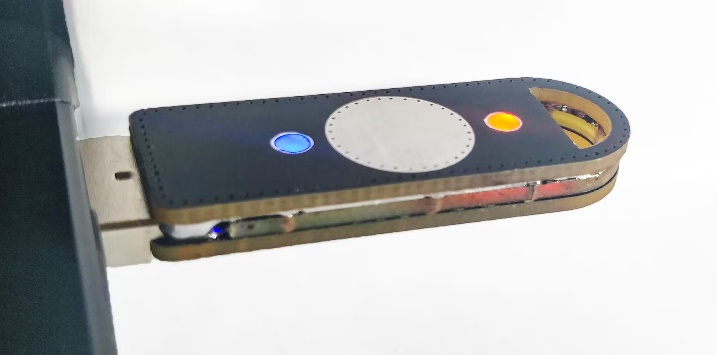
**HW-electronic-key communication protocol**

**This device is used as an electronic token/key that unlocks access locked files. Thus, only if it is available, the user can access encrypted files.**

*This key uses UART communication to receive and send commands.* **Command list:**  
All request commands are presented in eight-byte numeric format:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Command | Specific | | | | CRC |
| 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | 1 byte |

In which the first byte specifies the command number, the next 4 are responsible for the type of the sent command, and the six byte performs the calculation of the CRC. The device uses the XOR (example below) method to calculate CRC.

Example:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Command | Specific | | | | CRC |
| 0x05 | 0x01 | 0x00 | 0x00 | 0x01 | 0x03 |

1. 0x05 XOR 0x01 = 0x04 - *comparing the first and second byte;*  
2. 0x04 XOR 0x00 = 0x04 - *comparing the result of the first and second byte and the third byte*;  
3. 0x04 XOR 0x00 = 0x04 - *comparing the result of the previous step with the third byte;*  
4. 0x04 XOR 0x01 = 0x03 - *result of the CRC.*

A similar CRC counting system is used to check the response of the BD (STM32F0xx)

1. ***Check connection of the Key:***

*This command allows the computer to make sure that the HW-electronic-key is really connected to the device.*      -> PC: send request to working port “Is the key connected?”

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Command | Specific | | | | CRC |
| 0x01 | 0x00 | 0x00 | 0x00 | 0x01 | 0x00 |

      -> BD: send answer to request side

| Command | Response | | | | | | CRC |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0x01 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x01 | 0x00 |

Responses:  
          if the key was found, the next step.  
          else: message “Key is not connected”.   
         *\* PC App sends requests to BD until Key will be installed or User stops the App.*

1. ***Get key number:***

*This command allows the computer to read HW-electronic-key serial number. Number is like a “login” that allows the PC App to recognize whether the key is really in the database.*

*The key number consists of 4 bytes, which are generated by the application at the first entry or by a user with a special command.*

        -> PC: send request for key number

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Command | Specific | | | | CRC |
| 0x02 | 0x00 | 0x00 | 0x00 | 0x01 | 0x01 |

          -> BD: read and send from BD memory Key number

| Command | Response | | | | | | CRC |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0x02 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x01 | 0x01 |

Responses:

*\*Here, the validity of the key in the system database is checked.*  
         if the key number is equal to the number in the PC App database, allow the next step.  
         if the key number is “empty”, send the message “WARNING: NEW KEY!”, purpose to use the command that allows writing to BD key number or automatic generation.  
         else: message “key is not valid”, interrupt session with message: “NOT VALID KEY- NUMBER!”.  
  
Command allow to set Key number:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Command | Specific | | | | CRC |
| 0x02 | 0x01 | 0x01 | 0x01 | 0x01 | 0x00 |

Key number example: **A145** – one letter and three random digits.

1. ***Get the Key ID:***

*This command allows the computer to read HW-electronic-key ID. Unlike the Key number, an ID is like a “password” that allows the PC App to provide access to unlock the necessary files. As well as the key number, the key ID also consists of 4 bytes that are randomly generated by the program the first time it is entered, and then generated twice for security each time it is used.*

-> PC: send request for key ID

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Command | Specific | | | | CRC |
| 0x03 | 0x00 | 0x00 | 0x00 | 0x01 | 0x02 |

          -> BD: read and send from BD memory ID number

| Command | Response | | | | | | CRC |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0x03 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x01 | 0x02 |

            if key ID is equal to ID in PC App database, allow decoding.  
            else: message “key is not valid”, interrupt session.

1. ***Confirmation Key:***

*This command returns "yes" if the ID of the key matches the ID in our database + it must also match the* ***key number.*** *If something is wrong, then the program will not give access to files.*

-> PC: check if Key number and ID confirmed

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Command | Specific | | | | CRC |
| 0x04 | 0x00 | 0x00 | 0x00 | 0x01 | 0x03 |

        -> BD: read and send from BD memory ID number

| Command | Response | | | | | | CRC |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0x04 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x01 | 0x03 |

Responses:  
                        if “yes”, allow PC access to encrypted file   
                        else: “try another Key”, interrupt session

***Timeout****: 1 sec - a period, which sets a wait time for command checking.*