

**Vyšší odborná škola a Střední průmyslová škola
elektrotechnická Olomouc,
Božetěchova 3**

**PRAKTICKÁ ZKOUŠKA Z ODBORNÝCH
PŘEDMĚTŮ**
Model klavíru



ZADÁNÍ PRAKTICKÉ ZKOUŠKY Z ODBORNÝCH PŘEDMĚTŮ

pro: **Martina VRLÍKA**

Studijní obor: **26-41-M/01 Elektrotechnika**
Počítačové a informační systémy

Třída: **4A,**

Ředitelství Vyšší odborné školy a Střední průmyslové školy elektrotechnické Olomouc Vám podle vyhlášky Ministerstva školství, mládeže a tělovýchovy č. 177/2009 Sb., o bližších podmínkách ukončování vzdělávání ve středních školách maturitní zkouškou, ve znění vyhlášky č. 90/2010 Sb., vyhlášky č. 274/2010 Sb., vyhlášky č. 54/2011 Sb. a vyhlášky č. 273/2011 Sb., určuje tuto praktickou zkoušku z odborných předmětů.

Téma: **Model klavíru**

Způsob zpracování a pokyny k obsahu:

- Klavír je řízený procesorem fy ATMEL.
- Program je vytvořený v assembleru, bude doplněn vývojovým diagramem.
- Klavír bude obsahovat veškeré tóny včetně půltónů, které jsou na běžném klavíru v podobě menších černých kláves.
- Klavír bude doplněn tlačítky s možností hraní melodie.
- Práce bude obsahovat manuál celého zařízení.

Rozsah: 25 až 35 stran

Kritéria hodnocení: Hodnocení práce probíhá ve třech fázích.

Průběžné hodnocení zohledňuje postupné plnění zadaných úkolů, dodržování termínů, míru samostatnosti žáka. Hodnocení závěrečné posuzuje míru splnění všech požadavků vyplývajících ze zadání práce a funkčnost produktů. Hodnocena je přehlednost, úplnost, srozumitelnost a formální stránka textové části práce. Hodnocení obhajoby práce zahrnuje způsob a srozumitelnost projevu, vzhled prezentace, odpovědi na dotazy.

Počet vyhotovení: 1 výtisk

Vedoucí práce: Ing. Zuzana VESELÁ

Datum zadání: 14. října 2013

Datum odevzdání: 28. března 2014

V Olomouci dne 14. října 2013

VYŠŠÍ ODBORNÁ ŠKOLA 2
a STŘEDNÍ PRŮMYSLOVÁ ŠKOLA
ELEKTROTECHNICKÁ
Božetěchova 3, Olomouc 772 00
tel. 585 208 121 fax 585 208 121

Zadání převzal dne 22.10.2013...

Martin Vrlík
podpis studenta

Prohlašuji, že jsem praktickou zkoušku vypracoval samostatně a všechny
prameny jsem uvedl v seznamu použité literatury.

.....

Martin Vrlík

Chtěl bych vyslovit poděkování paní ing. Zuzaně Veselé za odborné konzultace
a poskytnuté informace.

.....

Martin Vrlík

Prohlašuji, že nemám námitek proti půjčování nebo zveřejňování mé práce nebo
její části se souhlasem školy.

.....

Martin Vrlík

OBSAH

Obsah	4
Úvod.....	5
1. Procesor.....	6
1.1. Časovače.....	7
1.2. Generování tónu	9
1.2.1. Délka tónu	10
1.2.2. Výpočty pro nastavení časovače	11
1.2.3. Tabulka vypočítaných časů	13
1.2.4. Ztlumení tónu	15
2. Klávesnice	17
2.1. Zapojení klávesnice	18
3. Výrobní dokumentace	19
3.1. Schéma výrobku	19
3.2. Návrh DPS.....	20
3.3. Seznam součástek.....	22
3.4. Fotodokumentace	23
3.5. Vývojový diagram	24
4. Návod k použití	25
Závěr	26
Seznam použité literatury a studijních materiálů	27
Seznam obrázků, grafů a tabulek	28

ÚVOD

Důvod proč jsem si zvolil tuto práci je ten, že jsem chtěl ukázat možnosti využití procesorů pro jiné účely než jen ovládání a řízení robotů a podobných věcí.

Samotný princip generování tónu se může zdát být jednoduchý, ale jde o přesnost všech výpočtů a zároveň o reprodukci tónu tak, aby zněl co nejvíce realisticky, jak je to jen možné. Mým cílem tedy je sestavení téměř plnohodnotné verze klavíru s různými dodatečnými funkcemi jako například přehrávání již naprogramovaných skladeb. Klavír by tedy měl být schopen tzv. Karaoke, které dnešní elektronické klavíry podporují. Další zvláštností u klavíru je také fakt, že tón stlačené klávesy nehraje neustále, ale pomalu se ztišuje. Tato vlastnost skutečného klavíru je velmi důležitá a proto jsem se rozhodl i tuto specifickou převést do elektronické formy.

Jak jsem již zmínil na začátku úvodu, práce ukazuje na další skutečné využití procesorů. Klavír by se tedy dal použít buď na běžné hraní anebo jen na ukázky toho, na co lze využít středoškolskou znalost programování.

V jednotlivých kapitolách detailně popisují jak samotný integrovaný obvod, tak i jednotlivé bloky zapojení. Poměrně zajímavou součástí je blok, který se stará o ztlumení tónu, který je neustále delší dobu stlačený.

1. PROCESOR

Srdcem celého zařízení je procesor firmy ATMEL a to konkrétně typ AT89S52. Důvod proč jsem zvolil právě tento typ, je ten, že jsem vyžadoval velkou programovou paměť a právě tento typ má 8kB, což mé práci bohatě postačuje. Jelikož jsem celý výrobek zasadil do zbytků starého již nefunkčního elektronického klavíru. Zároveň jsem tak dostal možnost využít slotu na 4 tužkové baterie, které mi poskytnuly napětí 6V. Vzhledem k tomu, že procesor má rozsah napájecího napětí od 4,0V do 5,5V měl jsem na výběr dvě možnosti, buď napětí srazit dvěma diodami, nebo využít stabilizátoru 7805. Vybral jsem stabilizátor z důvodů možného použití napájecího zdroje. Stabilizátor k plné funkci vyžaduje alespoň 7V, ale jelikož má Atmel minimální napájecí napětí 4V a tužkové baterie mají většinou něco málo přes jejich udávané napětí, stejně tak i zdroj takže vše funguje bez problémů.

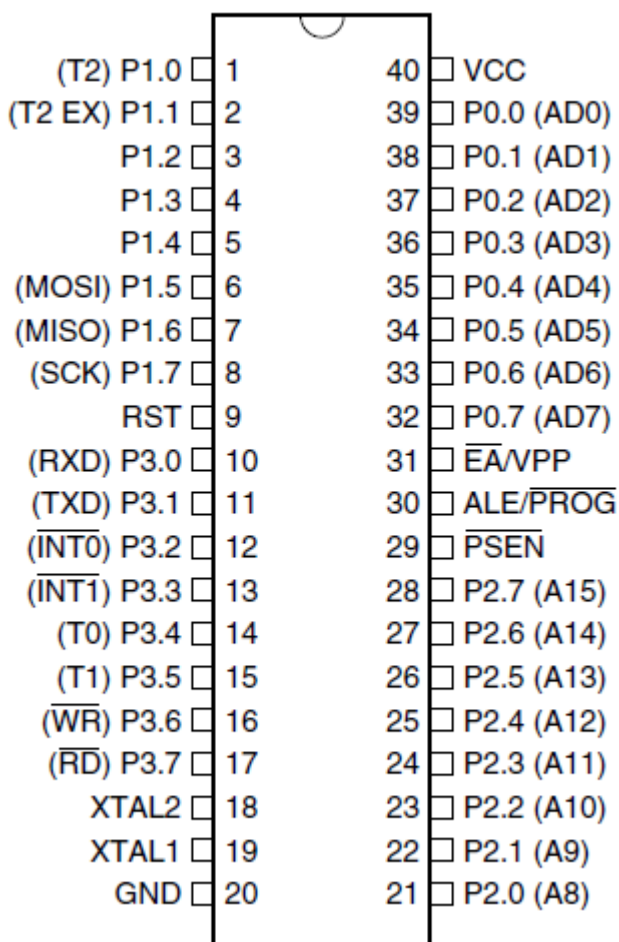
Základní parametry AT89S52:

- napájecí napětí 4,0V až 5,5V
- 8kB programovatelné Flash paměti
- 32 Programovatelných vstupů/výstupů
- tři 16-bitové Časovače/Čítače

Zmíněné parametry jsou jen ty hlavní a také ty, co jsem využil. Zbylé parametry jako například nízká spotřeba, rychlý programovací čas a rozsah hodnot krystalu jsem vypustil záměrně.

