COE538 Lab 2 - Adrian Omoruyi (501162620)

2.1

```
;*****************************************************************
;*Assembly program ot multiply two 8-bit numbers together
;*****************************************************************

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


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

;*****************************************************************
;* The actual program starts here *
;*****************************************************************
            ORG $4000

Entry:
_Startup:
            LDAA #$FF ; ACCA = $FF
            STAA DDRH ; Config. Port H for output
            STAA PERT ; Enab. pull-up res. of Port T

     Loop:  LDAA PTT ; Read Port T
            STAA PTH ; Display SW1 on LED1 connected to Port H
            BRA Loop ; Loop

;*************************************************************
;*                 Interrupt Vectors                         *
;*************************************************************
            ORG   $FFFE
            FDB   Entry           ; Reset Vector
```

This code is what controls the bar LED lights. The lights are activated based on the state of the SW1 switch. The code sets up the variables and starts an infinite loop that checks the active state of SW1.

2.2

```
;*****************************************************************
;*Assembly program ot multiply two 8-bit numbers together
;*****************************************************************

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


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

;*****************************************************************
;* The actual program starts here *
;*****************************************************************
            ORG $4000

Entry:
_Startup:


            BSET DDRP,%11111111 ; Configure Port P for output (LED2 cntrl)
            BSET DDRE,%00010000 ; Configure pin PE4 for output (enable bit)
            BCLR PORTE,%00010000; Enable keypad

     Loop:  LDAA PTS ; Read a key code into AccA
            LSRA     ; Shift right Acca
            LSRA
            LSRA
            LSRA
            STAA PTP ; Output AccA content to LED2
            BRA Loop ; Loop

;*************************************************************
;*                 Interrupt Vectors                         *
;*************************************************************
            ORG   $FFFE
            FDB   Entry           ; Reset Vector
```

This code changes the color of the LED based on the last pressed key on the keypad. Using an infinite loop, it stores the keycode in register A, shifting it right 3 times. Using this as the RGB code, it is displayed from LED2.

2.3

```
;*****************************************************************
;*Assembly program ot multiply two 8-bit numbers together
;*****************************************************************

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


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

;*****************************************************************
;* The actual program starts here *
;*****************************************************************
            ORG $4000

Entry:
_Startup:


            BSET DDRP,%11111111 ; Configure Port P for output
            LDAA #%10000000 ; Prepare to drive PP7 high

  Main Loop STAA PTP ; Drive PP7
            LDX #$1FFF ; Initialize the loop counter
      Delay DEX ;Decrement the loop counter
            BNE Delay ;If not done, continue to loop
            EORA #%10000000 ;Toggle the MSB of AccA
            BRA MainLoop ;Go to MainLoop


;*****************************************************************
;*                 Interrupt Vectors                             *
;*****************************************************************
            ORG   $FFFE
            FDB   Entry           ; Reset Vector
```

This code switches between playing a sound tone and staying quiet for the same duration. Within a loop, the sound plays until it ends, which is where the speaker is turned off. This process then loops until the system is turned off.