Milestone 1: UART Controlled RGB LED Nodes

Julia Konstantinos and John McAvoy
Rowan University

October 22, 2018

Contents

1 Design Overview

The goal of this lab was pick an MSP430 family processor and on a register level, design it to be an RGB LED controller that controls the color of an LED based on specifically encoded RGB values passed through UART. The controller must also be able to act as a node in a series of similar controllers to process a UART message containing multiple RGB values, set its LED's color to the first RGB value, and then pass the rest of the RGB values to the next controller in the chain. This goal was accomplished by an MSP430G2553 processor running a custom C program, as well as a simple circuit to connect the red, green, and blue leads of the LED to corresponding pins on the G2553. The final design was capable of setting the RGB LED to the color specified by a UART message, passing on the message to the next controller in the chain, and detecting and handling errors in the received message.

1.1 Design Features

TODO

1.2 Featured Applications

This controller can be used for RGB lighting effects to make a chain individually addressable LEDs.

1.3 Design Resources

The resources that were given to design this controller were: a G2553 launchpad, a common-cathode RGB led, various resistors, wire, and a breadboard.

1.4 Block Diagram

TODO

1.5 Board Image

TODO

2 Key System Specifications

TODO

3 System Description

The controller takes in an encoded RGB brightness instruction via UART, sets the duty cycle of the red, green, and blue

3.1 Detailed Block Diagram

TODO

3.2 Highlighted Devices

TODO

3.3 Device/IC 1

TODO?

3.4 Device/IC 2

TODO?

4 System Design Theory

TODO

4.1 Controlling RGB Brightness Using PWM

TODO

4.2 Handling UART

TODO

5 Getting Started/How to use the device

5.1 Wiring Diagram

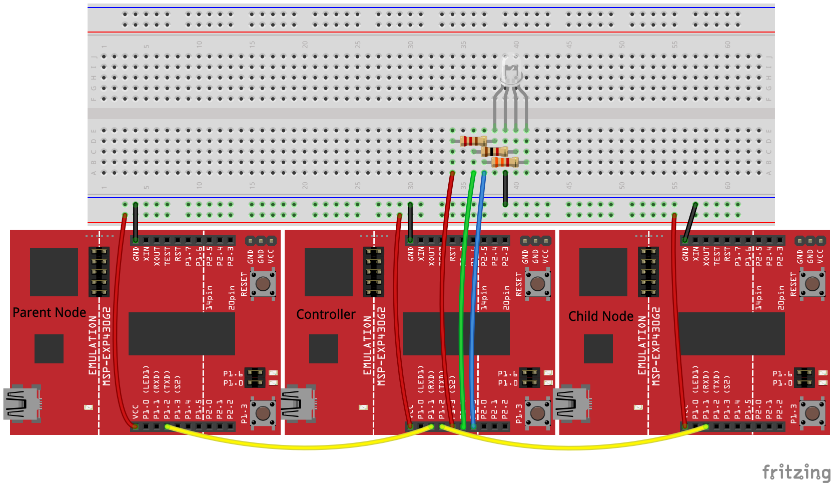
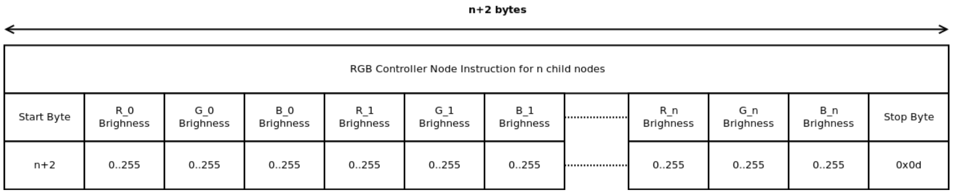


Figure 1: Circuit Wiring Diagram

5.2 UART Message Format

The RGB node controller is setup to parse a specifically formatted UART message. The UART message should be encoded according to Figure (2).



n+2 bytes

RGB Controller Node Instruction for n child nodes												
Start Byte	R_0 Brighness	G_0 Brighness	B_0 Brighness	R_1 Brighness	G_1 Brighness	B_1 Brighness	....	R_n Brighness	G_n Brighness	B_n Brighness	Stop Byte
n+2	0..255	0..255	0..255	0..255	0..255	0..255	....	0..255	0..255	0..255	0x0d

Figure 2: Instruction Format

Each instruction starts with a byte that indicates the length of the message in bytes, the following three bytes indicate the red, green, and blue brightness respectively for the controller, proceeding the RGB values for the child nodes in the chain are passed in the same way, and the message ends with a stop byte which is always 0x0d.

