

Programiranje AVR mikrokontrolerov v Linux Ubuntu 20.04

Kratka navodila za namestitev programa, programskih knjižnic in prvi program

Tomaž Kušar

April, 2020



Uvod

V skripti je opisan postopek namestitve vse potrebne programske opreme za programiranje AVR mikrokontrolerov z operacijskim sistemom Linux Ubuntu 20.04. Opisan je postopek namestitve programskih knjižnic in postopki, da na mikrokontrolnik Atmega 328p (Arduino UNO) prenesete prvi program. Pravzaprav ne potrebujete kaj veliko programske opreme.

Program AVRdude in knjižnica libusb

Preden začnemo s programiranjem namestimo kar potrebujemo. V linuxu ne potrebujemo kaj dosti, pa še to je hitro nameščeno. V terminalnem oknu vpišemo naslenje:

```
sudo apt-get install gcc-avr binutils-avr gdb-avr avr-libc avrdude
```

S tem ukazom namestimo vse potrebno.

Namestiti moramo knjižnico za komunikacijo preko USB vrat in program AVRdude, s pomočjo katerega bomo naš program prenesli na mikrokrmilnik. V preteklosti, sem običajno vedno namestil libusb knjižnico.

```
sudo apt-get install libusb-1.0-0-dev
```

```
sudo apt-get install avrdude
```

USB komunikacija med vmesnikom Arduino in računalnikom

V primeru da delamo z vmesnikom Arduino UNO R3, potem sta možna dva načina prenosa programa iz računalnika na mikrokontroler. Novejše različice Arduina imajo namesto FTDI komunikacijskega čipa nameščen na plati mikrokrmilnik ATmega16u2, ali ATmega8u2, kloni Arduino vmesnika pa čip CH341.

1. Prva možnost prenosa .hex kode je torej direktno preko USB priključka na vmesniku Arduino.
2. Druga možnost programiranja pa je preko zunanjega programatorja, kot je recimo tudi standardni programator [USBasp](#).

Ko programiramo s programom AVRdude, je potrebno v ukazno vrstico zapisati oznako USB priključka, na katerega je priključen Arduino vmesnik.

V terminalnem oknu odtipkamo:

```
dmesg | tail
```

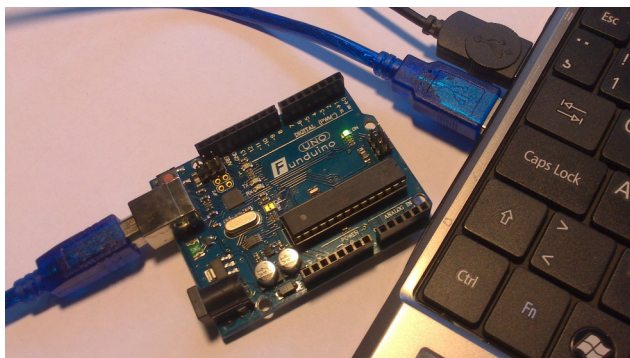
Spodaj sta zaslonski sliki obeh primerov, ko imamo vmesnik Arduino priključen direktno na USB vhod in ko imamo vmesnik povezan z računalnikom preko standardnega USBasp programatorja. Program AVRdude na srečo podpira oba programatorja. V prvem primeru bi morali na zaslonu opaziti oznako podobno `cdc_acm 2-1:1.0: ttyACM0: USB ACM device`. Iz tega razberemo, da je oznaka USB priključka, na katerem je vmesnik `ttyACM*`. To bo pomembno pri klicanju programa AVRdude.

```
skrbnik@crunchbang10: ~  
skrbnik@crunchbang10:~$ dmesg | tail  
[ 8953.014554] [Firmware Bug]: battery: (dis)charge rate invalid.  
[10677.730132] chrome[4593]: segfault at 22b74d34 ip b44eb7d0 sp bff7da40 error  
4 in chrome[b299a000+4d69000]  
[10902.792240] usb 2-1: new full-speed USB device number 3 using uhci_hcd  
[10902.992320] usb 2-1: New USB device found, idVendor=2341, idProduct=0043  
[10902.992352] usb 2-1: New USB device strings: Mfr=1, Product=2, SerialNumber=2  
20  
[10902.992376] usb 2-1: Manufacturer: Arduino (www.arduino.cc)  
[10902.992395] usb 2-1: SerialNumber: 55330313635351010170  
[10903.339781] cdc_acm 2-1:1.0: ttyACM0: USB ACM device  
[10903.342985] usbcore: registered new interface driver cdc_acm  
[10903.342998] cdc_acm: USB Abstract Control Model driver for USB modems and ISD  
N adapters  
skrbnik@crunchbang10:~$
```

