

## 「情報工学実験 3」 実験計画書

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タ

目的（作成予定のシステムは何を実現するのか？ どんな課題を解決するのか？）：

intel8080 は CPU における黎明期のものであり,アセンブリや機械語もとても簡素な命令なため,学習用途としてしばしば扱われる.しかし,個人が所有できる TK-80 互換機ではアセンブリのコーディングに対し,メモリに値を打ち込む時間,与えた値の確認などのプログラミングとは関係のない要素による手間が多い.

そこで,今回作成するのは,8080 マイコンに自動的にデータを入力する装置、またカセットテープなどに保存されている KCS を用いた音情報をバイナリに変換する機構を提案する.これにより,テキストエディタで作成した機械語を手入力することなく,高速かつ正確にマイコンにインプットできる.

アピールポイント（どこが新しいのか？ 何の役に立つのか？ 解決すべき技術的課題は何か？）：

8080 マイコンにはその時代的背景から,現代的なデバイスとはほぼ存在しない.今回使用するのはワンボードマイコン TK-80 の互換機,ZK-80 [ATD-B-001]だが,この動作は機械語レベルで intel8080A と互換性を持つ.ZK-80 の自動入力ロボットは存在するが,これは cpu が z80 対応のものに限る.また,KCS (時間当たりの音の高さによりデータを保存している形式) の読み込みは,70 年代当時,精度の悪いアナログ回路により実現していた.その波の解釈をデジタルなプログラミングにより実現することは,当時の考え方とは全く新しいものとなる.

ZK-80 の機械語の書き込みは,16 進キーボードによってメモリの電圧を操作することで行える.キーボードの入力ミスもしばしば起こる.これらの自動化によって,本来行いたい機械語やアセンブリの学習に多くの時間を割くことができるうえ,入力ミスによるバグ,本来は起きない,理解を妨げることもなくなる.

解決すべき課題として,ボーレートの指定が挙げられる.これは通信速度を表しているが,ボーレートを上げれば当然時間あたりに転送できる情報は増える.しかしその反面読み込みエラーの危険性も高まることになる.

アイデアの概要（現在考えていることを可能な限り詳細に説明、必ず図を使って説明すること）：

今回制御したいワンボードマイコンのことを ZK-80, Arduino マイコンのことは Arduino と以下に示す。

ZK-80 には、機械語の入力するための 16 進キーと 9 つのファンクションキーが実装されている。このうちプログラムの入力に必要なのは、17 個のボタンになる。これらをリレー回路でショートさせることで、回路としてはタクトスイッチが押下されたときと同じ状態になるので、これを利用する。ショートさせる ZK-80 のキーは、16 つの数値キーと 1 つのファンクションキーからなり、数値キーは Arduino の 4 つのピンの電圧から、4bit の情報として扱い、適切なキーをショートさせる。ファンクションキーについては、専用のピンを 1 つ設けることで実現する。

また、プログラムの解釈は、カンサスシティスタンダード(以下 KCS)により生成された音から行う。プログラムをバイナリに変換し、データ 0 は 1200Hz の正弦波を 16 回繰り返し、データ 2400Hz の正弦波を 32 回繰り返す。理論上はこれで 75bps の転送速度を実現し、8byte のデータ送信ごとに 1byte の誤り訂正を行うので、実働 66bps となる。

誤り訂正は 8byte のそれぞれが奇数なら 0、偶数なら 1 を出力し、8bit を集め読み込みと差異があれば読み込み失敗となる。音の入力には、雑音の排除を考えて 3.5mm のステレオミニジャックを用いる。

物理的な接続にはリレー回路を用いる。動作が早くなりすぎると打ち込みミスが発生すると予想できるので、必要に応じて入力スピード、ひいては音データのビットレートを調整する。

