instellingen MODBUS RTU kanaal

In bovenstaand voorbeeld is er gebruik gemaakt van functiecode 3 die enkel gegevens uit de registers kan lezen. Hier kan ook gekozen worden voor functie code 16 die enkel waarden naar de registers gaat schrijven, of functiecode 23 die een combinatie is van lezen en schrijven. Aangezien we geen instellingen of commando's naar de DigiTemp-E kunnen zenden is enkel functiecode 3 voldoende, maar voor toekomstige toestellen kunnen de andere 2 codes ook mogelijk zijn.

Voor de trigger wordt een cyclische tijd van 100 ms ingesteld. Dit betekent dat met dit interval telkens automatisch de waarden uit het register worden gelezen, waarna ze worden opgeslagen in de I/O mapping. De I/O mapping bepaalt aan welke ingangs- of uitgangswaarden de waarden worden gekoppeld, vb. de eerste waarde van kanaal 1 koppelt de I/O mapping aan %IW70.

De registeroffset bepaalt het eerste register dat uitgelezen wordt. Deze waarde moet een hexadecimale waarde zijn en deze is terug te vinden in de handleiding van het instrument samen met de lengte van het register. Vb. in de handleiding van de E-Line Digttemp is volgende Tabel 6 gegeven.

Tabel 6: MODBUS RTU registers E-Line Digttemp [8]

Range	Contents	Notes
0000h...00FFh	Read Process Data	PLC collects 128 data bytes starting from register 40256

Hierboven is te zien dat er 128 bytes beschikbaar zijn vanaf register 40256. Dit laatste getal betekent dat er functiecode 4 is gebruikt (Read input registers) vanaf register 256. De offset is dus gelijk aan 256 maar in hexadecimale waarde wordt dit 0x0100. De lengte wordt bepaald door het formaat waarin de gegevens worden opgeslagen. Aangezien de gegevens volgens de mapping worden opgeslagen in woordformaat (1 woord = 2 bytes) is de lengte de helft 64.

## **MODBUS TCP**

Voor de configuratie van MODBUS TCP communicatie is er voor de M258 geen I/O scanner beschikbaar en dus ook geen “Generic Slave”. Het enige wat ingesteld moet zijn, is het IP adres van de controller, wat besproken is in het begin van dit hoofdstuk. Verder wordt de verbinding tot stand gebracht door acyclische communicatie en de daarbij behorende functies die in het volgende hoofdstuk besproken worden.

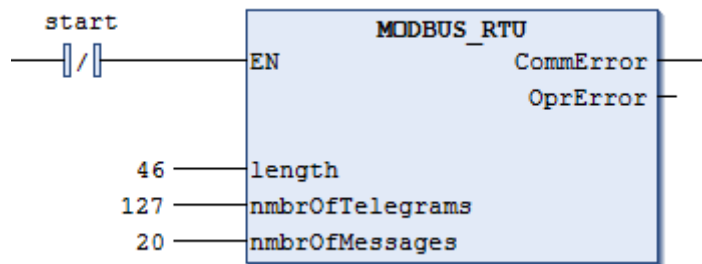
### 3.3.2 Telegram inlezen

Voor het inlezen zijn er twee verschillende programma's geschreven: één voor het inlezen via MODBUS RTU en een tweede voor MODBUS TCP. Dit is gebeurd om de reden dat de gebruikte functies en de nodige gebruikersinstellingen te veel van elkaar verschillen om ze in één programma samen te vatten.

#### MODBUS RTU

Het grootste gedeelte van het inlezen van de gegevens via MODBUS RTU gebeurt al automatisch door de I/O scanner. Deze zal automatisch de juiste functies oproepen om de waarden in te lezen en deze op te slaan in de variabelen die ingesteld zijn in de I/O mapping. In het geschreven programma wordt data opgeslagen in de globale variabele `generic_rtu_slave`, wat een array is met een grootte van 46 woorden.

Het programma kan opgeroepen worden in het hoofdprogramma (POU). De interface ziet er uit zoals in onderstaande Figuur 45.



Figuur 45: Interface voor het MODBUS\_RTU programma

Via de interface kunnen volgende instellingen gedaan worden.

- EN: activeren van het programma;
- Length: de lengte van het woord waar de ingelezen waarden zijn opgeslagen;
- nmbrOfTelegrams: het aantal telegrammen wat in de buffer worden opgeslagen;
- nmbrOfMessages: het aantal boodschappen wat in de buffer worden opgeslagen;
- CommError: geeft de errorcode voor een communicatie fout;
- OprError: geeft de errorcode voor een operatie fout.

Bij de start van het programma worden er twee pulstimers gestart, waarvan de tweede dubbel zo snel gaat als de eerste. Dit is gedaan omdat er, per woord dat het programma inleest, telkens 2 bytes naar de buffer gaan. De tragere pulstimer zal bij elke puls de waarde van de "inputCount" teller verhogen tot deze dezelfde waarde heeft als de ingestelde variabele "length". De snellere timer zal telkens de telegramCount verhogen tot die de waarde van "nmbrOfTelegrams" heeft bereikt. Als dit gebeurt, wordt de waarde van de "messageCount" verhoogd tot die de waarde van "nmbrOfMessages" heeft bereikt.

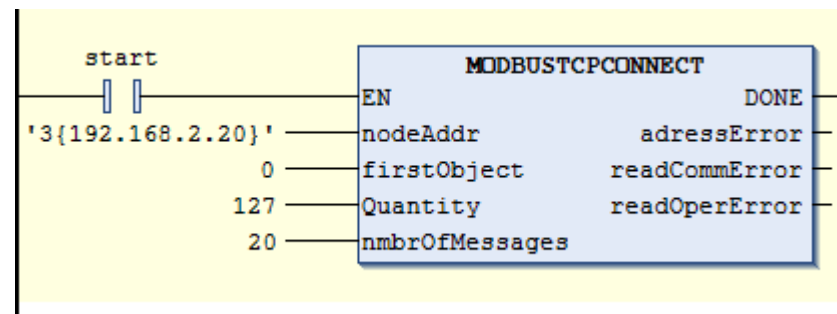
Vervolgens zet het programma de eerste byte van het ingelezen woord om met de WORD\_TO\_BYTE functie. Het ingelezen woord wordt bepaald door de tag `GVL.generic_rtu_slave[waarde van de input teller]`. Hierna worden de bits in het woord acht plaatsen naar rechts verschoven zodat de tweede byte op de plaats van de eerste komt te staan. Na het verschuiven zal de WORD\_TO\_BYTE functie het nieuwe woord omzetten naar een byte om zo de tweede waarde te bekomen.



Ten slotte zullen twee MOVE functies de twee omgezette bytes naar de buffer verplaatsen. Waarna de cyclus opnieuw start en het volgende woord ingelezen en verwerkt wordt. De plaats in de buffer is bepaald door de telegramteller en de boodschappenteller. De tagnaam is als volgt opgesteld: GVL.buffer.buffer[waarde van boodschappenteller].telegrams[waarde van de telegramteller].

## MODBUS TCP

De instellingen voor de communicatie via MODBUS TCP zullen voornamelijk in de interface van het programma “ModbusTCPConnect” instelbaar zijn. Het programma kan opgeroepen worden in het hoofdprogramma (POU) zoals wordt aangegeven in Figuur 46.



Figuur 46: MODBUSTCPCONNECT interface

Met de interface kunnen volgende instellingen gedaan worden

- EN: activeert het programma;
- nodeAddr: het adres van de Slave in STRING formaat;
- firstObject: start register;
- Quantity: de hoeveelheid gegevens die verwacht wordt;
- nمبرOfMessages: het aantal boodschappen die in de buffer kunnen worden opgeslagen;
- DONE: wordt TRUE telkens er een waarde succesvol is ingelezen;
- adressError: geeft de error code als er een probleem is met het Slave adres;
- readCommError: geeft de error code als er een probleem is met de communicatie;
- readOperError: error code voor problemen met de lees operatie.

Het Slave adres bestaat uit de communicatiepoort nummer (3 voor de geïntegreerde Ethernet poort) gevolgd door het IP adres ('<communicatiepoort nummer>{<IP adres A.B.C.D>}'). De standaard TCP poort 502 wordt ook gebruikt bij MODBUS communicatie. Het is echter ook mogelijk om via een andere poort te gaan door deze toe te voegen aan het adres ('<communicatie poort nummer>{<IP adres A.B.C.D>:<poort>}'). Indien de communicatie de standaard poort gebruikt moet deze niet vermeld worden in het adres.

De parameters firstObject en Quantity bepalen welke registers uitgelezen worden uit de Slave. In het bovenstaande voorbeeld zijn de instellingen weergegeven voor een DigiTemp-E waarbij volgens de handleiding het MODBUS TCP register start vanaf 30000, wat wil zeggen dat voor het uitlezen van de gegevens functiecode 3 (read holding registers) gebruikt en dat het register start vanaf 0. Deze start waarde moet in een hexadecimale waarde staan, maar in dit geval blijft dit 0. De lengte is afhankelijk van het formaat. In het voorbeeld met de DigiTemp-E zijn dit 127 bytes.

De werking van het programma verloopt als volgt: allereerst worden de ingangsparameters voor de leesfunctie opgeslagen in een tijdelijk geheugen. Vervolgens start een pulstimer die een kloksignaal genereert, dat verderop in het programma gebruikt wordt. Hierna start het inlezen van de waarden.

Het inlezen van de waarden start met het omzetten van het Slave adres van een STRING formaat naar een ADDRESS formaat door de ADDM-functie. Met dit omgezette formaat kan de READ\_VAR functie verbinding maken met de Slave. De READ\_VAR functie heeft ook een time-out tijd nodig om te bepalen wanneer de connectie te veel tijd nodig heeft en er dus een fout opgetreden is. De parameter "ObjType" bepaalt de functiecode die de READ\_VAR functie moet gebruiken (functiecode 3 staat voor read holding registers). Verder heeft de functie de ingangsparementers nodig voor het eerste object en de hoeveelheid.

Ten slotte heeft de functie een array nodig om alle gegevens in op te slaan. Dit is niet dezelfde array als de buffer, hier wordt een tijdelijke buffer gebruikt.

Na het inlezen van de Slaveregisters starten de tellers voor de telegrammen en voor de boodschappen. Met deze tellerwaarden kunnen daarna, door gebruik te maken van een MOVE instructie, de gegevens van de tijdelijke array naar de globale buffer verplaatst worden. Dit kan door de tag READBuffer[waarde van de telegramteller] en GVL.buffer.buffer[boodschapteller].telegrams[telegramteller].

### 3.3.3 Connectie controle

Om de verbinding tussen de PLC en het meetinstrument te controleren, koppelen de programma's de errorcodes van de leesfunctie terug naar het hoofdprogramma. De teruggekoppelde waarden zijn:

- communicatie error;
- operatie error;
- adres error (enkel voor MODBUS TCP).

De communicatie error code bevat informatie over de communicatie diagnoses, zoals onderbrekingen en gedetecteerde errors. Volgende waarden in Tabel 7 kunnen voorkomen.

Tabel 7: Communication errorcodes

Fout	Code (HEX)	Beschrijving
CommunicationOK	00	De verbinding is OK.
TimedOut	01	De verbindingstijd overschrijdt de time-out tijd.
Canceled	02	De verbinding is gestopt door de gebruiker.
BadAddress	03	Het adres formaat is incorrect.
BadRemoteAddr	04	Het remote adres is incorrect.
BadMgtTable	05	Het management tabel formaat is incorrect
BadParameters	06	Specifieke parameters zijn onjuist.
ProblemSendingRq	07	Problemen met de aanvraag verzending.
RecvBufferTooSmall	09	De ontvangstbuffer is te klein.
SendBufferTooSmall	0A	De verzendbuffer is te klein.
SystemResourceMissing	0B	Er mist een systeem bron.
BadTransactionNb	0C	De transactie nummer is onjuist.
BadLength	0E	De lengte is onjuist.
ProtocolSpecificError	FE	De operation error code bevat meer informatie.
Refused	FF	De boodschap is afgewezen.

De operatie errorcodes zijn afhankelijk van de waarde van de communicatie error. Wanneer de communicatie error gelijk is aan 00 geldt volgende Tabel 8.

Tabel 8: Errorcodes voor operatie error wanneer communicatie error 00 is

Fout	Code (HEX)	Beschrijving
OperationOK	00	De overdracht is geldig.
NotProcessed_or_TargetResourceMissing	01	De aanvraag werd niet behandeld.
BadResponse	02	Het ontvangen antwoord is onjuist.

Wanneer de communicatie error gelijk is aan FF geldt volgende Tabel 9.

*Tabel 9: Tabel errorcodes operation error als communicatie error FF is*

<b>Fout</b>	<b>Code (HEX)</b>	<b>Beschrijving</b>
NotProcessed_or_TargetResourceMissing	01	Het beoogde systeem is niet aanwezig
BadLength	05	De lengte is onjuist.
CommChannelError	06	Er is een probleem met de communicatie
BadAddr	07	Het adres is onjuist.
SystemResourceMissing	0B	Er mist een systeembron.
TargetCommInactive	0C	De communicatie van het doel is niet actief
TargetMissing	0D	Het doel is afwezig.
ChannelNotConfigured	0F	het kanaal is niet geconfigureerd.

Als de communicatie error gelijk is aan FE, wil dit zeggen dat de fout protocol specifiek is. Er moet in de handleiding van het instrument gekeken worden voor een oplossing. De adres errorcodes zijn enkel van toepassing bij MODBUS TCP en hebben dezelfde betekenis als Tabel 7. Maar deze errors hebben dan te maken met de omvorming van het adres door de ADDM-functie.

#### 3.3.4 Verbinding met pc

Voor de verbinding tussen de Schneider PLC en de pc is er momenteel nog geen werkende oplossing. Er is onderzoek gedaan naar twee mogelijke manieren, namelijk een open source bibliotheek NModbus en een “third party” oplossing wat door een extern bedrijf op maat gemaakt wordt. Beide oplossingen worden verder besproken in 3.4.2 Gevolgde stappen.

## 3.4 Csharp Applicatie

### 3.4.1 Introductie .NET

#### **Wat is C#?**

C#, uitgesproken als C-sharp, is een objectgeoriënteerde programmeertaal ontwikkeld door Microsoft als onderdeel van het .NET-initiatief, en later geaccepteerd als standaard door ECMA (ECMA-334) en ISO (ISO/IEC 23270). Het is een taal die ontwikkelaars in staat stelt om veilige en robuuste applicaties te bouwen die draaien op het .NET Framework

C# wordt gebruikt om applicaties te maken, zoals:

- Windows client applicaties,
- XML-webservices,
- gedistribueerde componenten,
- client-server applicaties,
- database applicaties,
- enzovoort

Visual Studio biedt een geavanceerde code-editor, een handige interface designers, geïntegreerde debugger, en vele andere hulpmiddelen die het gemakkelijker maken om toepassingen op basis van de C# taal en het .NET Framework te ontwikkelen.

C# ondersteunt, als een objectgeoriënteerde taal, de concepten van inkapseling, overerving en polymorfisme. Een klasse kan direct erven van één oudere klasse en kan meerdere interfaces implementeren.

#### **Wat is .NET?**

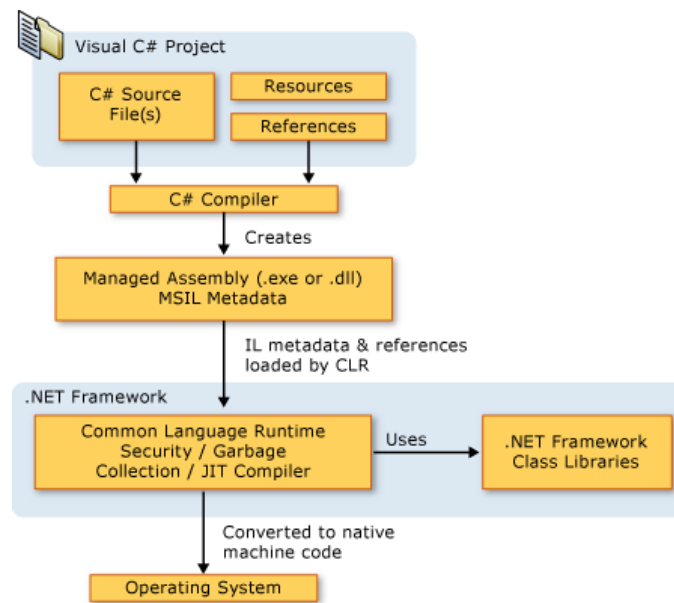
.NET is een applicatieframework ontwikkeld door Microsoft, ten behoeve van verschillende programmeertalen. .NET is volledig beschreven in publieke ECMA-standaarden.

Het .NET Framework bevat standaard klassebibliotheken. Deze zitten georganiseerd in een hiërarchie van de namespace. De meeste ingebouwde Application Programming Interfaces (API's) maken deel uit van de Microsoft-namespace of de System-namespace. Deze klassebibliotheken implementeren een groot aantal handig, veelgebruikte functies, zoals een bestand lezen en schrijven, grafische weergave, interactie met de database en manipulatie van XML-documenten.

De .NET-klassebibliotheken zijn beschikbaar voor alle door de Common Language Infrastructure (CLI) gebruikte talen:

- ASP.NET,
- C#,
- C++.NET,
- Delphi .NET,
- Delphi Prism,
- F#,
- J#,
- Visual Basic.NET,
- Vulcan.NET,
- IronPython.

Figuur 47 geeft de Architectuur weer van het .NET Framework Platform



Figuur 47: Architectuur van het .NET Framework [31]

Zoals bovenstaande figuur schematisch weergeeft, bestaat een C#-project uit de Source files, Resources en References. Deze worden gecompileerd tot een Intermediate Language (IL), conform met de CLI. De IL-code en Resources worden vervolgens opgeslagen in een assembly, een bestand met een .exe of .dll extensie.

Indien het C# programma uitgevoerd wordt, wordt de assembly in de Common Language Runtime (CLR) geladen. Als er voldaan werd aan de Security eisen dan converteert de Just In Time (JIT) compilatie de IL-code naar standaard machinetaal en wordt het programma uitgevoerd.

