CPE301 – SPRING 2022

Design Assignment 2

Student Name: Angelo Nolasco
Student #: 5005497011
Student Email: Nolasco@unlv.nevada.edu
Primary Github address: angelon3121@gmail.com
Directory:

1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

Atmel Studio 7.0

- Simulator
- Debugger
-

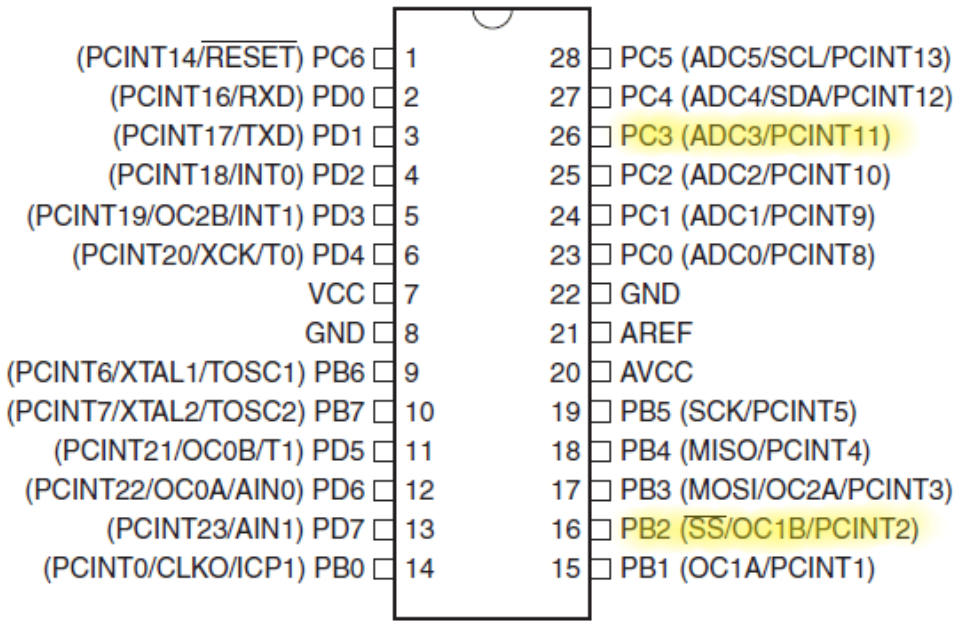
Atmega328PB-Xmini
Multi-Function Shield

- LEDS
- Switches

Logic Analyzer

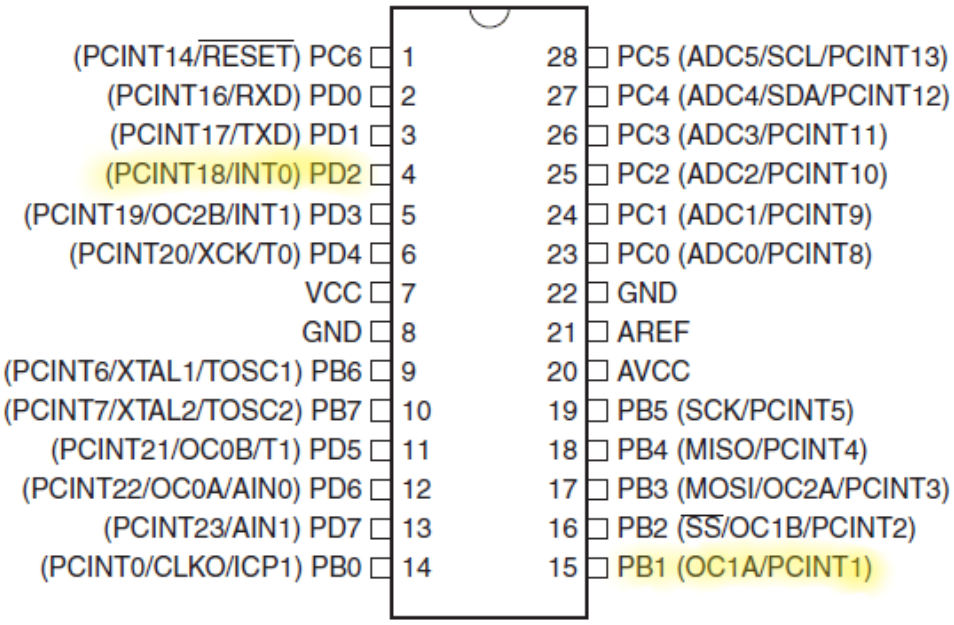
ATMEGA328

Port Pin used for task 2



ATMEGA328

Port Pin used for task 3



-

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

Assembly Code for Task

```
.org 0x00                              ; set program origin in memory

CALL DELAY_250ms     ; CALL DELAY_250ms
SBI PORTC,3          ; set PORTC.3 to pull-up resistor
main:

SBIS PINC,3          ; if button at PORTC.3 pressed
RJMP button          ; then it jumps to label button

RJMP main

button:
SBI DDRB, 2                    ; set PORTB.2 output ‘LED ON”

CALL DELAY_1250ms              ; CALL DELAY_1250ms
CBI DDRB, 2                    ; clear PORTB.2 output
CBI PORTB, 2                   ; turn off LED at PORTB.2

SBIS PINC, 3           ; if button at PORTC.3 pressed
RJMP button                    ; then it jumps to label button

RJMP main
```

```
; task 1
DELAY_250ms:   ; 0.25 second delay
LDI R20, 4
D0:LDI R21,100
D1:LDI R22, 124
NOP
NOP
D2:NOP
NOP
DEC R22
BRNE D2
DEC R21
BRNE D1
DEC R20
BRNE D0
RET

; task 2
DELAY_1250ms:  ; 1.25 second delay
LDI R20, 4
B0:LDI R21,249
B1:LDI R22, 250
NOP
NOP
B2:NOP
NOP
DEC R22
BRNE B2
DEC R21
BRNE B1
DEC R20
BRNE B0
RET
```

C Code for Task

```
#define F_CPU 16000000UL

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


int main(void)
{
```

```
