

# Automatische code en hardwaregeneratie met TIA Portal Openness

Student: Manhaeve Jarne,

Promotoren: naam Pervoot Frederick, Sweertvaegher Isabel

In samenwerking met: Actemium

Academiejaar 2022- 2023

## I. INLEIDING

### A. Actemium

Actemium behoort tot de groep van Vinci Energies en is in vrijwel alles van industriële processen gespecialiseerd.

### B. Opbouw TIA Portal project

Bij de opbouw van een PLC-programma wordt vertrokken van een lijst met alle nodige elektrische aansluitingen. Op basis hiervan wordt het PLC-programma gecreëerd, de PLC-tags en de hardwareconfiguratie aangemaakt, de hardware met de software gekoppeld, en de standaard mappenstructuur opgezet. De veelgebruikte functieblokken worden uit een globale bibliotheek in de mappenstructuur geslept. Verder wordt met deze lijst, code geschreven om signalen die binnenkomen in de PLC te bewerken, eventueel een alarm hoog te zetten op deze signalen of delay's toe te voegen. Deze opgesomde zaken, uitgezonderd het schrijven van de logica, vergen aanzienlijk veel tijd en vallen te automatiseren.

## II. DOELSTELLINGEN

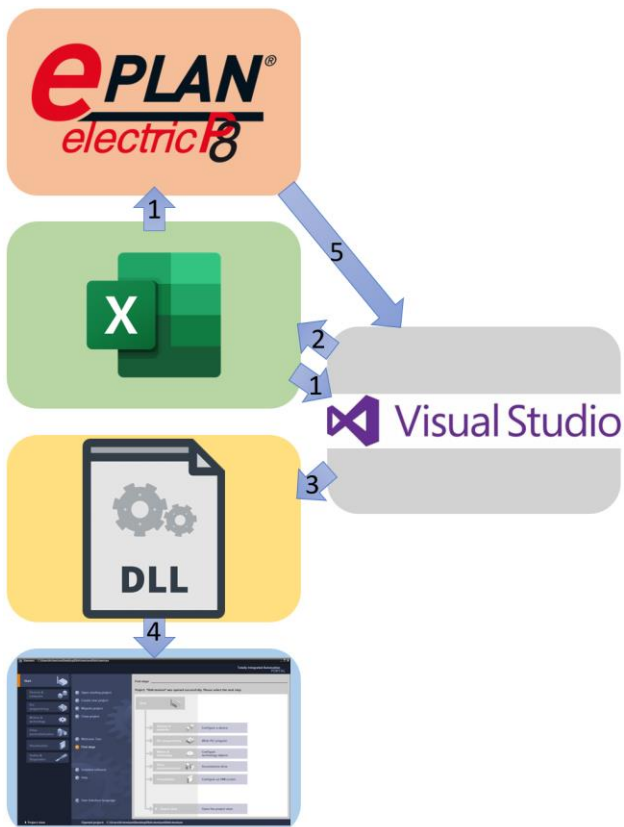
De hoofdoelstelling van deze masterproef is het genereren van code. Dit gaat over de standaardcode om signalen die binnenkomen in de PLC te bewerken. Naast het genereren van code vallen ook veelvoorkomende instellingen te automatiseren in het TIA Portal-project. Een .NET-

applicatie geschreven in C# kan communiceren met TIA Portal via de TIA Portal Openness API (Application Programming Interface). Deze API van Siemens gebruikt DLL-bestanden (Dynamic-Link Library) die worden toegevoegd aan de .NET-applicatie [1]. Ergens dient er een startpunt te zijn voor de codegeneratie. De instrumentatielijst gemaakt in Excel is een lijst waarop alle elektrische aansluitingen staan. Vanaf deze lijst kan een elektrische tekening in Eplan worden gemaakt en kan het programma geprogrammeerd worden in TIA Portal. Deze masterproef focust zich op het werk van het te programmeren gedeelte dat te automatiseren valt. De instrumentatielijst wordt ingelezen en omgevormd tot een objectlijst die de instellingen en parameters meegeeft voor de codegeneratie. De eerste grote doelstelling van deze masterproef is het automatisch omvormen van de instrumentatielijst naar de objectlijst. De tweede grote doelstelling is het genereren van standaardcode in TIA Portal uit deze nieuwe gegenereerde Excellijst. Door het uitlezen van de informatie in de objectlijst kan de standaardcode in XML-formaat worden ingevuld met de velden in de lijst. Doelstelling drie is het importeren van hardware in een TIA Portal-project a.d.h.v. de AML-standaard. Deze standaard is gemaakt voor de uitwisseling van verschillende hardware eigenschappen tussen software van verschillende fabrikanten. De laatste doelstelling is het updaten van de PLC tags, wat ook gebeurt met AML-bestanden.

