Desi Battle

CPE301 – SPRING 2016

Design Assignment 4

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

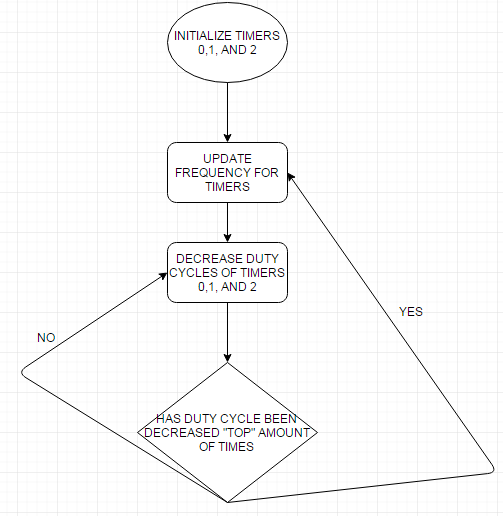
The student understands that all required components should be submitted in complete for grading of this assignment.

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **SUBMISSION ITEM** | **COMPLETED (Y/N)** | **MARKS**  **(/MAX)** |
| 0. | COMPONENTS LIST AND Flowchart |  |  |
| 1. | INITIAL CODE OF TASK 1/A |  |  |
| 2. | SCHEMATICS |  |  |
| 4. | SCREENSHOT OF EACH DEMO |  |  |
| 5. | VIDEO LINKS OF EACH DEMO |  |  |
| 6. | GOOGLECODE LINK OF THE DA |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 0. | COMPONENTS LIST AND FlowChart |  |  |

RGB LED

Atmega328P



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

Note: I tried several methods to implement the assignment and found this to be the most aesthetically appealing solution.

/\*

\* DA4.c

\*

\* Created: 4/8/2016 7:01:29 PM

\* Author: battled

\*/

#define F\_CPU 8000000UL

#include <util/delay.h>

#include <avr/io.h>

//delay constant

#define DEL 25

int main(void)

{

//configure timer0; set up fast pwm TOP = OCRA;

//inverting mode OC0B (set on match, clear at BOTTOM)

//use OCRA for frequency changes, OCRB for duty cycle

// the x array is created to shift colors starting dc for

// each frequency(more explanation below)

int x[3];

int i=0;

TCCR0A = (0<<COM0A1)|(0<<COM0A0)|(1<<COM0B1)|(1<<COM0B0)|(1<<WGM01) | (1<<WGM00);

TCCR0B = (0<<FOC0A )|(0<<FOC0B) |(1<<WGM02) |(0<<CS02) |(0<<CS01) | (1<<CS00);

//configure timer1 the same as timer 0

TCCR1A = (0<<COM1A1)|(0<<COM1A0)|(1<<COM1B1)|(1<<COM1B0)|(1<<WGM11)|(1<<WGM10);

TCCR1B = (0<<ICNC1) |(0<<ICES1) |(1<<WGM13) |(1<<WGM12) |(0<<CS12) |(0<< CS11)|(1<<CS10);

//configure timer2 identically to timer 0

TCCR2A = (0<<COM2A1)|(0<<COM2A0)|(1<<COM2B1)|(1<<COM2B0)|(1<<WGM21) | (1<<WGM20);

TCCR2B = (0<<FOC2A )|(0<<FOC2B) |(1<<WGM22) |(0<<CS22) |(0<<CS21) | (1<<CS20);

//SET PORT D AND B AS OUTPUTS

DDRB=0XFF;

DDRD=0xFF;

while(1)

{

//start at minimum frequency and gradually decrease

//frequency for all three colors

for(int freq=225; freq > 25; freq -= 25)

{

//incremented to shift which color has the dominant duty

//cycle for the next frequency

i++;

int freq=225;

\_delay\_ms(DEL);

OCR0A = freq;

OCR1A = freq;

OCR2A = freq;

// step up the duty cycle for each color

// they increment at the same rate but

// duty cycle starts 1/3\*freq apart from each other

// (100%, 66%, and 33%)

for(int dc=0; dc <= freq; dc++)

{

x[0] = freq/3;

x[1] = 2\*freq/3;

x[2] = 0;

\_delay\_ms(DEL);

//increment the duty cycles and reset them to their

//maximum when they get to 0% dc

//add increasing prime numbers to allow mixture of

//all possible duty cycles.

