

DAF TRUCKS

Realisatie Stageopdracht

Lucas De Greef
Richting: Elektronica-ICT

Voorwoord

In dit voorwoord wil ik allereerst de mensen bedanken die mij gesteund en begeleid hebben tijdens mijn stage: Thomas Verschoor, Erwin Geboers, Geert Beckers en Wim Rijken. Ik wil deze mensen vooral bedanken omdat ik de kans heb gekregen om veel van hen te leren, zowel op communicatief vlak als op technisch vlak.

Verder wil ik ook mijn docenten bedanken voor hun rol in mijn educatie. Zij hebben mijn kennis verbreed en hebben mij voorzien van boeiend en leerzaam onderwijs gedurende mijn hele schoolcarrière. Hun inzet heeft mijn leerervaring verrijkt en mij gemotiveerd om te blijven leren en groeien.

Tot slot wil ik mijn familie bedanken voor hun steun en motivatie die zij mij hebben gegeven gedurende deze periode, evenals voor hun aanmoediging vanaf het begin.

Ik wens u alvast veel leesplezier toe.

Met vriendelijke groeten

Lucas De Greef

Inhoud

VOORWOORD	2
LIJST MET AFKORTINGEN	6
1. INLEIDING	7
2. DAF-GESCHIEDENIS	8
3. DAF PROBLEEMSTELLING	10
3.1. Huidig proces	10
3.2. Alternatieve/beter oplossing	10
3.2.1. OPTIFACT	11
3.2.2. Werken met freelancers	11
3.2.3. Inschakeling van bachelor student	11
OPGELEGDE TECHNOLOGIEËN	12
3.3. Programmatietaal C#	12
3.4. Microsoft SQL-server	12
3.5. PowerBI	12
3.6. Robotstudio	12
3.7. ABB PCSDK	12
3.8. Windows server	13
4. ONTWERPCONCEPT	14
4.1. Architectuur van het project	14
5. C# APPLICATIE SETUP	15
5.1. Logging library	15
5.2. ABB PCSDK-development setup	17
5.2.1. Installatie ABB PCSDK	17
5.2.2. SDK refereren naar in C# project	18
5.2.3. Error fouten die optreden	19
6. APPLICATIE ARCHITECTUUR	20
7. ABB ROBOT EVENTLOG WAT IS DAT?	22
7.1. Eventlog in ABB-robot	22
8. ABB EVENTLOG APPLICATIE	23
8.1. Wat is task scheduler en waarom toegepast	23
8.2. Configureren van Windows Task	23
8.3. CultureInfo veranderen	25
8.4. Robot controllers zoeken & ip adressen uit Excel halen	26

8.5. Mappen structuur uit robot halen	26
8.6. Connectie met Database	27
8.7. Eventlog data uitlezen	28
8.8. Data capteren in database	29
8.9. CSV Aanmaken en schrijven	29
8.10. Deploy applicatie	29
9. NIEUW PROJECT RAPID DATA LOGGEN	32
9.1. Wat is rapid data?	32
9.2. Lastang robot	32
9.3. Test in simulatie	34
9.4. Test in testcel	35
9.5. Test in productie	35
10. TOEGEVOEGDE WAARDE BACHELORPROEF	37
10.1. Eventlog rapport	37
10.2. Tipwear rapport	38
11. APPENDIX	39
11.1. Opgeloste problemen	39
11.1.1..NET n commando not found	39
12. BRONVERMELDING	42

Lijst met figuren

Figuur 1: DAF logo 1948	8
Figuur 2: Eerste DAF trucks	8
Figuur 3: DAF kroning factory of the future	9
Figuur 4: Bestaand systeem van error te loggen	10
Figuur 5: OPTIFACT Prijzen	11
Figuur 6: ABB PCSDK communicatie met Robot	13
Figuur 7: Backup server	13
Figuur 8: Architectuur applicatie	14
Figuur 9: Locatie Nuget packetmanager	15
Figuur 10: Nlog config file	16
Figuur 11: Inhoud config file	16
Figuur 12: ABB developercenter	17
Figuur 13: Locatie ABB SDK op PC	17
Figuur 14: dll files toevoegen in project	18
Figuur 15: Locatie 3 .dll files op pc	18
Figuur 16: Configuratie error van PCSDK en C#	19
Figuur 17: Eventlog voorbeeld van robot	22
Figuur 18: Task shedular Algemeen tabel	23
Figuur 19: Trigger instellen voor Task	24
Figuur 20: Keuze voor programma starten en Path naar het programma:	25
Figuur 21: CultureInfo taalverandering	25
Figuur 22: Tussen resultaat Data uitlezen	28
Figuur 23: build applicatie	29
Figuur 24: Keuze hoe applicatie wordt opgeslagen	30
Figuur 25: path locatie voor het opslaan van applicatie	30
Figuur 26: Applicatie voorwaarde instellingen	31
Figuur 27: Publish applicatie	31
Figuur 28: Rapid data voorbeeld gundata	32
Figuur 29: Lastang ABB-robot	33
Figuur 30: Lasttip van lastang	34
Figuur 31: Virtuele controller	34
Figuur 32: Testcells	35
Figuur 33: Resultaat rapid data uitlezen op basis van verandering	36
Figuur 34: Eventlog rapport	37
Figuur 35: Rapid data rapport	38

Lijst met afkortingen

SDK	Software development kit
ABB	Asea Brown Bover
DAF	Doorne's Automobiel Fabriek
PLC	Programmable logic controller
SQL	Structured query language
SCADA	Supervisory Control And Data Acquisition
IDE	integrated development environment
Ip	Internet Protocol
csv	comma-seperated values
PERS	Persistent

1. Inleiding

Deze bachelorproef behandelt het project uitgevoerd bij DAF Trucks, met als doel het ontwikkelen van een oplossing voor het vastleggen en benutten van de data gegenereerd door ABB-robots. Deze robots spelen een cruciale rol in het automatische lassen en het hanteren van materialen bij DAF Westerlo.

Het huidige proces bij DAF Westerlo maakt geen optimaal gebruik van de robotdata, wat resulteert in gemiste kansen voor procesverbetering en probleemoplossing. De belangrijkste doelstelling van dit project was om een applicatie te ontwikkelen die de eventlogs van de ABB-robots kan uitlezen en opslaan. Deze logs bevatten belangrijke informatie over de operationele status en eventuele fouten van de robots.

De aanpak bestond uit verschillende stappen, waaronder het configureren van een Windows Task Scheduler om de applicatie automatisch te laten draaien, en het ontwikkelen van een applicatie die de robotdata verzamelt en opslaat in een database. Deze data kan vervolgens worden geanalyseerd en gebruikt om de efficiëntie van de productieprocessen te verbeteren.

Het project resulteerde in een werkende applicatie die succesvol de eventlogs van meerdere robots kan uitlezen en opslaan. Dit biedt DAF de mogelijkheid om diepgaand inzicht te krijgen in de operationele prestaties van hun robots en potentiële problemen tijdig te identificeren. De applicatie werd getest zowel in een virtuele als in een fysieke omgeving, en gaf positieve resultaten in de productieomgeving.

Deze bachelorproef toont aan dat het mogelijk is om met behulp van de ABB PCSDK waardevolle data te verzamelen uit de robots bij DAF Westerlo. De ontwikkelde applicatie draagt bij aan een efficiënter productieproces door beter inzicht te geven in de prestaties van de robots en de mogelijkheid te bieden om snel in te grijpen bij problemen. Hierdoor kan DAF niet alleen de kwaliteit van de productie verbeteren, maar ook de kosten verminderen en de levensduur van hun apparatuur verlengen.

2. DAF-geschiedenis

DAF is opgericht op 1 april 1928 door de heer Hub van Doorne.

Het bedrijf had in 1932 de focus gelegd op het maken van aanhangwagens en opleggers.

Aan de hand van deze focus werd de volgende naam gekozen "Van Doorne's Aanhangwagenfabriek N.V".

Daarna is de naam gewijzigd in het jaar 1948. Het kreeg de naam die de dag van vandaag bekend is als "Van Doorne's Automobiel Fabriek" afgekort "DAF".



Figuur 1: DAF logo 1948

1 jaar na de naamswijziging werden de eerste twee DAF-truckchassis geïntroduceerd, bekend als de A-30 en de A-50. Dit markeerde het beginpunt van het volledige verhaal van DAF-trucks.



Figuur 2: Eerste DAF trucks

Na jarenlang ontwikkelen en produceren van DAF-trucks, werd het bedrijf getroffen door het noodlot. In het jaar 1993 werd het faillissement van het bedrijf aangekondigd. Enkele weken later werd er echter een nieuw bedrijf opgericht, DAF Trucks N.V. Drie jaar later werd DAF Trucks N.V. overgenomen door PACCAR, een van de grootste truckfabrikanten ter wereld. Dit bracht een gevoel van zekerheid met zich mee.

DAF Westerlo van vandaag.

DAF Westerlo is de DAF Assen- en Cabine fabriek gevestigd. Deze vertakking is geopend in 1966. In totaal werken er 2000 mensen op DAF Westerlo. De afgelopen jaren is DAF Westerlo op gebied van technologie implementatie een van de kop lopers in Vlaanderen.

In 2024 zijn hun als factory of the future verklaard.



Figuur 3: DAF kroning factory of the future

3. DAF Probleemstelling

3.1. Huidig proces

Bij DAF Westerlo zijn er momenteel heel veel ABB-robots aanwezig die al het automatisch lassen en/of vast pakken van verschillende materialen uitvoeren. Tijdens deze verschillende processen zijn er veel stappen/events aanwezig. Voor dat deze stageopdracht werd uitgeschreven, was er geen functionaliteit aanwezig om deze events uit de robot te halen. Al deze data die aanwezig was, werd niet optimaal benut door DAF.

Bij DAF Westerlo worden er momenteel ABB-robots gebruikt voor het assemblageproces van de DAF cabines. Tijdens dit gehele processen gebeuren er veel stappen en events die opgeslagen worden op de robots. Het systeem dat nu wordt toegepast om informatie uit de robots te halen is het volgende. De ABB-robot stuurt een code door bv 60150. Deze code wordt uitgelezen door de Siemens PLC. Er wordt in de PLC-logica voorzien dat het de betekenis van deze code ontcijfert. Dus in dit voorbeeld



Figuur 4: Bestaand systeem van error te loggen

geeft de 6 aan welk soort applicatie fout is van de robot. De 150 geeft aan wat voor soort type fout het is van de robot. Dit resultaat wordt dan door Wincc (SCADA-systeem) uitgelezen en wordt vervolgens wegeschreven in een database.

In dit huidig proces worden niet alle fouten van de robot gestuurd naar de PLC. Niet alle mogelijke fouten worden niet-gelogs en ook als warnings berichten en als informatie berichten die aanwezig zijn in de ABB-Robot. Omdat deze berichten niet worden doorgestuurd naar de PLC, kan ook het SCADA-systeem deze berichten niet wegschrijven naar een database, alsook deze berichten weergeven.

3.2. Alternatieve/beter oplossing

Zoals eerder besproken was het huidige wel een voldoening om een deel van alle fouten te loggen en een visueel beeld te krijgen. Het minpunt is dat het geen beeld van alle berichten die de robot lokaal logs aangeeft, en ook de oudere berichten worden overschreven. Om een duidelijke historie te hebben van het gehele proces en alle nuttige data te benutten is er hierop toegepast op deze bachelor. Er zijn ook andere alternatieven, deze zullen later besproken worden.

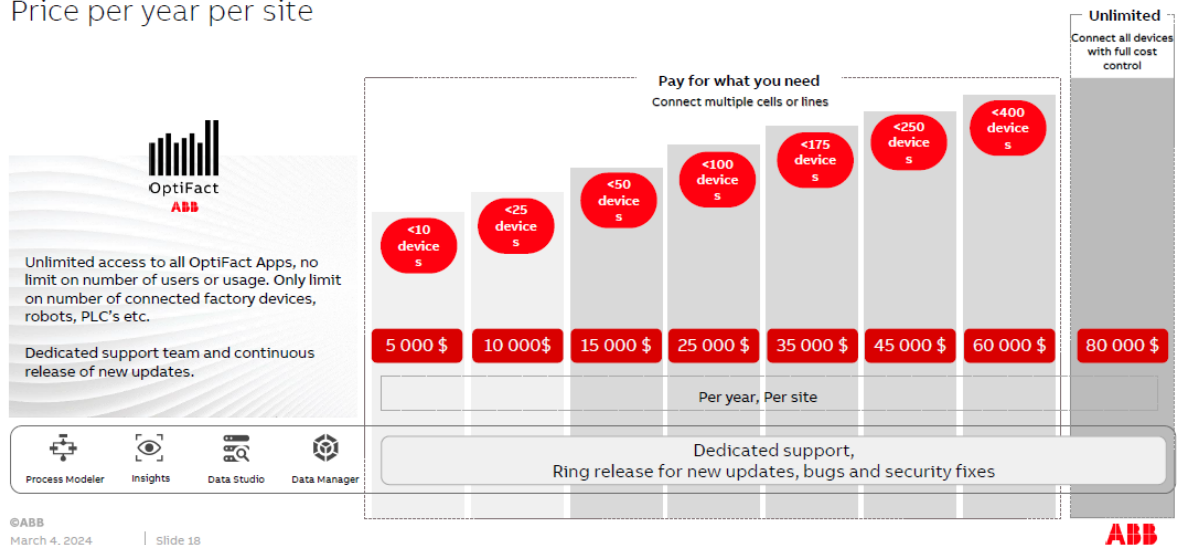
3.2.1. OPTIFACT

De robot fabrikant ABB heeft hun eigen monitoring product voor hun robots recent uitgebracht. Dit is een plug en play systeem. Het zorgt ervoor dat de klant, in dit geval DAF, niet veel tijd in moet steken om belangrijke data uit een robot te krijgen. Dit geeft wel mee dat ABB zelf weet wat zijn belangrijke parameters van de robots zijn. Daarnaast kunnen ze ook belangrijke data die relevant zijn voor een bepaalde gebruiken aantonen.

Grote nadeel aan deze keuze was dat er jaarlijkse service moet betaald worden, dat voor DAF een grote kost gaat zijn. Deze kost kan oplopen tot 85 000 euro.

OptiFact™ Pricing & Licensing

Price per year per site



Figuur 5: OPTIFACT Prijzen

3.2.2. Werken met freelancers

Er is ook de optie om voor externe personen een applicatie te maken dat de noden van DAF vereist. Hier zijn ook nadelen aan gebonden. Als er problemen in de toekomst optreden kan dit ook een extra kost opleveren als er geen broncode wordt mee gedeeld. Als deze optie erbij zit, heeft dit ook al een redelijk kostenplaatje wat extra met zich meegeeft.

De visualisatie gaat dan via een apart systeem worden gevisualiseerd wat bij OPTIFACT al inclusief was.

