Content:

1. INTRODUCTION 2

2. PHYSICAL LAYER REQUESTS 2

3. LAYER 2 2

3.1. LAYER 2 FRAMING 3

3.1.1. **Frames PC to TKS** 3

3.1.2. **Frames TKS to PC** 3

3.2. LAYER 6 TKS COMMANDS 5

3.2.1. **Service TKS commands** 5

3.2.2. **Active TKS commands** 5

4. FRAMES AND COMMAND EXAMPLES 6

5. APPENDIX A - KEYBOARD SCANCODES 10

6. APPENDIX B - CRC CALCULATION 11

7. APPENDIX C – INSTALLATION HIT. 11

1.Introduction

Telejet Keyboard Simulator - TKS is a simple device to simulate all keyboard functions and a hardware button-reset function on one or more PCs. For each of these PCs - Controlled PCs - it is necessary to use 1 TKS, plugged in the PC's keyboard plug. If a hardware button-reset function is requested, the TKS must be also connected with the hardware-reset pins on the motherboard of the PC (see here detailed manual for the motherboard).

One TKS is a component of large TKS's net. This Net topology has been done by placement of all TKS and TKS-Hubs. TKS-Hub is TKS's net repeater and allowed to build larger topology of Net.

TKS's net has only one leader - Server PC. The Server is the one to keep control over all local Net.

For better Net topology understanding see the „TKS User Manual“.

This document describe communication between this Server and TKS.

2. Physical layer requests

The set of TKS (at least 1 TKS) are connected in a TKS chain network. The server PC is connected to this TKS network via TKS-Hub or TKS-Mini-Hub-Cable. Both TKS-Hub or TKS-Mini-Hub-Cable are connected to the PC over serial card (COM-port). The communication speed is set to

2400bps, 8 databits, 1stopbit and no parity.

For communication with the TKS over TKS-Hub only lines TXD,RXD and ground are necessary, by using of TKS-Mini-Hub-Cable additionally lines DTR and RTS are requested, where for correct function DTR must be set to 0, RTS to 1.

3. Layer 2

Communication between server PC and target TKS (connected to the server PC via TKS-Hub or TKS-Mini-Hub-Cable) proceeds on a very simple way. The Server PC is leader at TKS net and only its send command-frame and waits for answer of target TKS. After answer receive Server PC is allowed to send next command frame immediately. (If there is no request for TKS’s answers and this command frames are for different TKSs there is no delay needs between this frames.)

Timeout for TKS answer is 200ms.

If there is no answer then :

a/ the address of the TKS transmitted in the frame doesn‘t match any TKS in the network.

b/ the content of the frame was damaged and CRC is bad.

c/ there could be the problem to transfer data from TKS into Controlled PC.

(There could happened follow situation : TKS Server PC is going to send any keyboard data into keyboard input of Controlled PC. The data blow from Server PC into TKS - his inner buffer. If TKS got full correct packet with the data the TKS start transmit this data into Controlled PC - his keyboard input. There could happened one problem. Theoretically Controlled PC can ask TKS to repeat transfer of last data whole the time. It is theoretically infinitive loop. To avoid it the TKS has inner timeout 5 sec. for all unpredicted situations like this. After expire this timeout TKS does not send any frame back into Server PC and generate inner initialize. It stop all action and start await new frame from Server PC.)

If there is no answer from addressed TKS then call this TKS again (after 200ms timeout). If there is not respond anymore wait more then 5 sec. and try again. If there is no respond check hardware.

3.1. Layer 2 Framing

3.1.1. Frames PC to TKS

Communication frame from WebServer PC to TKS.
There are our requests - commands into TKS
The packet has only one next form:

#TO_KBD xxx N y d..d CRC

where :

- #TO_KBD - it is the open and synchronize string It must be present at the beginning any time. (This string is send as ASCII - upper case string.)
- xxx - three bytes - binary address of destination TKS. There is address space from 0x000000 to 0xFFFFFE. Each TKS has unique dedicated address. This address is typed at his blue case. Address 0xFFFFFF is special "common address" and could be combined with “V” command only. (See follow.)
- N - one byte - binary - Number of bytes in the rest of this frame (include N, y, d and CRC).
- Y - one byte - ASCII format - command byte. (See follow.)
- d..d - 0 to 36 bytes - binary - keyboard scan code data. (Available after D command only.) - See example Nr.5
- CRC - two bytes - binary - 16 bits Cyclic Redundancy Check. CRC_msb and CRC_lsb - Most and Low Significant Byte. This is standard on PC and Windows system Computing of this CRC (CRC16) is fully explained in Appendix B. Data for CRC computing are all bytes in send packet. It mean from #TO_KBD..................d..d - all data (except CRC itself).

