Milestone 1: Stranger Things Light Wall

Shane Price and Kieran O'Connor
Rowan University

October 21, 2018

1 Design Overview

This project involved recreating a scene from a famous television show, Stranger Things. The scene that was recreated was the scene where Will's mom was communicating with Will using lights hung above letters and Will turned the light on to each letter to send messages. The project was similar to such, the task was to program a RGB (Red Green Blue) LED to be able to decipher a "packet" of hex code and to turn the RGB LED on of different values for each color. This was intended to be used by UART (univeral asynchronous receiver-transmitter) communication and in a "chain" of micro-controllers so that our controller would turn the correct color dependent on its place in the chain, from receiving the "packet" from the controller before and sending the "packet" to the controller after.

1.1 Design Features

- Change the color and brightness based on instructions given through the input of either the RXD input pin or the micro USB input
- Can transmit to another micro-controller giving the micro-controller the ability to use the next instructions in the set
- Transmits back information to the computer used verifying the instructions were used.

1.2 Featured Applications

- Communication
- Emergency Services Lights
- Cool Television Scenes

1.3 Design Resources

In this project the team used two resources to help build our code, one was a the TI resource explorer for echo information and the other was a website describing how to program a MSP430 for UART communications. These can be found below, respectively.

`http://dev.ti.com/tirex/#/Device/MSP430G2553/?link=Software%2FMSP430Ware%2FDevices%2FMSP430G2553%2FPeripheral%20Examples%2FRegister%20Level%2Fmsp430g2xx3_uscia0_uart_06_9600.c`

`https://www.embeddedrelated.com/showarticle/420.php`

The code for the finished product of the project can be found below.
`https://github.com/RU09342-F18/milestone-1-russell-s-muscle/blob/master/Milestone_StrangerThings/main.c`

1.4 Block Diagram

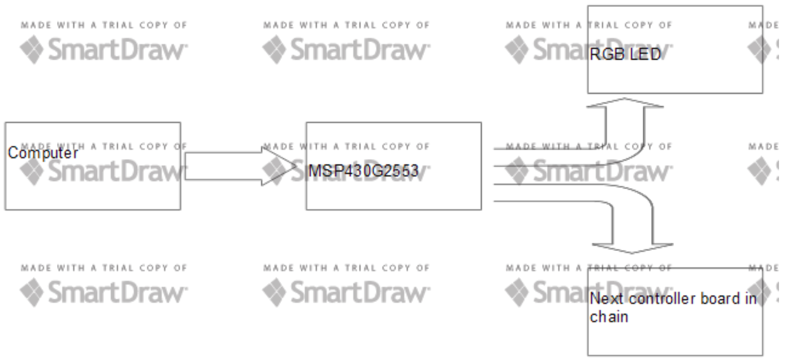

Figure 1: System Block Diagram

1.5 Board Image

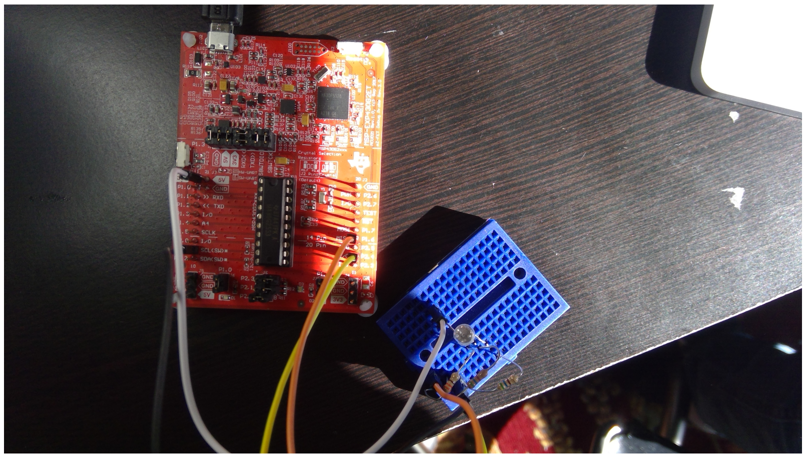

Figure 2: MSP430G2553 Launchpad and RGB LED

2 Key System Specifications

PARAMETER	SPECIFICATIONS	DETAILS
Red LED Voltage	1.8V	Minimum voltage needed to operate explanation about this parameter
Green LED Voltage	3.0V	Minimum voltage needed to operate explanation about this parameter
Blue LED Voltage	3.0V	Minimum voltage needed to operate explanation about this parameter
Input Voltage	3.3V	Input voltage to the system
Packet Size	5 Bytes	The minimum packet size for one RGB LED is 5 bytes, first byte specifying the amount of bytes in the packet, next three bytes specifying the values of RGB, respectively, and the final byte is the end of packet byte.

