



M Ű E G Y E T E M 1 7 8 2

**Budapesti Műszaki és Gazdaságtudományi Egyetem**  
Villamosmérnöki és Informatikai Kar  
Hálózati Rendszerek és Szolgáltatások Tanszék

Gergely Dániel

**DIGITÁLIS VEZÉRLÉSŰ ANALÓG  
DELAY EFFEKT  
MEGVALÓSÍTÁSA**

KONZULENS

**Firtha Gergely**

BUDAPEST, 2016

# Tartalomjegyzék

<b>Ábrajegyzék</b> .....	<b>6</b>
<b>Összefoglaló</b> .....	<b>9</b>
<b>Abstract</b> .....	<b>10</b>
<b>1 Bevezetés</b> .....	<b>11</b>
<b>2 BBD IC-k felépítése és működése</b> .....	<b>13</b>
2.1 Történeti áttekintés .....	13
2.2 Felépítés, Működés .....	13
2.2.1 Alapvető felépítés .....	14
2.2.2 MOS tetródák alkalmazása .....	14
2.2.3 Komplementer kimenetek .....	15
2.2.4 Párhuzamos elrendezés .....	16
2.2.5 Több kimenet: multitap BBD .....	16
<b>3 BBD alapú áramkörök általános felépítése</b> .....	<b>18</b>
3.1 BBD-vel megvalósítható zenei effektek .....	18
3.2 Órajel generátor .....	19
3.3 Szűrők .....	19
3.4 Kompander.....	21
3.5 Visszacsatolás .....	23
<b>4 A megépített áramkör</b> .....	<b>24</b>
4.1 Táp .....	25
4.2 Analóg jelút.....	27
4.2.1 Kompresszor .....	27
4.2.2 Átlapolódásgátló és rekonstrukciós szűrők.....	28
4.2.3 Expander .....	30
4.2.4 Bypass kapcsoló.....	31
4.3 Digitális vezérlés.....	31
4.3.1 A mikrokontroller .....	31
4.3.2 Potméterek .....	32
4.3.3 BBD órajelek előállítása .....	33
4.4 Az áramkör kalibrálása .....	34
4.4.1 A kalibrálás célja .....	34

4.4.2 Felhasznált eszközök .....	35
4.4.3 A kalibrálás folyamata .....	35
<b>5 Szoftver .....</b>	<b>38</b>
5.1 Szoftverfejlesztő környezet.....	38
5.2 Implementáció .....	38
5.3 Inicializálás .....	38
5.3.1 Analóg bemenetek beolvasása .....	39
5.3.2 Órajel generálás .....	40
5.3.3 Digitális potméterek – SPI kommunikáció .....	41
<b>6 Huzalozási rajz elkészítése .....</b>	<b>42</b>
<b>7 Mérések, értékelés.....</b>	<b>43</b>
7.1 Mérési elrendezés .....	43
7.2 Eredmények .....	43
<b>8 Összefoglalás.....</b>	<b>49</b>
<b>9 Irodalomjegyzék.....</b>	<b>50</b>
<b>Függelék.....</b>	<b>52</b>

# Ábrajegyzék

2-1. ábra Legegyszerűbb BBD felépítés [2] .....	14
2-2. ábra MOS tetródák alkalmazása [2] .....	15
2-3. ábra Komplementer kimenetek működése [5].....	15
2-4. ábra Komplementer kimenetekkel rendelkező elrendezés [2].....	16
2-5. ábra Több kimenettel rendelkező (multitap) BBD [2] .....	16
3-1. ábra A BBD alapú áramkörök általános felépítése.....	18
3-2. ábra Késleltetésen alapuló effektek jellemző paraméterei .....	18
3-3. ábra Különböző BBD IC-k.....	19
3-4. ábra Harmadfokú Sallen-Key szűrő .....	21
3-5. ábra A kompander hatása [8].....	22
3-6. ábra A kompander egy csatornájának blokkvázlata .....	22
4-1. ábra A megépített áramkör blokkvázlata.....	24
4-2. ábra Tápáramkör (1).....	26
4-3. ábra Tápáramkör (2).....	27
4-4. ábra A kompresszor blokkvázlata .....	28
4-5. ábra Harmadfokú Sallen-Key átlapolódásgátló szűrő .....	29
4-6. ábra Harmad- ill. másodfokú Sallen-Key rekonstrukciós szűrő.....	29
4-7. ábra Átlapolódásgátló és rekonstrukciós szűrők frekvenciamenete és törésponti frekvenciái .....	30
4-8. ábra Az expander blokkvázlata .....	31
4-9. ábra A digitális potméterek és az ISP csatlakozó.....	33
4-10. ábra Az órajel előállítása .....	34
4-11. ábra Kimenet a munkapont beállítás előtt és után .....	36
4-12. ábra Kimenet a CANCEL trimmer beállítása előtt.....	37
4-13. ábra Kimenet a CANCEL trimmer beállítása után.....	37
5-1. ábra Az inicializáció folyamatábrája .....	39
5-2. ábra A főciklus folyamatábrája .....	40
7-1. ábra Koherencia függvény ( $t_5$ - $f_4$ - $m_{10}$ ).....	43
7-2. ábra Az effekt impulzusválasza ( $t_0$ - $f_5$ - $m_{10}$ ).....	44
7-3. ábra Az effekt impulzusválasza ( $t_{10}$ - $f_5$ - $m_{10}$ ).....	44
7-4. ábra Az effekt impulzusválasza ( $t_5$ - $f_5$ - $m_{10}$ ).....	45

7-5. ábra Eredeti és késleltetett impulzus (t10-f5-m10) .....	45
7-6. ábra Az effekt frekvenciamenete (t5-f5-m10).....	46
7-7. ábra Az effekt frekvenciamenete (t0-f5-m10).....	46
7-8. ábra A BBD hurok frekvenciamenete visszacsatolás nélkül(t5-f0-m10) .....	47
7-9. ábra A visszacsatolt BBD hurok frekvenciamenete (t5-f4-m10) .....	48
9-1. ábra Az analóg effekt kapcsolási rajza .....	52
9-2. ábra A digitális vezérlés kapcsolási rajza.....	53
9-3. ábra Huzalozási rajz .....	61
9-4. ábra A huzalozási rajz felső rétege .....	61
9-5. ábra A huzalozási rajz alsó rétege .....	62

# HALLGATÓI NYILATKOZAT

Alulírott **Gergely Dániel**, szigorló hallgató kijelentem, hogy ezt a szakdolgozatot meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy hitelesített felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Kelt: Budapest, 2016. 12. 08.

.....  
Gergely Dániel

# Összefoglaló

Szakedolgozatomban áttekintem a delay zenei effekt, azaz audio jel változtatható idejű késleltetésének megvalósítására szolgáló különböző megoldásokat, ezek előnyeit és hátrányait. Ezt követően ismertetem az analóg áramköri megvalósításra lehetőséget nyújtó Bucket Brigade Device (BBD) IC-k felépítését és működését, továbbá a BBD chipeken alapuló effektek (delay, chorus, flanger, vibrato) általános felépítését. Ezt követi az általam megépített áramkör bemutatása. Itt kitérek a szükséges tápfeszültségek előállítására, és az effekt paramétereinek mikrokontrollerrel történő vezérlésére. A vezérlés két fontos eleme a paramétereket szabályzó hagyományos potméterek helyettesítése és a BBD számára szükséges órajel előállítása. Emellett ismertetem a vezérléshez nélkülözhetetlen szoftver működését, az áramkör kalibrálásának folyamatát és az áramkörhöz elkészített huzalozási rajzot, az itt figyelembe vett szempontokat. Az elkészült áramkört méréseket végeztem, melyek igazolják, hogy az effekt az elvárásoknak megfelelően működik, illetve megfigyelhetővé teszik a BBD-vel megvalósított késleltető jellegzetes átviteli tulajdonságait. Végül írok a további fejlesztéssel kapcsolatos célokról és lehetőségekről, illetve a megvalósított mikrokontroller alapú vezérlőáramkör felhasználási lehetőségeiről.

## **Abstract**

This thesis presents various methods for producing a time-delayed version of an audio signal, which is a popular musical effect across many genres. Such effect can be achieved with the use of analog circuitry containing Bucket Brigade Device (BBD) integrated circuits. I discuss the history, different topologies and uses of these devices, along with their important characteristics and limitations, including aliasing, distortion and noise. To reduce these artifacts BBDs are often accompanied by low-pass filters and compander circuitry. This produces unique sonic characteristics that are preferred by many musicians, but these analog devices unfortunately lack many useful functions offered by digital systems. During my work I used a schematic based on the Boss DM-2 guitar pedal, to realize an analog delay effect with three adjustable parameters: Time, Feedback and Mix. In my thesis I present a microcontroller based digital circuit and software for controlling the effect's parameters along with the advantages and possibilities of using such digital control. I also discuss the design considerations for producing a printed circuit board and present a possible board layout designed with the Eagle CAD software. Based on measurements, I verified that the effect is functioning as expected and also discuss some of the unique characteristics of BBD delay effects,



# 1 Bevezetés

Számos könnyűzenei műfajban használatos az ún. delay effekt, azaz audio jel változtatható idejű, visszacsatolt késleltetése. Ez megvalósítható a hang mágnesszalagra történő rögzítésével és visszajátszásával, analóg áramkörü elemekkel vagy digitális jelfeldolgozás alkalmazásával. A szalagos delayt jellegzetes hangzása miatt sokan kedvelik, azonban a nagy méret, a mozgó alkatrészek, illetve a szalag elhasználódása miatt ez a megoldás nem túl elterjedt. Az analóg áramkörrel történő megvalósításra az ún. Bucket Brigade Device (BBD) chip nyújt lehetőséget. Itt az audio jel feszültségével arányos töltésmennyiségeket kondenzátorok tárolják. A töltések külső órajellel vezérelt MOS tranzisztorokon haladnak keresztül egyik tárolóból a másikba, megvalósítva ezzel a változtatható idejű késleltetést. Bár a késleltetés során a jelszintek nem kerülnek kvantálásra, az órajel alkalmazása diszkrét idejű, mintavételezett jelet eredményez. Ennek következményeként az átlapolódás elkerülése érdekében aluláteresztő szűrésre van szükség. Ez a BBD chipek átviteli tulajdonságaival együtt sok zenész által kedvelt, jellegzetes hangzást eredményez, melynek köszönhetően a BBD alapú effektek napjainkban is népszerűek, sőt számos digitális jelfeldolgozást alkalmazó delay effekt is ezt a hangzást modellezi. A digitális rendszerek számos előnnyel rendelkeznek, ezek közé tartozik például a presetek kezelése, azaz a felhasználó által beállított paraméterek elmentése és betöltése vagy a késleltetési idő lábkapcsolóval történő beütése (tap tempo).

Szakedolgozatomban egy mikrokontrollerrel vezérelt BBD alapú delay gitáreffekt megépítését tűztem ki célul, mely a jellegzetes analóg hangzás mellett a digitális rendszerek előnyeivel is rendelkezik. Az analóg részhez számos kapcsolás található az interneten, ezek közül választottam egyet, melyet breadboard paneleken építettem meg. Ezután az effekt paramétereit szabályzó hagyományos potmétereket digitálisakra cserélve, illetve a mikrokontrollerrel a megfelelő órajelet előállítva megvalósítottam az effekt digitális vezérlését. Ezt követte a nyomtatott áramkör tervezése. A nyomtatott áramkör legyártására a félév során már nem volt idő, így az impulzusválasz és a frekvencia átvitel mérését a breadboardon megépített áramkörön végeztem el.

A dolgozat felépítése a következő: először ismertetem a BBD chipek történetét, működési elvét és az ezeken alapuló népszerű effektek felépítését és működését. A következő fejezetben a megépített áramkört tárgyalom, kitérve a szükséges

tápfeszültségek előállítására, az analóg jelút jellemzőire, és a digitális vezérlést megvalósító áramköri elemekre. Emellett ismertetem az áramkör megfelelő működéséhez szükséges kalibrálás folyamatát. Ezt követi a mikrokontrollert vezérlő C nyelven írt szoftver bemutatása, majd a következő fejezetben összefoglalom a nyomtatott áramkör tervezése során figyelembe vett szempontokat. Ez után következik az áramkörön elvégzett mérések tárgyalása. Itt ismertetem az alkalmazott mérési elrendezést és a megépített áramkör jellegzetes tulajdonságait a mérési eredmények alapján. Végül összefoglalom a félév során végzett munka eredményeit és áttekintem a további fejlesztési és alkalmazási lehetőségeket.

## 2 BBD IC-k felépítése és működése

### 2.1 Történeti áttekintés

A Bucket Brigade Device, azaz BBD chip feltalálása 1968-ra tehető és F. Sangster, illetve K. Teer nevéhez fűződik. Az elnevezés a régi tűzoltó brigádok működését idézi, ahol az emberek láncot alkotva adták tovább egymásnak a tűzoltáshoz használt vízzel teli vödröket. Ehhez hasonlóan haladnak végig az analóg jel feszültségével arányos töltések egy sor késleltető elemen.

Az első, bipoláris tranzisztorokkal megvalósított eszközök esetén az egyes késleltető elemek közötti átvitel rossz hatékonyságú volt, ezért ezek csak 100-200 késleltető fokozatot tartalmaztak, mely a legtöbb alkalmazás szempontjából elégtelen volt. Erre később a MOS tranzisztorok alkalmazása jelentett megoldást [1]. A kezdeti sikertelenséget követő fejlesztések eredményeként a BBD chip a 70-es évek közepére gitáreffektekből is használhatóvá vált és így lehetőséget adott az akkoriban elérhető nagy, drága, mozgó alkatrészeket tartalmazó mágnesszalagos késleltetők leváltására.

Azonban a digitális eszközök fejlődésének eredményeként a BBD-k iránti igény lecsökkent, gyártásukat 1999-ben leállították. Napjainkban – talán a gitáreffektek általános népszerűségének köszönhetően – újjáéledt a BBD chipek iránti érdeklődés. 2015-ben az Xvive Audio cég újraindította a klasszikus Panasonic IC-kkel azonos alkatrészek gyártását, számos érdeklődő számára elérhetővé téve az addig nehezen beszerezhető, drága alkatrészeket.

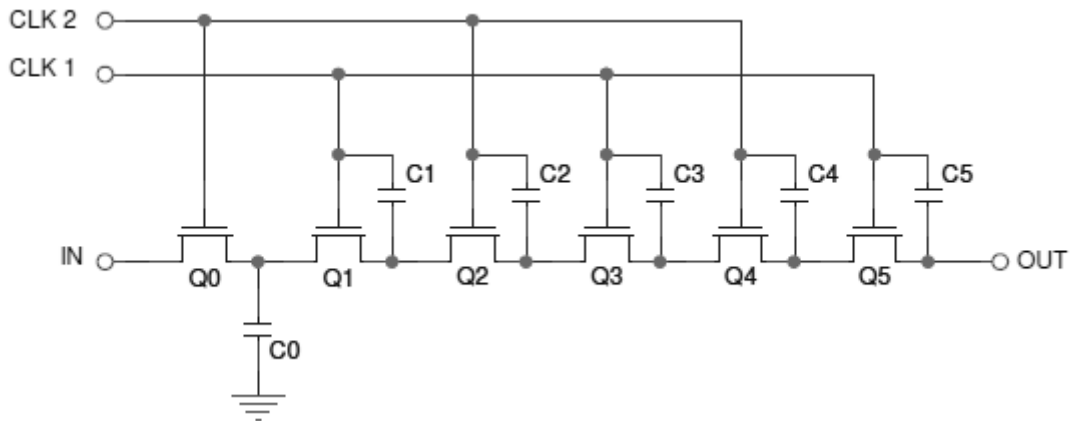
### 2.2 Felépítés, Működés

Az eszköz működési elvét először a legegyszerűbb felépítést (2-1. ábra) feltételezve mutatom be, majd áttekintést adok az ezt követő fejlesztésekről és a különleges alkalmazásokat (pl. zengető effekt) lehetővé tevő felépítésekről.

A chip számos soros késleltető elemet tartalmaz, melyek mindegyike egy töltést tároló kondenzátorból és egy kapcsoló MOS tranzisztorból épül fel. A páros és páratlan sorszámú elemek vezérlése két, ellentétes fázisban érkező órajellel történik. Az analóg jel feszültségével arányos mennyiségű töltés először CLK2 magas állapotában a C0 kondenzátoron jelenik meg. CLK2 lefutásával a Q0 tranzisztor lezár, leválasztva C0-t a bemenetről. Az ezzel egy időben felfutó CLK1 órajel hatására a Q1 tranzisztor kinyit, C0

töltését a C1 kondenzátorra áthelyezve. Minden órajel ciklusban ehhez hasonlóan halad tovább az analóg jelet leíró töltés a kimenetig. A chip tehát folytonos értékű, diszkrét idejű rendszerként működik, melynek mintavételezési frekvenciája az órajel frekvenciával egyezik meg.

### 2.2.1 Alapvető felépítés

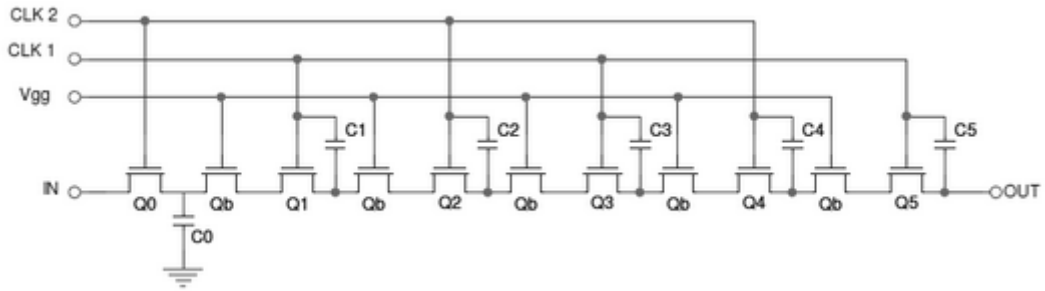


2-1. ábra Legegyszerűbb BBD felépítés [2]

Ebben az elrendezésben (2-1. ábra) az egyes kondenzátorok feszültsége a véges drain-source ellenállás miatt visszahat a MOS tranzisztorok source feszültségére, tehát az aktuális minta értéke befolyásolja a következő minta referencia feszültségét. Így az aktuális minta értékének egy tört része hozzáadódik a következő mintához, ami a magas frekvenciák csillapítását eredményezi [1]. Ennek következtében az ilyen felépítéssel rendelkező eszközök csak kevés késleltető elemet tartalmaztak és audio területen történő felhasználásra alkalmatlanok voltak. Ehhez képest jelentős előrelépést jelentett az egymást követő késleltető elemek elválasztása MOS tranzisztorok alkalmazásával [3].

