

平成 21 年度卒業論文

# 自動演奏ギターの高性能化

福岡工業大学工学部

知能機械工学科

06E1043 弘中佑希

指導教員 河村良行

## 目次

第1章 緒言-----	1
1.1 研究背景	
1.2 研究目的	
第2章 装置概要-----	3
第3章 ハードウェアの改良-----	6
3.1 ステッピングモータの改良	
3.1.1 ストローク制御用モータ	
3.1.2 ピック深さ制御用モータ	
3.2 可動部の軽量化	
3.2.1 モータ固定具	
3.2.2 ピック固定具	
3.3 スイッチング回路の改良	
3.3.1 装置概要	
第4章 制御プログラム-----	13
4.1 改良点	
4.2 動作テストプログラム	
第5章 音量変化の実験及び考察-----	22
5.1 ピック側	
5.1.1 強弱のつけ方	
5.1.2 解析方法	
5.1.3 実験方法	
5.1.4 実験結果及び考察	
5.2 ソレノイド側	
5.2.1 強弱のつけ方	
5.2.2 実験方法	
5.2.3 実験結果及び考察	
第6章 結言-----	30
6.1 結言	
6.2 今後の課題	
6.3 謝辞	
参考文献-----	32
付録-----	34

# 第 1 章

## 緒言

## 1.1 研究背景

近年、産業用ロボットだけでなく、人々楽しませたり、癒しを与えたりするアミューズメントロボットが注目されてきている。その中で自動演奏楽器はピアノを始め、弦楽器ではギターや沖縄三味線など開発されて進歩してきている。

## 1.2 研究目的

本研究の今年度の目的は、昨年開発された自動演奏ギターを高性能にすることが目的である。例えば、動作音の軽減、音に強弱をつけての演奏、ミュート（弦に軽く触れ音を短く切る奏法）などである。

## 第 2 章 装置概要

まず、装置全体図を図 2.1 に、システム図を図 2.2 に示す。次に弦を押さえる機構は図 2.3 に示すソレノイドを用いる。制御方法は 32ch デジタル入出力ボード (PCI-2703A) をプログラムにより出力させ、回路を通しスイッチングさせる。次にストロークさせる機構は3つのステッピングモータを用いる。まず図 2.4 に示す2つのモータで左右の制御、図 2.5 のモータで上下(ピック入れ込み深さ)の制御を行う。こちらもプログラムによりパルスモータ制御ボード (LPC-742020) からパルスを出力させ制御している。

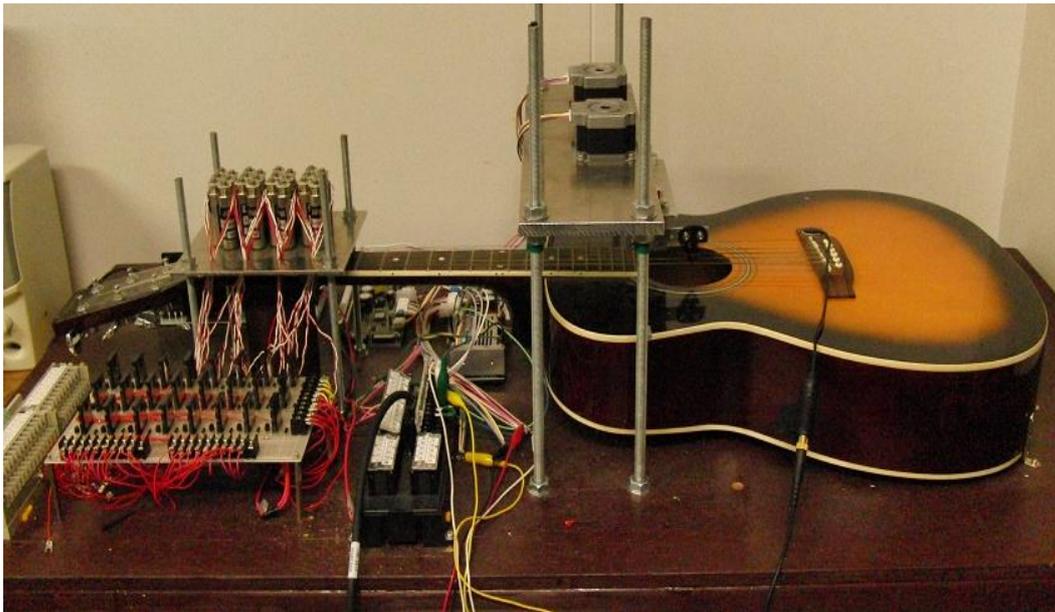


図 2.1 自動演奏ギター (装置全体図)

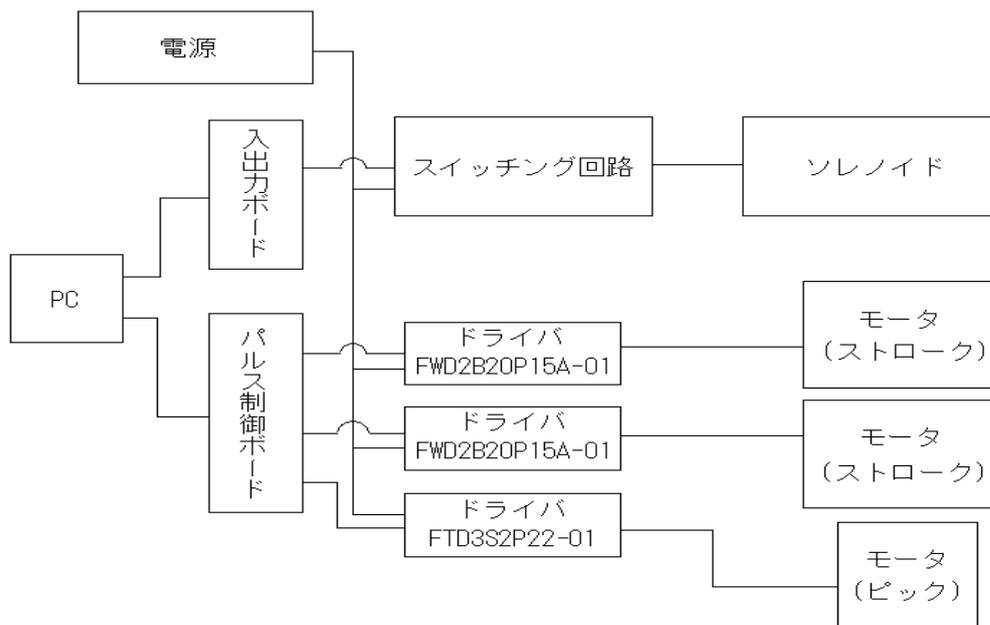


図. 2.2 自動演奏ギター (システム図)

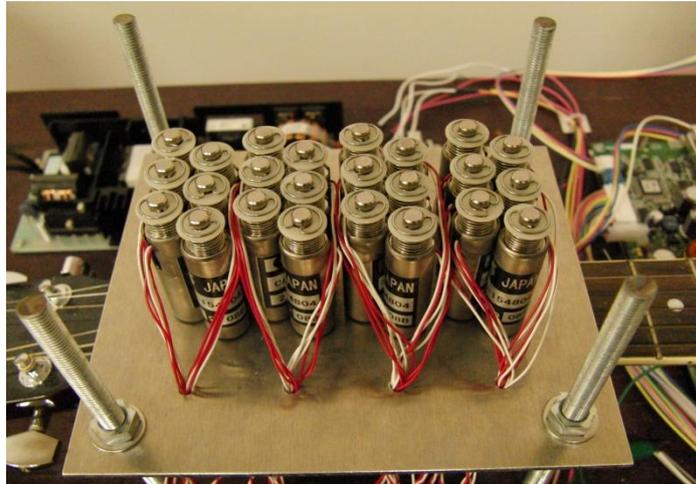


図 2.3 ソレノイド

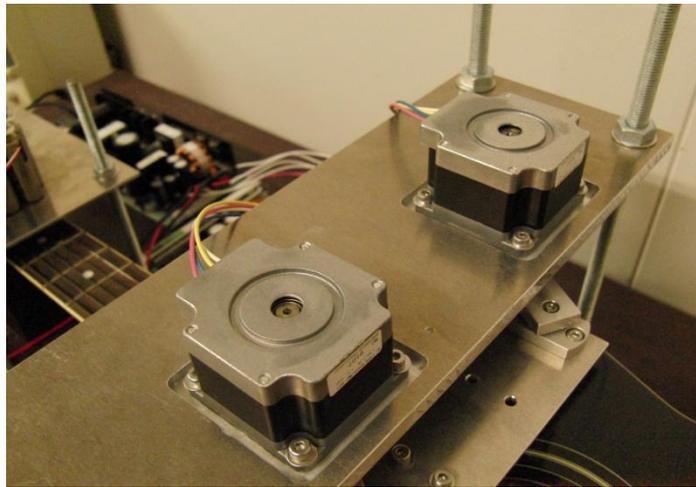


図 2.4 ステッピングモータ (ストローク制御)

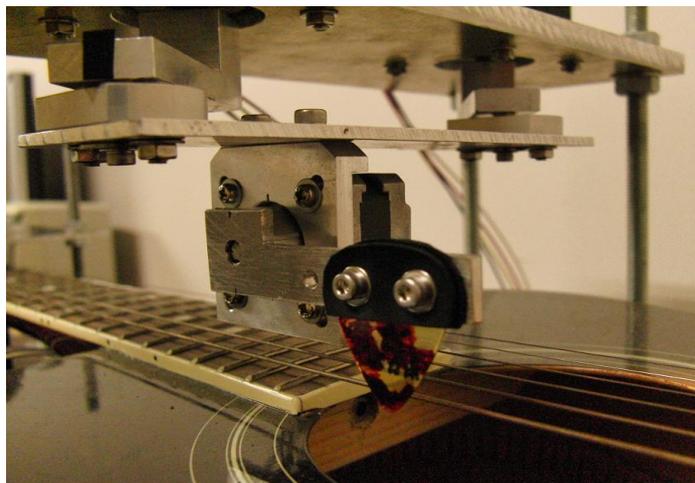


図 2.5 ステッピングモータ (ピック深さ制御)

