

# Vstupně výstupní výukový modul pro platformu Arduino

Václav Mach

---

Bakalářská práce  
2014



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky  
akademický rok: 2013/2014

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Václav Mach**  
Osobní číslo: **A11037**  
Studijní program: **B3902 Inženýrská informatika**  
Studijní obor: **Bezpečnostní technologie, systémy a management**  
Forma studia: **prezenční**

Téma práce: **Vstupně výstupní výukový modul pro platformu Arduino**

Zásady pro vypracování:

1. **Prostudujte a popište mikropočítačovou platformu Arduino a několik vybraných příkladů rozšiřujících modulů pro tuto platformu.**
2. **Navrhněte rozšiřující modul se vstupně-výstupními periferiemi vhodnými pro výuku programování mikropočítačů.**
3. **Vytvořte prototyp navrženého modulu.**
4. **Vytvořte základní programové vybavení demonstrující programovou obsluhu modulu.**

Rozsah bakalářské práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. **BANZI, Massimo. Getting Started with Arduino. Second Edition. Sebastopol: OReilly Media, 2011, 128 s. ISBN 978-14-4930-987-9.**
2. **MANN, Burkhard. C pro mikrokontroléry: ANSI-C, kompilátory C, spojovací programy – linkery, práce s ATMEL AVR a MSC-51, příklady programování v jazyce C, nástroje pro programování, tipy a triky. Praha: BEN, 2003. ISBN 80-730-0077-6.**
3. **MARGOLIS, Michael. Arduino Cookbook. Second Edition. Sebastopol: OReilly Media, 2011, 724 s. ISBN 978-1-449-31387-6.**
4. **PINKER, Jiří. Mikroprocesory a mikropočítače. Praha: BEN – technická literatura, 2004. ISBN 80-730-0110-1.**
5. **ARDUINO. Arduino – HomePage [online]. 2011 [cit. 2014-01-08]. Dostupné z: <http://www.arduino.cc/>.**

Vedoucí bakalářské práce:

**Ing. Jan Dolinay, Ph.D.**

Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce:

**7. března 2014**

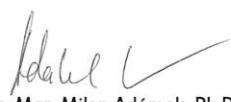
Termín odevzdání bakalářské práce:

**10. června 2014**

Ve Zlíně dne 7. března 2014

  
prof. Ing. Vladimír Vašek, CSc.  
*děkan*



  
doc. Mgr. Milan Adámek, Ph.D.  
*ředitel ústavu*

### **Prohlašuji, že**

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

### **Prohlašuji,**

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- Že odevzdaná verze diplomové/bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....  
podpis diplomanta

## **ABSTRAKT**

Cílem této práce je vytvořit rozšiřující modul (tzv. shield) pro platformu Arduino a umožnit jeho využití ve výuce programování mikropočítačů. Modul obsahuje různé vstupy a výstupy jako tlačítka, LED, potenciometr, apod. Náplní práce je také vytvoření návrhu zapojení modulu, jeho praktická realizace a naprogramování ukázkových programů.

Klíčová slova: Arduino, Výukový modul, Mikropočítač, EAGLE, Atmel

## **ABSTRACT**

The aim of this work is to create an expansion module (shield) for the Arduino platform to allow its use in teaching programming microcontrollers. The module contains of variety of inputs and outputs such as buttons, LEDs, potentiometer, etc. A design and module wiring, its practical implementation and programming of the sample programs was created.

Keywords: Arduino, Educational module, Microcontroller, EAGLE, Atmel

Děkuji vedoucímu mé bakalářské práce panu Ing. Janu Dolinayovi za jeho odbornou pomoc a vstřícné jednání během návrhu a vytváření této práce. Společnosti Semach bych chtěl touto formou poděkovat za realizaci navrženého modulu.

Prohlašuji, že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

*„Měřit postup programování podle počtu řádků zdrojového kódu je jako měřit stavbu hangáru jeho hmotností.“*

Bill Gates

# OBSAH

<b>ÚVOD</b> .....	<b>9</b>
<b>I TEORETICKÁ ČÁST</b> .....	<b>10</b>
<b>1 ARDUINO</b> .....	<b>11</b>
1.1 ZÁKLADNÍ DESKY .....	12
1.1.1 Arduino Mini.....	13
1.1.2 Arduino Uno.....	14
1.1.3 Arduino Tre .....	15
1.2 ROZŠÍŘUJÍCÍ SHIELDY .....	17
1.2.1 Ethernet shield.....	18
1.2.2 LCD Keypad shield .....	20
<b>2 MIKROPROCESOR ATMEL MEGA 2560</b> .....	<b>22</b>
2.1 ROZDĚLENÍ PAMĚTI .....	23
2.2 BLOKOVÉ SCHÉMA PROCESORU .....	24
2.3 PINY A PORTY .....	25
2.4 OVLÁDÁNÍ PORTŮ.....	26
<b>3 DESKA ARDUINO MEGA 2560</b> .....	<b>27</b>
3.1 OVLÁDÁNÍ PINŮ.....	28
3.2 KONSTRUKCE A ROZMĚRY .....	30
3.3 SERIAL PERIPHERAL INTERFACE .....	31
3.4 VÝVOJOVÉ PROSTŘEDÍ ARDUINA.....	32
<b>4 PROGRAM EAGLE</b> .....	<b>34</b>
4.1 OMEZENÍ LICENCE .....	35
<b>II PRAKTICKÁ ČÁST</b> .....	<b>36</b>
<b>5 NÁVRH MODULU</b> .....	<b>37</b>
5.1 ROZMÍSTĚNÍ SOUČÁSTEK NA MODULU .....	38
<b>6 REALIZACE MODULU</b> .....	<b>39</b>
6.1 VYTVOŘENÍ MASKY FOTOCESTOU.....	39
6.2 LEPTÁNÍ .....	39
6.3 OSAZOVÁNÍ SOUČÁSTEK A JEJICH PÁJENÍ.....	40

<b>7</b>	<b>POUŽITÉ PRVKY .....</b>	<b>41</b>
7.1	MATICOVÁ KLÁVESNICE .....	41
7.2	SEGMENTOVÝ DISPLEJ .....	43
7.3	LED MODULY .....	45
7.4	AUDIO ROZHRANÍ .....	46
7.5	TEPLOTNÍ ČIDLO .....	47
<b>8</b>	<b>UKÁZKOVÉ PROGRAMY .....</b>	<b>49</b>
8.1	SVĚTELNÝ HAD Z LED .....	49
8.2	STRÍDÁNÍ BAREV NA RGB MODULU .....	50
8.3	VOLTMETR .....	53
8.4	MELODIE .....	57
8.5	MĚŘENÍ TEPLoty .....	60
	<b>ZÁVĚR .....</b>	<b>62</b>
	<b>SEZNAM POUŽITÉ LITERATURY.....</b>	<b>63</b>
	<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK .....</b>	<b>66</b>
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>67</b>
	<b>SEZNAM TABULEK.....</b>	<b>69</b>
	<b>SEZNAM PŘÍLOH.....</b>	<b>70</b>

## ÚVOD

Mikropočítače nacházíme snad ve všech elektronických zařízeních, se kterými se denně setkáváme. Studenti technických oborů s nimi pracují na školách, kde se učí jejich využití a praktickému programování. Tato bakalářská práce navrhuje a prakticky realizuje rozšiřující modul pro podporu programování mikropočítačů, zejména předmětu Mikropočítače a PLC.

Platforma Arduino byla vybrána díky svým přednostem. První je fakt, že všechny programy a schémata jsou volně dostupné pod open-source licencí. Oproti konkurenčním platformám je Arduino jednou z nejlevnějších variant na českém, ale i zahraničním trhu. Struktura a syntaxe programovacího jazyka *wiring*, kterým jsou všechny výrobky programovatelné, jsou velmi podobné jazyku C. Tento jazyk je jedním z nejpoužívanějších programovacích jazyků na celém světě. Další významnou výhodou je celková kompaktnost výrobků této platformy a jejich snadné programování prostřednictvím USB rozhraní.

V teoretické části je stručně popsána platforma Arduino. Jsou zde představeny nejznámější základní desky, které firma vyrábí. Jsou zde uvedeny i některé rozšiřující desky, ze kterých by měl návrh modulu vycházet. V závěru teoretické části jsou popsány programy, ve kterých byl modul navržen a následně naprogramován.

V praktické části je nejprve proveden návrh samotného modulu, popsána jeho velikost, rozmístění součástí a způsob propojení modulu se základní deskou. Následuje popis jednotlivých fází výroby modulu. Jednotlivé prvky, které jsou na modulu použity, jsou přehledně popsány. Následuje několik základních programů, které mají demonstrovat možnosti a funkce sestaveného modulu.

Při návrhu tohoto modulu byl brán ohled na zkušenosti a znalost programovacího jazyka studentů, které mohli nashromáždit po dobu svého studia. Libovolnou kombinací použitých prvků lze velmi jednoduše nastavit míru obtížnosti zadaných cvičení.

## **I. TEORETICKÁ ČÁST**

## 1 ARDUINO

Arduino je jednou z celosvětově nejznámějších vývojových platforem. Firma byla založena v roce 2005 v Itálii. Původně se jednalo o podpůrný výukový set pouze pro studenty, ten se však nečekaně rychle uchytil a výrobci se rozhodli jej nabízet i veřejnosti. Produkty měly značný komerční úspěch a ze získaných financí se mohla společnost dále vyvíjet. [1,12]



Obrázek 1: Logo firmy Arduino [17]

Všechny hardwarové i softwarové podklady, podle kterých se jednotlivé desky vyrábí a programují, jsou volně k dispozici na stránkách výrobce. Jedná se tedy o čistý open-source projekt. Této možnosti využilo mnoho českých i zahraničních firem a vznikly tzv. Arduino klony. Ty používají stejné schéma zapojení, rozmístění součástek i rozteč pinů, aby byla zachována vzájemná kompatibilita mezi všemi zařízeními. Oblíbené jsou díky podstatně menší ceně za funkčně shodný produkt.

Celá platforma je charakteristická modrou barvou. Tuto barvu nalezneme u všech oficiálních produktů, které firma Arduino dodává na trh. Společnost Arduino dlouhodobě spolupracuje se společností Intel. Výsledkem této spolupráce by měla být základní deska, která jako první bude mít možnost samostatného grafického výstupu. O zpracování tak objemných dat se bude starat samostatný grafický čip od již zmíněné společnosti Intel. Deska ponese název Arduino Tre a její uvedení na trh se plánuje na rok 2014. Nejedná se však o první spolupráci. V roce 2013 vydala firma Arduino desku Intel Galileo ve spolupráci s Intelem. [1,12]

## 1.1 Základní desky

Většina desek je osazena 8-bitovými procesory od firmy Atmel. Jednotnost použitého čipu umožnila stejné programové vybavení pro všechny desky. Samotný program, který je volně ke stažení na stránkách výrobce, obsahuje knihovny, díky kterým je schopen komunikovat se všemi použitými procesory. U moderních počítačů probíhá tato komunikace pomocí USB rozhraní, které je pouhou simulací sériové linky RS-232. Softwarovou simulaci zpravidla zajišťuje sekundární procesor, který je součástí desky, není to však podmínkou.

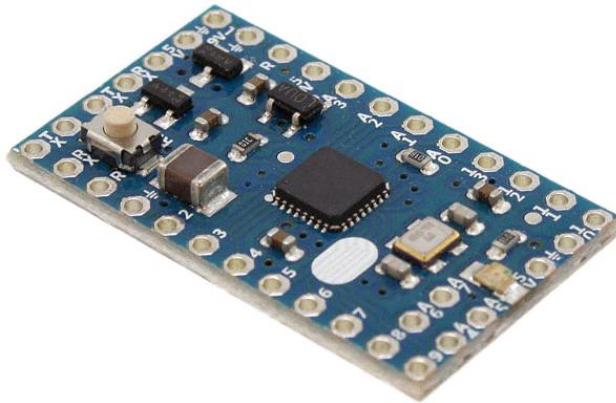
Aby se zabránilo postupnému stárnutí platformy, prochází navržené desky pravidelnými revizemi. Proces zahrnuje jen minimální obměnu stávajících komponentů, na samotnou funkci celku však nemá žádný vliv. Číslo provedené revize se uvádí na zadní stranu desky nebo shieldu ve formě potisku např. Rev3 jako třetí revize. Všechny změny a úpravy v rámci revize jsou popsány na stránkách výrobce. [12]

Název desky	Procesor	Vstupní napětí	Taktování procesoru	Analog In/Out	Digital In/Out	Flash paměť
Uno	ATmega328	7-12 V	16 MHz	6/0	14/6	32 KB
Due	AT91SAM3X8	7-12 V	84 MHz	12/2	54/12	512 KB
Tre	ATmega32u4	7-12 V	16 MHz	12/0	20/7	32 KB
Leonardo	ATmega32u4	7-12 V	16 MHz	12/0	20/7	32 KB
Mega 2560	ATmega2560	7-12 V	16 MHz	16/0	54/15	256 KB
Mega ADK	ATmega2560	7-12 V	16 MHz	16/0	54/15	256 KB
Micro	ATmega32u4	7-9 V	16 MHz	12/0	20/7	32 KB
Mini	ATmega328	7-9 V	16 MHz	8/0	14/6	32 KB
Nano	ATmega328	7-9 V	16 MHz	8/0	14/6	16 KB
Fio	ATmega328P	3,7-7 V	8 MHz	8/0	14/6	32 KB

Tabulka 1: Nejznámější desky Arduino a jejich základní parametry [1]

### 1.1.1 Arduino Mini

Co se plochy týče, jedná se o nejmenší model, který firma Arduino dodává na trh. Jádrem celé desky měl být původně procesor od Atmelu, konkrétně ATmega168. Z důvodu úspory místa byl později nahrazen procesorem ATmega328 od stejnojmenné firmy. [1]



Obrázek 2: Deska Arduino Mini [18]

V celkovém konceptu už nezbylo místo na rozhraní pro komunikaci pomocí standardního USB, které je u většiny desek samozřejmostí. Pro komunikaci prostřednictvím USB portu je uživatel nucen použít USB Serial Adapter. Vývody jednotlivých pinů jsou zakončeny průchodkou v desce, nikoliv standardními kolíky. Pro propojení desky s dalším hardwarem je zapotřebí připájení kolíku nebo vodiče do příslušné průchodky.

<b>Procesor</b>	ATmega328
<b>Taktovací frekvence</b>	16 MHz
<b>Vstupní napětí</b>	7-9 V
<b>Digitální I/O piny</b>	14 (6 použito pro PWM)
<b>Analogové vstupní piny</b>	8
<b>Flash paměť</b>	32 KB (2 KB použito pro Boot)
<b>SRAM</b>	2 KB
<b>EEPROM</b>	1 KB
<b>Max. zátěž na pin</b>	40 mA

Tabulka 2: Vlastnosti desky Arduino Mini [1]

### 1.1.2 Arduino Uno

V současné době je Arduino Uno nejznámější a nejpoužívanější deska. Tato skutečnost vychází z kompaktních rozměrů, dostatečného výpočetního výkonu a použitých prvků. Nachází se na ní dva procesory. Hlavní procesor ATmega328, který je taktován na frekvenci 16 MHz a umožňuje ovládat všechny prvky na desce. Druhý procesor ATmega16U2 umožňuje přímou komunikaci přes USB. Není zde zapotřebí žádný jiný převodník jako např. u Arduino Mini. [1,12]



Obrázek 3: Deska Arduino Uno [19]

U tohoto modelu jsou všechny ovladatelné prvky vyvedeny po obvodu celé desky ve formě kolíků. Vedle USB konektoru je umístěn i souosý napájecí konektor, který slouží jako primární napájení celé desky. Sekundárním zdrojem je zde využita možnost napájení přímo z USB konektoru.

