

**BALLUFF**

# **Software- Description**

**BIS M-4008**

**Function Block S7-300/400**



## **CONTEND**

<b>1</b>	<b>INTRODUCTION.....</b>	<b>2</b>
1.1	General Data .....	2
1.2	Recommendations of FB invoke.....	2
<b>2</b>	<b>COMMISSIONING .....</b>	<b>3</b>
2.1	DB parameter.....	3
2.2	Device parameter for BIS M-4008 .....	3
<b>3</b>	<b>FB PARAMETER DESCRIPTION.....</b>	<b>4</b>
3.1	FB illustration S7-300/400 .....	4
3.2	Input parameter .....	4
3.3	Output parameter .....	6
3.4	General error codes .....	6
3.5	FB internal error codes .....	7
3.6	Description of commandos.....	8
<b>4</b>	<b>DIACLAIMER OF LIABILITY .....</b>	<b>9</b>

### 1 INTRODUCTION

This function block is an example for the communication with an M-4008 processor unit. The example is free of charge. This function block enables a communication between Balluff - BIS M-4008 processor and a Simatic® S7-300/400 PLC.

The following commandos are supported:

- Read data carrier
- Write data carrier
- Store Auto Read start address
- Type and serial number
- Initialize CRC\_16 data check
- Write constant value to data carrier

For each read/write head the function block has to be called with an own instance data block.

#### ATTENTION

Please test carefully if the used commandos are supported by the BIS processor and the read/write head!

If an existing function block is replaced by this, the instance data block has to be re-generated, because the static variables have been extended.

#### 1.1 General Data

Function block name:	FB31
Instance data block:	(an own instance data block has to be setup for each antenna or r/w head)
Invoked blocks:	SFB4 TON
Reserved memory bits:	none
Reserved Timers:	none
Reserved Counters:	none
I/O length:	16 - 254 byte
Invoke:	absolute
Device compatibility:	Siemens Simatic® S7 300/400 with SIMATIC Manager

#### 1.2 Recommendations of FB invoke

The function block should be called only once for each read/write head. Multiple calls of the function lock at the same time are not allowed.

If the function block is conditionally called and the calling condition is false before FB sets it **Ready** output, the **Init** input have to be set.

If the PLC restarts the **Init** input have to be set for one cycle. FB parameters could be attached dynamic if necessary.

## 2 COMMISSIONING

The I/O data length of the processor unit depends on the available in-/outputs in the PLC (maximum 254 bytes). Modules of the same length have to be used for process data input and output.

### 2.1 DB parameter

The maximum read/write data length of the function block is 32,767 bytes. The data blocks for transmitting and receiving data have to be adjusted in size according to the parameters **Offset\_DBSend**, **Offset\_DBReceive** and **TAG\_NumbOfByte**.

### 2.2 Device parameter for BIS M-4008

Same values for the parameters **Dynamic Mode**, the **I/O start address** and the **IO length** have to be set in HW configuration and for FB call!

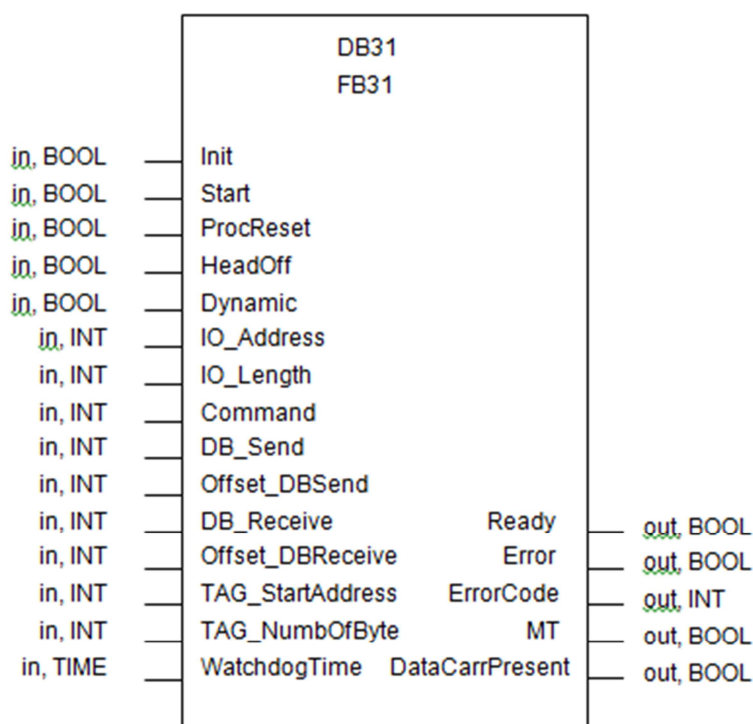
#### RFID Head Parameter:

<b>CRC</b>	disable	=	Not active
	enable	=	The CRC check is a procedure for determining a check value for data in order to be able to recognize transmission errors.
<b>Dynamic Mode</b>	disable	=	Dynamic mode deactivated. Error number 1 is applied if job is started without tag
	enable	=	The processor unit accepts the job and stores it regardless of whether a tag is in the active zone or not. If the tag enters the active zone the stored job is run.
<b>Type and Serial Number</b>	disable	=	First bytes of data carrier are displayed.
	enable	=	the type of the read/write head and serial number are displayed when CP occurs
<b>Slow Tag Detection</b>	disable	=	Default, fast tag detection
	enable	=	The antenna is switched on for tag detection only every 200ms

More information concerning the parameters is available in the device manual.

