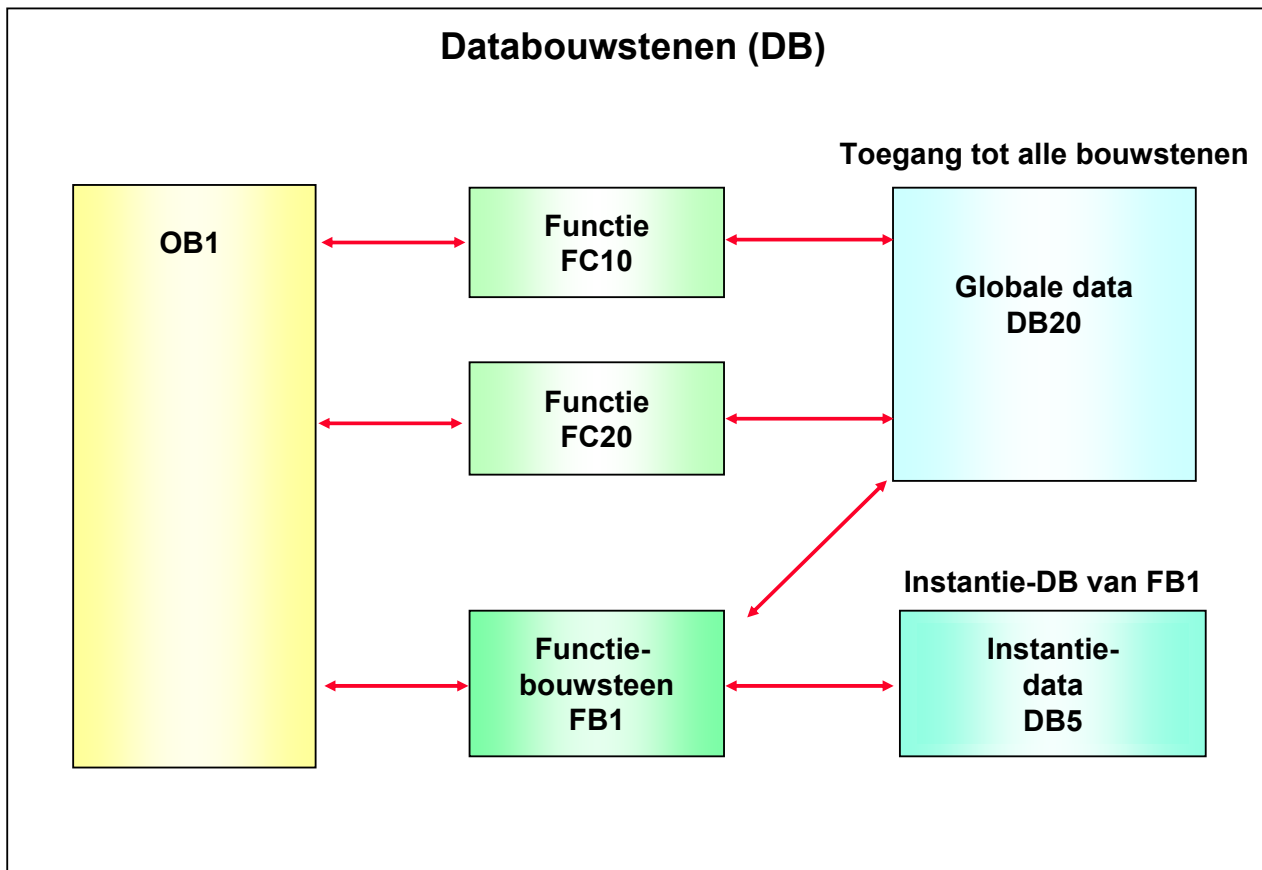


Gegevensopslag in databouwstenen



1

Basis PLC



2

Basis PLC

Algemeen

De databouwstenen dienen om gebruikersgegevens in op te slaan. Net zoals de programmabouwstenen nemen ze plaats in het gebruikersgeheugen in. De databouwstenen bevatten variabelen (bijvoorbeeld numerieke waarden) die nodig zijn voor de uitvoering van het gebruikersprogramma.

Het gebruikersprogramma heeft toegang tot de data van een databouwsteen via instructies op bits, bytes, woorden en dubbelwoorden. De toegang kan via symbolen of via absolute adressen gebeuren.

Toepassingsgebied

De databouwstenen kunnen, naargelang van hun inhoud, op verschillende manieren gebruikt worden door de gebruiker. We onderscheiden:

- de globale-databouwstenen: bevatten informatie waartoe men vanuit alle programmabouwstenen van het gebruikersprogramma toegang heeft.
- de instantie-databouwstenen: zijn altijd verbonden aan een oproep van een FB. De data van deze DB mogen in principe alleen bewerkt worden door de overeenkomstige FB. De instantie-databouwstenen worden in detail besproken in het hoofdstuk "Functies en functiebouwstenen" van de cursus NST-S7SYS2.