/*****************
*Task 2
***************/
  DDRC &=~(1<<3);//Makes PortC.3 input
  PORTC |=(1<<3); //Pull up




   while (1)
   {
if(!(PINC &(1<<3)))
{
DDRB |= (1<<2);                  // PORTB.2 OUTPUT "LED ON"
_delay_ms(1250);       //DELAY FOR 1.25sec
DDRB &=~(1<<2);  // PORTB.2 OUTPUT "LED OFF"
PORTB &=~ (1<<2); // PORTB.2 OUTPUT "LED OFF"
}

   }
}
```

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

Assembly Code for Task

```
JMP main
.ORG 0x02 ;location for external interrupt 0
JMP EX0_ISR

;task 1
CALL DELAY_250ms
;task 2
SBI PORTC,3          ;set PORTC.3 to pull-up resistor

;task 3
LDI R20,HIGH(RAMEND)
OUT SPH,R20 ;initialize higher stack
LDI R20,LOW(RAMEND)
OUT SPL,R20 ;initialize Lower stack

main:
;task 2
SBIS PINC,3          ;if button at PORTC.3 pressed
RJMP button          ; then it jump to label button
```

```
;task 3
LDI R20,0x2 ;make INT0 falling edge triggered
STS EICRA,R20
SBI DDRB,4 ;PORTB.1 = output
SBI PORTD,2 ;pull-up activated
LDI R20,1<<INT0 ;enable INT0
OUT EIMSK,R20
SEI ;enable interrupts



RJMP main

button:
;task 2
SBI DDRB, 2                    ;set PORTB.2 output ‘LED ON”

CALL DELAY_1250ms              ;CALL DELAY_1250ms
CBI DDRB, 2                    ;clear PORTB.2 output
CBI PORTB, 2                   ; turn off LED at PORTB.2

SBIS PINC, 3          ;if button at PORTC.3 pressed
RJMP button                    ; then it jump to label button

RJMP main


;task 3
EX0_ISR:
SBI DDRB, 1                    ;set PORTB.2 output ‘LED ON”
CALL DELAY_500ms
CBI DDRB, 1                    ;clear PORTB.2 output
CBI PORTB, 1                   ; turn off LED at PORTB.2
RETI



;task 1
DELAY_250ms: ; 0.25 second delay
LDI R20, 4
D0:LDI R21,100
D1:LDI R22, 124
NOP
NOP
```

