parameter: &B.MBMX
&B.MBMY

meaning: Specify the master bad mark position in coordinates relative to the PCB origin.

values: -999.99 to 999.99 [mm]

parameter: &B.MTEX

meaning: A flag that indicates that mount jobs must be executed.

values: 0 - Skip mount lines
1 - Execute mount lines

parameter: &B.OPMD

meaning: RUNNING MODE 2. This parameter can be set to 'Normal', 'Check' or 'Pass'.
* When set to normal, the boards are processed normally, mounting/dispensing is performed.
* When set to Check, the board is also processed as usual, only no checks on mounting or picking is performed. This allows you to run the machine in 'demo' mode. In this mode, the MIS data is not updated.
* When set to Pass, all boards that enter the CSM leave it without being processed. This is only possible when parameter &B.ASFG is set to 0 (AUTO).

values: 0 - Normal
1 - Check (no checking performed)
2 - Pass

parameter: &B.PCBW

meaning: Specifies the measured PCB width. This parameter will only be active when the CSM has an automatic width conveyor system.

values: -999.99 to 999.99 [mm]

parameter: &B.PDEX

meaning: A flag that indicates that pre-dispense jobs must be excuted.

values: 0 - Skip pre-dispense lines
1 - De execute pre-dispense lines

parameter: &B.SBST

meaning: This parameter specifies if the substopper must be used or not.

values: 0 - Do not use sub-stopper
1 - Use sub-stopper

8-4-3-3 The Pre-dispense Lines

The pre-dispense lines specify a number of points where the pre-dispense dots will be placed.

pre-dispense-line ::= **"J"number = x-position y-position z-disp**

where

number - sequence number of the pre-dispense line (0..)

x-position, y-position - position of the pre-dispense point
this position is given in **block** coordinates of block with line number 1 (see Origin-lines).

z-disp - string of 8 digits
Assume that this string reads ABCDEFGH, then:

- A - vision file; 0 = not used, 1 = used
- BCD - the glue time in [ms]
- EF - rotation of glue head; 00 = 0 degree, 01 = 90 degrees
- G - the head number that is used to glue
- H - skip-parameter; 0 = execute the line, 1 = skip the line

Pre-dispense lines are executed only when the skip parameter in **z-disp** specifies that the line must be executed and the running parameter PDEX indicates that the pre-dispense job must be done.

8-4-3-4 The Dot-dispense Lines

The dispense lines specify a number of points where dispense dots will be placed.

dispense-line ::= **"D"number = x-position y-position z-disp**

where

number - sequence number of the dispense line (0..)

x-position, y-position - position of the dispense point
this position is given in block coordinates.

z-disp - string of 7 digits
Assume that this string reads ABCDEFG, then:

- ABC - the glue time in [ms]
- DE - rotation of glue head; 00 = 0 degree, 01 = 90 degrees
- F - the head number that is used to glue
- G - skip-parameter; 0 = execute the line, 1 = skip the line

Dispense lines are executed only when the skip parameter in **z-disp** specifies that the line must be executed and the running parameter DDEX indicates that the dispense jobs must be done.

8-4-3-2 The Fiducial Lines

The fiducial lines specify the positions of PCB, Block and Point fiducials.

fiducial-line ::= "I"number = x1-position y1-position x2-position y2-position
"i"number = mark-dia srch-area x-off y-off z-fid

where

number - this is the reference number of the fiducial line. Number 0 is used to specify the PCB or Block fiducial (depends on parameter &B.FID1). Non-zero number specify point (local) fiducial lines, related to PCB mount point or BLOCK origin (depends on parameter &H.PFID). Range: 0.....15.

x1-position, y1-position - position of the first local fiducial, related to PCB point mount or BLOCK origin (depends on parameter &H.PFID). Except fiducial line number I0, this is related to PCB or BLOCK fiducial (depends on parameter &B.FID1).

x2-position, y2-position - position of the second local fiducial, related to PCB point mount or BLOCK origin (depends on parameter &H.PFID). Except fiducial line number I0, this is related to PCB or BLOCK fiducial (depends on parameter &B.FID1).

mark-dia - mark diameter in [mm]

srch-area - search area heigth and width in [mm]

x-off, y-off - position where to start scanning (relative to the fiducial position). **Currently not used.**

z-fid - a string of 5 digits.
Assume that this string reads ABCDE, then:

AB - vision file number, only if the reading equipment is the camera; range 00..29
C - reading equipment; 0 = beam sensor, 1 = camera
D - scan type; 0 = XY, 1 = XYX
The Scan type defines the algorithm used when a beam sensor is used. The movements of the beam are:
XY: first find X-centre, at X-centre find the Y-centre. So first move in X-direction an then in Y-direction.
XYX: first find X-centre, at X-centre find Y-centre, then at Y-centre determine the X-centre again. so, move in X-direction, then in Y-direction and then in X-direction again.
E - mark logic; 0 = on (reflective), 1 = off (non-reflective)

Examples:

```
I0 =  100.00  100.00  200.00  200.00
i0 =    1.00    2.00    0.00    0.00   29100
```

8-4-3 Point Data

The point-data group contains positions that tell the system where to execute certain actions. It contains Origin positions, Fiducial positions, Pre-dispense positions, Dot-dispense positions, Line-dispense positions, Line-dispense SUB, Mount positions and Bad Mark positions.

```
point-data ::=
        { origin-line |
          fiducial-line |
          pre-dispense-line |
          dot-dispense-line |
          line-dispense-line |
          line-dispense sub-line |
          mount-line |
          bad mark-line
        }
```

As shown, the sequence in which the lines appear in the file is irrelevant. The system will allways execute the lines in the order:

1 - pre-dispense lines
2 - dot-dispense lines
3 - line-dispense lines
4 - line-dispense sub lines
5 - bad mark lines
6 - mount lines

The origin and fiducial lines give reference data for the other lines.

