考古題

2024

1a 按鈕切換mode，燈泡循環

1b UART輸入(state num)切換mode，燈泡倒數

2a 按鈕切換mode，馬達角度內來回轉

2b 按鈕切換mode，馬達角度(45-90-180)變化

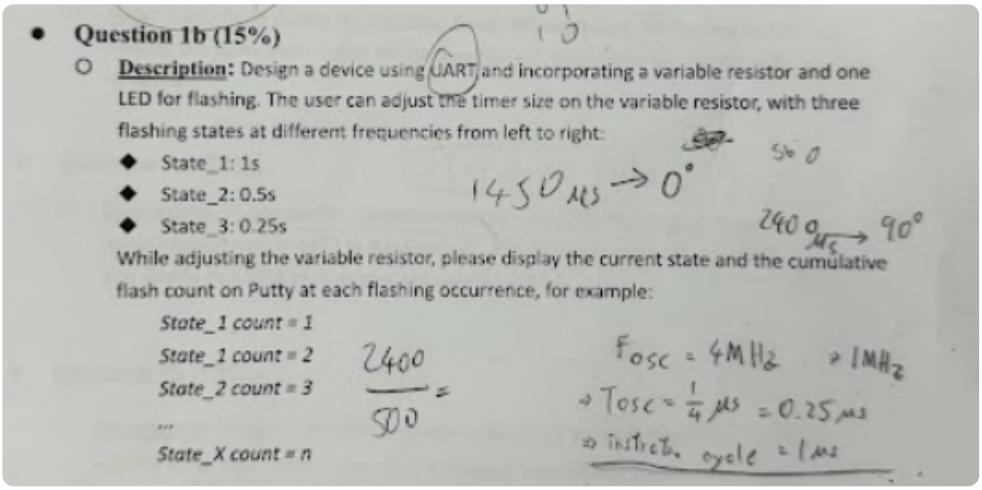
3a ADC分區間state，控制LED

3b UART顯示:mode1 現在燈泡情況(LD)， mode2 偵測輸入X，改變state並顯示

2023

1a ADC控制燈泡state延遲

1b



- **Question 1b (15%)**
  - **Description:** Design a device using UART and incorporating a variable resistor and one LED for flashing. The user can adjust the timer size on the variable resistor, with three flashing states at different frequencies from left to right:
    - State_1: 1s
    - State_2: 0.5s
    - State_3: 0.25s

  While adjusting the variable resistor, please display the current state and the cumulative flash count on Putty at each flashing occurrence, for example:

  State_1 count = 1
  State_1 count = 2
  State_2 count = 3
  ...
  State_X count = n

1450 μs → 0°
2400 μs → 90°
2400 / 500 =
Fosc = 4MHz → 1MHz
→ Tosc = 1/4 μs = 0.25 μs
⇒ instruct. cycle = 1 μs

2a 按鈕控制mode，馬達來回轉

2b UART控制mode，馬達轉速要不同

3a ADC控制馬達0-90度

3b UART控制除數，顯示(ADC/除數)的值

2021

1... ADC控制馬達

2... UART:motor 顯示角度， degree可輸入更改角度

1a(1) UART: mode1 輸入n按鈕每次控制馬達轉動角度(180/n)

1b(1) UART: mode2 輸入n按鈕啟動馬達轉動角度

2a(2) 按鈕控制燈泡亮暗

2b(2) UART 更新timer:00

3a(3) ADC控制馬達區間，並顯示燈泡

3b(3) UART:mode1 呈3a並顯示目前角度, mode2 輸入角度後並呈3a轉到該角度

清空terminal

```
            putch(0x1B);
            putch('[');
            putch('2');
            putch('K');   // 清除整行
            putch(0x1B);
            putch('[');
            putch('H');   // 游標移到左上角
```

清空一行

1.宣告clear_line()

```
void clear_line() {
    // 清整行
    putch(0x1B); putch('['); putch('2'); putch('K');

    // 回到行首（不會換行）
    putch(0x1B); putch('['); putch('0'); putch('G');
}
```

2.呼叫clear_line()

```
clear_line();
```

清空一字

```
putch('\b');
putch(' ');
putch('\b');
```

初始化timer1

1.iniyiallization加入

```
// =======================
// Timer1 Initialization
// =======================
T1CONbits.TMR1ON = 0;     // 先關閉 Timer1
T1CONbits.T1CKPS = 0b11;  // Prescaler = 1:8
T1CONbits.TMR1CS = 0;     // Clock = Fosc/4 (內部時鐘)
T1CONbits.T1OSCEN = 0;    // 關 Timer1 外部震盪器 (不需要)
TMR1H = 0;                // 計數器高位清零
TMR1L = 0;                // 計數器低位清零
T1CONbits.TMR1ON = 1;     // 開啟 Timer1
IPR1bits.TXIP = 0;   // Interrupt Priority bit
PIE1bits.RCIE = 1;   // Wanna use Interrupt (Receive)
IPR1bits.RCIP = 0;   // Interrupt Priority bit
```

2.H_ISR加入

```
if (PIR1bits.TMR1IF) {
        // Timer1 interrupt code here

        PIR1bits.TMR1IF = 0;  // 清中斷旗標
}
```