

Studie IEC-1131 Norm (deel 1 t/m 3)

HOGESCHOOL WEST-BRABANT PLC-practicum
C.J.N. de Kok

25 JANUARI 1994

1 IEC-1131 ALGEMEEN

Deze standaard heeft betrekking op programmable controllers ofwel: PLC's. Tevens heeft deze norm betrekking op alle mogelijke randapparatuur van PLC's, zoals: programmeer- en debugging tools (PADTs), testapparatuur en Mens Machine Interfaces (MMI).

Omdat een PLC met randapparatuur vaak beschouwd wordt als een component van een regelsysteem, wordt nadrukkelijk vermeld, dat deze norm alleen betrekking heeft op de PLC zelf. Dus niet op het regelsysteem waarin het PLC-systeem opgenomen kan zijn.

Daarnaast valt ook algemene systeem veiligheid, inclusief installatie en applicatie buiten de beschouwing van deze norm.

Het doel van deze norm is in een aantal punten samen te vatten:

- het vastleggen van definities en principiële karakteristieken die relevant zijn voor de selectie en toepassing van PLC's en hun randapparatuur;
- de minimale systeem eisen vastleggen m.b.t.: functionaliteit, service, constructie, algemene veiligheid en test mogelijkheden (voor PLC's en randapparatuur);
- voor elk van de meest bekende programmeertalen vastleggen wat de algemene toepassingen zijn, de syntax-regels, een compleet set basis instructies, toepassingstests en middelen waarmee de fabrikant de basis sets kan uitbreiden of aanpassen aan hun eigen PLC-implementatie;
- aanvullende informatie geven, inclusief toepassingsaanwijzingen voor de gebruiker;
- het vastleggen van de communicatie tussen PLC's en andere elektronische systemen m.b.v. Manufacturing Message Specification (MMS) volgens ISO/IEC 9506.

2 IEC-1131-1

In deel 1 van de IEC-1131 norm worden een aantal termen en definities besproken, die voor de delen 2 t/m 5 als vanzelfsprekend beschouwd worden. Ook worden de functionele- en karakteristieke eigenschappen besproken.

2.1 DEFINITIES

Hoofdstuk 2 van de IEC-1131-1 norm geeft een aantal definities die niet terug te vinden zijn in het International Elektrotechnical Vocabulary (IEV), deze zijn echter van belang voor algemeen gebruik in de IEC-1131 norm.

2.2 AANVULLING

Hoofdstuk 3 van de IEC-1131-1 norm bevat definities die verklaard zijn in het IEV of in andere IEC en ISO documenten. Ze worden echter frequent gebruikt in de IEC-1131 en daarom nog eens kort en overzichtelijk vermeld.

2.3 FUNCTIONELE EIGENSCHAPPEN

Hoofdstuk 4 van de IEC-1131-1 norm geeft een overzicht van de functionele eigenschappen van een PLC.

De functionele eigenschappen geven de mogelijkheden van een PLC weer, aan de hand van een blokschematische voorstelling (bladzijde 43/IEC-1131-1). Deze figuur is ook in bijlage 1 van deze studie afgebeeld. Te zien zijn de volgende functies:

- signaal bewerkingfunctie;
- interfacefunctie tussen sensoren en actuatoren;
- communicatiefuncties;
- mens-machine interfacefuncties;
- programmeer-, debug-, test- en documentatiefuncties;
- power-supply functies.

Deze blokken worden in de paragrafen van hoofdstuk 4 besproken, waarbij doel en eigenschappen aan de orde komen.

In paragraaf 4.9 van de IEC-1131-1 norm komen de ergonomische eigenschappen van een PLC aan de orde.

P.S.: Deze ergonomische eigenschappen bestaan uit een aantal aanwijzingen voor een mogelijk display en keyboard. Verder stelt dit niet al te veel voor.

2.4 BIJLAGE'S A EN B:

Bijlage A (bladzijde 61/IEC-1131-1) illustreert de hardware van een PLC-systeem. Deze afbeelding is weergegeven in bijlage 2 van deze studie.

Bijlage B (bladzijde 63/IEC-1131-1) illustreert een typisch PLC-systeem met zijn interfaces naar de buitenwereld. Dit is een beknopte weergave van het elektrische systeem, zie hoofdstuk 3 van deze studie. Deze afbeelding is weergegeven in bijlage 3 van deze studie.

3 IEC-1131-2

In deel 2 van de IEC-1131 norm komen de mechanische-, elektrische- en functionele eisen aan de orde, die aan een PLC en/of alle mogelijke randapparatuur gesteld kunnen worden. Daarnaast worden ook condities aangegeven, die gesteld kunnen worden aan service, opslag en transport.

In deze norm wordt ook een vermelding gemaakt van alle gegevens die de fabrikant verplicht dient te leveren aan de gebruiker.

Tenslotte worden ook een aantal testprocedures gegeven, die van toepassing zijn op de PLC

en/of zijn randapparatuur. Deze tests zijn niet geschikt voor een test in de toepassings situatie van een PLC.

3.1 DEFINITIES

Hoofdstuk 1 van de IEC-1131-2 norm geeft een aantal definities die specifiek van toepassing zijn op dit deel van de IEC-1131 norm. Ook worden een aantal, voor de samenstelling van deze norm gebruikte normen genoemd.

3.2 SERVICE CONDITIES

In hoofdstuk 2 van de IEC-1131-2 norm worden de grenzen afgebakend, die de gebruiker in het oog moet houden tijdens het operationeel gebruik van de PLC en/of randapparatuur. Deze condities beslaan de volgende vlakken:

- Fysische omgevingscondities zoals:

- * omgevingstemperatuur;
- * opslagtemperatuur;
- * luchtvochtigheidsgraad;
- * vervuilingsgraad;
- * hoogte;

- Elektrische condities en eisen:

- * ruiscondities;

P.S.: deze gestelde eisen zijn niet zo heel erg duidelijk weergegeven. Misschien kan hier nog wat onderzoek naar gedaan worden.

- * overvoltage periodiek/niet-periodiek;

- Mechanische condities en eisen:

- * trillingen;
- * schokken;
- * vrije val;

Voor speciale condities die hierin niet beschreven zijn, zoals zware stofvervuiling o.i.d. dient de fabrikant geraadpleegd te worden.

Naast de eisen die aan normaal gebruik gesteld kunnen worden, zijn er ook een aantal eisen m.b.t. transport en opslag gesteld:

- omgevingstemperatuur;
- luchtvochtigheidsgraad;
- omgevingsluchtdruk;
- vrije val;
- enkele uitzonderlijke condities.

3.3 ELEKTRISCHE EISEN

Figuur 1 op bladzijde 25 van de IEC-1131-2 norm geeft blokschematisch de elektrische opbouw van de PLC. Deze afbeelding is weergegeven in bijlage 3 van deze studie.

In de elektrische eisen, alsmede de mechanische eisen, zijn ook de functionele eisen geïntegreerd.

De hoofdpunten uit deze elektrisch eisen, alsmede een korte beschrijving zullen in de volgende paragrafen gegeven worden.

3.3.1 AC en DC power-supply

Van de voeding worden de volgende punten behandeld:

- voedingsspanningen en hun toleranties;
- spanningsval en onderbrekingen;
- niet-periodieke overspanningen;
- Memory back-up voorzieningen;
- informatie door de fabrikant te leveren.

Hier worden een aantal eisen behandeld, die verder niet van belang zijn voor de gebruiker van een PLC. Dit zijn enkele vastgestelde gegevens, die de fabrikant van de voeding nodig zal hebben.

3.3.2 Digitale in- en outputs

Paragraaf 3.3 van de IEC-1131-2 norm begint met een aantal algemene eisen en opmerkingen m.b.t. digitale in- en outputs. Daarnaast komen de volgende punten aan de orde:

- digitale inputs (current sinking):

- * benoeming van de verschillende spanning- en stroomnivo's; hierin zijn de grenzen gedefinieerd die de 'hoog'/'laag' gebieden aangeven;
- * werkspanningen van digitale "current sinking" input's;
- * aanvullende eisen;
- * informatie door de fabrikant te leveren;
- * methode voor het verzorgen van compatibiliteit van benaderingsschakelaars;

P.S.: De hier gestelde eisen zijn voor de gebruiker niet allemaal van even groot belang.

- digitale outputs voor AC (current sourcing):

- * benoeming van verschillende stroomnivo's, waarbij tevens de herhaling van een tijdelijke overload aan de orde komt;
- * aanvullende eisen;
- * informatie door de fabrikant te leveren;

P.S.: Hier geldt dezelfde opmerking als voorgaande.

- digitale output voor DC (current sourcing)
(hier gelden dezelfde punten als voor AC)

P.S.: Hier geldt dezelfde opmerking als voorgaande.

Current sinking: stroom ontvangen
Current sourcing: stroom leveren

3.3.3 Analoge in- en outputs

Voor zowel de analoge in- als output, worden de werkspanningen en de door de fabrikant te leveren informatie vermeld. Dit zijn vastgelegde waarden die voor de gebruiker grotendeels onbelangrijk zijn.

P.S.: Er zijn nog meer verschillende typen in- en uitgangskarten, die echter door deze norm niet, of onnauwkeurig beschreven worden. Dit is ook best enige studie waard.

3.3.4 Randapparatuur

Ditzelfde soort gegevens is vastgelegd voor de randapparatuur die kan bestaan uit:

- communicatie interfaces;
- CPU's en geheugens van het PLC-systeem;
- In- en Output stations op afstand van de PLC;
- Peripherals;

3.3.5 Ruis-immuniteit en uitgezonden ruis

Omdat een PLC-systeem allerlei signalen binnenkrijgt, kan hij erg gevoelig voor omgevingsruis worden. D.m.v. het nauwgezet opvolgen van de installatie eisen kan veel ruis worden weggenomen. Deze eisen zijn opgenomen in deel 4 van de IEC-1131 norm.

