# OBJECTIVE:

* Implement I/O Port interface and calculation instructions.

# REFERENCES:

* Lab Manual Chapter 1-2

# EXPERIMENT 1:

1. Connect one AVR port (e.g., PORT A) to a dip switch. Connect another port to a LED bar (e.g., PORT B).
2. Write a program to continuously read the state of the dip switch and send it to the LED. If the switch is in the OFF state, the corresponding LED will turn off.

# EXPERIMENT 2:

1. Write a program to read the value of the port connected to the dip switch, add 5 to it, and send it to the port connected to the LED bar.
2. Change the state of the dip switch and observe the status of the LED bar.

# EXPERIMENT 3:

1. Connect and implement a program to calculate the product of two nibbles (high and low) of PORT A and send it to PORT B. Consider these two nibbles as unsigned numbers.

Example: PORT A = 0b0111\_1111, then PORT B = 7 \* 15.

1. Change the state of the dip switch and observe the status of the LED bar.

# EXPERIMENT 4:

1. Connect and implement a program to calculate the product of two nibbles (high and low) of PORT A and send it to PORT B. Consider these two nibbles as signed numbers.

Example: PORT A = 0b0111\_1111, then PORT B = 3 \* (-1).

1. Change the state of the dip switch and observe the status of the LED bar.

# EXPERIMENT 5:

1. Connect PA0 to a single switch and PA1 to a single LED on the LED block (note that they are from the same port).
2. Write a program to turn on the LED if the switch is pressed and turn it off if the switch is released.

# EXPERIMENT 1

1. Answer the following questions:
   1. **How do you retrieve values from the two nibbles of PORT A?**

To retrieve values from the two nibbles of PORT A, we would typically use bit manipulation operations in asm. We read the entire PORT A and then mask out the bits corresponding to the nibble.

* 1. **How do you enable pull-up resistors?**

To enable pull-up resistors for individual pins on an AVR microcontroller, you need to set the corresponding bits in the PORT register to logic HIGH

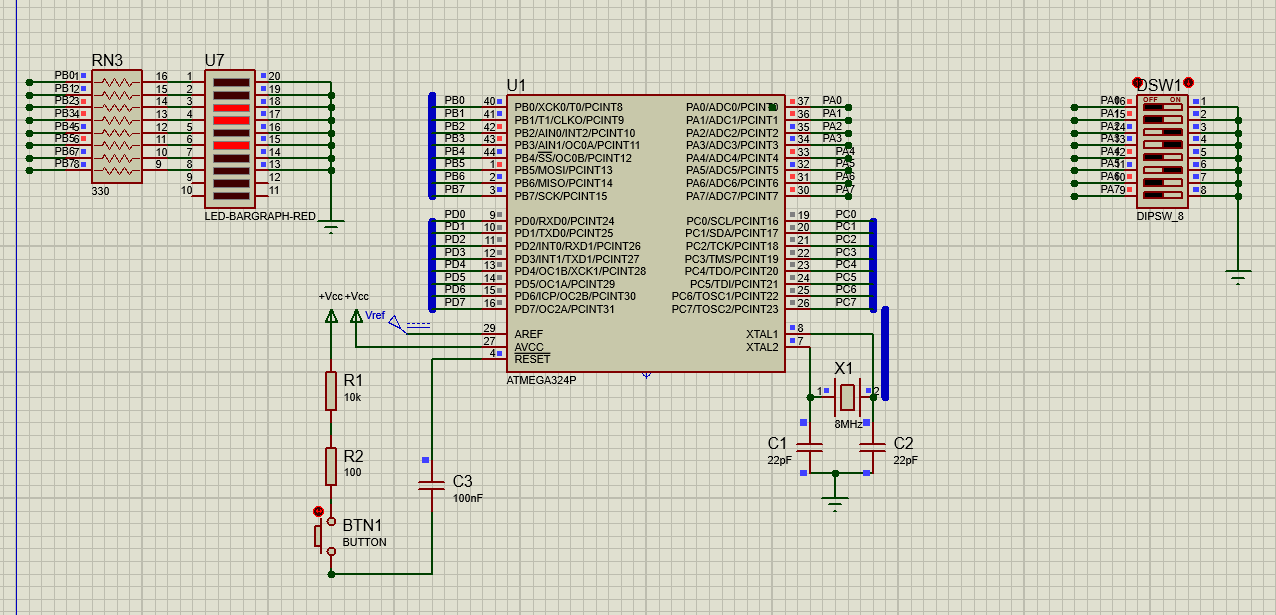
* 1. **When the switch is in the ON/OFF state, what is the pin value of the port?**