### Wat is WPF?

Windows Presentation Foundation (WPF) is een User Interface (UI) Framework dat de mogelijkheid biedt tot het creëren van standalone en webapplicaties. WPF breidt de kern uit met een set van applicatie-ontwikkelingsfuncties, waaronder Extensible Application Markup Language (XAML), controls, data binding, lay-out, 2-D en 3-D graphics, animatie, stijlen, sjablonen, documenten, media, tekst en typografie.

De fundamenteën van WPF programmatie komen overeen met die van ASP.NET en Windows Forms. WPF biedt verbeteringen voor Windows client-applicatie ontwikkeling. Een belangrijke verbetering is de mogelijkheid om een applicatie met behulp van zowel markup en code-behind te ontwikkelen. Algemeen wordt Extensible Application Markup Language (XAML) markup gebruikt om het uiterlijk van een toepassing te creëren en worden beheerde programmeertalen, zoals de code-behind, gebruikt om het gedrag te bepalen.

### Wat is XAML?

XAML is een Extensible Markup Language (XML) gebaseerde markup taal die gebruikt wordt voor de UI (User Interface) van een toepassing te implementeren. Hiermee wordt het creëren van een gebruikersinterface vergemakkelijkt voor de programmeur. De programmeur kan hierdoor gebruik maken van schermen, dialoogvensters, pagina's, user controls, etc.

XAML laat toe dat twee verschillende partijen, designers en programmeurs, aan één project kunnen werken. De designers ontwerpen de userinterface, zonder nood te hebben aan programmeerkennis, terwijl de programmeurs de achterliggende logica van de applicatie schrijven.

### Wat is de Code-behind?

Het belangrijkste van een applicatie is het implementeren van de functionaliteit die reageert op de gebruikersinteracties, zoals het afhandelen van events/gebeurtenissen, het oproepen van business logica en de toegang tot de gegevens logica. In WPF, wordt dit gedrag in het algemeen geïmplementeerd in de code die is gekoppeld aan de markup. Deze code is bekend als code-behind.

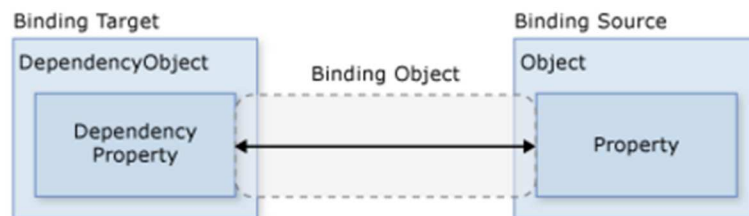
### Wat is Data Binding?

Applicaties worden gemaakt zodat gebruikers de mogelijkheid hebben om data te bekijken en te bewerken. Bij WPF-toepassingen zijn technologieën voor het opslaan en opvragen van gegevens al voorzien. Indien de data toegankelijk en in beheerde objecten geladen is, begint het harde werk voor WPF toepassingen.

In wezen gaat het over twee functies:

1. Het kopiëren van de gegevens uit de beheerde objecten naar de usercontrols, waar de gegevens weergegeven en bewerkt kunnen worden.
2. Ervoor zorgen dat de wijzigingen, door de usercontrols, van de gegevens terug naar de beheerde objecten gekopieerd worden.

WPF biedt een data binding engine aan om deze stappen automatisch uit te voeren. De kerneenheid van de data binding engine is de binding klasse, met de taak om een controle (de bindende doelstelling) naar een data-object (de bindende bron) te binden. Dit verband wordt geïllustreerd in Figuur 48.



Figuur 48: Databinding [20]

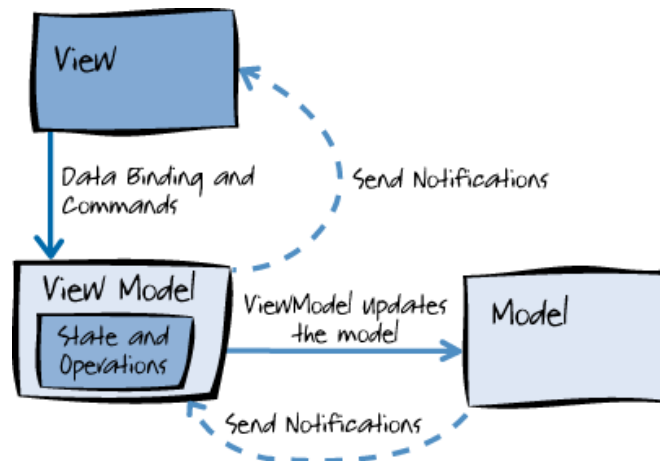
### Wat is Model-View-ViewModel (MVVM)?

Het MVVM patroon kan dienen als een vervanging voor MVC (Model-View-Controller). Het ViewModel vervangt niet zomaar de Controller. Om onafhankelijk testbaar en herbruikbaar te zijn, mag een ViewModel geen idee hebben wat de View weergeeft en mag hij geen idee hebben waar de data vandaan komt. De ViewModel is als het ware een domme connectie tussen de Programmeur(Code) en de ontwerper(UI).

Bij MVVM bevat de VM commands die beslissen wat er gebeurt op basis van ontvangen informatie van de View of de Model.

- Models bevatten de effectieve data;
- Views geven verschillende vormen van data weer, maar weten niet waar het vandaan komt;
- ViewModels houden data vast, waarvan zij niet weten waar het vandaan komt of waar het weergegeven wordt. De commands in de VM zorgen voor het verwerken van deze data.

Figuur 49 geeft een schematische weergave van de relaties binnen het MVVM-patroon.



Figuur 49: Schematische weergave van het MVVM-patroon [22]

De View "weet" van het bestaan van de ViewModel, en de ViewModel "weet" de Model, maar de Model is niet op de hoogte van de ViewModel, en de ViewModel is niet op de hoogte van de View. Het MVVM patroon zorgt voor een scheiding tussen applicatie logica en de UI, zodat de toepassing gemakkelijker te testen, te onderhouden en te evolueren is. Het verbetert de herbruik kansen van de code en verbetert de ontwikkelaar-designer workflow.



### 3.4.2 Gevolgde stappen

#### **Wensen van het bedrijf**

Als eerste werden de wensen van het bedrijf in kaart gebracht waarna er gezocht werd naar beschikbare oplossingen.

Het programma moet, volgens het bedrijf, voldoen aan de volgende eisen:

- Het volledige gamma meetinstrumenten van Heraeus moet getest kunnen worden.
- Dit op drie soorten PLC's: Siemens, Allen Bradley en Schneider.
- Het programma dat in staat voor de controle en logging van de data moet in de programmeertaal CSharp(C#) geschreven worden en dit in WPF(Windows Presentation Foundation) volgens de MVVM(Model-View-ViewModel) structuur.

#### **Beschikbare mogelijkheden**

Vervolgens werd de optie van SCADA-systemen bekeken. SCADA(Supervisory Control And Data Acquisition)-systemen staan in voor het verzamelen, doorsturen, verwerken en visualiseren van meet- en regelsignalen van verschillende machines in een industriële installatie.

Door de specifieke eisen van het bedrijf viel het moeilijk om een kant en klare toepassing te vinden. Aangezien bestaande SCADA-systemen:

- niet de mogelijkheid bieden tot het gebruik van C#.NET;
- beperkt zijn in het aantal te koppelen meetinstrumenten.

Op maat gemaakte SCADA-systemen kunnen deze optie wel bieden tegen een uitermate hoge prijs.

Aangezien SCADA-systemen al snel wegvielen, bleef enkel de optie over om het zelf te verwezenlijken. Dit brengt enkele mogelijkheden met zich mee om te communiceren met de PLC's:

- OPC,
- Open source drivers (gpl or lgpl licensed),
- Third-party libraries (activex, dlls and so on).

Deze mogelijkheden worden hieronder voor elke PLC overlopen.

#### **OPC**

Een OPC(OLE voor Process Control, waarin OLE staat voor Object Linking and Embedding) is de gemakkelijkste en veiligste manier om met PLC's te communiceren. Dit maakt gebruik van Closed Source bibliotheken waarmee data vergaard kan worden onafhankelijk van de gebruikte PLC.

Om een OPC te gebruiken moet er een client geprogrammeerd worden met de bibliotheken bijgeleverd met de aankoop. Er is ook de mogelijkheid om gebruik te maken van third-party clients.

De grote nadeel van OPC is dat het niet gratis is en er een kopie van een OPC Server aangekocht moet worden voor iedere pc die hieraan gekoppeld moet worden.

Hier is verder geen onderzoek naar gedaan.

## Open source drivers

### Siemens:

Voor Siemens werden snel verschillende drivers gevonden die bruikbaar kunnen zijn voor deze toepassing. De mogelijke drivers worden in onderstaande Tabel 10 aangehaald.

Tabel 10: Siemens Drivers

<b>Bibliotheek</b>	<b>Protocol</b>	<b>Programmeertalen</b>
LibNoDave	MPI, Profibus, Profinet	C#
S7.Net	Profinet	C#
Snap7	Profinet	C++, C#, Pascal, LabView, etc
DotNetSiemensPLCToolBoxLibrary	MPI, Profibus, Profinet	C#

Aangezien Profinet de keuze voor de communicatie tussen PLC en pc is, zijn alle bovenstaande drivers toepasbaar. Na verder onderzoek blijkt dat DotNetSiemensPLCToolBoxLibrary-driver een vereenvoudiging is van de LibNoDave-driver.

Om een concrete keuze te maken tussen deze drivers werd er contact opgenomen met de externe programmeurs welke voor Heraeus Elektro-Nite werken. Hieruit bleek dat één van de programmeurs al een project met een Siemens PLC heeft afgerond waarbij gebruik werd gemaakt van de bibliotheek "DotNetSiemensPLCToolBoxLibrary" uit de gelijknamige driver.

Na het ontvangen van het programma werd het in detail bestudeerd. Hierop volgde de keuze om vanuit dit bestaand programma verder te werken. Hierbij kwam veel extra zoekwerk te pas.

### Allen Bradley:

Voor Allen Bradley is er contact opgenomen met een vertegenwoordiger. Deze liet weten dat er een kaart bestond voor deze PLC-PC communicatietoepassing, namelijk de ProSoft MVI69E-LDM. Helaas bleek na ontvangst uit bijhorende documentatie dat deze kaart enkel geschikt is voor C en C++. Om zeker te zijn dat deze kaart niet van nut kon zijn, werd de fabrikant, ProSoft, van deze kaart gecontacteerd. Deze bracht het nadelige nieuws dat deze kaart geen .NET-formaten ondersteunt.

Vervolgens is er verder gezocht naar drivers om te implementeren in het C#.NET programma. De gevonden drivers worden weergegeven in onderstaande Tabel 11.

Tabel 11: Allen Bradley Drivers

<b>Drivers</b>	<b>Protocol</b>	<b>Geschikt voor:</b>
ControlLogixNET	Ethernet/IP	ControlLogix PLC's, PAC's en micro-PLC's.
TuxEIP	Ethernet/IP	ControlLogix en Micrologix via Linux
TUX DF1	Ethernet/IP	ControlLogix en Micrologix via Linux
AB DF1 Protocol	RS232	Micrologix

Aangezien het een driver moet zijn die ondersteuning biedt voor de communicatie met een Allen Bradley CompactLogix L30ER zijn bovenvermelde drivers niet van toepassing.

Een standaard Open Source Driver werd niet gevonden. Er werd wel een Open Source programma genaamd Advanced HMI gevonden die drivers bevat om te communiceren met een Allen Bradley (CompactLogix en MicroLogix) en Beckhoff TwinCAT. De drivers zelf worden niet apart aangeboden als Open Source, maar AdvancedHMI wordt aangeboden als "almost Open-Source". Het is bijna Open

Source omdat de code van de hoofd-drivers niet gegeven wordt, maar er wel wordt aangegeven hoe ze gebruikt dienen te worden.

Dit wordt op de website van de programmeur ook vermeld:

*“It contains free drivers written in Visual Basic.Net. The drivers are not open-source, but they are free and the interface to them is open-source. To use these libraries you need to import them in your C# project (not in Visual Studio Express edition, you need at least Professional edition) or build the dlls and add them as reference in your C# project.”* [9]

De gebruikte bibliotheken, die afkomstig zijn uit de “AdvancedHMI”-project, zijn:

- AdvancedHMIDrivers
- MfgControl.AdvancedHMI.Drivers

Na het toevoegen van deze bibliotheken in het C#.NET programma en het implementeren van hun functies, gaf het programma geen fouten aan waarmee het op het eerste zicht succesvol leek te zijn. Echter bleek na het uitvoeren van enkele testen dat het programma niet zelfstandig kon functioneren op een andere pc. Dit bleek na onderzoekswerk op forums en dergelijke dat er nog twee bibliotheken geïmplementeerd moesten worden, namelijk:

- AdvancedHMIControls
- MfgControl.AdvancedHMI.Controls

Van het moment dat deze bibliotheken toegevoegd waren in het C#-programma was deze fout verholpen.

### Schneider Electric

Om van een pc via Modbus TCP te communiceren met een PLC is er een bibliotheek vereist, namelijk nModbus. Deze bibliotheek is veruit een standaard voor Modbus TCP en RTU omdat het dezelfde methodes gebruikt voor beide protocollen. Door gebruik te maken van Modbus, om te communiceren met PLC's, zijn er geen licenties nodig.

Momenteel wordt er getracht om gebruik te maken van de nModbus driver. Hier is het gelukt om via het programma verbinding te maken met de Schneider PLC. Maar om data uit te lezen is er een probleem gevonden. Er wordt een startadres verwacht, maar hier is het nog onduidelijk waar er in de PLC precies wordt uitgelezen. Hiervoor is contact opgenomen met een specialist. Helaas is dit nog niet afgewerkt voor het einde van deze masterproef.

### **Third-party libraries**

Dit zijn bibliotheken die door externe firma's gemaakt worden, specifiek voor het gewenste project.

Dit geeft enkele nadelen inzake de wensen van het bedrijf:

- het limiteert de uitbreiding van het project;
- het heeft een kostprijs.

### Siemens

Third-party Libraries zijn voor Siemens zijn niet nodig aangezien er, zoals hierboven vermeld wordt, gebruik gemaakt werd van de bibliotheek “DotNetSiemensPLCToolBoxLibrary” uit het gelijknamige project.

Allen Bradley

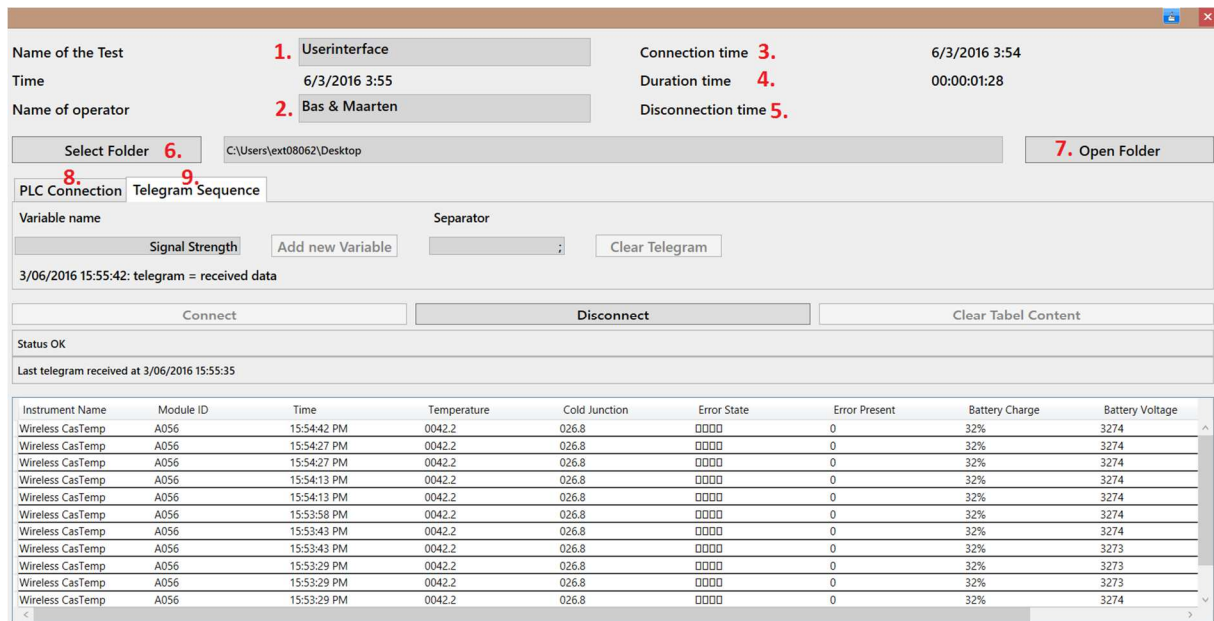
Third-party Libraries zijn ook voor Allen Bradley onnodig aangezien er gebruik gemaakt wordt van drivers uit het Open Source project AdvancedHMI.

Schneider Electric


Er is contact opgenomen met een extern bedrijf voor het schrijven van een bibliotheek die voor deze toepassing bruikbaar is. Hier is echter nog geen feedback op gekomen.

### 3.4.3 Handleiding Userinterface

Figuur 50 geeft de Userinterface weer.

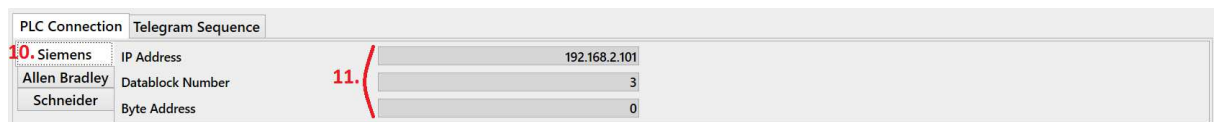


Figuur 50: De Userinterface van het C#-programma

Na het starten van het programma via shortcut “PlcTestTool”  dient de gebruiker een naam (1.) op te geven voor de test. Deze naam wordt gebruikt als naam voor het CSV- bestand, automatisch voorafgegaan door de datum en tijd. Hierna dient de gebruiker zijn of haar naam op te geven in het tekstvak “Name of operator” (2.). Vervolgens dient er een folder (6.) gekozen te worden waar het .CSV- bestand opgeslagen wordt. Deze locatie kan geopend worden met de knop “Open Folder” (7.).