図 1,実装予定：4 ピンの入力を唯一の 16 つの出力に変換する回路

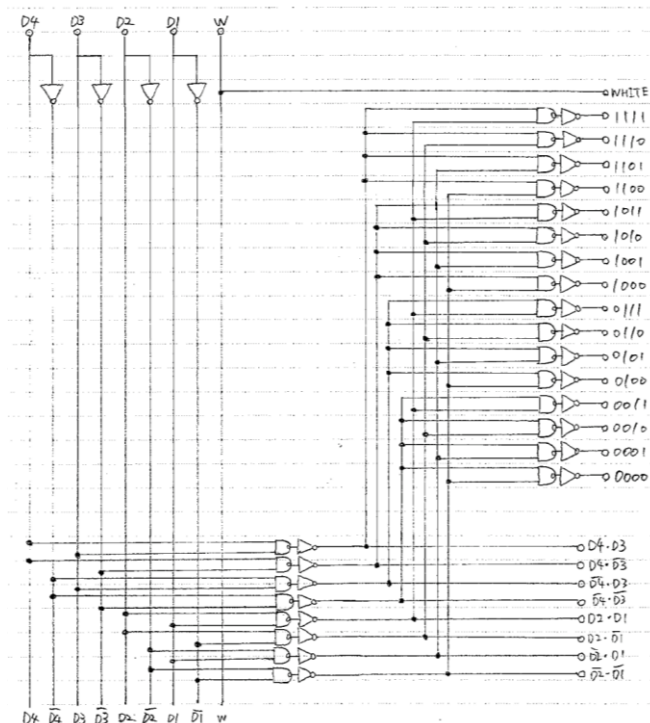
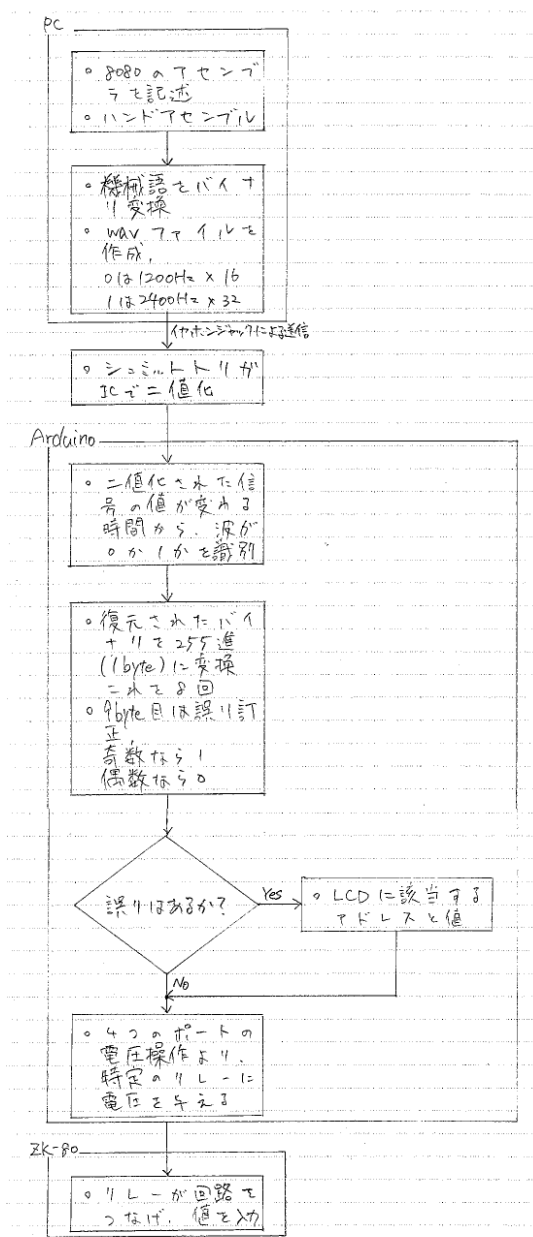


図2,全体のフローチャート



必要な機材（型番や個数，データシートの URL まで，可能な限り詳細に）：

はんだごて

マルチメータ

ユニバーサル基盤 多数

はんだ 多数

ジャンプワイヤ 多数

スズめっき線 多数

ピンヘッダ 多数

Arduino uno, AE-ATMEGA-UNO-R3, 1 個

ZK-80 組み立てキット, [ATD-B-001], 1 個

<https://www.aitendo.com/product/18528>

LCD 16x2 文字, SC1602BSLB-XA-GB-K, 1 個

[https://akizukidenshi.com/download/ds/sunlike/SC1602BSLB-XA-GB-K\\_20181119.pdf](https://akizukidenshi.com/download/ds/sunlike/SC1602BSLB-XA-GB-K_20181119.pdf)

