

Design Assignment 2A

Student Name: Itzel Becerril

Student #: 2000478001

Student Email: becerril2@unlv.nevada.edu

Primary Github address: https://github.com/HadidBuilds/hw_sub_da1

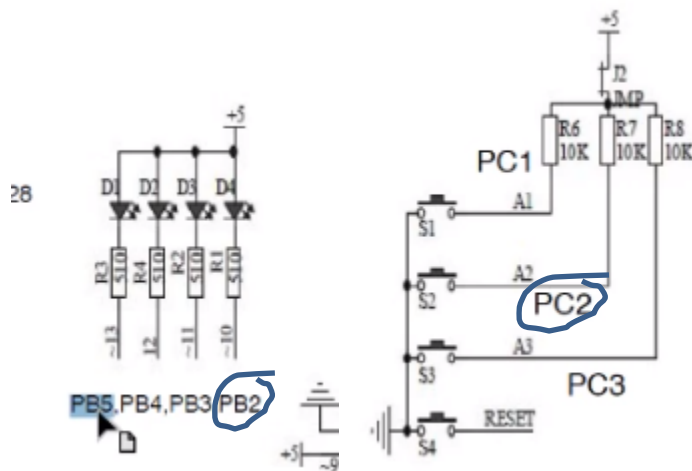
Directory: DA2

1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

List of Components used:

- Atmel Studio 7
- Atmega328P
- Xplained mini
- microUsb Cord
- Multifunctional shield
- Switch and LED on board

Block diagram with pins used in the Atmega328P



2. INITIAL/MODIFIED/DEVELOPED CODE OF TASK 1/A

Task One:

```
;
; DA2A_taskoneAssem.asm
;
; Created: 3/4/2019 8:30:48 PM
; Author : becerril2
;

; Replace with your application code
.org 0x0000
```

```

;Initialize the microcontroller stack pointer

ldi        r16, 0x24    ; sets p2 and p5
out        DDRB, r16    ; enableing pd5 and p2 as outputs
ldi        r16, 0x04    ; use for toggle later
ldi        r17, 0x20    ; used for toggle later
ldi        r20, 0x05    ; Prescalar to 1024

sts        TCCR1B,      r20 ; sets prescalar

begin:
ldi        r20, 0x00    ; setting up timer counter to 0
sts        TCNT1H, r20 ; Timer counter is 16 bits
sts        TCNT1L, r20 ; Thus, requires 2 8 bit regs

rjmp      delay_on      ; routine call for delay see below

returnOne:
eor        r17, r16     ; xor to toggle LED
out        PORTB, r17   ; enable pd2

rjmp      delay_off     ; routine call for delay see below

returnTwo:
eor        r17, r16     ; xor to toggle LED
out        PORTB, r17   ; enable pd2

rjmp      begin         ; repeat main loop

delay_off:
lds        r29, TCNT1H  ; load upper bytes of timer counter to r29
lds        r28, TCNT1L ; load lower bytes of timer counter to r28

cpi        r28, 0x12    ; check to see if lower 8 bits of timer counter are
0x12
brsh      body2         ;branch if same or equal to then body2
rjmp      delay_off     ; otherwise keep checking lower bytes

body2:
cpi        r29, 0x4F    ; check if upper timer counter have reached 0x4F
brlt      delay_off     ; otherwise recheck the lower bytes
rjmp      returnTwo

delay_on:
lds        r29, TCNT1H  ; load upper bytes of timer counter to r29
lds        r28, TCNT1L ; load lower bytes of timer counter to r28

cpi        r28, 0xDA    ; check to see if lower 8 bits of timer counter are
0xDA
brsh      body1         ;branch if same or equal to then body1
rjmp      delay_on      ; otherwise keep checking lower bytes

body1:
cpi        r29, 0x1A    ; check if upper timer counter is 0x1A

```

```

    brlt    delay_on      ; otherwise recheck the lower bytes
    rjmp    returnOne

```

Task Two:

```

;
; DA2A_tasktwoAssem.asm
;
; Created: 3/4/2019 8:51:28 PM
; Author : becerri2
;

