CPE301 – SPRING 2019

Design Assignment 5

Student Name: Dillon Archibald / Alira Coffman
Student #: 5004439916 /
Student Email: Archid1@unlv.nevada.edu /
Primary Github address:
https://github.com/Dil-bert/Alabaster.git
https://github.com/Alira-Coffman/submission-repo.git
Directory:
Alabaster/DesignAssignments/**DA5**/
submission-repo/ESD301/DA/**DA5**/

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

List of Components used

- 1x Atmega328p Xplained mini
- 1x nrf24l0
- 1x LM35 temp sensor
- Wire

Block diagram with pins used in the Atmega328P

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

This CODE was obtained from Venki Repository as a starting point along with the libraries.

Insert initial code here

```
//      MIT License
//
//      Copyright (c) 2018 Helvijs Adams
//
//      Permission is hereby granted, free of charge, to any person obtaining a copy
//      of this software and associated documentation files (the "Software"), to deal
//      in the Software without restriction, including without limitation the rights
//      to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
//      copies of the Software, and to permit persons to whom the Software is
//      furnished to do so, subject to the following conditions:
//
//      The above copyright notice and this permission notice shall be included in all
//      copies or substantial portions of the Software.
//
//      THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
//      IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
//      FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
//      AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
//      LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
//      OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
//      SOFTWARE.

//
//      Software was tested on ATmega328P and ATmega328PB (PB needs few changes in SPI)
//      RF module software was tested on - cheap nRF24L01+ from China
//      All the relevant settings are defined in nrf24l01.c file
//      Some features will be added later, at this moment it is bare minimum to send/receive
//

//      Set clock frequency
#ifndef F_CPU
#define F_CPU 16000000UL
```

```
#endif

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

//      Set up UART for printf();
#ifndef BAUD
#define BAUD 9600
#endif
#include "inc\STDIO_UART.h"

//      Include nRF24L01+ library
#include "inc\nrf24l01.h"
#include "inc\nrf24l01-mnemonics.h"
#include "inc\spi.h"
void print_config(void);

//      Used in IRQ ISR
volatile bool message_received = false;
volatile bool status = false;

int main(void)
{
//      Set cliche message to send (message cannot exceed 32 characters)
char tx_message[32];                          // Define string array
strcpy(tx_message,"Hello World!");    // Copy string into array

//      Initialize UART
uart_init();

//      Initialize nRF24L01+ and print configuration info
    nrf24_init();
print_config();

//      Start listening to incoming messages
nrf24_start_listening();

    while (1)
    {
if (message_received)
{
//      Message received, print it
message_received = false;
printf("Received message: %s\n",nrf24_read_message());
//      Send message as response
_delay_ms(500);
status = nrf24_send_message(tx_message);
if (status == true) printf("Message sent successfully\n");
}
    }
}

//      Interrupt on IRQ pin
ISR(INT0_vect)
{
message_received = true;
}

void print_config(void)
{
uint8_t data;
printf("Startup successful\n\n nRF24L01+ configured as:\n");
printf("-------------------------------------------\n");
nrf24_read(CONFIG,&data,1);
printf("CONFIG         0x%x\n",data);
nrf24_read(EN_AA,&data,1);
printf("EN_AA                 0x%x\n",data);
nrf24_read(EN_RXADDR,&data,1);
printf("EN_RXADDR             0x%x\n",data);
nrf24_read(SETUP_RETR,&data,1);
printf("SETUP_RETR            0x%x\n",data);
nrf24_read(RF_CH,&data,1);
printf("RF_CH                 0x%x\n",data);
nrf24_read(RF_SETUP,&data,1);
printf("RF_SETUP              0x%x\n",data);
nrf24_read(STATUS,&data,1);
printf("STATUS         0x%x\n",data);
nrf24_read(FEATURE,&data,1);
printf("FEATURE        0x%x\n",data);
printf("-------------------------------------------\n\n");
}
```