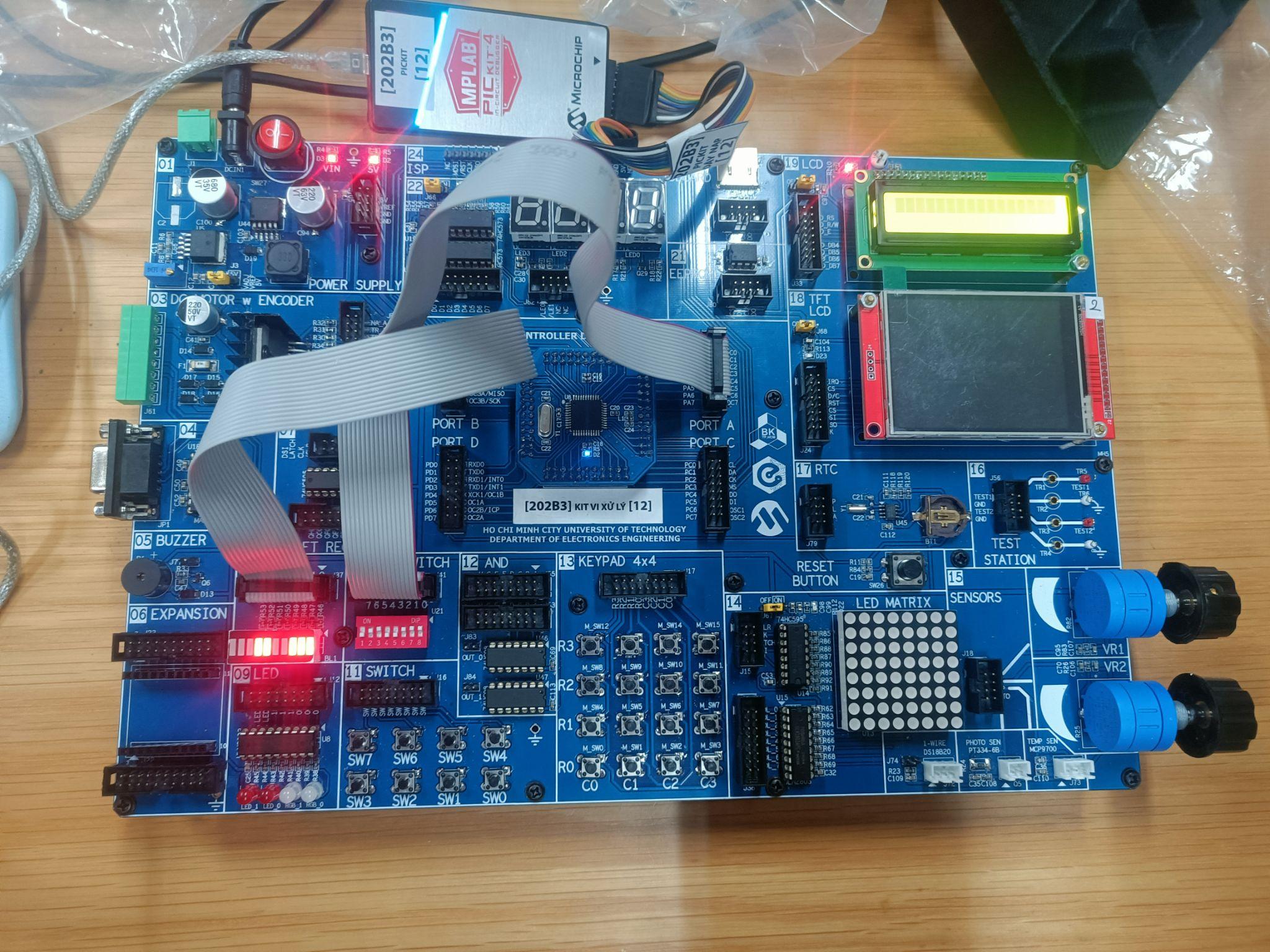
When the switch is ON, the pin value of the corresponding port pin (connected to the dip switch) would be LOW (0). When the switch is OFF, the pin value would be HIGH (1)

* 1. **When the port pin is in state 1, is the BAR LED on or off?**

The behavior of the LED bar (connected to PORT B) depends on how it is configured. If it's configured such that a logic HIGH (1) turns the LED on and a logic LOW (0) turns it off, then when the port pin connected to the LED bar is in state 1 (HIGH), the LED will be ON.

