

DFCP 4

Programmeerbare Controller

Gebruikershandleiding

Release 1.2 – Oktober 2018



Inhoud

A1- VERNIEUWINGEN VAN DFCP 4 VS DFCP EN VERKRIJGBARE VERSIES	5
1- HOOFDKENMERKEN	6
1.1- Introductie.....	6
1.2- Verplichte hardware en softwaregereedschap.....	7
1.3- Hoofdkenmerken van DFCP 4	7
1.4- Terminologie en syntax	8
2- VERGELIJKINGEN: TYPES EN SYNTAX	9
2.1- Vergelijkingen voor systeemconfiguratie.....	9
2.1.1- Configuratie van modules	9
2.1.2- Power ON status.....	9
2.1.3- Status van verkeerde inputmodules	9
2.1.4- Communicatieprotocol	10
2.1.5- Adres van DFCP 4.....	12
2.1.6- ID van DFCP 4.....	12
2.1.7- Richtlijn voor de berekening van zonsopkomst, zonsondergang en positie van de zon	12
2.1.8- Beheer van verkeerde modules.....	13
2.1.9- Opstelling van de outputs	13
2.1.10- Publiceren van interne virtuele punten op de bus	13
2.1.11- Het uitwisselen van data tussen meerdere DFCP 4 controllers	15
2.1.12- Aantal knopen in een DFCP 4 netwerk	17
2.1.13- Scheduler.....	18
2.2- Event triggered vergelijkingen	19
2.2.1- Logische vergelijkingen	19
2.2.2- SET – RESET vergelijkingen.....	19
2.2.3- TOGGLE vergelijkingen.....	20
2.2.4- COUNTER vergelijkingen	20
2.2.5- THRESHOLD vergelijkingen	22
2.2.6- TIMER vergelijkingen.....	23
2.2.7- Vergelijkingen voor wiskundige en logische berekeningen	24
2.2.8- Vergelijkingen voor aanmaak binaire code	25
2.2.9- Vergelijkingen voor het opnemen van statusveranderingen (EVENT).....	25
2.2.10- Vergelijkingen voor het opnemen van waardeveranderingen (LOG)	26
2.2.11- Beheer van de DALI module (DFDALI)	27
2.3- Tijd triggered vergelijkingen	28
2.3.1- Scheduler vergelijkingen	28
2.4- Macro.....	30
3- SCRIPT	32
3.1- Overzicht	32

3.2- Sleutelwoorden en syntax	33
3.2.1- Het gebruiken van de TRIGGER	33
3.2.2- VAR, GLOBAL VAR en EXTERN VAR	34
3.2.3- Logische en Mathematische operaties	34
3.2.4- IF...THEN...ELSE...ENDIF	35
3.2.5- CARRY en ZERO	36
3.2.6- DEFINE	36
3.2.7- GOTO	38
3.2.8- SUBROUTINES en FUNCTIES	38
3.2.9- BIT(x)	41
3.2.10- WORD(x) en pointers	43
3.2.11- @RAM k en @WORD k	44
3.2.12- SWAP(x)	44
3.2.13- RANDOM(0)	44
3.2.14- BMASK(x)	45
4- PROGRAMMA'S SCHRIJVEN	46
4.1- Regels voor programmeren	46
4.2- Het compileren van het programma	48
4.3- Programma uploaden naar DFCP 4 geheugen	48
5- HET INSTELLEN	49
5.1- Verbindingen	49
5.2- Baudrate selecteren	50
5.3- TCP/IP parameterinstellingen (Ethernet)	51
5.4- RS232 en RS485, seriële poorten van DFCP 4	52
5.5- WEBS communicatiepoort	52
6- DIAGNOSTIEK	53
6.1- Diagnostiek van Domino systeem via DFCP 4	53
7- Technische kenmerken	54
8- Afmetingen	54
9- FXP-XT COMMUNICATIEPROTOCOL	55
9.1- Berichtenformaat naar betekenis	55
9.2- Indeling RAM geheugen	57
9.2.1- Algemene indeling van het RAM geheugen	57
9.2.2- RAM indeling van de Scheduler	60
10- DCP IDE: GEÏNTEGREERD ONTWIKKELINGSPLATFORM VOOR APPLICATIES DIE DFCP 4 GEBRUIKEN ..	61
10.1- Beschrijving van het softwarepakket DCP IDE	61
10.2- DCP IDE	61
10.2.1- Seriële communicatie en firmware versies van DFCP 4	66
10.2.2- Compileren	66
10.2.3- Transfereren	67
10.3- DCP VISIO	68

10.3.1- Groepen.....	69
10.3.2- Projecten.....	72
10.3.3- Simulator.....	72
11- MODBUS COMMUNICATIEPROTOCOL.....	73
11.1- Overzicht	73
11.2- Ondersteunende MODBUS functies	73
11.3- Tabellen over relatieve Words – Parameters.....	73
11.3.1- Fysieke inputs.....	74
11.3.2- Fysieke outputs.....	74
11.3.3- Virtuele punten.....	75
11.3.4- Registers.....	78
11.3.5- Counters	80

A1- VERNIEUWINGEN VAN DFCP 4 VS DFCP EN VERKRIJGBARE VERSIES

Hardware:

- Snelste processor en geheugen
- RS485 communicatiepoort (COM4) toegevoegd
- Ethernetverbinding toegevoegd
- Alarmrelais verwijderd

Functies:

- Geïntegreerde wekelijkse Scheduler voor het beheer van 16 punten ("zones") met elk 8 tijdsvakken. Ieder tijdvak kan individueel aan- of uitgezet worden
- Een nieuw programma die naar de DFCP 4 wordt getransfereerd, wordt nu opgeslagen op een andere locatie in het geheugen. Dat zorgt ervoor dat wanneer een nieuw programma wordt ingeladen, het oude blijft werken totdat het inladen voltooid is, wat automatisch gebeurt
- MODBUS TCP/IP Slave op ETH-poort
- Geïntegreerde Ethernet Bridge, multi-user tot 8 gelijktijdige verbindingen
- Geïntegreerde webserver, multi-user tot 8 gelijktijdige verbindingen, inclusief beheer van scènes
- Simpele diagnostiek via een webbrowser waardoor het installeren van het DCP IDE programma overbodig is

De verkrijgbare versies zijn:

- **DFCP 4/STD:** 1 RS232 + 2 RS485
- **DFCP 4/ETH:** 1 RS232 + 2 RS485 + ETH

Correct disposal of this product

(Waste Electrical & Electronic Equipment)



(Applicable in the European Union and other European countries with separate collection systems). This marking on the product, accessories or literature indicates that the product should not be disposed of with other household waste at the end of their working life. To prevent possible harm to the environment or human health from uncontrolled waste disposal, please separate these items from other types of waste and recycle them responsibly to promote the sustainable reuse of material resources. Household users should contact either the retailer where they purchased this product, or their local government office, for details of where and how they can take these items for environmentally safe recycling. This product and its electronic accessories should not be mixed with other commercial wastes for disposal. Specifically about the battery, check local regulations for correct disposal. Never use municipal waste.

Installation and use restrictions

Standards and regulations

The design and the setting up of electrical systems must be performed according to the relevant standards, guidelines, specifications and regulations of the relevant country. The installation, configuration and programming of the devices must be carried out by trained personnel. The installation and the wiring of the bus line and the related devices must be performed according to the recommendations of the manufacturers (reported on the specific data sheet of the product) and according to the applicable standards. All the relevant safety regulations, e.g. accident prevention regulations, law on technical work equipment, must also be observed.

Safety instructions

Protect the unit against moisture, dirt and any kind of damage during transport, storage and operation. Do not operate the unit outside the specified technical data. Never open the housing. If not otherwise specified, install in closed housing (e.g. distribution cabinet). Earth the unit at the terminals provided, if existing, for this purpose. Do not obstruct cooling of the units. Keep out of the reach of children.

Setting up

The physical address assignment and the setting of parameters (if any) must be performed by the specific softwares provided together the device or by the specific programmer. For the first installation of the device proceed according to the following guidelines:

- Check that any voltage supplying the plant has been removed
- Assign the address to module (if any)
- Install and wire the device according to the schematic diagrams on the specific data sheet of the product
- Only then switch on the 230Vac supplying the bus power supply and the other related circuits

Applied standards

This device complies with the essential requirements of the following directives:

2004/108/CE (EMC)
2006/95/CE (Low Voltage)
2002/95/CE (RoHS)
EN 55022 Class B

Note

Technical characteristics and this data sheet are subject to change without notice.

1- HOOFDKENMERKEN

1.1- Introductie

De Domino bus bestaat eigenlijk uit input- en outputmodules, gedistribueerd in de installatie. De functies die uitgevoerd moeten worden door het systeem (het oproepen van vergelijkingen), zitten in het interne geheugen van de outputmodules en worden geprogrammeerd via het softwarepakket BDTools.

De DFCP 4 programmeerbare controller is gemaakt om complexe, technische operaties in een Domino bus te voltooien. De gevraagde functies zouden onmogelijk of gewoonweg te moeilijk zijn via het standaard programmeren om dan geïmplementeerd te worden. DFCP 4 is geboren uit 10 jaar ervaring in de wereld van bussystemen, met een gecertificeerde en welgekende betrouwbaarheid.

DCP IDE laat zowel het programmeren van DFCP 4 controllers als het programmeren van outputmodules toe, wat BDTools doet vergeten.

Deze handleiding is bedoeld voor gebruikers die het Domino systeem al kennen met het standaardprogrammeren. Daardoor zal deze handleiding enkel de kenmerken van de DFCP 4 beschrijven en de syntax voor de geschikte implementatie van de gevraagde functies. Refereer naar de gerelateerde technische documentatie voor onderwerpen zoals functies, het programmeren en karakteristieken van Domino modules. Deze handleiding zal ook de werking van het softwarepakket DCP IDE beschrijven.

Wanneer een programma wordt ontwikkeld voor het Domino systeem, waar een DFCP 4 controller is geïnstalleerd, hou dan altijd de volgende punten in gedachten:

1. De functies die ook moeten werken bij een disconnectie van de DFCP 4, moeten in het geheugen van de gerelateerde outputmodules zitten.
2. Een intern virtueel punt van DFCP 4 kan niet direct gebruikt worden als een input in een vergelijking binnen een outputmodule. Het is zowaar mogelijk om aan de DFCP 4 te vragen om de status van zijn interne virtuele punten te 'publiceren' naar de bus (slaat op die manier de installatie van DF4I/V modules in de installatie op). Die 'openbare' virtuele punten kan ook afgedwongen worden door de bus, zoals de virtuele punten van een DF4I/V module.
3. De status van een intern virtueel punt van de DFCP 4 kan getransfereerd worden naar een outputpunt door een simpele vergelijking binnen de DFCP 4, gelijkaardig met $O_{i,j} = V_x$, waar $O_{i,j}$ de output is van een module binnen de installatie en V_x een intern virtueel punt van DFCP 4.
4. De commando's van dimmermodules (Up, Down, Single Command en Preset from digital point) moeten geïmplementeerd worden enkel door het standaardprogrammatie die in het geheugen van de dimmermodule zit. Het is ook mogelijk om complexe functies te realiseren door het eerst in de DFCP 4 te implementeren en dan het commando te transfereren naar de dimmermodule zoals beschreven in puntje 2.
5. De commando's van rolluikmodules (Open, Close, beide lokaal en gecentraliseerde) moeten geïmplementeerd worden enkel door het standaardprogrammatie die in het geheugen van de rolluikmodule zit. Het is ook mogelijk om complexe functies te realiseren door het eerst in de DFCP 4 te implementeren en dan het commando te transfereren naar de rolluikmodule zoals beschreven in puntje 2.
6. De algemene outputmodules (DF4R, DF4RP) kunnen beide bestuurd worden door een programma in de module en de DFCP 4. Hou er rekening met dat hetzelfde commando naar meerdere outputmodules, sneller uitgevoerd zullen worden als het verwante programma in de outputmodules zit. Dit doordat de modules op hetzelfde moment de gevraagde functie uitvoeren wanneer ze een verandering van het inputpunt in hun programma detecteren, terwijl een commando van de DFCP 4 naar meerdere modules moeten verzonden worden, en dat natuurlijk op een sequentiële manier doet (één commando per outputmodule tegelijkertijd).

1.2- Verplichte hardware en softwaregereedschap

Om DFCP 4 te gebruiken is de software van DCP IDE verplicht. Minimum hardware nodig voor uw pc: 1000MHz processor en 512 Mbytes minimum RAM.

DCP IDE voorziet ook het programma DCPVISIO, wat de status van alle parameters van de DFCP 4 op een grafische manier toont en andere programma's, specifieke functies toelaat.

1.3- Hoofdkenmerken van DFCP 4

- 2032 virtuele digitale punten
- 1024 16 bit registers
- 1024 16 bit tellers
- 512 16 bit timers
- 255 real inputadressen 16 bit
- 255 real outputadressen 16 bit

Speciale virtuele punten:

- **V2032:** wanneer ingesteld op 1 dan schort DFCP 4 de berekening van de code en de uitvoering van het script op. Wanneer het reset wordt naar 0 dan werkt de DFCP 4 terug zoals normaal. Het punt V2032 zal echter automatisch na een time-out resetten naar 0.
- **V2020 – V2027:** wordt actief bij een communicatiestoring voor meer dan 10 seconden op de 'relative socket' 1 – 8 van de ETH-poort.
- **V2019:** de activatie van dit punt doet de herinitialisering van het Ethernet interface resetten.
- **V2018:** wordt geactiveerd wanneer een communicatieverlies op COM4 (RS485) van meer dan 5 seconden plaatsvindt.
- **V2017:** wordt geactiveerd wanneer één of meerdere MODBUS-slaven niet antwoorden op de DFCP 4 (ingesteld in MODBUS master mode) na 50 herzieningen. Het zal gedeactiveerd worden na een volledig polling cycle naar slaaf toestellen zonder errors.
- **V2016:** wordt geactiveerd wanneer de communicatie met WEBS faalt.
- **V2015:** wordt voorbehouden, nieuwe data is aanwezig in de NOTIFY buffer.
- **V2014:** is actief tijdens zonsopkomst en zonsondergang.
- **V2013:** is voorbehouden.
- **V2012:** wordt geactiveerd wanneer een communicatieverlies op COM2 (RS485) van meer dan 5 seconden plaatsvindt.
- **V2011:** wordt geactiveerd wanneer een communicatieverlies op COM1 (RS232) van meer dan 5 seconden plaatsvindt.
- **V2010:** wordt geactiveerd, 0.5 seconden van het einde van het initialisatieproces.
- **V2009:** de buffer van analoge voorvallen (LOG of LOGC) is volledig of oude voorvallen werden overschreven.
- **V2008:** de buffer van binaire voorvallen (EVENT of EVENTC) is volledig of oude voorvallen werden overschreven.
- **V2007:** is voorbehouden.
- **V2006:** is voorbehouden.
- **V2005:** error tijdens de uitvoering van een script (bv. geen geldige aanwijzing).
- **V2004:** time-out tijdens de uitvoering van een script (>500 msec).
- **V2003:** 1 sec periodeklok (tokkelt zijn status iedere 0.5 seconden).
- **V2002:** storing op de bus.
- **V2001:** modulestoring.

1.4- Terminologie en syntax

In deze handleiding worden sommige symbolen en notaties gebruikt waar de betekenis hieronder wordt uitgelegd.

Algemeen:

DI	echte of virtuele digitale input
DO	echte of virtuele digitale output
AI	analoge input of algemeen register
AO	analoge output of algemeen register
Ri	algemeen register

Adressen, kanalen en punten:

O3.1	punt 1 van outputmodule 3
R12	register R12
R14.5	bit 5 van register R14 (alleen voor script, zie gerelateerde paragraaf)
V100	virtueel punt 100
V17..V32	alle virtuele punten van V17 tot V32
O3.1..O4.16	alle outputpunten van O3.1 tot O4.16

Nummers:

328	decimaal getal
0b0001010011111011	16 bit binair nummer
0b11110010	8 bit binair nummer
0x14FB	16 bit hexadecimaal getal

2- VERGELIJKINGEN: TYPES EN SYNTAX

2.1- Vergelijkingen voor systeemconfiguratie

2.1.1- Configuratie van modules

Specialiseer de geïnstalleerde module binnen het systeem (zie DCP IDE sleutelwoordenlijst)

```
DF4I  = ( I1 )
DF4I  = ( I2 )
DF8IL = ( I129, I130, V131, V132 )
DF4R  = ( O1 )
DF4RP = ( O2 )
DFDM  = ( O3 )
```

2.1.2- Power ON status

Specialiseer de status of waarde die zijn gegeven aan outputs, registers of counters bij het opstarten of een reset.

```
POWERON = ( O1.1 = 1,      \
             O1.2 = -,      \
             AO2..AO4 = 45,  \
             R12 = -,        \
             C32 = 1245,     \
             C33..C35 = -,   \
             V100 = -,       \
             V1..V16 = 1,    \
             V17..V32 = -    )
```

O1.2 = - betekent dat **O1.2** bij het opstarten de waarde behoud van de vorige werking (als de optie **RAMtoFIELD** ingeschakeld is, lees dan het volgende). **V100 = -** betekent dat **V100** bij het opstarten de waarde behoud van de vorige werking (de RAM heeft een reservebatterij).

AO2..AO4 = 45 betekent dat de outputs van AO2 tot AO4 de waarde 45 zullen opnemen bij het opstarten(bv. dimmer).

R12 = - behoud, bij het opstarten, de waarde van de vorige werking.

De standaard (dus als het niet gespecificeerd is) is dat de status en waarde van alle parameters bij het opstarten of een reset, nul is.

2.1.3- Status van verkeerde inputmodules

De status van een gefaalde input module wordt door de DFCP 4 opgemerkt. Als het niet gespecificeerd is, dan gebruikt de DFCP 4 de laatst mogelijke waarde.

```
FAIL = ( I1.1 = 1,      \
          I3.2 = 0,      \
          I5.1..I5.4 = 1, \
          AI15 = 80      )
```

2.1.4- Communicatieprotocol

Stel de communicatieprotocollen die gebruikt worden voor ieder poort van de DFCP4 in. COM 1 is de communicatiepoort van het voorpaneel (RS232), COM2 en COM4 aan de busklem (RS485), COM 3 is de speciale poort onder het deksel linksonder en LAN1 de Ethernetpoort (enkel DFCP4/ETH). Aan de ETH-poort zullen beide, MODBUS TCP/IP en RTU, geactiveerd worden (via automatische detectie). De richtlijnen zijn de volgende (FXPXT kan weggelaten worden omdat het toch geactiveerd is):

```
COM1 = (FXPXT, MODBUS) // RS232
COM2 = (FXPXT, MODBUS) // RS485
COM3 = (FXPXT, MODBUS) // Speciale poort (WEBS)
COM4 = (FXPXT, MODBUS) // RS485
LAN1 = (FXPXT, MODBUS) // Ethernetpoort
```

De mogelijke opties voor alle COM-poorten (COM3 door de WEBS module) zijn de volgende:

FXPXT	gepatenteerd protocol, <u>altijd actief ook al is het niet gespecificeerd</u> .
MODBUS	MODBUS RTU protocol: volledige wisselwerking tussen het nummer van de Word, gespecificeerd in de Master MODBUS driver en het nummer van de Words gelijst in de RAM map in deze handleiding. Dit is de voorkeursoptie.
MODBUS-	MODBUS RTU protocol: het nummer van de Word, gespecificeerd in de Master MODBUS driver moet verhoogd zijn met 1 met betrekking naar wat er gelijst staat in de RAM map van deze handleiding.

MODBUS MASTER

DFCP 4 gedraagt zich zoals een slaaf-toestel binnen een MODBUS netwerk. Als alternatief kan de MODBUS **MASTER** modus geactiveerd worden voor de DFCP 4, exclusief op COM4, gebruikmakend van de volgende richtlijn:

```
MODBUS a (T, Vm) = ( Wx, n, Ry, \
                    ..... )
```

waar:

a	het adres is (binnen bereik van 1 tot 31) van de MODBUS SLAVE modus om met te communiceren
t	de communicatie time-out is in milliseconden. Indien weggelaten, dan is de standaard 1000 (1 sec).
Vm	het virtueel punt is dat geactiveerd wordt als de slaaf 'a' niet antwoordt op de DFCP4. Die parameter is optioneel en kan weggelaten worden. Het cumulatief punt V2017 is altijd verkrijgbaar, zie paragraaf 1.3.
x	het nummer is van de eerste MODBUS Word om gelezen of beschreven te worden.
n	het nummer is van Words om gelezen of beschreven te worden.
y	het startend nummer is van DFCP4 register waar de Words zijn opgeslagen.

Dezelfde richtlijn laat het schrijven van specifieke Words toe wanneer de waarde van de verwante register is aangepast (door programma, script, enz.) Als meerdere blokken van niet opeenvolgende Words moeten gelezen of beschreven worden aan dezelfde MODBUS, dan zullen er meer lijnen toegevoegd moeten worden.

```
MODBUS a (T, Vm) = ( Wx, n, Ry, \
                    Wj, m, Rk, \
                    ..... )
```

Opmerking: als MODBUS MASTER geactiveerd is, dan zullen alle SLAVE protocollen (inclusief FXPXT) op COM4 gedeactiveerd worden.

Om met meerdere MODBUS randapparatuur te communiceren, voeg dan code toe aan elk SLAVE adres zoals in het volgend voorbeeld. De time-out MOET hetzelfde zijn voor alle knopen.

```
MODBUS 1 (500) = ( W0,      10,   R101, \
                  W100,    10,   R111 )
MODBUS 2 (500) = ( W0,      10,   R121, \
                  W100,    10,   R131 )
MODBUS 3 (500) = ( W0,      10,   R141, \
                  W100,    10,   R151 )
MODBUS 4 (500) = ( W0,      10,   R161, \
                  W100,    10,   R171 )
```

Gezien de functies, gebruikt door DFCP4 in MODBUS MASTER modus, zijn de volgende regels van toepassing:

- Als het specifiek WORD nummer in het bereik van 00000 tot 30000 is, dan leest DFCP4 volgens functie 3 (Read Holding Registers) en schrijft het volgens functie 16 (Write Multiple Registers). De gelezen/geschreven woorden zijn diegene ingedeeld door WORD 00000 tot 30000 in de slaaf.
- Als het specifiek WORD nummer in het bereik van 30001 tot 39999 is, dan leest DFCP4 volgens functie 4 (Read Input Registers) en schrijft het niet (het gaat om input, dus het leest alleen maar). De gelezen woorden zijn diegene ingedeeld door WORD 0000 tot 9998 in de slaaf.
- Als het specifiek WORD nummer in het bereik van 40001 tot 49999 is, dan leest DFCP4 volgens functie 3 (Read Holding Registers) en schrijft het volgens functie 16 (Write Multiple Registers). De gelezen/geschreven woorden zijn diegene ingedeeld door WORD 0000 tot 9998 in de slaaf.
- Als het specifiek WORD nummer in het bereik van 50001 tot 65535 is, dan leest DFCP4 volgens functie 3 (Read Holding Registers) en schrijft het volgens functie 16 (Write Multiple Registers). De gelezen/geschreven woorden zijn diegene ingedeeld door WORD 9999 tot 25534 in de slaaf.

De volgende tabel vat het zojuist gegeven samen.

Word in de DFCP 4	Word in de slaaf	Gebruikte MODBUS functies
00000 - 30000	00000 - 30000	3 [R] en 16 [W]
30001 - 39999	0000 - 9998	4 [R]
40001 - 49999	0000 - 9998	3 [R] en 16 [W]
50000 - 65535	9999 - 25534	3 [R] en 16 [W]

Opmerkingen:

- De getoonde nummers in de tabel staan allemaal in decimaal formaat.
- Het eerste geval (Word 00000 - 30000) kan overbodig lijken omdat het ook al in de laatste twee zit, wat eigenlijk zo is, maar het laat FW versies van DFCP 4 toe die minder dan 2.0 zijn.

2.1.5- Adres van DFCP 4

Ken een adres toe aan de DFCP 4. De toegelaten waardes gaan van 1 tot 255. Dat is belangrijk wanneer het MODBUS protocol wordt gebruikt.

ADDRESS = (12)

2.1.6- ID van DFCP 4

Ken een identificatie toe, via een string, aan de DFCP 4 (max. 63 karakters).

ID = (My House)

2.1.7- Richtlijn voor de berekening van zonsopkomst, zonsondergang en positie van de zon

DFCP 4 kan iedere dag de tijden van zonsopkomst, zonsondergang en positie van de zon (azimut en hoogte) berekenen. De berekende waardes zullen dan in 4 registers geladen worden dat gedefinieerd moeten worden via de **LOCALIZE**, zoals hier beschreven. Voor deze berekeningen moet de data, verwant met de positie, voorzien zijn aan de DFCP 4 (longitude en latitude) samen met de gerelateerde tijdzone (bv. voor Italië is deze waarde 1). Bijkomend dat de DFCP 4 het virtueel punt **V2014** hanteert die het activeert wanneer de huidige tijd binnen het bereik van zonsopkomst naar zonsondergang is en gecorrigeerd wordt, zo nodig, door een hoeveelheid minuten die gedeclareerd kunnen worden in de **SUNRISE** en **SUNSET** parameters.

```
LOCALIZE = ( \
    LONGITUDE = 8.8638,      \
    LATITUDE = 45.3036,      \
    TIMEZONE = 1,           \
    SUNRISE = ( 0, R1 ),     \
    SUNSET = (0, R2 ),       \
    AZIMUTH = R3 ,           \
    ELEVATION = R4 ,         \
)
```

LONGITUDE	toegelaten waardes binnen het bereik -180.0000 tot +180.0000
LATITUDE	toegelaten waardes binnen het bereik -90.0000 tot +90.0000
TIMEZONE	met respect voor Greenwich: toegelaten waardes binnen het bereik -12 tot +12
SUNRISE	minuten die opgeteld of afgetrokken moeten worden van de tijd van zonsopkomst (binnen bereik van -127 tot +127) en de register (optioneel) rapporteert de berekende tijd van zonsopkomst.
SUNSET	minuten die opgeteld of afgetrokken moeten worden van de tijd van zonsondergang (binnen bereik van -127 tot +127) en het register (optioneel) rapporteert de berekende tijd van zonsondergang.
AZIMUTH	een register (optioneel) rapporteert de azimut van de zon. Een waarde, binnen het bereik 0 tot 360, rapporteert de hoek van de zon vanuit het noorden, gemeten in graden met de klok mee. Bijvoorbeeld als azimut = 90 dan betekent dat, dat de zon in het oosten zit.
ELEVATION	een register (optioneel) rapporteert de hoogte van de zon met 2 complementaire waarden. Het register zal een waarde van 65446 tot 65535 als negatieve waarden bezitten en 0 tot 90 als positieve waarden, overeenkomend met een waarde binnen het bereik van -90 tot +90, die de positie van de zon rapporteert vanuit de horizon, gemeten in graden. Een positieve waarde betekent dat de zon boven de horizon staat, negatief onder de horizon. Elevation = 0 betekent dat de zon precies op de horizonlijn staat.

De tijden verwant aan zonsopkomst en zonsondergang zal gerapporteerd worden door specifieke registers met een waarde die overeenkomt met het aantal minuten van de dag, startende van 0:00 (bv. 1439 = 23:59).

Opmerking: De 4 registers kunnen, optioneel, gedeclareerd worden. Als de berekening van zonsopkomst, zonsondergang en positie van de zon niet verplicht zijn, dan kunnen die vermeden worden. In dat geval zal het virtueel punt **V2014** echter toch behandeld worden.

2.1.8- Beheer van verkeerde modules

Ken een virtueel punt toe aan de falende werking van één of meerdere modules.

MODFAIL(V10) = (**I1**, **I3**, **O1**, **O4**)

MODFAIL(V11) = (**I15**)

Opmerking: **MODFAIL** accepteert geen notaties gelijkaardig aan O1..O15.

2.1.9- Opstelling van de outputs

DFCP 4 voert periodisch een statusaanvraag uit naar de outputmodules (beide digitaal als analoog). Als de DFCP 4 een mismatch detecteert tussen de status of de waarde, gelezen uit het veld en de verwante waarde, opgeslagen in het RAM geheugen van de controller, dan moet het een opstelling uitvoeren tussen het veld en de RAM.

Twee opstellingen zijn mogelijk:

- De status of de waarde van de veldoutput zal getransfereerd worden naar het RAM geheugen.
- De status of de waarde in de RAM zal getransfereerd worden naar de veldoutput.

Als standaard voert de DFCP 4 de eerste opstelling uit (van field naar RAM). In sommige gevallen (afhankelijk het type van module en de specifieke applicatie) wordt, zo niet verplicht, de tweede opstelling geprefereerd.

Om de outputs te specificeren die bestuurd moeten worden volgens de opstelling dan moet de vergelijking **RAMtoFIELD** gebruikt worden. Die vergelijking kan enkele outputpunten bevatten waar de waarden of puntintervallen zijn zoals in het volgende voorbeeld.

```
RAMtoFIELD = ( O20.3, \
                 O20.4, \
                 A01, \
                 O21.1..O21.4 )
```

Zelfs al is de tegengestelde optie, **FIELDtoRAM**, de standaard, toch is het mogelijk het te declareren op een expliciete manier zoals in het volgende voorbeeld.

```
FIELDtoRAM = ( O20.1 )
```

2.1.10- Publiceren van interne virtuele punten op de bus

Zoals eerder gezegd in een vorig paragraaf, kunnen de interne virtuele punten van DFCP 4 niet direct gebruikt worden als inputs van de vergelijking binnen de outputmodule. Stel dat twee interne virtuele punten van DFCP 4, bijvoorbeeld **V1** en **V2**, die het openen en sluiten van een rolluik moeten besturen op dezelfde manier wanneer het rolluik bestuurd zou worden door twee knoppen, dan is de volgende vergelijking, die in DFTP geladen moet worden, natuurlijk **niet toegelaten**:

O1.1 = **OV1** | **CV2**

Om dat probleem op te lossen kan een DF4I/V module (4 inputs + 12 virtuele punten) geïnstalleerd worden in de installatie. Zo kan de status van twee virtuele punten van DFCP 4 getransfereerd worden naar twee punten van DF4I/V. Om dat te doen, veronderstellend dat het basisadres 129 is toegekend aan DF4I/V, kunnen de volgende vergelijkingen in het programma van DFCP 4 geplaatst worden.

```
V130.1 = V1
V130.2 = V2
```

Op die manier kopieert de DFCP 4 de status van zijn interne virtuele punten naar de externe virtuele punten. De punten **v130.1** en **v130.2** kunnen dan gebruikt worden als inputs van de vergelijking binnen de DFTP zoals het volgende:

$$o1.1 = ov130.1 \mid cv130.2$$

Het grootste verschil tussen een intern virtueel punt van DFCP 4 en een extern virtueel punt, is dat de interne virtuele punten niet “gepubliceerd” worden op de bus. Daardoor kunnen de modules die hen normaal gebruiken, hun status niet kennen. De “truc”, hieronder beschreven, maakt de interne virtuele punten vrij voor de bus omdat de DF4I/V modules constant de status van hun punten naar de bus versturen.

Het nadeel van deze “truc” is dat één of meerdere DF4I/V modules geïnstalleerd moeten worden, wat de kosten doet stijgen en de complexiteit van het systeem programmeren. Vanaf firmware versie 1.1 kan de DFCP 4 geconfigureerd worden om een geselecteerd aantal interne virtuele punten te “publiceren” op de bus, wat de installatie van DF4I/V modules tegenhoudt. In de praktijk kan DFCP 4 het gedrag van DF4I/V modules simuleren, dus het zenden van de status van sommige virtuele punten naar de bus. Bovendien kunnen die “gepubliceerde” virtuele punten afgedwongen worden door de bus (bijvoorbeeld door DFTouch, DFCL, DFCKII, enz.) en dat maakt de virtuele punten gepubliceerd door de DFCP 4 gelijkwaardig aan die van DF4I/V modules.

De volgende vergelijking binnen de DFCP 4 configuratie moet geschreven worden om het publiceren van een virtueel punt in te schakelen.

$$PUBLICV = (Start_add, Num_V, Mode)$$

waarin:

Start_add	een startadres is en het iedere waarde kan gebruiken binnen het toegelaten bereik van Domino adressen
Num_V	het aantal virtuele punten die gepubliceerd moeten worden is en die een meervoud moeten hebben van 4 of 16 gezien de waarde van Mode. Vanuit een meer theoretisch oogpunt kunnen alle virtuele punten van DFCP 4 gepubliceerd worden maar er moet rekening gehouden worden met de limiet van mogelijke Domino adressen in de installatie.
Mode	aanduidt als de virtuele punten gepubliceerd moeten worden als 4 of 16 punten per adres.

De virtuele punten gepubliceerd naar de bus gaan altijd van V1 tot Vn, waar n dezelfde waarde heeft als Num_V.

