UMass Dartmouth ECE 263: Embedded System Design

Lab 2: USART and PWM

August 18 2022

We certify that this work is original
and not a product of anyone's work but our own

X__________________________	X__________________________
Sai Abhishek Bhattiprolu	Ayyappan Rajesh

Contents

1 Introduction

In this lab, students will using a “dev-board” or “shield” which attaches to the xPlain 328pb board. On powerup, the led should be turned on in white and a connection to the PuTTy should be established asking the user for input to change color on the led.

2 Abstract

The lab utilizes a Seven Segment add-on that has a tri-color led built in which is placed on top of the 328pb board, a similar lab had been done as part of ECE160 using the same board, although, the method in which it was programmed then was different to this lab.

3 Methods/Procedure

System Under Test: A 328pb board and a shield.

Hardware Needed:

- ATmega 328pb board

- Micro-USB Cable

- Dev-Board/Sheild with a tri color LED

Software Needed:

- ATmel Studio

- PuTTY

1) Mount the shield on top of the ATMega Board.

2) Connect the ATMega board to the computer.

3) Install PuTTY on the computer.

4) Verify that there are no errors in the code written and also include the mBuffer files for the code to run.

5) Check for the comm port of the 328pb board through device manager.

6) Run PuTTY, select serial and input the comm port.

7) Debug and run the code.

8) The PuTTY terminal should now ask for user inputs to display the colors. Input the desired color options.

9) The shield should now display the desired color.

10) The color options may be changed by the user if desired.

4 Experimental Results

The results worked as expected, the functions work, and the led is well lit with the desired colors. The commands and values have been properly sent to the board and the led lights up accordingly. The putty displays the option/value set to the shield once the user inputs "s" into the terminal.

5 Discussion of Important Results

This lab mainly focused on the dev-board and the objective was to to program the code to control rgb led on the board. The student will first need to use the functions given in the handout to initialize the PWM and set the led values.

The InitLED() will be called once to initialize all the timers for PWM mode. The SetLED() will be called multiple times in the code to set the values on the shield. The values of the duty cycles are as follows:

-> 0 for 0 percent duty cycle.

-> 128 for 50 percent duty cycle.

-> 255 for 100 percent duty cycle.

The respective LED is turned on when their respective pin is set to low. Which also means that the led is turned off when its duty cycle is 100 percent. From this, we can also understand that the led will be turned on when the duty cycle is 0 percent. The key to the options on the PuTTY Termimal are as follows:

1) "?" : Displays the list of applicable commands

2) "R" : Sets the value to Red

3) "G" : Sets the value to Green

4) "B" : Sets the value to Blue

5) "W" : This will write the value to led

6) "S" : Displays the color of the led that is being lit

6 Conclusion and Connection with Theory

The program had run successfully and had displayed the desired colors, performed the task successfully. The program was written using AVR in C and various functions were being used. The idea to gain knowledge about the Dev Board has been successfully implemented.