DB aanmaken

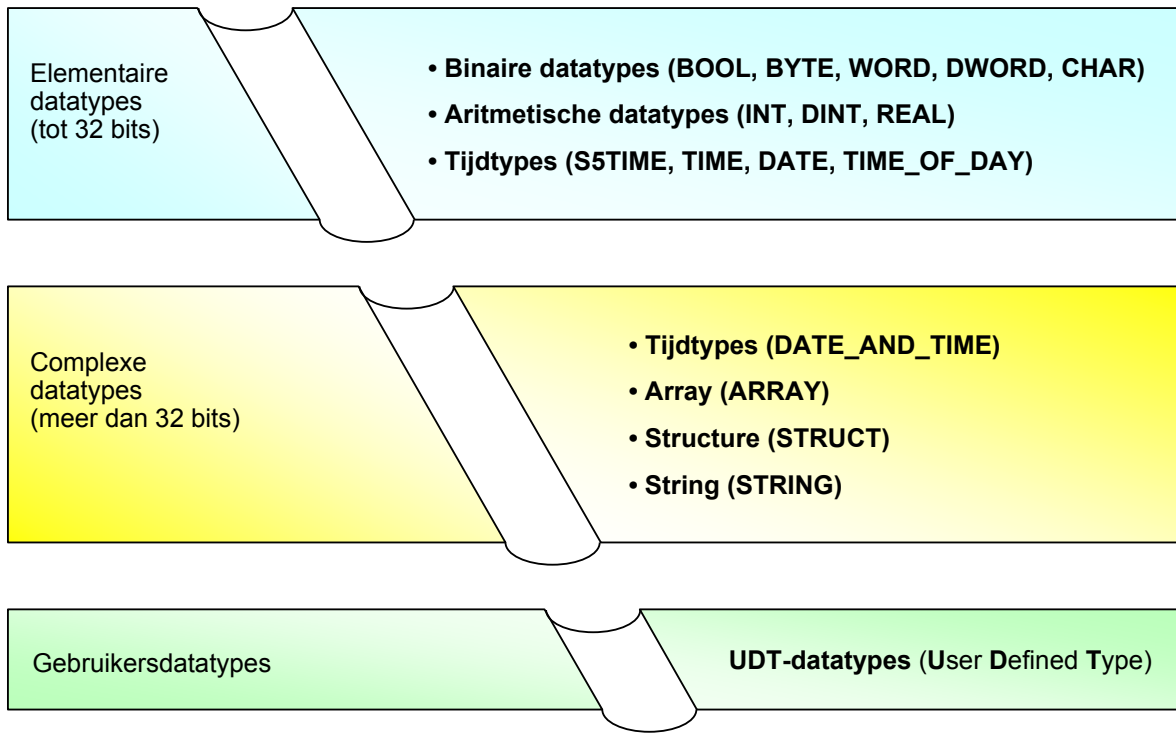
De globale-DB's worden via de programma-editor aangemaakt of volgens een vooraf gedefinieerd "gebruikersdatatype".

De instantie-databouwstenen worden aangemaakt bij de oproep van een FB.

Registers

De CPU bezit twee databouwsteenregisters, het DB-register en het DI-register, waardoor er twee databouwstenen tegelijkertijd kunnen worden geopend.

Overzicht van de datatypes in STEP7



3

Basis PLC

Overzicht

De datatypes definiëren de eigenschappen van de gegevens, meer bepaald de weergave van de inhoud van één of meer onderling afhankelijke operanden en de toegelaten waardenbereiken.

Het datatype bepaalt ook de instructies die mogelijk zijn.

Elementaire datatypes

De elementaire datatypes zijn gedefinieerd volgens IEC 1131-3. Het datatype bepaalt ondubbelzinnig de geheugenruimte die nodig is. Zo zal bijvoorbeeld het datatype "woord" (Word) 16 bits in het gebruikersgeheugen bezetten.

De elementaire datatypes hebben altijd een lengte die kleiner dan of gelijk aan 32 bits is; ze kunnen volledig in de accumulatoren geladen worden en door elementaire STEP7-instructies bewerkt worden.

Complexe datatypes

De complexe datatypes kunnen enkel gebruikt worden in combinatie met variabelen die in globale-databouwstenen gedeclareerd werden. De complexe datatypes kunnen niet volledig in de accumulator geladen worden met de laadinstucties. Om complexe datatypes te bewerken, beschikken we over standaardbouwstenen in de "IEC"-bibliotheek.

Gebruikersdatatypes

Een gebruikersdatatype (User Data Type, UDT) kan gebruikt worden voor databouwstenen of als datatype in een variabelendeclaratietabel.

De UDT's worden met de databouwsteeneditor aangemaakt.

De structuur van een UDT kan groepen van elementaire en/of complexe datatypes bevatten.

