Milestone 1: Stranger Things Light Wall

Alex Steel and Ryan Boylan
Rowan University

November 2, 2019

1 Design Overview

The objective for this Milestone is to create an addressable RGB node controlled by a microprocessor, in this case, the TI MSP430G2553. Each node must be able to be connected in series and be able to receive data from the previous node and transmit data to the next node. This system as a whole portrays the ”Light Wall” from the popular TV series, Stranger Things.

1.1 Design Features

The following design features represent the top level functions and capabilities of this project.

Design Features:

- Off-board addressable RGB LED
- Variable PWM
- Communication via UART
- Able to be connected in series with other boards

1.2 Featured Applications

Listed below are the featured applications of this project.

- Part of series of other addressable RGB LED nodes
- Color changing LED

1.3 Design Resources

The following links are to the Github repository where this lab is stored as well as the documentation for the specific microprocessor used in this lab.

- Github Repository.
- Datasheet.
- User's Guide.

1.4 Block Diagram

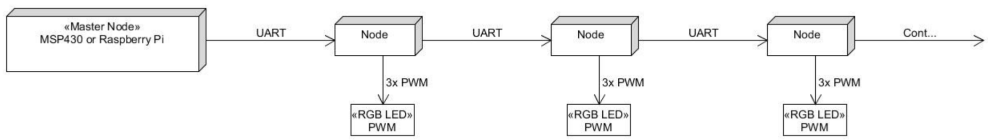

Figure 1: System Block Diagram

1.5 Board Image

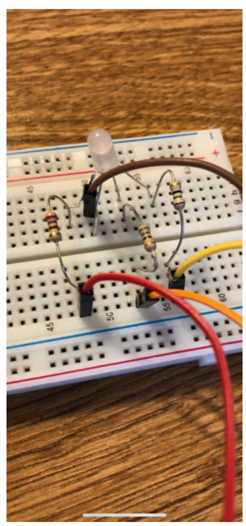

Figure 2: Breadboard Image

1.6 Combined Systems Image

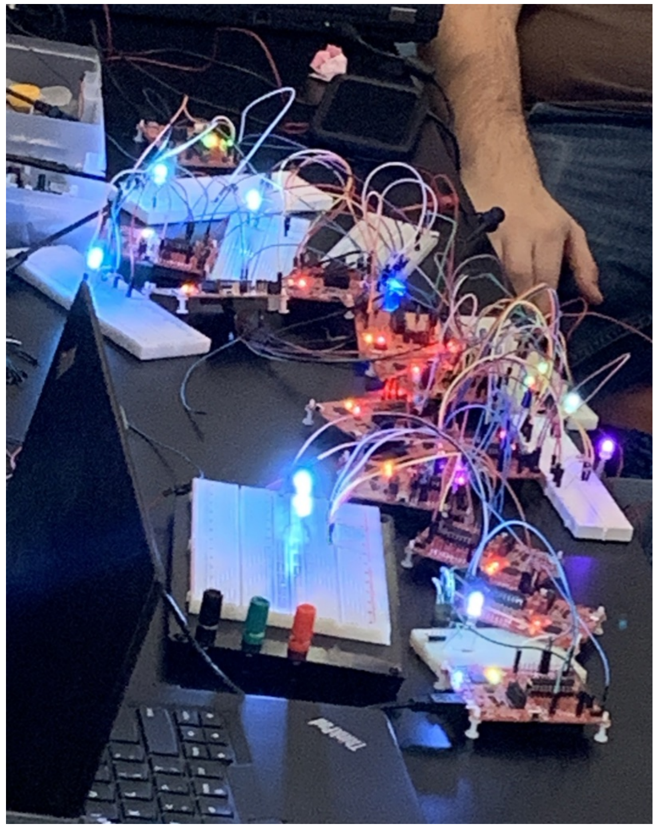

Figure 3: Overall Image of twelve nodes connected

2 Key System Specifications

Parameter	Specification	Details
SMCLK	1 MHz	PWM Timer Speed
Baud Rate	9600	UART Communication Speed
TA0CCR0	256	PWM Period
TA0CCR1	P1.6	Red PWM
TA1CCR0	256	PWM Period
TA1CCR1	P2.1	Green PWM
TA1CCR2	P2.5	Blue PWM

