

**ZÁPADOČESKÁ UNIVERZITA V PLZNI  
FAKULTA ELEKTROTECHNICKÁ**

**Katedra aplikované elektroniky**

**BAKALÁŘSKÁ PRÁCE**

**Mikrokontroléry s jádrem ARM**

**vedoucí práce: Petr Krist  
autor: Karel Hodic**

**2013**



ZÁPADOČESKÁ UNIVERZITA V PLZNI  
Fakulta elektrotechnická  
Akademický rok: 2012/2013

**ZADÁNÍ BAKALÁŘSKÉ PRÁCE**  
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Karel HODIC**  
Osobní číslo: **E10B0588P**  
Studijní program: **B2612 Elektrotechnika a informatika**  
Studijní obor: **Elektronika a telekomunikace**  
Název tématu: **Mikrokontroléry s jádrem ARM**  
Zadávající katedra: **Katedra aplikované elektroniky a telekomunikací**

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

1. Stručně popište architekturu a funkci jednotlivých jader ARM7, ARM9 a Cortex a přehledně shrňte jejich odlišnosti a charakterizujte jejich jednotlivé verze a rozšíření. Porovnejte jednotlivé použité instrukční módy a s nimi související výpočetní výkon a kódovou hustotu. Posuďte možnosti jednotlivých verzí jader z hlediska časového determinismu.
2. Vypracujte ucelený přehled mikrokontrolérů s jádrem ARM dostupných na trhu a proveďte srovnání jejich vlastností z hlediska výkonu, velikosti programové a datové paměti, systémových periférií, možné rozšiřitelnosti externího hardware a konektivity s ostatními systémy.
3. Proveďte detailní srovnání vlastností možných volně dostupných i komerčně dodávaných vývojových nástrojů - tzn. prostředků pro generování kódů, vývojových prostředí a emulátorů.

Rozsah grafických prací: **podle doporučení vedoucího**

Rozsah pracovní zprávy: **20 - 30 stran**

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

Seznam odborné literatury:

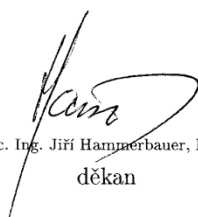
- 1. ARM Architecture Reference Manual - ARM DDI 0100E**
- 2. ARM Software Development Kit Version 2.50 - Reference Guide - ARM DUI 0041C**

Vedoucí bakalářské práce: **Ing. Petr Krist, Ph.D.**  
Katedra aplikované elektroniky a telekomunikací

Konzultant bakalářské práce: **Ing. Petr Krist, Ph.D.**  
Katedra aplikované elektroniky a telekomunikací


Datum zadání bakalářské práce: **15. října 2012**

Termín odevzdání bakalářské práce: **7. června 2013**



Doc. Ing. Jiří Hammerbauer, Ph.D.  
děkan

L.S.



Doc. Dr. Ing. Viatcheslav Georgiev  
vedoucí katedry

V Plzni dne 15. října 2012

**Anotace**

Bakalářská práce je zaměřena na ARM mikrokontroléry a popisuje stručně jejich vlastnosti. Dále porovnává parametry dostupných ARM mikrokontrolérů na trhu mezi jednotlivými výrobci do přehledné tabulky. Potom jsou porovnávána dostupná vývojová prostředí a je změřena jejich výsledná velikost kódu a rychlost testovacím kódu.

**Klíčová slova**

Mikrokontrolér, Vývojové prostředí, Debugger

**Abstract**

The bachelor thesis is focused to ARM microcontrollers and succinctly describes their properties. Further compares parameters available ARM microcontrollers in the market between individual producers in tabular form. Then are they compared the available development environments and measured the resulting code size and speed test code.

**Key words**

Microcontroller, Development solution, Debugger

## Prohlášení

Předkládám tímto k posouzení a obhajobě bakalářskou práci, zpracovanou na závěr studia na Fakultě elektrotechnické Západočeské univerzity v Plzni.

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této bakalářské práce.

Dále prohlašuji, že veškerý software, použitý při řešení této bakalářské práce, je legální.

V Plzni dne 6.6.2013

Karel Hodic

.....

# 1 Obsah

Úvod.....	11
Seznam zkratk a symbolů .....	12
2 Mikrokontroléry ARM.....	13
2.1 Popis architektur .....	13
2.1.1 Rodina ARM7 .....	13
2.1.2 Rodina ARM9 .....	13
2.1.3 Rodina Cortex .....	14
2.1.4 Porovnání ARM rodin.....	16
2.2 Instrukční sady.....	16
2.2.1 Instrukční sada ARM.....	16
2.2.2 Instrukční sada Thumb .....	17
2.2.3 Instrukční sada Thumb 2 .....	17
2.2.4 Srovnání jednotlivých instrukčních sad.....	18
3 Přehled dostupných ARM mikrokontrolérů.....	18
3.1 STMicroelectronic.....	18
3.1.1 Řada STM32F0.....	18
3.1.2 Řada STM32F1.....	19
3.1.3 Řada STM32F2.....	20
3.1.4 Řada STM32F3.....	21
3.1.5 Řada STM32F4.....	22
3.1.6 Řada STM32L1.....	23
3.1.7 Řada STM32W.....	24
3.2 Atmel.....	25
3.2.1 Řada SAM3A/X.....	25
3.2.2 Řada SAM3N.....	26
3.2.3 Řada SAM3S.....	26
3.2.4 Řada SAM3U.....	27
3.2.5 Řada SAM4L .....	27
3.2.6 Řada SAM4E.....	28
3.2.7 Řada SAM4S.....	28
3.2.8 Řada SAM7S/SE .....	29
3.2.9 Řada SAM7X/XC.....	29
3.2.10 Řada SAM7L .....	30



3.2.11	Řada SAM9XE.....	30
3.3	Texas Instruments.....	31
3.3.1	Řada TM4 Tiva C.....	31
3.3.2	Řada TMS470 Hercules.....	31
3.4	NXP.....	31
3.4.1	Řada LPC800.....	32
3.4.2	Řada LPC11xx.....	32
3.4.3	Řada LPC12xx.....	32
3.4.4	Řada LPC13xx.....	33
3.4.5	Řada LPC17xx.....	33
3.4.6	Řada LPC18xx.....	34
3.4.7	Řada LPC40xx.....	34
3.4.8	Řada LPC43xx.....	35
3.5	Analog Devices.....	36
3.5.1	Řady ADuC7000.....	36
3.5.2	Řady ADuCM.....	36
3.6	Cypress.....	36
3.6.1	Řada PSoC 4000.....	36
3.6.2	Řada PSoC 5000LP.....	37
3.7	Energy micro.....	37
3.7.1	Řada EFM32ZG - Zero Gecko.....	37
3.7.2	Řada EFM32TG - Tiny Gecko.....	38
3.7.3	Řada EFM32G – Gecko.....	38
3.7.4	Řada EFM32LG - Leopard Gecko.....	39
3.7.5	Řada EFM32GG - Giant Gecko.....	39
3.7.6	Řada EFM32WG - Wonder Gecko.....	40
3.8	Freescale.....	40
3.8.1	Řada Kinetis K.....	40
3.8.2	Řada Kinetis KL.....	43
3.9	Nuvoton.....	44
3.9.1	Řada NUC 100 Advanced Line.....	44
3.9.2	Řada NUC 110 LCD.....	44
3.9.3	Řada NUC 120 USB Connectivity.....	44
3.9.4	Řada NUC 130 Advanced.....	45

3.9.5	Řada NUC 140.....	45
3.9.6	Řady NUC 1XXK.....	45
3.10	Holtek.....	45
3.11	Srovnání.....	46
4	Dostupná vývojová prostředí .....	52
4.1	Keil uVision4.....	52
4.2	Coocox .....	52
4.3	AVR Studio 6.0.....	54
4.4	Raisonance.....	54
4.5	Atollic .....	54
4.6	IAR.....	55
4.7	Možnosti vývoje na Linuxu .....	55
4.7.1	CodeSoucery.....	55
4.7.2	ST-Link pro Linux.....	55
4.8	Srovnání vývojových prostředí.....	56
4.8.1	Přehledné tabulky velikosti zkompilevaného kódu.....	56
4.8.2	Přehledné tabulky rychlosti kódu.....	58
5	Závěr.....	60
6	Použitá literatura.....	61
7	Seznam obrázků.....	62
8	Seznam tabulek.....	62
9	Seznam použitého softwaru.....	63
10	Obsah příloženého DVD.....	64
11	Přílohy.....	65

## Úvod

Předkládaná práce je zaměřena na vytvoření materiálu, který poslouží každému, který chce začít pracovat s mikrokontroléry ARM.

ARM mikrokontroléry naleznou uplatnění v řadě aplikací, kde bez problému překonají ve vlastnostech dost používané 8 bitové mikrokontroléry svým výkonem, bohatou periferní výbavou, spotřebou a poslední dobou také i cenou.

Bakalářská práce je tvořena třemi částmi: první je samotný úvod do ARM mikrokontrolérů a rozdíly mezi rodinami, druhá část obsahuje samotný seznam dostupných ARM mikrokontrolérů a porovnání jejich parametrů, třetí část obsahuje seznam komerčních i volně dostupných vývojových prostředí pro mikrokontroléry ARM.

## Seznam zkratk a symbolů

ADC	Analog-to-digital converter	analogově-digitální převodník
AES	Advanced Encryption Standart	symetrická bloková šifra
CAN	Controller Area Network	multiplexní sériová sběrnice
CRC	Cyclic redundancy Check	cyklický redundantní součet
DAC	Digital-to-analog converter	digitálně analogový převodník
DES	Data Encryption Standart	symetrická šifra
DMA	Direct memory access	přímý přístup do paměti
ECC	Error Checking and Correnting	kontrola a oprava chyb
EEPROM	Electrically Erasable Programmable Read-Only Memory	elektricky mazatelná paměť
Flash	Flash Memory	programovatelná paměť
FS	Full Speed	12Mbit u USB
HDMI-CEC	High-Definition Multimedia Interface – Consumer Electronics Control	obousměrná sběrnice pro řízení připojených zařízení
I <sup>2</sup> C	Inter-Integrated Circuit	dvouvodičová sběrnice
IAP	programování v aplikaci	
JTAG	Joint Test Action Group	ladicí a programovací rozhraní
OTG	On-The-Go	umožňuje přepnout USB do Master režimu
MD5	Message-digest algorithm	hashovací algoritmus
PWM	Pulse-width modulation	šířkově pulzní modulace
RTC	Real-time clock	hodiny reálného času
SAI	Seriál audio interface	sériové rozhraní pro audio
SDIO	Secure Digital Input Output	sériové rozhraní používané např. u SD karet
SHA	Secure Hash Algorithm	zabezpečený hash algoritmus
SPI	Seriál Peripheral Interface	sériové periferní rozhraní
SRAM	Static Random Access Memory	statická paměť s náhodným přístupem
SWD	Seriál Wire Debug	dvouvodičové sériové rozhraní pro ladění a programování
TripleDes	Triple Data Encryption Algorithm	bloková šifra
TWI	Two Wire Interface	dvouvodičová sběrnice
UART	Universal asynchronous Receiver andTrasnmmitter	asynchronní sériový kanál
USART	Universal Synchronous Receiver and Trasnmmitter	synchronní sériový kanál
USB	Universal Serial Bus	univerzální sériová sběrnice
WDG	Watchdog	hlídací časovač

## 2 Mikrokontroléry ARM

### 2.1 Popis architektury

Mikropočítače s jádrem ARM prodělaly ve svém vývoji několik rodin. Zaměříme se na rodiny, které jsou dnes běžně používány, na ARM7, ARM9 a nejnovější Cortex. Obsahují 32 bitové RISC jádro s přibližně 35000 tranzistorů (ARM7TDMI), s nižší spotřebou jádra. V blízké budoucnosti bude vyráběna 64 bitová varianta, která bude zpětně kompatibilní se starší verzí ARM jader.

#### 2.1.1 Rodina ARM7

Tato rodina je založena na Von Neumannově architektuře, kde program a data jsou přenášeny po společné, v našem případě po 32-bitové sběrnici. Tvořily jí rodiny řady ARM7TDMI a ARM7TDMI-S, které byly využívány v mikrokontrolérech, ale i další, které jsou využívány v zařízení, jako jsou routery, mobilní telefony, přenosné PC apod. Obsahuje tří-úrovňovou pipeline (Fetch, Decode, Execute), která vykonávání instrukce rozděljuje do tří kroků zpracování, které se překrývají a je možné tyto kroky provádět paralelně, což ve výsledku zvýší výkon vykonávání instrukcí. V jádru nalezneme celkem 31 univerzálních registru se šířkou 32bit (v každém módu R0 až R15) a 6 status registrů, které se podle módu procesoru překrývají nebo obsahují svoji vlastní banku viz *Obr. 1*. Módů nalezneme 7: System and User, FIQ, Supervisor, Abort, IRQ, Undefined. Jádro také podporuje Thumb instrukce, ale při jejich použití se nám omezí celkový počet univerzálních 32bit registrů na 21 (nejsou využity registry R8 až R12) a počet status registrů bude stejný jako v ARM módu, které se podle režimu překrývají nebo jsou ve vlastní bance. Zpoždění před vykonáváním přerušení je až 29 procesorových cyklů, zaregistrování přerušení od FIQ nebo IRQ je zpoždění na hodnotě minimálně 5 procesorových cyklů.

#### 2.1.2 Rodina ARM9

Tato rodina byla přepracována na Harvard architekturu (data a instrukce nejsou přenášeny po stejné sběrnici) a má pěti-úrovňovou pipeline (Fetch, Decode, Execute, Memory, Write). To umožňovalo zvyšovat taktovací frekvence a výkon na vyšší hodnotu než u rodiny ARM7. Z této rodiny je pro mikrokontroléry využíván hlavně ARM926EJ-S. Obsahuje MMU (jednotka, která má na starosti ochranu paměti před zápisem např. od jiného procesu, virtualizace paměti a ochrana registrů v uživatelském režimu) a jednotku Jazelle (pro akceleraci java byte kódu). Stejně jako ARM7TDMI, tak i toto jádro využívá instrukční sady ARM a Thumb. Má také stejné registry jako ARM7TDMI, které mohou být rozšířeny o jednotku FPU (jednotka hardwarově zpracovávající operace s pohyblivou desetinou čárkou). V tomto jádře nalezneme Cache (vyrovnávací paměť) pro data a instrukce o velikosti od 4KB do 128KB, sloužící pro zvýšení výkonu.

Obecné registry					
System and user	FIQ	Supervisor	Abort	IRQ	Undefined
R0	R0	R0	R0	R0	R0
R1	R1	R1	R1	R1	R1
R2	R2	R2	R2	R2	R2
R3	R3	R3	R3	R3	R3
R4	R4	R4	R4	R4	R4
R5	R5	R5	R5	R5	R5
R6	R6	R6	R6	R6	R6
R7	R7	R7	R7	R7	R7
R8	R8_fiq *	R8	R8	R8	R8
R9	R9_fiq *	R9	R9	R9	R9
R10	R10_fiq *	R10	R10	R10	R10
R11	R11_fiq *	R11	R11	R11	R11
R12	R12_fiq *	R12	R12	R12	R12
R13	R13_fiq *	R13_svc *	R13_abt *	R13_irq *	R13_und *
R14	R14_fiq *	R14_svc *	R14_abt *	R14_irq *	R14_und *
R15(PC)	R15(PC)	R15(PC)	R15(PC)	R15(PC)	R15(PC)
Status registry					
CPSR	CPSR	CPSR	CPSR	CPSR	CPSR
	SPSR_fiq *	SPSR_svc *	SPSR_abt *	SPSR_irq *	SPSR_und *
* = registrová banka					

Obr. 1 Registry v jednotlivých módech na ARM7TDMI

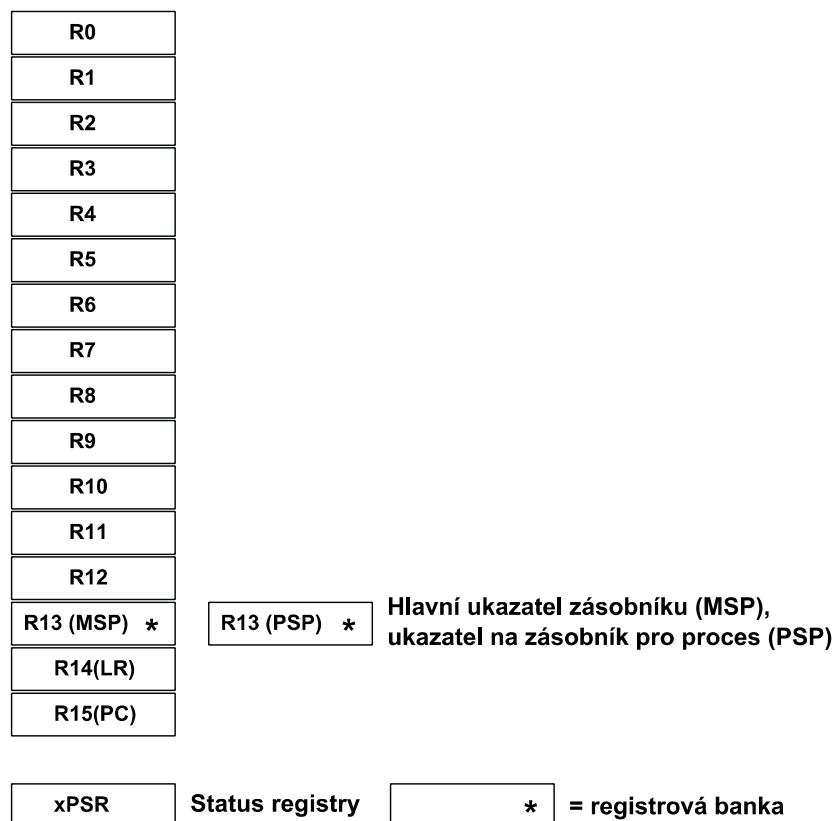
### 2.1.3 Rodina Cortex

Cortex je nejnovější rodina ARM procesoru. Existuje v profilech Cortex-A, Cortex-R a Cortex-M, které se pak dále rozdělují na menší. Všechna tato jádra podporují instrukční sadu Thumb a Thumb 2.

