Lab10

Nguyen Nam Tung 103181157

Exercise 10.1

Verify your understanding of our modifications by implementing the same changes yourself (*Try not to just copy the solution from lectures - give it a go yourself and use the lecture example as a check point if you need!).*

(a) First write the delay function. This function should take a single input, the number of seconds to delay for and be called from the main program every time there is a pause required.

Program

```
1       MOV R0, #.green
2       MOV R1, #.white
3       MOV R2, #1
4   flash:
5       STR R0,.Pixel367
6       LDR R3,.Time
7       push {R0}
8       MOV R0, R2
9       BL delay
10      Pop {R0}
11      STR R1,.Pixel367
12      LDR R3,.Time
13      Push {R0}
14      MOV R0, R2
15      BL delay
16      pop {R0}
17      B flash
18      halt
19  delay:
20      push {R3,R4,R5,R6}
21      MOV R3, R0
22      LDR R4, .Time
23  timer:
24      LDR R5, .Time
25      SUB R6, R5, R4
26      CMP R6, R3
27      BLT timer
28      pop {R3,R4,R5,R6}
29      RET
```

Load Save Edit

MOV R0, #.green

```
    MOV R1, #.white
    MOV R2, #1
flash:
    STR R0,.Pixel367
    LDR R3,.Time
    push {R0}
    MOV R0, R2
    BL delay
    Pop {R0}
    STR R1,.Pixel367
    LDR R3,.Time
    Push {R0}
    MOV R0, R2
    BL delay
    pop {R0}
    B flash
    halt
delay:
    push {R3,R4,R5,R6}
    MOV R3, R0
    LDR R4, .Time
timer:
    LDR R5, .Time
    SUB R6, R5, R4
    CMP R6, R3
    BLT timer
    pop {R3,R4,R5,R6}
```

```
RET
```

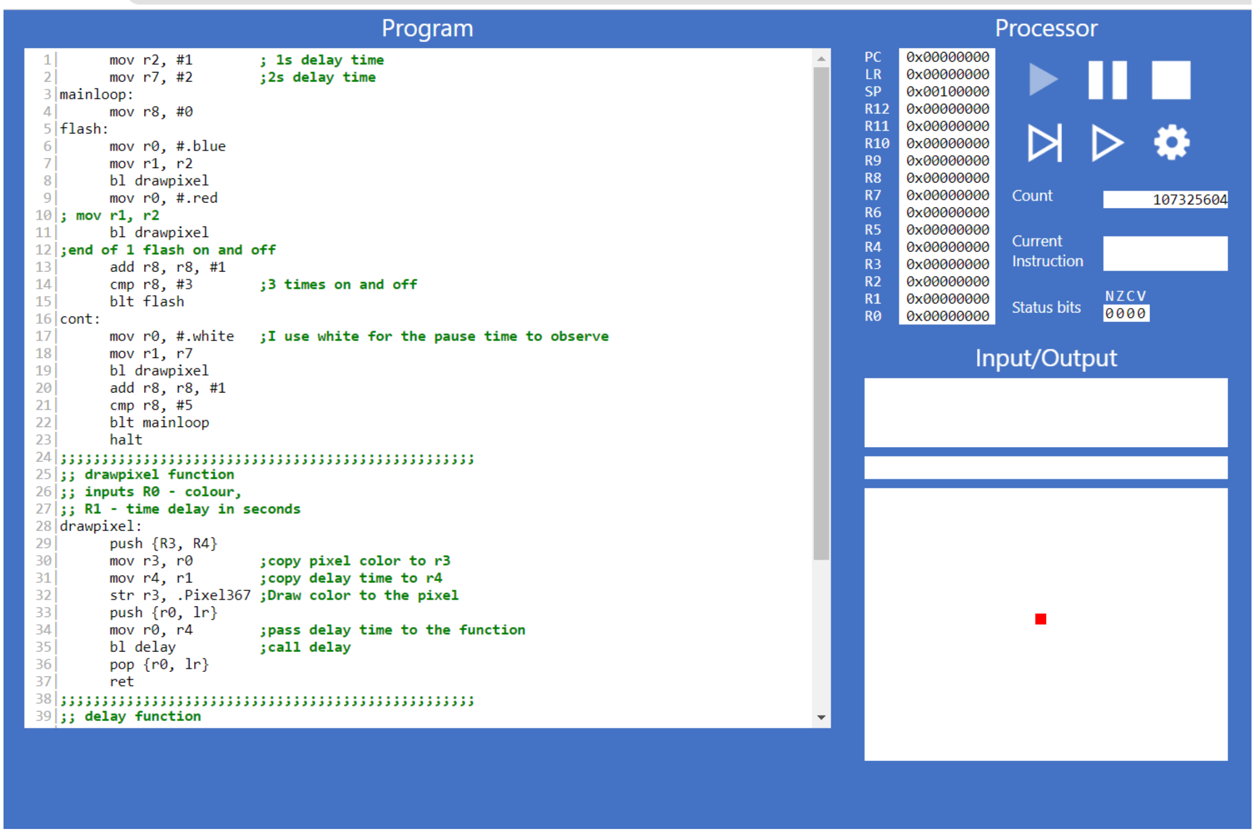
(b) Then write the drawpixel function. This function should take two inputs: the colour of the pixel to draw, and the time delay between on and off. This function should also call the delay function to insert the pauses between on and off.