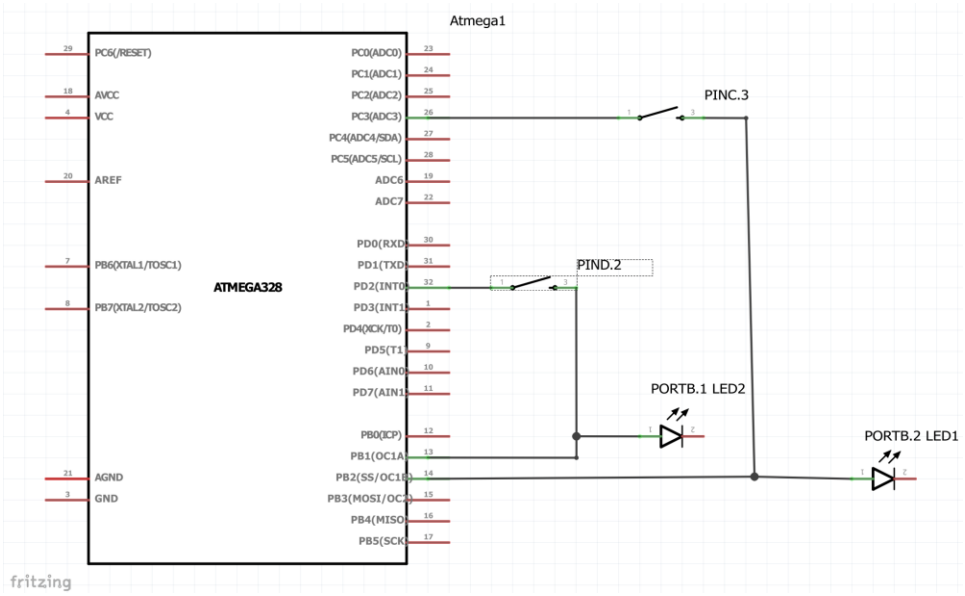
```
D2:NOP
NOP
DEC R22
BRNE D2
DEC R21
BRNE D1
DEC R20
BRNE D0
RET

;task 2
DELAY_1250ms: ; 1.25 seconds
LDI R20, 125
B0:
LDI R21, 125
B1:
LDI R22, 255
B2:
NOP
NOP
DEC R22
BRNE B2
DEC R21
BRNE B1
DEC R20
BRNE B0
RET

  ;task 3
DELAY_500ms:  ;0.5 second delay
LDI R20, 4
C0:
  LDI R21,200
C1:
  LDI R22, 124
C2:
NOP
NOP
DEC R22
BRNE C2
DEC R21
BRNE C1
DEC R20
BRNE C0
RET
```

