

PIC を用いた LED の点灯回路の政策とその外部割り込み実験

筒居 稔

2017 年 7 月 9 日

1 目的

PIC を用いて LED 点灯回路を制作し、その外部割り込み実験とその応用を行う。Arduino 等のマイコンによる電子制御回路を制作し、その動作確認実験を行う

2 原理

2.1 マイコンの種類

2.1.1 PIC

PIC とは、ペリフェラル インターフェース コントローラーの略称であり、マイクロチップ・テクノロジー社が製造している。CPU・メモリ (RAM,ROM)、I/O などが 1 チップに全て収められている。ROM に書き込まれたプログラムによって制御する。プログラムコード用の内蔵メモリは、古くはワンタイム ROM、EPROM(紫外線消去) 品があったが、現在では大多数がフラッシュ ROM 品となっている。

2.1.2 H8

H8 は日立製作所 (現在はルネサス エレクトロニクス) が開発したマイクロコントローラである。製品としては CPU コアに ROM、RAM、割り込みコントローラ、タイマ、入出力ポート、シリアルコントローラ (SCI)、A/D コンバータ、D/A コンバータ、DMA 等が統合されたパッケージで販売される。開発言語は C/C++/アセンブラである。

2.1.3 AVR

AVR とは、Atmel 社が製造している、RISC ベースの 8 および 32 ビットマイクロコントローラ (制御用 IC) 製品群の総称である。PIC 同様に回路構成が簡単で CPU、メモリ (RAM、ROM)、I/O、データ記憶用の EEPROM、クロック発信機、タイマ等が 1 チップに収められており、書き込まれたプログラムによって制御される。

2.2 PIC16F84A

PIC16F84A とはマイクロチップのフラッシュマイコンである。1k バイトのメモリを搭載しており、1000 回程度プログラムを即時消去し、簡単に書き換えられる。I/O ピン数は 13 であり、ピンごとに入出力設定が可能である。

2.3 Arduino とは

Arduino とは、Atmel AVR マイコンチップを実装した基板と開発システムから構成される、オープンソースハードウェアの一つである。開発システムは Arduino ホームページからダウンロードでき、開発環境もオープンソース・

マルチプラットフォーム対応なので Mac OS X や Linux でも開発できる。
開発環境では C++ 風の Arduono 言語によりプログラムを開発する。

2.4 部品について

2.4.1 セラミック発信機 (セラロック)

多結晶である電圧セラミックの機械的共振を利用した、固有の周波数で発振する電子部品である。主な用途は、マイクロプロセッサ等のデジタル回路におけるクロック信号源である。

2.4.2 発光ダイオード

ダイオードの一種であり、順方向に電圧を加えた際に発行する半導体素子である。

3 使用機器

表 1 使用機器

使用機器名	型番	メーカー
ノート PC	FMV-A 8260	FUJITSU
デスクトップ PC	CF-82	Pnasonic
Arduino		

4 方法

4.1 PIC 実験

PIC マイコン回路を作成し、サンプルプログラムを動作させて、オリジナルの改良を加えて実行する以下のプログラムを PIC マイコンに書き込み動作させた。

ソースコード 1 サンプルプログラム

```
#include <pic.h>
#include <../samples/delay/delay.h>
__CONFIG(HS & WDTDIS & UNPROTECT);
void DelayMs(unsigned char cnt)
{
    #if XTALFREQ <= 2MHZ
    do {
        DelayUs(996);
    } while (--cnt);
    #endif
    #if XTALFREQ <= 2MHZ
    unsigend char i;
    do {
        i = 4;
```

```

        do {
            DelayUs(250);
        } while (--i);
    }while (--cnt);

#endif
}

void main(void)
{
    unsigned char c;
    TRISA = 0x18;
    TRISB = 0x01;
    PORTA = 0x00;
    PORTB = 0x00;
    while (1) {
        while (1){
            if (RA3 == 0) break;
        }
        c = 5;
        while (1) {
            PORTA = 0x07;
            PORTB = 0xc0;
            DelayMs(250);
            DelayMs(250);
            PORTA = 0x00;
            PORTB = 0x3e;
            DelayMs(250);
            DelayMs(250);
            c = c-1;
            if (c == 0) break;
        }
        PORTB = 0x00;
    }
}
}

