CPE301 – FALL 2019

DA5

Student Name: Andrew Buchanan & Alira Coffman

Student #:5003154346

Student Email: buchaa2@unlv.nevada.edu

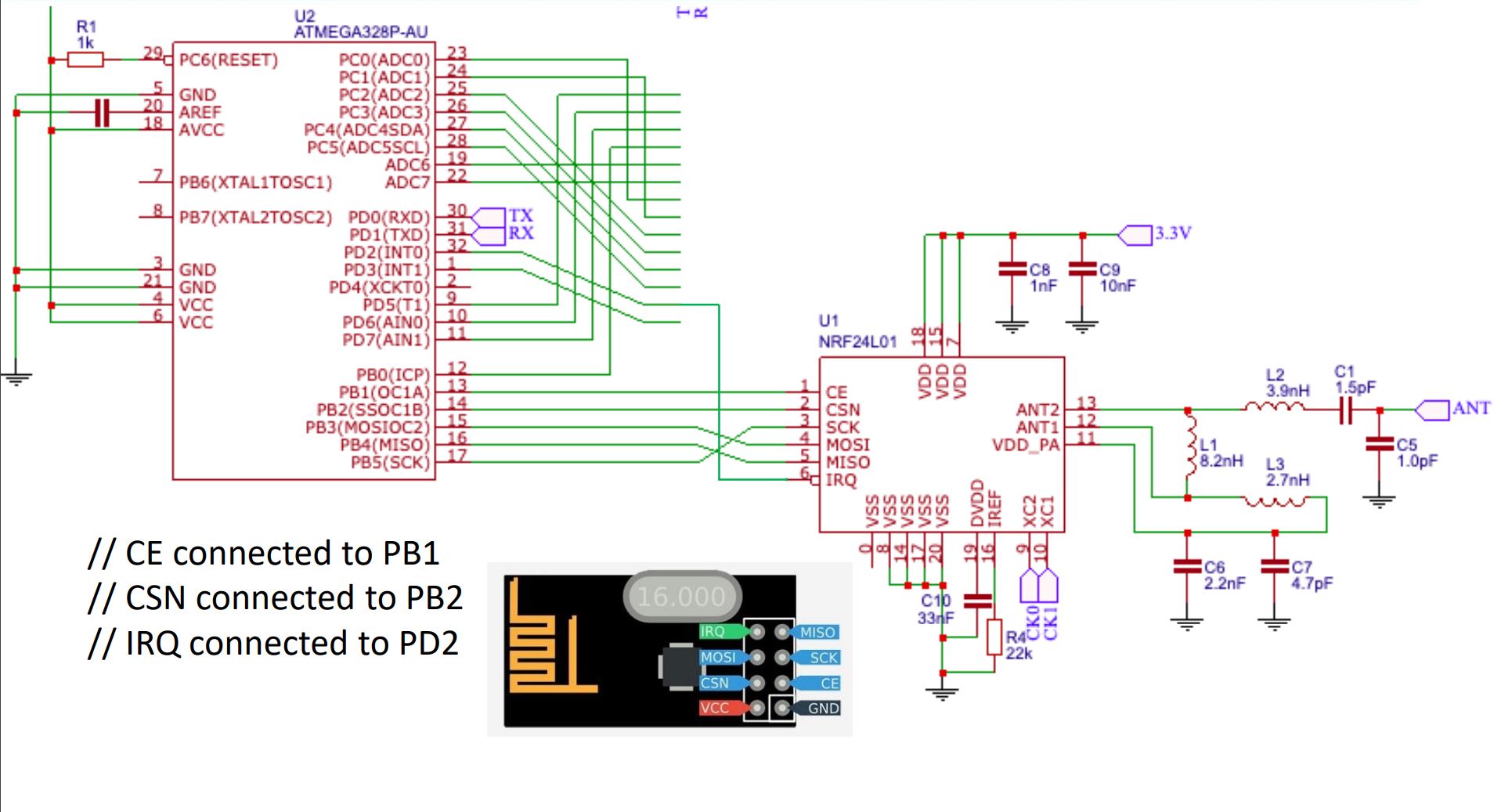
Primary Github address: <https://github.com/buchaa2/103EPC>

Directory:

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**



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

#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>

#ifndef BAUD

#define BAUD 9600

#endif

#include "STDIO\_UART.h"

#include "nrf24l01.h"

#include "nrf24l01-mnemonics.h"

#include "spi.h"

void print\_config(void);

volatile bool message\_received = false;

volatile bool status = false;

volatile *uint8\_t* ADCdata;

volatile unsigned char temp[10];

int main(void)

{

ADMUX |= (1 << REFS0); // use AVcc

ADMUX |= (1 << ADLAR); // Right adjust

ADCSRA = (1 << ADEN) // Enable

|(1 << ADPS1)

|(1 << ADPS0) // 128 prescaler 16Mhz

|(1 << ADATE) // ADC Auto Trigger

|(1 << ADSC); // Start ADC

// Set cliche message to send (message cannot exceed 32 characters)

char tx\_message[32]; // Define string array

unsigned char i;

char dummy[10];

uart\_init();

nrf24\_init();

print\_config();

nrf24\_start\_listening();