Elementaire datatypes in STEP 7

Sleutelwoord	Grootte (in bits)	Voorbeeld van een constante van dit type
BOOL	1	1 of 0
BYTE	8	B#16#A9
WORD	16	W#16#12AF
DWORD	32	DW#16#ADAC1EF5
CHAR	8	' w '
S5TIME	16	S5T#5s200ms
INT	16	123
DINT	32	65539
REAL	32	1.2 ou 34.5E-12
TIME	32	T#2D1H3M45S12MS
DATE	16	D#1993-01-20
TIME_OF_DAY	32	TOD#12:23:45.12

BOOL, BYTE, WORD, DWORD, CHAR De variabelen van het datatype BOOL vertegenwoordigen een bit, de variabelen met de datatypes BYTE, WORD, DWORD zijn bitsequenties van 8, 16 of 32 bits. De afzonderlijke bits worden niet geëvalueerd in deze datatypes.

De BCD-getallen en de telwaarde, evenals het datatype CHAR dat een karakter voorstelt in de ASCII-weergave, zijn speciale vormen van deze datatypes.

S5TIME De variabelen van het datatype S5TIME worden gebruikt om de tijdwaarden in de timers te definiëren. De tijdsduur wordt aangegeven in uren, minuten, seconden of milliseconden. De tijdwaarden kunnen ingegeven worden met een underscore (1h_4m) of zonder (1h4m).
De functies FC33 en FC40 van de bibliotheek "IEC Function Blocks" converteren het formaat S5TIME in TIME of TIME in S5TIME.

INT, DINT, REAL De variabelen met deze datatypes stellen getallen voor, waarmee er aritmetische bewerkingen kunnen worden uitgevoerd.

TIME Een variabele van het datatype TIME (tijdsduur) bezet een dubbelwoord. Deze variabele wordt bijvoorbeeld gebruikt om tijdwaarden te definiëren in de IEC-timers. De inhoud van een variabele wordt geïnterpreteerd als een DINT-getal in milliseconden en kan dus positief of negatief zijn (bijvoorbeeld: T#1s = L#1000, T#24d20h31m23s64msw = L#214748647).

DATE Een variabele van het datatype DATE (datum) wordt opgeslagen in een woord als geheel getal zonder voortekenbit. De inhoud van de variabele komt overeen met het aantal dagen dat verlopen is sinds de 01.01.1990 (bijvoorbeeld: D#1999-06-24 = W#16#0D85).

TIME_OF_DAY Een variabele van het datatype TIME_OF_DAY (uur van de dag) bezet een dubbelwoord dat het aantal milliseconden bevat dat verlopen is sinds het begin van de dag (0:00 h) onder de vorm van een geheel getal zonder voortekenbit (bijvoorbeeld: TOD#23:59:59.999 = DW#16#0526_5B77).

Complexe datatypes

Sleutelwoord	Grootte (in bits)	Voorbeeld				
DATE_AND_TIME (datum en uur)	64	DT#97-09-24-12:14:55.0				
STRING (String met maximum 254 karakters)	8 * (aantal karakters +2)	'Dit is een string' 'SIEMENS'				
ARRAY (array, groepering van componenten van hetzelfde datatype)	door de gebruiker gedefinieerd	Meetwaarden: ARRAY[1..20] INT				
STRUCT (structure, groepering van componenten van verschillende datatypes)	door de gebruiker gedefinieerd	Motor : STRUCT Snelheid : INT Stroom : REAL END_STRUCT				
UDT (User Defined Data Type = gebruikersdatatype, "profiel" op basis van elementaire of complexe datatypes)	door de gebruiker gedefinieerd	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">UDT „bouwsteen“</td> <td style="width: 50%;">UDT „array el.“</td> </tr> <tr> <td>STRUCT Snelheid : INT Stroom : REAL END_STRUCT</td> <td>Aandr.:ARRAY[1..4] UDT1</td> </tr> </table>	UDT „bouwsteen“	UDT „array el.“	STRUCT Snelheid : INT Stroom : REAL END_STRUCT	Aandr.:ARRAY[1..4] UDT1
UDT „bouwsteen“	UDT „array el.“					
STRUCT Snelheid : INT Stroom : REAL END_STRUCT	Aandr.:ARRAY[1..4] UDT1					

Complexe datatypes De complexe datatypes (arrays en structures) worden aangemaakt door verschillende elementaire of complexe datatypes te groeperen.

De gebruiker kan zo volgens zijn behoefte datatypes creëren waarmee hij grote gegevenshoeveelheden kan structureren en via symbolen bewerken.

De complexe datatypes kunnen niet onmiddellijk in hun geheel (meer dan 32 bits) bewerkt worden met de STEP7-instructies; de bewerking kan alleen component per component gebeuren.

De complexe datatypes zijn op voorhand gedefinieerd. De lengte van het datatype DATE_AND_TIME bedraagt 64 bits. De lengte van de datatypes ARRAY, STRUCT (structure) en STRING (karakterstring) wordt door de gebruiker bepaald.

De variabelen die complexe datatypes bevatten, kunnen enkel gedeclareerd worden in globale-databouwstenen en onder de vorm van parameters of als lokale variabelen van programmabouwstenen.

