– Milestone 1 –

Application Note Template

Will Cronin and Matt Whitley
Rowan University

October 22, 2018

1 Design Overview

Creating inventive ways of communication to compensate for a lack thereof is no simple task. In the Netflix show ”Stranger Things” when Will Byers was stuck in an alternate dimension, his mother used Christmas lights arranged similar to a Ouija board to communicate with him. Though only fiction, this feat inspired the creation of a system of addressable LEDs that took an input and gave an output back to the next LED. This is done by using the MSP430F5529 microprocessor to receive bytes of code, take certain bytes and convert those to colors for the LED, and then send the reduced signal to the next node in the system. This app note covers the functionality of the microprocessor and LED as well as explaining the connection of these RGB nodes and the signal that is passed through them.

1.1 Design Features

These are the design features:

- Receives a length of bytes that determine the color of the LED
- Takes the second, third and fourth bytes out of the string
- Converts them to the color codes for red, green and blue with each byte representing a color code for each
- Sends the bytes to the LED's nodes and have the LED light up in the appropriate color
- Takes the remaining bytes in the signal and transmits them to the next node in the system

1.2 Featured Applications

- Can be used for programming an LED display

1.3 Design Resources

https://github.com/RU09342-F18/milestone-1-spaghetti-forgetti

1.4 Schematic

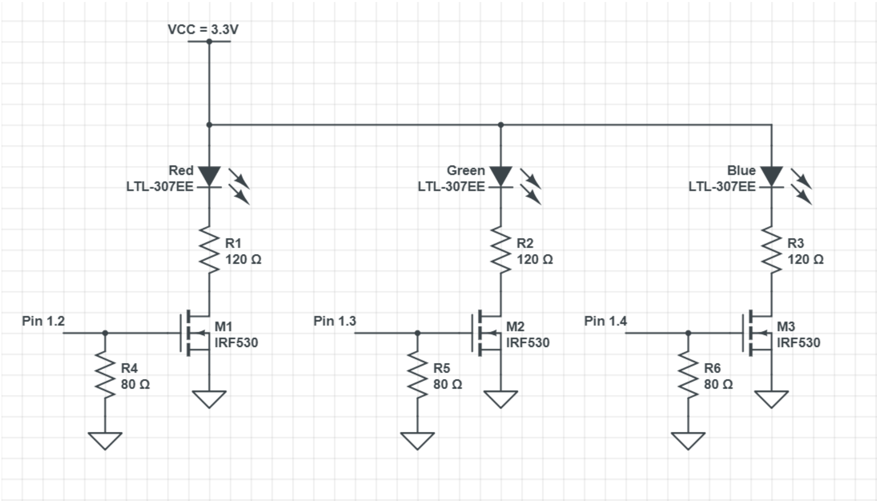

Figure 1: Schematic representation of the MSP430 interfacing assembly

1.5 Board Image

See Figure 2

2 Key System Specifications

PARAMETER	SPECIFICATIONS	DETAILS
SMCLK Frequency	1 MHz	PWM timer speed
TA0CCR0	0xFF	Period of PWM
TA0CCR1	Pin 1.2	Red LED time
TA0CCR2	Pin 1.3	Green LED time
TA0CCR3	Pin 1.4	Blue LED time

