The goal of the assignment is use GPIO, delays, and Interrupts:

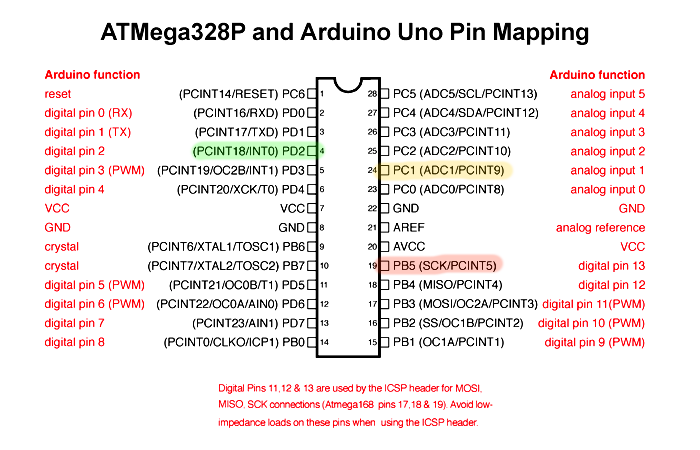
1. Design a delay subroutine to generate a delay of 0.15 sec.

2. Connect a switch to PORTC.1 to poll for an event to turn on the led at PORTB.5 for 1.5 sec after the event.

3. Continue with task 2, connect a switch to INT0 (PD2 pin) (active high - turn on the pull up transistor) and using an interrupt mechanism turn on the led at PORTB.5 for 3 sec after the event.

4. Submit codes and demos for three tasks, one for (2), another for (3), and another for (2 & 3) working together. Verify the delays using simulation and logic analyzer.

**Components Used/Connected**



**Task 1**

1. ; AssemblerApp\_DA2\_1.asm

2. ;

3. ; Created: 3/9/2025 7:21:43 PM

4. ; Author : enriq

5. ;

6.

7. .include "M328Pdef.inc"

8. .equ Value = 0xFB2C ;Timing value for 0.15 sec delay

9.

10. Main:

11. rcall Delay ; Call the 0.15 second delay

12. nop

13. Done:

14. rjmp Done ; Loop forever

15.

16. Delay: ; ~0.15 second delay using Timer1

17. ldi R30, high(Value)

18. sts TCNT1H, R30

19. ldi R30, low(Value)

20. sts TCNT1L, R30 ; Load TCNT1 with starting value

21.

22. ldi R31, 0x00

23. sts TCCR1A, R31 ; Normal mode

24.

25. ldi R31, 0x05 ; Prescaler 1024

26. sts TCCR1B, R31 ; Start Timer1

27.

28. Wait:

29. sbis TIFR1, TOV1 ; Wait for overflow flag

30. rjmp Wait

31.

32. sbi TIFR1, TOV1 ; Clear overflow flag

33. ldi R30, 0x00

34. sts TCCR1B, R30 ; Stop Timer1

35.

36. ret

37.

A screenshot of a computer program

AI-generated content may be incorrect.

1. /\*

2. \* GccApp\_DA2\_1.c

3. \*

4. \* Created: 3/9/2025 7:55:28 PM

5. \* Author : enriq

6. \*/

7.

8. #define F\_CPU 8000000UL

9.

10. #include <avr/io.h>

11. #include <util/delay.h>

12.

13. int main(void)

14. {

15. \_delay\_ms(150);

16.

17. return 0;

18. }

19.

A screenshot of a computer program

AI-generated content may be incorrect.

**Task 2**

1. /\*

2. \* DA2\_2\_C.c

3. \*

4. \* Created: 3/21/2025 8:58:26 PM

5. \* Author : enriq

6. \*/

7.

8. #include <avr/io.h>

9.

10.

11. #define F\_CPU 8000000UL

12.

13. #include <avr/io.h>

14. #include <util/delay.h>

15.

16. int main(void)

17. {

18. // Set PORTB5 as output (for LED)

19. DDRB |= (1 << PORTB5);

20.

21. // Set PORTC1 as input (for switch)

22. DDRC &= ~(1 << PORTC1);

23.

24. PORTC |= (1 << PORTC1);

25.

26. while (1)

27. {

28. // check if pressed (low)

29. if (!(PINC & (1 << PINC1)))

30. {

31. // Turn LED on

32. PORTB |= (1 << PORTB5);

33.

34. // Delay for 1.5 seconds

35. \_delay\_ms(1500);

36.

37. // Turn LED off

38. PORTB &= ~(1 << PORTB5);

39. }

40. }

41.

42. return 0;

43. }

44.

A screenshot of a computer

AI-generated content may be incorrect.

