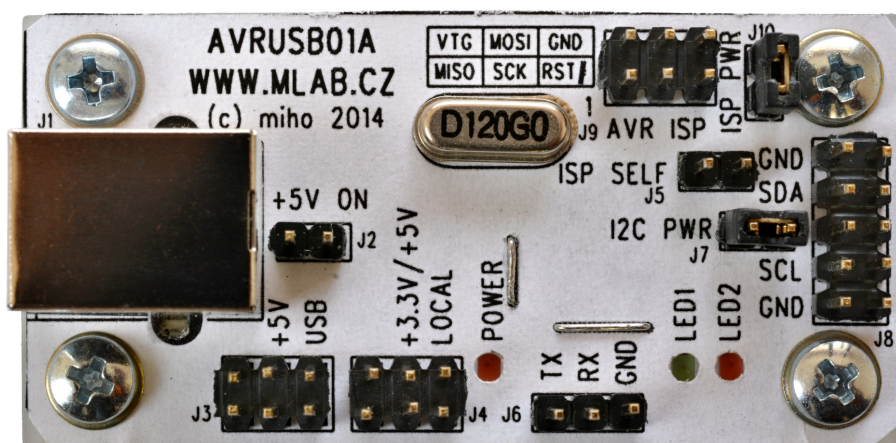


## Firmware USBasp pro modul AVRUSB

Milan Horkel

*Firmware USBasp umožňuje použít modul AVRUSB jako ISP programátor procesorů řady AVR pod všemi běžnými operačními systémy.*



### 1. Technické parametry

Parametr	Hodnota	Poznámka
Rozhraní	USB	Low Speed
Protokol	USBasp	VENDOR ID 0x16c0 DEVICE ID 0x05dc
Procesor	ATmega8 ATmega88	Varianty překladu
Podporované systémy	WindowXP/7/8 Linux/Ubuntu/... Android	avrdude avrdude ZFlasher AVR

## 2. Popis

Firmware USBasp je USB programátor pro procesory AVR s USB rozhraním realizovaným čistě programovými prostředky. Jako vhodný ovládací program lze použít rozšířený AVRDUDE. Firmware je přeložen ve variantách pro procesor ATmega8 a ATmega88.

### 2.1. Hardware

Konektor J9 slouží pro naprogramování firmwaru (při zapojené propojce J5, případně i J10), nebo pro připojení zařízení, které chceme programovat tímto USBasp programátorem. Zapojení je dle doporučení firmy Atmel (šestipinový ISP konektor).

Programátor má vyvedeno USB napětí +5V na konektoru J3 (přes samoopravnou pojistku) a vnitřní napětí na J4 (+5V z USB pokud je zapojen J2, nebo +3.3V z vnitřního stabilizátoru). Toto vnitřní napětí je možné připojit na ISP konektor J9 pomocí propojky J10.

Při programování cílového procesoru, který běží na nízkém kmotočtu je třeba snížit rychlost komunikace a toho lze docílit instalací propojky mezi piny 2 a 3 konektoru J6 (označené RX a GND).

Červená LED indikuje přenos dat, zelená LED svítí jakmile se zařízení zapne.

Propojka	Zapnuto	Vypnuto
J2	Napájení 5V z USB	Napájení 3.3V ze stabilizátoru
J5	Programování vlastního procesoru	Funguje jako programátor
J10	Napájení cílového zařízení	Bez napájení cílového zařízení
J6.2 - J6.3	Vnucený pomalý režim ISP	Rychlost přenosu nastavuje program

### 2.2. USBasp s Windows

Pod Windows je třeba nainstalovat driver libusb. Pro Windows 7/8 jsou novější verze knihovny libusb podepsané a tak nepůsobí potíže. Drivery vybalíme z příloženého originálního balíku, nebo stáhneme poslední verzi.

`SW/fw_usbasp/original/usbasp.2011-05-28.tar.gz`

Pokud používáte starší verzi programu avrdude, musíte použít starší verzi libusb (verze 0.x) a pod Windows 7/8 povolit použití nepodepsaných driverů. Nepodepsané drivery je možné povolit při startu systému pomocí volby F8 (při každém startu znovu).