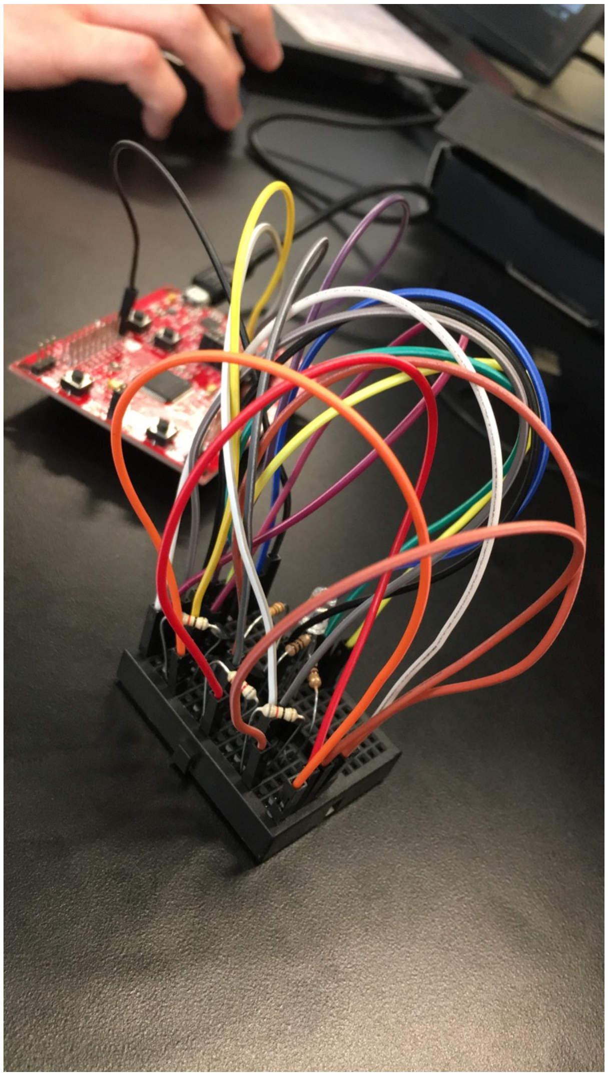

Figure 2: Board Image

3 System Description

The problem addressed by this lab is to create a program that would have an RGB LED light up a certain color dependent on what code it receives using the MSP430 processor. This microprocessor and LED need to work in series with other similar nodes where each receives the signal, takes the necessary information out about their LED color, and then transmits that signal to the next node.

3.1 Detailed Block Diagram

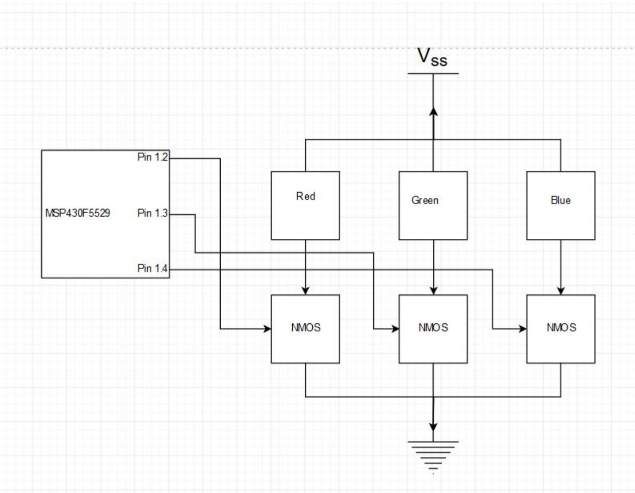

Figure 3: Block Diagram

3.2 Highlighted Devices

- MSP430F5529 - This microprocessor is used to transmit and receive signals and use a portion of the signal to light the LED to a certain RGB value based on said signal.
- LED and driving circuitry - An RGB LED was used for this lab, along with a configuration of resistors and NMOS transistors.

3.3 MSP430F5529

The processor used for this device is the MSP430F5529. The 5529 is better for this application due to its four timer registers. Other MSP430 boards only have three. The 5529 is configured with UART TX and RX (transmitting and receiving) nodes, which is how the strings of hexadecimal values are sent to the LED nodes. As seen in Figure 1, pins 1.2, 1.3, and 1.4 send their RGB values to the red, green, and blue LED nodes respectively. The initial RX node receives its values from the RealTerm Serial/TCP Terminal. These values are then transmitted to the next node via the TX path.

3.4 Device/IC 2

The micro-controller has two different voltage outputs: 3.3V and 5V. While the 3.3V can light the LED, the 5V makes it brighter and more distinct. Therefore, it is more prudent to use 5 volts to power the LED. However, the micro-controller cannot output 5V through the pins. This is solved using an NMOS at each input terminal of the RGB LED, creating a low-side switch. Each NMOS also has a 80 Ohm resistor attached to the gate to prevent floating when the micro-controller is switched from on to off.

