

Univerzita Hradec Králové
Přírodovědecká fakulta
Katedra informatiky

Využití mikrokontroléru PICAXE pro výuku
základů programování a robotiky

Bakalářská práce

Autor: Jan Jehlička
Studijní program: B1801 / Informatika
Studijní obor: Informatika se zaměřením na vzdělávání
Základy techniky se zaměřením na vzdělávání
Vedoucí práce: Ing. Jiří Jelínek, Ph.D.

Univerzita Hradec Králové

Přírodovědecká fakulta

Zadání bakalářské práce

Autor: Jan Jehlička

Studijní program: B1801 / Informatika

Studijní obor: Informatika se zaměřením na vzdělávání
Základy techniky se zaměřením na vzdělávání

Název práce: Využití mikrokontroléru PICAXE pro výuku základů programování a robotiky

Název práce v AJ: PICAXE microcontroller for teaching the basics of programming and robotics

Cíl a metody práce: Cílem bakalářské práce je vytvořit modulovou stavebnici včetně ukázkových úloh pro jednotlivé moduly. Teoretická část práce obsahuje představení mikrokontrolérů PICAXE a jejich přehled. Praktická část představuje konstrukci univerzální výukové stavebnice založené na PICAXE. Součástí práce jsou jednotlivé úlohy pro každý z modulů obsahující ukázkový program, vývojový diagram, schéma propojení úlohy a náměty na další možné rozšíření jednotlivých úloh.

Garantující pracoviště: Katedra informatiky Přírodovědecké fakulty UHK

Vedoucí práce: Ing. Jiří Jelínek, Ph.D.

Konzultant:

Oponent: doc. RNDr. Štěpán Hubálovský, Ph.D.

Datum zadání práce: 20.2.2014

Datum odevzdání práce:

Prohlášení:

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a že jsem v seznamu použité literatury uvedl všechny prameny, z kterých jsem vycházel.

V Hradci Králové dne

Jan Jehlička

Poděkování:

Chtěl bych poděkovat svému vedoucímu bakalářské práce Ing. Jiřímu Jelínkovi, Ph.D. za odborné vedení, za pomoc a rady při zpracování této práce.

Anotace

JEHLIČKA, Jan. *Využití mikrokontroléru PICAXE pro výuku základů programování a robotiky*. Hradec Králové, 2015. Bakalářská práce na Přírodovědecké fakultě Univerzity Hradec Králové. Vedoucí bakalářské práce Ing. Jiří Jelínek, Ph.D. 92s

Cílem bakalářské práce je vytvořit univerzální modulovou stavebnici s mikrokontrolérem PICAXE včetně ukázkových úloh pro jednotlivé moduly. Práce přináší praktickou konstrukci a výsledkem je výuková stavebnice – vývojová deska s mikrokontrolérem PICAXE a základní rozšiřující vstupní a výstupní moduly, použitelné pro výuku základů programování mikrokontrolérů a práci s nimi. Práce obsahuje sbírku vzorových programů pro mikrokontrolér.

Klíčová slova

Mikrokontrolér, PICAXE, programování, robotika

Annotation

JEHLIČKA, JAN. *PICAXE microcontroller for teaching the basics of programming and robotics*. Hradec Králové, 2015. Bachelor Thesis at Faculty of Science University of Hradec Králové. Thesis Supervisor Ing. Jiří Jelínek, Ph.D. 92p.

The aim of this bachelor thesis is to create a universal module kit with PICAXE microcontroller, including sample tasks for each module. Bachelor thesis obtain practical design and the result is a educational kit - development board with PICAXE microcontroller and basic external input and output modules, which are useful for learning the basics of programming microcontrollers. A collection of sample programs for the microcontroller is included.

Keywords

Microcontroller, PICAXE, programing, robotics

Obsah

Úvod.....	11
Cíle práce	12
Teoretická část.....	13
1 Mikropočítače a Mikrokontroléry	14
1.1 Co je mikrokontrolér?.....	14
1.2 Koncepce MCU.....	15
1.2.1 Koncepce podle von Neumanna.....	15
1.2.2 Harvardská koncepce.....	16
2 Programování vybraných druhů mikrokontrolérů.....	17
2.1 Zavaděč.....	17
2.1.1 Bootloader Arduino	17
2.1.2 PICAXE firmware	17
2.2 Platforma Arduino	18
2.2.1 Moduly	18
2.2.2 Shieldy.....	19
2.2.3 Programovací jazyk	19
2.2.4 Vývojové prostředí.....	20
2.2.5 Využití Arduina.....	21
2.3 Raspberry Pi.....	22
2.3.1 GPIO piny	22
2.3.2 Gertboard.....	23
2.3.3 UniPi	24
2.3.4 Programovací jazyk a vývojové prostředí.....	25
3 Mikrokontroléry PICAXE.....	26
3.1 Představení PICAXE.....	26
3.2 Porovnání typů.....	26
3.2.1 Řada X.....	27
3.2.2 Řada M	27
3.3 PICAXE 20M2	27
3.3.1 Napájecí napětí.....	27
3.3.2 Frekvence	27
3.3.3 Piny a pinout.....	28

3.4	Organizace paměti	28
3.4.1	Paměť programu.....	28
3.4.2	Paměť dat	29
3.4.3	Paměť RAM (proměnné).....	29
3.5	Programování PICAXE.....	30
3.6	Vývojové prostředí.....	30
4	Mikropočítačové vývojové stavebnice.....	32
4.1	Hotová stavebnice.....	32
4.2	Nepájivé kontaktní pole.....	33
4.3	Návrh plošného spoje	34
4.4	Univerzální plošné spoje	35
	Praktická část.....	36
5	Konstrukce výukové stavebnice.....	37
5.1	Vývojová deska.....	37
5.1.1	MCU část.....	37
5.1.2	Komunikační obvod.....	37
5.1.3	Napájecí část 5V	38
5.1.4	Výkonová část	38
5.2	Vstupní moduly.....	40
5.2.1	Modul Tlačítka	40
5.2.2	Modul Klávesnice.....	40
5.2.3	Modul Potenciometry.....	41
5.2.4	Modul TOUCH.....	42
5.3	Výstupní moduly.....	43
5.3.1	Modul LED	43
5.3.2	Modul 7 segment.....	43
5.3.3	Modul LED strip	44
5.3.4	Modul RGB LED strip.....	45
5.3.5	Modul Motor	45
5.3.6	Modul Servo.....	46
5.3.7	Modul Krokový motor.....	47
5.3.8	Modul H-můstek.....	48
5.3.9	Modul Reproduktor	50

5.4	Stolní držák stavebnice	50
5.5	Systém propojování.....	50
6	Úlohy s LED diodami.....	52
6.1	Světelný had	52
6.1.1	Ukázkový program.....	52
6.1.2	Schéma propojení.....	53
6.1.3	Vývojový diagram.....	53
6.1.4	Náměty na rozšíření programu	54
6.2	Zobrazovač čísel.....	54
6.2.1	Ukázkový program.....	54
6.2.2	Schéma propojení.....	55
6.2.3	Vývojový diagram.....	55
6.2.4	náměty na rozšíření programu	55
6.3	Stmívač led pásku.....	56
6.3.1	Syntaxe příkazů	56
6.3.2	Ukázkový program.....	56
6.3.3	Schéma propojení.....	57
6.3.4	Vývojový diagram.....	57
6.3.5	Náměty na rozšíření programu	57
6.4	Kontrolér RGB pásku	58
6.4.1	Ukázkový program.....	58
6.4.2	Schéma propojení.....	58
6.4.3	Vývojový diagram.....	59
6.4.4	Náměty na rozšíření programu	59
7	Generování zvuků	60
7.1	Příkaz sound.....	60
7.1.1	Ukázkový program.....	60
7.1.2	Schéma propojení.....	61
7.1.3	Vývojový diagram.....	61
7.1.4	Náměty na rozšíření programu	62
7.2	Příkaz tune	62
7.2.1	Ukázkový program.....	63
7.2.2	Schéma propojení.....	63

7.2.3	Vývojový diagram.....	64
7.2.4	Náměty na rozšíření programu.....	64
7.3	Bzučák.....	65
7.3.1	Ukázkový program.....	65
7.3.2	Schéma propojení.....	65
7.3.3	Náměty na rozšíření programu.....	65
7.3.4	Vývojový digram.....	66
8	Motorické úlohy.....	67
8.1	Ovládání serva.....	67
8.1.1	Ukázkový program.....	67
8.1.2	Schéma propojení.....	68
8.1.3	Vývojový diagram.....	69
8.1.4	Náměty na rozšíření programu.....	70
8.2	Reverzace chodu motoru.....	70
8.2.1	Ukázkový program.....	70
8.2.2	Schéma propojení.....	71
8.2.3	Vývojový diagram.....	71
8.2.4	Náměty na rozšíření programu.....	71
8.3	Krokový motor.....	72
8.3.1	Ukázkový program.....	72
8.3.2	Vývojový diagram.....	73
8.3.3	Schéma propojení.....	74
8.3.4	Náměty na rozšíření programu.....	74
9	Klávesnice a dotyková tlačítka.....	75
9.1	Klávesnice komunikující s počítačem.....	75
9.1.1	Ukázkový program.....	76
9.1.2	Schéma propojení.....	77
9.1.3	Vývojový digram.....	77
9.1.4	Náměty na další rozšíření programu.....	77
9.2	Dotykový vypínač.....	78
9.2.1	Ukázkový program.....	79
9.2.2	Schéma propojení.....	80
9.2.3	Vývojový diagram.....	80

9.2.4	Náměty na rozšíření programu	80
	Shrnutí vlastností stavebnice	81
	Závěr	82
	Seznam zkratk	83
	Citovaná literatura	84
	Seznam obrázků	87
	Seznam tabulek	89
	Přílohy	90
	Příloha A	90
	Příloha B	90
	Příloha C	91

Úvod

Práce se zabývá konstrukcí univerzální výukové stavebnice s mikrokontrolérem PICAXE. Práce přináší inspiraci možnosti konstrukce univerzální vývojové desky a základních nekomplikovaných vstupních a výstupních obvodů sestavitelných i na kontaktním nepájivém poli.

Při volbě tématu jsem musel počítat s větší časovou a finanční náročností, kdy nezanedbatelný čas a finance padnou na výrobu stavebnice, včetně možných slepých uliček, které budou výsledný čas a finanční náročnost ještě více zvyšovat.

Volba právě mikrokontroléru PICAXE vychází z jeho dobré dostupnosti, nízké ceny a minimálního dalšího příslušenství potřebného k nahrání programu do mikrokontroléru. Právě tyto podmínky se mi jeví jako důležité pro použití ve škole při výuce, ale ještě důležitější pro studenty navštěvující hodiny. Použití levného a dostupného mikrokontroléru při výuce, dává možnost zakoupit si mikrokontrolér včetně veškerého potřebného příslušenství pro nahrání programu a začít programovat a stavět zařízení i mimo školu.

Hlavní motivací při volbě tohoto tématu bylo moje rozhodnutí vyrobit v rámci práce projekt velmi praktického rázu, kde bych využil zkušenosti a vědomosti získané z mé praxe, které jsou bohužel ve škole rozvíjeny jenom minimálně. Vytvořené zařízení by mohlo být v budoucnosti použito při výuce, ale i při mých soukromých pokusech nejen s mikrokontroléry PICAXE.

Cíle práce

1. Cílem teoretické části práce je uvést základní informace o mikrokontrolérech PICAXE, ale i o jeho alternativách a nastítnit základní možnosti při konstrukci stavebnice.
2. Cílem praktické části je návrh a konstrukce vývojové desky s důrazem na univerzálnost a snadnou budoucí rozšiřitelnost externími vstupními a výstupními moduly nebo obvody, osazené některým mikrokontrolérem z rodiny PICAXE, použitelné pro výuku základů programování mikrokontrolérů.
3. Konstrukce základních externích vstupních a výstupních modulů připojitelných k vývojové desce demonstrující způsob jejich obsluhy mikrokontrolérem.
4. Vypracování úloh pro každý z modulů, které obsahuje jednoduchý ukázkový program, schéma propojení vývojové desky a modulů pro daný program, vývojový diagram programu a náměty na změnu nebo rozšíření programu pro samostatnou práci.

Teoretická část

Teoretická část obsahuje tyto části:

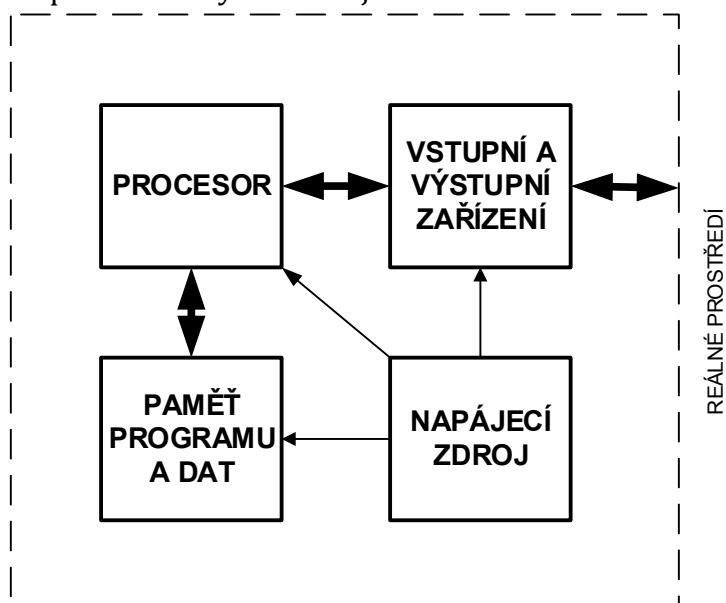
1. První kapitola obsahuje stručný obecný popis mikrokontroléru, navazuje základní popis dvou koncepcí: Harvardská a von Neumanova.
2. Druhá kapitola představuje základní informace o alternativách mikrokontroléru PICAXE, vhodných jako jeho náhrada či alternativa při konstrukci stavebnice.
3. Třetí kapitola obsahuje bližší seznámení s mikrokontroléry PICAXE.
4. Čtvrtá kapitola popisuje některé z možných způsobů konstrukce výukové stavebnice.

1 Mikropočítače a Mikrokontroléry

Název mikrokontrolér znamená totéž co monolitický mikropočítač a jednočipový mikropočítač (single chip microcomputer) [1]. V anglické literatuře se také objevuje zkratka **MCU** (Microcontroller unit), kterou budeme pro přehlednost v textu dále používat.

1.1 Co je mikrokontrolér?

Pro pochopení problematiky uveďme jednoduché blokové schéma počítače:



Obrázek 1 – Zjednodušené blokové schéma počítače (převzato z [2])

Všechny typy počítačů pracují na podobném principu a se stejnou strukturou z obrázku 1. Základ počítače tvoří procesor, který provádí posloupnou činnost zadanou programem. Procesor také vybírá jednotlivé instrukce z paměti, v níž je uložen právě vykonávaný program, dekóduje je a provádí odpovídající operace. Paměť programu a dat má v počítači zásadní roli, protože počítač bez paměti je pouze kombinačním obvodem. Vstupní a výstupní rozhraní jsou též podstatnou součástí, neboť bez nich není možné počítači zadat data ke zpracování a následně zpracovaná data z počítače získat. Může se jednat o klasickou klávesnici a myš jako vstupní zařízení a monitor nebo tiskárnu jako výstupní zařízení. Popřípadě u MCU analogový signál z potenciometru či logické hodnoty z tlačítka jako vstup a jako výstup PWM signál řídící rychlost motoru, nebo obyčejná LED dioda jako stavová kontrolka [2].

Velmi zjednodušeně řešeno, v počítači nalezneme procesor, paměť programu a dat, vstupní a výstupní obvody a napájecí zdroj. Pokud vynecháme blok napájecího zdroje, který je přítomný i u MCU, je počítač složen z těchto minimálně tří základních částí.

Když alespoň tyto tři části integrujeme na jeden jediný čip, vznikne tím mikropočítač zvaný mikrokontrolér. Vysokou úrovní integrace se dostaneme z rozměrů klasické ATX základní desky 305 x 244 mm potřebné k běhu stolního počítače, až na rozměry MCU s nejmenším pouzdem UDFN, které má osm vývodů, rozměry 2 x 2 x 0.6 mm a hmotnost pod 8 mg v případě ATtiny (Atmel) [3].

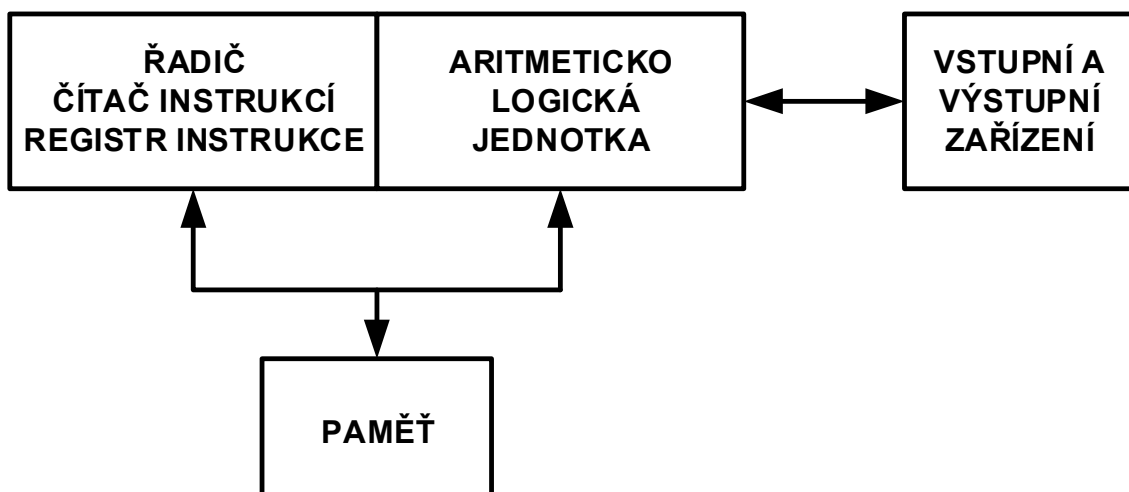
1.2 Koncepte MCU

Mezi dvě základní architektury patří koncepce von Neumannova a Harvardská. Koncepce jsou rozdílné v uspořádání paměťového prostoru pro program a data. Kapitola je zpracována podle Stanislava Pechala [2] a Pavla Tišnovského [4].

1.2.1 Koncepce podle von Neumanna

Používá jeden společný univerzální paměťový prostor pro data, tak i pro umístění programu. Není zde jednoznačně určeno, kde jsou v paměťovém prostoru uložena data a kde je uložen program. Rozložení velikosti paměťového prostoru pro program a pro data je čistě na tvůrci programu. Z toho vyplývá, že není předem známa přesná funkce programu, program se může během svého běhu měnit.

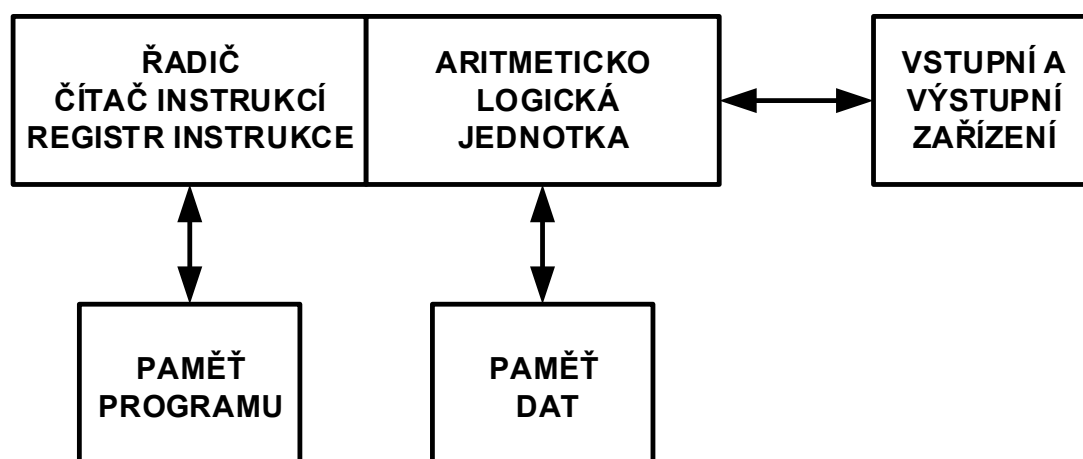
Dnes na von Neumannově koncepci pracují všechny osobní počítače. MCU z největší části pracují na koncepci druhé, tedy harvardské.



Obrázek 2 – von Neumannova koncepce (převzato z [2])

1.2.2 Harvardská koncepce

Jedinou odlišností harvardské koncepce od von Neumannovy je, že paměť určená pro data je oddělená od paměti určené pro program. Rozdělení pamětí vede k podstatnému zrychlení celého systému, protože koncepce předkládá dva různé, přesně určené a oddělené paměťové prostory, tedy možnost číst instrukce programu z paměti programu a současně zapisovat či číst data z paměti dat. Mezi výhody patří i možnost použití jiné technologie paměti programu. V současnosti nejčastěji EEPROM nebo flash.



Obrázek 3 – Harvardská koncepce (převzato z [2])

Rozdíl mezi koncepcemi je jasně patrný z obrázků 2 a 3.

Využití této koncepce u MCU vychází hlavně z jejich hlavního použití. Nejčastěji jako jednoduché řídicí prvky ve spotřební elektrotechnice, jednoúčelových zařízeních, hračkách a elektrických spotřebičích. Obecně se dnes najde jenom velice málo elektronických zařízení, které by v sobě nějaký MCU neměly. U těchto zařízení vývojář přesně zná, kolik bude mít zařízení funkcí a s jakými externími zařízeními a obvody bude pracovat. Podle toho se dá poměrně snadno zvolit standardní univerzální MCU z nabídky některého z výrobců, který bude pro požadovanou aplikaci splňovat všechna kritéria, jak po výkonové stránce, tak počtem a typem požadovaných vstupních a výstupních obvodů. Někteří výrobci jdou i cestou účelových MCU, navržených na základě architektury některého z univerzálních obvodů doplněného o periférie přesně pro danou aplikaci nebo zařízení.

Toto řešení by bylo u osobního počítače nepoužitelné, kdežto např. jako řídicí obvod mikrovlnné trouby stačí MCU v ceně řádově korun. Klasický osobní počítač by mikrovlnou troubu samozřejmě zvládl obsloužit také, ale celé řešení by bylo velmi komplikované a především mnohokrát dražší.

2 Programování vybraných druhů mikrokontrolérů.

Než se zaměříme na MPU PICAXE, uveďme alternativy, které jsou též primárně určeny jako výukové. Kapitola se bude přehledově zabývat open-source platformou Arduino (MCU řady ATmega(Atmel), mini počítačem Rapsberry Pi 2 (ARM Cortex-A7) a jeho GPIO.

Na začátek uveďme pár informací o zavaděči (angl. Bootloaderu).

2.1 Zavaděč

Každé MCU bez zavaděče se dá programovat přes programátor, v případě MCU PIC například vývojový programátor/debugger PICkit3. V případě druhého výrobce Atmel lze použít: Atmel AVRISP mkII In-System Programmer. Oba zmíněné programátory jsou oficiální zařízení, existuje i několik levnějších neoficiálních variant. Nutnost pořídit programátor však značně zneprůjemňuje začátky programování MCU. Hlavním důvodem je cena programátorů, ale také i fakt, že pro začátek nestačí pouze MCU, několik základních pasivních součástek a obyčejný propojovací kabel s některým z rozhraní počítače - ať už RS232 nebo USB.

Proto vznikají projekty, které by umožnily každému pořídit si jednoduchý vývojový prostředek s MCU, relativně levný a snadno programovatelný. Vzniká tedy velmi známá platforma Arduino založená na MCU řady ATMEGA (Atmel) a PICAXE používající MCU PIC (Microchip). Obě varianty jsou velmi snadno připojitelné a programovatelné počítačem za použití obyčejného propojovacího kabelu. Oba zástupci v sobě obsahují zavaděč nahraný v jejich paměti [5].

2.1.1 Bootloader Arduino

Je firmware nahraný ve flash paměti Arduina, umožňující nahrání programu do MCU bez použití programátoru. Zdrojový kód bootloaderu je volně dostupný. Komunikace s Arduinem probíhá přes USB, jednoduše tedy stačí Arduino propojit kabelem s počítačem a přes vývojové prostředí nahrát náš vlastní program [6].

2.1.2 PICAXE firmware

PICAXE obsahuje firmware, který je před programovaný v jeho paměti z výroby a umožňuje jednoduchou komunikaci po třech vodičích sériového portu PC. Odpadá

tedy nutnost použití drahého programátoru a vystačíme si s jednoduchým a levným tří vodičovým propojovacím kabelem s rozhraním RS232 [7].

2.2 Platforma Arduino

Jedná se o open-source výukovou platformu, která je založena na vývojové desce s MCU a jednoduchém vývojovém prostředí, které umožňuje tvorbu programů pro MCU.

Obrovskou výhodou je právě open-source, návrhy vývojových desek jsou vydávány pod licencí CC BY-SA 3.0 unported, zkušenější tedy mohou použít návrhy k tvorbě vlastních verzí desek. Každý si může svoji desku rozšířit nebo vylepšit. Možností, jak si pořídit Arduino je také více. Zkušební verzi lze postavit na kontaktním nepájivém poli nebo vlastnoručně vyrobit desku plošných spojů podle návrhu a osadit ji součástkami. Samozřejmostí je možnost koupit již hotový funkční modul z oficiální distribuce vyrobené v Itálii a podpořit koupí další vývoj Arduina. Další možností je využít nabídky většinou asijských výrobců nabízející moduly, zpravidla výrazně levnější. Začít tedy s programováním MCU nikdy nebylo jednodušší a začít může opravdu každý [5].

2.2.1 Moduly

Moduly jsou osazeny MCU řady ATmega (Atmel) až na jednu výjimku Arduino Due, které je osazeno 32 bitovým SAM3X8E ARM Cortex-M3 CPU (Atmel). Moduly se od sebe liší taktovací frekvencí MCU (od 8 MHz až po 84 MHz), počtem vstupních a výstupních obvodů, velikostí pamětí, ale i rozsahem napájecího napětí [8]. Většina z modulů komunikuje s počítačem přes USB port, jednou z možností je i odstranění bootloaderu a nahrávání programu přes externí programátor.

Konstrukce modulů je různá, liší se velikostí desky, ale hlavně způsobem jakým mají MCU vyvedené piny:

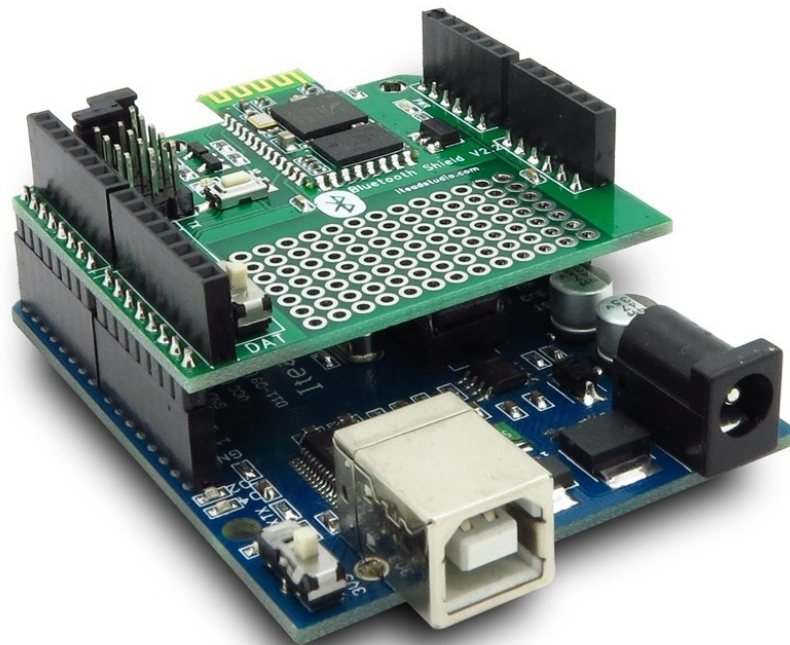
1. Konektory s piny např.: Arduino Micro viz obrázek 12.
2. Konektory s dutinkami např.: Arduino Uno viz obrázek 4.

Pro účely práce je zbytečné uvádět jejich detailní porovnání, z důvodu neustálého vývoje a různých revizí desek, kdy se jednotlivé parametry mohou nepatrně měnit. Proto bych rád čtenáře odkázal na oficiální internetové stránky. Pro zjištění aktuálních informací viz [8].

Zaměříme se spíše obecně na konstrukci shieldů, programovací jazyk a vývojové prostředí a příklad použití Arduina jako řídicí prvek 3D tiskárny.

