MagicKey Pro Network

serial protocol description

Version 0.9

Table of Contents

1. General protocol:

1.1. Hardware

serial hardware parameters of the serial device line is RS485, 38400, 8N1. (a single V24 version is also available)

1.2. Command

Each commandframe uses following format:

<STX>AACCCCC<ETX>

<STX>	Start Of Text (02_{HEX})
<ETX>	End Of Text (03_{HEX})
AA	2 Byte addresses (ascii notation: '0' means 30_{HEX}). The range of the address is '01' to '99'. '00' is the broadcast address. Every station has to acknowlegde on either the personal station or the broadcast address. Systemdesigner have to make shure that not two stations have the same address nor that broadcast address is used with more than one station connected. For ZeitControl software only the addresses from "00" to "90" are usable. '99' is used as a temporary download address (-> 'select' command).'98' is factory default address for new delivered devices. Due to hardware limitations there must not be more than 32 phyical devices connected to a single segment.
CCCC	Command with parameters, parameters are seperated through spaces. Each parameter has a fixed length that has to be filled up with spaces for alphanumerical parameters or has to be prefixed with 0 (zero) for numerical parameters.

1.3. Response

for each accepted command the device has to send a response.

<STX>AA:RRRRR<ETX>

<STX>	Start Of Text (02_{HEX})
<ETX>	End Of Text (03_{HEX})
AA	2 Byte addresses (ascii notation: '0' means 30_{HEX}). If not otherwise stated the device shows the current address. If the devices does not have a valid address the response should be "NN" ($4e_{HEX}$, $4e_{HEX}$).
RRRR	Response according to the command.

2. Commands.

All commands are casesensitive. Numeral parameters can be written in upper or lower case (eg. ABC_{HEX}, abc_{HEX}).

In command description values in squared brachets [] are optionals, values in curly brachets changes behaviour of the command if used. (mostly parameter get/set)

Visual Basic Class methods return the complete response string without the comprising <STX>/<ETX>

*2.1. **alive**, state alive message*

Currently this command respond a '0'. (zero)

Command:

`alive`

Parameters:

none

Response:

`0`

MagicKeyPro VBasic class

Following VisualBasic functions uses the alive command

CmdAlive alive check device response.

*2.2. **cfg**, device configuration*

Query or set serveral configurable device parameters. As a convention read/write parameters are in lower case, readony parameters are in upper case letters.

Command:

```
cfg[ P{VV}]
```

Parameters:

P		device parameters to get/set
	a{XX}	get/set the device station address. The device address has to be uniq. Station addesses are coded as BCD numbers. („01“ is station 1). Possible number rages from „01“ to „99“.
	r{XX}	get/set the device relais time. Relais time is coded as hexadecimal values as multiple of 1/10 seconds.
	F	Get the device features. Current device featues response is a 4 char string. The Byteposition have following meaning
		0 0 = Small EEprom (32K) 1700 log entries
		1 = Large EEprom (64K) 3600 log entries
		1 0 = not serial updatable
		1 = serial updatable
		2/3 currently not applicable (shown as 'X')
	R	Get the software release/build date. Release string is in format MMHHSS@mmddyy
	S	Get the device uniq device address (18 chars string). Device address for example is used in the „select“ command.

Response:

responses depending on the subcommand (see above)

MagicKeyPro VBasic class

Following VisualBasic functions uses the cfg command:

CmdGetCFG	cfg	get all possible CFG subcommands.
CmdGetStatID	cfg a	get the primary (permanent) stationid of the selected device.
CmdSetStatID	cfg aXX	set the primary (permanent) stationid of the currently selected device.
CmdGetRelaisTime	cfg r	get the preset relais open time. This value is measured in 10/Sec.
CmdSetRelaisTime	cfg rXX	set the preset relais open time.
CmdGetFeatures	cfg F	get features of the device.
CmdGetRelease	cfg R	get the release date of the device firmware.

*2.3. **catags**, clear all user tags*

Clear all usertag. Command duration is about 3 seconds. During this period the device is unaccessable.

Command:

`catags`

Parameters:

none

Response:

ok

MagicKeyPro VBasic class:

Following VisualBasic functions uses the catags command:

*2.4. **ctag**, clear single tag*

Command:

`ctag PPPP`

Parameters:

PPPP	position of the tag storage place to clear.

