Version : **0.01a**

The entire data exchange process occurs through Modbus RTU protocol.
The logic for managing the process of installing/updating and removing firmware is described below.

Available Modbus RTU functions:

- **0x03** – Read holding registers;
- **0x06** – Write single register;
- **0x14** – Read file records;
- **0x15** – Write file records;

Available files:

- **0x0001** – File with application;
- **0x0002** – File with bootloader information;

1. Application erase process:

   Writing value **0x0001** to register APP ERASE (**0x0002**) will delete the firmware file. After removing the firmware, the contents of the registers should be as follows:
   APP SIZE (0x0001) – **0x0001** (no application available);
   APP SIZE (0x0001) – **0x0000** (size is 0);

2. Application install/update process:

   Before each new firmware, the old one must be deleted through erase process.
   write expected number of records (N) into APP SIZE (0x0001)register. Expected number of records should be calculated as follows:
   N = (application file size in bytes)/ (content of BLOCK SIZE (0x0001) register)+ 1
   split firmware file into records and write them to file with id **0x0001**. In case of success after firmware update register APP SIZE (0x0001) will contain **0x0002** (application is ready to start).

Below is the file structure with bootloader information (file with id **0x0002**). The structure is 2-byte aligned.

```
typedef struct
{
    char boot_version[17];  // array with actual bootloader version in ASCII
    char boot_name[33];     // array with bootloader name in ASCII
    uint32_t available_rom;

}BootloaderInfo_TypeDef;
```

Table 1 – register map and description

Address	Register	Access	Description	Usage
0x0000	*APP CONTROL*	R/W	Enable read or write mode for the file. After writing a specific value, the loader enters extended message length mode until the read or write operation is completed.	Reading will always return **0x0000**; **0x0001** - prepare for reading; **0x0002** - prepare for writing.
0x0001	*APP SIZE*	R/W	Contains actual application size in records. APP size in bytes = number of records (value in APP SIZE) * record size in bytes (value in BLOCK SIZE). Locked for writing until the application file is removed from flash memory.	-
0x0002	*APP ERASE*	R/W	Used to delete a file with ID **0x0001** (Application)	Reading will always return **0x0000**; **0x0001** - erase application;
0x0003	*APP START*	R/W	Used to start application (Bootloader will be unavailable after this command).	Reading will always return **0x0000**; **0x0001** - start application;
0x0004	*BOOT CONTROL*	R/W	Used to control bootloader.	**0x0000** - stay in boot after restart disabled; **0x0001** - stay in boot request is active.
0x0005	*BOOT STATUS*	R	Contains actual bootloader status.	Will return **0x04** exception in case of writing; **0x0000** - unknown status of bootloader; **0x0001** - no application, bootloader is ready; **0x0002** - application is ready to start; **0x0003** - error, application file is corrupted or any other error occurred.
0x0006	*BLOCK SIZE*	R	Contains actual record size in bytes.	Will return **0x04** exception in case of writing;