2.2.2 Shiedly

Shiedly jsou ve světě Arduino rozšiřující vstupní a výstupní moduly, zasouvající se přímo na piny Arduina. Jak je patrné z obrázku 4, jako základní deska slouží Arduino a shiedly se zapojují na Arduino. Většina shieldů obsahuje průchozí konektory, které umožňují připojení dalšího shieldu na původní shield. Zpravidla tedy neexistuje omezení jedno Arduino jeden shield. Existuje však i menšina shieldů, které průchozí piny neobsahují a nelze na ně připojit již nic dalšího.



Obrázek 4 – Arduino Uno s Bluetooth shieldem (převzato z [9])

Nabídka shieldů je obrovská a vybere si opravdu každý. Např.:

1. Arduino Proto Shield – univerzální shield pro prototypy
2. Arduino Motor Shield – shield s obvodem L298 umožňující řízení 2 stejnosměrných motorů nebo jednoho krokového motoru.

A velké množství dalších oficiálních i neoficiálních shieldů viz [10].

2.2.3 Programovací jazyk

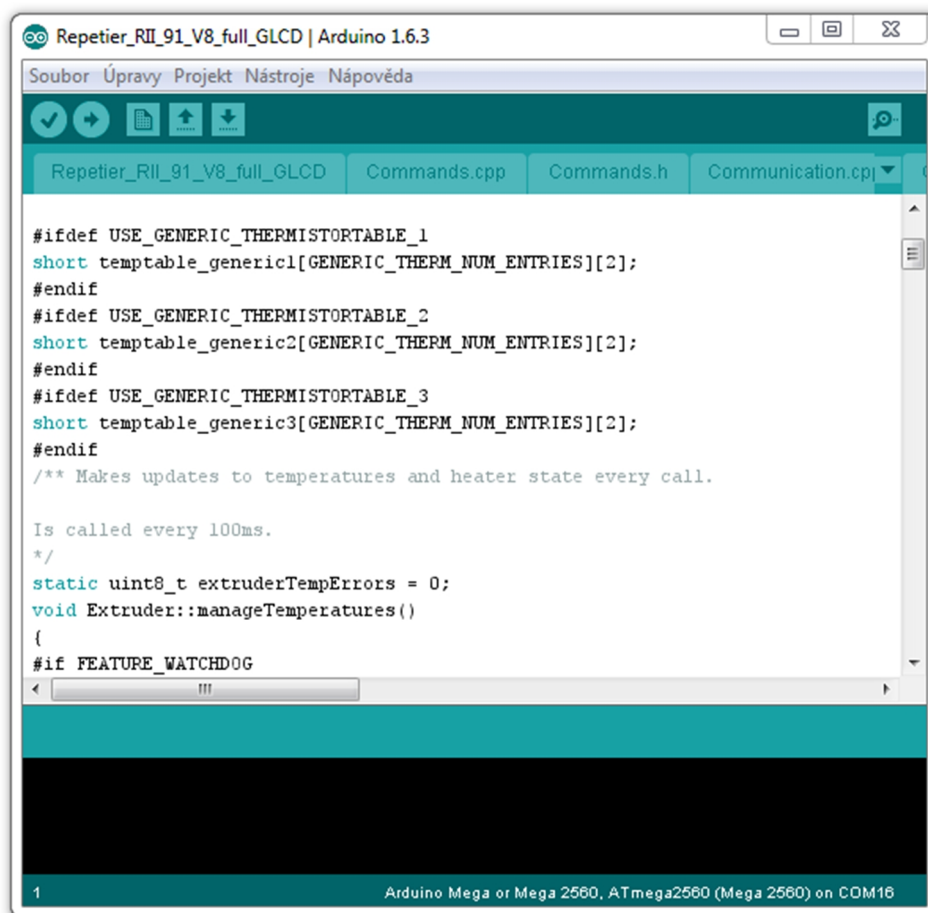
Programovací jazyk, který používá platforma Arduino se nazývá Wiring. Respektive nejedná se přímo o Wiring, ale jazyk který z něho vychází, přičemž rozdíly jsou minimální. Jazyk je velice podobný C++ [11].

Jazyk je také možné rozšířit knihovnami C++. Pro zkušené programátory lze přepnout na jazyk AVR-C, na kterém je Arduino založeno nebo vložit části programu napsaném v AVR-C do programu napsaném v jazyce Wiring [5].

Právě knihovny jsou mocným nástrojem pro usnadnění práce při programování. Dostupné jsou knihovny např. pro práci s LCD displejem, SD kartou, jednotlivými shieldy a mnoho dalších.

2.2.4 Vývojové prostředí

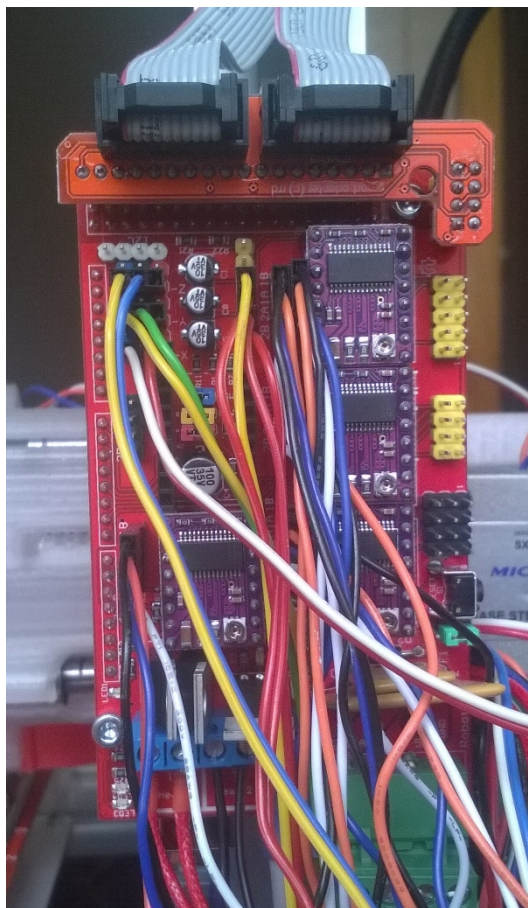
Jedná se o open-source vývojové prostředí, vytvořené v Javě, která zaručuje multiplatformnost. Vývojové prostředí je poměrně jednoduché a uživatelsky přátelské viz obrázek 5. Umožňuje tvorbu programů, nahrání programů do Arduina, ověření a kompilaci programů.



Obrázek 5 – Arduino IDE 1.6.3 s firmwre Repetier pro 3D tiskárnu Rebel2

2.2.5 Využití Arduina

Ačkoliv se původně jednalo o zařízení převážně pro výuku, Arduino si našlo tisíce fanoušků z veřejnosti. Můžeme vytvořit vlastní zařízení nebo se inspirovat četnými konstrukcemi jiných autorů. Zajímavou možností je použití Arduina jako řídicí jednotky v 3D tiskárně Rebel 2 a všeobecně možnost použití této kombinace ve všech open-hardware RepRap tiskárnách viz obrázek 6. Použití Arduina Mega 2560 vybaveného shieldem RAMPS v1.4, osazeného driverem krokových motorů Pololu DRV8825 k ovládání 5 krokových motorů, čtení G-kódu z SD karty, řízení teploty extruderu a vyhřívání podložky, navíc obsluhuje plně grafického displeje a to vše zcela plynule, dává dobrou představu o tom, jak je Arduino Mega 2560 univerzální a poměrně výkonné.



Obrázek 6 – Arduino Mega 2560 se shieldem RAMPS v1.4

2.3 Raspberry Pi

Raspberry Pi nepatří čistě mezi MCU. Jedná se o levný miniaturní počítač, přibližně o velikosti platební karty, osazený procesorem ARM. Je tedy schopný běhu některých z ARM GNU/Linux distribucí a je přislíbena i podpora nových Windows 10. Standardní a oficiální operační systém se nazývá Raspbian [12].

Poslední nový model Raspberry Pi 2 model B obsahuje:

- Čtyřjádrový procesor ARM Cortex-A7 CPU
- 1GB operační paměti RAM
- 4 USB porty
- Ethernet port
- HDMI výstup pro zobrazovací zařízení
- Kombinovaný 3,5 mm jack pro audio a kompozitní video
- Camera a Display interface (CSI a DSI)
- Grafické jádro VideoCore IV 3D
- Slot pro micro SD kartu na uložení operačního systému
- 40 GPIO pinů

Z výše vyjmenovaných parametrů je patrné, že s touto konfigurací lze Raspberry Pi použít jako jednoduchý počítač pro základní práci, multimediální centrum, NAS server a další nespočet aplikací, kde jsou výhodou malé rozměry, ale také nízká spotřeba. Tím hlavním a nejzajímavějším proč je Raspberry Pi zařazeno do této práce jsou GPIO piny.

2.3.1 GPIO piny

Nejnovější Raspberry Pi 2 jich celkem obsahuje 40. Piny se dají použít jako vstupně výstupní piny pro komunikaci s okolím např. pro připojení LED diod, tlačítek, čidel různých veličin a dalších zařízení. Jednotlivé piny se dají nastavit jako digitální vstupy nebo výstupy, je možná i UART, I2C a SPI komunikace. Hlavním rozdílem oproti Arduinu nebo PICAXE je použití 3,3 V logiky. Je tedy nutno dávat velký pozor, které zařízení připojíme, jelikož piny jsou přímo vyvedeny na procesor bez žádných dalších ochran. Připojení 5 V signálu k pinu znamená okamžité zničení procesoru! Je také nutno počítat s velmi malým proudovým zatížením pinů - maximálně 10 mA [13].

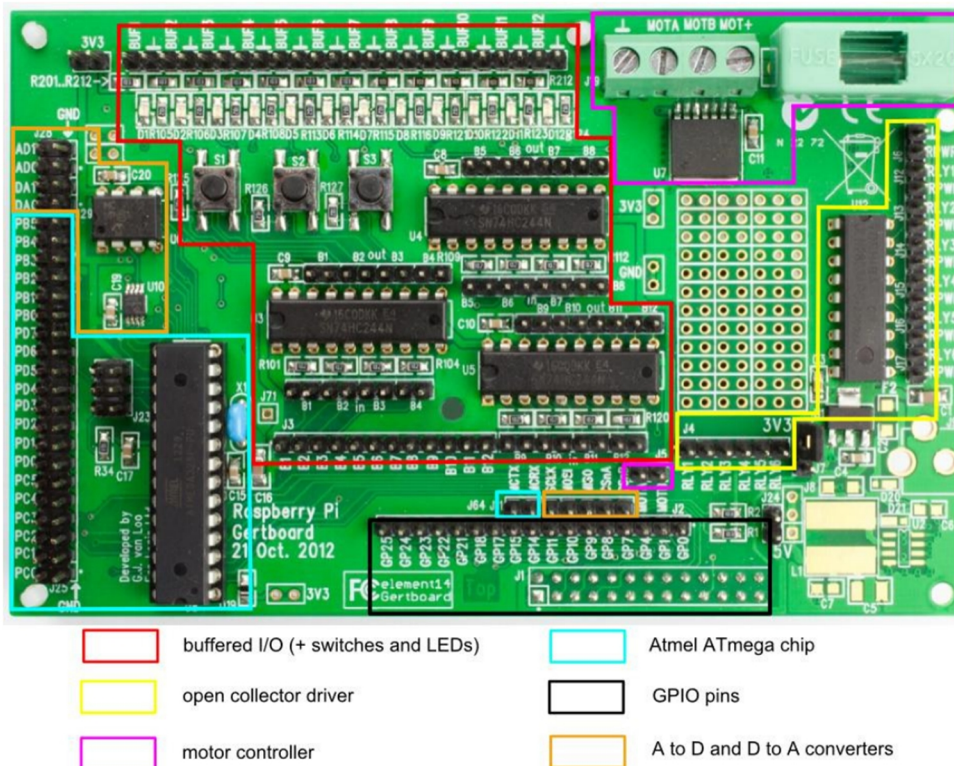
Z předešlého odstavce vyplývá, že tyto piny jsou poměrně choulostivé na použití. V případě použití Raspberry Pi jako stavebnice, je vhodné uvažovat o některé z rozšiřujících desek pro GPIO piny. Desek je k dispozici několik, od jednoúčelových jako je např. teploměr, reléová karta a další, až po univerzální desku zaměřenou na všeobecné pokusy nebo desku zaměřenou na automatizaci.

Představme velice krátce univerzální desku Gertboard a desku zaměřenou na automatizaci UniPi.

2.3.2 Gertboard

Gertboard je rozšiřující deska pro GPIO piny. Je velice univerzálně zaměřena. Uvedme základní funkční bloky z obrázku 7 [14]:

- 12 vstupně výstupních pinů
- 3 tlačítka
- 6 výstupů s otevřeným kolektorem
- H můstek pro ovládání motorů
- 28 pinový MCU ATmega
- 2 kanálový 8,10 nebo 12 bitový D/A převodník
- 2 kanálový 10 bitový A/D převodník



Obrázek 7 – Gertboard (převzato z [14])

Z konstrukce desky je patrné, že je řešena jako naprosto univerzální, lze tedy realizovat velké množství různých zapojení a pokusů. Bonusem je volně dostupné schéma desky i návrh plošného spoje.

2.3.3 UniPi

UniPi viz obrázek 8 je rozšiřující deska zaměřená spíše na automatizaci. S touto deskou lze Raspberry Pi proměnit na PLC, ale nic nebrání tomu desku používat i jako součást stavebnice pro pokusy.

Uvedme základní parametry desky [15]:

- 8 přepínacích relé pro ovládání připojených zařízení (250V AC/5A nebo 24V DC/5A)
- 14 opticky izolovaných digitálních vstupů pro napětí 5-24V
- 2 analogové vstupy 0-10V
- 1 analogový výstup 0-10V
- 1-Wire port pro připojení čidel teploty a vlhkosti
- I2C rozšiřující port
- UART port

Deska je zaměřena více na automatizaci např. objektů nebo automatizaci vytápění, ale zajisté by našla uplatnění i pro konstrukci stavebnice. Jedinou nevýhodou je její poměrně vysoká cena viz [15].

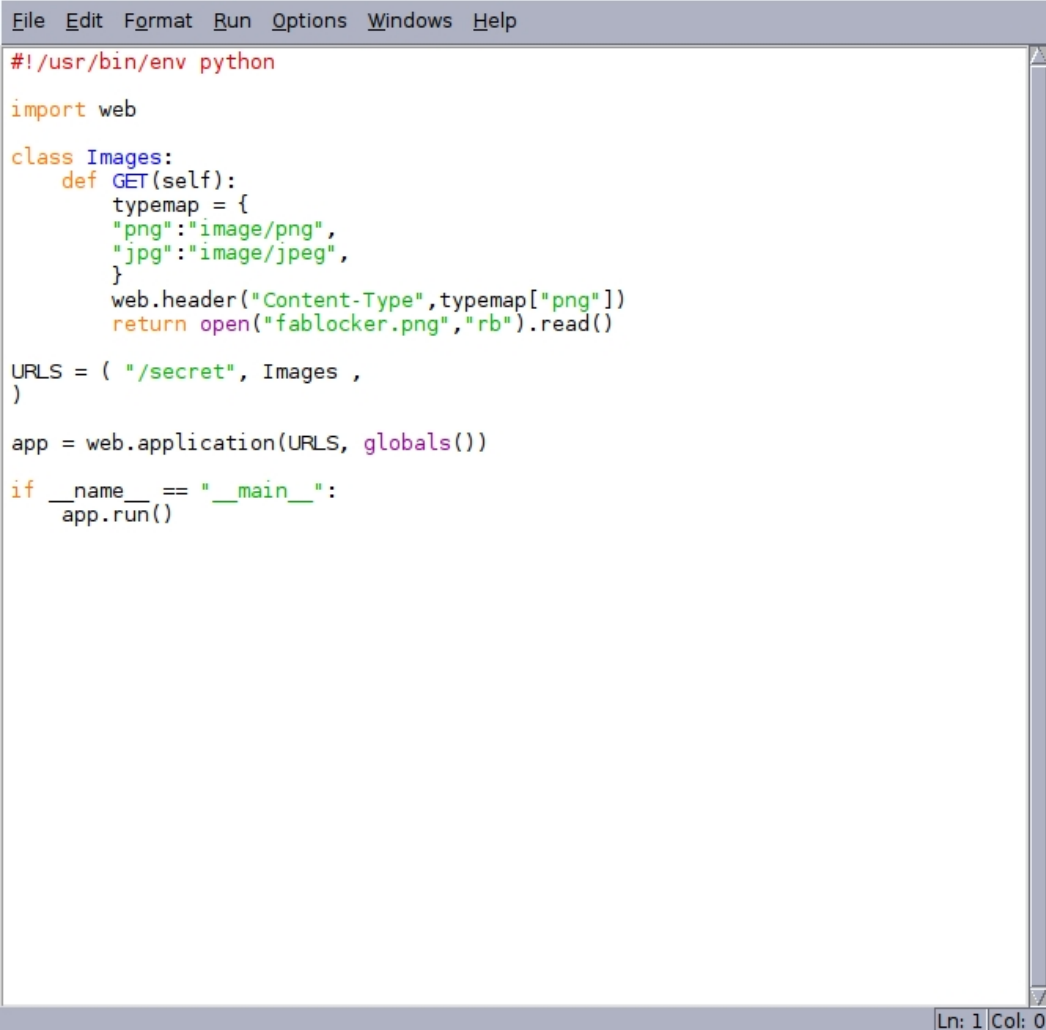


Obrázek 8 – UniPi (převzato z [15])

2.3.4 Programovací jazyk a vývojové prostředí

Raspberry Pi využívá programovací jazyk Python, který je zaměřen na dobrou čitelnost a využívá standardní anglická klíčová slova. Aktuálně Raspberry Pi umožňuje psát programy ve dvou verzích jazyka - Python 2 a Python 3.

Vývojové prostředí je přímo integrováno v operačním systému Raspbian a nazývá se IDLE. Vývojové prostředí je opět velmi jednoduché a přehledné viz obrázek 9 [16].



```
File Edit Format Run Options Windows Help
#!/usr/bin/env python

import web

class Images:
    def GET(self):
        typemap = {
            "png": "image/png",
            "jpg": "image/jpeg",
        }
        web.header("Content-Type", typemap["png"])
        return open("fablocker.png", "rb").read()

URLS = ( "/secret", Images ,
)

app = web.application(URLS, globals())

if __name__ == "__main__":
    app.run()

Ln: 1 Col: 0
```

Obrázek 9 – Prostředí IDLE (převzato z [27])

3 Mikrokontroléry PICAXE

Zaměříme se na MCU PICAXE, zvláště na typ 20M2, který bude součástí stavebnice. V kapitole se blíže seznámíme s PICAXE, uvedeme přehled jednotlivých typů, které se od sebe liší nejen počtem pinů, možnosti programování PICAXE a základní informace o programu PICAXE editor 6. Kapitola čerpá z oficiálních manuálů [17], [18], [19] a oficiálních internetových stránek [7].

3.1 Představení PICAXE

PICAXE podobně jako obě zmíněné alternativy v kapitole 2 bylo primárně vyvinuto a určeno pro výuku. Postupem času se však díky své jednoduchosti a nízké ceně stalo velmi oblíbené širokou veřejností. PICAXE je kromě výuky velice často používáno i jedinci kteří pro svoje zařízení potřebují jednoduchý, levný, snadno programovatelný MCU a nechtějí nebo nemají čas se hlouběji zabývat ať už zapojením, nebo programovacím jazykem obvykle používaným u běžných MCU. Místo obvyklého assembleru či jazyků C využívá lehce naučitelný PICAXE BASIC a obsahuje velké množství speciálních instrukcí pro práci s většinou běžně používaných vstupních a výstupních zařízení, kdy jeden příkaz zastoupí i několik desítek řádků kódu. Všechny tyto skutečnosti umožňují i začátečníkovi ve velmi krátké době vytvořit funkční program. Nízká cena a minimální další potřebné příslušenství dává možnost studentům zakoupit si PICAXE domů, provádět vlastní pokusy a při tom se neustále zdokonalovat mimo výuku.

Všechny MCU PICAXE jsou založeny na MCU PIC a z výroby v sobě obsahují firmware, který umožňuje programování přes sériový port SR232. Firmware také umožňuje programování přes jednoduchý, volně dostupný vývojový program, který mimo jiné obsahuje simulátor, umožňující otestování programu i bez MCU.

3.2 Porovnání typů

Existuje několik typů, přičemž hlavní a jasně viditelný rozdíl je v počtu pinů. K dispozici jsou obvody s 8, 14, 18, 20, 28 a 40 piny. Téměř všechny piny jdou nastavit jako digitální vstupy nebo výstupy, některé piny mohou být konfigurovány jako analogové vstupy či je využít jako vstupy pro dotyková tlačítka. Tím ale všechny možnosti nekončí. PICAXE si dokáže poradit i s pokročilejšími funkcemi jako je I2C, SPI a 1-Wire pro pokročilejší použití.

Při bližším pohledu jsou MCU PICAXE vyráběné ve dvou základních řadách. Řada M a řada X.

3.2.1 Řada X

V pokročilé řadě nalezneme 3 modely (20x2, 28x2 a 40x2). Nabízí tedy větší počet pinů než řada M. Řada X se liší od řady M rozšířenou podporou matematických a dalších pokročilých funkcí. Kromě interního oscilátoru umožňují připojení externího krystalového oscilátoru, lze je tedy provozovat až na maximální taktovací frekvenci 64MHz. Přehledová tabulka základních vlastností všech typů je umístěná v příloze B na obrázku 62.

3.2.2 Řada M

Základní řada MCU PICAXE nabízí aktuálně 4 modely. Je omezená o některé pokročilejší příkazy. Přehledová tabulka se základními rozdíly je umístěna v příloze A na obrázku 61. Zaměřme se na typ 20M2 použitý ve stavebnici.

3.3 PICAXE 20M2

Je dvaceti pinový MCU založený na PIC16F1829 doplněný o firmware. Firmware kromě kódu starajícího se o komunikaci s počítačem a vývojovým prostředím obsahuje i další části, v kterých jsou obsaženy speciální instrukce. Některé jsou popsány v praktické části práce.

Vyberme některé ze základních parametrů, které jsou nejdůležitější pro další práci. Kvůli rozsahu práce není možné zmínit vše a není to ani cílem. Kompletní informace jsou uvedeny v manuálech zmíněných na začátku kapitoly 3.

3.3.1 Napájecí napětí

Tento typ je možno napájet napětím v rozsahu 1,8 – 5,5 V. Vrchní hranice by měla být brána jako limitní a neměla by se tedy překračovat. Každý z pinů umožňuje zatížení až 20 mA.

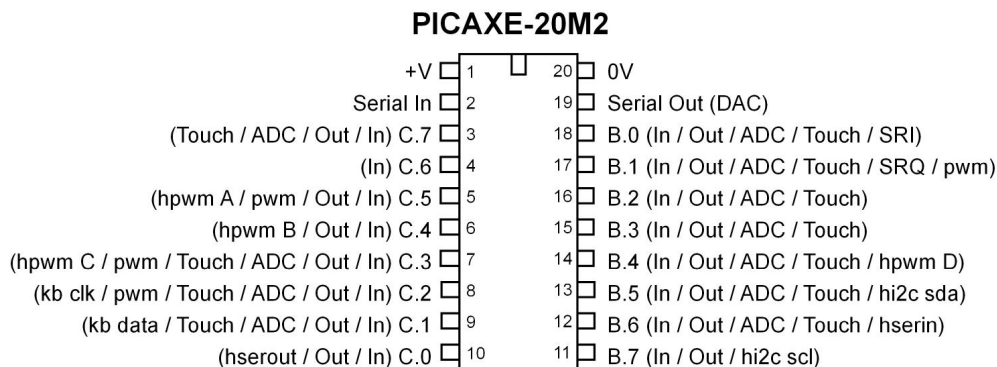
3.3.2 Frekvence

MCU disponuje vnitřním oscilátorem s maximální frekvencí 32 MHz. Frekvence jde velice snadno měnit jedním příkazem přímo v běžícím programu v krocích 4, 8, 16 a 32 MHz. Základní defaultní takt MCU je 4 MHz. Zvýšením frekvence dojde ke zrychlení programu, ale i zkrácení doby příkazů - např. pause. Některé speciální příkazy pro svou správnou funkci vyžadují přímo takt 4 MHz. V případě, že

používáme vyšší takt, je možné před vykonáním speciálního příkazu takt snížit na 4 MHz a po vykonání opět zvýšit na původní hodnotu.

3.3.3 Piny a pinout

Velkou pomocí při práci s MCU je pinout na obrázku 10.



Obrázek 10 – Pinout MCU PICAXE 20M2 [převzato z [17]]

Jasně a přehledně zobrazuje piny MCU včetně jejich názvu, s popsanou funkcí jednotlivých pinů v závorce. Piny označené v pouzdře čísly 1, 2, 19 a 20 jsou vyhrazeny pro napájení a komunikaci s počítačem přes programovací kabel. Pin 19 (serial out) je sdílen s D/A převodníkem.

Z obrázku 10 vyplývá, že tento typ obsahuje dvě 8 bitové vstupně výstupní brány B a C. Všechny piny obou bran lze individuálně nastavit jako vstupní či výstupní, kromě pinu C.6, který je pouze vstupní. Obě brány mimo pin C.6 jsou sdíleny s dalšími vnitřními perifériemi. Jasně tedy vyplývá, že lze použít až 11 pinů jako vstup analogového signálu pro A/D převodník. Tyto piny jsou sdíleny i pro vyhodnocování dotekových tlačítek (TOUCH). Celkem 4 piny jsou určeny pro generování PWM signálu, a dále je většina pinů sdílena s dalšími perifériemi pro pokročilejší funkce, kterými se práce nebude zabývat.

3.4 Organizace paměti

Paměť v MCU je rozdělena na 3 jednotlivé zóny. Velikost paměti se liší podle typu MCU. Následující podkapitoly platí pro typ 20M2.

3.4.1 Paměť programu

Paměť programu slouží pro uložení programu po naprogramování. Tato přepisovatelná paměť má výrobcem zaručeno minimálně 100 000 přepsání. Paměť je nevolatilní. V případě ztráty napájecího napětí program zůstává uložen a spustí se od začátku. Paměť před nahráváním programu nemusí být nejdříve vymazána. Nové

nahrání programu tuto paměť zcela vymaže a nahraje program nový. Samozřejmě je možné paměť pouze vymazat bez nahrání nového programu. Kapacita paměti je 2048 bajtů.

3.4.2 Paměť dat

Je úplně oddělena od paměti programu a je také nevolatilní. Pro zápis a čtení paměti jsou určeny speciální instrukce EEPROM, READ a WRITE. Její kapacita je 255 bajtů. V případě nahrání nového programu jsou paměťové buňky resetovány na hodnotu 0.

3.4.3 Paměť RAM (proměnné)

Paměť RAM je určena pro dočasné uložení hodnot proměnných během běhu programu. Paměť je volatilní, maže se i v případě resetu. Kapacita paměti je 512 bajtů. Paměť RAM je dále dělena na 3 části:

1. **General purpose variables** je celkem 28 bajtových proměnných označených b0 až b27 pro uložení kladných čísel 0 až 255. Při pokusu zapsat větší nebo záporné číslo paměť přeteče bez varování - např. $254+3=1$ nebo $2-3 = 255$. Pro uložení čísel v rozsahu 0 až 65535 slouží proměnné w0 – w13 a jsou spojeny ze dvou bajtových proměnných b např. $w2=b1:b0, w3=b3:b2$ atd. Nejvýznamnější bajt w0 je b1, nejméně významný bajt w0 je b0.
2. **Storage variables** jsou proměnné pro dočasné uchování bajtových čísel. Celkem lze uložit až 48 čísel. Proměnné jsou dostupné pomocí příkazů peek a poke. Tato paměť není dále v práci použita.
3. **Special function variables (SFR)** jsou speciální proměnné pro nastavení některých parametrů. Vyberme pouze dvě nejdůležitější dirsB a dirsC. Obě proměnné jsou určeny pro nastavení směru brán B a C mezi vstupní a výstupní. Obě proměnné je možno rozdělit na jednotlivé bity a lze nastavovat jednotlivé piny obou bran jako výstup (1) nebo vstup (0). Pro bránu B platí: $dirsB = dirB.7 : dirB.6 : dirB.5 : dirB.4 : dirB.3 : dirB.2 : dirB.1 : dirB.0$ konkrétní zápis: $dirsB = \%11000000$ (piny B7 a B6 nastaví jako výstup, zbytek pinů brány B jako vstup).

3.5 Programování PICAXE

Primárním programovacím jazykem je PICAXE BASIC, který je dále využit v praktické části. Jazyk je velice přehledný a uživatelsky přívětivý. Obsahuje již zmíněné velké množství speciálních instrukcí pro práci s většinou obvyklých vstupních a výstupních zařízení.

Existuje ale více způsobů. Kromě klasického zápisu programu je jednou z možností i grafické programování pomocí vývojových diagramů (flowcharts).

