

作業七：PWM 控制實驗 – 錄放音機

資工三乙 406262199 陳奕帆

資工三乙 406262216 劉品萱

繳交日期：_____

(1) 問題

Q1 : 何謂脈衝寬度調變(PWM -Pulse Width Modulation) ? 其應用為何?

是將類比訊號轉換為脈波的一種技術，一般轉換後脈波的週期固定，但脈波的工作週期會依類比訊號的大小而改變。

在類比電路中，類比訊號的值可以連續進行變化，輸入與輸出也呈線性變化。所以在類比電路中，電壓和電流可直接用來進行控制對象，例如應用在家用電器設備中的音量開關控制。

Q2 : Clock (時脈) 是一週而復始的訊號，請解釋頻率(frequency)1MHz 的定義?何謂 cycle time? 何謂 duty cycle?

Hz 表示每一秒週期性事件發生的次數。MHz 表示 10^6Hz

Cycle time 表示工作的速度 (所要計算產能的時段 / 週期時間)

Duty cycle 一個週期有數段 0 與 1，而 1 的時間總和占總時間的比例。

Q3 : 除頻器 (frequency eliminator) 作用為何 ?

將高頻除以一個常數，使頻率變低。

Q4 : WT58F2C8/WT58F2C9 32-bit Microcontroller 之 PWM Registers 對應的記憶體位址範圍。

PWM register 的 address range 為 0x0020_8C00 ~ 0x0020_8CFF

Q5 : The PWM module 可以產生幾個幾位元之 PWM output ? 請完整列出每個 PWM output 對應之 GPIO port 腳位。

PWM0 對應到 PORTE.6 GPIO	PWM1 對應到 PORTE.7 GPIO
PWM2 對應到 PORTE.8 GPIO	PWM3 對應到 PORTE.9 GPIO
PWM4 對應到 PORTE.10 GPIO	PWM5 對應到 PORTE.11 GPIO
PWM6 對應到 PORTC.10 GPIO	PWM7 對應到 PORTC.11 GPIO
PWM8 對應到 PORTC.12 GPIO	PWM9 對應到 PORTC.13 GPIO
PWM10 對應到 PORTC.14 GPIO	PWM11 對應到 PORTC.15 GPIO

Q6 : 請詳細解釋 PWM_EN Register 之使用方式與目的。

PWM_EN Register 設定為 1 的話為 啟用 PWM[x] function ; 設定為 0 則是停用 PWM[x] function

Q7：請詳細解釋 PWM_BAS_CLK Register 之使用方式與目的。

PWM 設置時間

00: system clock 01: symtem clock / 2

10: system clock / 3 11: system clock / 12

Q8：請詳細解釋 PWM_CLKX Register 之使用方式與目的, X 範圍為 0~11。

選擇 PWMX clock (PWM_CLK0~ PWM_CLK11)

Q9：請詳細解釋 PWMX Register 之使用方式與目的, X 範圍為 0~11。

Select duty cycle of PWM 0 output

0: PWM width = 0

1: PWM width= T pwm

2: PWM width= 2*T pwm ~ FFFF: PWM width= 65535*T pwm

Q10：請詳細解釋 PERIODX Register 之使用方式與目的, X 範圍為 0~11。

控制 PWMX 的輸出週期

Q11： 假設 PWM input clock = 12 MHz, if PWM base clock = 1 MHz = 1 us, 請問 PWM output clock 之 MAX clock 及 MIN clock 分別為多少?

PWM output clock = 1 us * (PERIOD+1) * (PWM_CLKx +1))

MAX clock = 2 us * 1 = 2us = 500KHz

MIN clock = 65536 us * 128 = 8388608us = 0.1192 Hz

Q12: 請列表說明七個音符 Do 、 Re 、 Mi 、 Fa 、 So 、 La 、 Ti 之高、中、低音的頻率對應關係。

	Do	Re	Mi	Fa	So	La	Ti
高音頻	1046	1175	1318	1397	1568	1760	1976
中音頻	523	587	659	698	784	880	988
低音頻	262	294	330	349	392	440	494

