1 Design Overview

The goal of this experiment was to create an addressable RGB LED that is controlled by the Microcontroller and can be connected with other LEDS in series to create a desired pattern. Using the timer and CCR registers in the microcontroller, each color of of the LED could be controlled through manipulating their duty cycles through pulse width modulation. UART was used to send hexadecimal values through the TX line of the microcontroller to change the duty cycle of the red, green, and blue lights in the LED.

1.1 Design Features

- RGB LED that can be change in brightness and color dictated by the user.
- CCR register for each color, addressable between 0 to 255 to dictate the brightness.
- UART messaging to send information to the LED.
- Ability to put LEDs in series and use UART to address each in the chain.

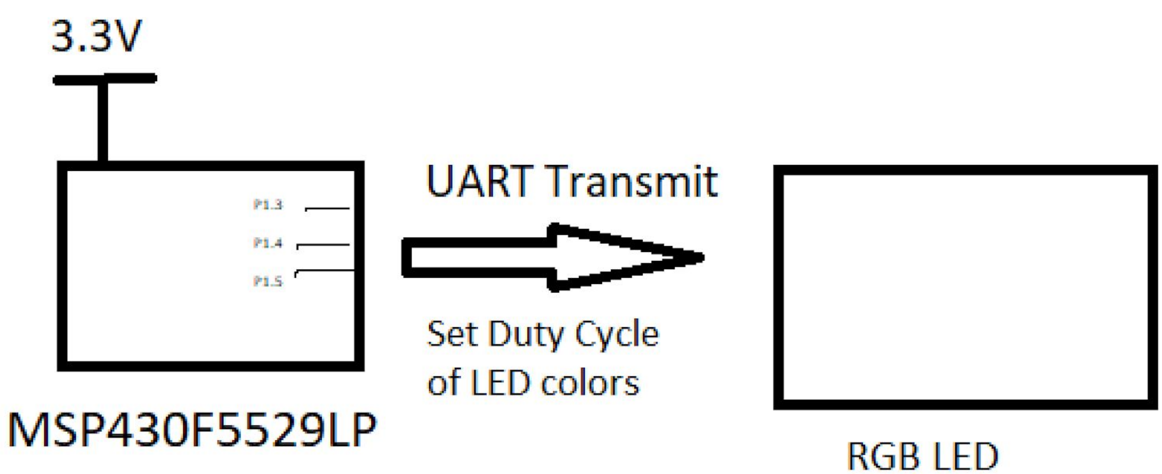
1.2 Featured Applications

- Could be used in a customizable lighting project.
- Can operate multiple lights, or a strip.
- Can communicate with patterns of lights if needed.

1.3 Design Resources

Original code provided by TI located at:
http://dev.ti.com/tirex/#/Device/MSP430F5229/?link=Software%2FMSP430Ware%2FDevices%2FMSP430F5229%2FPeripheral%20Examples%2FRegister%20Level%2Fmsp430f522x_uscia0_uart_01.c
Finished code located at:
https://github.com/RU09342-F18/milestone-1-team-1/tree/master/Milestone_StrangerThings/main.c