Použitý procesor je DIL40 z čehož tedy plyne, že má kompletní 4 programovatelné porty po 8mi bitech. Plně využité jsou 2 porty využití klávesnicí (P3 a P2). Na portu P1 jsou využity 2 piny pouze a jen pro audio část klavíru. Poslední port P0 zůstal nevyužitý s plány na budoucí možné použití například pro sedmissegment na zobrazování aktuálně přehrávané předprogramované melodie.

Na procesoru zabírá program samotné klávesnice s tóny téměř polovinu jeho kapacity. Což mi dává možnost druhou polovinu využít na speciální funkce a také na vytvoření několika dalších písniček. Další možností využití volného prostoru je již zmíněný sedmissegment, který by zobrazoval číslo aktuálně hrané skladby.

Obrázek 1-1: Rozložení pinů v pouzdře

1.1. ČASOVAČE

Procesor, který jsem si vybral, má k dispozici 3 časovače, které mohou pracovat i jako čítače. První dva časovače označené jako 0 a 1 mají možnost pracovat v 8b, 13b a nebo v 16b režimu. V mém případě jsem využil časovač 0 pro celou práci. Využívaný časovač pracuje v 16b režimu což tedy znamená, že mám možnost vytvořit čas o délce 65536 μ s bez nutnosti časovač opakovaně plnit.

Maximální délku času, kterou je časovač schopný vytvořit vypočítáme jednoduše. Jak jsem již zmínil, můj časovač pracuje v 16b režimu a tak tedy celkový čas v μ s je $2^{16}=65536\mu$ s. Číslo 2 je počet kombinací, tedy 1 a 0 a toto číslo umocníme na režim časovače tj. na šestnáctou. Z tohoto faktu vidíme, že 8b časovač je schopný vytvořit čas 256 μ s a 13b časovač zase 8192 μ s.

Zbylé výpočty uvedu v kapitole tomu věnované. Každý čas je nutné vždy doladit, protože čas nikdy není úplně přesný a obvykle je delší zhruba o 13-15 μ s což je hlavně v případě generování přesných tónů nepřijatelné. Tato nepřesnost má totiž velký vliv na výšku tónu.

Časovače 0 a 1 jsou řízeny registrem TCON, tento registr je také adresovatelný po bitech. Horní polovina tohoto Bytu obsahuje bity pro zapnutí a vypnutí časovače 1 a 0. Zatím co dolní polovina obsahuje bity, které obstarávají nastavení pro přijetí vnějšího přerušení a aktivaci vnějšího přerušení pro časovače 1 a 0.

Registr TCON:

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
b7	b6	b5	b4	b3	b2	b1	b0

Kde: TFX - bit naplnění (nastaví se na 1 při přetečení max. hodnoty dané módem)
 TRx - bit sloužící k zapnutí nebo vypnutí čítače
 (log 1=zapnuto, log 0=vypnuto)
 IEx - bit přijetí vnějšího přerušení
 ITx - bit konfigurace aktivace vnějšího přerušení

Pro nastavení módu časovače 1 a 0 se využívá registr TMOD. Tento registr opět pomyslně rozdělíme na horní a dolní polovinu. Horní polovina obsahuje prvky pro nastavení časovače 1. Máme tedy 4b, z toho 2b slouží k nastavení námi zvoleného módu. Ze zbylých 2b jeden slouží k výběru režimu časovače nebo čítače a druhý umožňuje nastavit čítač vstupem vnějšího přerušení. Dolní polovina je naprosto shodná s horní polovinou, pouze obstarává nastavení časovače 0.

Registr TMOD:

GATE	C/ $\bar{T}$	M1	M0	GATE	C/ $\bar{T}$	M1	M0
b7	b6	b5	b4	b3	b2	b1	b0

Kde: GATE - umožňuje nastavit čítač vstupem vnějšího přerušení
 - log 0 = časovač řízen pouze TRx
 - log 1 = časovač řízen pomocí TRx i INTx
 C/ $\bar{T}$ - nastavuje režim čítač nebo časovač
 - log 0 = režim časovač
 - log 1 = režim čítač
 M1,0 - nastavuje mód časovače

Tabulka 1-1: Kombinace nastavení bitů M1 a M0:

M1	M0	Mód
0	0	13b
0	1	16b
1	0	8b

Pro uvedení příkladu, jak vypadá již plně nastavený a funkční časovač, zde vložím program časovače obstarávající dobu $\frac{1}{2}$ periody prvního tónu.

```
C1:  mov tmod,#01h      ;(nastaví 16b mód)
      mov TH0,#0F8h     ;(nastavení časových parametrů)
      mov TL0,#95h     ;(nastavení časových parametrů)
      setb TR0          ;(spustí časovač)
      clr TF0           ;(preventivně vynuluje registr přetečení)
SE1: jnb TF0,SE1        ;(čekání na naplnění registru přetečení)
      clr TR0           ;(zastaví časovač)
      ret               ;(ukončení podprogramu)
```

Pro nastavování módu přes TMOD máme možnost buď nastavit mód pomocí binární kombinace nebo v mém případě za pomoci hexa kódu. Jelikož používám časovač 0, využívám tedy jen dolní polovinu bytu a je pro mě výhodnější napsat #01h než rozepisovat #00000001b. V obou případech časovač funguje, jak má. V případě potřeby použití 8b časovače stačí napsat #02h. Tento styl vybírání módu mi vyhovuje více a zabírá i méně místa. Z pohledu velikosti programu stačí jen ubrat pár řádků, kde se dá a program se tak poměrně efektivně zmenší z hlediska velikosti.

1.2. GENEROVÁNÍ TÓNU

Jelikož jsme nuceni generovat tón digitálně a ne analogově, generujeme zvolený tón jako obdélníkový signál, tedy vysíláním log 1 a log 0 na vstup zesilovače s reproduktorem po dobu půl periody. Vysílaný signál vychází na portu P1 z pinu 0 a jde přímo na vstup zesilovače LM386.

Vzor jedné periody generovaného tónu:

```
PER1: setb P1.0        ;(vyšle na výstup log 1)
      call C1           ;(zavolá čas o délce  $\frac{1}{2}$  periody)
      clr P1.0          ;(vyšle na výstup log 0)
      call C1           ;(zavolá znovu čas  $\frac{1}{2}$  periody a dokončí tak periodu)
      ret               ;(ukončí podprogram)
```

1.2.1. DÉLKA TÓNU

Každý tón, hlavně v různých písničkách není nikdy ve všech případech stejně dlouhý. Různé délky tónů jsou nedílnou a velmi podstatnou součástí hraní. Pokud by bylo vše stejně dlouhé, pak v případě použití karaoke nebude vycházet čas na zpívání a hudba by tak zvané utíkala. Z tohoto důvodu jsem do každého tónu vložil kousek kódu, který mi umožňuje nastavit jeho délku. Takto mám možnost nastavit jeho délku až na 0,2s což ve všech případech vystačuje. Pokud by bylo třeba delšího tónu, stačí dát dva stejné tóny za sebe.

