CPE301 – SPRING 2020

Design Assignment 2C

Student Name: Xianjie Cao

Student #: 5004222179

Student Email: [caox2@unlv.nevada.edu](mailto:caox2@unlv.nevada.edu)

Primary Github address: <https://github.com/c1029324620/Mocha.git>

Directory: Mocha/DesignAssignments/LAB2/DA2C

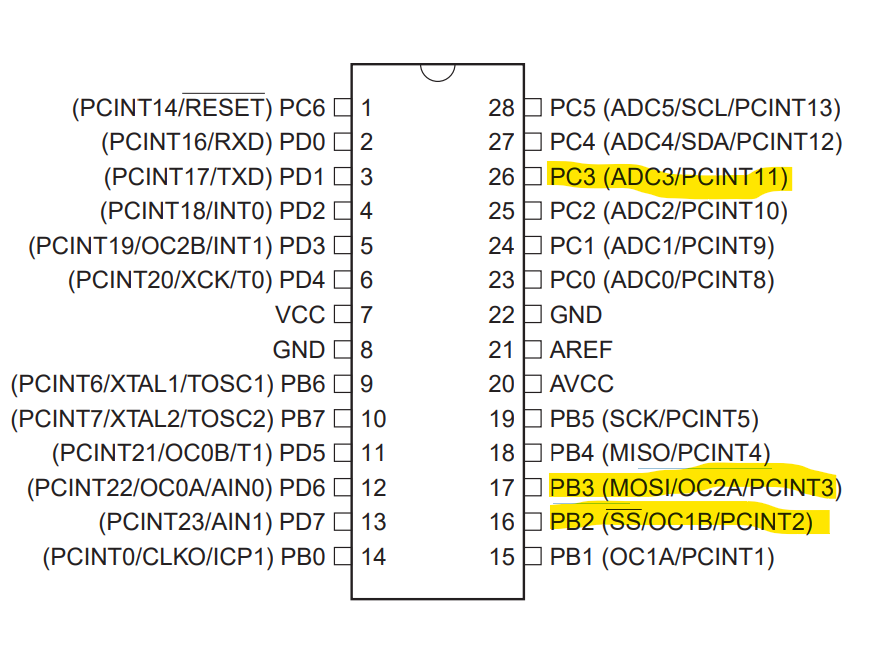
1. **COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS**

Atmel Studio 7: Assembler, debugger and simulator.

Atmega328PB-Xmini PC。

Multi-function shield: LED and Pushbutton.

Pins used



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

Task 1 C code:

/\*

\* DA2CT1.c

\*

\* Created: 3/4/2020 7:43:16 PM

\* Author : c1029

\*/

#define F\_CPU 16000000UL //1/16M = 62.5ns for one clock

#include <avr/io.h>

void on\_2sec()

{

int cycle = 122; //2sec/62.5ns/1024/255 = 122 cycles

while(cycle != 0)

{

cycle--;

while((TIFR0 & 0x01) == 0); //if overflow is set, clear it with reset to 1.

TIFR0 = 1<<TOV0;

}

}

void on\_412()

{

int cycle = 25; //412ms/62.5ns/1024 = 6437.5 clocks to overflow

PORTB |= (1<<3); //6438/255 = 25 cycles

while(cycle != 0)

{

if(!(PINC &(1<< PINC3))) //if PINC.3 is pressed.

{

PORTB = 0x08; //Turn LED on

on\_2sec(); //delay for 2 second.

PORTB |=(1<<2); //Turn LED off.

}

cycle--;

while((TIFR0 & 0x01) == 0); //if overflow is set, clear it with reset to 1.

TIFR0 |= 1<<TOV0;

}

}

void off\_338()

{

int cycle = 20; //338ms/62.5ns/1024 = 5281.25 clocks to overflow

PORTB &=~(1<<3); //5281.25/255 = 20 cycles

while(cycle != 0)

{

if(!(PINC &(1<< PINC3))) //if PINC.3 is pressed.

