Communicating With Will Byer: Addressable LEDs

Jake Fraser and Tomas Uribe
Rowan University

October 16, 2017

1 Overview

Addressable LED strips are one of the latest advancements in the world of light emitting diode modules. In this project, an addressable strip of RGB LEDs, each controlled by a processor in the MSP430 line of processors is created. Each node in the strip will receive a UART signal composed of up to 80 bytes. From there, the processor linked to each LED will process the first four bytes in order to adjust the color of the LED. Each node will then alter the signal accordingly, and send it on its way to the next node of the system, allowing the next node to do the same.

2 Introduction

2.1 RGB LED

A normal LED works when a suitable voltage differential is applied to its two leads. This allows electrons to recombine with electron holes within the device, releasing energy in the form of photons. Normal LEDs are manufactured to be one set color when turned on, usually red, green, or blue. An RGB LED, like the one used in this project, has four leads in order to take in values for red, green, and blue, as well as a ground or power pin. Different strengths for the red, green, and blue nodes in the LED can create virtually any color in the visible spectrum.

Once the RGB pins on the LED are all connected to power, the LED will turn on as white. If the color red is desired, only the red pin will be connected. The same goes for green and blue. Now, heres where things get interesting. In order to change the strengths of each color node and create a different color, each pin can be connected to a PWM that modulates the amount of time the signal on that pin is high for. Because of this, it is easy to realize that the RGB nodes are actually all blinking at different speeds, creating the illusion of different strengths, and an overall different color.

RGB LEDS come are made in two different types of packages. Depending on whether the LED is common cathode or anode will determine how the LED is connected. A common anode LED will have its long lead connected to power, and its RGB leads connected to a resistor and then to ground. The common cathode LED will need ground to the long lead and a resistor value and power connected to the RGB pins. A figure showing the correct way to connect each type of LED is shown below. The only way to know if an RGB LED is common anode or common cathode is to connect it and test its setup.

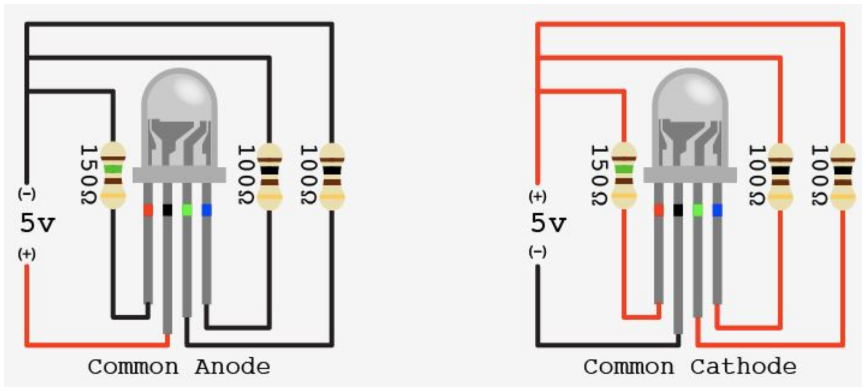

Figure 1: The correct way to connect an RGB LED

2.2 Addressable LED strip

Addressable LEDs are an extremely powerful device in LED technology. They allow users to individually program LEDs in strips as long as they want. Before this advancement, the user had to use a strip of LEDs whose emission color were all the same, which is a huge drawback if the LEDs are being used for a decorative design or as a signal.

Addressable LEDs work when a user inputs a specific string of HEX values that the LED strip uses to determine the color of each LED. Each LED will take the first three HEX values in the string using each one to determine the red, green, and blue components respectively. From there, the signal is sent to the next LED. The LEDS can be fed a string as long as the user wants, allowing a blinking pattern, a color pattern, even allowing the user to sync them with music. This technology is a huge breakthrough in blinking lights, and will continue to help users create new decorative designs.

3 Application

In this network of devices, a master node will generate and send out an 80 byte signal comprised of instructions for each node. This UART signal will then be sent from node to node, allowing each LED to turn on at the correct color. The block diagram for this system can be seen below in Figure 2

