Derek Nhan

CPE301 – SPRING 2016

Design Assignment 4

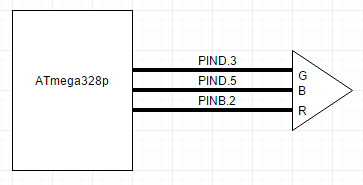
**DO NOT REMOVE THIS PAGE DURING SUBMISSION:**

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. | FLOWCHART |  |  |
| 2. | INITIAL CODE OF TASK 1/A |  |  |
| 3. | SCHEMATICS |  |  |
| 4. | SCREENSHOTS OF EACH TASK OUTPUT |  |  |
| 5. | SCREENSHOT OF EACH DEMO |  |  |
| 6. | VIDEO LINKS OF EACH DEMO |  |  |
| 7. | GOOGLECODE LINK OF THE DA |  |  |
| 8. |  |  |  |
| 9. |  |  |  |
| 10. |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

* RGB Led (common cathode)
* ATmega328p Xplained board
* Breadboard



|  |  |  |  |
| --- | --- | --- | --- |
| 1. | FLOWCHART |  |  |



|  |  |  |  |
| --- | --- | --- | --- |
| 2. | INITIAL CODE OF TASK 1/A |  |  |

#define *F\_CPU* 8000000UL

#include <avr/io.h>

#include <util/delay.h>

void wait(unsigned int us)

//subroutine to cause a time delay relative to the frequency

{

unsigned char n;

n=us/2;

while (n!=0)

{

n--;

*\_delay\_ms*(1);

}

}

int main()

{

DDRB=0xFF; //set PORTB as output

DDRD=0xFF; //set PORTD as output

TCCR0A=0x21; //Set OCR0B as I/O

TCCR0B=0x09; //Set OCR0A as TOP, prescaler=1

TCCR1A=0x23; //Set OCR1B as I/O, OCR1A as top

TCCR1B=0x11; //Set OCR1A as top, prescaler=1

TCCR2A=0x21; //Set OCR2B as I/O, OCR2A as top

TCCR2B=0x09; //Prescaler=1

while (1)

//loop forever

{

for (unsigned int a=25; a<=230; a+=25)

//for loop to control 10%-90% frequency for green led

{

OCR0A=a; //sets OCR0A as the top value, thus changing freqeuncy

for(unsigned int b=25; b<=230; b+=25)

//for loop to control 10%-90% frequency for blue led

{

OCR1A=b;

OCR1AH=b>>8; //sets OCR1A as the top value, thus changing frequency

for (unsigned int c=25; c<=230; c+=25)

//for loop to control 10%-90% frequency for red led

{

OCR2A=c; //sets OCR2A as the top value, thus changing frequency

for(unsigned int i=0; i<=a; i+=1)

//change duty cycle of green led at frequency a

{

OCR0B=i; //sets OCR0B of timer2 for phase correct pwm

for(unsigned char j=0; j<=b; j+=1)

//change duty cycle of blue led at frequency b

{

OCR1B=j; //sets OCR2B of timer1 for phase correct pwm

for(unsigned char k=0; k<=c; k+=1)

//change duty cycle of red led at frequency c

{

OCR2B=k; //sets OCR2B of timer2 for phase correct pwm

wait(a); //delay to allow user to see colors/changes

}

}

}

}

}

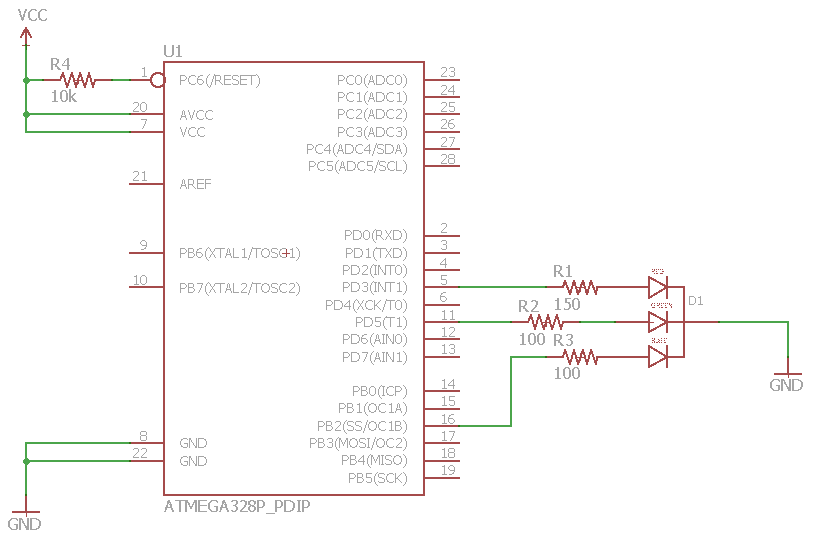
}

}

return 0;

