

TECHNICKÁ DOKUMENTÁCIA · PRG945

USB Blaster pre FPGA a CPLD Intel / Altera

Verzia s čipom WCH CH552G – ako funguje, zapojenie,
programovanie v Quartuse, riešenie problémov a best practices



Ilustračné foto modulu s čipom CH552G (doska Rev. C). Foto: github.com/rtek1000/USB-Blaster_CH552 (MIT).

Obsah

1	Prehľad a rýchly štart
2	Technické parametre
3	Ako to funguje
4	Konektor a zapojenie pinov
5	Pripojenie k cieľovej doske
6	Inštalácia ovládačov (Windows, Linux)
7	Programovanie v Quartuse
8	Riešenie problémov
9	Oprava / aktualizácia firmvéru CH552G
10	Best practices
11	Zdroje a ďalšie čítanie

O tomto dokumente

Tento dokument zhrňa overené postupy komunity pre lacné klony USB Blastera s čipom CH552G. Kľúčové zdroje, z ktorých čerpáme a na ktoré odkazujeme, sú:

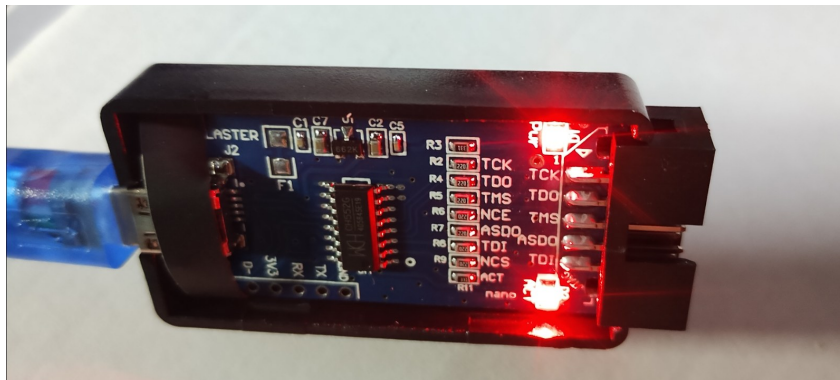
- vlákno na fóre EEVblog o probléme „No Hardware“ – [eevblog.com/forum/fpga/usb-blaster-\(ftdi245-vs-ch552x\)...](http://eevblog.com/forum/fpga/usb-blaster-(ftdi245-vs-ch552x)...)
- repozitár s opraveným firmvérom a návodom – github.com/rtek1000/USB-Blaster_CH552

1

Prehľad a rýchly štart

USB Blaster (katalógové číslo PRG945) je lacný programátor pre FPGA a CPLD obvody Intel (pôvodne Altera). Prevádza USB z počítača na rozhranie JTAG, cez ktoré program Quartus nahrá konfiguráciu priamo do obvodu alebo do jeho konfiguračnej pamäte. Hodí sa pre CPLD (napr. MAX II, MAX V) aj FPGA (napr. Cyclone, MAX 10).

Táto verzia je postavená na mikrokontroléri **WCH CH552G**. Jeho firmvér napodobňuje pôvodný USB-Blaster, takže sa v počítači hlási ako **Altera USB-Blaster (VID 09FB, PID 6001)** a používa rovnaký ovládač ako originál. Modul je navrhnutý pre USB 2.0; na portoch USB 3.0 môže mať problémy.



Obr. 1 – Doska v otvorenom puzdre: vľavo micro-USB, v strede CH552G, vpravo 10-pinový konektor s popisom signálov a LED indikátory. Foto: rtek1000/USB-Blaster_CH552 (MIT).

Rýchly štart

1. Nainštalujte **Intel Quartus Prime** alebo samostatný **Quartus Prime Programmer** (obsahuje ovládač USB-Blaster).
2. Pripojte programátor do **USB 2.0** portu. V Správcovi zariadení (Windows) alebo cez **lsusb** (Linux) overte zariadenie **09FB:6001**; ak chýba ovládač, postupujte podľa kapitoly 6.
3. Cieľovú dosku **zapnite z jej vlastného zdroja** a pripojte plochý kábel – pin 1 (TCK) na pin 1 konektora dosky.
4. V Quartus Programmer kliknite **Hardware Setup...**, vyberte **USB-Blaster [USB-0]**, režim **JTAG** a dajte **Auto Detect**.
5. Pridajte súbor (.sof / .pof / .jic), zaškrtnite **Program/Configure** a stlačte **Start**.