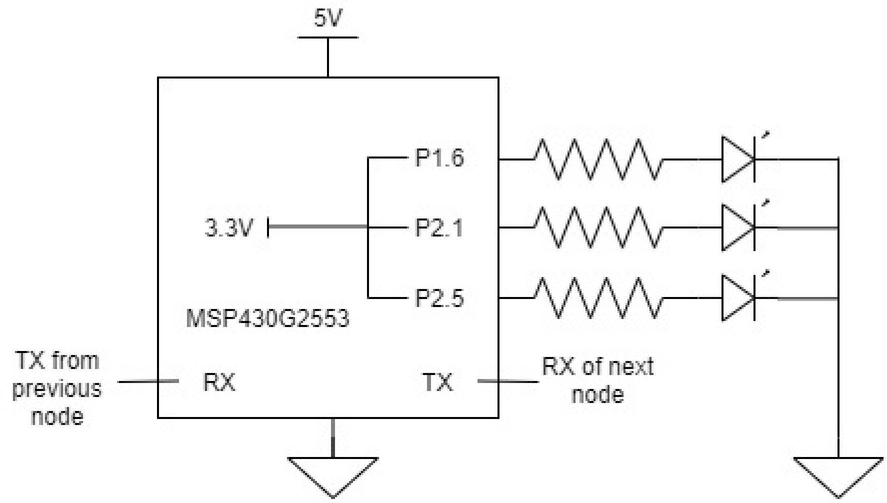

Figure 4: Block Diagram showing how the pins on the MSP430G2553 connect with other nodes and with the LED

3 System Description

The objective is to make multiple microprocessors connect in series to make a string of lights, each one addressable by a master node to create the desired color at each node of the string. The master node transmits a signal via UART to the first RGB node, the microprocessor will use this signal to determine what RGB pwm values to produce, then, it will transmit the signal on to the next node. This process repeats until the signal ends or there are no more nodes.

3.1 Detailed Block Diagram

3.2 Highlighted Devices

- MSP430G2553: Microprocessor to control RGB node
- RGB LED: Displays specified color
- Resistors: Limit current to LED and control color balance

3.3 MSP430

The MSP430 is the microprocessor that was chosen to take an input and control the output. The input is a string of bytes in a specified format and contains information as outlined in Table 1 below.

Byte Number	Contents
0	Number of bytes (N) in the package
1-(N-2)	RGB colors for each node
N-1	End of Message Character (0x0D)

Table 1: Breakdown of bytes contained in a package

The outputs of the MSP430 are the PWM signals used to control the red, green, and blue pins on the LED, as well as a string of bytes transmitted over UART to the next node in the series. To control the LED the microprocessor utilizes Timer A0 and Timer A1 both in up mode, as is explained further in Section 4.2.

3.4 RGB LED and Current Limiting Resistors

The LED is the output of this system. The MSP430 take an input, and controls the color of the LED based off of the combination of bytes. The current limiting resistors are meant to balance the current going to the three different colors of the LED, as the forward voltage of the red LED is lower than that of the green and the blue LEDs. The current going to an LED is given in Equation 1 below.

$$I_{Led} = \frac{V_{in} - V_f}{R_{lim}} \tag{1}$$

In this equation, V_{in} is the output voltage from the MSP430 and is 3.3V, V_f is the forward voltage of the LED, which differs for different LED colors, and R_{lim} is the current limiting resistor which is in series with the LED.

4 SYSTEM DESIGN THEORY

This system is made up of only a few parts. The system as a whole is made up of multiple nodes. Each node consists of three parts, although two are passive. The two passive parts are the resistors and the LED. The major part of the system is the microprocessor, which in this case, is the MSP430G2553. Each node is responsible for receiving data, setting the LED to the desired color, and transmitting data to the next node.

4.1 Design Requirement 1 - Communication

The first design requirement that had to be taken care of was the communication to and from each node. The communication is done via UART which stands for Universal Asynchronous Receiver/Transmitter. An input signal is received into the receive buffer and is analyzed byte by byte. The first byte is stored as it indicates the length of the input signal. The second, third, and fourth byte indicate the red, green, and blue PWMs respectively. These values are stored. Every byte after the fourth is not relevant to the processor. All of these bytes are sent to the following processor in the series by allocating the data to the transmit buffer. One thing to note is that the data is read, the needed data is stored, and the rest of the data is transmitted, then those stored values are assigned to the respective PWMs after transmitting. This is done to keep a flow of data to the next node and not slowing down the signal. Necessary data is transmitted before the LED actually turns on with the given PWM values.