Gebruikersdatatypes De gebruikersdatatypes vertegenwoordigen een zelfbepaalde structuur, die opgeslagen is in de UDT-bouwstenen (UDT1 tot UDT65535) en die gebruikt kan worden als "profiel" voor het datatype van een andere variabele.

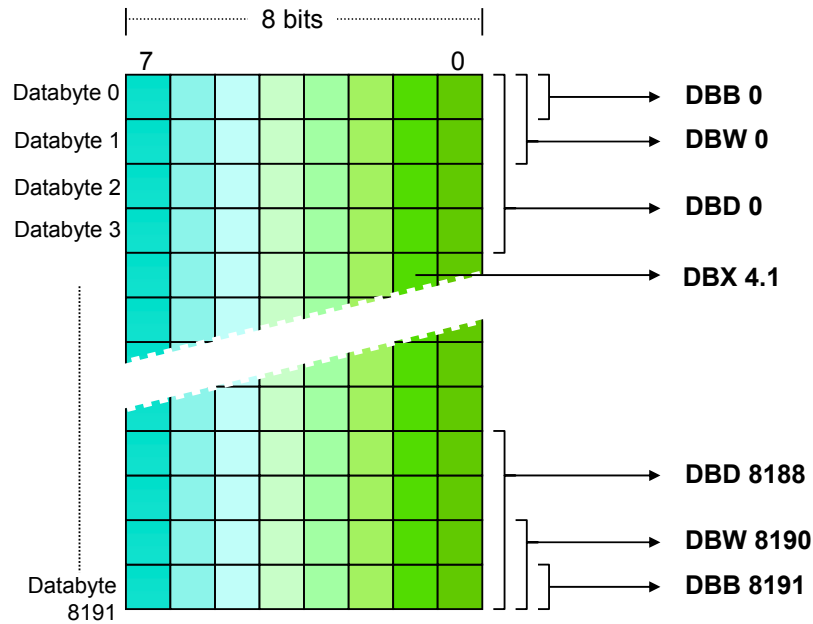
Deze oplossing vermijdt overbodig tikwerk tijdens het ingeven van een databouwsteen, omdat eenzelfde structuur verschillende keren gebruikt wordt.

Voorbeeld: we hebben dezelfde structuur 10 keer nodig in een databouwsteen. We definiëren dan eerst de structuur en bewaren deze, bijvoorbeeld als UDT1. In de DB definiëren we een variabele "Adressen" onder de vorm van een array met 10 elementen van het type UDT1:

```
Adressen : array[1..10]
           UDT1
```

Op die manier creëren we 10 gegevensbereiken met de in UDT1 gedefinieerde structuur zonder ze opnieuw te moeten ingeven.

Adressering van de data-elementen



6

Basis PLC

Algemeen

De adressering van de data-elementen van een databouwsteen gebeurt byte per byte, zoals bij de merkers.

We kunnen databytes, datawoorden of datadubbelwoorden laden en transfereren. In het geval van de datawoorden geven we het eerste byte-adres voor de instructie in (bijvoorbeeld L DBW 2), er worden dan vanaf dit byte-adres 2 bytes geladen. In het geval van de dubbelwoorden worden er 4 bytes geladen vanaf het aangegeven byte-adres.

