Milestone 1 Report

Douglas Fisher and Nick Papas
Rowan University

November 1, 2019

1 Design Overview

This project utilized a MSP430G2553 to receive and transmit Uart communications and control an RGB LED. The microcontroller receives a package on a specific protocol, removes second, third, and forth byte and sends the rest of the package on its transmit line. The bytes then correlate to the duty cycle of a PWM outut. These PWM outputs are connected to the anodes of a common cathode RBG LED.

1.1 Design Features

These are the design features:

- Receives and transmits serial communication
- Modifies received package to collect data and transmits remainder
- Utilizes three PWM outputs
- Controls a common cathode RBG LED

1.2 Featured Applications

This project could be used for the following applications:

- Light strip
- Transferring information
- Communicating with the upside down

1.3 Design Resources

The github project and software can be found in the links below:

Project

Software

1.4 Block Diagram

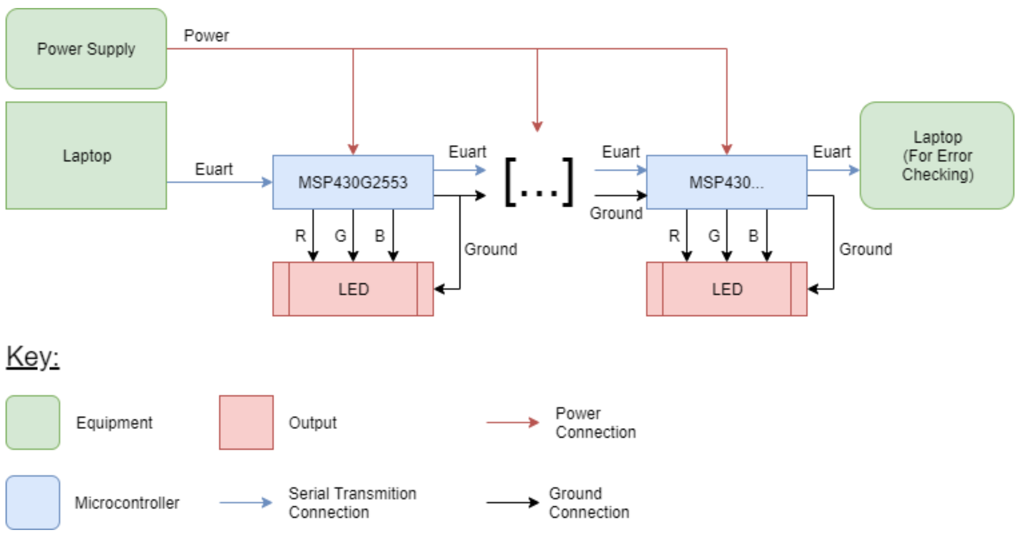

Figure 1: System Block Diagram

1.5 Board Image

Figure 2: Bottom of Board

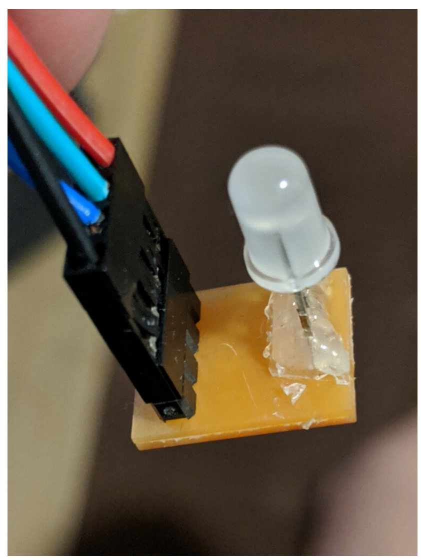

Figure 3: Top of Board

2 Key System Specifications

PARAMETER	SPECIFICATIONS	DETAILS
Communication	UART	An important specification when talking about UART is Baud Rate. Baud rate is the maximum amount of bits that can be sent per second. To ensure easy communication between nodes the baud rate should be set to 9600.
Messaging Protocol	N Bytes	For the messaging protocol 3 bytes are needed per RGB node. Another byte will be added at the end of the package so the master will know if all nodes extracted their bytes accordingly. Another byte should be added to the beginning of the package to indicate to each node how many bytes it should be getting.
RGB Node	MSP430	The RGB node is the MSP430 and it functions as a buffer and a PWM driver for each diode. It receives information by UART, converts the HEX bits to a duty cycle, then drives each diode in the RGB LED accordingly.