Najčastejší problém

Ak Quartus hlási „**No Hardware**“, hoci Správca zariadení programátor vidí, ide o známy problém klonov s CH552 – riešenie je v kapitolách 8 a 9.

2

Technické parametre

Parameter	Hodnota
Katalógové číslo	PRG945
Typ	kompatibilný klon Altera / Intel USB-Blaster (doska „Rev. C“)
Riadiaci čip	WCH CH552G – jadro 8051, USB 2.0 Full-Speed (12 Mbit/s), 14 KB pamäte programu
Identifikácia na USB	VID 09FB, PID 6001 (Altera USB-Blaster)
Bootloader CH552G	VID 4348, PID 55E0 (WinChipHead) – len pri aktualizácii firmvéru
Pripojenie k PC	USB 2.0, konektor micro-USB na doske; USB 3.0 porty môžu robiť problémy
Výstupné rozhranie	JTAG; režim AS (Active Serial) pri firmvéri s podporou AS
Konektor	10-pin IDC 2×5, rozteč 2,54 mm (štandard Altera)
Logické úrovne	pozri kapitolu 4 – pre 3,3 V logiku úprava podľa e-shopu
Podporované obvody	FPGA a CPLD Intel/Altera (napr. MAX II, MAX V, MAX 10, Cyclone), konfiguračné pamäte EPCS
Indikácia	LED POWER (napájanie), LED ACT (aktivita, pin P1.1)
Softvér	Intel Quartus Prime / Quartus Programmer (Windows, Linux); alternatívne OpenOCD, openFPGALoader
Napájanie	z USB; cieľová doska sa napája z vlastného zdroja
Hmotnosť	46 g (údaj e-shopu)

Údaje o čipe a identifikátoroch vychádzajú z firmvéru a výpisov v repozitári rtek1000 (nástroj wchisp hlási CH552, 14 KiB code flash, bootloader 02.50). Konkrétne kusy z rôznych výrobných sérií sa môžu líšiť – pri pochybnostiach overte vlastný kus.

3 Ako to funguje

JTAG je sériové rozhranie so štyrmi signálmi. Programátor posiela hodiny **TCK**, signálom **TMS** riadi stavový automat TAP v obvode, cez **TDI** posiela dáta a cez **TDO** ich číta späť. Obvody môžu byť zapojené do reťazca (TDO jedného ide na TDI ďalšieho), Quartus ich potom vidí všetky naraz.



TCK – hodiny · TMS – riadenie stavového automatu TAP · TDI – dáta do obvodu · TDO – dáta z obvodu

Obr. 2 – Tok dát od Quartusu po cieľový obvod.

Pôvodný USB-Blaster aj staršie klony používajú prevodník **FTDI FT245** a malé CPLD, ktoré generuje JTAG signály. Lacné klony s **CH552G** nemajú ani FTDI, ani CPLD – mikrokontrolér vo firmvéri napodobňuje komunikáciu FT245 a JTAG signály vytvára priamo na svojich pinoch (TCK, TDI, TDO cez hardvérové SPI). Ak je táto emulácia nepresná, Windows zariadenie rozpozná (správne VID/PID), ale Quartus s ním nedokáže komunikovať – to je podstata problému „No Hardware“ z vlákna na EEVblogu. Na trhu sa objavujú aj klony s čipom PIC18F14K50, ktoré podľa toho istého vlákna fungujú bez úprav.