8-4-3-1 The Origin Lines

The origin lines specify the position of the PCB origin and the BLOCK origins. The coordinates in these lines are given in machine coordinates in units of [mm] or [degree].

origin-line ::= **"O"number = x-position y-position rotation block-skip**

where

number - an integral number (0,1,..).
Number 0 is used to indicate the PCB origin, other numbers to specify block origin points.

x-position, y-position - position of the block/PCB-origin in [mm]

rotation - rotation in degrees

block-skip - indicates if a block must be processed or not.
This parameter has no meaning for O0.
Value 0 means: process the block.
Value 1 means: do not process the block.

Examples:

```
O0  =   300.0   0.0    0.0   0
O1  =   100.34  20.5   180.0 01
```

Mount lines are only executed when their skip parameter (see z-mount) is set to 'execute' and the running condition parameter MTEX is set to 'execute'.

Common collects a set of successive mount lines until it detects that the head number of the next mount line is lower then the head number in the current mount line. The set of mount lines that are collected this way is called **a cluster.**
First <COMMON> picks the components in the cluster in the sequence in which the mount lines appear in the cluster. Then <COMMON> starts the mount parts of the mount lines in the cluster. The sequence in which the mount parts are executed depend on the optimize parameter of the **first** mount line in the cluster.

8-4-3-8 The BAD MARK Lines

The BAD MARK lines specify where the bad mark are situated. This line is only available when machine parameter &H.BMRK =1 (bad mark position is related to PCB origin).

bad mark-line ::= "N"number = x-position y-position z-badmark

where

number - sequence number of the bad mark line (0,1..)

x-position, y-position - bad mark position. The bad mark position is related to PCB origin.

z-badmark - a 4 digit string
Assume that the string reads ABCD, then:

A - Defines the equipment that is used for bad mark sensing (like &B.BSEN)
0 - beam sensor
1 - camera

B - Specifies the logic of the beam sensor. When set to ON, the bad mark is reflective, when set to OFF, the bad mark is non-reflective (like &B.BMLG)
0 - ON
1 - OFF

CD- Specifies the vision file to be used for scanning the bad marks when the camera is used, 0 to 29 (like &B.BMLG)

8-4-4 Limitations

8-4-4-1 Amount of Point Data Lines

The total number of (allocated) point data lines in all board files together can not exceed 2560 lines.

Point data lines are allocated in groups of 16 to the individual board files. So if a board file contains 162 lines, space for 176 lines is allocated to the file.

8-4-4-2 Number of Board Files

Upto 14 board files can be stored.

z-disp - string of 8 digits
Assume that this string reads ABCDEFGH, then:

AB - start % of the line (only active when job set = DISP)
CD - end % of the line (only active when job set = DISP)
EF - rotation angle 00 = 0 degree, 01 = 90 degrees
GH - job set; 0 = MOVE
1 = DISP.
2 = CLEANING

Line Dispense SUB lines are executed only when the skip parameter in the Line dispense **z-disp** specifies that the line must be executed and the running parameter LDEX indicates that the line dispense jobs must be done.

8-4-3-7 The Mount Lines

The mount lines specify where and what components must be placed on the board.

mount-line ::= "M"**number = x-position y-position rotation z-mount**

where

number - sequence number of the mount line (0,1,..)

x-position, y-position, rotation - mount position
the mount position is given in block coordinates. The rotation is a rotation relative to the rotation at which the component is picked (see feeder file).

z-mount - a 8 digit string
Assume that the string reads ABCDEFGH, then:

A - point fiducial line number; 0 means not used, 1...9 an A.....F indicates the fiducial line number.
B - ***Not use***
C - head number; 1..3
D - nozzle number; 1..3 For nozzle station (B1,B2,B3)
- nozzle number; 4..9 For Chuck staion (A1,A2,..A6) Not used by PHILIPS
E - skip parameter; 0 = execute the line, 1 = skip the line.
FGH - part number; value 1..120; this is a index into the feeder file. The system determines through this number which feeder definition must be used for picking. Refer to the feeder file description for more information.

8-4-3-5 The Line-dispense Lines

The line dispense lines specify a number of points where a dispense line will be placed. The line dispense and line dispense sub functions are used together, as a set. For more details for the line dispense sub lines see 9-4-3-6 The line dispense sub lines.

line dispense-line ::= **"L"number = x-position y-position rotation z-disp**

where

number - sequence number of the line dispense line (0..)

x-position, y-position - position of the line dispense point

rotation - indicates the pattern rotation set in the line dispense sub settings

z-disp - string of 8 digits
Assume that this string reads ABCDEFGH, then:

- A - skip -parameter; 0= execute the line, 1 = skip the line
- B - the head number that is used to glue
- CD - 00
- EFGH - this indicates the number of the pattern set in the line dispense sub settings (SUB NO)

Line Dispense lines are executed only when the skip parameter in **z-disp** specifies that the line must be executed and the running parameter LDEX indicates that the line dispense jobs must be done.

8-4-3-6 The Line-dispense Sub Lines

The line dispense sub lines specify linear patterns called out by the SUB NO, of the line dispense (see 9-4-3-5).

line dispense sub-line ::= **"C"number = x-position y-position w-disp z-disp**

where

number - sequence number of the line dispense sub lines (0..)

x-position y-position - indicates a point of the line dispense pattern

w-disp - string of 5 digits
Assume that this string reads ABCDE, then:

- ABC - dispense time [ms]
- DE - velocity; this is used for changing the thickness of the line. Normally it is set to 0.
  1 9
  FINE HEAVY

8-4-5 File Names

The board files can be read and written through the serial port via on-line commands. The collection of all board files can be accessed under the name "BRD". Individual board files can be accessed under the name "B<pcb-name>". If you write to a board file twice without deleting it in between, the data adds up, that is, the new data is added to the old data, it does not overwrite it.

