



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV MIKROELEKTRONIKY

DEPARTMENT OF MICROELECTRONICS

## PROGRAMÁTOR OBVODŮ FPGA

FPGA PROGRAMMER

### BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

### AUTOR PRÁCE

AUTHOR

Jan Sedlář

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Marek Bohrn, Ph.D.

BRNO 2016



# Bakalářská práce

bakalářský studijní obor **Mikroelektronika a technologie**  
Ústav mikroelektroniky

**Student:** Jan Sedlář

**ID:** 164777

**Ročník:** 3

**Akademický rok:** 2015/16

**NÁZEV TÉMATU:**

## Programátor obvodů FPGA

**POKYNY PRO VYPRACOVÁNÍ:**

V rámci závěrečné bakalářské práce provedte návrh a realizaci programátoru obvodů FPGA. Programátor bude připojen k PC přes rozhraní USB, k programovaným obvodům přes rozhraní JTAG. Programátor by měl podporovat obvody FPGA od spol. Lattice, Altera a Xilinx.

V rámci návrhu se zaměřte na minimalizaci počtu potřebných součástek, plochy a ceny.

Zvažte možnost realizace dalších podpůrných funkcí (např. sériový přenos dat mezi FPGA a PC, virtuální prepínače a zobrazovače, programovatelné uživatelské vstupy a výstupy). Tyto funkce nesmí mít zásadní vliv na počet potřebných součástek a zvyšovat náročnost obvodové realizace.

**DOPORUČENÁ LITERATURA:**

Podle pokynů vedoucího práce.

**Termín zadání:** 8.2.2016

**Termín odevzdání:** 2.6.2016

**Vedoucí práce:** Ing. Marek Bohm, Ph.D.

**Konzultant bakalářské práce:**

**doc. Ing. Jiří Háze, Ph.D., předseda oborové rady**

**UPOZORNĚNÍ:**

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## Abstrakt

Práce se zabývá rozborem realizace FPGA programátorů. Jejím cílem je vytvořit funkční programátor. Návrh je zaměřen na minimalizaci součástek, jejich ceny a plochy. Programátor by měl podporovat obvody od výrobců Altera, Lattice a Xilinx. Programátor by měl umožňovat podpůrné funkce jako například přenos dat, virtuální přepínače a programovací tlačítko.

## Klíčová slova

FPGA, FTDI , JTAG, SVF, Programátor

## Abstract

The work deals with the analysis of the realization of FPGA programmers. Its aim is to create a functional programmer. The proposal aims to minimize the components, their cost and area. The programmer should support chips from manufacturers Altera, Lattice and Xilinx. The programmer should enable support functions such as data transfer, virtual switches and the programming button.

## Keywords

FPGA, FTDI , JTAG, SVF Programmer

SEDLÁŘ, J. *Programátor obvodů FPGA*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Ústav mikroelektroniky, 2016. 44 s. Bakalářská práce. Vedoucí práce Ing. Marek Bohrn, Ph.D.

## PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma Programátor obvodů FPGA jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a-jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne .....

.....

(podpis autora)

## Poděkování:

Děkuji Ing. Markovi Bohrnovi, Ph.D. za vedení mé bakalářské práce.

V Brně dne .....

.....

(podpis autora)

## OBSAH

<b>ÚVOD</b> .....	<b>7</b>
<b>1 ROZBOR PROGRAMÁTORŮ</b> .....	<b>8</b>
1.1 EXTERNÍ PROGRAMÁTORY .....	8
1.1 ZABUDOVANÉ PROGRAMÁTORY .....	10
1.2 POUŽÍVANÉ OBVODY .....	11
1.3 SHRUTÍ.....	13
<b>2 STANDARD JTAG</b> .....	<b>15</b>
2.1 PRINCIP.....	15
2.2 SIGNÁLY.....	17
2.3 STAVOVÝ AUTOMAT .....	18
<b>3 NÁVRH PROGRAMÁTORU</b> .....	<b>20</b>
3.1 BLOKOVÝ NÁVRH .....	20
3.2 REALIZACE PROGRAMÁTORU.....	22
3.2.1 <i>Software</i> .....	22
3.2.1 <i>Hardware</i> .....	26
3.3 VÝSLEDKY .....	28
<b>4 ZÁVĚR</b> .....	<b>30</b>
<b>LITERATURA</b> .....	<b>31</b>
<b>SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK</b> .....	<b>32</b>
<b>SEZNAM OBRÁZKŮ</b> .....	<b>34</b>
<b>SEZNAM TABULEK</b> .....	<b>35</b>
<b>PŘÍLOHY</b> .....	<b>36</b>

# Úvod

V minulosti měly obvody přesně dané funkce. Ale s postupem času se začaly vyvíjet obvody konfigurovatelné, které poskytují výhody při návrhu a zároveň šetří náklady. Všechny konfigurovatelné obvody se však musí nejprve nakonfigurovat. Pro tento účel slouží programátory. Programátory jsou dostupné od výrobců obvodů, ale s celkovou cenou mnohonásobně převyšující náklady na jejich výrobu a materiál. Zároveň jsou fixovány na vývojové prostředí a jeho obvody. Takové obvody jsou programovány způsobem jak proprietálním, tak standardizovaným. Je proto pochopitelné, že je těžké vytvořit univerzální programátor. Moderní obvody využívají typicky rozhraní JTAG. Toto rozhraní lze kromě samotného programování využít i pro testování obvodů.

Práce si dává za cíl navrhnout programátor pro obvody FPGA. Programátor má umožnit programovat obvody od firem Altera, Xilinx, Lattice a umožnit, aby byl řízen přes PC. Programátor má být malý, levný a má podporovat možnost integrace do systému s programovaným FPGA, jako příklad lze uvést vývojové desky pro studenty.

Práce začíná rozбором existujících řešení jednotlivých programátorů a jejich obvodů, kterými by bylo možné realizovat vlastní programátor. Před realizací bude prozkoumán standard JTAG a jeho funkce pro následné využití. Na základě získaných informací je proveden návrh obvodového řešení programátoru a potřebného programového vybavení. Vzhledem k tomu, že některé programátory obvodů FPGA na trhu obsahují i rozhraní pro přenášení dat mezi PC a FPGA, je zvažena i možnost realizace těchto funkcí v navrhovaném programátoru.

# 1 Rozbor programátorů

Existuje mnoho možností, jakým způsobem programovat a zkonstruovat desku. Například samotný programátor může být zabudován ve finálním produktu nebo fungovat jako externí zařízení. Dále můžeme zvážit, jak jsou data do programátoru přenášena, jestli pomocí USB, paralelního portu, ethernet nebo jinou metodou. Způsob programování obvodu může mít zásadní vliv na jeho rychlost a bezpečnost programování. Způsoby zahrnují JTAG, SPI a různé speciální módy od různých firem, například AS Mode a PS Mode od Altery. Je nutné vzít v potaz zabezpečení proti krádeži firmware, hlavně kvůli použití běžně dostupných součástek.

Externí programátor je samostatné zařízení, které je zapojeno mezi počítačem a programovatelným obvodem. Externí programátor obvykle umožňuje programovat různé obvody, typicky od jednoho výrobce. Tyto obvody mohou mít různá napájecí napětí a různé rozmístění programovacích pinů. Proto také bývají složitější a mívají více konektorů.

Využívají se hlavně při výrobě a vývoji. Výhodou je jeden konektor na desce pro naprogramování celé desky. To nám snižuje potřebnou plochu a výdaje, které by se jinak zaplatili, za zabudovaný programátor.

Zabudovaný programátor se nachází na desce spolu s programovatelnými obvody. Má za úkol naprogramovat pouze obvody na desce, nemusí podporovat další obvody a napětí. Celkový návrh je tím snadnější.

Takové programátory bývají využívány ve vývojových deskách nebo v zařízení, ve kterých je možné přeprogramovat firmware.

V následující podkapitole jsou uváděny příklady programátorů a jejich dovedností.

## *1.1 Externí programátory*

V následujících odstavcích jsou rozebrány příklady externích programátorů.

### **Cable USB II**

Značkový programátor pro Xilinx zařízení s velmi omezenou podporou pro cizí zařízení. Umožňuje programování FPGA, CPLD a PROM od Xilinx. Umožňuje programování flash zařízení třetích stran pomocí SPI. [3]

### **Bus Blaster v3**



Univerzální Open Source programátor a debugger pro JTAG zařízení. Lze s ním programovat a ladit ARM procesory, CPLD, FPGA, FLASH a další.

Pro univerzálnost je přidán programovatelný CPLD, který lze přizpůsobit cílové aplikaci. Jako programovací software lze použít UrJTAG a pro ladění OpenOCD.

K programování je zapotřebí BSDL soubor, který je dostupný od výrobců FPGA. [4]

### **USB Blaster**

Programátor od Altery, který je určený pro obvody téže společnosti. Má tři režimy: JTAG, PS a AS mód. [5]

### **HW-USBN-2B a HW-DLN-3C**

Programátory od LATTICE SEMICONDUCTOR. HW-USBN-2B používá na přenos dat USB a HW-DLN-3C paralelní port. HW-USBN-2B využívá obvod FT2232H. [13]

### **Silicon Sculptor 3**

Programátor od Microsemi, který nabízí zápis s klíčem nebo permanentním zámekem, který znemožňuje jakékoliv operace typu čtení, verifikace a zápis.[14]

Tab. 1: Souhrn vlastností externích programátorů

	Bus Blaster v3	Cable USB II	USB Blaster
Programovací módy	JTAG, KT-link	JTAG, SERIAL, SPI	JTAG, PS, AS
Podporovaná napětí desek s FPGA	1.5 V – 3.3 V	1.5 V – 5 V	1.5 V – 5 V
JTAG CLK	MAX 6 MHz	750KHz – 24MHz	--
Podporuje software	UrJTAG, OpenOCD	ISE webPACK, ISE impact	Quartus II.
Podporuje FPGA	Xilinx, Altera, Lattice	Xilinx	Altera
Použité obvody	FT2232H	CY7C68013A	FT245BL
Cena	\$34.95	\$225	\$300

## 1.1 Zabudované programátory

V následujících odstavcích jsou rozebrány příklady zabudovaných programátorů.

### Vývojová deska Cmod S6 od digilent

Vývojová deska se Spartanen 6 a programátorem na desce, kde mikrokontrolér zastává funkci převodníku.

Kromě rozhraní JTAG pro programování nabízí také paralelní 8 bitové rozhraní mezi převodníkem a FPGA. Rozhraní jde ovládat přes počítač pomocí software od Digilent.

Deska chytrě využívá jeden 8 MHz krystal jak pro mikrokontrolér, tak pro FPGA, čímž je ušetřen jeden krystal.

Deska může být napájena skrze USB nebo z externího zdroje 5 až 15V . Oba napájecí zdroje mohou být současně připojeny, vstup je chráněn dvěma diodami fungujícími jako OR brána. Napájecí napětí je přeměněno na 3.3V a 1.2V skrz buck měnič pro vyšší účinnost. [6]

Tab. 2: Vývojová deska Cmod S6

Napájení	5 až 15V
Převodník	ATmega16u2
Cena desky	\$69

## Vývojová deska Atlys od digilent

Vývojová deska se Spartanen 6 a programátorem na desce. Obsahuje 1 GB DDR2 paměti, 128 MB FLASH paměti, ethernet, audio, USB-UART převodník a základní I/O.

Deska podporuje programovací software Adept od Digilentu a Xilinx's impact. [12]

Tab. 3: Vývojová deska Atlys

Napájení	Externích 5V
Převodník	CY7C68013A
Cena desky	\$419.00

## 1.2 Používané obvody

Na základě rozboru existujících řešení byly vybrány často používané obvody jak z externích, tak ze zabudovaných programátorů a byly detailněji rozebrány. Nakonec byla vytvořena tabulka srovnávající vlastnosti obvodů viz. Tab. 4 a Tab. 5.

### FT2232H

Podporuje USB 2.0 Hi-speed (480Mb/s)

Má dva nezávislé kanály. Každý kanál může být nastaven na JTAG, SPI, I2C, Bit-Bang, Uart, FIFO a další. Je potřeba, aby byly nainstalovány ovladače. K řízení obvodů lze použít knihovnu od výrobce FTDI - FTD2XX.DLL nebo open source knihovnu libusb. K plné funkčnosti obvodu je zapotřebí 12MHz krystal, EEPROM a regulátor napětí.