Opmerking: als de 16-punt modus gebruikt werd, voor het publiceren van virtuele punten, dan is het nodig om in de module equation editor voor elk gebruikte adres te declareren. Volgens deze regel moet de gebruikte adressen gedeclareerd worden als:

$$DFCP\ 4 = (Address\ of\ the\ used\ virtual\ point)$$

Bijvoorbeeld, stel dat **PUBLICV = (130, 8, 4)**. De DFCP 4 zal op de bus zijn interne virtuele punten van **v1** tot **v8** publiceren. Die 8 punten zullen op de Domino bus te zien zijn als **v130.1 .. v130.4** en **v131.1 .. v131.4**. In detail zal de relatie het volgende zijn:

V in DFCP 4	V op bus
v1	v130.1
v2	v130.2
v3	v130.3
v4	v130.4
v5	v131.1
v6	v131.2
v7	v131.3
v8	v131.4

De relatie kan geverifieerd worden in de Resource TAB van DCP IDE (na het compileren van het programma). Naast de hoeveelheid informatie in deze TAB, staat de volgende lijst gerapporteerd (ter voorbeeld):

```
////////////////////////////////////  
// PUBLIC VIRTUALS  
////////////////////////////////////  
V130: 1, 2, 3, 4,  
V131: 5, 6, 7, 8,
```

Deze lijst toont mooi aan dat er twee busadressen werden gebruikt (130 en 131, zie eerste kolom) en elk adres telt 4 punten. De interne virtuele punten van **v1** tot **v4** en van **v5** tot **v8** zijn toegekend aan die twee adressen.

Deze lijst is heel bruikbaar omdat het ons toelaat mooi te zien vanwaar een intern virtueel punt **v_x** werd gepubliceerd, het is bijvoorbeeld goed te zien dat **v7** was toegekend aan **v131.3**.

Ook omgekeerd, met een gegeven virtueel punt **v_{j.k}** op de bus, kan gemakkelijk zijn verwant intern virtueel punt gevonden worden. Als voorbeeld zien we hier duidelijk dat **v131.2** gerelateerd is aan **v6**.

Om een onnodig verkwisten van Domino adressen te vermijden, om zo het verkeer op de bus niet te laten toenemen, is het aangeraden om enkel de minimum onontbeerlijke virtuele punten te publiceren, punten die effectief gebruikt zullen worden door de outputpunten.

Voorbeeld

Stel dat de applicatie eist om het output level van 4 dimmermodules DFDM door 2 paar knoppen (Up en Down) te reguleren op een touchscreen die geconnecteerd is aan een RS485 poort van de DFCP 4. Dit is een typisch voorbeeld waar de operationele vergelijking in de output module moet zitten (in dit geval de dimmer) omdat het anders niet mogelijk is die functie direct naar het outputadres zelf te schrijven, uit te voeren. De gemakkelijkere manier om die applicatie te realiseren is dat het touchscreen sommige verkrijgbare virtuele punten naar de bus schrijft en dat de dimmermodules een gepaste vergelijking bezit om die punten te gebruiken. Als de touchscreen is verbonden met DFCP 4 dan kan het zonet beschreven 'publiceren' gebruikt worden.

Veronderstel dat we een touchscreen met 8 knoppen op het scherm configureren en dat elke knop een intern virtueel punt activeert van de DFCP 4, laten we zeggen van **v1** tot **v8**. Dan moet het volgende in DFCP 4 gedeclareerd worden.

PUBLICV = (142, 8, 4)

Met deze vergelijking zullen de virtuele punten van **v1** tot **v8** gekopieerd en gepubliceerd worden naar **v142.1 .. v143.4**. In de 4 dimmermodules moet, veronderstellend dat ze een outputadres hebben van 1 tot 4, de volgende vergelijking ingeladen worden.

```
O1.1 = UV142.1 | DV142.2  
O2.1 = UV142.3 | DV142.4  
O3.1 = UV143.1 | DV143.2  
O4.1 = UV143.3 | DV143.4
```

2.1.11- Het uitwisselen van data tussen meerdere DFCP 4 controllers

Als meerdere DFCP 4 controllers zijn geïnstalleerd in de installatie dan is het mogelijk een feature te activeren die het uitwisselen van data toelaat tussen hen. Om dit te doen moet simpelweg ieder DFCP 4 geconnecteerd zijn aan elkaar om zo een RS485 netwerk (gebruikmakend van de COM2 poort) te creëren. Ook iedere DFCP 4 die bij het netwerk hoort moet onderricht worden om zijn data te "publiceren" via gespecificeerde richtlijnen. Daarom zijn er geen meerdere componenten nodig om de RS485 aan de geïnstalleerde DFCP 4 controllers te verbinden.

Bovendien is de uitwisseling van data, zoals daarnet beschreven, ook van toepassing tussen Domino DFCP 4 controllers en Contatto MCP XT controllers, wat dus interactie toelaat tussen de twee systemen.

De informatie die gepubliceerd kan worden op het netwerk zijn virtuele punten en registers waarmee, sinds dat ieder digitale en analoge variabele kan ondersteund worden door deze, bijna alle informatie volgens iedere DFCP 4 of de daarmee verbonden modules getransfereerd kan worden.

Iedere DFCP 4 kan aan een maximum van 125 Words publiceren. Sinds dat iedere register 1 Word aanneemt en dat een Word 16 interne, aangrenzende virtuele punten kan bezitten, zijn de volgende combinaties, als voorbeeld, toegelaten.

- 2000 virtuele punten – 0 registers
- 1000 virtuele punten – 62 registers
- 512 virtuele punten – 93 registers
- 0 virtuele punten – 125 registers

Met andere woorden, het aantal virtuele punten gedeeld door 16, opgeteld bij het aantal registers, moet minder of gelijk zijn aan 125:

$$(nr.V) / 16 + nr.R \leq 125$$

Zoals eerder gezegd moet de feature van data-uitwisseling geactiveerd zijn gedurende het programmeren door één of meerdere configuraties die de hoeveelheid virtuele punten en registers specificeren die gepubliceerd worden op de RS485 poort. Die configuraties zijn zoals het volgende:

```
NETWORK = (Vstart .. Vstop)
NETWORK = (Rstart .. Rstop)
```

Vstart en Vstop betekenen een eerste en een finaal virtueel punt. Ieder gekozen waarde voor Vstart en Vstop zal hoe dan ook geforceerd zijn tot een meervoud van 16 door de compiler van DCP IDE. Bijvoorbeeld als **V3..V9** werd gekozen, zal de compiler er **V1..V16** van maken.

Rstart en Rstop betekenen een eerste en een finaal register. Die twee waarden kunnen ingesteld worden zoals gewenst, maar herinner dat de totale aantal Words die gepubliceerd kunnen worden minder of gelijk moeten zijn aan 125.

Iedere DFCP 4 die bij het netwerk hoort zal in zijn geheugen de informatie, gepubliceerd door andere DFCP 4 controllers, schrijven. Iedere Word zal opgeslagen worden op dezelfde positie die het origineel had, daarom zal de inhoud van register R50 van een DFCP 4, als voorbeeld, ook als R50 door alle DFCP 4 controllers opgeslagen worden. Voor die reden is het vanzelfsprekend dat gepubliceerde Words van elke DFCP 4 verschillend moeten zijn van elkaar. In andere woorden moet superpositie, van de gepubliceerde Words, vermeden worden voor controllers die bij hetzelfde netwerk horen.

Het is ook mogelijk om niet aangrenzende blokken van virtuele punten en registers te publiceren door meerdere **NETWORK** te specificeren, tot een maximum van 8 (als totaal van V-Words en R-Words). Ter voorbeeld kan dezelfde DFCP 4 de volgende code bezitten.

```
NETWORK = (V1 .. V16)
NETWORK = (V513 .. V576)
NETWORK = (V1025 .. V1056)
NETWORK = (R0 .. R8)
NETWORK = (R33 .. R37)
NETWORK = (R50 .. R52)
NETWORK = (R100 .. R100)
NETWORK = (R251 .. R255)
```

Om alleen maar één register te publiceren, specificeer dan gewoon dezelfde waarde voor Rstart en Rstop (zie R100 in het vorig voorbeeld).

Iedere DFCP 4 verwerft op hetzelfde netwerk ook de gepubliceerde informatie van andere componenten, ook al bevat het geen enkele **NETWORK**. Bijvoorbeeld als alleen één DFCP 4 informatie moet verzenden naar andere componenten op het netwerk, maar niet vice versa, dan kan alleen **NETWORK** geactiveerd worden voor de “master” DFCP 4 controller.

Herinner:

- Als een **NETWORK** werd ingevoegd in een DFCP 4, dan moet ook een **ADDRESS** ingevoegd worden en dat laatste moet gebeuren **voor** ieder **NETWORK**
- De gepubliceerde Words door en DFCP 4 moeten verschillend zijn van elkaar om zo superpositie te vermijden
- In ieder component van het netwerk kan tot 8 **NETWORK** bediend worden
- Iedere DFCP 4 verwerft gepubliceerde informatie door andere componenten in hetzelfde netwerk, ook al bezit het geen **NETWORK**
- De data-uitwisseling geldt ook tussen Domino DFCP 4 en Contatto MCP XT controllers.

Voorbeeld:

Stel dat er twee DFCP 4 controllers zijn geïnstalleerd waar ook bij elk van hen 1 DF4I en 1 DF4R zijn geconnecteerd, beide met adres 1. Veronderstel ook dat de applicatie, de outputs van een bus moet controleren via de input van het ene en omgekeerd.

De programma's die dan geschreven moeten worden voor de twee DFCP 4 controllers zijn:

```
// Programma 1 voor DFCP 4
ADDRESS = ( 1 )
NETWORK = (V1..V16)
V1 = I1.1
V2 = I1.2
V3 = I1.3
V4 = I1.4
O1.1 = V17
O1.2 = V18
O1.3 = V19
O1.4 = V20

// Adres van eerste DFCP 4
// Verzendt 16 virtuele punten naar de andere DFCP 4
// Kopieert de lokale inputs naar de eerste 8 virtuele punten

// Kopieert de verkregen virtuele punten naar de outputs

// Programma 2 voor DFCP 4
ADDRESS = ( 2 )
NETWORK = (V17..V32)
V17 = I1.1
V18 = I1.2
V19 = I1.3
V20 = I1.4
O1.1 = V1
O1.2 = V2
O1.3 = V3
O1.4 = V4

// Adres van tweede DFCP 4
// Verzendt 16 virtuele punten naar de andere DFCP 4
// Kopieert de lokale inputs naar de virt. punten om verzonden te worden

// Kopieert de verkregen virtuele punten naar de outputs
```

In de praktijk zal de status van de inputs **I1.1..I1.4** van de module, verbonden met de eerste DFCP 4, gekopieerd worden naar **V1..V4** van dezelfde DFCP 4 en gepubliceerd worden op de netwerken. De tweede DFCP 4 ontvangt de status van die virtuele punten en slaat ze op dezelfde positie **V1..V4** op. De status transfereert, op die manier, naar de outputmodule **O1** die verbonden is met zijn bus. Hetzelfde gebeurt ook op de omgekeerde manier door de virtuele punten **V17..V20**.

2.1.12- Aantal knopen in een DFCP 4 netwerk

In een netwerk gemaakt zoals beschreven in het vorig paragraaf, kunnen er tot 32 DFCP 4 modules geïnstalleerd worden, elk die zijn data publiceert om het zo beschikbaar te maken voor andere componenten op het netwerk. Sinds dat enkel één DFCP 4 contact kan maken met de RS485 communicatielijn, zijnde dit van het seriële type, heeft de publicatie van informatie door alle componenten of knopen binnen het netwerk wat tijd nodig.

Door de bijzondere besturing van het mechanisme, om toegang tot het netwerk te krijgen, kan dus deze keer geoptimaliseerd worden door de DFCP 4 te instrueren met hoeveel deelnemers er zijn van hetzelfde netwerk. Om dat te doen moet gebruik gemaakt worden van de volgende code:

NODESNUM = (num)

Waar num het aantal knopen is en binnen het bereik van 1 tot 32 moet zijn.

De tijd die nodig is voor alle knopen om hun informatie te publiceren, in het slechtste geval 125 Words die gepubliceerd moeten worden aan 115.2 Kbaud, duurt ongeveer minimum 80 milliseconden in het geval van 2 DFCP 4, maximum 1 seconden bij 32 DFCP 4.

Als het aantal knopen niet gespecificeerd zijn door **NODESNUM**, dan zal die parameter als standaard ingesteld staan op 32. In dat geval zal het netwerk hoe dan ook werken maar als bijvoorbeeld het netwerk uit 2 DFCP 4 bestaat, de tijd die nodig is voor een volledige cyclus zal stijgen van 80 milliseconden naar 340 milliseconden (aan 115.2 Kbaud en in het slechtste geval van 125 Words).

De **NODESNUM** is dus optioneel maar het is heel nuttig om de tijd die nodig is om data tussen DFCP 4 controllers uit te wisselen, drastig te doen verlagen wanneer er minder knopen zijn dan de toegelaten maximum waarde.

2.1.13- Scheduler

De Scheduler laat het besturen van een digitale outputpunt (een exclusief virtueel punt) toe in functie tot de huidige dag en tijd. DFCP 4 bezit een timekeeper circuit met reservebatterij om het verliezen van datum en tijd tegen te gaan wanneer de voeding zou worden afgesloten. De transitie van standaard tot zomertijd wordt automatisch door de DFCP 4 gemaakt.

DFCP 4 bezit een complete, wekelijkse Scheduler waarmee zo iedere dag van de week gespecificeerd kan worden met wanneer een specifieke output geactiveerd moet worden en welke niet.

Dat soort programmeren wordt identiek herhaald voor alle weken van het jaar. Het virtueel punt, bestuurd door de Scheduler, moet dan eventueel gerapporteerd worden naar de gewenste, fysische outputs, in combinatie met andere punten als dat nodig is.

De Scheduler, geïntegreerd in de DFCP 4, laat het controleren van 16 verschillende punten ("zones") met elk 8 tijdvakken (8 ON-OFF intervallen) toe voor ieder dag van de week. De beslissing van de Scheduler duurt 1 minuut.

De Scheduler wordt geactiveerd door de volgende code:

SCHEDULER n (Vx)

n is het Scheduler nummer ("zone") en moet in het bereik van 1 tot 16 zijn

Vx is het bestuurd virtueel punt

Als er meer punten zijn die bestuurd moeten worden, dan moet er meer code toegevoegd worden zoals in het volgende voorbeeld.

SCHEDULER 1 (V100)

SCHEDULER 2 (V101)

SCHEDULER 3 (V102)

Verschillend tot de CLOCK vergelijking, die later beschreven zal worden, waar specifieke tijden ingesteld staan in het gebruikersprogramma die enkel veranderd kunnen worden via de herprogrammering van de DFCP 4, laat de scheduler een gemakkelijkere installatie toe van ieder supervisieprogramma of webserver (optioneel) geïntegreerd in DFCP 4. Voor details over het invoeren en aanpassen van planningen, refereer dan naar de specifieke documentatie.

De volgende schermafbeelding illustreert een pagina dat kan ontwikkeld worden voor het beheer van de Scheduler.



2.2- Event triggered vergelijkingen

2.2.1- Logische vergelijkingen

Operators: & (AND), | (OR), ! (NOT), ^ (XOR)

(XOR wordt geëvalueerd door de compiler als volgende: $A \wedge B = !A \& B \mid A \& !B$)

```
O10.3 = I1.1
O2.5 = (I1.1 | I1.2)
V6 = (I4.3 | I8.2) & V4
O1.6 = V100 & I1.3
O1.7 = !I1.3 & I1.2
O1.1 = I2.1 & (I4.3 | I2.4)
O8.1 = V7 ^ I43.2
```

2.2.2- SET – RESET vergelijkingen

Operators:

S	SET op de dalende flank
SP	SET prioriteit op de dalende flank
SL	SET op de stijgende flank
SPL	SET prioriteit op de stijgende flank
R	RESET op de dalende flank
RP	RESET prioriteit op de dalende flank
RL	RESET op de stijgende flank
RPL	RESET prioriteit op de stijgende flank

```
O1.1 = SI1.1 & RI1.2
O1.1 = SI1.1 & RL1.2
O1.1 = SPL1.1 & RL1.2
```

Set/Reset op de dalende flank.

Reset op de stijgende flank: output staat vast op OFF als I1.2 is ON.

Set/Reset op de stijgende flank, maar output staat vast op ON als I1.1 ON is (vermits het gespecificeerd is om een priority commando te zijn).

```
O1.5 = I2.3 & RI2.1 & SI4.4
O1.1 = (SI1.1 | SI1.2) & RI1.3
O1.1 = SI1.1 & RI1.2 & RI1.3
O1.1 = SL1.1 & SL1.2 & RI1.3
```

Input I2.3 is een consent.

Gebruik haakjes: output gaat ON wat I1.1 of I1.2 activeert.

Output gaat OFF wat I1.2 of I1.3 activeert.

Output gaat ON wat BEIDE I1.1 en I1.2 activeert.

2.2.3- TOGGLE vergelijkingen

Operators:

T	TOGGLE op de dalende flank
S	SET op de dalende flank
SP	SET prioriteit op de dalende flank
SL	SET op de stijgende flank
SPL	SET prioriteit op de stijgende flank
R	RESET op de dalende flank
RP	RESET prioriteit op de dalende flank
RL	RESET op de stijgende flank
RPL	RESET prioriteit op de stijgende flank

De termen moeten exclusief gelinkt worden door OR operators. Er kunnen geen 'vrije' inputs gebruikt worden.

O1.1 = TI6.1 | TV6 Output schakelt tussen iedere OFF-ON van I6.1 of V6.

O1.1 = T!I6.1 Output schakelt tussen ON-OFF van de input.

V100 = TV1 | SV2 | RV3 Set en Reset op de dalende flank.

V100 = TV1 | SV2 | RLV3 Aangezien Reset commando op de stijgende flank is, dan staat output vast op OFF als V3 is geactiveerd.

O1.1 = TI1.1 | TI1.2 | SI1.3 | SI1.4 | RI2.1 | RI2.2 Meerdere commando inputs.

2.2.4- COUNTER vergelijkingen

Een counter bestuurt een digitale output als functie tussen de vergelijking van de counterwaarde en een grens. Er kunnen 1024 counters gedefinieerd worden. Elke counter kan bestuurd worden door echte of virtuele inputs, ieder met zijn eigen specifieke functie:

- Eén of meerdere inputs om vooruit of achteruit te tellen. (**S (k)**)
- Eén of meerdere inputs voor de "zeroing" of om in de counter een gedefinieerde waarde te voeren. (**P (z)**, **PP (z)**, **PL (z)**, **PPL (z)**)
- Eén of meerdere inputs om het tellen te stoppen. (**H**, **HP**)

De counter, afhankelijk van de variaties op zijn inputs, zal eerst geüpdatet worden om dan vergeleken te worden met de grenswaarde om zo de output te besturen. Toegelaten operators zijn:

<	lager dan
<=	lager dan of gelijk aan
==	gelijk aan
!=	niet gelijk aan
>	groter dan
>=	groter dan of gelijk aan

Besturingsoperators:

S (k)	Som k naar counter op de dalende flank (bereik k: -32768 tot 32767)
P (z)	Preset counter naar z op de dalende flank (bereik z: 0 tot 65535)
PP (z)	Priority Preset counter naar z op de dalende flank (bereik z: 0 tot 65535)
PL (z)	Preset counter naar z op de stijgende flank (bereik z: 0 tot 65535)
PPL (z)	Priority Preset counter naar z op de stijgende flank (bereik z: 0 tot 65535)
H	Zet counter vast op de huidige waarde op de stijgende flank (Halt)
HP	Het Priority vastzetten van de counter op de huidige waarde op de stijgende flank (Halt)

Alle counters van DFCP 4 staan in een 16 bit formaat wat maakt dat de inhoud van iedere counter in het bereik van 0 tot 65535 zit.

Voor de counter functie kunnen ook de volgende opties gespecificeerd worden:

- **AUTORESET/AUTORELOAD**
- **MIN**
- **MAX**
- **Cn,R** kopieert de waarde van counter naar een register (met dezelfde identifier)
- Variabele parameters

De syntax die het toelaat deze opties te specificeren is het volgende (**Vx** kan elk toegelaten punt zijn):

Vx = Cy,R > 30, AR, MIN, MAX

- **R** betekent dat de counterwaarde steeds gekopieerd wordt in een register (met dezelfde identifier)
- **AR** is de (optionele) waarde voor de autoreset en autoreload om het automatisch in te stellen op 0 om zo het automatisch inladen van de counter mogelijk te maken. Wanneer het vooruit tellen de waarde van **AR** overschrijdt dan zal de counter automatisch ingesteld worden op nul, terwijl wanneer het achteruit tellen onder de waarde 0 gaat, dan zal de counter automatisch opnieuw ingeladen worden met de waarde **AR**. Die waarde kan ook de inhoud van een register zijn (zie de volgende voorbeelden). **Opmerking:** als de autoreset/autoreload waarde niet gespecificeerd werd, dan zal het tellen stoppen op 0 (bij het naar beneden tellen) en op de maximaal toegelaten waarde (bij het naar boven tellen) wat een underflow of overflow op de counter doet voorkomen.
- **MIN** is de minimumwaarde dat kan opgenomen worden door de counter. De standaardwaarde is 0.
- **MAX** is de maximumwaarde dat kan opgenomen worden door de counter. De standaardwaarde is 65535.

De waardes voor de grens, autoreset, minimum, maximum, step en preset kunnen ook de inhoud van registers zijn (zie de volgende voorbeelden).

Als één van de opties **AR**, **MIN** en **MAX** achterwege werden gelaten dan zal de standaardwaarde opgenomen worden. De beschreven opties moeten hoe dan ook gescheiden worden door komma's zoals in de volgende voorbeelden.

Voorbeelden:

V1 = C0 >300 S (2) I1.1 & S (-1) I1.2	Naar boven tellen met step 2, beneden tellen met step 1, V1 gaat ON wanneer counter groter is dan 300.
V1 = C0 >30,50 S (1) I1.1 & S (-1) I1.2	Autoreset/Autoreload: wanneer counter 50 overschrijdt, wordt het naar 0 reset. Wanneer counter onder nul gaat, wordt het herladen met 50.
V1 = C0 >30,50 S (1) I1.1 & S (-1) I1.2	MIN en MAX: het naar boven tellen stopt bij 50 en het beneden tellen stopt bij 5.
V1 = C0 >3,5,1 S (1) I1.1 & S (-1) I1.2	Autoreset/Autoreload en MIN: wanneer de counter 5 overschrijdt, wordt het herladen met 5. Wanneer de counter onder 1 gaat, wordt het herladen met 5.
V15 = C10 , R > 100 S (1) I1.1 & S (-1) I1.2 \ & PL (0) I1.3	Kopieert Counter naar Register: de waarde van C10 wordt gekopieerd naar register R10.
V10 = C1 > R0 , R1 , R2 , R3 S (R4) I1.1 \ & P (R4) I1.3	Variabele parameters.

2.2.5- THRESHOLD vergelijkingen

Een threshold vergelijking bestuurt een digitale output als functie tussen de vergelijking van een analoge waarde (inputmodule, bv. DFAI of register), een Threshold en een Hysteresis. Toegelaten operators zijn:

<	lager dan
<=	lager dan of gelijk aan
==	gelijk aan
!=	niet gelijk aan
>	groter dan
>=	groter dan of gelijk aan

Opties:

- Hysteresis
- Variabele parameters

De Hysterese heeft een verschillende betekenis afhankelijk van de operator:

<	OUT gaat ON wanneer AI < T en OUT gaat OFF wanneer AI >= (T + H)
<=	OUT gaat ON wanneer AI <= T en OUT gaat OFF wanneer AI > (T + H)
==	OUT gaat ON wanneer AI == T en OUT gaat OFF wanneer AI > (T + H of wanneer AI < (T - H)
!=	OUT gaat OFF wanneer AI == T en OUT gaat ON wanneer AI > (T + H) of wanneer AI < (T - H) . Dat gedrag is complementair met de vorige zaak.
>	OUT gaat ON wanneer AI > T en OUT gaat OFF wanneer AI <= (T - H)
>=	OUT gaat ON wanneer AI >= T en OUT gaat OFF wanneer AI < (T - H)

Threshold en Hysteresis moeten in het bereik van 0 tot 65535 zijn. Andere toegelaten operators: AND (&) en OR (|).

Voorbeelden:

```

O1.1 = AI1 >= 240,2
V2 = AI1 == 40 | AI2 == 30
V2 = AI1 == 40,5
O1.4 = AI1 < 128 & AI1 > 30

O1.5 = AI1 < 600 & R50 >= 30
O1.1 = AI1 > R51, R52 & AI1 < 820,5
V8 = R51 > 40,5

```

2.2.6- TIMER vergelijkingen

Een timer vergelijking bestuurt een digitale output als functie van twee delay tijden. DFCP 4 laat het definiëren van max. 512 timers toe. De timer resolutie is 0.1s en het bereik van de tijd is 0 tot 6553.5s (1h:49'). De gespecificeerde tijden in de timer zijn bedoeld om vermenigvuldigd te worden door 10 ($T_e = 100$ betekent 10 seconden).

De input die de timer doet starten is de “trigger” input en werkt altijd op de dalende flanken.

Sleutelwoorden:

TIMER	Standaardtimer
TIMERP	Geen re-triggerable Pulse timer (monostabiel)
TIMERPR	Re-triggerable Pulse timer (monostabiel)

Uitgebreide besturingsoperators:

H	Zet de timer vast op de huidige waarde op de stijgende flank (Halt)
Z	Nul, dwing het aflopen van de huidige timerwaarde (als die loopt) op de dalende flank
ZL	Nul, dwing het aflopen van de huidige timerwaarde op de stijgende flank

Opmerking: als de Zero on the level commando actief is dan wordt de trigger status getransfereerd naar de output zonder vertragingen. De prioriteitssequentie voor de besturingen van de timer staat bepaald als Halt, Zero en Trigger. Opties:

➤ Variabele parameters

```

O1.1 = TIMER(I2.4, 30, 10)    3s vertraging, bij de activatie en 1s bij de deactivatie.

V23 = TIMER(!I1.1, 0, 23)    Output is complementair ten opzichte van de trigger input, met 2.3s vertraging
                               bij de output deactivatie.

O1.1 = TIMER(I2.5 & H15.1 & Z15.2, 90, 50)    Trigger, Halt en Zero: I5.1 halts de timer,
                                               I5.2 dwingt het aflopen van de huidige, lopende tijd af.
                                               Als de timer in stand-by staat, dan hebben Halt en
                                               Zero geen effect.

O1.1 = TIMERP(I1.1, 0, 20)    2s pulse bij de activatie van de trigger input; geen actie bij de deactivatie van
                               de input.

O1.1 = TIMERP(I1.1, 10, 20)   2s pulse vertraagd met 1s van de activatie van de trigger input.
O1.1 = TIMERPR(I1.1, 0, 20)   Re-triggerable 2s pulse (berekend van de laatste activatie van de trigger).

O1.1 = TIMERPR(I1.1, 10, 20)  Re-triggerable 2s pulse vertraagd met 1s van de laatste activatie van de
                               trigger input.

O1.1 = TIMER(I1.1, R47, R48)  Variabele parameters.

```


2.2.7- Vergelijkingen voor wiskundige en logische berekeningen

Toegelaten MATH en LOGIC operatoren:

MATH		LOGIC	
Symbool	Functie	Symbool	Functie
+	Sum	&	AND
-	Subtract		OR
*	Multiply	^	XOR
/	Divide	P ()	Preset

Preset opties:

- P** Preset op de dalende flank: laadt de specifieke waarde op de dalende flank van de besturingsinput in.
PL Preset op de stijgende flank: laadt de specifieke waarde in en zet het resultaat vast op die waarde

Opmerkingen: Preset op de stijgende flank is altijd een prioriteitsbesturing met betrekking tot elk andere berekening in de vergelijking en tot een Preset op de dalende flank. Als meerdere Presets op de stijgende flank gespecificeerd zijn in dezelfde vergelijking, wordt de hoogste prioriteit toegekend van de linker tot de rechterkant in de vergelijking.

Wanneer een Preset op de stijgende flank actief is dan staat de berekening vast op de waarde die ingeladen is door de Preset besturing zelf. Als de Preset op dalende flank is, het resultaat van de vergelijking zal dan hetzelfde zijn als het ene ingeladen door de Preset besturing zelf totdat een nieuwe variatie zich voordoet in de termen binnen de vergelijking. Elke term betrokken in een berekening van een vergelijking is een 16bit nummer. De partiële resultaten worden geëvalueerd als 32bit nummers, maar het finaal resultaat zal afgeknot worden naar de minder significante 16 bits.

Met uitzondering wat er eerder werd gezegd over de Preset besturing is er **geen prioriteit in de berekening van een MATH/LOGIC vergelijking**: dat zal geëvalueerd worden in dezelfde sequentie zoals de vergelijking was geschreven, van links naar rechts. **Haakjes zijn niet toegelaten.**

```
AO1 = AI1 + 128
R12 = AI1 + 12 & 0x00F0 + R1 & P(30) I23.4
R54 = R52 / R53 + R54 * 2
```

Een wiskundige vergelijking kan ook alleen gemaakt worden door één of meerdere Preset termen. Dat is nuttig om een waarde in een registers of naar een output van de activatie (of deactivatie) van een besturingsinput in te laden.

```
R0 = P(1527) V1
R1 = P(0x1AB7) I1.2 & P(0) !I1.2
AO23 = P(12000) V148 & P(0b11000011) I12.1 & PL(0) !I32.3
```

Exclusief voor echte outputs (dus registers en counters zijn uitgesloten) is het mogelijk om meerdere Presets te definiëren die geactiveerd zijn door hetzelfde digitale punt. In dat geval, bij de activatie van het gemeenschappelijk digitaal punt, zullen de waarden, gedefinieerd door de Preset, sequentieel verzonden worden naar de output in de volgorde hoe ze in de vergelijking ingebracht werden (van links naar rechts). Hier zijn er wat voorbeelden.

Wanneer **V1** ON is, dan zullen de waardes 10, 20 en 30 sequentieel verzonden worden naar de output:

```
AO1 = P(10) V1 & P(20) V1 & P(30) V1
```

Wanneer **V1** ON is, dan zullen de waardes 10, 20 en 30 sequentieel verzonden worden naar de output. Wanneer **V2** ON is, dan zullen de waardes 30 en 40 sequentieel verzonden worden naar de output:

```
AO1 = PL(10) V1 & P(20) V1 & P(30) V2 & P(40) V2
```

Wanneer **V1** ON is, dan zal enkel waarde 20 verzonden worden naar de output omdat het een Preset op stijgende flank is.

```
AO1 = P(10) V1 & PL(20) V1 & P(30) V1
```

Wanneer **v1** ON is, dan zal enkel waarde 10 verzonden worden naar de output omdat de Preset op stijgende flank helemaal links in de vergelijking staat.

A01 = PL(10)**v1** & PL(20)**v1** & P(30)**v1**

2.2.8- Vergelijkingen voor aanmaak binaire code

Sleutelwoorden:

SEND_n (Tr) Verzendt de specifieke code naar output **n** bij de activatie van de gerelateerde input (of als de deactivatie vervolledigd is), met refresh time **Tr** seconden (wanneer meerdere inputs geactiveerd worden)

SENDER_n (Tr) Verzendt de specifieke code naar register **R_n** bij activatie van de gerelateerde input (of als de deactivatie vervolledigd is), met refresh time **Tr** seconden (wanneer meerdere inputs geactiveerd worden)

De verzendingscode (Bx) moet in het bereik van 0 tot 255 zijn. Als de refresh time weggelaten werd dan zal het ingesteld staan op 2 seconden. De refresh time moet in het bereik van 1 tot 254 seconden zijn. Het is mogelijk om refresh uit te schakelen door het de waarde 255 te geven. In dat geval zal de verzendingscode altijd verwant zijn met de laatste verandering van één van de inputs, opgesomd in het SEND blok.

De inputpunten die het verzenden van de gerelateerde binaire code veroorzaken kunnen echte en virtuele zijn. Die kunnen ook altijd aangevuld worden.

Er kunnen tot 16 onafhankelijke SEND blokken gedefinieerd worden.

```
SEND4 (5) = ( I1.1, B001, \
               I1.2, B002, \
               V354, B003, \
               !I4.3, B006, \
               !V450, B129 \
            )
```

```
SENDER123 (2) = ( I5.2, B001, \
                  V100, B002, \
                  V101, B003, \
                  !V470, B004, \
                  !V480, B005 \
                )
```

Opmerking: komma's zijn verplichte symbolen.

2.2.9- Vergelijkingen voor het opnemen van statusveranderingen (EVENT)

Deze functie laat het opslaan, in chronologische volgorde, van statusveranderingen van zowel echte inputpunten als virtuele punten toe die gespecificeerd zijn in het EVENT blok. Elke statusverandering zal samen opgeslagen worden op:

Day/Month Hour:Minutes:Seconds

De EVENT functie specificeert de DFCP 4 als de OFF-ON of de ON-OFF statusverandering, of zelfs beide opgeslagen moeten worden. De EVENT functie zal ook automatisch de zogenoemde "system events" registreren wat het falen en herstellen van de modules en bus zijn. Tot 2048 events kunnen opgeslagen worden in de RAM. Aangezien het deel van de RAM waar deze events worden opgeslagen een reservebatterij bezit, zullen de events ook opgeslagen blijven wanneer de stroom het zou begeven.