3 System Description

This project is trying to improve the understanding of the use of UART communications as well as programming a RGB LED in a chain.

3.1 Detailed Block Diagram

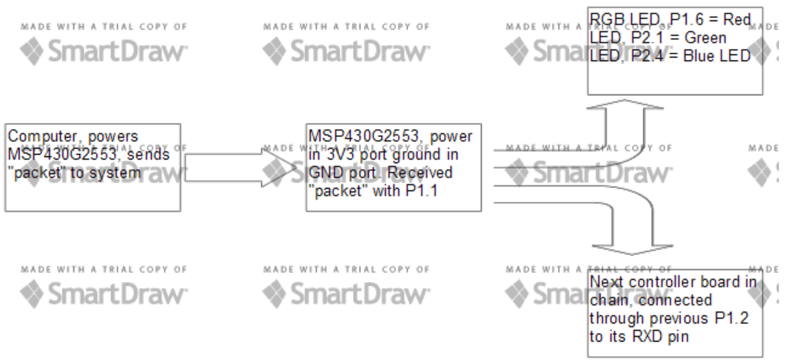
Computer, powers MSP430G2553, sends "packet" to system
MSP430G2553, power in 3V3 port ground in GND port. Received "packet" with P1.1
RGB LED, P1.6 = Red LED, P2.1 = Green LED, P2.4 = Blue LED
Next controller board in chain, connected through previous P1.2 to its RXD pin
MADE WITH A TRIAL COPY OF SmartDraw

Figure 3: Detailed System Block Diagram

3.2 Highlighted Devices

- MSP430G2553
  - Processes the designated package, applies the settings to the desired pins and ultimately determines what color is on and the level of light displayed.
- RGB LED
  - The LED is the output of the instructions processed by the MSP430G2553. The LED is common anode with the long end attached to 3.3V and the other three attached to resistors going to the pin matching the color desired.

3.3 MSP430G2553

This TI micro-controller used in the system processes the code and the inputted package. Using this component's two timers and three capture compare registers for each timer the desired effect was possible. The clocks used for each of the two timers was the Sub Main clock which is a 1 MHz clock.

4 SYSTEM DESIGN THEORY

This project has two main components in the chain of micro-controllers. The micro-controller and the RGB LED. Each are described below on their functionality and how they each interact in the system of chained micro-controllers.

4.1 MSP430G2553

The main driver of this project is the MSP430G2553 micro-controller. The micro-controller receives a ”packet” of information from an outside source such as a computer or another micro-controller. The information is sent through P1.1 on the MSP430 micro-controller used. The information was deciphered as the first byte of information describing how many bytes are in the ”packet” the next three bytes specify what values of RGB, respectively, these set the overall color of the LED. The colors are sent by P1.6 for the red LED, P2.1 for the green LED, and P2.4 for the blue LED. Then a byte dictating the end of the ”packet”. Once the end of the fifth byte is read and there is still information in the ”packet” it is sent through P1.2 to the next micro-controller in the chain. The first byte will now be adjusted to show how many bytes are left after the previous micro-controller used the bytes.

4.2 RGB LED

As described in the previous section the LEDs are driven through the pins P1.6, P2.1, P2.4. The longest lead on the LED was power, second longest was green, third longest was red and shortest was blue each of the colors was connected to ground from the pins on the MSP430 board. The LEDs had resistors in series with them, determined by using ohms law. The current was 20mA as it being the max forward voltage and using the minimum voltage ratings of 1.8V for red and 3.0 for green and blue. The results ended up being 15 ohms for the green and blue LEDs, and 75 ohms for the red LED. Using the resources at hand the team got the closest values to the calculated resistances.

5 Getting Started

Once the device is programmed and connected to the RGB LED the next step is to send information to the micro-controller and have that information displayed. This can be done by following the information specified below.

5.1 Terminal Program

In order to send information to the micro-controller you will need a terminal program on your computer. The terminal program used by the team was Real Term, once on the program specify the port the controller is connected to. The controller should be connected using a UART communication cable to each of the required pins; 3V3, GND, P1.1 (RXD), P1.2 (TXD). Using real term send a "packet" of information to the system, an example packet would be as follows: 0x05 0xFF 0x00 0x00 0x0D. This example packet would turn on the red part of the LED with full intensity. The final test to ensure it is completely working, 0x0D will be transmitted back to Real Term. This is the stop character, and will be transmitted back since the "packet" was only long enough for one micro-controller, if the "packet" was longer it would return the values left over and the first byte would be "-3" from the original value.

6 Getting Started Software

As stated before the main way to communicate with the system is by using a terminal program using UART. UART was used due to its simplicity to implement with the MSP430 boards as well as it being a common form of data communication.

