Damian Cisneros

CPE301 – SPRING 2018

Design Assignment 2

**DO NOT REMOVE THIS PAGE DURING SUBMISSION:**

The student understands that all required components should be submitted in complete for grading of this assignment.

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **SUBMISSION ITEM** | **COMPLETED (Y/N)** | **MARKS**  **(/MAX)** |
| 1 | COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS |  |  |
| 2. | INITIAL CODE OF TASK 1/A |  |  |
| 3. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 2/B |  |  |
| 3. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 3/C |  |  |
| 3. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 4/D |  |  |
| 3. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 5/E |  |  |
| 4. | SCHEMATICS |  |  |
| 5. | SCREENSHOTS OF EACH TASK OUTPUT |  |  |
| 5. | SCREENSHOT OF EACH DEMO |  |  |
| 6. | VIDEO LINKS OF EACH DEMO |  |  |
| 7. | GOOGLECODE LINK OF THE DA |  |  |
|  |  |  |  |
|  |  |  |  |

Damian Cisneros

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

List of Components used:

ATmega328p

Breadboard

1k resistor

Pushbutton for switch

LED

Block diagram with pins used in the Atmega328P

1. **INITIAL/DEVELOPED CODE OF TASK 1/Assembly**

;

; DA2\_Task1\_asm.asm

;

; Created: 3/15/2018 2:18:30 PM

; Author : Damian Cisneros

; Description : Program generates waveform with 0.5s period and 50% duty cycle

; waveform is output to LED connected to PINB2 (uses 8MHz clock)

;

.ORG 0

LDI R21**,** 0 ;Used to toggle LED

SBI DDRB**,** 2 ;Set PINB2 to output

LDI R20**,** 8

**//**STS TCCR1B**,** R20 ;Set to CTC mode

LDI R20**,** 13

STS TCCR1B**,** R20 ;Set prescalar to 1024

LDI R20**,** 0X0A ;Higher bits of 3905 calculated as 0.25s

LDI R21**,** 0X07 ;Lower bits of 3905 calculated as 0.25s

STS OCR1AH**,** R21 ;Set output compare register high bits

STS OCR1AL**,** R20 ;Set output compare register low bits

**LOOP:**

**IN** R20**,** TIFR1 ;Grab TIFR0 (Interrupt Flag Register)

ANDI R20**,** 4 ;Isolate OCFA (Output compare A match flag)

BREQ **LOOP** ;Check if OCFA flag is set. If it is move on to next instruction.

;Else loop back

EOR R21**,** R20;Toggle LED

**OUT** PORTB**,** R21 ;Output LED

LDI R20**,** 4 ;Used to clear TIFR flag

**OUT** TIFR1**,** R20 ; Clear TIFR flag

RJMP **LOOP**

1. **INITIAL/DEVELOPED CODE OF TASK 1/C Code**

/\*

\* DA2\_Task1.c

\*

\* Created: 3/15/2018 1:02:45 PM

\* Author : Damian Cisneros

\* Description : Program generates waveform with 0.5s period and 50% duty cycle

\* waveform is output to LED connected to PINB2 (uses 8MHz clock)

\*/

#include <avr/io.h>

int main**(**void**)**

**{**

int maxCT**=**3905**;** //set max calculated for 0.5s

DDRB**=** 0b00000100**;** //set PINB2 to output

TCCR1A **=** 0**;** //normal mode

TCCR1B **=** 5**;** //set prescale: 1024

**while** **(**1**)**

**{**

**if(**TCNT1**>=(**maxCT**/**2**))** //50% duty cycle

**{**

TCNT1**=**0**;** //reset count

PORTB **^=** DDRB**;** //toggle led

**}**

**}**

**return** 0**;**

**}**

1. **INITIAL/DEVELOPED CODE OF TASK 2/Assembly**

;

; DA2\_Task2\_asm.asm

;

; Created: 3/15/2018 7:10:59 PM

; Author : Damian Cisneros

; Description : This program lights up LED on PINB2 1s after a pushbutton is

; pressed on PORTD2 (8MHz clock)

