Design Assignment 4

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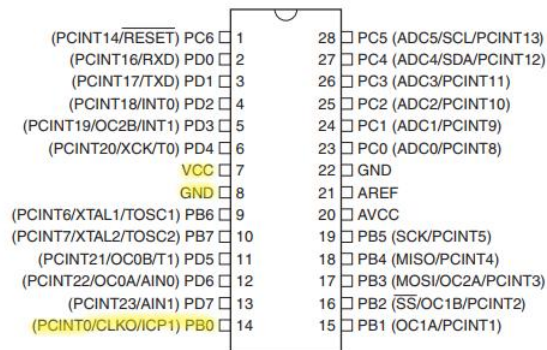Directory:

1.   COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

Atmel Studio 7.0
-    Simulator
-    Debugger
-    Atmega328PB-Xmini
-    3-pin ultrasonic

ATMEGA328

Port Pin

```
        (PCINT14/RESET) PC6 ⌐1      28⌐ PC5 (ADC5/SCL/PCINT13)
          (PCINT16/RXD) PD0 ⌐2      27⌐ PC4 (ADC4/SDA/PCINT12)
          (PCINT17/TXD) PD1 ⌐3      26⌐ PC3 (ADC3/PCINT11)
         (PCINT18/INT0) PD2 ⌐4      25⌐ PC2 (ADC2/PCINT10)
    (PCINT19/OC2B/INT1) PD3 ⌐5      24⌐ PC1 (ADC1/PCINT9)
       (PCINT20/XCK/T0) PD4 ⌐6      23⌐ PC0 (ADC0/PCINT8)
                       VCC ⌐7      22⌐ GND
                       GND ⌐8      21⌐ AREF
   (PCINT6/XTAL1/TOSC1) PB6 ⌐9      20⌐ AVCC
   (PCINT7/XTAL2/TOSC2) PB7 ⌐10     19⌐ PB5 (SCK/PCINT5)
       (PCINT21/OC0B/T1) PD5 ⌐11     18⌐ PB4 (MISO/PCINT4)
    (PCINT22/OC0A/AIN0) PD6 ⌐12     17⌐ PB3 (MOSI/OC2A/PCINT3)
          (PCINT23/AIN1) PD7 ⌐13     16⌐ PB2 (SS/OC1B/PCINT2)
     (PCINT0/CLKO/ICP1) PB0 ⌐14     15⌐ PB1 (OC1A/PCINT1)
```

-

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

C Code for Task

```c
int main(void)
{
char string[10];
long count;
```

```c
double distance;

DDRB = 0x02;                        /* Make trigger pin as output */
/* PB0 is the Echo Pin & PB1 is the Trigger in */
USART_init();



sei();                              /* Enable global interrupt */
TIMSK1 = (1 << TOIE1);    /* Enable Timer1 overflow interrupts */
TCCR1A = 0;                         /* Set all bit to zero Normal operation */

while(1)
{

PORTB |= (1 << Trigger_pin);/* Give 10us trigger pulse on trig. pin to HC-SR04 */
_delay_us(10);
PORTB &= (~(1 << Trigger_pin));

TCNT1 = 0;                  /* Clear Timer counter */
TCCR1B = 0x41;              /* Setting for capture rising edge, No pre-scaler*/
TIFR1 = 1<<ICF1;           /* Clear ICP flag (Input Capture flag) */
TIFR1 = 1<<TOV1;           /* Clear Timer Overflow flag */



/*Calculate width of Echo by Input Capture (ICP) on PortB PB0 */
PORTB &= (~(1 << Trigger_pin));

while ((TIFR1 & (1 << ICF1)) == 0);        /* Wait for rising edge */
TCNT1 = 0;                  /* Clear Timer counter */
TCCR1B = 0x01;              /* Setting for capture falling edge, No pre-scaler */
TIFR1 = 1<<ICF1;           /* Clear ICP flag (Input Capture flag) */
TIFR1 = 1<<TOV1;           /* Clear Timer Overflow flag */

TimerOverflow = 0;   /* Clear Timer overflow count */

while ((TIFR1 & (1 << ICF1)) == 0); /* Wait for falling edge */
count = ICR1 + (65535 * TimerOverflow);   /* Take value of capture register */
/* 8MHz Timer freq, sound speed =343 m/s, calculation mentioned in doc. */
distance = (double)count / (58*16);


dtostrf(distance, 2, 2, string);/* Convert distance into string */
strcat(string, " cm   ");
USART_putstring("Dist = ");
```
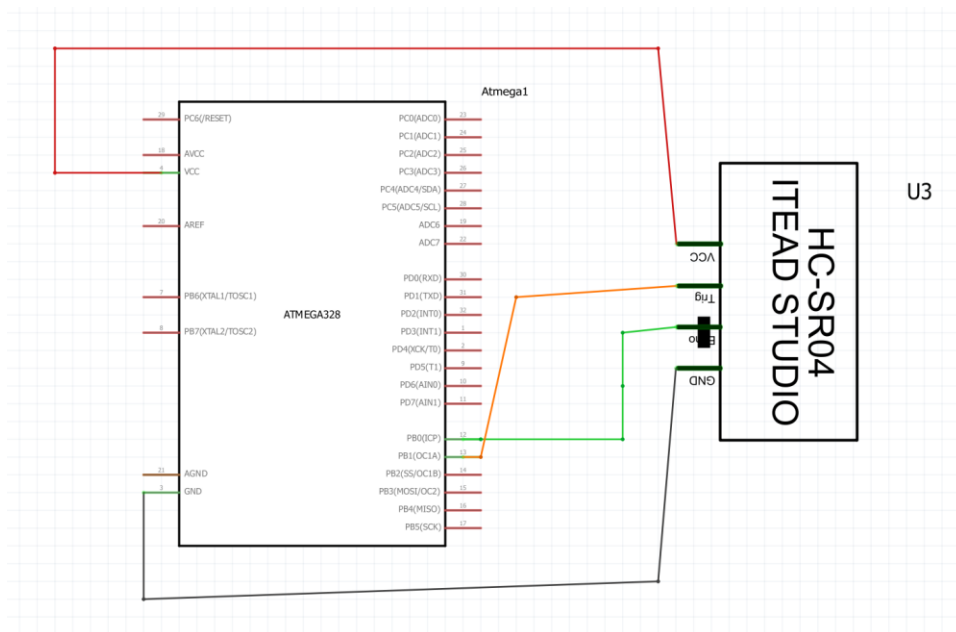
```
USART_putstring(string);        /* Print distance on Terminal */
USART_putstring("\n");
_delay_ms(1000);
}
}
```

