

S7 Handling Blocks for R355, R6000 and R2700

1 FB90, Zyklus_R/W

This block enables central communication between the application program and the R355 module, as well as R6000 and R2700 devices, via Profibus DP.

Attention: FB90 necessitates FC90 internally.

Call:

The module must be called cyclically in OB1, or in the OB1call level.

Call for R355 or R6000:

```
CALL  "ZYKLUS_R/W" , "inst DB"    // R355 or R6000
  Adr      :=256
  DB_channel1 := "DB Kanal 1"
  DB_channel2 := "DB Kanal 2"
  DB_channel3 := "DB Kanal 3"
  DB_channel4 := "DB Kanal 4"
  DB_channel5 := "DB Kanal 5"
  DB_channel6 := "DB Kanal 6"
  DB_channel7 := "DB Kanal 7"
  DB_channel8 := "DB Kanal 8"
  DB_Device  := "DB FM355"
  DB_Parameters := "DB_Param_wld"
  DP_mode    := FALSE
  reset      := M10.0           // First cycle flag (OB100)
  param_automwrite:=TRUE
  read_all_blocks:=FALSE
  ready      := M10.1
```

Call for R2700:

```
CALL  "ZYKLUS_R/W" , "inst DB"    // R2700
  Adr      :=256
  DB_channel1 := "DB R27 Ch1"     // Channel 1 only
  DB_channel2 :=
  DB_channel3 := "DB R27 Prog"   // Current program
  DB_channel4 :=
  DB_channel5 :=
  DB_channel6 :=
  DB_channel7 :=
  DB_channel8 :=
  DB_Device  := "DB R27 Dev"     // No parameter DB !!!
  DB_Parameters :=
  DP_mode    := FALSE
  reset      := M10.0           // First cycle flag (OB100)
  param_automwrite:=FALSE
  read_all_blocks:=TRUE
  ready      := M10.1           // Cycle mode
```

Parameters:

Adr	Periphery address of the module, input and output identical!
DB_channel1...8	Channel DBs: Each controller channel has its own data block. This block contains channel-specific data, for example setpoint, actual value, status, controller function etc. This involves up to eight channels for the R355 and the R6000, and only one for the R2700. If the R2700 is used as a program controller, the DB for the current program must be parametrized at channel 3. If any channels are unused, the respective DBs do not have to be parametrized. The corresponding DB in the controller is not necessary in this case.
DB_Device	Controller DB: There is exactly one DB of this type for each controller module or device. It contains device-specific data such as error status, acknowledge bits etc.
DB_Parameters	DB_Param_wld: This DB contains complete module parametrization for the R355 and the R6000, which is generated with a parametrizing tool and transmitted to the PLC. This allows for complete, automatic parametrizing of the new module in the event that a module is replaced. This DB does not exist for the R2700 because, as opposed to the R355 and the R6000, complete parametrization is included in the other DBs.
DP_mode	false: When the module is plugged in locally to the CPU's backplane bus. true: When the module is connected to the Profibus (VIPA IM 353).
reset	Start-up bit: This bit must be set to true for one cycle when the CPU is started up, e.g. by setting it in OB100. This causes initialization of the communication FB and the associated data blocks. After calling "ZYKLUS_R/W", the bit must be cleared.
param_automwrite	true: Parametrization from DB_Param_wld is automatically loaded to any replaced module in the event of inequality. Module replacement without PG is thus made possible. false: Parametrization is not transmitted automatically. This function is controlled depending upon the parameter set ID, i.e. parametrization contains a time stamp which is compared between the module and the PLC. In the case of inequality, parametrization is transmitted to the module. (Always false for the R2700, because there is no DB_Param_wld.)
read_all_blocks	true: When communication with the module is started up, all of the data blocks from the module are loaded one time only, i.e. the start-up data always come from the module. false: Start-up data are taken from the DBs.
ready (out)	false: Communication is still being initialized. This may take several seconds. true: Communication has been fully initialized, i.e. the controller is now ready for use.

Internal Parameters:

param_DB_write_req	true: Can be used to request transmission of complete controller parametrization from DB_Param_wld to the module. false: Transmission has not been requested, or its has been completed.
param_DB_read_req	true: Can be used to request transmission of complete controller parametrizing from the module to DB_Param_wld. false: Transmission has not been requested, or it has been completed.

