情報工学実験 I 報告書

	点

実験クール (1・2・3・4)	3	実験回 (1・2・3)	3	クラス (A・B)	B

実験番号 4　実験題目 **AD/DA 変換**

グループ (A・B・C・D)	B	学籍番号	EP20050	氏名	小池 正基

	実験実施日	実験項目（実施概要）
1	2023／05／30	アナログ/デジタルについて，AD/DA 変換手法
2	2023／06／06	AD 変換のプログラム作成，実行
3	2023／06／13	ADC，DAC を用いた LED への印加電圧制御プログラム作成，実行

共同実験者（学籍番号）

中間①提出日	中間②提出日	完成提出日	完成再提出①	完成再提出②
2023／06／05	2023／06／12	2023／06／16	／　／	／　／

コメント

目 次

1 目的

本実験演習は，AD/DA変換に関するものである．DA変換は，デジタル信号をアナログ信号に変換する回路であり、AD変換はその逆の動作をする回路である．基本となる電気電子回路について学び，続いて実際の演習によりそれぞれの動作を体験学習し，知見を深めることを目的とする．

2 実験事例

AD/DA 変換に関する，学生実験の事例として次の物をとりあげる．
タイトル：AD/DA 変換 著者：大城和也
url:https://ie.u-ryukyu.ac.jp/~e045713/files/ie_lab2/report4.pdf

本実験報告書の概要は次の通りである．
本実験は，AD 変換についての仕組みを学ぶため，マイコンに AD 変換を行うプログラムを実装し，結果を確認した後，AD 変換における手法について調査を行ったレポートである．学生にはあらかじめいくつかのミスが含まれた状態のサンプルプログラムが配布され，そのプログラムを修正・実行することによってコンパイルが可能となる．具体的なプログラムの内容について解説する．実験に用いるプログラムは，特定のマイコンボードに対し入力された電圧をデジタル値に変換し，変換結果を PC 上に出力するといった処理内容となっている．このプログラムによって電圧を取得するマイコンはライブラリ¡iso32.h¿から，32 ビットの I/O ポートを持つものであることが推測できる．具体的な手順としては，最初にサンプリングモード，クロック，サンプリング回数などの設定データを定義し，サンプリングを開始することで入力ポート 0x240 に入力されるアナログ電圧値をマイコン内で AD 変換し，デジタルデータとしてバッファに取得，バッファ内のデータを読み取り配列に格納し，格納されたデジタル値を再度アナログ形式で表せるように処理してから PC 上に表示するといった流れである．このようなプログラムによって入力された電圧を PC 上に表示することができており，実験は成功したと言える．

3 初回所感

今回の AD/DA 変換については，ある程度の手順は知っていたものの，具体的にどのような構造の機器で AD/DA 変換が行われているかは覚えていなかったので，ぜひ覚えておきたいと思った．私は何度か基板を自作しており，電子回路の設計やマイコンのプログラミングを何回かしているのですが，電子系の知識不足で苦労したので，このような理論もプログラマにとって大切だと考えている．

4 AD 実験の目的

本実験は，AD 変換処理に関して，その動作を実体験しながら使用方法について学ぶことを目的とする．また，測定結果について，表にまとめ，グラフに表すことについても練習をする．最後に，技術文書読解力向上を目的として，AD 変換に関する既存論文を読み，その要約を記述する．

5 測定結果

5.1 ボリューム No.32 の測定結果

測定値を、表 1 に示す．図 1 は測定値に関するグラフである．横軸は回転角，縦軸は抵抗比である．この測定結果から，No.32 のボリュームの特性はBカーブ型であると推定できる．