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

4.   SCHEMATICS

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

```
        sei();                    /* Enable global interrupt */
        TIMSK1 = (1 << TOIE1);    /* Enable Timer1 overflow interrupts */
        TCCR1A = 0;               /* Set all bit to zero Normal operation */

        while(1)
        {

            PORTB |= (1 << Trigger_pin);/* Give 10us trigger pulse on trig. pin to HC-SR04 */
            _delay_us(10);
            PORTB &= (~(1 << Trigger_pin));

            TCNT1 = 0;            /* Clear Timer counter */
            TCCR1B = 0x41;        /* Setting for capture rising edge, No pre-scaler*/
            TIFR1 = 1<<ICF1;      /* Clear ICP flag (Input Capture flag) */
            TIFR1 = 1<<TOV1;      /* Clear Timer Overflow flag */


                /*Calculate width of Echo by Input Capture (ICP) on PortB PB0 */
                PORTB &= (~(1 << Trigger_pin));

                while ((TIFR1 & (1 << ICF1)) == 0); /* Wait for rising edge */
                TCNT1 = 0;            /* Clear Timer counter */
                TCCR1B = 0x01;        /* Setting for capture falling edge, No pre-scaler */
                TIFR1 = 1<<ICF1;      /* Clear ICP flag (Input Capture flag) */
                TIFR1 = 1<<TOV1;      /* Clear Timer Overflow flag */
                TimerOverflow = 0;    /* Clear Timer overflow count */

                while ((TIFR1 & (1 << ICF1)) == 0); /* Wait for falling edge */
                count = ICR1 + (65535 * TimerOverflow); /* Take value of capture register */
                /* 8MHz Timer freq, sound speed =343 m/s, calculation mentioned in doc. */
                distance = (double)count / (58*16);


            dtostrf(distance, 2, 2, string);/* Convert distance into string */
            strcat(string, " cm   ");
            USART_putstring("Dist = ");
            USART_putstring(string);     /* Print distance on Terminal */
            USART_putstring("\n");
            _delay_ms(1000);
        }
```

