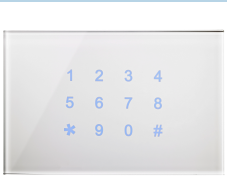
blumotix
TOUCH YOUR WORLD

Technical manual - KRISTAL LINE

KNX Access Contrai Touch Keyboard



KNX

TOUCH YOUR WORLD

INDEX

1 General

2 Technical data

3 ETS Library

1 General

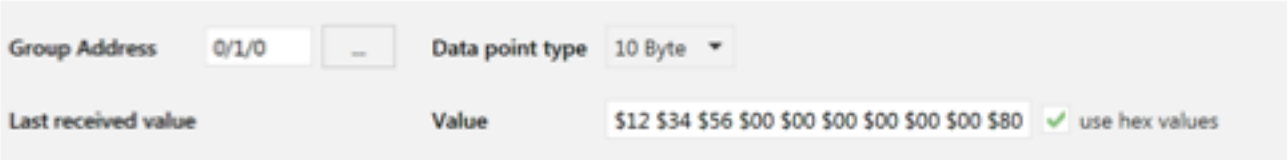
It's a KNX Access Contro Touch Keyboard by Blumotix.
It is a 12 keys traditional keyboard designed with glass capacitive touch technology to join access control performance with very sophisticated design.
DOORY is configurable with your Smartphone to manage access control from a remote site.
Kb&b application permits to set check in and check out to have a valid access code during the vacancy reservation.
Each device can store up to 100 access to manage the access in common area with each personal code.

2 Technical data

Itis powered through KNX bus.
The keyboard has a backlight to achieve night vision that you can activate with proximity sensor.
Internal buzzer is a feedback to confirm a valid input.
DOORY accepts 6 digit code.
and * keys are to confirm END of CODE.

3 ETS Library

Object #0 - Code - Store / delete request (10bytes)
It is the main communication object dedicated to store and delete access code.
The datatype is a 10 byte size to maintain compatibility with RFID access control already present in the market.
First 6 characters are reserved to insert access code (3 bytes) and the last 2 characters are reserved to specify if you want to SAVE ($80) or DELETE ($00).
See following ETS example to understand better how to set the data.
Open DIAGNOSTICS and select your group address with 10 bytes Datapoint type.
Here you see what appears on the display:



Suppose to set the new code 123456.
Please, enable **use hex value**.
Now you can write new code inside the first 3 bytes of value:
Please, digit $12 $34 $56, and the last one with $80.
Confirm the transmission with WRITE.

Now, if you want to delete the code you can repeat the transmission by changing $80 with $00.

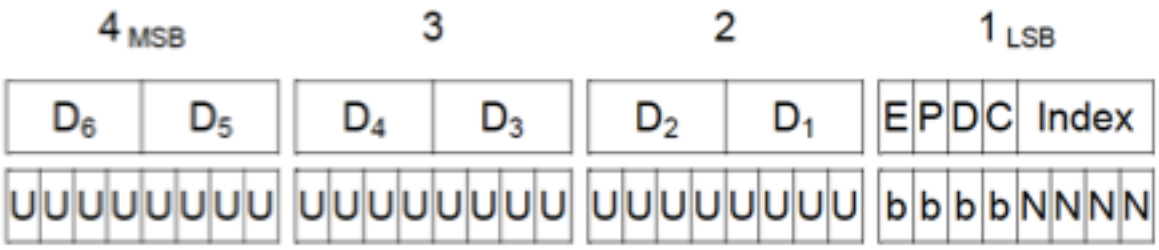
Following table shows all available Communication Object.

