RALPH MAGO

CPE301 – SPRING 2016

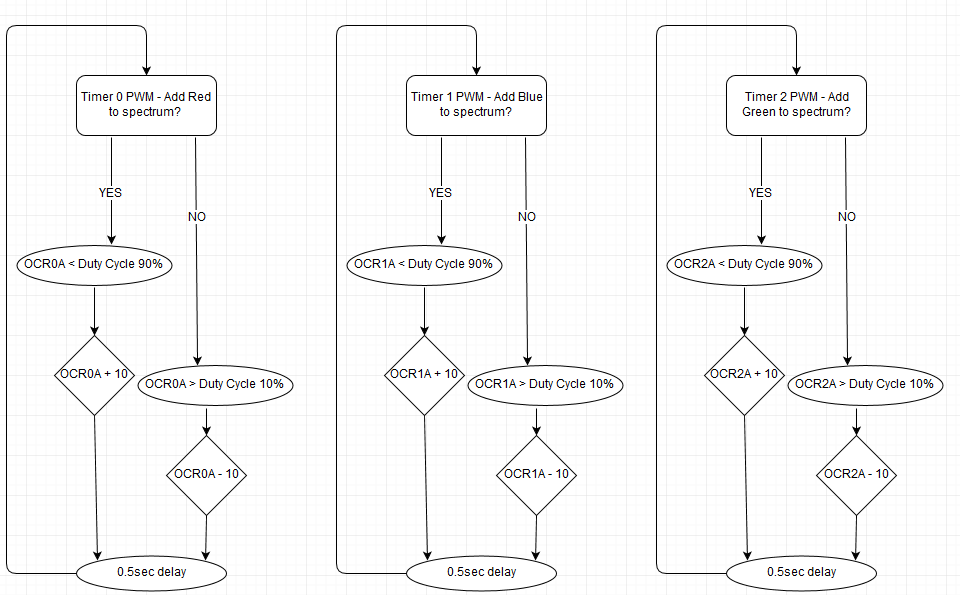
Design Assignment 4

**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. | FLOWCHART OF THE CODE | Y |  |
| 1. | COMPONENTS LIST | Y |  |
| 2. | INITIAL CODE OF TASK 1/A | Y |  |
| 3. | SCHEMATICS | Y |  |
| 4. | SCREENSHOTS OF EACH TASK OUTPUT | Y |  |
| 5. | SCREENSHOT OF EACH DEMO | Y |  |
| 6. | VIDEO LINKS OF EACH DEMO | Y |  |
| 7. | GITHUB LINK OF THE DA | Y |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 0. | FLOWCHART OF THE CODE |  |  |



|  |  |  |  |
| --- | --- | --- | --- |
| 1. | COMPONENTS LIST |  |  |

|  |  |
| --- | --- |
| Atmel Xplained Mini (328P) | 1 |
| Pololu #1074 Multicolor RGB LED | 1 |
| Resistor – 1KΩ | 1 |
| Resistor – 250Ω | 1 |
| Resistor - 500Ω | 1 |

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

/\*

\* DA4T1.c

\*

\* Created: 4/4/2016 7:24:44 PM

\* Author : r

\*/

//PINOUTS for RGB LED (flat side of LED facing north, pins from top to bottom)

// pin 1 = Red (connects to PD6 which is 0C0A pin)

// pin 2 = Ground (also connect this ground to AVR)

// pin 3 = Blue (connects to PB1 which is 0C1A pin)

// pin 4 = Green (connects to PB3 which is 0C2A pin)

//

// Timer 0 and Timer 2 are 8-bit timers, Timer 1 is a 16-bit timer

//

// Duty Cycle Calculation => 255 \* [% duty cycle]

// For example, if we want 90% duty cycle, we do 255 \* .90 = 230 or 0xE6

// and set that for the OCR registers

//TASK:

//1.) Write an AVR C program to generate three PWM signals to drive the RGB LED using

//TIMERs. Use the OCnX pins to generate the output. Increment individually each

//PWM period from min. (10%) to max (90%) value, at the same time alter the each

//PWM duty cycle. The RGB LED will display different colors as the PWM periods are

//changed and the brightness of the LED with vary with the change in duty cycle.