Examples:

@READ BRD	-	reads all board files
@READ B<TEST>	-	reads the board file for board with PCBNA-ME=TEST
@WRITE BRD	-	write one or more board files
@WRITE B<TEST>	-	write one board file

When writing to BRD, the line "PCBNAME=...." indicates the board file name and should be send first.
When a single file is written using the B<pcb-name> convention then the PCB-name given in the filename is the one that is used by UFOS, not the name that follows "PCB-NAME=".

You can erase one or all board files with the @ERA on-line command. This will also delete the MIS data for the involved board.

E.g.:

@ERA B<TEST>	-	erase one board file
@ERA BRD	-	erase all board files

8-4-6 Example

```
PCBNAME=TEST
&B.COMMENT=/          /
&B.ASFG=0
&B.OPMD=0
&B.PDEX=1
&B.DDEX=1
&B.LDEX=1
&B.MTEX=1
&B.FIX=0
&B.SBST=0
&B.BMRK=0
&B.FID1=0
&B.CONV=0
&B.BMKX=  11.22
&B.BMKY=  34.00
&B.MBMX=  56.00
&B.MBMY=  78.00
&B.BSEN=1
&B.BMLG=0
&B.BFNO=29
&B.PCBW=   0.00
&B.DCHK=0
O0  =  429.61   95.72    0.00    0
O1  = -218.52    7.7     0.00    0
I0  =    0.00    0.00  218.44  111.76
```

It also contains per feeder that was is in processing the PCB type and per head:

- feeder number
  - head number
  - feed count
  - error count
  - feed ratio

The layout of the cumulative PCB area is a repetition for each PCB name of:

```
$CUMULATIVE
PCB NAME    QTY. MR[%] MT[S] CT[S]  PQ
######## ****** ***.* ***.* ***.* ***
$FEEDER
 NO NAME        QTY. ERR     RATE
 ** ######## ****** ***  ***.**%
            ............
            ............
 ** ######## ****** ***  ***.**%
$HEAD
 NO    QTY. ERR1    RATE1   ERR2   RATE2
  1 ****** ****  ***.**%   **** ***.**%
  2 ****** ****  ***.**%   **** ***.**%
  3 ****** ****  ***.**%   **** ***.**%
  4 ****** ****  ***.**%   **** ***.**%
$F**
H1= ****** **** ****
H2= ****** **** ****
H3= ****** **** ****
H4= ****** **** ****
      .........
      .........
$F**
H1= ****** **** ****
H2= ****** **** ****
H3= ****** **** ****
H4= ****** **** ****
```

8-5-2-3 Trip Data

The trip data contains identical information as in one section of the cumulative PCB area. It contains the data for the last processed PCB type.

The layout is, apart from the keyword TRIP (which is CUMULATIVE in the previous section), identical to that of the cumulative PCM area:

Single time:

4- the total time the machine has been powered on since that last time that power was applied to the machine

5- the total time the machine has been running since the last time that power was applied to the machine

6- the ration between 4 and 5

Layout:

```
$PRD$
/OPR/
*****H **M **S          *****H **M **S
/RUN/
*****H **M **S          *****H **M **S
/RAT/
***.**%                 ***.**%
```

8-5-2-2 Cumulative PCB Area

The cumulative PCB area contains a collection of data blocks, each for one type of PCB (one PCBNAME).

For each PCB-type the following data is part of this block:

- PCB Name
- Number or processed boards (QTY)
- Mounting rate (MR)
- Total mounting time (MT)
- Average cycle time (CT)
- Amount of components used for this type (PQ)

Each block contains for each feeder that is used in for the PCB type (determined from board file) the following:

- feeder number (part number/index in feeder file)
- feeder name (comes from the comment field in the feeder file)
- feed count (incl. errors) (QTY)
- feed errors (ERR)
- feed rate (RATE)

Each block contains for each head the following:

- head number
- number of feed actions (including retries) (QTY)
- number of feed errors the head was involved in (ERR1)
- feed rate (RATE1)
- number of mounting errors the head was involved in (ERR2)
- mount rate (RATE2)

```
i0   =    0.00    0.00    0.00    0.00    29100
J0   =  123.00  123.00                   11110010
J1   =  245.00  456.00                    2220121
J2   =   45.60   45.70                    3330130
D0   =  191.44   72.54                    4440010
D1   =  188.29   72.54                    5550121
D2   =   45.20   45.30                    6660130
L0   =   11.00   22.00  180.00            1000012
L1   =   45.78   89.42  360.00           12001222
L2   =   45.10   45.20  270.00            3003333
C0   =    0.00    0.00              13          0
C1   =   10.00    0.00              12   33220001
C2   =   10.00   10.00           66614          2
M0   =   24.76    4.11    0.00            6100021
M1   =  117.17   62.86   90.00             200024
M2   =   29.84   30.48    0.00             100014
M3   =   24.76    9.86    0.00            6100021
M4   =  117.17   83.52   90.00             200024
N0   = -120.00    2.50                      1018
N1   = -130.00    2.50                       118
N2   = -140.00    2.50                        18
```

8-5 The MIS Data Files

8-5-1 Introduction

The CSM collects figures about the production process. It measures the time it is powered on, the time it is actually processing, the number of boards processed, pick and place error and cycle counts.

This data can be inspected from on the video monitor or read through on-line commands.