```

```

; Replace with your application code
.org 0

```

```

ldi r16,0      ;r16 = 0
out ddrc, r16  ;enable as output
ldi r16, 0xff  ;r16 = 0xff
out ddrb, r16  ;enable as output
out portb, r16 ;enable as output

```

Start:

```

    SBIC    PINC, 2      ; PINC2 == high, skip
    RJMP    Start ; jumps to start until button is pressed
    RCALL   LED ; Call subroutine LED

```

Done:

```

    RJMP    Start; will restart

```

LED:

```

    CBI      PORTB, 2      ; PB2 = 0
    LDI      R20, 5 ; prescalar is set to 1024
    STS      TCCR1B, R20 ; set prescalar
    LDI      R20, 0 ; timer is set to 0
    STS      TCNT1H, R20 ; clear high 8 bits
    STS      TCNT1L, R20 ; clear lower 8 bits
    RJMP     Delayforlowbyte ; jump to delay routine

```

Delayforlowbyte:

```

    LDS      R29, TCNT1H      ; load upper bytes of timer counter to
r29
    LDS      R28, TCNT1L      ; load lower bytes of timer counter to r28
    CPI      R28, 0x4A        ; check to see if lower 8 bits of timer
counter are 0x4A
    BRSH     Delayforhighbyte
    RJMP     Delayforlowbyte  ; otherwise keep checking

```

Delayforhighbyte:

```

    CPI      R29, 0x4C        ; check if upper timer counter have
reached value
    BRLT     Delayforlowbyte  ; otherwise recheck the lower bytes
    SBI      PORTB, 2        ; will turn off LED
    RET

```

3. DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A

Task One in C:

```

/*
 * DA2A_taskoneC.c

```

```

*
* Created: 3/4/2019 8:27:36 PM
* Author : becerri2
*/

#include <avr/io.h>

int main(void)
{
    TCCR1B = 5; //set prescalar as 1024
    //set port b
    DDRB |= (1<<2);
    PORTB |= (1<<2);
    DDRB |= (1<<5);
    PORTB |= (1<<5);
    while(1)
    {
        //reset timer1
        TCNT1 = 0;
        while(TCNT1 != 4687)
        {
            //DELAY
        }

        //Toggle pb.2
        PORTB ^= (1<<2);

        //reset timer2
        TCNT1 = 0;
        while(TCNT1 != 6874)
        {
            //Delay
        }
        //toggle pb.2
        PORTB ^= (1<<2);
    }
    return 0;
}

```

Task Two in C:

```

/*
* DA2A_tasktwoC.c
*
* Created: 3/4/2019 8:29:03 PM
* Author : becerri2
*/

#define F_CPU 16000000UL //Setting up to 16Mhz

#include <avr/io.h>
#include <util/delay.h>

int main (void)
{
    //set portb for output
    DDRB |= (1<<2);

```

```

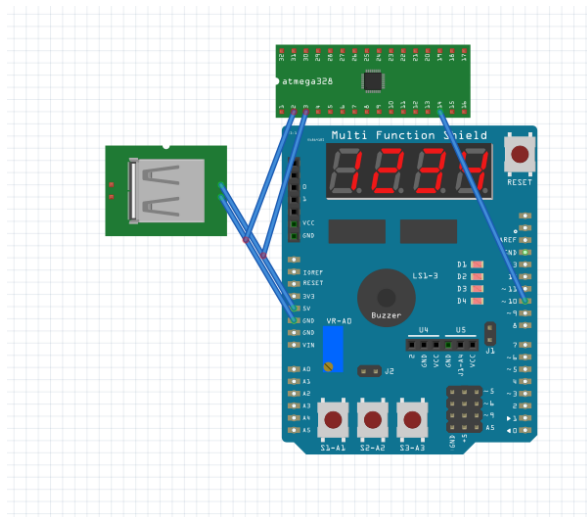
DDRB |= (1<<5);
PORTB |= (1<<5); //turn led off
PORTB |= (1<<2); //turn led off

//set switch portc for input
DDRC &= (0<<2);
PORTC |= (0<<2);

while(1)
{
    if(!(PINC & (1<<PINC2)))
    {
        PORTB &= ~(1<<2); //turn pb.2 on
        _delay_ms(1250); //set delay of 1.250
    }
    else
    {
        PORTB |= (1<<2); //turn pb.2 off
    }
}
return 0;
}

