

UNIVERZITA PARDUBICE  
Fakulta elektroniky a informatiky

Studium možností využití SOC systémů  
Michal Rokyta

Bakalářská práce

2010

UNIVERSITY OF PARDUBICE  
FACULTY OF ELECTRICAL ENGINEERING AND  
INFORMATICS

Study of design possibilities with System On Chip  
Michal Rokyta

Bachelor work

2010

Univerzita Pardubice  
Fakulta elektrotechniky a informatiky  
Akademický rok: 2009/2010

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Michal ROKYTA**  
Osobní číslo: **I07535**  
Studijní program: **B2612 Elektrotechnika a informatika**  
Studijní obor: **Komunikační a mikroprocesorová technika**  
Název tématu: **Studium možností využití SOC systémů**  
Zadávající katedra: **Katedra elektrotechniky**

### Z á s a d y p r o v y p r a c o v á n í :

Zpracovat základní informace o SOC architektuře (system on chip). Zaměřte se na mikroprocesory firmy Cypress PSoC. Seznamte se s jejich strukturou, návrhem analogových a digitálních bloků pomocí vývojového prostředí PSoC Designer 5.0 (obsahuje PSoC Designer a PSoC Express) a vypracujte metodiku návrhu. Možnosti praktického použití demonstруйте na jednoduché ukázkové aplikaci.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

Ashby R.: *Designer's Guide to the Cypress PSoC (Embedded Technology)*, Newnes, 2005 ISBN-13: 978-0750677806

Krüger F.: *PSoC Mikrocontroller*, Franzis Verlag GmbH, 2006, ISBN-13:978-3772353994

Cypress: *Technical Reference Manual (TRM)*

<http://www.cypress.com/design/MR10291>

Vedoucí bakalářské práce:

**Ing. Radim Vondra, Ph.D.**

Katedra elektrotechniky

Datum zadání bakalářské práce: **15. ledna 2010**

Termín odevzdání bakalářské práce: **14. května 2010**



prof. Ing. Simeon Karamazov, Dr.

děkan



Ing. Zdeněk Němec, Ph.D.

vedoucí katedry

V Pardubicích dne 31. března 2010

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 17. 5. 2010

Michal Rokyta

## Poděkování

Rád bych poděkoval vedoucímu bakalářské práce, Ing. Radimu Vondrovi, Ph.D., za rady, připomínky a návrhy týkající se bakalářské práce.

V neposlední řadě děkuji prarodičům a přátelům, kteří mi v průběhu celého studia na vysoké škole pomáhali a podporovali mé úsilí.

## ABSTRAKT

Tato práce se zabývá mikroprocesory firmy Cypress PSoC. Její praktická část se zabývá demonstrací na jednoduché ukázkové aplikaci, vytvořené pomocí vývojového prostředí PSoC Designer 5.0.

## KLÍČOVÁ SLOVA

PSoC, SOC, PSoC® Designer™ , Cypress

## ABSTRACT

This study treat of PSoC systems Cypress In practical parts is mentioned simple, specimen application, created by the help of evolutionary environment PSoC Designer 5.0.

## KEYWORDS

PSoC, SOC, PSoC® Designer™, Cypress



## OBSAH

1. Úvod .....	12
2. SoC-PSoC .....	13
2.1. Historie PSoC .....	14
2.2. Popis architektury .....	18
2.2.1. Jádro (CPU) .....	19
2.2.2. Analogové pole .....	22
2.2.3. Digitální pole .....	23
2.2.4. Systémové prostředky .....	24
2.3. Obchodní dostupnost .....	26
3. Programové vybavení .....	29
3.1. PSoC Designer™ .....	29
3.1.1. Chip-level .....	29
3.1.2. System-level .....	32
3.2. PSoC Programmer™ .....	34
3.3. PSoC Creator™ .....	36
3.4. CyClockWizard 1.0 .....	36
3.5. Ukázkové programy .....	37
3.5.1. Chip-level .....	38
3.5.2. System-level .....	42
3.6. Závěrečné shrnutí .....	51
4. Závěr .....	53
Údaje pro knihovnickou databázi .....	55

## SEZNAM OBRÁZKŮ

Obr. 1 Logo společnosti Cypress Semiconductor .....	14
Obr. 2 Blokové schéma mikroprocesoru .....	18
Obr. 3 Analogový sloupec .....	22
Obr. 4 Vstupní multiplex.....	24
Obr. 5 CY3210+MiniProg1 .....	27
Obr. 6 CY3271 With Cyfi.....	27
Obr. 7 CY3270 – First touch kit.....	28
Obr. 8 Založení projektu .....	30
Obr. 9 Nastavení projektu .....	31
Obr. 10 Pracovní prostředí.....	31
Obr. 11 Programovací okno.....	34
Obr. 12 PSoC Programmer.....	35
Obr. 13 Programátor s možností I2C komunikace .....	37
Obr. 14 Použitý kit.....	37
Obr. 15 Pracovní okno s potřebnými součástkami.....	39
Obr. 16 Nastavení kapacitního slideru .....	40
Obr. 17 Úvodní okno pro výběr typu programu .....	43
Obr. 18 Nastavení slideru .....	44
Obr. 19 Nastavení kapacitních snímačů .....	45
Obr. 20 Tříbarevná dioda .....	46
Obr. 18 Okno simulace .....	47
Obr. 22 Transfer Function .....	48
Obr. 23 Výběr procesoru.....	49
Obr. 24 Připojení prvků na jednotlivé porty .....	50
Obr. 25 Programování .....	51

## SEZNAM TABULEK

Tab. 1 Přehled integrovaných obvodů .....	16
Tab. 2 Přehled specifických obvodů .....	17
Tab. 3 Ukázky z řad procesorů .....	20
Tab. 4 Seznam programovatelných generátorů hodinového signálu, pro aplikaci ClockWizard 1.0.....	37

## SEZNAM ZKRATEK

Acronym	Výklad	Zvolený český ekvivalent
CPU	central processing unit	jádro
CRC	cyclic redundancy check	CRC generátor
CT	continuous time	průběžný čas
ECO	external crystal oscillator	vnější krystalový oscilátor
FLASH		paměť FLASH
GAI	global analog interconnect	sběrnice GAI
GDI	global digital interconnect	sběrnice GDI
ILO	internal low speed oscillator	vnitřní oscilátor pomalých signálů
IMO	internal main oscillator	vnitřní oscilátor
IO	input/output	vstup / výstup
IrDA	Infrared Data Association	komunikační port IrDA
LVD	low voltage detect	podnapěťová ochrana
MAC	multiplay accumulator	znaménková násobička
POR	power on reset	podnapěťový mod
PRS	pseudo random sequence	modul PRS
PSoC	Programmable System-on-Chip	programovatelný procesor - PSoC
PWM	pulse width modulator	modul PWM
RAM	random access memory	paměť RAM - datová
ROM	read only memory	paměť ROM - programová
SC	switched capacitor	spínací kapacitor
SMP	switch mode pump	prvek napěťové regulace
SOC	systém - on - a - chip	procesor SOC
SPI	serial peripheral interconnect	sběrnice SPI
SRAM	static random access memory	paměť SRAM
UART	universal asynchronous receiver-transmitter	komunikační modul UART
USB	universal serial bus	USB
WDR	watchdog reset	reset při zauzlení
XRES	external reset	externí reset

## 1. Úvod

Prezentovaná bakalářská práce soustřeďuje základní informace o univerzálně konfigurovatelných mikroprocesorech PSoC , jejich struktuře, způsobu a možnostech jejich programování podle požadavků koncového uživatele.

Práce je členěna do 4. kapitol, soustřeďující se vždy na jedno ucelené téma.

První kapitola popisuje stručně zaměření bakalářské práce.

Druhá kapitola zahrnuje stručný přehled historie vývoje programovatelných mikroprocesoru, jejich současné struktuře. Popisuje jednotlivé části. Obsahuje rovněž zmínku o obchodní dostupnosti.

Třetí kapitola je zaměřena na možnosti a způsoby programování. Zahrnuje výčet nástrojů programování a na jednoduchých příkladech ukazuje možnosti praktického užití. Nezbytnou součástí je stručná závěrečná sumarizace poznatků autora.

Čtvrtá kapitola je věnována stručnému závěrečné sumarizaci získaných poznatků a hodnocení.

## 2. SoC-PSoC

Jedním ze směrů při vývoji, vedoucím k vysoké integraci obvodů, byla metoda tzv. „systému na čipu „

### **SoC** - System on Chip

Jde o architekturu, integrující všechny součásti počítače nebo jiného elektronického zařízení do jednoho čipu. Tento čip většinou obsahuje také různé periferie, jako jsou například bloky pro zpracování analogových a digitálních signálů nebo pro práci s bezdrátovou technologií. Typickou aplikací je pak jeho užití ve vestavěných systémech.

Architektura obsahuje tedy vždy vedle vlastního výkonového procesoru i pro jeho práci potřebnou paměť. Užití tohoto systému se vyskytuje nejčastěji v malých, ale stále složitějších elektronických zařízeních. Proto se, se systémy užívajícími SOC architekturu velmi často setkáváme například v PDA, herních konzolách, domácí elektronice nebo v mobilních telefonech.

Typická SOC architektura je tvořena z následujících částí:

- paměťovými bloky, nejčastěji s možností výběru ROM, RAM, FLASH.
- oscilátory, časovači, generátory pro napájecí a
- jedním mikroprocesorem nebo digitálním signálovým procesorem (někdy se vyskytují multiprocesorové systémy)
- bloky pro práci s vnějšími rozhraními jako jsou USB, FireWire, Ethernet ...
- analogovým a digitálním rozhraním
- regulátory napětí a obvody pro řízení spotřeby energie.

Při vývoji, malosériové výrobě, ale i při praktickém užívání těchto obvodů stáli pracovníci často před problémem, jak rychle a s minimálními náklady přidat další periferní zařízení. Potřebné doplnění stávajícího systému o další komponenty, zpravidla vyvolalo, mimo jiné, potřebu dalšího místa na desce plošných spojů, eventuelně zvýšení spotřeby elektrické energie. V souhrnu to pak znamenalo vyšší nároky na čas a potřebu dodatečných finančních nákladů.

Odstranit valnou většinu těchto problémů umožnil vývoj univerzálně konfigurovatelného mikroprocesoru -

## **PSoC** - Programmable System-on-Chip

a jeho následné, urychlené zavedení do rutinní uživatelské praxe.

Užití těchto mikroprocesorů je obzvláště výhodné pro zpracování signálů z různých senzorů. Ať již jde o analogový nebo digitální výstup. Další a to ne nepodstatnou výhodou je jejich nízká spotřeba energie, protože nepoužívané bloky zůstávají odpojeny od napájecího napětí.

### **2.1. Historie PSoC**

Vývoj PSoC probíhal ve společnosti Cypress Semiconductor.



**Obr. 1 Logo společnosti Cypress Semiconductor**

Její PSoC výrobní program patří v současnosti k nejrozšířenějším a nejucelenějším. Většímu rozšíření užívání výrobků tohoto typu v České republice však brání patrně i nižší informovanost technické veřejnosti. A proto prezentovaná práce by mohla být i nepatrným příspěvkem k lepší informovanosti a k eventuelnímu většímu praktickému užití tohoto progresivního výrobku.

V roce 2001 ukončila firma Cypress Microsystems vývoj a uvedla na trh prvou generaci programovatelných systémů založených na SoC architektuře označovaných PSoC, označovaných jako

**CY8C 25XX x XX** nebo **CY8C 26XX x XX**.

Oba typy měly paměť FLASH o kapacitě 4–16 kilobytů, paměť SRAM o kapacitě 256 bytů, 8 digitálních a 12 analogových bloků v pouzdrech s 8-44 piny.

Tyto systémy byly velmi brzo nahrazeny druhou generací pod označením

**CY8C 27 x xx**

s lepším zpracováním signálů a konečně řadou CY8C 29x xx zajišťující mnohem dokonalejší zpracování digitálních signálů. Základní technické parametry jednotlivých PSoC ze současné-ho výrobního programu firmy Cypress Microsystems jsou uvedeny v tab. 1 přehled integrovaných obvodů.

Podle posledních informací firmy je typ CY8C 22 x 13 již považován za výběhový a nahrazován typem CY8C 24 x 23.

Firma svou flexibilitou se snaží pokrýt i velmi speciální požadavky odběratelů, jak dokumentuje přehled několika vybraných systémů, uvedených tab. 2 přehled specifických obvodů.