* 1. **Source code with commentsư**

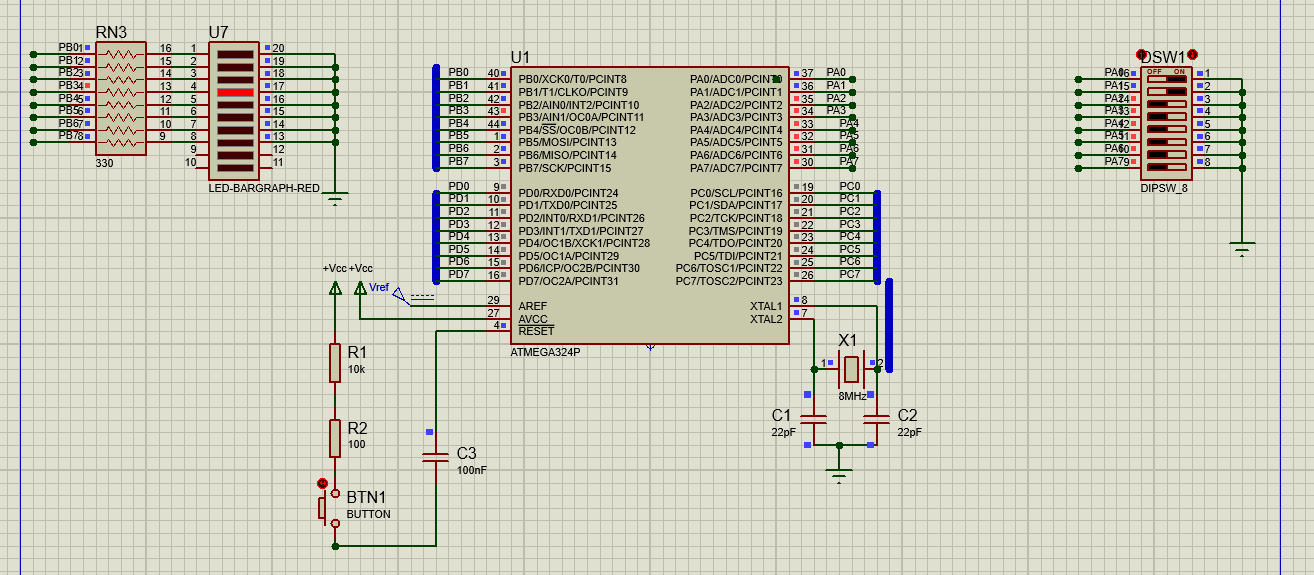


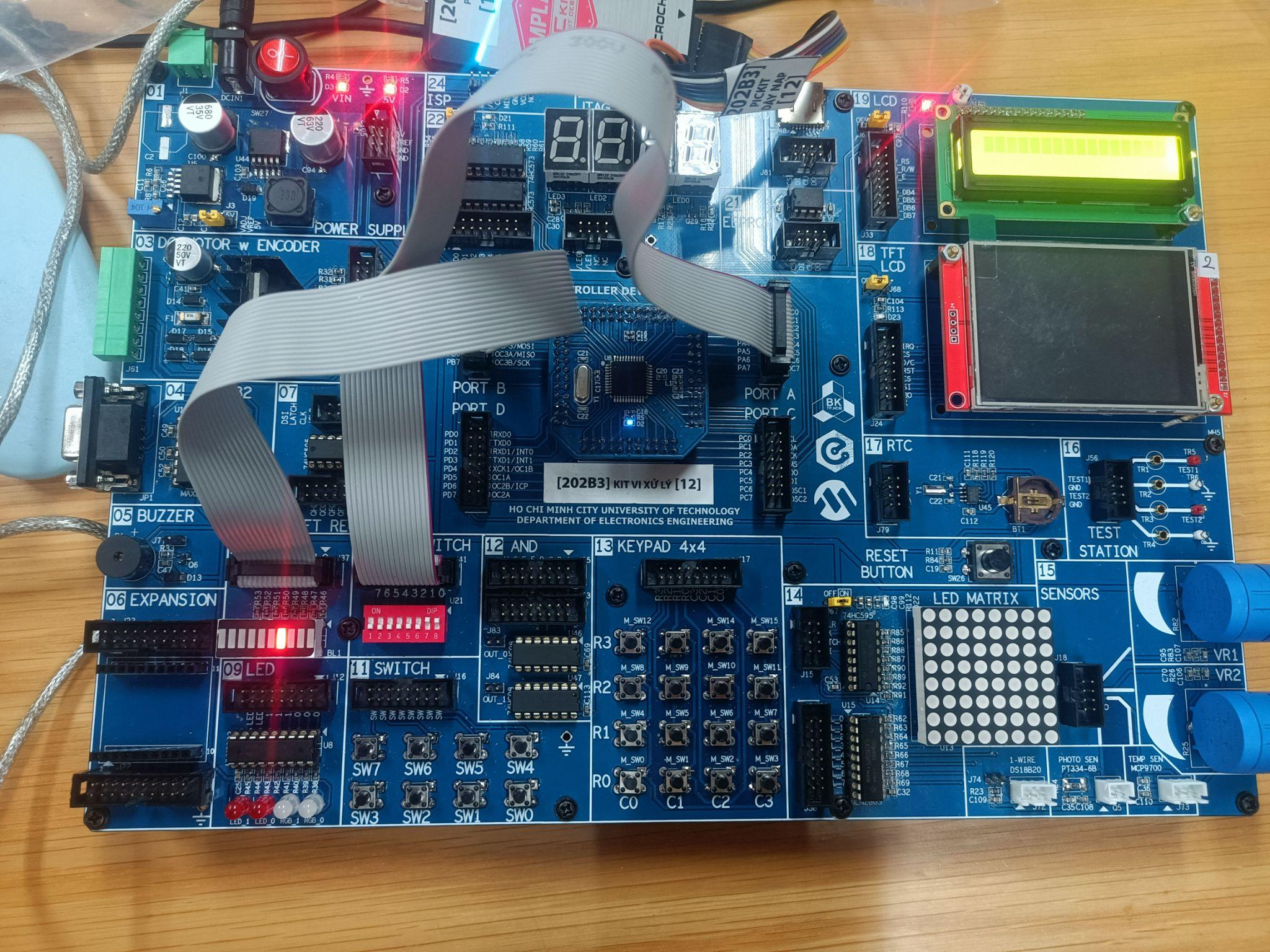


| LDI R16, 0x00 ; Initialize R16 with 0  OUT DDRA, R16 ; Set all bits in DDRA (PORTA data direction register) to 0 (input)    LDI R16, $FF ; Initialize R16 with 0xFF (binary 11111111)  OUT PORTA, R16 ; Set all bits in PORTA to high (enable pull-up resistors for input)    OUT DDRB, R16 ; Set all bits in DDRB (PORTB data direction register) to 0 (output)    start:IN R16, PINA ; Read the input from PORTA and store it in R16  COM R16  OUT PORTB, R16 ; Output the value in R16 to PORTB (LED bar)    rjmp start ; Jump back to the start of the loop |
| --- |

# EXPERIMENT 2

1. Answer the following questions:
   1. **Source code with comments.**





| LDI R16, $00 ; Initialize R16 with 0  OUT DDRA, R16 ; Set all bits in DDRA (PORTA data direction register) to 0 (input)    LDI R16, $FF ; Initialize R16 with 0xFF (binary 11111111)  OUT PORTA, R16 ; Set all bits in PORTA to high (enable pull-up resistors for input)    OUT DDRB, R16 ; Set all bits in DDRB (PORTB data direction register) to 0 (output)    start: LDI R17, 0x05 ; Load the value 0x05 into R17 for addition  ; MAIN    IN R16, PINA ; Read the input from PORTA and store it in R16  COM R16  ADD R16, R17 ; Add the value in R17 (0x05) to R16    OUT PORTB, R16 ; Output the result in R16 to PORTB    rjmp start ; Jump back to the start of the loop |
| --- |

