```
// --------------------------------------------------------------------------
// SIOC - System for IOCARDS                    Script by RHO Standard Protocol
// --------------------------------------------------------------------------
// Designer      : René Hoep
// FileName      : RDS_SIOC_COMMANDO_ALL.txt
//
// Version       : 2021.01
//
// Last Update   : 20210615   1845
//
//
// Summery       : Uitleg SIOC Commands
//                 Uitleg SIOC Links
//
// Explanation   : SIOC = System for IOCARDS

//
// --------------------------------------------------------------------------

// --------------------------------------------------------------------------
FSUIPC Private Offset
66C1 - 66FF
Bv RDSOffSet66D1
// --------------------------------------------------------------------------

// --------------------------------------------------------------------------
// STILL IN PROGRESS             STILL IN PROGRESS             STILL IN PROGRESS
// --------------------------------------------------------------------------

INDEX     - COMMANDO's
INDEX     - LINK's
VOORBEELD - OFFSET CYCLUS
VOORBEELD - INPUT TYPE P



// --------------------------------------------------------------------------
// INDEX COMMANDOS                                          INDEX COMMANDOS
// --------------------------------------------------------------------------

-
*
/
+
<
<=
<>
=
>
>=
ABS
AND
ARCT
C0
C1
C2
CALL
CHANGEBIT
CHANGEBITN
CLEARBIT
COS
DELAY
DIV
ELSE
```

```
FROMBCD
IF and ELSE
L0
L1
L2
LIMIT
LOGN
MOD
NOT
OR
POWER
RANDOM
ROTATE
ROUND
SETBIT
SETSOUND
SIN
TESTBIT
TIMER
TOBCD
TOGGLE
TRUNC

// --------------------------------------------------------------------------



// ------------------------------------------------
// OPMERKING
// ------------------------------------------------
// Je kunt niet direct rekenen met 2 variabelen.
// Daartoe zijn de volgende hulpmiddelen beschikbaar
//
// L0 [ L nul ] L1  en L2   voor getal waarden
// C0 [ C nul ] C1  en C2   voor true of not true waarden
// ------------------------------------------------



// ------------------------------------------------
// VOORBEELD :  //               commentaar toevoegen
// ------------------------------------------------
v1000 = 9  // waarde invoer

Var 1000, Static, Name Test_Invoer // Deze variabele gebruik ik voor de tekst



// ------------------------------------------------
// VOORBEELD :  -                            aftrekken
// ------------------------------------------------

v1000 = 9


Var 1000, Static, Name Test_Invoer
{
L0 = v1000 - 3
v1001 = L0
}

Var 1001, Static, Name Test_Uitkomst

// -------------------------
```

```
// UITKOMST
// -------------------------
L0 = 9 - 3
v1001 = 6



// ------------------------------------------------
// VOORBEELD :  *                    vermenigvuldigen
// ------------------------------------------------

v1000 = 9


Var 1000, Static, Name Test_Invoer
{
L0 = v1000 * 3
v1001 = L0
}

Var 1001, Static, Name Test_Uitkomst

// -------------------------
// UITKOMST
// -------------------------
L0 = 9 * 3
v1001 = 27



// ------------------------------------------------
// VOORBEELD :  /                               delen
// ------------------------------------------------

v1000 = 9


Var 1000, Static, Name Test_Invoer
{
L0 = v1000 / 3
v1001 = L0
}

Var 1001, Static, Name Test_Uitkomst

// -------------------------
// UITKOMST
// -------------------------
L0 = 9 / 3
v1001 = 3



// ------------------------------------------------
// VOORBEELD :  +                             optellen
// ------------------------------------------------

v1000 = 9


Var 1000, Static, Name Test_Invoer
{
L0 = v1000 + 3
v1001 = L0
```

```
}

Var 1001, Static, Name Test_Uitkomst

// -------------------------
// UITKOMST
// -------------------------
L0 = 9 + 3
v1001 = 12



// ------------------------------------------------
