

Briefing:

These utilities support **only** interfaces based on **ATmega162** chip. NEC, ARM or others are not supported!

To detect if your current cable has bootloader enabled just run **VAGCOM_HWType.exe** and press the **Read** button. Make sure that the cable is connected to both – computer's USB port and OBD2 port (or 12V power source). It will also show the HW type of the cable.

If the tool shows that bootloader is enabled you can flash EEPROM using **VAGCOM_EEWriteLang.exe**. Default language is English but you can choose any other available language from the dropdown list in **VAGCOM_EEWriteLang.exe** utility. If you choose any additional custom language then the interface will be enabled for two languages in total: the one which was selected and English as well.

Install official VCDS version 17.1.3, 17.8, 18.2, 18.9, 19.6 (or later), copy **VCDSLoader.exe** into installed directory and amend the shortcut on Desktop (and Start menu) to start **VCDSLoader.exe** each time instead of original shortcut to VCDS.exe. VCDS.exe will be renamed to VCDS.exeL. **Internet connection can stay ON**. Antivirus, Anti-Malware or any other similar software (including Windows Defender) will need to have exception added to prevent it deleting the Loader.

Upgrading from previous Loader versions

When upgrading from previous **minor** Loader version (e.g. from 8.0 to 8.1) it is enough just to copy/overwrite the new **VCDSLoader.exe** file in VCDS directory.

However, if upgrading from previous **major** Loader version (e.g. from 7.4 to 8.0), you also must reflash the EEPROM using **VAGCOM_EEWriteLang.exe** utility (see the *Briefing* section).

Bootloader disabled and manual programming

If bootloader is disabled in your cable then it must be programmed using wires and a programmer (search eBay for USBASP). In this case it would be best to determine the HW type manually because the cable might be incorrectly flashed at factory. See sections below on how to correctly identify the hardware type.

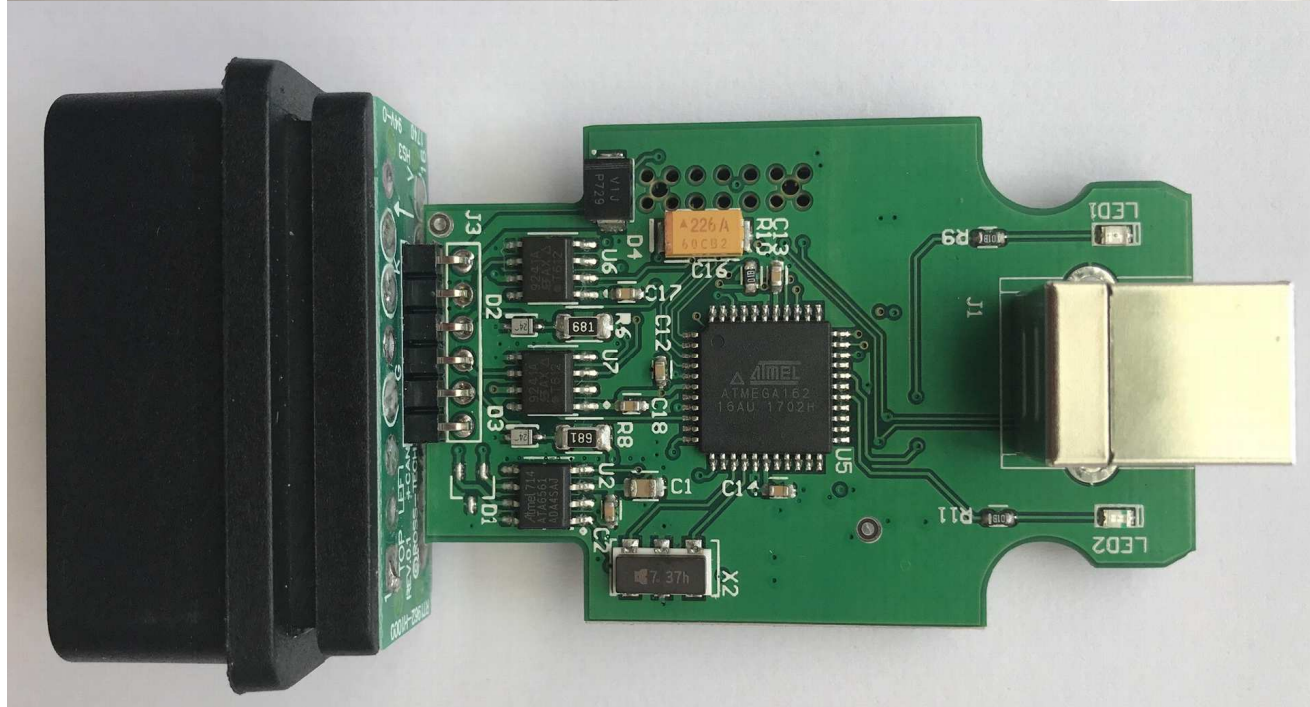
If in doubt or in case **VAGCOM_HWType.exe** shows "Interface not found" error even with the cable connected to USB and 12V power source it is advised to reflash FTDI too: **ftdi_ft_prog_*.xml** file is for flashing using **FT_Prog**; **ftdi_mprog_*.ept** file is for flashing using **MProg**.

