CPE301 – SPRING 2019

Design Assignment 2A

Student Name: Francisco Mata Carlos

Student #: 1012593607

Student Email: matacarl@unlv.nevada.edu

Primary Github address: https://github.com/chicosisco/da\_sub.git

Directory: repository/cpe301/DesignAssignments/DA2A

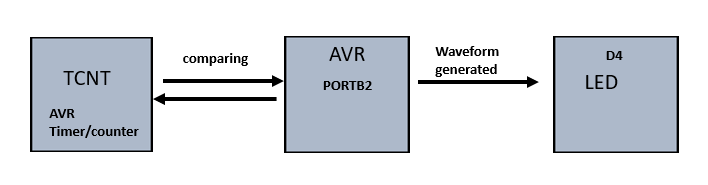
1. **COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS**

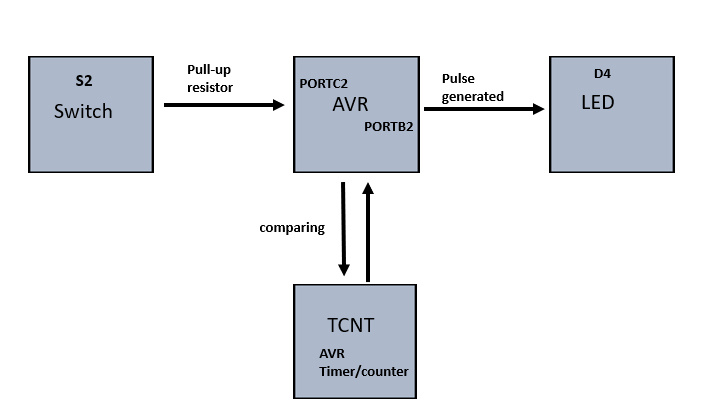
The components used for this assignment are the next:

1. Atmega328p Xplained Mini
2. Multi-functional Shield
3. Oscilloscope and compensated probe
4. Atmel Studio 7

**Block diagram with pins used in the Atmega328P**

**Block for generating a waveform with 60% duty cycle and a period of 0.725 seconds**



**Block diagram for part 2 of assignment. Generating a pulse when a pushbutton is pressed and is demonstrated by an LED.**

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

**Task 1\_A**

**1. Design a delay subroutine to generate a waveform on PORTB.2 with 60% DC and 0.725 sec period; C code.**

\*This C code generates a waveform with a period of 0.725s; 60% duty cycle. It sets the PB2 with a 1 value which causes the led to turn off.

\* This is due to the reason that VDD is connected to the LED and then a resistor, which is connected to pin B2. Thus, when PB2 is high the

\* voltage drop across the resistor and LED is minimum, and the LED turns off. But if PB2 is low then there's a voltage drop across the LED

\* resistor which causes to turn it on. To make a 60% duty cycle, a 435ms on time is given to the PB2 and then an off time of 290ms.

\*

\* DA2A.c

\*

\* Created: 3/2/2019 1:15:16 PM

\* Author : Francisco Mata Carlos

\*/

#define *F\_CPU* 16000000UL /\* clock runs at 16 MHz\*/

#include <avr/io.h>

#include <util/delay.h>

int main(void)

{

DDRB = 0x04; /\* sets PORTB as output \*/

while(1)

{

PORTB = (1 << 2); /\* sets PB2 high \*/

*\_delay\_ms*(435); /\* time that the led is (60% DC with 0.725s period) \*/

PORTB = ~(1 << 2); /\* displays output \*/

*\_delay\_ms*(290); /\* time that the led is on (40% DC with 0.725s period) \*/

}

return 0;

}

**Task 1\_B**

**1. Design a delay subroutine to generate a waveform on PORTB.2 with 60% DC and 0.725 sec period; in Assembly**.

; This assembly code designs a delay subroutine to generate a waveform on PORTB.2 with 60% DC and 0.725sec period.

; This code takes advantage of the Timer Counter (TCNT) and the Timer Control Register (TCCR) in order preset a value,

; which will be used to compared with the registers and toggle a value or bit.