## 第3章

### ハードウェアの改良

### 3.1 ステッピングモータの改良

昨年までのステッピングモータではステップ角が大きく動作音が大きかったため、モータをマイクロステップ対応のものに変更することによって、より滑らかで細かい制御ができるようにした。この改良により、動作音を小さくすることができた。

#### 3.1.1 ストローク制御用モータ

左右の動きを制御するモータはステップ角を4分割から32分割( $0.05625[^\circ/\text{step}]$ )にした。モータは日本電産サーボのKH56JM2-951で外形は昨年と同じものを使用した。これを図3.1に示す。ドライバはFWD2B20P15A-01を使用する。

#### 3.2.1 ピック深さ制御用モータ

上下の動きを制御するモータもステップ角を4分割から8分割( $0.225[^\circ/\text{step}]$ )にした。モータはストローク制御用と同様、日本電産サーボのKT35FM1-552で、こちらは外形42[mm]から35[mm]、重量260[g]から110[g]となった。これを図3.2に示す。ドライバはFTD3S2P22-01を使用する。

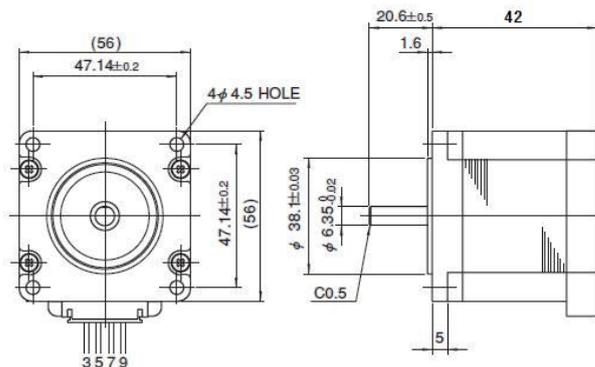


図 3.1 ストローク制御用モータ

<http://catalog.nidec-servo.com/digital/general/pdf/FWD-A.pdf>

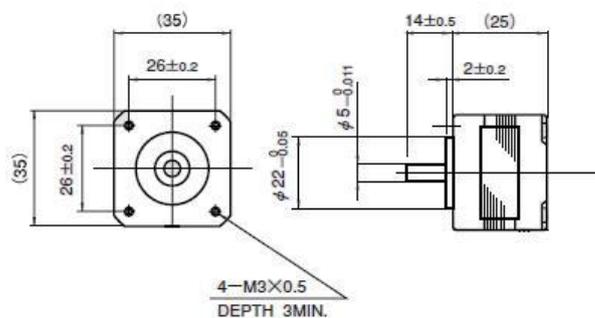


図 3.2 ピック深さ制御用モータ

<http://catalog.nidec-servo.com/digital/general/pdf/KT35.pdf>

### 3.2 可動部の軽量化

パーツを軽量化することで負荷を減らし、動作音の軽減をした。

#### 3.2.1 モータ固定具

ピック深さ制御用モータを小型のものにしたため、今年のモータ固定具に無駄な部分があったため、その部分を削った。そうすることで昨年と比べ約半分の厚さになった。これを図 3.3 に示す。