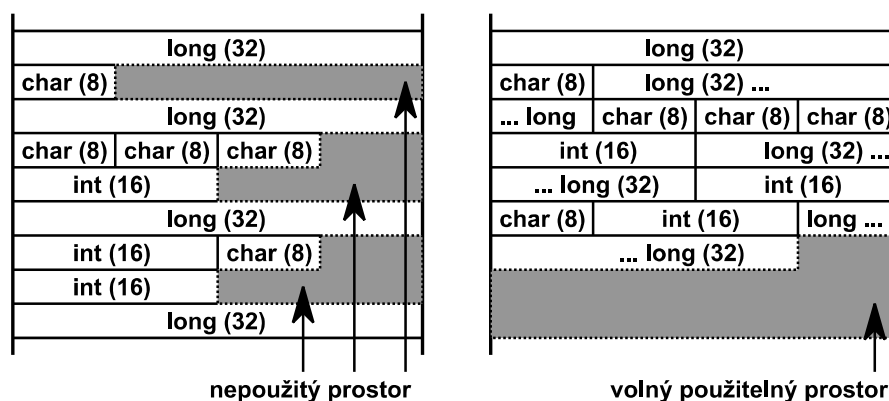
ARM řady Cortex-M podporují pouze Thumb a Thmub 2 instrukce. Jsou za to optimalizovány na nízkou spotřebu a výrobní cenu. Tato řada je určena do jednočipových mikropočítačů, kde vlastnostmi konkuruje tradičním 8-bit mikrokontrolérům (zejména řada Cortex-M0), ale jejich výkon je mnohonásobně větší, dosahují větší kódové hustoty. Obsahuje celkem 17 32-bit univerzálních registrů, status registry a další viz. Obr 2. Podle toho, v jakém jsme v módu (uživatelský nebo privilegovaný) se přepíná registr s ukazatelem na zásobník, to znamená, že v privilegovém režimu máme jiný zásobník než v uživatelském. Jádra Cortex, na rozdíl od předešlých rodin, umožňují přístup k nezarovnaným proměnným (umístěné na jiné adrese než násobky 4), efektivněji se využije paměť pro data, rozdíl je na Obr. 3. Všechna přerušení mají zpoždění před začátkem vykonávání 12 hodinových taktu a 12 hodinových taktu trvá opuštění přerušení. Pokud nastanou dvě nebo více přerušení zároveň, je zapotřebí mezi dalším vykonáváním přerušením 6 hodinových taktů. Pokud nenastanou zároveň, do dalšího vykonávání přerušení je třeba 7 až 18 hodinových taktů. Tato přerušení mohou být

maskována, mohou mít různou prioritu a přerušení s vyšší prioritou může přerušit kód vykonávaný v přerušení s nižší prioritou. Vstup do přerušení nám také automaticky zálohuje registry R0 až R3, R12, LR, PC a PSR. V jádru nalezneme SysTick časovač o velikosti 24bitů používaný např. pro podporu OS a vnořený vektorový řadič přerušení (NVIC).

#### Obecné registry



Obr. 2 Registry Cortex-M



Obr. 3 Zarovnané a nezarovnané proměnné v paměti

Cortex-M0 má tří-úrovňovou pipeline a má ze všech Cortex rodiny nejmenší jádro (obsahuje přibližně 12000 logických hradel). To omezilo množství instrukcí na hodnotu 56. Je navrženo, aby přímo konkurovalo tradičním 8/16-bit mikrokontrolerům, jak s výrobní cenou, tak i s potřebou a výkonem, který je mnohonásobně větší. Může obsluhovat až 32 zdrojů přerušení, u kterých můžeme nastavit prioritu.

Cortex-M0+ má stejné vlastnosti jádro jako Cortex-M0, ale jeho pipeline má jenom dva stupně. V tomto jádře nalezneme jednotku MPU (jednotka ochrany paměti).

Cortex-M3 je velice rozšířen v mikrokontrolérech. Má větší výkon než Cortex-M0 nebo Cortex-M0+. Má 3 úrovně pipeline a obsahuje také jednotku MPU. Na rozdíl od předchozích Cortex jader může obsluhovat až 256 zdrojů přerušení.

Cortex-M4 má vlastnosti jako jádro Cortex-M3 a je rozšířeno o předvídání skoku (zajistí, aby se pipeline plnila instrukcemi umístěné v místě skoku), 8/16-bit aritmetické SIMD (instrukce pro práci s vektorovými daty), obsahuje také MPU jednotku a FPU (pro zpracování float aritmetiky).

Řada Cortex-R je stavěna na vysoký výkon real-time aplikací s nízkou dobou odezvy zpracování kódu. Tato řada kromě instrukcí Thumb 2 podporuje také instrukce předchozích rodin a to ARM a Thumb. To zaručuje binární kompatibilitu s rodinami např. ARM7, ARM9 a ARM1156. Uplatnění této řady můžeme nalézt v aplikacích požadující vysoký výkon a nízkou odezvu a bezpečnostní aplikace.

Řada Cortex-A je nejvýkonnější z rodiny Cortex. Je zaměřena na OS platformu a uživatelské aplikace. Obsahuje původní ARM instrukční sadu, Thumb a Thumb 2 instrukční sady, které jsou doplněny o NEON instrukce (instrukce pro práci s vektorovými daty v desetinné pohyblivé čáře). Můžeme je nalézt v celé řadě zařízení od mobilních telefonů, přes smart TV až po notebooky.

## 2.1.4 Porovnání ARM rodin

Zásadní rozdíly, které můžeme vidět u porovnání rodin Cortex s předchozími jádry, jsou podporované instrukce a jiný typ řadiče přerušení. Srovnání nalezneme v Tab. 1.

Rodina	jádro	Architektonická verze	Podporované instrukce	Jednotka ochrany paměti	řadič přerušení
ARM7	ARM7TDMI	ARMv4T	ARM/Thumb	není	VIC
ARM9	ARM926EJ-S	ARMv5TEJ	ARM/Thumb	MMU	VIC
Cortex	Cortex-M0	ARMv6-M	Thumb/Thumb2	není	NVIC
	Cortex-M0+	ARMv6-M	Thumb/Thumb2	MPU	NVIC
	Cortex-M3	ARMv7-M	Thumb/Thumb2	MPU	NVIC
	Cortex-M4	ARMv7-ME	Thumb/Thumb2	MPU	NVIC

Tab. 1 Celkové srovnání rodin

## 2.2 Instrukční sady

Podíváme se na popis jednotlivých instrukčních sad ARM, Thumb a Thumb 2 a v závěru porovnáme jejich výhody, nevýhody.

### 2.2.1 Instrukční sada ARM

Je původní instrukční sada, která vznikla v prvních verzích ARM procesorů. Šířka instrukčního slova je pevně dána na 32bit. V kódování instrukce nalezneme vždy 4bity (jsou umístěny nejvíce vlevo), které umožňují podmíněné spuštění instrukce (14 podmínek) nebo



nepodmíněné. Další možností této instrukční sady je možnost využití více operandů, dva registry a 8bit konstanta nebo tři registrů. Další možnost, která je umožněna u aritmetických operací je rotace jednoho vstupního operandu a rozhodnutí, které příznakové bity budou po provedení instrukce nastavovány. Instrukce B (branch) a BL (branch and link), které provádí skok v programu za využití 24-bitové konstanty, která je o dva bity posunuta vlevo a rozšířena na 26 bitů, sečtena s PC a odečtena o 8 (protože než se provede skok, rozpracují se dvě následující instrukce, na novějších procesorech je z důvodu kompatibility), to umožňuje provést skok v rozsahu  $\pm 32\text{MB}$ , instrukce BL navíc uloží adresu následující instrukce do link registru (R14). Velice problematické u této instrukční sady je vkládání konstant do registrů, protože existuje pouze možnost vložení 1byte, který může být ještě o několik bitů posunut, nebo musíme využít instrukcí pro práci s pamětí a s využitím ukazatele uloženého v registru, přičemž konstanta musí být uložena v paměti. Pro ukládání nebo načítání dat do paměti lze pouze přes instrukce typu LDR a STR, můžeme také ukládat nebo načítat až 16 námi zvolených registrů najednou.

## 2.2.2 Instrukční sada Thumb

Thumb je podmnožina původní ARM instrukcí, která byla vytvořena z analýzy kompilátoru jazyka C/C++. Byla vytvořena z nutnosti šetřit paměť, protože se ARM procesory rozšiřovaly na vestavěné (embedded) systémy. Šířka instrukčního slova byla zmenšena na 16 bitů, proto nemůže kódovat všechny instrukce, které máme na ARM instrukční sadě. U této sady nenalezneme horní 4bity, které u ARM instrukcí byly, proto tady nemáme možnost provádět podmíněné instrukce, pouze máme k dispozici podmíněné a nepodmíněné skoky nebo operace s bity. Dalším omezením je použití pouze dvou operandů v instrukci a omezení registru na R0 až R7. Pro práci s registry LR (R14), SP (R13) a PC (R15) existují speciální instrukce. Omezená je také instrukce B (branch), které může provést skok v rozsahu  $\pm 2\text{KB}$ . Instrukce BL (branch a link) má složitější chování než instrukce v ARM. První její varianta má v instrukčním kódu nastaven bit H (viz [10] strana 42) na 0, posune 11bitovou hodnotu uloženou v instrukci o 12bitu vlevo, sečte a uloží výsledek do LR. Druhá varianta má tento bit H nastavený na hodnotu 1, posune 11bitovou hodnotu uloženou v instrukci o 1bit vlevo, sečte s LR a pak prohodí obsah registru LR s PC. Vkládání konstant je stejně problematické jako v případě ARM instrukcí. Přepínání z/do instrukčního módu ARM a Thumb se musí provést pomocí speciálního skoku. Existují dvě instrukce:

- BX skok a přepnutí do Thumb a zpět
- BLX skok a uložení návratové adresy do LR a přepnutí do Thumb a zpět

## 2.2.3 Instrukční sada Thumb 2

Byla vytvořena za účelem využít přednosti ARM a Thumb instrukcí. Nalezneme zde instrukce o šířce 16bitů nebo 32bitů. Tato instrukční sada je nadmnožinou sad ARM a Thumb. Výhoda této instrukční sady je, že má podobnou kódovou hustotu jako Thumb instrukce, ale také podobný výkon jako ARM instrukce a odstraňuje nevýhodu přepínání mezi 16bitovými instrukcemi a 32bitovými instrukcemi (přepnutí nevyžaduje speciální instrukci skoku). Má pokročilé operace s bity nebo bitovými poli. Vkládání konstant je zde daleko lépe vyřešeno pomocí dvou instrukcí, jedna nahrává spodních 16 bitů do registru a druhá nahrává horní 16 bitů do registru, přičemž spodních 16 bitů se nezmění.

## 2.2.4 Srovnání jednotlivých instrukčních sad

Největší rozdíl, který můžeme vidět z Tab. 2 je, že instrukční sada Thumb 2 má lépe vyřešeno vkládání konstanty do registru nebo má instrukce pro práci s bitovými poli. Také šířka instrukcí v Thumb 2 nejsou stejné, kompilátor vybere nejlepší variantu instrukce buď 16bitové nebo 32bitové v závislosti na rychlosti nebo velikosti kódu.

Instrukční sada	Šířka instrukce v bitech	Velikost konstanty v instrukci v bitech	Instrukce pro práci bitovými poli
ARM	32	8	ne
Thumb	16	8	ne
Thumb 2	16/32	16	ano

Tab. 2 Celkové srovnání instrukčních sad

## 3 Přehled dostupných ARM mikrokontrolérů

V této kapitole se podíváme na dostupné mikrokontroléry na trhu a srovnáme, jaké jsou mezi nimi rozdíly z ohledu na použité jádro, výkon, velikost programové paměti a datové paměti, systémových periférií a možné rozšiřitelnosti externího hardware a konektivity s ostatními systémy.

### 3.1 STMicroelectronic

Jejich ARM mikrokontroléry mají dobrou výbavu, jsou optimalizovány na nízkou spotřebu a nízké napětí. Jejich jádra jsou založena na řadě Cortex-M, jsou vyráběny v 7 seriích: STM32F0, STM32F1, STM32F2, STM32F3, STM32F4, STM32L1, a STM32W. Všechny mikropočítače od STMicroelectronic obsahují tyto periferie:

- 1x RTC
- 1x WDG

#### 3.1.1 Řada STM32F0

Tato řada konkuruje tradičním 8 a 16 bitovým mikrokontrolerům svoji nízkou spotřebou, vysokým výkonem a nízkými výrobními náklady. Je založena na jádrech nejnižší verzi jádra Cortex-M0. Teoretický výkon této řady je 0,79 DMIPS/MHz (Dhrystone 2.1). Jejich výbava obsahuje:

- jádro Cortex-M0 pracující až na 48MHz
- až 2xSPI, až 2x USART, až 2x I<sup>2</sup>C
- až 11 časovačů
- 5 DMA kanálů
- 1x 12-bit ADC, až 16 kanálů
- 6 kanálů PWM
- SWD

Pouze řada STM32F050

- 16 až 32Kb Flash paměti
- 4KB SRAM paměti s hardwarovou kontrolou parity
- Až 25 I/O s 5V tolerancí
- TSSOP20, UFQFPN28, UFQFPN32, LQFP48

Pouze řada STM32F051

- 16 až 64Kb Flash paměti
- 8KB SRAM paměti s hardwarovou kontrolou parity
- HDMI-CEC
- 1x DAC převodník
- až 18 kanálů pro kapacitní snímání s podporou touchkey
- dva rychlé komparátory
- Až 36 I/O s 5V tolerancí
- UFQFPN32, LQFP32, LQFP48, LQFP64

### 3.1.2 Řada STM32F1

Tato řada je založena na výkonnějším jádře Cortex-M3, jenž má uplatnění v široké škále aplikací a zachovává si nízkou spotřebu, nízké napětí a vysoký výkon, který dosahuje 1,25 DMIPS/MHz (Dhrystone 2.1). Jejich výbava obsahuje:

- Jádro Cortex-M3
- až 3xSPI, až 5x USART, až 2x UART, až 2x I<sup>2</sup>C
- až 16x časovačů
- až 12 DMA kanálů
- až 2x 12-bit ADC, až 16 kanálů
- až 6 kanálů PWM
- SWD a JTAG
- až 112 I/O s 5 V tolerancí

Pouze řada STM32F100

- Pracovní frekvence až 24 MHz
- 16 až 512KB Flash paměti
- 4 až 32KB SRAM paměti
- LCD paralelní rozhraní 8080/6800 mod
- až 1xřadič externí paměti SRAM, PSRAM, NOR a NAND
- 2x12-bit DAC převodník
- LQFP48, LQFP64, TFBGA64, LQFP100, LQFP144

Pouze řada STM32F101

- pracovní frekvence 36MHz
- 16KB až 1MB Flash paměti
- 4 až 80KB SRAM paměti
- LCD paralelní rozhraní 8080/6800 mód
- až 1x řadič externí paměti SRAM, PSRAM, NOR a NAND
- 2x12-bit DAC převodník

- VFQFPN36, VFQFPN48, LQFP48, LQFP64, LQFP100, LQFP144

#### Pouze řada STM32F102

- pracovní frekvence až 48MHz
- 16 až 128KB Flash paměti
- 4 až 16KB SRAM
- 1x USB 2.0 FS
- LQFP48, LQFP64

#### Pouze řada STM32F103

- pracovní frekvence až 72MHz
- 16KB až 1 MB Flash paměti
- 6 až 96 KB SRAM paměti
- USB 2.0 FS
- CAN rozhraní 2.0B
- SDIO rozhraní
- VFQFPN36, VFQFPN48, LQFP48, TFBGA64, BGA64, WLCSP64, LQFP64, LQFP100, BGA100, UFBGA100, LFBGA100, LQFP144, LFBGA144,

#### Pouze řada STM32F105/107

- pracovní frekvence až 72MHz
- 64 až 256 KB Flash paměti
- 64KB univerzální SRAM paměti
- 10/100 Ethernet MAC s 4KB vyhrazené paměti SRAM
- USB 2.0 FS OTG s 1,25KB vyhrazené paměti SRAM
- 2x CAN rozhraní 2.0A s 512 byte vyhrazené paměti SRAM
- 2x12-bit DAC převodník
- LQFP64, LQFP100, LFBGA100

### 3.1.3 Řada STM32F2

Tato řada opět používá jádro Cortex-M3, které používá pokročilejší 90nm výrobní technologii. To umožňuje u této řady vysoký stupeň integrace periferií např. hardwarového šifrování a podporu externího paměťového rozhraní. Má také zvýšený výpočetní výkon na 1,25 DMIPS/MHz (Dhrystone 2.1) a nízkou spotřebu na úrovni 188uA/MHz. Jsou vybaveny:

- Jádro Cortex-M3
- Pracovní frekvence až 120 MHz
- až 1MB Flash paměti
- až 128+4 KB SRAM
- až 3xSPI, až 4x USART, až 3x I<sup>2</sup>C
- 10/100 Ethernet MAC
- USB 2.0 FS s OTG
- 2x CAN rozhraní 2.0A
- LCD paralelní rozhraní 8080/6800 mod
- SDIO rozhraní
- 1xřadič externí paměti SRAM, PSRAM, NOR a NAND

- 17x časovačů
- 16x DMA
- 3x 12-bit ADC, až 16 kanálů
- 2x 12-bit DAC převodník
- 8 až 14-bit paralelní kamerové rozhraní s max. 48MB/s
- Analogový generátor náhodných čísel
- až 4 kanálů PWM
- Až 138 I/O 5 V tolerancí
- SWD a JTAG
- WLCSP64+2, LQFP64, LQFP100, LQFP144, LQFP176, UFBGA176

Pouze řada STM32F215/217

- hardwarový akcelerátor pro AES128, AES192, AES256, Triple DES, HASH

### 3.1.4 Řada STM32F3

Tato řada používá výkonnější jádro Cortex-M4 a má DPS a FPU instrukce schopné pracovat až na 72 MHz. Je vybavena pokročilými analogovými periferiemi. Výkon této řady je 1,25 DMIPS/MHz (Dhrystone 2.1). Jsou vybaveny:

- Jádro Cortex-M4
- pracovní frekvence až 72MHz
- až 3xSPI, až 5x USART/UART, až 2x I<sup>2</sup>C
- až 13 časovačů
- 12 DMA kanálů
- 4x ADC s nastavitelnou šířkou 12/10/8/6-bit a rychlostí 0.2uS, až 39 kanálů
- 2x 12-bit DAC převodník
- Až 6 kanálů PWM
- Až 24 kanálů kapacitního snímače
- SWD a JTAG
- Až 87 I/O s 5V tolerancí

Pouze řada STM32F302/303

- 128 až 256KB Flash paměti
- 40KB SRAM pro datovou sběrnici s hardwarovou kontrolou parity
- 8KB SRAM pro instrukční sběrnici s hardwarovou kontrolou parity (CCM)
- 7 rychlých analogových komparátorů
- USB 2.0 FS
- CAN rozhraní 2.0B
- až 24 kanálů pro kapacitní snímání s podporou touchkey
- LQFP48, LQFP64, LQFP100

Pouze řada STM32F313

- 128 až 256KB Flash paměti
- 40KB SRAM pro datovou sběrnici s hardwarovou kontrolou parity
- 8KB SRAM pro instrukční sběrnici s hardwarovou kontrolou parity (CCM)
- CAN rozhraní 2.0B