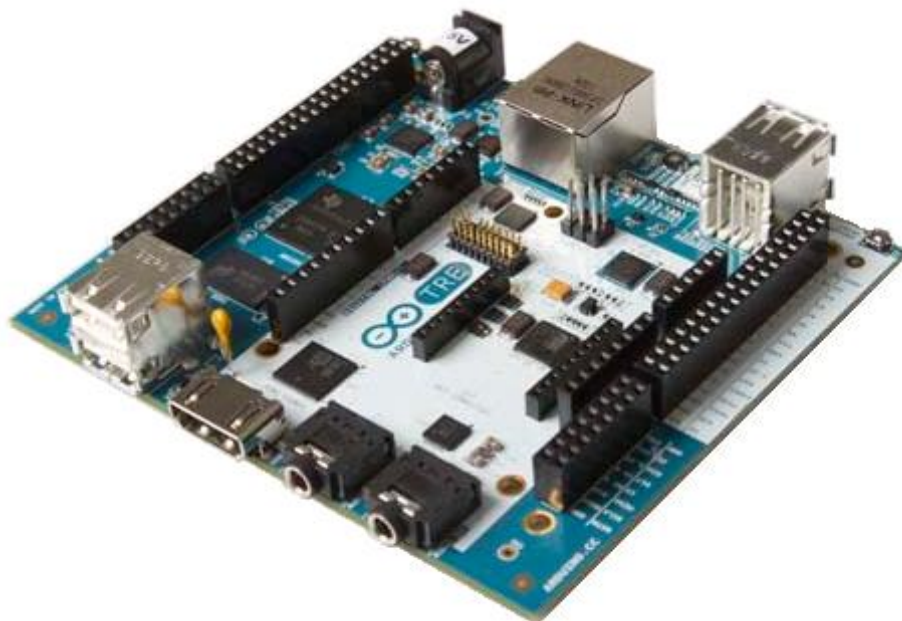
Deska dále disponuje dvanácti piny pro vstupně-výstupní digitální komunikaci a šesti piny pro komunikaci vstupně-analogovou. Samotný procesor je zde umístěn v provedení DIP. Tím pádem jej lze z patice, do které je zasunut, vyjmout bez potřeby pájecího stroje.

<b>Procesor</b>	ATmega328
<b>Taktovací frekvence</b>	16 MHz
<b>Vstupní napětí</b>	7-12 V
<b>Digitální I/O piny</b>	14 (6 použito pro PWM)
<b>Analogové vstupní piny</b>	6
<b>Flash paměť</b>	32 KB (0,5 KB použito pro Boot)
<b>SRAM</b>	2 KB
<b>EEPROM</b>	1 KB
<b>Max. zátěž na pin</b>	40 mA

Tabulka 3: Vlastnosti desky Arduino Uno [1]

### 1.1.3 Arduino Tre

Arduino Tre je zatím nejvýkonnější a nejnovější deska. Vznikla ve spolupráci se společností Intel za účelem podpory aplikací a programů běžících na jádře Linuxu. Jako první má grafický výstup HDMI, jehož zpracování zajišťuje procesor Sitara AM3359AZCZ100 od již zmíněné společnosti Intel. Ten je taktován na frekvenci 1 GHz a celková operační paměť RAM, která je procesoru přidělena, činí 512 MB.



Obrázek 4: Deska Arduino Tre [20]

Pro zpětnou kompatibilitu a funkci jádra Arduino nalezneme zde i procesor ATmega32U4. Jako u většiny desek i tady je možnost o rozšíření dalším modulem. Většina portů je rezervována pro komunikaci s Intel procesorem, ale pro ovládání čistě jádrem Arduino je vyvedeno do podoby kolíků pouze několik.

Nově byly použity USB huby pro připojení nebo ovládání zařízení, kterými mohou být klávesnice nebo myš. Pro práci se zvukem zde přibýly dva audio konektory, jeden vstupní a druhý výstupní. Celkový koncept míří k plně samostatnému grafickému prostředí, jehož výstupem je monitor.

<b>Procesor Arduino</b>	ATmega32U4
<b>Taktovací frekvence</b>	16 MHz
<b>Digitální I/O piny</b>	14 (7 použito pro PWM)
<b>Analogové vstupní piny</b>	6
<b>Flash paměť</b>	32 KB
<b>SRAM</b>	2,5 KB
<b>EEPROM</b>	1 KB

Tabulka 4: Vlastnosti procesoru Atmel desky Arduino Tre [1,12]

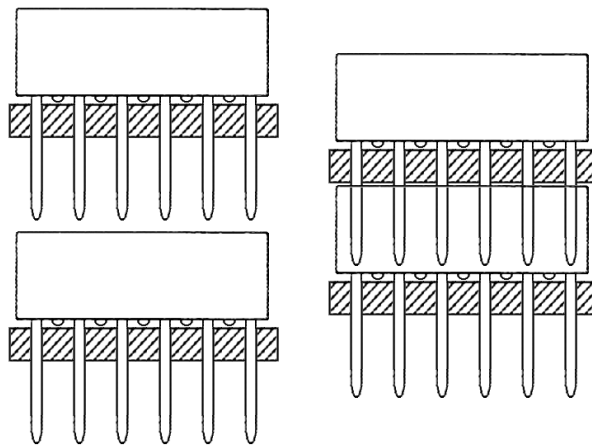
<b>Procesor Intel</b>	Sitara AM3359AZCZ100
<b>Taktovací frekvence</b>	1 GHz
<b>Digitální I/O piny</b>	12 (+ 7 PWM pinů)
<b>Analogové vstupní piny</b>	-
<b>Flash paměť</b>	-
<b>DRAM</b>	DDR3L 512 MB
<b>EEPROM</b>	-

Tabulka 5: Vlastnosti procesoru Intel desky Arduino Tre [1,12]

## 1.2 Rozšiřující shieldy

Pro maximální univerzálnost jsou na všech deskách umístěny pouze nejzákladnější komponenty, kterými jsou procesor a komunikační rozhraní. Pro možnost rozšíření např. o ethernetovou nebo bezdrátovou komunikaci dodává firma Arduino na trh řadu rozšiřujících desek, nazývaných *shieldy*.

Všechny větší desky, jako např. model Uno, Due nebo Mega, mají stejnou rozteč výstupních pinů. Také adresace a uspořádání jsou totožné, proto lze libovolný rozšiřující shield použít na libovolnou desku. Pro snadné připojení jsou použity speciální dutinkové kolíky s prodlouženými piny. Tento způsob umožňuje umístit i více rozšiřujících shieldů na sebe.



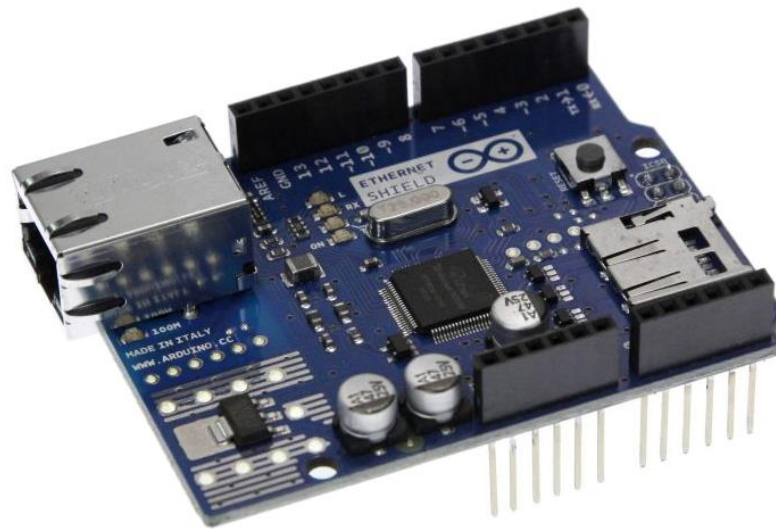
Obrázek 5: Systém napojení více shieldů na sebe

Firma Arduino dodává k hotovým shieldům i volně dostupné programy pro jejich obsluhu. Komunikace mezi jednotlivými mikročipy probíhá většinou přes SPI. Všechny ostatní piny jsou pro komunikaci nevyužity a tvoří cestu pro připojení dalších potřebných periférií uživatele. [1]

Přídavné shieldy nepotřebují ke své činnosti žádné vnější napájení. To je zajištěno skupinou pinů, které prostupují skrze propojovací kolíky. Tyto piny jsou označeny jako Power. Při propojení více shieldů je zapotřebí dbát maximálního výstupního výkonu desky. Opomenutím této skutečnosti může dojít k poškození samotné desky.

### 1.2.1 Ethernet shield

Pro připojení desky do lokální sítě lze použít ethernetový shield. Veškerou komunikaci a přístup do sítě poskytuje čip W5100 od firmy WIZnet. Čip je plně kompatibilní se standardy IEEE 802.3i (10 BASE-T) a IEEE 802.3u (100 BASE-TX). Z toho vyplývá, že komunikace v síti je možná pouze po metalickém vedení. Pro fyzické připojení slouží konektor RJ-45, který je pevně realizován na okraji shieldu. [1,13]



Obrázek 6: Ethernet shield [21]

Fyzické propojení mezi čipem W5100 a konektorem RJ-45 je realizováno sběrníci, která je popsána v tabulce č. 6. Jedná se o dvojici párů, na kterých jsou fyzicky odesílána a přijímána data mezi shieldem a routerem.

Název	Pin	Popis
RXIP	5	Přijímání dat
RXIN	6	
TXOP	8	Odesílání dat
TXON	9	

Tabulka 6: Adresace pinů mezi W5100 a RJ-45 [13]

Pro zpětnou verifikaci jsou na desce umístěny LED diody, informující uživatele o aktuálním stavu systému. Činnost těchto LED diod nelze uživatelsky ovládat, informují pouze o stavu, rychlosti připojení a poruchách. Funkce je vysvětlena v následující tabulce.

Název	Pin	Funkce
TX LED	73	Odesílání dat
RX LED	72	Příjem dat
SPD LED	67	Rychlost připojení
LINK LED	66	Přenos dat

Tabulka 7: Funkce signalizačních diod na shieldu [13]

Dále se na shieldu nachází i slot na klasické microSD karty. Toto zařízení lze ovládat přes SPI pouze pomocí připojené desky, nikoliv již zmíněným čipem W5100. Po vložení karty lze modul použít např. jako externí paměť. Na stránkách výrobce Arduino je k dispozici volně dostupná knihovna pro snadnou komunikaci mezi kartou a deskou.



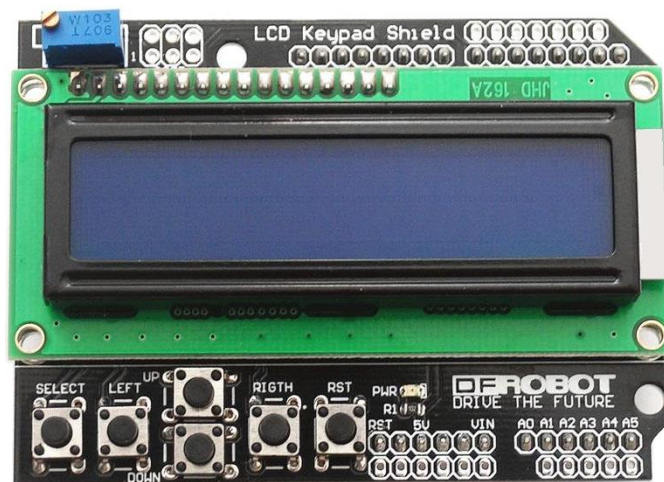
Obrázek 7: Adresace pinů mezi SD kartou a deskou Arduino [23]

K samotnému shieldu lze dodatečně připojit PoE modul. S Ethernet shieldem je kompatibilní řada AG9000 od firmy Silvertel. Ta zpracovává externí napájení, které je přivedeno pomocí ethernetového kabelu. Zařízení dále splňuje normu IEEE 802.3af. Modul AG9000 je nutno na shield připájet, nejsou zde k dispozici žádné dutinkové lišty. Výstup z modulu je přiveden na pin Vin, který sleduje aktuální napětí. Pokud vstupní napětí dosáhne požadované hodnoty, celé zařízení se obejde bez primárního napájení ze sítě. [14]

Jelikož se na shieldu nachází více periférií, ovládaných pomocí SPI, musí mít každá svůj jedinečný identifikátor. U Arduina se identifikace děje pomocí aktivace SS příslušného periférie. V případě ethernetového shieldu se se jedná o pin 10 pro adresaci čipu W5100 a pinu 4 pro volbu slotu pro SD kartu. Tato kombinace platí pouze pro desky Arduino Uno a Mega2560. [1]

### 1.2.2 LCD Keypad shield

Velmi rozšířený shield, díky kterému lze zobrazovat text na alfanumerickém dvouřádkovém displeji. Nejedná se o modul, který navrhla a vyrábí přímo firma Arduino. Na trhu je mnoho výrobců, kteří tento komponent vyrábí. Shieldy jsou k dispozici v několika variantách, které se liší pouze použitým displejem. Všechny tyto displeje řady 1602 s řadičem, používají stejnou adresaci i rozložení pinů. Rozdíly jsou pouze v použitém podsvícení či barvě zobrazovaného textu.



Obrázek 8: LCD Keypad shield [22]

Samotný řadič displeje HD44780 od firmy Hitachi je zhotoven pro komunikaci pomocí osmi datových a dvou řídicích vodičů. Při nedostatku pinů na řídicím procesoru je zde možnost využít komunikaci pouze po čtyřech datových vodičích. V konečném důsledku se ušetří vodičů na úkor rychlosti, která klesne přesně na polovinu, protože komunikace již nadále probíhá sériově. Tento způsob komunikace je použit i u popisovaného LCD Keypad shieldu. [15]

Samotný přepis znaku na kód, kterým se zobrazí na displeji, zpracovává některá z open-source knihoven. Ta konvertuje ASCII znak na osmi bitovou informaci, která se dále přeposílá na řadič displeje. Před samotným zobrazením znaku na displeji je zapotřebí nastavit řádek a sloupec, na kterém se má znak zobrazit. Všechny tyto funkce jsou již implementovány ve zmíněné knihovně, která je volně dostupná ke stažení.

<b>Pin</b>	<b>Funkce</b>
A0	Adresace tlačítek
D4	DB4
D5	DB5
D6	DB6
D7	DB7
D8	RS
D9	Enable
D10	Intenzita podsvícení

Tabulka 8: Adresace vývodu na desku Arduino [16]

Pro větší univerzálnost jsou na shieldu umístěna také ovládací tlačítka. S ohledem na nedostatek pinů na řídicím procesoru je adresace všech tlačítek zpracována na jednom analogovém vstupu. Jedná se o velice ojedinělý způsob zapojení, který je realizovatelný pouze na procesorech s analogovými vstupními piny. [15]

Adresace jednotlivých tlačítek spočívá v sériovém řazení odporů. Při stisku tlačítek se mění počet odporů zapojených do smyčky a tím i výsledná hodnota napětí, která je přivedena na analogový vstup procesoru. Dále je spojitá informace převedena na digitální pomocí A/D převodníků. Ze zpracované hodnoty lze pak jednoznačně určit, které tlačítko bylo stisknuto. Samotný příjem a převod informace je také součástí knihovny.

Podsvícení displeje je uživatelsky nastavitelné. V případě možnosti využití PWM výstupů je regulace intenzity podsvícení přivedena na PWM výstup procesoru. Adresace všech pinů je uvedena níže v Tabulce č. 8.