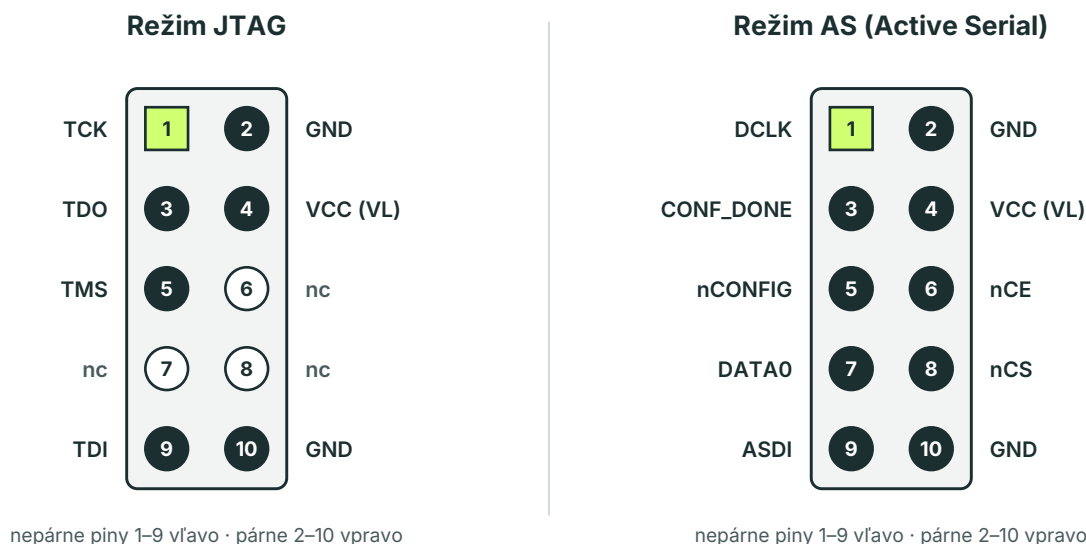
Priradenie pinov CH552G (firmvér rtek1000 / dsm)

Signál	Pin CH552G	Signál	Pin CH552G
TCK / DCLK	P1.7	TMS / nCONFIG	P3.2
TDI / ASDI	P1.5	ASDO	P3.3
TDO / CONF_DONE	P1.6	nCE	P3.4
nCS	P1.4	LED (ACT)	P1.1

Podľa repozitára je možné CH552G nahradiť CH551G alebo CH554G (rovnaký čip). Pôvodný firmvér VladimírDuan mal podľa používateľa „nirvana“ na EEVblogu nesprávne priradený pin TMS – opravené verzie sú v repozitároch dsm a rtek1000.

4 Konektor a zapojenie pinov

Programátor má 10-pinový konektor 2×5 s roztečou 2,54 mm v štandardnom zapojení Altera. Pin 1 je na puzdre označený trojuholníkom a na plochom kábli býva označený farebným (spravidla červeným) vodičom. Konektor má kľúč (výrez), ktorý bráni otočeniu kábla.



Obr. 3 – Zapojenie 10-pinového konektora pri pohľade na piny. Pin 1 je zvýraznený limetkovo, nc = nezapojené. Názvy podľa repozitára rtek1000 a potlače dosky.

Pin	JTAG	AS	Popis
1	TCK	DCLK	hodiny z programátora
2	GND	GND	zem
3	TDO	CONF_DONE	dáta z obvodu do programátora / stav konfigurácie
4	VCC (VL)	VCC (VL)	referenčné napätie I/O z cieľovej dosky (VCCIO)
5	TMS	nCONFIG	riadenie TAP / spustenie konfigurácie
6	nc	nCE	– / povolenie čipu
7	nc	DATA0 (ASDO)	– / dáta z konfiguračnej pamäte
8	nc	nCS	– / výber konfiguračnej pamäte
9	TDI	ASDI	dáta z programátora do obvodu / pamäte
10	GND	GND	zem

