Dongyun Lee

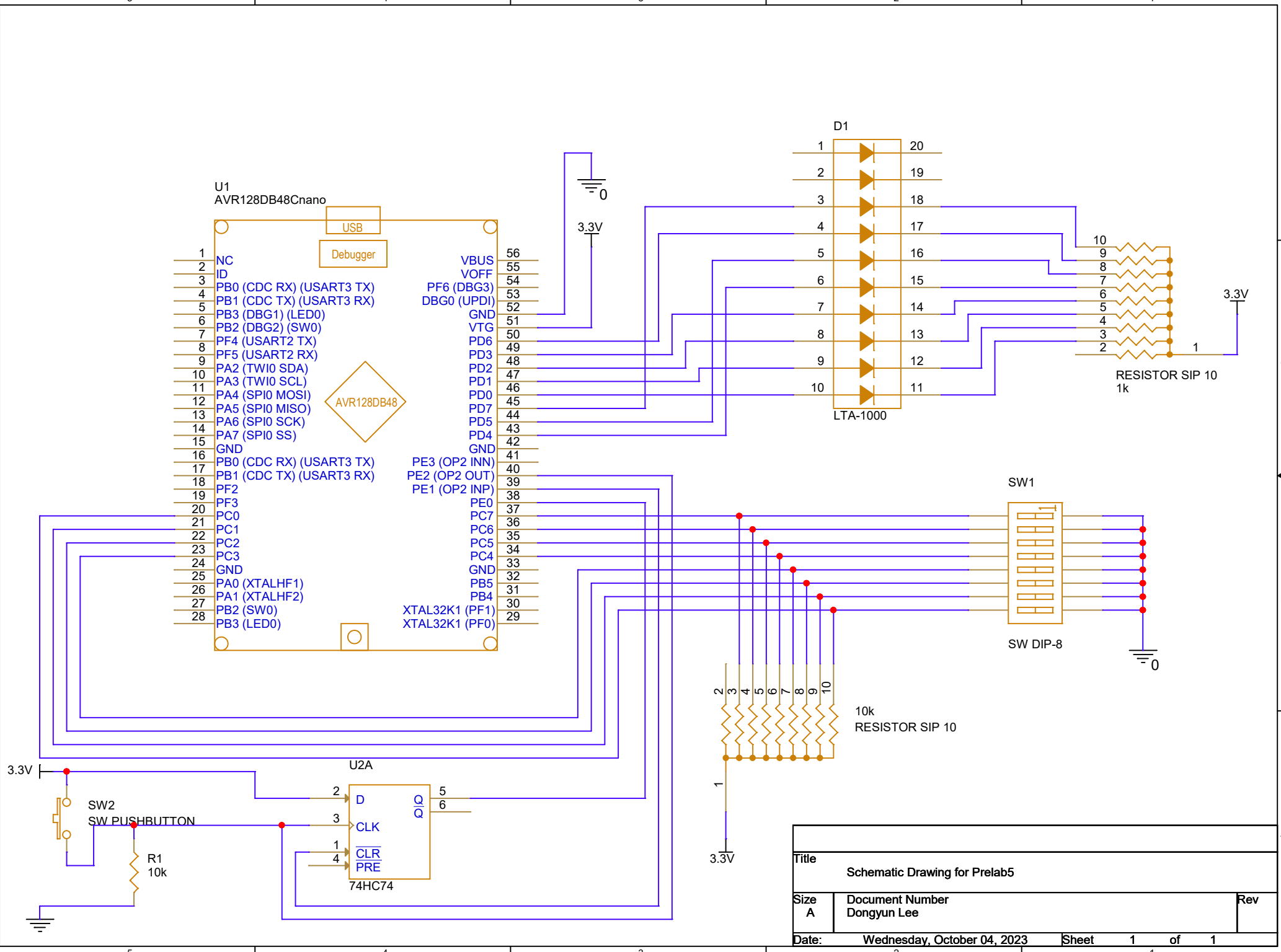
ID: 112794190

PreLab05: Switch Debouncing, Polling, and Conditional I/O

ESE280-L03

Bench #3

```
;
; prelab-task1.asm
;
; Created: 10/2/2023 4:23:59 PM
; Author : CAD
;


; Replace with your application code
start:
    ldi r16, 0xFF       ; set PD output
    out VPORTD_DIR, r16
    ldi r16, 0x00       ;set PE input
    out VPORTE_DIR, r16

wait_for_0:
    sbic VPORTE_IN, 0   ;wait for PE0 being 0
    rjmp wait_for_0
wait_for_1:
    sbis VPORTE_IN, 0   ;wait for PE0 being 1
    rjmp wait_for_1
    inc r16

check_full:
    cpi r16, 0xFF   ; check if r16 is 0xFF which is full
    breq reset      ; if it is true that r16 is equal to 0xFF, go to reset
    rjmp output

reset:
    ldi r16, 0x00
    rjmp wait_for_0

output:
    com r16
    out VPORTD_OUT, r16
    com r16
    rjmp wait_for_0
```


