Emmanuel Rodriguez Lopez

CPE301 – SPRING 2016

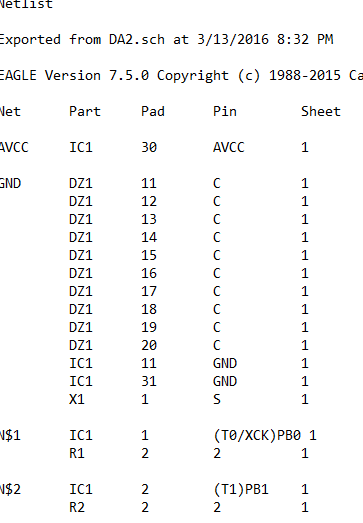
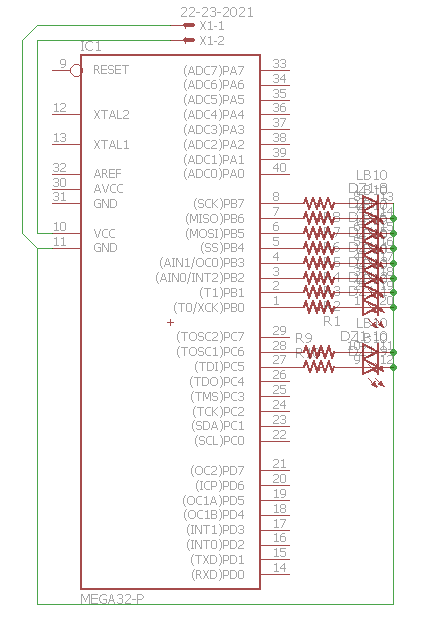
Design Assignment 2

**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 |  |  |
| 5. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 5/E |  |  |
| 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 |  |  |

ASSEMBLY CODE FOR TASK 1

SBI DDRC, 0

BEGIN:

LDI R20, 0x07

STS OCR1AH, R20

LDI R20, 0xA1

STS OCR1AL, R20 ;ocr1a set to ¼ second if prescaler is 1024

LDI R20, 0x00

STS TCCR1A, R20

LDI R20, 0x0D

STS TCCR1B, R20 ;prescaler of 1024, CTC mode

LDI R20, 1

out PORTC, R20

L1:

IN R20, TIFR1

SBRS R20, OCF1A ;polling for flag

RJMP L1

LDI R20, 0

STS TCCR1B, R20 ;turning off timer 1

LDI R20, 1<<OCF1A ;clear flag

OUT TIFR1, R20

LDI R20, 0

OUT PORTC, R20 ;toggle pin

LDI R20, 0x07

STS OCR1AH, R20 ;set timer again, same as before since 50% DC

LDI R20, 0xA1

STS OCR1AL, R20

LDI R20, 0x00

STS TCCR1A, R20

LDI R20, 0x0D

STS TCCR1B, R20

L2:

IN R20, TIFR1 ;polling for flag

SBRS R20, OCF1A

RJMP L2

LDI R20, 0

STS TCCR1B, R20 ;clear flag

LDI R20, 1<<OCF1A

OUT TIFR1, R20

JMP BEGIN

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

C Code for Task 1

/\*

\* GccApplication1.c

\*

\* Created: 3/4/2016 1:33:49 PM

\* Author: RODRIE2

\*/

#define F\_CPU 8000000UL //setting frequency

#include <avr/io.h>

#include <util/delay.h>

int main(void)

{

DDRC |= (1<<PORTC0); //makes PC0 an output

PORTC= 1; //PC0 high to start

while(1)

{

\_delay\_us(250000); //1/4 sec delay

if (PORTC == 1) //to toggle PC0

{

PORTC = 0; //happens every 1/5 second, the period

}

else

{

PORTC = 1;

}

}

return 0;

}

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

ASSEMBLY CODE FOR TASK 2

/\*

\* DA2TASK3\_ASM.asm

\*

\* Created: 3/7/2016 5:40:38 PM

\* Author: erl25\_000

\*/

SBI DDRB, 1 ;all PORTB pins as outputs

SBI DDRB, 0

SBI DDRB, 2

SBI DDRB, 3

SBI DDRB, 4

SBI DDRB, 5

SBI DDRB, 6

SBI DDRB, 7

LDI R21, 0 ;used for keeping track of inc rising edges

BEGIN:

LDI R20, 0x07

STS OCR1AH, R20

LDI R20, 0xA1

STS OCR1AL, R20 ;set to ¼ sec

LDI R20, 0x00

STS TCCR1A, R20

LDI R20, 0x0D

STS TCCR1B, R20 ;CTC mode and prescaler = 1024

INC R21

Out PORTB, R21 ;output # of rising edges

L1:

IN R20, TIFR1 ;poll flag

SBRS R20, OCF1A

RJMP L1

LDI R20, 0

STS TCCR1B, R20

LDI R20, 1<<OCF1A ;clear flag

OUT TIFR1, R20

LDI R20, 0x07

STS OCR1AH, R20

LDI R20, 0xA1

STS OCR1AL, R20

LDI R20, 0x00

STS TCCR1A, R20

LDI R20, 0X0D

STS TCCR1B, R20

L2:

IN R20, TIFR1

SBRS R20, OCF1A

RJMP L2

LDI R20, 0

STS TCCR1B, R20

LDI R20, 1<<OCF1A

OUT TIFR1, R20

JMP BEGIN

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

C CODE FOR TASK 2

/\*

\* GccApplication1.c

\*

\* Created: 3/4/2016 1:33:49 PM

\* Author: RODRIE2

\*/

#define F\_CPU 8000000UL //setting frequency

#include <avr/io.h>

#include <util/delay.h>

int main(void)

{

DDRB = 0xFF; //all of PORTB is output

PORTB = 0;

int i = 1; //indicator for duty cycle

while(1)

{

\_delay\_us(250000); //1/4 sec delay, 50% DC

if (i == 1)

{

PORTB++; //inc every rising edge occurring every ½ sec

i = 0;

}

else

{

i = 1;

}

}

return 0;

}

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

ASSEMBLY CODE FOR TASK 3

/\*

\* DA2TASK3\_ASM.asm

\*

\* Created: 3/7/2016 5:40:38 PM

\* Author: erl25\_000

\*/

SBI DDRC, 5

SBI DDRC, 6

SBI DDRB, 1

SBI DDRB,0

SBI DDRB, 2

SBI DDRB, 3

SBI DDRB, 4

SBI DDRB, 5

SBI DDRB, 6

SBI DDRB, 7

LDI R21, 0

LDI R22, 0 ;5th rising edge indicator

LDI R23, 0 ;10th rising edge indicator

BEGIN:

LDI R20, 0x07

STS OCR1AH, R20

LDI R20, 0xA1

STS OCR1AL, R20 ; 1/4 sec of ½ sec 50% DC

LDI R20, 0x00

STS TCCR1A, R20

LDI R20, 0x0D

STS TCCR1B, R20 ;CTC mode and 1024 prescaler

INC R22

INC R23

CPI R22, 5 ;checking for 5th rising

BRNE CHK10

LDI R20, 0x20

OUT PORTC, R20 ;toggling PC5

LDI R22, 0 ;resetting 5th indicator

CHK10:

CPI R23, 10 ;checking for 10th rising edge

BRNE NORM

LDI R20, 0x40

OUT PORTC, R20 ;toggling PC6

LDI R23, 0 ;resetting 10th indicator

NORM:

INC R21

Out PORTB, R21 ;inc portb every rising edge

L1:

IN R20, TIFR1 ;polling for flag

SBRS R20, OCF1A

RJMP L1

LDI R20, 0

STS TCCR1B, R20

LDI R20, 1<<OCF1A ;clearing flag

OUT TIFR1, R20

LDI R20, 0x07

STS OCR1AH, R20

LDI R20, 0xA1

STS OCR1AL, R20

LDI R20, 0x00

STS TCCR1A, R20

LDI R20, 0X0D

STS TCCR1B, R20

L2:

IN R20, TIFR1

SBRS R20, OCF1A

RJMP L2

LDI R20, 0

STS TCCR1B, R20

LDI R20, 1<<OCF1A

OUT TIFR1, R20

JMP BEGIN

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

C CODE FOR TASK 3

/\*

\* DA2TASK3\_C.c

\*

\* Created: 3/13/2016 5:55:59 PM

\* Author: Rodriguez

\*/

#define F\_CPU 8000000UL //setting frequency

#include <avr/io.h>

#include <util/delay.h>

int main(void)

{

DDRB = 0xFF; //all PORTB as output

DDRC = 0x60; //PC5, PC6 as output

PORTB = 0;

//PORTC = 0;

int i = 1; //Duty cycle indicator

int five = 0; //5th rising edge indicator

int ten = 0; //10th rising edge indicator

while(1)

{

\_delay\_us(250000); //1/4 sec delay, 50DC of ½ sec period

if (i == 1) //at rising edge

{

PORTB++; //rising edge # out

five++;

ten++;

i = 0;

if (ten == 10) //at 10th rising edge, toggle PC6, clear indicators

{

PORTC = 0x40;

ten = 0;

five = 0;

}

if (five==5) //at 5th rising edge, toggle PC5, clear indicator

{

PORTC = 0x20;

five = 0;

}

}

else

{

i = 1;

}

}

return 0;

}

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

ASSEMBLY CODE FOR TASK 4

/\*

\* AssemblerApplication1.asm

\*

\* Created: 3/10/2016 10:04:08 AM

\* Author: RODRIE2

\*/

.org 0x00

JMP MAIN

.org 0x1C

JMP COMP\_INT

MAIN:

SBI DDRC, 5

SBI DDRC, 6

SBI DDRB, 1

SBI DDRB,0

SBI DDRB, 2

SBI DDRB, 3

SBI DDRB, 4

SBI DDRB, 5

SBI DDRB, 6

SBI DDRB, 7

LDI R21, 0

LDI R18, 0

LDI R19, 0

BEGIN:

LDI R20, HIGH(RAMEND)

OUT SPH, R20

LDI R20, LOW(RAMEND)

OUT SPH, R20

LDI R20, 0x07 ;1/2 SECOND

STS OCR1AH, R20 ;OCR1A = 07A1

LDI R20, 0xA1

STS OCR1AL, R20

LDI R20, 0x00

STS TCCR1A, R20

LDI R20, 0x0d

STS TCCR1B, R20

LDI R20, (1<<OCIE1A)

sts TIMSK1, R20

SEI

LOOP:

JMP LOOP

COMP\_INT:

INC R18

INC R19

CPI R18, 5 ;for every 5th rising edge, toggle PC5

BRNE CHK10

LDI R20, 0x20

OUT PORTC, R20

LDI R18, 0

CHK10:

CPI R19, 10 ;for every 10th rising edge, toggle PC6

BRNE NORM

LDI R20, 0x40

OUT PORTC, R20

LDI R19, 0

NORM:

INC R21

Out PORTB, R21 ;inc PORTB every rising edge

;RESETTING CLOCK

LDI R20, 0x0F

STS OCR1AH, R20

LDI R20, 0x45

STS OCR1AL, R20

LDI R20, 0x00

STS TCCR1A, R20

LDI R20, 0x0D

STS TCCR1B, R20

RETI

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

C CODE FOR TASK 4

/\*

\* DA2TASK4\_C.c

\*

\* Created: 3/13/2016 7:20:38 PM

\* Author: Rodriguez

\*/

#define F\_CPU 8000000UL

#include <util/delay.h>

#include <avr/io.h>

#include <avr/interrupt.h>

int i = 1, five, ten; //indicators for DC, 5th and 10th rising edge

int main(void)

{

DDRB = 0xFF;

DDRC = 0x60;

PORTC = 0;

PORTB = 0;

TCCR1A = 0x00;

TCCR1B = 0x0D; //CTC mode with 1028 prescaler

OCR1A = 0x07A1; //1/4 sec

TIMSK1 |= (1<<OCIE1A); //enable interrupt mask

sei(); /enable global interrupts

while(1)

