

UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

Příruční kalkulátor
Karel Juryca

Bakalářská práce
2014

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Karel Juryca**
Osobní číslo: **I11266**
Studijní program: **B2612 Elektrotechnika a informatika**
Studijní obor: **Komunikační a mikroprocesorová technika**
Název tématu: **Příruční kalkulačtor**
Zadávací katedra: **Katedra elektrotechniky**

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

Cílem práce je vytvoření vlastního demonstračního kalkulačtoru s procesorem AVR a vizualizací prostřednictvím LCD, sedmissegmentového displeje, či jednotky s digitrony. Realiujte základní algebraické operace plus alespoň 5 operací dalších pomocí iteračních vzorců bez využití knihovny Math, implementujte minimálně jeden paměťový registr a prioritizaci operací. Výpočet bude prováděn s přesností min. 7 míst za řádovou čárkou. Kalkulačtor bude napájen bateriemi.

- Analýza problematiky aproximace základních mat. operací.
- Návrh zařízení a DPS.
- Konstrukce řídicí části.
- Konstrukce a oživení přístroje.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

[1] Mrázek J.: Hry s kalkulátory, SPN Praha 1984

[2] HP35s tutoriály dostupné z www.hp.com

Vedoucí bakalářské práce:

Ing. Martin Dobrovolný, Ph.D.

Katedra elektrotechniky

Datum zadání bakalářské práce:

20. prosince 2013

Termín odevzdání bakalářské práce:

9. května 2014



prof. Ing. Simeon Karamazov, Dr.
děkan

L.S.

Ing. Zdeněk Němec, Ph.D.
vedoucí katedry

V Pardubicích dne 31. března 2014

Prohlášení autora

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

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

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 4. 5. 2014

Karel Juryca

Poděkování

Chtěl bych zejména poděkovat svému vedoucímu bakalářské práce panu Ing. Martinu Dobrovolnému, Ph.D. za cenné rady a připomínky a motivaci při tvorbě bakalářské práce a samotného zařízení.

Dále bych chtěl poděkovat panu Ing. Pavlu Rozsivalovi za pomoci při výrobě desky plošných spojů.

V neposlední řadě bych chtěl poděkovat své rodině, která mě během celého studia na vysoké škole podporovala.

Anotace

Práce je věnována návrhu vlastního příručního kalkulátoru. V úvodu práce je popsána historie výpočetních pomůcek a výpočetní techniky se zaměřením na kalkulátory. V práci jsou popsány vybrané iterační metody a algoritmy pro výpočet pokročilých matematických funkcí. Navržené postupy byly implementovány do kalkulátoru v jazyce C.

Klíčová slova

Kalkulátor, ATmega32, maticová klávesnice, iterační vzorce, programování, sedmissegmentový displej

Title

Handheld calculator

Annotation

This work deals with design and construction of handheld calculator based on AVR microcontroller. The first part of work focused on the history of computer equipment and historical tools for calculations. The paper describes iterative methods and algorithms for advanced mathematical functions. The proposed methods were implemented in the developed calculator in C language.

Keywords

Calculator, ATmega32, matrix keyboard, iterative formulas, programming, 7-segment display

Obsah

SEZNAM ZKRATEK	9
SEZNAM OBRÁZKŮ	10
SEZNAM TABULEK	10
ÚVOD	11
1. HISTORIE	12
1.1. VÝPOČETNÍ POMŮCKY	12
1.1.1. Abakus	12
1.1.2. Napierovy kosti	12
1.1.3. Logaritmické pravítko	14
1.2. MECHANICKÉ KALKULÁTORY	15
1.2.1. Pascalina	15
1.2.2. Krokovací kalkulátor	15
1.2.3. Arithmometr	15
1.2.4. Milionář	15
1.2.5. Curta	16
1.3. ELEKTRONICKÉ PŘÍRUČNÍ KALKULÁTORY	16
1.3.1. Kapesní kalkulátory	17
1.3.2. První vědecký příruční kalkulátor	17
1.3.3. První finanční příruční kalkulátor	18
1.3.4. První programovatelný příruční kalkulátor	18
1.3.5. Příruční kalkulátory s LCD displejem	18
1.3.6. Hodinky s kalkulátorem	19
1.3.7. Příruční kalkulátor se solárním článkem	19
1.3.8. Kuličkové pero s funkcí kalkulátoru	19
2. APROXIMAČNÍ METODY VÝPOČTU POKROČILÝCH MATEMATICKÝCH FUNKCÍ.....	20
2.1. DRUHÉ ODMOCNINA	20
2.2. VÝPOČET K-TÉ ODMOCNINY.....	21
2.3. VÝPOČET LUDOLFOVA ČÍSLA.....	21
2.4. GONIOMETRICKÉ FUNKCE	22
2.5. LOGARITMY	23
2.6. EXPONENCIÁLNÍ FUNKCE	24
2.7. VÝPOČET FAKTORIÁLU	25
3. ARCHITEKTURA NÁVRHOVÉHO ŘEŠENÍ	26
4. HARDWAROVÉ ŘEŠENÍ.....	28
4.1. ZÁKLADNÍ DESKA KALKULÁTORU.....	28
4.1.1. Mikrokontrolér	28

4.1.2.	Napájení kalkulátoru.....	29
4.1.3.	Maticová klávesnice	30
4.1.4.	Programovatelný konektor ISP.....	32
4.2.	DISPLEJ.....	33
4.2.1.	Rozložené displeje.....	34
4.2.2.	Sedmisegmentový displej.....	34
4.2.3.	4094	35
4.3.	SEZNAM SOUČÁSTEK.....	38
5.	SOFTWAREVÉ ŘEŠENÍ.....	39
5.1.	VÝVOJOVÝ DIAGRAM CELÉHO PROGRAMU.....	39
5.2.	DETEKCE TLAČÍTEK	39
5.3.	VKLÁDÁNÍ ČÍSEL.....	39
5.4.	ALGORITMY ITERAČNÍCH VZORCŮ	40
5.4.1.	Sinus	40
5.4.2.	Cosinus	41
5.4.3.	Faktoriál.....	41
5.4.4.	Přirozený logaritmus	42
5.4.5.	Dekadický logaritmus.....	43
5.4.6.	Druhá odmocnina	43
5.4.7.	Ludolfovo číslo.....	44
5.4.8.	Exponenciální funkce	45
5.5.	VYPOČET A VYHODNOCENÍ	45
5.6.	ZOBRAZENÍ NA DISPLEJI.....	46
5.6.1.	Výběr typu zobrazení na displeji.....	46
5.6.2.	Vložení dat do pole.....	47
5.6.3.	Posílání dat z pole na displej	47
6.	DOSAŽENÉ VÝSLEDKY.....	48
7.	ZÁVĚR.....	50
	SEZNAM POUŽITÉ LITERATURY	51
	PŘÍLOHA A - SCHÉMA ZAPOJENÍ ZÁKLADNÍ DESKY KALKULÁTORU	52
	PŘÍLOHA B - DESKA PLOŠNÝCH SPOJŮ ZÁKLADNÍ DESKY KALKULÁTORU (MĚŘÍTKO 2:1).....	52
	PŘÍLOHA C - SCHÉMA ZAPOJENÍ DISPLEJE	54
	PŘÍLOHA D - DESKA PLOŠNÝCH SPOJŮ DISPLEJE (MĚŘÍTKO 3:2).....	55
	PŘÍLOHA E - VÝVOJOVÝ DIAGRAM PROGRAMU.....	56
	PŘÍLOHA F - CD-ROM.....	57

Seznam zkratek

IEEE	Institute of Electrical and Electronics Engineers
LCD	Liquid crystal display
LED	Light-Emitting Diode
LDO	Low-dropout
NiMH	Nikl-metal hydrid
NiCd	Nikl-kadmium
ISP	In System Programming
NC	Not Connection

Seznam obrázků

OBRÁZEK 1 - NAPIEROVY KOSTI.....	13
OBRÁZEK 2 - NAPIEROVY KOSTI – SEŘAZENÍ	13
OBRÁZEK 3 - NAPIEROVY KOSTI – VÝPOČET	14
OBRÁZEK 4 - LOGARITMICKÉ PRAVÍTKO	14
OBRÁZEK 5 - MECHANICKÝ KALKULÁTOR CURTA.....	16
OBRÁZEK 6 - PRVNÍ VĚDECKÝ PŘÍRUČNÍ KALKULÁTOR HP-35	18
OBRÁZEK 7 - ARCHITEKTURA SYSTÉMU	27
OBRÁZEK 8 - ROZLOŽENÍ PINŮ ATMEGA32L V POUZDŘE TQFP44	29
OBRÁZEK 9 - ROZLOŽENÍ KLÁVES NA Maticové KLÁVESNICE.....	31
OBRÁZEK 10 - PRINCIPIÁLNÍ SCHÉMA Maticové KLÁVESNICE.....	32
OBRÁZEK 11 - ISP KONEKTOR (ZAPOJENÍ ATMEL - 5V).....	33
OBRÁZEK 12 - ROZLOŽENÍ SEGMENTŮ.....	34
OBRÁZEK 13 - SPOLEČNÁ ANODA.....	35
OBRÁZEK 14 - BLOKOVÉ SCHÉMA 4094	35
OBRÁZEK 15 - ROZLOŽENÍ PINŮ 4094	36
OBRÁZEK 16 - VÝVOJOVÝ DIAGRAM VKLÁDÁNÍ ČÍSEL.....	40
OBRÁZEK 17 - VÝVOJOVÝ DIAGRAM SIN(X).....	41
OBRÁZEK 18 - VÝVOJOVÝ DIAGRAM VÝPOČTU FAKTORIÁLU.....	42
OBRÁZEK 19 - VÝVOJOVÝ DIAGRAM VÝPOČTU PŘIROZENÉHO LOGARITMU	43
OBRÁZEK 20 - VÝVOJOVÝ DIAGRAM PRO VÝPOČET DRUHÉ ODMOCNINY	44
OBRÁZEK 21 - VÝVOJOVÝ DIAGRAM PRO VÝPOČET LUDOLFOVA ČÍSLA.....	44
OBRÁZEK 22 - VÝVOJOVÝ DIAGRAM PRO VÝPOČET EXPONENCIÁLNÍ FUNKCE.....	45
OBRÁZEK 23 - VÝVOJOVÝ DIAGRAM PRO VÝBĚR TYPU ZOBRAZENÍ NA DISPLEJ....	46
OBRÁZEK 24 - VÝVOJOVÝ DIAGRAM ALGORITMU POSÍLÁNÍ NA DISPLEJ	47
OBRÁZEK 25 - VÝSLEDNÁ KONSTRUKCE PŘÍRUČNÍHO KALKULÁTORU	49

Seznam tabulek

TABULKA 1 - VÝPOČET LUDOLFOVA ČÍSLA – PRVNÍ CYKLUS ALGORITMU	22
TABULKA 2 - VÝPOČET LUDOLFOVA ČÍSLA - KONEČNÁ TABULKA	22
TABULKA 3 - VÝPOČET MAXIMÁLNÍHO ODEBÍRANÉHO PROUDU	29
TABULKA 4 - PRAVDIVOSTNÍ TABULKA FUNKCÍ KLÁVESNICE	32
TABULKA 5 - PRAVDIVOSTNÍ TABULKA 4094.....	37
TABULKA 6 - SEZNAM SOUČÁSTEK	38
TABULKA 7 - RYCHLOST VÝPOČTŮ FUNKCÍ.....	48

Úvod

V této části uvedu, proč jsem si toto téma bakalářské práce vybral. Hlavním důvodem byl můj zájem o matematické předměty při studiu na vysoké škole. Dalším důvodem byla snaha o sestrojení výpočetního přístroje, na kterém by bylo možno experimentálně ověřit chování implementovaných algoritmů.

Existuje celá řada možností, jakým způsobem se na tuto práci zaměřit. Například použít programovatelný kalkulátor a iterační algoritmy do něj naprogramovat, a poté provést jejich ověření. Tento návrh byl zavržen z důvodů, že bychom se na zadaný problém zaměřovali pouze softwarově, a proto byly také zavrženy možnosti otestování iteračních algoritmů pomocí některých softwarových simulací nebo výpočetního softwaru Matlab. Další možností bylo realizovat kalkulátor pomocí klopných obvodů, pomocí nichž bychom mohli provádět výpočty. Tato možnost by přesahovala rámec bakalářské práce.

Nakonec bylo rozhodnuto, že se práce zaměří na sestrojení vlastního příručního kalkulátoru a naprogramování funkcí do mikrokontroléru.

V práci je popsána konstrukce příručního kalkulátoru založeného na mikrokontroléru AVR. I když byl zvolen výkonný mikrokontrolér s dostatečnou programovou pamětí, do kterého lze implementovat celou knihovnu *math.h*, tak návrh byl veden k implementaci na co nejnižší úroveň.

Hlavní cíle bakalářské práce:

- a) Analýza problematiky aproximace základních matematických funkcí, rešerše iteračních metod a rozvést různé algoritmy pro výpočet některých matematických funkcí, které budou použity v programu kalkulátoru.
- b) Konstrukce a vytvoření softwaru pro příruční kalkulátor, který bude schopen vypočítat základní matematické operace a několik složitějších operací. Složité operace budou realizovány ze znalostí iteračních metod.

Metody, které jsem u této bakalářské práci použil, jsou:

- Obsahová analýza odborné literatury: historie, iterační metody a datasheety elektronických součástek.
- Metoda kompilace citovaných informací formou kapitol bakalářské práce.
- Metoda zpracování a srovnávání dat získaných v odborné literatuře mi sloužila k návržení příručního kalkulátoru a srovnání více možností hardwarového řešení.
- Metoda syntézy mi sloužila při vytvoření prototypu příručního kalkulátoru.

1. Historie

Úvodní část se zaměřuje na historii výpočetních pomůcek, mechanických kalkulátorů a počátky vývoje elektronických příručních kalkulátorů.