### 2.2.2 MOS tetródák alkalmazása

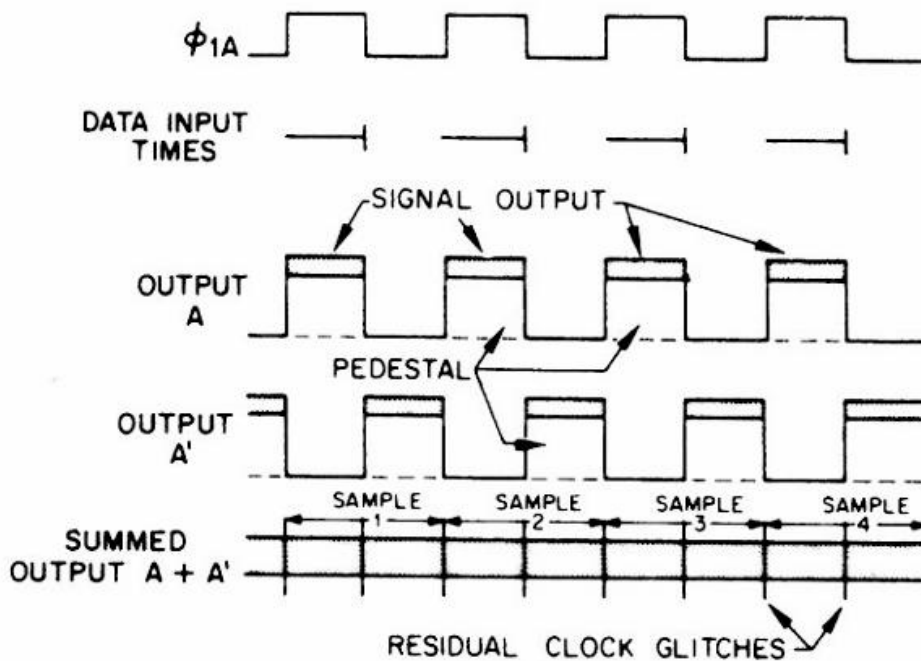
A fent említett visszahatás jelentősen csökkenthető egy második, telítésben működő tranzisztor alkalmazásával, melyeket a 2-2. ábrán Qb jelöl. Ezek az eredeti tranzisztorokkal együtt tetródákat alkotnak, melyek az eredeti egyetlen tranzisztorhoz képest nagyobb belső ellenállással rendelkeznek, jelentősen csökkentve az átvitel során fellépő veszteséget. Ennek köszönhetően lehetséges akár több ezer tárolóból álló késleltetők megvalósítása [1].



2-2. ábra MOS tetródák alkalmazása [2]

### 2.2.3 Komplementer kimenetek

A következő jelentős fejlesztés a komplementer kimenetek alkalmazása volt: az utolsó két késleltető elemhez source-követő tranzisztorokat csatlakoztatva két kimenetet kapunk. Ez folytonosabb audio jelet eredményez, ugyanis egyetlen kimenet esetén csak CLK1 magas állapota alatt jelenik meg a jelnek megfelelő kimenet, a 2-3. ábra Output A sávjában látható módon. Ezzel szemben az A' kimenethez tartozó tranzisztort az előzővel ellentétes fázisú órajel vezérli, így itt az A kimenet alacsony állapotában jelennek meg a jelet leíró feszültségek. A BBD chipek egyik nem ideális, de jellemző tulajdonsága, hogy az órajel változásai az adott órajelhez tartozó kimeneten tranziens jelenségeket (clock glitch) eredményeznek [4]. Mivel ezek a két kimeneten ellentétes fázisban jelennek meg, ezért a kimenetek összegzésével nagymértékben elnyomhatók.

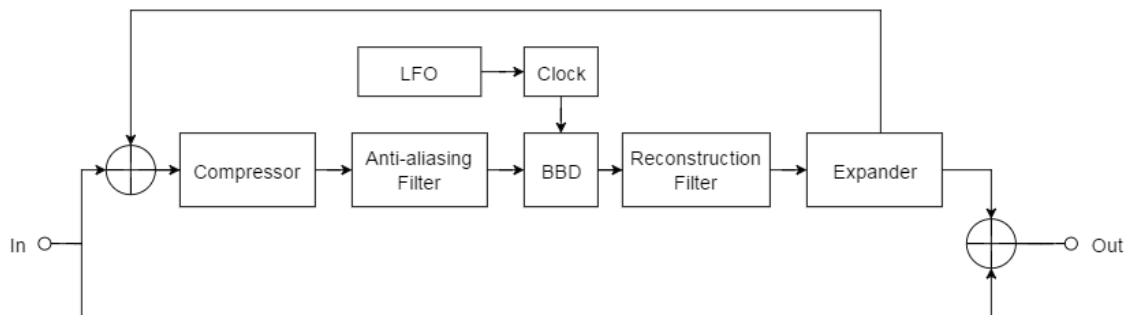


2-3. ábra Komplementer kimenetek működése [5]





### 3 BBD alapú áramkörök általános felépítése



3-1. ábra A BBD alapú áramkörök általános felépítése

BBD chipek felhasználásával természetesen nem csak delay effekt létrehozására van lehetőség. A 3-1. ábrán látható felépítés alapján az összes népszerű jelkésleltetésen alapuló effekt megvalósítható.

#### 3.1 BBD-vel megvalósítható zenei effektek

Effekt	Késleltetés(ms)	Moduláció	Mélység(ms)	Visszacsatolás
Vibrato	5-10	0.1-5 Hz szinusz	0-3	nincs
Chorus	1-30	kisfrekvenciás zaj	1-30	van
Flanger	0-15	0.1-1 Hz szinusz	0-2	van

3-2. ábra Késleltetésen alapuló effektek jellemző paraméterei

A késleltetési időt, a visszacsatolást és a modulációt változtatva több népszerű effekt létrehozható, például:

- **Vibrato:** Csak az LFO-val modulált 5-10 ms késleltetésen átesett jelet halljuk. Az idő rövidege miatt a késleltetést nem érzékeljük, azonban a késleltetési idő változtatása hangmagasságbeli változást eredményez.
- **Chorus:** A vibratohoz hasonló, azonban itt a halkabb késleltetett jel mellett az eredeti jel is hallható.
- **Flanger:** Szintén modulált késleltetést alkalmaz, a kimenet azonban az eredeti és a késleltetett jel összege vagy különbsége.

Az említett effektek jellemző paramétereit a 3-2. ábrán látható táblázat foglalja össze [6]. Bár a különböző effektek működési alapelve hasonló, mégis szükséges az adott alkalmazáshoz megfelelő BBD IC kiválasztása, ezt szemlélteti a 3-3. ábrán látható táblázat [7]:



### SELECTION GUIDE by STAGES

Stage	Type No.	Noise	Application	
64	Dual Type	<b>MN3003</b>	Low Noise Types	
128		<b>MN3006</b>		
3, 5, 190	Triple Type	<b>MN3012</b>		
256		<b>MN3009</b>		
512	Single Type	<b>MN3004</b>	Standard Types	
		<b>MN3002</b>		
	Dual Type	<b>MN3001</b>		Echo Effect (Signal Delay over 10 msec.)
		<b>MN3010</b>		
1024	Single Type	<b>MN3007</b>	Low Noise Types	
2048		<b>MN3008</b>		
4096		<b>MN3005</b>		
3328		Multi-Type		<b>MN3011</b>
			Reverberation Effect (Signal Delay over 100 msec.)	
			Reverberation Effect (Signal Delay over 100 msec.)	
			Reverberation Effect (Six different delay outputs)	

\*1 Vibrato Effect: Several Hz modulation effect of the clock frequency for BBD.  
 \*2 Chorus Effect: Mixing effect of the original signal and the attenuated delayed signal.  
 \*3 Phasor/Flanger Effect: Effect of either the sum or difference of the original signal and the delayed signal.  
 \*4 Double Voicing Effect: Mixing effect of the original signal and the delayed signal.

### 3-3. ábra Különböző BBD IC-k

## 3.2 Órajel generátor

A BBD-k működéséhez szükséges ellentétes fázisú órajeleket általában egy erre a célra tervezett IC (pl. MN3101) állítja elő. Az órajel frekvenciája – és ezzel együtt a késleltetési idő - egy rezgőkör hangolásával változtatható. Az órajel kisfrekvenciás modulálásával (LFO) chorus ill. vibrato effekt is megvalósítható. Az MN3101 órajel IC emellett biztosítja a BBD-k működéséhez szükséges VGG feszültséget.

## 3.3 Szűrők

Delay effekt esetében a 20-250 ms tartományba eső késleltetési idők jellemzőek [7]. A késleltetési idő az alábbi képlettel kiszámítható:

$$t_D = \frac{N}{2 \cdot f_{cp}}$$

ahol N a késleltető fokozatok száma (MN3005 esetén N = 4096),  $f_{cp}$  az órajel frekvenciája. Ebből következik, hogy az említett 250 ms-os késleltetési időhöz tartozó órajel frekvencia:

$$f_{cp} = \frac{4096}{2 \cdot 0.250} = 8192 \text{ Hz}$$

Mivel az bőven a hallható tartományon belül van, ezért fontos az órajel komponens minél nagyobb mértékű elnyomása. Ezt segítik (a BBD két kimenetének keverése mellett) a késleltetők előtt és után található szűrők.

Emellett, mivel diszkrét idejű rendszerről van szó, a mintavételi tétel értelmében szükséges a bemenő jel sávkorlátozása, ezzel elkerülhető az  $\frac{f_{cp}}{2}$ -nél nagyobb frekvenciájú komponensek átlapolódása (aliasing). A legtöbb effektpedálban erre a célra egyszerűségük miatt tranzisztoros harmadfokú Sallen-Key szűrőket alkalmaznak (3-4. ábra) [8]. A mintavételezett jel visszaállításához szintén aluláteresztő szűrés szükséges, erre gyakori megoldás egy harmadfokú Sallen-Key szűrő, melyet egy másodfokú szűrő követ. Ezek a szűrők természetesen befolyásolják a késleltetésen átesett jel hangzását, ezért fontos megvizsgálni a jelre gyakorolt hatásukat. Erre a megépített áramkörben alkalmazott szűrőket vizsgálva a 3.3. fejezetben térek ki.

A másodfokú szűrő átviteli függvénye az alábbi egyenlettel írható le [8]:

$$H(s) = \frac{1}{R^2 \cdot C_1 \cdot C_2} \frac{1}{s^2 + \frac{2}{R \cdot C_1} s + \frac{1}{R^2 \cdot C_1 \cdot C_2}} \quad (1)$$

Harmadfokú szűrő esetén:

$$H(s) = \frac{b_0}{a_3 s^3 + a_2 s^2 + a_1 s + a_0} \quad (2)$$

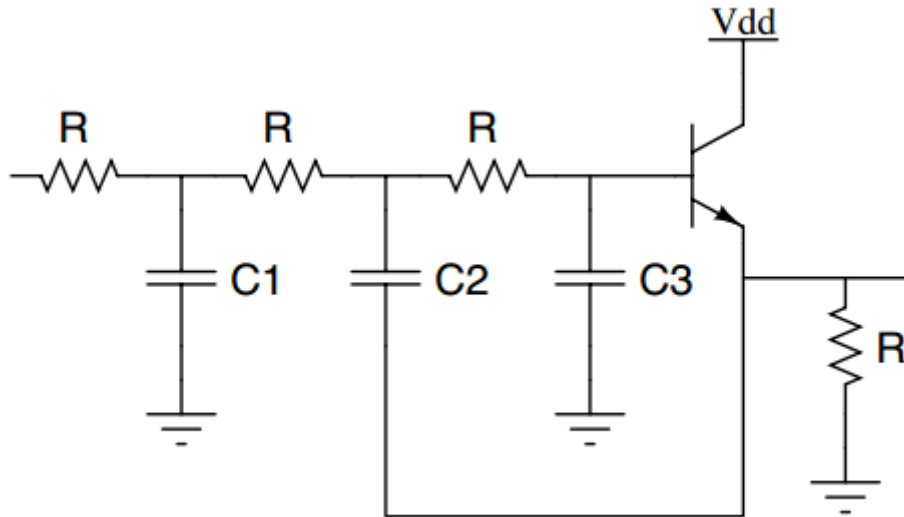
ahol

$$b_0 = a_0 = \frac{1}{R^3 \cdot C_1 \cdot C_2 \cdot C_3} \quad (3)$$

$$a_1 = \frac{1}{R^2 \cdot C_1 \cdot C_2} + \frac{1}{R^2 \cdot C_2 \cdot C_3} \quad (4)$$

$$a_2 = \frac{1}{R \cdot C_1} + \frac{1}{R \cdot C_2} \quad (5)$$

$$a_3 = 1 \quad (6)$$

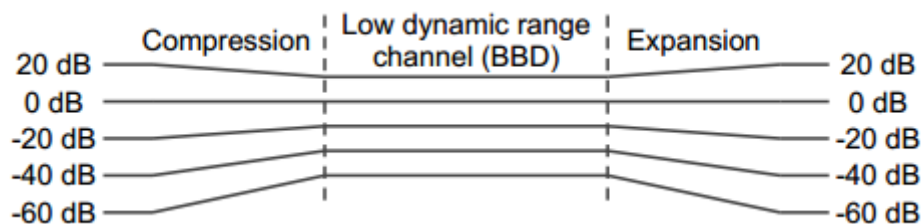


3-4. ábra Harmadfokú Sallen-Key szűrő

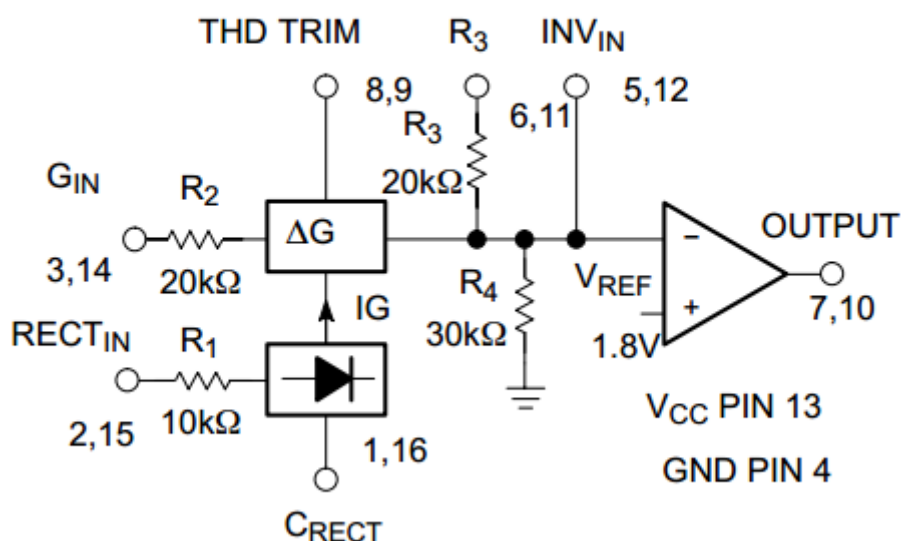
### 3.4 Kompander

A BBD chip nem ideális tulajdonságai korlátozott dinamikatartományt eredményeznek [8], a kivezérelhetőség határa jelentősen kisebb a tápfeszültségnél. Alacsony harmonikus torzítás és a legjobb jel-zaj viszony eléréséhez ezt a határt minél jobban megközelítő bemenő jelre van szükség. Ez különösen igaz a delay áramkörökben alkalmazott széles órajel tartományban működő, sok késleltető elemet tartalmazó BBD-k esetén [8]. Az említett hatások csökkentésére szolgálnak a kompander áramkörök, melyek egy dinamikakompresszort és egy expandert tartalmaznak. A kompresszor 2:1 arányban [9] korlátozza az audio jel dinamikáját a BBD chip előtt, azaz a bemenő jel 2 dB-es megváltozása a kimeneten 1 dB változást eredményez. A késleltetés után az expander bemenetén mérhető 1 dB-es változás a kimeneten 2 dB-es jelszint változást eredményez, mely az eredetit megközelítő dinamika visszaállítására szolgál. Ezt szemlélteti a 3-5. ábra.

A dinamikaszabályozás áramköri kapcsolástól függően történhet az anti-aliasing és visszaállító szűrőkön kívül, vagy közvetlenül a BBD be- és kimenetén. Analóg áramkörök esetén a megvalósítás legegyszerűbb módját a kompander IC-k jelentik (pl. NE570, SA571). Ezek egyetlen tápfeszültséget igényelnek, és két azonos áramkört tartalmaznak, melyek mindegyike működhet expanderként vagy kompresszorként [9].