Paragraaf 3.9 van de IEC-1131-2 norm geeft nog enkele testgegevens.

P.S.: De hier vermelde gegevens zijn zeer onduidelijk in dienen nog wat nauwkeuriger doorgenomen te worden. Dit geldt overigens ook voor de volgende paragraaf: Diëlektrische waarden.

3.3.6 Diëlektrische waarden

Deze waarden staan in relatie met de isolatie ter bescherming tegen elektrische schokken of immuniteit tegen RFI signalen. Paragraaf 4.3 van de IEC-1131-2 norm geeft een aantal isolatieeisen met als doel: bescherming tegen elektrische schokken.

Paragraaf 3.10 van de IEC-1131-2 norm geeft nog enkele testgegevens.

3.3.7 Zelftest gegevens

Een PLC-systeem dient te zijn uitgevoerd met een aantal zelftestfuncties die in deze paragraaf van de IEC-1131-2 norm kort besproken worden.

3.4 MECHANISCHE EISEN

Deze mechanische eisen hebben volgens hoofdstuk 4 van de IEC-1131-2 norm betrekking op de losse onderdelen van het PLC-systeem, zoals: CPU, RIO's, permanente- niet permanente randapparatuur en verder alle componenten die verbonden zijn met de power-supply van het systeem.

Hieruit kan geconcludeerd worden, dat deze mechanische eisen eigenlijk het geheel van het PLC-systeem beslaan. Voor de schrijvers van deze norm, is dit gewoon een keuze om het systeem vast te leggen.

In de norm wordt uitdrukkelijk vermeld, dat naast deze mechanische constructies een PLC-systeem ook moet voldoen aan de in de vorige paragrafen genoemde elektrische- en omgevingscondities.

De belangrijkste punten zullen hier kort aan de orde komen:

- bescherming tegen elektrische stroomstoten, waarbij een drietal klassen onderscheiden worden;
- de eisen gesteld aan clearance- en creepage distance;

clearance = de kortste afstand tussen twee geleiders of tussen een geleider en de geleidende behuizing, gemeten door lucht.

creepage = de kortste afstand tussen twee geleiders of tussen een geleider en de geleidende behuizing gemeten, langs de oppervlakte van het isolatiemateriaal.

- ontvlambaarheid van isolatiemateriaal;
- eisen gesteld aan de behuizing van het geheel;
- mechanische eisen gesteld aan de draadverbindingen;
- voorziening voor beschermingsaarde;
- functionele aarde;
- verbindingen van kabels;
- het in- en uitpluggen van externe units;
- eisen gesteld aan de batterij- en backupvoeding;
- typeplaatje.

3.5 INFORMATIE DOOR DE FABRIKANT TE LEVEREN

In dit deel is een aantal malen naar voren gekomen, dat de fabrikant bepaalde informatie dient aan te leveren. Hoofdstuk 5 van de IEC-1131-2 norm geeft een cross-referencelijst over dit onderwerp. Tevens wordt hier aangegeven in welke vorm de informatie geleverd dient te worden. Er zijn een drietal informatiemedia die gebruikt moeten worden:

- catalogus en datasheets;
- gebruikershandleidingen;
- technische documentatie.

Verder wordt er aandacht besteed aan informatie betreffende betrouwbaarheid en veiligheid van de PLC-systemen.

P.S. Dit hoofdstuk uit de IEC-1131-2 norm is eigenlijk een samenvatting van de informatie die door de fabrikant geleverd dient te worden. De gedetailleerde omschrijving van de benodigde informatie is over de hele IEC-1131-2 norm verspreid.

Misschien is hiervan een overzicht te maken, evt. m.b.v. software. Dit zou als een standaard dienst kunnen gaan doen, zodat de bijgeleverde documentatie van iedere fabrikant op een zelfde manier opgebouwd is.

3.6 TESTPROCEDURES

Het doel van hoofdstuk 6 van de IEC-1131-2 norm is het aangeven, hoe men kan nagaan, of het PLC-systeem en de aanverwante apparatuur voldoet aan de eisen die gesteld worden in deel 1 en 2 van de IEC-1131 norm.

Deze tests zijn opgebouwd uit een typetest en een deugdelijk onderzoek, visuele inspectie en/of een meting. Daarbij wordt voorop gesteld, dat het hier gaat om tests voor het PLC-systeem, evt. inclusief randapparatuur en niet voor het hele besturingssysteem.

Een aantal testprocedures die aan de orde komen zijn o.a. de volgende:

- Verificatietests; In deze test wordt de goede werking van het PLC-systeem getest aan de hand van de door de fabrikant geleverde informatie en testprogramma's;
- Klimaattests; In deze test wordt de apparatuur getest op de geschiktheid voor klimatologische condities;
- Mechanische tests; Ook hier wordt de geschiktheid van de apparatuur op verschillende vlakken getest. De volgende punten komen hierbij aan de orde:

- * vibraties;
- * schokken;

- * vrije val (normaal bedrijf);
- * vrije val (transport);
- * aan veiligheid gerelateerde tests;
- * verificatie van aansluitgegevens;
- * aan- en afkoppelen van losse apparatuur.

- Elektrische tests; Een aantal punten die hier aan de orde komen zijn:

- * aan veiligheid gerelateerde tests;
- * ruis;

* controle op de specificaties van de power-supply; hiervoor worden een viertal punten getest:

- + inkomende voedingsspanning- en frequentievariatie test;
- + externe voeding onderbrekingstest;
- + externe voeding variatietest;
- + verkeerde aansluitings test;

Daarnaast wordt er nog aandacht besteedt aan de backup van het geheugen;

* verificatie van de in- en output karakteristieken; ook hier komen een aantal punten aan de orde, die echter niet verder vermeld zullen worden;

- * controle op de MPU (CPU) karakteristieken;
- * controle van de afstand I/O-stations;
- * controle op de randapparatuur karakteristieken;
- * controle van het basis PLC-systeem, zoals zelftest en diagnostics;
- * controle op de merkplaatjes en documentatie.

- Routinetests; deze routinetests hebben betrekking op:

- * standaard diëlektrische sterktest;
- * alternatieve diëlektrische sterktest;
- * beschermingsaarde continuïteitstest.

P.S. Al deze testprocedures maken gebruik van de specificaties uit voorgaande paragrafen. De gegevens van deze testprocedures, zoals deze opgesteld zijn in de IEC -1131-2 norm, zijn zeer beknopt en erg onoverzichtelijk.

Om dit tot een overzichtelijk geheel te maken, evt. zelfs tot een standaard, kunnen deze gegevens verwerkt worden met een computerprogramma. Dit kan dan waarschijnlijk verwerkt worden in combinatie met de door de fabrikant te leveren gegevens.

Het onderwerp kan in ieder geval nog wel enige studie gebruiken.

Opmerking:

==> Op bladzijde 30 van de IEC-1131-2 norm, staan een tweetal opmerkingen onder NOTES. NOTE 2 gaat over PLC-systemen die niet geleverd zijn met interfaces die deze norm beslaan. Dit gaat dan onder andere over TTL- en CMOS interfaces.

Dit hardwarematige probleem kunnen wij als school best aan een studie onderwerpen, en dan eventueel hierover een eisenpakket samenstellen.

4 IEC-1131-3

Deel 3 van de IEC-1131 norm heeft betrekking op de presentatie van de programmeertaal, waarbij gebruik gemaakt wordt van het ISO-646 karakterset.

Deze presentatie kan zowel in gedrukte vorm als op een beeldscherm geschieden, maar heeft altijd betrekking op PLC-systemen die volgens de IEC-1131-1 norm gedefinieerd zijn.

De IEC-1131-3 heeft geen betrekking op grafische- of semigrafische presentaties van de programmeertaal.

P.S.: Deze opmerking, zoals deze geplaatst is in de IEC-1131-3 norm is eigenlijk niet erg duidelijk. Wat hiermee daadwerkelijk bedoeld wordt, moet uitgezocht worden.

4.1 OVERZICHT EN ALGEMENE EISEN

In deze norm komen de volgende zaken aan de orde:

- de syntax- en semanticregels van de programmeertaal bestaande uit twee tekstgeoriënteerde talen:

- * instructielijst;
- * structurele tekst;

- SFC's die gebruikt worden voor het structureren van de opbouw van PLC-programma's en functieblokken;

- Ondersteunende eigenschappen m.b.t. communicatie tussen PLC's en andere componenten van geautomatiseerde systemen.

De programmeertaalelementen die in dit deel gedefinieerd worden, mogen gebruikt worden in een interactieve programmeeromgeving. Dit valt echter buiten het bestek van deze norm.

In hoofdstuk 1.4 van de IEC-1131-3 norm worden een aantal modellen besproken die van belang zijn bij het programmeren van een PLC-systeem. Dit zijn o.a.:

- het softwaremodel;
- het communicatiemodel;
- het programmeermodel.

Daarnaast wordt er een overzicht gegeven van de eisen van deze norm waaraan een PLC-programma dient te voldoen.

4.2 DEFINITIES

Paragraaf 1.2 van de IEC-1131-3 norm geeft een lijst met normen, die bij de samenstelling van de IEC-1131-3 aan de orde kwamen.

Paragraaf 1.3 van de IEC-1131-3 norm geeft weer een aantal definities die in deze norm specifiek vermeld worden.

4.3 PROGRAMMEER ELEMENTEN

Hoofdstuk 2 van de IEC-1131-3 norm definieert tekstuele- en grafische elementen voor algemeen gebruik in de programmeertalen, die in dit deel van de IEC-1131 norm vastgelegd worden.

