

```

1  /***** /
2  /* User Level #define Macros */
3  /* User Function Prototypes */
4  /***** /
5  /*
6  == PWM 0-199 ==
7  * PWM=係數 / (ADC(V*I)
8  * 電壓:47K與10K分壓，電流0.05Ω測試
9  265°C -- 2.95V * 1.3A = 3.835W 6728070.175(係數1)
10 335°C -- 3.7V * 1.625A = 6.0125W 10548245.61(係數2)
11 410°C -- 4.35V * 2.065A= 8.98275W 15759210.53(係數3)
12 */
13 #include <htc.h>
14 #include "DataTypes.h"
15 // #define poweroff_cc 500 // 25sec(50ms)
16 // #define poweroff_pc 100 // 5sec提醒
17 #define poweroff_cc 24000 // 20分(50ms)
18 #define poweroff_pc 200 // 10sec提醒
19 #define pwm_cpcc 3 // PWM 3次比較相同
20 #define longpress 2000 // 2sec(1ms)
21 #define key_scc 1000 // 輕按計時1sec(間格時間)
22 #define shortpress 10 // 20ms(1ms)
23 #define t125ms_cc 125 // 125mS
24 #define t50ms_cc 50 // 50mS
25 #define standby_cc 2000 // 1.25sec(standby flash)
26 #define standby_pcc standby_cc+950 // 電力開啟LEDD-ON 1sec
27 #define standby_fcc standby_cc-50 // standby LED-ON 125mS
28 #define pcode_adr 0
29 #define sp_adr 10
30 #define store_cc 5000 // 5sec後儲存資料(1ms)
31 #define pcode_c 0xa55a // power code
32
33 #define adc_sample_cc 16 //
34 #define average_sc 3 // 平均移位3次(8次平均)
35
36 #define current_ch 0b00001001 // AN2
37 #define voltage_ch 0b00001101 // AN3
38 // == define in/out port ==
39 #define GLED_p LATA0 // 綠色LED
40 #define RLED_p LATA1 // 紅色LED
41 #define key_p RA3 // key port
42 #define pwm_op LATA5 //
43 // bit0:綠燈，bit1:紅燈
44 const unsigned char led_table[] = { 0b0001,0b0011,0b0010 };
45 const float power_table[] = { 6728070.175,10548245.61,15759210.53 };
46 //////////////////////////////////////

```

```

47 vbit      adc_f=0;           // adc convert flag
48 vbit      adc_start;        // A/D開始取樣
49 vbit      adc_sync;         //
50 vbit      sample_f;         // ADC取樣結束
51 vbit      timebase_f;       // time-base flag(2mS)
52 vbit      t125ms_f;         // 125mS flag
53 vbit      f500ms_f;         // 500mS flash
54 vbit      f250ms_f;         // 250mS flash
55 vbit      fl25ms_f;         // 125mS flash
56 vbit      led_enable;       // led點亮翻新
57 vbit      store_f;          // 儲存旗號
58 vbit      power_f;          // 電源開啟(1)/關閉(0)
59 vbit      keylock_f;        //
60 vbit      remind_f;         // 關閉提醒
61 vbit      pwm_f;
62 vbit      second_f;         // 第二次輕按旗號
63
64 uint       sum[2];           // sum
65 uint       average[2];       // average
66 //uint      voltage;
67 //uint      current;
68 uchar      adc_sample_c;     // A/D取樣計數器
69 uchar      t125ms_c;         // 125mS time-counter
70 uchar      t50ms_c;          // 50mS time-counter
71 uchar      flash_c;          // 125mS-base
72 uchar      led_r;            // LED暫存器
73 uchar      led_fr;           // LED閃設暫存器
74 uint       key_count;        // 按鍵計數器
75 uint       key_sc;           // 第二次輕按計時器
76 uchar      power_select;     // 輸出功率選擇
77 uchar      history;          // 輸出功率選擇原存資料
78 uint       poweroff_c;       // 電源關閉計時器
79 uint       store_c;          // 儲存計時器
80 uchar      pwm_c;
81 uchar      pwm_r;
82 uchar      pwm_b;
83 uchar      pwm_h;
84 uchar      pwm_l;
85 uchar      pwm_cpc;          //
86 uint       standby_c;        //
87 float      factor;           // PWM係數
88 uint       pcode;
89
90 #define      voltage          average[1]
91 #define      current          average[0]
92 const uchar adc_channel[] = { 3,2 }; // AN3(voltage),AN2(current)

```

```
93 ////////////////////////////////////////////
94 void InitApp();                      // i/o port initize
95 void led();
96 void pwm_out();
97 void eeprom_write_nbyte(uchar adr,uchar *p,uchar lc);
98 void eeprom_read_nbyte(uchar adr,uchar *p,uchar lc);
99 void power_on();
100 void power_off();
101 void keybroad();
102 void power_set();
103 uchar pwm_set();
104 void adc_clear();
105 void adc_sub();
106 void store();
```