//2.) Use the delay subroutine to hold the colors for specific time period.

#define *F\_CPU* 16000000UL //16 MHz clock frequency simulation

#include <avr/io.h> //AVR input/output for PWM and LED output

#include <util/delay.h> //library for \_delay\_ms

#define DUTYHIGH 0xE6 //0xE6 = 90% Duty Cycle (FF \* .90)

#define DUTYLOW 0x19 //0x19 = 10% Duty Cycle (FF \* .10)

void red\_in(); //introduce red spectrum

void blue\_in(); //introduce blue spectrum

void green\_in(); //introduce green spectrum

void red\_out(); //decrease red spectrum

void blue\_out(); //decrease blue spectrum

void green\_out(); //decrease green spectrum

int main(void)

{

//PWM INITIALIZATION - 3 channels of PWM

DDRD |= (1<<PD6); //Red - PD6 - OC0A (Output Compare A)

DDRB |= (1<<PB1) | (1<<PB3); //Blue - PB1 - 0C1A (Output Compare B), Green - PB3 - OC2A (Output Compare C)

TCCR0A |= (1<<COM0A1) | (1<<WGM01) | (1<<WGM00); //T0 Clear mode set, WGM01=1 and WGM00=1 => Fast PWM

TCCR1A |= (1<<COM1A1) | (1<<WGM10); //T1 Clear mode set, WGM10=1 and

TCCR1B |= (1<<WGM12); //WGM12=1 => Fast PWM(8bit)

TCCR2A |= (1<<COM2A1) | (1<<WGM21) | (1<<WGM20); //T2 Clear mode set, WGM21=1 and WGM20=1 => Fast PWM

TCCR0B |= (1<<CS00); //T0 No Prescaling

TCCR1B |= (1<<CS10); //T1 No Prescaling

TCCR2B |= (1<<CS20); //T2 No Prescaling

OCR0A = DUTYLOW; //Red on 10% DC

OCR1A = DUTYLOW; //Blue on 10% DC

OCR2A = DUTYLOW; //Green on 10% DC

while(1)

{

red\_in(); //increment DC T0

green\_in(); //increment DC T2

blue\_in(); //increment DC T1

red\_out(); //decrement DC T0

blue\_out(); //decrement DC T1

red\_in(); //increment DC T0

blue\_in(); //increment DC T1

green\_out(); //decrement DC T2

red\_out(); //decrement DC T0

blue\_out(); //decrement DC T1

}

}

void red\_in()

{

while(OCR0A < DUTYHIGH){ //when timer counter less than 90% DC

OCR0A = OCR0A + 0xF; //increase Duty Cycle by 10

*\_delay\_ms*(50);} //0.5 sec delay hold

}

void blue\_in()

{

while(OCR2A < DUTYHIGH){ //when timer counter less than 90% DC

OCR2A = OCR2A + 0xF; //increase Duty Cycle by 10

*\_delay\_ms*(50);} //0.5 sec delay hold

}

void green\_in()

{

while(OCR1A < DUTYHIGH){ //when timer counter less than 90% DC

OCR1A = OCR1A + 0xF; //increase Duty Cycle by 10

*\_delay\_ms*(50);} //0.5 sec delay hold

}

void red\_out()

{

while(OCR0A > DUTYLOW){ // when timer counter higher than 10% DC

OCR0A = OCR0A - 0xF; //decrease Duty Cycle by 10

*\_delay\_ms*(50);} //0.5 sec delay hold

}

void blue\_out()

{

while(OCR2A > DUTYLOW){ // when timer counter higher than 10% DC

OCR2A = OCR2A - 0xF; //decrease Duty Cycle by 10

*\_delay\_ms*(50);} //0.5 sec delay hold

}

void green\_out()