## 2 MIKROPROCESOR ATMEL MEGA 2560

Osmi-bitové mikroprocesory od firmy Atmel využívají jak výhody RISC, tak i CISC architektury. Čipy používají jen okolo 120 instrukcí, tím se přibližují spíše RISC architektuře. Na druhou stranu stejná délka bitového slova a zpracování celé instrukce v jednom hodinovém cyklu připomíná CISC. Vzájemné sladění těchto architektur má za následek rychlost a snadnou implementaci programovacího jazyka C, který se ve velké míře používá při programování mikročipů, jakým je i ATmega2560. [3]



Obrázek 9: Mikroprocesor ATmega2560 [24]

Při výrobě toho čipu se také kladl značný důraz na jeho celkovou spotřebu elektrické energie. Mnohé procesory při vykonávání jednodušších programů čekají např. na vnější přerušení, které vykoná změnu, nebo provedení instrukce a následně zase upadá do výpočetní nečinnosti. Proti zbytečnému plýtvání disponují procesory ATmega řadou úsporných režimů. Tyto režimy zachovají pouze obsah všech registrů v RAM a obvody vnějšího přerušení pro navrácení k plně funkčnímu stavu, ostatní výpočetní aktivita procesoru utichne. [11]

<b>Čip</b>	8-bit AVR
<b>Flash paměť</b>	256 KB
<b>Taktovací frekvence</b>	16 MHz
<b>Celkový počet pinů</b>	100
<b>Počet I/O pinů</b>	86
<b>Externí přerušení</b>	32
<b>Počet SPI</b>	5

Tabulka 9: Vlastnosti procesoru Atmel Mega 2560 [6]

## 2.1 Rozdělení paměti

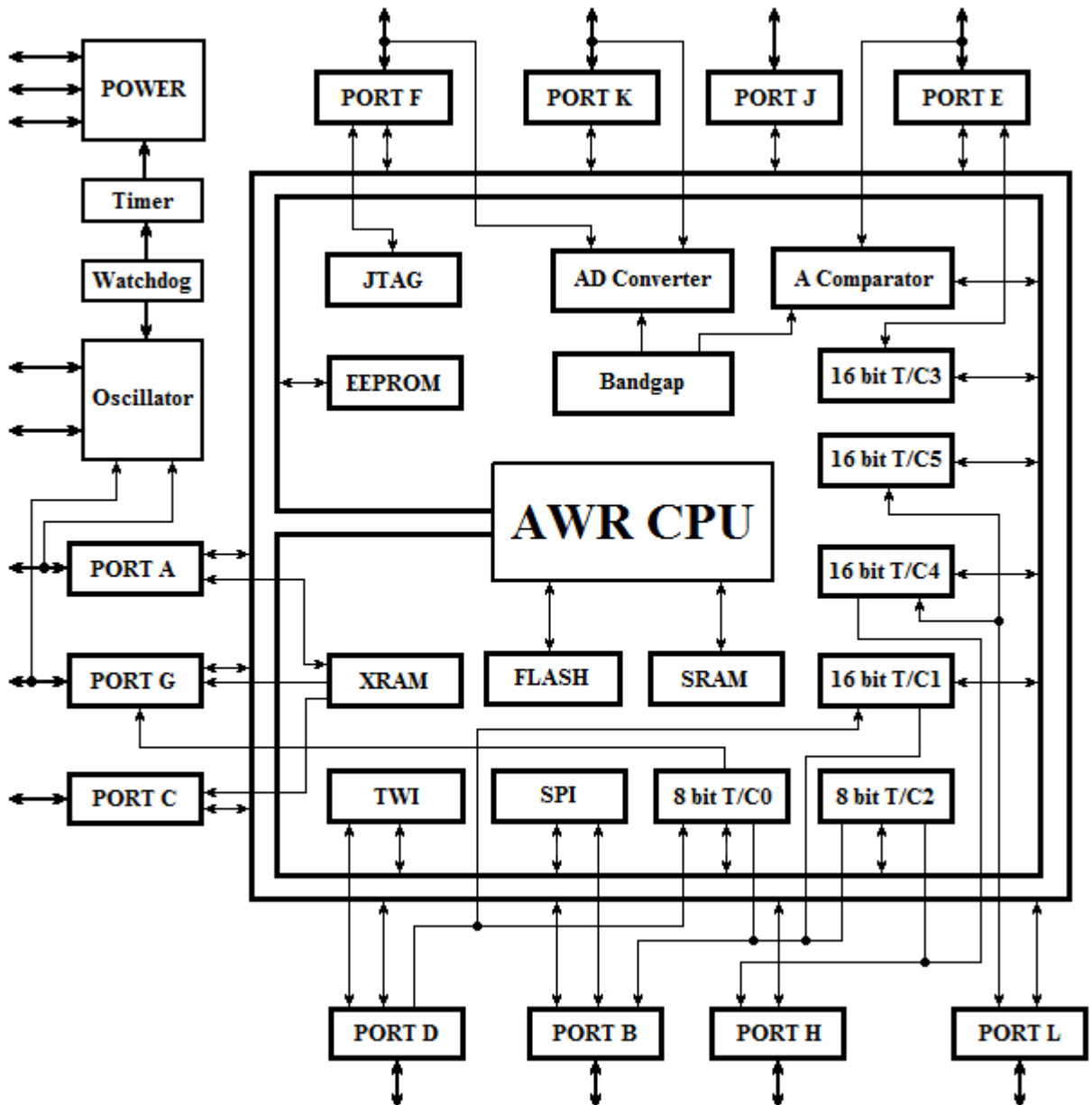
Všechny mikroprocesory řady ATmega používají Harvardskou architekturu. Ta se vyznačuje oddělenými sběrnicemi zvlášť pro data a zvlášť adresy. Dále taky využívají stejné rozčlenění paměťového prostoru, které je znázorněno na následujícím obrázku.

<b>32 Registry</b>	0x0000 0x001F	
<b>64 I/O Registry</b>	0x0020 0x005F	<b>0,5KB</b>
<b>416 Externí I/O Registry</b>	0x0060 0x01FF	
<b>8 192 Interní SRAM</b>	0x0200 0x21FF	<b>8KB</b>
<b>65 832 Externí SRAM</b>	0x2200 0xFFFF	<b>55,5KB</b>

Obrázek 10: Rozdělení paměti procesoru ATmega2560 [6]

## 2.2 Blokové schéma procesoru

Všechny komponenty jsou mezi sebou propojeny pomocí osmi bitové sběrnice, která znázorňuje jak adresní, tak i datovou komunikaci. Z blokového schématu lze snadno vyčíst, na kterých portech je možnost využívání speciálních služeb, kterými jsou např. A/D převodník, nebo časovače a čítače.



Obrázek 11: Blokové schéma mikroprocesoru ATmega2560 [6]

### 2.3 Piny a porty

Z pouzdra mikroprocesoru Atmega2560 je vyvedeno celkem 100 pinů. 86 vyvedených pinů zabírají ovladatelné piny. Zbytek tvoří napájení, reset, externí připojení oscilátoru a podpůrné větve pro převodníky.

**VCC:** Vstupní napětí na procesoru, které dodává elektrickou energii všem jeho součástem. Na vstupní napětí jsou kladeny velmi přísné požadavky na filtraci. Aby byly dodrženy maximální přípustné hodnoty zatížení na aktuální větve, je napájecí napětí do procesoru přivedeno hned několika piny.

**GND:** Společná zem všech připojených komponentů.

**RESET:** Fyzický reset na procesoru je obvykle vyveden ve formě tlačítka. Při jeho aktivaci se ihned přerušuje vykonávaný začátek. Reset je proveden jen tehdy, pokud se na vstupním pinu objeví úroveň logické nuly déle, než je minimální délka hodinového signálu.

**XTAL:** Každý procesor obsahuje vnitřní generátor hodinového signálu, pomocí těchto pinů lze připojit i externí. Nejznámější součástkou, kterou lze použít, je krystal. Každý nese přesnou hodnotu frekvence, kterou nelze měnit.

**AVCC:** Externí napájení pro A/D převodník. Defaultně je tento pin připojen na VCC přes dolnoproustní filtr, který je realizován indukci a kondenzátorem. Bez zapojení tohoto pinu by nebylo možné snímat a převádět analogový vstupní signál. [10]

**AREF:** Referenční hodnota A/D převodníku. Zpravidla se zapojuje přes kondenzátor na GND. Velikost přivedeného napětí slouží jako referenční úroveň pro porovnání zpracovávaného signálu. [10]

## 2.4 Ovládání portů

Pro práci s piny mají procesory řady ATmega vymezenou oblast v paměti, která se nazývá I/O Registry. Zde se ukládají všechny stavy a změny daného portu. Každému portu přísluší v paměti trojice po sobě jdoucích registrů, z nichž každý má statickou adresu. Trojice uchovává fyzickou adresu daného portu, směr provozu a jeho hodnotu. [6]

		<b>Pin Address</b>							
		7	6	5	4	3	2	1	0
<b>0x0020</b>		PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0
<b>Read</b>		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>Write</b>		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

		<b>Data Direction Register</b>							
		7	6	5	4	3	2	1	0
<b>0x0021</b>		DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0
<b>Read</b>		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>Write</b>		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

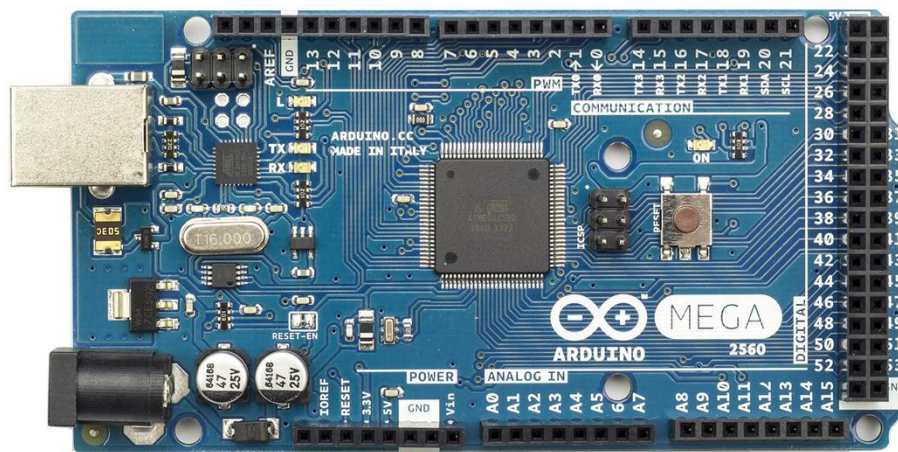
		<b>Data Register</b>							
		7	6	5	4	3	2	1	0
<b>0x0022</b>		PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0
<b>Read</b>		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>Write</b>		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Obrázek 12: Ovládání portu A [6]

Hodnoty těchto registrů lze nezávisle na sobě měnit. Samotný zápis je možný pouze celým přepisem registru. Pro změnu jednoho bitu je zapotřebí přepsat všech osm bitů. Jediné nastavení, které je nutné vykonat před spuštěním programu, je nastavení směru komunikace pinů registru. To se děje při tzv. inicializaci.

### 3 DESKA ARDUINO MEGA 2560

Arduino Mega 2560 dostalo své jméno po použitém procesoru ATmega2560 od firmy Atmel. Ten je taktován na frekvenci 16 MHz. Celá deska se vyznačuje velkým počtem ovladatelných pinů, celkem jich je 54. Jako u většiny větších desek, ani zde nechybí USB konektor pro připojení k počítači. Komunikaci, která probíhá prostřednictvím USB, obstarává sekundární procesor ATmega16U2.



Obrázek 13: Arduino Mega 2560 [26]

Napájení desky elektrickým proudem je možné hned dvěma způsoby. Na desce se nachází konektor, kterým lze přes vhodný adaptér napájení přímo ze sítě. Napětí by se mělo pohybovat v rozmezí 7-12 V. To se dále zpracovává stabilizátory, které jej sniží a vyhladí na 5 V. Další možnost napájení je přímo z USB, které je připojeno k počítači. Nestandardní, zato velmi používanou variantou, je i použití 9 V alkalické baterie.

<b>Procesor</b>	ATmega2560
<b>Taktovací frekvence</b>	16 MHz
<b>Vstupní napětí</b>	7-12 V
<b>Digitální I/O piny</b>	54 (15 použito pro PWM)
<b>Analogové vstupní piny</b>	16
<b>Flash paměť</b>	256 KB (8 KB použito pro Boot)
<b>SRAM</b>	8 KB
<b>EEPROM</b>	4 KB
<b>Max. zátěž na pin</b>	40 mA

Tabulka 10: Vlastnosti desky Arduino Mega 2560 [1]

### 3.1 Ovládání pinů

Jednotlivé piny jsou vyvedeny v podobě dutinkových lišt. Ty jsou rozděleny většinou po šesti nebo osmi pinech. Číslování těchto pinů nekoresponduje se zapojením jednotlivých portů procesoru ATmega2560. Piny jsou umístěny po okrajích desky ve dvou řádcích a dvou sloupcích. U každé skupiny nalezneme popis daných pinů a u každého pinu je uvedeno číslo, pod kterým je prezentován v programovacím prostředí.

**Digital Pins:** Všech 54 pinů lze použít jako vstupní, nebo výstupní. Při jejich výstupní aktivaci se na jejich vývodech objeví napětí 5 V a jejich maximální výstupní výkon je omezen na 40 mA. Pro jejich ovládání je zapotřebí použít funkce, které jsou uvedeny a popsány v následujícím zdrojovém kódu.

```
void setup()
{
  pinMode(ledPin, OUTPUT); // nastavení pinu jako vystup
  pinMode(buttonPin, INPUT); // nastavení pinu jako vstup
}

void loop()
{
  digitalWrite(ledPin, HIGH); // zapis na pin
  digitalRead(ButtonPin); // cteni z pinu
}
```

Obrázek 14: Ovládání digitálních pinů [2]

**PWM Pins:** Celkem 10 pinů disponuje funkcí pulzně šířkové modulace. Díky této funkci lze na výstupech těchto pinů vysílat spojitou hladinu napětí od 0 V až do 5 V ve 255 krocích. Tyto piny lze použít i jako digitální piny, ale pouze jako výstup.

```
void setup()
{
  pinMode(ledPin, OUTPUT); // nastavení pinu jako vystup
}

void loop()
{
  analogWrite(ledPin, 64); // zapis hladiny 64 na vystup (1,25V)
}
```

Obrázek 15: Ovládání výstupních pinů pomocí PWM [2]

**Interrupts:** Nastavení přerušení se děje při inicializaci. Do syntaxe se zadává číslo přerušení, název funkce, kterou program vykoná a druh změny. Činnost obsluhy přerušení se spustí ve chvíli, kdy se daný pin spojí se zemí. Po vykonání určeného podprogramu se vrátí k naposledy zpracovávané instrukci.

```
void setup()
{
  pinMode(pin, OUTPUT);           // nastavení pinu jako výstup
  attachInterrupt(0, blink, CHANGE); // nastavení přerušeni int.0
}

void loop()
{
  digitalWrite(pin, state);
}

void blink()                       // vykonana funkce přerusenim
{
  state = !state;
}
```

Obrázek 16: Ovládání pinů přerušení [4]

**Analog Pins:** Na desce Mega 2560 se nachází celkem 16 pinů, kterými lze snímat vstupní hodnoty napětí. Maximální úroveň napětí, kterou lze na pin přivádět, je 5 V. Při jejím překročení může dojít až k poškození samotného procesoru. Jelikož jsou piny velmi jednostranné, při jejich inicializaci odpadá nastavení směru komunikace. Ta je vždy vstupní.

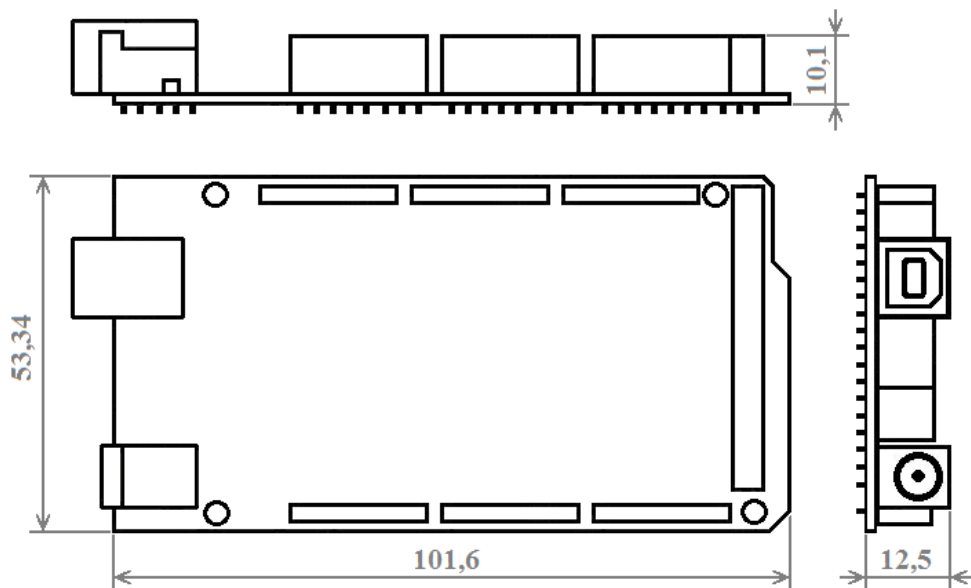
```
void setup()
{
}

void loop()
{
  val = analogRead(analogPin); // sejmuti hodnoty z pinu
}
```

Obrázek 17: Ovládání vstupních analogových pinů [2]

### 3.2 Konstrukce a rozměry

Deska Arduino Mega 2560 patří mezi největší, které firma vyrábí. Je tvořena oboustrannou deskou plošných spojů. Všechny součástky jsou v provedení SMD, to zaručuje značnou úsporu místa na desce. V desce jsou dále vyhotoveny i montážní otvory, kterými lze desku přichytit do případné krabičky nebo na konstrukci. Slouží také k trvalému přichycení přídatných shieldů.