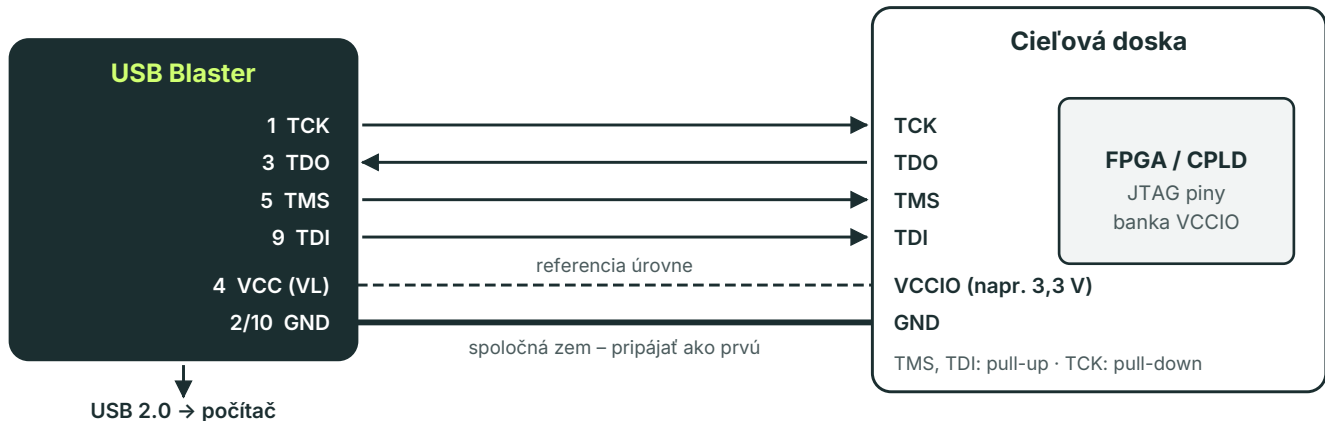
Logické úrovne a úprava pre 3,3 V

Na doske je pin 4 označený **VL**. Pri originálnom USB-Blasteri slúži ako referencia napätia výstupných budičov a napája sa z cieľovej dosky (VCCIO banky, kde sú JTAG piny).

Podľa popisu na e-shope je pri tomto module pre prácu s **3,3 V** logikou potrebné prispájkovať drôtik medzi plošku **3V3** a pin **VCC (VL)** konektora. Pozor: **pin 2 je podľa štandardu aj podľa potlače dosky GND** – spájať 3V3 s pinom 2 by znamenalo skrat. Pred úpravou si pinout na svojom kuse overte multimetrom a po úprave skontrolujte úroveň TCK proti GND (má byť $\approx 3,3$ V).

5

Pripojenie k cieľovej doske



Obr. 4 – Typické zapojenie. Šípky ukazujú smer signálov; GND je spoločná, VCC (VL) prenáša referenčné napätie z cieľovej dosky.

1. Odpojte napájanie cieľovej dosky a USB kábel programátora.
2. Nasuňte plochý kábel na konektor programátora a na JTAG konektor dosky – **pin 1 na pin 1** (šípka / červený vodič).
3. Zapnite napájanie cieľovej dosky. Programátor ju nenapája – potrebuje len referenciu na pine VCC (VL).
4. Pripojte USB kábel do počítača. Svetí LED POWER; LED ACT bliká počas komunikácie.
5. Pri odpájaní postupujte opačne: najprv USB / napájanie dosky, až potom plochý kábel.

Doska bez 10-pinového konektora

Ak doska nemá štandardný 10-pinový konektor (napr. vývojové dosky s jednoradovou lištou), prepojte ju vodičmi Dupont podľa tabuľky v kapitole 4. Vždy zapojte **GND ako prvú** a vodiče nechajte čo najkratšie.

6

Inštalácia ovládačov

Windows

Ovládač USB-Blaster je súčasťou inštalácie Quartus Prime aj samostatného Quartus Prime Programmer. Ak sa po pripojení nenainštaluje automaticky:

1. Otvorte **Správcu zariadení** a nájdite zariadenie USB-Blaster (prípadne neznáme zariadenie so žltým výkričníkom).
2. Vo vlastnostiach overte **Podrobnosti** → **ID hardvéru**: má byť **USB\VID_09FB&PID_6001**.
3. Zvoľte **Aktualizovať ovládač** → **Vyhľadať ovládač v počítači** a zadajte priečinok **<inštalácia Quartus>\drivers\usb-blaster**.
4. Po inštalácii odpojte a znovu pripojte programátor a reštartujte Quartus Programmer.

Tip z praxe

Používateľ „Mark“ na EEVblogu uvádza, že klony mu na Windows 10 začali fungovať až s **QuartusProgrammerSetup-23.1std**. Ak máte problém s jednou verziou, oplatí sa vyskúšať inú verziu samostatného Programmeru (najmä 23.1std).