De voordelen om het via deze weg te doen was dat de noden van DAF afstembaar waren. Er waren functionaliteiten met de OPTIFACT-keuze.

3.2.3. Inschakeling van bachelor student

Van deze 2 keuzes is er een keuze gemaakt om voor een extern persoon een applicatie op maat te maken. Hierbij werd er gekozen dat dit ideaal zou zijn als bachelorproef waar een student aan kan werken. Deze dient een onderzoekwerk te doen en er een uitgewerkt proef neer te schrijven. Ook is er als doel de sterke en zwakke punten van deze SDK weer te geven.

Opgelegde technologieën

3.3. Programmatietaal C#

In deze proef werd C# als opgelegde programmeertaal gekozen. De hoofdreden is dat de projecten waar een applicatie voor gemaakt moeten worden in C# worden gemaakt. Voor onderhoud is dit ook beter, omdat dit ervoor zorgt dat alles onder programmeertaal valt en ook in het gehele microsoft co-systeem. Ook een grote troef is dat deze programmeertaal zeer flexibel is. Het kan gebruikt worden om API's te maken en/of applicaties.

3.4. Microsoft SQL-server

Microsoft SQL-server is het type database dat wordt toegepast bij DAF Westerlo. Microsoft SQL-server is een relatie type database. Dat betekent dat het de gegevenspunten opslaat en toegankelijk maakt die aan elkaar gerelateerd zijn. (What Is a Relational Database?, n.d.).

Relationele databases zijn gebaseerd op het relationele model, een intuïtieve, eenvoudige manier om gegevens in tabellen weer te geven. In een relationele database is elke rij in de tabel een record met een unieke ID, de sleutel. (What Is a Relational Database?, n.d.).

De kolommen van de tabel bevatten attributen van de gegevens, en elk record heeft gewoonlijk een waarde voor elk attribuut, waardoor het gemakkelijk wordt om de relaties tussen gegevenspunten vast te stellen. (What Is a Relational Database?, n.d.).

Het doel van de microsoft sql server omvat alle data die afkomstig is van de robots hier in op te slaan.

3.5. PowerBI

PowerBI heeft als doel om binnen DAF Westerlo rapportages te creëren met alle data die binnen het bedrijf worden verzameld. Ook de gegevens die afkomstig zijn van de robot zullen hierin gevisualiseerd en gerapporteerd worden.

3.6. Robotstudio

Robotstudio werd ook aangeraden om te installeren omdat de code van de ABB-robots hierin wordt geschreven, maar ook de simulatie van robots kan hier in worden uitgevoerd. Voordat er een productie omgeving testing ging uitgevoerd worden, kon dit allemaal eerst op een virtuele controller uitgevoerd worden.

Ook de hele mappen structuur van deze specifieke robot wordt in robotstudio weergegeven.

De navigatie naar de juiste module en parameters kan ook hierin gebeuren.

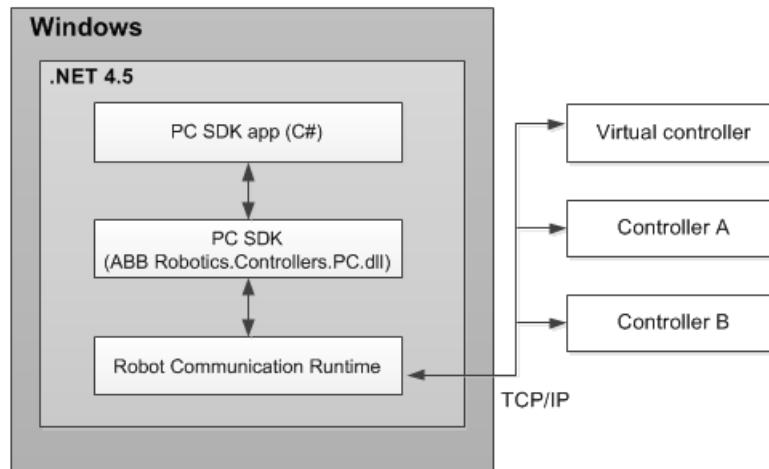
Alle bewegingen die de robot maakt werd hier visueel ingezien.

3.7. ABB PCSDK

De SDK dat ABB voorziet kan ervoor zorgen dat er communicatie is tussen C# applicatie en ABB-robot.

PC SDK biedt API's voor het ontwikkelen van pc-gebaseerde (C#) applicaties voor interactie met de robotcontroller. Het bevat de openbare API-laag, ABB.Robotics.Controllers.PC.dll, die de openbare API's blootlegt die kunnen worden gebruikt om applicaties te ontwikkelen voor het uitvoeren van verschillende bewerkingen met de robotcontroller.

De volgende afbeelding toont de softwarearchitectuur van PC SDK.
Deze softwarearchitectuur ondersteunt: Native 32 bit en 64 bit PC SDK Applicatie samen met Robotstudio Add-Ins (32 bit en 64 bit) (Overview, Software Architecture ABB PCSDK, n.d.)



Figuur 6: ABB PCSDK communicatie met Robot

3.8. Windows server

De Windows server heeft de volgende doelen in deze bachelor proef. Deze server gaat zich in het netwerk bevinden waar alle robots in cabine fabriek zijn op aangesloten. De C# applicatie die met de robots communiceert gaat hier ook op de achtergrond uitgevoerd worden. Ook gaan verder uitbreidingen op deze Windows server uitgevoerd worden.



Figuur 7: Backup server

4. Ontwerpconcept

4.1. Architectuur van het project

Nu gaat de architectuur route besproken worden.
De code ontwikkeling is gebeurd in Visual studio.

De Visual Studio IDE is een creatief startpunt dat kan gebruikt worden om codes te bewerken, fouten op te sporen en te bouwen, en vervolgens een app te publiceren.
(Visual Studio: IDE and Code Editor for Software Developers and Teams, 2024)

Door alle deze ingebouwde tools werd het ook aangeraden om in deze omgeving te ontwikkelen en ook om dezelfde standaard te behouden dat door andere collega's wordt gebruikt.
In IDE wordt er een applicatie ontwikkelt in C#. Met deze applicatie kunnen ABB-robots worden gecommuniceerd en de benodigde data worden verkregen voor volledige benutting.

Microsoft SQL-server is een databasebeheersysteem dat lokaal in DAF Westerlo staat. Alle sql querys worden in dit systeem ook uitgevoerd.

Als deze data correct is opgeslagen, is de laatste stap dit visueel in kaart te brengen.

Hiervoor is gebruik gemaakt van PowerBI. Deze gaat connectie maken met de lokale SQL-server. Als deze connectie is gelukt gaat het alle data dat afkomstig is van de robots ook duidelijke in kaart brengen.



Figuur 8: Architectuur applicatie

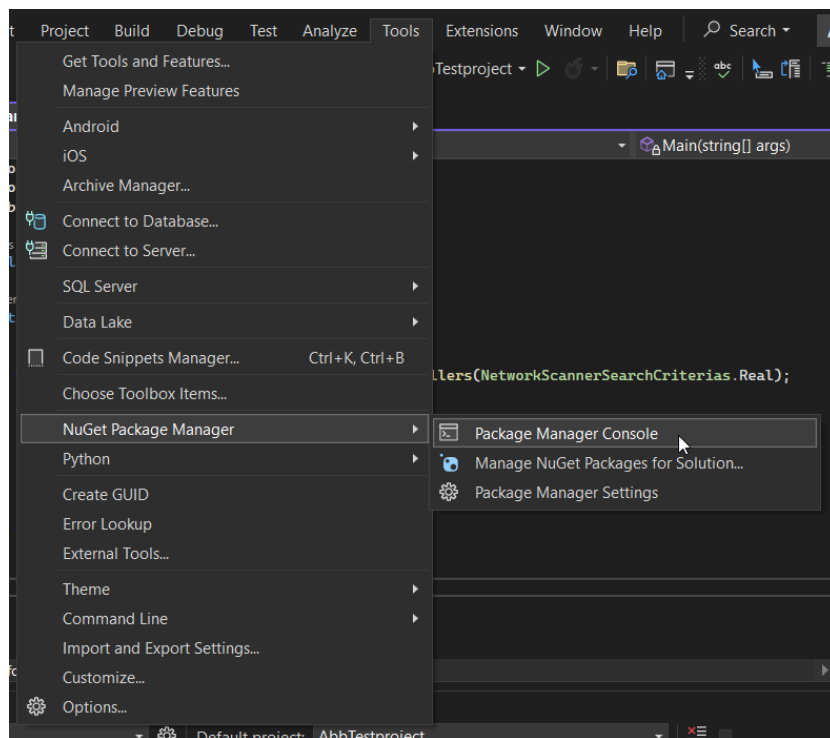
5. C# applicatie setup

5.1. Logging library

Tijdens de eerste weken was er een kennismaking met verschillende mensen van DAF. Ook waren er meetings gepland voor deze bachelor proef. In deze meeting was er de voorkeur gelegd om NLog als logging library te gebruiken. De reden hiervoor is dat de personeel van DAF hier een eerdere ervaring al van hadden.

Tijdens de ontwikkeling van de applicatie was dit ook de eerste stap dat werd ondernomen.

Om gebruik te maken van deze library is het aangeraden om dit te doen via de Nuget packetmanager.

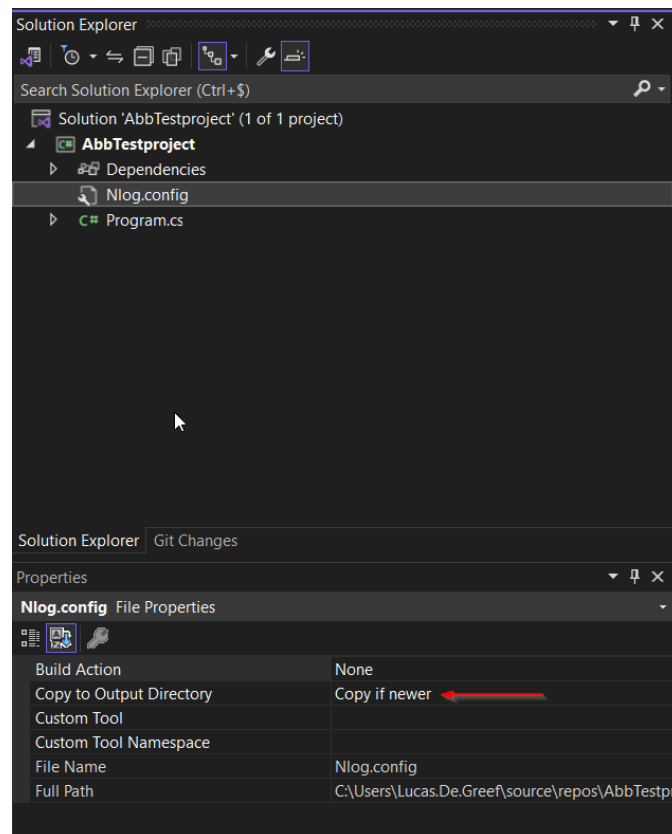


Figuur 9: Locatie Nuget packetmanager

Dit is de commando dat moet uitgevoerd worden in de packetmanager cli:

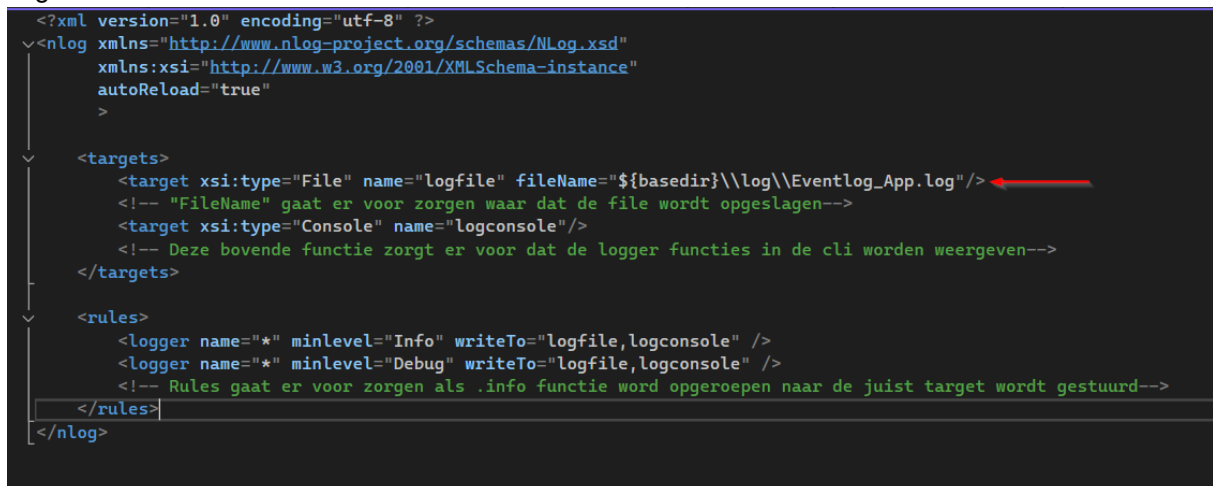
```
NuGet\Install-Package NLog -Version X.X.X
```

Als deze installatie is gebeurd, moet er een Nlog.config file aanwezig zijn met deze bepaalde voorwaarde:



Figuur 10: Nlog config file

Door dit te installeren gaat het programma weten wat in deze file wordt geconfigureerd, zodat het ook uitgevoerd kan worden.



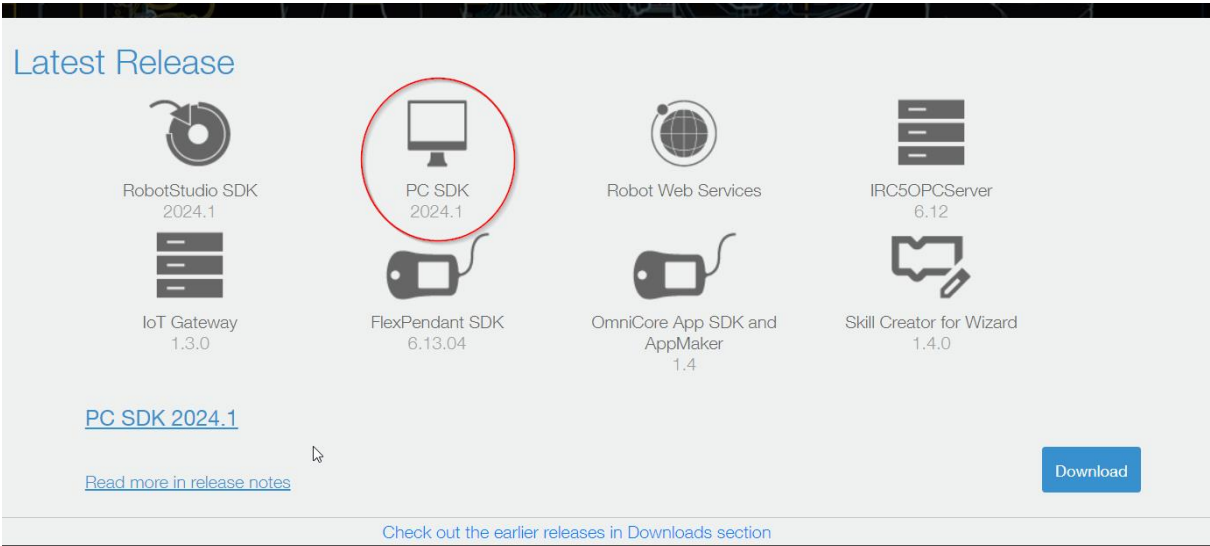
Figuur 11: Inhoud config file

In deze configuratie is het ingesteld in welke mappenstructuur de logfile zich moet bevinden. In het begin van de development fase was er eerst gekozen voor een relatief pad, maar nadat het project vorderde moest er een alternatief komen wegens geen voldoening van rechten op Windows server. Het element genaamd target zorgt ervoor naar welk doeleinde er gelogt moet worden. In dit voorbeeld is dit naar een file en een cli console.