```

ソースコード 2 オリジナルプログラム

```

#include <pic.h>
#include <../samples/delay/delay.h>

```

```

_CONFIG(HS & WDTDIS & UNPROTECT);
void DelayMs(unsigned char cnt)
{
    #if XTALFREQ <= 2MHZ
    do {
        DelayUs(996);
    } while (--cnt);
    #endif
    #if XTALFREQ <= 2MHZ
    unsigned char i;
    do {
        i = 4;
        do {
            DelayUs(250);
        } while (--i);
    } while (--cnt);
    #endif
}

void main(void)
{
    unsigned char c;
    TRISA = 0x18;
    TRISB = 0x01;
    PORTA = 0x00;
    PORTB = 0x00;
    while (1) {
        if (RA3 == 0){
            RB1=1;
            RB2=1;
            RB3=1;
            RB4=1;
            DelayMs(2500);
            DelayMs(2500);
        }
        if (RB0 == 0){
            RB1=0;
            RB2=0;
            RB3=0;
            RB4=0;

```

```

    }
    /*while (1){
        if (RA3 == 0) break;
    }
    c = 5;
    while (1) {
        PORTA = 0x07;
        PORTB = 0xc0;
        DelayMs(250);
        DelayMs(250);
        PORTA = 0x00;
        PORTB = 0x3e;
        DelayMs(250);
        DelayMs(250);
        c = c-1;
        if (c == 0) break;
    }
    PORTB = 0x00;
}
*/
}
}

```

4.2 熱電対実験

熱電対の出力を AD 変換し、USB 経由でパソコンで電圧を測定する。

4.3 Arduino 実験

Arduino を用いて応用実験を行う。以下のプログラムを Arduino に書き込み実行した。

ソースコード 3 施錠確認プログラム

```

#include<Servo.h>
Servo myServo;

const int piezo = A0;
const int switchPin = 2;
const int yellowLed = 3;
const int greenLed = 4;
const int redLed = 5;

```

```

int knockVal;
int switchVal;

const int quietKnock = 10;
const int loudKnock = 100;

boolean locked = false;
int numberOfKnocks = 0;

void setup(){
  myServo.attach(9);
  pinMode(yellowLed, OUTPUT);
  pinMode(redLed, OUTPUT);
  pinMode(greenLed, OUTPUT);
  pinMode(switchPin, INPUT);
  Serial.begin(9600);
  digitalWrite(greenLed, HIGH);
  myServo.write(0);
  Serial.println("hey");
}

void loop(){
  if(locked == false){
    switchVal = digitalRead(switchPin);
    if(switchVal == HIGH){
      locked = true;
      digitalWrite(greenLed, LOW);
      digitalWrite(redLed, HIGH);
      myServo.write(90);
      Serial.println("The box is locked!");
      delay(1000);
    }
  }

  if(locked == true){
    knockVal = analogRead(piezo);

    if(numberOfKnocks < 3 && knockVal > 0){
      if(checkForKnock(knockVal) == true){
        numberOfKnocks++;
      }
    }
  }
}

```

```

    }
    Serial.print(3-numberOfKnocks);
    Serial.println(" more knocks to go");
}

if(numberOfKnocks >= 3){
    locked = false;
    myServo.write(0);
    delay(20);
    digitalWrite(greenLed,HIGH);
    digitalWrite(redLed,LOW);
    Serial.println("The box is unlocked!");
}
}
}

boolean checkForKnock(int value){
    if(value > quietKnock && value < loudKnock){

        digitalWrite(yellowLed , HIGH);
        delay(50);
        digitalWrite(yellowLed , LOW);
        Serial.print(" valid knock of value ");
        Serial.println(value);
        return true;
    }
    else{
        Serial.print("Bad knock value ");
        Serial.println(value);
        return false;
    }
}
}

