Isaias Osorio

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 |  |  |

Atmel studio

LEDs

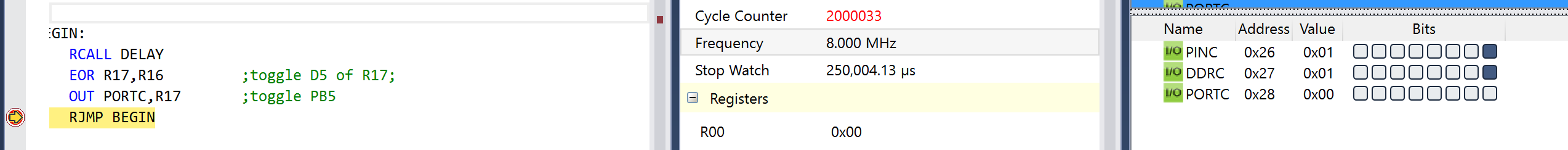
10Pin LED

Atmega328p

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

Design a delay subroutine to generate a waveform on PORTC.0 with 50% DC and 0.5 sec period.

Screen shot before running code:



; TASK 1 - ASM

;

; Created: 2/18/2016 8:07:11 PM

; Author : Isaias Osorio

;

SBI DDRC,0 ;PC0 as an output

LDI R17,0x01

LDI R16, 0x01

OUT PORTC,R17 ;PB5&4 = 0

BEGIN:

RCALL DELAY

EOR R17,R16 ;toggle D5 of R17;

OUT PORTC,R17 ;toggle PB5

RJMP BEGIN

DELAY:

LDI R20, HIGH(34286)

STS TCNT1H, R20 ;TCNT1H = 0xA4 timer1 high

LDI R20, LOW(34286)

STS TCNT1L, R20 ;TCNT1L = 0x72 timer1 low

LDI R20,0x00

STS TCCR1A, R20 ;WGM11:10 = 00

LDI R20,0x03

STS TCCR1B, R20 ;WGM13:12 = 05, Normal mode, prescaler = 1024

AGAIN:

IN R20,TIFR1 ;read TIFR

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

RJMP AGAIN

LDI R20,0x00

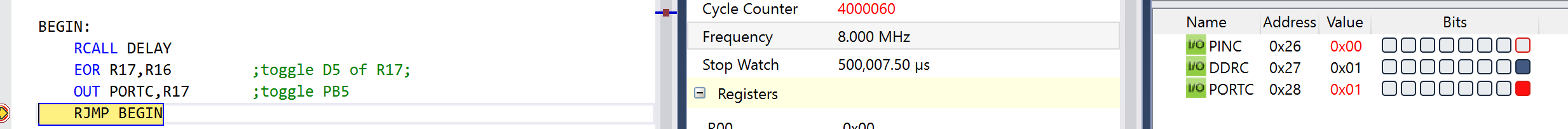
STS TCCR1B,R20 ;stop Timer1

LDI R20,0x01

OUT TIFR1, R20 ;clear TOV1 flag

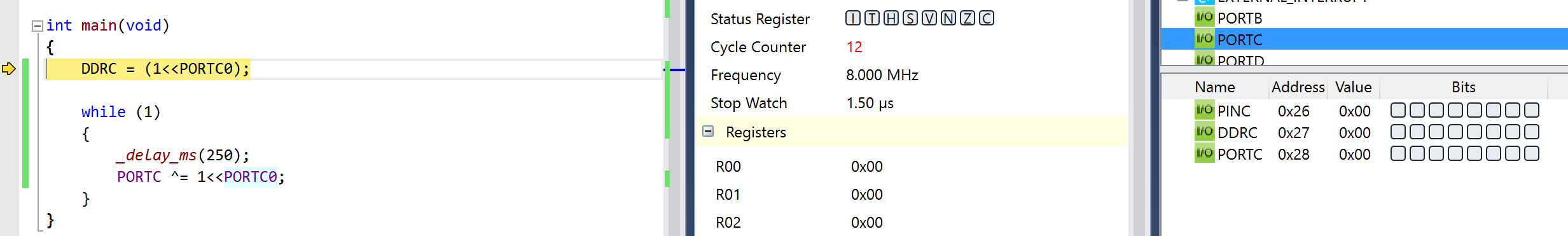
RET

Screen shot after running code:



In C

Screen shot before running code:



/\* TASK 1 - C

\* DA2T1C.c

\*

\* Created: 3/13/2016 10:14:17 PM

\* Author : Isaias Osorio

\*/

#define *F\_CPU* 8000000L //8MHz clock

#include <avr/io.h>

#include <util/delay.h> //for the use of delays

int main(void)