EEPROM lze vynechat, ale pak se ztratí možnost nakonfigurovat USB VID, PID, sériové číslo a možnosti konfigurace módů se zmenší. Bez EEPROM se nastaví oba kanály na asynchronně sériový přenos s možností změnit pouze jeden kanál. Přehled viz Tab. 4.[7]

Tab. 4: Nastavitelné módy převodníku

	Async serial UART	ASYNC 245 FIFO	SYNC 245 FIFO	ASYNC Bit-Bang	SYNC Bit-Bang	MPSSE	Fast Serial Interface	Cpu-style FIFO	Host bus emulation
EPROM osazena	Ano	Ano	Ano				Ano	Ano	
EPROM neosazena			Ano	Ano	Ano	Ano			Ano

## **FT245B**

Podporuje USB 2.0 Full Speed (12Mb/s). K dispozici má dva módy, paralelní FIFO nebo Bit-Bang. K činnosti potřebuje krystal a EEPROM je optimální. Pro naprogramování FPGA se přepne do módu Bit-Bang, nahraje firmware a po skončení se přepne na FIFO, kde si s počítačem může předávat data.

FT245R je doporučený jako náhrada staršího FT245B. Obvod integruje oscilátor spolu s EEPROM do jediného celku a minimalizuje podpůrné obvody.

Cena je \$4,5 za náhradu, původní obvod je nedostupný. [8] [9]

## **ATmega16u2**

Osmibitový mikrokontrolér s USB 2.0 Full Speed (12 Mb/s). Disponuje základními perifériemi. Řídící firmware musí být naprogramováno pro funkci obvodu. Je nutné mít znalost USB nebo použít open source knihovnu LUFA.

K provozu USB je potřebná frekvence  $48\text{Mhz} \pm 0.25\%$ . Pro tak přesnou frekvenci je nutné použít externí krystal. [10]

## **CY7C68013A**

Mikrokontrolér s jádrem 8051, který podporuje USB 2.0. Cypress poskytuje vývojový nástroj pro tento obvod. Má nastavitelné všeobecně programovatelný interface a master/slave FIFO. Obvod je zamýšlen pro použití v tiskárnách, skenerech, mobilech audio atd. [11]

Tab. 5: Pouzdra a ceny obvodů

	<b>FT2232H</b>	<b>FT245B</b>	<b>FT245R</b>	<b>ATmega16u2</b>	<b>CY7C68013A</b>
QFN – 32	-	5x5 mm	5x5mm	5x5 mm	-
QFN – 56	-	-	-	-	8x8mm
QFN – 64	9x9 mm	-	-	-	-
TQFP - 32	-	-	-	9x9 mm	-
LQFP- 32	-	9x9 mm	-	-	-
LQFP- 64	12x12 mm	-	-	-	-
SSOP – 28		-	8.2x10.5mm	-	-
SSOP – 56	-	-	-	-	10.7x18.3mm
VFBGA – 56	-	-	-	-	5x5mm
Cena	~\$6.7	Nevyrábí	~\$4.5	~\$5.7	~\$9.9

Tab. 6: Vlastnosti obvodů

	<b>FT2232H</b>	<b>FT245B</b>	<b>FT245R</b>	<b>ATmega16u2</b>	<b>CY7C68013A</b>
Napětí Vcc	2.97–3.63V	4.35–5.25V	4–5.25V	2.7–5.5V	3–3.6V
Krystal	Externí	Externí	Interní	Externí	Externí
EEPROM	Možný	Možný	Ne	Ne	Ne

EEPROM není vždy nutná a pro účely programátoru může být vynechána pro zmenšení plochy obvodu viz. Tab. 4 Výhoda interního krystalu je stejná jak u EEPROM, zabírá méně plochy.

### ***1.3 Shrnutí***

Z vypracovaných tabulek je vidět, že k návrhu programátoru lze přistoupit více způsoby, buď mikrokontrolérem, nebo USB převodníkem s možností volby funkce chování obvodu.

Možnost s mikrokontrolérem si vyžaduje naprogramovat firmware, což je časově náročnější s možností zanesení softwarové chyby. Takové řešení má výhody v menší ceně obvodu a je vhodné pro velice malé desky s vlastním programátorem. Takový mikrokontrolér by měl obsahovat od výroby nahraný bootloader, aby nebylo nutné vlastnit programátor pro daný obvod.

U hotového převodníku se není třeba zabývat firmware, zde stačí nastavit funkce obvodu přes počítač a o zbytek se postará řídicí program. Nevýhodou některých USB převodníků je nutnost mít nainstalované ovladače, tomu se ale dá zamezit, pokud by převodník měl vlastní USB HID třídu.

Některé programátory nabízejí bonusové funkce jako je externí programovací tlačítko pro zrychlení práce s programátorem, nebo sériový port mezi FPGA a PC a virtuální přepínače a zobrazovače.

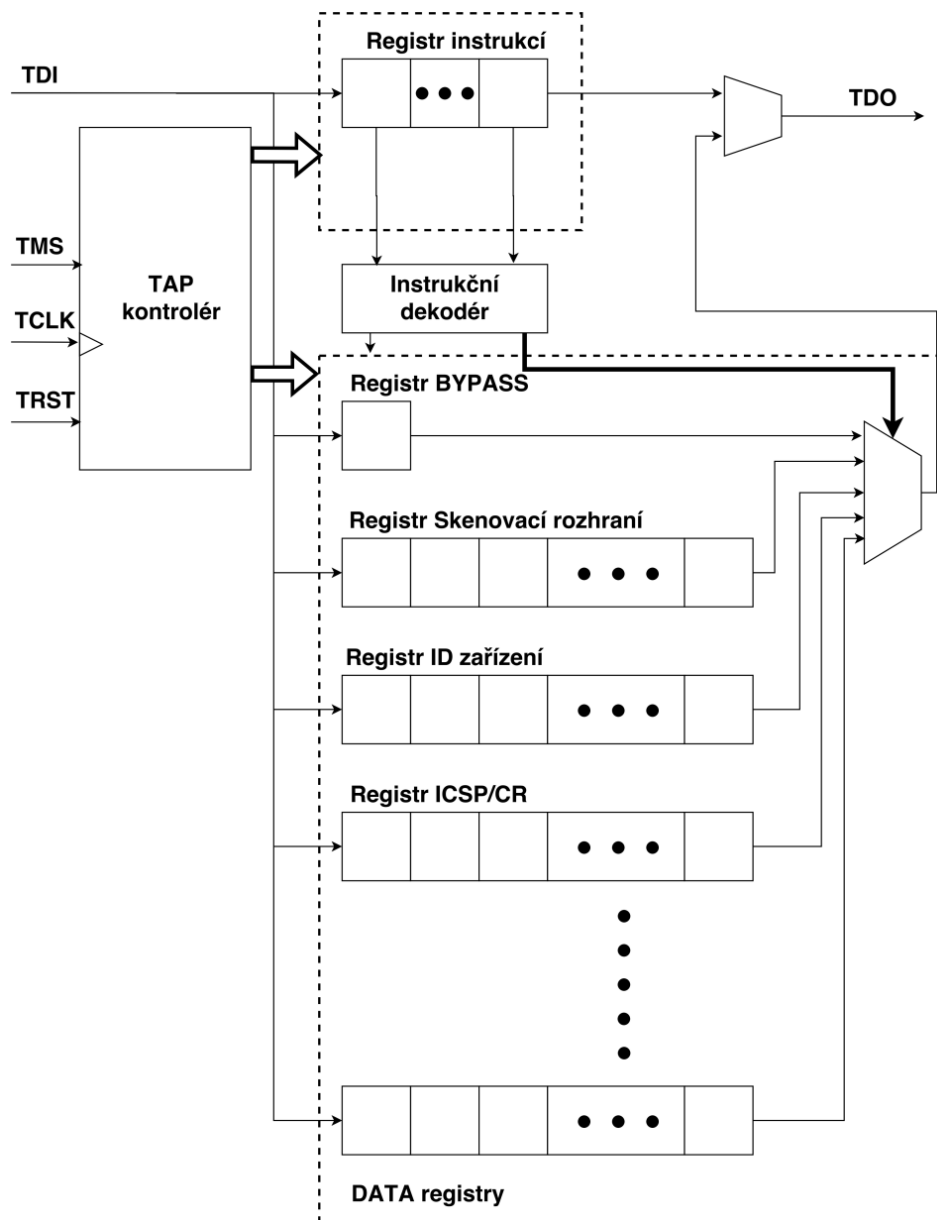
## 2 Standard JTAG

JTAG je rozšířený standard pro testování desek, ladění a programování obvodů. Protože FPGA a CPLD tento standard využívají a poskytují přístup k jejich programovacím funkcím, jsou touto cestou většinou programovány. Z důvodu realizace programátoru přes JTAG se využijí podrobnější informace o tomto standardu.

Standard JTAG (IEEE 1149.1) vznikl za účelem sjednotit testovací postupy a vytvořit univerzální rozhraní pro testování plošných spojů. Standard podporuje test funkčnosti, propojení a test součástek. S postupem času se také začal používat pro programování obvodů a k přístupu k hardwarovému ladícímu rozhraní.

### 2.1 Princip

Rozhraní JTAG je realizováno pomocí posuvného registru mezi jeho piny. Řetězec začíná od TDI a končí v TDO. Rozhraní JTAG může mít více posuvných registrů, které nejsou jen připojeny mezi obvodem a výstupními piny. Nastavením instrukčního registru je možné přistupovat k více posuvným registrům a pracovním režimům. Data jsou posouvány sériově po hodinových cyklech. JTAG obsahuje jeden instrukční registr (IR), několik data registrů (DR) a kontrolér (TAP). V kontroléru TAP je obsažen stavový automat, který řídí celé rozhraní, více podrobností v kapitole 1.3. Jednotlivé datové buňky (registry) jsou připojeny k pinům nebo linkám obvodu. Tyto buňky jsou pak spojeny do řetězce, který vytvoří posuvný datový registr (DR). Podle toho, v jakém stavu je řadič (TAP), mohou být buňky nahrány novými daty nebo mohou stávající data předat. Data jsou pak nasunuta či vysunuta po jednotlivých bitech. Obr. 1 ukazuje popisovaný princip.



Obr. 1: JTAG princip [1]



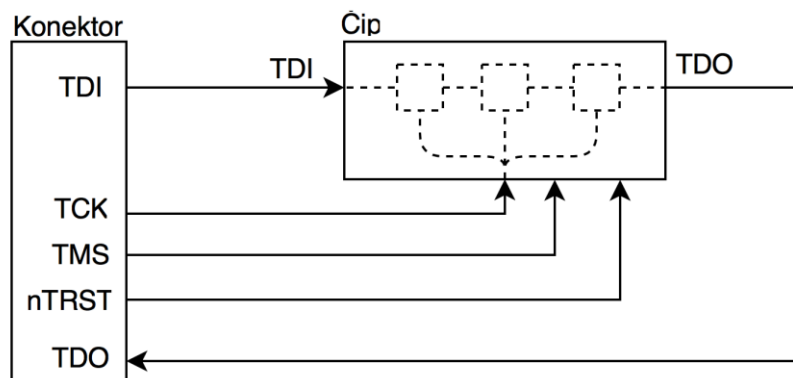
## 2.2 Signály

Tab. 7: JTAG přehled signálů[1]

Pin	Popis	Funkce
TDI	Test Data Input	sériový vstup. Data jsou nasunuta s nástupnou hranou TCK.
TDO	Test Data Output	sériový výstup. Data jsou vysunuta při sestupné hraně TCK.
TCK	Test Clock Input	hodinový signál
TMS	Test Mode Select	vstupní kontrolní signál, který rozhoduje o přechodech stavového TAP kontroléru. Přechody ve stavovém automatu se dějí při nástupné hraně TCK. Proto musí být TMS signál nastaven a ustálen dříve než přijde nástupná hrana TCK.
TRST	Test Reset Input	vstup - asynchronní reset, boundary scan. TRST je optimální podle IEEE Srd. 1149.1. Pin by měl být připojený k zemi, pokud není používán JTAG.

