Martin Jaime-Viveros

CPE301 – SPRING 2016

Design Assignment 04

**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 |  |  |

* RGB LED
* Atmega328P
* Atmel Studio 7

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

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.

/\*

\* DA4T1.c

\*

\* Created: 4/10/2016 13:11:34

\* Author : Martin

\*/

#define *F\_CPU* 8000000UL //XTAL = 8 MHz

#include <avr/io.h>

#include <util/delay.h>

#define BOTTOM 0x19 // 8-bit OCR Value for 10% duty cycle

#define TOP 0xE5 // 8-bit OCR Value for 90% duty cycle

#define OFFSET 0

#define DELAY\_MS 5

#define STEP 1 // must be 1

// fade output compare register to control red's pwm signal.

void fadeRed(unsigned char, unsigned char, char);

// fade output compare register to control green's pwm signal.

void fadeGreen(unsigned char, unsigned char, char);

// fade output compare register to control blue's pwm signal.

void fadeBlue(unsigned char, unsigned char, char);

int main ()

{

// Set output compare pins as output.

DDRD = (1<<DDD5) | (1<<DDD6);

DDRB = (1<<DDB3);

// Set up timer0 for colors Red and Green

// Phase correct, clear on compare match when up counting,

// and set on compare match when down counting.

// set phase correct mode.

TCCR0A = (1<<COM0A1) | (0<<COM0A0) | (1<<COM0B1) | (0<<COM0B0) | (1<<WGM00);

TCCR0B = (1<<CS01) | (0<<CS00); // prescaler of 64

// Set up timer2 for color Blue

// Same configuration as timer0

TCCR2A = (1<<COM2A1) | (0<<COM2A0) | (1<<WGM20);

TCCR2B = (1<<CS22); // prescaler of 64

// Initialize output compare registers to BOTTOM value for 10% duty cycle

OCR0A = BOTTOM; // RED

OCR0B = BOTTOM; // GREEN

OCR2A = BOTTOM; // BLUE

while (1)

{

// rgb values traversed as 3 bit grey code

// Increment blue pwm signal 0 0 1

fadeBlue(BOTTOM, TOP, STEP);

// Increment green pwm signal 0 1 1

fadeGreen(BOTTOM, TOP, STEP);

// decrement blue pwm signal 0 1 0

fadeGreen(TOP, BOTTOM, -STEP);

// Increment red pwm signal 1 1 0

fadeRed(BOTTOM, TOP, STEP);

// Increment blue pwm signal 1 1 1

fadeBlue(BOTTOM, TOP, STEP);

// Decrement green pwm signal 1 0 1

fadeGreen(TOP, BOTTOM, -STEP);

// Decrement blue pwm signal 1 0 0

fadeBlue(TOP, BOTTOM, -STEP);

// Decrement red pwm signal 0 0 0

fadeRed(TOP, BOTTOM, -STEP);

}

return 0;

}

void fadeRed(unsigned char start, unsigned char stop, char step)

{

for (OCR2A = start; OCR2A != stop; OCR2A += step)

{

*\_delay\_ms*(DELAY\_MS);

}

}

void fadeGreen(unsigned char start, unsigned char stop, char step)

{

for (OCR0A = start; OCR0A != stop; OCR0A += step)

{

*\_delay\_ms*(DELAY\_MS);

}

}

void fadeBlue(unsigned char start, unsigned char stop, char step)

{

for (OCR0B = start; OCR0B != stop; OCR0B += step)