{

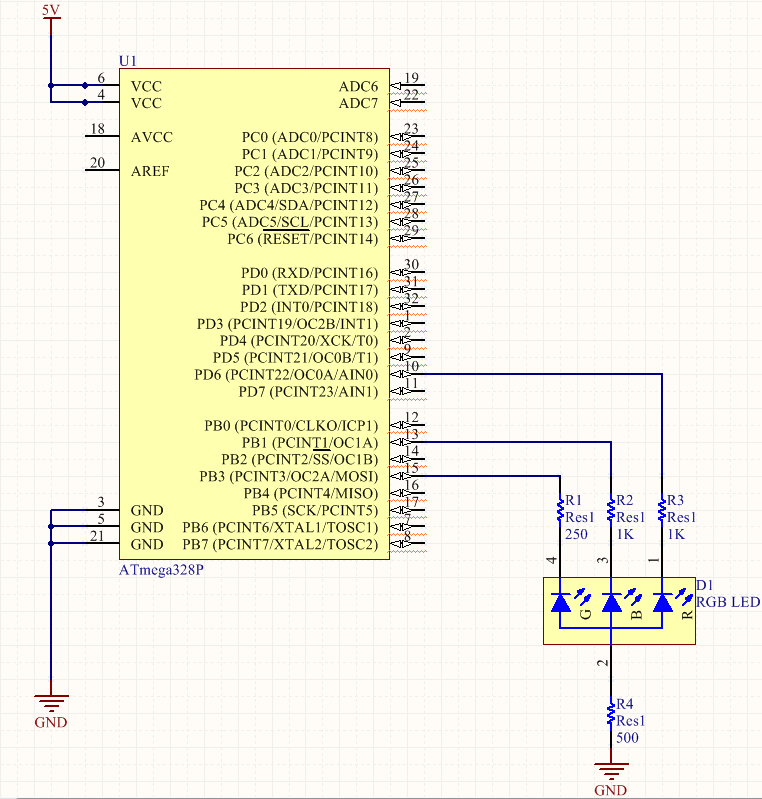
while(OCR1A > DUTYLOW){ // when timer counter higher than 10% DC