5.2. ABB PCSDK-development setup

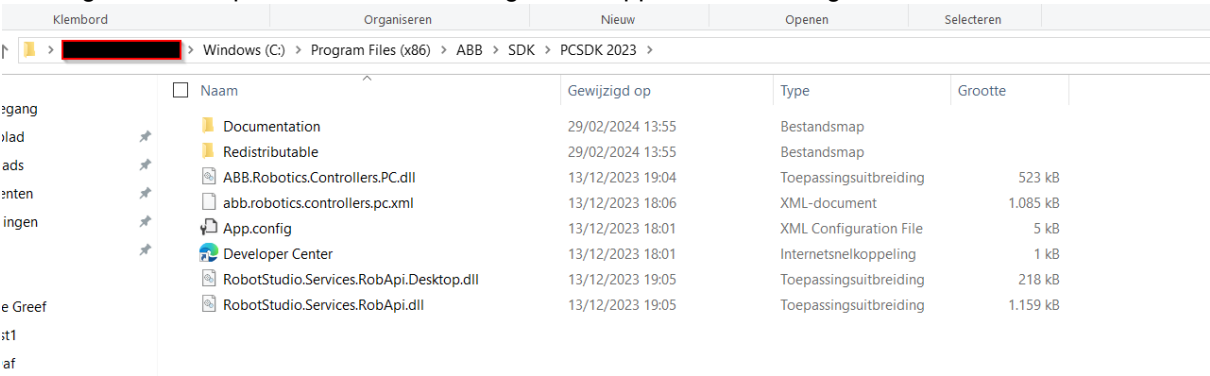
5.2.1. Installatie ABB PCSDK

De installatie van PCSDK is te vinden bij ABB [developercenter](#).



Figuur 12: ABB developercenter

Daarna moet dit geïnstalleerd worden volgens de installeer setup te volgen.
Als dit goed is verlopen dan moeten de volgende mappen en files aangemaakt worden:



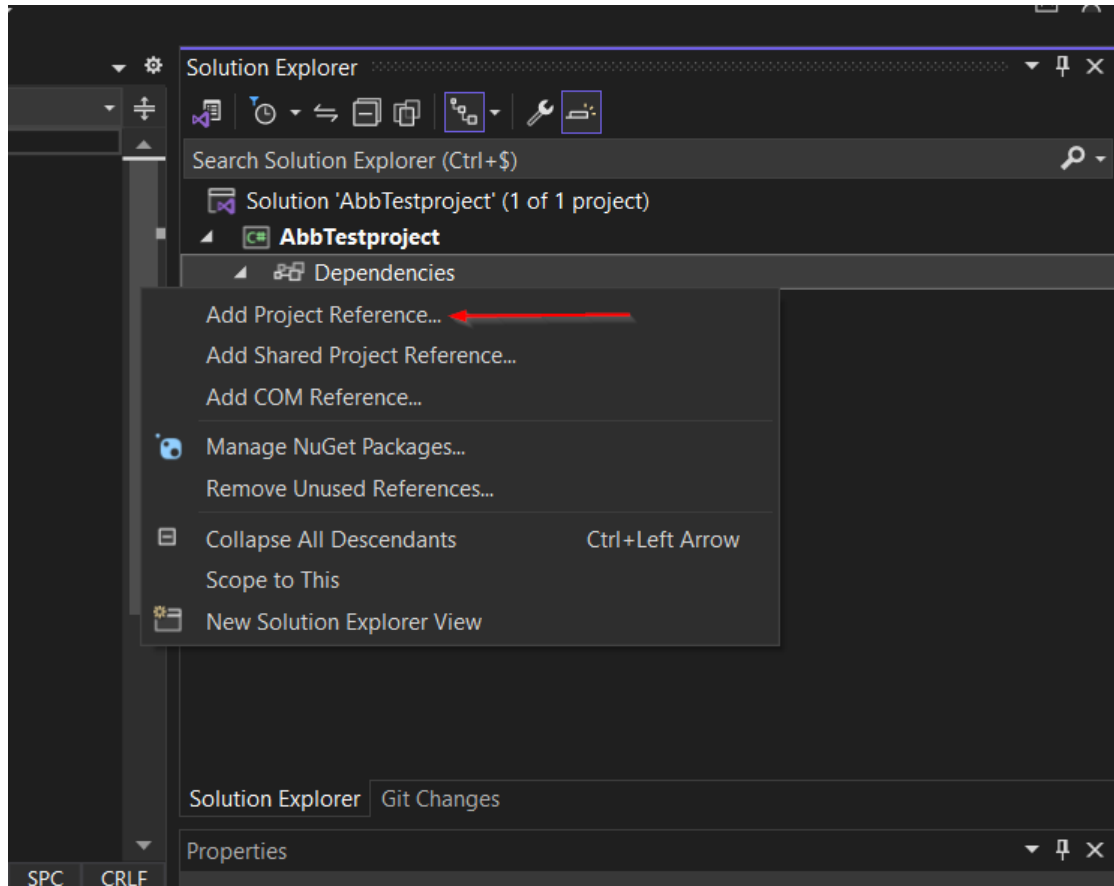
Figuur 13: Locatie ABB SDK op PC

Als deze mappen en files aanwezig zijn, is de PCSDK correct geïnstalleerd.

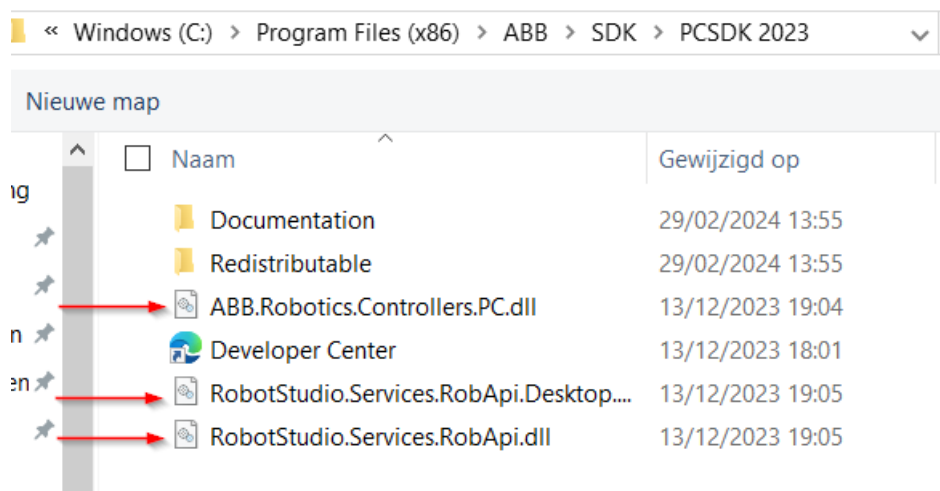
5.2.2. SDK refereren naar in C# project

In het begin van de applicatieontwikkeling moet er worden gerefereerd naar ABB PCSDK dat geïnstalleerd is op de pc.

Door op dependencies rechtermuisknop te drukken, wordt er een nieuw venster geopend. Door op het tabblad browse te klikken, kan er verwezen worden naar 3 .dll files die aanwezig zijn op de pc.



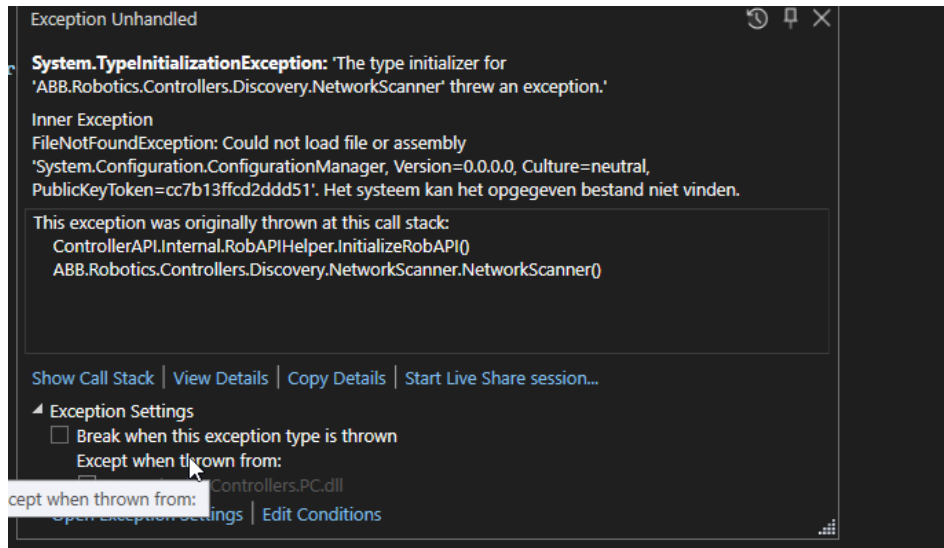
Figuur 14: dll files toevoegen in project



Figuur 15: Locatie 3 .dll files op pc

5.2.3. Error fouten die optreden

Er kunnen 2 soorten errors optreden. De eerste error zal een fout bevatten over de configuration manager, die niet is gevonden. Dit is simpel op te lossen om via Nuget packetmanager de volgende commando in te geven:



Figuur 16: Configuratie error van PCSDK en C#

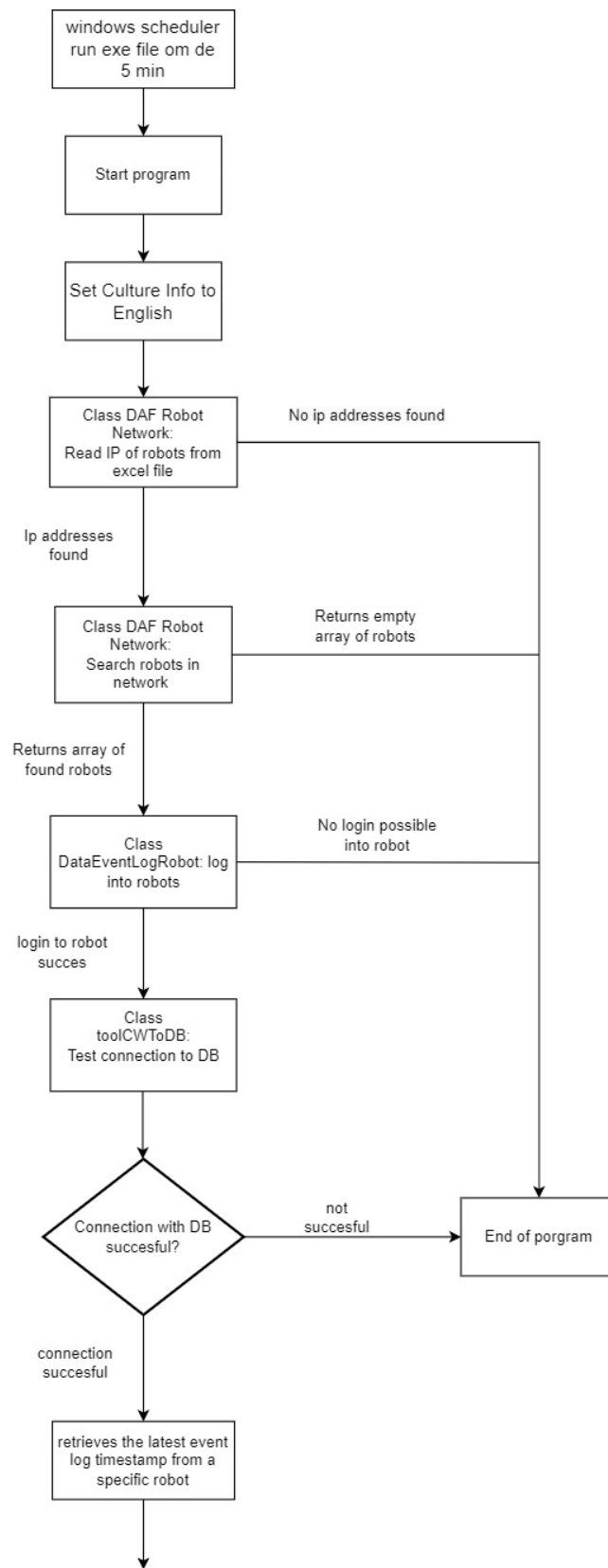
NuGet\Install-Package System.Configuration.ConfigurationManager -Version X.X.X

De 2^{de} error is in verband met de correctheid van performance counter.