3.1.2. Frames TKS to PC

Communication frame from TKS back into WebServer PC.
There is a response from addressed TKS.
The packet has only one next form:

#TO__PC xxx N S R d..d CRC

where :

- #TO__PC - it is the open and synchronize string. It is in all TKS responds (Attention there are two underlines !!!)
- xxx - three bytes - binary address of answering JKBD.
- N - one byte - binary - Number of bytes in this frame. (include N, S, R, d..d and CRC).
- S - one byte - binary - keyboard status. This keyboard status TKS had got from Controlled PC. (It is the same as LEDs at standard keyboard.) Keyboard status byte has 8 bits (from 0 to 7) :
  - bit 0 - Scroll lock (if bit 0 is set -> Scroll lock is on)

- bit 1 - Num. lock (if bit 1 is set -> Num lock is on)
- bit 2 - Caps lock (if bit 2 is set -> Caps lock is on)
- bit 3 - bit not used
- bit 4 - ACK (acknowledge) bit. If this bit is set then the communication - data transfer between TKS and Controlled PC has been done successful. If this flag is zero it signalize some troubles at communication between TKS and Controlled PC.

  SW Ver. <= 1.06:

  ACK bit is a copy last ACK bit comes from PC. (ACK from last transferred Byte into PC.)

  SW Ver. = 2.01:

  A lot of PCs does not respond ACK bit correctly. Especially USB adapters. All standard keyboards does not check this ACK bit. Therefore TKS “emulate” ACK bit and place into bit 4 of keyboard status.

  The ACK bit is set on follow cond.:
  - PC ask TKS to make reset.
  - PC set TKS’s bits: Num lock, Caps lock, Scroll lock.
  - We got good ACK from PC.

  All the ACK set on real communication from PC. It mean the PC <-> TKS communication is OK and we can set ACK.

  The ACK bit is kept on until power fall down.
- bit 5 - not supported from SW ver. 1.02. long ACK bit. This bit is set if Controlled PC in data frame start ACK responding but end it after ACK timeout. It could happened at old mainboard (time of 8086 CPUs)

- bit 6 - bit not used
- bit 7 - bit not used

Power up condition this keyboard status - all clear.

- R - one byte - binary - keyboard repeat rate. This byte is set by BIOS after power up of Controlled PC. TKS power up condition -all clear.

- d..d - 1 to 36 bytes - data - This data are the TKS's respond only and could be :
  - "OK" - command has been successfully implemented.
  - "Error" - any error - like unknown command.
  - "E-LACK" - error - long ACK is detected. - bit 5 in Status byte is set.
  - "E-parity" - error - parity error in communication between TKS and Controlled PC.

- CRC - two bytes - binary - 16 bits Cyclic Redundancy Check. CRC_msb and CRC_lsb - Most and Low Significant Byte. This is standard on PC and Windows system Computing of this CRC (CRC16) is fully explained in Appendix B. Data for CRC computing are all bytes in send packet. It mean from #TO__PC..................d..d - all data (except CRC itself).

3.2. Layer 6 TKS Commands

3.2.1. Service TKS commands

This services commands are for TKS control and setting only.
His place in the frame (3.2.1.) is at "Y" position.

- "V" - question for TKS's SW version.
  - TKS respond by string containing program version in TKS.
  - Only this command has next special implementation. Standard TKS address space is from 0x000000 to 0xFFFFFE. The address 0xFFFFFF is "Common address". The TKS respond for "Common address" and "V" command only after TKS's power up (or reset). If you send any packet into TKS with his unique dedicated address then this TKS does not respond anymore at "Common address". (Up to next power up or reset) This special "Common address" function allowed us to include new TKS into our TKS's net. There is small example for this: Existing TKS net has 3 TKS. A software at WebServer looking for this TKS each 10 sec. It scan this 3 address (addresses are in it's data table) and at the end it scan address 0xFFFFFF. If there is no respond for this "Common address" it mean - there is nothing new. But if the Server software got respond - there is new TKS address - the Server SW have to update his address data table - and the new TKS is implemented into system.
- "I" - make general SW initialize of TKS. (Power up setting.)
- "W" - make general HW initialize of TKS. (Power up setting over TKS's WDT)
  - production test command only.