8-5-2 Structure of the MIS Data

The data that is collected consistes of a number of items:

- time data
- cumulative PCB area
  consisting of a data area for each PCB-type
- trip data

8-5-2-1 The Time Data

The time data consists of:

Cumulative time:

1- the total time during which the machine was powered-on since the last time the memory was initialised
2- the total time the machine has been running since the last time the memory was initialised
3- the ration between 1 and 2

8-5-5 Example

```
$PRD$
/OPR/
  179H 18M 40S          0H  3M 33S
/RUN/
   49H 10M 34S          0H  0M  0S
/RAT/
  27.43%               0.00%

$CUMULATIVE
PCB NAME   QTY. MR[%] MT[S] CT[S]  PQ
TOPSIDE       0 100.0   0.0   0.0   0
$FEEDER
 NO NAME       QTY. ERR     RATE
 19 1206          4   3   25.00%
$HEAD
NO   QTY. ERR1    RATE1   ERR2   RATE2
 1      0    0    0.00%      0   0.00%
 2      4    3   25.00%      0 100.00%
 3      0    0    0.00%      0   0.00%
$F19
H1=      0    0    0
H2=      4    3    0
H3=      0    0    0

$CUMULATIVE
PCB NAME   QTY. MR[%] MT[S] CT[S]  PQ
GERARD       10 100.0   3.3   3.3   1
$FEEDER
 NO NAME       QTY. ERR     RATE
 22 0603          5   0  100.00%
 40 0603          6   2   66.67%
$HEAD
NO   QTY. ERR1    RATE1   ERR2   RATE2
 1      0    0    0.00%      0   0.00%
 2     11    2   81.82%      0 100.00%
 3      0    0    0.00%      0   0.00%
$F22
H1=      0    0    0
H2=      5    0    0
H3=      0    0    0
$F40
H1=      0    0    0
H2=      6    2    0
H3=      0    0    0

$TRIP
PCB NAME   QTY. MR[%] MT[S] CT[S]  PQ
GERARD       10 100.0   3.3   3.3   1
$FEEDER
 NO NAME       QTY. ERR     RATE
 19 1206          0   0    0.00%
 22 0603          5   0  100.00%
 40 0603          6   2   66.67%
```

8-5-4 Files

The MIS data can be obtained in various parts:

8-5-4-1 All Data

All data can be obtained by reading the PRD file. It is a concatenation of the time data, cumulative PCB area and trip data (in this sequence).
Each section is terminated by an empty line (just CR/LF).

```
@READ PRD
```

It can be cleared by initialising the memory of the CSM or by explicitely initialising the file:

```
@INIT MEM   -- this deletes more then just PRD!!
               (eg. VIS, BRD, MCH, FDR)
@INIT PRD
```

8-5-4-2 Cumulative PCB Area

The cumulative data for all PCB-types can be obtained by reading the PCB file. The data for all PCB types are concatenated. The entire data block is terminated by an empty line (just CR/LF).

```
@READ PCB       read entire cumulative area
```

The data for a PCB type is deleted when the board file is deleted. It cannot be deleted in another way through on-line commands. It is possible to clear the figures through the VDU however.

```
@ERA B<TEST>   clear board data and also MIS
```

8-5-4-3 Trip Data

The trip data can be read via the file TRP. The block of data is terminated by an empty line.

```
@READ TRP
```

It is cleared during the first time a @PCBRUN command is executed after giving a @SWI command. It can explicitely be cleared by an initialise command:

```
@INIT TRP
```

```
$TRIP
PCB NAME   QTY. MR[%] MT[S] CT[S]  PQ
######## ****** ***.* ***.* ***.* ***
$FEEDER
 NO NAME       QTY. ERR     RATE
 ** ######## ****** ***  ***.**%
             ............
             ............
 ** ######## ****** ***  ***.**%
$HEAD
NO   QTY. ERR1    RATE1   ERR2   RATE2
 1 ****** ****  ***.**%   **** ***.**%
 2 ****** ****  ***.**%   **** ***.**%
 3 ****** ****  ***.**%   **** ***.**%
 4 ****** ****  ***.**%   **** ***.**%
$F**
H1= ****** **** ****
H2= ****** **** ****
H3= ****** **** ****
H4= ****** **** ****
      .........
      .........
$F**
H1= ****** **** ****
H2= ****** **** ****
H3= ****** **** ****
H4= ****** **** ****
```

8-5-3 Update of Data

The data of the various parts of the MIS data are updated or initialised as described below.

The cumulative time fields are set to zero when memory is initialised explicitely (@INIT MEM).

The single time fields are set to 0 when the machine is powered-up.

An entry for a PCB type is created in the cumulative PCB data when the board file for the PCB is compiled (so during execution of the @PCBRUN command after a @SWI command).

An entry for a PCB type is deleted from the cumulative area when the board file for it is deleted (@ERA B<...>) or when the PRD file is initialised (@INIT PRD).

The trip data is cleared and initialised when another PCB type is selected, during compilation of the board file (so during execution of the @PCBRUN command after a @SWI command).

