

Hoofdstuk 102

PLC

Doelstellingen

1. Toelichten hoe de PLC juist werkt
2. Basisprincipes van de grafcet toelichten

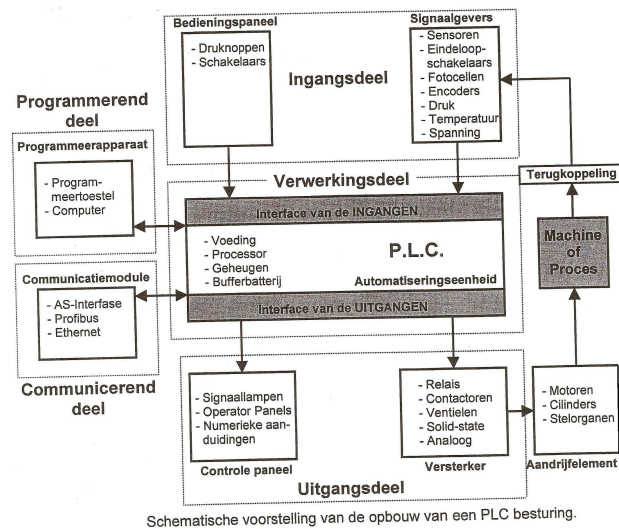
In dit hoofdstuk bespreken we de werking van de Programmable Logic Controller. Dit wil zeggen dat we van deze machine de hardware gaan bekijken, met andere woorden gaan we eigenlijk de werking van een microcomputer bestuderen. Bijbehorend bekijken we ook hoe de de communicatie tussen de verschillende delen van deze machine en zulke machines onderling in zijn werk gaat, dit zijn de zogenaamde bussystemen. De software, met andere woorden het programmeren van een PLC gebeurt hands-on in het labo en wordt in dit hoofdstuk enkel bekeken vanuit de basis namelijk de grafcet.

102.1 Algemene beschrijving

Eigenlijk doet een PLC hetzelfde als een relaissturing, namelijk:

1. Ingangsinformatie opnemen via bedieningspaneel en signaalgevers
2. Deze ingangsinformatie combineren volgens een bepaalde logica, die vastligt in het gebruikersprogramma
3. Hieruit uitgangsbevelen afleiden en naar buiten sturen

Onderstaande figuur geeft een schematische voorstelling weer van een PLC.



Figuur 102.1: bron: cursus PLC, APHogeschool

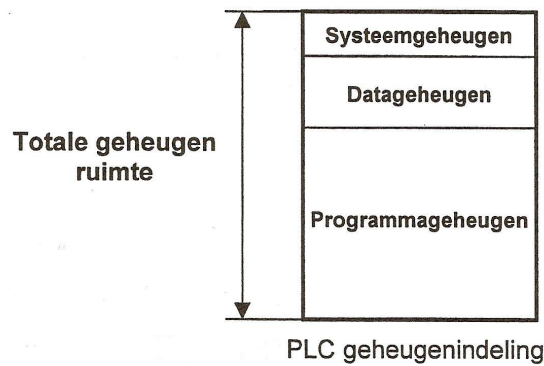
De pijlen geven de zin aan van de informatieuitwisseling tussen het ingangsideel, het uitgangsideel, het sturend deel en het programmerend deel.

102.1.1 Algemene kenmerken

1. Het ingangsideel: Hierin gebeurt het opnemen van de ingangsinformatie via de signaalgevers van de bedieningsconsole en de detectoren in de machine, die aangesloten worden op de klemmenstrook van de ingangsmodule. Dit geldt ook voor andere ingangsinformatie zoals meetgegevens of het bereik van alarmsituatie's.
2. Het verwerkend deel: De verwerking gebeurt door een processor. Deze bestaat uit een centrale verwerkingseenheid (CPU), een intern systeemgeheugen (ROM), het datageheugen en een programmeergeheugen (RAM, EPROM, ...).
 Het intern systeemgeheugen bevat de systeemsoftware, zoals de instructieset, die nodig zijn voor het functioneren van het systeem.
 Het programmeergeheugen bevat het specifieke stuurprogramma dat opgesteld werd door de gebruiker en waarin de besturingslogica van de machine of het proces vastligt. De CPU haalt een programmaregel op uit het programmeergeheugen en weet alzo op welke wijze de ingangsinformatie die aanwezig is op de ingangen moet worden verwerkt om de gewenste commando's te verkrijgen op de uitgangen.
 Het datageheugen wordt gebruikt voor de opslag van tijdelijke gegevens.