4 SYSTEM DESIGN THEORY

The main design challenges with this assembly include interfacing the MSP430 with an LED. Through the use of a ”Convert-o-box,” or interfacing network, a useful voltage is achieved to power the LED. Additionally, a UART transmission and reception data path needs to be utilized to send and receive hexadecimal values. These values set the RGB inputs and are transformed to duty cycles in order to create the proper LED color.

4.1 Design Requirement 1

Creating a useful voltage from a smaller microprocessor is imperative in several applications, and is important to know how to do on a smaller scale. The interfacing network for this LED assembly is comprised of three NMOS transistors. The gates of each transistor are fed by transmitting pins on the MSP430, with a 120 Ohm resistor in parallel going to ground. These pins send bytes that correspond to each RGB component of the LED. Each drain of a transistor are connected to either the red, green, or blue LED components, with an 80 Ohm resistor in series. I.E. one transistor is connected to the red LED, one is connected to the blue LED, etc. The source of each transistor is connected to ground.

This arrangement of NMOS transistors allows the 3.3V MSP430 output to light the RGB LED, which runs at 5V.

4.2 Design Requirement 2

In order to send and receive the data from the MSP430, a UART lines need to be utilized to send the string of hex values. The UART RX line receives the string of bytes, while the UART TX line transmits between each node. Each node reads the least significant bits from the string and uses those values to set the RGB values.

5 Getting Started/How to use the device

5.1 Materials Needed

- MSP430F5529 and attached design assembly
- Micro-USB to USB connector

- Computer or laptop with Code Composer Studio open and running the supplied code
- RealTerm Serial/TCP Terminal software running on the same computer

5.2 Connecting the Assembly

The MSP430 and breadboard assembly will be given. To begin experimentation, connect the assembly to the computer or laptop using the Micro-USB to USB connector. A small green LED should flash on the MSP430 indicating that it is being supplied power.

6 Getting Started Software/Firmware

6.1 Software Needed

- RealTerm Serial/TCP Terminal
- Code Composer Studio, with given code files

6.2 Using Code Composer Studio

The code needed to run this assembly will be given, and should be opened in Code Composer Studio (CCS). On the main page of CCS, just click the ”Build” icon in the toolbar at the top of the page. This compiles the code. Next, ensure the assembly is plugged into a USB port and run the code. The LEDs on the MSP430 should flash momentarily, indicating that the code is now running on the board.

6.3 Preparing and Sending Inputs in RealTerm

In RealTerm, click on the tab labeled ”Port” and click the drop-down menu marked by the same name. Click the port from the drop-down menu in which the MSP430 assembly is connected to the computer. You can find which port this is in the Device Manager program in your computer's settings.

Then, in RealTerm, click on the tab labeled ”Send.” There will be a bar on the top of that page that you will enter in RGB values. These values must be sent in hexadecimal form, and there must be a space in between each number. The first number you enter will be the number of packets you are sending to the MSP430. The second number is the value for the Red component of the LED. The third number will be the value for the Blue component, and the fourth value is the value for the Green component. For the RGB component values, you can enter any number up to 0xFF, or 255 in decimal. When the inputs are ready to be sent to the MSP430, click the ”Send Numbers” button, to the right of the input line.

7 Test Setup

The device is set up for testing in 5 steps

- Connect 3.3V and ground from the microprocessor board or an external source to the breadboard
- Connect Pins 1.2, 1.3 and 1.4 to the red, green, and blue pins on the RGB LED respectfully.
- Plug the MSP430F5529 into a computer and flash the code
- Use a program such as PuTTY or Realterm to send a package of bytes to the board
- If there are other nodes in the system, connect the RX pin to the previous board's TX pin and the TX pin to the next board's RX pin.

8 Design Files

8.1 Bill of Materials

- MSP430F5529 x1
- RGB LED (Anode) x1
- Breadboard x1
- NMOS x4
- 120 Ohm resistor x3
- 80 Ohm resistor x4