Stranger Things Light Wall

Christopher Satriale and Don Stahlberger
Rowan University

October 22, 2018

1 Design Overview

The recreation of the light wall from Netflix's "Stranger Things" will be accomplished by programming an RGB LED on the MSP430G2553. Our individual LED will receive data and send data to another processor and LED in the chain. Altogether, each group in the class will connect and display the original data sent in the form of an LED.

1.1 Design Features

The following features will be utilized and displayed in the lab.

Each are for the MSP430G2553:

- UART
- PWM
- C Programming
- Circuit Analysis

1.2 Featured Applications

- Transmitting Data
- Receiving Data

1.3 Design Resources

- Stranger Things Light Wall
- Main.c

1.4 Block Diagram

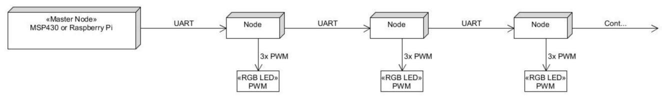

Figure 1: MSP430G2553 Data Processing

1.5 Board Image

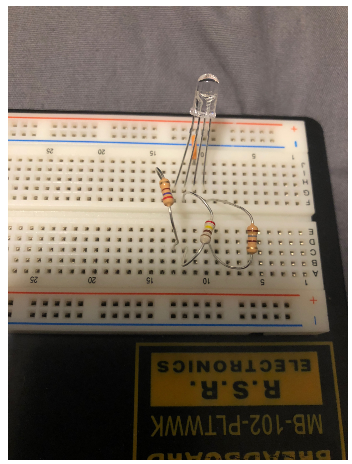

Figure 2: LED Breadboard setup

2 Key System Specifications

PARAMETER	SPECIFICATIONS	DETAILS
Light up RGB LED	Correct amount of bits are sent	If too many bits or not enough bits are passed through the incorrect color will show

3 System Description

Data must be sent from one processor to another and decode the data into different duty cycles for an RGB LED. Each LED, R, G, and B will have a duty cycle determined by the 3 least significant bytes of data. The bytes will be removed after being processed and converted into duty cycles. The string will then be transmitted to the next processor in the chain.

3.1 Detailed Block Diagram

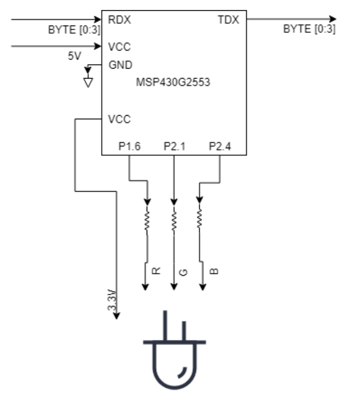

Figure 3: MSP430G2553 and RGB LED

3.2 Highlighted Devices

- MSP430G2553
- Resistors
  - 180
  - 240
  - 270
- 5V Source
- Common Anode RGB Led
- Jumper Cables

3.3 Device - MSP430G2553

The MSP430G2553 is used for this lab due to the ease of use with the clocks.The clock frequencies allow for easier calculations with the duty cycle for each lead on the LED.

3.4 Device - RGB LED

The LED used is a common anode RGB LED meaning it is connected to 5V power supply and each node RGB to a resistor. The resistors are in place to regulate the voltage to each diode. Each diode, R, G and B have a different voltage drop that is required to overcome before light emits.

4 SYSTEM DESIGN THEORY

The system to replicate the Stranger Things ”Light Wall” is composed of two major parts, the processor and the supporting circuit. The responsibility of the processor is to receive data, process and finally transmit data. The circuit is necessary to take the output from the processor and output it as a unique color of an RGB LED. The following sections will go into more detail about the functions of each unit.

4.1 Design Requirement 1

The processor used in the ”Light Wall” circuit is MSP40G2553. The pins addressed in the code are P1.6, P2.1, and P2.4 as well as the TXD and RXD for UART communication. P1.6, P2.1 and P2.4 is connected to TimerA which triggers the pin on every high edge of the clock because of the up mode configuration of the clock. Each pin is connected to a different color of the RGB LED. Therefore, the LEDs change their output on each rising edge of the clock. The TXD and RXD pins are used for the UART where TXD is the transmit pin that sends data to the next processor in the chain. Whilw RXD is the receive pin that accepts data from the previous processor or computer.

The PWM implemented in the code controls the color of the LED by dimming each R, G and B to produce a single color. The frequency of the PWM is controlled by the value of the capture compare register for each R, G and B. The value of the capture compare registers rely on the data received by the processor. The frequency is also referred to as the duty cycle which is the amount of time the diode is on per clock cycle.