3. Het communicatiedeel: De communicatiemodule is het middelpunt van het netwerk en transporteert informatie van de ene deelnemer naar de andere. Deze verbindingen zijn meestal bidirectioneel. Een verbinding tussen twee deelnemers wordt een point-to-point verbinding (PPI) genoemd en een netwerkverbinding tussen meerdere deelnemers wordt een multipoint verbinding (MPI) genoemd.
4. Het programmerend deel: Het gebruikersprogramma wordt opgesteld met een programmeerapparaat, al dan niet merkgebonden.

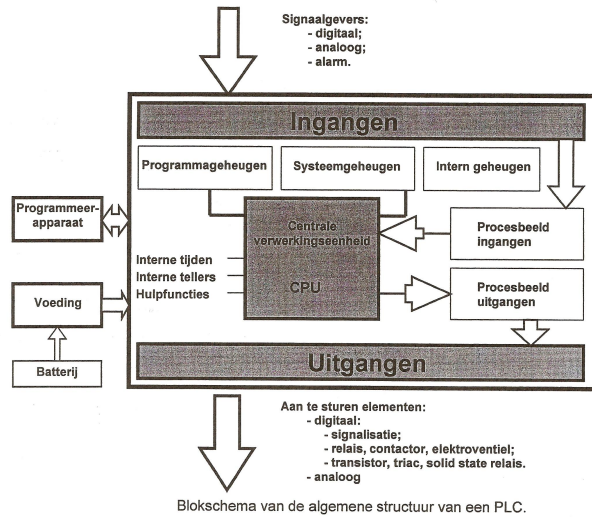
Onderstaande figuur geeft de PLC-geheugenindeling



Figuur 102.2: bron: cursus PLC, APHogeschool

102.1.2 Algemene structuur

Onderstaande figuur geeft een schematische opbouw van de PLC



Figuur 102.3: bron: cursus PLC, APHogeschool

Er wordt een onderscheid gemaakt tussen modulaire en niet modulaire opgebouwde PLC's.

- Niet-modulaire PLC: hebben een beperkt aantal in- en uitgangen en zijn als één geheel opgebouwd. Het nadeel is dat type beperkt uit te breiden is.
- Modulaire PLC: Elk deel is een apart blok en het type en aantal blokken kan worden gekozen. Elk blok wordt gerealiseerd door een specifieke module. De PLC is dan samengesteld uit een aantal functionele blokken die geordend zijn rond een communicatiekanaal: *de interne bus*.

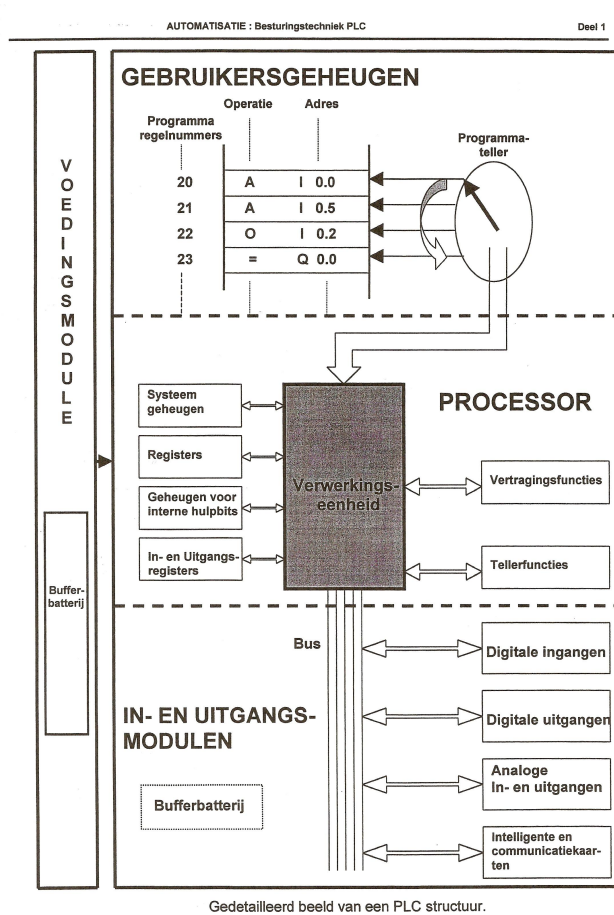
De bus bestaat uit meerdere parallelle signaalleidingen