Pro použití při výuce nejmladších se nabízí programování v offline verzi programu Scratch. Ve Scratchi lze vytvořit program a nahrát do PICAXE. Další možností je interakce mezi PICAXE a programem Scratch např. při stisku tlačítka kočka udělá krok, podle polohy potenciometru jde kočka směrem doleva nebo doprava a mnoho dalších variant, které mohou pomoci výuku nejmenších velice ozvláštnit.

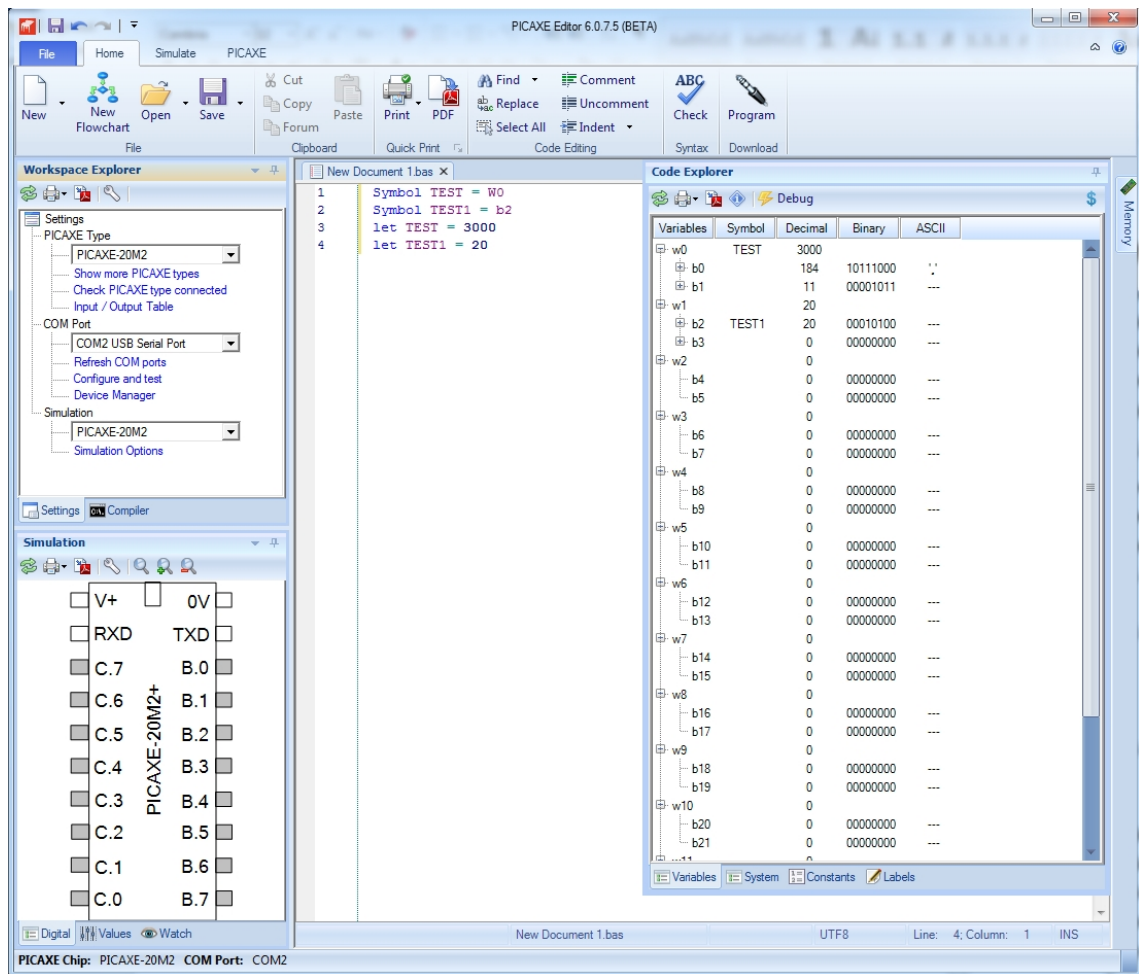
3.6 Vývojové prostředí

Vývojové prostředí zvolené pro naši práci je PICAXE Editor 6 viz obrázek 11, který je dostupný pro operační systém Windows. Je volně dostupný pro vzdělávací, soukromé, ale i komerční použití. Program je kompatibilní se všemi typy MCU PICAXE.

Program kromě samotné tvorby a nahrávání programů poskytuje velké množství dalších funkcí:

1. Kontrola syntaxe a nahrávání programu do MCU.
2. Code explorer umožňující zobrazení hodnot proměnných a konstant.
3. Simulátor programů.
4. Podpora ladění programu (debug) včetně breakpointů.
5. Integrovaný sériový terminál.
6. Kalibrační nástroje např. pro doteková tlačítka a další.
7. Generátory kódu a aplikace pro konfiguraci (Wizards).
8. Možnost grafického programování pomocí vývojových diagramů (flowchart).

Vývojové prostředí je přehledné a intuitivní. Obsahuje všechny potřebné funkce usnadňující práci při vývoji programů. Nevýhodou je podpora pouze operačního systému Windows. Pro Linux nebo Mac OS je určený program AXE pad.



Obrázek 11 – PICAXE Editor 6

4 Mikropočítačové vývojové stavebnice

Při konstrukci výukové stavebnice můžeme zvolit několik řešení, popřípadě jejich kombinaci. Každé z následujících řešení v sobě ukrývá určité výhody, ale i nevýhody. Výběr z největší části omezuje cena, ale také naše technické a materiální vybavení a zázemí, které máme dostupné ve škole, na zájmovém kroužku nebo kdekoli jinde, kde plánujeme stavebnici vyrábět nebo používat.

4.1 Hotová stavebnice

Hotové řešení je nejrychlejší a nejsnadnější možný způsob, jak pořídit funkční zařízení na pokusy nebo výuku.

Možnými řešeními jsou např.:

1. Platforma Arduino a některý ze shieldů podle plánovaného zaměření stavebnice [10].
2. Mini počítač Raspberry Pi s některou rozšiřujících vývojových desek popsaných v kapitole 2. Možností je použití i jednoúčelových rozšiřujících desek podle zaměření stavebnice.
3. MERKUR TOYS je tradiční český výrobce stavebnic, který svou nabídku rozšířil na rozsáhlé možnosti robotických a mechatronických stavebnic. Stavebnice mají na výběr dvě možnosti ovládacích desek s MCU PICAXE a MCU Atmel [20].
4. V nabídce PICAXE jsou kromě samostatných mikrokontrolérů i hotové vývojové a výukové desky v podobě startovacích balíčků, projektových desek, přídatných modulů a dalších zařízení bezproblémově fungujících s PICAXE [21].
5. Nespočet výukových stavebnic jako lego mindstorms a desítky dalších.

Hlavní výhodou je záruka na stavebnici, manuál a většinou i technická podpora, která ovšem nemusí být v českém jazyce. Mezi nevýhody může patřit určitá neuniverzálnost, která vychází z konstrukcí stavebnic, což platí hlavně u stavebnic firmy MERKUR TOYS. Mezi další nevýhody patří vyšší cena a také poměrně malá dostupnost v Čechách, zvláště u vývojových a projektových desek PICAXE.

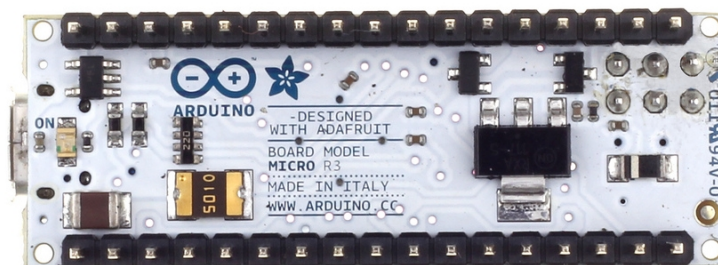
4.2 Nepájivé kontaktní pole

Vytvoření stavebnice na kontaktním poli je rychlé, levné a až na některé výjimky není potřeba žádné pájení a zaručuje maximální univerzálnost.

PICAXE je k tomuto způsobu velmi vhodná díky patci DIP. Obvod jde přímo vložit do kontaktního pole bez dalších úprav.

Při realizaci stavebnice s Arduinem je vhodné zvolit Nano, Micro nebo jiné verze, které mají výstupem kolíky viz obrázek 12. Samozřejmě lze použít všechny verze, ale bude nutné mezi Arduinem a kontaktním polem vytvořit drátové propojky, které ale přispívají k větší nepřehlednosti.

V případě Raspberry Pi bude pole velkým pomocníkem při realizaci vstupních a výstupních obvodů.



Obrázek 12 – Arduino Micro (převzato z [26])

Kontaktní nepájivé pole má však celou řadu nevýhod. Vlivem častého používání se kontakty v poli vymačkají, nebo vyhřejí při průtoku většího proudu. Tím dojde ke zhoršení kontaktu a spoj je nevodivý úplně, nebo v horším případě vede a nevede nahodile. S tím souvisí poměrně velká časová náročnost zapojení při použití ve výuce – student má vše zapojeno správně, vše je funkční a jediný problém je v kontaktním poli, kde kontakt chvíli vede a chvíli ne.

Časté vytahování a zasouvání obvodů a součástek z pole také vede k ulomení pinů nebo vývodů. Za zmínku ještě stojí velká nepřehlednost při realizaci složitějších zapojení.

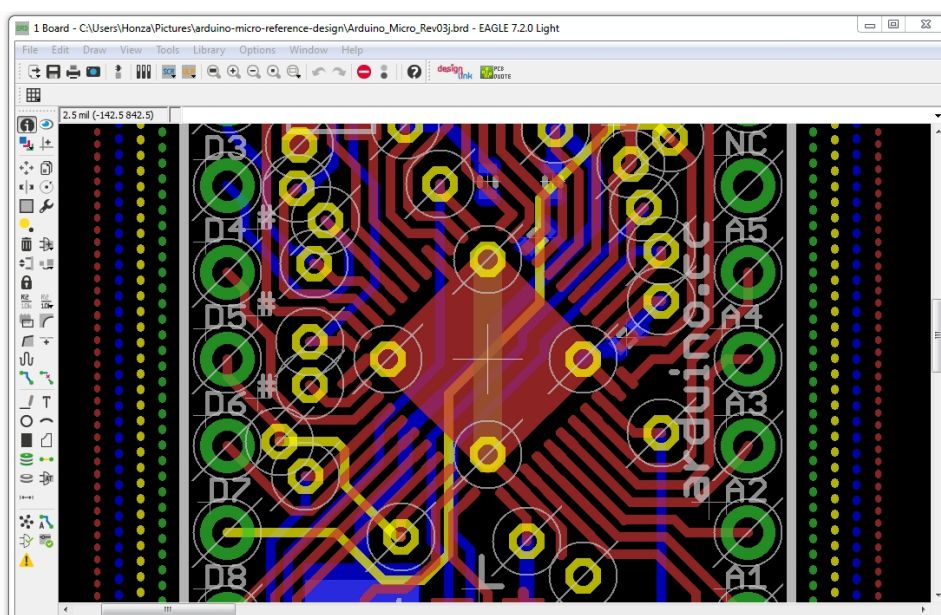
Z těchto důvodů je kontaktní pole vhodné spíše jen na odzkoušení prototypů a je vhodné alespoň MCU umístit na vhodně navrženou desku plošných spojů k eliminaci poškození vývodů a kontaktní pole používat pouze na zapojení vstupních a výstupních obvodů.

4.3 Návrh plošného spoje

Vlastní návrh plošného spoje a jeho výroba je nejlepší cestou, jak vytvořit profesionálně vypadající výrobek a autorovi dává volnou ruku k maximální individualitě desky.

K návrhu desky je vhodné použít SW k tomu přímo určený. Asi nejznámější je program EAGLE viz obrázek 13. Program disponuje rozsáhlými knihovnami součástek, jejich pouzder všech možných druhů a je vhodný i pro kreslení schémat. Ve většině případů postačí verze light, která je freeware pro nevýdělečné aplikace a studijní účely. Jistým omezením může být maximální rozměr desky 100x80 mm. Alternativou může být DesignSpark PCB, je také volně dostupný i pro výdělečné aplikace, je uživatelsky přívětivější než EAGLE. Též disponuje poměrně dobře vybavenými knihovnami.

Vlastní návrh však vyžaduje od tvůrce desky znalosti v tomto odvětví. V případě použití SW EAGLE bez předchozích zkušeností je nezanedbatelný i čas, než se tvůrce seznámí s nepříliš intuitivním prostředím, ale hlavně vyžaduje dokonalé promyšlení celé desky předem. Na již hotové desce plošných spojů nelze provádět žádné další úpravy, přidávání obvodů nebo i změnu typu konektoru s jiným pouzdem, který se ukázal v praxi nevhodný pro danou aplikaci. V tomto případě bude nutné znovu zasednout k danému SW, provést změny a desku vyrobit znovu, což v případě zadání výroby odborné firmě může být někdy velice zdlouhavé a v sérii pouze několika kusů i finančně náročné.



Obrázek 13 – program EAGLE light a část návrhu desky Arduino micro

4.4 Univerzální plošné spoje

Univerzální plošné spoje jsou běžně dostupné ve specializovaných obchodech s elektrotechnickými součástkami nebo modelářských obchodech. Plošné spoje se liší hlavně rozměry, ale i dalšími vlastnostmi. Lze pořídit spoje jednostranné i oboustranné, mohou být vrtané i nevrtané. Mimo desek pro klasickou montáž součástek THT, lze pořídit plošné spoje i pro povrchovou montáž součástek SMD či smíšené desky umožňující kombinaci obou technologií viz aktuální nabídka firmy GES electronics [22].

Výsledný výrobek sice nevypadá profesionálně, ale v případě výukové stavebnice, kterou chceme provozovat na škole, nebo na zájmovém kroužku plně dostačuje a právě zde využijeme výhod, které tento způsob konstrukce poskytuje.

Hlavní výhodou je, že práce na desce může začít okamžitě po zakoupení, není potřeba všechna zapojení včetně použitých konektorů a dalších technických detailů mít přesně promyšlené předem, odpadne i čas strávený návrhem a výrobou plošného spoje. Umožňuje tedy desku vyvíjet a rozšiřovat za běhu, ovšem je nutné mít předem základní představu o prostorovém rozmístění komponent umístěných na desce z důvodu omezeného prostoru desky. Případná drobná či větší pozdější změna na desce není příliš velkým problémem, pokud si necháme dostatečný prostor kolem komponent navíc.

Propojování jednotlivých součástek na desce (vytvoření vodivých cestiček mezi jednotlivými součástkami) lze realizovat vytvářením cestiček pájky mezi jednotlivými body nebo ploškami pole. Využít jdou i dlouhé vývody jednotlivých součástek, nebo cestičky tvořit drátěnými propojkami z izolovaných a neizolovaných drátů vhodných průřezů. Dráty mohou být bez izolace, v případě kdy není nutné spoje křížit, v případě křížení drátů je vhodnější použít dráty s izolací alespoň na jeden křížený spoj.

Doposud kapitola zmiňuje jen možnosti konstrukce desky s MCU. Všechny uvedené způsoby popřípadě jejich kombinace lze použít i k tvorbě přídatných modulů se vstupními a výstupními obvody.

Kapitola určitě nezmiňuje všechny možné způsoby konstrukce stavebnice. Vybral jsem pouze dostupné a reálně proveditelné ve školním prostředí. Každého čtenáře jistě napadne mnoho dalších způsobů konstrukce.

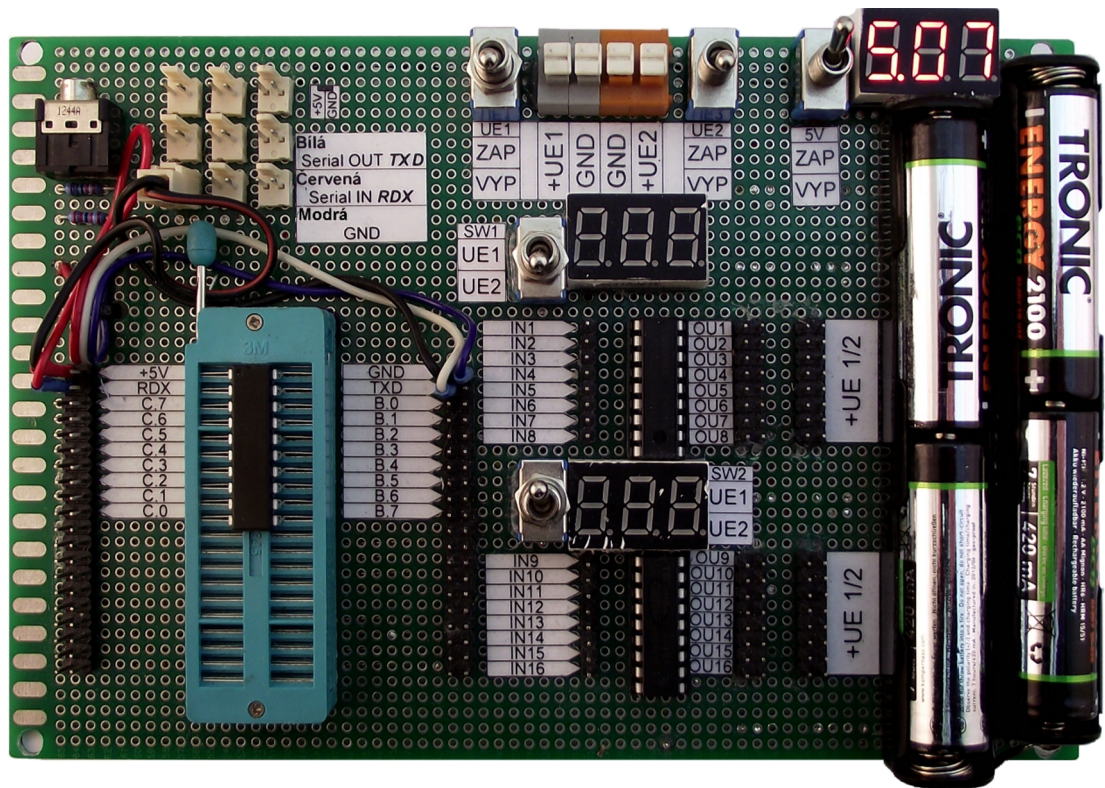
Praktická část

Praktická část se věnuje konstrukci vlastní výukové stavebnice, vývojové desky pro MCU PICAXE osazené typem 20M2 a konstrukcí vstupních a výstupních modulů. U vlastní konstrukce modulu je uvedeno schéma zapojení modulu, v případě hotových zakoupených modulů pouze jejich popis.

Nedílnou součástí praktické části jsou ukázkové úlohy, obsahující funkční ukázkové programy pro MCU, vývojové diagramy programů, schémata propojení modulů a náměty na rozšíření a úpravu ukázkových programů.

5 Konstrukce výukové stavebnice

5.1 Vývojová deska



Obrázek 14 – Vývojová deska

Vývojová deska je zachycena na obrázku 14 a je postavena na oboustranném univerzální plošném spoji o rozměrech 18x12cm. Pro účely této práce je deska osazena MCU PICAXE 20M2.

V následujících podkapitolách popíšeme jednotlivé části vývojové desky. Kompletní schéma nalezneme na obrázku 15 na konci kapitoly.

5.1.1 MCU část

Skládá se pouze z tří součástí. Patice pro MCU byla zvolena 40 pinová testovací patice ZX1. Všechny piny patice jsou vyvedeny na 2 dvouřadé kontaktní lišty SV3 a SV4. Návrh tedy umožňuje osadit vývojovou desku jakýmkoliv typem MCU PICAXE v standardních pouzdrech DIP8 až DIP40. Patice také zaručuje velmi rychlou výměnu MCU bez použití jakéhokoliv nářadí nebo přípravků a díky nulové zasouvací síle nehrozí poškození pinů MCU ani při velmi časté manipulaci.

5.1.2 Komunikační obvod

Komunikační obvod plně odpovídá zapojení uvedeném v oficiálním manuálu. Obvod se skládá pouze ze tří součástí – rezistoru R2 a R1 a konektoru X12 Jack 3,5 mm

stereo. Ačkoliv je konektor velice neobvyklý pro přenos dat, protože jeho obvyklé použití je spíše v audiotechnice, byl použit pro zachování kompatibility s oficiálním programovacím kabelem AXE027 USB cable. Vývody pro připojení na MCU jsou řešeny jako 3 vodiče s dutinkami připojitelné na kontaktní lištu. Řešení zaručuje kompatibilitu se všemi MCU PICAXE. Avšak při změně typu MCU musíme pamatovat na skutečnost, že piny serial in, serial out a GND (0 V) nejsou vždy stejně umístěny, proto je důležité přepojit tyto tři vodiče podle pinoutu daného MCU.

Upozornění: Tato část musí být připojená k MCU pro správnou funkci vždy.

5.1.3 Napájecí část 5 V

Zdrojem 5 V pro napájení MCU a některých modulů jsou 4 tužkové Ni-MH akumulátory umístěné v držácích. Nabíjení akumulátorů je řešeno v externí nabíječce. Napětí je přes páčkový přepínač S1 a keramický kondenzátor C1 vedeno na celkem 9 (X2 až X10) dvou pinových napájecích konektorů se zámkem pro omezení možnosti přepólování. Součástí držáků akumulátorů je miniaturní voltmetr s LED displejem, plní funkci kontrolky zapnutí napájení a také poskytuje informaci o úrovni nabití akumulátorů.

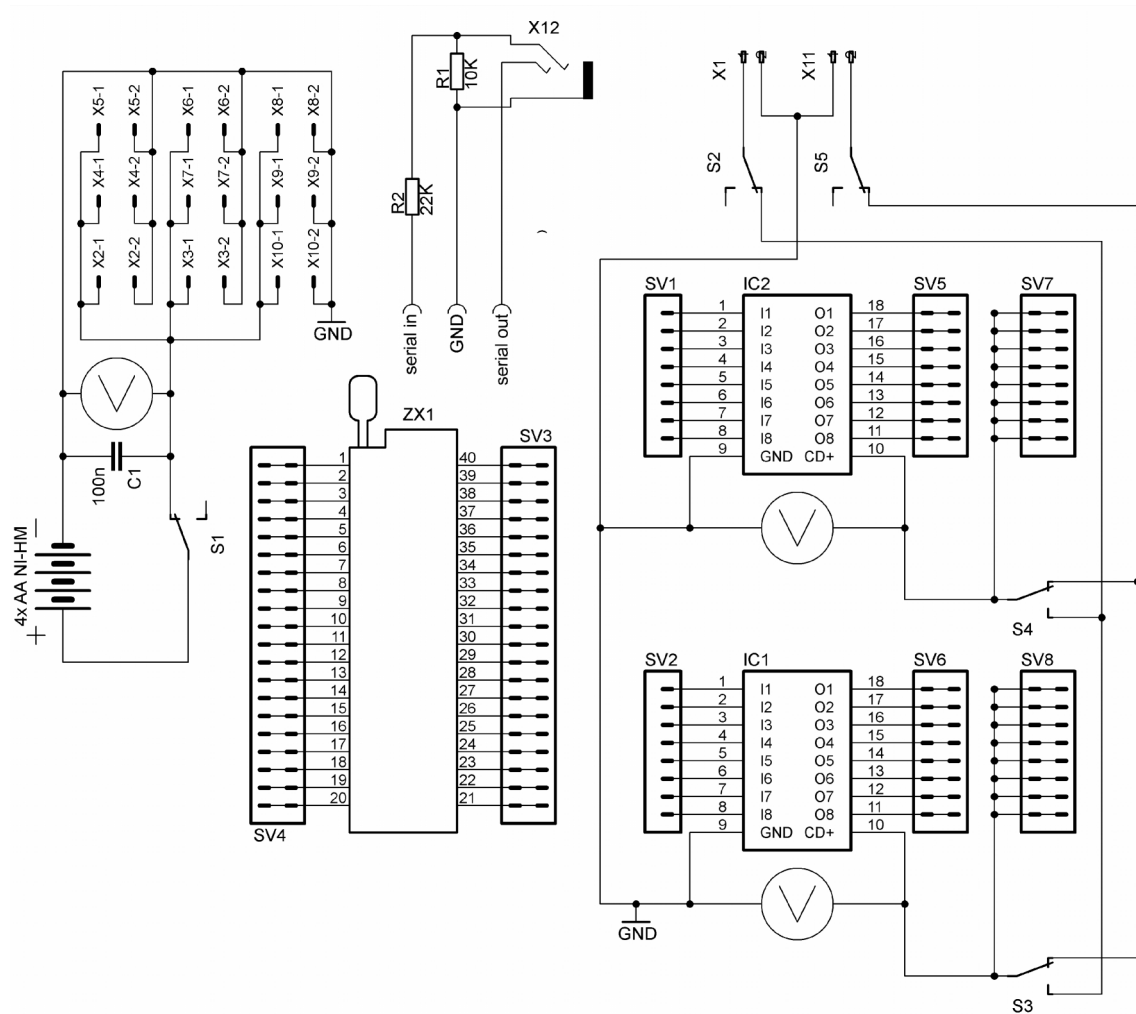
Na jeden z konektorů je dle pinoutu daného typu připojen i samotný MCU pomocí propojovacího kabelu. Opět z důvodů kompatibility se všemi typy MCU PICAXE.

Upozornění: Navrženo pouze pro 4 kusy Ni-MH akumulátorů s napětím 1,2 V. V případě použití klasických baterií s napětím 1,5 V je nutno použít pouze 3 kusy baterií, vhodný držák baterií a připojit na jeden z napájecích konektorů.

5.1.4 Výkonová část

Zajišťuje výkonové výstupy pro obvody, které nelze připojit k MCU přímo. Buď z důvodu vyššího proudu, než který MCU dokáže zvládnout nebo obvod ke své funkčnosti vyžaduje větší napětí než 5 V. Poslední část je osazena dvěma obvody ULN2803A (IC1 a IC2). Jedná se o tranzistorové pole s osmi NPN tranzistory v jednom integrovaném obvodu. Obvod je přímo určen pro spínání indukčních zátěží, má v sobě integrovány ochranné diody. Maximální možný proud na jeden kanál je 500 mA a maximální vstupní napětí obvodu je 30V. Je plně kompatibilní s TTL logikou, není potřeba žádných dalších součástek. Obvody jsou osazeny v patičkách zaručující snadnou vyměnitelnost. Vstupy obvodů jsou vyvedeny na jednořadé kontaktní lišty SV1 a SV2, výstupy jsou vyvedeny na dvouřadé kontaktní lišty SV5 a SV7.

Výkonová část umožňuje připojení 2 odlišných napájecích napětí z externích zdrojů přes WAGO svorky X1 a X11. Zem (GND) obou napájení je vedena přímo do obvodů IC1 a IC2. Kladný pól každého z napětí je veden přes páčkové přepínače S2 a S5, které umožňují vypnutí napájecích napětí do přepínačů S4 a S5. Přepínače S4 a S5 umožňují volbu jednoho ze dvou externích napětí, které bude připojeno na vstup integrovaného obvodu. Kladný pól je z těchto přepínačů veden i do dvouřadých kontaktních lišt SV7 a SV8. U každého integrovaného obvodu je přítomný miniaturní voltmetr v roli kontrolky a jednoznačně informuje o hodnotě napětí, které je přivedeno na daný integrovaný obvod.

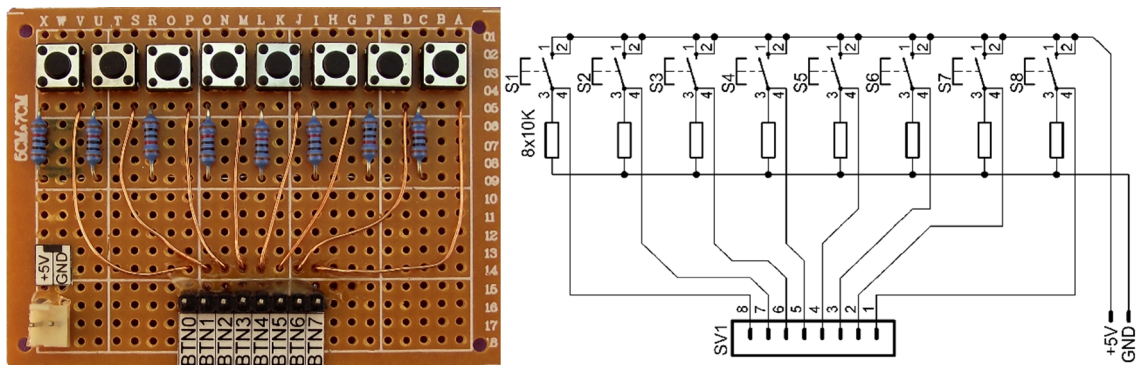


Obrázek 15 – Schéma vývojové desky

5.2 Vstupní moduly

Vstupní moduly poskytují MCU vstupní signály. Signály u tlačítek a klávesnice jsou digitální hodnoty logická 1 (H – 5V) a logická 0 (L – 0V/GND). V případě potenciometrů se jedná o analogové hodnoty napětí 0 až 5 V, které MCU převádí pomocí A/D převodníku na číslo. Pro jednoduchost jsou použity jen tlačítka a maticová klávesnice jako zdroje digitálního signálu a potenciometry zapojené jako odporové děliče reprezentující zdroje analogového signálu. Nic ovšem nebrání dalšímu rozšíření stavebnice např. termistory a fotorezistory jako zdroje analogového signálu, vyjadřující hodnotu teploty a hodnotu intenzity osvětlení. Popřípadě rozšíření stavebnice o další digitální vstupní obvody.

