目次

1 実験機材

マイコンボードの通し番号: 72
使用したパソコン: Lenovo Ideapad slim 550（持ち込み）
OS: Windows11 Home

2 理論

2.1 PWM

PWM は，周期一定の矩形波を出力し続けながら，パルス幅を変化させる機能である．本実験においては，サーボモーターの制御に用いられる．LaunchPad には最大 16 個の PWM 波形を生成する機能があり，各波形を自由に調整することができる．PWM は 2 系統存在し，各系統別に 4 つの PWM 生成器を持っている．各生成器は 2 本の出力ピンを制御することができ，内蔵 RGB LED を制御するためにはピンの割り当てを適切に行う必要がある．

2.2 I^2C

I^2C は，低速な周辺機器をマザーボードへ接続するのに用いられる通信規格である．SCL（クロック）と SDA（データ）の 2 本の信号線によるシリアル通信であり，この信号線上に複数の機器を接続することができる．複数の周辺機器は異なるアドレスが付与され，通信の最初にアドレスを指定する．送信と受信は同時に行わず時分割する．通信速度は遅いため高速に大量のデータをやりとりするような用途には向かないが，信号線の本数を劇的に削減できるメリットがあるため，速度の要求されない表示パネルなどに利用される．

2.3 マイコンボード

本実験で扱うボードは，Tiva C Series LaunchPad の EK-TM4C123GXL というボードである．このボードには 80MHz で動作するプロセッサ，256KB のフラッシュメモリ，32KB の SRAM が搭載されており，各種 I/O を制御することができる．また，システムクロックには 16MHz の水晶発振器を利用している．主な機能はすべて中央のプロセッサに内蔵されており，プロセッサから利用できるピンを周囲のヘッダに引き出した形になっている．

2.4 LED

LaunchPad には 3 色 LED が内蔵されており，この LED は GPIO ポート F に接続されている．LED を点灯させるには割り当てられた GPIO ピンに H(1) を，消灯するには L(0) を出力する．また，PWM の制御によって LED の光量や色合いを変えられる．

2.5 ブザー

圧電ブザーは GPIO ピンに接続されており，PWM 機能を利用して周期的に ON-OFF を繰り返すパルス波形を出力することによって音を鳴らす．一般的な音階の周波数 f は次の式で与えられる．

$$f = 440 \times 2^{\frac{n}{12}} \tag{1}$$

n の値を -9 から $+3$ と変化させることで音階を表せる．また，ブザーの ON と OFF の時間を同じにするためにはパルス幅を周期の半分として与えるなど，様々な工夫をすることで多種多様な音を表すことができる．

2.6 LCD

LCD とはキャラクタ液晶ディスプレイのことである．LCD を周辺機器として接続し，LCD のアドレスに適切な命令とデータを I^2C バスに流すことで LCD の設定や文字情報を制御することができる．

3 実験内容

3.1 【課題 1】仮想シリアルポートへの出力

プロジェクト blinky を CCS にインポートした．そして，仮想シリアルポートへの出力を Terminal Plugin によって受信するために，以下のように設定した．

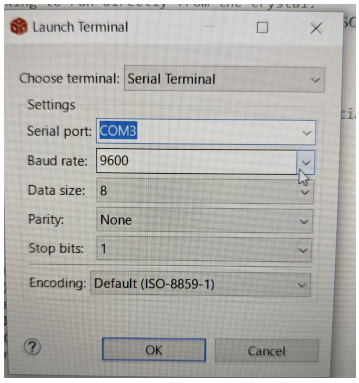

図 1: Terminal Plugin の設定

この設定の後，blinky_main.c の main() 関数内の UARTprintf() のコメントアウトを外してビルドを行い，仮想シリアルポートへの出力をターミナルを通して確認した．

3.2 【課題 2】blinky の改造

blinky_main.c の main() 関数内で GPIO 割り込みを有効化し，initInterruptPins() 関数と SW1PinIntHandler() 関数を実装し，SW1 の割り込みによって LED の点灯色を変化させるプログラムに改造した．また，このとき初期状態で有効化されていた Systick の割り込みを無効にした．