V Tab. 7 je seznam signálů definovaných standardem JTAG a jejich základní popis. Signál TCK je hlavní hodinový signál, může být připojen k několika zařízením s různou pracovní frekvencí. Signál TCK může být zastaven v log. 0 na neomezenou dobu bez jakékoliv změny v testovací logice. Signál může být také zastaven v 1, ale standard negarantuje, že nenastanou žádné změny v testovací logice.

Signál TMS je vzorkován náběžnou hranou signálu TCK a je očekávána změna při sestupné hraně pro TAP kontrolér. Pro normální operaci obvodu musí být TMS v log. 1. Pro nezapojený pin je v logice obvodu zajištěn log 1. Stavový automat je navržen, aby se při TMS v log 1 po pěti hodinových cyklech dostal odkudkoliv ve stavovém automatu zpátky do Test-Logic-Reset.

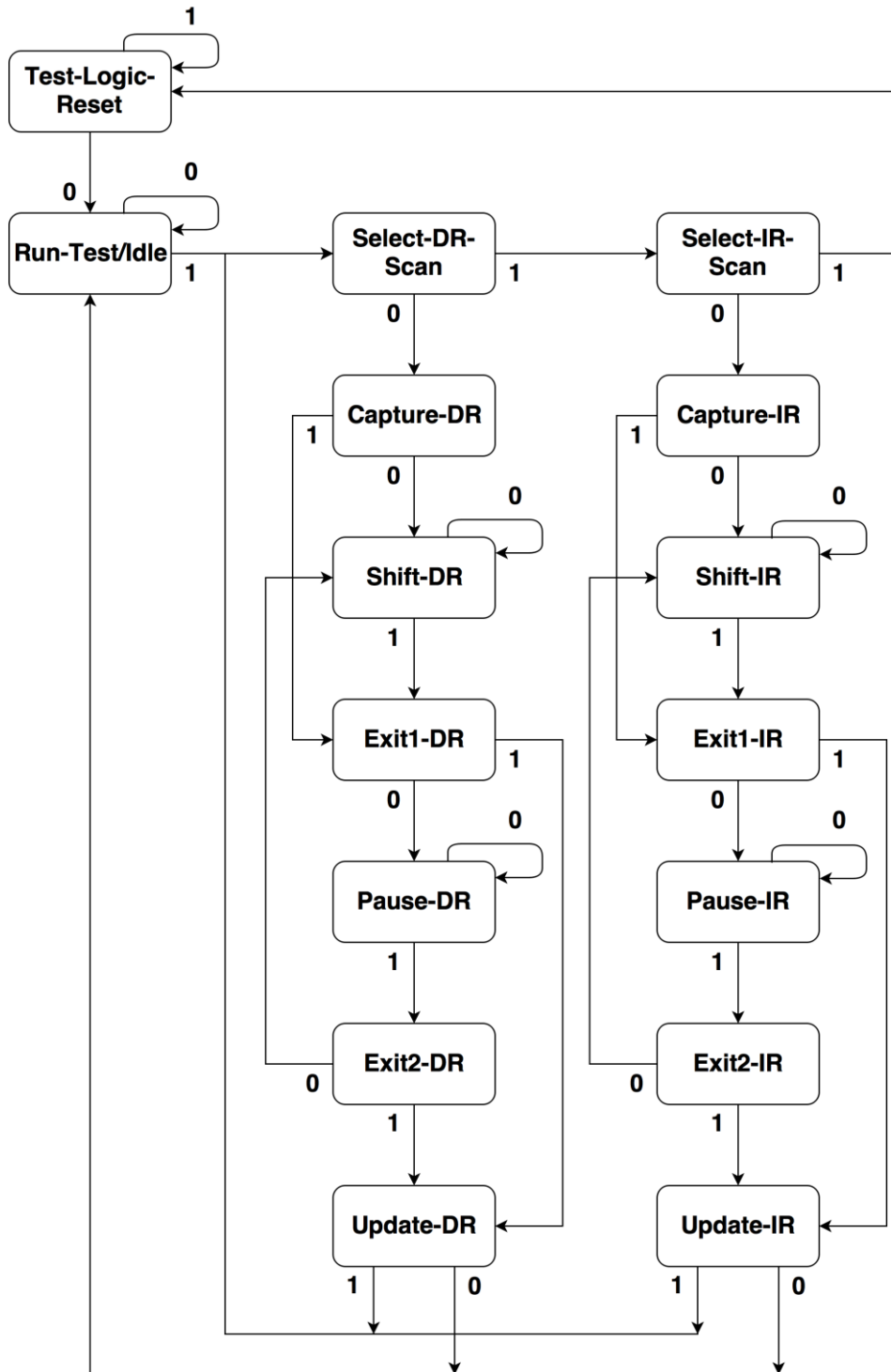


Obr. 2: JTAG signály

Signál TDI je vzorkován náběžnou hranou signálu TCK do obvodu. Při nezapojení tohoto pinu se nastaví signál TDI do log. 1. Data jsou nahrávána sériově přes vstup TDI a přenášejí se do výstupu TDO. Signál TDO se změní při sestupné hraně signálu TCK a měl by být navzorkován s následující náběžnou hranou signálu TCK do převodníku.

### ***2.3 Stavový automat***

Všechny JTAG operace jsou řízeny stavovým automatem implementovaným v TAP kontroléru, který je zobrazen na obr. 3. Stavový automat je řízen TMS a časován na náběžnou hranu TCK. Pro zahájení s testovacími operacemi, je zapotřebí inicializovat stavový automat do stavu Test-Logic-Reset. Toho lze dosáhnout buď asynchronním resetem přes TRST na nulu nebo pěti hodinovými cykly TCK s TMS v log 1. Když je dosaženo stavu Test-Logic-Reset, nastaví se instrukce na IDCODE a pokud není tato instrukce dostupná, tak na BYPASS. Pomocí TMS a 1 hodinovým cyklem se nedostatek do stavu Run-Test/Idle. Podle aktuální vybrané instrukce může být testovací operace vykonána nebo je testovací logika nečinná. Z tohoto stavu se lze dostat do stavu Select-DR/IR-scan, který je pouze stavem přechodným. Z tohoto stavu se přejde do Capture-DR/IR. V tomto stavu mohou být data paralelně nahrána do data registrů vybrané instrukce při nástupné hraně TCK. Pokud data registr vybrané instrukce nemá paralelní vstup, nebo když není potřeba zachytit tyto data pro vybranou instrukci, pak si registry uchovávají svůj původní stav beze změny. Jakmile je Shift-DR/IR dosažen a TMS je v log 0, začne obvod vysouvat bity data registru momentálně zapojeny mezi TDI a TDO na každou sestupnou hranu TCK. Pause-DR/IR může být použit pro pozastavení posouvání dat mezi TDI a TDO. Při sestupné hraně TCK ve stavu Update-DR/IR se nasunutá data v posuvných registrech předají na paralelní výstup. [2]



Obr. 3: JTAG stavový automat

### 3 Návrh programátoru

Navrhovaný programátor by měl identifikovat obvody připojené do JTAG řetězce a získat jejich ID. Dále by měl umět interpretovat soubor SVF pro řízení JTAG řetězce. Tyto informace by měl dále upravit a poslat do převodníku a z něho do programovaného obvodu. Programátor by měl naprogramovat obvod do půl minuty. Pro usnadnění by měl automaticky vygenerovat soubor SVF z poskytnutého souboru od vývojových prostředích Altera, Lattice a Xilinx pro minimalizaci času, který by strávil uživatel při manuálním převodu. Dále by měl umožňovat podpůrné funkce jako je virtuální tlačítko a externí programovací tlačítko.

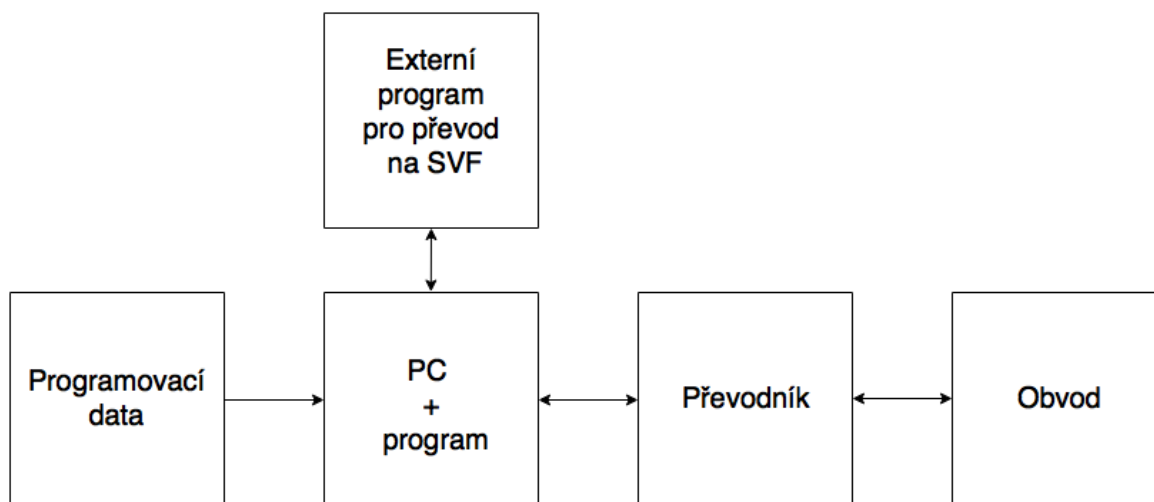
Návrh programátoru byl rozdělen do dílčích částí: blokový návrh, software, hardware a výsledky.

Tab. 8: Požadavky na programátor.

Velikost obvodu:	Malá
Možnost programování FPGA od různých firem:	Ano
Napětí pro převodník a FPGA:	Min 3.3V
Nastavitelná frekvence	Ano
Rychlost naprogramování:	Do 30 sekund
Přenos dat mezi počítačem a FPGA:	Dodatečně
Propojení deska-počítač	USB
Cena:	Nízká

#### 3.1 Blokový návrh

Je výhodné blokově rozdělit programátor na jednotlivé části viz Obr. 4.



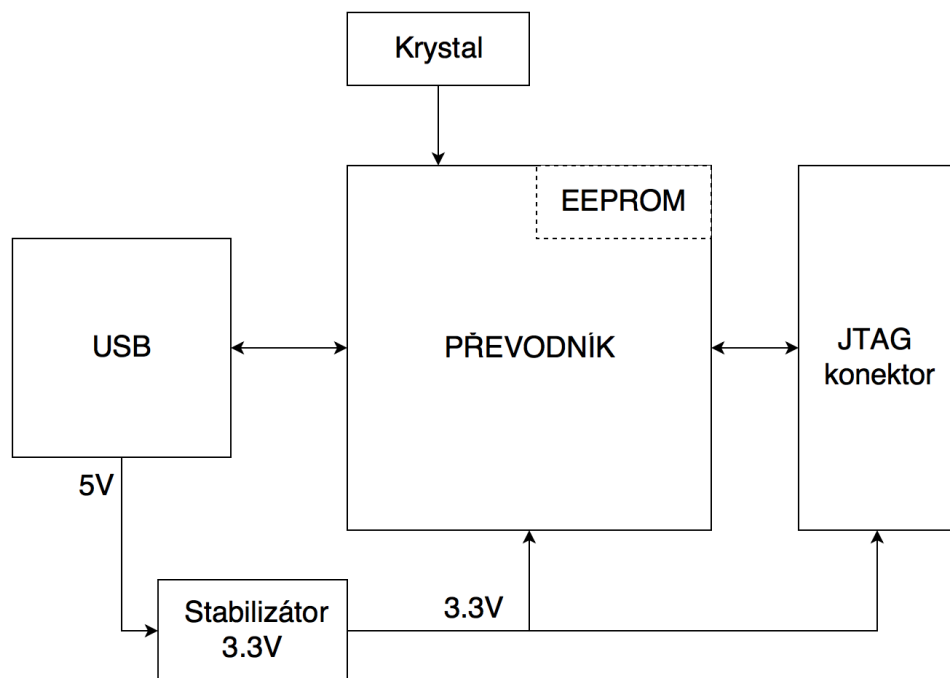
Obr. 4: Konceptní schéma programátoru

## PC + Program

Program je nejdůležitější součástí programátoru. Slouží ke komunikaci s převodníkem, ke čtení programovacích dat a následné úpravě pro převodník. Pro data vytvořená vývojovým prostředím je nutné data konvertovat externími programy od výrobců daných čipů. Vytváření samostatného konvertoru by bylo nad rámec této práce, zejména kvůli množství různorodých obvodů a způsoby, jak jsou programovány.