*.hex files are Intel HEX formatted files of the *.bin files.

Recommended interface to buy

You can buy «**HEX-V2**» interface from www.OBD2CarTool.com shop. Please note that this is not a true HEX-V2 interface, it is just a restyled case, that's why the name.

The interface comes with **ATmega162** chip, protective diode on OBD2 pin 16, **L9637D** chipsets instead of transistors for K/L-Line and detachable USB cable. Bootloader is enabled on this interface from the factory so there is no need to use USBASP programmer – it can be reflashed using **VAGCOM_EEWriteLang.exe** utility.



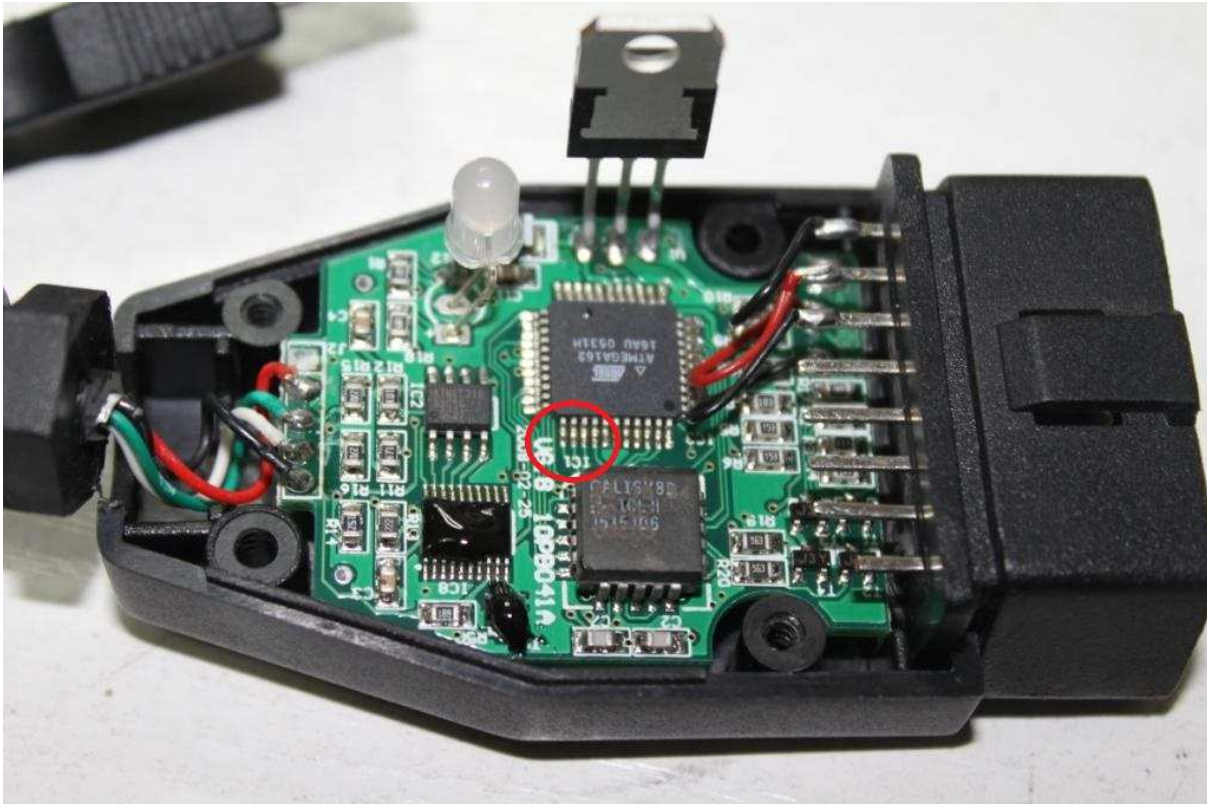
Hardware with STC chip

If the cable contains STC chip then before programming you must cut the line between pin 9 of STC chip and the R50 resistor. No need to remove the chip or do any other actions.