U1
AVR128DB48Cnano
USB
Debugger
AVR128DB48
1 NC
2 ID
3 PB0 (CDC RX) (USART3 TX)
4 PB1 (CDC TX) (USART3 RX)
5 PB3 (DBG1) (LED0)
6 PB2 (DBG2) (SW0)
7 PF4 (USART2 TX)
8 PF5 (USART2 RX)
9 PA2 (TWI0 SDA)
10 PA3 (TWI0 SCL)
11 PA4 (SPI0 MOSI)
12 PA5 (SPI0 MISO)
13 PA6 (SPI0 SCK)
14 PA7 (SPI0 SS)
15 GND
16 PB0 (CDC RX) (USART3 TX)
17 PB1 (CDC TX) (USART3 RX)
18 PF2
19 PF3
20 PC0
21 PC1
22 PC2
23 PC3
24 GND
25 PA0 (XTALHF1)
26 PA1 (XTALHF2)
27 PB2 (SW0)
28 PB3 (LED0)
56 VBUS
55 VOFF
54 PF6 (DBG3)
53 DBG0 (UPDI)
52 GND
51 VTG
50 PD6
49 PD3
48 PD2
47 PD1
46 PD0
45 PD7
44 PD5
43 PD4
42 GND
41 PE3 (OP2 INN)
40 PE2 (OP2 OUT)
39 PE1 (OP2 INP)
38 PE0
37 PC7
36 PC6
35 PC5
34 PC4
33 GND
32 PB5
31 PB4
30 XTAL32K1 (PF1)
29 XTAL32K1 (PF0)
0
3.3V
D1
LTA-1000
RESISTOR SIP 10
1k
SW1
SW DIP-8
10k
RESISTOR SIP 10
U2A
74HC74
D
CLK
CLR
PRE
Q
SW2
SW PUSHBUTTON
R1
10k
Title
Schematic Drawing for Prelab5
Size
A
Document Number
Dongyun Lee
Rev
Date: Wednesday, October 04, 2023
Sheet 1 of 1

```
;
; prelab-task2.asm
;
; Created: 10/2/2023 4:23:59 PM
; Author : CAD
;


; Replace with your application code
start:
    ldi r16, 0xFF       ; set PD output LED bargraph
    out VPORTD_DIR, r16
    ldi r16, 0x00       ;set PE input pushbutton
    out VPORTE_DIR, r16

wait_for_0:
    sbic VPORTE_IN, 0   ;wait for PE0 being 0
    rjmp wait_for_0     ;skips this line if PE0 is 0

wait_for_1:
    sbis VPORTE_IN, 0   ;wait for PE0 being 1
    rjmp wait_for_1     ;skip this line if PE0 is 1
    rjmp delay_make     ;jump to delay when PE0 is 1

delay_make:             ;delay label for make delay
outer_loop_make:
    ldi r17, 133
inner_loop_make:
    dec r17
    brne inner_loop_make
    dec r16
    brne outer_loop_make
    rjmp still_1        ; jump to still_1

wait_for_0_delay_after: ;comes here after output
    sbic VPORTE_IN, 0
    rjmp wait_for_0_delay_after ;skips this line if PE0 is 0
    rjmp delay_break

delay_break:            ;delay lable for break delay
outer_loop_break:
    ldi r17, 133
inner_loop_break:
    dec r17
    brne inner_loop_break
    dec r16
    brne outer_loop_break
    rjmp still_0

still_1:
    sbis VPORTE_IN, 0   ;check if PE0 is still 1
    rjmp wait_for_0     ; if PE0 is 0 then go jump to wait_for_0
```

```
    rjmp output         ; outputs the value

still_0:
    sbic VPORTE_IN, 0   ;check if PE0 is still 0
    rjmp wait_for_0_delay_after
    rjmp wait_for_0     ;go back to start

check_full:
    cpi r16, 0xFF   ; check if r16 is 0xFF which is full
    breq reset      ; if it is true that r16 is equal to 0xFF, go to reset
    rjmp output

reset:
    ldi r16, 0x00
    rjmp wait_for_0

output:
    rcall check_full