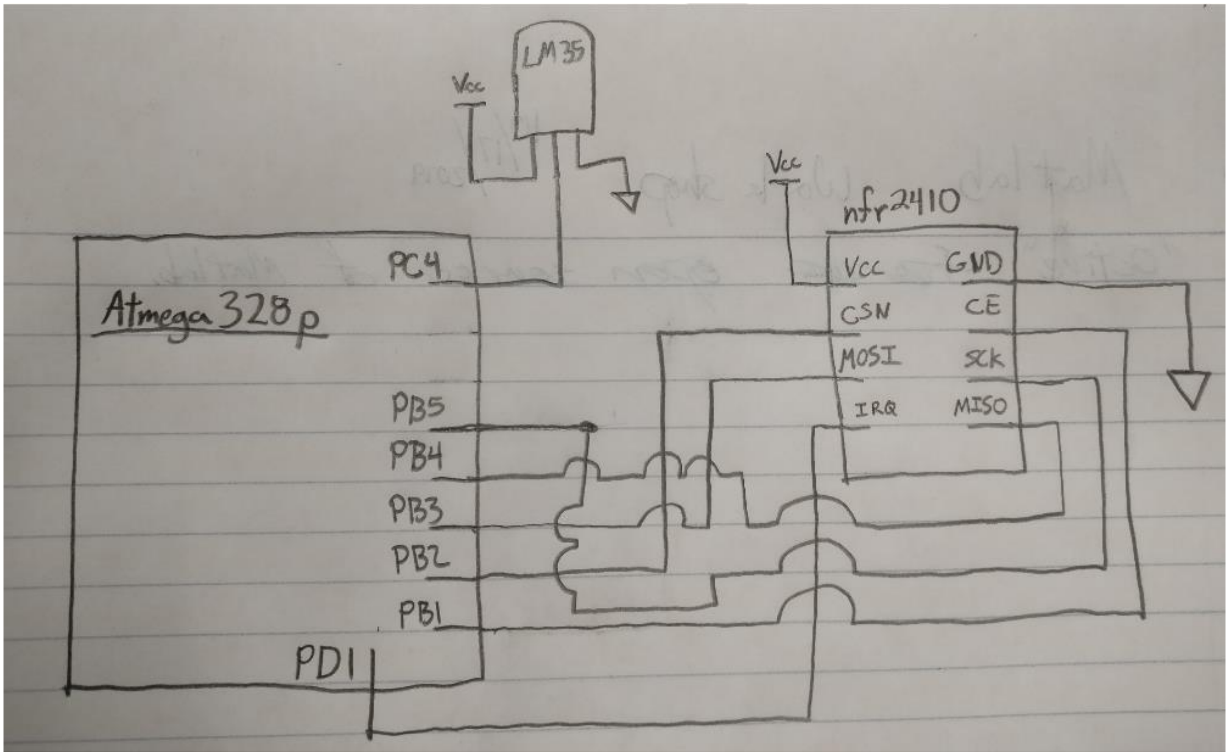
3. DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A