4.3.1 Gebruik van karakters

- karakter set:

Tekstuele programmeertalen en tekstelementen van grafische programmeertalen moeten voldoen aan de "Basic Code Table" van het ISO-646 karakterset.

Een aantal aanvullingen en uitzonderingen hierop worden in paragraaf 2.1.1. van de IEC-1131-3 norm vermeld;

- identiërs:

Hieronder worden verstaan: de namen van toestanden, subroutines, variabelen etc. Enkele hieraan gestelde eisen worden vermeld in paragraaf 2.1.2 van de IEC-1131-3 norm;

- sleutelwoorden:

Dit zijn unieke combinaties van karakters die individuele syntaxelementen voorstellen en die daarom ook niet als identiër gebruikt mogen worden;

- gebruik van spaties:

De gebruiker mag overal spaties toevoegen, behalve in sleutelwoorden, literals (getallen), identiërs of afschermd combinaties;

- commentaar:

De gebruiker kan commentaar toevoegen wat gemarkeerd wordt door de combinaties "(" en ")" aan het begin of aan het eind van het programma. Met uitzondering van instructielijsttalen, waar commentaar overal geplaatst mag worden, waar ook spaties geplaatst mogen worden.

Commentaar mag geen syntactic- of semanticwaarde hebben.

4.3.2 Externe presentatie van data

- Numeric literals:

Numeric literals zijn numerieke waarden, die in twee klassen voor kunnen komen: integer- en real literals.

* Integer literal:

Numerieke waarden komen voor als een decimaal getal, of als een getal dat gebaseerd is op een talstelsel (2, 8 of 16). Tussen de cijfers van een numerieke waarde, kunnen underscores "_" voorkomen, die echter geen decimale waarde hebben.

* Real literal:

Een decimaal getal met de aanwezigheid van een punt, wordt een real literal genoemd, wordt eventueel vergezeld van een plus- of minteken.

Hoe dit allemaal weergegeven dient te worden, is te zien in tabel 4 op bladzijde 27 van de IEC-1131-3 norm;

- characterstring literals:

Dit is een combinatie van nul of meerdere karakters, die beginnen en eindigen met een 'singlequote' karakter ('). Hierbij worden driekarakter codes, voorafgegaan door een dollarteken (\$), gezien als de hexadecimale presentatie van een 8-bits karaktercode.

Zie ter informatie de tabellen 5 en 6 op bladzijde 28 van de IEC-1131-3 norm.

P.S.: Geheel is een moeilijk te volgen verhaal.

- time literals:

Er zijn een tweetal tijdsafhankelijke data die in tabel 7 en 8 van de IEC-1131-3 norm behandeld worden:

* duration, ofwel het tijdsverloop of de tijdsduur van een bepaalde gebeurtenis of handeling;

* time of day and data; deze worden gebruikt om bepaalde gebeurtenissen op elkaar te synchroniseren d.m.v. een koppeling aan een datum of een tijd.

4.3.3 Datatypes

Onder datatypes worden verschijningsvormen van data verstaan, waarbij boolean, integer- en real getallen, datum en strings enkele van de mogelijkheden zijn. Tevens wordt hier de mogelijke lengte van een datablok aangegeven, b.v.: 1, 8, 16, 32 of 64 bits.

Er bestaan een drietal datatypes die kort aan de orde zullen komen:

- Elementary datatypes:

Dit zijn een aantal vooraf gedefinieerde datatypes, zoals deze door de IEC-1131-3 norm beschreven zijn (zie tabel 10). In deze tabel wordt het sleutelwoord, aantal bits en het bereik per datatype gegeven;

- Generic datatypes:

Deze zijn kenmerkend door hun soort data: numeriek, strings, bits, etc. Ze worden gebruikt in combinatie met zogenaamde "overloaded inputs", die kunnen werken met verschillende dataelementen binnen een generic datatype (zie tabel 11, IEC-1131-3 norm). In tabel 11 wordt ook de hiërarchie van deze generic datatypes aangegeven. Deze datatypes worden aangegeven met de prefix "ANY";

- Derived datatypes:

Dit zijn datatypes die door de gebruiker gespecificeerd kunnen worden d.m.v. de tekstuele constructie TYPE...END_TYPE. In paragraaf 2.3.3 van de IEC-1131-3 norm wordt dit uitvoerig behandeld.

Datatypes worden zodanig gedefinieerd dat ze vaste grenswaarden hebben.

P.S. Datatypes definiëren is een lastige bezigheid die zeker nog enige studie vereist. Zie ook het eerste deel van PLC-opening (february 1994) dat wij ontvangen hebben.

4.3.4 Variabelen

In paragraaf 4.3.2 van deze studie werd de presentatie van externe data besproken. Dit zijn vormen van data, waarvan de inhoud niet wijzigt. Daarnaast bestaan er 'data-objecten' waarvan de inhoud wel kan wijzigen. Deze worden variabelen genoemd en zijn gekoppeld aan ingangen, uitgangen of geheugenplaatsen van de PLC.

Er bestaan een aantal vormen waarin deze variabelen gepresenteerd kunnen worden:

- single-element variabelen:

Deze vertegenwoordigen één data-element van de elementaire datatypes of van de derived datatypes. Ze kunnen op twee manieren gepresenteerd worden: symbolisch of op een alternatieve manier.

De symbolische notatie geschied m.b.v. identiërs, volgens paragraaf 2.1.2 van de IEC-1131-3 norm.

De alternatieve presentatie geeft mede de locatie (adres) van de input, output of geheugenplaats en wordt ook wel de directe notatie genoemd.

|%: directe notatie

|Q: locatie

|X: grootte

Voorbeelden: * %QX75 output bit 75

* %IW215 input word locatie 215

* %QB7 output byte locatie 7

* %MD48 double word op geheugen- locatie 48

* %IW2.5.7.1 (uitleg)

Uitleg: op deze manier kan op een hiërarchische wijze het adres vanaf het hoogste tot aan het laagste level weergegeven worden. Dit is b.v. het eerste 'woord' van de zevende 'module' in het vijfde 'rack' op de tweede 'I/O-bus'.

Een voordeel hiervan is, dat er geheugen- of adreslocaties vanuit andere PLC's gelezen kunnen worden. De betekenis van de letters in de adressering kan teruggevonden worden in tabel 15 op bladzijde 38 van de IEC-1131-3 norm.

- multi-element variabelen:

De multi-element variabelen die in de IEC-1131-3 norm gedefinieerd worden, zijn array's en structures.

Een array is een collectie van data-elementen die alle van hetzelfde datatype zijn en genummerd worden met één of meer subscripten tussen haakjes, gescheiden door komma's.

Een structured variabele is een variabele bestaande uit een datatype dat gedefinieerd is als een datastructure. Dit is verder uitgelegd op bladzijde 38 (onderaan) van de IEC-1131-3 norm.

- initialisatie:

Als een PLC-programma opgestart wordt, zullen de betreffende variabelen een bepaalde initiële waarde krijgen. Hierin zijn een drietal mogelijkheden te onderscheiden:

- * de waarde die de variabele had, voordat het programma gestopt werd (beschermd);
- * een door de gebruiker gegeven startwaarde;
- * de defaultwaarde die bij het datatype van de variabele hoort.

Indien de hardware het toelaat, kan de gebruiker aangeven of de waarde van de variabele beschermd moet worden. De waarde van de variabele wordt bepaald aan de hand van de volgende regels:

- * bij een 'warme start', zal een beschermd variabele zijn voorgaande waarde onthouden;
- * bij een 'koude start', zal een beschermd variabele de door de gebruiker gegeven startwaarde krijgen, of de defaultwaarde behorende bij het datatype;
- * niet beschermd variabelen krijgen altijd de door de gebruiker gegeven startwaarde, of de defaultwaarde behorende bij het datatype;
- * variabelen die een input van het PLC-systeem vertegenwoordigen, worden afhankelijk van de gebruikte hardware, geïnitieerd.

- declaration:

Aan het begin van ieder programma of functieblok moet er op z'n minst één blok zijn, waarin de variabelen worden gedefinieerd. Dit bestaat uit een tekstueel gedeelte, waarin gebruik gemaakt wordt van de sleutelwoorden: VAR, VAR_INPUT of VAR_OUTPUT en wordt afgesloten met END_VAR. Hier zijn nog een aantal mogelijkheden beschikbaar, maar dat is terug te vinden in tabel 16 op bladzijde 40 van de IEC-1131-3 norm.

De geldigheid van deze gedefinieerde variabelen, is alleen binnen het PLC-programma of functieblok waarin zij gedefinieerd, zijn en niet in een ander programma of functieblok. Tenzij de variabelen gedefinieerd zijn als inputs of outputs van het programmagedeelte. Dit zijn de zogenaamde 'locale' variabelen.

De 'globale' variabelen zijn via de VAR_EXTERNAL definitie beschikbaar te maken voor in een programma blok. Het datatype moet dan wel overeenstemmen met het datatype wat gedefinieerd werd m.b.v. VAR_GLOBAL in een ander programma of functieblok.

Samenvattend kan voor een variabele dus een datatype en een initiële waarde aangegeven worden. De tabellen 17 en 18 op bladzijde 40 t/m 43 van de IEC-1131-3 norm geven hiervan een aantal voorbeelden.

- opmerking:

Het definiëren van variabelen is een lastige bezigheid, maar toont overeenkomsten met andere 'high-level' programmeertalen, zoals PASCAL en C. Het is een nieuwe dimensie van het PLC-programmeren en men zal even de tijd nodig hebben om hieraan gewoon te raken. Voor een duidelijke uitleg, zie PLCopening van februari 1994, bladzijde 4.