1.1. Výpočetní pomůcky

Pravděpodobnou první výpočetní pomůckou byly prsty na ruce a nohou. Z toho důvodu používáme dnes desítkovou soustavu. Jenže při počítání s vyššími čísly nám ruce a nohy nestačily, a proto se lidé zaměřili na počítání přírodních předmětů (např. kamínky, větvičky, ...). Při vývoji člověka se začaly objevovat dokonalejší výpočetní pomůcky až k počítačům a kalkulátorům, které známe dnes.[1]

1.1.1. Abakus

Jde o latinské slovo, které má svůj původ v řeckých slovech **abax** nebo **abakon** (tabule nebo deska). Možná pochází ze semitského slova **abq** (písek). Nejstarší exemplář je Salamiská deska, která byla používána Babyloňany asi 300 let př. n. l. a byla objevena roku 1846 na ostrově Salamis. Abakus je počítací pomůcka založená na systému korálku, které jsou zavěšeny na tyčkách, či pokládány do žlábků. S touto počítací pomůckou se daly vykonávat základní matematické operace. Abakus se hojně využíval už ve starověku až do současnosti. Ščot je typ ruského abaku vynalezený v 17. století, který se používá v některých částech Ruska dodnes. V dnešní době známe abakus v podobě dětského počítadla. [1]

1.1.2. Napierovy kosti

Tuto matematickou pomůcku vynalezl John Napier (1550-1617). Tato pomůcka pomáhala zjednodušit násobení čísel. Proto se daly provádět součiny velkých čísel daleko rychleji než bez této pomůcky. Na následujícím obrázku (Obrázek 1) jsou zobrazeny Napierovy kosti, které byly tvořeny z deseti hranolů (někdy bylo zapotřebí více), které byly popsány malou násobilkou. Byly vyráběny dokonce i ze slonoviny. [6]

1	1	2	3	4	5	6	7	8	9
2	0/2	0/4	0/6	0/8	1/0	1/2	1/4	1/6	1/8
3	0/3	0/6	0/9	1/2	1/5	1/8	2/1	2/4	2/7
4	0/4	0/8	1/2	1/6	2/0	2/4	2/8	3/2	3/6
5	0/5	1/0	1/5	2/0	2/5	3/0	3/5	4/0	4/5
6	0/6	1/2	1/8	2/4	3/0	3/6	4/2	4/8	5/4
7	0/7	1/4	2/1	2/8	3/5	4/2	4/9	5/6	6/3
8	0/8	1/6	2/4	3/2	4/0	4/8	5/6	6/4	7/2
9	0/9	1/8	2/7	3/6	4/5	5/4	6/3	7/2	8/1

Obrázek 1 - Napierovy kosti¹

Na příkladu je ukázáno, jakým způsobem lze vynásobit 268×598 . Nejdříve si připravíme sloupce 2, 6 a 8 a položíme je vedle sebe v tomto pořadí. Jak vidíme na následujícím obrázku (Obrázek 2), jsou zvýrazněny řádky 5, 9, 8, které budeme potřebovat k výpočtu. [6]

1	2	6	8
2	0/4	1/2	1/6
3	0/6	1/8	2/4
4	0/8	2/4	3/2
5	1/0	3/0	4/0
6	1/2	3/6	4/8
7	1/4	4/2	5/6
8	1/6	4/8	6/4
9	1/8	5/4	7/2

Obrázek 2 - Napierovy kosti – seřazení²

¹ Vlastní zdroj

² Vlastní zdroj

Z těchto řádků sestavíme tabulku, která je znázorněna na následujícím obrázku (Obrázek 3). Nesmíme ovšem zapomenout, že řádky musí být ve správném pořadí v závislosti na druhém činiteli. Násobení provádíme pomocí sčítání tak, že postupně sčítáme členy v každé diagonále počínaje pravou diagonálou. Tyto diagonály jsou pro přehlednost zvýrazněny modrou a zelenou barvou. Musíme si však dát pozor, že pokud součet diagonály překročí číslo 9, například 12, musíme přičíst číslo 1 k následující levé diagonále. Výsledkem součinu 268×598 je sestava mezivýsledků jednotlivých diagonál. Nezapomeneme brát jenom jednotky z následujících mezivýsledků. [6]

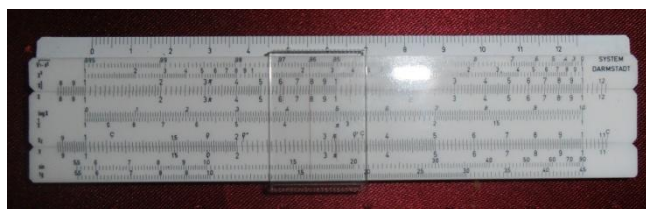
	2	6	8	
1	1	3	4	5
$4+2=6$	1	5	7	9
$18+2=20$	1	4	6	8
	21+1=22	16	4	

160264

Obrázek 3 - Napierovy kosti – výpočet³

1.1.3. Logaritmické pravítko

První ho zkonstruoval anglický matematik William Oughtred a to v roce 1632. Během historie došlo k jejímu zdokonalení a zvýšení přesnosti. Logaritmické pravítko bylo matematickou pomůckou, která nám pomáhala urychlit výpočty. S touto pomůckou šlo násobit, dělit a vypočítávat různé funkce například druhé mocniny, odmocniny, goniometrické funkce a logaritmy. Ovšem přesnost byla omezená. Tato pomůcka byla velmi používána až do 20. století, kde ji nahradily elektronické kalkulačky, které byly schopny funkci vypočítat daleko přesněji a rychleji. [12]



Obrázek 4 - Logaritmické pravítko⁴

³ Vlastní zdroj

⁴ Vlastní zdroj

1.2. Mechanické kalkulátory

Hlavní snahou bylo vynalézt a vyrobit stroj, který pomocí různých převodů a ozubených kol dokáže provádět základní matematické operace. Profesor Wilhelm Schickart roku 1623 takový počítací stroj vyrobil a šlo o první tohoto druhu. Tento počítací stroj zvládl čtyři základní matematické operace, ovšem ne úplně automaticky a přesně. Stroj byl pouze rekonstruován podle dochovaných materiálů. Originál se nedochoval. [4]

Na začátku 20. století se výroba mechanických kalkulátorů obrovsky rozšířila, tisíce společností především z Německa a USA, vyrobili přes milion kalkulátorů a účetních strojů. Vše skončilo po roce 1960, kdy se začaly vyrábět první elektronické kalkulátory. [13]

1.2.1. Pascalina

V roce 1641 zkonstruoval francouzský fyzik a matematik Blaise Pascal (1623-1662) mechanický kalkulátor zvaný Pascalina, který uměl pouze sčítat a odčítat. Pascal tento kalkulátor zkonstruoval pro svého otce, který pracoval jako výběrčí daní, aby mu ulehčil práci. [4]

1.2.2. Krokovací kalkulátor

Německý matematik a filozof Gottfried Wilhelm von Leibniz (1646-1716) zkonstruoval v roce 1671 mechanický kalkulátor pod názvem Krokovací kalkulátor (anglicky The Stepped Reckoner). Leibniz rozšířil Pascalův kalkulátor. Krokovací kalkulátor už uměl násobit pomocí postupného sčítání. Leibniz silně zastával binární systém. Věděl, že pro přístroje je to daleko jednodušší mít pouze dvě úrovně, tedy vypnuto a zapnuto. Leibniz předvídal, že binární systém je vhodný pro kalkulátory a počítací stroje, ale u svého kalkulátoru ho nepoužil. [8]

1.2.3. Arithmometr

Francouzský matematik Charles Xavier Thomas de Colmar (1785-1870) zkonstruoval v roce 1820 mechanický kalkulátor zvaný Arithmometr. Zatímco Pascalina od Pascala a Krokovací kalkulátor od Leibnize byly jen rarity, tak Arithmometr byl první komerčně vyráběný kalkulátor. Hlavním důvodem jejich výroby byla průmyslová revoluce a potřeba lidí provádět opakující početní operace. Arithmometr uměl sčítat, odčítat, násobit a s více komplikovanou uživatelskou účastí i dělit. Nevýhodou tohoto kalkulátoru byly jeho rozměry, ale i přesto byl velice populární a prodával se 90 let. [2]

1.2.4. Milionář

Švýcarský inženýr Otto Steinger (1858-1923) dostal v roce 1892 svůj první patent na počítací stroj, který uměl přímé násobení. Byl čtvrtým člověkem (po Edmundu Barbourovi, Ramónovi Vereaoovi a Leónu Bolléeovi), kterému se to podařilo. Steinger byl ovšem komerčně úspěšnější, protože jeho počítací stroj byl ve výrobě až do roku 1935, a to i přes

svou vysokou cenu. Na začátku dvacátého století byla cena tohoto počítačového stroje od 475 do 1100 dolarů. Hlavní výhodou Steingerova počítačového stroje byla jeho ohromná rychlost zejména při násobení a dělení. Zkušený uživatel tohoto počítačového stroje byl schopen vynásobit dvě osmimístná čísla asi za sedm sekund, což bylo na tu dobu obdivuhodné. V roce 1895 se začal vyrábět pod názvem Milionář (Millionär v originále). Tento stroj byl vytvořen pro obchodní výpočty, vědce a vládní úřady. [6]

1.2.5. Curta

Šlo o velice zajímavý příruční mechanický kalkulátor, který navrhl v roce 1930 rakouský konstruktér Curt Herzstark (1902-1988). Do výroby se dostal až v roce 1948. Jeho hlavní výhodou byla jeho velikost. Šlo o kompaktní válcovitý přístroj, který se vešel do dlaně. Tento kalkulátor uměl sčítat, odčítat, násobit, dělit, s vyššími obtížemi i další funkce, například odmocňovat. Vyráběly se dva typy, Curta typ I a typ II. U druhého typu bylo možno vkládat až jedenáctimístná čísla na rozdíl od prvního typu, kde šly vkládat pouze osmimístná čísla. Šlo o velmi užitečný a praktický příruční kalkulátor, který se v době před elektronickými příručními kalkulátory hodil hlavně do terénu. V dnešní době by šlo spíše o kuriozitu či umělecký předmět. [6]



Obrázek 5 - Mechanický kalkulátor Curta⁵

1.3. Elektronické příruční kalkulátory

S postupným rozvojem integrovaných obvodů bylo možné sestavit elektronický kalkulátor, který bude přenosný. V roce 1967 vyrobila společnost Texas Instrument první prototyp elektronického příručního kalkulátoru pod názvem Cal-Tech. Nešlo o první model na trhu,

⁵ Zdroj [13]

protože byl prodán společnosti Canon v roce 1970. Prvním elektronickým příručním kalkulátorem, který se dostal na trh, byl Canon Pocketronic. Neměl ještě displej, a proto se výsledky tiskly na tepelnou pásku. Byl napájen třinácti NiCd články, což odpovídá 15,6V. V této době začal velký rozvoj výroby příručních kalkulátorů. Texas Instrument, Canon, Sharp, Casio, Hewlett-Packard jsou jen některé společnosti, které se výrobou příručních kalkulátorů zabývaly. [13]

1.3.1. Kapesní kalkulátory

Šlo o myšlenku, že by se kalkulátor vešel do náprsní kapsy. První kalkulátor, který se dal pokládat za kapesní, byl Busicom LE-120A. Byl uveden do prodeje v lednu roku 1971 a stál 395 dolarů. Jeho rozměry byly 72x124x24 mm. Byl napájen čtyřmi AA bateriemi a měl dvanáctimístný LED displej. Ovládal čtyři základní matematické operace a měl buď pevnou desetinnou čárku, nebo nastavitelnou na dvě nebo na čtyři desetinná místa. [13]

1.3.2. První vědecký příruční kalkulátor

Byl zkonstruován společností Hewlett-Packard a dostal název HP-35. Uvádí se jako první elektronické logaritmické pravítko. Displej kalkulátoru HP-35 byl reprezentován 10 místy pro číslo a dvěma pro exponent. Šlo o červený LED displej. Tento kalkulátor používal reverzní polskou notaci. Byl napájen třemi AA nabíjecími bateriemi. Byl představen 1. února 1972. Šlo vůbec o první kalkulátor, který jediným stiskem tlačítka mohl vypočítat logaritmus, goniometrickou funkci a další matematické funkce. Když byl stolní kalkulátor HP9100 ukázán Williamu Hewlettovi, požádal inženýry o verzi, která by se vešla do jeho náprsní kapsy. V počátku chtěl Hewlett-Packard vyrobit jen malé množství HP-35 pro své inženýry. Poté zkusili uvést na trh tento model a prodalo se jich statisíce. HP-35 byl pojmenován z toho důvodu, že obsahuje 35 tlačítek. Když vyšlo najevo, že kalkulátor HP-35 je mnohem populárnější, než výrobce očekával, došlo ke zvýšení výroby. Při uvedení na trh stál kalkulátor 395 dolarů. Tato cena byla poté snížena na 195 dolarů. V únoru 1975 byla výroba kalkulátoru ukončena. Úspěch kalkulátoru HP-35 a tlak konkurence vedl Hewlett-Packard k návrhům více modelů s vylepšenými nebo novými funkcemi. Hewlett-Packard obdržel 14. dubna 2009 cenu „IEEE Milestone in Electrical Engineering and Computing“ za kalkulátor HP-35. V červnu 2007 uvedl na trh Retro HP-35s, který připomíná 35. výročí prvního vědeckého příručního kalkulátoru. [13]

Reverzní polská notace

U klasických dnešních kalkulátorů používáme infixovou notaci zápisu čísel. Nejdříve zadáváme číslo, následuje operace, kterou chceme vykonat, a po ní zadáváme druhé číslo a takto můžeme pokračovat dále. U reverzní polské notace (postfixová notace) je tomu tak, že nejdříve zadáváme čísla, která oddělujeme konkrétním tlačítkem (u kalkulátoru HP-35 je to tlačítko ENTER), a teprve poté zadáváme operace, které chceme u těchto zadaných čísel provést. [9]



Obrázek 6 - První vědecký příruční kalkulátor HP-35⁶

1.3.3. První finanční příruční kalkulátor

Společnost Hewlett-Packard se pyšní ještě dalším prvenstvím. V roce 1973 vyrobila první příruční finanční kalkulátor po názvem HP-80. Obsahoval dost podobných funkcí jako HP-35, například základní matematické operace, druhou odmocninu, x -tou mocninu, exponenciální funkci. Hlavním rozdílem bylo, že tento kalkulátor byl určen hlavně pro ekonomické než pro vědecké výpočty. Obsahoval některé vhodné funkce pro ekonomii a práci s penězi. Uživatel nemusel ani znát vzorce, pouze stačilo, aby zadal příslušné parametry a kalkulátor mu zobrazil výsledek, který chtěl znát. [5]

1.3.4. První programovatelný příruční kalkulátor

Opět se vracíme ke společnosti Hewlett-Packard, které v lednu roku 1974 jako první vyrobila příruční kalkulátor HP-65, který měl možnost naprogramovat algoritmus výpočtů. Byl velice podobný HP-35, ale HP-65 mohl naprogramovat a nahrát algoritmus na magnetické karty, které se daly už naprogramované koupit. Naprogramovat šlo až sto řádků a existovalo 64 instrukcí. Další zajímavou funkcí byl převod mezi osmičkovou a desítkovou soustavou. Displej byl realizován stejně jako u vědeckého kalkulátoru HP-35. [5]

1.3.5. Příruční kalkulátory s LCD displejem

První byl vyroben roku 1972 společností Lloyd's. Byl nazván Lloyd's Accumatic 100. Ovládal pouze základní matematické operace. Hlavní výhodou byl LCD displej, který má nižší spotřebu, než LED displej, a proto se později začaly vyrábět kalkulátory s LCD displejem a LED displej se postupně z kapesních a příručních kalkulátorů vytrácel. [13]

⁶ Zdroj [13]

Společnost Sharp se může chlubit roku 1973 prvním kapesním kalkulátorem Sharp EL-805, který měl LCD displej. Tento kalkulátor disponoval pouze základními matematickými operacemi a byl napájen 1,5V baterií. Hlavním důvodem nízkého napájení byl LCD displej. Cena tohoto kalkulátoru byla 109,95 dolarů. [13]

1.3.6. Hodinky s kalkulátorem

Společnost Pulsar je uvedla na trh v roce 1975. Zvládaly čtyři základní matematické operace, čas a datum. LED displej obsahoval šest znaků a kvůli úspoře energie byl vždy displej po určité době vypnut a zapnout ho šlo pomocí tlačítka *pulsar*. Hodinky byly k dispozici ve dvou variantách, buď nerezové, nebo zlaté. Elektronika byla vyráběna v USA a pouzdro ve Švýcarsku. [13]

1.3.7. Příruční kalkulátor se solárním článkem

V roce 1976 společnost Sharp vyrobila první kalkulátor Sharp EL-8026, který měl možnost dobíjení baterií pomocí solárních článků, když nebyl kalkulátor právě používán. Solární články byly umístěné na zadní straně kalkulátoru, což dnes u kalkulátorů nevidáme. Spolu se základními matematickými operacemi zvládl též druhou odmocninu. Kalkulátory, které byly poháněny pouze solárními články, přišli na trh až na konci sedmdesátých let 20. století. [13]

1.3.8. Kuličkové pero s funkcí kalkulátoru

Bylo uvedeno na trh v Japonsku roku 1975. Byl vyroben pod názvem Calku-pen. Umožňoval výpočet základních matematických operací. Vyroben byl z kovu. Obsahoval pět tlačítek a každé z těchto tlačítek mělo čtyři funkce. Displej byl tvořen červeným osmimístným LED displejem. Pero bylo napájeno 1,5V baterií a vypínač byl konstruován na konci pera. [13]

Z úvodní části bylo rozhodnuto, jakým způsobem se bude zaobírat hardwarové a softwarové řešení příručního kalkulátoru. Použití infixové notace místo reverzní polské notace, protože toto zadávání používá většina dnešních kalkulátorů a je pro dnešní populaci přirozenější. I když má LED displej větší příkon, byl použit místo LCD displeje, pro výraznější zobrazení čísel. Snížení příkonu displeje byl vyřešen použitím nízkovýkonového sedmsegmentového LED displeje. Zobrazení čísel displeje byl zvolen, jak klasický zápis čísel (jako u obyčejných kalkulátorů), tak i vědecký zápis čísel (jako u vědeckých kalkulátorů např. HP-35), v závislosti na velikosti zobrazovaného čísla. Napájení kalkulátoru bylo zvoleno pomocí baterií, pomocí transformátoru ze sítě nebo pomocí USB portu z počítače. Dalším důležitým poznatkem byla možnost více funkcí jednotlivých tlačítek klávesnice, které budou ovládány pomocí přepínačů.