Linux

Na Linuxe nie je potrebný ovládač jadra, Quartus však potrebuje práva na USB zariadenie. Vytvorte pravidlo udev (podľa repozitára rtek1000):

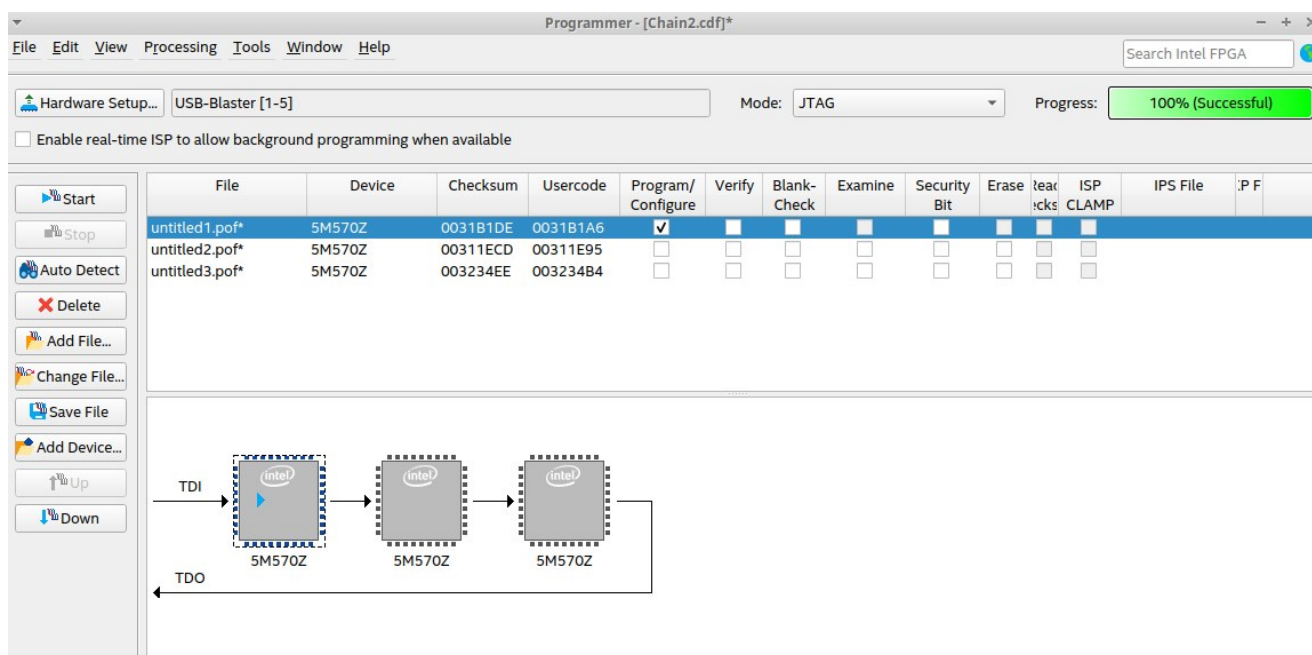
```
# /etc/udev/rules.d/51-usbblaster.rules
SUBSYSTEM=="usb", ATTR{idVendor}=="09fb", ATTR{idProduct}=="6001", MODE="0666"
SUBSYSTEM=="usb", ATTR{idVendor}=="09fb", ATTR{idProduct}=="6002", MODE="0666"
```

```
# načítanie pravidiel a kontrola
sudo udevadm control --reload-rules && sudo udevadm trigger
lsusb | grep 09fb # očakávané: ID 09fb:6001 Altera Blaster
killall jtagd # reštart JTAG servera po zmene práv
jtagconfig # vypíše programátor a obvody v reťazci
```

7 Programovanie v Quartuse

Postup v grafickom rozhraní

1. Spustíte **Tools** → **Programmer** (alebo samostatný Quartus Prime Programmer).
2. **Hardware Setup...** → v poli *Currently selected hardware* vyberte **USB-Blaster [USB-0]**.
3. Režim (**Mode**) nastavte na **JTAG** (alebo **Active Serial Programming** pre priame programovanie EPCS).
4. Kliknite **Auto Detect** – Quartus prečíta ID obvodov v reťazci. Ak sa obvod nenájde, programovať zatiaľ nemá zmysel (pozri kap. 8).
5. Pri obvode zvolíte **Change File...**, vyberte výstupný súbor, zaškrtnite **Program/Configure** (a ideálne **Verify**) a stlačte **Start**.



