CPE301 – SPRING 2019

Design Assignment 2C

Student Name: Cody Hudson

Student #: 8000603642

Student Email: hudsoc1@unlv.nevada.edu

Primary Github address: https://github.com/eed911/class\_proj.git

Directory:

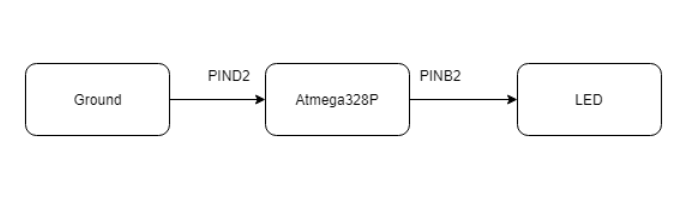
https://github.com/eed911/class\_proj/tree/master/DesignAssignments/DA2C/Project\_2C

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

List of Components used

* ATmega328P Xplained
* Muli Function Shield
  + LED
  + GROUND

Block diagram with pins used in the Atmega328P



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

Task 1: Generate A waveform with 60% duty cycle and a period of 724ms using timers.

Code:

/\*

\* DA2C\_A.c

\*

\* Created: 3/22/2019 11:09:35 PM

\* Author : Cody Hudson

\*/

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

int Count = 0;

int main(void)

{

DDRB |= (1<<DDB2); //SETS DIRECTION OF PORT B2 AS OUTPUT

TCNT0 = 0x8F; //SETS HIGHEST VALUE FOR TIMMER 0x8F

TCCR0A = 0x00; //SETS TIMMER TO BE NORMAL MODE

TCCR0B = 0x05; //SETS PRESCALLER TO 1024

while(1){

while((TIFR0 & 0x01) == 0) //WHEN FLAG ISNT SET STAY HERE

;

TCNT0 = 0x8F; //SET TNCT BACK UP TO 0x8F

TIFR0 = 0x01; //CLEAR FLAG

Count++; //INCRIMENT COUNTER

if(Count == 60) //40% OF THE TIME DO THIS

PORTB &= (0<<PORTB2); //SET WAVEFORM LOW

if(Count == 100){ //60% OF THE TIME DO THIS

PORTB |= (1<<PORTB2); //SET WAVEFORM HIGH

Count = 0; //SET COUNTER BACK TO 0

}

}

}

Task 2: Turn on the LED for 1.25s when Pin D.2 goes low Using timers.

/\*

\* DA2C\_C.c

\*

\* Created: 3/13/2019 9:22:00 PM

\* Author : Cody Hudson

\*/

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

int main(void)

{

DDRB |= (1<<DDB2); //SETS DIRECTION OF PORT B2 AS OUTPUT

PORTB |= (1<<PORTB2); //TURNS ON PULL UP RESISTOR OF PORT B2

DDRD &= (0<<DDD2); //SETS DIRECITON OF PORT D2 AS INPUT

PORTD |= (1<<PORTD2); //TURNS ON PULL UP RESISTOR OR PORT D2

EICRA = 0x00; //SETS THE INTRUPT TO ACTIVATE ON LOW LEVEL

EIMSK = (1<<INT0); //SETS THE INTRUPT TO BE EXTERNAL INTERUPTS 0

sei (); //SETS GLOBAL INTRUPT

while(1); //WAITS HERE FOR INTRUPT

}

ISR (INT0\_vect) //EXTERNAL INTERUPT SUBROUTINE

{

*uint8\_t* Count = 0; //SETS INTERGER VARIABLE COUNT TO 0

TCCR0A = 0; //SETS TIMMER TO NORMAL MODE

TCCR0B |= (1<<CS02)|(1<<CS00); //SETS PRESCALER TO 1024

while (Count < 76){ //WHILE COUNT IS LESS THAN 76 LOOP

PORTB &= (0<<PORTB2);//TURNS LED ON

if ((TIFR0 & 0x01) == 1){ //IF BIT 1 IN FLAG IS SET HIGH DO THIS

TIFR0 = 0X01; //CLEAR FLAG REGISTER

Count++; //INCRIMENT COUNT

}

}

PORTB |= (1<<PORTB2); //TURNS LED OFF

}

Task 3: Generate A waveform with 60% duty cycle and a period of 724ms using timers and TIMER0\_OVF\_vect interrupt mechanism in normal mode.

