Assembly Code Programming

1. Connect the eight LEDs to PORTA and write a loop to flash each one in turn from right to left, repeating for ever.

2. Connect the seven segment display to PORTB and make it count 0, 1, 2, 3 on the right display, repeating for ever.

3. Connect the switches to PORTA and the LEDs to PORTB. Poll PORTA and use the data to control PORTB. Each switch should control the corresponding LED.

4. Add a bit mask to task 3 so the middle two switches are ignored and the middle two LEDs never come on.

5. Connect the heater and thermostat to PORTA. Write code to form a closed loop control system. The heater should warm to about 21 or 22 degrees C.

6. Connect the traffic lights to PORTA. Write code to step the lights through a realistic sequence ignoring the timing.

7. Add a time delay to task 6 to make the simulation more realistic. Write a loop that counts down to zero. Store the countdown number in the W register.

8. Connect the switches and the H-Bridge controller to PORTA. It the switch on D6 is closed, make the motor run clockwise. If the switch on D7 is closed, make the motor run anticlockwise. If both switches are open, the motor should free-wheel and slowly slow down. Ignore all the other switches.

9. Display the letter B on the LED Matrix display.

10. Write some code using PRE and TMR to flash an LED on and off. The code should run soon after 200 clock pulses / instructions have executed.