2. Aproximační metody výpočtu pokročilých matematických funkcí

V navrhovaném systému budou implementovány pokročilé matematické funkce (goniometrické funkce, logaritmy, ...). Z toho důvodu bylo zapotřebí provést rozbor postupů, jakým způsobem se dají vypočítat tyto matematické operace při co nejjednodušší algoritmičtější náročnosti a co nejméně spotřebovaném paměťovém prostoru. Tato kapitola se zaměřuje na aproximační metody některých pokročilých matematických funkcí.

Pomocí aproximačních metod lze realizovat některé funkce i na nejjednodušších kalkulátorech. Použitím iteračních metod lze vypočítat poměrně přesný výsledek námi zadané funkce. Čím vyšší přesnost požadujeme, tím vyšší počet iteračních kroků musíme provést. Jednoduché postupy nám neudávají moc přesné výsledky. Pokud bychom chtěli přesnější výsledky, museli bychom použít složitější algoritmy. Existují dvě skupiny iteračních metod:

- Vyčíslení **polynomů** resp. **racionálních lomených funkcí**. U této metody stačí pouze dosadit proměnné a provést výpočet.
- **Rekurzivní vzorce** - tato metoda je výhodnější, protože pokud použijeme tento vzorec vícekrát za sebou, tím bude výsledek přesnější. Tedy přesnost výsledku můžeme ovlivnit.

Detailnější popis problematiky v [4 str. 97]

V kalkulátoru budou implementovány tyto matematické funkce a dále bude proveden výpočet Ludolfova čísla.

- Druhá odmocnina
- Goniometrické funkce (sinus, cosinus, tangens)
- Logaritmy (přirozený, dekadický)
- Exponenciální funkce (e^x)
- Faktoriál

2.1. Druhé odmocnina

Druhou odmocninu čísla lze aproximovat pomocí např. Newtonovy metody. Postup této metody je následující:

Nejdříve odhadneme výsledek zadané druhé odmocniny a pomocí tohoto odhadu vypočteme nový odhad, který je samozřejmě přesnější. Tento postup opakujeme v závislosti, na jakou přesnost chceme počítat. [4 str. 100]

$$x_{n+1} = \frac{1}{2} \cdot \left(\frac{A}{x_n} + x_n \right) \quad (2.1)$$

Pokud se hodnoty $x_{n+1} \approx x_n$, tak $x_{n+1} \approx \sqrt{A}$

kde A je odmocňované číslo a x_n n-té přiblížení

Například máme za úkol vypočítat druhou odmocninu ze 7. Postup výpočtu je následující:

1. Odhadneme přibližně druhou odmocninu ze 7, což odpovídá přibližné hodnotě 2,5.
2. 7 vydělíme odhadem v našem případě 2,5 a připočteme tento odhad. Mezivýsledek vydělíme dvěma a vyjde nám nový odhad 2,65.
3. Opakujeme bod 2) s novým odhadem. To lze opakovat libovolně krát v závislosti na přesnosti výsledku, kterou požadujeme.

Po druhém opakování je přibližný výsledek 2,645754, což se liší od skutečné druhé odmocniny ze 7 až na šestém desetinném místě. [9]

2.2. Výpočet k-té odmocniny

Newtonovu metodu lze analogicky použít i pro aproximaci výpočtu k-té odmocniny.

$$x_{n+1} = \frac{1}{k} \cdot \left[\frac{A}{x_n^{k-1}} + (k-1) \cdot x_n \right] \quad (2.2)$$

Pokud se hodnoty $x_{n+1} \approx x_n$, tak $x_{n+1} \approx \sqrt[k]{A}$

kde A je číslo, které chceme umocnit a x_n n-té přiblížení. [4 str. 101]

2.3. Výpočet Ludolfova čísla

Existuje velké množství algoritmu a aproximačních metod pro výpočet přibližného Ludolfova čísla. Například rozvoj do Leibnizovy řady, Eulerovy řady, použití metody Monte Carlo a mnohé další. U metody Monte Carlo je zapotřebí velké množství náhodných hodnot, abychom mohli přesněji definovat Ludolfovo číslo, a proto tato metoda nebyla použita.

V následující části je ukázán iterační postup, který se dá aplikovat na nejobyčejnějších kalkulátorech. Výpočet provedeme tak, že budeme počítat níže uvedeným způsobem a výsledky si psát do dvou sloupců. Do levého sloupce nadepíšeme 0 a do pravého 0,5.

Nejdříve vypočteme aritmetický průměr řádku a výsledek napíšeme do levého sloupce na další řádek. Poté tento výsledek vynásobíme s číslem z předchozího řádku v pravém sloupci a tento součin odmocníme a výsledek napíšeme do dalšího řádku pravého sloupce.

Tabulka 1 - Výpočet Ludolfova čísla – první cyklus algoritmu

Levý sloupec	Pravý sloupec
0,00000000	0,50000000
0,25000000	0,35355339

Tento celý postup provádíme do té doby, než se poslední hodnoty sloupců rovnají na požadované desetinné místo, jak je vidět na následující tabulce (Tabulka 2).

Tabulka 2 - Výpočet Ludolfova čísla - konečná tabulka

Levý sloupec	Pravý sloupec
0,00000000	0,50000000
0,25000000	0,35355339
0,30177670	0,32664074
0,31420872	0,32036443
0,31728657	0,31882179
0,31805418	0,31843775
0,31824597	0,31834185
0,31829391	0,31831788
0,31830589	0,31831188
0,31830889	0,31831039
0,31830964	0,31831001
0,31830982	0,31830992
0,31830987	0,31830989
0,31830988	0,31830989
0,31830989	0,31830989

Přibližné Ludolfovo číslo získáme převrácením hodnoty posledního čísla v pravém sloupci, což odpovídá přibližné hodnotě 3,141592649. Přesnost výpočtu s takovým počtem opakování je na sedm desetinných míst. [9 str. 36 – 37]

Dnešní kalkulátory mají Ludolfovo číslo uloženo v paměti na určitý počet desetinných míst. To ovšem záleží na konkrétním kalkulátoru.

2.4. Goniometrické funkce

Mezi goniometrické funkce patří funkce sinus, cosinus, tangens, kotangens, sekans, kosekans. Jsou definovány jako poměr stran v pravoúhlém trojúhelníku nebo se dají definovat jako určitá vzdálenost na jednotkové kružnici. V následující části jsou uvedeny některé iterační vzorce a algoritmy, které nám pomáhají aproximovat uvedené funkce. [11]

Existují vzorce, které nám pomohou vypočítat goniometrické funkce i na nejobyčejnějších kalkulátorech, akorát tyto vzorce mají omezenou přesnost. Níže uvedený vzorec, slouží k výpočtu přibližné hodnoty sinu.

$$\sin x \cong \left[\left(\left(\frac{x}{100} \cdot 0,5924 \right)^2 - 1,257 \right)^2 + 0,1645 \right] \cdot \frac{x}{100} \quad (2.3)$$

Ovšem jeho přesnost je v závislosti na použitých koeficientech pouze tři desetinná místa. [9 str. 40]

Je známo, že lze goniometrické funkce sinus a cosinus rozložit do řad.

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{(2n+1)!} \quad (2.4)$$

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{(2n)!} \quad (2.5)$$

Pro obě rovnice platí, že ($|x| < +\infty$)

$$\tan x = \frac{\sin x}{\cos x} \quad (2.6)$$

$$\cot x = \frac{1}{\tan x} = \frac{\cos x}{\sin x} \quad (2.7)$$

Detailnější popis v [3 str. 704]

2.5. Logaritmy

Logaritmická funkce je inverzní k funkci exponenciální a lze jí napsat takto:

$$y = \log_a x \quad (2.8)$$

kde a značí základ logaritmu. Pokud by základem bylo Eulerovo číslo, šlo by o tzv. přirozený logaritmus (značený speciálně \ln). Pokud by základem bylo číslo deset, jde o tzv. dekadický logaritmus, u kterého se většinou číslo a neudává (např. $y = \log(x)$) [11]

Následující část je zaměřena na aproximační metody pro výpočet logaritmů. Pro výpočet logaritmů si musíme nejdříve vypočítat pomocné číslo z , pomocí tohoto vzorce

$$z = \frac{x-1}{x+1} \quad (2.9)$$

kde x odpovídá číslu, u které chceme logaritmus vypočítat. Tato metoda má však své podmínky. Maximální hodnota logaritmu, kterou můžeme vypočítat, je druhá odmocnina ze základu. Minimální hodnota, kterou lze vypočítat, je převrácená hodnota maximální hodnoty logaritmu. Například pro dekadický logaritmus můžeme použít čísla v rozmezí přibližně od 0,316 do 3,162. Pro přirozený logaritmus v rozmezí přibližně od 0,6065 do 1,649. Pokud bychom použili čísla mimo tato rozmezí, narůstala by chyba nad

akceptovatelnou mez. Nesmíme zapomenout, že velikost základu není omezena. Tedy pro výpočet logaritmu slouží tento vzorec:

$$\log x \cong (z^2 \cdot a + b) \cdot z \quad (2.10)$$

Přesnost tohoto vzorce je však pouze na tři desetinná místa v závislosti na uvedených zaokrouhlených koeficientech a , b . Pro dekadický logaritmus mají tyto koeficienty přibližnou hodnotu $a = 0,36415$ a $b = 0,86304$ a pro přirozený logaritmus $a = 0,70225$ a $b = 1,99938$. Pokud bychom chtěli vypočítat logaritmus čísla, který neleží v rozsahu, musíme toto číslo rozložit na součin čísel, které už do tohoto rozsahu patří, a poté vypočítat logaritmus pro každý z těchto činitelů a nakonec výsledky sečíst. [9 str. 56 - 58]

Pro přesnější výpočty existují rozvoje do řady, u kterých si můžeme přesnost určit sami v závislosti na horní hranici řady. Pro výpočet přirozeného logaritmu existuje tento rozvoj.

$$\ln x = 2 \left(\frac{x-1}{x+1} + \frac{(x-1)^3}{3(x+1)^3} + \frac{(x-1)^5}{5(x+1)^5} + \dots \right) = 2 \sum_{i=1}^{\infty} \frac{(x-1)^{2i-1}}{(2i-1)(x+1)^{2i-1}} \quad (2.11)$$

Platí, že $x > 0$.

Pro výpočet dekadického logaritmu lze použít tento vzorec.

$$\log_{10} a = \frac{\ln a}{\ln 10} \quad (2.12)$$

Zde je uvedena přibližná hodnota přirozeného logaritmu deseti, který se nám hodí k výpočtu dekadického logaritmu. Tuto hodnotu můžeme mít v paměti kalkulátoru uloženou, abychom jí nemuseli při každém výpočtu dekadického logaritmu počítat.

$$\ln 10 = 2,302\ 585\ 09 \dots \quad (2.13)$$

Detailnější popis v [3 str. 703]

2.6. Exponenciální funkce

Jde o matematickou funkci, kterou lze zapsat takto:

$$y = a^x \quad (2.14)$$

kde a - je základem a náleží všem kladným reálným číslům, kromě jedné.

x - je exponent a náleží všem reálným číslům (dokonce i komplexním číslům). [11]

Následující část je zaměřena na aproximační metody výpočtu exponenciální funkce. Výpočet exponenciální funkce na nejobyčejnějším kalkulátoru je možný a slouží nám k tomu tento vzorec.

$$e^x \cong \left(\frac{2,122 \cdot x}{8,469 - x} + 1 \right)^4 \quad (2.15)$$

Má však nevýhody. Uvedený vzorec je platný pro každé x mezi nulou a jedničkou, pokud bychom chtěli například vypočítat $e^{-0,75}$, tak bychom nejdříve vypočítali $e^{0,75}$ pomocí tohoto vzorce a výsledek nakonec převrátili. Jeho přesnost je v závislosti na zaokrouhlených koeficientech pouze na tři desetinná místa. [9 str. 59 - 60]

Pokud vyžadujeme přesnější výpočet, je třeba použít rozvoj exponenciálních funkcí do řady, které jsou uvedeny zde:

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots = \sum_{k=0}^{\infty} \frac{x^k}{k!} \quad (2.16)$$

Platí pro $|x| < +\infty$.

$$a^x = e^{x \cdot \ln a} = 1 + \frac{x \ln a}{1!} + \frac{x^2 \ln^2 a}{2!} + \frac{x^3 \ln^3 a}{3!} + \dots = 1 + \sum_{k=1}^{\infty} \frac{x^k \ln^k a}{k!} \quad (2.17)$$

Platí pro $|x| < +\infty, a > 0$. [3 str. 703]

2.7. Výpočet faktoriálu

Faktoriál čísla n nazýváme funkci $f_{(n)}$ na množině všech nezáporných celých čísel definovanou takto:

$$f(0) = 1 \quad (2.18)$$

$$f(n+1) = (n+1) \cdot f(n) \quad (n \in \mathbb{N}_0) \quad (2.19)$$

Místo $f_{(n)}$ píšeme $n!$. Platí $0! = 1$. Pro $n \geq 1$ platí

$$n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot (n-1) \cdot n \approx \left(\frac{n}{e}\right)^n \cdot \sqrt{2\pi n} \quad (2.20)$$

Detailnější popis v [3 str. 156]

Z těchto výše uvedených aproximačních metod jsme se dozvěděli, jakým jiným způsobem jdou přibližně vypočítat některé matematické funkce. Některé z nich byly implementovány do programu příručního kalkulátoru. Byly brány v úvahu metody, které by aproximovaly dané funkce co nejpřesněji.

3. Architektura návrhového řešení

Dle zadání práce byla zvolena koncepce příručního kalkulátoru ve formě vývojového kitu s osmibitovým mikrokontrolérem. Navržený kalkulátor by dle zadání měl být schopen výpočtu základních matematických operací a dále by měl být rozšířen o výpočty pokročilých matematických funkcí (s omezenou přesností). Do kalkulátoru bude dále implementována prioritizace matematických operací a jeden paměťový registr.

