



How to program MAYA volume controller

We use and recommend [usbtinyisp](#) as a very good programmer.

It is a very popular AVR programmer, very well documented, easy to make, works great with avrdude, is AVRStudio-compatible and tested under Windows, Linux and MacOS X.

Below procedure assume you are a Linux user. Some commands are for MacOSX or Windows.

Unless you want to build your own usbtinyisp from scratch, a kit with all components is available at <http://www.adafruit.com/products/46>

Detailed instructions on how to install this device under Windows, Linux and MacOS X can be found here <https://learn.adafruit.com/usbtinyisp>

Just a small parenthesis

```
{  
In case you are are compiling avrdude from source, as we did, you may get an error at first run  
(libtinfo.so.5 not found). This may be solved by creating a symbolic link inside lib directory:  
cd /lib  
ln -s libncurses.so.5.9 libtinfo.so.5  
}
```

Next step is to get an Atmel AVR fuse calculator. Use Google to search for one, or use this one

<http://www.engbedded.com/fusecalc/>

In our particular case (ATMega32) AVRDUDE arguments are **-U lfuse:w:0xd4:m -U hfuse:w:0xd9:m**

Connect your usbtinyisp with ISP connector to Vicol-Audio volume-controller board, exactly as you can see in picture. ISP connector on Maya correspond to J8.

On Maya you should have led D1 light ON and on usbtinyisp the green led ON.

Connect your usbtinyisp to PC – please note that if you run Windows you need

<https://learn.adafruit.com/usbtinyisp/download> , no drivers are needed for Linux/Unix/MacOS X.



Test your communication with microcontroller (command you need to type is in red, output in blue):
`root@tibix:~# avrdude -c usbtiny -p m32`

`avrdude: AVR device initialized and ready to accept instructions`

`Reading | ##### | 100% 0.00s`

`avrdude: Device signature = 0x1e9502`



avrdude: safemode: Fuses OK (H:FF, E:D9, L:C4)

avrdude done. Thank you.

Next step is to write fusebit (command you need to type is in red, output in blue):

```
root@tibix:/# avrdude -c usbtiny -p m32 -U lfuse:w:0xc4:m -U hfuse:w:0xd9:m
```

avrdude: AVR device initialized and ready to accept instructions

Reading | ##### | 100% 0.00s

avrdude: Device signature = 0x1e9502
avrdude: reading input file "0xc4"
avrdude: writing lfuse (1 bytes):

Writing | ##### | 100% 0.00s

avrdude: 1 bytes of lfuse written
avrdude: verifying lfuse memory against 0xc4:
avrdude: load data lfuse data from input file 0xc4:
avrdude: input file 0xc4 contains 1 bytes
avrdude: reading on-chip lfuse data:

Reading | ##### | 100% 0.00s

avrdude: verifying ...
avrdude: 1 bytes of lfuse verified
avrdude: reading input file "0xd9"
avrdude: writing hfuse (1 bytes):

Writing | ##### | 100% 0.00s

avrdude: 1 bytes of hfuse written
avrdude: verifying hfuse memory against 0xd9:
avrdude: load data hfuse data from input file 0xd9:
avrdude: input file 0xd9 contains 1 bytes
avrdude: reading on-chip hfuse data:

Reading | ##### | 100% 0.00s

avrdude: verifying ...
avrdude: 1 bytes of hfuse verified

avrdude: safemode: Fuses OK (H:FF, E:D9, L:C4)



avrdude done. Thank you.

Now change directory to where your firmware is located and proceed with last step.
Write firmware to chip (command you need to type is in red, output in blue):

```
root@tibix:/home/tibi//firmware# avrdude -c usbtiny -p m32 -U flash:w:maya_v_0_7_rays_lcd.hex
```

```
avrdude: AVR device initialized and ready to accept instructions
```

```
Reading | ##### | 100% 0.00s
```

```
avrdude: Device signature = 0x1e9502  
avrdude: NOTE: "flash" memory has been specified, an erase cycle will be performed  
    To disable this feature, specify the -D option.  
avrdude: erasing chip  
avrdude: reading input file "maya_v_0_7_rays_lcd.hex"  
avrdude: input file maya_v_0_7_rays_lcd.hex auto detected as Intel Hex  
avrdude: writing flash (21782 bytes):
```

```
Writing | ##### | 100% 33.45s
```

```
avrdude: 21782 bytes of flash written  
avrdude: verifying flash memory against maya_v_0_7_rays_lcd.hex:  
avrdude: load data flash data from input file maya_v_0_7_rays_lcd.hex:  
avrdude: input file maya_v_0_7_rays_lcd.hex auto detected as Intel Hex  
avrdude: input file maya_v_0_7_rays_lcd.hex contains 21782 bytes  
avrdude: reading on-chip flash data:
```

```
Reading | ##### | 100% 20.03s
```

```
avrdude: verifying ...  
avrdude: 21782 bytes of flash verified
```

```
avrdude: safemode: Fuses OK (H:FF, E:D9, L:C4)
```

avrdude done. Thank you.

Disconnect usbtinyisp and power off/on Maya volume-controller.
Congratulations, you volume-controller is ready for use with a new firmware !