#### 3.2.2 ピック固定具

ピック固定具も同様に無駄な部分を削り、軽量化した。これを図 3.4 に示す。

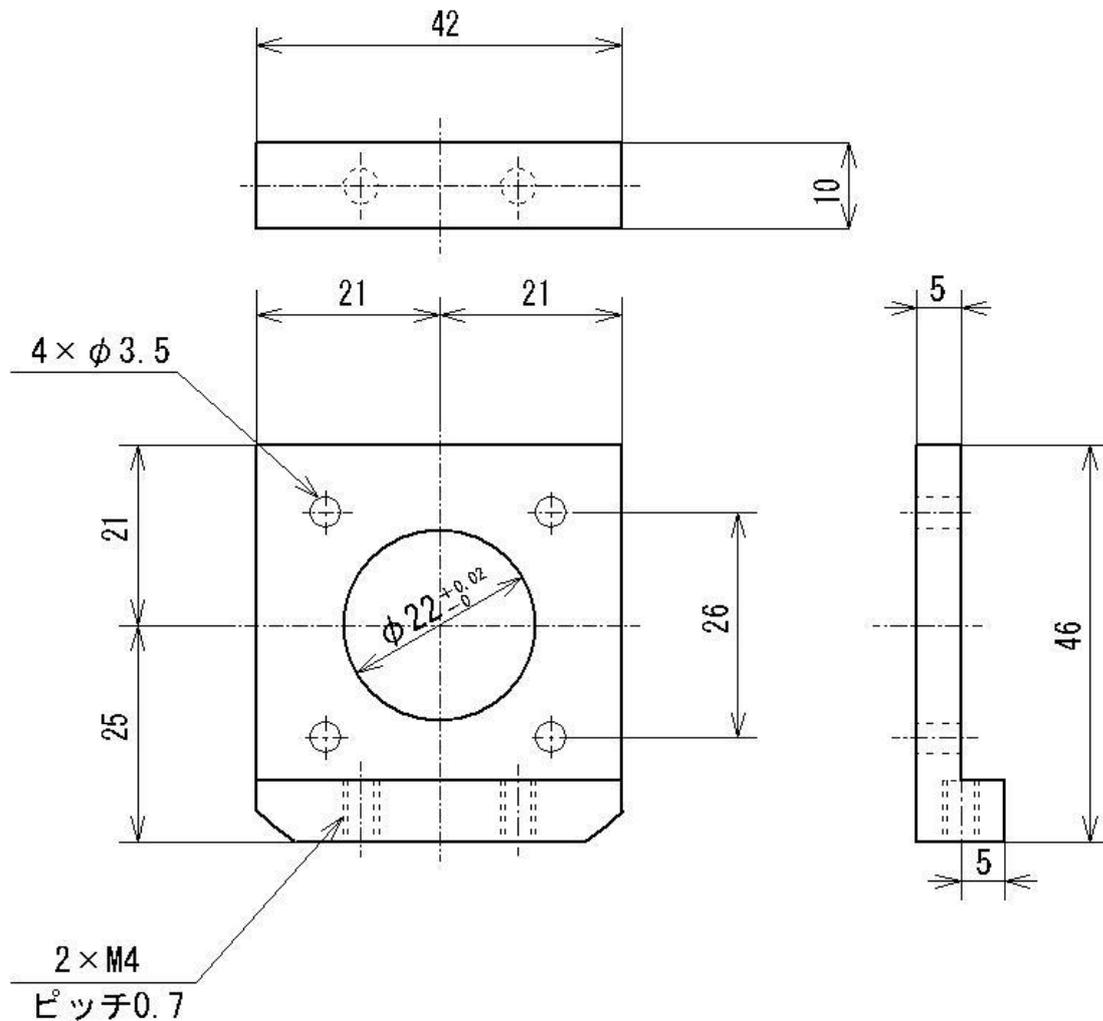


図 3.3 モータ固定具

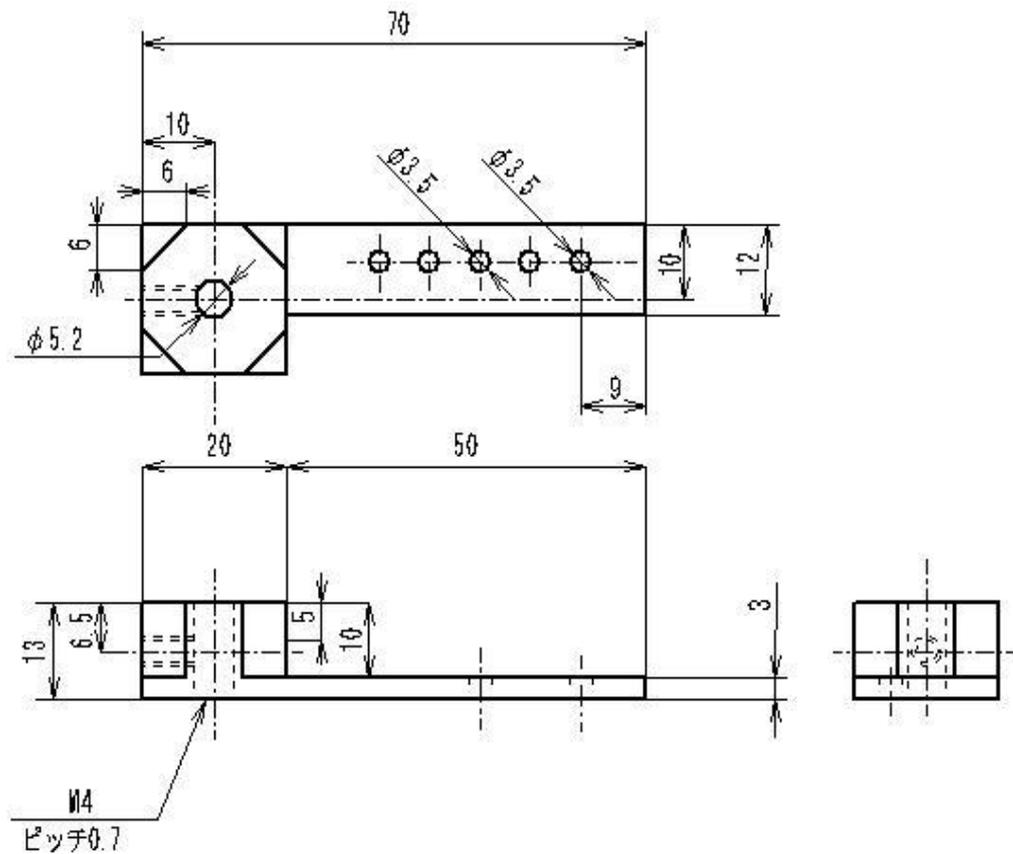
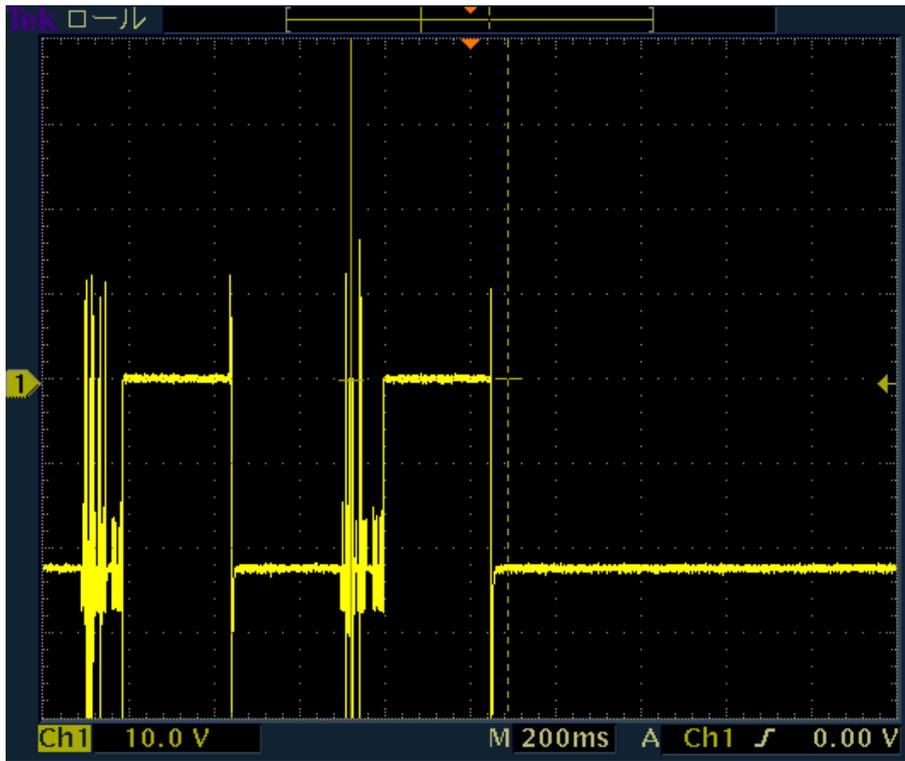


図 3.4 ピック固定具

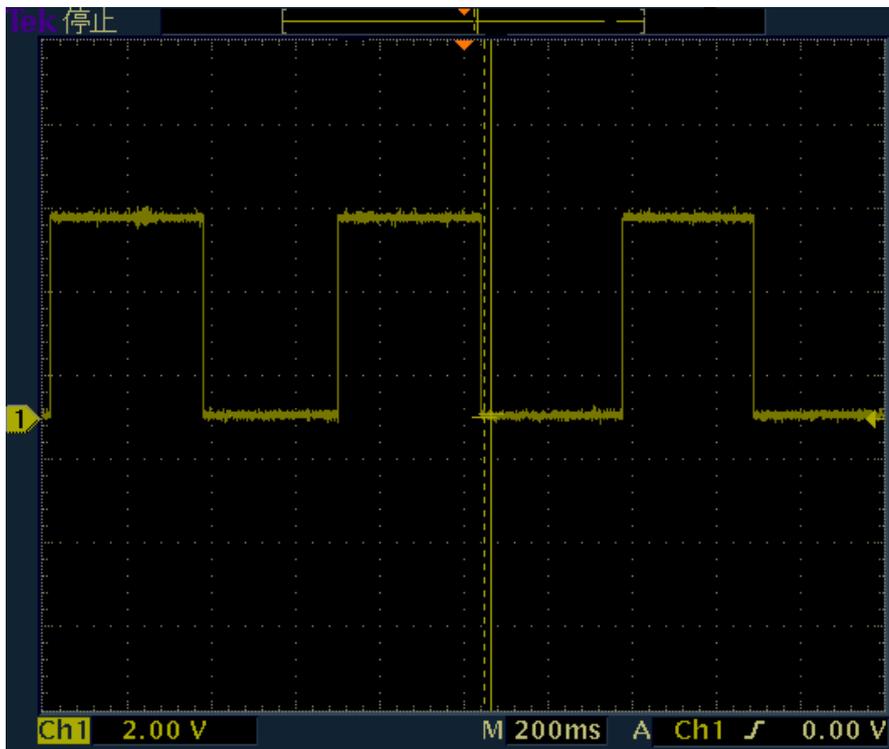
### 3.3 スイッチング回路の改良

#### 3.3.1 装置概要

目的の1つである、ミュートのためにソレノイドを高速でスイッチングする必要があった。そのためソレノイドのスイッチング回路を昨年のリレーを使った回路から、より高速のスイッチングができるトランジスタを使った回路に変更した。まずリレーとトランジスタのスイッチング時のオシロスコープでの波形の違いを図 3.5 に示す。



(a) リレー



(b) トランジスタ

図 3.5 スイッチング時の波形の違い

装置は、Tr1 がスイッチングによく用いられる 2SC1815GR、Tr2 が大きな電圧、熱量に耐えられるパワートランジスタの 2SC5200 をダーリントン接続してソレノイドが駆動できる電流まで増幅している。各トランジスタの増倍率を測ると Tr1 が約 350、Tr2 が約 73 となり、2 つを合わせた増倍率は 25550 となる。そのためベース電流を抵抗を入れて約 0.03[mA] まで落とした。抵抗は R1 が 105[k $\Omega$ ]、R2 が 500[ $\Omega$ ]となっている。ちなみに R2 はトランジスタを確実に OFF させるための抵抗である。また、逆起電力防止のために整流ダイオードをソレノイドと並列につないでいる。この装置図を図 3.6、回路図を図 3.7 に示す。また使用したトランジスタの仕様を図 3.8、図 3.9 に示す。

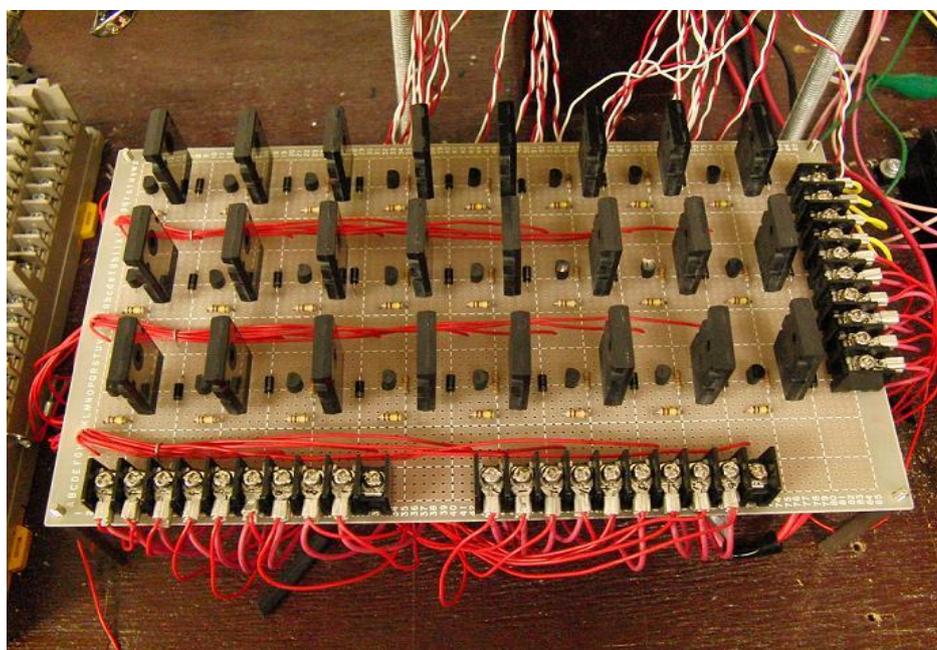


図 3.6 トランジスタを用いたスイッチング装置

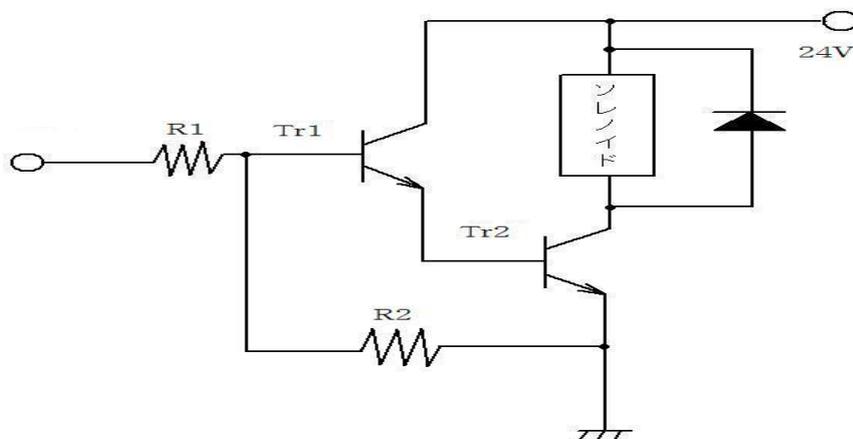
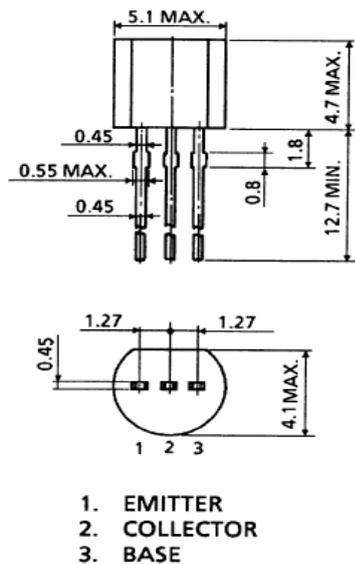