4.3.5 Programma opbouwunits

De units waaruit een programma opgebouwd kan worden, zijn: functies, functieblokken en programma's. Deze kunnen geleverd worden door de fabrikant, of geprogrammeerd worden door de gebruiker.

De units mogen elkaar onder geen beding beïnvloeden, d.w.z. dat bij het aanroepen van een functieblok, een ander functieblok of ander deel van het programma niet beïnvloed kan en mag worden. Het gaat dan wel over units van hetzelfde type.

4.3.5.1 functies

Voor de toepassing van PLC-programmeertalen zijn er functies gedefinieerd (een programma opbouwunit), die aangeroepen kunnen worden vanuit een tekstuele omgeving d.m.v. een operand in een expressie.

Bijvoorbeeld: $Y:=\text{SIN}(X)$ {expressie}, waarin SIN {operand} de functie is.

Als een functie aangeroepen wordt, is het resultaat altijd 1 data-element, dat eventueel ook meerdere waarden kan hebben (b.v een array of een structure).

Functies mogen geen interne variabele hebben die een waarde kan vasthouden. Dit zou als effect namelijk verschillende uitgangswaarden bij dezelfde ingangswaarden geven.

Iedere functie die gedefinieerd is, kan in een andere programma opbouwunit gebruikt worden.

- presentatie:

Functies kunnen grafisch of tekstueel gepresenteerd worden. De tekstuele presentatie van een functie, wordt besproken in hoofdstuk 3 van de IEC-1131-3 norm. Voor de grafische presentatie zijn een aantal regels gegeven op bladzijde 45 van deze norm.

- het aanroepen van deze functies:

Voor het aanroepen van de functie wordt gebruik gemaakt van een in- en uitgang, respectievelijk: "EN" (enable) input en "ENO" (enable out) output.

Deze in- en uitgang zijn noodzakelijk voor gebruik in een ladderdiagram (LD), en optie in een Functie Blok Diagram (FBD). Deze in- en uitgang moeten natuurlijk ook gedefinieerd worden en worden aangestuurd volgens een drietal regels, zie bladzijde 46 van de IEC-1131-3 norm.

- definiëren van functies:

Een functie kan tekstueel of grafisch gedefinieerd worden. Zie paragraaf 2.5.1.3. op bladzijde 47 van de IEC-1131-3 norm.

- type, overload en typeomzetting:

Een functie kan 'overloaded' genoemd worden, wanneer hij verschillende datatypes kan verwerken, die weliswaar binnen een bepaald generic-datatype vallen.

Een functie kan ook een bepaald datatype toegewezen krijgen, door i.p.v. ADD, ADD_INT als commando te geven. Het datatype is dan van het type integer.

Daarnaast kan er typeomzetting van de input of de output plaatsvinden. Paragraaf 2.5.1.4. van de IEC-1131-3 norm geeft dit allemaal duidelijk weer d.m.v. afbeeldingen en tabellen.

4.3.5.2. standaard functies

In deze paragraaf worden een aantal functies vermeld, die algemeen in elke programmeertaal voorkomen. In de IEC-1131-3 norm wordt van elke standaardfunctie de grafische en tekstuele presentatie gegeven.

Enkele functies hebben een variabel aantal inputs, waarvan het maximale aantal afhankelijk van de hardware is.

- datatype omzet functies:

Van deze functies is de inputvariabele van het type * en de outputvariabele van het type **. Dit is te schrijven als *_TO_**. Daarnaast bestaan ook de functies TRUNC, BCD TO ** en * TO BCD.

- numerieke functies:

Hiervan zijn de datatypes van de input en de output wel gelijk. De mogelijkheden zijn: ABS, SQRT, LN, LOG, EXP, SIN, COS, TAN, ASIN, ACOS en ATAN.

- rekenkundige functies:

Dit zijn functies met meerdere ingangen die hetzelfde datatype voor in- en uitgang hebben. De mogelijkheden: ADD, MUL, SUB, DIV, MOD, EXPT en MOVE.

- bit-string functies:

Dit zijn functies voor het bewerken van het generic datatype BIT, dus van 1 tot 64 bits. De mogelijkheden: SHL, SHR, ROR en ROL.

- booleaanse functies:

Voor de bewerking van het datatype BOOL, waarbij de mogelijkheden zijn: AND, OR, XOR en NOT.

- selecterende functies:

Voor het selecteren van een ingang op diverse criteria: SEL, MAX, MIN, LIMIT en MUX.

- vergelijkende functies:

Functies die vergelijkende operaties uit kunnen voeren op 2 of meerdere ingangen: GT, GE, EQ, LE, LT en NE.

- karakterstring functies:

Dit zijn een aantal functies die op karakterstrings losgelaten kunnen worden: LEN, LEFT, RIGHT, MID, CONCAT, INSERT, DELETE, REPLACE en FIND.

4.3.5.3 functieblokken

Naast de functies zijn er functieblokken (programma opbouwunit) die i.p.v. 1 data-element, 1 of meerdere data-elementen als resultaat kunnen hebben. Hiervan kunnen 1 of meerdere kopieën gemaakt worden, die ieder hun eigen identiër (naam) hebben. (Denk hierbij bijvoorbeeld aan verschillende timers)

Naast deze identiër hebben ze ieder een datastructuur waarin alle mogelijke input- en outputvariabelen verwerkt zijn, waarbij ook eventueel een waarde of een referentie aan een inputvariabele gegeven kan worden.

De waarden van de uitgangs- en interne variabelen blijven behouden na een executie van het betreffende functieblok. Daarom kan het aanspreken van een functieblok verschillende resultaten opleveren bij gebruik van gelijke ingangsvariabelen.

Van een functieblok mogen alleen de in- en uitgangsparemeters beschikbaar zijn, dus niet de interne variabelen.

Een gedefinieerd functieblok kan weer gebruikt worden in de definitie van een ander functieblok of programmagedeelte. Verder kan een functieblok alleen gebruik maken van lokale variabelen, binnen de programma opbouwunit waarin het is gedefinieerd, tenzij er van globale variabelen gebruik wordt gemaakt (VAR_GLOBAL).

- presentatie:

Ook functieblokken kunnen grafisch of tekstueel gepresenteerd worden. De tekstuele presentatie geschied door een data-element te definiëren dat gebruik maakt van een vooraf gedefinieerd functieblok m.b.v. VAR..END_VAR constructie, identiek aan het gebruik van een structured datatype (paragraaf 2.4.3 van de IEC-1131-3 norm).

V.B.: VAR FF75: SR; END_VAR (figuur 9, bladzijde 62 van de IEC-1131-3 norm)

FF75 = Flip-flop 75, de naam of identiër van het functieblok;

SR = vooraf gedefinieerd functieblok.

In figuur 9 is ook de grafische presentatie van dit functieblok voorbeeld te zien. Ook hiervoor gelden de regels van bladzijde 45 van de IEC-1131-3 norm. Er zijn hierop nog een tweetal aanvullingen:

* de grootte en oriëntatie van het blok mag variëren afhankelijk van het aantal inputs, outputs en andere informatie;

* Voormalige in- en outputparameters (gedefinieerd functieblok) moeten aan de linker- en rechterzijde aan de binnenkant van het blok afgedrukt worden.

Een functieblok kan ook zodanig gedefinieerd worden, dat de uitgangswaarde behouden blijft, bij uitval van de spanning.

Tabel 32 van de IEC-1131-3 norm geeft de mogelijkheden van het gebruik van een I/O-parameter weer.

- definiëren van functieblokken

Het definiëren van een functieblok kan dus tekstueel of grafisch geschieden. Bladzijde 63-69 van de IEC-1131-3 norm geeft overzichtelijk de mogelijkheden weer. Hier zal verder niet op ingegaan worden, mede omdat dit een specifieke programmeeraangelegenheid is.

4.3.5.4 standaard functieblokken

In deze paragraaf worden een aantal functieblokken vermeld, die algemeen in elke programmeertaal voorkomen.

De grafische weergave wordt in een aantal gevallen vergezeld van de tekstuele weergave.

- bistabiele elementen:

Hier vinden we de volgende functieblokken terug: SR (dit is een SET/RESET functie, waarvan SET dominant), RS (dit is een SET/RESET functie, waarvan RESET dominant) en SEMA (dit functieblok wordt gebruikt om de toegang tot hulpfuncties van het operating system te besturen);

- flank detectie:

Dit zijn de functieblokken: R TRIG (opgaande flankdetectie) en F TRIG (neergaande flank detectie);

- counters:

Voor counters zijn er een drietal standaard functieblokken gedefinieerd: CTU (up-counter), CTD (down-counter) en CTUD (up- down-counter);

- timers:

Hier bestaat de mogelijkheid uit een 3-tal timers en een realtime clock. Voor de timers geldt: TP (puls-gever), TON (vertraagd inschakelen) en TOF (vertraagd uitschakelen). Daarnaast kan met de realtime clock de actuele tijd uitgelezen worden, of de actuele waarde aangepast worden.

- communicatie functieblokken:

Deze worden in de IEC-1131-5 norm besproken.

4.3.5.5 programma's

Tot slot bespreekt paragraaf 2.5.3. van de IEC-1131-3 norm kort, hoe een programma opgebouwd wordt en wat een programma is. In ieder geval zijn de benodigde sleutelwoorden PROGRAM...END_PROGRAM en moet er een variabelen declaratie in voorkomen.

4.3.6 Sequential Function Chart (SFC) elementen

Om een programma opbouwunit op een gestructureerde manier op te bouwen, wordt gebruik gemaakt van SFC-elementen. Deze SFC-elementen zijn in principe algemeen toepasbare blokken (gedefinieerd volgens de IEC-848 norm) voor sequentiële processen.