{

//infinite loop

}

}

ISR(TIMER1\_COMPA\_vect)

{

if (i==1)

{

PORTB++;

five++;

ten++;

i = 0;

if (ten == 10)

{

PORTC = 0x40;

ten = 0;

five = 0;

}

if (five==5)

{

PORTC = 0x20;

five = 0;

}

}

else if (i == 0)

{

i=1;

}

// RESETTING CLOCK

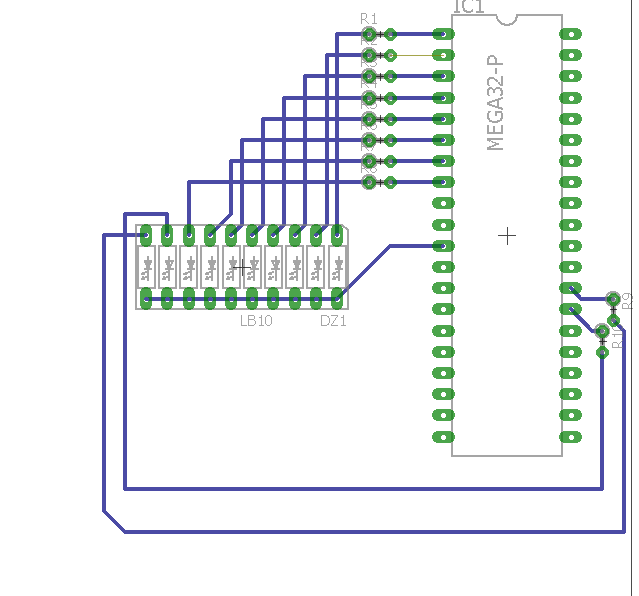
TCCR1A = 0x00;

TCCR1B = 0x0D;

OCR1A = 0x07A1;

}

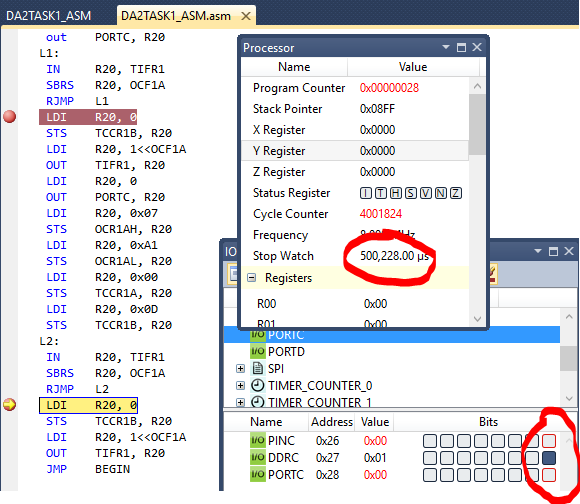
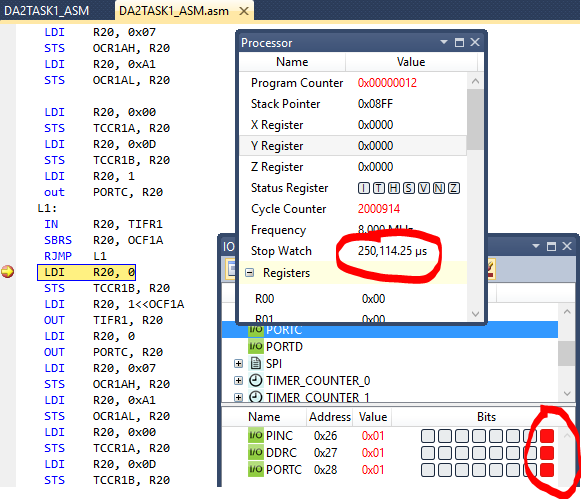
|  |  |  |  |
| --- | --- | --- | --- |
| 6. | SCHEMATICS |  |  |