1. ;

2. ; DA2\_2\_asm.asm

3. ;

4. ; Created: 3/21/2025 11:29:28 PM

5. ; Author : enriq

6. ;

7.

8. .include "M328Pdef.inc"

9. .equ Value = 0xFB2C ; Timer preload for ~0.15 sec

10. .equ DelayCount = 10 ; 10 \* 0.15 sec = 1.5 sec

11.

12.

13. Main:

14. ; Set PORTB5 as output

15. sbi DDRB, DDB5

16.

17. ; Set PORTC1 as input

18. cbi DDRC, DDC1

19.

20. ; Enable internal pull-up on PORTC1

21. sbi PORTC, PORTC1

22.

23. Loop:

24. ; Check if button on PORTC1 is pressed (low)

25. sbic PINC, PINC1 ; Skip next if PINC1 is pressed

26. rjmp Loop ; Not pressed, loop again

27.

28. ; Turn on LED on PORTB5

29. sbi PORTB, PORTB5

30.

31. ; Delay for 1.5 seconds using 10x 0.15 sec Delay

32. ldi R20, DelayCount

33. DelayLoop:

34. rcall Delay

35. dec R20

36. brne DelayLoop

37.

38. ; Turn off LED on PORTB5

39. cbi PORTB, PORTB5

40.

41. rjmp Loop ; Wait for next press

42.

43. ; 0.15 Second Delay

44.

45. Delay:

46. ldi R30, high(Value)

47. sts TCNT1H, R30

48. ldi R30, low(Value)

49. sts TCNT1L, R30 ; Load TCNT1 with starting value

50.

51. ldi R31, 0x00

52. sts TCCR1A, R31 ; Normal mode

53.

54. ldi R31, 0x05 ; Prescaler 1024

55. sts TCCR1B, R31 ; Start Timer1

56.

57. Wait:

58. sbis TIFR1, TOV1 ; Wait for overflow flag

59. rjmp Wait

60.

61. sbi TIFR1, TOV1 ; Clear overflow flag

62. ldi R30, 0x00

63. sts TCCR1B, R30 ; Stop Timer1

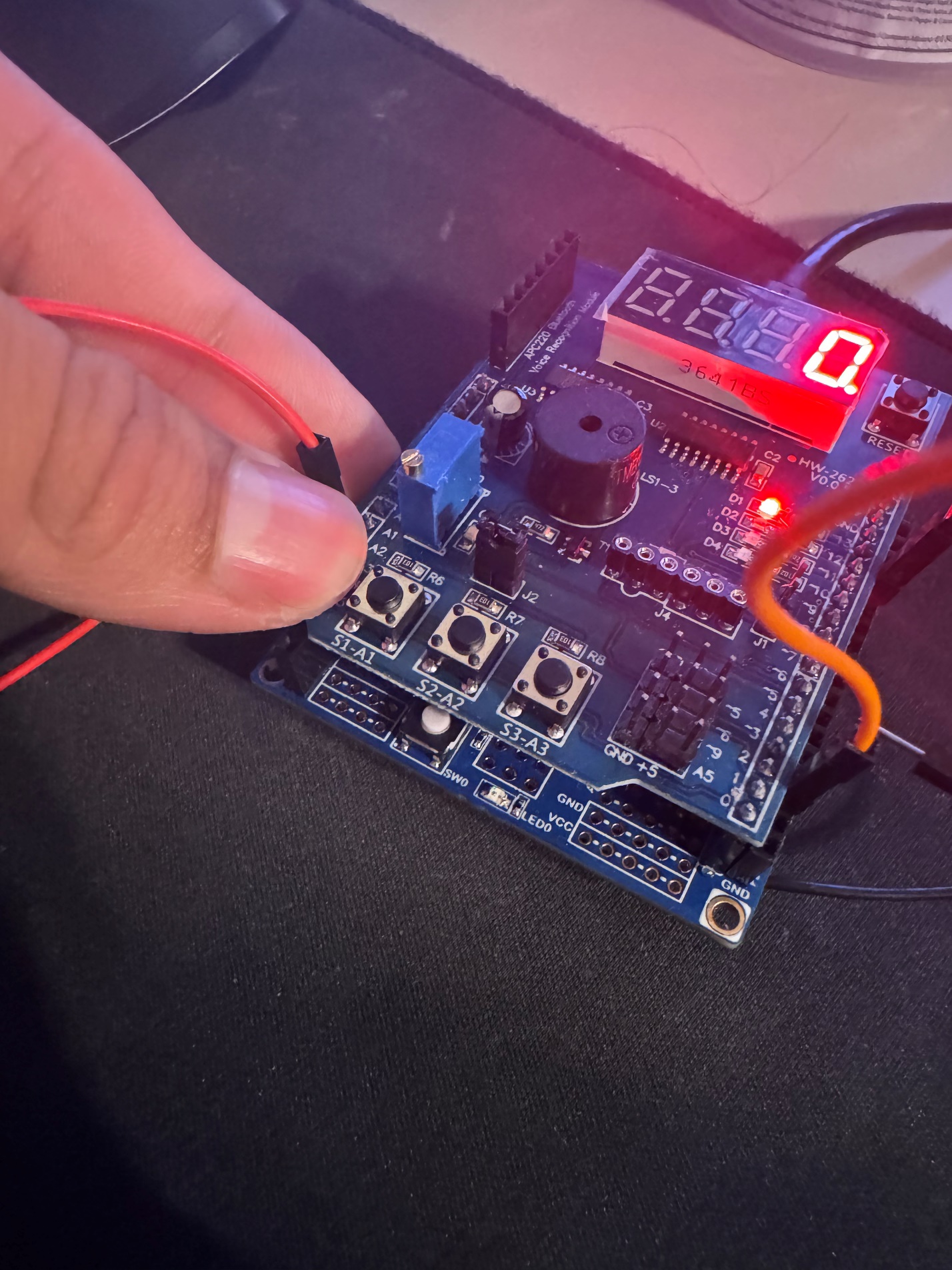
64.