O z n a ě n í	Digitální systém			Analogový systém				Paměť	
	počet			počet				bytů	
	vstupů / výstupů	buněk	bloků	vstupů	výstupů	kolon	bloků	SRAM	FLASH
CY8C 21 x 23	16	1	4	8	0	2	4*	256	4000
CY8C 21 x 34	28	1	4	28	0	2	4*	512	8000
CY8C 22 x 13	18	1	4	8	1	1	3	256	2000
CY8C 24 x 23	24	1	4	12	2	2	6	256	4000
CY8C 24 x 23A	24	1	4	12	2	2	6	256	4000
CY8C 24794	50	1	4	48	2	2	6	1000	16000
CY8C 27 x 43	44	2	8	12	4	4	12	256	16000
CY8C 27 x 66	44	2	8	12	4	4	12		
CY8C 29 x 66	64	4	16	12	4	4	12	2000	32000

Tab. 1 Přehled integrovaných obvodů

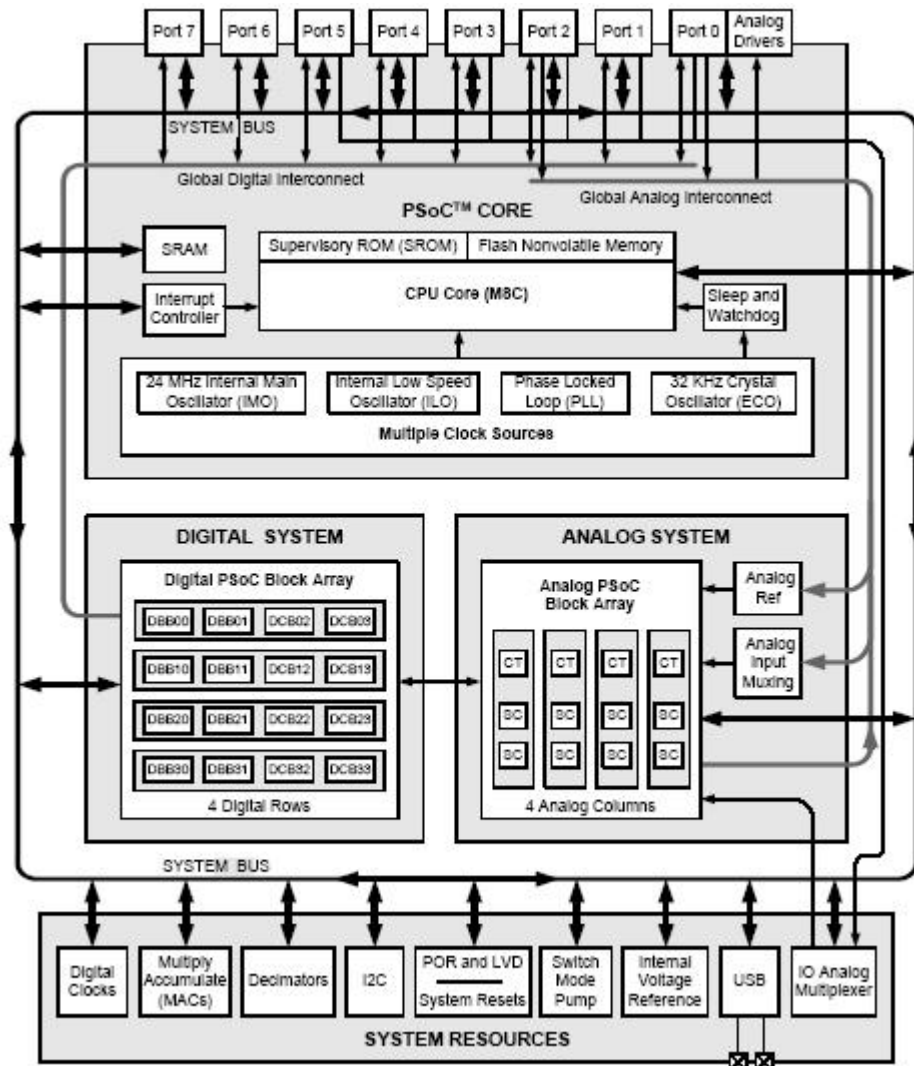


O z n a č e n í	Digitální systém	Analogový systém	Vstup / Výstup	Paměť		POZNÁMKA
	počet bloků	počet bloků	počet - do	bytů	bytů	
				SRAM	FLASH	
CY8C 20 x 34	0	1	32	512	8000	I2C / SPI
CY8C 24 x 94	4	6	100	1000	16000	USB
CYWUSB 6953	4	0*4	48	512	8000	Bezdrátového USB

**Tab. 2 Přehled specifických obvodů**

## 2.2. Popis architektury

Pro popis vnitřního uspořádání integrovaného obvodu PSoC, umožňující seznámit se s jeho částmi a jejich vzájemným propojením bylo použito blokové schéma uvedené na následujícím obr. 2. Blokové schéma mikroprocesoru.



Obr. 2 Blokové schéma mikroprocesoru

Z tohoto blokového schématu je zřejmé, že obvod PSoC tvoří následující základními čtyřmi segmenty :

- Jádro ( PSoC CORE )
- Analogový segment ( Analog system )
- Digitální segment ( Digital system )
- Systémové prostředky ( System resources )

Nezbytnou součástí obvodu je propojovací pole , zajišťující jak vzájemné propojení jednotlivých segmentů tak spojení se soustavou vstupních a výstupních pinů . Je tvořeno zejména :

Analogovou sběrnici ( Global Analog Interconnect )

Digitální sběrnici ( Global Digital Interconnect )

Systémovou sběrnici ( System BUS )

Podrobnému popisu jednotlivých segmentů a jejich funkci jsou věnovány další pododdíly .

### **2.2.1. Jádru (CPU)**

Tento segment je základní součástí , která vydává pokyny pro splnění požadavku daného příchozí instrukcí a kontroluje chod jednotlivých součástí.

V současnosti je možno setkat se třemi různými druhy jader u procesorů PSoC a to :

PSoC 1 s jádrem M8C

PSoC 3 s jádrem 8051

PSoC 5 s jádrem ARM Cortex – M3

Přehled základních charakteristik těchto řad umožňující jejich porovnání je uveden v tab. 3 ukázky z řad procesorů.

K nejrozšířenějším v produkci firmy CYPRESS patří PSoC s jádrem M8C. Tento typ jádra je založen na čtyřnásobné MIPS 8 - bitové Harvardské architektuře. A proto jeho základní charakteristiky budou popsány v následujících odstavcích .

Jádru obsahuje **5 interních registrů** mezi které patří:

*programový čítač* – ukazuje na adresu následné vykonávané funkci

*ukazatel zásobníku* – ukazuje v paměti SRAM ze kterého je čteno

*akumulátor* – užívá ALU pro aritmetické, logické operace nebo operace přesunu

*index registr* – pro indexaci a někdy jako akumulátor

*registr příznaků* – pro popis výsledků již vykonaných instrukcí

	PSoC 1	PSoC 3	PSoC 5
Jádro	8MC	8051	32-ARM Cortex-M3
Maximální frekvence	24MHz	67MHz	80MHz
Počet milionu instrukcí za sekundu	4	33	100
Současný počet procesorů	85	181	115
Počet sérií	10	4	4
Například série	CY8C21x23	CY8C32xxx	CY8C52xxx
	CY8C23x33	CY8C34xxx	CY8C53xxx
	CY8C29X66	CY8C36xxx	CY8C54xxx
	CY8C24x94	CY8C38xxx	CY8C55xxx

Tab. 3 Ukázky z řad procesorů

S ohledem na harvardskou architekturu, kdy paměti programu a dat jsou odděleny, jsou ze **3 adresních prostorů** k dispozici současně jen dva. Adresní prostory tvoří:

*paměť programu (ROM)* - do které je ukládán zdrojový kód. Její realizace technologií FLASH umožňuje velký počet přepisů. Její velikost se liší podle druhu mikroprocesoru a může dosáhnout až 32 kB. Má vyhrazenou řídicí část která se používá při startu, nebo základním nastavení komponent .

paměť dat (RAM)

registry – celkem 512 registrů je rozděleno do dvou bank po 256 bytech. Pro správný přístup je nutno nastavit správnou banku.

Pro správný a koordinovaný chod procesoru jako celku a jednotlivých jeho částí je zapotřebí **generátor hodinových signálů**. Tento požadavek zajišťují dva nezávislé systémy :

*vnitřní 24MHz oscilátor (IMO)* – základní zdroj signálů, plní i funkci referenčního signálu a je vždy použit při startu procesoru . Pro běžný provoz může být nahrazen externím oscilátorem až do hodnoty 24 MHz.

*vnitřní oscilátor pomalých signálů (ILO)* - s frekvencí 32kHz je používán pro činnosti se sníženou spotřebou energie. Pro případ zálohování nebo dosažení vyšší přesnosti je možno propojit systém tohoto oscilátoru s *externím krystalovým oscilátorem (ECO)* , pracujícím se stejnou frekvencí 32kHz .

Nezbytnou součástí pro komunikaci s okolím je **soustava pinů** a to vstupních, výstupních a výstupně-vstupních. Najednou je přístupno vždy 8 pinů , které tvoří součást jednoho portu.

Závěrem je nutno poznamenat , že v systému je možno použít i časový signál s frekvencí 48 MHz , ale při značně vyšší spotřebě energie. Tabelovaná hodnota napájecího napětí je buď 3,3V nebo 5V.

## 2.2.2. Analogový sektor

Základním prvkem analogového sektoru je programovatelný blok, do něj lze umístit :

A / D převodníky s maximálním rozlišením až 14 bitů

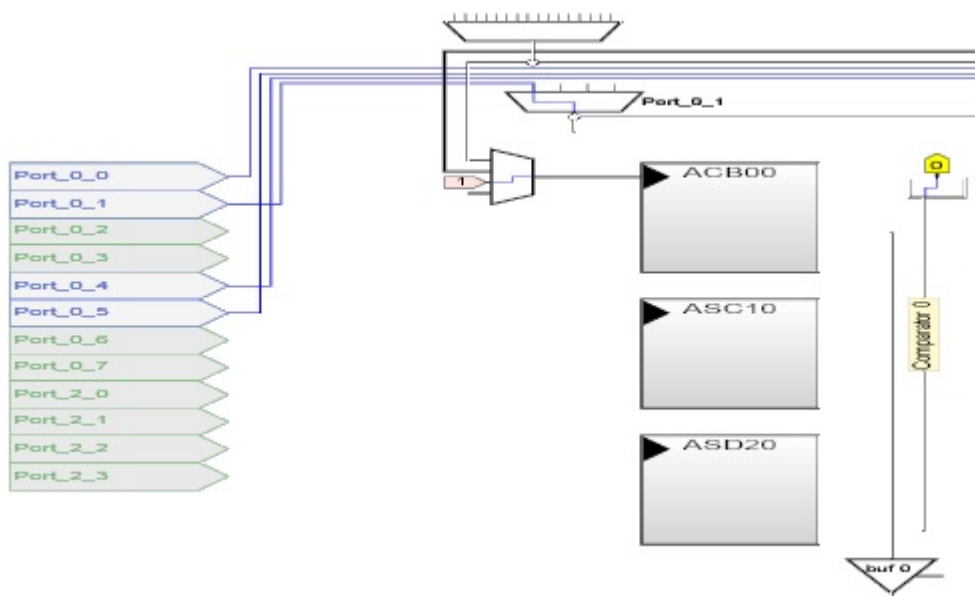
D / A převodníky s maximálním rozlišením až 9 bitů

programovatelné napěťové zesilovače

programovatelné filtry

programovatelné komparátory

Analogové bloky jsou uspořádány do sloupců, kterých bývá 1 až 4, dle typu procesoru. Každý sloupec tvoří většinou tři , výjimečně dva analogové bloky. Každý ze sloupců má vlastní analogový multiplexer, multiplexer hodinového signálu, a výstupní analogové a komparátorové linky. Struktura a vazby sloupce s třemi bloky je schematicky znázorněna na obr. 3 analogový sloupec.



Obr. 3 Analogový sloupec

Při realizaci analogových sloupců se užívá dvou druhů programovatelných bloků a to - CT (continuous time) a SC – switched capacitor.

Při realizaci sloupce s třemi bloky jej tvoří bloky :

ACB (typ od CT) přizpůsobený pro vkládání zesilovačů. Podle vnitřního zapojení je možné provést realizaci invertujícího nebo

neinvertujícího zesilovače, případně komparátoru. Jako vstupní hodinový signál nejčastěji používá signál z multiplexeru.

ASC a ASD (typ od CT) obsahují analogové zesilovače, vstupní multiplexery, AD a DA převodníky.

Při realizaci sloupce s dvěma bloky jej tvoří bloky :

ACE ( typ od CT )

ASE ( typ od SC )

Každý analogový sloupec má alespoň jednu komparátorovou linku. Ta může být použita pro digitální vstup některého z analogových bloků a porovnáním vstupního napětí umožňuje povolit některé funkce jako čítač, časovač PWM a nebo jiné digitální periferie. Pro analogové bloky je možné použít také interní hodinový signál VC1 a VC2.

### **2.2.3. Digitální sektor**

I u digitálního sektoru je základním prvkem programovatelný blok , do kterého je možno umístit :

čítače

časovače 8 , 16 , 24 , 32 bitů

PWM moduly

PRS moduly

CRC generátory

1 až 2 duplexy UART

SPI – Master nebo Slave

IrDA

Programové bloky jsou seřazena do 1 až 4 vodorovných řad , jejich počet se mění podle druhu procesoru. Každá řada přitom obsahuje čtyři programovatelné bloky. I když je možné použít pro umístění komponent kterýkoliv blok, pro umístění komunikačních modulů (SPI, UART aj.) platí jisté pravidlo a to, že a je nutno je umisťovat do pravé poloviny digitálních bloků.

Pro správný chod digitálních bloků je potřeba zajistit hodinový signál.