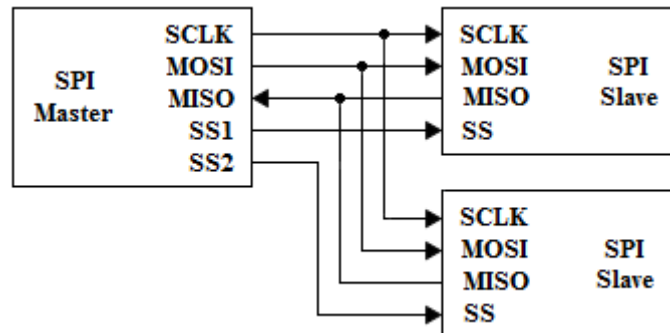


Obrázek 18: Rozměry desky Arduino Mega 2560 [8]

Nejnovější verze Rev3. Ta přinesla rozšíření desky o dva další piny, které slouží k podpoření komunikace mezi ostatními komponenty. Pro vyšší rychlost byl sekundární procesor ATmega8U2 pro zprostředkování USB komunikace nahrazen výkonnějším procesorem ATmega16U2. [1]

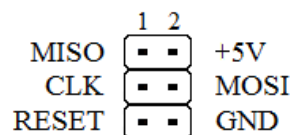
### 3.3 Serial periperal interface

Tento standard slouží jako způsob komunikace mezi jednotlivými obvody na jednom plošném spoji. Na všech větších deskách nalezneme toto SPI rozhraní, které je většinou realizováno jako konektor. Ten má přesné umístění a rozložení pinů, aby bylo zajištěno spolehlivé propojení. Adresace jednotlivých pinů se u některých modelů liší. [5]



Obrázek 19: Princip komunikace pomocí SPI [25]

Díky dostatku místa se na desce Arduino Mega 2560 nachází toto rozhraní ve formě konektoru. Komunikace probíhá na pinech: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). Pro bezproblémový provoz lze na stránkách Arduina bezplatně stáhnout knihovnu, která podporuje komunikaci mezi ostatními zařízeními pomocí SPI. [1]

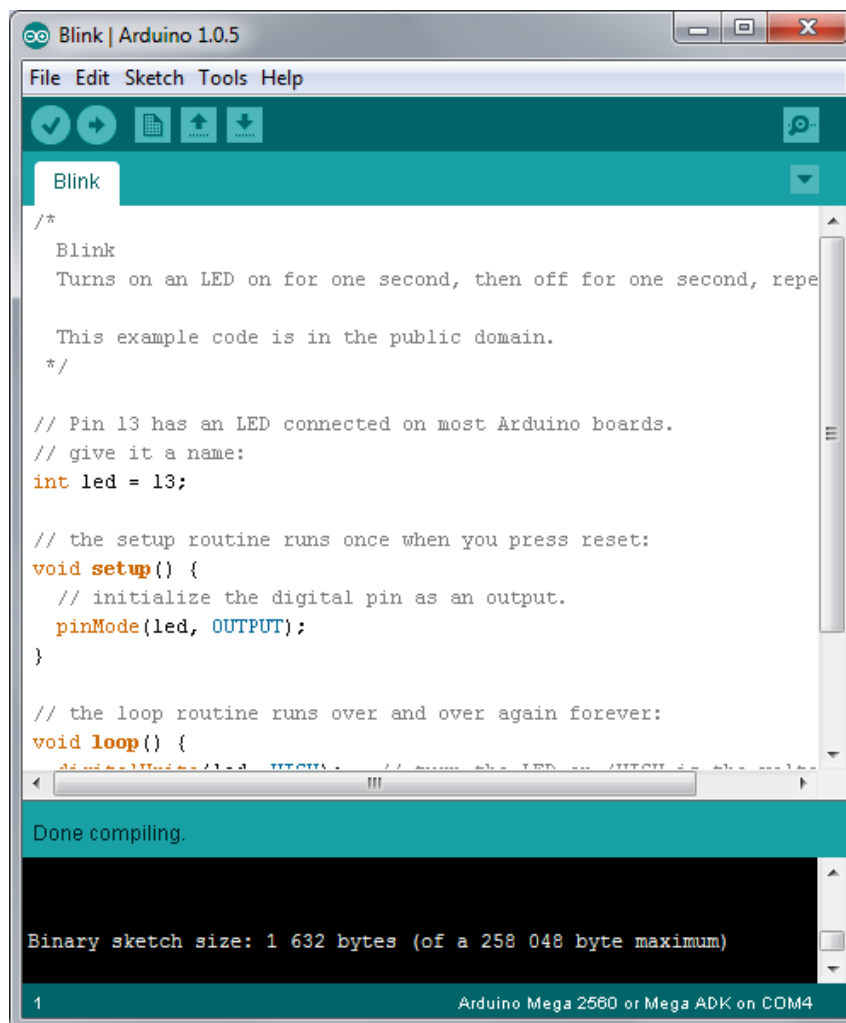


Obrázek 20: Funkce a rozložení konektoru SPI

Piny, použité pro tuto komunikaci, lze však využít i pro normální komunikaci, v programovém prostředí se nijak neoddělují. Je zde také nutno podotknout, že čísla pinů, použitých pro SPI komunikaci, se u jiných desek liší.

### 3.4 Vývojové prostředí Arduina

Pro programování procesorů používá firma Arduino vlastní grafické programovací prostředí. Program vychází ze dvou programovacích jazyků, kterými jsou *processing* a *wiring*. Díky *processingu* je možné vytvořit zpětnou vizualizaci dat a *wiring* obstarává veškerou komunikaci s programovatelným procesorem. Výsledný programovací jazyk je velmi podobný jazyku C a platí pro něj stejná pravidla. [1]



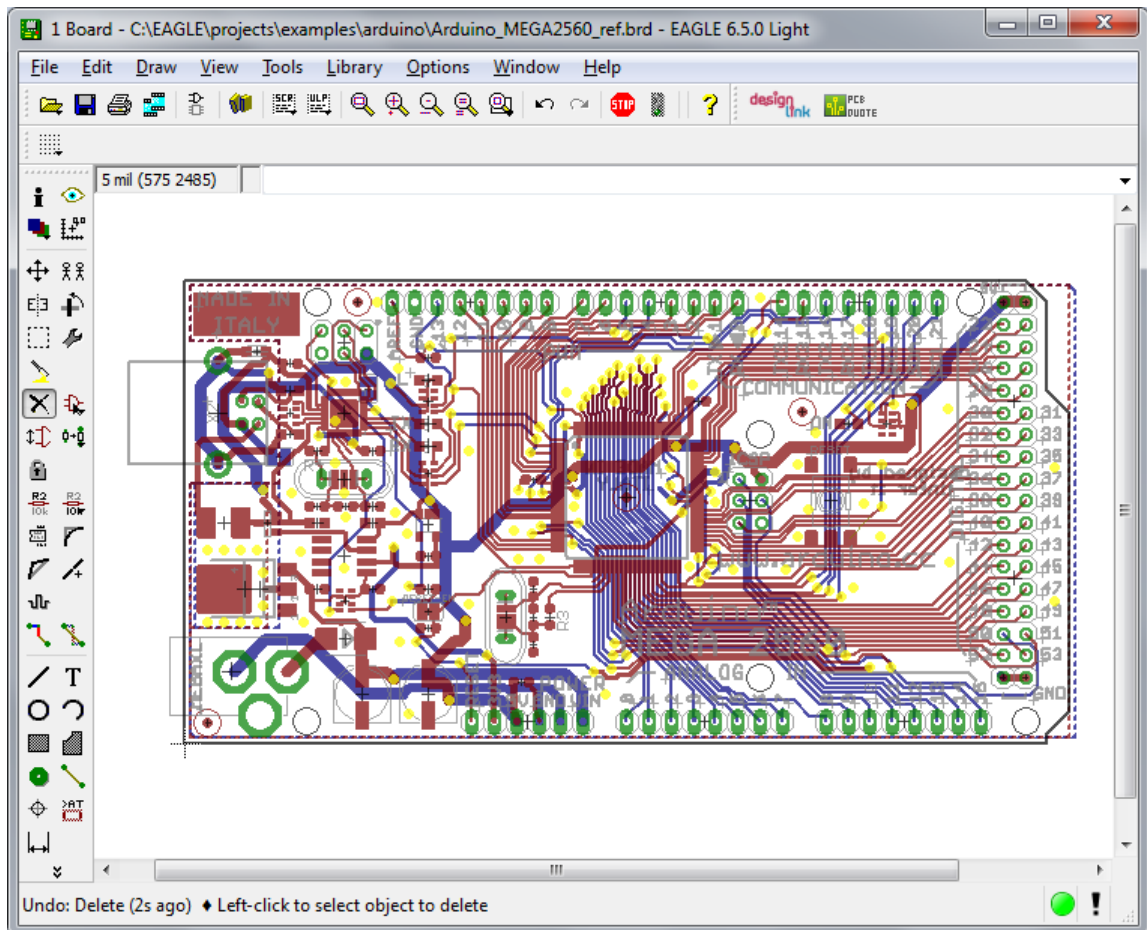
Obrázek 21: Vývojové prostředí Arduino 1.0.5

Pro všechny desky je dostupné pouze jedno programovací prostředí. Program obsahuje všechny knihovny, které jsou zapotřebí pro komunikaci mezi jednotlivými deskami. Pro bezproblémové propojení stačí zadat verzi aktuálně připojené desky a číslo sériového portu, na kterém je zařízení připojeno. Na všech modernějších zařízeních jsou sériové porty vyvedeny jako USB.

Program barevně odlišuje argumenty syntaxe a zpřehledňuje celkový program. Před samotným nahráním programu do procesoru lze program zkompilovat a tím zkontrolovat případné syntaktické chyby. Tlačítkem Upload se aktuální program uloží do procesoru. Procesor ATmega2560 disponuje celkem 256 volnými kilobajty, které lze do paměti uložit. Stav zaplnění připojeného modelu se zobrazuje ve spodní části okna programu.

## 4 PROGRAM EAGLE

Jedná se o grafický program, kterým lze navrhovat a vytvářet plošné spoje. Úplný název programu zní: **Easily Applicable Graphical Layout Editor**. Aplikace podporuje jak jednovrstvé, tak i vícevrstvé desky. Je však nutno mít na paměti, že program se stará pouze o schéma zadané uživatelem. Dále již neověřuje logiku a správnost zapojení.



Obrázek 22: Vývojové prostředí EAGLE 6.5.0

Program je rozdělen na dvě samostatné, ale vzájemně provázané části. První slouží pouze pro tvorbu schémat. V ní si uživatel sestaví celé schéma pomocí již přednastavených značek. Ve druhé dostanou součástky fyzickou podobu v editoru, které odpovídá skutečným součástkám. Zde také lze s komponenty libovolně pohybovat a otáčet. Ve finální části dojde k propojení jednotlivých vývodů jednotlivých součástek podle předem vytvořeného schématu.

**Auto-Routing:** Při větší četnosti součástek na plošném spoji je bezkolizní propojení jednotlivých vývodů součástek pro člověka takřka nemožné. Tuto záležitost lze obejít využitím zmíněné funkce auto-routing. Podle schématu dokáže program najít optimální umístění cest.

#### 4.1 Omezení licence

Pro běžné neziskové použití je tato licence zcela zdarma. Obsahuje ovšem určitá omezení. Plocha, kterou lze v editoru využít, je omezena na 10 x 8 cm. Dále jsou podporované pouze dvě vrstvy, spodní a vrchní. *Freeware* verze je volně dostupná ke stažení na stránkách výrobce.

## **II. PRAKTICKÁ ČÁST**

## 5 NÁVRH MODULU

Součástky, které budou na modulu použity, byly voleny s ohledem na to, že modul bude sloužit jako pomůcka při výuce programování mikropočítačů a PLC. Složitost ovládání je cílena na studenty se středně pokročilými znalostmi programování v jazyce C. Použité prvky nejsou na sobě nikterak závislé, tudíž je lze používat samostatně. Pro větší složitost je lze navzájem kombinovat k dosažení složitějších zadání pro studenty.

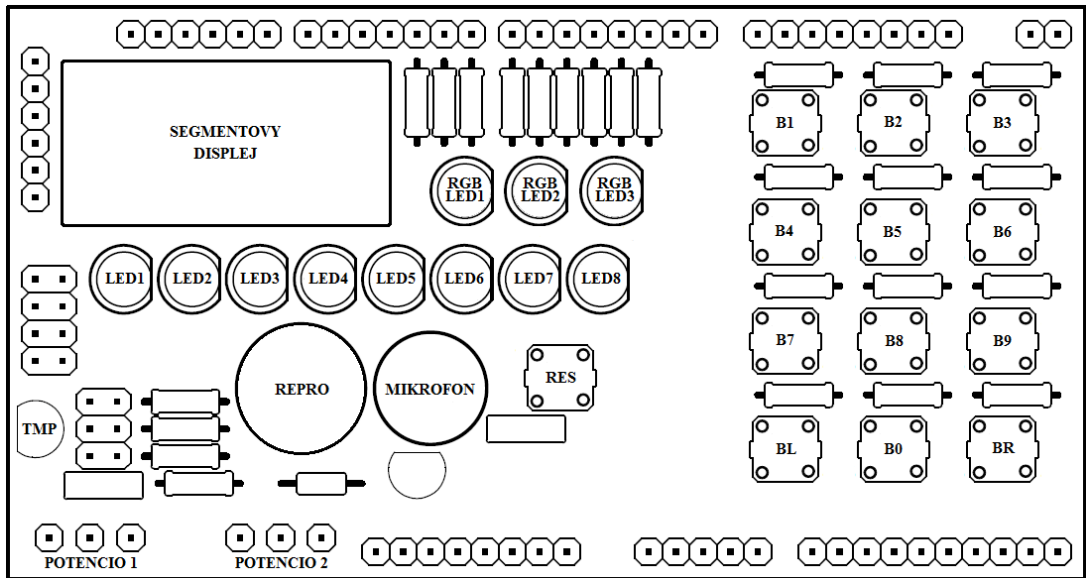
Z praktického a estetického hlediska by rozměry vyrobeného modulu měly korespondovat s deskou, ke které bude připojen. Díky značnému množství aktivních součástí nelze při použité technologii výsledný modul realizovat pouze na jedné desce plošných spojů o rozměrech 100 x 60 mm. Po důkladném zvážení všech možných řešení bylo rozhodnuto o realizaci modulu na dvou deskách, které nebudou překračovat rozměry desky Arduino Mega 2560.

Obě desky budou odnímatelné pomocí dutinkových kolíků, které jsou umístěny vždy na okrajích desky. Stejným způsobem bude modul připojen i na řídicí desku. Na spodní desce budou umístěny integrované obvody, které se starají o funkci jednotlivých prvků. Uživatel k těmto obvodům nebude mít důvod fyzicky přistupovat, proto jsou umístěny na spodní desce. Na vrchní desce budou realizovány všechny aktivní prvky, které uživatel používá pro zpětnou verifikaci nebo jako vstupní prvky.

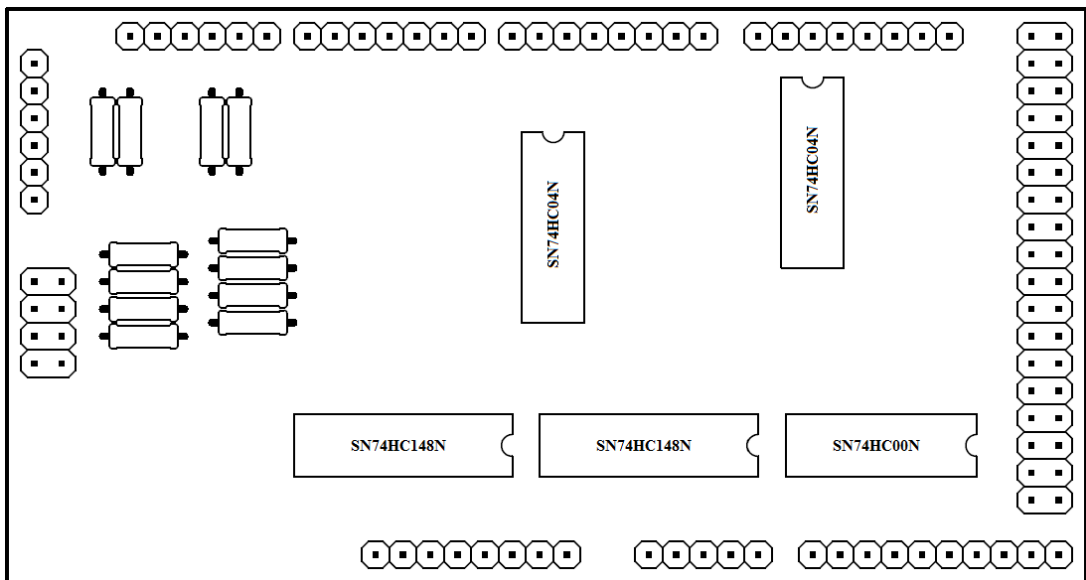
Díky nutnosti většího počtu pinů je výsledný modul kompatibilní pouze s deskami Arduino Due, Mega ADK a Mega 2560. K ostatním deskám nelze tento modul připojit, důvodem je již zmíněné četné obsazení pinů. Vyrobený modul je propojen s deskou pomocí více než čtyřiceti pinů. Jediné desky, disponující možností zapojení tolika pinů, jsou pouze ty, na kterých je použit procesor ATmega2560 nebo deska Arduino Due.