3-5. ábra A kompander hatása [8]



3-6. ábra A kompander egy csatornájának blokkvázlata

A kompander IC egy csatornájának felépítését mutatja a 3-6. ábra [9]. Ez egy műveleti erősítő és a munkapont beállításért felelős áramkörök mellett egy full-wave egyenirányítót és egy változtatható erősítésű cellát ( $\Delta G$ ) tartalmaz. A  $\Delta G$  cella erősítését az egyenirányító átlagolt kimenete határozza meg, melynek időállandója a  $C_{REF}$  kondenzátor megválasztásával befolyásolható az alábbi képlet szerint [9]:

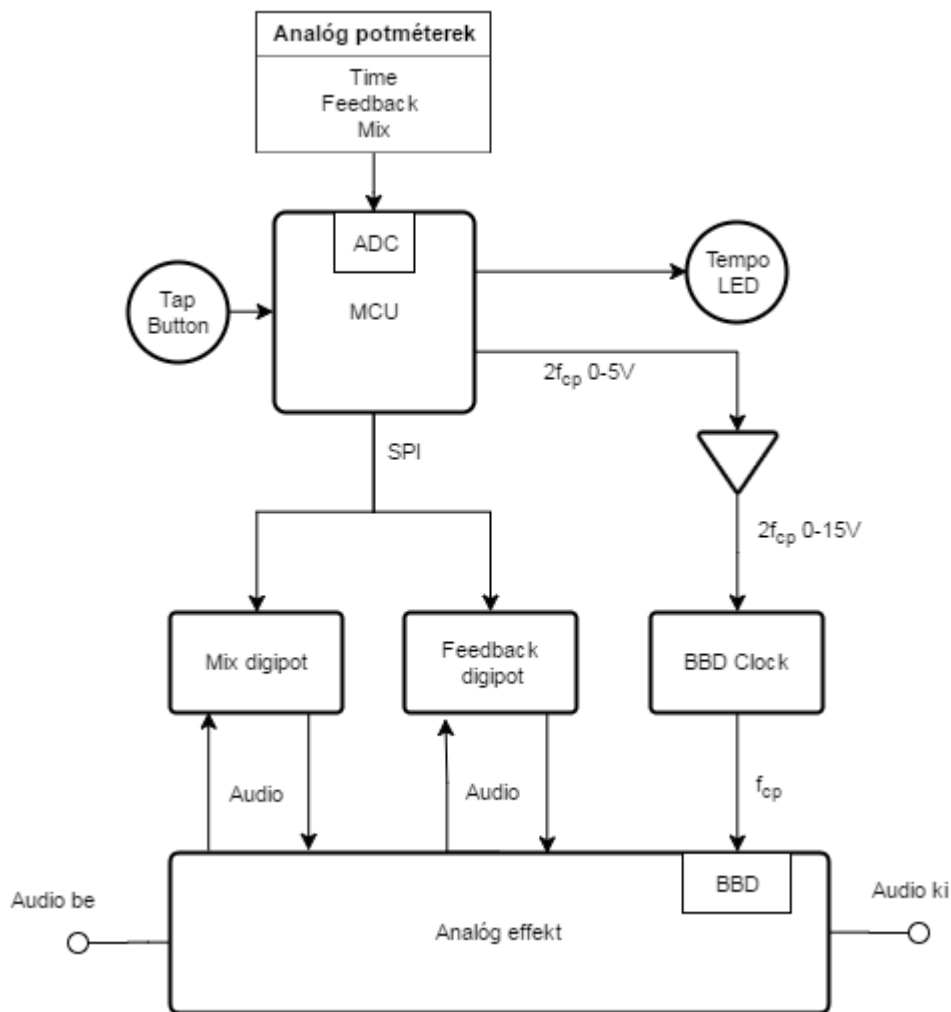
$$\tau = 10k\Omega \cdot C_{RECT}$$

Az IC felhasználására jó példa a megépített delay áramkör, így a konkrét alkalmazás részleteit a 4.2 fejezetben ismertetem.

### **3.5 Visszacsatolás**

A késleltetett jel visszacsatolásával nagymértékben befolyásolható az effekt hangzása. Flanger, ill. chorus esetén a visszacsatolás erőteljesebbé teheti a hatást, míg delay esetében fokozatosan lecsengő ismétlődéseket, vagy akár gerjedést eredményez. A visszacsatolás változtatása reverb esetén is a lecsengési időt befolyásolja. Vibrato esetén csak a hangmagasság változtatása a cél, így itt nincs visszacsatolás. A visszacsatolás megvalósításának módja az adott áramkörtől függ, a megoldható a kompresszor és expander között, vagy azokon kívül.

## 4 A megépített áramkör



4-1. ábra A megépített áramkör blokkvázlata

BBD alapú delay effektekhez számos kapcsolás található az interneten. Az általam megvalósított áramkör analóg részének alapjául a Boss DM-2 pedál, illetve a "Madbean Aquaboy Deluxe" projekt szolgált, melynek dokumentációja a kapcsolási rajz mellett számos hasznos információt tartalmaz az áramkör működésével, megépítésével, kalibrálásával kapcsolatban. [10]

Az általam megépített áramkör három változtatható paraméterrel rendelkezik:

- Time: a késleltetési idő beállítása 40 és 400 ms között
- Feedback: a visszacsatolás mértékét változtatja, így meghatározva a lecsengési időt
- Mix: a késleltetett jel hangereje

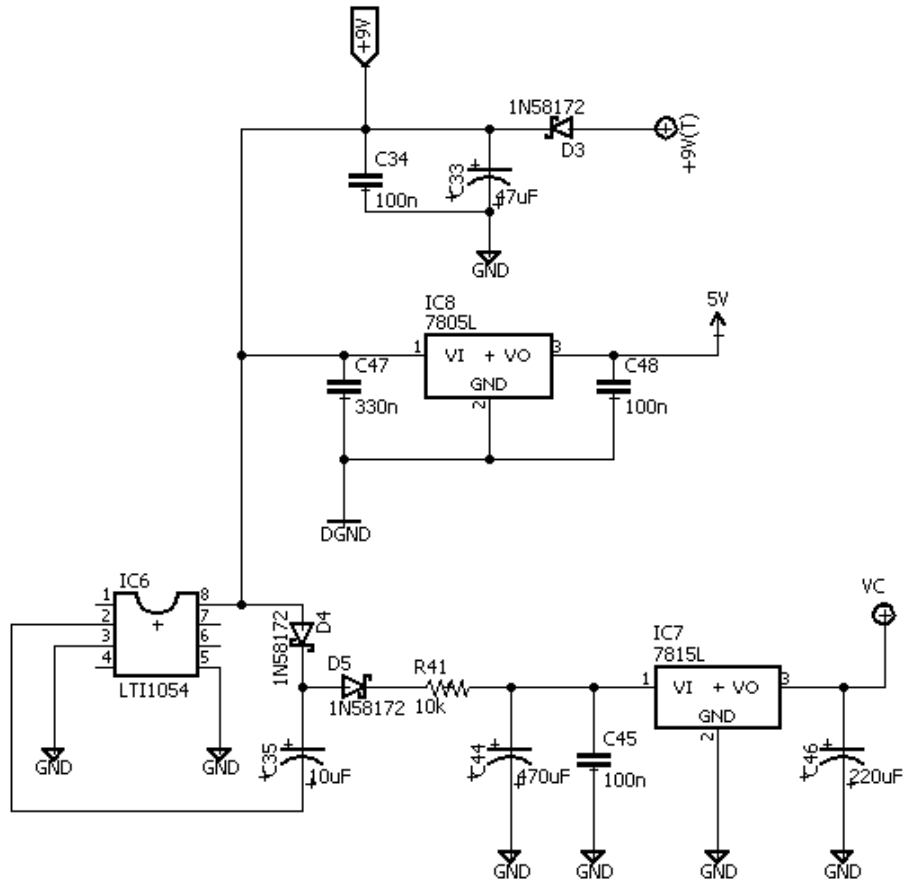
A "tap button" nyomógomb a késleltetési idő beállítására szolgál a gombnyomások között eltelt idő alapján, azonban ez jelenleg nem működik megbízhatóan, további szoftveres fejlesztést igényel. A "tempo LED" felvillanásai a beállított késleltetési időről adnak visszajelzést. Az áramkör felépítését a 4-1. ábra szemlélteti.

A kapcsolási rajz kiválasztása után a következő lépés az alkatrészek beszerzése volt. Az áramkör megépítéséhez szükség volt néhány nehezebben elérhető alkatrésze. Az MN3005-ös BBD chipeket és az MN3101 IC-t egy ausztrál weboldalról rendeltem, mivel magyarországi forgalmazót nem találtam. A többi felhasznált alkatrész hazánkban is egyszerűen beszerezhető.

## 4.1 Táp

A szükséges tápfeszültségek előállításánál a kiindulási pontot az effektpedálok esetén általánosan elterjedt 9V jelentette. A mikrokontroller, illetve a digitális potméterek működéséhez szükséges 5V-os feszültség ebből egy L7805 feszültség stabilizátorral és a megfelelő szűrő kondenzátorokkal [11] egyszerűen előállítható. Ennek megvalósítását mutatja a 4-2. ábra.

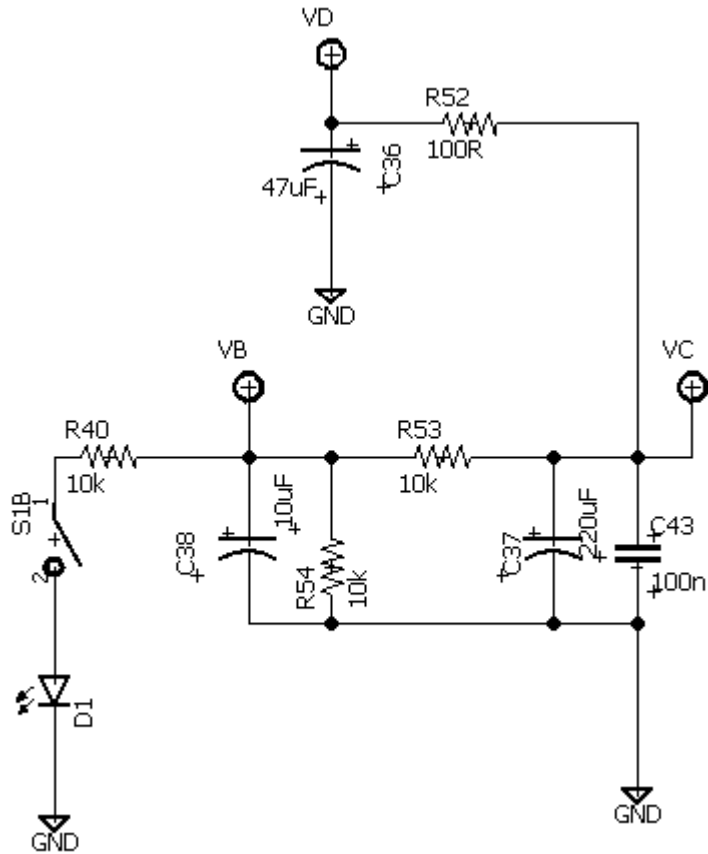
A BBD IC-k számára szükséges 15V feszültség előállítása két lépésben történik: először egy LT1054 feszültség átalakító IC a 9V-os feszültség duplázásával 18V-ot állít elő, majd egy L7815 feszültség stabilizátor ezt 15V-ra csökkenti (4-2. ábra). Ebből már passzív alkatrészekkel előállítható az analóg áramkör számára szükséges VB, VC, ill. VD feszültség, (4-3. ábra). A 15V-os VC feszültség a műveleti erősítő és a tranzisztorok tápfeszültsége. Ennek fele a VB, mely az említett áramköri elemek munkapontjának beállítására szolgál. A BBD chipek precíz munkapont beállítást igényelnek, ezt biztosítja a VD feszültség és a BIAS1, ill. BIAS2 trimmer.



4-2. ábra Táparamkör (1)

Mivel a BBD-k és az MN3101 órajel-generator is -15V-ot igényel a VDD lábán, ezért ezeket a következőképpen kell bekötni: az IC-k GND lábai a +15V-os VD feszültséget kapják, míg VDD lábak az áramkör földjére csatlakoznak [10].





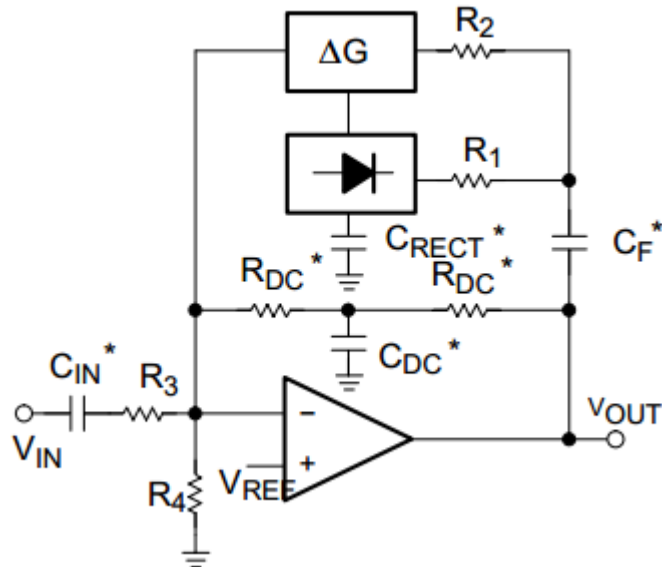
4-3. ábra Tápáramkör (2)

## 4.2 Analóg jelút

Az általam épített delay effekt felépítése megfelel a 3-1. ábrán látható struktúrának, eltekintve a modulációt megvalósító LFO részegység hiányától. Ezt digitális vezérlés esetén érdemes lehet analóg áramköri elemek helyett szoftveresen megoldani. Az audio jel a 3-1. ábrának megfelelően két úton halad. A tranzisztoros és műveleti erősítős fokozatokból álló "tiszta" ág nem eredményez hangzásbeli változást.

### 4.2.1 Kompreszor

A késleltetésért felelős jelút első eleme egy kompreszor áramkör, melyet az (SA571-el megegyező) NE570-es IC egyik fele valósít meg.



4-4. ábra A kompresszor blokkvázlata

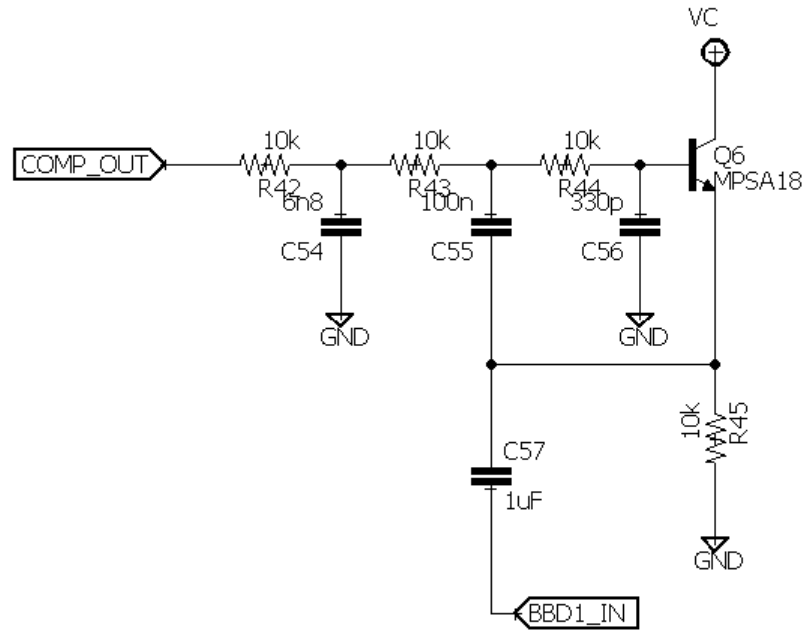
A 4-4. ábrának megfelelően a bemenő (tiszta) jel a  $C_{IN}$  csatoló kondenzátoron keresztül a műveleti erősítő invertáló bemenetére jut. Mind az egyenirányító, mind a  $\Delta G$  cella bemenete a műveleti erősítő kimenete. Az így létrejövő AC negatív visszacsatolásnak köszönhetően a bemenő jel megváltozása a kimeneten kisebb mértékű változást eredményez. A stabilitás érdekében szükséges DC visszacsatolást az  $R_{DC}$  ellenállások határozzák meg [9]. Az időállandót meghatározó  $C_{RECT}$  kondenzátorok értéke jellemzően 220 nF és 1  $\mu$ F között van. A megépített áramkörben ezt a szerepet a C10 és C28 kondenzátorok töltik be, melyek értéke 220 nF (ld.: Függelék 9-1. ábra). Ez az érték jó kompromisszum a dinamikusabályozás gyors működése és az erősítés "hullámossága" között. Az említett hullámosságot az okozza, hogy túl kicsi  $C_{RECT}$  kapacitás esetén a mélyebb frekvenciakomponensek befolyásolják az erősítést, ezzel a jel torzulását eredményezve [12].

#### 4.2.2 Átlapolódásgátló és rekonstrukciós szűrők

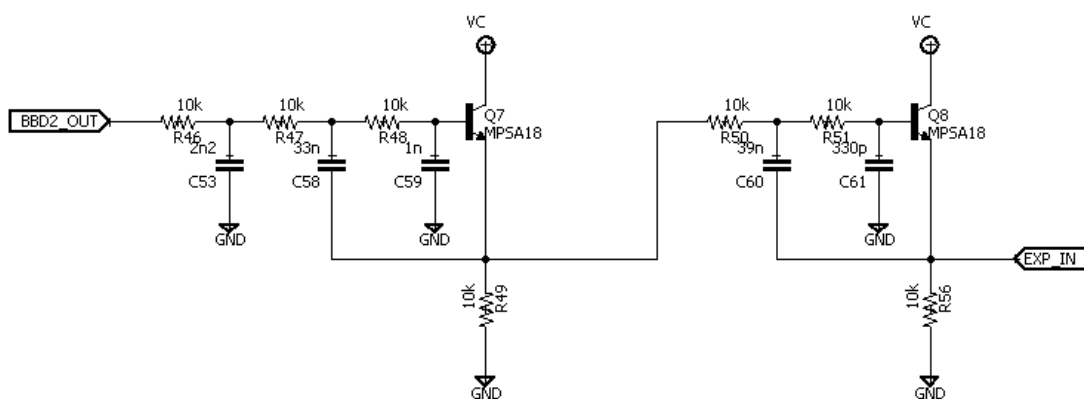
A kompresszort egy harmadfokú Sallen-Key szűrő követi, melynek alapja egy MPSA18 NPN tranzisztor (4-5. ábra). A szűrő kimenetén megjelenő csökkentett dinamikájú, aluláteresztő szűrésen átesett jel már alkalmas a BBD-vel történő késleltetésre. A két soros késleltető IC után a jel az előzőhöz hasonló harmadfokú, majd egy másodfokú aluláteresztő szűrőn halad át (4-6. ábra).