6.   SCREENSHOT OF EACH DEMO (BOARD SETUP)
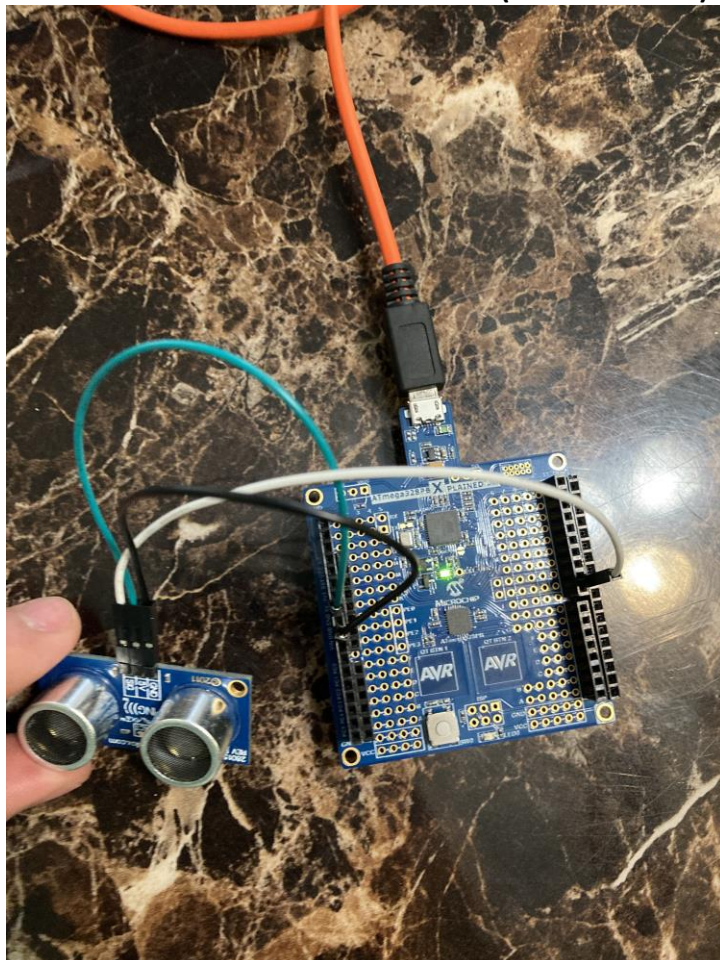
## 7.	VIDEO LINKS OF EACH DEMO
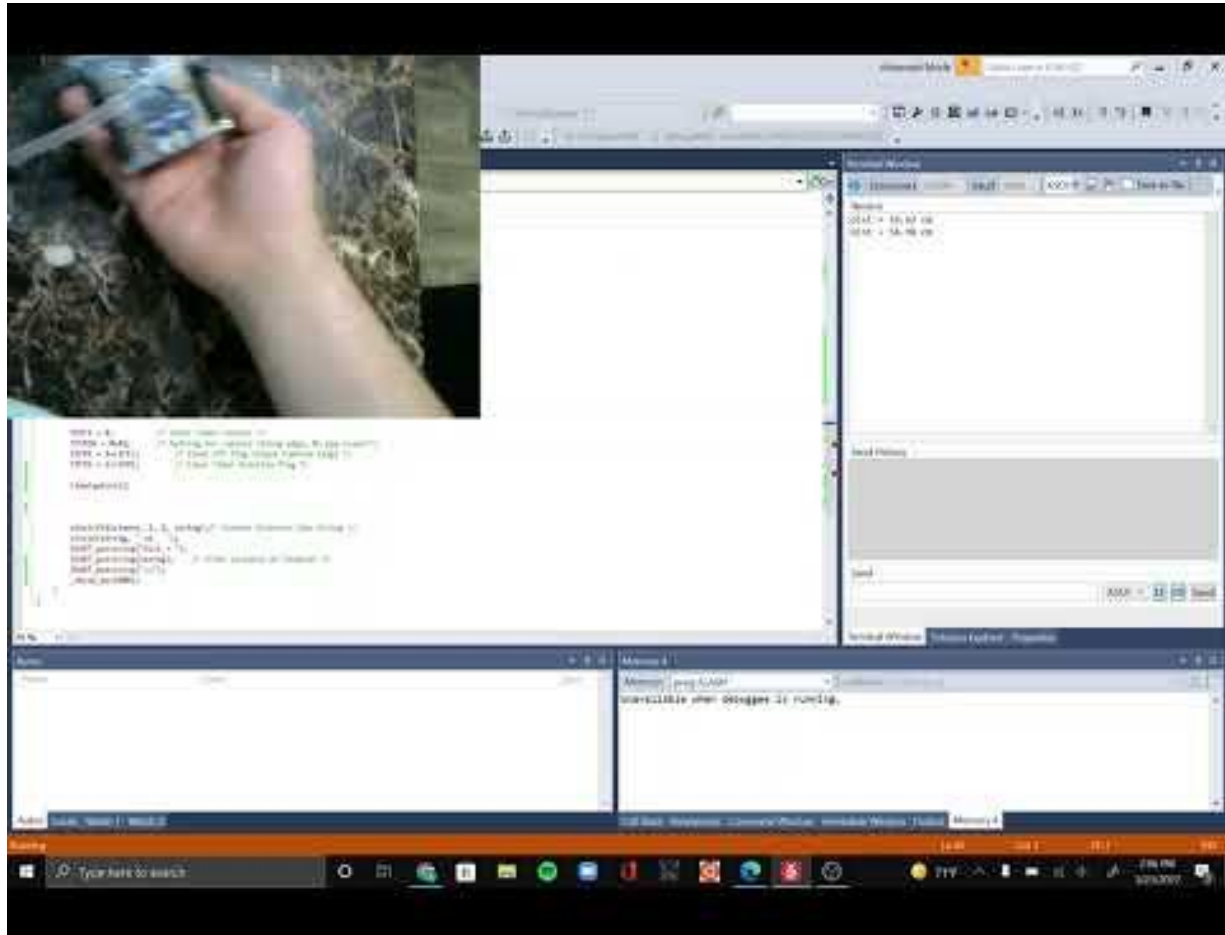
DA4 C

## 8.	GITHUB LINK OF THIS DA
https://github.com/AngeloNol/DA_submission