4.2 Design Requirement 2

The supporting circuit in the ”Light Wall” is composed of 3 resistors and a common anode RGB LED. It is important to note the forward voltage for each diode. This value must be overcome before the diode lights up. For red and green the drop is 1.8V and for blue the drop is 3.3V. Therefore, the voltage across each diode is Vcc - Vdrop. The resistor values chosen were 180 for blue, 240 for green and 270 for blue. The resistor values allow the diodes to be the same brightness when at a full duty cycle.

5 Getting Started/How to use the device

In order to interface with the processor it must be connected via USB to either a computer or to the transmit pin of a previous processor in the chain. If connected to a computer, the COM port the MSP430 is connected to must be sent a package of data. The first number in the sequence of data tells the processor how many bytes of information there are. After, each number sets the value for each color's capture compare register which then controls the duty cycle for each. The final number in the sequence is the end byte.

If connected to a previous processor, the transmit pin of the previous processor must be connected to the receive pin on the current processor. The transmit pin on the current processor must be connected to the receive pin on the next processor.

The electrical circuit must be set up on a breadboard. The positive pin from the LED connects via a jumper to the 3.3V output from the MSP430G2553.

6 Getting Started Software/Firmware

The software and hardware being implemented in this project is relatively straightforward. There are several different components. First, on the software side, this project was written in the C language. The code was developed for use with the MSP430G2553 as stated in other sections. The different sections of the code will be elaborated upon below. As for the hardware, a hardware based PWM was used to control the LEDs, an off board RGB LED was used, and a breadboard was used to connect the MSP430G2553 with the LED.

6.1 Software design

6.1.1 Methods

The code is split up into several different parts. There are three methods used to initialize different parts of the code. There is a PinSetup method to initialize the different pins to be used later in the code. In this method the pins 1.6, 2.1 and 2.4 were initialized to be connected to Timer A. Pin 1.0 was also initialized to be used in debugging. There is a timerASetup method to initialize the different timer A's being used by the code. In this method, two identical Timer A's are set up using the secondary master clock, the internal divide by 4, and being run in up mode. The capture compare registers for each timer are set to run in set/reset mode to allow the PWM to work correctly. Last, there is a UARTsetup method to initialize the different parameters of the UART such as the BAUD rate, the input, the output, and the interrupt enable.

6.1.2 Interrupts

Besides methods, there are also interrupts being used to enable the micro controller to use the files being received through UART. There are two interrupts used in this code, an RX interrupt and a TX interrupt. The RX interrupt triggers whenever a message is received through UART, the TX interrupt triggers whenever a message is sent. In this code, the TX interrupt is used for debugging purposes to see whether a message is sent or not. The RX interrupt takes care of everything else. In the interrupt the message received will be walked through until it ends. It will take the first byte, subtract three, then pass it on to the next node. It will then utilize the next three bytes to set the duty cycles for the RGB parts of the LED. The rest of the bits will be passed along through to the next node. This is accomplished using a counter set to the value of the first byte of the package. It is then decremented until it hits zero. If statements are used to compare the counter to the total number of packets in the message.

6.1.3 Main

The main of the code is simple. The watch dog timer is stopped, the methods are called, and global interrupts are enabled.

6.2 Hardware Design

The hardware design aspect of this project was using a hardware based PWM to enable the duty cycles of the LED to be changed. This is accomplished by using physical pins connected to the Timer A and setting the capture compare registers to different values and putting them in set/reset mode. The set/reset mode will allow the LED to be on when the timer hits the value in a certain CCR and then reset when the timer hits CCR0. Therefore, the timer will run from 0 to 255 (the value set in CCR0). The other CCRs are set according to the values of the package received over UART. Then the LED will have the correct duty cycle values and will be dimmed/brightened accordingly.

6.3 Communicating with the Device

The communication with the MSP430G2553 in this project happens over USB. More specifically, UART. The baud rate used for our project was 9600 because it was specified in the design documentation. UART stands for Universal asynchronous receiver-transmitter, basically a message is either sent or received.

6.4 Device Specific Information

The device specifics for the MSP430G2553 are minor. Most of the code can be used on other boards with little changes. The major specific is the pin connections. Pins 1.6, 2.1, and 2.4 are hooked up to the Timer A clock and represent the different colors of the LED. The Timer A itself is a specific of the MSP430G2553 and several other boards but may need to be changed if the code needs to be implemented on the MSP430F5529.

7 Test Setup

To set up the test of the ”Light Wall,” follow the directions in sections 5 and 6. Also, insert the processor and circuit in a sequence with multiple devices.

8 Design Files

8.1 Bill of Materials

- MSP430G2553
- Resistors
- Breadboard
- RGB LED
- Jumper cables