

# LAB 4-C

## 7-SEGMENT

### OBJECTIVES:

- ② To interface seven-segment to the AVR simulator.

### MATERIAL:

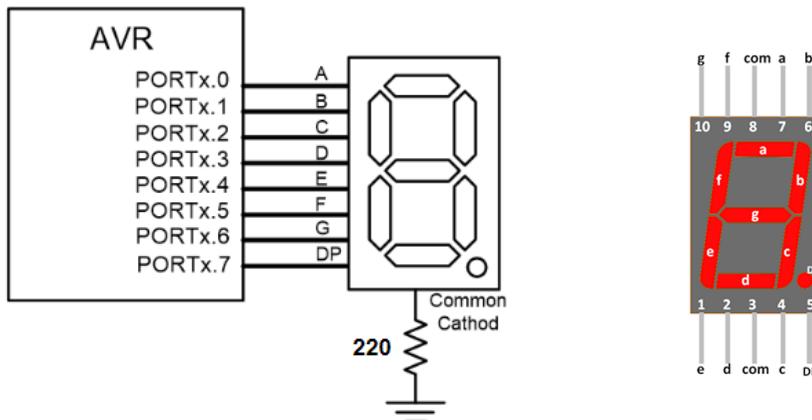
- ② Atmel Studio or the assembler of your choice.
- ② [https://lcgamboa.github.io/js/picsimlab.html?..../picsimlab\\_examples/](https://lcgamboa.github.io/js/picsimlab.html?..../picsimlab_examples/) (Simulator)

### WEB SITES:

- ② [www.microchip.com](http://www.microchip.com) for Atmel Studio Software

### ACTIVITY 1

- Connect a common cathode 7-segment directly to PORTD.



- Write the following program in the AVR Studio, build and download to the picsimlab.

```
LDI    R20, 0xFF
OUT   DDRD, R20
LDI    R20, 0b00000111
OUT   PORTD, R20
HERE: RJMP HERE
```

### Analysis:

- The code loads 0b00000111 into PORTD.
- Assuming the standard segment connection (Bit 0 = a, Bit 1 = b, ... Bit 6 = g), this activates segments a, b, and c.
- Observation: The 7-segment display will show the number 7.

## LAB 4-C

### 7-SEGMENT

#### ACTIVITY 2

Modify the previous program to display 9 on the 7-segment.

Logic: To display '9', we need segments a, b, c, d, f, g to be ON.

- Binary: 0110 1111 (g, f, e, d, c, b, a, p) → Note: segment 'e' is OFF.
- Hex: \$6F

Code:

```
.INCLUDE "m328pdef.inc"
```

```
.ORG 0x0000
LDI R20, 0xFF
OUT DDRD, R20      ; Set PORTD as Output
```

```
LDI R20, 0x6F      ; Load hex code for '9' (Binary 01101111)
OUT PORTD, R20    ; Output to 7-segment
```

HERE: RJMP HERE

#### ACTIVITY 3

Use a **look-up table** to write a subroutine that displays the value stored in R21 on the 7-segment.

```
.INCLUDE "m328pdef.inc"
```

```
.CSEG
.ORG 0x0000
; Initialize Stack Pointer
LDI R16, LOW(RAMEND)
OUT SPL, R16
LDI R16, HIGH(RAMEND)
OUT SPH, R16
```

```
; Configure PORTD as Output
LDI R16, 0xFF
OUT DDRD, R16
```

```
; Example: Display '5'
LDI R21, 5
RCALL DISPLAY_DIGIT
```