Sleutelwoorden:

EVENT	Creëert de eventlijst (vaste buffer): wanneer de buffer vol is dan accepteert het geen andere events meer (op die manier bevat de lijst de eerste 2048 events sinds de laatste opkuis van de buffer)
EVENTC	Creëert de eventlijst (flexibele buffer): wanneer de buffer vol is dan schrijft het over de oude events (op die manier bevat de lijst de laatste 2048 events)

Niet meer dan 1 EVENT blok kan gedeclareerd worden in dezelfde DFCP 4 controller. Als de buffer vol is (EVENT) of als de oudere events overschreven zijn (EVENTC), dan zal het virtuele punt **V2008** geactiveerd worden om dit voorval te rapporteren.

```
EVENTC = (
    V1, ON,          \      Opstart blok, flexibele buffer
    V2, OFF,         \      Event 1, bij 0-1 verandering van V1
    I3.2, ON, OFF    \      Event 2, bij 1-0 verandering van V2
)                    \      Event 3, bij beide 0-1 en 1-0 veranderingen van I3.2
```

Opmerking: komma's zijn verplichte symbolen.

2.2.10- Vergelijkingen voor het opnemen van waardeveranderingen (LOG)

Deze functie laat het opslaan, in chronologische volgorde, de verandering van de waarden, teruggestuurd door inputmodules of registers toe die gespecificeerd staan in de LOG blok. Verandering van de waarde betekent exclusief een verandering van iedere waarde tegenover een ander, op voorwaarde dat de nieuwe waarde niet nul is tenzij het niet uitdrukkelijk zo verklaard werd. In andere woorden, iedere verandering van nul tot elk andere waarde, of van iedere waarde naar een ander (maar niet nul) zal geregistreerd worden, terwijl een verandering van elke waarde naar nul niet, tenzij het niet uitdrukkelijk gedeclareerd werd in de LOG blok. Bijvoorbeeld:

- Een verandering van 0 naar 287 zal geregistreerd worden
- Een verandering van 287 naar 584 zal geregistreerd worden
- Een verandering van 584 naar 321 zal geregistreerd worden
- Een verandering van 321 naar 0 zal niet geregistreerd worden, tenzij het niet uitdrukkelijk werd gedeclareerd

Deze functie is nuttig om bijvoorbeeld de codes van transponders die een toegang tot een gebouw besturen op te nemen. In de LOG blok kan zowel inputadressen als registers gespecificeerd worden. Iedere waardeverandering zal samen opgeslagen worden naar:

Day/Month Hours:Minutes:Seconds

Tot 1024 16 bit waardes (of codes) kunnen opgeslagen worden in de RAM van de DFCP 4. Aangezien het deel van de RAM waar deze events opgeslagen worden, een reservebatterij bezit, zullen de events ook opgeslagen blijven wanneer de voeding het zou begeven.

Sleutelwoorden:

LOG	Creëert de waardenlijst (vaste buffer): wanneer de buffer vol is dan accepteert het geen andere waardes meer (op die manier bevat de lijst de eerste 1024 waardes sinds de laatste opkuis van de buffer)
LOGC	Creëert de waardenlijst (flexibele buffer): wanneer de buffer vol is dan schrijft het over de oude waardes (op die manier bevat de lijst de laatste 1024 waardes)

Optie:

ZERO Verklaart dat, voor de gerelateerde input of register, ook veranderingen van elke waarde naar nul opgenomen moeten worden

Niet meer dan 1 LOG blok kan gedeclareerd worden in dezelfde DFCP 4 controller. Als de buffer vol is (LOG) of als de oudere waarden overschreven zijn (LOGC), dan zal het virtuele punt **V2009** geactiveerd worden om dit voorval te rapporteren.

```
LOGC = (
    AI47:2, \ opstart blok, flexibele buffer
    AI3, ZERO, \ veranderingen van input AI47 kanaal 2, veranderingen naar nul
    R230, ZERO, \ uitgesloten
    R321 \ veranderingen van input AI3 kanaal 1, veranderingen naar nul
           \ uitgesloten
           \ veranderingen van register R230, veranderingen naar
           \ nul uitgesloten
           \ veranderingen van register R321, veranderingen naar
           \ nul uitgesloten
)
```

Opmerking: komma's zijn verplichte symbolen.

2.2.11- Beheer van de DALI module (DFDALI)

Deze vergelijking versimpelt het beheer van de DFDALI module, vooral in systemen waar een automatische helderheidsregelaar geïmplementeerd moet worden. De syntax van de DALI vergelijking is het volgende:

AOUT = **DALI** (Code, AIN(offset))

AOUT	output (adres van DFDALI die beheerd moet worden) of register waar het resultaat van de vergelijking naar verzonden zal worden
Code	DALI commando type (uitzending, enkele ballast of groep)
AIN	analoge input diens waarde naar de DFDALI module of register, die de waarde bevat die getransfereerd moet worden
offset	waarde of register die een waarde bezit in het bereik van -100 tot 100. Die waarde zal algebraïsch toegevoegd worden aan de AIN . Dat is nuttig om bijvoorbeeld wanneer de helderheidsregelaar van een unieke sensor gedifferentieerd moet worden tussen rijen van lampen, gezien de afstand van de ramen

Bij iedere verandering van één van de **AIN** waardes binnen de **DALI** blok, zal de gerelateerde waarde getransfereerd worden naar de specifieke DFDALI module en via dat ene naar alle ballasten, één ballast of een groep afhankelijk van de waarde Code.

De toegelaten Code, de bestemmingen van DALI, zijn de volgende:

Code = 0xXX	directe declaratie van de code in hexadecimaal formaat. Bijvoorbeeld 0x81 betekent, volgens de specificaties van DALI, dat AIN naar groep 1 verzonden moet worden
Code = B1..B32	gelijk aan 0x01..0x20, voor commando's naar één ballast
Code = G1..G16	gelijk aan 0x81..0x90, voor groepcommando's
Code = ALL	gelijk aan 0x00, voor uitzendcommando's

Voorbeeld:

```

Ao1 = DALI ( G1, AI44,      \      // Zendt AI44 naar Groep 1
             G2, R1,       \      // Zendt R1 naar Groep 2
             G3, R1(10),   \      // Zendt R1 verhoogd met 10 naar Groep 3
             G2, R1(-10),  \      // Zendt R1 verlaagd met 10 naar Groep 4
             B1, AI45,     \      // Zendt AI45 naar Ballast 1
             B2, R5,       \      // Zendt R5 naar Ballast 2
             B3, R6(10),   \      // Zendt R6 verhoogd met 10 naar Ballast 3
             B4, R6(-10),  \      // Zendt R6 verlaagd met 10 naar Ballast 4
             ALL, R100     \      // Zendt R100 naar alle Ballasten (broadcast)
)

```

Deze commando's zullen verzonden worden, in dit voorbeeld, naar een DFDALI module geadresseerd met 1 (AO1).

Opmerking: komma's zijn verplichte symbolen.

2.3- Tijd triggered vergelijkingen

2.3.1- Scheduler vergelijkingen

De scheduler vergelijking bestuurt een digitale output als functie van de specifieke ON/OFF tijd of datum. DFCP 4 bezit een timekeeper met reservebatterij om het verlies van datum en tijd te vermijden wanneer het losgemaakt wordt van de hoofdvoeding. De transitie van standaard naar zomertijd wordt automatisch gemaakt door de DFCP 4, waarvoor dus een tussenkomst van de gebruiker niet nodig is.

De tijden gespecificeerd in de scheduler vergelijkingen kunnen dagelijkse of wekelijkse tijden zijn. De scheduler datums kunnen jaarlijkse of absolute datums zijn.

Sleutelwoorden:

CLOCK	bestuurt de output in functie van de huidige tijd
DATE	bestuurt de output in functie van de huidige datum

Opties:

- **Variable daily scheduling times** gespecificeerd in een register (Rx) of in een Word (@WORD x) bevattende een nummer in het bereik van 0 tot 1439, corresponderend aan het nummer van de minuten van de dag startend van 0:00 (1439 = 23:59). De formule die het nummer verwant aan de tijd hh:mm geeft, is het volgende: $(hh \times 60) + mm$
- **Variable weekly scheduling times** gespecificeerd in een register (Rx) of in een Word (@WORD x) bevattende een nummer in het bereik van 0 tot 10079, corresponderend aan het nummer van de minuten van de week startend van 0:00 van maandag (10079 = 23:59 van zondag). De formule die het nummer verwant aan de tijd DW:hh:mm geeft, veronderstellend voor de dagen van de week (DW) MON=0...SUN=6, is het volgende: $(DW \times 1440) + (hh \times 60) + mm$
- **Variable yearly dates** gespecificeerd in een register (Rx) of in een Word (@WORD x) bevattende een nummer in het bereik van 1 tot 372, corresponderend aan de dag van het jaar startend van 1 januari (372 = december 31). De formule die het nummer verwant aan de dag DD (1-31) van de maand MM (1-12) geeft, is het volgende: $(MM-1) \times 31 + DD$
- **Variable absolute dates** gespecificeerd in een register (Rx) of in een Word (@WORD x) bevattende een nummer in het bereik van 1 tot 37200, corresponderend aan de dag van de eeuw startend van 1 januari 00 (2000) (37200 = 31 december 99 (2099)). De formule die het nummer verwant aan de dag DD (1-31) van de maand MM (1-12) van het jaar YY (0-99) geeft, is het volgende: $(372 \times YY) + (MM-1) \times 31 + DD$

Opmerkingen:

- het argument x van de notatie @WORD x kan in het bereik van 0 tot 65535 liggen. Dit is waar, tenzij anders gespecificeerd, enkel voor CLOCK en DATE vergelijkingen.
- 24:00 is niet toegelaten. Gebruik in plaats daarvan 00:00, met het idee dat het ochtend is op die specifieke dag.

O1.1 = **CLOCK**(8:15, 17:30)

Output is iedere dag ON van 8:15 tot 17:30 (dagelijkse planning).

V3 = **CLOCK**(MON:8:00, FRI:20:00)

Output is ON van maandag 8:00 tot vrijdag 20:00 (wekelijkse planning).

O3.2 = **DATE**(31/07, 02/09)

Output is ON van 31 juli tot 2 september (jaarlijkse planning).

O3.2 = **DATE**(31/07/07, 02/09/08)

Output is ON van 31 juli 2007 tot 2 september 2008 (absolute planning).

V4 = **CLOCK**(TUE:8:00, TUE:12:00) | \
CLOCK(THU:14:30, SAT:00:00)

Output is ON op dinsdag 8:00 tot 12:00 en ook van donderdag 14:30 tot zaterdag 0:00.

V6 = **DATE**(12/01/06, 15/01/06) | \
DATE(20/01/06, 22/01/06)

Output is ON van 12/01/06 tot 15/01/06 en van 20/01/06 tot 22/01/06.

V8 = **DATE**(12/01/06, 15/01/06) & \
CLOCK(10:00, 17:00)

Output is ON van 10:00 tot 17:00 maar enkel op die specifieke dagen.

O1.1 = **CLOCK**(XX:R0, XX:R1)

Schakelt dagelijks ON bij tijd gespecificeerd door register R0 en schakelt dagelijks OFF bij tijd gespecificeerd door R1. Bijvoorbeeld als R0=675 en R1=1280 dan zal de output iedere dag ON zijn van 11:15 tot 21:20.

O1.1 = **CLOCK**(XX:@WORD32770, XX:@WORD32771)

Zoals de vorige vergelijking maar de tijden zijn gespecificeerd door Words.

O1.1 = **CLOCK**(R0, R1)

Schakelt wekelijks ON bij de tijd gespecificeerd door register R0 en schakelt wekelijks OFF bij de tijd gespecificeerd door R1. Bijvoorbeeld als R0=675 en R1=6780, dan zal de output iedere week van maandag 11:15 tot vrijdag 17:00 ON zijn.

O1.1 = **CLOCK**(@WORD32770, @WORD32771)

Zoals de vorige vergelijking maar de tijden zijn gespecificeerd door Words.

O1.1 = **DATE**(R0/XX, R1/XX)

Schakelt ieder jaar ON bij de datum gespecificeerd door register R0 en schakelt ieder jaar OFF bij de datum gespecificeerd door R1. Bijvoorbeeld als R0=48 en R1=82, dan zal de output ieder jaar van 17 februari tot 20 maart ON zijn.

O1.1 = **DATE**(@WORD32770/XX, @WORD32771/XX)

Zoals de vorige vergelijking maar de datums zijn gespecificeerd door Words.

O3.2 = **DATE** (**R3**, **R4**)

Schakelt ON bij de absolute datum gespecificeerd door register R3 en schakelt OFF bij de datum gespecificeerd door R4. Als R3=675 en R4=6780, dan zal de output van 24 oktober 2001 tot 22 maart 2018 ON zijn.

O3.2 = **DATE** (@**WORD32776**, @**WORD32777**)

Zoals de vorige vergelijking maar de absolute datums zijn gespecificeerd door Words.

2.4- Macro

Een MACRO is een sequentie van vergelijkingen dat ingebracht kan worden in meerdere punten van de DFCP 4 source programma door de MACRO simpelweg zelf op te roepen. De MACRO moet eerst gedefinieerd worden in de Macros TAB van de DFCP IDE software, dan kan het herroepen worden in het programma zoveel u maar wilt (in de Equations TAB van DCP IDE).

Elke MACRO kan verschillende argumenten (parameters) hebben. Het aantal argumenten moeten dezelfde zijn in de MACRO definitie en in iedere vermelding. De compiler zal de argumenten linken in de vermelding met de argumenten in de MACRO definitie, in dezelfde volgorde zoals ze werden geschreven.

Het is belangrijk om te begrijpen dat:

- MACRO van toepassing is op beide programma's van de DFCP 4 en de Domino modules.
- MACRO een hulpprogramma is van de compiler, niet van de DFCP 4. Met andere woorden laat de compiler iedere vermelding naar een MACRO 'ontploffen' in de gespecificeerd vergelijkingen in de definitie van dezelfde MACRO wat simpelweg ieder argument in de definitie vervangt met het gerelateerd argument van de vermelding

Tot 256 MACRO's, elk tot 32 argumenten, kunnen gedefinieerd worden in een DFCP 4 programma.

De definitie van een MACRO wordt geopend door het sleutelwoord **MACRO** gevolgd door de gekozen naam voor de MACRO en binnen de haakjes komen de argumenten. De definitie van een MACRO wordt gesloten door het sleutelwoord **ENDMACRO**.

De verplichte vergelijkingen moeten binnen het blok horen in rekening houdend dat de argumenten in de MACRO definitie (dat variabele parameters zijn omdat ze van de ene vermelding naar het ander veranderen) niet dezelfde namen kunnen hebben als diegene die al gereserveerd zijn door de parameters of sleutelwoorden van de DFCP 4.

Het volgend voorbeeld definieert een MACRO, genaamd OUTCTRL dat bij de activatie van een virtueel punt de 4 outputpunten van een DF4RP module naar ON schakelt op een sequentiële manier en met 1 seconde vertraging van het ene punt naar het ander. Bij deactivatie van het virtueel punt zullen de 4 punten in omgekeerde sequentie naar OFF schakelen. De MACRO definitie is het volgende:

```
MACRO OUTCTRL (OUT, VIRT)
  OUT.1 = TIMER (VIRT, 0, 30)
  OUT.2 = TIMER (VIRT, 10, 20)
  OUT.3 = TIMER (VIRT, 20, 10)
  OUT.4 = TIMER (VIRT, 30, 0)
ENDMACRO
```


Als 6 DF4RP outputmodules geïnstalleerd moeten worden in de installatie en ze moeten werken zoals daarnet beschreven, bestuurd door verschillende virtuele punten, dan kan de MACRO, vooraf gedefinieerd, meerdere keren in het programma opgeroepen worden zoals het volgende:

```
OUTCTRL (O20 , V50)
OUTCTRL (O21 , V51)
OUTCTRL (O22 , V52)
OUTCTRL (O23 , V53)
OUTCTRL (O24 , V54)
OUTCTRL (O25 , V55)
```

Zoals u ziet wordt een verschillende argumentenlijst doorgestuurd bij iedere oproep. De compiler zal dit programma doen “exploderen” in een sequentie van vergelijkingen wat moeilijker is om te interpreteren en aan te passen. Met andere woorden gaat de compiler de weinige programmeerlijnen van het vorig voorbeeld vertalen zoals het volgende:

```
O20.1 = TIMER ( V50 , 0 , 30 )
O20.2 = TIMER ( V50 , 10 , 20 )
O20.3 = TIMER ( V50 , 20 , 10 )
O20.4 = TIMER ( V50 , 30 , 0 )
O21.1 = TIMER ( V51 , 0 , 30 )
O21.2 = TIMER ( V51 , 10 , 20 )
O21.3 = TIMER ( V51 , 20 , 10 )
O21.4 = TIMER ( V51 , 30 , 0 )
O22.1 = TIMER ( V52 , 0 , 30 )
O22.2 = TIMER ( V52 , 10 , 20 )
O22.3 = TIMER ( V52 , 20 , 10 )
O22.4 = TIMER ( V52 , 30 , 0 )
O23.1 = TIMER ( V53 , 0 , 30 )
O23.2 = TIMER ( V53 , 10 , 20 )
O23.3 = TIMER ( V53 , 20 , 10 )
O23.4 = TIMER ( V53 , 30 , 0 )
O24.1 = TIMER ( V54 , 0 , 30 )
O24.2 = TIMER ( V54 , 10 , 20 )
O24.3 = TIMER ( V54 , 20 , 10 )
O24.4 = TIMER ( V54 , 30 , 0 )
O25.1 = TIMER ( V55 , 0 , 30 )
O25.2 = TIMER ( V55 , 10 , 20 )
O25.3 = TIMER ( V55 , 20 , 10 )
O25.4 = TIMER ( V55 , 30 , 0 )
```

Dit voorbeeld toont goed aan hoe je het hulpprogramma van MACRO gebruikt om het blok van repetitieve vergelijkingen uit te voeren waar enkel sommige parameters veranderen.

Ter aanvulling, en dit is opnieuw een groot voordeel bij het gebruiken van het hulpprogramma van de MACRO, een verplichte verandering in de werking van het systeem kan via een kleine aanpassingen aan de MACRO definitie.

3- SCRIPT

3.1- Overzicht

Scripts laten het toe om delen van het programma te implementeren dat sequentieel zal uitgevoerd worden door DFCP 4. Elke script kan opgestart worden ("triggered") door een event of het kan uitgevoerd worden voor een specifieke tijdsperiode. Ieder gedefinieerd script moet genummerd worden. Tot 127 scripts kunnen gedefinieerd worden. De scripts mogen enkel gebruikt worden om functies uit te voeren die niet gerealiseerd kunnen worden door de standaardvergelijkingen van de DFCP 4. De duur van een script moet lager dan 500msec zijn, zo niet zal de DFCP 4 de uitvoering onderbreken (en zal het, het gerelateerd virtueel punt V2004 instellen). Wees daarom waakzaam voor de loops in een script.

Sleutelwoord	Betekenis
SCRIPT... ENDSCRIPT	Omvat de instructies die bij het script horen; SCRIPT verklaart de start en ENDSCRIPT het einde
TRIGGER	Specificeert het event dat het SCRIPT opstart of duur tijdsperiode van uitvoering in seconden
EXIT	Verplicht het verlaten van het script
VAR	Declareert een lokale variabele die niet met andere scripts wordt gedeeld
GLOBAL VAR	Declareert een globale variabele die met andere scripts wordt gedeeld
EXTERN VAR	Dit specifieke variabele werd gedeclareerd als globaal in een ander script
&, , ^, !	Logische operators (haakjes zijn niet toegelaten en niet meer dan één operatie per lijn is toegelaten)
+, -, *, /, =	Wiskundige operators (haakjes zijn niet toegelaten en niet meer dan één operatie per lijn is toegelaten)
IF...THEN...ELSE... ENDIF	IF en ENDIF omvatten het blok. Een IF moet altijd gesloten worden met een ENDIF.
>, >=, ==, <, <=, !=	Vergelijkingsoperators (groter dan, groter dan of gelijk aan, gelijk aan, minder dan, minder dan of gelijk aan, niet gelijk aan)
CARRY	Bit (flag) vanwaar waarde 1 is als het resultaat van de vorige operatie de waarde 65535 overschrijdt (overflow) of als het resultaat van de vorige operatie negatief is (underflow) of als een deling door 0 plaatsvond. De waarde van deze bit is in alle andere gevallen 0.
ZERO	Bit (flag) vanwaar de waarde 1 is als het resultaat van de vorige operatie nul is. De waarde van deze bit is in alle andere gevallen 0.
DEFINE	Ken een naam toe aan een variabele, parameter of een constante.
GOTO	Onvoorwaardelijke sprong
CALL	Spring naar een subroutine of functie (wat een deel is van een script). Vanuit een script is het mogelijk om een subroutine binnenin een ander script op te roepen.
SUB...ENDSUB	Omvat een blok van instructies als subroutine of als functie. De subroutines dat gedeclareerd waren in een script kunnen "gezien" en gebruikt worden door ieder ander script.
RET	Uitgang van een subroutine of functie.
BIT (x)	Declareert dat parameter x van een subroutine, functie of de waarde teruggekeerd door een functie een bit is. De declaratie BIT(x) is enkel van toepassing op subroutines of functies.
WORD (x)	Stuurt het nummer van de Word waar punt x is mapped.
[ptr]	Pointer. Het stuurt de inhoud van de Word waarvan het adres de waarde is van de variabele binnen de vierkante haakjes (ptr in dit geval). Met andere woorden wijst de ptr naar het Word-adres en [ptr] is de inhoud van de aangewezen Word (zie voorbeelden)
@WORD k	Het keert de inhoud van het Word k terug, waar k een constante waarde is binnen het bereik van 0 tot 32767.
@RAM k	Het keert de inhoud van twee opeenvolgende bytes startende van adres k terug waar k een constante waarde is binnen het bereik van 0 tot 65535.
SWAP (x)	Verwisselt de hoge byte met de lage byte van de specifieke Word (x)
RANDOM (0)	Functie dat een willekeurig 16 bit nummer terugstuurt
BMASK (x)	Functie dat een 16 bit nummer terugstuurt die, in zijn binaire formaat, enkel één bit ingesteld op 1 heeft op de positie van (x-1)%16 (dat betekent (x-1) module 16). Deze functie is nuttig voor het werken met bits.

Bijna alle notaties die bij de vergelijkingssyntax van de DFCP 4 horen kunnen gebruikt worden in de scripts. Bijvoorbeeld de volgende notaties:

```
IF AI1:2 > 230; THEN.....  
AO4 = 197  
R54.1 = 1  
IF I81.1 == 1; THEN.....  
O34.7 = 0  
V781 = 1  
IF V542 == 0; THEN.....
```

Raadpleeg de volgende pagina's om meer voorbeelden te zien van toegelaten notaties.

Opmerking:

1. De schrijfp opdrachten naar de outputs en registers zullen op een opeenvolgende volgorde uitgevoerd worden, in dezelfde volgorde zoals ze in het script staan.
2. De sleutelwoorden kunnen zowel in hoofdletters als kleine letters geschreven worden
3. Meerdere instructies (statements) op dezelfde lijn moeten gescheiden worden via het symbool “;”
4. Wanneer scripts worden geschreven, gebruik de tab om het lezen van de script zelf te verbeteren, zie voorbeeld voor meer details.

3.2- Sleutelwoorden en syntax

3.2.1- Het gebruiken van de TRIGGER

De sleutelwoorden SCRIPT en ENDSRIPT “omringen” het script. SCRIPT moet gevolgd worden door een nummer binnen bereik van 1 tot 127. TRIGGER specificeert het event die de uitvoering van het script doet triggeren of hoeveel keer het uitgevoerd moet worden. Sleutelwoord EXIT dwingt het verlaten van het SCRIPT.

De events die de scripts triggeren kunnen alleen echte inputs (direct of aangevuld) of virtuele punten (direct of aangevuld) zijn. Het is toegelaten, in hetzelfde programma van DFCP 4, een script te hebben die getriggered wordt door een echt of virtueel punt en een ander script die getriggered wordt door hetzelfde maar een aangevuld punt. Op die manier is het mogelijk om een script uit te voeren bij de activatie van een punt en een ander script bij de deactivatie van hetzelfde punt.

Het volgende script (SCRIPT 1) zal uitgevoerd worden om de 1 seconde (TRIGGER=1):

```
SCRIPT 1  
    TRIGGER = 1  
    .....  
ENDSCRIPT
```

Het volgende script (SCRIPT 2) zal uitgevoerd worden bij iedere verandering van OFF naar ON van V1:

```
SCRIPT 2  
    TRIGGER = v1  
    .....  
ENDSCRIPT
```

Het volgende script (SCRIPT 3) zal uitgevoerd worden bij iedere verandering van ON naar OFF van V1:

```
SCRIPT 3  
    TRIGGER = !v1  
    .....  
ENDSCRIPT
```

Het volgende script (SCRIPT 4) zal uitgevoerd worden bij iedere verandering van OFF naar ON van I2.1:

```
SCRIPT 4  
    TRIGGER = I2.1  
    .....  
ENDSCRIPT
```

3.2.2- VAR, GLOBAL VAR en EXTERN VAR

De scripts laten het gebruik van vele variabelen toe die nodig zijn voor de uitvoering van het programma. De variabelen die gebruikt worden in de scripts moeten expliciet gedeclareerd zijn. De variabelen kunnen werkelijk in twee klassen verdeeld worden.

- **Lokaal:** deze variabelen zullen niet gedeeld worden tussen verschillende scripts. Twee variabelen die dezelfde naam hebben maar gedeclareerd zijn in twee verschillende scripts zullen dus elk apart behandeld worden. Een lokale variabele wordt aangemaakt bij de input van het script en vernietigd bij de output van hetzelfde script.
- **Globaal:** deze variabelen worden gedeeld tussen de scripts wat dus gebruikt kan worden in alle scripts. Een globale variabele, eens aangemaakt, zal altijd bij het einde van een script gehouden worden wat ervoor zorgt dat het script altijd de laatste waarde, dat aan de variabele zelf werd toegekend, zal lezen.

De VAR instructie in een script definieert een lokale variabele en de GLOBAL VAR een globale variabele. Aangezien alle variabelen in een script gedeclareerd moeten worden, informeert de instructie EXTERN VAR een script dat al een variabele gedeclareerd is in een ander script.

```
script 1
  trigger = 2
  var TEMP1
  global var TEMP2
  .....
```

endscript

```
script 2
  trigger = 2
  var TEMP1
  extern var TEMP2
  .....
```

endscript

De lokale variabelen, gebruikt door een subroutine, moeten gedeclareerd worden binnen de subroutine zelf en niet in het script. Met andere woorden, als een lokale variabele gedeclareerd is in een script dan kan dezelfde variabele niet toegankelijk zijn voor een subroutine in hetzelfde script.

3.2.3- Logische en Mathematische operaties

De scripts laten het uitvoeren van algemene logische en mathematische operaties toe. De toegelaten logische en mathematische operators zijn:

&	AND
	OR
^	EXOR
!	NOT
+	Optellen
-	Aftrekken
*	Vermenigvuldigen
/	Delen
=	Gelijk aan

Haakjes zijn niet toegelaten in logische en mathematische operaties en ook niet meer dan één operatie voor elke lijn is toegelaten. Herinner dat het resultaat van de logische en mathematische operaties altijd een 16 bit geheel getal is. Als het resultaat een negatief getal is dan bevindt dit zich in het complement-formaat van de twee.

Het volgende script toont een paar voorbeelden over logische en mathematische operaties.

```
script 1
  trigger = 1
  R0 = R1 + R2
  R0 = R0 + 10
  AO1 = R100 / 2
  R50 = R51 & 0b1111111100000000
endscript
```

Een operatie van het type VAR = VAR [op] K, waar VAR een variabele is, K een constant getal en [op] één van de beschreven logische/mathematische operators (= uitgesloten) is, kan ook de optionele notatie VAR += 10 gebruikt worden. Bijvoorbeeld R0 = R0 + 10 en R0 += 10 zijn absoluut evenwaardige notaties.

3.2.4- IF...THEN...ELSE...ENDIF

De IF...THEN...ENDIF blok laat het uitvoeren van, als de specifieke voorwaarde waar is, instructies binnen de THEN en ENDIF toe. Als de voorwaarde niet waar is, dan zal de uitvoering naar ENDIF springen of ELSE als die gespecificeerd is (ELSE is een optioneel sleutelwoord). Als ELSE gespecificeerd is, dan zal de instructie binnen ELSE en ENDIF uitgevoerd worden.

Iedere IF blok moet gesloten worden door een ENDIF wat verplicht is (tegengestelde met ELSE wat optioneel is).

De voorwaarde van een IF...THEN...ENDIF blok moet gespecificeerd worden via de volgende vergelijingsoperators:

>	Groter dan
>=	Groter dan of gelijk aan
==	Gelijk aan
<	Minder dan
<=	Minder dan of gelijk aan
!=	Niet gelijk aan

Het volgende script bevat twee IF...THEN...ENDIF blokken. Merk op dat het eerste blok geschreven is op dezelfde lijn, vandaar de “;” symbool die gebruikt moet worden om verschillende instructies van elkaar te scheiden. De tweede IF...THEN...ENDIF blok geschreven is met meer lijnen, wat “;” overbodig maakt.

```
SCRIPT 1
  TRIGGER = 1

  IF R0>25 THEN; R0=1; ENDIF
  IF R0==0 THEN
    R1=140
    R2=50
    V1=1
  ENDIF
ENDSCRIPT
```

Het volgende script bevat een IF blok met ELSE.

```
SCRIPT 1
  TRIGGER = 1
  IF I4.7 = 1 THEN
    O1.1 = I1.1
  ELSE
    O1.1 = 0
  ENDIF
ENDSCRIPT
```

Merk op, in beide voorbeelden, hoe het gebruik van tabs het identificeren van begin en einde van de IF blokken vergemakkelijkt. Als het argument van de voorwaarde een bit is dan kan de vergelijingsoperator vermeden worden. Bijvoorbeeld deze twee notaties:

`if R0.1==1 then` en `if R0.1 then`
zijn absoluut evenwaardige statements.

3.2.5- CARRY en ZERO

CARRY en ZERO zijn twee bits systemen (ook flags genoemd) die informatie voorziet over het resultaat van de zojuist uitgevoerde mathematische of logische operatie.

De CARRY flag waarde is 1 als het resultaat van de vorige operatie de waarde 65535 (overflow) overschreed of als het resultaat van de vorige operatie negatief (underflow) is, of als er een deling door 0 was.

De ZERO flag waarde is 1 als het resultaat van de vorige operatie nul is. Het volgende SCRIPT toont het gebruik van deze flags.

```
script 1
    trigger = 2
    R0 = R1 + R2      // som R1 + R2
    if CARRY then
        R0 = 65535    // als resultaat >= 65535 dan R0=65535
    endif

    R3 = R4 - R5      // verschil R4 - R5
    if CARRY then
        R3 = 0        // als resultaat < 65535 dan R0=0
    endif

    R6 = R7 - R8      // verschil R7 - R8
    if ZERO then
        V1 = 1        // als resultaat = 0 dan V1=1
    else
        V1 = 0        // anders v1=0
    endif
endscript
```

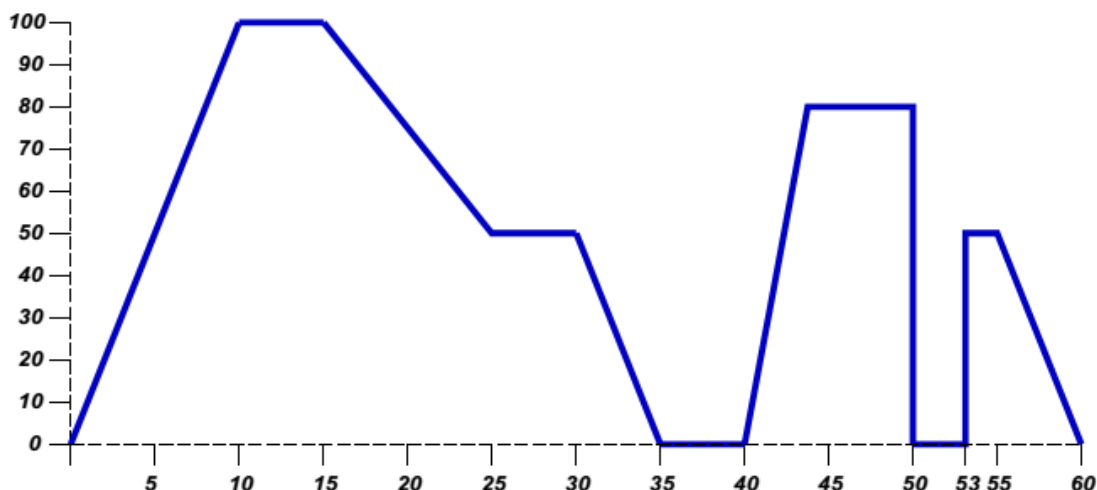
3.2.6- DEFINE

Het volgende script gebruikt het sleutelwoord **define** om een mnemonische naam aan sommige punten toe te kennen, wat het lezen van het programma verbetert.

```
SCRIPT 1
    TRIGGER = 1
    define    Enable      R0.1
    define    Input       I1.1
    define    Lamp        O1.1

    IF Enable = 1 THEN
        Lamp = Input
    ELSE
        Lamp = 0
    ENDIF
ENDSCRIPT
```

Het volgende script toont hoe het mogelijk is om complexe functies te implementeren. Het implementeert dus één kanaal Dynamic Light systeem, wat een lichtspel is verkregen door een dimmer output module (bv. DFDIM). Het dynamisch lichtspel, te zien in de volgende grafiek, moet geïmplementeerd worden (het helderheidspercentage is op de verticale as, horizontaal de tijd in seconden). Na 60 seconden zal de cyclus herhaald worden vanaf het begin.



