CPE301 – FALL 2019

DA4A

Student Name: Andrew Buchanan

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**

AtMini xplained

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

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <avr/interrupt.h>

#include <util/delay.h>

volatile unsigned int ADC\_Val; // variable for the value of the ADC

int flag = 0; // used in the interrupt to toggle

int main(void){

DDRB |= (1<<PB1)|(1<<PB5); //PB1 and PB5 are outputs

DDRC = 0; //port c is an input

PORTB = 0; //port b is set to low

PORTC |= (1<<PC1); //pc1 is set to high

TCCR1A |= (1<<COM1A1)|(1<<COM1B1)|(1<<WGM11); // we are usingFast PWM, Non-inverted mode

TCCR1B |= (1<<WGM13)|(1<<WGM12)|(1<<CS11); // Pre-scaler is set to 8

ICR1 = 9999; // Top of timer1 set to 9999

PCICR = (1<<PCIE1); //Enable PCINT

PCMSK1 = (1<<PCINT9); //Set interrupt for PC1

ADMUX = (1<<REFS0); // the input for ADC is pc0

ADCSRA |= (1<<ADEN) | (1<<ADPS2) | (1<<ADPS1) | (1<<ADPS0); // enable ADC, system clock used for ADC

ADCSRB = 0x0; // free running mode

DIDR0 = 0x1; // Filter out any digital signals

sei(); //Enable interrupt

while (1); //wait for interrupt

}

//ISR for PC1

ISR(PCINT1\_vect){

*\_delay\_ms*(500);

if(!(PINC & (1 << PINC1))){

while(!(PINC & (1 << PINC1))){

if(flag == 1){

PORTB |= (1 << PORTB1) | (1 << PORTB5); //turn on OC1A and LED off

ADCSRA |= (1 << ADSC); // start conversion

while((ADCSRA&(1<<ADIF))==0){ // wait for conversion to finish

ADC\_Val = ADC & 0x03FF; // we only need first 10 bits

OCR1A = 10\*ADC\_Val; // Calculate duty cycle

}

}

else if(flag == 0){

OCR1A = 0; //reset

PORTB &= ~(1 << PORTB1);

