

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2019

Bc. Tomáš Koleček



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

LABORATORNÍ MIKROKONTROLÉROVÝ SYSTÉM FEKTIS

LABORATORY MICRONOCTROLLER SYSTEM FEKTIS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Tomáš Koleček

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Zdeněk Bradáč, Ph.D.

BRNO 2019

Diplomová práce

magisterský navazující studijní obor **Kybernetika, automatizace a měření**

Ústav automatizace a měřicí techniky

Student: Bc. Tomáš Koleček

ID: 154768

Ročník: 2

Akademický rok: 2018/19

NÁZEV TÉMATU:

Laboratorní mikrokontrolerový systém FEKTis

POKYNY PRO VYPRACOVÁNÍ:

1. Prostudujte koncepci stávajícího laboratorního systému I51 a proveďte literární rešerši možného upgradu na platformu ARM.
2. Navrhněte koncepci komplexního laboratorního výukového kitu vybaveného mikrokontrolérem ARM ve verzi napájení 5V a periferiemi. Systém navrhněte tak, aby byl plně kompatibilní se stávajícími koncovými periferiemi s možností rozšíření jejich portfolia.
3. Navrhněte elektronické zapojení centrální jednotky mikrokontrolerového systému, vytvořte DPS, zrealizujte a oživte.
4. Navrhněte a oživte vhodný volně dostupný toolchain, který půjde využít pro výuku programování. Toolchain zvolte tak, aby umožňoval programování a on-line ladění uživatelských programů. Orientujte se na prostředí Windows.
5. Navrhněte vzorové laboratorní úlohy, vytvořte programové vybavení pro mikrokontrolér pro vybrané vzorové úlohy a zdokumentujte vzorová řešení. Demonstrujte správnou funkci.

DOPORUČENÁ LITERATURA:

Pavel Herout: Učebnice jazyka C, KOPP, 2004, IV. přepracované vydání, ISBN 80-7232-220-6

Dle pokynů vedoucího práce.

Termín zadání: 4.2.2019

Termín odevzdání: 13.5.2019

Vedoucí práce: doc. Ing. Zdeněk Bradáč, Ph.D.

Konzultant:

doc. Ing. Václav Jirsík, CSc.
předseda oborové rady

UPOZORNĚNÍ:

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

ABSTRAKT

Tato práce se zabývá návrhem a realizací laboratorního mikrokontrolérového systému, postaveného na architektuře ARM, pro účely výuky programování. Tento systém umožní uživateli práci s vnitřními i vnějšími periferiemi mikrokontroléru, kterými je systém vybaven. Práce zahrnuje tvorbu konceptu, jeho obvodové řešení i samotnou realizaci. Pro systém je určeno vhodné vývojové prostředí a práce obsahuje i vzorová řešení pro navržené úlohy.

KLÍČOVÁ SLOVA

ARM, Cortex-M, USART, I2C, SPI, ADC, DAC, ATSAM21C

ABSTRACT

This thesis deals with design and implementation of laboratory microcontroller system, based on ARM architecture. Purpose of this thesis is to teach programming. This system allows the user to work with the internal and external peripherals of the microcontroller, which are equipped with the system. This thesis includes creating a concept its peripheral solution and realization itself. A suitable development environment is determined for the system and the thesis includes sample solutions for the proposed tasks.

KEYWORDS

ARM, Cortex-M, USART, I2C, SPI, ADC, DAC, ATSAM21C

KOLEČEK, Tomáš. *Laboratorní mikrokontrolérový systém FEKTis*. Brno, 2019, 64 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav elektroenergetiky. Vedoucí práce: doc. Ing. Zdeněk Bradáč, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Laboratorní mikrokontrolerový systém FEKTis“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

Brno

.....

podpis autora

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu doc. Ing. Zdeňku Bradáčovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno

.....

podpis autora

Obsah

Úvod	11
1 Laboratorní systém I51	12
1.1 Mikrokontrolér AT89C51RD	12
1.2 Popis procesorové desky	14
1.3 Rozšiřující desky	16
1.3.1 Rozšiřující deska 24 Digital Outputs + 24 Digital Inputs	16
1.3.2 Rozšiřující deska pro Displej, DC motor a klávesnici	17
1.4 Koncové periferie	17
1.4.1 Testovací modul 8x digitální výstup	18
1.4.2 Testovací modul 8x Digital Input	18
1.4.3 Jednotka krokového motoru	19
1.4.4 Jednotka rotačního a křížového ovladače	20
1.4.5 Jednotka DC motoru	21
1.4.6 Jednotka alfanumerického inteligentního displeje 4x20	22
1.4.7 Jednotka maticové klávesnice a akustického indikátoru	23
1.4.8 Jednotka I ² C	24
1.5 Shrnutí	24
2 Mikroprocesory s architekturou ARM	25
2.1 Výběr mikrokontroléru s jádrem Cortex-M	27
3 Koncept Laboratorního systému FEKTIS	31
3.1 Definování blokových částí systému	31
3.2 Kompatibilita se současnými moduly	31
3.3 USART	32
3.4 I ² C	33
3.5 SPI	35
3.6 AD převodník	36
3.7 DA převodník	36
3.8 Programátor Atmel-ICE	37
3.9 Výsledný model systému FEKTis	37
4 Realizace procesorové desky systému FEKTis	39
4.1 Rozvržení I/O pinů mikrokontroléru	39
4.2 Realizace modulů	40
4.2.1 Modul ADC	40
4.2.2 Modul DAC	41

4.2.3	Modul USART	41
4.2.4	Modul I ² C	42
4.3	Oživení systému	42
5	Vývojové prostředí Atmel Studio 7	43
5.1	Tvorba projektu a nahrávání programu	43
5.2	Ladění programu	43
6	Práce s laboratorním systémem FEKTis	46
6.1	Obsluha vnitřních periférií mikrokontroléru	46
6.1.1	Práce s GPIO	46
6.1.2	Modul USART	48
6.1.3	Modul I ² C	50
6.1.4	Periferie ADC převodníku	51
6.1.5	Ovládání Rotačního ovladače	52
6.1.6	Ovládání DC motoru	53
6.1.7	Ovládání krokového motoru	53
6.1.8	Ovládání maticové klávesnice 4x4	54
6.1.9	Ovládání LCD displeje	55
6.2	Vzorové úlohy	55
6.2.1	Čtení maticové klávesnice 4x4 s prezentací dat na LCD displeji a odesílání do PC	56
6.2.2	Ovládání krokového motoru pomocí rotačního ovladače	57
6.2.3	Prezentace dat přijatých z PC pomocí sériově-paralelního I ² C převodníku	58
6.2.4	Zobrazení aktuálního času z modulu RTC na LCD displeji	60
6.2.5	Čtení pomocí ADC převodníku, ukládání hodnot do paměti EEPROM a odesílání zpráv do PC	60
7	Závěr	63
	Literatura	64

Seznam obrázků

1.1	Procesorová deska systému I51	15
1.2	Celkové zapojení laboratorního systému	15
1.3	Rozšiřovací deska 24 vstupů a 24 výstupů	16
1.4	Rozšiřovací deska pro připojení LCD, Klávesnice a DC motoru	17
1.5	Modul 8x digitální výstup	18
1.6	Modul 8x digitální vstup	19
1.7	Jednotka krokového motoru	20
1.8	Jednotka rotačního a křížového ovladače	21
1.9	Jednotka DC motoru	22
1.10	Jednotka alfanumerického inteligentního displeje 4x20	23
1.11	Jednotka maticové klávesnice a akustického indikátoru	24
2.1	Přehled mikroprocesorů řady Cortex a jejich předchůdců	26
3.1	Univerzální konektor pro připojení koncových periférií	32
3.2	Univerzální konektor pro připojení koncových periférií	34
3.3	I ² C konektor na procesorové desce	35
3.4	Propojení zařízení na sběrnici SPI	36
3.5	Propojení ATMEL-ICE a cílového zařízení	37
3.6	Základní koncept propojení koncových zařízení s procesorovou deskou	38
4.1	Konektor pro připojení programátoru Atmel-ICE	39
4.2	Schéma modulu ADC	41
4.3	Schéma modulu DAC	41
5.1	Vytvoření nového projektu	43
5.2	Volba cílového mikrokontroléru	44
5.3	Výběr zařízení pro programování a ladění	44
5.4	Vypnutí optimalizace	45
5.5	Modul pro zobrazení registrů komponent mikrokontroléru	45
6.1	Kód hlavičkového souboru port.h	47
6.2	Kód hlavičkového souboru port.h	48
6.3	Kód hlavičkového souboru port.h	48
6.4	Blokové schéma zapojení periférie USART	48
6.5	Rámeček zprávy na sériové sběrnici UART	49
6.6	Vztahy pro výpočet baudrate	49
6.7	Blokové schéma I ² C periférie	50
6.8	Stavový automat I ² C periférie	51
6.9	Blokové schéma periférie ADC	52
6.10	Průběh logických úrovní na kontaktech rotačního ovladače při otáčení po směru hodin	53

6.11	Posloupnost kroků pro otáčení krokového motoru po směru a proti směru hodin	54
6.12	Schéma zapojení maticové klávesnice	55
6.13	Výsledek běhu programu příkladu č. 1 na LCD displeji	57
6.14	Výsledek běhu programu příkladu č. 1 na PC	57
6.15	Zadané hodnoty z terminálu PC pro příklad č. 3	59
6.16	Výsledek zobrazený na LED modulu pro příklad č. 3	59
6.17	Výsledek zobrazený na LED modulu pro příklad č. 4	60
6.18	Výsledek běhu programu příkladu č. 5 na PC	62

Seznam tabulek

1.1	Popis portu P3 a jeho alternativních funkcí	13
1.2	Popis registru Serial Control pro nastavení sériové linky	14
2.1	Přehled binární kompatibility všech jader Cortex-M	27
2.2	Přehled architektur CPU/MCU vzhledem k jednotlivým jádrům skupiny Cortex-M	27
2.3	Seznam rodin vybraných mikrokontrolérů a jejich parametry	28

Úvod

Cílem této práce je návrh a realizace komplexního laboratorní systému, pro praktickou část výuky předmětu s názvem Počítače pro řízení. Současný systém je postaven na již málo používané architektuře I51, která byla nahrazena modernějšími mikroprocesory. Bylo tedy rozhodnuto o renovaci. Nově navržený systém bude využívat moderní a velice rozšířenou architekturu mikroprocesorů typu ARM.

Práce zahrnuje v první části studii konceptu aktuálního systému, založeného na mikroprocesoru I51. Pro výběr vhodné náhrady je následně proveden průzkum dostupných mikrokontrolérů s architekturou ARM ve verzi napájení 5V. Vybraný čip musí být vybaven periferiemi vhodnými pro výuku. Nový systém s vybraným mikrokontrolérem musí být navržen tak, aby bylo možné připojit moduly původního systému a dále musí nabízet možnost rozšíření jejich portfolia. Před realizací je zpracován nový koncept, který nejlépe odpovídá potřebám výuky. Součástí práce je dále návrh a realizace obvodového řešení DSP s vybraným mikrokontrolérem a oživení tohoto nově vzniklého systému.

Následně se práce zaměřuje na softwarové vybavení a pro vybraný mikrokontrolér je zvoleno vhodné vývojové prostředí. Předpokládá se vývoj aplikací v jazyce C. Závěrečná část je zaměřena na praktickou činnost s celým systémem. Je zde popsáno, jakým způsobem je možné s jednotlivými původními i nově vzniklými moduly pracovat. Poslední kapitoly definují slovní zadání, popisují jeho řešení a demonstrují výsledky řešení. Všechny programy jsou dokumentovány a lze je nalézt v elektronické příloze této práce.

1 Laboratorní systém I51

V této kapitole se práce zaměřuje na studii stávajícího laboratorního systému. Samostatné části systému je možné rozdělit do 3 kategorií. Základ tvoří hlavní deska osazená mikroprocesorem s architekturou I51 a k ní připojené rozšiřující desky, pomocí kterých jsou se systémem propojeny koncové periferie. Následující podkapitoly se nejprve zaměří na samotný popis mikrokontroléru AT89C51RD a dále budou rozebrány funkce a význam jednotlivých částí současného systému.

1.1 Mikrokontrolér AT89C51RD

Celý systém je založen na architektuře I51. Konkrétně se jedná o mikrokontrolér AT89C51RD ve verzi CMOS. Mikroprocesor typu 80C51 je 8-bitový jednočipový mikroprocesor s harwardskou strukturou s oddělenou programovou a datovou pamětí. Procesor je schopen samostatné činnosti po připojení vnějšího piezokeramického rezonátoru (krystalu) na vývody XTAL1 a XTAL2, napájecího napětí 5V a připojení nulovacího obvodu na vývod RST. Mikroprocesor tvoří centrální procesorová jednotka společně s aritmeticko-logickou jednotkou. CPU je vnitřní 8-bitovou společnou sběrnici propojena s pamětí programu a pamětí dat. Vnitřní paměť programu může být typu ROM, EPROM a nebo mikroprocesor nemusí mít žádnou vnitřní paměť programu. Vnitřní paměť dat je typu RAM. Ke společné sběrnici jsou dále připojeny 4 I/O porty P0 až P3, které umožňují komunikaci mikroprocesoru s vnějšími periferiemi. Pro připojení větší paměti programu, než kterou poskytuje mikroprocesor, je možné k mikroprocesoru připojit samostatnou vnější paměť programu a také vnější paměť dat. K tomuto účelu jsou z mikroprocesoru vyvedeny řídicí signály PSEN (paměť programu) a WR, RD (paměť dat).

Mikroprocesor je vybaven také řadičem přerušení, který obsluhuje celkem 5 zdrojů přerušení. Jsou to 2 externí, které lze připojit na vývody INT0 a INT1, dále přerušení od čítačů a časovačů a od sériového kanálu. Jednotlivá přerušení mají definovanou prioritu na každé ze dvou volitelných úrovní priority. Mikroprocesor obsahuje dva 16-bitové čítače/časovače, s volitelným režimem funkce.

Mikroprocesor ke své činnosti a správné funkci potřebuje zdroj hodinových (taktovacích) impulzů o určité frekvenci (f_{osc}), které využívají vnitřní obvody mikroprocesoru a od kterého je pak odvozena délka trvání strojového cyklu. Pro tento účel je možné využít vnitřního obvodu oscilátoru, nebo externí obvod oscilátoru. V praxi se nejvíce používá zapojení s vnitřním oscilátorem, to je s připojením krystalu. Pro výpočet frekvence platí, že frekvence použitého krystalu se rovná frekvenci oscilátoru (f_{osc}).

Z hlediska I/O portů disponuje mikroprocesor následujícími:

Port 0 je obvod s otevřeným kolektorem. V případě, že má být používán jako obecný I/O port, musí být k pinům připojeny externí odpory, jejichž druhý konec je připojen na +Ucc. Port 0 se také může využívat při styku s vnější pamětí

Port 1 nemá žádnou alternativní funkci a díky tomu jej lze bez omezení libovolně používat.

Port 2 se kromě své normální funkce může využívat spolu s portem 0 ke komunikaci s externí pamětí. K tomuto účelu se využívají jen části obou portů. Konkrétně výstupní budiče portu P0 a P2 a vstupní vyrovnávací paměť portu P0. Na portu P0 je časově přepínán výstup nižšího bytu adresy pro externí paměť s datovým bytem, který je zapisován nebo čten z paměti. Port P2 pak vysílá vyšší byte adresy. Tato adresa je 16-bitová, je možné tedy adresovat až 64kB paměti. Port P2 jinak pracuje jako normální port.

Port 3 má všechny piny vícefunkční. Kromě standardní funkce, navíc tento port poskytuje piny pro potřeby jiných vnitřních obvodů mikroprocesoru, seznam pinů a jim příslušné alternativní funkce popisuje tabulka 1.1.

Pin	Alternativní funkce
P3.0	RXD (serial input)
P3.1	TXD (serial output)
P3.2	INT0 (external interrupt)
P3.3	INT1 (external interrupt)
P3.4	T0 (Timer/Counter 0 external input)
P3.5	T1 (Timer/Counter 1 external input)
P3.6	WR (external Data Memory write strobe)
P3.7	RD (external Data Memory read strobe)

Tab. 1.1: Popis portu P3 a jeho alternativních funkcí

Mikrokontrolér obsahuje také zařízení pro komunikace po sériové lince, kterou lze využít pro spojení s PC a pro komunikaci se zařízeními vybavenými stejným rozhraním. Sériový kanál je u 8051 plně duplexní, to znamená, že dokáže současně vysílat i přijímat data. Umožňuje komunikovat ve standardním osmibitovém a devítibitovém asynchronním režimu nebo osmibitovém synchronním režimu. Pro výběr funkce komunikační linky je určen registr SCON (Serial Control). Tabulka 1.2 popisuje kombinace nastavení registru podle zvolené funkce.

SM1	SM2	Mód	Typ přenosu	Přenosová rychlost (b/s)
0	0	0	Synchronní osmibitový	$f_{osc}/12$
0	1	1	osmibitový UART	Časovač 1
1	0	2	Devítibitový UART	$f_{osc}/32$, $f_{osc}/64$
1	1	3	Devítibitový UART	Časovač 1

Tab. 1.2: Popis registru Serial Control pro nastavení sériové linky

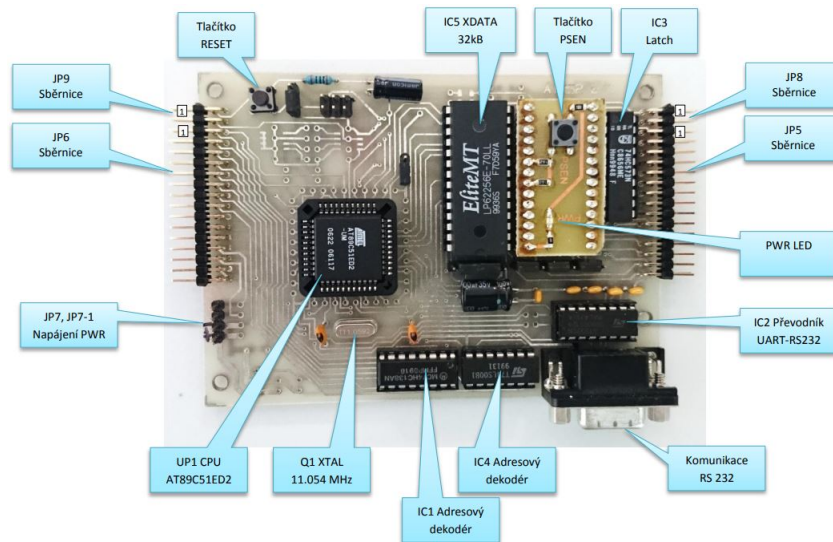
Celé komunikační rozhraní tvoří následující 3 vodiče:

- RxD
- TxD
- GND

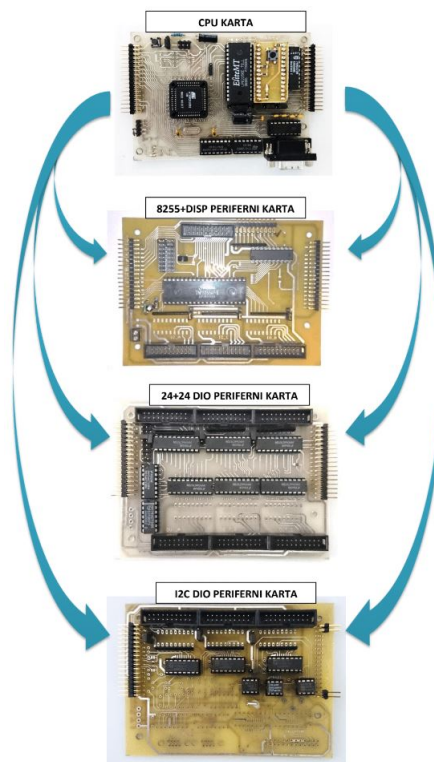
Přijímací část sériového kanálu je vybavena vyrovnávacím registrem (bufferem), do kterého je uložena právě přijatá hodnota a je umožněn okamžitý příjem další hodnoty. Přijatá hodnota však musí být z vyrovnávacího registru převzata dříve, než je dokončen příjem následující hodnoty, který by způsobil přepsání hodnoty původní. Mikroprocesor není vybaven příznaky, které indikují ztrátu přijaté hodnoty, chybu rámce a parity nebo indikaci přerušení, které jsou obvyklé u modernějších periférií tohoto typu [1].