3.2.2. Active TKS commands

This services commands are for TKS control and setting only.
His place in the frame (3.2.1.) is at "Y" position.

- "R" - make immediately HW reset of Controlled PC.
  - This reset is through special 2 wire reset cable. (Between TKS and mainboard.) This HW reset is short 2,5 sec. pulse to ground at both wires of reset cable.
- "C" - enable "Delayed" HW reset of Controlled PC.
  - After receiving this command TKS activates its "Inner Timer" for about 4 min. If the "Inner Timer" expire then the TKS generate HW reset through special 2 wire reset cable. If TKS receive <u>any</u> valid command before its "Inner Timer" expires then this "Inner Timer" is reset back to 4 min. This command is usually used if you need "watch dog" function of any server. Power up setting – OFF.
- "K" - enable "Delayed" SW reset of Controlled PC.
  If the "Inner Timer" expire then the TKS generate keyboard combination Ctrl Alt Del and send it into PC' keyboard input. All other conditions are the same as with "C" command.
- "P" - option for PS2 version only.
  Command generate about 5,3 sec. long pulse (log.0) on Data PS2 wire.
  This function is dedicated to control "power switch" external HW.
- "D" - Data prefix command. Only this command is followed by data d..d .
  This data d..d are going to be copy one to one into keyboard Controlled PC's input. Data d..d available for PC's keyboard input – scan code – are in Appendix A.

4. Frames and Command examples

in this chapter the following notions are used :

- "abcd" - all inside quotes is ASCII string.
- 0x00000A - Number at Hex representation.
  There are 3 bytes - one byte is represented by two labels
  0,1,2,3,4....9,A,B,C,D,E,F
- <xx> - one binary byte - his Hex representation

1/ ask for new TKS's address by Common address (could be after his power up)

```
  - Send frame  :      "#TO_KBD"  <FF><FF><FF>   <04>       "V"      <90> <A0>
                        -------------    ------------------     -----         ----       ------------
                              |                  |                |           |            |
                            open            Common         Nr.of    Command      CRC
                            string            address        bytes
```

It is together 14 bytes.

The Sending frame start by opening string "#TO_KBD" then follow common address 0xFFFFFF = 3 bytes <FF> <FF> <FF>. Next byte is Number of bytes to the end of this frame. (Itself include.) At this example : Nr of bytes : this byte + command + CRC_msb + CRC_lsb => Therefore <04>. After "Nr. of Byte" byte follow Command byte. At this example the Command byte is : "V" - question for TKS's SW version. (Only this command is useable for common address.) All frames end by CRC16. There are two CRC_msb and CRC_lsb bytes. For this case (after CRC16 computing we got :) <90> <A0>

```
  - Received frame :  "#TO__PC" <00><2A><01>  <0B>     <10>       <00>   "1.02OK"   <25><C8>
                        -------------    ------------------     -----      ----          ----      ----------       ------------
                              |                |                |        |               |         |                |
                            open             TKS           Nr.of   Keyboard    Repeat   Data           CRC
                            string           address       bytes    status       rate
```

It is together 21 bytes.

Here this receive - packet started with open string "#TO__PC". Then follow 3 bytes - TKS's unique address <00><2A><01> = 0x002A01 - this address is typed at TKS's sticker. Then <0B> - Nr. of bytes (to the end itself include) - this is hex form. Decimal representation = 11 bytes. Then follow two bytes <10> <00> keyboard status and repeat rate. Then are data we asked for - software version - "1.02OK" - there is SW TKS's version 1.02 and O.K. At the end CRC_msb = <25> and CRC_lsb = <C8>

2/ Ask for SW version in TKS through his unique address = 0x002A01.

```
  - Send packet  :     "#TO_KBD" <00><2A><01> <04>       "V"      <78> <A0>
                        -------------    ------------------    -----         ----       ------------
```

```
                         |              |           |          |           |
                       open           TKS        Nr.of    Command    CRC
                       string        address     bytes

- Received packet :  "#TO__PC" <00><2A><01> <0B>    <00>        <00>  "1.02OK"  <25><C8>
                      -------------   ------------------   -----     ----          ----   ----------   ------------
                         |              |           |          |           |         |           |
                       open           TKS        Nr.of    Keyboard   Repeat    Data       CRC
                       string        address     bytes     status      rate
```