{

DDRC = (1<<PORTC0);

while (1)

{

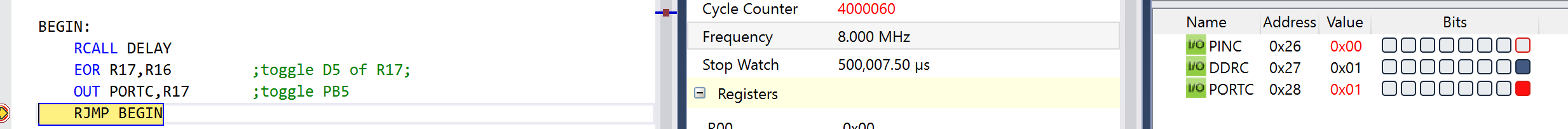
*\_delay\_ms*(250); // delay for 250us

PORTC ^= 1<<PORTC0;

}

}

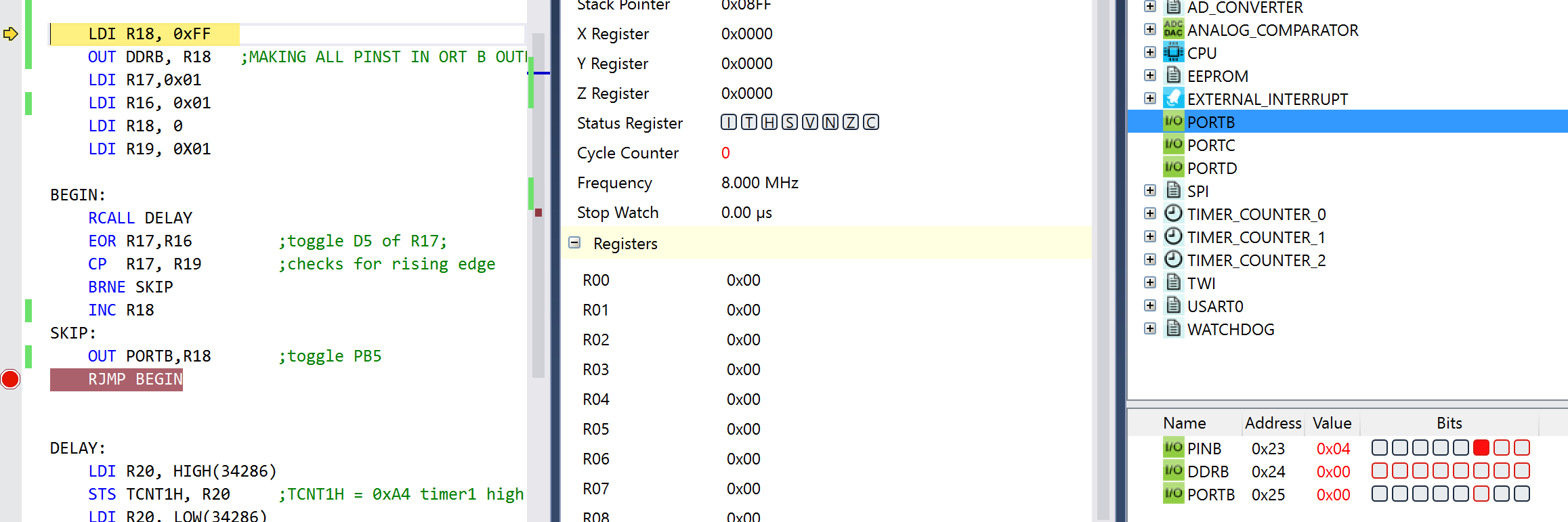
Screen shot after running code:



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

Implement a 8-bit counter to count on every rising edge of the above waveform. The state of the counter needs to be displayed (display 8 bits only) on a 10-bit LED bar connected to PORTB. Do not worry about the counter overflow.

Screen shot before running code:



; TASK 2 - ASM

;

; Author : Isaias Osorio

;

LDI R18, 0xFF

OUT DDRB, R18 ;MAKING ALL PINST IN ORT B OUTPUT

LDI R17,0x01

LDI R16, 0x01

LDI R18, 0

LDI R19, 0X01

BEGIN:

RCALL DELAY

EOR R17,R16 ;toggle D5 of R17;

CP R17, R19 ;checks for rising edge

BRNE SKIP

INC R18

SKIP:

OUT PORTB,R18 ;toggle PB5

RJMP BEGIN

DELAY:

LDI R20, HIGH(34286)

STS TCNT1H, R20 ;TCNT1H = 0xA4 timer1 high

LDI R20, LOW(34286)