表 1: No.32 のボリュームの計測結果

回転角	抵抗比
0	0
1	0
2	8
3	19
4	30
5	41
6	53
7	64
8	76
9	93
10	100

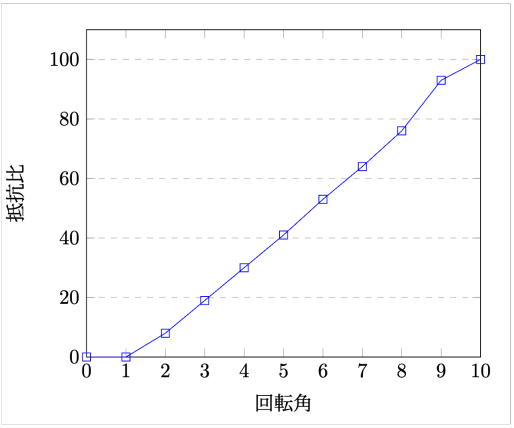

図 1: No.32 のボリュームの特性

5.2 ボリューム No.20 の測定結果

測定値を、表 2 に示す．図 2 は測定値に関するグラフである．横軸は回転角，縦軸は抵抗比である．この測定結果から，No.20 のボリュームの特性はAカーブ型であると推定できる．

表 2: No.20 のボリュームの計測結果

回転角	抵抗比
0	0
1	0
2	8
3	19
4	30
5	41
6	53
7	64
8	76
9	93
10	100

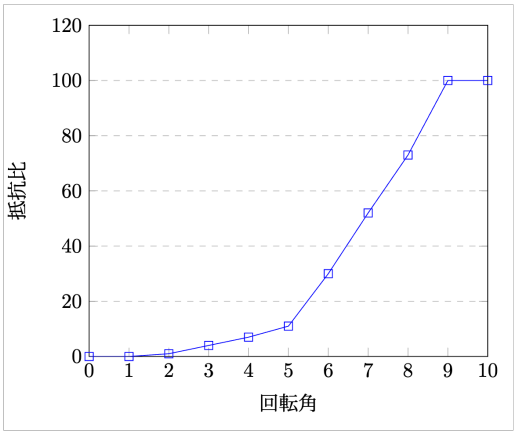

図 2: No.20 のボリュームの特性

6　処理プログラムについて

AD 変換値から抵抗比を求めるために，以下の式を用いている．
val=((unit32_t)analogRead(ADC_PIN)*RATIO)/MAX_ADC_VALUE;

まず，analogRead()を使って AD 変換値を読み取る．次に，RATIOを乗算している．ADC の最大値による除算以前にこれを行う理由は，先に乗算を行わないと小数点以下が切り捨てられてしまい，本来の値を計算することができないからである．

7　AD 変換に関する既存論文

AD 変換に関する論文について，以下を取り上げる．
ディジタルビジョンチップのためのソフトウェア A-D 変換手法，鏡 慎吾, 小室 孝, 藤村 英範, 石川 正俊掲載先 URL:https://www.jstage.jst.go.jp/article/itej1997/57/3/57_3_385/_article/-char/ja/
アブストラクト：
A pixel-level analog-to-digital (A-D) conversion method suitable for digital vision chips is proposed. In this method, each quantization boundary is compared by determining whether the signal level integrated for a certain time has reached a certain voltage, where both the time and voltage are properly controlled by software for each boundary. We derive an algorithm to generate an optimal controlling schedule to realize an arbitrary A-D conversion scale subject to the condition that noise is minimized. Experimental results have shown that images with given A-D characteristics can be obtained with low noise.

以下に本論文の要約を記述する．
本論文では，汎用的な視覚情報を取得できる 1 手法である，デジタルビジョンチップにおいて用いられている CMOS イメージセンサの AD 変換を効率化する手法を提案するものである．特に，CMOS イメージセンサの持つ特徴である高速性を保持するため，AD 変換の正確性だけでなく，変換の効率も視野に入れ，総合的な性能の向上を図っている．
まず，過去に行われた研究として，一定時間露光した際の信号レベルをを外部から与えるランプ電圧と比較し，ビットごとに変換を行うといったタイプの AD 変換器をイメージセンサ 4 つに 1 つ搭載し，複数回の露光結果を合成することで高精度の撮像を行っている．しかし，この方法では最初に複数回の露光が必要になるため，効率が悪い．
本論文で提案された手法においては，イメージセンサが露光したとき，照射量に応じて電荷が放電され，比較器の出力が論理値 0 から論理値 1 に変化する現象を利用し，比較器の出力が変化するまでの時間 t から光量を計測する．しかし，AD 変換に割かれる時間には限りがあるため，光の検出下限をできるだけ小さく，ノイズを抑えなければならない．そのため，参照電位をプログラムによって動的に制御する．このときのアルゴリズムは，入力として光の照射によって比較器の出力が変化する光量であり，出力は参照電位と読み出しにかかる時系列である．
以上で述べた理論を実際の機器を用いて実験を行う．実験の概要は撮像面に LED 照明による一様な光を投射し，計測値のばらつきを測定し，評価するといった形式で行った．時間計測の制度を高めるため，露光時間を 0 から 200ns ずつ 1ms まで変化させ，5000 回撮像した．結果，同じ照度に対し，参照電位の値が低いほうがノイズ量が抑えられており，参照電位を制御することで撮像におけるノイズを低減することができるとわかる．

