CPE301 – SPRING 2019

Design Assignment 2A

Student Name: Geovanni Portillo
Student #: 8000603824
Student Email: portig1@unlv.nevada.edu
Primary Github address: https://github.com/portig1/submissions_E
Directory: submissions_E/DA/LAB2A/

Submit the following for all Labs:

1. In the document, for each task submit the modified or included code (only) with highlights and justifications of the modifications. Also, include the comments.
2. Use the previously create a Github repository with a random name (no CPE/301, Lastname, Firstname). Place all labs under the root folder ESD301/DA, sub-folder named LABXX, with one document and one video link file for each lab, place modified asm/c files named as LabXX-TYY.asm/c.
3. If multiple asm/c files or other libraries are used, create a folder LabXX-TYY and place these files inside the folder.
4. The folder should have a) Word document (see template), b) source code file(s) and other include files, c) text file with youtube video links (see template).

1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

Atmel Studio 7
ATmega328PB Xplained mini
Multi-function Shield

Figure 1-1. ATmega328P Xplained Mini Headers and Connectors

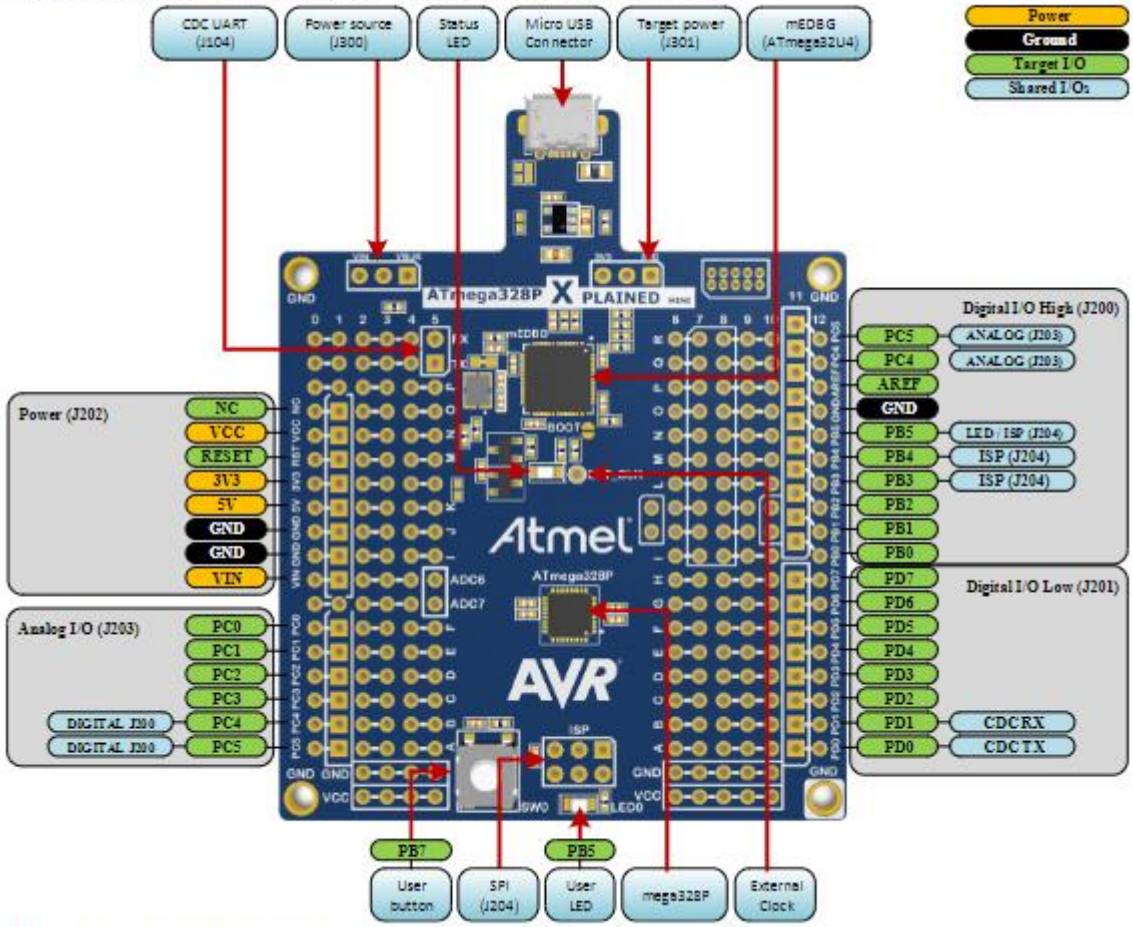
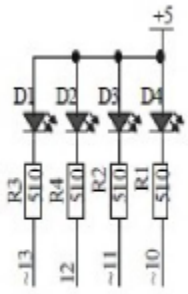
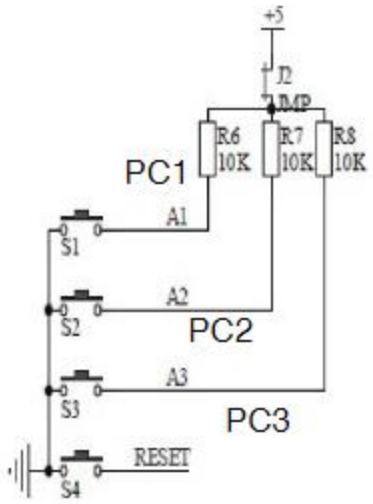