; DA2A\_AS1.asm

;

; Created: 3/2/2019 1:17:44 PM

; Author : Francisco Mata Carlos

.org 0

LDI R16,4 ;representing PB2

OUT DDRB, R16 ;enable PB2 as output

LDI R17,4 ;used to set or reset PB2

LDI R20,5 ;set clock prescaler to 1024

STS TCCR1B,R20

begin:

LDI R20,0x00 ;resetting the counter to 0

STS TCNT1H,R20

STS TCNT1L,R20

RCALL delay ;calling timer to wait for 435ms

EOR R17,R16 ;XOR to toggle led

OUT PORTB,R17

LDI R20,0x00 ;resetting the counter to 0

STS TCNT1H,R20

STS TCNT1L,R20

RCALL delay2 ;calling timer to wait for 290ms

EOR R17,R16 ;XOR to toggle led

OUT PORTB,R17

RJMP begin ;repeat main loop

delay:

LDS R29, TCNT1H ;loading upper bit of counter to R29

LDS R28, TCNT1L ;loading lower bit of counter to R28

CPI R28,0x8C ;comparing if lower 8 bits of timer is 0x8C

BRSH body ;if lower bits of timer have reached desired amount, check the upper bits

RJMP delay ;otherwise, keep checking lower bits

body:

CPI R29,0x1A ;check to see if upper timer bits have reached the desired value

BRLT delay ;if not, recheck the lower bits

RET ;once the timer reached the desired value, toggle the LED

delay2:

LDS R29, TCNT1H ;loading upper bit of counter to R29

LDS R28, TCNT1L ;loading lower bit of counter to R28

CPI R28,0xB2 ;comparing if lower 8 bits of timer is 0xB2

BRSH body2 ;if lower bits of timer have reached desired amount, check the upper bits

RJMP delay2 ;otherwise, keep checking lower bits

body2:

CPI R29,0x11 ;check to see if upper timer bits have reached the desired value

BRLT delay2 ;if not, recheck the lower bits

RET ;once the timer reached the desired value, toggle the LED

**Task 2\_A**

**2. Connect a switch to PORTC.2 (active high - turn on the pull up transistor) to poll for an event to turn on the led at PORTB.2 for 1.250 sec after the event; C code.**

/\*

\* This C code generates a 0.725 pulse when a push down switch is pressed. The switch (S2) is

\* pressed it creates a pulse that is demonstrated by using an LED. The LED turns on for

\* about 0.725 seconds, which is the requirement for this assignment.

\*

\*

\* AD2A\_C2.c

\*

\* Created: 3/2/2019 1:24:19 PM

\* Author : Francisco Mata Carlos

\*/

#define *F\_CPU* 16000000UL /\* clock runs at 16 MHz\*/

#include <avr/io.h>

#include <util/delay.h>

int main(void)

{

/\* set PORTB2 for output \*/

DDRB |= (1<<2);

PORTB |=(1<<2); // led off

/\* set PORTC2 for input \*/

DDRC &= (0<<2);

PORTC |=(1<<2); //enable pull-up

while (1) {

if (!(PINC & (1<<PINC1))) //checking if pinc is high and complement

{

PORTB &= ~(1<<2); // turn led on because sending a zero to the connection

*\_delay\_ms*(1250); // 1.25s delay

}

else

PORTB |= (1<<2); // keep PORTB2 high

}

return 0;

}

**Task 2\_B**

**2. Connect a switch to PORTC.2 (active high - turn on the pull up transistor) to poll for an event to turn on the led at PORTB.2 for 1.250 sec after the event**.

; This assembly code connects a switch 2 (SW2) to PORTC2, and connects PORTB2 to an LED. When the switch closes,

; PORC2 gets pull down and sees a zero value, which triggers PB2 to go high and turn off the LED for a about 1.25s.

; This switch is control by a push down switch, so when the user press down on the swithc the LED turns on.

