Guillermo Gálvez

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 |  |  |
|  |  |  |  |
|  |  |  |  |

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

Red LED

Red Push Button “switch”

AtMega XPlained Mini

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

*/\**

*\* task1da2C.c*

*\**

*\* Created: 3/11/2018 7:36:48 PM*

*\* Author : Guillermo Gálvez*

*\*/*

#define F\_CPU 8000000UL

#include<avr/io.h>

#include<util/delay.h>

#include <avr/interrupt.h>

int main(void)

{

DDRB = (1 << 2); *//PB2 is an output*

**while** (1)

{

\_delay\_ms(250);

PORTB ^= (1 << 2); *//toggle PB2*

}*//end while (1)*

}*//end main*

;

; task1da2.asm

;

; Created: 3/11/2018 11:26:25 AM

; Author : Guillermo Gálvez

start:

;T\_on 1952

;T\_off 1952

:T = 3905

;TCNT = (8MHz/1024 \* .5/2)-1 = 1952

;TCNT = 0x07A0

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

.org 0

LDI R16, (1 << 2) ;To set PB2

OUT DDRB, R16 ;PB2 as output

LDI R17,0 ;needed to toogle led

OUT PORTB,R17

LDI R20,5 ;to set prescaler

STS TCCR1B,R20 ;Prescaler: 1024

begin:

RCALL delay ;calling timer to wait for 1 sec

EOR R17,R16 ;XOR to toogle led

OUT PORTB,R17

RJMP begin ;repeating i.e, while(1)

delay:

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

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

CPI R28,0xA0 ;comparing if lower is 0xA0

BRSH body

RJMP delay

body:

CPI R29,0x07 ;comparing if higher is 0x07

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

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

*/\**

*\* task2.c*

*\**

*\* Created: 3/14/2018 4:00:04 PM*

*\* Author : Guillermo Gálvez*

*\*/*

#define F\_CPU 8000000UL

#include <avr/io.h>

#include <util/delay.h>

int main ()

{

DDRB |= (1 << DDB2); *//PB2 set to ouput*

DDRD &= ~(1 << DDD2); *//Set to input*

PORTD |= (1 << 2); *//Set Pull-up*

PORTB = 0; *//Initilize output to 0*

**while** (1)

{

**if**(!(PIND & (1 << PIND2))) *//Pull-up resistor switch to ground*

{

\_delay\_ms(1000); *//Delay 1 second*

PORTB ^= (1 << PB2); *//LED\_on*

}

**else**

{

PORTB = 0; *//LED\_off*

}

}*//end while(1)*

}*//Main*

*/\**

*DA2 task3*

*\* Created: 3/11/2018 7:36:48 PM*

*\* Author : Guillermo Gálvez*

*\*/*

#define F\_CPU 8000000

#include<avr/io.h>

#include<util/delay.h>

#include <avr/io.h>

#include <avr/interrupt.h>

*/\**

*Task 3*

*repeat task 1 T = .5s generate a 50% DC*

*8,000,000/64 = 125,000=f, T=1/125,000 8µS*

*1/125,000 = 8µS \* 62,500 cycles =.5sec*

*62,500/2 = 31,250 \* .8µS=250ms or .25s*

*T\_on = .25s*

*T\_off = .25s*

*T = .5s*

*\*/*

int main(void)

{

DDRB |= (1 << 2); *//Set PB2 output*

PORTB = 0; *//initialize to 0*

TCCR0A = 0; *//normal mod*

TCCR0B |= (1 << CS02) | (0 << CS01) | (0 << CS00); *//prescaler 256*

TCNT0 = 0; *//initialize counter*

unsigned int i; *//loop control variable*

**while**(1)

{

**for**(i=0; i < 62500; i++)

{

**if**(i == 31250)

PORTB ^= (1 << 2);

**if**(i == 62499)

PORTB ^= (0 << 2);

}

}*//end while(1)*

}*//end main*

*/\**

*\* task4da2C.c*

*\**

*\* Created: 3/11/2018 7:36:48 PM*

*\* Author : Guillermo Gálvez*

*\*/*

#define F\_CPU 8000000UL

#include<avr/io.h>

#include<util/delay.h>

#include <avr/interrupt.h>

#define l\_on PORTB |= (1 << PORTB2)

#define l\_off PORTB &= ~(1 << PORTB2)

#define l\_tog PINB |= (1 << PINB2)