- Adresbus: Deze gaat de verschillende adressen op de verschillende kaarten aanspreken.
- Databus: Hiermee worden gegevens afkomstig van ingangskarten of gegevens die naar uitgangskarten moeten, overgedragen.
- Sturingsbus: Hiermee worden stuur- en bewakingssignalen overgedragen.

102.2 Werking

102.2.1 Algemeen

Onderstaande figuur geeft een meer gedetailleerd beeld van de PLC structuur die toelaat de werking ervan te begrijpen.



Figuur 102.4: bron: cursus PLC, APHogeschool

Het gebruikersprogramma bestaat uit een aantal programmaregels en werd door het programmeerapparaat in het gebruikersgeheugen geschreven.

Elke regel bevat een instructie die op te delen is in:

- Een *operatiecode* die zegt welke verwerking er moet gebeuren.
- De *operand* of *adres* die zegt met welke data die verwerking moet gebeuren.

De processor bevat volgende functies

- De *besturingseenheid* die de verwerking van de informatie regelt op basis van de instructies die in het gebruikersprogramma geschreven staat
- De *accumulatoren* of *registers* waarin het tussenresultaat van een bewerking wordt opgeslagen. Bij het uitvoeren van een volgende bewerking wordt de inhoud overschreven door het nieuwe resultaat.
- Het *geheugen* voor de interne hulpbits dat door de besturingseenheid gebruikt wordt om de tussentijdse verwerkingsresultaten te onthouden.
- Het *ingang- en uitgangsregister* dat eveneens een geheugen is en de toestand van in- en uitgangen onthoudt tijdens één programmacyclus.
- De *tijd- en tellerfuncties* of *counters* zijn softwarefuncties die door de fabricant werden voorzien met als doel te beschikken over middelen die overeenkomen met tijdsfuncties en tellers uit de bedrade technologie.

De fabricant voorziet twee soorten geheugens

1. Remanent: de logische toestand na een spanningsonderbreking is gelijk aan deze van *voor* de spanningsonderbreking.
2. Niet remanent: de logische toestand wordt na spanningsonderbreking steeds op nul gebracht.

102.2.2 De werking van het systeem

Na het inschakelen van de netspanning geeft de besturingseenheid een stuurpuls waardoor de niet-remanente tijdsfuncties, tellerfuncties en interne hulpbits, de accu's en de in- en uitgangregisters op nul worden gezet. We veronderstellen een cyclisch programmaverloop, wat meestal ook zo is. Nu leest de besturingseenheid de signaaltoestanden die zich in het ingangsregister bevinden. Het programma wordt doorlopen, regel per regel, en verwerkt in overeenstemming met de instructies. Indien op basis van het programma en de bemonsterde toestand van de ingangen, een uitgang hoog wordt zal de besturingseenheid deze toestand vastleggen in het uitgangsregister. Op het einde van een cyclus transfereert de besturingseenheid de volledige inhoud van het uitgangsregister naar de buitenwereld via de uitgangsmodulen.

Deze procedure wordt enkel onderbroken bij een fout of het op stop schakelen van de PLC.