- 7 rychlých analogových komparátorů
- až 23 kanálů pro kapacitní snímání s podporou touchkey
- LQFP48, LQFP64, LQFP100

Pouze řada STM32F372/383

- 64 až 256KB Flash paměti
- 32KB SRAM s hardwarovou kontrolou parity
- 1x 12-bit ADC, až 16 kanálů
- 3x 16-bit Sigma Delta ADC
- 3x 12-bit DAC převodník
- 2 rychlé analogové komparátory
- HDMI-CEC
- LQFP48, LQFP64, LQFP100, UFBGA100

### 3.1.5 Řada STM32F4

Používá také jádro Cortex-M4, které je vyráběno 90nm výrobní technologií a umožňuje je taktovat na frekvenci až 180Mhz, ale přitom odběr zůstává na nízké hodnotě 238uA/Mhz. Jsou také vybaveny jednotkami FPU a DSP. Výkon dosahuje hodnoty 1,25DMIPS/MHz (Dhrystone 2.1). Jsou vybaveny:

- Jádro Cortex-M4
- až 6xSPI, až 4x USART, až 4x UART až 3x I<sup>2</sup>C
- 10/100 Ethernet MAC
- USB 2.0 FS OTG
- LCD paralelní rozhraní 8080/6800 mod
- řadič externí paměti SRAM, CF, PSRAM, NOR a NAND
- 3x 12bit ADC s rychlostí 2,4MSPS, až 24 kanálů
- 2x 12bit D/A převodník
- Až 17 časovačů
- 16 kanálové DMA
- 4 kanály PWM
- SWD a JTAG
- Generátor náhodných čísel
- 8 až 14-bit paralelní kamerové rozhraní až s rychlostí 54MB/s

Pouze řada STM32F405/407

- Pracovní frekvence až 168 MHz
- Až 1MB Flash paměti
- Až 192+4KB SRAM paměti vložené 64KB z CCM
- 2x CAN rozhraní 2.0B
- SDIO rozhraní
- LQFP64, WLCSP90, LQFP100, LQFP144, LQFP176, UFBGA176

Pouze řada STM32F415/417

- Pracovní frekvence až 168 MHz
- Až 1MB Flash paměti
- Až 192+4KB SRAM paměti vložené 64KB z CCM

- 2x CAN rozhraní 2.0B
- Šifrovací akcelerátor pro AES 128, 192, 256, Triple DES, HASH a HMAC
- SDIO rozhraní
- LQFP64, WLCSP90, LQFP100, LQFP144, LQFP176, UFBGA176

Pouze řada STM32F427

- Pracovní frekvence až 168 MHz
- Až 2MB Flash paměti
- Až 256+4KB SRAM paměti vložené 64KB z CCM
- 2x CAN rozhraní 2.0B
- SDIO rozhraní
- LQFP100, LQFP144, LQFP176, UFBGA176

Pouze řada STM32F429/439

- Pracovní frekvence až 180 MHz
- Až 2MB Flash paměti
- Až 256+4KB SRAM paměti vložené 64KB z CCM
- 2x CAN rozhraní 2.0B
- LCD-TFT řadič podporující až VGA rozlišení s dedikovaným Chrom-ART akcelerátorem
- 1x SAI
- LQFP100, LQFP144, LQFP176, LQFP208, UFBGA176, TFBGA216, WLCSP144

Pouze řada STM32F437

- Pracovní frekvence až 168 MHz
- Až 2MB Flash paměti
- Až 256+4KB SRAM paměti vložené 64KB z CCM
- 2x CAN rozhraní 2.0B
- Šifrovací akcelerátor pro AES 128, 192, 256, Triple DES, HASH a HMAC
- SDIO rozhraní
- LQFP100, LQFP144, LQFP176, UFBGA176

### 3.1.6 Řada STM32L1

Tato řada využívá jádro Cortex-M3. Ve výbavě nalezneme LCD řadič pro znakové displeje. Je zaměřena velmi nízkou spotřebu s několika režimy nízké spotřeby:

- 0.35  $\mu$ A Standby mode
- 1.3  $\mu$ A Stanby mode + RTC
- 0.65  $\mu$ A Stop mode
- 1.5  $\mu$ A Stop mode + RTC
- 11  $\mu$ A ultra-low Run mode
- 238  $\mu$ A/MHz ve standartním módu
- 8  $\mu$ s probouzení

Výkon dosahuje 1,04 DMIPS/MHz (Dhrystone 2.1) - jsou schopny pracovat až na 32 MHz. Je vybavena:

- Jádro Cortex-M3 pracující až na 32MHz

- 4 až 12KB EEPROM
- až 3xSPI, až 5x USART až 2x I<sup>2</sup>C
- 1x USB 2.0
- řadič znakové LCD až 40x8 znaku
- až 11 časovačů
- 12 kanálů DMA
- 1x 12-bit ADC, až 40 kanálů
- 1x 12-bit DAC, 2 kanály
- SWD a JTAG

Pouze řada STM32L100

- až 128 KB Flash paměti
- až 10 KB SRAM paměti
- Podpora USART bootloader
- 2 nízko příkonové analogové komparátory
- LQFP64, UFQFPN48

Pouze řada STM32L15x

- až 384KB Flash paměti s ECC
- až 48KB SRAM paměti
- Podpora USB a USART bootloader
- 1x SDIO rozhraní
- 2 nízko příkonové analogové komparátory
- až 34 kanálů pro kapacitní senzor
- LQFP48, UFQFP48, LQFP64, BGA64, LQFP100, BGA100

Pouze řada STM32L162

- až 384KB Flash paměti s ECC
- až 48KB SRAM paměti
- 12KB EEPROM s ECC
- 128B záložních registrů
- Podpora USB a USART bootloader
- řadič externí paměti SRAM, CF, PSRAM, NOR a NAND
- 1x SDIO rozhraní
- 2 nízko příkonové komparátory
- až 34 kanálů pro kapacitní senzor
- LQFP64, LQFP100, BGA100, LQFP144, UFBGA132, WLCSP64

### 3.1.7 Řada STM32W

Tato řada je opět postavena na jádru Cortex-M3 a je určena na bezdrátové účely. Výkon této řady je 1,25 DMIPS/MHz (Dhrystone 2.1) taktovací frekvence může být až 24 MHz.

Výbava je:

- Jádro Cortex-M3 pracující až na 24MHz
- 2,4 GHz IEEE 802.15.4 transceiver a low MAC
- 1x SPI, 1x UART, 1x I<sup>2</sup>C
- až 256KB Flash paměti



- až 16KB paměti
- šifrovací akcelerátor pro AES128
- 2x časovač
- 6 DMA
- 1x 12-bit ADC, až 6 kanálů
- -99 dBm citlivost přijímače, nastavitelná až na -100 dBm(1% chyb přenosu)
- Vysílač nastavitelný až na +8 dBm
- SWD a JTAG
- VFQFPN48

## 3.2 Atmel

Atmel je známý výrobce mikrokontrolérů např. jejich AVR, x51, ARM a atd. a dalších integrovaných obvodů. Mikrokontroléry založené na ARM obsahují dobrou výbavu v široké škále zaměření v řadách: SAM3A/X, SAM3N, SAM3S, SAM3U, SAM4L, SAM4E, SAM4S, SAM7S/SE, SAM7X/XC, SAL7L a SAM9XE (vyrobí i další, ale nejsou schopny samostatně fungovat, protože jim chybí Flash paměť).

### 3.2.1 Řada SAM3A/X

Tato řada je založena na jádru Cortex-M3 a je zaměřena na konektivitu. Podporuje vysokorychlostní přenosy dat a je tedy ideální pro použití v síťových aplikacích a inteligentních sítích. Ve výbavě nalezneme:

- Jádro Cortex-M3 pracující až na 84Mhz
- 256 až 512KB Flash paměti s 128-bitovou šířkou a paměťovým akcelerátorem
- 32 až 100KB SRAM s dvěma bankami
- až 6x SPI, 1x SSC, 4x USART, 1x UART, 2x TWI(I<sup>2</sup>C kompatibilní)
- Jeden Ethernet MAC 10/100
- USB 2.0 Device/Mini Host s rychlostí 480 Mbps s 4KB FIFO
- 2x CAN 2.0B řadič s 8 Mailboxes
- 16KB ROM s bootladerem podporující UART, USB a IAP rutiny
- Řadič paměti typu SRAM, NOR, NAND a NAND Flash s 4KB RAM buffrem a ECC
- 1x HSMCI(SDIO/SD/MMC) až dva sloty
- 9x 32-bit časovačů/čítačů
- 17x kanálové periferní DMA, 6x kanálové centrální DMA a dedikované DMA pro USB a Ethernet
- 1x 12-bit ADC, až 16 kanálů
- 1x dvoukanálový 12-bit DAC s rychlostí 1MSPS
- RTC obvod
- Až 8 kanálů 16-bit PWM
- 1x WDT
- SWD a JTAG
- Úsporný režim sleep se spotřebou 2.5μA
- 1x generátor náhodných čísel
- Ochrana registrů před zápisem
- LQFP100, LFBGA100, LQFP144, LFBA144

### 3.2.2 Řada SAM3N

Tato řada je založena na jádru Cortex-M3 a je optimalizována na vysoký výkon v aplikacích a na úsporu energie. Je také pinově kompatibilní s SAM7S a SAM3S, což umožňuje migraci mezi rodinami. Ve výbavě nalezneme:

- Jádro Cortex-M3 pracující až na 48MHz
- 16 až 256KB Flash paměti s 128-bitovou šířkou a paměťovým akcelerátorem
- 4 až 24KB SRAM
- 1x SPI, až 2x USART s RS-485 a SPI módem kompatibilní, 2x USART, 2x I<sup>2</sup>C
- 16KB ROM s bootloaderem podporující UART a IAP rutiny
- 10 kanálů DMA
- 6x časovač
- 1x 10-bit ADC rychlostí 384 KSPS, až 16 kanálů
- 1x 10-bit DAC rychlostí 500 KSPS
- 1x RTC
- 4 kanály 16-bit PWM
- 1x WDT
- SWD a JTAG
- úsporný režim Sleep se spotřebou 3μA
- LQFP48, LQFP64, LQFP100

### 3.2.3 Řada SAM3S

Tato řada je opět založena na jádru Cortex-M3, ale je zaměřena na nízkou spotřebu. Je pinově kompatibilní s SAM7S a umožňuje tedy migraci na novější řadu SAM3S. Ve výbavě nalezneme:

- Jádro Cortex-M3 pracující až na 64 MHz
- 64 až 256KB Flash paměti s 128-bitovou šířkou a paměťovým akcelerátorem
- 16 až 48KB SRAM
- 1x SPI, až 2 USART, až 2x I<sup>2</sup>C
- USB 2.0 Device rychlostí 12 Mbps s 2668byte FIFO
- 16KB ROM s bootloaderem podporujícím UART a USB a IAP rutiny
- 8-bitový řadič paměti: SRAM,PSRAM,NOR a NAND Flash
- 1x vysokorychlostní rozhraní pro SDIO/SD/MMC
- 6x časovač
- 22 kanálů DMA
- 1x 12-bit ADC rychlostí 1Msps, až 16 kanálů
- 2 kanály 12-bit DAC s rychlostí 1Msps
- 4 kanály PWM
- 1x WDT
- SWD a JTAG
- Ochrana registrů před zápisem
- LQFP48, LQFP64, LQFP100

### 3.2.4 Řada SAM3U

Je zaměřena na vysokorychlostní přenosy, hlavně přes rychlé USB rychlostí 480 Mbps. Uplatnění nalezne v aplikacích pro intenzivní komunikaci, jako vysokorychlostní brány v průmyslu, zdravotnictví a spotřebitelských aplikacích. Ve výbavě nalezneme:

- Jádru Cortex-M3 pracující až na 96MHz
- 64 až 256KB Flash paměti s 128-bitovou šířkou a paměťový akcelerátorem
- 16 až 48KB SRAM
- 1x SPI, až 4x USART, 1x UART, až 2x TWI (I<sup>2</sup>C kompatibilní)
- USB 2.0 480 Mbps
- 16KB ROM s bootloaderem podporujícím UART a USB a IAP rutiny
- Řadič pamětí typu SRAM, NOR, NAND a NAND Flash kontrolováno s 4KB RAM buffrem a ECC
- 3x časovač
- 17 kanálů DMA
- 1x HSMCI (SDIO/SD/MMC)
- 8-kanálové 1x 12-bit ADC a 8-kanálové 1x 10-bit ADC
- 1x RTC
- 4 kanály PWM
- 1x WDT
- SWD a JTAG
- LQFP100, TFBGA100 ,LQFP144, LFBGA144

### 3.2.5 Řada SAM4L

Tato řada využívá jádro Cortex-M4 a je zaměřena na velmi nízkou spotřebu a nalezne uplatnění u bateriově napájených aplikacích. Ve výbavě nalezneme:

- Jádru Cortex-M4 pracující až na 48MHz
- picoPower Technologie pro ultra nízkou spotřebu
- v aktivním módu spotřeba 90μA/MHz
- 128 až 256KB Flash paměti s 64-bitovou šířkou
- Až 32KB SRAM
- 1x SPI, 1x USART s ISO7816, 3x USART s SPI módem, 4x TWI (I<sup>2</sup>C kompatibilní)
- USB 2.0 Device a Host s rychlostí 12 Msps
- 1x šifrovací akcelerátor pro AES128
- LCD řadič až 40x4 segmentů
- 6x 16-bit časovač/čítač
- 16 kanálů periferních DMA
- 1x 12-bit ADC rychlostí 300Ksps, až 16 kanálů
- 1x 10-bit DAC rychlostí 500Ksps
- 1x RTC
- 1x WDT
- CRC32
- SWD a JTAG
- Generátor náhodných čísel
- GLOC
- LQFP48, LQFP64, LQFP100

### 3.2.6 Řada SAM4E

Je založena na jádrech Cortex-M4 se zaměřením na vysoký výkon, obsahuje FPU jednotku pro práci s čísly v pohyblivé řádové čárce s jednoduchou přesností a pokročilé možnosti připojení pro aplikace. Obsahuje bohatou sadu periférií pro spojení s okolím. Ve výbavě nalezneme:

- Jádro Cortex-M4 s 2 KB Cache pracující až na 120MHz
- Až 1MB Flash paměti
- 128KB SRAM
- 1x SPI 2x USART, 2x UART, 2x TWI (I<sup>2</sup>C kompatibilní)
- 1x Ethernet MAC 10/100 Mbps
- 1x USB 2.0 s rychlostí 12Mbps a s 2668B FIFO
- 2x CAN 2.0B
- Hardwarové šifrování AES 256-bit
- 16KB ROM s bootloaderem podporující UART a IAP rutiny
- Řadič paměti typu SRAM, NOR, NAND a NAND Flash
- 1x HSMCI(SDIO/SD/MMC)
- 3x 32-bit časovač/čítač
- 33 kanálů DMA
- 2x 16-bit ADC, až 24 kanálů
- 1x 2-kanálový 12-bit DAC s rychlostí 1Msps
- 1x analogový komparátor
- 1x RTC
- 4 kanály PWM
- 1x WDT
- SWD a JTAG
- LFBGA144, LQFP144

### 3.2.7 Řada SAM4S

Je založena na jádru Cortex-M4 s vyšším výkonem a energetickou účinností obsahující bohatou periferní výbavu. Má pinovou kompatibilitu se současnými SAM3 a umožňuje snadný přechod na SAM4S. Ve výbavě nalezneme:

- Jádro Cortex-M4 s 2KB Cache pracující až na 120MHz
- Až 2MB Flash
- Až 160KB SRAM
- 1x SPI, 2x USART, 2x UART, 2x TWI (I<sup>2</sup>C kompatibilní)
- 1x USB 2.0 s rychlostí 12Mbps a s 2668B FIFO
- LCD řadič až 40x4 segmentu
- 16KB ROM s bootloaderem podporující UART a USB a IAP rutiny
- 8-bitový řadič paměti: SRAM,PSRAM,NOR a NAND Flash
- 1x vysokorychlostní rozhraní pro SDIO/SD/MMC
- 2x 16-bit časovač/čítač
- 22 kanálů DMA
- 1x 12-bit ADC s 1Msps, až 16 kanálů

- 1x12-bit DAC rychlostí 1Msps, 2 kanály
- 1x analogový komparátor
- 1x RTC
- 4 kanály PWM
- CRC32
- 1x WDT
- SWD a JTAG
- Ochrana registrů před zápisem
- LQFP64, LQFP100

### 3.2.8 Řada SAM7S/SE

Je založena na starším jádru ARM7TDMI, je určena na práci v síťových aplikacích nebo v dalších náročných aplikacích, Ve výbavě nalezneme:

- Jádro ARM7TDMI pracující až na 55MHz
- 16 až 512KB Flash paměť
- 4 až 64KB SRAM
- 1x SPI, 2x USART, 1x TWI (I<sup>2</sup>C kompatibilní)
- 1x USB 2.0 s rychlostí 12Mbits
- Boot asistent
- až 1x paměťový řadič pro paměti typu SRAM, SDRAM, CompactFlash NAND Flash
- 3x časovač
- až 11 kanálů DMA
- 1x 10-bit ADC, až 8 kanálů
- 1x RTC
- 4 kanálové PWM
- 1x WDT
- JTAG
- QFN48, LQFP48, LQFP64, QFN64

### 3.2.9 Řada SAM7X/XC

Je opět založena na jádru ARM7TDMI a je určena na práci v síťových aplikacích. Ve výbavě nalezneme:

- Jádro ARM7TDMI pracující až 55MHz
- 128 až 512KB Flash paměti
- 32 až 128KB SRAM
- 2x SPI, 2x USART, 1x TWI (I<sup>2</sup>C kompatibilní)
- 1x Ethernet MAC 10/100 base-T
- 1x USB 2.0 FS
- až 1x CAN 2.0B
- 3x časovač
- 17 kanálů DMA
- 1x 10-bit ADC, až 8 kanálů
- 1x RTC
- 4 kanály PWM

- 1x WDT
- JTAG
- LQFP100, TFBGA100

### 3.2.10 Řada SAM7L

Je opět založena na jádru ARM7TDMI a je určena pro aplikace s nízkou spotřebou. Ve výbavě nalezneme:

- Jádro ARM7TDMI pracující až na 36MHz
- 64 až 128 KB Flash paměti
- 6KB SRAM paměti
- 1x SPI, 2x USART, 2x TWI (I<sup>2</sup>C kompatibilní)
- LCD řadič až 40 segmentu
- 3x časovač
- 11 kanálu DMA
- 1x 10-bit ADC, až 4 kanály
- 1x RTC
- 4 kanály PWM
- 1x WDT
- JTAG
- LQFP128, LFBGA144

### 3.2.11 Řada SAM9XE

Tato řada je založena na ARM926EJ-S. Je určena do náročných aplikací, kde požadujeme vysoký výkon. Ve výbavě nalezneme:

- Jádro ARM926EJ-S pracující až na 180MHz
- 128 až 512KB Flash paměti s 128-bitovým přístupem
- 16 až 32KB SRAM
- 2x SPI, 4x USART, 1x UART, 2x TWI (I<sup>2</sup>C kompatibilní)
- 1x Ethernet MAC 10/100 Base-T
- 1x USB 2.0 FS
- Paměťový řadič pro paměti SRAM, SDRAM NAND Flash a CompactFlash
- 1x MCI pro SD a MMC
- 6x Časovač
- 24 kanálu DMA
- 1x 8 nebo 10-bit ADC, 4 kanály
- 1x RTC
- 2x PWM
- 1x WDT
- JTAG
- Image Sensor Interface
- PQFP208, LFBGA217

### 3.3 Texas Instruments

Tento výrobce je poměrně známý svými mikrokontroléry v oblasti např. řízení motorů apod. Periferie jsou bohaté hlavně na časovače, nebo také na množství PWM kanálu, které jsou často využívány pro řízení.