## III. RESULTATEN

De instrumentatielijst is het startpunt van de projectopbouw (Figuur 1). Hieruit wordt een elektrische tekening in Eplan gemaakt (1) en de

objectlijst worden opgemaakt. Een eigen .NET-applicatie leest de instrumentatielijst in (1) en zet deze om in de objectlijst (2). Door de objectlijst opnieuw in te lezen kan code worden gegenereerd in XML formaat. In TIA Portal kan m.b.v. TIA Openness deze gegenereerde code worden ingeladen. Dit gebeurt a.d.h.v. de .NET-applicatie die de DLL-bestanden aanspreekt (3) en deze sturen de commando's door naar de TIA omgeving (4). Vooraleer de XML-bestanden geïmporteerd worden in een TIA Portal-project, dient eerst de standaardmappenstructuur met bijhorende functieblokken ingeladen te worden. Zo zullen bij het inladen van de XML-bestanden geen compileerfouten meer voorkomen. Als de elektrische tekening in Eplan klaar is kan hieruit een AML worden gegenereerd. De gegenereerde AML wordt geselecteerd met de .NET-applicatie (5) en wordt gebruikt om de PLC tags te updaten in TIA Portal (3) (4) en hiervan de PLC tags te updaten in de objectlijst. Ook wordt met deze AML de gevonden hardware geïmporteerd in TIA Portal.



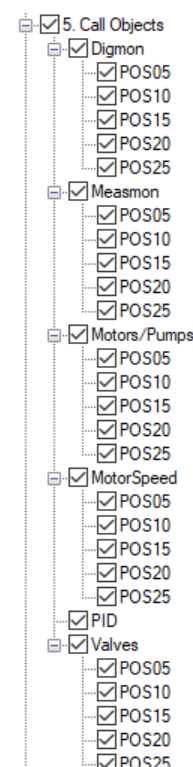
Figuur 1: Flowchart programma

De eigenlijke .NET-applicatie bestaat uit vier tabbladen die ook de doelstellingen vertegenwoordigen; generatie van de objectlijst,

codegeneratie, hardwaregeneratie en PLC tags updaten. Figuur 2 toont het eerste tabblad van de applicatie, die verantwoordelijk is voor het genereren van een objectlijst. Hier kan een instrumentatielijst gekozen worden met de tabbladnaam waarop de nodige info staat. Het tweede tabblad zorgt voor de generatie van XML-bestanden gebaseerd op een objectlijst waarna deze worden ingeladen in TIA Portal. De mappenstructuur aanwezig in een TIA bestand, kan hier gekozen worden met checkvakken. De standaardmappen worden altijd aangemaakt terwijl de posities bijkomstig zijn. Een positie geeft aan in welke elektrische kast welke aansluitingen zich bevinden. Een aangevinkte positie zorgt ervoor dat code van de positie wordt aangemaakt in het XML-bestand dat vervolgens geïmporteerd wordt in het TIA Portal-project (Figuur 3).

| F1  | F2      | Motor-en instrumentatielijst | F4    | F5  | F6     | F7    | F8     |
|-----|---------|------------------------------|-------|-----|--------|-------|--------|
| #   | Edit by | Positie                      | Groep | Tag | SubTag | Type  | Omschr |
| C9  |         |                              |       |     |        |       |        |
| C9  |         |                              |       |     |        |       |        |
| C9  |         |                              |       |     |        |       |        |
| C9  |         | 05                           |       |     |        |       | Rollen |
| C9  |         | 05                           | MA1   | IU1 | V20    | Drive | Rollen |
| C10 |         | 05                           | MA1   | IQ1 | TH     | DI    | Beveel |
| C11 |         | 05                           | MA1   | ZU1 | V20    | Drive | Rollen |
| C12 |         | 05                           | MA1   | ZG1 | TH     | DI    | Beveel |