Het SCRIPT zal één keer per seconde uitgevoerd worden. Allereerst declareert het script een lokale variabele (Step) en twee parameters (twee keer A01, de reden van die dubbele definitie zal later uitgelegd worden). De variabele Step representeert de hoeveelheid seconden die verstreken zijn vanaf het begin van het dynamisch lichtspel.

Aan de output van het script zal de waarde van Step met 1 toenemen. Als het resultaat van die toeneming ≥ 60 is, dan zal Step opnieuw geïnitieerd worden op nul.

Wanneer de Step waarde gelijk is aan één van de momenten wanneer de helderheid veranderd moet worden, dan zullen de twee instructies **RAMP1=K** en **PERC1=Z** uitgevoerd worden. Het gevolg van die instructies (die identiek zijn aan A01=K en A01=Z) is het transfereren van specifieke waarden naar de output A01, in dezelfde volgorde zoals ze geschreven waren. De eerste waarde om naar de output te sturen zal de rampwaarde zijn en het tweede het gevraagde percentage. Het gebruik van twee verschillende definities voor dezelfde output (A01) wordt enkel gebruikt om het script leesbaar te maken.

Voor de wisselwerking tussen de code en de rampwaarde kan de DFDM technische documentatie geraadpleegd worden.

```
SCRIPT 1
  TRIGGER=1
  var Step
  Define RAMP1      A01
  Define PERC1      A01

  IF Step==0 THEN; RAMP1=110; PERC1=100; ENDIF
  IF Step==15 THEN; RAMP1=115; PERC1=50; ENDIF
  IF Step==30 THEN; RAMP1=110; PERC1=0; ENDIF
  IF Step==40 THEN; RAMP1=105; PERC1=80; ENDIF
  IF Step==50 THEN; RAMP1=101; PERC1=0; ENDIF
  IF Step==53 THEN; RAMP1=101; PERC1=50; ENDIF
  IF Step==55 THEN; RAMP1=110; PERC1=0; ENDIF
  Step = Step+1
  IF Step>=60 THEN; Step=0; ENDIF
ENDSCRIPT
```

Natuurlijk kunnen andere dimmeroutputs in hetzelfde script toegevoegd worden om zo een multi-channel dynamisch lichtspel te realiseren.

3.2.7- GOTO

De GOTO instructie veroorzaakt een sprong naar een lijn van hetzelfde script die geïdentificeerd is als label. Het label die gebruikt wordt om de bestemming van de sprong te identificeren moet gevolgd worden door het ":" symbool en geplaatst worden op een lege lijn.

Het label op de GOTO lijn moet niet gevolgd worden door ":" (zie volgend voorbeeld).

```
script 1
    trigger = 1
    if R0 == 1 then
        goto ABC
    endif
    if R0 == 2 then
        goto DEF
    endif
    R10 = 0
    exit
ABC:
    R10 = 101
    exit
DEF:
    R10 = 237
    exit
endscript
```

3.2.8- SUBROUTINES en FUNCTIES

Een subroutine of een functie is een sequentie van instructies die meerdere keren door één of meerdere scripts uitgevoerd kunnen worden. De instructies in een subroutine moeten omringd worden door SUB en ENSUB sleutelwoorden. Alle subroutines van een programma kunnen in één enkele script, optioneel, zitten. In dat geval, met het script die de subroutine bezit, moet het de instructie niet triggeren (TRIGGER) maar wel als hetzelfde script enkel subroutines bezit.

De definitie van een subroutine veroorzaakt automatisch een declaratie van een globale variabele die dezelfde naam van de subroutine heeft, wat gebruikt kan worden om een waarde terug te sturen (typisch het resultaat van de functie). Om een subroutine op te roepen kan de CALL instructie gebruikt worden of kan de functie opgeroepen worden in een directe modus. De volgende regels zijn altijd waar:

- Gebruik CALL als de subroutine, na het oproepen, geen enkele waarde terugstuurt
- Roep direct de functie op als het, na het oproepen, een waarde moet terugsturen

Bij iedere oproep van een subroutine of functie kunnen één of meerdere paramaters doorgegeven worden als inputs (beide variabelen en constanten), die gespecificeerd worden binnen ronde haakjes. De variabele parameters kunnen doorgegeven worden als referentie of als waarde. Het verschil tussen deze twee zaken is het volgende:

- Parameters doorgegeven als **referentie**: het Word adres van de parameter (input, output, register, enz.) zullen gekopieerd worden naar de verwante parameter van de subroutine en zal gebruikt worden als pointer binnen de subroutine zelf. Op die manier kan, de parameter die doorgegeven werd naar de subroutine, beide gelezen en aangepast worden door de subroutine.
- Parameters doorgegeven als **waarde**: de waarde van de parameter (input, output, register, constante, enz.) zal gekopieerd worden naar de verwante parameter van de subroutine en zal gebruikt worden als variabele binnen de subroutine zelf. Op die manier kan, de parameter die doorgegeven werd naar de subroutine, beide gelezen maar niet aangepast worden door de subroutine. Een aanpassing aan die parameter binnen de subroutine zal de waarde van de lokale variabele, aangemaakt om de parameter te ontvangen, veranderen maar het zal de doorgegeven parameter bij het oproepen niet veranderen.

De syntax gebruikt om te specificeren welke methode toegepast moet worden op ieder doorgegeven parameter is het volgende:

```
SUB NAMESUB( PAR1, PAR2, [PAR3], [PAR4])
```

Waar:

- **PAR1** en **PAR2** zijn parameters doorgegeven als waarde
- **PAR3** en **PAR4** zijn parameters doorgegeven als referentie, omringd door vierkante haakjes

Om te specificeren dat een parameter geïnterpreteerd moet worden als een referentie is het dus noodzakelijk en genoeg om gewoon de gerelateerde parameter te omringen met vierkante haakjes in de lijn die de subroutine definieert (en enkel in die lijn).

Opmerking: parameters bestaande uit bit type (bv. V1, O3.2, I4.3, enz.) kunnen niet doorgegeven worden als referentie. Deze parameters kunnen enkel als waarde doorgegeven worden.

De volgende twee voorbeelden tonen elk een oproep naar een subroutine met parameters aan:

<pre>call SETUP(R0, AO1, 128) sub SETUP(REG, [OUT], K) endsub</pre>	het oproepen van de subroutine waar drie parameters worden doorgegeven. Er wordt geen waarde teruggestuurd door de subroutine. De parameter AO1 zal doorgegeven worden als referentie wat ervoor zorgt dat de subroutine zelf de waarde van de parameter kan veranderen. tegenstellend tot R0 zal de waarde doorgegeven worden wat de subroutine de originele waarde binnen de parameter niet zelf kan veranderen. Laatste parameter is een numerieke, constante waarde.
<pre>R100 = CALCULATE(R10, R11) sub CALCULATE(REG1, [REG2]) endsub</pre>	een functie waar twee parameters worden doorgegeven en dat een waarde zal terugsturen, gekopieerd in R100. Parameter R11 zal doorgegeven worden als referentie wat ervoor zorgt dat de functie zelf de waarde van de parameter kan veranderen. tegenstellend tot R10 zal de waarde doorgegeven worden wat de subroutine de originele waarden binnen de parameter niet zelf kan veranderen.

Opmerking:

- Een subroutine of een functie, wanneer ze worden opgeroepen door een script anders dan het script waar de functie was inbegrepen, moet geplaatst worden voor de oproep zelf.
- Als een subroutine lokale variabelen gebruikt dan moeten die binnen de subroutine zelf gedeclareerd worden
- Een subroutine kan een andere subroutine oproepen met een maximum van 16 oproepen.

Voor de subroutines en functies zonder parameters moeten de volgende punten in rekening gehouden worden:

1. Als een subroutine of een functie geen nood heeft aan inputparameters, moet het toch gedeclareerd worden met "()". Bijvoorbeeld: **sub TEMPERATURE()**
2. De oproepen naar subroutines en functies zonder parameters kunnen met of zonder haakjes geschreven worden. Ter voorbeeld zijn de volgende oproepen perfect gelijk aan elkaar:

```
R0 = TEMPERATURE()
R0 = TEMPERATURE
call TEMPERATURE()
call TEMPERATURE
```

Voorbeeld:

Het volgende script zet de vier analoge waarden gelezen van vier DFTA modules (die normaal gezien uitgedrukt staan in Kelvin maal 10) om naar °C. Het resultaat van die omzetting wordt geschreven naar registers van R0 tot R3. Er zal gebruik gemaakt worden van een functie omdat er een wiskundige operatie moet uitgevoerd worden die herhaald moet worden voor elke DFTA. Het algemene script geeft de waarde van de DFTA door naar de functie. Het resultaat zal teruggestuurd worden naar de variabele CONVERT. Merk op dat de instructie EXIT het script sluit (het is zoals een GOTO naar een ENDSRIPT instructie).

```
script 1
  trigger = 2
  define DFTA1      AI1
  define DFTA2      AI2
  define DFTA3      AI3
  define DFTA4      AI4

  R0 = CONVERT(DFTA1)
  R1 = CONVERT(DFTA2)
  R2 = CONVERT(DFTA3)
  R3 = CONVERT(DFTA4)
  exit

  sub CONVERT(TEMPER)
    CONVERT = TEMPER - 2730
    CONVERT = CONVERT / 10
  endsub
endscript
```

Hetzelfde resultaat kan ook bereikt worden via het volgende script, waar de bestemmingsregisters zullen doorgegeven worden als referentie wat ervoor zorgt dat de subroutine direct met hen kan werken. Verkies eerder voor het vorige voorbeeld omdat het efficiënter is.

```
script 1
  trigger = 2
  define DFTA1      AI1
  define DFTA2      AI2
  define DFTA3      AI3
  define DFTA4      AI4

  call CONVERT(R0, DFTA1)
  call CONVERT(R1, DFTA2)
  call CONVERT(R2, DFTA3)
  call CONVERT(R3, DFTA4)
  exit

  sub CONVERT([REG], TEMPER)
    TEMPER = TEMPER - 2730
    REG = TEMPER / 10
  endsub
endscript
```

Voorbeeld:

Het volgende script toont hoe de RET instructie het verlaten van de subroutine toelaat (het is zoals een GOTO naar de ENDSUB instructie). Het script zet de analoge waarde, gelezen van een DFTA module, om naar °C en plaatst het resultaat in register R1. Ter toevoeging, output O1.1 gaat ON als het resultaat binnen het bereik van 18 tot 23 graden is, anders gaat de output OFF.

```
script 1
  trigger = 5
  define DFTA1      AI100

  R1 = CONVERT(DFTA1)
  exit

  sub CONVERT(TEMPER)
    CONVERT = TEMPER - 2730
    CONVERT = CONVERT / 10
    if CONVERT >= 23 then
      O1.1 = 0
      ret
    endif
    if CONVERT <= 18 then
      O1.1 = 0
      ret
    endif
    O1.1 = 1
  endsub
endscript
```

3.2.9- BIT(x)

De parameter doorgestuurd naar een subroutine of de functie en de optionele teruggestuurde waarde zijn, als standaard, integer 16 bit nummers. Als een bit doorgegeven moet worden naar een functie of de teruggestuurde parameter moet een bit zijn dan moet dat expliciet gedeclareerd zijn door BIT(x).

BIT(x) declareert dat parameter x van een subroutine, functie of een teruggestuurde waarde een bit is. De declaratie BIT(x) moet alleen in subroutines of functies gebruikt worden.

De declaratie BIT(x) mag ENKEL geplaatst worden in de subroutine declaratie.

Het volgende script gebruikt een functie met als inputparameters een waarde (REG) en een bit (ENABLE), dat gespecificeerd werd door de declaratie BIT(ENABLE). De functie stuurt ook een waarde terug (RSET).

```
script 1
  TRIGGER = 5
  var RTEMP

  R82 = RSET(R50, V1)
  R83 = RSET(R51, V2)
  R84 = RSET(R52, V3)
  R85 = RSET(R53, V4)

  exit

  sub RSET( REG, BIT(ENABLE) )
    if ENABLE == 1 then
      RSET = REG / 2
      RSET = RSET + 128
    else
      RSET = 0
    endif
  endsub
endscript
```

Het volgende script gebruikt een functie met als input parameters twee waardes (REG1 en REG2). De functie keert een bit (TEST) terug dat gespecificeerd werd door de declaratie BIT(TEST) (REG1,REG2).

```
script 2
    TRIGGER = 5
    var RTEMP

    RTEMP.1 = TEST(R0, R1)
    if RTEMP.1 == 1 then
        R20 = 100
    else
        R20 = 0
    endif
    RTEMP.1 = TEST(R2, R3)
    if RTEMP.1 == 1 then
        R21 = 200
    else
        R21 = 0
    endif
    exit
    sub BIT(TEST) (REG1, REG2)
        REG1 = REG1 / 2
        REG2 = REG2 / 4
        if REG1 > REG2 then
            TEST = 1
        else
            TEST = 0
        endif
    endsub
endscript
```

Het volgende script is een script uit de combinatie van de vorige twee voorbeelden. Deze script gebruikt als inputparameters een waarde (REG) en een bit (ENABLE), gedeclareerd door BIT(ENABLE). De functie stuurt een bit (TEST) terug, gedeclareerd door BIT(TEST) (REG1, BIT(ENABLE)).

```
script 3
    TRIGGER = 5

    V17 = TEST(R50, V1)
    V18 = TEST(R51, V2)
    V19 = TEST(R52, V3)
    V20 = TEST(R53, V4)
    exit

    sub BIT(TEST) ( REG, BIT(ENABLE) )
        if ENABLE == 1 then
            REG = REG / 2
            if REG > 100 then
                TEST = 1
            else
                TEST = 0
            endif
        else
            TEST = 0
        endif
    endsub
endscript
```

3.2.10- WORD(x) en pointers

De WORD(x) functie stuurt het nummer (adres) van de Word binnen parameter x terug, waar parameter x een input, output, virtueel punt, register of een counter hoort te zijn zoals in de volgende voorbeelden:

```
A1 = WORD(I18)           // stuurt het nummer van de Word bevattende I18
A2 = WORD(I18.1)         // stuurt het nummer van de Word bevattende I18.1
A3 = WORD(O93)           // stuurt het nummer van de Word bevattende O93
A4 = WORD(V46)           // stuurt het nummer van de Word bevattende V46
A5 = WORD(R37)           // stuurt het nummer van de Word bevattende R37
A6 = WORD(C42)           // stuurt het nummer van de Word bevattende C42
```

Het volgende script toont ons hoe de WORD(x) functie en pointers gebruikt worden. Stel dat de applicatie een script nodig heeft dat, om de twee seconden, telt hoeveel registers, in het bereik R0 tot R10, een waarde bevatten anders dan nul. De resultaten (de hoeveelheid registers !=0) moeten geplaatst worden in register R15.

De functie WORD(R0) stuurt het nummer van de Word terug waar register R0 zit. Het script definieert een variabele (in dit voorbeeld is zijn naam ptr) dat vanaf het begin gelijk is aan het Word nummer van register R0. De notatie [ptr] (binnen vierkante haakjes) stuurt de inhoud van de “aangeduide” register terug. In het volgende script zal de waarde van R15 verhoogt worden met 1 iedere keer wanneer de inhoud van elke geadresseerde register, in de loop, anders is dan nul. De notatie ptr += 1 is gelijkwaardig aan ptr = ptr + 1, zoals R15 = R15 + 1 dat geschreven kan worden als R15 += 1. Wanneer de pointer groter wordt dan het adres van R10, dan zal de loop onderbroken worden en wordt het script beëindigd.

```
script 1
    trigger = 2
    var ptr
    ptr = WORD(R0)
    R15 = 0
LOOP:
    if ptr <= WORD(R10) then
        if [ptr] <> 0 then
            R15 = R15 + 1
        endif
        ptr += 1
        goto LOOP
    endif
endscript
```

Ander voorbeeld: de dag van de maand zit in Word 1924 (zie RAM map). Om die waarde te kopiëren (en dus de inhoud van Word 1924) naar register R2 kan de volgende instructie geschreven worden:

```
ptr = 1924
R2 = [ptr]
```

Langs de andere kant is het ook mogelijk de inhoud van R2 te kopiëren naar Word 1924 zoals het volgende:

```
ptr = 1924
[ptr] = R2
```

De pointers zijn nuttig wanneer de Word die nodig zijn voor het lezen en schrijven, niet geïdentificeerd kunnen worden op andere manieren. Met andere woorden, wanneer het niet geïdentificeerd kan worden door notaties zoals Cx, Ry, enz.).

3.2.11- @RAM k en @WORD k

De functies @RAM k en @WORD k verlenen toegang tot paren van RAM locaties of enkele Words. De gespecificeerde waarde (k) is het beginadres van RAM of het Word nummer en **moet een constante waarde zijn** binnen het bereik van 0 tot 65535, in het eerste geval, en 0 tot 32767 in het tweede geval.

Als voorbeeld, de dag van de maand staat in het RAM-geheugen op het adres 0x0F08-0x0F09, corresponderend aan Word 1924. Om die waarde te kopiëren in bijvoorbeeld register R2 kunnen de volgende notaties gebruikt worden:

R2 = @RAM0x0F08 of **R2 = @WORD1924**.

Zo kan de inhoud van Word ook nog beschreven worden: **@RAM0x0F08 = R2** of **@WORD1924 = R2**.

Deze functies zijn nuttig wanneer de Word die nodig is voor het lezen en schrijven, niet geïdentificeerd kunnen worden op andere manieren en zo een optie zijn voor de pointer methode zoals eerder beschreven.

3.2.12- SWAP(x)

De SWAP(x) functie wisselt de hoge byte met de lage byte van de gespecificeerde Word (x). De Word kan gespecificeerd worden op de volgende manieren:

1. Direct door zijn symbolische naam (bv. R34, C48, AI24, enz.)
2. Direct door @WORD of @RAM
3. Door een pointer

Voorbeelden van de eerste manier:

R0 = SWAP (I18)

R1 = SWAP (R1)

Voorbeeld van de tweede manier:

R66 = SWAP (@WORD1924)

Voorbeelden van de derde manier:

ptr = 1924

R45 = SWAP ([ptr])

3.2.13- RANDOM(0)

De RANDOM(0) functie stuurt een willigkeurige nummer terug. Het nummer wordt gegenereerd volgens een bijzondere algoritme (Lehmer Random Number Generator) die een gelijkmatig verdeelde, pseudo willigkeurige waarde terugstuurt.

Het volgende script roept de RANDOM(0) functie op om de 60 seconden en de terugstuurde willigkeurige waarde zal naar R0 gekopieerd worden.

```
script 1
    trigger = 60
    R0 = RANDOM(0)
endscript
```

3.2.14- BMASK(x)

De BMASK(x) functie stuurt een 16 bit nummer terug die, in zijn binaire formaat, enkel één bit heeft die ingesteld staat op 1 bij de positie van $(x-1)\%16$. Die notatie betekent $(x-1)$ module 16 en het is equivalent met de rest van de deling van $(x-1)$ door 16. De BMASK(x) functie is daarom een dekmantel die nuttig kan zijn voor bitoperaties.

Het script in het volgend voorbeeld roept 4 keer een subroutine op die een virtueel punt moet instellen of resetten als de waarde van een register groter of kleiner is dan een constante waarde. Aangezien het virtueel punt, de register en de constante waarde veranderen bij iedere oproep dan moeten die parameters doorgestuurd worden naar de subroutine. Aangezien het virtueel punt geschreven moet worden dan zou die parameter doorgestuurd moeten worden als referentie maar dat is niet toegelaten omdat het een bit is (zie paragraaf SUBROUTINES en FUNCTIES).

Dit is een typisch geval waar de BMASK(x) functie voor gevraagd wordt. De oproep stuurt het adres van de Word met het virtueel punt (WORD(Vn)) door naar de subroutine en de dekmantel laat het identificeren, in de Word, van de positie van de bit toe verwant aan dat virtueel punt (BMASK(n)).

Om het virtueel punt in te stellen voert de subroutine de OR tussen de Word met het punt en de dekmantel uit (dat, zoals eerder gezegd, alleen één bit bezit ingesteld op 1 op de positie van de bit verwant met het gewenste punt).

Om het virtueel punt te resetten voert de subroutine de AND tussen de Word met het punt en het complementaire van de dekmantel uit (wat enkel een 0 zal bevatten op die positie van de bit verwant met het gewenste punt).

```
script 1
    trigger = 1
    call TEST(R0, 50, WORD(V49), BMASK(49))
    call TEST(R1, 100, WORD(V50), BMASK(50))
    call TEST(R2, 150, WORD(V51), BMASK(51))
    call TEST(R3, 200, WORD(V52), BMASK(52))
    exit

    sub TEST(REGIN, KAPPA, [WVIRT], MSK)
        if REGIN > KAPPA then
            WVIRT = WVIRT | MSK    // stelt virtueel punt in
        else
            WVIRT = WVIRT & !MSK    // reset virtueel punt
        endif
    endsub

endscript
```

De BMASK(x) functie kan ook toegepast worden op andere bit parameters. Het volgende voorbeeld is gelijkaardig met het vorige maar hier switcht de subroutine echte inputs ON en OFF i.p.v. virtuele punten.

```
script 1
    trigger = 1
    call TEST(R0, 50, WORD(O1.5), BMASK(5))
    call TEST(R1, 100, WORD(O1.6), BMASK(6))
    call TEST(R2, 150, WORD(O1.7), BMASK(7))
    call TEST(R3, 200, WORD(O1.8), BMASK(8))
    exit

    sub TEST(REGIN, KAPPA, [WOUT], MSK)
        if REGIN > KAPPA then
            WOUT = WOUT | MSK    // output ON
        else
            WOUT = WOUT & !MSK    // output OFF
        endif
    endsub

endscript
```


4- PROGRAMMA'S SCHRIJVEN

Een programma schrijven is de eerste stap bij het installeren van de DFCP 4 controller. De vergelijkingen en alles wat erbij hoort moeten geschreven worden volgens de gerelateerde syntax wat beschreven staat in de vorige paragrafen.

Om een programma te schrijven voor DFCP 4 moet het softwarepakket DCP IDE (Integrated Design Environment) gebruikt worden. Dit pakket wordt gratis voorzien door DUEMMEGI samen met de DFCP 4 module. Dat programma moet geïnstalleerd worden op een PC met de volgende minimale vereisten:

- WINDOWS® XP, 7 of hoger
- Processor met 1000MHz clock minimum
- 512M RAM-geheugen
- HD met 50MB vrije ruimte
- Video met grafische resolutie 1024x768 pixel
- Muis

DCP IDE kent ook alle operaties gerelateerd met het instellen en het onderhoud. Voor meer details over het gebruik van dit programma, raadpleeg het toepasselijk hoofdstuk van deze handleiding.

De DCP IDE software beschikt over:

- Een teksteditor om zowel het programma voor de DFCP 4 te schrijven als het programma die in outputmodules geladen moet worden (SCRIPT, configuratie, MACRO, enz.).
- Een compiler die het vertalen van een ASCII file naar een binaire file toelaat die geschikt is om getransfereerd te worden naar een niet-vluchtig geheugen (type FLASH) van de DFCP 4.
- Een deel om het programma van de PC naar DFCP 4 en veldmodules te transfereren (en vice versa).
- DCP VISIO, een grafisch hulpprogramma om de status van de installatie weer te geven (input en output modules, counters, virtuele punten, registers, enz.).
- Een simulator om het deel van het programma verwant aan DFCP 4 te testen voordat het in zijn geheugen wordt getransfereerd.

De file die het programma (DFCP 4 en veld modules) bevat, staat in ASCII formaat en moet dus de extensie **.EQU** (of **.EXT**) hebben. Bijvoorbeeld:

filename.EQU

waar *filename* de gewenste naam is voor de file.

De extensie .EQU of .EXT is **verplicht** omdat de opeenvolgende stappen (compileren en programmeren) die extensie nodig hebben.

Het programmeren van DFCP 4 vindt plaats op 3 sequentiële stappen, allemaal ondersteund door het DCP IDE pakket.

1. Aanmaak (of aanpassen) van *filename.EQU* file, bevattende het bedieningsprogramma in leesbaar formaat (ASCII)
2. Compileren van *filename.EQU*, wat de conversie is van de ASCII file naar een *filename.BIN* wat gepast is om naar het DFCP 4 geheugen te transfereren.
3. Het uploaden van *filename.BIN* naar het DFCP 4 geheugen.

Als er sommige syntax errors worden ontdekt tijdens de tweede stap, dan zullen die gerapporteerd worden door de compiler die informatie geeft over het type error en het lijnnummer waar het gebeurde.

4.1- Regels voor programmeren

Het programma moet geschreven worden volgens de syntax zoals beschreven in vorige paragrafen (logic, counter, timer, SCRIPT, enz.). Het is niet noodzakelijk om de DFCP 4 te verbinden met de PC tijdens het schrijven of compileren van een programma.

De volgende regels moeten nageleefd worden:

- Het gebruik van spaties en tabs hebben geen betekenis. Ze zullen genegeerd worden door de compiler maar het gebruik van spaties en tabs vergemakkelijken de leesbaarheid van het programma.
- Een vergelijking kan verdeeld worden over verschillende lijnen door het symbool \ (backslash) te gebruiken bij het einde van een lijn om aan te tonen dat de vergelijkingen verdergaat op de volgende lijn.
- Een vergelijking eindigt op het einde van een lijn (als \ niet gebruikt wordt).
- Het symbool // declareert dat de volgende woorden, tot het einde van de lijn, opmerkingen zijn wat dus genegeerd zal worden door de compiler. Die opmerkingen zijn heel nuttig voor de leesbaarheid en documentatie van het programma. Het wordt sterk aanbevolen om iedere vergelijkingen in het programma te beschrijven.
- Zowel hoofdletters als kleine letters kunnen gebruikt worden tijdens het schrijven van vergelijkingen.

In plaats van input- en outputsymbolen (**Ij.k**, **Ox.y**, **Vn**, **Aj**), is het mogelijk om sommige namen van variabelen, gedefinieerd door de programmeur, te gebruiken a.d.h.v. de **define** zoals hieronder beschreven:

```
define      Pump1      O1.1 // Definitie output
define      Command    I1.1 // Definitie input
Pump1 = Command          // Vergelijking
```

De vorige vergelijking is gelijkwaardig met:

```
O1.1 = I1.1
```

maar het kan gemakkelijker geïnterpreteerd worden. De namen van variabelen, gedefinieerd door de **define**, kunnen geen spaties bevatten. De compiler zal hoofd- of kleine letters negeren.

Het volgende voorbeeld toont een mogelijk en simpel programma aan die gebruik maakt van de **define**:

```
//////////
// Definities //////////
//////////
define      StairLight  O1.1
define      Floor1Button I1.1
define      Floor2Button I1.2
define      Floor3Button I1.3

// Definieert een virtueel punt als OR van elke knop (parallele verbinding)
V1 = Floor1Button | Floor2Button | Floor3Button

// Lichtoutput
StairLight = TIMER (V1, 0, 450)
```

In het bovenste voorbeeld zijn er 3 knoppen, één per iedere verdieping van een gebouw. Het indrukken van een knop schakelt de trapverlichting aan. Dat licht zal blijven branden gedurende een 45 seconden nadat de knop werd losgelaten, wat dan automatisch zal uitschakelen met dank aan de TIMER functie. Hetzelfde programma kan ook geschreven worden zonder het gebruik van de namen van de variabelen, zoals het volgende:

```
// Commando via de knoppen
V1 = I1.1 | I1.2 | I1.3

// Lichtoutput
O1.1 = TIMER (V1, 0, 450)
```

Merk op dat het gebruiken van **define**, het programma op zijn best en mnemonisch leesbaar is. Over het gebruik van **define** in het SCRIPT, raadpleeg het gerelateerde hoofdstuk.

4.2- Het compileren van het programma

Houd goed in gedachten dat in een Domino systeem met DFCP 4 er altijd twee programma's zijn, één voor de controller en één voor de veldmodule. Om preciezer te zijn, het programma voor de veldmodules zijn gemaakt door een set van kleine programma's die getransfereerd moeten worden naar ieder gerelateerde outputmodule (geïdentificeerd door het adres). Het is hoe dan ook mogelijk dat de gevulde modules geen programma bezitten en zal in dat geval alle nodige functies uitgevoerd worden door de DFCP 4.

Het compileren van een programma bestaat uit de vertaling van een "leesbare" file met .EQU (of .EXT) als extensie zoals, getoond in de editor van DCP IDE, naar een file met .BIN extensie dat geschikt is om opgeslagen te worden in het geheugen van DFCP 4 of het geheugen van Domino output modules.

Het compileren is de tweede stap van een Domino systeem. De file die het programma (.EQU of .EXT) bezit moet gecompileerd worden via het geschikte menu onderdeel van DCP IDE.

De compiler verwerkt de geschreven vergelijkingen, controleert de syntax en congruentie, waarschuwt voor errors en linkt de data in een binaire file wat dezelfde naam heeft als de .EQU file maar met .BIN extensie. De binaire file kan niet afgedrukt worden maar is wel geschikt om het te transfereren naar het DFCP 4 geheugen.

Het is niet noodzakelijk om de DFCP 4 te verbinden met de PC tijdens het schrijven of compileren van een programma.

Als er tijdens het compileren één of meerdere errors plaatsvinden, dan zullen die weergegeven worden op het scherm van uw PC op een apart venster (één voor deel DFCP 4 en één voor deel veldmodules).

De compiler kan ook altijd WARNINGS rapporteren. Dat betekent dat er geen errors werden ontdekt maar dat er sommige punten geverifieerd moeten worden om het programma naar het DFCP 4 geheugen te kunnen uploaden.

4.3- Programma uploaden naar DFCP 4 geheugen

Het programmeren van een DFCP 4 controller bestaat uit het uploaden van een binaire file naar zijn FLASH geheugen, wat de systeemconfiguratie en code bevat. Dat is de derde en laatste stap na het schrijven en compileren van vergelijkingen.

Hetzelfde concept is waar voor het programmagedeelte dat in de veldmodules moet zitten.

Het uploaden gebeurt via het geschikte menu onderdeel van DCP IDE door de RS232 poort van uw PC verbonden met de DFCP 4 seriële poort. Het programma kan hoe dan ook getransfereerd worden door de RS485 poort of door het LAN netwerk (bijvoorbeeld het gebruiken van de WEBS module in brugmodus).

De DFCP 4 controller moet gevoed worden om het programma up te loaden. Als er ook veldmodules worden geprogrammeerd dan moet de Domino bus gevoed worden waaraan de gevulde modules geconnecteerd moeten zijn.

Een nieuw programma die getransfereerd is naar DFCP 4 wordt bewaard op een verschillende geheugenlocatie wat ervoor zorgt dat tijdens de download, het vorig ingeladen programma werkt zonder onderbrekingen. Alleen wanneer het downloaden van het nieuwe programma voltooid is, en alles goed gaat, dan wordt er een automatische switch uitgevoerd tussen het oude en het nieuwe programma.

5- HET INSTELLEN

5.1- Verbindingen

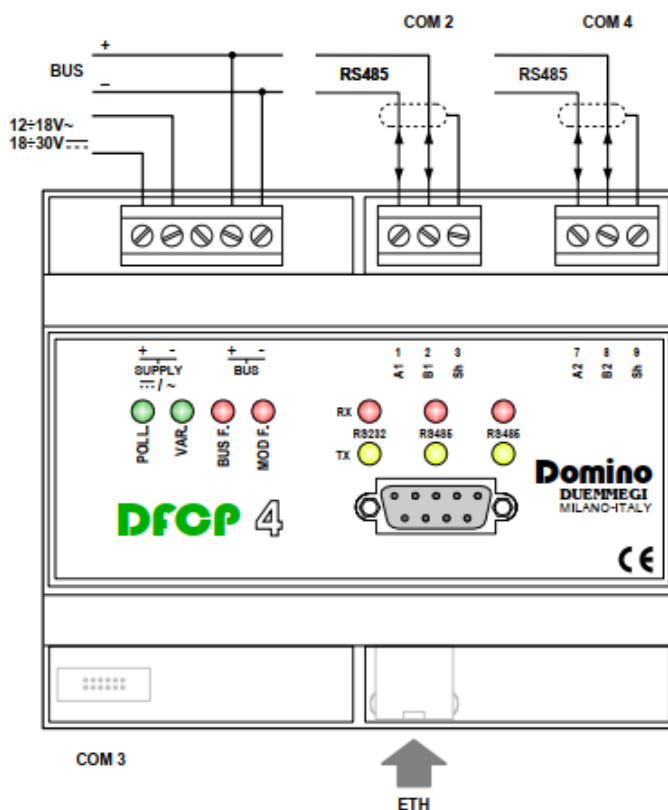
DFCP 4 is verkrijgbaar in een DIN modulaire behuizing (grootte van 6 modules) en het voorziet een verwijderbare 5 polen aansluitblok voor de connectie naar de bus en de voeding.