{

PORTB = 0x08; //Turn LED on.

on\_2sec(); //delay for 2 second.

PORTB |=(1<<2); //Turn LED off.

}

cycle--;

while((TIFR0 & 0x01) == 0);

TIFR0 = 1<<TOV0; //clear overflow flag

}

}

int main(void)

{

DDRC &= ~(1<<3); //set PORTC.3 as input

DDRB = 0x0C; //set PORTB.3 and PORTB.2 as output

PORTC |= (1<<3); //active pull-up resistor for pushbutton

PORTB |= (1<<2); //pull-up resistor for LED

/\* Replace with your application code \*/

while (1)

{

TCCR0A = 0x00; //normal mode operation

TCCR0B = 0x05; //prescaler of 1024

TCNT0 = 0X00;

on\_412(); //hold PORTB.3 HIGH for 412ms

off\_338(); //hold PORTB.3 LOW for 338ms

}

}

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

Task 2 C code:

/\*

\* DA2CT2.c

\*

\* Created: 3/4/2020 9:58:21 PM

\* Author : c1029

\*/

#define F\_CPU 16000000UL

#include <avr/io.h>

#include <avr/interrupt.h>

volatile *uint8\_t* cycle\_cnt; //global variable modified by ISR.

int main(void)

{

DDRC &= ~(1<<3); //set PORTC.3 as input

DDRB = 0x0C; //set PORTB.3 and PORTB.2 as output

PORTC |= (1<<3); //active pull-up resistor for pushbutton

PORTB |= (1<<2) | (1 << 3); //pull-up resistor for LED and PORTB.3

TCNT0 = 0x00;

TCCR0A = 0x00; //normal mode operation

TCCR0B = 0x05; //prescaler of 1024

TIMSK0 = (1<<TOIE0); //enable timer0 overflow interrupt

sei();

/\* Replace with your application code \*/

while (1)

{

if(!(PINC & (1<<PINC3)))

{

cycle\_cnt = 0;

PORTB = 0x08;

while(cycle\_cnt < 123);

PORTB |=(1<<2);

cycle\_cnt = 0;

}

if((cycle\_cnt < 45) && (cycle\_cnt >= 20))

{

PORTB |= (1 << 3); //set PORTB.3 HIGH

}

if(cycle\_cnt > 45)

{

PORTB &= ~(1<<3);

cycle\_cnt = 0; //set PORTB.3 LOW

}

}

}

ISR(TIMER0\_OVF\_vect)

{

cycle\_cnt++;

}

Task 3 C code:

/\*

\* DA2CT3.c

\*

\* Created: 3/4/2020 11:05:15 PM

\* Author : c1029

\*/

#define F\_CPU 16000000UL

#include <avr/io.h>

#include <avr/interrupt.h>

volatile *uint8\_t* cycle\_cnt; //global variable modified by ISR.

int main(void)

{

DDRC &= ~(1<<3); //set PORTC.3 as input

DDRB = 0x0C; //set PORTB.3 and PORTB.2 as output

PORTC |= (1<<3); //active pull-up resistor for pushbutton

PORTB |= (1<<2) | (1 << 3); //pull-up resistor for LED and PORTB.3

TCNT0 = 0x00;

TCCR0A = (1<<WGM01); //CTC mode operation

TCCR0B = 0x05; //prescaler of 1024

OCR0A = 0x80; //overflow happens when TCNT exceed 0x80 or 128

TIMSK0 = (1<<OCIE0A); //enable timer0 compare interrupt

sei();

/\* Replace with your application code \*/

while (1)

{

if(!(PINC & (1<<PINC3)))

{

cycle\_cnt = 0;

PORTB = 0x08;

while(cycle\_cnt < 245);

PORTB |=(1<<2);

cycle\_cnt = 0;