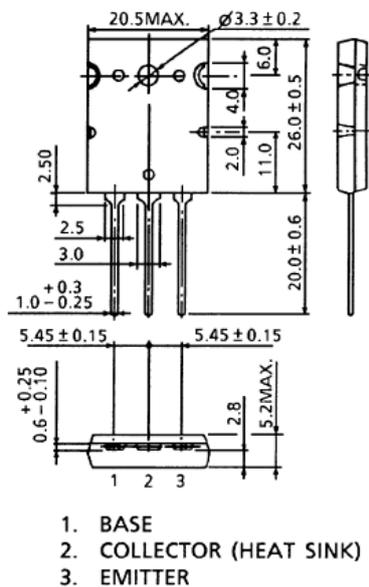
図 3.7 スwitchング装置の回路図



Characteristics	Symbol	Rating	Unit
Collector-base voltage	$V_{CB0}$	60	V
Collector-emitter voltage	$V_{CEO}$	50	V
Emitter-base voltage	$V_{EBO}$	5	V
Collector current	$I_C$	150	mA
Base current	$I_B$	50	mA
Collector power dissipation	$P_C$	400	mW
Junction temperature	$T_j$	125	°C
Storage temperature range	$T_{stg}$	-55~125	°C

図 3.8 2SC1815 の仕様

[http://www.semicon.toshiba.co.jp/docs/datasheet/ja/Transistor/2SC1815\\_ja\\_datasheet\\_071101.pdf](http://www.semicon.toshiba.co.jp/docs/datasheet/ja/Transistor/2SC1815_ja_datasheet_071101.pdf)



Characteristics	Symbol	Rating	Unit
Collector-base voltage	$V_{CB0}$	230	V
Collector-emitter voltage	$V_{CEO}$	230	V
Emitter-base voltage	$V_{EBO}$	5	V
Collector current	$I_C$	15	A
Base current	$I_B$	1.5	A
Collector power dissipation ( $T_c = 25^\circ\text{C}$ )	$P_C$	150	W
Junction temperature	$T_j$	150	°C
Storage temperature range	$T_{stg}$	-55 to 150	°C

図 3.9 2SC5200 の仕様

[http://www.semicon.toshiba.co.jp/docs/datasheet/en/Transistor/2SC5200\\_en\\_datasheet\\_061110.pdf](http://www.semicon.toshiba.co.jp/docs/datasheet/en/Transistor/2SC5200_en_datasheet_061110.pdf)

# 第4章

## 制御プログラム

#### 4.1 改良点

制御プログラムをモータの改良にプログラムの細かな調整を行った。またピック入れ込み深さを画面上ですぐに変えることができるようにした。プログラムの画面を図4.1に、深さを変える部分を図4.2に示す。



(a) 動作確認画面



(b) 演奏用画面

図 4.1 制御用プログラム



図 4.2 ピック入れ込み深さを変えるプログラム

### 3.2 動作テストプログラム

動作確認のためにプログラムを作った。それを図 4.3 に示す。

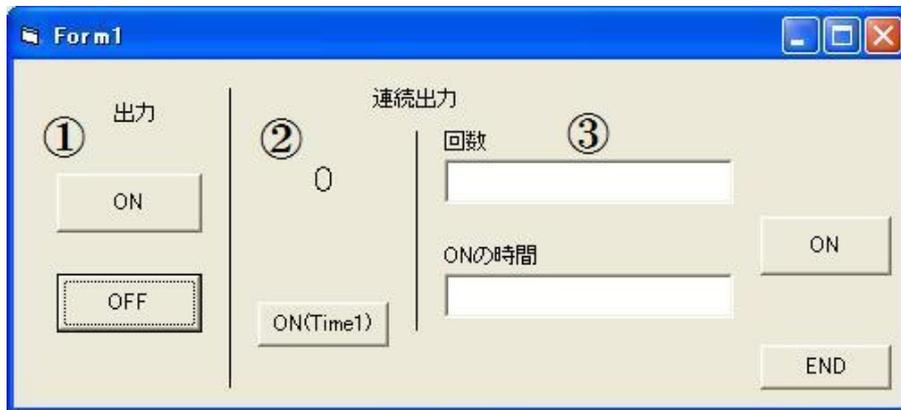


図 4.3 動作確認プログラム

- ① 出力確認用プログラム
- ② タイマーを用いたスイッチングプログラム
- ③ ムダ時間を用いた高速スイッチングプログラム

以下にプログラムコードを示す。

```

'-----変数宣言-----
Dim lpszName As String 'デバイス名
Dim nRet As Long '関数の実行結果
Dim pnBuffer(32) As Long '状態を設定するバッファ
Dim t As Long
Dim i As Long
'-----ここまで-----
'-----Load (どのギタープログラムでも必須)-----
Private Sub Form_Load()
'ボードの初期化処理-----
lpszName = "FBIDI01" & Chr(0)

```

```

hDevicehandle = DioOpen(ipszName, FBIDIO_FLAG_SHARE)

If hDevicehandle = -1 Then
    MsgBox ("デバイスのオープンに失敗しました")
    Exit Sub
End If

pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 4, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 8, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 9, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 10, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 11, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 12, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 13, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 14, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 15, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 16, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 17, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 18, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 19, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 20, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 21, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 22, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 23, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 24, 1)

pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 25, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 26, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 27, 1)

```

```

pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 28, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 29, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 30, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 31, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 32, 1)

```

‘終了処理-----

```

If nRet <> 0 Then
    MsgBox ("データの入力に失敗しました")
    nRet = DioClose(hDevicehandle)
    Exit Sub
End If

```

‘終了処理

```

nRet = DioClose(hDevicehandle)
If nRet <> 0 Then
    MsgBox ("デバイスのクローズに失敗しました")
End If
End Sub

```

‘-----ここまで-----

‘-----①-OFF ボタン-----

```

Private Sub Command1_Click()
' OFF ↑
' ボード初期化を行います
lpszName = "FBIDIO1" & Chr(0)
hDevicehandle = DioOpen(lpszName, FBIDIO_FLAG_SHARE)
If hDevicehandle = -1 Then
MsgBox ("デバイスのオープンに失敗しました")
Exit Sub
End If

```

‘出力接点の状態を設定します

```

pnBuffer(0) = 1

‘データ出力(OUT1~OUT4 までの状態を設定します)
nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 18, 1)
If nRet <> 0 Then
MsgBox ("データの出力に失敗しました")
nRet = DioClose(hDevicehandle)
Exit Sub
End If

```

’ 終了処理

```

nRet = DioClose(hDevicehandle)
If nRet <> 0 Then

```

```

MsgBox ("デバイスのクローズに失敗しました")
End If
Command2.SetFocus
End Sub
'-----ここまで-----
'-----①-(出力確認 ON ボタン)-----
Private Sub Command2_Click()
'ON↑
'ボード初期化を行います
lpszName = "FBIDIO1" & Chr(0)
hDevicehandle = DioOpen(lpszName, FBIDIO_FLAG_SHARE)
If hDevicehandle = -1 Then
MsgBox ("デバイスのオープンに失敗しました")
Exit Sub
End If

'出力接点の状態を設定します
pnBuffer(0) = 0

'データ出力(OUT1~OUT4 までの状態を設定します)
nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 18, 1)
If nRet <> 0 Then
MsgBox ("データの出力に失敗しました")
nRet = DioClose(hDevicehandle)
Exit Sub
End If

'終了処理
nRet = DioClose(hDevicehandle)
If nRet <> 0 Then
MsgBox ("デバイスのクローズに失敗しました")
End If
Command1.SetFocus
End Sub
'-----ここまで-----
'-----②-(タイマーを用いたスイッチングプログラム)-----
Private Sub Command3_Click()
Timer1.Enabled = True
End Sub

Private Sub Timer1_Timer()
'ボード初期化を行います
lpszName = "FBIDIO1" & Chr(0)
hDevicehandle = DioOpen(lpszName, FBIDIO_FLAG_SHARE)
If hDevicehandle = -1 Then
MsgBox ("デバイスのオープンに失敗しました")
Exit Sub
End If

