Desi Battle

CPE301 – SPRING 2016

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)** |
| 0. | COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS |  |  |
| 1. | INITIAL CODE OF TASK 1/A |  |  |
| 2. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 2/B |  |  |
| 3. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 3/C |  |  |
| 4. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 4/D |  |  |
| 6. | SCHEMATICS |  |  |
| 7. | SCREENSHOTS OF EACH TASK OUTPUT |  |  |
| 8. | SCREENSHOT OF EACH DEMO |  |  |
| 9. | VIDEO LINKS OF EACH DEMO |  |  |
| 10. | GOOGLECODE LINK OF THE DA |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 0. | COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 1. | INITIAL CODE OF TASK 1/A |  |  |

; DA2T1ASM.asm

;

; Created: 3/12/2016 8:23:42 PM

; Author: battled

;

; Created: 3/9/2016 10:21:51 AM

;task 1 in asm (toggle pc0

; Author : desi.battle

;

.MACRO INITSTACK

LDI R16, HIGH(RAMEND)

OUT SPH,R16

LDI R16,LOW(RAMEND)

OUT SPL,R16

.ENDMACRO

INITSTACK ;use Macro here

LDI R16, 0X01

OUT DDRC, R16; MAKE PC0 OUTPUT

;set 0th bit in r16 to 1 for EOR with PC0

LDI R16,0b00000001 ;eor reg for PC0

LDI R17,0x80 ;portc temp holder

OUT PORTC,R17 ;PC = 0 ;reset ==1 (active low)

LDI R20,0x00

STS TCCR1A,R20 ;WGM11:10 = 00 Normal mode

BEGIN:

RCALL DELAY

EOR R17,R16 ;toggle Bit0 of R17;

OUT PORTC,R17 ;toggle PC0

RJMP BEGIN

;TIMER1 DELAY

DELAY:

LDI R20,HIGH(34286)

STS TCNT1H,R20 ;TCNT1H = high portion of 34285

LDI R20,LOW(34286)

STS TCNT1L,R20 ;TCNT1L = low portion of 34285

LDI R20,0x03

STS TCCR1B,R20 ;WGM13:12 = 00, Normal mode, prescaler = 3 (1/64)

;poll for timer1 overflow interrupt

AGAIN:

IN R20,TIFR1 ;read TIFR

SBRS R20,TOV1 ;if TOV1 is set skip next instruction

RJMP AGAIN

LDI R20,0x00

STS TCCR1B,R20 ;stop Timer1

LDI R20,0x01

OUT TIFR1,R20 ;clear TOV1 flag

RET

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*

\* DA2.c

\*task 1

\* Created: 3/9/2016 9:00:36 PM

\* Author: battled

\*/

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

#include <avr/io.h> //include stdio

#include <util/delay.h> //include delay subroutine

//stack auto initialized in C

int main(void)

{

DDRC = 0X01; //MAKE PC0 OUTPUT

while (1)

{

//toggle portc every 250 ms for .5 second period

\_delay\_ms(250);

PORTC ^= (1<<PC0);

}

}

|  |  |  |  |
| --- | --- | --- | --- |
| 2. | INITIAL CODE OF TASK 2/B |  |  |

;

; DA2new.asm

;

; Created: 3/9/2016 10:21:51 AM

; task 2 in asm

; Author : desi.battle

;

.MACRO INITSTACK

LDI R16, HIGH(RAMEND)

OUT SPH,R16

LDI R16,LOW(RAMEND)

OUT SPL,R16

.ENDMACRO

INITSTACK ;use Macro here

;Make entire port B output.

LDI R16, 0XFF

OUT DDRB, R16

LDI R16, 0X01

OUT DDRC, R16; MAKE PC0 OUTPUTS

;set 0th bit in r16 to 1 for EOR with PC0

LDI R16,0b00000001 ;eor reg for PC0

LDI R17,0x80 ;portc temp holder

LDI R18,0 ;R18 IS COUNTER

OUT PORTC,R17 ;PC = 0

LDI R20,0x00

STS TCCR1A,R20 ;WGM11:10 = 00 Normal mode

BEGIN:

RCALL DELAY

EOR R17,R16 ;toggle Bit0 of R17;

OUT PORTC,R17 ;toggle PC0

SBIS PORTC, 0 ;only increment Portb if portc.0 is 1

RJMP NOCOUNT

INC R18

OUT PORTB, R18

NOCOUNT:

RJMP BEGIN

;TIMER1 DELAY

DELAY:

LDI R20,HIGH(34286)

STS TCNT1H,R20 ;TCNT1H = high portion of 34285

LDI R20,LOW(34286)

STS TCNT1L,R20 ;TCNT1L = low portion of 34285

LDI R20,0x03

STS TCCR1B,R20 ;WGM13:12 = 00, Normal mode, prescaler = 3 (1/64)

;poll for timer1 overflow interrupt

AGAIN:

IN R20,TIFR1 ;read TIFR

SBRS R20,TOV1 ;if TOV1 is set skip next instruction

RJMP AGAIN

LDI R20,0x00

STS TCCR1B,R20 ;stop Timer1

LDI R20,0x01

OUT TIFR1,R20 ;clear TOV1 flag

RET

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* DA2.c

\* task 2

\* Created: 3/9/2016 9:00:36 PM

\* Author: battled

\*/

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

#include <avr/io.h> //include std io

#include <util/delay.h> //include delay

//stack auto initialized in C

int main(void)

{

DDRB = 0XFF; //MAKE PORT B OUTPUT FOR COUNTING

DDRC = 0X31; //MAKE PC0 OUTPUTS

while (1)

{

\_delay\_ms(250);

PORTC ^= (1<<PC0);

if (PORTC ==0X01) //only count when portc0 == 1

PORTB++;

}

}

|  |  |  |  |
| --- | --- | --- | --- |
| 3. | INITIAL CODE OF TASK 3/C |  |  |

;

; DA2new.asm

;task 3 asm

; Created: 3/9/2016 10:21:51 AM

; Author : desi.battle

;

.MACRO INITSTACK

LDI R16, HIGH(RAMEND)

OUT SPH,R16

LDI R16,LOW(RAMEND)

OUT SPL,R16

.ENDMACRO

INITSTACK ;use Macro here

;Make entire port B output.

LDI R16, 0XFF

OUT DDRB, R16

LDI R16, 0X31

OUT DDRC, R16; MAKE PC5,PC4,PC0 OUTPUTS

;set 0th bit in r16 to 1 for EOR with PC0

LDI R16,0b00000001 ;eor reg for PC0

LDI R17,0x80 ;portc temp holder

LDI R18,0 ;R18 IS COUNTER

OUT PORTC,R17 ;PC = 0

LDI R21, 0 ;5COUNT

BEGIN:

RCALL DELAY

EOR R17,R16 ;toggle Bit0 of R17;

OUT PORTC,R17 ;toggle PC0

SBIS PORTC, 0 ;only alter portc.5,4 or inc portb when portc.0 == 1

RJMP NOCOUNT

INC R21 ;r21 tracks rising edges

CPI R21, 5

BREQ DOFIVE ;on the 5th rising edge flip portc.4

CPI R21, 10

BREQ DOTEN ;on the 10th rising edge flip portc.4,5 reset r21

RJMP DONEFLIP

DOFIVE:

LDI R19, 0X10

RJMP STOREFLIP

DOTEN:

LDI R19, 0X30

LDI R21, 0

STOREFLIP:

EOR R17, R19

OUT PORTC, R17

DONEFLIP:

INC R18 ;increment portb every rising edge

OUT PORTB, R18

NOCOUNT:

RJMP BEGIN

;TIMER1 DELAY

DELAY:

LDI R20,HIGH(34286)

STS TCNT1H,R20 ;TCNT1H = high portion of 34285

LDI R20,LOW(34286)

STS TCNT1L,R20 ;TCNT1L = low portion of 34285

LDI R20,0x00

STS TCCR1A,R20 ;WGM11:10 = 00 Normal mode

LDI R20,0x03

STS TCCR1B,R20 ;WGM13:12 = 00, Normal mode, prescaler = 3 (1/64)

AGAIN:

IN R20,TIFR1 ;read TIFR

SBRS R20,TOV1 ;if TOV1 is set skip next instruction

RJMP AGAIN

LDI R20,0x00

STS TCCR1B,R20 ;stop Timer1

LDI R20,0x01

OUT TIFR1,R20 ;clear TOV1 flag

RET

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* DA2.c

\* task3

\* Created: 3/9/2016 9:00:36 PM

\* Author: battled

\*/

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

#include <avr/io.h>

#include <util/delay.h>

//stack auto initialized in C

int main(void)