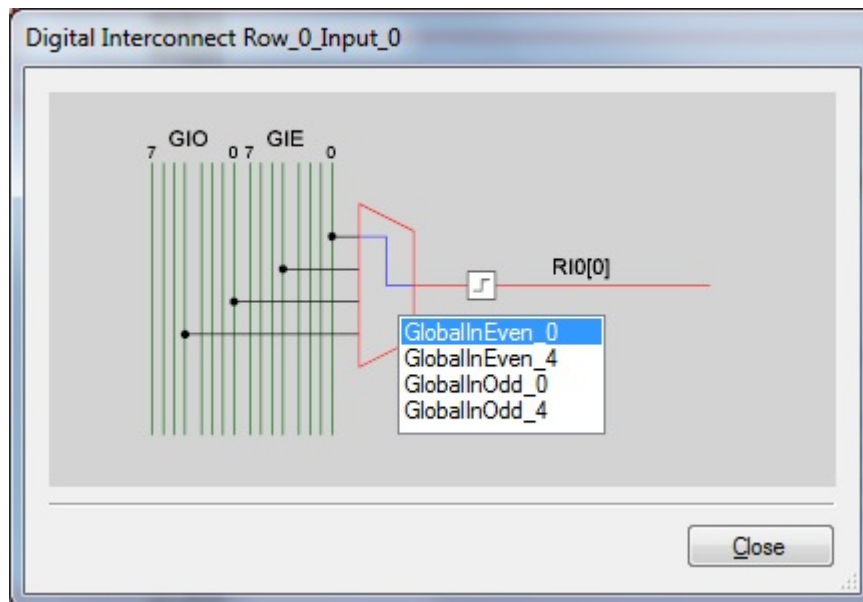
S ohledem na předpokládanou spotřebu elektrické energie a požadovanou rychlost je možno zvolit některý vnitřní zdroj hodinového

signálu (např. V 1 – 3 , CPU 32 kHz aj.) nebo použít výstup z jiného digitálního bloku, naprogramovaného jako čítač a využít jej ve funkci děliče napětí.

Propojení vstupně – výstupních pinů s digitálními bloky je realizováno přes globální linky a multiplexery. Globální linky jsou GIO s lichým indexem a GIE se sudým indexem .

Platí pravidlo, že mohou být propojeny pouze piny a linky označené stejným indexem.

Na vstupním multiplexeru je možno zvolit, zda propojení bude realizováno prostřednictvím globální linku GIO a nebo GIE jak zřejmé z obr. 4 vstupní multiplex.



Obr. 4 Vstupní multiplex

#### 2.2.4. Systémové prostředky

V tomto segmentu jsou soustředěna všechny prostředky , které umožňují maximální flexibilitu PSoC podle požadavků uživatele . Jejich rozsah je závislý na typu zvoleného procesoru a jeho užití . Při dodržení výčtu a označení těchto prostředků z obr.2 je na následujících řádcích uvedena jejich stručná charakteristika :

**Digital Clocks** součástí systém hodinových signálů tvořený z vnitřních a vnějších zdrojů umožňující vzájemnou propojenost se spotřebiči.



**MAC** 8 bitová násobička s možností uložení výsledku v 32bitovém akumulátoru .Podle typu procesoru a praktického užití je realizována v 0 ,1 nebo 2 blocích a s akumulátorem nebo bez akumulátoru.

**Decimators** modulátor hustoty analogického signálu.

**I2C** řadič , umožňující realizovat I<sup>2</sup>C v režimu Slave nebo Masters.

**POR + LVD** zajišťují ochranu před poklesem napětí

**System resets** existují 3 možnosti resetování a to:

*POR* je využíván když během práce s procesorem dojde k poklesu napájecího napětí pod kritickou mez. V tom případě je spuštěn Power On Reset mod, ve kterém je procesor do doby než proběhne ustálení napájecí hodnoty na hodnotu jenž je definována parametrem Trip Voltage.

*XRES* externí reset, umožňující uvést mikroprocesor do předem definovaného stavu. Nejjednodušší realizací je připojení tlačítka s odporem na resetovací port.

*WDR* využívá se pro resetování, když dojde k zacyklení z nepředvídatelných důvodů a vzniku následných nepříjemností.

**SMP** je možno považovat za součást napěťové regulace. Propojením s obvodem LC umožňuje překlenou dočasný rozdíl mezi zdrojem a spotřebou .

**Internal Voltage Reference** vnitřní napěťový normál s korekcí na teplotní závislost.

**USB** je použit systém USB 2.0 s operační rychlostí 12 MB/sec umožňuje prostřednictvím dvou speciálních USB pinů propojení s exteriérem.

**IO Analog Multiplexer** systém umožňující u některých procesorů odstranit problém nedostatku možností propojení pinů s analogovými bloky tím, že zajistí propojení na společnou vnitřní sběrnici.

### **2.3. Obchodní dostupnost**

Přestože se nejedná o technickou novinku je , není dostupnost procesorů PSoC v ČR není zatím moc velká. V současné době je možno , dle získaných informací , získat tyto procesory u 4 společností a to :

- Cypress Semiconductor - samotný výrobce
- MSC-VERTRIEBS-CZ s.r.o.
- Microdis s.r.o.
- Farnell

pro zájemce o studium tohoto zařízení nebo začínající uživatele, lze doporučit následující postup:

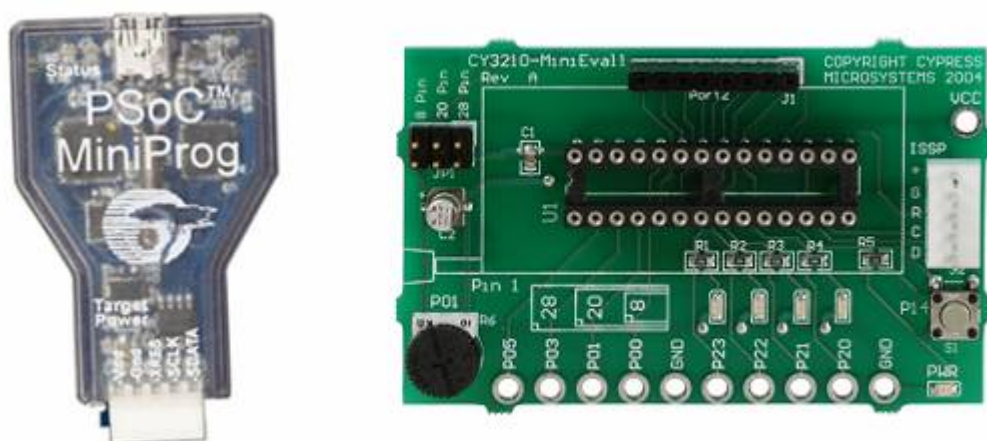
pořídit některý ze startovních kitů a následně po bezplatné registraci na stránkách výrobce si stáhnout nejnovější verzi aplikace PSoC Designer™ nebo PSoC Creator™, tím získat samotná programovací prostředí pro PSoC 1, nebo PSoC 3 a PSoC5.

Pro první seznámení s vytvářením programů může postačit PSoC Designer™, obsahující zjednodušenou verzi PSoC Programmer, ale pro programování v plné šíři je zapotřebí získat samotný PSoC Programmer. Jehož nejnovější verze nese označení 3.10.

Pokud jde o samotné kity, tak výrobce - společnost Cypress Semiconductor nabízí celou řadu pod označením First touch kit a nebo také Training Kit.

Pro začátek lze ze série First touch kit, použít následující:

## CY3210-MiniProg1



Obr. 5 CY3210+MiniProg1

Součástí balení tohoto kitu je i miniprogramátor , k němu je přiložena pokusná DPS a pár základních obvodů jako vzorek . Rovněž je přiložen USB kabel pro připojení programátoru k PC.

## CY3271 with Cyfi



Obr. 6 CY3271 With Cyfi

Obsahem balení je CD s referenční příručkou o samotném kitu ukázky, jeho zapojení a dále programovací prostředí. Rovněž obsahuje program na možné vyzkoušení Cyfi technologie při bezdrátovém měření

teploty. Dále balení obsahuje programátor, který umí i Cyfi technologii a 3 testovací DPS. První DPS umožňuje vyzkoušet si práci se součástkami jakou jsou 3-barevná dioda, dále termistor pro možnost snímání teploty, fotorezistor, kapacitní snímač. Dále kit obsahuje 2 DPS pro možnost napájení desky s vysílačem. Jde o 2 možnosti a to napájení z ploché „peckové“ baterie nebo z klasických AAA baterií.

#### CY3270 - FIRST TOUCH KIT



**Obr. 7 CY3270 – First touch kit**

Součástí je programátor podobný programátoru u CY3271 s tím rozdílem, že neumí CYFI technologii. Dále balení obsahuje testovací desku s kapacitním slidrem, termistorem, fotočidlem ... .

Z uvedeného přehledu bude se pro většinu zájemců jevit jako nejvšestrannějším kit CY3271, jistým omezením může však být jeho pořizovací cena. Ale i přes ní je při porovnání všech vlastností oproti ostatním kitů nejvýhodnější.

### 3. Programové vybavení

Výrobce, společnost Cypress nabízí vývojová prostředí pro práci se všemi třemi architekturami. Všechna tato prostředí jsou poskytována zdarma po bezplatné registraci a následném vyplnění krátkého dotazníku pro osobní potřebu společnosti Cypress. Tento dotazník nám později umožní i získání vzorků procesorů. Pro architekturu PSoC 1 slouží program PSoC Designer<sup>TM</sup>. Architektura PSoC 3 a 5 se dočkala také svého programovacího prostředí, kterým je PSoC Creator<sup>TM</sup>. Všechny architektury mají společné prostředí pro nahrání programu do procesoru a tím je PSoC Programmer<sup>TM</sup>.

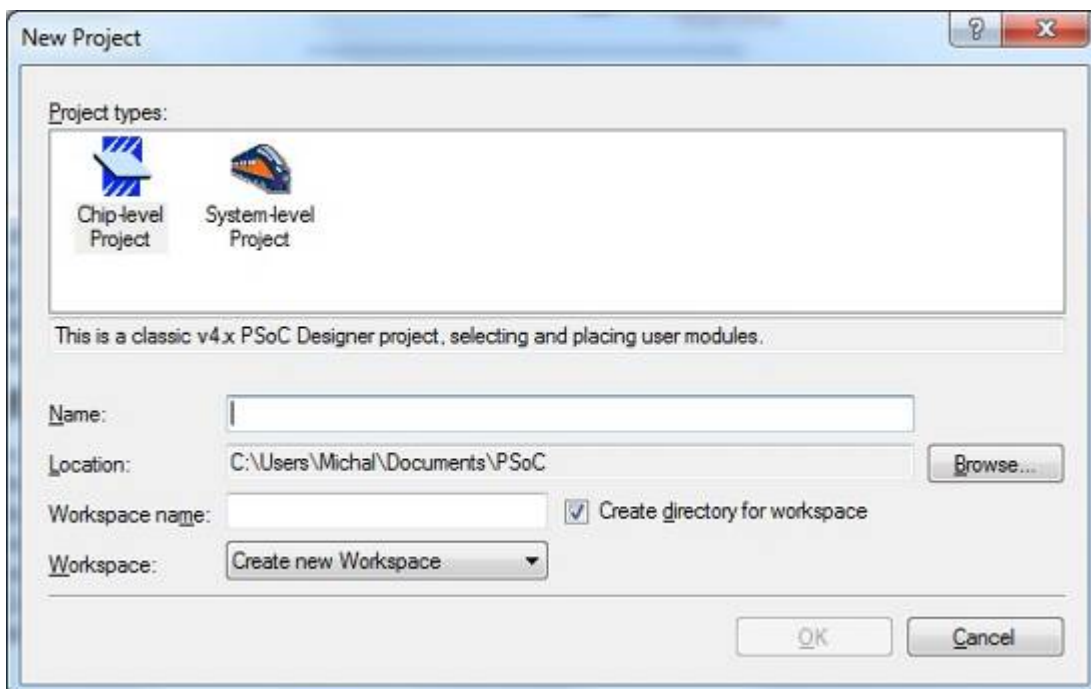
#### 3.1. PSoC Designer<sup>TM</sup>

Jak již bylo řečeno toto prostředí slouží pro naprogramování první architektury. Samotný program se skládá z 2 podprogramů a těmi jsou Chip-level a Systém-level. Tyto 2 podprogramy dříve vystupovali samostatně sami PSoC Designer 4.4 a PSoC Express 3.0 při uvedení programu pro architekturu 3 a 5 došlo k jejich sloučení do jediného programu.

##### 3.1.1. Chip-level

Jedná se o programovací prostředí spíše pro osoby, zkušenější v programování. Toto prostředí je podobné například AVR studiu. Programování a samotnou konfiguraci může provést dvojím způsobem. Za prvé je možno celý program napsat ručně anebo si do programu „naklikáme“ dostupné předem připravené bloky a ty si nastavíme buď v okně vlastností a pokračujeme v psaní kódu a nebo si toto nastavení také napíšeme ručně. Ať již nastavení v okně vlastností nebo napsání přímo kódu se používá častěji než samotné psaní kódu. Samotný program je psán v jazyce C, nebo Assembleru. Od počátku je program vytvářen pro určitý typ procesoru, samozřejmě je, možno před výběrem ověřit, zdali vybraný procesor bude stačit na daný program a bude umět pracovat se všemu použitými prvky. Toto ověření si uživatel provede při výběru procesoru přímo v programu. Spuštění podprogramu se provede spuštěním programu PSoC Designer<sup>TM</sup> následným vybráním založení nového projektu, objeví se okno, které je vidět

na založení projektu obr. 8 založení projektu. V tomto okně se klepnutím na Chip-level Project provede vybrání této části programu. Nyní je potřeba zadat nějaký název programu a nastavit cílovou složku kam se bude program ukládat, jestliže nemá dojít k použití výchozí složky. Název programu jako spousta anglických programů podobného počítá se zadáním jednoslovného názvu programu. V případě, že chceme použít víceslovný název je možné použít jako oddělovací symbol \_ . Po kliknutí na tlačítko OK se zobrazí obr. 9 nastavení projektu. V tomto okně je potřeba vybrat typ procesoru pro který se bude program tvořit a jakým programovacím jazykem. Jak již bylo řečeno při kliknutí na View Catalog ... se vybere procesor a zároveň si v tabulce ověří zdali zvládne vše co je od něj očekáváno.