#### 3.3.1 Řada TM4 Tiva C

Tato řada využila ve svých mikrokontrolérech jádro Cortex-M4 v 65 nm výrobní technologie umožnila zvětšit integraci, aniž by byla zvětšena spotřeba celého mikrokontroléru. Ve výbavě nalezneme:

- Jádro Cortex-M4 pracující až na 100MHz
- 256KB Flash paměti
- 32KB SRAM paměti
- 2 KB EEPROM paměti
- 4x SPI, 8 UART, 6x I<sup>2</sup>C
- 1x USB 2.0 s OTG
- 2x CAN 2.0 A/B
- 6x 16/32-bit a 6x 32/64-bit GPTM
- 32 kanálu DMA
- 2x 12-bit ADC, až 24 kanálů
- 2x PWM celkem 16 kanálů
- 3x analogový komprátor
- 2x WDT
- SWD a JTAG
- LQFP64, LQFP100, BGA157

#### 3.3.2 Řada TMS470 Hercules

Tato řada využila pro své mikrokontroléry jádro Cortex-M3, Flash a SRAM paměť mají korekci chyb (ECC), a proto se hodí do aplikací, kde vyžadujeme spolehlivou paměť. Ve výbavě nalezneme:

- Jádro Cortex-M3 pracující až na 80MHz
- až 640KB Flash paměti s ECC
- až 64KB SRAM paměti s ECC
- 2x SPI, 2x UART,
- 2x CAN
- až 18 kanálu časovače HET
- 1x 10-bit ADC, až 16 kanálů
- CRC
- JTAG
- PZ100

### 3.4 NXP

Je výrobce, který vyrábí celou řadu různých polovodičových součástek a jednou z jeho činností je výroba ARM mikrokontrolérů. Nejnížší řada ARM mikrokontrolérů mají nejchudší

periferní vybavu, ale cena je nízká a je na úrovni levných 8-bit mikrokontrolérů. Další řady mají už bohatší periferní vybavu.

### 3.4.1 Řada LPC800

Tato řada je ze všech nejmenší a je určena jako náhrada 8-bit mikrokontrolérů s vysokou energetickou účinností. Ve vybavě nalezneme:

- Jádro Cortex-M0+ pracující až na 30MHz
- 4 až 16KB Flash paměti
- 1 až 4KB SRAM
- Až 2x SPI, až 3x UART, 1xI<sup>2</sup>C
- Bootloader
- 4x časovač
- 1x WDT
- 4 kanály PWM
- SWD a JTAG
- DIP8, TSSOP16, TSSOP20, SO20

### 3.4.2 Řada LPC11xx

Stejně jako řada LPC800 i tato řada je zaměřena, aby konkurovala 8-bit mikrokontrolérům. Periferní vybava je už bohatší, ale používá jádro Cortex-M0, které nemá integrovanou jednotku ochrany paměti (MPU). Ve vybavě nalezneme:

- Jádro Cortex-M0 pracující až na 50MHz
- 4 až 128KB Flash paměti
- 1 až 12KB SRAM paměti
- 4KB EEPROM
- až 2x SPI, až 1x UART, 1xI<sup>2</sup>C
- až 1x USB
- až 1x CAN
- až 1x řadič LCD 4x40 segmentu
- 16KB boot ROM
- 4x časovač
- 1x 10-bit ADC, až 8 kanálů
- až 13 kanálů PWM
- 1x WDT
- SWD a JTAG
- SO20, TSSOP20, WLCSP20, HVQFN24, WLCS25, HVQFN33, LQFP48, TFBGA48, LQFP64, LQFP100

### 3.4.3 Řada LPC12xx

Tato řada je na tom dost podobně jako LPC11xx. Ve vybavě nalezneme:

- Založena na jádře Cortex-M0 pracující až na 30MHz



- 32 až 128KB Flash paměti
- 4 až 8KB SRAM paměti
- Až 1x SPI, až 2x UART, 1x I<sup>2</sup>C
- Až 1x CAN
- Až 1x řadič LCD 4x40 segmentu
- 4x časovač
- Až 21x DMA
- 1x 10-bit ADC, až 8 kanálů
- 1x RTC
- Až 13 kanálů PWM
- Až 1x WDT
- SWD
- LQFP 48, LQFP 64, LQFP100

#### 3.4.4 Řada LPC13xx

V této řadě nalezneme už výkonnější jádro a to Cortex-M3. Periferní výbava je na tom podobně jako předchozí řady. Ve výbavě nalezneme:

- Jádro Cortex-M3 pracující až na 72MHz
- 8 až 64KB Flash paměti
- 4 až 12KB SRAM paměti
- 4KB EEPROM
- Až 2x SPI, až 1x UART, 1x I<sup>2</sup>C
- 1x USB FS
- 16KB boot ROM
- 4x časovač
- 1x 10-bit ADC, až 8 kanálů
- Až 11 kanálů PWM
- 1x WDT
- SWD a JTAG
- HVQFN33, LQFP48

#### 3.4.5 Řada LPC17xx

Je určena do náročnějších aplikací, kde jsou vyžadovány větší možnosti konektivity nebo možnosti zobrazování na LCD. Ve výbavě nalezneme:

- Jádro Cortex-M3 pracující až na 120MHz
- 32 až 512KB Flash paměti
- 8 až 96KB SRAM paměti
- Až 3x SPI, až 5x UART, 3x I<sup>2</sup>C
- 1x Ethernet MAC
- 1x USB FS s OTG
- 1x CAN
- 1x řadič LCD podporující STN a TFT displeje
- 1x SDIO
- 1x řadič pamětí podporující FLASH, SRAM a SDRAM

- 4x časovač
- 8x DMA
- 1x 12-bit ADC, až 8 kanálů
- 1x 10-bit DAC
- 1x RTC
- Až 6 kanálů PWM
- 1x WDT
- SWD a JTAG
- LQFP80, LQFP100, TFBGA100, LQFP144, TFBGA180, TFBGA208, LQFP208,

### 3.4.6 Řada LPC18xx

Podobně jako řada LPC17xx i tato je určena na aplikace náročné na konektivitu nebo práci s LCD. Ve výbavě nalezneme:

- Jádro Cortex-M3 pracující až na 180MHz
- 512 až 1024KB Flash paměti
- 104 až 200KB SRAM paměti
- Až 4x SPI, až 4x UART, 2x I<sup>2</sup>C
- 1x Ethernet MAC 10/100T
- 2x CAN 2.0B
- 2x USB, první s OTG
- 1x řadič LCD podporující STN a TFT displeje
- 1x SDIO
- 1x řadič pamětí podporující FLASH, SRAM a SDRAM
- 4x časovač
- 8x DMA
- 1x 10-bit ADC, až 8 kanálů
- 1x 10-bit DAC
- 1x RTC
- až 2 kanálů PWM
- SWD a JTAG
- TFBGA100, LQFP144, LQFP208, LBGA256

### 3.4.7 Řada LPC40xx

Tato řada je zaměřena na podobné využití jako dvě předchozí řady, obsahuje narozdíl od předešlých výkonnější jádro Cortex-M4. Ve výbavě nalezneme:

- Jádro Cortex-M4 pracující až na 120MHz
- 512 až 1024KB Flash paměti
- 104 až 200KB SRAM paměti
- Až 4x SPI, až 5x UART, 3x I<sup>2</sup>C
- 1x Ethernet MAC
- 1x USB FS s OTG

- 1x CAN
- 1x řadič LCD podporující STN a TFT displeje
- 1x SDIO
- 1x řadič pamětí podporující FLASH, SRAM a SDRAM
- 4x časovač
- 8x DMA
- 1x 12-bit ADC, až 8 kanálů
- 1x 10-bit DAC
- Dva komparátory
- 1x RTC
- Až 2 kanálu PWM
- 1x WDT
- SWD a JTAG
- LQFP80, LQFP144, TFBGA180, TFBGA208, LQFP208

### 3.4.8 Řada LPC43xx

Je nejvýkonnější řada, která na rozdíl od předchozích má dvě jádra, Cortex-M4 a Cortex-M0. Má ještě lepší možnosti konektivity než předchozí uvedené řady. Ve výbavě nalezneme:

- Jádro Cortex-M4 pracující až na 204MHz a Cortex-M0 sloužící jako koprocesor pro hlavní jádro pracující až na 204MHz.
- 512 až 1024KB Flash paměti
- 104 až 200KB SRAM paměti
- Až 4x SPI, až 5x UART, 2x I<sup>2</sup>C
- 1x Ethernet MAC 10/100T
- 2x USB, první s OTG
- 2x CAN 2.0B
- 1x řadič LCD podporující STN a TFT displeje
- 4x časovač
- 8x DMA
- 1x10-bit ADC, až 8 kanálů
- 1x10-bit DAC
- 1x RTC
- Až 2 kanály PWM
- 1x WDT
- SWD a JTAG
- TFBGA100, LQFP144, TFBGA180, LQFP208, LBGA256

### 3.5 Analog Devices

Tento výrobce se zabývá komponenty pro zpracování analogových signálů. Jeho ARM mikrokontroléry jsou vybaveny větším počtem ADC převodníku nebo převodníky s větší rozlišovací schopností. Součástí také bývá DAC převodník a PWM generátor.

#### 3.5.1 Řady ADuC7000

- Jádro ARM7TDMI pracující až na 41,78MHz
- Až 62KB Flash paměti
- 8KB SRAM
- 1x SPI, 1, UART, 2x I<sup>2</sup>C
- Rozhraní pro externí paměť do max. velikosti 512KB
- 1x 12-bit ADC, 16 kanálů nebo 2x 16(24)-bit ADC sigma-delta
- 1x 10(12)-bit DAC
- Teplotní senzor
- 4x časovač
- až 6 kanálů PWM
- JTAG
- LFCSP32, LFCSP40, LQFP48, LFCSP48, LFCSP80

#### 3.5.2 Řady ADuCM

- Jádro Cortex-M3 pracující až na 16MHz
- až 128KB Flash paměti
- až 8KB SRAM paměti
- 2x SPI, 1x UART, 1x I<sup>2</sup>C
- 2x časovač
- 11 kanálů DMA
- až 2x 24-bit ADC, 6 rozdílových nebo 11 nerozdílových vstupu, 4 vnitřní kanály pro sledování DAC, programovatelné zesílení (1 až 128) RMS, integrováno přesné referenční napětí
- 1x teplotní senzor
- 1x 12-bit DAC
- 1x 16-bit PWM
- SWD
- LFCSP48

### 3.6 Cypress

Jako jediný na světě vyrábí mikrokontroléry řady PSoC, které mají přímo v chipu programovatelné hradlové pole (PLD). To umožňuje určitým aplikacím zvýšení výkonu. Nejnovější tyto mikrokontroléry jsou založeny na architektuře Cortex.

#### 3.6.1 Řada PSoC 4000

- Jádro Cortex-M0 pracující až na 48MHz
- 32KB Flash paměti
- 4KB SRAM
- 2KB EEPROM
- 2x SPI, 2x UART, 2x I<sup>2</sup>C

- segmentový LCD řadič
- 1x 12-bit ADC, 8 kanálů
- 2x DAC
- 2x analogový komparátor
- 4x časovač
- 4x programovatelný digitální subsystém UDB
- SWD a JTAG
- SSSOP28, QFN40, TQFP48

### 3.6.2 Řada PSoC 5000LP

- Jádro Cortex-M3 pracující až na 67MHz
- až 32KB Flash s ECC
- až 64KB SRAM
- 2KB EEPROM
- až 24xSPI, až 24x USART, až 24x I<sup>2</sup>C
- 1x CAN 2.0b
- 1x USB 2.0 FS
- LCD řadič až 46x16 segmentu
- Podpora bootloaderu přes I2C, SPI, UART, USB
- 24x DMA
- 4x časovač
- až 2x 12-bit ADC, 8 kanálů a 1x20-bit delta-sigma ADC
- 4x DAC
- 4 analogové komparátory
- 4x PWM
- 20 až 24 UDB
- SWD a JTAG
- QFN68, TQFP100

## 3.7 Energy micro

Jejich ARM mikrokontroléry jsou pro aplikace, kde se musí šetřit s energií např. u bateriových řešení. Mají bohatou periferní výbavu a jádra vždy obsahují jednotku MPU.

### 3.7.1 Řada EFM32ZG - Zero Gecko

Tato řada obsahuje jádro Cortex-M0+ a je zaměřena na velmi nízkou spotřebu. Ve výbavě nalezneme:

- Jádro Cortex-M0+ pracující až na 24MHz
- 4 až 32KB Flash paměti
- 2 až 4KB SRAM paměti
- 1x USART, 1x SPI, 1x I2S, 1x UART
- Hardwarový ASE s 128-bit klíčem v 54 cyklech
- 1x SmartCard( ISO 7816)
- 4x časovače
- 4x DMA
- až 1x 12-bit ADC, až 4 kanály
- 1 analogový komparátor

- 1x RTC
- 6 kanálů PWM
- 1x WDT
- Kapacitní snímač, až 5 vstupu
- SWD
- QFN24, QFN32, TQFP48

### 3.7.2 Řada EFM32TG - Tiny Gecko

Tato řada je podobná předchozí, ale je použito výkonnější jádro Cortex-M3. Ve výbavě nalezneme:

- Jádro Cortex-M3 pracující až na 32MHz
- 8 až 32KB Flash
- 1 až 4KB SRAM
- 2x USART, 1x UART, 1x I2C
- Hardwarový AES s 128/256-bit klíčem v 54/75 cyklech
- Řadič LCD až na 8x20 segmentu
- 6x časovač
- 8x DMA
- 1x 12-bit ADC, 8 kanálů
- 1x 12-bit DAC
- 2x analogový komparátor
- 1x RTC
- 6 kanálů PWM
- 1x WDT
- SWD
- QFN24, QFN32, QFN64

### 3.7.3 Řada EFM32G – Gecko

Je zaměřena na náročnější aplikaci vyžadující více paměti, ale zároveň si zachovává svoji energetickou účinnost. Ve výbavě nalezneme:

- Jádro Cortex-M3 pracující až na 32MHz
- 16 až 128KB Flash
- 8 až 16KB SRAM
- 3x USART, 2x UART, 1x I2C
- Hardwarový AES s 128/256-bit klíčem v 54/75 cyklech
- Řadič LCD až na 4\*40 segmentu
- Podpora bootloaeru
- Paměťový řadič až na 64MB paměti
- 8x časovač
- 8x DMA
- 1x 12-bit ADC, 8 kanálů
- 1x 12-bit DAC
- 2x analogový komparátor
- 1x RTC
- Až 9 kanálů PWM

- 1x WDT
- SWD
- QFN32, QFN64, LQFP100, BGA112

### 3.7.4 Řada EFM32LG - Leopard Gecko

Má podobně vlastnosti jako předešlá řada, umožňuje připojit větší externí paměť. Ve výbavě nalezneme:

- Jádru Cortex-M3 pracující až na 48MHz
- 64 až 256KB Flash
- 32KB SRAM
- 3x USART, 2x UART, 2x I2C
- 1x USB 2.0
- Hardwarový AES s 128/256-bit klíčem v 54/75 cyklech
- Řadič LCD až na 8\*36 segmentu
- Podpora bootloaeru
- Paměťový řadič až na 256MB paměti
- 7x časovač
- 12x DMA
- 1x 12-bit ADC, 8 kanálů
- 1x 12-bit DAC
- 2x analogový komparátor
- 1x RTC
- Až 12 kanálů PWM
- 1x WDT
- SWD
- QFN64, TQFP64, LQFP100, LFBGA112, VFBGA120

### 3.7.5 Řada EFM32GG - Giant Gecko

Řada podobná předchozí řadám. Ve výbavě nalezneme:

- Jádru Cortex-M3 pracující až na 48MHz
- 512 až 1024KB Flash
- 128KB SRAM
- 3x USART, 2x UART, 2x I2C
- 1x USB 2.0 s OTG
- Hardwarový AES s 128/256-bit klíčem v 54/75 cyklech
- Řadič LCD až na 8\*36 segmentu
- Paměťový řadič až na 256MB paměti
- 7x časovač
- 12x DMA
- 1x 12-bit ADC, 8 kanálů
- 1x 12-bit DAC
- 2x analogový komparátor
- 1x RTC
- Až 12 kanálů PWM
- 1x WDT

- SWD
- QFN64, TQFP64, LQFP100, LFBGA112, VFBGA120

### 3.7.6 Řada EFM32WG - Wonder Gecko

Tato řada na rozdíl od předchozích používá výkonnější verzi Cortex-M4 a hodí se proto na náročnější výpočetní úlohy. Ve výbavě nalezneme:

- Jádro Cortex-M4 pracující až na 48MHz
- 64 až 256KB Flash paměti
- 32KB SRAM
- 3x SPI, 2x USART, 2x UART, 2x I<sup>2</sup>C
- 1x USB 2.0 s OTG
- Hardwarový AES s 128/256-bit klíčem v 54/75 cyklech
- až 1x Řadič LCD až na 8\*36 segmentu
- až 1x Paměťový řadič až 4x 256MB paměti
- 9x časovač
- 12x DMA
- 1x 12-bit ADC, 8 kanálů
- 2x 12-bit DAC
- 2x analogový komparátor
- 1x RTC
- Až 12 kanálů PWM
- 1x WDT
- SWD
- QFN64, TQFP64, LQFP100, LFBGA112, VFBGA120

## 3.8 Freescale

Freescale je výrobce, který vyrábí několik druhů mikrokontrolérů a jedny z nich jsou právě ARM mikrokontroléry. Nejrozšířenější verzí jejich mikrokontrolérů je Kinetis, které jsou poměrně slušně vybaveny na periferie.