;

; DA2A\_part\_2.asm

;

; Created: 3/2/2019 3:59:44 AM

; Author : Francisco Mata Carlos

;

.org 0

.include <m328pdef.inc>

CBI DDRC, 2 ;make PC2 an input

SBI DDRB, 2 ;make PB2 an output

AGAIN: SBIS PINC, 2 ;skip next if PC2 is high

RJMP OVER ; jump to lable OVER

SBI PORTB, 2

RJMP AGAIN ; jump to label AGAIN

OVER: CBI PORTB, 2 ; make port B2 high

LDI R20,5 ;set clock prescaler to 1024

STS TCCR1B,R20

begin:

LDI R20,0x00 ;resetting the counter to 0

STS TCNT1H,R20

STS TCNT1L,R20

delay:

LDS R29, TCNT1H ;loading upper bit of counter to R29

LDS R28, TCNT1L ;loading lower bit of counter to R28

CPI R28,0x4A ;comparing if lower 8 bits of timer is 0x4A

BRSH body ;if lower bits of timer have reached desired amount, check the upper bits

RJMP delay ;otherwise, keep checking lower bits

body:

CPI R29,0x4C ;check to see if upper timer bits have reached the desired value

BRLT delay ;if not, recheck the lower bits

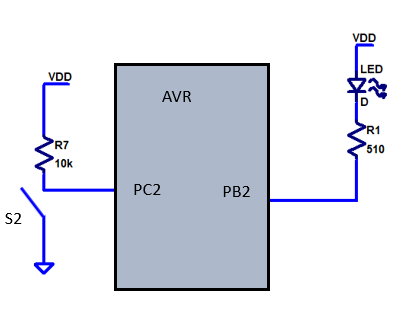
RET ;once the timer reached the desired value, toggle the LED

