Milestone 1: Communicating with Will Byers

Michael Maloney and Michael Sorce
Rowan University

October 21, 2018

1 Design Overview

An MSP430 board was configured to send and receive UART commands. These commands were used to drive PWM signals, which in turn lit up an RGB LED. As expected, the code required for this process is not as simple as stated. The lab's objective was to effectively create the code and circuit to drive an RGB LED through UART. Since the product can send and receive data, it is possible to link multiple boards together in a way similar to the blinking Christmas lights in Stranger Things. Since each board needs to be individually addressable in the chain, another challenge is implementing a way to determine what slice of the UART code each board needs to accept and what it needs to then send to the next board.

1.1 Design Features

The design features are as follows:

- UART Compatible
- Full Range RGB LED Color Spectrum
- Chain-able with similar designs
- Individually Addressable in Chain
- MOSFET Pin Protection

1.2 Featured Applications

- RGB LED String

1.3 Design Resources

The following is a link to GitHub page containing the C code that drove this project and additional images of the design: Link to GitHub Page

1.4 Block Diagram

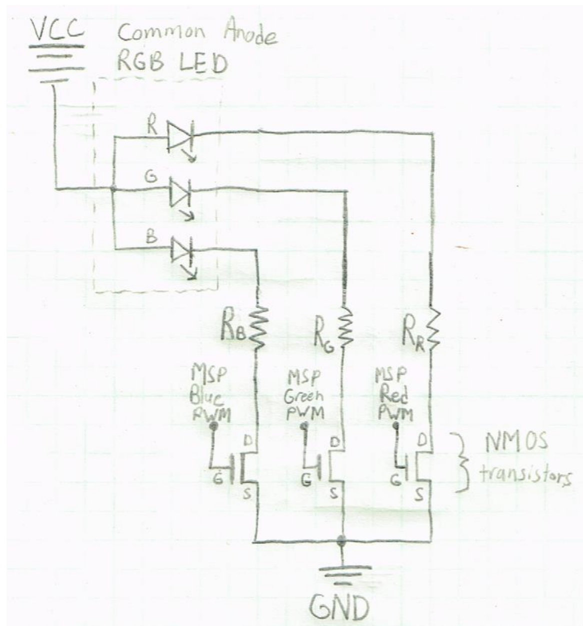

Figure 1: Block Diagram

This figure shows a general block diagram of the external circuit used with the MSP430F5529 to display the projects functionality. Here you can seen what components were used and how they were connected. These components include NMOS transistors, a common anode RGB LED and resistors. The pins from the MSP will be connected to the noted areas, such as the three PWM inputs, VCC, and GND.

1.5 Board Image

Included are images of the breadboard circuit and MSP430F5529. Here the general connections can be seen. The black and white wire on the MSP and breadboard represent ground and Vcc respectively. The red, green, and blue wires sequentially on the MSP to the breadboard are the PWM outputs from the MSP (each representing their respective color for the RGB LED). The green and orange wire on the MSP are the transmit and receive pins used when hooking the boards up in a chain. By using

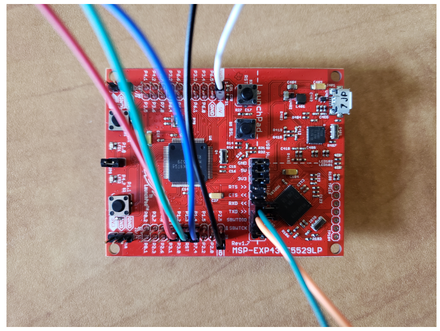

Figure 2: MSP430F5529

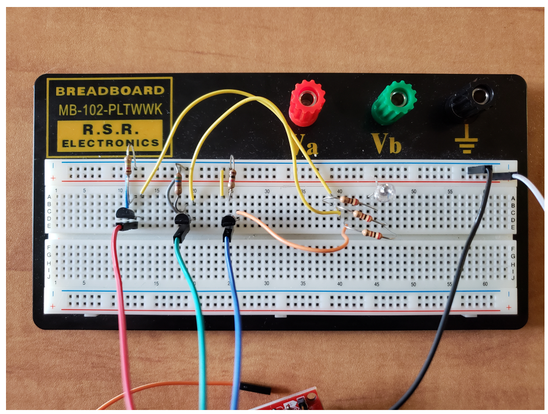

Figure 3: Breadboard Circuit

the block diagram as a reference, one can see how the connections are set up on the breadboard.

2 Key System Specifications

Table 1: System Specifications

Baud	9600bps
Clock Speed	1 MHz

3 System Description

In the first season of Stranger Things, the character Will Byers communicates from another dimension by flashing individual Christmas lights. In a similar fashion, this board coupled with additional boards can be told to light up individual LEDs in the string. This design was further pushed to be able to individually address the boards through UART. The data received through UART would contain a byte with the total number of bytes in the message, and three 0-255 RGB values for each board used. The code would decrement the first value by three, accept its three RGB values, and transmit any remaining data. These RGB values would drive PWM signals for each color. By connecting these PWM signals to external circuits the color of the LED can be controlled through the UART command.