OCR1A = OCR1A - 0xF; //decrease Duty Cycle by 10

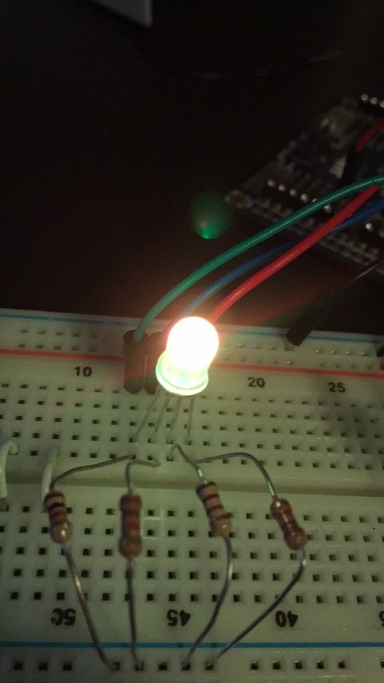
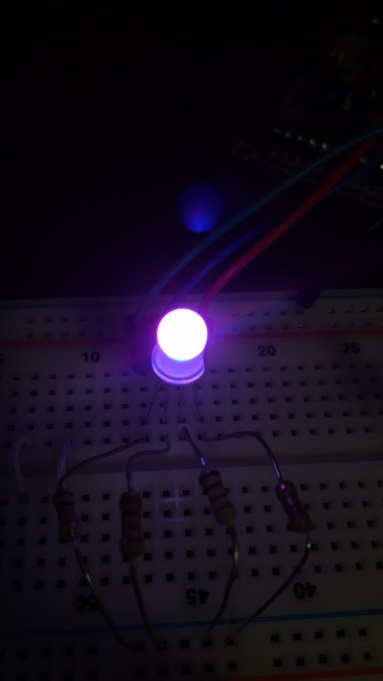
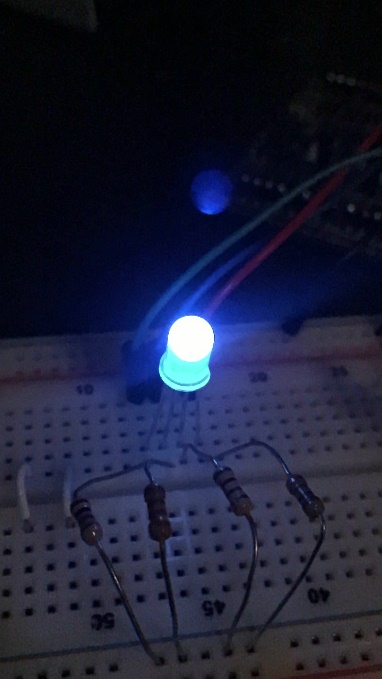
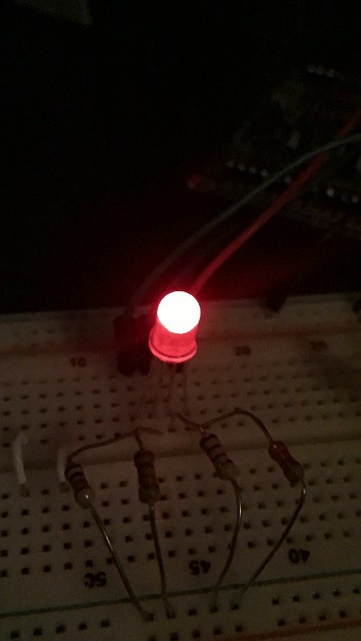
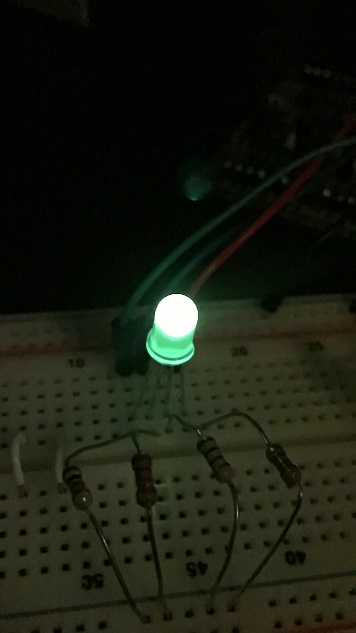
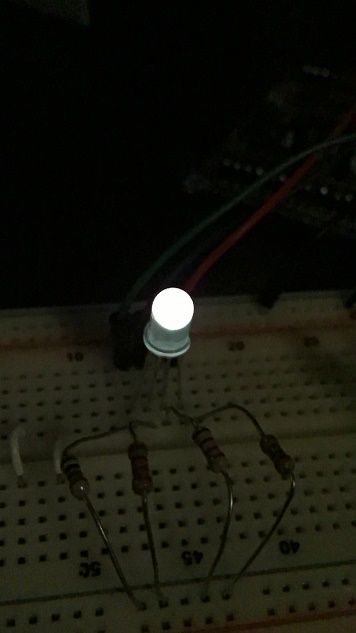
*\_delay\_ms*(50);} //0.5 sec delay hold