De Connection time (3.) geeft de datum en tijd op het moment dat het programma gestart wordt. De Duration time (4.) houdt bij hoelang de test duurt. En de Disconnection time (5.) geeft de datum en tijd weer als het programma afgesloten wordt.

Bij de volgende stap dient de gebruikte PLC gekozen te worden in het tabblad “PLC Connections” (8.). Dit wordt weergegeven is Figuur 51.



Figuur 51: Het PLC Connection tabblad

Afhankelijk van de gekozen PLC (10.) zijn er enkele parameters (11.) vereist:

- Voor Siemens:
  - Het IP adres;
  - Datablok nummer waar de buffer zich bevindt;
  - Byte adres van de eerste plaats in de datablok waar de pc gaat uitlezen.
- Voor Allen Bradley:
  - Het IP adres;
  - Tag naam van de buffer.

Voor Schneider: IP adres,  
Startadres,  
Aantal waarden, dit kan een vaste waarde worden.

Vervolgens dient het telegram (9.) ingesteld te worden, dat wordt weergegeven is Figuur 52.

Figuur 52: Het Telegram Sequence tabblad

Het in te geven telegram moet overeenkomen met het telegram dat ingesteld is in het meetinstrument of het programma zal aangeven als het ingegeven telegram kleiner of groter is dan het telegram dat ontvangen wordt.

Bij “Variable name” (12.) wordt de naam van een variabele uit het telegram ingegeven via knop “Add” (13.). Er dient wel op gelet te worden dat de juiste volgorde van het telegram behouden blijft. Indien hier een fout tegen gemaakt wordt, kan het telegram door de gebruiker verwijderd worden met de knop “Clear Telegram” (15.). Ook dient de gebruiker altijd te specificeren welke separator (14.) deze gebruikt heeft, standaard wordt een puntkomma(;) gebruikt.

In ditzelfde tabblad wordt een melding (16.) geven i.v.m. de verhouding van het ingestelde en ontvangen telegram.

Alvorens de gebruiker op Connect (17.) kan klikken, dienen er aan enkele voorwaarden voldaan te zijn:

- naam van de test moet opgegeven zijn;
- naam van de gebruiker moet opgegeven zijn;
- er moet een opslagfolder geselecteerd zijn;
- minstens één variabele moet bij het telegram ingegeven worden;
- er moet verplicht een Separator gebruikt worden.

Indien hier aan voldaan is zal de knop “Connect” (17.) beschikbaar worden, zoals weergegeven in Figuur 53.

Figuur 53: De actie knoppen en meldingen

Met de knop “Disconnect” (18.) wordt de verbinding verbroken nadat de test voor een bepaalde tijd data gelogd heeft. Indien meteen een nieuwe test gestart dient te worden, kunnen de weergegeven waarden verwijderd worden met de knop “Clear Table Content” (19.).


In status balk 20. worden foutmeldingen weergegeven indien deze optreden. In de balk (21.) eronder wordt telkens de datum en tijd van het laatst ontvangen telegram weergegeven.

Het ingegeven telegram wordt meteen zichtbaar gemaakt aan de bovenzijde van de tabel (22.). Indien er connectie gemaakt wordt met de PLC, wordt de data (23.) onderaan het scherm weergegeven, georganiseerd volgens het ingestelde telegram met telkens de nieuwste waarde bovenaan de tabel. Dit wordt weergegeven in Figuur 54.

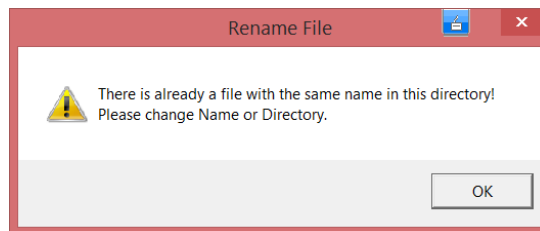
Instrument Name	22. Module ID	Time	Temperature	Cold Junction	Error State	Error Present	Battery Charge	Battery Voltage
Wireless CasTemp	A056	15:54:42 PM	0042.2	026.8	0000	0	32%	3274
Wireless CasTemp	A056	15:54:27 PM	0042.2	026.8	0000	0	32%	3274
Wireless CasTemp	A056	23. 15:54:27 PM	0042.2	026.8	0000	0	32%	3274
Wireless CasTemp	A056	15:54:13 PM	0042.2	026.8	0000	0	32%	3274
Wireless CasTemp	A056	15:54:13 PM	0042.2	026.8	0000	0	32%	3274
Wireless CasTemp	A056	15:53:58 PM	0042.2	026.8	0000	0	32%	3274
Wireless CasTemp	A056	15:53:43 PM	0042.2	026.8	0000	0	32%	3274
Wireless CasTemp	A056	15:53:43 PM	0042.2	026.8	0000	0	32%	3273
Wireless CasTemp	A056	15:53:29 PM	0042.2	026.8	0000	0	32%	3273
Wireless CasTemp	A056	15:53:29 PM	0042.2	026.8	0000	0	32%	3273
Wireless CasTemp	A056	15:53:29 PM	0042.2	026.8	0000	0	32%	3274

Figuur 54: De data weergave

### 3.4.4 Werking programma

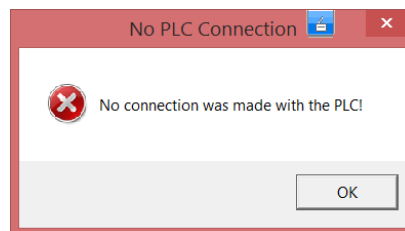
Het programma wordt gestart door te dubbel klikken op de “PlcTestTool”-shortcut . Vervolgens dient de gebruiker alle vereiste gegevens in te voeren zoals hierboven beschreven staat in hoofdstuk 3.4.3 “Handleiding User Interface”. Indien aan alle voorwaarden voldaan is, kan de gebruiker op “connect” klikken.

Eerst controleert het programma of het bestand al bestaat. Indien deze al bestaat, krijgt de gebruiker volgende meldingen. Figuur 55 geeft de melding weer als er al een bestand met dezelfde naam bestaat.



Figuur 55: Bestand bestaat al

De connectie wordt vervolgens afgebroken, zodat de gebruiker een andere naam kan ingeven of een andere locatie kan kiezen. Figuur 56 geeft de melding weer dat er geen connectie gemaakt is.



Figuur 56: Geen PLC connectie

Indien de naam verandert of als het bestand nog niet bestaat, kan het programma zonder problemen gestart worden. Hierop begint het programma eerst met te controleren welke PLC geselecteerd is. Met de ingegeven PLC parameters probeert het programma verbinding te maken. Indien dit niet lukt, geeft deze een foutmelding. Dit wordt getest in hoofdstuk 4.5.

Voor de verbinding van de pc met de **Siemens PLC** wordt gebruikt gemaakt van een bedrijfseigen platform welke op zijn beurt de “DotNetSiemensPLCToolBoxLibrary.dll” implementeert. Via het platform kan er gebruik gemaakt worden van de klasse “PlcManager” welke toegang biedt tot het gebruik van functies zoals:

- `protected bool Connect(string ipAddress, int? cpuSlot);` ,
- `protected override T ReadPlcValue<T>(int datablockNumber, int byteAddress);` ,
- `public override void Disconnect();` .

Deze functies worden in een aparte klasse “PlcDatablockReader”, welke enkel voor de connectie met de Siemens zorgt, geïmplementeerd om de connectie te verzorgen. Vanuit de userinterface wordt de waarde van het IP-adres, het databloknummer en het byte-adres meegegeven zodat het programma weet waar de data zich bevindt. De verbinding wordt gemaakt met de functie “Connect” en het Ip-adres en



de data wordt uitgelezen door het gebruik van de “Read”-functie, het databloknummer en het byte-adres. Indien de test lang genoeg gelopen heeft kan de verbinding gesloten worden met de “Disconnect” functie. Vervolgens wordt de data ingelezen als een gedefinieerde “PlcStruct”. Dit is een gereserveerde geheugenplaats met een capaciteit van 320 Bytes. Dit dient om alle data afkomstig van de PLC op te slaan. Vervolgens wordt deze data omgezet naar een byte-array voor verdere bewerking, welke dezelfde is voor de andere PLC’s. Dit wordt later besproken na de alinea van de Schneider PLC.

Voor de verbinding met de **Allen Bradley PLC** wordt gebruikt gemaakt van de drivers uit het in 3.4.2 “Gevolgde stappen” genoemde Open-Source project Advanced HMI. Door de benodigde drivers te implementeren kan er gebruik gemaakt worden van de klasse “EthernetIPforCLXCom”. Deze klasse biedt verschillende methodes die gebruikt kunnen worden voor de communicatie met de PLC. Voor dit project werd er gebruik gemaakt van volgende functies:

- `public string IPAddress { get; set; }; ,`
- `public string[] Read(string startAddress, int numberOfElements); ,`
- `public void CloseConnection(); .`

Deze functies worden in een aparte klasse “AllenBradleyConnect”, welke enkel voor de connectie met de Allen Bradley zorgt, geïmplementeerd om de connectie te verzorgen. Vanuit de userinterface wordt de waarde van het IP-adres en de Tagname meegegeven, zodat het programma weet waar de data zich bevindt. De verbinding wordt gemaakt door het inladen van het IP-adres en de data wordt uitgelezen met de “Read”-functie, het startadres(TagName) en het aantal elementen die uitgelezen dienen te worden. Indien de test lang genoeg gelopen heeft, kan de verbinding gesloten worden met de “Disconnect” functie. Vervolgens wordt de data rechtstreeks omgezet naar een Byte-array voor verdere bewerking, welke dezelfde is voor de andere PLC’s. Dit wordt later besproken na de alinea van de Schneider PLC.

Voor de verbinding met de **Schneider PLC** wordt gebruikt gemaakt van de nModbus driver, besproken in 3.4.2 “Gevolgde stappen”. Door de benodigde driver te implementeren kan er gebruik gemaakt worden van de klassen “IPAddress”, “TcpClient” en “ModbusIpMaster”. Deze klasse biedt verschillende methodes die gebruikt kunnen worden voor de communicatie met de PLC. In dit project werd er geprobeerd met volgende functies:

```
public IPAddress(byte[] address); ,  
public TcpClient(string hostname, int port); ,  
public static ModbusIpMaster CreateIp(TcpClient tcpClient); .
```

Dit is niet verder uitgewerkt omdat er connectie met de PLC gemaakt kan worden, maar nog geen data uitgelezen kan worden.

Vanaf hier volgt de data van elke PLC dezelfde weg. De data wordt meegegeven met de “Parse”-functie naar de klasse “TelegramStructure” waar de data ontdaan wordt van alle lege plaatsen en geconverteerd wordt naar een ASCII formaat. Vervolgens wordt er gekeken of het voorgaande telegram leeg is of dat het telegram al geupdate is. Hierna wordt het gesplitst in een String array op basis van de opgegeven separator, welke als default waarde “;” heeft. Hierna wordt de data toegevoegd aan een record op basis van het ingegeven telegram. Indien deze niet overeenkomen wordt er een melding gegeven die wordt duidelijk gemaakt in 4.4 “C# programma”.

Figuur 57 geeft de melding weer indien het ingesteld telegram even groot is al het ontvangen telegram.

30/05/2016 11:11:18: telegram = received data

Figuur 57: Correct telegram

Van het moment dat het telegram op het scherm wordt getoont, wordt de data ook automatisch gelogd naar een .CSV-bestand. Figuur 58 toont een .CSV bestand geopend in Excel.

	A	B	C	D	E	F	G	H	I	J
1	<b>ErrorTest</b>	<b>executed by:</b>	<b>Bas</b>	<b>on</b>	<b>30/05/2016 15:28</b>					
2	<b>Instrument Name</b>	<b>Module ID</b>	<b>Time</b>	<b>Temperature</b>	<b>Cold Junction</b>	<b>Error State</b>	<b>Error Present</b>	<b>Battery Charge</b>	<b>Battery Voltage</b>	<b>Signal Strength</b>
3	Wireless CasTemp	A056	16:31:02 PM	42.2	26.7		0	33%	3281	86
4	Wireless CasTemp	A056	16:31:02 PM	42.2	26.7		0	33%	3281	86
5	Wireless CasTemp	A056	16:31:17 PM	42.2	26.7		0	33%	3281	86
6	Wireless CasTemp	A056	16:31:31 PM	42.2	26.7		0	33%	3281	86
7	Wireless CasTemp	A056	16:31:31 PM	42.2	26.7		0	33%	3281	85
8	Wireless CasTemp	A056	16:31:46 PM	42.2	26.7		0	33%	3281	85
9	Wireless CasTemp	A056	16:31:46 PM	42.2	26.7		0	33%	3281	81
10	Wireless CasTemp	A056	16:32:01 PM	42.2	26.7		0	33%	3281	81
11	Wireless CasTemp	A056	16:32:01 PM	42.2	26.7		0	33%	3281	87
12	Wireless CasTemp	A056	16:32:01 PM	42.2	26.7		0	33%	3281	88

Figuur 58: .CSV bestand

In het .CSV-bestand wordt als eerste de titel, gebruikersnaam en datum/tijd geschreven. Dit wordt gevolgd door het ingegeven telegram. Waaronder dan de data geordend wordt.

## 4 Testen en resultaten

Er zijn verschillende scenario's getest voor zowel de PLC's als het C# programma. Hiervoor is er gebruik gemaakt van verschillende sensor-simulatiemodules, welke verbonden worden met de meetinstrumenten. Deze instrumenten worden op hun beurt verbonden met de PLC, die vervolgens verbonden zijn met het C#-programma.

Om korte en snelle testen uit te voeren wordt er gebruik gemaakt van de Quick-Check modules. Deze worden enkel gebruikt om te controleren als er data binnenkomt. Figuur 59 geeft een voorbeeld van de Quick Check modules.



Figuur 59: Quick Check modules

Voor langdurige testen wordt gebruik gemaakt van een Checkmate IV, deze stuurt verschillende, in te stellen, waarden uit. Figuur 60 toont een Checkmate IV.



Figuur 60: Checkmate IV [37]

Voor de CasTemp Wireless is er een aparte sensor module voorzien; de CasTemp Wireless A056. Deze biedt continu waarden aan, welke door het CasTemp Wireless meetinstrument uitgelezen wordt. Figuur 61 geeft het CasTemp Wireless instrument en de sensormodule weer.



*Figuur 61: CasTemp Wireless instrument met sensormodule [38]*

Hierna volgen enkele uitgevoerde testen. Eerst zullen de PLC testen besproken worden in de volgorde dat het project verlopen is, namelijk Siemens, Allen Bradley en Schneider Electric. Vervolgens komen de testen met het C#-programma aan bod.

## 4.1 Siemens

De Siemens PLC kan communiceren via twee protocollen, Profibus en Profinet. Beiden zijn uitvoerig getest met alle beschikbare meetinstrumenten.

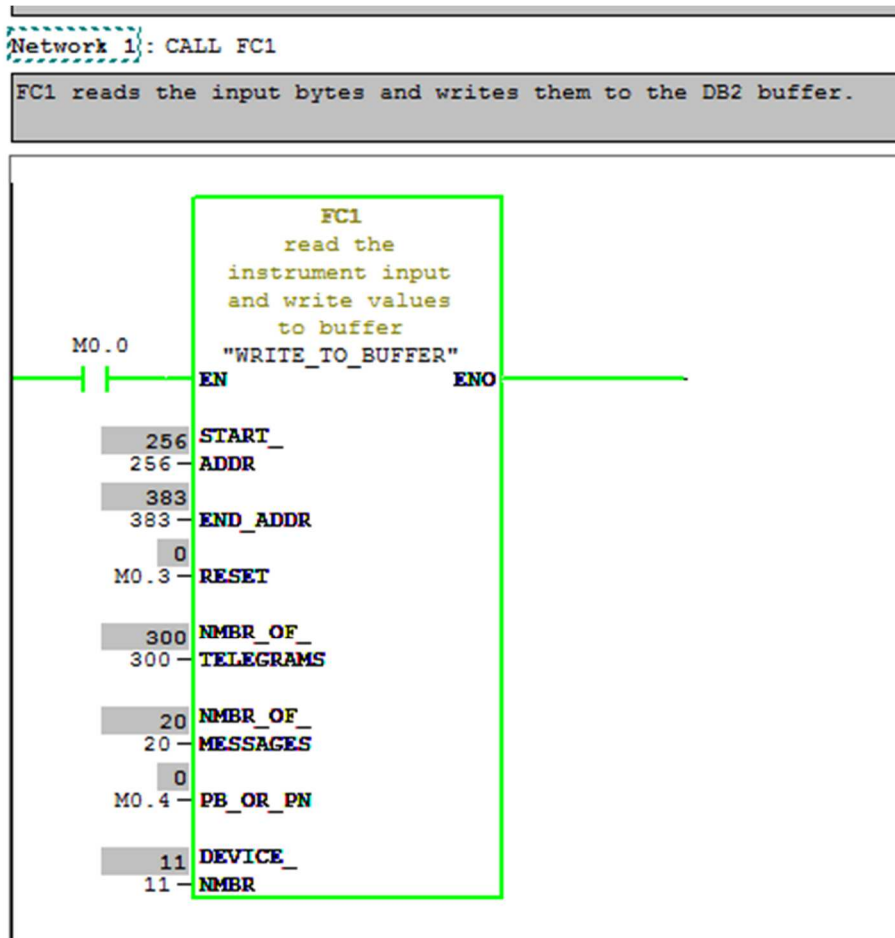
### Profibus

Onderstaande test maakt gebruik van een Sensor Lab als meetinstrument. De Sensor Lab is aangesloten op de Quick-Check sensor-simulatiemodule. Het meetinstrument is voor de test toegevoegd aan de het Profibus netwerk met adres 11. Dit is weergegeven in Figuur 62.