Aantal, lengte

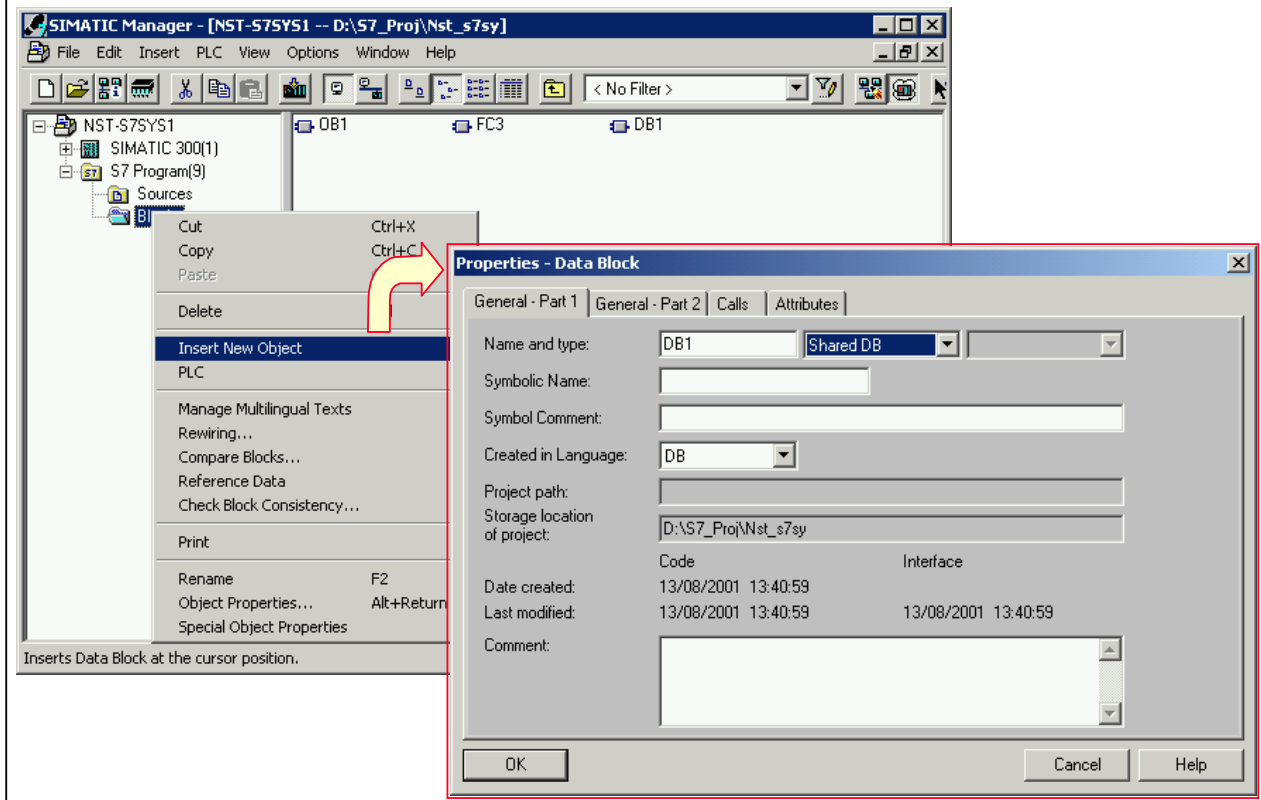
Het aantal databouwstenen is afhankelijk van de gebruikte CPU.

De maximale lengte van de bouwsteen bedraagt 8 Kb op de S7-300 en 64 Kb op de S7-400.

Nota

In geval van poging tot toegang tot onbestaande data-elementen of databouwstenen gaat de CPU in stop als er geen fout-OB geprogrammeerd werd.

Een nieuwe databouwsteen aanmaken



Programma-editor

Met de LAD/STL/FBD-editor kan een bestaande databouwsteen geopend worden of een nieuwe databouwsteen aangemaakt worden.

Dialogbox "New"

Wanneer u op het knopje "New" klikt, verschijnt de dialogbox "New" op het scherm. Selecteer het project en het gebruikersprogramma en tik dan de "Object name" in, bijvoorbeeld DB2 (het veld "Object Type" zou moeten aangeven: "Data Block" of "All that can be processed"). Na bevestiging met het "OK"-knopje verschijnt de dialogbox "New Data Block" op het scherm.

Dialogbox "New Data Block"

In deze dialogbox definieert u, welk type databouwsteen u wil aanmaken:

- databouwsteen (globale-databouwsteen).
- databouwsteen verbonden met een gebruikersdatatype (maakt een DB met de structuur van een UDT-bouwsteen).
- databouwsteen verbonden met een functiebouwsteen (maakt een instantie-DB voor een FB). Dit punt wordt meer in detail uitgelegd in het hoofdstuk "Functies en functiebouwstenen".

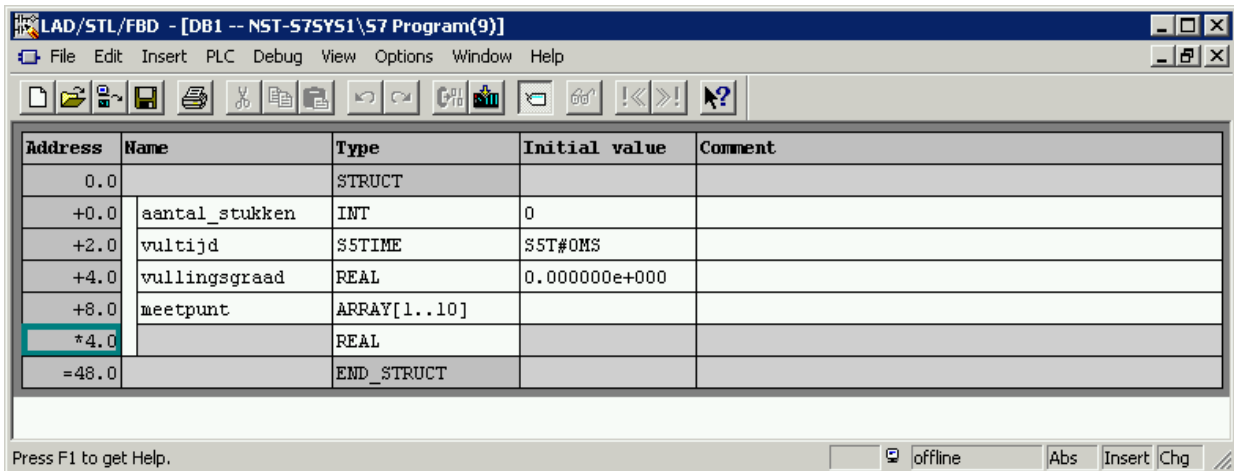
In de SIMATIC Manager

Wanneer u op *Insert -> S7 Block* klikt, verschijnt er een lijst met de verschillende types van bouwstenen. Wanneer u het bouwsteentype (databouwsteen) hebt geselecteerd, krijgt u toegang tot de dialogbox "Properties" waarin u het bouwsteenummer definieert.