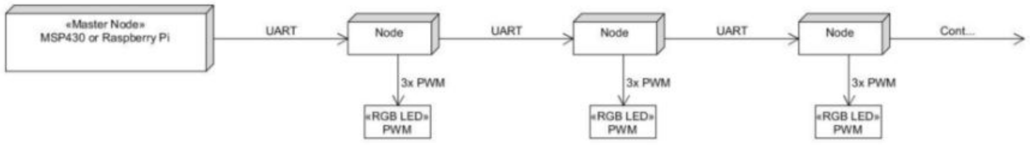

Figure 2: The block diagram for the addressable LED system

Each RGB node in the system will be responsible for taking in a string of HEX signals which will be responsible for determining the color of each LED. The signal will be received at each node through the processors UART RX line, using bytes 1 through 3 as its respective RGB values. Using a PWM, those values are then transformed into duty cycles to represent the proper color. The duty cycle of a PWM determines the length of time that a signal is high for. For example, a signal with a 95 percent duty cycle is high for 95 percent of the time. In other words, each LED will receive a red, green, and blue value which will decide the speed at which each color blinks. Each overall color that the RGB LED can output is comprised of a combination of red, blue, and green blinking at different speeds. An in depth system diagram of each node can be seen below.

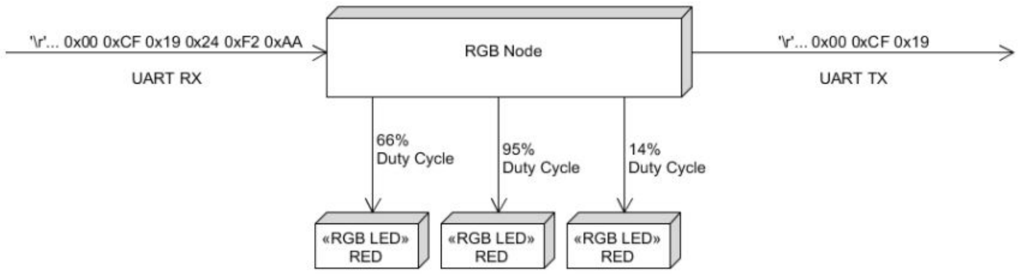

Figure 3: An in depth view at a node in the addressable LED system

Upon starting up, each node will be configured to initialize all timers and peripherals before immediately entering a low power mode while waiting for a UART message to be received. When a message is received, bytes 1 through 3 are stored to provide the LED with the proper color while byte 0 is reduced by 3. Byte 0 is used to represent how many bytes the signal is comprised of. The rest of the message is placed into a UART TX buffer and when the incoming message is complete, it is sent out over the UART TX Line to the next node.

Since the system is to be comprised of a maximum of 26 nodes, it will need at least 3 x 26 = 78 nodes. Adding a node in the beginning to be used as a byte count will bring the number of bytes in the signal to 79. Finally a byte is added at the end for debugging purposes. This byte will be able to tell the users if each of the nodes actually removed 3 bytes for the LED. If every node works correctly, then at the end of the string, the signal should be comprised of that placeholder and the first byte telling us that there is only one other byte in the signal.

4 Functionality

4.1 Software

The software portion of this milestone was broken up into 2 essential components; Pulse Width Modulation and UART. UART was used to receive and transmit data while PWM was used to initialize the RGB values of the LED

4.1.1 Pulse Width Modulator

Although the PWM was part of the software section, it was implemented using more of a hardware approach. This basically means that instead of enabling the capture/compare registers (CCRs) of the timers used, the values of these registers were just output to pins. Since the CCRx values were only used to determine the color of the RGB LED, they only had to be set and then output to their respective pin. No timer interrupts were needed because a specific task did not have to be performed when the timer counted to the different CCRx values.

A different CCR had to be assigned to each RGB node so each individual color of the LED had an independent duty cycle. By doing so each node of the RGB LED could blink at a different rate, producing the desired color.

The 3 independent CCRs were distributed among two timers. The reason for this was because each timer only has 3 CCRs which is not enough because one CCR has to be used as the period of the PWM. In this case, TimerA0 and TimerA1 implemented their CCR0 to create the period. In TIMERA0, CCR1 was used as the duty cycle for the RED node of the LED. As for TimerA1, CCR1 and CCR2 were used for the GREEN and BLUE nodes respectively.

Each CCR was output to a specific pin. However, choosing the correct pins was essential. CCRs can only be output to certain pins meaning that if a CCR was output to a pin that did not accept it, the PWM would not function properly. In the data sheet for the MSP430G2553 the pin assignments are well documented. By using this data sheet it was determined that TA0CCR1 could be output to Pin 1.6, while TA1CCR1 and TA1CCR2 could be output to Pin 2.1 and 2.4 respectively.
In order to actually output the CCRs, the selection registers, PxSEL and PxSEL2, had to be set. Table 1 below shows what the values of the Direction and Selection registers had to be set to in order to output the CCRs to the desired pin.