# EXPERIMENT 3

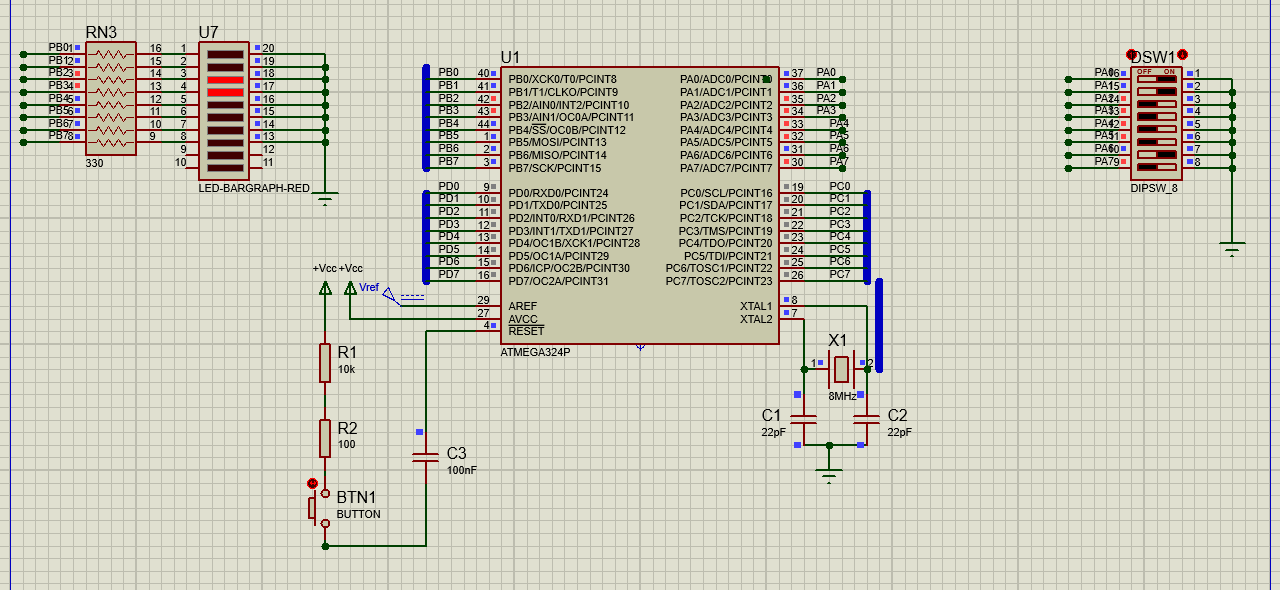
1. Answer the following questions:
   1. **How do you retrieve values from the two nibbles of PORT A?**

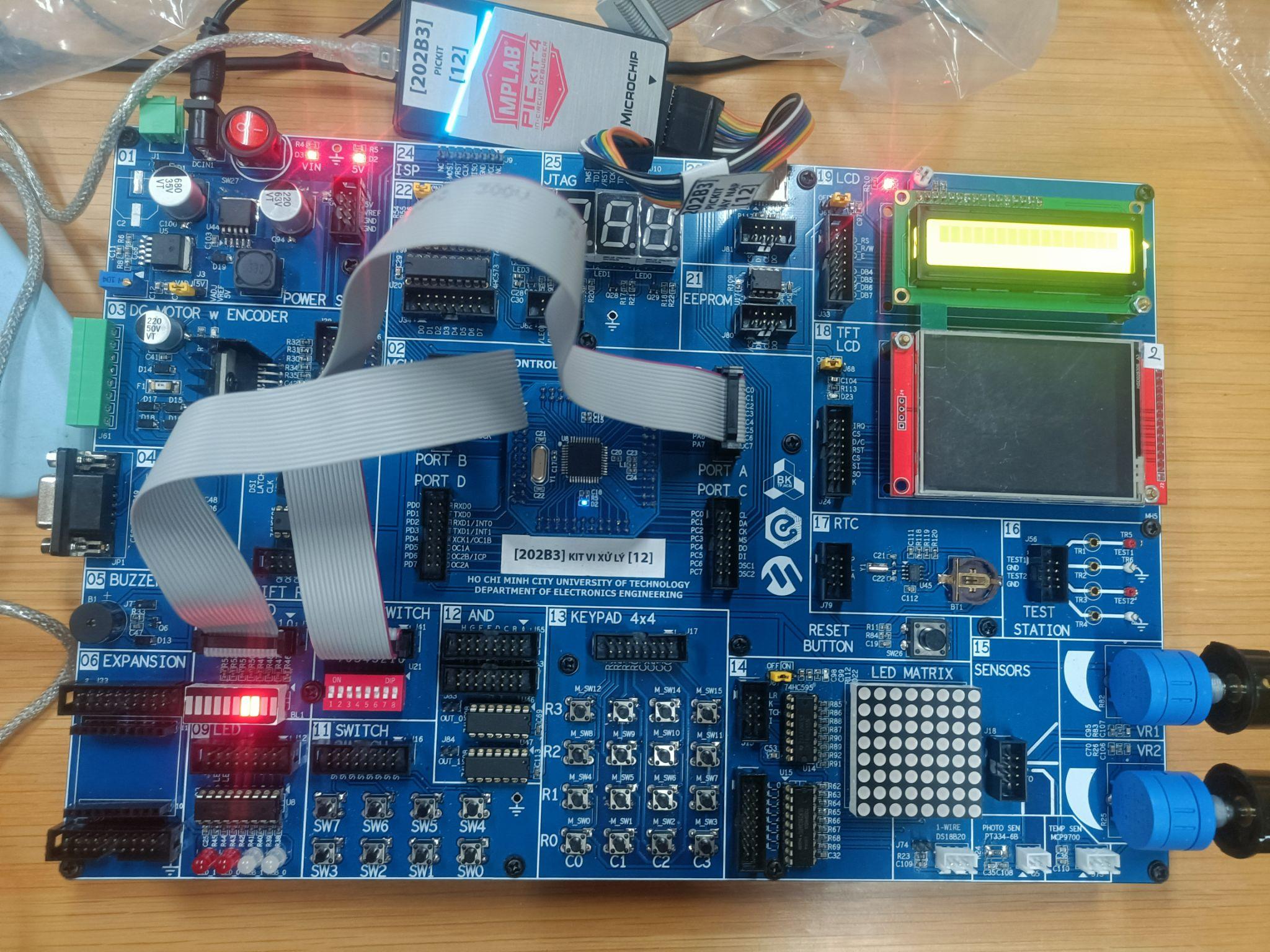
PortA has two nibbles (low and high), to get low\_nibble, we AND portA with 0x0f (00001111), vice-versa we AND 0xf0 (11110000) to get high\_nibble, and then we swap high\_nibble (0000xxxx).