Insert only the modified sections here

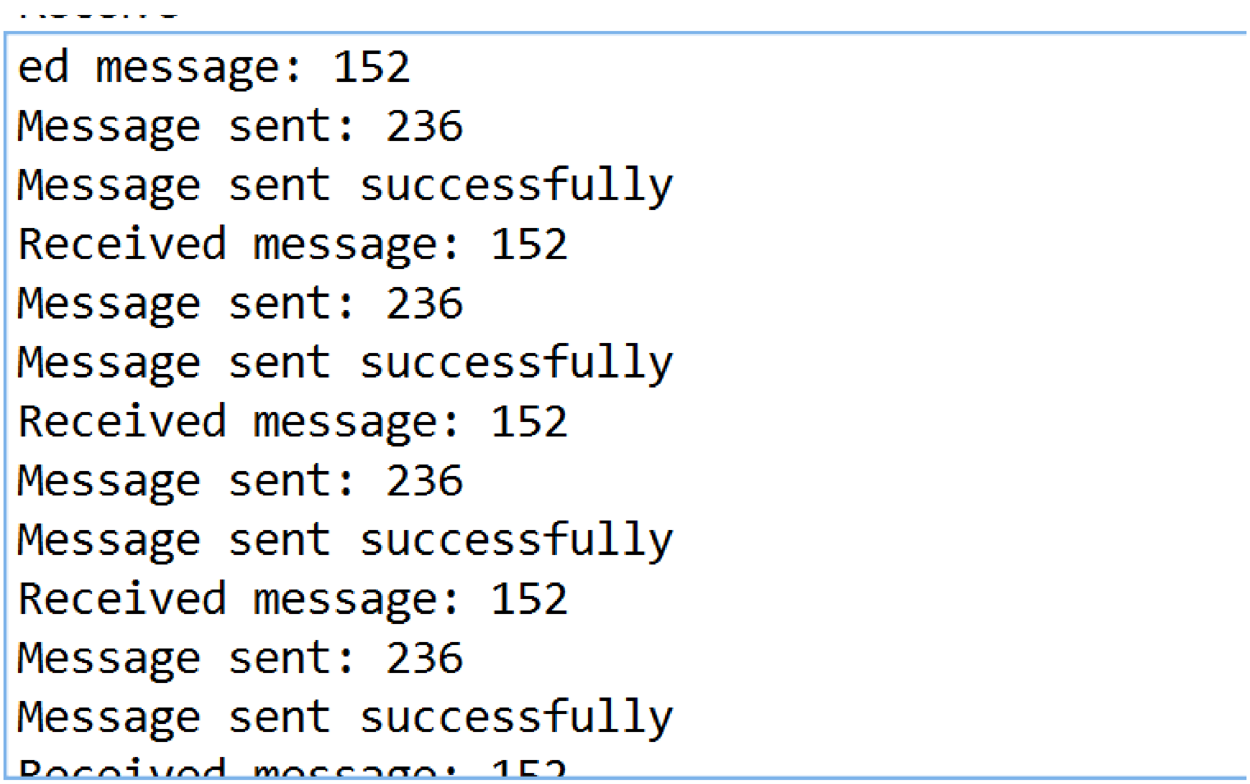
```
/*GOT PART OF THIS CODE FROM VENKI REPOSITORY*/
//      MIT License
//
//      Copyright (c) 2018 Helvijs Adams
//
//      Permission is hereby granted, free of charge, to any person obtaining a copy
//      of this software and associated documentation files (the "Software"), to deal
//      in the Software without restriction, including without limitation the rights
//      to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
//      copies of the Software, and to permit persons to whom the Software is
//      furnished to do so, subject to the following conditions:
//
//      The above copyright notice and this permission notice shall be included in all
//      copies or substantial portions of the Software.
```

```
//
//      THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
//      IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
//      FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
//      AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
//      LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
//      OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
//      SOFTWARE.

//
//      Software was tested on ATmega328P and ATmega328PB (PB needs few changes in SPI)
//      RF module software was tested on - cheap nRF24L01+ from China
//      All the relevant settings are defined in nrf24l01.c file
//      Some features will be added later, at this moment it is bare minimum to send/receive
//

//      Set clock frequency
#ifndef F_CPU
#define F_CPU 16000000UL
#endif

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

//      Set up UART for printf();
#ifndef BAUD
#define BAUD 9600
#endif
#include "STDIO_UART.h"

//      Include nRF24L01+ library
#include "nrf24l01.h"
#include "nrf24l01-mnemonics.h"
#include "spi.h"
void print_config(void);

//      Used in IRQ ISR
volatile bool message_received = false;
volatile bool status = false;



//adc functions
void adc_init(void){
 ADMUX = (0<<REFS1)|    // Reference Selection Bits
  (1<<REFS0)|    // AVcc - external cap at AREF
  (0<<ADLAR)|    // ADC Left Adjust Result
  (1<<MUX2)|     // Analog Channel Selection Bits
  (0<<MUX1)|     // ADC4 (PC4 PIN27)
  (1<<MUX0);