*strcpy*(tx\_message,"GOOD");

nrf24\_send\_message(tx\_message);

while (1)

{

ADCdata = (ADCH << 1) \* 2 + 32; // Convert Celsius to Fahrenheit

*itoa*(ADCdata, dummy, 10); //convert char to ascii

for(i = 0 ; i < 10 ; i++)

{

temp[i] = dummy[i]; //move converted ascii

}

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(temp);

if (status == true) *printf*("Message sent successfully\n");

}

}

}

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");

}

// Interrupt on IRQ pin

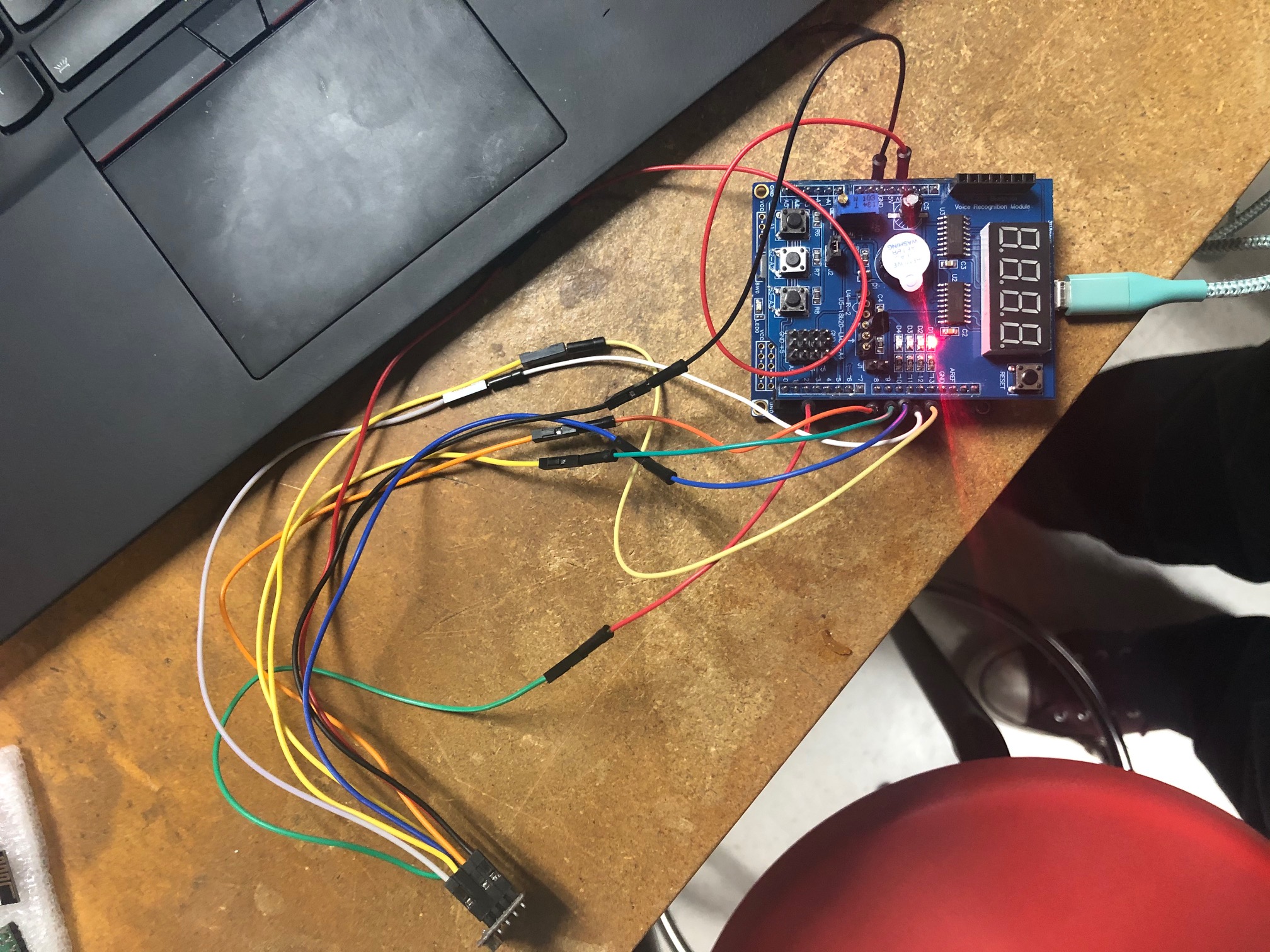
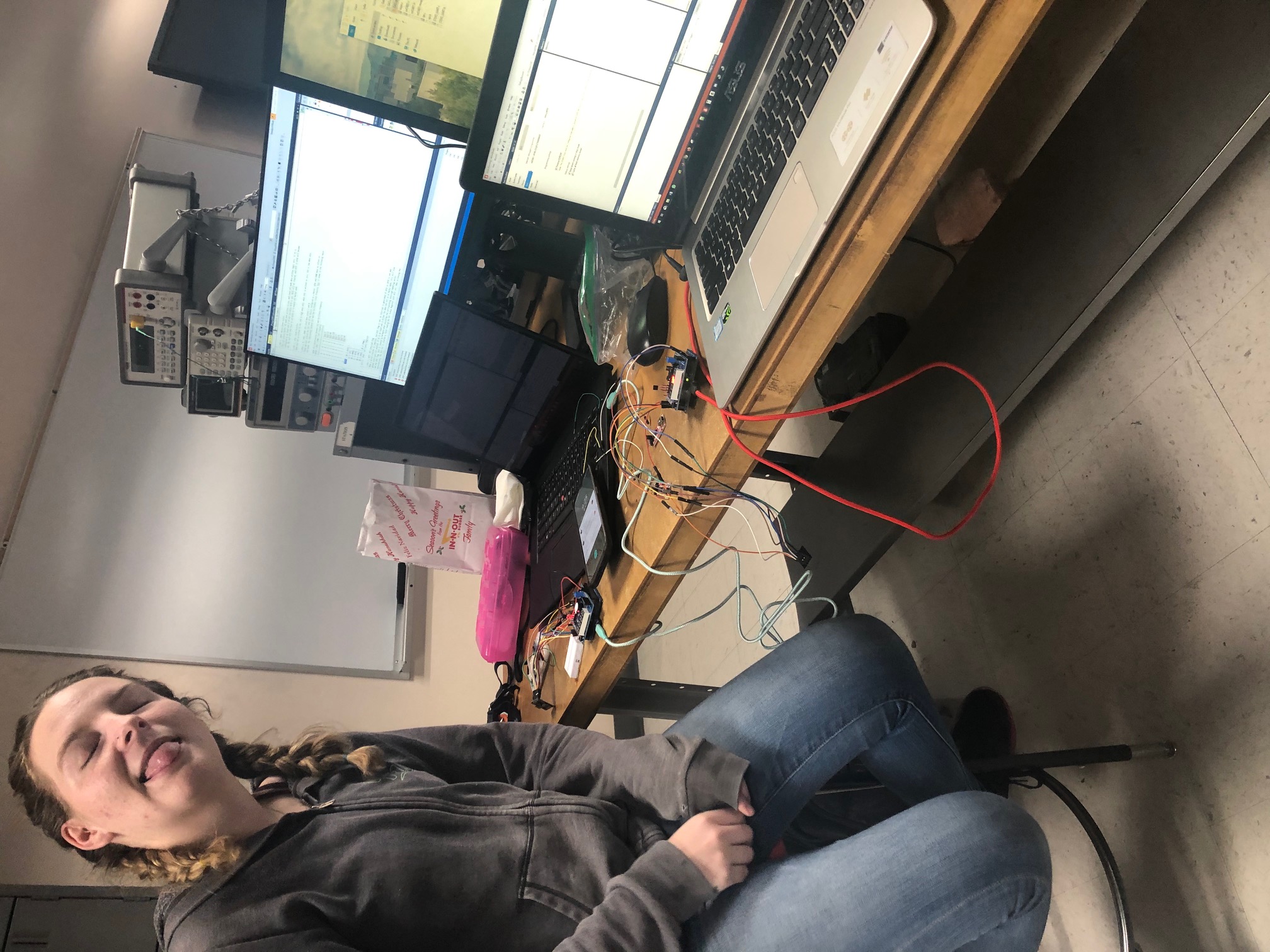
ISR(INT0\_vect)

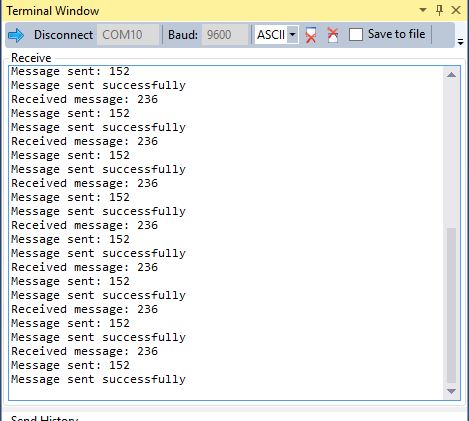
{

message\_received = true;

}

1. **SCREENSHOT OF EACH DEMO (BOARD SETUP)**





1. **VIDEO LINKS OF EACH DEMO**

Task 1

<https://www.youtube.com/watch?v=girBBwEW0g8>

1. **GITHUB LINK OF THIS DA**

**Student Academic Misconduct Policy**

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

“This assignment submission is my own, original work”.

Andrew Buchanan