# Milestone 1

*Eric Schroeder and Ian Nielsen*
Rowan University

October 22, 2018

## 1 Design Overview

The goal of this milestone is to generate an RGB node using an MSP430 micro controller. This node could interface with other similar nodes to generate different LED patterns. The RBG node takes the instructions from a master node via UART and runs its set of instructions. The node then repackages the remaining instructions for the remaining nodes to complete.

### 1.1 Design Features

In order for the full features of this design to be realized multiple boards must be used. A single board will only enable the user to operate one LED.

These are the design features:

- Displays various colors via RGB LED
- Uses standardized HEX color inputs
- Lights up many LEDs in different colors using one input signal

### 1.2 Featured Applications

This design can be used for a wide range of applications, but the design was created with a few specific applications in mind.

Featured Applications Include:

- Digital Ouija Board
- Programmable Party Lights
- Moving movie theater aisle lights to help customers find the exit

## 1.3 Design Resources

Our code can be found on GitHub at the following link: [GitHub Code]

## 1.4 Block Diagram

This program does three things in order to complete the task, each of which are depicted in Fig. 1. It begins by initializing the IO ports, initializing the UART signal, set the RGB LED, then enable interrupts and enter low power mode. After this we can look at what happens when the interrupt vector is triggered. This happens when a UART signal is recieved. When the UART signal is received it sends the first three values to the RGB buffer, which determines the duty cycles of the three LEDs. Next the UART signal is sent to the UART TX Buffer, which removes the first three values from the signal and then sends the signal to the next node in the circuit. Finally the TX Buffer is cleared and the next UART signal is ready to be received.

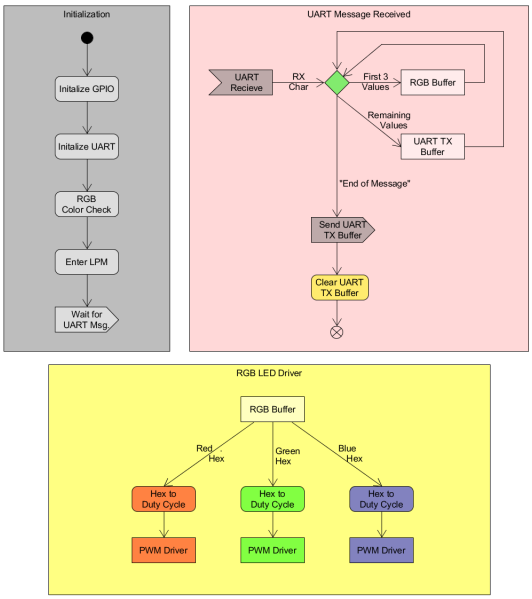


Figure 1: System Block Diagram

## 1.5 Board Image

The board that we constructed and tested our design with can be found below as Fig. 2.

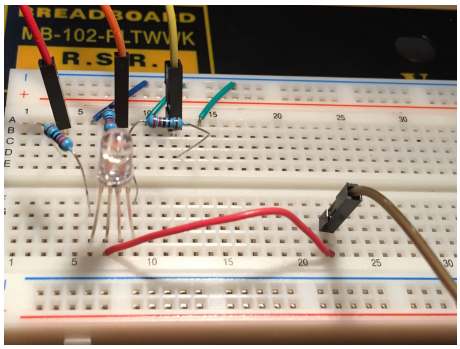

Figure 2: Breadboard Circuit

| PARAMETER | SPECIFICATIONS | DETAILS |
|---|---|---|
| Parameter 1 | 0xFFFFFF | This parameter turns all three LEDs within the RGB on, producing white. |
| Parameter 2 | 0x000000 | This parameter turns all three LEDs within the RGB off, producing black/no color. |
| Parameter 3 | 0xFF0000 | This parameter turns the red LED within the RGB on, producing red. |
| Parameter 4 | 0x00FF00 | This parameter turns the green LED within the RGB on, producing green. |
| Parameter 5 | 0x0000FF | This parameter turns the blue LED within the RGB on, producing blue. |
| Parameter 6 | 0xFFA500 | This parameter turns the red and green LEDs within the RGB on, producing orange. |
| Parameter 7 | 0xFF00FF | This parameter turns the red and blue LEDs within the RGB on, producing purple. |

## 2 Key System Specifications

Above is a chart showing different specifications that the system is capable of performing based on the specific UART code entered into the system.