3.1 Detailed Block Diagram

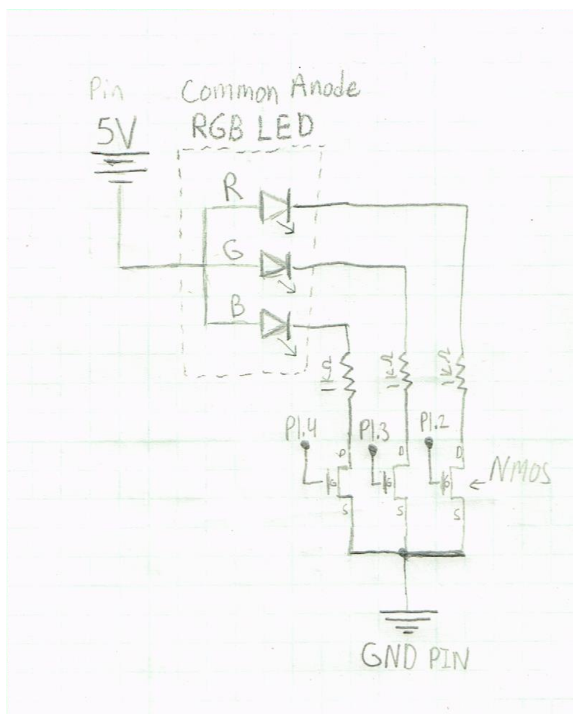

Figure 4: Detailed Block Diagram

This block diagram shows more specific pin connections and voltages used in this design.

3.2 Highlighted Devices

- MSP430F5529
  - This launchpad runs the code and sends and receives data.
- LED - RGB Clear Common Anode
  - An RGB LED with colors modulated by the launchpad pins.

3.3 Device/IC 1

The brain of the whole system resides in the processor. The MSP430F5529 microcontroller is connected to the MSP430F5529 LaunchPad. The microcontroller handles all of the code and logic flashed on the board, as well as processing all of the inputs. The LaunchPad is the circuit board connected to the microcontroller that does all of the work, like signal sending and receiving, but is run by the microcontroller.

More specifically for this lab, the microcontroller runs the code, creates PWM signals, reads UART, and changes UART code. The LaunchPad sends and receives the UART code provided from the microcontroller, and pushes PWM signals from the microcontroller.

3.4 Device/IC 2

The LED used was a clear common anode RGB LED. The forward voltage for the red, green and blue component were 2V, 3.2V and 3.2V respectively. For the circuit design, the anode pin was connected directly to the 5V source. Each color cathode was connected in series to a 1kohm resistor and the drain of an NMOS transistor. The source of each (of three) NMOS was connected to ground. The gates of each NMOS was connected to the respective PWM signals coming from the MSP430F5529. By doing this, the NMOS would allow current to flow at the same PWM frequency that the gate voltage was changing for each color. This way, the voltage for the LED was coming directly from the 5V source and not directly from the PWM pins.

4 SYSTEM DESIGN THEORY

There are essentially three main components of the system: The processor, the MOSFETs, and the LED. In this case, the LaunchPad and microcontroller are lumped together as a single unit, the processor, for simplicity. This whole system can be further simplified into hardware and software components.

The hardware components, the LED and the MOSFETs are driven by the software, i.e. the processor. The processor receives and sends UART signals, while simultaneously driving the hardware components. The code inside the processor is the logic required to interpret the UART code and convert it into useable data. This data then drives PWM signals sent directly to the gates of the MOSFETS. The MOSFETs add a layer of protection to the whole board.

Before talking about the necessity of the MOSFETs, the RGB LED needs to be discussed. The LED is common anode, meaning that the modulation of the color comes from changing the lower pin voltages. This, in turn, means that current would be running into the board from the LED if MOSFETs were not used. To combat this, each driving pin from the board was connected to the gate of a MOSFET, creating a low-side switch with the MOSFET, LED, and pin. This means that no issues can arise from a high current running into the board. A higher voltage on the MOSFET gate allows a higher voltage to run through the LED, causing a brighter LED. This was repeated three times, one for each cathode pin of the RGB LED.

4.1 Design Requirement 1

The first design decision made for this project was selecting an MSP430 development board to use. The dev board picked was the MSP430F5529. Pins P1.2, P1.3, and P1.4 were connected directly to Timer1A0 CCR1 - CCR3 (capture compare registers) respectively. This made it easy to hardware PWM to these pins. Additionally they were in close proximity, making the physical design look cleaner and more organized. The RXD and TXD for receiving and transmitting are clearly marked, and there was a multitude of online resources that explained how to use the UART for this board.

Overall, this board was selected because it allowed for a simple approach to the issue at hand. The available resources, pin layouts, and functionality of this board made it the right pick for this assignment.