Část kódu ovlivňující délku tónu:

TON1:

```
O1:  call PER1      ;(volání periody daného tónu)
      djnz R1,O1     ;(kontrola registru R1 určujícího délku tónu)
      ret           ;(ukončení podprogram)
```

V případě, že jsme do registru R1, který určuje délku tónu, vložily jakoukoli hodnotu v rozmezí 0-255 bude instrukce djnz opakovat danou periodu dokud se registr nevynuluje. Instrukce funguje tak, že porovnává hodnotu v registru s nulou. Jestli má registr nenulovou hodnotu program skočí na uvedené návěští O1 a odečte z registru 1.

V původním plánu bylo, aby měly všechny tóny již danou délku, ale z hlediska různých frekvencí by se délky tónů lišily a ladění by bylo komplikované.

Pro příklad uvedu část jedné naprogramované písničky, u které jsem využil proměnnou délku tónů.

Příklad z “Holka modrooká”:

```
mov R1,#100      ;(Nastavení délky běžného tónu)
call TON42       ;F
call DEL2        ;(Zpoždění mezi jednotlivými tóny)
mov R1,#255      ;(Nastavení delšího tónu)
call TON32       ;G
call DEL2        ;(Zpoždění mezi jednotlivými tóny)
mov R1,#255      ;(Nastavení delšího tónu)
call TON34       ;A
call DEL2        ;(Zpoždění mezi jednotlivými tóny)
mov R1,#100      ;(Nastavení délky běžného tónu)
call TON32       ;G
call DEL2        ;(Zpoždění mezi jednotlivými tóny)
```

1.2.2. VÝPOČTY PRO NASTAVENÍ ČASOVAČE

Výpočty nutné pro přesné nastavení časovačů nejsou velmi složité, ale vždy vyžadují jemné doladění vzniklého času. Každý časovač má jinou délku maximálního času, který je schopný vytvořit na jedno celé naplnění a proto se snažíme volit časovače vhodně podle potřebné délky času. Přesné délky časovačů podle módu jsem již uvedl, z tohoto důvodu je zde nebudu vypisovat znovu. Veškeré výpočty je nutné provádět v μs .

Jako příklad zde uvedu tvorbu času 50ms a potom čas větší, který vyžaduje opakované plnění časovače. Pro tento čas je vhodné zvolit 16b mód, protože můžeme vytvořit čas až 65,536ms. Po zvolení módu začneme s výpočty pro nastavení časovače.

Zde uvedu příklad s průběžným popisem pro tvorbu času 50ms.

$$65536 - 50000 = 15536$$

První krok: Odečíst od maximální hodnoty časovače náš požadovaný čas.

$$15536 = 3CB0H$$

Druhý krok: Zbytek převést na hexadecimální číslo. Vzniklé hexa číslo si rozdělíme na horní a dolní polovinu. Horní polovina je tedy 3C a dolní je B0. Po rozdělení vytvoříme časovač a tyto hodnoty dosadíme do registrů TH0 a TL0. V mém případě využívám časovač 0. Pokud by se jednalo o časovač 1, plnili bychom TL1 a TH1. Vznikne nám tento podprogram časovače.

```
S2:  mov tmod,#01h      ; (Zvolení 16b módu)
      mov TH0,#3Ch      ; (Nastavení horních 8b časovače)
      mov TL0,#0B0h     ; (Nastavení dolních 8b časovače)

      setb TR0
      clr TF0
CYC:  jnb TF0,CYC
      clr TR0
      ; ret
```

Třetí krok: Po odsimulování časovače zjistíme, že vzniklý čas je přesně 50011 μs . Chyba 11 μs se může zdát být zanedbatelnou, ale není. Odstraníme ji tak, že její velikost přičteme ke zbytku po odčítání u prvního kroku. Jelikož jsme simulovali program, když byla instrukce ret zablokována, nevidíme čas, který tato instrukce přidává a to jsou konkrétně 2 μs . Pokud tedy chceme přesný čas, musíme ke zbytku přičíst ne 11 ale 13.

$$15536+13=15549$$

Čtvrtý krok: Opět převedeme na hexadecimální vyjádření.

$$15549=3CBD$$

Pátý krok: Vzniklý čas je 49997 μ s bez instrukce ret. Tedy nám vznikla další chyba a to je 1 μ s. Odstraníme ji skoro stejnou cestou, akorát tentokrát odečteme od zbytku 1. Čas 50ms tedy máme hotový.

Nyní uvedu příklad s průběžným popisem pro vytvoření času delšího než je 65536 μ s. Vybral jsem čas 250ms, který pro tento příklad bude dostačující.

$$250000:65536=3,8147$$

První krok: V případě většího času je potřeba vydělit náš požadovaný čas časem maximálním tedy 65536 μ s.

$$4*65536=262144$$

Druhý krok: Vzniklé číslo vždy zaokrouhlíme nahoru tedy na 4 v našem případě. Vzniklým číslem vynásobíme maximální možný čas na jedno naplnění.

$$262144-250000=12144$$

Třetí krok: Od vzniklého čísla odečteme námi požadovaný čas.

$$12144=2F70H$$

Čtvrtý krok: Zbytek převedeme na hexa vyjádření. Číslo opět rozdělíme na dvě poloviny a použijeme je k nastavení časovače. Vzniklý časovač bude téměř shodný s podprogramem času 50ms, pouze se bude 4 krát nulovat. Počet opakování jsme vlastně vypočítali hned v prvním kroku. Vznikne nám tento podprogram časovače.

```

S2:  mov tmod,#01h      ; (Nastavení 16b módu)
      mov TH0,#2Fh      ; (Nastavení horních 8b časovače)
      mov TL0,#70h      ; (Nastavení dolních 8b časovače)
      mov R1,#4         ; (Nastavení opakování)

      setb TR0          ; (Zapnutí časovače 0)
N1:  clr TF0            ; (Vynulování bitu přetečení)
CYC: jnb TF0,CYC        ; (Čekání na přetečení)
      djnz R1,N1        ; (Kontrola opakování)
      clr TR0          ; (Vypnutí časovače 0)
;    ret               ; (Konec podprogramu)

```

Vzniklý čas opět nebude přesný a je nutné ho odladit stejným způsobem jako v případě podprogramu na 50ms. Tento postup je stejný i u všech ostatních módů časovače. Ostatní módy se liší pouze nastavením registru tmod a také hodnotou maximálního času časovače.

1.2.3. TABULKA VYPOČÍTANÝCH ČASŮ

Jedná se o tabulku všech tónů s vypočítanými a odladěnými hodnotami pro časovače. Tabulka obsahuje celý postup výpočtu. Všechny časy jsou realizované pomocí 16b časovače 0. Rozhodl jsem se využít 16b mód z důvodu, že vzniklá chyba je téměř neustále stejná a proto vznikaly v závěru chyby v řádech jednotek μ s. Z důvodu malé chyby pak bylo i následné odladění jednoduché a dalo se počítat bez pomoci kalkulačky.