These functions can be requested at any desired point in time. However, this is only sensible during start-up.

Transmission of parametrization can also be triggered from the start-up tool. In this case, DB_Param_wld is automatically overwritten with current controller parametrization.

2 Channel DBs and Controller DB

Shadow mode structure:

DBB	0 ... 9	Internal administration data
DBW	10 ... 9 + 8*n	Data range including n data blocks, $n \leq 16$
DBW 10 + 8*n ... 9 + 16*n		Shadow range with a copy of the data

Administration data are required for functionality of the scanner, and may not be changed by the user.

Only the data range is intended for the user, which includes, for example, setpoints and actual values.

The "actual values" are transmitted automatically from the controller to the control system when they change, and are always up-to-date in the data range.

"Setpoints" are simply written to the data range by the user, after which they are detected by the scanner as changed and are made available for transmission to the controller.

Dirty mode structure:

DBB	0 ... 9	Internal administration data
DBW	10 ... 9 + 8*n	Data range including n data blocks, $n \leq 16$

In the dirty mode the shadow range is omitted, and the DB is thus shorter. This mode is advisable when memory capacity in the CPU is limited.

Advantages and disadvantages of the shadow mode and the dirty mode

Shadow:	Advantage: Setpoints are transmitted automatically when changed. Disadvantage: More memory capacity is required, data range is required twice. Disadvantage: Increased cycle load because scanning (comparison) of the data for changes takes time (irrelevant for fast CPUs).
Dirty:	Advantage: Only minimal memory capacity is required by the DBs. Advantage: Minimal cycle load. Disadvantage: When a "setpoint" is changed, the respective shadow bit must be set, and this function has to be executed by the application program.

3 FC90, DB_Scanner

This block is required internally by FB90. It's used to initialize channel and controller DBs.

During start-up (initialization), all "dirty bits" are set in all of the DBs (except for DB_Param_wld), and "shadow data" are set to false. This serves to flag the data blocks as dirty, i.e. to flag them for transmission.

During normal operation, the scanner checks to see if blocks need to be transmitted and triggers transmission if necessary.

The data blocks are thus only transmitted to the controller if necessary. Transmission only takes place when data are changed, or when the corresponding dirty bit is set in the dirty mode.

4 FC91, FM355_write_Outputs

This block is used for direct read-out to the R355 module, as well as to R6000 and R2700 devices, via Profibus DP.

Available binary outputs at the R355 and the R6000 must thus be controlled directly, and the setpoint can be set at the R2700.

Call:

The function is called as required or cyclically before "ZYKLUS_R/W".

Example for R355 and R6000:

```
L      W#16#36C9           // Binary pattern only
T      #W0                 // Out 9...16, 1...8
L      W#16#36C9
T      #W2                 // Out (25...32), 17...24
CALL   "FM355_write_Outputs" // Write available
      FM355_Inst_DB:="inst DB" // binary outputs
      Set_W0          :=#W0
      Set_W2          :=#W2
```

Example for R2700:

```
L      49                 // Setpoint = 49° or 4.9°
T      #SP                // Int => word
TAW
T      #W0
CALL   "FM355_write_Outputs" // Set setpoint directly
      FM355_Inst_DB:="inst DB"
      Set_W0          :=#W0
      Set_W2          :=W#16#AA54 // Write code = W#16#AA55
```

5 FC92, FM355_read_Inputs

This block is used for direct read-in of the R355 module, as well as R6000 and R2700 devices, via Profibus DP.

The statuses of the binary inputs are read in from the R355 and the R6000, and current control and manipulated variables are read in from the R2700.

Call:

The function is called as required or cyclically after "ZYKLUS_R/W".

Example for R355 and R6000:

```
CALL  "FM355_read_Inputs"           // Get binary inputs
      FM355_Inst_DB:="inst DB"
      Get_W0           :=#W0         // In 9...16, 1...8
      Get_W2           :=#W2         // In (25...32), 17...24
```

Example for R2700:

```
CALL  "FM355_read_Inputs"           // Get values
      FM355_Inst_DB:="inst DB"
      Get_W0           :=#W0
      Get_W2           :=#W2
      L      #W0           // Controlled variable
      TAW
      T      #PV
      L      #W2           // Manipulated variable
      TAW
      T      #ED
```