Slot	DP ID	Order Number / Designation	I Address	Q Address	Comment
1	64	64 WORD Slave-Out/Master-In	256...383		
2	128	64 WORD Slave-In/Master-Out		256...383	
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Figuur 62: Hardware configuratie Siemens met Sensor Lab

Hierna worden de instellingen voor de communicatie ingevoerd in OB1 voor het programma zoals in Figuur 63.



Figuur 63: FC1 interface met instellingen voor Sensor Lab en Profibus

- start- en stopadres zijn terug te vinden in de hardware configuratie;
- reset is enkel nodig om het programma te herstarten;
- aantal telegrammen en aantal boodschappen zijn afhankelijk van de buffer;
- de diagnose is ingesteld op Profibus en controleert toestel 11.

De gebufferde data welke naar de pc verzonden wordt is terug te vinden in de pconnect DB. Figuur 64 geeft de waarde weer die tijdens de test naar de pc werd verzonden.

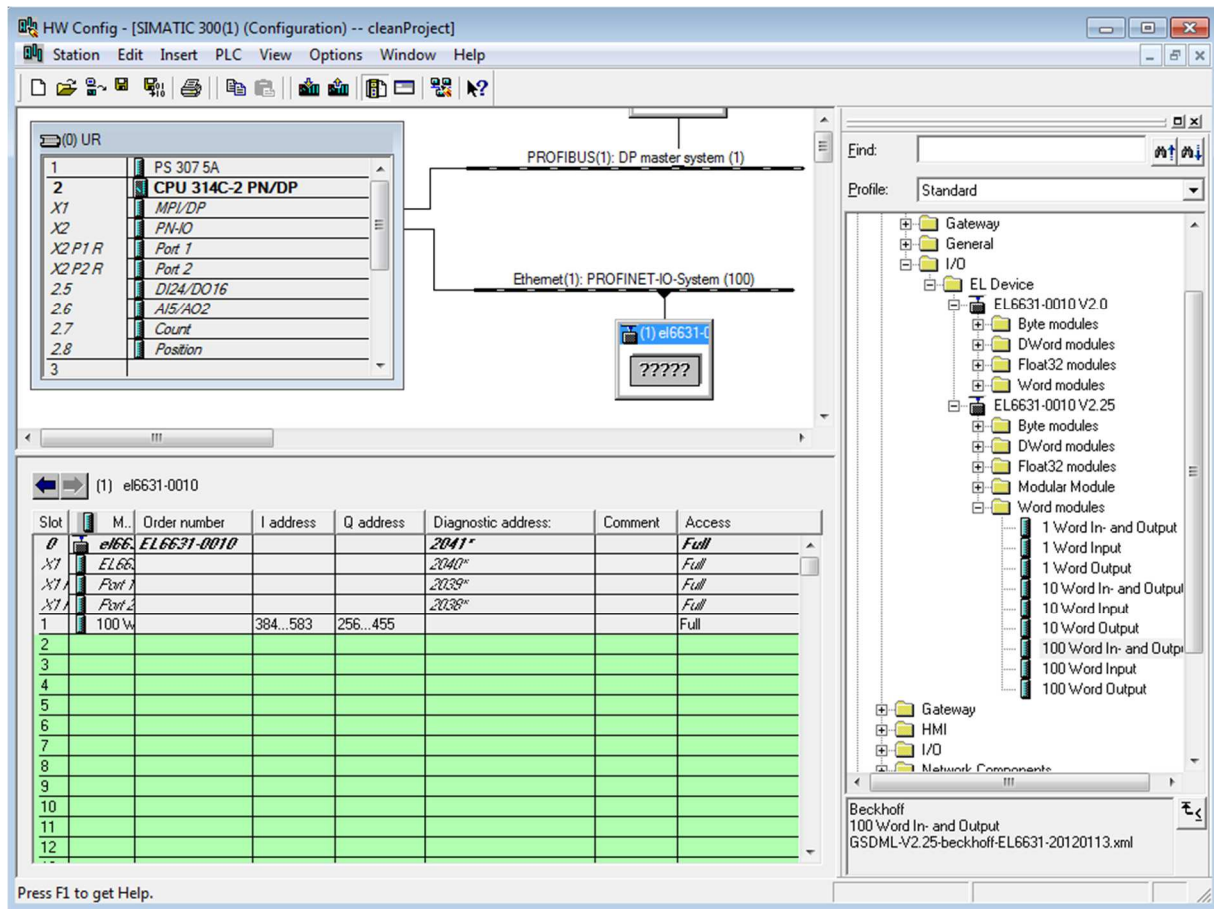
Address	Name	Type	Initial value	Actual value	Comment
0.0	telegram[0]	CHAR	' '	'1'	Temporary place!
1.0	telegram[1]	CHAR	' '	'6'	
2.0	telegram[2]	CHAR	' '	'5'	
3.0	telegram[3]	CHAR	' '	'7'	
4.0	telegram[4]	CHAR	' '	'.'	
5.0	telegram[5]	CHAR	' '	'7'	
6.0	telegram[6]	CHAR	' '	','	
7.0	telegram[7]	CHAR	' '	'0'	
8.0	telegram[8]	CHAR	' '	'0'	
9.0	telegram[9]	CHAR	' '	'0'	
10.0	telegram[10]	CHAR	' '	'0'	
11.0	telegram[11]	CHAR	' '	'.'	
12.0	telegram[12]	CHAR	' '	'0'	
13.0	telegram[13]	CHAR	' '	','	
14.0	telegram[14]	CHAR	' '	'0'	
15.0	telegram[15]	CHAR	' '	'0'	
16.0	telegram[16]	CHAR	' '	'0'	
17.0	telegram[17]	CHAR	' '	'0'	
18.0	telegram[18]	CHAR	' '	'0'	
19.0	telegram[19]	CHAR	' '	','	
20.0	telegram[20]	CHAR	' '	'0'	
21.0	telegram[21]	CHAR	' '	'0'	
22.0	telegram[22]	CHAR	' '	'0'	
23.0	telegram[23]	CHAR	' '	'0'	
24.0	telegram[24]	CHAR	' '	'0'	
25.0	telegram[25]	CHAR	' '	','	
26.0	telegram[26]	CHAR	' '	'0'	
27.0	telegram[27]	CHAR	' '	'0'	
28.0	telegram[28]	CHAR	' '	'0'	
29.0	telegram[29]	CHAR	' '	'0'	
30.0	telegram[30]	CHAR	' '	'0'	
31.0	telegram[31]	CHAR	' '	','	
32.0	telegram[32]	CHAR	' '	'0'	
33.0	telegram[33]	CHAR	' '	'1'	
34.0	telegram[34]	CHAR	' '		

Figuur 64: Pconnect DB tijdens de Sensor Lab test over Profibus

Tijdens de test werd er gebruik gemaakt van een Quick-Check welke enkel een temperatuur waarde kan simuleren.

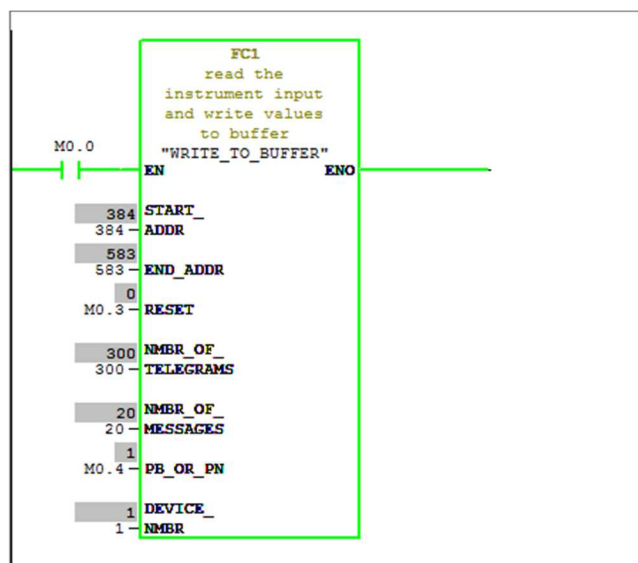
## Profinet

Voor de test met Profinet werd er een soortgelijke opstelling gebruikt als bij de Profibus test. Figuur 65 geeft de hardware configuratie weer voor de Profinet test met de Sensor Lab.



Figuur 65: Hardware configuratie Sensor Lab met Profinet

Vervolgens werd de communicatie ingesteld in OB1 zoals in Figuur 66.



Figuur 66: FCI interface met instellingen voor Sensor Lab en Profinet



In vergelijking met de Profibus test zijn de start- en stopadressen gewijzigd en zal de diagnosefunctie kijken naar toestel 1 op het Profinet netwerk.

Van zodra het programma is ingeladen en gestart is zullen alle waarden die naar de pc gestuurd worden opgeslagen zijn in de pconnect DB. Onderstaande Figuur 67 geeft de pc-connect DB tijdens de test.

Address	Name	Type	Initial value	Actual value	Comment
0.0	telegram[0]	CHAR	' '	'1'	Temporary place!
1.0	telegram[1]	CHAR	' '	'4'	
2.0	telegram[2]	CHAR	' '	'9'	
3.0	telegram[3]	CHAR	' '	'8'	
4.0	telegram[4]	CHAR	' '	' '	
5.0	telegram[5]	CHAR	' '	'8'	
6.0	telegram[6]	CHAR	' '	' '	
7.0	telegram[7]	CHAR	' '	'0'	
8.0	telegram[8]	CHAR	' '	'0'	
9.0	telegram[9]	CHAR	' '	'0'	
10.0	telegram[10]	CHAR	' '	'0'	
11.0	telegram[11]	CHAR	' '	' '	
12.0	telegram[12]	CHAR	' '	'0'	
13.0	telegram[13]	CHAR	' '	' '	
14.0	telegram[14]	CHAR	' '	'0'	
15.0	telegram[15]	CHAR	' '	'0'	
16.0	telegram[16]	CHAR	' '	'0'	
17.0	telegram[17]	CHAR	' '	'0'	
18.0	telegram[18]	CHAR	' '	'0'	
19.0	telegram[19]	CHAR	' '	' '	
20.0	telegram[20]	CHAR	' '	'0'	
21.0	telegram[21]	CHAR	' '	'0'	
22.0	telegram[22]	CHAR	' '	'0'	
23.0	telegram[23]	CHAR	' '	'0'	
24.0	telegram[24]	CHAR	' '	'0'	
25.0	telegram[25]	CHAR	' '	' '	
26.0	telegram[26]	CHAR	' '	'0'	
27.0	telegram[27]	CHAR	' '	'0'	
28.0	telegram[28]	CHAR	' '	'0'	
29.0	telegram[29]	CHAR	' '	'0'	
30.0	telegram[30]	CHAR	' '	'0'	
31.0	telegram[31]	CHAR	' '	' '	
32.0	telegram[32]	CHAR	' '	'0'	
33.0	telegram[33]	CHAR	' '	'1'	
34.0	telegram[34]	CHAR	' '	' '	

Figuur 67: Pconnect DB tijdens de test.

Ook tijdens de Profinet test is er gebruik gemaakt van een Quick-Check wat enkel een temperatuur simuleert.

## Connectie controle

De PLC is in staat om de connectie met het meetinstrument te controleren op fouten, ook dit is getest door tijdens de connectie de verbinding tussen PLC en meetinstrument te verbreken. Dit wordt bijgehouden in de Error Buffer. Onderstaande Figuur 68 geeft de error buffer weer wanneer de verbinding tussen PLC en meetinstrument functioneel is.

Address	Name	Type	Initial value	Actual value	Comment
0.0	OB1_PREV_CYCLE	WORD	W#16#0	W#16#0000	cycle time of th
2.0	OB1_MAX_CYCLE	WORD	W#16#0	W#16#0008	maximum cycle t:
4.0	OB1_MIN_CYCLE	WORD	W#16#0	W#16#0000	minimum cycle t:
6.0	PROFIBUS_BF	BOOL	FALSE	FALSE	busfault at pro:
6.1	PROFINET_BF	BOOL	FALSE	FALSE	busfault at pro:
6.2	COMM_OK	BOOL	FALSE	TRUE	OK
6.3	COMM_MAINTENANCE	BOOL	FALSE	FALSE	maintenance
6.4	COMM_FAULTY	BOOL	FALSE	FALSE	faulty
6.5	COMM_FAILED	BOOL	FALSE	FALSE	failed
6.6	COMM_WAS_DEACTIVATED	BOOL	FALSE	FALSE	was deactivated
6.7	COMM_DPV1	BOOL	FALSE	FALSE	DPV1
7.0	COMM_WAS_MAINTENANCE	BOOL	FALSE	FALSE	was under maite:
7.1	COMM_WAS_FAULTY	BOOL	FALSE	FALSE	was faulty
7.2	COMM_HAD_FAILED	BOOL	FALSE	TRUE	had failed
7.3	COMM_DEACTIVATED	BOOL	FALSE	FALSE	deactivated
7.4	COMM_RESERVED	BOOL	FALSE	FALSE	reserved
7.5	COMM_RESERVED1	BOOL	FALSE	FALSE	reserved
7.6	COMM_RESERVED2	BOOL	FALSE	FALSE	reserved
7.7	COMM_RESERVED3	BOOL	FALSE	FALSE	reserved
8.0	COMM_RESERVED4	BOOL	FALSE	FALSE	reserved
10.0	READ_CLK_ERROR	WORD	W#16#0	W#16#0000	error code if re
12.0	YEAR	WORD	W#16#0	W#16#0016	last 2 digits of
14.0	MONTH1	WORD	W#16#0	W#16#0006	month of the PLC
16.0	DAY	WORD	W#16#0	W#16#0003	day of the PLC
18.0	HOOR	WORD	W#16#0	W#16#0011	hour of the PLC
20.0	MINUTE	WORD	W#16#0	W#16#0022	minute of the P
22.0	SECOND	WORD	W#16#0	W#16#0000	second of the P
24.0	MILISECOND	WORD	W#16#0	W#16#0057	milisecond of th
26.0	WEEKDAY	WORD	W#16#0	W#16#0006	weekday of the 1

Figuur 68: Error buffer tijdens een goede connectie tussen PLC en meetinstrument.

De buffer onthoudt:

- cyclustijden,
- connectiestatus,
- PLC datum en tijd.

Wanneer de verbinding tussen PLC en meetinstrument in orde is, wordt dit weergegeven door de waarde "COMM\_OK".

Als nu de verbinding verbroken wordt zal de “COMM\_OK” waarde op “FALSE” komen te staan en zal de “COMM\_FAILED” waarde op “TRUE” staan zoals te zien is in Figuur 69.

Address	Name	Type	Initial value	Actual value	Comment
0.0	OB1_PREV_CYCLE	WORD	W#16#0	W#16#0001	cycle time of t:
2.0	OB1_MAX_CYCLE	WORD	W#16#0	W#16#0008	maximum cycle t:
4.0	OB1_MIN_CYCLE	WORD	W#16#0	W#16#0000	minimum cycle t:
6.0	PROFIBUS_BF	BOOL	FALSE	FALSE	busfault at pro:
6.1	PROFINET_BF	BOOL	FALSE	FALSE	busfault at pro:
6.2	COMM_OK	BOOL	FALSE	FALSE	OK
6.3	COMM_MAINTENANCE	BOOL	FALSE	FALSE	maintenance
6.4	COMM_FAULTY	BOOL	FALSE	FALSE	faulty
6.5	COMM_FAILED	BOOL	FALSE	TRUE	failed
6.6	COMM_WAS_DEACTIVATED	BOOL	FALSE	FALSE	was deactivated
6.7	COMM_DPV1	BOOL	FALSE	FALSE	DPV1
7.0	COMM_WAS_MAINTENANCE	BOOL	FALSE	FALSE	was under maite:
7.1	COMM_WAS_FAULTY	BOOL	FALSE	FALSE	was faulty
7.2	COMM_HAD_FAILED	BOOL	FALSE	TRUE	had failed
7.3	COMM_DEACTIVATED	BOOL	FALSE	FALSE	deactivated
7.4	COMM_RESERVED	BOOL	FALSE	FALSE	reserved
7.5	COMM_RESERVED1	BOOL	FALSE	FALSE	reserved
7.6	COMM_RESERVED2	BOOL	FALSE	FALSE	reserved
7.7	COMM_RESERVED3	BOOL	FALSE	FALSE	reserved
8.0	COMM_RESERVED4	BOOL	FALSE	FALSE	reserved
10.0	READ_CLK_ERROR	WORD	W#16#0	W#16#0000	error code if r:
12.0	YEAR	WORD	W#16#0	W#16#0016	last 2 digits o:
14.0	MONTH1	WORD	W#16#0	W#16#0006	month of the PLC
16.0	DAY	WORD	W#16#0	W#16#0003	day of the PLC
18.0	HOUR	WORD	W#16#0	W#16#0011	hour of the PLC
20.0	MINUTE	WORD	W#16#0	W#16#0023	minute of the P
22.0	SECOND	WORD	W#16#0	W#16#0056	second of the P
24.0	MILLISECOND	WORD	W#16#0	W#16#0009	milisecond of t:
26.0	WEEKDAY	WORD	W#16#0	W#16#0006	weekday of the l

Figuur 69: Error buffer tijdens onderbroken connectie

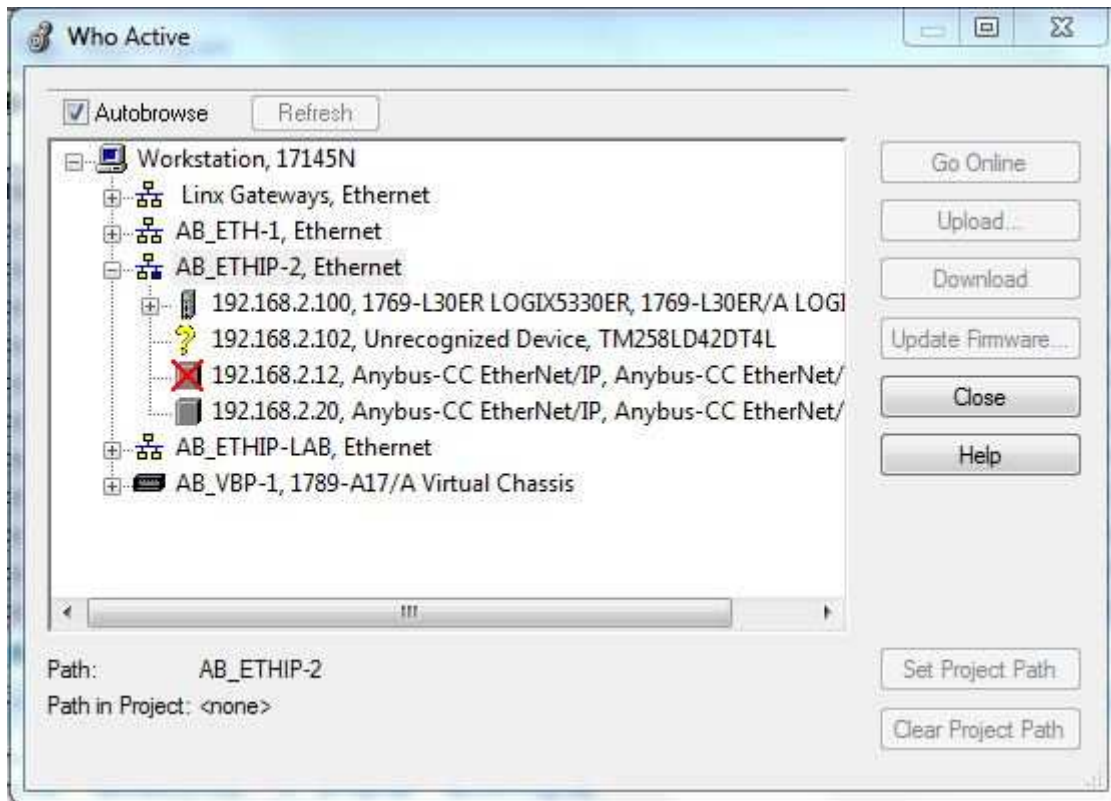
De waarde “COMM\_HAS\_FAILED” blijft op “TRUE” staan om aan te geven dat er een fout is geweest.