Tabulka 1-2: Hodnoty pro časovače

16b časovač

Číslo tónu	Tón	Frekvence (Hz)	T/2 [us]	Th+Tl [DEC]	Th+Tl [HEX]
1	C	130,81	3 822,3	61 726	F11E
2	C#/Db	138,59	3 607,8	61 940	F1F4
3	D	146,83	3 405,3	62 143	F2BF
4	D#/Eb	155,56	3 214,2	62 334	F37E
5	E	164,81	3 033,8	62 514	F432
6	F	174,61	2 863,5	62 684	F4DC
7	F#/Gb	185	2 702,7	62 845	F57D
8	G	196	2 551,0	62 997	F615
9	G#/Ab	207,65	2 407,9	63 140	F6A4
10	A	220	2 272,7	63 275	F72B
11	A#/Bb	233,08	2 145,2	63 403	F7AB
12	B	246,94	2 024,8	63 523	F823
13	C1	261,63	1 911,1	63 637	F895
14	C#1/Db1	277,18	1 803,9	63 744	F900

Číslo tónu	Tón	Frekvence (Hz)	T/2 [us]	Th+TI [DEC]	Th+TI [HEX]
15	D1	293,66	1 702,6	63 845	F965
16	D#1/Eb1	311,13	1 607,0	63 941	F9C5
17	E1	329,63	1 516,9	64 031	FA1F
18	F1	349,23	1 431,7	64 116	FA74
19	F#1/Gb1	369,99	1 351,4	64 197	FAC5
20	G1	392	1 275,5	64 272	FB10
21	G#1/Ab1	415,3	1 203,9	64 344	FB58
22	A1	440	1 136,4	64 412	FB9C
23	A#1/Bb1	466,16	1 072,6	64 475	FBDB
24	B1	493,88	1 012,4	64 536	FC18
25	C2	523,25	955,6	64 592	FC50
26	C#2/Db2	554,37	901,9	64 646	FC86
27	D2	587,33	851,3	64 697	FCB9
28	D#2/Eb2	622,25	803,5	64 744	FCE8
29	E2	659,25	758,4	64 790	FD16
30	F2	698,46	715,9	64 832	FD40
31	F#2/Gb2	739,99	675,7	64 872	FD68
32	G2	783,99	637,8	64 910	FD8E
33	G#2/Ab2	830,61	602,0	64 946	FDB2
34	A2	880	568,2	64 980	FDD4
35	A#2/Bb2	932,33	536,3	65 012	FDF4
36	B2	987,77	506,2	65 042	FE12
37	C3	1046,5	477,8	65 070	FE2E
38	C#3/Db3	1108,73	451,0	65 097	FE49
39	D3	1174,66	425,7	65 122	FE62
40	D#3/Eb3	1244,51	401,8	65 146	FE7A
41	E3	1318,51	379,2	65 169	FE91
42	F3	1396,91	357,9	65 190	FEA6
43	F#3/Gb3	1479,98	337,8	65 210	FEBA
44	G3	1567,98	318,9	65 229	FECD
45	G#3/Ab3	1661,22	301,0	65 247	FEDF
46	A3	1760	284,1	65 264	FEF0
47	A#3/Bb3	1864,66	268,1	65 280	FF00
48	B3	1975,53	253,1	65 295	FF0F
49	C4	2093	238,9	65 309	FF1D

Všechny hodnoty uvedené v tabulce jsou využité při tvorbě tónů. Hodnoty ve sloupci s hexa hodnotami časovačů jsou odladěné na 1 μ s. Postup použitý při výpočtu byl stejný jako postup, který jsem uvedl v předchozí kapitole.

1.2.4. ZTLUMENÍ TÓNU

Jedna z věcí, které jsem chtěl dosáhnout u svého klavíru bylo to, aby se alespoň do určité míry choval jako skutečný klavír. Jakmile na obyčejném strunovém klavíru stisknete klávesu, udeří kladívko do struny a ta vydá zvuk. Zvuk vyprodukovaný takovýmto klavírem ale netrvá věčně. Tón skutečného klavíru pomalu utichne, i když držíte klávesu neustále stlačenou. Samozřejmě, že na elektronickém klavíru nejsou struny a tón tak může hrát nepřetržitě po celou dobu stačení klávesy. Dal jsem si tedy za úkol, najít řešení tohoto problému.

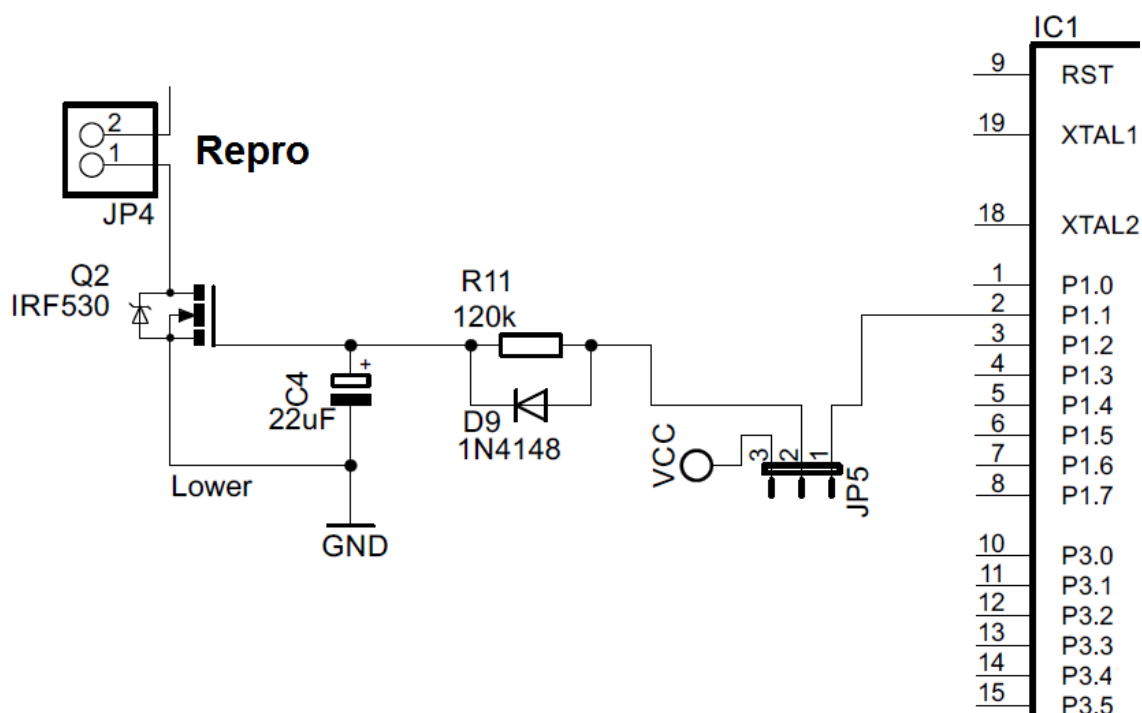
Vytvořené řešení se nakonec ukázalo ještě překvapivější, než jsem očekával. Použil jsem pouze pár součástek a menší úpravu programu. Řešení je takové, že po kontrole celé klávesnice, což trvá zhruba 135 μ s se nastaví pin 1 na portu P1 na logickou 1. Na tento pin je připojeno mé řešení nekonečného tónu. Po celou dobu neaktivity kláves je na P1.1 logická 1, ale ve chvíli kdy se klávesa stiskne se na P1.1 nastaví logická 0, která je signálem pro aktivaci ztlumení. Instrukce na nastavení logické 0 na P1.1, je ukrytá v každém časovači u každého tónu (*).

Všechny časovače tedy vypadají takto.

```
C15:      mov tmod,#01h
          mov TH0,#0FEh
          mov TL0,#62h
          clr P1.1          ; (*)
          setb TR0
          clr TF0
SE15:     jnb TF0,SE15
          clr TR0

          ret
```

Obrázek 1-2: Část schématu s řešením tlumení tónu



Ve schématu je vynechána část zesilovače, která vede z P1.0 a napojuje se na druhý výstup do reproduktoru. Jakmile je na pinu P1.1 logická 1, drží se kondenzátor C4 nabitý a tranzistor Q2 otevřený. V momentě vypnutí pinu P1.1 se začne kondenzátor vybíjet a tranzistor Q2 se začne pomalu uzavírat. Toto chování způsobí, že se tón začne pomalu zeslabovat a chová se tak, jako na skutečném klavíru. Propojka JP5 je vyvedena do přepínače, který je přístupný pro obsluhu klavíru. Slouží vlastně k vypnutí a zapnutí funkce ztlumování tónu.