Az alkatrészértékek ismeretében a szűrők átviteli függvényei a 3.3 fejezetben ismertetett (1) - (6) egyenletek és a Matlab tf(), illetve bode() függvényeivel könnyedén

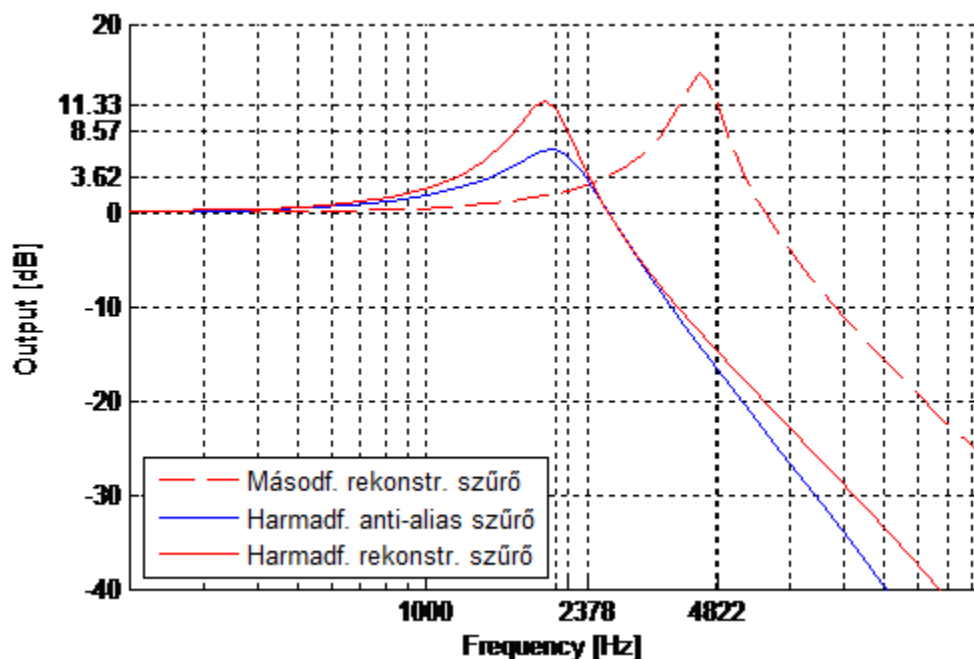
kiszámíthatók és ábrázolhatók. Ennek eredményét mutatja a 4-7. ábra. Az ábráról a -3dB-es törésponti frekvenciák egyszerűen leolvashatók. Látható, hogy a harmadfokú szűrők törésponti frekvenciája 2139, ill. 2378 Hz-nél van, ami megközelítőleg az ajánlott minimális órajel (6-10 kHz)[10] egyharmada. Ez jellemző érték ilyen áramkörök esetében [8].



4-5. ábra Harmadfokú Sallen-Key átlapolódásgátló szűrő



4-6. ábra Harmad- ill. másodfokú Sallen-Key rekonstrukciós szűrő

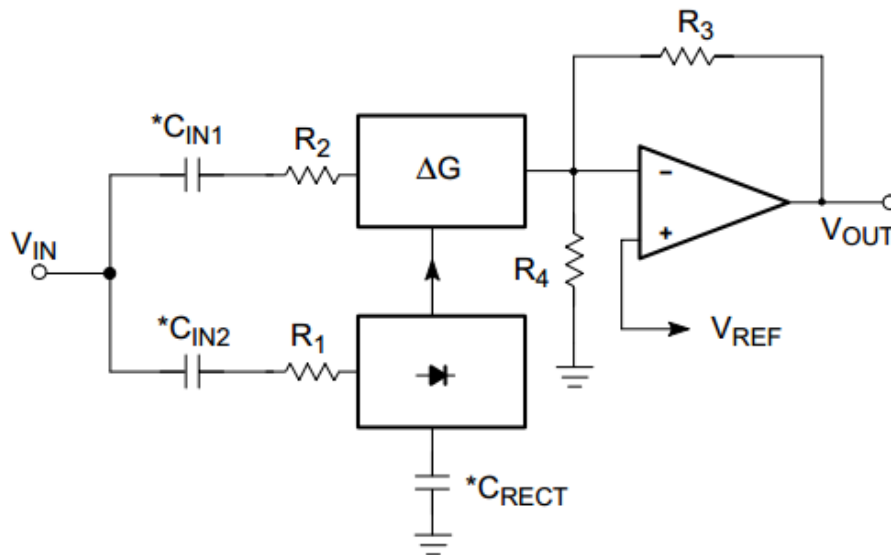


4-7. ábra Átlapolódásgátló és rekonstrukciós szűrők frekvenciamenete és törésponti frekvenciái

### 4.2.3 Expander

A rekonstrukciós szűrőt egy expander áramkör követi, melyet szintén az NE570 valósít meg (4-8. ábra). Itt a késleltetésen és szűrésen átesett jel AC csatolt módon az egyenirányító és a  $\Delta G$  cella bemenetére jut. Mivel  $\Delta G$  erősítése arányos az egyenirányító kimenetével [9], így például 6 dB csökkenés a bemeneten felére csökkenti az erősítést, ami a kimeneten 12 dB csökkenést eredményez. Ezzel megvalósul a 2:1 arányú expander funkció.

Az expander kimenetén egyrészt megtörténik a jel változtatható mértékű visszacsatolása a Feedback potméteren keresztül, másrészt megvalósul a tiszta és a késleltetett jel keverése. Utóbbi hangereje a Mix potméterrel szabályozható, a visszacsatolás maximális mértéke pedig a T1 trimmerrel állítható.



4-8. ábra Az expander blokkvázlata

#### 4.2.4 Bypass kapcsoló

Az S1 kapcsolóval lehetőség van a késleltető ág megszakítására. Így az effekt kikapcsolható, azonban az ismétlődések nem szűnnek meg egyből, hanem a Feedback potméter állásától függően fokozatosan elhalkulnak.

### 4.3 Digitális vezérlés

#### 4.3.1 A mikrokontroller

Az áramkör vezérlését egy Atmel Atmega32u4 mikrokontroller végzi. Ennek feladata az effekt paramétereikhez tartozó potméterek (Time, Feedback és Mix) értékeinek beolvasása, az ezeknek megfelelő parancsok elküldése a digitális potmétereknek SPI buszon keresztül, illetve a BBD-ket működtető órajel előállítás. Fontos megjegyezni, hogy a vezérlés ellátására egy sokkal egyszerűbb eszköz is megfelelne, azonban az említett mikrokontrollert már korábban megvásároltam, ezért úgy döntöttem, ezt használom a prototípus megépítéséhez. Ha több azonos áramkör megépítéséről lenne szó, akkor nyilvánvalóan fontos lépés lenne a lehető legolcsóbb alkalmas mikrokontroller kiválasztása.

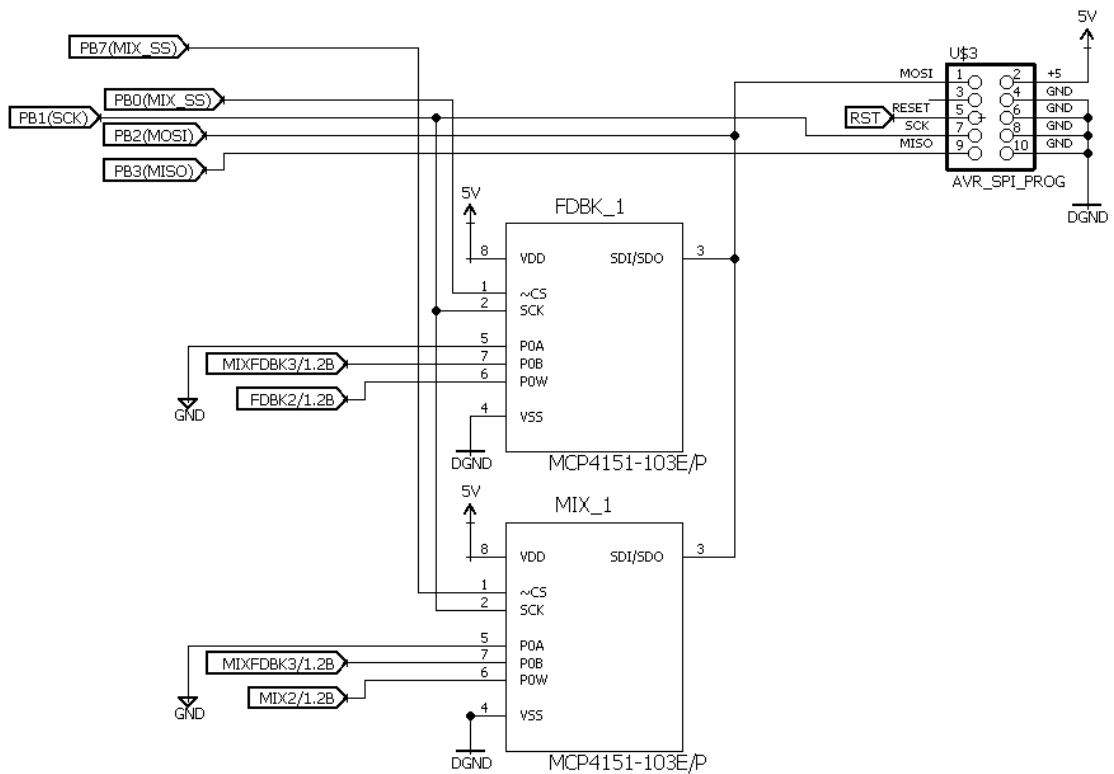
Az Atmega ADC bemeneteire kapcsolódik három potméter, melyekkel az effekt paramétereit szabályozhatók. A beállított késleltetési időről egy megfelelő időközönként felvillanó LED ad visszajelzést. A nyomtatott áramkörön a mikrokontroller mellett elhelyezett csatlakozó a programozást végző eszköz csatlakoztatására szolgál. Ez az SPI

buszra, illetve a  $\overline{\text{RESET}}$  lábra csatlakozik (4-9. ábra). A vezérlést megvalósító áramkör kapcsolási rajza a Függelék 9-2. ábráján látható. A mikrokontroller felhasznált lábainak irányát (ki- vagy bemenet) és funkcióját az alábbi táblázat foglalja össze:

Port	Láb	Irány	Funkció
B	0	output	Feedback Slave Select
	1	output	SPI
	2	output	
	3	output	
	4	output	Tempo LED
	7	output	Mix Slave Select
C	6	output	BBD Clock
D	1	input	Tap button
F	0	input	ADC: Time
	1	input	ADC: Feedback
	4	input	ADC: Mix

### 4.3.2 Potméterek

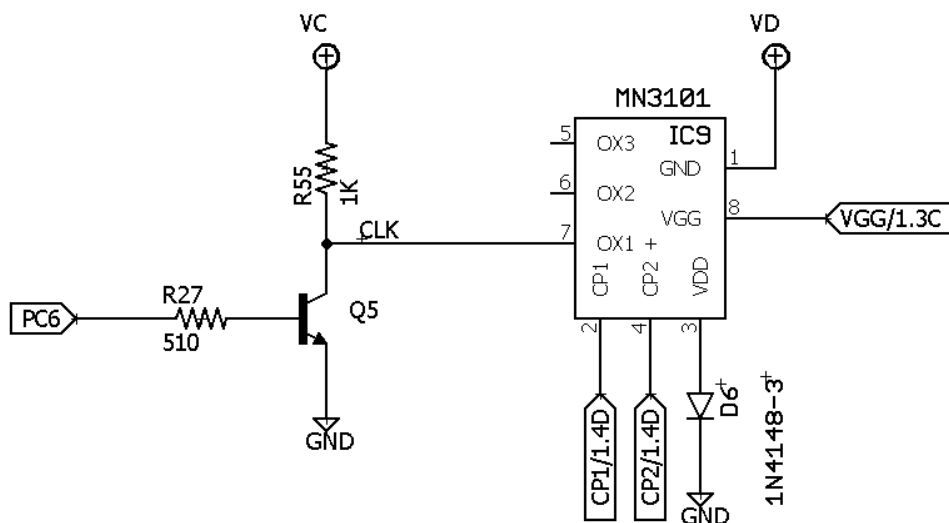
A delay effekt hagyományos analóg potmétereinek helyére digitális potméterek (Microchip MCP4151) kerültek, melyek vezérlése SPI buszon történik. A PDIP-8 tokozású IC-k 5, 6, ill. 7-es lába egy hagyományos potméter három kivezetésének felel meg. Az IC többi lába a tápfeszültség és az SPI kommunikáció fogadására szolgál. A kapcsolási rajz ezen részlete látható a 4-9. ábrán.



4-9. ábra A digitális potméterek és az ISP csatlakozó

### 4.3.3 BBD órajelek előállítása

Az BBD-k kétfázisú órajele úgy is előállítható, ha az MN3101 IC OX1 lábára négyszögjelet adunk [13], melynek forrása (megfelelő erősítés után) akár egy mikrokontroller is lehet. Ez az eredeti frekvencia felével rendelkező négyszögjeleket eredményez a chip CP1 és CP2 órajel kimenetein.



4-10. ábra Az órajel előállítása

Így a mikrokontroller egy kimenetén a kívánt frekvenciával rendelkező négyszögjelet előállítva megvalósítható a BBD késleltetési idejének változtatása. Továbbá ez a megoldás kevesebb alkatrészt igényel a Delay potméter digitális potméterrel történő helyettesítéséhez képest.

## **4.4 Az áramkör kalibrálása**

### **4.4.1 A kalibrálás célja**

A lehető legjobb hangzás biztosítása érdekében szükséges az áramkör kalibrálása. Ez trimmer potméterek beállításával végezhető el. Első lépésként fontos az egyes BBD chipek bemenetén mérhető DC munkaponti feszültség beállítása, ugyanis nem megfelelő beállítás esetén nagymértékben torzulnak a létrejövő visszhangok, sőt akár az is előfordulhat, hogy nem hallunk késleltetett jelet a kimeneten. Emellett az órajel által létrehozott zaj is elnyomható az ellentétes órajelekhez tartozó kimenetek megfelelő arányú keverésével. Nem megfelelő beállítás esetén a hallható tartományba eső órajel komponens magas, sípoló hangként hallható a kimeneten. Továbbá lehetőség van a visszacsatolás maximális mértékének változtatására, mellyel akár az is elérhető, hogy az effekt a Feedback potméter maximális állása közelében gerjedni kezdjen. Ezen beállítás meghatározása nem méréssel történik, leginkább a felhasználó preferenciájától függ.

### **4.4.2 Felhasznált eszközök**

Az áramkör tesztelése és kalibrálása során egy Red Pitaya [14] eszközt használtam, mely drága mérőműszerek helyettesítésére nyújt lehetőséget. Alapja egy ARM Cortex A9+ processzor és egy FPGA. Az eszköz két RF bemenettel, két RF kimenettel, ill. 14 bites AD/DA konverterekkel rendelkezik és 125MS/s mintavételi sebességre képes. A be- és kimeneti BNC aljzatokhoz hagyományos oszcilloszkóp mérőfejek csatlakoztathatók. A szoftver nyílt forráskódú, így az adott hardverrel számos funkció megvalósítható, azonban az alap szoftver is hasznos eszközöket tartalmaz. Ezek közé tartozik a 2 csatornás oszcilloszkóp és függvénygenerátor. Az eszköz Wi-Fi hálózaton keresztül vagy egy UTP kábel segítségével csatlakoztatható a számítógéphez. A kétcsatornás oszcilloszkóp, függvénygenerátor és egyéb funkciók legegyszerűbben a webböngészőben megjelenő grafikus kezelőfelület segítségével használhatók. Itt lehetőség van a képernyőn megjelenő értékek elmentésére CSV formátumban. Az

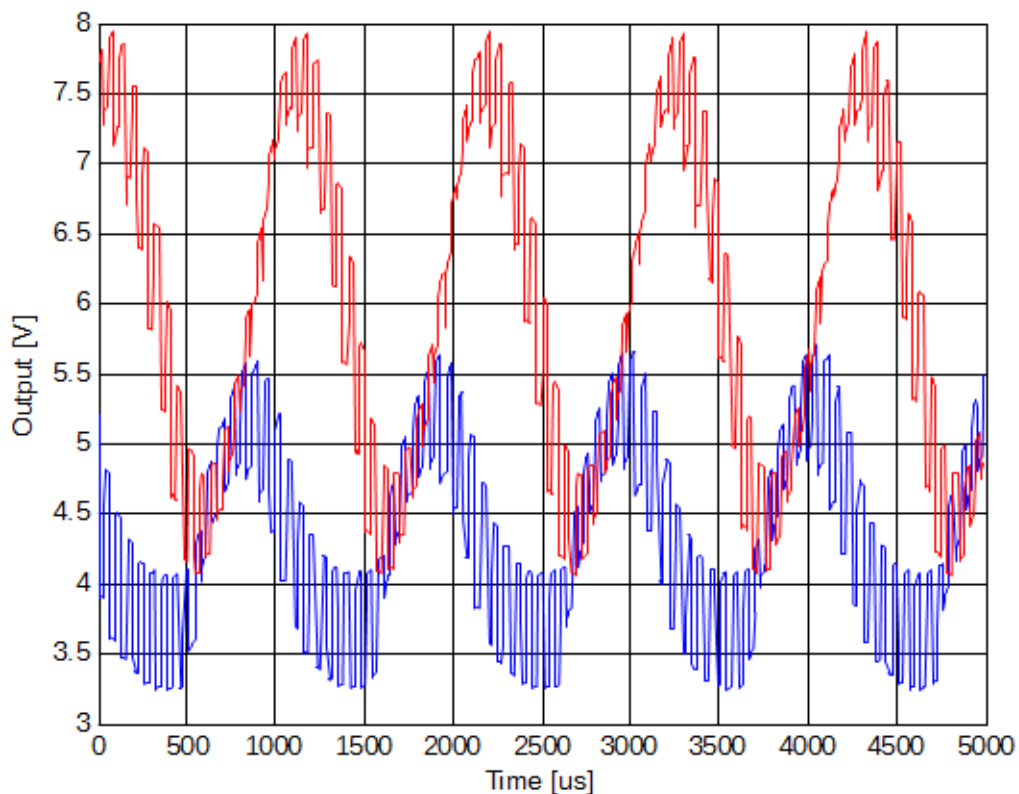


elmentett adatok könnyedén importálhatók Matlab változókként, ami további feldolgozásra, illetve precíz ábrák elkészítésére is lehetőséget nyújt.

