

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

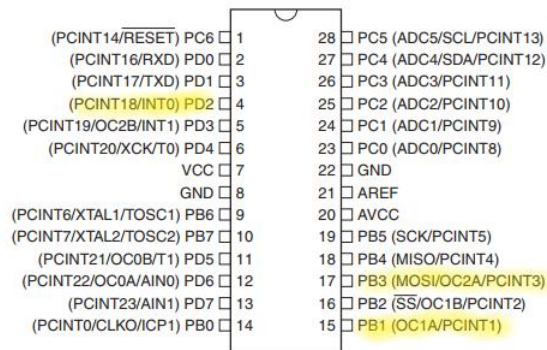
Atmega328PB-Xmini

Servo motor

HC-SR04 ultrasonic

ATMEGA328

Port Pin Used



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<<TXEN0); //enables send of data
    UCSROC = (3<<UCSZ00); //it is set to 8 bit
}

void init()
{
```

```

DDRB = 0xFF; //set PortB as output
PORTB = 0x00; //set PORTB as input
DDRD &= ~(1<<echoPin); //echoPin is input
EIMSK |= (1<<INT0); //interrupt INT0 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 &= ~(1<<trigPin); //set trigPin low again
}

```

```

void putStringUSART(char* stringPtr)
{
  while (*stringPtr != 0) { //while the data the pointer points to is not null
    while (!(UCSR0A & (1<<UDRE0))); //wait to receive data
    UDRO = *stringPtr; //UDRO = dereferenced stringPtr characters
    stringPtr++; //advance stringPtr by 1
  }
}

```

```

void Loop()
{
  int16_t pulseRange = 0; //counter is 0 initially
  char pulseString[5]; //char array, used for sending string to USART
  char angleString[5]; //char array, for sending angle to USART

```

```

//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);
}
}

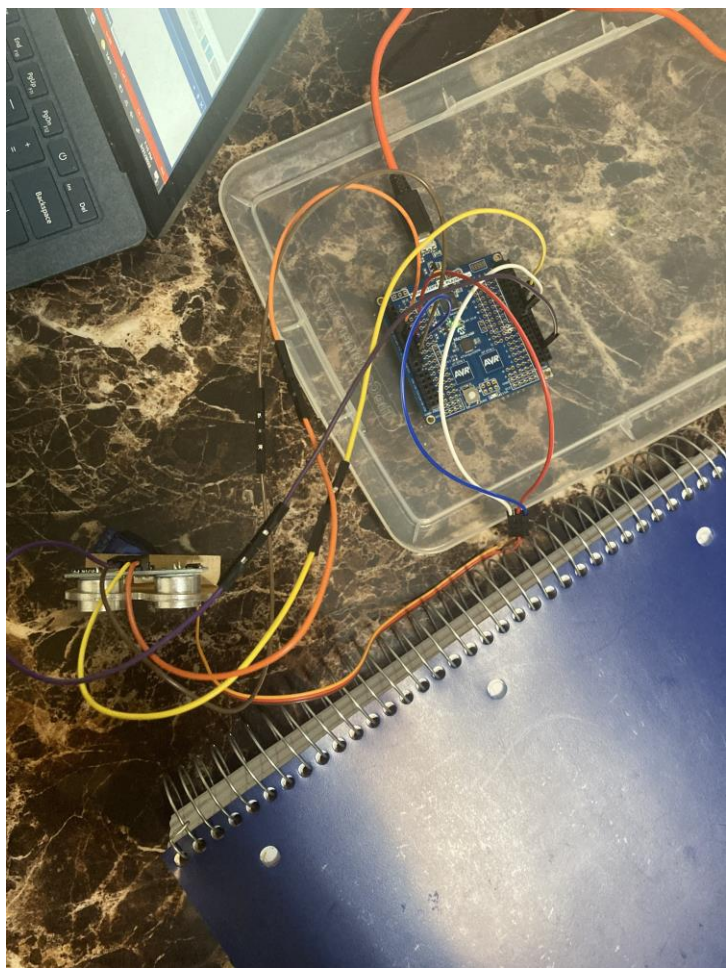
```

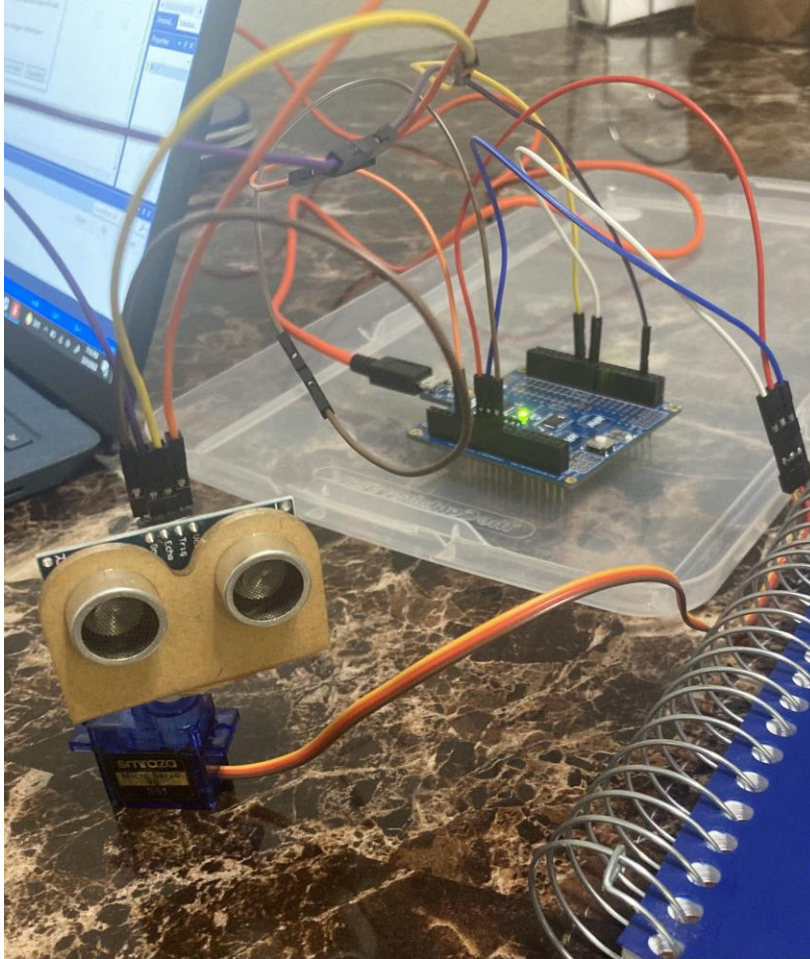
3. DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A

N/A

4. SCHEMATICS

Use fritzing.org





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".
Angelo Nolasco