## 3 System Description

The premise for this lab comes from a scene in the Netflix exclusive series ”Stranger Things,” in which the character Will Byers must communicate through a parallel dimension through a makeshift Ouija board set up by his mother.

## 3.1 Detailed Block Diagram

In order to implement this lab we need to connect multiple MSP430 processors in series. This can be seen in Fig. 3, which depicts the entire system working when all of the processors are connected.

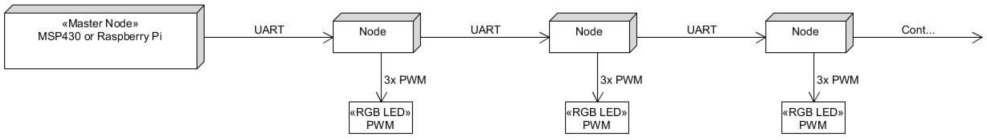


Figure 3: Processor Nodes

As you can see by the figure, the Master Node sends the UART signal to the first node, which processes it and sends the first three bytes to the RGB LED PWN, then reduces the signal by three and sends the updated UART code to the next node. This is then repeated until either the signal is empty or there are no more nodes left to send to or both.

## 3.2 Highlighted Devices

The processor in this project is used as a node for which the UART code is passed through and is the vehicle for the message to be processed and implemented. This device was chosen as it contains more capture compare registers for a single timer than the other MSP boards available.

- MSP430F5529

## 3.3 UART Buffers

The UART signal that is sent to the processor must be processed and sent as an output correctly. This is done with two buffers. The first is the UART RGB Buffer and the second is the UART TX Buffer. The RGB Buffer takes the first three values from the UART signal and sends them to RBG node. That means that there are three one byte duty cycle values that are saved in the buffer then sent to the RGB node, making a total of 24 bits. The second buffer is the TX Buffer, which deletes the first three values which were sent to the RGB node, then sends the shortened UART signal to the next processor/node in the total system.

## 3.4 RGB Node

In the previous subsection we discussed the signal that is sent to the LED. This UART signal can be seen entering the left side of the RGB Node in Fig. 4.

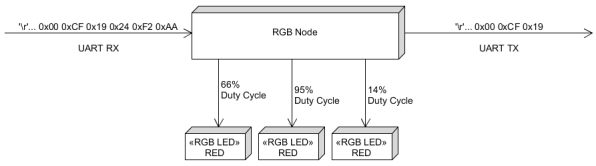


Figure 4: RGB Node

That signal is three bytes and it goes to the RGB Node where it is then sent to the red, green and blue lead of the LED. The first byte determines the duty cycle of the red lead, the second determines the duty cycle of the green lead and the third byte determines the duty cycle of the blue lead. Using this combination of controls we can make every color we would like to.

# 4 SYSTEM DESIGN THEORY

This lab is based involves lighting a series of RGB LEDs which are each connected to the MSP430. The idea for the system is that the first board gets the full set of bytes, which turn on the LED then delete the used LED information, then pass on the signal to the next board. The idea is to be able to send one package of information to the first board that will cascade to each other board in order to light up the entire string of lights to the desired color and brightness. This is far more simple to interface with than sending the LED information to each board individually.

# 5 Getting Started/How to use the device

In order to interface with this design you will need to understand the input format and how to connect the boards in series.

## 5.1 Input Format

The input format for this device is based on HEX color codes that are used in almost all devices that use a screen. The format consists of three bytes, each of which describes the duty cycle/intensity that the LED will be on at. This is the format for each individual color, but the format for the entire signal includes an initial byte which tells the processor how long the UART signal is. So the total signal is the number of bytes followed by each of the HEX color values in groups of three bytes.

### 5.2 Connecting Boards in Series

In order to connect these boards in series you must connect the UART input, which is RX at pin 3.3, and you also must connect the UART output, which is TX at pin 3.4. This first input would just be connected to a device that can send the initial signal, then the rest of the boards are connected from the output of the previous board to the input of the next board and the signal will then be able to travel the entire length of the system.