STS TCNT1L, R20 ;TCNT1L = 0x72 timer1 low

LDI R20,0x00

STS TCCR1A, R20 ;WGM11:10 = 00

LDI R20,0x03

STS TCCR1B, R20 ;WGM13:12 = 05, Normal mode, prescaler = 1024

AGAIN:

IN R20,TIFR1 ;read TIFR

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

RJMP AGAIN

LDI R20,0x00

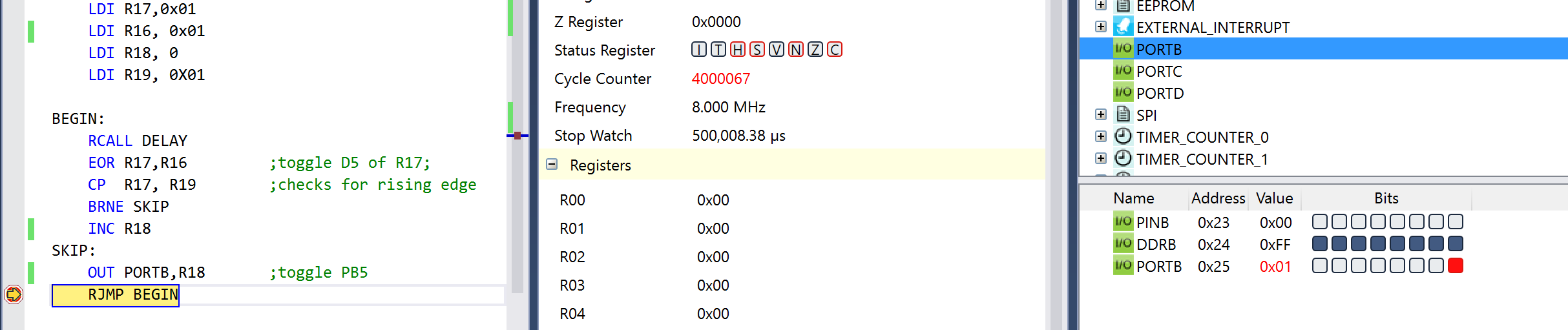
STS TCCR1B,R20 ;stop Timer1

LDI R20,0xFF

OUT TIFR1, R20 ;clear TOV1 flag

RET

Screenshot after running code:



In C

/\*

\* DA2T2C.c

\*

\* Created: 3/13/2016 10:46:48 PM

\* Author : Isaias Osorio

\* TASK 2 - C

\*/

#define *F\_CPU* 8000000L

#include <avr/io.h>

#include <util/delay.h>

int main(void)

{

unsigned char i = 0; //count of rising edge cycles

unsigned char j = (1<<PORTC0); //use to compare

DDRB = 0xFF;

while (1)

{

*\_delay\_ms*(250); // delay for 250us

j ^= 1<<PORTC0; // toggle

if(j == 1) // check if its rising edge

i++;

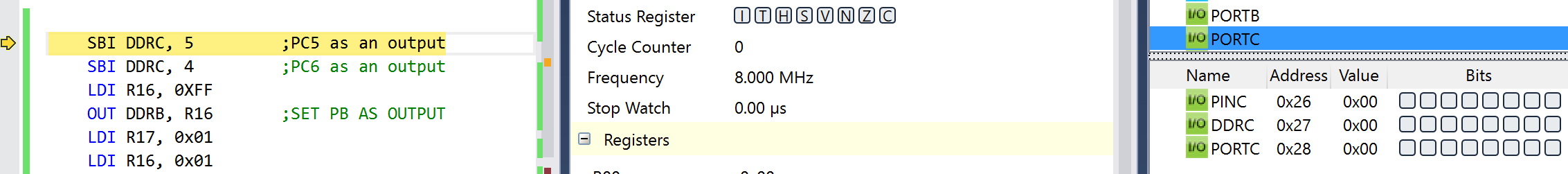
PORTB = i; // output to port b the count i

}

}

|  |  |  |  |
| --- | --- | --- | --- |
| 4. | INITIAL CODE OF TASK C |  |  |

Also connect the 9th and 10th bit of the LED bar to PORTC.5 and PORTC.6 pins. Toggle PORTC.5 and PORTC.6 for every 5th rising pulses and every 10th rising pulse of the counter respectively.