3 System Description

In this project the problem being attacked is being able to precisely control the color of an RGB LED over UART communication. The other main goal is to be able to put as many nodes in series as possible and have the master control them all accordingly.

3.1 Detailed Block Diagram

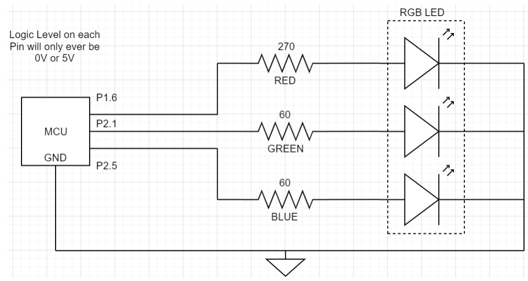

Figure 4: Low Level Block Diagram

3.2 Highlighted Devices

- MSP430G2553, acts as the RGB node and the PWN driver for each diode
- MSP430G2553 launchpad, contains external modules for output, filtering, input, and protection
- FTDI chip, converts USB to serial

3.3 Device/IC 1

The MSP430G2553 is a 16-bit low power microcontroller. It has an operating voltage of 1.8V-3.6V, 16 I/O ports, and a max clock frequency of 16 MHz. It was chosen instead of the MSP430FR6989 because the G2 is more simplistic. The FR6989 has many more functionalities adding more delay to the system. The G2 was more than capable for this project and it would be more efficient.

3.4 Device/IC 2

The LED driving circuitry consists of the internal PWM drivers of the microcontroller. The inputs to the circuit are the red LED, the green LED, the blue LED, and ground. These connect to the P1.6, P2.1, P2.5, and ground pins respectively.

4 SYSTEM DESIGN THEORY

Communications come in over the USB connector, which goes into the FTDI chip. The FTDI chip outputs serial Uart which goes to the PIC processor. The MSP430 processor receives the bytes of data one at a time processing each package quickly as it is received. Most bytes are immediately sent back out on the MSP430's transmit line. The first bit is the length of the package so it is subtracted by three since the MSP430 takes three bytes from the package. The second, third, and fourth bytes are accepted by the MSP430 and are used for the duty cycles of the RBG LED. Bytes are then sent as they are received until the carriage return byte (0x0D) is received and the length of the package is reached. Once that occurs, the package counter variable is reset and the MSP430 is ready for a new package.

For the RGB LED, two timers were set with periods of 0 to FF. The values taken from the serial communication are set to the capture and compare registers of the timers, so an interrupt is flagged when it counts to the entered value. This give off a PWM output to the pins, which set the brightness of the LEDs.

4.1 Uart Communications

An interrupt is triggered every time a byte is received by the MSP430. This byte is then read and then based on the byte counter variable, it is handled differently. In most cases, it is immediately returned without editing. However, the fist through the fourth byte and the last byte are handled differently. The first byte is the length which needs to be deceased by three since three bytes are being taken from the package. Those bytes are the second, third, and fourth, which become the duty cycles for the PWM. Then the bytes are sent as they are received until the length of the package is reached. It then outputs the carriage return byte and resets the byte counter variable.

4.2 PWM RGB Output

The RBG LED uses two timers for operation. The timers are both set to upmode with a period of FF. This means every time the timer reaches FF, it will reset to zero and continue. These timers have additional capture and compare registers that will throw interrupts when the timer reaches, but wont reset the timer. The values taken from the serial package will set in the capture and compare registers, and when the value is reached it throws an interrupt. The output pins are then set to PWM mode, which automatically operates off of those timers. When the duty cycles value is reached, the output is automatically set low, and when the cycle length is reached (0xFF), it sets the output to high.

5 Getting Started/How to use the device

This project uses the MSP430G2553 launchpad. However, can be replicated using a MSP430G2553, an FTDI chip, and a USB connection. This setup guide will focus on using the launchpad to control the LED.

5.1 Launchpad Setup

First make sure your jumpers are correct. The two ground pins should be bridge and the Tx and Rx lines should be bridged. The latter is so that the USB connection and processor are able to communicate. All other jumpers should be connect for normal operation. Then, connect your laptop over USB. See section 6 for how to setup the software, and how to setup the Uart communications.

5.2 RBG LED Setup

For the LED, first connect the ground on the PCB to the ground pin. Then connect the PWM outputs to the red, green, and blue anodes. The red output is P1.6, the blue is P2.1, and the green is P2.5. The PCB already has resistors on the bottom plate so you don't need to add additional resistance.