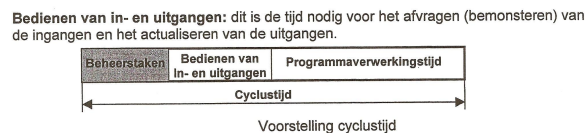
102.2.3 Cyclustijd

De cyclustijd is de tijd die de processor nodig heeft om een volledig gebruikersprogramma te verwerken. Hierin is begrepen het bemonsteren van de ingangen en het actualiseren van de uitgangen.

Hierin zijn drie elementen van belang.

1. Beheerstaken: De systeemsoftware moet zorgen voor de organisatie van de verschillende eenheden en de communicatie.
2. Programmaverwerkingstijd: De tijd nodig om de volledige programmalogica te verwerken. Hier moet men opletten met het grote verschil in verwerkingstijd tussen bit- en woordinstructies.
3. Bedienen van in- en uitgangen: De tijd nodig voor afvragen van ingangen en actualiseren van uitgangen.

Onderstaande figuur geeft een voorstelling van de cyclustijd. Deze ligt meestal in de grootteorde van microseconden.



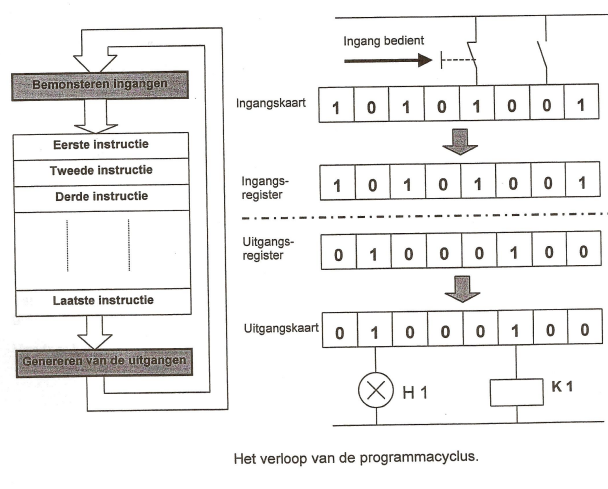
Figuur 102.5: bron: cursus PLC, APHogeschool

102.3 Programmaverwerking

102.3.1 Programmaverwerking synchroon

Een programmacycclus bevat drie fasen:

1. Alleingangssignalen worden in het ingangsregister gebufferd
2. Het programma wordt verwerkt met de informatie uit dit ingangsregister
3. Tijdens de programmaverwerking van het gebruikersprogramma worden de stuursignalen voor de uitgangen in een uitgangsregister opgeslagen



Figuur 102.6: bron: cursus PLC, APHogeschool

Pas na het einde van de programmacycclus wordt de inhoud van het uitgangsregister doorgestuurd naar de eigenlijke uitgangsmodule. Een programma wordt cyclisch doorlopen en men moet hier rekening mee houden in de volgorde van de instructies.

102.3.2 Programmaverwerking asynchroon

In dit geval is er geen verband tussen lezen van ingangen, het actualiseren van de uitgangen en programmaverwerking.

Om betrouwbaar en optimaal te kunnen programmeren moet er een goed inzicht bestaan op de wijze van programmaverwerking van het gehanteerde systeem.

102.3.3 Reactietijd

De reactietijd van een PLC kan gedefinieerd worden als de tijd tussen de verandering van de ingang, die door een sensor op de ingang wordt gemeten, en de verandering van het overeenkomstig uitgangssignaal op een actuator.

Praktische reactietijd

In praktijk is de reactietijd de som van

1. De eigentijd van de sensor

2. De vertragingstijd van de ingangsmodule
3. De systeemtijd van de PLC
4. De cyclustijd van de PLC
5. De vertragingstijd van de uitgangsmodule
6. De eigentijd van de actuator

Een PLC heeft een bewakingssysteem die bewaakt of de cyclustijd een bepaalde waarde niet overschrijdt. Als deze tijd overschreden wordt zullen alle uitgangen op nul gezet worden.