6.1 Communicating with the Device

The devices communicate using UART, UART is a standard simple form of communication for data. UART consists of four wires, two for power and ground and two for transmitting and receiving. These are unidirectional data lines, meaning that they will not function properly if reversely connected.

The device is connected with an outside system that sends data through UART cable allowing the device to then display to information specific to that device. While capturing and displaying its information it sends the left over information to the next device in the chain.

6.2 PWM on Device

A necessary function on the device was using the hardware PWM. This was used to set the timers to a specified frequency (BAUD rate of 9600) to allow the LED to be toggled at certain intervals. The BAUD rate was specified to be 9600, from the requirements of the project. The interval for the LED to be toggled was the frequency of the clock used (SMCLK)/9600; 1MHz/9600, we set the rate to be 104 by rounding down.

6.3 Device Specific Information

The micro-controller used was the MSP430G2553 this is an easy micro-controller to use however it only has three capture compare registers per timer because of this both timers would have to be used. With each of the timers set to the same settings for the SMCLK, Divider 4, UP Mode, and their capture compare register 0 set to 255 they operated at the same frequency.

The pins used were determined by the capture compare registers with P1.6 for red set to timer 0, capture compare register 1 because the microprocessor made this possible using select in the original code. P2.1 was set to timer 1, capture compare register 1 for green and P2.4 was set to timer 1, capture compare register 2 for blue.

7 Test Setup

Testing was done by using the terminal program, Real Term. Necessary steps to take before testing was to specify the ports of the device, using device manager and input that to Real Term. Then sending a packet of information. Initially send a packet of 5 bytes to ensure the LED comes on and the proper data set is transmitted. then sent a longer packet to check the size that is returned to verify it can handle extra bytes, and the same by sending less bytes than necessary.

7.1 Test Data

The data collected for this project was to see the output of the code which was displayed by the LED's change in colors according to the specified instructions. There was also a shown transmitted if connected to computer with Real Term if the system was working properly with one instruction 0x0D should be received from the microcontroller. Another way to check if the transmitted data is correct is connecting the board to another board with the same code using the RXD of the other board connected to the TXD to the original board with a common ground. If multiple instruction sets in the packet are sent and both LEDs operated as designated the code is transmitting properly.

7.2 Schematics

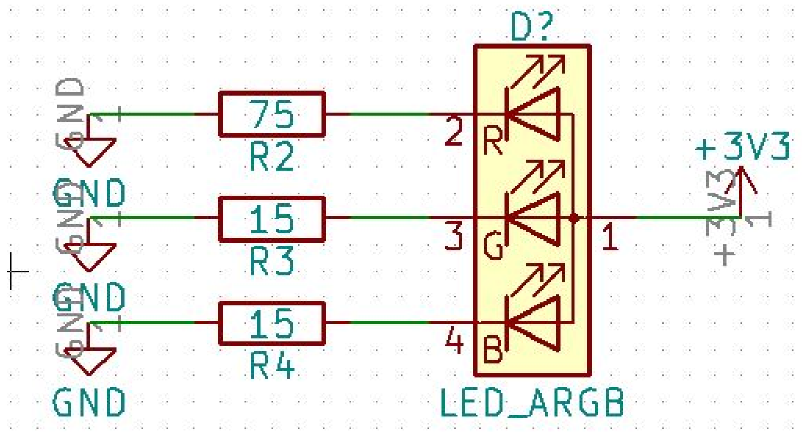

Figure 4: Schematic of LED Connection

This figure describes how the common anode RGB LED was connected. Where each area connected to ground they were connected to the corresponding pins specified for each color on the MSP430G2553. These would pull them down to ground and allow for the circuit to be complete and the LED to turn on.

7.3 Bill of Materials

- 1 75 ohm resistor
- 2 15 ohm resistors
- 1 MSP430G2553
- 1 MSP430G2553 Launchpad
- 1 UART Communication Cable/ Jumper wires when in a chain
- Real Term Terminal Program
- 1 RGB LED
- 1 Breadboard

8 References

- Family User Guide
  `http://www.ti.com/lit/ug/slau144j/slau144j.pdf`
- MSP430G2553 Datasheet
  `http://www.ti.com/lit/ds/symlink/msp430g2553.pdf`
- RGB LED Datasheet
  `https://www.sparkfun.com/datasheets/Components/YSL-R596CR3G4B5C-C10.pdf`
- TI Resource Explorer UART 9600 Echo Example Code
  `http://dev.ti.com/tirex/#/Device/MSP430G2553/?link=Software%2FMSP430Ware%2FDevices%2FMSP430G2553%2FPeripheral%20Examples%2FRegister%20Level%2Fmsp430g2xx3_uscia0_uart_01_9600.c`
- Example UART Configuaration Code
  `https://www.embeddedrelated.com/showarticle/420.php`