### 4.4.3 A kalibrálás folyamata

A kalibrálás a BIAS1, BIAS2, CANCEL, ill. T1 trimmer potméterek beállításával történik. A BIAS trimmerek a BBD chipek bemeneteihez tartozó DC munkapontok beállítására, a CANCEL a második BBD két kimenetének keverésére szolgál. A T1 trimmerrel a visszacsatolás (Feedback) maximális mértéke állítható. A BBD órajelek minimális, ill. maximális frekvenciáját a mikrokontrolleren futó szoftver határozza meg.

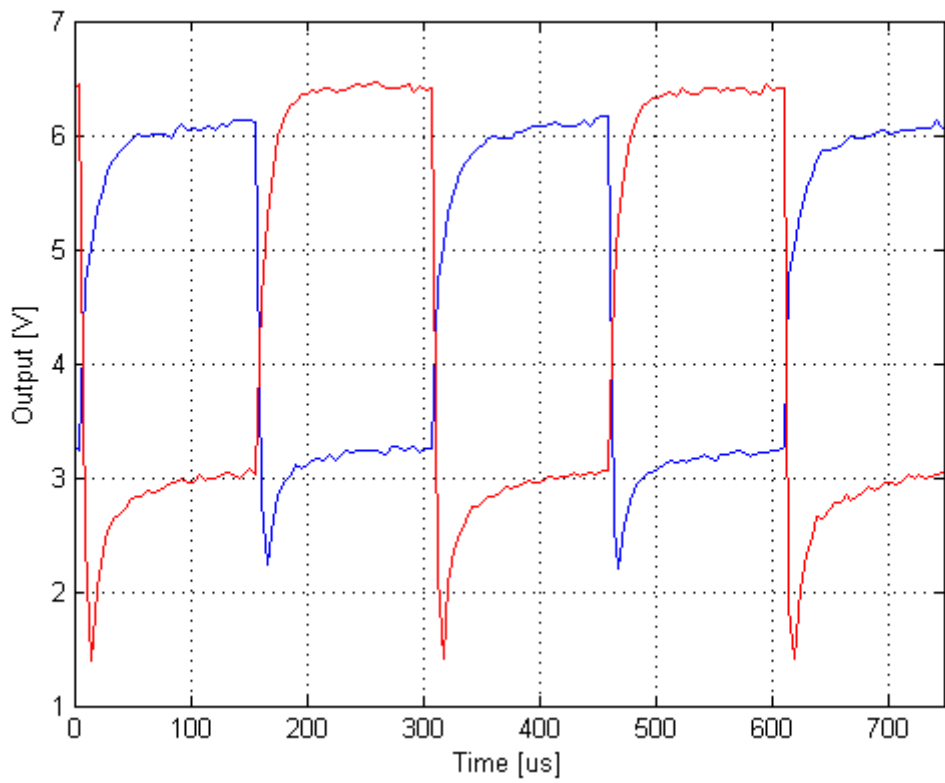
A kalibrálás megkezdése előtt az oszcilloszkóp frekvenciamérés funkcióját használva meggyőződtem arról, hogy az MN3101 IC órajel kimenetein az előírt frekvenciával rendelkező (10-100 kHz) négyzögjelek jelennek meg a Time potméter minimális, ill. maximális állásában. Ezután a Time, Mix, BIAS és CANCEL potmétereket középállásba, a Feedbacket kb. a maximum egyharmadára állítottam és ideiglenesen eltávolítottam a második BBD-t. A függvénygenerátoron 200mV amplitúdójú 1kHz-es szinuszelet állítottam be, kimenete az áramkör bemenetére csatlakozik. Az oszcilloszkóp bemenetét a megmaradt BBD egyik kimenetére (3. láb) kötöttem.



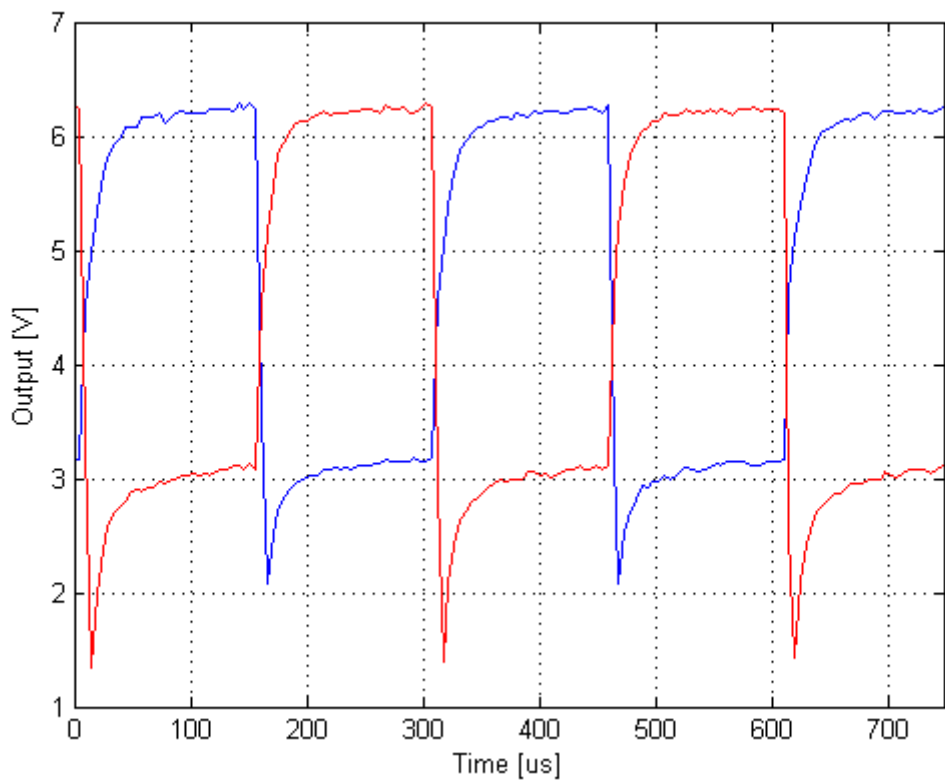
4-11. ábra Kimenet a munkapont beállítás előtt és után

A képernyőn megjelenő hullámformát figyelve a BIAS1 trimmert a legkisebb torzítást eredményező helyzetbe állítottam. A BIAS2 trimmer beállítását is ilyen módon végeztem a második BBD visszahelyezése után. A 4-11. ábra az IC kimenetét mutatja kalibrálás előtt (kékkel) és után (pirossal). A szinuszjelre szuperponálódó nagyfrekvenciás jel a kimeneten megjelenő órajel komponens. Ennek elnyomását szolgálja a BBD-k kimeneteinek keverése és a késleltetést követő aluláteresztő szűrés is.

A CANCEL trimmer beállításához a függvénygenerátort kikapcsolva az oszcilloszkóp két bemenetét a második BBD IC két kimenetére (3. és 4. láb) kötöttem és a CANCEL trimmert az órajel komponens maximális kioltását eredményező helyzetbe állítottam. A 4-12. ábrán megfigyelhető, hogy a kimeneteken megjelenő ellentétes fázisú jelek eltérő nagyságúak, így nem jön létre kioltás, az audio tartományba eső órajel a kimeneten is hallható lehet. A beállítás elvégzése utáni állapotot mutatja a 4-13 ábra. Itt azonos nagyságú, ellentétes fázisú jeleket láthatunk a két kimeneten, melyek így kioltják egymást.



**4-12. ábra Kimenet a CANCEL trimmer beállítása előtt**



**4-13. ábra Kimenet a CANCEL trimmer beállítása után**

## 5 Szoftver

A mikrokontroller és a hozzá tartozó szoftver feladata a Time, Feedback és Mix potméterek, illetve a nyomógomb állapotának feldolgozása, az ezeknek megfelelő parancsok elküldése SPI buszon a digitális potmétereknek és a BBD órajel előállítása.

### 5.1 Szoftverfejlesztő környezet

A szoftverfejlesztéshez rendelkezésemre állt egy Teensy 2.0 USB fejlesztő panel. Ez egy kisméretű eszköz, mely a beforrasztott tűskesor segítségével könnyedén elhelyezhető egy forrasztás nélküli próbapanelen. Az eszköz egy Atmel Atmega32U4 mikrokontrollert tartalmaz, melynek programozása USB-n keresztül történik.

A szoftvert C nyelven írtam. A megfelelő Makefile és a WinAVR C fordító használatával előállított .hex fájlt be kell tölteni a Teensy Loader alkalmazásba, majd az eszközön található nyomógombot lenyomva megtörténik a program feltöltése.

### 5.2 Implementáció

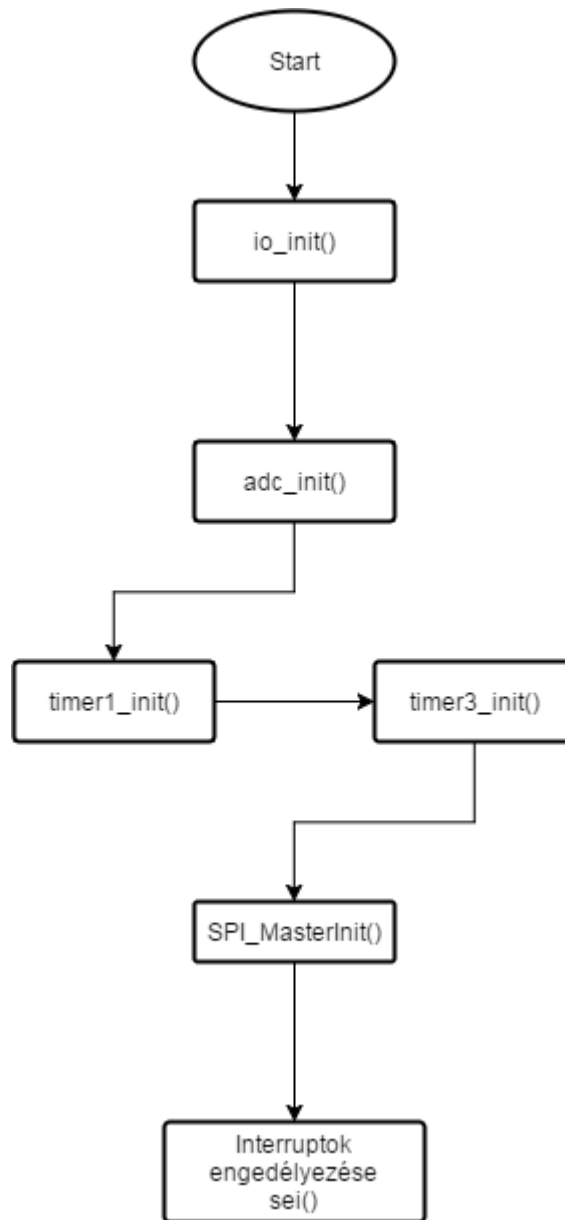
Az áttekinthetőség érdekében a C kódot a következő forrásfájlokra bontottam:

- BBD\_main.c: az inicializációhoz szükséges függvényhívásokat és a főciklust tartalmazza
- BBD.h: makródefiníciók, globális változók és függvények, interrupt rutinok deklarációja
- BBD.c: a függvénydefiníciókat tartalmazza.

Az elkészített C kódot a Függelék tartalmazza (52-58. oldal).

### 5.3 Inicializálás

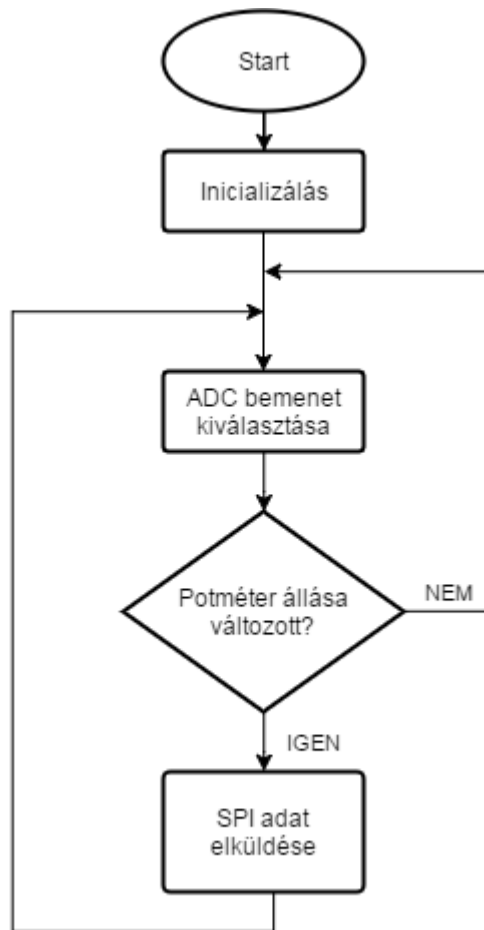
A program indulásakor megtörténik a ki- és bemenetek, az analóg-digitális átalakító, a tempó számlálásához és az órajel generáláshoz használt időzítők, illetve az SPI kommunikáció inicializálása, majd az interruptok engedélyezése.



5-1. ábra Az inicializáció folyamatábrája

### 5.3.1 Analóg bemenetek beolvasása

Az analóg bemenetek kezelése a főciklusban történik. A ciklus minden egyes lefutásakor egy bemenet kerül kiválasztásra egy switch – case szerkezet használatával, majd megtörténik a kiválasztott bemenetre csatlakozó potméter értékének beolvasása az `adcRead()` függvénnyel. Ha a potméter értéke változott az eltárolt előző értékhez képest, akkor a potméter értékét tároló változó felveszi az új értéket és a program végrehajtja az adott paraméterhez tartozó vezérlést. Ez a megvalósítás könnyedén bővíthető, lehetővé teszi további potméterek vagy kapcsolók csatlakoztatását és kezelését.



5-2. ábra A főciklus folyamatábrája

### 5.3.2 Órajel generálás

Az órajel előállításához egy "Clear Timer On Compare Match" (CTC) módban működő számlálót használtam. A számlálóhoz tartozó regiszter (TCNT3) értéke minden órajelciklus alkalmával növelődik egészen addig, amíg értéke egyezést mutat az OCR3A regiszterben beállított értékkel. A TCNT3 regiszter ekkor 0-tól kezdi újból a számlálást. Emellett a "toggle on compare match" beállítás miatt a számlálóhoz tartozó OC3A kimenet értéke invertálódik. Az OCR3A regiszter értékét módosítva megvalósítható a kívánt frekvencia előállítása, az alábbi képlet szerint [15]:

$$f_{OC3A} = \frac{f_{CLK\_IO}}{2 \cdot N \cdot (1 + OCR3A)}$$

Ahol  $f_{CLK\_IO}$  a rendszer órajelének frekvenciája (16 MHz), N a beállított prescaler (N = 1). Mivel a mikrokontroller által előállított négyszögjel frekvenciáját az MN3101 IC megfelelzi [13], ezért a BBD-k számára szükséges 10-100 kHz frekvencia kétszeresét

kell a mikrokontroller kimenetén előállítani. Az ehhez szükséges OCR3A érték az alábbi képlettel egyszerűen kiszámítható:

$$OCR3A = \frac{f_{CLK_{IO}}}{2 \cdot 2 \cdot f_{OCR3A}} - 1 \quad (N = 1 \text{ esetén})$$

Az ebből adódó OCR3A érték 39 és 399 között változik, ez a Time potméter értékének olvasásával és átskálázásával beállítható. Az előállított órajel a mikrokontroller PC6/OC3A lábán jelenik meg.

### 5.3.3 Digitális potméterek – SPI kommunikáció

A Feedback és Mix paraméterek szabályozásához használt digitális potméterek vezérlése SPI buszon keresztül történik. A vezérlés során a digitális potméterek nem küldenek adatot a mikrokontroller felé, így a megvalósított SPI kommunikáció egyirányú. A mikrokontroller Master, a potméterek pedig Slave üzemmódban működnek.

Az SPI kommunikáció inicializálását az SPI\_MasterInit() függvény végzi. Itt megtörténik a MOSI, SCK, ill. a két  $\overline{CS}$  kimenet konfigurálása, a Master üzemmód beállítása és az SPI kommunikáció engedélyezése.

A potméterek vezérlése során először az adott eszköz  $\overline{CS}$  jelének aktívra állítása történik, ezt követi az adatok kiküldése, melyet az SPI\_MasterTransmit() függvény valósít meg. Először szükséges egy utasítás byte elküldése (0x00). Ennek első 4 bitje a potméter állását tartalmazó cím, melyet a Write utasítás két bitje, illetve két nem használt adatbit követ [16]. Ezt egy adatbajt követi, melynek értéke a felhasználó által beállított potméter állásnak felel meg. A  $\overline{CS}$  jel megszüntetése után a fenti folyamat megismételhető a másik potméter beállításának érdekében. Mindez csak akkor történik meg, ha a felhasználó megváltoztatja egy potméter állását, vagyis ha az adott potméterhez rendelt érték és az újonnan beolvasott érték különbsége nagyobb az ADC\_TOLERANCE makróban megadott számnál.