Hardware type HW 0x44

Identification: ATmega162 pins 34 and 35 must be connected to GAL chip.

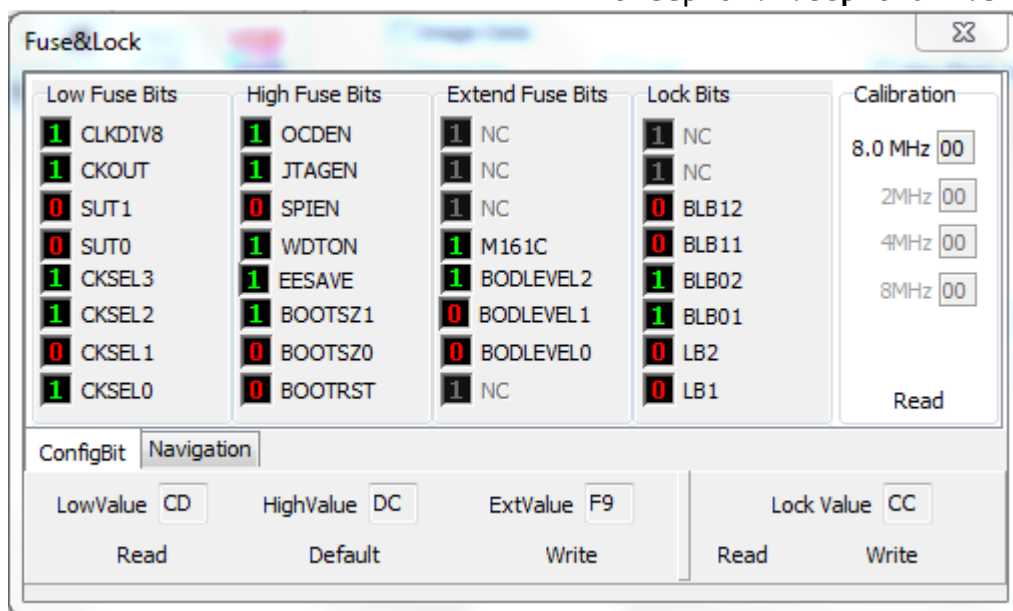
If in doubt use a multimeter to confirm the connectivity.



FuseBits: E:F9, H:DC, L:CD

Programming:

