

**OB1 - <offline>**

"Cycle Execution"

**Name:****Family:****Author:****Version:** 0.1**Block version:** 2**Time stamp Code:**

07/14/2010 03:50:41 PM

**Interface:**

01/06/2004 11:26:53 AM

**Lengths (block/logic/data):** 00678 00558 00034

Name	Data Type	Address	Comment
TEMP		0.0	
OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
OB1_PRIORITY	Byte	2.0	Priority of OB Execution
OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
OB1_RESERVED_1	Byte	4.0	Reserved for system
OB1_RESERVED_2	Byte	5.0	Reserved for system
OB1_PREV_CYCLE	Int	6.0	Cycle time of previous OB1 scan (milliseconds)
OB1_MIN_CYCLE	Int	8.0	Minimum cycle time of OB1 (milliseconds)
OB1_MAX_CYCLE	Int	10.0	Maximum cycle time of OB1 (milliseconds)
OB1_DATE_TIME	Date_And_Time	12.0	Date and time OB1 started

**Block: OB1 "Read/Write MP85ADP"**

Use SFC 14 to read measurement values consistently

Network: 1 Read gross value X (status included)

Read from peripheral device to MD50 with SFC 14

```
CALL "DPRD_DAT"          //Call SFC14                      SFC14          -- Read data cons
                           //                               istently
LADDR :=W#16#100          //read from HW address 256dec -> 100hex
RET_VAL:=MW200            //Address if Error occurs
RECORD :=P#M 50.0 BYTE 6  //Store the data to the memory 50, 6 Byte
```

Network: 2 Read gross value Y (status included)

Read from peripheral device to MD50 with SFC 14

```
CALL "DPRD_DAT"          //Call SFC14                      SFC14          -- Read data cons
                           //                               istently
LADDR :=W#16#106          //read from HW address 262dec -> 106hex
RET_VAL:=MW201            //Address if Error occurs
RECORD :=P#M 60.0 BYTE 6  //Store the data to the memory 60, 6 Byte
```

Network: 3 Write control word

Write to the peripheral device from flag double word 70

```
CALL "DPWR_DAT"          //Call SFC15                      SFC15          -- Write data con
                           //                               sistently
LADDR :=W#16#100          //write control word to 256dez -> 100hex
RECORD :=P#M 70.0 BYTE 4  //load 32 Bit pattern (control word 1 a. 2 ) from memory, 4 byte
RET_VAL:=MW202            //Address if Error occurs
```

Network: 4 Read Workpiece\_ID acyclic via SFB52, DPV1

Read data via DPV1

```
CALL "RDREC" , DB52          SFB52          -- Read a Process
                           Data Record
REQ :=M15.1                //activate the SFB52 function box, set the 15.1 flag
ID :=DW#16#10C             //real address from particular HW-Config slot (Process-Satus 268dez -> 10C hex)
```

```
INDEX :=40           //Index from object dictionary, Process_Stat = 28h -> 40dez = read workpiece_
                      name
MLEN  :=59           //data length, visible string 59 character = 59 bytes
VALID :=DB52.DBX10.0
BUSY  :=DB52.DBX10.1
ERROR :=DB52.DBX10.2
STATUS:=DB52.DBD12
LEN   :=DB52.DBW16
RECORD:=P#M 80.0 BYTE 59 //write to flag word 80, 59 byte
```

Network: 5	Write Workpiece_ID acyclic via SFB53, DPV1
Write data via DPV1	

```
CALL  "WRREC" , DB53           SFB53           -- Write a Proces
                                   s Data Record
REQ   :=M15.2                 //activate the SFB53 function box, set the 15.2 flag
ID    :=DW#16#10C             //real address from particular HW-Config slot 268dec -> 10Chex
INDEX :=127                   //Index from object dictionary, Process_Stat = 7Fh = write Workpiece name
LEN   :=59
DONE  :=DB53.DBX10.0
BUSY  :=DB53.DBX10.1
ERROR :=DB53.DBX10.2
STATUS:=DB53.DBD16
RECORD:=P#M 140.0 BYTE 59 //write to flag word 140, 59 byte

BE
```