KissBox Application Note

Using KissBox IO3 and IO8 Modbus/TCP firmware

This application note explains Modbus/TCP parameters for the KissBox IO3C and IO8C products in order to integrate them into a Modbus/TCP installation.

Firmware configuration

The IO3C/IO8C must be loaded with the Modbus/TCP firmware. Please refer to KissBox Editor User's Manual for details about the procedure to load a new firmware if needed.

There is no specific configuration to create with the KissBox Editor, except the IP address used by the IO3C / IO8C. TCP port number is fixed to 502 and can not be changed. Modbus slave address is always 1.

Modbus/TCP objects vs. KissBox I/O cards

Modbus/TCP protocol is compatible with all types of KissBox I/O cards :

- Digital Inputs (e.g : DI8DC cards)
- Digital Outputs (e.g: DO4PR or DO8SK cards)
- Analog Inputs (e.g: AI8RA cards)
- Analog Outputs (e.g : AO8DA cards).

It is possible to use any number of any type of card in the card cage, from filling the whole card cage with a single type of card to mixing the four types in the same card cage.

The list hereafter describes the Modbus objects associated with each type of KissBox I/O card

Read Coils (function 0x01) : reads back Digital Outputs
Read Input (function 0x02) : reads Digital Inputs
Read Holding Registers (function 0x03) : reads Analog Inputs
Read Input Registers (function 0x04) : same as Read Holding Registers
Write Single Coil (function 0x05) : writes Digital Outputs
Write Single Register (function 0x06) : writes Analog Outputs
Write Multiple Coils (function 0x0F) : writes Digital Outputs
Write Multiple Registers (function 0x10) : writes Analog Outputs

IMPORTANT : a function shall be used to read/write I/O objects only if they are compatible with the Modbus function. Otherwise, the command is either ignored or will return null values. E.g:

- an attempt to read Input Registers for a slot containing a Digital Output will return 0 value for all inputs
- an attempt to send a Write Coils to a slot with Analog Outputs card is ignored

I/O addresses vs. Slots

Each slot is the IO3 or IO8 card cage is always seen as containing 8 channels, even if the I/O card in the slot is a 4 channels model (like DO4PR or DO4AR).

Slot	Modbus Object Addresses
1	0 to 7
2	8 to 15
3	16 to 23

I/O addresses for IO3C

Slot	Modbus Object Addresses

1	0 to 7
2	8 to 15
3	16 to 23
4	24 to 31
5	32 to 39
6	40 to 47
7	48 to 55
8	56 to 63

I/O addresses for IO8C

Automatic TCP disconnection by timeout

Modbus/TCP relies on TCP protocol sessions. Before any Modbus command can be sent to the KissBox, a TCP session must be opened by the Modbus/TCP client.

A client is expected to use “graceful” session termination once it does not need to communicate anymore with the KissBox IO3/IO8. In most cases, this is done automatically by the Operating System of the client when the Modbus software closes the communication socket.

However, it is possible that a client terminates a session “ungracefully”. This happens for example when the Modbus application crashes or if the computer running the Modbus/TCP client is forced to power off. In that case, the session would remain opened forever in the KissBox IO3/IO8, making the communication channel lost until the KissBox is reset.

To avoid this situation, the KissBox firmware implements a TCP communication timeout. Once a Modbus/TCP session is opened by a client, the KissBox must receive at least one Modbus/TCP command every minute. If no command is received for more than 60 seconds, the communication socket in the KissBox is closed automatically and becomes available again.

This disconnection system is totally transparent for most applications, since PLCs and other real-time systems poll constantly multiple times per second Modbus devices, ensuring the KissBox timeout to be reset.

In other cases, you have to make sure that your application performs at least one communication exchange every 60 seconds. An easy way is to perform a simple read operation on any slot of the KissBox at least a few times per minute. The read operation ensures that no output will be affected by the client if the application does not know what to send.

Modbus/TCP parameters

You will find hereafter the Modbus/TCP parameters as defined by “MODBUS Messaging on TCP/IP Implementation Guide” used by the KissBox.

NumberMaxOfClientTransaction : 1 (only one command is allowed in a TCP message)

NumberMaxOfServerTransaction : 4 (up to four clients can be connected at the same time on the same KissBox IO3 / IO8)

Unit Identifier : 1

Typical response time : less than 1 millisecond