1.4 Block Diagram



1.5 Board Image

PLACEHOLDER TO BE ADDED SOON

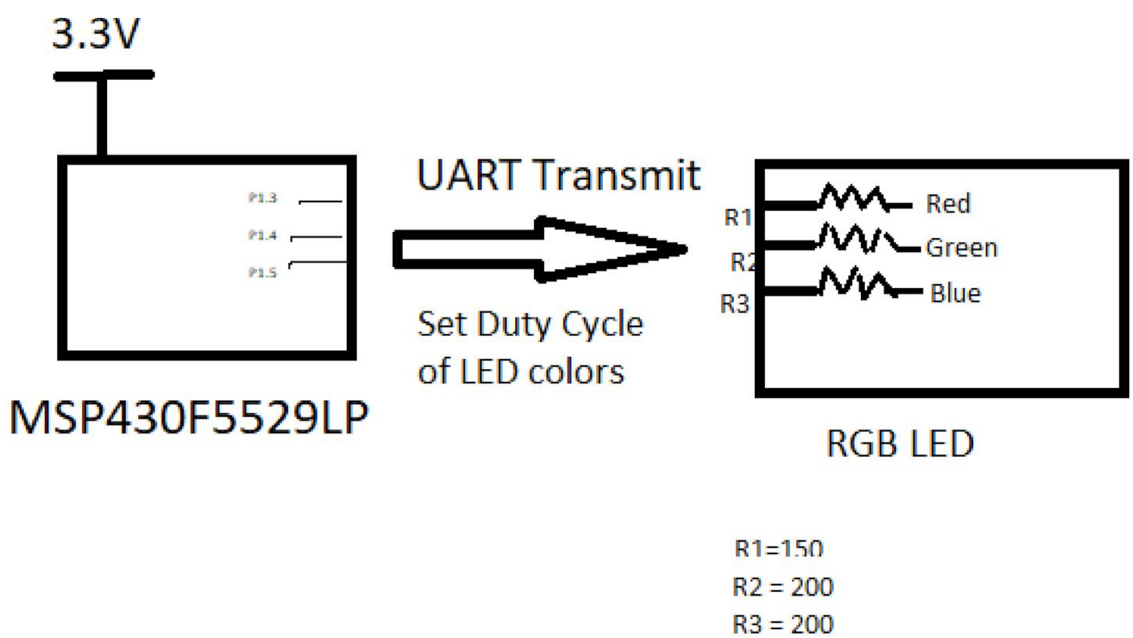
2 Key System Specifications

PARAMETER	SPECIFICATIONS	DETAILS
Micro Controller	MSP430F5229	Microcontroller used
RGB LED	Common Anode	RGB Light Used
Anode Voltage	3.3V	Voltage for Anode RGB LED
Baud Rate	Speed of data	9600 baud

3 System Description

The goal of the system was to allow the user to customize the color and brightness of an RGB LED using UART through the microcontroller. RealTerm was used to transmit bytes to the microcontroller to dictate the duty cycle of each color of the RGB LED. A capture-compare register was assigned for each color. The length of the duty cycle dictates the brightness of the corresponding color in the LED.

3.1 Detailed Block Diagram



3.2 Highlighted Devices

- MSP4305529LP
- Common Anode RGB LED

3.3 MSP430F5529LP

The processor within the microcontroller initializes pins 1.3, 1.4, and 1.5 to connect to the three colors of the RGB LED. Capture compare registers are used as well as the timer within the processor to emulate pulse width modulation for the pins. Each register corresponds to a color on the LED, and when RealTerm is used, the packets of data sent through UART dictates the duty cycle for each color by setting its capture compare register to a number between 0 and 255. 0 sets the duty cycle to 0% (off) on the corresponding color and FF sets it to 100% which is full brightness.

3.4 Common Anode RGB LED

The RGB connection for this design was constructed using a common anode RGB LED. The voltage for each color was originally calculated and tested to let the same amount of current through; however, after testing the RGB physically it was found that the resistor amounts had to be changed to match the brightness of each color. The final resistor amounts was 200Ω resistor for red, 200Ω resistor for green, and 100Ω resistor for blue.

4 SYSTEM DESIGN THEORY

4.1 Design Requirement: PWM

Implementing pulse width modulation is central to the design of this project. PWM allows the user to customize the brightness of each color of the LED. When the CCR registers are set to a lower value, the duty cycle for that color decreases, which means it runs at a lower frequency. Because the frequency is faster than the human eye can detect, the color looks as if it is getting brighter or darker.

4.2 Design Requirement: UART

UART allows for the processor to actually receive each byte on the UCA0RXBUF pin/register and transmit it to the UCA0TXBUF pin/register. The code was mostly provided by TI, which made it very easy to implement and is the easiest method to transmit 8 bit data for use on something like and RGB LED.

5 Getting Started/How to use the device

The device is quite user friendly. To operate, one must input hexadecimal values using Realterm to dictate the brightness of each light. Knowledge of the hexadecimal number system would be helpful to understand the scaling values for each color. One must use the proper format when inputting the data to the microcontroller to control the LED. When using RealTerm, a string of values must be written. First, the length of the data to be sent to the LED in bytes, next the scaling duty cycle for the red, green, and blue LEDs, respectively. For instance, if the user wanted the LED to be completely off, the input into RealTerm would be, 0x03 0x00 0x00 0x00. This is made up of four bytes of data to be transmitted to the microcontroller.

6 Getting Started Software/Firmware

6.1 Communicating with the Device

Communication with this device is done through RealTerm. Using UART, packets of data are sent from RealTerm to the microcontroller. The user can input bytes with hexadecimal values to dictate brightness of the LED.

7 Test Setup

UART transmission can be tested by initializing a pin to an LED built onto the board. This LED should be set to toggle whenever a packet of bytes was sent to the device. This will allow the user to know if the device is receiving a signal from RealTerm or any other software used to send data to the microcontroller. The functionality of the RGB LED can be tested by sending 3 bytes through RealTerm to check if the corresponding colors light the way the user expects them to.

7.1 Test Data

The easiest way to test the setup is by flashing the code to a MSP430F5229LP and opening RealTerm. If you configure RealTerm and send a packet of 0x03 0xFF 0xFF 0xFF, you will receive back 0x00 and if you have an RGB LED setup you will see the LED go full brightness on all three colors. You can test other packets as well using this method by configuring the length byte followed by as many red blue and green bytes as you want. The return values seen in RealTerm should be the length byte decreased by three followed by the fourth byte entered and onward.

8 Design Files

8.1 Bill of Materials

- MSP430F5529LP microcontroller
- 1x 150 ohm resistor and 2x 200 ohm resistor
- RGB LED
- Breadboard
- Male-female wires
- Male-male wires
- USB power source
- RealTerm program to transmit data