3.5mm ステレオミニジャック DIP 化キット, AE-PHONE-JACK-DIP, 1 個

3.5mm オーディオケーブル, 1 本

シュミットトリガインバータ, TC74HC14AP, 1 個

[https://akizukidenshi.com/download/ds/Toshiba/TC74HC14AP\\_datasheet\\_ja\\_20140301.pdf](https://akizukidenshi.com/download/ds/Toshiba/TC74HC14AP_datasheet_ja_20140301.pdf)

NAND ゲート, TC74HC00AP, 6 個

<https://akizukidenshi.com/download/ds/toshiba/TC74HC00A.pdf>

NOT ゲート, TC74HC04AP, 6 個

[https://akizukidenshi.com/download/ds/Toshiba/TC74HC04AF\\_datasheet\\_ja\\_20140301.pdf](https://akizukidenshi.com/download/ds/Toshiba/TC74HC04AF_datasheet_ja_20140301.pdf)

リレー, Y14H-1C-5DS, 17 個

<https://akizukidenshi.com/download/ds/hsinda/y14h.pdf>

評価指標（「目的」を達成できたか否かを判断する指標（数値）は何か？ 具体的に示す）

手入力でのデータ入力速度, 誤入力率と, 実験で制作したものの入力速度, 誤入力率を比較し, 十分に後者の方が秀でていることを確認する. 確認すると一人が値を目で読み, 手で入力する方式だと 3bps 程度, 誤りは 50byte に 1 回程度, もう一人や読み上げソフトにメモリの値を順番に読み上げさせ, それを耳で聞き, 手で入力する方式は 5bps 程度, 誤りは 40byte に 1 回程度であった.

達成目標（「目的」を達成できたと判断できる評価指標の目標値はいくつか？ 具体的に示す）

本実験での目標は, 75bps 以上の通信で, 誤りは 1000byte に 1 回より少ない値を目指す.

評価実験計画（上記の評価指標をどのように計測するのか？ 実験計画を可能な限り詳細に説明）

転送速度に関しては,プログラムの音声を作成した時点で確定する.特定のボーレートでの入力に対し,8byte のデータと 1byte の誤り訂正を通信したらすぐにリレーに電圧をかけるため,入力と音声のビットレートは一致する.

以下 16 進数の整数については末尾に H をつける.

誤りについては,アドレスの一致不一致にて計測する.ZK-80 はアドレス 8000H から 83FF までの RAM を実装している.このうち一部メモリはプログラムとモニタワークエリアで占有するため,以下のようなメモリ配置にて誤入力率を算出する.

まず,ランダムな 00H から FFH(16 進数で 255)までの乱数を 255 個用意する.これをメモリアドレス 8000H から 80FFH に実験で作成したデバイスを用い入力する.8000H から 80FFH に入力した乱数をそのまま 8100H から 81FFH にも入力する.つまり 8000H と 8100H や 8023H と 8123H には同じ値を入力させる.この対応する 2 つのアドレスの値が一致するかしないかのプログラムを 8200H から記述する.スタックポインタやデータの配置は 8000H から 81FFH の領域を使用しないようコーディングする.

図 3, データ配置と一致の判定

adrs	8000H	8001H	8002H	8003H	...	80FFH	8100H	8101H	8102H	8103H	...	81FFH
value	79	F5	EE	C1	...	85	79	F5	1E	C1	...	85
結果	一致	一致	不一致	一致	...	一致						

スケジュール（いつまでに何を達成するのか？）

週	作 業 予 定
第 5 週	素子の調達 データシートの理解
第 6 週	Arduino,ZK-80 間のキーボード回路の作成
第 7 週	シリアル通信によるテキスト送信のプログラム作成と動作した回路の評価
第 8 週	バイナリデータを音に変換するプログラム作成
第 9 週	音をバイナリに変換するプログラムの作成，動作した回路の評価
第 10 週	ハードウェア,ソフトウェアの改良,考察
第 11 週	スケジュール遅延のためのマージン
第 12 週	最終報告書提出・成果発表会