Code:

/\*

\* DA2C\_INT\_NORMAL\_A.c

\*

\* Created: 3/23/2019 1:41:52 AM

\* Author : Cody Hudson

\*/

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

int Count = 0;

int main(void)

{

DDRB |= (1<<DDB2); //SETS DIRECTION OF PORT B2 AS OUTPUT

TIMSK0 |= (1<<TOIE0); //THE INTERRUPT TO ACTIVATE WHEN TIMER 0 HAS OVERFLOW

sei(); //SETS GLOBAL INTERRUPT

TCNT0 = 0x8F; //SETS TCNT0 TO 0x8F

TCCR0A = 0x00; //SETS TIMMER 0 TO NORMAL MODE

TCCR0B = 0x05; //SETS PRESCALLER TO 1024

while(1);

}

ISR (TIMER0\_OVF\_vect){ //TV0V INTERUPT SUBROUTINE

while(1){

while((TIFR0 & 0x01) == 0) //WHILE FLAG ISNT SET STAY HERE

;

TCNT0 = 0x8F; //SET TCNT0 BACK TO 0x8F

TIFR0 = 0x01; //CLEAR FLAG

Count++; //INCRIMENT TIMMER

if(Count == 60) //40% OF THE TIME DO THIS

PORTB &= (0<<PORTB2); //SET WAVEFORM LOW

if(Count == 100){ //60% OF THE TIME DO THIS

PORTB |= (1<<PORTB2); //SET WAVEFORM HIGH

Count = 0; //SET COUNTER BACK TO 0

}

}

}

Task4: Turn on the LED for 1.25s when Pin D.2 goes low Using timers and TIMER0\_OVF\_vect interrupt mechanism in normal mode.

/\*

\* DA2C.c

\*

\* Created: 3/13/2019 8:59:41 PM

\* Author : Cody Hudson

\*/

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

int Count = 0;

int main(void)

{

DDRB |= (1<<DDB2); //SETS DIRECTION OF PORT B2 AS OUTPUT

PORTB |= (1<<PORTB2); //TURNS ON PULL UP RESISTOR OF PORT B2

DDRD &= (0<<DDD2); //SETS DIRECITON OF PORT D2 AS INPUT

PORTD |= (1<<PORTD2); //TURNS ON PULL UP RESISTOR OR PORT D2

EICRA = 0x00; //SETS THE INTRUPT TO ACTIVATE ON LOW LEVEL

EIMSK = (1<<INT0); //SETS THE INTRUPT TO BE EXTERNAL INTERUPTS 0

sei (); //SETS GLOBAL INTRUPT

while(1); //WAITS HERE FOR INTRUPT

}

ISR (INT0\_vect) //EXTERNAL INTERRUPT ON PIN D.2

{

TIMSK0 |= (1<<TOIE0); //SETS THE INERRUPT TO ACTIVATE WHEN TIMMER 0 GETS OVERFLOW FLAG

sei (); //SETS GLOBAL INTERRUPT

TCCR0A = 0x00; //SETS NORMAL MODE ON TIMMER

TCCR0B |= (1<<CS02)|(1<<CS00); //SETS PRESCALER TO 1024

while (1){ //WAIT HERE UNITL FURTHER NOTICE

if (Count > 75){ //WHEN COUNT = 75 DO THIS

Count = 0; //SET COUNT TO 0

break; //EXIT WHILE LOOP

}

}

}

ISR (TIMER0\_OVF\_vect){ //TV0V FLAG INTERUPT

while (Count < 76){ //WHILE COUNT IS LESS THAN 76 LOOP

PORTB &= (0<<PORTB2);//TURNS LED ON

if ((TIFR0 & 0x01) == 1){ //IF TV0V FLAG IS HIGH DO THIS

TIFR0 = 0X01; //CLEAR FLAG

Count++; //INCREMENT COUNT

}

}

PORTB |= (1<<PORTB2); //TURNS LED OFF

}

Task 5: Generate A waveform with 60% duty cycle and a period of 724ms using timers and TIMER0\_COMPA\_vect interrupt mechanism in CTC mode.

Code:

/\*

\* DA2C\_INT\_CTC\_A.c

\*

\* Created: 3/23/2019 1:47:49 AM

\* Author : Cody Hudson

\*/

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

int Count = 0;

int main(void)

{

DDRB |= (1<<DDB2); //SETS DIRECTION OF PORT B2 AS OUTPUT

OCR0A = 0x71; //SETS THE COUNTER TO COUNT TO 113

TIMSK0 |= (1<<OCIE0A); //SETS INTERRUPT TO ACTIVATE ON A COMPARE OF OCR0A

sei(); //SETS GLOBAL INTERRUPT

TCCR0A |= (1<<WGM01); //SETS TIMMER TO CTC MODE

TCCR0B = 0x05; //SETS PRESCALLER TO 1024

while(1); //WAITS HERE FOR INTERRUPT

}

ISR (TIMER0\_COMPA\_vect){ //TIMMER COMPARE SUBROUTINE

while(1){

while((TIFR0 & 0x02) == 0) //WHILE FLAG ISNT SET STAY HERE

;

TIFR0 = 0x02; //CLEARS FLAG

Count++; //INCRIMENTS COUNT

if(Count == 60) //40% OF THE PERIOD DO THIS

PORTB &= (0<<PORTB2); //SET WAVFORM LOW

if(Count == 100){ //60% OF THE PERIODDO THIS

PORTB |= (1<<PORTB2); //SET WAVEFORM HIGH

Count = 0; //SET COUNT BACK TO 0

}

}

}

Task 6: Turn on the LED for 1.25s when Pin D.2 goes low Using timers and TIMER0\_COMPA\_vect interrupt mechanism in CTC mode.