### 5.1 Rozmístění součástek na modulu

Zde jsou rozepsány jednotlivé prvky dílčích komponentů na deskách pro lepší orientaci ve schématech. Poskytují také vodítko pro zadávání nebo plnění různých programů.



Obrázek 23: Rozmístění součástek horní desky modulu



Obrázek 24: Rozmístění součástek spodní desky modulu

## 6 REALIZACE MODULU

Při samotné výrobě fyzického modulu se ukázalo, že daná hustota součástek a plošných spojů není v domácích podmínkách realizovatelná. Proto bylo využito možnosti výroby navrženého modulu u specializované firmy, která se zabývá výrobou desek plošných spojů. Celý postup výroby je popsán níže.

### 6.1 Vytvoření masky fotocestou

Samotné desky se skládají s nevodivého sklolaminátu a na jehož povrchu je přilepena vodivá vrstva mědi. Spojením těchto součástí vznikne tzv. cuprextit. Pro vytvoření jednotlivých cest, kterými povedou elektrické signály, je zapotřebí přebytečnou měď leptáním odstranit. Jednou z možností je vytvořit masku, pod jejíž vrstvou zůstane měď při leptání zachována. Tato metoda se provádí tzv. fotocestou.

Na polotovar cuprextitu se sprejem nanese tenká vrstva laku, která je citlivá na dopadající UV záření. Následně se na celý polotovar přiloží negativní otisk požadované struktury cest a otvorů. Vše se umístí pod zdroj UV záření. Na místech, která nejsou zakryta maskou, dojde k odstranění vrstvy laku chemickou reakcí. Nakonec se odstraní maska a celá deska se nechá vyvolat v roztoku NaOH (hydroxid sodný). Po jeho vytvrzení se lak stane stálým vůči UV záření.

### 6.2 Leptání

Leptání se provádí v roztoku  $\text{FeCl}_3$  (chlorid železitý). Plochy mědi, které nejsou chráněny vytvrzenou vrstvou laku z předchozího kroku, budou roztokem odleptány. Tím se vytvoří požadovaná struktura cest.

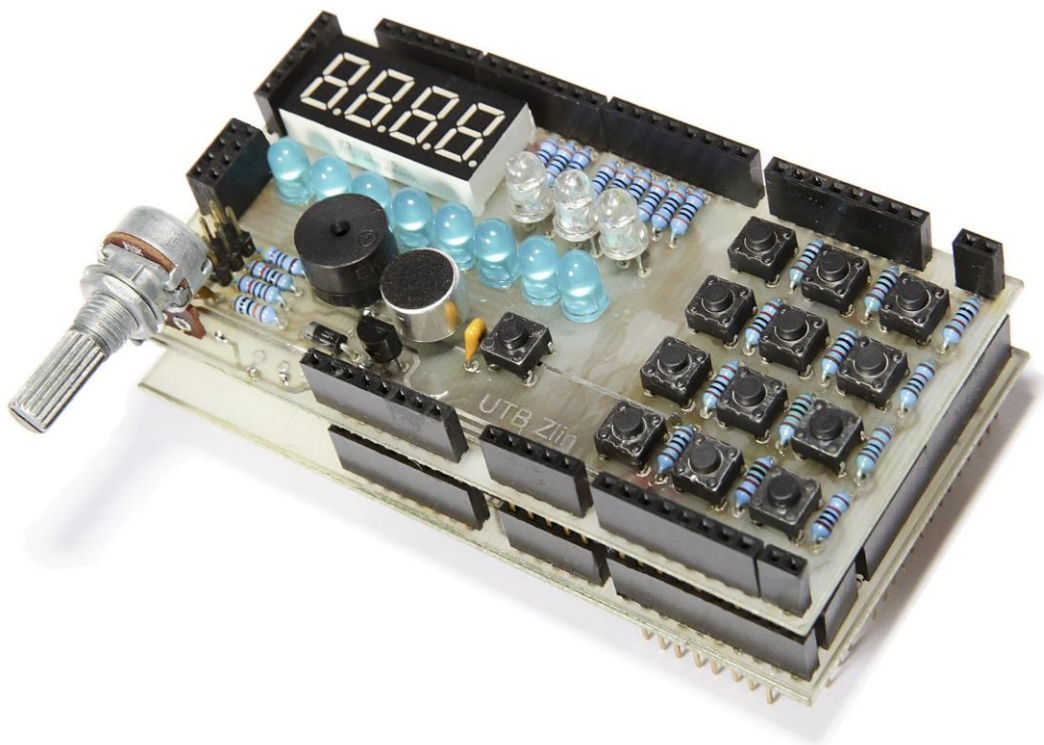
Samotné leptání se provádí přiložením desky stranou, kde se nachází měď k odleptání, dolů na hladinu leptacího roztoku. Díky zvýšené hustotě kapaliny desky na hladině plavou a nepotopí se. Proces leptání lze urychlit umělým prouděním kapaliny. Zde však hrozí riziko podleptání cest, které vede ke zničení výrobku. Průměrný čas leptání se pohybuje okolo deseti minut, záleží na koncentraci a stáří použitého roztoku.

### 6.3 Osazování součástek a jejich pájení

Ve finální části se deska osadí součástkami a jejich vývody se pájením spojí s vytvořenými cestami. Pro pájení se používají slitiny cínu a mědi, které mají bod tání okolo 220°C. Pro pájení velké hustoty cest a součástek se nejčastěji používá tzv. mikropájka. Úzkým hrotem, kterým disponuje, je možné se dostat snadněji k vývodům menších součástek.

Před pájením je nutné celou desku očistit a zbavit ji mastnoty. K neočištěným plochám mědi nepřilne roztavená slitina cínu. Pro větší přilnutí lze celou plochu desky potřít tzv. kalafunou. Jedná se pouze o přečištěnou pryskyřici. Ta navíc chrání plošné spoje proti oxidaci a korozi.

Na následujícím obrázku je uveden vyhotovený vstupně výstupní výukový modul pro platformu Arduino, který je realizován na dvou deskách.



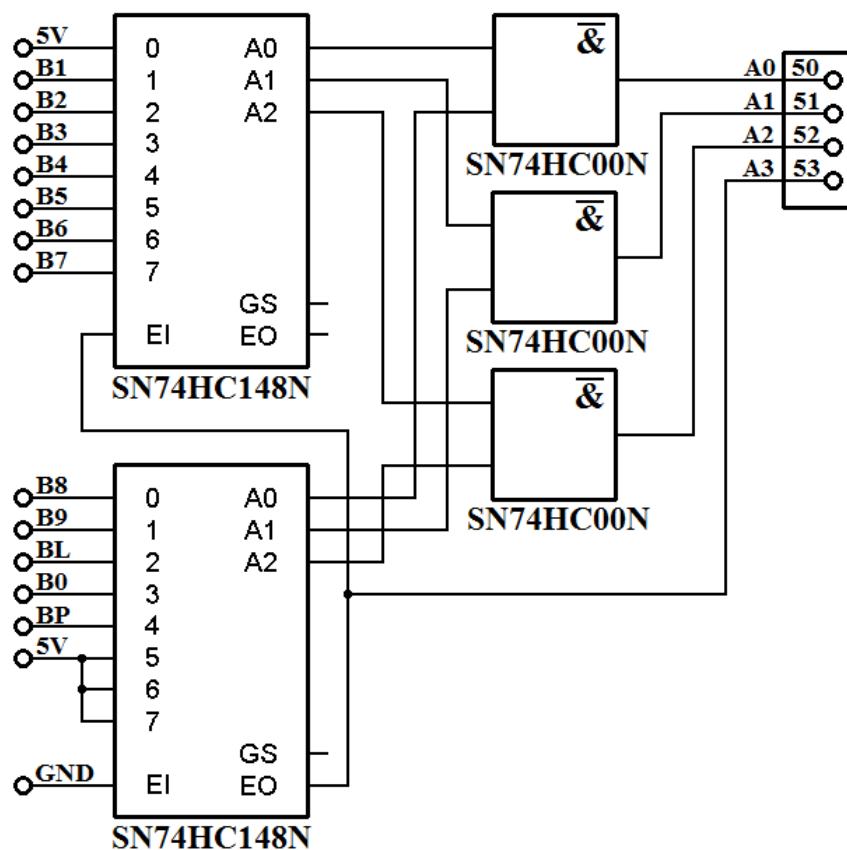
Obrázek 25: Vstupně výstupní výukový modul

## 7 POUŽITÉ PRVKY

V této části práce jsou vysvětleny a popsány jednotlivé prvky, které jsou na modulu zakomponovány. U každého prvku je uvedena pouze funkční zapojení pro jeho pochopení. Celková zapojení jednotlivých prvků jsou uvedena v příloze.

### 7.1 Maticová klávesnice

Nejdominantnější prvek, který je na modulu použit, je realizován pomocí dvanácti tlačítek. Ještě před samotným zpracováním signálu procesorem se stisknutá kombinace převádí obvody SN74HC148N na čtyř-bitové slovo. Redundantním převodem je snížen počet nutně obsazených pinů, které je nutné propojit s procesorem, pouze na čtyři. Dále je programové zpracování signálu zjednodušeno na čtení bitové informace, která přesně odpovídá adrese stisknutému tlačítku ve dvojkové soustavě.



Obrázek 26: Zapojení maticové klávesnice

Ke každému z dvanácti tlačítek je přiveden tzv. pull-up rezistor o velikosti 10 k $\Omega$ . V klidovém stavu tlačítka je na každou větev přivedeno napětí 5 V přes již zmíněné rezistory. Všech dvanáct větví je samostatně přivedeno na obvody SN74HC148N, kde probíhá přepočítání. Při stisku tlačítka je jeho větev přes pull-up rezistor zkratována na zem a na větvi, která je přivedena do enkodéru, se objeví napětí 0 V. Právě úroveň napětí 0 V (logickou nula) je klíčovou hodnotou pro překlopení obvodů a vygenerování jiného výstupu podle předem stanovených pravidel. Tato pravdivostní tabulka je uvedena níže.

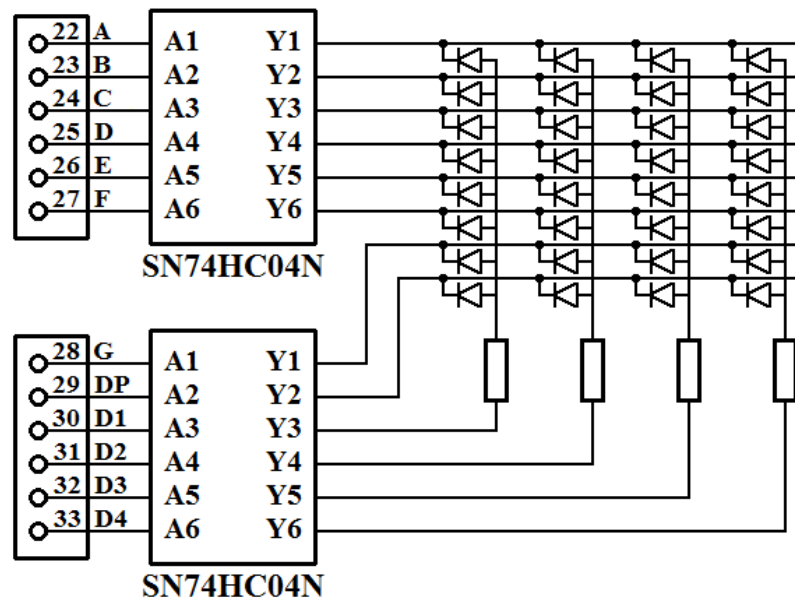
Vstupy												Výstupy			
BR	B0	BL	B9	B8	B7	B6	B5	B4	B3	B2	B1	A3 (53)	A2 (52)	A1 (51)	A0 (50)
1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1
1	1	1	1	1	1	1	1	1	1	0	x	0	0	1	0
1	1	1	1	1	1	1	1	1	0	x	x	0	0	1	1
1	1	1	1	1	1	1	1	0	x	x	x	0	1	0	0
1	1	1	1	1	1	1	0	x	x	x	x	0	1	0	1
1	1	1	1	1	1	0	x	x	x	x	x	0	1	1	0
1	1	1	1	0	x	x	x	x	x	x	x	1	0	0	0
1	1	1	0	x	x	x	x	x	x	x	x	1	0	0	1
1	1	0	x	x	x	x	x	x	x	x	x	1	0	1	0
1	0	x	x	x	x	x	x	x	x	x	x	1	0	1	1
0	x	x	x	x	x	x	x	x	x	x	x	1	1	0	0

Tabulka 11: Pravdivostní tabulka maticové klávesnice

Z předchozí tabulky je patrné, že při stisku více než jednoho tlačítka se vždy bere v potaz pouze tlačítko s vyšší prioritou. Tento princip však funguje pouze v rámci jednoho enkodéru. Při zapojení více obvodů se daný problém řeší softwarově, nikoliv změnou zapojení.

## 7.2 Segmentový displej

Sedmi-segmentový displej, na kterém lze zobrazovat číslice Segment obsahuje celkem čtyři digity a jeho výrobní označení je HD-M324RD. Tento model disponuje pouze dvanácti vyvedenými piny. Osm pinů pro zobrazení jedné číslice s desetinnou tečkou a čtyři pro uzemnění každého digitu zvlášť. Pro správnou funkčnost displeje je zapotřebí znalost tzv. multiplexingu.



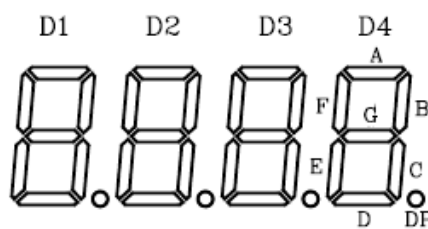
Obrázek 27: Zapojení segmentového displeje

Multiplexing ve spojení se segmentovým displejem, který používá uzemnění jednotlivých digitů, znamená zobrazení všech digitů v jednom časovém intervalu. Ten musí být kratší, než je doba, za kterou je schopno lidské oko zareagovat na změnu. V praxi to znamená, že se využije setrvačnosti lidského oka a na displeji se zobrazují digity po jednom neustále dokola. Dostatečně rychlá přeměna se pak lidskému oku jeví, jakoby svítily všechny digity současně, i když tomu tak není.

Pro lepší pochopení a využití při výuce je vhodnější použít displej, který je zapojen tzv. se společnou katodou. Bohužel, použitý displej je k dispozici pouze se společnou anodou. Za použití dvou invertorů SN74HC04N je však dosaženo tíženého zapojení. Další výhodou použití invertorů je ten, že oba mají vyšší maximální výstupní proud. Tím je chráněn samotný procesor před případným poškozením.

Segment	Pin Arduino
A	22
B	23
C	24
D	25
E	26
F	27
G	28
DP	29
D1	30
D2	31
D3	32
D4	33

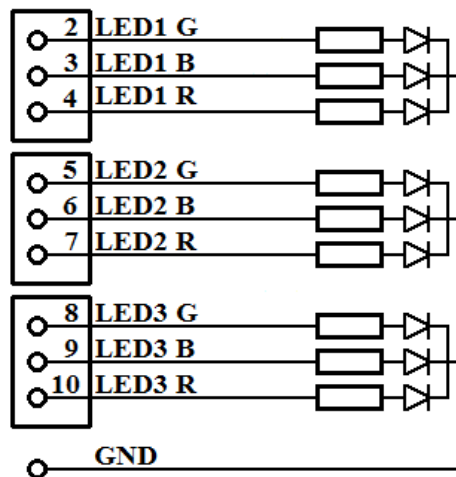
Tabulka 12: Propojení segmentového displeje s deskou [9]



Obrázek 28: Adresace jednotlivých diod segmentového displeje [9]

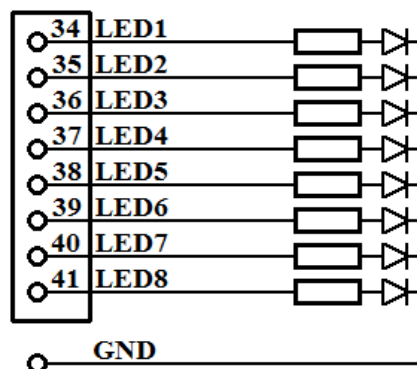
### 7.3 LED moduly

Dalším použitým prvkem jsou RGB LED. Díky nutnosti zapojení tří pinů na jednu součástku jsou na desce realizovány pouze tři. Všechny jsou připojeny na výstupní piny, které disponují PWM. Dále má každá barva svůj vlastní kanál, takže je zde možnost realizovat vzájemné propojení barev. Prakticky lze pomocí červené, zelené a modré vytvořit kteroukoliv barvu z viditelného světelného spektra.



