

# 「情報工学実験 3」 進捗レポート

学生番号： 2031133

氏名： 増田瑞樹

題目：

KCS を用いた 8080 マイコンの音声データインプット

詳しいデータ置き場 <https://github.com/nex-finger/jikken3>

前回から行った作業内容：

私は先々週に計画書に OK が出たので、先々週から行った内容を示す。

まず論理回路について、情報工学科の学生は熟達していないと自覚しており、電気研究会に伺い意見をもらった。結果、計画書の内容から大きく回路設計を刷新することになった。

まず、設計した出力回路について示す。当初は 4bit のデコーダを 24 個のインバータと 24 個の NAND 回路による設計を想定していたが、電気研究部員から 4bit デコーダの IC があることを教わり、後日入手した。

4bit デコーダ IC SN74159N データシート

<https://html.alldatasheet.com/html-pdf/27375/TI/74159/19/1/74159.html>

次に、回路のショートにはリレーを用いた実装を予定していたが、フォトカプラに変更した。リレーよりも小さい電流で動作する(40mA → 10mA)ので、オープンコレクタである 74159(4bit デコーダ)は電流をトランジスタで増幅する必要がなくなった(許容電流 < 16mA)。

フォトカプラ TLP592A データシート

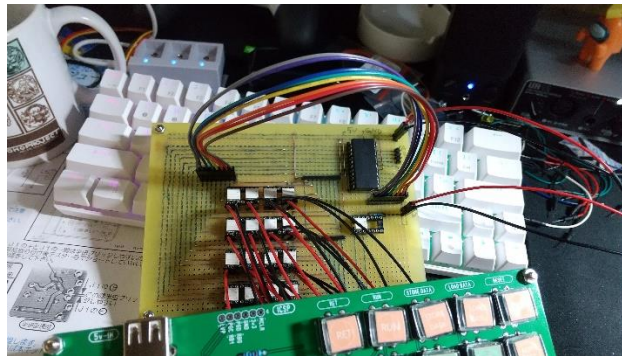
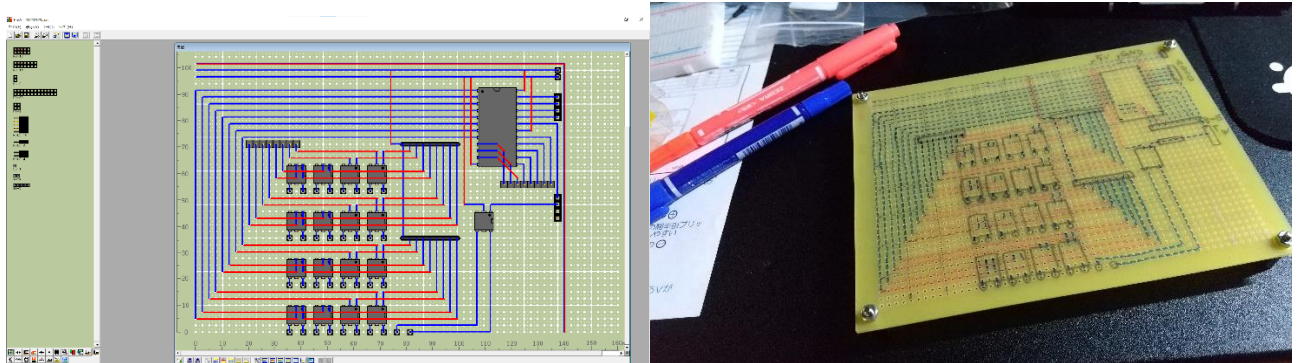
<https://toshiba.semicon-storage.com/jp/semiconductor/product/isolators-solid-state-relays/photorelay-mosfet-output/detail.TLP592A.html>

電源について、~~電流の大きさを考慮し Arduino からではなく個別に用意したが~~、リレーからフォトカプラに部品を変更したのでこの必要はなくなった。

ユニバーサル基板で回路を作成し、自室ではんだ付けを行った。出力回路を用いない小さいプログラムの入力も考えて、取り外しができる設計にした。回路は完成した。

以下に完成した回路を掲示する。

配線と実際の基板(写真 1,2,3)