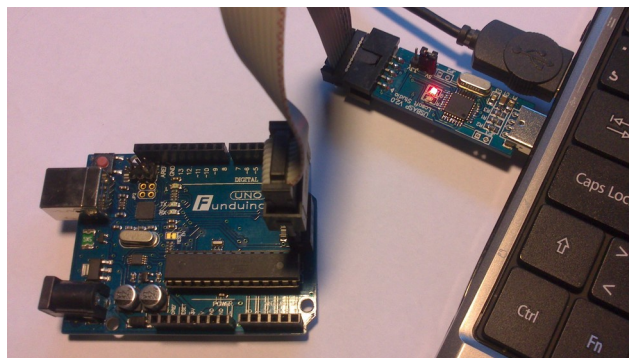
V drugem primeru imamo vmesnik Arduino povezan preko USBasp programatorja. Spet opazimo oznako `ttyACM*`, ki označuje USB vrata. Izpiše se nam tudi proizvajalec [USBasp programatorja](#).

```
skrbnik@crunchbang10: ~  
skrbnik@crunchbang10:~$ dmesg | tail  
[10902.992395] usb 2-1: SerialNumber: 55330313635351010170  
[10903.339781] cdc_acm 2-1:1.0: ttyACM0: USB ACM device  
[10903.342985] usbcore: registered new interface driver cdc_acm  
[10903.342998] cdc_acm: USB Abstract Control Model driver for USB modems and ISDN adapters  
[10999.464502] usb 2-1: USB disconnect, device number 3  
[11005.408411] usb 2-1: new low-speed USB device number 4 using uhci_hcd  
[11005.578258] usb 2-1: New USB device found, idVendor=16c0, idProduct=05dc  
[11005.578281] usb 2-1: New USB device strings: Mfr=1, Product=2, SerialNumber=0  
[11005.578297] usb 2-1: Product: USBasp  
[11005.578309] usb 2-1: Manufacturer: www.fischl.de  
skrbnik@crunchbang10:~$
```

Recimo, da imamo vse pripravljeno, da na mikrokrmilnik naložimo prvi program, s pomočjo programa AVRdude preko terminalnega okna.



Programiranje preko integriranega programatorja.



Programiranje z zunanjim programatorjem.

Dovoljenje za pisanje na USB vodilo

Preden prenašamo program na mikrokrmilnik, moramo v OS linux uporabniku dovoliti pisanje na USB vodilo. To storimo samo enkrat in sicer z naslednjim ukazom:

```
sudo usermod -a -G tty <uporabniško_ime>
```

Arudino UNO original ali pa s čipom CH341

Imamo več vrst Arduino vmesnikov, tiste z originalnim čipom za komunikacijo in tiste s čipom CH341. Poglejmo, kako računalnik vidi oba tipa:

V terminal vpišimo:

```
dmesg | tail
```

V primeru Arduino klona, se nam izpiše naslednje:

```
[17275.302131] usb 1-1: USB disconnect, device number 10
[17281.541093] usb 3-1: new full-speed USB device number 10 using uhci_hcd
[17281.728123] usb 3-1: New USB device found, idVendor=1a86, idProduct=7523, bcdDevice= 2.62
[17281.728126] usb 3-1: New USB device strings: Mfr=0, Product=2, SerialNumber=0
[17281.728128] usb 3-1: Product: USB2.0-Serial
[17281.731169] ch341 3-1:1.0: ch341-uart converter detected
[17281.737182] usb 3-1: ch341-uart converter now attached to ttyUSB0
tomaz@tomaz-pc:~$
```