5.2.1 Modul Tlačítka



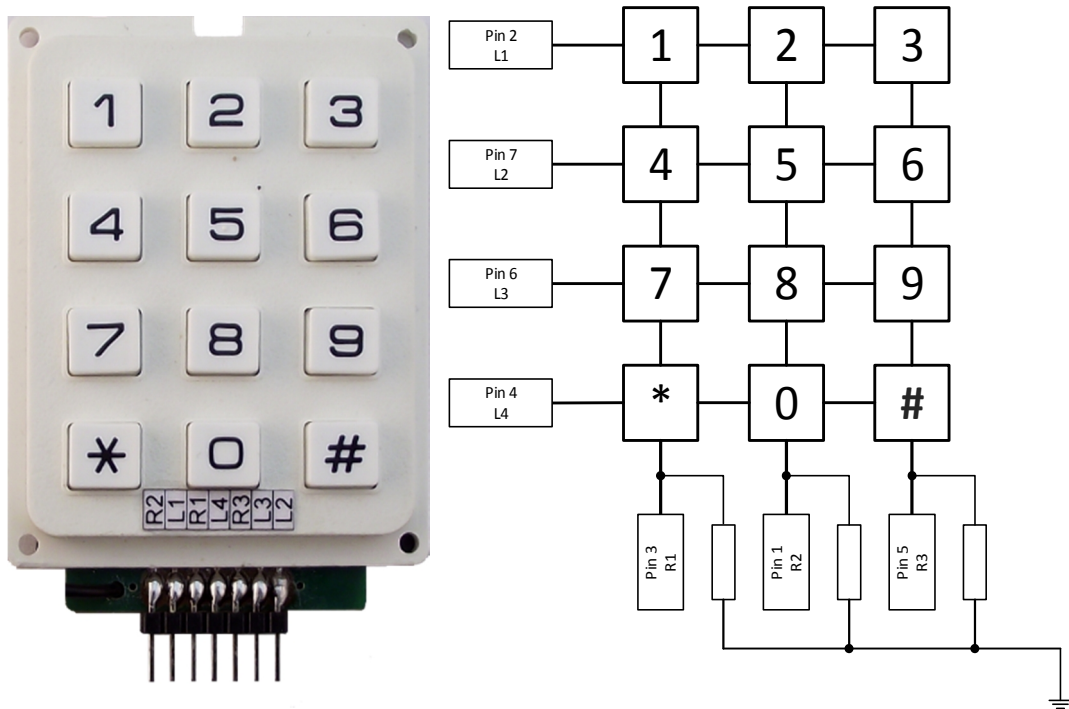
Obrázek 16 – Modul Tlačítka

Modul obsahuje 8 běžně dostupných mikrospínačů o rozměrech 6x6mm s jedním spínacím kontaktem použitých jako tlačítka. Pull down rezistory 10 kΩ jsou připojeny k GND a slouží k zaručení logické hodnoty 0 při nestisknutém tlačítku. Signál je vyveden na jednořadou kontaktní lištu.

5.2.2 Modul Klávesnice

Základem modulu je maticová klávesnice STD HT34.1. Klávesnice obsahuje celkem 12 tlačítek označených 0-9, * a # uspořádaných do matice. Vhodným uspořádáním tlačítek do řad (L) a sloupců (R) lze ušetřit počet pinů potřebných pro připojení klávesnice. V případě, že by klávesnice byla realizována jako modul tlačítka, bylo by potřeba celkem 12 pinů MCU. Při uspořádání tlačítek do matice se počet pinů zredukuje na pouhých 7 a v MCU použijeme vhodný program pro čtení maticové klávesnice.

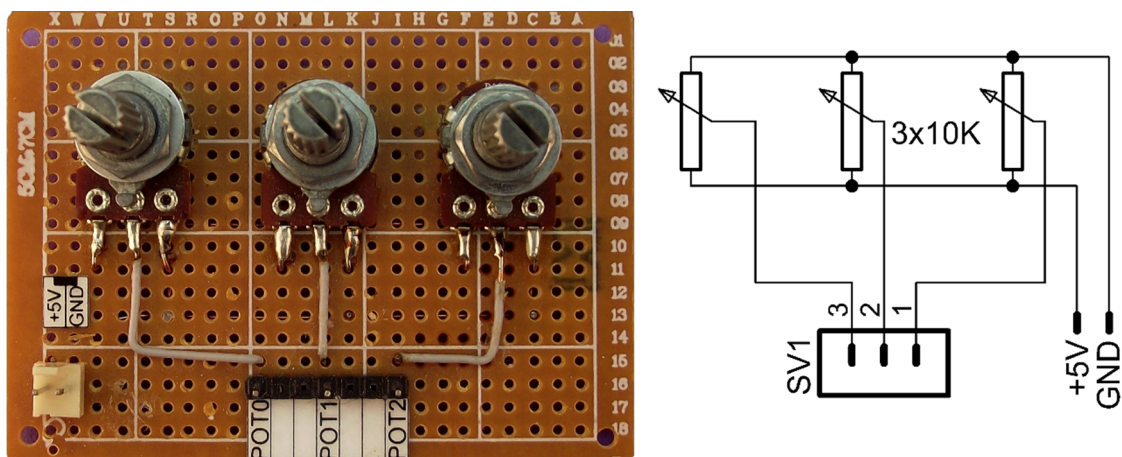
Klávesnice je doplněna třemi 10 kΩ pull down rezistory připojenými k pinům 1, 3 a 5 viz obrázek 17. Signál z klávesnice je vyveden jednořadou kontakto-
 vou lištou.



Obrázek 17 – Modul Klávesnice

Poznámka: na obrázku 17 chybí černý vodič s napájecím konektorem pro připojení pull down rezistorů k GND vývojové desky. Pro funkčnost je vhodné vodič připojit na příslušný konektor.

5.2.3 Modul Potenciometry

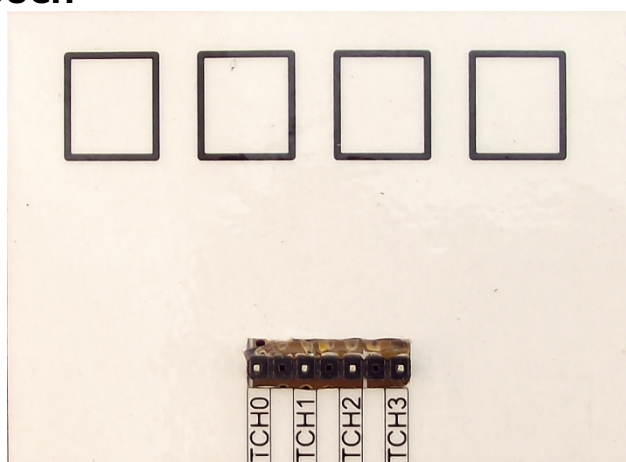


Obrázek 18 – Modul Potenciometry

Modul obsahuje 3 lineární potenciometry 10kΩ/0,25W. Potenciometry jsou zapojeny jako odporové děliče napětí viz obrázek 18. Z výstupů jezdců

potenciometrů jsou analogové signály vyvedeny na jednořadou kontaktní lištu. Modul slouží jako zdroj analogového signálu.

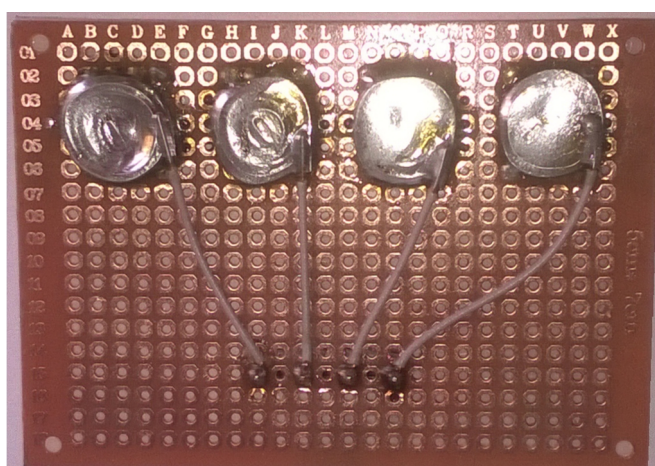
5.2.4 Modul TOUCH



Obrázek 19 – Modul TOUCH

Modul TOUCH není konstrukčně příliš zajímavý, jedná se pouze o vodivé plošky vytvořené na univerzální desce. Plošky jsou vyvedeny na jednořadou kontaktní lištu viz obrázek 20.

Plošky lze vytvořit více způsoby. Jediná nutná podmínka je, aby plošky byly elektricky izolovány ze strany dotyku. V tomto případě izolaci zabezpečuje deska samotná a další izolací je i vrchní popisek - laminovaný papír. Plošky by od sebe měly mít také malý rozestup.



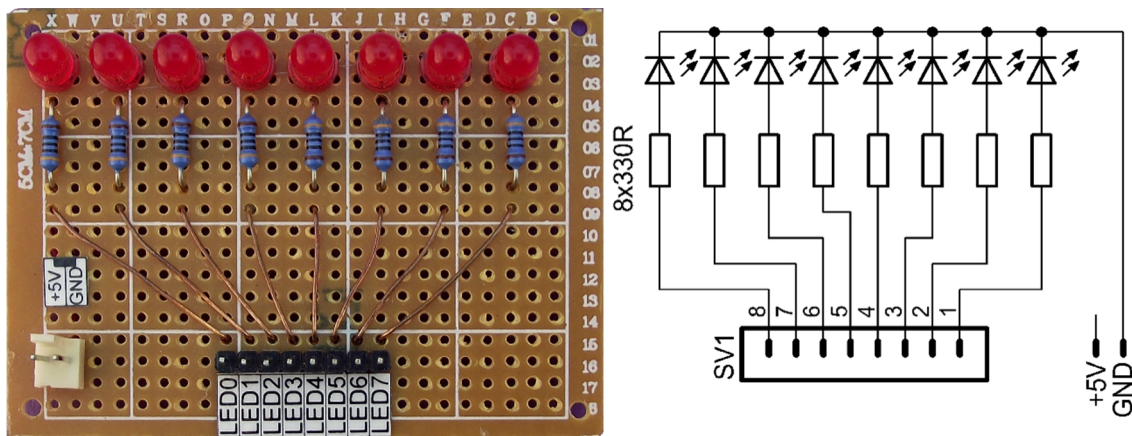
Obrázek 20 – Vodivé plošky

5.3 Výstupní moduly

Výstupní moduly zaujímají místo výstupních zařízení z obrázku 1. Pro potřeby práce byla vybrána základní modelová zapojení, s kterými se při začátkách setká téměř každý. Obdobně jako u modulů vstupních lze vymyslet nespočet zapojení. V podstatě jsme limitováni pouze počtem pinů MCU. Z dalších uskutečněných zapojení mohou uvést zapojení s klasickými relé tak i s SSR relé, ovládání krokových motorů z DVD mechanik pro vystavování čtecí hlavy a další pokusy, které bohužel nejsou pro takto zaměřenou stavebnici příliš vhodné.

Moduly jsou postaveny na jednostranných univerzálních plošných spojích o rozměrech 5x7 cm. Moduly Servo a Krokový motor jsou přilepeny na deskách z plexiskla shodných rozměrů pro zachování formátu pro stolní držák.

5.3.1 Modul LED



Obrázek 21 – Modul LED

Modul obsahuje 8 standardních LED diod průměr 5mm 40-60mcd/20mA ($U_f=2,0V$), zapojených se společnou katodou. Ke každé z osmi anod je připojen předřadný rezistor 330 Ω . Přívody k předřadným rezistorům jsou vyvedeny na jednořadou kontaktní lištu.

Modul je použitelný jako jednoduchá signalizace stavu.

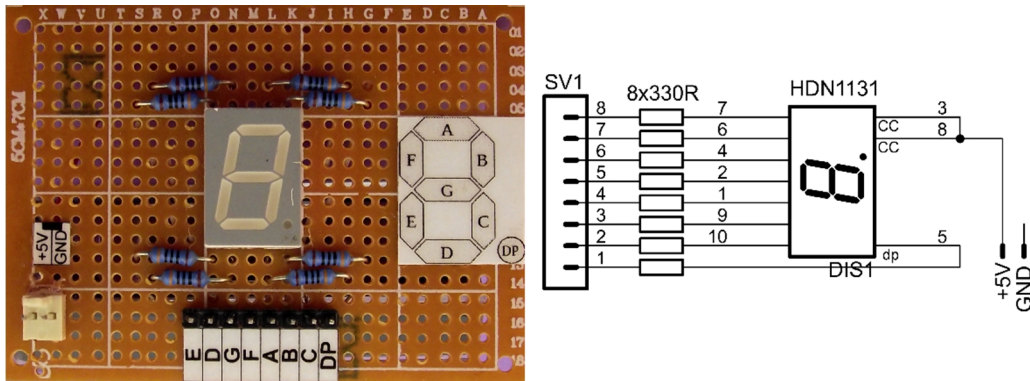
5.3.2 Modul 7 segment

Modul obsahuje sedmi segmentový LED displej HDN1131. Display používá zapojení se společnou anodou, tedy opačně než modul LED. Každá LED dioda z jednotlivých segmentů je připojena přes předřadné rezistory hodnoty 330 Ω na jednořadou kontaktní lištu.

V podstatě je tento modul shodný s modulem LED. Jedinými rozdíly je zapojení se společnou anodou a rozmístění LED diod. Jednotlivé LED diody jsou, umístěny

v jednom pouzdře ve tvaru číslice. Rozsvěcením vhodných segmentů můžeme tedy velmi snadno na displeji zobrazovat číslice např.: pro zobrazení číslice 2 přivedeme na segmenty ABGED hodnotu L pro zobrazení číslice 8 přivedeme hodnotu L na všechny segmenty kromě DP – desetinné tečky.

Modul lze velmi snadno použít k zobrazování číslic.



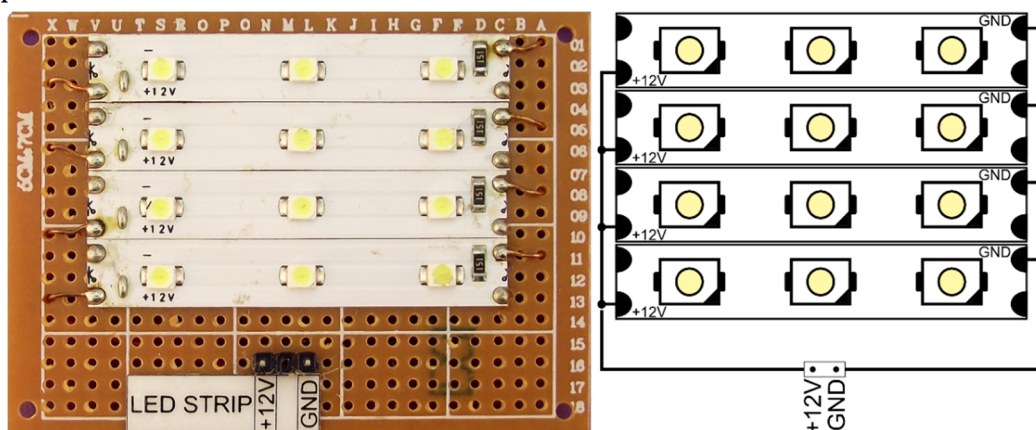
Obrázek 22 – Modul 7 segment

5.3.3 Modul LED strip

Modul je osazen čtyřmi dílky standardního 12V bílého LED pásku používaného pro osvětlení. Pásky jsou přilepeny na univerzální desce, z důvodu zachování jednotného formátu modulů. Jednotlivé pásky jsou spolu propojeny a přívody jsou vyvedeny na jednořadou kontaktovou lištu.

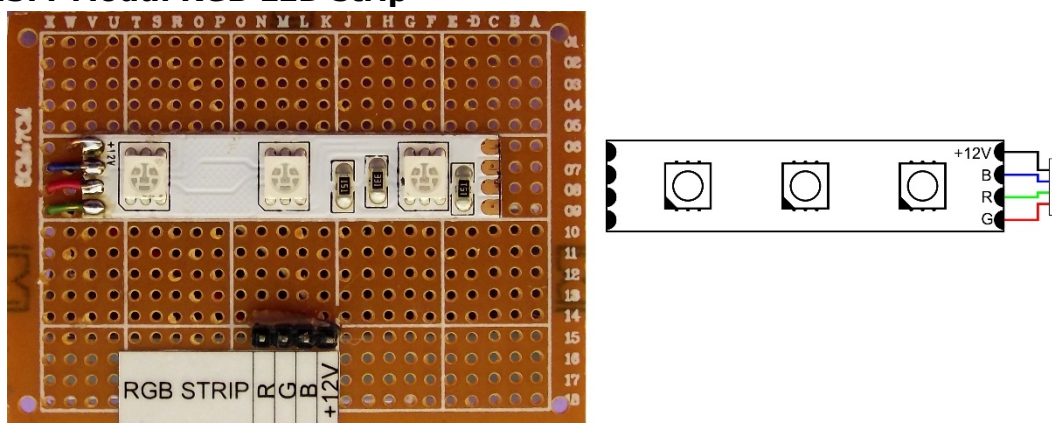
Modul demonstruje možnosti PWM regulace osvětlení.

Upozornění: Modul je určen pro napětí 12V, připojením přímo na piny PICAXE by mohlo dojít k jeho poškození. Modul vždy připojujte přes tranzistorové pole na napětí 12V.



Obrázek 23 – Modul LED strip

5.3.4 Modul RGB LED strip



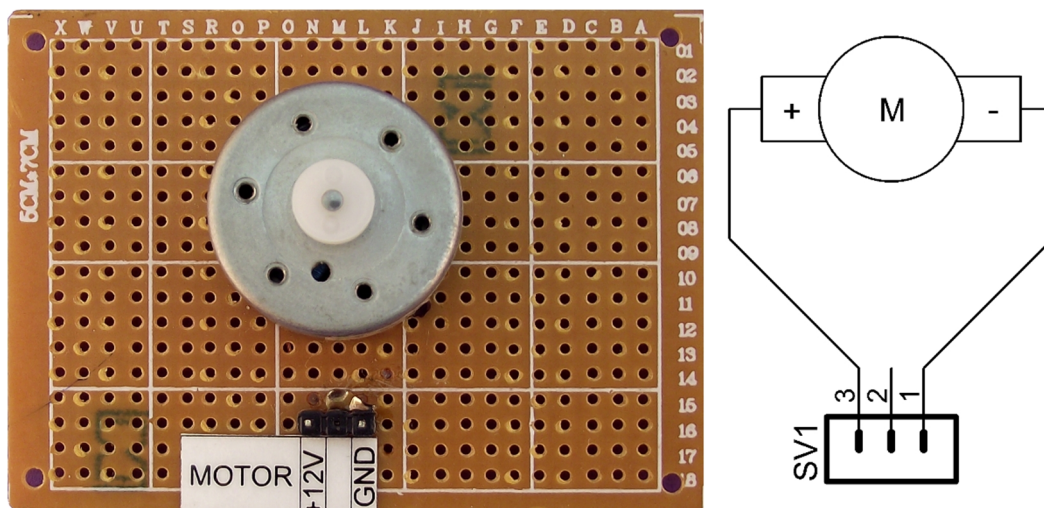
Obrázek 24 – Modul RGB LED strip

Modul je osazen jedním dílkem 12V RGB LED pásku s SMD diodami typu 5050 obsahující v jednom pouzdře všechny 3 barevné složky. Pásek je přilepen na univerzální desce a přívody pásku jsou vyvedeny na jednořadou kontaktovou lištu.

Modul demonstruje možnosti míchání barev RGB pomocí PWM regulace.

Upozornění: Modul je určen pro napětí 12V, při připojení přímo na piny PICAXE modul nebude svítit. Modul je určen pro připojení k tranzistorovému poli na napětí 12V.

5.3.5 Modul Motor



Obrázek 25 – Modul Motor

Modul obsahuje malý stejnosměrný motor neznámých parametrů pocházející z DVD mechaniky LG, avšak pro naši demonstrační aplikaci plně vyhovuje. Motor bez výrazného zahřívání pracuje i na 12V, při připojení na 5V se točí pomaleji.

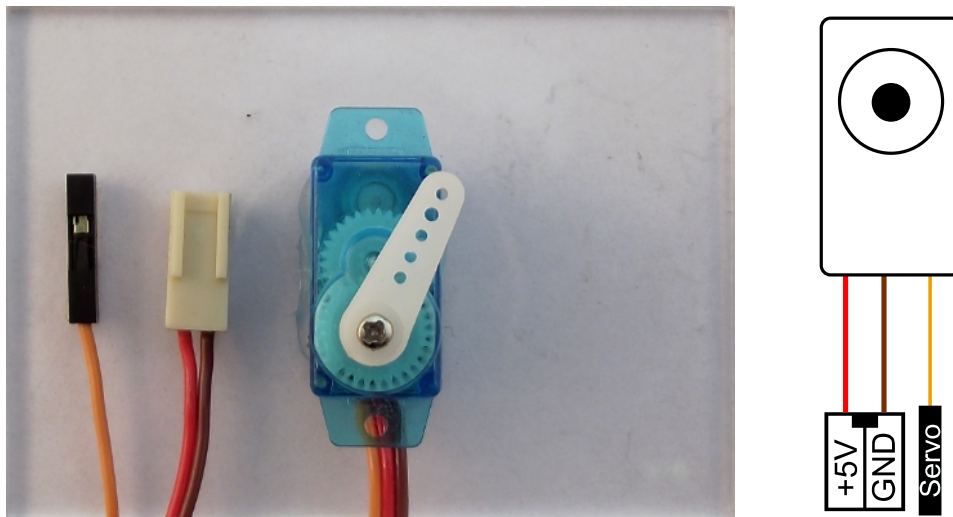
Motor je připájen na univerzální desku pro zachování jednotného formátu, přívody jsou vyvedeny na jednořadou kontaktovou lištu. Změna směru otáčení je

možná změnou polarity. Při zapojení podle obrázku 25 se otáčí po směru hodinových ručiček.

Modul demonstruje ovládání stejnosměrných motorů.

Upozornění: Modul je určen pro připojení na tranzistorové pole nebo na modul H-můstek. Připojením modulu přímo na piny PICAXE způsobí jeho přetížení a zničení.

5.3.6 Modul Servo



Obrázek 26 – Modul Servo

Základem modulu Servo je miniaturní analogové modelářské servo GO-05 přilepené na destičce plexiskla se stejnými rozměry jako univerzální desky pro zachování formátu. Základní parametry serva GO-05 jsou [23]:

- 0,14 s na 60° natočení při 4,8 V
- Napájecí napětí 4,8 – 6 V
- Rozsah natočení 0° - 180°

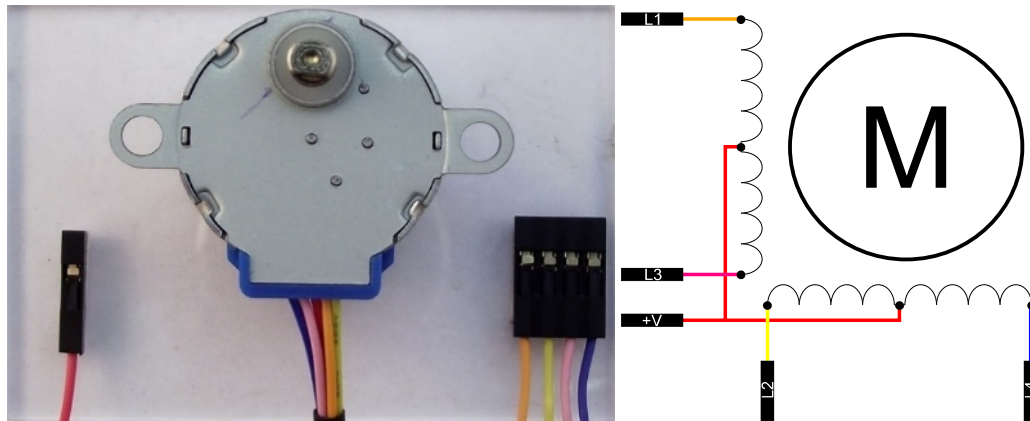
Právě modelářská serva jsou velmi často požívaná při amatérských konstrukcích malých robotů a v robotice pro jejich přesnost a dostupnost. Všeobecně jsou všechna modelářská serva tří vodičová. Dva vodiče slouží k napájení a zbylý jeden vodič pro ovládání pozice serva.

Ovládání serva je poměrně jednoduché. Servo mimo převodů a motoru obsahuje elektroniku a zpětnovazební potenciometr pro informaci o poloze hřídele. K ovládání tedy stačí pouze jeden vodič. V našem případě oranžový s označením Servo, na který vysíláme pulzy o délce 0,75 až 2,25 ms každých 20 ms. Pulzy o délce 0,75 ms vysílané na ovládací vodič znamenají, že elektronika uvnitř serva vyhodnotí pozici hřídele a přestaví servo do jedné krajní polohy, pokud v ní servo právě není a

v této krajní poloze servo dále drží. Pulzy o délce 2,25 ms znamenají opačnou krajní polohu a např. pulzy o délce 1,5 ms znamenají prostřední polohu serva. Ovládacím vodičem tedy dáváme pouze informaci, do jaké polohy chceme servo natočit, a o samotné ovládní motoru serva se stará elektronika uvnitř serva.

Použité servo je jedno z nejmenších a také nejlevnějších, ale pro demonstrační účely práce s modelářskými servy naprosto dostačuje.

5.3.7 Modul Krokový motor



Obrázek 27 – Modul Krokový motor

Základem modulu Krokový motor je krokový motor 28BYJ-48. Jedná se o unipolární krokový motor s převodovkou. Uvedme základní parametry z datasheetu:

- Převodový poměr 1/64
- 64 kroků na 1 otáčku
- Napájecí napětí 5 V

Princip krokového motoru je poměrně jednoduchý. Motor je složen ze statoru a rotoru. Ve statoru se nachází vinutí složené z cívek. Při průchodu proudu cívkami se vytváří magnetické pole, které přitahuje opačný magnetický pól na rotoru. Správnou sekvencí spínání cívek lze tedy vytvořit točivé magnetické pole, které bude otáčet rotorem [24].

Sekvenci řízení krokového motoru s polovičním krokem vyjadřuje následující tabulka 1:

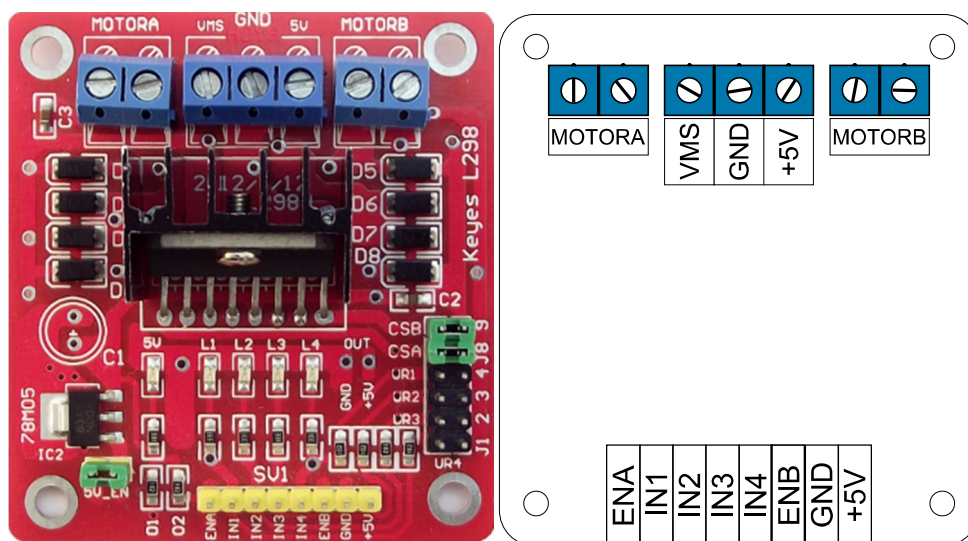
Krok	Cívka			
	L1	L2	L3	L4
1	H	L	L	L
2	H	H	L	L
3	L	H	L	L
4	L	H	H	L
5	L	L	H	L
6	L	L	H	H
7	L	L	L	H
8	H	L	L	H

Tabulka 1 – Řídicí sekvence krokového motoru [24]

Změna směru otáčení motoru se provádí změnou posloupnosti řídicí sekvence z 1-8 na 8-1.