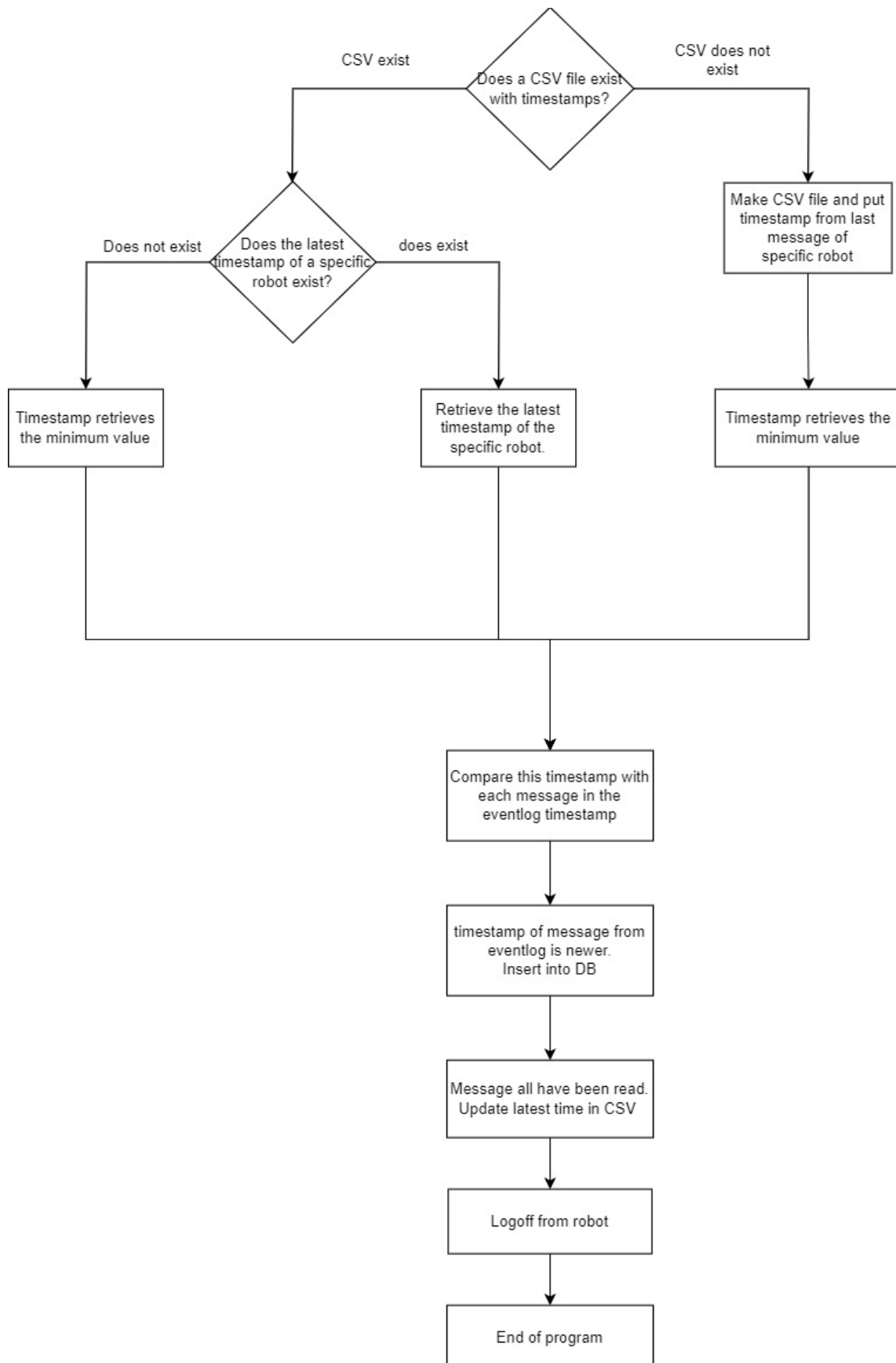
Dit is ook op te lossen door via Nuget packetmanager dit commando ingegeven:

NuGet\Install-Package System.Diagnostics.PerformanceCounter -Version X.X.X

6. Applicatie architectuur



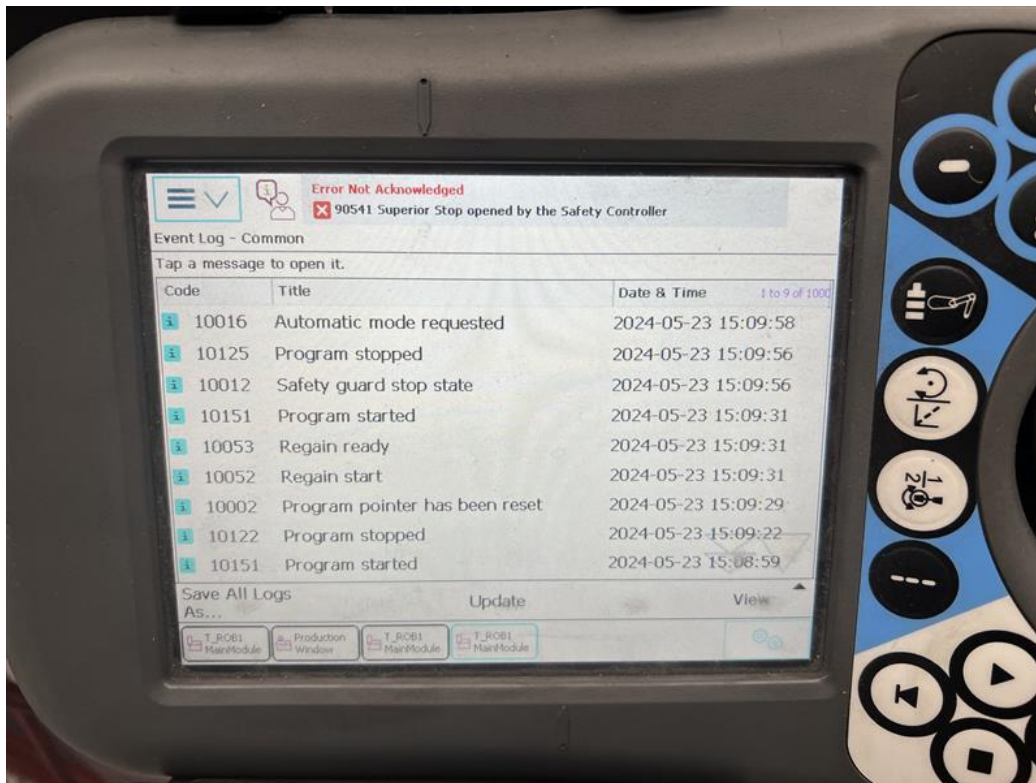
Vervolg gaar verder op de volgende pagina



7. ABB robot eventlog wat is dat?

7.1. Eventlog in ABB-robot

In het systeem van de ABB-robot is er een eventlog aanwezig. In deze eventlog worden de errors, informatie en waarschuwing berichten opgeslagen van het proces dat de robot uitvoert.



Figuur 17: Eventlog voorbeeld van robot

Er is een mappen structuur dat filtert welke errors en waarschuwingen worden opgeslagen. In deze mappen structuur is er ook een algemene map waar dat alle berichten ook worden opgeslagen. In deze map staat er een max aantal dat kunnen opgeslagen worden. Dit aantal is 250 berichten. De eerste stap in de applicatie is om de connectie met de robot mogelijk maken.

8. ABB Eventlog applicatie

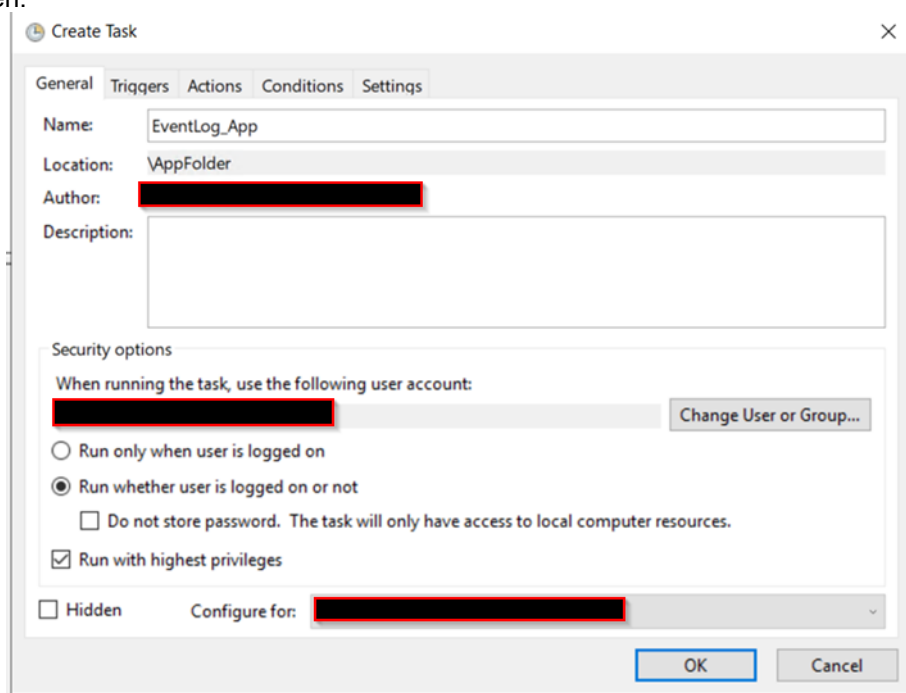
8.1. Wat is task scheduler en waarom toegepast

Met de Task scheduler kan er automatisch routinetaken uitgevoerd worden op een gekozen computer. De Task Scheduler doet dit door de criteria die er wordt gekozen (triggers genoemd) te controleren en vervolgens de taken uit te voeren wanneer er aan de criteria wordt voldaan. (Stevewhims, 2023)

De reden waarom er voor task scheduler is gekozen, is omdat deze functionaliteit beschikbaar is op de window Server OS. Dit maakt het efficiënter omdat het programma niet continue moet aanstaan. Deze eventlog applicatie moet 1 maal uitgevoerd worden en daarna gaat de task scheduler er voor zorgen dat het om de 5 min terug het programma gaat starten. Hiervoor moest er geen externe software gedownload worden.

8.2. Configureren van Windows Task

In het begin van het configuratie proces is er een aparte user aangemaakt op de windows server, met als doel dat het gehele applicatie gescheiden is van alle andere processen die deze windows server kan verstoren.



Figuur 18: Task shedular Algemeen tabel

Deze task is zodanig ingesteld dat de user niet moet ingelogd zijn om deze uit te voeren. Het is ook ingesteld met de hoogste rechten in het systeem. Dit is ervoor te zorgen dat het de andere bestanden kan lezen en/of aanpassen zonder de melding te krijgen dat er geen toegang is tot deze file.

New Trigger

Begin the task: On a schedule

Settings

☐ One time
 ☒ Daily
 ☐ Weekly
 ☐ Monthly

Start: 23/04/2024 13:42:00 ☐ Synchronize across time zones

Recur every: 1 days

Advanced settings

☐ Delay task for up to (random delay): 1 hour

☒ Repeat task every: 30 minutes for a duration of: 1 day

☐ Stop all running tasks at end of repetition duration

☒ Stop task if it runs longer than: 1 day

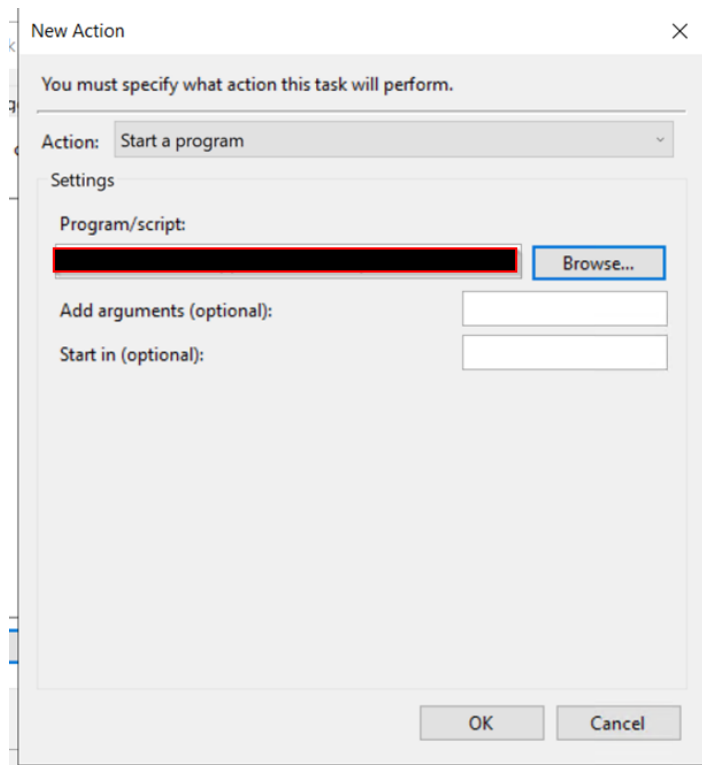
☐ Expire: 23/04/2025 13:35:28 ☐ Synchronize across time zones

☐ Enabled

OK Cancel

Figuur 19: Trigger instellen voor Task

In deze instellingen werd de tijd ingesteld wanneer deze task voor het eerst moest opstarten. Daarna werd aangeduid hoe vaak deze task uitgevoerd moest worden. Dit werd ingesteld om de 5 minuten, om er voor te zorgen dat er niet overtollig nieuwe data aanwezig is op de ABB robots . Voor duratie is er gekozen voor oneindig verloopt van de task, zodat het starten van de applicatie niet wordt onderbroken. Deze task is ook Enabled vanaf het begin.



Figuur 20: Keuze voor programma starten en Path naar het programma:

Deze actie is als doel ingesteld geweest om een programma te starten. De path moet beegegeven worden waar het programma zich bevindt.

8.3. CultureInfo veranderen

In de loop van het project konden de verschillende mappenstructuren waargenomen worden die aanwezig zijn in de ABB robot en in de C# applicatie. Deze werden in het Nederlands weergegeven, in de robot was dit in het Engels weergegeven. Dit was ook hetzelfde bij de berichten die aanwezig waren in de robot. De reden waarom is dat de C# applicatie dit kon aanpassen naar de gekozen taal van de gebruiker. Dit kon worden aangepast door de CultureInfo classe aan te passen in de applicatie zelf. Doormiddel van deze aanpassingen gaan de berichten die uitgelezen worden door de c# applicatie,

```

// Reference
public static void changelanguageEnglish()
{
    // Create a new CultureInfo object with the desired culture (English)
    CultureInfo englishCulture = new CultureInfo("en-US");

    // Set the current thread's culture to the desired culture
    Thread.CurrentThread.CurrentCulture = englishCulture;

    // Set the current thread's UI culture to the desired culture
    Thread.CurrentThread.CurrentUICulture = englishCulture;
}

```

Figuur 21: CultureInfo taalverandering

daadwerkelijk overeen komen met de berichten die aanwezig zijn in de ABB-robot

8.4. Robot controllers zoeken & ip adressen uit Excel halen

In de documentatie van ABB PCSDK staat er beschreven hoe je naar robotcontrollers zoekt in het netwerk.

Class NetworkScanner

This class is used to scan the network for controllers and changes in the infrastructure

(Class NetworkScanner, n.d.)

In deze class kan er gebruik gemaakt worden van een scan functie. Als deze functie wordt toegepast, gaat het een exe file oproepen genaamd ABB Robot Communication Runtime. Deze scan functie kan op 2 manieren opgevangen worden. Er kan enerzijds de class controllers opgehaald worden in het geheugen van het proces, of anderzijds door een andere functie Getcontrollers op te roepen.

Door middel van deze classe kan er worden weergegeven welke robots gevonden zijn in het netwerk.

Class ControllerInfo

This class contains the "simple" information that is available about a controller without connecting to it

(Class ControllerInfo, n.d.).

Er werd in verder verloopt van de eventlog applicatie nog 2 veranderingen gemaakt, zodat het zoeken van robots op deze manier gaat verlopen.

