

ESET 349 Microcontroller Architecture

Lab 3: Toggling LEDs using assembly language programming

Objectives

1. Build a simple electric circuit on a breadboard with LEDs and resistors in series.
2. Develop a flowchart for programming and use this flowchart to implement the flowchart in assembly language.
3. Use function calls and the link register to decide the flow of control.
4. Become familiar with GPIO memory mappings and utilize the MSP432 Specifications document to identify addresses and offsets for programing ports.

MSP432 specifications and technical reference material available [here](#).

Your Tasks

1. Program an MSP432 microcontroller to toggle three LEDs at a given interval.
The program configures three pins as output pins. Use P6.5, P6.4, and P6.0 as output pins in your program if they are functional. These pins are connected to a circuit placed on a breadboard. Each LED is connected in series with a 330 Ω resistor. Figure 1 shows a sketch. The program repeatedly sends HIGH and LOW signals at a predetermined time interval such that the LEDs toggle in a **sequence**.
2. Implement a two-loop delay routine as described in the flowchart on the next page.

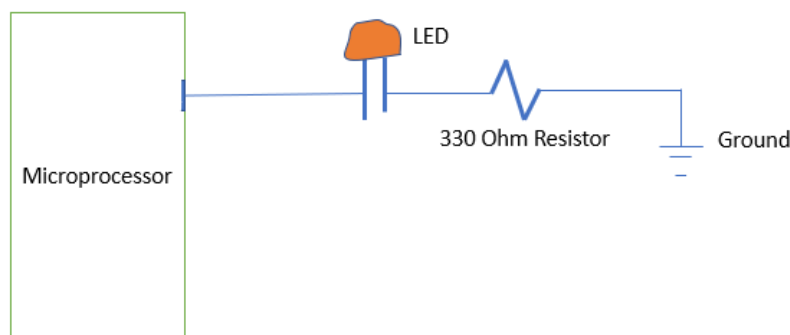


Figure 1. Circuit sketch

Flowchart

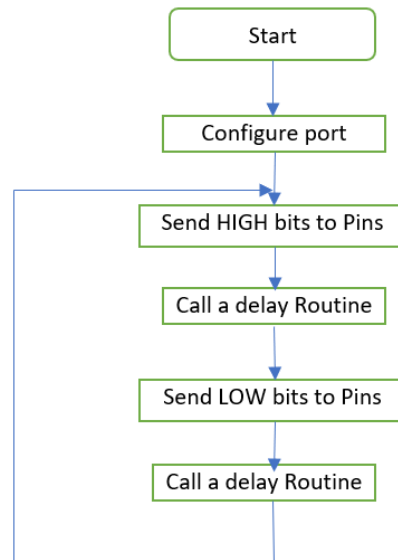


Figure 2. A flowchart for the main body of the program

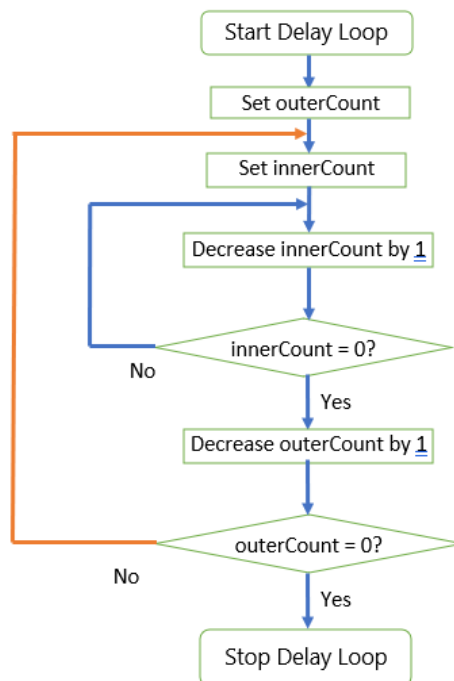


Figure 3. A flowchart for the two-loop (nested) delay routine

Program Sketch

Incomplete program as a guide only; please complete the code. Notice this skeleton program sends a HIGH signal to Pin 6.0 only. If you suspect your hardware is not functional, try using a different set of three pins, or reach out to your TA for assistance.

We encourage you to come up with the entire program and use this snippet only as a reference.

Complete toggling with the given single-loop delay function. After demonstrating to your TA, replace it with the two-loop delay function as suggested in Figure 3.

```
1 ; Lab 3 Toggling LED
2
3         area Lab3, code, readonly
4         export __main
5 __main  proc
6
7         ; Configure GPIO
8         LDR R0, =0x40004C00      ; Port 1 base address
9         ADD R0, #0x41           ; Port 6 base address
10        MOV R1, #0x31           ; Byte to configure pins 5,4,0 output
11        STRB R1, [R0, #0x04]    ; Configure pins 6.5,6.4,6.0 as output
12
13        ; Loop and toggle LEDs
14 repeat MOV R1, #0x01           ; Byte to make only pin 0 HIGH
15        STRB R1, [R0, #0x02]    ; Make only pin 6.0 HIGH
16        BL delay                ; Remain ON for a while
17
18        ; ADD CODE TO SEND LOW SIGNAL
19
20        BL delay                ; Remain OFF for a while
21
22        ; ADD CODE TO TOGGLE PINS 6.4 and 6.5
23
24        B repeat                ; Loop infinitely
25        endp                   ; End of procedure __main
26
27        ; Single loop delay
28        ; Complete toggle using below delay routine
29
30        ; After completing toggling, ADD CODE FOR NESTED DELAY below
31
32 delay  function                ; Declare new procedure
33        MOV R12, #0x5000        ; NOTE: tune this constant if needed
34 continue SUB R12, #0x01
35        CMP R12, #0x00
36        BNE continue           ; Continue until R12 is positive
37        BX LR                  ; Return to address in LR
38        endp                   ; End of procedure delay
39
40        end
```