CPE301 – SPRING 2019

Design Assignment DA5

Student Name: Cody McDonald

Student #: 5000382538

Student Email: mcdonc4@unlv.nevada.edu

Primary Github address: https://github.com/elev8rProcrastinator/submission\_da.git

Directory: <https://github.com/elev8rProcrastinator/submission_da/tree/master/DA5>

Partner: Christopher Barr

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).

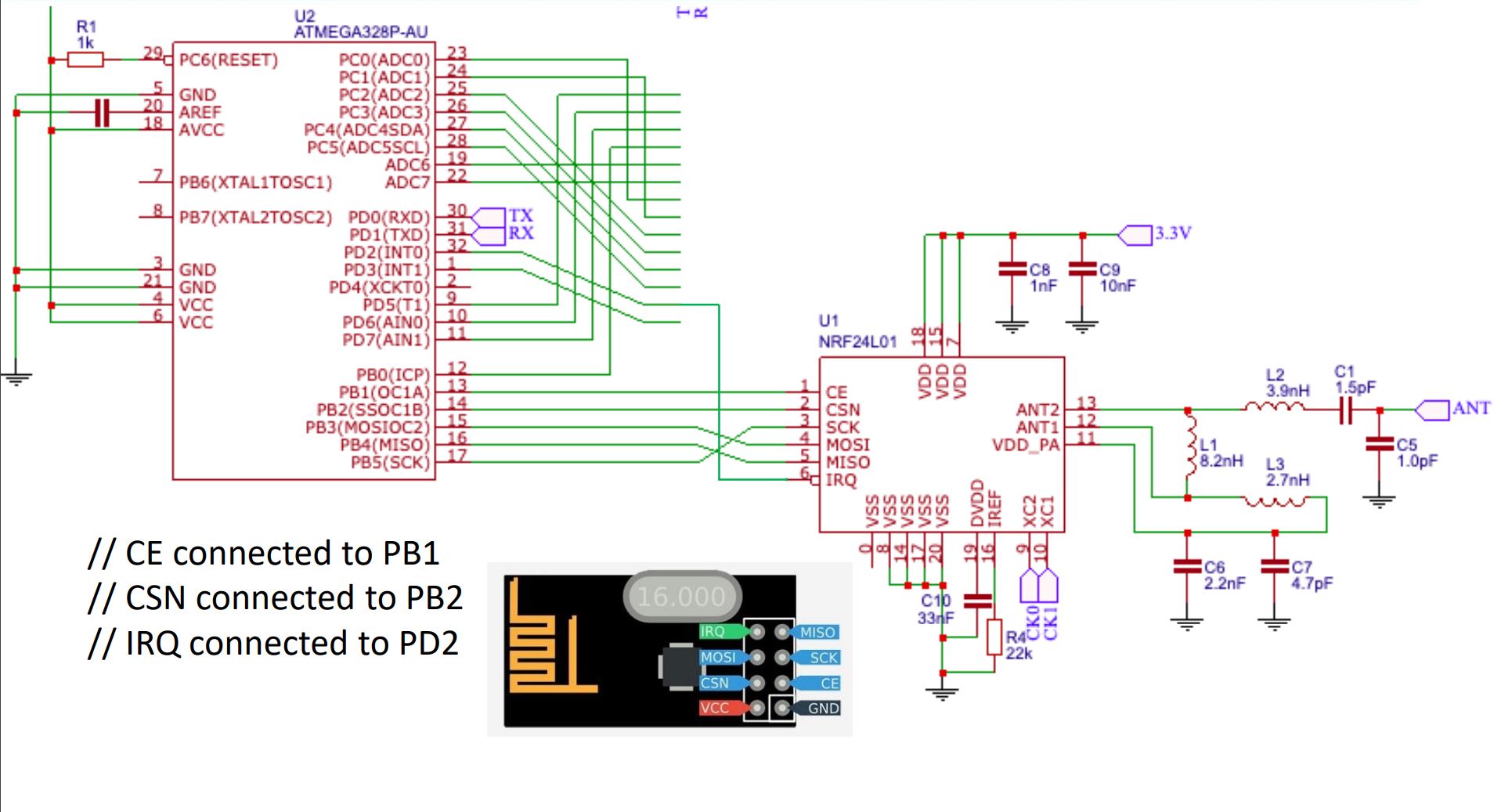
1. **COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS**

Atmini Xplained

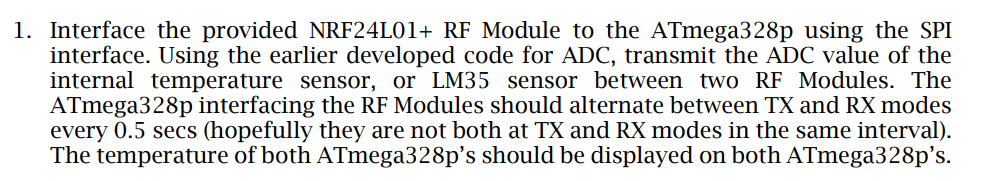
LM35

FTDI USB to serial converter

NRF241L01



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



|  |  |
| --- | --- |
|  | // 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"); |
|  | } |