In de map waar het project zich bevindt moet er een excel file aanwezig zijn met de nodig IP-adressen van de robot. De reden is dat niet alle poorten open zijn in firewall hun netwerk en deze afblokken. Maar door deze IP-adressen toe te voegen aan de netwerkscanner classe kunnen deze controllers wel gevonden worden in de applicatie.

Als 2^{de} oplossing is er een filtering op naam van de robots zodat deze niet leeg is of null.

Er was tijdens de testfase een robot controller aanwezig die geen naam had. Dit was een virtuele controller die aanwezig was in het netwerk. Ook in de functie Getcontroller wordt het meegegeven dat het een vereiste van echte controllers moet zijn.

Als laatste stap worden alle gevonden robots in een array gestoken.

8.5. Mappen structuur uit robot halen

Nadat alle robot controllers in het netwerk zijn gevonden gaat de volgende functie deze array met robots een voor een connectie proberen te maken cyclisch gewijs.

Voor een correcte connectie te maken met de robot is er gebruik gemaakt van de classe Controller.

In de documentatie van deze classe staat de volgende uitgelegd:

Class Controller

The main entry point for any operation against the controller. (Class Controller, n.d.)

Als er ingelogd moet worden in deze controller, moet er worden gebruik gemaakt van de methode Connect (Guid, ConnectionType, Boolean)

Creates a Controller object with a connection to a controller. (Class Controller, n.d.)

Deze functie verwacht een systeem id van de robotcontroller dat uniek is, alsook een connectie. Dit connectie type heeft als doel een applicatie dat helemaal alleen werkt. Je kan ook een applicatie maken dat kan worden gebruikt als extensie van robot studio.

Dit is een losstaande applicatie zijn

Daarna moet er een functie opgeroepen worden om te kunnen inloggen in de robot.

Logon(UserInfo)

Logon as the specified user (Class Controller, n.d.)

Er kan een keuze gemaakt worden uit 2 User in deze functie. DefaultUser of de DefaultLocalUser.

Deze is enkel mogelijk voor virtuele controllers. Bij elke geslaagde connectie met de robot wordt deze info wegeschreven naar een logfile.

De volgende stap in dit proces is om het Eventlog domain uit de robotcontroller te halen. Dit is mogelijk door eventlog domain op te roepen in de Controller classe. Door dit is er een property acces van de eventlog. Dit wordt dan in een variable gestoken die de classe Eventlog gebruikt.

Door dat er gebruik is gemaakt van deze eventlog klasse is er een filtering mogelijk bij mappen structuren. Er is de keuze gemaakt voor enkel de data berichten die binnenkomen in de algemene map te laten uitlezen.

8.6. Connectie met Database

Nadat de mappen structuur uit bepaalde robots zijn gehaald, moet er gecontroleerd worden of dat de connectie met de database succesvol is. Zo lang dat deze connectie niet succesvol is, gaat er ook geen data uitgelezen worden van de robot. Er werd gebruik gemaakt van een .config file waar dat de connectie string van de database in is verwerkt. Dit is uit veiligheidsredenen zodat het wachtwoord van de database niet in plain tekst in de code zichtbaar is. Wanneer deze connectie gelukt was is er een table aangemaakt in een database waar dat alle robot data aanwezig gaat zijn.

De query dat werd gebruikt is de volgende:

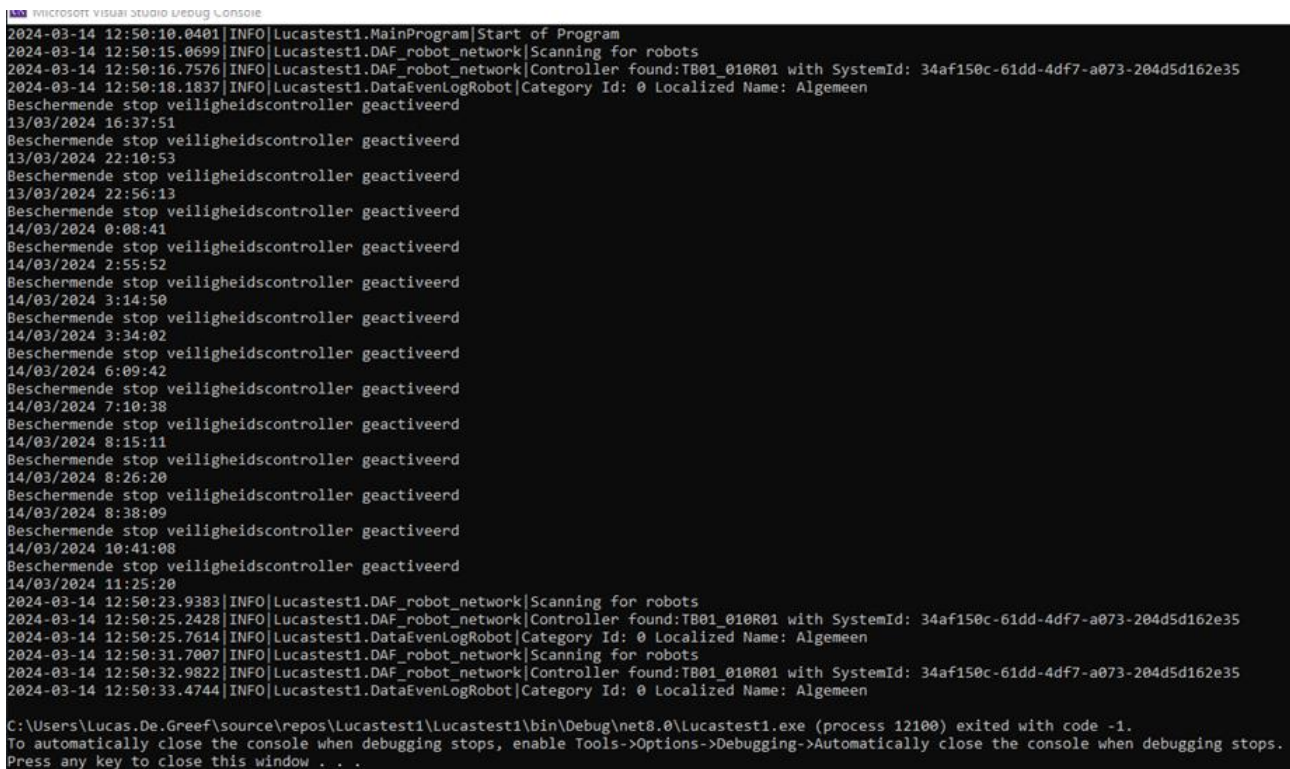
```
CREATE TABLE [...]. [...].[...] (
  [ID] INT PRIMARY KEY, -- Aannemende dat ID een unieke identifier is
  [DateAndTime] DATETIME,
  [CelRobotName] NVARCHAR(255),
  [CelName] NVARCHAR(255),
  [StationName] NVARCHAR(255),
  [Consequences] NVARCHAR(MAX),
  [Cause] NVARCHAR(MAX),
  [Actions] NVARCHAR(MAX),
  [CauseTitle] NVARCHAR(255),
  [TypeMessage] NVARCHAR(255),
  [Descriptions] NVARCHAR(MAX),
  [Date] AS CAST([DateAndTime] AS DATE) -- Computed column
);
```

8.7. Eventlog data uitlezen

Nadat de connectie testen is gelukt en filtratie van mappen structuur, gaat het uitlezen van berichten van de eventlog de volgende stap zijn.

Doordat er al een filtering is gebeurd op het kiezen welke map uit de robot wordt gehaald, is er ook een methode die hierin beschikbaar is genaamd Messages. Door deze toe te passen kan er een filtering gebeuren op de message die afkomstig zijn van de ABB-robot. Wanneer deze bericht voor het eerst werden uitgelezen, waren de nieuwste event als eerste en het oudste event als laatste.