### 3.8.1 Řada Kinetis K

Řada K obsahuje výkonné jádro Cortex-M4, které je doplněno o bohatou periferní výbavu, proto tato řada nalezne uplatnění v náročných aplikacích. Všechny tyto mikrokontroléry jsou vybaveny:

- Jádro Cortex-M4
- Až 5x SPI, 4x UART, 2x I<sup>2</sup>C , 1x I<sup>2</sup>S
- Až 1x SDHC řadič
- Až 2x CAN 2.0B
- 4x časovač
- 1x RTC
- 8 kanálů PWM
- 1x CRC
- Až 1x WDT
- 16 vstupu pro kapacitní snímač



- SWD a JTAG

### 3.8.1.1 Řada K10

- Pracovní frekvence až 120MHz
- 32 až 1024KB Flash paměti
- 8 až 128KB SRAM
- 2 až 16KB EEPROM
- Až 1x řadič externí paměti
- Až 1x Krypto akcelerátor podporující DES, 3DES, AES, MD-5, SHA-1a SHA-256
- až 16x kanálu DMA
- Až 2x 16-bit ADC, až 46 kanálů
- Až 2x 12-bit DAC
- Až 3x analogový komparátor

### 3.8.1.2 Řada K20

- Pracovní frekvence až 120MHz
- 32 až 1024KB Flash paměti
- 8 až 128KB SRAM
- 2 až 16KB EEPROM
- Až 1x řadič externí paměti
- 1x USB FS s OTG
- Až 1x Krypto akcelerátor podporující DES, 3DES, AES, MD-5, SHA-1a SHA-256
- až 32x kanálu DMA
- Až 4x 16-bit ADC, až 58 kanálů
- Až 2x 12-bit DAC
- Až 4x analogový komparátor

### 3.8.1.3 Řada K30

- Pracovní frekvence až 100MHz
- 64 až 512KB Flash paměti
- 8 až 128KB SRAM
- 2 až 4KB EEPROM
- Až 1x řadič externí paměti
- 1x řadič LCD až 38x8 nebo 42x4 segmentu
- až 16x kanálu DMA
- Až 2x 16-bit, až 27 kanálů
- Až 1x 12-bit DAC
- Až 4x analogový komparátor

### 3.8.1.4 Řada K40

- Pracovní frekvence až 100MHz
- 64 až 512KB Flash paměti
- 16 až 128KB SRAM
- 2 až 4KB EEPROM

- Až 1x řadič externí paměti
- Až 1x SDHC řadič
- 1x USB FS s OTG
- 1x řadič LCD až 38x8 nebo 42x4 segmentu
- až 16x kanálu DMA
- Až 2x 16-bit ADC, až 27 kanálů
- Až 2x 12-bit DAC
- Až 4x analogový komparátor

### 3.8.1.5 Řada K50

- Pracovní frekvence až 100MHz
- 128 až 512KB Flash paměti
- 16 až 128KB SRAM
- 2 až 4KB EEPROM
- Až 1x řadič externí paměti
- 1x Ethernet
- 1x USB FS s OTG
- Až 1x Krypto akcelerátor podporující DES, 3DES, AES, MD-5, SHA-1a SHA-256
- 1x řadič LCD až 38x8 nebo 42x4 segmentu
- až 16x kanálu DMA
- Až 2x 16-bit ADC, až 25 kanálů
- Až 2x 12-bit DAC
- Až 3x analogový komparátor

### 3.8.1.6 Řada K60

- Pracovní frekvence až 150MHz
- 256 až 1024KB Flash paměti s Cache 16KB
- 64 až 128KB SRAM
- 4 až 16KB EEPROM
- Až 1x řadič externí paměti podporující NAND Flash a DDR
- 1x Ethernet MAC 10/100Mbps
- 1x USB FS s OTG
- až 1x Krypto akcelerátor podporující DES, 3DES, AES, MD-5, SHA-1a SHA-256
- až 32x kanálu DMA
- Až 4x 16-bit ADC, až 48 kanálů
- Až 2x 12-bit DAC
- Až 3x analogový komparátor

### 3.8.1.7 Řada K70

- Pracovní frekvence až 150MHz
- 1024KB Flash paměti s Cache 16KB
- 128KB SRAM
- 4 až 16KB EEPROM
- Až 1x řadič externí paměti podporující NAND Flash a DDR
- 1x Ethernet MAC 10/100Mbps

- 1x USB FS s OTG
- Až 1x Krypto akcelerátor podporující DES, 3DES, AES, MD-5, SHA-1a SHA-256
- 1x řadič pro grafické LCD
- až 32x kanálu DMA
- Až 4x 16-bit ADC, až 48 kanálů
- Až 2x 12-bit DAC
- Až 3x analogový komparátor

### 3.8.2 Řada Kinetis KL

Tato řada je zaměřena na velkou energetickou účinnost a snadné použití. Je proto schopná konkurovat 8/16-bit mikrokontrolerům. V této řadě nalezneme úsporné jádro Cortex-M0+.

#### 3.8.2.1 Řada KL0

- Jádro Cortex-M0+ pracující až na 48MHz
- Až 32KB Flash paměti
- Až 4KB SRAM paměti
- 1x SPI, 1x UART, 1x I<sup>2</sup>C
- 2x časovač
- 4x DMA
- 1x 12-bit ADC
- 2 kanály PWM
- 1x RTC
- rozhraní pro kapacitní senzor
- SWD
- QFN24, QFN32, LQFP32, LQFP48

#### 3.8.2.2 Řada KL1

- Jádro Cortex-M0 pracující až na 48MHz
- Až 64KB Flash paměti
- Až 8KB SRAM paměti
- 2x SPI, 1x UART, 2x I<sup>2</sup>C
- 3x časovač
- 4x DMA
- 1x 12-bit ADC
- 1x analogový komparátor
- 2 kanály PWM
- 1x RTC
- rozhraní pro kapacitní senzor
- SWD
- QFN32, QFN48, LQFP64, LQFP80

#### 3.8.2.3 Řada KL2

- Jádro Cortex-M0 pracující až na 48MHz
- Až 64KB Flash paměti
- Až 8KB SRAM paměti

- 2x SPI, 2x UART, 2x I<sup>2</sup>C
- 1x USB FS s OTG
- 3x časovač
- 4x DMA
- 1x 12-bit ADC
- 1x analogový komparátor
- 2 kanály PWM
- 1x RTC
- rozhraní pro kapacitní senzor
- SWD
- QFN32, LQFP48, LQFP64, LQFP80

### 3.9 Nuvoton

Je výrobce, který se zaměřil pouze na mikrokontroléry s jádrem Cortex-M0, které mají slušnou periferní výbavu. Ve všech řadách nalezneme:

- Jádro Cortex-M0 pracující až na 50MHz
- 32 až 128KB Flash paměti
- 4 až 16KB SRAM paměti
- ISP bootloader
- 3x rozhraní pro Smart Cart
- 4x časovače
- 9x DMA
- 2x12-bit DAC
- 2x analogový komparátor
- 1x RTC
- 16x kanálu kapacitního snímače
- 1x WDT
- Teplotní senzor s rozlišením 1°C
- SWD
- LQFP48, LQFP64, LQFP100, LQFP128

#### 3.9.1 Řada NUC 100 Advanced Line

- 2x SPI, až 2x UART, 2x I<sup>2</sup>C
- 1x 12-bit ADC, až 8 kanálů
- 4 kanály PWM
- QFN33

#### 3.9.2 Řada NUC 110 LCD

- 3x SPI, až 5x UART, 2x I<sup>2</sup>C
- Řadič LCD až 4x40 a 6x38 segmentu
- až 8 kanálů PWM

#### 3.9.3 Řada NUC 120 USB Connectivity

- Až 4x SPI, až 3x UART, 2x I<sup>2</sup>C
- 1x USB FS
- 1x 12-bit ADC, až 8 kanálů

- až 8 kanálů PWM

### 3.9.4 Řada NUC 130 Advanced

- Až 4x SPI, až 3x UART, 2x I<sup>2</sup>C
- 1x USB FS
- 1x CAN 2.0B
- Řadič LCD až 4x40 a 6x38 segmentu
- 1x 12-bit ADC, až 8 kanálů
- až 8 kanálů PWM

### 3.9.5 Řada NUC 140

- Až 4x SPI, až 3x UART, 2x I<sup>2</sup>C
- 1x USB FS
- 1x CAN 2.0B
- 2x LIN
- Řadič LCD až 4x40 a 6x38 segmentu
- 1x 12-bit ADC, až 8 kanálů
- až 8 kanálů PWM

### 3.9.6 Řady NUC 1XXK

- Řadič externí paměti

## 3.10 Holtek

Výrobce vyrábějící ARM mikrokontroléry založené na jádře Cortex-M3 s malou periferní blahostí. Ve výbavě nalezneme:

- Jádro Cortex-M3 pracující až na 72MHz
- 1x 12-bit ADC, až 8 kanálů
- 2x analogový komparátor
- 4 kanály PWM
- 1x WDT
- 1x RTC

Pouze řada HT32F125x

- 8 až 32KB Flash paměti
- 2 až 8 KB SRAM paměti
- 1x SPI, 1x USART, 1x I<sup>2</sup>C
- 2x časovač
- SWD
- LQFP48

Pouze řada HT32F17xx/275x

- 128KB Flash paměti
- až 64KB SRAM paměti
- 2x SPI, 2x USART, 2x I<sup>2</sup>C

- 1x Smart Cart rozhraní
- 5x časovač
- 12x DMA
- 1x USB FS
- 1x rozhraní pro CMOS snímač
- SWD a JTAG
- LQFP48, LQFP64, LQFP100

### 3.11 Srovnání

Přehledně shrneme důležité parametry všech uvedených mikrokontrolérů do přehledné tabulky. Pokud některá řada měla další řady, do tabulky je sloučena do jedné a vypsán max. počet periférií v dané řadě.

řada	frekvence [MHz]	typ jádra	paměť[KB]		konektivita				
			Flash	SRAM	SPI	USART	UART	I2C	Ethernet
STM32F0	48	Cortex-M0	16-64	4-8	2	2	0	2	Ne
STM32F1	72	Cortex-M3	16-1024	4-96	3	5	2	2	Ano
STM32F2	120	Cortex-M3	1024	132	3	4	0	3	Ano
STM32F3	72	Cortex-M4	64-256	48	3	5	5	2	Ne
STM32F4	180	Cortex-M4	2048	260	6	4	4	3	Ano
STM32L1	32	Cortex-M3	384	48	3	5	0	2	Ne
STM32W	24	Cortex-M3	256	16	1	0	1	1	Ne
SAM3A/X	84	Cortex-M3	256-512	32-100	6	4	2	2	Ano
SAM3N	48	Cortex-M3	16-256	2-24	1	2	2	2	Ne
SAM3S	64	Cortex-M3	64-256	16-48	1	2	0	2	Ne
SAM3U	96	Cortex-M3	64-256	16-48	1	4	1	2	Ne
SAM4L	48	Cortex-M4	64-256	32	1	1	3	4	Ne
SAM4E	120	Cortex-M4	1024	128	1	2	2	1	Ano
SAM4S	120	Cortex-M4	2048	160	1	2	2	2	Ne
SAM7S/SE	55	ARM7 TDMI	16-512	4-64	1	2	0	1	Ne
SAM7X/XC	55	ARM7 TDMI	128-512	32-128	2	2	0	1	Ano
SAM7L	36	ARM7 TDMI	64-512	6	1	2	0	1	Ne

<i>SAM9XE</i>	180	<i>ARM926 EJ-S</i>	128- 512	16-32	2	4	1	2	Ano
<i>TM4</i>	100	<i>Cortex- M4</i>	256	32	4	0	8	6	Ne
<i>TMS470</i>	80	<i>Cortex- M3</i>	640	64	2	0	2	0	Ne
<i>LPC800</i>	30	<i>Cortex- M0+</i>	4-16	1-4	2	0	3	1	Ne
<i>LPC11xx</i>	50	<i>Cortex- M0</i>	4- 128	1-12	2	0	1	1	Ne
<i>LPC12xx</i>	30	<i>Cortex- M0</i>	32- 128	4-8	1	0	2	1	Ne
<i>LPC13xx</i>	72	<i>Cortex- M3</i>	8-64	4-12	2	0	1	1	Ne
<i>LPC17xx</i>	120	<i>Cortex- M3</i>	32- 512	8-96	3	0	5	3	Ano
<i>LPC18xx</i>	180	<i>Cortex- M3</i>	512- 1024	104- 200	4	0	4	2	Ano
<i>LPC40xx</i>	120	<i>Cortex- M4</i>	512- 1024	104- 200	4	0	5	3	Ano
<i>LPC43xx</i>	204	<i>Cortex- M4</i>	512- 1024	104- 200	4	0	5	2	Ne
<i>ADuC7000</i>	41,78	<i>ARM7 TDMI</i>	62	8	1	0	1	2	Ne
<i>ADuCM</i>	16	<i>Cortex- M3</i>	128	8	2	0	1	1	Ne
<i>PSoC 4000</i>	48	<i>Cortex- M0</i>	32	4	2	0	2	2	Ne
<i>PSoC 5000LP</i>	67	<i>Cortex- M3</i>	32	64	24	0	24	24	Ne
<i>EFM32ZG</i>	24	<i>Cortex- M0</i>	4-32	2-4	1	1	1	1	Ne
<i>EFM32TG</i>	32	<i>Cortex- M3</i>	4-32	1-4	0	2	1	1	Ne
<i>EFM32G</i>	32	<i>Cortex- M3</i>	16- 128	8-16	0	3	2	1	Ne
<i>EFM32LG</i>	48	<i>Cortex- M3</i>	64- 256	32	0	3	2	2	Ne
<i>EFM32GG</i>	48	<i>Cortex- M3</i>	512- 1024	128	0	3	2	2	Ne
<i>EFM32WG</i>	48	<i>Cortex- M4</i>	64- 256	32	3	2	2	2	Ne
<i>K10</i>	120	<i>Cortex- M4</i>	32- 1024	8-128	5	0	4	2	Ne
<i>K20</i>	120	<i>Cortex- M4</i>	32- 1024	8-128	5	0	4	2	Ne
<i>K30</i>	100	<i>Cortex- M4</i>	64- 512	8-128	5	0	4	2	Ne
<i>K40</i>	100	<i>Cortex- M4</i>	64- 512	16- 128	5	0	4	2	Ne
<i>K50</i>	100	<i>Cortex- M4</i>	128- 512	16- 128	5	0	4	2	Ne
<i>K60</i>	150	<i>Cortex- M4</i>	256- 1024	64- 128	5	0	4	2	Ano

K70	150	Cortex-M4	1024	128	5	0	4	2	Ano
KL0	48	Cortex-M0	32	4	1	0	1	1	Ne
KL1	48	Cortex-M0	64	8	2	0	1	2	Ne
KL2	48	Cortex-M0	64	8	2	0	2	2	Ne
NUC 100	50	Cortex-M0	32-128	4-16	2	0	2	2	Ne
NUC 110	50	Cortex-M0	32-128	4-16	3	0	5	2	Ne
NUC 120	50	Cortex-M0	32-128	4-16	4	0	3	2	Ne
NUC 130	50	Cortex-M0	32-128	4-16	4	0	3	2	Ne
NUC 140	50	Cortex-M0	32-128	4-16	4	0	3	2	Ne
HT32F125x	72	Cortex-M3	8-32	2-8	1	1	0	1	Ne
HT32F17xx /275x	72	Cortex-M3	128	64	2	2	0	2	Ne

Tab. 3 Porovnání ARM mikrokontrolérů část 1.

řada	konektivita		šifrovací akcelerátor	LCD řadič	bootloader	externí paměť	
	USB	CAN				SDIO	SRAM/Flash
STM32F0	0	0	Ne	Ne	Ne	Ne	Ne
STM32F1	1	1	Ne	Ano	Ne	Ano	Ano
STM32F2	1	2	Ano	Ano	Ne	Ano	Ano
STM32F3	1	1	Ne	Ne	Ne	Ne	Ne
STM32F4	1	2	Ano	Ano	Ne	Ano	Ano
STM32L1	1	0	Ne	Ano	Ano	Ano	Ano
STM32W	0	0	Ano	Ne	Ne	Ne	Ne
SAM3A/X	1	2	Ne	Ne	Ano	Ani	Ano
SAM3N	0	0	Ne	Ne	Ano	Ne	Ne
SAM3S	1	0	Ne	Ne	Ano	Ano	Ano
SAM3U	1	0	Ne	Ano	Ano	Ano	Ano
SAM4L	1	0	Ano	Ano	Ne	Ne	Ne
SAM4E	1	2	Ano	Ne	Ne	Ano	Ano
SAM4S	1	0	Ne	Ano	Ano	Ano	Ano
SAM7S/SE	0	0	Ne	Ne	Ano	Ne	Ano
SAM7X/XC	1	1	Ne	Ne	Ne	Ne	Ne
SAM7L	0	0	Ne	Ano	Ne	Ne	Ne
SAM9XE	1	0	Ne	Ne	Ne	Ne	Ano
TM4	1	2	Ne	Ne	Ne	Ne	Ne
TMS470	0	2	Ne	Ne	Ne	Ne	Ne
LPC800	0	0	Ne	Ne	Ano	Ne	Ne
LPC11xx	1	1	Ne	Ano	Ano	Ne	Ne
LPC12xx	0	1	Ne	Ano	Ne	Ne	Ne
LPC13xx	1	0	Ne	Ne	Ano	Ne	Ne
LPC17xx	1	1	Ne	Ano	Ne	Ano	Ano
LPC18xx	2	2	Ne	Ano	Ne	Ano	Ano
LPC40xx	1	1	Ne	Ano	Ne	Ano	Ano



LPC43xx	2	2	Ne	Ano	Ne	Ne	Ne
ADuC7000	0	0	Ne	Ne	Ne	Ne	Ano
ADuCM	0	0	Ne	Ne	Ne	Ne	Ne
PSoC 4000	0	0	Ne	Ne	Ne	Ne	Ne
PSoC 5000LP	1	1	Ne	Ano	Ne	Ne	Ne
EFM32ZG	0	0	Ano	Ne	Ne	Ne	Ne
EFM32TG	0	0	Ano	Ano	Ne	Ne	Ne
EFM32G	0	0	Ano	Ano	Ne	Ne	Ano
EFM32LG	1	0	Ano	Ano	Ne	Ne	Ano
EFM32GG	1	0	Ano	Ano	Ne	Ne	Ano
EFM32WG	1	0	Ano	Ano	Ne	Ne	Ano
K10	0	2	Ano	Ne	Ne	Ne	Ano
K20	1	2	Ano	Ne	Ne	Ano	Ano
K30	0	2	Ne	Ano	Ne	Ano	Ano
K40	1	2	Ne	Ano	Ne	Ano	Ano
K50	1	2	Ano	Ano	Ne	Ano	Ano
K60	1	2	Ano	Ne	Ne	Ano	Ano
K70	1	2	Ano	Ano	Ne	Ano	Ano
KL0	0	0	Ne	Ne	Ne	Ne	Ne
KL1	0	0	Ne	Ne	Ne	Ne	Ne
KL2	1	0	Ne	Ne	Ne	Ne	Ne
NUC 100	0	0	Ne	Ne	Ano	Ne	Ano
NUC 110	0	0	Ne	Ano	Ano	Ne	Ano
NUC 120	1	0	Ne	Ne	Ano	Ne	Ano
NUC 130	1	1	Ne	Ano	Ano	Ne	Ano
NUC 140	1	1	Ne	Ano	Ano	Ne	Ano
HT32F125x	0	0	Ne	Ne	Ne	Ne	Ne
HT32F17xx /275x	1	0	Ne	Ne	Ne	Ne	Ne

Tab. 4 Porovnání ARM mikrokontrolérů část 2.

