

Índice

DW13Modbus	3
<i>IMPLEMENTATION</i>	3
Error	3

DW13Modbus

D = Device(MODBUS)

W = Writing functions

The DW13Modbus function manages the interchange via MODBUS protocol among any apparatus that acts as a Master and Qmove (Slave).

In particular the function sorts an array (named "aswBuffer" in the below example) that must be declared by the user and passed to the function. This array will mirror faithfully the Modbus address table. The number of elements in this array must be at least equal to the number of the highest address of your device (for example, if the highest address between the variables you want to exchange is 600, the minimum size of the array must be of 600 items).

For the use of the function is mandatory set the 2 value on the "mode" parameter of the Modbus device passed to the function.

IMPLEMENTATION

DW13Modbus (Modbus, aswBuffer, slrdelay, gbWriteRead, sbError)

Parameters:

IN/OUT	VARIABLE TYPE	EXAMPLE NAME	DIM	
IN	INTDEVICE	Modbus	-	Mnemonic name of MODBUS devices used
IN	ARRSYS / ARRGBL	aswBuffer	W	Address Buffer
IN	SYSTEM	slrdelay	L	It is the time to wait before transmitting the reply.
OUT	GLOBAL	gbWriteRead	B	Variable that indicates the read or write request from the Master. 0 = no request 1 = reading 2 = writing
OUT	SYSTEM	sbError	B	Variable containing the error code

Error

After calling the function if there are any errors the error variable having the following values:

- 0 - No error
- 1 - "Mode" parameter not set correctly (<> 2)
- 2 - Address exceeds the Buffer size
- 3 - The number of word be written exceeds the Buffer size
- 4 - Address \leq 0
- 5 - Number of word writing \leq 0

Example 1

(Without the "gbWriteRead" flag address)

TASK_00

```

Modbus:idcard = 1
Modbus:mode=2
Modbus:prot=1
Modbus:wider=0
Modbus:brate=38400
Modbus:stopb=1
Modbus:par=0
Modbus:toutsyc=100
OPENCOM Modbus
WAIT Modbus:st_opencom

slrdelay = 0
MAIN:
    DW13Modbus (Modbus, aswBuffer, slrdelay, gbWriteRead, sbError)
    IF NOT sbError
        aswBuffer[20] = swPippo
        aswBuffer[21] = swPluto

        ELSE
            swMinnie = aswBuffer[30]
        ENDIF
        gbMessaggio = sbError ;Variable to shown the error message
    END
END

```

Example 2

(With the "gbWriteRead" flag address)

TASK_00

```

Modbus:idcard = 1

```

```

Modbus:mode=2
Modbus:prot=1
Modbus:wider=0
Modbus:brate=38400
Modbus:stopb=1
Modbus:par=0
Modbus:toutsyc=100
OPENCOM Modbus
WAIT Modbus:st_opencom

slrdelay = 0

MAIN:
  DW13Modbus (Modbus, aswBuffer, slrdelay, gbWriteRead, sbError)
  IF NOT sbError
    IF (gbWriteRead EQ 1)
      ;-- Reading Master -----
      aswBuffer[20] = swPippo
      aswBuffer[21] = swPluto
    ENDIF
    IF (gbWriteRead EQ 2)
      ;-- Writing Master -----
      swPippo = aswBuffer[20]
      swPluto = aswBuffer[21]
    ENDIF
  ELSE
    gbMessaggio = sbError ;Variable to shown the error message
  ENDIF
END

```

Note

- The “ gbWriteRead” variable allows you to save time for data exchange between Master and Slave. This variable takes the value 1 when the Master has requested a read and takes the value 2 when the Master demanded a writing. This allows you to upgrade the array only if that variable has a non-zero value and avoid doing it continuously. See the EXAMPLE 2 to understand how it works. For the first time, that variable is set to 1 (reading) even if the master has not made any request, to make sure that there is at least one buffer update. The 1 example does not use the gbWriteRead variable as it is optional.

Limits

The restriction on the number of variables that you can exchange is as follows

- Most readable/writable 29 word simultaneously

Documento generato automaticamente da **Qem Wiki** - <https://wiki.qem.it/>

Il contenuto wiki è costantemente aggiornato dal team di sviluppo, è quindi possibile che la versione online contenga informazioni più recenti di questo documento.