RJMP AGAIN ; jump to label AGAIN

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

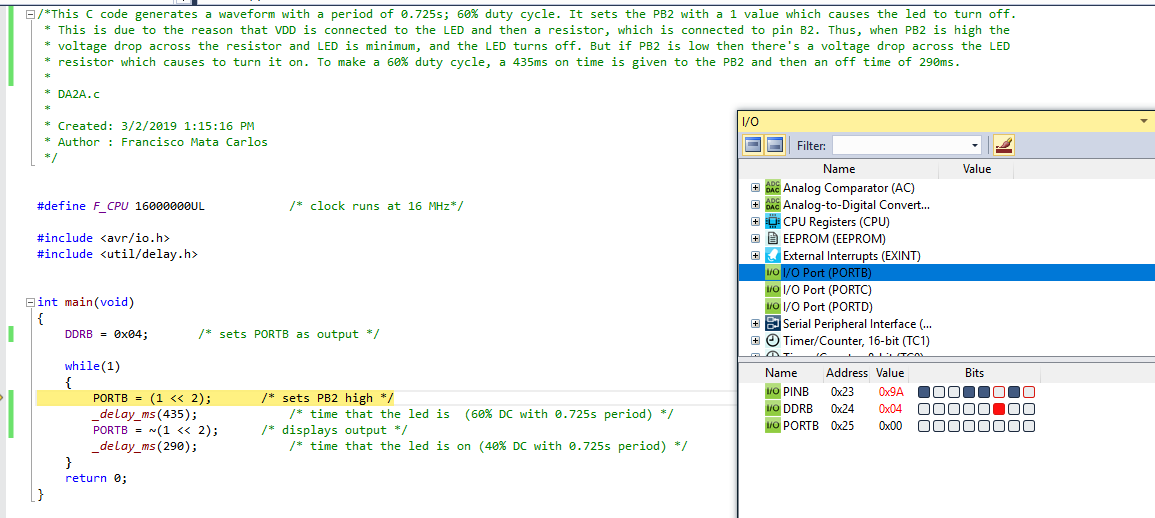
Same as above

1. **SCHEMATICS**

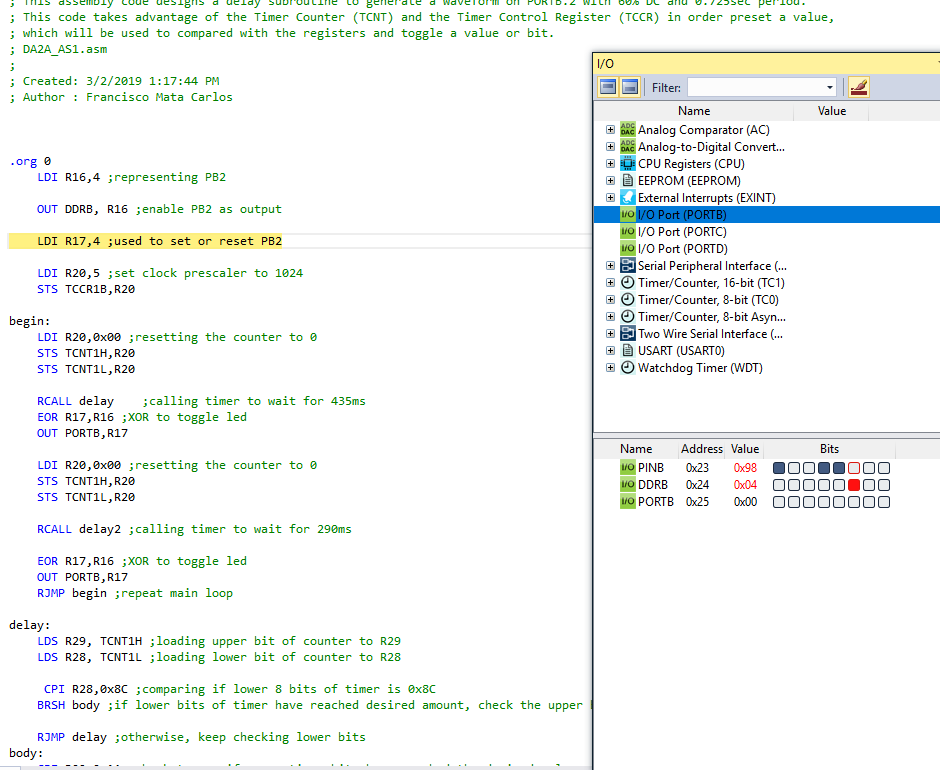