```

```

Label3.Caption = Label3.Caption + 1
If Label3.Caption = 1 Then

    pBuffer(0) = 0
        nRet = DioOutputPoint(hDevicehandle, pBuffer(0), 11, 1)
    ElseIf Label3.Caption = 2 Then
        pBuffer(0) = 1
            nRet = DioOutputPoint(hDevicehandle, pBuffer(0), 11, 1)
    ElseIf Label3.Caption = 4 Then
        pBuffer(0) = 0
            nRet = DioOutputPoint(hDevicehandle, pBuffer(0), 11, 1)
    ElseIf Label3.Caption = 5 Then
        pBuffer(0) = 1
            nRet = DioOutputPoint(hDevicehandle, pBuffer(0), 11, 1)
    ElseIf Label3.Caption = 7 Then
        pBuffer(0) = 0
            nRet = DioOutputPoint(hDevicehandle, pBuffer(0), 11, 1)
    ElseIf Label3.Caption = 8 Then
        pBuffer(0) = 1
            nRet = DioOutputPoint(hDevicehandle, pBuffer(0), 11, 1)
    End If

' 終了処理-----
If nRet <> 0 Then
    MsgBox ("データの入力に失敗しました")
    nRet = DioClose(hDevicehandle)
    Exit Sub
End If
' 終了処理
nRet = DioClose(hDevicehandle)
If nRet <> 0 Then
    MsgBox ("デバイスのクローズに失敗しました")
End If
End Sub
'-----ここまで-----
'-----③- (ムダ時間を用いた高速スイッチングプログラム) -----
Private Sub Command6_Click()
' ボードの初期化処理-----
    ipzName = "FBIDIO1" & Chr(0)
    hDevicehandle = DioOpen(ipzName, FBIDIO_FLAG_SHARE)

    If hDevicehandle = -1 Then
        MsgBox ("デバイスのオープンに失敗しました")
        Exit Sub
    End If

    For i = 1 To Val(Text1.Text)
        pBuffer(0) = 0
            nRet = DioOutputPoint(hDevicehandle, pBuffer(0), 32, 1)
    
```

```

    For t = 1 To Val(Text2.Text): Next t
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 32, 1)
    For t = 1 To Val(Text2.Text): Next t
Next i
End Sub
'-----ここまで-----
'-----Unload(どのギタープログラムでも必須)-----
Private Sub Form_Unload(Cancel As Integer)
'ボードの初期化処理-----
ipszName = "FBIDI01" & Chr(0)
hDevicehandle = DioOpen(ipszName, FBIDIO_FLAG_SHARE)

If hDevicehandle = -1 Then
    MsgBox ("デバイスのオープンに失敗しました")
    Exit Sub
End If

pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 4, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 8, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 9, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 10, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 11, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 12, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 13, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 14, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 15, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 16, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 17, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 18, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 19, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 20, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 21, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 22, 1)

```

```

pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 23, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 24, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 25, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 26, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 27, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 28, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 29, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 30, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 31, 1)
pnBuffer(0) = 1
    nRet = DioOutputPoint(hDevicehandle, pnBuffer(0), 32, 1)

```

```

' 終了処理-----

```

```

If nRet <> 0 Then
    MsgBox ("データの入力に失敗しました")
    nRet = DioClose(hDevicehandle)
    Exit Sub
End If
' 終了処理
nRet = DioClose(hDevicehandle)
If nRet <> 0 Then
    MsgBox ("デバイスのクローズに失敗しました")
End If
End Sub

```

```

'-----ここまで-----

```

```

'-----END ボタン-----

```

```

Private Sub Command5_Click()

```

```

End

```

```

End Sub

```

```

'-----ここまで-----

```

## 第 5 章

# 音の強弱についての実験及び考察

## 5.1 ピック側

### 5.1.1 強弱のつけ方

去年のプログラムではピックを入れる深さを変えただけでは、聞いてとれるほど音量に変化はなかった。しかしいろいろ試しているうちに音の強弱が明確に出るポイントが見つかった。音に強弱がつくのが、弦のすぐ隣から弾くか、弾きたい弦に対して左右に 200 パルス以上の距離がある時である。なので、今まではピックの止まる位置を常に弦と弦の間にしてしたが、弦のすぐ隣で止まるようにし、止まる位置を 7 ヲ所から 13 ヲ所に増やした。これを点線を去年まで、実線を今年の設定として図 5.1 に示す。

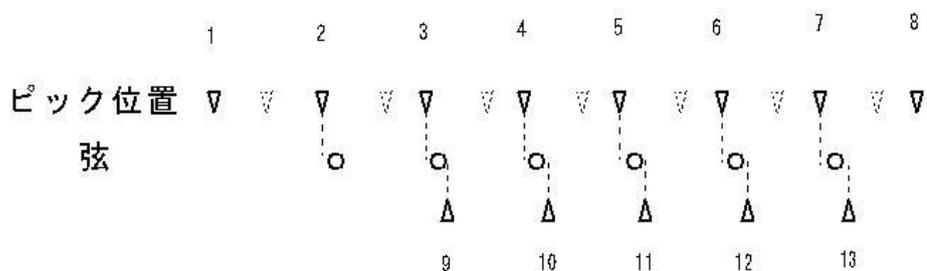


図 5.1 ピック停止位置

### 5.1.1 解析方法

音の解析方法については、PC からの命令で装置を動かし、その音をギターに取り付けたピックアップから PC のマイク端子に繋げ、解析ソフトにより録音した。解析ソフトには Sound Engine Free というフリーソフトを使用した。

### 5.1.2 実験方法

5 種類の深さでの音量の変化を調べた。ちなみにピックの入れ込みの深さは 15 パルスの時で約 2mm で、これ以降約 0.7mm ずつ増加する。実験は 1 秒に 1 回弾き、それを 10 秒間続け、その時の音量を測定した。ここでは 6 弦で実験を行った。