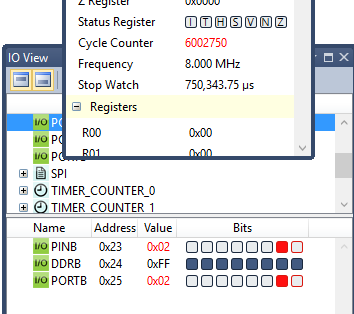
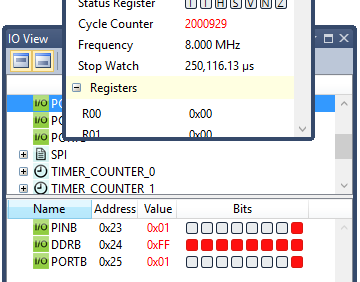
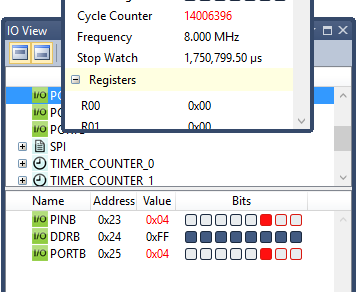
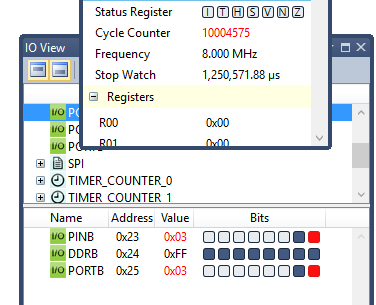


|  |  |  |  |
| --- | --- | --- | --- |
| 7. | SCREENSHOTS OF EACH TASK OUTPUT |  |  |

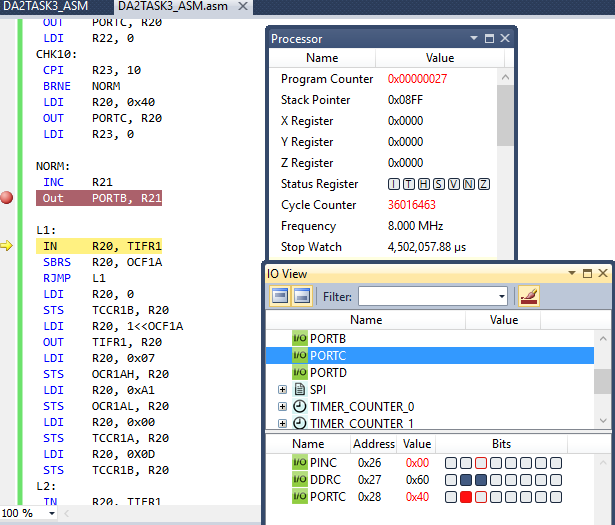
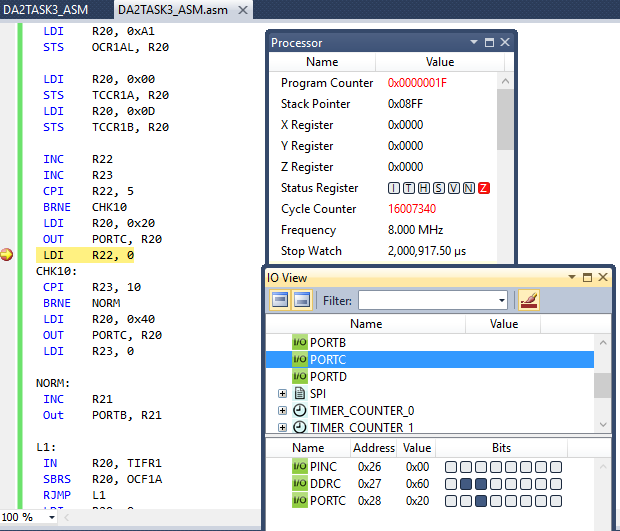
TASK 1:

50% duty cycle on PC0, period = 0.5 second   
The first image displays the rising edge of the waveform at ¼ second. PC0 is high. The second image shows the second half of the duty cycle where PC0 is low.

