6. Der Code

6.1. Code Example 1 (Timer0)

This code **toggles PD5 every 0.5s**. The `delay_timer0` subroutine uses **Timer0** in **CTC Mode** with a **1024 prescaler** and a **compare value of 156**, creating a **10ms** hardware delay per call. This subroutine is called **50 times** using software loop with `R18` as the counter.

```
;-----------------------
; Assembly Code
;-----------------------
#define __SFR_OFFSET 0x00
#include "avr/io.h"
;-----------------------
.global main
;===============================================================
    main:
    LDI R16, 0b00100000 ; to toggle PD5
    LDI R17, 0b00000000
    ;-----------------------------------------
    SBI DDRD, 5         ; set PD5 for o/p
    OUT PORTD, R17      ; PD5 = 0
    ;-----------------------------------------
    LDI R18, 50         ; set loop counter
l1: RCALL delay_timer0  ; apply delay via timer0
    DEC R18
    BRNE l1             ; & go back & repeat
    ;-----------------------------------------
    EOR R17, R16        ; R17 = R17 XOR R16
    OUT PORTD, R17      ; toggle PD5
    LDI R18, 50         ; re-set loop counter
    RJMP l1             ; go back & repeat toggle
;===============================================================
delay_timer0:           ; ~10ms delay via Timer0
    ;-------------------------------------------------------
```

```
    CLR R20
    OUT TCNT0, R20      ; initialize timer0 with count=0
    LDI R20, 156
    OUT OCR0A, R20      ; OCR0 = 9
    LDI R20, 0b00000010
    OUT TCCR0A, R20
    LDI R20, 0b00000101
    OUT TCCR0B, R20     ; timer0: CTC mode, prescaler 1024
    ;-----------------------------------------------------
l2: IN R20, TIFR0       ; get TIFR0 byte & check
    SBRS R20, OCF0A     ; if OCF0=1, skip next instruction
    RJMP l2             ; else, loop back & check OCF0 flag
    ;-----------------------------------------------------
    CLR R20
    OUT TCCR0B, R20     ; stop timer0
    ;-----------------------------------------------------
    LDI R20, (1<<OCF0A)
    OUT TIFR0, R20      ; clear OCF0 flag
    RET
```

6.2. Code Example 2 (Timer1)

This code toggles PD5 every 0.5s (just like Code Example 1). The `delay_timer1` subroutine uses **Timer1** in **Normal Mode** with a **1024 prescaler** and a **preload value of 57724**, creating a **500ms** hardware delay per call.

```
;------------------------
; Assembly Code
;------------------------
#define __SFR_OFFSET 0x00
#include "avr/io.h"
;------------------------
.global main
;===============================================================
main:
    LDI R16, 0b00100000 ; to toggle PD5
    LDI R17, 0b00000000
    ;-------------------------------------------
    SBI DDRD, 5         ; set PD5 for o/p
    OUT PORTD, R17      ; PD5 = 0
```

```
    ;-------------------------------------------
l1: RCALL delay_timer1  ; 0.5 sec delay via timer1
    ;-------------------------------------------
    EOR R17, R16        ; R17 = R17 XOR R16
    OUT PORTD, R17      ; toggle PD5
    LDI R18, 61         ; re-set loop counter
    RJMP l1             ; go back & repeat toggle
;=============================================================
delay_timer1:           ; 0.5 sec delay via timer1
;-----------------------------------------------------
    .EQU value, 57724   ; value to give 0.5 sec delay
    LDI R20, hi8(value)
    STS TCNT1H, R20
    LDI R20, lo8(value)
    STS TCNT1L, R20     ; initialize counter TCNT1 = value
    ;-----------------------------------------------------
    LDI R20, 0b00000000
    STS TCCR1A, R20
    LDI R20, 0b00000101
    STS TCCR1B, R20     ; normal mode, prescaler = 1024
    ;-----------------------------------------------------
l2: IN R20, TIFR1       ; get TIFR1 byte & check
    SBRS R20, TOV1      ; if TOV1=1, skip next instruction
    RJMP l2             ; else, loop back & check TOV1 flag
    ;-----------------------------------------------------
    LDI R20, 1<<TOV1
    OUT TIFR1, R20      ; clear TOV1 flag
    ;-----------------------------------------------------
    LDI R20, 0b00000000
    STS TCCR1B, R20     ; stop timer1
    RET
```

Revision #1
Created 2026-03-03 15:48:26 UTC by RE
Updated 2026-03-03 15:49:02 UTC by RE