```

5 結果

5.1 PIC 実験

以下にサンプルプログラムとオリジナルプログラムの結果を示す。図 1 はサンプルプログラムの結果である。スイッチを押すとすべての LED (写真の回路はハンダ付けがうまくできなかったため一部の LED 以外) が点灯する。スイッチから手を離すと消灯する。

図 2、図 3 はオリジナルプログラムの結果である。スイッチを押すと任意の LED が点灯し、スイッチを離しても点灯し続ける。スイッチ 2 を押すと点灯していた LED が消灯する。

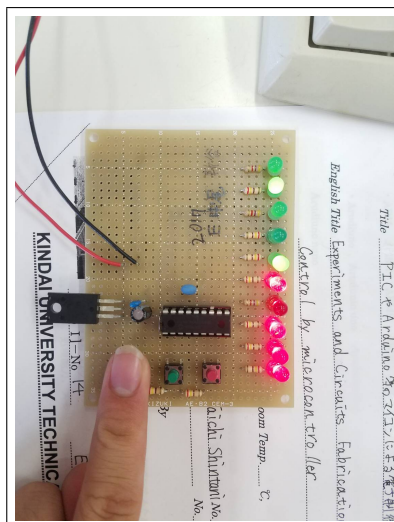


図 1 サンプルプログラム結果

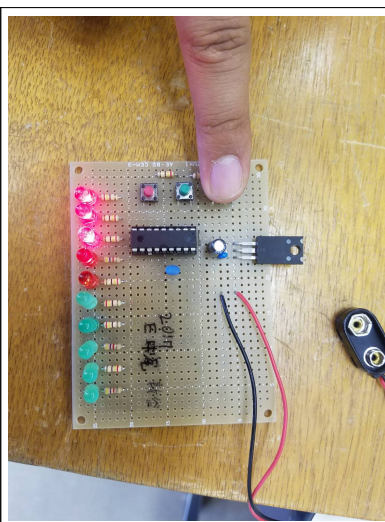