**Ověříme funkčnost (z příkazové řádky):**

```
C:\Users\miho>avrdude -c USBasp -p ATmega88 -F
avrdude: AVR device initialized and ready to accept instructions
Reading | ##### |
100% 0.00s
avrdude: Device signature = 0x1e930a
avrdude: safemode: Fuses OK
avrdude done. Thank you.
C:\Users\miho>
```

## 2.3. USBasp s Linuxem

Pod linuxem by vše mělo fungovat bez komplikací. Program avrdude spouštíme s root oprávněním, aby se dostal k USB zařízení. Případně je možné systému říci, že pro toto zařízení není třeba root oprávnění vytvořením souboru /etc/udev/rules.d/99-USBasp.rules s tímto obsahem (maso je jedna dlouhá řádka):

```
# USBasp - USB programmer for Atmel AVR controllers
# Copy this file to /etc/udev/rules.d so

SUBSYSTEM=="usb", ENV{DEVTYPE}=="usb_device",
SYSFS{idVendor}=="16c0", SYSFS{idProduct}=="05dc", MODE="0666"
```

Vzorový soubor opět nalezneme v příloženém originálním balíku.

```
SW/fw_usbasp/original/usbasp.2011-05-28.tar.gz
```

**Ověříme funkčnost (z terminálu):**

```
miho@bobik:~$ avrdude -c USBasp -p ATmega88 -F
avrdude: AVR device initialized and ready to accept instructions
Reading | ##### |
100% 0.01s
avrdude: Device signature = 0x1e930a
avrdude: safemode: Fuses OK
avrdude done. Thank you.
miho@bobik:~$
```

## 2.4. USBasp s Androidem

Do Android zařízení nakoupíme aplikaci ZFlasher AVR (je zdarma) a přes OTG-USB kabel připojíme programátor a je to.

### 3. Programování firmwaru

Do modulu AVRUSB musíme neprogramovat firmware. Děláme to jiným programátorem přes konektor J9. Nutno zapojit propojky J5 (povolení programování) a J10 (propojení napájení přes ISP konektor). Pokud neumí použitý programátor napájet cílovou aplikaci přes ISP konektor, musíme zapojit napájecí zdroj +3.3V nebo +5V na napájecí konektor J4, nebo připojit USB kabel.

Dejte extra pozor při programování propojek ať si omylem nezakážete ISP programování. Raději dvakrát zkontrolujte typ procesoru a hodnotu propojek.

Kdo nechce hledat parametry programu avrdude, tady je příklad pro procesor ATmega88 (je to 1 dlouhá řádka a platí pro programátor picoweb na portu LPT1):

```
avrdude -c picoweb -p ATmega88 -P lpt1 -u -U hfuse:w:0xDE:m -U
lfuse:w:0xD7:m -U flash:w:usbasp_atmega88.hex
```

Procesor	Soubor	EFUSE	HFUSE	LFUSE
ATmega8	usbasp_atmega8.hex		HFUSE=0xC9	LFUSE=0x9F
ATmega88	usbasp_atmega88.hex	EFUSE=0xF9	HFUSE=0xDE	LFUSE=0xD7

## 4. Překlad firmwaru

### 4.1. Zdrojáky a úpravy

Zdojáky jsou z adresy <http://www.fischl.de/usbasp>

Provedené úpravy spočívají v opravě definic IO vstupů a výstupů (důsledné použití definic pinů) a v doplnění překladu o automatický překlad všech variant firmwarů najednou.

Pokud budete chtít upravovat zdroják pro jiný hardware, je třeba dát pozor na to, že definice hardware jsou rozstrkané do více zdrojáků. To jsem upravovat nechtěl, aby byla zachována co největší podobnost s původním stavem.

Konfiguraci firmwaru shrnuje tabulka

Parametr	Hodnota	Poznámka
USB D+	PD2	USB Data+, je současně signálem externího přerušení INTO
USB D-	PD4	USB Data-
LED Red	PC0	Červená LED, aktivní L, indikuje přenos dat
LED Green	PC1	Zelená LED, aktivní L, svítí trvale
SW	PD0	Zpomalení rychlosti ISP komunikace, aktivní L
Krystal	12.0MHz	Nutno naprogramovat FUSE na externí krystal
Procesor	ATmega8 ATmega88	

### 4.2. Překlad

Pro překlad je připravený Makefilepro je překlad variant pro procesory ATmega8 i ATmega88.

```
make all
```

Pro překlad pod Windows stačí nainstalovat balík WinAVR (mám WinAVR-20100110-install.exe).

Pro překlad pod Linuxem potřebujete balíky gcc-avr a avr-libc.