1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**

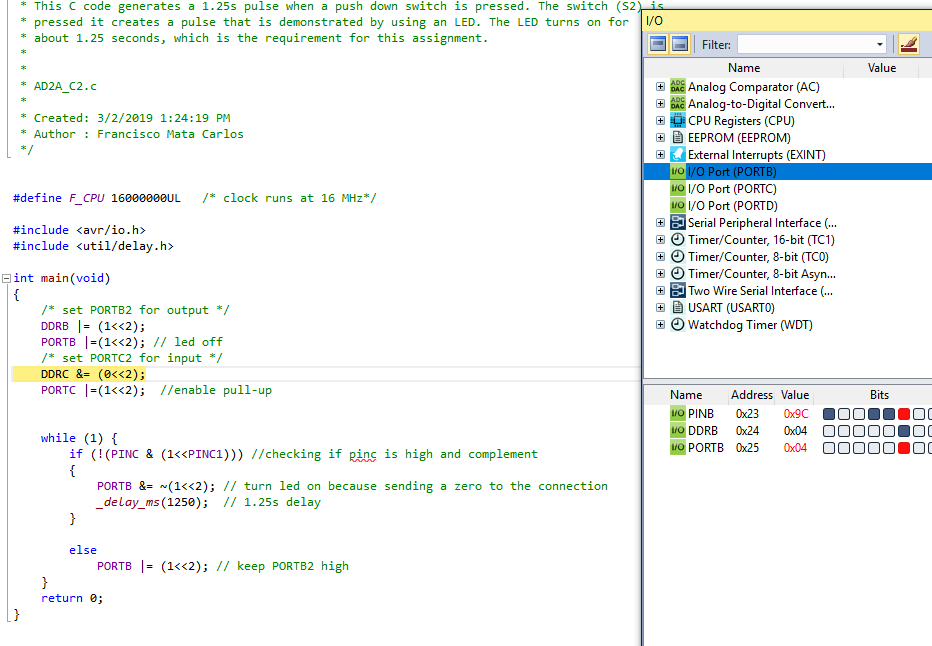
**Task 1\_A C code**



**Task 1\_B Assembly**



**Task 2\_A, C code**

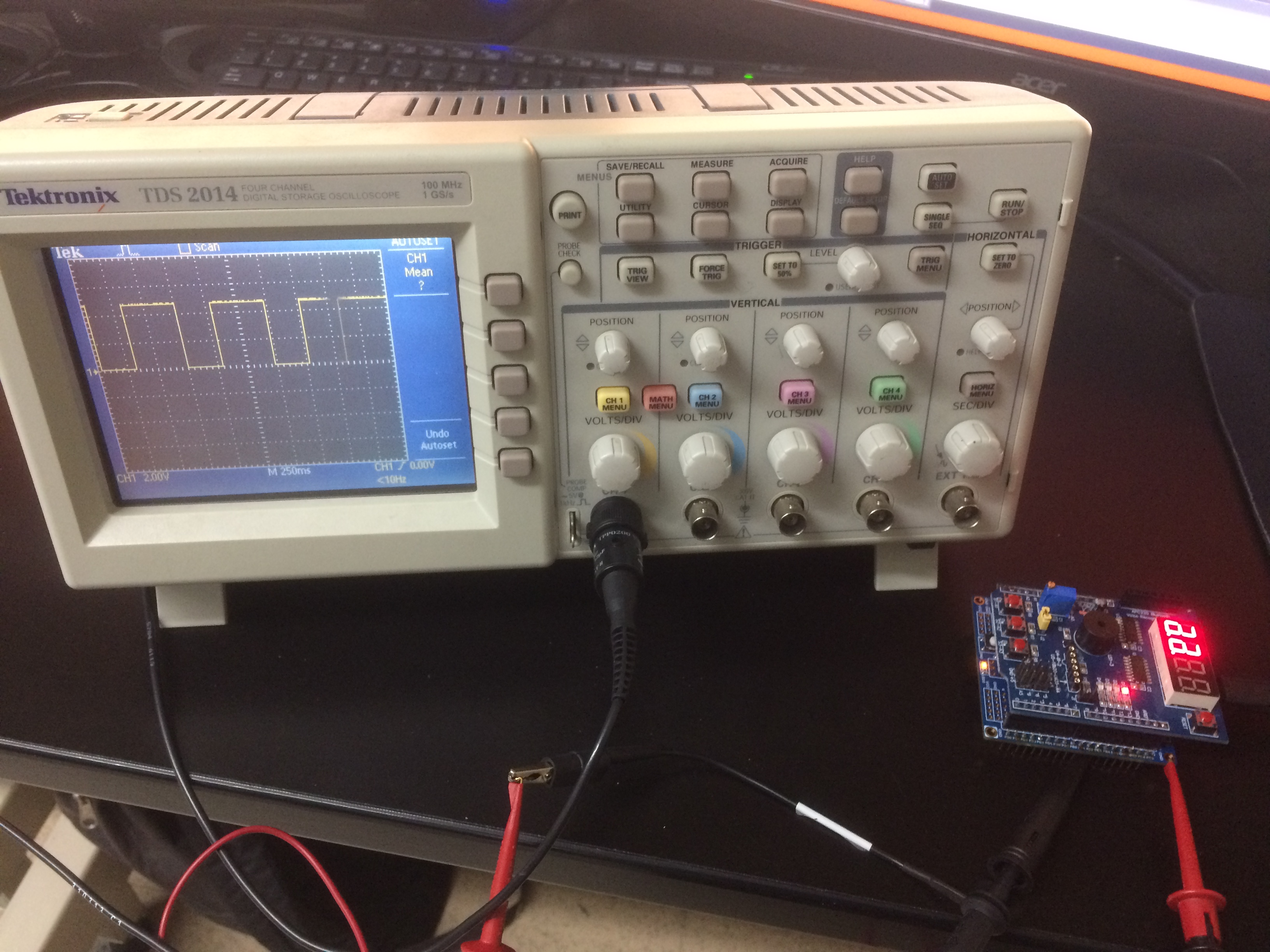


