Group A: Basic Up/Down Counters

Fundamental counting logic for LEDs.

1. Standard Up-Counter (0-255)

The base code from Lab 3-C Activity 1. Counts 0x00 to 0xFF then rolls over.

Code snippet

```
MAIN_UP:
    INC R20              ; Increment Counter
    OUT PORTB, R20       ; Display
    RCALL DELAY          ; Wait for visibility
    RJMP MAIN_UP         ; Repeat forever
```

2. Standard Down-Counter (255-0) *Counts backwards.*

Code snippet

```
MAIN_DOWN:
    DEC R20              ; Decrement Counter
    OUT PORTB, R20
    RCALL DELAY
    RJMP MAIN_DOWN
```

3. Bounded Up-Counter (0 to Limit)

Counts 0 to 9, then resets to 0. Essential for Decimal counters.

Code snippet

```
MAIN_LIMIT:
    INC R20
    OUT PORTB, R20
    RCALL DELAY

    CPI R20, 10          ; Check Limit (10)
    BRNE MAIN_LIMIT      ; If != 10, keep counting
    CLR R20              ; Else Reset to 0
    RJMP MAIN_LIMIT
```

4. Bounded Down-Counter (Limit to 0) *Counts 9 down to 0, then resets to 9.*

Code snippet

```
MAIN_DOWN_LIMIT:
    OUT PORTB, R20
    RCALL DELAY
    DEC R20

    CPI R20, 0xFF          ; Check Underflow (0 -> -1)
    BRNE MAIN_DOWN_LIMIT
    LDI R20, 9             ; Reset to Top
    RJMP MAIN_DOWN_LIMIT
```

Group B: Advanced Counter Logic

Counters that react to inputs (Scenario 1).

5. The "Smart" Directional Counter *Switch 0 OFF = Count UP. Switch 0 ON = Count DOWN.*

Code snippet

```
SMART_CNT:
    SBIC PINC, 0           ; Check Switch
    RJMP GO_DOWN           ; If Pressed (0), Down
    INC R20                ; Else Up
    RJMP UPDATE_DISP

GO_DOWN:
    DEC R20

UPDATE_DISP:
    OUT PORTB, R20
    RCALL DELAY
    RJMP SMART_CNT
```

6. The "Pause" Counter *Counts only when Switch is NOT pressed. Freezes when pressed.*

Code snippet

```
PAUSE_CNT:
    SBIC PINC, 0           ; Check Pause Button
    RJMP SKIP_INC          ; If Pressed, Skip Increment
    INC R20                ; Count Up

SKIP_INC:
    OUT PORTB, R20
    RCALL DELAY
    RJMP PAUSE_CNT
```

7. Dynamic Limit Counter (Set by Switches)

Counts up to the binary value currently set on PORTC switches.

Code snippet

```
DYN_LIMIT:
    IN R19, PINC        ; Read Limit from Switches
    INC R20
    OUT PORTB, R20
    RCALL DELAY

    CP R20, R19         ; Compare Count vs Limit
    BRLO DYN_LIMIT      ; If Lower, Keep counting
    CLR R20             ; Else Reset
    RJMP DYN_LIMIT
```

8. One-Shot Counter (Click-to-Count) *Increments exactly once per button press.*

Code snippet

```
CLICK_CNT:
    SBIC PINC, 0        ; Wait for Press
    RJMP CLICK_CNT

    INC R20             ; Increment Once
    OUT PORTB, R20

    ; Debounce / Wait Release
WAIT_REL:
    SBIS PINC, 0
    RJMP WAIT_REL
    RJMP CLICK_CNT
```

Group C: Delay Loops (Timing)

Creating visible pauses.

9. Standard Delay (Constant Time)

The classic triple-nested loop for ~0.5s delay.

Code snippet

```
DELAY:
   LDI R21, 50          ; Outer Loop (Coarse Tune)
L1: LDI R22, 255          ; Middle Loop
L2: LDI R23, 255          ; Inner Loop (Fine Tune)
L3: DEC R23
   BRNE L3
   DEC R22
   BRNE L2
   DEC R21
   BRNE L1
   RET
```

10. Short Delay (Debounce) *Fast delay (~20ms) for switch noise filtering.*

Code snippet

```
DELAY_SHORT:
   LDI R21, 20          ; Just 20 iterations
L1_S: LDI R22, 255
L2_S: DEC R22
   BRNE L2_S
   DEC R21
   BRNE L1_S
   RET
```

11. Variable Speed Delay (Fast/Slow)

Speed depends on Switch Input (Scenario 2).