4.2 Design Requirement 2 - Pulse Width Modulation

The second design requirement was managing and outputting three different signals with separate PWMs(Pulse Width Modulation). PWM affects the brightness of the color in the LED. One immediate roadblock that was encountered was the fact that the MSP430G2553s A0 timer only has three capture compare registers when four are needed for the objective. To solve this, Timer A0 and A1 are both used in the up mode, with the same reset value, the same clock, and the same clock divider. The reset value for each timer is set to the max PWM frequency of 255 or 0xFF in hex. CCR0(Capture Compare Register Zero) is used to store that reset value for both timers, A0 and A1. A0CCR1 is then used to store the red PWM, A1CCR1 is used for the green PWM, and A1CCR2 is used for the blue PWM. This method works to create the desired PWM effect. The reset frequency is 255 or 0xFF. All PWMs are set the same way, red will be used as an example. If the input data indicates to set the red LED to have a PWM value of 0x7D, which is half of 0xFF, then the timer starts at zero, and will interrupt at 0x7D, switching the state of the LED, then when the timer resets at 0xFF, the LED switches states, creating a PWM.

4.3 Design Requirement 3 - Off-Board LED

The final design requirement was setting up the off-board RGB LED. There were a few small calculations that went into the decisions. The circuit itself is very simple. There are three pins on the microprocessor that correspond to each of the three colors of the LED. Each pin is connected to a resistor, then to the LED, then to ground. The question is what resistor values to use. Knowing the max current output from the microprocessor is 6mA, and that there is a voltage drop across each LED, the resistor value for optimal brightness, or the smallest safe resistor value, is easily calculated. It is also necessary to keep in mind that the red LED has a voltage drop of approximately 2.2V while the green and blue LEDs have a voltage drop of approximately 3V.

5 Getting Started/How to use the device

To set up the device for testing and operation, begin by having the microprocessor plugged into a computer. The correctly working code must be uploaded to the board. Now the microprocessor can be unplugged from the computer. The breadboard must be set up according to Figure 2. Each LED pin must correctly be matched to its corresponding pin on the microprocessor. Corresponding pins are shown under the Key System Specifications section of this application note. Now, with the node set up to operate, the board must be powered, either by USB or the 5V power pins. If only one node is used, it must be plugged into the computer, so that data can be sent to it. If using multiple nodes, the first node must be connected to a computer, each of the next nodes can be connected in series with the transmit pin of node one connected to the receive pin of the next node, and this must be done for each of the next nodes in the system. To send a signal to the nodes, RealTerm is utilized. RealTerm can send bytes of data via UART to the first board. How the data sent effects the color of the LED on each node is described in Table 1.

6 Getting Started Software/Firmware

To utilize the MSP430 as an RGB node, first it must have the proper software uploaded to it. This involves using some compiler to upload code to the board. In this milestone TI Code Composer Studio was utilized, as it provided an easy way to upload code as well as debug due to the inbuilt register level monitoring. Once the MSP430 has the proper code uploaded to it, it can be disconnected from the computer and be powered from an external 5V source. In this setting, Realterm, a serial terminal, was used to send a data package to the MSP430.

6.1 Hierarchy Chart

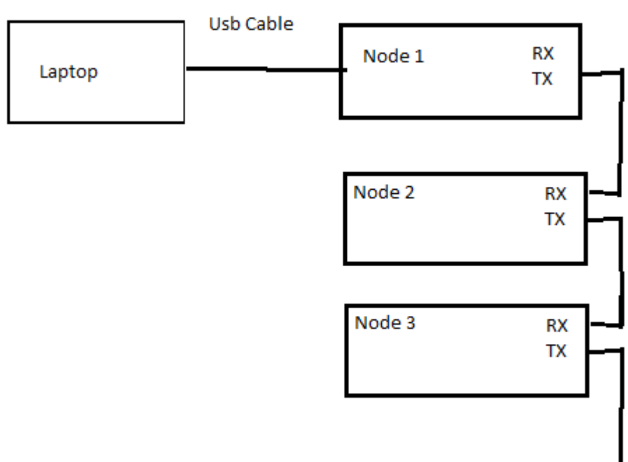

Figure 5: Simplified Diagram showing communication between Nodes

6.2 Communicating with the Device

Once the code has been uploaded to the MSP430, communicating with it involves connecting it to some serial port. If it is the first node in the chain, then the serial port will come from a laptop or desktop, otherwise the serial information sent over UART will come from the previous node in the chain. This is achieved by connecting the TX pin to the RX pin on the next node in the chain.