## Převodník USB – JTAG

Převodník má za úkol vysílat a přijímat data skrz USB a těmito daty pak řídit jednotlivé piny připojené k FPGA. Převodník k činnosti potřebuje podpurné součástky, jakými je krystal a stabilizátor napětí viz Obr. 5. Některé převodníky mají tyto součásti už implementované jako FT245R. Pro bezpečnost se přidává i oddělování napájení mezi převodníkem a programátorem, ale tato možnost nebyla využita.



Obr. 5: Blokový návrh programátoru

## ***3.2 Realizace programátoru***

### ***3.2.1 Software***

#### **Řídící program - koncepce**

Základním blokem vytvořeného programátoru je program, který umí přečíst soubor SVF a převést je na datový tok vhodný pro poslání přes USB do převodníku. Jako zdroj dat může být použit přímo soubor SVF nebo soubor vygenerovaný ve vývojovém prostředí pro FPGA. V případě druhé možnosti se program pokusí převést data na SVF automaticky, jinak by jej uživatel musel převést manuálně. Je nutné v programu zadat, od jakého výrobce je daný soubor, aby mohl určit, jakému konvertoru příkaz poslat. Program pošle příkaz do převáděcího programu daného výrobce a vytvoří soubor make.svf. Je nutné mít nainstalované dané vývojové prostředí tohoto souboru, jinak se obvod nepřevéde a operace selže. Pokud je programovaný obvod v JTAG řetězci jediný, může být ihned přečten a poslán převodníku. Pokud se nachází další obvody před a za obvodem, musí se upravit instrukce TIR, TDI, HDR, HIT v make.svf, podle toho, jak dlouhé IR registry mají okolní obvody. Poté se může programovat.

Program také nabízí konfiguraci JTAG pinů na dostupných pinech. Rychlost baud rate převodníku se také dá nastavit s nejvyšší rychlostí 5 MBad/s

Program je napsán v C++ ve vývojovém prostředí QT Creator, které nabízí jednoduchý návrh aplikací s grafickým rozhraním a pomocnými třídami. QT aplikace lze používat ve více platformách.

#### **Přehrávač SVF**

Pro čtení SVF souborů bylo využito Lib(X)SVF dostupné na [www.clifford.at/libxsvf/](http://www.clifford.at/libxsvf/). Je to knihovna pro přehrání SVF souboru napsaná v C. Umožňuje přehrávání jak SVF, tak XSVF, to však nebylo využito. Tato knihovna byla v programu implementována do třídy JTAG.

#### **Externí programy pro konverzi**

Jsou využity tři programy, impact od Xilinx, ddtcmd od Lattice a quartus\_cpf od Altera. Všechny tři programy pracují v příkazovém řádku. Tyto programy zajišťují konverzi mezi soubory, které vygenerují jednotlivé vývojové prostředí do SVF formátu, který náš programátor dokáže přehrát. Každý program má jinou sadu příkazů, kterou programátor zavolá při startu programování. Příkaz obsahuje vstupní soubor, cestu pro výstupní soubor a funkci, kterou má výsledný SVF vykonat a další informace. Při převodu se vypíše příkaz do textového pole.

## Komunikace s převodníkem v bit-bang režimu

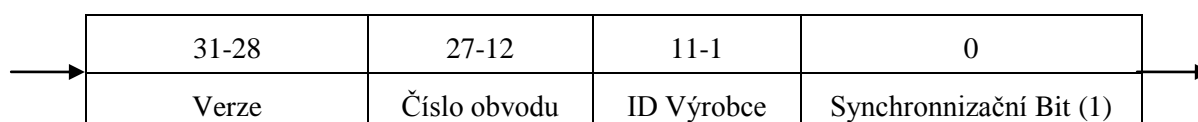
Převodník je pro účely programování přepnut do módu synchronního bit-bangu. Převodník v tomto režimu musí dostávat data na všechny své piny pro každou změnu. V programu se proto předem vytvoří datový řetězec, který je naplněn příkazem ze SVF souboru a postupně jej odešle na převodník. Kdyby byla data vysílána po bitech, tak by maximální rychlost byla 1000bd/s neboli 1bit na 1ms rámeček v USB.

Převodník v synchronním režimu napřed uloží stav na svých pinech do zásobníku a pak načte nová data ze zásobníku zaslaného počítačem. Uložená data jsou vždy o jeden krok za výstupem. Jestliže převodník nemá nová data, zůstávají hodnoty na pinech stejné.

## Rozeznání JTAG řetězce

Podle standardu IEEE 1449.1 je dobrovolné přidat do obvodu svůj ID kód, který nese informaci o firmě, typu obvodu a jeho verzi, znázorněno na Tab. 9. V dnešní době tento přídavek má většina obvodů a dají se tak lehce identifikovat obvody v řetězci.

Tab. 9: Rozdělení 32 bitového ID



Jako první je synchronizační bit s hodnotou 1. Pokud jej programátor zachytí, očekává, že obvod je v režimu IDCODE a čeká 32 bitů. Pokud narazí na obvod začínající nulou, znamená to, že obvod nemá ID kód a je tedy v BYPASS režimu se šířkou registru jedna a hodnotou 0.

Následuje ID výrobce, které je v upravené verzi ze standardu pro identifikaci výrobců JEP106AR. Například ARM Ltd. Má ID (7F 7F 7F 7F 3B) podle JEP106AR. Každé (7F) značí, že je výrobce posouván o jednu stranu dál, následuje samostatné ID(3B), osmý bit ID je bit paritním. Do registru JTAG ID by se takto dlouhý název nevešel a proto se spočítá na které straně je daný výrobce, struktura viz Tab. 10. Výsledné ID obsahuje stranu a ID bez parity. ARM Ltd. (23B).

Tab. 10: Struktura ID výrobce

ID Výrobce (11 Bitů)	
11-8 (4 Bity)	7-1 (7 Bitů)
Strana	ID výrobce bez parity

Poté následuje číslo obvodu specifické pro danou skupinu obvodů. Dá se zjistit, z jaké rodiny obvodů daný obvod je, ale informace, jakými je druh pouzdra a počet vývodů může

být stejný i pro více než deset různých obvodů. Tyto informace jsou většinou zapsány v dokumentaci od výrobců obvodů. Pro účely programátoru by takto postupně získávané informace byly časově náročné, a proto bylo zvoleno tyto informace získávat strojově ze souborů BSDL, které popisují daný obvod pro JTAG. Tyto soubory jsou dostupné buď ze stránek výrobců, nebo ve složkách jejich programů. Také existuje stránka bsd1.info na které je možno stáhnout potřebné modely. Ze skupin BSDL souborů byla pak strojově vyseparována ID, IR délka a název obvodu. Tyto informace byly pak zapsány do textového souboru. Viz tabulka dole.

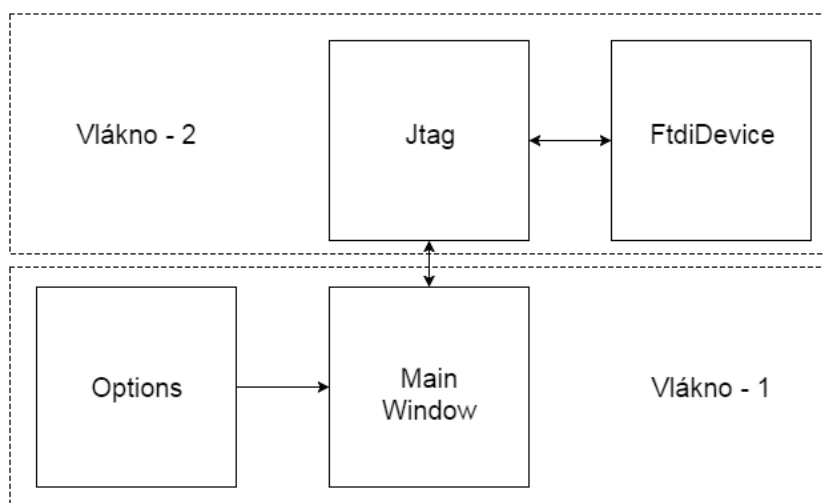
Poslední 4 bity značí variaci obvodu.

Tab. 11: Seznam obvodů s ID a IR délkou

ID(Hex)	ID(Bit)	Název	IR
6d88093	XXXX0110110110001XXX000010010011	xa2c128.bsd	8
6d8b093	XXXX0110110110001011000010010011	xa2c128_cp132.bsd	8
6d8e093	XXXX0110110110001110000010010011	xa2c128_ft256.bsd	8
6d8c093	XXXX0110110110001100000010010011	xa2c128_tq144.bsd	8
6d8a093	XXXX0110110110001010000010010011	xa2c128_vq100.bsd	8

### Řídící program – realizace

Program je realizován jak na Obr. 6.

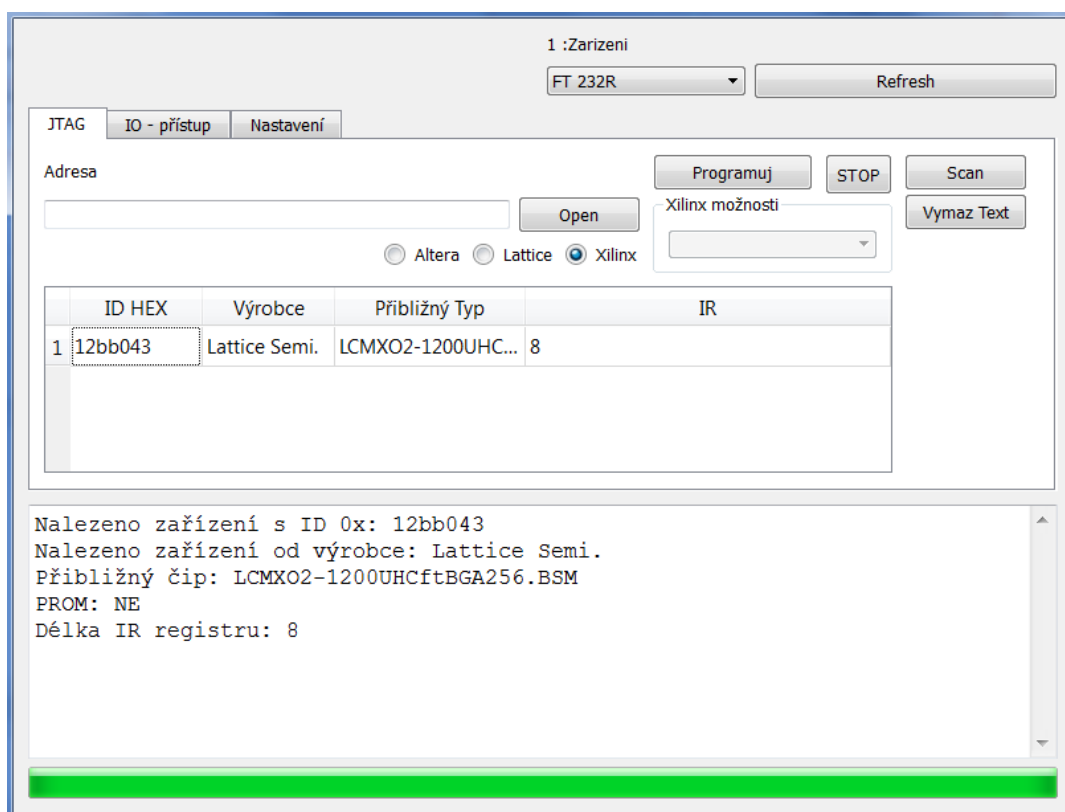


Obr. 6: Blokové reprezentace programu třídami – JTAG, FTDI, MW, Options

Centrální blok MainWindow zajišťuje konfiguraci grafického prostředí pro řízení uživatelem. Dále zaznamenává obvody v řetězci do tabulky a rozhoduje, jak se bude daný