(2) C 程式碼—PWM 錄放音機程式。

```
1 int main()
```

```

2  {
3      OS_PowerOnDriverInitial();
4      unsigned int tmp = 0;
5      // 紀錄現在是否為錄音階段，0 為無錄音，1 為錄音
6      unsigned int RecordFlag = 0;
7      unsigned int i = 0;
8
9      DRV_Printf("=====
===\r\n", 0);
10     DRV_Printf("    ADP-WT58F2C9 PWM demo program    \r\n", 0);
11     DRV_Printf("=====
===\r\n", 0);
12
13     GPIO_PTA_FS = 0x0000;
14     GPIO_PTA_PADINSEL = 0x0000;
15
16     // Setting for 7LED select
17     GPIO_PTA_DIR = 0x0000;
18     GPIO_PTA_CFG = 0x0000;
19     GPIO_PTA_GPIO = Digit_8;
20     // Setting for 7LED number
21     GPIO_PTD_DIR = 0x0000;
22     GPIO_PTD_CFG = 0x0000;
23     GPIO_PTD_GPIO = Number_8 | Number_Dot;
24
25     float scale = 1;
26
27     unsigned int col;
28     unsigned int check = 0;
29     unsigned int key;
30     unsigned int index_7LED_NUM[17] = {Number_0, Number_1,
Number_2, Number_3, Number_4, Number_5, Number_6, Number_7,
31     Number_8, Number_9, Number_A, Number_b,

```

```

    Number_C, Number_d, Number_E, Number_F, Number_Dot, };
32     unsigned int RecordIndex = 0;
33
34     // 錄音存放的陣列
35     float RecordMusic[100];
36
37     DRV_Printf("PWM testing ...\r\n", 0);
38
39     GPIO_PTC_FS = 0xF000; //for PWM Function
40     PWM_EN = 0x000; //Disable all PWM function
41     PWM_BAS_CLK = 0x01;
42     PWM_CLK8 = 60;
43     PWM8 = 0xFF;
44     PWM_EN = 0x100; //PWM8
45
46     while(1)
47     {
48         key = 0xFF;
49         GPIO_PTA_DIR = 0x0FF0;
50         GPIO_PTA_CFG = 0x0000;
51         for (col=0; col<4; col++)
52         {
53             GPIO_PTA_BS = 0x000F;
54             GPIO_PTA_BR = 0x0000 | (1 << col);
55             tmp = ((~GPIO_PTA_PADIN) & 0xFF0) >> 4;
56             if (tmp > 0)
57             {
58                 if (tmp & 0x1)
59                     key = 0*4 + col;
60                 else if (tmp & 0x2)
61                     key = 1*4 + col;
62                 else if (tmp & 0x4)
63                     key = 2*4 + col;

```

```

64         else if (tmp & 0x8)
65             key = 3*4 + col;
66         else if (tmp & 0x80)
67             GPIO_PTA_GPIO = Digit_8;
68         else if (tmp & 0x40)
69             GPIO_PTA_GPIO = Digit_7;
70         else if (tmp & 0x20)
71             GPIO_PTA_GPIO = Digit_6;
72         else if (tmp & 0x10)
73             GPIO_PTA_GPIO = Digit_5;
74         break;
75     }
76 }
77 if (key != 0xFF)
78 {
79     PWM_EN = 0x100; //PWM8
80     if(!check)
81     {
82         GPIO_PTD_GPIO = index_7LED_NUM[key+1];
83         // Do
84         if(key == 0)
85         {
86             if(RecordFlag == 1)
87             {
88                 RecordMusic[RecordIndex] = 200000 /
(NOTE_C5 * scale);
89                 RecordIndex++;
90             }
91             PERIOD_8 = 200000 / (NOTE_C5 * scale);
92             PWM8 = (int) PERIOD_8 / 60;
93             delay1(300000);
94
95         }

