

Design Assignment 2B

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/LAB2B/

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

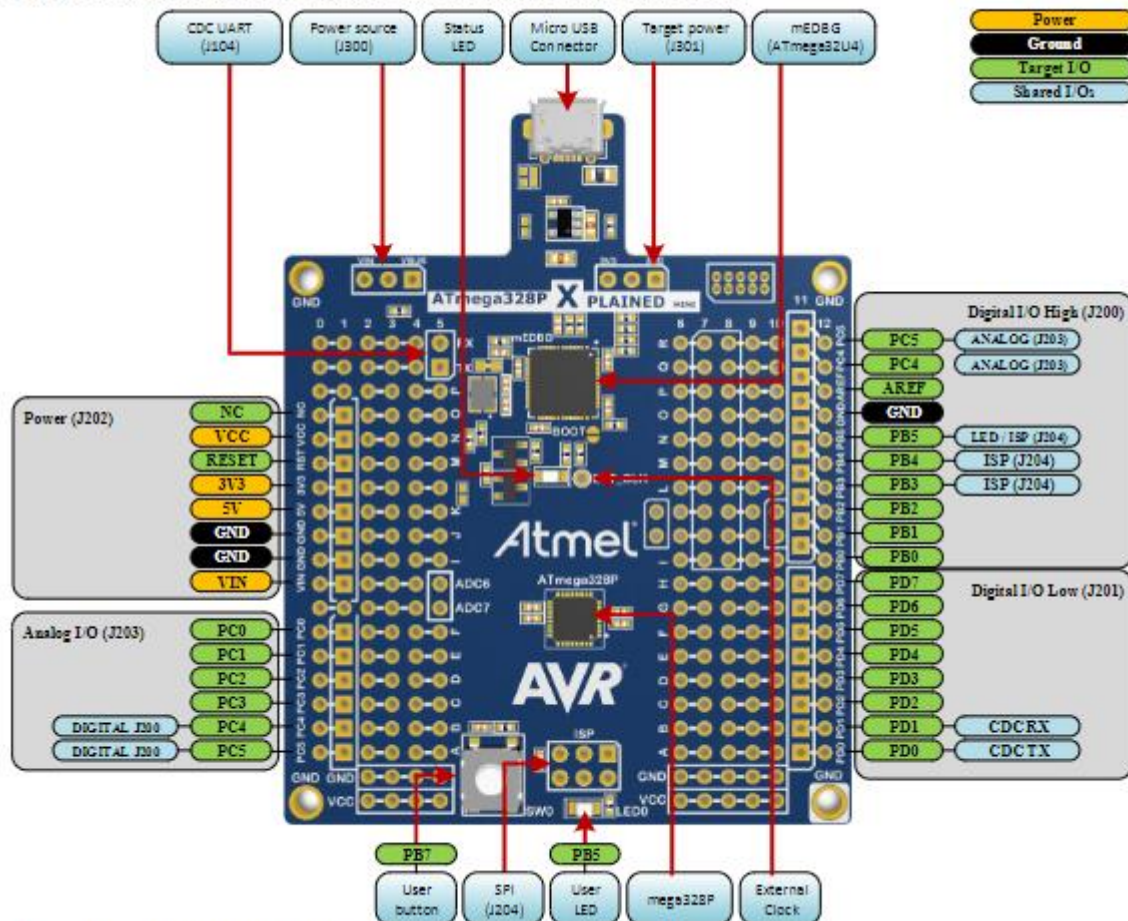
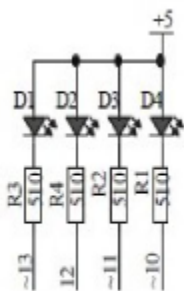


Table 1-1. Default Configurations

Tied a wire to PD2 to be able to press to GND to active INT0



PB5,PB4,PB3,PB2

Multifunction Shield LED schematic

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

```
.ORG 0 ;location for reset
    JMP MAIN
.ORG 0x02 ;location for external interrupt 0
    JMP EX0_ISR
MAIN:
    LDI R20, HIGH(RAMEND)
    OUT SPH, R20
    LDI R20, LOW(RAMEND)
    OUT SPL, R20 ;initialize stack

    SBI DDRB, 2 ;PORTB.2 = output
    SBI PORTB, 2 ;Turn off LED (reverse logic)
    SBI PORTD, 2 ;pull-up activated

    LDI R20, 0x2 ;make INT0 falling edge triggered
    STS EICRA, R20

    LDI R20, 1<<INT0 ;enable INT0
    OUT EIMSK, R20
    SEI ;enable interrupts
```

HERE:JMP HERE

```
EX0_ISR:
    CBI PORTB, 2
    RCALL delay1250ms
    SBI PORTB, 2
    RETI
```

delay1250ms:

```
                                ;Taken from DA2A, Task 2
                                ;Delay function is meant to have a delay of
                                ;approximately 1250ms for a clock at 16MHz
                                ;Delay function from delayL0 and below takes 120,603
                                ;clock cycles
                                ;Clock cycles total needed are 20,000,000 and require
                                ;165.8 iterations rounded up
                                ;However, when running on hardware the time it takes
                                ;for the delay is about twice as long when the delay is in the
                                ;interrupt. To mitigate this I have halved the 166 to
                                ;83 and that gives about 1.250s
```

```
    LDI R19, 83
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
```