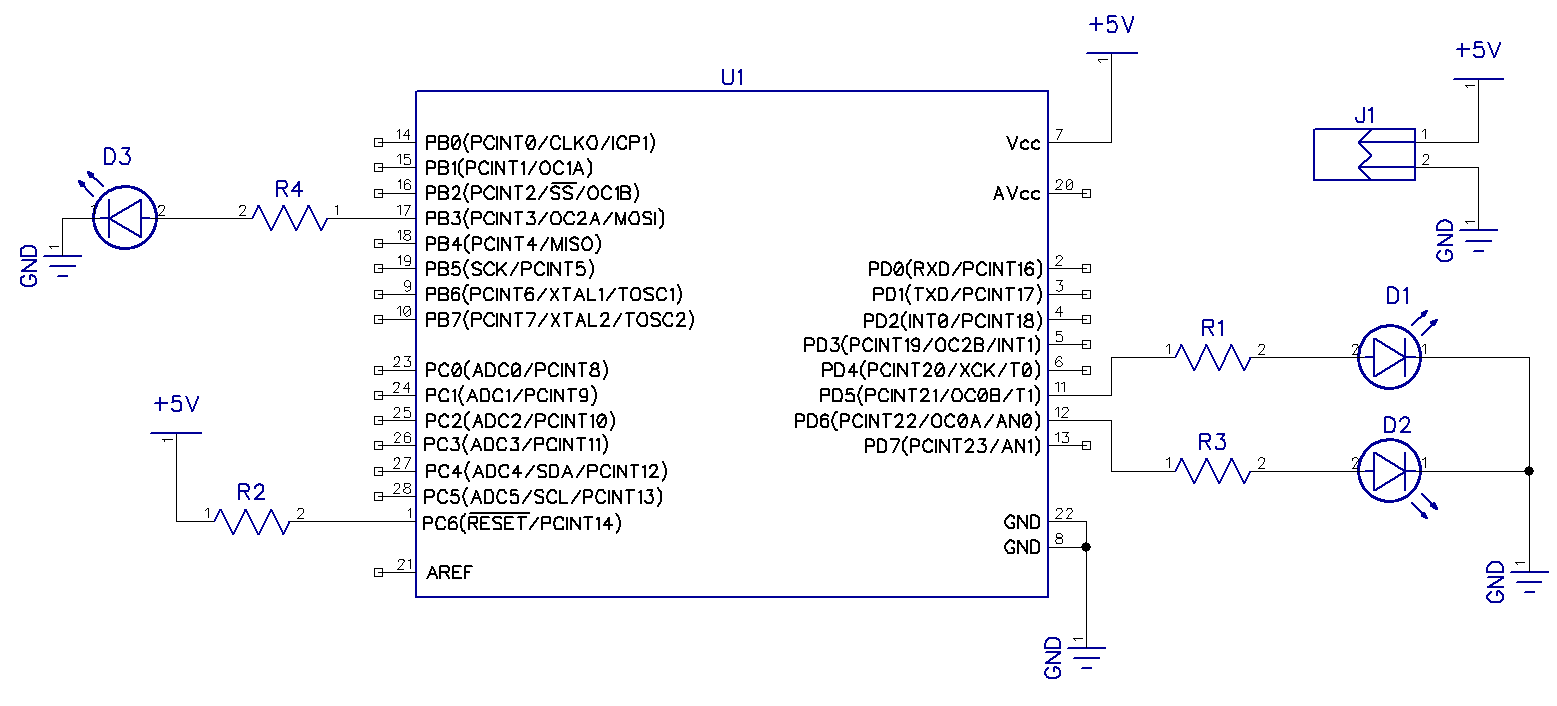
{

*\_delay\_ms*(DELAY\_MS);

}

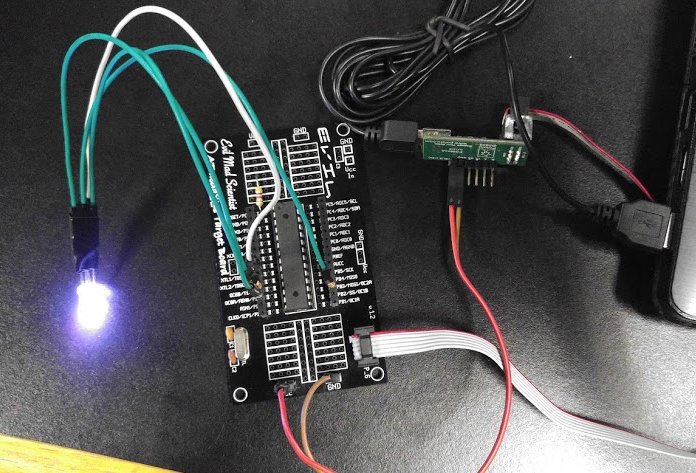
}

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



|  |  |  |  |
| --- | --- | --- | --- |
| 8. | SCREENSHOT OF TASK 1 |  |  |

TASK 1:



Colors are iterated as three-bit binary grey code.

|  |  |  |  |
| --- | --- | --- | --- |
| 9. | VIDEO LINKS OF EACH DEMO |  |  |
| https://drive.google.com/folderview?id=0B4ItVBjMqlnyZ2Znd2YtWEdzdjg&usp=sharing | | | |
| 10. | Github LINK OF THE DA |  |  |
| https://github.com/martinjaime/CpE301\_Assignments2016S.git | | | |

**Student Academic Misconduct Policy**

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

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

Martin Jaime-Viveros