Při návrhu byla rovněž uvažována možnost programovatelného módu. Vzhledem k nárokům na komplexnost softwaru byla tato možnost zamítnuta a kalkulátor byl rozšířen o možnost programování přes konektor ISP.

Architektura systému

Pro vlastní konstrukci byl zvolen rozšířený mikrokontrolér AVR s mnoha výhodnými vlastnostmi a vysokou výpočetní rychlostí. Protože se předpokládá výpočet pokročilých matematických funkcí pomocí iteračních vzorců, byl zvolen výkonný mikrokontrolér Atmel AVR ATmega32L s dostatečnou rychlostí a kapacitou programové paměti.

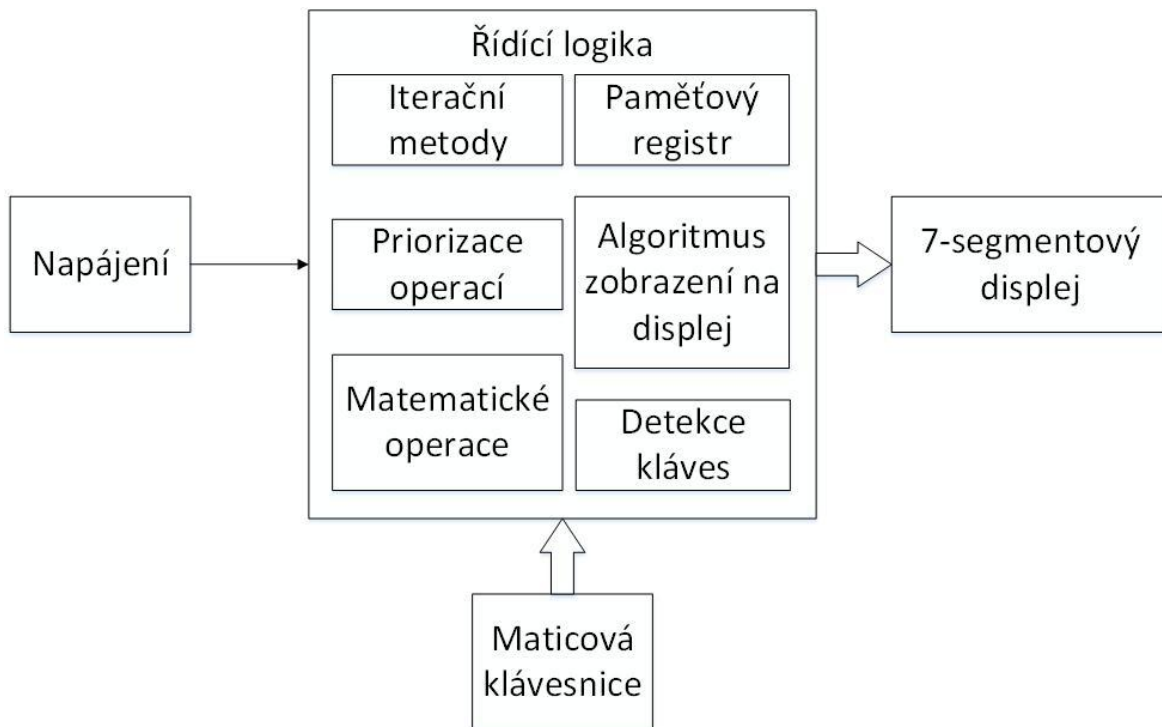
System byl navržen jako obdoba skutečných kalkulátorů. Předpokládá se, že uživatel bude vkládat data maticovou klávesnicí. Vzhledem k malému počtu tlačítek se předpokládá použití sekundárních funkcí. Zobrazování bude realizováno pomocí sedmisegmentových displejů, z důvodu jednoduché implementace a vzhledu zobrazení čísel.

Pro možnost budoucího rozšíření bude kalkulátor rozšířen programovatelným rozhraním.

Celkově bylo zařízení navrhováno jako samostatná jednotka s jednovrstvou deskou plošných spojů s centrální řídicí částí osazenou mikrokontrolérem, zobrazovací částí realizovanou ve formě sedmisegmentového displeje. Pro zadávání vstupů byla zvolena maticová klávesnice.

Takto navržená jednotka umožňuje implementaci i poměrně složitých algoritmů a poskytuje v kódovém segmentu dostatečný paměťový prostor.

Na následujícím obrázku (Obrázek 7) je znázorněno blokové schéma architektury systému.



Obrázek 7 - Architektura systému⁷

⁷ Vlastní zdroj

4. Hardwarové řešení

Příruční kalkulátor byl navržen na dvě jednovrstvé desky plošných spojů. První z nich reprezentuje základní desku kalkulátoru a druhý displej. Důvodem odděleného displeje byla možnost jeho použití i pro jiné aplikace. V příloze A a B jsou znázorněny schémata obou desek a v přílohách C a D jejich desky plošných spojů. Během návrhu byl kladen důraz na použití nízkého počtu součástek a snížení ceny celého zařízení. Z těchto důvodů byla provedena úprava řízení displeje. Viz kap 4.2

4.1. Základní deska kalkulátoru

Dle přílohy A je deska tvořena mikrokontrolérem, LDO stabilizátorem, baterií, kterou lze k základní desce připojit externě, USB mini B konektorem, který slouží pouze k napájení kalkulátoru, pinové lišty k připojení maticové klávesnice, k výstupu na displej. Dále je vybaven vypínačem kalkulátoru a přepínačem funkcí maticové klávesnice. Základní deska také obsahuje jeden volný port, který může sloužit k připojení dalšího zařízení, a programový konektor ISP, který nám slouží k nahrání programu do mikrokontroléru.

4.1.1. Mikrokontrolér

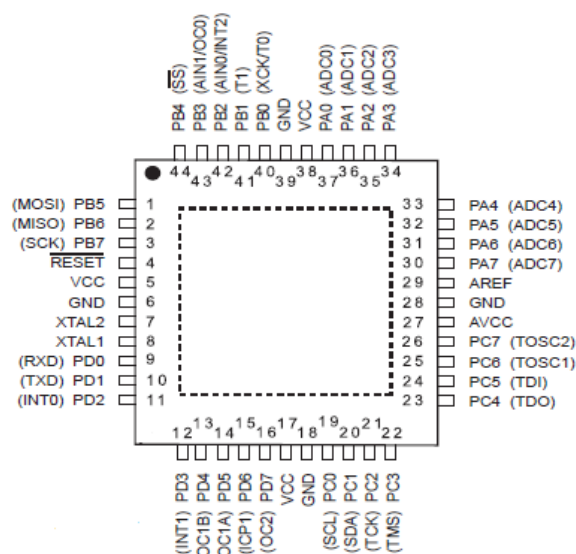
Hlavní funkcí mikrokontroléru je detekce kláves z maticové klávesnice, její zpracování, provedení výpočtu a poslání hodinového signálu a dat pro zobrazení na sedmisegmentový LED displej.

Hlavním řídicím prvkem kalkulátoru byl zvolen osmibitový mikrokontrolér rodiny AVR ATmega32L.

Parametry ATmega32L

Tento mikrokontrolér má interní taktovací kmitočet do 8 MHz. Významnou výhodou volby ATmega32L je ta, že může pracovat už od napájení 2,7 V, zatímco u klasické ATmega32 od 4,5 V. Obsahuje 32kB programovatelnou flash paměť, 1024B EEPROM paměť a 2kB SRAM. Dále obsahuje 32 programovatelných vstupních/výstupních portů (PORTA až PORTD) [14]

Externí krystal nebyl použit a to z toho důvodu, že maximální kmitočet externího krystalu, který lze pro tento typ mikrokontroléru použít, je 8 MHz a jak je uvedeno výše, tak interní taktovací kmitočet je také do 8 MHz.



Obrázek 8 - Rozložení pinů ATmega32L v pouzdře TQFP44⁸

4.1.2. Napájení kalkulátoru

Napájení kalkulátoru bylo navrženo pomocí USB přes PC, notebook nebo pomocí externího adaptéru. Napájet kalkulátor lze i pomocí externí baterie. Napájecí napětí bylo zvoleno 3,6V, z toho důvodu byl použit LDO stabilizátor, který má nízký úbytek napětí. Při napájení kalkulátoru pomocí baterie je nutné odpojit USB kabel.

LDO Stabilizátor

Jde o paralelní stabilizátor, který má nízký úbytek napětí tzv. LDO stabilizátor. Výstup ze stabilizátoru byl navržen na 3,6 V a maximální proud, který dokáže předat, je 300 mA. Na následující tabulce (Tabulka 3) je vidět výpočet maximálního odebíraného proudu.

Tabulka 3 - Výpočet maximálního odebíraného proudu⁹

Součástka	Počet	Proud
sedmisegmentový displej	9	144 mA
BU4094	9	90 mA
Maticová klávesnice	1	4 mA
Kontrolní LED	1	2 mA
ATmega32	1	8 mA
Celkem		248 mA

Tyto hodnoty jsou ovšem maximální. Proud, který prochází displejem, není konstantní, protože velikost proudu závisí na počtu rozsvícených segmentů.

⁸ Zdroj [14]

⁹ Vlastní zdroj

Maximální ztrátový výkon stabilizátoru je 800mW. Pro výpočet ztrátového výkonu použijeme následující vzorec.

$$P_{ztrátový} = (U_{vstupní} - U_{odebírané}) \cdot I_{odebíraný} \quad (4.1)$$

$$U_{vstupní} = 5 V \quad (4.2)$$

$$U_{odebírané} = 3,6 V \quad (4.3)$$

$$I_{odebíraný} = 0,248 A \quad (4.4)$$

$$P_{ztrátový} = (5 - 3,6) \cdot 0,248 = 0,3472 W = 347,2 mW \quad (4.5)$$

Jak je vidět z výsledku (4.5), tak vypočítaný ztrátový výkon je menší než maximální povolený ztrátový výkon.

Baterie

Při výběru baterie, která byla použita pro příruční kalkulátor, byla zvolena NiMH s napětím 3,6V. Proto byly vybrány tři nabíjecí baterie typu AA, které mají kapacitu zvolenou tak, aby kalkulátor vydržel alespoň několik hodin v provozu. Hlavní výhodou nabíjecích baterií typu AA je ta, že jsou k dostání na našem trhu. Nenabíjecí baterie se nesmějí používat, protože mají větší napětí ($3 \times 1,5 V$), které by mohlo ohrozit integrovaný obvod 4094. NiMH články neobsahují žádné nebezpečné látky, které by mohly ohrozit životní prostředí. Nabízely se tu další možnosti výběru např. Lithiová baterie nebo NiCd. Lithiové baterie nebyla použita a to z toho důvodu, že by musela být nabíjena přímo z kalkulátoru pomocí USB a musela by být kontrolována speciálním integrovaným obvodem. Další důvodem byla její cena, která je přibližně dvojnásobná oproti NiMH baterii. NiCd baterie jsou sice levnější, ale tento typ baterie také nebyl použit, protože obsahuje nebezpečnou látku pro životní prostředí zvanou kadmium. [7]

4.1.3. Maticová klávesnice

Při návrhu vstupní periférie šla realizovat několika způsoby. K omezenému počtu vstupních/výstupních portů bylo rozhodnuto, že vstupní periférie bude navržena jako maticová klávesnice. Samostatný návrh maticové klávesnice byl zavržen z důvodů možnosti koupě membránové klávesnice za přijatelnější cenu, která se externě připojí k základní desce kalkulátoru.

Další možností byla analogová maticová klávesnice. Její hlavní výhodou bylo použití pouze jednoho portu (port AD převodníku). Analogová klávesnice nebyla použita. Hlavním důvodem byla změna přechodového odporu tlačítek po opotřebení. S touto změnou by nebylo možné správně detekovat tlačítko, které bylo stisknuto.

Při určení počtu tlačítek bylo rozhodnuto použití maticové klávesnice 4x4.

Hlavní vlastností této vstupní periférie je vkládání čísel a funkcí pro ovládání příručního kalkulátoru. Maticová klávesnice 4x4 využívá osm portů mikrokontroléru k určení, které tlačítko bylo stisknuto. Principiální schéma maticové klávesnice je znázorněno na následujícím obrázku (Obrázek 10).

Rozložení kláves

Rozložení kláves bylo zvoleno podobně jako na klasických kalkulátorech, které jsou na trhu. Musíme si uvědomit, že maticová klávesnice 4x4 má pouze 16 tlačítek, která by nestačila na čísla a funkce, které byly do kalkulátoru naprogramovány. Proto byly přidány přepínače, které nám umožňují rozšířit funkce klávesnice. Na následujícím obrázku (Obrázek 9) je vidět rozložení kláves i jejich obě funkce.

7	8	9	+
\sqrt{x}	x^2	π	M+
4	5	6	-
x!	$\ln(x)$	$\log(x)$	M-
1	2	3	*
$\sin(x)$	$\cos(x)$	$\text{tg}(x)$	MR
.	0	/	=
+/-	CE	e^x	MC

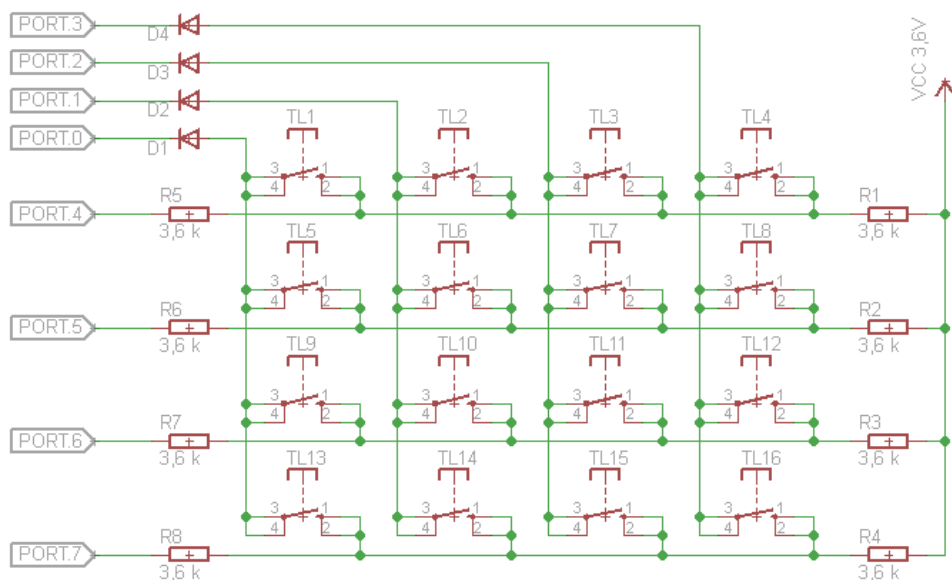
Obrázek 9 - Rozložení kláves na maticové klávesnice¹⁰

Princip funkce

Zvolená maticová klávesnice byla k mikrokontroléru připojena multiplexním způsobem pomocí 8 pinů. Při použití se předpokládá stisk pouze jedné klávesy v čase. Z těchto důvodů bylo zvoleno zapojení s usměrňovacími diodami, realizujícími společnou anodu.