102.4 Componenten

Meestal vinden we volgende componenten terug

- Voedingsmodule
- Centrale eenheid
- Digitale in- en uitvoermodule

102.4.1 Voedingsmodule

De voornaamste functie is de beschikbare spanning te transformeren die nodig is voor de werking van de elektronische componenten van de PLC.

Er zijn meestal twee gescheiden stroomkringen te onderscheiden

- De stroomkring voor de voeding van de processor en de uitbreidingsmodules
- De stroomkring voor de uitwendige belasting

102.4.2 CPU

Dit is het hart van de PLC en bestaat uit de centrale verwerkingseenheid en de geheugen. Om de CPU te kunnen programmeren zijn er één of meerdere aansluitingen voorzien om te kunnen programmeren met het programmeerapparaat (PC), een veldbus of andere CPU's.

Het geheugen is opgebouwd uit meerdere geheugenlocatie's die elk de toestand 0 of 1 onthouden. De meest toegepaste geheugencomponenten zijn de RAM, ROM, PROM, EPROM, EEPROM, FEPRM.

102.4.3 Signaalmodules

Deze zijn de interface tussen het de besturen proces en de PLC. Dit zijn zowel digitale als analoge modules.

De beide stroomkringen kunnen op verschillende manieren zijn opgebouwd in functie van de potentiaalbinding tussen beide stroomkringen.

- Potentiaal gebonden modules: er is een centraal aardingspunt en een gemeenschappelijke massa.
- Potentiaal vrije modules: er is geen potentiaalverbinding tussen de verschillende modules

102.5 Programmeertechnieken

Er bestaan drie soorten methodes om combinatorische oplossingen te beschrijven namelijk

- Boolse uitdrukking
- Ladderdiagram
- Logische schema's

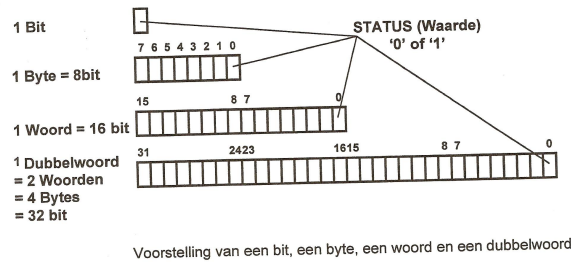
Worden de problemen ingewikkelder wordt meestal gebruik gemaakt van de GRAFCET. Binnen de IEC norm worden vijf programmeermethoden aangegeven namelijk

- Instructielijst (STL of AWL)
- Ladderdiagram (LD of KOP)
- Functieblokkendiagram (FDB of FUP)
- Gestructureerde tekst (ST of SCL)
- Sequentieel functiediagram (SFC)

Verder moet men in de programmeertechnieken ook rekening houden met het onderscheid tussen bit, byte, woord en dubbelwoord.

- bit: kleinste eenheid van het binair systeem, 0 of 1.
- byte: 8 bits
- woord: 16 bits of twee bytes
- dubbelwoord: 32 bits

Onderstaande figuur geeft deze begrippen weer

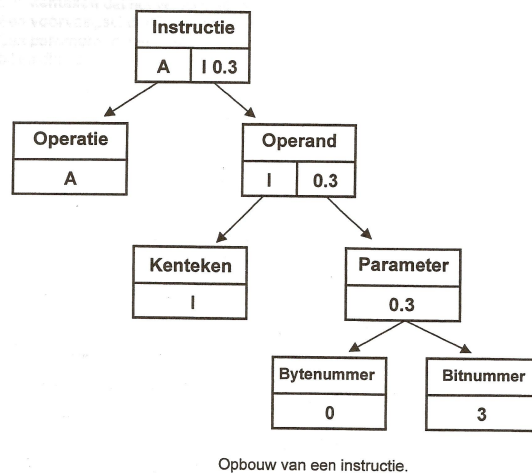


Figuur 102.7: bron: cursus PLC, APHogeschool

102.5.1 Instructie

In een programma worden ingangen afgevraagd en uitgangen aangestuurd door middel van instructies. Een instructie verwijst dus naar deze in- en uitgangen en elke in- en uitgang moet dus worden benoemd. Dit gebeurt door er een parameter aan toe te kennen.