3/ Generate reset of Controlled PC over special 2 wire reset cable. (address is 0x002A01)

```
 - Send packet  :     "#TO_KBD" <00><2A><01> <04>       "R"      <BB> <EC>
                      -------------   ------------------   -----       ----       ------------
                         |              |           |          |           |
                       open           TKS        Nr.of    Command    CRC
                       string        address     bytes     reset

- Received packet :  "#TO__PC" <00><2A><01> <07>    <00>       <00>     "OK"      <25><54>
                      -------------   ------------------   -----     ----          ----   ----------   ------------
                         |              |           |         |             |         |           |
                       open           TKS        Nr.of    Keyboard   Repeat    Data       CRC
                       string        address     bytes     status      rate
```

4/ Initialize TKS at address 0x002A01

```
 - Send packet  :     "#TO_KBD" <00><2A><01> <04>       "I"      <B0> <AC>
                      -------------   ------------------   -----      ----       ------------
                         |              |           |          |           |
                       open           TKS        Nr.of    Command    CRC
                       string        address     bytes     initialize
```

- Received packet : - none. After receive initialize command he make reset immediately !!

5/ Send label "A" into Controlled PC (his keyboard scan code into Controlled PC's keyboard input).

There is a time to explain something about data - communication between PC and keyboard. If at the keyboard is press (and hold) any button - for example "A" then keyboard sent into PC not ordinal number of label "A" but his keyboard scan code only. This keyboard scan codes are at Appendix A. There the scan code for "A" button is 0x1C. If the "A" button released then keyboard sends two bytes - 0xF0 and 0x1C.There is prefix 0xF0 - "release button action" and the second byte - which button has been released - 0x1C - button "A". Then all data for press "A" and release it are : 0x1C 0xF0 0x1C

```
 - Send packet  :     "#TO_KBD"   <00><2A><01>  <07>      "D"      <1C><F0><1C>   <42> <77>
                      -------------   ------------------   -----       ----      ----------------------   ---------------
                         |              |           |          |              |                |
                       open           TKS        Nr.of    Command      button "A"         CRC
                       string        address     bytes     for data

- Received packet :  "#TO__PC" <00><2A><01> <07>    <10>        <00>     "OK"      <E5><50>
                      -------------   ------------------   -----     ----          ----   ----------   ------------
                         |              |           |          |           |         |           |
                       open           TKS        Nr.of    Keyboard   Repeat    Data       CRC
                       string        address     bytes     status      rate
```

6/ Send into Controlled PC string "hallo".
Scan codes for "H" = 0x33; "A" = 0x1C; "L" = 0x4B; "O" = 0x44

```
- Send packet  :    "#TO_KBD"   <00><2A><01>  <13>      "D"
                     -------------   ------------------      -----       ----
                          |                  |              |          |
                        open              TKS          Nr.of    Command
                        string           address      bytes    for data

  <33><F0><33> <1C><F0><1C> <4B><F0><4B> <4B><F0><4B> <44><F0><44>  <23> <58>
  ------------------   -----------------   ------------------   ------------------   -----------------   ------------
          |                   |                    |                   |                   |               |
     button "H"        button "A"         button "L"        button "L"        button "O"         CRC


 - Received packet :  "#TO__PC" <00><2A><01> <07>    <10>        <00>     "OK"     <E5><50>
                       -------------   ------------------    -----      ----           ----      ----------     ------------
                            |                  |              |         |              |          |             |
                          open              TKS          Nr.of   Keyboard    Repeat     Data        CRC
                          string           address      bytes     status        rate
```

7**/ Send into Controlled PC key combination Ctrl Alt Del - SW reset.**
Scan codes for Ctrl = 0x14; Alt = 0x11; Del = 0x71
This buttons have to be pressed together.

```
- Send packet  :    "#TO_KBD"   <00><2A><01>  <0D>      "D"
                     -------------   ------------------      -----       ----
                          |                  |              |          |
                        open              TKS          Nr.of    Command
                        string           address      bytes    for data

                  <14>   <11>   <71>  <F0><71>     <F0><11>    <F0><14>    <23> <08>
                   ----     ----     ----      -----------       ------------      -----------     -------------
                    |        |        |          |              |            |            |
                  press  press  press     release        release      release        CRC
                   Ctrl    Ald     Del        Del             Alt           Ctrl


 - Received packet :  "#TO__PC" <00><2A><01> <07>    <30>        <00>     "OK"     <28><FE>
                       -------------   ------------------    -----      ----           ----      -----       ------------
                            |                  |              |         |              |          |             |
                          open              TKS          Nr.of   Keyboard    Repeat     Data        CRC
                          string           address      bytes     status        rate
```

There could be some troubles with ACK respond from Controlled PC. The problem is the Controlled PC make reset and do not care about ACK respond.