The PCB/TRP data is updated when the head moves up at the end of a pick cycle and when the head moves up at the end of mounting a component.

```
CAMERA    :2
OBJECT    :BLACK
BINARY    :AUTO
FILL      :2
CUT       :NO
DISPLAY   :CROSS0
WINDOW1   :ON+    (  0,   0)(255,239)
WINDOW4   :OFF
OPTION01  :     30
OPTION02  :     50
OPTION03  :     65
OPTION04  :     50
OPTION05  :     40
OPTION06  :     40
OPTION07  :      0
OPTION08  :      0
OPTION09  :      0
OPTION10  :      0
OPTION11  :      0
OPTION12  :      0
OPTION13  :      0
OPTION14  :      0
OPTION15  :      0
OPTION16  :      0
OPTION17  :      0
OPTION18  :      0
OPTION19  :      0
OPTION20  :      0
HOLD      :FIXED
DIRECTION:UP
SCALE     :  0.14249   0.14336
SHIFT     : 338.19  599.99  179.72
```

8-7-9 Accessing the VISION File by On-line Commands

All vision files together can be read by reading the file VIS:

```
@READ VIS                    -- read all vision files
```

The vision files can be written by sending the vision file:

```
@WRITE VIS
```

All vision files to be written are sent. When all files are sent, an empty line is sent to the CSM to indicate the end of the data.

8-7-10 Example

```
V#0=FIDUCIAL
MODE      :FID
CAMERA    :1
OBJECT    :WHITE
BINARY    :MANUAL( 98)
FILL      :NO
CUT       :NO
DISPLAY   :TEST1
WINDOW1   :ON+    ( 32, 35)(223,205)
WINDOW4   :OFF
OPTION01 :   176
OPTION02 :    10
OPTION03 :   471
OPTION04 :    10
OPTION05 :     0
OPTION06 :     0
OPTION07 :     0
OPTION08 :     0
OPTION09 :     0
OPTION10 :     0
OPTION11 :     0
OPTION12 :     0
OPTION13 :     0
OPTION14 :     0
OPTION15 :     0
OPTION16 :     0
OPTION17 :     0
OPTION18 :     0
OPTION19 :     0
OPTION20 :     0
HOLD      :MOVING
DIRECTION:DOWN
SCALE     :  0.01870   0.01892
SHIFT     :  -2.42    2.21    1.09
V#2=QFP160P
MODE      :QFP1
```

Option 13: Calibration rotation offset correction at 180° in [0.01°]
Option 14: Calibration rotation offset correction at 270° in [0.01°]
Option 15: Bend detect tolerance in %. 0= no check
Option 16: not used
Option 17-20 are not used and should be set to 0

8-7-7 Options for MODE=DISP

When the MODE field is set to DISP, the options have the following meaning:

Option 1: Maximum area in 0.01 mm2. 0= no check
Option 2: Minimum area in 0.01 mm2. 0= no check
Option 3: Shape code in 0.01
Option 4: Shape tolerance in 0.01%
Option 5: H1 standard area
Option 6: H2 standard area
Option 7: H3 standard area
Option 8: H1 dispensing timer
Option 9: H2 dispensing timer
Option 10: H3 dispensing timer
Option 11: Minimum object number
Option 12: algorithm selection (not used)
option 13-20 are not used and should be set to 0
Option 5..12 are automatically filled by the CALIB utility

8-7-8 Options for MODE=BGA

When the MODE field is set to BGA, the options have the following meaning:

Option 1: Dot diameter in 0.01 mm
Option 2: not used
Option 3: Dot pitch in 0.01 mm
Option 4: Dot pitch tolerance in 0.01%
Option 5: Vertical dot number
Option 6: Horizontal dot number
Option 7: Dot amount
Option 8-10 are not used and should be set to 0
Option 11: Calibration rotation offset correction at 0° in [0.01°]. The rotation offset, when mounting at 0° is normally determined by the rotation value in the camera Shift. Therefore this option is normally set to 0
Option 12: Calibration rotation offset correction at 90° in [0.01°]
Option 13: Calibration rotation offset correction at 180° in [0.01°]
Option 14: Calibration rotation offset correction at 270° in [0.01°]
Option 15-16 are not used and should be set to 0
Option 17: Lighting level (0-7)
Option 18-20 are nott used and should be set to 0

Option 11: Calibration rotation offset correction at 0° in [0.01°]. The rotation offset, when mounting at 0° is normally determined by the rotation value in the camera Shift. Therefore this option is normally set to 0
Option 12: Calibration rotation offset correction at 90°in [0.01°]
Option 13: Calibration rotation offset correction at 180° in [0.01°]
Option 14: Calibration rotation offset correction at 270° in [0.01°]
Option 15: Bend detect tolerance (%) 0= no check
Option 16: ***not used***
Option 17-20 are not used and should be set to 0

8-7-5 Options for MODE=QFP1

When the MODE field is set to QFP1 (Quad flat pack), the options have the following meaning:

Option 1: Lead width in 0.01 mm
Option 2: Lead width tolerance in % [1..100] 0= no check
Option 3: Lead pitch in 0.01 mm
Option 4: Lead pitch tolerance in % [1..100] 0= no check
Option 5: Vertical (Y) number of leads [1..100] 0= no check
Option 6: Horizontal (X) number of leads [1..100] 0= no check
Option 7: Lead detection line offset [-5..5]
Option 8: ***not used***
Option 9: ***not used***
Option 10: Recognition type: 0 = no multiple recognition
1 = multiple recognition
Option 11: Calibration rotation offset correction at 0° in [0.01°]. The rotation offset, when mounting at 0° is normally determined by the rotation value in the camera Shift. Therefore this option is normally set to 0
Option 12: Calibration rotation offset correction at 90° in [0.01°]
Option 13: Calibration rotation offset correction at 180° in [0.01°]
Option 14: Calibration rotation offset correction at 270° in [0.01°]
Option 15: Bend detect tolerance (%) 0= no check
Option 16: ***not used***
Option 17-20 are not used and should be set to 0

8-7-6 Options for MODE=QFP1

Option 1: Lead width in 0.01 mm
Option 2: Lead width tolerance in %. 0= no check
Option 3: Lead pitch in 0.01 mm
Option 4: Lead pitch tolerance in %. 0= no check
Option 5: Lead number [0...100] 0= no check
Option 6: Lead direction [0,1,2,3] 0= upper, 1=lower, 2=left,3=right
Option 7: Detection line offset [-5...5]
Option 8: not used
Option 9: Algorithm selection
Option 10: Recognition type: 0 = no multiple recognition
1 = multiple recognition
Option 11: Calibration rotation offset correction at 0° in [0.01°]. The rotation offset, when mounting at 0° is normally determined by the rotation value in the camera Shift. Therefore this option is normally set to 0
Option 12: Calibration rotation offset correction at 90° in [0.01°]

8-7-1 Options for MODE=BINARY

When the MODE field is set to BINARY (BAD mark), all options are ignored.

