REQUIREMENTS NOT MET

N/A

PROBLEMS ENCOUNTERED

I had a vast amount of problems writing my code because I could not figure out how to get my program to work even though my logic seemed correct. Since I did not initially write to the high bytes of the CCx registers ot clear them, the RGB LED would not turn on at all, and I did not account for this because I thought the register values were configured to 0x00 in the hardware. Additionally, I had some issues with remapping the pins and outputting the PWM values to the LED. It took me many hours of debugging to realize what the issues were.

HOMEWORK EXERCISES

i. How many TC0 channels are necessary to control all three of the LEDs within the on-board RGB LED package?
   a. Three channels because you need one PWM channel per LED (R, G, B), so three TC0 compare channels (CCA/CCB/CCC) are sufficient. A TC0 block provides four compare channels total, so using three covers all LEDs independently.

ii. In the context of the program specified above, would any difference (theoretically) result from setting the RGB period to be $FFFF, instead of $FF?
   a. If the period were set to $FFFF instead of $FF, the theoretical difference is that the PWM would have much higher duty-cycle resolution (16-bit vs. 8-bit) but a frequency 256 times lower. In practice, this slower frequency could cause visible flicker, which is why $FF is preferred for smooth LED control.

PSEUDOCODE/FLOWCHARTS

SECTION 1

hw2.asm:

```
;define constants
;assembler directives

.org 0
Rjmp MAIN

.org 0x100
MAIN:
;init stack

Rcall IO_INIT
Rcall TC_INIT

LOOP:
;check S1 on SLB and read values
;if pressed (low)
    ;write value into TCD0_CCA (red duty)
    ;clear TCD0_CCB and TCD0_CCC (green and blue)
;else continue

;check S1 on MB and read values
;if pressed (low)
    ;write value into TCD0_CCB (green duty)
    ;clear TCD0_CCA and TCD0_CCC (red and blue)
;else continue

;check S2 on SLB and read values
```

```
;if pressed (low)
    ;write value into TCD0_CCC (Blue duty)
    ;clear TCD0_CCA and TCD0_CCB (red and green)
;else continue

;if no switch pressed
    ;Clear all three compare registers to turn RGBs off

Rjmp LOOP



/*****************************************************************************
* Name:     INIT_IO
* Purpose:  Configure RGB LEDs on PD4–PD6 as outputs (inverted, OFF by default),
*           DIP switches as inputs, and tactile switches as inputs with pull-ups.
*
* Inputs:   None
* Outputs:  I/O configured for LEDs, DIP switches, and tactile switches
* Affected: PORTD_DIRSET, PORTD_PINnCTRL, PORTD_OUTSET,
*           PORTA_DIR, PORTF_DIRCLR, PORTE_DIRCLR,
*           PORTF_PINnCTRL, PORTE_PINnCTRL
*****************************************************************************/
INIT_IO:
;push registers

;configure PD4-PD6 as outputs (RGB) (off at first HIGH)

;enable inversion on PD4-PD6 in PORTD_PINnCTRL to ensure duty cycle logic matches brightness

;configure DIP switches as inputs (PORT A)
```

```
;Configure tactile switches as inputs
    ;slb s1 = PF2, slb s2 = PF3
    ;mb s1 = PE0

;pop registers
Ret



/*****************************************************************************
* Name:     TC_INIT
* Purpose:  Configure TCD0 for single-slope PWM on PD4–PD6 (RGB LEDs).
*
* Inputs:   None
* Outputs:  PWM channels initialized on PD4–PD6 (all LEDs OFF by default)
* Affected: PORTD_REMAP, TCD0_PER, TCD0_CCA/CCB/CCC,
*           TCD0_CTRLB, TCD0_CTRLA
*****************************************************************************/
TC_INIT:
;push registers

;enable remap so TC0 outputs go to pd4-pd7 to access the rgb leds
;write to portd_REMAP
;set the bits TC0A, TC0B, TC0C

;configure PWM mode (single slope)
;enable compare channels A, B, C in CTRLB

;set period to 0xFF in TCD0
```

```
;initialize duty cycles to 0 (write 0 to TCD0_CCA, TCD0_CCB, TCD0_CCC)


;Start the timer with PRE = div 8


;pop registers
ret
```