PORTB &= ~(1 << PORTB5); //turn off OC1A and LED on

}

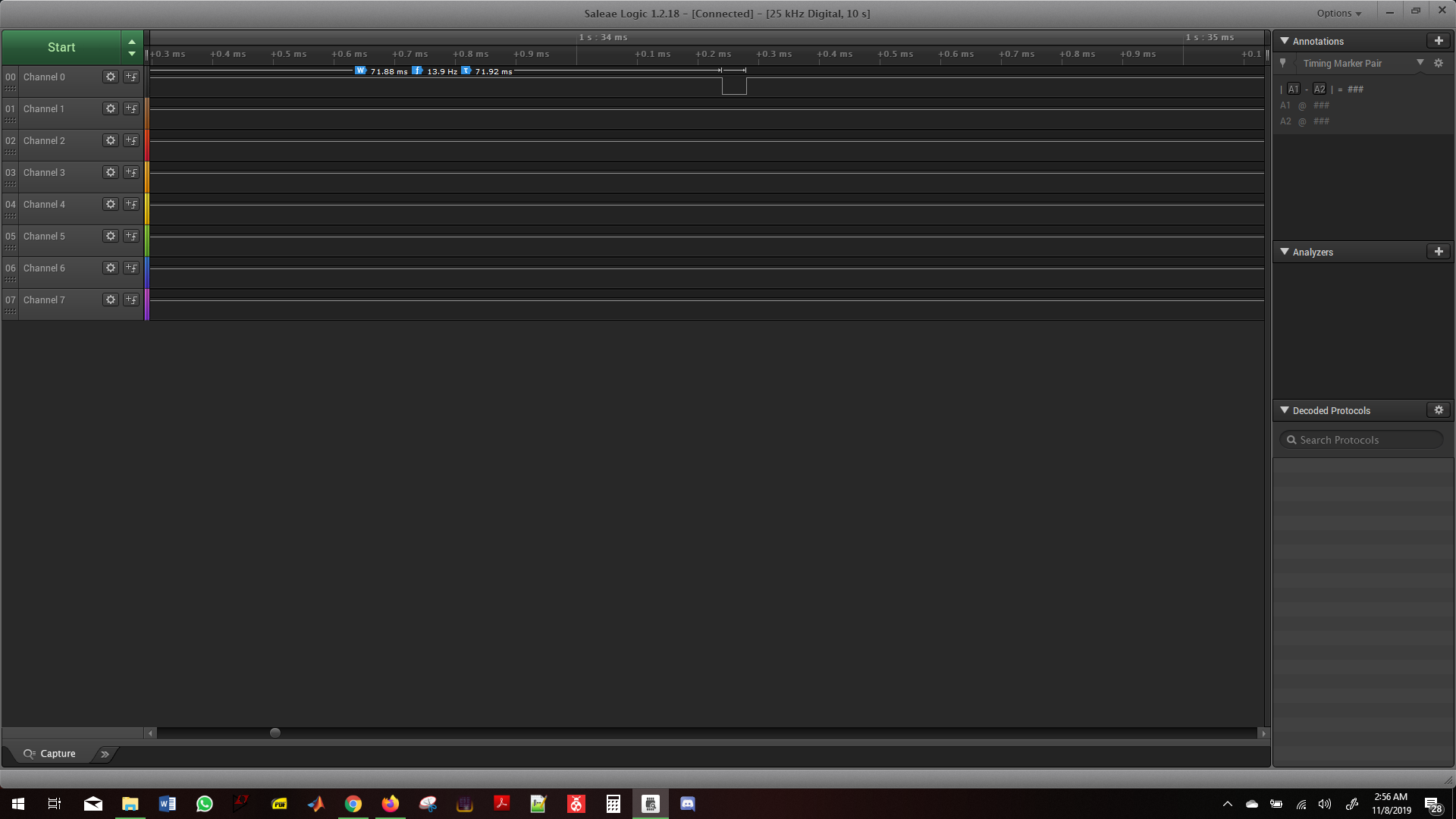
}

flag ^= 1; //toggle

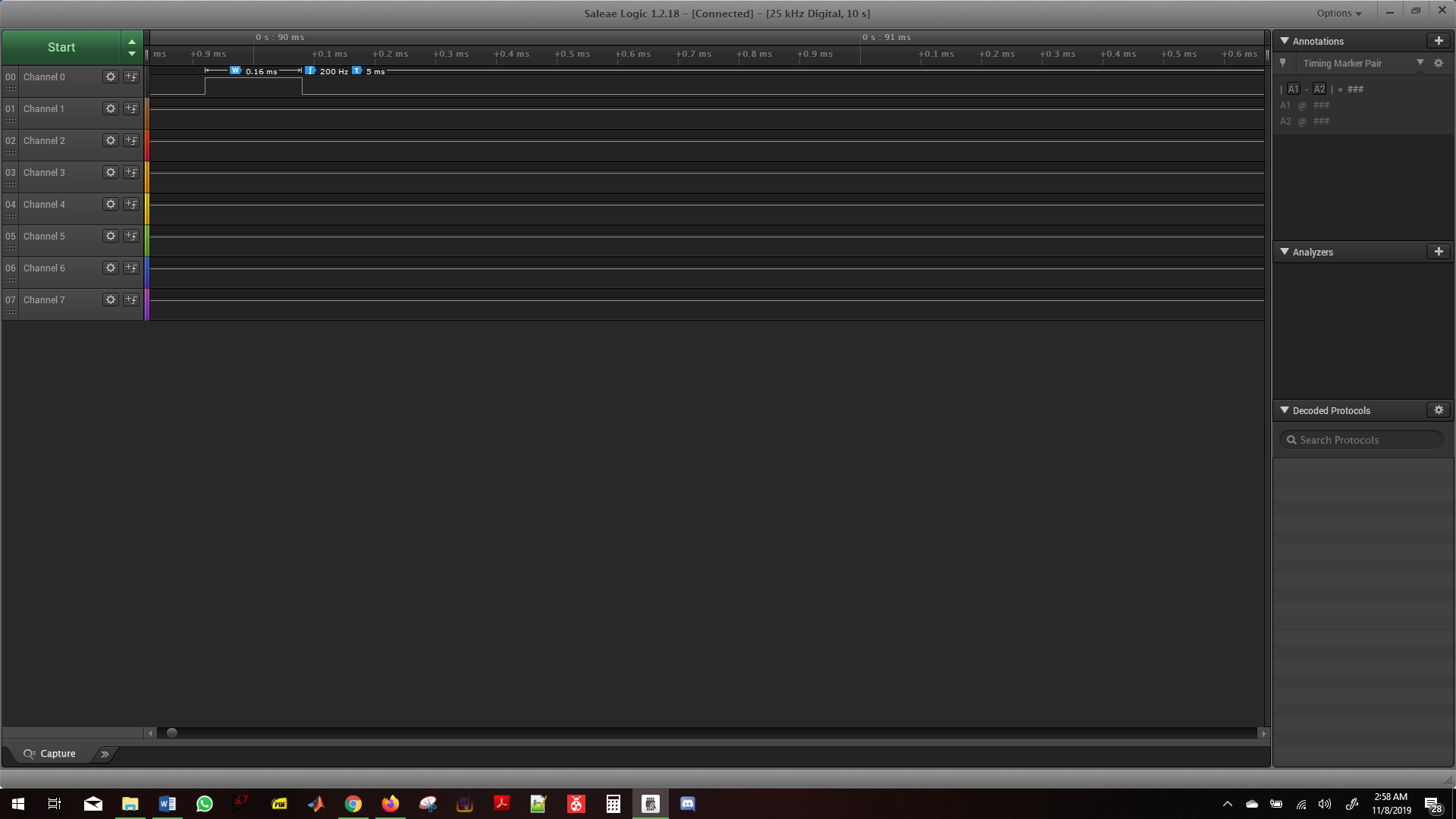
}

}

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

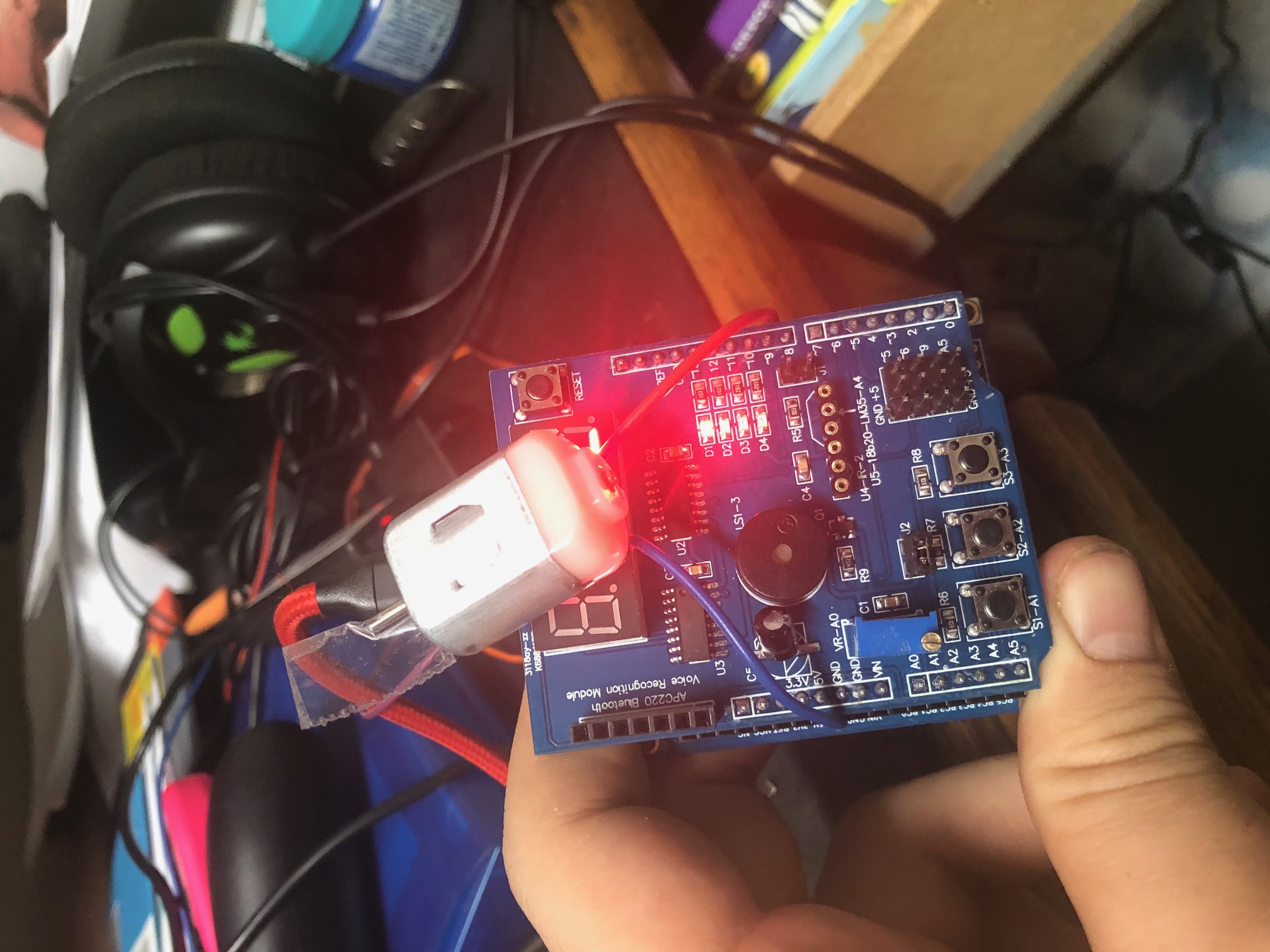


This is at full speed



This is at low speed

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



1. **VIDEO LINKS OF EACH DEMO**

https://www.youtube.com/watch?v=HxtLeAuYQJs

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