1.2 Popis procesorové desky

Pro přehlednost dále popisovných komponent desky je uveden obrázek 1.1. Celkové schéma je umístěno v příloze této práce. Celá základní deska je napájena napětím +5V. Pro správnou funkci mikrokotroléru, je dále deska vybavena krystalem, sloužícím jako externí oscilátor (hodiny), resetovacím tlačítkem, externí paměti o velikosti 32kB a k ní vyvedené signály mikroprocesoru RD a WR pro čtení a zápis. Tlačítko PSEN dává čipu příkaz pro čtení právě z této externí paměti a využívá se při nahrávání programu. Pro obousměrnou komunikace s externí paměti se využívá také osmibitový paralelní registr typu latch, jehož hodinový signál pro synchronizaci je přiveden z výstupu ALE (Address Latch Enable). Na základní desce se také nachází obvody pro účely asynchronní sériové komunikace s PC. Konkrétně jde o převodník (MAX-232) z UART na RS232. K tomuto čipu jsou přivedeny signály mikroprocesoru Rx a Tx. Pro připojení k PC je deska vybavena devítipinovým konektorem D-SUB. V poslední řadě lze uvést sběrnice s označením JP6, JP9 a JP8, JP5, které jsou použity pro připojení k dalším částem laboratorního systému. Na obrázku 1.2 je znázorněno propojení mezi procesorovou deskou a rozšiřujícími moduly.



Obr. 1.1: Procesorová deska systému I51



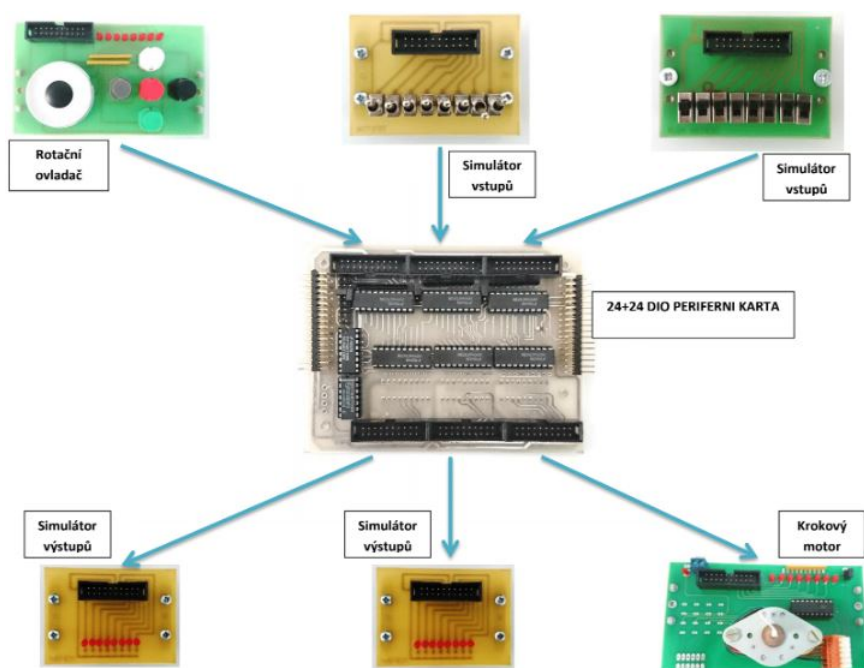
Obr. 1.2: Celkové zapojení laboratorního systému

1.3 Rozšiřující desky

Koncept laboratorního systému je navržen tak, že koncové periferie nejsou připojeny přímo se základní deskou a to z toho důvodu, že řídicí čip nedisponuje dostatečným počtem I/O portů, což bylo vysvětleno v teoretické části. Jsou proto připraveny rozšiřující desky, které jsou navrženy pro konkrétní koncové periferie. Následující podkapitoly budou popisovat jednotlivé desky a jejich význam pro celý systém. Obrázek níže zobrazuje jejich vzájemné propojení s procesorovou deskou.

1.3.1 Rozšiřující deska 24 Digital Outputs + 24 Digital Inputs

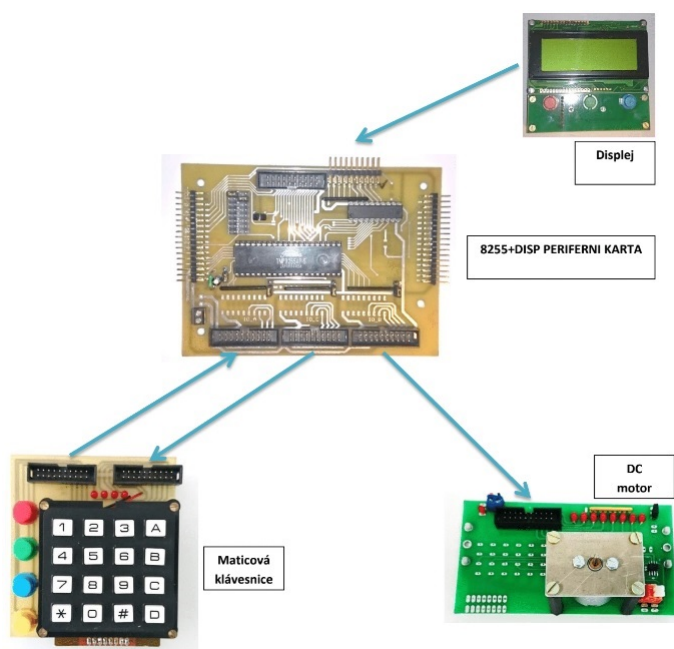
K této desce je možné připojit celkem 3 vstupní a 3 výstupní koncové periferie. Přepínání mezi jednotlivými periferiemi je řešeno následovně. Datová sběrnice mikroprocesoru je připojena k šesti integrovaným obvodům 74HC573, jedná se o osmikanálový D-klopný obvod. Přepínání mezi těmito obvody je realizováno demultiplexerem 74HC138 a pro adresaci jsou použity 3 adresovací piny mikrokontroléru. Jelikož D-klopný obvod propouští data pouze v jednom směru, musí být definováno, jestli je koncová periferie, připojená k této desce, vstupní a nebo výstupní. Toto rozřazení bude bráno z pohledu řídicí jednotky (mikrokontroléru). Desku popisuje obrázek 1.3.



Obr. 1.3: Rozšiřovací deska 24 vstupů a 24 výstupů

1.3.2 Rozšiřující deska pro Displej, DC motor a klávesnici

Účelem této rozšiřující desky, je možnost připojení a ovládání koncových periférií s DC motorem, LCD displejem a maticovou klávesnicí 4x4. Pro obsluhu modulu s LCD displejem se využívá 11 signálových pin a další 3 jsou přivedeny na samostatné tlačítka umístěna na tomto modulu. Osmice datových pinů je připojena přes čip s osmi obousměrnými D klopnými obvody. Maticová klávesnice a DC motor jsou připojeny zvlášť ke 3 tranzistorovým polím ULN2803 jejichž vstupy jsou ovládány z integrovaného obvodu intel 8255. Tento čip je připojen na sběrnici mikrokontroléru z procesorové desky. Dále na adresovací piny A0, A1 a také na řídicí RW a RD. Tímto způsobem je řešeno čtení a zápis tří koncových periférií za pomoci jedné sběrnice mikrokontroléru. Připojení koncových zařízení k této desce je znázorněno na obrázku 1.4.



Obr. 1.4: Rozšiřovací deska pro připojení LCD, Klávesnice a DC motoru

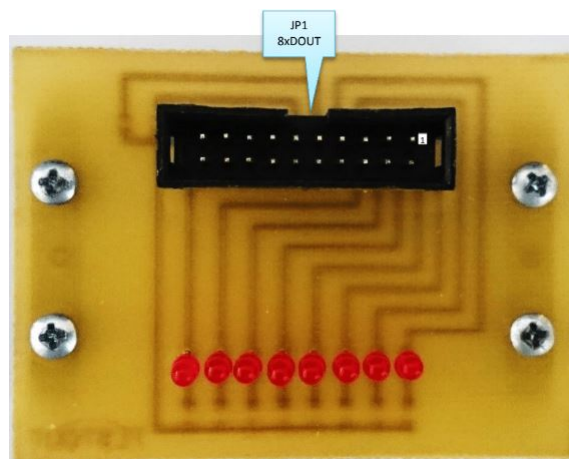
1.4 Koncové periferie

Tato kapitola popisuje koncové periferie, využívané v současném systému. Ze zadání vyplývá, že mají být kompatibilní i se systémem novým a proto bude nutné prostudovat jejich funkce, rozhraní, HW zapojení a parametry použitých integrovaných obvodů, které budou potřebné při tvorbě konceptu nového systému. Koncových periférií je celkem 8 a jedná se o samostatné jednotky připojené k rozšiřujícím deskám

popsaným výše. Další podkapitoly se zaměřují detailněji na každou z koncových periférií.

1.4.1 Testovací modul 8x digitální výstup

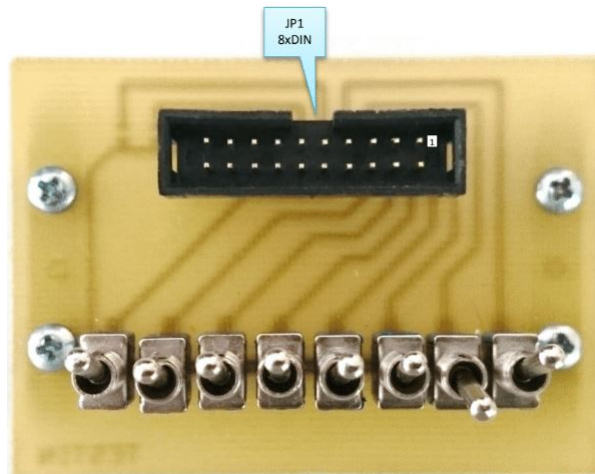
Tento modul je osazen osmi indikačními LED diodami, kde anoda každé z těchto diod je připojena zvlášť k jednomu pinu konektoru a indikují tedy jeho logický stav. Katoda je připojena k logické 0. V případě, že je pin v logické 1, dioda svítí. V opačném případě je dioda zhaslá. Z pohledu řídicí jednotky je tedy tento modul výstupní. Modul je připojen k rozšiřující desce s názvem Rozšiřovací deska 24 vstupů a 24 výstupů. Piny konektoru jsou dále připojeny na napájecí napětí +5V a zem k příslušné rozšiřovací desce. Tento modul je z pohledu řídicí jednotky výstupní. Modul je na obrázku 1.5.



Obr. 1.5: Modul 8x digitální výstup

1.4.2 Testovací modul 8x Digital Input

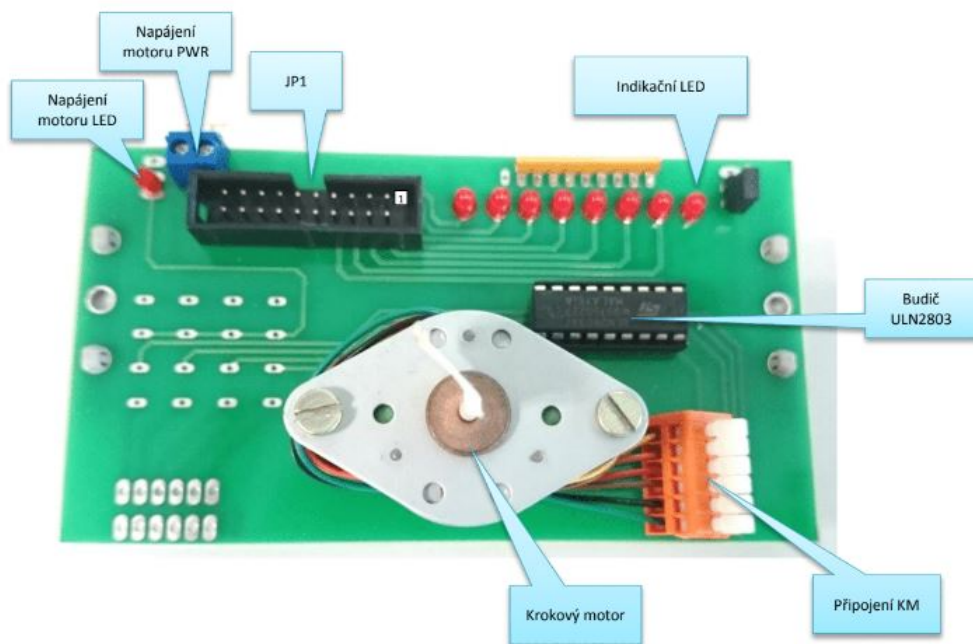
Tato periférie je osazena osmi dvupolohovými přepínači. Každý přepínač je opět připojen na samostatný pin konektoru. Přepínač je v 1. poloze připojen na logickou 0 a v 2. poloze je rozpojen. Pomocí přepínačů je možné měnit stavy jednotlivých pinů. Lze definovat, že tento modul je z pohledu řídicí jednotky vstupní. Piny konektoru jsou dále připojeny na napájecí napětí +5V a zem k příslušné rozšiřovací desce. Celý modul je připojen konektorem přivádějícím napětí +5V, zem a 8 pinů. Modul je na obrázku 1.6.



Obr. 1.6: Modul 8x digitální vstup

1.4.3 Jednotka krokového motoru

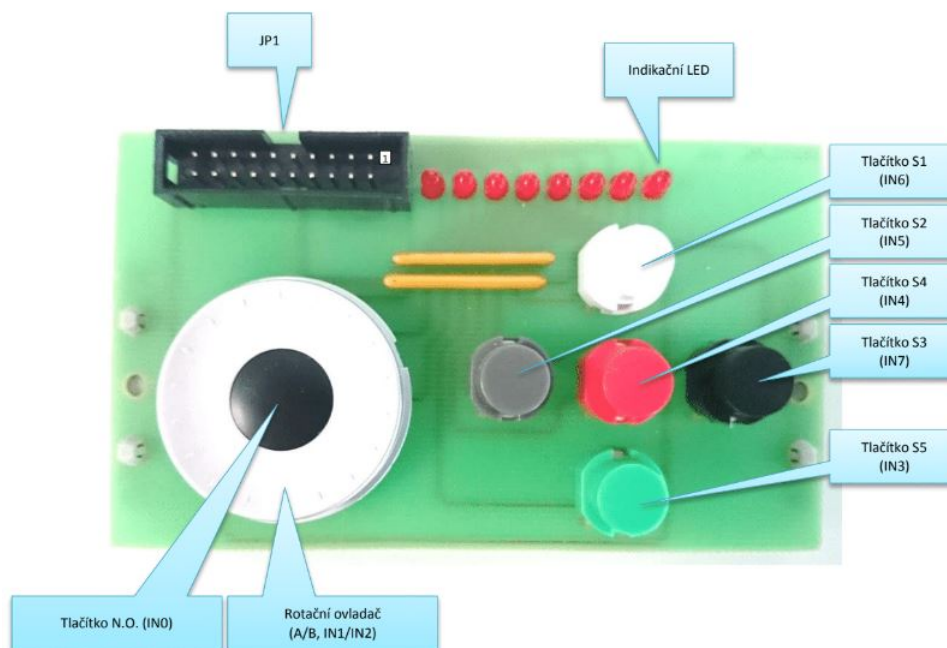
Modul je určen pro řešení úlohy ovládání krokového motoru. Napájení motoru je z důvodu vyššího odběru přivedeno externě. Potřebné napětí je +5V a pro kontrolu napájení je připojena LED dioda. Pro dostatečné buzení krokového motoru je využit integrovaný obvod ULN2803, který obsahuje pole osmi tranzistorů. Ke každému vstupu (tranzistoru) tohoto obvodu je připojen pin vstupního konektoru modulu. Ovládaný krokový motor má 4 fáze a lze ho tedy řídit pomocí 4 pinů. Indikaci logického stavu pinů zajišťuje připojená soustava LED diod. Pomocí těchto diod je možné kontrolovat správné přepínání fází krokového motoru. Zbylé 4 piny (výstupy tranzistoru) byly na desce vyvedeny pro připojení spínacích relé. Ty ale nejsou osazeny. Celý modul je připojen konektorem přivádějícím napětí +5V, zem a 8 pinů. U této periferie je nutné definovat požadavky na správnou funkčnost. Z dokumentace obvodu ULN2803 vyplývá, že minimální hodnota napětí připojená na vstupy tranzistorů musí být alespoň +3,5V. Modul je na obrázku 1.7.



