#### **CPE301 – SPRING 2022**

# MIDTERM 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

Servo motor

HC-SR04 ultrasonic

ATMEGA328

Port Pin Used

```
(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)
                                 23 PC0 (ADC0/PCINT8)
   (PCINT20/XCK/T0) PD4 ☐ 6
                  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)
  (PCINTO/CLKO/ICP1) PB0 ☐ 14
                                 15 PB1 (OC1A/PCINT1)
```

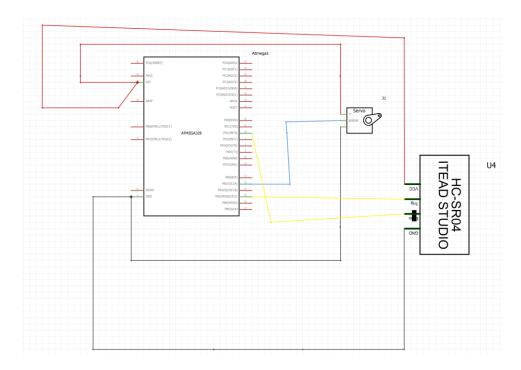
## 2. INITIAL/MODIFIED/DEVELOPED CODE OF TASK 1/A

```
void initUSART()
{
UBRROH = (uint8_t)(BAUD_PRESCALLER>>8); //this sets the BAUD rate, must shift UBRROH 8
bits to the right
UBRROL = (uint8_t)(BAUD_PRESCALLER); //sets lower 8 bits of UBRRO
UCSROB = (1<<TXENO); //enables send of data
UCSROC = (3<<UCSZOO); //it is set to 8 bit
}

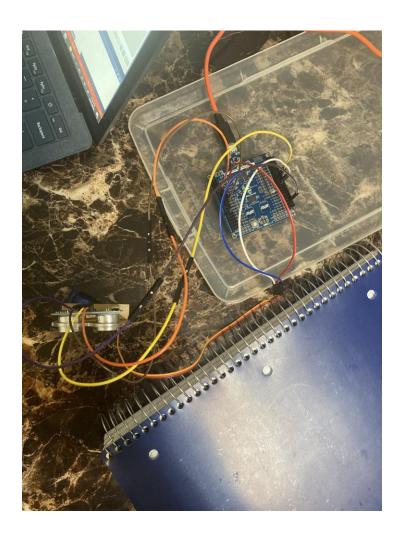
void init()
{</pre>
```

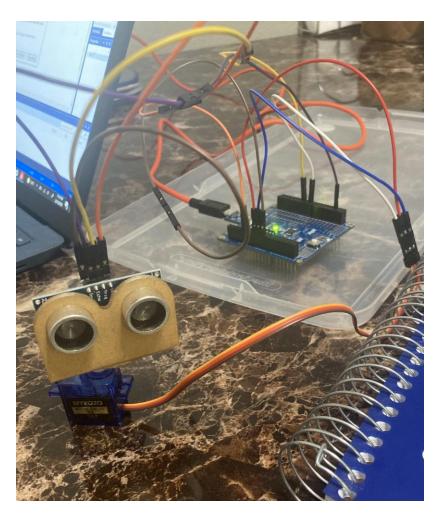
```
DDRB = 0xFF; //set PortB as output
PORTB = 0x00; //set PORTB as input
DDRD &= ^(1 << echoPin);
                            //echoPin is input
EIMSK |= (1<<INT0);
                            //interrupt INTO enabled
TIMSK2 |= (1<<TOIE2);
                            //enable overflow interrupt on Timer 2
EICRA |= (1<<ISC00); // interrupt on rising edge and falling edge
}
void initServo()
DDRB |= (1<<PINB1); // Set pin 1 to output PWM
TCCR1A |= (1 << WGM11) | (1 << COM1A1);// Set Fast PWM mode
TCCR1B |= (1 << WGM13) | (1 << WGM12) | (1 << CS11);//Set pre-scale of 8,non-inverting mode
ICR1 = 40000;
}
void signalPulse()
PORTB |= (1<<trigPin);
                            //set trigPin high
_delay_us(10); // delay 10us
PORTB \&= \sim (1 << trigPin);
                                   //set trigPin low again
}
void putStringUSART(char* stringPtr)
while (*stringPtr != 0) {
                            //while the data the pointer points to is not null
                                  //wait to receive data
while (!(UCSROA & (1<<UDRE0)));
UDR0 = *stringPtr; //UDR0 = dereferenced stringPtr characters
stringPtr++; //advance stringPtr by 1
}
void Loop()
int16 t pulseRange = 0;
                            //counter is 0 initially
                            //char array, used for sending string to USART
char pulseString[5];
                            //char array, for sending angle to USART
char angleString[5];
//measure from 0 to 180 degrees
```

```
for (int angle = 0; angle <= 180; angle++) {
OCR1A = angle*25;
                     //OCR1A scales with the angle, multiplied by a constant
_delay_ms(30);
                     //30ms delay
signalPulse();
                     //call signal pulse
pulseRange = pulseEnd;
                            //pulseRange takes the value of TCNT2
putStringUSART("Distance: ");
itoa(pulseRange, pulseString, 10);
putStringUSART(pulseString);
putStringUSART(" cm, ");
putStringUSART("Angle: ");
itoa(angle, angleString, 10);
putStringUSART(angleString);
putStringUSART(".\n");
_delay_ms(1);
}
//measure from 180 back to 0 degrees
for (int angle = 180; angle >= 0; angle--) {
OCR1A = angle*25;
                            //OCR1A scales with the angle, multiplied by a constant
delay ms(30);
                            //30ms delay
signalPulse();
                     //call signal pulse
pulseRange = pulseEnd;
                                    //pulseRange takes the value of TCNT2
putStringUSART("Distance: ");
itoa(pulseRange, pulseString, 10);
putStringUSART(pulseString);
putStringUSART(" cm, ");
putStringUSART("Angle: ");
itoa(angle, angleString, 10);
putStringUSART(angleString);
putStringUSART(".\n");
delay ms(1);
}
}
3.
       DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A
N/A
4.
       SCHEMATICS
Use fritzing.org
```



- 5. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)  $_{\mbox{\scriptsize N/A}}$
- 6. SCREENSHOT OF EACH DEMO (BOARD SETUP)





7. VIDEO LINKS OF EACH DEMO Mid-Term 2 C program

## 8. GITHUB LINK OF THIS DA

https://github.com/AngeloNol/DA\_submission

"This assignment submission is my own, original work".  ${\bf Angelo\ Nolasco}$