Paragraaf 2.6 van de IEC-1131-3 norm bespreekt deze SFC-elementen in een aangepaste vorm. In de IEC-1131 norm spreekt men van de omzetting van een "documentatie standaard" naar een set van "uitvoer controle elementen".

De SFC-elementen geven een middel om een PLC-programma op te bouwen uit een set van "stappen" en "overgangen" die verbonden worden d.m.v. bijbehorende overgangen. Elke stap vertegenwoordigt een aantal acties en elke overgang een overgangsvoorwaarde.

Omdat SFC-elementen het vastleggen van een bepaalde toestand vereisen (SET/RESET), kunnen alleen de functieblokken en de programma's m.b.v. SFC-elementen geprogrammeerd worden.

4.3.6.1 stappen(toestanden)

Een stap vertegenwoordigt een situatie waarin het gedrag van een programma opbouwunit afhankelijk is van een set acties die bij deze stap horen. De acties zijn mede afhankelijk van de in- of uitgangen die beschikbaar zijn. Een stap is actief of niet-actief.

Verder geldt dat op een willekeurig moment, die status van een programma opbouwunit afhankelijk is van de actieve stappen en de waarden van de interne- of uitwendige variabelen.

Een stap wordt grafisch vertegenwoordigd door een blok met daarin de toestandsnaam (meestal nummer) of tekstueel d.m.v. een STEP...END_STEP constructie. Bij de grafische voorstelling worden de in- en uitgang voorgesteld door een lijn, respectievelijk boven of onder aan het blok. Tekstueel wordt dit gedaan d.m.v. een TRANSITION...END_TRANSITION constructie.

Een stap heeft een vlag (een logische variabele), die aangeeft of een stap actief of niet actief is. Deze vlag bestaat uit het element `***.X` waarin `***` de stapnaam is. `***.X` is '0' of '1' afhankelijk van de logische toestand. Daarnaast bestaat er een variabele van het type TIME, die de verstreken tijd van de stap aangeeft. De waarde die deze variabele heeft op het moment dat de stap niet-actief wordt, blijft hierin opgeslagen totdat de stap opnieuw aangeroepen wordt. Deze variabele wordt aangegeven met `***.T`.

De stapnamen, staptijden en stapvlaggen zijn lokale variabelen. De initiële status van een programma opbouwunit is afhankelijk van de initiële waarden van de variabelen (interne- en uitwendige variabelen) en de initiële toestand(en).

Elk SFC-netwerk heeft één initiële toestand. Een programma kan uit meerdere SFC-netwerken bestaan, zodat er ook meerdere initiële toestanden in een programma voor kunnen komen. Een initiële toestand wordt grafisch voorgesteld door een blok met dubbele lijnen, waarin de stapnaam geplaatst wordt. Tekstueel wordt dit aangegeven met de `INITIAL_STEP...END_STEP` constructie.

Als het systeem opgestart wordt, zijn de staptijden 0 sec. (=initiële waarde), de status van de normale stappen "niet geactiveerd" en de status van de initiële stappen "wel geactiveerd". Tenzij een functieblok of een programma "blijvend" gedefinieerd is. Dan zullen deze variabelen hun oude waarde behouden.

Voor informatie, zie tabel 40 bladzijde 78 van de IEC-1131-3 norm.

4.3.6.2 overgangen

Een overgang vertegenwoordigt de voorwaarde om van een voorgaande stap naar een daarop volgende stap te gaan. Deze overgang vindt plaats via de betreffende directed link (verticale streep) en wordt vertegenwoordigd door een horizontaal streepje door deze verticale lijn. De richting van de overgang moet altijd vanaf de onderkant van de voorgaande stap naar de bovenkant van de daarop volgende stap gaan.

Iedere overgang heeft een overgangsvoorwaarde, welke het resultaat is van de logische vergelijking van boolean variabelen. Een overgangsvoorwaarde die altijd waar is, wordt aangegeven met een "1".

Een overgangsvoorwaarde kan op de volgende manieren worden weergegeven:

- een logische vergelijking aan de rechterzijde van de overgang plaatsen;
- een LD (ladderdiagram) netwerk plaatsen waarvan de rechtse verticale lijn en de uitgang vervangen wordt door de verticale lijn van de overgang;
- een FBD (function block diagram) netwerk plaatsen, waarvan de uitgang de verticale lijn van de overgang snijdt;

- een FBD- of LD netwerk waarvan de uitgang de verticale lijn van de overgang snijdt via een connector;
- in de ST (structured text) taal d.m.v. een TRANSITION...END_TRANSITION constructie;
- in de IL (instruction list) taal d.m.v. een TRANSITION...END_TRANSITION constructie;
- met gebruik van een overgangsnaam rechts van de verticale lijn van de overgang. Deze overgangsnaam heeft betrekking op één van de voorgaande constructies, waarvan de bewerking resulteert in de toekenning van een boolean waarde aan een variabele. Deze variabele is de overgangs-naam.

Een overgangsnaam is lokaal in de programma opbouwunit waarin deze gebruikt wordt. Tabel 41 op de bladzijden 80-82 van de IEC-1131-3 norm geeft het voorgaande overzichtelijk weer.

4.3.6.3 acties

Elke stap vertegenwoordigt 0 of meerdere acties. Een stap die geen acties heeft wordt als een wachtfunctie of wachttoestand gezien.

Een actie voldoet aan een aantal eigenschappen, die in de volgende punten aan de orde komen.

- een actie kan bestaan uit:
 - * een boolean variabele;
 - * een verzameling instructies in de IL-taal;
 - * een verzameling instructies in de ST-taal;
 - * een verzameling van takken in de LD-taal;
 - * een verzameling van netwerken in de FBD-taal;
 - * een SFC zoals in deze paragraaf besproken wordt.

Dit laatste punt geeft tevens weer, dat een stap in combinatie met een actie, eventueel weer door een apart SFC-diagram verder gedetailleerd kan worden.

- aangeven van acties:

Acties kunnen op een aantal manieren aangegeven worden. Dit zijn natuurlijk weer de bekende grafische en tekstuele notatie. Er zijn een vijftal mogelijkheden:

- * grafisch aangeven m.b.v. LD-taal;
- * grafisch aangeven m.b.v. SFC-taal;
- * grafisch aangeven m.b.v. FBD-taal;

dat aan iedere actie dus een actiekwalificatie toegekend dient te worden.

Als de actie vastgelegd is als een boolean variabele, dan zal de Q-uitgang van dit functieblok dezelfde toestand hebben als deze variabele.

Als de actie een verzameling van statements of netwerken is, dan zullen deze uitgevoerd worden zolang de Q-uitgang van het functieblok "1" is. De laatste uitvoering vindt plaats op de neergaande flank van deze Q-uitgang;

*2 Een boolean input voor het actieblok, m.b.t. een actie, moet gekoppeld zijn aan een stap of aan een actieblok. De boolean input kan dan alleen waar worden indien de bijbehorende stap of actieblok actief is. Hiermee is dus de koppeling te maken naar alle stappen die het ACTION_CONTROL functieblok kunnen activeren;

*3 De waarde van de T-input van een ACTION_CONTROL functieblok moet de waarde hebben van de benodigde tijdsduur voor tijdsafhankelijke ingangen, zoals: L, D, SD, DS of SL. Indien deze koppeling niet aanwezig is, moet deze T-input de waarde t#0s krijgen;

*4 Bij de volgende condities zal er een foutmelding op moeten treden:

- + als er meer dan één van de ingangen van het functieblok met een tijdsafhankelijk kwalificatie aangeropen wordt per actie;
- + als de SD(Set Delayed)-ingang "1" is, terwijl de Q-uitgang van het SL_FF(Set Limited) blok "1" is;
- + als de SL(Set Limited)-ingang "1" is, terwijl de Q-uitgang van het SD_FF(Set Delayed) blok "1" is;

Het is niet vereist dat het ACTION_CONTROL blok zelf geïmplementeerd wordt. Wel moet het aansturen (de control) van de acties functioneel gelijk zijn aan de voorgaande regels. Een voorbeeld hiervan is te vinden in figuur 16a en b op bladzijde 91 en 92 van de IEC-1131-3 norm.

4.3.6.4 voortgangsregels

De initiële situatie van een SFC netwerk wordt getypeerd door de initiële stap, die actief wordt zodra het programma of functie blok geïnitieerd wordt. De voortgang van de ene actieve stap naar de andere actieve stap vindt plaats via de directed links, die ontstaan wanneer de overgangen gecleared worden.

Een overgang is vrij gemaakt (enabled) wanneer alle voorgaande stappen actief zijn. Het clearen van een overgang vindt dus alleen plaats, indien een overgang enabled is en de overgangsvoorwaarden waar zijn.

Wanneer een overgang gecleared wordt, heeft dit als gevolg dat de alle voorgaande toestanden gedeactiveerd worden (resetten) en de volgende toestanden geactiveerd (geset)

worden. Deze toestanden zijn dan natuurlijk wel gekoppeld via een overgang.

P.S.: In de IEC-848 norm wordt gesproken over eerst het zetten van de volgende toestand en daarna pas het resetten van de voorgaande toestand. Hiervoor heb ik geen verklaring gevonden!

De combinaties stap/overgang en overgang/stap moeten in een SFC altijd behouden blijven, vanwege een tweetal regels:

-1 twee stappen mogen nooit direct aan elkaar gekoppeld zijn en moeten daarom altijd gescheiden worden door een overgang;

-2 twee overgangen mogen nooit direct aan elkaar gekoppeld zijn en moeten daarom altijd gescheiden worden door een stap.

Als bij het clearen van een overgang een aantal toestanden gelijktijdig actief worden, dan worden de takken waarbij deze verschillende toestanden behoren, simultane takken genoemd. Nadat ze gelijktijdig geactiveerd worden, zullen ze zelfstandig hun weg vervolgen. Dit wordt aangegeven met een dubbele, horizontale streep.