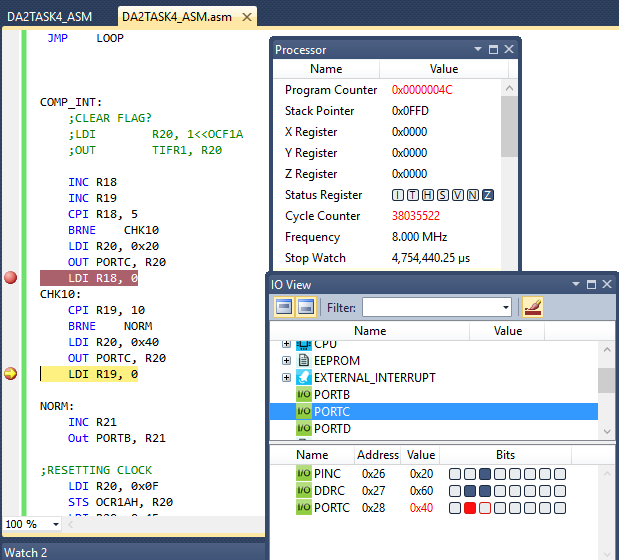
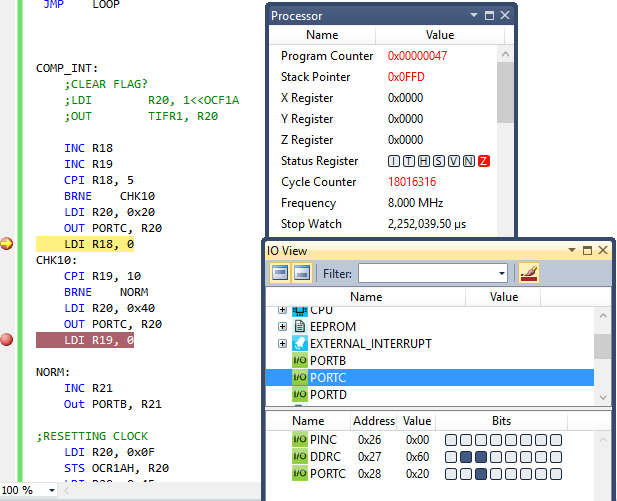


TASK 2:  
Images show PORTB outputting number of rising edges. With 50% DC of ½ sec period, each once happens every 500000us.  
  


TASK 3:  
In addition to Task 2’s duties, at every 5th and 10th rising edge, PC5 and PC6 will be toggled.

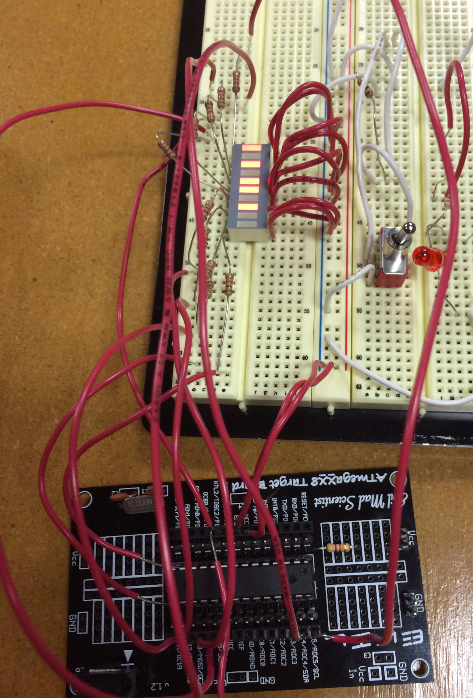


TASK 4  
Task 3, using interrupts.



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

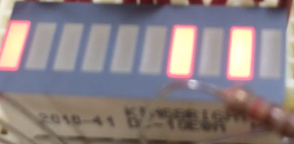
The red wired circuit is the one used for all experiments.



TASK 2: Counting number of rising edges, in 8 bit form on LED bar.

………..

TASK 3 / 4: Toggle 9th led every fifth and 10th led every tenth rising edge



|  |  |  |  |
| --- | --- | --- | --- |
| 9. | VIDEO LINKS OF EACH DEMO |  |  |
| https://www.youtube.com/watch?v=DXYXBecOVYg | | | |
| 10. | GOOGLECODE LINK OF THE DA |  |  |
| https://github.com/rodrie2/UNLVCPE301 | | | |

**Student Academic Misconduct Policy**

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

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

Emmanuel Rodriguez Lopez