```

4. SCHEMATICS



5. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)

Task 1:

C code:

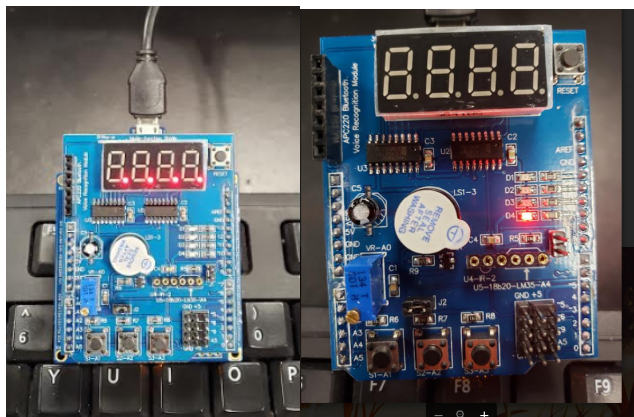
Processor Status					
Name	Value		Name	Value	
Program Counter	0x00000052		Program Counter	0x00000059	
Stack Pointer	0x08FD		Stack Pointer	0x08FD	
X Register	0x0000		X Register	0x0000	
Y Register	0x08FF		Y Register	0x08FF	
Z Register	0x0084		Z Register	0x0084	
Status Register	I T H S V N Z C		Status Register	I T H S V N Z C	
Cycle Counter	4799516		Cycle Counter	11838487	
Frequency	16.000 MHz		Frequency	16.000 MHz	
Stop Watch	299,969.75 μ s		Stop Watch	739,905.44 μ s	
Registers					

Assembly code:

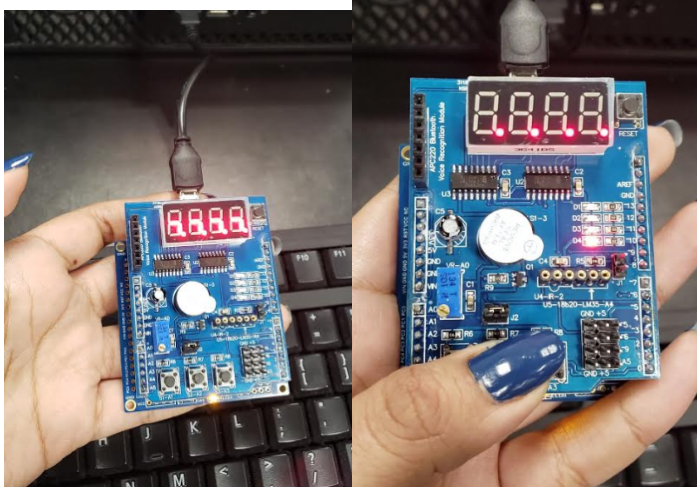
Stack Pointer	0x08FF
X Register	0x0000
Y Register	0x1ADA
Z Register	0x0000
Status Register	I T H S V N Z C
Cycle Counter	7038991
Frequency	16.000 MHz
Stop Watch	439,936.94 μ s
Registers	

6. SCREENSHOT OF EACH DEMO (BOARD SETUP)

Task one:



Task two:



7. VIDEO LINKS OF EACH DEMO

Assembly Code Task One

https://www.youtube.com/watch?v=_zfD2UYDyS0

Assembly Code Task Two

<https://www.youtube.com/watch?v=abMpAnRxQPk>

C Code Task One

<https://www.youtube.com/watch?v=MUqPTjGyb14>

C Code Task Two

https://www.youtube.com/watch?v=YdtJcg_O_zA

8. GITHUB LINK OF THIS DA

https://github.com/HadidBuilds/hw_sub_da1

Student Academic Misconduct Policy

<http://studentconduct.unlv.edu/misconduct/policy.html>

"This assignment submission is my own, original work".

Itzel Becerril