Tato funkce musí být při přehrávání naprogramovaných skladeb vypnuta. Kondenzátor se nestíhá nabíjet a dochází tak k tomu, že se skladba ztlumí a není tak slyšet.

2. KLÁVESNICE

Princip funkce klávesnice je takový, že procesor kontroluje zvolené bity řádků a čeká, než se na nich objeví logická 0. Klávesnice klavíru vypadá jako řada tlačítek vedle sebe, ale ve skutečnosti se chová jako 7 sloupců po 8 řádcích, které leží vedle sebe. Zmínil jsem pouze 7 sloupců z důvodu, že sloupec číslo 8 je použitý pro předprogramované funkce včetně přehrávání skladeb. V jeden moment je pouze jeden aktivní sloupec, který má na svém pinu logickou 0.

Když je například na sloupci 5 logická 0, je na všech ostatních sloupcích logická 1. Tímto způsobem tedy není cesta, jak by se mohla stisknutá klávesa vyhodnotit špatně. Kontrola kláves funguje tak, že dokud je na pinech jednotlivých řádků logická 1, tak se program kontroly kláves opakuje stále dokola. Jakmile se stiskne klávesa, dojde ke zkratu, protože na řádky je přivedena nepřetržitě logická 1 a na aktuálně aktivní sloupec byla přivedena logická 0. Jediné co tedy program mění, je aktivní sloupec, řádky se nikterak nemění.

Pro příklad zde uvedu program kontroly jednoho sloupce klávesnice.

```
SL1: mov P2,#11111110b      ; (Aktivace sloupce 1)

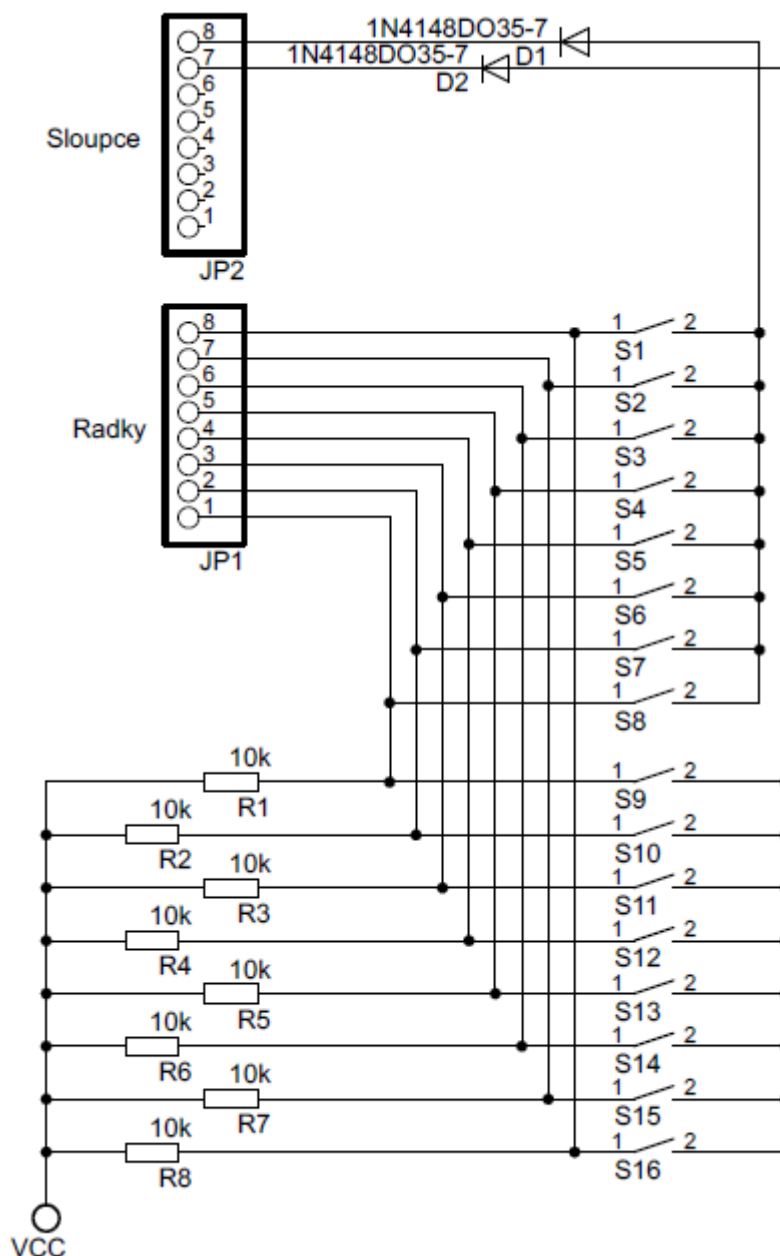
      jb P3.0,K1             ; (Kontrola stavu bitu 0)
      call TON1              ; Ton C
      jmp SL1                ; (Skočí zpět na sloupec 1)
K1:   jb P3.1,K2
      call TON2              ; Ton Db
      jmp SL1
K2:   jb P3.2,K3
      call TON3              ; Ton D
      jmp SL1
K3:   jb P3.3,K4
      call TON4              ; Ton Eb
      jmp SL1
K4:   jb P3.4,K5
      call TON5              ; Ton E
      jmp SL1
K5:   jb P3.5,K6
      call TON6              ; Ton F
      jmp SL1
K6:   jb P3.6,K7
      call TON7              ; Ton Gb
      jmp SL1
K7:   jb P3.7,SL2
      call TON8              ; Ton G
      jmp SL1
```

Pokud je na kontrolovaném pinu logická 1, program skočí na další klávesu, která provede stejnou kontrolu. Jakmile program dorazí k poslednímu pinu z daného sloupce a je na něm logická 1, program skočí na další sloupec.

2.1. ZAPOJENÍ KLÁVESNICE

Pro příklad je zde vloženo schéma prvních 16 kláves, tedy 2 sloupce použité klávesnice. Kompletní klávesnice čítá 49 kláves a zabrala by příliš velké množství místa. Na každý sloupec je připojená dioda, která je nasměrovaná katodou k procesoru. Diody zajistí, že na klávesy dojde pouze logická 0.