0	Code	Store/delete request	Gestione Codice	0/1/0	10 bytes	C	R	W	-	-			Low
1	Code	Store/delete request result			4 bytes	C	R	-	T	-	entrance access		Low
2	Codes	Number request			1 byte	C	R	W	-	-	counter pulses (0..255)		Low
3	Codes	Number reply			2 bytes	C	R	-	T	-	pulses		Low
4	Codes	Erase request			1 bit	C	R	W	-	-	boolean		Low
5	Codes area	Filled			1 bit	C	R	-	T	-	boolean		Low
6	Access	Enabled	Serratura Porta #1	0/0/1	1 bit	C	R	-	T	-	enable		Low
7	Live	Request			1 bit	C	R	W	-	-	boolean		Low
8	Live	Request reply			1 bit	C	R	-	T	-	boolean		Low
9	Live	Transmit message			1 bit	C	R	-	T	-	boolean		Low
10	Keypad ID	Transmit message			4 bytes	C	R	-	T	-	counter pulses (unsig...		Low
11	Code	Request by index			1 byte	C	R	W	-	-	counter pulses (0..255)		Low
12	Code	Reply by index			4 bytes	C	R	-	T	-	entrance access		Low
13	Access	Try result			4 bytes	C	R	-	T	-	entrance access		Low
14	Numeric key	Pressed			1 byte	C	R	-	T	-	counter pulses (0..255)		Low
15	Confirm key	Pressed			1 bit	C	R	-	T	-	boolean		Low
16	Access	Simulation			4 bytes	C	R	W	-	-	entrance access		Low
17	Backlight	On/Off			1 bit	C	-	W	-	-	switch		Low
18	Backlight	Status			1 bit	C	R	-	T	-	switch		Low
19	Buzzer on touch	Enable/Disable			1 bit	C	-	W	-	-	switch		Low
20	Buzzer on touch	Enable/Disable status			1 bit	C	R	-	T	-	switch		Low

Object #1 - Code - Store / delete request result (4 bytes - DPT15.000)

This object can send back a feedback about the Store / Delete request.
Notification is communicated through a DTP 15.000 data structure.

4 octets: $U_4U_4U_4U_4U_4U_4B_4N_4$



Following table shows how to decode each data field.

Field	Description	Encoding	Range
D_6, D_5, D_4, D_3, D_2, D_1	digit x (1…6) of access identification code. Only a card or key number should be used. System number, version number, country code, etc are not necessary. Ciphered access information code should be possible in principle. If 24 bits are not necessary, the most significant positions shall be set to zero.	Values binary encoded.	[0 … 9]
E	Detection error	0 = no error 1 = reading of access information code was not successful).	{0,1}
P	Permission (informs about the access decision made by the controlling device)	0 = not accepted 1 = accepted	{0,1}
D	Read direction (e.g. of badge) If not used (e.g. electronic key) set to zero.	0 = left to right 1 = right to left	{0,1}
C	Encryption of access information.	0 = no 1 = yes	{0,1}
Index	Index of access identification code (future use)	Value binary encoded.	[0 … 15]

First 3 bytes of each answer are reserved to access code.

If you attempts a store:
- D=1 and P=1 if store command has success.
- E=1 if store has not success.

If you attempts a delete:
- P=1 if delete command has success.
- E=1 if delete command has not success.

Object #2 - Codes - Number request (1 byte - DPT5.10)
Object #3 - Codes - Number reply (2 bytes - DPT7.01)

Request and Reply work together to provide informations.
First one is to ask information and the last one is to receive the answer.

It is possible to ask 3 different informations:

Sending 1 you ask how many codes are in memory.

Sending 2 you ask how many free cell you have to store new codes.

Sending 3 you ask how many codes can store the device. (answer=100)