## 6 Huzalozási rajz elkészítése

A nyomtatott áramkör tervezéséhez az ingyenesen elérhető Cadsoft EAGLE szoftvert választottam, melyhez népszerűsége miatt a tervezéshez szükséges alkatrészkönyvtárak mellett számos oktatóanyag is rendelkezésre áll. A szoftver használatának, illetve a nyáktervezés alapjainak elsajátításában elsősorban a Sparkfun.com anyagai nyújtottak segítséget [17] [18] [19]

A tervezés első lépéseként kiválasztottam egy megfelelő méretű, gitáreffektek esetén gyakran használt műszerdobozt és meghatároztam a csatlakozók, potméterek és kapcsolók elhelyezkedését. Ez alapján már adott volt a nyomtatott áramkör maximális mérete, illetve az említett alkatrészek helye a huzalozási rajzon. A többi alkatrész elhelyezése során ügyeltem a logikus elrendezésre: az összetartozó alkatrészek egymáshoz közel, blokkokba rendezve helyezkednek el. Fontos szempont volt az analóg és digitális alkatrészek, ill. földek elkülönítése is. A digitális vezérlést megvalósító alkatrészek a panel tetején (a táp mellett, de attól elkülönülve) kaptak helyet. A mikrokontrollerhez kapcsolódó hidegítő kondenzátorok és a kristály oszcillátor elhelyezése során figyelembe vettem a gyártó által megadott tervezési szempontokat [20]:

- A hidegítő kondenzátorokat a Vcc – GND lábpárokhoz közel helyeztem el
- AVCC és VCC tápok elválasztása az L1 ferrittel
- A 16MHz-es kristály a mikrokontrollerhez közel helyezkedik el, földdel kitöltött területtel körülvéve

Az elkészített huzalozási rajz a Függelék 9-3. ábráján látható. A 9-4. és 9-5. ábrákon a felső, illetve alsó réteg egymástól elkülönítve látható.



## 7 Mérések, értékelés

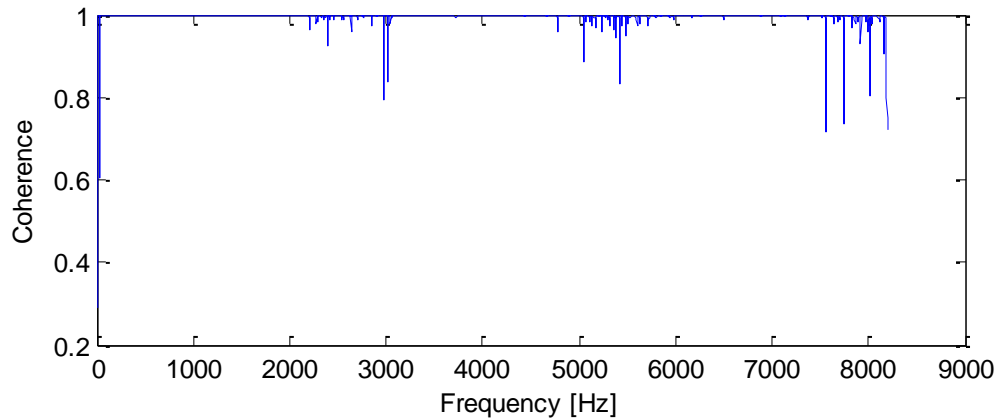
### 7.1 Mérési elrendezés

A megépített delay effekt átvitelét a National Instruments cég cRio mérőhardvere, illetve a NiHu Matlab toolbox használatával végeztem. A breadboard paneleken megépített áramkör miatt (egy jól megtervezett nyomtatott áramkörhöz képest) zajosabb lehet a mérés, mégis jól értékelhető eredményeket kaptam.

A mérések során sweep szinusz gerjesztést használtam, amely minden esetben az áramkör bemenetére csatlakozott. A mért jel vagy az effekt kimenete - azaz a tiszta és az effektezett jelek együtt - vagy csak az effektezett jel volt (a Mix potméter 3. lábán mérve). A méréseket a potméterek minimális, közép és maximális állásában is elvégeztem, kivéve a Feedback paraméter esetében, ugyanis egy bizonyos pont felett gerjedni kezd az effekt, ami zenei szituációkban érdekes lehet, azonban nem vezet jól értékelhető mérési eredményekhez. A különböző beállításokat az ábrák alatt a paraméterek kezdőbetűivel és egy 0 és 10 közé eső számmal jelöltem, ahol a 0 minimális, a 10 maximális állást jelent.

Mivel a vizsgált eszköz átvitele a korábban tárgyalt okok miatt erőteljesen aluláteresztő, az alkalmazott mintavételi frekvencia  $f_s = 17.067 \text{ kHz}$  volt. Minden mérés négy egymást követő gerjesztésből áll, melyek eredményét a szoftver átlagolja, illetve a mérés bizonytalanságairól is visszajelzést kapunk egy koherencia függvény formájában. Ennek a függvénynek az értéke 1 azokon a helyeken, ahol a négy mérés eredménye konzisztens, azaz minden mérés ugyanazt az eredményt adja. Ahol ez a mérés zajossága vagy az áramköri nemlinearitások miatt nem teljesül, ott a koherencia függvény értéke egynél kisebb.

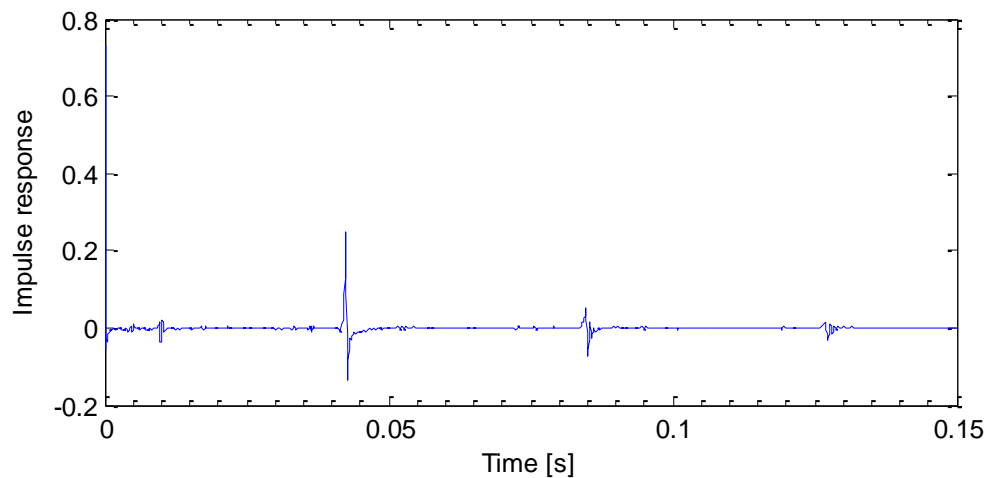
## 7.2 Eredmények



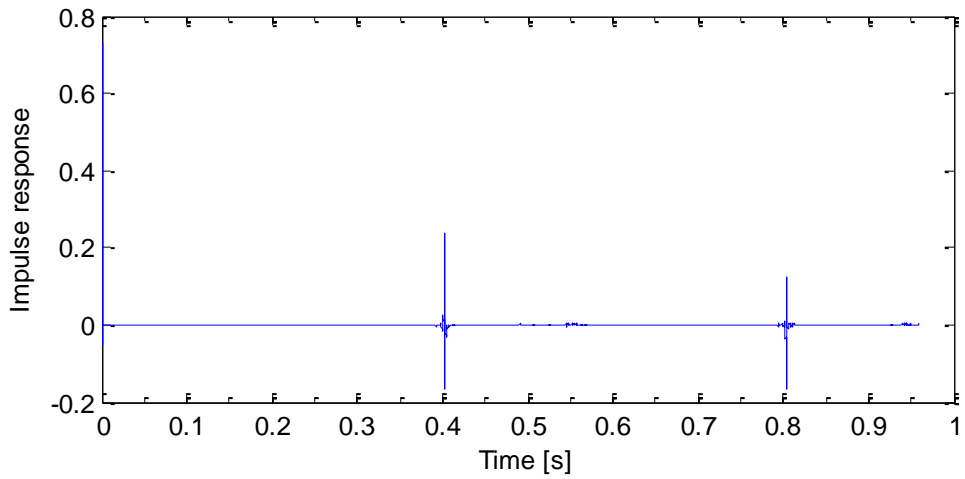
7-1. ábra Koherencia függvény (t5-f4-m10)

A breadboard paneleken történő megvalósítás és a szükségesnél hosszabb vezetékek használata zaj szempontjából nem ideális, erre utal, hogy a koherencia függvény helyenként egynél kisebb értékeket vesz fel (7-1. ábra). A mérési eredmények feldolgozása során a kapott impulzusválasz elején és végén megjelenő zajokat manuálisan eltávolítottam, majd ezt felhasználva ábrázoltam a frekvenciamenetet a Matlab `fft()` függvényének segítségével.

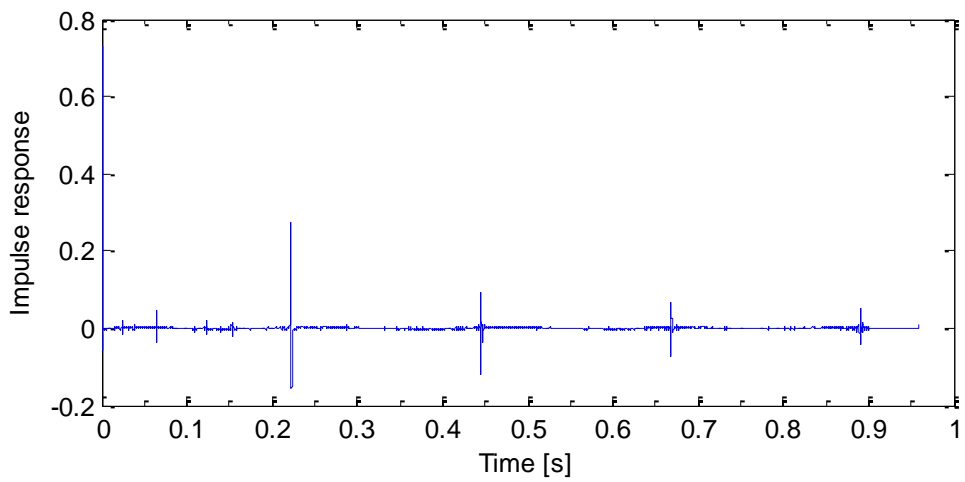
A 7-2. - 7-4. ábrákon megfigyelhető, hogy a különböző beállítások mellett a késleltetési idő az elvártaknak megfelelő: a Time potméter minimum állásában kb.  $2 \cdot 20.5$  ms, maximum állásban  $2 \cdot 205$  ms. Emellett megfigyelhető a visszacsatolás által okozott fokozatos lecsengés.



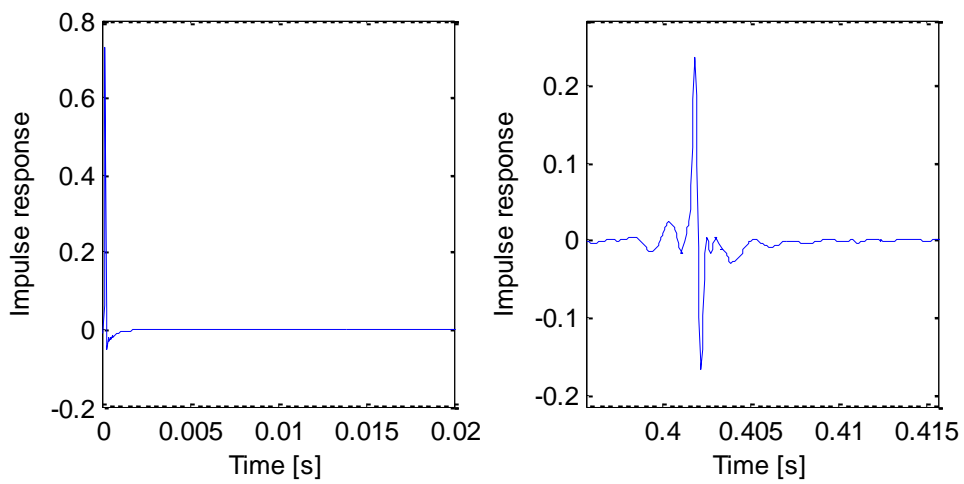
7-2. ábra Az effekt impulzusválasza (t0-f5-m10)



7-3. ábra Az effekt impulzusválasza (t10-f5-m10)



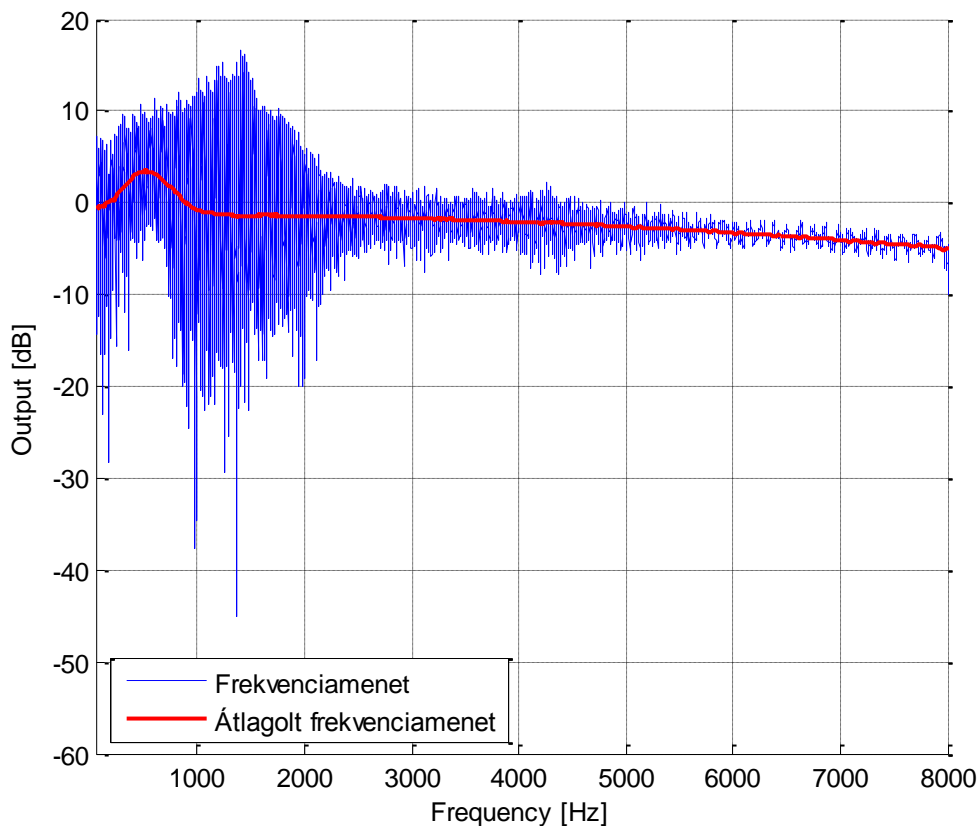
7-4. ábra Az effekt impulzusválasza (t5-f5-m10)



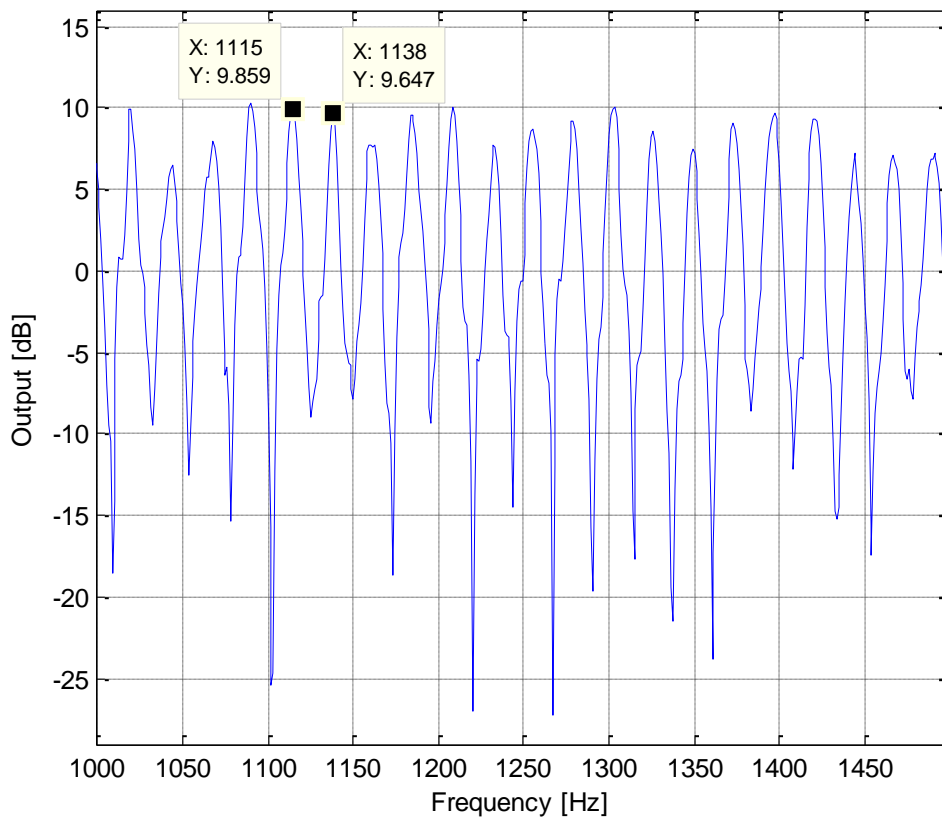
7-5. ábra Eredeti és késleltetett impulzus (t10-f5-m10)

Az eredeti és az első késleltetett impulzust azonosan skálázott időtengelyekkel ábrázolva jól látható, hogy az impulzus jelentősen kiszélesedik (7-5. ábra). Ez az erőteljes aluláteresztő szűrésnek köszönhető, ugyanis nagyfrekvenciás komponensek hiányában csökken a jel fel- ill. lefutási meredeksége.