Response:

PPPP ok	everything is ok. (transponder was cleared)
PPPP overflow	if PPPP is out of range (transponder number is out of range)

MagicKeyPro VBasic class:

Following VisualBasic functions uses the ctag command:

*2.5. **date**, set/get device date*

Command:

```
date[ DD.MM.YYYY:d]
```

Parameters:

DD.MM.YYYY:d	set the date	
	DD	current day.
	MM	current month.
	YYYY	current year.
	d	current day of week, Week started with monday = 1.
none	get the current date used by the device.	

Response:

The device alway respond with the current date in the same format as above

```
DD.MM.YYYY:d
```

MagicKeyPro VBasic class

Following VisualBasic functions uses the date command:

*2.6. **info**, get device info*

Get information about the device.

Command:

```
info
```

Parameter

none

Response

Versions number are in the form <major number>.<minor number><revision>. (eg. 1.0a) Revisions are letters with an underscore stating a development version.

```
(c) ZeitControl 2004, MKP-N 1.0_
```

MagicKeyPro VBasic class:

Following VisualBasic functions uses the info command:

*2.7. **flash**, flashing the status leds*

Identify the device through flashing alternative with both LEDs.

Command:

`flash`

Parameters:

none

Response:

`ok`

MagicKeyPro VBasic class:

Following VisualBasic functions uses the info command:

*2.8. **lgcmt**, insert a comment in the log/protocol*

Insert a comment in the log. The comment has to be 6 alphanumeric chars.

Command:

`lgcmt CCCCCC`

Parameters:

CCCCC	Alphanumerical comment to insert. 6 Bytes with additional spaces

Response

PPPP NNNN EE TT:TT:TT DD.DD.DDDD:D CCCCCC

PPPP	Position in protocol buffer
NNNN	consecutivly numbered (0 to $FFFF_{hex}$)
EE	event number (constantly 8)
TT:TT:TT	time of event (hour:minute:second)
DD.DD.DDDD:D	date of event (day.month.year:day of week)
CCCCCC	comment itself according to the command

`0002 0003 08 02:10:02 01.00.2000:1 CCCCCC`

MagicKeyPro VBasic class

Following VisualBasic functions uses the info command:

*2.9. **lock**, lock/unlock transponder detection*

Command:

```
lock[ P]
```

Parameters:

P	set/clear current lock state	
	1	set the lock. During locked state device will not check for transponder. Lock will be cleared after 15 seconds of inactivity. A command that will change data in the device will retrigger the lock. During lock phase the red led is light.
	0	clear the lock.
none	query the current lock state	

Response

```
L    state of current lock
```

MagicKeyPro VBasic class

*2.10. **log**, protocol/log query*

Command:

```
log[ PPPP]
```

Parameter:

PPPP	get the log entry on position
none	get the protocol counters.

Response:

Format of log entry

```
PPPP NNNN EE TT:TT:TT DD.DD.DDDD:D VVVVVVV
```

PPPP Position in protocol buffer
NNNN consecutivly numbered (0 to $FFFF_{hex}$)
EE event. (in braches additional informations are shown)
- 1 access granted (transponder number and timezone)
- 2 access rejected (transonder number and reason to reject)
- 4 system restart
- 5 system date changed (date before)
- 6 system time changed (time before)
- 8 user comment (comment)
- 9 Hardware failure (failure code)

TT:TT:TT time of event (hour:minute:second)
DD.DD.DDDD:D date of event (day.month.year:day of week)
VVVVVVV additional data according to the event.

Example:
0000 2B00 01 13:35:13 02.09.2004:4 60230ACB 00

Format of protocol counters

```
PPPP NNNN
```

PPPP position in protocol buffer
NNNN consecutivly numbered (0 to $FFFF_{hex}$)

MagicKeyPro VBasic class

*2.11. **read**, read current (in field) transponder*

Read transponder currently in field. For not interferring with the normal functionally the device should be locked during action.

Command:

```
read
```

Parameter:

none

Response:

```
empty     no transponder in field
TTTTTTTT   transponder number currently in field
no sid    hardware error
```

MagicKeyPro VBasic class

*2.12. **relais**, activate the relais*

Activate the relais for the programmed time

Command:

```
relais
```

Parameter:

none

Response:

ok

MagicKeyPro VBasic class

*2.13. **reset**, device restart*

Restart the device. After the reboot the device is unaccessable for about 5 second.

Command:

```
reset
```

Parameter:

none

Response:

ok

MagicKeyPro VBasic class

*2.14. **rstlog**, reset the log pointer*

Reset both the next log storing position and the log numbering to zero. The contents of the log itself in not cleared.

Command:

```
rstlog
```

Parameter:

none

Response:

ok

MagicKeyPro VBasic class

*2.15. **select**, select a device*

Select a device according the unique device serial number.