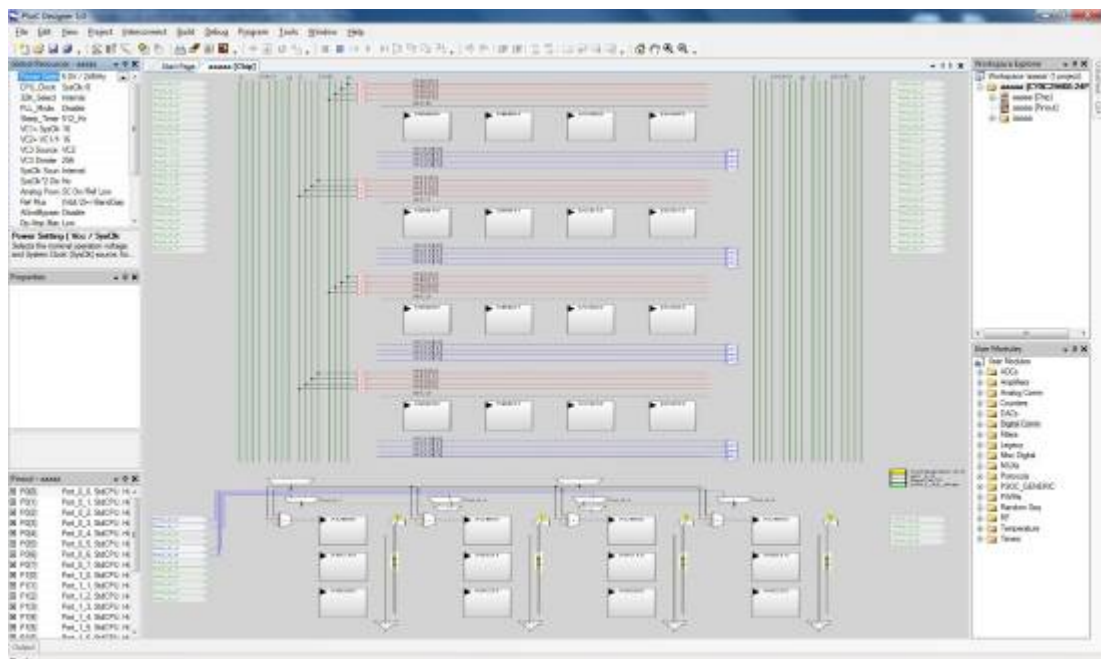


Obr. 8 Založení projektu



**Obr. 9 Nastavení projektu**

Po kliknutí opět na tlačítko OK se po chvíli zobrazí již připravená struktura vybraného procesoru, tu je možné vidět například na obr. 10 pracovní prostředí.



**Obr. 10 Pracovní prostředí**

Samotné okno se skládá z několika podoken. První podokno je okno Global Resource, kde je možné provést nastavení některých vlastností procesoru jako napájecí napětí, signál pro generování hodin.

Pod tímto podoknem se nám nacházejí další 2 podokna, první z nich ukazuje vlastnosti součástky jenž byla vybrána. Při výběru součástky je zde možné měnit vlastnosti, jako například název součástky na jaký port bude připojena. Tyto vlastnosti bývají většinou u každé součástky. Následující podokno dává uživateli přehled všech portů a také podává informaci co se na daném portu nachází.

Prostřední největší podokno je samotná pracovní plocha, na kterou se přetahují potřebné součástky. Díky tomu, že došlo k vybrání typu procesoru program vygeneroval automaticky strukturu a počet digitálních segmentů a pod ně analogových pro vybraný procesor. V pravé části se nám nacházejí pod sebou 2 podokna. V prvním je možné vidět samotnou adresářovou strukturu programu. A v druhém pod ním se provádí výběr některé z již předdefinovaných struktur součástek.

Jestliže si tvůrce programu vybere některou součástku, jsou 2 možnosti pro její umístění. První je přetáhnout danou součástku do pracovního okna a případně jí usadit do některého z bloků. Anebo druhou a ta je snažší je na danou součástku klepnout pravým tlačítkem a vybrat volbu place. Ta nám součástku usadí na volné místo do potřebného typu bloku. Některé součástky jsou pouze v podokně průzkumníka. A to proto, že neobsazují žádný analogový ani digitální blok.

### **3.1.2. System-level**

Tato část programu je velice vhodná pro uživatele, kteří nejsou tak zblhlí v programování. Jedná se o velmi rychlý nástroj pro vyváření jednoduchých aplikací, který je založen plně na „klikání“. Zde se program nepíše ručně, jeho vytvoření se provádí nataháním jednotlivá bloků. Následně se správná funkce aplikace provádí doklikáním podmínek za kterých se má co dít. S takovýmto stylem tvoření programu je možné se setkat například v podprogramu programu Matlab, kde se tvorba simulace provádí stejným stylem také.

Samozřejmě tento způsob programování má svou nevýhodu a tou je omezení programátora v nastavování pro specifického chování jednotlivých prvků, ve srovnání s možnostmi programu Chip-levelu. Velikou výhodou, která toto omezení předčí, je možnost „naklikat“ program bez nutnosti si

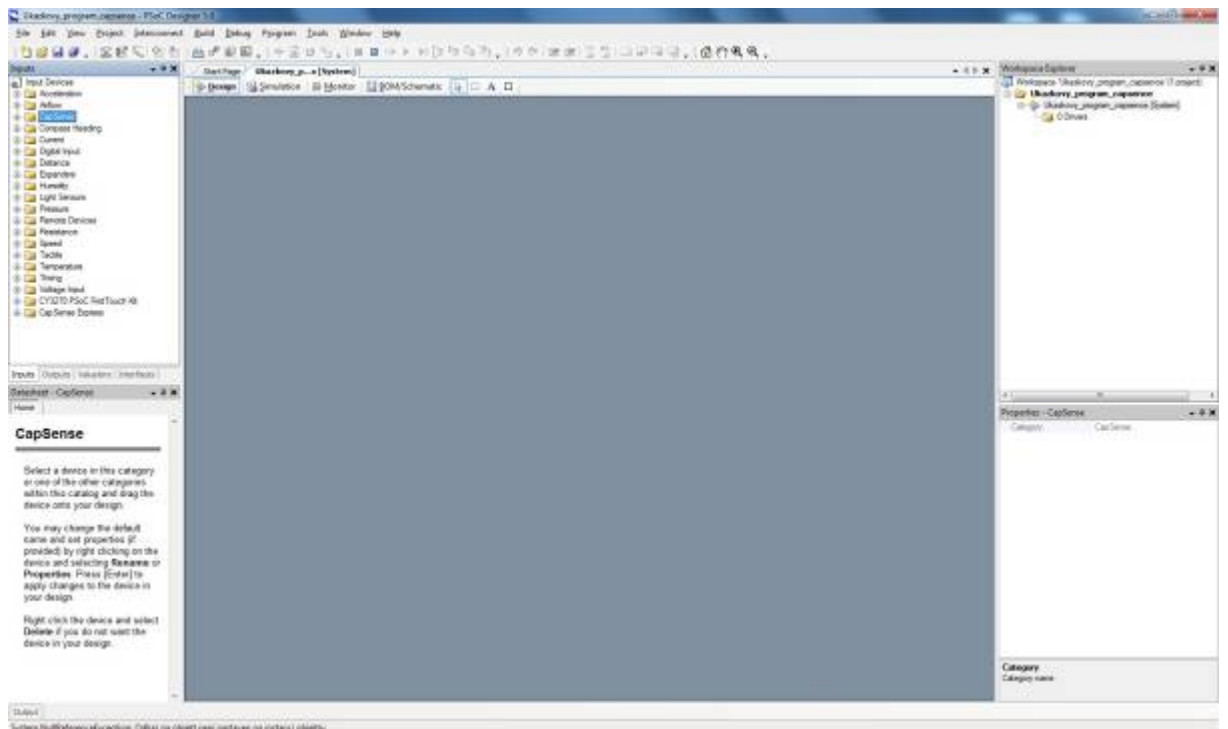


pamatovat nějaké příkazy. To určitě ocení uživatelé méně zblhlí v programování.

Mezi výhody této části programu je možné zařadit vlastnost, že program si sám hlídá, zdali je dostupný procesor pro danou aplikaci vhodný a člověku odpadá nutnost studování datasheetů pro následný výběr typu procesoru, se kterým bude projekt realizován. Program si tuto věc hlídá sám a případně upozorní, že nemá procesor, který by danou práci byl schopen vykonávat. Také při překládání programu uživateli nabídne pouze ty procesory, které jsou schopné daný program splnit.

Po spuštění nadřazeného programu PSoC Designer™ se objeví opět okno, které je vidět na obr. 8 založení projektu. Kliknutím na System-level Project a napsáním názvu programu se otevře samotné prostředí podobné například grafické aplikaci programu C# v programu Microsoft Visual Studiu. Toto okno je znázorněno na obr. 11 programovací okno.

Na první pohled se zdá podobné okno Chip-level. Ale jde opravdu jen o částečnou podobu. Okno se opět skládá opět z několika podoken. Stejně je pouze okno průzkumníku v pravo. Pod ním se nachází podokno vlastností. Uprostřed se nachází samotná pracovní plocha kam přetahujeme potřebné součástky a následně se zde provádí jejich propojení a nastavení podmínek. Toto podokno má své užitečné záložky. Mezi tato záložky patří možnost simulace daného programu, dále monitorování v případě, že probíhá nějaká komunikace s počítačem. A také je možno nechat si vygenerovat schéma zvoleného projektu anebo i kompletní datasheet hotového a přeloženého projektu. Vlevo se nachází 2 podokna, ve vrchním se provádí výběr součástky jenž chceme použít. Součástky jsou děleny do několika částí těmi jsou vstupní a výstupní součástky. Dále jsou to součástky pro komunikaci s PC.

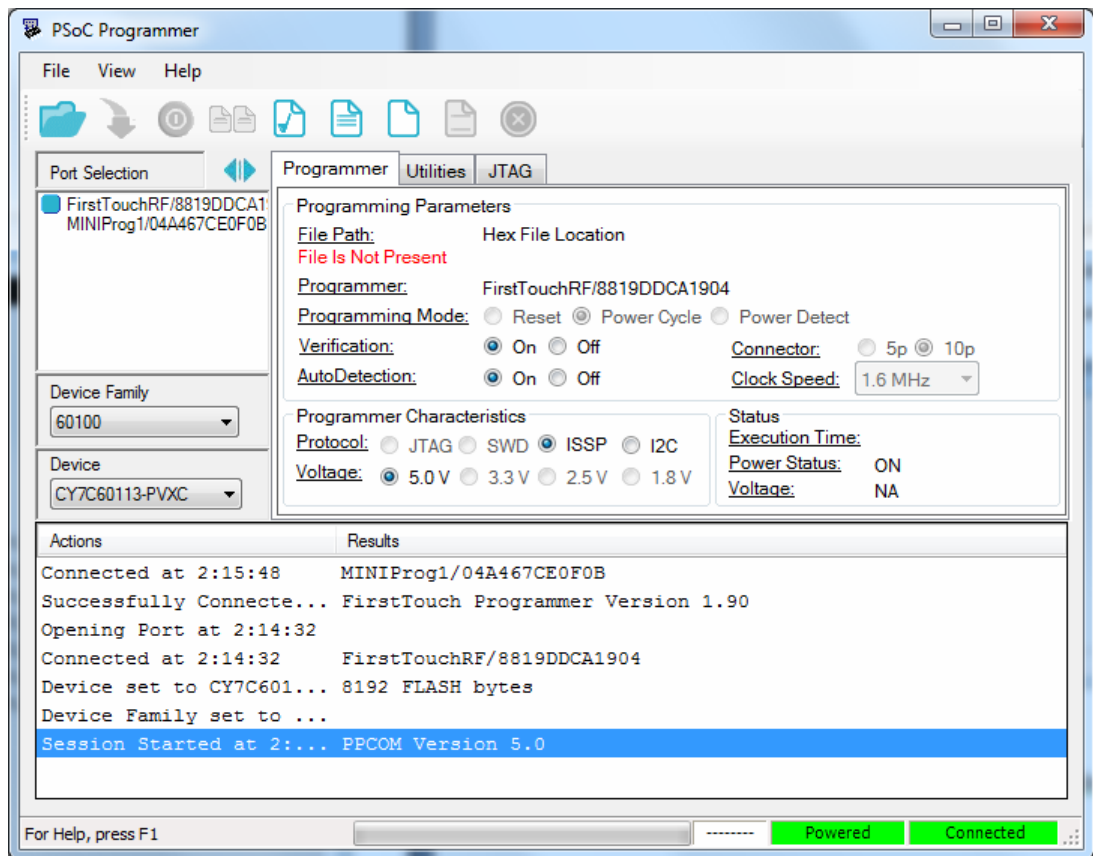


Obr. 11 Programovací okno

### 3.2. PSoC Programmer™

Tento program slouží nahrání vytvořeného programu do procesoru. Pro nahrání programu do procesoru je možné použít jednodušší verzi programátoru, která je součástí. Je možné při tvorbě použít menší variantu programátoru. Zjednodušenou variantu programátoru obsahuje jako svojí součást vývojové prostředí programu PSoC Designer™ během vytváření finální podoby programu v nabídce Program. Ale odlehčená verze je určená pouze pro programování řady PSoC1 jakožto celý program. V případě, že chceme provést programování řady PSoC3 a PSoC5 je nutné použít externí program k naprogramování.