### 3 FB PARAMETER DESCRIPTION

#### 3.1 FB illustration S7-300/400



#### 3.2 Input parameter

<b>Init</b>	Function block initialization Must be set for one cycle each time the PLC is restarted. Static variables, control bits and upcoming commands are reset. The function is done when <b>Ready</b> is set again.
<b>Start</b>	Start function <b>Start</b> = 1 starts a job. This signal must be set until the Ready output goes to 0. The function is done when <b>Ready</b> or <b>Error</b> is set again.
<b>ProcReset</b>	Reset function block and r/w head <b>ProcReset</b> = 1 sets the function block and ground state r/w head. The signal must be set until the Ready parameter goes to false. The function is done when Ready is set again.  <b>The processor unit needs a few seconds for a reset sequence!</b>
<b>HeadOff</b>	Head power is switched off.
<b>Dynamic</b>	Activation or deactivation of the WatchdogTime for dynamic mode  Dynamic = 0: Watchdog Timer is on Dynamic = 1: Watchdog Timer is off  If the Dynamic Mode is used, the watchdog timer has to be disabled.

## Function Block description for BIS M-4008

### 3 FB PARAMETER DESCRIPTION

<b>IO_Address</b>	Start address of the in-/ output range of the PLC. The address may lie in the normal I/O range of the PLC or in the peripheral range.
<b>IO_Length</b>	Length of the in-/ output range, specified in byte. The value has to be set assuming parameterization in HW configurator.
<b>Command</b>	General Job type. Job type in hex format.  Command = 01 <sub>hex</sub> : Read data carrier Command = 02 <sub>hex</sub> : Write data carrier Command = 07 <sub>hex</sub> : Store start address for "Auto Read" Command = 09 <sub>hex</sub> : Type and serial number Command = 12 <sub>hex</sub> : Initialize CRC_16 data check Command = 32 <sub>hex</sub> : Write constant value
<b>DB_Send</b>	Data block for write data
<b>Offset_DBSend</b>	Start address for write data in the data block
<b>DB_Receive</b>	Data block for read data
<b>Offset_DBReceive</b>	Start address of read data in the data block
<b>TAG_StartAddr</b>	Write/read start address in the data carrier or parameter number
<b>TAG_NumbOfByte</b>	Length in the code tag for read or write procedures. For the jobs 1 (read), 2 (write), 12 (CRC Init), 32 write constant value, length have to be set there.
<b>WatchdogTime</b>	Monitoring timer for commands

### 3 FB PARAMETER DESCRIPTION

#### 3.3 Output parameter

<b>Ready</b>	Job completed This bit is set when the job was completed. This output will be reset by a rising edge of <b>Start</b> or <b>Reset</b> input.
<b>Error</b>	Job completed with error This bit is set if the job was completed with an error and is reset with a rising edge at <b>Reset</b> or <b>Start</b> input.
<b>ErrorCode</b>	If the <b>Error</b> bit is set, the error number will be displayed here as hex value.
<b>MT</b>	Notification <b>Multiple Tags</b> are detected, more than one data carrier in range of antenna.
<b>DatCarrPresent</b>	Data carrier present / data valid. This bit is only true when a tag is activated from processor. A positive edge of the codetag present signal means data are available starting with address 0 of the code tag in the input buffer of the instance data module without requiring that a read request be initiated. The length of data to be read, is the buffer size of read/write head minus 2 (bit headers).

#### 3.4 General error codes

Error No.	Meaning	Effect	Remedy
00 <sub>hex</sub>	No error		
01 <sub>hex</sub>	No data carrier in range of antenna	depends on parameter <b>Dyn-amic</b>	Check distance between code tag and read/write head
02 <sub>hex</sub>	Read error	Command canceled. Processor- and FB go to base state.	Check distance between code tag and read/write head
03 <sub>hex</sub>	Read canceled because data carrier was removed	Command canceled. Processor- and FB go to base state.	Check distance between code tag and read/write head. and read/write head. For dynamic mode: Check velocity
04 <sub>hex</sub>	Write error	Command canceled. Processor- and FB go to base state Command <b>Caution:</b> Some data may have already been written to the code tag	Check distance between code tag and read/write head
05 <sub>hex</sub>	Write canceled because data carrier was removed	Command canceled. Processor- and FB go to base state Command <b>Caution:</b> Some data may have already been written to the code tag	Check distance between code tag and read/write head. and read/write head. For dynamic mode: Check velocity