}

|  |  |  |  |
| --- | --- | --- | --- |
| 3. | SCHEMATIC |  |  |

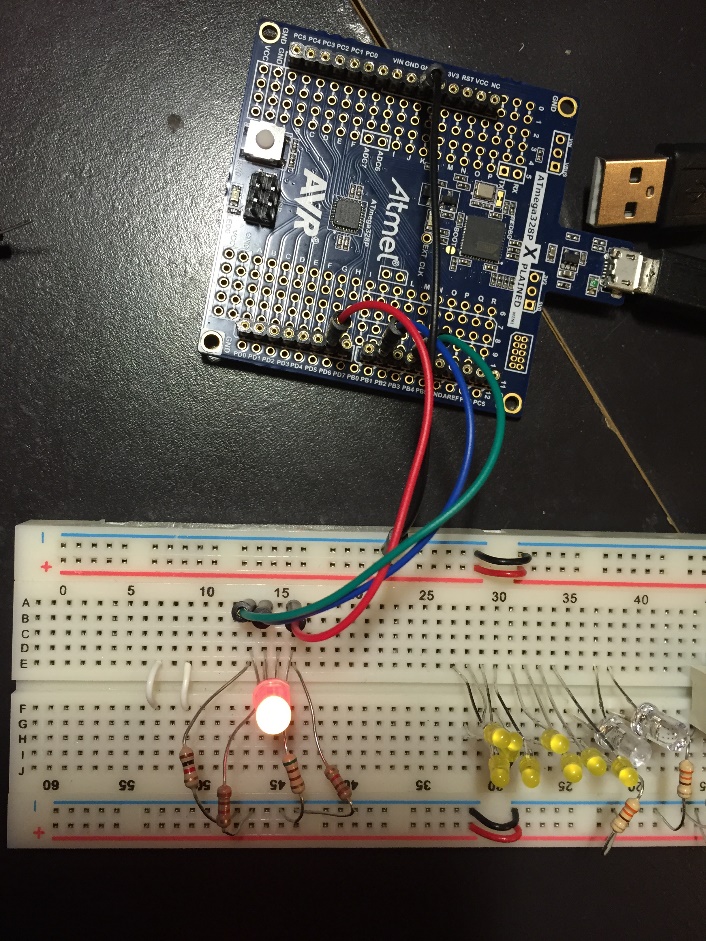
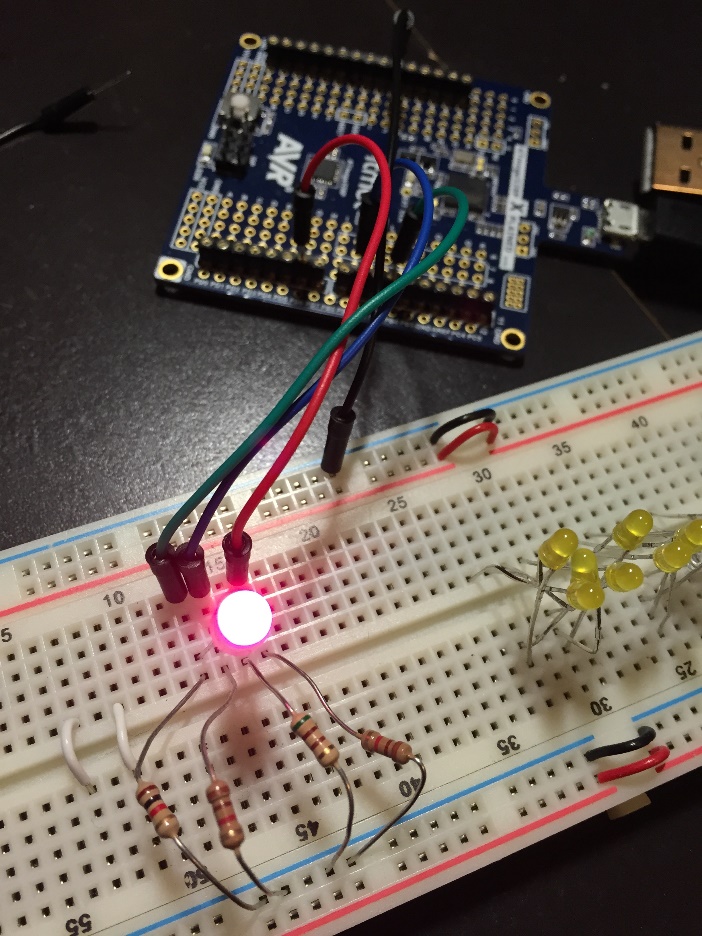


|  |  |  |  |
| --- | --- | --- | --- |
| 4. | SCREENSHOTS OF EACH TASK OUTPUT |  |  |



Displayed are various colors displayed from the RGB LED. Sample colors shown here are white, green, red, blue, purple, and yellow. The full spectrum of the LED is better illustrated in the video demonstration.

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

Breadboard, Atmel Xplained Mini 328P, and RGB LED Connections

|  |  |  |  |
| --- | --- | --- | --- |
| 6. | VIDEO LINK OF DEMO |  |  |
|  | https://www.youtube.com/watch?v=AceEPT3jkeA |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 7. | GITHUB LINK OF THE DA |  |  |
|  | https://github.com/magor1/embedded-design-VM/tree/master/DA4 |  |  |

**Student Academic Misconduct Policy**

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

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

RALPH MAGO