Table 1-1. Default Configurations



PB5,PB4,PB3,PB2

Multifunction Shield LED schematic



Multifunction Shield Switch Schematic

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

```
.org 0
	;Set PORTB.2 For Output
	SBI DDRB,2
	LDI R16, 0
	OUT PORTB, R16

	; Toggle PORTB.2 on and off. On for 435ms and off for 290ms to achieve a period of
725ms with a duty cycle of 60%
	; The LED connected to PORTB is active low so by setting PORTB.2 to 0, it is
actually being turned on
L1:	LDI R16, 0
	OUT PORTB, R16
	RCALL delay435ms
	LDI R16, 4
	OUT PORTB, R16
	RCALL delay290ms
	RJMP L1

delay435ms:
			;Delay function is meant to have a delay of approximately 435ms for
a clock at 16MHz, actual time is about 437ms
			;Delay function from delayL0 and below takes 120,603 clock cycles
			;So using the clock's ratio of 16 x 10^3 calculations per second and
knowing that the period we need is 0.435s
			;I found the clock cycles needed to be 6,960,000. 57.7 Iterations of
the delay loop is needed so I rounded up to 58
	LDI R19, 58
delayL0_A: LDI R20, 200
delayL1_A: LDI R21, 200
delayL2_A: DEC R21
	BRNE delayL2_A
	DEC R20
	BRNE delayL1_A
	DEC R19
	BRNE delayL0_A
```

```
	RET

delay290ms:
                    ;Delay function is meant to have a delay of approximately 290ms for
a clock at 16MHz but is closer to about 294ms
                    ;Process is identical to that of the function above. Clock cycles
needed are 4,640,000 and require 38.5 iterations rounded up
	LDI R19, 39
delayL0_B: LDI R20, 200
delayL1_B: LDI R21, 200
delayL2_B: DEC R21
	BRNE delayL2_B
	DEC R20
	BRNE delayL1_B
	DEC R19
	BRNE delayL0_B
	RET
```

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

```
.org 0
	;Set PORTB.2 For Output
	SBI DDRB,2
	LDI R16, 4
	OUT PORTB, R16

	;Set PORTC.2 For Input
	CBI DDRC, 2
	LDI R16, 4
	OUT PORTC, R16

	;If the switch connected to PINC.2 is pushed (Set to Low), PORTB.2 will be set to
0 and turn on the LED for 1.25s

L1:
	SBIC PINC, 2  ;If PINC isn't cleared (Switch is pushed), RJMP LEDoff is exectued
and the LED will be turned off, if not already
	RJMP LEDoff
	LDI R16, 0
	OUT PORTB, R16
	RCALL delay1250ms
	RJMP L1
	LEDoff: LDI R16, 4
	OUT PORTB, R16
	RJMP L1

delay1250ms:
                                ;Delay function is meant to have a delay of
approximately 435ms for a clock at 16MHz, actual time is about 437ms
                                ;Identical to delay function used in Task 1, clock
cycles needed are 20,000,000 and require 165.8 iterations rounded up
	LDI R19, 166
delayL0_A: LDI R20, 200
delayL1_A: LDI R21, 200
delayL2_A: DEC R21
	BRNE delayL2_A
	DEC R20
```

```
        BRNE delayL1_A
        DEC R19
        BRNE delayL0_A
        RET
```