DFCP 4 bezit een seriële RS232 poort (COM1), twee RS485 poorten (COM2 en COM4), een toegewijde poort voor speciale modules (COM3) en een Ethernet poort (optioneel).

De volgende afbeelding toont de geschikte verbindingen die gemaakt moeten worden en de beschrijving van de aansluitingen. Aansluiting A3 is niet inwendig verbonden.

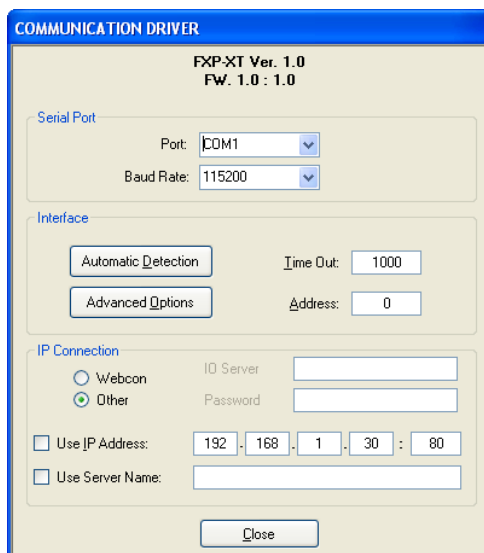
DFCP 4 kan gevoed worden door een wisselspanning in het bereik van 12 tot 18V, of door een gelijkspanning in het bereik van 18V tot 30V. Wanneer een gelijkspanningsvoeding wordt gebruikt, connecteer het dan volgens de getoonde polariteit.

Verbindingen van DFCP 4 controller



5.2- Baudrate selecteren

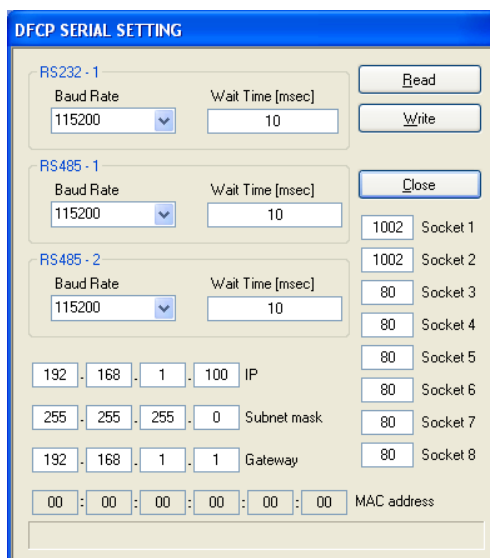
De fabrieksinstelling van Baudrate voor DFCEP 4, zowel voor RS323 en RS485 poorten, staat vast op 115200 Baud. Als voor enige reden die snelheid veranderd moet worden, dan moet er gebruik gemaakt worden van DCEP IDE. Verbindt de DFCEP 4 met de PC en start DCEP IDE op. Selecteer vanuit het menu "Communication", "Enable Communication". Het volgende venster zal verschijnen:



The dialog box is titled "COMMUNICATION DRIVER" and contains the following sections:

- Serial Port:** Port: COM1, Baud Rate: 115200.
- Interface:** Buttons for "Automatic Detection" and "Advanced Options". Time Out: 1000, Address: 0.
- IP Connection:** Radio buttons for "Webcon" and "Other". Fields for "IO Server" and "Password". Checkboxes for "Use IP Address" (with fields 192, 168, 1, 30, 80) and "Use Server Name".
- A "Close" button at the bottom.

Selecteer de poort (bv. COM1) of druk op Automatic Detection om het automatisch zoeken naar een DFCEP 4 uit te voeren. Druk dan op "Advanced Options" waarna het volgende venster verschijnt:



The dialog box is titled "DFCEP SERIAL SETTING" and contains the following sections:

- RS232 - 1:** Baud Rate: 115200, Wait Time [msec]: 10. Buttons: Read, Write.
- RS485 - 1:** Baud Rate: 115200, Wait Time [msec]: 10. Button: Close.
- RS485 - 2:** Baud Rate: 115200, Wait Time [msec]: 10.
- IP:** Fields for IP (192, 168, 1, 100), Subnet mask (255, 255, 255, 0), and Gateway (192, 168, 1, 1).
- MAC address:** Fields for MAC address (00, 00, 00, 00, 00, 00).
- Sockets:** A list of 8 sockets with values 1002, 1002, 80, 80, 80, 80, 80, 80.

Druk Read om de huidige instellingen voor de Baurate op RS232 en de twee poorten RS485 van de DFCEP 4 te lezen. De andere twee parameters (Wait Time) zijn de vertragingen voordat het antwoord van de DFCEP 4 naar een Host aanvraag gaat. Voor die parameters wordt er gesuggereerd om die niet te veranderen, als het echt niet nodig is. Kies de gevraagde Baudrate voor iedere poort en druk dan op Write om de nieuwe instelling naar de DFCEP 4 te transfereren. Druk dan op Close. Houd er rekening mee dat wanneer de Baudrate wordt aangepast van een poort die verbonden is met de PC, dat er een nieuwe aanzet tot communicatie moet gebeuren naar de nieuwe Baudrate. De toegelaten Baudrates zijn: 2400, 4800, 9600, 19200, 38400, 57600, 115200.

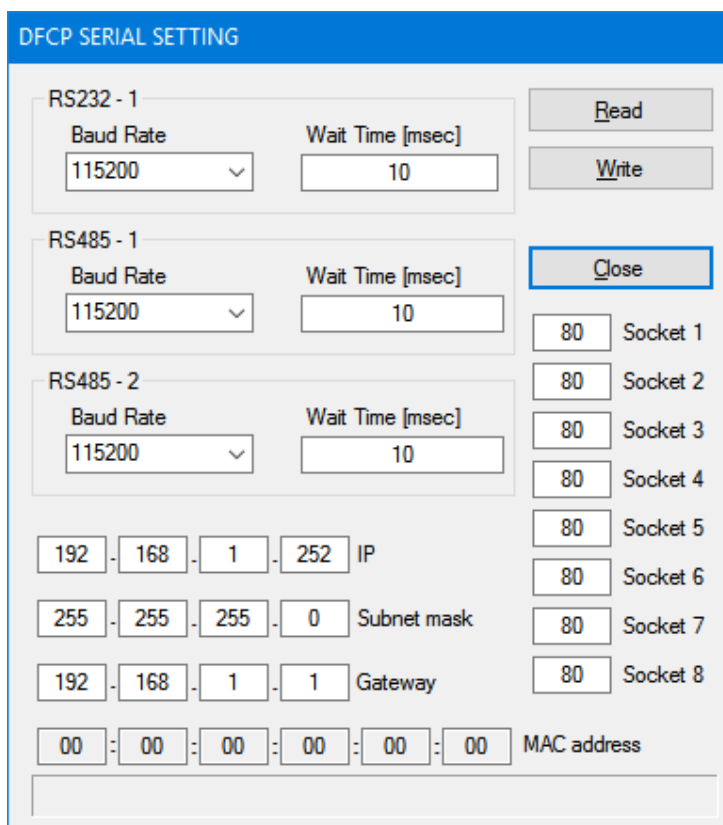
5.3- TCP/IP parameterinstellingen (Ethernet)

Het configureren van TCP/IP parameters is mogelijk op verschillende manieren, twee worden hieronder beschreven.

1- Via RS232 seriële verbinding tussen de PC en DFCP 4/ETH

Dit is de meest verkozen methode omdat het niet van de configuratie van het netwerk afhangt waar DFCP 4/ETH geïnstalleerd moet zijn.

Verbindt de PC aan de RS232 poort van DFCP 4/ETH (zo nodig via een USB-RS232 converter). Open het configuratiepaneel door het te selecteren uit het DFCP IDE menu, "Communication", "Enable Communication" en open de communicatie met de DFCP 4/ETH. Selecteer dan "Advanced Options" waarop het volgende scherm verschijnt:



De vertoonde waardes staan er standaard. Voer de gewenste parameters (IP-adres, Subnet mask, Gateway en poorten voor de verschillende stopcontacten) in en druk dan op Write. Vanaf dat moment zal DFCP 4/ETH zichtbaar zijn op het netwerk met de ingestelde parameters en voorzien zijn dat ze met de netwerkconfiguratie congruent zijn. "Read" laat het rapporteren van DFCP 4/ETH huidige configuratie en zijn MAC-adres toe.

2- Via LAN en browser

DFCP 4/ETH hebben als fabrieksinstellingen, zoals de afbeelding hierboven, het IP-adres 192.168.1.252, 80 poorten voor alle stopcontacten, Subnet mask 255.255.255.0 en Gateway 192.168.1.1.

Als het netwerk al congruent is met deze instellingen van de parameters dan is het mogelijk om de DFCP 4/ETH direct te connecteren met het lokaal netwerk (zorg ervoor dat het standaard IP-adres al niet gebruikt wordt door een ander toestel) en verander die instellingen zoals gewenst via de "Domino Web Server" binnen de DFCP 4/ETH.

De "Domino Web Server" van DFCP 4/ETH kan geopend worden via het gebruik van iedere browser. Voer bij het huidige IP-adres van DFCP 4/ETH (standaard is 192.168.1.252), in de adresbalk van de browser, /webmenu.htm. Bijvoorbeeld:

192.168.1.252/webmenu.htm

De hoofdpagina van "Domino Web Server" zal dan tevoorschijn komen. Selecteer "LAN Configuration" om toegang te krijgen tot de subpagina van de TCP/IP parameterinstellingen.

5.4- RS232 en RS485, seriële poorten van DFCP 4

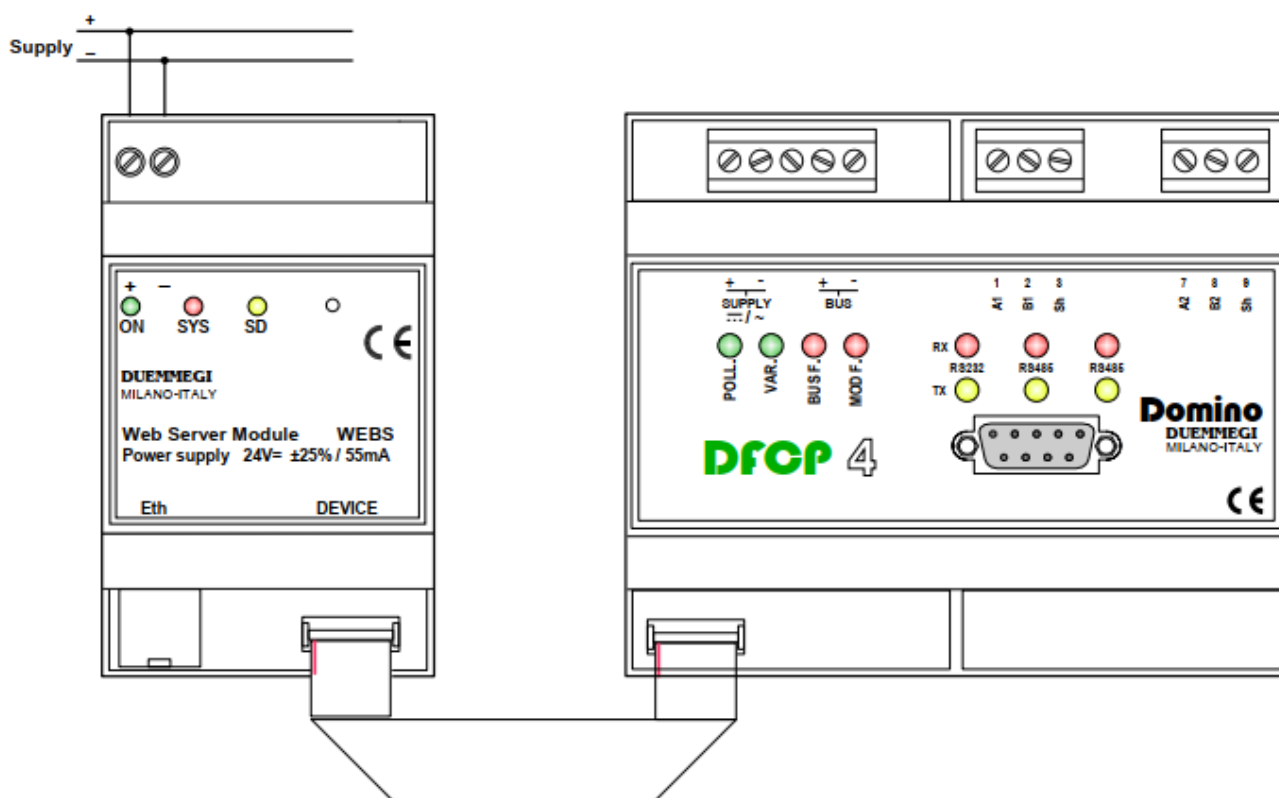
DFCP 4 voorziet zowel RS232 (COM1, op het voorpaneel) als RS485 seriële poorten (COM2 en COM4, aansluitingen 1-2-3 en 7-8-9). Deze poorten worden elektrisch geïsoleerd van de bus door middel van sommige interne opto-couplers en een dc/dc converter (geen extra externe voeding is nodig).

Hoe dan ook, de RS232 en RS485 poorten zijn niet geïsoleerd van elkaar.

WAARSCHUWING: zoals voor alle RS485 netwerken moeten radiale verbindingen vermeden worden. Ook moet de RS485 lijn geladen worden door een 120 Ohm 1/2W weerstand, zowel bij het begin als het einde, tussen de aansluitingen A en B. Het maximum aantal toestellen die op de RS485 lijn verbonden kunnen worden is gelimiteerd tot 32.

5.5- WEBS communicatiepoort

DFCP 4 bezit een speciale communicatiepoort (COM3, onder de bekleding aan de beneden linkerkant, zie volgende figuur) voor de connectie aan de WEBS module, wat zowel Web Server functies als de brug functie voorziet om de DFCP 4 te koppelen aan een LAN netwerk (Ethernet). Voor meerdere details, raadpleeg dan de handleiding van de WEBS module.



6- DIAGNOSTIEK

6.1- Diagnostiek van Domino systeem via DFCP 4

De DFCP 4 module voorziet een waarschuwing via twee rode LEDs op het voorpaneel.

De twee rode LEDs rapporteren het alarm verwant met het falen van de module (MOD.F) en het falen van de bus (BUS.F). De MOD.F signalisatie gebeurt na 5 seconden vertraging gezien vanaf het moment van het falen van de module. De zoektocht naar gefaalde modules kan gedaan worden via de DCP IDE software, wat de map van de installatie op DCP VISIO weergeeft.

Als zowel MOD.F als BUS.F LEDs blijven branden dan betekent dat dat het DFCP 4 geheugen niet correct geprogrammeerd is.

Als het falen van de bus plaatsvind moeten de busverbindingen gecontroleerd worden. Dat falen treed op wanneer de DFCP 4 niets kan verzenden naar de Domino bus.

De twee groene LEDs op het DFCP 4 voorpaneel rapporteert de activiteit van de bus. De POLL LED, als de bus werkt zoals het hoort, flinkt op een reguliere manier gezien de hoeveelheid verbonden modules.

De VAR LED toont, via een flits, het optreden van een statusverandering op één of meerdere inputmodules.

Gedurende de firmware update van de algemene microcontroller binnen de DFCP 4, flitsen de twee rode LEDs afwisselend, terwijl gedurende de firmware update van de tweede microcontroller de twee groene LEDs afwisselend flitsen.

Drie paren van LEDs (rood en geel) op het voorpaneel van de DFCP 4 laten het controleren van de activiteit op de gerelateerde seriële poorten RS232 en RS485 toe.

De volgende tabel vat het gedrag van de LEDs samen met de verschillende statussen:

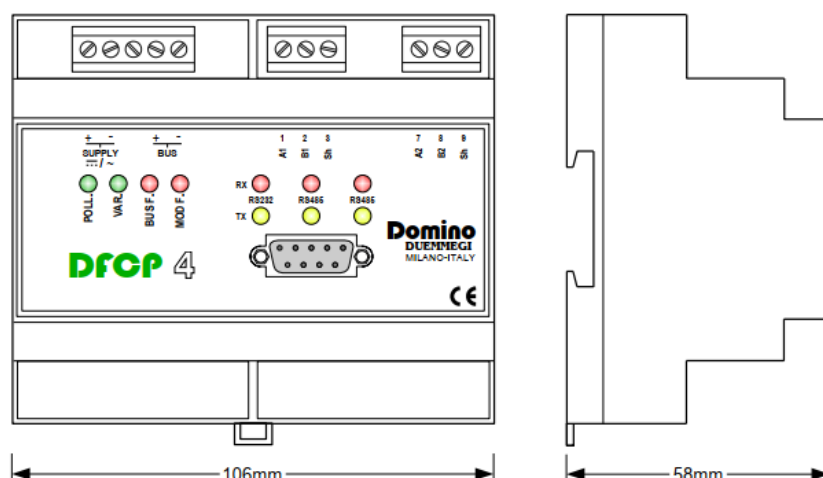
Operating status	POLL	VAR	BUS.F	MOD.F
Normaal	Knippert periodisch	Flitst bij geval van een verandering van een input module	Vast op OFF	Vast op OFF
Falen module	Knippert periodisch	Flitst bij het geval van een verandering van een input module	Vast op OFF	Vast op ON
Falen bus	Vast op OFF	Vast op OFF	Vast op ON	Vast op OFF
FLASH niet geprogrammeerd	Knippert periodisch gelijktijdig		Vast op ON	Vast op ON
Update van algemene microcontroller of FW niet geldig	Knippert periodisch gelijktijdig		Knippert afwisselend	
Update van secundaire microcontroller of FW niet geldig	Knippert afwisselend		X	X

7- Technische kenmerken

Voeding	15V~ (± 20%) or 24V --- (± 25%)
Maximum stroomconsumptie	160mA @ 12V~ / 110mA @ 24V ---
Aantal interne processors	2
Automatisch schakelen standaard/zomeruur	Ja
Typische input naar output reactietijd	40msec
Grootte geheugen gebruikersprogramma	FLASH type 16 Mbytes
Grootte geheugen RAM	256 KWords
Toegelaten virtuele punten	2032
Toegelaten registers	1024, elk 16-bit
Toegelaten timers	512 met tijden van 0 tot 6553 seconden, resolutie 0.1 sec.
Toegelaten counters	1024, elk 16-bit
Programmeerklok	Dagelijks, Wekelijks, Jaarlijks
Geïntegreerde gevorderde scheduler	Wekelijks
Toegelaten inputadressen	255 adressen, elk 16 bit
Toegelaten outputadressen	255 adressen, elk 16 bit
Toegelaten communicatiepoorten	1 x RS232 opto-coupled 2 x RS485 opto-coupled 1 x toegewijde poort 1 x Ethernet poort (optioneel)
Omgang randapparatuur	- Touchscreen video-aansluitingen - SCADA Supervision systems op PC
Koppelen met andere systemen	Via MODBUS RTU en MODBUS TCP/IP protocollen
Behuizing	Modulaire 6M doos voor DIN rail
Bedrijfstemperatuur	-5 - +50 °C
Bewaartemperatuur	-20 - +70 °C
Beveiligingsgraad	IP20

Waarschuwing: DFCP 4 module bezit een herlaadbare batterij. Verwijder die batterij wanneer het toestel wordt weggesmeten. De batterij moet verwijderd worden op een veilige manier volgens de huidige wetten.

8- Afmetingen



9- FXP-XT COMMUNICATIEPROTOCOL

9.1- Berichtenformaat naar betekenis

Het gepatenteerde protocol geïmplementeerd in de DFCP 4 heet FXP-XT. Dat protocol werd speciaal ontworpen om DFCP 4 te laten koppelen met de externe wereld (PC, PLC, enz.) en het NRZ bestaat uit 1 start bit, 8 data bit, geen pariteit en 1 stop bit. De Baudrate kan geselecteerd worden uit: 2400, 4800, 9600, 19200, 38400, 57600, 115200 Baud. DFCP 4 gedraagt zich als een slaafunit, het antwoordt dus enkel op aanvragen van een HOST.

Met het volgende zien jullie dat de numerieke data gerepresenteerd wordt met de 0x notatie, wat ook zo in hexadecimaal formaat hoort te zijn.

De berichten tussen DFCP 4 en HOST hebben het volgende formaat:

Adres	Code	#Byte	Data 1		Data N	ChkSum H	ChkSum L
-------	------	-------	--------	-------	--------	----------	----------

Waar:

- **Adres:** 1 byte, node adres van DFCP 4. Het adres 0x00 is geldig voor ieder node adres
- **Code:** 1 byte, het specificeert de functie van het bericht
- **#Byte:** 1 byte, het aantal bytes op het volgende gegevensveld
- **Data 1 - N:** N databytes
- **ChkSum:** 2 bytes (hoog, laag) van checksum, gelijk aan de complementaire som van de bytes van het bericht, inclusief het adres, de code en het aantal bytes.

De verkrijgbare berichten zijn:

Aanvragen van HOST naar DFCP 4

Code	# Byte	Data Bytes	Beschrijving
0x7F	4	Add_U, Add_H, Add_L, N	Het lezen van het RAM geheugen van N bytes (1-255), startende van het adres gespecificeerd door de eerste 3 databytes. N=0 betekent het lezen van 256 bytes.
0x7E	5 - 256	Add_U, Add_H, Add_L, N, Data1... DataN	Het schrijven naar het RAM geheugen van N bytes (1-252), startende van het adres gespecificeerd door de eerste 3 databytes. (Opmerking 1)
0x7D	4	Add_U, Add_H, Add_L, N	Het lezen van het microcontrollergeheugen van N bytes (1-255), startende van het adres gespecificeerd door de eerste 3 databytes. N=0 betekent het lezen van 256 bytes. (Opmerking 2)
0x7C	5 - 256	Add_U, Add_H, Add_L, N, Data1... DataN	Het schrijven naar het microcontrollergeheugen N bytes (1-252), startende van het adres gespecificeerd door de eerste 3 databytes. (Opmerking 2)
0x7B	2	Mod_Add, N	Het lezen van N (1-32) outputmodules startende van het module adres Mod_Addr.
0x7A	2	Mod_Add, N	Het lezen van N (1-32) inputmodules startende van het module adres Mod_Addr.
0x79	6	Mod_Addr, Ch, Status_H, Status_L, Mask_H, Mask_L	Het schrijven van een kanaal (Ch = 1-4) van een outputmodule (Mod_Addr=1-255). Ch is een constante en moet gelijk aan 1 zijn. Het dekmantel (bit ingesteld op 1) identificeert welke outputpunten aangepast moeten worden.
0x78	3	V_H, V_L, Status	Het schrijven van een virtueel punt. V_H-V_L is het nummer van het punt (1-2032), Status kan 0x00 (voor Vx=0) of 0x01 (voor Vx=1) zijn.
0x70	2	'ID'	ID aanvraag. Het gegevensveld bezit de ASCII code van de twee letters 'I' en 'D' (daarom 0x49 en 0x44)

Opmerking 1: Als het schrijven een output, virtueel punt, register, counter, enz. aanpast, dan zal het commando uitgevoerd worden wanneer de minst belangrijke byte van de Word is geschreven, terwijl het commando niet zal uitgevoerd worden wanneer de belangrijkste byte van de Word wordt geschreven.

Opmerking 2: Om het EEPROM geheugen van DFCP 4 te schrijven/lezen, de berichten 0x7D/0x7C met het adres startend vanaf 0x7FF000 moeten dan gebruikt worden.

Antwoorden van DFCP 4 naar HOST

Code	# Byte	Data Bytes	Beschrijving
0x7F	1 - 256	Data1...DataN	Antwoord op het lezen van een bericht van N bytes van het RAM geheugen.
0x7E	1	0xFF als het schrijven OK is 0x00 als het schrijven niet OK is	Antwoord op het schrijven van een bericht van N bytes naar het RAM geheugen.
0x7D	1 - 256	Data1...DataN	Antwoord op het lezen van het bericht van N bytes van het microcontrollergeheugen
0x7C	1	0xFF als het schrijven OK is 0x00 als het schrijven niet OK is	Antwoord op het schrijven van het bericht van N bytes naar microcontrollergeheugen
0x7B	8 - 256	Data1...Data(Nx8)	Antwoord op het lezen van het bericht van N (1-32) outputmodules startend van adresmodule Mod_Addr. Het antwoord bevat Nx8 bytes in het gegevensveld. De betekenis van iedere blok van 8 bytes is het volgende: Data1-Data2: status/waarde van module Mod_Addr Data3...Data8: altijd 0 (voor toekomstig gebruik)
0x7A	8 - 256	Data1...Data(Nx8)	Antwoord op het lezen van het bericht van N (1-32) outputmodules startend van adresmodule Mod_Addr. Het antwoord bevat Nx8 bytes in het gegevensveld. De betekenis van iedere blok van 8 bytes is het volgende: Data1-Data2: status/waarde van module Mod_Addr Data3...Data8: altijd 0 (voor toekomstig gebruik)
0x79	1	0xFF als het schrijven OK is 0x00 als het schrijven niet OK is	Antwoord op het schrijven van het bericht van een outputmodule (Mod_Addr=1-255).
0x78	1	0xFF als het schrijven OK is 0x00 als het schrijven niet OK is	Antwoord op het schrijven van het bericht van een virtueel punt.
0x70	68	FV1_H, FV1_L, FV2_H, FV2_L, ID1....ID64	Antwoord op aanvraag identificatiecode. Bytes FV1_H -FV2_L keren de versie van de ingeladen firmware terug naar DFCP 4. ID1-ID64 zijn de ASCII codes van de 64 tekens van de identificatiestring.

9.2- Indeling RAM geheugen

De volgende tabel beschrijft de indeling van de RAM in de DFCP 4 voor de algemene, gebruikte parameters.

Opmerkingen: Niet gespecificeerde RAM locaties, in de volgende tabel, zijn bedoeld om gereserveerd of niet gebruikt te worden.

Wanneer het MODBUS RTU protocol gebruikt wordt dan moet het nummer van ieder Word, in de tabel van de volgende paragraaf, door 1 verhoogd worden ALS EN ENKEL ALLEEN ALS de MODBUS-optie gebruikt werd (zie 2.1.4).

9.2.1- Algemene indeling van het RAM geheugen

Byte (HEX)	Word (DEC)	Beschrijving	Opmerkingen
0002-01FF	1-255	Status of waarde van inputmodules	Iedere status of waarde neemt 1 Word in. De inputmodules zijn 255. (Opmerking 1)
0402-05FF	513-767	Status of waarde van outputmodules	Iedere status of waarde neemt 1 Word in. De outputmodules zijn 255. (Opmerking 1)
0902-09FF	1153-1279	Map van de virtuele punten	2032 virtuele punten (enkel digitale) georganiseerd als een blok van 16 punten voor iedere Word (8 punten per byte). (Opmerking 2)
0A00-0A27	1280-1299	Beheer Scheduler	Zie paragraaf 9.2.2 voor details
0F00-0F01	1920	Uren in BCD formaat	Leest van de interne timekeeper chip. (Opmerking 3)
0F02-0F03	1921	Minuten in BCD formaat	Leest van de interne timekeeper chip. (Opmerking 3)
0F04-0F05	1922	Seconden in BCD formaat	Leest van de interne timekeeper chip. (Opmerking 3)
0F06-0F07	1923	Dag van de week in BCD formaat	Leest van de interne timekeeper chip. (Opmerking 3) 1=maandag, 2=dinsdag,... 7 (0=zondag)
0F08-0F09	1924	Dag van de maand in BCD formaat	Lees van de interne timekeeper chip. (Opmerking 3)
0F0A-0F0B	1925	Maand in BCD formaat	Lees van de interne timekeeper chip. (Opmerking 3)
0F0C-0F0D	1926	Jaar in BCD formaat	Lees van de interne timekeeper chip. (Opmerking 3)
0F10-0F11	1928	Aantal binaire events in wachtrij	Enkel lezen.
0F12-0F13	1929	Aantal binaire events die verwijderd moeten worden	Hoeveel opeenvolgende events moeten er verwijderd worden uit de wachtrij.
0F14-0F15	1930	Pointer naar de eerste binaire event	Is het adres van de eerste event na het verwijderen van de laatste.
0F16-0F17	1931	Aantal analoge events in wachtrij	Enkel lezen.
0F18-0F19	1932	Aantal analoge events die verwijderd moeten worden	Hoeveel opeenvolgende events moeten er verwijderd worden uit de wachtrij.
0F1A-0F1B	1933	Pointer naar de eerste analoge event	Is het adres van de eerste event na het verwijderen de laatste.
1000-17FF	2048-3071	Map van de algemene geschikte registers	R0-R1023. 1 Word voor elke register.
1800-1FFF	3072-4095	Map van de counters	C0-C1023. 1 Word voor elke counter.
2000-2FFF	4096-6143	Map van de timer	TIMER0-TIMER511. 4 Words voor elke timer, het eerste bevat de huidige tijd, de andere drie zijn gereserveerd.
3000-3FFF	6144-8191	Data Scheduler	Zie paragraaf 9.2.2 voor details.
4000-7FFF	8192-16383	Lijst van binaire events	2048 events, 8 bytes per event, totaal 16384 bytes. (Opmerking 4)
A000-BFFF	20480-24575	Lijst van analoge events	1024 events, 8 bytes per event, totaal 8192 bytes. (Opmerking 5)
E800-EBFF	29696-30207	Informatie over geconfigureerde input modules	4 bytes per module, offset = 4 x (Module_Address). (Opmerking 6)
EC00-EFFF	30208-30207	Informatie over geconfigureerde output modules	4 bytes per module, offset = 4 x (Module_Address). (Opmerking 6)
F000-F0FF	29952-30719	Diagnostische informatie over input modules	1 byte per module, offset = (Module_Address). (Opmerking 7)
F100-F1FF	30720-30975	Diagnostische informatie over output modules	1 byte per module, offset = (Module_Address). (Opmerking 7)

Opmerking 1: Algemeen, voor digitale inputs en outputs, betekent bit=1 een actieve status en bit=0 non-actieve status. Voor analoge modules bezit de Word de waarde verwijzend naar dat kanaal. De minst belangrijke bit van een Word refereert naar punt 1, de belangrijkste naar punt 16.

Opmerking 2: De minst belangrijke bit van de eerste Word in de map van virtuele punten (Word 1153) is de status van virtueel punt V1, de belangrijkste van dezelfde Word is de status van het virtueel punt V16 enzoverder voor de volgende Words. Bit=1 betekent een actieve status, bit=0 een non-actieve status. Het virtueel punt n is de bit $(n-1)\%16$ (n-1 module 16) van de Word $1153 + \text{INT}[(n-1)/16]$.

Opmerking 3: Deze Words bevatten de huidige status van DFCEP timekeeper chip. Naast het lezen kunnen deze cellen overschreven worden en in dat geval zal de timekeeper chip geüpdatet worden met nieuwe, doorgestuurde parameters (ook in MODBUS protocol). Alle Words gerelateerd aan de informatie van de timekeeper hebben MSByte altijd ingesteld staan op nul, terwijl LSByte de gerelateerde informatie bezit (hh, mm, ss, dag van de week, dag, maand, jaar) in BCD formaat.

Opmerking 4: De binaire eventlijst kan tot 2048 events opslaan en wordt georganiseerd in blokken van 8 bytes per event. Elke 8-byte blok (verwant met een event) is gecodeerd als het volgende:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Module_Address (1-255)							
STATUS		SYS. F	VIRT		I/O		Point (0-15)
-							
-		-	-		Uren		
-		-	Minuten				
-		-	Seconden				
-		-	-		Dag van de maand		
-		-	-		-		Maand

- (SYS.F = 1) & (Module_Address = 0) & (STATUS=1) betekent BUS.F
- (SYS.F = 1) & (Module_Address = 0) & (STATUS=0) betekent BUS. OK
- (SYS.F = 1) & (Module_Address <> 0) & (STATUS=1) betekent MOD.F
- (SYS.F = 1) & (Module_Address <> 0) & (STATUS=0) betekent MOD.OK
- I/O = 1 betekent outputmodule, I/O = 0 betekent inputmodule
- Het symbool '-' betekent "niet gebruikt"

Opmerking 5: De lijst van analoge events (waardes of codes) kan tot 1024 events opslaan en wordt georganiseerd in blokken van 8 bytes per event. Elke blok van 8 bytes (verwant met een event) is gecodeerd als het volgende:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Module_Address (0-255) / nummer Register (laagste deel)							
Waarde of code (Byte_H)							
Waarde of code (Byte_L)							
REG	-	-	Uren				
-	-	Minuten					
Registrnummer (hoogste deel)		Seconden					
-	-	-	Dag van de maand				
-	-	-	-	Maand			

REG = 0: event van inputmodule, REG = 1: event van register
Het symbool '-' betekent "niet gebruikt".