図 2 オリジナルプログラム

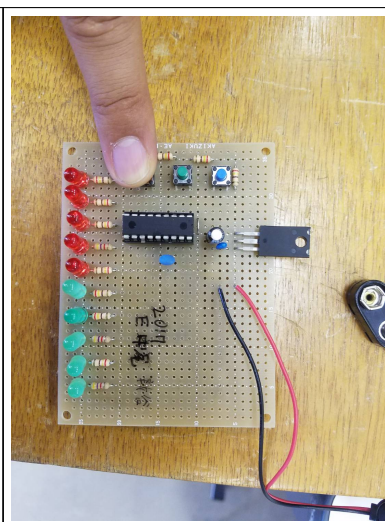


図 3 オリジナルプログラム 2

5.2 熱電対実験

測定した結果図 4～6 のような結果が得られた。図 4 は 0 度、図 5 は 24 度図 6 は 100 度の時の結果である。

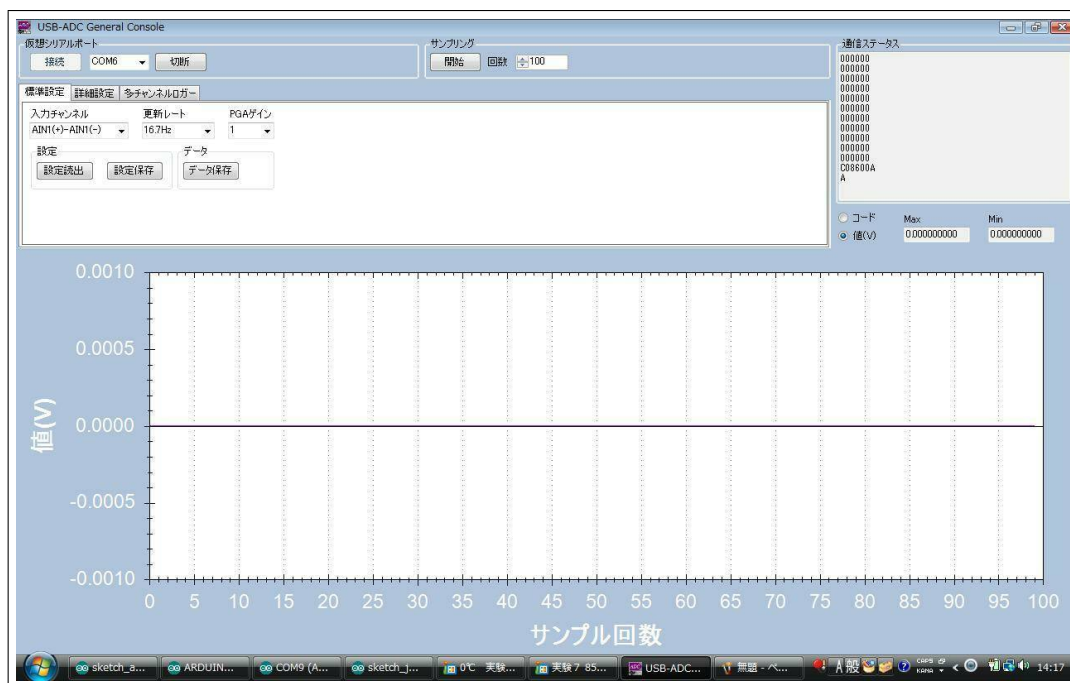


図 4 0 度

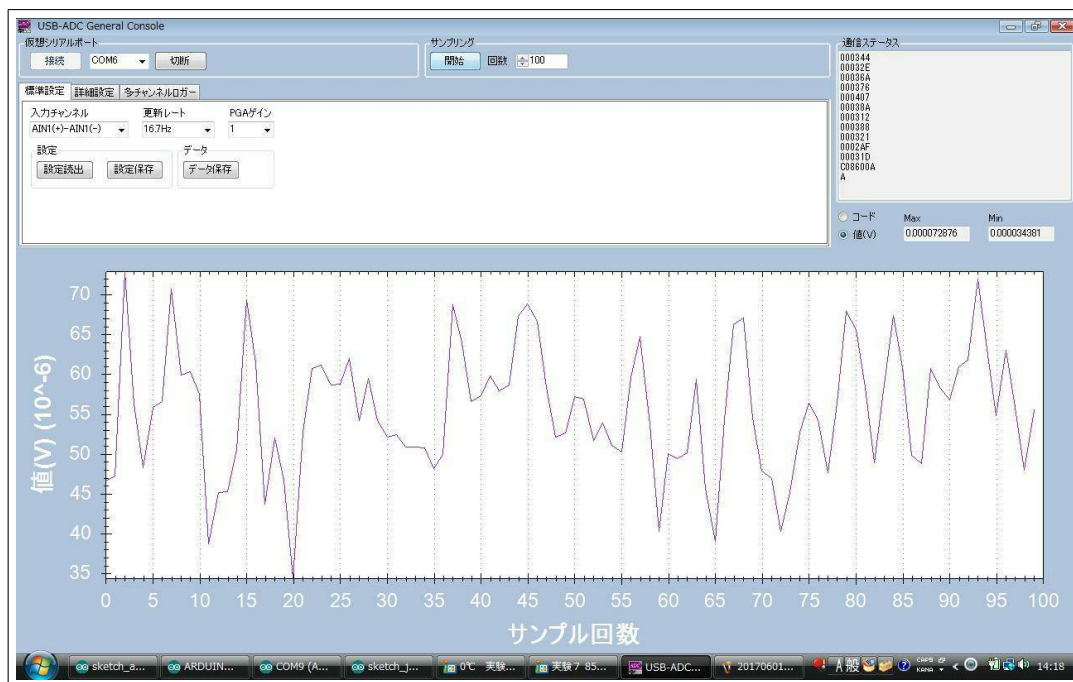


図 5 24 度

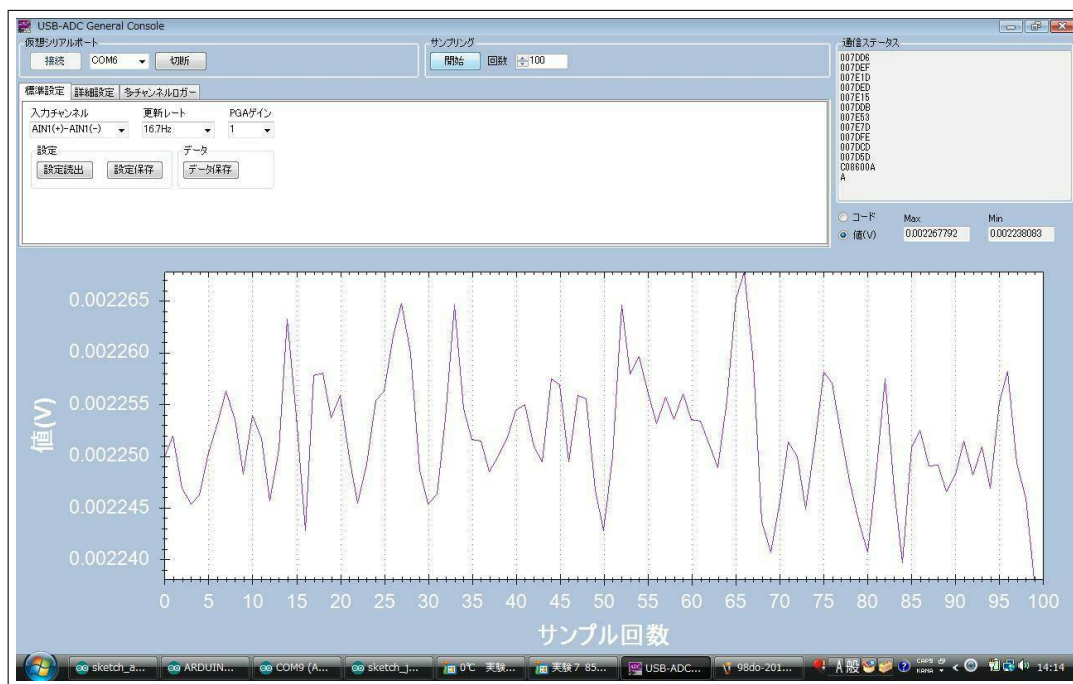


図 6 100 度

5.3 Arduino 実験

Arduino にソースコード 3 を書き込み実行した。