6 Getting Started Software/Firmware

6.1 Hierarchy Chart

TODO

6.2 Communicating with the Device

The instruction bytes (specified in Figure 2) can be sent via the G2553 launchpad's USB port or by connecting P1.1 (RX) to a UART signal. The controller's output can also be viewed by connecting the launchpad's USB to a computer and running a serial terminal (such as cutecom) that is able to view the hexadecimal bytes transmitted by the controller, or by connecting P1.2 (TX) to a UART decoder.

6.3 Device Specific Information

TODO

7 Test Setup

In order to test that the receiver and transmitter functioned as expected, the RX and TX pins of the G2553 were attached to an oscilloscope with UART serial decoding functionality. This allowed both the raw RX and TX voltage signals and the decoded hexadecimal RX and TX values to be visualized. Test UART messages were sent to the G2553's RX pin (P1.1) via a scope probe and the G2553's TX signal (P1.2) was sent via a second scope probe to the oscilloscope. Next, the RGB LED was setup on a breadboard and the red, green, and blue leads were connected through resistors to the corresponding I/O pins (Figure 3) to ensure that the color output matched what was expected from the test input.

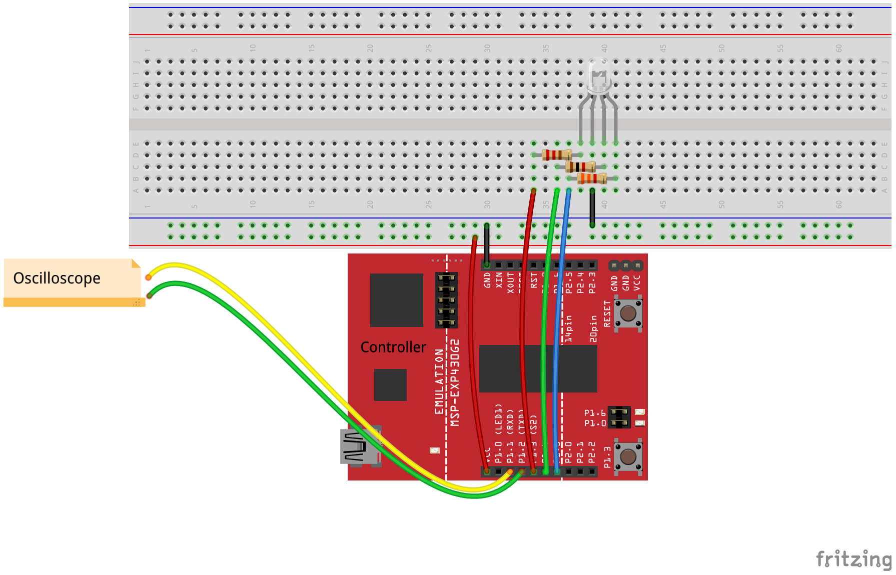


Figure 3: Test Circuit Wiring Diagram

Three tests were applied to the system: a series of 5-byte packets to test how the controller behaves as a single node, a series of 8- and 16-byte packets to test how the controller behaves with child nodes, and a series of messages with dropped packets to test how the controller handles errors.

7.1 Test Data

7.1.1 Single Node Tests

TODO

7.1.2 Children Node Tests

TODO

7.1.3 Dropped Packet Tests

TODO

7.2 Schematics

TODO

7.3 Bill of Materials

- 1 x MSP430G2553 Launchpad
- 1 x Breadboard
- 1 x Common-Cathode RGB LED
- 1 x 220Ω Resistor
- 1 x 1kΩ Resistor
- 1 x 3.3kΩ Resistor
- Wire