Enkel voor de databouwstenen moet u nog het type databouwsteen kiezen ("Shared DB" (globale DB), "Instance DB" (verbonden met FB) of "DB of type" (verbonden met een UDT)) en, indien van toepassing, de bijhorende bouwsteen.

Wanneer de parametrisering klaar is, bevestigt u met OK: de nieuwe bouwsteen wordt dan in het actieve programma toegevoegd.

Een databouwsteen ingeven, opslaan, laden en weergeven



The screenshot shows the SIMATIC Manager software interface. The title bar reads "LAD/STL/FBD - [DB1 -- NST-57SYS1\S7 Program(9)]". The menu bar includes "File", "Edit", "Insert", "PLC", "Debug", "View", "Options", "Window", and "Help". The toolbar contains various icons for file operations and navigation. The main window displays a table with the following data:

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	aantal_stukken	INT	0	
+2.0	wultijd	SSTIME	\$ST#0MS	
+4.0	vullingsgraad	REAL	0.000000e+000	
+8.0	meetpunt	ARRAY[1..10]		
*4.0		REAL		
=48.0		END_STRUCT		

At the bottom of the window, there is a status bar with the text "Press F1 to get Help." and a button labeled "offline".

Data ingeven

De verschillende data-elementen worden in de tabel ingeschreven. Selecteer het eerste vrije veld in de kolom "Name" en tik de benaming van het element in. Met de tab-toets kan u naar de andere kolommen gaan: Type, Initial Value en Comment.

Kolommen

De kolommen hebben de volgende betekenis:

- Address - Wordt bij het aanmaken van variabelen ingevuld. Vertegenwoordigt het begin-bitadres dat deze variabelen in de databouwsteen innemen.
- Name - Symboolnaam van het element.
- Type - Datatype (selectie met de rechtermuisknop of rechtstreeks invoeren met het toetsenbord).
- Initial Value - Dient om een initiële waarde te definiëren. Afwezigheid van een vermelding hier betekent een default-parametrisering met de waarde nul.
- Comment - Dient om het data-element te beschrijven (facultatief).

Save

Het icoontje met het disketje kan gebruikt worden om de databouwsteen te bewaren op de harddisk van het programmeertoestel.



Download

Net als de programmabouwstenen moeten ook de databouwstenen in de CPU geladen worden.

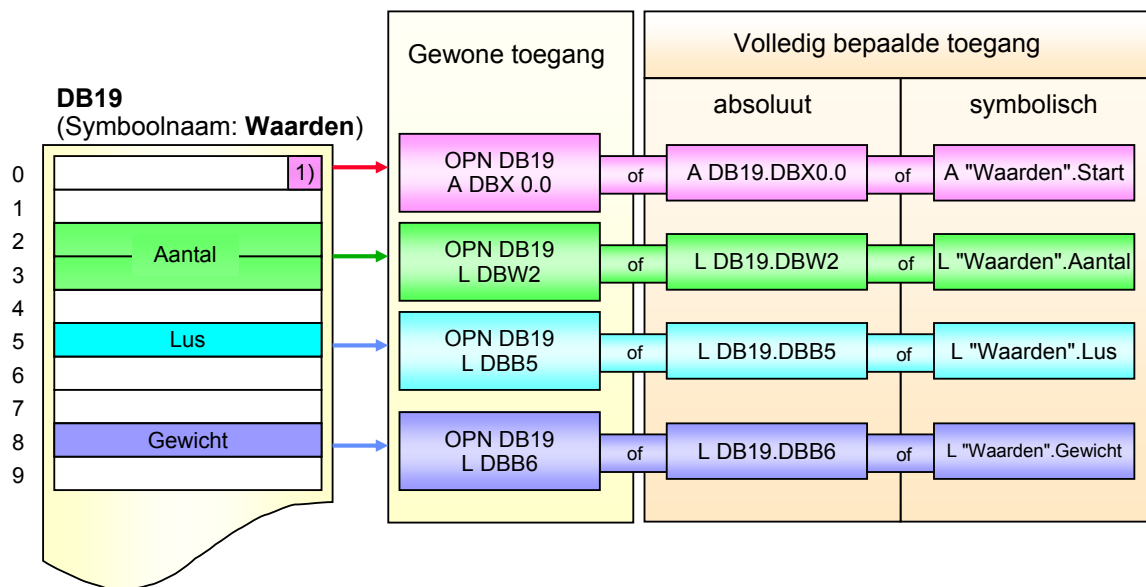


Monitor

Om de actuele waarden in de databouwsteen te bekijken, gaat u naar het scherm "Data View". U kan de databouwsteen visualiseren met het icoontje met het "brillettje" in de knoppenbalk (cyclische weergave van de actuele waarden van de DB in de CPU). Op deze manier kunnen enkel elementaire datatypes dynamisch worden weergegeven.