Table 1: Appropriate Values of the Selection and Direction Register in Order to Output CCRs

Pin	**CCRx**	**PxDIR**	**PxSEL**	**PxSEL2**
P1.6	TA0CCR1	1	1	0
P2.1	TA1CCR1	1	1	0
P2.4	TA1CCR2	1	1	0

Using the table above, each selection and direction register was set with the appropriate values necessary to output the desired CCR

NOTE: The capture/compare registers were not set until a UART interrupt was triggered. The following section will highlight how UART was implemented.

4.1.2 UART

UART was simply used to receive 80 bytes of data. However, these 80 bytes contained the information necessary to program an entire array of addressable LEDs. Each byte included information such as the number of bytes left in the message and the RGB values for each LED. It is important to note that the information was sent in a pattern and stored in a register called UCA0RXBUF when it was received. The number of bytes left in the message was the first byte received and then the bytes containing the value of the RED, GREEN, and BLUE nodes of the LED were sent second, third , and fourth respectively. Since UART can only receive data one byte at a time, logic was written to handle each byte of information as it was received. Additionally, since the pattern of the data was known it was simple to manipulate the code in order to function according to the byte that was sent. The first byte of data was recorded into a local variable called num0fBytes to save the value of bytes left in the message. The second byte was saved into the value of TA0CCR1 since that byte held the desired value of the RED node. Similarly, when the third and fourth bytes were received their values were saved to TA1CCR1 and TA1CCR2 respectively. Once all the CCRx values were set the LED displayed the specific color that was sent from the last three bytes.

After the LED was set with the correct color the remaining bytes were no longer pertinent to the processor. The data received after the BLUE node had been set was simply directly transmitted to the remaining processors. However, the first byte that had to be transmitted was the value of numOfBytes -3 to account for the bytes that were used to set the current processor's RGB values

4.2 Hardware

This portion of the project was used to take the values that were output to the pins by each CCR and physical wire them to a RGB LED. The circuit that was used can be seen below in Figure 4

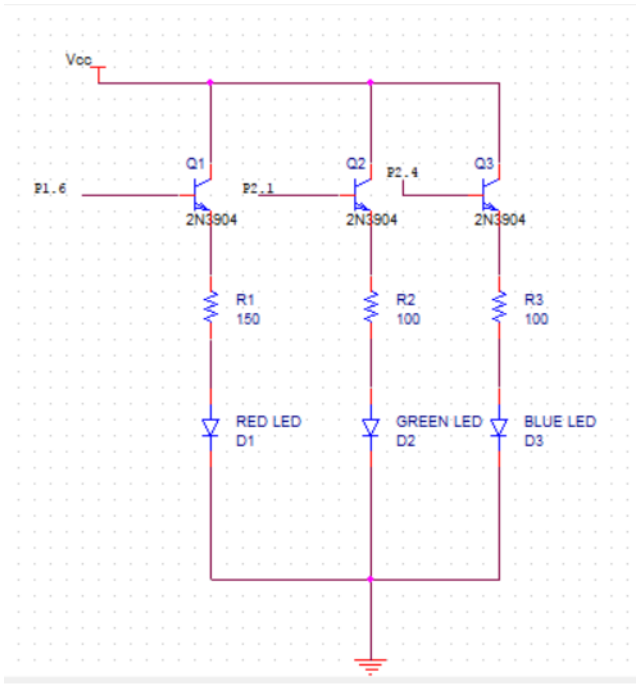

Figure 4: Hardware Used to Display the Signals from the MSP430G2553 onto the RGB LED

The LED that was used was a common cathode meaning that Vcc was be given to the RGB nodes while the PWR/GND node was grounded. This allowed the 2N3904 BJTs to be implemented as high-side switches. The reason the BJTs were used was because the processor is only able to output 6mA but the LED requires 20mA in order to turn on. The BJTs allowed the current coming out of the processor to be amplified. Once the bytes were received through UART and the CCRs were set, their values were sent to their respective nodes producing the desired color.

5 Conclusion

Through the completion of this project, a node for an addressable RGB LED strip was designed and built. This node will receive a string of HEX values through a UART receiver port. The first byte of this signal will signify how many HEX values there are. Because of this, the node can be placed anywhere within the LED strip. The second three bytes will signify the CCR values for the red, green, and blue values, which will adjust the duty cycle of the PWM for each of these pins. Once the signal is received the first byte is reduced by three, the second three bytes are processed, and the remaining bytes are sent to the next node. This modularity allows the user to program each node individually and place each one anywhere in the LED strip.