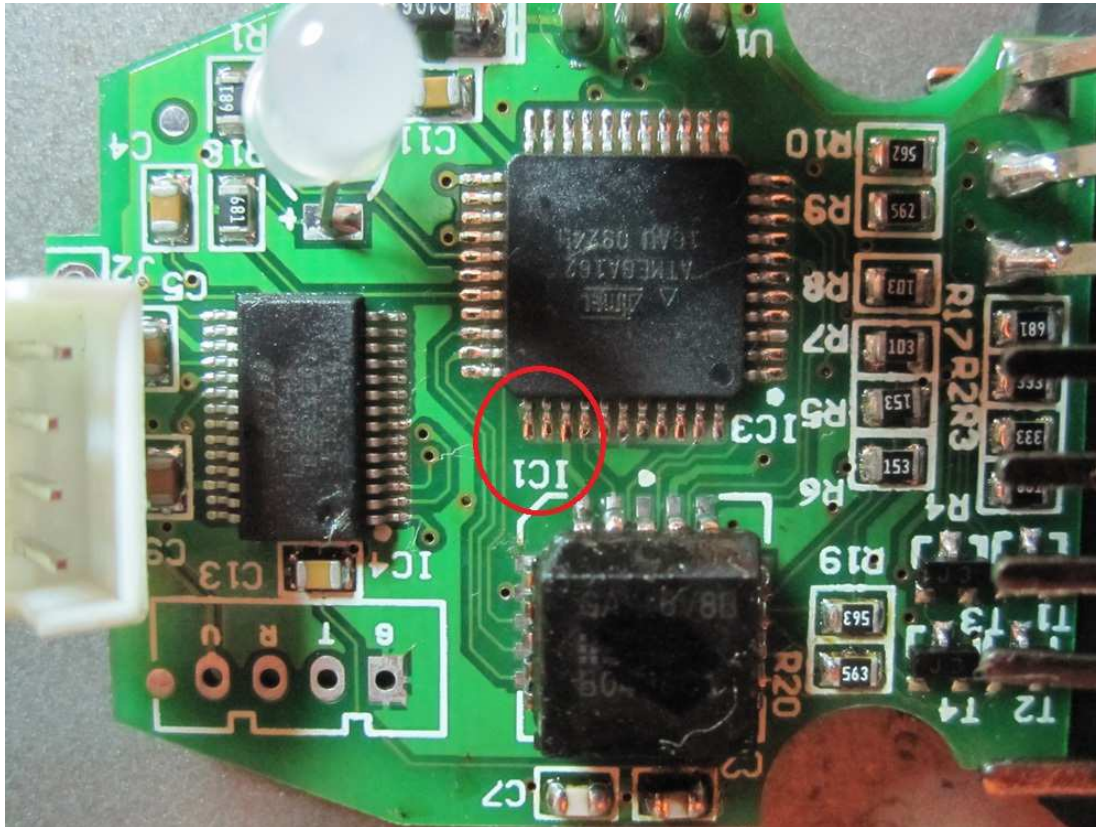
```
avrdude.exe -p m162 -c usbasp -e -U lock:w:0x3f:m  
-U efuse:w:0xf9:m  
-U hfuse:w:0xdc:m  
-U lfuse:w:0xcd:m  
-U flash:w:Flash1.96_HW_0x44.bin:r  
-U eeprom:w:eepromavr.bin:r
```



Hardware type HW 0x46:

Identification: no connection from ATmega162 pins 34 and 35 to GAL chip.

If in doubt use a multimeter to check the connectivity.



FuseBits: E:F9, H:DA, L:CD

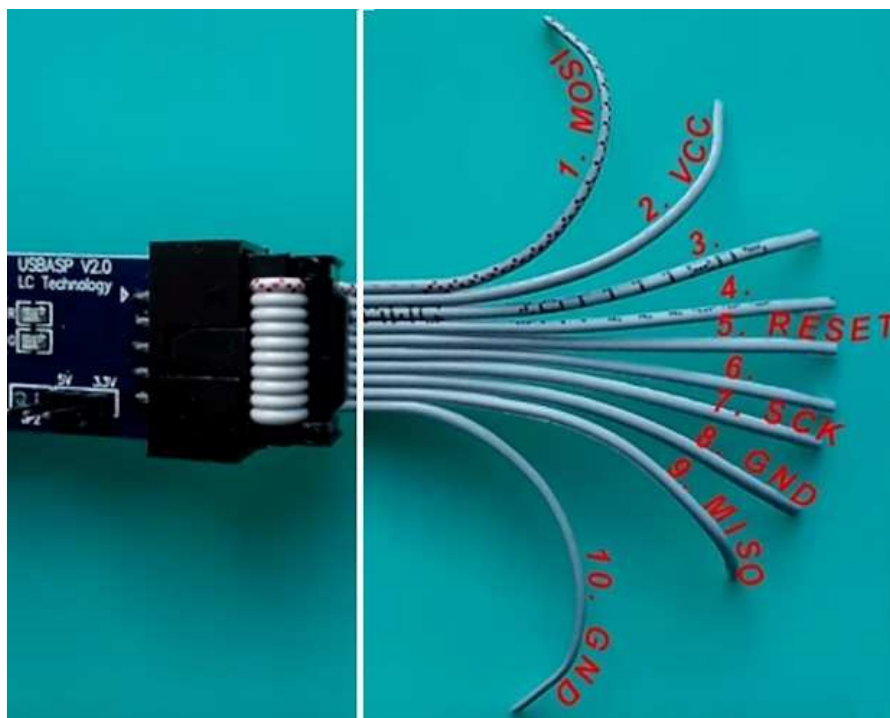
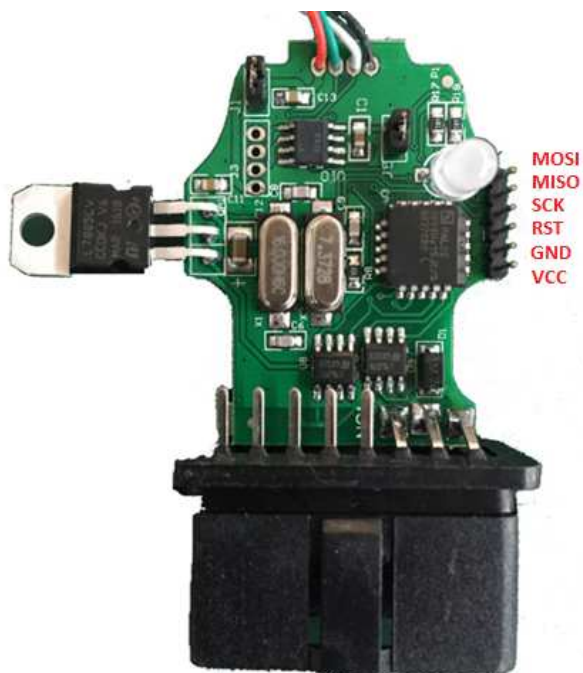
Programming:

```
avrdude.exe -p m162 -c usbasp -e -U lock:w:0x3f:m  
-U efuse:w:0xf9:m  
-U hfuse:w:0xda:m  
-U lfuse:w:0xcd:m  
-U flash:w:Flash1.96_HW_0x46.bin:r  
-U eeprom:w:eepromavr.bin:r
```