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

// 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 "lib\STDIO\_UART.c"

// Include nRF24L01+ library

#include "lib\nrf24l01.c"

#include "lib\nrf24l01-mnemonics.h"

#include "lib\spi.c"

void print\_config(void);

// Used in IRQ ISR

volatile bool message\_received = false;

volatile bool status = false;

// Global variables

volatile *uint8\_t* ADCdata;

volatile unsigned char temp[10];

int main(void)

{

// Set up ADC

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

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

ADCSRA = (1 << ADEN) // Enable

|(1 << ADPS1)

|(1 << ADPS0) // 128 prescaler for 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];

// Initialize UART

uart\_init();

// Initialize nRF24L01+ and print configuration info

nrf24\_init();

print\_config();

// Start listening to incoming messages

nrf24\_start\_listening();

*strcpy*(tx\_message,"Chris, my bagels are wet"); // Copy string into array

nrf24\_send\_message(tx\_message);

while (1)

{

ADCdata = (ADCH << 1) \* 1.8 + 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");

}

}

}

// 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");

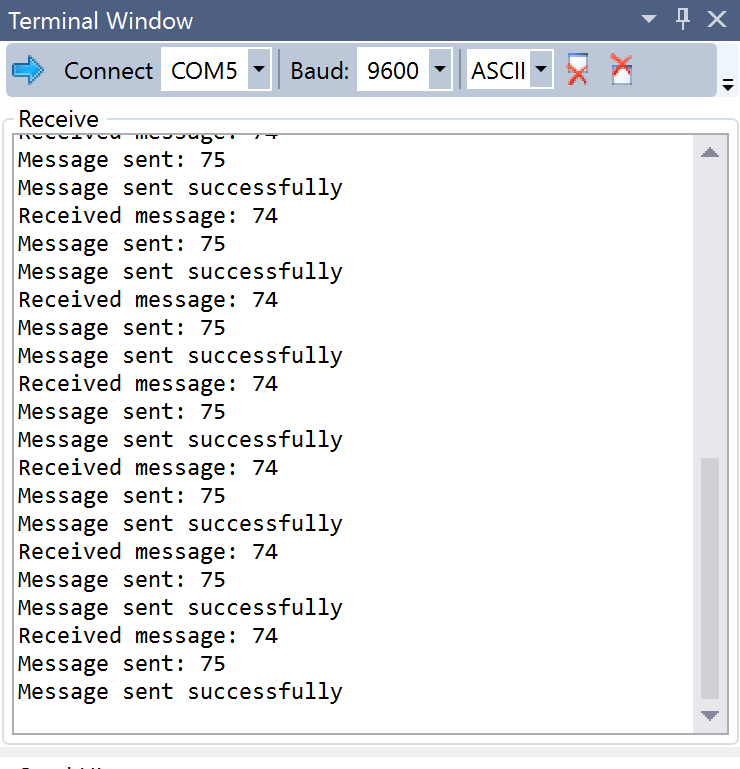
}

1. **SCHEMATICS**

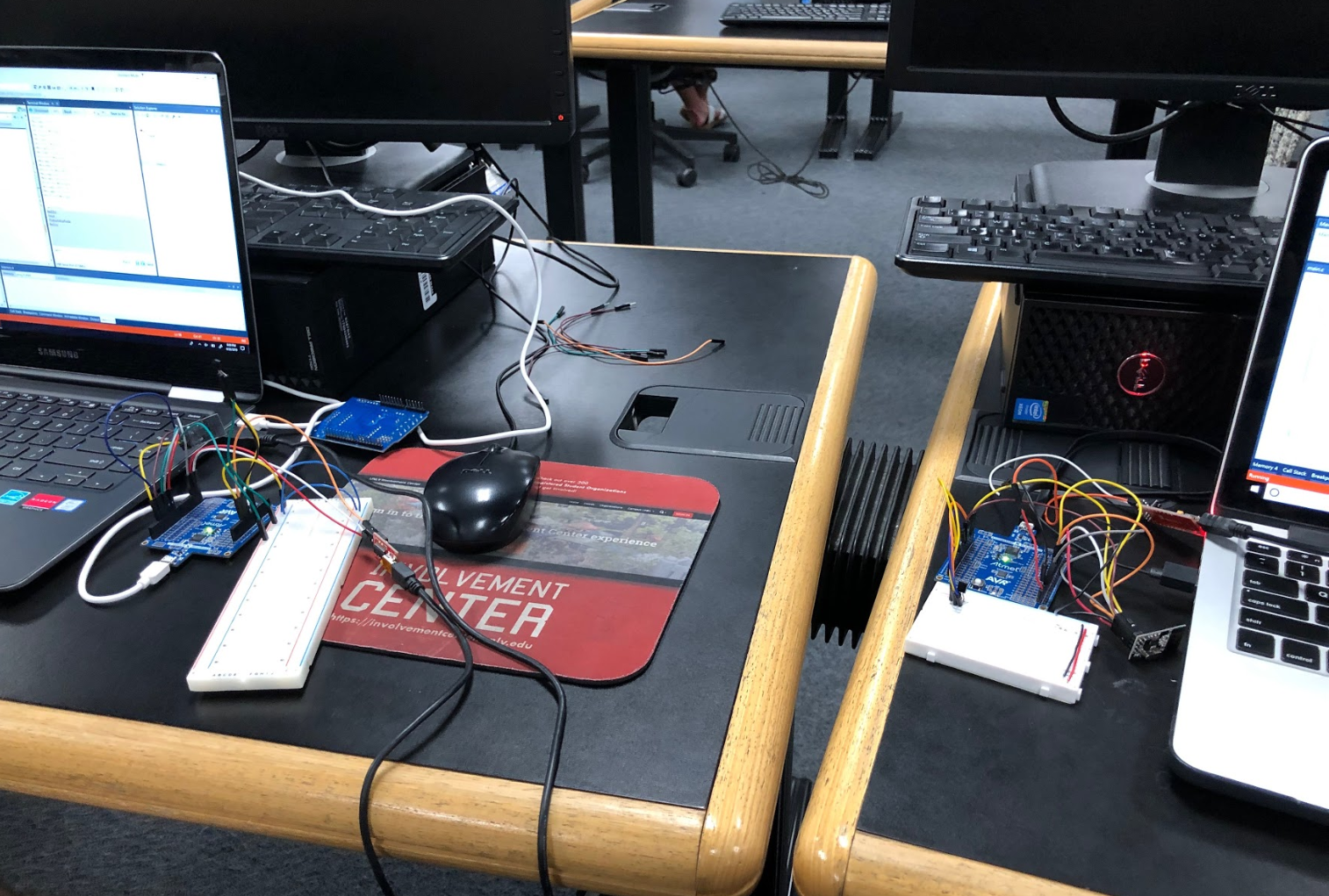
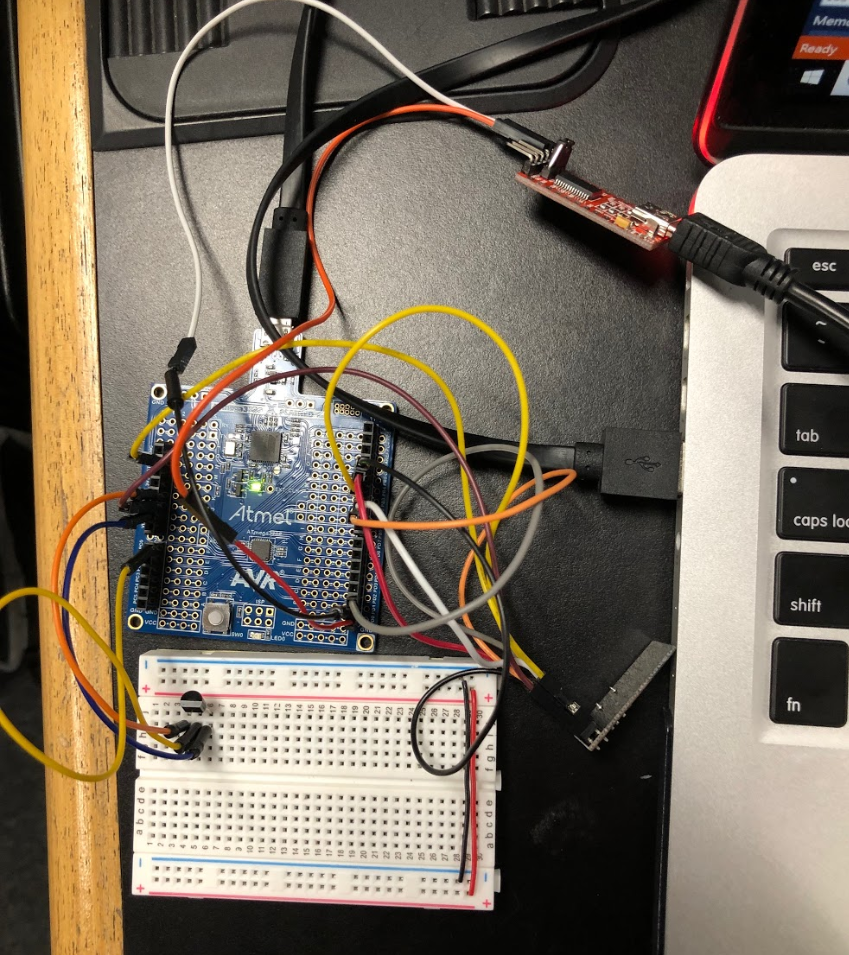
Refer to board demos for connections of the motors and drivers to the pins on the microcontroller

1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**

Terminal window of the communication between the two microcontrollers.



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



This is the set-up of my controller setup and the microcontroller I transferred data to

1. **VIDEO LINKS OF EACH DEMO**

**https://youtu.be/qpUf0vQSzPM**

1. **GITHUB LINK OF THIS DA**

https://github.com/elev8rProcrastinator/submission\_da/tree/master/DA5

**Student Academic Misconduct Policy**

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

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

Cody McDonald