6.3 Device Specific Information

On the MSP430G2553, there is a header with several jumpers on it that change the configuration of the board. In order to switch between hardware and software UART, the jumpers labeled RX and TX must be in the orientation that is printed onto the silkscreen. In this case, hardware UART is used, so the two jumpers must be turned so that they are pointing perpendicular to their original positions, which is shown in Figure 6 below.

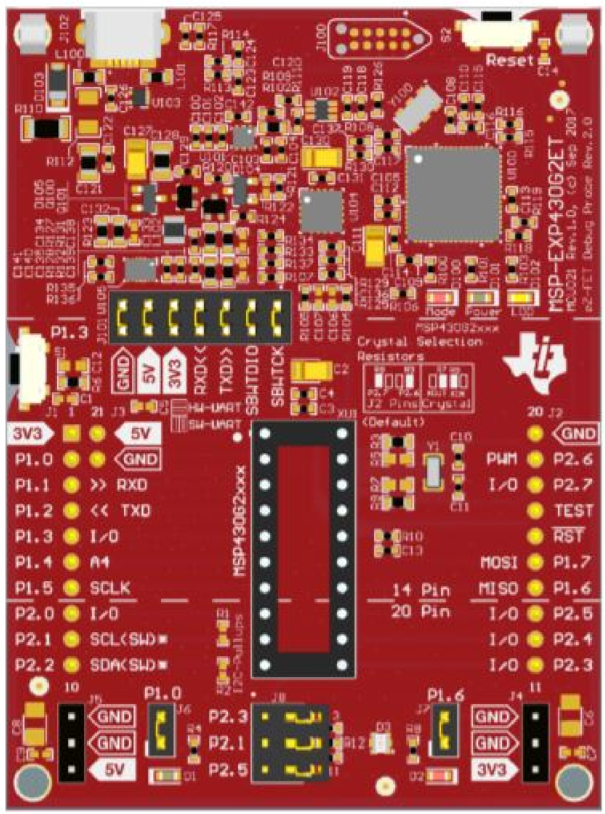

Figure 6: Image showing the original configuration of the MSP430G2553

7 Test Setup

7.1 Test Data

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	LED Color
0x05	0xFF	0x00	0x00	0x0D	Red
0x05	0x00	0xFF	0x00	0x0D	Green
0x05	0x00	0x00	0xFF	0x0D	Blue
0x05	0xFF	0xFF	0xFF	0x0D	White

8 Design Files

8.1 Schematics

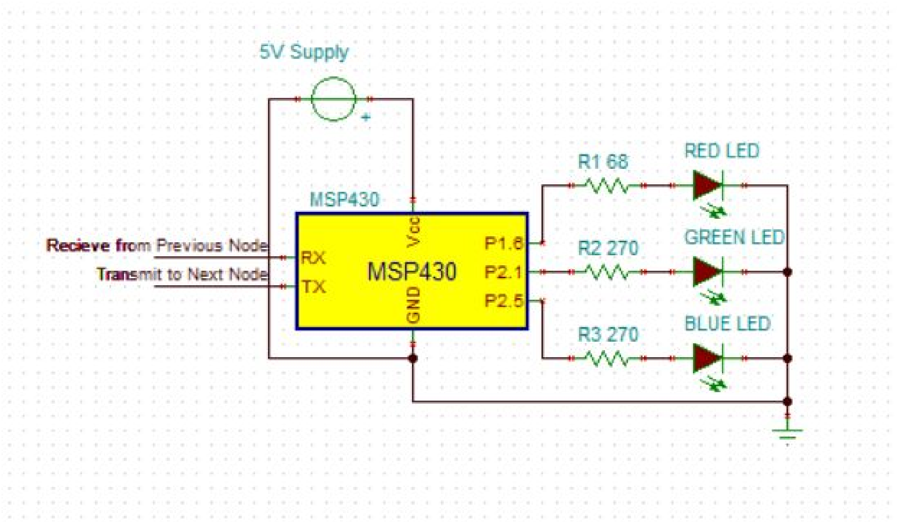

Figure 7: Schematic showing how the MSP430 connects to the LED and how it would connect to other nodes

8.2 Bill of Materials

- MSP430G2553 x1
- Common Anode 4-pin RGB LED x1
- 68 Ω Resistor x2 to limit current for Green and Blue LEDs
- 270 Ω Resistor x1 to limit current for Red LED

- Various Jumper Cables (M-M) (F-M) (F-F) for various connections