Fuse&Lock				Calibration
Low Fuse Bits	High Fuse Bits	Extend Fuse Bits	Lock Bits	
<input checked="" type="checkbox"/> CLKDIV8	<input checked="" type="checkbox"/> OCDEN	<input checked="" type="checkbox"/> NC	<input checked="" type="checkbox"/> NC	8.0 MHz <input type="text" value="00"/>
<input checked="" type="checkbox"/> CKOUT	<input checked="" type="checkbox"/> JTAGEN	<input checked="" type="checkbox"/> NC	<input checked="" type="checkbox"/> NC	2MHz <input type="text" value="00"/>
<input type="checkbox"/> SUT1	<input type="checkbox"/> SPIEN	<input checked="" type="checkbox"/> NC	<input type="checkbox"/> BLB12	4MHz <input type="text" value="00"/>
<input type="checkbox"/> SUT0	<input checked="" type="checkbox"/> WDTON	<input checked="" type="checkbox"/> M161C	<input type="checkbox"/> BLB11	8MHz <input type="text" value="00"/>
<input checked="" type="checkbox"/> CKSEL3	<input checked="" type="checkbox"/> EESAVE	<input checked="" type="checkbox"/> BODLEVEL2	<input checked="" type="checkbox"/> BLB02	
<input checked="" type="checkbox"/> CKSEL2	<input type="checkbox"/> BOOTSZ1	<input type="checkbox"/> BODLEVEL1	<input checked="" type="checkbox"/> BLB01	
<input type="checkbox"/> CKSEL1	<input checked="" type="checkbox"/> BOOTSZ0	<input type="checkbox"/> BODLEVEL0	<input type="checkbox"/> LB2	
<input checked="" type="checkbox"/> CKSELO	<input type="checkbox"/> BOOTRST	<input checked="" type="checkbox"/> NC	<input type="checkbox"/> LB1	
				Read
ConfigBit Navigation				
LowValue <input type="text" value="CD"/>	HighValue <input type="text" value="DA"/>	ExtValue <input type="text" value="F9"/>	Lock Value <input type="text" value="CC"/>	
Read	Default	Write	Read	Write

Example hardware and connection schema

This is **HW type 0x44**. Notice the programming header available on the right side of the board. For best results with this interface ensure that the jumpers J1 and J5 are removed (open).



If your interface does not have the programming header (unlike the shown above) then wires will need to be soldered directly to the chips. Set Vcc voltage to 5V on the programmer; there should be no power on the OBD2 side.

Recommended schema is this:

INTERFACE TO USBASP CONNECTION