Upozornění: Modul je určen pro připojení na tranzistorové pole na napětí 5 V. Připojením modulu přímo na piny PICAXE způsobí jeho přetížení a zničení.

5.3.8 Modul H-můstek



Obrázek 28 – Modul H-můstek



Modul H-můstek není vlastní konstrukce, je použit hotový modul Keyes L298 V1. Hlavně z ekonomických důvodů, kdy cena za hotový modul od asijského obchodníka je téměř stejná, jako cena za totožný samostatný obvod L298 v obvyklé obchodní síti v České republice.

Obvod L298 je integrovaný obvod obsahující 2 plné H-můstky. Zvládá napájecí napětí až 46 V a maximální výstupní proud při dobrém chlazení je až 4 A (2 A na

jeden H-můstek). Obvod plně podporuje vstupní logiku TTL a umožňuje ovládání solenoidových ventilů, nebo jednoho bipolárního krokového motoru, nebo dvou stejnosměrných motorů. Vstupy IN1 až IN4 ovládají výstupy H-můstků. IN1 a IN2 pro první H-můstek, jehož výstupem na modulu je svorkovnice MOTORA. Vstupy IN3 a IN4 ovládají druhý H-můstek, na modulu je jeho výstupem svorkovnice MOTORB. Vstupy ENA a ENB umožňují vypnutí výstupů H-můstků nezávisle na hodnotách vstupů IN (při přivedení logické hodnoty L na vstupy dojde k vypnutí výstupu H-můstku) [25].

Modul navíc obsahuje napěťový regulátor 78M05. Není tedy nutné obvod napájet 5 V přes vstupy GND a +5V, postačí pouze připojení modulu na napájecí napětí GND a VMS na prostřední svorkovnici. V případě použití externího zdroje napájení, které nemá společné GND s vývojovou deskou je nutné propojit pin GND na modulu s pinem GND z jednoho z napájecích 5V konektorů na vývojové desce.

Pravdivostní tabulka získaná proměřením výstupů voltmetrem je zachycena v tabulce 2.

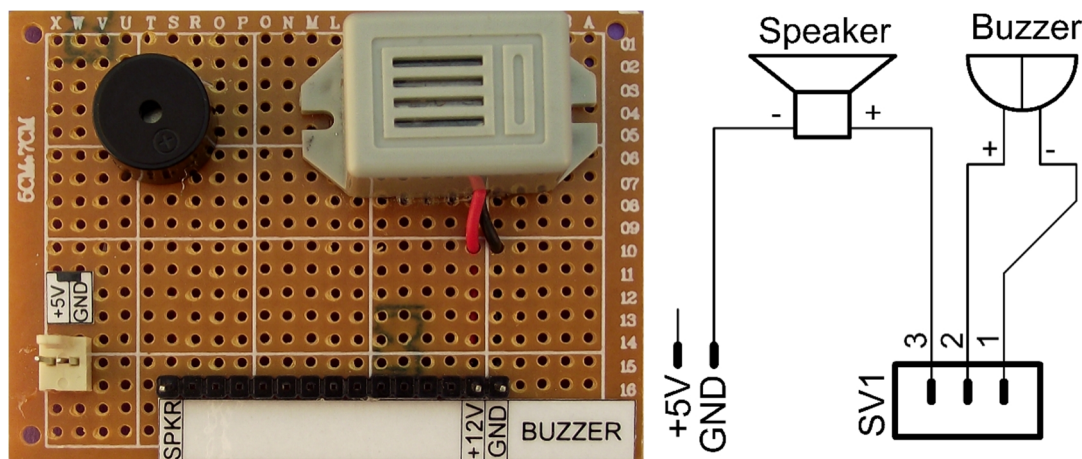
Vstup		Výstup		Svorkovnice
IN1	IN2	MOTORA		
L	L	0 V	0 V	
L	H	+VMS	GND	
H	L	GND	+VMS	
H	H	0 V	0 V	
IN3	IN4	MOTORB		
L	L	0 V	0 V	
L	H	+VMS	GND	
H	L	GND	+VMS	
H	H	0 V	0 V	

Tabulka 2 – Pravdivostní tabulka modulu H-můstek

Modul je v naší stavebnici vybaven propojovacím kabelem s konektorem pro připojení na napájecí napětí 5 V z vývojové desky, nic však nebrání připojení modulu až na maximální možné napětí 36 V z externího zdroje.

Podrobnější informace o obvodu L298 viz datasheet [25].

5.3.9 Modul Reprodaktor



Obrázek 29 – Modul Reprodaktor

Modul je osazen piezo reproduktorem neznámého typu, a bzučákem KST-3612. Obě součásti mají přívody vyvedeny na jednořadou kontaktovou lištu. Piezoreproduktor (SPKR) se připojuje přímo na piny PICAXE, bzučák (BUZZER) je určený pro napětí 12 V a pro správnou funkci musí být připojen přes tranzistorové pole.

5.4 Stolní držák stavebnice

V průběhu práce se stavebnicí vznikl stolní držák klávesnice na vývojovou desku a celkem až 6 externích modulů. Usnadňuje práci s celou stavebnicí, zvláště s externími moduly. Držák umožňuje pohodlnou manipulaci se stavebnicí i v případě, že je zapojena úloha s více moduly.

Držák je uzpůsoben rozměrům vývojové desky a externích modulů. Je vyroben z jeklu o rozměru 10x10x0,7 mm a závitových tyčí M5. Elektrická izolace modulů a vývojové desky je řešena za použití samolepícího těsnění profilu D šířky 10 mm. Povrchová úprava je černá akrylátová barva odstínu RAL 9005. Fotografie držáku jsou přiloženy v příloze C na obrázcích 63 a 64.

5.5 Systém propojování

K propojování vývojové desky a modulů slouží propojovací vodiče s dutinkami na obou koncích. Součástí stavebnice jsou i propojovací vodiče s dutinkami na jednom konci a kolíky na konci druhém, pro propojení modulu H-můstek a modulu motor, nebo pro budoucí propojení stavebnice s kontaktním polem. Obou druhů propojovacích vodičů je dostatek i pro budoucí rozšiřování stavebnice.

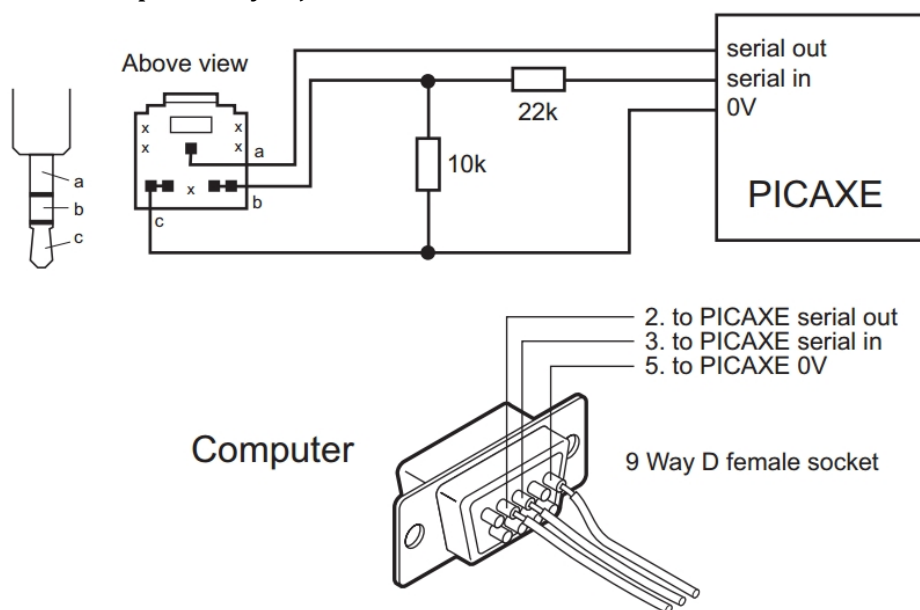
Propojovací schémata vycházejí z popisků pinů nebo barev přípojných vodičů modulu a jsou zpracovány graficky. V případě páčkových přepínačů je v propojovacím schématu vždy barevně označena jejich poloha pro danou úlohu. Pro všechny programy je dáno, že externí napětí UE1 je připojeno na 5 V napájecí části vývojové desky přes napájecí konektor se zámkem. UE2 jako druhé napájecí napětí pro stavebnici je uvažováno 12 V. Zdrojem tohoto napětí jsou tři v sérii zapojené li-ion akumulátory 18650 v pouzdře s ochrannou proti zkratu, nadproudu a ochranou proti podvybití. Pouzdro je přišroubováno na držák stavebnice.

Pro přehlednost jsou ze schémat propojení vynechány napájecí konektory. V případě, že je konektor na modulu umístěn, je na modulu označen značkou z obrázku 30 a je vždy nutné pro funkci modul propojit s vývojovou deskou propojovacím kabelem s příslušnými konektory. Propojovací kabely jsou také součástí stavebnice v dostatečném množství.



Obrázek 30 – Označení napájecího konektoru

Programovací kabel plně vychází ze zapojení v manuálu viz obrázky 31. Vývojová deska se tedy připojuje k počítači přes sériový port nebo je možné použít převodník USB na sériový port. Doporučuji použití některého s čipem FTDI pro bezproblémovou práci s vývojovou deskou.



Obrázek 31 – Programovací kabel (převzato z [19])

6 Úlohy s LED diodami

Následující série úloh představuje základní ukázkové programy pro moduly:

1. Modul LED
2. Modul 7 segment
3. Modul LED strip
4. Modul RGB LED strip
5. Modul Tlačítka
6. Modul Potenciometry

6.1 Světelný had

Program vytváří světelný efekt běžícího hada pomocí 8 led diod. Směr blikání led diod je z pravé strany na levou. Délka svitu jednotlivých led diod, tedy rychlost efektu, je dána hodnotou proměnné Pauza. V tomto případě 200ms.

6.1.1 Ukázkový program

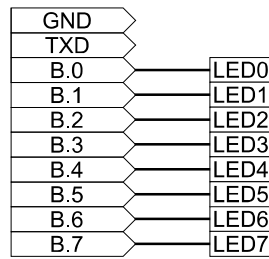
```
;Program LED had.bas

dirsb = %11111111 ;port B nastaví jako binární výstup
symbol Pauza = w0
Pauza = 200

main:
  let pinsb = %10000000 ;rozsvítí LED7 na pinu B.7
  pause Pauza ;pauza 200ms
  let pinsb = %01000000 ;rozsvítí LED6 na pinu B.6
  pause Pauza
  let pinsb = %00100000 ;rozsvítí LED5 na pinu B.5
  pause Pauza
  let pinsb = %00010000 ;rozsvítí LED4 na pinu B.4
  pause Pauza
  let pinsb = %00001000 ;rozsvítí LED3 na pinu B.3
  pause Pauza
  let pinsb = %00000100 ;rozsvítí LED2 na pinu B.2
  pause Pauza
  let pinsb = %00000010 ;rozsvítí LED1 na pinu B.1
  pause Pauza
  let pinsb = %00000001 ;rozsvítí LED0 na pinu B.0
  pause Pauza

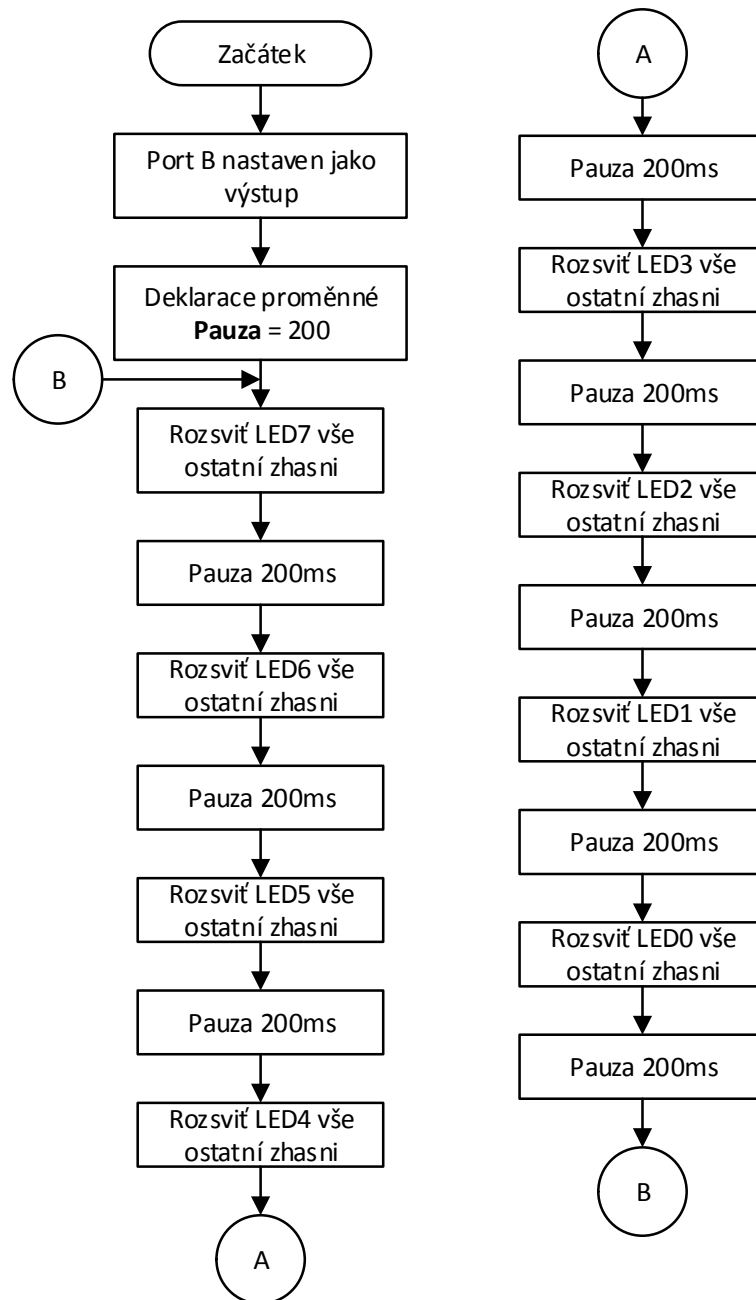
goto main
```


6.1.2 Schéma propojení



Obrázek 32 – Schéma propojení úlohy LED had.bas

6.1.3 Vývojový diagram



Obrázek 33 – Vývojový diagram programu LED had.bas

6.1.4 Náměty na rozšíření programu

1. Změna směru blikání led diod na směr zleva doprava a změna rychlosti světelného efektu.
2. Zkombinování obou směrů tak, aby efekt plynul zleva doprava a zase zpět.
3. Navýšení počtu led diod, které svítí současně např. na 2.

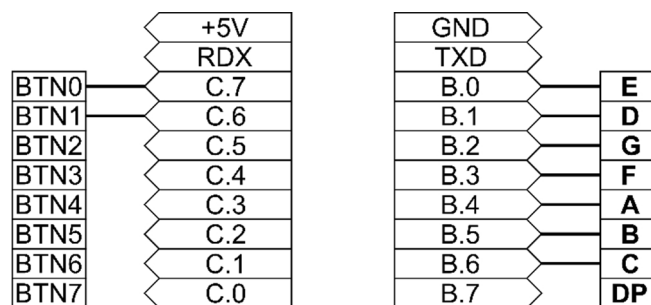
6.2 Zobrazovač čísel

Program po spuštění PICAXE zobrazuje na modulu 7 segmentovka číslo nula. Tlačítkem BTN0 se číslo s každým stiskem zvětšuje o jedna až do 9, stiskem tlačítka BTN1 se číslo s každým stiskem zmenšuje o jedna až do nuly.

6.2.1 Ukázkový program

```
;program cisla.bas
dirsb = %11111111
main:
  if pinC.7 = 1 and b0 != 9 then ; podmínka pro tlačítko BTN0,
  zaručující aby proměnná dosáhla maximální hodnoty 9
    inc b0 ; když je stisknuto tlačítko BTN0 inkrementuje
    proměnnou b0
    pause 300 ; pauza 300ms
  endif
  if pinC.6 = 1 and b0 !=0 then ; podmínka pro tlačítko BTN1,
  zaručující aby proměnná nepřekročila minimální hodnotu 0
    dec b0 ; když je stisknuto BTN1 dekrementuje proměnnou b0
    pause 300
  endif
select case b0 ; podmínka case, podle hodnoty proměnné b0 skočí na
příslušný příkaz
  case 0
  let pinsb = %10000100 ; číslice 0 - svítí segmenty ABCDEF
  case 1
  let pinsb = %10011111 ; číslice 1 - svítí segmenty BC
  case 2
  let pinsb = %11001000 ; číslice 2 - svítí segmenty ABGED
  case 3
  let pinsb = %10001001 ; číslice 3 - svítí segmenty ABCDG
  case 4
  let pinsb = %10010011 ; číslice 4 - svítí segmenty BCGF
  case 5
  let pinsb = %10100001 ; číslice 5 - svítí segmenty ACDFG
  case 6
  let pinsb = %10100000 ; číslice 6 - svítí segmenty ACDEFG
  case 7
  let pinsb = %10001111 ; číslice 7 - svítí segmenty ABC
  case 8
  let pinsb = %10000000 ; číslice 8 - svítí segmenty ABCDEFG
  case 9
  let pinsb = %10000011 ; číslice 9 - svítí segmenty ABCFG
endselect
goto main ; zpět na návěští main
```

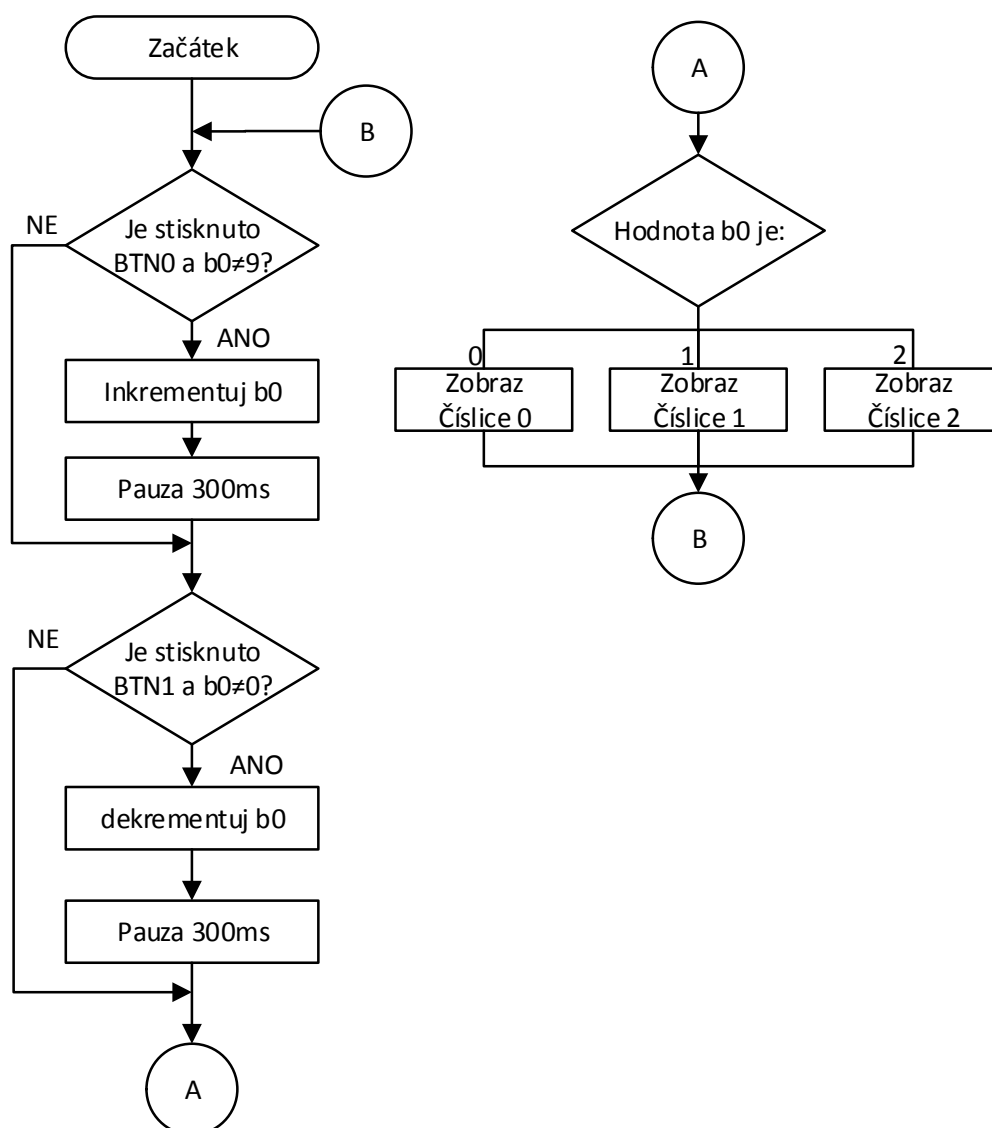
6.2.2 Schéma propojení



Obrázek 34 – Schéma propojení programu cisla.bas

6.2.3 Vývojový diagram

Pro přehlednost jsou v obrázku 35 vynechány možnosti case 3 až case 9.



Obrázek 35 - Vývojový diagram programu cisla.bas

6.2.4 náměty na rozšíření programu

1. Vložení cyklů pro automatické počítání od 0 do 9 a zpět od 9 do nuly.

6.3 Stmívač led pásku

Program plynule rozsvěcí do plného svitu bílý led pásek. Po dosažení plného svitu ho plynule zhasíná do zhasnutého stavu. Program využívá PWM regulaci a s ní spojené příkazy PWMOUT a PWMDUTY. Popis syntaxe čerpá z manuálu [18].

6.3.1 Syntaxe příkazů

PWMOUT pin, period, duty

Pin - konstanta označující pin, na kterém bude generováno PWM (u 20M2 použitého v našem případě piny: B.1, C.2, C.3 a C.5)

Period - proměnná nebo konstanta (0-255) nastavující periodu PWM.

Duty - proměnná nebo konstanta (0-1023) vyjadřující střidu PWM.

Příkaz generuje kontinuální PWM signál interním PWM modulem. Tento příkaz je odlišný od většiny jiných příkazů, běží nepřetržitě na pozadí. Vypnutí generování signálu se provádí příkazem PWMOUT pin, OFF, k deaktivaci dojde vlivem příkazů NAP, SLEEP a END.

PWMDUTY pin, duty

Pin - konstanta označující pin, na kterém bude generováno PWM (u 20M2 použitého v našem případě piny: B.1, C.2, C.3 a C.5)

Duty – proměnná nebo konstanta (0-1023) nastavující střidu PWM

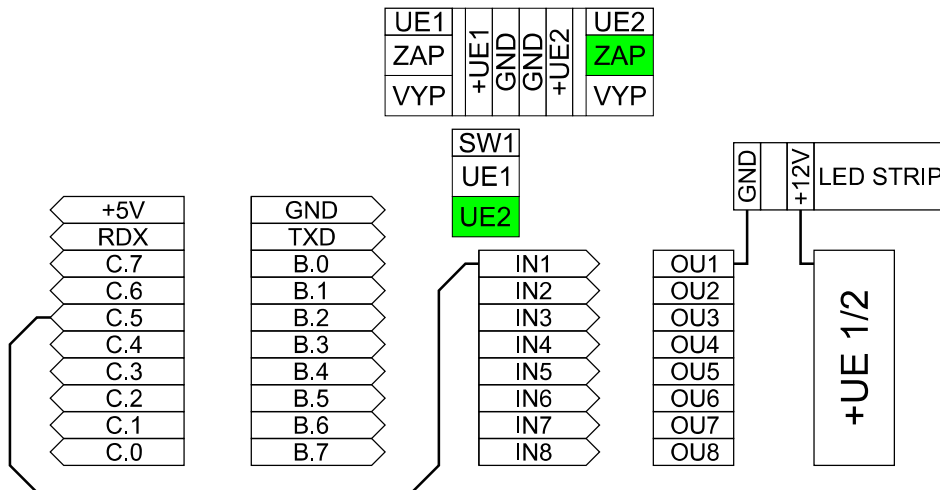
Nastavuje střidu PWM bez resetování vnitřního časovače, příkaz musí být použit až po aktivaci PWM příkazem PWMOUT.

6.3.2 Ukázkový program

```
;program LEDstrip_PWM.bas
pwmout C.5,255,w0 ;zapíná generování PWM signálu na pinu C.5
main:
for w0 = 0 to 1023 step 1 ;cyklus for od 0 do 1023 krok 1
    PWMduty C.5,w0 ;nastavuje střidu na hodnotu proměnné w0
next w0

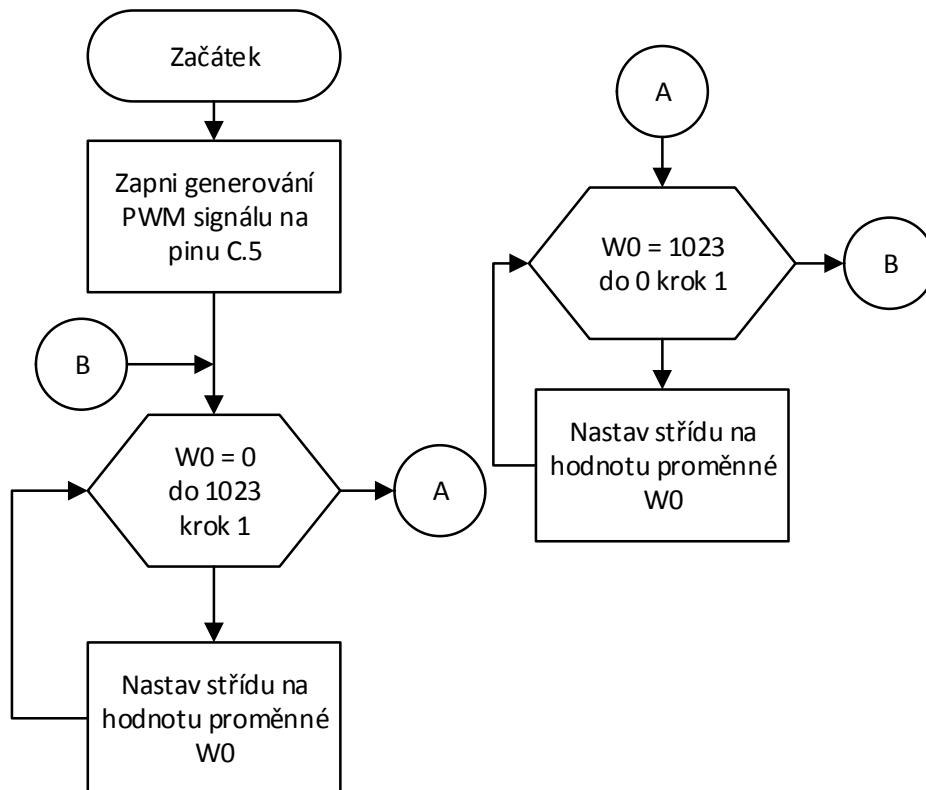
for w0 = 1023 downto 0 step 1 ;cyklus for od 1023 do 0 krok 1
    PWMduty C.5,w0 ;nastavuje střidu na hodnotu proměnné w0
next w0
goto main
```

6.3.3 Schéma propojení



Obrázek 36 – Schéma propojení programu LEDstrip_PWM.bas

6.3.4 Vývojový diagram



Obrázek 37 – Vývojový diagram programu LEDstrip_PWM.bas

6.3.5 Náměty na rozšíření programu

1. Změna rychlosti rozsvícení a zhasínání.
2. Přidání modulu tlačítka, kdy při stisku BTN0 se pásek plynule rozsvítí, při stisku BTN1 pásek plynule zhasne.
3. Nahrazení modulu LED strip, modulem motor.

6.4 Kontrolér RGB pásku

Program čte hodnoty napětí z potenciometrů a na základě hodnoty napětí mění intenzitu svitu RGB LED pásku od 0 do maxima. POT0 ovládá červené LED, POT1 ovládá zelené LED a POT3 ovládá modré LED.

