w16: Microcontroller Experiments

Dates	@December 29, 2022
Topic	Sound control

Problem Description

基礎：利用操控蜂鳴器的頻率發出七個連續音階

進階：寫入歌譜，播放一首歌。

Code and Explanations

basic

```
#include "C8051F040.h"

void
Port_Configuration ()
{
  ...
}//end of function Port_Configuration

void
Timer_Configuration ()
{
  TMOD = 0x11;
  TCON = 0x50;
  CKCON = 0x10;

  IE = 0x8a;
  TL0 = 0xfd;
  TH0 = 0xeb;
  TL1 = 0x35;
  TH1 = 0xff;
}//end of function Timer_Configuration

void
Config ()
{
  ...
}//end of function Default_Config
```

這次Lab會使用到兩個timer，因此在Timer_Conguration中需要更改數值，並且加上TL1和TH1。

```
unsigned char status;
int count;
int half_period;
int countSec;
int i=0;

void Timer0_ISR ();
void Timer1_ISR ();

int song[8];

int
main (){
  Config ();
  status = 0x00;
  count = 0;
  countSec = 0;
  //half_period = 12;//set half period
  song[0]=18;
  song[1]=17;
  song[2]=15;
  song[3]=14;
  song[4]=12;
  song[5]=11;
  song[6]=10;
  song[7]=9;
  while (1) {
    P2 = status;
  }//end while (1)
}//end of function main
```

song是儲存半個週期中要數的clock次數，利用音階的頻率、clock的週期可以算出來，可是數字不會整除，所以聽起還來是有點不準，因此可以撥放出來之後再做微調。

```
void
Timer0_ISR () interrupt 1 //play tone by vary period
{
  count++;  //supposed 100us-per-count

  if (count==half_period) {
    count = 0;
    status = ~status;
  }

  TH0 = 0xff;
  TL0 = 0x00;
}//end of function Timer0_ISR
```

```
void
Timer1_ISR () interrupt 3 //play song, change tone per second
{
  countSec++;
  if (countSec==30000) {
    countSec = 0;
    count = 0;
    half_period = song[i];
    i=(i+1)%60;
  }
  TH1 = 0xff; //set interupt every 0.1 msec
  TL1 = 0x00;
}//end of function Timer1_ISR
```

Timer0負責音調、頻率的計數，Timer1則是負責數秒數，切換音階。Timer0的interrupt訊號為1，Timer1的interrupt訊號為3。

Bonus

```
#include "C8051F040.h"

void Timer0_ISR ();
void Timer1_ISR ();

int song[8];
int sheet[30] = {0,1,2,0,0,1,2,0,2,3,4,4,2,3,4,4,4,5,4,3,2,0,4,5,4,3,2,0};//老虎
// {0,0,1,2,2,1,0,1,2,0,2,2,3,4,4,3,2,3,4,2,7,6,5,4,2,7,6,5,4,3}; 蝴蝶

void
Timer0_ISR () interrupt 1 //play tone by vary period
{
  count++;  //supposed 100us-per-count

  if (count==half_period) {
    count = 0;
    status = ~status;
  }

  TH0 = 0xff;
  TL0 = 0x35;
}//end of function Timer0_ISR

void
Timer1_ISR () interrupt 3 //play song, change tone per second
{
  countSec++;
  if (countSec==30000) {
    countSec = 0;
    count = 0;
    tone = sheet[i];
    half_period = song[tone];
    i=(i+1)%28;
  }
  TH1 = 0xff; //set interupt every 0.1 msec
```

```
  TL1 = 0x00;
}//end of function Timer1_ISR
```

Bonus與Basic不同的就是Timer1的功能，Timer1要負責讀取sheet中的音調，然後根據此音調切換半周期計數的次數。

Difficulties and Solutions

▼ 撥放音樂時聲音非常魔幻，聽起來像是有兩種拍子。

尝試更改Timer1中的counterSec、TH1和TL1，counterSec大於一定數值就不會再發出聲音了，而TH1和TL1嘗試多種同學計算出來的組合總算能跑，至今不知道為何有些組合聽起特別神秘，猜測可能是上面Timer0的聲音頻率都還沒算好時就已經切換為下一種聲音頻率，因此TH1和TL1相差大一些就不會出錯。

Discussions

做這個Lab是個充滿歡樂的過程，尤其當我最後不管怎麼調音總是有一個音不在調上，又要demo一首歌時非常引人發笑，而在製作的過程中最令人挫折的是，這個Lab的原理不複雜，可是不知為何聲音就是發不出來，要使用以往沒有使用過的Timer也讓人不知所措，好險有同學的互相幫助和教授的細心指導。