C Code for Task

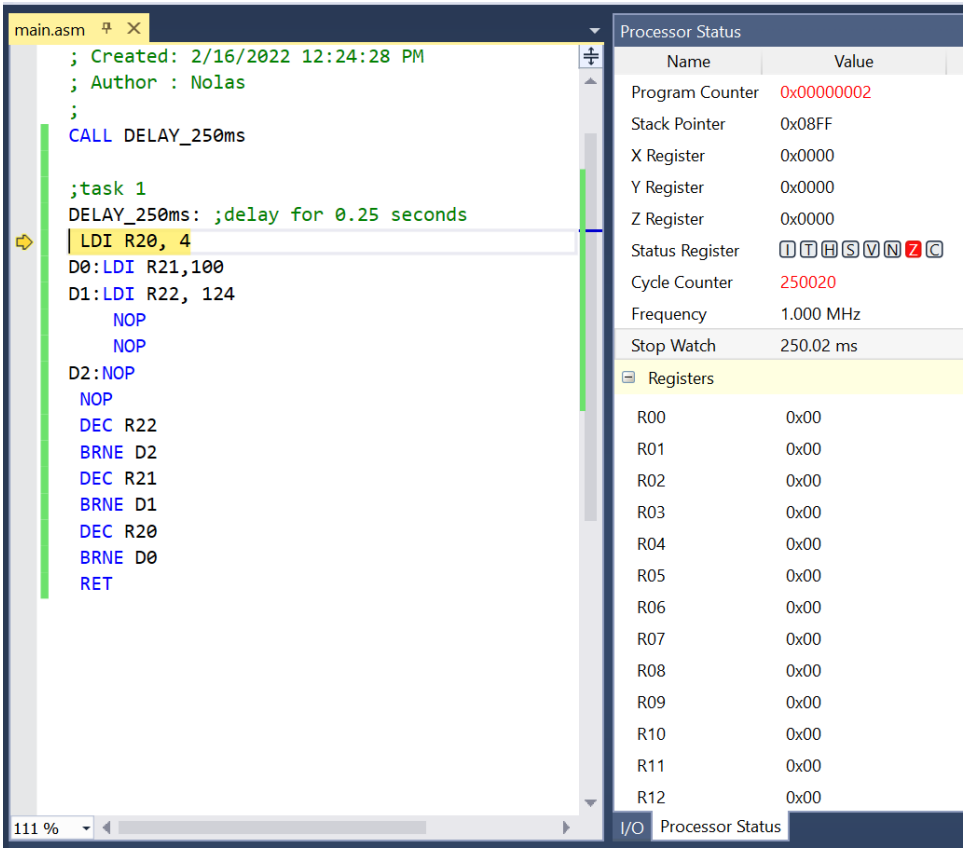
```
#define F_CPU 16000000UL

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


int main(void)
{
/****************
*Task 2
**************/
  DDRC &=~(1<<3);//Makes PortC.3 input
  PORTC |=(1<<3); //Pull up

  /************************
  *Task 3
  ************************/
  DDRB |= (1<<4);//PORTB.4 OUTPUT
  PORTB &=~(1<<4);//PORTB.4 OUTPUT
  DDRD &=~(1<<2);//PORTD.2 INPUT
  PORTD |= (1<<2);//PORTD.2 INPUT
  EICRA =0x2;//falling edge

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



   while (1)
   {
if(!(PINC &(1<<3)))
{
DDRB |= (1<<2);                    // PORTB.2 OUTPUT "LED ON"
_delay_ms(1250);          //DELAY FOR 1.25sec
DDRB &=~(1<<2);  // PORTB.2 OUTPUT "LED OFF"
PORTB &=~ (1<<2); // PORTB.2 OUTPUT "LED OFF"
}

   }
}

ISR(INT0_vect)
{
DDRB |= (1<<4);                    // PORTB.4 OUTPUT "LED ON"
_delay_ms(500);           //DELAY FOR 1.25sec
```