}

if((cycle\_cnt < 90) && (cycle\_cnt >= 40))

{

PORTB |= (1 << 3); //set PORTB.3 HIGH

}

if(cycle\_cnt > 90)

{

PORTB &= ~(1<<3);

cycle\_cnt = 0; //set PORTB.3 LOW

}

}

}

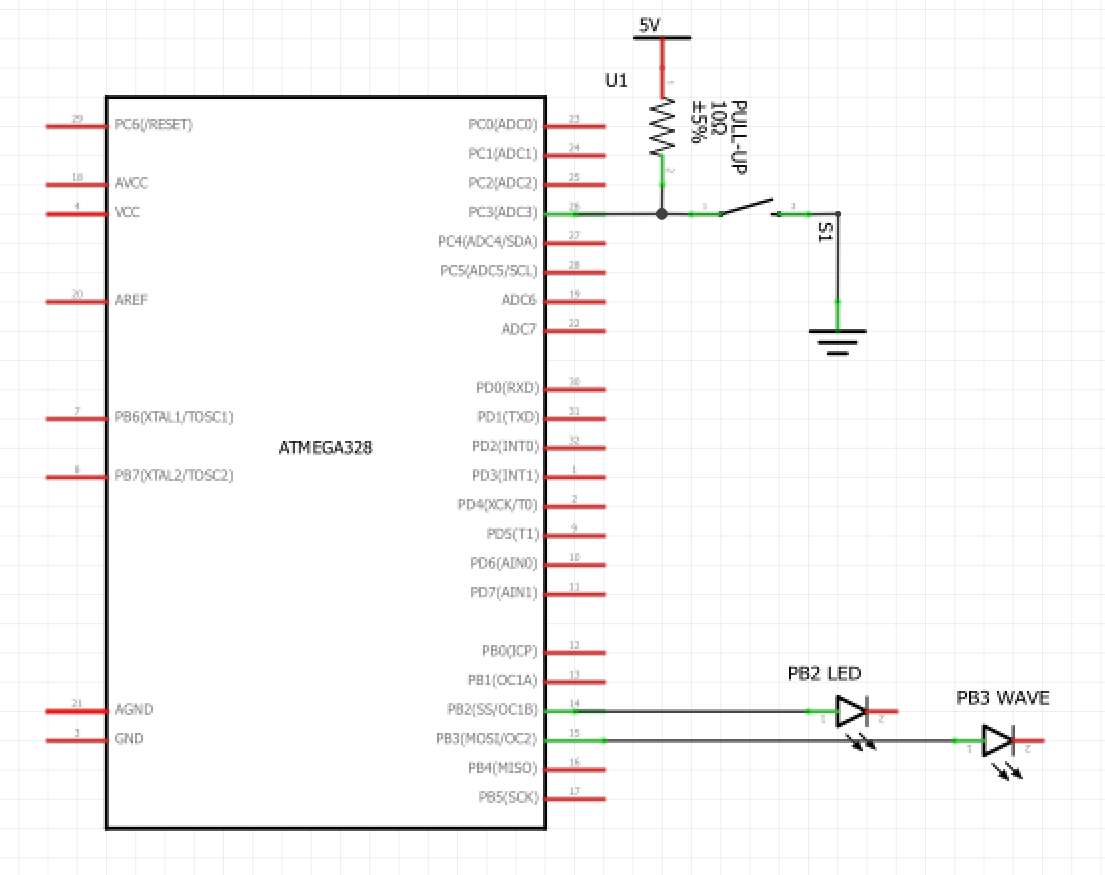
ISR(TIMER0\_COMPA\_vect)

{

cycle\_cnt++;