次に，提案手法である動的な参照電位の変更を取り入れて実験を行う．提案手法と既存手法で撮像結果を比較した結果，既存手法では参照電位が一定であるためノイズの影響が大きかったり，暗部を撮影できなかったりで正常に撮像できなかったのに対し，提案手法ではそのようなノイズがない状態で撮像できていた．以上より，提案手法である参照電位を動的に制御することがより効率的な AD 変換に有効であることが実証された．

8　2回目所感

今回の実験では，AD 変換を行うプログラムを用いて，アナログボリュームの抵抗の変化を確認し，AD 変換における論文の調査を行った．ボリュームの抵抗値を計測する実験は早く終わったが，論文の調査に時間がかかってしまった．このような論文を読む機会は多くなっていくと思われるので，長文を読む練習，理論を理解する練習を怠らないでおきたい．

9　DAC 実験の目的

LED は，ある一定の電圧を越えないと電流が流れ始めないという性質がある．この電圧を順方向電圧という．本実験では，DAC を使って LED に印加する電圧を変化させ，目視による点灯を観察することで，順方向電圧の，おおよその値を推定することを目的とする．

10　実験方法

1. DAC を 0 から 5 ずつ増加させていき，LED が点灯開始する値を求める．
2. 1 で求めた値より 5 小さい値から，DAC を 1 ずつ増加させていき，LED が点灯開始する値を求める．
3. 求めた値から，LED に印加した電圧（＝ DAC の出力電圧）を求める．
   DAC の出力電圧は，$\frac{\text{基準電圧} \times DAC\ \text{値}}{DAC\ \text{最大値}}$ で求められる
4. 分子の基準電圧は，3.3V，分母の DAC 最大値は256である．
5. 計算値を求めるにあたっては，小数点以下の切り捨て処理に注意する．

11　実験結果

測定値は107であった．このことから，実験に使用した LED の順方向電圧は，1.4V 付近と推定できる．

```
#include <M5Stack.h>
#define DISP_SIZE 3
#define DISP_XPOS 20
#define DISP_YPOS 40
#define DAC_PORT 26
#define DAC_MIN 100
#define DAC_MAX 255
#define DAC_STEP 1

#define MEASURE_CYCLE 500
#define RATIO 127
#define MAX_ADC_VALUE 4095
#define ADC_PIN 36

uint8_t u8DacValue = 0;
void setup() {
  M5.begin();
  M5.Lcd.fillScreen( BLACK );
  M5.Lcd.setTextColor( WHITE, BLACK );
  M5.Lcd.setTextSize( DISP_SIZE );
  u8DacValue = DAC_MIN;
  M5.Lcd.setCursor( DISP_XPOS, DISP_YPOS );
  M5.Lcd.printf("DAC␣=␣%3u", u8DacValue );
  dacWrite( DAC_PORT, u8DacValue );

  pinMode(ADC_PIN, INPUT);
  return;
}


void loop() {
  uint8_t u8PreValue = u8DacValue;
  uint32_t val;
  val = (uint32_t)analogRead(ADC_PIN); //get value(0~4096)
  u8DacValue = ((val * RATIO) / MAX_ADC_VALUE) + 128; //reshape value(128~255)
  if ( u8DacValue > DAC_MAX ) {
    u8DacValue = DAC_MAX;
  }
  if ( u8DacValue != u8PreValue ) {
    M5.Lcd.setCursor( DISP_XPOS, DISP_YPOS );
    M5.Lcd.printf("DAC␣=␣%3u\nRatio␣=␣%3u", u8DacValue, val);
    dacWrite( DAC_PORT, u8DacValue );
  }
  delay( MEASURE_CYCLE );
  return;
}
```