V primeru originalnega Arduina se izpiše naslednje:

```
tomaz@tomaz-pc:~/Dokumenti/TEMP$ dmesg | tail
[17281.737182] usb 3-1: ch341-uart converter now attached to ttyUSB0
[27862.125142] usb 3-1: USB disconnect, device number 10
[27862.125376] ch341-uart ttyUSB0: ch341-uart converter now disconnected from ttyUSB0
[27862.125401] ch341 3-1:1.0: device disconnected
[28741.481083] usb 3-1: new full-speed USB device number 11 using uhci_hcd
[28741.708124] usb 3-1: New USB device found, idVendor=2341, idProduct=0043, bcdDevice= 0.01
[28741.708127] usb 3-1: New USB device strings: Mfr=1, Product=2, SerialNumber=220
[28741.708129] usb 3-1: Manufacturer: Arduino (www.arduino.cc)
[28741.708131] usb 3-1: SerialNumber: 75334323935351518032
[28741.711194] cdc_acm 3-1:1.0: ttyACM0: USB ACM device
tomaz@tomaz-pc:~/Dokumenti/TEMP$
```

OPOMBA: Pomemben je tisti del, kjer se izpiše ttyACM0, oziroma ttyUSB0. To si zapomnimo, saj bomo kasneje potrebovali pri prenosu .hex kode v mikrokontrolnik.

Moj prvi program

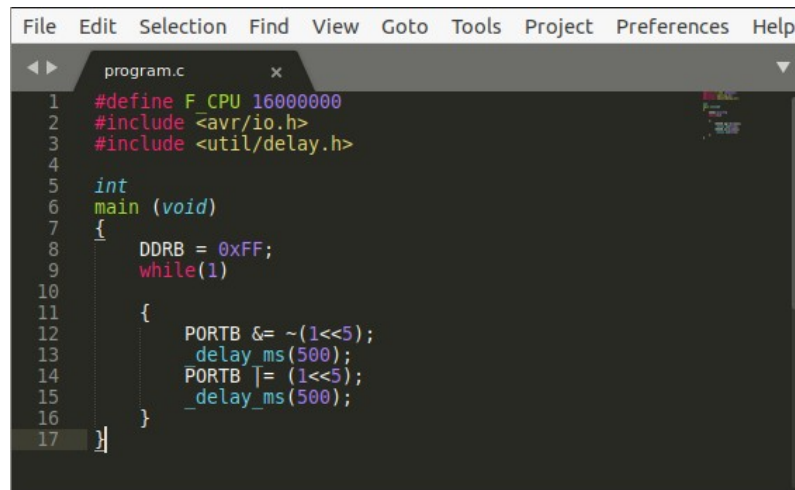
Odpremo program za branje in pisanje tekstovnih datotek (beležnico oziroma notepad). Lahko uporabimo program **Geany** ali **nano**, spodaj je zaslonska slika, kako je videti C koda v programu **Sublime**.

Napišimo enostaven program, da bo na vmesniku utripala LED dioda – Arduino vmesnik ima na ploščici LED diodo, prepoznali jo bomo po oznaki L. Ta dioda je povezana s priključkom B5. Zapišimo naslednji program:

```
#define F_CPU 16000000
#include <avr/io.h>
#include <util/delay.h>

int
main (void)
{
    DDRB = 0xFF;
    while(1)
    {
        PORTB &= ~(1<<5);
        _delay_ms(500);
        PORTB |= (1<<5);
        _delay_ms(500);
    }
}
```

Datoteko shranimo pod imenom `program.c`, torej shranimo jo s končnico `.c`.