Obr. 5 – Quartus Programmer 22.1 s preflashovaným CH552G klonom: reťazec troch CPLD MAX V (5M570Z), naprogramované na 100 %. Snímka: rtek1000/USB-Blaster_CH552 (MIT).

Ktorý súbor použiť

Súbor	Kam sa zapíše	Kedy použiť
.sof	do konfiguračnej SRAM FPGA	rýchle testovanie; po vypnutí napájania sa stratí
.pof	do internej flash CPLD / MAX 10, alebo do EPCS v režime AS	trvalé uloženie; pre MAX II / MAX V vždy .pof
.jic	do konfiguračnej flash cez JTAG (nepriamo cez FPGA)	trvalé uloženie, keď doska nemá AS konektor

Príkazový riadok

zoznam programátorov a obvodov

jtagconfig

quartus_pgm -l

FPGA – nahratie do SRAM

quartus_pgm -c "USB-Blaster [USB-0]" -m JTAG -o "p;output_files/projekt.sof"

CPLD / MAX 10 – trvalé naprogramovanie s overením

quartus_pgm -c "USB-Blaster [USB-0]" -m JTAG -o "pv;output_files/projekt.pof"

konfiguračná flash cez JTAG (súbor .jic)

quartus_pgm -c "USB-Blaster [USB-0]" -m JTAG -o "pvi;output_files/projekt.jic"

Bez Quartusu

Ako alternatívu k Quartusu môžete skúsiť open-source nástroje **openFPGALoader** (kábel **usb-blaster**) alebo **OpenOCD** (ovládač **usb_blaster**). Keďže ide o emuláciu, kompatibilita s klonmi CH552 nie je zaručená – overte si ju na vlastnom kuse.

8 | Riešenie problémov

Príznak	Pravdepodobná príčina	Riešenie
Quartus: „ No Hardware “, Správca zariadení pritom ukazuje USB-Blaster (VID_09FB&PID_6001)	nepresná emulácia vo firmvéri CH552 klonu (EEVblog)	1) v Hardware Setup vybrať USB-Blaster; 2) skúsiť inú verziu Quartus Programmer (napr. 23.1std); 3) aktualizovať firmvér – kap. 9
Neznáme zariadenie so žltým výkričníkom	chýba ovládač	inštalovať z priečinka drivers\usb-blaster (kap. 6)
Programátor sa neobjaví vôbec / náhodne sa odpája	USB 3.0 port, predlžovačka, zlý kábel	použiť USB 2.0 port (príp. USB 2.0 hub), kratší a kvalitný dátový kábel
Linux: <i>No JTAG hardware available</i>	chýbajú práva (udev) alebo beží starý jtagd	pridať pravidlo udev, killall jtagd , znovu jtagconfig
Auto Detect: <i>Unable to scan device chain / JTAG chain broken</i>	doska nie je napájaná, otočený kábel, chýba VCC (VL), nesprávne logické úrovne	skontrolovať napájanie, orientáciu pinu 1, napätie na pine 4; osciloskopom overiť TCK/TMS/TDI (tip z EEVblogu)
Programovanie zlyhá v priebehu / chyba pri Verify	dlhý alebo poškodený kábel, rušenie, slabá zem	skratiť kábel, pridať GND vodič, napájať dosku stabilným zdrojom
Bootloader CH552G sa nezobrazí (4348:55E0)	prepojka odstránená príliš skoro alebo neskoro	zopakovať; čip po chvíli nečinnosti bootloader sám ukončí – príkaz majte pripravený vopred
wchisp na Windows: chyba CH375DLL64.dll	chýba knižnica WCH	stiahnuť DLL z webu WCH, alebo nainštalovať ovládač WinUSB cez Zadig