Command:

```
select GGGGGGGGGGGGGGGGG SS
```

Parameter:

G{18}	Device address. Each device has an undestroyable uniq id.
SS	new station address

Response:

ok

MagicKeyPro VBasic class

*2.16. **time**, get/set device time*

Set/get the current time. Note for exact time you should use a radio controlled clock (in germany DCF77) since the software clock on the PC is very inacurate.

Command:

```
time[ hh:mm:ss]
```

Parameter:

hh	hour
mm	minute
ss	second

Response:

```
hh:mm:ss
```

MagicKeyPro VBasic class

*2.17. **tz**, get/set a timezone entry*

Get / set a timezone entry. It is possible to define up to 8 timezones. Each timezone consist onf a starting time an ending time and a combination of the accepted days.

Command:

```
tz PP[ hh:mm:ss hh:mm:ss DD]
```

Parameter:

PP	Timezone entry position. (0 to 7)
hh	either the start or the end hour
mm	either the start or the end minute
ss	either the start or the end second
DD	combination of the allowed day of weeks. These number contains following bits.

0	Monday
1	Tuesday
2	Wednesday
3	Thursday
4	Friday
5	Saturday
6	Sunday

Response:

hh:mm:ss hh:mm:ss DD

MagicKeyPro VBasic class

*2.18. **rtag**, read transponder of a tag position*

Command:

```
rtag PPPP
```

Parameter:

PPPP	position of the transponder to tag to read

Response:

```
PPPP empty
PPPP TTTTTTTT ZZ
```

MagicKeyPro VBasic class

*2.19. **wtag**, write transponder to a tag position*

Write a transponder to

Command:

```
wtag PPPP TTTTTTTT ZZ
```

Parameter:

PPPP	storage position if this value is FFFFhex the transponder is stored on the first free position
TTTTTTTT	Transponder number
ZZ	combination of 8 possible timezones. (Bitposition)

0	Timezone 1
1	Timezone 2
2	Timezone 3
3	Timezone 4
4	Timezone 5
5	Timezone 6
6	Timezone 7
7	Timezone 8

Response:

```
PPPP TTTTTTTT ZZ
```

MagicKeyPro VBasic class

*2.20. **boot**, jump to bootloader (not yet supported)*

Command:

```
boot
```

Parameter:

none

Response:

```
ok
```

MagicKeyPro VBasic class

*2.21. **baud**, set the baudrate of the device (not yet supported)*

Command:

```
baud[ BB]
```

Parameter:

BB

Response:

ok

MagicKeyPro VBasic class

*2.22. **dump**, dump certain areas of device eeprom (debugging only)*

Command:

```
dump AAAA SS
```

Parameter:

AAAA
SS

Response:

MagicKeyPro VBasic class

Note: This command is for debugging purpose and might only be enabled in development versions.

2.23. dbg, set special device parameters (debugging only)

Command:

dbg CCSSSS

Parameter:

CC	Command		
SSSS	SubCommand		
	1		Debug values for ReaderAsic
		02.0X	Set IQ register
		04.0X	Set Set1 register
		06.0X	Set Set2 register
		08.0X	Set Test register
		1X.XX	Get Status register

Response:

response is depending on the command and not documented.

MagicKeyPro VBasic class

Note: This command is for debugging purpose and might only be enabled in development versions.

*2.24. **clmem**, clear device eeprom (debugging only)*

This command clears (write FFhex) to the I2C eeprom.

Command:

clmem

Parameter:

none

Response:

ok

MagicKeyPro VBasic class

Note: This command is for debugging purpose and is only enabled in development versions.

3. Command summary

Command	*String*	*Cmd length*	*Rsp length*	*Timeout*	*Lock*
getConfig	`cfg`	3	-1[1]		N
getDate	`date`	4	12		N
getDeviceId	`cfg S`	5	18		N
getFeatures	`cfg F`	5	4		N
getInfo	`info`	4	35		N
getLock	`lock`	4	1		N
getRelease	`cfg R`	5	30		N
getStatId	`cfg a`	5	2		N
getTime	`time`	4	8		N
reset	`reset`	5	2	50[2]	N
select	`select <S*16> <DD>`	28	2	500	Y
setLock	`lock <L>`	6	1		Y
setLogComment	`Lgcmt <CCCCCC>`	14	44		Y
setRelais	`cfg r<HH>`	7	2		Y
setStatId	`cfg a<DD>`	7	2		Y

1 Length of reponse depends on software version. Currently the size is 4 characters.
2 Device is inaccessable for about 2 seconds during reset period.