Screenshot before running code

; TASK 3 - ASM

; Created: 2/18/2016 8:07:11 PM

; Author : Isaias Osorio

;

SBI DDRC, 5 ;PC5 as an output

SBI DDRC, 4 ;PC6 as an output

LDI R16, 0XFF

OUT DDRB, R16 ;SET PB AS OUTPUT

LDI R17, 0x01

LDI R16, 0x01

LDI R18, 0 ;count on every rising edge

LDI R21, 0 ;count on every 5 rising edges

LDI R22, 0 ;count on every 10 rising edges

LDI R23, 0X00 ;use for to toggle pin 4

LDI R24, 0X00 ;used to toggle pin 5

BEGIN:

RCALL DELAY

EOR R17,R16 ;toggle D5 of R17;

CP R17, R16

BRNE SKIP

INC R18

INC R21 ;increment every5rising

INC R22 ;increment every10rising

CPI R21, 5 ;Checks to see if 5 rising edges have passed

BREQ FIVE

CHECKTEN:

CPI R22, 10 ;Checks to see if 10 rising edges have passed

BREQ TEN

SKIP:

OUT PORTB, R18 ;toggle PB5

RJMP BEGIN

FIVE:

LDI R21, 0

LDI R16, 0x10

IN R23, PORTC

EOR R23, R16

OUT PORTC,R23 ;toggle PB5

LDI R16, 0x01

RJMP CHECKTEN

TEN:

LDI R22, 0

LDI R16, 0x20

EOR R24, R16 ;TOGGLE PB4

OUT PORTC, R24

LDI R16, 0X01

RJMP SKIP

DELAY:

LDI R20, HIGH(34286)

STS TCNT1H, R20 ;TCNT1H = 0xA4 timer1 high

LDI R20, LOW(34286)

STS TCNT1L, R20 ;TCNT1L = 0x72 timer1 low

LDI R20,0x00

STS TCCR1A, R20 ;WGM11:10 = 00

LDI R20,0x03

STS TCCR1B, R20 ;WGM13:12 = 05, Normal mode, prescaler = 1024

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 R20,0xFF

OUT TIFR1, R20 ;clear TOV1 flag

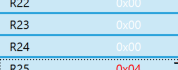
RET

Screenshot before running code

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

Use X/Y/Z register to parse through the 25 numbers and add all numbers divisible by 3 and place the result in R23:24. Parsing of the numbers for task b and c has to be done simultaneously.

Screenshot before running code



; Task c

; reset y register to x

movw yl, xl ; y <- x

ldi count, 25 ; counter = 25

ldi r17, 3 ; r17 = 3 // divisor

call Add25

mov r23, r10

mov r24, r11 ; r24:r23 <- r11:r10

Screenshot after running code

; TASK 3 - ASM

; Created: 2/18/2016 8:07:11 PM

; Author : Isaias Osorio

;

SBI DDRC, 5 ;PC5 as an output

SBI DDRC, 4 ;PC6 as an output

LDI R16, 0XFF

OUT DDRB, R16 ;SET PB AS OUTPUT

LDI R17, 0x01

LDI R16, 0x01

LDI R18, 0 ;count on every rising edge

LDI R21, 0 ;count on every 5 rising edges

LDI R22, 0 ;count on every 10 rising edges

LDI R23, 0X00 ;use for to toggle pin 4

LDI R24, 0X00 ;used to toggle pin 5

BEGIN:

RCALL DELAY

EOR R17,R16 ;toggle D5 of R17;

CP R17, R16

BRNE SKIP

INC R18

INC R21 ;increment every5rising

INC R22 ;increment every10rising

CPI R21, 5 ;Checks to see if 5 rising edges have passed

BREQ FIVE

CHECKTEN:

CPI R22, 10 ;Checks to see if 10 rising edges have passed

BREQ TEN

SKIP:

OUT PORTB, R18 ;toggle PB5

RJMP BEGIN

FIVE:

LDI R21, 0

LDI R16, 0x10

IN R23, PORTC

EOR R23, R16

OUT PORTC,R23 ;toggle PB5

LDI R16, 0x01

RJMP CHECKTEN

TEN:

LDI R22, 0

LDI R16, 0x20

EOR R24, R16 ;TOGGLE PB4

OUT PORTC, R24

LDI R16, 0X01

RJMP SKIP

DELAY:

LDI R20, HIGH(34286)

STS TCNT1H, R20 ;TCNT1H = 0xA4 timer1 high

LDI R20, LOW(34286)