### 5.1.3 実験結果及び考察

結果は図 5.2 のようになり、図 5.3 に各音量での波形を示す。

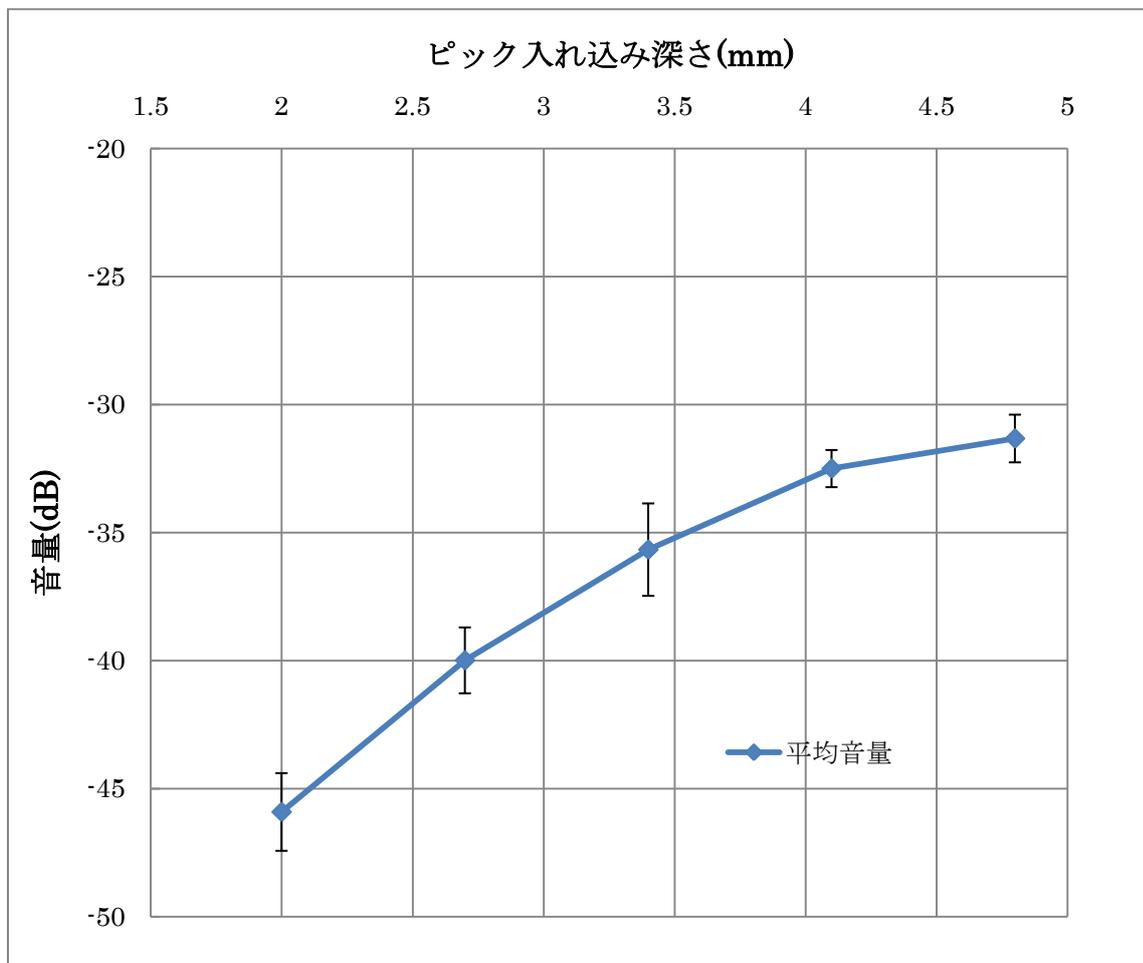


図 5.2 ピック入れ込み深さによる音量変化

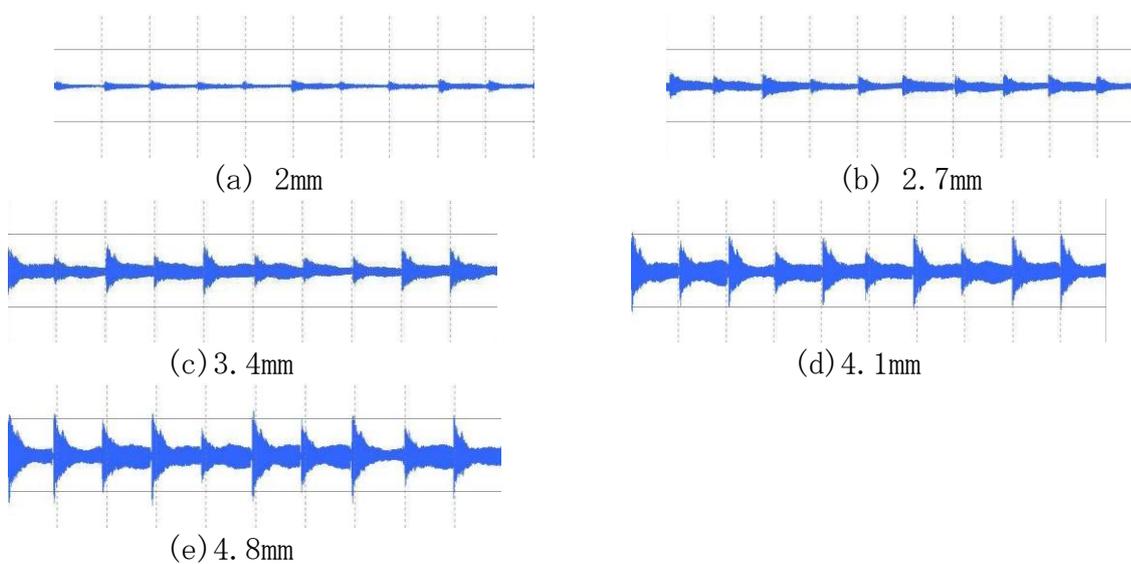


図 5.3 実験時の波形

## 5.2 ソレノイド側

### 5.2.1 強弱のつけ方

ソレノイドに流す電流を高速でオンとオフを繰り返すことによって制御し、押弦圧を調節してミュート（軽く弦に触れ音を切る奏法）を行えるようにする。

### 5.2.2 実験方法

I/O ボードからの電流を、図 5.4 に示す動作確認プログラムの③で高速でスイッチングすることでソレノイドに流れる電流を制御し、押弦圧を弱くすることでミュートを行った。測定はスイッチングの周期を  $40\ \mu\text{s}$  から増やしながら測定した。

### 5.2.3 実験結果及び考察

ミュートした時の音の波形を図 5.4 に示す。左から 7 番目までがミュート時で、一番右の波形が普通に弾いたときの波形である。ミュート時は普通に弾いた時と比べ、音が短く切れていることが分かる。今回の実験で周期が  $280\ \mu\text{s}$  の時が最も安定し、動作音も小さくミュートができた。この時のオシロスコープでの波形を図 5.5 に示す。また  $500\ \mu\text{s}$  をこえると動作音が大きくなりミュートもできなくなった。

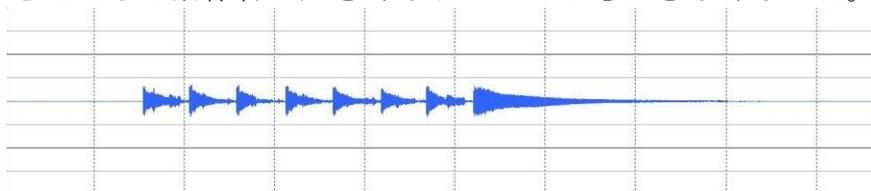


図 5.4 ミュート時の波形

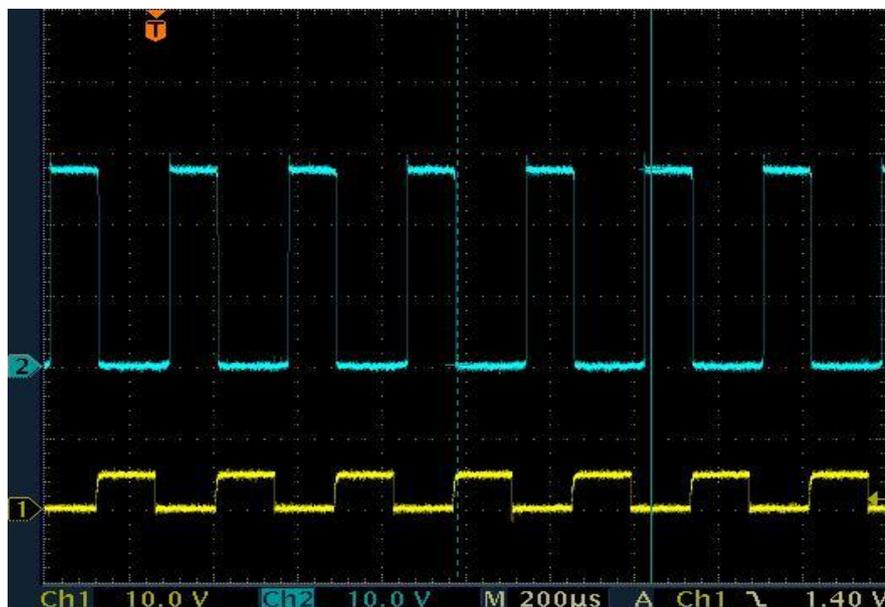
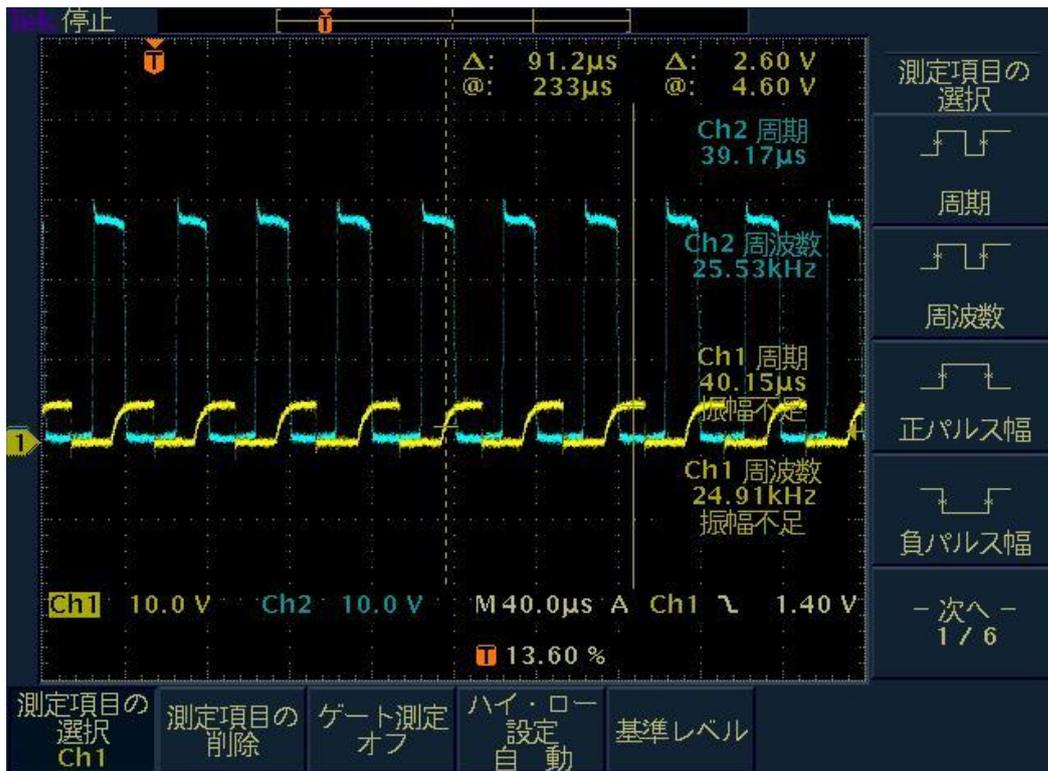