**Task 2\_B, Assembly**

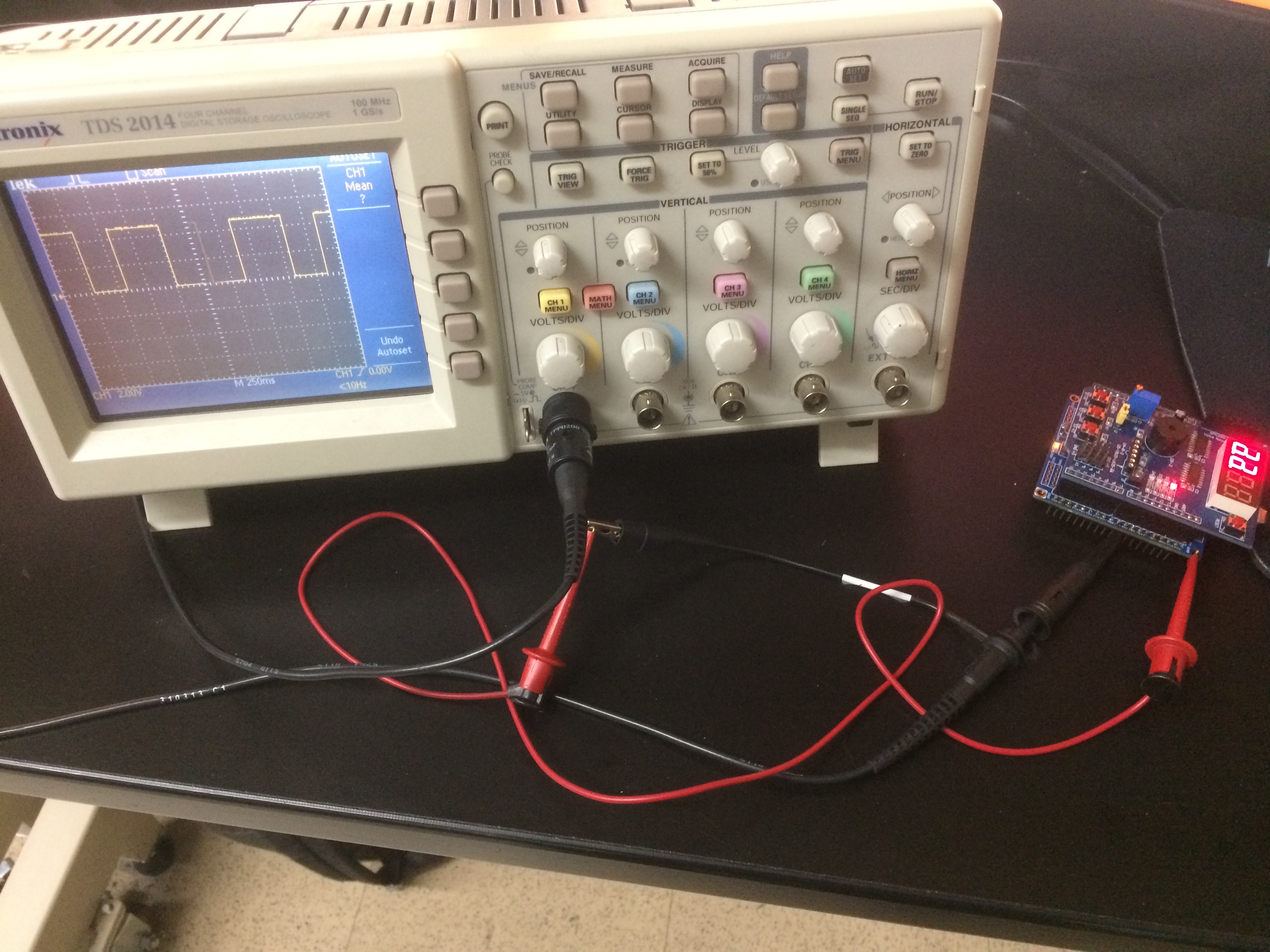


1. **SCREENSHOT OF EACH DEMO (BOARD SETUP)**

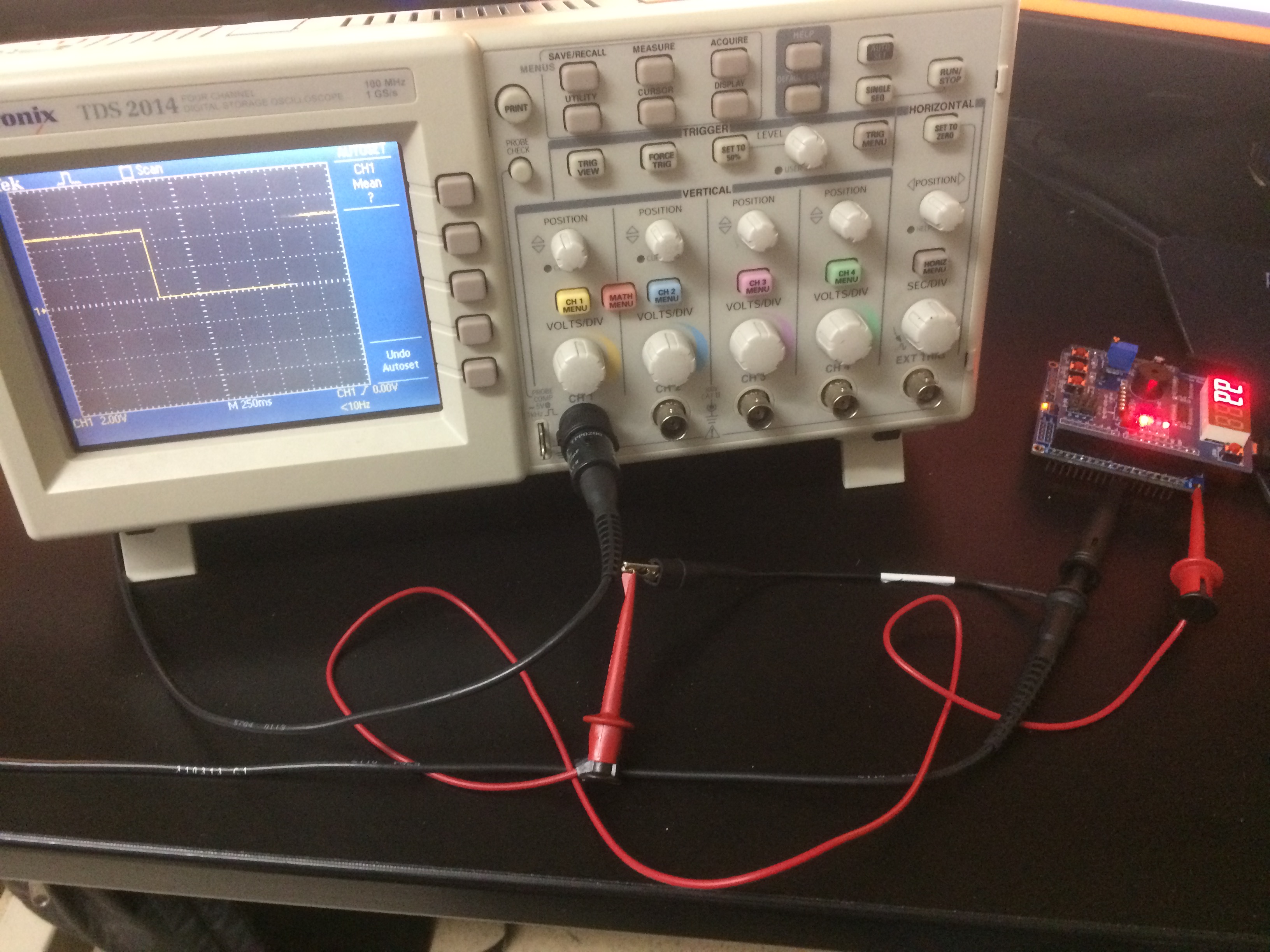
**Task 1\_A C code**



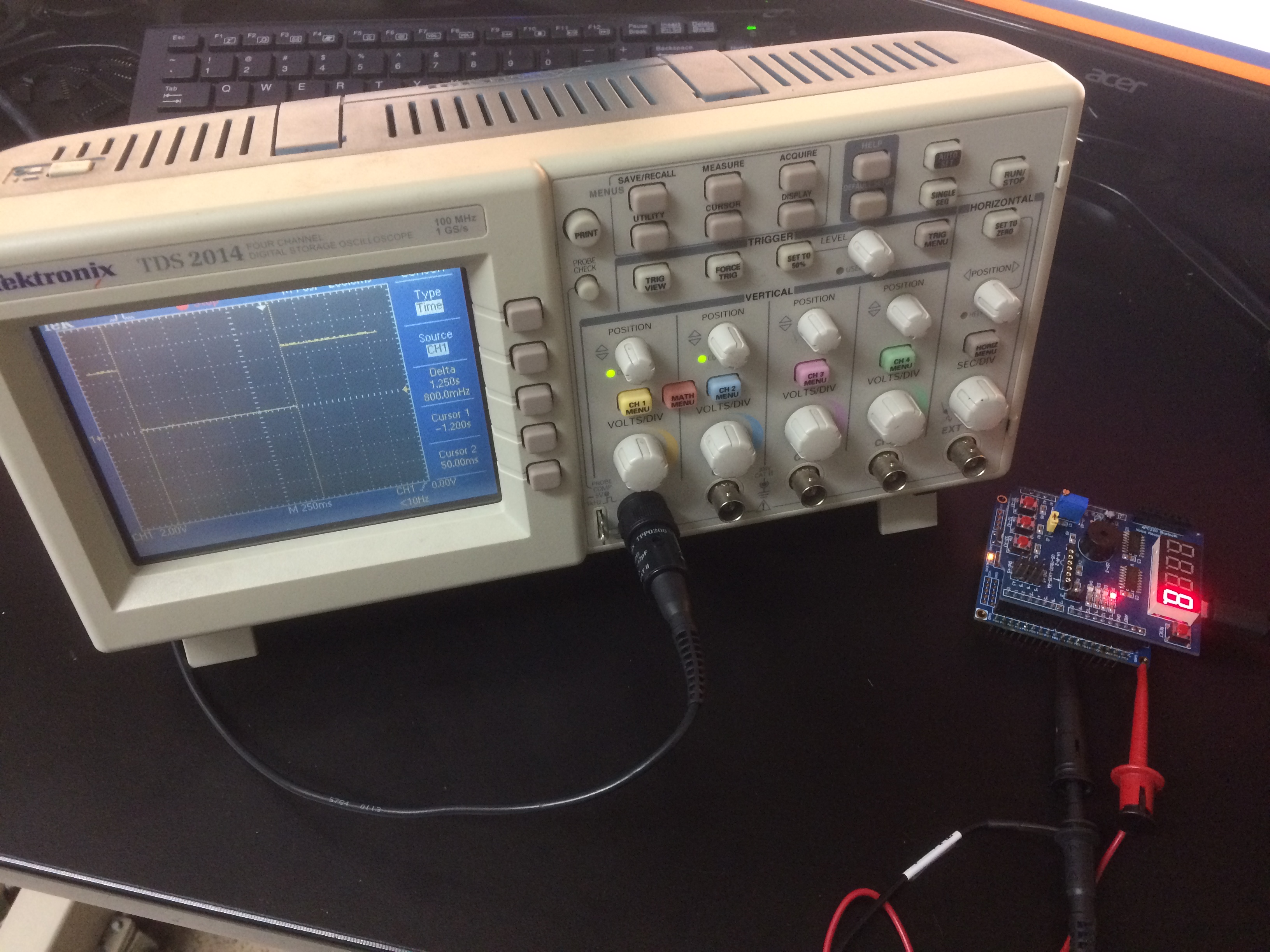
**Task 1\_B, Assembly**



**Task 2\_A C code**



**Task 2\_B, Assembly**



1. **VIDEO LINKS OF EACH DEMO**

DA2A part 1, C code, waveform

<https://youtu.be/qvfmL4q4bm8>

DA2A part 2, Assembly waveform

<https://youtu.be/xaFxvEmrp9g>

DA2A part 3, C code, LED pulse

<https://youtu.be/1XfX2OyALfo>

DA2A part 4, Assembly, LED pulse

<https://youtu.be/SNEZ-ofCva0>

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”.

Francisco Mata Carlos