

Raymond Cao Jiang, 501183087

COE538 - Section 10

Lab 2 Part 1: Programming the I/O Devices

Code 1

The code's purpose is for reading the state of the switches (SW1) and then displaying them on the LED bar (LED1). First the code loads #\$FF (%1111 1111 in binary), then Port H is configured for output since 1 indicates output. Following this, the enabling of the pull-up resistor on Port T makes sure the inputs on the switch can be read. This pull-up resistor ensures that it is kept at a high (1). Next is a loop which continuously reads the state of the switch from Port T and updates the display of Port H.

Code 2

The purpose of this code is to read the state of the keypad and use 3 bits of the key code to control the color of LED2. Port P is configured for LED output and allows a bit to activate the keypad. Following this is a loop used to read and store key inputs from the keypad into the accumulator and shifts 4 times to the right to isolate 3 bits for the color of the LED. Finally, right before the end of the loop it outputs the accumulator content to Port P and displays it on the LED.

Code 3

This code is responsible for generating a sound tone by using a frequency determined by a software delay loop. First Port P is configured for the output (1 is output) followed by preparing to drive PP7 high (meaning setting the output to a high voltage 1). After this in the main loop, PP7 is set high and a delay counter is initialized. This keeps getting decremented with an inner loop until it is 0 as seen with the BNE back to Delay. After the delay loop, MSB is toggled (with %1000 0000) producing a sound tone. This process repeats endlessly until interrupted.



```
main.asm
* Path: C:\Users\ray\Desktop\coe538lab2\lab2part1q1\Source\main.asm
; This stationary serves as the framework for a
; user application (single file, absolute assembly application)
; For a more comprehensive program that
; demonstrates the more advanced functionality of this
; processor, please see the demonstration applications
; located in the examples subdirectory of the
; Preamble CodeGenerator for the 8051 Program directory

; export symbols
; XREF Entry, _Startup ; export 'Entry' symbol
; ABSENTRY Entry ; for absolute assembly, mark this as application entry point

; Include derivative-specific definitions
; INCLUDE 'derivative.inc'

; code section
; ORG 64000

Entry:
_Startup:
; 1 making sure the led on board turns on based on input
; IDAA #$FF ; ACCA = 0FF %1111 1111 ($ to specify an exact number to be used at this moment)
; STAA DDRE ; Config Port B for output
; STAA PTT ; Enab pull-up res. of Port T
; STAA PTT ; Read Port T
; STAA SW1 ; Display SW1 on LED1 connected to Port H
; BRA Loop

; =====
; Interrupt Vectors
; =====
; ORG 0FFFF
; IC.V Entry ; Reset Vector

Code 2
main.asm
* Path: C:\Users\ray\Desktop\coe538lab2\lab2part1q2\Source\main.asm
; This stationary serves as the framework for a
; user application (single file, absolute assembly application)
; For a more comprehensive program that
; demonstrates the more advanced functionality of this
; processor, please see the demonstration applications
; located in the examples subdirectory of the
; Preamble CodeGenerator for the 8051 Program directory

; export symbols
; XREF Entry, _Startup ; export 'Entry' symbol
; ABSENTRY Entry ; for absolute assembly, mark this as application entry point

; Include derivative-specific definitions
; INCLUDE 'derivative.inc'

; code section
; ORG 64000

Entry:
_Startup:
; 2
; BSET DDRE,%11111111 ; Configure Port P for output (LED2 ctrl) (using led as output)
; BSET DDRE,%00010000 ; Configure pin PE4 for output (enable bit)
; BSET PTT,%00010000 ; Enable keypad
; LDAA PTT ; Read a key code into AccA
; LSRA ; Shift right AccA
; LSRA ;
; LSRA ;
; STAA PTP ; Output AccA content to LED2
; BRA Loop

; =====
; Interrupt Vectors
; =====
; ORG 0FFFF
; IC.V Entry ; Reset Vector

Code 3
main.asm
* Path: C:\Users\ray\Desktop\coe538lab2\lab2part1q3\Source\main.asm
; This stationary serves as the framework for a
; user application (single file, absolute assembly application)
; For a more comprehensive program that
; demonstrates the more advanced functionality of this
; processor, please see the demonstration applications
; located in the examples subdirectory of the
; Preamble CodeGenerator for the 8051 Program directory

; export symbols
; XREF Entry, _Startup ; export 'Entry' symbol
; ABSENTRY Entry ; for absolute assembly, mark this as application entry point

; Include derivative-specific definitions
; INCLUDE 'derivative.inc'

; code section
; ORG 64000

Entry:
_Startup:
; BSET DDRE,%11111111 ; Config Port B for output (change last four bits to 0 for input)
; LDAA #10000000 ; Prepare to drive PP7 high (high source 0 for output)
; STAA PTP ; Drive PP7
; LDAA #1FFFF ; Initialize the loop counter
; DEC ; Decrement the loop counter
; BNE Delay ; If not done, continue to loop (branch if not equal to 0)
; BRA MainLoop ; Toggle the MSB of AccA
; BRA MainLoop ; Go to MainLoop

; =====
; Interrupt Vectors
; =====
; ORG 0FFFF
; IC.V Entry ; Reset Vector
```