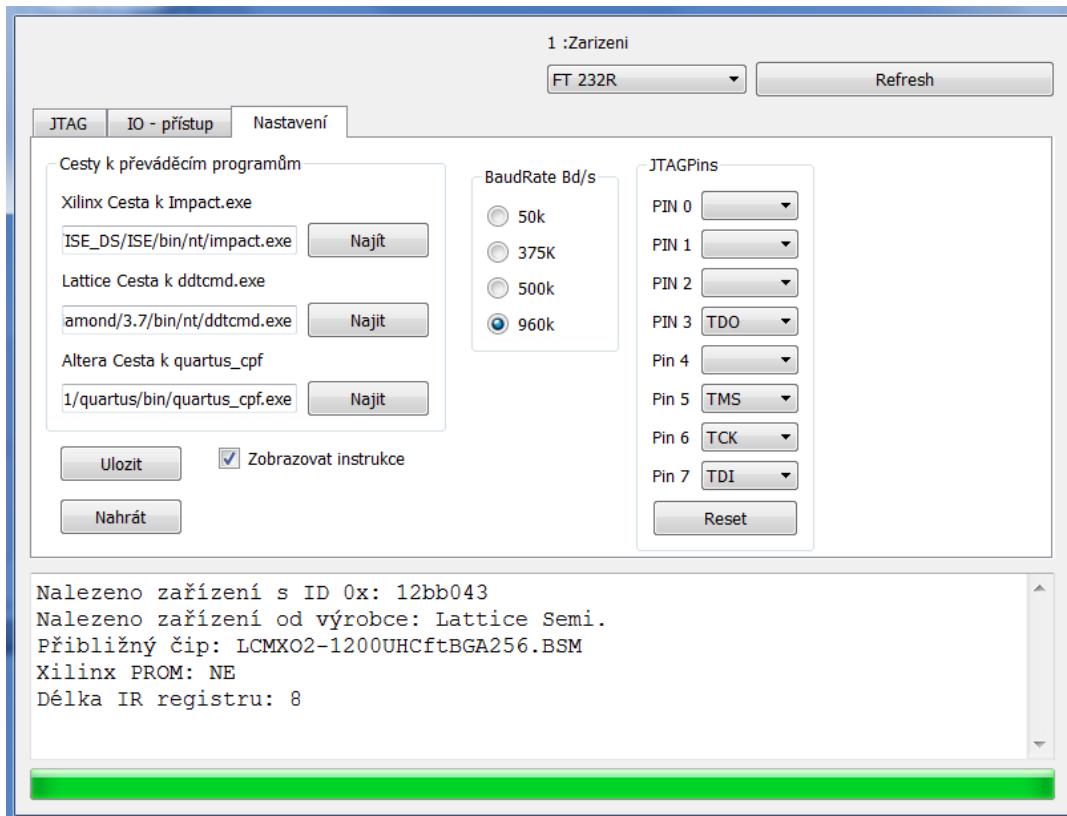


obvod převádět a programovat. Programovací rozhraní obsahuje typické prvky pro programátory. Prostředí by mělo být pro znalého uživatele FPGA intuitivním viz Obr. 7.



Obr. 7: Hlavní okno programátoru (MainWindow)

Blok Options konfiguruje rozvržení okna „Nastavení“, jeho tlačítka a textové pole a interakci mezi nimi viz Obr. 8. Dále se stará o načítání a ukládání dat, jako jsou cesty k externím programům a konfigurace pinů. Tyto data jsou uložena ve složce s programem pod názvem settings.ini. Tato třída poskytuje funkce pro získávání cest k externím programům a momentálnímu nastavení vývodů, které využívá MainWindow.



Obr. 8: Okno nastavení programátoru (Options)

JTAG blok implementuje přehrávač SVF a nalézání obvodů v JTAG řetězci. Zřetězuje jednotlivé bity instrukce do jediné zprávy pro FtdiDevice. JTAG blok běží ve vlastním vlákně, kvůli možnosti průběžného monitorování a ovládání programu.

FtdiDevice zajišťuje komunikaci s převodníkem. Zde se nastavuje komunikační rychlost v baudech, vybírá zvolený převodník a řeší se přenos dat mezi počítačem a převodníkem.

### 3.2.1 Hardware

#### Vybraný převodník

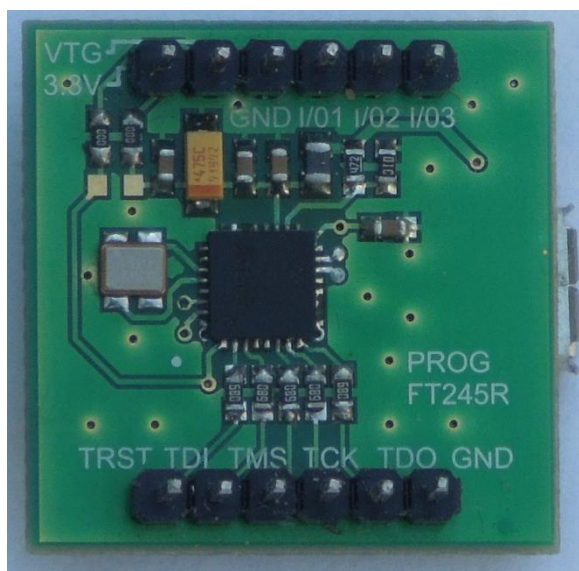
Na základě rozboru programátorů byly zvoleny dva převodníky od firmy FTDI, FT2232H a FT245R. Tyto převodníky byly zvoleny oproti časové náročnosti mikrokontrolérů a pro svou snadnou implementovatelnost. Tyto převodníky budou pracovat v režimu synchronií Bit-Bang. Bit-Bang byl zvolen kvůli plné kontrole 8 pinů a dále díky možnosti vyměnit použité převodníky za jiné s podporou režimu Bit-Bangu od FTDI, oproti režimu MPSSE, který je na JTAG připraven, ale je podporován pouze několika obvody od FTDI, jako je třeba FT2232H.

Převodník FT245R bude pomalejší kvůli velikosti datových zásobníků. Jeho zásobník má 256B oproti 4kB u FT2232H. S menším zásobníkem nastane častější komunikace přes USB, a tím i pomalejší přenos.

Výsledné desky byly vytvořeny co nejmenší, deska s FT245 vyšla na 25x25mm a deska S FT2232H na 25x50. Desky jsou opatřeny piny, které bude možné zasunout do budoucích desek s FPGA pro jednoduchou implementaci programátoru.

### Deska s FT245R

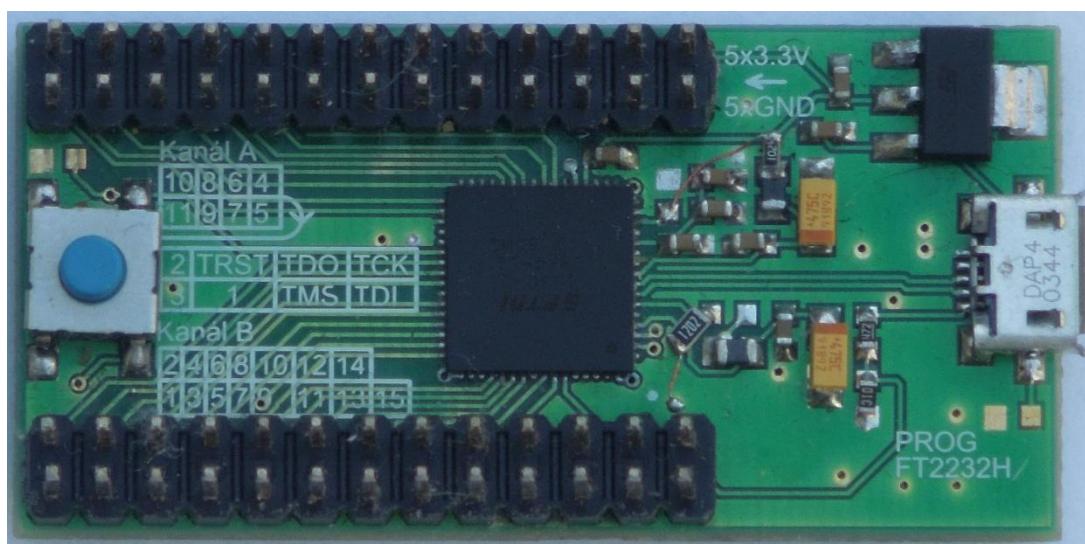
Tato deska má osm signálových pinů, které jsou vyvedeny ven. Tyto piny mohou být nastaveny na jakýkoliv JTAG pin anebo mohou být ovládány přímo uživatelem z aplikace. Schéma v příloze PROG FT245T. Deska se dá napájet dvěma způsoby vhodným osazením součástek. Při napájení z USB pěti volty je nutné odstranit komponenty R10, R11 a R2. Při této konfiguraci je možné odebírat 5V z J1-1 a 3.3V z J1-2. Pokud nejsou napětí na pinech potřeba, dají se přerušit odstraněním R1 a R3. U této konfiguraci není nutný externí krystal. Při napájení desky z externího zdroje 5 až 3.3V je nutné odstranit součástky L1, R4 a R2. Napájení 3.3 až 5 volt je aplikováno na pin J1-1 a na pin J1-2 se přivede napětí požadované pro signálové piny tj. od +1.8V do +5.25V. Při napájení tohoto obvodu externími 3.3V je nutné zajistit osazení externího oscilátoru. Výsledná deska na Obr. 9.



Obr. 9: Hotová deska Prog FT245R

## Deska s FT2232H

Tato deska má 16 signálových pinů, které jsou dostupné po 8 ve dvou kanálech, ke kterým je přístup přes USB. Piny jsou v programu nastavitelné stejně jak s FT245R. Schéma v příloze PROG FT2232H. Desku je možné napájet přes USB 5V nebo externě 3.3V. Při externím napájení nesmí být přítomný napěťový regulátor IO1. Výsledná deska na Obr. 10.



Obr. 10: Hotová deska Prog FT2232H

### 3.3 Výsledky

Dokázaly se realizovat dva programátory.

Deska s FT245R je při programování 3x pomalejší než FT2232H kvůli menšímu datovému zásobníku a má pouze jeden kanál, ale nepotřebuje napěťový stabilizátor a je menší.

Deska s FT2232H je rychlejší a má dva kanály. Nevýhodou je velikost obvodu a nutný krystal se stabilizátorem. Dvouřadé připojovací piny nejsou zcela využity a mohlo se tak ušetřit místo. Programátor se dokáže v některých případech vyrovnat komerčním protějškům.

Pro zabudovaný programátor je FT245R celkem vhodný, jenom je pomalejší a oproti FT2232H je zase příliš velký. Možný kompromis by byl obvod FT232H, který má větší datový zásobník 1kB a jeden kanál, zato má rozměry 8x8mm oproti 5x5mm FT245R. Program v současném stavu by takový obvod měl detekovat a uměl ho použít.

Program dokáže rozpoznat obvod na JTAG řetězci a přiřadit mu výrobce a pokud je od Altera, Xilinx nebo Lattice, dokáže i přiřadit přibližný název. Bylo dosaženo automatického konvertování externími programy. Při běžném programování by nabízené možnosti, jak konvertovat, měly být dostačující, ale jsou situace a módy, které automatická konverze

nezvládne a musí se tak vygenerovat soubor SVF nezávisle uživatelem. Při přípravě dat pro odeslání na převodník by se dalo vylepšit přecházení mezi datovými instrukcemi a zrychlit tak programování. Tento nedostatek se projevuje hlavně při velkém množství instrukcí v SVF souboru po malých datových paketech.

## 4 Závěr

V rámci bakalářské práce byl proveden rozbor existujících programátorů a následná realizace. Při rozboru programátorů byla pozornost zaměřena na jeho velikost, cenu a podporu programovatelných obvodů. Řídící obvod v těchto zařízeních byl většinou převodník nebo mikrokontrolér.