// VOORBEELD :  <                          kleiner dan
// ------------------------------------------------

v1000 = 9


Var 1000, Static, Name Test_Invoer
{
IF v1000 < 10
   {
   v1001 = 1
   }
}

Var 1001, Static, Name Test_Uitkomst

// -------------------------
// UITKOMST
// -------------------------
Als 9 < 10 [ en dat is hier dus waar ]
v1001 = 1



// ------------------------------------------------
// VOORBEELD :  <=          kleiner dan of gelijk aan
// ------------------------------------------------

v1000 = 10


Var 1000, Static, Name Test_Invoer
{
IF v1000 <= 10
   {
   v1001 = 1
   }
}

Var 1001, Static, Name Test_Uitkomst

// -------------------------
// UITKOMST
// -------------------------
Als 10 <= 10 [ en dat is hier dus waar ]
v1001 = 1



// ------------------------------------------------
// VOORBEELD :  =                          gelijk aan
```

```
// -------------------------------------------------

v1000 = 10


Var 1000, Static, Name Test_Invoer
{
IF v1000 = 10
   {
   v1001 = 1
   }
}

Var 1001, Static, Name Test_Uitkomst

// -------------------------
// UITKOMST
// -------------------------
Als 10 = 10 [ en dat is hier dus waar ]
v1001 = 1



// -------------------------------------------------
// VOORBEELD :  >                          groter dan
// -------------------------------------------------

v1000 = 12


Var 1000, Static, Name Test_Invoer
{
IF v1000 > 10
   {
   v1001 = 1
   }
}

Var 1001, Static, Name Test_Uitkomst

// -------------------------
// UITKOMST
// -------------------------
Als 12 > 10 [ en dat is hier dus waar ]
v1001 = 1



// -------------------------------------------------
// VOORBEELD :  >=           groter dan of gelijk aan
// -------------------------------------------------

v1000 = 10


Var 1000, Static, Name Test_Invoer
{
IF v1000 > 10
   {
   v1001 = 1
   }
}

Var 1001, Static, Name Test_Uitkomst
```

```
// -------------------------
// UITKOMST
// -------------------------
Als 10 >= 10 [ en dat is hier dus waar ]
v1001 = 1



// ---------------------------------------------------
// VOORBEELD : ABS    absoluut maakt van - een + getal
// ---------------------------------------------------

v1000 = 10


Var 1000, Static, Name Test_Invoer
{
L0 = v1000 - 22
v1001 = ABS L0
}

Var 1001, Static, Name Test_Uitkomst

// -------------------------
// UITKOMST
// -------------------------
L0 = 10 - 22
v1001 = ABS -12
v1001 = 12



// ---------------------------------------------------
// VOORBEELD :  AND        vergelijkt waar variabelen
// ---------------------------------------------------

v0999 = waar
v1000 = niet waar

Var 1000, Static, Name Test_Invoer
{
C0 = v0999
C1 = v1000

IF C0 AND C1
   {
   v1001 = 0
   }
ELSE
   {
   v1001 = 1
   }

Var 1001, Static, Name Test_Uitkomst

// -------------------------
// UITKOMST
// -------------------------
C0 = waar
C1 = niet waar

Als 0 AND 0 // Als C0 en C1 gelijk zijn aan elkaar
               is hier dus niet het geval, dus v1001 = 1
```

```
// -------------------------------------------------
// VOORBEELD :  C0
// VOORBEELD :  C1
// VOORBEELD :  C2
// -------------------------------------------------

// C0 [ C nul ] C1  en C2   voor true of not true waarden



// -------------------------------------------------
// VOORBEELD :  CALL             aanroepen subroutine
// -------------------------------------------------

v1000 = 1


Var 1000, Static, Name Test_Invoer
{
IF v1000 = 1
   {
   v1001 = 1
   }
}

Var 1001, Static, Name Test_Uitkomst
{
CALL v9001  // Ga naar de Subroutine v9001 voor extra berekeningen
{


Var 9001, Static, Name Sub_Voorbeeld, Link SUBRUTINE  // De naam is in deze taal echt
SUBRUTINE zonder de O
{
// subroutine berekeningen
}

// -------------------------
// UITKOMST
// -------------------------
// Een Subroutine gebruik je als je vanaf diverse variabelen
// steeds dezelfde berekening moet uitvoeren.