PROGRAM CODE

SECTION 2

```
;
; hw2.asm
;
; Created: 9/29/2025 11:44:13 PM
; Author : arist
;


;define constants
;assembler directives
.include "ATxmega128A1Udef.inc"

.org 0
Rjmp MAIN


.org 0x100
MAIN:
;init stack
ldi r16, low (0x3FFF)
sts CPU_SPL, r16
ldi r16, high (0x3FFF)
sts CPU_SPH, r16

Rcall INIT_IO
Rcall TC_INIT

clr r17

LOOP:
;output dip to leds for debugging
lds r18, PORTA_IN
sts PORTC_OUT, r18
com r18

;check S1 on SLB and read values
lds r16, PORTF_IN
;if pressed (low)
sbrc r16, 2
rjmp NOT_S1_SLB
	;write value into TCD0_CCA (red duty)

	sts TCD0_CCA, r18
	sts TCD0_CCA+1, r17
	;clear TCD0_CCB and TCD0_CCC (green and blue)
	sts TCD0_CCB, r17
	sts TCD0_CCB+1, r17
	sts TCD0_CCC, r17
	sts TCD0_CCC+1, r17

rjmp loop

;else continue

NOT_S1_SLB:
;check S1 on MB and read values
```

```
lds r16, PORTE_IN
;if pressed (low)
sbrc r16, 0
rjmp NOT_S1_MB
	;write value into TCD0_CCB (green duty)
	sts TCD0_CCB, r18
	sts TCD0_CCB+1, r17
	;clear TCD0_CCA and TCD0_CCC (red and blue)
	sts TCD0_CCA, r17
	sts TCD0_CCA+1, r17
	STS TCD0_CCC, r17
	sts TCD0_CCC+1, r17

	rjmp LOOP
;else continue


NOT_S1_MB:
;check S2 on SLB and read values
lds r16, PORTF_IN
;if pressed (low)
sbrc r16, 3
rjmp NO_BUTTONS
	;write value into TCD0_CCC (Blue duty)
	sts TCD0_CCC, r18
	sts TCD0_CCC+1, r17
	;clear TCD0_CCA and TCD0_CCB (red and green)
	sts TCD0_CCA, r17
	sts TCD0_CCA+1, r17
	sts TCD0_CCB, r17
	sts TCD0_CCB+1, r17

	rjmp LOOP
;else continue

NO_BUTTONS:
;if no switch pressed
	;Clear all three compare registers to turn RGBs off
	sts TCD0_CCA, r17
	sts TCD0_CCA + 1, r17
	sts TCD0_CCB, r17
	sts TCD0_CCB + 1, r17
	sts TCD0_CCC, r17
	sts TCD0_CCC + 1, r17

	Rjmp LOOP


/*****************************************************************************
* Name:     INIT_IO
* Purpose:  Configure RGB LEDs on PD4-PD6 as outputs (inverted, OFF by default),
*           DIP switches as inputs, and tactile switches as inputs with pull-ups.
*
* Inputs:   None
* Outputs:  I/O configured for LEDs, DIP switches, and tactile switches
* Affected: PORTD_DIRSET, PORTD_PINnCTRL, PORTD_OUTSET,
*           PORTA_DIR, PORTF_DIRCLR, PORTE_DIRCLR,
*           PORTF_PINnCTRL, PORTE_PINnCTRL
*****************************************************************************/
INIT_IO:
;push registers
push r16
```

```
;enable inversion on PD4-PD6 in PORTD_PINnCTRL to ensure duty cycle logic matches brightness
ldi r16, PORT_INVEN_bm
sts PORTD_PIN4CTRL, r16
sts PORTD_PIN5CTRL, r16
sts PORTD_PIN6CTRL, r16

;configure PD4-PD6 as outputs (RGB) (off at first HIGH)
ldi r16, 0b01110000
sts PORTD_OUTCLR, r16
sts PORTD_DIRSET, r16

;configure DIP switches as inputs (PORT A)
ldi r16, 0xFF
sts PORTA_DIRCLR, r16

;Configure tactile switches as inputs
       ;slb s1 = PF2, slb s2 = PF3
       ldi r16, (1<<2 | 1<<3)
       sts PORTF_DIRCLR, r16
       ; mb s1 = PE0
       ldi r16, (1<<0)
       sts PORTE_DIRCLR, r16

;set slb leds for debuggin purposes
ldi r16, 0xFF
sts PORTC_OUTSET, r16
sts PORTC_DIRSET, r16
;pop registers
pop r16

Ret

/*****************************************************************************
* Name:     TC_INIT
* Purpose:  Configure TCD0 for single-slope PWM on PD4–PD6 (RGB LEDs).
*
* Inputs:   None
* Outputs:  PWM channels initialized on PD4–PD6 (all LEDs OFF by default)
* Affected: PORTD_REMAP, TCD0_PER, TCD0_CCA/CCB/CCC,
*           TCD0_CTRLB, TCD0_CTRLA
*****************************************************************************/
TC_INIT:
;push registers
push r16

;enable remap so TC0 outputs go to pd4-pd7 to access the rgb leds
;write to portd_REMAP
;set the bits TC0A, TC0B, TC0C

;ldi r16, (PORT_TC0C_bm |PORT_TC0B_bm |PORT_TC0A_bm)
ldi r16, 0b00000111
sts PORTD_REMAP, r16


;configure PWM mode (single slope) in CTRLB
; enable compare channels A, B, C in CTRLB
ldi r16, (TC_WGMODE_SINGLESLOPE_gc | TC0_CCAEN_bm | TC0_CCBEN_bm | TC0_CCCEN_bm)
sts TCD0_CTRLB, r16
;set period to 0xFF in TCD0
```

```
ldi r16, low(0xFF)
sts TCD0_PER, r16
ldi r16, high (0xFF)
sts TCD0_PER + 1, r16

;initialize duty cycles to 0 (write 0 to TCD0_CCA, TCD0_CCB, TCD0_CCC)
ldi r16, 0x00
sts TCD0_CCA, r16
sts TCD0_CCA + 1, r16
sts TCD0_CCB, r16
sts TCD0_CCB + 1, r16
sts TCD0_CCC, r16
sts TCD0_CCC + 1, r16

;Start the timer with PRE = div 8
ldi r16, TC_CLKSEL_DIV8_gc
sts TCD0_CTRLA, r16

;pop registers
pop r16

ret
```