### Conclusie

Na deze testen is het duidelijk dat het programma voor de Siemens PLC in staat is om de data van het meetinstrument in te lezen, te bufferen en klaar te zetten om door de pc te gebruiken. Het programma kan eveneens ook de verbinding met het meetinstrument te controleren.

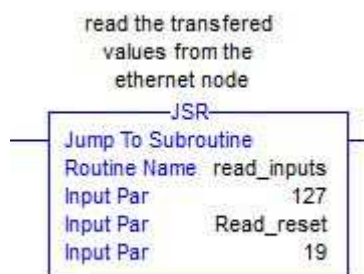
## 4.2 Allen Bradley

De Allen-Bradley communiceert via EtherNet IP, dit is onder andere getest met behulp van een CasTemp Wireless meetinstrument. Het meetinstrument wordt bijgevoegd in het Ethernet netwerk. Figuur 70 geeft de “Who active” functie weer tijdens de Ethernet IP test.



Figuur 70: Who Active functie tijdens Ethernet IP test

De instellingen voor de communicatie gebeuren via het hoofdprogramma (MainRoutine), via de JSR functie voor “read\_inputs” programma. Figuur 71 geeft de instellingen voor de communicatie met de CasTemp Wireless via Ethernet IP.



Figuur 71: Jump SubRoutine functie voor het read\_inputs programma.

De ingestelde waarden zijn in volgorde: aantal telegrammen, reset en het aantal boodschappen in de buffer tellende vanaf 0.

Wanneer het programma wordt uitgevoerd zal de data van het meetinstrument eerst gebufferd worden en vervolgens klaargezet worden voor de pc.

Figuur 72 geeft de waarden weer tijdens de test van de CasTemp Wireless.

Name	Value	Force Mask	Style	Data Type	Description	Constant
pc	{...}	{...}		messages	previous messag...	<input type="checkbox"/>
pc.telegram	{...}	{...}	ASCII	SINT[300]	previous messag...	
pc.telegram[0]	'N'		ASCII	SINT	previous messag...	
pc.telegram[1]	'i'		ASCII	SINT	previous messag...	
pc.telegram[2]	'z'		ASCII	SINT	previous messag...	
pc.telegram[3]	'e'		ASCII	SINT	previous messag...	
pc.telegram[4]	'1'		ASCII	SINT	previous messag...	
pc.telegram[5]	'e'		ASCII	SINT	previous messag...	
pc.telegram[6]	'a'		ASCII	SINT	previous messag...	
pc.telegram[7]	'a'		ASCII	SINT	previous messag...	
pc.telegram[8]	' '		ASCII	SINT	previous messag...	
pc.telegram[9]	'C'		ASCII	SINT	previous messag...	
pc.telegram[10]	'a'		ASCII	SINT	previous messag...	
pc.telegram[11]	'a'		ASCII	SINT	previous messag...	
pc.telegram[12]	'T'		ASCII	SINT	previous messag...	
pc.telegram[13]	'e'		ASCII	SINT	previous messag...	
pc.telegram[14]	'm'		ASCII	SINT	previous messag...	
pc.telegram[15]	'p'		ASCII	SINT	previous messag...	
pc.telegram[16]	','		ASCII	SINT	previous messag...	
pc.telegram[17]	'A'		ASCII	SINT	previous messag...	
pc.telegram[18]	'0'		ASCII	SINT	previous messag...	
pc.telegram[19]	'5'		ASCII	SINT	previous messag...	
pc.telegram[20]	'6'		ASCII	SINT	previous messag...	
pc.telegram[21]	','		ASCII	SINT	previous messag...	
pc.telegram[22]	'1'		ASCII	SINT	previous messag...	

Figuur 72: Pc.telegram tag tijdens de CasTemp Wireless test

Deze waarden zijn reeds door het programma gebufferd en staan klaar om door de pc applicatie gebruikt te worden.

### Connectie controle

Voor de Allen-Bradley zijn er niet veel manieren om de status van de verbinding tussen de pc en de PLC te controleren. In de controller tags van de toestellen is meestal een extra boolean voorzien om weer te geven of het meetinstrument is verbonden of niet. Onderstaande Figuur 73 geeft deze controle weer voor de CasTemp Wireless.

Name	Value	Force Mask	Style	Data Type	Description	Constant
anybus:1	{...}	{...}		_005A:ABCC_81EAGAAD:1:0		<input type="checkbox"/>
anybus:1.ConnectionFaulted	0		Decimal	BOOL		
anybus:1.Data	{...}	{...}	ASCII	SINT[128]		

Figuur 73: Controle voor verbinding tussen Allen-Bradley en meetinstrument.

### Conclusie

Na het testen van het Allen-Bradley programma met de CasTemp Wireless en andere meetinstrumenten is de conclusie dat het programma in staat is om data van de meetinstrumenten te ontvangen, te bufferen en deze klaar te zetten voor de pc. voor de connectie controle kan er enkel gekeken worden of het instrument nog verbonden is of niet.

### 4.3 Schneider Electric

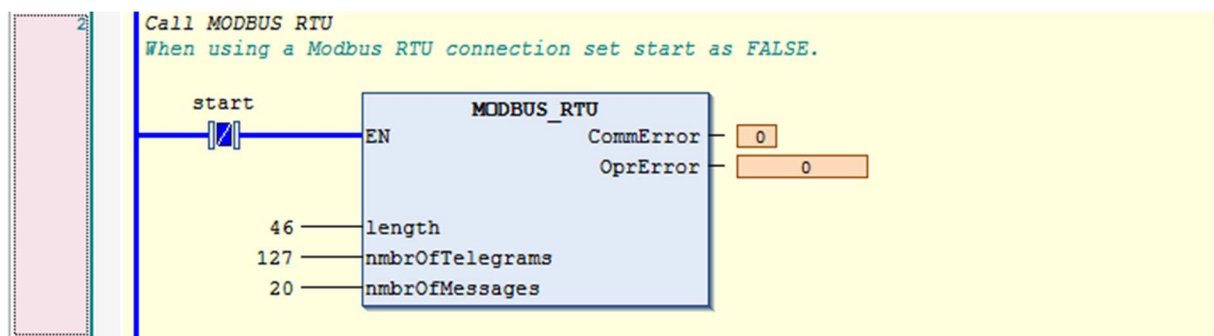
Voor communicatie met de meetinstrumenten kan de Schneider PLC twee protocollen kiezen, Modbus RTU voor een seriële verbinding en Modbus TCP voor een soortgelijke Ethernet verbinding. Dit is onder andere getest met de DigiTemp-E Wireless als meetinstrument. De DigiTemp-E Wireless is aangesloten op een Checkmate IV simulatiemodule om zo verschillende temperaturen te simuleren.

#### Modbus RTU

De gevolgdde stappen om de Schneider te verbinden met het meetinstrument via Modbus RTU staan beschreven in 3.3.1 Configuratie en worden hier niet verder besproken.

De instellingen voor het programma worden ingegeven in het hoofdprogramma (POU).

Figuur 74 geeft de instellingen weer voor de Modbus RTU communicatie met de DigiTemp-E Wireless.



Figuur 74: Instellingen voor Modbus RTU programma

De ingestelde waarden zijn:

- de lengte van de holding registers;
- het aantal telegrammen;
- het aantal boodschappen in de buffer.

De ontvangen data wordt eerst door de I/O scanner opgeslagen in WORD formaat.

Figuur 75 geeft de ontvangen data weer zoals de I/O scanner het inleest.

Variable	Mapping	Channel	Address	Type	Default Value	Current Value	Prepared Value	Unit	Description
* Application.GVL.gener...		Channel 1	%HW32	WORD		18756			READ 16#0100 (=00256)
* Application.GVL.gener...		Channel 1	%HW33	WORD		18759			READ 16#0101 (=00257)
* Application.GVL.gener...		Channel 1	%HW34	WORD		17748			READ 16#0102 (=00258)
* Application.GVL.gener...		Channel 1	%HW35	WORD		20557			READ 16#0103 (=00259)
* Application.GVL.gener...		Channel 1	%HW36	WORD		17709			READ 16#0104 (=00260)
* Application.GVL.gener...		Channel 1	%HW37	WORD		12603			READ 16#0105 (=00261)
* Application.GVL.gener...		Channel 1	%HW38	WORD		13875			READ 16#0106 (=00262)
* Application.GVL.gener...		Channel 1	%HW39	WORD		11829			READ 16#0107 (=00263)
* Application.GVL.gener...		Channel 1	%HW40	WORD		15160			READ 16#0108 (=00264)
* Application.GVL.gener...		Channel 1	%HW41	WORD		12336			READ 16#0109 (=00265)
* Application.GVL.gener...		Channel 1	%HW42	WORD		14386			READ 16#010A (=00266)
* Application.GVL.gener...		Channel 1	%HW43	WORD		12350			READ 16#010B (=00267)
* Application.GVL.gener...		Channel 1	%HW44	WORD		12347			READ 16#010C (=00268)
* Application.GVL.gener...		Channel 1	%HW45	WORD		12336			READ 16#010D (=00269)
* Application.GVL.gener...		Channel 1	%HW46	WORD		15153			READ 16#010E (=00270)
* Application.GVL.gener...		Channel 1	%HW47	WORD		13104			READ 16#010F (=00271)
* Application.GVL.gener...		Channel 1	%HW48	WORD		12335			READ 16#0110 (=00272)
* Application.GVL.gener...		Channel 1	%HW49	WORD		12086			READ 16#0111 (=00273)
* Application.GVL.gener...		Channel 1	%HW50	WORD		13873			READ 16#0112 (=00274)
* Application.GVL.gener...		Channel 1	%HW51	WORD		12603			READ 16#0113 (=00275)
* Application.GVL.gener...		Channel 1	%HW52	WORD		14899			READ 16#0114 (=00276)
* Application.GVL.gener...		Channel 1	%HW53	WORD		13360			READ 16#0115 (=00277)
* Application.GVL.gener...		Channel 1	%HW54	WORD		13626			READ 16#0116 (=00278)

Figuur 75: I/O mapping van de I/O scanner

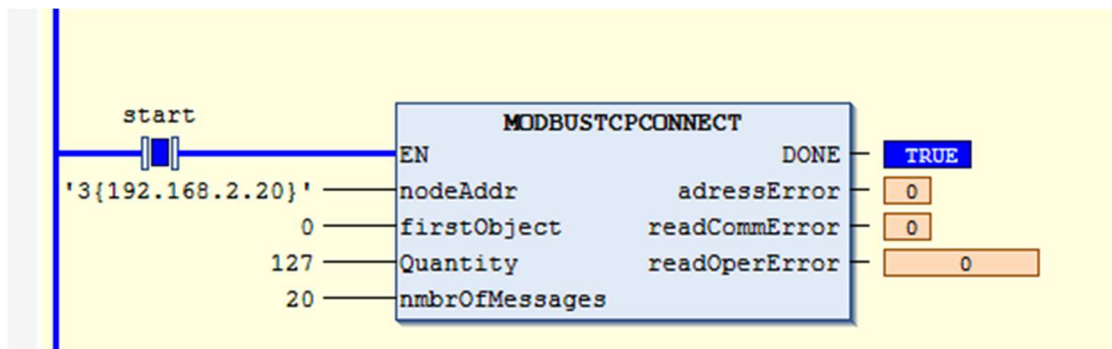
Het programma zal hierna alle Word waarden omzetten naar 2 byte waarden. Deze waarden worden opgeslagen in de globale variabele GVL.buffer. Figuur 76 geeft de waarden in de buffer weer, de SoMachine kan deze waarden niet in ASCII formaat weergeven enkel de decimale waarden.

Expression	Type	Value	Prepared value	Comment
buffer	ARRAY [0..20] OF m...			
buffer[0]	message			
telegrams	ARRAY [0..300] OF...			
telegrams[0]	BYTE	0		
telegrams[1]	BYTE	73		
telegrams[2]	BYTE	71		
telegrams[3]	BYTE	73		
telegrams[4]	BYTE	84		
telegrams[5]	BYTE	69		
telegrams[6]	BYTE	77		
telegrams[7]	BYTE	80		
telegrams[8]	BYTE	45		
telegrams[9]	BYTE	69		
telegrams[10]	BYTE	59		
telegrams[11]	BYTE	49		
telegrams[12]	BYTE	52		
telegrams[13]	BYTE	57		
telegrams[14]	BYTE	50		
telegrams[15]	BYTE	46		
telegrams[16]	BYTE	53		
telegrams[17]	BYTE	59		
telegrams[18]	BYTE	48		
telegrams[19]	BYTE	48		
telegrams[20]	BYTE	50		
telegrams[21]	BYTE	56		
telegrams[22]	BYTE	46		
telegrams[23]	BYTE	50		
telegrams[24]	BYTE	59		
telegrams[25]	BYTE	48		
telegrams[26]	BYTE	48		
telegrams[27]	BYTE	48		
telegrams[28]	BYTE	49		
telegrams[29]	BYTE	59		
telegrams[30]	BYTE	48		

Figuur 76: Buffer in de globale variabele lijst

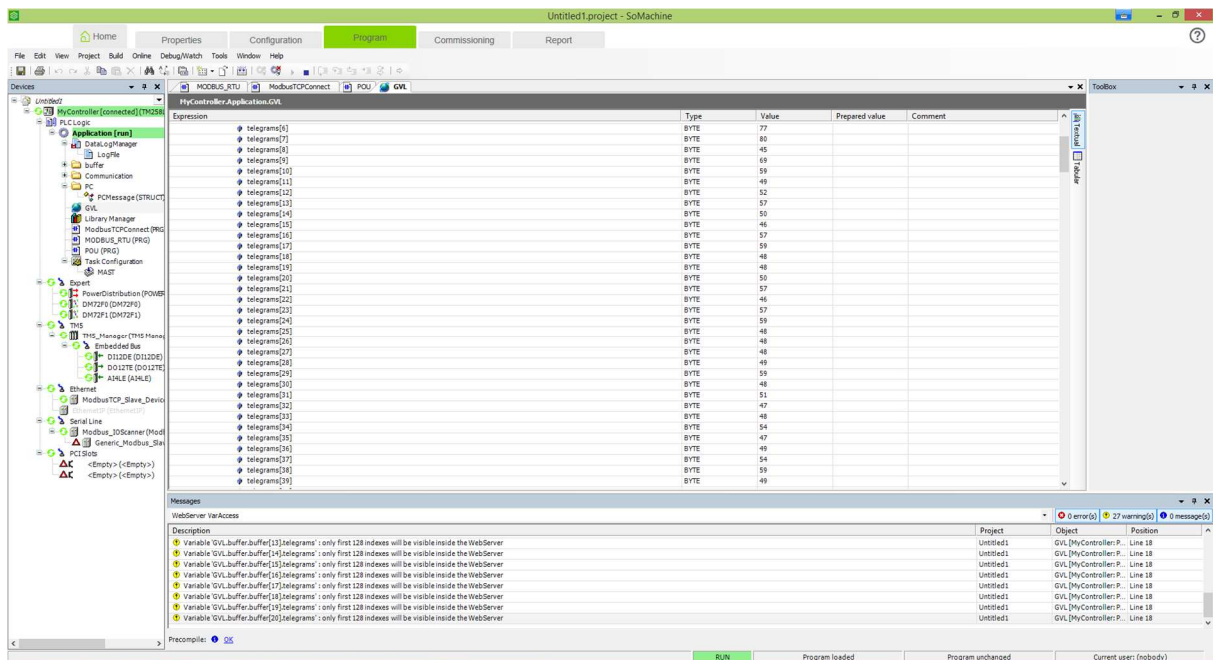
## Modbus TCP

Voor Modbus TCP gebeurt de configuratie en de instelling voor de communicatie rechtstreeks in het hoofdprogramma (POU). Figuur 77 geeft de instellingen weer voor het Modbus TCP programma.