řada	periferie						
	časovače	DMA	ADC/kanály	DAC	komparátor	RTC	PWM
STM32F0	11	5	1/16	0	2	1	6
STM32F1	16	12	1/16	2	0	1	6
STM32F2	17	16	3/16	2	0	1	4
STM32F3	13	12	4/39	2	7	1	6
STM32F4	17	16	3/24	2	0	1	4
STM32L1	11	12	1/40	1	2	1	0
STM32W	2	6	1/6	0	0	1	0
SAM3A/X	9	17	1/16	1	0	1	8
SAM3N	6	10	1	1	0	1	4
SAM3S	6	22	1/16	1	0	1	4
SAM3U	3	17	2/16	0	0	1	4
SAM4L	6	16	1/16	1	0	1	0
SAM4E	3	33	2/24	1	1	1	4
SAM4S	2	22	1/16	1	1	1	4
SAM7S/SE	3	11	1/8	0	0	1	4
SAM7X/XC	3	17	1/8	0	0	1	4
SAM7L	3	11	1/4	0	0	1	4
SAM9XE	6	24	1/4	0	0	1	2
TM4	12	32	2/24	0	3	0	16

TMS470	18	0	1/16	0	0	0	0
LPC800	4	0	0	0	0	0	4
LPC11xx	4	0	1/8	0	0	0	13
LPC12xx	4	21	1/8	0	0	1	13
LPC13xx	4	0	1/8	0	0	0	11
LPC17xx	4	8	1/8	1	0	1	6
LPC18xx	4	8	1/8	1	0	1	2
LPC40xx	4	8	1/8	1	2	1	2
LPC43xx	4	8	1/8	1	0	1	2
ADuC7000	4	0	2/2	0	0	0	1
ADuCM	2	11	2/11	1	0	0	1
PSoC 4000	4	0	1/8	2	2	0	0
PSoC 5000LP	4	24	2/2	4	4	0	4
EFM32ZG	4	4	1/4	0	1	1	6
EFM32TG	4	8	1/8	1	2	1	6
EFM32G	8	8	1/8	1	2	1	9
EFM32LG	7	12	1/8	1	2	1	12
EFM32GG	7	12	1/8	1	2	1	12
EFM32WG	9	12	1/8	1	2	1	12
K10	4	16	2/46	2	3	1	8
K20	4	32	4/56	2	4	1	8
K30	4	16	2/27	1	4	1	8
K40	4	16	2/27	2	4	1	8
K50	4	16	2/25	2	3	1	8
K60	4	32	4/48	2	3	1	8
K70	4	32	4/48	2	3	1	8
KL0	2	4	1/1	0	0	1	2
KL1	3	4	1/1	0	1	1	2
KL2	3	4	1/1	0	1	1	2
NUC 100	4	9	1/8	2	2	1	4
NUC 110	4	9	0	2	2	1	8
NUC 120	4	9	1/8	2	2	1	8
NUC 130	4	9	1/8	2	2	1	8
NUC 140	4	9	1/8	2	2	1	8
HT32F125x	2	0	1/8	2	2	1	4
HT32F17xx /275x	5	9	1/8	2	2	1	4

Tab. 5 Porovnání ARM mikrokontrolérů část 3.

řada	WDT	debug
STM32F0	1	SWD
STM32F1	1	SWD a JTAG
STM32F2	1	SWD a JTAG
STM32F3	1	SWD a JTAG
STM32F4	1	SWD a JTAG
STM32L1	1	SWD a JTAG
STM32W	1	SWD a JTAG
SAM3A/X	1	SWD a JTAG
SAM3N	1	SWD a JTAG
SAM3S	1	SWD a JTAG
SAM3U	1	SWD a JTAG
SAM4L	1	SWD a JTAG

SAM4E	1	SWD a JTAG
SAM4S	1	SWD a JTAG
SAM7S/SE	1	JTAG
SAM7X/XE	1	JTAG
SAM7L	1	JTAG
SAM9XE	1	JTAG
TM4	2	SWD a JTAG
TMS470	0	JTAG
LPC800	1	SWD a JTAG
LPC11xx	1	SWD a JTAG
LPC12xx	1	SWD
LPC13xx	1	SWD a JTAG
LPC17xx	1	SWD a JTAG
LPC18xx	1	SWD a JTAG
LPC40xx	1	SWD a JTAG
LPC43xx	1	SWD a JTAG
ADuC7000	0	JTAG
ADuC	0	SWD
PSoC 4000	0	SWD a JTAG
PSoC 5000LP	0	SWD a JTAG
EFM32ZG	1	SWD
EFM32TG	1	SWD
EFM32G	1	SWD
EFM32LG	1	SWD
EFM32GG	1	SWD
EFM32WG	1	SWD
K10	1	SWD a JTAG
K20	1	SWD a JTAG
K30	1	SWD a JTAG
K40	1	SWD a JTAG
K50	1	SWD a JTAG
K60	1	SWD a JTAG
K70	1	SWD a JTAG
KL0	0	SWD
KL1	0	SWD
KL2	0	SWD
NUC 100	1	SWD
NUC 110	1	SWD
NUC 120	1	SWD
NUC 130	1	SWD
NUC 140	1	SWD
HT32F125x	1	SWD
HT32F17xx /275x	1	SWD a JTAG

Tab. 6 Porovnání ARM mikrokontrolérů část 4.

## 4 Dostupná vývojová prostředí

V tomto bodě se zaměříme na vývojová prostředí pro vývoj na ARM mikrokontroléry a provedeme detailní srovnání vlastností volně dostupných, ale i komerčních z hlediska podporovaných ARM mikrokontrolérů, velikosti výsledného kódu, rychlosti kódu.

### 4.1 Keil uVision4

Keil uVision4 je oficiální prostředí společnosti ARM, které je plně vybaveno. Jde o komerční produkt, který ve volné verzi omezuje velikost výsledného kódu na hodnotu 32KB. Pro jednodušší projekty tato hodnota postačuje. Práce s Keil u Vision je doporučena zkušenějším uživatelům, protože zprovoznění projektu vyžaduje hlubší znalosti s prací ARM mikrokontroléry, vložení potřebných knihoven a správné nastavení inicializace mikrokontrolérů spojené se startupem napsaného v assembleru. Nalezneme zde podporu 36 mikrokontrolérů a kitu. Jádra jsou podporována od ARM7TDMI až po Cortex-R4. Pro některé kity zde nalezneme ukázkové příklady. V Keil uVision4 nalezneme dobrou podporu programovacích a ladicích prostředků (debuggeru), konkrétně:

- ULINK2/ME Cortex Debugger
- RDI Interface Driver
- Altera Blaster Cortex Debugger
- Stellaris ICDI
- Signum Systems JTAGjet
- J-LINK / J-Trace Cortex
- ST-Link (Deprecated Version)
- ULINK Pro Cortex Debugger
- NULink Debugger
- SiLabs UDA Debugger
- ST-Link Debugger
- CMSIS-DAP Debugger
- Fast Models Debugger

Práce s debuggrem je přehledná a rychlá. Můžeme krokovat program po jednotlivých řádcích napsaného v jazyce C (programovací jazyk) nebo v assembleru, který vznikne při zkompileování projektu. Také tu máme možnost vložit break points (bod nebo body, na kterém se pozastaví ladění, když běží program) a nechat si zastavit ladění v bodech, které potřebujeme. Můžeme detailně zobrazovat nebo měnit obsah registru v jádru, v perifériích, obsah paměti SRAM případně obsah EEPROM. Další informací je, v jakém módu procesor pracuje. Další užitečnou věcí je měření času vykonání kódu, které se hodí pro odladění časově náročného kódu. Nalezneme zde také možnost použít RTX (deterministický Real-Time operační systém určený pro ARM Cortex-M mikrokontroléry), který nám pomůže řešit např. běh několika vláken (podprogramů) a další věci spojené s operačním systémem.

### 4.2 Coocox

Coocox je volně dostupné vývojové prostředí vhodné pro začátečníky. Projekty se snadno vytváří a nastavují, protože má přehledné a jednoduché nastavení. Při vytváření projektu se vyberou potřebné knihovny, které budou následně vloženy do projektu, není tedy nutné znát podrobně registry mikrokontrolérů, jenom několik funkcí. Můžeme také využít přímo ovladače pro periférie dodávané s Coocoxem. Starup není napsán v assembleru jako v předchozím prostředí, ale je napsán v jazyce C a lze ho tedy snadno upravovat bez znalosti

assembleru. V pravé části prostředí se po vybrání potřebné periferie objeví detailní nápověda, jak danou periferii nastavit a obsluhovat. Při psaní kódu v jazyce C se vhodně zvýrazňují klíčová slova a podmíněný kód (část nastaveného kódu, která v době překladu je podle podmínky vybrána nebo zahozena), který zbarví do šeda, když podle podmínky nebude použit. Pro kompilaci a ladění je nutné doinstalovat CoDebugger, jde o vlastní kompilátor jazyka C a nastavit ve vývojovém prostředí cestu k tomuto kompilátoru. Podpora ARM mikrokontrolérů je omezena na jádra Cortex-M0, Cortex-M+, Cortex-M3 a Cortex-M4. Mikrokontroléry jsou podporované jenom některé, konkrétněji:

- STMicroelectronic – řady: STM32F0, STM32F1 a STM32F4
- Atmel – řada: SAM3U
- Energy Micro - řada: EFM32G
- Holtek – řady: HT32F125, HT32F175, HT32F275
- Texas instruments – řada: LM3S
- NXP – řady: LPC11xx, LPC12xx, LPC13xx, LPC17xx
- Novoton – řady: NUC100 až NUC140, Nano100 až Nano140, NucMicroM051, NucMicroMini51

Podporované mikrokontroléry jsou většinou ty nejdostupnější na trhu. Podpora debuggeru je na tom lépe, konkrétněji:

- Colink
- ColinEx
- CMSIS-DAP
- ARLinkEx
- CoiNel-USB-JTAG
- CoiNet-ColinkEx
- e-Link32
- Nu-Link
- Olimex-OpenOCD
- Stellaris-ICDI
- KT-Link
- NGX-USB-JTAG
- picoJTAG
- OpenJTAG
- ST-Link
- Amontec-JTAGkey
- J-Link

Debuggeru v tomto vývojovém prostředí chybí funkce pro měření času vykonávaného kódu. Funkce pro nastavení a prohlížení obsahu registrů, periferii, proměnných a paměti SRAM je v debuggeru přehledně obsažena. Samozřejmostí je možnost vkládat break pointy jak do kódu jazyka C, tak i do výsledného assembleru. Vložení více jak jednoho break pointu do kódu assembleru způsobuje problémy se správným zastavením kódu v tomto bodě. V kódu v jazyce C break pointy fungují správně. Může se také stát, že při použití ST-Linku nepůjde nahrát program do mikrokontrolérů, protože ho nemůže v systému nalézt. Lze to vyřešit tak, že nainstalujeme ST-link Utility a z tohoto programu zkopírujeme knihovnu STLinkUSBdriver.dll do složky bin v adresáři, kde je nainstalován Coocox (složka CoIDE) a jeho stávající knihovnu nahradíme.

### 4.3 AVR Studio 6.0

AVR Studio je vývojové prostředí původně určeno pro AVR mikrokontroléry, ale s příchodem verze 6 přináší podporu pouze mikrokontroléry od Atmelu SAM3 a SAM4. Přesto, že ho společnost sama vyvíjí pro svoje mikrokontroléry, je volně dostupné s plnou sadou knihoven. Horší situace je s debuggery, který pro ARM mikrokontroléry podporuje pouze SAM-ICE.

### 4.4 Raisonance

Je volně dostupné vývojové prostředí, které vyžaduje registraci, jinak do 30 dnů dojde k zablokování. Do verze 7.30 a verze kitu 1.3 toto omezení nebylo, pouze na velikost výsledného zkompilovaného kódu. Vytvoření projektu je obtížnější, knihovny a kód musíme do projektu vložit ručně. Nalezneme zde v pravé části vývojového prostředí datasheety podporovaných mikrokontrolérů. Podpora debuggeru je omezena pouze na RLink. Od jiných prostředí se odlišuje svým simulátorem ARM mikrokontrolérů, který je přehledný, umožňuje krokovat program jak v kódu jazyka C, tak i v assembleru. Také umožňuje spustit běh programu a pozastavit ho na break pointy, které mohou mít jak v kódu C, tak ve výsledném kódu v assembleru. V simulátoru můžeme přehledně zobrazovat nebo měnit stav registru jádra procesoru, registry periférií a obsah paměti SRAM. Podpora mikrokontrolérů není příliš velká a je omezena na:

- STMicroelectronic – všechny řady
- Texas instruments – řada: LM3S
- NXP – řady: LPC13xx, LPC17xx, LPC2xx

### 4.5 Atollic

Atollic je komerční produkt, který nabízí i volnou verzi s omezením výsledného kódu na velikost 32KB. Poměrně snadno se zakládá projekt, všechny potřebné knihovny jsou při vytvoření vloženy do projektu. Nalezneme poměrně bohatou sadu knihoven. Při psaní kódu se vhodně vyznačují vyhrazená slova jazyka C a část kódu, který má podmíněný překlad, zešedne v případě, že do výsledného kódu nebude vkládán. Podpora debuggeru není příliš velká, máme možnost ji rozšířit, konkrétněji:

- Segger J-Link
- STMicroelectronics ST-LINK a ST-LINK/V2
- Atmel SAM-ICE™
- OSJTAG a P&E Multilink
- Other gdbserver compatible JTAG probes

Atollic se od ostatních liší větší podporou ARM jader od ARM7TDMI, po Cortex-M až po Cortex-A15 a dvou procesorové. Podpora mikrokontrolérů je menší, konkrétněji:

- Atmel – řady: SAM3S a SAM9
- Energy Micro – řady: EFM32G, EFM32GG, EFM32LG, EFM32TG, EFM32WG
- Freescale – řady: K10 až K60, KL04 až KL25, VF60 až VF65,
- Fujitsu – řada: MB9A/B
- Infineon – řady: XMC1000 až XMC4504
- NXP – řady: LPC11xx, LPC12xx, LPC13xx, LPC17xx, LPC18xx, LPC40xx, LPC43xx, LPC800
- STMicroelectronic – všechny řady

- Texas instruments – řada: LM3S
- Toshiba – řady: TMPA9xx, TMPM3xx

K těmto mikrokontrolérům jsou také podporovány všechny kity.

## 4.6 IAR

IAR je komerční vývojové prostředí, které nabízí i volnou verzi s omezením velikosti zkompilevaného kódu na 32KB. Je poměrně dobře vybaven na knihovny a má podporu 30 ARM mikrokontrolérů. Vytvoření projektu obtížnější, je třeba provést po vytvoření projektu konkrétní nastavení, jinak nepůjde zkompilevat kód. Na rozdíl od ostatních má v sobě integrován simulátor. Podpora debuggeru je poměrně velká, konkrétně:

- Angel
- CMSIS DAP
- GDB Server
- IAR ROM-monitor
- I-jet/JTAGjet
- J-Link/J-Trace
- TI Stellaris
- Macraigor
- PE micro
- RDI
- ST-Link
- Thirf-Party Driver
- TI XDS100

V simulátoru nebo debuggeru má podobné vlastnosti jako ostatní, opět chybí možnost změřit čas vykonávaného kódu.

## 4.7 Možnosti vývoje na Linuxu

### 4.7.1 CodeSourcery

Je kompilátor, který ve verzi lite obsahuje řádkový překladač jazyka C nebo C++ pro zkompilevání kódu pro ARM mikrokontroléry. V placené verzi je k tomuto kompilátoru dodáváno vývojové prostředí. Tento kompilátor lze provozovat nejen na operačních systémech Windows, ale je také dostupná verze pro Linux. Můžeme také tento kompilátor propojit s volně dostupným vývojovým prostředím např. Code::Blocks, které je také dostupné pro operační systém Linux. Knihovny pro konkrétní ARM mikrokontroléry je potřeba doinstalovat. Více viz. [28].

### 4.7.2 ST-Link pro Linux

Podpora ST-Linku pro Linux dlouho nebyla až do nedávna. Tímto programem je *texane*. Instalace toho programu není úplně snadná a není vhodná pro začátečníka. Nejprve je nutné nainstalovat knihovny příkazem `sudo apt-get install libusb-1.0-0-dev git` a pak stáhnou zdrojové kódy příkazem `git clone https://github.com/texane/stlink stlink.git`, po stažení přepnout adresář příkazem `cd stlink.git` a poté spustit kompilaci příkazem `make`. Po úspěšné kompilaci se nainstaluje program `st-flash` příkazy `cd flash` a `sudo cp st-link /usr/bin`. Dále

nastavíme, aby bylo možné spustit st-flash bez příkazu sudo příkazy *cd.* a *sudo cp \*.rules /etc/udev/rules.d* a posledním příkazem *sudo restart udev* a restartujeme. Více viz. [27].

#### 4.8 Srovnání vývojových prostředí

V tomto bodě srovnáme předchozí uvedená prostředí: Keil uVision4, Keil uVision4 speed (optimalizace na rychlost), Coccox, AVR Studio 6.0 a Atollic z hlediska velikosti zkompilevaného kódu a rychlosti kódu hlavní smyčky testované na STM32VLDISCOVERY kitu (jádro Cortex-M3). Pro změření rychlosti kódu je použito nahazování a shazování bitu na výstupní bráně GPIOC pin PC8 a tento signál (vstup SIGNAL) je zaveden do jednoduchého přípravku na odměření rychlosti kódu tvořeného mikrokontrolérem AVR mega8 (je využita funkce Capture pro přesné odečtení hodnoty časovače) a výsledek zobrazován na znakovém LCD 20x4 znaků (program a schéma v příloze B a C). Zároveň jsou také propojeny hodiny kitu s přípravkem (vstup CLOCK). Použitý testovací kód pro ARM mikrokontrolér nalezneme příloze A. Vývojové prostředí AVR studio bylo otestováno jenom na velikost výsledného kódu pro jádro Cortex-M3, protože nemá podporu použitého kitu.