Object #4 - Codes - Erase request

Sending 1 you ask to delete all codes

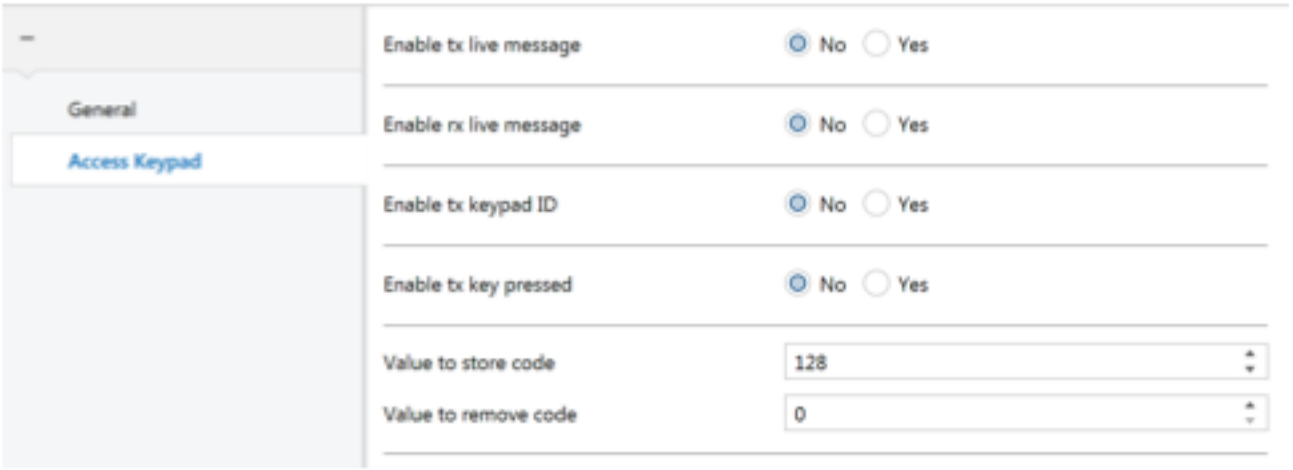
Object #5 -Codes area - Filled

This object can notify when memory is full.
After you store last available code this object send back 1 to notify that memory is full.
As soon as code is deleted this data come back to 0 to advise that memory is available to store new code.

Object #6 - Access - Enabled

This is the object to open the door!
Whenever a valid code is entered this object can notify door opening request.
When you confirm a valid code with # or * then it starts automatically a command OPEN.
When you drive an electric access may be interesting to set actuator with staircase function to limit at 2 seconds the driving impulse.
Sometime OPEN command can switch on courtesy light to facilitate access.

Safety controls



It is possible to set a periodic live alert to prevent abnormal conditions.
Periodic live alert is a check procedure to improve security grade.

It is possible to set the keyboard to notify a live alert (TX)
or it is possible to set keyboard to wait a live alert from a remote device (RX).

Enable RX live message

Object #7 - Live - Request
Input communication object to receive live alert from a remote device.

Object #8 - Live - Request reply
Output communication object to transmit ALARM when live alert failed.

If you set YES Enable RX live message then Keyboard wait each 10 seconds a live alert message = 1 trough object #7.
If notification is regular the keyboard stay silent.
If notification is not present or alert message = 0 then Keyboard send an ALARM = 1 through object #8.
Alarm will be reset after a live alert message = 1 through object #7.
Then Keyboard send ALARM = 0 trough object #8.

Enable TX live message

Object #9 - Live - Transmit message
Communication object to provide live alert message = 1 after each time out (Live tx interval).

If Enable tx live message = YES then Keyboard will send a live alert = 1 after each TX interval (sec).

Enable TX keypad ID

It is possible to enable a periodic live alert message with keypad ID.

Object #10 - Keypad ID - Transmit message
Keypad ID is transmitted through Object #10 with 4 bytes telegram in DPT12.001.

Check keyboard commands

It is possible to receive data from keyboard memory.

Object #11 - Code - Request by index
This object send a request to receive an access code.
You can transmit an index between 0 and 99 to receive the access code from that memory location.

Object #12 - Codice - Risposta per indice
When keyboard receive a request it can answer through this communication object by transmitting data with DPT15.000 (already described in object#1).

If access code is present then it comes back inside first 3 bytes with flags D=1 and P=1.
If access code is not present then 3 bytes = 0 come back with flag E=1.

Object #13 - Access - Try result
Each time someone try an access by typing a code then it is possible to receive a feedback from this communication object (DPT15.000).
First 3 bytes are the typed code.
Last byte has P=1 if access code is good.
Last byte is 0 if access code is bad.

Object #14 - Numeric key - Pressed
Each time someone types a numeric pad this communication object comes back the value with DPT5.010.

Object #15 - Confirm key - Pressed
Each time someone types # or * this communication object comes back a bit =1.

Object #16 - Access - Simulation
This communication object permits to send an access code to the keyboard to try a real access (DPT15.000)
Fill first three bytes with access code and last one with P=1 (0x04).

Keyboard feedback

Object #17 - Backlight - On Off
It permits to set or reset backlight

Object #18 - Backlight - Status
It comes back backlight status

Object #19 - Buzzer on touch - Enable / Disable
It permits to set or reset acoustic feedback

Object #20 - Buzzer on touch - Status
It comes back acoustic feedback.

TOUCH YOUR WORLD