    inc r16
    com r16
    out VPORTD_OUT, r16
    com r16
    rjmp wait_for_0_delay_after ; jump to wait for 0 but that has delay after
```

```
;
; conditional_input.asm
;
; Created: 10/2/2023 6:53:51 PM
; Author : CAD
;


; Replace with your application code
start:
    ldi r16, 0xFF   //make into output
    out VPORTD_DIR, r16
    ldi r16, 0x00   //make into input
    out VPORTC_DIR, r16

    cbi VPORTE_DIR, 0   ; makes PE0(pushbutton) to input
    sbi VPORTE_DIR, 1   ; makes PE1(clear) to output
    cbi VPORTE_DIR, 2   // input directly from pushbutton

wait_for_0:
    sbic VPORTE_IN, 0   ;wait for PE0 being 0
    rjmp wait_for_0     ;skips this line if PE0 is 0

wait_for_1:
    sbis VPORTE_IN, 0   ;wait for PE0 being 1
    rjmp wait_for_1     ;skip this line if PE0 is 1
    rjmp delay_make     ;jump to delay when PE0 is 1

delay_make:             ;delay label for make delay
outer_loop_make:
    ldi r17, 133
inner_loop_make:
    dec r17
    brne inner_loop_make
    dec r16
    brne outer_loop_make
    rjmp still_1        ; jump to still_1

wait_for_0_delay_after: ;comes here after output
    rcall check_pushbutton_released     // check if the pushbutton has completely
      released
    rcall clear_flip_flop
    sbic VPORTE_IN, 0
    rjmp wait_for_0_delay_after ;skips this line if PE0 is 0
    rjmp delay_break

delay_break:            ;delay lable for break delay
outer_loop_break:
    ldi r17, 133
inner_loop_break:
    dec r17
    brne inner_loop_break
```

```
    dec r16
    brne outer_loop_break
    rjmp still_0

still_1:
    sbis VPORTE_IN, 0   ;check if PE0 is still 1
    rjmp wait_for_0     ; if PE0 is 0 then go jump to wait_for_0
    rjmp output         ; outputs the value

still_0:
    sbic VPORTE_IN, 0   ;check if PE0 is still 0
    rjmp wait_for_0_delay_after
    rjmp wait_for_0     ;go back to start

output:
    //code for output the values from PC ports to PD ports
    inc r16
    com r16
    out VPORTD_OUT, r16
    com r16
    rjmp wait_for_0_delay_after ; jump to wait for 0 but that has delay after

clear_flip_flop:
    cbi VPORTE_DIR, 1
    sbi VPORTE_DIR, 1
    rjmp wait_for_0

display:
    in r16, VPORTC_DIR
    com r16
    out r16, VPORTD_DIR
    rjmp clear_flip_flop

check_pushbutton_released:
    rcall delay_break_pushbutton
    sbic VPORTE_IN, 2   //check if PE) which is directly connected to pushbutton is
      still 0
    rjmp check_pushbutton_released

delay_break_pushbutton:         ;delay lable for break delay for pushbutton directly
outer_loop_break:
    ldi r17, 133
inner_loop_break:
    dec r17
    brne inner_loop_break
    dec r16
    brne outer_loop_break
```