Programátory od výrobců obvodů FPGA byly fixovány na své obvody a vývojové prostředí. Takové programátory jsou většinou drahé a nepodporují obvody od ostatních výrobců FPGA. Programátory třetích stran podporovali pouze vybrané spektrum obvodů a byly kompatibilní s pouze některým programovacím softwarem nebo mají svůj vlastní, ale s další podporou nelze vždy počítat. Cena vychází níž než u oficiálních programátorů.

Z rozboru byly vybrány dva převodníky pro realizaci a to FT2232H a FT245R. Tyto převodníky byly zvoleny oproti časové náročnosti mikrokontrolérů a pro svou snadnou implementovatelnost.

Před realizací byly zvoleny vlastnosti, jakými programátor bude disponovat a vytvořilo se blokové schéma. Na základě schématu byly vytvořeny dvě programovací desky PROG FT245R s velikostí 5x5mm a PROG FR2232H s velikostí 5x10mm. Vytvořil se programovací software, který dokáže rozpoznat obvody v JTAG řetězci a také je naprogramovat. Nejvíce časově náročné z celé práce byl vývoj programu. Přidané funkce jako přepínače a komunikace přes FPGA se nepodařily včas realizovat. Nevyužité piny lze ale přes program manuálně ovládat. Programátor se dokázal vyrovnat s některými existujícími programátory v rychlosti přenosu a někdy je i přesáhl. Programátor v současném stavu lze dále vylepšovat, například přípravou dat pro převodník pro zrychlení programování a přidání funkčnosti.

# Literatura

- [1] ALTERA. *IEEE 1149.1 JTAG Boundary-Scan Testing: in Altera Devices* [online]. 2005 [cit. 2015-12-12]. Dostupné z: [https://www.altera.com/content/dam/altera-www/global/en\\_US/pdfs/literature/an/an039.pdf](https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/an/an039.pdf)
- [2] IEEE 1149.1-2001. *IEEE STANDARD ASSOCIATION. IEEE Standard Test Access Port and Boundary-Scan Architecture* [online]. 2008 [cit. 2015-12-12]. Dostupné z: <http://standards.ieee.org/findstds/standard/1149.1-2001.html>
- [3] XILINX. *Platform Cable USB II* [online]. 2015 [cit. 2015-12-12]. Dostupné z: [http://www.xilinx.com/support/documentation/data\\_sheets/ds593.pdf](http://www.xilinx.com/support/documentation/data_sheets/ds593.pdf)
- [4] Bus Blaster [online]. [cit. 2015-12-12]. Dostupné z: [http://dangerousprototypes.com/docs/Bus\\_Blaster](http://dangerousprototypes.com/docs/Bus_Blaster)
- [5] ALTERA. *USB-Blaster Download Cable User Guide* [online]. 2007 [cit. 2015-12-12]. Dostupné z: [https://www.altera.com/content/dam/altera-www/global/en\\_US/pdfs/literature/ug/ug\\_usb\\_blstr.pdf](https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/ug/ug_usb_blstr.pdf)
- [6] DIGILENT . *Cmod S6* [online]. 2014 [cit. 2015-12-12]. Dostupné z: <http://store.digilentinc.com/cmod-breadboardable-spartan-6-fpga-module/>
- [7] FTDI. *FT2232H Dual High Speed USB to Multipurpose UART/FIFO IC* [online]. 2012 [cit. 2015-12-12]. Dostupné z: [http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS\\_FT2232H.pdf](http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT2232H.pdf)
- [8] FTDI. *FT245BL USB FIFO ( USB - Parallel ) I.C.* [online]. 2005 [cit. 2015-12-12]. Dostupné z: [http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS\\_FT245BL.pdf](http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT245BL.pdf)
- [9] FTDI. *FT245R USB FIFO IC* [online]. 2015 [cit. 2015-12-12]. Dostupné z: [http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS\\_FT245R.pdf](http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT245R.pdf)
- [10] ATMEL. *ATmega16U2* [online]. 2012 [cit. 2015-12-12]. Dostupné z: <http://www.atmel.com/devices/ATMEGA16U2.aspx>
- [11] Cypress. *EZ-USB ® FX2LP™ USB Microcontroller High-Speed USB Peripheral Controller: CY7C68013A, CY7C68014A, CY7C68015A, CY7C68016A* [online]. 2015 [cit. 2015-12-16]. Dostupné z: <http://www.cypress.com/file/138911/download>
- [12] DIGILENT . *ATLYS* [online]. 2014 [cit. 2015-12-12]. Dostupné z: <http://store.digilentinc.com/atlys-spartan-6-fpga-trainer-board-limited-time-see-nexys-video>
- [13] LATTICESEMI . *HW-USB-N-2B* [online]. 2014 [cit. 2016-5-12]. Dostupné z: <http://www.latticesemi.com/Products/DevelopmentBoardsAndKits/ProgrammingCablesforPCs.aspx>
- [14] MICROSEMI . *SILICON SCULPTOR 3* [online]. 2014 [cit. 2016-5-12]. Dostupné z: <http://www.microsemi.com/products/fpga-soc/design-resources/programming/silicon-sculptor-3#documentation>

## Seznam symbolů, veličin a zkratek

ARM	Advanced RISC Machine
AS	Aktive Serial Programming
CPLD	Complex Programmable Logic Device
DR	Data Registr
DSP	Digital Signal Processor
EEPROM	Electrically Erasable Programmable Read-Only Memory
FIFO	First In, First Out
FPGA	Field-Programmable Gate Array
I2C	Inter-Integrated Circuit
IR	Instruction registr
JTAG	Joint Test Action Group
LUFA	Lightweight USB Framework for AVR's
PID	Product ID
PROM	Programmable Read Only Memory
PS	Passive Serial Programming
SPI	Serial Peripheral Interface
SVF	Serial Vector Format
TAP	Test Access Port
TCK	Test Clock Input
TDI	Test Data Input
TDO	Test Data Output
TMS	Test Mode Select
TRST	Test Reset Input
UART	Universal Asynchronous Receiver and Transmitter
USB	Universal Serial Bus
VID	Vendor ID
SVF	Serial Vector Format
TIR	Trailer Instruction Register



TDI	Trailer Data Register
HDR	Header Data Register
HIT	Header Instruction Register
BSDL	Boundary Scan Description Language

## Seznam obrázků

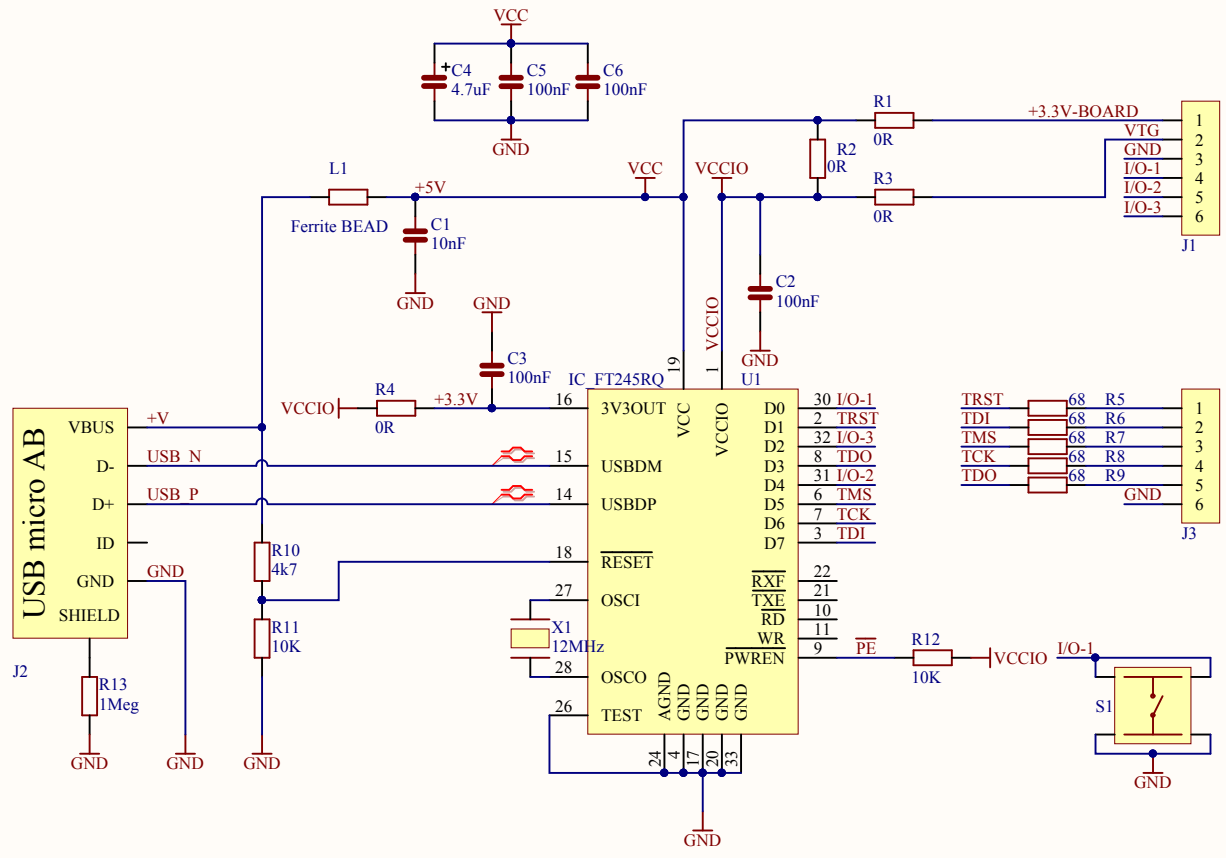
OBR. 1:	JTAG PRINCIP [1] .....	16
OBR. 2:	JTAG SIGNÁLY .....	17
OBR. 3:	JTAG STAVOVÝ AUTOMAT .....	19
OBR. 4:	KONCEPČNÍ SCHÉMA PROGRAMÁTORU .....	20
OBR. 5:	BLOKOVÝ NÁVRH PROGRAMÁTORU .....	21
OBR. 6:	BLOKOVÉ REPREZENTACE PROGRAMU TŘÍDAMI – JTAG, FTDI, MW, OPTIONS.....	24
OBR. 7:	Hlavní okno programátoru (MAINWINDOW) .....	25
OBR. 8:	Okno nastavení programátoru (OPTIONS).....	26
OBR. 9:	HOTOVÁ DESKA PROG FT245R .....	27
OBR. 10:	HOTOVÁ DESKA PROG FT2232H.....	28

## Seznam tabulek

TAB. 1:	SOUHRN VLASTNOSTÍ EXTERNÍCH PROGRAMÁTORŮ .....	10
TAB. 2:	VÝVOJOVÁ DESKA CMOD S6 .....	10
TAB. 3:	VÝVOJOVÁ DESKA ATLYS .....	11
TAB. 4:	NASTAVITELNÉ MÓDY PŘEVODNÍKU .....	11
TAB. 5:	POUZDRA A CENY OBVODŮ .....	13
TAB. 6:	VLASTNOSTI OBVODŮ .....	13
TAB. 7:	JTAG PŘEHLED SIGNÁLŮ[1] .....	17
TAB. 8:	POŽADAVKY NA PROGRAMÁTOR .....	20
TAB. 9:	ROZDĚLENÍ 32 BITOVÉHO ID .....	23
TAB. 10:	STRUKTURA ID VÝROBCE .....	23
TAB. 11:	SEZNAM OBVODŮ S ID A IR DÉLKOU .....	24

# Přílohy

<b>A</b>	<b>PROG FT245R</b>	<b>37</b>
A.1	Obvodové zapojení .....	37
A.2	Deska plošného spoje – top .....	38
A.3	Deska plošného spoje – bottom .....	39
A.4	Seznam součástek .....	40
<b>B</b>	<b>PROG FT2232H</b>	<b>41</b>
B.1	Obvodové zapojení .....	41
B.2	Deska plošného spoje – top .....	42
B.3	Deska plošného spoje – bottom .....	43
B.4	Seznam součástek .....	44