 ADCSRA = (1<<ADEN)|    // ADC ENable
  (0<<ADSC)|     // ADC Start Conversion
  (0<<ADATE)|    // ADC Auto Trigger Enable
  (0<<ADIF)|     // ADC Interrupt Flag
  (0<<ADIE)|     // ADC Interrupt Enable
  (1<<ADPS2)|    // ADC Prescaler Select Bits
  (0<<ADPS1)|
  (1<<ADPS0);
}

int main(void)
{
 //     Set cliche message to send (message cannot exceed 32 characters)
 char tx_message[32];                           // Define string array
 strcpy(tx_message,"Hello World!");   // Copy string into array
 char dummy[10];
 //     Initialize UART
 uart_init();
 adc_init();        //Setup the ADC

 //     Initialize nRF24L01+ and print configuration info
 nrf24_init();
 print_config();

 //     Start listening to incoming messages
 nrf24_start_listening();

 strcpy(tx_message,"GOOD");   // Copy string into array
 nrf24_send_message(tx_message);

 while (1)
  {
 _delay_ms(150);
 ADCSRA|=(1<<ADSC);    //start conversion
 while((ADCSRA&(1<<ADIF))==0);//wait for conversion to finish

 ADCSRA |= (1<<ADIF);
 int a = ADCL;
 a = a | (ADCH<<8);
 a = a*.488;
 a = a* (9/5) + 32;
 itoa(a, dummy, 10);
 if (message_received)
  {
 //     Message received, print it
 message_received = false;
 printf("Received message: %s\n",nrf24_read_message());
 //     Send message as response
 _delay_ms(500);
 status = nrf24_send_message(dummy);
 if (status == true) printf("Message sent successfully\n");
 }
 }
}

//      Interrupt on IRQ pin
ISR(INT0_vect)
{
 message_received = true;
}

void print_config(void)
{
 uint8_t data;
 printf("Startup successful\n\n nRF24L01+ configured as:\n");
 printf("------------------------------------------\n");
 nrf24_read(CONFIG,&data,1);
 printf("CONFIG        0x%x\n",data);
 nrf24_read(EN_AA,&data,1);
 printf("EN_AA                 0x%x\n",data);
 nrf24_read(EN_RXADDR,&data,1);
 printf("EN_RXADDR             0x%x\n",data);
 nrf24_read(SETUP_RETR,&data,1);
 printf("SETUP_RETR            0x%x\n",data);
 nrf24_read(RF_CH,&data,1);
 printf("RF_CH                 0x%x\n",data);
 nrf24_read(RF_SETUP,&data,1);
 printf("RF_SETUP              0x%x\n",data);
 nrf24_read(STATUS,&data,1);
 printf("STATUS        0x%x\n",data);
 nrf24_read(FEATURE,&data,1);
 printf("FEATURE               0x%x\n",data);
 printf("------------------------------------------\n\n");
}
```