.ORG 0

CBI DDRD**,** 2 ;set PIND2 to input

SBI DDRB**,** 2 ;set PINB2 to output

LDI R20**,** 0XFF ;Turn on pull-up resistor for PIND2

**OUT** PORTD**,** R20

LDI R20**,** 13

STS TCCR1B**,** R20 ;Set CTC mode with prescalar 1024

**LOOP:**

**IN** R20**,** PIND ;grab PIND

**NEG** R20 ;Pull-up resistor logic

ANDI R20**,** 4 ;check PIND2 input

BREQ **LOOP**

LDI R20**,**0x00

STS TCNT1H**,**R20 ;resetting the counter to 0

LDI R20**,**0x00

STS TCNT1L**,**R20 ;resetting the counter to 0

RCALL delay ;delay 1s

LDI R20**,** 4

**OUT** PORTB**,** R20 ;turn on LED for 1s

RCALL delay ;delay 1s

LDI R20**,** 0

**OUT** PORTB**,** R20 ;turn off LED

RJMP **LOOP**

delay**:**

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

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

CPI R28**,** 0x41 ;comparing if lower is 0x41

BRSH body ;if false repeat

RJMP delay ;if true repeat

body**:**

CPI R29**,**0x0f ;comparing if higher is 0x0f

BRSH done

RJMP delay

done**:**

LDI R20**,**0x00

STS TCNT1H**,**R20 ;resetting the counter to 0 for next round

LDI R20**,**0x00

STS TCNT1L**,**R20 ;resetting the counter to 0 for next round

**RET**

1. **INITIAL/DEVELOPED CODE OF TASK 2/C Code**

/\*

\* DA2\_Task2\_c.c

\*

\* Created: 3/15/2018 6:03:57 PM

\* Author : Damian Cisneros

\* Description: Program lights up LED in PORTB2 1s after Pushbutton is pressed

\* which stays lit for 2s before it turns off again.

\*/

#define F\_CPU 8000000UL //define 8MHz

#include <avr/io.h>

#include <util/delay.h>

int main**(**void**)**

**{**

DDRD **=** 4**;** //set PIND2 to input

DDRB **=** 4**;** //set PINB2 to output

PORTB **|=** 0b00000100**;** //Turn on pull-up resistor

**while** **(**1**)**

**{**

**if(**PIND **&** 0b00000100**){** //check if PIND2 is set

\_delay\_ms**(**1000**);** //delay 1s

PORTB **|=** **(**PINB2 **<<** 1**);** //turn on LED

\_delay\_ms**(**2000**);** //delay 2s

**}**

**else{**

PORTB **&=** **~(**PINB2 **<<** 1**);** //turn off LED

**}**

**}**

**}**

1. **INITIAL/DEVELOPED CODE OF TASK 3/C Code**

/\*

\* DA2\_Task3\_c.c

\*

\* Created: 3/16/2018 10:49:48 AM

\* Author : Damian Cisneros

\* Description : This program works with 1MHz clock and toggles an LED in PINB2

\* every .5s with a duty cycle of 50% which toggles at .25s rate

\*/

//clock is 1MHz

#include <avr/io.h>

int main**(**void**)**

**{**

DDRB **=** 4**;** //set PINB2 as output

TCCR1B **=** 5**;** //set CTC mode with 1024 prescale

TCNT1 **=** 65293**;** //calculated count for 0.5s/2 for 50% duty cycle

**while** **(**1**)**

**{**

**if** **(**TIFR1 **&** **(**1 **<<** TOV1**)){** //check overflow flag

PORTB **^=** DDRB**;** //toggle LED

TCNT1 **=** 65293**;** // reload calculated count for 0.5s/2 for 50% duty cycle

TIFR1 **|=** **(**1 **<<** TOV1**);** //clear overflow flag

**}**

**}**

**}**

1. **INITIAL/DEVELOPED CODE OF TASK 4/C Code**

/\*

\* DA2\_Task4\_c.c

\*

\* Created: 3/15/2018 9:26:13 PM

\* Author : Damian Cisneros

\* Description : This program blinks LED connected to PINB2 with a square wave

\* at a rate 0.25s which is 50% duty cycle of 0.5s using TIMER1

\* OVF interrupt (1MHz clock)

\*/

#include <avr/io.h>

#include <avr/interrupt.h>

ISR**(**TIMER1\_OVF\_vect**){**

PORTB **^=** DDRB**;** //toggle LED

TCNT1 **=** 65293**;** //reset TCNT1 to 0.25s calculated based on 50% duty cycle of 0.5s

**}**

int main**(**void**)**

**{**

DDRB **=** 4**;** //set PORTB2 as output

TCCR1B **=** 5**;** //set prescalar 1024

TCNT1 **=** 65293**;** //set TCNT1 to 0.25s calculated based on 50% duty cycle of 0.5s

TIMSK1 **|=** **(**1 **<<** TOIE1**);** //enable OVF interrupt

sei**();** //enable global interrupts

**while** **(**1**)**

**{**

**}**

**}**

1. **INITIAL/DEVELOPED CODE OF TASK 5/C Code**

/\*

\* DA2\_Task5\_c.c

\*

\* Created: 3/15/2018 9:47:24 PM

\* Author : Damian Cisneros

\* Description : Program lights up LED 1s after INT0 interrupt occurs by

\* pressing the button. It stays on for 2s then turns off

\* (1MHz clock, LED connected to PINB2, pushbutton INT0 pin)

\*/

#include <avr/io.h>

#include <avr/interrupt.h>

#define F\_CPU 1000000UL

#include <util/delay.h>

int main**(**void**)**

**{**

DDRB **=** 4**;** //set PORTB2 as output for LED

DDRD **&=** **~(**1 **<<** PIND2**);** //set PORTD2 as input for pushbutton switch