{

int i=0; //used to control port 9 and 10

DDRB = 0XFF; //MAKE PORT B OUTPUT FOR COUNTING

DDRC = 0X31; //MAKE PC5, PC4, PC0 OUTPUTS

while (1)

{

\_delay\_ms(250);

PORTC ^= (1<<PC0);

//must account for each state portc may be in due to flipping of bits portc4,5

if ( (PORTC ==0X01 ) || (PORTC == 0X11 ) || (PORTC == 0X21) || (PORTC == 0X31))

{

PORTB++;

i++;

if (i==5)

PORTC ^= (1<<PC4); //(0x10); flip portc4

else if (i ==10)

{

PORTC ^= (1<<PC4)|(1<<PC5); //0x30 flip portc4,5

i = 0;

}

}

}

}

|  |  |  |  |
| --- | --- | --- | --- |
| 4. | INITIAL CODE OF TASK 4/D |  |  |

;

; DA2new.asm

;

; Created: 3/9/2016 10:21:51 AM

; Author : desi.battle

;

.ORG 0x0000 ;RESET LOCATION

JMP MAIN

.ORG 0X0008 ;PCINT1 ADDR NOTE ONCE INTERRUPT HANDLER STARTED THE OFFENDING INTERRUPT FLAG AND I BIT IN STATUS REG ARE DISABLED

JMP PCINTX

PCINTX:

SBIS PORTC, 0

RJMP NOCOUNT

INC R21

CPI R21, 5

BREQ DOFIVE

CPI R21, 10

BREQ DOTEN

RJMP DONEFLIP

DOFIVE:

LDI R19, 0X10

RJMP STOREFLIP

DOTEN:

LDI R19, 0X30

LDI R21, 0

STOREFLIP:

EOR R17, R19

OUT PORTC, R17

DONEFLIP:

INC R18

OUT PORTB, R18

NOCOUNT:

RETI ;RETI RETURNS TO CODE BEING EXECUTED WHEN INTERRUPT OCCURS AND SETS I BIT TO ONE IN SREG

.MACRO INITSTACK

LDI R16, HIGH(RAMEND)

OUT SPH,R16

LDI R16,LOW(RAMEND)

OUT SPL,R16

.ENDMACRO

MAIN:

INITSTACK ;use Macro here

LDI R22, 1<< PCINT8

STS PCMSK1, R22 ;SET PCINT8 BIT TO ONE TO ENABLE PC0 TO TRIGGER INTERRUPTS

LDI R22, 1<< PCIE1

STS PCICR, R22 ;ENABLE PINCHANGE1 INTERRUPT

SEI ;SET I FLAG ENABLING GLOBAL INTERRUPTS

;Make entire port B output.

LDI R16, 0XFF

OUT DDRB, R16

LDI R16, 0X31

OUT DDRC, R16; MAKE PC5,PC4,PC0 OUTPUTS

;set 0th bit in r16 to 1 for EOR with PC0

LDI R16,0b00000001 ;eor reg for PC0

LDI R17,0x80 ;portc temp holder

LDI R18,0 ;R18 IS COUNTER

OUT PORTC,R17 ;PC = 0

LDI R21, 0 ;5COUNT

BEGIN:

RCALL DELAY

EOR R17,R16 ;toggle Bit0 of R17;

OUT PORTC,R17 ;toggle PC0

RJMP BEGIN

;TIMER1 DELAY

DELAY:

LDI R20,HIGH(34286)

STS TCNT1H,R20 ;TCNT1H = high portion of 34285

LDI R20,LOW(34286)

STS TCNT1L,R20 ;TCNT1L = low portion of 34285

LDI R20,0x00

STS TCCR1A,R20 ;WGM11:10 = 00 Normal mode

LDI R20,0x03

STS TCCR1B,R20 ;WGM13:12 = 00, Normal mode, prescaler = 3 (1/64)

AGAIN:

IN R20,TIFR1 ;read TIFR

SBRS R20,TOV1 ;if TOV1 is set skip next instruction

RJMP AGAIN

LDI R20,0x00

STS TCCR1B,R20 ;stop Timer1

LDI R20,0x01

OUT TIFR1,R20 ;clear TOV1 flag

RET

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* DA2.c

\* task 4

\* Created: 3/9/2016 9:00:36 PM

\* Author: battled

\*/

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

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

//stack auto initialized in C

ISR(PCINT1\_vect)

{

static int i=0; //declare i as static int to prevent re-initialization every interrupt