Figuur 77: Interface voor Modbus TCP programma

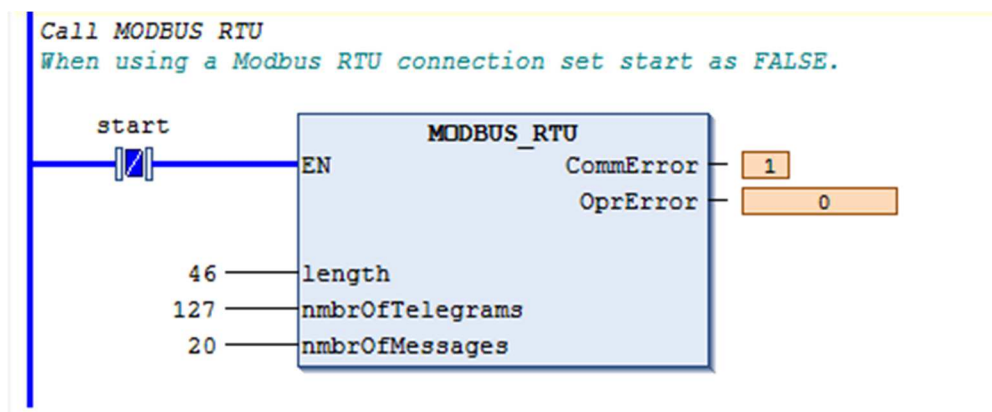
Het programma zal de gegevens via de Modbus TCP connectie inlezen en bijhouden in de globale buffer. Figuur 78 toont de globale buffer tijdens de test met een DigiTemp-E Wireless als meetinstrument.



Figuur 78: Globale buffer tijdens de test via Modbus TCP

### Connectie controle

De status van de verbinding wordt gecontroleerd door de errorcodes van de leesfunctie. Deze codes staan beschreven in 3.3.3 Connectie controle. Onderstaande Figuur 79 geeft de interface weer tijdens het verbreken van de verbinding tussen PLC en meetinstrument.



Figuur 79: RTU interface tijdens verbroken verbinding

De weergegeven error code is de communicatie error voor “timed out”, wat aangeeft dat er binnen de opgegeven tijd geen verbinding werd gevormd.



## 4.4 C# programma

Voor het testen van het programma zijn volgende problemen gesimuleerd:

1. verkeerde PLC instellingen;
2. ingeven van een verkeerd telegram;
3. onderbreken van de communicatie tussen sensor en meettoestel;
4. uitschakelen van het meettoestel;
5. onderbreken van de communicatie tussen meetinstrument en PLC;
6. uitschakelen van de PLC;
7. onderbreken van de communicatie tussen PLC en pc;
8. Duurtest met Siemens PLC en Castemp;
9. Duurtest met Allen Bradley en CasTemp;
10. E-Line test met Allen Bradley;
11. SensorLab test met Siemens.

De resultaten van bovenstaande problemen worden hieronder opgesomd.

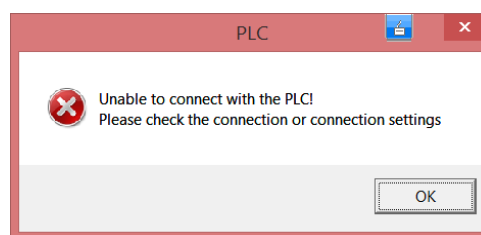
### 1. Verkeerde PLC instellingen.

Hierop begint het programma eerst met te controleren welke PLC geselecteerd is. In Figuur 80 is de Siemens PLC geselecteerd en de parameters zijn ingevuld.

PLC Connection	Telegram Sequence
Siemens	IP Address 192.168.2.101
Allen Bradley	Datablock Number 3
Schneider	Byte Address 0

Figuur 80: PLC connection tabblad

Met de ingegeven PLC parameters probeert het programma verbinding te maken. Indien dit niet lukt, geeft deze de foutmelding weergegeven in Figuur 81.



Figuur 81: Geen connectie mogelijk

De foutmelding kan verschillende oorzaken hebben, zoals de netwerkkabel die niet is aangesloten of een foutief IP-adres van de PLC.

Bij een fout tegen het databloknummer of het byte-adres krijgt het programma geen data binnen en geeft het de melding weergegeven in Figuur 82.

27/05/2016 16:36:22: Connected to PLC but not receiving data!

Figuur 82: Verbonden met PLC maar ontvangt geen data

## 2. Ingeven van een verkeerd telegram.

Na de controle van de PLC-connectie wordt de data toegevoegd aan een record op basis van het ingegeven telegram. Als beide telegrammen overeenkomen wordt de melding in Figuur 83 weergegeven.

```
30/05/2016 11:11:18: telegram = received data
```

*Figuur 83: Correct telegram*

Indien deze niet overeenkomen worden volgende meldingen gegeven. Figuur 84 geeft de melding weer als het ingegeven telegram te klein is.

```
30/05/2016 11:09:34: The received data is larger then the set telegram !
```

*Figuur 84: Ingesteld telegram is te klein*

Figuur 85 geeft de melding weer als het ingegeven telegram te groot is.

```
30/05/2016 11:11:44: The set telegram is larger then the received data!
```

*Figuur 85: Ingesteld telegram is te groot*

Gevolg door volgende melding. Figuur 86 geeft de opvolgende melding dat de gebruiker het ingegeven telegram moet nakijken.

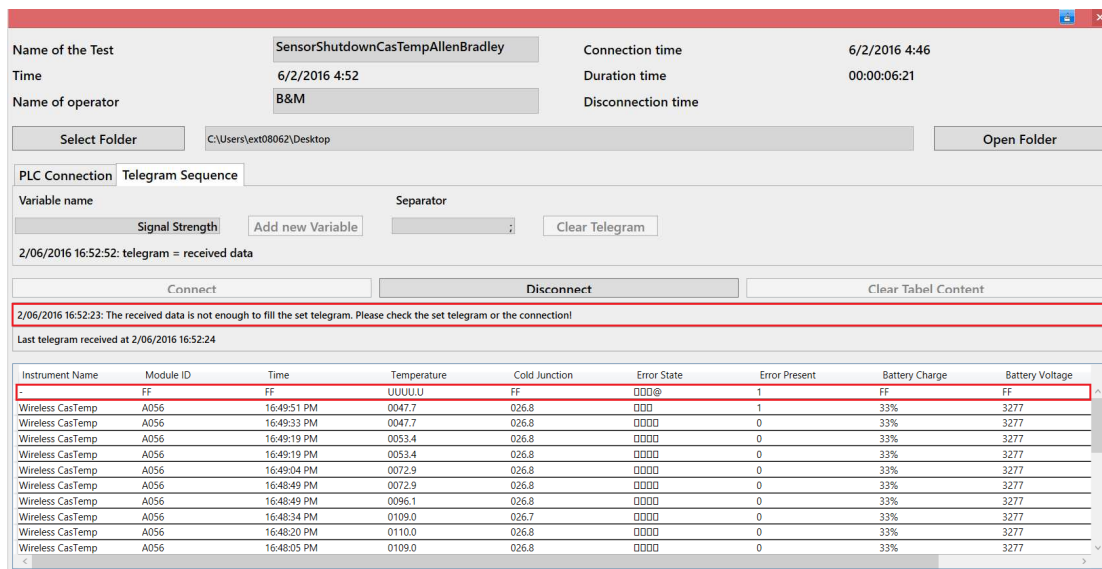
```
30/05/2016 16:25:43: The received data is not enough to fill the set telegram. Please check the set telegram or the connection!
```

*Figuur 86: Ontvangt geen of te weinig data*

In het geval dat het ingesteld telegram te groot gekozen is, wordt er geen data meer getoond en moet de gebruiker de verbinding verbreken en het telegram correct ingeven. Deze melding wordt ook getoond als de verbinding tussen PLC en pc wegvalt, aangezien er dan geen data meer binnenkomt.

### 3. Onderbreken van de communicatie tussen sensor en meettoestel.

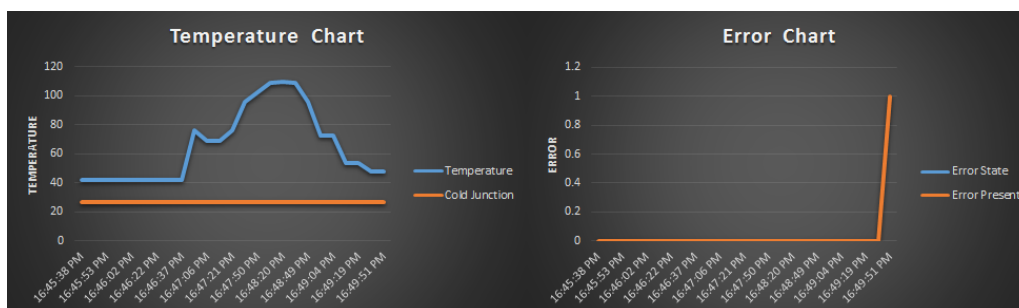
Door de communicatie te verbreken tussen de sensor, hier de CasTemp Wireless A056, stopt de dataoverdracht. Figuur 87 geeft de userinterface weer.



Figuur 87: Userinterface onderbreken van de communicatie tussen sensor en meettoestel

Indien de sensor niet meer bereikbaar is dan stuurt de CasTemp eerst een fout door en vervolgens stopt deze met het uitsturen van data. Het programma geeft dan ook de melding dat deze te weinig data binnen krijgt.

Volgende Figuur 88 geeft de temperatuur en error als grafiek weer.



Figuur 88: Temperatuur en error grafiek communicatietest sensor en meettoestel

Hier is te zien dat de datatransmissie stopt op het moment dat de error optreedt.

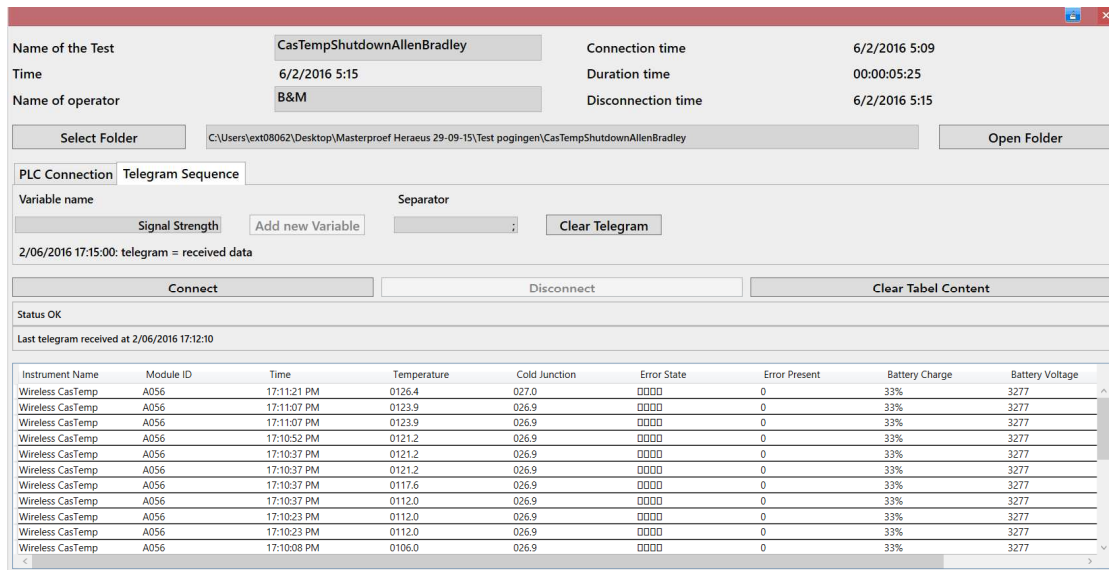
Onderstaande Figuur 89 geeft het opgeslagen CSV-bestand weer.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	SensorShutdownCasTempAllenBradley	executed by:	B&M	on	2/06/2016 16:42									
2	Instrument Name	Module ID	Time	Temperature	Cold Junction	Error State	Error Present	Battery Charge	Battery Voltage	Signal Strength				
3	Wireless CasTemp	A056	16:45:38 PM	42.2	26.7		0	33%	3277	90				
4	Wireless CasTemp	A056	16:45:53 PM	42.2	26.7		0	33%	3277	90				
5	Wireless CasTemp	A056	16:45:53 PM	42.2	26.7		0	33%	3277	89				
6	Wireless CasTemp	A056	16:46:08 PM	42.2	26.7		0	33%	3277	88				
7	Wireless CasTemp	A056	16:46:02 PM	42.2	26.7		0	33%	3277	88				
8	Wireless CasTemp	A056	16:46:02 PM	42.2	26.7		0	33%	3277	91				
9	Wireless CasTemp	A056	16:46:22 PM	42.2	26.7		0	33%	3277	91				
10	Wireless CasTemp	A056	16:46:37 PM	42.2	26.7		0	33%	3277	92				
11	Wireless CasTemp	A056	16:46:37 PM	42.2	26.7		0	33%	3277	90				
12	Wireless CasTemp	A056	16:46:52 PM	76.2	26.7		0	33%	3277	90				

Figuur 89: Opgeslagen CSV-bestand communicatietest tussen sensor en meettoestel

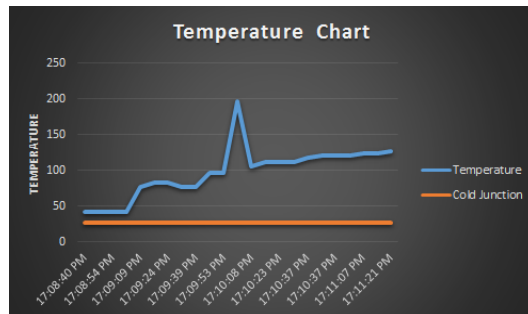
#### 4. Uitschakelen van het meettoestel.

Hier werd de CasTemp Wireless uitgeschakeld. Hierdoor is deze niet instaat om nog een foutmelding te verzenden. Het programma zal hier niets van merken dit is zichtbaar in Figuur 90.



Figuur 90: Userinterface uitschakelen van het meettoestel

Het programma blijft werken ondanks dat deze geen nieuwe waarde binnen krijgt. Dit komt omdat de CasTemp geen foutmelding heeft door kunnen sturen. Figuur 91 geeft de Temperatuur grafiek weer.



Figuur 91: Temperatuur grafiek uitschakelen meettoestel

De onderstaande Figuur 92 wordt bijhorend CSV-bestand getoond.

Instrument Name	Module ID	Time	Temperature	Cold Junction	Error State	Error Present	Battery Charge	Battery Voltage	Signal Strength
Wireless CasTemp	A056	17:08:40 PM	42.2	26.9	0	0	33%	3277	91
Wireless CasTemp	A056	17:08:54 PM	42.2	26.9	0	0	33%	3277	91
Wireless CasTemp	A056	17:08:54 PM	42.2	26.9	0	0	33%	3277	79
Wireless CasTemp	A056	17:09:09 PM	76.2	26.9	0	0	33%	3277	87
Wireless CasTemp	A056	17:09:24 PM	83.4	26.9	0	0	33%	3277	87
Wireless CasTemp	A056	17:09:24 PM	83.4	26.9	0	0	33%	3277	89
Wireless CasTemp	A056	17:09:39 PM	76.2	26.9	0	0	33%	3277	89
Wireless CasTemp	A056	17:09:39 PM	76.2	26.9	0	0	33%	3277	85
Wireless CasTemp	A056	17:09:53 PM	96.1	26.9	0	0	33%	3277	85
Wireless CasTemp	A056	17:09:53 PM	96.1	26.9	0	0	33%	3277	87
Wireless CasTemp	A056	17:10:08 PM	196.1	26.9	0	0	33%	3277	87
Wireless CasTemp	A056	17:10:08 PM	106	26.9	0	0	33%	3277	84
Wireless CasTemp	A056	17:10:23 PM	112	26.9	0	0	33%	3277	84
Wireless CasTemp	A056	17:10:23 PM	112	26.9	0	0	33%	3277	89
Wireless CasTemp	A056	17:10:37 PM	112	26.9	0	0	33%	3277	89
Wireless CasTemp	A056	17:10:37 PM	117.6	26.9	0	0	33%	3277	89
Wireless CasTemp	A056	17:10:37 PM	121.2	26.9	0	0	33%	3277	89
Wireless CasTemp	A056	17:10:37 PM	121.2	26.9	0	0	33%	3277	90
Wireless CasTemp	A056	17:10:52 PM	121.2	26.9	0	0	33%	3277	90
Wireless CasTemp	A056	17:11:07 PM	123.9	26.9	0	0	33%	3277	90
Wireless CasTemp	A056	17:11:07 PM	123.9	26.9	0	0	33%	3277	87
Wireless CasTemp	A056	17:11:21 PM	126.4	27	0	0	33%	3277	87
The test took:		00:00:05:25							

Figuur 92: CSV-bestand uitschakelen van het meettoestel

## 5. Onderbreken van de communicatie tussen meetinstrument en PLC.

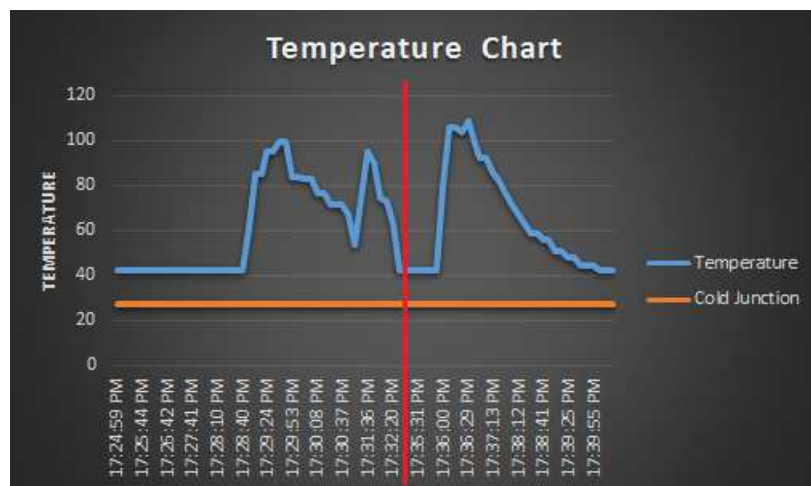
Indien de connectie tussen meetinstrument en PLC verbroken wordt, zal het programma dit niet registreren. Dit wordt weer gegeven in Figuur 93.