Na následujícím obrázku (Obrázek 10) je znázorněno principiální schéma maticové klávesnice. Programové vybavení obsluhy maticové klávesnice následně realizuje vyčítání pomocí PORT.0 až PORT.3, které jsou nastaveny jako výstupní porty, a PORT.4 až PORT.7, které jsou nastaveny jako vstupní porty. Pokud není žádné tlačítko stisknuté, tak proud prochází přes odpory R1 až R8 do vstupů PORT.4 až PORT.7. Nastavíme-li logickou 0 na některý výstupní port, určíme, který sloupec tlačítek chceme detekovat. Například nastavíme-li na výstupu PORT.0 logickou 0, tak detekujeme stisk tlačítek TL1, TL5, TL9, TL13. Pokud za těchto podmínek stiskneme tlačítko TL1, tak proud neprotéká odporem R5, ale pouze diodou D1 a na vstupu PORT.4 je logická 0.

¹⁰ Vlastní zdroj



Obrázek 10 - Principiální schéma maticové klávesnice¹¹

Přepínač funkcí klávesnice

Pomocí přepínačů je možno přepínat funkce maticové klávesnice, jak je vidět z následující tabulky (Tabulka 4). Pokud tedy máme 4 funkce na každou klávesu, tak maximální počet funkcí, které může klávesnice používat, je 64. Jak je vidět na obrázku (Obrázek 9), tak v této práci je zapotřebí pouze dvě funkce na každé tlačítko. Další funkce jsou jen příprava na další možné vylepšení tohoto příručního kalkulačtoru.

Tabulka 4 - Pravdivostní tabulka funkcí klávesnice¹²

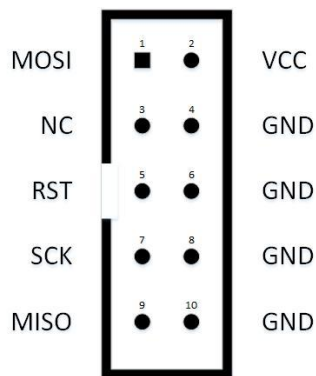
Přepínač 2	Přepínač 1	Funkce
0	0	1
0	1	2
1	0	3
1	1	4

4.1.4. Programovatelný konektor ISP

Pomocí ISP konektoru lze naprogramovat mikrokontrolér rodiny AVR, i když už je osazen na desce. Samozřejmě je zapotřebí si obstarat programátor, kterých je na trhu velké množství. Rozložení deseti pinového ISP konektoru je uveden na následujícím obrázku (Obrázek 11). Existuje i šesti pinový ISP konektor, ale ten už se dá pomocí redukce vytvořit.

¹¹ Zdroj [10] Eagle 6.4

¹² Vlastní zdroj



Obrázek 11 - ISP konektor (zapojení ATMEL - 5V)¹³

Při programování mikrokontroléru se musí odpojit baterie, protože má programátor jiné napětí než je napětí baterie. Dále se při programování musí odpojit displej, protože při napájení programátoru hrozí poškození integrovaných obvodů 4094. Programátor není určen k napájení zařízení, které potřebují vyšší proudy, mohlo by dojít k problémům při nahrávání programu do mikrokontroléru a zařízení by nemuselo pracovat správně.

4.2. Displej

Displej mohl být realizován více možnostmi (LCD displej, digitrony, sedmissegmentový displej) Nakonec bylo rozhodnuto použití sedmissegmentového displeje z důvodu výraznějšího zobrazení čísel. Při návrhu displeje bylo uvažováno použití devíti sedmissegmentových displejů. Pro snížení výkonu byly zvoleny sedmissegmentové displeje, které mají proud segmentem pouze 2 mA.

Displej mohl být řízen mnoha způsoby např. řízení pomocí demultiplexoru, řízení pomocí BCD/sedmissegment, pomocí obvodu 4094.

Při zapojení pomocí dekodéru BCD/sedmissegment je hlavní nevýhoda zobrazování desetinné tečky (pin DP), a proto tento způsob řízení nebyl použit. Dekodéry osahují většinou pouze čísla od 0 po 9 a některé speciální znaky. Realizace pro zobrazení desetinné tečky by musela být pomocí několika tranzistorů, které by byly řízeny demultiplexorem.

Při zapojení pomocí demultiplexoru jde o řízení, který rozhoduje o výběru displeje, který má svítit. U algoritmu zobrazování na displeji by muselo být zajištěno, že nejdříve vybere demultiplexorem první sedmissegmentový displej a mikrokontrolér vloží do vstupů první číslo. Poté druhý displej a druhé číslo atd. Tento celý algoritmus musíme opakovat s periodou T, která musí být menší 1/25 sekundy, abychom nezaznamenávali blikání displeje. Tento způsob nebyl vybrán, protože se našlo lepší řešení pomocí integrovaného odvodu 4094.

Nakonec bylo rozhodnuto řízení pomocí posuvného registru 4094, o kterém je popsáno dále. Hlavní výhodou tohoto způsobu je počet vstupů, které jsou VCC (napájení),

¹³ Vlastní zdroj

GND (zem), CLK (hodinový signál), DATA (datový signál). Ovšem musí se pro něj naprogramovat algoritmus sériového posílání dat na displej. Schéma displeje je znázorněno v příloze C.

4.2.1. Rozložené displeje

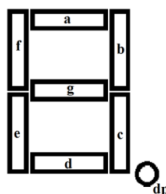
Zobrazení výsledného čísla bylo navrženo na displej tak, že první segment indikuje uložení do paměti, druhý záporné či kladné číslo a ostatní segmenty reprezentují číslo.

Pro velká nebo malá čísla je číslo na displeji zobrazeno tzv. vědeckým zápisem čísel. První segment indikuje uložení do paměti a druhý záporné či kladné číslo, třetí až šestý reprezentuje mantisu čísla, sedmý záporný či kladný exponent a osmý a devátý velikost exponentu. Rozsah čísel, který je možno zobrazit, je teoreticky od -9,999-99 do 9,999 99.

4.2.2. Sedmisegmentový displej

Následující oddíl je zaměřen na základní informace a typ zapojení sedmisegmentového displeje.

Displej je tvořen osmi LED diodami, které zobrazují sedm čar (čtyři svislé a tři vodorovné) a jednu tečku. Tímto způsobem lze vyjádřit všechny arabské číslice, desetinnou tečku a lze rovněž zobrazit další značky jako například mínus, signalizace chyby „--Error--“, nebo signalizace, že je v paměti uloženo číslo znakem „P“. [4 str. 20-23]



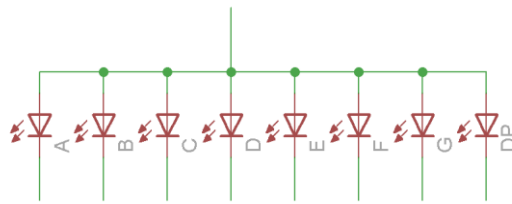
Obrázek 12 - Rozložení segmentů¹⁴

Existují dva druhy sedmisegmentového displeje podle typu zapojení. Zapojení se společnou anodou a se společnou katodou.

Společná anoda

U zapojení se společnou anodou obvod napájí všechny diody a pomocí řídicí logiky přivádíme na katody logickou 0 (GND), pokud chceme, aby dioda svítila, nebo logickou 1 (VCC), aby dioda nesvítila. Tento typ zapojením byl použit pro konstrukci displeje. U zapojení se společnou katodou je tomu právě naopak.

¹⁴ Vlastní zdroj



Obrázek 13 - Společná anoda¹⁵

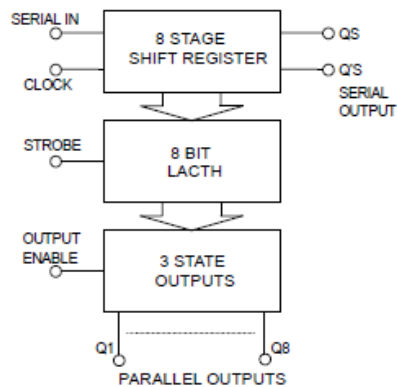
4.2.3. 4094

Následující oddíl je zaměřen na popis a princip funkce integrovaného obvodu 4094, který byl použit na konstrukci displeje.

Jedná se o osmibitový posuvný registr, který obsahuje osmibitovou západku, která umožňuje udržet hodnotu, která byla poslána pomocí mikrokontroléru. Západka posílá osm bitů na paralelní výstup. Výhoda tohoto integrovaného obvodu je ta, že se dá řadit v sérii. Tedy jich můžeme zapojit více za sebou. Pomocí mikrokontroléru musíme naprogramovat algoritmus posílání dat po bitech s časovou synchronizací. [15]

Princip funkce

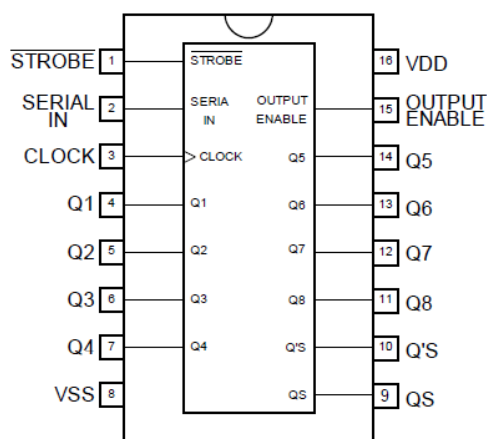
Tento integrovaný obvod je tvořen třemi bloky. První z nich je osmibitový posuvný registr, do kterého přivádíme data na pin SERIAL IN a hodinový signál na CLOCK. Piny QS a Q'S odpovídají sériovému výstupu. Druhý blok odpovídá západce, která ukládá osm bitů, které posíláme po sérii. Západku lze ovládat pomocí pinu STROBE. Třetí blok slouží k paralelnímu výstupu, který lze ovládat pomocí pinu ENABLE OUTPUT. [15]



Obrázek 14 - Blokové schéma 4094¹⁶

¹⁵ Vlastní zdroj Eagle 6.4

¹⁶ Zdroj [15]



Obrázek 15 - Rozložení pinů 4094¹⁷

Popis jednotlivých pinů

STROBE – Ovládání povolení osmibitové západky

SERIAL IN – Vstup pro sériová data

CLOCK – Synchronizační hodinový takt

Q1 až Q8 – Paralelní výstupy

QS – Sériový výstup, zde je možno připojit k dalšímu integrovanému obvodu 4094 do vstupu SERIAL IN

OUTPUT ENABLE – Ovládání povolení paralelního výstupu

VDD – Napájení integrovaného obvodu (od -0.3 do 18 V)

VSS – Zem

Piny STROBE a OUTPUT ENABLE jsou připojeny na napájení, aby bylo možné používat paralelní výstupy pro sedmsegmentový displej. Tato situace je znázorněna na prvním a druhém řádku níže uvedené pravdivostní tabulky 4094 (Tabulka 5). Nastává tu sice problém, že pokud jsou piny STROBE a OUTPUT ENABLE nastaveny vždy na logickou 1 a data jsou posílána po SERIAL IN, tak by mohlo být vidět, jak se jednotlivé segmenty při posílání dat mění. [15]

¹⁷ Zdroj [15]

Tabulka 5 - Pravdivostní tabulka 4094¹⁸

CLOCK	ENABLE OUTPUT	STROBE	SERIAL IN	Paralelní výstup		Sériový výstup	
				Q1	Qn	Qs	Q's
Náběžná hrana	H	H	L	L	Qn-1	Q7	NC
Náběžná hrana	H	H	H	H	Qn-1	Q7	NC
Náběžná hrana	H	L	X	NC	NC	Q7	NC
Náběžná hrana	L	X	X	Z	Z	Q7	NC
Sestupná hrana	L	X	X	NC	NC	NC	Qs
Sestupná hrana	H	H	X	Z	Z	NC	Qs

¹⁸ Zdroj [15]

4.3. Seznam součástek

Tabulka 6 - Seznam součástek¹⁹

Základní deska kalkulátoru			
Název součástky	Hodnota	Pouzdro	Počet
ATmega32L		TQFP44	1
LD6836TD/36H		SOT95	1
LUMBERG - 2486 01 - MINI USB TYPE B		32005-301	1
C1	100 nF	C0603	1
C2, C3	1 μ F	C0603	2
R1	71.5 Ω	R0805	1
R2, R3, R4, R5, R6, R7, R8, R9, R10, R11	3,6 k Ω	R0603	10
R12	10 k Ω	R0603	1
D1, D2, D3, D4	1N4148	SOD323-W	4
D5	MBR0520LT1G	SOD123	1
Tlačítko		B3F-1000	1
Přepínač 2 kolíbkový		MCNDS-02-V	1
LED dioda HLMP-4700 RED		LED5MM	1
Vypínač		09.03201.02	1
Pinová lišta 8x1 90°		1X08/90	1
Pinová lišta 4x1 90°		1X04/90	1
Pinová lišta 2x1		1X02	1
3M konektor 2x5 90°		PAK100/2500-5-10	2
Displej kalkulátoru			
Název součástky	Hodnota	Pouzdro	Počet
KINGBRIGHT - SA56-11LSRWA		BS-A554RD	9
BU4094BCFV-E2		S016	9
R1-R72	1,8 k Ω	R0603	72
Pinová lišta 4x1 90°		1X04/90	1

¹⁹ Vlastní zdroj

5. Softwarové řešení

Při návrhu softwarového řešení bylo rozhodnuto, že program bude naprogramován pomocí programovacího jazyku C. Tento program mohl být naprogramován také pomocí assembleru, ale tato možnost byla zavrhnuta z důvodů náročnosti programu. V následujících částech je popsán program, který byl naprogramován do kalkulátoru.

5.1. Vývojový diagram celého programu

V příloze E je znázorněn obecný vývojový diagram celého programu. Program lze rozložit na tři části. První částí je detekce tlačítek z maticové klávesnice. Druhou částí je vyhodnocení a výpočet a poslední částí je zobrazení na sedmsegmentovém displeji.

V následujících oddílech jsou znázorněny některé zajímavé části programu.

5.2. Detekce tlačítek

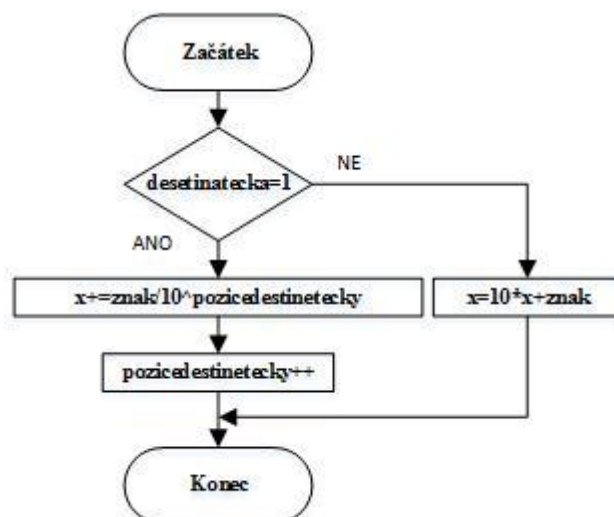
V následujícím odstavci je vysvětlen princip algoritmu pro detekci tlačítek maticové klávesnice.

Postupně dochází k detekci jednotlivých sloupců a při detekci každého sloupce dochází k podmínce zda je na PORTA.4 až PORTA.7 logická 0. Pokud je podmínka splněna, tak je do proměnné *znak* uložena hodnota, která odpovídá konkrétní funkci či číslu.

5.3. Vkládání čísel

Algoritmus vládání čísel se provádí v případě, že bylo stisknuto nějaké číslo od 0 až po 9. Při stisku tlačítka desetinné tečky se do proměnné *desetinatecka* nahraje 1.