//enter if statement if pc0 == 1

if ( (PORTC ==0X01 ) || (PORTC == 0X11 ) || (PORTC == 0X21) || (PORTC == 0X31))

{

PORTB++;

i++;

if (i==5)

PORTC ^= (1<<PC4); //(0x10); //flip portc4

else if (i ==10)

{

PORTC ^= (1<<PC4)|(1<<PC5); //0x30; flip portc4,5 reinit i

i = 0;

}

}

}

int main(void)

{

while(1)

{

PCICR = 1 << PCIE1; //enable pcint1

PCMSK1 =1 << PCINT8; //enable pcint8 to allow pc0 to trigger interrupt (set flag)

sei();

DDRB = 0XFF; //MAKE PORT B OUTPUT FOR COUNTING

DDRC = 0X31; //MAKE PC5, PC4, PC0 OUTPUTS

while (1)

{

\_delay\_ms(250);

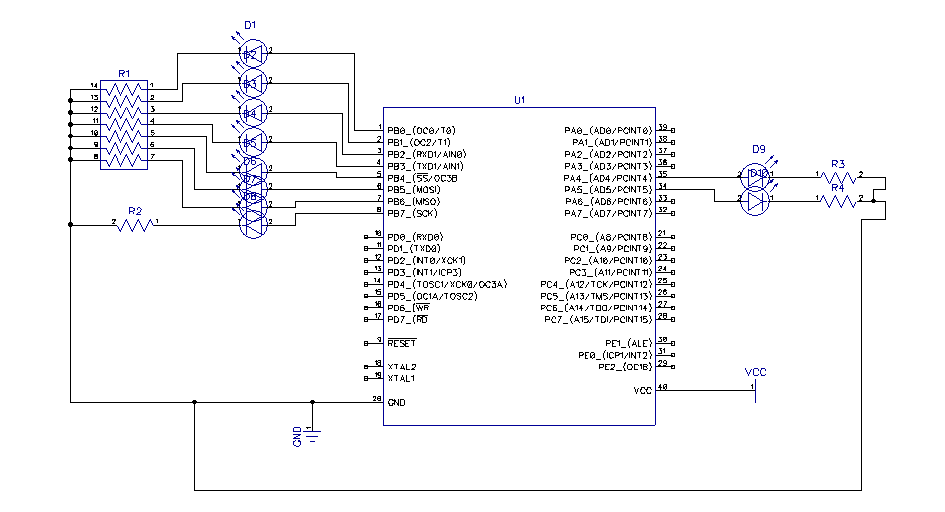
PORTC ^= (1<<PC0);

}

}

}

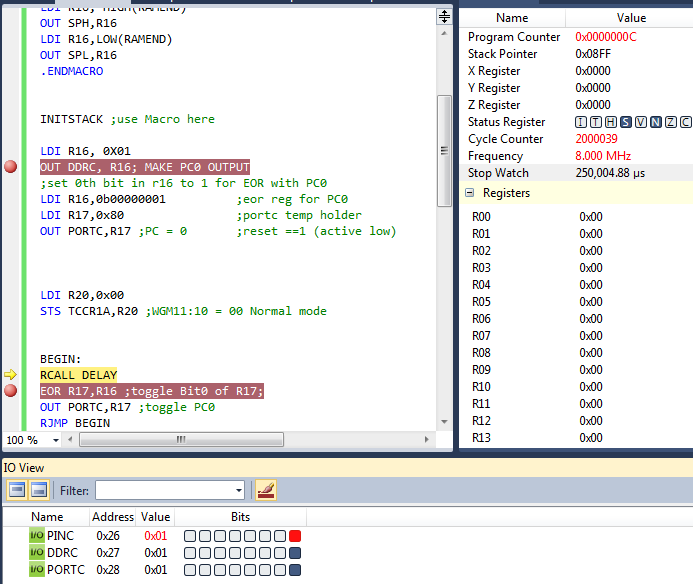
|  |  |  |  |
| --- | --- | --- | --- |
| 6. | SCHEMATICS |  |  |