次に、入力音声について調査したことを示す。

入力は 3.5mm ステレオイヤホン端子を用いる(実際にはモノラルで十分であるが、現在ステレオのほうが圧倒的に世間に入手性に優れているため、このうち左右のどちらかだけ聴取する)。

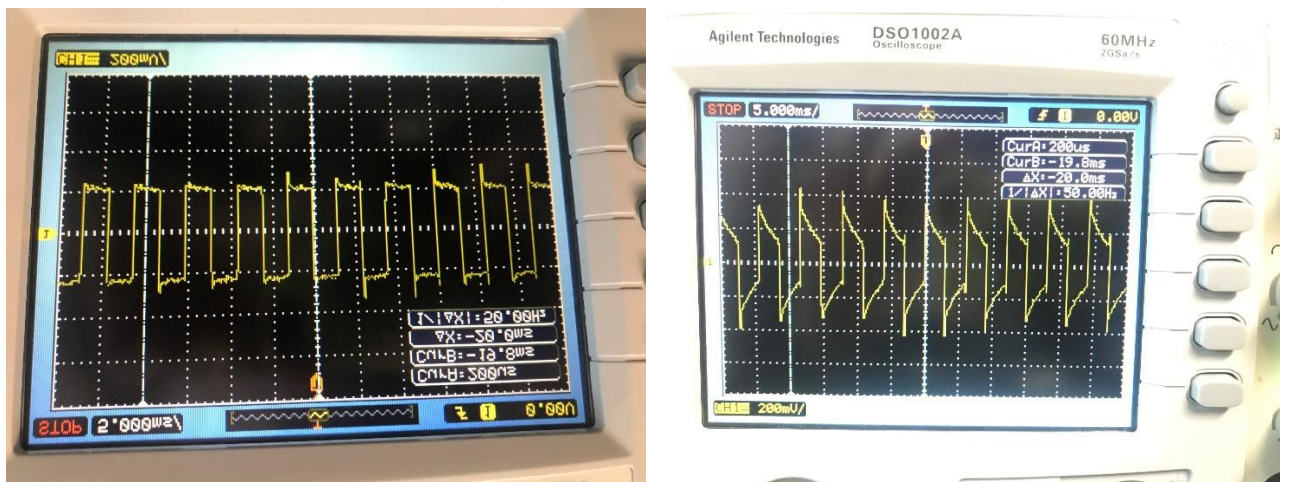
波の生の電圧を測定したところ-0.2V と 0.2 の矩形波を観測できた(写真 4)。