APPENDIX

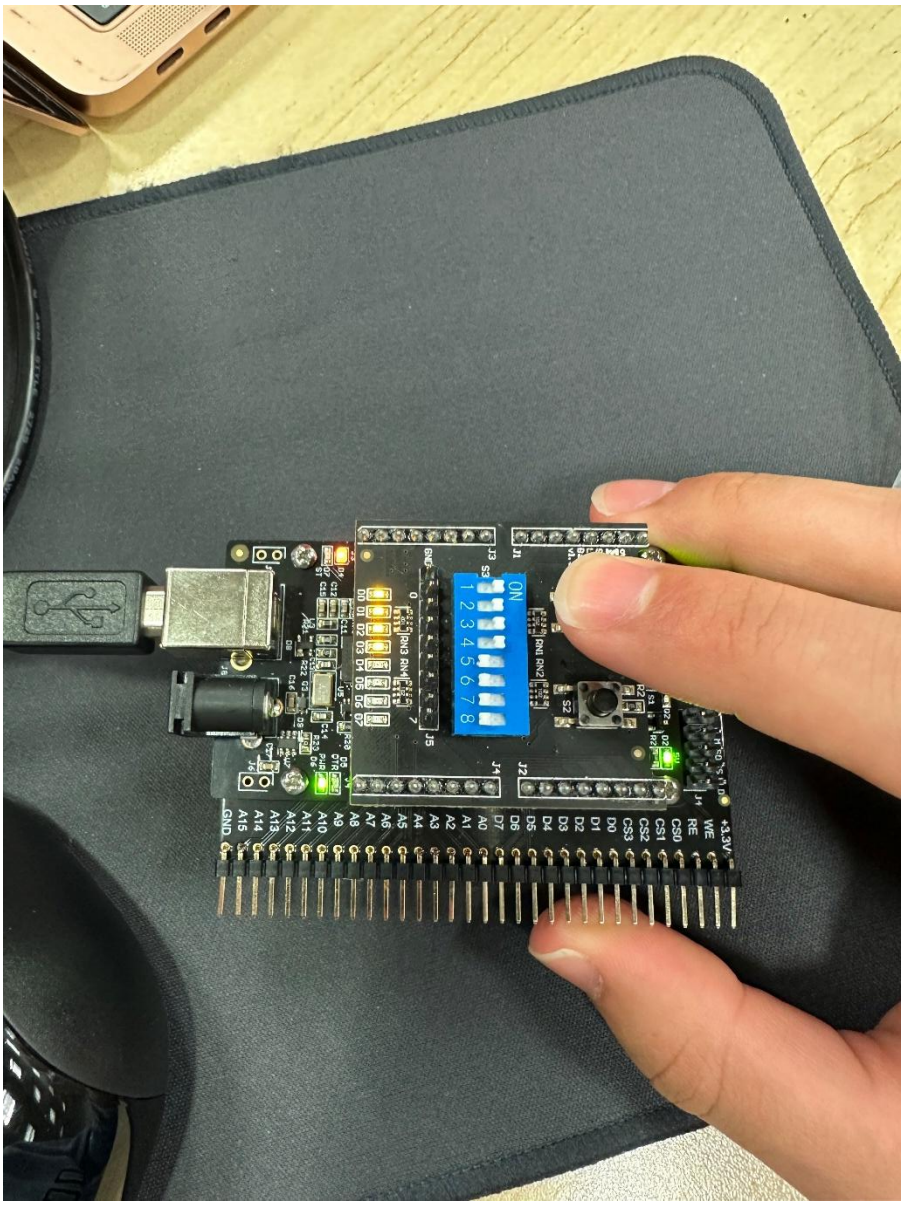

Figure 1: The red LED at about 50% duty cycle with the other LEDs at a very low duty cycle