8-7-2 Options for MODE=FID

When the MODE field is set to FID (fiducial), the options have the following meaning:

Option 1: fiducial area in 0.01 mm^2
Option 2: fiducial area tolerance in % [1..100] 0= no area check
Option 3: fiducial outline peripheral length in 0.01 mm
Option 4: fiducial outline peripheral length tolerance in % [1..100] 0= no check
Option 5: cursor lenght [0...50] pixel. if 0 then default 8 pixels
Option 6..20 are not used and should be set to 0.

8-7-3 Options for MODE=QUAD

When the MODE field is set to QUAD (PLCC), the options have the following meaning:

Option 1: Area in 0.01 mm^2
Option 2: Area tolerance in % [1..100] 0= no area check
Option 3: Perimeter in 0.01 mm
Option 4: Perimeter tolerance in % [1..100] 0= no check
Option 5-8 are not used and should be set to 0
Option 9: Algorithm selection [0..2]
0: use method least square
1: use half-conversion of 0.5 degree unit
2: use half-conversion of 1 degree unit
Option 10: Recognition type: 0 = no multiple recognition
1 = multiple recognition
Option 11: Calibration rotation offset correction at 0° in [0.01°]
The rotation offset, when mounting at 0° is normally determined by the rotation value in the camera Shift. Therefore this option is normally set to 0
Option 12: Calibration rotation offset correction at 90° in [0.01°]
Option 13: Calibration rotation offset correction at 180° in [0.01°]
Option 14: Calibration rotation offset correction at 270° in [0.01°]
Option 15-20 are not used and should be set to 0

8-7-4 Options for MODE=SOP

When the MODE field is set to SOP, the options have the following meaning:

Option 1: Lead width in 0.01 mm
Option 2: Lead width tolerance in % [1..100] 0= no check
Option 3: Lead pitch in 0.01 mm
Option 4: Lead pitch tolerance in % [1..100] 0= no check
Option 5: Vertical (Y) number of leads [1..100] 0= no check
Option 6: Horizontal (X) number of leads [1..100] 0= no check
Option 7: Lead detection line offset [-5...5]
Option 8: ***not used***
Option 9: ***not used***
Option 10: Recognition type: 0 = no multiple recognition
1 = multiple recognition

fillvalue ::= | "NO" | "1" | "2" |.....| "9" |

After the image undergoes binary processing with the BINARY function, it may, depending on the conditions, have shadowy characteristics or holes where binary processing could not be carried out cleanly. The hole FILL and area CUT functions are used to remove these.

cutvalue ::= | "NO" | "1" | "2" |.....| "9" |

After the image undergoes binary processing with the BINARY function, it may, depending on the conditions, have shadowy characteristics or holes where binary processing could not be carried out cleanly. The hole FILL and area CUT functions are used to remove these.

displaymode ::= | "INPUT0" | "INPUT1" | "EDGE0" | "EDGE1" |
| "CROSS0" | "CROSS1" | "ALL0" | "ALL1" |"TEST0"|
| "TEST1" |

Indicates what should be displayed on the monitor when measuring the object. For more information read your operating manual Section 7 Vision System.

windowdescr ::= | "ON+" (lowerleftpixel) (upperrightpixel) |
| "ON-" | "OFF" |

Describes the window within the field of view that is included for processing. WINDOW1 is used, WINDOW4 is allways set "OFF".

lowerleftpixel ::= pixel
upperrightpixel ::= pixel
pixel ::= "(" pix(X) "," pix(Y) ")"
pix(X) ::= 0..255
pix(y) ::= 0..239

holdvalue ::= "MOVING" | "FIXED"

This specifies the type of camera, either a moving or a fixed one.

cameradirection ::= "UP" | "DOWN"

Specifies the direction in which the camera looks. This is needed to determine how the measured offsets relate to the machine coordinate system.

camerascaling ::=
x-scale y-scale
x-scale ::= scale
y-scale ::= scale
scale ::= realnumber

Specifies the size of a camera pixel in the area of measurement in mm/pixel.

x-shift ::= **realnumber**
y-shift ::= **realnumber**

the x- and y-shift give the last measured x- and y- shift in mm.

r-shift ::= **realnumber**

the r-shift gives the last measured rotationaly shift in degrees.

