

BECS 31421
EXPERIMENT 02

Introduction to Input and Output in PIC Programming

STUDENT NAME: G.P.D. THAMARA
STUDENT NUMBER: EC/2021/005

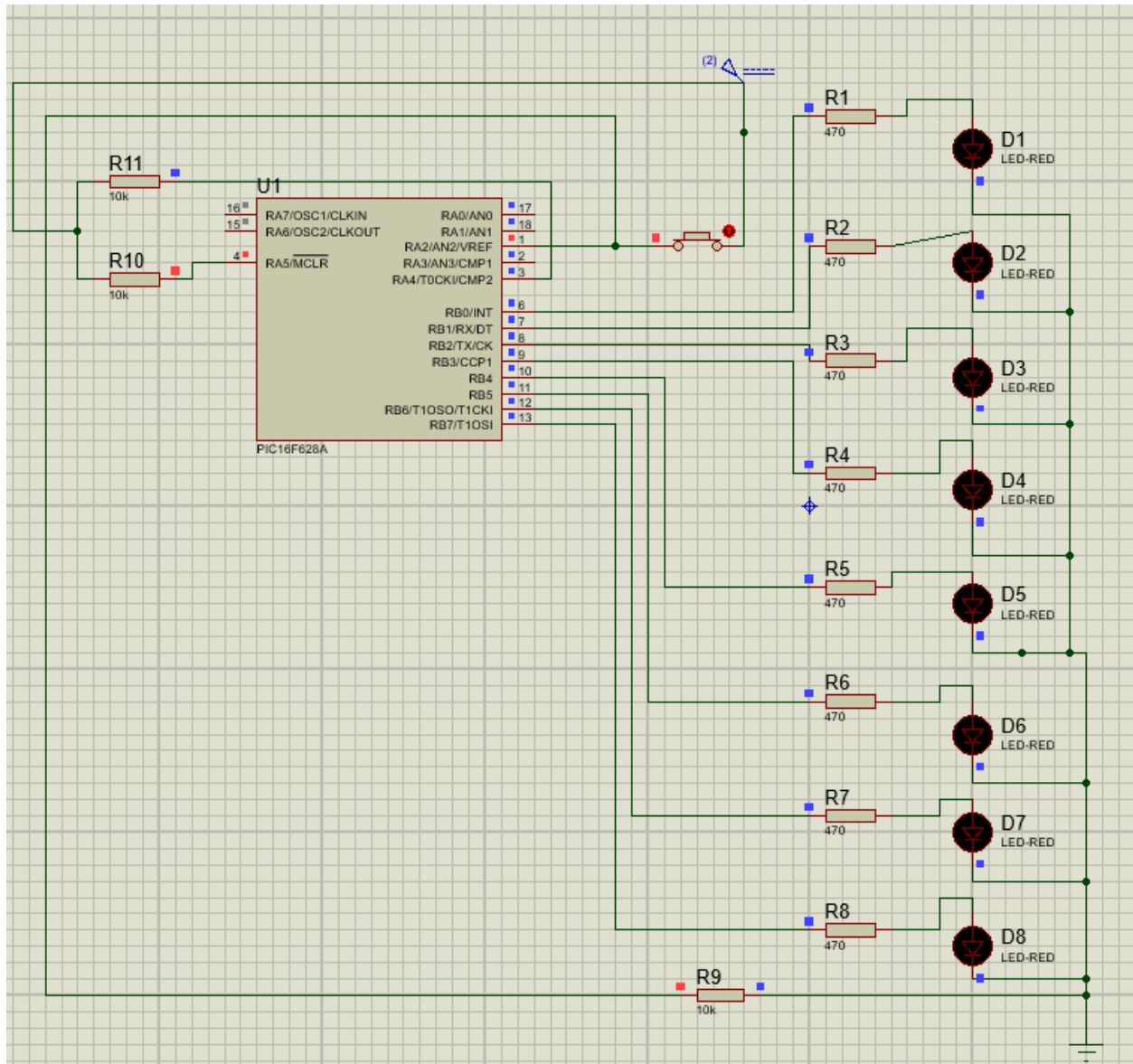
Source Code

```
sbit sw at RA2_bit;
void main() {
    CMCON = 0x07 ;    // Disable Comparator
    TRISA = 0x04 ;    //configure TRISA register
    TRISB = 0x00 ;    // Configure TRSB register
    PORTB = 0xFF;     //Intialize PORTB register
    RA2_bit = 0x00;    // Set RA_2bit to low state

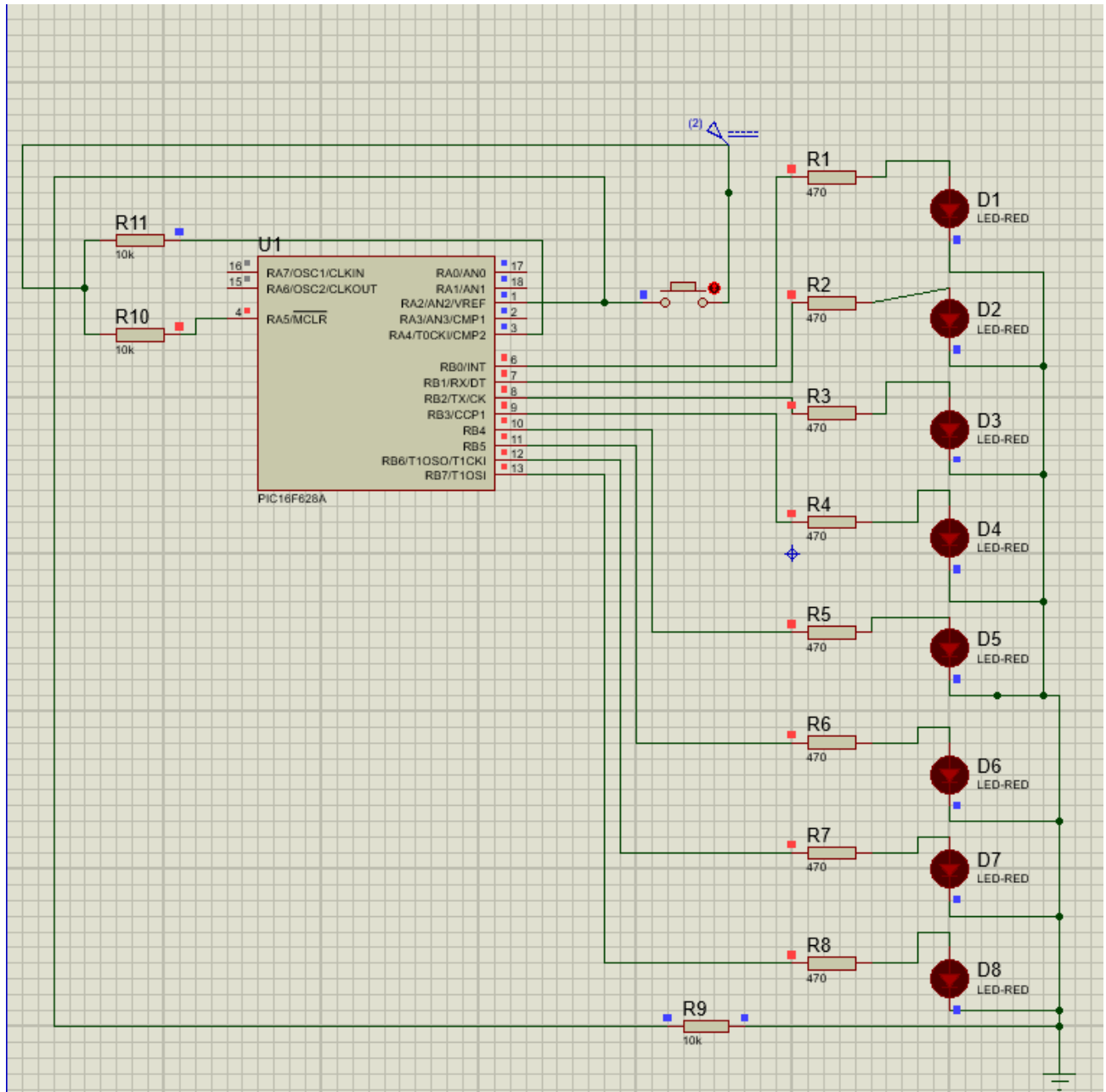
    //Enter the first loop
    do{
        //Check the state of the switch
        if(sw==1){
            PORTB = 0x00;
            //If the switch is pressed ,set PORTB to be low
        }
        else{
            //If the switch is not pressed,set PORTB to be high
            PORTB = 0xFF;
        }

    }while(1);        // Enter a condition for the infinite loop
}
```

When the switch is pressed.



When the switch is released.



Discussion

The experiment on configuring input and output pins in PIC microcontroller programming effectively illustrates the critical role of the TRIS register in embedded systems design. By setting TRISA (e.g., 0b00000100) and TRISB (e.g., 0b00000000), the experiment successfully designated RA2 as an input and PORTB as an output, enabling precise control over pin functionality. The use of binary notation ensured accurate pin allocation, facilitating seamless interaction with external devices such as switches and LEDs. The 'sbit' declaration for the switch input (sw) enhanced real-time state monitoring, allowing conditional logic to dictate PORTB's behavior—transitioning to 0b11111111 when the switch was pressed (sw == 1) and reverting to 0b00000000 upon release (sw == 0). This dynamic response underscores the importance of input-output synchronization in microcontroller applications. Overall, the experiment provided valuable insights into TRIS register manipulation and conditional programming, strengthening foundational skills in embedded systems development.