7 Appendices

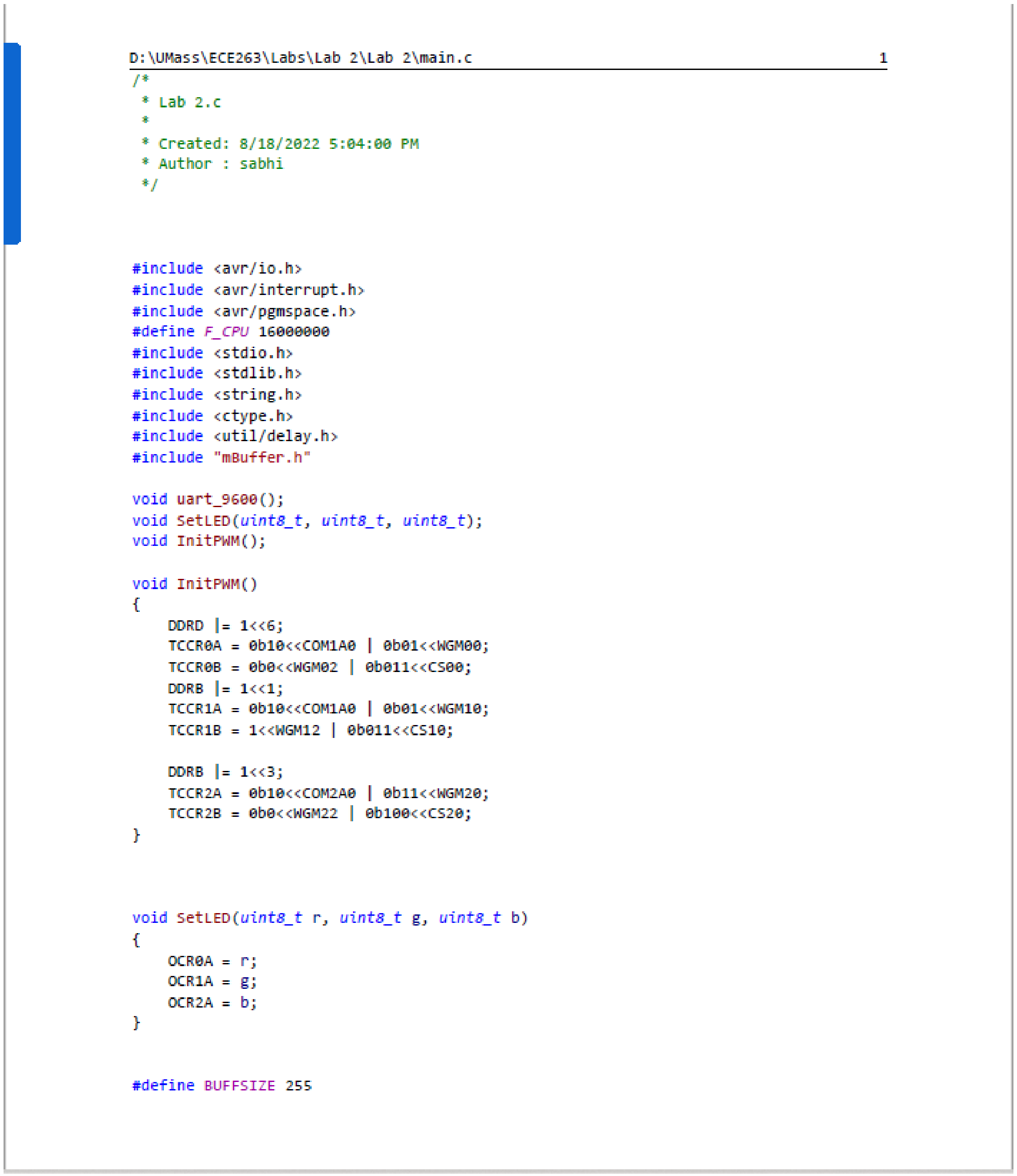
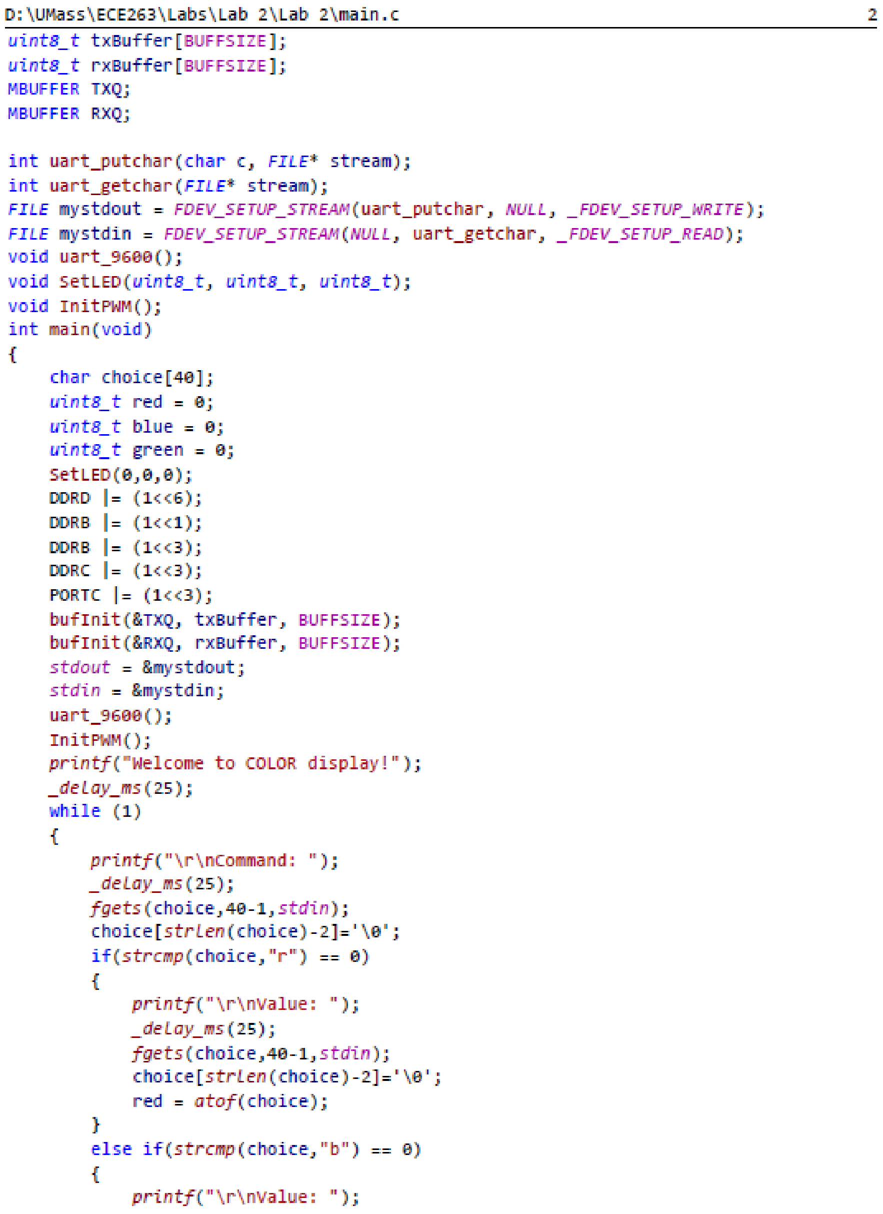
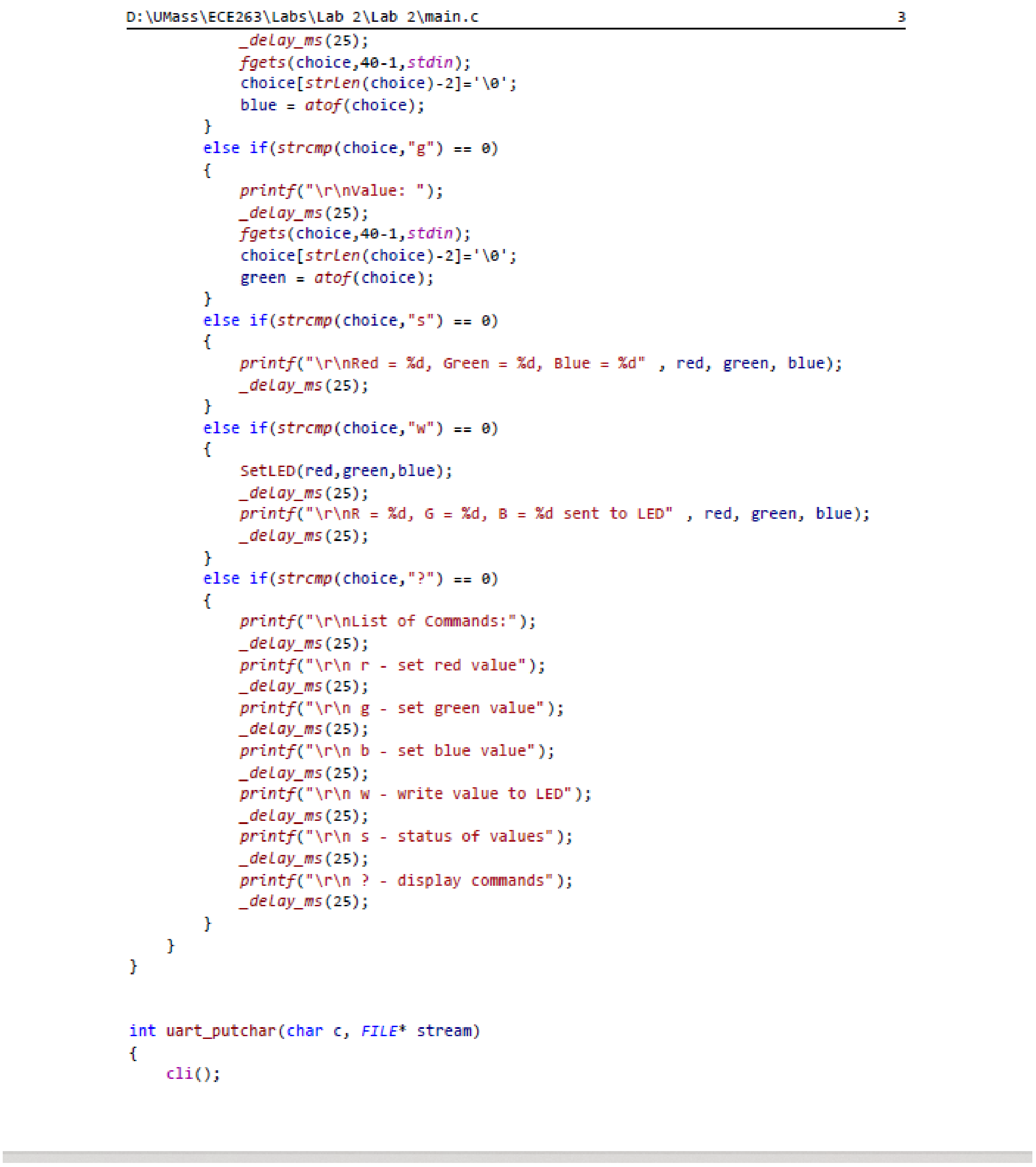

```
D:\UMass\ECE263\Labs\Lab 2\Lab 2\main.c                                          1
/*
 * Lab 2.c
 *
 * Created: 8/18/2022 5:04:00 PM
 * Author : sabhi
 */



#include <avr/io.h>
#include <avr/interrupt.h>
#include <avr/pgmspace.h>
#define F_CPU 16000000
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include <util/delay.h>
#include "mBuffer.h"

void uart_9600();
void SetLED(uint8_t, uint8_t, uint8_t);
void InitPWM();

void InitPWM()
{
    DDRD |= 1<<6;
    TCCR0A = 0b10<<COM1A0 | 0b01<<WGM00;
    TCCR0B = 0b0<<WGM02 | 0b011<<CS00;
    DDRB |= 1<<1;
    TCCR1A = 0b10<<COM1A0 | 0b01<<WGM10;
    TCCR1B = 1<<WGM12 | 0b011<<CS10;

    DDRB |= 1<<3;
    TCCR2A = 0b10<<COM2A0 | 0b11<<WGM20;
    TCCR2B = 0b0<<WGM22 | 0b100<<CS20;
}


void SetLED(uint8_t r, uint8_t g, uint8_t b)
{
    OCR0A = r;
    OCR1A = g;
    OCR2A = b;
}


#define BUFFSIZE 255
```