|  |  |  |  |
| --- | --- | --- | --- |
| 7. | SCREENSHOTS OF EACH TASK OUTPUT |  |  |

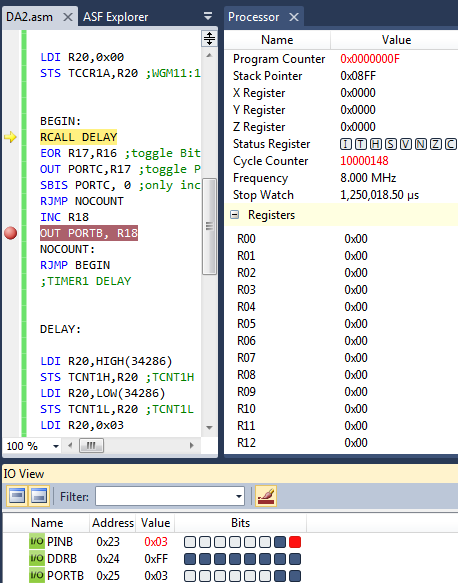
TASK 1/A:

Verify duty cycle and period: 50% duty cycle, period = 0.5 second



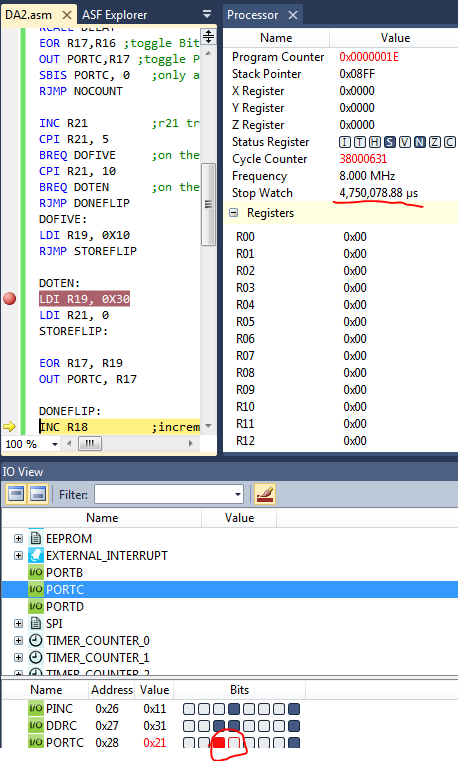
TASK 2/B:

Verify port B counts on rising edge. (@1,250,000 us) == PortB = 3



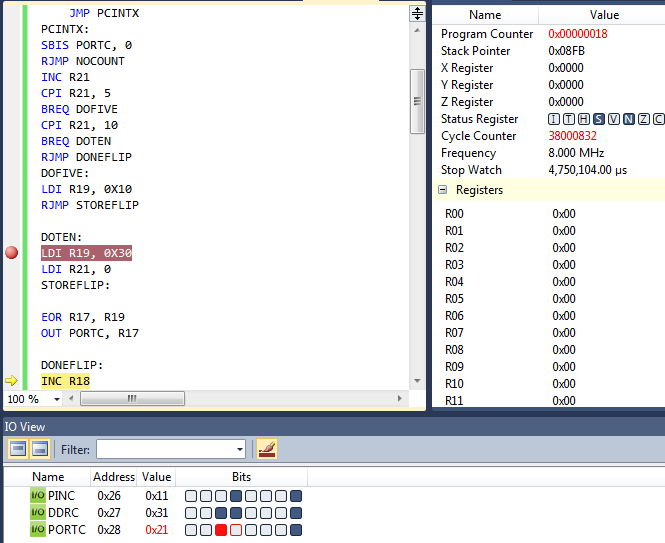
TASK 3/C:

Verify PortC.4 toggles every 5th rising edge and PortC.5 toggles every tenth (shown below is tenth rising edge)



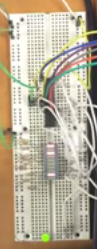
TASK 4/D:

Verify (almost) same output when using interrupt



|  |  |  |  |
| --- | --- | --- | --- |
| 8. | SCREENSHOT OF EACH DEMO |  |  |

On the 5th rising edge light 9 is toggled on. On the 10th rising edge the 10th light is toggled on while the 9th light is switched off. On the 15th pulse the 9th light is toggled back on while the 10th light remains on. On the 255th rising edge all lights are high.



|  |  |  |  |
| --- | --- | --- | --- |
| 9. | VIDEO LINKS OF EACH DEMO |  |  |
| https://youtu.be/Fniq5enrHW8 | | | |
| 10. | GOOGLECODE LINK OF THE DA |  |  |
| https://github.com/battled/DA0.git | | | |

**Student Academic Misconduct Policy**

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

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

NAME OF THE STUDENT