65. ret

66.

A screenshot of a computer program

AI-generated content may be incorrect.

A hand holding a circuit board

AI-generated content may be incorrect.

**Task 3**

1. /\*

2. \* DA2\_3\_C.c

3. \*

4. \* Created: 3/21/2025 11:46:25 PM

5. \* Author : enriq

6. \*/

7.

8. #define F\_CPU 8000000UL

9.

10. #include <avr/io.h>

11. #include <avr/interrupt.h>

12. #include <util/delay.h>

13.

14.

15. void setup()

16. {

17. DDRB |= (1 << DDB5); // LED as output

18.

19. DDRD &= ~(1 << DDD2); // PD2 as input

20. PORTD |= (1 << PORTD2); // Enable internal pull-up

21.

22. // INT0 triggers on falling edge

23. EICRA |= (1 << ISC01);

24. EICRA &= ~(1 << ISC00);

25.

26. EIMSK |= (1 << INT0); // Enable INT0

27.

28. sei(); // Global interrupt enable

29. }

30.

31. int main(void)

32. {

33. setup();

34.

35. while (1)

36. {

37.

38. }

39. }

40.

41. ISR(INT0\_vect)

42. {

43. PORTB |= (1 << PORTB5); // LED ON

44. \_delay\_ms(3000); // Wait 3 seconds

45. PORTB &= ~(1 << PORTB5); // LED OFF

46. }

47.

A screenshot of a computer program

AI-generated content may be incorrect.

1. ;

2. ; DA2\_3\_asm.asm

3. ;

4. ; Created: 3/22/2025 6:27:19 PM

5. ; Author : enriq

6. ;

7.

8. .include "M328Pdef.inc"

9. .equ Value = 0xFB2C ; Timer preload for ~0.15 sec

10. .equ DelayCount = 20 ; 20 x 0.15 s = 3 s total

11.

12. .cseg

13. .org 0x0000

14. rjmp Setup

15.

16. .org 0x0002 ; INT0 vector

17. rjmp INT0\_ISR

18.

19.

20. Setup:

21. ldi r16, high(RAMEND)

22. out SPH, r16

23. ldi r16, low(RAMEND)

24. out SPL, r16

25.

26. sbi DDRB, DDB5 ; Set PB5 as output

27.

28. cbi DDRD, DDD2 ; PD2 input

29. sbi PORTD, PORTD2 ; Enable pull-up on PD2

30.

31.

32. ; set INT0

33. ldi r16, (1 << ISC01)

34. sts EICRA, r16

35.

36. ; Enable INT0 interrupt

37. ldi r16, (1 << INT0)

38. sts EIMSK, r16

39.

40. sei ; Global interrupt enable

41.

42. Main:

43. rjmp Main

44.

45. INT0\_ISR:

46. sbi PORTB, PORTB5 ; Turn on LED at PB5

47. ldi r20, DelayCount

48.

49. DelayLoop:

50. rcall Delay

51. dec r20

52. brne DelayLoop

53.

54. ; Turn off LED at PB5

55. cbi PORTB, PORTB5

56.

57. reti

58.

59.

60. Delay: ; ~0.15 second delay using Timer1

61. ldi R30, high(Value)

62. sts TCNT1H, R30

63. ldi R30, low(Value)

64. sts TCNT1L, R30 ; Load TCNT1 with starting value

65.