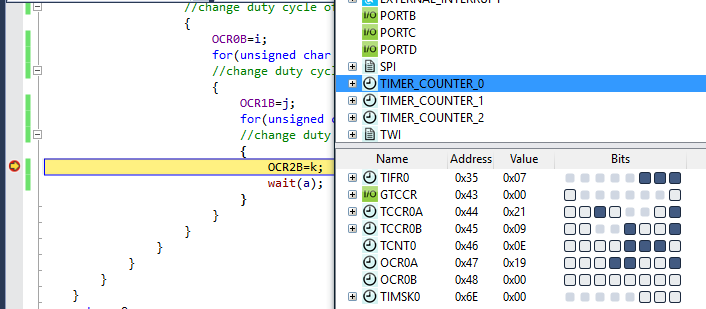
}

|  |  |  |  |
| --- | --- | --- | --- |
| 3. | SCHEMATICS |  |  |

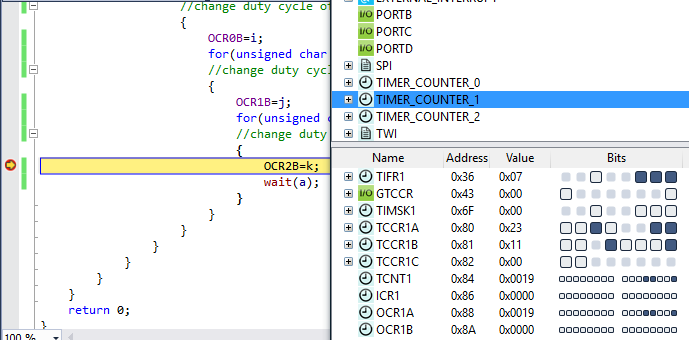


|  |  |  |  |
| --- | --- | --- | --- |
| 4. | SCREENSHOTS OF EACH TASK OUTPUT |  |  |

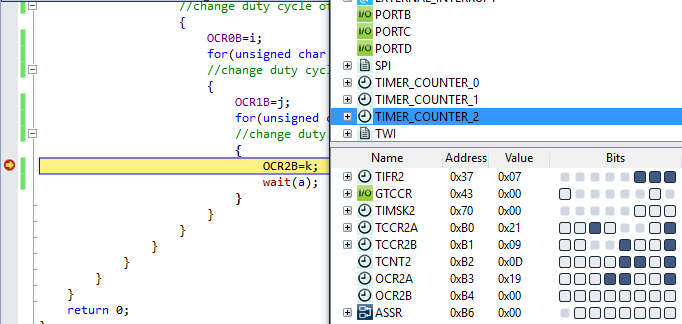
TASK 1/A:

Verified changing duty cycles  


: Timer0 set at 10% frequency



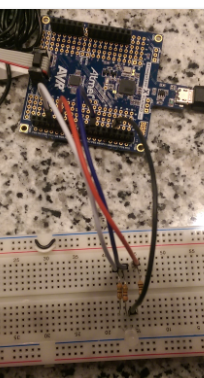
: Timer1 at 10% frequency



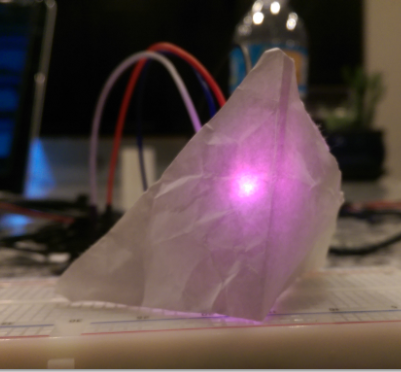
: Timer2 at 10% frequency

|  |  |  |  |
| --- | --- | --- | --- |
| 5. | SCREENSHOT OF EACH DEMO |  |  |

The circuit wired up to the ATmega328p and RGB led with resistors.



TASK 1/A: Changing frequency and duty cycles will cycle through all color variations of the RGB led. The paper is used as a diffused to decrease glare from LED and we can view color changes.



(see demo video)

|  |  |  |  |
| --- | --- | --- | --- |
| 6. | VIDEO LINKS OF EACH DEMO |  |  |
| https://youtu.be/Vlh2NEIWxqo | | | |
| 7. | GOOGLECODE LINK OF THE DA |  |  |
| https://github.com/nhand2/CPE301S16 | | | |

**Student Academic Misconduct Policy**

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

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

Derek Nhan