## 6 Getting Started Software/Firmware

In order to send a UART signal to the boards the program Realterm was used. Realterm is a serial communication program that allows a specific signal to be sent at will. Before signals are able to be sent Realterm must be configured for your device. First the baud rate for this code is set to 9600 for all boards which allows them to communicate. Next the port number on your specific device must be figured out and entered into Realterm to allow it to communicate. Once this is done signals can be sent via UART to the processor.

### 6.1 Device Specific Information

The MSP430F5529 is able to communicate using UART through the USB port on board providing it is addressed correctly throughout the code. This is one reason this board was chosen as it would reduce cost of replicating this milestone as TI provides a USB cable when the board is purchased.

## 7 Test Setup

This system was tested using the program Realterm. Within Realterm you can send and received UART codes, which is what we did. We send a UART code to the RX port for the desired color output and then checked the resulting output from the TX port to ensure that it was the value we expected, which is the same as the RX signal minus the first three values.

### 7.1 Test Data

In Fig. 5 we are testing the data being entered into our design and being sent out of our design. The information that was input into the system was 0x06 0xFF 0x00 0xFF 0x00 0xFF 0x00. The first bit tells us that the signal is six bytes then the RGB values follow. The signal that is output from our design is seen in yellow at the top of Fig. 5 as 0x03 0x00 0xFF 0x00. This shows that the signal 0xFF 0x00 0xFF was sent to the

board, lighting up the LED to be purple as seen in Fig. 6, then was deleted from the signal that is sent as the output. This creates a total signal that is three bytes shorter than the input, which is correct.

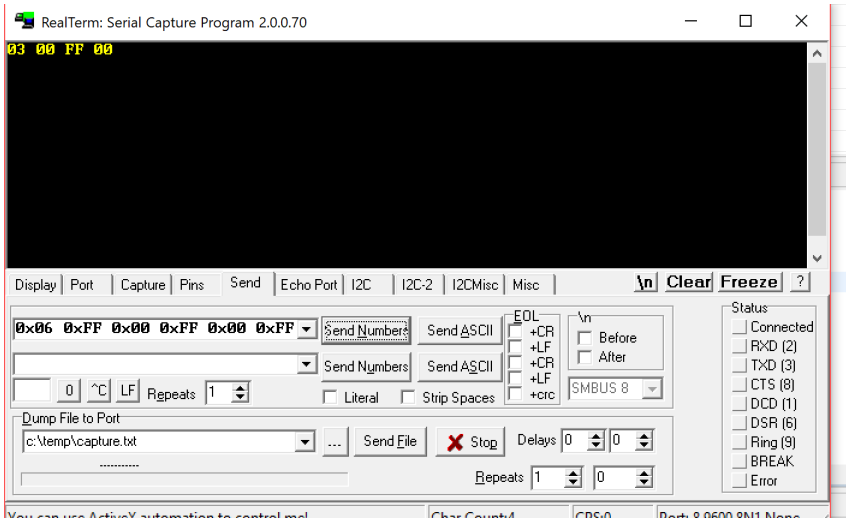


Figure 5: RealTerm Testing

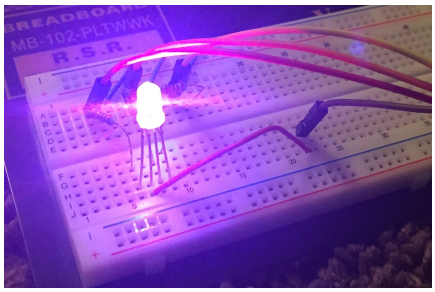

Figure 6: Breadboard Testing

# 8 Design Files

## 8.1 Schematics

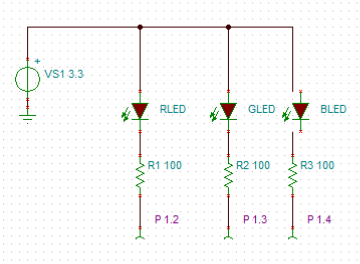


Figure 7: Breadboard Schematic

## 8.2 Bill of Materials

Materials Used:

- TI LaunchPad kit with MSP430F5529 MCU
- Breadboard
- Resistors
- RGB LED
- Breadboard wires