8/ Set " Caps Lock " at Controlled PC.

If you like type Upper or Lower case from your keyboard the information about Upper or Lower case is not in the keyboard.
The scan code Up or Low case of button is the same !! But inside in PC is one flag called "Caps Lock " and it keep information about Up or Low case.
The same is for Num Lock and Scroll Lock. The whole information about this flags are in Keyboard status byte.
At the keyboard are only LEDs signalize this Keyboard status. The LEDs are controlled by PC not by keyboard !

If we are going to change "Caps Lock" from Low to Up or from Up to Low we will have to send scan code for Caps Lock.
Scan codes for Caps Lock = 0x58

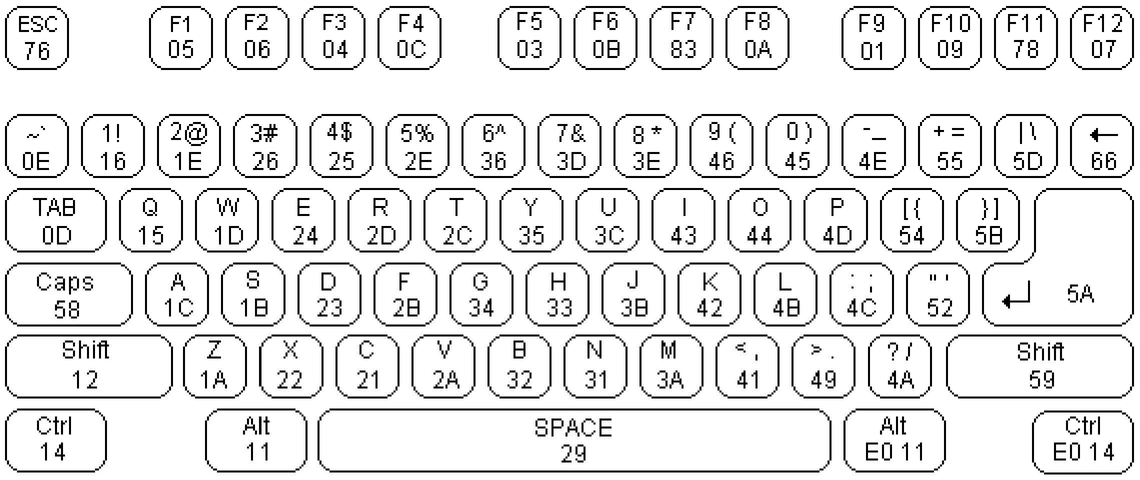
```
 - Send packet  :    “#TO_KBD” <00><2A><01> <07>      “D”    <58><F0><58>   <64> <37>
                     -------------  ------------------  -----     ----     -------------------   ------------
                         |               |            |        |             |               |
                       open            TKS          Nr.of   Command   Caps Lock        CRC
                       string        address       bytes

- Received packet : “#TO__PC” <00><2A><01> <0B>   <14>      <00>    ”OK”  <D5><51>
                     -------------  ------------------  -----   ----      ----    -----   ------------
                         |               |            |      |          |       |       |
                       open            TKS          Nr.of  Keyboard  Repeat  Data    CRC
                       string        address       bytes  status    rate
```

