**Question 1**

|  |
| --- |
| **#include** <msp430.h>  **#define** RED\_ON 0x01 // P1.0 is the red LED -> 0000 0001  **#define** RED\_OFF 0x7E // Used to turn off the red LED -> 1111 1110  **#define** BUTTON 0x04 // P1.2 is the push-button -> 0000 0100  **#define** DEVELOPMENT 0x5A80 // Stop the watch dog timer  **#define** ENABLE\_PINS 0xFFFE // Required to use inputs and outputs  **void** **delay\_ms**(**unsigned** **int** ms){ // ~1000 cycle equal to 1ms.  **while** (ms){  **\_\_delay\_cycles**(1000);  ms--;  }  }  **void** **main**(){  WDTCTL = DEVELOPMENT; // Need for development mode  PM5CTL0 = ENABLE\_PINS; // Prepare pins for I/O usage  P1DIR = RED\_ON; // P1.0 is output  P1REN = BUTTON; // Input Button(p1.2) resistors activated.  P1OUT = BUTTON; // Input Button(p1.2) used to be a pull-up resistor  **volatile** **int** counter = 0;  **while**(1){  **if**((BUTTON & P1IN) == 0x00){ // Button pressed = 0.  **while**((BUTTON & P1IN) == 0x00){ // Hold pressed state.  delay\_ms(20); // to debouncing.  }  counter++;  }  **if**(counter & 0x1){  P1OUT = P1OUT ^ RED\_ON; // Led toggling.  delay\_ms(100);  }**else**{  P1OUT = P1OUT & RED\_OFF;  }  }  } |

**Question 2**

|  |
| --- |
| **#include** <msp430.h>  **#define** RED\_ON 0x01 // P1.0 is the red LED -> 0000 0001  **#define** RED\_OFF 0x7E // Used to turn off the red LED -> 1111 1110  **#define** BUTTON 0x04 // P1.2 is the push-button -> 0000 0100  **#define** DEVELOPMENT 0x5A80 // Stop the watch dog timer  **#define** ENABLE\_PINS 0xFFFE // Required to use inputs and outputs  **void** **delay\_ms**(**unsigned** **int** ms){  **while** (ms){  **\_\_delay\_cycles**(1000);  ms--;  }  }  **void** **blink**(**unsigned** **int** ms){  P1OUT = P1OUT ^ RED\_ON;  delay\_ms(ms);  }  **void** **main**(){  WDTCTL = DEVELOPMENT; // Need for development mode  PM5CTL0 = ENABLE\_PINS; // Prepare pins for I/O usage  P1DIR = RED\_ON; // P1.0 is output  P1REN = BUTTON; // Input Button(p1.2) resistors activated.  P1OUT = BUTTON; // Input Button(p1.2) used to be a pull-up resistor (Button pressed = 0)  **while**(1){  **if**((BUTTON & P1IN) == 0x00){ // Button pressed = 0.  **while**((BUTTON & P1IN) == 0x00) // Hold pressed state.  blink(100);  }**else**{  blink(500);  }  delay\_ms(50); // to debouncing  }  } |

**Conclusion:**

In this experiment I’ve used fundamental I/O devices with simple algorithm. Also, I’ve applied bitwise operands to support my algorithm. I noticed that when I push the button to change mode, I must wait delay over. In that case we going to use hardware interrupts to catch button pressed.