3.3 【課題 3】buzzer の改造

まず，未完成の buzzer.c において，音階を表す変数を受け取ってその音を鳴らす toneBuzzer() 関数と，呼び出すことで音が鳴り止む restBuzzer() 関数を実装した．そして，blinky_main.c で initInterruptPins() 関数を実装し，SW1PinIntHandler() 関数にて SW1 を押すことで音を鳴らしたり消したりするのに加えて，押した回数に従って「ド」から音階を上げて規定回数以降にまた「ド」に戻るようにプログラムを改造した．

3.4 【課題 4】setAddressLCD() と writeTextLCD() の実装およびプロジェクト lcd の書き換え

lcd_SB1602.c に，LCD の文字位置を指定する setAddress() 関数と，LCD への文字表示を実行する writeTextLCD() 関数を実装した．そして，lcd_main.c で直書きされている文字位置の指定と文字表示の実行のコードを setAddress() と writeText() を使うように改造した．

3.5 【課題 5】SW1 を押した回数の LCD への表示

まず，initInterruptPins() 関数を実装し，SW1PinIntHandler() 関数を，Systick に割り込みを無効化した後に SW1 の押下によってカウントが実行されて LCD へカウント文字列を表示するように実装した．また，main() 関数内で GPIO 割り込みを有効化した．

4 実験結果

4.1 【課題 1】仮想シリアルポートへの出力

仮想シリアルポートへの出力を確認するためにターミナルの設定をした後 UARTprintf() のコメントアウトを外すと，以下のようなメッセージがターミナルで確認できた．

4.2 【課題 2】blinky の改造

以下に，initInterruptPins() のソースコードを示す．

```
1 void initInterruptPins(void) {
2     // Clear Interrupt
3     GPIOIntClear(GPIO_PORTF_BASE,INT_ALL_BUTTONS);
4
5     // Register a handler function
6     GPIOIntRegister(GPIO_PORTF_BASE,SW1PinIntHandler);
```

```
7
8       // Set type of interrupt to falling edge
9       GPIOIntTypeSet(GPIO_PORTF_BASE,INT_ALL_BUTTONS,GPIO_FALLING_EDGE);
10  }
```

ソースコード 1: 【課題 2】における initInterruptPins 関数

この関数内では，3 行目にポート F に接続された全てのスイッチによる割り込みの処理を完了させ，6 行目でポート F に SW1PinIntHandler 関数を割り当てて割り込みハンドラを指定し，9 行目で SW1 が押されたとき，すなわち波形が L(0) のときに割り込みが発生するように設定した．

次に，SW1PinIntHandler() のソースコードを示す．

```
1   void SW1PinIntHandler(void) {
2       disableSW1PinInt();
3       // Clear Interrupt
4       clearSW1PinInt();
5       SysTickIntDisable();
6
7       GPIOPinWrite(GPIO_PORTF_BASE,base_led_color,0);
8
9       int light[]={LED_BLUE,LED_RED,LED_GREEN,LED_WHITE};
10      static int cnt=0;
11      base_led_color=light[(++cnt)%4];
12      GPIOPinWrite(GPIO_PORTF_BASE,base_led_color,led_color);
13
14      enableSW1PinInt();
15  }
```

ソースコード 2: 【課題 2】における SW1PinIntHandler 関数

この関数内では，2 行目に新たな SW1 の押下による割り込みを防ぎ，4 行目で SW1 による割り込みの処理を除去した．そして，6 行目で LED を一旦消灯させて，8 行目から 11 行目にかけて LED を点灯させるコードを加えた．初期状態が青でそこから赤，緑，白，青となるように，LED の色を表すマクロ定数を配列に格納し，static 宣言されたカウント用変数 cnt の値に応じて色を選択するようにした．これらの処理を終えたあと，13 行目で SW1 の押下による新たな割り込みを許可した．

4.3 【課題 3】buzzer の改造

4.4 【課題 4】setAddressLCD() と writeTextLCD() の実装およびプロジェクト lcd の書き換え

4.5 【課題 5】SW1 を押した回数の LCD への表示

5 考察

6 問題回答