Obrázek 29: Zapojení GBR LED modulu

Na desce se nachází ještě jeden LED modul, který je umístěn pod sedmi-segmentovým displejem. Ten už je sestaven pouze z osmi modrých LED. S procesorem jsou spojeny všechny samostatně pro jednodušší ovládání.

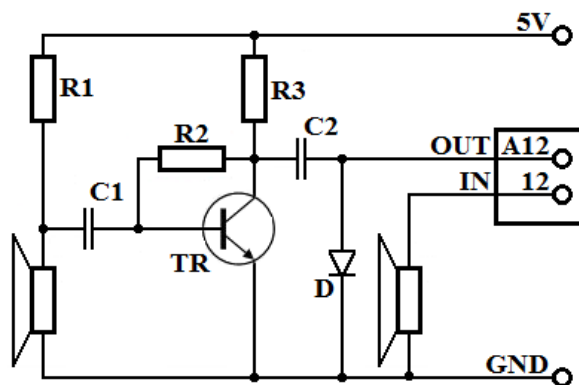


Obrázek 30: Zapojení LED modulu

## 7.4 Audio rozhraní

Na samotné desce se nachází i základní zvukové rozhraní, přes které lze s deskou komunikovat. Toto rozhraní je sestaveno z vysílací a přijímací části.

Vysílací část obsahuje magnetodynamický měnič, který je připojen na pin, který disponuje PWM. Díky tomu lze ovládat výstupní frekvenci a tím i tón měniče. Vývojové prostředí obsahuje přímo funkci `tone()`, kterou lze magnetodynamický měnič ovládat. Ovládání tohoto zařízení vyžaduje vyšší znalosti jazyka C.

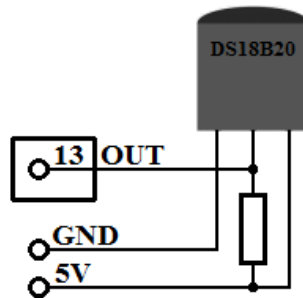


Obrázek 31: Zapojení audio modulu

Výstupní část je sestavena z elektrického obvodu, který zesiluje slabé elektrické impulzy, které vystupují z elektretového mikrofónu. Tyto impulzy se vyvolávají na základě dopadu zvukové vlny na mikrofón. Míra tlaku, vytvořeného na něj, je přímo úměrná velikosti výstupního signálu, který zpracovává procesor.

## 7.5 Teplotní čidlo

Jedná se o nejsložitější část celého modulu. Komunikace mezi teploměrem a procesorem probíhá sériově po jednom vodiči, nikoli analogově. Uvnitř pouzdra teploměru se nachází samostatný procesor, který zpracovává aktuální teplotu na šestnácti-bitové slovo, které pak posílá hlavnímu procesoru.



Obrázek 32: Zapojení teplotního čidla

Ovládání samotného procesoru je velmi složité. Před samotným sejmutím aktuální teploty z teploměru se musí procesoru posílat řídicí signály pro přípravu, sejmutí a odeslání. Tato komunikace navíc probíhá sériově pouze na jednom vodiči. Pro práci s tímto prvkem je nutné stažení speciální knihovny *OneWire.h*, která komunikaci po jednom vodiči podporuje. Teplotní čidlo je schopno vyhodnocovat teplotu od  $-55\text{ }^{\circ}\text{C}$  do  $+125\text{ }^{\circ}\text{C}$  s citlivostí  $0,5\text{ }^{\circ}\text{C}$ .

Teplota	Binární hodnota
+ 125 °C	0000 0000 1111 1010
+ 25 °C	0000 0000 0011 0010
+ 0,5 °C	0000 0000 0000 0001
0 °C	0000 0000 0000 0000
- 0,5 °C	1111 1111 1111 1111
- 25 °C	1111 1111 1100 1110
- 55 °C	1111 1111 1001 0010

Tabulka 13: Přepoččet teploty na bitovou informaci [7]

Zde jsou uvedeny základní příkazy pro bezchybné čtení teploty. Doplnkové příkazy jsou k dispozici v manuálu daného zařízení. Samotné zaslání řídicích signálů se provádí vždy příkazem `ds.write()`. Jako argument této funkce se uvádí hexadecimální číslo uvedené v závorkách jednotlivých příkazů.

### Převod teploty [0x44]

Tento příkaz slouží k převedení a zpracování teploty. Zpracování spočívá pouze v uložení sejmuté teploty do příslušného místa v paměti, konkrétně do dvou bytových registrů. [7]

### Odeslání dat [0xBE]

Tímto příkazem je umožněno samotné čtení z mapy paměti, ve kterých jsou uložena data z předešlých činností. Před vykonáváním dalších příkazů je nutné toto spojení nejdříve přerušit. [7]

Po otevření komunikačního kanálu mezi teplotním čidlem a procesorem má programátor na výběr z celé paměti teplotního čidla. Konkrétní typ informace vybere pomocí příkazu `data[i] = ds.read()`, kde `i` prezentuje číslo čteného bytu z paměti. Popis jednotlivých bytů je uveden na následujícím obrázku.

byte 0	Temperature LSB
byte 1	Temperature MSB
byte 2	Register of User Byte 1
byte 3	Register of User Byte 2
byte 4	Configuration Register
byte 5	Reserved
byte 6	Reserved
byte 7	Reserved
byte 8	CRC

Obrázek 33: Mapa paměti teplotního čidla [7]

## 8 UKÁZKOVÉ PROGRAMY

V následující kapitole jsou uvedeny některé ukázkové programy, které mohou být na modulu demonstrovány. Pro pochopení principu programu jsou zde uvedeny pouze nejdůležitější části zdrojových kódů. Kompletní zdrojové kódy jsou uvedeny v příloze.

Ukázkové programy jsou navrženy s postupně se zvyšující obtížností. Jsou koncipovány tak, aby se student seznámil a osvojil si všechny jeho části. Každá dílčí podkapitola obsahuje kromě názvu programu a zadání také jeho kompletní vypracování, včetně komentářů.

Pro větší přehlednost je tabulka všech použitých vstupů a výstupů na piny Arduina Mega 2560 je uvedena v příloze.

### 8.1 Světelný had z LED

Vytvořte program, který bude automaticky ovládat LED modul (LED1 až LED8). Program po spuštění bude nezávislý na uživateli. V první části programu se postupně zleva doprava za sebou rozsvítí všech osm LED po intervalu 100 ms. Po rozsvícení poslední se začnou LED postupně zleva doprava zhasínat. Celý cyklus se následně zopakuje v opačném směru, tedy zprava doleva.

Adresaci všech LED realizujte pomocí jednorozměrného pole, které obsahuje čísla pinů jednotlivých diod. Samotné cykly realizujte pomocí funkce **for()**, nikoli nastavením každé LED zvlášť.

#### Řešení:

Hlavní část programu je tvořena nekonečným cyklem, ve kterém jsou umístěny čtyři cykly **for()**. Každý cyklus se provede 8x. Při každém provedení se automaticky pomocí proměnné **i** přepne číslo pinu LED v poli **Pins[]** a na aktuální pin se zapíše logická jednička, nebo logická nula příkazem **digitalWrite()**.

```
void loop()
{
  int Pins[] = {34, 35, 36, 37, 38, 39, 40, 41};

  for (i = 0; i <= 7; i++)
  {
    digitalWrite(Pins[i], HIGH);
    delay(100);
  }
  for (i = 0; i <= 7; i++)
  {
    digitalWrite(Pins[i], LOW);
    delay(100);
  }
}
```

Obrázek 34: Hlavní část programu světelný had pomocí LED

## 8.2 Střídání barev na RGB modulu

Vytvořte program, který se bude ovládat pomocí tlačítka B1. Program bude dělen na tři režimy, které budou ovládat střídání barev na RGB LED modulu. Po spuštění programu budou všechny tři LED blikat zeleně, v intervalu 200 ms. Při stisknutí tlačítka B1 se bude měnit barva na všech třech LED. V celém programu se tedy budou měnit blikající barvy červená, zelená a modrá v nekonečné smyčce pomocí tlačítka B1.

Program bude reagovat i na stisknutí tlačítka B2. Po jeho aktivaci se přeruší smyčka a blikání se zastaví. Při opětovném stisku bude program pokračovat ve své činnosti v místě jeho přerušení.

### Řešení:

Z následujícího zdrojového kódu je patrné, že piny, na kterých jsou připojeny RGB LED, 2 až 10 jsou nastaveny jako výstupní piny. Zatímco piny klávesnice 50 a 51 jsou nastaveny jako vstupní.

```
void setup()
{
  for (int thisPin = 2; thisPin <= 10; thisPin++)
  {
    pinMode(thisPin, OUTPUT);
  }
  pinMode(50, INPUT);
  pinMode(51, INPUT);
}
```

Obrázek 35: Inicializace pinů programu střídání barev na RGB modulu

Střídání tří režimů pomocí tlačítka B1 bude zjištěno pomocí tří podmínek funkcí **for()** a pomocné proměnné **x**. Po spuštění programu je proměnná **x** rovna nule, tím se v nekonečném cyklu funkce **void loop()** vykonává pouze první podmínka **for(x == 0)**. Při stisku tlačítka B1 se proměnná inkrementuje a program následně vykonává pouze druhou podmínku **for(x == 1)**. Při stisknutí tlačítka B1 při **x == 3** se tato proměnná automaticky přepíše na nulu a celý proces se opakuje.

```
void loop()
{
  button1 = digitalRead(50);
  button2 = digitalRead(51);

  if(button1 == HIGH)
  {
    x++;
    delay(400);
  }

  if(button2 == HIGH)
  {
    delay(400);
    while(y == 0)
    {
      button2 = digitalRead(51);
      if(button2 == HIGH)
      {
        y = 1;
      }
    }
    y = 0;
  }
}
```

Obrázek 36: Hlavní část programu střídání barev na RGB modulu

Stisknutím tlačítka B2 se program dostane pomocné smyčky **while(y == 0)**. Zde setrvá až do chvíle, než bude splněna její podmínka. V této chvíli program pouze aktualizuje stav tlačítka B2 pomocí funkce **digitalRead()**. Při opětovném stisku tlačítka B2 se splní podmínka pomocné smyčky a program pokračuje ve své činnosti.

Samotné blikání LED je realizováno pomocí již zmíněných tří cyklů **for()**, z nichž má každý jinou podmínku. Na základě hodnoty **x** vykonává program pouze jednu ze zmíněných smyček. Každá smyčka obsahuje stejný zápis, mění se pouze číslo pinu, na který je daná logická hodnota zapisována.

```
if(x == 0)
{
    digitalWrite(4, HIGH);
    digitalWrite(7, HIGH);
    digitalWrite(10, HIGH);
    delay(200);
    digitalWrite(4, LOW);
    digitalWrite(7, LOW);
    digitalWrite(10, LOW);
    delay(200);
}

if(x == 1)
{
    digitalWrite(2, HIGH);
    digitalWrite(5, HIGH);
    digitalWrite(8, HIGH);
    delay(200);
    digitalWrite(2, LOW);
    digitalWrite(5, LOW);
    digitalWrite(8, LOW);
    delay(200);
}
```

Obrázek 37: Blikání programu střídání barev na RGB modulu

### 8.3 Voltmetr

Vytvořte program voltmetr, který zobrazí na segmentovém displeji naměřenou hodnotu napětí (0 V až 5 V) z adresy A12. Na této adrese se nachází potenciometr, jehož výstupní hodnota se bude dynamicky měnit a program musí pravidelně hodnotu aktualizovat, aby odpovídala aktuálnímu nastavení potenciometru.

Tlačítkem B1 bude možno měření pozastavit a zobrazit hodnotu v době jeho stisknutí na displeji. V této části program nebude reagovat na změnu polohy potenciometru. Opětovným stiskem tlačítka se funkce deaktivuje a displej bude opět zobrazovat hodnotu odpovídající poloze potenciometru.

#### Řešení:

První příkaz, kterým musí program provést, je sejmутí dané hodnoty z pinu A12 pomocí funkce **analogRead()**. Výchozím rozlišením převodníku je 10 bitů. Nejvyšší hodnota, kterou může převodník zobrazit, je 1023. Pro dosažení požadovaného rozmezí stačí naměřenou hodnotu vždy vydělit dvěma a posunout desetinnou čárku o dvě místa doleva. Pro snadnější zobrazování hodnot na displeji se ještě číslo převede do jednorozměrného pole, které obsahuje čtyři pozice.

```
number = (analogRead(A11)/2);  
  
for (i = 4; i >= 0; i--)  
{  
    x[i] = number % 10;  
    number /= 10;  
}
```

Obrázek 38: Sejmутí a převod hodnoty programu Voltmetr

Po sejmutí a převedení hodnoty do pole `x[]` už stačí hodnotu pouze správně zobrazit. Jelikož používáme displej bez řadiče, kde mají všechny digity společnou anodu a aktivace jednotlivých digitů se děje uzemněním jednotlivých katod, musíme použít již zmíněný multiplexing. Ten je realizován ve dvou smyčkách pomocí funkcí `for()`. Tyto funkce se provedou celkem 4x vždy s prvkem `i`, který je uložen v poli `x[i]`.

Pro snadné vybrání argumentu z více možností je v první smyčce `for()` použit příkaz `switch()`, jejímž argumentem je již zmíněná proměnná `i`. Ta postupně uzemňuje jednotlivé digity displeje. Uzemnění je realizováno zapsáním logické nuly na příslušný pin funkcí `digitalWrite()`. Tento postup je zachycen na následujícím zdrojovém kódu.

```
for (i = 1; i <= 4; i++)
{
  switch (i)
  {
    case 1:
      digitalWrite(D1, LOW);
      break;

    case 2:
      digitalWrite(D2, LOW);
      break;

    case 3:
      digitalWrite(D3, LOW);
      break;

    case 4:
      digitalWrite(D4, LOW);
      break;
  }
}
```

Obrázek 39: Uzemnění jednotlivých digitů programu Voltmetr

Ve druhé funkci `for()` je také pomocný příkaz `switch()`. Ta vybírá už jenom uložené číslice z pole `x[]`, které následně projdou výběrem, který je zobrazí na displeji. Situace je zobrazena na následujícím zdrojovém kódu.

```
for (i = 1; i <= 4; i++)
{
switch (x[i])
{
case 1: // zobrazení cislice 1
break;
case 2: // zobrazení cislice 2
break;
case 3: // zobrazení cislice 3
break;
case 4: // zobrazení cislice 4
break;
case 5: // zobrazení cislice 5
break;
case 6: // zobrazení cislice 6
break;
case 7: // zobrazení cislice 7
break;
case 8: // zobrazení cislice 8
break;
case 9: // zobrazení cislice 9
break;
case 0: // zobrazení cislice 0
break;
}
}
```

Obrázek 40: Zobrazení číslic programu Voltmetr

```
switch (x[i])
{
case 1:
digitalWrite(A, LOW);
digitalWrite(B, HIGH);
digitalWrite(C, HIGH);
digitalWrite(D, LOW);
digitalWrite(E, LOW);
digitalWrite(F, LOW);
digitalWrite(G, LOW);
digitalWrite(DP, LOW);
break;
}
```

Obrázek 41: Zobrazení číslice 1 programu Voltmetr

Pozastavení měření je realizováno pomocí pomocné podmínky `if()`, do které se program dostane po stisknutí tlačítka B1. Zde se hodnota pomocné proměnné `z` změní na jedničku. Následující podmínka `if()` na základě právě změněné proměnné `z` zabrání aktualizaci naměřené hodnoty a program zobrazuje hodnotu před stiskem tlačítka B1.

```
void loop()
{
  button1 = digitalRead(50);

  if(button1 == HIGH)
  {
    delay(100);
    if(z == 0)
    {
      z = 1;
    }
    else
    {
      z = 0;
    }
  }

  if(z == 1)
  {
    number = freeze;
  }
  else
  {
    number = (analogRead(A11)/2);
    freeze = number;
  }
}
```

Obrázek 42: Pozastavení měření programu Voltmetr

Při opětovném stisku tlačítka B1 se předchozí cyklus opakuje a umožní programu aktualizovat hodnotu na vstupu A/D převodníku.

## 8.4 Melodie

Vytvořte program pomocí magnetodynamického měniče, který je připojen na pinu 12. Pomocí změn frekvence tónu, jeho trvání a pauzách mezi nimi vytvořte libovolnou melodii. Pro ovládání magnetodynamického měniče používejte pouze funkci `tone()`, nikoliv funkci PWM.