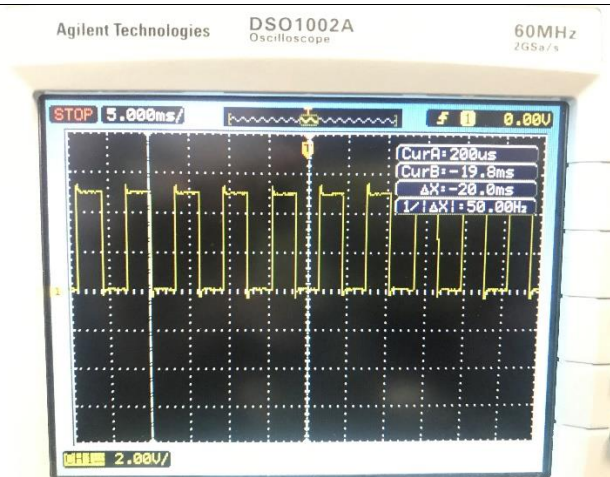
次に、電圧の微分回路の出力を測定したところ以下ようになった(写真 5)。

微分した電圧をコンパレータにかけると綺麗な 0V と 5V の矩形波になった(写真 5)。

抵抗は 49:1 で  $V_{cc}$  は 5V なので、閾値の電圧は 0.1V になる。

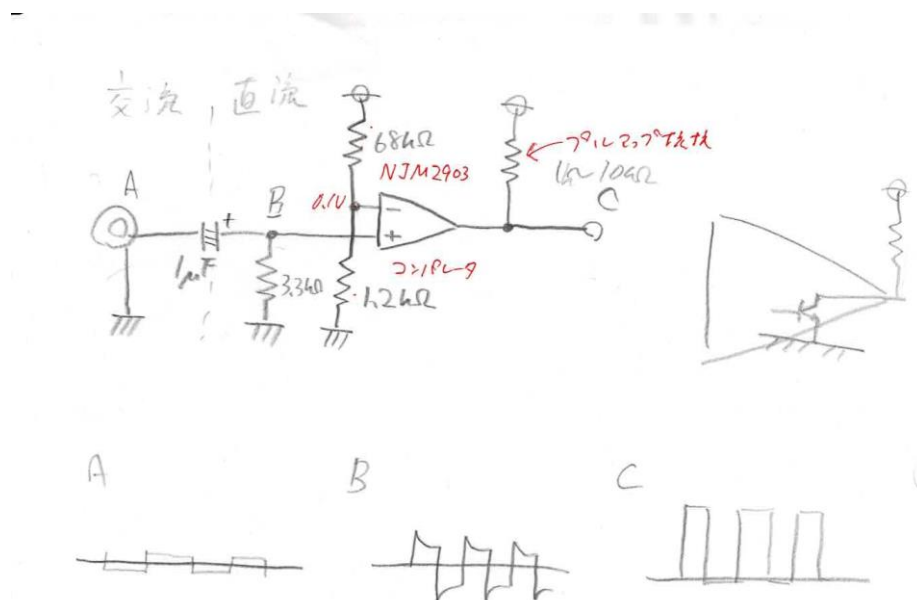
写真 4,5,6(A 点, b 点, c 点)





回路は以下のように作成した. (pdf)

### 音声入力回路概略



これより, まず Arduino に負電圧がかかることは無くなった. その上, 5V と 0V の波になったことで analogread ではなく digitalread でプログラムを実装できるようになった. analogread の動作は 120 $\mu$ s, digitalread の動作は 4 $\mu$ s であることから, この変更はとても有益であると確信する. 回路は完成していない, 次週以降完成を目指す.

最後に, 8080 マイコンの方の比較プログラムについて考えた.

8080 マイコンは 8000H から 83FFH までのメモリが存在し, そのうち 8000H から 81FFH までを比較データ, 8200H から 82FFH をプログラム, 8300H 以降をスタックや変数などに, 83C6H 以降はモニタプログラムの管理下にすると考えていたが, 実際にプログラムについて記述してみると, 256Byte どころか, 80Byte 程度で比較プログラム及びそれに使用する変数を格納することができた. よって, 1 回の比較は

256Byte よりも少し大きい数字にできることがわかった。プログラムは完成していない、バグによるもの(2レジスタの交換ができるものとできないものがあるのに全てできる体でコーディングしてしまった?)

次回までの作業予定：

- ・Arduino の割り込み処理の理解, 実装  
MsTimer2 というライブラリを使う予定
- ・比較プログラムの完成  
なるべく小さいサイズでの実装
- ・音声入力装置の作成  
秋月で必要部品を購入、ユニバーサル基板うえお重ねて実装する

要検討事項 (To Do リスト)：

- ・KCS のフォーマット詳細について  
何 bps に設定するか、またもっと低い周波数での実装の方が良いのではないか
- ・信号の周波数特定について  
コンパレータの出力を Arduino の入力とし、一定時間毎に電圧を測定し、何回連続で High になったかで検出？上手く行く気がしない
- ・誤り訂正の実装について  
ハミング符号を使う？ 散発的誤りに適している。4bit のデータ送信につき 3bit の誤り符号、合計 7bit 中誤りが 0 回か 1 回ならば訂正可能
- ・ノイズ対策について  
具体的な対策のアイデアが全くない。