Obrázek 2-1: Schéma části klávesnice



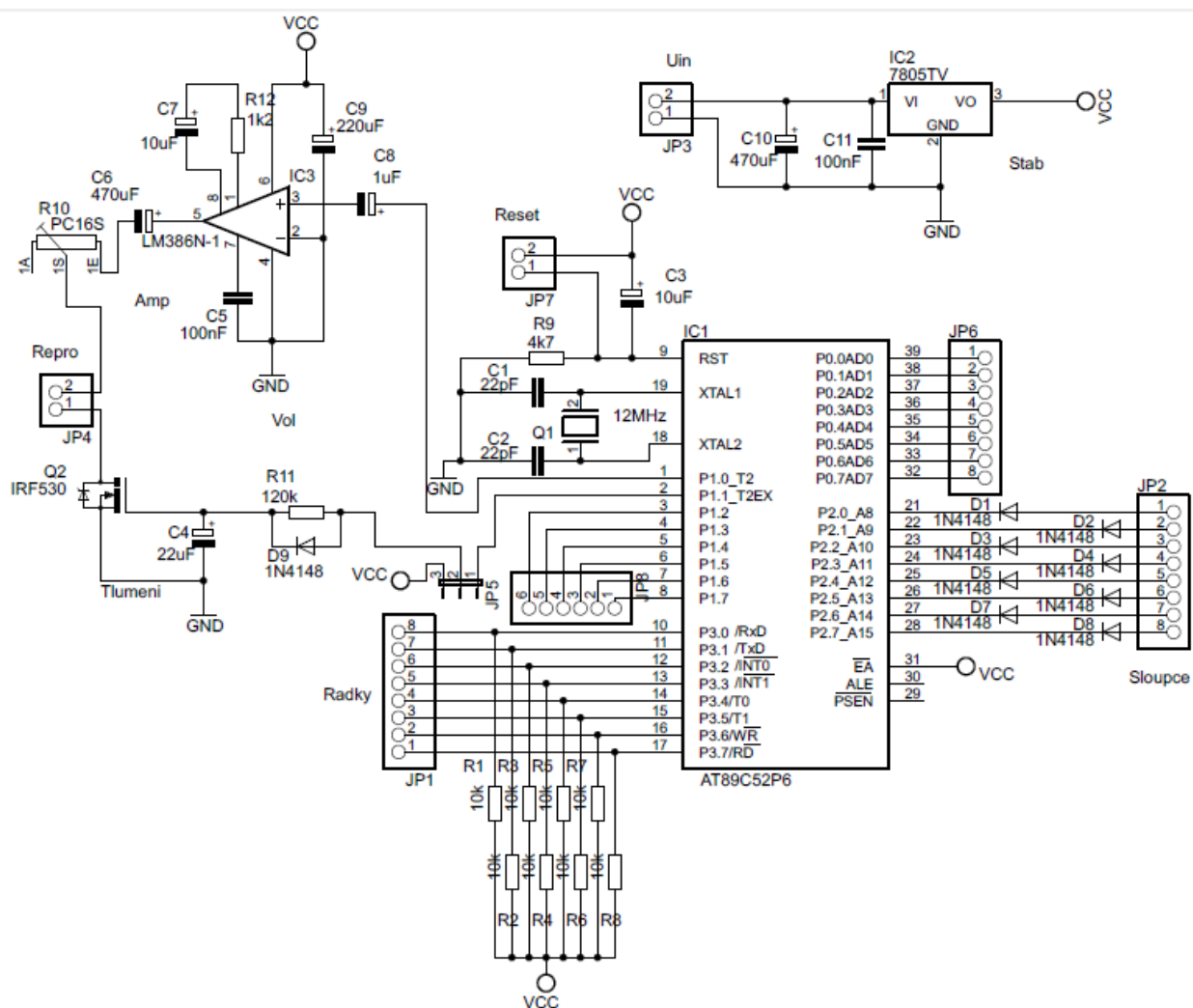
3. VÝROBNÍ DOKUMENTACE

Aby byla dokumentace přehlednější, rozhodl jsem se vytvořit kapitolu zaměřenou na technickou stránku projektu. V této kapitole proberu výrobek ze strany schématu a navrzení desky plošných spojů. Vysvětlím zde důvody, proč jsem desku navrhl a realizoval tak jak je.

3.1. SCHÉMA VÝROBKU

Schéma zobrazuje klavír jako celek až na klávesnici, která je příliš rozměrná. Veškeré prvky výrobku se připojují k řídicí desce pomocí konektorů. Každý konektor ve schématu je popsán, aby bylo viditelně poznat, k čemu slouží. Schéma klávesnice je pouze orientační aby bylo vidět jak je realizovaná.