4. INITIAL/MODIFIED/DEVELOPED CODE OF TASK 1/C Code

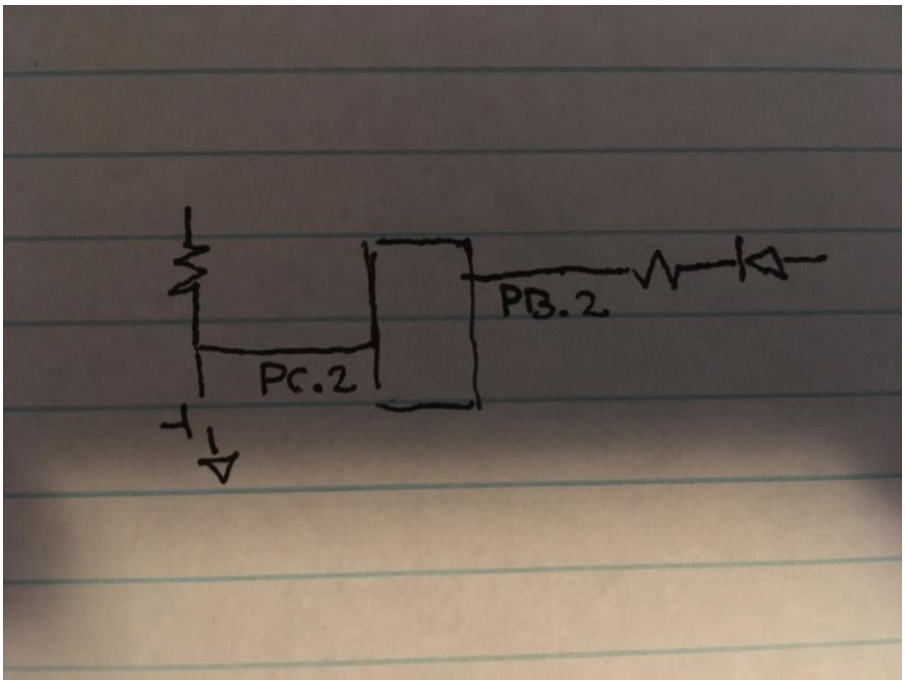
```
#include <avr/io.h>
/*      - LED connected to PORTB.2 */

int main (void)
{
        /* set PORTB.2 for output*/
        DDRB |= (1 << 2);
        PORTB |= (1 << 2);
        long i;
        /* Toggle PORTB.2 on and off. On for 435ms and off for 290ms to achieve a period
of 725ms with a duty cycle of 60%
                The LED connected to PORTB is active low so by setting PORTB.2 to 0, it is
actually being turned on */
        /* For a counter set to 16,000,000, it takes 96000004 clock cycles to complete a
countdown and it takes 6s to complete
                using this ratio, I calculated to get a time of 0.435s the counter needs to
be set around 1,160,000 and for 0.29s a counter of 773,333 */
        while (1)     {
                i = 1160000;
                PORTB = _BV(PINB2);
                while (i > 0) {
                        i--;
                };
                i = 773333;
                PORTB = ~(_BV(PINB2));
                while (i > 0) {
                        i--;
                };
        }
        return 0;
}
```

5. DEVELOPED MODIFIED CODE OF TASK 2/C Code from TASK 1/C Code

```
#include <avr/io.h>
/*      - LED connected to PORTB.2 */
/*      - Switch connected to PORTC.2 */

int main (void)
{
        long i; //initialize i for counter
        /* set PORTB.2 for output*/
        DDRB |= (1 << 2);
        PORTB |= (1 << 2);

        /* set PORTC.2 for input*/
        DDRC &= (0 << 2);
        PORTC |= (1 << 2);   //enable pull-up
```

```
	/* A switch is connected to PORTC.2 and when pressed PINC.2 is set low. Condition
for the if statment is then true
	and the LED will turn on for 1.25s. Using the ratio found in Task1_ I calculated
the counter needs to be set to
	333,333 to have a delay of approximately 1.25s*/

	while (1) {
		if(!(PINC & (1 << PINC2)))
		{
			PORTB &= ~(1 << 2);
			i = 3333333;
			while (i > 0) {
				i--;
			};
		}
		else
			PORTB |= (1 << 2);

	}
	return 0;
}
```

