# FA Technical Report

| | |
|---|---|
| **Product:** | **IC-KP2-1HBx; IQH1-…-V1; IQC22-…** |
| **Author:** | **Karsten Reinhardt** |
| **Document version number:** | **V2.0** |
| **Date:** | **2019-10-18** |

## Description

This document describes the commissioning of the function block FB3222 inside the Siemens Step-7 environment. With the FB3222 it is possible to read or write large data volumes from the IQC22 RFID tag without complicated command parameterisations.

The function block is able to read and write the whole memory of the IQC22 (256 Byte) with 5 command cycles. The start address is W#16#0, but can changed by the user. The function block is able to read and write memory areas with a size of 60 Bytes per command cycle.

Requirements:

| | |
|---|---|
| Evaluation unit: | IC-KP2-1HBx |
| RFID head: | IQH1-…-V1 |
| RFID tag: | IQC22- |
| Software: | Function block FB32 (IDENTControl) for commissioning the IC-KP2-1HBx |

Inside the function block FB3222, the FB32 used for the command configuration and execution. In the first step, you have to configure the FB32 with the correct values and variables before starting an operation with the FB3222.

Configuration of the FB 32 “IDENTControl”

```
⊟ Netzwerk 5 : Titel:

      CALL  "IDENTControl" , "InstDB"              FB32 / DB32
       IC_INPUT_Address    :=W#16#200
       IC_OUTPUT_Address   :=W#16#200
       Length_IN           :=64
       Length_OUT          :=64
       Timeout             :=T#3S
       Head1DataFixcode    :=FALSE
       Head1SingleEnhanced:=FALSE
       Head1SpecialCommand:=FALSE
       Head1Read           :=#Head1Read             #Head1Read
       Head1Write          :=#Head1Write            #Head1Write
       Head1Quit           :=FALSE
       QuitErrorHead1      :=FALSE
       IC_Command_on_Head1:=FALSE
       Head1WordNum        :=#HeadWordNum           #HeadWordNum
       Head1WordAddress    :=#HeadWordAddress       #HeadWordAddress
       Head1TagType        :=W#16#3232
       Head1Done           :=#Head1Done             #Head1Done
       Head1NoDataCarrier :=#Head1NoDataCarrier     #Head1NoDataCarrier
       Head1Error          :=#Head1Error            #Head1Error
       Head1Busy           :=#Head1Busy             #Head1Busy
       Head1Status         :=#Head1Status           #Head1Status
       Head1ReplyCounter   :=#Head1ReplyCounter     #Head1ReplyCounter
       InitFinish          :="InitFinish"           M0.0
       SetRestart          :="SetRestart"           M0.1
```

Parameter for the I/O configuration (input parameter):

| | |
|---|---|
| IC_INPUT_Address: | Start address of the input area of the IDENTControl in hexadecimal format (e.g. W#16#200 if the input area starts at 512 in the hardware configuration) |
| IC_OUTPUT_Address: | Start address of the output area of the IDENTControl in hexadecimal format (e.g. W#16#200 if the output area starts at 512 in the hardware configuration) |
| Length_IN: | Length of the input telegram (fix at 64; Length_IN:=64) |
| Length_OUT: | Length of the output telegram (fix at 64; Length_OUT:=64) |

Parameter for command configuration (input parameter):

| | |
|---|---|
| Head1DataFixcode: | Access to data or fixcode area; permanently parameterized with false (e.g.:= FALSE) |
| Head1SingleEnhanced: | Execution of a one time or a continuous command; permanently parameterized with false (e.g.:= FALSE) |
| Head1SpecialCom: | Execution of a special command; permanently parameterized with false (e.g.:= FALSE) |
| Head1Quit: | Execution of a quit command to abort a continuous command; permanently parameterized with false (e.g.:= FALSE) |
| Head1QuitError: | Execution of an error routine to clear an error status; permanently parameterized with false (e.g.:= FALSE) |
| IC_Command_on_He..: | Execution of a special command which is send to the unit and not to one of the channels; permanently parameterized with false (e.g.:= FALSE) |
| Head1Read: | Start of the reading command execution; connect with variable #HeadXRead of the local Instanz DB (“DB_Command”) of the FB3222.<br>Head1Read := DB3222.DBX 156.0 (#Head1Read) |
| Head1Write: | Start of the writing command execution; connect with variable #HeadXWrite of the local Instanz DB (“DB_Command”) of the FB3222.<br>Head1Write := DB3222.DBX 156.1 (#Head1Write) |
| Head1WordNum: | Number of read or write data blocks each command cycle; connect with variable #HeadWordNum of the local Instanz DB (“DB_Command”) of the FB3222.<br>Head1WordNum := DB3222.DBW 154 (#HeadWordNum) |
| Head1WordAddress: | Memory address of the data carrier from which the reading or writing begins; connect with variable #HeadWordAddress of the local Instanz DB (“DB_Command”) of the FB3222.<br>Head1WordAddress := DB3222.DBW 152 (#HeadWordAddress) |
| Head1TagType: | Type of the used data carrier coded in hexadecimal format; permanently parameterized with W#16#3232 (e.g. IQC22-..)<br>Head1TagType := W#16#3232 |