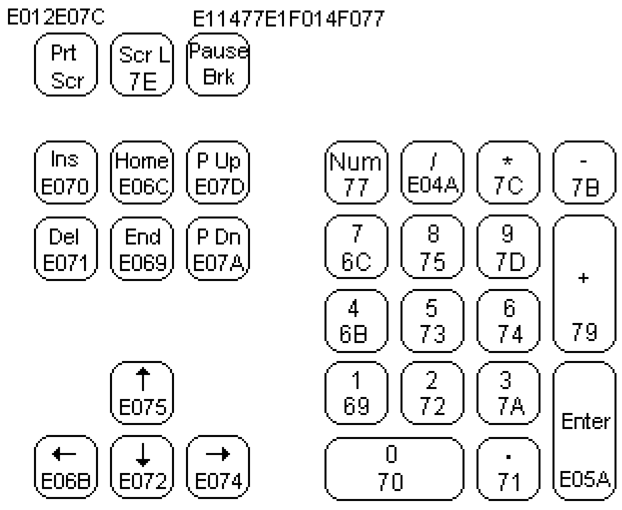
Here according Keyboard status byte - bit 2 - Caps Lock is ON. (Num Lock and Scroll Lock are OFF)
Now if we type "hallo" it is typed in Upper Case !!
If we send it again then Caps Lock should be OFF.

```
- Send packet  :      “#TO_KBD” <00><2A><01> <07>     “D”    <58><F0><58>  <64> <37>
- Received packet :  “#TO__PC” <00><2A><01> <0B>   <10>   <00>  ”OK”       <E5><50>
                                                    -----
                                                      |
                                               Keyboard status
```

There according Keyboard status byte - bit 2 - Caps Lock is OFF.

5. Appendix A - Keyboard scancodes





6. Appendix B - CRC Calculation

```
unsigned short data CRC_msb word_;     /* Global variable
unsigned short data CRC_lsb word_;     /* Global variable

// ##########################################################
// CRC 16
// ##########################################################
void CRC_make(unsigned char data_byte) small
  {
  unsigned int code crc16_table_[16] = {
    0x0000,0xCC01,0xD801,0x1400,
    0xF001,0x3C00,0x2800,0xE401,
    0xA001,0x6C00,0x7800,0xB401,
    0x5000,0x9C01,0x8801,0x4400 };
  unsigned short data crc16_word_;
  unsigned short data r1_word_;

  crc16_word_ = (CRC_msb << 8) + CRC_lsb;
  r1_word_    = crc16_table_[crc16_word_ & 0x0F];
  crc16_word_ = (crc16_word_ >> 4) & 0x0FFF;
  crc16_word_ = crc16_word_ ^ r1_word_ ^ crc16_table_[data_byte & 0x0F];
  r1_word_    = crc16_table_[crc16_word_ & 0x0F];
  crc16_word_ = (crc16_word_ >> 4) & 0x0FFF;
  crc16_word_ = crc16_word_ ^ r1_word_ ^ crc16_table_[data_byte >> 4];

  CRC_msb = crc16_word_ >> 8;
  CRC_lsb = crc16_word_ & 0xFF;
  }
```

7. Appendix C – Installation hit.

TKS properties:
1/ each TKS has two addresses.
  a/ The fist one is his own real address typed on top sticker.
    This real address is for the one TKS communication.
  b/ Global address 0xFFFFFF
    This address is dedicated for installation only.
    TKS is enabled to receive global address only:
      - on "Power up" => TKS start powered from PC (not from "blue line " !!!)
      - before any real address command.
        (
          If TKS got any command with his real address => I was already installed and server know
          It's address.
          Therefore TKS switch off flag to be sensitive for global address.
          This situation is kept to next power up or low level reset.
)
      - with "V" command.

2/ Installation schedule:
  step a/ The TKS server should address with some not danger command like "V" all TKSs
      on server's TKS address table.
  step b/ Check by global address - is there some not yet addressed TKS.
      If there is some (maybe not correct !!) respond

```
              {
                  => there is some TKS was not addressed in step a.
                  => something bad happened at previous installation.
                       On this case is no chance to find this TKS by SW. To make a loop over all address
                       space --> good like.
                       Is better to to find TKS by disconnecting "blue line" and check where the TKS lying.
                       ( Use method " half interval cutting " )
              }
              Else
              { All is O.K. and go to step c. }

    step c/  plug new TKS into system.
    step d/  check by global address existence and new address of plugged TKS.
             If  ( ( there is some bad respond )  and  ( step b pass correct ) )
             {
                - It is real problem of " blue line " length
                - TKS is bad
                - all you can imagine should be bad at HW.
             }
             else
             {
               no problem - you got new TKS's address => include into server address table as new one.
              }
     step e/ make some real addressing ( "V" command ) of new TKS from a system to switch off new
             TKS's flag to be sensitive for global address.
```

If all pass through there should not be any respond to global address.
But take a care on any (local or global) power up. The server has to check all TKS by real address!
=> installation step b/