図 5.5 周期  $280\ \mu\text{s}$  時の波形

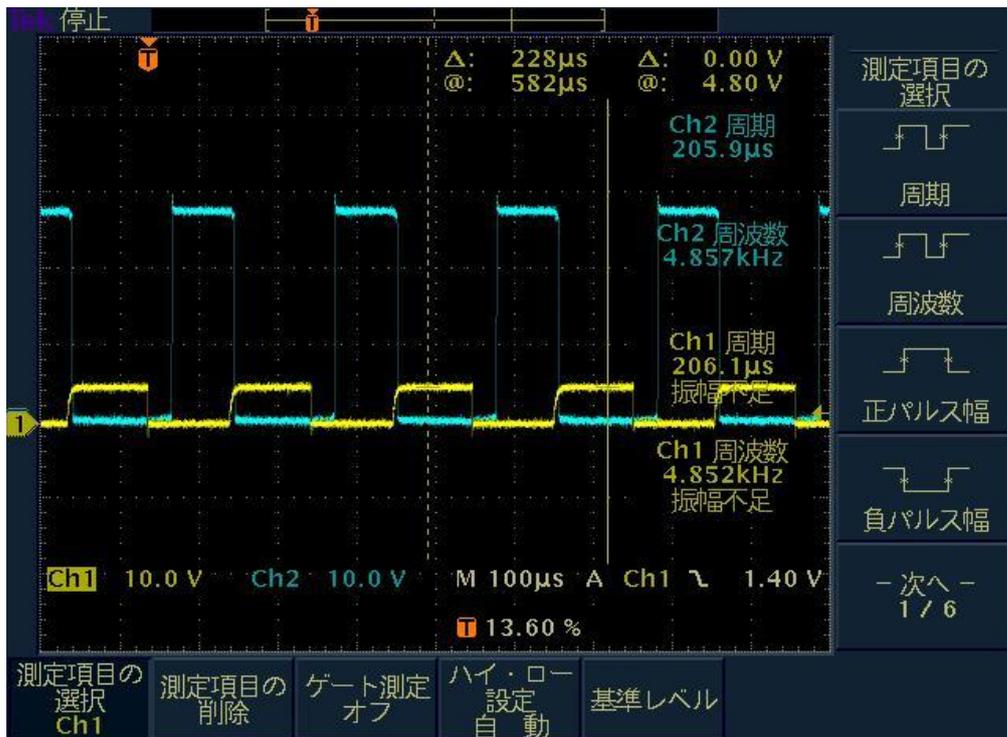
また、以下に実験で測定した周波数の波形を示す。



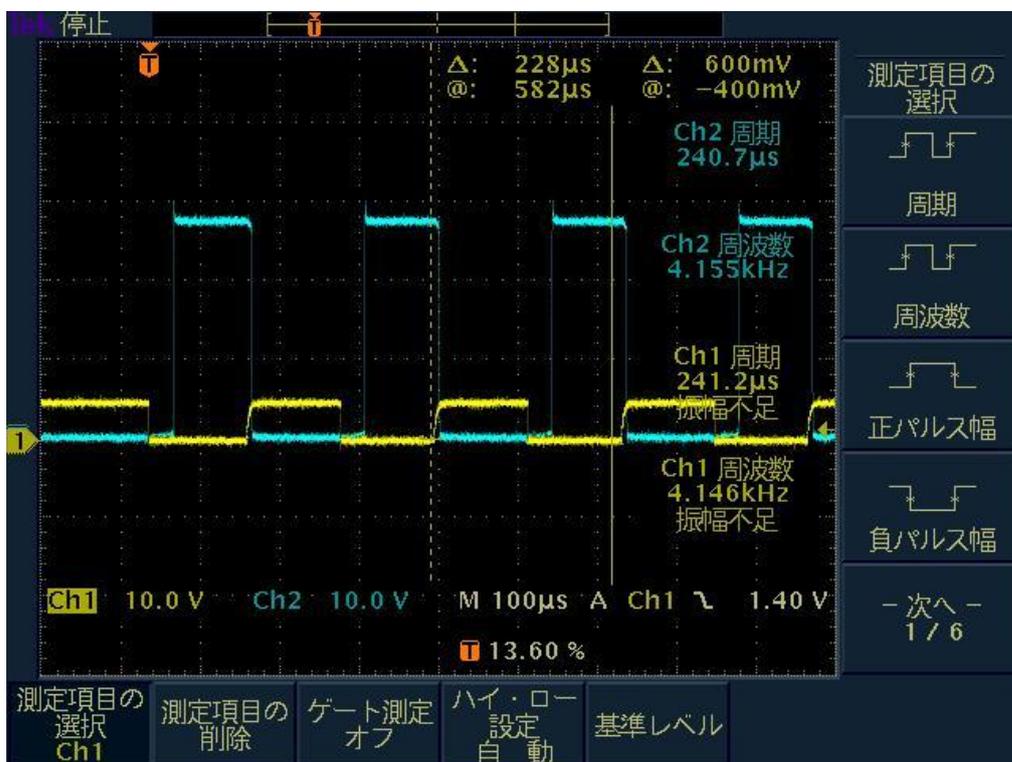
(a) 40 μs



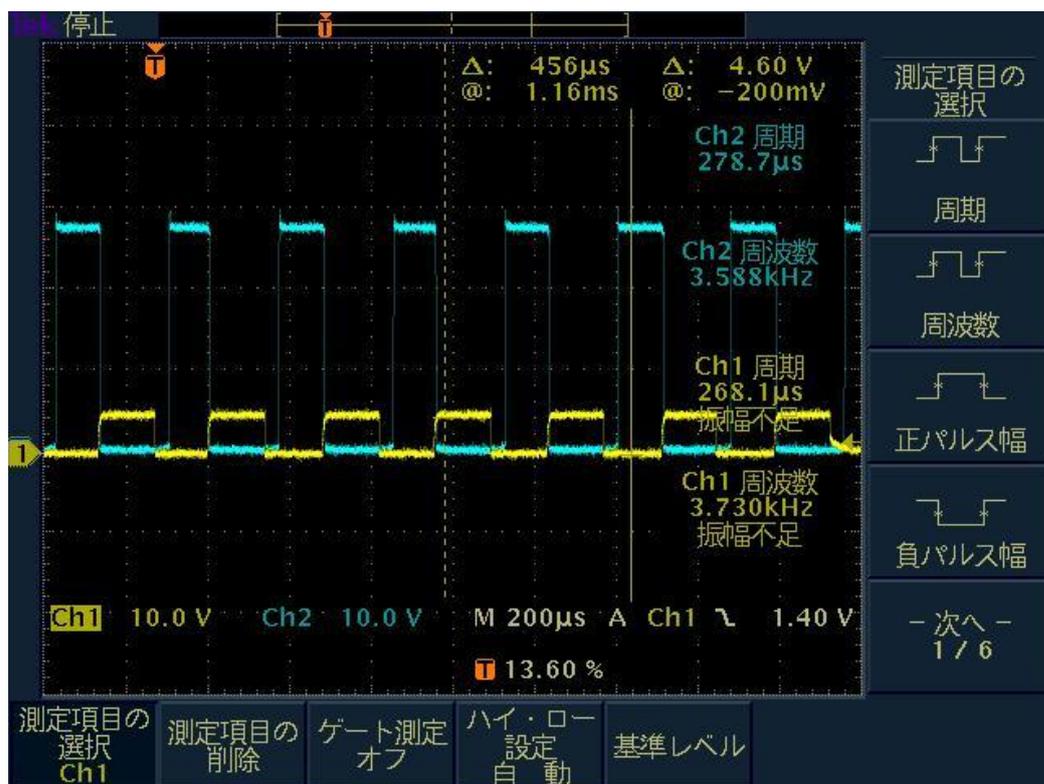
(b) 80 μs



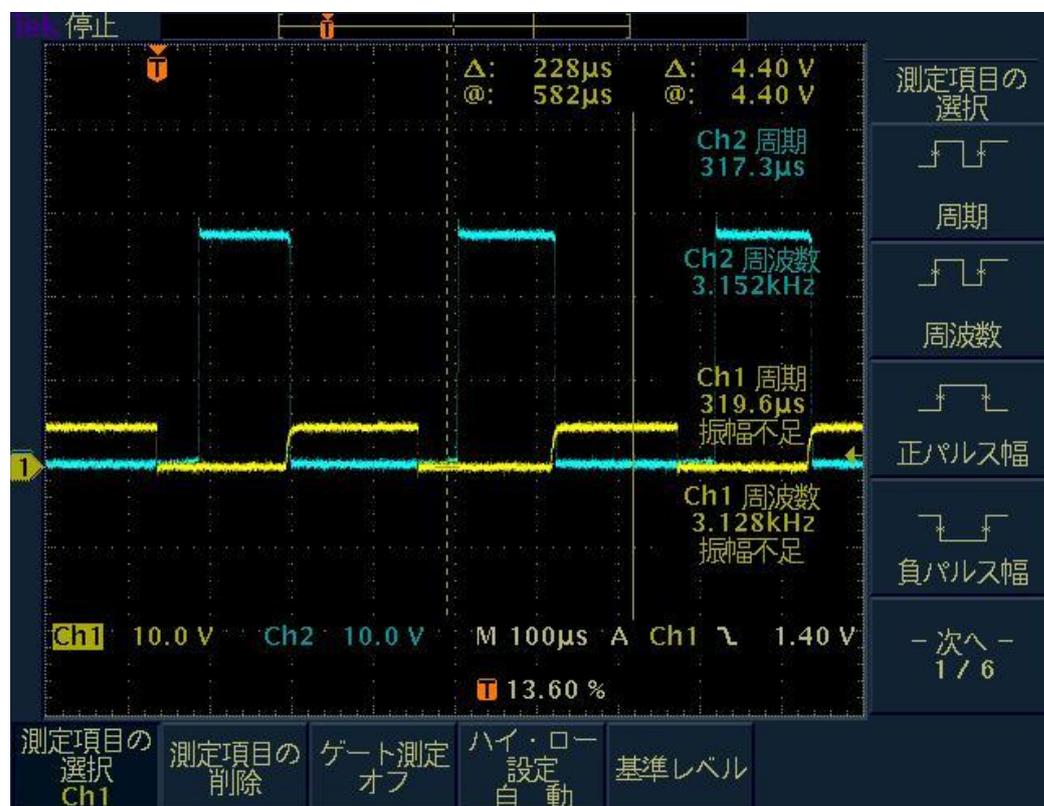
(c) 200  $\mu$  s



(d) 240  $\mu$  s



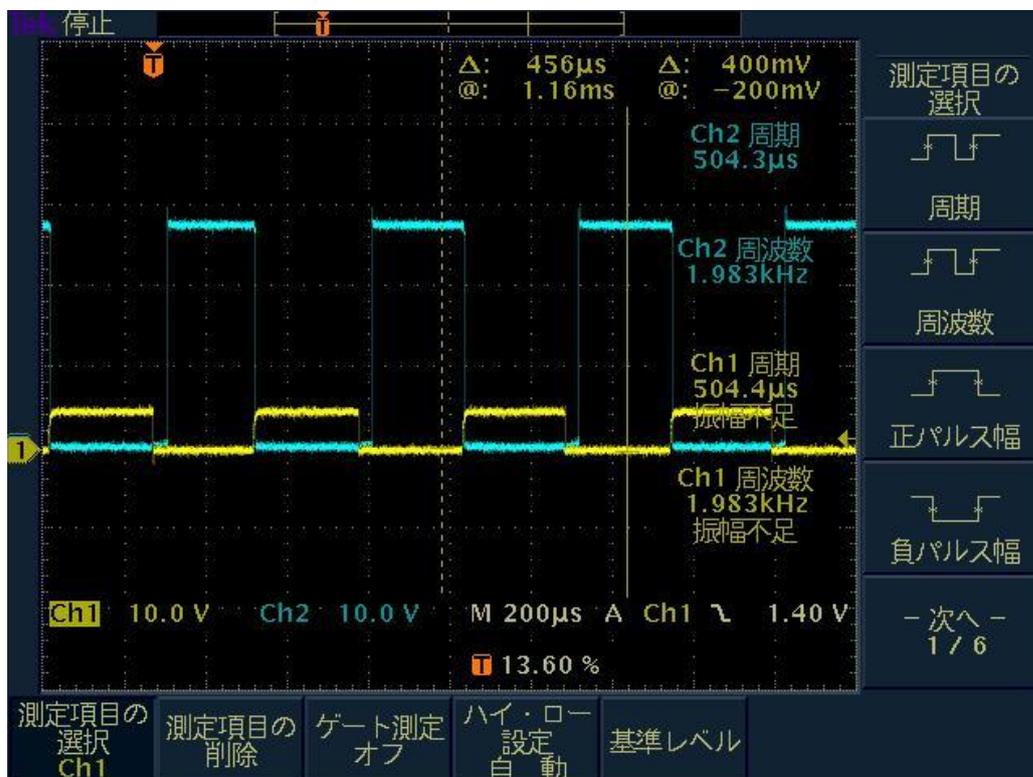
(e) 280  $\mu$  s



(f) 320  $\mu$  s



(g) 400  $\mu$  s



(h) 500  $\mu$  s

図 5.6 各周期での波形

# 第 6 章

## 結言

## 6.1 結言

- (1) ステッピングモータの変更とパーツの軽量化によって、去年よりも動作音を軽減することができた。
- (2) 去年よりも明確に音の強弱をつけることができた。
- (3) トランジスタに変えることにより、高速スイッチングによるミュートができた。

## 6.2 今後の課題

- (1) 高性能にするだけでなく、もっと実験や解析を行う。
- (2) 曲のプログラミングが難しいので誰でも行えるように簡単にする。
- (3) ピックの数を増やすなど、さらに演奏に幅を持たせる。

## 6.3 謝辞

本研究に当たり数々のご指導、ご助言をして頂いた河村良行教授、パーツを加工するにあたりご指導、ご助言を頂いた工作センターの先生方、また、プログラムなどご指導いただいた米岡和樹先輩、本当にありがとうございました。

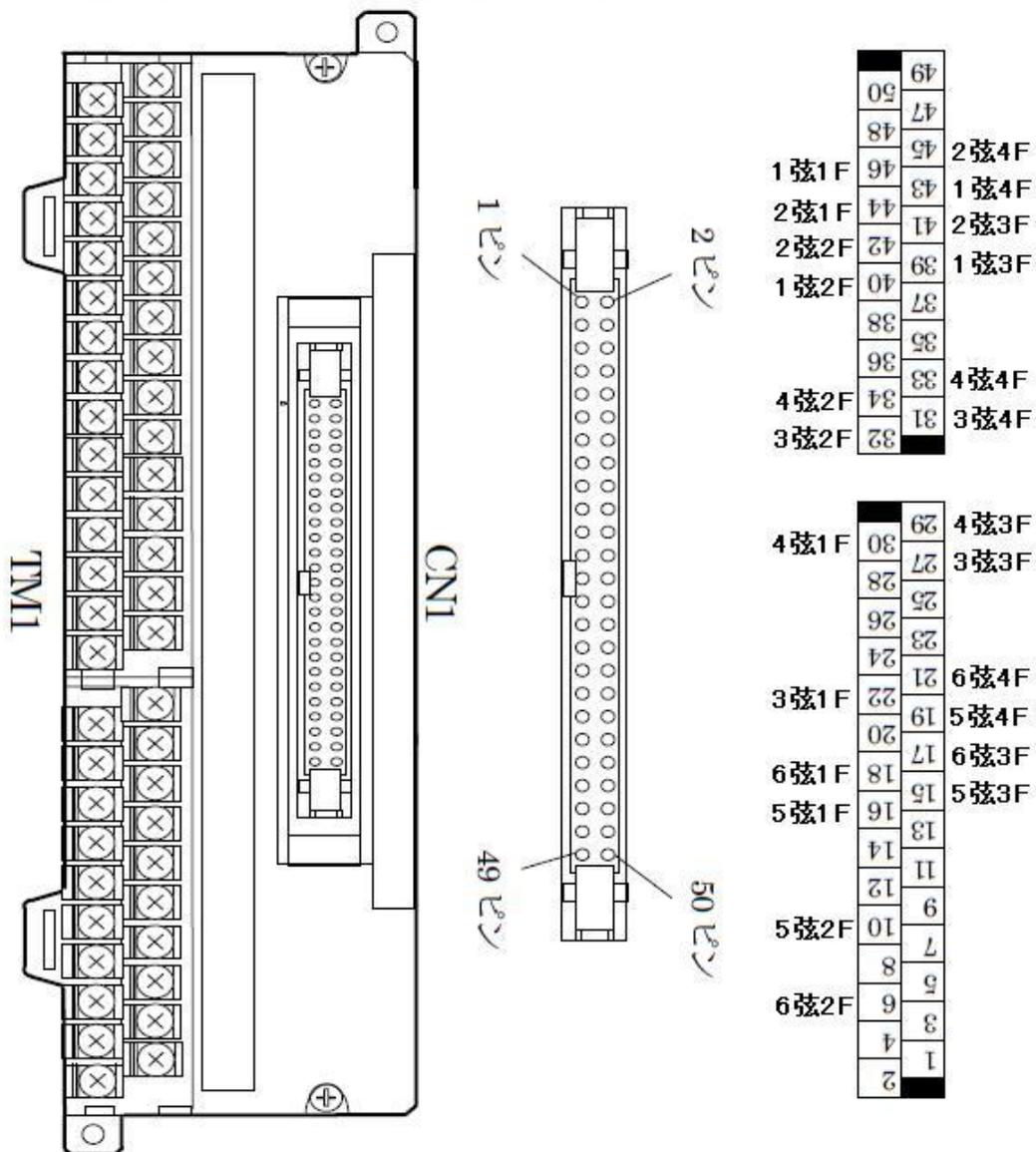
## 参考文献

- (1) 吉川和光「実践メカトロニクス アクチュエータ」産業図書 p. 30-38
- (2) トランジスタを使う上での基礎知識  
<http://www.asahi-net.or.jp/~bz9s-wtb/doc/circuit/No1/tic1c1a.pdf>
- (3) 日本電産サーボ  
<http://www.nidec-servo.com/jp/>
- (4) トランジスタのデータベース  
2SC1815GR  
[http://www.semicon.toshiba.co.jp/docs/datasheet/ja/Transistor/2SC1815\\_ja\\_datasheet\\_071101.pdf](http://www.semicon.toshiba.co.jp/docs/datasheet/ja/Transistor/2SC1815_ja_datasheet_071101.pdf)  
2SC5200  