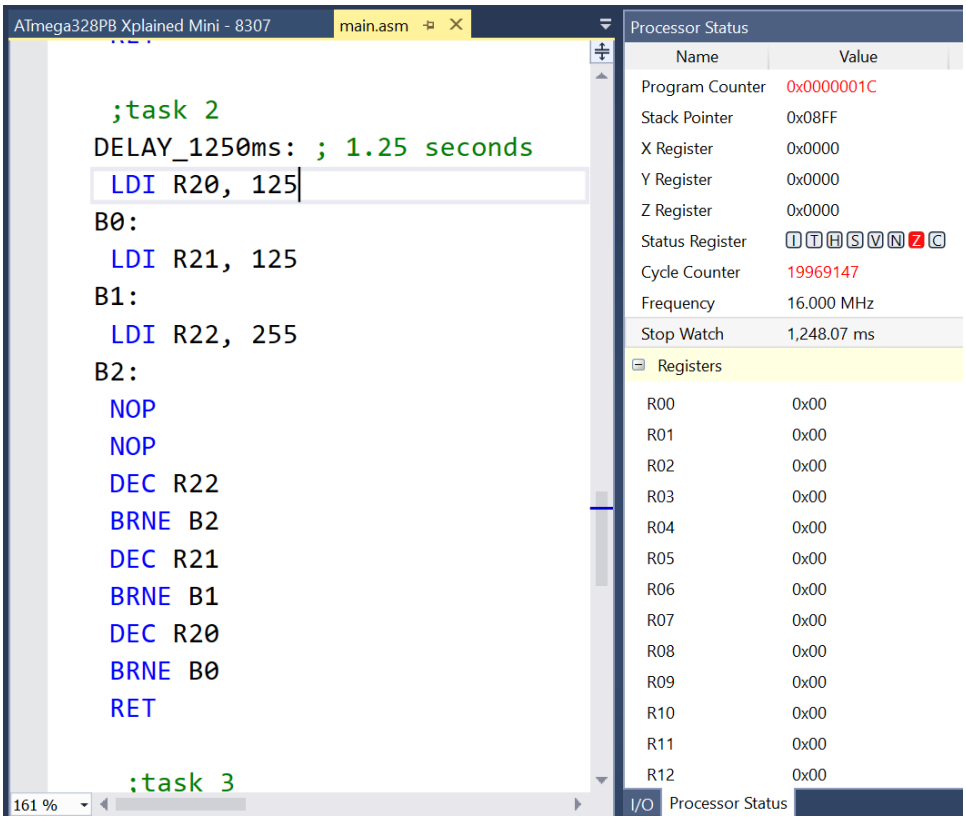
```
DDRB &=~(1<<4);  // PORTB.4 OUTPUT "LED OFF"
PORTB &=~ (1<<4); // PORTB.4 OUTPUT
}
```