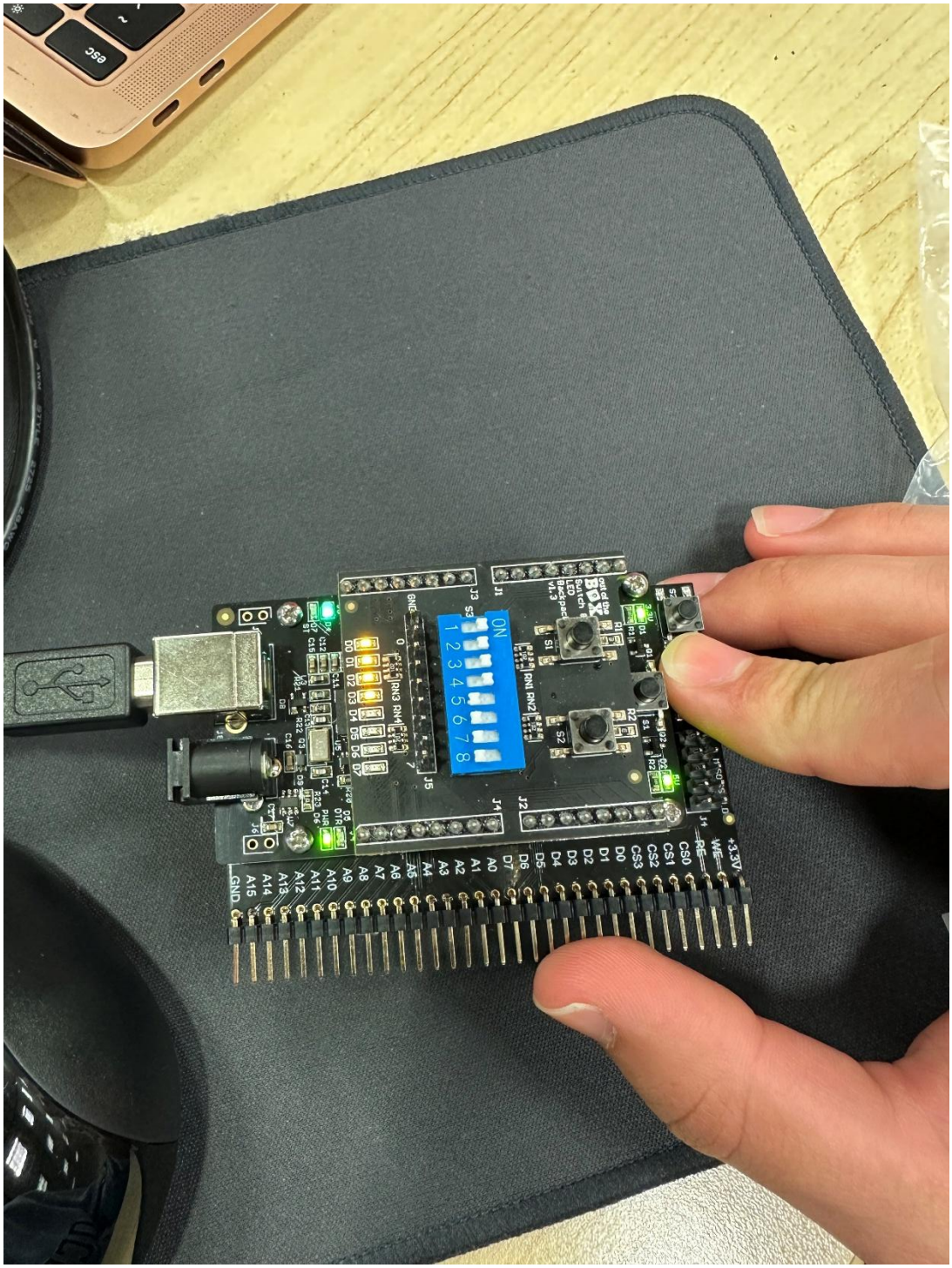

Figure 2: The green LED at about 50% duty cycle with the other LEDs at a very low duty cycle

