CPE301 – SPRING 2019

Design Assignment 2 A

Student Name: Brian West

Student #: 5003032874

Student Email: westb2@unlv.nevada.edu

Primary Github address: https://github.com/westbrian2/Spring2019

Directory: Spring2019/DesignAssignments/DA2A\_submission

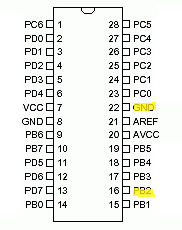
Submit the following for all Labs:

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**

COMPONENTS: Xplained Mini board, two wires, Logic Analyzer

BLOCK DIAGRAM WITH PINS USED:



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

No initial code given

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

**TASK1:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ASM CODE\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

.org 0

;use PORTB2 for wave output pin

waveGen:

ldi r20,4

out ddrb, r20 ; setting PB2 as output

ldi r21, 0; used to turn pin off

ldi r22, 5 ; Prescaler= 1024

sts tccr1b, r22

waveBegin: ;Beginning the wave generation

sts tcnt1h, r21; setting counter to 0

sts tcnt1l,r21

out portb,r20; Toggles on

rcall delayOn ; Creating on of 60% or .435sec

sts tcnt1h,r23; reset counter

sts tcnt1l,r23

out portb,r21 ; turns pin off

rcall delayOff; creating off of 40% (.29sec)

jmp waveBegin; creating next period

delayOn: //60% on

lds r25,tcnt1h; loading upper bits for comparison

lds r26,tcnt1l

cpi r26, 0x8C; checking lower bits

brsh upperOn

rjmp delayOn ; loops back to keep checking

upperOn:

cpi r25,0x1A ; checking upper bits

brlt delayOn

ret

delayOff: //40% off

lds r25,tcnt1h

lds r26,tcnt1l

cpi r26,0xB3 ; checking lower bits

brsh upperOff

rjmp delayOff

upperOff:

cpi r25,0x11; checking upper bits

brlt delayOff

ret

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*C CODE\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#define F\_CPU 16000000UL

#include <avr/io.h>

#include <util/delay.h>

int main(void)

{

DDRB=0x04; //setting PB2 as an output

while (1)

{

PORTB=0x04; //setting signal high

\_delay\_ms(435); //60% on

PORTB=0x00; //setting low

\_delay\_ms(290); //40% off

}

}

**TASK2:**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ASM CODE\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**.org 0**

**cbi ddrc,2; make c2 an input**

**sbi portc,2; pull up**

**ldi r16,0x00;**

**out ddrb, r16; disable output**

**sbi ddrb,2; sets pb2 as out**

**ldi r20,5 ; prescaler 1024**

**sts tccr1b,r20;**

**ldi r20,0**

**delayReset:**

**sbi portb,2 ; reset led**

**sts tcnt1h,r20; counter = 0**

**sts tcnt1l,r20;**

**button:**

**sbic pinc,2 ; checking if C2 is 0**

**jmp button ;continures to check if c2 is 1**

**cbi portb,2 ; lights led**

**jmp delay; sets delay**

**delay:**

**lds r21,tcnt1h; getting upper bits from counter**

**lds r22,tcnt1l; getting lower**

**cpi r22,0x4B; comparing against target value**

**brsh upper**

**jmp delay ; if target value isn't met restart loop**

**upper:**

**cpi r21,0x4C**

**brlt delay**

**end:**

**sbi portb,2; turn back off**

**jmp end**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*C CODE\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

#define F\_CPU 16000000UL

#include <avr/io.h>

#include <util/delay.h>

int main(void){

DDRB=0x04; //setting pb2 as output

PORTC=0x02; //setting pull up

DDRC=0x00; //setting input

while(1){

PORTB=0xff; //setting LED off

if(!(PINC&(1<<2))){

PORTB=0xFB; //lighting LED

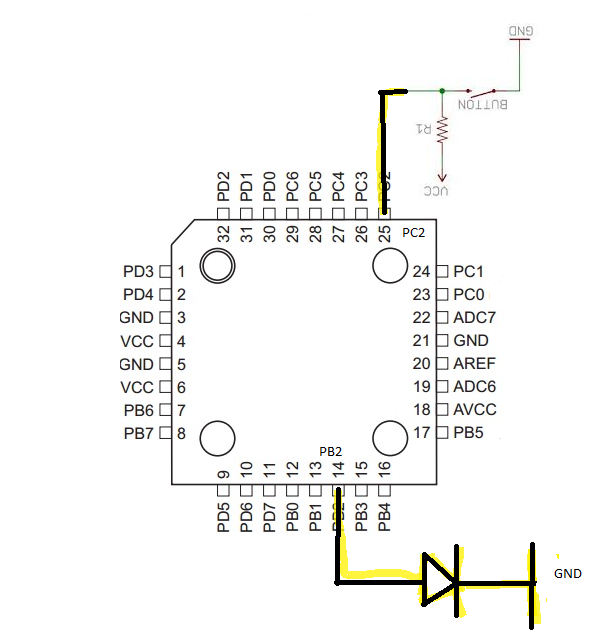
\_delay\_ms(1250);