Voor dit probleem op te lossen is er gebruik gemaakt van een ingebouwde functie Orderby. Deze functie zorgt ervoor dat er in de berichten kon gefilterd worden op de timestamp van elk individueel bericht.



```
Microsoft Visual Studio Debug Console
2024-03-14 12:50:10.0401|INFO|Lucastest1.MainProgram|Start of Program
2024-03-14 12:50:15.0699|INFO|Lucastest1.DAF_robot_network|Scanning for robots
2024-03-14 12:50:16.7576|INFO|Lucastest1.DAF_robot_network|Controller found:TB01_010R01 with SystemId: 34af150c-61dd-4df7-a073-204d5d162e35
2024-03-14 12:50:18.1837|INFO|Lucastest1.DataEvenLogRobot|Category Id: 0 Localized Name: Algemeen
Beschermende stop veiligheidscontroller geactiveerd
13/03/2024 16:37:51
Beschermende stop veiligheidscontroller geactiveerd
13/03/2024 22:10:53
Beschermende stop veiligheidscontroller geactiveerd
13/03/2024 22:56:13
Beschermende stop veiligheidscontroller geactiveerd
14/03/2024 0:08:41
Beschermende stop veiligheidscontroller geactiveerd
14/03/2024 2:55:52
Beschermende stop veiligheidscontroller geactiveerd
14/03/2024 3:14:50
Beschermende stop veiligheidscontroller geactiveerd
14/03/2024 3:34:02
Beschermende stop veiligheidscontroller geactiveerd
14/03/2024 6:09:42
Beschermende stop veiligheidscontroller geactiveerd
14/03/2024 7:10:38
Beschermende stop veiligheidscontroller geactiveerd
14/03/2024 8:15:11
Beschermende stop veiligheidscontroller geactiveerd
14/03/2024 8:26:20
Beschermende stop veiligheidscontroller geactiveerd
14/03/2024 8:38:09
Beschermende stop veiligheidscontroller geactiveerd
14/03/2024 10:41:08
Beschermende stop veiligheidscontroller geactiveerd
14/03/2024 11:25:20
2024-03-14 12:50:23.9383|INFO|Lucastest1.DAF_robot_network|Scanning for robots
2024-03-14 12:50:25.2428|INFO|Lucastest1.DAF_robot_network|Controller found:TB01_010R01 with SystemId: 34af150c-61dd-4df7-a073-204d5d162e35
2024-03-14 12:50:25.7614|INFO|Lucastest1.DataEvenLogRobot|Category Id: 0 Localized Name: Algemeen
2024-03-14 12:50:31.7007|INFO|Lucastest1.DAF_robot_network|Scanning for robots
2024-03-14 12:50:32.9822|INFO|Lucastest1.DAF_robot_network|Controller found:TB01_010R01 with SystemId: 34af150c-61dd-4df7-a073-204d5d162e35
2024-03-14 12:50:33.4744|INFO|Lucastest1.DataEvenLogRobot|Category Id: 0 Localized Name: Algemeen
C:\Users\Lucas.De.Greef\source\repos\Lucastest1\Lucastest1\bin\Debug\net8.0\Lucastest1.exe (process 12100) exited with code -1.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

Figuur 22: Tussen resultaat Data uitlezen

8.8. Data capteren in database

Alle berichten zijn op juiste manier gesorteerd zodat er een vergelijking mogelijk is tussen het vorige bericht en het volgende bericht. De volgende stap is dan deze data correct in de database te capteren. Er werd een functie gemaakt dat het de naam van de robot ook direct meeneemt. Er is ook een ontleding van elk bericht:

Wat soort bericht is het, welke type bericht is het (Informatie, Waarschuwing, error), wat was de reden dat dit bericht is gelogt (Cause), welke actie moet ondernomen worden bij dit event (Actions), datum en tijd wanneer het event is opgetreden (DateAndTime). Door deze ontleding van elke event is er meer duidelijk wat elke bericht juist inhoudt.

Voor elk element van elk event is er een insert query aangemaakt en deze werd opgeslagen in app.config. Na de feedback moment zijn de volgende tips gegeven.

Zet de query in een sql bestand. Als de query wordt opgeroepen, hernoem deze hetzelfde. Wanneer er veranderingen gebeuren in de query, moet de code niet aangepast worden.

Hier werd ook een voorbeeld code gegeven over hoe dit toegepast moet worden. Aan de hand van deze feedback zijn er aanpassingen gebeurd in de code.

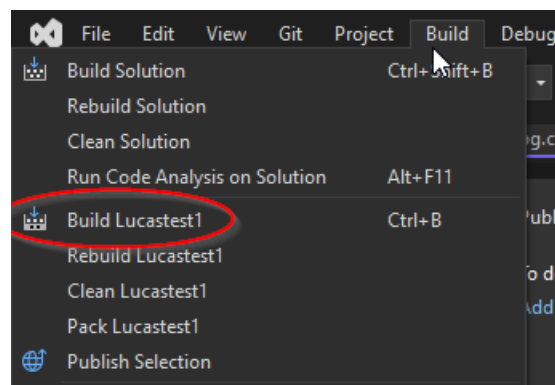
8.9. CSV Aanmaken en schrijven

Omdat er gekozen is voor een Task scheduler was het probleem dat het laatste recentste event uit de ABB-robot niet werd bijgehouden. De oplossing voor dit probleem was gebruik maken van een tekstfile die eerst de datum en tijd bijhield. Het werd snel duidelijk dat ook de naam van elke robot apart toegevoegd moest worden. Doordat de robot naam er ook moest bijstaan in deze tekstfile, is er gekozen voor een csv file, omdat de robotnaam, de datum en tijd beter samengeroepen werden. Er werd ook in de code een check gedaan of deze file aanwezig is in het pad waar dat deze applicatie staat. Zo niet, wordt deze csv fil ook aangemaakt en gaat het telkens wanneer er een nieuwe robot, die geen datum en tijd data heeft, toegevoegd worden. Als deze datum en tijd niet aanwezig zijn, of de csv is pas aangemaakt en daardoor leeg, gaat bij het vergelijken van het eerste event dat gelezen wordt alle berichten toevoegen in de database. Als het laatste event gelezen is, wordt de datum en tijd van dit event in de csv opgeslagen alsook de naam van de robot waar dit event afkomstig van was.

8.10. Deploy applicatie

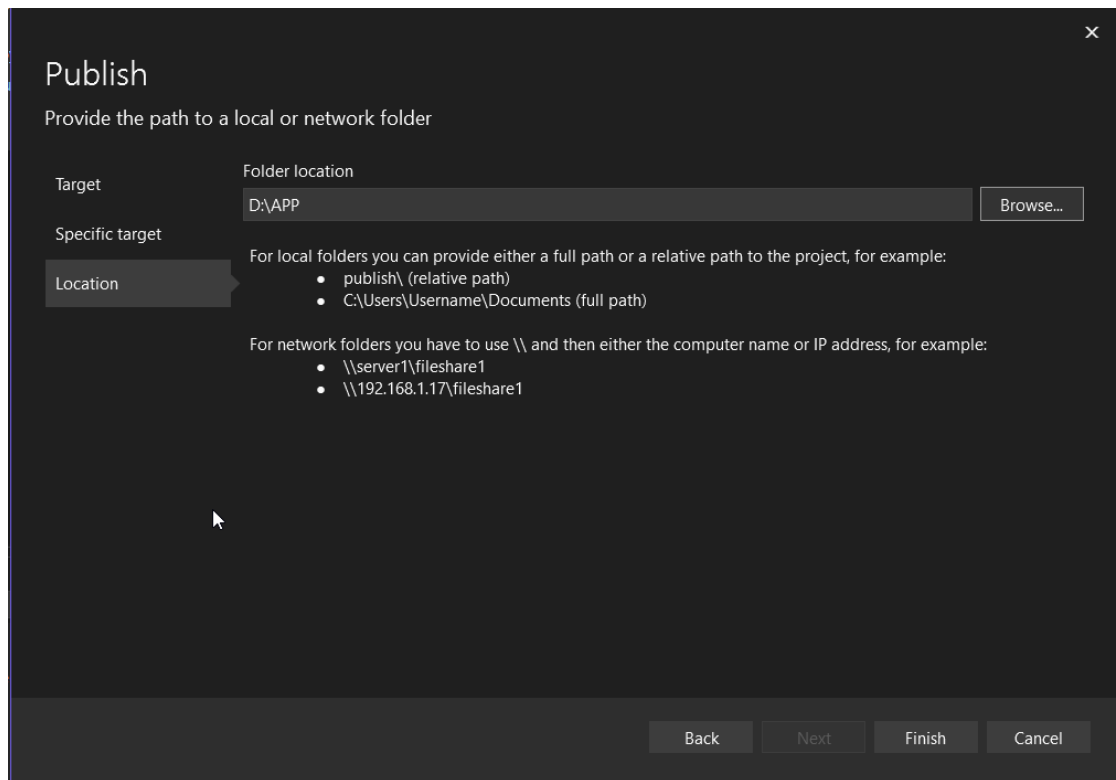
De applicatie heeft zijn eerst afgewerkte versie. De bedoeling is om deze applicatie op de windows server te laten uitvoeren. Er werd gebruik gemaakt van publish selection in visual studio zelf.

De eerste voorwaarde is dat deze applicatie geen build errors bevat.

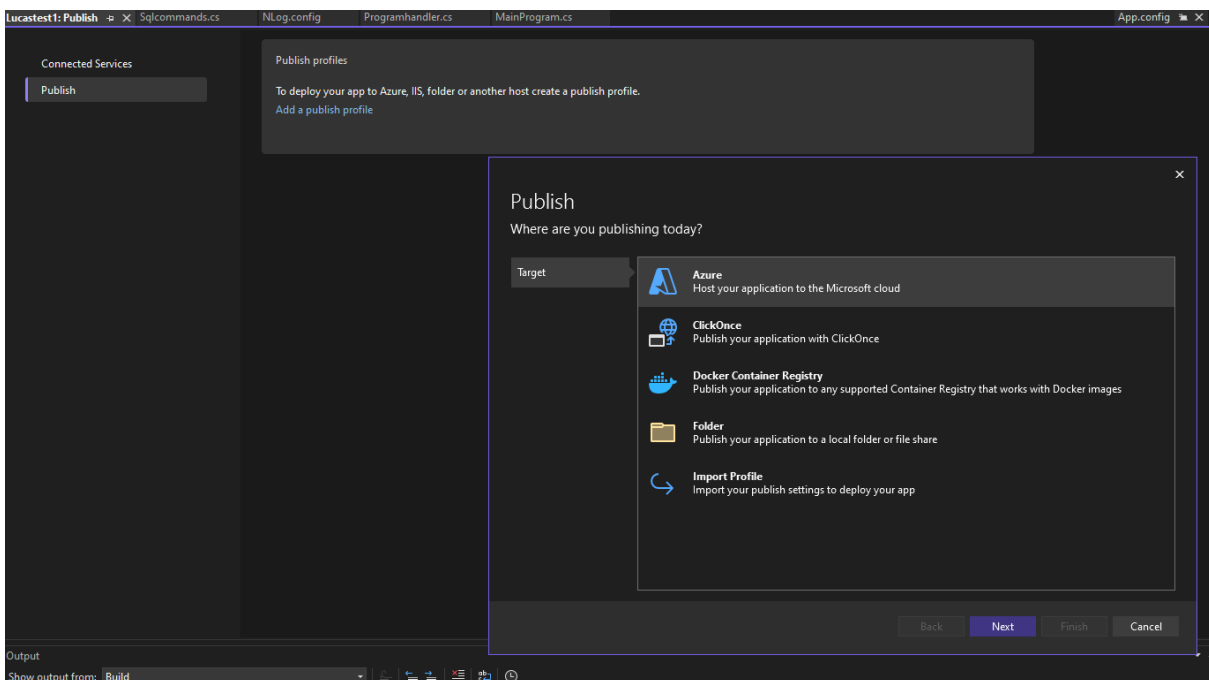


Figuur 23: build applicatie

Daarna is de volgende stap om naar publish selection te gaan. Er is gekozen om een afgewerkte applicatie in een map op te slaan.



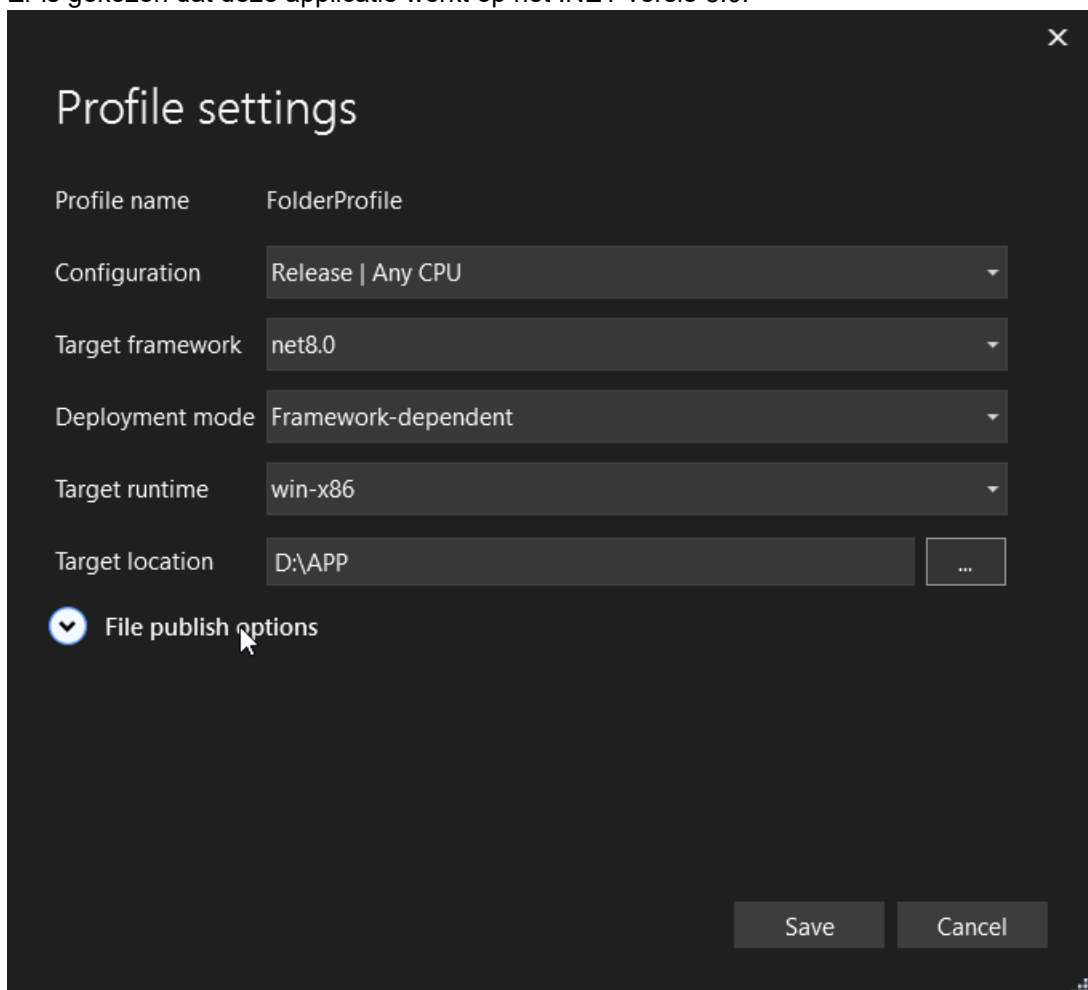
Figuur 25: path locatie voor het opslaan van applicatie



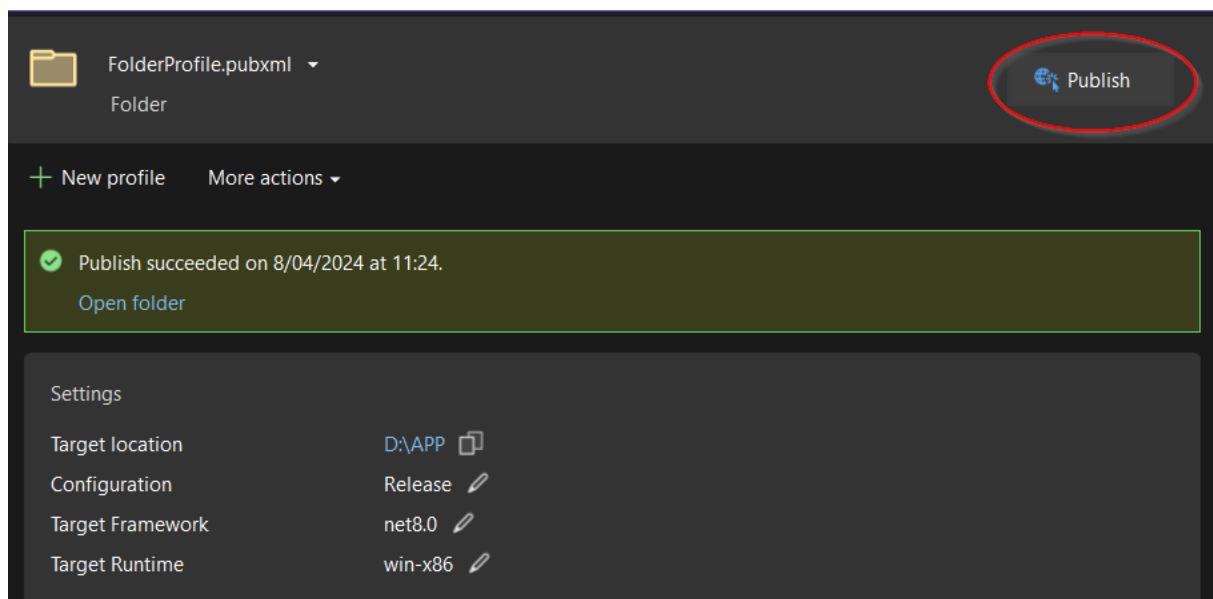
Figuur 24: Keuze hoe applicatie wordt opgeslagen

Daarna werd er gekozen op welk path dit opgeslagen moest worden. Dit werd op een usb geplaatst zodat dit later op de windows server geplaatst kan worden.

In dit optie venster is het volgende ingesteld waar dat de applicatie afhankelijk moet zijn.
Er is gekozen dat deze applicatie werkt op het .NET versie 8.0.



Figuur 27: Applicatie voorwaarde instellingen



Figuur 26: Publish applicatie

9. Nieuw project Rapid Data loggen

9.1. Wat is rapid data?