```

```
96         // Re
97         if(key == 1)
98         {
99             if(RecordFlag == 1)
100             {
101                 RecordMusic[RecordIndex] = 200000 /
(NOTE_D5 * scale);
102                 RecordIndex++;
103             }
104             PERIOD_8 = 200000 / (NOTE_D5 * scale);
105             PWM8 = (int) PERIOD_8 / 60;
106             delay1(300000);
107
108         }
109         // Mi
110         if(key == 2)
111         {
112             if(RecordFlag == 1)
113             {
114                 RecordMusic[RecordIndex] = 200000 /
(NOTE_E5 * scale);
115                 RecordIndex++;
116             }
117             PERIOD_8 = 200000 / (NOTE_E5 * scale);
118             PWM8 = (int) PERIOD_8 / 60;
119             delay1(300000);
120
121         }
122         // Fa
123         if(key == 3)
124         {
125             if(RecordFlag == 1)
126             {
```

```

127             RecordMusic[RecordIndex] = 200000 /
(NOTE_F5 * scale);
128             RecordIndex++;
129         }
130         PERIOD_8 = 200000 / (NOTE_F5 * scale);
131         PWM8 = (int) PERIOD_8 / 60;
132         delay1(300000);
133
134     }
135     // So
136     if(key == 4)
137     {
138         if(RecordFlag == 1)
139         {
140             RecordMusic[RecordIndex] = 200000 /
(NOTE_G5 * scale);
141             RecordIndex++;
142         }
143         PERIOD_8 = 200000 / (NOTE_G5 * scale);
144         PWM8 = (int) PERIOD_8 / 60;
145         delay1(300000);
146
147     }
148     // La
149     if(key == 5)
150     {
151         if(RecordFlag == 1)
152         {
153             RecordMusic[RecordIndex] = 200000 /
(NOTE_A5 * scale);
154             RecordIndex++;
155         }
156         PERIOD_8 = 200000 / (NOTE_A5 * scale);

```



```
157         PWM8 = (int) PERIOD_8 / 60;
158         delay1(300000);
159
160     }
161     // Si
162     if(key == 6)
163     {
164         if(RecordFlag == 1)
165         {
166             RecordMusic[RecordIndex] = 200000 /
167             (NOTE_B5 * scale);
168             RecordIndex++;
169         }
170         PERIOD_8 = 200000 / (NOTE_B5 * scale);
171         PWM8 = (int) PERIOD_8 / 60;
172         delay1(300000);
173     }
174     if(key == 7)
175         PWM8 = 0;
176     // 高
177     if(key == 8)
178     {
179         scale = 2;
180         PWM8 = 0; // 按下的時候不發出聲音
181     }
182     // 中
183     if(key == 9)
184     {
185         scale = 1;
186         PWM8 = 0; // 按下的時候不發出聲音
187     }
188     // 低
```

```

189         if(key == 0xA)
190         {
191             scale = 0.5;
192             PWM8 = 0; // 按下的時候不發出聲音
193         }
194         // 錄音
195         if(key == 0xC)
196         {
197             PWM8 = 0; // 按下的時候不發出聲音
198             if(RecordFlag)
199                 RecordFlag = 0;
200             else
201                 RecordFlag = 1;
202         }
203         // 播放錄音
204         if(key == 0xD)
205         {
206             for(i = 0; i < RecordIndex; i++)
207             {
208                 PERIOD_8 = RecordMusic[i];
209                 PWM8 = (int) PERIOD_8 / 50;
210                 delay1(300000);
211             }
212         }
213         // 播放預設歌曲
214         if(key == 14)
215         {
216             while(1)
217             {
218                 PWM_CLK8 = 0x009;
219
220                 for ( i = 0; i < sizeof music/ sizeof music[0];
i++)

```

```
221         {
222             PERIOD_8 = 200000 / music[i][0];
223             PWM8 = ( int )PERIOD_8 / 20;
224             delay1( 250000 * music[i][1] );
225
226             PWM8 = 0;
227             delay1( ONE_SECOND/100 );
228         }
229         delay1(2000000);
230     }
231 }
232 }
233     check = 1;
234 }
235 else
236 {
237     PWM_EN = 0x000;
238     check = 0;
239 }
240 }
241
242     DRV_Printf("=====  

243     ===\r\n", 0);
244
245     return 0;
246 }
```