Toegang tot de data-elementen



1) Databit 0.0 met de elementnaam "Start"

Een DB openen

Een globale-databouwsteen wordt geopend met de instructie **"OPN DB..."**. Als er al een globale-DB geopend was, wordt deze gesloten. Als er voor de DB een symboolnaam gedefinieerd werd (bijvoorbeeld "Waarden"), kan de DB even goed geopend worden met de instructie **OPN "Waarden"**.

Toegang tot de DB

De figuur toont de instructies die toegang verlenen tot de DB: lezend (laden) of schrijvend (transfereren). Als de DB al open is, volstaat een eenvoudige laad- of transferinstructie. In geval van een gecombineerde instructie, bijvoorbeeld: L DB19.DBW2, wordt de gewenste DB ook aangegeven. De instructie omvat ook het openen van de databouwsteen.

Toegang met symbolen

Toegang met symbolen is enkel mogelijk onder de volgende voorwaarden:

1. Er is in de symbolenlijst aan de DB een symboolnaam toegekend.
2. De LAD/STL/FBD-editor heeft aan de verschillende data-elementen van de databouwsteen een symboolnaam toegewezen.

Voorbeeld: de instructie **L "Waarden".Aantal** opent de DB met de naam "Waarden" en laadt het data-element met de naam "Aantal".

Nota

U zou over het algemeen de toegang met symbolen tot de DB moeten gebruiken. Dit biedt de volgende voordelen:

- het programma is gemakkelijker te lezen,
- we zijn zeker van toegang tot de juiste DB,
- de correcties achteraf in de datastructuur van de DB zijn gemakkelijker te realiseren.

In geval van absolute toegang tot de DB moeten we met de hand al die plaatsen in het programma met toegang tot de DB corrigeren. Bij toegang met symbolen is de aanpassing gemakkelijk via een bronprogramma. De bewerking van de bronprogramma's wordt bestudeerd in een programmeercursus voor gevorderden.

Voorbeeld van een array

Meetpunt

1e meetpunt, datatype Real
2e meetpunt, datatype Real
3e meetpunt, datatype Real
⋮
10e meetpunt, datatype Real

Array met de naam "meetpunt"
(verschillende elementen
van hetzelfde datatype)

Voorstelling in de programma-editor (databouwsteen DB2):

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	Meetpunt	ARRAY[1..10]		
*4.0		REAL		
=40.0		END_STRUCT		

10

Basis PLC

Array

Een array bestaat uit verschillende elementen van hetzelfde datatype. Bovenstaande figuur toont een array "meetpunt" met 10 elementen van het datatype REAL. Verschillende meetwaarden zullen vervolgens in deze array bewaard worden.

Array in DB definiëren

Het sleutelwoord voor een array is "ARRAY[n..m]". Het eerste element (n) en het laatste element (m) zijn aangegeven tussen haakjes. In ons voorbeeld betekent [1..10] 10 elementen, waarbij het eerste element geadresseerd wordt met index [1] en het laatste met index [10]. In plaats van [1..10] kunnen we bijvoorbeeld [1..9] definiëren. Dit heeft enkel een invloed op de toegang tot de elementen.