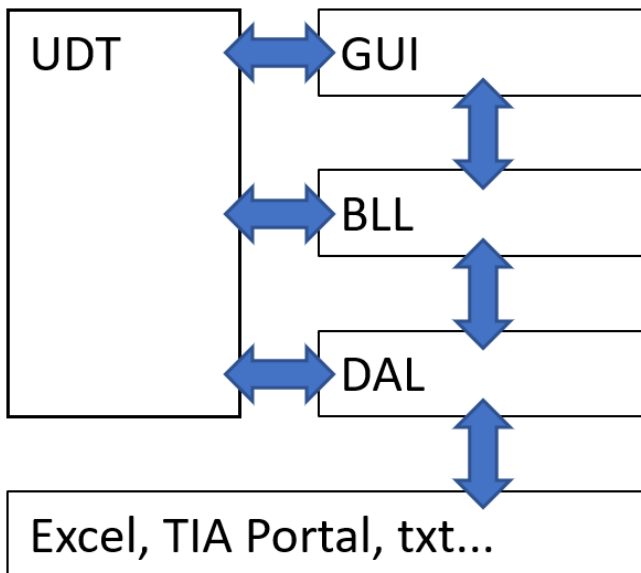
Figuur 2: Applicatie voor de code- en hardwaregeneratie



Figuur 3: Posities aan- of uitvinken

De aanwezige FB's en FC's in deze mappenstructuur worden standaard ingeladen zodat bij het inladen van een gegenereerde XML geen fouten meer voorkomen. Een TIA Portal bestand wordt hier aangemaakt of gekozen net als de gekozen bibliotheek die verder gebruikt wordt voor de standaardblokken in te laden. Het derde tabblad zorgt voor het inladen van de gevonden hardware uit een Eplan project a.d.h.v. een AML-bestand. Het laatste tabblad zorgt voor een update van de PLC tags naar het TIA Portal-project. Vanuit het TIA Portal-project worden de PLC tags in de objectlijst geüpdatet. In de applicatie kan telkens gekozen worden of de data van de vorige stap wordt gebruikt of dat er van een ander project vertrokken wordt. Als bijvoorbeeld een objectlijst gegenereerd wordt kan er gekozen worden voor de nieuwe lijst of een reeds bestaande lijst.

Om de code overzichtelijk te houden, wordt gewerkt met vier verschillende layers (Figuur 4) volgens de 4 lagen architectuur; BLL, DAL, GUI en UDT.



Figuur 4: Structuur van de code

De BLL-map bevat de klassen met de logica. Dit zijn de klassen die bestanden openen, opslaan, bestandspaden hervormen en XML-bestanden creëren. Ook de klasse die de foutafhandeling op zich neemt valt hieronder.

De DAL-map heeft de klassen onder zich die data uitwisselen met softwareprogramma's. In deze masterproef is de data uitwisseling voornamelijk tussen Excel, om de objectlijst te genereren en te

lezen, en TIA Portal, om acties uit te voeren met Openness. Het aanmaken van een nieuw Excel-bestand gebeurt met een API van Microsoft: de Office.Interop bibliotheek.

De derde map GUI heeft één klasse, de interface van de applicatie. De vierde en laatste map UDT bevat de klassen waarbij een eigen datatype is gedefinieerd om het programmeren van de andere klassen te vereenvoudigen. Dit zijn de klassen om bij het inladen van een AML, de fouten uit te lezen en vervolgens op te lossen, en twee klassen die zijn toegevoegd om aangeduide mappen in de mappenstructuur mee te importeren.

#### IV. BESLUIT

De doelstellingen van de masterproef zijn min of meer gehaald. A.d.h.v. een eigen .NET-applicatie wordt het aanmaken van een TIA project zeer efficiënt geautomatiseerd, enkel logische keuzes moeten nog worden gemaakt door de programmeur m.b.v. de objectlijst. Verder werkt het genereren van een objectlijst vertrekkende van een instrumentatielijst zoals het hoort en de gegenereerde XML-bestanden worden op een correcte manier ingeladen in een TIA project. Enkel de tabbladen digmon en measmon worden tot op heden gegenereerd, dit is dus nog uitbreidbaar met motors/pumps, motorspeed en valves. Het inladen van een AML-bestand in TIA Portal voor de hardwaregeneratie gemaakt in Eplan en het updaten van de PLC-tags worden ook uitgevoerd. Naar de toekomst toe kan de code die de applicatie bevat nog meer worden getest op uitzonderingen en uitgebreid voor een nog optimalere automatisatie. Deze scriptie is een aanzet om met de TIA Openness API zoveel als mogelijk TIA projecten optimaal te automatiseren.

#### V. REFERENTIES

AG, S. (sd). *TIA Portal Openness*. (Siemens AG) Opgeroepen op September 30, 2022, van <https://support.industry.siemens.com/cs/document/108716692/tia-portal-openness-introduction-and-demo-application?dti=0&lc=en-WW>