### 3 FB PARAMETER DESCRIPTION

Error No.	Meaning	Effect	Remedy
07 <sub>hex</sub>	Wrong command identifier ( <b>Job</b> ) or number of bytes for read/write command is 0.	Processor- and FB go to base state.	Check parameter settings
09 <sub>hex</sub>	Cable break on selected read/write head, r/w head not connected or r/w head defective.	Command canceled. Processor- and FB go to base state.	Check r/w head
0D	Communication to the R/W head disrupted.	Command canceled. Processor- and FB go to base state.	Check r/w head
0E <sub>hex</sub>	CRC for the read data and CRC for the data carrier do not agree.	Command canceled. Processor- and FB go to base state.	Check data carrier
0F <sub>hex</sub>	Bit headers are not equal	Processor and FB go to base state.	Check programming and content of bitheaders, I/O Addresses correct.
20 <sub>hex</sub>	Address assignment of the read/write job is outside the memory range of the data carrier.	Processor and FB go to base state.	Check programming
21 <sub>hex</sub>	This function is not possible for this data carrier	Processor and FB go to base state.	Check data carrier

This error list is corresponding to the status codes in user's manual.

#### 3.5 FB internal error codes

Error No.	Meaning	Effect	Remedy
30 <sub>hex</sub>	Monitoring time expired	Processor and FB go to base state	Check programming
31 <sub>hex</sub>	Undefined command	Processor and FB go to base state	Check programming
32 <sub>hex</sub>	Wrong Index for select EPC command	Processor and FB go to base state	Check programming Limit between 1 and 25



### 3 FB PARAMETER DESCRIPTION

#### 3.6 Description of commandos

The commandos are selected by a hexadecimal value at the “**Command**” input. With a rising edge at “**Start**” input the commando is executed. After successful execution the “**Ready**” Output is true and “**Error**” is false.

##### Read data carrier 01<sub>hex</sub>:

Data carrier is read from “**TAG\_StartAddr**”. The amount of data is defined by the parameter “**TAG\_NumbOfByte**”. The data are stored in “**DB\_Receive**”. The byte offset is defined by the parameter “**Offset\_DBReceive**”.

##### Write data carrier 02<sub>hex</sub>:

The data are read out of “**DB\_Send**” and written to carrier at the byte “**TAG\_StartAddress**”. The amount of data is defined by the parameter “**TAG\_NumbOfByte**”.

##### Store start address for “Auto Read” function 07<sub>hex</sub>:

The “Auto Read” start address is stored in the processor EEPROM. The “Auto Read” start address is defined by the parameter “**TAG\_StartAddress**”.

##### Type and serial number 09<sub>hex</sub>:

Read the read/write head type, data carrier type and UID of data carrier in the field. The dates are stored in “**DB\_Receive**”. The ReadByte [0] contains the UID length information, ReadByte [1] contains the head type, ReadByte [2] contains the data carrier type, ReadByte [3..n] contains the UID. The amount of bytes stored in ReadByte[0] are transferred to data-block.

##### Initialize CRC 16 data check 12<sub>hex</sub>:

The data are read out of “**DB\_Send**” and written to carrier at the byte “**TAG\_StartAddress**” with CRC\_16 checksum. The amount of data is defined by the parameter “**TAG\_NumbOfByte**”. Parameter CRC must be selected for data carrier initialization, otherwise the command behaves the same as 02<sub>hex</sub> write data carrier.

##### Write constant value to data carrier 32<sub>hex</sub>:

Constant values are written to data carrier the parameter for start byte on tag is “**TAG\_StartAddress**”. The amount of constant values is defined by the parameter “**TAG\_NumbOfByte**”. The data byte for constant value is read from “**DB\_Send**”.

Please refer to the device manual for further information.

### 4 DIACLAIMER OF LIABILITY

This demo function block is free of charge and is a universal application example. This demo function block shall help program and configure PLC applications and shall provide possible solutions.

The user is not entitled to claim for warranty, error correction and updates. In particular there is excluded any claims against Balluff GmbH for damages that might result from the use of this demo program. Excluded from this limitation of liability shall be (a) those damages that are based on injury to life, limb or health, (b) a liability according to the Produkthaftungsgesetz (German Product Liability Law) and (c) cases of willful intent.

Please check if the function block is intended for your application before adapting it in plants and machineries.

By using the S7 sample, made available free of charge you accept the limitation of warranty and liability!

Balluff GmbH  
Schurwaldstraße 9  
73765 Neuhausen a.d.F.  
Deutschland  
Tel. +49 7158 173-0  
Fax +49 7158 5010  
balluff@balluff.de  
www.balluff.com