Milestone 1: Stranger Things Light Wall

Ryan Baker
Rowan University

October 23, 2018

1 Design Overview

In this milestone project, an MSP430 launchpad was used to create an addressable RGB LED node based on some provided specifications. During the course of this project, the specific launchpad being used had to be decided upon and justified. To pick the launchpad, qualities such as cost, features, and convenience were considered. Then, the MSP430 launchpad had to be configured with software and hardware to be able to address the RGB LED using PWM signals. The values used to set the PWM duty cycles were received from another node over UART and then the values for the remaining nodes were passed down to the next node over UART. By the end of this project, a functioning RGB LED node had been created using the MSP430G2553 launchpad and its functionality was verified by communicating with the board over a UART connection from a terminal.

1.1 Design Features

These are the design features:

- Addressable RGB LED with 256 levels of brightness for each color
- Capable of communicating with other nodes using a baud rate of 9600
- Each node uses three bytes to address its own LED than passes on all remaining bytes to the next node

1.2 Featured Applications

These are the featured applications:

- Stranger Things Light Wall

1.3 Design Resources

$https : //github.com/RU09342 - F18/milestone - 1 - if - you - too - have - procrastinated - join - me/blob/master/Milestone_StrangerThings/main.c$

1.4 Block Diagram

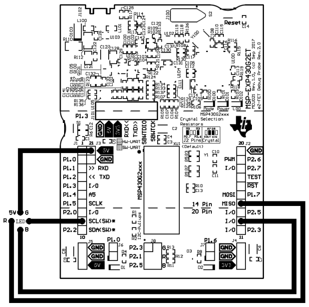

Figure 1: Schematic

1.5 Board Image

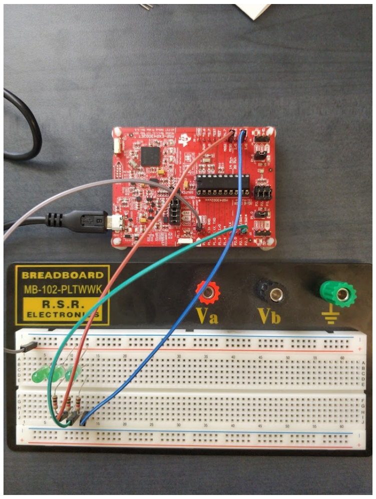

Figure 2: Board Image

PARAMETER	SPECIFICATIONS	DETAILS
In byte 1	0 - 255	Specifies the number of remaining bytes in the message
In byte 2	0 - 255	Specifies the value of the red LED
In byte 3	0 - 255	Specifies the value of the green LED
In byte 4	0 - 255	Specifies the value of the blue LED
In byte n	0 - 255	Passed to next node
Out byte 1	0 - 255	Equal to the value of (In byte 1) - 3 unless the value of (In byte 1) is less than 6, then this byte is equal to 0
Out byte n	0 - 255	Equal to (In byte n + 3)

2 Key System Specifications

3 System Description

The system is designed to be an addressable RGB LED node. The task is to use one of the MSP430 launchpads to create a node that can communicate to nodes down the chain over UART.

3.1 Detailed Block Diagram

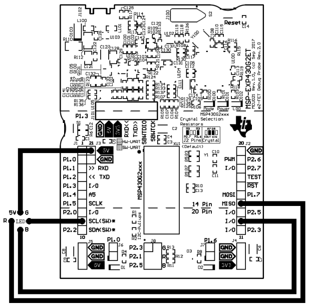

Figure 3: Schematic Detailed

3.2 Highlighted Devices

- MSP430G2553 - This device was used to control the RGB LED through software

3.3 MSP430G2553

The MSP430G2553 uses timer A0 and A1 to control the PWM signals for each color of the RGB LED. Timer A1 is made responsible for two of the colors by using TA1CCR1 and TA1CCR2 as well as TA1CCTL1 and TA1CCTL2. The LEDs are turned on when the timers hit 0, and off when the timers hit their CCR registers. The values of those CCR registers are set to 0 by default, but then they are set during the UART interrupt. If the first byte received during the UART interrupt indicates that there are at least 3 more bytes coming, then the next byte will set the value for red, the next one for green, and the next one for blue. If there are at least 3 more bytes after that, then the number of remaining bytes is sent out over UART and then the remaing bytes themselves are sent out too.

3.4 Off Board RGB LED

The RGB LED has its anode connected to 5V. Then its red leg is connected to P1.6 through a 1k ohm resistor. Its green leg is connected to P2.1 through a 1k ohm resistor. Lastly, its blue leg is connected to P2.4 through a 1k ohm resistor.

4 SYSTEM DESIGN THEORY

For this device, each of the colors in the RGB LEDs are controlled with PWM. The PWM signal has a period of 256 counts. Two change the duty cycle of each of the colors, the value in each color's CCR register can be changed. A value of 0 in a CCR register will turn that color off while a value of 255 will turn that color on at its max brightness. The values of the CCR registers are set during the UART interrupt. Every time the device receives a byte over UART, this interrupt is triggered. If the first byte indicates that there are more bytes to come, then the next 3 bytes will contain the new values of the red, green, and blue CCR registers, respectively. If there are sufficient bytes remaining, then they will be transmitted to the next node over UART.