Tento program je možné si stáhnout zdarma na stránkách výrobce. Po stažení a následné instalaci si jej můžeme spustit přes ikonku PSoC Programmer™. Po jeho spuštění se nám zobrazí okno, které je možno vidět na Obr. 12 PSoC Programmer.



**Obr. 12 PSoC Programmer**

Samotnou práci je možné ovládat 2 možnostmi a těmi jsou řada ikoněk a nebo příkazy z nabídky File. Okno je členěno na několik částí:

Programming Parameters – v tomto podokně je adresa, kde se nachází vybraný soubor, který budeme programovat. Dále jsou zde informace o vybraném programátoru, který si v případě více programátorů, můžeme vybrat v podokně Port Selection a vybraný programátor je označen modrým čtverečkem. Dále se zde nachází informace o tom, zdali má být programování provedeno při resetu procesoru a nebo automaticky ... .

Programmer Characteristic – možnost nastavení podnabídek protocol a voltage je možná pouze u některých programátorů. Tyto vlastnosti záleží na daném typu programátoru. Například MiniProg má pouze vyvedený programovací konektor, nejčastěji používaný ISSP konektor. Existuje také programátor z kity CY3271. Tento programátor po vybrání automaticky na svůj výstup, na rozdíl od programátoru MiniProg, pouští napájecí napětí. Obsahem tohoto programátoru je i vyvedení I2C komunikace a tudíž, existuje i možnost vybrat protokol I2C. Ale jak již bylo řečeno většinou se pro programování využívá ISSP konektor.

Status – v tomto okně je například možnost podívat se na informaci, zdali je do procesoru přes programátor přivedené napájecí napětí.

Device Family – zde se vybere pro jaký typ procesoru bude programování provedeno.

Device – umožňuje upřesnit procesor z Device Family.

Informační podokno – poskytuje informace o vybraném, nahrávaném .hex souboru. Dále informace o paměti vybraného procesoru a po upřesnění procesoru informace o otevření portu. Rovněž poskytuje informace o tom kdy byl programátor připojen, odpojen a kdy byl vybrán jaký typ čipu. Ve pravém dolním rohu, pod tímto podoknem se nám pod zobrazují barevně informace o připojení programátoru, zapnutí napájení a o práci se sběrnici.

### **3.3. PSoC Creator™**

Je nejmladším programem od společnosti Cypress. Tento program jak již hrubý překlad Creator – „stvořitel“ trochu napovídá o předpokládané funkci a tou je logická tvorba hierarchických návrhů zaměřených převážně na návrhy systému, než nějaké detaily. Pro usnadnění obsahuje Creator plné grafické prostředí a pro usnadnění práce navazuje na starší software s možností vložení některých analogových a digitálních komponent. PSoC Creator je vytvořen pro podporu novějších procesorů a to pro řady PSoC 3 a PSoC 5.

### **3.4. CyClockWizard 1.0**

Jedná se o softwarový nástroj, který umožňuje uživateli přizpůsobit časová řešení, tak aby došlo k vyhovění požadavku zákazníka. Toto řešení se provádí ve 3 snadných krocích:

- Identifikace ideálního řešení pomocí parametrického vyhledávacího nástroje
- Konfigurace pomocí grafického uživatelského rozhraní
- Naprogramování uživatelem

Tato aplikace je dostupná pouze pro programovatelné generátory hodinového signálu uvedené v tabulce:

CY2077	CY22394	CY25701
CY2213	CY22395	CY25702
CY2291	CY22801	CY2X013
CY2292	CY22M1L	CY2X014
CY22050	CY22M1S	CY2XF23
CY22150	CY22U1L	CY2XF24
CY22381	CY22U1S	CY2XF32
CY22392	CY25100	CY2XF33
CY22393	CY25200	CY2XF34

Tab. 4 Seznam programovatelných generátorů hodinového signálu, pro aplikaci ClockWizard 1.0

### 3.5. Ukázkové programy

Pro ukázkové programy byl použit jeden modul ze startovního kitu Cyfi spolu s programátorem dodávaným s tímto kitem. (CY3271 PSoC® firsttouch starter kit. With CyFi™ low-power RF)



Obr. 13 Programátor s možností I2C komunikace



Obr. 14 Použitý kit

Jako ukázka byla zvolena změna barev 3 barevné Led diody, v závislosti na nastavení kapacitním slideru s rozlišením 100 vzorků. Barvy diody se budou měnit dle následujících podmínek:

pozice na slideru  $\leq 33\%$  bude svítit modrá Led dioda

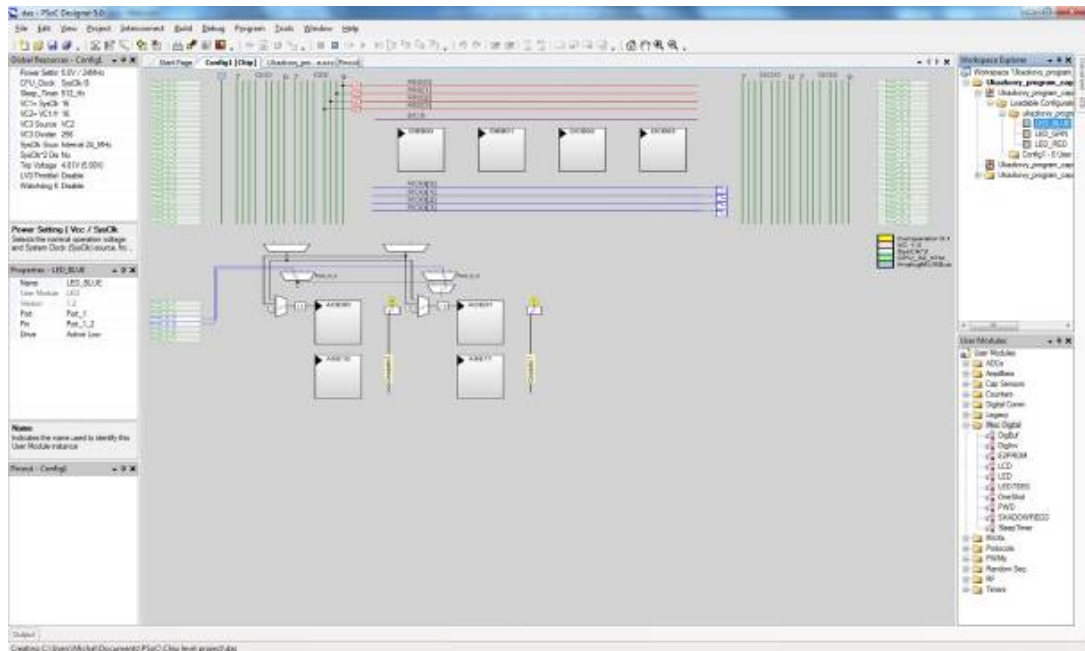
pozice na slideru  $\leq 66\%$  bude svítit zelená Led dioda

pozice na slideru  $\leq 99\%$  bude svítit červená Led dioda.

V ostatních případech nebude svítit ani jedna Led dioda.

### **3.5.1. Chip-level**

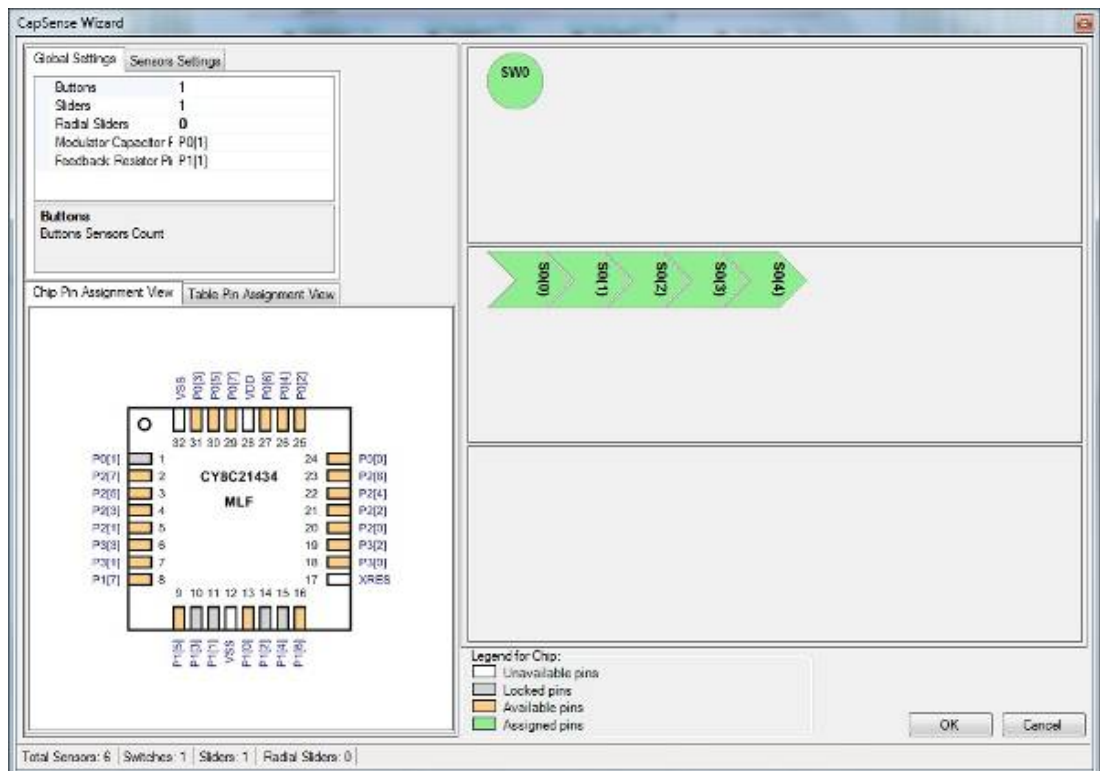
Jak založit nový projekt již bylo řečeno a nyní bude předvedena tvorba nějakého ukázkového programu. Byl zvolen příklad z kapitoly ukázkové programy. Tento projekt budeme založen pro procesor ve zvoleném kitu a tím je procesor CY8C21434-24LFXI a jako programovací jazyk použít jazyk C. Po načtení se ukáže základní obrazovka již s předem připraveným vnitřním uspořádáním, kterou je možné vidět na obr. 15 pracovní okno s potřebnými součástkami. Nyní je vhodné přetahat základní strukturu součástek a připojit je na dané porty dle schématu DPS. Je možné použít například 3 x LED diodu, jelikož program sám o sobě neobsahuje 3 barevnou Led diodu. Led diodu je možné najít v kategorii Misc Digital a je potřeba jí přetáhnout do projektu anebo případně použít volbu Place, která nám jí automaticky vloží. Je potřeba toto udělat 3 krát, abychom dostali 3 potřebné barvy. Ani jedna z diod neobsadí žádný vnitřní blok, ale pouze nějaký port a proto informace o Led diodě je pouze na pravé straně v okně průvodce projektem.



**Obr. 15 Pracovní okno s potřebnými součástkami**

Na každou diodu je nejlépe zvlášť klepnout a provést její přejmenování a nastavení příslušný port a pin podle vnitřního zapojení našeho kitu. A také je potřeba změnit hodnotu Drive na Active High aby byla zapojena klasicky proti zemi.

Nyní z Cap Sensors umístíme kapacitní slider. Ten se usadí do bloků automaticky pomocí volby Place, nebo do předem zvoleného bloku. Následuje jeho přejmenování v podokně properties a následně nastavení jako kapacitní slider s parametry zvoleného na kitu. Toto nastavení se provede kliknutím pravým tlačítkem na slider v podokně průzkumníku a výběrem nabídky CSD Wizard... . Tím se otevře okno, uvedené na obr. 16 nastavení kapacitního slideru.



**Obr. 16** Nastavení kapacitního slideru

V tomto okně se zobrazí výchozí nastavení s možností 1 tlačítka a 1 slideru skládajícího se z 5 dílů to neodpovídá požadované realitě. Je třeba zvolit možnost pouze jeden slider. Po přepnutí do záložky Sensors Settings a klepnutí na slider, který se označí červeně je dána možnost nastavení tohoto slideru. Nyní je zapotřebí nastavit připojení jednotlivých částí slideru na správné porty a piny. Toto nastavení je možné provést přetažením jednotlivých segmentů na jednotlivé piny procesoru a nebo do tabulky, ve které jsou všechny piny možné pro připojení na zvoleném.

Nyní jak zpracovat požadavek chodu do samotného programu Na začátku bylo zvoleno, že kód bude psát pomocí jazyka C.

Před samotným psaním kódu je potřeba nechat nejdříve vygenerovat ovládací knihovny pro jednotlivé části našeho programu. Toto generování ovládacích knihoven se provede stiskem klávesy CTRL+F6. Nyní je možné začít psát samotný kód a to do souboru main.c m který je možné nalézt v okně průzkumníku.