* 1. **Source code with comments.**

Example: 3 \*4 =12

PORT A = 0b0011\_0100, then PORT B = 3 \* 4 = (12)10 = (0000\_1100)2 .





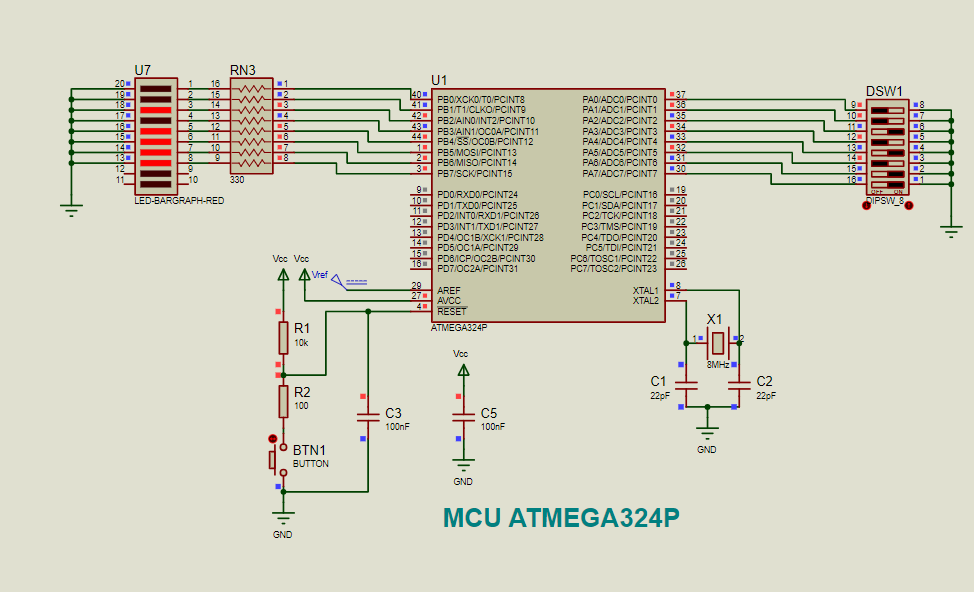
| LDI R16, 0X00  OUT DDRA, R16 ;set portA as input  LDI R16, 0xFF  OUT PORTA, R16 ; put-up resistor at portA  OUT DDRB, R16 ;set portB as output    start: IN R16, PINA ;pinA as switch  COM R16  MOV R17, R16 ;copy content in R16 to R17  ANDI R16, 0x0F ;take low\_nibble  ANDI R17, 0xF0 ;take high\_nibble  SWAP R17 ;inchange high\_nibble and low\_nibble  MUL R16, R17 ;multiply two nibbles  OUT PORTB, R0 ;out result to LED    rjmp start ;loop |
| --- |