// -------------------------------------------------
// VOORBEELD : CHANGEBIT            wijzig BIT waarde
// -------------------------------------------------

// [ SWITCH ] ON-OFF
Var 1000, Static, Name Test_InvoerSW, Link IOCARD_SW, Device 0, Input 10, Type I
{
v1001 = CHANGEBIT 4, v1000  // [ Als BIT = 1 wordt deze 0 en VISA VERSA ]
}


Var 1001, Static, Name Test_Uitkomst_, Link FSUIPC_OUT, Offset $56E2, Length 1



// -------------------------------------------------
// VOORBEELD : CHANGEBITN wijzig BIT waarde omgekeerd
```

```
// -------------------------------------------------

// [ SWITCH ] ON-OFF
Var 1000, Static, Name Test_InvoerSW, Link IOCARD_SW, Device 0, Input 10, Type I
{
v1001 = CHANGEBITN 4, v1000 // [ Als Changebit maar nu net anders om ]
}

// ----- KUN JE GEBRUIKEN ALS DE SCHAKELAAR NET VERKEERD OM REAGEERD -----


Var 1001, Static, Name Test_Uitkomst_, Link FSUIPC_OUT, Offset $56E2, Length 1



// -------------------------------------------------
// VOORBEELD :  CLEARBIT       schoon BIT waarde in 0
// -------------------------------------------------

// [ SWITCH ] ON-OFF
Var 1000, Static, Name Test_InvoerSW, Link IOCARD_SW, Device 0, Input 10, Type I
{
IF v1000 = 1            // schakelaar aan
   {
   v1001 = CLEARBIT 4   // [ BIT 4 wordt op 0 [nul] gezet ]
   }
ELSE                    // schakelaar uit
   {
   v1001 = SETBIT 4     // [ BIT 4 wordt op 1 gezet ]
   }

Var 1001, Static, Name Test_Uitkomst_, Link FSUIPC_OUT, Offset $56E2, Length 1




// -------------------------------------------------
// VOORBEELD :  COS                          co-sinus
// -------------------------------------------------



// -------------------------------------------------
// VOORBEELD :  DELAY                        vertraag
// -------------------------------------------------

v1000 = 1


Var 1000, Static, Name Test_Invoer
{
IF v1000 = 1
   {
   v1002 = DELAY 1 400 // wacht 400/100 = 4 sec voor de LED aangaat
   }
ELSE
   {
   v1002 = 0 // Zet LED uit
   L0 = 5
   v1003= DELAY L0 300
   }
}

Var 1002, Static, Name Test_LED, Link IOCARD_OUT, Device 0, Output 1
```

```
var 1003, Static, Name Test_Sound

// -------------------------
// UITKOMST
// -------------------------

DELAY para1 Para2
Para1 = de waarde die moet worden uitgevoerd
Para2 = Tijd in 1/100 sec dat er moet worden gewacht 1 sec is dus 100

L0 = 5
v1003= DELAY L0 300 // Hier wordt 3 sec gewacht voordat de waarde 5 wordt
                       aangeboden aan de SOUND variabele

v1002 = 1           // Zet LED aan
v1002 = DELAY 0 400 // 4 sec nabranden voordat LED uit gaat



// ------------------------------------------------------------
// VOORBEELD :  DIV  deelt met een uitkomst van een geheel getal
// ------------------------------------------------------------

v1000 = 10


Var 1000, Static, Name Test_Invoer
{
L0 = DIV v1000 3
v1001 = L0
}

Var 1001, Static, Name Test_Uitkomst

// -------------------------
// UITKOMST
// -------------------------
L0 = DIV 10 / 3   = 3,3333333
v1001 = 3



// ------------------------------------------------
// VOORBEELD :  ELSE                          anders
// ------------------------------------------------


v1000 = 1


Var 1000, Static, Name Test_Invoer
{
IF v1000 = 1
   {