Obr. 1.7: Jednotka krokového motoru

1.4.4 Jednotka rotačního a křížového ovladače

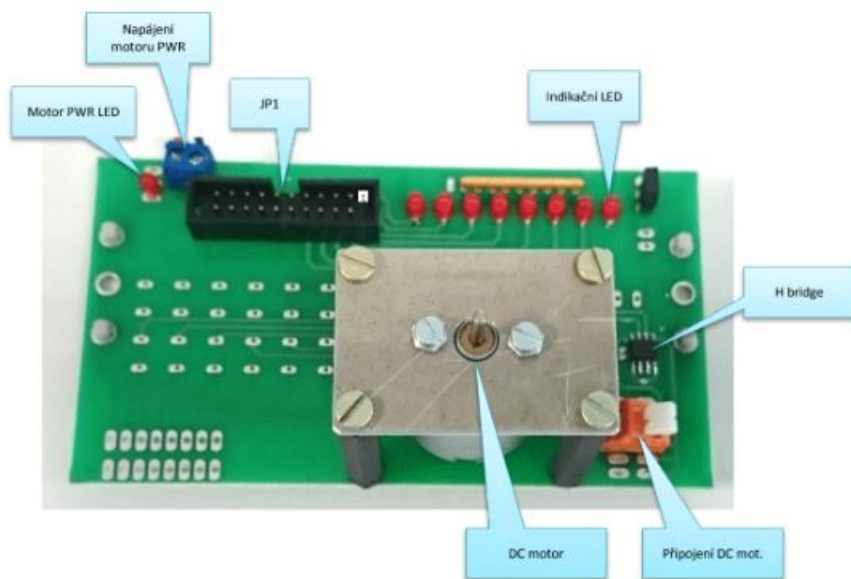
Jak již název napovídá, tato periferie je vybavena rotačním ovladačem s tlačítkem v jeho středu a křížovým ovladačem, který je realizován pomocí čtyř tlačítek a jednoho tlačítka uprostřed. Celý modul je připojen konektorem přivádějícím napětí +5V, zem a 8 pinů. Periferie je z pohledu řídicí jednotky vstupní. Rotační ovladač je připojen na 3 vstupní piny a křížový ovladač na zbylých 5. Samotná Rotační část ovladače využívá 2 piny. Podle nich lze vyhodnocovat jak pootočení, tak i směr otáček. Pin připojený k tlačítku uprostřed stejně jako všechny tlačítka křížového vypínače jsou v rozepnutém stavu rozpojeny a při stisku jsou spojeny s logickou 0. Pro indikaci stavu tlačítek a kontaktů rotačního snímače je modul vybaven osmicí LED diod. Celý modul je připojen konektorem přivádějícím napětí +5V, zem a 8 pinů. Modul je na obrázku 1.8.



Obr. 1.8: Jednotka rotačního a křížového ovladače

1.4.5 Jednotka DC motoru

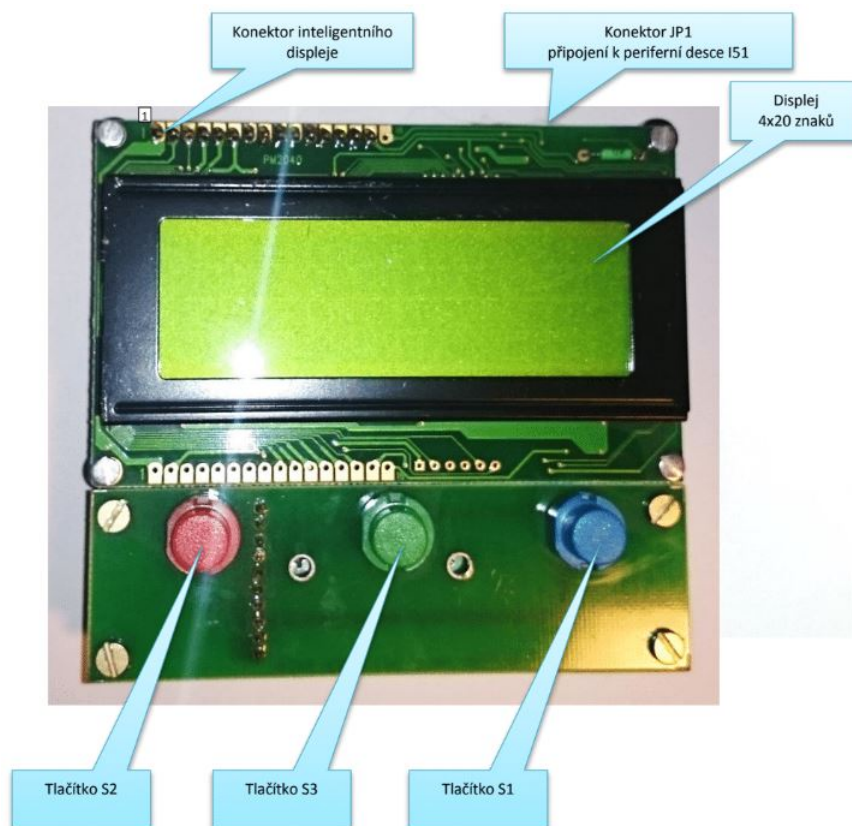
Periferie je vybavena DC motorem, který je připojen k integrovanému obvodu s jedním H-můstkem. Za pomoci tohoto obvodu je možné řídit rychlost a směr otáčení motoru. Pro řízení jsou využity 2 výstupní piny konektoru. Směr otáčení se určí podle toho, na který ze 2 vstupních pinů je přiveden řídicí signál. Rychlost otáčení je pak dána PWM signálem. S vyšší střídou logické 1 se frekvence otáček zvyšuje. Jednotka je připojena konektorem přivádějícím napětí +5V, zem a 8 pinů. Modul je na obrázku 1.9.



Obr. 1.9: Jednotka DC motoru

1.4.6 Jednotka alfanumerického inteligentního displeje 4x20

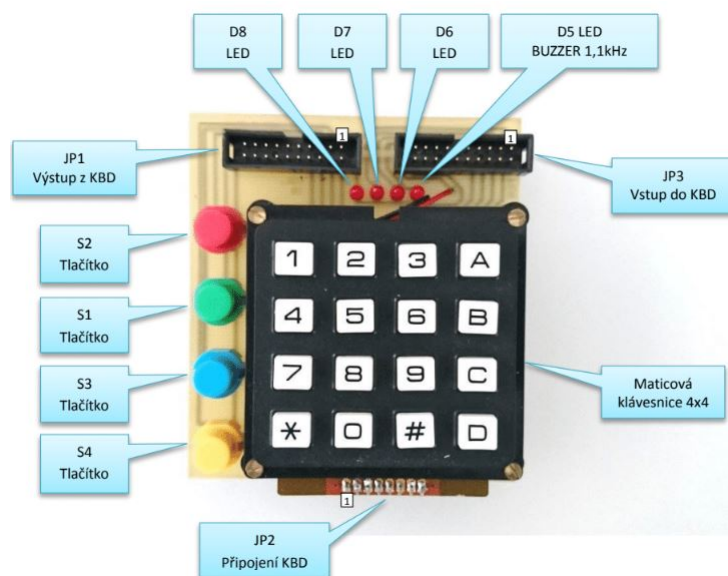
Hlavní částí jednotky je LCD displej s označením CM1624S1, který může zobrazit maximálně 4 řádky po dvaceti znacích. Periferie využívá, na rozdíl od výše popísaných periferií, celkem 11 výstupních pinů pro řízení displeje a další 3 vstupní pro detekci stisknutí tlačítek. Displej je připojen na napájení +5V z propojovacího konektoru a jas je nastavován pomocí trimru na rozšiřující desce, ke které je tato jednotka připojena. Tento modul je vstupně výstupní. Jednotka je na obrázku 1.10.



Obr. 1.10: Jednotka alfanumerického inteligentního displeje 4x20

1.4.7 Jednotka maticové klávesnice a akustického indikátoru

Tento modul, na rozdíl od modulů předešlých využívá dvojici konektorů, jeden konektor je z pohledu řídicího systému vstupní a druhý výstupní. Jedna čtveřice vstupních pinů slouží pro detekci stisknutých tlačítek a druhá čtveřice je připojena k maticové klávesnici, kde jsou využity pro přepínání sloupců tlačítek. Z výstupního konektoru je naopak jedna čtveřice využita pro vyčítání stisknutého tlačítka na maticové klávesnici a druhá čtveřice je připojena k indikačním LED diodám a akustickému indikátoru. Jednotka je připojena dvojicí konektorů přivádějícím napětí +5V, zem a 8 pinů. Celý modul je na obrázku 1.11.



Obr. 1.11: Jednotka maticové klávesnice a akustického indikátoru

1.4.8 Jednotka I²C

Poslední z uvedených periférií, která má být kompatibilní s novým systémem je vybavena několika integrovanými obvody komunikující s řídicí jednotkou pomocí rozhraní I²C. Periferie je osazena pamětí typu EEPROM a SRAM, dále pak RTC čipem a převodníky z I²C na paralelní sběrnici. Tento modul na rozdíl od ostatních koncových periférií je připojen přímo k procesorové desce a také má odlišné rozložení pinů.

1.5 Shrnutí

V tomto bodě, byla zakončena studie současného laboratorního systému a byly popsány všechny důležité poznatky. Tyto informace budou zahrnuty do návrhu. Především jsou důležité definice rozhraní koncových periférií, které budou muset být zachovány pro jejich kompatibilitu s novým laboratorním systémem.

2 Mikroprocesory s architekturou ARM

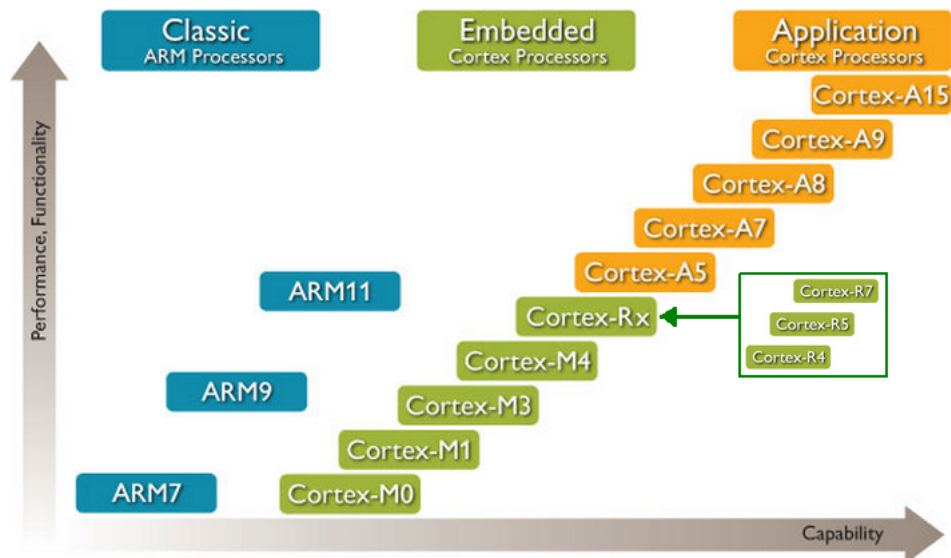
Základním komponentem nového laboratorního systému bude mikrokontrolér postavený na platformě ARM. V této kapitole je architektura ARM stručně popsána a následně se zabývá rešerší, která pomůže s výběrem vhodného mikrokontroléru.

Historie mikroprocesorů ARM sahá až do první poloviny osmdesátých let. Vývoj architektury byl veden dvojicí Roger Wilson a Steve Furber ze společnosti Acorn RISC Machine s cílem vyvinout mikroprocesor s velmi nízkou cenou v porovnání s rychlostí zpracování dat. První procesor byl uveden v roce 1984 s označením ARM1 a v následujícím roce také ARM2, který byl již postaven na 32 bitové architektuře. Dnes se využívají v různých elektronických zařízeních napříč různými odvětvími. Jednou z největších výhod je jejich poměrně malá energetická náročnost. Jak bylo již zmíněno, mikroprocesor je založen na 32 bitové architektuře s modifikovanou Harvardskou architekturou paměti. Pro dosažení maximálního výpočetního výkonu je použita instrukční sada typu RISC, která poskytuje sice menší počet instrukcí, ale zároveň je možné optimalizovat kód lépe, než je tomu u instrukční sady CISC [2] [3].

V současné době je na trhu velké množství mikroprocesorů různých modifikací. Tématu této práce bude však nejbliž skupina mikrokontrolérů s jádrem Cortex. Tuto skupinu lze rozdělit do následujících podskupin:

- ARM Cortex-A (Application)
- ARM Cortex-R (Real-time)
- ARM Cortex-M (Microcontrollers)

Pro obecnou představu o výkonu a vybavenosti mikroprocesorů Cortex, lze použít obrázek 2.1. Na pravé straně jsou zástupci skupiny Cortex-A a jak bylo uvedeno výše, jedná se o nejvýkonější jádra z celé skupiny Cortex. Tu následují z hlediska výkonu a vybavy zástupci řad Cortex-R a Cortex-M. Pro porovnání jsou v grafu vyznačeny i předchůdci řady Cortex. Jmenovitě se jedná o řady ARM7, ARM9 a ARM11. V současné době je poptávka po této starší generaci téměř nulová [4] [5]. Každá skupina se zaměřuje na jinou kategorii aplikací. Skupina Cortex-A se prezentuje nejvyšším výpočetním výkonem. Je dostupná v jedno, nebo vícejádrovém provedení. Využívá se v oblasti chytrých zařízení, jako jsou chytré telefony, televize, set-topboxy a další skupiny využívající pro svou funkci náročnější operační systémy. Patří sem také čipy určené pro servery (cloudové aplikace) Skupina ARM Cortex-R (Real-time) nabízí menší množství dostupných variant ale především je zaměřena na embedded real-time a safety-critical aplikace. V této oblasti jsou kladeny nároky jak na výkon, tak především na rychlé a deterministické chování při přerušení. Aplikace tohoto typu lze nalézt v oblasti zdravotnictví, robotiky, kontrolních jednotkách automobilů a průmyslových automatů. Posledním z jmenovaných je skupina Cortex-M



Obr. 2.1: Přehled zástupců mikroprocesorů řady Cortex a jejich předchůdců v závislosti na jejich výkonu a výbavě

a lze ji specifikovat jako nejméně výkonou. Avšak cenově velice zajímavou pro široké spektrum embedded zařízení. Pro spoustu aplikací nabízí dostatečný výkon a nízkou spotřebu. Nachází využití především v mikrokontrolérech s různým výpočetním výkonem a energetickými nároky. Aktuálně do kategorie Cortex-M spadá šest odlišných typů jader. Každá z uvedených architektur má odlišné vlastnosti a obecně i jinou instrukční sadu. Tabulka 2.1 popisuje rozdíly v instrukčních sadách jednotlivých jader. Kategorie Cortex-M nepodporuje původní sadu instrukcí s názvem A32. Namísto toho je podporována instrukční sada Thumb a u některých také Thumb-2. Lze například vyčíst, že podporu instrukcí pro zpracování digitálního signálu (DSP) využívají pouze jádra Cortex-M4 a Cortex-M7. V případě binární kompatibility mezi jednotlivými čipy se odvíjí od instrukční sady. Jak již bylo zmíněno, sada instrukcí není jednotná. Je možné ji ale vyčíst z tabulky. Programy přeložené pro Cortex-M0, M0+ a M1 budou kompatibilní jak mezi sebou, tak i pro ostatní jádra. V případě Cortex-M3, M4 a M7 je zajištěna vzájemná kompatibilita, ale není zajištěna pro Cortex-M0, M0+ a M1.

Řada Cortex-M bude nejlepší volbou pro budoucího systému. Mikrokontroléry s touto architekturou nejlépe odpovídají konceptu laboratorního systému, sloužícího pro výuku programování na požadované úrovni. Tabulka 2.2 shrnuje všechny současné varianty čady Cortex-M, jejich architekturu a také jednotku MPU [7].

#	Jádro	DSP	A32	Thumb	Thumb-2
1	Cortex M0	Ne	Ne	Bez 3 instrukcí	Částečně
2	Cortex M0+	Ne	Ne	Bez 3 instrukcí	Částečně
3	Cortex M1	Ne	Ne	Bez 3 instrukcí	Částečně
4	Cortex M3	Ano	Ne	Kompletní	Kompletní
5	Cortex M4	Ano	Ne	Kompletní	Kompletní
6	Cortex M7	Ano	Ne	Kompletní	Kompletní

Tab. 2.1: Přehled binární kompatibility všech jader Cortex-M

#	Jádro	Architektura ARM	Architektura CPU/MCU	MPU
1	Cortex M0	ARMv6-M	Von Neumann	Ne
2	Cortex M0+	ARMv6-M	Von Neumann	Volitelná
3	Cortex M1	ARMv6-M	Von Neumann	Ne
4	Cortex M3	ARMv7-M	Harvardská	Volitelná
5	Cortex M4	ARMv7E-M	Harvardská	Volitelná
6	Cortex M7	ARMv7E-M	Harvardská	Volitelná

Tab. 2.2: Přehled architektur CPU/MCU vzhledem k jednotlivým jádrům skupiny Cortex-M

2.1 Výběr mikrokontroléru s jádrem Cortex-M

Jelikož je na trhu dostupný opravdu velký počet mikrokontrolérů, tak před samotným vyhledáváním konkrétního čipu je nutné zadefinovat alespoň několik důležitých parametrů tak, aby se počáteční okruh potencionálních kandidátů co nejvíce zúžil. Jak vyplývá z výše uvedeného rozboru, vyhledávání zúžíme pouze na skupinu Cortex-M. Dalším důležitým parametrem, který vyplývá ze samotného zadání, je napěťová úroveň čipu, která má být 5V. Dá se předpokládat, že k základní desce bude připojeno současně několik periférií, aby byla zajištěna dostatečná variabilita možných zadání. Proto je nutné hledat čip s větším počtem I/O pinů pro připojení periférií. V nebude brán do úvahy požadavek na velikost programové paměti, jelikož rozsah tvořených projektů je s největší pravděpodobností nepřesáhne. Stejně tak rychlost CPU, která by pro plánované aplikace neměla být limitující. Výběr dále musí brát v potaz vnitřní periferie mikrokontroléru. Pro potřeby výuky musí být zahrnuty běžně používané komunikační rozhraní. Čip bude muset být vybaven následujícími vnitřními periferiemi:

- I²C
- SPI
- UART

Jako databázi bude vhodné využít některého z dodavatelů elektronických součástek.

Dostatečnou databází s požadovanými filtry disponuje webová stránka firmy Farnell a jelikož bude nutné později vybraný čip zakoupit, tak představuje pro naši oblast nejlepší volbu.

Při zadání výše uvedených parametrů typu jádra, napětí a počtu I/O v rozsahu 52 až 103 pinů se vyfiltruje několik desítek kandidátů od různých výrobců s různými pouzdry. Jmenovitě jsou to výrobci NXP, Microchip, Cypress Semiconductor, Fujitsu a Renesas. Systém bude nutné realizovat a oživit. Je nutné vybrat tedy pouzdro, které bude možné osadit ručně. Tomu odpovídají pouzdra LQFP, TQFP a QFP. Ne všechny z dostupných čipů se v této době prodávají a je možné je zakoupit, proto byly také odstraněny z výběru.