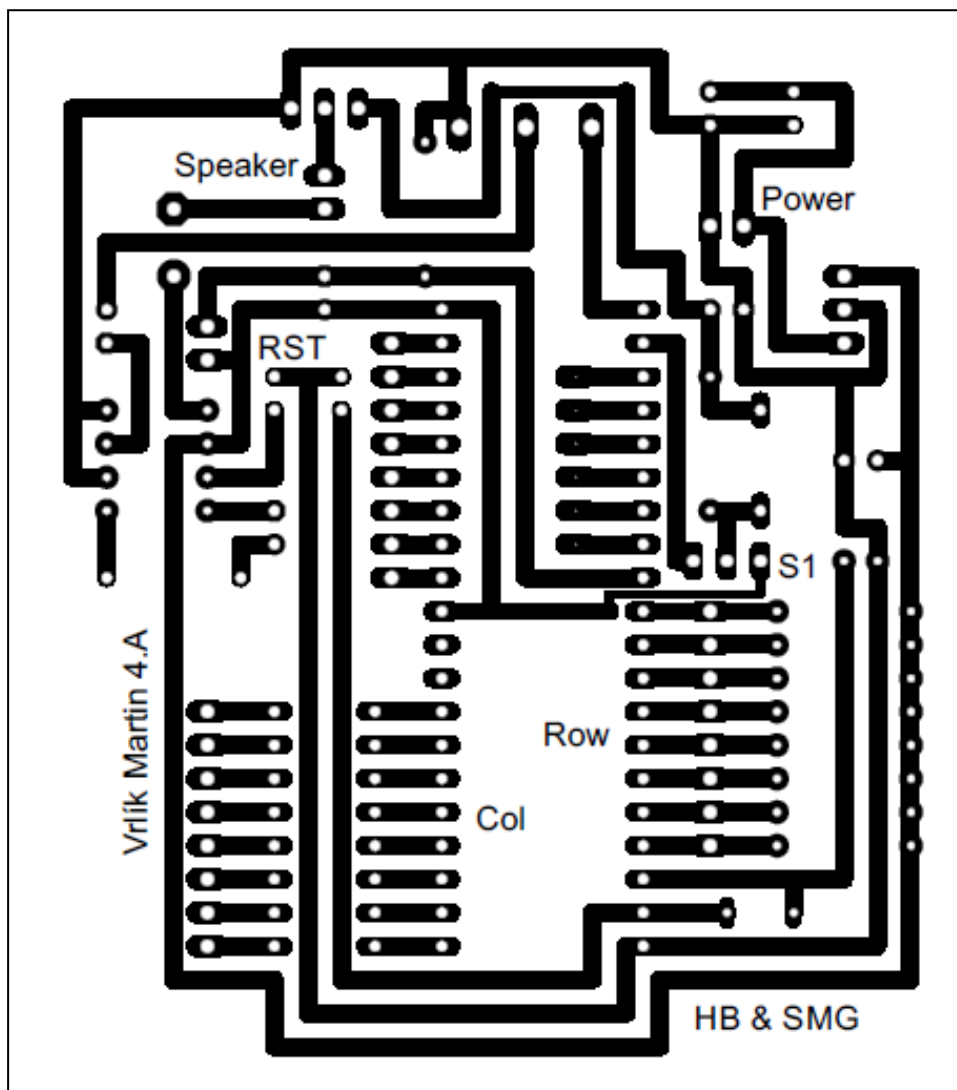
Obrázek 3-1: Schéma základní desky

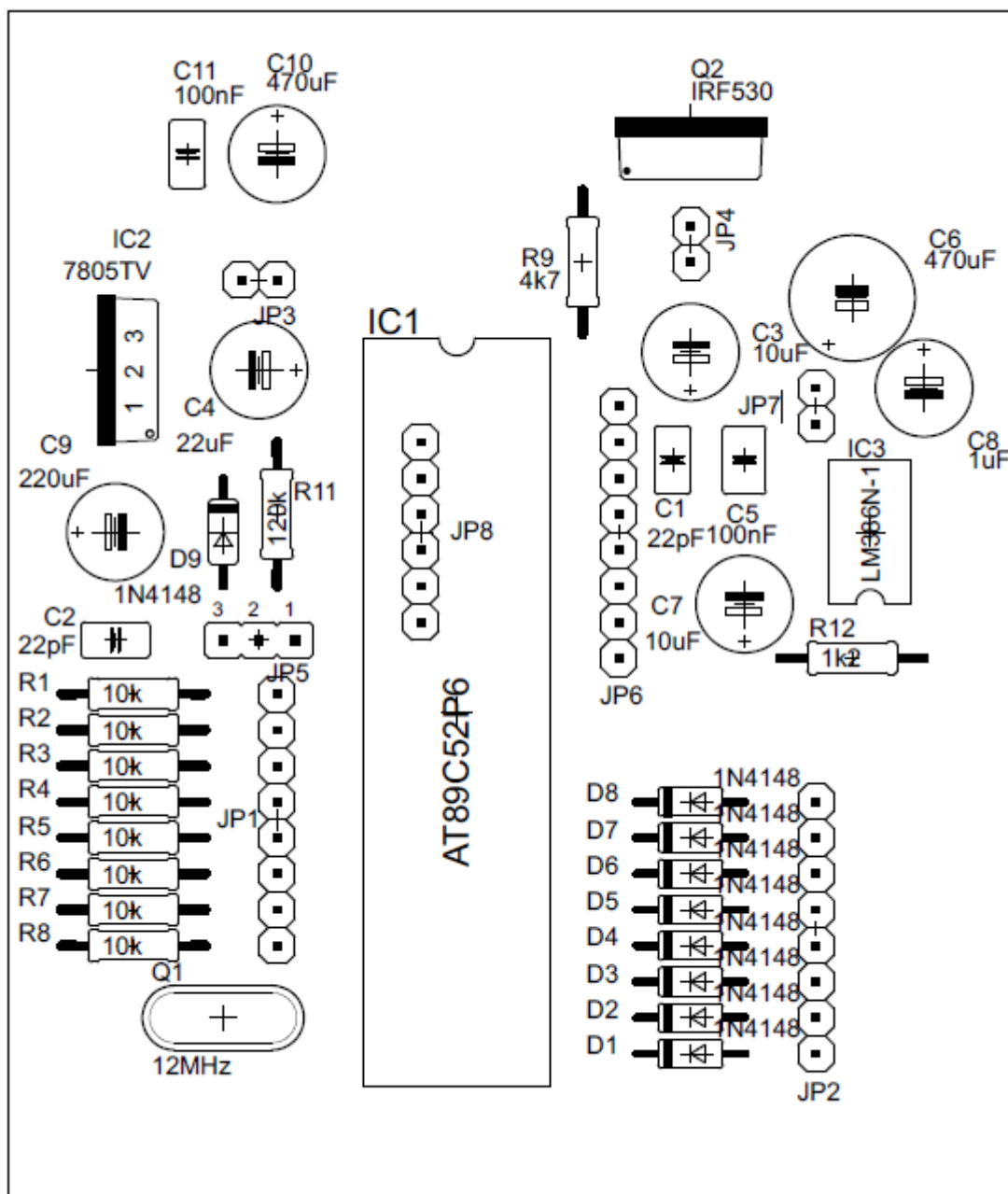


3.2. NÁVRH DPS

Už na schématu je viditelně poznat řešení připojování jednotlivých součástí výrobku. Stejně jako u schématu, jsem se rozhodl i na desce jednotlivé konektory popsat. Jelikož šasi, do kterého jsem chtěl desku umístit, mělo poměrně omezené rozměry, musel jsem desku navrhnout a realizovat s co nejmenšími rozměry.

Obrázek 3-2: Pohled ze strany spojů. (zvětšené zobrazení) 71mm x 84mm





Lišta jumper pinů pod označením JP8 není na desce ve skutečnosti osazená a je ve schématu zobrazena pouze pro účel budoucího využití zbylých pinů portu P1. Celý port P0 je pouze vyveden do lišty s označením JP6 a je připraven na již zmíněné možné využití. Využití portu P0 vidím v použití sedmissegmentu na zobrazení čísla právě přehrávané skladby. Využití všech portů je uvedeno na začátku v první kapitole.

3.3. SEZNAM SOUČÁSTEK

Seznam součástek použitých při výrobě základní desky klavíru. Zbylé součástky jsou uvedené pod čarou.

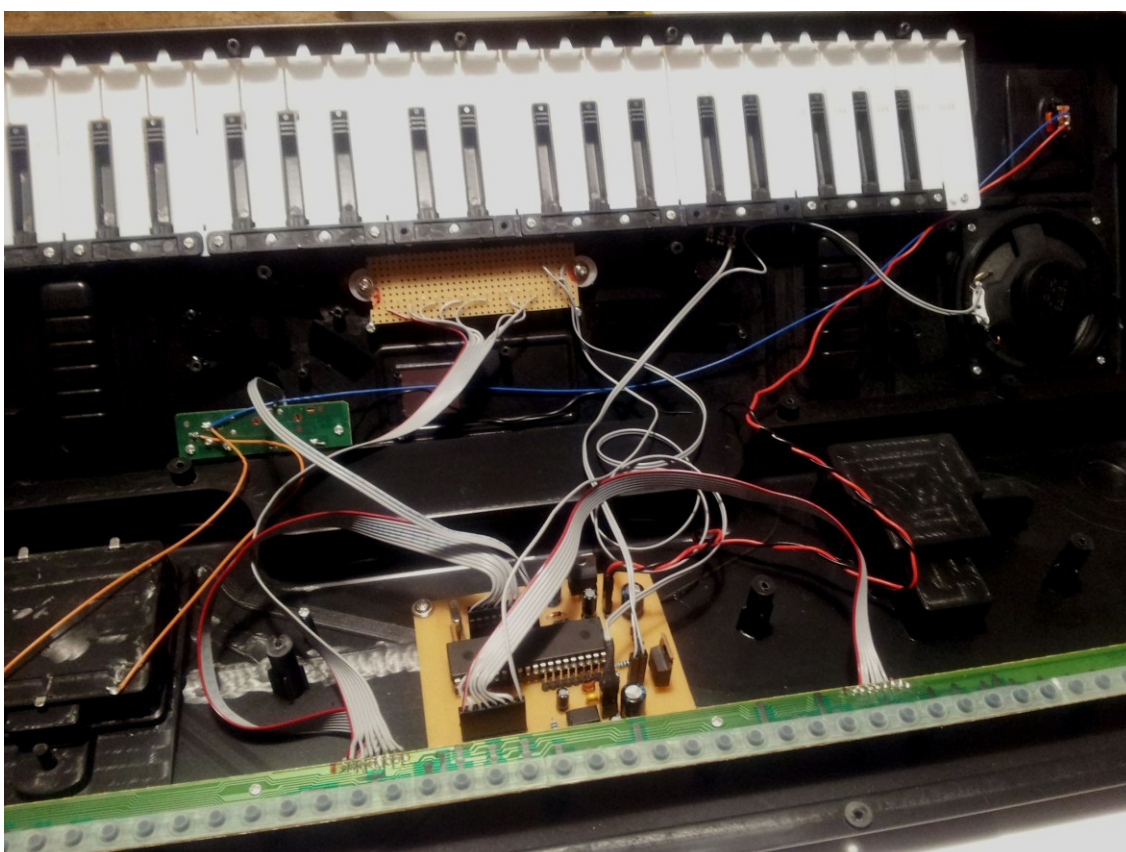
Značka ve schématu	Typ	Hodnota	Poznámka
R1,R2,R3,R4,R5,R6, R7,R8	Rezistor	10k Ω	
R9	Rezistor	4k7	
R10	Potenciometr	5k Ω	Plastová hřídel
R11	Rezistor	120k Ω	
R12	Rezistor	1k2	
C1, C2	Kondenzátor	22pF	
C3, C7	Kondenzátor	10 μ F	
C4	Kondenzátor	22 μ F	
C5, C11	Kondenzátor	100nF	
C6, C10	Kondenzátor	470 μ F	
C8	Kondenzátor	1 μ F	
C9	Kondenzátor	220 μ F	
D1,D2,D3,D4,D5,D6, D7,D8,D9	Dioda	1N4148	
IC1	Procesor Atmel		AT89S52
IC2	Stabilizátor		7805
IC3	OZ		LM386
Q1	Krystal	12MHz	
Q2	Tranzistor	IRF530	
JP1-JP8	Jumper lišta		Různé rozměry
Tlačítka	TM043		
Konektory	SPK-8,SPK-3,SPK-2		