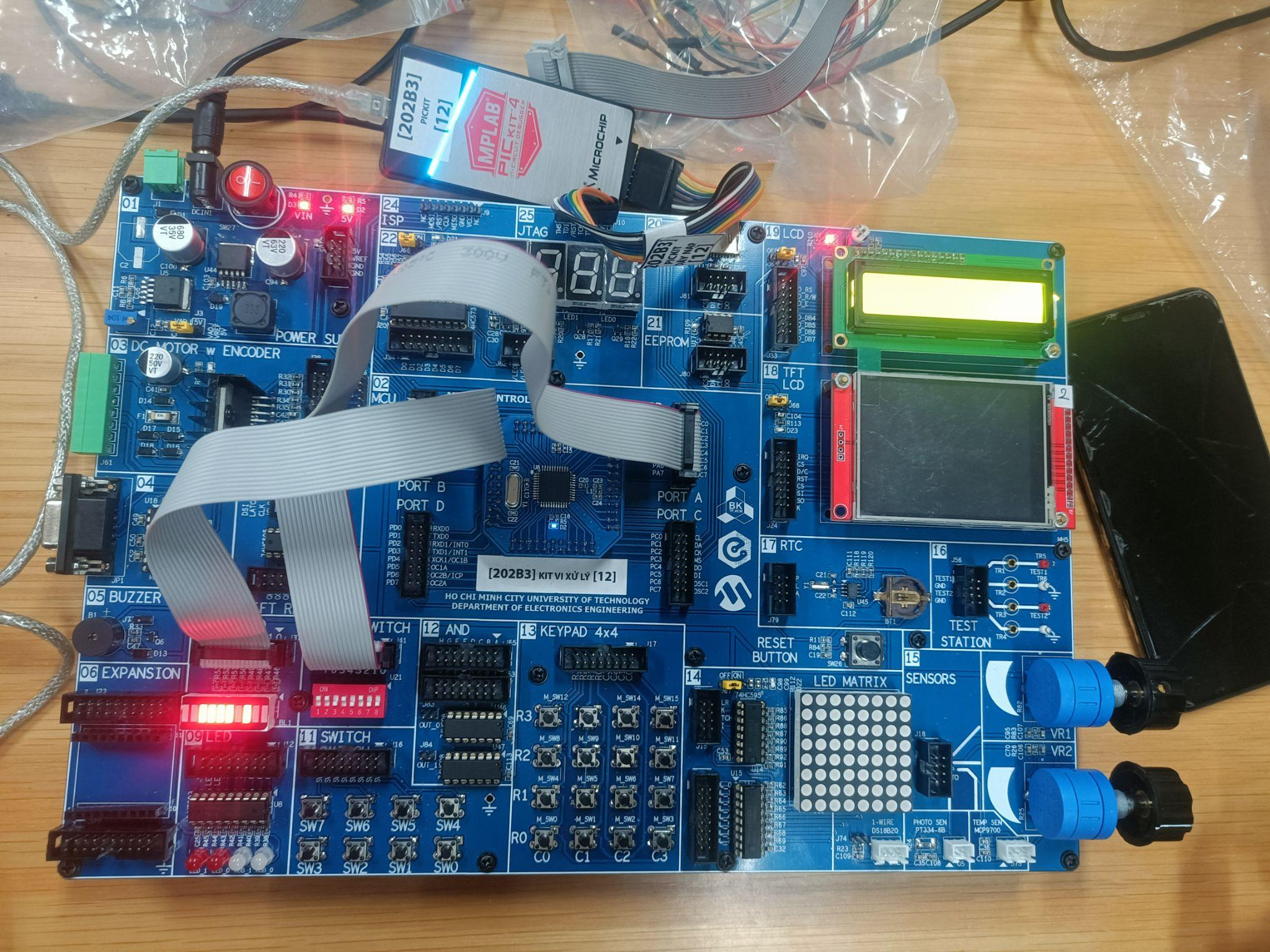
# EXPERIMENT 4

1. Answer the following questions:
   1. Source code with comments.

Example: (-3) \* 4 = (-12)10 =(1111\_0100)2

PORT A = 0b1101\_0100, then PORT B = (-3) \* 4 = (-12)10 =(1111\_0100)2





| LDI R16, $00  OUT DDRA, R16  LDI R16, $FF  OUT PORTA, R16  LDI R16, 0xFF  OUT DDRB, R16  LDI R16, $00  OUT PORTB, R16  ; STATUS INPUT  LOOP:  IN R16, PINA  COM R16  MOV R17, R16  ANDI R16, $F0 ;4 BIT HIGH  SWAP R16  MOV R18, R16  RCALL EXTEND  MOV R16, R18  ANDI R17, $0F ;4 BIT LOW  MOV R18, R17  RCALL EXTEND  MULS R16, R17  OUT PORTB, R0  RJMP LOOP  EXTEND:  SBRS R18, 3  RET  ORI R18, 0xF0  RET |
| --- |