Výčet vyfiltrovaných mikrokontrolérů s výše zadanými vstupními parametry už je poměrně dost úzký. V seznamu jsou nyní čipy s architekturou Cortex-M0, Cortex-M0+, Cortex-M3 a Cortex-M4. Obecný výčet nalezených rodin mikrokontrolérů společně s jejich dostupnými variantami je uveden v tabulce 2.3.

#	Rodina MCU	Výrobce	Architektura	Rychlost CPU	Programová paměť	Počet I/O
1	PSoC5	Cypress Semiconductor	Cortex-M3	80MHz 67MHz	64KB 128KB 256KB	62 72
2	PSoC4	Cypress Semiconductor	Cortex-M0+	24MHz 48MHz	64KB 128KB	54
3	Kinetis E	NXP	Cortex M0+ Cortex M4	168MHz 160MHz 72MHz 48MHz 40MHz 20MHz	512KB 256KB 128KB 64KB 32KB	57 58 89
4	Kinetis EA	NXP	Cortex M0+	48MHz	64KB 128KB	58
5	S32	NXP	Cortex M0+	48MHz	256KB	58
6	Synergy	Renesas	Cortex M0+ Cortex M4	32MHz 48MHz	256KB	53 84
7	SAM 32	Microchip	Cortex M0+	48MHz	256KB	58 84

Tab. 2.3: Seznam rodin vybraných mikrokontrolérů a jejich parametry

Variant možných čipů, splňující výše zadané požadavky je několik. Jak již bylo zmíněno, výkon všech čipů je dostatečný. V tabulce lze sledovat poměrně velký rozptyl v počtu vyvedených I/O pinů. Tento parametr bude důležitý při návrhu

a realizaci konceptu základní desky a v případě vyššího počtu pinů bude nabízet větší univerzálnost, variabilitu možných aplikací a také nebude nutné celý systém několikrát v průběhu výuky hardwarově měnit. V tabulce je několik mikrokontrolérů, které mají až přes 80 vyvedených I/O pinů. Konkrétně se jedná o produktové řady:

- Kinetis E Family KE1x - 89 I/O
- SAM 32 Family SAM C 21 - 84 I/O
- Synergy Family S3 - 84 I/O

S přihlédnutím k dalšímu postupu práce je zajímavá myšlenka ohledně volby vývojového prostředí. Ve výčtu jsou následující produktové řady výrobců: Kinetis E Family - NXP, SAM 32 Family SAM C - Microchip, PSOC 5 Family - Cypress Semiconductor, PSOC 4 Family - Cypress Semiconductor, Synergy Family S1 - Renesas, Synergy Family S3 - Renesas. Výrobce NXP uvádí pro svou rodinu Kinetis E, že poskytuje podporu několika vývojových prostředí. Mezi kterými jsou například i Keil uVision a IAR Embedded Workbench. Vývojové prostředí firmy Keil využívá taktéž Rodiny PSOC 5 a PSOC 4 od Cypress Semiconductor. Firma Renesas využívá vlastní nadstavbu založenou nad multiplatformním prostředím Eclipse. Poslední z uvedených je produktová řada SAM 32 Family SAM C od firmy Microchip a výrobce doporučuje ke svým produktům prostředí Atmel Studio.

Pro přehlednost jsou v seznamu níže uvedeny produktové řady a k nim přiřazená vývojová prostředí, které doporučuje výrobce. Ne všechna prostředí uvedená výrobcem, budou vhodná pro účely výuky, proto jsou v seznamu ponechána pouze ta použitelná a nejznámější.

- PSOC 5 - Keil uVision
- PSOC 4 - Keil uVision
- Kinetis E - Keil uVision, IAR Embedded Workbench
- Kinetis EA - Keil uVision, IAR Embedded Workbench
- S32 - Keil uVision, IAR Embedded Workbench
- SAM 32 - Atmel Studio
- Synergy - Eclipse (nádstavba e2 studio), IAR Workbench

V seznamu jsou celkem 4 prostředí. V případě Keil uVision a IAR Embedded Workbench se jedná o placené produkty, které sice umožňují využívat i bezplatnou variantu, ale s několika omezeními, jako je například délka kódu a s ním spojená samostatná velikost programu. Pro aplikace vytvářené v laboratorní výuce nemusí být limitující, ale v seznamu se nachází také bezplatná vývojová prostředí od firmy Microchip (Atmel Studio) a Renesas (Eclipse). Pro multiplatformní Eclipse nabízí společnost Renesas vlastní bezplatnou nadstavbu s názvem e2. Posledním uvedeným v seznamu je Atmel Studio. Výrobce ho poskytuje zdarma a svým pojetím je velmi podobné populárnímu Visual Studiu od firmy Microsoft.

Závěrem byla z výše uvedených produktových řad vybrána produktová řada

SAM32 od firmy Microchip a to konkrétně čip ATSAMC21N18 s pouzdem TQFP o celkovém počtu 84 I/O. Dokumentace mikrokontroléru je v elektronické příloze této práce. Čip je vybaven všemi periferiemi pro potřeby výuky, poskytuje dostatečný počet I/O a je přímo podporován vývojovým prostředím Atmel Studio.

3 Koncept Laboratorního systému FEKTIS

V předchozí kapitole byl vybrán nejvhodnější čip pro nový systém. Po úvodní studii současného systému a především se znalostí periférií je možné přistoupit k návrh konceptu. Procesorová deska bude odstraněna a nahrazena novou. Rozšiřující desky vzhledem k novému konceptu již nebudou potřeba a samotné nemají žádný význam, jelikož jsou úzce spjaty s původním mikrokontrolérem.

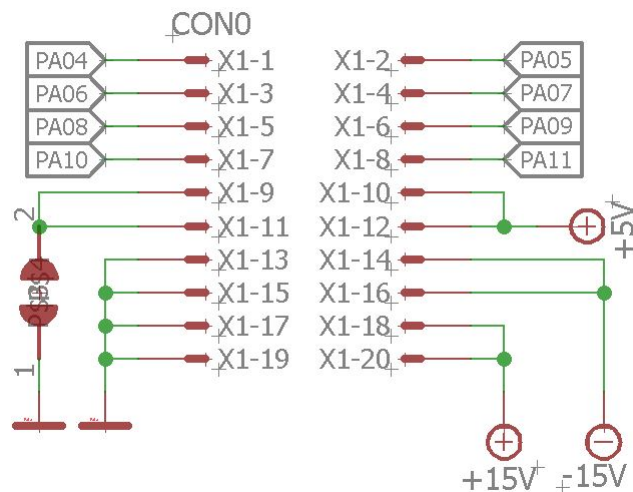
Systém a všechny jeho komponenty budou mít napětovou úroveň +5V, dále má být dostatečně univerzální, aby bylo možné rozšířit portfolio současných modulů a hlavně bude muset splňovat potřeby výuky ve stejném, nebo rozšířeném rozsahu oproti současnému konceptu. Následující podkapitoly budou diskutovat jednotlivé části vhodné pro nový systém a jejich zakomponování do celkového konceptu.

3.1 Definování blokových částí systému

Mikrokontrolér ATSAMC21N18A nabízí dostatečné množství interních periférií a také I/O pinů. Celkem disponuje počtem 84 I/O, což by mělo být dostatečné množství a nebude nutné do systému přidávat žádné moduly pro rozšíření jak tomu bylo u systému I51. Bude vhodné vyvést co největší počet pinů ke konektorům základní desky, aby mohly být připojeny jak současné, tak i nové koncové periferie. Z těchto poznatku vyplývá, že vhodným základním konceptem bude procesorová deska vybavená konektory pro připojení periférií a bude vhodné umístit některé nové moduly i na samotnou procesorovou desku. Pro potřeby výuky budou jistě přínosné periferie sériové komunikace USART, I²C, SPI a dále AD převodník a DA převodník. V dalších kapitolách budou tedy diskutovány možnosti jak tyto části vhodně do konceptu zapracovat.

3.2 Kompatibilita se současnými moduly

Z úvodní studie systému vyplynulo, že všechny koncové periferie mají stejné rozhraní a jelikož odpadají omezení, které byly spojené s původním mikroprocesorem, tak nebude problém připojit periférii k libovolnému univerzálnímu konektoru procesorové desky. Tyto konektory se budou tedy držet současného rozvržení pinů. Tímto bude zachována kompatibilita se současnými koncovými perifériemi. Na 3.1 je schématický návrh univerzálního konektoru.



Obr. 3.1: Univerzální konektor pro připojení koncových periférií

Koncová periférie LCD displeje potřebuje ke své činnosti celkem 11 pinů mikrokontroléru. To znamená, že nemůže být připojena pomocí univerzálního konektoru, který je společný u všech ostatních koncových jednotek. Jeden konektor bude muset být přímo přiřazen této periférii. Koncový modul není vybaven ani potřebným ovládáním kontrastu v podobě trimru, který byl v původním systému osazen na rozšiřovací desce. Tento poznatek bude muset být promítnut i v následné realizaci.

Modul I²C, podobně jako jednotka LCD displeje, se od ostatních odlišuje. Pro deska s I²C výbavou by mohlo být hodným řešením vytvoření dalšího speciálního konektoru, pouze pro potřeby tohoto modulu. Další variantou je umístění těchto komponent přímo na procesorovou desku.

Tyto a další části konceptu budou diskutovány v dalších kapitolách.

3.3 USART

Ke komunikaci slouží dvě linky – Rx a Tx. RS232 obsahuje ještě další linky RTS, CTS, DTR, které slouží k řízení toku dat. Tyto linky, ale v systému nejsou připojeny. Rx značí vstupní pin jednotky UART. Vývod označený Tx je výstupní pin jednotky UART. Vždy tedy existuje u zařízení samostatný přijímač a samostatný vysílač. Propojení dvou zařízení vždy vypadá tak, že Tx jednoho mikrokontroléru je připojen do Rx druhého zařízení. Komunikace může probíhat plně duplexně. To znamená, že oba spojené mikrokontroléry mohou kdykoli začít vysílat zprávu jeden druhému (i zároveň). V systému FEKTis se využívá asynchronní mód zařízení (UART) a modul UB232R s FTDI čipem, který zastává funkci převodníku z UART na RS232. Slouží

k propojení mikrokontroléru s PC pomocí USB rozhraní. Komunikace v asynchronním režimu probíhá následovně. Přijímač sleduje linku (RX pin). Jakmile se na ní objeví log. 0 (tzv. “start bit”), tak začne přijímač v pravidelných intervalech vzorkovat logickou úroveň na lince a uloží její hodnotu do zásobníku (bufferu). Jakmile zaznamená 8 hodnot, zastaví čtení. Vysílač i přijímač musí být předem nastaveny na stejné parametry. Musí vědět v jakém intervalu číst stav linky (baudrate), musí vědět jak dlouhá bude zpráva a dále je možné nastavit paritní bit a délku stop sekvence [8].

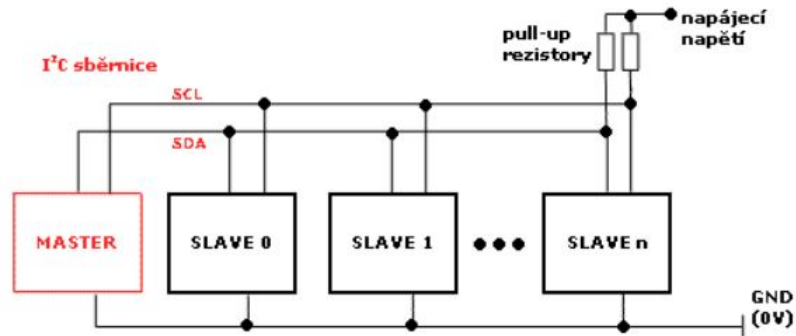
Tento modul bude určen pro komunikaci s PC a proto bude muset být vybaven převodníkem na RS232 stejně jak tomu bylo u původního systému.

3.4 I²C

I²C je zkratka která vznikla z IIC (Internal-Integrated-Circuit Bus). Jedná se o datovou sběrnici sloužící pro komunikaci a přenos dat mezi jednotlivými integrovanými obvody. Jde především o mikrokontroléry, sériové paměti, inteligentní LCD, audio a video obvody, a/d a d/a převodníky a další digitálně řízené obvody. Hlavní výhodou je, že obousměrný přenos probíhá pouze po dvou vodičích, SDA (serial data) a SCL (serial clock). To především u mikrokontrolérů výrazně optimalizuje nároky na počet I/O pinů a celkově zjednodušuje výsledné zapojení. Na jednu sběrnici může být připojeno více integrovaných obvodů. V základní verzi jsou obvody adresovány 7bitově a v rozšířené verzi 10bitově. To umožňuje připojení 128 až 1024 čipů s různou adresou na jednu společnou sběrnici. V praxi jsou tato čísla však podstatně nižší, protože adresa čipu většinou nelze určit plnými 7 (10) bity ale třeba jen třemi. Někdy nelze určit vůbec a je dána na pevně pro daný typ čipu, takových čipů tedy na jedné sběrnici nemůže být více než jeden. Přenosová rychlost sběrnice je pro většinu aplikací dostatečná i v základní verzi, kde je frekvence hodin 100kHz. Ve vylepšených verzích to může být 400kHz nebo 1MHz, ale ne všechny integrované obvody tuto verzi podporují. Rychlost přenosu pak musí být přizpůsobena nejpomalejšímu čipu na sběrnici. Oba vodiče musí být implicitně v logické jedničce a to je zajištěno pull-up rezistory. Jejich odpory mají hodnotu v řádech jednotek kiloohmů. Čím je vyšší komunikační frekvence, tím musí být nižší hodnoty těchto odporů. Pro 100kHz postačuje 4k7.

Jeden z integrovaných obvodů (většinou mikrokontrolér) je nastaven jako MASTER a všechny ostatní obvody jsou SLAVE. Obvody se dají zapojit i jako tzv. multi-master, kdy je čipů master několik. Master při jakémkoli přenosu generuje hodinový signál na vodiči SCL. Když jeden čip vysílá, přijímají všechny ostatní a pouze podle adresy určují, zda jsou data určena jim. Čip, který chce vyslat/přijmout data musí nejprve definovat adresu čipu, s kterým chce komunikovat a zda půjde o

čtení, nebo zápis. To určuje R/W (read/write) bit, který je součástí adresy. Obecné zapojení zařízení na sběrnici I²C je na obrázku 3.2.



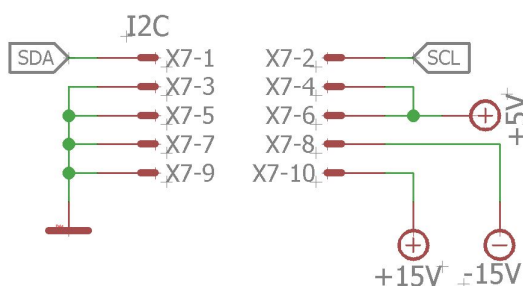
Obr. 3.2: Univerzální konektor pro připojení koncových periférií

Přenos probíhá v následujících krocích:

1. Stav klidu - Je zajištěn logickými jedničkami na obou vodičích, master tedy negeneruje hodinový signál a neprobíhá žádný přenos. Logické jedničky jsou na obou vodičích zajištěny pull-up rezistory, takže klidový stav nastane i pokud jsou výstupy obvodu master ve stavu vysoké impedance (tedy odpojeny).
2. Start bit - Zahajuje přenos. Je vygenerován tak, že se změní úroveň SDA z 1 na 0 zatímco je SCL v logické 1.
3. Stop bit - Ukončuje přenos. Je vygenerován podobně jako start bit. Logická úroveň SDA se změní z 0 na 1 zatímco je SCL v logické 1. Stop bit může být generován pouze po "nepotvrzení přenosu", tedy pouze po příjmutí Ack v logické 1.
4. Přenos dat - Data jsou přenášena po 8 po sobě jdoucích bitů. Při přenosu dat se může logická úroveň na SDA měnit pouze pokud je SCL v logické 0. Při každém pulzu na SCL je přenesen jeden bit.
5. Potvrzující bit Ack (acknowledge) - Tento bit slouží k potvrzení správného příjmutí dat. Ack bit se odesílá stejným způsobem jako by se odesílal devátý bit dat, ale s tím rozdílem, že ho generuje čip, který přijímal (Master uvolní linku). Pokud přenos proběhl v pořádku tak odešle logickou 0. Logická 0 potvrzujícího bitu znamená také to, že je přijímač připraven na příjem dalšího byte, který následuje okamžitě po něm při dalším pulzu na SCL. Pokud přenos selhal odešle logickou 1. Nebo pokud má dojít k ukončení přenosu, tak neodešle nic. Pull-up rezistor pak zajistí, že bude na SDA logická 1 a Ack bit (v logické 0) odešle vysílač [9].

Tento modul byl u původního systému realizován na periferní desce, připojené přímo k desce procesoru. Z důvodu nekompatibility konektorů, nebude moct být

tato deska připojena stejným způsobem jako ostatní moduly k univerzálním konektorům. Komunikace pomocí I²C je realizována pomocí dvou signálů, datového SDA a hodinového SCL. K těmto vodičům může být připojeno současně několik periférií a proto není problém vytvořit na desce konektor pro připojení zmiňovaného současného modulu. Současně nebude problém vybavit procesorovou desku dalšími komponenty komunikující pomocí I²C. Tímto způsobem bude řešena kompatibilita s původním modulem a zároveň bude zajištěn přístup k signálům SDA a SCL buď pro další rozšíření a nebo pro připojení logického analyzátoru umožňujícího sledování logických stavů na sběrnici. Náhled na konektor je na obrázku 3.3

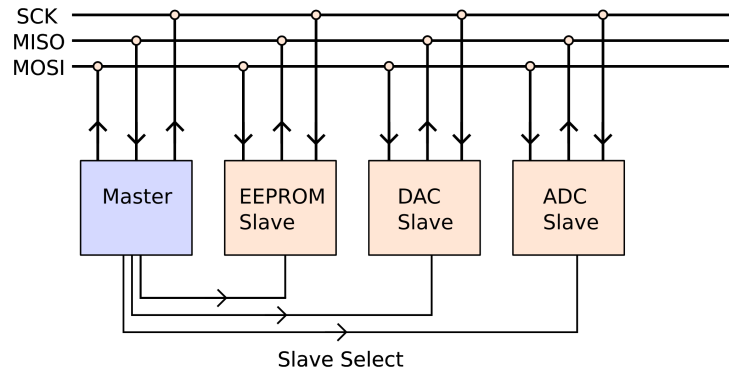


Obr. 3.3: I²C konektor na procesorové desce

3.5 SPI

Podobně jako rozhraní I²C, by měl v konceptu být zasazen i modul pro rozhraní SPI. Sběrnice SPI (Serial Peripheral Interface) představuje jednu z forem sériových sběrnic sloužících pro vzájemné propojení dvou či více komunikujících zařízení, přičemž jeden uzel obvykle zastává funkci master a ostatní uzly pracují v režimu slave. Zařízení, které pracuje jako master, obsahuje generátor hodinového signálu, který je rozveden do všech ostatních uzlů, čímž je umožněn zcela synchronní přenos dat. Hodinový signál je rozváděn vodičem označovaným symbolem SCK. Kromě vodiče s hodinovým signálem jsou zařízení propojeny dvojicí vodičů označovaných jako MISO (Master In, Slave Out) a MOSI (Master Out, Slave In), pomocí kterých se obousměrně (full duplex) přenáší data. Posledním signálem, který se u této sběrnice používá, je signál CS (Chip Select), jenž slouží k výběru zařízení pracujícího v režimu slave. Všechny čtyři signály – SCK, MISO, MOSI i CS, pro svoji funkci vyžadují pouze jednosměrné porty, což přispívá k jednoduché a především levné implementaci této sběrnice. Sběrnice SPI se kvůli své implementační jednoduchosti používá v mnoha různých systémech, například pro komunikaci s některými typy pamětí EEPROM, A/D a D/A převodníky, hodinami reálného času (RTC) atd [11]. Na

obrázku 3.4 je blokové propojení jednoho zařízení master a několika zařízení slave. Opět se nabízí možnost osazení komponent přímo na procesorovou desku a nebo na samostatný modul, který bude připojen pomocí univerzálního konektoru. Vhodnými moduly pro práci s rozhraním SPI by mohly být moduly pamětí, převodníků, digitálních potenciometrů atd.