Let op men moet een onderscheid maken tussen bit-, byte-, woord- en dubbelwoordadres.



Figuur 102.8: bron: cursus PLC, APHogeschool

Een **merker** is een intern hulprelais en kan als tussenresultaat worden opgeslagen. De toestand van een merker houdt dezelfde waarde in één cyclus en wordt ter beschikking gesteld voor de opvolgende verbindingen. Een merker kan in een programma vaak willekeurig als opener of als sluiters opgevraagd worden.

Bij de meeste fabrikanten bestaat het adres uit

- Een kenteken dat het verschil aangeeft de in- en uitgangen
- Een voorvoegsel dat de grootte van het kenteken aangeeft
- Een parameter die een cijfercombinatie aangeeft voor de betreffende in- of uitgangsbits en die gescheiden wordt door een punt

Soort	Type	Opmerking
I	Ingang	Procesbeeld van de digitale ingangen van de PLC
Q	Uitgang	Procesbeeld van de digitale uitgangen van de PLC
M	Geheugen	Geheugengebied dat door de PLC kan gebruikt worden voor tussen resultaten

Kenteken aanduiding	Grootte (bit)	Opmerking
X	1	Bit
(geen)	1	Bit
B	8	Byte
W	16	Woord
D	32	Dubbelwoord
L	64	Lang woord

Overzicht van de gebruikte kentekens en voorvoegsels

Figuur 102.9: bron: cursus PLC, APHogeschool

Voorbeelden

- * I0.0:adres ingangsbits
- * Q4.8:adres uitgangsbits

Zo heeft men ook

- * Byteadressering:IB0(ingangsbite) of QB6(uitgangsbite)
- * Woordadressering:IW0(ingangswoord) of QW5(uitgangswoord)
- * Dubbelwoord:IDW0 of QDW6

In een PLC kunnen niet enkel uitgangen worden aangestuurd maar ook interne hulpbits de zogenaamde *merkers*. Deze worden aangeduid met M. Voorbeeld M0.4

In het engels wordt dit F van flag. Een interne hulpbit is niets anders dan een aantal bits in het RAMgeheugen.

Zo heeft men ook interne bytes en (dubbel)woorden die dan respectievelijk MBx en MWy(MDWy) worden met x en y respectievelijk byte en woordnummer. Voorbeelden MB30,MW32.

102.6 Bussystemen

Het doel van een computerbus is de data-overdracht te bewerkstelligen tussen computer en rand- of meetapparatuur.

De onmiddellijke randapparatuur bestaat uit kaarten die men in de computerrack kan pluggen.

Overzicht van mogelijke kaarten

1. Geheugenuitbreiding
2. Videokaart
3. Disk of tapekaart
4. Netwerkaart
5. Parrallele of seriële kaart
6. Digitale of analoge meetkaart

Hoe kunnen de gegevens behandeld worden. Er bestaan een viertal mogelijkheden

1. Geprogrammeerde input/output: Het programma zal zelf snel genoeg de statusregister van de I/Okaart lezen en indien nodig de I/Obewerkingen uitvoeren.
2. Interrupts: Eén of meerdere interruptlijnen worden voorzien op de bus. Een I/O zal indien nodig deze lijnen activeren; De processor zal het lopende programma onderbreken, het nodige bewaren en na de interrupt het programma voortzetten.
3. DMA: De I/O-kaart bezit een DMA controller die weet waar hij de volgende data moet wegschrijven of weghalen. In dit geval wordt de processor niet belast.
4. I/Oprocessor: Deze processor voert de I/Otaken uit.

Er bestaan twee soorten bussen namelijk de seriële en de parallelle.