Figure 1: Code Part-1



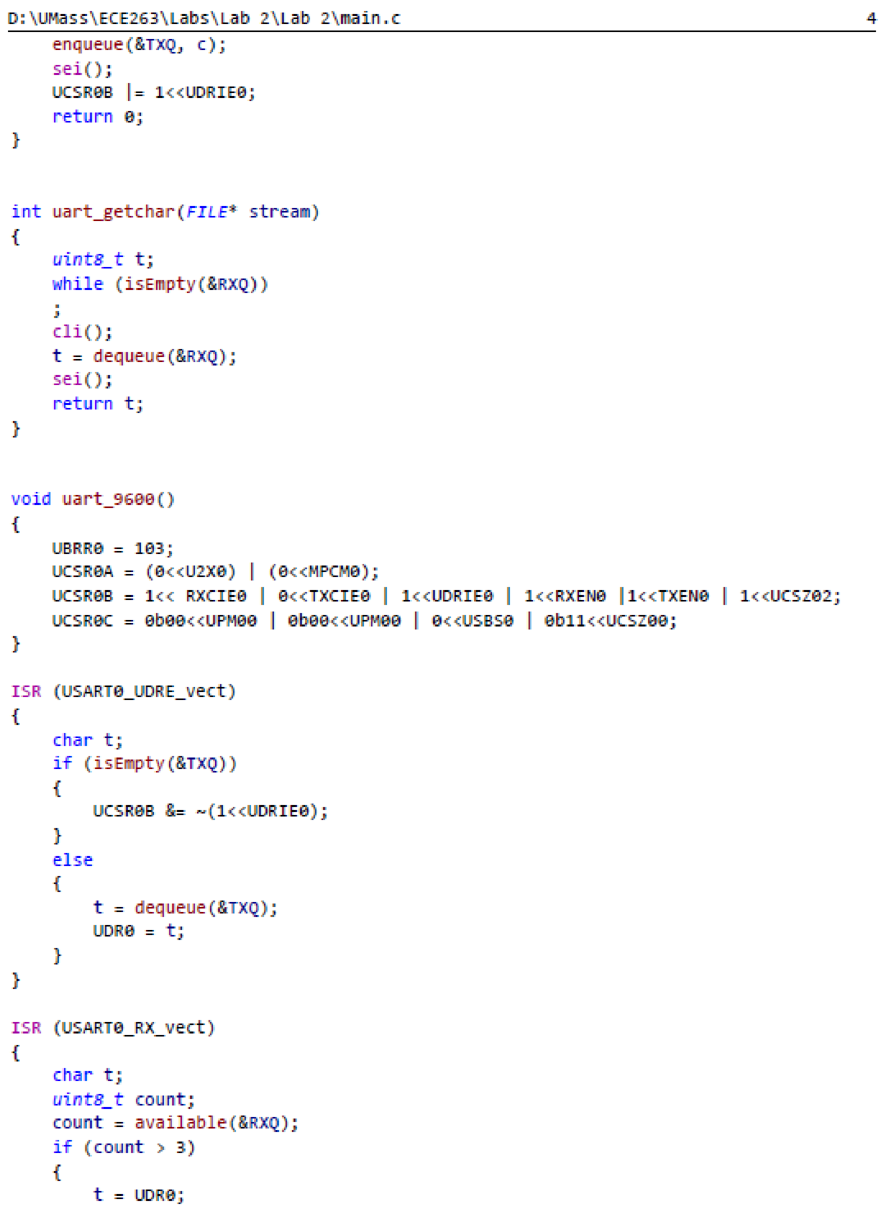
```
D:\UMass\ECE263\Labs\Lab 2\Lab 2\main.c                                      2
uint8_t txBuffer[BUFFSIZE];
uint8_t rxBuffer[BUFFSIZE];
MBUFFER TXQ;
MBUFFER RXQ;

int uart_putchar(char c, FILE* stream);
int uart_getchar(FILE* stream);
FILE mystdout = FDEV_SETUP_STREAM(uart_putchar, NULL, _FDEV_SETUP_WRITE);
FILE mystdin = FDEV_SETUP_STREAM(NULL, uart_getchar, _FDEV_SETUP_READ);
void uart_9600();
void SetLED(uint8_t, uint8_t, uint8_t);
void InitPWM();
int main(void)
{
    char choice[40];
    uint8_t red = 0;
    uint8_t blue = 0;
    uint8_t green = 0;
    SetLED(0,0,0);
    DDRD |= (1<<6);
    DDRB |= (1<<1);
    DDRB |= (1<<3);
    DDRC |= (1<<3);
    PORTC |= (1<<3);
    bufInit(&TXQ, txBuffer, BUFFSIZE);
    bufInit(&RXQ, rxBuffer, BUFFSIZE);
    stdout = &mystdout;
    stdin = &mystdin;
    uart_9600();
    InitPWM();
    printf("Welcome to COLOR display!");
    _delay_ms(25);
    while (1)
    {
        printf("\r\nCommand: ");
        _delay_ms(25);
        fgets(choice,40-1,stdin);
        choice[strlen(choice)-2]='\0';
        if(strcmp(choice,"r") == 0)
        {
            printf("\r\nValue: ");
            _delay_ms(25);
            fgets(choice,40-1,stdin);
            choice[strlen(choice)-2]='\0';
            red = atof(choice);
        }
        else if(strcmp(choice,"b") == 0)
        {
            printf("\r\nValue: ");
```