   v1002 = DELAY 1 400 // wacht 400/100 = 4 sec voor de LED aangaat
   }
ELSE
   {
   v1002 = 0 // Zet LED uit
   }
}
```

```
// -------------------------------------------------
// VOORBEELD :  FROMBCD
// -------------------------------------------------



// -------------------------------------------------
// VOORBEELD :  IF                                als
// -------------------------------------------------

v1000 = 9


Var 1000, Static, Name Test_Invoer
{
IF v1000 < 10
   {
   v1001 = 1
   }
}


// -------------------------------------------------
// VOORBEELD :  L0
// VOORBEELD :  L1
// VOORBEELD :  L2
// -------------------------------------------------

// L0 [ L nul ] L1  en L2   voor getal waarden



// -------------------------------------------------
// VOORBEELD :  LIMIT                    limiet waarde
// -------------------------------------------------



// -------------------------------------------------
// VOORBEELD :  MOD                            moduleer
// -------------------------------------------------

// In dit voorbeeld laten we een LED knipperen

v1001 = 0

// [ SWITCH ] ON-OFF
Var 1000, Static, Name Test_InvoerSW, Link IOCARD_SW, Device 0, Input 10, Type I
{
IF v1000 = 1                 // schakelaar aan
   {
   IF v1002 = 0              // Start enkele als de LED uit was [ niet knipperde ]
      {
      v1001 = 10             // Begin Tijd voor de TIMER
      v1001 = TIMER 0 -1 50   // 0 = uit Tijd Timer   -1 = telwaarde  50 = 50/100 sec is
interval timer
      }
   }
}


var 1001, Static, Name Test_Knipper
{
```

```
L1 = MOD v1001 2     // MOD = Modulator 2

IF L1 = 0
   {
   v1002 = 0         // Zet LED uit
   }
ELSE
   {
   v1002 = 1         // Zet LED aan
   }
}

Var 1002, Static, Name Test_Uitkomst_, Link FSUIPC_OUT, Offset $56E2, Length 1

// -------------------------
// UITKOMST
// -------------------------
v1001 = Tijdwaarde 10,9,8,7,6,5,4,3,2,1,0
MOD 2 = 10 MOD 2 = 0, 9 MOD 2 = 1, 8 MOD 2 = 0, 7 MOD 2 = 1,  etc
De TIMER interval zorgt voor de knipper frequentie



// ------------------------------------------------
// VOORBEELD :  NOT                              niet
// ------------------------------------------------



// ------------------------------------------------
// VOORBEELD :  OR                                 of
// ------------------------------------------------



// ------------------------------------------------
// VOORBEELD :  POWER
// ------------------------------------------------



// ------------------------------------------------
// VOORBEELD :  RANDOM                    willekeurig
// ------------------------------------------------

v1000 = 1


Var 1000, Static, Name Test_Invoer
{
IF v1000 = 1
   {
   v1002 = 1  // Zet LED aan
   }
ELSE
   {
   L1 = RANDOM 100 400
   v1002 = DELAY 0 L1 // Tussen de 1 en 4 sec blijft de LED na branden voor hij uitgaat
   }
}

Var 1002, Static, Name Test_LED, Link IOCARD_OUT, Device 0, Output 1
```

```
// -------------------------
// UITKOMST
// -------------------------

RANDOM para1 Para2
Para1 = Minimale waarde
Para2 = Maximale waarde

L1 = RANDOM 100 400
v1002 = DELAY 0 L1 // Hier wordt tussen de 1 en 4 sec gewacht voordat de
                      waarde 0 wordt aangeboden aan de LED variabele