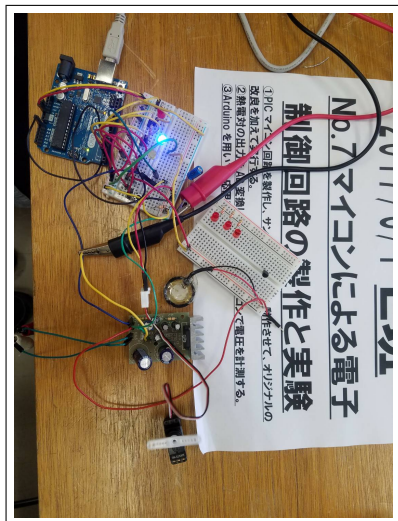


図 7 Arduino 結果 1



図 8 Arduino 結果 2

プログラムの内容はサーボモータにより箱をロックするものである。

図 7 は箱が閉じられている時、図 8 は箱が開いている時の結果である。

6 考察

6.1 PIC

ソースコード 1 とソースコード 2 のフローチャートを下に記す。

サンプルプログラムではスイッチ 1"RA3"を読み取り、入力が入れば while 文で LED を点灯させていた。

オリジナルプログラムではスイッチ 1"RA3"の他に、スイッチ 2"RB0"を使い、while 文の内部でスイッチ 1 と 2 の状態を読み取ることにより LED の点灯、消灯を行う。

