

4D-Joystick – Backup and Restore Interface

1. Introduction

The CLI interface of the 4D-Joystick support an option to backup the configuration of the system and restore it later or on another device. The interface is described in this document.

Following synonyms are used in this document:

- Carriage Return → CR or \r (ASCII 0x0D)
- Line Feed → LF or \n (ASCII 0x0A)
- Acknowledge → ACK (ASCII 0x06)
- Negative Acknowledge → NAK (ASCII 0x15)
- Command Line Interface → CLI

Basically, the backup is a dump of the EEPROM, where all configurations are stored. The EEPROM data have a total size of 8448 bytes. The backup data is divided into 132 packages of 64 bytes, each separated by a carriage return.

2. Backup

The backup can be started by typing “backup” to the CLI (“backup\r\n”).

The CLI answers this with the four characters for a new line (CR, LF and two spaces). These characters can be ignored. Afterwards, the CLI starts sending the EEPROM contents in packages of 64 Byte (separated by CR).

After all packages are transmitted (132 packages total) a acknowledge (ACK) is sent.

Example (green = user input, yellow = CLI outputs, white = backup output):

```
> backup\r\n_____
<package 1: 64 Byte of data>\r
<package 2: 64 Byte of data>\r
<package 3: 64 Byte of data>\r
...
<package 132: 64 Byte of data>\r ACK \r\n
>
```

3. Restore

The restore process can be started by typing “restore” to the CLI (“restore\r\n”). The CLI answers this with the four characters for a new line (CR, LF and two spaces). These characters can be ignored.

The CLI sends an ACK next, which shows, that the joystick is waiting for the backup data. The backup data must be transmitted in packages of 64 byte, each separated by a CR. Each package is answered with an ACK or a NACK.

Before the next package can be transmitted, the ACK must be received (the joystick needs some time for processing each package).

After finishing the restore process the joystick will be restarted to load the new configuration.

If problems occur the joystick will answer with a NACK and immediately stop the restore-process. No data will be written to the EEPROM in this case.

Example (green = user input, yellow = CLI outputs, white = restore output):

```
> restore\r\n_____
<package 1: 64 Byte of data>\r ACK
<package 2: 64 Byte of data>\r ACK
<package 3: 64 Byte of data>\r ACK
...
<package 132: 64 Byte of data>\r ACK \r\n
>
```

4. Notes

The first few bytes of the EEPROM contain a special pattern, which is used to check if the storage is initialized. Each time the joystick boots up, this pattern is checked. If it is not valid, the joystick will override the storage with its default values.

However, in future firmware versions this pattern will be changed, if the storage structure will be changed. During the backup/restore process this pattern is copied/overwritten. Therefore, it will depend on the new firmware if restoring backups from old firmwares is accepted (migrating features must be implemented) or if they will be overwritten with default values, because the pattern is different.

The data from the EEPROM is sent as it is stored. In some terminal programs (for example “Putty”) this can lead to strange behaviour, because the raw data can accidentally contain special characters or strings (chars used for control, escaped sequences, ...). To correctly use the backup/restore function, the data must be processed raw and shall not be interpreted.