// ------------------------------------------------
// VOORBEELD :  ROTATE                          draaien
// ------------------------------------------------

Var 1000, Static, Name Test_InvoerEN, Link IOCARD_ENCODER Input 1 Aceleration 2 Type 2
{
L0 = v1000
v1001 = ROTATE 0 359 L0
}

Var 1001, Static, Name Test_Uitkomst

// -------------------------
// UITKOMST
// -------------------------

ROTATE para1 Para2 Para3
Para1 = Minimale waarde
Para2 = Maximale waarde
Para3 = Rotatie stappen

Als je de encoder draait [ clockwise ] zal de waarde tussen de 0 en 359
in stapjes van para3 waarde worden verhoogt



// ------------------------------------------------
// VOORBEELD :  ROUND                          afronden
// ------------------------------------------------

v1000 = 27


Var 1000, Static, Name Test_Invoer
{
L0 = v1000 / 4
v1001 = ROUND L0
}

Var 1001, Static, Name Test_Uitkomst

// -------------------------
// UITKOMST
// -------------------------
L0 = 27 / 4
v1001 = ROUND 6,75
v1001 = 7
```

```
// ------------------------------------------------
// VOORBEELD :  SETBIT            Zet BIT waarde op 1
// ------------------------------------------------

// [ SWITCH ] ON-OFF
Var 1000, Static, Name Test_InvoerSW, Link IOCARD_SW, Device 0, Input 10, Type I
{
IF v1000 = 1            // schakelaar aan
   {
   v1001 = CLEARBIT 4   // [ BIT 4 wordt op 0 [nul] gezet ]
   }
ELSE                    // schakelaar uit
   {
   v1001 = SETBIT 4     // [ BIT 4 wordt op 1 gezet ]
   }

Var 1001, Static, Name Test_Uitkomst_, Link FSUIPC_OUT, Offset $56E2, Length 1



// ------------------------------------------------
// VOORBEELD :  SETSOUND
// ------------------------------------------------



// ------------------------------------------------
// VOORBEELD :  SIN
// ------------------------------------------------



// ------------------------------------------------
// VOORBEELD :  TESTBIT            test de BIT waarde
// ------------------------------------------------

// [ FSUIPC_IN ]
Var 1000, Static, Name Test_Invoer, Link FSUIPC_IN, Offset $3122, Length 1
{
C1 = TESTBIT v1000, 1     // Hier wordt de waarde ingelezen van Offset $3122 [ = FSUIPC
waarde ]
IF C1                     // Als deze waar is dan zetten wij de LED aan
   {
   v1001 = 1              // Offset $3122 LED ON
   }
ELSE
   {
   v1001 = 0              // Offset $3122 LED OFF
   }
}

Var 1001, Static, Name Test_Uitvoer, Link IOCARD_OUT, Device 0, Output 10



// ------------------------------------------------
// VOORBEELD :  TIMER                        Tijdklok
// ------------------------------------------------

// In dit voorbeeld laten we een LED knipperen

v1001 = 0

// [ SWITCH ] ON-OFF
```

```
Var 1000, Static, Name Test_InvoerSW, Link IOCARD_SW, Device 0, Input 10, Type I
{
IF v1000 = 1                  // schakelaar aan
   {
   IF v1002 = 0               // Start enkele als de LED uit was [ niet knipperde ]
      {
      v1001 = 10              // Begin Tijd voor de TIMER
      v1001 = TIMER 0 -1 50   // 0 = uit Tijd Timer   -1 = telwaarde  50 = 50/100 sec is
interval timer
      }
   }
}


var 1001, Static, Name Test_Knipper
{
L1 = MOD v1001 2     // MOD = Modulator 2

IF L1 = 0
   {
   v1002 = 0         // Zet LED uit
   }
ELSE
   {