Figure 2: Code Part-2



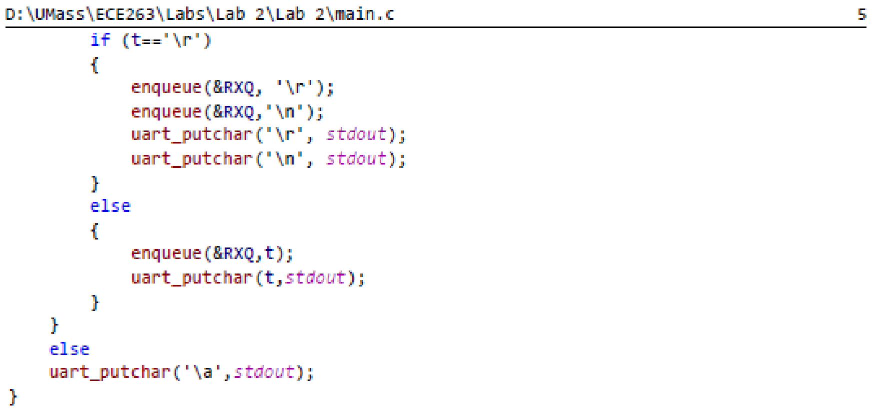
```
D:\UMass\ECE263\Labs\Lab 2\Lab 2\main.c                                          3
            _delay_ms(25);
            fgets(choice,40-1,stdin);
            choice[strlen(choice)-2]='\0';
            blue = atof(choice);
        }
        else if(strcmp(choice,"g") == 0)
        {
            printf("\r\nValue: ");
            _delay_ms(25);
            fgets(choice,40-1,stdin);
            choice[strlen(choice)-2]='\0';
            green = atof(choice);
        }
        else if(strcmp(choice,"s") == 0)
        {
            printf("\r\nRed = %d, Green = %d, Blue = %d" , red, green, blue);
            _delay_ms(25);
        }
        else if(strcmp(choice,"w") == 0)
        {
            SetLED(red,green,blue);
            _delay_ms(25);
            printf("\r\nR = %d, G = %d, B = %d sent to LED" , red, green, blue);
            _delay_ms(25);
        }
        else if(strcmp(choice,"?") == 0)
        {
            printf("\r\nList of Commands:");
            _delay_ms(25);
            printf("\r\n r - set red value");
            _delay_ms(25);
            printf("\r\n g - set green value");
            _delay_ms(25);
            printf("\r\n b - set blue value");
            _delay_ms(25);
            printf("\r\n w - write value to LED");
            _delay_ms(25);
            printf("\r\n s - status of values");
            _delay_ms(25);
            printf("\r\n ? - display commands");
            _delay_ms(25);
        }
    }
}


int uart_putchar(char c, FILE* stream)
{
    cli();
```

Figure 3: Code Part-3



```
D:\UMass\ECE263\Labs\Lab 2\Lab 2\main.c                                                    4
    enqueue(&TXQ, c);
    sei();
    UCSR0B |= 1<<UDRIE0;
    return 0;
}


int uart_getchar(FILE* stream)
{
    uint8_t t;
    while (isEmpty(&RXQ))
    ;
    cli();
    t = dequeue(&RXQ);
    sei();
    return t;
}


void uart_9600()
{
    UBRR0 = 103;
    UCSR0A = (0<<U2X0) | (0<<MPCM0);
    UCSR0B = 1<< RXCIE0 | 0<<TXCIE0 | 1<<UDRIE0 | 1<<RXEN0 |1<<TXEN0 | 1<<UCSZ02;
    UCSR0C = 0b00<<UPM00 | 0b00<<UPM00 | 0<<USBS0 | 0b11<<UCSZ00;
}

ISR (USART0_UDRE_vect)
{
    char t;
    if (isEmpty(&TXQ))
    {
        UCSR0B &= ~(1<<UDRIE0);
    }
    else
    {
        t = dequeue(&TXQ);
        UDR0 = t;
    }
}

ISR (USART0_RX_vect)
{
    char t;
    uint8_t count;
    count = available(&RXQ);
    if (count > 3)
    {
        t = UDR0;
```

Figure 4: Code Part-4



D:\UMass\ECE263\Labs\Lab 2\Lab 2\main.c 5

```
        if (t=='\r')
        {
            enqueue(&RXQ, '\r');
            enqueue(&RXQ,'\n');
            uart_putchar('\r', stdout);
            uart_putchar('\n', stdout);
        }
        else
        {
            enqueue(&RXQ,t);
            uart_putchar(t,stdout);
        }
    }
    else
    uart_putchar('\a',stdout);
}
```

Figure 5: Code Part-5