66. ldi R31, 0x00

67. sts TCCR1A, R31 ; Normal mode

68.

69. ldi R31, 0x05 ; Prescaler 1024

70. sts TCCR1B, R31 ; Start Timer1

71.

72. Wait:

73. sbis TIFR1, TOV1 ; Wait for overflow flag

74. rjmp Wait

75.

76. sbi TIFR1, TOV1 ; Clear overflow flag

77. ldi R30, 0x00

78. sts TCCR1B, R30 ; Stop Timer1

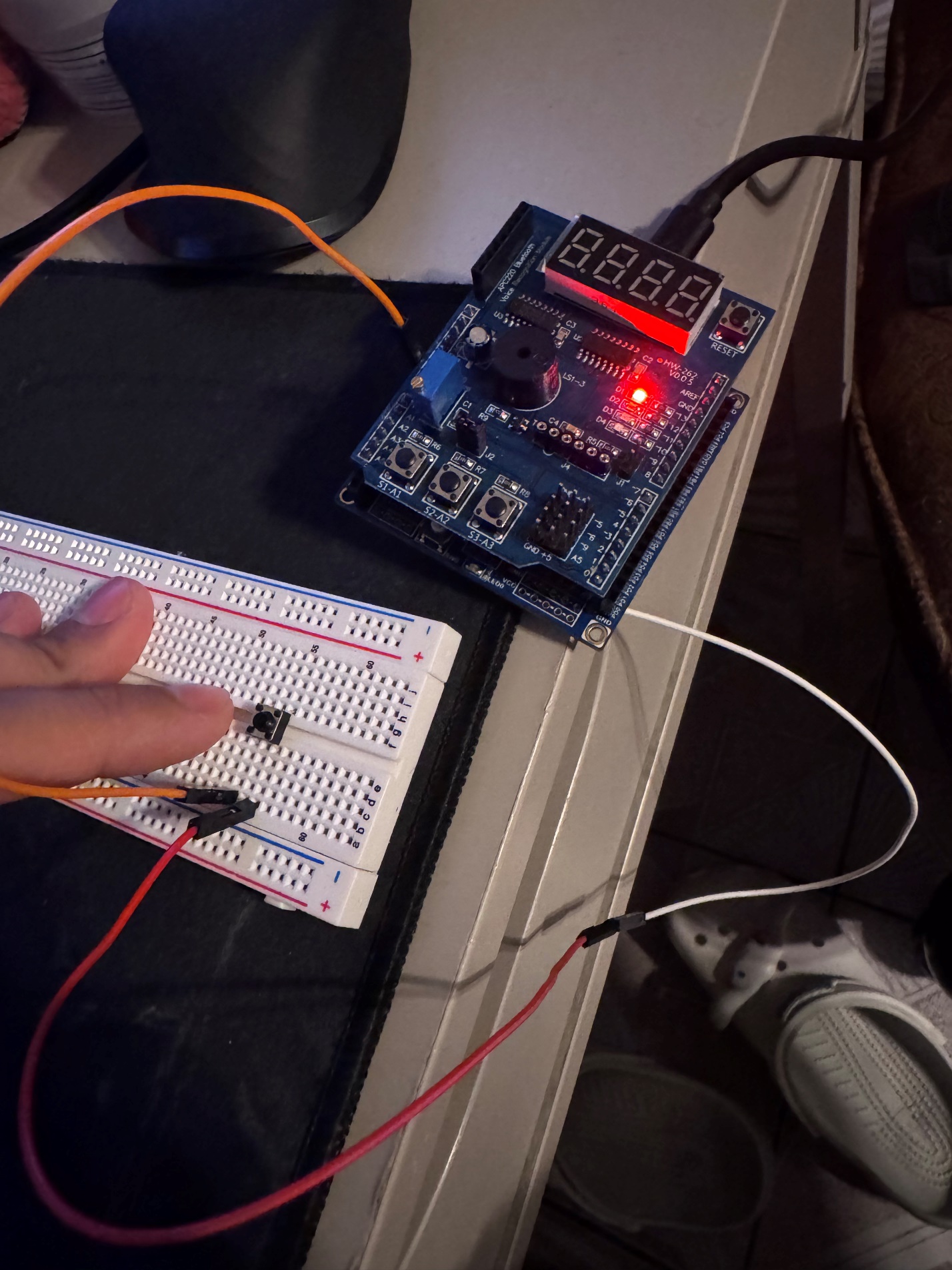
79.

80. ret

81.

A screenshot of a computer program

AI-generated content may be incorrect.

A hand holding a white board with orange wires

AI-generated content may be incorrect.