3.4.FOTODOKUMENTACE

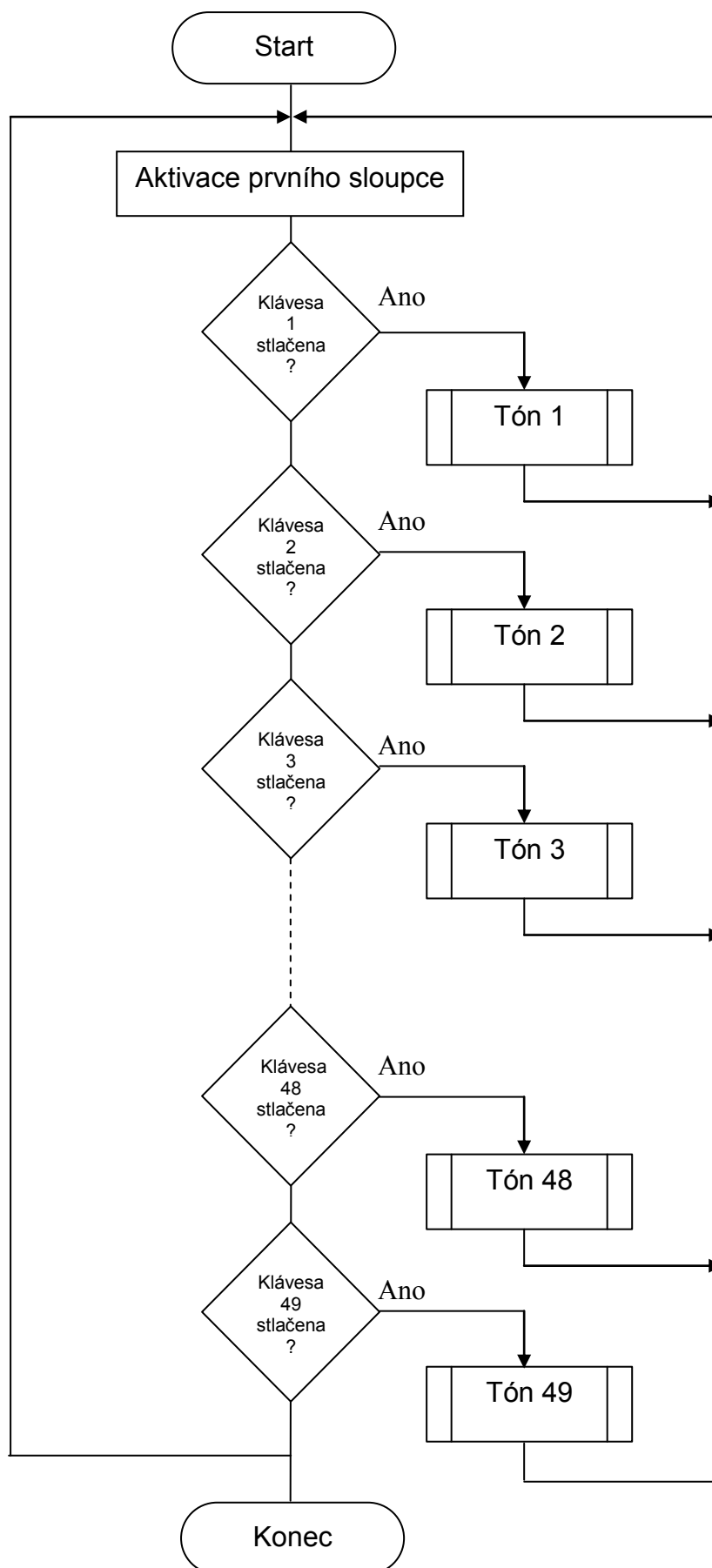
Pohled na celý výrobek z vrchní strany.



Pohled na vnitřní uspořádání klavíru.



3.5. VÝVOJOVÝ DIAGRAM



4. NÁVOD K POUŽITÍ

Pro zapnutí výrobku přepněte oranžový přepínač na pravé straně klavíru do polohy ON.

Uprostřed klavíru je ovládací panel. Panel obsahuje ovládací prvky pro dodatečné funkce. Zde je umístěno 8 ovládacích tlačítek označených čísly 1-8.

1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---

Každé tlačítko má vlastní funkci. Tlačítko 1 slouží pro test kláves. Stisknutí tohoto tlačítka přehraje všechny tóny za sebou. Tlačítka 2-5 obsahují naprogramované skladby. Poslední tlačítko číslo 8 je tlačítko STOP, které přeruší právě přehrávanou skladbu nebo test kláves.

- | |
|--------------------|
| 1. Test kláves |
| 2. Holka modrooká |
| 3. Pro Elišku |
| 4. Ovčáci čtveráci |
| 5. Pec nám spadla |
| 6. --- |
| 7. --- |
| 8. STOP |

Na pravé straně je umístěn ovladač hlasitosti, kterým můžete nastavit hlasitost celého klavíru.



+ -

Poslední řídicí prvek je v levé horní části ovládacího panelu. Je zde umístěný konektor napájení z externího zdroje a přepínač pro aktivaci tlumení tónu. Pod tímto přepínačem je umístěn popisek, zdali je přepínač v poloze zapnuto nebo vypnuto.

Externí napájení	Tlumení tónu
---------------------	-----------------

Zapnuto / Vypnuto

ZÁVĚR

Dle zadání jsem sestrojil funkční verzi elektronického klavíru. Klavír zahrnuje 49 kláves včetně půltónů, tedy celé 4 oktávy a ještě 8 tlačítek s dodatečnými funkcemi jako například hraní naprogramovaných skladeb a tlačítka pro přerušení aktuální činnosti. Celý výrobek byl osazen do šasi staršího nefunkčního elektronického klavíru. Možnost využít tento klavír mi umožnila, aby byl výrobek více realistický.

Funkce, kterou klavír obsahuje a která není obsažena v zadání, je chování stisknuté klávesy. Stlačíte-li klávesu libovolného tónu, bude se tón pomalu ztlumovat a simulovat tak chování struny v reálném klavíru. Tuto funkci jsem realizoval za pomoci tranzistoru MOSFET, kondenzátoru a rezistoru. Princip funkce je popsán v kapitole 1.2.4.

Další rozšíření klavíru by bylo připojení sedmisegmentového displeje, který by zobrazoval číslo aktuálně přehrávané skladby. Případně by byl připojený dvouřádkový alfanumerický displej, který by ukazoval přímo název skladby nebo přímo její text.

Celkové náklady na výrobu klavíru se pohybují okolo 400Kč, protože jsem měl k dispozici šasi, do kterého jsem klavír umístil a neměl jsem tak problémy s řešením do čeho klavír usadit.

SEZNAM POUŽITÉ LITERATURY A STUDIJNÍCH MATERIÁLŮ

- 1) Ing. Zuzana Veselá: Programování 8051 (učební texty praxe), 2010, Olomouc.
- 2) <http://www.atmel.com> – Datasheet použitého procesoru.
- 3) <http://www.alldatasheet.com> – Datasheet pro zesilovač.
- 4) <http://cs.wikipedia.org/> – Wikipedie, Klavíry obecně

SEZNAM OBRÁZKŮ, GRAFŮ A TABULEK

V této části dokumentace jsou shrnuty všechny obrázky a příklady, které se vyskytují v průběhu celého dokumentu.

Obrázky:

Obrázek 1-1: Rozložení pinů v pouzdře

Obrázek 1-2: Část schématu s řešením tlumení tónu

Obrázek 2-1: Schéma části klávesnice

Obrázek 3-1: Schéma základní desky

Obrázek 3-2: Pohled ze strany spojů. (zvětšené zobrazení)

Obrázek 3-3: Osazovací plán (zvětšené zobrazení)

Tabulky:

Tabulka 1-1: Kombinace nastavení bitů M1 a M0

Tabulka 1-2: Hodnoty pro časovače