Výsledný kód by měl vypadat takto:



```

//-----
// C main line
//-----
-----

#include <m8c.h>           // part specific constants and macros
#include "PSoC_API.h"     // PSoC API definitions for all User Modules

void main(void)
{
    int pos,neakt;
    M8C_EnableGInt;

    LED_BLUE_Start(); //inicializace LED     BLUE
    LED_GREEN_Start(); //inicializace LED     GREEN
    LED_RED_Start(); //inicializace LED     RED
    CSD_Start(); //inicializace CSD slider
    CSD_InitializeBaselines() ; //scan all sensors first time,
init baseline
    CSD_SetDefaultFingerThresholds() ;

    while (1)
    {
        CSD_ScanAllSensors(); //scan all sensors in array
(buttons and sliders)
        CSD_UpdateAllBaselines(); //Update all baseline levels;

        {

            pos = CSD_wGetCentroidPos(1);
            neakt = CSD_bIsAnySensorActive();

            if (neakt==0)
            {
                LED_BLUE_Off();
                LED_GREEN_Off();
                LED_RED_Off();
            }

            else if (pos<=33)
            {
                LED_BLUE_On();

```

```

        LED_GREEN_Off();
        LED_RED_Off();
    }

    else if (pos<=66)
    {
        LED_BLUE_Off();
        LED_GREEN_On();
        LED_RED_Off();
    }

    else if (pos<=99)
    {
        LED_BLUE_Off();
        LED_GREEN_Off();
        LED_RED_On();
    }

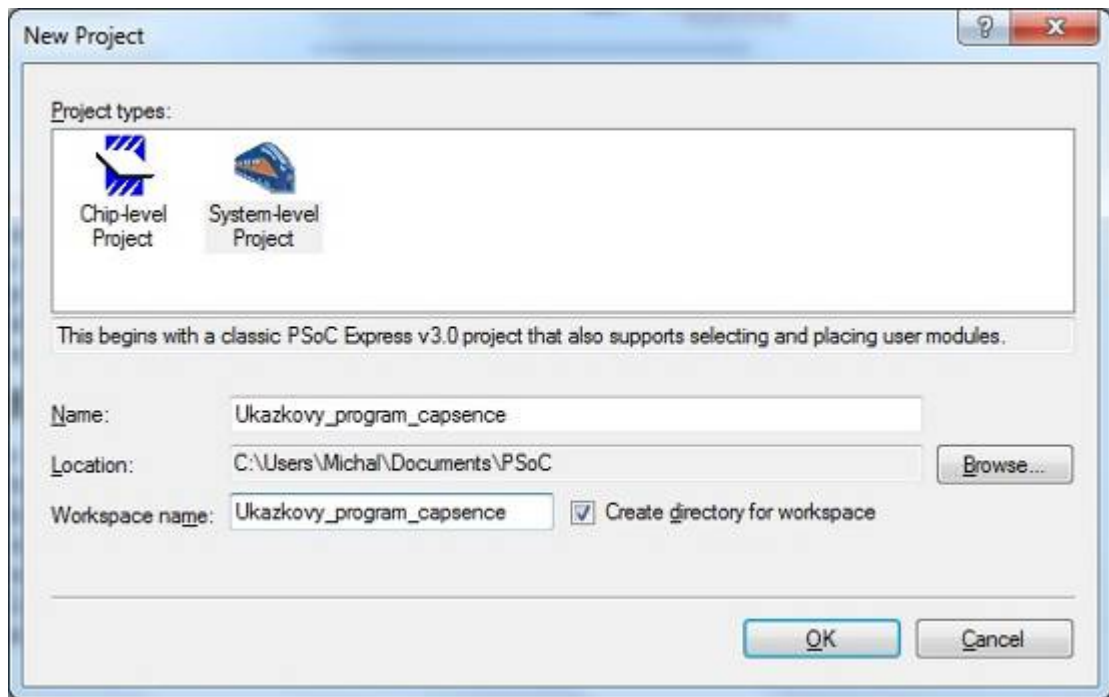
    else if (pos>99)
    {
        LED_BLUE_Off();
        LED_GREEN_Off();
        LED_RED_Off();
    }
}
}
}

```

Po vytvoření je možno výsledný program přeložit stisknutím klávesy F6 a v případě, že nedojde k ohlášení žádné chyby, dát nahrát program do do procesoru a odzkoušet ho v realitě.

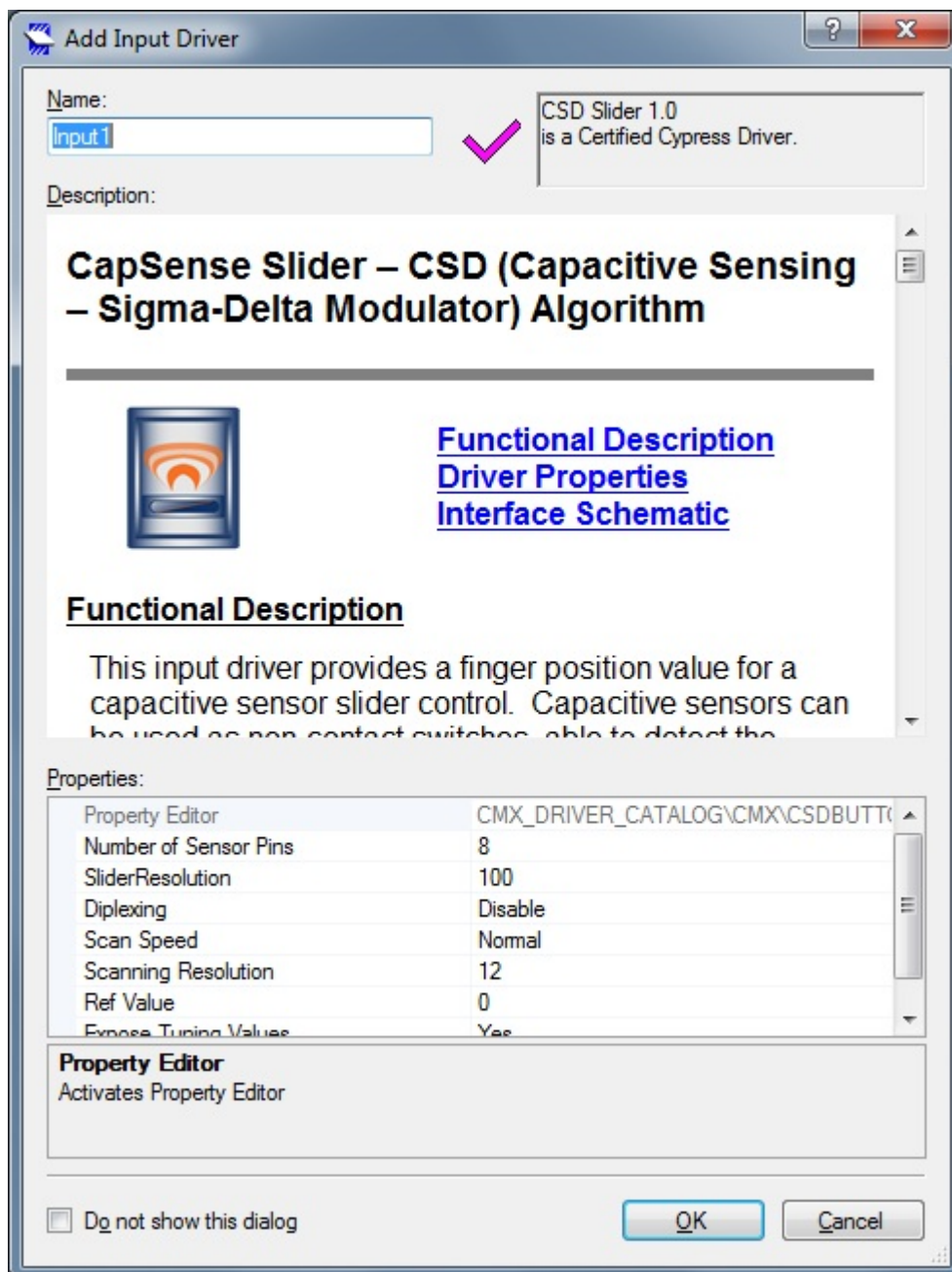
### 3.5.2. System-level

Postup pro založení nového programu byl popsán již dříve. Zvolen byl opět příklad z kapitoly ukázkové programy to pro možnost porovnání v závěrečném zhodnocení. První krok. založme si tedy nový projekt v System-level. Název zvolme tak aby nám v budoucnu co nejjasněji naznačoval o co v projektu jde.



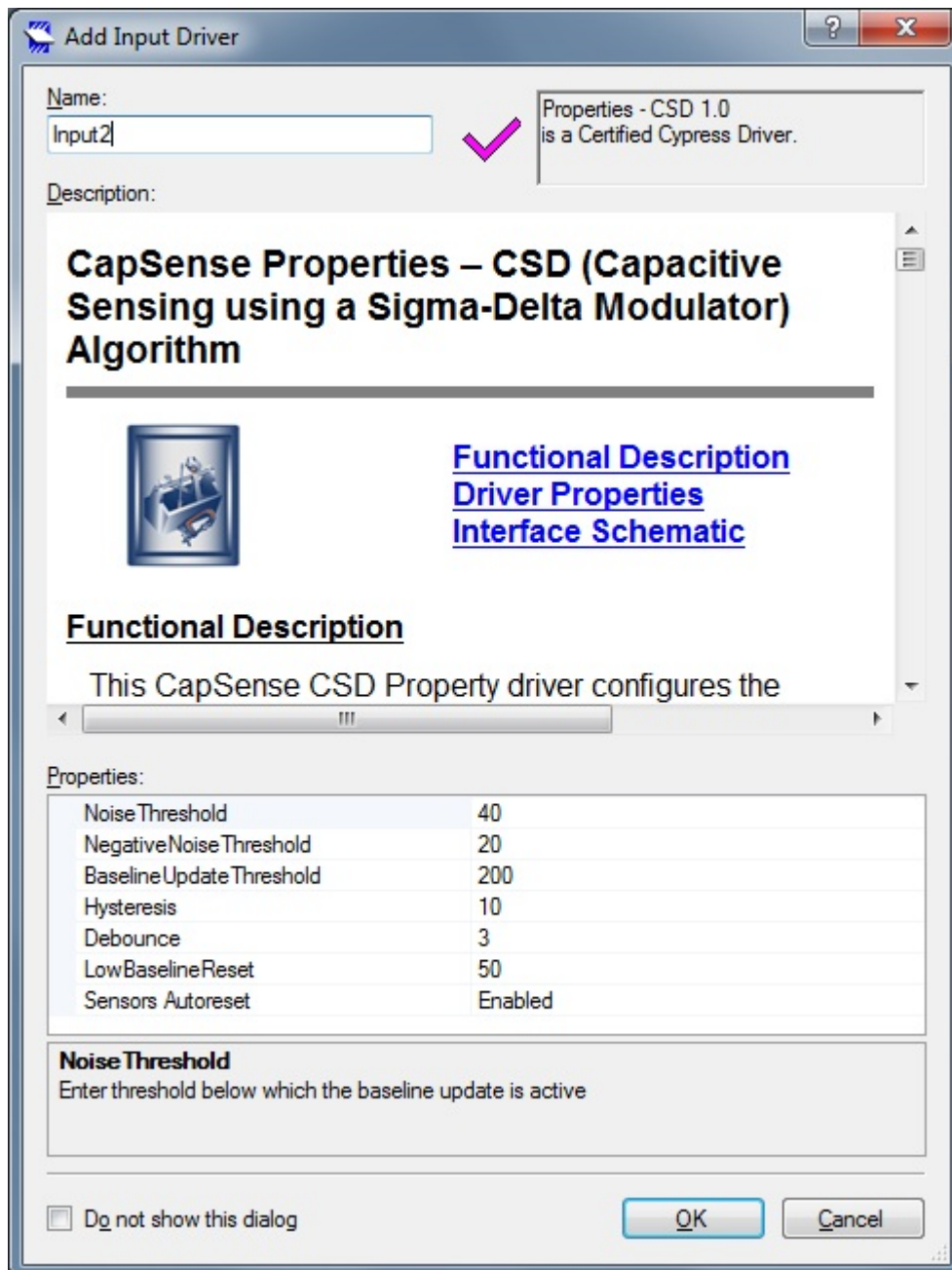
**Obr. 17 Úvodní okno pro výběr typu programu**

Dalším krokem je přetahování jednotlivých potřebných bloků. Při rozboru co bude v programu zapotřebí je jasné, že jako vstupní součástku bude CapSense slider a jako výstupní 3 barevná Led dioda. Z vstupního bloku zvolena z nabídky CapSense – CSD součástka nazvaná Slider – CSD. Tento prvek se přetáhne do hlavního podokna. Ukáže se okno, které je možné vidět na obr. 18 nastavení slideru. V tomto okně je potřeba provést jedinou změnu a tou je počet snímacích pinů, kterých je na použitém kitu 7. Dále je užitečné provést přejmenování součástky, což není samozřejmě nutností. Pro tuto ukázkou je zadán název Slider. Samozřejmě všechny změny je možné provést i později v podokně vlastností. Změny je nutno potvrdit tlačítkem OK. Po potvrzení za tímto oknem vyskočí další okno pro nastavení kapacitních snímačů. Toto okno je znázorněno na obr. 19 nastavení kapacitních snímačů.