Nota

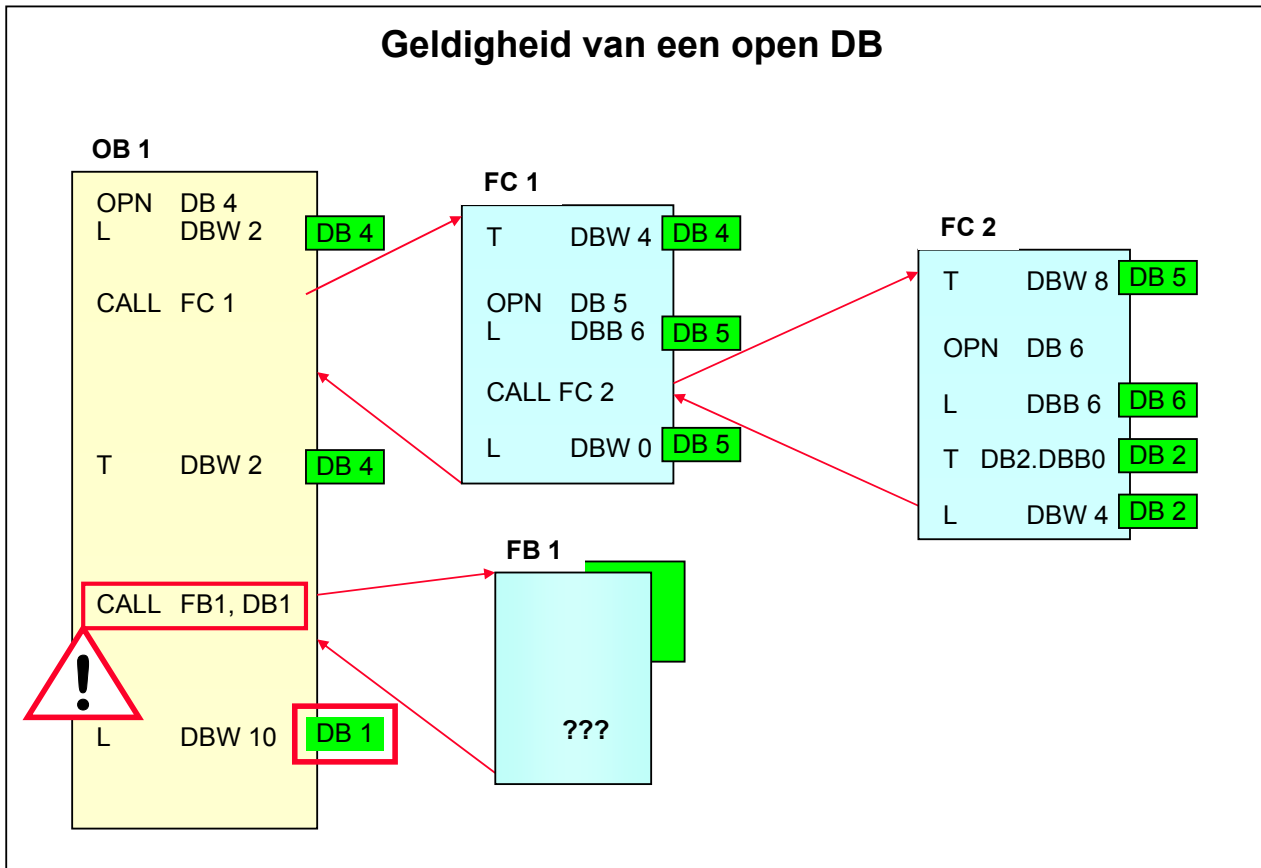
Om een lege databouwsteen te creëren, definieert u een array met het gewenste datatype.

Data View

Om te zien welke waarden er in de verschillende elementen opgeslagen zijn, gaan we naar een andere weergave met het menu *View -> Data View*. Het scherm *Data View* toont de op dit moment opgeslagen waarden in de kolom "Actual Value".

Address	Name	Type	Initial value	Actual value	Comment
0.0	Meetpunt[1]	REAL	0.000000e+000	0.000000e+000	
4.0	Meetpunt[2]	REAL	0.000000e+000	0.000000e+000	
8.0	Meetpunt[3]	REAL	0.000000e+000	0.000000e+000	
12.0	Meetpunt[4]	REAL	0.000000e+000	0.000000e+000	
16.0	Meetpunt[5]	REAL	0.000000e+000	0.000000e+000	
20.0	Meetpunt[6]	REAL	0.000000e+000	0.000000e+000	
24.0	Meetpunt[7]	REAL	0.000000e+000	0.000000e+000	
28.0	Meetpunt[8]	REAL	0.000000e+000	0.000000e+000	
32.0	Meetpunt[9]	REAL	0.000000e+000	0.000000e+000	
36.0	Meetpunt[10]	REAL	0.000000e+000	0.000000e+000	

Geldigheid van een open DB



11

Basis PLC

Inleiding

Een geopende databouwsteen blijft open totdat er een andere databouwsteen geopend wordt of totdat er toegang verkregen is tot een andere DB met een gecombineerde instructie (bijvoorbeeld L DB4.DBW6).

FC oproepen

Wanneer we een OB of een FC verlaten om een andere functie op te roepen, blijft de actieve databouwsteen geldig. Terugkeren naar de oproepende FC brengt met zich mee dat de databouwsteen die daarvoor geldig was, opnieuw wordt geopend.

FB oproepen

De oproep van een functiebouwsteen is een bijzonder geval. Er is altijd een instantie-DB verbonden aan een FB (voor verdere informatie, zie hoofdstuk "Functies en functiebouwstenen"). Deze instantie-DB wordt automatisch geopend bij de oproep van de FB.

De globale-DB die daarvoor geopend was, verliest zijn geldigheid bij het terugkeren naar de oproepende bouwsteen.

Dit wil zeggen dat we de nodige globale-DB's opnieuw moeten openen na het oproepen van een FB.

Nota

We kunnen een DB openen met de daartoe strekkende instructies, bijvoorbeeld OPN DB, maar ook door parameters over te brengen naar de parametreerbare bouwstenen.

Zo kunnen we bijvoorbeeld "DB4.DBW6" gebruiken als actuele parameter. In dit geval wordt de DB4 geopend.

Tip: Gebruik zoveel mogelijk een volledige toegangs instructie voor toegang tot de databouwstenen.