Code:

/\*

\* DA2C\_Int\_CTC.c

\*

\* Created: 3/13/2019 9:38:47 PM

\* Author : Cody Hudson

\*/

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

int Count = 0; //SETS GLOBAL VARIABLE COUNT TO 0

int main(void)

{

DDRB |= (1<<DDB2); //SETS DIRECTION OF PORT B2 AS OUTPUT

PORTB |= (1<<PORTB2); //TURNS ON PULL UP RESISTOR OF PORT B2

DDRD &= (0<<DDD2); //SETS DIRECITON OF PORT D2 AS INPUT

PORTD |= (1<<PORTD2); //TURNS ON PULL UP RESISTOR OR PORT D2

EICRA = 0x00; //SETS THE INTRUPT TO ACTIVATE ON LOW LEVEL

EIMSK = (1<<INT0); //SETS THE INTRUPT TO BE EXTERNAL INTERUPTS 0

sei (); //SETS GLOBAL INTRUPT

while(1)

Count = 0; //WAITS HERE FOR INTRUPT

}

ISR (INT0\_vect){ //EXTERNAL INTERUPT ON PIN D.2

OCR0A = 0xC2; //SETS THE HIGHEST VALUE OF THE TIMMER TO BE 194

TIMSK0 |= (1<<OCIE0A); //SETS INTRUPT TO ACTIVATE ON COMPARE OF OCR0A

sei (); //SETS GLOBAL INTRUPT

TCCR0A |= (1<<WGM01); //SETS TIMMER TO CTC MODE

TCCR0B |= (1<<CS02) | (1<<CS00); //SETS PRESCALLER TO 1024

while (Count < 100); //WAITS HERE UNTIL TIMMER INTERUPT IS COMPLETE

}

ISR (TIMER0\_COMPA\_vect){ //TIMMER COMPARE INTERUPT A

while (Count < 100){ //CONINUES LOOP UNTIL COUNT = 100

PORTB &= (0<<PORTB2);//TURNS LED ON

if ((TIFR0 & 0x02) == 0x02){ //IF THE OCF0A FLAG IS SET HIGH DO THIS

TIFR0 = 0X02; //CLEAR FLAG

Count++; //INCRIMENT COUNT

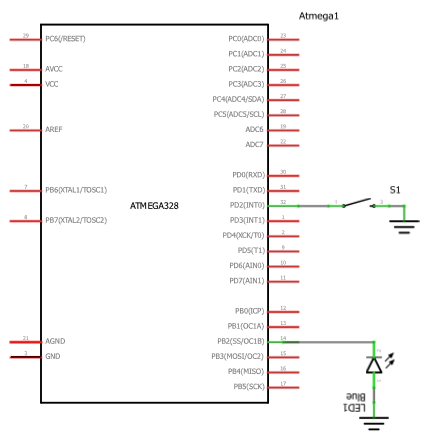
}

}

PORTB |= (1<<PORTB2); //TURNS LED OFF

}

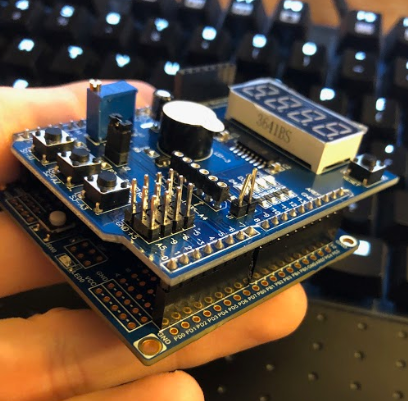
1. **SCHEMATICS**



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

N/a all demonstration is done via video on YouTube

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



1. **VIDEO LINKS OF EACH DEMO**

Demo1:

<https://youtu.be/kQJVRoGPj64>

Demo2:

<https://youtu.be/vYM88h3z_5I>

Demo3:

<https://youtu.be/OLJRZnXzdcs>

Demo4:

<https://youtu.be/iMryfgcbQrY>

Demo5:

<https://youtu.be/fiQu3nS5XEo>

Demo6:

<https://youtu.be/N3Y76dcDLR8>

1. **GITHUB LINK OF THIS DA**

https://github.com/eed911/class\_proj.git

**Student Academic Misconduct Policy**

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

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

NAME OF THE STUDENT