   v1002 = 1         // Zet LED aan
   }
}

Var 1002, Static, Name Test_Uitkomst_, Link FSUIPC_OUT, Offset $56E2, Length 1

// -------------------------
// UITKOMST
// -------------------------
v1001 = Tijdwaarde 10,9,8,7,6,5,4,3,2,1,0
MOD 2 = 10 MOD 2 = 0, 9 MOD 2 = 1, 8 MOD 2 = 0, 7 MOD 2 = 1,  etc
De TIMER interval zorgt voor de knipper frequentie



// ------------------------------------------------
// VOORBEELD :  TOBCD
// ------------------------------------------------



// ------------------------------------------------
// VOORBEELD :  TOGGLE
// ------------------------------------------------



// ------------------------------------------------
// VOORBEELD :  TRUNC                       afknotten
// ------------------------------------------------

v1000 = 27


Var 1000, Static, Name Test_Invoer
{
L0 = v1000 / 4
v1001 = TRUNC L0
}
```

```
Var 1001, Static, Name Test_Uitkomst

// -------------------------
// UITKOMST
// -------------------------
L0 = 27 / 4
v1001 = TRUNC 6,75
v1001 = 6



// ------------------------------------------------------------------------------
// INDEX LINKS                                                      INDEX LINKS
// ------------------------------------------------------------------------------

FSUIPC_IN
FSUIPC_OUT
FSUIPC_INOUT

IOCARD_DISPLAY
IOCARD_ENCODER
IOCARD_OUT   [ USB_MASTER & USB_OUTPUTS]
IOCARD_SW    [ USB_MASTER ]
SOUND
SUBRUTINE

USB_ANALOGIC
USB_DCMOTOR
USB_KEYS
USB_RELAYS
USB_SERVOS


// -------------------------------------------------
// VOORBEELD :  FSUIPC_IN
// -------------------------------------------------
Var VVVV, Static, Name NNNN, Link FSUIPC_IN, Offset $XXXX, Length Y

VVVV = variabel nummer    [tussen 0000-9999 Dec]
NNNN = alias naam         [max. 14 karakters   ]
XXXX = offset variabele   [tussen 0000-FFFF Hex]
Y    = offset lengte


// -------------------------------------------------
// VOORBEELD :  FSUIPC_OUT
// -------------------------------------------------
Var VVVV, Static, Name NNNN, Link FSUIPC_OUT, Offset $XXXX, Length Y

VVVV = variabel nummer    [tussen 0000-9999 Dec]
NNNN = alias naam         [max. 14 karakters   ]
XXXX = offset variabele   [tussen 0000-FFFF Hex]
Y    = offset lengte


// -------------------------------------------------
// VOORBEELD :  FSUIPC_INOUT       [ NIET GEBRUIKEN ]
// -------------------------------------------------
Var VVVV, Static, Name NNNN, Link FSUIPC_INOUT, Offset $XXXX, Length Y

VVVV = variabel nummer    [tussen 0000-9999 Dec]
NNNN = alias naam         [max. 14 karakters   ]
XXXX = offset variabele   [tussen 0000-FFFF Hex]
Y    = offset lengte
```

```
// ------------------------------------------------
// VOORBEELD :  IOCARD_DISPLAY
// ------------------------------------------------
Var VVVV, Static, Name NNNN, Link IOCARD_DISPLAY, Device DD, Digit EE Numbers FF

VVVV = variabel nummer     [tussen 0000-9999 Dec]
NNNN = alias naam          [max. 14 karakters   ]
DD   = IDX nummer overeenkomstig in sioc.ini
EE   = Nummer 1ste Display [max. 16 display's per KAART]
FF   = Lengte vanaf het 1ste display


// ------------------------------------------------
// VOORBEELD :  IOCARD_ENCODER
// ------------------------------------------------
Var VVVV, Static, Name NNNN, Link IOCARD_ENCODER, Device DD, Input EE, Aceleration FF,