A 7-6. ábrán a kimeneten mért frekvenciamenet látható. Itt körülbelül 2 kHz-ig erőteljes kiemelések és elnyomások jelentkeznek, melyet a késleltetett és az eredeti audio jel összegzése okoz (fésűszűrő-hatás). Bár az ábrázolás felbontása és a leszívások kis távolsága miatt az említett hatás itt nem válik jól láthatóvá, az ábrát kinagyítva egyértelműen megfigyelhető a jelenség (7-7. ábra). A 7-6. ábrán viszont jól látható, hogy a fésűszűrő-hatás 2000 Hz fölött jelentősen csökken, ebből arra következtethetünk, hogy a késleltetett jelen alkalmazott aluláteresztő szűrők vágási frekvenciája az említett frekvencia körül van. Ez megfelel a 4.2.2 fejezetben elvégzett számításnak, mely alapján a harmadfokú Sallen-Key szűrők vágási frekvenciái 2139, ill. 2378 Hz. A frekvenciamenetet mozgó átlaggal szűrve a kiemelések és elnyomások ellenére láthatóvá válik az áramkör átlagos frekvenciamenete.



7-6. ábra Az effekt frekvenciamenete (t5-f5-m10)



7-7. ábra Az effekt frekvenciamenete (t0-f5-m10)

Az eredményt egy szűkebb frekvenciatartományban ábrázolva, az x tengelyt lineárisan skálázva láthatók az egyenlő távolságokra lévő kiemelések és leszívások. A minimum- ill. maximumhelyek távolságai az alábbi képlettel számíthatók:

$$\Delta f = \frac{1}{t_D}$$

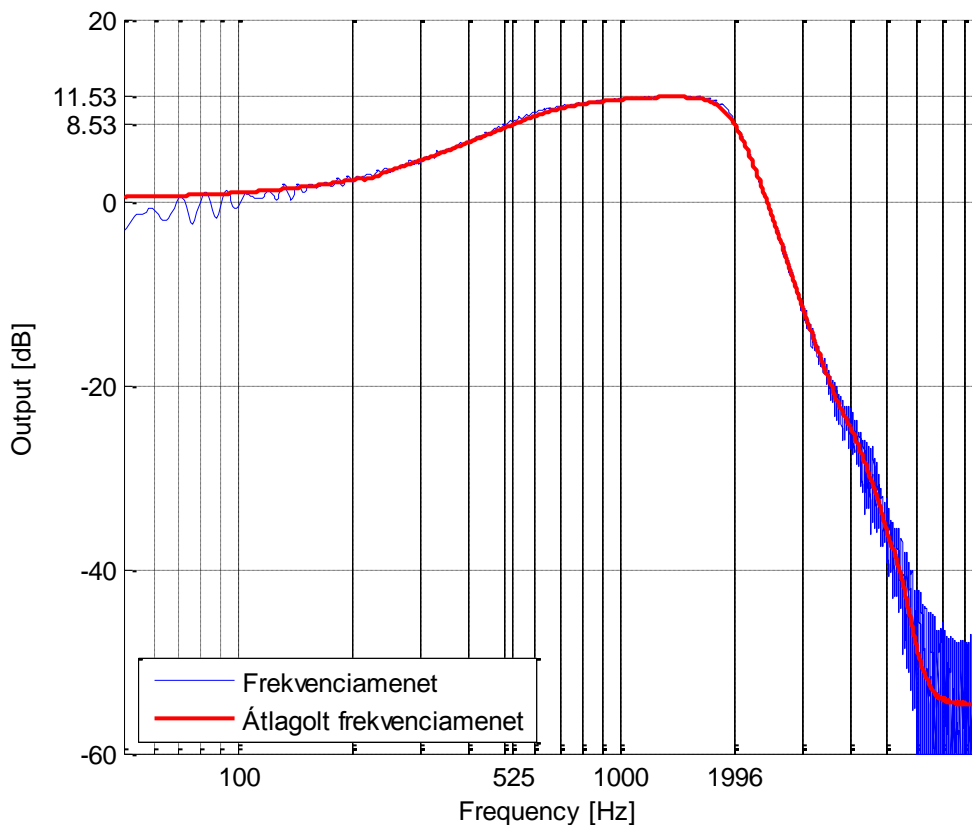
ahol  $t_D$  a késleltetési idő. Ez alapján a leszívások távolsága minimális késleltetést beállítva

$$\Delta f = \frac{1}{0.0425s} = 23.36 \text{ Hz}$$

A  $\Delta f$  frekvenciát grafikusán meghatározva is hasonló eredményt kapunk, ezt szemlélteti a 7-7. ábra.

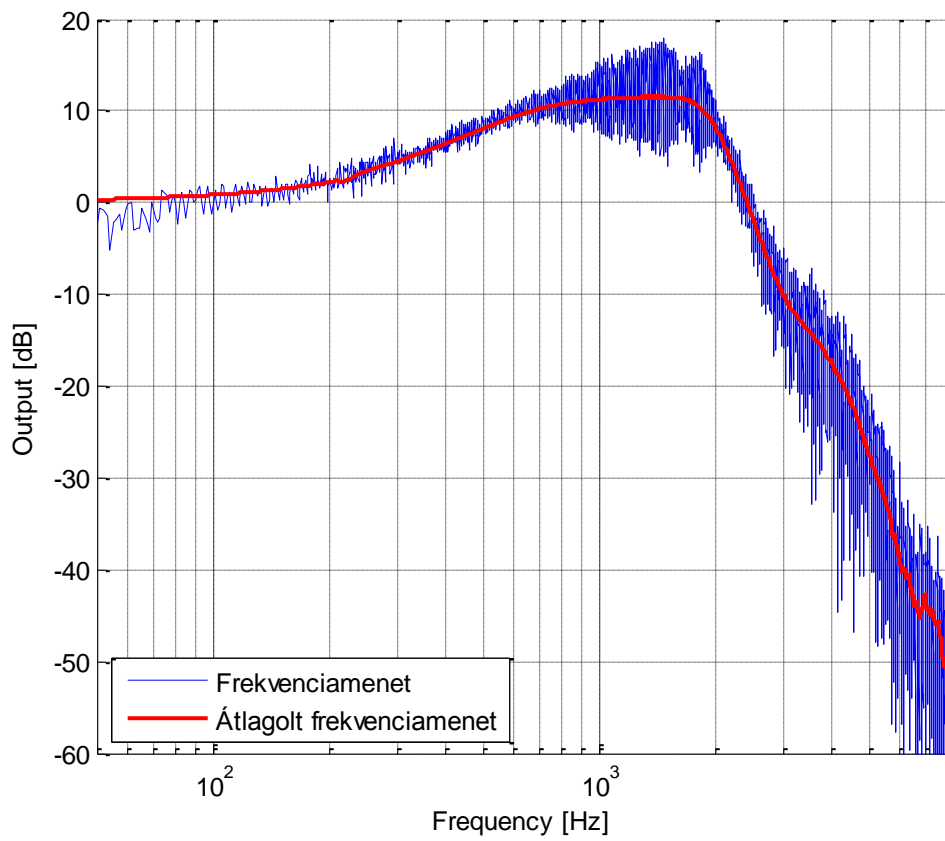
Természetesen az említett fésűszűrő-hatás minden késleltető alapú effektúknél jelentkezik, függetlenül attól, hogy analóg vagy digitális megvalósításról van szó. Emellett érdemes megvizsgálni az analóg megoldás jellegzetességeit, melyek a BBD chipek, az erős teljes aluláteresztő szűrés és a kompander áramkörök együttes hatásából adódnak. Ezt szemlélteti a 7-8. ábra, mely alapján észrevehető, hogy a BBD-ket tartalmazó hurok erőteljesen aluláteresztő: a -3dB-s pont 1996 Hz-nél található.

Ez megfelel az elvárásoknak, ugyanis az analóg delay pedálokban alkalmazott szűrők töréspontja jellemzően a minimális órajel frekvencia (6-10 kHz) [10] egyharmadához közeli érték [8]. Emellett a visszacsatolás miatt a tiszta jellel való összegzés nélkül is megfigyelhető a már említett fésűszűrő-hatás (7-9. ábra).



7-

7-8. ábra A BBD hurok frekvenciamenete visszacsatolás nélkül(t5-f0-m10)



7-9. ábra A visszacsatolt BBD hurok frekvenciamenete (t5-f4-m10)

## 8 Összefoglalás

Bár a megépített effekt az elvárásoknak megfelelően működik, sok hasznos, a digitális vezérlés előnyeit kihasználó funkció nem került megvalósításra a félév során. Egy ilyen funkció a "tap tempo", azaz a késleltetési idő beállítása lábkapcsoló segítségével, a gombnyomások között eltelt idő alapján. Szintén jelentős fejlesztés lenne a már említett moduláció megvalósítása. Mikrokontroller segítségével a szokásos négyszög és háromszög jelalakok mellett egyéb tetszőleges jelalakok is használhatók moduláló jelként, illetve a moduláció az órajel mellett egyéb paraméterekre is kiterjeszhető, mindez az alkatrészek számának növekedése nélkül. A digitális vezérlés lehetőséget nyújt presetek kezelésére, azaz a felhasználó által létrehozott beállítások elmentésére és betöltésére. Így gyakorlatilag tetszőleges számú előre beállított hangzás állhat rendelkezésre, melyek például nyomógombok vagy egy többállású kapcsoló segítségével kiválaszthatók. A mikrokontroller emellett alkalmas lehet szabványos MIDI üzenetek fogadására és feldolgozására is. Ez lehetőséget nyújt például presetek betöltésére vagy a zenei tempó beállítására akár több eszközön egyidejűleg egy külső MIDI kontroller használatával.

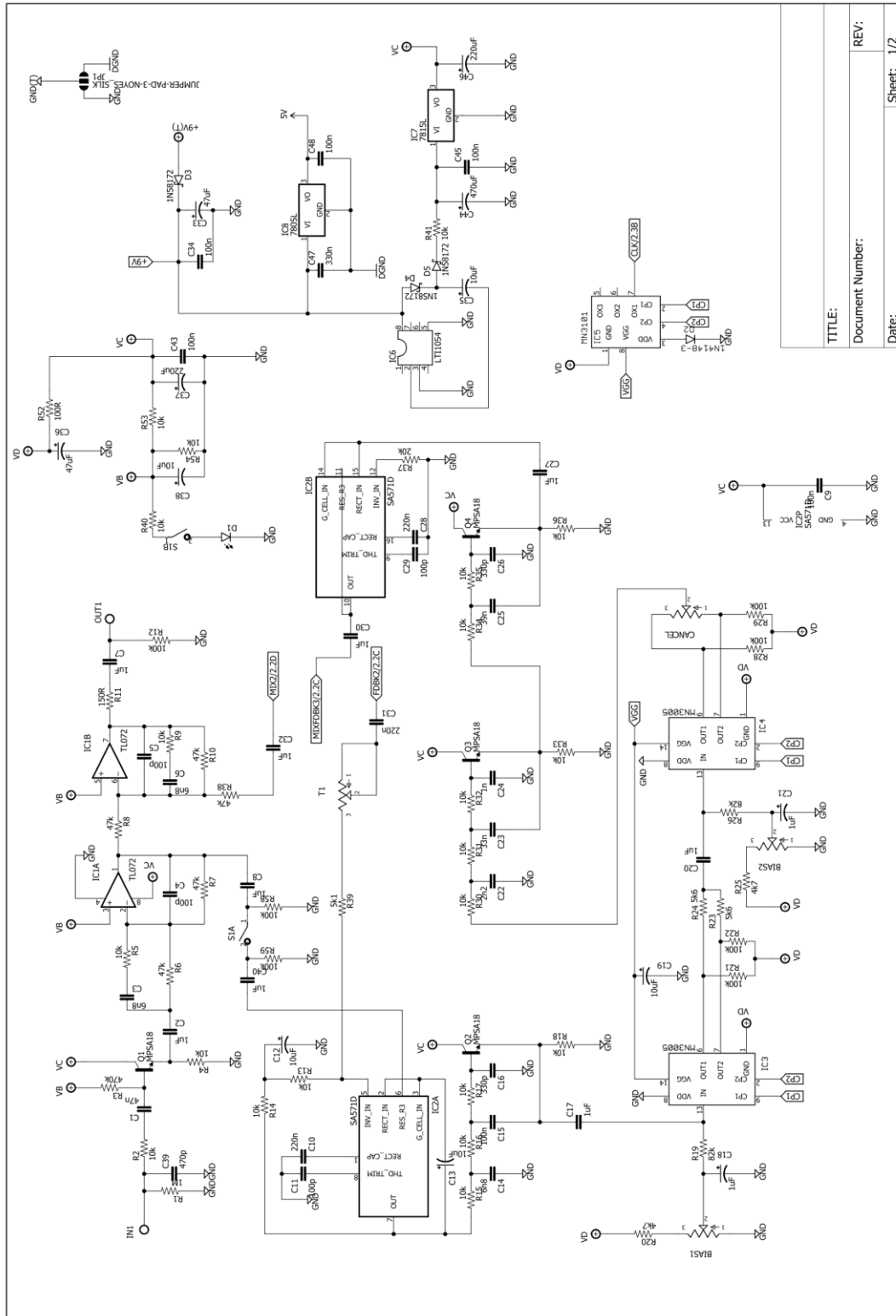
Az ilyen módon megvalósított digitális vezérlés minimális módosításokkal más BBD alapú effektek (chorus, flanger, vibrato) vezérlésére is alkalmas lehet, illetve további egyszerűsítéseket alkalmazva bármilyen analóg effekt esetén biztosíthatja a fent említett előnyöket és funkciókat.



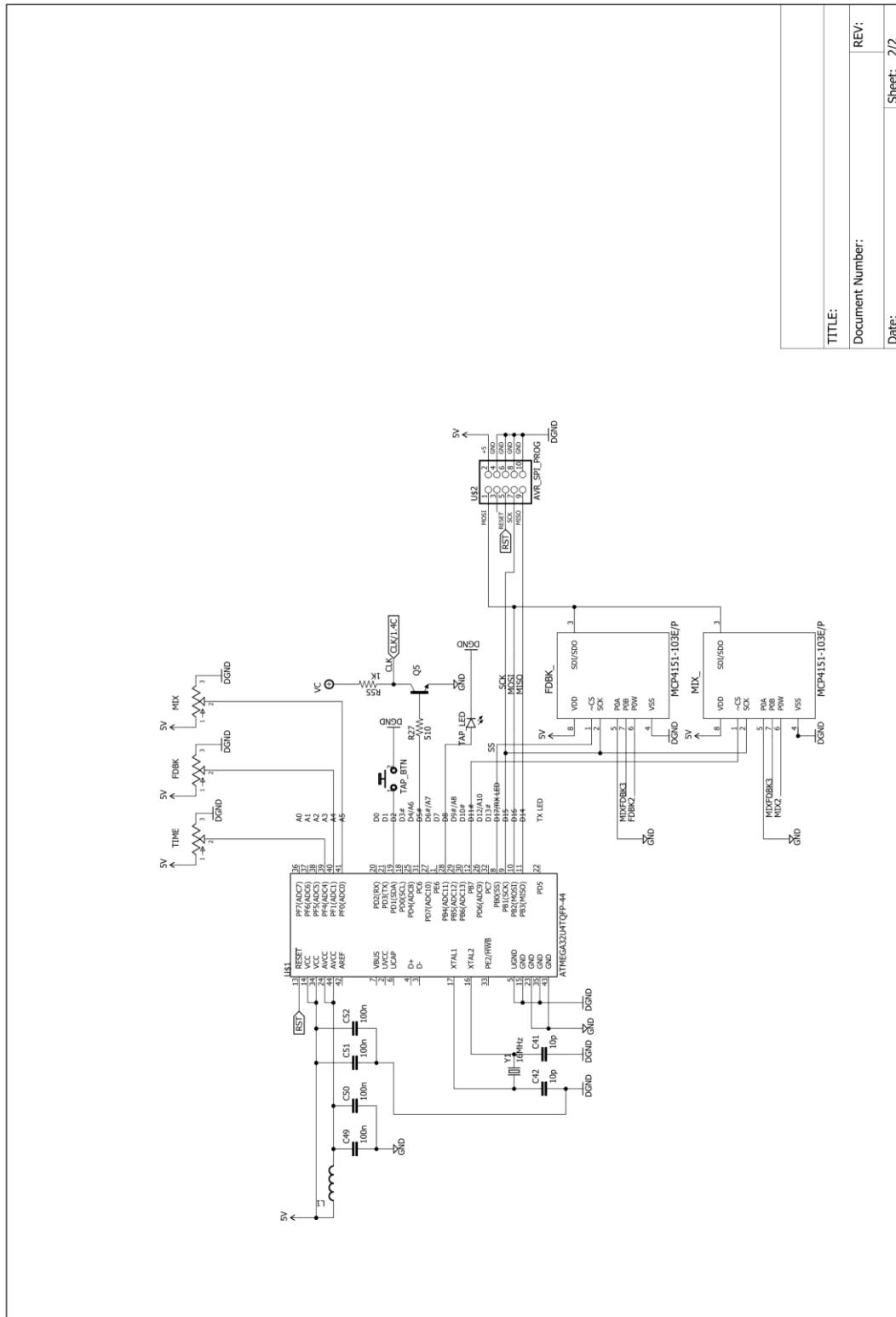
## 9 Irodalomjegyzék

- [1] F. L. J. Sangster, „Integrated bucket-brigade delay line using MOS tetrodes,” *Phillips Technical Review Volume 31*, 1970.
- [2] Electrosplash.com, „Bucket Brigade Devices: MN3007”.
- [3] G. P. Weckler, *Bucket Brigade Devices Circa 1977*, 1977.
- [4] EC&G Reticon, „AD-512D Sampled Analog Delay Line”.
- [5] Radio Shack (Archer), „SAD-1024 Dual Analog Delay Line”.
- [6] U. Zölzer, *DAFX - Digital Audio Effects*, 2002.
- [7] Panasonic Company, *BBD Series Datasheet*.
- [8] J. S. Colin Raffel, *Practical Modeling Of Bucket-Brigade Device Circuits*, 2010.
- [9] ON Semiconductor, „SA571 Datasheet,” 2006.
- [10] Madbeanpedals.com, *Aquaboy Deluxe*, 2015.
- [11] STMicroelectronics, *L7800 Series Datasheet*, 2006.
- [12] ON Semiconductor, „Applications for Compandors NE570/SA571,” 2005.
- [13] MoltenVoltage.com, *PedalSync™ BBD Delay Clock chip MV-64 Datasheet*, 2013.
- [14] Wikipedia, *Red Pitaya (hardware)*, 2016.
- [15] Atmel Corporation, *ATmega16U4 / ATmega32U4 Datasheet*, 2015.
- [16] Microchip Technology Inc., *MCP413X/415X/423X/425X Datasheet*.
- [17] Sparkfun.com, *Using EAGLE: Schematic*.
- [18] Sparkfun.com, *Using EAGLE: Board Layout*.
- [19] Sparkfun.com, *SparkFun According to Pete #34: PCB Layout*.
- [20] Atmel Corporation, *AVR042: AVR Hardware Design Considerations*, 2016.