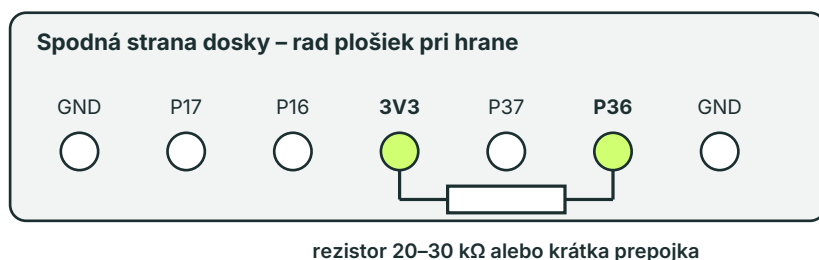
9 | Oprava / aktualizácia firmvéru CH552G

Ak programátor so žiadnou verziou Quartusu nefunguje, dá sa do CH552G nahráť opravený open-source firmvér. Jeho história: pôvodný projekt **VladimirDuan/CH55x-USB-Blaster** → opravy od **dougg3** (článok Downtown Doug Brown) → repozitár **dsm/USB-Blaster_CH552** s automatickým buildom, podporou režimu AS a hardvérového SPI → fork **rtek1000** s podrobným návodom a hotovým súborom **BIN/usb_blaster.hex**. Podľa autorov firmvér funguje na Linuxe aj vo Windows bez špeciálnych ovládačov.

Skôr než začnete

Zmena firmvéru je zásah na vlastné riziko a pôvodný firmvér sa spravidla nedá zálohovať. Ak modul nefunguje hneď po kúpe, **kontaktujte nás skôr, ako začnete firmvér meniť**.

Vstup do bootladera



P36 = USB D+

1. Prepojte 3V3 a P36 (odpojené od USB).
2. Pripojte USB kábel.
3. Asi po 1 s prepojenie odstráňte.
4. Čip sa hlási ako 4348:55E0 (WinChipHead – bootloader).

Obr. 6 – Prepojenie plošiek 3V3 a P36 (USB D+) pri pripájaní USB uvedie CH552G do bootladera. Na EEVblogu sa odporúča rezistor 20–30 kΩ, repozitár rtek1000 uvádza priame skratovanie.



Obr. 7 – Spodná strana dosky: rad plošiek GND, P17, P16, 3V3, P37, P36, GND a plošky USBISP (VCC, D-, D+, GND). Foto: rtek1000/USB-Blaster_CH552 (MIT).

Možnosť A – Windows, WCHISPTool (podľa EEVblogu)

1. Stiahnite a nainštalujte oficiálny [WCHISPTool](#) od výrobcu čipu.
2. Stiahnite firmvér – hotový `usb_blaster.hex` z priečinka BIN v [repozitári rtek1000](#), alebo binárky z releases repozitára [dsm/USB-Blaster_CH552](#).
3. Otvorte puzdro, prepojte **D+ (P36)** s **3,3 V** rezistorom 20–30 kΩ a pripojte USB.
4. WCHISPTool by mal rozpoznať CH552. Vyberte súbor firmvéru v sekcii sťahovania a stlačte **Download**.
5. Odpojte USB, odstráňte rezistor a pripojte znova – zariadenie sa hlási ako 09FB:6001.

Možnosť B – Linux / macOS, wchisp (podľa rtek1000)

Príkazy

```
# 1) firmvér: hotový hex, alebo kompilácia (SDCC 4.0+)
git clone https://github.com/rtek1000/USB-Blaster_CH552
cd USB-Blaster_CH552 && make # vznikne usb_blaster.hex

# 2) bootloader: prepojiť 3V3–P36, pripojiť USB, po ~1 s prepojenie odstrániť
lsusb # ID 4348:55e0 WinChipHead

# 3) nahratie (wchisp: github.com/ch32-rs/wchisp)
sudo ./wchisp info
sudo ./wchisp flash usb_blaster.hex # očakávané: Verify OK
```

- Bootloader sa po krátkej nečinnosti sám ukončí. Príkaz si napíšte vopred, pripojte kábel, odstráňte prepojkú a hneď potvrdte Enter.
- Ak príkazy zlyhávajú kvôli ceste k priečinku, presuňte súbory napr. do **/tmp** (poznámka z rtek1000).
- Po nahratí pridajte pravidlo udev (kap. 6) a overte funkčnosť príkazom **jtagconfig**.