STS TCNT1L, R20 ;TCNT1L = 0x72 timer1 low

LDI R20,0x00

STS TCCR1A, R20 ;WGM11:10 = 00

LDI R20,0x03

STS TCCR1B, R20 ;WGM13:12 = 05, Normal mode, prescaler = 1024

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

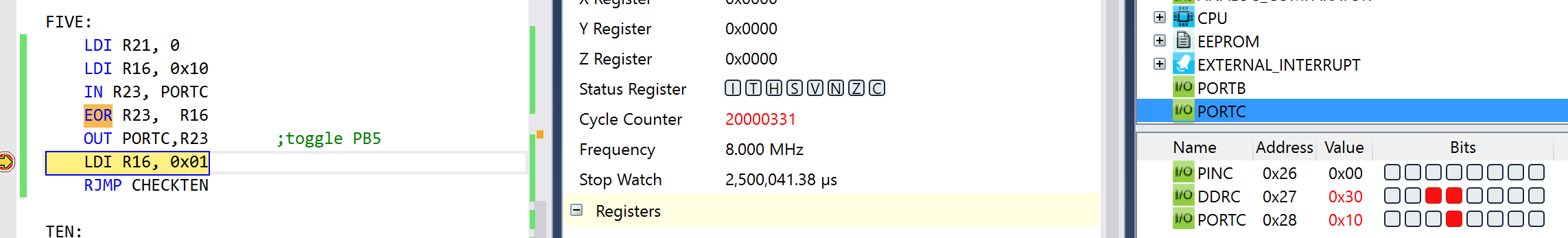
LDI R20,0xFF

OUT TIFR1, R20 ;clear TOV1 flag

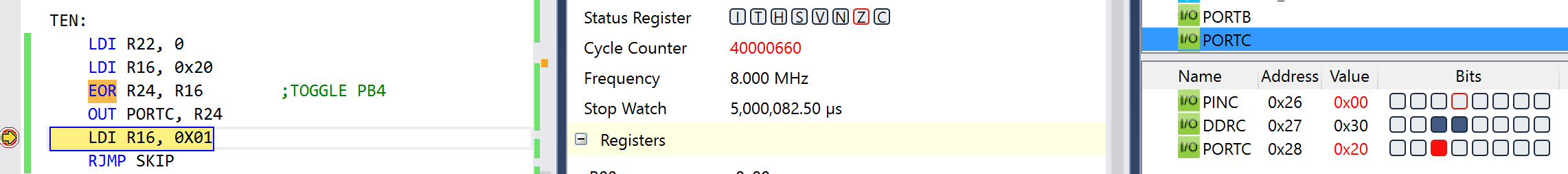
RET

Screenshot after running code

After every 5



After every 10



|  |  |  |  |
| --- | --- | --- | --- |
| 4. | Continuation CODE OF TASK D |  |  |

/\*

\* DA2T3C.c

\*

\* Created: 3/13/2016 10:58:54 PM

\* Author : Isaias Osorio

\* TASK 4 - C

\*/

#define *F\_CPU* 8000000L

#include <avr/io.h>

#include <avr/interrupt.h>

int counter; // global variable for counter

int multiple; // global variable for check counter (5th/10th)

ISR(TIMER1\_COMPA\_vect);

int main(void)

{

counter = 0; // initialize counter to zero

multiple = 0; // initialize check counter to zero

DDRB = 0xFF; // set all of PORTB as outputs

DDRC = 0x30; // set PC4 and PC5 as outputs

TCCR1B |= 0x0D; // setup up CTC mode and prescaler (1024) for timer

TCNT1 = 0; // initialize timer1 to zero

OCR1A = 3906; // set CTC value

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

sei(); // enable global interrupts

while (1) // loop forever and execute interrupt

{ // subroutine every 0.5 seconds

}

}

ISR(TIMER1\_COMPA\_vect) // interrupt subroutine body

{

counter++; // increment counter for LED bargraph

if(counter > 255) // if counter went above 255 (9th bit)

counter = 0; // reset counter back to zero

PORTB = counter; // output counter to LED bargraph

multiple++;

// increment counter for 5th/10th checks

switch(multiple)

{

case 5: // if first 5th rising edge

PORTC = 0x20; // PC4 = 0 and PC5 = 1

break;

case 10: // if first 10th rising edge

PORTC = 0x10; // PC4 = 1 and PC5 = 0

break;

case 15: // if second 5th rising edge

PORTC = 0x30; // PC4 = 1 and PC5 = 1

break;

case 20: // if second 10th rising edge

PORTC = 0x00; // PC4 = 0 and PC5 = 0

multiple = 0; // clear check counter

break;

default:

break;

}

}

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

n/a

|  |  |  |  |
| --- | --- | --- | --- |
| 7. | VIDEO LINKS OF EACH DEMO |  |  |
| https://drive.google.com/folderview?id=0B4H0WHqN3RMbZUtJeGtQcTFuWjA&usp=sharing | | | |
| 11. | Github Repository |  |  |
| https://github.com/isaiasmoioso/CPE-301.git | | | |

**Student Academic Misconduct Policy**

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