# Függelék



9-1. ábra Az analóg effekt kapcsolási rajza



TITLE:	
Document Number:	
Date:	
REV:	
Sheet:	2/2

9-2. ábra A digitális vezérlés kapcsolási rajza

## BBD.h

```
#ifndef BBD_H_
#define BBD_H_

#include <stdlib.h>
#include <avr/io.h>
#include <avr/interrupt.h>

#include "print.h"
#include "usb_debug_only.h"

#define CPU_PRESCALE(n)  (CLKPR = 0x80, CLKPR = (n))

// DEFINE BOOL
#define bool_t  uint8_t
#define true  1
#define false 0

// DEFINE CLOCK / TIMER PARAMETERS
#ifndef F_CPU
#define F_CPU 16000000UL // 16 MHz clock speed
#endif
#define TIMER1_PRE_SCALER 64.0
#define TIMER1_FREQUENCY 0.001 // IN ms

// TEMPO PARAMETERS
#define MIN_TEMPO_MS 204.8*2.0 // IN MS ->40bpm
#define MAX_TEMPO_MS 20.48*2.0 // IN MS ->240bpm

// DEBOUNCE PARAMETERS
#define CHECK_MSEC 1 // READ HARDWARE EVERY X MSEC
#define PRESS_MSEC 10 // STABLE TIME BEFORE REGISTERING PRESSED
#define RELEASE_MSEC 20 // STABLE TIME BEFORE REGISTERING RELEASED

// ADC PARAMETERS
#define ADC_TOLERANCE 5 // 0..255

// DEFINE I/O PINS
#define TAP_BTN_PIN PD1
#define TEMPO_LED_PIN PB4 // OCR0A
#define FDBK_SS_PIN PB0
#define MIX_SS_PIN PB7

// GLOBAL VARIABLES
extern bool_t tapButtonState;
extern bool_t pressed;
extern bool_t changed;

extern uint16_t tempoCounter;
extern uint16_t blinkCounter;
extern uint16_t holdCounter;
extern uint16_t currentTempo;

extern uint8_t adcChannel;
extern uint16_t sampleValue; // adc sample

extern uint16_t timePotValue;
extern uint8_t fdbkPotValue;
```

```
extern uint8_t mixPotValue;

// FUNCTION PROTOTYPES
void io_init(void);
void adc_init(void);
void timer1_init(void);
void timer3_init(void);

void SPI_MasterInit(void);
void SPI_MasterTransmit(char cData);
uint16_t adcRead(uint8_t ch);

bool_t getButtonState(void);
void debounceButton(bool_t* changed, bool_t* pressed);
void setTempo(void);

#endif /* BBD_H_ */
```

## BBD\_main.c

```
#include "BBD.h"

int main(void)
{
    // set for 16 MHz clock
    CPU_PRESCALE(0x00);

    io_init();
    adc_init();
    timer1_init();    // tempo
    timer3_init();    // BBD Clock
    SPI_MasterInit();

    sei();

    while (1) {
    if (adcChannel <= 7)
    {
        sampleValue = 255.0 - adcRead(adcChannel)/4;
        switch (adcChannel)
        {
            case 0:    // TIME
                ;
                if ( abs((int)timePotValue - (int)sampleValue) > ADC_TOLERANCE) {
                    timePotValue = sampleValue;
                    currentTempo = (float)MAX_TEMPO_MS + timePotValue / 256.0 *
((float)MIN_TEMPO_MS - (float)MAX_TEMPO_MS);
                }

                break;

            case 1:    // FEEDBACK
                ;
                bool_t buttonChanged;
                bool_t buttonPressed;
                debounceButton(&buttonChanged, &buttonPressed);

                if (buttonPressed) {
                if ( abs((int)fdbkPotValue - (int)sampleValue) > ADC_TOLERANCE) {
                    fdbkPotValue = sampleValue;
                    PORTB &= ~(1<<FDBK_SS_PIN);
                    SPI_MasterTransmit(fdbkPotValue);
                    PORTB |= (1<<FDBK_SS_PIN);
                    print("  fdbk\n");
                    phex(fdbkPotValue);
                }
                }

                break;

            case 3:
                break;

            case 4:    // MIX
                ;
                if ( abs((int)mixPotValue - (int)sampleValue) > ADC_TOLERANCE) {
                    mixPotValue = sampleValue;
                }
            }
        }
    }
}
```

```

        PORTB &= ~(1<<MIX_SS_PIN);
        SPI_MasterTransmit(mixPotValue);
        PORTB |= (1<<MIX_SS_PIN);
        print("  mix\n");
        phex(mixPotValue);
    }
    break;

case 5: // RATE
    break;

case 6: // DEPTH
    break;

case 7: // SHAPE
    ;
    break;
}

    adcChannel++;
}
else adcChannel = 0;
}
}

```

## BBD.c

```
#include "BBD.h"

bool_t tapButtonState;
bool_t pressed = false;
bool_t changed = false;

uint16_t tempoCounter = 0;
uint16_t blinkCounter = 0;
uint16_t holdCounter = 0;
uint16_t currentTempo = 500;

uint8_t adcChannel = 0;
uint16_t sampleValue = 0; // adc sample

uint16_t timePotValue;
uint8_t fdbkPotValue;
uint8_t mixPotValue;

void io_init(void) {
    DDRD = 0xFF; // Makes PORTD Output
    DDRF = 0x00; // PORTF: INPUT
    PORTF = 0x00; // DISABLE PULLUP R ON PORTF
    PORTD |= (1<<TAP_BTN_PIN); // ENABLE INPUT PULLUP FOR BTN
    DDRB |= (1<<TEMPO_LED_PIN); // OCR0A: TEMPO LED PWM
    DDRC |= (1<<PC6); // OC3A
}

void adc_init(void) {
    ADMUX = (1<<REFS0); // VOLTAGE REFERENCE:AVCC (cap on AREF pin)
    ADCSRA |= (1<<ADEN) | (1<<ADPS2) | (1<<ADPS1) | (1<<ADPS0); // SET
    PRESCALER: 128
}

void timer1_init(void) {
    OCR1A = ((F_CPU / TIMER1_PRE_SCALER) * TIMER1_FREQUENCY) - 1; // Trigger
    value.
    TCNT1 = 0x00; // Starting at 0.
    TCCR1B |= (1 << WGM12); // TURN ON CTC
    TCCR1B |= (1 << CS10) | (1 << CS11); // PRESCALER: 64
    TIMSK1 |= (1 << OCIE1A); // ENABLE TIMER COMPARE INTERRUPT
}

void timer3_init(void) {
    TCCR3B = (1<<WGM32); //CTC MODE
    TCCR3B |= (1<<CS30); // no PRESCALER
    TCCR3A = (1<<COM3A0); // CTC toggle COMPARE
    TCNT3 = 0;
    OCR3A = 80;
}

void SPI_MasterInit(void)
{
    DDRB |= (1<<PB2)|(1<<PB1)|(1<<FDBK_SS_PIN)|(1<<MIX_SS_PIN);
    SPCR = (1<<SPE)|(1<<MSTR)|(1<<SPR0);
}

void SPI_MasterTransmit(char cData)
```



```

{
    SPDR = 0x00;    // command byte
    while(!(SPSR & (1<<SPIF)));
    SPDR = cData;  // data
    while(!(SPSR & (1<<SPIF)));
}

uint16_t adcRead(uint8_t ch)
{
    // select the corresponding channel 0~7
    // ANDing with '7' will always keep the value
    // of 'ch' between 0 and 7
    ch &= 0b0000111; // AND operation with 7
    ADMUX = (ADMUX & 0xF8)|ch; // clears the bottom 3 bits before ORing

    // start single conversion
    // write '1' to ADSC
    ADCSRA |= (1<<ADSC);

    // wait for conversion to complete
    // ADSC becomes '0' again
    // till then, run loop continuously
    while(ADCSRA & (1<<ADSC));
    return (ADC);
}

bool_t getButtonState(void) {
    bool_t state = false;

    if (PIND & (1<<TAP_BTN_PIN)) {
        state = true;
    }

    return state;
}

void debounceButton(bool_t *changed, bool_t *pressed) {
    // http://www.ganssle.com/debouncing-pt2.htm

    static uint8_t count = RELEASE_MSEC / CHECK_MSEC;
    bool_t state;
    *changed = false;
    *pressed = tapButtonState;
    state = getButtonState();

    if (state == tapButtonState) {

        if (tapButtonState)
            count = RELEASE_MSEC / CHECK_MSEC;
        else
            count = PRESS_MSEC / CHECK_MSEC;
        } else {

        if (--count == 0) {
            tapButtonState = state;
            *changed = true;
            *pressed = tapButtonState;
            if (tapButtonState) {
                count = RELEASE_MSEC / CHECK_MSEC;
            }
        }
    }
}

```

```

    }
    else {
        count = PRESS_MSEC / CHECK_MSEC;
    }
}
}
}

void setTempo(void) {
    float avgTempo;

    if (tempoCounter < MAX_TEMPO_MS) {
        currentTempo = MAX_TEMPO_MS;
    }

    else if (tempoCounter < MIN_TEMPO_MS) {
        avgTempo = (currentTempo + tempoCounter) / 2.0;
        currentTempo = avgTempo;
    }

    tempoCounter = 0;
}

ISR(TIMER1_COMPA_vect) {
    bool_t buttonChanged, buttonPressed;
    tempoCounter++;

    debounceButton(&buttonChanged, &buttonPressed);

    // blink tempo led
    if (blinkCounter++ >= currentTempo || (!buttonPressed &&
buttonChanged) ) {
        blinkCounter = 0;
        PORTB |= (1<<TEMPO_LED_PIN);
    }
    if (blinkCounter == 10) {PORTB &= ~(1<<TEMPO_LED_PIN); }

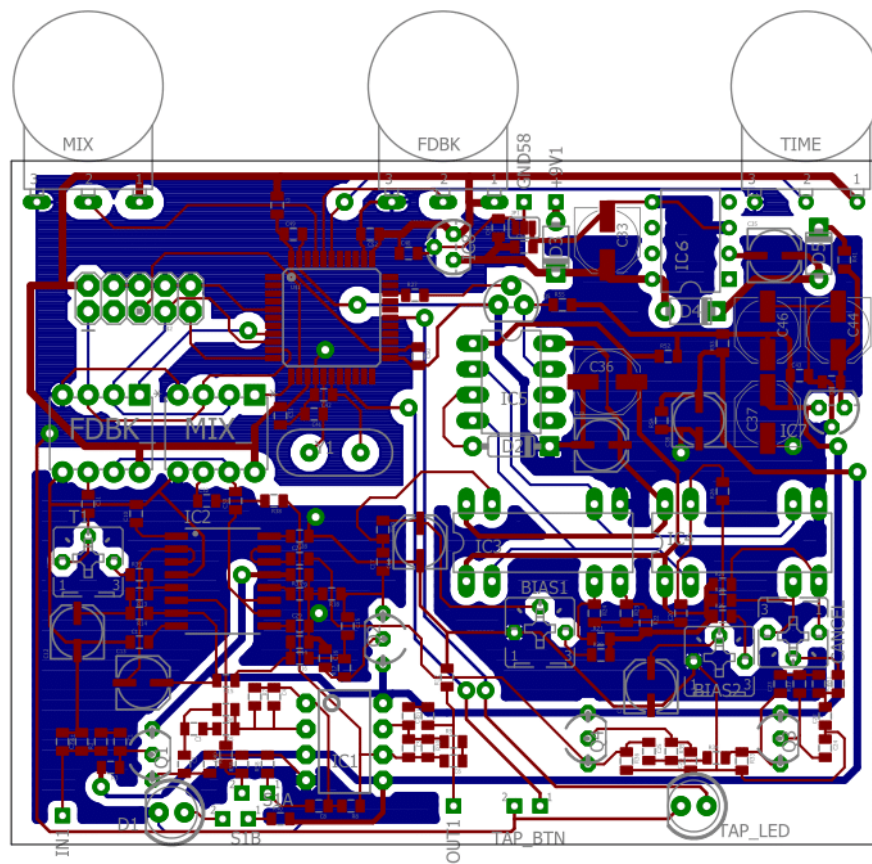
    // tap
    if (buttonChanged && !buttonPressed) { // PRESSED
        setTempo();
    }

    if (buttonChanged && buttonPressed) { // RELEASED
        holdCounter = 0;
        PORTB &= ~(1<<FDBK_SS_PIN);
        SPI_MasterTransmit(fdbkPotValue);
        PORTB |= (1<<FDBK_SS_PIN);
    }

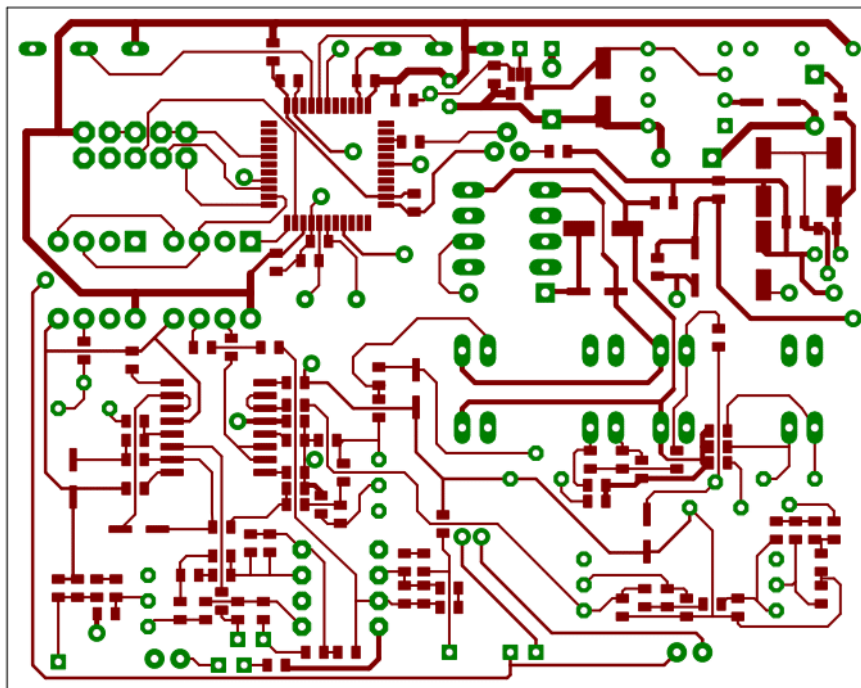
    // hold
    if (!buttonPressed && (holdCounter++ >= 300) ) {
        holdCounter = 0;
        PORTB &= ~(1<<FDBK_SS_PIN);
        SPI_MasterTransmit(0xFF);
        PORTB |= (1<<FDBK_SS_PIN);
    }

    OCR3A = ((currentTempo - MAX_TEMPO_MS) / (MIN_TEMPO_MS - MAX_TEMPO_MS)
* 360.0) + 39.0;
}

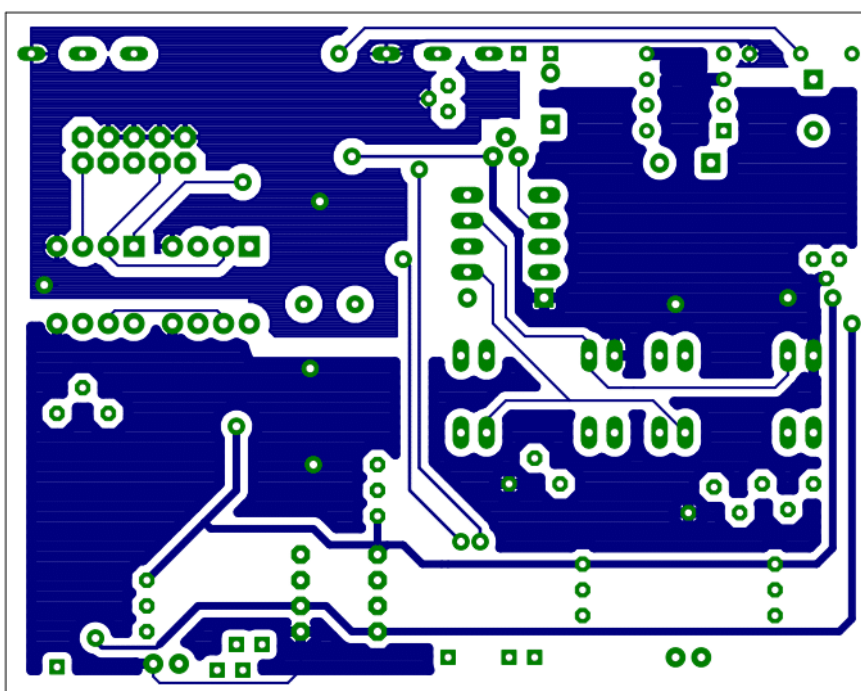
```



9 3. ábra Huzalozási rajz



9-4. ábra A huzalozási rajz felső rétege



9-5. ábra A huzalozási rajz alsó rétege