The meaning of OPTION01 .. OPTION20 depends on the selected algorithm (MODE).

```
OPTION11 :integernumber
OPTION12 :integernumber
OPTION13 :integernumber
OPTION14 :integernumber
OPTION15 :integernumber
OPTION16 :integernumber
OPTION17 :integernumber
OPTION18 :integernumber
OPTION19 :integernumber
OPTION20 :integernumber
HOLD     :holdvalue
DIRECTION:cameradirection
SCALE    :camerascaling
SHIFT    : x-shift y-shift r-shift
```

where

filenr ::= `0..29`

The vision file number as used in other parts of UFOS (see board and feeder files).

name ::= `string of 1..8 characters`

This field is meant as a kind of comment field. Through the name here, you can easily remember what this vision file is meant for.

mode ::= `| "BINARY" | "FID" | "QUAD" | "SOP" | "QFP1" | "CON" | "DISP" | "BGA" |`

This field specifies the vision algorithm to be used. FID stands for fiducial marks, BINARY for bad marks, QUAD for a PLCC, CON for connector, DISP for dispenser (dot recognition), BGA for BGA components.

cameranr ::= `| "1" | "2" | "3" | "4" |`

Specifies the camera to be used for measuring the object.
1 is the camera on the beam
2 is the fixed camera
3 is the fixed camera
4 is the fixed camera

objcolour ::= `| "WHITE" | "BLACK" |`

The color of the object body to be measured, eg. a reflective mark is WHITE, the contour of a component is BLACK.

binvalue ::= `| "AUTO" | "MANUAL" | "AREA" | "+1" | "-1" | "+10" | "-10" |`

This selects the method by which the binary threshold value will be decided. There are 3 methods and numeric data that is used with each one.

```
$HEAD
NO  QTY. ERR1   RATE1  ERR2  RATE2
 1    0  0  0.00%    0  0.00%
 2   11  2 81.82%    0 100.00%
 3    0  0  0.00%    0  0.00%
$F19
H1=    0  0  0
H2=    0  0  0
H3=    0  0  0
$F22
H1=    0  0  0
H2=    5  0  0
H3=    0  0  0
$F40
H1=    0  0  0
H2=    6  2  0
H3=    0  0  0
```

8-6 The Vision File

8-6-1 Introduction

When a CSM machine is equiped with a vision system, then this vision system needs data about the way it should measure fiducial points or components. This data is stored in the VISion file.

The vision file contains a block of data for each type of fiducial or component.

VIS-file ::= { vision-file }

8-6-2 The Vision Files

The vision description, that describes one type of component or fiducial, contains the following data:

```
V#filenr=name
MODE      :mode
CAMERA    :cameranr
OBJECT    :objcolour
BINARY    :binvalue
FILL      :fillvalue
CUT       :cutvalue
DISPLAY   :displaymode
WINDOW1   :windowdescriptor
WINDOW4   :windowdescriptor
OPTION01  :integernumber
OPTION02  :integernumber
OPTION03  :integernumber
OPTION04  :integernumber
OPTION05  :integernumber
OPTION06  :integernumber
OPTION07  :integernumber
OPTION08  :integernumber
OPTION09  :integernumber
OPTION10  :integernumber
```

8-8 On-line commands

8-8-1 Introduction

This chapter discusses a set of commands that can be give through the serial line. This is only possible when the CSM is set in the CMU MODE = ONLINE (see machine parameter &H.CMU=0).

The commands can be grouped into the following groups:

- Key Operation commands
- Data Handling commands
- Utility commands
- Robot commands
- Error Messages

The robot commands are not discussed.

8-8-2 Key Operation Commands

Key operation commands are used to bring the robot in the various operational states. The commands are:

- MAIN
- PCBRUN
- ORIGIN
- RESET
- RUN
- STOP
- REMOTE
- ERROR CLEAR

8-8-2-1 Main

When the MAIN command is issued, the 'MAIN MENU' is displayed on the VDU. The command has the same effect as pressing the "Main Menu" key on the control keyboard.

Precondition: robot not in running mode
Syntax: @MAIN
Response: OK
141:CANNOT EXECUTE. ROBOT IS RUNNING

8-8-2-2 PCBRUN

When the PCBRUN command is issued, the 'SELECT JOB' menu is displayed on the VDU. When this is the first time that the PCBRUN command is given since a SWI command, the new selected program is compiled. This includes initialisation of the TRP file.

Precondition: robot not in running mode
Syntax: @PCBRUN
Response: OK
141:CANNOT EXECUTE. ROBOT IS RUNNING
74:UNDEFINED PCB NAME
Remark: You can only be sure about what board file will be selected if you did a SWItch command before. If no board file is present, the error 74 will appear, otherwise the last selected (or if deleted meanwhile, then the first) board file is selected.

M.I.S. MODE
OPTION MODE

Remark: The responses shown below correspond to the menu names of the menus on the VDU.

Read error message

The ?MSG command returns the error message displayed on the VDU. If no error exists, OK is returned.

Precondition: None
Syntax: @? MSG
Response: OK
999:@@@@@@@@@@
Remark: 999 stands for any number, @@@ for a string of text.

Read robot version number

Precondition: None
Syntax: @? VER
Response: CSM60V,NONE Z,SERVO R V3.71

8-8-3-2 Read

With the READ command you can read data files of the CSM. Various files exist, these are:

	VIS	-- vision file
	PRD	-- all MIS data (3 subfiles)
	PCB	-- all cumulative PCB data of MIS
	TRP	-- trip data of MIS
	MCH	-- machine configuration file
	FDR	-- feeder file
	BRD	-- all board files together
	B<pcb-name>	-- single board file
	B<>	-- current selected board file
*	PGM	-- program data (COMMON)
*	MEM	-- memory image
	DIn	-- Digital inputs of port n
	DOn	-- Digital outputs of port n
*	SFT	-- shift data
*	LCK	-- lock data
*	VAR	-- variables (ALL)
*	ARY	-- array variables (ALL)
*	DIR	-- directory of programs (PGM)
	BDIR	-- directory of board files

8-8-2-7 Remote

The remote command lets you disable or enable the control keyboard.

Precondition: None
Syntax: @REMOTE ON -- disable
@REMOTE OFF -- enable
Response: OK

8-8-2-8 Error Clear

The error clear command is at this moment not implemented. However you can use the following command to generate the error clear:

Precondition: The yellow lamp an the buzzer is on
Syntax: @DO0= 0
Response: OK
101:FORMAT ERROR

8-8-3 Data Handling Commands

Through data handling commands you can read data from and write data to the CSM.