Opmerking 6: De configuratie map (bytes 0xE800-0xE9FF) bezit de informatie gerelateerd aan de busmodules binnen de pollingcyclus van de DFCEP 4. De informatie wordt georganiseerd in twee bytes per module met offset = $2 \times (\text{Module_Address})$, zoals het volgende:

offset 0 (Byte 0xE800-0xE803): niet gebruikt
 offset 4 (Byte 0xE804-0xE807): inputmodule 1
 offset 8 (Byte 0xE808-0xE80B): inputmodule 2
 offset 12 (Byte 0xE80C-0xE80F): inputmodule 3

 offset 1020 (Byte 0xEBFC-0xEBFF): inputmodule 255
 offset 1024 (Byte 0xEC00-0xEC03): niet gebruikt
 offset 1028 (Byte 0xEC04-0xEC07): outputmodule 1

 offset 2044 (Byte 0xEFFC-0xEFFF): outputmodule 255

Tegenstellend tot de andere gevallen, de eerste byte (dat met een even adres) moet geïnterpreteerd worden als de lage byte van de Word en het tweede (met het oneven adres) als hoge byte. Met andere woorden, de bits van iedere Word in deze map moet geïnterpreteerd worden als het volgende:

Bit7 Bit6 Bit5 Bit4 Bit3 Bit2 Bit1 Bit0 Bit15 Bit14 Bit13 Bit12 Bit11 Bit10 Bit9 Bit8

De betekenis van die bits is het volgende:

Eerste Word:

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
- (*)								Type module (**)							

(*) bit 15-8: niet gebruikt

(**) bit 7-0: type module. Contacteer Duemmegi als de codes van de module nodig zijn.

Tweede Word:

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
- (*)					Virtueel punt voor informatie over het falen van de module										

(*) bit 15-11: niet gebruikt

(**) bit 10-0: virtueel punt (zo nodig) voor informatie over het falen van de module, staat in punt/adres formaat. Bits 6-0 toont het adres, bits 10-7 toont het punt. Het virtueel punt zal V_n zijn, waar $n = ((\text{bit6-bit0}) - 1) \times 16 + (\text{bit10-bit7}) + 1$

Opmerking 7: De map van diagnostiek (bytes 0xF000-0xF1FF) bevat de informatie verwant met de verkeerde busmodules. De informatie zit in één byte voor iedere module zoals het volgende:

offset 0 (Byte 0xF000): niet gebruikt
 offset 1 (Byte 0xF001): inputmodule 1
 offset 2 (Byte 0xF002): inputmodule 2

 offset 255 (Byte 0xF0FF): inputmodule 255
 offset 256 (Byte 0xF100): niet gebruikt
 offset 257 (Byte 0xF101): outputmodule 1

 offset 511 (Byte 0xF1FF): outputmodule 255

De betekenis van de bits voor elke byte in deze map is het volgende:

bit 7: niet gebruikt
 bit 6: niet gebruikt
 bit 5: niet gebruikt
 bit 4: het falen van module (1=fout, 0=OK)
 bit 0-3: counter van het opeenvolgende verlies

9.2.2- RAM indeling van de Scheduler

De Scheduler gebruikt twee blokken van RAM-geheugen binnenin de DFCP 4; Words 1280-1299 en Words 6144-8191. Het eerste blok wordt gebruikt om het lezen en schrijven van een gegeven dag Dy van een gegeven scheduler Sx (Dy= 1 (MON) tot 7 (SUN) en Sx= 1 tot 16) parametrisch (en dus gemakkelijker) te maken. De tweede blok bevat in plaats daarvan alle 8 ON en OFF tijden van alle 7 dagen van alle 16 schedulers.

Voor het beheer door een supervisor of een webserver, wordt het aangeraden om enkel te handelen op het eerste RAM blok, uitgesloten waar ze eigenlijk echt de informatie van iedere scheduler mappen. Die aanpak versimpelt nogal het werk en maakt het parameteriseerbaar door de specificatie van twee waarden: het nummer van de scheduler (Sx = 1-16) en de dag van de week (Dy = 1-7). Wanneer één van deze twee waarden worden geschreven, dan zal DFCP 4 de data gerelateerd aan de dag Dy van de scheduler Sx kopiëren in de eerste RAM blok. Omgekeerd, het schrijven van een waarde (bv. een tijd) in een Word van het eerste RAM blok, zal DFCP 4 de data verwant aan de dag Dy van de scheduler Sx doen kopiëren van het blok naar zijn echte positie in de tweede RAM blok.

Byte (HEX)	Word (DEC)	Inhoud	Opmerkingen
START BLOK 1			
A00-0A01	1280	1ste ON scheduling time	1ste byte: uur (0-24), 2 ^{de} byte: minuten (0-59) als uur=24, dan moeten minuten = 0 zijn; zie ook Opmerking 1 en Opmerking 2. Het schrijven van een waarde in deze Words, zal de DFCP 4 de verwante scheduler data, aangeduid door Word 1297 (nummer scheduler Sx) en 1298 (nummer dag Dy) doen updaten.
A02-0A03	1281	1ste OFF scheduling time	Idem
A04-0A05	1282	2de ON scheduling time	Idem
A06-0A07	1283	2de OFF scheduling time	Idem
...	...	...	...
A1C-0A1D	1294	8ste ON scheduling time	Idem
A1E-0A1F	1295	8ste OFF scheduling time	Idem
A20-0A21	1296	Scheduling times aanleggen	De eerste 8 bits representeren het aanleggen van de gerelateerde scheduling time (bit=0: scheduler uit, bit=1: scheduler aan). Bit0 = 1ste scheduling time Bit1 = 2de scheduling time Bit2 = 3de scheduling time Bit3 = 4de scheduling time Bit4 = 5de scheduling time Bit5 = 6de scheduling time Bit6 = 7de scheduling time Bit7 = 8ste scheduling time
A22-0A23	1297	Nummer scheduler Sx	Het schrijven van een waarde 1 tot 16 over deze Word, zal de DFCP 4 de data van de dag Dy van de scheduler Sx over de Words 1280-1296 (8 scheduling times ON-OFF + aanleggen) doen kopiëren.
A24-0A25	1298	Nummer dag Dy	Het schrijven van een waarde 1 (MON) tot 7 (SUN) over deze Word, zal de DFCP 4 de data van de dag Dy van de scheduler Sx naar de Words 1280-1296 (8 scheduling times ON-OFF + aanleggen) doen kopiëren.
A26-0A27	1299	Kopie scheduler	Het schrijven van ieder waarde over deze Word, de DFCP 4 zal de data van de huidige dag naar de volgende (van dezelfde scheduler Sx) doen kopiëren en de dag Dy zal verhoogd worden met 1 of, als het 7 was, terugzet worden op 1.
EINDE BLOK 1			
...	...	...	...
START BLOK 2			
000-30FF	6144-6271	Data van scheduler 1	128 Words
100-31FF	6272-6399	Data van scheduler 2	128 Words
...	...	...	...
F00-3FFF	8064-8191	Data van scheduler 16	128 Words
EINDE BLOK 2			

Opmerkingen:

- (1) Het is mogelijk om de tijd 24:00 te specificeren als het bestuurd punt ON moet zijn tijdens middernacht, om zo het uitschakelen te vermijden tussen 23.59 en 00:00.
- (2) Waarden van uren groter dan 24 en/of minuten groter dan 59 zijn niet geldig.

10- DCP IDE: GEÏNTEGREERD ONTWIKKELINGSPLATFORM VOOR APPLICATIES DIE DFCP 4 GEBRUIKEN

10.1- Beschrijving van het softwarepakket DCP IDE

DCP IDE is een 'Integrated Development Environment' om het ontwikkelen van programma's voor DFCP 4 te ondersteunen. Het DCP IDE pakket komt compleet met een Editor, Compiler, Transfer, Simulator en Supervisor voor de werking van DFCP 4 en van de installaties.

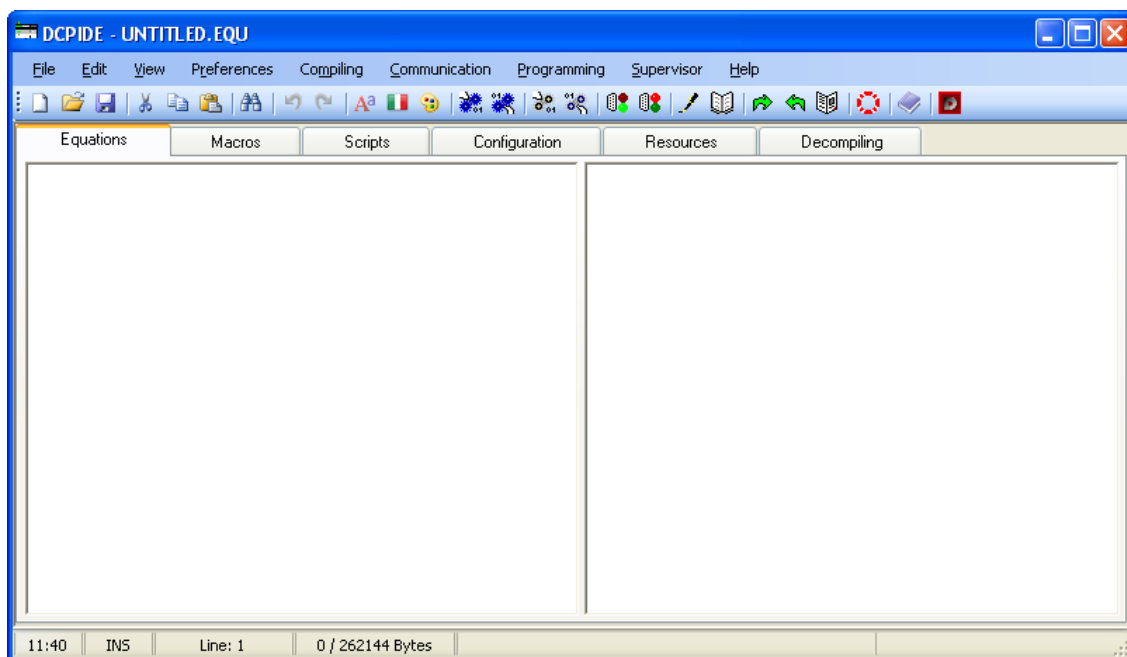
Het pakket is gemaakt uit verschillende programma's zoals beschreven in het volgende:

- DCP IDE is DFCP 4's program editor, integreert de compiler, heeft de configuratietools voor speciale modules (bv. DFDP, DFDM, DFWRX, DFCT, enz.) en de "write to / read from" voor DFCP 4 en "write to / read from" voor Domino modules.
- DCP VISIO is een tool die toezicht op geïnstalleerde input- en outputmodules en alle andere DFCP 4 parameters (counters, registers, virtuele punten, enz.) toelaat. Deze tool kan werken via een verbinding met DFCP 4 via de seriële poort of het kan het programma, geschreven door DCP IDE, simuleren om het zo te debuggen voordat het getransfereerd wordt naar het FLASH-geheugen van DFCP 4.
- DCP MAP is een gevorderde tool wat de toegang tot het 'hart' van de DFCP 4 toelaat. Het gebruik van deze tool wordt vrijgehouden enkel voor gebruik door experts.

Door het gemakkelijk gebruik van DCP IDE en zijn vele features en nuttigheden, laat het een snelle ontwikkeling en configuratie van de DFCP 4 controller toe, volgens de vereisten van de installatie waar het geïnstalleerd wordt. De werking en duidelijke menu items laten het gebruik van DCP IDE direct toe, waardoor meer tijd gaat naar het ontwikkelen van applicaties dan het lezen van gebruikershandleidingen.

10.2- DCP IDE

DCP IDE ziet er zoals de volgende afbeelding uit:



Het algemene scherm van het programma heeft 6 tabbladen: Equations, Macros, Scripts, Configuration, Resources en Decompiling.

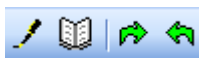
Het Equations tabblad laat het schrijven en aanpassen van vergelijkingen toe en is verdeeld in twee delen: het linkerdeel voor het intern programma van DFCP 4, het rechterdeel voor het programma dat naar de veldmodules moet. Het rechterdeel kan ook de editor van BDTools vervangen, ook al is het mogelijk om dat laatste te gebruiken voor het programmeren van veldmodules en DCP IDE voor het programmeren van DFCP 4.

Iedere knop op de knoppenbalk toont de beschrijving van zijn functie door simpelweg de muis op de knop zelf te plaatsen.



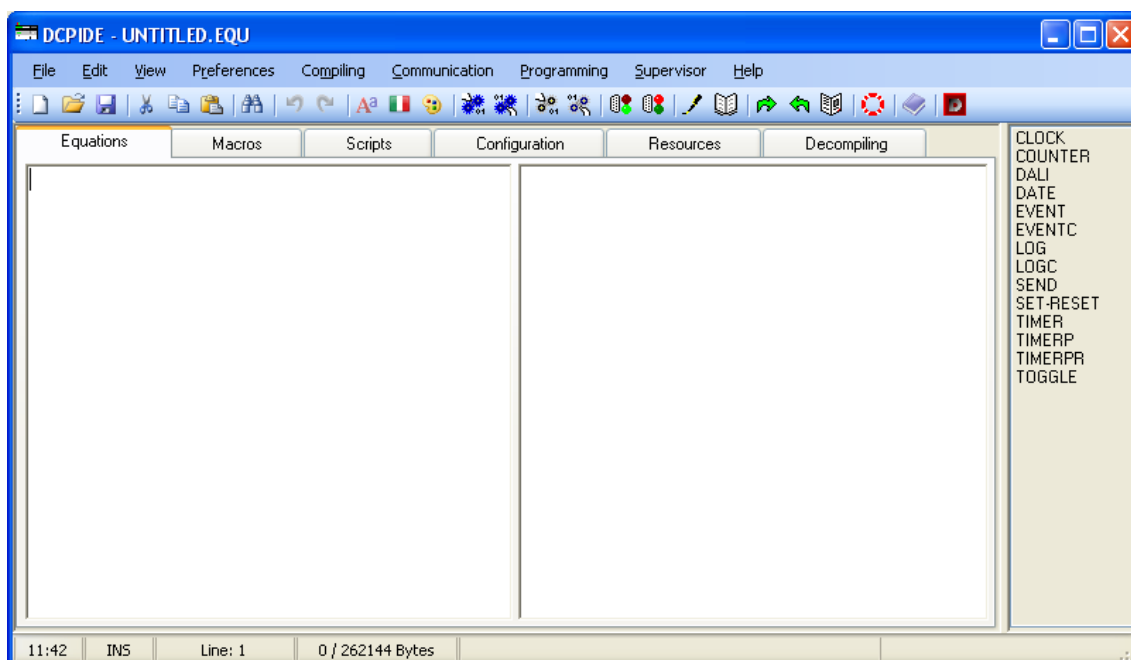
De meerderheid van de knoppen en menu items zijn zo vanzelfsprekend dat er geen verdere uitleg nodig is.

De knoppen  verwijzen naar het compileren en decompileren van het DFCP 4 programma, terwijl de knoppen  gelijkaardig zijn maar die verwijzen naar het programma in de veldmodules.

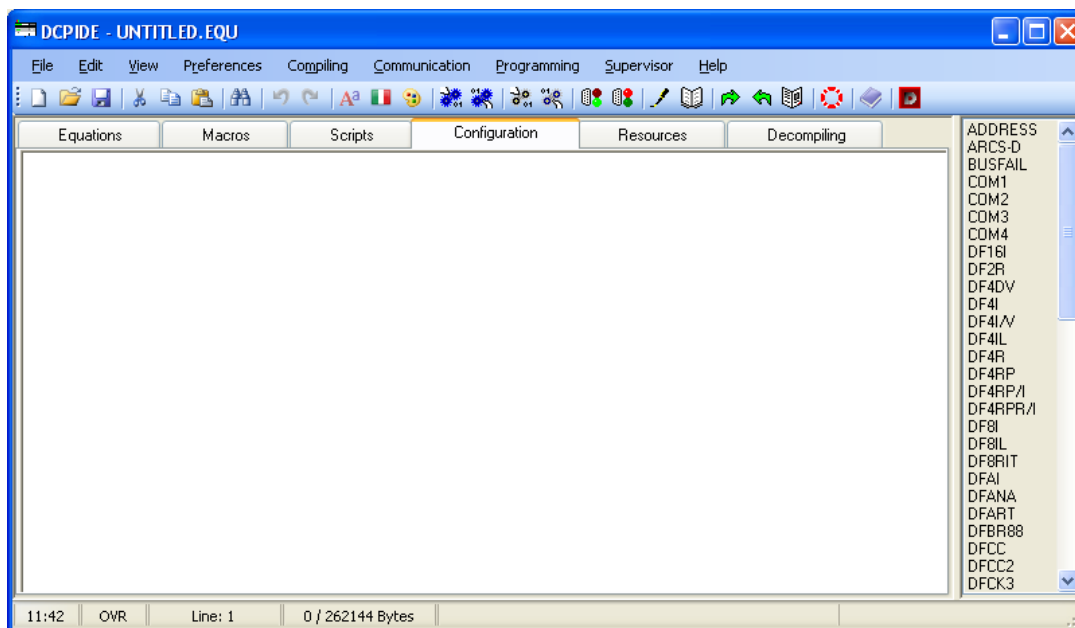
Hetzelfde concept is ook van toepassing op de knoppen : de eerste twee voeren de transfer van het DFCP 4 programma uit, de andere twee de transfer naar veldmodules.

Knop  laat het toewijzen van Domino moduleadressen toe.

Knop  (of menu item View – Show Keywords List) is de 'reddingsgordel' en laat het toe om een deel, op de rechterkant, aan- of uit te schakelen die alle sleutelwoorden in de gerelateerde TAB bevat.

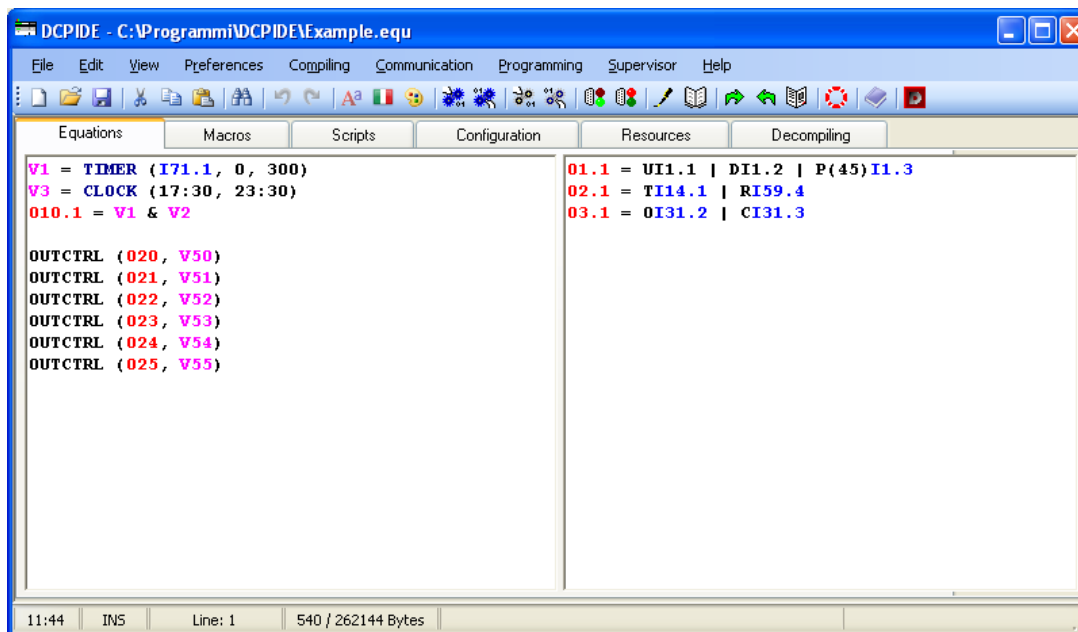


Wanneer de Configuration TAB wordt geselecteerd, zal een lijst van alle verkrijgbare Domino modules verschijnen zoals op de volgende screenshot:

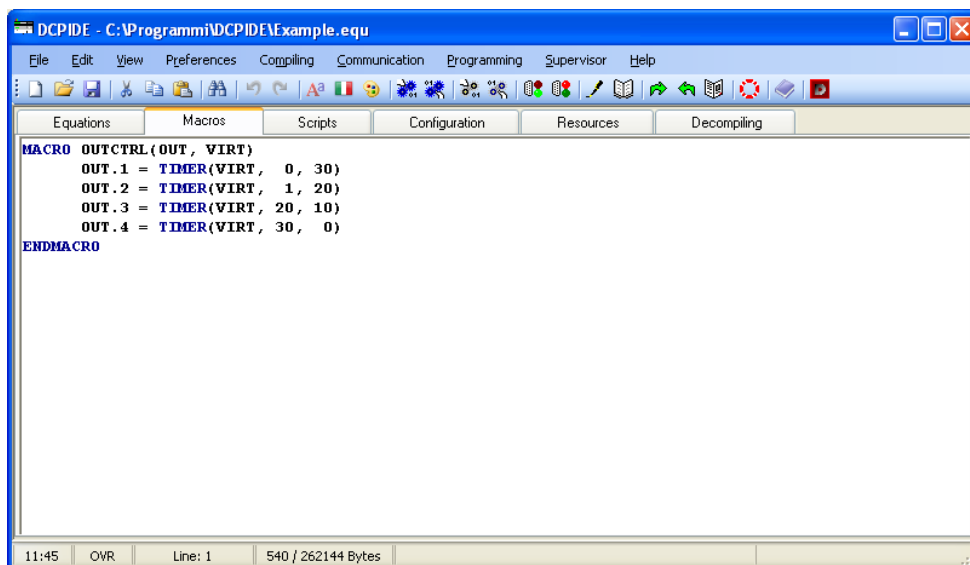


Door dubbel te klikken op één van de sleutelwoorden in de reddingsgordel zal het gerelateerd voorbeeld geplaatst worden in de geopende werkruimte. Het ingebracht voorbeeld moet afgemaakt worden zoals gevraagd.

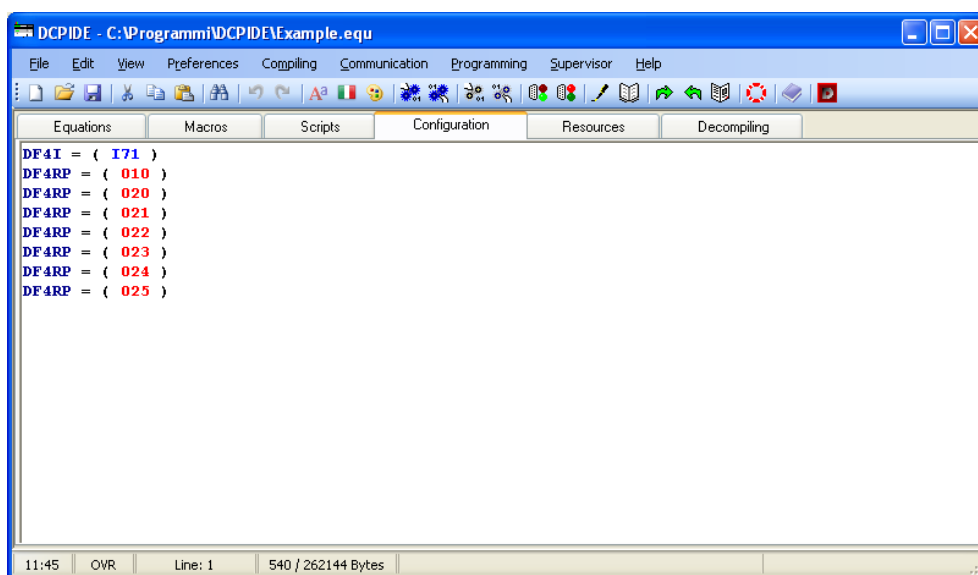
De Equations TAB laat het schrijven van standaardvergelijkingen voor DFCP 4 (linkerdeel) en Domino modules (rechterdeel) toe.



De Macros TAB laat het schrijven van Macro definities (geldig voor zowel DFCP 4 als Domino modules) toe.



De Configuratie TAB laat het schrijven van een lijst van de gebruikte modules in het DFCP 4 programma of een lijst van modules die door DFCP 4 worden gepolled, toe. Andere informatie, zoals bijvoorbeeld het adres, kan ook geplaatst worden in deze werkruimte.



De Resources TAB (alleen lezen) bevat, na het compileren, de informatie over de gebruikte resources in het zojuist gecompileerd programma.

[illegible]

De Decompiling TAB (alleen lezen) is voorbehouden voor experts en bevat, na het decompileren, de informatie hoe de compiler het geschreven programma heeft geïnterpreteerd.

DCPIDE - C:\Programmi\DCPIDE\Example.equ

File Edit View Preferences Compiling Communication Programming Supervisor Help

Equations Macros Scripts Configuration Resources **Decompiling**

```

DF4I = ( I71 )
DF4RP = ( 010 )
DF4RP = ( 020 )
DF4RP = ( 021 )
DF4RP = ( 022 )
DF4RP = ( 023 )
DF4RP = ( 024 )
DF4RP = ( 025 )

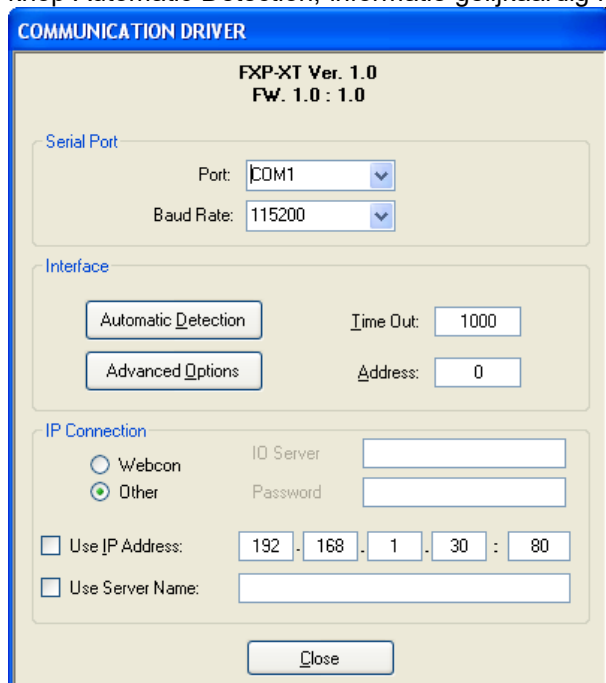
010.1 = V1 & V2
020.1 = TIMER1 ( V50 , 0 , 30 )
020.2 = TIMER2 ( V50 , 1 , 20 )
020.3 = TIMER3 ( V50 , 20 , 10 )
020.4 = TIMER4 ( V50 , 30 , 0 )
021.1 = TIMER5 ( V51 , 0 , 30 )
021.2 = TIMER6 ( V51 , 1 , 20 )
021.3 = TIMER7 ( V51 , 20 , 10 )
021.4 = TIMER8 ( V51 , 30 , 0 )
022.1 = TIMER9 ( V52 , 0 , 30 )
022.2 = TIMER10 ( V52 , 1 , 20 )

```

11:46 OVR Line: 1 540 / 262144 Bytes

10.2.1- Seriële communicatie en firmware versies van DFCP 4

De knop  opent de seriële communicatie met DFCP 4, terwijl de knop  het sluit. Het scherm dat verschijnt bij het openen van de communicatie is het ene wat hier getoond wordt. Eens de communicatie aangelegd werd via de knop Automatic Detection, informatie gelijkaardig met “FW – 1.0 : 1.0” zal dan verschijnen. Het eerste nummer op de



linkerkant is de FW versie van de algemene microcontroller van DFCP 4 (1.0 in dit voorbeeld), terwijl het nummer op de rechterkant het FW versie is van de tweede microcontroller (1.0 in dit voorbeeld).

Het label “FXP-XT Ver. 1.0” op het getoonde scherm is de versie van de communicatie driver wat inbegrepen is in het DCP IDE pakket.

De “Time Out” is de maximumtijd dat de DCP IDE wacht voor een antwoord van de DFCP 4 en “Address” is het adres toegekend aan DFCP 4 om gepolled te worden (hou er rekening mee dat door het specificeren van het adres naar 0, de communicatie dan zal plaatsvinden ongezien welk adres aan de DFCP 4 werd toegekend).

DCP IDE kan ook met DFCP 4 communiceren via het ethernet. In dat geval kan die communicatie aangelegd worden via het aanklikken van “Other” en het IP-adres en de poort van DFCP 4 (of een ander toestel die DFCP 4 aan het netwerk koppelt) en door dan te klikken op het “Use IP Address” selectievakje. Op die manier zal DCP IDE berichten versturen naar de ethernetpoort van de PC waarop geïnstalleerd is, in plaats van RS232. Het antwoord van DFCP 4 zal dan de omgekeerde weg volgen.

“Use Server Name” laat het specificeren van een adres voorzien

door een DNS service toe (bv. wanneer toegang gevraagd wordt van WAN en waar DFCP 4 geïnstalleerd is in een LAN dat communiceert met WAN via een dynamische IP).

Een andere mogelijkheid is via Webcon. In dat geval, na het selecteren van het gerelateerde item, voer de IO servernaam (over het algemeen Domino) en het wachtwoord in, dan het IP-adres en de poort van Webcon en selecteer uiteindelijk “Use IP Address” (of voer het DNS service adres in en selecteer dan “Use Server Name”). Raadpleeg Webcon’s gebruikershandleiding voor meer details.

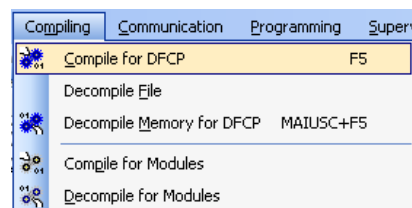
10.2.2- Compileren

Houd altijd in gedachten dat in een Domino systeem met DFCP 4 er twee programma’s betrokken zijn. Eén voor de controller en één voor de veldmodules. Om precies te zijn, het programma voor de veldmodules is gemaakt door een set van kleine programma’s die getransfereerd moeten worden naar iedere gerelateerde outputmodule (geïdentificeerd door zijn adres). Het is ook mogelijk dat de veldmodules geen enkel programma bevatten omdat in dat geval alle functies door de DFCP 4 uitgevoerd zullen worden.

Zoals gezegd in een eerder hoofdstuk bestaat het compileren van een programma uit de translatie van een file met extensie .EQU (corresponderend met de inhoud van de editor van DCP IDE) in een file met extensie .BIN, geschikt om opgeslagen te worden in het geheugen van de DFCP 4 of het geheugen van veldmodules.

Het compileren van een programma (zowel voor DFCP 4 of veldmodules) is de voorafgaande toestand voor een transfer. DCP IDE kan kiezen tussen de volgende mogelijkheden:

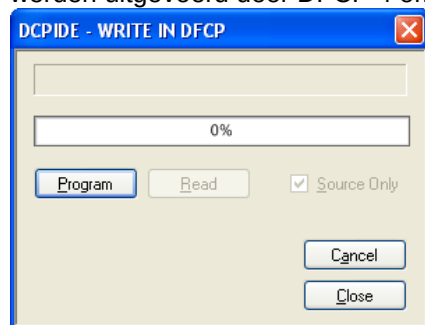
- Compileer enkel het deel van het programma gerelateerd aan DFCP 4 (knop  of item “Compile for DFCP 4” in het Compile menu.
- Compileer enkel het deel van het programma gerelateerd aan de Domino outputmodules (knop  of “Compile for Modules” in de Compile menu)



Het compileerproces heeft geen nood aan een verbinding tussen DFCP 4 en de PC. Als één of meerdere errors gedetecteerd zouden worden tijdens het compileren, dan zullen ze getoond worden op het scherm in een apart venster. Het is ook mogelijk dat de compiler WARNINGS vertoond. Dat betekent dat er geen errors werden gevonden (het compileerproces werd dus uitgevoerd) maar dat er sommige zaken werden gedetecteerd die tot problemen kunnen leiden, wat best wordt bekeken voordat het programma naar de DFCP 4 wordt getransfereerd.

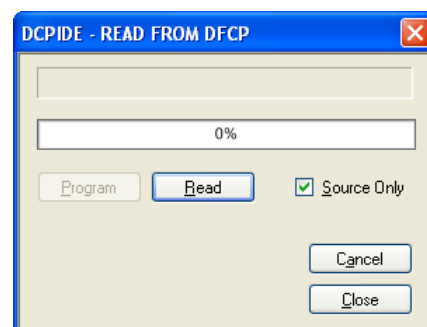
10.2.3- Transfereren

Zoals eerder gezegd is er in een Domino systeem met een DFCP 4 controller altijd twee programma's (tenzij alle functies werden uitgevoerd door DFCP 4 omdat het geheugen van de veldmodules leeg waren).