```
37      ! Definition of process gun data
38  RECORD gundata
39      string gun_name;
40      num weld_counter;
41      num max_nof_welds;
42      num curr_tip_wear;
43      num max_tip_wear;
44      num curr_wear_fix;
45      num curr_wear_mov;
46      num release_dist;
47  ENDRECORD
```

Figuur 28: Rapid data voorbeeld gundata

De programmatie van de ABB-robot en welke handeling en bewegingen moeten uitgevoerd worden, wordt gedaan in robotstudio. Dit is een code dat gebruikt wordt voor de robot te programmeren. In deze code kunnen belangrijke variabelen aanwezig zijn die bepaalde handelingen van een robot bijhoudt op het systeem. Dit kan relevant zijn en daarom noemt men deze data rapid data.

9.2. Lastang robot

Doordat er relevante rapid data aanwezig zijn in de code van de ABB robots, was de eerste stap welke data effectief kan relevant zijn. De eerste vraag was de coördinate, waar bepaalde laspunten opgelast worden. Deze variabelen zijn constant in de code, dus er was zodanig onduidelijkheid wanneer en door wie deze werden aangepast. Door onderzoek te doen had de SDK geen ingebouwde mogelijkheid om veranderingen waar te nemen op constante variabelen. Door dat dit niet mogelijk was is er uitgekeken naar andere opties die ook belangrijke data hadden. Er was sprake van 2 variabelen namen: spotdata en gundata.

Deze 2 variabelen zijn niet constant. In het programma zijn dit PERS variabelen.

Een Persistent variabele is hetzelfde als een gewone variabele, maar met één belangrijk verschil. Een persistente variabele onthoudt de laatste waarde die eraan is toegewezen, zelfs als het programma wordt gestopt en opnieuw wordt gestart. (Rapid Data, n.d.)

De gebruiker moet weten dat RAPID-gegevens zijn gewijzigd

Om op de hoogte te worden gesteld dat RAPID-gegevens zijn gewijzigd, moet er een abonnement toegevoegd worden aan de ValueChanged-gebeurtenis van de RapidData-instantie. Dat dit echter alleen werkt voor Persistent RAPID-gegevens. (Working With RAPID Data, n.d.)

Deze vereiste moest voldaan zijn om veranderingen van de variabele op te merken tijdens het proces. Bij Gundata variable waren enkele parameters die interessant waren zoals current tip wear. Door deze variabele uit te lezen kon er een lastip van de lastang de slijtage uit het systeem halen.



Figuur 29: Lastang ABB-robot



Figuur 30: Lasttip van lastang

9.3. Test in simulatie

Voor test omgeving is er eerst gebruikt gemaakt van virtuele robot/controller. Dit is mogelijk in robotstudio als een backup is genomen van het systeem dat op de robot staat. Als deze is toegevoegd

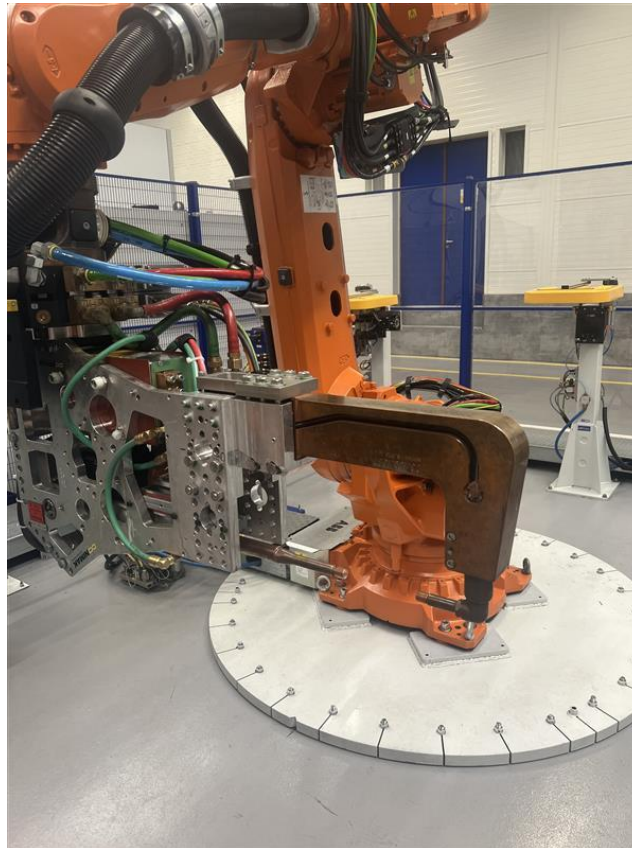


Figuur 31: Virtuele controller

in robotstudio kan je de variabele wijzigen die in het virtueel systeem aanwezig, zoals dit ook zou kunnen bij een echte systeem.

9.4. Test in testcel

De code is gebaseerd op de virtuele omgeving. De volgende stap is naar een echte robot over te stappen. In deze testcel is er een testlas programma aanwezig. Als deze wordt uitgevoerd gaat de robot deze variabele aanpassen. Als bij elke aanpassing een verandering gebeurt gaat de nieuwe applicatie deze verandering opvangen.



Figuur 32: Testcels

9.5. Test in productie

Als het programma op een virtuele en fysieke controller getest is geweest, was de volgende stap dit op meerde niveaus op productionniveau te testen. Dit gaf een positief resultaat. De verandering van de variabele die afkomstig waren van verschillende robots werden ook te zien in de applicatie. Dit was door de tijdsdruk ook een proof of concept project, maar het uitlezen van meerdere robots was mogelijk. Alsook meerdere variabele was mogelijk. Deze applicatie werd ook getest op de windows

server, waar ook de vorige applicatie momenteel uitgevoerd staat. Het programma werd ook uitgevoerd maar bleef na enkele seconde vasthangen.

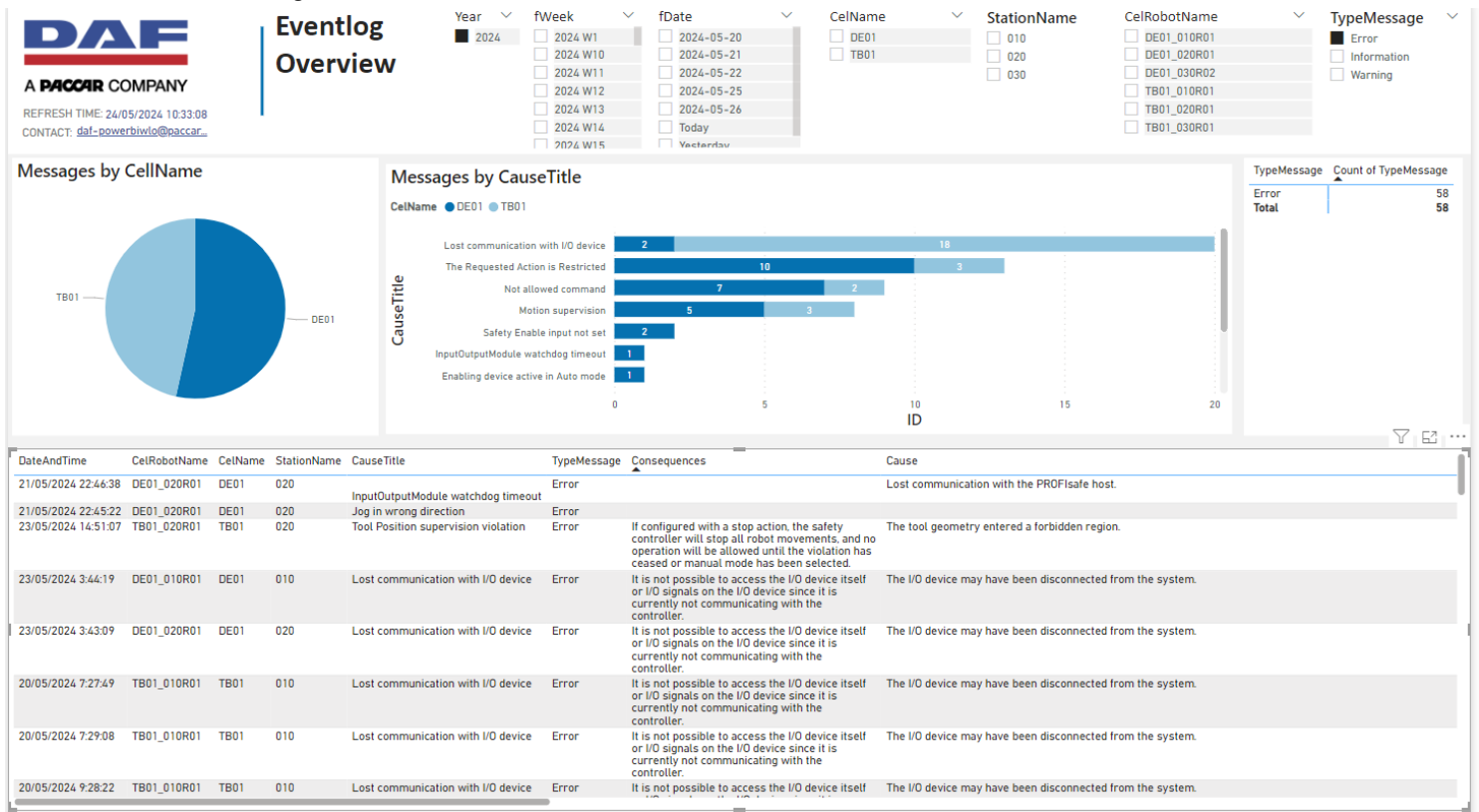
```
C:\Users\LucasDeGreef\source\repos\DAF_Rapid_Data\DAF_Rapid_Data\bin\Debug\net8.0\DAF_Rapid_Data.exe
2024-05-23 14:50:03.6785 INFO|DAF_Rapid_Data.Taskcreator.LogInToRobots|Start task with robotname: [redacted]
2024-05-23 14:50:04.9659 INFO|DAF_Rapid_Data.Taskcreator.LogInToRobots|logged in to robot [redacted]
2024-05-23 14:50:05.0917 INFO|DAF_Rapid_Data.Taskcreator.LogInToRobots|logged in to robot [redacted]
2024-05-23 14:50:05.2189 INFO|DAF_Rapid_Data.Taskcreator.LogInToRobots|logged in to robot [redacted]
2024-05-23 14:50:05.3530 INFO|DAF_Rapid_Data.Taskcreator.LogInToRobots|Login to robot [redacted]
2024-05-23 14:50:05.3530 INFO|DAF_Rapid_Data.Taskcreator.LogInToRobots|logged in to robot [redacted]
2024-05-23 14:50:05.3530 INFO|DAF_Rapid_Data.Taskcreator.LogInToRobots|Login to robot [redacted]
2024-05-23 14:50:05.4193 INFO|DAF_Rapid_Data.Taskcreator.LogInToRobots|No SMUSER module found, Log off to robot [redacted]
2024-05-23 14:50:05.4502 INFO|DAF_Rapid_Data.Taskcreator.LogInToRobots|Login to robot [redacted]
2024-05-23 14:50:05.4892 INFO|DAF_Rapid_Data.Taskcreator.LogInToRobots|logged in to robot [redacted]
2024-05-23 14:50:05.5859 INFO|DAF_Rapid_Data.Taskcreator.LogInToRobots|No SMUSER module found, Log off to robot [redacted]
2024-05-23 14:50:06.9846 INFO|DAF_Rapid_Data.Taskcreator.LogInToRobots|RapidData form [redacted] curr_spotdata has changed to [[255,2550013,1480.78,0,0]]
2024-05-23 14:50:06.9846 INFO|DAF_Rapid_Data.Taskcreator.LogInToRobots|RapidData form [redacted] curr_gundata has changed to [[["SERVO_G1",0,1000,0.18,10,0.09,0.09,5]]]
2024-05-23 14:50:06.9846 INFO|DAF_Rapid_Data.Taskcreator.LogInToRobots|RapidData form [redacted] curr_gundata has changed to [[["SERVO_G1",120,1000,3.38,10,1.66,1.72,5]]]
2024-05-23 14:50:06.9846 DEBUG|DAF_Rapid_Data.Taskcreator.LogInToRobots|Time of event 23/05/2024 14:50:05
2024-05-23 14:50:06.9846 DEBUG|DAF_Rapid_Data.Taskcreator.LogInToRobots|Time of event 23/05/2024 14:50:05
2024-05-23 14:50:06.9846 INFO|DAF_Rapid_Data.Taskcreator.LogInToRobots|RapidData form [redacted] curr_spotdata has changed to [[8,8610014,2780.24,1.9,0.7]]
2024-05-23 14:50:06.9846 INFO|DAF_Rapid_Data.Taskcreator.LogInToRobots|RapidData form [redacted] curr_spotdata has changed to [[255,2550013,1480.78,0,0]]
2024-05-23 14:50:06.9926 INFO|DAF_Rapid_Data.Taskcreator.LogInToRobots|RapidData form [redacted] curr_gundata has changed to [[["SERVO_G1",0,1000,2.29,10,1.11,1.18,5]]]