Instrument Name	Module ID	Time	Temperature	Cold Junction	Error State	Error Present	Battery Charge	Battery Voltage
Wireless CasTemp	A056	17:35:45 PM	0042.2	027.0	0000	0	33%	3276
Wireless CasTemp	A056	17:35:31 PM	0042.2	027.0	0000	0	33%	3276
Wireless CasTemp	A056	17:35:31 PM	0042.2	027.0	0000	0	33%	3277
Wireless CasTemp	A056	17:35:16 PM	0042.2	027.0	0000	0	33%	3277
Wireless CasTemp	A056	17:35:16 PM	0042.2	027.0	0000	0	33%	3277
Wireless CasTemp	A056	17:35:02 PM	0042.2	027.0	0000	0	33%	3277
Wireless CasTemp	A056	17:32:20 PM	0063.2	027.1	0000	0	33%	3277
Wireless CasTemp	A056	17:32:05 PM	0073.2	027.1	0000	0	33%	3277
Wireless CasTemp	A056	17:32:05 PM	0074.6	027.0	0000	0	33%	3277
Wireless CasTemp	A056	17:31:50 PM	0089.8	027.1	0000	0	33%	3276
Wireless CasTemp	A056	17:31:36 PM	0094.8	027.0	0000	0	33%	3276

Figuur 93: Userinterface onderbreken van de communicatie tussen meetinstrument en PLC

Zoals weer gegeven zal het programma de datalogging ongestoord verder zetten als de connectie terug herstelt wordt.

In onderstaande Figuur 94 wordt de temperatuur grafiek weergegeven.

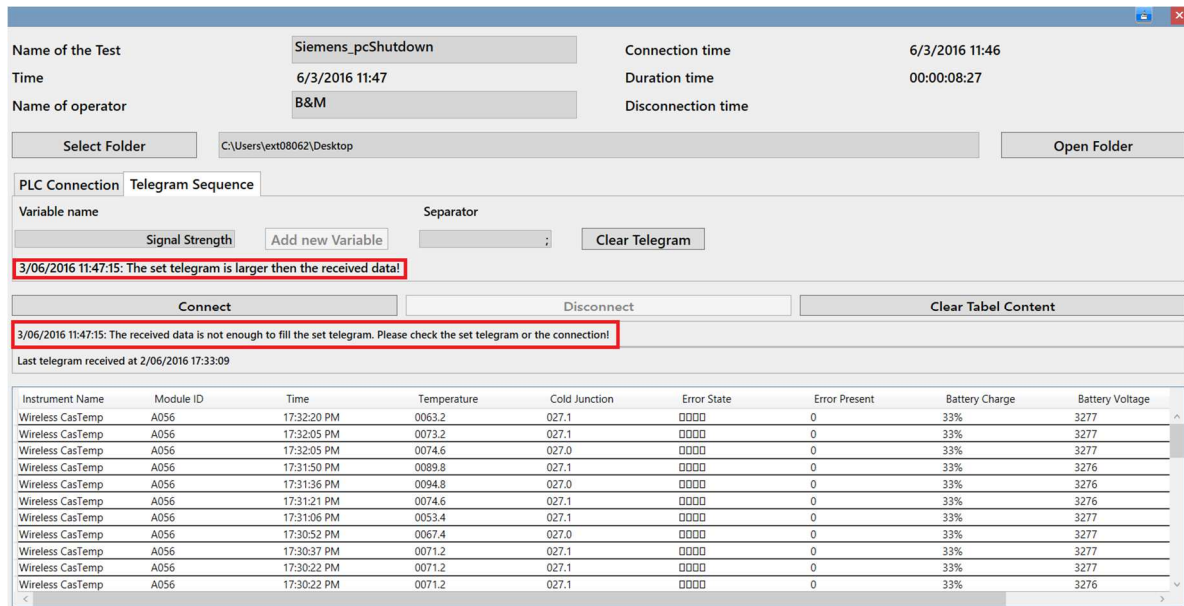


Figuur 94: De temperatuur grafiek communicatietest meetinstrument en PLC

Hier is wederom duidelijk te zien dat na het herstellen van de connectie de datatransmissie ongestoord verder wordt gezet. Dit heeft als voordeel dat de datatransmissie verder gaat, maar het nadeel is dat het programma dit niet registreert als fout.

## 6. Uitschakelen van de PLC.

Indien de PLC uitvalt of uitgeschakeld worden, wordt onderstaande melding gegeven. Figuur 95 geeft de Userinterface weer.

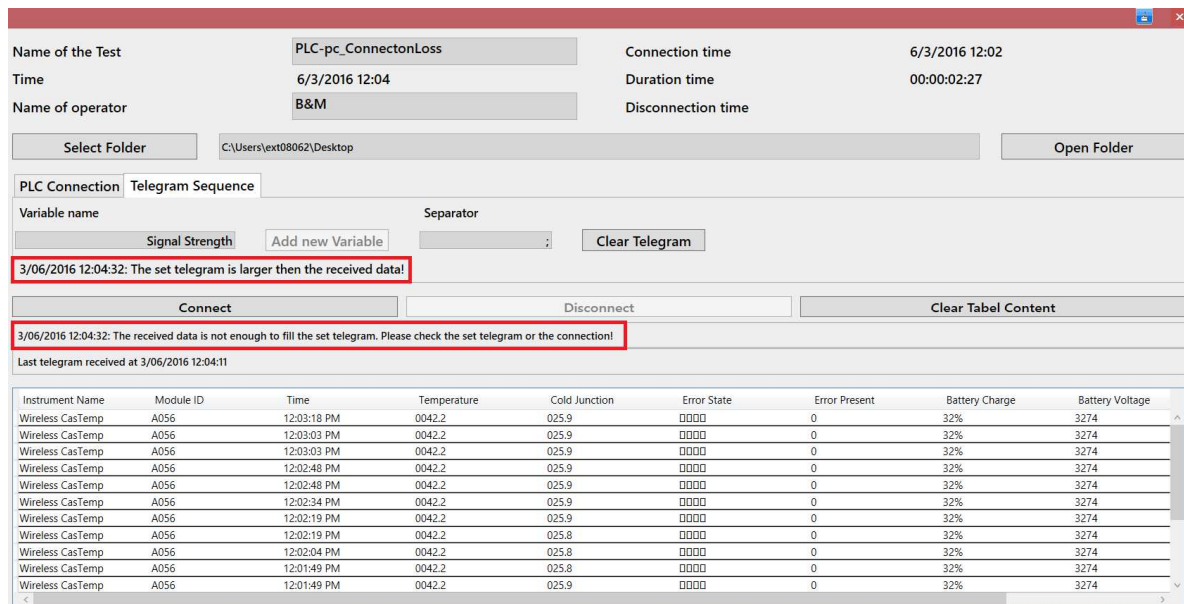


Figuur 95: Userinterface uitschakelen van de PLC

De melding geeft aan dat er te weinig data binnenkomt. In dit geval komt er dus geen data binnen aangezien de PLC uitgeschakeld is.

## 7. Onderbreken van de communicatie tussen PLC en pc.

Hier werd de verbinding tussen PLC en pc verbroken. Figuur 96 geeft de Userinterface weer.



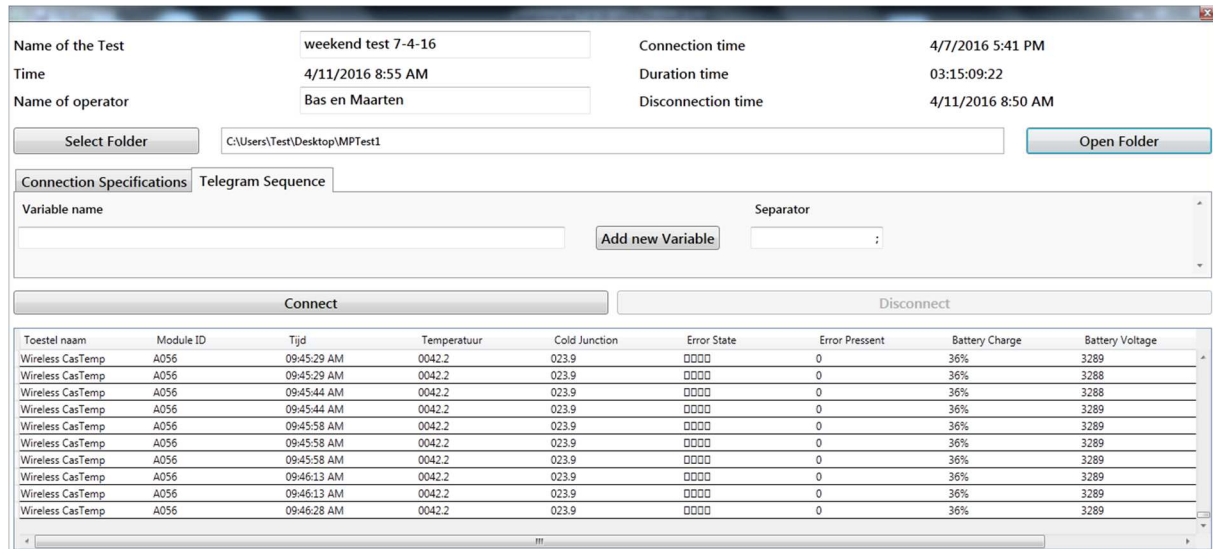
Figuur 96: Userinterface onderbreken van de communicatie tussen PLC en pc

Deze test geeft hetzelfde resultaat als het uitschakelen van de PLC, aangezien er ook hier geen data meer binnenkomt.

## 8. Duurtest met Siemens PLC en Castemp.

Na het afronden van de algemene programmatie en het aanpassen van de Userinterface werd er een duurtest met de Siemens PLC uitgevoerd. De PLC werd gekoppeld aan de CasTemp. Bij de duurtesten werd gekozen voor de CasTemp Wireless omdat deze, in combinatie met zijn gelijknamig module de A056, in staat is om continu met een interval van ca. 15 seconden data uit te sturen. Voor andere toestellen moeten testmodules gebruikt worden, welke niet altijd beschikbaar zijn.

Figuur 97 geeft de interface weer na de duurtest.



Figuur 97: Userinterface duurtest met Siemens PLC en Castemp

Op deze afbeelding is te zien wanneer de test gestart en gestopt is en dat de test 3 dagen 15 uur 9 minuten en 22 seconden geduurd heeft. De data werd succesvol gelogd naar een .CSV-bestand, weergegeven in Figuur 98.

Toestel naam	Module ID	Tijd	Temperatuur	Cold Junction	Error State	Error Present	Battery Charge	Battery Voltage	Signal Strength
Wireless CasTemp	A056	18:37:46 PM	42,2	25,3	0	0	36%	3290	87
Wireless CasTemp	A056	18:37:46 PM	42,2	25,3	0	0	36%	3290	89
Wireless CasTemp	A056	18:38:01 PM	42,2	25,3	0	0	36%	3290	89
Wireless CasTemp	A056	18:38:01 PM	42,2	25,4	0	0	36%	3290	85
Wireless CasTemp	A056	18:38:16 PM	42,2	25,4	0	0	36%	3290	85
Wireless CasTemp	A056	18:38:30 PM	42,2	25,3	0	0	36%	3290	87
Wireless CasTemp	A056	18:38:45 PM	42,2	25,3	0	0	36%	3290	82
Wireless CasTemp	A056	18:39:00 PM	42,2	25,3	0	0	36%	3290	87
Wireless CasTemp	A056	18:39:14 PM	42,2	25,3	0	0	36%	3290	87
Wireless CasTemp	A056	18:39:29 PM	42,2	25,3	0	0	36%	3290	87
Wireless CasTemp	A056	18:39:44 PM	42,2	25,3	0	0	36%	3290	83
Wireless CasTemp	A056	18:39:58 PM	42,2	25,3	0	0	36%	3290	85
Wireless CasTemp	A056	18:39:58 PM	42,2	25,3	0	0	36%	3290	85
Wireless CasTemp	A056	18:39:58 PM	42,2	25,3	0	0	36%	3290	82
Wireless CasTemp	A056	18:39:58 PM	42,2	25,3	0	0	36%	3290	81
Wireless CasTemp	A056	18:40:13 PM	42,2	25,3	0	0	36%	3290	81
Wireless CasTemp	A056	18:40:13 PM	42,2	25,3	0	0	36%	3290	86
Wireless CasTemp	A056	18:40:28 PM	42,2	25,3	0	0	36%	3290	86
Wireless CasTemp	A056	18:40:42 PM	42,2	25,3	0	0	36%	3290	86
Wireless CasTemp	A056	18:40:42 PM	42,2	25,3	0	0	36%	3290	88
Wireless CasTemp	A056	18:40:57 PM	42,2	25,4	0	0	36%	3290	88
Wireless CasTemp	A056	18:40:57 PM	42,2	25,4	0	0	36%	3290	85
Wireless CasTemp	A056	18:41:12 PM	42,2	25,3	0	0	36%	3290	85
Wireless CasTemp	A056	18:41:12 PM	42,2	25,3	0	0	36%	3290	86
Wireless CasTemp	A056	18:41:12 PM	42,2	25,3	0	0	36%	3290	88
Wireless CasTemp	A056	18:41:26 PM	42,2	25,3	0	0	36%	3290	88
Wireless CasTemp	A056	18:41:26 PM	42,2	25,3	0	0	36%	3290	88
Wireless CasTemp	A056	18:41:41 PM	42,2	25,3	0	0	36%	3290	88

Figuur 98: CSV-bestand duurtest met Siemens PLC en Castemp

Gedurende deze test werden 36766 waarden gelogd zonder dat er foutmeldingen zijn op getreden.

## 9. Duurtest met Allen Bradley en CasTemp.

Na het afronden van de programmatie, om te communiceren met de Allen Bradley PLC, en het verder aanpassen van de Userinterface werd er een duurtest met de Allen Bradley PLC uitgevoerd. De PLC werd ook hier gekoppeld aan de CasTemp Wireless.

Figuur 99 geeft de interface weer na de duurtest.

The screenshot shows the user interface for the Allen Bradley PLC test. At the top, there are fields for 'Name of the Test' (AllenBradleyWeekendTest28-04), 'Time' (5/2/2016 8:59 AM), 'Name of operator' (Bas & Maarten), 'Connection time' (4/28/2016 5:11 PM), 'Duration time' (03:15:48:11), and 'Disconnection time'. Below these are buttons for 'Select Folder' and 'Open Folder'. The 'PLC Connection' section includes a 'Telegram Sequence' tab, a 'Variable name' field (SignalStrength), an 'Add new Variable' button, a 'Separator' field, and a 'Clear Telegram' button. At the bottom, there are 'Connect', 'Disconnect', and 'Clear Label Content' buttons. A data table is displayed with the following columns: InstrumentName, ModuleID, Time, Temperature, ColdJunction, ErrorState, ErrorPresent, BatteryCharge, and BatteryVoltage. The table contains 15 rows of data, all showing a temperature of 0042.2 and a battery charge of 34%.

InstrumentName	ModuleID	Time	Temperature	ColdJunction	ErrorState	ErrorPresent	BatteryCharge	BatteryVoltage
Wireless CasTemp	A056	09:58:33 AM	0042.2	025.6	0000	0	34%	3287
Wireless CasTemp	A056	09:58:18 AM	0042.2	025.6	0000	0	34%	3287
Wireless CasTemp	A056	09:58:18 AM	0042.2	025.6	0000	0	34%	3287
Wireless CasTemp	A056	09:58:04 AM	0042.2	025.6	0000	0	34%	3287
Wireless CasTemp	A056	09:57:49 AM	0042.2	025.6	0000	0	34%	3287
Wireless CasTemp	A056	09:57:49 AM	0042.2	025.6	0000	0	34%	3287
Wireless CasTemp	A056	09:57:34 AM	0042.2	025.6	0000	0	34%	3287
Wireless CasTemp	A056	09:57:20 AM	0042.2	025.6	0000	0	34%	3287
Wireless CasTemp	A056	09:57:20 AM	0042.2	025.6	0000	0	34%	3287
Wireless CasTemp	A056	09:57:20 AM	0042.2	025.6	0000	0	34%	3287
Wireless CasTemp	A056	09:57:05 AM	0042.2	025.6	0000	0	34%	3287

Figuur 99: Userinterface duurtest met Allen Bradley en CasTemp

Op deze afbeelding is te zien wanneer de test gestart is en dat de test 3 dagen 15 uur 48 minuten en 11 seconden geduurd heeft. De data werd succesvol gelogd naar een .CSV-bestand, weergegeven in Figuur 100.