In tabel 46 op de bladzijden 94 t/m 99 van de IEC-1131-3 norm worden de mogelijke combinaties weergegeven.

De clearingtime van een overgang kan theoretisch nagenoeg gelijk aan 0 gemaakt worden, maar zal in de praktijk hieraan nooit gelijk zijn. In de praktijk is dit namelijk afhankelijk van de PLC-implementatie. Dit geldt natuurlijk ook voor de tijdsduur van een stap. Een aantal overgangen die gelijktijdig gecleared kunnen worden (geen simultane takken), kunnen van een prioriteit voorzien worden (tabel 46-2b).

Het testen van de opvolgende overgang zal niet plaatsvinden, voordat de gevolgen (acties) van het activeren van de huidige stap verwerkt zijn door de programma opbouwunit. Dit is de bijna letterlijke vertaling van deze zin uit de norm.

UITLEG: Mijn implementatie van deze zin is, dat de acties uit een stap eerst verwerkt moeten worden, eerdan er naar een nieuwe overgang naar de opvolgende stap gekeken mag worden. Dat houdt dan in, dat eerst de toestanden met de overgangen gedefinieerd moeten worden, daarna de timers/counters en als laatste de aansturing van de uitgangen. Dat is dus de opbouw van een PLC-programma zoals wij die al toepassen. Deze zin geeft de ladderdiagram structuur aan.

De overgang van de ene toestand in de andere wordt m.b.v. figuur 17 op bladzijde 100 en 101 van de IEC-1131-3 norm weergegeven aan de hand van twee mogelijkheden.

Tot slot moet er opgemerkt worden, dat m.b.v. de gegeven regels niet voorkomen kan worden dat er onveilige SFC's gecreëerd kunnen worden. Ook bestaat de mogelijkheid dat er onbereikbare SFC's ontstaan. Figuur 18 van de IEC-1131-3 norm geeft hiervan voorbeelden.

De CPU moet in deze gevallen een foutmelding geven.

4.3.7 Configuratie elementen

Om de functie hiervan duidelijk te maken, moet teruggekeken worden naar het softwaremodel op bladzijde 15 en 16 van de IEC-1131-3 norm.

HIGH-LEVEL programmeertaal elementen:

M.b.v. dit model wordt de basis van "HIGH-LEVEL programmeertaal elementen" besproken (zie bijlage 4 van deze studie). Deze elementen worden geprogrammeerd m.b.v. de programmeertalen die in deel 3 van de IEC-1131 norm vastgelegd zijn. Deze elementen zijn ten eerste: de programma's en de functieblokken en als tweede: de configuratie elementen.

Configuratie elementen:

De configuratie elementen bestaan uit 'configurations', 'resources', 'tasks', 'global variables' en 'access paths'. Zij ondersteunen de installatie van PLC programma's in PLC implementaties.

- Een configuration is het programmeertaal element dat overeenstemt met een PLC-systeem, zoals in deze norm gedefinieerd;
- Een resource bestaat uit een 'signal processing functie' en zijn 'Man Machine Interface (MMI)'- met 'sensor en actuator interface' functies voor zover aanwezig;
- Een task is een executie controle element, dat zorgt voor een periodieke- of getriggerde aansturing van gekoppelde programma opbouwunits.

Een 'configuration' bestaat altijd uit één of meerdere 'resources' die elk weer één of meerdere programma's bevatten, die onder het beheer van nul of meerdere 'tasks' gedraaid worden. Een programma bestaat uit nul of meerdere functie-blokken of andere programmeertaal elementen zoals deze in de IEC-1131-3 norm vastgelegd zijn.

De werking van deze configuratie elementen is als volgt:

Configurations en resources kunnen worden gestart of gestopt via de 'operator interface', 'programmeer, test en monitor' of 'operating system' functies, volgens de IEC-1131-1 norm. Bij het starten van een configuratie worden de globale variabelen geïnitieerd en vervolgens alle resources binnen de configuratie gestart. Het starten van een resource heeft als gevolg, dat alle variabelen binnen de resource geïnitieerd worden en alle tasks enabled. Bij het stoppen van een resource worden deze tasks disabled en bij het stoppen van een configuratie worden alle resources gestopt.

De mechanismen voor het beheer van tasks wordt in deze paragraaf besproken, terwijl de

mechanismen voor het starten en stoppen van 'configurations' en 'resources' in de IEC-1131-5 norm behandeld worden. (De zogenaamde communicatie functies)

Programma's, resources, globale variabelen, acces paths en configurations kunnen geladen of gewist worden d.m.v. de 'communicatie-functie'. Het laden of wissen van een configuration of resource is gelijk aan het laden of wissen van alle elementen die ze bezitten.

4.3.7.1. configuration

Een voorbeeld van een configuration is dus weergegeven a.d.h.v. de figuur op bijlage 4 van deze studie. Hierin zijn een aantal functieblokken en programma's aanwezig, die gedefinieerd moeten worden op de manier zoals dat in paragraaf 4.3.5. van deze studie aan de orde gekomen is. In figuur 19a van de IEC-1131-3 norm is de definitie van de programma's en functieblokken gegeven in instructies.

Om echter een configuration te definiëren, zijn er weer een aantal features nodig, die in tabel 49 van de IEC-1131-3 norm weergegeven zijn.

4.3.7.2. tasks

Zoals dus al naar voren gekomen is, dient een task voor de executie van een programma opbouwunit, waarbij er een onderscheid gemaakt kan worden tussen een periodieke executie en een executie op een opgaande flank van een boolean variabele.

Een task en de samenhang tussen programma opbouwunits kan grafisch en textueel weergegeven worden, zoals dat in tabel 50 op bladzijde 111-114 van de IEC-1131-3 norm weergegeven is.

De controle over programma opbouwunits d.m.v. enabled tasks moet volgens een aantal regels geschieden:

- de betreffende programma opbouwunit is gereed gemaakt voor executie op elke opgaande flank van de SINGLE input van de bijbehorende task;
- is de INTERVAL input niet gelijk aan nul en de SINGLE input wel, dan is de betreffende programma opbouwunit gereed voor periodieke executie;
- de PRIORITY input geeft de mogelijkheid om prioriteit aan het gereed maken van een programma opbouwunit te geven. Nul is de hoogste prioriteit, afnemend met de hoogte van het getal. Voor de prioriteit van een programma opbouwunit (prioriteit van de TASK), kan preemptive of non-preemptive scheduling gebruikt worden:

non-preemptive:

de uitvoer mogelijkheid is pas beschikbaar voor een resource als de uitvoer van een andere programma opbouwunit of van een operating system functie gereed is. Zodra de uitvoer mogelijk is, dan zal de programma opbouwunit met de hoogste prioriteit beginnen met de uitvoer. Als er meerdere programma opbouwunits met de hoogste prioriteit wachten, dan zal de programma opbouwunit met de langste wachttijd uitgevoerd worden.

preemptive:

als een programma opbouwunit beschikbaar is, kan het de executie van een programma opbouwunit met lagere prioriteit in dezelfde resource onderbreken, totdat de betreffende opbouw unit zijn handelingen uitgevoerd heeft. Zo'n programma opbouwunit kan geen andere programma opbouwunit met dezelfde of hogere prioriteit beïnvloeden.

opmerking:

als gevolg van de verschillende prioriteiten zal een programma opbouwunit niet altijd direct uitgevoerd worden, zodra de betreffende unit beschikbaar wordt. Toch is hieraan een 'deadline' gebonden en zullen ze alle uitgevoerd moeten zijn, voordat ze weer beschikbaar gesteld worden voor uitvoer. De fabrikant dient hierover gegevens beschikbaar te stellen;

- Een programma (opbouw unit) dat geen TASK heeft, heeft de laagste systeemprioriteit. Zulke programma's worden beschikbaar gesteld, zodra de betreffende resource gestart wordt en worden opnieuw beschikbaar gesteld zodra hun uitvoer afgelopen is;

- Als een functieblok gekoppeld is aan een task, dan zal de uitvoer van dit functieblok onder de exclusieve controle van deze task zijn. Dit is dan onafhankelijk van de evolutieregels van de betreffende programma opbouwunit waarin dit functie blok gedefinieerd is;

- Is een functieblok niet gekoppeld aan een task, dan zal het de normale regels voor evolutie volgen binnen de programma opbouwunit waarin het is gedefinieerd;

- De uitvoer van functieblokken binnen een programma moeten gesynchroniseerd worden, om er voor te zorgen dat er geen data gemixed wordt. Hieraan zijn een aantal regels gebonden:

* als een functieblok (nr 1) meer dan één input van een ander functieblok (nr 2) ontvangt, dan moet het resultaat van functieblok (nr 1) verkregen zijn door het resultaat van één berekening van functieblok (nr 2). Dus niet de eerste ingang van berekening Z en de tweede ingang van berekening Z+1;

* als twee of meer functieblokken het resultaat van een voorgaand functieblok ontvangen en deze twee functieblokken gerelateerd zijn aan dezelfde TASK, dan moeten de ingang van de destinationblokken op het moment van uitvoer altijd dezelfde waarden bevatten.

Deze regels worden geïllustreerd aan de hand van tabel 50 en figuur 21 van de IEC-1131-3 norm op bladzijde 111 t/m 117.

4.4 TEKSTUELE PROGRAMMEERTAAL ELEMENTEN

In paragraaf 3.2 van de IEC-1131-3 norm zijn een tweetal tekstuele programmeertalen beschreven te weten: Instruction List (IL) en Structured Text (ST). Natuurlijk kunnen de SFC-elementen in combinatie met deze twee talen gebruikt worden.