- * Parallele bus: Iedere bit heet een fysieke lijn. Er zijn wel veel lijnen nodig om de overdracht te bewerkstelligen, maar men heeft wel een hoge transfertsnelheid.
- * Seriële bus: Nu is er weinig kabel nodig, maar de transfertsnelheid is laag.

102.6.1 Computerbussen

Hier geven we enkel een overzicht van wat er zoal bestaat want dit onderwerp gaat ons veel te ver leiden. Het gaat hier over het koppelen van computers dus ook PLC maar dit hoort al thuis in een (ver gevorderde) cursus electronica.

- * RS-232-C
- * RS485/422
- * Inter C bus
- * IEEE-488bus

102.6.2 Veldbussen

Dit onderdeel hoort al eerder thuis in een cursus regeltechniek, maar we houden we het bij een overzicht met wat verduidelijking. In dit geval gaat het over aansluitingen tussen computers en sensoren.

- * **ASiBus:** Actuator Sensor Interface. Dit is een niet-afgeschermd tweeaderige kabel met specifieke verbindingselementen. De kabel verstuurt zowel bedieningsboodschappen als data. Het is de ASi interface chip die de kracht van dit systeem bewaard. In deze chip wordt de parallele data van sensoren omgezet in seriële gegevens, die dan bij de ASi beheerder belanden wat eigenlijk de PLC is. Er zijn dus twee interfaces; de kant van de randapparatuur en de kant van de besturing.
- * **Bitbus:** De overdracht gebeurt snel via een RS-485 verbinding via SDLC protocol. Is vooral geschikt voor uitwisselen van berichten tussen systemen in industriële toepassingen.
- * **CANbus:** Controller Area Network, wordt vooral gebruikt in de automobiellindustrie. Het gebruikte protocol is de CSMA/CD-BAType.
- * **FIPbus:** Factory Instrumentation Protocol; bedoeld voor industriële automatisering. Het medium is glasvezel of getwiste kabel. Het protocol is gebaseerd op het feit dat toepassingen een cyclisch gedrag vertonen.
- * **Inter C bus:** principe berust op synchrone communicatie en daarom zijn er twee signalen nodig: datalijn (SDA) en klok (CLK). Dit systeem is bijzonder makkelijk toe te passen over lange afstanden.
- * **Interbus-S:** De filosofie gaat ervan uit dat datacommunicatie in een hiërarchie wordt ingedeeld.
- * **Profibus:** Process Field Bus volgt het OSI model.

102.7 Grafcet

102.7.1 Inleiding

Omdat de technologische evolutie het automatiseren van processen steeds ingewikkelder maakte, is men tot een manier moeten komen om de technische en functionele specificaties ondubbelzinnig en duidelijk te omschrijven. Deze manier van werken werd in de jaren 70 en 80 in Duitsland en Frankrijk ontwikkeld en wordt GRAFCET genoemd. Dit staat voor Graphe Fonctionnel de Commande Etape/Transition. In Duitsland wordt deze werkwijze Funktionsplan genoemd en in België en Nederland functiediagram. De norm is de **NEN-EN-IEC 60848**.

102.7.2 Algemene principes

Een automatisch systeem bestaat uit twee delen die van mekaar afhangen

1. Bestuurd systeem: machine of installatie die het fysisch proces uitvoert
2. Besturend systeem: systeem dat informatie ontvangt van operator of bestuurd systeem (meting in terugkoppeling) en als gevolg hiervan commando's genereert voor het bestuurd systeem

Men gaat een proces analyseren en correcte beschrijving geven van de relaties tussen de ingangen (voorwaarden) en de uitgangen (acties). Men gaat dus de procescyclus opsplitsen in een aantal duidelijk gedefinieerde opeenvolgende stappen die gescheiden worden door overgangen. Het einde van een stap wordt gevolgd door een voorwaarde om naar de volgende stap over te gaan zodat stappen mekaar niet kunnen overlappen. Een functiediagram (grafcet) wordt samengesteld uit drie genormaliseerde symbolen.