4.1 UART Inputs

The device was required to accept inputs that followed these specifications. The first byte received stores the number of bytes that the device is expecting to receive following the first byte. The second, third and fourth bytes received contain the values used to set the duty cycle of the red, green and blue parts of the LED, respectively. The rest of the bytes are accepted into a buffer that is used for the output. To do this, two variables are used to keep track of the number of total bytes expected and the number of bytes yet to be processed. When the number of bytes yet to be processed, the device is ready to accept the number of bytes in the next string of data. This byte is used to reset the both variables. Then when the number of bytes yet to be processed is 1 less than the total bytes expected, that byte is used to set the value of the red component of the LED. When the number of bytes yet to be processed is 2 less than the total bytes expected, that byte is used to set the value of the green component of the LED. Lastly, when the number of bytes yet to be processed is 3 less than the total bytes expected, that byte is used to set the value of the blue component of the LED. Then if the number of remaining bytes is at least 3, those bytes are passed onto the next node. Otherwise, the variables are reset so the device can get ready to receive the next string of data.

4.2 UART Outputs

The device was required to send outputs based on these specifications. If the first byte received by the device was at least 6, then the first byte sent was equal to the value of the first byte minus 3. If that byte did not contain a value of at least 6 then the first byte sent was equal to 0. If there are at least 7 total bytes, then the rest of the bytes, starting with the fifth byte, are passed to the next node. If there are less than 7 total bytes, then no additional bytes are sent to the next node. To do this, the value of the first byte is checked and the value of the first byte being sent is either set to that value minus 3 or 0. For the rest of the bytes, the value of the first byte is checked and the rest of the bytes are either passed to the next node or none are.

5 Getting Started/How to use the device

This section will discuss how to use this device. Whether the device is being connected to the computer and acting as the first node in the chain or it is a different node in the chain, these are the steps need to start using the device.

5.1 Using with a computer terminal

To use this device as the first node, connect the board to a computer. Let the drivers automatically install then open up device manager and take note of the COM number for the UART connection to the board. Then open realterm. In the display tab, under display as, check Hex[space]. Then, in the port tab, switch the baud rate to 9600 and the port to the COM number from device manager, then hit the change button. In the send tab, type the full string of data with each byte starting with 0x and each byte also being separated by a space. Then hit send. The LED will change and the bytes returned will be displayed in the terminal.

5.2 Using with another node

To use this device anywhere in the node other than the first node, follow these steps. Connect the RX pin of this node to the TX pin of the node before it. Connect the TX pin of this node to the RX pin of the node after it, if there is one.

6 Getting Started Software/Firmware

This section will discuss how to communicate with the device as well as how to specifically program for this device.

6.1 Communicating with the Device

To begin communicating with this device, simply connect it to a computer using the included USB cable. The drivers will automatically install and this will allow for the code to be flashed onto the microcontroller.

6.2 Device Specific Information

To flash code onto the MSP430G2553 launchpad, in specific, follow these steps. Once the board is connected, open code composer studio. Create a new project that uses the MSP430G2553. Copy the provided code into the main.c file or write new code. Debug and let code composer studio flash the code to the launchpad. If there are any updates, let the board update. Once the code is running on the board. The device can be used or tested once the code has been flashed.

7 Test Setup

To test this device, connect the board to a computer. Make sure the drivers are installed then check the COM number for the UART connection to the board in device manager. Open realterm to begin testing. In the display tab, under display as, check Hex[space]. Then, in the port tab, switch the baud rate to 9600 and the port to the COM number from device manager, then hit the change button. In the send tab, type the tests from the next subsection. Then hit send. The LED will change and the bytes returned will be displayed in the terminal. Verify that the color is correct and that the bytes received are correct.

7.1 Test Data

8 Design Files

This section will discuss what parts were used to create this device. It will also include the schematics for the device.

INPUT	LED	OUTPUT
0x03 0xFF 0x00 0x00	Red	0x00
0x03 0x00 0xFF 0x00	Green	0x00
0x03 0x00 0x00 0xFF	Blue	0x00
0x03 0xFF 0xFF 0xFF	White	0x00
0x03 0x7F 0x7F 0x7F	Half White	0x00
0x06 0x00 0xFF 0xFF 0xFF 0x00 0x00	Teal	0x03 0xFF 0x00 0x00

Part	Quantity
MSP430G2553	1
Breadboard	1
RGB LED	1
1k ohm resistor	3

8.1 Bill of Materials

8.2 Schematics

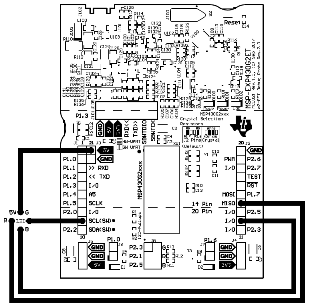

Figure 4: Schematic