U tohoto algoritmu bylo zapotřebí vzít v úvahu zadávání čísel celých i čísel desetinných. Jak je vidět na následujícím obrázku (Obrázek 16), algoritmus začíná podmínkou, zda bylo stisknuto tlačítko desetinné tečky. Pokud ano, tak se připočte k proměnné *x* podíl čísla, které bylo stisknuto na klávesnici, a mocninou *pozicedesetinatecky* o základu 10. Následně je proměnná *pozicedesetinatecky* inkrementovaná celočíselnou hodnotou 1. Pokud podmínka na začátku algoritmu splněna není, tak se původní číslo vynásobí deseti a připočte se k ní hodnota čísla, kterou jsme stiskli na klávesnici.



Obrázek 16 - Vývojový diagram vkládání čísel²⁰

5.4. Algoritmy iteračních vzorců

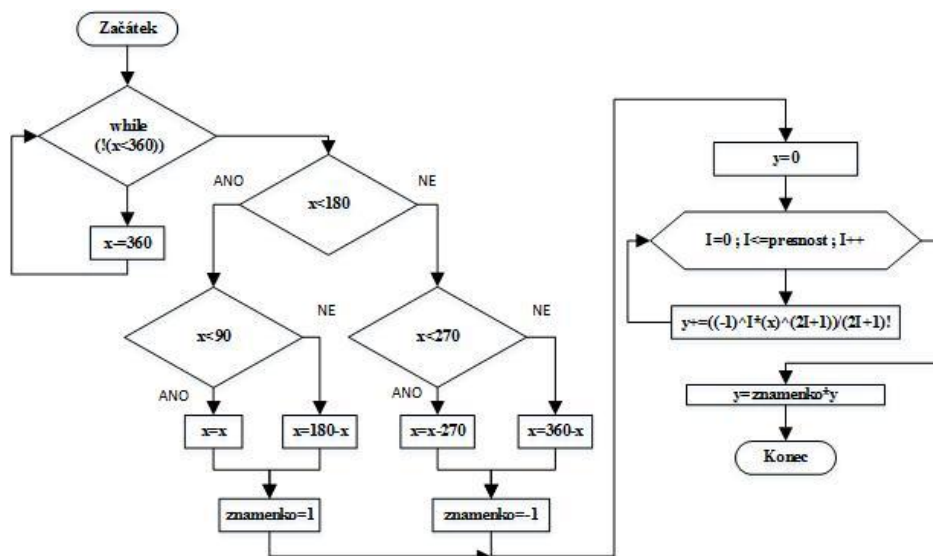
V následujícím oddíle jsou naznačeny algoritmy použitým iteračních vzorců, které jsou implementovány v programu příručního kalkulátoru.

5.4.1. Sinus

Pokud se stiskne tlačítko s funkcí sinus, následuje výpočet této funkce s parametrem, který je uložen v proměnné x a výsledek opět uložen do této proměnné. V následujícím odstavci je popsán algoritmus výpočtu této funkce.

Z vývojového diagramu znázorněného na následujícím obrázku (Obrázek 17) je vidět cyklus *while*, který postupně odečítá periodu (360°) do té doby, dokud je proměnná x větší než 360° . Následuje sada podmínek, které rozdělí, v jakém kvadrantu se úhel nachází. V těchto podmínkách dochází k přepočtu úhlu, aby byl v intervalu od 0° do 90° a znaménko výsledku je indikováno pomocí proměnné *znamenko*. Následuje cyklus *for*, který má určitý počet opakování v závislosti na přesnosti. V těle cyklu je znázorněn vzorec, který byl upraven podle vzorce (2.4). Po ukončení cyklu dojde k vynásobení výsledku s proměnnou *znamenko*. Hodnota byla uložena do proměnné x , se kterou je možno dále počítat. Proměnná y zde vystupuje jako pomocná proměnná pro výpočet. Parametr sinu se musí pro vzorec (2.4) zadávat v radiánech, a proto byl naprogramován algoritmus, který nám umožňuje přepočít ze stupňů na radiány. Hlavní výhodou tohoto algoritmu je možnost zadávat jako vstupní parametr úhly místo radiánů.

²⁰ Vlastní zdroj



Obrázek 17 - Vývojový diagram $\sin(x)$ ²¹

5.4.2. Cosinus

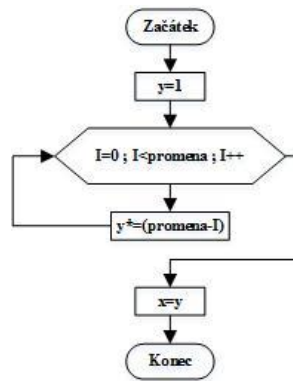
Algoritmus funkce cosinus je velice podobný jako u sinu. Jediná změna je pouze v použití vzorce v těle cyklu a proměnná *znamenko* se liší oproti sinu. U cosinu se musí použít iterační vzorec (2.5).

5.4.3. Faktoriál

Při stisku tlačítka s funkcí faktoriál dojde k výpočtu a výsledek je uložen do proměnné *x*. V následujícím odstavci je popsán algoritmus výpočtu této funkce. Nesmíme zapomenout, že před tímto algoritmem musí dojít k podmínce, že vstupní parametr je pouze nezáporné celé číslo.

Algoritmus je znázorněn na vývojovém diagramu na následujícím obrázku (Obrázek 18). Nejdříve pomocnou proměnnou *y* nadefinujeme na 1 z důvodu postupného násobení. U algoritmu výpočtu faktoriálu použijeme cyklus *for*. Počet opakování tohoto cyklu je v závislosti na proměnné *promenna*, která odpovídá číslu, u kterého chceme faktoriál vypočítat. V těle cyklu je nadefinováno postupné násobení čísel, která se snižují po jedné od *promenna* do 1. Na konci algoritmu je ještě do proměnné *x* uložena hodnota faktoriálu k dalšímu možnému výpočtu.

²¹ Vlastní zdroj



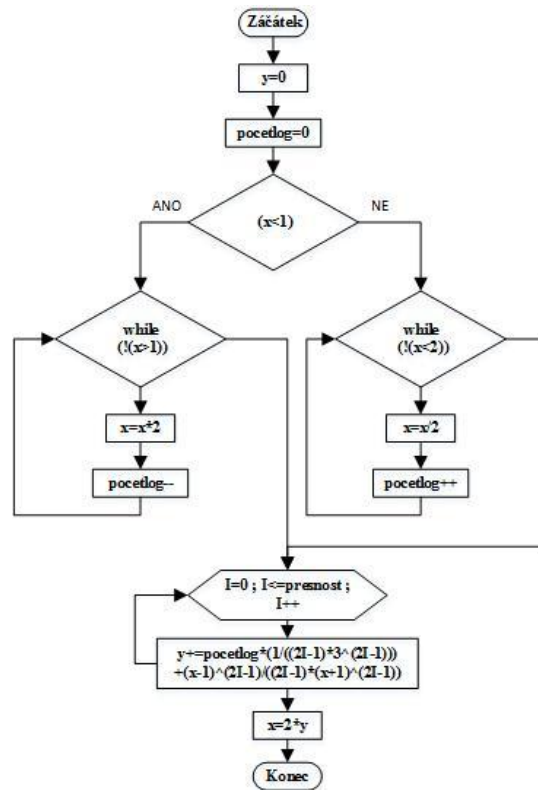
Obrázek 18 - Vývojový diagram výpočtu faktoriálu²²

5.4.4. Přirozený logaritmus

Pro výpočet logaritmu byl navržen algoritmus podle výše uvedené iterační metody (2.11). Tento algoritmus je popsán v následujícím odstavci.

Na následujícím obrázku (Obrázek 19) je znázorněn vývojový diagram algoritmu pro výpočet přirozeného logaritmu. Ještě před algoritmem dojde k podmínce, že hodnota parametru je větší než nula. Pokud by tato podmínka nebyla splněna, nedojde k výpočtu a na displeji se objeví chybová hláška. Pokud byla podmínka splněná, tak následuje vynulování pomocné proměnné y a $pocetlog$. Následuje podmínka, která určuje, zda hodnota parametru bude menší než 1. V obou větvích podmínky je vnořen cyklus *while*, který se opakuje do té doby, dokud není x menší než 2 (resp. větší než 1). V těle cyklu se x postupně dělí dvěma (resp. násobí dvěma) a inkrementuje (resp. dekrementuje) se proměnná $pocetlog$. Po ukončení cyklu následuje další cyklus, ale tentokrát cyklus *for*, který má určitý počet opakování v závislosti na přesnosti. V těle cyklu použijeme postupné sčítání podle vzorce $pocetlog * \ln(2) + \ln(x)$, kde se $\ln(2)$ a $\ln(x)$ vypočítají podle vzorce (2.11), z toho důvodu je na konci algoritmu výsledek vynásoben dvěma. Tento postup nám zaručuje větší přesnost výsledku, protože maximální parametr výpočtu přirozeného logaritmu je číslo 2.

²² Vlastní zdroj



Obrázek 19 - Vývojový diagram výpočtu přirozeného logaritmu²³

5.4.5. Dekadický logaritmus

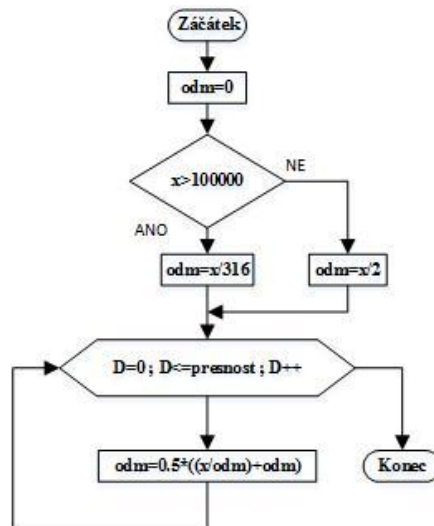
U dekadického logaritmu postupujeme naprosto stejně, jako u výpočtu přirozeného logaritmu až na výjimku, že výsledek podle vzorce (2.11) vydělíme koeficientem (2.12).

5.4.6. Druhá odmocnina

V následujícím odstavci je popsán algoritmus pro výpočet druhé odmocniny pomocí tzv. Newtonovy metody, o které bylo zmíněno v kapitole 2.

Na následujícím obrázku (Obrázek 20) je znázorněn vývojový diagram pro výpočet druhé odmocniny. Nejdříve proměnnou *odm*, která reprezentuje odhad odmocniny, vynulujeme. Poté dojde k podmínce, která určí, v jakém intervalu se číslo nachází a podle toho vypočítá první odhad. Proměnná *x* zde vystupuje jako parametr druhé odmocniny. Následuje cyklus *for*, který má určitý počet opakování v závislosti na přesnosti. V těle cyklu dojde k výpočtu nového odhadu, který je od předchozího přesnější. Po dokončení cyklu se druhá odmocnina ukládá v proměnné *odm*.

²³ Vlastní zdroj

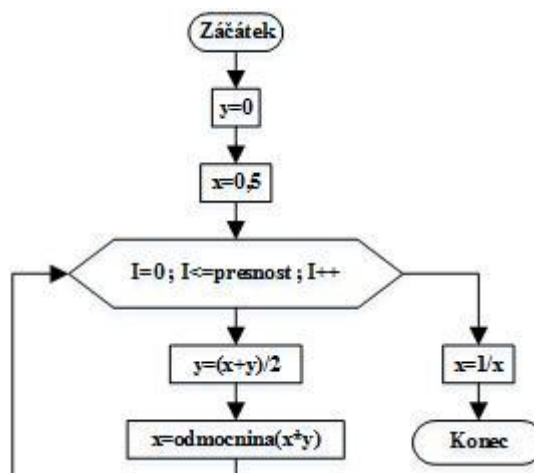


Obrázek 20 - Vývojový diagram pro výpočet druhé odmocniny²⁴

5.4.7. Ludolfovo číslo

V následujícím odstavci je popsán algoritmus výpočtu Ludolfova čísla pomocí iterační metody, která byla vysvětlena v kapitole 2.

Vývojový diagram tohoto algoritmu je znázorněn na následujícím obrázku (Obrázek 21). Nejdříve dojde k vynulování proměnné y a vložení čísla 0,5 do proměnné x . Následuje cyklus *for*, který má určitý počet opakování v závislosti na přesnosti. V těle cyklu dojde k aritmetickému průměru současných hodnot x a y . Následuje výpočet odmocniny ze součinu těchto proměnných. Po ukončení cyklu dojde k převrácení hodnoty proměnné x , která odpovídá přibližné hodnotě Ludolfova čísla.



Obrázek 21 - Vývojový diagram pro výpočet Ludolfova čísla²⁵

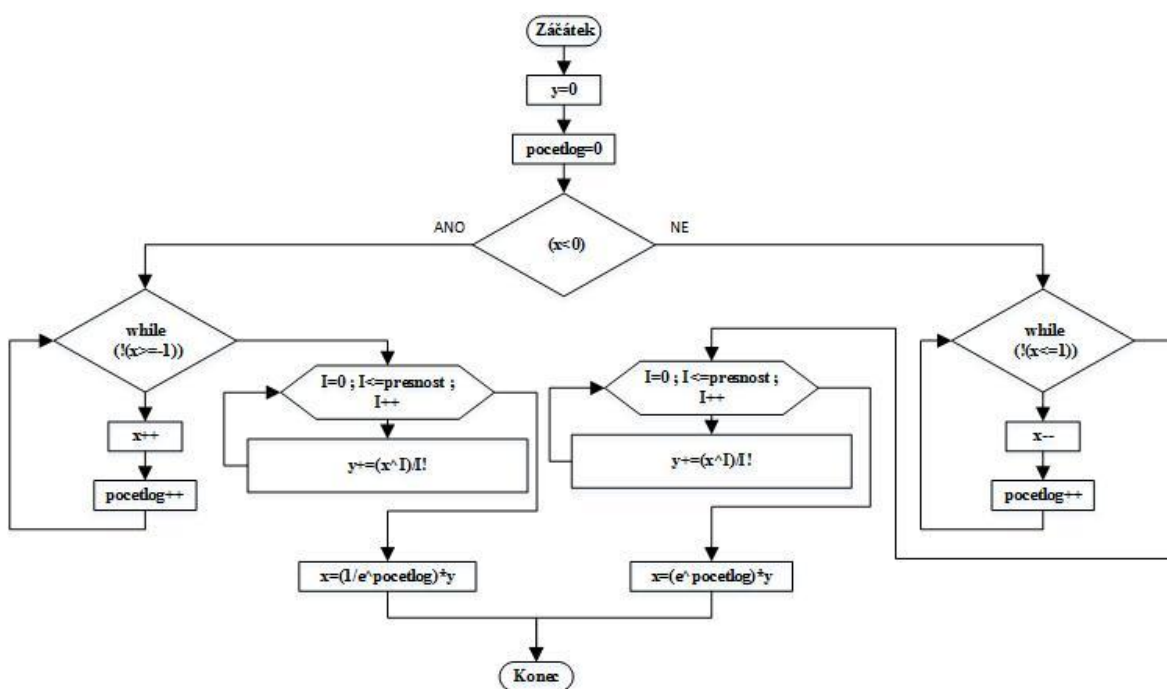
²⁴ Vlastní zdroj

²⁵ Vlastní zdroj

5.4.8. Exponenciální funkce

V následujícím odstavci je vysvětlen algoritmu výpočtu exponenciální funkce pomocí rozložení do řady (2.16).