3. RET INITIAL/MODIFIED/DEVELOPED CODE OF TASK 1/C Code

```
int main(void)
{
```

```

    DDRB |= (1 << 2); //PB2 as output
    PORTB |= (1 << 2); //Turn off LED (reverse logic)
    PORTD |= (1 << 2); //Pull-up activated
    EICRA = 0x02; //Make INT0 falling-edge triggered

    EIMSK = (1 << INT0); //enable external interrupt 0
    sei(); //enable interrupts

    while(1) {
    }
}

ISR (INT0_vect) { //ISR for external interrupt 0

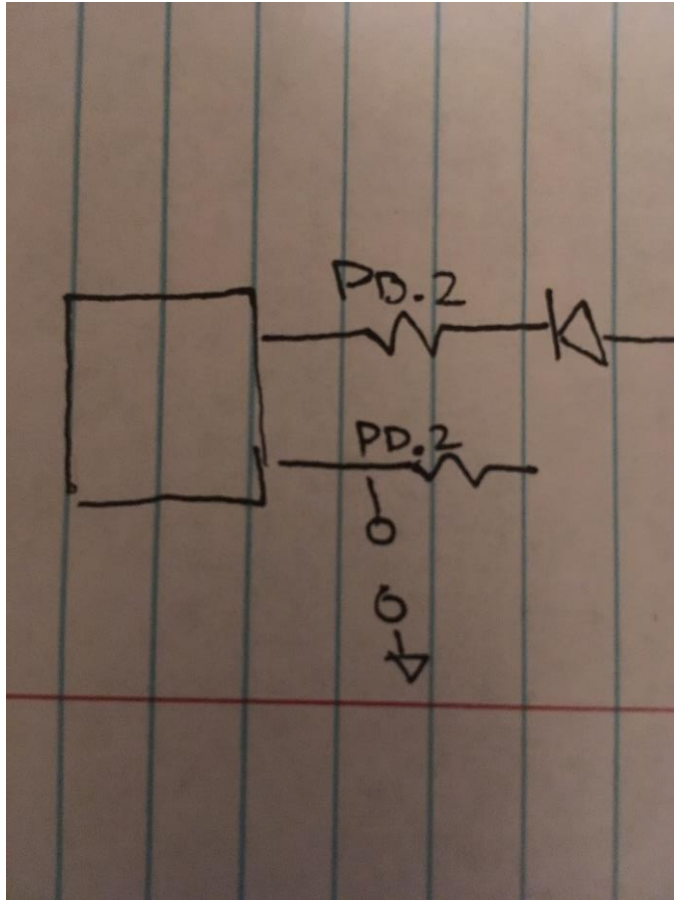
    PORTB = (0 << 2); //Turn on

    /* For a counter set to 16,000,000 it takes 6s to complete and using that ratio
    and knowing I need a delay of 1.25s, I calculated
    that the counter needs to be set at 3,333,333. However, when running on hardware
    with this loop in the interrupt, the delay takes about
    twice as long to complete. So for this I halved 3,333,333 to 1,666,667 to get the
    1.250s delay*/

    long i; //initialize i for counter
    i = 1666667;
    while (i > 0) {
        i--;
    };
    PORTB = (1 << 2); //Turn off
}

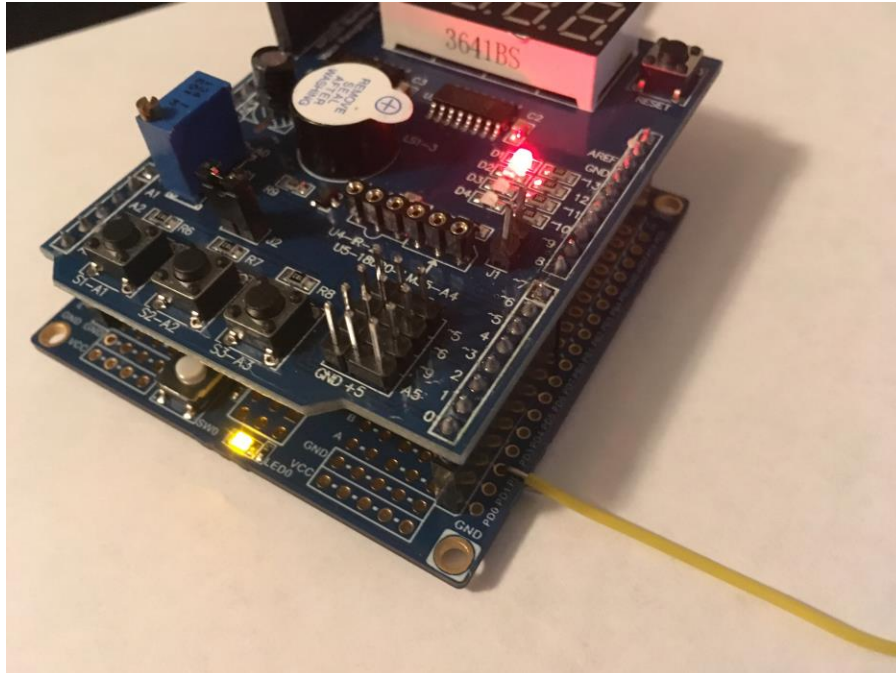
```

4. SCHEMATICS



Schematic showing connections for PORTB.2 and PORTD.2 (INT0) which needed to be grounded to activate

5. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**
Not asked for in Design Assignment 2A instructions
6. **SCREENSHOT OF EACH DEMO (BOARD SETUP)**



Board setup for Task 1 (Assembly and C), wire tied to PD.2 to connect to ground for external interrupt 0 (INT0)

7. VIDEO LINKS OF EACH DEMO

<https://youtu.be/rjn2Ocekjtw>

8. GITHUB LINK OF THIS DA

https://github.com/portig1/submissions_E/tree/master/DA/LAB2B

Student Academic Misconduct Policy

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

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

Geovanni Portillo