6.4.1 Ukázkový program

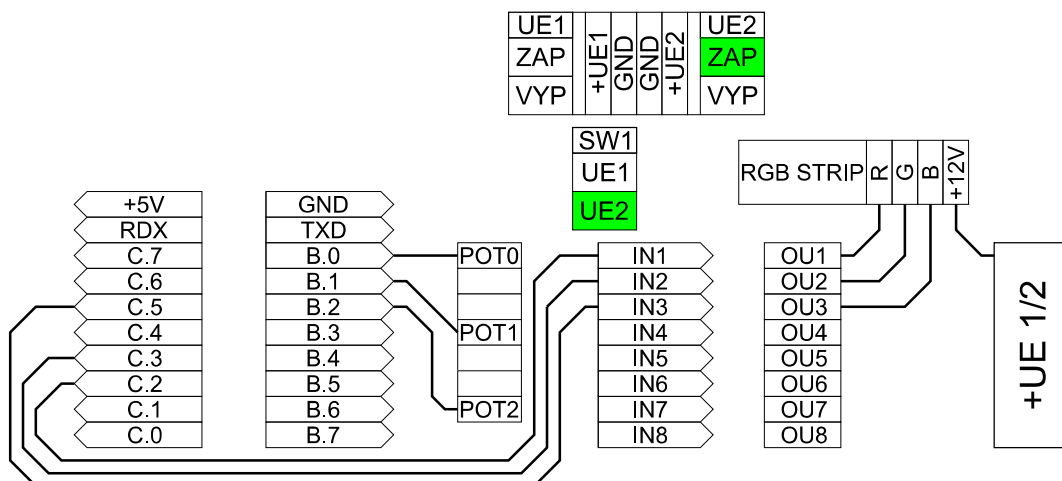
```

;program RGBstrip_PWM.bas
pwmout C.2,255,w0
pwmout C.3,255,w1
pwmout C.5,255,w2
;zapíná generování PWM signálu na pinu C.2, C.3 a C.5
main:
readadc10 B.0, w0
readadc10 B.1, w1
readadc10 B.2, w2
;čte hodnoty napětí z POT0, POT1 a POT2 a převádí je na 10 bitové
číslo, výsledek převodu ukládá do příslušných proměnných w0 - w2
if w0 <= 5 then
  w0 = 0
endif
if w1 <= 5 then
  w1 = 0
endif
if w2 <= 5 then
  w2 = 0
endif
; podmínky pro úplné zhasnutí led, při krajní poloze běžce
potenciometru na výstupu není přesně 0 V, ale řádově desítky mV a ADC
již na toto napětí reaguje. Podmínky zaručí zhasnutí LED při krajní
poloze jezdece potenciometru.

PWMDuty C.2,w0
PWMDuty C.3,w1
PWMDuty C.5,w2
;nastavuje střihu na hodnotu proměnných w0 - w2
goto main

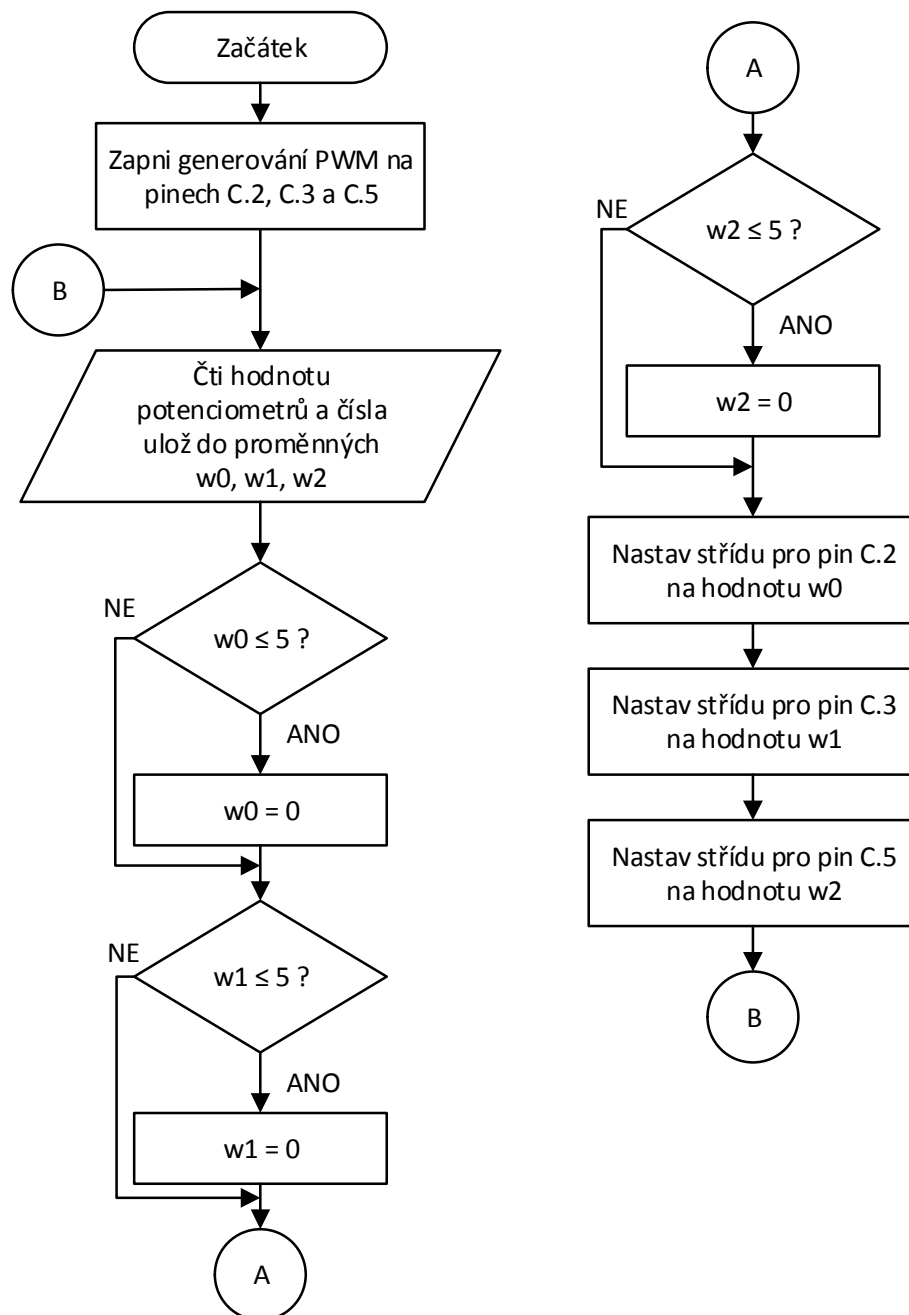
```

6.4.2 Schéma propojení



Obrázek 38 – Schéma propojení RGBstrip_PWM.bas

6.4.3 Vývojový diagram



Obrázek 39 – Vývojový diagram programu RGBstrip_PWM.bas

6.4.4 Náměty na rozšíření programu

1. Přidání modulu tlačítka a přepsání programu tak, aby se barvy nastavovaly jedním potenciometrem, např.: při stisku BTN0 se nastavují červené LED, BTN1 nastavuje zelené, BTN2 modré.
2. Odebrání modulu potenciometry, přepsání programu na automatickou změnu barev.

7 Generování zvuků

V následující úloze popíšeme jednotlivé možnosti, jak lze generovat zvuky a krátké melodie modulem Reproduktor. Je důležité zmínit, že PICAXE nedokáže reprodukovat rozsáhlejší a kvalitní zvukové nahrávky hlavně kvůli velmi omezené vnitřní paměti, ale v praxi nám často postačí velice jednoduché zvuky jako pípnutí při stisku tlačítka jako zpětná vazba nebo krátká monotónní melodie při dokončení úkolu. Zvuky můžeme generovat následovně. Popisy syntaxí v kapitole čerpají z manuálu [18].

7.1 Příkaz sound

Tento příkaz je určený pro generování jednoduchých a krátkých zvuků piezo reproduktorem. Syntaxe příkazu je následující. Popis syntaxe čerpá z manuálu [18].

SOUND pin,(note, duration, note, duration ...)

Pin - konstanta, která určuje pin na kterém je připojen piezoměnič.

Note - proměnná nebo konstanta (0-255), udávající výšku tónu. Hodnoty 0-127 jsou tóny a hodnoty 128-255 jsou šумы.

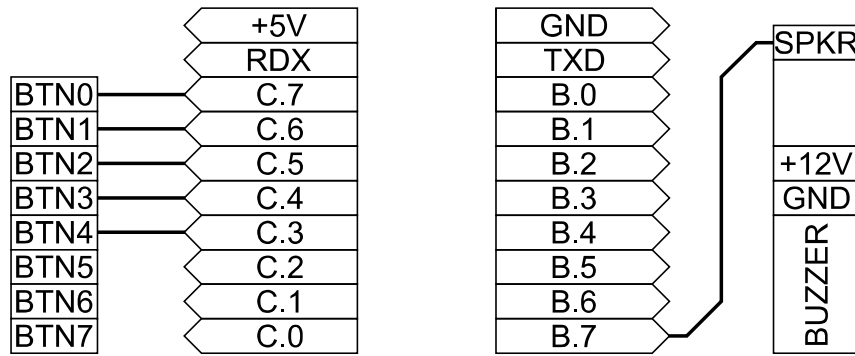
Duration - proměnná nebo konstanta (0-255), která určuje dobu trvání tónu v násobcích 10ms.

Při změně taktovací frekvence se doba zkracuje na polovinu při 8 MHz, při 16 MHz se zkracuje na ¼.

7.1.1 Ukázkový program

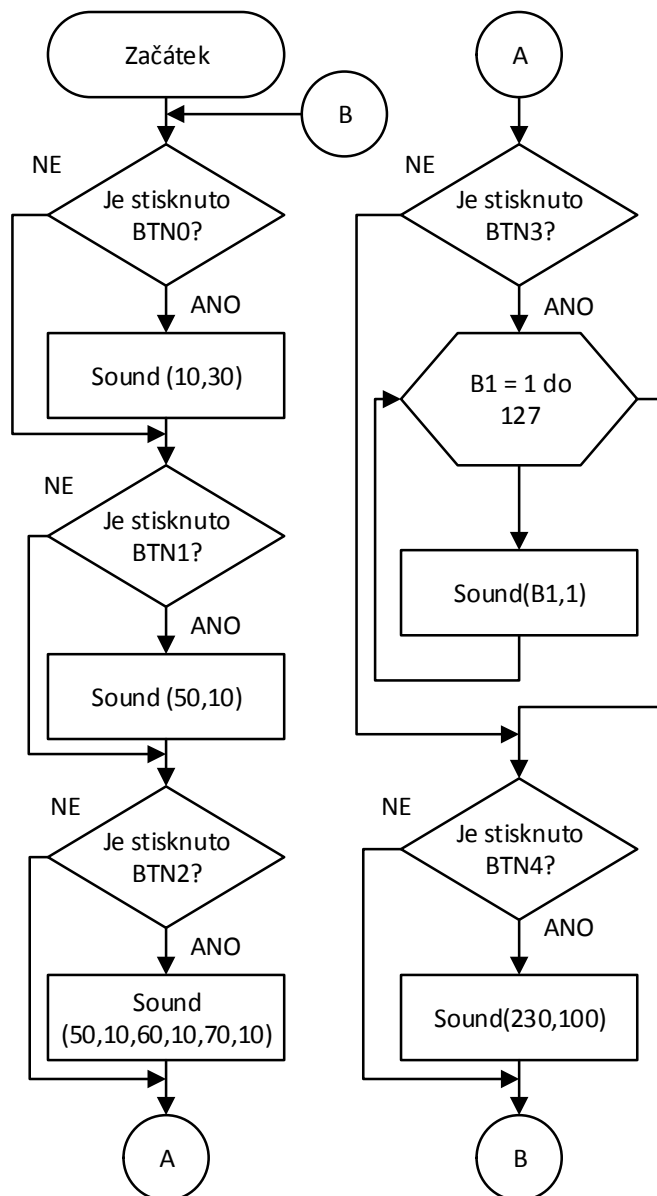
```
;Program Sound_0.bas
main:
  if pinC.7 = 1 then
    sound B.7, (10,30) ;zvuk tónu 10 o délce 300ms
  endif
  if pinC.6 = 1 then
    sound B.7, (50,10) ;zvuk tónu 50 o délce 100ms
  endif
  if pinC.5 = 1 then
    sound B.7, (50,10,60,10,70,10) ;sekvence tónů 50, 60 a 70
  endif
  if pinC.4 = 1 then
    for b1 = 1 to 127
      sound B.7, (b1,1) ;stoupající zvuk použitím cyklu
    next b1
  endif
  if pinC.3 = 1 then
    sound B.7, (230,100) ;šum o délce 1s
  endif
goto main
```


7.1.2 Schéma propojení



Obrázek 40 – Schéma propojení Sound_0.bas

7.1.3 Vývojový diagram



Obrázek 41 – Vývojový diagram programu Sound_0.bas

7.1.4 Náměty na rozšíření programu

1. Změna frekvence na 8 MHz pomocí příkazu **setfreq m8** nebo na 16 MHz příkazem **setfreq m16** pro zkrácení délky tónů.
2. Změna konstant note, nebo tvorba dalších zvuků.

7.2 Příkaz tune

Příkazem **tune** můžeme generovat delší, jednoduché monofonní melodie. Syntaxe příkazu je následující. Popis syntaxe čerpá z manuálu [18].

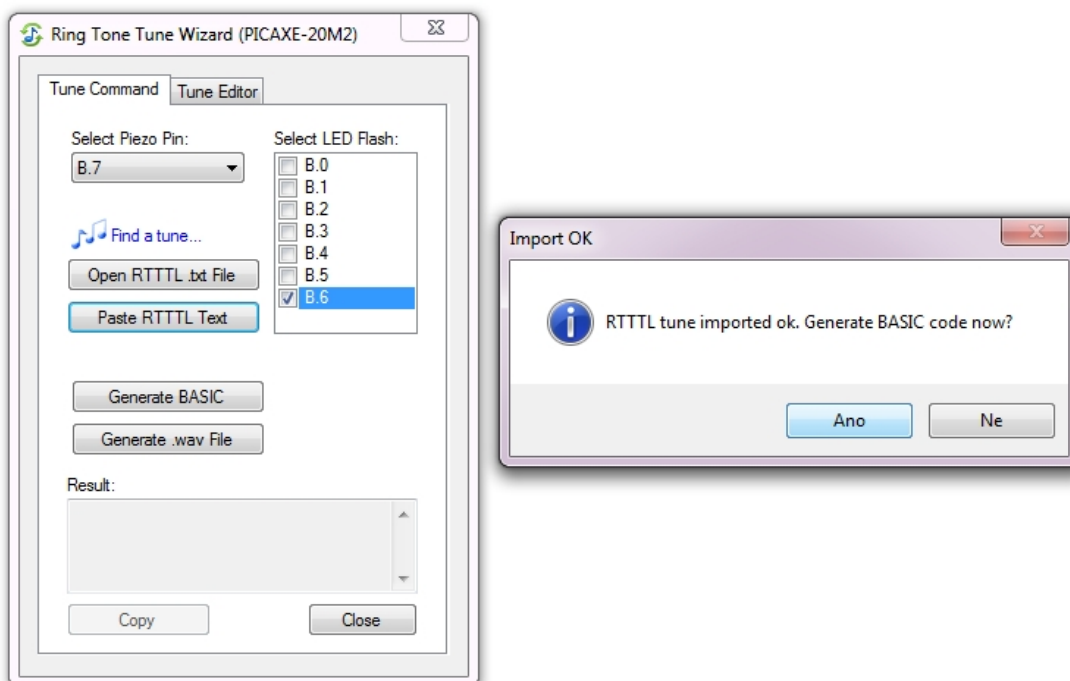
TUNE pin, speed, LED, (note, note, note...)

Pin - proměnná nebo konstanta, která určuje pin na kterém je připojen piezoměnič

Speed - proměnná nebo konstanta (1-15), určuje rychlost melodie

LED - proměnná nebo konstanta, která nastaví piny na daném portu, které budou blikat v rytmu např.: %00000010 –bliká led na pinu B.6 (nemusí být použito)

Pro generování příkazu použijeme program PICAXE Tune Wizard, který je součástí programu PICAXE Editor viz obrázek 42 (PICAXE>Wizard>Tune Wizard menu). Program využívá melodie ve formátu RTTTL pro generování vlastního kódu pro MCU.



Obrázek 42 – Program Tune Wizard

Zvolíme pin s připojeným piezo reproduktorem (B.7) a piny s připojenými LED diodami (B.6 ale nemusí být použito). Za předpokladu, že máme ve schránce okopírovanou melodii volíme: Paste RTTTL text viz obrázek 42. Výsledný kód v jazyku basic nalezneme v poli Result. Kód okopírujeme a vložíme do vlastního programu.

7.2.1 Ukázkový program

```

;Program tune_0.bas
main:

if pinC.7 = 1 then

tune B.7, 4, %01000000, ($73, $74, $FC, $76, $77, $FC, $73, $74, $7C, $76, $77, $7C,
$40, $7B, $7C, $73, $74, $7C, $7B, $FA, $FC, $79, $77, $74, $72, $F4)
;po stisknutí tlačítka BTN0 přehraje melodii Pink panther

endif

if pinC.6 = 1 then

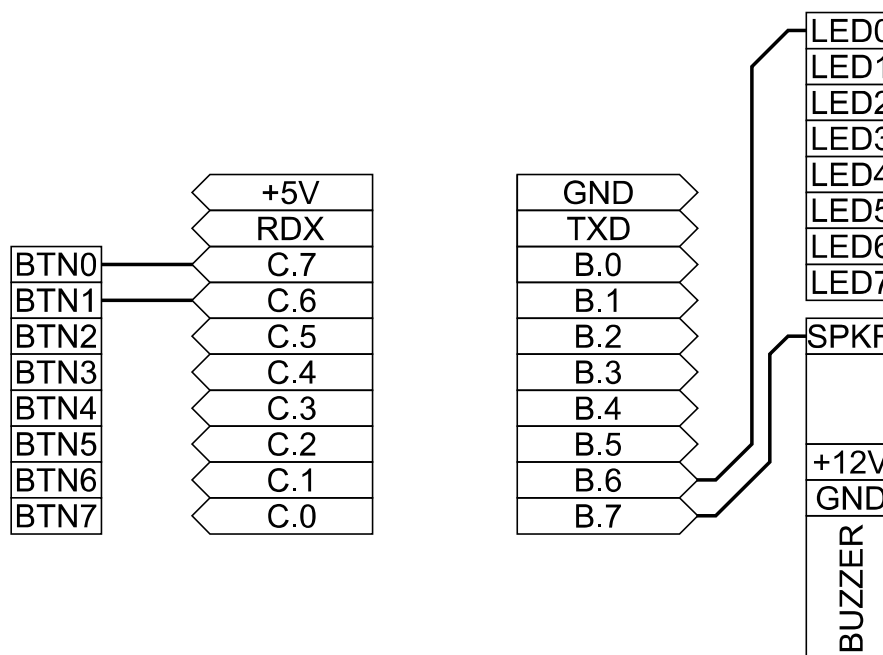
tune B.7, 6, %01000000, ($3C, $7C, $7B, $79, $3B, $34, $3C, $7C, $40, $7B, $40,
$7B, $39, $3C, $7C, $40, $7B, $00, $34, $3C, $7C, $79, $77, $79, $77, $76, $79, $37,
$76, $77, $39, $77, $79, $7B, $79, $77, $76, $34, $00, $FB, $7B, $40, $7B, $79, $BB)
; po stisknutí tlačítka BTN1 přehraje melodii Final countdown

endif

goto main

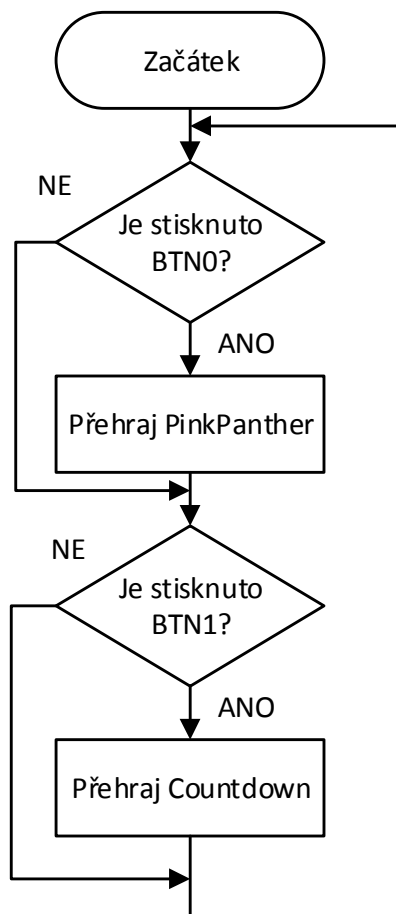
```

7.2.2 Schéma propojení



Obrázek 43 – Schéma propojení tune_0.bas

7.2.3 Vývojový diagram



Obrázek 44 – Vývojový diagram programu tune_0.bas

7.2.4 Náměty na rozšíření programu

1. Změna frekvence na 8 MHz pomocí příkazu **setfreq m8** nebo na 16 MHz příkazem **setfreq m16** pro zkrácení délky tónů.
2. Vyhledání dalších melodií a jejich vložení do programu.
3. Změna konstanty speed.

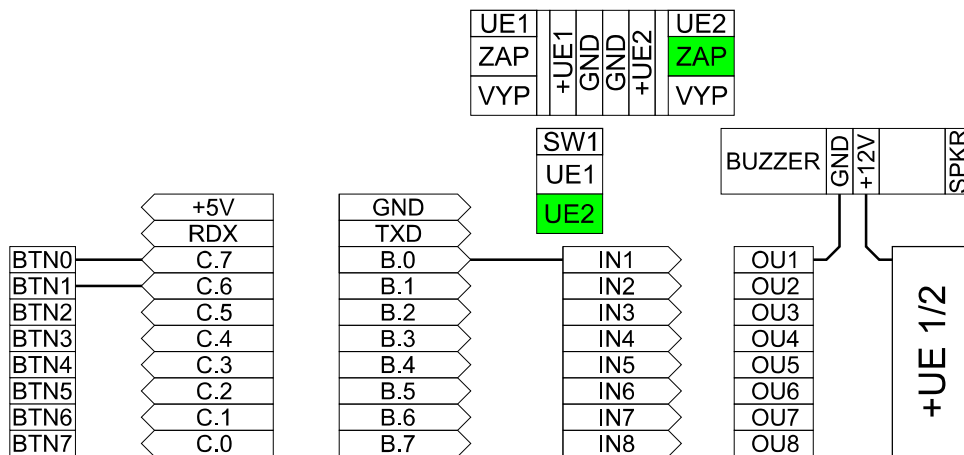
7.3 Bzučák

Bzučák můžeme použít jako celkem hlasitou sirénu. Bohužel není možné u bzučáku měnit tón, takže po přivedení napájecího napětí bzučí stále stejně. Můžeme ale určit délku tónu.

7.3.1 Ukázkový program

```
;program buzzer.bas
main:
if pinC.7 = 1 then
  high B.0
  pause 100
  low B.0
  pause 200 ;při stisku BTN0 bzučák vytvoří zvukový signál
end if
if pinC.6 = 1 then
  high B.0
  pause 400
  low B.0
  pause 50
  high B.0
  pause 400
  low B.0
  pause 500 ;při stisku BTN1 bzučák vytvoří jiný zvukový signál
end if
goto main
```

7.3.2 Schéma propojení

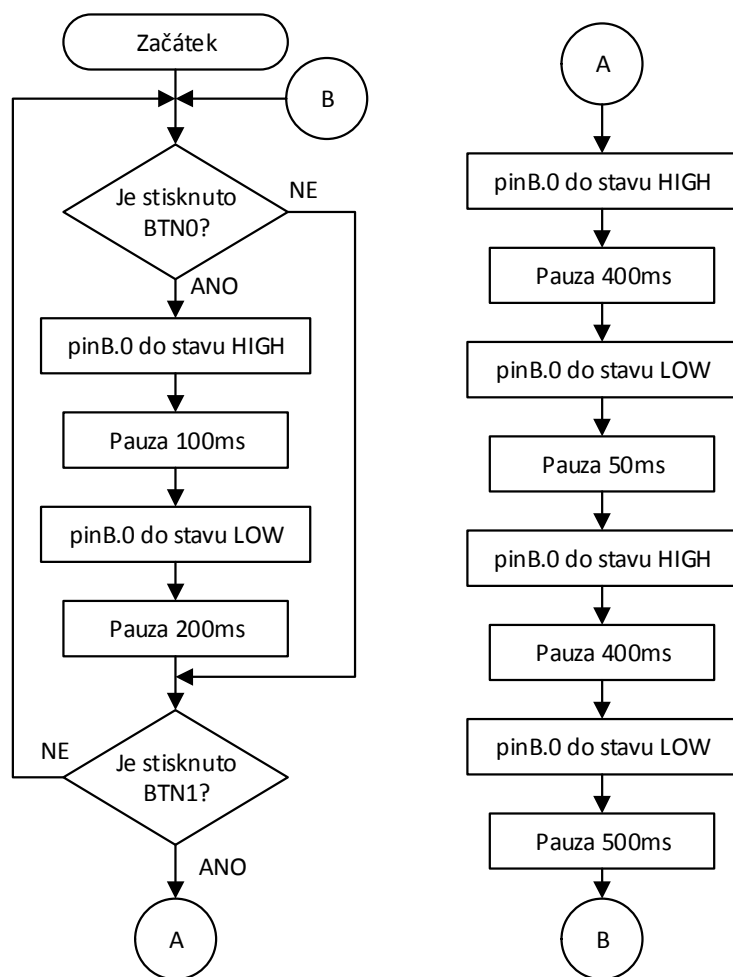


Obrázek 45 – Schéma propojení buzzer.bas

7.3.3 Náměty na rozšíření programu

1. Tvorba dalších zvukových znamení nebo efektů.

7.3.4 Vývojový digram



Obrázek 46 – Vývojový diagram programu buzzer.bas

8 Motorické úlohy

V následující kapitole uvedeme vzorové úlohy pro práci s moduly:

1. Modul Servo
2. Modul Motor a modul H-můstek
3. Modul Krokový motor

8.1 Ovládání serva

Pro ovládání serva jsou k dispozici speciální instrukce SERVO a SERVOPOS. Syntaxe příkazů je následující. Popis syntaxí čerpá z manuálu [18].

SERVO pin, pulse

Pin - konstanta označující pin na kterém je připojeno servo (v našem případě piny B.0 až B.7).

Pulse – proměnná nebo konstanta určující pozici serva. Doporučené hodnoty jsou 75-225, v našem případě 65-235. Před prvním použitím serva je nutné začít s hodnotami doporučenými a pokusem zmenšovat/zvětšovat hodnotu. Při moc malé nebo velké hodnotě dojde k přeskočení serva za dorazy a je nutné hodnoty zvětšit/zmenšit.

Pulzy jsou generovány na pozadí nezávisle na běhu programu. Generování pulzů lze ukončit příkazem SERVO pin, OFF. Generování je také zastaveno příkazy HIGH nebo LOW pro daný pin. Příkaz je určen pro inicializaci, dále v programu se používá příkaz SERVOPOS.