##### 4.8.1 Přehledné tabulky velikosti zkompilevaného kódu

V tabulkách Tab. 7 až Tab. 14 jsou naměřené výsledky velikosti zkompilevaného kódu z přílohy A. V místě, kde je pomlčka, znamená, že tato optimalizace není dostupná.

Vývojové prostředí	Optimalizace a velikost kódu v bytech				
	O0	O1	O2	O3	Os
Keil uVision 4	588	552	522	540	-
Keil uVision 4 speed	588	556	660	544	-
Coccox	980	692	700	596	616
AVR Studio 6.0	1264	1112	992	880	960
Raisonance	1224	1024	968	808	996
Atollic	1312	1044	984	860	976

Tab. 7 velikost zkompilevaného kódu pro matice 2x2 typu int uložené jako lokální proměnné

Vývojové prostředí	Optimalizace a velikost kódu v bytech				
	O0	O1	O2	O3	Os
Keil uVision 4	588	552	660	540	-
Keil uVision 4 speed	588	556	660	936	-
Coccox	1012	728	1452	600	672
AVR Studio 6.0	1296	1152	1052	880	1020
Raisonance	1256	1044	1692	812	996
Atollic	1344	1548	1492	868	996

Tab. 8 velikost zkompilevaného kódu pro matice 4x4 typu int uložené jako lokální proměnné

Vývojové prostředí	Optimalizace a velikost kódu v bytech				
	O0	O1	O2	O3	Os
Keil uVision 4	1448	1408	749	741	-



Keil uVision 4 speed	1448	1412	1516	1436	-
Coocox	2204	1940	1992	596	1860
AVR Studio 6.0	2500	2396	2280	880	2252
Raisonance	2448	2300	2253	808	2292
Atollic	2540	2332	2276	860	2276

Tab. 9 velikost zkompilovaného kódu pro matice 2x2 typu float uložené jako lokální proměnné

Vývojové prostředí	Optimalizace a velikost kódu v bytech				
	O0	O1	O2	O3	Os
Keil uVision 4	1448	1408	1416	1404	-
Keil uVision 4 speed	1448	1412	1516	2064	-
Coocox	2236	1960	3276	600	1908
AVR Studio 6.0	2532	2408	2312	880	2272
Raisonance	2480	2276	3408	812	2244
Atollic	2572	3372	3288	868	2232

Tab. 10 velikost zkompilovaného kódu pro matice 4x4 typu float uložené jako lokální proměnné

Vývojové prostředí	Optimalizace a velikost kódu v bytech				
	O0	O1	O2	O3	Os
Keil uVision 4	592	564	564	556	-
Keil uVision 4 speed	592	564	664	580	-
Coocox	972	700	664	636	632
AVR Studio 6.0	1252	1116	1008	880	972
Raisonance	1208	992	940	844	968
Atollic	1304	1052	992	896	980

Tab. 11 velikost zkompilovaného kódu pro matice 2x2 typu int uložené jako globální proměnné

Vývojové prostředí	Optimalizace a velikost kódu v bytech				
	O0	O1	O2	O3	Os
Keil uVision 4	592	564	564	556	-
Keil uVision 4 speed	592	564	664	924	-
Coocox	972	752	1232	656	680
AVR Studio 6.0	1252	1164	1056	880	1024
Raisonance	1208	1048	1592	872	996
Atollic	1304	1520	1592	936	996

Tab. 12 velikost zkompilovaného kódu pro matice 4x4 typu int uložené jako globální proměnné

Vývojové prostředí	Optimalizace a velikost kódu v bytech				
	O0	O1	O2	O3	Os
Keil uVision 4	1452	1412	1424	1416	-
Keil uVision 4 speed	1452	1420	1528	1532	-
Coocox	2196	1956	2000	640	1960

<i>AVR Studio 6.0</i>	2488	2416	2336	880	2300
<i>Raisonance</i>	2432	2288	2268	848	2300
<i>Atollic</i>	2532	2352	2320	904	2308

Tab. 13 velikost zkompilevaného kódu pro matice 2x2 typu float uložené jako globální proměnné

Vývojové prostředí	Optimalizace a velikost kódu v bytech				
	O0	O1	O2	O3	Os
<i>Keil uVision 4</i>	1452	1412	1424	1416	-
<i>Keil uVision 4 speed</i>	1452	1420	1528	2112	-
<i>Coocox</i>	2196	1984	3036	664	1944
<i>AVR Studio 6.0</i>	2488	2420	2308	880	2264
<i>Raisonance</i>	2432	2268	2352	867	2228
<i>Atollic</i>	2532	3384	3308	940	2220

Tab. 14 velikost zkompilevaného kódu pro matice 4x4 typu float uložené jako globální proměnné

Z výsledku je patrné, že Keil uVision 4 generuje ze všech vývojových prostředí v optimalizaci O2 nejmenší funkční kód, naopak největší funkční kód generují přibližně stejně vývojová prostředí AVR Studio 6.0, Raisonance a Atollic.

#### 4.8.2 Přehledné tabulky rychlosti kódu

V tabulkách Tab. 15 až Tab. 22 jsou naměřené výsledky rychlosti zkompilevaného kódu z přílohy A. V místě, kde je pomlčka, znamená, že tato optimalizace není dostupná. Znakem x je označeno špatná kompilace, kdy do smyčky byl vložen jenom kód pro nahazování a shazování bitu na výstupní bráně.

Vývojové prostředí	Optimalizace a rychlost kódu v strojových cyklech				
	O0	O1	O2	O3	Os
<i>Keil uVision 4</i>	339	249	221	215	-
<i>Keil uVision 4 speed</i>	341	244	210	x	-
<i>Coocox</i>	694	122	95	x	126
<i>Raisonance</i>	712	116	96	x	112
<i>Atollic</i>	700	107	87	x	100

Tab. 15 rychlost zkompilevaného kódu pro matice 2x2 typu int uložené jako lokální proměnné

Vývojové prostředí	Optimalizace a rychlost kódu v strojových cyklech				
	O0	O1	O2	O3	Os
<i>Keil uVision 4</i>	1577	1213	1107	1101	-
<i>Keil uVision 4 speed</i>	1579	1198	1118	337	-
<i>Coocox</i>	3489	1131	449	x	968
<i>Raisonance</i>	3590	1008	483	x	1104
<i>Atollic</i>	3520	458	440	x	998

Tab. 16 rychlost zkompilevaného kódu pro matice 4x4 typu int uložené jako lokální proměnné

Vývojové prostředí	Optimalizace a rychlost kódu v strojových cyklech				
	O0	O1	O2	O3	Os
Keil uVision 4	874	766	749	741	-
Keil uVision 4 speed	876	772	733	x	-
Coocox	1240	674	626	x	677
Raisonance	1251	645	623	x	654
Atollic	1244	633	611	x	623

Tab. 17 rychlost zkopilovaného kódu pro matice 2x2 typu float uložené jako lokální proměnné

Vývojové prostředí	Optimalizace a rychlost kódu v strojových cyklech				
	O0	O1	O2	O3	Os
Keil uVision 4	5828	5324	5293	5275	-
Keil uVision 4 speed	5830	5384	5289	4579	-
Coocox	8773	6199	5721	x	6350
Raisonance	8820	6151	5675	x	6218
Atollic	8789	5570	5590	x	6177

Tab. 18 rychlost zkopilovaného kódu pro matice 4x4 typu float uložené jako lokální proměnné

Vývojové prostředí	Optimalizace a rychlost kódu v strojových cyklech				
	O0	O1	O2	O3	Os
Keil uVision 4	379	564	228	220	-
Keil uVision 4 speed	381	245	207	x	-
Coocox	692	116	58	x	104
Raisonance	697	98	78	x	108
Atollic	700	104	78	x	94

Tab. 19 rychlost zkopilovaného kódu pro matice 2x2 typu int uložené jako globální proměnné

Vývojové prostředí	Optimalizace a rychlost kódu v strojových cyklech				
	O0	O1	O2	O3	Os
Keil uVision 4	1889	1165	1168	1118	-
Keil uVision 4 speed	1891	1199	1149	248	-
Coocox	3529	1048	414	x	529
Raisonance	3513	953	502	x	1306
Atollic	3561	424	451	x	1220

Tab. 20 rychlost zkopilovaného kódu pro matice 4x4 typu int uložené jako globální proměnné

Vývojové prostředí	Optimalizace a rychlost kódu v strojových cyklech				
	O0	O1	O2	O3	Os
Keil uVision 4	907	778	758	750	-
Keil uVision 4 speed	907	764	737	x	-
Coocox	1244	646	597	x	646
Raisonance	1243	636	619	x	646

Atollic	1244	638	621	x	634
---------	------	-----	-----	---	-----

Tab. 21 rychlost zkopilovaného kódu pro matice 2x2 typu float uložené jako globální proměnné

Vývojové prostředí	Optimalizace a rychlost kódu v strojových cyklech				
	O0	O1	O2	O3	Os
Keil uVision 4	6119	5425	5382	5332	-
Keil uVision 4 speed	6121	5322	5361	4633	-
Coocox	8828	6268	5564	x	5781
Raisonance	8816	6105	5594	x	6414
Atollic	8828	5597	5597	x	6394

Tab. 22 rychlost zkopilovaného kódu pro matice 4x4 typu float uložené jako globální proměnné

Nejrychlejší funkční kód optimalizace O2 generují vývojová prostředí Coocox a Atollic, naopak nejpomalejší funkční kód generuje Keil uVision4

## 5 Závěr

Cílem bakalářské práce je stručně seznámit uživatele s ARM mikrokontroléry, usnadnit výběr ARM mikrokontrolérů dostupné na trhu a dostupného vývojového prostředí.

Nejprve se stručně uvedly vlastnosti ARM mikrokontrolérů z hlediska rodin. Rodiny ARM7TDMI a ARM926EJ-S jsou sice na trhu stále dostupná, jde však o starší verzi, a proto nenalezneme moc výrobců, kteří by je vyráběli. Rodina Cortex je naopak nejnovější a je mezi výrobci hodně rozšířena, integruje do svého jádra další periferie jako je časovač SysTick a nebo vnořený vektorový řadič přerušení. Cortex-M0+ a vyšší mají v jádru integrovanou jednotku ochrany paměti, která např. v operačním systému chrání Flash paměť před neoprávněným přepsáním. Dále se porovnaly vlastnosti z hlediska použitých instrukčních sad. Rodiny ARM7TDMI a ARM926EJ-S používají instrukční sady ARM a Thumb, které mají problematické vkládání konstant, která má velikost pouze 8bit. Daleko lépe je vyřešena instrukční sada Thumb2, která používá jak 16bit instrukce, tak i 32bit, což vede ve většině případů k menšímu kódu než by byl na instrukční sadě ARM, ale zároveň má téměř stejný výkon. Vkládání konstant je lépe vyřešeno dvojicí instrukcí, které nahrávají konstantu o velikosti 16bitů, jedna spodních 16bitů a druhá horních 16bitů, takže je možné pomocí dvou instrukcí nahrát do registru např. ukazatel.

Další bod je zaměřen na vytvoření seznamu dostupných ARM mikrokontrolérů a porovnat jejich vlastnosti. Ze seznamu je vidět, že STMicroelectronics má v nabídce největší výběr mikrokontrolérů a nabízí všechna Cortex-M, dobrou periferní vybavu, velkou Flash a SRAM paměť. Jejich nejnížší řada STM32F0 má na trhu podobnou cenu jako 8-bit mikrokontrolér se stejně velkou Flash, ale vlastnosti jsou lepší. Atmel se omezil u rodiny Cortex na Cortex-M3 a Cortex-M4. Vlastnosti jsou podobné jako u STMicroelectronics. Mikrokontroléry od Texas Instruments jsou zaměřeny na řízení a měření, proto u nich nalezneme větší počet kanálů ADC, větší počet kanálů PWM a větší počet časovačů. NXP a jeho nejnížší řady patří vůbec k nejlevnějším ARM mikrokontrolérům a jsou s cenou na úrovni levných 8-bit mikrokontrolérů. Vybava nejnížší řady není moc bohatá, je spíše na úrovni 8-bit mikrokontrolérů. Vyšší řady mají daleko lepší vybavu a je u nich použito výkonnější jádro. Analog Devices má ARM mikrokontroléry vybavené přesnějšími ADC převodníky sigma-delta a hodí se na přesné měření analogových hodnot. Cypress se odlišuje od ostatních ARM mikrokontrolérů integrovaným programovatelným hradlovým polem.

Enegr micro mají slušnou výbavu, jejich řady jsou vhodně odstupňovány. Freescale nabízí řady, které mají jádro Cortex-M4 a nebo jádro Cortex-M0+. Řada Kinetis K s Cortex-M4 mají poměrně bohatou periferní výbavu, naopak řada Kinetis KL je zaměřena na přímou konkurenci s 8-bit mikrokontroléry. Řady ARM mikrokontroléry od Nuvoton jsou dost mezi sebou podobné, liší se minimálně, řada s písmenem K v označení řady má možnost připojit externí paměť. Podobné to je s Holtek, řady jsou podobné a neobsahují moc bohatou periferní výbavu.

Poslední částí je porovnání vlastnosti vývojových prostředí a porovnat rozdíl velikosti a rychlosti zkompilevaného kódu. Vývojové prostředí Keil uVision4 je oficiálně podporován společností ARM a má tedy nejlepší podporu mikrokontrolérů. Má také velmi dobrou podporu debuggeru a obsahuje ukázkové příklady ke koum. Nevýhodou může být ve volné max. velikost výsledného kódu na hodnotě 32KB. Coocox naopak omezení velikosti kódu nemá a je volně dostupný. Má pěkné a jednoduché nastavení projektu, ST-Link s ním měl problémy, proto bylo třeba do něj zkopírovat novější knihovnu STLinkUSBDriver.dll z ST-Link Utility. Dalším problémem, který je na této verzi, je nesprávné fungování break pointu v assembleru, kdy více než jeden break point způsobuje při spuštění programu vynechávání těchto nastavených break pointu. AVR Studio 6.0 je slušné prostředí, které bylo původně pro AVR mikrokontroléry, ale tato verze má možnost kompilovat i pro ARM mikrokontroléry, ale pouze od Atmel. Výhodou je, že toto prostředí je volně dostupné, ale nevýhoda je malý počet podporovaných ARM mikrokontrolérů a podporovaný pouze jeden debugger. Raisonance je ve volné verzi poměrně slušné prostředí, které obsahuje i simulátor ARM mikrokontrolérů. Verze, které přišli po verzi 7.30, vyžadují zakoupení licence nebo programátoru RLink, jinak do 30 dnů dojde k zablokování. Atollic je dobré prostředí, podporuje dostatečné množství ARM mikrokontrolérů a všechny ARM jádra začínající od ARM7TDMI. Nevýhodou může být u ST-Linku mít v něm nejnovější verzi, jinak odmítá programovat a debugerovat. Z porovnání velikosti zkompilevaného kódu je na tom nejlíp Keil uVision4, ale AVR Studio 6.0, Raisonance a Atollic generují největší kód. Coocox, který je volně dostupný a bez omezení, měl lepší kompilaci na velikost kódu než komerční omezené verze vývojových prostředí Raisonance a Atollic. Nejrychlejší kód generuje v optimalizace O2 Coocox, ale Keil uVision4 optimalizace speed a Atollic si také vede slušně. Pokud v Keil uVision4 vypneme optimalizaci na rychlost, generuje nejpomalejší kód.

## 6 Použitá literatura

- [1] STMicroelectronics, [online]. [cit.2013-06-03]. Dostupné z: <http://www.st.com>
- [2] Atmel, [online]. [cit.2013-06-03]. Dostupné z: <http://www.atmel.com>
- [3] ARM, [online]. [cit.2013-06-03]. Dostupné z: <http://www.arm.com>
- [4] NXP, [online]. [cit.2013-06-03]. Dostupné z: <http://www.nxp.com>
- [5] Freescale, [online]. [cit.2013-06-03]. Dostupné z: <http://www.freescale.com>
- [6] Texas Instruments, [online]. [cit.2013-06-03]. Dostupné z: <http://www.ti.com>
- [7] Analog Devices, [online]. [cit.2013-06-03]. Dostupné z: <http://www.analog.com>
- [8] Cypress, [online]. [cit.2013-06-03]. Dostupné z: <http://www.cypress.com>
- [9] Energy micro, [online]. [cit.2013-06-03]. Dostupné z: <http://www.energymicro.com>
- [10] ARM7TDMI, [online]. [cit.2013-06-03]. Dostupné z: <http://infocenter.arm.com/help/topic/com.arm.doc.ddi0210c/DDI0210B.pdf>
- [11] ARM926EJ-S, [online]. [cit.2013-06-03] Dostupné z: [http://infocenter.arm.com/help/topic/com.arm.doc.ddi0198e/DDI0198E\\_arm926ejs\\_r0p5\\_trm.pdf](http://infocenter.arm.com/help/topic/com.arm.doc.ddi0198e/DDI0198E_arm926ejs_r0p5_trm.pdf)

- [12] Cortex-M0, [online]. [cit.2013-06-03] Dostupné z: [http://infocenter.arm.com/help/topic/com.arm.doc.dui0497a/DUI0497A\\_cortex\\_m0\\_r0\\_p0\\_generic\\_ug.pdf](http://infocenter.arm.com/help/topic/com.arm.doc.dui0497a/DUI0497A_cortex_m0_r0_p0_generic_ug.pdf)
- [13] Cortex-M0+, [online]. [cit.2013-06-03] Dostupné z: [http://infocenter.arm.com/help/topic/com.arm.doc.dui0662b/DUI0662B\\_cortex\\_m0p\\_r0p1\\_dgug.pdf](http://infocenter.arm.com/help/topic/com.arm.doc.dui0662b/DUI0662B_cortex_m0p_r0p1_dgug.pdf)
- [14] Cortex-M3, [online]. [cit.2013-06-03] Dostupné z: [http://infocenter.arm.com/help/topic/com.arm.doc.dui0552a/DUI0552A\\_cortex\\_m3\\_dgug.pdf](http://infocenter.arm.com/help/topic/com.arm.doc.dui0552a/DUI0552A_cortex_m3_dgug.pdf)
- [15] Cortex-M4, [online]. [cit.2013-06-03] Dostupné z: [http://infocenter.arm.com/help/topic/com.arm.doc.dui0553a/DUI0553A\\_cortex\\_m4\\_dgug.pdf](http://infocenter.arm.com/help/topic/com.arm.doc.dui0553a/DUI0553A_cortex_m4_dgug.pdf)
- [16] Nuvoton, [online]. [cit.2013-06-03] Dostupné z: <http://www.nuvoton.com/hq/enu/pages/default.aspx>
- [17] Holtek, [online]. [cit.2013-06-03] Dostupné z: <http://www.holtek.com/english/>
- [18] Sloss, Andrew N.; Symes, Dominic.; Wright.: ARM System Developer's Guide : Designing and Optimizing System Software, 2004
- [19] Trevor, Martin.: The Insider's Guide To The STM32 ARM Based Microcontroller, 2008
- [20] Yiu, Joseph.: The Definitive Guide to the ARM Cortex-M3, 2007
- [21] Trevor, Martin.; Michael, Beach.: The Insider's Guide To The STR91x ARM®9, 2006
- [22] Keil, [online]. [cit.2013-06-03] Dostupné z: <http://www.keil.com/>
- [23] Coocox, [online]. [cit.2013-06-03] Dostupné z: <http://coocox.org/index.html>
- [24] IAR, , [online]. [cit.2013-06-03] Dostupné z: <http://www.iar.com/>
- [25] Atollic, , [online]. [cit.2013-06-03] Dostupné z: <http://www.atollic.com/>
- [26] Raisonance, [online]. [cit.2013-06-03] Dostupné z: <http://www.raisonance.com/>
- [27] ST-Link pro Linux, [online]. [cit.2013-06-03] Dostupné z: <http://startingelectronics.com/tutorials/STM32-microcontrollers/programming-STM32-flash-in-Linux/>
- [28] CodeSourcery, [online]. [cit.2013-06-03] Dostupné z: <https://sourcery.mentor.com/GNUToolchain/>