### Řešení:

Pro snadnou implementaci melodie, například z notového zápisu, je vytvořen velmi podobný princip pro vkládání a čtení not, jako je tomu například u klavíru. Před samotným přehráváním melodie je zapotřebí určit defaultní délku trvání tónu, oktávu a BPM. Tyto náležitosti se zapisují před samotné noty, které jsou uloženy v poli `char *song = “ “;`

```
if(*p == 'd')
{
    p++; p++;
    num = 0;
    while(isdigit(*p))
    {
        num = (num * 10) + (*p++ - '0');
    }
    if(num > 0) default_dur = num;
    p++;
}
```

Obrázek 43: Určení defaultní délky trvání tónu programu Melodie

```
if(*p == 'o')
{
    p++; p++;
    num = *p++ - '0';
    if(num >= 3 && num <=7) default_oct = num;
    p++;
}
```

Obrázek 44: Určení oktávy programu Melodie

```
if(*p == 'b')
{
    p++; p++;
    num = 0;
    while(isdigit(*p))
    {
        num = (num * 10) + (*p++ - '0');
    }
    bpm = num;
    p++;
}
```

Obrázek 45: Určení BPM programu Melodie

Samotná nota pak obsahuje několik informací, aby mohla být správně interpretována. Těmito informacemi jsou změna délky trvání tónu, samotná nota, změna tónu na půltón a udání stupnice. Pokud některá z těchto informací chybí, tak se automaticky pracuje s výchozí hodnotou, nastavenou za začátku programu.

```
num = 0;
while(isdigit(*p))
{
    num = (num * 10) + (*p++ - '0');
}

if(num) duration = wholenote / num;
else duration = wholenote / default_dur;
```

Obrázek 46: Informace o délce trvání tónu programu Melodie

```
if(*p == '#')
{
    note++;
    p++;
}
```

Obrázek 47: Informace o změně tónu na půltón programu Melodie

```
if(isdigit(*p))
{
    scale = *p - '0';
    p++;
}
else
{
    scale = default_oct;
}
```

Obrázek 48: Informace o stupnici programu Melodie

V poslední části programu se už pouze přehraje tón se všemi náležitostmi, které mu patří. Celou smyčkou postupně projdou všechny noty, které jsou uvedeny v textovém řetězci `char *song = “ “;`

```
if(note)
{
    tone(TONE_PIN, notes[(scale - 4) * 12 + note]);
    delay(duration);
    noTone(TONE_PIN);
}
else
{
    delay(duration);
}
```

Obrázek 49: Přehrání noty programu Melodie

## 8.5 Měření teploty

Vytvořte program, který bude snímat aktuální teplotu na teplotním čidle, připojeném k mikropočítači na adrese 13. Měření provedte pouze pro kladné hodnoty v rozsahu od 0 do 99. Naměřené hodnoty zobrazte na prvních dvou pozicích segmentového displeje. Na posledních pozicích se bude zobrazovat jednotka naměřené veličiny. V tomto případě se jedná o °C. Hodnota teploty se bude každou sekundu automaticky aktualizovat.

### Řešení:

Pro správnou komunikaci musí být v programu přesně implementovány instrukce, které nalezneme v popisu teplotního čidla na straně 47 nebo v manuálu. Nejprve se příkazem **ds.search(addr)** provede inicializace připojeného zařízení a načtení jeho adresy. Dále se příkazem **ds.write(0x44)** pošle do zařízení příkaz na zpracování aktuální teploty. Nakonec se ještě pošle příkaz **ds.write(0xBE)**, kterým se zpracovaná teplota pošle zpět do mikroprocesoru.

```
int getTemp()
{
    if(!ds.search(addr))
    {
        ds.reset_search();
    }

    ds.reset();
    ds.select(addr);
    ds.write(0x44,1);

    byte present = ds.reset();
    ds.select(addr);
    ds.write(0xBE);

    data[0] = ds.read();
    ds.reset_search();
    byte LSB = data[0];
    int TemperatureSum = LSB / 2;

    return TemperatureSum;
}
```

Obrázek 50: Hlavní část programu Měření teploty

Teplotní čidlo posílá po vodiči celkem osm informací **data[0]** až **data[7]**, kde pouze první dvě slova byte nesou informaci o teplotě, ostatní přenášejí informaci o stavu nebo případných chybách. Pro měření kladných hodnot stačí sejmout pouze první slovo, které se následně převede z binární podoby na dekadickou. Jelikož změna teploty narůstá o 0,5 °C, musí se konečná hodnota před zobrazením vydělit dvěma.

K mikroprocesoru není připojen žádný externí přerušovač, který by signalizoval přesné uplynutí jedné vteřiny, za kterou se má hodnota teploty aktualizovat. Proto je zde doba jedné vteřiny pouze odhadnuta.

```
if(time == 38)
{
    temperature = getTemp();
    time = 0;
}

time++;
```

Obrázek 51: Časové zpoždění programu Měření teploty

Část zdrojového kódu, pro zobrazování aktuální teploty na displeji, je totožná se zdrojovým kódem, který byl použit v úloze Potenciometr.

## ZÁVĚR

Cílem této práce bylo vytvořit výukový modul pro platformu Arduino. Ten je, díky zachování kompaktních rozměrů, realizován na dvou deskách o rozměrech 100 x 66 mm. Na spodní desce jsou umístěny podpůrné komponenty, které se starají o převod nebo modifikaci elektrických signálů. Na vrchní jsou umístěny komponenty, které jsou určeny pro ovládání modulu, nalezneme zde i komponenty sloužící pro zpětnou verifikaci.

Na modulu jsou použity prvky maticová klávesnice, segmentový displej, RGB LED, LED, potenciometry, teplotní čidlo, elektretový mikrofon a magnetodynamický měnič. Samotný návrh byl vypracován v grafickém prostředí EAGLE. V příloze jsou uvedeny jak celkové zapojení jednotlivých desek modulu, tak i čísla pinů, se kterými jsou jednotlivé komponenty propojeny.

Díky velkému množství jednotlivých prvků a cest mezi nimi byl nakonec návrh realizován firmou, která se výrobou plošných spojů zabývá. Samotný postup výroby je popsán v kapitole realizace modulu.

Součástí práce je i vypracování ukázkových programů, které by měly demonstrovat možnosti využití výukového modulu. Škála programů je velmi široká, pro názornou ukázkou byly vybrány programy pro zobrazování světelného hada z LED, střídání barev na RGB modulu, simulace voltmetru, přehrávání melodie a měření teploty. Programy byly v této podobě zvoleny z důvodu postupně se zvyšující obtížnosti. Každé cvičení obsahuje slovní zadání a jeho následné zpracování včetně komentářů dílčích částí programu. V příloze jsou uvedeny kompletní zdrojové kódy všech programů. Tyto kódy jsou také uvedeny na přiloženém CD ve formátu \*.ino.

**SEZNAM POUŽITÉ LITERATURY**

- [1] ARDUINO. Arduino - HomePage [online]. 2011 [cit. 2014-01-08]. Dostupné z: <http://www.arduino.cc/>
- [2] BANZI, Massimo. Getting Started with Arduino. Second Edition. Sebastopol: O'Reilly Media, 2011, 128 s. ISBN 978-14-4930-987-9.
- [3] MANN, Burkhard. C PRO MIKROKONTROLÉRY: ANSI-C, kompilátory C, spojovací programy - linkery, práce s ATMEL AVR a MSC-51, příklady programování v jazyce C, nástroje pro programování, tipy a triky. Praha: BEN - technická literatura, 2003, 280 s. ISBN 80-7300-077-6.
- [4] MARGOLIS, Michael. Arduino Cookbook. Second Edition. Sebastopol: O'Reilly Media, 2011, 724 s. ISBN 978-1-449-31387-6.
- [5] PINKER, Jiří. Mikroprocesory a mikropočítače. Praha: BEN - technická literatura, 2004, 159 s. ISBN 80-7300-110-1.
- [6] ATMEL CORPORATION. 8-bit Atmel Microcontroller with 64K/128K/256K Bytes In-System Programmable Flash [pdf]. 2012, 448 s. [cit. 2014-01-08]. Dostupné z: <http://www.atmel.com/Images/doc2549.pdf>
- [7] DALLAS SEMICONDUCTOR. DS18B20 [pdf]. 2002, 20 s. [cit. 2014-03-08]. Dostupné z: <https://www.gme.cz/img/cache/doc/530/067/ds18b20-datasheet-1.pdf>
- [8] FEBOSS. Arduino Mega 2560 [pdf]. 2010, 8 s. [cit. 2014-01-08]. Dostupné z: <http://www.mantech.co.za/datasheets/products/A000047.pdf>
- [9] BRIGHT LED ELECTRONIC CORPORATION. BQ-M324RD [pdf]. 2005, 3 s. [cit. 2014-01-08]. Dostupné z: <http://www.gme.cz/img/cache/doc/512/924/hd-m324rd-datasheet-1.pdf>
- [10] HYLANDS, Jon. AVR AREF pin and AVCC pin. In: Society of Robots - Robot Forum [online]. 2007 [cit. 2014-03-22]. Dostupné z: <http://www.societyofrobots.com/robotforum/index.php?topic=606.0>

- [11] ŠMÍD, David. Netrpělivě očekávaná ATmega2560 nyní ihned k dispozici. HW SERVER S.R.O. HW.cz [online]. 2006 [cit. 2014-03-22]. Dostupné z: <http://www.hw.cz/produkty/netrpelive-ocekavana-atmega2560-nyni-ihned-k-dispozici.html>
- [12] HORÁČEK, Oldřich. #1 - Arduino Úvod. HORÁČEK, Oldřich. HW Kitchen [online]. 2007 [cit. 2014-03-22]. Dostupné z: <http://www.hwkitchen.com/news/arduino-uvod-1/>
- [13] WIZNET. W5100 Datasheet [pdf]. 2008, 70 s. [cit. 2014-03-22]. Dostupné z: [https://www.sparkfun.com/datasheets/DevTools/Arduino/W5100\\_Datasheet\\_v1\\_1\\_6.pdf](https://www.sparkfun.com/datasheets/DevTools/Arduino/W5100_Datasheet_v1_1_6.pdf)
- [14] SILVER TELECOM. Ag9000-S datasheet [pdf]. 2006, 14 s., 2007-03-01 [cit. 2014-02-22]. Dostupné z: <http://www.datasheet-pdf.com/datasheetdownload.php?id=573487>
- [15] STROLENÝ, Jaroslav. Znakové LCD displeje. STROLENÝ, Jaroslav. Procesory PIC12F629, PIC12F675, PIC16F84, PIC16F628, PIC16F630, PIC16F676, PIC16F876 a PIC16F877 [online]. 2000, 2014-01-30 [cit. 2014-02-22]. Dostupné z: <http://doveda.byl.cz/lcd/index.htm>
- [16] ITEAD STUDIO. LCD Keypad shield [pdf]. 2011, 2 s. [cit. 2014-02-22]. Dostupné z: <http://www.exp-tech.de/service/datasheet/LCD1602shield.pdf>
- [17] ARDUINO CORPORATION. Send in the clones [online]. 2013-07-10 [cit. 2014-03-20]. Dostupné z: <http://blog.arduino.cc/2013/07/10/send-in-the-clones/#.UysFv7TQRoD>
- [18] DIGI-KEY CORPORATION. A000088 Arduino [online]. 2013-02-25 [cit. 2014-03-24]. Dostupné z: <http://media.digikey.com/Photos/Arduino/A000088.jpg>
- [19] ARDUINO CORPORATION. Arduino Uno [online]. 2011-11-07 [cit. 2014-03-22]. Dostupné z: [http://arduino.cc/en/uploads/Main/ArduinoUno\\_R3\\_Front\\_450px.jpg](http://arduino.cc/en/uploads/Main/ArduinoUno_R3_Front_450px.jpg)

- [20] BANZI, Massimo. Open source matters in hardware, too [online]. 2013-10-23 [cit. 2014-03-22]. Dostupné z: <http://cdn.arstechnica.net/wp-content/uploads/2013/10/arduino-tre.jpg>
- [21] SOLARBOTICS CORPORATION. Arduino Ethernet Shield [online]. 2013-02-25 [cit. 2014-03-24]. Dostupné z: [https://content.solarbotics.com/products/photos/976c9706ae36521d4122cb13c7a474c1/med/51830\\_-\\_img\\_8769.jpg](https://content.solarbotics.com/products/photos/976c9706ae36521d4122cb13c7a474c1/med/51830_-_img_8769.jpg)
- [22] DROBOTICS CORPORATION. Arduino LCD Keypad Shield [online]. 2009-10-22 [cit. 2014-03-25]. Dostupné z: <http://www.droboticonline.com/media/catalog/product/cache/1/image/5e06319eda06f020e43594a9c230972d/a/r/arduinolcdkeypadforstore3.jpg>
- [23] INFINEON TECHNOLOGIES CORPORATION. New ESD/EMI Protection Device for microSD Card Interface - BGF 117 [online]. [cit. 2014-03-25]. Dostupné z: [http://www.infineon-designlink.com/images/article/diagram\\_0000095.jpg](http://www.infineon-designlink.com/images/article/diagram_0000095.jpg)
- [24] FUTURASHOP CORPORATION. Microcontrollore ATmega2560-16AU [online]. [cit. 2014-03-25]. Dostupné z: <https://www.futurashop.it/images/1340-ATMEGA2560-16AU.jpg>
- [25] LITOVSKY, Gustavo. How do I get the msp430 EZ430-RF2500 to communicate with an MPU-6000 gyroscope over SPI [online]. 2013-02-22 [cit. 22.4.2014]. Dostupné z: <http://i.stack.imgur.com/gzP0a.jpg>
- [26] ARDUINO CORPORATION. Arduino Mega 2560 [online]. [cit. 22.3.2014]. Dostupné z: [http://arduino.cc/en/uploads/Main/ArduinoMega2560\\_R3\\_Fronte.jpg](http://arduino.cc/en/uploads/Main/ArduinoMega2560_R3_Fronte.jpg)

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

A/D	Analog to digital
ASCII	American Standard Code for Information Interchange
BPM	Beats Per Minute
CISC	Complex Instruction Set Computer
CLK	Clock
CRC	Cyclic Redundancy Check
DIP	Dual Inline Package
DPS	Deska Plošných Spojů
EAGLE	Easily Applicable Graphical Layout Editor
EEPROM	Electrically Erasable Programmable Read Only Memory
I/O	Input/Output
IEEE	Institute of Electrical and Electronics Engineers
LED	Light-Emitting Diode
MISO	Master In Slave Out
MOSI	Master Out Slave In
PLC	Programmable Logic Controller
PoE	Power over Ethernet
PWM	Pulse Width Modulation
RISC	Reduced Instruction Set Computing
SMD	Surface Mount Device
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
SS	Slave Select
TTL	Transistor Transistor Logic
USB	Universal Serial Bus
UV	Ultraviolet

**SEZNAM OBRÁZKŮ**

Obrázek 1: Logo firmy Arduino [17].....	11
Obrázek 2: Deska Arduino Mini [18] .....	13
Obrázek 3: Deska Arduino Uno [19].....	14
Obrázek 4: Deska Arduino Tre [20] .....	15
Obrázek 5: Systém napojení více shieldů na sebe .....	17
Obrázek 6: Ethernet shield [21] .....	18
Obrázek 7: Adresace pinů mezi SD kartou a deskou Arduino [23].....	19
Obrázek 8: LCD Keypad shield [22] .....	20
Obrázek 9: Mikroprocesor ATmega2560 [24] .....	22
Obrázek 10: Rozdělení paměti procesoru ATmega2560 [6] .....	23
Obrázek 11: Blokové schéma mikroprocesoru ATmega2560 [6] .....	24
Obrázek 12: Ovládání portu A [6] .....	26
Obrázek 13: Arduino Mega 2560 [26].....	27
Obrázek 14: Ovládání digitálních pinů [2] .....	28
Obrázek 15: Ovládání výstupních pinů pomocí PWM [2] .....	28
Obrázek 16: Ovládání pinů přerušování [4] .....	29
Obrázek 17: Ovládání vstupních analogových pinů [2] .....	29
Obrázek 18: Rozměry desky Arduino Mega 2560 [8].....	30
Obrázek 19: Princip komunikace pomocí SPI [25] .....	31
Obrázek 20: Funkce a rozložení konektoru SPI .....	31
Obrázek 21: Vývojové prostředí Arduino 1.0.5.....	32
Obrázek 22: Vývojové prostředí EAGLE 6.5.0.....	34
Obrázek 23: Rozmístění součástek horní desky modulu .....	38
Obrázek 24: Rozmístění součástek spodní desky modulu .....	38
Obrázek 25: Vstupně výstupní výukový modul .....	40
Obrázek 26: Zapojení maticové klávesnice .....	41
Obrázek 27: Zapojení segmentového displeje .....	43
Obrázek 28: Adresace jednotlivých diod segmentového displeje [9].....	44
Obrázek 29: Zapojení GBR LED modulu .....	45