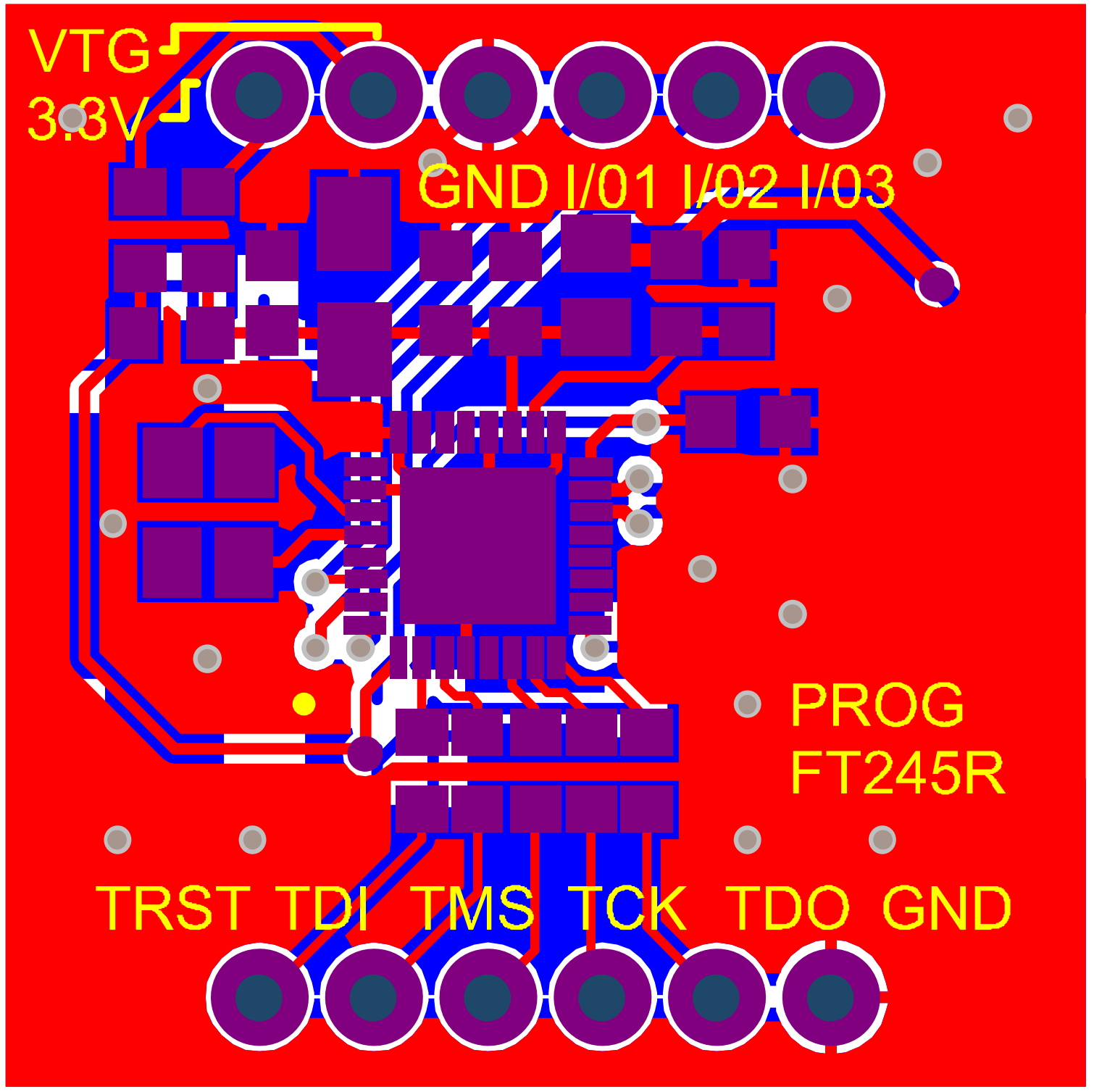
Title		
Size	Number	Revision
A4		
Date:	29.5.2016	Sheet of
File:	C:\Users\...\Sheet1.SchDoc	Drawn By:

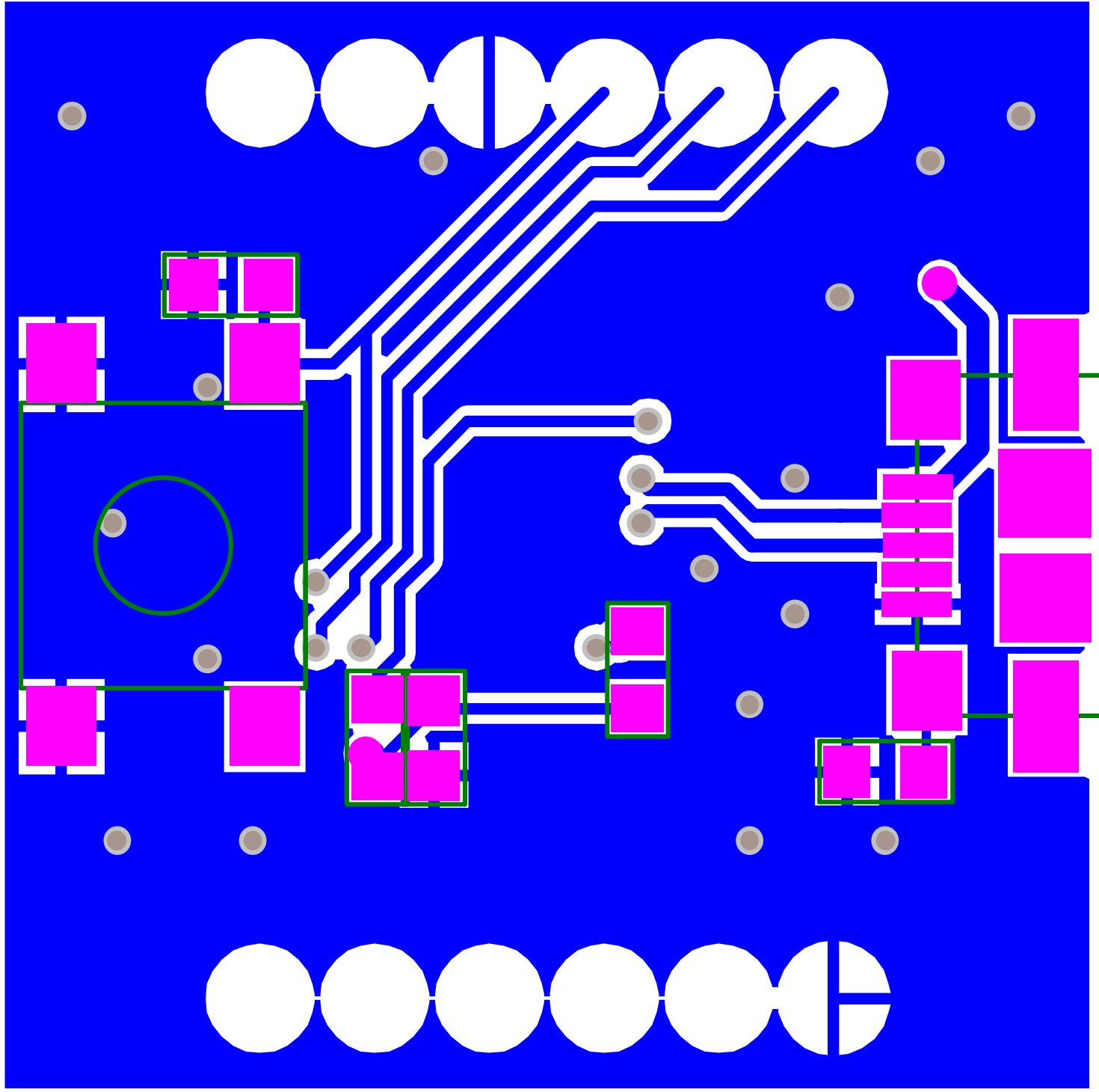
VTG  
3.3V

GND I/O1 I/O2 I/O3

PROG  
FT245R

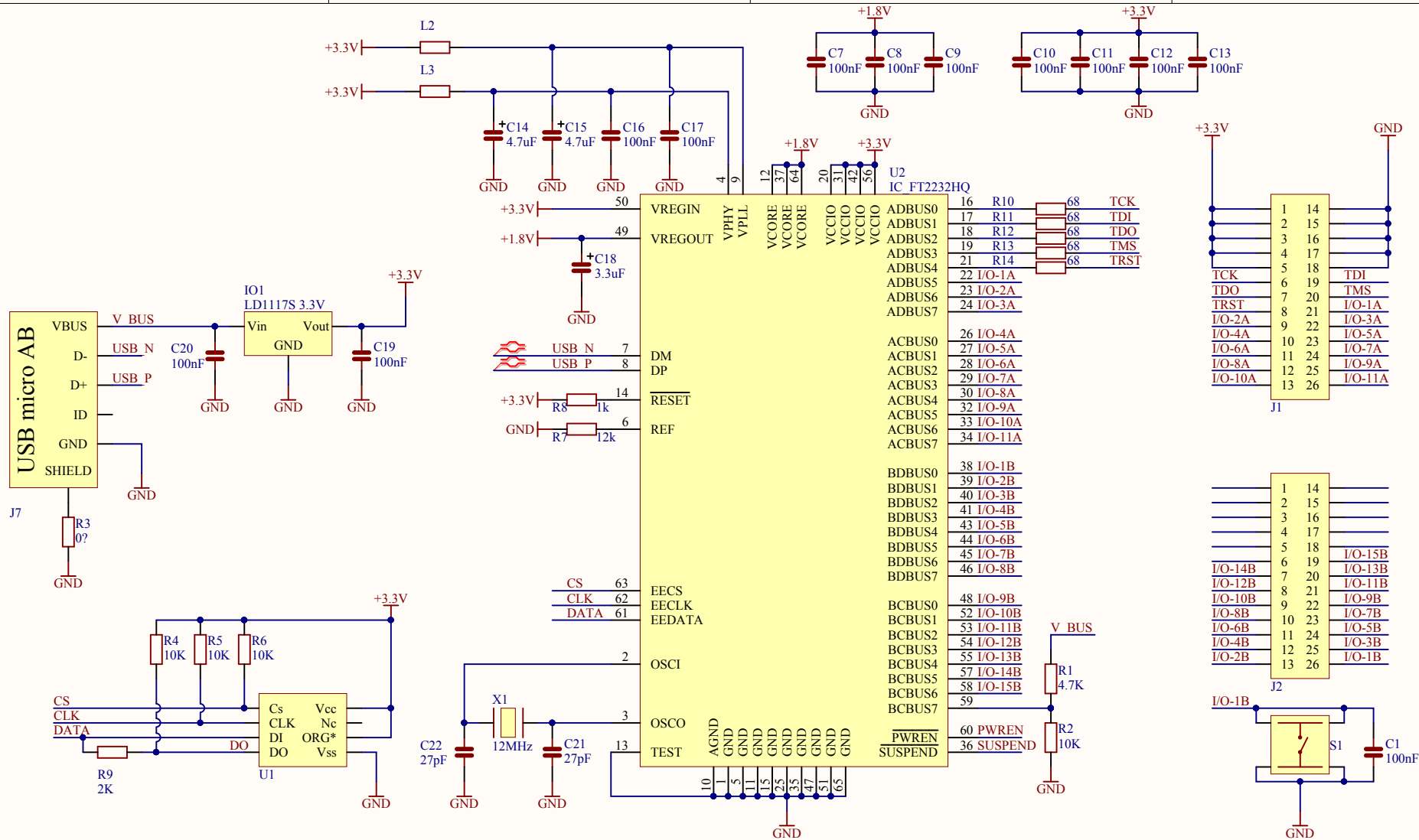
TRST TDI TMS TCK TDO GND



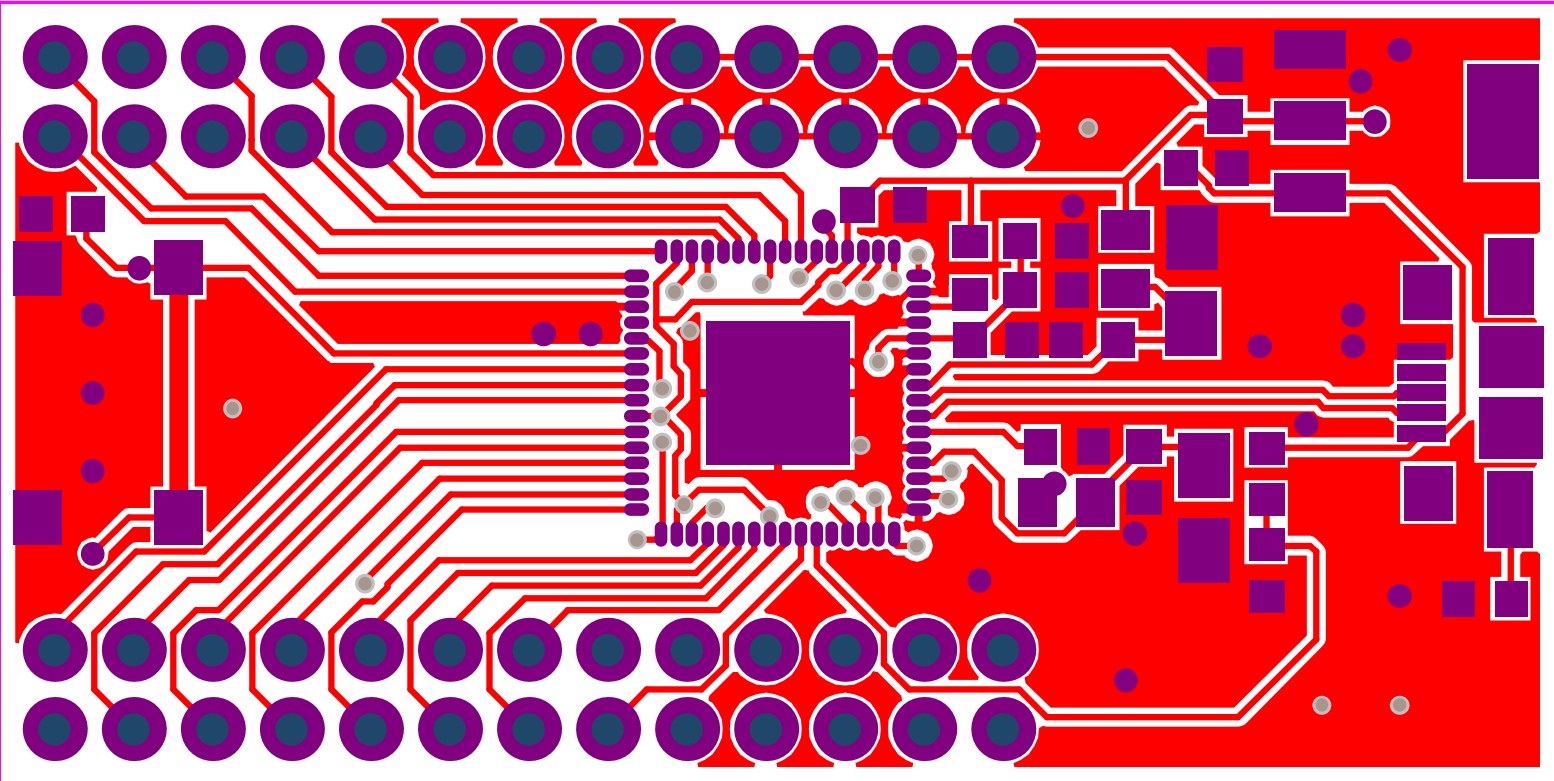


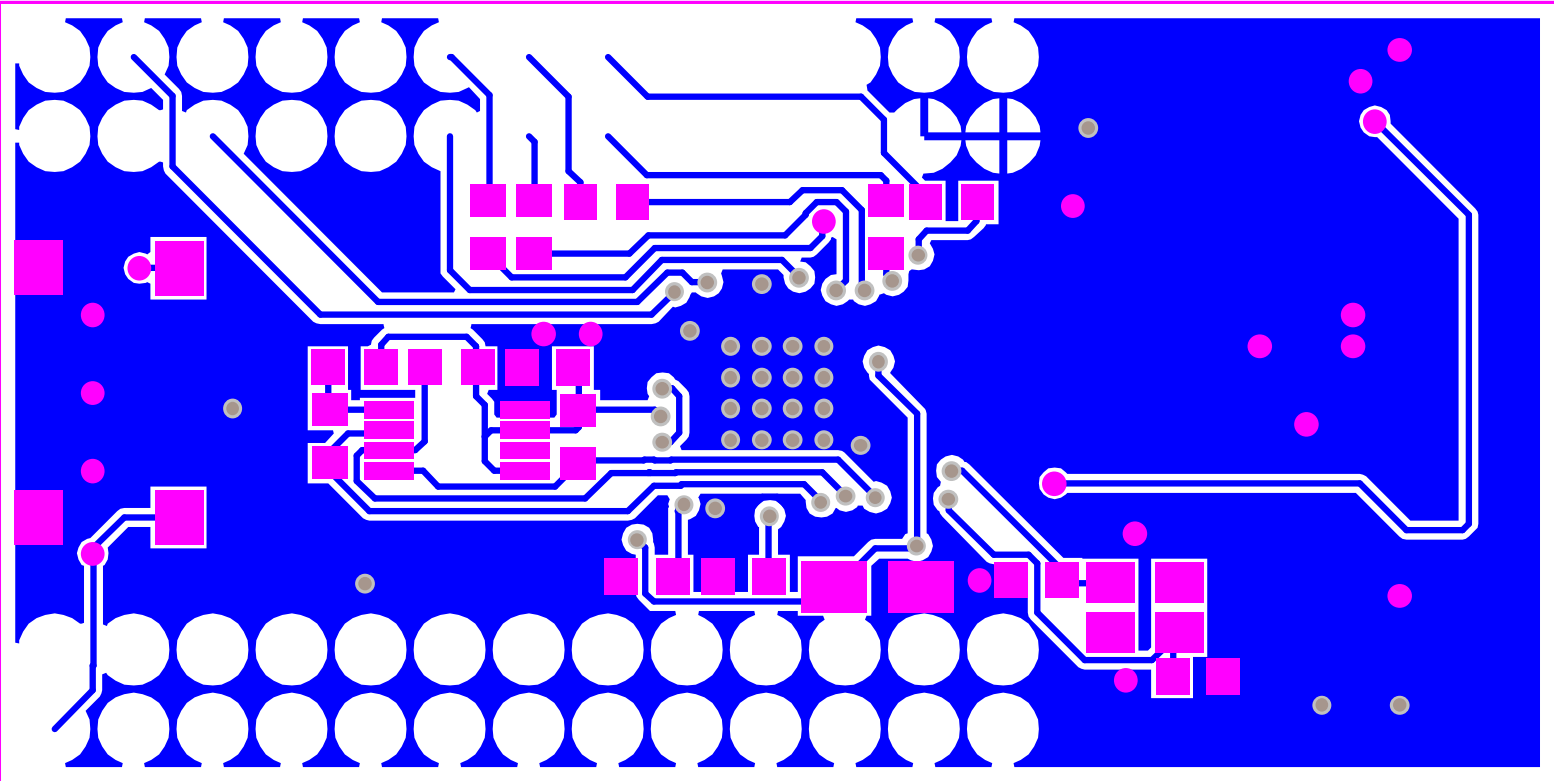
Comment	ComponentLink1 URL	Description	Designator	Footprint	LibRef	Quantity
10nF		CAPACITOR CHIP	C1	CAPC0603N	CAPC	1
100nF		CAPACITOR CHIP	C2, C3, C5, C6	CAPC0603N	CAPC	4
4.7uF		CAPACITOR POLARIZED TANTAL	C4	SMA/3216 - 18	CAPC-POL- TANTAL	1
6_HEADER			J1, J3	6-PIN-HEADER	6_HEADER	2
USB_Micro_Type_ AB		TE CONNECTIVITY / AMP 1981584-1 MICRO USB, 2.0	J2	MICRO USB AB	USB_Micro_Type_ AB	1
Ferrite BEAD		Ferrite BEAD	L1	RESC0805N	RESC	1
OR		RESISTOR CHIP	R1, R2, R3, R4	RESC0603N	RESC	4
68		RESISTOR CHIP	R5, R6, R7, R8, R9	RESC0603N	RESC	5
4k7		RESISTOR CHIP	R10	RESC0603N	RESC	1
10K		RESISTOR CHIP	R11, R12	RESC0603N	RESC	2
1Meg		RESISTOR CHIP	R13	RESC0603N	RESC	1
B3FS-1015	<a href="http://www.tme.eu/cz/details/b3fs-1015/mikrospina-ce-tact-pcb/omron/">http://www.tme.eu/cz/details/b3fs-1015/mikrospina-ce-tact-pcb/omron/</a>	Mikrospínač; 1- polohové; SPST- NO; 0,05A/24VDC; SMT; 2,55N; 6x6mm	S1	B3FS	B3FS-1015	1
IC_FT245RQ		FT245R USB FIFO	U1	QFN-32	IC_FT245RQ	1
12MHz	<a href="http://www.tme.eu">http://www.tme.eu</a>	CFPX-180	X1	CFPX-180	Crystal 12MHz	1





Title		
Size	Number	Revision
A4		
Date:	29.5.2016	Sheet of
File:	C:\Users\...\Sheet2.SchDoc	Drawn By:





Comment	ComponentLink1 URL	Description	Designator	Footprint	LibRef	Quantity
100nF		CAPACITOR CHIP	C1, C7, C8, C9, C10, C11, C12, C13, C16, C17, C19, C20	CAPC0603N	CAPC	12
4.7uF		CAPACITOR POLARIZED TANTAL	C14, C15	SMA/3216 - 18	CAPC-POL-TANTAL	2
3.3uF		CAPACITOR POLARIZED TANTAL	C18	SMA/3216 - 18	CAPC-POL-TANTAL	1
27pF		CAPACITOR CHIP	C21, C22	CAPC0603N	CAPC	2
LD1117S 3.3V	<a href="http://www.tme.eu/cz/details/ld1117s33tr/stabilizatory-napeti-neregulovane-ldo/">http://www.tme.eu/cz/details/ld1117s33tr/stabilizatory-napeti-neregulovane-ldo/</a>	Stabilizátor napětí; LDO, nenastavitelný; 3,3V; 0,8A; SOT223; SMD	IO1	SOT223	LD1117S33TR	1
2x13_HEADER			J1, J2	2x13-PIN-HEADER	2x13_HEADER	2
USB_Micro_Type_AB		TE CONNECTIVITY / AMP 1981584-1 MICRO USB, 2.0	J7	MICRO USB AB	USB_Micro_Type_AB	1
Ferrite BEAD		Ferrite BEAD	L2, L3	RESC0805N	RESC	2
4.7K		RESISTOR CHIP	R1	RESC0603N	RESC	1
10K		RESISTOR CHIP	R2, R4, R5, R6	RESC0603N	RESC	4
0?		RESISTOR CHIP	R3	RESC0603N	RESC	1
12k		RESISTOR CHIP	R7	RESC0603N	RESC	1
1k		RESISTOR CHIP	R8	RESC0603N	RESC	1
2K		RESISTOR CHIP	R9	RESC0603N	RESC	1
68		RESISTOR CHIP	R10, R11, R12, R13, R14	RESC0603N	RESC	5
B3FS-1015	<a href="http://www.tme.eu/cz/details/b3fs-1015/mikrospina-ce-tact-pcb/omron/">http://www.tme.eu/cz/details/b3fs-1015/mikrospina-ce-tact-pcb/omron/</a>	Mikrospínač; 1-polohové; SPST-NO; 0,05A/24VDC; SMT; 2,55N; 6x6mm	S1	B3FS	B3FS-1015	1
93LC46B-I/MS	<a href="http://www.farnell.com/datasheets/1688094.pdf">http://www.farnell.com/datasheets/1688094.pdf</a>	MICROCHIP - 93LC46B-I/MS - SERIAL EEPROM, 1KBIT, 2MHZ, MSOP-8	U1	MSOP8	93LC46B-I/MS	1
IC_FT2232HQ		Dual High Speed USB to Multipurpose UART/FIFO	U2	QFN-64	IC_FT2232HQ	1
12MHz	<a href="http://www.tme.eu">http://www.tme.eu</a>	CFPX-180	X1	CFPX-180	Crystal 12MHz	1