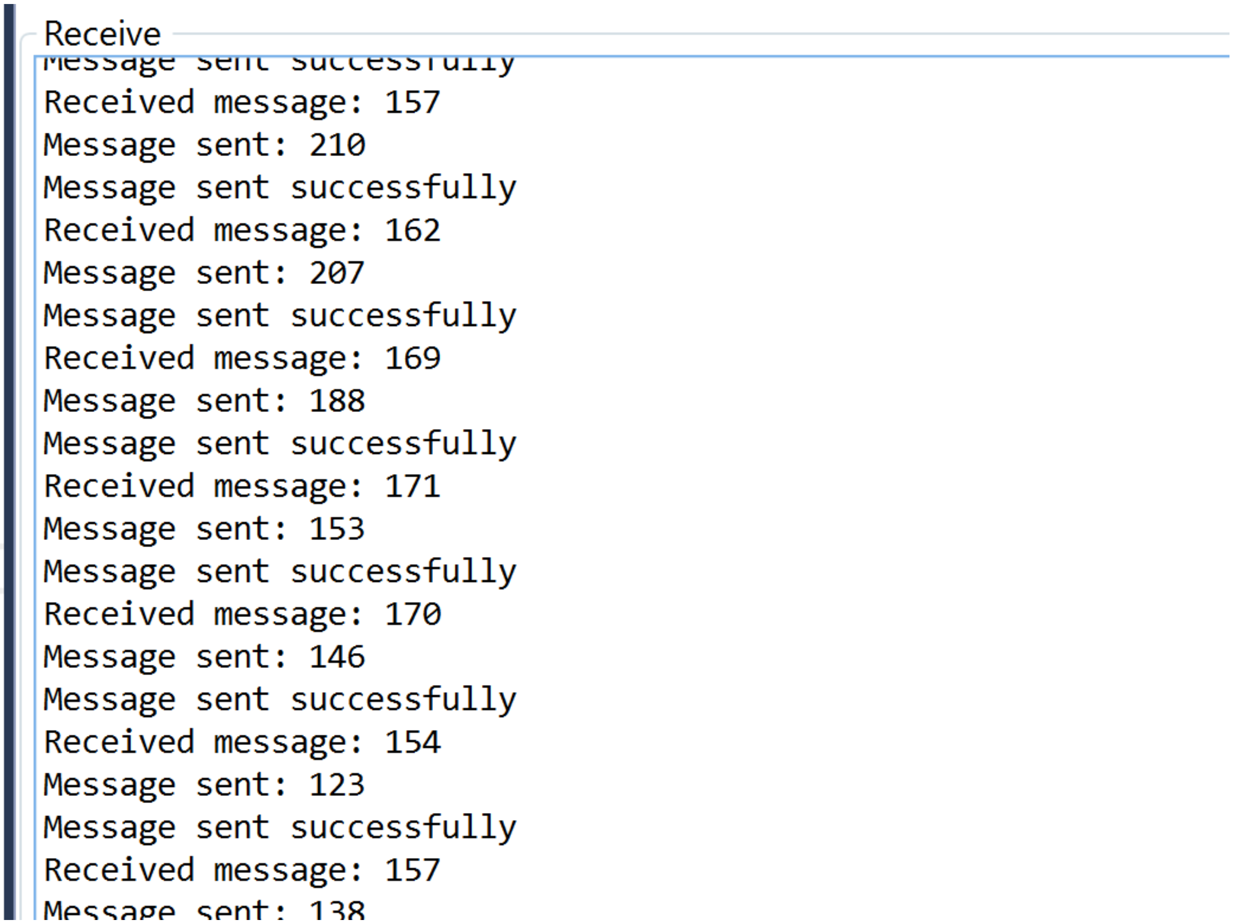
4. SCHEMATICS



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

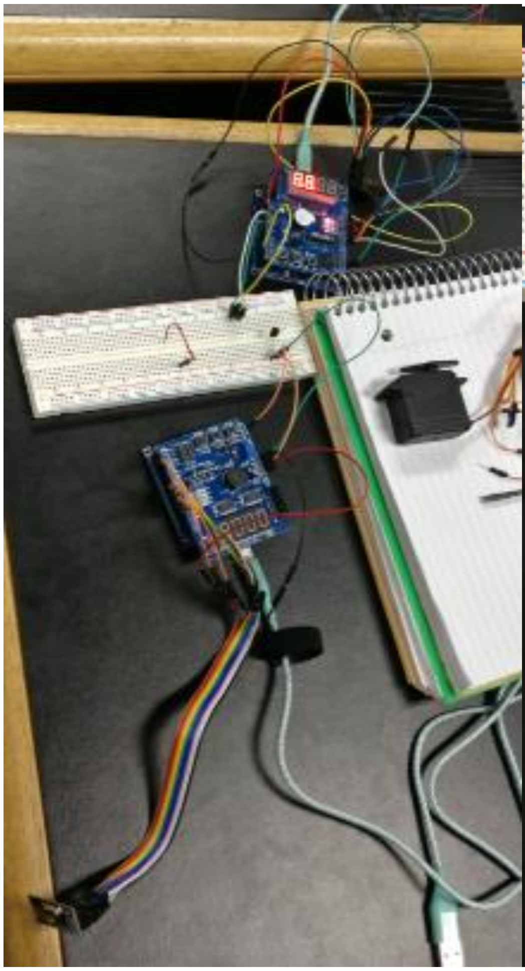


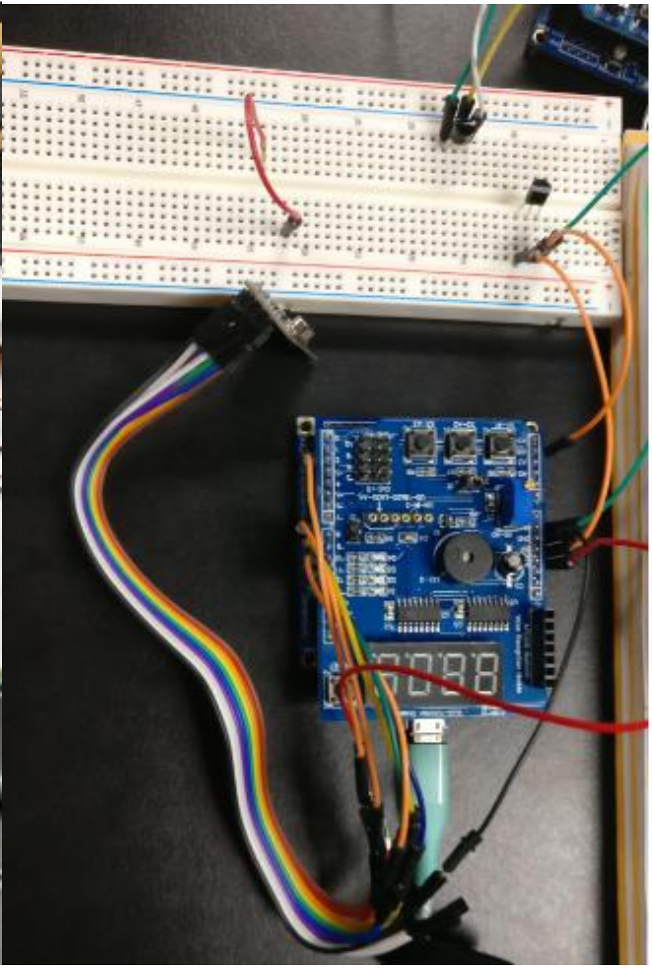
```
ed message: 152
Message sent: 236
Message sent successfully
Received message: 152
Message sent: 236
Message sent successfully
Received message: 152
Message sent: 236
Message sent successfully
Received message: 152
Message sent: 236
Message sent successfully
Received message: 152
```



```
Message sent successfully
Received message: 157
Message sent: 210
Message sent successfully
Received message: 162
Message sent: 207
Message sent successfully
Received message: 169
Message sent: 188
Message sent successfully
Received message: 171
Message sent: 153
Message sent successfully
Received message: 170
Message sent: 146
Message sent successfully
Received message: 154
Message sent: 123
Message sent successfully
Received message: 157
Message sent: 138
```

6. SCREENSHOT OF EACH DEMO (BOARD SETUP)





7. VIDEO LINKS OF EACH DEMO

Dillon’s

https://youtu.be/EK1uQx6BQhE

Alira’s

https://youtu.be/V3H2gRw2VyM

8. GITHUB LINK OF THIS DA

https://github.com/Dil-bert/Alabaster/tree/master/DesignAssignments/DA5

https://github.com/Alira-Coffman/submission-repo/tree/master/ESD301/DA/DA5

Student Academic Misconduct Policy

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

“This assignment submission is my own, original work”.
Dillon Archibald
Alira Coffman