Type GG

VVVV = variabel nummer     [tussen 0000-9999 Dec]
NNNN = alias naam          [max. 14 karakters   ]
DD   = IDX nummer overeenkomstig in sioc.ini
EE   = analog input nummer [afhankelijk op welke kaart]
FF   = Snelheid factor bij continue draaiing
GG   = type 1
GG   = type 2              [een gray type rotary encoder]


Var 0001 Link IOCARD_ENCODER Input 40 Aceleration 2 Type 2
{
   L0 = v0001 // * -1 turning clockwise should be plus
   v0002 = ROTATE 0 359 L0
}

Var 0002 // heading (0 .. 359)


// ------------------------------------------------
// VOORBEELD :  IOCARD_OUT
// ------------------------------------------------
Var VVVV, Static, Name NNNN, Link IOCARD_OUT, Device DD, Output XX

VVVV = variabel nummer     [tussen 0000-9999 Dec]
NNNN = alias naam          [max. 14 karakters   ]
DD   = IDX nummer overeenkomstig in sioc.ini


// ------------------------------------------------
// VOORBEELD :  IOCARD_SW
// ------------------------------------------------
Var VVVV, Static, Name NNNN, Link IOCARD_SW, Device DD,Input XX, Type Y

VVVV = variabel nummer     [tussen 0000-9999 Dec]
NNNN = alias naam          [max. 14 karakters   ]
DD   = IDX nummer overeenkomstig in sioc.ini
XX   = Input nummer kaart
Y    = type P              [Var changes from n -> n+1 so on]
Y    = type I              [Var = 1 if ON and 0 OFF]

// ------------------------------------------------
// VOORBEELD :  SOUND
// ------------------------------------------------
Var 0001, Static, Name NNNN,  Link SOUND, Type S
```

```
VVVV = variabel nummer     [tussen 0000-9999 Dec]
NNNN = alias naam          [max. 14 karakters   ]
Y    = type niets of S


// -------------------------------------------------
// VOORBEELD :  USB_ANALOGIC
// -------------------------------------------------
Var VVVV, Static, Name NNNN, Link USB_ANALOGIC, Device DD, Input EE, posL LLL, posC CCC,
posR RRR

VVVV = variabel nummer     [tussen 0000-9999 Dec]
NNNN = alias naam          [max. 14 karakters   ]
DD   = IDX nummer overeenkomstig in sioc.ini
EE   = analog input nummer [afhankelijk op welke kaart]
LLL  = maximale positie potentiometer naar links
CCC  = maximale positie potentiometer
RRR  = maximale positie potentiometer naar rechts


Var 0001, Static, Name tst_trim, Link USB_ANALOGIC, Device 51, Input 1, posL 1, posC
128, posR 255
{
   v0002 = v0001
}

Var 0002, Static, Name tst_servo, Link USB_SERVOS, Device 61, Output 1, PosL 1, PosC
512, PosR 1023


// -------------------------------------------------
// VOORBEELD :  USB_RELAYS
// -------------------------------------------------
Var VVVV, Static, Name NNNN, Link USB_RELAYS, Device DD, Output XX

VVVV = variabel nummer    [tussen 0000-9999 Dec]
NNNN = alias naam         [max. 14 karakters   ]
DD   = IDX nummer overeenkomstig in sioc.ini
XX   = relais nummer      [tussen 1-7 ]


Var 0001, Static, Name tst_switch, Link IOCARD_SW, Device 0,Input 1
{
If v0001 = 1
   {
   v0002 = 1
   }
ELSE
   {
   v0002 = 0
   }
}

Var 0002, Static, Name tst_relay, Link USB_RELAYS, Device 91, Output 1


// -------------------------------------------------
// VOORBEELD :  USB_SERVOS
// -------------------------------------------------