Pretvorba tekstovne datoteke v .hex datoteko in programiranje

Da naš prvi program presnamemo v mikrokrmilnik, potrebujemo zapisati še nekaj ukazov.

Prevajanje programa v strojno kodo

Preko terminalnega okna se postavimo v mapo, kamor smo shranili naš program `program.c`. Nato prevedemo program, potrebna sta dva koraka:

```
avr-gcc -mmcu=atmega328p -Wall -Os -o ime_programa.elf ime_programa.c
```

in nato še:

```
avr-objcopy -j .text -j .data -O ihex ime_programa.elf ime_programa.hex
```

OPOMBA: Besedo `ime_programa` nadomestimo z imenom datoteke, v tem primeru `program.c`, `program.elf`, `program.hex`. To velja tudi za primere v nadaljevanju.

Prenos programa v mikrokrmilnik

Kot smo omenili imamo dve možnosti, preko USB priključka ali z zunanjim programatorjem.

Če programiramo direktno preko USB priključka, vnesemo naslednjo ukazno vrstico:

```
sudo avrdude -c arduino -p m328p -B 115200 -P /dev/ttyACM0 -U flash:w:program.hex:a
```

No, na tem mestu je pomembno, kateri vmesnik Arduino imamo, original ali clone.

Če imamo clone vmesnika Arduino, potem moramo najverjetneje spremeniti parameter USB vrat:

```
sudo avrdude -c arduino -p m328p -B 115200 -P /dev/ttyUSB0 -U flash:w:program.hex:a
```

OPOMBA: Pri parametrih moramo biti pozorni na velike in male črke!

Če programiramo z zunanjim **USBasp** programatorjem, vpišemo naslednji ukaz:

```
sudo avrdude -c usbasp -p m328p -B 19200 -P /ttyUSB0 -E reset -U flash:w:program.hex:a
```

Seveda moramo imeti ob programiranju priključen vmesnik oziroma programator in vmesnik, sicer nam računalnik vrne napako.

OPOMBA: Pri programiranju direktno preko USB priključka, moramo imeti na mikrokrmilniku ATmega328p nameščen ustrezen **bootloader**. Ko kupite nov Arduino vmesnik je bootloader že nameščen. Potrebno pa ga je na novo namestiti, če zamenjate mikrokrmilnik na vmesniku, ali če ste mikrokrmilnik preprogramirali z zunanjim programatorjem – takrat se je bootloader prepisal. Kako z zunanjim programatorjem namestimo bootloader, je opisano v poglavju **Bootloader**.

Tako, kar se tiče gonilnikov, programskih knjižnic imamo vse nameščeno. Če LED dioda utripa oziroma če smo uspešno prenesli program v mikrokrmilnik, potem je naša namestitvev uspešna.

Lahko pa sedaj programiranje še malenkostno poenostavimo. Namreč, pisanje programa v beležnici je težko, še posebej za začetnike. Poleg tega je pisanje ukaznih vrstic v terminalno okno zamudno.

To je vse. Sedaj pa je vse odvisno od volje za pisanje programov.

Bootloader

Vmesniki Arduino imajo prednaložen program, ki se imenuje Bootloader. Če povemo poenostavljeno, ta program omogoča, da lahko vmesnike programiramo brez zunanje programatorja.

Preko terminala se postavimo se v mapo z bootloader datoteko `optiboot_atmega328_novo.hex` in vpišemo naslednji ukaz, s katerim naložimo ustrezen bootloader.

```
sudo avrdude -c usbasp -p m328p -B 19200 -P /ttyUSB0 -V -u -U flash:w:optiboot_atmega328_novo.hex -U lock:w:0x0F:m
```

Vse dokler z zunanjim programatorjem ne naložimo kakšnega drugega programa, lahko programiramo Arduino vmesnik direktno preko USB konektorja ki je na vmesniku.



Slika 1: Zunanji programator USBasp, s katerim lahko naložimo bootloader, ali pa katerikoli drug program.