The screenshot shows an Excel spreadsheet with the following data:

InstrumentName	ModuleID	Time	Temperature	ColdJunction	ErrorState	ErrorPresent	BatteryCharge	BatteryVoltage
Wireless CasTemp	A056	18:10:59 PM	42.2	27.7	0	0	35%	3289
Wireless CasTemp	A056	18:11:14 PM	42.2	27.7	0	0	35%	3289
Wireless CasTemp	A056	18:11:28 PM	42.2	27.7	0	0	35%	3289
Wireless CasTemp	A056	18:11:28 PM	42.2	27.7	0	0	35%	3289
Wireless CasTemp	A056	18:11:43 PM	42.2	27.7	0	0	35%	3289
Wireless CasTemp	A056	18:11:58 PM	42.2	27.6	0	0	35%	3289
Wireless CasTemp	A056	18:11:58 PM	42.2	27.6	0	0	35%	3289
Wireless CasTemp	A056	18:12:12 PM	42.2	27.6	0	0	35%	3289
Wireless CasTemp	A056	18:12:27 PM	42.2	27.6	0	0	35%	3289
Wireless CasTemp	A056	18:12:27 PM	42.2	27.6	0	0	35%	3289
Wireless CasTemp	A056	18:12:42 PM	42.2	27.6	0	0	35%	3289
Wireless CasTemp	A056	18:12:56 PM	42.2	27.6	0	0	35%	3289
Wireless CasTemp	A056	18:13:11 PM	42.2	27.6	0	0	35%	3289
Wireless CasTemp	A056	18:13:26 PM	42.2	27.6	0	0	35%	3289
Wireless CasTemp	A056	18:13:26 PM	42.2	27.6	0	0	35%	3289
Wireless CasTemp	A056	18:13:40 PM	42.2	27.6	0	0	35%	3289
Wireless CasTemp	A056	18:13:40 PM	42.2	27.6	0	0	35%	3289
Wireless CasTemp	A056	18:13:55 PM	42.2	27.6	0	0	35%	3289
Wireless CasTemp	A056	18:14:10 PM	42.2	27.6	0	0	35%	3289
Wireless CasTemp	A056	18:14:10 PM	42.2	27.6	0	0	35%	3289
Wireless CasTemp	A056	18:14:10 PM	42.2	27.6	0	0	35%	3289
Wireless CasTemp	A056	18:14:24 PM	42.2	27.5	0	0	35%	3289
Wireless CasTemp	A056	18:14:39 PM	42.2	27.5	0	0	35%	3287
Wireless CasTemp	A056	18:14:54 PM	42.2	27.5	0	0	35%	3289
Wireless CasTemp	A056	18:15:08 PM	42.2	27.5	0	0	35%	3289
Wireless CasTemp	A056	18:15:23 PM	42.2	27.5	0	0	35%	3289
Wireless CasTemp	A056	18:15:38 PM	42.2	27.5	0	0	35%	3289
Wireless CasTemp	A056	18:15:52 PM	42.2	27.5	0	0	35%	3289

Figuur 100: CSV-bestand duurtest met Allen Bradley en CasTemp

Gedurende deze test werden 32219 waarden gelogd, zonder dat er foutmeldingen zijn op getreden.



## 10. E-Line test met Allen Bradley.

Dit zijn bijkomstige testen om de verbinding met een ander meetinstrument te testen. Hier is een Allen Bradley gekoppeld aan een DigiTemp-E Wireless meetinstrument. De Figuur 101 geeft de Userinterface van de test weer.

The screenshot shows the software interface for an E-Line test. It includes fields for test name (ElineAB), time (6/3/2016 1:01 PM), operator (B&M), and connection details. A table below displays recorded data points for temperature, cold junction, and place ID over time.

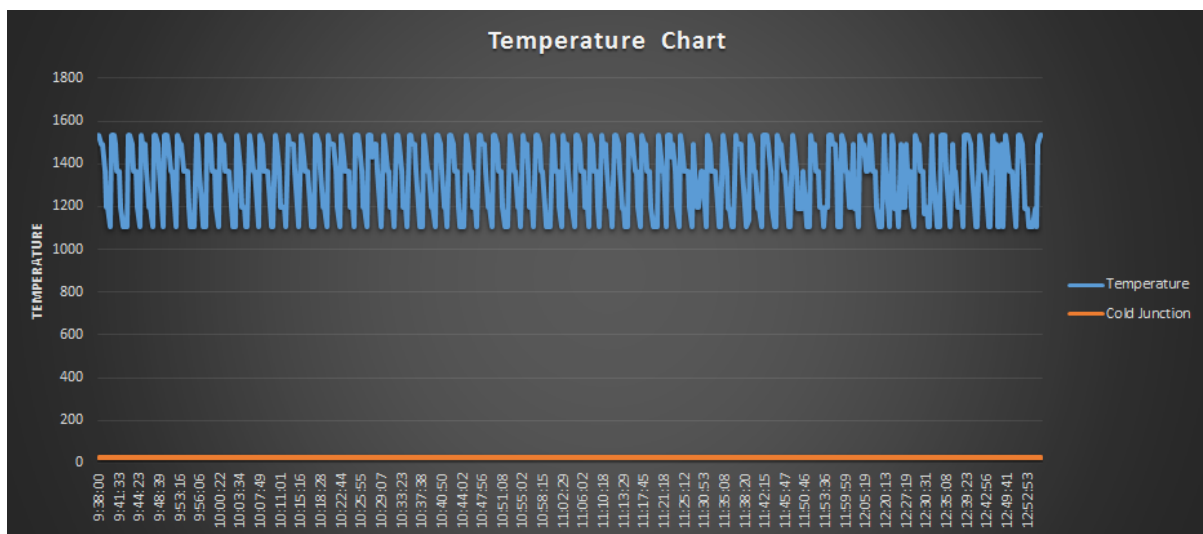
Instrument Name	Temperature	Cold Junction	Place ID	Date	Time	Empty
DIGITEMP-E	1534.3	0028.2	0001	03/06/16	12:59:16	
DIGITEMP-E	1534.3	0028.2	0001	03/06/16	12:57:08	
DIGITEMP-E	1492.8	0028.2	0001	03/06/16	12:57:08	
DIGITEMP-E	1104.5	0028.2	0001	03/06/16	12:55:43	
DIGITEMP-E	1192.3	0028.2	0001	03/06/16	12:55:22	
DIGITEMP-E	1104.3	0028.2	0001	03/06/16	12:55:22	
DIGITEMP-E	1104.4	0028.3	0001	03/06/16	12:53:14	
DIGITEMP-E	1104.7	0028.3	0001	03/06/16	12:52:53	
DIGITEMP-E	1191.7	0028.3	0001	03/06/16	12:52:53	
DIGITEMP-E	1191.7	0028.3	0001	03/06/16	12:52:32	
DIGITEMP-E	1364.8	0028.3	0001	03/06/16	12:52:32	
DIGITEMP-E	1491.9	0028.3	0001	03/06/16	12:52:10	

Figuur 101: Userinterface E-Line test met Allen Bradley

Op de userinterface is te zien dat de test 3uur 21 minuten en 25seconden gelopen heeft. Ook is er te zien dat om 11u32 als laatste een fout is opgetreden. Deze fout een onderbreking in de communicatie van meetinstrument naar PLC. De melding geeft ook weer dat er verbinding was met de PLC maar er geen dataoverdracht gebeurde.

Uit het .CSV-bestand kan afgeleid worden dat er 514 waardes gelogd zijn.

Figuur 102 geeft de temperatuurcurve in functie van de tijd.



Figuur 102: Temperatuur curve E-Line test met Allen Bradley



## **Conclusie:**

Door in de PLC de informatie, in verband met de connectie tussen meetinstrument en PLC, uit te lezen kunnen fouten zoals:

- het uitvallen of uitschakelen van het meetinstrument;
- het wegvallen van de connectie tussen meetinstrument en PLC.

wel gedetecteerd en verwerkt worden. Het C#-programma zal hier niets van merken en verder gaan als de fout verholpen is. De PLC's zijn al voorzien van een extra datablok welke informatie bijhoudt over de connectie met het meetinstrument. Informatie over deze datablokken is per PLC terug te vinden:

- voor Siemens in hoofdstuk 3.1.3;
- voor Allen Bradley in hoofdstuk 3.2.3;
- voor Schneider Electric in hoofdstuk 3.3.3.

De in voorgaande hoofdstukken vernoemde datablokken dienen nog uitgelezen te worden in het C#-programma.

Fouten die ontstaan door:

- het uitvallen of afschakelen van de PLC;
- het wegvallen van de connectie tussen PLC en pc.

geven, volgens de huidige programmatie, een melding dat er te weinig informatie binnenkomt. Zo blijft het programma doorlopen tot de verbinding hersteld wordt, waarna de normale werking hervat wordt. De melding blijft wel staan met datum en tijd waarop deze als laatste voorgekomen is. Door het implementeren van een, op tijd gebaseerde, lus kan het programma afgesloten worden indien er te lang geen data binnenkomt.

Uit deze conclusies ontstaan een aantal toekomstige uitbreidingen:

- Het uitlezen van de PLC-datablokken met informatie over de connectie;
- Het implementeren van een lus voor het programma af te sluiten indien er te lang geen data binnenkomt.



## 5 Besluit

Deze masterproef is ontstaan uit de vraag naar een klantvriendelijke low-cost ondersteuning bij de integratie van de meetinstrumenten van Heraeus. Dankzij de testopstelling is het bedrijf nu in staat om een betere ondersteuning te bieden aan al hun klanten, voor de integratie van de meetinstrumenten in de klant zijn machinenetwerk (Level2-netwerk). Ook zal het op voorhand zoeken naar fouten gemakkelijker worden.

Als de vooropgestelde doelstellingen vergeleken worden met de behaalde resultaten, kunnen we besluiten dat volgende doelstellingen gehaald zijn:

- ✓ de testopstelling kan gestart worden door een individu zonder kennis van PLC- of C#-programmatie;
- ✓ de Siemens, Allen Bradley en Schneider PLC's zijn in staat om data afkomstig van de meetinstrumenten te bufferen;
- ✓ data wordt binnen drie seconden in de PLC gebufferd;
- ✓ de PLC controleert zelf de connectie met het meetinstrument en steekt de statusinformatie in een aparte datablok;
- ✓ de pc applicatie is in staat om de data uit de Siemens en Allen Bradley PLC te lezen;
- ✓ de PLC-buffer wordt ook binnen drie seconden uitgelezen door de applicatie;
- ✓ de communicatie tussen PLC en meetinstrument verloopt in één richting;
- ✓ de setup is eenvoudig uit te breiden voor toekomstige instrumenten waarvan de configuratiefiles (GSD-files) beschikbaar zijn;
- ✓ de applicatie controleert de voltalligheid van de datapakketten en de updatetijd van de datapakketten;
- ✓ de userinterface is een eenvoudige weergave van de data met de mogelijkheid tot het exporteren van de data;
- ✓ de testopstelling is in staat om minimaal 24u data te loggen.

De niet behaalde doelstelling zijn volgende:

- ✗ de applicatie is niet in staat om data uit de Schneider PLC te lezen;
- ✗ de mogelijkheid tot meerdere dataformaten toe te laten, wat nu enkel ASCII is;
- ✗ de door de PLC gecontroleerde cyclustijd wordt nog niet door het programma uitgelezen en verwerkt.

Buiten de niet behaalde doelstelling zijn mogelijke toekomstige uitbreidingen:

- de communicatie tussen PLC en meetinstrument half tot full duplex laten verlopen;
- de statusinformatie uit de PLC halen, verwerken en weergeven in de User Interface;
- meerdere communicatie protocollen en PLC-types;
- de userinterface uitbreiden met een handleiding om de verschillende PLC's te configureren, GSD-files in te lezen en te downloaden.



## LiteratuurLijst

- [1] P. N. e. (PNO), PROFIBUS System Description - Technology and Application, Karlsruhe: PROFIBUS Nutzerorganisation e.V. (PNO), 2010.
- [2] „Engeneering-online.nl,” 2009. [Online]. Available: <http://www.engineering-online.nl/?com=content&action=profibus>. [Geopend 25 februari 2016].
- [3] A. Grenier, „Redundancy in EtherNet/IP systems,” *ISA Publications*, p. Juli/Augustus, 2011.
- [4] A. Ochoa, „arturochoa.wordpress.com,” wordpress, 07 11 2013. [Online]. Available: <https://arturochoa.files.wordpress.com/2013/07/jumbotag.png>. [Geopend 15 04 2016].
- [5] „dreamstime.com,” [Online]. Available: <http://nl.dreamstime.com>. [Geopend 21 maart 2016].
- [6] „www.rtaautomation.com,” Real Time Automation, 2016. [Online]. Available: <http://www.rtaautomation.com/technologies/modbus-rtu/>. [Geopend 21 maart 2016].
- [7] m. org, MODBUS Application Protocol Specification V1.1b3, 2012.
- [8] V. Pignato, *Wireless E-line Manual*, Houthalen, Belgium: Heraeus Electro-Nite International N.V..
- [9] „<http://www.mesta-automation.com>,” mesta-automation, 2012. [Online]. Available: <http://www.mesta-automation.com/plc-pc-communication-with-c-a-quick-resume-about-data-exchange-libraries/#opensource>. [Geopend 04 2016].
- [10] „www.fs.com,” Fiberstore, 2009. [Online]. Available: <http://www.fs.com/blog/wp-content/uploads/2014/10/cable-media.gif>. [Geopend Maart 2016].
- [11] „www.webanswers.com,” WebAnswers , 2016. [Online]. Available: <http://www.webanswers.com/post-images/1/19/1350D05B-12B1-FC60-E07213BA65654642.jpg>. [Geopend Maart 2016].
- [12] „technet.microsoft.com,” Microsoft, 2016. [Online]. Available: [https://technet.microsoft.com/nl-nl/library/cc786900\(v=ws.10\).aspx](https://technet.microsoft.com/nl-nl/library/cc786900(v=ws.10).aspx). [Geopend Maart 2016].
- [13] „www.yourdictionary.com/,” LoveToKnow, Corp, 1996-2016. [Online]. Available: <http://cf.ydcdn.net/1.0.1.49/images/computer/TOPOLOGY.GIF>. [Geopend 2016].
- [14] D. Lane, „slideshare.net,” Wago Ltd, 28 augustus 2013. [Online]. Available: <http://www.slideshare.net/ProfibusUK/introduction-to-profinet-derek-lane>. [Geopend 26 april 2016].
- [15] „<http://w3.siemens.com/>,” Siemens AG, [Online]. Available: [http://w3.siemens.com/mcms/automation/en/industrial-communications/profinet/PublishingImages/profibus/profibus\\_chart\\_en\\_576.jpg](http://w3.siemens.com/mcms/automation/en/industrial-communications/profinet/PublishingImages/profibus/profibus_chart_en_576.jpg). [Geopend 28 april 2016].
- [16] B. Lieberth, „<http://www.profibus.com/>,” PROFIBUS/PROFINET Association South East Asia, [Online]. Available: <http://www.profibus.com/nc/pi-organization/regional-pi-associations/south-east-asia/downloads-and-media/downloads/iatex-presentations>. [Geopend 28 april 2016].
- [17] „<https://www.belden.com/>,” [Online]. Available: <https://www.belden.com/upload/lumberg-profinet-ethernetIP-ethernet-connectivity-solutions-catalog.pdf>. [Geopend 28 april 2016].
- [18] „<http://w3.siemens.com/>,” Siemens AG, [Online]. Available: [http://w3.siemens.com/mcms/automation/en/industrial-](http://w3.siemens.com/mcms/automation/en/industrial-communications/profinet/PublishingImages/profibus/profibus_chart_en_576.jpg)

communications/profinet/features\_and\_benefits/flexibility/pages/flexible\_topologies.aspx.  
[Geopend 28 april 2016].

- [19] „<http://www.vaeprosystech.cz/>,” 08 mei 2012. [Online]. Available:  
<http://www.vaeprosystech.cz/dokumentace/ac500/English/CHM-Files/CAA-Merger-7/System-Technology-Communication-Module/The-PROFINET-Communication-Module/PROFINET-Basics.htm>. [Geopend 02 mei 2016].
- [20] „<https://msdn.microsoft.com/>,” microsoft, 2016. [Online]. Available:  
[https://msdn.microsoft.com/en-us/library/aa970268\(v=vs.100\).aspx](https://msdn.microsoft.com/en-us/library/aa970268(v=vs.100).aspx). [Geopend 28 april 2016].
- [21] „<https://msdn.microsoft.com/>,” Microsoft, 2016. [Online]. Available:  
<https://msdn.microsoft.com/en-us/library/z1zx9t92.aspx>. [Geopend 28 april].
- [22] „<https://msdn.microsoft.com/>,” Microsoft, 2016. [Online]. Available: <https://msdn.microsoft.com>.  
[Geopend 28 april 2016].
- [23] P. Valerie, „Checkmate,” Heraeus Electro-Nite International N.V., Houthalen, Belgium.
- [24] V. Pignato, „CasTemp Wireless,” Heraeus Electro-Nite International N.V., Houthalen, Belgium, 2014.



## Bijlagen

De bijlagen zijn te vinden op de CD-ROM. Hierop zijn volgende delen te vinden:

- De voortzetting van de literatuurstudie met onderstaande protocollen:
  - CC-Link,
  - BACNet,
  - ControlNet,
  - RS232,
  - CAN bus.
- De PLC programma's (in pdf-formaat en de programma's zelf),
- De C#-applicatie (in pdf-formaat en het programma zelf).

# Auteursrechtelijke overeenkomst

Ik/wij verlenen het wereldwijde auteursrecht voor de ingediende eindverhandeling:  
**Datalogging en analyse door netwerkintegratie van PLC en PC**

Richting: **master in de industriële wetenschappen: energie-automatisering**  
Jaar: **2016**

in alle mogelijke mediaformaten, - bestaande en in de toekomst te ontwikkelen - , aan de Universiteit Hasselt.

Niet tegenstaand deze toekenning van het auteursrecht aan de Universiteit Hasselt behoud ik als auteur het recht om de eindverhandeling, - in zijn geheel of gedeeltelijk -, vrij te reproduceren, (her)publiceren of distribueren zonder de toelating te moeten verkrijgen van de Universiteit Hasselt.

Ik bevestig dat de eindverhandeling mijn origineel werk is, en dat ik het recht heb om de rechten te verlenen die in deze overeenkomst worden beschreven. Ik verklaar tevens dat de eindverhandeling, naar mijn weten, het auteursrecht van anderen niet overtreedt.

Ik verklaar tevens dat ik voor het materiaal in de eindverhandeling dat beschermd wordt door het auteursrecht, de nodige toelatingen heb verkregen zodat ik deze ook aan de Universiteit Hasselt kan overdragen en dat dit duidelijk in de tekst en inhoud van de eindverhandeling werd genotificeerd.

Universiteit Hasselt zal mij als auteur(s) van de eindverhandeling identificeren en zal geen wijzigingen aanbrengen aan de eindverhandeling, uitgezonderd deze toegelaten door deze overeenkomst.

Voor akkoord,

**Janssen, Bas**

**Sciot, Maarten**

Datum: **7/06/2016**