# EXPERIMENT 5

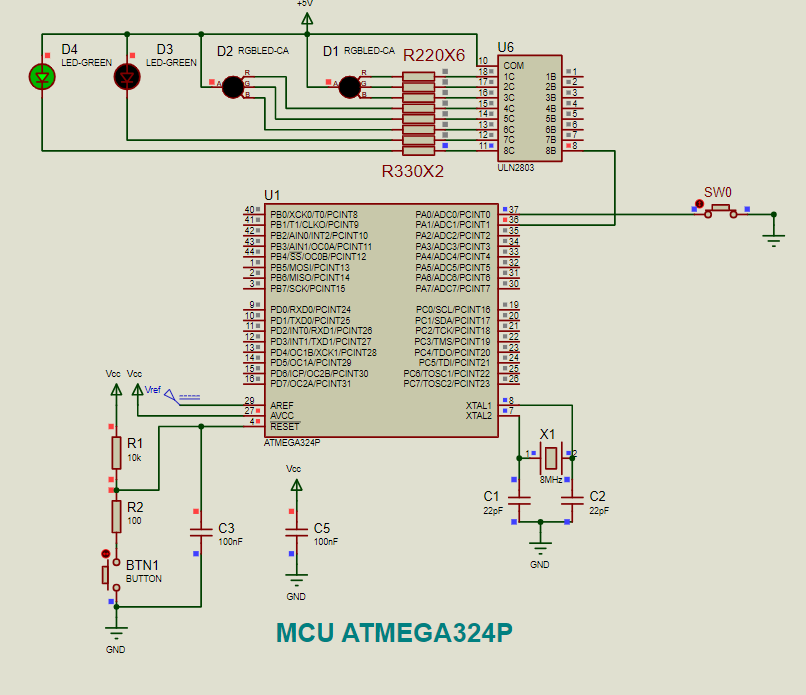
1. Answer the following questions:
   1. **When the switch is pressed/released, what is the pin value of the port?**

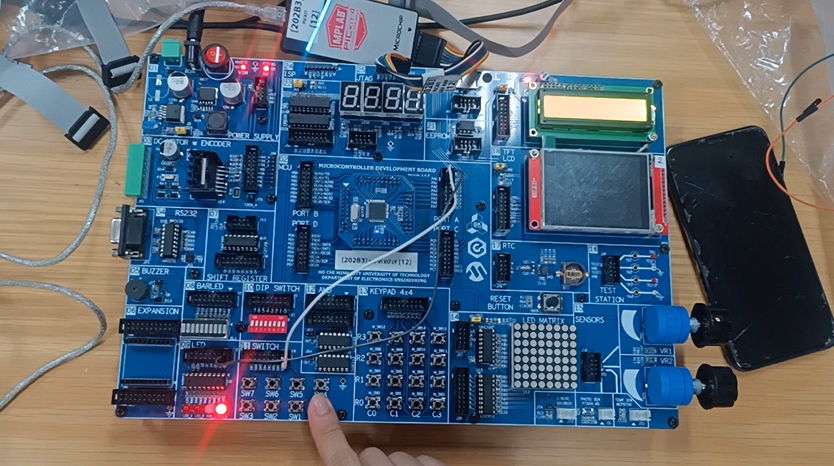
When the switch is pressed, the PA0 =1 and LED is turn on PA1=1 while the switch is released, PA0=0, and LED is turn off, PA1=0.

* 1. **To make the LED light up, what logic level should the port pin output?**

To make the LED light up, you should set the PA1 pin to a logic level of 1 (high) because you mentioned that PA1=1 when the switch is pressed and the LED should be turned on in that case. PA0's logic level does not affect the LED in your description; it is primarily related to the switch state. So, to turn the LED on, you need to set PA1 to a logic high (1) and PA0 can be at either level (0 or 1) depending on the switch state.

* 1. **Source code with comments.**

****

****

| start:  CBI DDRA, 0  SBI PORTA, 0  SBI DDRA, 1  CBI PORTA, 1  MAIN:  SBIS PINA, 0  RJMP RELEASE  PRESSED:  CBI PORTA, 1  RJMP MAIN  RELEASE:  SBI PORTA, 1  RJMP MAIN  LDI R16, 100  DELAY\_LOOP:  DEC R16  BRNE DELAY\_LOOP |
| --- |