6 Getting Started Software/Firmware

This software is for the MSP430G2553 device. The project was made using the code composer IDE for editing and programming. Follow the device setup and the protocol specifications to control the RGB LED.

6.1 Communicating with the Device

In order to communicate with the device, download a terminal window capable of serial communications. Realterm is recommended, but PuTTY will also work. After connecting the MSP430G2553 to your laptop, set the baud rate, find the port it is connected to (using device manager). Then connect to the port and send over your bytes of data. The device will receive the information and output it over its Tx line.

6.2 Device Specific Information

In order for the serial communications to work properly, the jumpers for the MSP430G2553 must be changed. The jumpers should be bridging the Tx and Rx lines on the processor. This allows for the device to communicate through the USB port, as well as over the output pins.

7 Test Setup

To setup the device for testing a breadboard, 3 resistors, and jumper wires are required. The first thing that needs to be found in the pinout of the RGB LED. A common cathode LED is needed for this experiment however by reconfiguring software a common anode RGB LED can be used. After finding the pinout the forward voltage per diode is also very important information to have. To calculate the resistor values per diode one would take the supply voltage, subtract the forward voltage and divide the resulting difference by the amount of current. (For the MSP430G2553 the maximum output current of any pin is 6mA.) Once the resistor values have been calculated put each of them in series with their respective diode, tie the common cathode of the LED to ground, and using the jumper wires connect the output PWM pins to the correct colors. Once the circuit has been properly wired the LED should turn on with the correct color scheme after the RGB node (MSP430) has received the package over UART.

Resistor values calculated, accounting for the forward voltages and a max current draw of 5mA per diode, were R(red)=270ohms R(blue) and R(green)=60ohms.

8 Design Files

8.1 Schematics

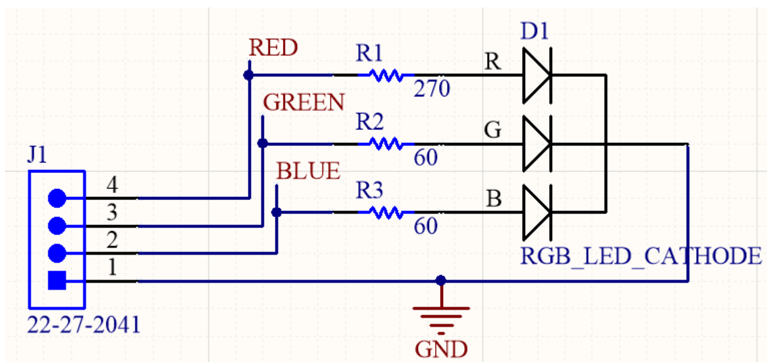

Figure 5: Schematic

8.2 Bill of Materials

	Item Details						
	Line #	Name	Descript...	Designator	Quantity	Manufacturer 1	Manufacturer Part Number 1
1		RGB_LED_CATHODE		D1	1		
2		22-27-2041	Male Hea...	J1	1	Molex	22-27-2041
3		243K 1% 0805(2012)	243K 0.12...	R1	1	Panasonic	ERJ-6ENF2433V
4		9M1 1% 0603(1608)	9M1 0.1W...	R2, R3	2	Vishay	CRCW06039M10FKEA

Figure 6: Bill of Materials

8.3 Layout Prints

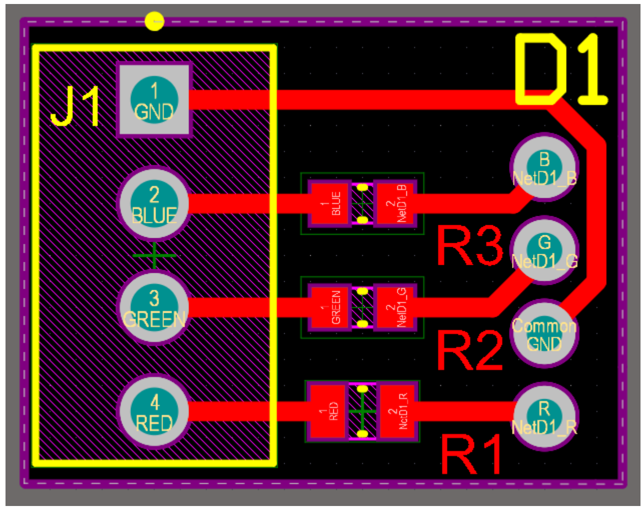

Figure 7: PCB Layout