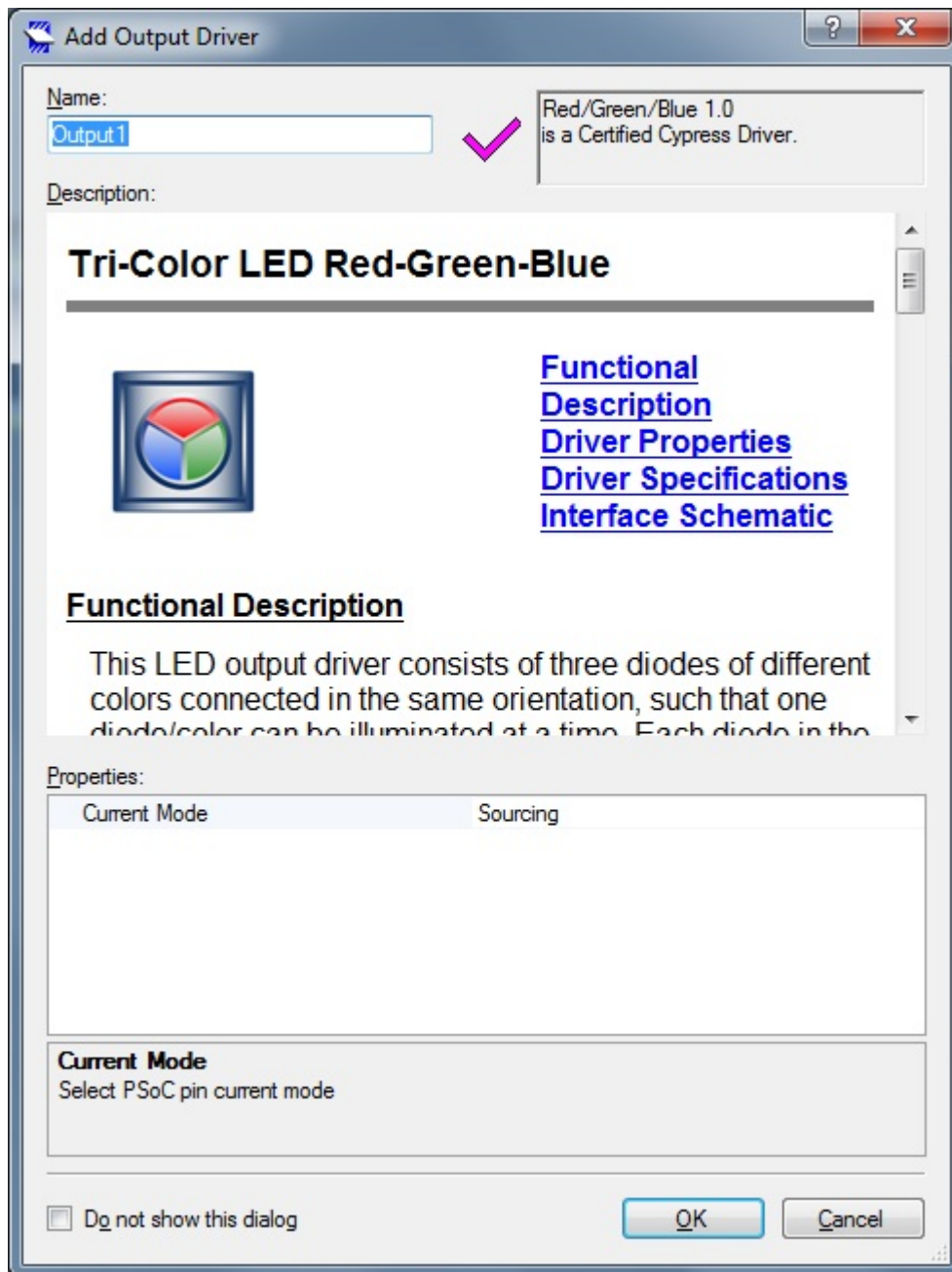


**Obr. 18** Nastavení slideru

V tomto okně není nic potřeba nic podstatného měnit, pouze provést přejmenování na prvku na Propertis a vypnout automatické resetování senzoru. Posledním požadovaným prvkem a to z výstupního bloku je 3 barevná Led dioda. Ta se nachází v knihovně Display-LED a jedná se o Tri-Color led. Okno, které se následně ukáže je dobré přejmenovat na LED a je znázorněno na obr. 20 tříbarevná dioda.



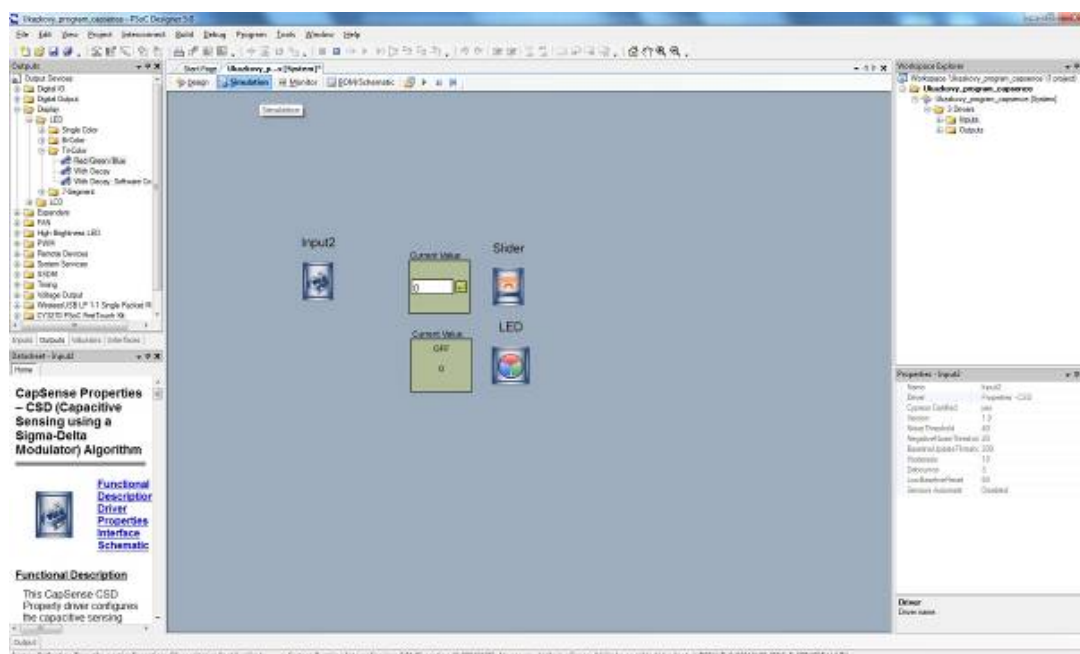
Obr. 19 Nastavení kapacitních snímačů



Obr. 20 Tříbarevná dioda

Následně je možno rozbalit okno průzkumníku, kde je možné vidět, že došlo ke změně v složce programu a to v přírůstku složek ve kterých se nám nachází informace o tom, že jsou používány 3 ovladače 2 vstupní a 1 výstupní. Celý příklad se dá velice snadno modifikovat a to nahrazením jedné součástky za druhou. Toto nahrazení se provede při kliknutí na součástku, kterou chceme změnit pravým tlačítkem a vybráním volby Replace a nebo je možné součástku smazat a vložit jinou.

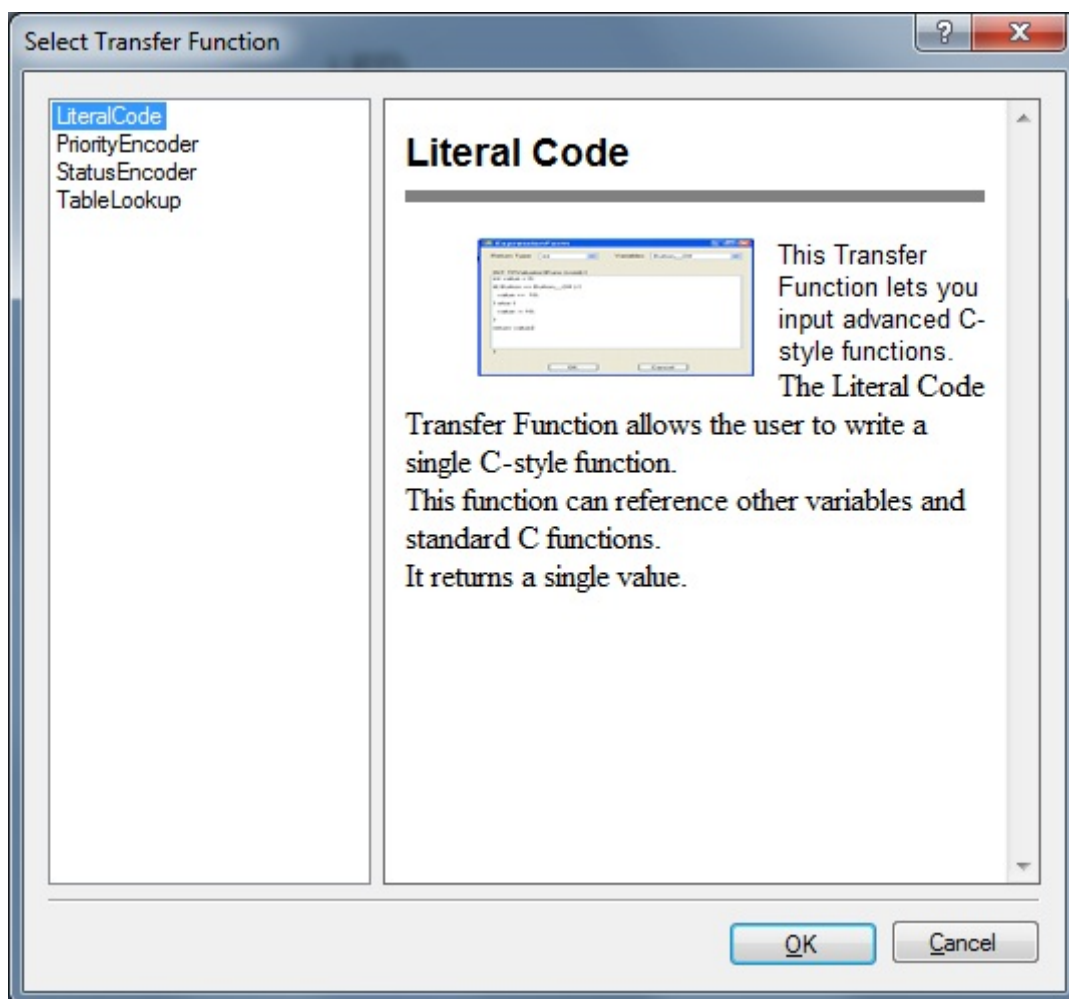
Fázi simulace zahájíme klepnutím na záložku Simulace. Nyní je možné se podívat do podokna simulace, které vidíme na obr. 21 okno simulace.



**Obr. 21 Okno simulace**

Prostým dosazením hodnoty nedojde k žádné změně. Je potřeba nastavit podmínky. A to přepnutím do okna záložky Design a klepnutím pravým tlačítkem na součástku, která se bude měnit. Dle zadání se bude měnit 3 barevná LED dioda. Při kliknutí je potřeba vybrat volbu transfer function. Otevře se okno, znázorněné na obr. 22 transfer function. Pro samotnou tvorbu podmínek. Například Priority Encoder. Jedná se o klasickou podmínku if, kterou určitě není třeba představovat. A volba se potvrdí tlačítkem OK. Pak je již možné vyplnit samotné podmínky co se má dít při určité kapacitě a to dle zadání úkolu.

Po zadání všech podmínek je třeba je opět potvrdit tlačítkem OK. Nyní je možné si všimnout propojení Slideru a LED virtuálním oranžovým vodičem. Nastavené podmínky lze opět stejným způsobem změnit.