1. De stap
2. De gerichte verbinding
3. De overgang

Het principe van programmeren is zoals het schrijven van een flowchart van eender welk ander programma. De stappen zijn als volgt

1. Analyse van het probleem en dus opsplitsen van het proces in stappen
2. Aan elke stap koppelen we al de nodige acties
3. Een stap wordt actief als de voorgaande stap actief is en de overgangsvoorwaarden voldaan zijn
4. De eerste stap is de initialisering

102.7.3 Symbolen

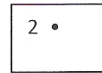
a) De stap



initiele stap: hier start de cyclus



een stap



een actieve stap

b) De gerichte verbinding

De evolutie van het proces wordt aangegeven door vertikaal gerichte verbindingen.

c) De overgang

De overgang is de voorwaarde om van de ene stap naar de andere over te gaan. De overgang kan enkel plaats vinden als de voorgaande stappen allemaal actief zijn. We werken dus met booleaanse structuren.

Dus twee stappen zijn gescheiden door een overgang en twee overgangen zijn gescheiden door een stap.

Hier moeten we opletten. Als we een lamp aansteken door een drukknop te bedienen en het doven gebeurt door het bedienen van dezelfde drukknop is er een probleem. Dit alles gebeurt zo snel dat van toestand 1 naar toestand 2 (AAN) we de drukknop bedienen maar we houden de drukknop nog steeds vast als de overgang gebeurt is zodat het lijkt dat we al terug willen naar toestand 2 (UIT). Dit is onmogelijk. Om dit te verhinderen maakt men gebruik van zogenaamde flanksturing.

102.7.4 Sequenties

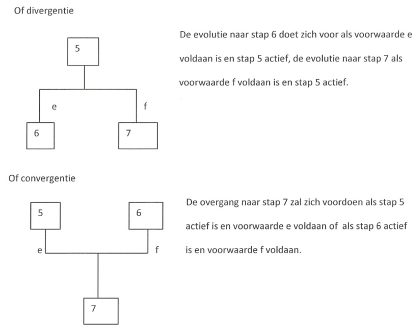
Enkelvoudige sequentie

Ook wel de lineaire sequentie genoemd.

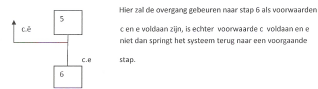


Keuzesequentie

Hier maakt men gebruik van de OF voorwaarde

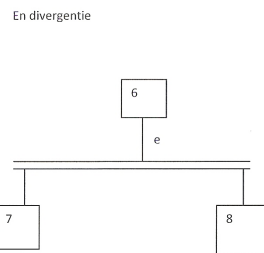


Ook kan men in de OF voorwaarde een voorwaardelijke sprong 'terug' programmeren

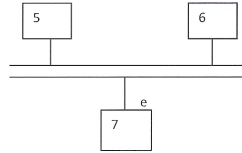


Gelijktijdige sequentie

Hier gebruikt men de EN voorwaarde



EN convergentie



Commando's

Men heeft vier types commando's

1. S:commando wordt gememoriseerd(S:Stored)
2. D:commando wordt vertraagd(D:Delay)
3. L:commando wordt tijdsbegrensd(L:Limited)
4. C:commando wordt voorwaardelijk(C:Conditional)

Als men commando's ingeeft moet men oppassen met de volgorde waarin men deze aangeeft. Stel dat het commando als volgt wordt ingegeven

- * CLS:de stap is gememoriseerd en tijdsbegrensd en wordt pas uitgestuurd als aan een voorwaarde werd voldaan
- * SCL:de stap zal worden gememoriseerd maar de tijdsbegrenzing en het uitsturen hangt af van een voorwaarde

Dit lijken twee dezelfde acties programmatorisch gezien maar zijn in praktijk totaal verschillende acties.

102.8 Voorbeeld

Onderstaand voorbeeld werd ontleend uit de cursus 'Inleiding in de PLC', KHLim dep IWT.