Na začátku algoritmu dojde k vynulování proměnných y a $pocetlog$. Následuje podmínka, která nám určuje, jaké znaménko bude mít exponent. V obou větvích podmínky následuje cyklus *while*, který se opakuje do té doby než je číslo větší než -1 (resp. menší než 1). V těle cyklu se postupně inkrementuje proměnná x a $pocetlog$ (resp. dekrementuje x a inkrementuje $pocetlog$). Po dokončení cyklu následuje cyklus *for*, který má určitý počet opakování v závislosti na přesnosti. V těle cyklu použijeme postupné sčítání podle vzorce (2.16). Na konci algoritmu vynásobíme $e^{pocetlog}$ (resp. $1/e^{pocetlog}$) a proměnnou y a výsledek uložíme do proměnné x .



Obrázek 22 - Vývojový diagram pro výpočet exponenciální funkce²⁶

5.5. Výpočet a vyhodnocení

V následujícím odstavci je popsán algoritmus výpočtu základních matematických operací s prioritizací. Priorizace operací je pouze pro tři proměnné.

Pokud je stisknuto tlačítko sčítání, odčítání, násobení nebo dělení dojde k uložení hodnoty na displeji do jedné ze tří proměnných v závislosti na proměnné $zmena$, která určuje, kolikátá operace se provádí, a dojde k uložení identifikační hodnoty do proměnné co , které reprezentuje, jaká operace byla stisknuta. Pokud bychom překročili dvě výpočetní operace, dojde nejprve k výpočtu a výsledek je uložen do proměnné x , se kterým lze dále počítat.

²⁶ Vlastní zdroj

Pokud se stiskne tlačítko *rovná se*, tak dojde k výpočtu a prioritizaci operací. Nejdříve se rozhoduje, která operace se má vykonat jako první. Aby byla vykonána druhá operace jako první, musí být první operace buď sčítání nebo odčítání a druhá operace násobení nebo dělení. Jinak jde o postupné provádění operací. Na konci algoritmu jsou vynulovány pomocné proměnné a výsledek je uložen do proměnné *x*.

5.6. Zobrazení na displeji

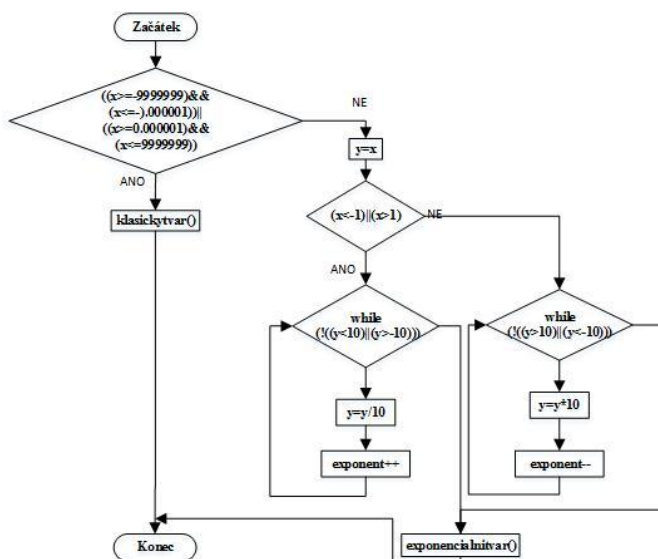
Tuto část programu lze rozdělit do několika pododdílů. Nejdříve musí dojít k zaokrouhlení, následuje zadání čísel a znamének do pole a nakonec poslání tohoto pole po bitech na displej s časovou synchronizací. Tyto algoritmy jsou popsány v následujících odstavcích.

5.6.1. Výběr typu zobrazení na displeji

Na následujícím obrázku (Obrázek 23) je vyobrazen vývojový diagram výběru typu zobrazení na displeji. Na začátku algoritmu je podmínka, která nám určuje, o jaké zobrazení půjde. Pokud je podmínka splněna, půjde o klasický zápis čísla (např. -12,45687), pokud ne, tak půjde o vědecký zápis čísla (např. -2,549-32).

U klasického zápisu čísla dochází k volání metody *klasickyvar()*, která nám rozkládá číslo na cifry a následuje vložení do pole. Tato metoda dále detekuje kladné či záporné číslo.

U vědeckého zápisu čísla nahrajeme do pomocné proměnné *y* výpočetní proměnnou *x*, aby se nám během tohoto algoritmu nezměnila. Následuje podmínka, která rozhoduje, zda půjde o kladný či záporný exponent. U obou větví je cyklus *while*, který nám upravuje hodnotu *y* a proměnou *exponent*. Následuje k volání metody *exponencialnitvar()*, která velice podobně rozkládá číslo a *exponent* na cifry a vkládá do pole, dále detekuje znaménko čísla i znaménko exponentu.



Obrázek 23 - Vývojový diagram pro výběr typu zobrazení na displej²⁷

²⁷ Vlastní zdroj

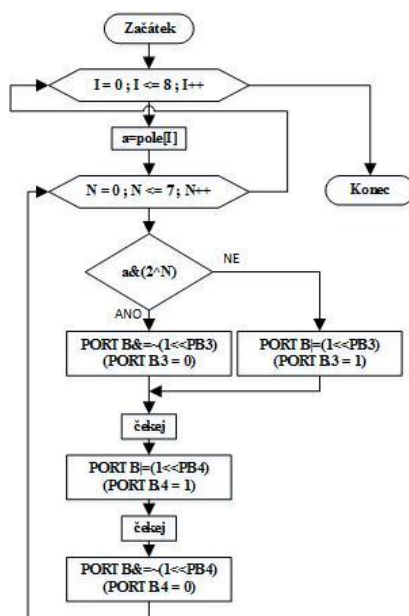
5.6.2. Vložení dat do pole

V této části dochází vkládání postupných cifer do devítimístného pole, které je poté posláno na displej. Postupně jsou ukládána jednotlivá čísla z proměnné y , která jsou pomocí příkazu *switch* přiřazeny konkrétní bitové kombinace do pole. Tato bitová kombinace reprezentuje kombinaci 1 a 0 v závislosti, který segment na displeji má a nemá být rozsvícený. S touto bitovou kombinací a proměnou *pozdesetinnatecka* proveden bitový součin, se kterou se mění nejvyšší bit. Tato operace rozhoduje o řízení desetinné tečky (pin DP).

5.6.3. Posílání dat z pole na displej

Pokud máme naplněné devítimístné pole, dojde k sériovému poslání dat na displej. Tento algoritmus je popsán v následujícím odstavci.

Na následujícím obrázku (Obrázek 24) je znázorněn vývojový diagram algoritmu pro posílání dat na displej. V tomto algoritmu jsou implementovány dva vnořené cykly *for*. Vnější cyklu slouží k uložení do proměnné a hodnota z I . místa pole. V těle vnitřního cyklu dochází k podmínce, která je realizována jako bitový součin proměnné a a mocniny dvou o základu N (index cyklu). Pokud je podmínka splněna, tak se na PORTB.3 pošle nula a pokud podmínka není splněna, tak se pošle jednička. Následuje hodinový takt, bez kterého by tento algoritmus nefungoval. Funkce integrovaného obvodu 4094 funguje na náběžnou hranu hodinového taktu.



Obrázek 24 - Vývojový diagram algoritmu posílání na displej²⁸

Pomocí výše uvedených algoritmů byl naprogramován program, který byl implementován do příručního kalkulátoru. Při návrhu programu byla snaha o nejvyšší přesnost a vysokou rychlost výpočtů.

²⁸ Vlastní zdroj

6. Dosažené výsledky

V této kapitole jsou zachyceny informace o dosažených výsledcích, jako například rychlost výpočtů matematických funkcí a výdrž baterie. Na následujícím obrázku (Obrázek 25) je zachycena fotografie výsledné konstrukce demonstračního příručního kalkulátoru.

Rychlosti výpočtů

V následující tabulce (Tabulka 7) jsou popsány rychlosti pokročilých matematických funkcí, které byly realizovány pomocí iteračních metod.

Tabulka 7 - Rychlost výpočtů funkcí²⁹

Funkce	Doba výpočtu
Sinus (1000)	32,978 ms
Cosinus (1000)	30,940 ms
Tangens (1000)	64,073 ms
Faktoriál (34!)	1,010 ms
Přirozený logaritmus (50)	12,936 ms
Dekadický logaritmus (50)	12,979 ms
Druhá odmocnina (200)	1,125 ms
Výpočet Ludolfova čísla	17,306 ms
Exponenciální funkce (5)	12,783 ms

U některých algoritmů je rychlost výpočtu závislá na velikosti zadaného parametru. Proto je u některých funkcí v tabulce (Tabulka 7) uvedena rychlost výpočtu pro konkrétní vstupní parametr funkce. Například při výpočtu goniometrické funkce na příručním kalkulátoru se nejdříve vstupní parametr upravuje, aby byl zajištěn co možná nejpřesnější výsledek, ale je tím ovlivněna rychlost výpočtu. Rychlosti výpočtu jsou dostačující pro kalkulátor.

Výdrž baterie

Při použití třech nabíjecích baterií typu AA s kapacitou 2100 mAh, by příruční kalkulátor měl teoreticky při maximálním proudu (rozsvícený celý displej) vydržet přibližně 8 hodin 30 minut. Hodnota kapacity může být libovolná, akorát tím ovlivňujeme výdrž kalkulátoru. Při otestování výdrže baterie v běžném provozu bylo zjištěno, že baterie vydrží přibližně 21 hodin, po této době sice ještě displej slabě svítí, ale už kalkulátor nereaguje na stisk maticové klávesnice.

²⁹ Vlastní zdroj



Obrázek 25 - Výsledná konstrukce příručního kalkulátoru³⁰

³⁰ Vlastní zdroj

7. Závěr

V práci byl popsán návrh vlastního příručního kalkulátoru, který byl vyvinut pro ověření iteračních metod a aproximačních algoritmů. Během práce vznikla deska řídicí části a deska displeje a do kalkulátoru byly vyvinuty softwarové algoritmy, které byly posléze úspěšně implementovány, a byla ověřena jejich funkčnost.

Vyvinutý kalkulátor má rozměry odpovídající příručnímu kalkulátoru. Konstrukce byla přizpůsobena pro demonstraci funkce kalkulátoru s řídicím mikrokontrolérem, a proto byl zvolen návrh montáže desek plošných spojů na plastovou destičku. Při ověřování hardwaru musel být vyřešen nedostatek. Nedopatřením byly prohozeny porty napájení a země u konektoru USB mini typ B. Tento nedostatek byl vyřešen přepájením na desce pomocí propojovacích drátků. Kalkulátor je možné přeprogramovat pomocí programátoru přes konektor ISP, což umožňuje další vylepšení funkcí kalkulátoru a vývoj není ukončen. Kalkulátor je možné napájet přes konektor USB mini nebo přes tři nabíjecí baterie typu AA.

Ze softwarové stránky je příruční kalkulátor schopen vkládat celá i desetinná čísla a vypočítat základní matematické operace s prioritizací, která je naprogramována na dvě operace. Tato nedokonalost bude do budoucna vyřešena programově. Dále jsou zde implementovány pokročilejší matematické funkce, které byly řešeny ze znalostí iteračních metod. Během návrhu softwarové části byly provedeny optimalizace některých výpočtových postupů s ohledem na zvýšení přesnosti a rychlosti výpočtu (zejména při výpočtu logaritmů). Celé softwarové řešení je naprogramováno bez implementace knihovny *math.h*. Displej je schopen teoreticky zobrazit rozsah čísel od $-9,999 \times 10^{99}$ do $9,999 \times 10^{99}$, ale při implementaci datového typu *double* je rozsah čísel omezen. Toto snížení rozsahu by se dalo vyřešit vytvořením nového datového typu, který by tento rozsah čísel umožnil. Kalkulátor disponuje dvěma možnostmi zobrazení na displeji (klasická a vědecká notace). Kalkulátor obsahuje paměťový registr, který umožňuje uložit hodnotu pro další možný výpočet.

Závěrem lze konstatovat, že se podařilo splnit všechny vytyčené cíle a příruční kalkulátor umožní provádět matematické výpočty.