2024-05-23 14:50:06.9926 DEBUG|DAF_Rapid_Data.Taskcreator.LogInToRobots|Time of event 23/05/2024 14:50:05
2024-05-23 14:50:06.9926 DEBUG|DAF_Rapid_Data.Taskcreator.LogInToRobots|Time of event 23/05/2024 14:50:05
2024-05-23 14:50:06.9926 DEBUG|DAF_Rapid_Data.Taskcreator.LogInToRobots|Time of event 23/05/2024 14:50:05
2024-05-23 14:50:07.3340 INFO|DAF_Rapid_Data.Taskcreator.LogInToRobots|logged in to robot [redacted]
2024-05-23 14:50:07.4059 INFO|DAF_Rapid_Data.Taskcreator.LogInToRobots|No SMUSER module found, Log off to robot [redacted]
2024-05-23 14:50:07.4744 INFO|DAF_Rapid_Data.Taskcreator.LogInToRobots|logged in to robot [redacted]
2024-05-23 14:50:07.7415 INFO|DAF_Rapid_Data.Taskcreator.LogInToRobots|Login to robot [redacted]
2024-05-23 14:50:08.0752 INFO|DAF_Rapid_Data.Taskcreator.LogInToRobots|RapidData form [redacted] curr_gundata has changed to [[["SERVO_G1",226,1000,2.78,10,1.39,1.39,5]]]
2024-05-23 14:50:08.0752 INFO|DAF_Rapid_Data.Taskcreator.LogInToRobots|RapidData form [redacted] curr_spotdata has changed to [[8,8560004,2780.24,1.8,0.8]]
2024-05-23 14:50:08.0752 DEBUG|DAF_Rapid_Data.Taskcreator.LogInToRobots|Time of event 23/05/2024 14:50:07
2024-05-23 14:50:08.0752 DEBUG|DAF_Rapid_Data.Taskcreator.LogInToRobots|Time of event 23/05/2024 14:50:07
Queue heeft 8 items, begint met uitlezen...
2024-05-23 14:50:08.8963 INFO|DAF_Rapid_Data.Write_Read_DB.DBTools|Database connection successful.
Data insertion successful
Queue uitlezen result: [redacted] curr_gundata,[[["SERVO_G1",120,1000,3.38,10,1.66,1.72,5]],23/05/2024 14:50:05
Data insertion successful
Queue uitlezen result: [redacted] curr_spotdata,[[255,2550013,1480.78,0,0]],23/05/2024 14:50:05
Data insertion successful
Queue uitlezen result: [redacted] curr_spotdata,[[255,2550013,1480.78,0,0]],23/05/2024 14:50:05
Data insertion successful
Queue uitlezen result: [redacted] curr_spotdata,[[8,8610014,2780.24,1.9,0.7]],23/05/2024 14:50:05
Data insertion successful
Queue uitlezen result: [redacted] curr_gundata,[[["SERVO_G1",0,1000,0.18,10,0.09,0.09,5]],23/05/2024 14:50:05
Data insertion successful
Queue uitlezen result: [redacted] curr_gundata,[[["SERVO_G1",0,1000,2.29,10,1.11,1.18,5]],23/05/2024 14:50:05
Data insertion successful
Queue uitlezen result: [redacted] curr_gundata,[[["SERVO_G1",226,1000,2.78,10,1.39,1.39,5]],23/05/2024 14:50:07
Data insertion successful
Queue uitlezen result: [redacted] curr_spotdata,[[8,8560004,2780.24,1.8,0.8]],23/05/2024 14:50:07
Pause (Ctrl+P)
```

Figuur 33: Resultaat rapid data uitlezen op basis van verandering

10. Toegevoegde waarde bachelorproef

10.1. Eventlog rapport

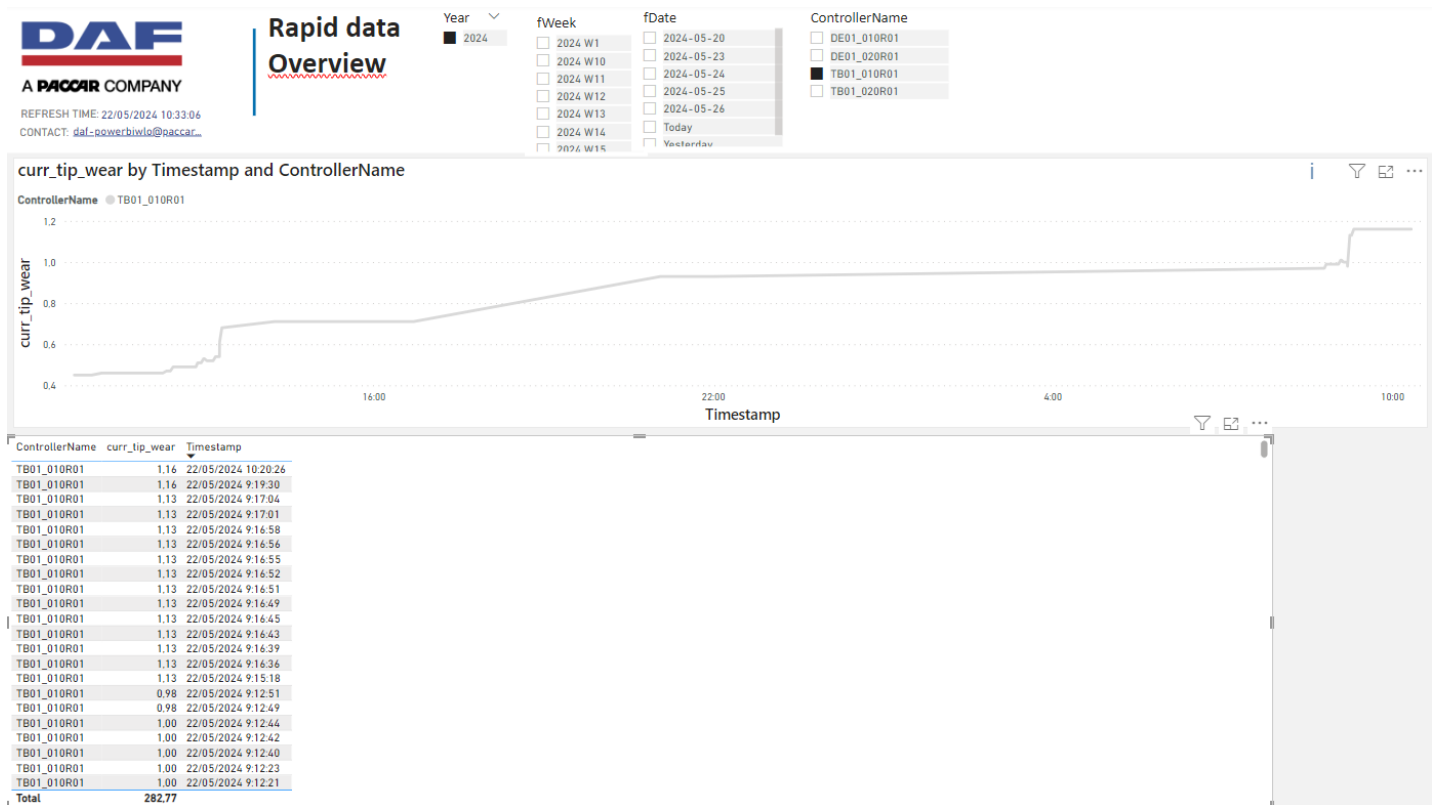
Dit rapport heeft meer inzicht gegeven welke berichten allemaal aanwezig waren in de verschillende robots alsook de filtering kon gedaan worden de verschillende cellen waar dat robots aanwezig waren. Er kan ook een filtering gebeuren over wat voor soort type berichten er zijn vb welke zijn de meeste errors dat zijn voorgekomen, van welke periode tot welke periode? Dit kan door dit rapport allemaal in kaart worden gebracht.



Figuur 34: Eventlog rapport

10.2. Tipwear rapport

Dit rapport kan de bouwstenen geven over welke mogelijkheden er nog allemaal zijn met ABB PCSDK. Hier wordt een historiek weergegeven van de tipwear toestand in de loop de tijd. Dit kan een interessant beeld geven op hoeveel dat de las teller staat. Dit kan er voor zorgen dat een hetzelfde las tip langer kan meegaan in het proces, of minder afhankelijk, van welke materiaal het last alsook hoe



Figuur 35: Rapid data rapport

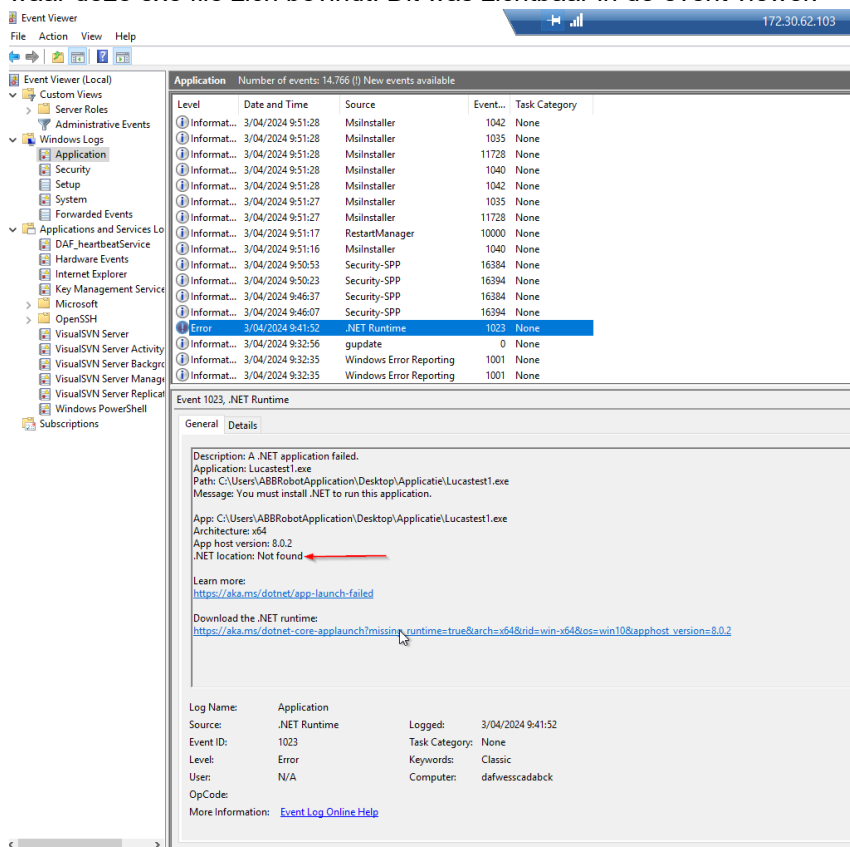
dik de plaat is.

11. Appendix

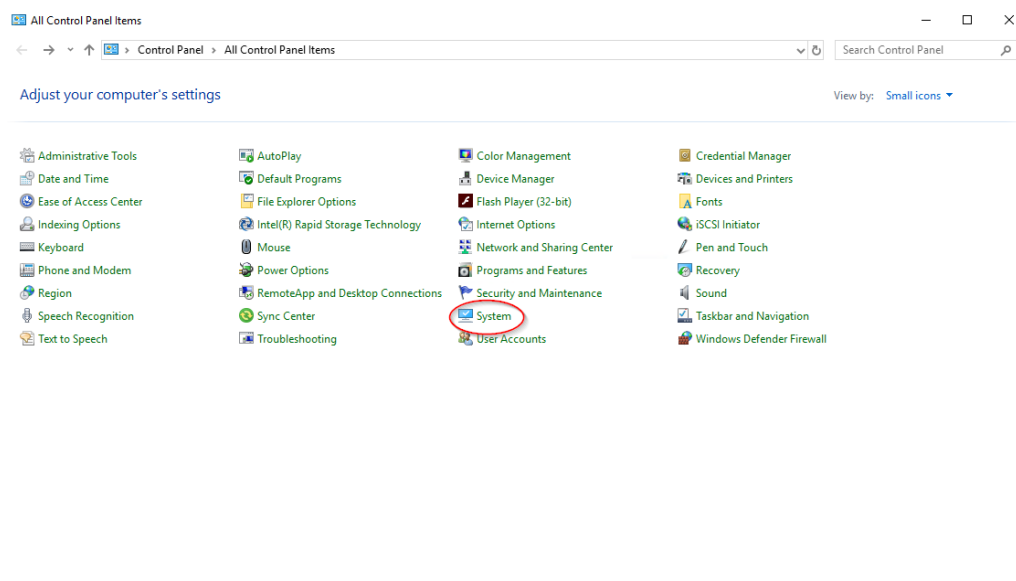
11.1. Opgeloste problemen

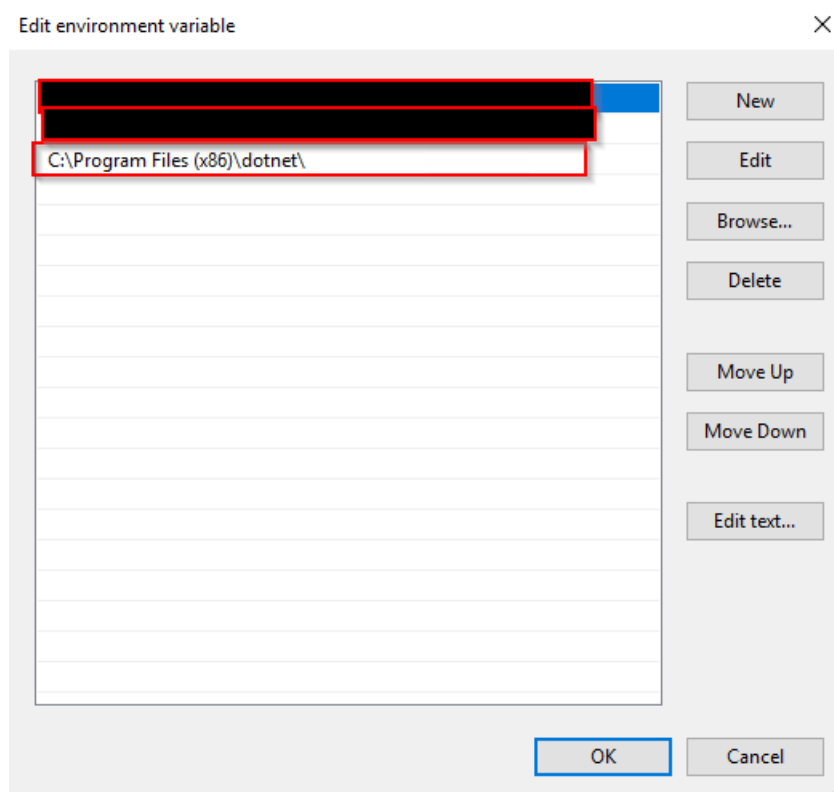
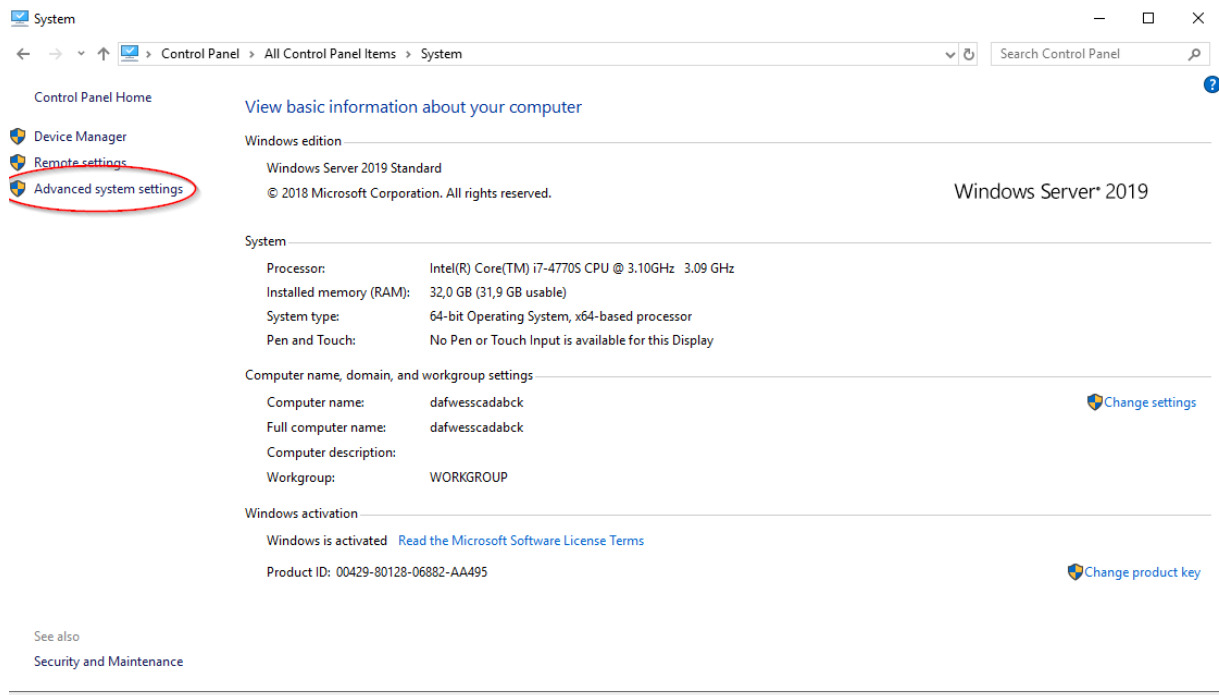
11.1.1. .NET n commando not found

De exe file van ABB eventlog date werd rechtstreeks geëxporteerd samen met een .dll file, App.config file en ook Nlog.config file. Wanneer de exe file gerunt ging worden voor een test kwam er een foutmelding dat het niet kon runnen. Er werd toen naar de event viewer gekeken op de backup server waar deze exe file zich bevindt. Dit was zichtbaar in de event viewer.



Er werd aangegeven dat .Net niet gevonden kan worden. De dotnet commando is niet aanwezig in het systeem dus door dit aan te passen moest dit commando toegevoegd worden aan de environment variable. Om dit te doen moet er naar control panel in windows systeem worden gegaan en daarna wordt systeem geselecteerd.





Voeg dan in user variables for administrator de map waar de dotnet.exe aanwezig is. Daarna kan de commando dotnet --version uitgevoerd worden en wordt er weergegeven welke soort dotnet versie er gebruikt wordt.

Daarna wordt ook de Runtime van .net gedownload, zodat de systeem alle benodigheden heeft voor de applicatie te kunnen uitvoeren.

12. Bronvermelding

Stevewhims. (2023, February 8). Task Scheduler for developers - Win32 apps. Microsoft Learn.
<https://learn.microsoft.com/en-us/windows/win32/taskschd/task-scheduler-start-page>

BillWagner. (2024, April 9). A tour of C# - Overview - C#. Microsoft Learn.
<https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>

Visual Studio: IDE and code editor for software developers and teams. (2024, May 28). Visual Studio.
<https://visualstudio.microsoft.com/>

What is a relational database? (n.d.). <https://oracle.com/be/database/what-is-a-relational-database/>

Overview, Software architecture ABB PCSDK (n.d.).
<https://developercenter.robotstudio.com/api/pcsdk/articles/Manual/Run-time-environment/Overview.html>

Class NetworkScanner. (n.d.).
<https://developercenter.robotstudio.com/api/pcsdk/api/ABB.Robotics.Controllers.Discovery.NetworkScanner.html>

Class ControllerInfo. (n.d.).
<https://developercenter.robotstudio.com/api/pcsdk/api/ABB.Robotics.Controllers.ControllerInfo.html>

Class controller. (n.d.).
<https://developercenter.robotstudio.com/api/pcsdk/api/ABB.Robotics.Controllers.Controller.html>

Rapid data. (n.d.).
<https://developercenter.robotstudio.com/api/robotstudio/articles/Concepts/Rapid/Rapid-Data.html>

Working with RAPID data. (n.d.).
<https://developercenter.robotstudio.com/api/pcsdk/articles/Manual/Using-the-PC-SDK/Rapid-domain/Working-with-RAPID-data.html>



THOMAS
MORE