The avialable data handling commands are:

- ?
- READ
- WRITE
- SPECIAL CODES

8-8-3-1 The ? Command

Through the ? command, some status data can be obtained about the robot. Various forms of the command exists:

Request position of servo

Precondition: None
Syntax: @? PXY
Response: X= 0.00 Y= 0.00 R= 0.00

Request status of robot

Precondition: None
Syntax: @? MOD
Response: MAIN MENU
RUNNING MODE
RUNNING MODE---RUNNING
MANUAL MODE
DATA IN MODE

8-8-2-3 Origin

When this command is issued, the robot servo is returning to its origin. When it arrives at the origin it responds with it's position in some undefined quantity.

Precondition: Main menu displayed and origin incomplete
Syntax: @ORIGIN
Response: COMPLETED
55% 46% 52%
10:CANNOT EXECUTE
Remark: note the space(s) at the beginning of the sucess response; the actual value of the numbers may differ

The command will result in COMPLETED when the origin command has already been executed before. No physical action will be taken by the machine in this situation.

8-8-2-4 Reset

The execution of the current program is reset to it's beginning. I/O ports are set to their default value (as with power-up of the system).

Precondition: Select Job menu displayed
Syntax: @RESET
Response: OK
141:CANNOT EXECUTE. ROBOT IS RUNNING
10:CANNOT EXECUTE

8-8-2-5 Run

The robot continues executing the selected program, in this case the selected board file. It proceeds from the point where it previously stopped (or at the beginning when a new one is selected using the SWI program).

Precondition: Select Job menu displayed
Syntax: @RUN
Response: OK
10:CANNOT EXECUTE
Remark: the RUN command can also be given in other modes, but it will have no effect, though OK is responded.

8-8-2-6 Stop

The robot stops execution of the current program immediately. This means that any active outputs remain active. So when the glue is flowing out of the dispense head, it will continue flowing out.

Precondition: None
Syntax: @STOP
Response: OK

8-8-4-1 Init

The init commands initialises data. It can be performed on the files:

- MEM
- PRD
- TRP

Precondition: Main menu must be displayed
Syntax: @INIT filename
Response: OK

Example: @INIT TRP

8-8-4-2 SWI

With the SWI command you can select a board file for execution. Before a new board file can be selected, a RESET command is required to free the old board file.

Precondition: Main menu displayed
Syntax: @SWI B<**filename**>
Response: OK
69:NO SUCH PCB
10:CANNOT EXECUTE

8-8-4-3 ERA

With the ERA command, files can be deleted or cleared (depending on the file).

Precondition: Main menu displayed
Syntax: @ERA **filename**
Response: OK
78:CANNOT DELETE. PLEASE RESET

Files that can be erased:

B<pcbname>
<progname>

Note: When you erase a B<xxxx> file then the MIS data for the PCB type (xxxx) is also deleted.

8-8-4-4 ATTR

The PGM memory contains a number of programs. One of these is COMMON. Through the ATTR command they can be made write protected or not.

Precondition: Main menu must be displayed
Syntax: @ATTR <**filename**> TO RW
@ATTR <**filename**> TO RO
Response: OK

When a file is read, the file data is sent, followed by an empty line. When a compound file is sent, like PRD, each subfile is ended by an empty line (no additional empty line after last subfile!).

The files marked with an astrix (*) can only be accessed when the machine is configured in debug mode (dipswitch 2-6).

Precondition: None, when used in running mode it may interrupt normal functioning of the program since the CMU has priority over program execution.
Syntax: @READ **file-name**
Response: **file data** followed by empty line

8-8-3-3 Write

With the write command you can write data to files on the CSM.

Precondition: None, when used in running mode it may interrupt normal functioning of the program since the CMU has priority over program execution.
Syntax: @WRITE **file-name**
Response CSM: *** PLEASE ENTER !
You send: **file-data**
Response CSM: OK

In error situations the CSM responds as soon as it detects the error with an error message: 999:@@@@@@@@.

file data followed by empty line

8-8-3-4 Special Codes

You can use two special codes. These codes are:

[ctrl Y]: This initializes the reception buffer of the CSM. It is used when online commands have been input up to a certain point but the user wants to cancel them.

[ctrl C]: This interrupts the online command presently being run. If the CSM is in operation, this command also stops the operation.
When an online command is interrupted, the following response is displayed:
*** ABORTED c/r
--- ENDs normally

8-8-4 Utility Commands

Utility commands let you perform various functions. They are:

- INIT
- SWI
- ERA
- ATTR
- REN

8-8-4-5 REN

With the REName command, files in the PGM memory can be given a new name. It can only be used when the debug mode is selected with the dip switches (switch 2-6).

Precondition: Main menu must be displayed
Syntax: @REN <oldfilename> TO <newname>
Response: OK

8-8-4-6 COPY

With the Copy command, files in the PGM memory can be copied to another file or point data lines can be copied. It can only be used when the debug mode is selected with the dip switches (switch 2-6).

Precondition: Main menu must be displayed
Syntax: @COPY <origname> TO <newname>
Response: OK

8-8-4 Error Messages

When the parameter (in the CSM) MSG OUT TO CMU = YES (see machine parameter &H.MOUT=1) an the CMU MODE = ONLINE (see machine parameter &H.CMU=0) then the following error messages are send to the HOST port.

#001: COMPLETED (mounting operation has ended)
#002: MACH. ERR (yellow lamp is burning, buzzer is active)
#010: PICK UP ERR
#020: FIDUCIAL ERR
#030: VISION ERR