Om het programma te transfereren naar DFCP 4, het programma aan de linkerkant van de Equations TAB, druk op de knop  of selecteer "Write to DFCP 4" vanuit het Programming menu. Het volgende scherm, linker afbeelding, zal dan verschijnen. Druk op "Program" om het huidige programma naar DFCP 4 over te schrijven.

Het drukken op de knop  of het selecteren van "Read from DFCP 4" vanuit het Programming menu zal een omgekeerd proces doen starten. Het volgende scherm, rechter afbeelding, zal dan verschijnen.

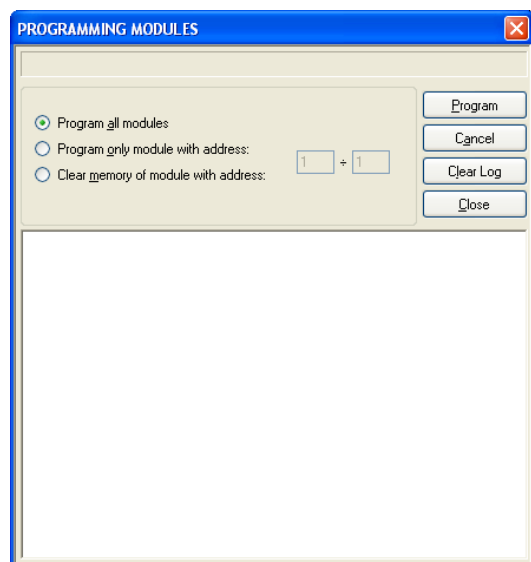


Het aanvinken van de "Source Only" optie zal het source programma van DFCP 4 en het programma van de veldmodules naar uw PC transfereren, zoals ze werden aangemaakt met eventueel de opmerkingen.

Het niet aanvinken van de "Source only" functie op het scherm "Read from DFCP 4" zal de hele inhoud van FLASH naar uw PC doen downloaden. Die laatste procedure neemt veel tijd in beslag en wordt enkel gebruikt voor specifieke redenen (bv. diagnostiek).

Opmerking: het programma van de veldmodules die worden gelezen door de DFCP 4 door de zojuist beschreven optie "Source Only" is het ene wat door DFCP 4 is opgeslagen uit de laatste transfer.

De knop  (of "Write to Modules" vanuit het Programming menu) activeert het transfereren naar de veldmodules. Het volgende venster, linker afbeelding, zal dan verschijnen. In dat venster is het mogelijk om te kiezen tussen het programmeren van alle veldmodules (echte en virtuele) of het programmeren van één enkele module aangeduid door zijn adres in het tekstvak te geven. Het is ook altijd mogelijk het programma binnen de gekozen module helemaal te verwijderen wat belangrijk is wanneer de functie uitgevoerd moet worden door een outputmodule die exclusief door de DFCP 4 bestuurd moet worden.



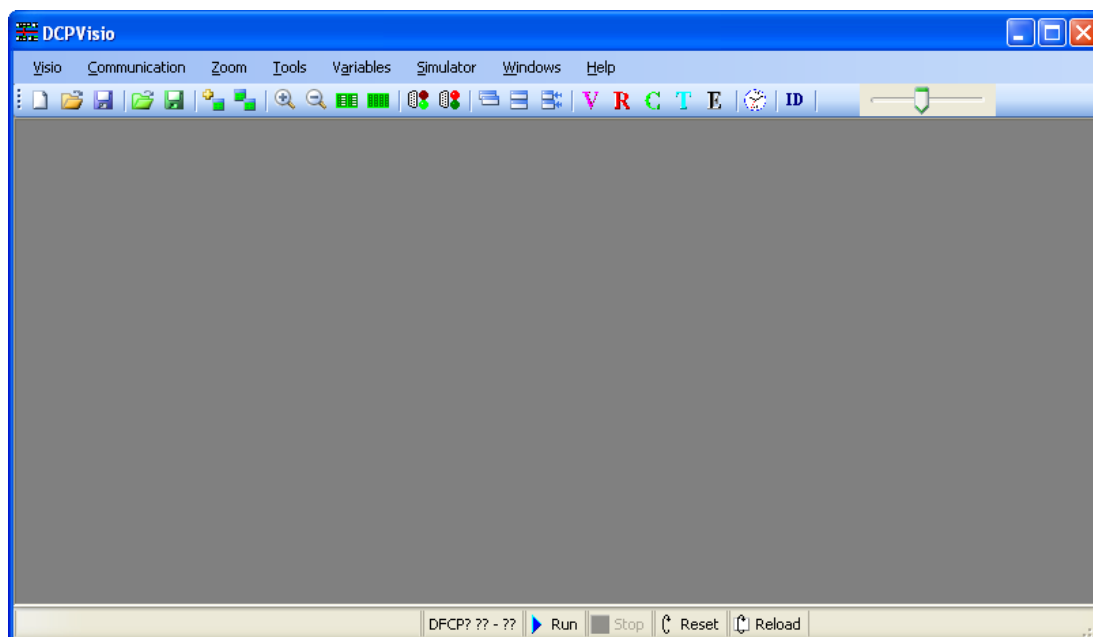
Het witte deel van het getoonde venster zal de lopende status van het transfereren bevatten waar mogelijke errors kunnen verschijnen.

De knop  (of "Read from Modules" vanuit het Programming menu) veroorzaakt de omgekeerde werking wat het lezen van het programma van alle veldmodules is of het lezen van het programma binnen een aangeduide module door zijn adres in het tekstvak te geven. Het programma gelezen van de veldmodule zal verschijnen op het linker beneden deel van het venster. Om dat gelezen programma te transfereren naar de editor van DCP IDE, druk dan op "Copy" in het venster MODULES READING, sluit dan dat laatste scherm en selecteer "Paste" in het Edit menu.

Hou er rekening mee dat er geen opmerkingen worden geplaatst in de veldmodules, het programma wordt wel degelijk in de module geladen alleen worden de opmerkingen in de source programma overgeschreven.

10.3- DCP VISIO

DCP VISIO ziet er zoals de volgende afbeelding uit:



Iedere knop op de knoppenbalk toont de beschrijving van zijn functie door simpelweg de muis op de knop te plaatsen.



De knop  opent de seriële communicatie met DFCP 4, terwijl de knop  het sluit.

Opmerking: Aangezien de communicatie driver hetzelfde is voor het hele software pakket, als de seriële communicatie opengezet werd vanuit DCP IDE, dan zijn de resultaten van de communicatie hetzelfde als het werd opgezet in DCP VISIO en vice versa.

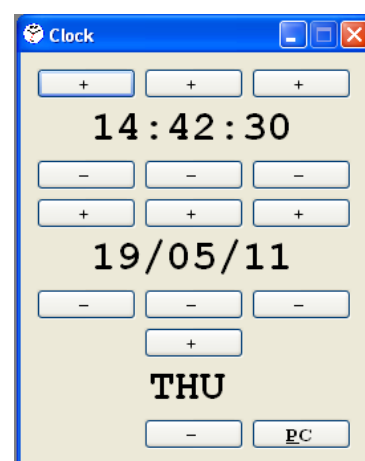
De knoppen  laten het toe om respectievelijk het venster van virtuele punten, registers, counters, timers en events te tonen (zowel digitaal als analoog).

Het venster om te lezen of in te stellen, geopend door de knop  is zoals de rechterafbeelding. Het klokkenpaneel vertoont Uren:Minuten:Seconden op de eerste lijn, Dag/Maand/Jaar op het tweede en de dag van de week op de derde.

Als de seriële communicatie met DFCP 4 geopend wordt, dan zal de gerelateerde huidige tijd en datuminstellingen getoond worden. Als, langs de andere kant, de seriële communicatie met DFCP 4 gesloten wordt, dan zal een sequentie van streepjes getoond worden. De knoppen + en – zullen het verwante item doen toenemen of afnemen.

Bij iedere verandering in de instellingen, via de knoppen + en –, worden de nieuwe instellingen automatisch geüpdatet door de DFCP 4.

De knop “PC” transfereert datum en tijd ingesteld op de PC naar de DFCP 4.



De knop  zal op de status bar, aan de onderkant van het DCP VISIO venster, het firmware versie van de geconnecteerde DFCP 4 tonen.

Het Visio menu laat de volgende operaties toe:

New Group: opent een nieuwe groep

Open Group: laadt een opgeslagen groep van een file in

Save Group: slaat de huidige groep op in een file

Save Group As: slaat de huidige groep op met een verschillende filenaam

New Project: maakt een nieuw project aan

Open Project: laadt een opgeslagen project van een file in

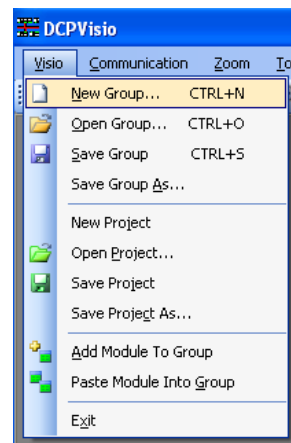
Save Project: slaat het huidig project op naar een file

Save Project As: slaat het huidig project op met een verschillende filenaam

Add module To Group: voegt een module toe aan de huidige groep

Paste Module Into Group: plakt een module in de huidige groep

Exit: verlaat het programma



De Tools menu laat de volgende functies toe:

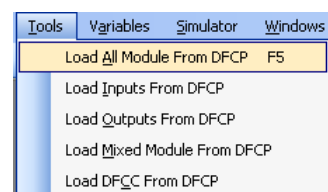
Load All Modules from DFCP: maakt vier groepen aan (Inputs, Outputs, Mixed en DFCC)

Load Inputs from DFCP: maakt een groep aan met alle geconfigureerde inputmodules

Load Outputs from DFCP: maakt een groep aan met alle geconfigureerde outputmodules

Load Mixed Modules from DFCP: maakt een groep aan met alle mixed modules

Load DFCC van DFCP: maakt een groep aan met alle DFCC modules



Om de map van alle geïnstalleerde modules te tonen, volstaat het normaal om de optie “Load All Modules from DFCP 4” te selecteren. Op die manier zullen alle geconfigureerde modules in DFCP 4 getoond worden en natuurlijk enkel en alleen maar wanneer de DFCP 4 is verbonden of als de simulator geactiveerd is.

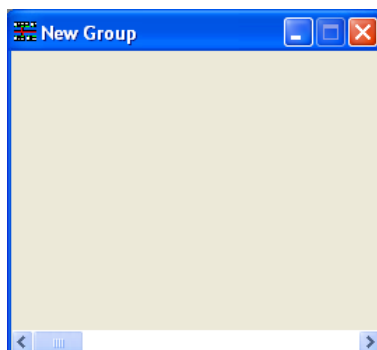
Het is ook mogelijk om afgestemde groepen te maken waar input- en outputmodules en virtuele punten zijn inbegrepen. Hoe zo'n groepen worden aangemaakt zal nu beschreven worden.



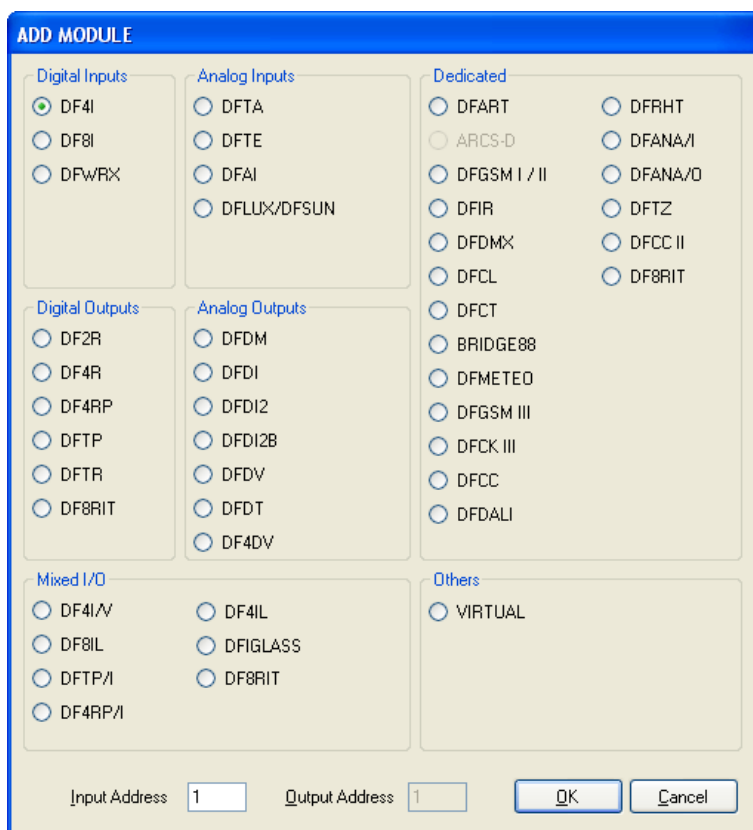
De slider op de knoppenbalk laat het veranderen van de polling periode, van PC naar DFCP 4 (als het verbonden is), toe. Het bewegen van de slider naar links verlaagt de polling zodat het updaten van de objecten op de vensters snel gaat. Het naar rechts bewegen van de slider doet de periode toenemen zodat het updaten van de objecten op de vensters traag gaat. Als de slider helemaal rechts staat dan zal de communicatie met DFCP 4 stopgezet worden.

10.3.1- Groepen

Selecteer “New Group” vanuit het Visio menu. Een nieuw venster zal dan geopend worden zoals op de volgende afbeelding:



De input- en outputmodules en de virtuele punten kunnen, zoals gewenst, in dat venster ingebracht worden. Druk op de knop  om een module toe te voegen. Het volgende scherm zal dan verschijnen:



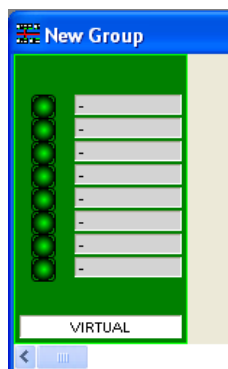
The 'ADD MODULE' dialog box contains several categories of modules:

- Digital Inputs:** DF4I, DF8I, DFWRX
- Analog Inputs:** DFTA, DFTE, DFAI, DFLUX/DFSUN
- Digital Outputs:** DF2R, DF4R, DF4RP, DFTR, DF8RIT
- Analog Outputs:** DFDM, DFDI, DFDI2, DFDI2B, DFDV, DFDT, DF4DV
- Mixed I/O:** DF4I/V, DF8IL, DFTR/I, DF4RP/I
- Others:** VIRTUAL
- Dedicated:** DFART, ARCS-D, DFGSM I / II, DFIR, DFDMX, DFCL, DFCT, BRIDGE88, DFMETEO, DFGSM III, DFCK III, DFCC, DF4ALI, DFRHT, DFANA/I, DFANA/O, DFTZ, DFCC II, DF8RIT

At the bottom, there are fields for 'Input Address' (set to 1) and 'Output Address' (set to 1), along with 'OK' and 'Cancel' buttons.

Selecteer één van de gelijste modules en specificeer het adres in het tekstvakje (voor mixed en speciale modules kan zowel het input- en outputadres gespecificeerd worden).

Om een kaart in te brengen die de virtuele punten van DFCEP 4 rapporteert (gemaakt door 8 punten, toegekend in verschillende volgorde), selecteer dan "VIRTUAL" in de "Others" kolom. Voor dat geval is er natuurlijk geen adres nodig. Het nummer van ieder virtueel punt zal zoals het volgende toegekend worden. Na het selecteren van de "VIRTUAL" optie, druk dan "OK". Het venster ziet er dan uit zoals het volgende:



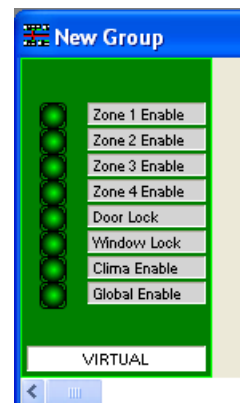
The 'New Group' dialog box shows a list of virtual points. The first point is selected and labeled 'VIRTUAL'.

Houd dan de Shift toets in en dubbelklik op de virtuele LED die aan een virtueel punt van de DFCEP 4 toegekend moet worden (wees zeker om OP de LED te drukken). Een gele label zal getoond worden op de plaats van de geselecteerde LED. Typ dan een nummer binnen het bereik van 1 tot 2032 in om die LED aan het gewenste virtuele punt toe te kennen.

Klik op de rechterkant van iedere LED terwijl Shift ingedrukt blijft om een label aan het verwante punt toe te kennen. Klik dan uiteindelijk op de witte band aan de bovenkant van de virtuele module (hou altijd Shift in) om het een naam te geven.

Om het virtueel punt die toegekend is aan een virtuele LED te controleren of aan te passen, klik dan simpelweg op de LED zelf met Shift ingedrukt.

Dezelfde operatie laat het ook toe om de andere velden aan te passen.

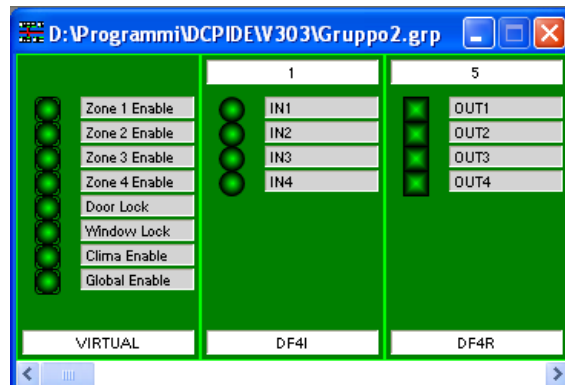


The 'New Group' dialog box shows a list of virtual points. The first point is selected and labeled 'VIRTUAL'.

Het finaal resultaat kan eruit zien zoals de afbeelding hier rechts.

Stel dat u een DF4I module en een DF4R module aan dezelfde groep wilt toevoegen zoals op de afbeelding. Merk op dat het adres van elke module vertoond zal worden in een wit tekstvakje aan de onderkant van dezelfde module.

Het schrijven en aanpassen zoals eerder beschreven (klikken terwijl Shift wordt ingehouden) kan uitgevoerd worden op elke module in het groepsvenster. Gebruik het dus om de naam te veranderen die toegekend is aan iedere module (het wit tekstvakje aan de onderkant) of om het adres te veranderen (het wit tekstvakje aan de bovenkant).



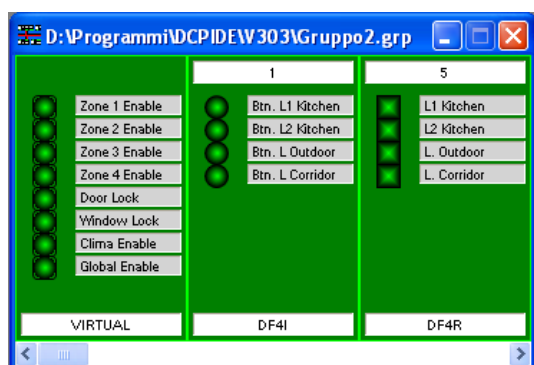
De kleur van de symbolen kunnen grijs en rood zijn.

Grijs: de module is niet geconfigureerd in het huidig programma

Rood: de module antwoordt niet op DFCP 4



De vier knoppen in het groepsvenster kunnen de grafische modus veranderen. De eerste twee modi kunnen in- en uitzoomen. Er zijn tot vier zoomlevels beschikbaar. De andere twee knoppen werken op de labels, het respectievelijk tonen of verstoppen van hen.



De namen van elke input en output worden op dezelfde manier toegekend zoals eerder beschreven (dubbeltklik op de rechterkant van de LED).

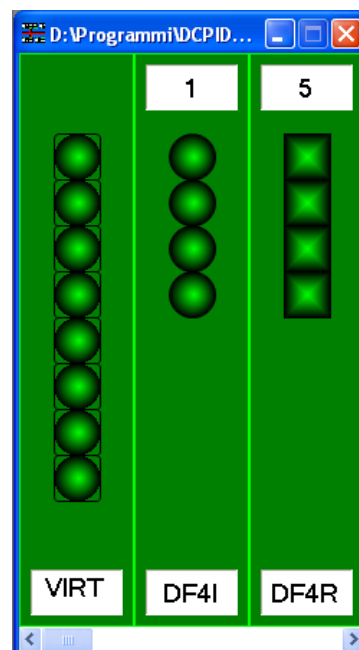
De afbeelding aan de rechterkant toont de optie verwant aan de grafische modus zonder labels en waar een zoom level 3 werd toegepast.

Om een module (echte of virtuele) uit een groep te verwijderen, klikt u op een willekeurig deel van de module zelf

waarbij de Alt-toets is ingedrukt. Een bevestiging van annulering wordt dan nog gevraagd.

Het is ook mogelijk om een module te kopiëren, binnen dezelfde groep of tussen verschillende groepen, door het te kopiëren en te plakken. Om een module te kopiëren klik je op een deel van de module terwijl Ctrl is ingedrukt. Om dan te plakken druk je

op de knop  om de bestemming te bepalen.

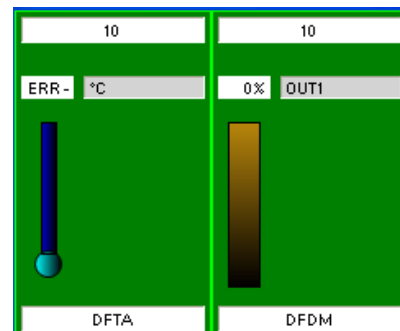


Iedere groep kan opgeslagen (of geüpdatet) worden in een file door de knop  of door het gerelateerde menu item en kan dan herladen worden door de knop .

Voor analoge modules die bij het analoge type horen, laat DCP VISIO het instellen van een meetschaal toe. De afbeelding aan de rechterkant toont twee analoge modules (DFTA en DFDM) binnen groep van DCP VISIO.

De analoge input- en outputwaarde wordt vertoond zowel als staafdiagram als numerieke waarde.

Als de module dat toelaat (maar dat is niet het geval in dit voorbeeld), klik met rechter muis op het numerieke vak om een venster te openen die het veranderen van de instelling van de meetschaal toelaat. De inhoud van het venster hangt af van de gekozen module.



10.3.2- Projecten

DCP VISIO laat het opslaan van al zijn huidige instellingen, bedoeld als geopende groepen, vensters, posities en dimensies van het scherm, zoom levels en grafische levels, enz... toe.

Om een project aan te maken, druk op de knop  van DCP VISIO of selecteer het menu item "Save Project" vanuit het Visio menu.

Om een eerder opgeslagen project terug te roepen, druk dan op de knop  van DCP VISIO, of selecteer het menu item 'Open Project' vanuit het Visio menu.

10.3.3- Simulator

DCP VISIO bezit een simulator wat het DFCP 4 programma (een deel ervan) kan testen en debuggen. De simulator is een volledige software tool waarvoor geen seriële verbinding met DFCP 4 nodig is.

De simulator toont het gedrag van de outputparameters van een programma (bv. echte- en virtuele outputpunten, registers, counters, enz.) als het resultaat van sommige prikkels aan de inputparameters.

Om de simulator te activeren, selecteer "Run" in het simulator menu. Selecteer dan de file die het programma bevat die gesimuleerd moet worden (de file moet extensie .BIN hebben wat automatisch wordt aangemaakt door het compileerproces van DCP IDE). Eens ingeladen zal de naam van de file die gesimuleerd wordt, samen met zijn pad, getoond worden in de statusbar aan de onderkant van DCP VISIO, samen met de verwante bedieningen "Run", "Stop", "Reset" en "Reload".

Terwijl de simulatie bezig is, dan is het nog mogelijk om waarden, status van inputmodules, virtuele punten, enz. te veranderen. Klik daarvoor simpelweg op het object die veranderd moet worden. Voor digitale punten (inputmodules, virtuele punten, enz.) schakelt de linker muis het in, terwijl de rechter muis het uitschakelt.

De "Stop" knop stopt de simulatie terwijl de "Reset" knop alle parameters herstelt op hun opstartwaarde (dit is gelijkaardig met het opstarten van de DFCP 4).

De "Reload" knop herlaadt de file getoond in het besturingspaneel. Wanneer iets aangepast wordt in een programma moet het opnieuw gecompileerd en herladen worden in de simulator.

11- MODBUS COMMUNICATIEPROTOCOL

11.1- Overzicht

DFCP 4 kan aan de externe wereld gekoppeld worden via het MODBUS RTU protocol. Dat protocol zit in de DFCP 4 en kan ingeschakeld worden via de PROTOCOL code (zie gerelateerde paragraaf). Anders is het protocol FXP XT altijd ingeschakeld. Dat betekent dat:

- DFCP 4 zal antwoorden volgens het MODBUS protocol, als het in staat is, op iedere MODBUS aanvraag.
- DFCP 4 zal antwoorden volgens het FXP XT protocol op iedere FXP XT aanvraag.

De communicatieparameters voor het MODBUS protocol geïmplementeerd in de DFCP 4, zijn de volgende:

- **1 opstart bit**
- **8 data bits**
- **Geen pariteit**
- **1 of 2 stop bits (automatische detectie)**

De Baudrate kan ingesteld worden, zoals eerder beschreven, op de volgende waarden: 2400, 4800, 9600, 19200, 38400, 57600, 115200 Baud. DFCP 4 gedraagt zich altijd als een slaaf (het is een randapparaat van MODBUS). Dat betekent dat het alleen antwoordt op aanvragen van een MASTER MODBUS toestel.

In MODBUS netwerken moet ieder randapparaat zijn eigen adres hebben (normaal genaamd "station address"). Het adres van DFCP 4 moet ingesteld worden door de ADDRESS functie zoals beschreven in de gerelateerde paragraaf. Om de input- en outputpunten, virtuele punten, registers, enz. te lokaliseren, raadpleeg het deel over het externe RAM-geheugen beschreven in een vorig hoofdstuk of nog beter, raadpleeg de tabellen op de volgende pagina's.

11.2- Ondersteunende MODBUS functies

DFCP 4 ondersteunt, in SLAVE modus, de volgende MODBUS functies:

Functiecode	Beschrijving
1	Leest outputtabel
2	Leest inputtabel
3	Leest registers (RAM-geheugen)
4	Leest analoge input
5	Dwingt één digitale outputpunt af
6	Preset één register
16	Preset meerdere registers
17	Rapporteert type toestel

In MASTER modus gebruikt DFCP 4 enkel MODBUS functies 3,4 en 16.

11.3- Tabellen over relatie Words – Parameters

In de volgende tabellen kan het nummer van de MODBUS Word met de gevraagde parameter snel gevonden worden. De volgende tabellen zijn geldig als de code PROTOCOL = (MODBUS) en niet PROTOCOL = (MODBUS-) gebruikt wordt (zie beschrijving van PROTOCOL). Alle nummers in de tabellen staan in een decimaal formaat.

DUEMMEGI voorziet een klein, gratis en nuttig programma genaamd WordFinder. Het geeft dus direct het Modbus Word adres wanneer een gevraagde parameter wordt ingegeven. Dat programma kan gedownload worden van www.duemmegi.it, uit het software gedeelte.

11.3.1- Fysieke inputs

IN	000	010	020	030	040	050	060	070	080	090	100	110	120
000	-	010	020	030	040	050	060	070	080	090	100	110	120
001	001	011	021	031	041	051	061	071	081	091	101	111	121
002	002	012	022	032	042	052	062	072	082	092	102	112	122
003	003	013	023	033	043	053	063	073	083	093	103	113	123
004	004	014	024	034	044	054	064	074	084	094	104	114	124
005	005	015	025	035	045	055	065	075	085	095	105	115	125
006	006	016	026	036	046	056	066	076	086	096	106	116	126
007	007	017	027	037	047	057	067	077	087	097	107	117	127
008	008	018	028	038	048	058	068	078	088	098	108	118	128
009	009	019	029	039	049	059	069	079	089	099	109	119	129

IN	130	140	150	160	170	180	190	200	210	220	230	240	250
000	130	140	150	160	170	180	190	200	210	220	230	240	250
001	131	141	151	161	171	181	191	201	211	221	231	241	251
002	132	142	152	162	172	182	192	202	212	222	232	242	252
003	133	143	153	163	173	183	193	203	213	223	233	243	253
004	134	144	154	164	174	184	194	204	214	224	234	244	254
005	135	145	155	165	175	185	195	205	215	225	235	245	255
006	136	146	156	166	176	186	196	206	216	226	236	246	-
007	137	147	157	167	177	187	197	207	217	227	237	247	-
008	138	148	158	168	178	188	198	208	218	228	238	248	-
009	139	149	159	169	179	189	199	209	219	229	239	249	-

11.3.2- Fysieke outputs

OUT	000	010	020	030	040	050	060	070	080	090	100	110	120
000	-	522	532	542	552	562	572	582	592	602	612	622	632
001	513	523	533	543	553	563	573	583	593	603	613	623	633
002	514	524	534	544	554	564	574	584	594	604	614	624	634
003	515	525	535	545	555	565	575	585	595	605	615	625	635
004	516	526	536	546	556	566	576	586	596	606	616	626	636
005	517	527	537	547	557	567	577	587	597	607	617	627	637
006	518	528	538	548	558	568	578	588	598	608	618	628	638
007	519	529	539	549	559	569	579	589	599	609	619	629	639
008	520	530	540	550	560	570	580	590	600	610	620	630	640
009	521	531	541	551	561	571	581	591	601	611	621	631	641

OUT	130	140	150	160	170	180	190	200	210	220	230	240	250
000	642	652	662	672	682	692	702	712	722	732	742	752	762
001	643	653	663	673	683	693	703	713	723	733	743	753	763
002	644	654	664	674	684	694	704	714	724	734	744	754	764
003	645	655	665	675	685	695	705	715	725	735	745	755	765
004	646	656	666	676	686	696	706	716	726	736	746	756	766
005	647	657	667	677	687	697	707	717	727	737	747	757	767
006	648	658	668	678	688	698	708	718	728	738	748	758	-
007	649	659	669	679	689	699	709	719	729	739	749	759	-
008	650	660	670	680	690	700	710	720	730	740	750	760	-
009	651	661	671	681	691	701	711	721	731	741	751	761	-

11.3.3- Virtuele punten

W/bit	1153	1154	1155	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167	1168
Bit 0	V1	V17	V33	V49	V65	V81	V97	V113	V129	V145	V161	V177	V193	V209	V225	V241
Bit 1	V2	V18	V34	V50	V66	V82	V98	V114	V130	V146	V162	V178	V194	V210	V226	V242
Bit 2	V3	V19	V35	V51	V67	V83	V99	V115	V131	V147	V163	V179	V195	V211	V227	V243
Bit 3	V4	V20	V36	V52	V68	V84	V100	V116	V132	V148	V164	V180	V196	V212	V228	V244
Bit 4	V5	V21	V37	V53	V69	V85	V101	V117	V133	V149	V165	V181	V197	V213	V229	V245
Bit 5	V6	V22	V38	V54	V70	V86	V102	V118	V134	V150	V166	V182	V198	V214	V230	V246
Bit 6	V7	V23	V39	V55	V71	V87	V103	V119	V135	V151	V167	V183	V199	V215	V231	V247
Bit 7	V8	V24	V40	V56	V72	V88	V104	V120	V136	V152	V168	V184	V200	V216	V232	V248
Bit 8	V9	V25	V41	V57	V73	V89	V105	V121	V137	V153	V169	V185	V201	V217	V233	V249
Bit 9	V10	V26	V42	V58	V74	V90	V106	V122	V138	V154	V170	V186	V202	V218	V234	V250
Bit 10	V11	V27	V43	V59	V75	V91	V107	V123	V139	V155	V171	V187	V203	V219	V235	V251
Bit 11	V12	V28	V44	V60	V76	V92	V108	V124	V140	V156	V172	V188	V204	V220	V236	V252
Bit 12	V13	V29	V45	V61	V77	V93	V109	V125	V141	V157	V173	V189	V205	V221	V237	V253
Bit 13	V14	V30	V46	V62	V78	V94	V110	V126	V142	V158	V174	V190	V206	V222	V238	V254
Bit 14	V15	V31	V47	V63	V79	V95	V111	V127	V143	V159	V175	V191	V207	V223	V239	V255
Bit 15	V16	V32	V48	V64	V80	V96	V112	V128	V144	V160	V176	V192	V208	V224	V240	V256