Obr. 3.4: Propojení zařízení na sběrnici SPI

3.6 AD převodník

Dalším vhodným rozšířením systému je možnost práce s interním ADC (Analog to Digital Converter) mikrokontroléru. Jedná se o často využívaný komponent a proto by měl být zahrnut i do výuky. Ke správné funkci postačí připojit vstupní signál na vývod mikrokontroléru, který je interně připojen k periférii AD převodníku.

ADC převádí analogový signál na digitální podobu. Převodník mikrokontroléru ATSAMC21N má 12 bitové rozlišení a umožňuje vzorkování až do hodnoty 1MSPS. Lze jej zapojit jednovodičově, což je případ modulu na procesorové desce FEKTis, nebo diferenciálně. ADC převodník umožňuje převod znaménkový, nebo bezznaménkový. Mikrokontrolér obsahuje celkem 2 nezávislé převodníky ADC0 a ADC1 s až 12 kanály na jeden převodník. Spouštění měření lze řešit sepnutím z aplikace a nebo přijatou událostí z jiné periférie. Jako referenční napájení je možné využít interní, nebo externí zdroj napětí. Převodník může být nastaven na osmi, deseti, dvanáctibitovou hodnotu.

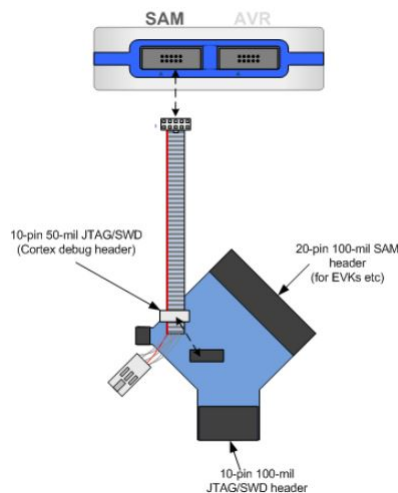
3.7 DA převodník

Interní periférie DA převodníku, která slouží k převodu digitálního signálu na analogový, by měla být součástí nového konceptu. Ve výbavě mikrokontroléru je jeden

desetibitový DAC (Digital to Analog Converter) s rychlostí vzorkování 350ksps. Tato jednotka je připojena k jedinému pinu pouzdra a tím je pin PA02. Aplikace by spočívala v generování analogového výstupu pomocí DAC a vyvedením tohoto signálu na měřicí svorky. Součástí zapojení by mohl být i operační zesilovač pro zesílení výstupního signálu.

3.8 Programátor Atmel-ICE

Jako vhodný programátor byl vybrán Atmel-ICE, který přímo podporuje zvolený mikrokontrolér a několik jich bylo už dříve zakoupeno. Atmel Studio přímo toto zařízení podporuje a lze jej okamžitě po připojení obsluhovat z prostředí Atmel Studio. Propojení mezi programátorem a cílovým zařízením je na obrázku 3.5. Pro připojení programátoru k PC slouží USB rozhraní.



Obr. 3.5: Propojení ATMEL-ICE a cílového zařízení

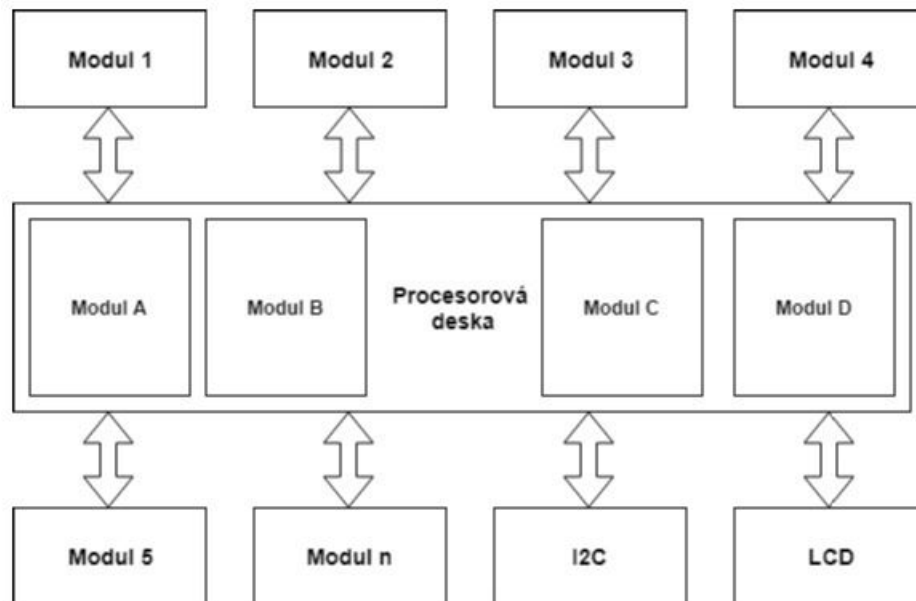
3.9 Výsledný model systému FEKTis

Z výše uvedených kapitol vyplývá několik poznatků, které budou směřovat následnou realizaci výsledného modelu. Z nich je nutné uvést tyto:

- Realizace napájení s úrovněmi napětí +5V, +15V a -15V
- Realizace konektoru pro připojení programátoru Atmel-ICE
- Realizace konektoru kompatibilního s LCD modulem
- Osazení trimru pro řízení kontrastu LCD modulu LCD
- Realizace konektoru kompatibilního s I²C modulem

- Realizace modulu pro rozhraní USART
- Realizace modulu pro práci s AD převodníkem
- Realizace modulu pro práci s DA převodníkem
- Realizace modulu pro rozhraní SPI
- Realizace maximálního počtu univerzálních konektorů v kompatibilní podobě se současnými koncovými moduly

Konceptem systému bude obecný model na obrázku 3.6.



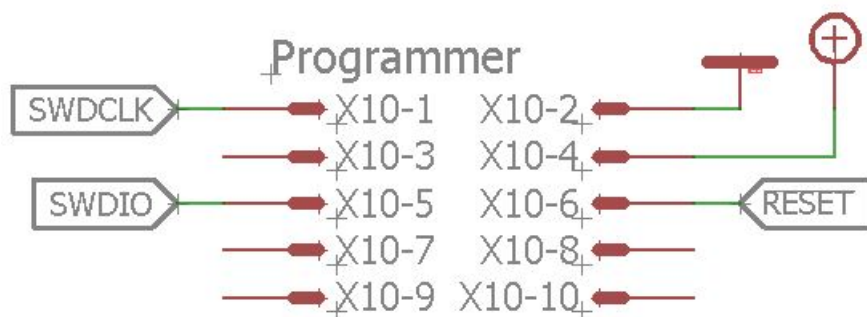
Obr. 3.6: Základní koncept propojení koncových zařízení s procesorovou deskou

4 Realizace procesorové desky systému FEK-Tis

Pro návrh schémat a DSP byl využit program EAGLE. Nejprve bylo nutné provést analýzu na základě dokumentace mikrokontroléru ohledně jednotlivých pinů a jejich možných funkcí. Piny mohou být interně propojeny s periferiemi mikrokontroléru a je vhodné určit takovou kombinaci, která by využila maximální potenciál všech 84 I/O pinů.

4.1 Rozvržení I/O pinů mikrokontroléru

První bylo určení pinů pro programátor, kterými jsou piny SWDIO a SWCLK. Pro tyto vývody existoval pouze jediná varianta a tou je pin PA31 pro SWDIO a PA30 pro SWCLK. Konektor pro připojení programátoru je na obrázku 4.1.



Obr. 4.1: Konektor pro připojení programátoru Atmel-ICE

V dalším kroku byly vybrány piny pro vytvoření I²C sběrnice. Mikrokontrolér disponuje celkem sedmi obecnými zařízeními pro sériovou komunikaci, které mohou být využity pro rozhraní USART, SPI, nebo I²C. V dokumentaci mikrokontroléru je uvedeno, ke kterému z obecných zařízení je možné interně připojit konkrétní piny. U tohoto čipu je možné přiřadit pro obsluhu obecného zařízení v módu I²C periferie pouze piny PA08, PA09, PA16, PA17, PB12, PB13, PB16 a PB17. Z tohoto seznamu byly vybrány piny PB12 pro signál SDA a PB13 pro signál SCL.

Další výběr byl ohledně pinů pro připojení periferie USART. Tato periferie může být realizována všemi obecnými zařízeními mikrokontroléru pro sériovou komunikaci. V tomto případě byly vybrány piny PC17 pro signál Rx a PC18 pro Tx.

Bylo rozhodnuto, že komponenty využívající sběrnici I²C budou umístěny přímo na procesorové desce a to z důvodu fyzických rozměrů samotné desky. Proto se v

případě vyvedení signálů pro rozhraní SPI rozhodlo, že bude vyvedeno na některý z univerzálních portů a komponenty budou na samostatné desce. Proto byla vytvořena osmice pinů, která splňuje požadavky univerzálního konektoru a zároveň poskytuje přístup k obecnému zařízení pro sériovou komunikaci, která může být nastaveno do módu SPI. Pro tento konektor byly vybrány piny PC9, PC10, PC11, PC12, PC13, PC14, PC15, PC16.

V případě modulu DAC byla volba jasná, jelikož k této periférii může být připojen pouze pin PA02.

Mikrokontrolér obsahuje 2 AD převodníky a lze k němu připojit až 12 kanálů. Pro potřeby ADC modulu však postačí pouze jeden pin a tím byl zvolen PB00.

V dalším kroku byl řešen požadavek na vytvoření konektoru s 11 piny vyvedenými z mikrokontroléru pro potřeby LCD modulu. Pro tento požadavek byly vybrány piny PB15, PB16, PB17, PB18, PB19, PB20, PB21, PB22, PB23, PB24 a PB25.

Tímto způsobem byly vybrány piny pro nové a původní periférie. Nakonec bylo ze zbývajících pinů vytvořen maximální počet univerzálních konektorů, tak aby vždy jedna celá osmice náleže k jednomu portu. Což zajistí usnadnění při nastavování příslušných registrů při programování. Nakonec byla vytvořena kombinace celkem sedmi osmic pinů mikrokontroléru, které mohou být vyvedeny k univerzálním konektorům.

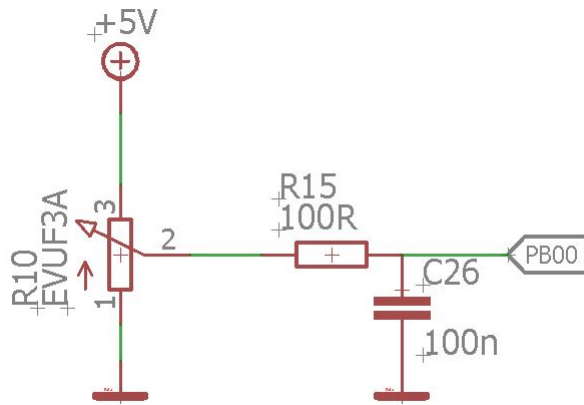
Ze zbylých pinů byly následně vyvedeny PA01 a PA02 pro připojení externího oscilátoru a piny PB01, PB02 a PB03 se využily pro připojení LED diod na procesorové desce. Modul LCD displeje byl nakonec doplněn ještě o další 3 vývody a to z toho důvodu, že je vybaven navíc třemi tlačítky. Z celkového počtu všech pinů zůstal nevyveden pouze jediný.

4.2 Realizace modulů

Zvolený koncept systému předpokládá vytvoření nových modulů a v této kapitole budou navrženy, realizovány a také budou vybrány konkrétní čipy pro jejich osazení.

4.2.1 Modul ADC

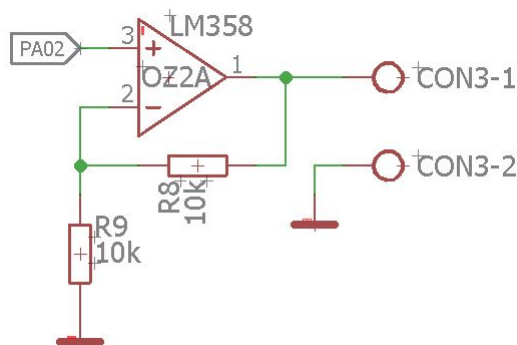
Tento jednoduchý modul umožňuje práci s AD převodníkem mikrokontroléru. A byl realizován pomocí potenciometru a napěťové úrovně +5V. Změnou polohy potenciometru se bude měnit i napětí přivedené na vstup AD převodníku. Schéma zapojení je na obrázku 4.2. Vynecháním potenciometru může být vytvořen konektor pro přivedení libovolného signálu, například z generátoru signálu.



Obr. 4.2: Schéma modulu ADC

4.2.2 Modul DAC

Tento modul je vybaven operačním zesilovačem LM358 s diferenciálním napájením pro zesílení výstupního signálu z DA převodníku mikrokontroléru. Výstup operačního zesilovače je připojen na svorky konektoru, kde může být kontrolována měřena aktuální hodnota pomocí měřícího přístroje. Schéma je na obrázku 4.3.



Obr. 4.3: Schéma modulu DAC

4.2.3 Modul USART

Modul pro práci se sériovým rozhraním USART, bude výhradně využíván pro komunikaci s PC. Proto byl vybrán hotový modul převodníku z USART na RS232, který je možné k PC připojit pomocí USB rozhraní. V PC bude možné otevřít jakýkoliv dostupný terminál pro sériovou komunikaci a systém FEKTis tak bude moct přijímat a odesílat data za pomoci vybraného modulu UB232R. Na procesorové desce

tedy byla vytvořena patice pro připojení modulu UB232R. Dokumentace tohoto čipu je v elektronické příloze této práce.

4.2.4 Modul I²C

Tento modul je možné rozdělit do tří částí, které jsou však připojeny na stejnou sběrnici I²C. První část s názvem I²C RTC je vybavena čipem RTC. Pro tyto účely byl vybrán čip. Další část s názvem I²C Memory Sockets je osazena čtyřmi univerzálními paticemi pro připojení pamětí. Každá patice má na adresových pinech (téměř všechny čipy tohoto typu mají stejné rozložení vývodů) přivedenou jinou hodnotu kombinace adresy. Pro začátek byl vybrán typ paměti EEPROM 24LC64 a FRAM FM24C16. Poslední část tvoří převodníky ze sériové na paralelní sběrnici a byly zvoleny čipy PCF8574T. Schématické zapojení na procesorové desce a dokumentace zmíněných čipů je v elektronické příloze práce.

4.3 Oživení systému

Procesorová deska je osazena celkem dvanácti konektory a nově vytvořenými moduly popsány v předchozích kapitolách. Po kompletním schématickém návrhu následovalo vytvoření PCB, které bylo odesláno do výroby a poté manuálně osazeno vybranými komponenty. Celé schéma včetně navrženého PCB je v elektronické příloze této práce.

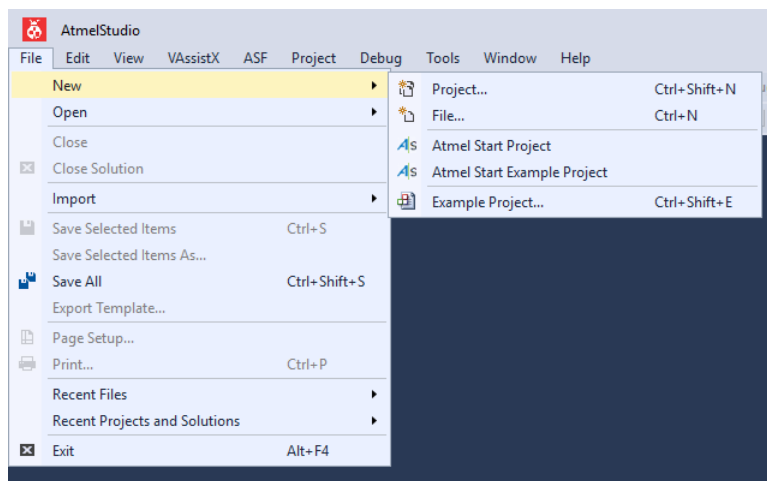
Po osazení se na přívodní svorky připojilo napájecí napětí s úrovní +5V, což je indikováno LED diodou. Byl připojen programátor. A ve vývojovém prostředí byl vytvořen nový projekt a následně nahrán jednoduchý program, který měl blikat LED diodou, připojenou na procesorové desce. Tento test proběhl úspěšně a bylo vyzkoušeno i online ladění programu za běhu aplikace. Byla tedy splněna počáteční podmínka pro další práci na nově vzniklém laboratorním systému FEKTis.

5 Vývojové prostředí Atmel Studio 7

Výběr vývojového prostředí byl již předem určen výběrem mikrokontroléru. Samozřejmě by mohly být nalezeny i alternativy, ale vývojové prostředí Atmel Studio bude i tak stále tím nejvhodnějším. Proto se tato kapitola zaměří na samotné prostředí, jeho součásti, zajímavé funkce při ladění kódu a práci s programátorem Atmel-ICE.

5.1 Tvorba projektu a nahrávání programu

Po spuštění programu je prvním krokem vytvoření nového projektu. To se provede otevřením rolovacího menu *New Project*, dále *New* a zvolení položky *Project* podle obrázku 5.1. Následně zvolit projektu název a typ projektu *GCC C Executable Project*. Jak vyplývá z názvu, projekt bude používat sadu překladačů *GNU Compiler Collection*.