}

}

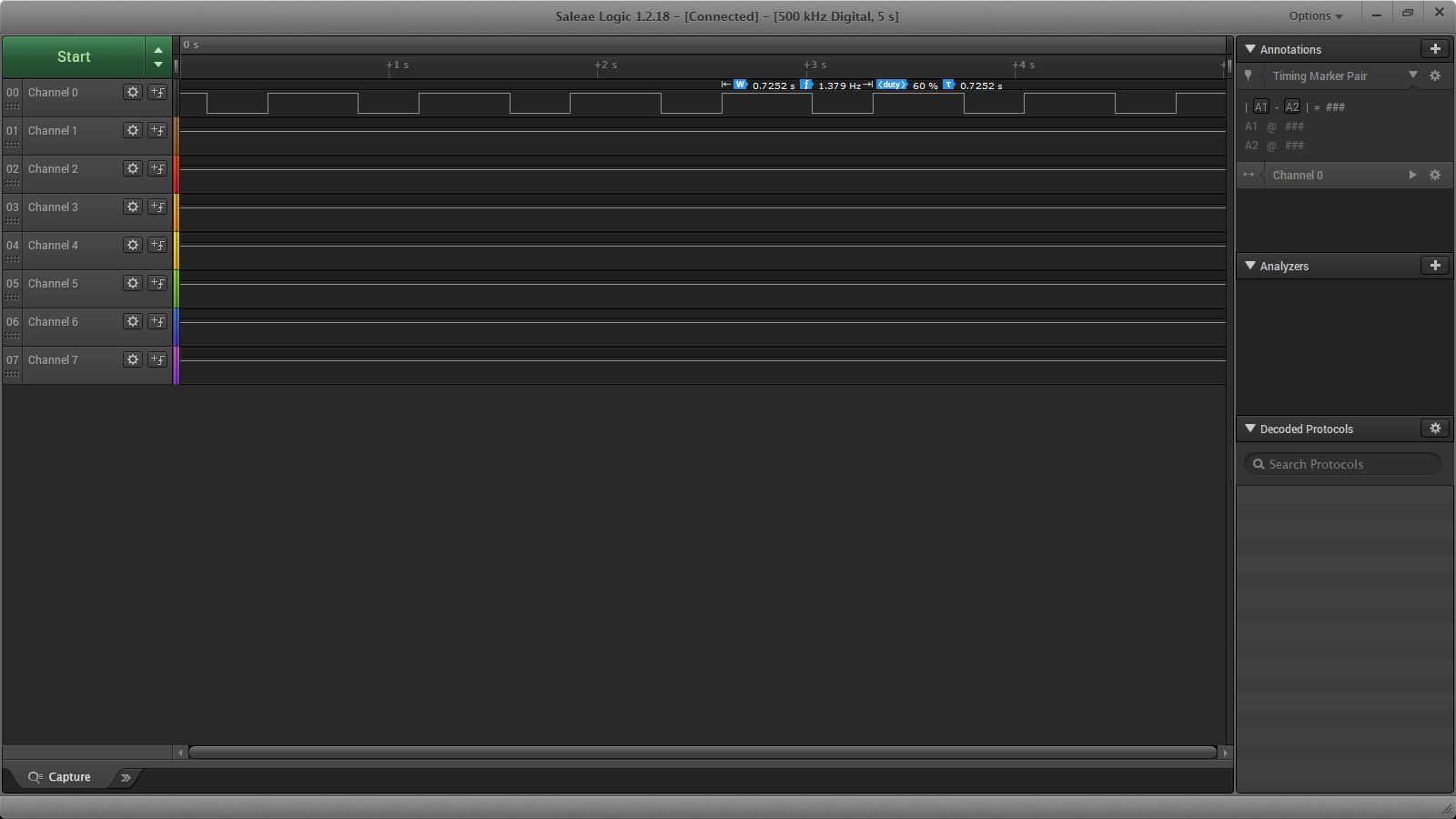
}

1. **SCHEMATICS**

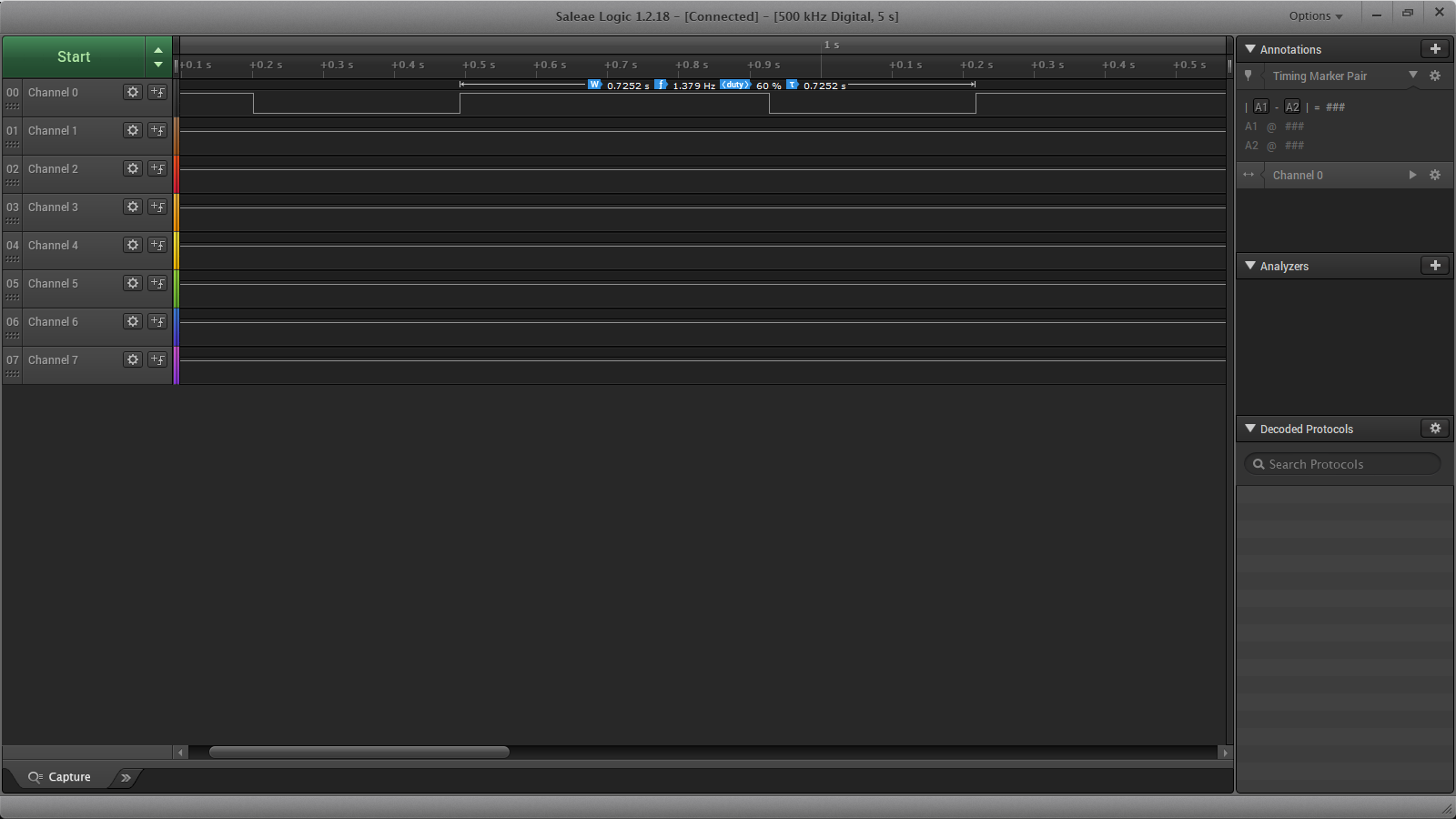


1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**

SCREENSHOTS FROM LOGIC



Period= .725s with a duty cycle of 60% (ASM code)



Period=.725s with a duty cycle of 60% (C code)

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



The board was setup with the multifunction shield.

1. **VIDEO LINKS OF EACH DEMO**

Task 2 Demo: https://youtu.be/ydF8utaQK1c

1. **GITHUB LINK OF THIS DA**

https://github.com/westbrian2/Spring2019/tree/master/DesignAssignments/DA2A\_submission

**Student Academic Misconduct Policy**

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

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

Brian West