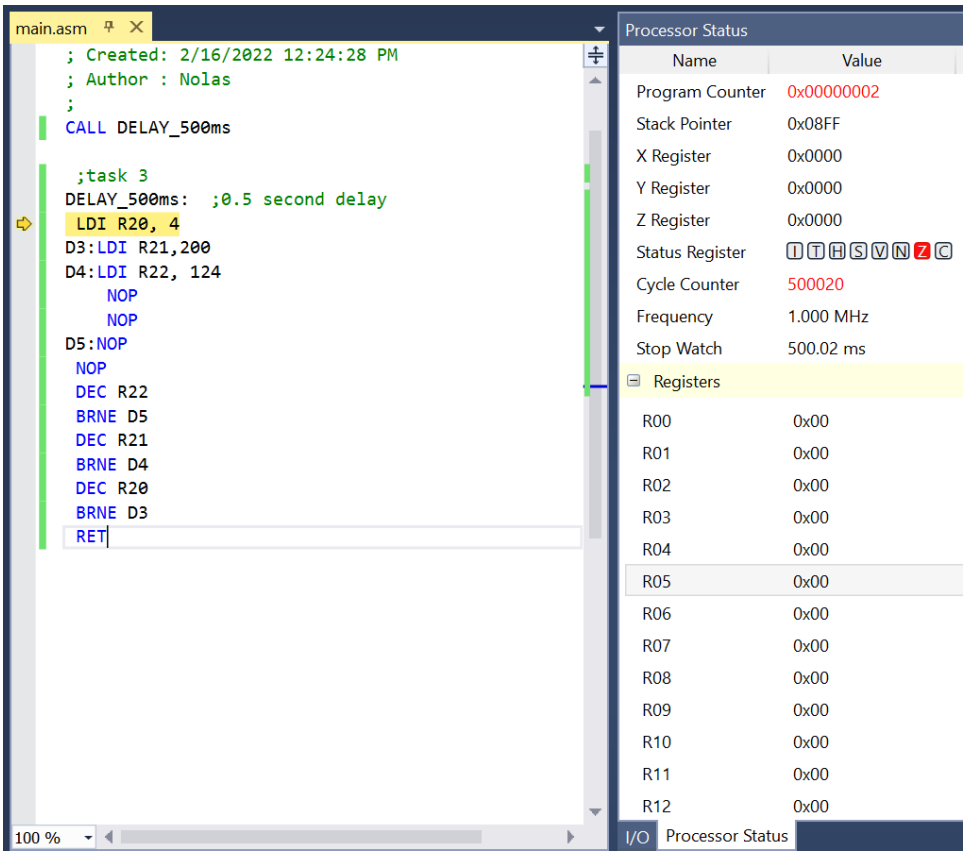
4. SCHEMATICS



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


main.asm
; Created: 2/16/2022 12:24:28 PM
; Author : Nolas
;
CALL DELAY_250ms
;task 1
DELAY_250ms: ;delay for 0.25 seconds
LDI R20, 4
D0:LDI R21,100
D1:LDI R22, 124
NOP
NOP
D2:NOP
NOP
DEC R22
BRNE D2
DEC R21
BRNE D1
DEC R20
BRNE D0
RET
111 %
Processor Status
Name
Value
Program Counter
0x00000002
Stack Pointer
0x08FF
X Register
0x0000
Y Register
0x0000
Z Register
0x0000
Status Register
I T H S V N Z C
Cycle Counter
250020
Frequency
1.000 MHz
Stop Watch
250.02 ms
Registers
R00 0x00
R01 0x00
R02 0x00
R03 0x00
R04 0x00
R05 0x00
R06 0x00
R07 0x00
R08 0x00
R09 0x00
R10 0x00
R11 0x00
R12 0x00
I/O
Processor Status


ATmega328PB Xplained Mini - 8307
main.asm
;task 2
DELAY_1250ms: ; 1.25 seconds
LDI R20, 125
B0:
LDI R21, 125
B1:
LDI R22, 255
B2:
NOP
NOP
DEC R22
BRNE B2
DEC R21
BRNE B1
DEC R20
BRNE B0
RET
;task 3
161 %
Processor Status
Name
Value
Program Counter
0x0000001C
Stack Pointer
0x08FF
X Register
0x0000
Y Register
0x0000
Z Register
0x0000
Status Register
I T H S V N Z C
Cycle Counter
19969147
Frequency
16.000 MHz
Stop Watch
1,248.07 ms
Registers
R00 0x00
R01 0x00
R02 0x00
R03 0x00
R04 0x00
R05 0x00
R06 0x00
R07 0x00
R08 0x00
R09 0x00
R10 0x00
R11 0x00
R12 0x00
I/O
Processor Status