EIMSK **=** **(**1 **<<** INT0**);** //enable INT0 interrupt

EICRA **=** 3**;** //enable INT0 flag on rising edge

sei**();** //enable global interrupt

**while** **(**1**)**

**{**

**}**

**}**

ISR**(**INT0\_vect**){**

\_delay\_ms**(**5**);** //debounce

**if** **(**PIND **&** 0x04**){**

\_delay\_ms**(**1000**);** //initial 1s delay after pressing button

PORTB **|=** **(**1 **<<** PINB2**);** //turn on LED connected to PINB2 for 2s

\_delay\_ms(2000); //delay on mode for 2s

PORTB &= ~(1 << PINB2); //turn off LED

}

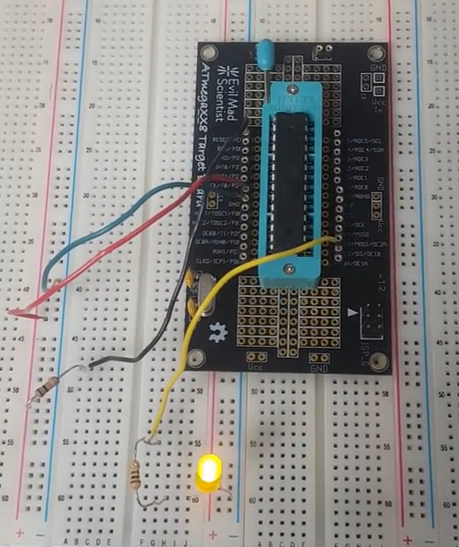
}

1. **SCHEMATICS**

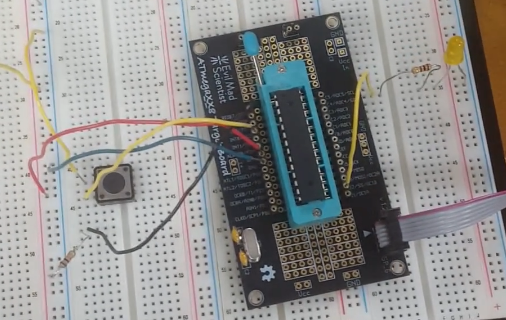
Use fritzing.org

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

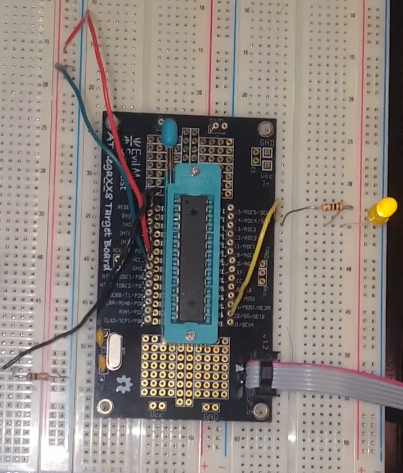
Task 1 Assembly and C Board Setup



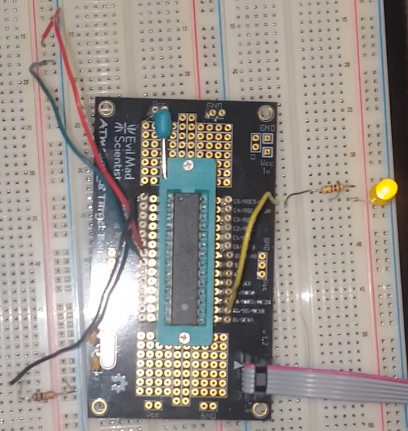
Task 1 Assembly and C Board Setup



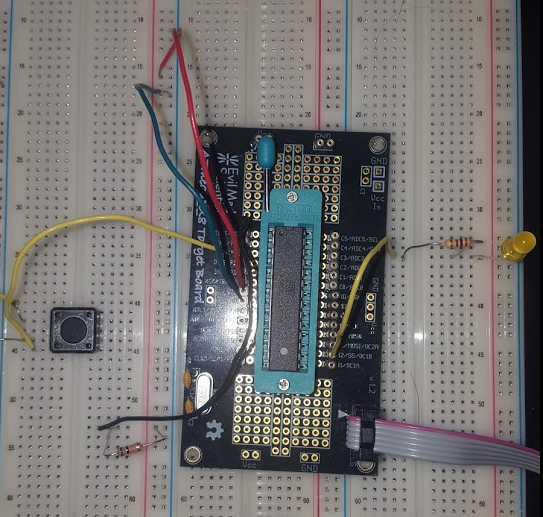
Task 2 Assembly and C Board Setup



Task 3 C Board Setup



Task 4 C Board Setup



Task 5 C Board Setup

1. **VIDEO LINKS OF EACH DEMO**

Playlist - <https://www.youtube.com/playlist?list=PL5RuXbzEXwes4cYuR7YLbzXgePpAIsU8P>

Task 1/Assembly - <https://youtu.be/_-Z28W37Cok>

Task 1/C Code - <https://youtu.be/O0jzXN2m8TA>

Task 2/Assembly - <https://youtu.be/8KltKFdTjsA>

Task 2/C Code - <https://youtu.be/gRL5-Tx6MFY>

Task 3/C Code - <https://youtu.be/vfq_NPs05nc>

Task 4/C Code - <https://youtu.be/jwo2JOPgxP4>

Task 5/C Code - <https://youtu.be/OTYh48hzvEs>

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”.