Answers

```
; ======
; Task 1
; ======
LOOP:
    MOVW    0X01
    MOVWR   PORTA
    MOVW    0X02
    MOVWR   PORTA
    MOVW    0X04
    MOVWR   PORTA
    MOVW    0X08
    MOVWR   PORTA
    MOVW    0X10
    MOVWR   PORTA
    MOVW    0X20
    MOVWR   PORTA
    MOVW    0X40
    MOVWR   PORTA
    MOVW    0X80
    MOVWR   PORTA
    JMP     LOOP
```

```
; ======
; Task 2
; ======
LOOP:
    MOVW    0X7D    ; 0 = 01111101
    MOVWR   PORTA
    MOVW    0X05    ; 1 = 00000101
    MOVWR   PORTA
    MOVW    0X5B    ; 2 = 01011011
    MOVWR   PORTA
    MOVW    0X4F    ; 3 = 01001111
    MOVWR   PORTA
    JMP     LOOP
```

```
; ======
; Task 3
; ======
INIT:
    MOVW    0xFF    ; Eight Inputs
    MOVWR   TRISA   ; on PORTA
    MOVW    0x00    ; Eight Outputs
    MOVWR   TRISB   ; on PORTB

POLL:
    MOVRW   PORTA
    MOVWR   PORTB
    JMP     POLL
```

```
; ======
; Task 4
; ======
INIT:
    MOVW    0xFF    ; Eight Inputs
    MOVWR   TRISA   ; on PORTA
    MOVW    0x00    ; Eight Outputs
    MOVWR   TRISB   ; on PORTB
POLL:
    MOVRW   PORTA
    ANDW    0xE7    ; 11100111
    MOVWR   PORTB
    JMP     POLL
```

```
; ======
; Task 5
; ======
INIT:
    MOVW    0x01    ; One Input
    MOVWR   TRISA   ; on PORTA

LOOP:
    MOVRW   PORTA
    ANDW    0x01
    JPZ     TOOCOOL
    JMP     TOOWARM

TOOCOOL:
    MOVW    0x81
    MOVWR   PORTA
    JMP     LOOP

TOOWARM:
    MOVW    0x00
    MOVWR   PORTA
    JMP     LOOP
```

```
; ======
; Task 6
; ======
LOOP:               ; GARxxRAG
    MOVW    0x21    ; 00100001
    MOVWR   PORTA   ; GARxxRAG
    MOVW    0x62    ; 01100010
    MOVWR   PORTA   ; GARxxRAG
    MOVW    0x84    ; 10000100
    MOVWR   PORTA   ; GARxxRAG
    MOVW    0x46    ; 01000110
    MOVWR   PORTA
    JMP     LOOP
```

```
; ======
; Task 7
; ======
LOOP:               ; GARxxRAG
    MOVW    0x21    ; 00100001
    MOVWR   PORTA   ; GARxxRAG
    MOVW    0x30
    CALL    DELAY

    MOVW    0x62    ; 01100010
    MOVWR   PORTA   ; GARxxRAG
    MOVW    0x07
    CALL    DELAY

    MOVW    0x84    ; 10000100
    MOVWR   PORTA   ; GARxxRAG
    MOVW    0x30
    CALL    DELAY

    MOVW    0x46    ; 01000110
    MOVWR   PORTA
    MOVW    0x07
    CALL    DELAY

    JMP     LOOP

DELAY:
    SUBW    0x01
    JPZ     END_DELAY
    JMP     DELAY

END_DELAY:
    RET
```

```
; ======
; Task 8
; ======
INIT:
    MOVW    0xC0    ; D6 and D7 are inputs
    MOVWR   TRISA
POLL:
    MOVRW   PORTA   ; Poll
    ANDW    0xC0    ; Bit Mask
    SUBW    0x40    ; Test D6
    JPZ     CLOCK

    MOVRW   PORTA   ; Poll
    ANDW    0xC0    ; Bit Mask
    SUBW    0x80    ; Test D7
    JPZ     ANTICLOCK

    JMP     FREEWHEEL

CLOCK:
    MOVW    0x09    ; Clockwise
    MOVWR   PORTA
    JMP     POLL

ANTICLOCK:
    MOVW    0x06    ; Anti-Clockwise
    MOVWR   PORTA
    JMP     POLL

FREEWHEEL:
    MOVW    0x00    ; Free-wheel
    MOVWR   PORTA
    JMP     POLL
```

```
; ========================================
; TASK 9 – This example shows an "A"
; ========================================
;   Set PORTA to be an output
    MOVW    0x00    ; Copy 0 (0000 0000) into W
    MOVWR   TRISA   ; Copy W to TRISA

;   PORTB D0 - D4 as outputs
    MOVW    0xE0    ; Copy 0xE0 1110 0000 to W
    MOVWR   TRISB   ; Copy W to TRISB

loop:
;   Display the first column
    MOVW    0x00    ; Copy 0 into W
    MOVWR   PORTB   ; Copy W to PORTB
    MOVW    0x7C    ; Copy 0x7C into W
    MOVWR   PORTA   ; Copy W to PORTA
    MOVW    0x01    ; Copy 1 into W
    MOVWR   PORTB   ; Copy W to PORTB
    CALL    DELAY
    NOP
    NOP
    MOVW    0x00    ; Copy 0 into W
    MOVWR   PORTB   ; Copy W to PORTB

;   Display the second column
    MOVW    0x0A    ; Copy 0x0A into W
    MOVWR   PORTA   ; Copy W to PORTA
    MOVW    0x02    ; Copy 2 into W
    MOVWR   PORTB   ; Copy W to PORTB
    CALL    DELAY
    NOP
    NOP
    MOVW    0x00    ; Copy 0 into W
    MOVWR   PORTB   ; Copy W to PORTB
```

```
; ==== STILL TASK 9 ====
;   Display the third column
    MOVW    0x09    ; Copy 0x09 into W
    MOVWR   PORTA   ; Copy W to PORTA
    MOVW    0x04    ; Copy 4 into W
    MOVWR   PORTB   ; Copy W to PORTB
    CALL    DELAY
    NOP
    NOP
    MOVW    0x00    ; Copy 0 into W
    MOVWR   PORTB   ; Copy W to PORTB

;   Display the fourth column
    MOVW    0x0A    ; Copy 0x0A into W
    MOVWR   PORTA   ; Copy W to PORTA
    MOVW    0x08    ; Copy 8 into W
    MOVWR   PORTB   ; Copy W to PORTB
    CALL    DELAY
    NOP
    NOP
    MOVW    0x00    ; Copy 0 into W
    MOVWR   PORTB   ; Copy W to PORTB

;   Display the fifth column
    MOVW    0x7C    ; Copy 0x7C into W
    MOVWR   PORTA   ; Copy W to PORTA
    MOVW    0x10    ; Copy 0x10 into W
    MOVWR   PORTB   ; Copy W to PORTB
    CALL    DELAY
    JMP     loop    ; Go back to the beginning
; ========================================
DELAY:
    NOP
    NOP
    NOP
    NOP
    RET
```

```
; ========================================
; TASK 10 – Using TMR and PRE for timing
; ========================================
; CONNECT THE LEDs TO PORTA
    MOVW    0x10    ; Decrement TMR after 16 + 1
    MOVWR   PRE     ; Set the Prescaler

INIT:
    MOVW    0x0C    ; Put 12 into the W register
    MOVWR   TMR     ; Copy 12 to the Timer reg'

MAIN:
    MOVRW   SR      ; POLL the Status Register
    ANDW    0x02    ; BIT MASK FOR T FLAG
    JPZ     MAIN    ; Test for T flag
    MOVRW   PORTA   ; T flag not set. Read PORTA
    XORW    0XFF    ; Invert PORTA data
    MOVWR   PORTA   ; Write inverted data to
PORTA
    JMP     INIT
```