4.4.1 programmeerelementen

In paragraaf 4.3 van deze studie zijn er een aantal programmeerelementen aan de orde gekomen. Om deze elementen te kunnen programmeren, zijn er een aantal tekstuele commando's nodig, die bij wijze van een korte samenvatting hier weergegeven zijn:

TYPE...END_TYPE	2.3.3
VAR...END_VAR	2.4.3
VAR_INPUT...END_VAR	2.4.3
VAR_OUTPUT...END_VAR	2.4.3
VAR_IN_OUT...END_VAR	2.4.3
VAR_EXTERNAL...END_VAR	2.4.3
FUNCTION...END_FUNCTION	2.5.1.3
FUNCTION_BLOCK...END_FUNCTION_BLOCK	2.5.2.2
PROGRAM...END_PROGRAM	2.5.3
STEP...END_STEP	2.6.2
TRANSITION...END_TRANSITION	2.6.3
ACTION...END_ACTION	2.6.4

De tweede kolom geeft een verwijzing naar de betreffende paragraaf van de IEC-1131-3 norm.

4.4.2 Instruction List

In tabel 51 op bladzijde 119 van de IEC-1131-3 norm, is te zien dat de IL bestaat uit een sequentie van instructies. Elke instructie begint op een nieuwe regel en bevat een operator met een mogelijke modifier en afhankelijk van de operator één of meerdere operands gescheiden door komma's.

Deze operands kunnen bestaan uit de datapresentaties of variabelen zoals deze reeds besproken zijn en natuurlijk in- of uitgangen.

Een instructie kan voorafgegaan worden door een label, gevolgd door een dubbele punt(:). Als laatste elementen in een regel kan er commentaar geplaatst worden. Tussen een tweetal instructieregels, kan er een open ruimte (regel) overgelaten worden.

operators, modifiers en operands:

In tabel 52 op bladzijde 120 van de IEC-1131-3 norm is een overzicht van de (standaard) operators en hun mogelijke modifiers en operands gegeven.

De semantics voor deze operators zijn:

result := result OP operand

waarbij de waarde van de variabele result vervangen wordt door het resultaat van de handeling van de operator met de huidige waarde van result en de waarde van de operand. Zo zal bijvoorbeeld het IL-commando AND %IX1 geïnterpreteerd worden als:

result := result AND %IX1

Als de modifier N toegevoegd wordt, krijgen we ANDN %IX1 dat geïnterpreteerd zal worden als:

result := result AND NOT %IX1

Met haakje openen en haakje sluiten, kunnen gescheiden berekeningen aangegeven worden. Zo zal

```
AND( %IX1  
OR %IX2  
)
```

geïnterpreteerd worden als:

result := result AND (%IX1 OR %IX2)

Door toevoeging van de modifier C zal de instructie alleen uitgevoerd worden als het huidige resultaat de boolean waarde '1' heeft. Of in combinatie met de modifier N als het resultaat '0' is.

functies en functieblokken:

Functies kunnen aangeroepen worden door de functienaam in het operatorveld te plaatsen. Het huidige resultaat zal dan gebruikt worden als het eerste argument van de functie. In het

operandveld kan hieraan het nodige toegevoegd worden.

Functieblokken kunnen conditioneel of niet-conditioneel aangeroepen worden m.b.v. het commando CAL. Hierin zijn een drietal vormen te onderscheiden, wat weergegeven wordt in tabel 53 op bladzijde 121 van de IEC-1131-3 norm.

4.4.3 Structured Text

In deze paragraaf zal kort de opbouw van structured text besproken worden. Structured text bestaat uit een aantal expressions en statements. De Structured Text (ST) is te vergelijken met een programmeertaal als PASCAL en C.

4.4.3.1 expressions

Een expression is een constructie die bij uitvoer resulteert in een waarde die overeenkomt met één van de reeds besproken datatypes. Ze zijn opgebouwd uit operators en operands. Bijvoorbeeld:

$A+B$ waarin A en B de operands in de vorm van een variabele zijn en + is de operator.

Een operand is een literal (getal), variabele, functie aanroep of een andere expressie. De operators zijn samengevat in tabel 55 op bladzijde 123 van de IEC-1131-3 norm en bestaan uit een aantal bewerkingen met verschillende prioriteiten.

Het resultaat van een expression wordt verkregen door het sequentieel uitwerken van de combinatie operands/operators aan de hand van de in tabel 55 gegeven prioriteit van de operators.

De operator met de hoogste prioriteit wordt het eerst toegepast, gevolgd door de operator van een één stap lagere prioriteit. Dit wordt herhaald totdat de gehele expressie uitgevoerd is. Als hierin operators met gelijke prioriteit in voorkomen, dan zullen deze van links naar rechts uitgevoerd worden. Als voorbeeld zijn de integer variabelen A, B, C en D met respectievelijk de waarden 1, 2, 3, -4 gegeven:

$A+B-C*ABS(D)$ geeft: $1+2-3*4=-9$

$(A+B-C)*ABS(D)$ geeft: $(1+2-3)*4=0$

Als een operator 2 operands heeft, die elk weer bestaan uit het aanroepen van een functie, dan zal de linkse functie het eerst berekend worden, daarna de rechtse en tot slot zal de operator aan de orde komen. Bijvoorbeeld:

$\sin(A)*\cos(B)$ eerst wordt $\sin(A)$ berekend, daarna $\cos(B)$ en tot slot het produkt van deze twee resultaten.

Boolean expressions worden berekend voor zover ze berekend behoeven te worden voor een goede uitkomst. Bijvoorbeeld:

$(A>B)\&(C<D)$ als $A<B$ is, dan wordt er verder niet meer gerekend, omdat de uitkomst van deze expressie toch nul wordt. Dus $C<D$ wordt overgeslagen.

Zoals hier al te voorschijn gekomen is, kunnen functies dus ook aangeroepen worden.

Tussen haakjes kunnen een aantal argumenten aan een functie aangevoerd worden.
Bijvoorbeeld:

MAX(X,Y) waarbij de argumenten X en Y een tweetal ingangsvARIABLEN zijn.

4.4.3.2 statements

In tabel 56 op bladzijde 124 van de IEC-1131-3 norm zijn de mogelijke statements uit de Structured Text (ST) weergegeven. Hierin zijn een viertal verschillende statements te herkennen, die kort besproken zullen worden:

assignment statements:

Dit statement overschrijft de waarde van een variabele met het resultaat van een expressie. Het assignment statement is opgebouwd uit een referentievariabele aan de linkerkant, de assignment operator '=' en aan de rechterkant een expressie.

Het assignment statement wordt ook gebruikt om de waarde te definiëren die terugkomt van een functie. De functienaam wordt aan de linkerkant van de assignmentoperator geplaatst en aan de rechterkant een expressie. Dit moet wel binnen de definitie van de functie gebeuren. Dit is weergegeven in afbeelding 6 op bladzijde 48 van de IEC-1131-3 norm.

functie- en functieblok controle statements:

Deze statements worden gebruikt om functies en functieblokken aan te roepen, maar ook om terug te keren naar het deel van het programma waar de betreffende functie aangeroepen wordt.

Functies worden aangeroepen als een onderdeel van een expressie en functieblokken d.m.v. een statement, bestaande uit de naam met een door haken gemarkeerde lijst van input variabelen (zie tabel 55).

M.b.v. het RETURN statement kan vroegtijdig uit een functie of functieblok gestapt worden.

selectie statements:

Dit zijn de IF- of CASE statements, die één of een groep bijbehorende statements selecteren aan de hand van een voorwaarde.

Het IF statement laat een groep statements alleen uitvoeren als de bijbehorende boolean expressie de waarde '1' heeft. Is dit niet het geval, dan gebeurt er niets, of het groepje statements dat volgt op ELSE of ELSEIF wordt uitgevoerd.

het CASE statement bestaat uit een selector variabele van het datatype INT. Een groep statements wordt gemerkt d.m.v. één of meerdere integers of bereiken van integers. Indien de selector de waarde van een van de gemerkte statements heeft, dan zal dit statement aangesproken worden. Als de waarde van de selector niet overeenstemt met één van de gemerkte statements, dan zullen de statements die op ELSE volgen uitgevoerd worden.

Indien er echter geen ELSE statements zijn, dan wordt er niets uitgevoerd.

Iteration statements:

Dit statement zorgt voor een herhaling van een aantal statements voor een vooraf vastgesteld aantal keer (FOR-statement) of voor een onbekend aantal malen (WHILE of REPEAT).

M.b.v. het EXIT-statement kan hier voortijdig uitgesprongen worden naar het volgende statement na de eerste loopterminator (zie figuur 22 bladzijde 126 van de IEC-1131-3 norm).

Dit wordt verder in paragraaf 3.3.2.4 van de IEC-1131-3 norm uitgelegd.

4.5 GRAFISCHE PROGRAMMEERTAAL ELEMENTEN

In de IEC-1131-3 norm zijn een tweetal grafische programmeer talen beschreven te weten: Ladder Diagram (LD) en Function Block Diagram (FBD). Natuurlijk kunnen de SFC-elementen in combinatie met deze twee talen gebruikt worden.

4.5.1 gemeenschappelijke elementen

Dit zijn een aantal elementen die voor zowel de LD-, FBD- en SFC-programmeertaal gelden.

Lijnen en blokken:

Alle grafische taalelementen (meestal blokken) worden m.b.v. karakters uit de ISO-646 karakterset getekend zoals dit weergegeven is in tabel 57 op bladzijde 129 van de IEC-1131-3 norm.