HERE: RJMP HERE

## LAB 4-C

### 7-SEGMENT

```
; -----
; Subroutine: Display value in R21 to PORTD
; -----
DISPLAY_DIGIT:
    LDI ZL, LOW(2*SEG_TABLE) ; Load Z pointer with table address
    LDI ZH, HIGH(2*SEG_TABLE)

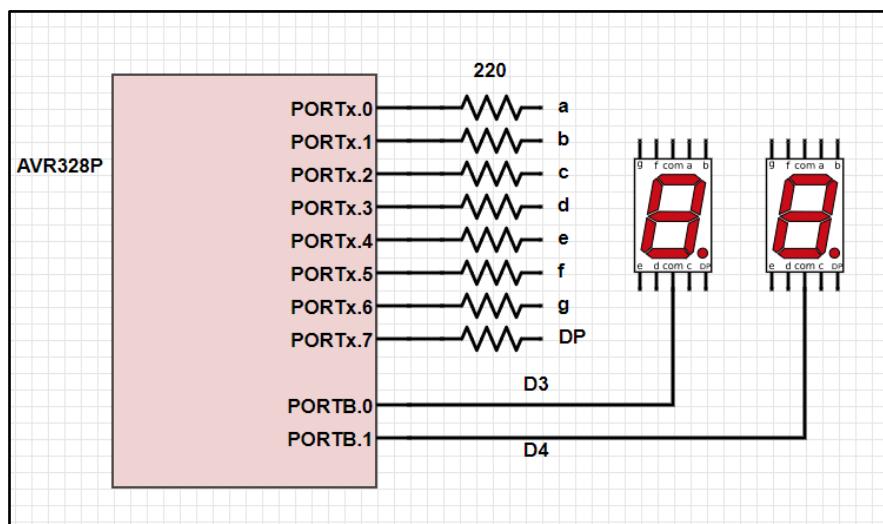
    ADD ZL, R21           ; Add offset (the number to display)
    LDI R16, 0
    ADC ZH, R16           ; Handle carry for high byte

    LPM R16, Z            ; Load bit pattern from Flash
    OUT PORTD, R16         ; Send to 7-segment
    RET

; -----
; Look-up Table (0-9 Common Cathode Codes)
; -----
SEG_TABLE:
    .DB 0x3F, 0x06, 0x5B, 0x4F, 0x66 ; 0, 1, 2, 3, 4
    .DB 0x6D, 0x7D, 0x07, 0x7F, 0x6F ; 5, 6, 7, 8, 9
```

#### ACTIVITY 4

- a) Connect two 7-segments to the same port of AVR, as shown below.



# LAB 4-C

## 7-SEGMENT



- b) Display 23 by scanning on those two 7-segments.

- Connections (based on source):
  - Segments connected to PORTD (assuming PORTx in diagram corresponds to Data Port).
    - Common Cathode Digit 1 (Left) connected to PORTB.0.
    - Common Cathode Digit 2 (Right) connected to PORTB.1.
- Logic:
  1. Output '2' to PORTD → Activate Digit 1 (PB0=0, PB1=1) → Delay.
  2. Output '3' to PORTD → Activate Digit 2 (PB0=1, PB1=0) → Delay.
  3. Repeat quickly to create persistence of vision.

Code:

```
.INCLUDE "m328pdef.inc"

.EQU DIGIT_1_PIN = 0 ; PORTB.0 for Left Digit
.EQU DIGIT_2_PIN = 1 ; PORTB.1 for Right Digit
```

```
.ORG 0x0000
; 1. Configure Ports
LDI R16, 0xFF
OUT DDRD, R16 ; PORTD as Output (Segments)
OUT DDRB, R16 ; PORTB as Output (Control)
```

LOOP\_SCAN:

```
; -----
; Display '2' on Left Digit (Digit 1)
; -----
```

## LAB 4-C

### 7-SEGMENT

```
LDI R16, 0x5B    ; Segment code for '2'  
OUT PORTD, R16
```

```
CBI PORTB, DIGIT_1_PIN ; Enable Digit 1 (Low)  
SBI PORTB, DIGIT_2_PIN ; Disable Digit 2 (High)
```

```
RCALL DELAY_SHORT    ; Small delay for visibility
```

```
; -----  
; Display '3' on Right Digit (Digit 2)  
; -----  
LDI R16, 0x4F    ; Segment code for '3'  
OUT PORTD, R16
```

```
SBI PORTB, DIGIT_1_PIN ; Disable Digit 1 (High)  
CBI PORTB, DIGIT_2_PIN ; Enable Digit 2 (Low)
```

```
RCALL DELAY_SHORT    ; Small delay
```

```
RJMP LOOP_SCAN     ; Repeat
```

```
; -----  
; Short Delay Subroutine  
; -----  
DELAY_SHORT:  
    LDI R17, 20    ; Adjust this value for scan speed  
    D1: LDI R18, 255  
    D2: DEC R18  
        BRNE D2  
        DEC R17  
        BRNE D1  
        RET
```