Var VVVV, Static, Name NNNN, Link USB_SERVOS, Device DD, Output EE, PosL LLL, PosC CCC,
PosR RRR

VVVV = variabel nummer     [tussen 0000-9999 Dec]
```

```
NNNN = alias naam          [max. 14 karakters   ]
DD   = IDX nummer overeenkomstig in sioc.ini
EE   = analog input nummer [afhankelijk op welke kaart]
LLL  = maximale positie servo naar links
CCC  = maximale positie servo
RRR  = maximale positie servo naar rechts


Var 0001, Static, Name tst_trim, Link USB_ANALOGIC, Device 51, Input 1, posL 1, posC
128, posR 255
{
   v0002 = v0001
}

Var 0002, Static, Name tst_servo, Link USB_SERVOS, Device 61, Output 1, PosL 1, PosC
512, PosR 1023



// ------------------------------------------------------------------------
// VOORBEELD : OFFSET CYCLUS
//
//             op basis van ervaring        bv PARKING BRAKES
// ------------------------------------------------------------------------

// -------------------------------------------------
// [ FLAG Parking Brake SWITCH ]
Var 0502, Static, Name FL_PrkBrkSW   // Voor eigen gebruik ivm programmering


// [ 1e INLEZEN OFFSET ]

// -------------------------------------------------
// [ Parking Brake FSUIPC_IN ]
Var 3001, Static, Name PrkBrk, Link FSUIPC_IN, Offset $0BC8, Length 2
{
IF v3001 = 0 // Parking Brake OFF
   {
   v3002 = 0 // LED OFF = 0 bij USBInput Card en 1 bij Input Card
   }
ELSE         // Parking Brake ON
   {
   v3002 = 1 // LED ON = 1 bij USBInput Card en 0 bij Input Card
   }
}



// [ 2e OUTPUT ]

// -------------------------------------------------
// [ Parking Brake LED ] RED
Var 3002, Static, Name PrkBrkLD, Link IOCARD_OUT, Device 0, Output 00



// [ 3e INPUT ]

// -------------------------------------------------
// [ Parking Brake SWITCH ] ON-OFF
Var 3003, Static, Name PrkBrkSW, Link IOCARD_SW, Device 0, Input 00, Type I
{
IF v3003 = 1
   {
```

```
   v3004 = 32767 // FSUIPC Offset $0BC8 OUT
   v0502 = 1     // FLAG Parking Brake SWITCH
   }
ELSE
   {
   v3004 = 0     // FSUIPC Offset $0BC8 OUT
   v0502 = 0     // FLAG Parking Brake SWITCH
   }
}



// [ 4e WIJZIGEN OFFSET ]

// -------------------------------------------------
// [ Parking Brake FSUIPC_OUT ]
Var 3004, Static, Name PrkBrk_, Link FSUIPC_OUT, Offset $0BC8, Length 2



// ------------------------------------------------------------------------
// VOORBEELD : INPUT TYPE P
//
//             op basis van ervaring        Pulse Switch type P
// ------------------------------------------------------------------------

// -------------------------------------------------
// [ FS Chronometer Time Date SWITCH ] MON-OFF Type PULSE
Var 1015, Static, Name CnoTmeDatSW, Link IOCARD_SW, Device 0, Input 00, Type P
{
IF v1015 = 1
   {
   L0 = v0507 + 1
   v0507 = L0

   IF v0507 = 1     // FLAG Chronometer Time Date SWITCH
      {
      v1013 = 1     // MODE Time Local
      }

   IF v0507 = 2
      {
      v1013 = 4     // MODE Date & Month
      }

   IF v0507 > 2
      {
      v0507 = 0
      v1013 = 5     // MODE Year
      }
   }
}

// Elke keer dat de Push-Button wordt ingedrukt zal de waarde v0507 worden
// verhoogt met 1. Zodra de v0501 waarde groter is dan 2, in dit geval 3
// wordt hij weer teruggezet op 0. Zo kan ik een mode cyclus doorlopen.


// ------------------------------------------------------------------------
```