6. SCHEMATICS

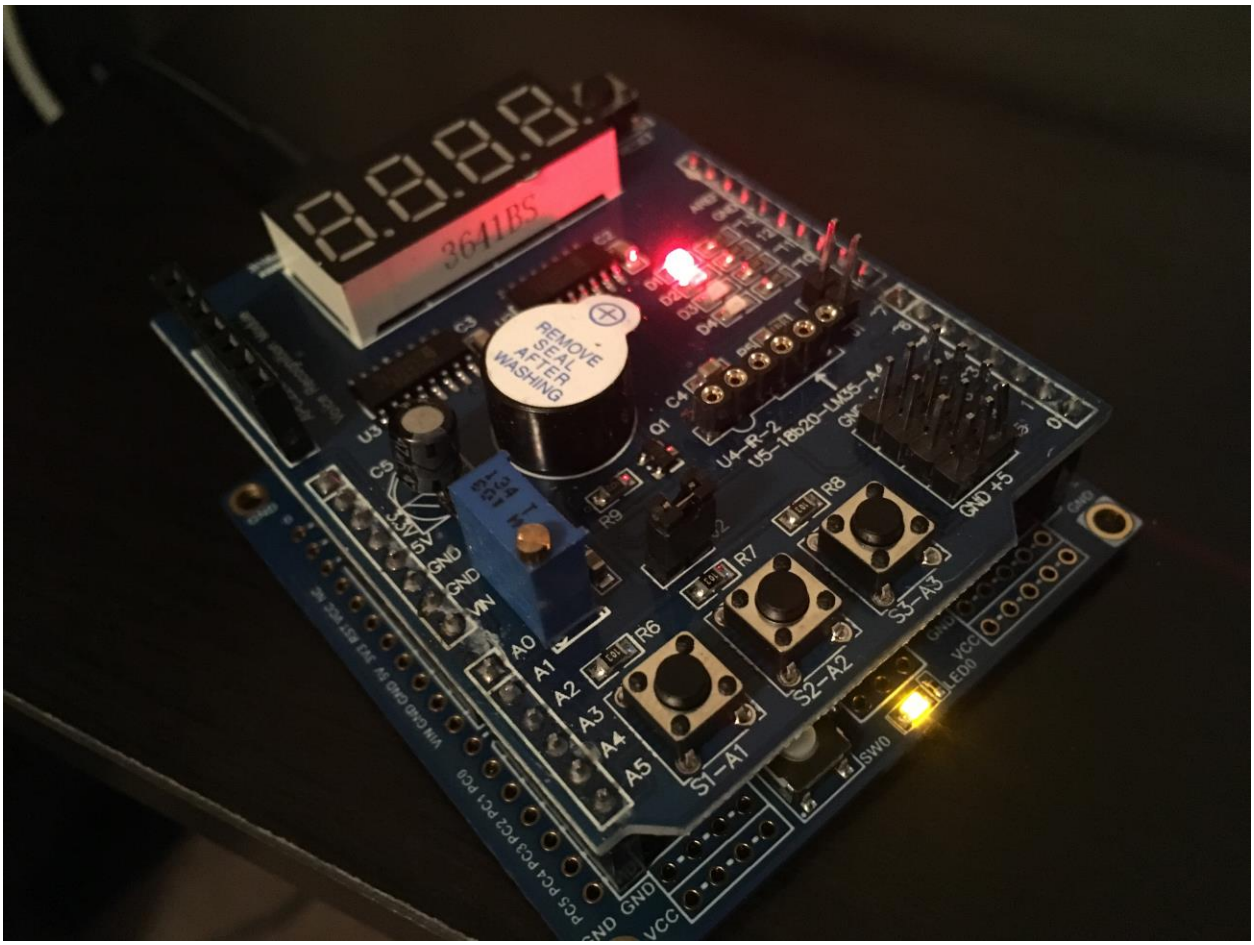


Schematic showing connections for PORTB.2 and PORTC.2

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

Not asked for in Design Assignment 2A instructions

8. SCREENSHOT OF EACH DEMO (BOARD SETUP)



Board setup for Tasks 1 and 2 (Assembly and C)

9. VIDEO LINKS OF EACH DEMO

https://youtu.be/Jv3t5SKwIiE

10. GITHUB LINK OF THIS DA

https://github.com/portig1/submissions_E/tree/master/DA/LAB2A

Student Academic Misconduct Policy

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

"*This assignment submission is my own, original work*".
Geovanni Portillo