ISR(TIMER0\_OVF\_vect)

{

l\_tog;

\_delay\_ms(250); *//LED\_on half the period .5s*

l\_tog;

\_delay\_ms(250);

}

int main(void)

{

DDRB |= (1 << DDB2); *//set ouput*

TCCR0B |= (1 << CS02) | (1 << CS00); *//Pre-Scaler 1024*

TIMSK0 |= (1 << TOIE0); *//Set-OVF*

sei();

**while** (1)

{

}*//end while (1)*

}*//end main*

*/\**

*\* task5.c*

*\**

*\* Created: 3/14/2018 4:00:04 PM*

*\* Author : Guillermo Gálvez*

*\*/*

#define F\_CPU 8000000UL

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

ISR(INT0\_vect)

{

EIFR |= (1 << INTF0);

**if**(!(PIND & (1 << PIND2))) *//Pull-up resistor switch to ground*

{

\_delay\_ms(1000); *//Delay 1 second*

PORTB ^= (1 << PB2); *//LED\_on*

}

**else**

{

PORTB = 0; *//LED\_off*

}

}*//end ISR(INT0\_vect)*

int main ()

{

DDRB |= (1 << DDB2); *//PB2 set to ouput for LED*

DDRD &= ~(1 << PD2); *//Set to input for switch*

PORTD |= (1 << PD2); *//Set Pull-up*

EIMSK |= (1 << INT0); *//activate extern int*

EICRA |= (0 << ISC01) | (1 << ISC00); *//activiate external int on rise*

PORTB = 0; *//Initilize output to 0*

sei(); *//enagle global int*

**while** (1)

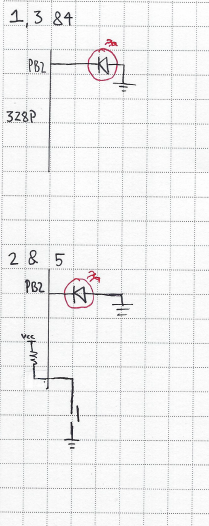
{

}*//end while(1)*

}*//end Main*

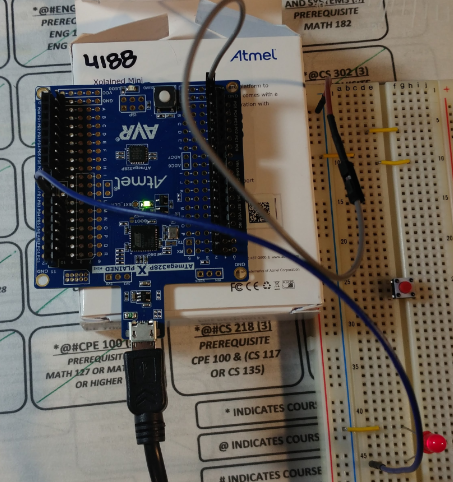
1. **SCHEMATICS**

Use fritzing.org 🡨 kept freezing

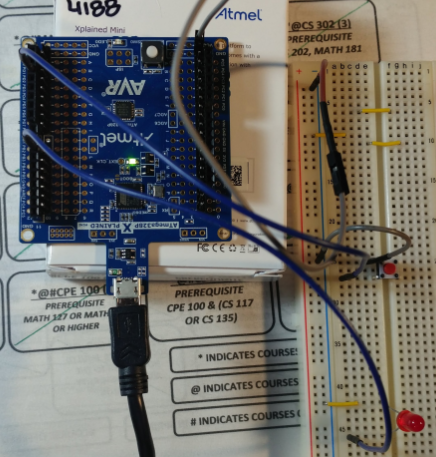


1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**
2. **SCREENSHOT OF EACH DEMO (BOARD SETUP)**

Setup for 1, 3 & 4



Setup for 2 & 5



1. **VIDEO LINKS OF EACH DEMO**

**T1** <https://youtu.be/ioQhbm5Fx-A>

**T2** <https://youtu.be/DytUQHgWp_A>

**T3** <https://youtu.be/SfZQFTl89JE>

**T4** <https://youtu.be/byQ5tLtj-7U>

**T5** <https://youtu.be/zLHmBcHWPas>

1. **GITHUB LINK OF THIS DA**

<https://github.com/galveg1/Design-Assignments/tree/master/DA2>

**Student Academic Misconduct Policy**

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

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

Guillermo Gálvez