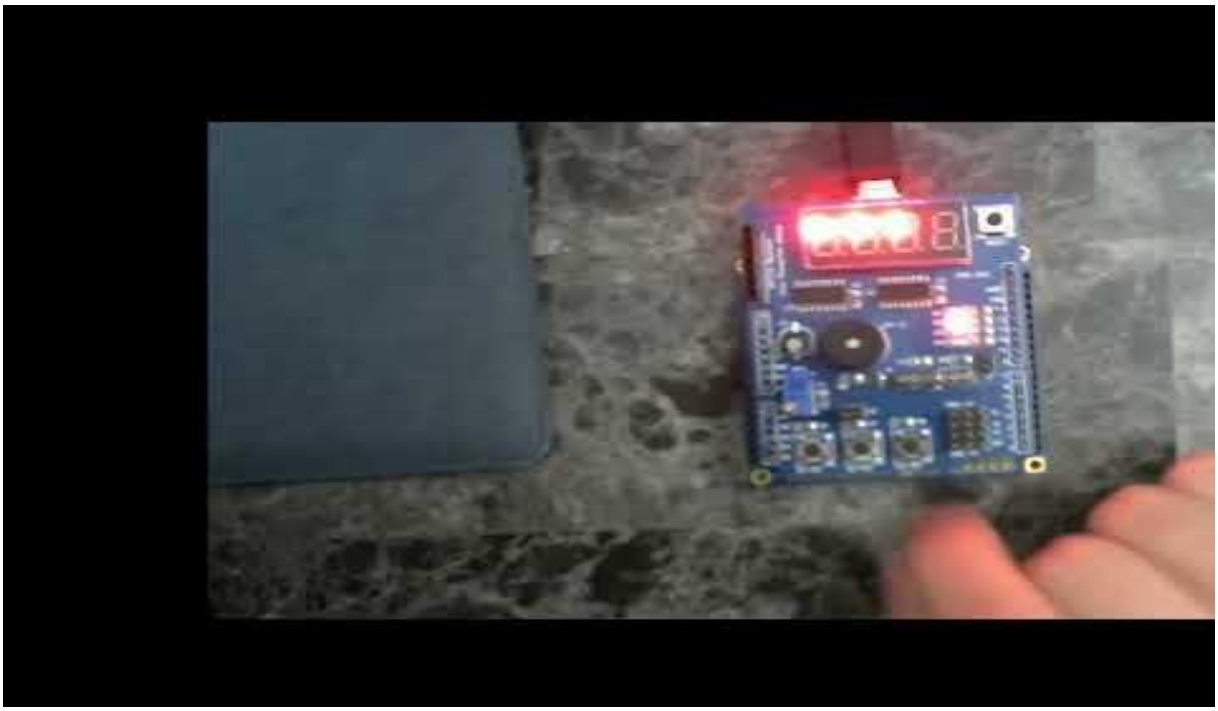


6. SCREENSHOT OF EACH DEMO (BOARD SETUP)

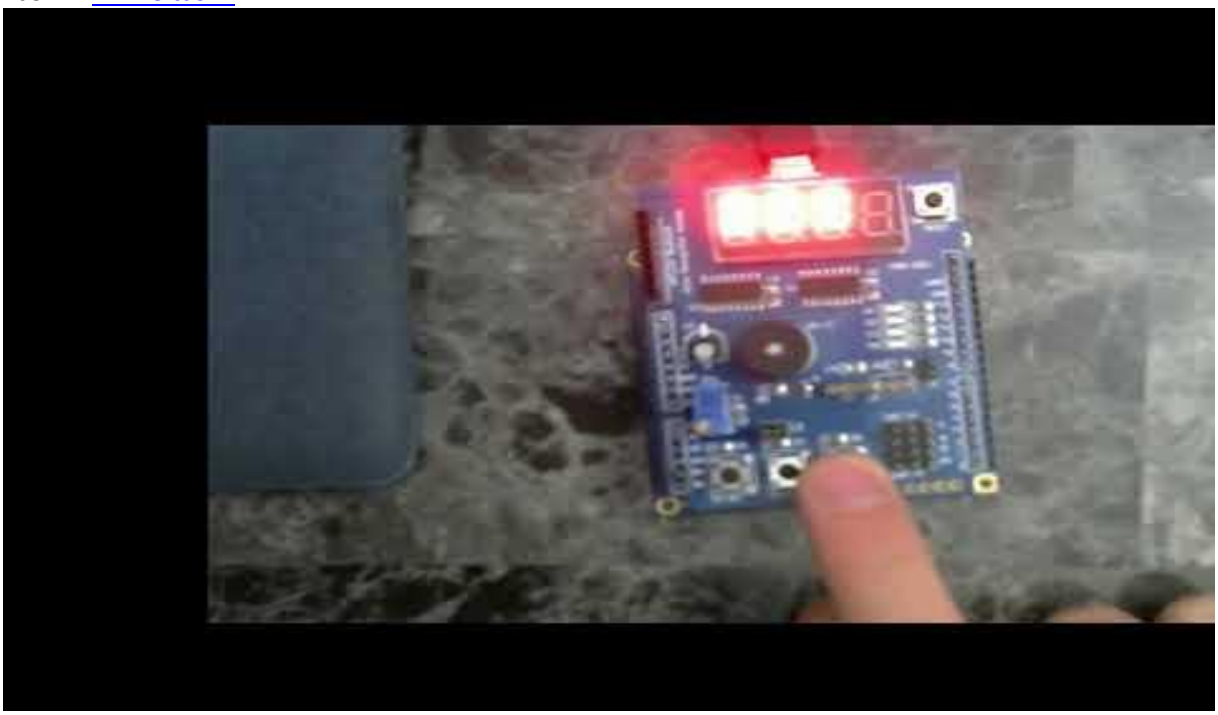
7. VIDEO LINKS OF EACH DEMO

ASM

Task 2: DA2 asm task2



C
Task 2: DA2 C task2



8. GITHUB LINK OF THIS DA

https://github.com/AngeloNol/DA_submission