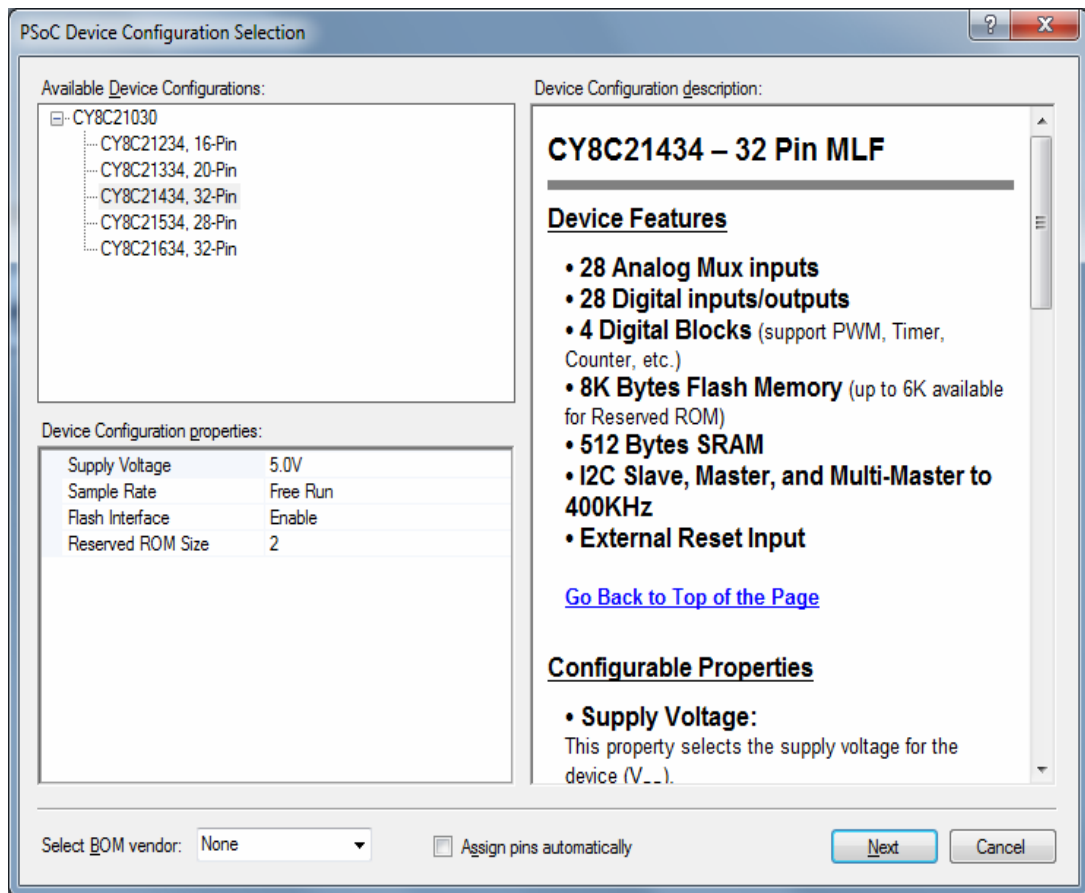


**Obr. 22 Transfer Function**

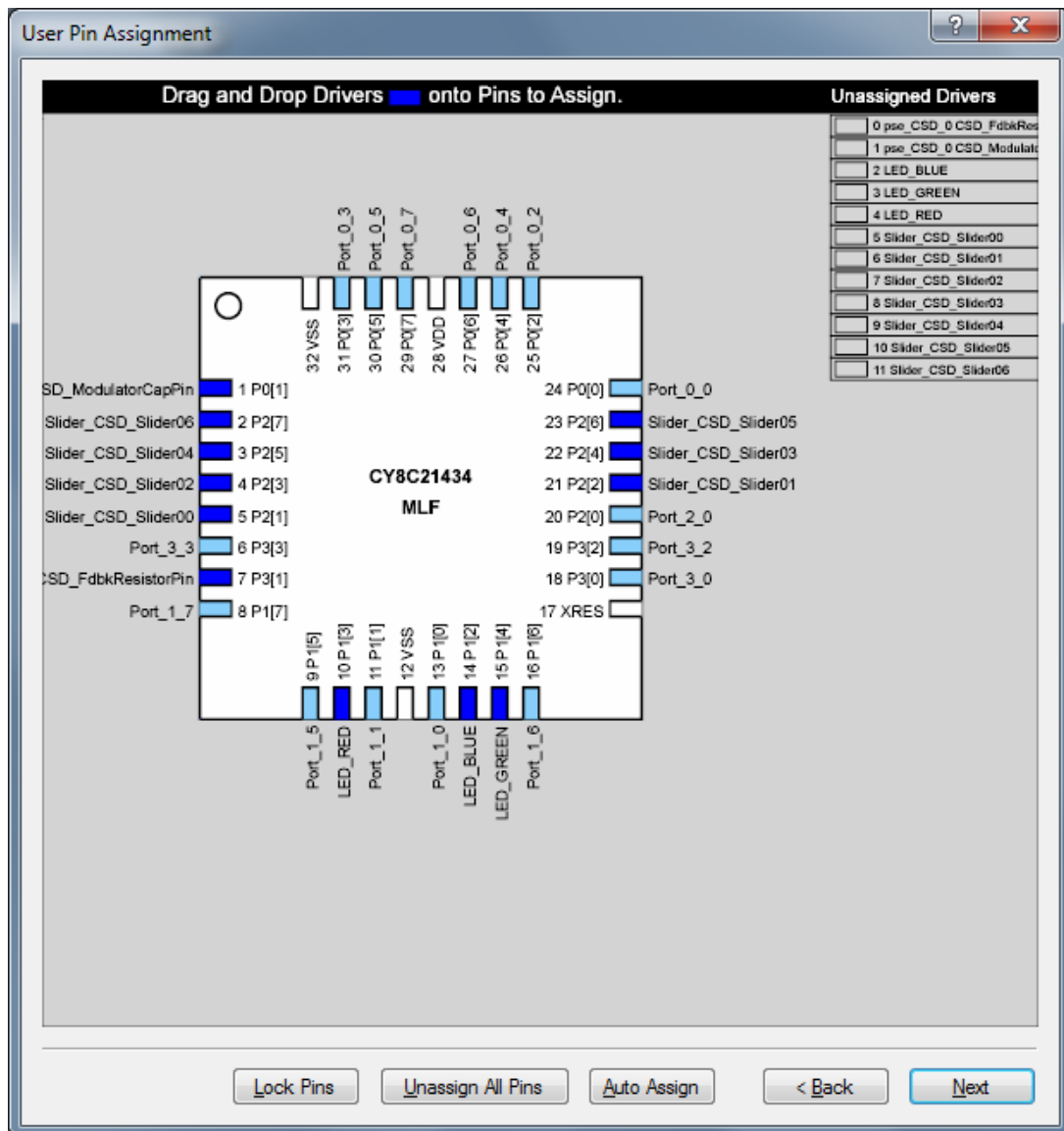
Nyní už záložka Simulace bude chodit tak jak je zadáno pokud došlo ke správnému nastavení. Jestliže ano, je možné tedy přistoupit k další části a tou je přeložení programu pro zvolený procesor. Přeložení se provádí klávesou F6 případně nabídkou v Build. Po stisknutí přeložení program nabídne procesory, které zvládnou požadovanou aplikaci. Je potřeba vybrat procesor, který se nachází na kitu tedy CY8C21434, 32-Pin. Nabídka znázorněna na obr. 23 výběr procesoru. Po potvrzení otevře okno s obrázkem procesoru a s jeho piny, občas program provede automatické připojení součástí na jednotlivé porty. Toto zapojení je možné si ovšem přetáhnout dle potřeby. Toto okno je znázorněno na obr. 24 připojení prvků na jednotlivé porty. Jelikož se používá již určité zapojení je dobré použít



volbu Unassing All pins a provést naklikání jednotlivých součástek na piny jak je třeba dle schématu a stisknout tlačítko next.



Obr. 23 Výběr procesoru



**Obr. 24 Připojení prvků na jednotlivé porty**

Pak proběhne překlad samotného programu, po kterém v okně output vyskočí, že nedošlo k žádným chybám, ani k žádnému varování a v hlavním okně je možné vidět zapojený procesor a pod ním je možné vygenerovat datasheet schéma zapojení.... . Program je v pořádku vytvořen a je tedy potřeba jej nahrát do programátoru. K tomu se nachází v nabídce program nabídka program part, díky níž se nám spustí zjednodušená verze programátoru. Po připojení programátoru s kitem, na kterém se bude program používat, proběhne jeho automatické nadetekování. Jako poslední krok je již potřeba vybrat možnost programování. To se provádí stiskem zelené šipkou viz obr. 25 programování. Samozřejmě chod závisí i částečně na zvoleném programátoru. Některé nejsou automaticky napájené a tak po

naprogramování je potřeba je pro chod je zapnout. Ale zvolený kit je automaticky napájen.



**Obr. 25 Programování**

V případě, že dojde k nahrání bez potíží je možné provést závěrečné vyzkoušení již v realitě a případně provést změnu podmínek.

Nutno poznamenat, že nahrání je možné provést i přes program PSoC Programmer™.

### **3.6. Závěrečné shrnutí**

Programování bylo z důvodu dostupnosti a vývoje programového prostředí provedeno pouze pro architekturu PSoC1. Během programování bylo zjištěno, že výsledný program se ukládá na 2 místa do složky projektu a to jak do samotné složky a také do složky output. Toto ukládání probíhá tak, že program přeložený bez chyb se nejprve uloží do složky output a následně proběhne překopírování do složky programu. V případě, že se provede změnu v programu a výsledné přeložení bude s chybami .hex soubor se ve složce output smaže a ve složce programu zůstane poslední v pořádku přeložená verze. A proto je nejlepší používat vždy poslední verzi ze složky output.

I přestože se jedná relativně o jednodušší zkušební program je možné vidět rozdíl mezi programováním v Chip-level Project a System-level Project. Hlavní rozdíl spočívá v náročnosti psaní kódu v Chip levelu oproti System levelu. A proto Chip level je jednodušší a pro lidi, s menšími znalostmi vhodnější a to v některých případech předčí i nevýhodu omezených možností v programování. Mezi další výhody je možnost volby vybrání procesoru po

vytvoření programu, což může ve výsledku umožnit výběr jiného a levnějšího procesoru než s jakým uživatel v počátku počítal.

Pro výsledné nahrání programu do procesoru bylo shledáno, že je bohatě dostačující použití zjednodušené verze programátoru přímo v programu, kdy není nutno starat se o vybírání nahrávaného programu. Samozřejmě někdy může být lepší využít externí program a to z důvodu delší doby náběhu vývojového programu.

## 4. Závěr

Na základě studia dostupné literatury , zvláště od výrobce Cypress Semiconductor Corporation a jejím zhodnocení byl zpracována a v této práci prezentován soubor informací o mikroprocesoru

### **PSoC** – Programmable Systém – on – Chip.

Tento soubor zahrnuje :

- stručný přehled o podmínkách a důvodech vzniku tohoto procesoru ,
- popis jeho vnitřní struktury a možnosti užití jednotlivých dílů
- přehled nástrojů programování pro různé druhy praktického použití
- informace o dostupnosti a dosažitelném sortimentu
- rady pro začínající uživatele
- přehled a zhodnocení nástrojů pro programové nastavení procesoru.

Na základě získaných poznatků byl zpracován metodický postup pro řešení praktického užití tohoto typu mikroprocesoru . Postup je demonstrován na řešení jednoduché aplikaci a to –

světelné indikaci dosažení tří volitelných úrovní na kapacitním slideru v rozsahu 0 až100 %

Postup byl zpracován ve dvou variantách užitím dvou částí programu PSoC Designer™ a těmi jsou Chip-level a Systém - level . Užití těchto nástrojů umožňuje nejen vlastní řešení zadané úlohy

(určení příslušných bloku , způsobu spojení se systémem pinů aj.) , ale zkontroluje vhodnost použití uvažovaného procesoru eventuelně navrhne lepší variantu a pomocí simulace prověří správnost navrženého propojení.

Použití dvou programovacích nástrojů umožnilo jejich vzájemné posouzení a poskytnout dalším potencionálním uživatelům podklad pro správnou volbu.

Lze proto závěrem opodstatněně konstatovat , že bylo splněno zadání prezentované práce a s mírným optimismem považovat práci také za malý příspěvek ve snahách a větší rozšíření tohoto univerzálního prvku , jakým **PSoC** bezesporu je.

## Použitá literatura

Cypress Semiconductor [online]. Cypress Semiconductor Corporation, c2008 [cit. 2009-11-20]. Kódováno ve Unicode (UTF-8). Text v angličtině. Dostupný z WWW: <<http://www.cypress.com/>>.

PÍČ, Vlastimil. Citlivé měření kapacity s mikrokontrolery Cypress PSoC. HW.cz : vše o elektronice a programování [online]. 23. duben 2008, [cit. 2009-11-20]. Dostupný z WWW: <<http://hw.cz/firemni-clanky/msc-vertriebs-cz/art2323-mikrokontrolery-cypress-psoc-v-kapacitnich-senzorech-klavesn>>.

PSoC™ Embedded Microcontrollers pro analogové i digitální zpracování signálů. HW.cz : vše o elektronice a programování [online]. 29. srpen 2005, [cit. 2009-11-20]. Dostupný z WWW: <<http://hw.cz/produkty/nove-soucastky/art1369-psoc-embedded-microcontrollers-pro-analogove-i-digitalni-zpracovani->>.

MIČAKOVIĆ, Predrag . Architecture and Programming of PSoC Microcontrollers [online]. Belgrade : MikroElektronika, c1998–2010 [cit. 2009-11-20]. Dostupné z WWW: <<http://www.easypsoc.com/book/index.html>>.

Cypress Semiconductor Corporation. PSoC® Mixed-Signal Array : Technical Reference Manual (TRM) [online]. San Jose : Cypress Semiconductor, c2005 - 2006 [cit. 2009-11-20]. Dostupné z WWW: <[http://download.cypress.com.edgesuite.net/design\\_resources/more\\_resources/contents/psoc\\_r\\_\\_technical\\_reference\\_manual\\_\\_trm\\_\\_14.pdf](http://download.cypress.com.edgesuite.net/design_resources/more_resources/contents/psoc_r__technical_reference_manual__trm__14.pdf)> .

## Údaje pro knihovnickou databázi

Název práce	Studium možností využití SOC systémů
Autor práce	Michal Rokyta
Obor	Komunikační a mikroprocesorová technika
Rok obhajoby	2010
Vedoucí práce	Ing. Radim Vondra, Ph.D.
Anotace	Tato práce se zabývá mikroprocesory firmy Cypress PSoC. Její praktická část se zabývá demonstrací na jednoduché ukázkové aplikaci, vytvořené pomocí vývojového prostředí PSoC Designer 5.0.
Klíčová slova	PSoC, SOC, Designer 5.0.