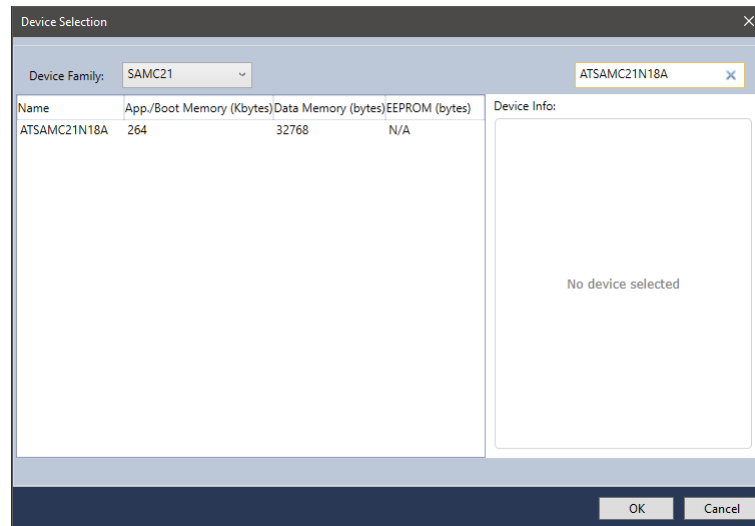


Obr. 5.1: Vytvoření nového projektu

Dalším je výběr cílového mikrokontroléru. To se provede výběrem čipu AT-SAMC21N18A z výběrového menu podle obrázku . Po potvrzení volby se již vytvoří stromová struktura projektu, včetně nahrání hlavičkových funkcí, které obsahují definované adresy všech komponent mikrokontroléru.

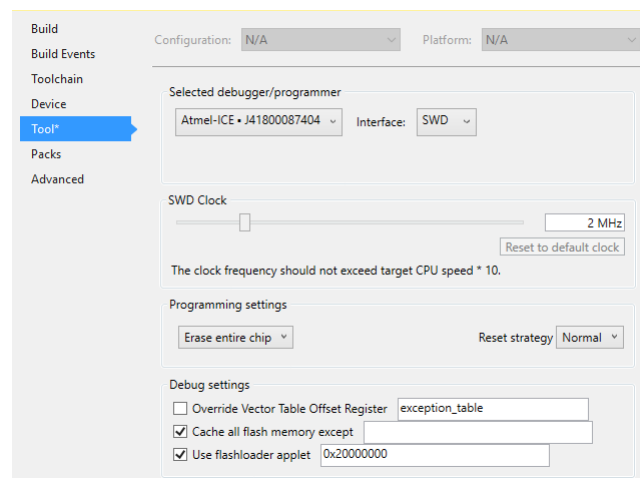
5.2 Ladění programu

Během vytváření každé aplikace je potřeba kód odladit (debugovat). Vybraný programátor neslouží pouze pro nahrávání přeloženého kódu do mikrokontroléru, ale



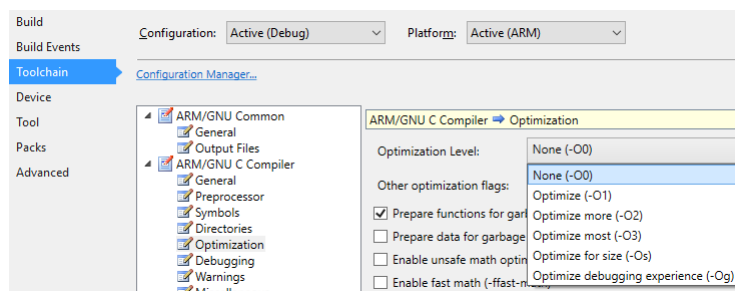
Obr. 5.2: Volba cílového mikrokontroléru

také umožňuje online ladění za běhu programu. Nejprve je však nutné zvolit v nastavení v záložce Tool v části *Selected debugger/programmer* zařízení Atmel-ICE. Jak lze vidět na obrázku 5.3.



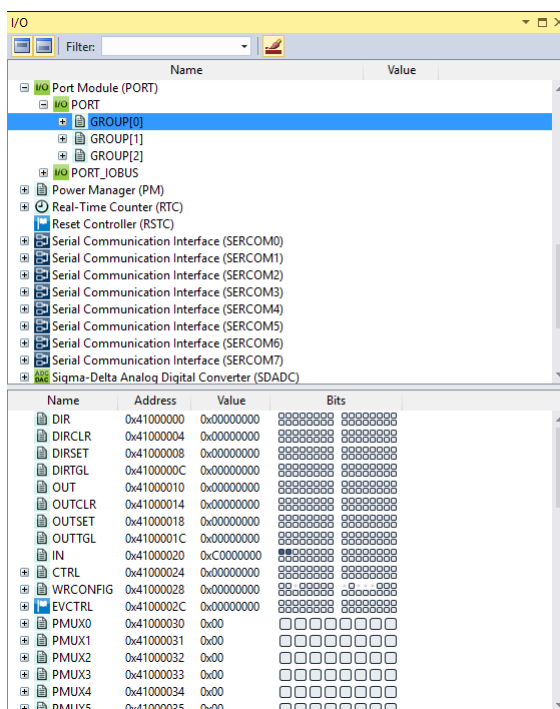
Obr. 5.3: Výběr zařízení pro programování a ladění

Nyní je možné program nahrát a ladit. Občas může dojít k situaci, že umístěný *breakpoint* na řádce aplikace je nedosažitelný a nebo není možné číst hodnoty lokálních proměnných. To může být způsobeno tím, že projekt má výchozí nastavení optimalizace kódu na úrovni "O1". Toto nastavení je dobré změnit a optimalizaci vypnout. To lze provést podle obrázku 5.4.



Obr. 5.4: Vypnutí optimalizace

Při nastavování registrů periferií je velice užitečným nástrojem modul s názvem "I/O", který zobrazuje všechny aktuální stavy registrů jednotlivých komponent.



Obr. 5.5: Modul pro zobrazení registrů komponent mikrokontroléru

6 Práce s laboratorním systémem FEKTis

Posledním bodem zadání je praktická část, která má zahrnovat práci s interními periferiemi mikroprocesoru a moduly systému FEKTis. Obsahuje již pouze konkrétní postupy jak s periferiemi pracovat a popíše metody a algoritmy potřebné pro SW realizaci požadovaných funkcí. Závěrem bude představeno několik demonstrativních úloh, které ověří správný návrh a realizaci systému a také ukáže několik možností jak se systémem pracovat. Prvních několik podkapitol se zaměří na to, jak nastavit a také jak pracovat s interními periferiemi mikrokontroléru, kterými jsou:

- GPIO
- USART
- I²C
- SPI
- ADC
- DAC

A bude popsáno jak se pracuje s jednotlivými koncovými moduly, které jsou součástí laboratorního systému FEKTis.

6.1 Obsluha vnitřních periférií mikrokontroléru

V této části je uveden nejprve teoretický popis konkrétních zařízení, dále implementace na mikrokontroléru ATSAMC21N a nakonec samotný popis inicializace a registrů, které souvisí s nastavením a ovládáním periferie.

6.1.1 Práce s GPIO

I/O Pin řadič, umožňuje ovládání I/O pinů zařízení. I/O piny jsou organizovány do skupin, které se označují port. Každý port má 32 pinů a každý může být ovládnán zvlášť a nebo společně v rámci celého portu. V případě mikrokontroléru ATSAMC21N18A lze přistupovat ke 3 portům. Každý pin může být využit jako obecný, tak aby jeho stav mohl být ovládnán/kontrolován přímo z aplikace a nebo může být přiřazen k interní periferii, např. UART, ADC, I²C. Pokud je využíván jako obecný, může být nastaven jako vstupní, výstupní a zároveň s možností nastavení interního pull rezistoru a dalších vlastností (push-pull). Všechny piny mohou být nastavovány i resetovány. Porty jsou připojeny na hodinové zdroje AHB, nebo APB. Všechny piny jsou nastaveny jako GPIO. Mají však i alternativní využití. Jak bylo zmíněno, porty a jejich pin, je možné přiřadit také k perifériím mikrokontroléru a je jim přiřazena funkce dle zvolené periferie. Seznam všech kombinací je uveden v dokumentaci mikrokontroléru a jsou označeny písmeny A až I. Pro povolení možnosti alternativní

funkce pinu musí být povolen bit v registru daného portu (PINCFGn.PMUXEN), kde n značí číslo pinu. Poté pro výběr konkrétní funkce je nutné nastavit registr PMUXn.PMUXE, nebo PMUXn.PMUXO podle toho, jestli je pin sudý, nebo lichý. Je nutné si uvědomit, že číslo pinu bude nyní děleno 2. Tato konfigurace následně připojí zvolenou část periferie k vybranému pinu a přemostí spojení mezi ním portem. Pro ovládání všech registrů mikrokontroléru obecně, slouží knihovna samc21n18a, kde lze najít všechny potřebné adresy registrů všech zařízení, včetně definovaných datových typů a konstant. Hlavičkový soubor lze najít ve složce s vytvořeným projektem. Dále je v hlavičkových souborech možné nalézt prototypy funkcí potřebné pro nastavení obsluhy přerušení. Dělení hlavičkových souborů je vytvořeno podle konkrétních instancí periférií mikrokontroléru. K samotnému ovládání pinů, je možné přistoupit několika způsoby. Pro názornost je uveden příklad. V hlavičkovém souboru "instance/port.h", jehož část je na obrázku 6.1, lze nalézt 3 podobné kopie registrů, které se liší pouze číslem na konci. Příslušné číslo značí pro kterou skupinu portů jsou registry určeny. Číslo 0 ukazuje na port A, číslo 1 na port B a číslo 3 na port C. V případě, že má být nastavena hodnota registru, která nastavuje pin PB30 jako výstupní, je třeba nastavit na adresu s názvem REG_PORT_DIR1 logickou 1 a pro volbu logické úrovně na pinu použít definici REG_PORT_OUT1. Zápisem kódu z obrázku 6.2 bude nastavena logická 0 na výstupní pin PB30.

```

76 #define REG_PORT_DIR0      (*(RwReg *)0x41004400U) /**< \brief (PORT) Data Direction 0 */
77 #define REG_PORT_DIRCLR0  (*(RwReg *)0x41004404U) /**< \brief (PORT) Data Direction Clear 0 */
78 #define REG_PORT_DIRSET0  (*(RwReg *)0x41004408U) /**< \brief (PORT) Data Direction Set 0 */
79 #define REG_PORT_DIRTGL0  (*(RwReg *)0x4100440CU) /**< \brief (PORT) Data Direction Toggle 0 */
80 #define REG_PORT_OUT0     (*(RwReg *)0x41004410U) /**< \brief (PORT) Data Output Value 0 */
81 #define REG_PORT_OUTCLR0  (*(RwReg *)0x41004414U) /**< \brief (PORT) Data Output Value Clear 0 */
82 #define REG_PORT_OUTSET0  (*(RwReg *)0x41004418U) /**< \brief (PORT) Data Output Value Set 0 */
83 #define REG_PORT_OUTTGL0  (*(RwReg *)0x4100441CU) /**< \brief (PORT) Data Output Value Toggle 0 */
84 #define REG_PORT_IN0     (*(RoReg *)0x41004420U) /**< \brief (PORT) Data Input Value 0 */
85 #define REG_PORT_CTRL0    (*(RwReg *)0x41004424U) /**< \brief (PORT) Control 0 */
86 #define REG_PORT_WRCFG0   (*(WoReg *)0x41004428U) /**< \brief (PORT) Write Configuration 0 */
87 #define REG_PORT_PMUX0    (*(RwReg *)0x41004430U) /**< \brief (PORT) Peripheral Multiplexing 0 */
88 #define REG_PORT_PINCFG0  (*(RwReg *)0x41004440U) /**< \brief (PORT) Pin Configuration 0 */
89 #define REG_PORT_DIR1     (*(RwReg *)0x41004480U) /**< \brief (PORT) Data Direction 1 */
90 #define REG_PORT_DIRCLR1  (*(RwReg *)0x41004484U) /**< \brief (PORT) Data Direction Clear 1 */
91 #define REG_PORT_DIRSET1  (*(RwReg *)0x41004488U) /**< \brief (PORT) Data Direction Set 1 */
92 #define REG_PORT_DIRTGL1  (*(RwReg *)0x4100448CU) /**< \brief (PORT) Data Direction Toggle 1 */
93 #define REG_PORT_OUT1     (*(RwReg *)0x41004490U) /**< \brief (PORT) Data Output Value 1 */
94 #define REG_PORT_OUTCLR1  (*(RwReg *)0x41004494U) /**< \brief (PORT) Data Output Value Clear 1 */
95 #define REG_PORT_OUTSET1  (*(RwReg *)0x41004498U) /**< \brief (PORT) Data Output Value Set 1 */
96 #define REG_PORT_OUTTGL1  (*(RwReg *)0x4100449CU) /**< \brief (PORT) Data Output Value Toggle 1 */
97 #define REG_PORT_IN1     (*(RoReg *)0x410044A0U) /**< \brief (PORT) Data Input Value 1 */
98 #define REG_PORT_CTRL1    (*(RwReg *)0x410044A4U) /**< \brief (PORT) Control 1 */
99 #define REG_PORT_WRCFG1   (*(WoReg *)0x410044A8U) /**< \brief (PORT) Write Configuration 1 */
100 #define REG_PORT_PMUX1    (*(RwReg *)0x410044B0U) /**< \brief (PORT) Peripheral Multiplexing 1 */

```

Obr. 6.1: Kód hlavičkového souboru port.h

Další možností, je přístup k portům a jejich pinům jako ke struktuře. Následující příklad využívá vlastností struktur, jak je definuje jazyk C. Ze skupiny číslo 1 (A) je vybrán registr PINCFG, který nastavuje vlastnosti pinu a odkazuje na prvek číslo 15, který reprezentuje daný pin v portu A. Nastaví ho jako vstupní a s možností


```
REG_PORT_DIR1 |= ( 1 << 30 );
REG_PORT_OUT1 &= ~( 1 << 30 );
```

Obr. 6.2: Kód hlavičkového souboru port.h

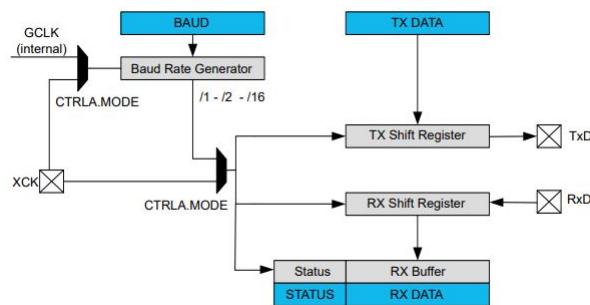
využití pull rezistoru. Další řádek ukazuje nastavení pull-up rezistoru. Takto lze nastavit pin PB30 jako vstupní. Tento postup je přepsán do kódu na obrázku 6.3.

```
PORT->Group[0].PINCFG[15].reg = PORT_PINCFG_INEN | PORT_PINCFG_PULLEN;
REG_PORT_OUTSET0 = SW0;
```

Obr. 6.3: Kód hlavičkového souboru port.h

6.1.2 Modul USART

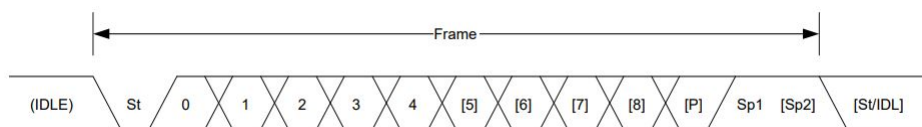
Blokové schéma zapojení periferie USART mikrokontroléru ATSAMC21N je na obrázku 6.4 a odráží teoretické informace.



Obr. 6.4: Blokové schéma zapojení periferie USART

Generování baudrate je realizováno pomocí vnitřního oscilátoru a děličky frekvence. Piny RxD a TxD jsou připojeny ke svým posuvným registrům a je zde znázorněn i signál XCK, který je využíván v synchronním módu. Bloky CTRLA.MODE představují vnitřní nastavitelný registr. Celý rámec jedné zprávy je na obrázku 6.5 a dle nastavení se může skládat z jednoho start bitu, datové sekvence v rozsahu 5 až 9 bitů, paritního bitu a stop sekvence v rozsahu 1 až 2 bity.

Každé z dostupných obecných sériových zařízení, může být nastaveno jako rozhraní UART. Dle dokumentace lze nalézt kombinace pinů, které jsou připojeny ke konkrétnímu zařízení SERCOM. Nastavení pinu pro TxD bude výstupní a naopak pro RxD bude vstupní. Inicializace obecného zařízení SERCOM pro definici UART by měla probíhat v následujících krocích:



Obr. 6.5: Rámec zprávy na sériové sběrnici UART

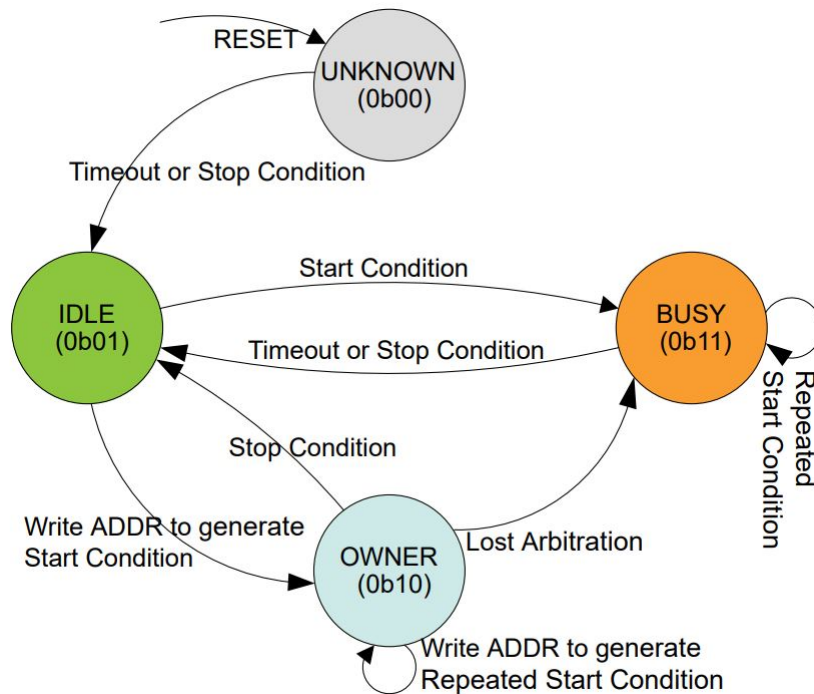
1. Výběr externího (0x0), nebo interního (0x1) zdroje hodin. Provádí se zápisem do oblasti registru CTRLA (CTRLA.MODE).
2. Výběr synchronního (0x1), nebo asynchronního (0x0) módu komunikace. Zápis do oblasti registru CTRLA (CTRLA.CMODE).
3. Fyzický výběr pinu pro příjem (RxD), zapsáním hodnoty do registru CTRLA (CTRLA.RXPO).
4. Fyzický výběr pinu pro odesílání (TxD), zapsáním hodnoty do registru CTRLA (CTRLA.TXPO).
5. Nastavení délky datové sekvence zapsáním hodnoty do registru CTRLB (CTRLB.CHSIZE).
6. Výběr pořadí odesílání bitů (první bude odesílán MSB, nebo LSB). Zápis do registru CTRLB. (CTRLB.DORD).
7. Povolení partiního bitu zapsáním do CTRLA registru (CTRLA.FORM) a případně nastavení typu parity (lichá/sudá) zápisem do registru CTRLB (CTRLB.PMODE).
8. Nastavení délky stop sekvence v registru CTRLB (CTRLB.SBMODE).
9. Nastavení baudrate v registru CTRLBAUD.
10. Povolení přijímání a odesílání se provede nastavením bitů v registru CTRLB (CTRLB.TXEN a CTRLB.RXEN).