## 7 Seznam obrázků

<i>Obr. 1 Registry v jednotlivých módech na ARM7TDMI</i> .....	14
<i>Obr. 2 Registry Cortex-M</i> .....	15
<i>Obr. 3 Zarovnané a nezarovnané proměnné v paměti</i> .....	15

## 8 Seznam tabulek

<i>Tab. 1 Celkové srovnání rodin</i> .....	16
<i>Tab. 2 Celkové srovnání instrukčních sad</i> .....	18
<i>Tab. 3 Porovnání ARM mikrokontrolérů část 1</i> .....	48

Tab. 4 Porovnání ARM mikrokontrolérů část 2.....	49
Tab. 5 Porovnání ARM mikrokontrolérů část 3.....	50
Tab. 6 Porovnání ARM mikrokontrolérů část 4.....	51
Tab. 7 velikost zkompilevaného kódu pro matice 2x2 typu int uložené jako lokální proměnné .....	56
Tab. 8 velikost zkompilevaného kódu pro matice 4x4 typu int uložené jako lokální proměnné .....	56
Tab. 9 velikost zkompilevaného kódu pro matice 2x2 typu float uložené jako lokální proměnné.....	57
Tab. 10 velikost zkompilevaného kódu pro matice 4x4 typu float uložené jako lokální proměnné.....	57
Tab. 11 velikost zkompilevaného kódu pro matice 2x2 typu int uložené jako globální proměnné.....	57
Tab. 12 velikost zkompilevaného kódu pro matice 4x4 typu int uložené jako globální proměnné.....	57
Tab. 13 velikost zkompilevaného kódu pro matice 2x2 typu float uložené jako globální proměnné.....	58
Tab. 14 velikost zkompilevaného kódu pro matice 4x4 typu float uložené jako globální proměnné.....	58
Tab. 15 rychlost zkompilevaného kódu pro matice 2x2 typu int uložené jako lokální proměnné .....	58
Tab. 16 rychlost zkompilevaného kódu pro matice 4x4 typu int uložené jako lokální proměnné .....	58
Tab. 17 rychlost zkompilevaného kódu pro matice 2x2 typu float uložené jako lokální proměnné.....	59
Tab. 18 rychlost zkompilevaného kódu pro matice 4x4 typu float uložené jako lokální proměnné.....	59
Tab. 19 rychlost zkompilevaného kódu pro matice 2x2 typu int uložené jako globální proměnné .....	59
Tab. 20 rychlost zkompilevaného kódu pro matice 4x4 typu int uložené jako globální proměnné .....	59
Tab. 21 rychlost zkompilevaného kódu pro matice 2x2 typu float uložené jako globální proměnné.....	60
Tab. 22 rychlost zkompilevaného kódu pro matice 4x4 typu float uložené jako globální proměnné.....	60

## 9 Seznam použitého softwaru

Keil uVision4 verze 7.71.20

Coocox verze 1.7.2

AVR Studion 6.0 verze 6.0.0.1996

Ride7 verze 7.44.13.0109

Rkit-ARM for Ride verze 1.50.13.0109

Attolic lite verze 4.0.1

IAR verze 6.50.6.4958

ST-Link Utility verze 3.0.0

## 10 Obsah přiloženého DVD

Datasheet	Analog Devices		
	ARM		
	Atmel		
	Cypress	PSoC4000	
		PSoC5000LP	
	Energy micro	EFM32 Gecko	
		EFM32 Giant Gecko	
		EFM32 Leopard Gecko	
		EFM32 Tiny Gecko	
		EFM32 Wonder Gecko	
		EFM32 Zero Gecko	
	Freescale	Kinetis K	K10
			K20
			K30
			K40
			K50
			K60
			K70
		Kinetis KL	KL0
			KL1
			KL2
	Holtek		
	Nuvoton	Nuc100 serie Advance Line	
		Nuc120 serie USB Line	
		Nuc130 serie Autotive line	
		Nuc140 serie Connectivity Line	
	NXP	LPC8xx	
		LPC11xx	
		LPC12xx	
		LPC13xx	
		LPC18xx	
		LPC40xx	
		LPC43xx	
	STMicroelectronics	STM32F0	
		STM32F1	
		STM32F2	
		STM32F3	
		STM32F4	
		STM32L1	
		STM32W	
	Texas Instruments	TM4 Tiva C	
		TMS470 Hercules	
Přípravek na odměření rychlosti kódu			datasheet program



schéma

Software	Atmel Atollic Coocox Keil Raisonance ST-Link Utility	Starší verze
Ukázkové projekty z měření	Atollic AVR Studio 6.0 Coocox Keil uVision4 Raisonance	

## 11 Přílohy

### Seznam příloh

**Příloha A – testovací kód násobení matic**

**Příloha B – program pro atmega8 v měřicím přípravku**

**Příloha C – schéma zapojení přípravku pro odměření rychlosti kódu**

### Přílohy

#### Příloha A – testovací kód násobení matic

```
#include "stm32f10x.h"

//testovaný rozměr matice 2 - 2x2, 4 - 4x4
#define ROZMERMATICE 2
//počet násobení matic, pro naše měření bude mít hodnotu 1
#define POCETNASOBENI 1
//datový typ matice - float nebo int
#define MATICEDATATYP int
//hodnota pro nastavení počáteční hodnoty matice
#define NASTAVENIHODNOTY 1.23456f
//definice umístění matic v paměti, 0 - lokálně, 1 -globálně
#define MATICEUMISTENI 0

#if MATICEUMISTENI == 1
//globální matice
MATICEDATATYP f_MaticePrvni[ROZMERMATICE][ROZMERMATICE],
f_MaticeDruha[ROZMERMATICE][ROZMERMATICE],
f_MaticeVysledek[ROZMERMATICE][ROZMERMATICE];
#endif

// nastavíme potřebné periferie
void SystemInit(void)
{
    RCC->CR |= RCC_CR_HSION; //RC oscilátor
    RCC->CR |= RCC_CR_HSEON; //zapnutí krystalového oscilátoru
    while((RCC->CR & RCC_CR_HSERDY ) == 0); /*počkáme, až se rozběhne
krystalový oscilátor*/
}
```

```

    RCC->CFGR |=RCC_CFGR_SW_HSE; // přepneme hodiny na krystalový
oscilátor
    RCC->CIR = 0x00FF0000; // vyčistíme interrupt register
}

//inicializace bran
void InitGates()
{
    RCC->APB2ENR |= RCC_APB2ENR_IOPCEN; //zapneme hodiny do GPIOC
    GPIOC->CRH &= 0xFFFFF00; // konfigurace testovacího pinu
    GPIOC->CRH |= 0x00000003; /* konfigurace testovacího pinu na výstup s
max. frekvencí 50MHz*/
}

//nahození bitu
void RisePin()
{
    GPIOC->BSRR = 1UL<<8; // nahodíme pin na GPIOC
}

//shozeni bitu
void FallPin()
{
    GPIOC->BRR =1UL<<8; // shodíme pin na GPIOC
}

//Násobení matic uložené lokálně
#if MATICEUMISTENI == 0
void NasobeniMatic( MATICEDATATYP f_Prvni[ROZMERMATICE][ROZMERMATICE],
                  MATICEDATATYP f_Druha[ROZMERMATICE][ROZMERMATICE],
                  MATICEDATATYP f_Vysledek[ROZMERMATICE][ROZMERMATICE])
{
    char c_x,c_y,c_index;
    MATICEDATATYP f_tmp;
    for(c_y=0;c_y<ROZMERMATICE;c_y++)
        for(c_x=0;c_x<ROZMERMATICE;c_x++)
        {
            f_tmp = 0;
            for(c_index=0;c_index<ROZMERMATICE;c_index++)
            {
                f_tmp += f_Prvni[c_index][c_y] *
                f_Druha[c_x][c_index];
            }
            f_Vysledek[c_x][c_y] = f_tmp;
        }
}
#endif

//Násobení matic uložené globálně
#if MATICEUMISTENI == 1
void NasobeniMatic()
{
    char c_x,c_y,c_index;
    MATICEDATATYP f_tmp;
    for(c_y=0;c_y<ROZMERMATICE;c_y++)
        for(c_x=0;c_x<ROZMERMATICE;c_x++)
        {
            f_tmp = 0;
            for(c_index=0;c_index<ROZMERMATICE;c_index++)
            {

```

```
        f_tmp += f_MaticePrvni[c_index][c_y] *
        f_MaticeDruha[c_x][c_index];
    }
    f_MaticeVysledek[c_x][c_y] = f_tmp;
}
}
#endif

int main()
{
    InitGates(); //nastaví nám piny
    FallPin(); //clear pin
    while(1)
    {
        #if MATICEUMISTENI == 0
        //lokální matice
        MATICEDATATYP    f_MaticePrvni[ROZMERMATICE][ROZMERMATICE],
                        f_MaticeDruha[ROZMERMATICE][ROZMERMATICE],
                        f_MaticeVysledek[ROZMERMATICE][ROZMERMATICE];
        #endif

        int c_x,c_y;
        int i_index;

        //naplnime matice
        for(c_y=0;c_y<ROZMERMATICE;c_y++)
            for(c_x=0;c_x<ROZMERMATICE;c_x++)
            {
                f_MaticePrvni[c_x][c_y] = NASTAVENIHODNOTY;
                f_MaticeDruha[c_x][c_y] = NASTAVENIHODNOTY;
            }

        RisePin(); //set pin

        for(i_index=0;i_index<POCETNASOBENI;i_index++)
        {
            #if MATICEUMISTENI == 0
            NasobeniMatic(f_MaticePrvni,f_MaticeDruha,
                        f_MaticeVysledek);
            #endif

            #if MATICEUMISTENI == 1
            NasobeniMatic();
            #endif
        }

        FallPin(); //clear pin
    }
}
```

**Příloha B – program pro atmega8 v měřicím přípravku**

```
#include <mega8.h>
// Alphanumeric LCD Module functions
#include <alcd.h>
#include <stdio.h>
#include <stdlib.h>
#include <delay.h>

volatile unsigned int PocetTick=0;
register unsigned int ZacatekPocitani=7; //kalibrace : 4 takty trvá reakce
na přerušení, 2 takty nakopírování nové hodnoty 1 načtení hodnoty

// Timer1 overflow interrupt service routine
interrupt [TIM1_CAPT] void timer1_capt_isr(void) //TIM1_CAPT
{
    TCNT1 = ZacatekPocitani;
}

interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
    PocetTick++;
}

void main(void)
{
    // Declare your local variables here

    // Input/Output Ports initialization
    // Port B initialization
    // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
    // State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
    PORTB=0x00;
    DDRB=0x00;

    // Port C initialization
    // Func6=In Func5=In Func4=In Func3=Out Func2=Out Func1=Out Func0=Out
    // State6=T State5=T State4=T State3=0 State2=0 State1=0 State0=0
    PORTC=0x00;
    DDRC=0x0F;

    // Port D initialization
    // Func7=Out Func6=Out Func5=In Func4=In Func3=In Func2=In Func1=In
    Func0=In
    // State7=0 State6=0 State5=T State4=T State3=T State2=T State1=T State0=T
    PORTD=0x00;
    DDRD=0xC0;

    // Timer/Counter 0 initialization
    // Clock source: System Clock
    // Clock value: Timer 0 Stopped
    TCCR0=0x00;
    TCNT0=0x00;

    // Timer/Counter 1 initialization
    // Clock source: System Clock
    // Clock value: 8000,000 kHz
    // Mode: CTC top=ICR1
    // OC1A output: Discon.
    // OC1B output: Discon.
```

```
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: On
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B = (1<<CS10); // CLK/1
TIMSK = (1 << TOIE1); // Timer/Counter1 Overflow Interrupt Enable
TCCR1B |= (1<<ICNC1); //povolíme externí událost
TIMSK |= (1<<TICIE1);
TCNT1H = 0;
TCNT1L = 0;

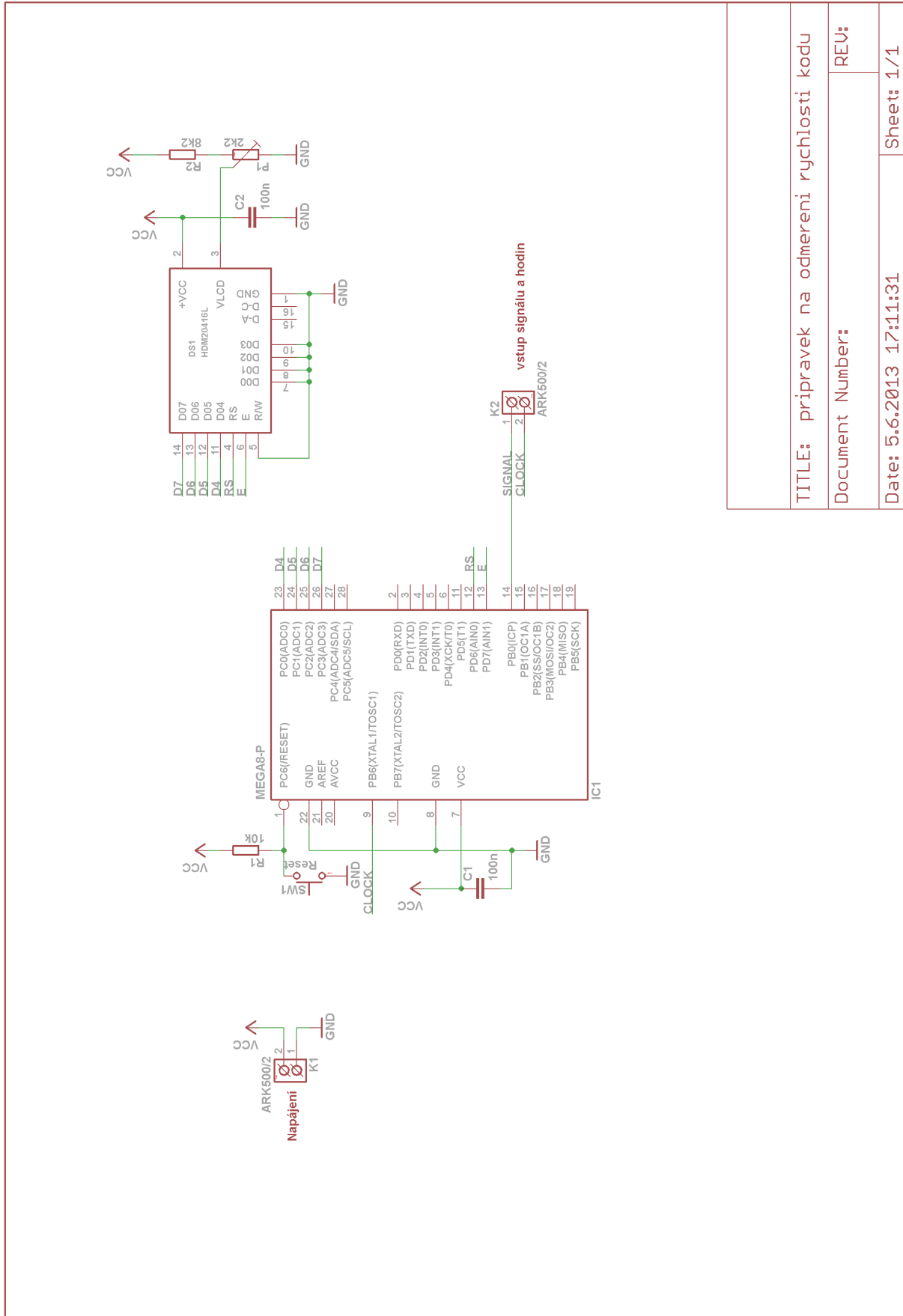
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer2 Stopped
// Mode: Normal top=0xFF
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;
// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
MCUCR=0x00;
// Timer(s)/Counter(s) Interrupt(s) initialization
// USART initialization
// USART disabled
UCSRB=0x00;
// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;
// ADC initialization
// ADC disabled
ADCSRA=0x00;
// SPI initialization
// SPI disabled
SPCR=0x00;
// TWI initialization
// TWI disabled
TWCR=0x00;
// Alphanumeric LCD initialization
// Connections specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
// RS - PORTD Bit 6
// EN - PORTD Bit 7
// D4 - PORTC Bit 0
// D5 - PORTC Bit 1
// D6 - PORTC Bit 2
// D7 - PORTC Bit 3
// Characters/line: 20
```

```
lcd_init(20); //inicializace displeje o 20 sloupcích
// Global enable interrupts
#asm("sei")
lcd_gotoxy(0,0);
lcd_puts("mereni");
while (1)
{
    char szText[21];
    sprintf(szText,"%u Tick      ",ICR1);
    lcd_gotoxy(0,1);
    lcd_puts(szText);

    if (PocetTick > 0)
    {
        sprintf(szText,"casovac pretekl");
        lcd_gotoxy(0,2);
        lcd_puts(szText);
    }else
    {
        sprintf(szText,"casovac ok      ");
        lcd_gotoxy(0,2);
        lcd_puts(szText);
    }

    PocetTick=0;
    delay_ms(1000);
}
}
```

**Příloha C – schéma zapojení přípravku pro odměření rychlosti kódu**



TITLE: prípravek na odmereni rychlosti kodu

Document Number: REV:

Date: 5.6.2013 17:11:31 Sheet: 1/1