OCR0B +=2; //BLUE

if (OCR0B >=freq)

OCR0B= x[(i+1)%3];

OCR1B +=3; //GREEN

if (OCR1B >=freq)

OCR1B=x[(i+2)%3];

OCR2B +=5; //RED

if (OCR2B >=freq)

OCR2B=x[i%3];

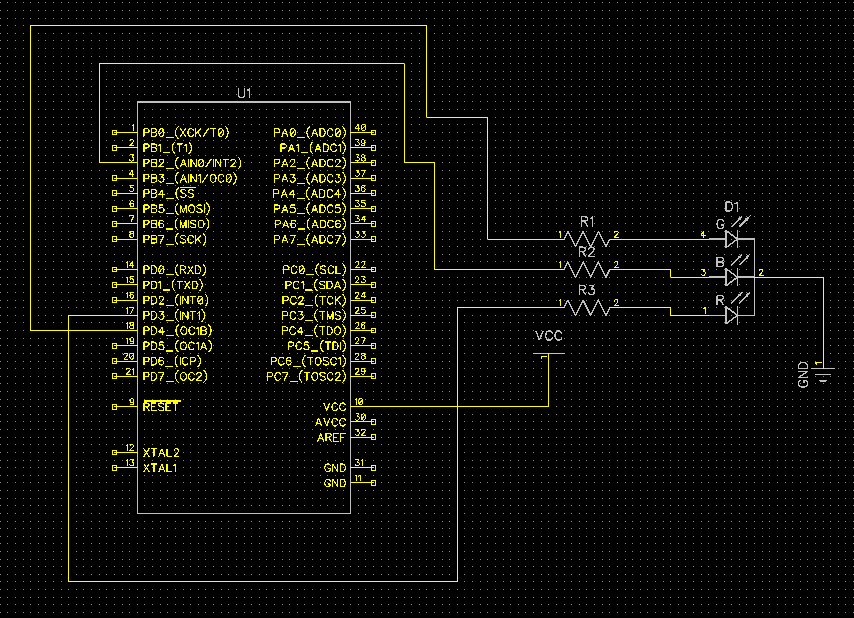
}

}

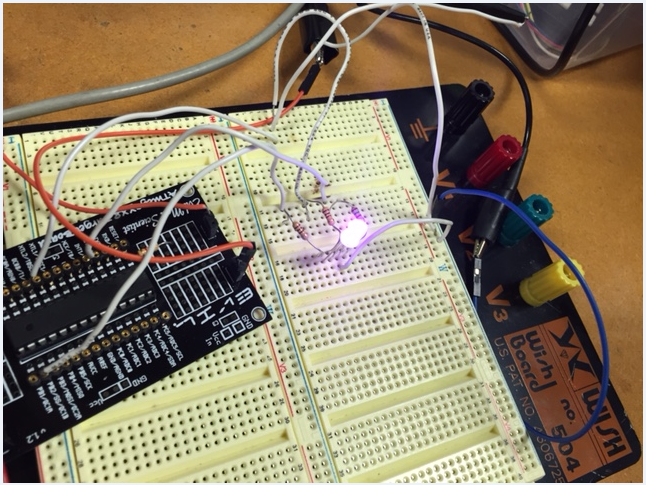
}

}

|  |  |  |  |
| --- | --- | --- | --- |
| 2. | SCHEMATICS |  |  |



|  |  |  |  |
| --- | --- | --- | --- |
| 3. | SCREENSHOT OF EACH DEMO |  |  |



|  |  |  |  |
| --- | --- | --- | --- |
| 4. | VIDEO LINKS OF EACH DEMO |  |  |
| https://www.youtube.com/watch?v=hmncq39LFg0 | | | |
| 5. | GOOGLECODE LINK OF THE DA |  |  |
| https://github.com/battled/DA1.git | | | |

**Student Academic Misconduct Policy**

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

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

Desi Battle