Registr pro zápis baudrate je v případě asynchronního módu šestnáctibitový a v případě synchronního módu osmibitový. Výpočet baudrate je reprezentován následujícími vztahy na obrázku 6.6 [12].

Operating Mode	Condition	Baud Rate (Bits Per Second)	BAUD Register Value Calculation
Asynchronous Arithmetic	$f_{BAUD} \leq \frac{f_{ref}}{16}$	$f_{BAUD} = \frac{f_{ref}}{16} \left(1 - \frac{BAUD}{65536}\right)$	$BAUD = 65536 \cdot \left(1 - 16 \cdot \frac{f_{BAUD}}{f_{ref}}\right)$
Synchronous	$f_{BAUD} \leq \frac{f_{ref}}{2}$	$f_{BAUD} = \frac{f_{ref}}{2 \cdot (BAUD + 1)}$	$BAUD = \frac{f_{ref}}{2 \cdot f_{BAUD}} - 1$

Obr. 6.6: Vztahy pro výpočet baudrate

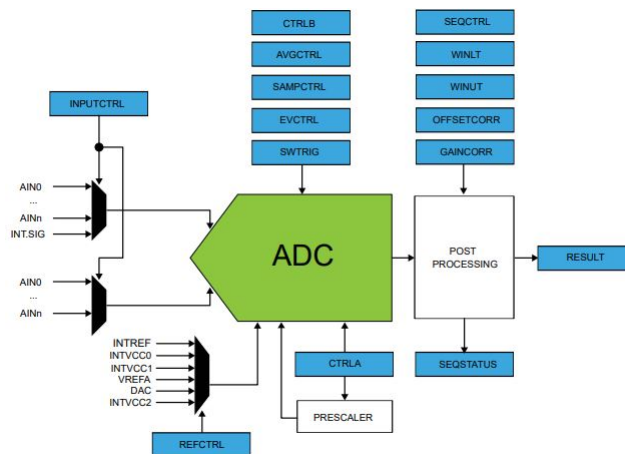
4. Povolení periferie SERCOM, zápisem bitu do registru CTRLA (CTRLA.ENABLE) Chování I²C zařízení je možné definovat stavovým popisem, jenž je na obrázku 6.8. Tento stavový automat je aktivní v případě, že zařízení je v módu master. Vstupním stavem je stav UNKNOWN, ze kterého přechází do stavu IDLE, což repretuje zápis hodnoty 0x1 do registru STATUS.BUSSTATE. Do stavu IDLE se automat také dostane v případě stop podmínky, nebo v případě vypršení inactive-timeout na sběrnici. Start podmínka se generuje zápisem adresy do registru ADDR.ADDR během stavu IDLE. Po zápisu přejde na stav OWNER a v registru STATUS.BUSSTATE bude hodnota 0x2. V případě, že nedošlo ke kolizi, pokračuje po dokončení vygenerování stop podmínky a automat přejde do stavu IDLE. V případě, že dojde ke kolizi, přejde na stav BUSY dokud se detekuje stop podmínka nebo než vyprší time-out [12].



Obr. 6.8: Stavový automat I²C periferie

6.1.4 Periferie ADC převodníku

Blokové schéma na 6.9 popisuje zapojení všech potřebných komponent pro jeho funkci. Pro svou funkci potřebuje hodinový signál CLK_APB_ADCx a GCLK_ADCx, které musí být aktivovány před spuštěním ADC. Piny, ke kterým bude analogový signál připojen jsou uvedeny v dokumentaci a opět je nutné připojit interně pomocí přepínače daný pin k periferii ADC.



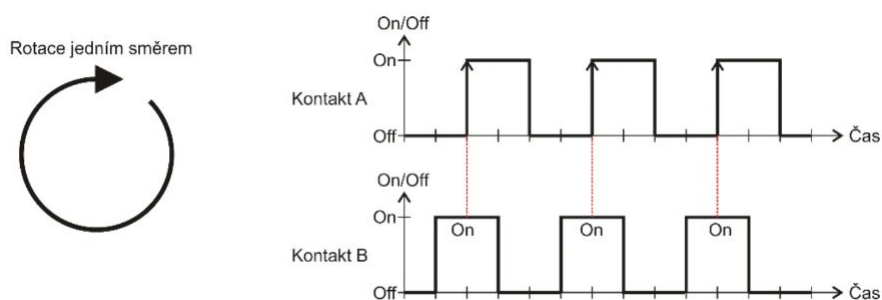
Obr. 6.9: Blokové schéma periferie ADC

Nastavení ADC se provádí pomocí několika registrů. Převodník se spouští nastavením pinu CTRLA.ENABLE. Rychlost převodu závisí na frekvenci hodinového signálu z GCLK_ADCx a nastavené předděličky, nastavení předděličky se provede zápisem do registru (CTRLB.PRESCALER), která může snížit vzorkovací frekvenci. Samotné spuštění převodu se provádí nastavením bitu v registru (SWTRIG.START), nebo automaticky v případě automatického spuštění. Výsledek se ukládá do registru RESULT. V případě, kdy bude k převodníku připojeno více kanálů, může nastat, že přepíše hodnotu ještě dřív, než se přečte. Aby se tomu dalo vyvarovat, tak je možné pomocí příznaku v registru (INTFLAG.RESRDY) použít přerušování a okamžitě hodnotu z registru RESULT přečíst. Registr REFCTRL.REFSEL rozhoduje o použitém referenčním napětí převodníku a je možné vybrat z interního zdroje napětí, nebo externího zdroje napětí. Rozlišení převodníku se určí zápisem do registru (CTRLC.RESSEL). Výchozím nastavením je hodnota 12 bitů. Dále je možné vybírat mezi jednovodičovým, nebo diferenciálním zapojením. To se provede zápisem do oblasti registru (CTRL.DIFFMODE). Mimo uvedené, má ADC převodník tohoto mikrokontroléru ještě další možnosti nastavení, které ale nejsou pro potřeby práce relevantní [12].

6.1.5 Ovládání Rotačního ovladače

Rotační ovladač na koncové periférii je vybaven kontakty A a B. Navíc je vyvedeno i tlačítko, ale to nemá na funkci žádný vliv. 6.10 představuje časový průběh logických úrovní na kontaktech, při otáčení po směru hodin. Při otáčení proti směru hodin by se signály na kontaktech předbíhaly v opačném sledu. Vyhodnocování se provede porovnáním známého průběhu uloženého v paměti a vzorku po sobě jdoucích sekvencí

zaznamenaných při změně logické úrovně na kontaktech.



Obr. 6.10: Průběh logických úrovní na kontaktech rotačního ovladače při otáčení po směru hodin

6.1.6 Ovládání DC motoru

Stejnoseměrný motor je připojen k obvodu známého jako H-můstek. Nejběžnějším způsobem jak regulovat rychlost otáčení a jeho směr je přivedením signálu PWM na vstup H-můstku. Směr otáčení se poté určí podle toho, na který ze vstupů H-můstku byl signál přiveden.

6.1.7 Ovládání krokového motoru

Ke krokovému motoru na koncové periférii jsou přes čip ULN2803A (Darlingtonovo pole) přivedeny 4 výstupní piny mikrokontroléru. Fáze jsou připojeny k cívkám a mohou se libovolně spínat (přivádět proud) pomocí nastavení nebo resetování připojených pinů. Řízení směru otáčení je možné řešit pomocí několik způsobů. Existují možnosti spínání jednofázově, kdy je v jeden okamžik zapnuta jen jedna fáze a nebo dvoufázově, kdy jsou v jeden okamžik zapnuty dvě vedlejší fáze motoru. Také možnosti řízení může být buď čtyřtaktní a nebo osmitaktní kdy se spínají vždy fáze jedné a dvou vedlejších fází. Směr otáčení udává posloupnost spínaných fází. Obrázek 6.11 ukazuje jak postupnými kroky dosáhnout otáčení jedním a nebo druhým směrem v dvoufázovém čtyřtaktním módu.

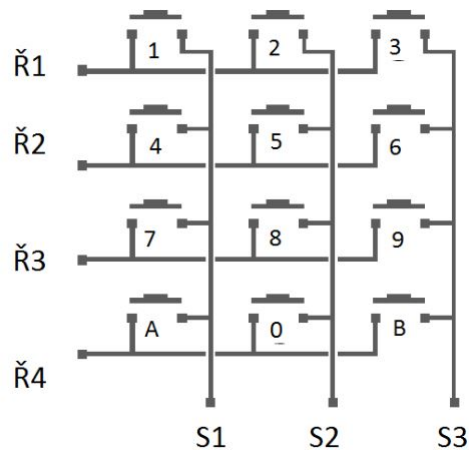
Takt	F1	F2	F3	F4
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
1	1	0	0	0
...				

Takt	F1	F2	F3	F4
1	0	0	0	1
2	0	0	1	0
3	0	1	0	0
4	1	0	0	0
1	0	0	0	1
...				

Obr. 6.11: Posloupnost kroků pro otáčení krokového motoru po směru a proti směru hodin

6.1.8 Ovládání maticové klávesnice 4x4

Maticová klávesnice osazená na koncové periférii má 4 sloupce a 4 řádky jak vyplývá z názvu. Obecně jsou maticové klávesnice zapojeny podle obrázku 6.12. Modul obsahuje 16 tlačítek a aby nemuselo být každé tlačítko připojeno k jednomu pinu mikrokontroléru, tak se obecně využívá pro získání aktuální pozice stisknutého tlačítka následovný algoritmus. Řádky matice jsou připojeny ke čtyřem výstupním pinům mikrokontroléru a postupně se do logické 1 překlopí pouze jeden pin. Takto se cyklicky s definovaným časovým intervalem opakuje přepínání pro všechny řádky matice. Další 4 vstupní piny jsou připojeny k sloupcům a čekají dokud se na jeden ze vstupů nepřivede logická 1. Po této události se přečte aktuální aktivní řádek a také se zaznamená konkrétní pin na, který byla přivedena logická 1. S touto dvojicí hodnot je pak přesně definovat přesnou pozici stisklého tlačítka. Je také vhodné ošetřit dlouhý stisk tlačítka pomocí podmínky, která po konci převodu čeká, dokud nejsou všechny vstupní piny v logické 0.



Obr. 6.12: Schéma zaojení maticové klávesnice

6.1.9 Ovládání LCD displeje

Obecně lze ovládat LCD displej za pomoci osmi a nebo čtyř paralelních datových vstupů. Dále je k ovládání za potřeby řízení pomocí signálů Registr Select (RS), Read/Write (R/W) a Enable (E). Displej má kromě vypisování znaků také několik řídicích funkcí popsaných v dokumentaci. Přivedení dat na datové vstupy je prvním krokem při zadávání příkazů jednotce displeje. Signál RS určuje, jak s přivedenými daty jednotka bude dále pracovat. Pokud se bude jednat o data pro výpis na displej tak bude signál RS v logické 1. V případě, že data mají být jednotkou zpracována jako příkaz (vymazání displeje, nastavení módu zobrazení), tak bude signál RS v logické 0. Řídicí signál R/W udává jestli se bude z jednotky číst a nebo do ní zapisovat. A nakonec se po přivedení všech signálů dá příkaz, přivedením sestupné hrany na pin E, k vykonání procesu s daty přivedenými na vstupy.

6.2 Vzorové úlohy

V poslední kapitole byly uvedeny všechny teoretické informace o konkrétních modulech, které systém FEKTis obsahuje. Po seznámení se všemi výše představenými interními periferiemi mikrokontroléru a stejně tak se znalostmi o tom, jak řídit jednotlivé externí periferie, nebo vyhodnocovat jejich "chování", je možné postupně realizovat několik konkrétních zadání. Předem nebylo definováno žádné zadání a proto bylo vytvořeno několik slovních úloh se záměrem demonstrace práce se systémem FEKTis v rozsahu, který by měl odpovídat výuce v laboratorních cvičeních. Práce

se v následujících kapitolách zaměřuje na konkrétní realizaci vybraných úkolů s definovaným zadáním. Aby nebyly představovány všechny realizace pro každou periférii zvlášť, tak následující úlohy jsou již kombinací několika předem připravených příkladů. V řešeních jsou uváděny postupy řešení za pomoci funkcí, jejichž konstrukce lze nalézt v elektronické příloze této práce.

6.2.1 Čtení maticové klávesnice 4x4 s prezentací dat na LCD displeji a odesílání do PC

V tomto příkladu je realizováno zadání v následujícím znění. Realizujte funkci čtení znaků z maticové klávesnice 4x4 připojené na libovolný univerzální konektor procesorové desky a následně zobrazujte stisknuté znaky odpovídající popisům na klávesnici na připojeném LCD displeji. Současně připojte systém FEKTis k PC. inicializujte periférii komunikačního rozhraní UART s těmito parametry:

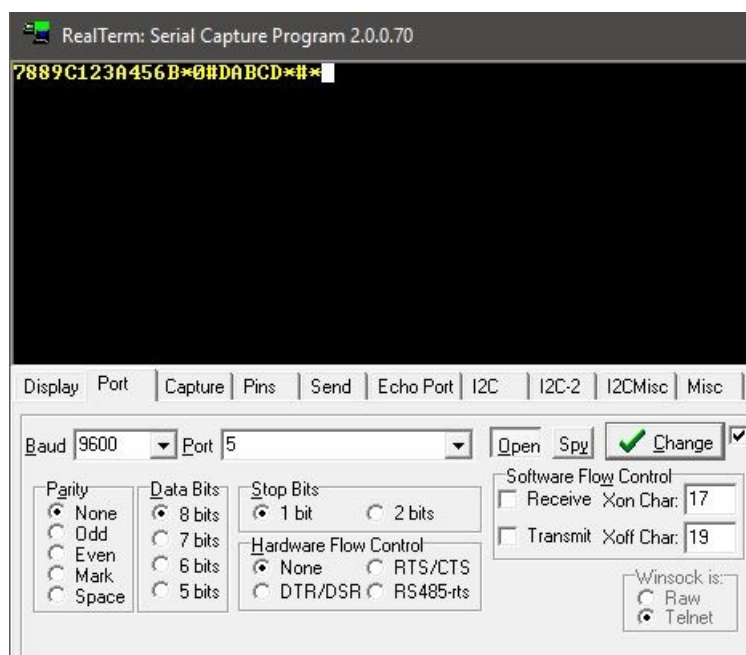
- Baudrate: 9600
- Paritní bit: Ne
- Stop bit: 1
- Datových bitů: 8

Po inicializaci periférii funkcemi spusťte a odesílejte stisknuté znaky přes sériovou linku do terminálu PC.

Realizování tohoto úkolu bylo následovné. Maticová klávesnice je připojena ke konektorům CON0 pro řízení řádků a konektoru CON1 pro čtení sloupců. LCD displej má vlastní konektor LCD. Po nastavení vlastností I/O pinů funkcemi *MatrixKeyboard_Init* a *LCD_Init* přichází na řadu nastavení sériové linky pomocí funkce *UART_Init* na zadáním požadované parametry. Všechny inicializační funkce jsou uvedeny v elektronické příloze práce. Algoritmus detekce stisknutého tlačítka na maticové klávesnici byl uveden v teoretické průpravě a pro jeho realizaci byly vytvořeny funkce *MatrixKeyboard_GetKeyPressed*, *MatrixKeyboard_GetColumnIndex* a pro převedení na konkrétní znak byla vytvořena tabulka *Matrix_Dictionary*. Funkce *MatrixKeyboard_GetKeyPressed* vrací již hodnotu ASCII znaku stisknutého tlačítka. V dalším kroku má tuto hodnotu mikrokontrolér odeslat na LCD displej kde má být zobrazována celá posloupnost stisknutých znaků. Řízení displeje obstarávají funkce *LCD_Cmd*, pro zadávání příkazů LCD displeji a funkce *LCD_Send_Char*, která odesílá jednotlivé znaky určené k zobrazení. Podle zadání mají být znaky současně odesílány po sériové lince do terminálu PC. K tomu se využívá funkce *UART_Putc*. Ta odešle ASCII hodnotu jednoho znaku. Kód hlavní aplikace je v elektronické příloze práce. Běh aplikace zachycují obrázky 6.13 a 6.14



Obr. 6.13: Výsledek běhu programu příkladu č. 1 na LCD displeji



Obr. 6.14: Výsledek běhu programu příkladu č. 1 na PC

6.2.2 Ovládání krokového motoru pomocí rotačního ovladače

V tomto příkladu je realizováno zadání v následujícím znění. Připojte k procesorové desce modul s rotačním ovladačem a krokovým motorem. Rotační ovladač bude zastávat funkci vstupní jednotky, která bude vracet mikrokontroléru informace o změně stavu. Aplikovaný algoritmus musí vyhodnotit změnu otočení a jeho směr. Tuto informaci použije k rotaci krokového motoru o jeden krok ve stejném směru v

jakém se pootočil rotační ovladač.

Řešení tohoto úkolu bylo realizováno následovně. Modul rotačního ovladače byl připojen ke konektoru CON1 a modul s krokovým motorem ke konektoru CON2. Dále byly identifikovány potřebné piny a k nim příslušné vývody obou modulů. Následuje nastavení pinů mikrokontroléru. V případě ovladače na vstupní a v případě motoru na výstupní. Dále vytvoření funkce pro detekci změny stavu a směru otočení ovladače s názvem *Rotary_Process*, která využívá tabulku stavů s názvem *Rotary_State_Table*, jejíž stavy vrací zpět hlavní aplikaci. Pro řízení rotace krokového motoru jsou vytvořeny funkce *StepMotor_Step_CW()* a *StepMotor_Step_CCW()*. Funkce jsou navrženy pro čtyřtaktní řízení a přepínají fáze tak, aby se motor točil v jednom a nebo druhém směru. Spojením výše uvedených funkcí podle následujícího kódu bude realizovat požadavky zadání. Funkčnost této aplikace byla ověřena.