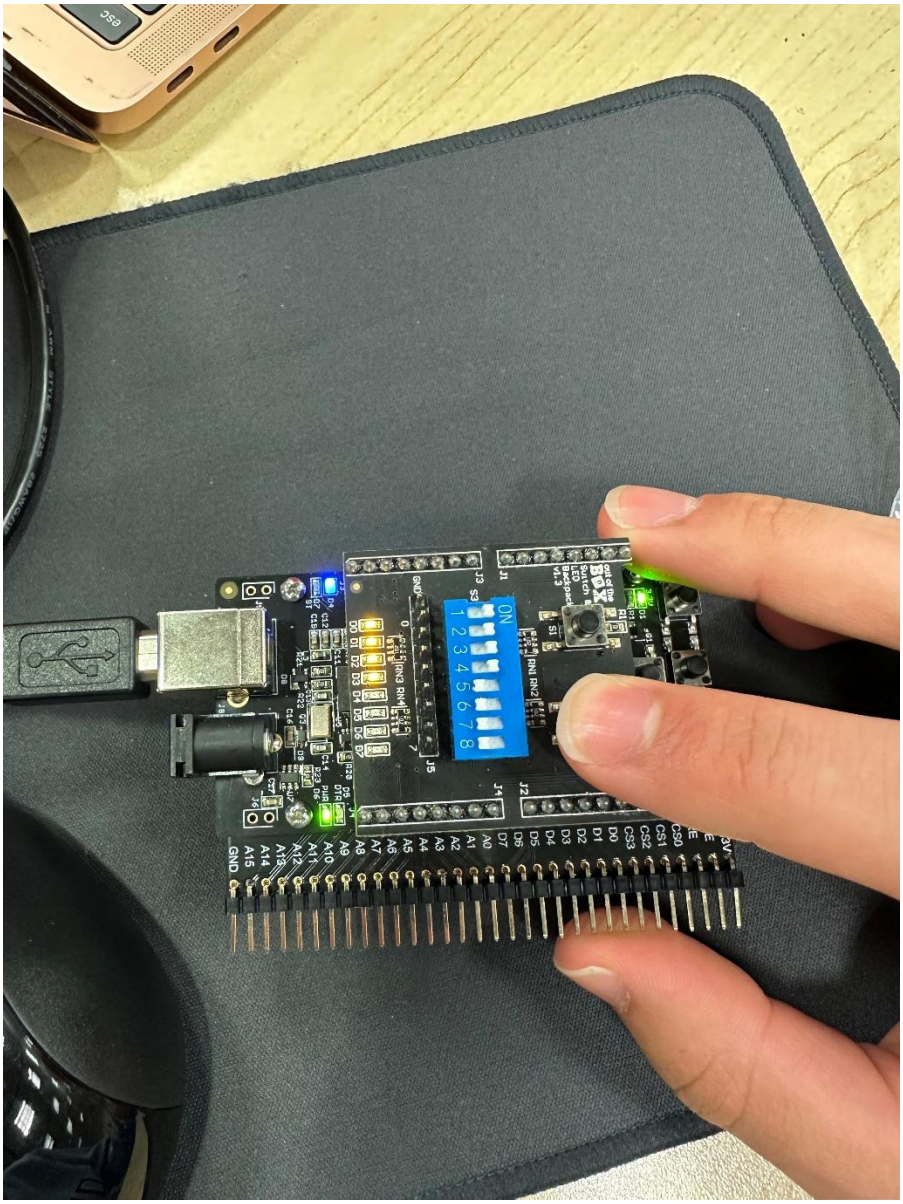

Figure 1: The blue LED at about 50% duty cycle with the other LEDs at a very low duty cycle