4.2 Design Requirement 2

For the external circuit, many decisions needed to be made. As seen throughout the lab, the RGB LED is common anode. This means that the LED has a common pin linking the anodes of the red, green, and blue LEDs and separate cathode pins for the three. If connected in the simplest way to the LaunchPad, the anode pin would connect to Vcc and each cathode pin would connect to the launchpad PWM pins through a resistor. This means current would flow into the pins and into the whole board. While this could be an issue, if precautions are taken it is perfectly safe. Another problem still remains, however. Reverse logic is required to light the LED, meaning that a 0V from the pin is a high voltage on the LED. Even furthering the problem is the fact that a high voltage on the pin does not ensure a 0V on the LED because the voltage on the pin might still be lower than the anode voltage.

This is why MOSFETs were used on each pin. The MOSFETs in a low-side switch configuration negate the reverse logic and allow the code to stay unchanged. The pins drive an NMOS gate where a high voltage on the gate allows more current to flow through the drain to source, raising the LED brightness.

This design simultaneously isolates the board from potential current overload on an input pin while keeping the code from switching to reverse logic.

5 Getting Started/How to use the device

For initial set up for this module, make sure the MSP430F5529 development board and external circuits are connected as specified in the block diagrams and descriptions of this report.

If you are creating a chain with this module, first decide if this module is the first of the chain.

If it is first, connect the dev board to a laptop via USB. Using a third party program such as Realterm, data packets can be manually sent over UART to the first board. For this set up no connection needs to be made to the RXD pin on the first dev board. If chaining on additional boards, connect the new board's RXD pin to the TXD pin of the preceding board. Continue to connect the TXD of the board you are on to the RXD of the new boards you add to make the chain longer.

If you are adding this module to an existing chain, simply connect the RXD pin of this module to the TXD pin of the preceding board, and connect the TXD pin to the next board in the chain's RXD pin.

Additional notes:
-Manual byte packets need to be initially sent through the USB or from an additional external source.
-All boards need to be individually powered by some means.

6 Getting Started Software/Firmware

6.1 Communicating with the Device

The software is programmed in C using Code Composer Studio. Once the code is uploaded, the board can operate separate from the laptop, as long as an external power source is present.

To interface with the device (either modular or in a chain) one module must be connected to a laptop via USB. Using a third part program such as Realterm, data packets can be sent manually to the first board in the chain. From their, each module will function independently, accepting the data from the previous chain, manipulating it, and sending it to the next link in the chain.

This Realterm simulates a UART signal to the board. This way a user can input a packet however many byte long as they like, and initialize it for the chain to manipultate the output lights.

6.2 Device Specific Information

Specific to this device are the Rx and Tx pins. These pins receive and transmit UART code from other boards. Since the first board always receives code from a microusb port connected directly to whichever MSP LaunchPad is chosen, the receiving pin in those cases are always the same. If the boards are chained together they need to be connected by UART Rx and Tx pins. On the MSP430F5529 the pins are connected at P4.4 and P4.5, which on the LaunchPad correlates to TxD and RxD. These are the specific pins on the board to transmit and receive UART commands from other boards in a chain.

7 Test Setup

Since a major objective of the lab was to make this board modular and individually addressable, the board was connected in a string-like fashion to other boards through UART connections. The first board in the chain was connected to a laptop to initially upload UART code. This code was then run through each of the consecutive boards through the UART connections. If a board ineffectively processed the UART commands or did not send/receive, the chain would break. This chain, again, is similar to the Christmas lights in Stranger Things, but more closely resembles an individually addressable RGB light strip, except much more complex. If a specific UART command with an expected outcome is pushed through the chain without fault, all boards in the string can be considered working, except for the last board. The last board in the chain needs to be tested twice to check if it could send UART code. The last board would simply be switched with any other board in the chain to check this.

7.1 Board Testing

For single board testing, the MSP430F5529 Development Board was connected to a laptop via usb connection. Using a program such as Realterm, data packets could be manually transmitted to the board, which it would receive through UART. Test values were input in hex such as : 0x05 0xFF 0x66 0x05 0xFF. The code on the board would receive this and decipher it as previously explained, decrementing the first byte by 3, using the next three bytes for the RGB PWM values, and transmitting the rest (including the initial byte) through the RX pin. To test this in chain, more board would then be connected to this board through the RX and TX (receive and transmit) pins. Only one board would need some laptop or another external source to receive the initial data message, and the rest can just be chained onto the other boards.

7.2 Test Data

From our class demos, the results show a successful product. A chain of RGB LEDs were hooked up, each controlled by a MSP processor, and each individually addressable. This design was one of the many successful designs that needed to work together for the ultimate goal of making the chain.

8 Design Files

8.1 Bill of Materials

Materials used include:
- Breadboard
- MSP430F5529 Development Board
- Clear Common Anode RGB LED
- Three 2N7000G NMOS : 60V 200mA
- Three 1Kohm Resistors
- Misc. Wires