(c) when you implemented drawpixel, what did you have to do with LR to make it work? Why?

When your happy with your solution, show your tutor the solution you have written and demonstrate it. Take a screen shot and include, and provide your answers in your solution document.

Exercise 10.2

Modify your program above so that your program flashes the LED rapidly (i.e., 1 second between on and off) three times, and then pauses for 2 seconds before repeating the rapid pattern. You should make use of your drawpixel and delay functions to do this.




```
mov r2, #1       ; 1s delay time
mov r7, #2       ;2s delay time
loop:
```

```
    mov r8, #0
flash:
    mov r0, #.blue
    mov r1, r2
    bl drawpixel
    mov r0, #.red
; mov r1, r2
    bl drawpixel
;end of 1 flash on and off
    add r8, r8, #1
    cmp r8, #3        ;3 times on and off
    blt flash
cont:
    mov r0, #.white   ;I use white for the pause time to observe
    mov r1, r7
    bl drawpixel
    add r8, r8, #1
    cmp r8, #5
    blt loop
    halt
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; drawpixel function
;; inputs R0 - colour,
;; R1 - time delay in seconds
drawpixel:
    push {R3, R4}
    mov r3, r0        ;copy pixel color to r3
```

```
    mov r4, r1      ;copy delay time to r4
    str r3, .Pixel367 ;Draw color to the pixel
    push {r0, lr}
    mov r0, r4      ;pass delay time to the function
    bl delay        ;call delay
    pop {r0, lr}
    ret
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; delay function
;; inputs R0 - time delay in seconds
delay:
    push {R3, R4, R5, R6}
    mov r3, r0
    ldr r4, .Time
timer:
    ldr r5, .Time
    sub r6, r5, r4
    cmp r3, r6
    bne timer
    pop {R3,R4,R5,R6}
    RET
```