10**Best practices**

Napájanie a zem

- Cieľovú dosku napájajte vlastným stabilným zdrojom; programátor ju nenapája.
- GND pripájajte ako prvú a odpájajte ako poslednú. Nepripájajte ani neodpájajte JTAG kábel pod napätím.
- Pin VCC (VL) má dostávať napätie banky VCCIO, v ktorej sú JTAG piny (napr. 3,3 V). Na I/O FPGA nikdy nepúšťajte 5 V.

Kábel a USB

- Plochý kábel majte čo najkratší (ideálne do 25 cm) a nepoškodený; dlhé vodiče Dupont sú častým zdrojom chýb.
- Používajte USB 2.0 port priamo na počítači, nie predlžovačky ani pasívne rozbočovače.

Návrh vlastnej dosky

- Dodržte zapojenie 10-pinového konektora Altera (kap. 4) a použite konektor s kľúčom.
- JTAG signály ošetríte podľa príručky konkrétnej rodiny obvodov – typicky pull-up na TMS a TDI a pull-down na TCK.
- Pri viacerých obvodoch v reťazci vedite TCK a TMS ku všetkým obvodom a TDO→TDI v poradí reťazca.
- Ak plánujete konfiguračnú flash, vyvedte aj AS konektor, alebo počítajte s programovaním .jic cez JTAG.

Práca v Quartuse

- Pred programovaním vždy spustíte Auto Detect (alebo **jtagconfig**) – overíte reťazec aj kábel.
- Pri trvalom programovaní (.pof, .jic) zapínajte Verify.
- Používajte ovládač a Programmer z tej istej verzie Quartusu; pri problémoch s klonom skúste verziu 23.1std.

Manipulácia

- Pracujte s ochranou proti elektrostatickému výboju, najmä pri otvorenom puzdre.
- Do firmvéru zasahujte, len ak modul s Quartusom nefunguje – a podľa postupu v kapitole 9.

11

Zdroje a ďalšie čítanie

Zdroj	Čo tam nájdete
Techfun – USB blaster ALTERA pre FPGA čipy	produktová stránka, parametre, poznámka k 3,3 V
EEVblog: USB BLASTER (FTDI245 vs ch552x) Quartus „No Hardware“	popis problému, tip na Quartus 23.1std, postup s WCHISPTool, odkazy na firmvér
github.com/rtek1000/USB-Blaster_CH552	opravený firmvér, hotový .hex, pinout, návod na bootloader, wchisp a udev
github.com/dsm/USB-Blaster_CH552	pôvodný fork s automatickým buildom, podporou AS a HW SPI
github.com/VladimirDuan/CH55x-USB-Blaster	pôvodný firmvér USB-Blaster pre CH55x
Downtown Doug Brown: Fixing a knockoff Altera USB Blaster	článok o oprave klonu, na ktorý odkazuje repozitár
github.com/ch32-rs/wchisp	nástroj na nahratie firmvéru cez USB bootloader WCH
WCH – WCHISPTool	oficiálny nástroj WCH pre Windows
gist TCWORLD – USB-Blaster s PIC18F14K50	firmvér a schéma variantu s PIC (pre porovnanie)

Kód firmvéru v repozitári je pod licenciou MIT. Fotografie a snímky obrazovky v tomto dokumente pochádzajú z repozitára rtek1000/USB-Blaster_CH552. Postupy sú prevzaté zo zdrojov vyššie a doplnené o všeobecne platné zásady práce s JTAG; Techfun nie je ich autorom ani výrobcom originálneho USB-Blastera.



Váš nápad, náš hardvér



Techfun s. r. o.

Kopčianska 92A, 851 01 Bratislava



techfun@techfun.sk



+421 952 559 014

Predajňa: pracovné dni 7:30 – 17:00

Produkt: techfun.sk/produkt/usb-blaster-altera-pre-fpga-cipy/