[http://www.semicon.toshiba.co.jp/docs/datasheet/en/Transistor/2SC5200\\_en\\_datasheet\\_061110.pdf](http://www.semicon.toshiba.co.jp/docs/datasheet/en/Transistor/2SC5200_en_datasheet_061110.pdf)
- (5) インターフェイス  
<http://www.interface.co.jp/>
- (6) ソレノイドカタログ  
<http://www.kgs-jpn.co.jp/Catalog/ca/ca1548.html>

# 付録

- ・ 入出力ボード (PCI-2703A) のピンとそれに対応したソレノイドを示す。

## コネクタピンアサインメント



入出力ボード(PCI-2703A)の仕様

デジタル入出力部

	項目	仕様
入力仕様	入力信号名	IN1～IN32(入出力共用)
	入力形式	TTLシュミット入力
	入力論理	“1”←“Low”(ON, 短絡) “0”←“High”(OFF, 開放)
	最大定格電圧	DC+5V
	入力信号電圧範囲	DC0V～DC+5V
	入力抵抗(保護機能)	10k $\Omega$ プルアップ抵抗, 10 $\Omega$ 保護抵抗付き
	Lowレベル入力電流	$I_{IL} = -0.9\text{mA}(\text{MAX})$
	Highレベル入力電流	$I_{IH} = +40\mu\text{A}(\text{MAX})$
	Lowレベル入力電圧	$V_{IL} = +0.8\text{V}(\text{MAX})$
	Highレベル入力電圧	$V_{IH} = +2.0\text{V}(\text{MIN})$
	入力応答時間	$T_{RON}: 0.1\mu\text{s}(\text{TYP}), T_{ROFF}: 0.1\mu\text{s}(\text{TYP})$
出力仕様	出力信号名	OUT1～OUT32(入出力共用)
	出力形式	TTLオープンコレクタ出力(10k $\Omega$ プルアップ抵抗付き)
	出力論理	“1”→“Low”(ON, 短絡) “0”→“High”(OFF, 開放)(初期値)
	最大定格電圧	DC+5V
	出力信号電圧範囲	DC0V～DC+5V
	出力抵抗	10k $\Omega$ プルアップ抵抗, 10 $\Omega$ 保護抵抗付き
	最大出力電流	$I_{OL} = +40\text{mA}(\text{MAX})$
	Lowレベル出力電圧	$V_{OL} = +0.5\text{V}(\text{MAX})(I_{OL} = +12\text{mA時})$
	出力応答時間	$T_{RON}: 0.5\mu\text{s}(\text{TYP})(\text{最大負荷時})$ $T_{ROFF}: 0.5\mu\text{s}(\text{TYP})(\text{最大負荷時})$ (立ち上がりは負荷が小さいほど遅くなります。)

- 実際の演奏は河村研究室のホームページに示す。

<http://www.fit.ac.jp/~y-kawa/>