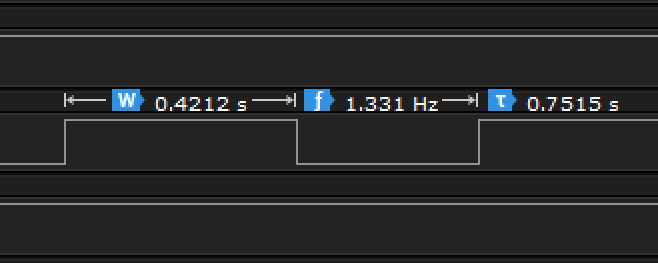
}

1. **SCHEMATICS**

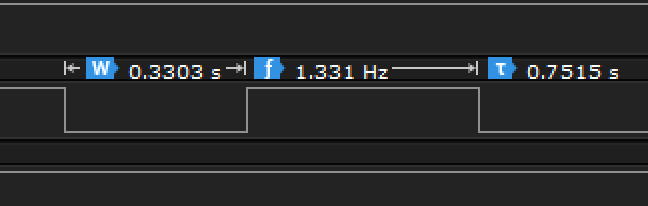


1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**

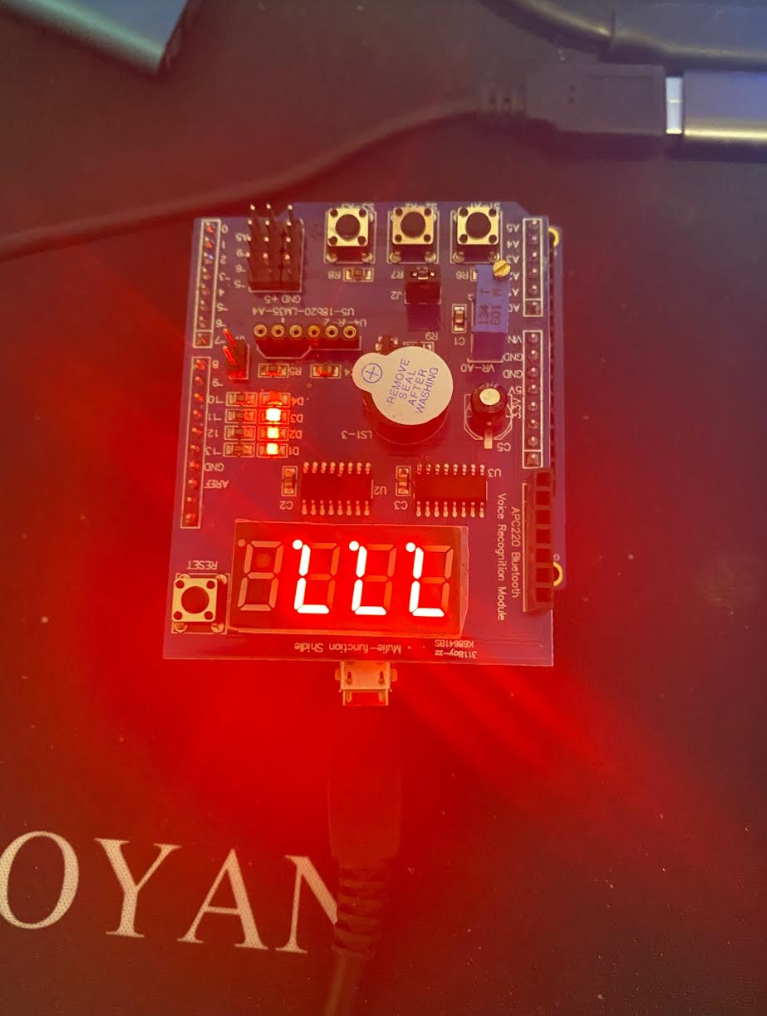
**On for 0.42 sec:**

****

**Off for 0.33 sec:**



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



1. **VIDEO LINKS OF EACH DEMO**

Task 1: <https://youtu.be/azjUurHv-o0>

Task 2: <https://youtu.be/5ENykd-rViE>

Task 3: <https://youtu.be/EeoZShHN83c>

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”.

Xianjie Cao