Obrázek 30: Zapojení LED modulu.....	45
Obrázek 31: Zapojení audio modulu .....	46
Obrázek 32: Zapojení teplotního čidla.....	47
Obrázek 33: Mapa paměti teplotního čidla [7] .....	48
Obrázek 34: Hlavní část programu světelný had pomocí LED .....	50
Obrázek 35: Inicializace pinů programu střídání barev na RGB modulu.....	51
Obrázek 36: Hlavní část programu střídání barev na RGB modulu .....	51
Obrázek 37: Blikání programu střídání barev na RGB modulu .....	52
Obrázek 38: Sejmутí a převod hodnoty programu Voltmetr.....	53
Obrázek 39: Uzemnění jednotlivých digitů programu Voltmetr .....	54
Obrázek 40: Zobrazení číslic programu Voltmetr .....	55
Obrázek 41: Zobrazení číslice 1 programu Voltmetr .....	55
Obrázek 42: Pozastavení měření programu Voltmetr.....	56
Obrázek 43: Určení defaultní délky trvání tónu programu Melodie.....	57
Obrázek 44: Určení oktávy programu Melodie .....	57
Obrázek 45: Určení BPM programu Melodie.....	58
Obrázek 46: Informace o délce trvání tónu programu Melodie.....	58
Obrázek 47: Informace o změně tónu na půltón programu Melodie .....	58
Obrázek 48: Informace o stupnici programu Melodie .....	59
Obrázek 49: Přehraní noty programu Melodie .....	59
Obrázek 50: Hlavní část programu Měření teploty.....	60
Obrázek 51: Časové zpoždění programu Měření teploty.....	61

**SEZNAM TABULEK**

Tabulka 1: Nejznámější desky Arduino a jejich základní parametry [1].....	12
Tabulka 2: Vlastnosti desky Arduino Mini [1] .....	13
Tabulka 3: Vlastnosti desky Arduino Uno [1] .....	15
Tabulka 4: Vlastnosti procesoru Atmel desky Arduino Tre [1,12].....	16
Tabulka 5: Vlastnosti procesoru Intel desky Arduino Tre [1,12] .....	16
Tabulka 6: Adresace pinů mezi W5100 a RJ-45 [13].....	18
Tabulka 7: Funkce signalizačních diod na shieldu [13] .....	19
Tabulka 8: Adresace vývodu na desku Arduino [16] .....	21
Tabulka 9: Vlastnosti procesoru Atmel Mega 2560 [6] .....	23
Tabulka 10: Vlastnosti desky Arduino Mega 2560 [1].....	27
Tabulka 11: Pravdivostní tabulka maticové klávesnice.....	42
Tabulka 12: Propojení segmentového displeje s deskou [9].....	44
Tabulka 13: Přepočítání teploty na bitovou informaci [7] .....	47

**SEZNAM PŘÍLOH**

- Příloha P I: Adresace všech využitých pinů
- Příloha P II: Použité součástky
- Příloha P III: Schéma zapojení Vrchní desky modulu
- Příloha P IV: DPS Vrchní desky modulu
- Příloha P V: Schéma zapojení Spodní desky modulu
- Příloha P VI: DPS Spodní desky modulu
- Příloha P VII: CD s bakalářskou prací a soubory obsahující zdrojové kódy

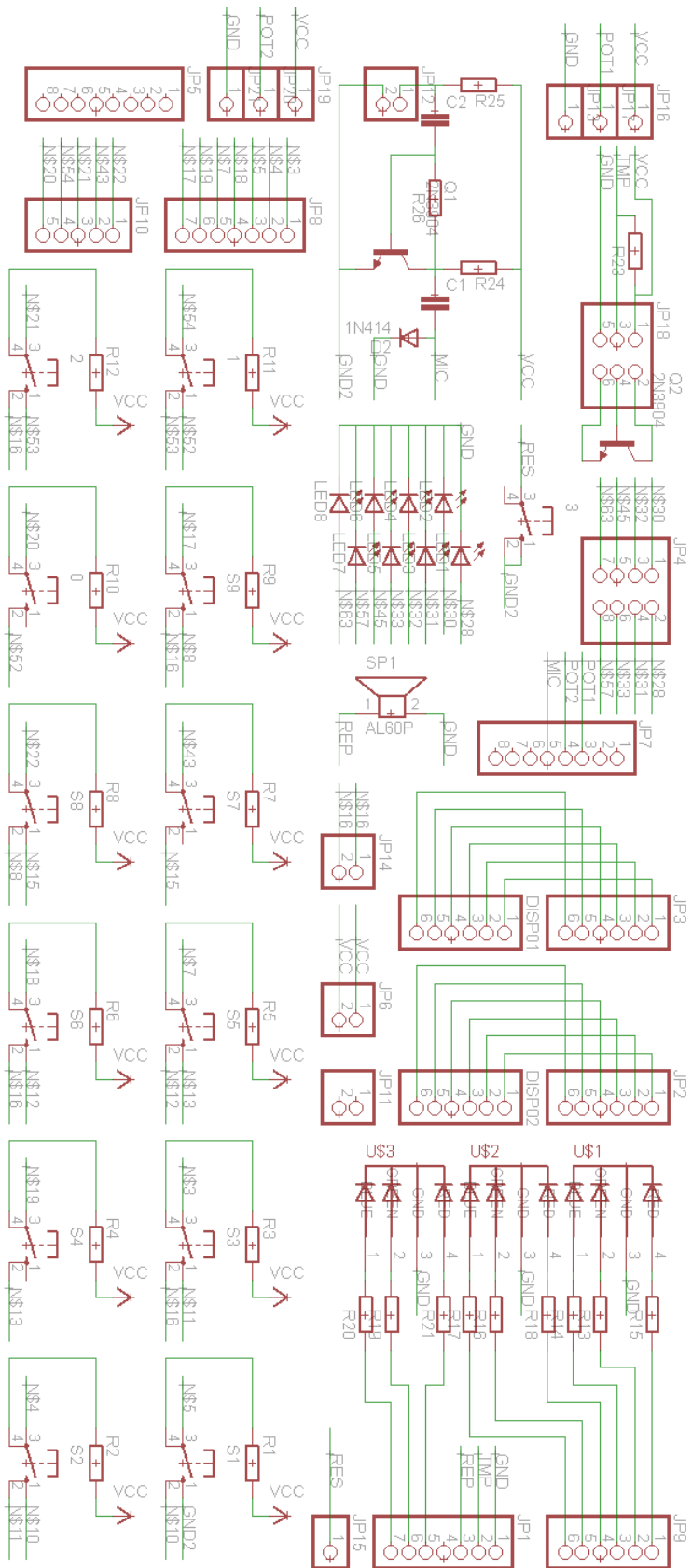
## PŘÍLOHA P I: ADRESACE VYUŽITÝCH PINŮ

Číslo pinu	Název	Modul
2	RGB 1 zelená	RGB modul
3	RGB 1 modrá	
4	RGB 1 červená	
5	RGB 2 zelená	
6	RGB 2 modrá	
7	RGB 2 červená	
8	RGB 3 zelená	
9	RGB 3 modrá	
10	RGB 3 červená	
A10	Potenciometr 1	
A11	Potenciometr 2	
12	Piezo reproduktor	Audio
A12	Mikrofon	
13	Teplotní čidlo	Teplotní čidlo
22	A	Segmentový displej
23	B	
24	C	
25	D	
26	E	
27	F	
28	G	
29	DP	
30	D1	
31	D2	
32	D3	
33	D4	
34	LED1	LED modul
35	LED2	
36	LED3	
37	LED4	
38	LED5	
39	LED6	
40	LED7	
41	LED8	
50	A0	Maticová klávesnice
51	A1	
52	A2	
53	A3	

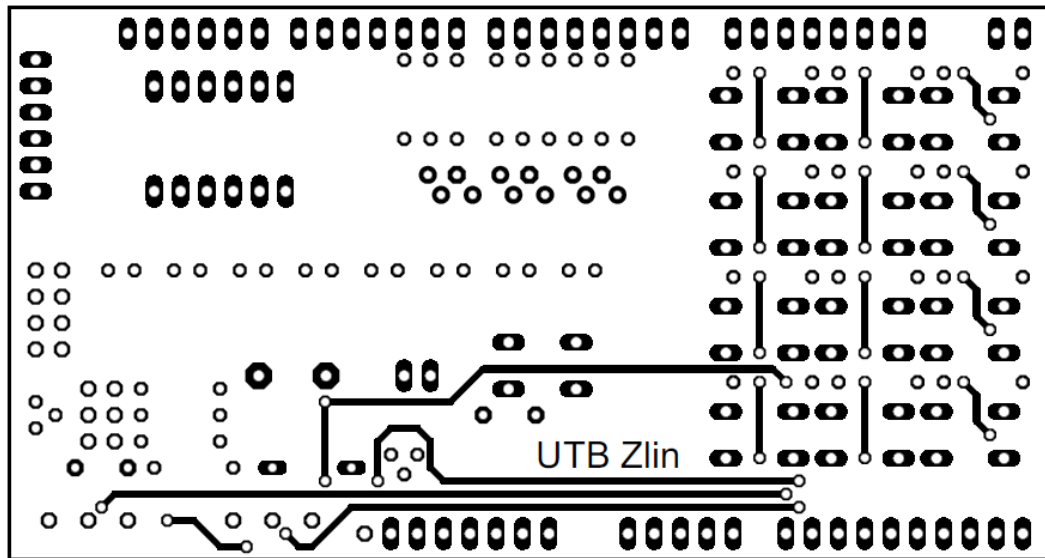
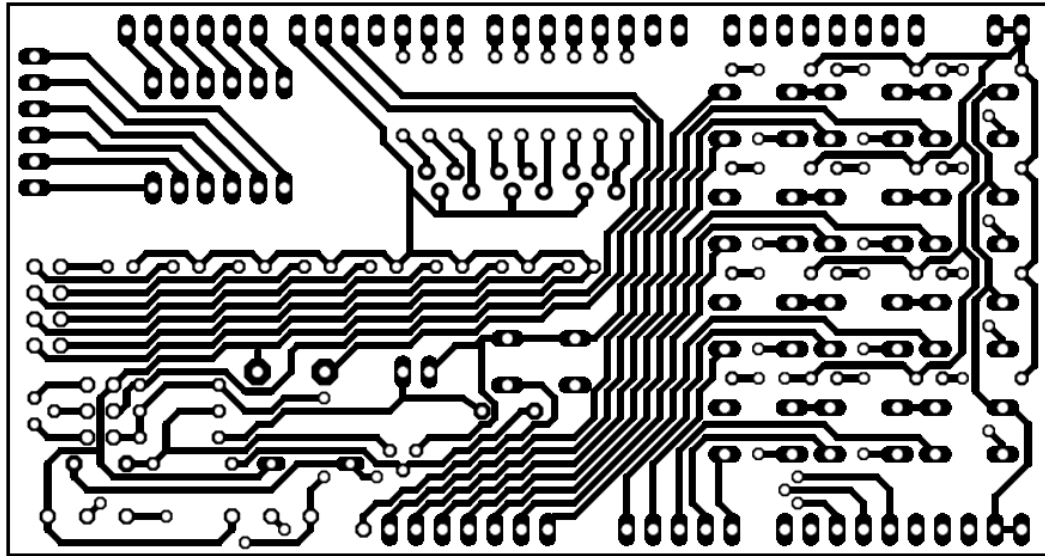
## PŘÍLOHA P II: POUŽITÉ SOUČÁSTKY

Název součástky	Označení	Hodnota	Počet
Rezistory	-	300	21
	-	10k	14
	-	100k	1
	-	4k7	1
Kondenzátory	-	0,1uF	2
Mikrospínače	TC-0103-T	-	13
Segmentový displej	HD-m324rd	-	1
Diody	LED RGB CC	-	3
	LED 5MM blue	-	8
	1N4007	-	1
Integrované obvody	SN74HC00N	-	1
	SN74HC04N	-	2
	SN74HC148N	-	2
Tranzistory	2N 3904	-	1
Elektretové mikrofony	MCE100	-	1
Magnetodynamické měniče	KPB1220	-	1
Teplotní čidla	DS 18B20	-	1
Potenciometry	PC1621NBKO10	-	2

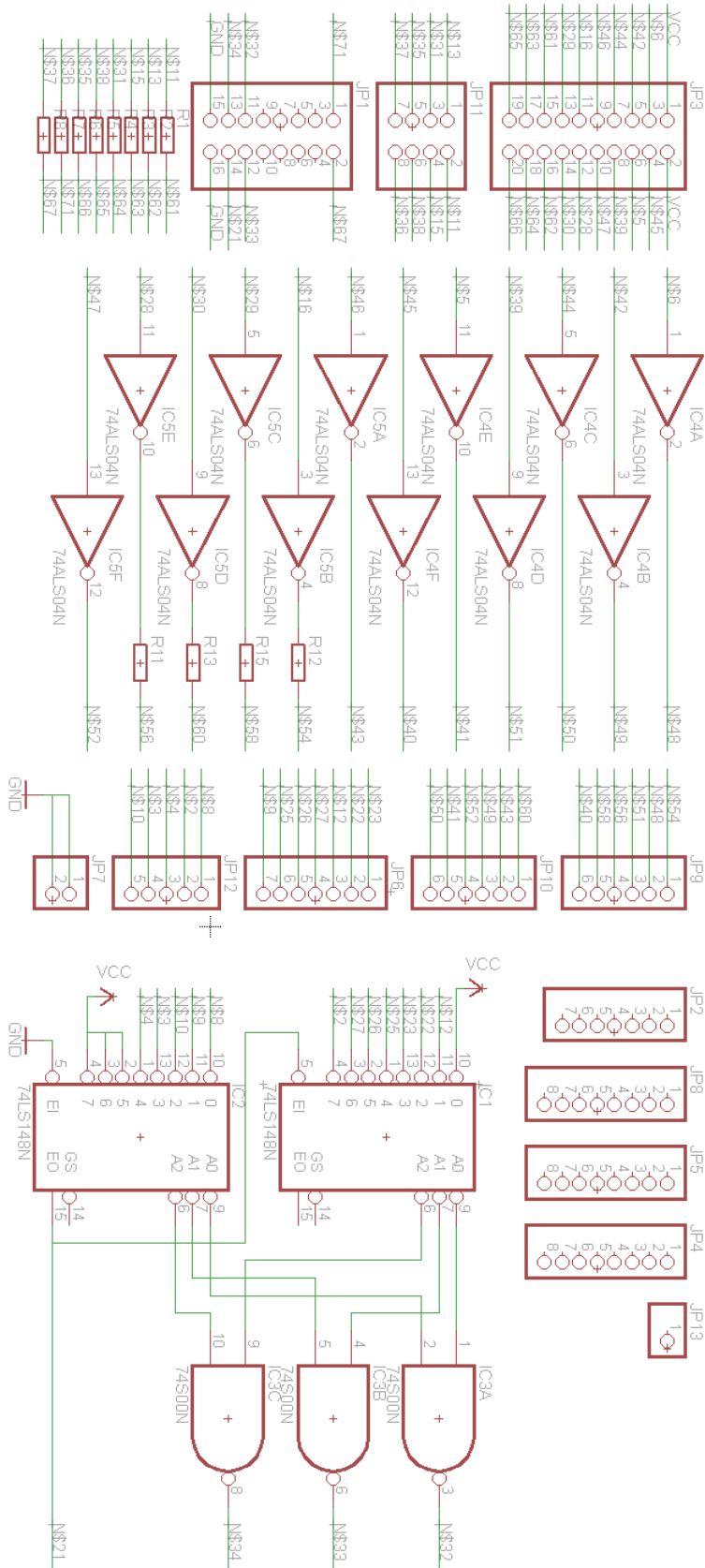
# PŘÍLOHA P III: SCHÉMA ZAPOJENÍ VRCHNÍ DESKY MODULU



# PŘÍLOHA P IV: DPS VRCHNÍ DESKY MODULU



# PŘÍLOHA P V: SCHÉMA ZAPOJENÍ SPODNÍ DESKY MODULU



# PŘÍLOHA P VI: DPS SPODNÍ DESKY MODULU