Parameter for command execution information (output parameter):

| | |
|---|---|
| Head1Done: | signalize the end of a command execution (positive edge); connect with variable #HeadXDone of the local Instanz DB (“DB_Command”) of the FB3222.<br>Head1Done := DB3222.DBX 156.2 (#Head1Done) |
| Head1NoDataCarr: | signalize that no data carrier was in the detection range of the head while command execution; connect with variable #HeadXNoDataCarrier of the local Instanz DB (“DB_Command”) of the FB3222.<br>Head1NoDataCarrier := DB3222.DBX 156.5 (#Head1NoDataCarrier) |

| | |
|---|---|
| Head1Error: | signalize an error status while command execution; connect with variable #HeadXError of the local Instanz DB (“DB_Command”) of the FB3222.<br>Head1Error := DB3222.DBX 156.4 (#Head1Error) |
| Head1Busy: | signalize the current command execution; connect with variable #HeadXBusy of the local Instanz DB (“DB_Command”) of the FB3222.<br>Head1Busy := DB3222.DBX 156.3 (#Head1Busy) |
| Head1Status: | shows the actual status of the command execution; not used for the implementation of the FB3222 |
| Head1ReplyCounter: | shows the actual value of the reply counter of the unit; not used for the implementation of the FB3222 |

Parameter for the initialisation and error handling (in/out parameter):

| | |
|---|---|
| InitFinish: | Signalize (positive edge) the successfully execution of the initialisation routine; not used for the implementation of the FB3222; it should be used to program an error handling routine inside the application program after an error occurs |
| SetRestart: | by setting this variable the initialisation routine begins; it not used for the implementation of the FB3222; it should used to program an error handling routine inside the application program after an error occurs |

Configuration of the FB 3222 “Command_Head_X”:

```
⊟ Netzwerk 2 : Call FB3222 to read/write IQC22 completly

      CALL  "Command_HeadX" , "DB_Command"     FB3222 / DB3222
       HeadNumber         :=B#16#1
       CommandCycles      :="CommandCycles"     MB711
       StartAddress       :="StartAddress"      MW712
       CommandLength      :=B#16#3C
       NumberReadDBHead1  :=321
       NumberWriteDBHead1:=322
       NumberInstDB       :=32
       Read               :="ReadCommand"       M700.2
       Write              :="WriteCommand"      M700.3
       End_Command        :="EndCommand"        M700.1
       Error              :="ErrorCommand"      M700.5
       Busy               :="BusyCommand"       M700.4
       NoDataCarrier      :="NoDataCarrier"     M700.6
       StartCommand       :="StartCommand"      M700.0
```

---

The function block FB3222 “Command_Head_X” need to call together with the according instanz data block “DB_Command” inside the application program.

Example:

CALL FB3222; DB3222

It is possible to rename the function and the data block. The internal variables used as local variables. No global variables used inside the FB3222.

**Input parameter:**

| | |
|---|---|
| HeadNumber: [Byte] | defines the channel on which the command should executed;<br>Channel 1: HeadNumber := B#16#1<br>This variable should connected to a byte variable of the customer specific application program |
| CommandCycles: [Byte] | defines the number of executed command cycles inside the read or writes routine of the FB3222. The configuration is in decimal format. The maximal number is 5. The first command Read/write 60 Bytes of user data.<br>To read or write a specific data quantity following configuration is necessary:<br>60 Bytes: CommandCycles := 1<br>120 Bytes: CommandCycles := 2<br>180 Bytes: CommandCycles := 3<br>240 Bytes: CommandCycles := 4<br>256 Bytes: CommandCycles := 5 |
| CommandLength: [Byte] | defines the length of the user data which are inside one telegram; by using the IQC22 RFID tag the length is fixed at 60 Bytes<br>60 Bytes: CommandLength := B#16#3C<br>It is possible to configure this variable permanently with this value. |
| NumberReadDBHead1/2:[INT] | defines the number of the data block in which the read in data of head 1/2 are copied while read command execution. The length of the data block should be 256 Bytes (mirror of the complete memory of the IQC22) or it should have the same length like the read in data<br>NumberReadDBHead1 := 321 |
| NumberWriteDBHead1/2:[INT] | defines the number of the data block in which the write data for head 1/2 are copied before starting command execution. The length of the data block should be 256 Bytes (mirror of the complete memory of the IQC22) or it should have the same length like the data which should be written<br>NumberWriteDBHead1 := 322 |
| NumberInstDB: [INT] | defines the number of the according instanz data block of the according function block for the IDENTControl<br>NumberInstDB := 32 |
| Read: [Bool] | defines that a reading cycle will executed (start of the execution triggered by input “StartCommand”) |
| Write: [Bool] | defines that a writing cycle will executed (start of the execution triggered by input “StartCommand”) |

**In/Out parameter:**

| | |
|---|---|
| StartCommand: [Bool] | defines the start of a command routine by setting to true |

**Out parameter:**

| | |
|---|---|
| End_Command: [Bool] | signalize with a positive edge the end of a command routine; this output will reset to false if a command routine starts |
| Error: [Bool] | signalize that an error occurs while execution of a command routine |
| Busy: [Bool] | signalize that the execution of a command routine is still active; after the start this bit is set to true until the command routine is finished |
| NoDataCarrier: [Bool] | signalize that a RFID tag leaves the detection range of the head while execution of the command routine |