6.2.3 Prezentace dat přijatých z PC pomocí sériově-paralelního I²C převodníku

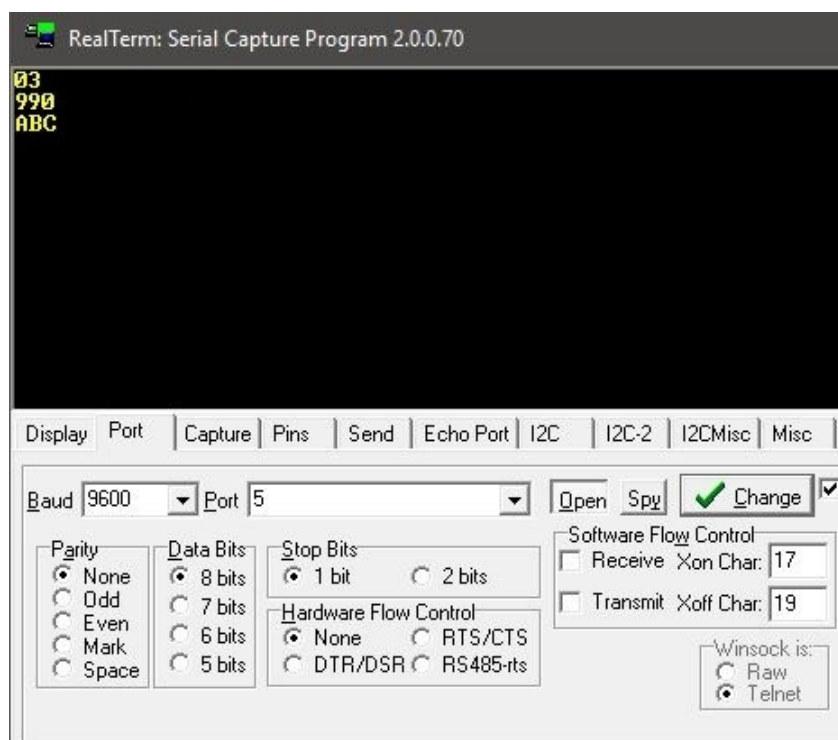
V tomto příkladu je realizováno zadání v následujícím znění. Přes sériovou linku přijme mikrokontrolér data, která budou postupně zobrazeny v binární podobě na LED modulu. LED modul bude připojen přes konektor, který obsluhuje sériově-paralelní převodník s rozhraním I²C. Uživatel PC bude osílat postupně 3 zprávy. Zadá kolik vzorů bude chtít na LED modulu zobrazit, s jakým intervalem se bude posloupnost vzorů měnit a nakonec odešle všechny vzory v jedné zprávě. Po této inicializaci mikrokontrolér zajistí přenos po sériové lince I²C do převodníku a zobrazí postupně všechna data. Po ukončení čeká aplikace na další vstup z PC. inicializujte periférii komunikačního rozhraní UART s těmito parametry:

- Baudrate: 9600
- Paritní bit: Ne
- Stop bit: 1
- Datových bitů: 8

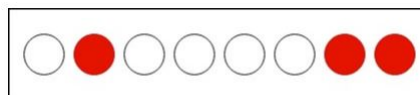
Řešení této úlohy bylo realizováno následovně. LED modul byl připojen k převodníku, jenž má piny vyvedeny do konektoru I2C_CON7. Volitelná část adresy je tvořena kombinací přivedených napěťových úrovní na vstupech čipu. V tomto případě je to hodnota 0x. Podle dokumentace je možné určit, že adresa bude mít hodnotu 0x40 pro zápis a 0x41 pro čtení. Nejprve byly inicializovány zařízení pro sériovou komunikaci pomocí funkcí *UART_Init* a *i2c_Init*. UART pro komunikaci s PC a I²C pro obsluhu převodníku, kód pro inicializaci jednotlivých periférií a příslušných pinů je v elektronické příloze práce. Dále byly vytvořeny funkce pro jejich obsluhu. Čtení dat z PC řeší funkce s názvem *UART_Read* s parametry funkce ukazatele na buffer a délky zprávy. Zápis do převodníku po sériové lince je realizován

funkcemi *i2c_write_start* pro začátek komunikace s periferií a *i2c_write_byte* pro odeslání dat. Dalším z požadavků je zavedení funkce, která bude řešit časové intervaly. To je řešeno systémovým 24-bitovým časovačem, který je implementován v jádru cortex-M0+. Po jeho inicializaci byly vytvořeny funkce *delay_us* a *delay_ms*. Kód algoritmu podle následujícího diagramu je v elektronické příloze práce.

Po spuštění kódu, aplikace čeká na zadání parametrů, které byly odeslány terminálu PC a poté aplikace začne vykonávat požadovanou funkci. Na obrázku 6.15 je vstup z terminálu PC, který je přijat mikrokontrolérem a výsledek běhu aplikace je znázorněn na obrázku. Výsledek pořízený vyfocením stavu LED modulu nebyl optimální a proto bylo zvoleno následující znázornění na obrázku 6.16, které zobrazuje binární hodnotu ASCII znaku "C".



Obr. 6.15: Zadané hodnoty z terminálu PC pro příklad č. 3

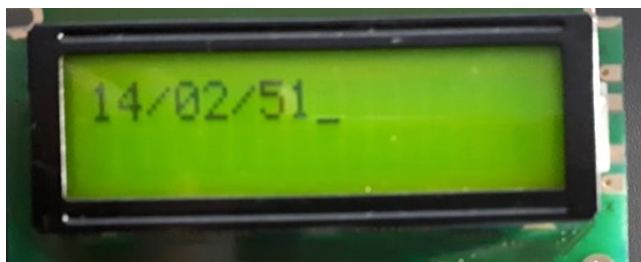


Obr. 6.16: Výsledek zobrazený na LED modulu pro příklad č. 3

6.2.4 Zobrazení aktuálního času z modulu RTC na LCD displeji

V tomto příkladu je realizováno zadání v následujícím znění. Využijte čipu RTC hodin s I²C rozhraním umístěným na procesorové desce a zobrazte aktuální čas na LCD displeji ve formátu hh/mm/ss. Čtete opakovaně hodnotu každou jednu sekundu.

Řešení zadání bylo provedeno následovně. Nejprve je za pomoci funkce *i2c_Init* provedena inicializace sériové linky I²C a dále pomocí funkce *UART_Init* je inicializován LCD displej. Z dokumentace RTC modulu je možné vyčíst, že aktuální hodnoty, které odpovídají sekundám, minutám a hodinám jsou uloženy na adrese 0x0, 0x1 a 0x2. Celkem je nutné přečíst 3 byty. Po úspěšném spojení se zařízením se vyše adresa a pomocí funkce *i2c_read()*. Odpovídající byty se vymaskují tak aby zůstala pouze požadovaná hodnota. Pro informaci o aktuálním čase, je použit formát BCD. Tyto hodnoty jsou následně upraveny do takového formátu, který je možné pomocí funkce *LCD_Print* zapsat na LCD displej. Před zapsáním nových dat vymaže obsah zobrazený na displeji. Celý cyklus je opakován každou sekundu. Na obrázku 6.17 je zaznamenán běh aplikace.



Obr. 6.17: Výsledek zobrazený na LED modulu pro příklad č. 4

6.2.5 Čtení pomocí ADC převodníku, ukládání hodnot do paměti EEPROM a odesílání zpráv do PC

V tomto příkladu je realizováno zadání v následujícím znění. Inicializujte ADC převodník pro čtení analogového signálu, jehož amplitudu je možné měnit pomocí potenciometru na procesorové desce. Čtení má probíhat každou sekundu a hodnota je odesílána po sériové lince do PC. Po převedení pěti hodnot se tento blok dat zapíše na začátek adresového prostoru EEPROM paměti s rozhraním I²C připojené na procesorové desce. Při úspěšném zápisu do paměti se odešle po sériové lince do PC zpráva "OK". inicializujte periferii komunikačního rozhraní UART s těmito parametry:

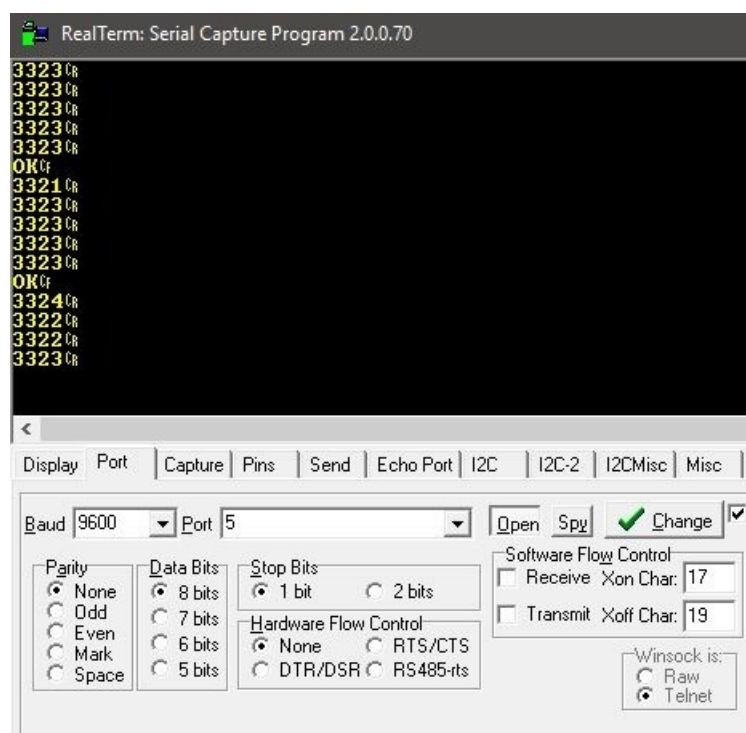
- Baudrate: 9600

- Paritní bit: Ne
- Stop bit: 1
- Datových bitů: 8

Převodník ADC má být inicializován s následujícími povinnými parametry:

- Rozlišení převodníku: 12 bitů
- Referenční napětí: Externí z analogového vstupu mikrokontroléru
- Spouštění měření z hlavní aplikace

Řešení zadání bylo provedeno následovně. Opět byly inicializovány periferie pro sériovou komunikaci. Ke komunikaci s PC je určen UART, který je inicializován funkcí *UART_Init*. Komunikaci s pamětí EEPROM probíhá přes I²C a ta je inicializována funkcí *i2c_Init*. Nakonec je nutné provést inicializaci převodníku ADC1 pro pin PB0. Na tomto pinu je připojen modul s napětím +5V a potenciometrem pro změnu amplitudy stejnosměrného signálu. Interval spuštění měření realizují funkce *delay_us* a *delay_ms*, jež jsou dostupné až po inicializaci funkcí *System_time_Init*. Hlavní aplikace využívá pro spuštění měření a čtení hodnoty z AD převodníku funkci *ADC_Read_Channel*. Jelikož je vrácená hodnota v rozlišení 12 bitů, tak je uložena do dvou bytů registru. Každá přečtená hodnota je poslána pomocí sériové linky do PC. K tomuto účelu aplikace používá funkci *UART_Puts*. Tato funkce umožňuje odeslání celého bloku dat. Následně aplikace čeká na další vykonání smyčky po uplynutí požadovaného času. Takto se vykoná měření a odeslání dat celkem pětkrát a přečtená data se ukládají do lokální proměnné. Po posledním čtení se celý blok zapíše pomocí rozhraní I²C do paměti EEPROM a při úspěšném zápisu odešle do PC zprávu "OK". V dalším kroku se opakuje celá sekvence znova. Výsledky běhu programu lze vidět na obrázku 6.18.



Obr. 6.18: Výsledek běhu programu příkladu č. 5 na PC

7 Závěr

První část této práce byla zaměřena na studii původního systému I51. Představila mikroprocesor AT89C51 a dále popisovala všechny část původního systému a jejich význam. V závěru kapitoly byly definovány potřebné parametry, které zahrnuje koncept systému nového. V druhé části byla provedena rešerše, která z definovaných požadavků postupně zužovala výběr vhodných mikrokontrolérů pro nový systém, až byl nakonec vybrán mikrokontrolér od firmy Microchip, konkrétně ATSAMC21N18A. Třetí část práce diskutovala vhodný koncept nového systému s vybraným mikrokontrolérem. Byl definován základní model systému, který zahrnul požadavek na kompatibilitu s původními koncovými periferiemi a nakonec byly vybrány další moduly, které rozšiřují možnosti práce s tímto systémem. Po definování modelu, bylo vytvořeno obvodové řešení a byla navržena DPS. Vyrobená deska byla osazena komponenty a úspěšně oživena. Pro systém byl v dalším kroku vybráno vývojové prostředí Atmel Studio a k němu programátor Atmel-ICE, který umožňuje nahrávání a ladění programu běžícího v mikrokontroléru. Závěrečná část popsala práci se všemi moduly systému a také s interními periferiemi mikrokontroléru. Nakonec bylo sestaveno několik slovních zadání příkladů a pro tyto příklady byly vypracovány a zdokumentovány vzorová řešení, které lze nalézt v elektronické příloze práce. Výsledkem práce je kompletní obvodové řešení systému, které splňuje všechny požadavky zadání a navíc zahrnuje i další rozšiřující moduly. Celý systém je možné programovat a ladit pomocí vývojového prostředí a zvoleného programátoru. Práce se systémem byla demonstrována vzorovými úlohami a systém samotný lze tedy využít pro účely výuky programování. Jedinou součástí, která nebyla dostatečně zdokumentována je modul SPI, který se nepodařilo zrealizovat v dostatečném předstihu a práce s tímto modulem bude dokončena po vydání této dokumentace. S výše uvedeným je možné označit tuto práci, jako kompletní řešení laboratorního mikrokontrolérového systému FEKTis.

Literatura

- [1] Alespoň něco o 8051. [online]. Michal Fuksa, 2009 [cit. 2019-05-12]. Dostupné z: http://1-2-8.net/mwva/fuksam/?fbclid=IwAR0CrN9Kg9VBZ-iSOOIE1FS__npWWMhypYjxngqTJVz_1-yeAEmJbVA77mY
- [2] Mikroprocesory s architekturou ARM [online]. Pavel Tišnovský, 2012 [cit. 2019-05-12]. Dostupné z: https://www.root.cz/clanky/mikroprocesory-s-architekturou-arm/?fbclid=IwAR1pnGbTQDoJ2DSiZPbd_Knk-KCs9yJVVEpG39G9-oM8Gqgbk6IhCWE8KAk
- [3] What is the Difference between 8051, PIC, AVR and ARM? [online]. Hyderabad: ElProCus., 2017 [cit. 2019-05-12]. Dostupné z: https://www.elprocus.com/difference-between-avr-arm-8051-and-pic-microcontroller/?fbclid=IwAR27ouHvDEIPI05AHK0N_7QCdFGGYqjpFc3xn1nBOE-o_mQkBTYK4MjewHY
- [4] Which ARM Cortex Core Is Right for Your Application: A, R or M? [online]. Silicon Labs [cit. 2019-05-12]. Dostupné z: <https://www.silabs.com/documents/public/white-papers/Which-ARM-Cortex-Core-Is-Right-for-Your-Application.pdf?fbclid=IwAR1cp0y93cYV8UmSIGz6TwtKVYMQUwgKFvcVRHa8fgh-yQWvtQbcR1kMy4E>
- [5] Cortex Family [online]. 2012 [cit. 2019-05-12]. Dostupné z: http://www.emcu.it/CortexFamily/CortexFamily.html?fbclid=IwAR3L5Rcsy0v5eni4g2h7Cmh4VA923MtlexrW-6FszmRVcMm_Hb8iWTIII1U
- [6] Five things you may not know about Cortex-R Series processors [online]. Arm Limited: Neil Werdmuller, 2014 [cit. 2019-05-12]. Dostupné z: <https://community.arm.com/developer/ip-products/processors/b/processors-ip-blog/posts/five-things-you-may-not-know-about-cortex-r-series-processors?fbclid=IwAR2Mpbx7pyWQ1VwDSt88XQ3UynTgn6XsO4DOpfghk6sr6SnFEO-hMGV4zsY>
- [7] Architektura mikrořadičů s jádry ARM Cortex-M0 a ARM Cortex-M0+ [online]. Pavel Tišnovský: Internet Info, 2015 [cit. 2019-05-12]. Dostupné z: <https://www.root.cz/clanky/architektura-mikroradicu-s-jadry-arm-cortex-m0-a-arm-cortex-m0/?fbclid=IwAR2FndG0NgYGa2Xv2SJAEGCp8IJbE3c7fytkiN3D5cAQMaFPccRCkaoHe9o>
- [8] USART u AVR 1. Díl – Teoretický popis [online]. Michal Dudka, 2016 [cit. 2019-05-12]. Dostupné z: <http://www.tajned.cz/2016/09/usart-u-avr-1-dil-teoreticky-popis/?fbclid=IwAR1qkQ3qPvA63ELwgN3hfvSG2IYACpZlof2VQexvBCaDI5yRNBYrlwJ4o4>

- [9] Stručný popis sběrnice I2C a její praktické využití k připojení externí eeprom 24LC256 k mikrokontroléru PIC16F877 [online]. Martin Olejář, 2000 [cit. 2019-05-12]. Dostupné z:
https://vyvoj.hw.cz/navrh-obvodu/strucny-popis-sbernice-i2c-a-jeji-prakticke-vyuziti-k-pripojeni-externi-eeeprom-24lc256?fbclid=IwAR1AO5v4lIDGmG3x14Y0hb-moy_LA001YG2ssHq0dcbXrh_PNzhY0hb3yys
- [10] Externí sériové sběrnice SPI a I2C [online]. Pavel Tišnovský, 2008 [cit. 2019-05-12]. Dostupné z:
https://www.root.cz/clanky/externi-seriove-sbernice-spi-a-i2c/?fbclid=IwAR2HclahFX0TfFo2JKPZ95GsMuV7ORvZ3EGt3rqnOz_MKFX_vu2Ap4n3qcs
- [11] SAM C20/C21 Family Data Sheet: 32-bit ARM Cortex-M0+ with 5V Support, CAN-FD, PTC, and Advanced Analog [online]. © 2019 Microchip Technology [cit. 2019-05-13]. Dostupné z:
http://ww1.microchip.com/downloads/en/DeviceDoc/SAM-C20C21-Family-Data-sheet-DS60001479C.pdf?fbclid=IwAR2J7Ku1oVY_3YpNHYBr_NcT54fvSl_VEERLb19BuF3DW6Qk3NnQkdiT8IY