Code snippet

```
DELAY_VAR:
   SBIC PINC, 0          ; Check Switch
   RJMP FAST_MODE          ; If Pressed, Fast
   LDI R21, 100          ; Slow (Large Number)
   RJMP DO_DELAY
FAST_MODE:
   LDI R21, 10          ; Fast (Small Number)
DO_DELAY:
   ; (Insert Inner Loops L2/L3 here using R21)
   RET
```

12. Dynamic Delay (Speed set by Potentiometer/Switch Bank) *Speed is proportional to the binary value on PORTC.*

Code snippet

```
DELAY_DYN:
   IN R21, PINC         ; Read Delay Value
   TST R21            ; Check if 0 (Avoid infinite fast loop)
   BREQ SAFE_MIN
   RJMP DO_DELAY_DYN
SAFE_MIN:
   LDI R21, 1          ; Minimum safe delay
DO_DELAY_DYN:
   ; (Insert Inner Loops here)
   RET
```

Group D: Pattern Sequences (Chasers)

Using counters to drive visual effects.

13. Ping-Pong Shift (Left then Right) *LED moves L->R then R->L.*

Code snippet

```
PING_PONG:
   LDI R20, 1          ; Start Bit 0

GO_LEFT:
   OUT PORTB, R20
   RCALL DELAY
   LSL R20            ; Shift Left
   BRCC GO_LEFT         ; Keep going until Carry

   LDI R20, 0x80        ; Reset to Top
GO_RIGHT:
   OUT PORTB, R20
   RCALL DELAY
   LSR R20            ; Shift Right
   BRCC GO_RIGHT
   RJMP PING_PONG
```

14. Binary Up-Count Display *Displays 0000 to 1111 (0-15) on lower 4 LEDs.*

Code snippet

```
BIN_COUNT:
  INC R20
  ANDI R20, 0x0F       ; Mask Upper 4 bits
  OUT PORTB, R20
  RCALL DELAY
  RJMP BIN_COUNT
```

15. Johnson Counter (Walking Ring) *Fill 1s then Fill 0s.*

Code snippet

```
JOHNSON:
  SEC               ; Set Carry Flag = 1
  ROL R20             ; Rotate Left through Carry
  OUT PORTB, R20
  RCALL DELAY
  RJMP JOHNSON
```

16. Random Flicker (Pseudo-Random) *Uses a simplified shift for "random" noise.*

Code snippet

```
RANDOM_LED:
  IN R16, TCNT0       ; Read Hardware Timer (Entropy)
  EOR R20, R16        ; XOR with Counter
  OUT PORTB, R20
  RCALL DELAY
  RJMP RANDOM_LED
```

Group E: Special Timing Logic

17. Timeout Timer

- waits for input, but gives up after a while.*

Code snippet

```
WAIT_WITH_TIMEOUT:
  LDI R21, 255        ; Timeout Counter
CHECK_INPUT:
  SBIC PINC, 0        ; Check Input
  RJMP SUCCESS          ; Input Received!

  DEC R21             ; Decrease Timeout
```

```
    BREQ TIMEOUT_FAIL      ; If 0, Failed
    RJMP CHECK_INPUT
```

18. Pulse Width Modulation (Software PWM) *Dims an LED by turning it On/Off rapidly.*

Code snippet

```
PWM_LOOP:
    SBI PORTB, 0          ; On
    LDI R21, 5            ; On Time
    RCALL DELAY_MICRO

    CBI PORTB, 0          ; Off
    LDI R21, 20           ; Off Time
    RCALL DELAY_MICRO
    RJMP PWM_LOOP
```

19. Traffic Light Timer (State Machine) *Red (5s) -> Green (5s) -> Yellow (2s).*

Code snippet

```
TRAFFIC:
    SBI PORTB, 0          ; Red On
    LDI R21, 50           ; Long Delay
    RCALL DELAY_VAR
    CBI PORTB, 0

    SBI PORTB, 1          ; Green On
    LDI R21, 50
    RCALL DELAY_VAR
    CBI PORTB, 1
    RJMP TRAFFIC
```

20. Reaction Timer *Counts how long it takes to press a button.*

Code snippet

```
REACTION_TEST:
    CLR R20               ; Reset Time Counter
WAIT_USER:
    SBIC PINC, 0          ; Wait for Press
    RJMP SHOW_TIME

    INC R20               ; Count Time passing
    RCALL DELAY_SHORT      ; Small increment
    RJMP WAIT_USER

SHOW_TIME:
    OUT PORTB, R20          ; Display Reaction Time
    RJMP SHOW_TIME
```