図 3: ADC と DAC の組み合わせ動作

12 ADC と DAC の組み合わせ動作

図 3 に ADC と DAC の組み合わせ動作を行うプログラムを示す．上記のプログラムは，アナログボリュームによって LED に流す電圧を制御するプログラムである．しかし，アナログボリュームから入力される抵抗値は 0〜4095 の範囲であり，出力値は 128 〜 255 の範囲にしなければならないため工夫が必要である．当プログラムでは，ADC_PIN に接続されたアナログボリュームの抵抗値を 34 行の val = (uint32_t)analogRead(ADC_PIN); によって取得し，その次の行 u8DacValue = ((val * RATIO) / MAX_ADC_VALUE) + 128; で DAC 値に変換し，LED に出力している．このように値の取り得る範囲内の最大値で割り，変換後の値の取り得る値の最大値をかけることで値の範囲変換を行うことができる．

13 2 個の LED の交互点滅プログラム

図 4 に 2 個の LED の交互点滅を行うプログラムを示す.

```
#include <M5Stack.h>
#define DISP_SIZE 3
#define DISP_XPOS 20
#define DISP_YPOS 40
#define DAC_PORT_A 25
#define DAC_PORT_B 26

#define DAC_MIN 130
#define DAC_MAX 255
#define DAC_STEP 1
#define MEASURE_CYCLE 10

uint8_t u8DacValue_A = 0;
uint8_t u8DacValue_B = 0;
void setup() {
  M5.begin();
  M5.Lcd.fillScreen( BLACK );
  M5.Lcd.setTextColor( WHITE, BLACK );
  M5.Lcd.setTextSize( DISP_SIZE );
  u8DacValue_A = DAC_MIN;
  u8DacValue_B = DAC_MAX;
  M5.Lcd.setCursor( DISP_XPOS, DISP_YPOS );
  M5.Lcd.printf("DAC_A␣=␣%3u\nDAC_B␣=␣%3u", u8DacValue_A, u8DacValue_B );
  dacWrite( DAC_PORT_A, u8DacValue_A );
  dacWrite( DAC_PORT_B, u8DacValue_B );

  return;
}


void loop() {
  float u8PreValue_A = u8DacValue_A;
  float u8PreValue_B = u8DacValue_B;
  float st = DAC_STEP;
  u8DacValue_A += st;
  u8DacValue_B -= st;
  if (u8DacValue_A == 230 || u8DacValue_B == 230) {
    st *= -1;
  }
  if ( u8DacValue_A != u8PreValue_A || u8DacValue_B != u8PreValue_B ) {
    M5.Lcd.setCursor( DISP_XPOS, DISP_YPOS );
    M5.Lcd.printf("DAC_A␣=␣%3u\nDAC_B␣=␣%3u", u8DacValue_A, u8DacValue_B);
  }
  dacWrite( DAC_PORT_A, u8DacValue_A );
  dacWrite( DAC_PORT_B, u8DacValue_B );

  delay( MEASURE_CYCLE );
  return;
}
```

図 4: 2 個の LED の交互点滅

上記のプログラムは，1 秒ごとに 2 個の LED の片方を明るく，片方を暗くし，交互に点滅させるプログラムである．DAC 値の範囲は 130̃230 であるため，1 秒間で 100 だけ出力を増減させる必要がある．そこで，当プログラムはそれぞれの DAC 出力値をループごとに 1 ずつ増加・減少させる処理を 1 秒間で 100 回，つまり 10ms 間隔で繰り返すことによって実現した．片方の出力値が閾値を超えた場合，増加・減少値が反転し，明るくなっていた LED は暗く，暗くなった LED は明るくなる動作を開始する．

14　3 回通しての所感

3 回通して，AD/DA 変換の奥の深さをよく感じた．特に，変換手法については，パルス変調型，抵抗ラダー型を始めとしていくつもあり，論文からも AD/DA 変換の重要性と研究の活発さを実感した．私は電子系の造形が無いため全てを理解することは難しいが，プログラマとして機械，電子系の職業の方々と共同で働くときに，円滑に業務を進められるような人材になれるよう，今回学んだことを始めとした電子系の技術についてよく理解しておきたいと思った．