The best way to update the firmware is a **volz\_can\_boot** software. It works both for Windows and Linux and supports Peak and Kvaser CAN adapters.

Detailed description of the volz\_can\_boot software can be found in the document “**Bootloader software for CAN actuators**”.

Default settings for the new PCB after flashing the bootloader:

**Bitrate:** 500 kbit/s

**Base CAN ID:** 7E0h

**Device ID:** 15

You must flash both ACEs and then restart the actuator.

**Examples:**

**volz\_can\_boot.exe -f firmware.hex -w -i 15 -p 1 -b 7E0**: write firmware.hex to a servo flash memory, don’t reset bootloader after writing, use 15 as unit identity and ACE1 as updated PCB, base CAN ID = 7E0h

**volz\_can\_boot.exe -f firmware.hex -w -i 15 -p 2 -b 7E0**: write firmware.hex to a servo flash memory, don’t reset bootloader after writing, use 15 as unit identity and ACE2 as updated PCB, base CAN ID = 7E0h

**Volz CAN**

Default settings after restart of the flashed actuator:

**Bitrate:** 500 kbit/s

**Base CAN ID:** 3E0h

**Device ID:** 0

**Examples:**

**volz\_can\_boot.exe -f firmware.hex -w -i 0 -p 1 -b 3E0**: write firmware.hex to a servo flash memory, don’t reset bootloader after writing, use 0 as unit identity and ACE1 as updated PCB, base CAN ID = 3E0h

**volz\_can\_boot.exe -f firmware.hex -w -i 0 -p 2 -b 3E0**: write firmware.hex to a servo flash memory, don’t reset bootloader after writing, use 0 as unit identity and ACE2 as updated PCB, base CAN ID = 3E0h

You can check functionality of the Volz CAN firmware after startup by sending **7E0h:00h** message, it’s a backdoor ID, so you will receive a response like **3Fxh:00h**, where x is a device ID.

To change the device ID use this command: **3E0h:11h 01h** to set device ID = 1. It will be responded with **3F0h:11h 01h**. Second byte in CAN data packet is a new ID to set.

**Pipistrel**

Default settings after restart of the flashed actuator:

**Bitrate:** 500 kbit/s

**Base CAN ID:** 7E0h

**Device ID:** 0xFF

**Examples:**

**volz\_can\_boot.exe -f firmware.hex -w -i 15 -p 1 -b 7E0**: write firmware.hex to a servo flash memory, don’t reset bootloader after writing, use 255 as unit identity and ACE1 as updated PCB, base CAN ID = 7E0h

**volz\_can\_boot.exe -f firmware.hex -w -i 15 -p 2 -b 7E0**: write firmware.hex to a servo flash memory, don’t reset bootloader after writing, use 255 as unit identity and ACE2 as updated PCB, base CAN ID = 7E0h

Actuator with default ID (FFh) doesn’t set anything and must be configured first.

Use **7DEh: 01h 00h 00h 11h 00h 00h 00h 00h 00h** to configure Device ID = 0. The actuator responses with **7DFh: 01h 00h 00h 11h 00h 00h 00h 00h 00h**.