SERVOPOS pin, pulse

- Parametry příkazu jsou shodné s příkazem SERVO

8.1.1 Ukázkový program

Program po spuštění otestuje servo – nastaví ho do minimální krajní polohy, poté do maximální krajní polohy a na konci testovací části servo nastaví do výchozí minimální krajní polohy. Po otestování serva umožňuje pohyb servem tlačítka BTN0 a BTN1.

```

;program servo.bas
symbol servomin = 65
symbol servomax = 235
symbol pozice = b0
;deklarace proměnných
servo B.5,servomin ;inicializace serva, nastavení do minimální krajní
polohy
pause 500 ;pauza 500ms
servopos B.5, servomax ;přesun serva do maximální krajní polohy
pause 500
servopos B.5, servomin ;přesun serva do minimální krajní polohy
let pozice = servomin ;přiřazení hodnoty servomin proměnné pozice
;konec testovací sekvence, následuje nekonečná smyčka
main:

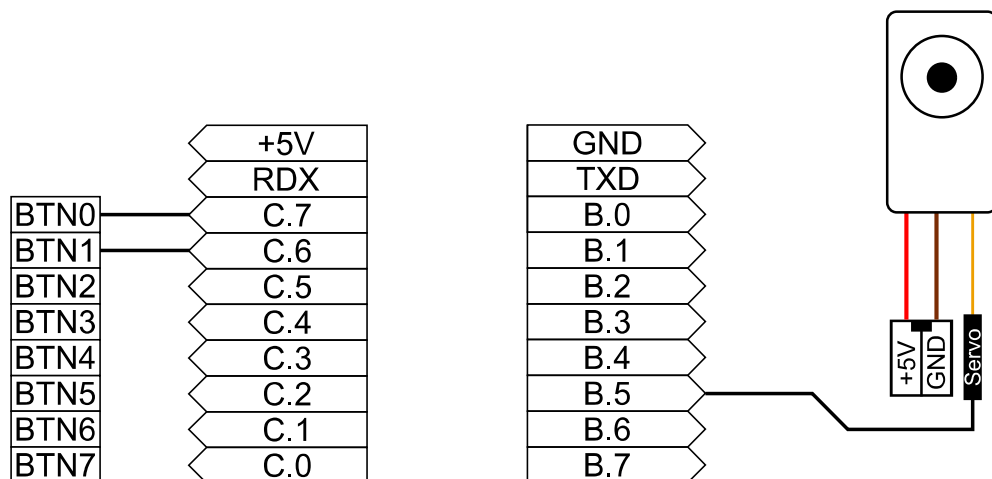
if pinC.7 = 1 then ;podmínka IF pro tlačítko BTN0
    inc pozice ;inkrementuje proměnnou pozice
    servopos B.5,pozice ;nastaví servo do pozice odpovídající
proměnné pozice
    pause 1
    if pozice >= servomax then ;podmínka pro udržení proměnné pozice
v hodnotě mezi 65 až 235
        pozice = servomax
    endif
end if

if pinC.6 = 1 then
    dec pozice
    servopos B.5,pozice
    pause 1
    if pozice <= servomin then
        pozice = servomin
    endif
end if

goto main

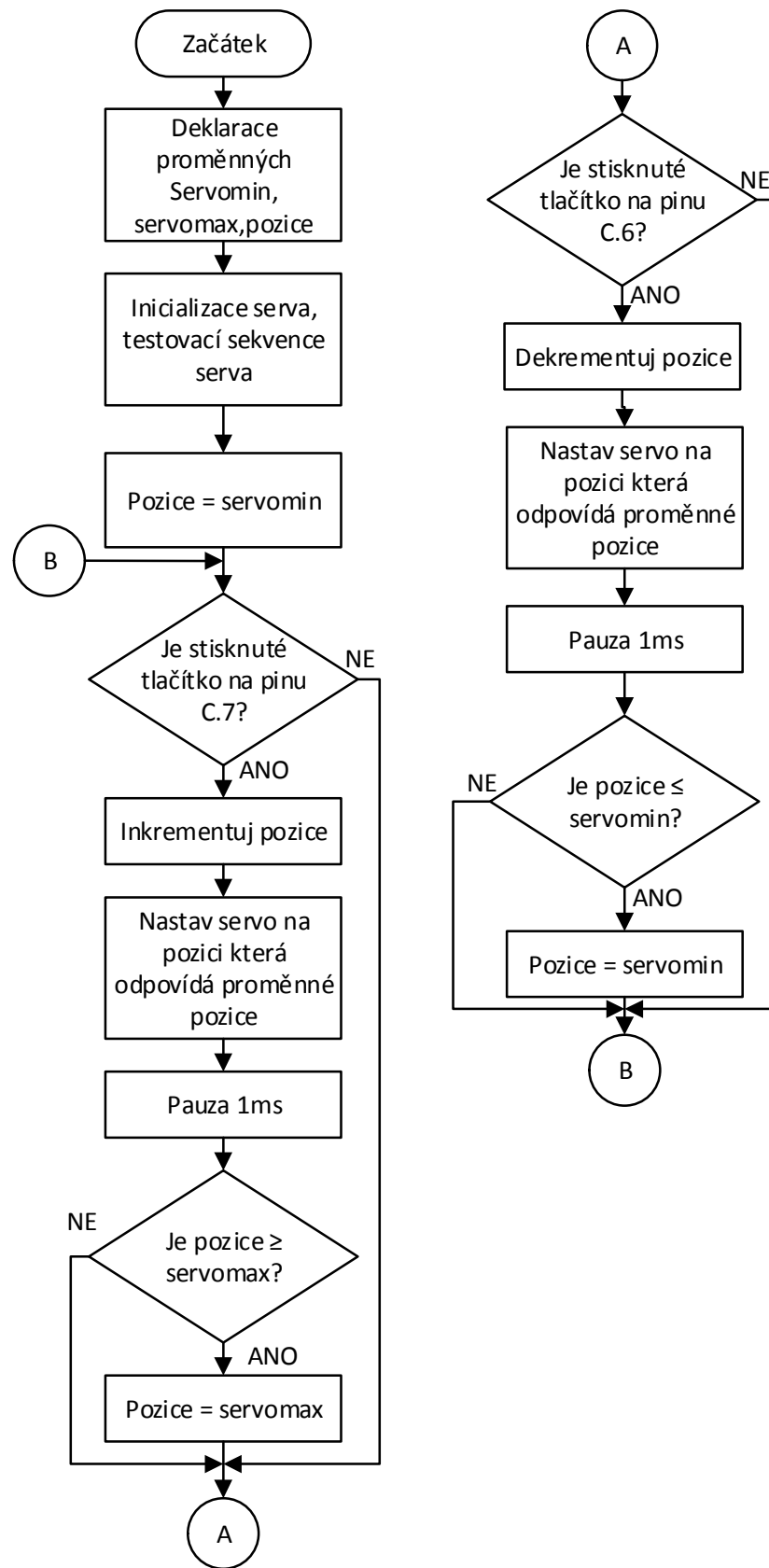
```

8.1.2 Schéma propojení



Obrázek 47 – Schéma propojení servo.bas

8.1.3 Vývojový diagram



Obrázek 48 – Vývojový diagram programu servo.bas

8.1.4 Náměty na rozšíření programu

1. Připojení dalších tlačítek a definování jednotlivých poloh serva pro každé z tlačítek.
2. Odebrání modulu Tlačítka, přidání modulu Potenciometry. Úprava programu tak, aby se servo nastavovalo do polohy, která je určena natočením potenciometru.

8.2 Reverzace chodu motoru

V následující úloze si popíšeme velmi jednoduchý způsob, jak lze měnit směr otáčení motoru pomocí modulu H-můstek. Pro práci s modulem není potřeba žádných speciálních instrukcí, vystačíme si pouze s pravdivostní tabulkou modulu H-můstek na tabulce 2. Program při stisku tlačítka BTN0 roztočí motor po směru hodinových ručiček. Při stisku BTN1 se motor točí proti směru hodinových ručiček. Při stisku tlačítka BTN2 se motor přestane točit, po dalším stisku se točí původním směrem.

8.2.1 Ukázkový program

```
;program H_bridge.bas

main:

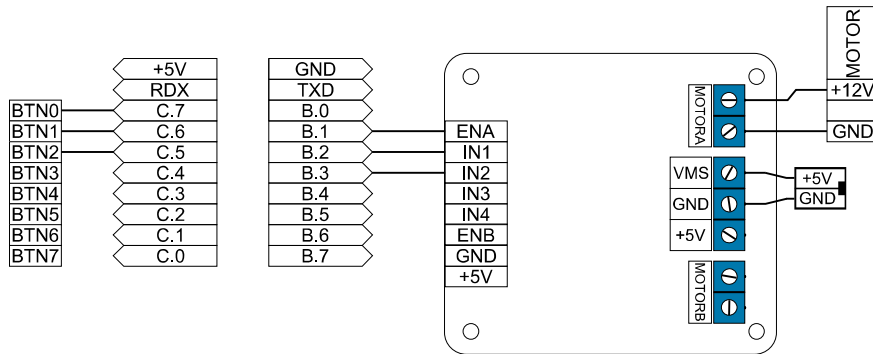
if pinC.7 = 1 and pinC.6 != 1 then
    let pinsb = %00001010 ;motor se točí po směru hodinových ručiček
endif

if pinC.6 = 1 and pinC.7 != 1 then
    let pinsb = %00000110 ; ;motor se točí proti směru hodinových
ručiček
endif

if pinC.5 = 1 then
    toggle B.1 ;Využívá vstupu ENA, při stisku se motor přestane
točit, při dalším stisku se začne točit původním směrem
    pause 200
endif

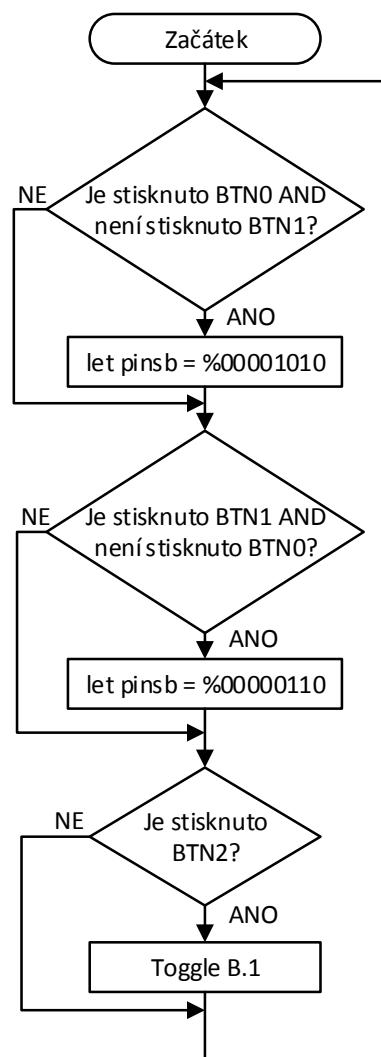
goto main
```

8.2.2 Schéma propojení



Obrázek 49- Schéma propojení H_bridge.bas

8.2.3 Vývojový diagram



Obrázek 50 – Vývojový digram programu H_bridge.bas

8.2.4 Náměty na rozšíření programu

1. PWM regulace motoru – využít ENA.
2. Motor před změnou směru zastaví, až poté změní směr.

8.3 Krokový motor

Úloha popisuje práci s unipolárním krokovým motorem. Program využívá jednu z hlavních výhod krokového motoru a to možnost přesného natočení hřídele bez zpětné vazby o její poloze. Řídící sekvence vychází z tabulky 1.

8.3.1 Ukázkový program

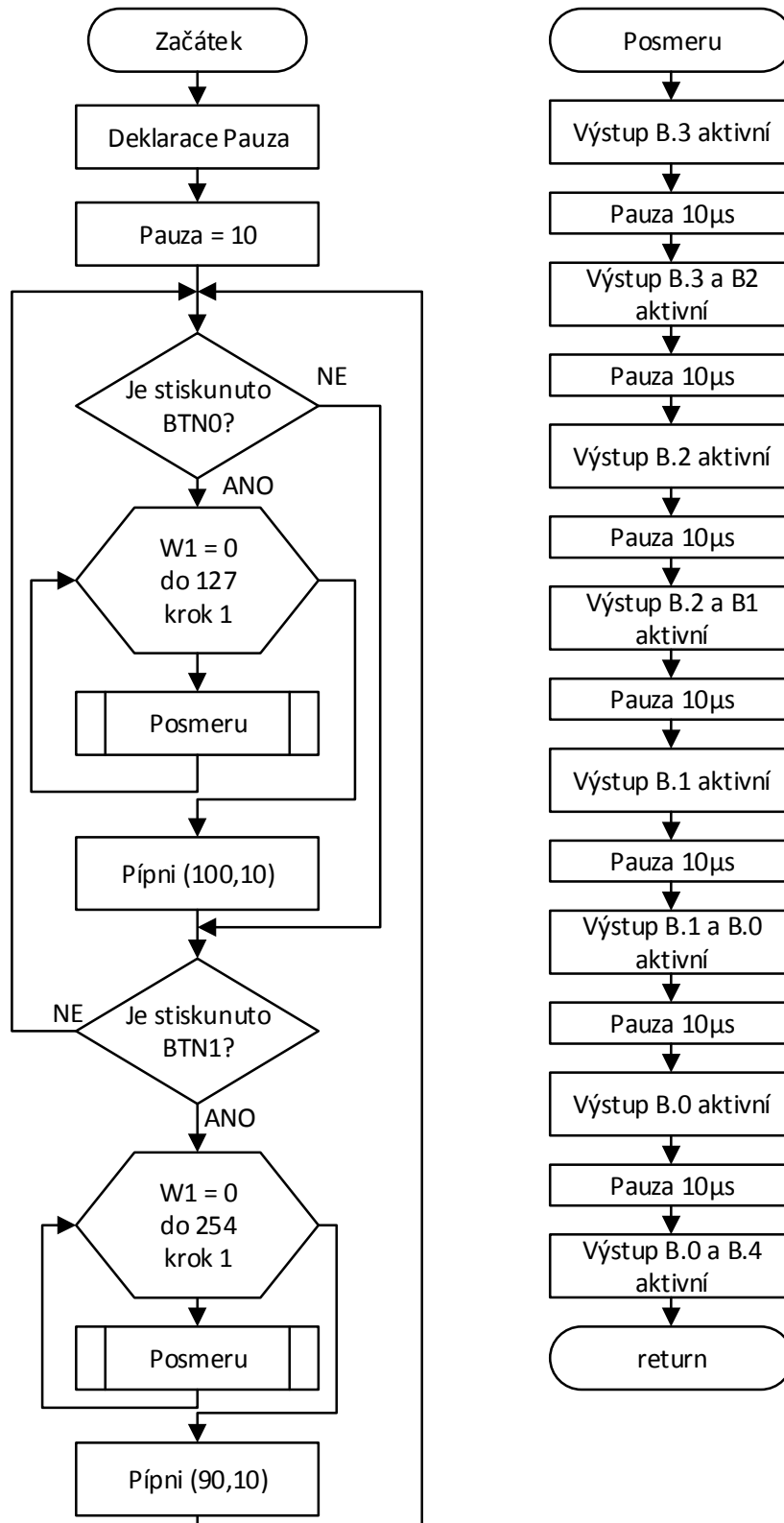
```
;program stepper.bas
dirsb = %11111111
symbol pauza = b0
let pauza = 10
main:
if pinC.7 = 1 then ;
    for w1=0 to 127 ;cyklus for, hřidel motoru se otočí o ¼ otáčky
        gosub Posmeru
    next w1
sound B.4, (100,10)
endif

if pinC.6 = 1 then
    for w1=0 to 254 ;cyklus for, hřidel motoru se otočí o ½ otáčky
        gosub Posmeru
    next w1
sound B.4, (90,10)
endif

goto main

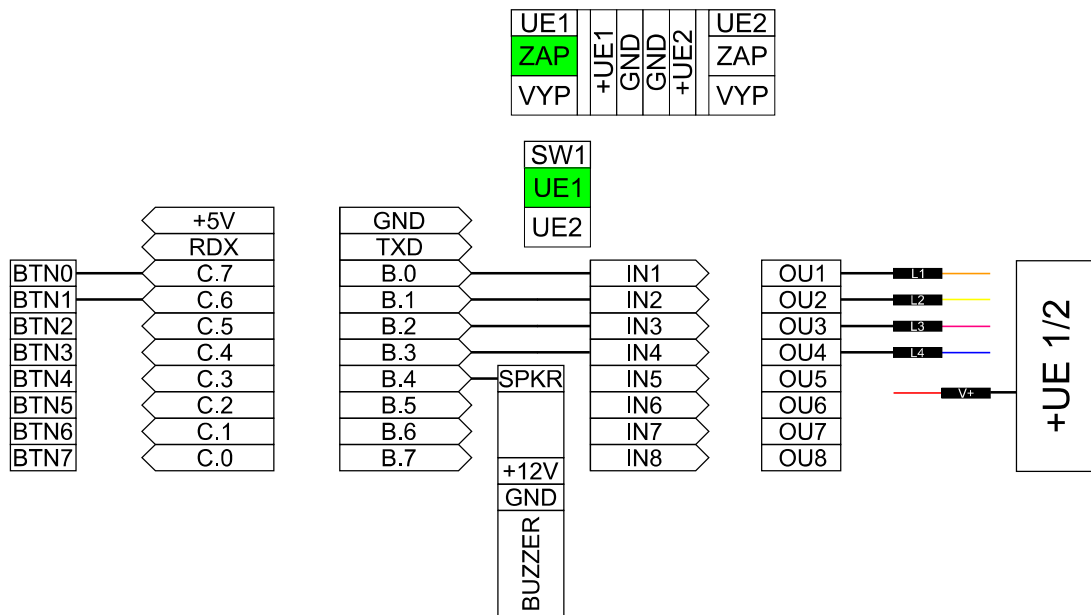
Posmeru: ;podprogram - řídící sekvence krokového motoru, směr otáčení
je po směru hodinových ručiček
let pinsb = %00001000
pauseus pauza ;pauza 10µs
let pinsb = %00001100
pauseus pauza
let pinsb = %00000100
pauseus pauza
let pinsb = %00000110
pauseus pauza
let pinsb = %00000010
pauseus pauza
let pinsb = %00000011
pauseus pauza
let pinsb = %00000001
pauseus pauza
let pinsb = %00001001
pauseus pauza
return ;návrát na příkaz který následuje po příkazu gosub Posmeru
```

8.3.2 Vývojový diagram



Obrázek 51 – Vývojový diagram programu stepper.bas

8.3.3 Schéma propojení



Obrázek 52 – Schéma propojení stepper.bas

8.3.4 Náměty na rozšíření programu

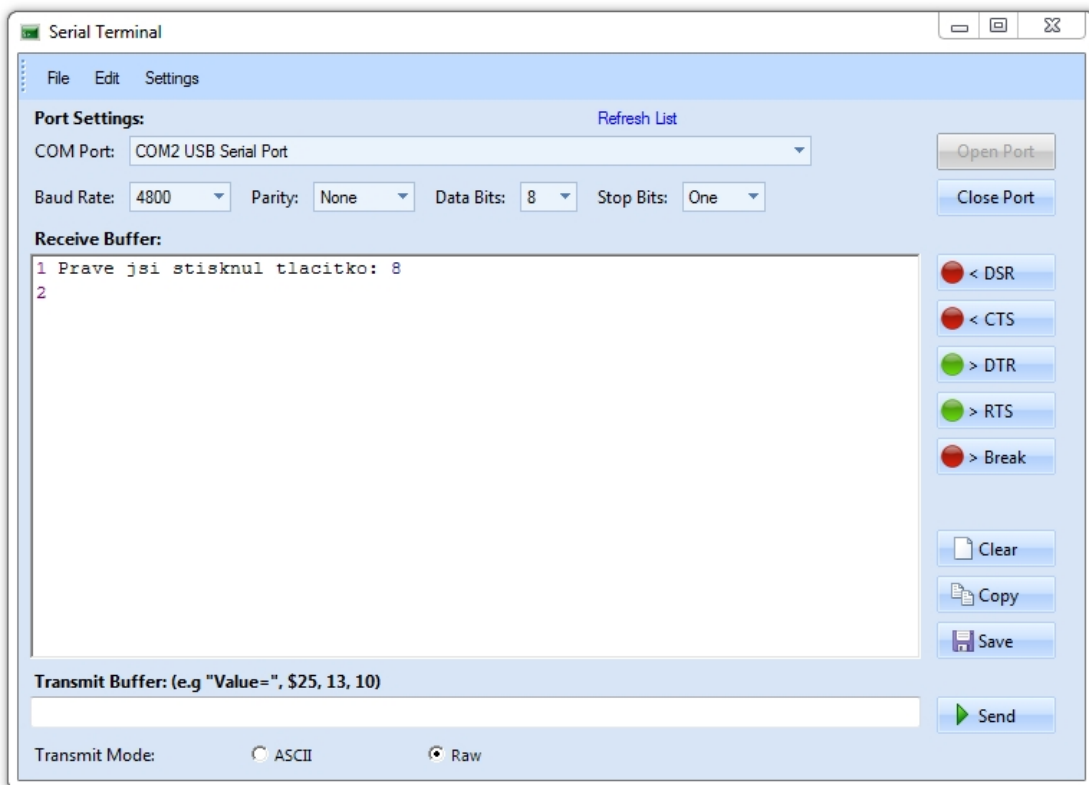
1. Otáčení motoru proti směru hodinových ručiček.
2. Přidání modulu potenciometry. Rychlost otáčení je závislá na poloze potenciometru.

9 Klávesnice a dotyková tlačítka

Závěrečná kapitola obsahuje ukázkové programy pro práci s maticovou klávesnicí a práci s dotykovými tlačítky.

9.1 Klávesnice komunikující s počítačem

Program obsluhuje modul Klávesnice, po stisknutí odešle přes programovací kabel hlášku: Právě jsi stisknul tlačítko: číslo tlačítka. Zobrazení hlášky probíhá v programu Serial Terminal (PICAXE > Terminal). Defaultní nastavení je na obrázku 53. Vývojová deska musí být pro zobrazení hlášky propojena s počítačem programovacím kabelem.



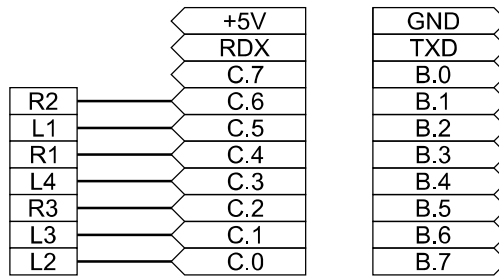
Obrázek 53 – Serial Terminal

9.1.1 Ukázkový program

```
;program keyboard.bas
symbol pom = b1
let dirsC = %10101011
main:
let pom = 1
let pinsC = %00100000
;program prochází 1. řadu (123)
gosub sloupec
let pom = 4
let pinsC = %00000001
;program prochází 2. řadu (456)
gosub sloupec
let pom= 7
let pinsC = %00000010
;program prochází 3. řadu (789)
gosub sloupec
let pom = 10
let pinsC = %00001000
;program prochází 4. řadu (*0#)
gosub sloupec

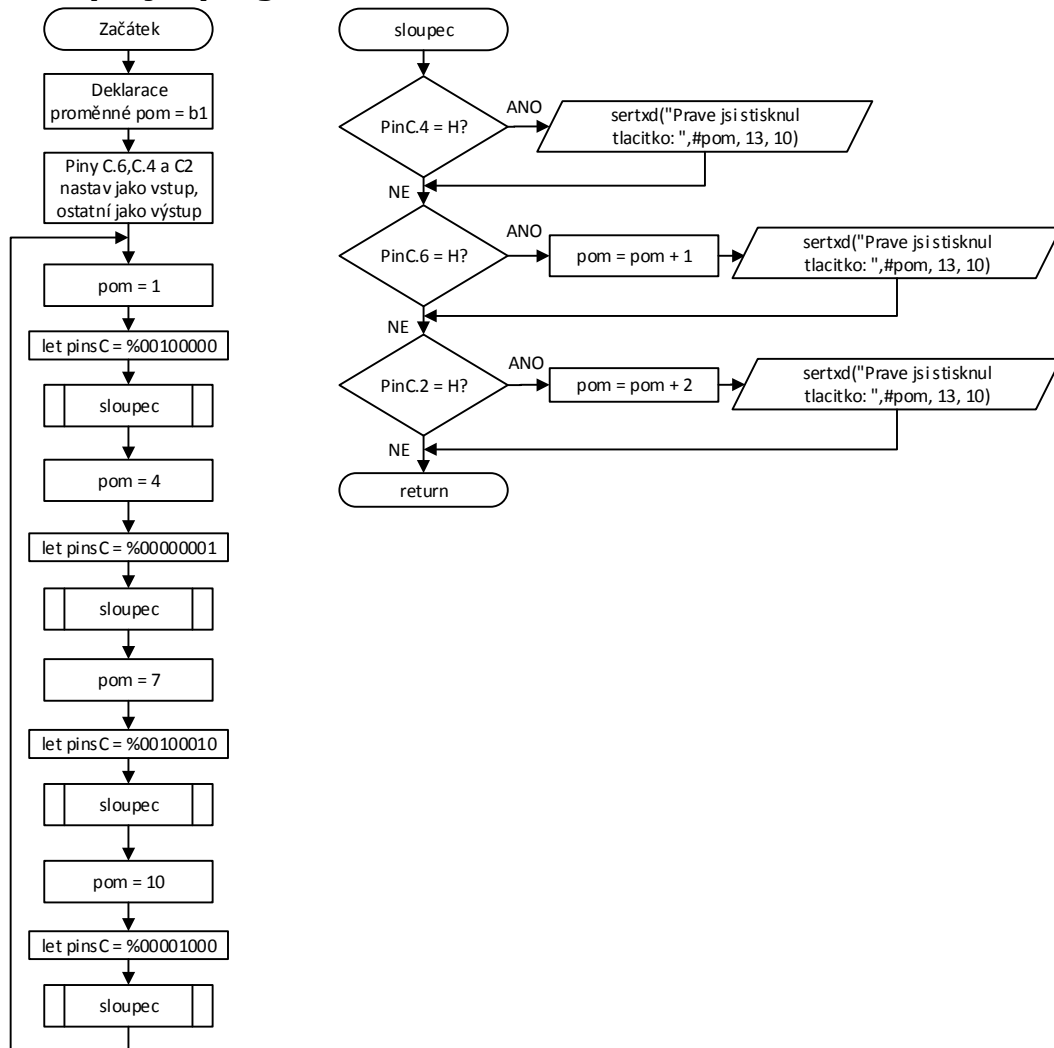
goto main
sloupec:
    if pinC.4 = 1 then ;program prochází 1. sloupec (147*)
        sertxd("Prave jsi stisknul tlacitko: ",#b1, 13, 10)
    endif
    if pinC.6 = 1 then ;program prochází 2. sloupec (2580)
        let pom = pom + 1
        sertxd("Prave jsi stisknul tlacitko: ",#b1, 13, 10)
    endif
    if pinC.2 = 1 then ;program prochází 3. sloupec (369#)
        let pom = pom + 2
        sertxd("Prave jsi stisknul tlacitko: ",#b1, 13, 10)
    endif
return
```


9.1.2 Schéma propojení



Obrázek 54 – Schéma propojení keyboard.bas

9.1.3 Vývojový digram



Obrázek 55 – Vývojový diagram programu keyboard.bas

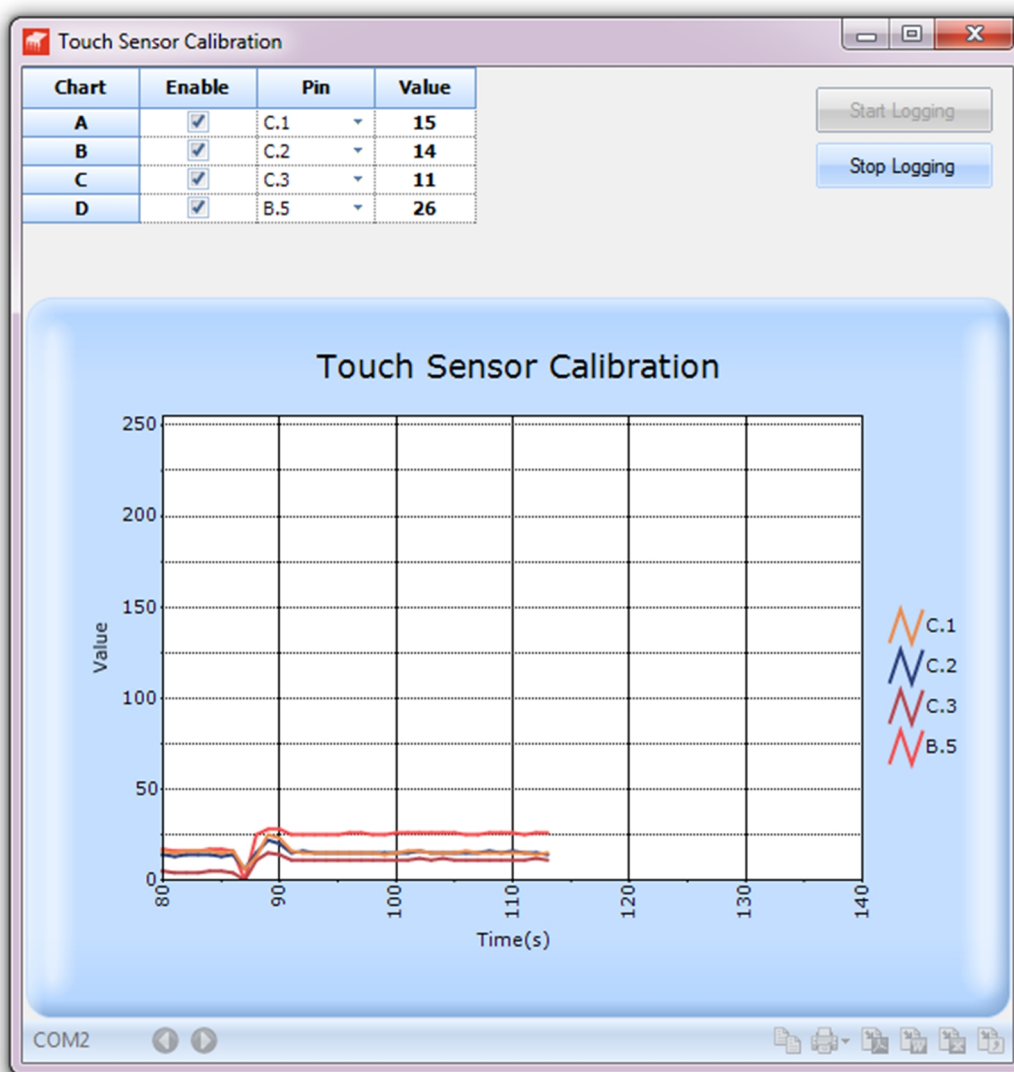
9.1.4 Náměty na další rozšíření programu

1. Přidání modulu 7 segment, na modulu se bude zobrazovat číslice podle stisknutého tlačítka.
2. Upravit program aby při stisku tlačítek *0# nezobrazoval čísla 10, 11, 12, ale zobrazovaly se jejich znaky.

9.2 Dotykový vypínač

Program využívá speciálního příkazu TOUCH, určeného pro práci s dotykovými tlačítky. Po dotyku plošky rozsvítí příslušnou LED0 až LED4, při dalším dotyku LED zhasne. V případě nefunkčnosti programu je nutné provést kalibraci jednotlivých tlačítek. Ukázkový program vychází z kalibrace provedené při psaní a testování programu, ale hodnoty se mohou při každém dalším zapojení úlohy nepatrně měnit.