W/bit	1169	1170	1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1183	1184
Bit 0	V257	V273	V289	V305	V321	V337	V353	V369	V385	V401	V417	V433	V449	V465	V481	V497
Bit 1	V258	V274	V290	V306	V322	V338	V354	V370	V386	V402	V418	V434	V450	V466	V482	V498
Bit 2	V259	V275	V291	V307	V323	V339	V355	V371	V387	V403	V419	V435	V451	V467	V483	V499
Bit 3	V260	V276	V292	V308	V324	V340	V356	V372	V388	V404	V420	V436	V452	V468	V484	V500
Bit 4	V261	V277	V293	V309	V325	V341	V357	V373	V389	V405	V421	V437	V453	V469	V485	V501
Bit 5	V262	V278	V294	V310	V326	V342	V358	V374	V390	V406	V422	V438	V454	V470	V486	V502
Bit 6	V263	V279	V295	V311	V327	V343	V359	V375	V391	V407	V423	V439	V455	V471	V487	V503
Bit 7	V264	V280	V296	V312	V328	V344	V360	V376	V392	V408	V424	V440	V456	V472	V488	V504
Bit 8	V265	V281	V297	V313	V329	V345	V361	V377	V393	V409	V425	V441	V457	V473	V489	V505
Bit 9	V266	V282	V298	V314	V330	V346	V362	V378	V394	V410	V426	V442	V458	V474	V490	V506
Bit 10	V267	V283	V299	V315	V331	V347	V363	V379	V395	V411	V427	V443	V459	V475	V491	V507
Bit 11	V268	V284	V300	V316	V332	V348	V364	V380	V396	V412	V428	V444	V460	V476	V492	V508
Bit 12	V269	V285	V301	V317	V333	V349	V365	V381	V397	V413	V429	V445	V461	V477	V493	V509
Bit 13	V270	V286	V302	V318	V334	V350	V366	V382	V398	V414	V430	V446	V462	V478	V494	V510
Bit 14	V271	V287	V303	V319	V335	V351	V367	V383	V399	V415	V431	V447	V463	V479	V495	V511
Bit 15	V272	V288	V304	V320	V336	V352	V368	V384	V400	V416	V432	V448	V464	V480	V496	V512

W/bit	1185	1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199	1200
Bit 0	V513	V529	V545	V561	V577	V593	V609	V625	V641	V657	V673	V689	V705	V721	V737	V753
Bit 1	V514	V530	V546	V562	V578	V594	V610	V626	V642	V658	V674	V690	V706	V722	V738	V754
Bit 2	V515	V531	V547	V563	V579	V595	V611	V627	V643	V659	V675	V691	V707	V723	V739	V755
Bit 3	V516	V532	V548	V564	V580	V596	V612	V628	V644	V660	V676	V692	V708	V724	V740	V756
Bit 4	V517	V533	V549	V565	V581	V597	V613	V629	V645	V661	V677	V693	V709	V725	V741	V757
Bit 5	V518	V534	V550	V566	V582	V598	V614	V630	V646	V662	V678	V694	V710	V726	V742	V758
Bit 6	V519	V535	V551	V567	V583	V599	V615	V631	V647	V663	V679	V695	V711	V727	V743	V759
Bit 7	V520	V536	V552	V568	V584	V600	V616	V632	V648	V664	V680	V696	V712	V728	V744	V760
Bit 8	V521	V537	V553	V569	V585	V601	V617	V633	V649	V665	V681	V697	V713	V729	V745	V761
Bit 9	V522	V538	V554	V570	V586	V602	V618	V634	V650	V666	V682	V698	V714	V730	V746	V762
Bit 10	V523	V539	V555	V571	V587	V603	V619	V635	V651	V667	V683	V699	V715	V731	V747	V763
Bit 11	V524	V540	V556	V572	V588	V604	V620	V636	V652	V668	V684	V700	V716	V732	V748	V764
Bit 12	V525	V541	V557	V573	V589	V605	V621	V637	V653	V669	V685	V701	V717	V733	V749	V765
Bit 13	V526	V542	V558	V574	V590	V606	V622	V638	V654	V670	V686	V702	V718	V734	V750	V766
Bit 14	V527	V543	V559	V575	V591	V607	V623	V639	V655	V671	V687	V703	V719	V735	V751	V767
Bit 15	V528	V544	V560	V576	V592	V608	V624	V640	V656	V672	V688	V704	V720	V736	V752	V768

W/bit	1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215	1216
Bit 0	V769	V785	V801	V817	V833	V849	V865	V881	V897	V913	V929	V945	V961	V977	V993	V1009
Bit 1	V770	V786	V802	V818	V834	V850	V866	V882	V898	V914	V930	V946	V962	V978	V994	V1010
Bit 2	V771	V787	V803	V819	V835	V851	V867	V883	V899	V915	V931	V947	V963	V979	V995	V1011
Bit 3	V772	V788	V804	V820	V836	V852	V868	V884	V900	V916	V932	V948	V964	V980	V996	V1012
Bit 4	V773	V789	V805	V821	V837	V853	V869	V885	V901	V917	V933	V949	V965	V981	V997	V1013
Bit 5	V774	V790	V806	V822	V838	V854	V870	V886	V902	V918	V934	V950	V966	V982	V998	V1014
Bit 6	V775	V791	V807	V823	V839	V855	V871	V887	V903	V919	V935	V951	V967	V983	V999	V1015
Bit 7	V776	V792	V808	V824	V840	V856	V872	V888	V904	V920	V936	V952	V968	V984	V1000	V1016
Bit 8	V777	V793	V809	V825	V841	V857	V873	V889	V905	V921	V937	V953	V969	V985	V1001	V1017
Bit 9	V778	V794	V810	V826	V842	V858	V874	V890	V906	V922	V938	V954	V970	V986	V1002	V1018
Bit 10	V779	V795	V811	V827	V843	V859	V875	V891	V907	V923	V939	V955	V971	V987	V1003	V1019
Bit 11	V780	V796	V812	V828	V844	V860	V876	V892	V908	V924	V940	V956	V972	V988	V1004	V1020
Bit 12	V781	V797	V813	V829	V845	V861	V877	V893	V909	V925	V941	V957	V973	V989	V1005	V1021
Bit 13	V782	V798	V814	V830	V846	V862	V878	V894	V910	V926	V942	V958	V974	V990	V1006	V1022
Bit 14	V783	V799	V815	V831	V847	V863	V879	V895	V911	V927	V943	V959	V975	V991	V1007	V1023
Bit 15	V784	V800	V816	V832	V848	V864	V880	V896	V912	V928	V944	V960	V976	V992	V1008	V1024

W/bit	1217	1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229	1230	1231	1232
Bit 0	V1025	V1041	V1057	V1073	V1089	V1105	V1121	V1137	V1153	V1169	V1185	V1201	V1217	V1233	V1249	V1265
Bit 1	V1026	V1042	V1058	V1074	V1090	V1106	V1122	V1138	V1154	V1170	V1186	V1202	V1218	V1234	V1250	V1266
Bit 2	V1027	V1043	V1059	V1075	V1091	V1107	V1123	V1139	V1155	V1171	V1187	V1203	V1219	V1235	V1251	V1267
Bit 3	V1028	V1044	V1060	V1076	V1092	V1108	V1124	V1140	V1156	V1172	V1188	V1204	V1220	V1236	V1252	V1268
Bit 4	V1029	V1045	V1061	V1077	V1093	V1109	V1125	V1141	V1157	V1173	V1189	V1205	V1221	V1237	V1253	V1269
Bit 5	V1030	V1046	V1062	V1078	V1094	V1110	V1126	V1142	V1158	V1174	V1190	V1206	V1222	V1238	V1254	V1270
Bit 6	V1031	V1047	V1063	V1079	V1095	V1111	V1127	V1143	V1159	V1175	V1191	V1207	V1223	V1239	V1255	V1271
Bit 7	V1032	V1048	V1064	V1080	V1096	V1112	V1128	V1144	V1160	V1176	V1192	V1208	V1224	V1240	V1256	V1272
Bit 8	V1033	V1049	V1065	V1081	V1097	V1113	V1129	V1145	V1161	V1177	V1193	V1209	V1225	V1241	V1257	V1273
Bit 9	V1034	V1050	V1066	V1082	V1098	V1114	V1130	V1146	V1162	V1178	V1194	V1210	V1226	V1242	V1258	V1274
Bit 10	V1035	V1051	V1067	V1083	V1099	V1115	V1131	V1147	V1163	V1179	V1195	V1211	V1227	V1243	V1259	V1275
Bit 11	V1036	V1052	V1068	V1084	V1100	V1116	V1132	V1148	V1164	V1180	V1196	V1212	V1228	V1244	V1260	V1276
Bit 12	V1037	V1053	V1069	V1085	V1101	V1117	V1133	V1149	V1165	V1181	V1197	V1213	V1229	V1245	V1261	V1277
Bit 13	V1038	V1054	V1070	V1086	V1102	V1118	V1134	V1150	V1166	V1182	V1198	V1214	V1230	V1246	V1262	V1278
Bit 14	V1039	V1055	V1071	V1087	V1103	V1119	V1135	V1151	V1167	V1183	V1199	V1215	V1231	V1247	V1263	V1279
Bit 15	V1040	V1056	V1072	V1088	V1104	V1120	V1136	V1152	V1168	V1184	V1200	V1216	V1232	V1248	V1264	V1280

W/bit	1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245	1246	1247	1248
Bit 0	V1281	V1297	V1313	V1329	V1345	V1361	V1377	V1393	V1409	V1425	V1441	V1457	V1473	V1489	V1505	V1521
Bit 1	V1282	V1298	V1314	V1330	V1346	V1362	V1378	V1394	V1410	V1426	V1442	V1458	V1474	V1490	V1506	V1522
Bit 2	V1283	V1299	V1315	V1331	V1347	V1363	V1379	V1395	V1411	V1427	V1443	V1459	V1475	V1491	V1507	V1523
Bit 3	V1284	V1300	V1316	V1332	V1348	V1364	V1380	V1396	V1412	V1428	V1444	V1460	V1476	V1492	V1508	V1524
Bit 4	V1285	V1301	V1317	V1333	V1349	V1365	V1381	V1397	V1413	V1429	V1445	V1461	V1477	V1493	V1509	V1525
Bit 5	V1286	V1302	V1318	V1334	V1350	V1366	V1382	V1398	V1414	V1430	V1446	V1462	V1478	V1494	V1510	V1526
Bit 6	V1287	V1303	V1319	V1335	V1351	V1367	V1383	V1399	V1415	V1431	V1447	V1463	V1479	V1495	V1511	V1527
Bit 7	V1288	V1304	V1320	V1336	V1352	V1368	V1384	V1400	V1416	V1432	V1448	V1464	V1480	V1496	V1512	V1528
Bit 8	V1289	V1305	V1321	V1337	V1353	V1369	V1385	V1401	V1417	V1433	V1449	V1465	V1481	V1497	V1513	V1529
Bit 9	V1290	V1306	V1322	V1338	V1354	V1370	V1386	V1402	V1418	V1434	V1450	V1466	V1482	V1498	V1514	V1530
Bit 10	V1291	V1307	V1323	V1339	V1355	V1371	V1387	V1403	V1419	V1435	V1451	V1467	V1483	V1499	V1515	V1531
Bit 11	V1292	V1308	V1324	V1340	V1356	V1372	V1388	V1404	V1420	V1436	V1452	V1468	V1484	V1500	V1516	V1532
Bit 12	V1293	V1309	V1325	V1341	V1357	V1373	V1389	V1405	V1421	V1437	V1453	V1469	V1485	V1501	V1517	V1533
Bit 13	V1294	V1310	V1326	V1342	V1358	V1374	V1390	V1406	V1422	V1438	V1454	V1470	V1486	V1502	V1518	V1534
Bit 14	V1295	V1311	V1327	V1343	V1359	V1375	V1391	V1407	V1423	V1439	V1455	V1471	V1487	V1503	V1519	V1535
Bit 15	V1296	V1312	V1328	V1344	V1360	V1376	V1392	V1408	V1424	V1440	V1456	V1472	V1488	V1504	V1520	V1536

W/bit	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260	1261	1262	1263	1264
Bit 0	V1537	V1553	V1569	V1585	V1601	V1617	V1633	V1649	V1665	V1681	V1697	V1713	V1729	V1745	V1761	V1777
Bit 1	V1538	V1554	V1570	V1586	V1602	V1618	V1634	V1650	V1666	V1682	V1698	V1714	V1730	V1746	V1762	V1778
Bit 2	V1539	V1555	V1571	V1587	V1603	V1619	V1635	V1651	V1667	V1683	V1699	V1715	V1731	V1747	V1763	V1779
Bit 3	V1540	V1556	V1572	V1588	V1604	V1620	V1636	V1652	V1668	V1684	V1700	V1716	V1732	V1748	V1764	V1780
Bit 4	V1541	V1557	V1573	V1589	V1605	V1621	V1637	V1653	V1669	V1685	V1701	V1717	V1733	V1749	V1765	V1781
Bit 5	V1542	V1558	V1574	V1590	V1606	V1622	V1638	V1654	V1670	V1686	V1702	V1718	V1734	V1750	V1766	V1782
Bit 6	V1543	V1559	V1575	V1591	V1607	V1623	V1639	V1655	V1671	V1687	V1703	V1719	V1735	V1751	V1767	V1783
Bit 7	V1544	V1560	V1576	V1592	V1608	V1624	V1640	V1656	V1672	V1688	V1704	V1720	V1736	V1752	V1768	V1784
Bit 8	V1545	V1561	V1577	V1593	V1609	V1625	V1641	V1657	V1673	V1689	V1705	V1721	V1737	V1753	V1769	V1785
Bit 9	V1546	V1562	V1578	V1594	V1610	V1626	V1642	V1658	V1674	V1690	V1706	V1722	V1738	V1754	V1770	V1786
Bit 10	V1547	V1563	V1579	V1595	V1611	V1627	V1643	V1659	V1675	V1691	V1707	V1723	V1739	V1755	V1771	V1787
Bit 11	V1548	V1564	V1580	V1596	V1612	V1628	V1644	V1660	V1676	V1692	V1708	V1724	V1740	V1756	V1772	V1788
Bit 12	V1549	V1565	V1581	V1597	V1613	V1629	V1645	V1661	V1677	V1693	V1709	V1725	V1741	V1757	V1773	V1789
Bit 13	V1550	V1566	V1582	V1598	V1614	V1630	V1646	V1662	V1678	V1694	V1710	V1726	V1742	V1758	V1774	V1790
Bit 14	V1551	V1567	V1583	V1599	V1615	V1631	V1647	V1663	V1679	V1695	V1711	V1727	V1743	V1759	V1775	V1791
Bit 15	V1552	V1568	V1584	V1600	V1616	V1632	V1648	V1664	V1680	V1696	V1712	V1728	V1744	V1760	V1776	V1792

W/bit	1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275	1276	1277	1278	1279
Bit 0	V1793	V1809	V1825	V1841	V1857	V1873	V1889	V1905	V1921	V1937	V1953	V1969	V1985	V2001	V2017
Bit 1	V1794	V1810	V1826	V1842	V1858	V1874	V1890	V1906	V1922	V1938	V1954	V1970	V1986	V2002	V2018
Bit 2	V1795	V1811	V1827	V1843	V1859	V1875	V1891	V1907	V1923	V1939	V1955	V1971	V1987	V2003	V2019
Bit 3	V1796	V1812	V1828	V1844	V1860	V1876	V1892	V1908	V1924	V1940	V1956	V1972	V1988	V2004	V2020
Bit 4	V1797	V1813	V1829	V1845	V1861	V1877	V1893	V1909	V1925	V1941	V1957	V1973	V1989	V2005	V2021
Bit 5	V1798	V1814	V1830	V1846	V1862	V1878	V1894	V1910	V1926	V1942	V1958	V1974	V1990	V2006	V2022
Bit 6	V1799	V1815	V1831	V1847	V1863	V1879	V1895	V1911	V1927	V1943	V1959	V1975	V1991	V2007	V2023
Bit 7	V1800	V1816	V1832	V1848	V1864	V1880	V1896	V1912	V1928	V1944	V1960	V1976	V1992	V2008	V2024
Bit 8	V1801	V1817	V1833	V1849	V1865	V1881	V1897	V1913	V1929	V1945	V1961	V1977	V1993	V2009	V2025
Bit 9	V1802	V1818	V1834	V1850	V1866	V1882	V1898	V1914	V1930	V1946	V1962	V1978	V1994	V2010	V2026
Bit 10	V1803	V1819	V1835	V1851	V1867	V1883	V1899	V1915	V1931	V1947	V1963	V1979	V1995	V2011	V2027
Bit 11	V1804	V1820	V1836	V1852	V1868	V1884	V1900	V1916	V1932	V1948	V1964	V1980	V1996	V2012	V2028
Bit 12	V1805	V1821	V1837	V1853	V1869	V1885	V1901	V1917	V1933	V1949	V1965	V1981	V1997	V2013	V2029
Bit 13	V1806	V1822	V1838	V1854	V1870	V1886	V1902	V1918	V1934	V1950	V1966	V1982	V1998	V2014	V2030
Bit 14	V1807	V1823	V1839	V1855	V1871	V1887	V1903	V1919	V1935	V1951	V1967	V1983	V1999	V2015	V2031
Bit 15	V1808	V1824	V1840	V1856	V1872	V1888	V1904	V1920	V1936	V1952	V1968	V1984	V2000	V2016	V2032

11.3.4- Registers

R	000	010	020	030	040	050	060	070	080	090	100	110	120	130	140	150
000	2048	2058	2068	2078	2088	2098	2108	2118	2128	2138	2148	2158	2168	2178	2188	2198
001	2049	2059	2069	2079	2089	2099	2109	2119	2129	2139	2149	2159	2169	2179	2189	2199
002	2050	2060	2070	2080	2090	2100	2110	2120	2130	2140	2150	2160	2170	2180	2190	2200
003	2051	2061	2071	2081	2091	2101	2111	2121	2131	2141	2151	2161	2171	2181	2191	2201
004	2052	2062	2072	2082	2092	2102	2112	2122	2132	2142	2152	2162	2172	2182	2192	2202
005	2053	2063	2073	2083	2093	2103	2113	2123	2133	2143	2153	2163	2173	2183	2193	2203
006	2054	2064	2074	2084	2094	2104	2114	2124	2134	2144	2154	2164	2174	2184	2194	2204
007	2055	2065	2075	2085	2095	2105	2115	2125	2135	2145	2155	2165	2175	2185	2195	2205
008	2056	2066	2076	2086	2096	2106	2116	2126	2136	2146	2156	2166	2176	2186	2196	2206
009	2057	2067	2077	2087	2097	2107	2117	2127	2137	2147	2157	2167	2177	2187	2197	2207

R	160	170	180	190	200	210	220	230	240	250	260	270	280	290	300	310
000	2208	2218	2228	2238	2248	2258	2268	2278	2288	2298	2308	2318	2328	2338	2348	2358
001	2209	2219	2229	2239	2249	2259	2269	2279	2289	2299	2309	2319	2329	2339	2349	2359
002	2210	2220	2230	2240	2250	2260	2270	2280	2290	2300	2310	2320	2330	2340	2350	2360
003	2211	2221	2231	2241	2251	2261	2271	2281	2291	2301	2311	2321	2331	2341	2351	2361
004	2212	2222	2232	2242	2252	2262	2272	2282	2292	2302	2312	2322	2332	2342	2352	2362
005	2213	2223	2233	2243	2253	2263	2273	2283	2293	2303	2313	2323	2333	2343	2353	2363
006	2214	2224	2234	2244	2254	2264	2274	2284	2294	2304	2314	2324	2334	2344	2354	2364
007	2215	2225	2235	2245	2255	2265	2275	2285	2295	2305	2315	2325	2335	2345	2355	2365
008	2216	2226	2236	2246	2256	2266	2276	2286	2296	2306	2316	2326	2336	2346	2356	2366
009	2217	2227	2237	2247	2257	2267	2277	2287	2297	2307	2317	2327	2337	2347	2357	2367

R	320	330	340	350	360	370	380	390	400	410	420	430	440	450	460	470
000	2368	2378	2388	2398	2408	2418	2428	2438	2448	2458	2468	2478	2488	2498	2508	2518
001	2369	2379	2389	2399	2409	2419	2429	2439	2449	2459	2469	2479	2489	2499	2509	2519
002	2370	2380	2390	2400	2410	2420	2430	2440	2450	2460	2470	2480	2490	2500	2510	2520
003	2371	2381	2391	2401	2411	2421	2431	2441	2451	2461	2471	2481	2491	2501	2511	2521
004	2372	2382	2392	2402	2412	2422	2432	2442	2452	2462	2472	2482	2492	2502	2512	2522
005	2373	2383	2393	2403	2413	2423	2433	2443	2453	2463	2473	2483	2493	2503	2513	2523
006	2374	2384	2394	2404	2414	2424	2434	2444	2454	2464	2474	2484	2494	2504	2514	2524
007	2375	2385	2395	2405	2415	2425	2435	2445	2455	2465	2475	2485	2495	2505	2515	2525
008	2376	2386	2396	2406	2416	2426	2436	2446	2456	2466	2476	2486	2496	2506	2516	2526
009	2377	2387	2397	2407	2417	2427	2437	2447	2457	2467	2477	2487	2497	2507	2517	2527

R	480	490	500	510	520	530	540	550	560	570	580	590	600	610	620	630
000	2528	2538	2548	2558	2568	2578	2588	2598	2608	2618	2628	2638	2648	2658	2668	2678
001	2529	2539	2549	2559	2569	2579	2589	2599	2609	2619	2629	2639	2649	2659	2669	2679
002	2530	2540	2550	2560	2570	2580	2590	2600	2610	2620	2630	2640	2650	2660	2670	2680
003	2531	2541	2551	2561	2571	2581	2591	2601	2611	2621	2631	2641	2651	2661	2671	2681
004	2532	2542	2552	2562	2572	2582	2592	2602	2612	2622	2632	2642	2652	2662	2672	2682
005	2533	2543	2553	2563	2573	2583	2593	2603	2613	2623	2633	2643	2653	2663	2673	2683
006	2534	2544	2554	2564	2574	2584	2594	2604	2614	2624	2634	2644	2654	2664	2674	2684
007	2535	2545	2555	2565	2575	2585	2595	2605	2615	2625	2635	2645	2655	2665	2675	2685
008	2536	2546	2556	2566	2576	2586	2596	2606	2616	2626	2636	2646	2656	2666	2676	2686
009	2537	2547	2557	2567	2577	2587	2597	2607	2617	2627	2637	2647	2657	2667	2677	2687

R	640	650	660	670	680	690	700	710	720	730	740	750	760	770	780	790
000	2688	2698	2708	2718	2728	2738	2748	2758	2768	2778	2788	2798	2808	2818	2828	2838
001	2689	2699	2709	2719	2729	2739	2749	2759	2769	2779	2789	2799	2809	2819	2829	2839
002	2690	2700	2710	2720	2730	2740	2750	2760	2770	2780	2790	2800	2810	2820	2830	2840
003	2691	2701	2711	2721	2731	2741	2751	2761	2771	2781	2791	2801	2811	2821	2831	2841
004	2692	2702	2712	2722	2732	2742	2752	2762	2772	2782	2792	2802	2812	2822	2832	2842
005	2693	2703	2713	2723	2733	2743	2753	2763	2773	2783	2793	2803	2813	2823	2833	2843
006	2694	2704	2714	2724	2734	2744	2754	2764	2774	2784	2794	2804	2814	2824	2834	2844
007	2695	2705	2715	2725	2735	2745	2755	2765	2775	2785	2795	2805	2815	2825	2835	2845
008	2696	2706	2716	2726	2736	2746	2756	2766	2776	2786	2796	2806	2816	2826	2836	2846
009	2697	2707	2717	2727	2737	2747	2757	2767	2777	2787	2797	2807	2817	2827	2837	2847

R	800	810	820	830	840	850	860	870	880	890	900	910	920	930	940	950
000	2848	2858	2868	2878	2888	2898	2908	2918	2928	2938	2948	2958	2968	2978	2988	2998
001	2849	2859	2869	2879	2889	2899	2909	2919	2929	2939	2949	2959	2969	2979	2989	2999
002	2850	2860	2870	2880	2890	2900	2910	2920	2930	2940	2950	2960	2970	2980	2990	3000
003	2851	2861	2871	2881	2891	2901	2911	2921	2931	2941	2951	2961	2971	2981	2991	3001
004	2852	2862	2872	2882	2892	2902	2912	2922	2932	2942	2952	2962	2972	2982	2992	3002
005	2853	2863	2873	2883	2893	2903	2913	2923	2933	2943	2953	2963	2973	2983	2993	3003
006	2854	2864	2874	2884	2894	2904	2914	2924	2934	2944	2954	2964	2974	2984	2994	3004
007	2855	2865	2875	2885	2895	2905	2915	2925	2935	2945	2955	2965	2975	2985	2995	3005
008	2856	2866	2876	2886	2896	2906	2916	2926	2936	2946	2956	2966	2976	2986	2996	3006
009	2857	2867	2877	2887	2897	2907	2917	2927	2937	2947	2957	2967	2977	2987	2997	3007

R	960	970	980	990	1000	1010	1020
000	3008	3018	3028	3038	3048	3058	3068
001	3009	3019	3029	3039	3049	3059	3069
002	3010	3020	3030	3040	3050	3060	3070
003	3011	3021	3031	3041	3051	3061	3071
004	3012	3022	3032	3042	3052	3062	-
005	3013	3023	3033	3043	3053	3063	-
006	3014	3024	3034	3044	3054	3064	-
007	3015	3025	3035	3045	3055	3065	-
008	3016	3026	3036	3046	3056	3066	-
009	3017	3027	3037	3047	3057	3067	-

11.3.5- Counters

C	000	010	020	030	040	050	060	070	080	090	100	110	120	130	140	150
000	3072	3082	3092	3102	3112	3122	3132	3142	3152	3162	3172	3182	3192	3202	3212	3222
001	3073	3083	3093	3103	3113	3123	3133	3143	3153	3163	3173	3183	3193	3203	3213	3223
002	3074	3084	3094	3104	3114	3124	3134	3144	3154	3164	3174	3184	3194	3204	3214	3224
003	3075	3085	3095	3105	3115	3125	3135	3145	3155	3165	3175	3185	3195	3205	3215	3225
004	3076	3086	3096	3106	3116	3126	3136	3146	3156	3166	3176	3186	3196	3206	3216	3226
005	3077	3087	3097	3107	3117	3127	3137	3147	3157	3167	3177	3187	3197	3207	3217	3227
006	3078	3088	3098	3108	3118	3128	3138	3148	3158	3168	3178	3188	3198	3208	3218	3228
007	3079	3089	3099	3109	3119	3129	3139	3149	3159	3169	3179	3189	3199	3209	3219	3229
008	3080	3090	3100	3110	3120	3130	3140	3150	3160	3170	3180	3190	3200	3210	3220	3230
009	3081	3091	3101	3111	3121	3131	3141	3151	3161	3171	3181	3191	3201	3211	3221	3231

C	160	170	180	190	200	210	220	230	240	250	260	270	280	290	300	310
000	3232	3242	3252	3262	3272	3282	3292	3302	3312	3322	3332	3342	3352	3362	3372	3382
001	3233	3243	3253	3263	3273	3283	3293	3303	3313	3323	3333	3343	3353	3363	3373	3383
002	3234	3244	3254	3264	3274	3284	3294	3304	3314	3324	3334	3344	3354	3364	3374	3384
003	3235	3245	3255	3265	3275	3285	3295	3305	3315	3325	3335	3345	3355	3365	3375	3385
004	3236	3246	3256	3266	3276	3286	3296	3306	3316	3326	3336	3346	3356	3366	3376	3386
005	3237	3247	3257	3267	3277	3287	3297	3307	3317	3327	3337	3347	3357	3367	3377	3387
006	3238	3248	3258	3268	3278	3288	3298	3308	3318	3328	3338	3348	3358	3368	3378	3388
007	3239	3249	3259	3269	3279	3289	3299	3309	3319	3329	3339	3349	3359	3369	3379	3389
008	3240	3250	3260	3270	3280	3290	3300	3310	3320	3330	3340	3350	3360	3370	3380	3390
009	3241	3251	3261	3271	3281	3291	3301	3311	3321	3331	3341	3351	3361	3371	3381	3391

C	320	330	340	350	360	370	380	390	400	410	420	430	440	450	460	470
000	3392	3402	3412	3422	3432	3442	3452	3462	3472	3482	3492	3502	3512	3522	3532	3542
001	3393	3403	3413	3423	3433	3443	3453	3463	3473	3483	3493	3503	3513	3523	3533	3543
002	3394	3404	3414	3424	3434	3444	3454	3464	3474	3484	3494	3504	3514	3524	3534	3544
003	3395	3405	3415	3425	3435	3445	3455	3465	3475	3485	3495	3505	3515	3525	3535	3545
004	3396	3406	3416	3426	3436	3446	3456	3466	3476	3486	3496	3506	3516	3526	3536	3546
005	3397	3407	3417	3427	3437	3447	3457	3467	3477	3487	3497	3507	3517	3527	3537	3547
006	3398	3408	3418	3428	3438	3448	3458	3468	3478	3488	3498	3508	3518	3528	3538	3548
007	3399	3409	3419	3429	3439	3449	3459	3469	3479	3489	3499	3509	3519	3529	3539	3549
008	3400	3410	3420	3430	3440	3450	3460	3470	3480	3490	3500	3510	3520	3530	3540	3550
009	3401	3411	3421	3431	3441	3451	3461	3471	3481	3491	3501	3511	3521	3531	3541	3551

C	480	490	500	510	520	530	540	550	560	570	580	590	600	610	620	630
000	3552	3562	3572	3582	3592	3602	3612	3622	3632	3642	3652	3662	3672	3682	3692	3702
001	3553	3563	3573	3583	3593	3603	3613	3623	3633	3643	3653	3663	3673	3683	3693	3703
002	3554	3564	3574	3584	3594	3604	3614	3624	3634	3644	3654	3664	3674	3684	3694	3704
003	3555	3565	3575	3585	3595	3605	3615	3625	3635	3645	3655	3665	3675	3685	3695	3705
004	3556	3566	3576	3586	3596	3606	3616	3626	3636	3646	3656	3666	3676	3686	3696	3706
005	3557	3567	3577	3587	3597	3607	3617	3627	3637	3647	3657	3667	3677	3687	3697	3707
006	3558	3568	3578	3588	3598	3608	3618	3628	3638	3648	3658	3668	3678	3688	3698	3708
007	3559	3569	3579	3589	3599	3609	3619	3629	3639	3649	3659	3669	3679	3689	3699	3709
008	3560	3570	3580	3590	3600	3610	3620	3630	3640	3650	3660	3670	3680	3690	3700	3710
009	3561	3571	3581	3591	3601	3611	3621	3631	3641	3651	3661	3671	3681	3691	3701	3711

C	640	650	660	670	680	690	700	710	720	730	740	750	760	770	780	790
000	3712	3722	3732	3742	3752	3762	3772	3782	3792	3802	3812	3822	3832	3842	3852	3862
001	3713	3723	3733	3743	3753	3763	3773	3783	3793	3803	3813	3823	3833	3843	3853	3863
002	3714	3724	3734	3744	3754	3764	3774	3784	3794	3804	3814	3824	3834	3844	3854	3864
003	3715	3725	3735	3745	3755	3765	3775	3785	3795	3805	3815	3825	3835	3845	3855	3865
004	3716	3726	3736	3746	3756	3766	3776	3786	3796	3806	3816	3826	3836	3846	3856	3866
005	3717	3727	3737	3747	3757	3767	3777	3787	3797	3807	3817	3827	3837	3847	3857	3867
006	3718	3728	3738	3748	3758	3768	3778	3788	3798	3808	3818	3828	3838	3848	3858	3868
007	3719	3729	3739	3749	3759	3769	3779	3789	3799	3809	3819	3829	3839	3849	3859	3869
008	3720	3730	3740	3750	3760	3770	3780	3790	3800	3810	3820	3830	3840	3850	3860	3870
009	3721	3731	3741	3751	3761	3771	3781	3791	3801	3811	3821	3831	3841	3851	3861	3871

C	800	810	820	830	840	850	860	870	880	890	900	910	920	930	940	950
000	3872	3882	3892	3902	3912	3922	3932	3942	3952	3962	3972	3982	3992	4002	4012	4022
001	3873	3883	3893	3903	3913	3923	3933	3943	3953	3963	3973	3983	3993	4003	4013	4023
002	3874	3884	3894	3904	3914	3924	3934	3944	3954	3964	3974	3984	3994	4004	4014	4024
003	3875	3885	3895	3905	3915	3925	3935	3945	3955	3965	3975	3985	3995	4005	4015	4025
004	3876	3886	3896	3906	3916	3926	3936	3946	3956	3966	3976	3986	3996	4006	4016	4026
005	3877	3887	3897	3907	3917	3927	3937	3947	3957	3967	3977	3987	3997	4007	4017	4027
006	3878	3888	3898	3908	3918	3928	3938	3948	3958	3968	3978	3988	3998	4008	4018	4028
007	3879	3889	3899	3909	3919	3929	3939	3949	3959	3969	3979	3989	3999	4009	4019	4029
008	3880	3890	3900	3910	3920	3930	3940	3950	3960	3970	3980	3990	4000	4010	4020	4030
009	3881	3891	3901	3911	3921	3931	3941	3951	3961	3971	3981	3991	4001	4011	4021	4031

C	960	970	980	990	1000	1010	1020
000	4032	4042	4052	4062	4072	4082	4092
001	4033	4043	4053	4063	4073	4083	4093
002	4034	4044	4054	4064	4074	4084	4094
003	4035	4045	4055	4065	4075	4085	4095
004	4036	4046	4056	4066	4076	4086	-
005	4037	4047	4057	4067	4077	4087	-
006	4038	4048	4058	4068	4078	4088	-
007	4039	4049	4059	4069	4079	4089	-
008	4040	4050	4060	4070	4080	4090	-
009	4041	4051	4061	4071	4081	4091	-