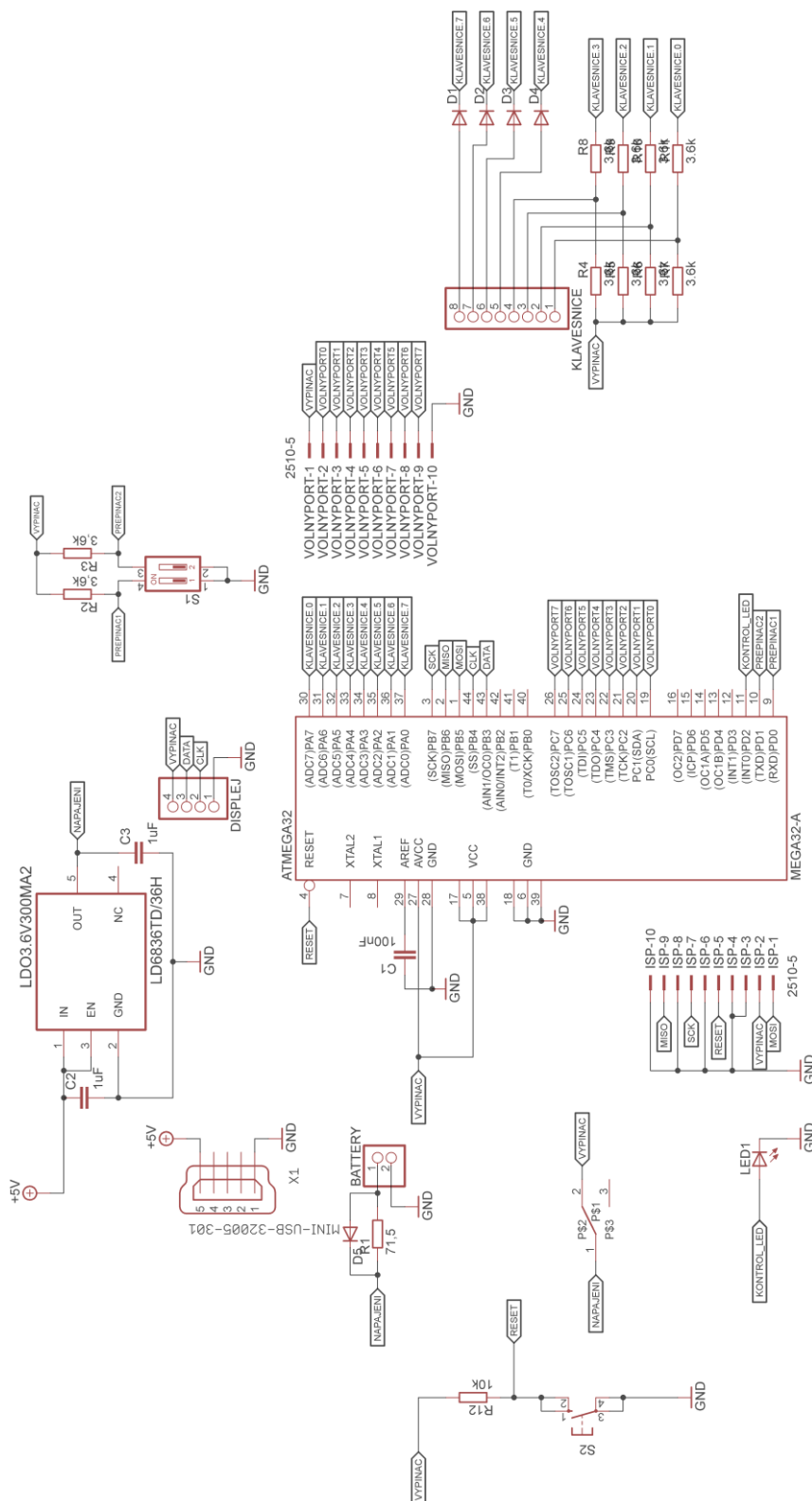
Seznam použité literatury

- [1] A Brief History of the Abacus. In: RYESON UNIVERSITY [online]. 2013 [cit. 2014-02-23]. Dostupné z: <http://www.ee.ryerson.ca/~elf/abacus/history.html>
- [2] Arithmometer. 2014. *Encyclopædia Britannica Online*. Retrieved 04 březen, 2014, Dostupné z: <http://www.britannica.com/EBchecked/topic/726021/Arithmometer>
- [3] BARTSCH, Hans-Jochen. *Matematické vzorce*. 3. vyd. Praha: Mladá fronta, 2002, 831 s. ISBN 80-204-060sedmi7.
- [4] CSÁKÁNY, Antal. *Co umí kapesní kalkulátor*. 1. vyd. Praha: SNTL, 1982. ISBN 04-021-82
- [5] DAVID G. HICKS. *The Museum of HP Calculators* [online]. 1995 - 2013 [cit. 2014-03-06]. Dostupné z: <http://www.hpmuseum.org/>
- [6] *History of computers* [online]. 27.2.2014 [cit. 2014-03-03]. Dostupné z: <http://history-computer.com/>
- [7] HUSÁK, Miroslav. *Návrh napájecích zdrojů pro elektroniku: přednášky*. Vyd. 1. Praha: Česká technika - nakladatelství ČVUT, 2006, 154, v s. ISBN 80-010-3398-8.
- [8] Leibniz's Calculating Machine. *Leibnitiana* [online]. 2002 [cit. 2014-03-02]. Dostupné z: <http://www.gwleibniz.com/calculator/calculator.html>
- [9] MRÁZEK, Jiří. *Hry s kalkulátory*. 3. vyd. Praha: SNP, 1988. ISBN 14-494-88.
- [10] PK DESIGN. *Modul 4 LED displejů, klávesnice a LCD rozhraní v1.0*. 19.4.2005. Dostupné z: http://www.pk-design.net/Datasheets/Modul_Disp_LED_LCD_matrix_kbd_bpr_v10.pdf
- [11] REKTORYS, Karel. *Přehled užití matematiky*. Praha: SNTL, 1981. ISBN 04-003-81.
- [12] Slide rule. 2014. *Encyclopædia Britannica Online*. Retrieved 04 březen, 2014, Dostupné z: <http://www.britannica.com/EBchecked/topic/548710/slide-rule>
- [13] *Vintage Calculator Web Museum* [online]. © 2000-2014, 29.12.2013 [cit. 2014-02-24]. Dostupné z: <http://www.vintagecalculators.com/>

Datasheety:

- [14] AVR ATmega32L, *Atmel*
- [15] BU4094BCF, *ROHM SEMICONDUCTOR*

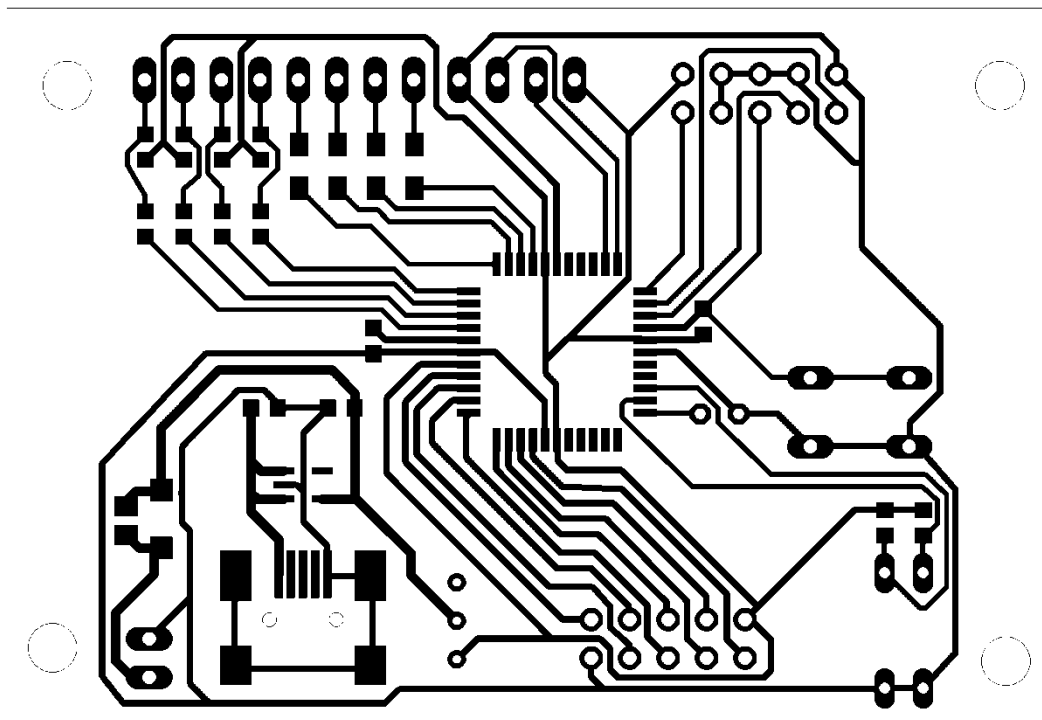
Příloha A - Schéma zapojení základní desky kalkulatoru



Obrázek příloha 1- Schéma zapojení základní desky kalkulatoru³¹

³¹ Vlastní zdroj Eagle 6.4

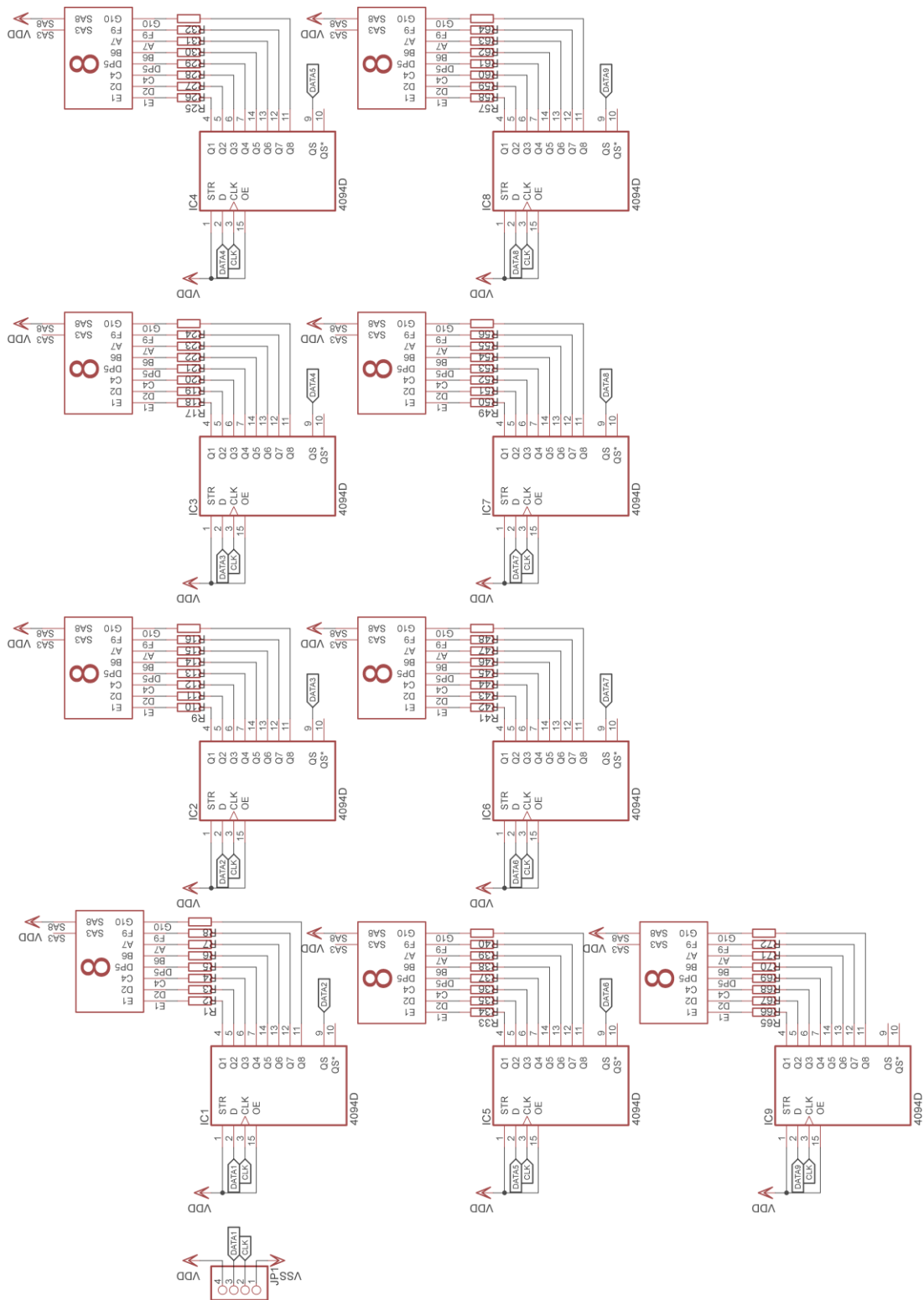
**Příloha B - Deska plošných spojů základní desky kalkulátoru
(měřítko 2:1)**



Obrázek příloha 2 - Deska plošných spojů základní desky kalkulátoru (měřítko 2:1)³²

³² Vlastní zdroj Eagle 6.4

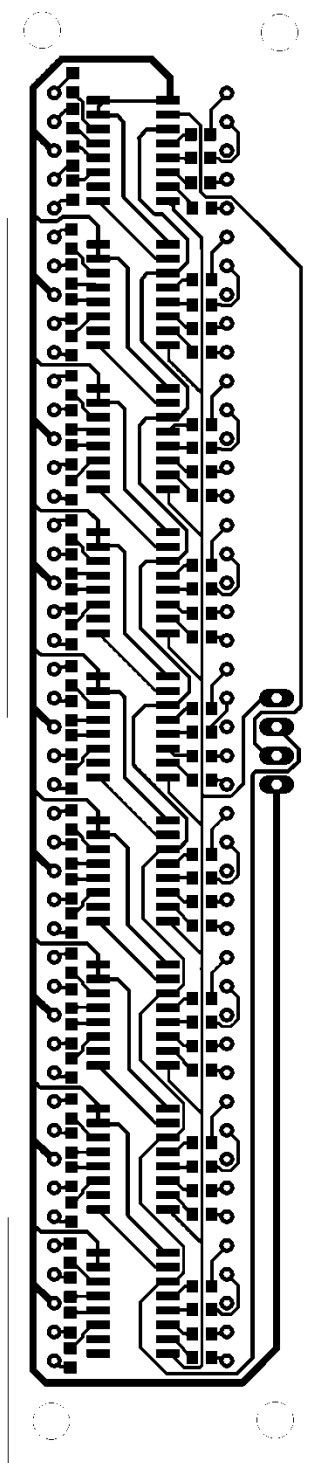
Příloha C - Schéma zapojení displeje



Obrázek příloha 3 - Schéma zapojení displeje³³

³³ Vlastní zdroj Eagle 6.4

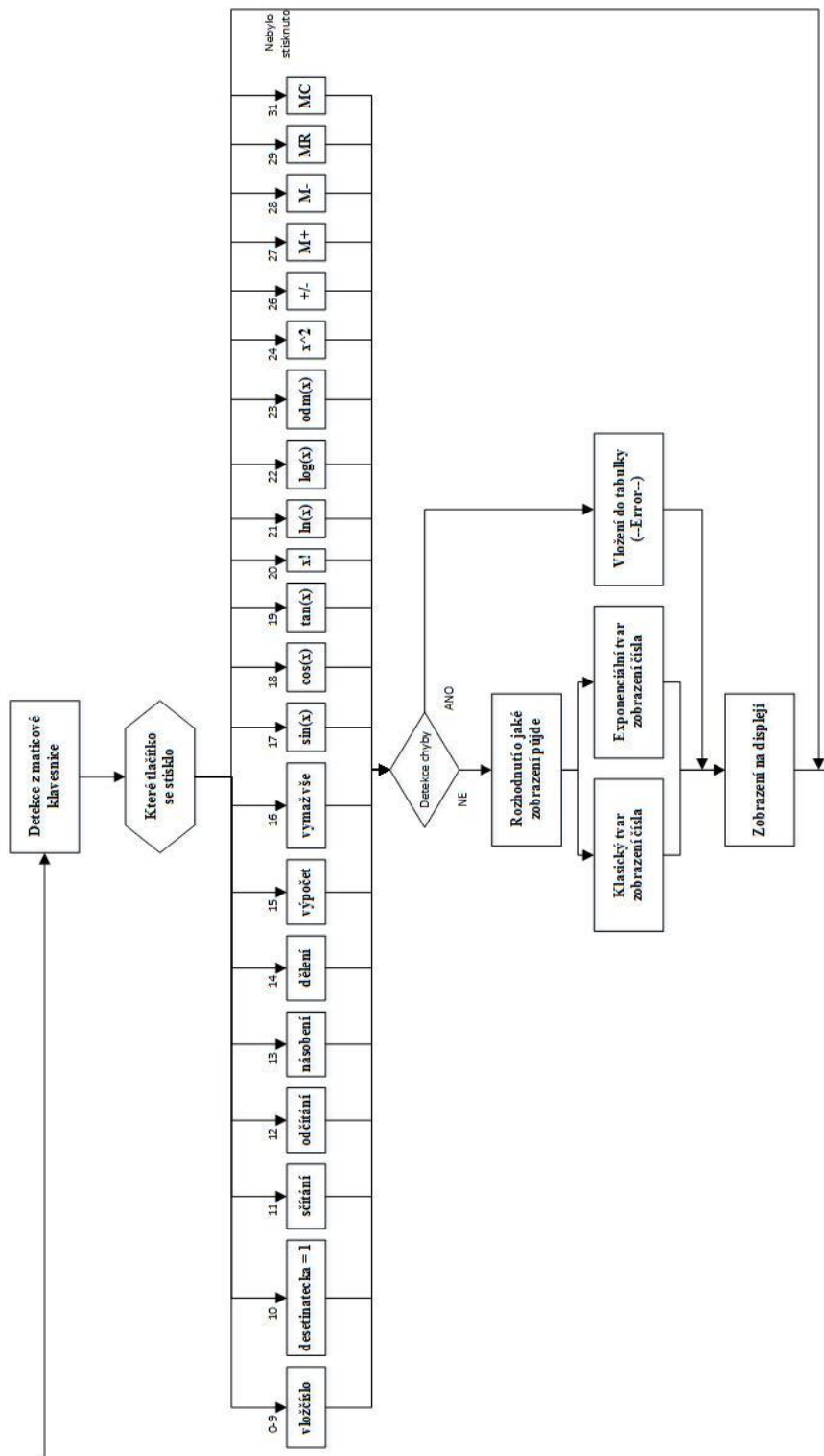
Příloha D - Deska plošných spojů displeje (měřítko 3:2)



Obrázek příloha 4 - Deska plošných spojů displeje (měřítko 3:2)³⁴

³⁴ Vlastní zdroj Eagle 6.4

Příloha E - Vývojový diagram programu



Obrázek příloha 5 - Vývojový diagram programu³⁵

³⁵ Vlastní zdroj

Příloha F - CD-ROM

Obsah CD

- Bakalářská práce ve formátu PDF
- Zdrojový kód příručního kalkulátoru v jazyku C
- Návrh hardwarového řešení v programu Eagle 6.4