Lijnen kunnen verlengd worden d.m.v. het gebruik van 'connectors', die echter niet als geheugen element mogen gaan dienen. Daarom moet er een foutmelding gegeven worden, indien er voor de 'connector' een naam gebruikt wordt, die op een andere plaats in de programma opbouwunit al voorkomt.

flowrichting in netwerken:

Netwerken zijn gedefinieerd als een set van met elkaar verbonden grafische elementen. Bij LD worden de linker- en rechter lijn niet meegerekend. Er moeten voorzieningen zijn getroffen, dat elk netwerk of groepje netwerken gemarkeerd kan worden met een label in de vorm van een 'unsigned decimal integer'.

Grafische programmeertalen worden gebruikt om de flow aan te geven van een mogelijk (elektrisch) medium door één of meerdere netwerken volgens een bepaalde strategie (eisen van de klant). Voorbeelden zijn:

- 'Power Flow', identiek aan de flow van elektrische energie in een elektromagnetisch relais systeem (vooral voor LD);
- 'Signal Flow', identiek aan de flow van signalen tussen signaal bewerkings elementen (vooral voor FBD);
- 'Activity Flow', identiek aan de flow tussen organisatieelementen of tussen de toestanden

van een elektromechanische sequencer (vooral SFC).

Het mogelijke medium zal zich voortplanten tussen de elementen van het betreffende netwerk volgens een aantal regels:

- 'Power Flow' in een LD-taal moet altijd van links naar rechts zijn;
- 'Signal Flow' in een FBD-taal moet altijd van de uitgang (rechts) van een blok, naar de ingang (links) van een volgend blok;
- 'Activity Flow' in een SFC-taal moet altijd van de onderkant van een toestand, via de overgangsvoorwaarde, naar de bovenkant van een volgende toestand.

Verplaatsingen binnen netwerken:

De volgorde waarin netwerken en netwerk-elementen afgehandeld worden, hoeft natuurlijk niet gelijk te zijn aan de volgorde waarin ze genummerd zijn. Ook hoeven de netwerken nog niet allemaal doorlopen te zijn, voordat een ander netwerk weer herhaald wordt. De afwerking dient echter wel aan een aantal functionele regels te voldoen:

- een element mag pas een resultaat weergeven, indien alle inputs van het element verwerkt zijn (status stabiel);
- het resultaat van een element is pas compleet, als al de uitgangen van het element een stabiele waarde hebben;
- datzelfde geldt, als het gaat om een uitvoer controle element;
- de volgorde waarin netwerken afgehandeld worden staat beschreven in paragraaf 4.2.6 en 4.3.3 van de IEC-1131-3 norm voor respectievelijk LD- en FBD programmeertaal elementen.

Een 'feedback' in een netwerk ontstaat, als de uitgang van een functie(blok) wordt gekoppeld aan de ingang van een ander functie(blok) en vervolgens de waarde weer meeneemt in het netwerk. Dit is dan een zogenaamde 'feedbackvariabele'. Een voorbeeld hiervan wordt gegeven in figuur 23 op bladzijde 131 van de IEC-1131-3 norm.

Een 'feedback' moet aan de volgende regels voldoen:

- bepaalde loops, zoals in figuur 23a mogen alleen in FBD-taal voorkomen;
- het moet mogelijk zijn voor de gebruiker om de volgorde van uitvoer te bepalen aan de hand van een 'feedbackloop' volgens figuur 23b;
- 'feedbackvariabelen' moeten op de normale manier gedefinieerd worden, met een initiële waarde;
- als de waarde van de 'feedbackvariabele' bij een bewerking van een element veranderd is, dan zal hij deze waarde moeten blijven behouden totdat het volgende resultaat beschikbaar komt.

uitvoer controle elementen:

Verplaatsing van programmacontrole (programma uitvoer) kan gerealiseerd worden m.b.v.

een aantal grafische elementen (LD en FBD). Deze elementen zijn weergegeven in tabel 58 van de IEC-1131-3 norm en hebben als functie, het springen binnen een programma netwerk.

Dit zal verder hier niet uitgewerkt worden.

4.5.2 LD-taal

Een LD-programma geeft de mogelijkheid om een programma te testen en te wijzigen d.m.v. gestandaardiseerde symbolen. Deze symbolen zijn op zodanige manier opgebouwd, dat dit te vergelijken is met een logisch relais diagram. LD-netwerken zijn aan de linker- en rechterzijde afgesloten d.m.v. een power rails.

Power-rails:

Dit zijn twee verticale lijnen waartussen de LD-symbolen geplaatst worden.

Verbindings elementen:

Dit zijn horizontale- of verticale lijnen, die de boolean waarde '1' of '0' kunnen hebben. Deze status is te vergelijken met de power flow.

De status van de linker power-rails is ON als deze verbonden is met een actieve stap in het SFC. De status van de rechter power-rail is niet gedefinieerd.

Een horizontale verbinding verzend de status van het element aan zijn onmiddellijke linkerkant naar het element aan zijn onmiddellijke rechterkant.

Een verticale verbinding bestaat uit een verticale lijn, met één of meerdere horizontale verbindingselementen aan beide zijden. De verticale verbinding vertegenwoordigd de OF-functie van de horizontale verbindingen aan de linkerkant. De status kan weer AAN- of UIT zijn.

De status van de verticale lijn zal verzonden worden naar alle horizontale verbindingen aan de rechter zijde. Niet andersom.

Contacten:

Een contact is een element dat een status aan de horizontale verbinding aan zijn rechterzijde geeft. Het vertegenwoordigd de AND-functie met de mogelijke contacten aan zijn linkerzijde.

Uitgangen (spoelen):

Een uitgang kopieert de status van de verbinding aan de linkerkant naar de rechterkant, zonder enige wijziging en tevens naar de betreffende variabele. Hierin zijn een aantal variaties mogelijk, zoals:

- status van de linker verbinding naar betreffende variabele (zonder SET);
- de inverse status van de linker verbinding naar de betreffende variabele (zonder SET);
- status van de linker verbinding naar betreffende variabele (met SET);
- RESET van deze variabele;

- status van de linker verbinding naar geheugen plaats (zonder SET);
- status van de linker verbinding naar geheugen plaats (met SET);
- RESET van deze geheugen plaats;
- gedurende één scanpuls wordt een betreffende variabele hoog na een opgaande- of neergaande flank van de linker verbinding.

Functies en functieblokken:

Dit is in paragraaf 2 van de IEC-1131-3 norm aan de orde gekomen, waarbij aan de volgende uitzonderingen gedacht moet worden:

- actuele parameter verbinding mogen zichtbaar gemaakt worden door de bijbehorende data of variabele aan de buitenkant van het blok te schrijven en de eigenlijke parameter naam aan de binnenzijde van het blok;

- er moet tenminste één boolean in- en output zichtbaar zijn, om de power flow door het blok toe te laten.

Volgorde van verplaatsing binnen LD-netwerken:

Bij een programma opbouwunit geschreven in LD-taal, zullen de netwerken van top tot bodem doorlopen worden. Behalve wanneer er gebruik gemaakt wordt van uitvoer controle elementen (zie vorige paragraaf).

De grafische symbolen voor de LD-taal zijn te vinden in tabel 61 en 62 op bladzijde 136 en 137 van de IEC-1131-3 norm.

4.5.3 FBD-taal

In deze paragraaf komt tot slot de FBD-taal aan de orde. Deze programmeertaal komt voor zover dat mogelijk is, overeen met de IEC-617 standaard, deel 12.

In paragraaf 4.5.1 van deze norm zijn enkele elementen besproken die ook van toepassing zijn voor de FBD-taal. Verder zijn in hoofdstuk 2 van de IEC-1131-3 norm een aantal voorbeelden van FBD elementen terug te vinden en bijlage F van deze norm geeft nog een aantal voorbeelden.

Verder geldt dat een FBD programma, dat een netwerk pas uitgevoerd zal worden, nadat het voorgaande netwerk (waarvan uitgangen als input gebruikt wordt) volledig afgewerkt is.

4.6 BIJLAGE's IEC-1131-3 NORM

A: Deze bijlage geeft de normering voor syntax- en semantic regels.

B: Als een compiler in staat is om de syntax-regels die in deze bijlage vermeld worden te herkennen, dan zal deze compiler ook in staat zijn om alle syntax-regels en tekstuele programmeertaal implementaties te herkennen, die volgens de regels van deze norm geschreven zijn.

Dit zijn de zogenaamde 'formal specifications' van programmeertalen.

C: Bijlage C geeft een overzicht van alle 'delimiters' en 'keywords'. Dit zijn alle symbolen en gereserveerde woorden die in de normering van de IEC-1131-3 norm voorkomen.

D: Bijlage D geeft een aantal implementatie afhankelijke parameters weer. Bijvoorbeeld het maximale aantal acties per stap in SFC programmering.

E: In deze paragraaf zijn alle foutcondities die volgens de IEC-1131-3 norm voor moeten kunnen komen weergegeven.

F: Tot slot zijn in bijlage F een aantal voorbeelden van mogelijke programma's gegeven. Deze programma's zijn volgens de IEC-1131-3 norm geschreven en bieden veel informatie over de toepassing van alle mogelijke regels die in deze norm aan de orde gekomen zijn.

BIJLAGE 1
BIJLAGE 2
BIJLAGE 3
BIJLAGE 4

VERKLARENDE WOORDENLIJST

PLC: Programmable Logic Controller
PADTs: Programming And Debugging Tools
MMI: Man Machine Interface
MMS: Manufacturing Message System
IEV: International Elektrotechnical Vocabulary
RFI: Radio Frequence I...
RIO's: Remote In- and Output Stations
CPU: Central Processing Unit
MPU: Main Processing Unit
SFC: Sequential Flow Chart
ST: Structured Text
LD: Ladder Diagram
FBD: Function Block Diagram
IL: Instruction List