Pro vlastní kalibraci je vhodné použít aplikaci Touch Sensor Calibration obsaženou v programu (PICAXE > touch). V aplikaci nastavíme jednotlivé piny, na kterých jsou připojena dotyková tlačítka a spustíme kalibraci viz obrázek 56.



Obrázek 56 – Touch Sensor Calibration

V tomto stádiu jsou všechna tlačítka volná bez jakéhokoliv doteku. V levém horním rohu je tabulka s jednotlivými piny. Pro kalibraci je stěžejní sloupec Value kde nalezneme hodnotu. Hodnota se také vykresluje v grafu, ale pro přehlednost

budeme raději vycházet z tabulky. Je patrné, že bez doteku tlačítek je hodnota pro piny C.1-15, C.2-14, C.3-11, B.5-26.

Nyní přiložíme prst na první plošku připojenou na pinu C.1 a hodnota 15 se změní na hodnotu 102. Tuto hodnotu si zaznamenáme, jedná se o hodnotu kdy je vyhodnocen dotyk na plošce a je důležitá pro sestavení programu. Všimáme si i hodnoty vedlejší dotykové plošky (pin C.2), hodnota se také mění, pro vyhodnocení dotyku na pinu C.1 bude hodnota mezi 43 a 102 viz obrázek 57Obrázek 57.

Chart	Enable	Pin	Value
A	<input checked="" type="checkbox"/>	C.1	102
B	<input checked="" type="checkbox"/>	C.2	43
C	<input checked="" type="checkbox"/>	C.3	19
D	<input checked="" type="checkbox"/>	B.5	26

Obrázek 57 – Kalibrace plošek

Analogicky pokračujeme s kalibrací všech použitých dotekových plošek a zaznamenáváme si hodnoty. Výsledná použitá hodnota v programu bude přibližně mezi dvěma hodnotami, pro pin C.2 tedy přibližně 72.

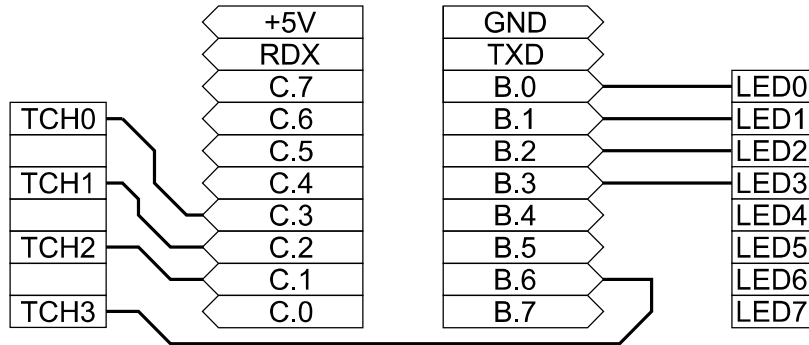
9.2.1 Ukázkový program

```
;program touch.bas
Main:
  Touch C.1,b0 ;čte hodnotu dotykové plošky na pinu C.1 a uloží ji
do proměnné b0
  Touch C.2,b1
  Touch C.3,b2
  Touch B.6,b3
  if b2 > 72 then
    toggle B.0
    pause 50
  end if

  if b1 > 50 then
    toggle B.1
    pause 50
  end if

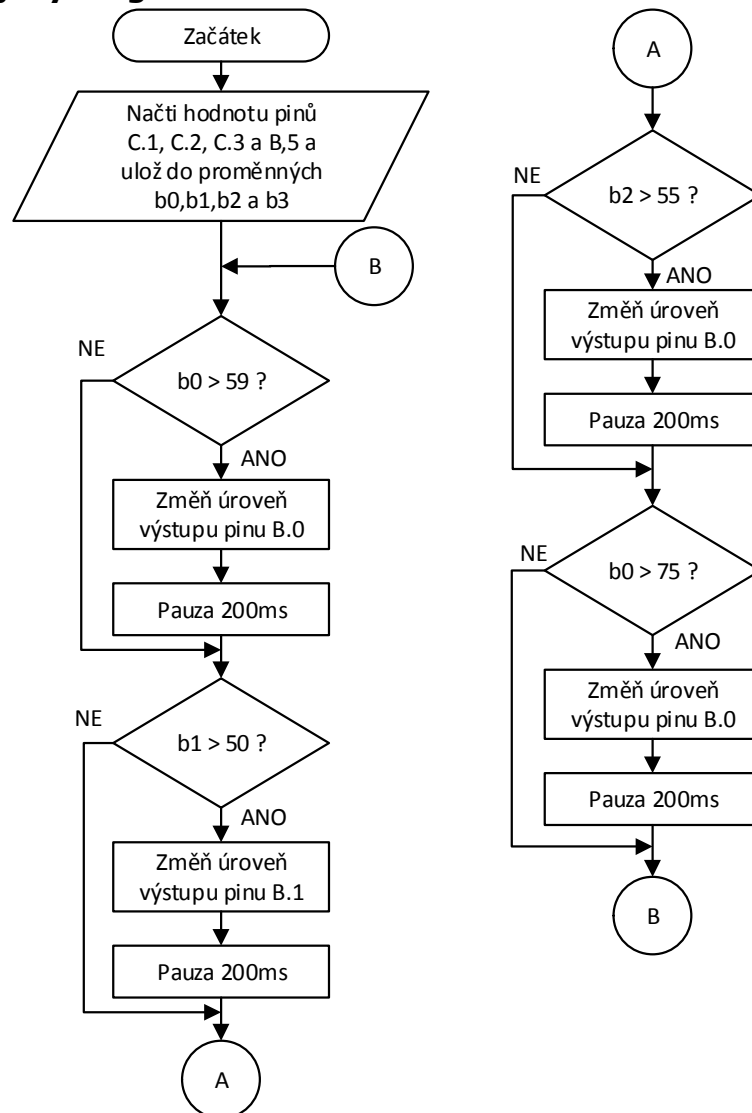
  if b0 > 50 then
    toggle B.2
    pause 50
  end if
  if b3 > 53 then
    toggle B.3
    pause 50
  end if
goto Main
```

9.2.2 Schéma propojení



Obrázek 58 – Schéma propojení touch.bas

9.2.3 Vývojový diagram



Obrázek 59 – Vývojový diagram programu touch.bas

9.2.4 Náměty na rozšíření programu

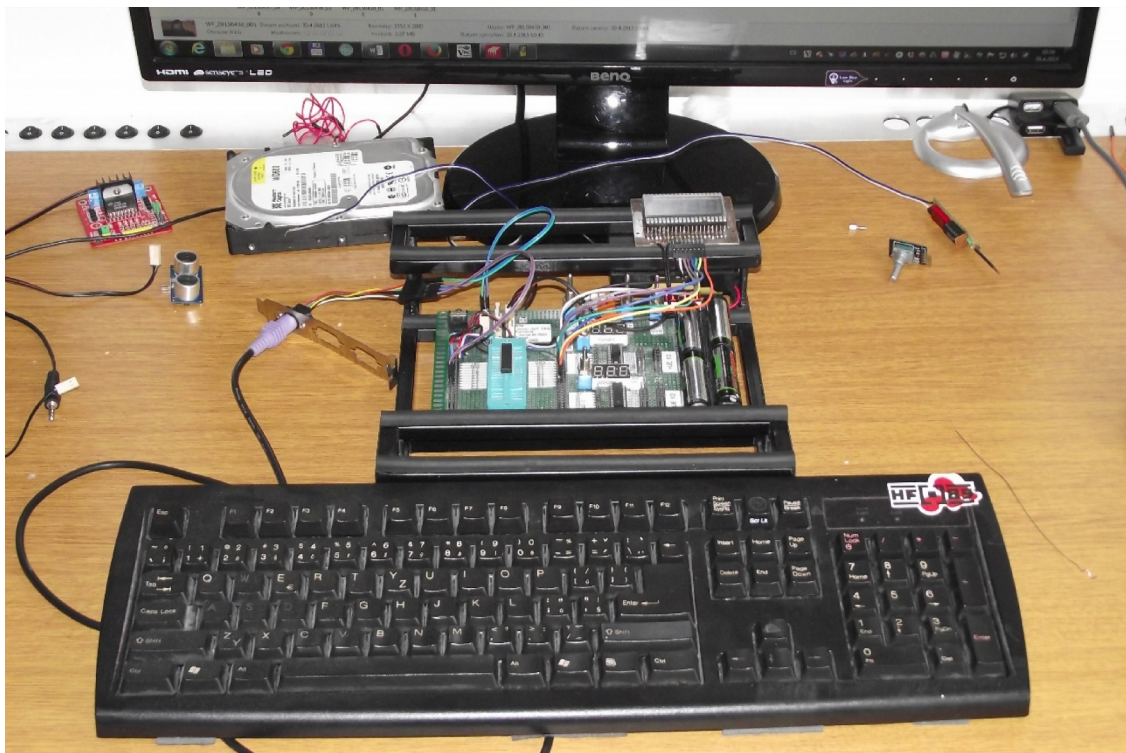
1. Při stisku dvou krajních plošek zhasnou všechny LED.
2. Rozšíření programu o další funkce po stisknutí některé z dotykových plošek.

Shrnutí vlastností stavebnice

Vyrobená výuková stavebnice spolu se stolním držákem, je naprosto univerzálním vývojovým prostředkem použitelným nejen ve výuce. Díky vhodně navržené vývojové desce je plně kompatibilní se všemi aktuálně dostupnými typy MCU PICAXE.

Díky akumulátorovému napájení je celá stavebnice maximálně mobilní v případě, kdy vytvářený program a zapojení potřebujeme otestovat například venku. Dále dává možnost stavebnici použít i jako názornou didaktickou pomůcku v některých teoretických vyučovacích hodinách. V případě náhlého vybití akumulátorů je možné vývojovou deskou napájet i ze stabilizovaného zdroje nebo nouzově i z rozhraní USB, za předpokladu připojení napájecího napětí na některý z konektorů napájecí 5V části. Životnost používaných akumulátorů o kapacitě 2100 mAh je však poměrně velká, při občasných pokusech byla jejich životnost přibližně 2 měsíce.

Moduly zastoupené v této práci představují základní a typické druhy zapojení a demonstrují práci s nimi. Tím možnosti ani zdaleka nekončí, stavebnici je možné rozšířit mnoha dalšími zapojeními a stavebnici si tak přizpůsobit pro své účely viz obrázek 60.



Obrázek 60 – Stavebnice s modulem 4DM8010 a PS2 klávesnicí

Závěr

Cílem teoretické části práce bylo uvést základní informace o mikrokontrolérech PICAXE, ale i o alternativách. Jsou zmíněny opravdu pouze základní informace o platformě Arduino a Raspberry Pi jakožto vybrané alternativy. O mikrokontrolérech PICAXE jsou zmíněny také základní informace. Nejdůležitější informace jsou zmíněny i přímo o typu 20M2 osazeného na vývojové desce. Jako možnosti konstrukce stavebnice jsem vybral čtyři druhy s kterými mám osobní a praktické zkušenosti.

Cílem praktické části bakalářské práce bylo vytvoření univerzální výukové stavebnice s MCU PICAXE, rozšiřujících modulů a ukázkových úloh pro každý z modulů. Výsledkem práce je plně funkční univerzální stavebnice a sbírka úloh s funkčními programy umístěnými v práci.

Při stavbě bylo řešeno několik problémů, hlavně s levnými a většinou nepříliš kvalitně zpracovanými rozšiřujícími moduly, které jsem zakoupil na ebay.com. Původní myšlenka byla více začlenit tyto hotové moduly do stavebnice, ale ukázaly se jako velice nespolehlivé, proto bylo tedy od této myšlenky upuštěno a byl použit pouze jeden funkční a naprosto spolehlivý.

Chybným řešením byla také stavba první verze stavebnice na kontaktním nepájivém poli, kdy následné zapojování vstupních a výstupních obvodů bylo velice zdouhavé a při větším rozsahu i náročné na pozornost, navíc kladlo na uživatele znalost čtení elektrotechnických schémat.

Předpokládám další rozvoj stavebnice včetně návrhu plošného spoje vývojové desky a pokračování v diplomové práci na některé více konkrétní téma.

Seznam zkratek

1-Wire	rozhraní s jedním datovým vodičem
A/D	Analog/digital
ATX	(Advanced Technology Extended) – formát základní desky počítače
D/A	Digital/analog
DIP	(Dual In-Line Package) – pouzdro integrovaných obvodů
EEPROM	(Electrically Erasable Programmable Read-Only Memory) – elektricky mazatelná paměť
GND	(Ground) – zem, 0V
GPIO	(General Purpose Input/Output) – piny programovatelné pomocí software
I2C	(Inter-Integrated Circuit) – sériová sběrnice master/slave
PLC	(Programmable Logic Controller) – průmyslový počítač používaný v automatizaci
PWM	(Pulse Width Modulation) – pulzně šířková modulace
RS232	sériová linka
RTTTL	(Ring Tone Text Transfer Language) – jazyk pro tvorbu vyzváněcích melodií mobilních telefonů Nokia
SMD	(Surface mount device) – součástky pro povrchovou montáž
SPI	(Serial Peripheral Interface) – synchronní sběrnice pro sériovou komunikaci
SSR	(Solid State Relay) – polovodičové relé
THT	(Through-hole technology) – technologie montáže součástek
TTL	(Transistor-transistor-logic) – tranzistorově-tranzistorová logika
UART	(Universal asynchronous receiver/transmitter) – univerzální asynchronní přijímač/vysílač
UDFN	pouzdro integrovaných obvodů

Citovaná literatura

- [1] VALÁŠEK, Pavel a Roman LOSKOT. *Mikrokontrolery*. Vyd. 1. Hradec Králové: Gaudeamus, 1997, 63 s. ISBN 80-704-1446-4.
- [2] PECHAL, Stanislav. *Monolitické mikropočítače*. 1. vydání. Praha: BEN - technická literatura, 1995, 269 s. ISBN 80-860-5630-9.
- [3] ROBENEK, Jan. Nejmenší pouzdro pro mikrokontroléry s Flash. In: *hw.cz* [online]. 2010 [cit. 2015-04-05]. Dostupné z: <http://www.hw.cz/prumysl/nejmensi-pouzdro-pro-mikrokontrolery-s-flash.html>
- [4] TIŠNOVSKÝ, Pavel. Jak pracuje počítač?. In: *ROOT.CZ Informace nejen ze světa Linuxu* [online]. 2008 [cit. 2015-04-04]. Dostupné z: <http://www.root.cz/clanky/jak-pracuje-pocitac/#k02>
- [5] Introduction. In: *Arduino* [online]. ©2015 [cit. 2015-04-06]. Dostupné z: <http://arduino.cc/en/guide/introduction>
- [6] Bootloader Development. In: *Arduino* [online]. ©2015 [cit. 2015-04-07]. Dostupné z: <http://arduino.cc/en/Hacking/Bootloader?from=Tutorial.Bootloader>
- [7] What Is PICAXE?. In: *PICAXE* [online]. 2015 [cit. 2015-04-07]. Dostupné z: <http://www.picaxe.com/What-is-PICAXE/>
- [8] Compare board specs. In: *Arduino* [online]. ©2015 [cit. 2015-04-07]. Dostupné z: <http://arduino.cc/en/Products.Compare>
- [9] Bluetooth Shield for Arduino. In: *Robotshop* [online]. ©2015 [cit. 2015-04-07]. Dostupné z: <http://www.robotshop.com/ca/en/bluetooth-shield-arduino-master-slave.html>
- [10] Arduino - Products. In: *Arduino* [online]. © 2015 [cit. 2015-04-02]. Dostupné z: <http://arduino.cc/en/Main/Products>
- [11] Arduino: jak pro něj začít programovat. In: MALÝ, Martin. *root.cz* [online]. 2010 [cit. 2015-04-07]. Dostupné z: <http://www.root.cz/clanky/arduino-jak-pro-nej-zacit-programovat/>

- [12] Raspberry PI 2 model B. In: *Raspberrypi.org* [online]. 2015 [cit. 2015-04-10]. Dostupné z: <https://www.raspberrypi.org/products/raspberrypi-2-model-b/>
- [13] GPIO: Raspberry PI models A and B. In: *raspberrypi.org* [online]. 2015 [cit. 2015-04-10]. Dostupné z: <https://www.raspberrypi.org/documentation/usage/gpio/>
- [14] *Gertboard User Manual* [online]. Revision 2.0. 2012 [cit. 10.4.2015]. Dostupné z: http://www.element14.com/community/servlet/JiveServlet/previewBody/51727-102-1-265829/Gertboard_UM_with_python.pdf
- [15] UniPi. In: *UniPi* [online]. 2015 [cit. 2015-04-10]. Dostupné z: <http://unipi.technology/product/unipi/>
- [16] Python. In: *raspberrypi.org* [online]. 2015 [cit. 2015-04-10]. Dostupné z: <https://www.raspberrypi.org/documentation/usage/python/>
- [17] *PICAXE Manual: getting started* [online]. 2013, 110 s. [cit. 21.4.2015]. Dostupné z: http://www.picaxe.com/docs/picaxe_manual1.pdf
- [18] *PICAXE manual: basic commands* [online]. 2013, 271 s. [cit. 20.4.2015]. Dostupné z: http://www.picaxe.com/docs/picaxe_manual2.pdf
- [19] *PICAXE manual: Microcontroller interfacing circuits* [online]. 2010, 46 s. [cit. 20.4.2015]. Dostupné z: http://www.picaxe.com/docs/picaxe_manual3.pdf
- [20] MERKUR Robotika a mechatronika. In: *MERKUR* [online]. © 2015 [cit. 2015-04-02]. Dostupné z: <http://www.merkurtoys.cz/vyrobyky/robotika-a-mechatronika>
- [21] Hardware. In: *PICAXE* [online]. 2015 [cit. 2015-04-02]. Dostupné z: <http://www.picaxe.com/Hardware>
- [22] Univerzální a zkušební desky. In: *GES electronics* [online]. © 1991–2015 [cit. 2015-04-18]. Dostupné z: <http://www.ges.cz/cz/chemie-plosne-spoje/plosne-spoje/univerzalni-a-zkusebni-desky/SMAN.html>
- [23] GO-05 servo 5g. In: *Pelikan* [online]. 2015 [cit. 2015-04-21]. Dostupné z: <http://www.pelikandaniel.com/?sec=product&id=66572#1>

- [24] ŘEZÁČ, Kamil. Krokové motory. In: *robotika.cz* [online]. 2002 [cit. 2015-04-12]. Dostupné z: <http://robotika.cz/articles/steppers/cs>
- [25] *L298: DUAL FULL-BRIDGE DRIVER* [online]. ©2000, 13 s. [cit. 2015-04-13]. Dostupné z: <http://www.st.com/web/en/resource/technical/document/datasheet/CD00000240.pdf>
- [26] Arduino Micro. In: *Arduino* [online]. ©2015 [cit. 2015-04-02]. Dostupné z: <http://arduino.cc/en/Main/ArduinoBoardMicro>
- [27] Sidekick 3. In: *Raspberry pi python adventures* [online]. 2013 [cit. 2015-04-10]. Dostupné z: <http://raspberry-python.blogspot.cz/2013/01/sidekick-3.html>

Seznam obrázků

Obrázek 1 – Zjednodušené blokové schéma počítače	14
Obrázek 2 – von Neumannova koncepce	15
Obrázek 3 – Harvardská koncepce	16
Obrázek 4 – Arduino Uno s Bluetooth shieldem.....	19
Obrázek 5 – Arduino IDE 1.6.3 s firmware Repetier pro 3D tiskárnu Rebel2	20
Obrázek 6 – Arduino Mega 2560 se shieldem RAMPS v1.4	21
Obrázek 7 – Gertboard	23
Obrázek 8 – UniPi	24
Obrázek 9 – Prostředí IDLE	25
Obrázek 10 – Pinout MCU PICAXE 20M2	28
Obrázek 11 – PICAXE Editor 6.....	31
Obrázek 12 – Arduino Micro	33
Obrázek 13 – program EAGLE light a část návrhu desky Arduino micro	34
Obrázek 14 – Vývojová deska	37
Obrázek 15 – Schéma vývojové desky.....	39
Obrázek 16 – Modul Tlačítka	40
Obrázek 17 – Modul Klávesnice.....	41
Obrázek 18 – Modul Potenciometry.....	41
Obrázek 19 – Modul TOUCH.....	42
Obrázek 20 – Vodivé plošky	42
Obrázek 21 – Modul LED	43
Obrázek 22 – Modul 7 segment.....	44
Obrázek 23 – Modul LED strip.....	44
Obrázek 24 – Modul RGB LED strip.....	45
Obrázek 25 – Modul Motor	45
Obrázek 26 – Modul Servo	46
Obrázek 27 – Modul Krokový motor.....	47
Obrázek 28 – Modul H-můstek.....	48
Obrázek 29 – Modul Reproduktor	50
Obrázek 30 – Označení napájecího konektoru.....	51
Obrázek 31 – Programovací kabel	51

Obrázek 32 – Schéma propojení úlohy LED had.bas	53
Obrázek 33 – Vývojový diagram programu LED had.bas	53
Obrázek 34 – Schéma propojení programu cisla.bas.....	55
Obrázek 35 - Vývojový diagram programu cisla.bas	55
Obrázek 36 – Schéma propojení programu LEDstrip_PWM.bas	57
Obrázek 37 – Vývojový diagram programu LEDstrip_PWM.bas.....	57
Obrázek 38 – Schéma propojení RGBstrip_PWM.bas	58
Obrázek 39 – Vývojový diagram programu RGBstrip_PWM.bas	59
Obrázek 40 – Schéma propojení Sound_0.bas.....	61
Obrázek 41 – Vývojový diagram programu Sound_0.bas	61
Obrázek 42 – Program Tune Wizard.....	62
Obrázek 43 – Schéma propojení tune_0.bas	63
Obrázek 44 – Vývojový diagram programu tune_0.bas.....	64
Obrázek 45 – Schéma propojení buzzer.bas	65
Obrázek 46 – Vývojový diagram programu buzzer.bas.....	66
Obrázek 47 – Schéma propojení servo.bas.....	68
Obrázek 48 – Vývojový diagram programu servo.bas	69
Obrázek 49- Schéma propojení H_bridge.bas.....	71
Obrázek 50 – Vývojový digram programu H_bridge.bas.....	71
Obrázek 51 – Vývojový diagram programu stepper.bas.....	73
Obrázek 52 – Schéma propojení stepper.bas.....	74
Obrázek 53 – Serial Terminal	75
Obrázek 54 – Schéma propojení keyboard.bas.....	77
Obrázek 55 – Vývojový diagram programu keyboard.bas	77
Obrázek 56 – Touch Sensor Calibration	78
Obrázek 57 – Kalibrace plošek.....	79
Obrázek 58 – Schéma propojení touch.bas.....	80
Obrázek 59 – Vývojový diagram programu touch.bas	80
Obrázek 60 – Stavebnice s modulem 4DM8010 a PS2 klávesnicí	81
Obrázek 61 – Porovnání typů řady M.....	90
Obrázek 62 – Porovnání typů řady X.....	91
Obrázek 63 – Stolní držák	92
Obrázek 64 – Stolní držák s moduly a vývojovou deskou.....	92

Seznam tabulek

Tabulka 1 – Řídící sekvence krokového motoru	48
Tabulka 2 – Pravdivostní tabulka modulu H-můstek.....	49

Přílohy

Příloha A

Feature	PICAXE Command	08M2	18M2	18M2+	14M2	20M2
Voltage Range (V)		2.3-5.5	1.8-5.5	1.8-5.5	1.8-5.5	1.8-5.5
Memory Capacity (bytes)		2048	2048	2048	2048	2048
Parallel Tasks (starts)	resume, suspend	4	4	8	8	8
Max Internal Freq (MHz)	setfreq	32	32	32	32	32
Variables RAM (bytes)	peek, poke @bptr	128	256	512	512	512
Table data (bytes)	table, readtable, tablecopy	-	-	512	512	512
I2C master support	hi2cin, hi2cout, hi2csetup	Yes	Yes	Yes	Yes	Yes
Pwmout channels	pwmout	1	2	2	4	4
Hpwm support	hpwm	No	No	No	Yes	Yes
Keyboard support	kbin, kbled	No	No	Yes	Yes	Yes
RF radio support	rfin, rfout	No	No	Yes	Yes	Yes
Internal temp. sensor	readinternal-temp	Yes	No	Yes	Yes	Yes
Configurable input type	inputtype	No	No	No	Yes	Yes

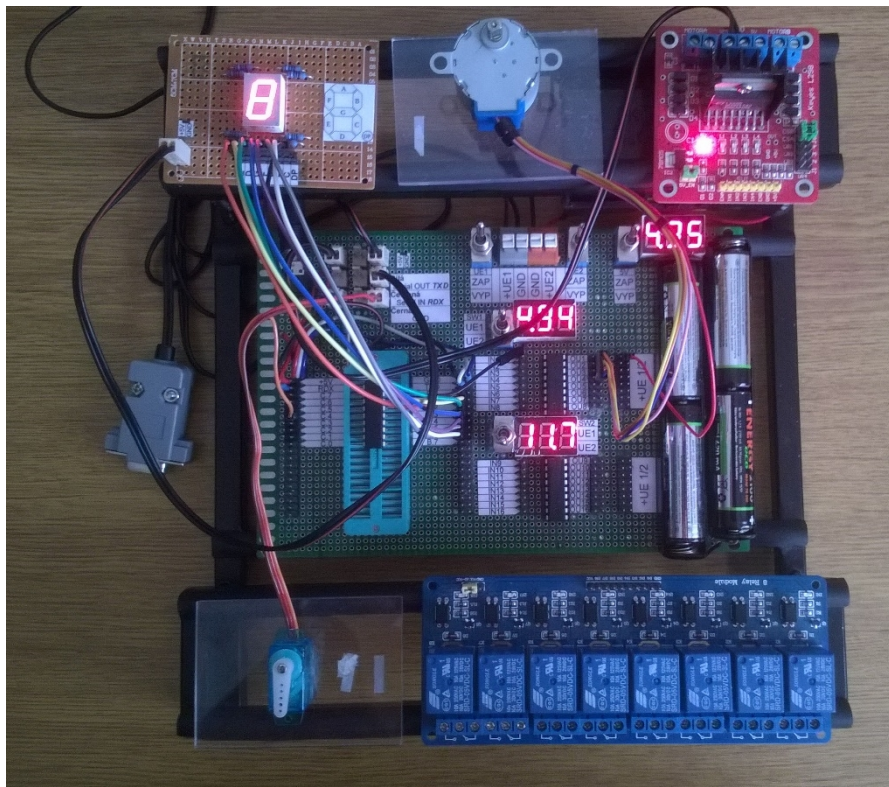
Obrázek 61 – Porovnání typů řady M (převzato z [18])

Příloha B

Feature	PICAXE Command	20X2	28X2	28X2 -5V	28X2 -3V	40X2	40X2 -5V	40X2 -3V
Base PIC micro (PIC18F series)		14K22	25K22	2520	25K20	45K22	4520	45K20
Voltage Range (V)		1.8-5.5	2.1-5.5	4.5-5.5	1.8-3.6	2.1-5.5	4.5-5.5	1.8-3.6
PICAXE Firmware Version Range		C.0+	B.3+	B.0-B.2	B.A-B.C	B.3+	B.0-B.2	B.A-B.C
Current (still in production) part		Yes	Yes	No	No	Yes	No	No
Max Internal Freq (MHz) Max External Freq (MHz)	setfreq	64 n/a	16 64	8 40*	16 64	16 64	8 40*	16 64
Touch Sensor Support	touch	No	Yes	No	No	Yes	No	No
ADC Setup seq. or individual.	adcsetup	ind.	ind.	seq.	ind.	ind.	seq.	ind.
Internal ADC reference (V)	calibadc	1.024	1.024	No	1.2	1.024	No	1.2
Variables RAM (bytes)	peek, poke @bptr	128	256	256	256	256	256	256
Scratchpad RAM (bytes)	put, get @ptr	128	1024	1024	1024	1024	1024	1024
Internal Program slots External Program slots	run	1 32	4 32	4 32	4 32	4 32	4 32	4 32
Hardware Interrupt pins	hintsetup	2	3	3	3	3	3	3
Pwmout channels	pwmout	1	4	2	2	2	2	2
hpwm support	hpwm	Yes	Yes	No	Yes	Yes	Yes	Yes
power steering mode within hpwm	hpwm	Yes	Yes	No	Yes	Yes	No	Yes
pullups individually controller	pullup	Yes	Yes	No	Yes	Yes	No	Yes
SRLatch, FVR and DAC modules	slatch, fvrsetup dacsetup	Yes	Yes	No	No	Yes	No	No

Obrázek 62 – Porovnání typů řady X (převzato z [18])

Příloha C



Obrázek 64 – Stolní držák s moduly a vývojovou deskou



Obrázek 63 – Stolní držák