6.2 熱電対

図 4~6 を比較すると、0 度 (図 4) では電圧は 0[V] 一定で変化は無かった。24 度 (図 5) と 100 度 (6) では互いに電圧が変化している。24 度では電圧の値が最大で約 0.00007[V]、最小で約 0.00003[V] であるのに対して 100 度では、最大最小ともに約 0.002[V] と温度が高くなるに連れて電圧が高く変化している。このことから温度が高くなると抵抗値が高くなると思われる。

6.3 Arduino

プログラム 3 では実行時は赤色 LED を点灯し 1 秒間ブザーを鳴らす。"numberOfknocks"でスイッチの入力の回数を判断し、もし 3 回入力が入れば緑色 LED を点灯しサーボモータを 90 度回し、箱の lock を解除しシリアルモニタに"The box is unlocke!"と出力する。もう一度スイッチを入力するとサーボモータが 90 度回転し、赤色 LED が点灯、1 秒間ブザーが鳴りシリアルモニタに"The box is locked!"と表示される。

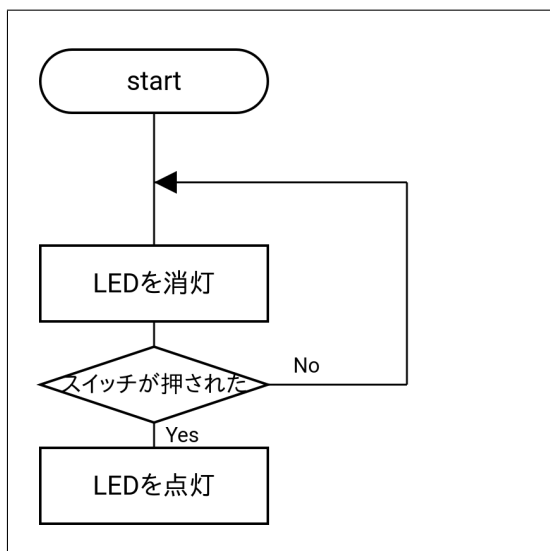


図 9 サンプルプログラムフローチャート

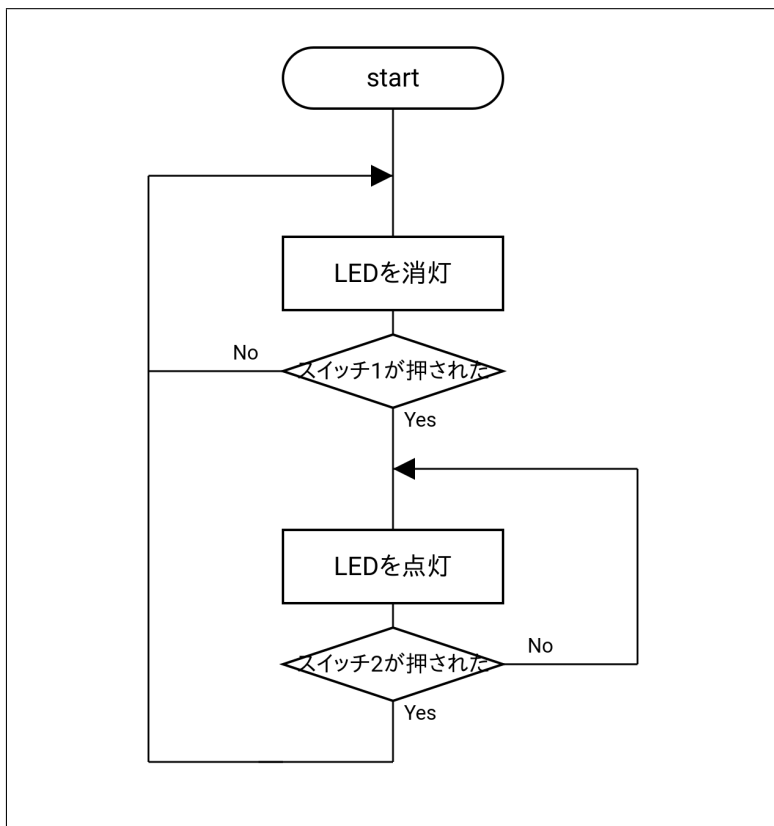


図 10 オリジナルプログラムフローチャート