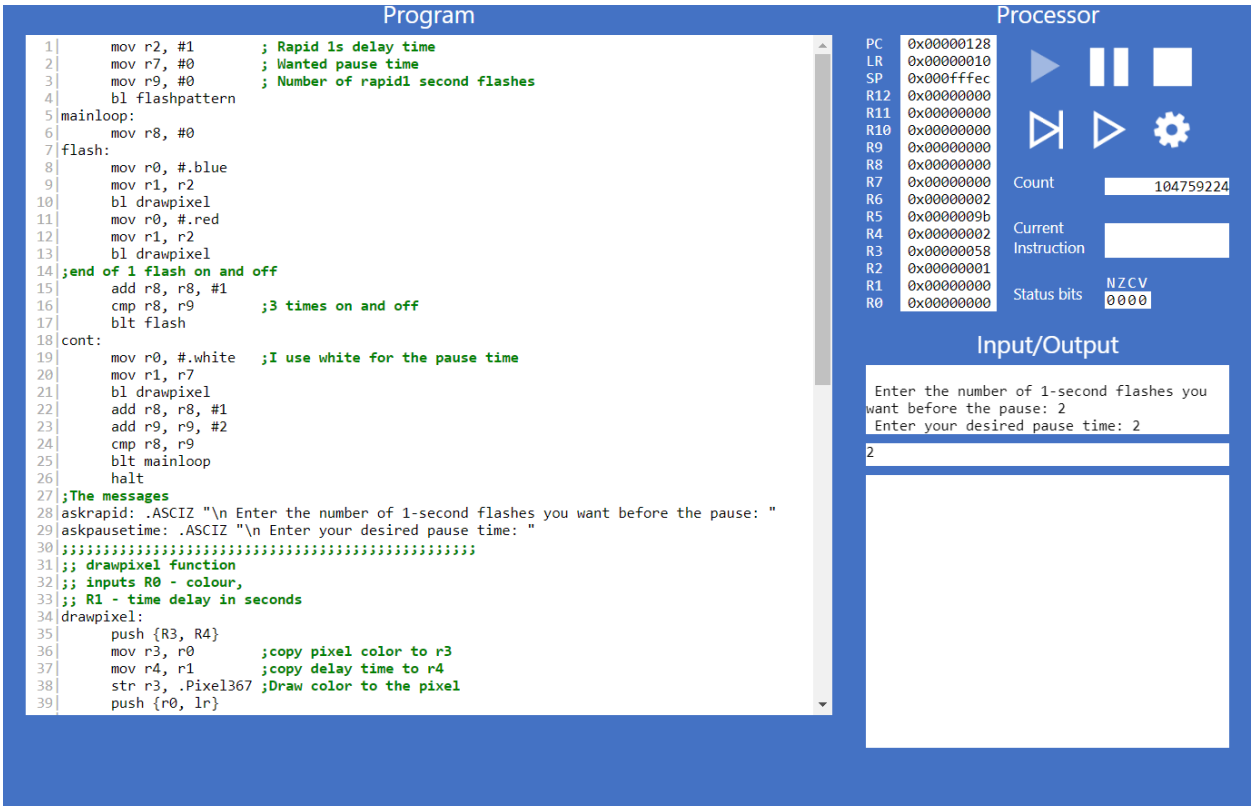
Show your solution to your tutor. Take a screen shot and include in your solution document.

Exercise 10.3

Modify your program from 10.2 by writing a function called flashpattern: that accepts two inputs:

- the number of "rapid" 1 second flashes before the pause
- the pause time (in seconds) between each set of rapid flashes

You should call this function from your main program and allow it to manage the flashing of your pixel LED. This function should make use of your previous functions to perform the task.




```
1|      mov r2, #1        ; Rapid 1s delay time
2|      mov r7, #0        ; Wanted pause time
3|      mov r9, #0        ; Number of rapid1 second flashes
4|      bl flashpattern
5|loop:
6|      mov r8, #0
7|flash:
8|      mov r0, #.blue
9|      mov r1, r2
10|     bl drawpixel
11|     mov r0, #.red
12|     mov r1, r2
13|     bl drawpixel
```

```
;end of 1 flash on and off
    add r8, r8, #1
    cmp r8, r9      ;3 times on and off
    blt flash
cont:
    mov r0, #.white   ;I use white for the pause time
    mov r1, r7
    bl drawpixel
    add r8, r8, #1
    add r9, r9, #2
    cmp r8, r9
    blt loop
    halt
;The messages
askrapid: .ASCIZ "\n Enter the number of 1-second flashes you want before the pause: "
askpausetime: .ASCIZ "\n Enter your desired pause time: "
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; drawpixel function
;; inputs R0 - colour,
;; R1 - time delay in seconds
drawpixel:
    push {R3, R4}
    mov r3, r0      ;copy pixel color to r3
    mov r4, r1      ;copy delay time to r4
    str r3, .Pixel367 ;Draw color to the pixel
    push {r0, lr}
    mov r0, r4      ;pass delay time to the function
```

```
    bl delay        ;call delay
    pop {r0, lr}
    ret
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; delay function
;; inputs R0 - time delay in seconds
delay:
    push {R3, R4, R5, R6}
    mov r3, r0
    ldr r4, .Time
timer:
    ldr r5, .Time
    sub r6, r5, r4
    cmp r3, r6
    bne timer
    pop {R3,R4,R5,R6}
    RET
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; flashpattern function
flashpattern:
    push {R3, R4, R5, R6, lr}
;Take first input
    mov r3, #askrapid
    str r3, .WriteString
    ldr r4, .InputNum
    str r4, .WriteUnsignedNum
;Take second input
```

```
68|     mov r5, #askpausetime
69|     str r5, .WriteString
70|     ldr r6, .InputNum
71|     str r6, .WriteUnsignedNum
72|;Add input into the right register
73|     mov r9, r4
74|     mov r7, r6
75|     pop {R3,R4,R5,R6, lr}
76|     ret
```

Show your solution to your tutor. Take a screen shot and include in your solution document