Milestone 1

Nate Hoffman and Nikola Kosaric
Rowan University

October 21, 2018

1 Design Overview

For this Milestone, UART communication with an MSP430 and PIC18F was utilized to control the PWM signal to an RGB LED. Each node controls a single RGB LED. Multiple nodes can be connected together using UART communication to form a chain of controllable RGB LEDs.

1.1 Design Features

Because of UART communication and PWM control of the LEDs, these are the design features:

- Up to 319 nodes can be connected together
- Brightness control of each of the 3 different colors of the LED
- Possibility of 16 million different colors
- Works with any device that has the ability to utilize serial communication

1.2 Featured Applications

- Room Decorations
- Police Lights
- Back-end to Music Visualizer

1.3 Design Resources

Links to GitHub repositories and data sheets utilized:

- MSP430G2553 GitHub Repository
- PIC18F2520 GitHub Repository
- MSP430x2xx Family User Guide
- MSP430G2553 Data Sheet
- PIC18F2420/2520/4420/4520 Data Sheet

1.4 Block Diagram

Figure 1 shows the connections of multiple MSP430G2553s in a chain of RGB LEDs. VCC should be connected to 3.3 volts. If using the PIC18F2520, 5 volts should be used instead.

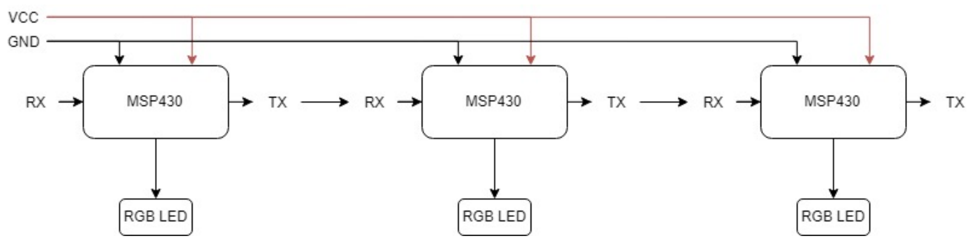

Figure 1: MSP430G2553 System Block Diagram

1.5 Board Image

Figure 2 shows the RGB LED connections on a breadboard.

2 Key System Specifications

Table 1: Table 1. System Specifications

Baud Rate	9600
Maximum Number of Systems Connected	319
Maximum Number of Colors per LED	16.7 Million
MSP430G2553 PWM Cycle	1 KHz PWM
PIC18F2520 PWM Cycle	500 Hz PWM

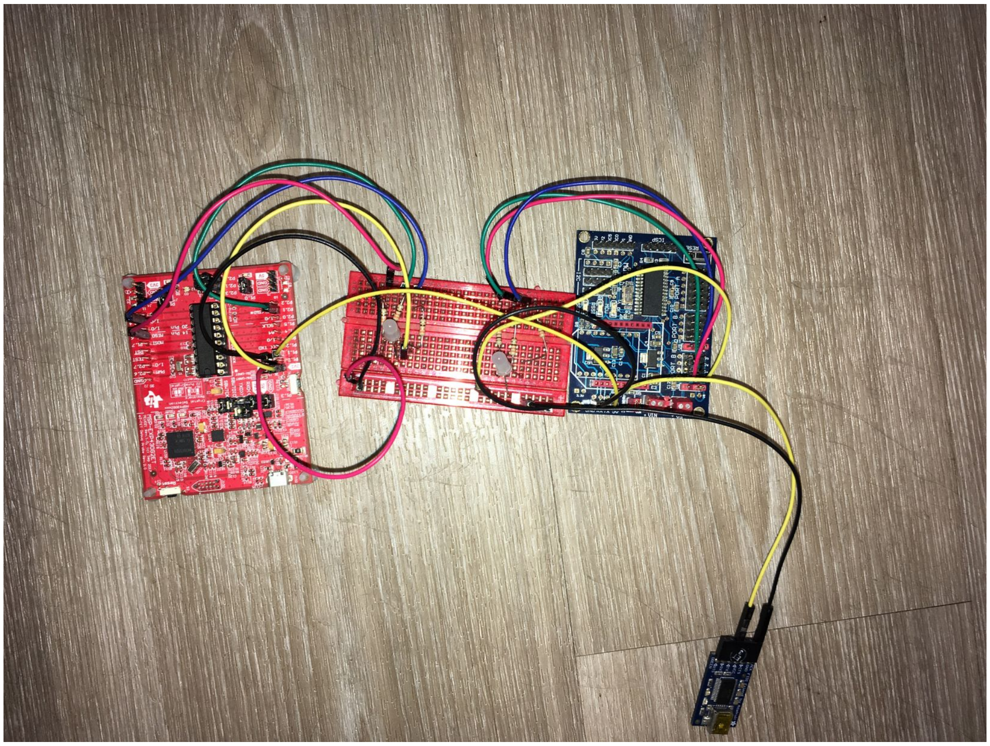

Figure 2: MSP430G2553 and PIC18F2520 RGB LED connections on a breadboard

3 System Description

The issue at hand is finding a means to control an RGB LED through serial communication, allowing for versatility through a variety of processors.

3.1 Detailed Block Diagram

Figure 3 shows the connections of the MSP430G2553 to the RGB LED and the UART lines.

3.2 Highlighted Devices

- MSP430G2553
- PIC18F2520

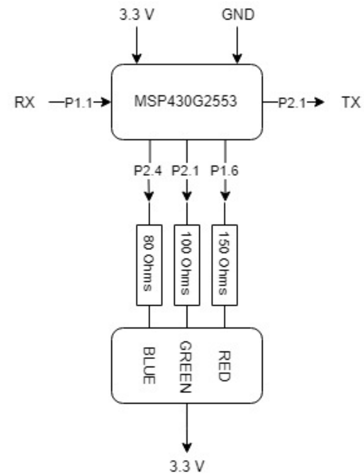

Figure 3: MSP430G2553, connections to an RGB LED, and connections to UART

3.3 MSP430G2553

This processor is manufactured by TI and was used as the primary development board for the class. It is an 8 bit processor and two timer As were used to provide enough CCRs to create a hardware PWM output. As each timer A only had 3 CCRs (one of which is used to set the rate of the PWM) up to two hardware PWM signals could be created per timer.

3.4 PIC18F2520

This processor is manufactured by Microchip and was used to provide an example of a completely different brand of processor for completion of this milestone. It is an 8 bit processor with 4 timers. There are only two CCPRs in the processor, so a software PWM was created to drive the three LEDs in the RGB LED. The clock of the processor runs at 40 MHz and the processor completes 10 MIPs.

4 SYSTEM DESIGN THEORY

As this milestone dealt with both PWM output to an RGB LED and UART communication, those were the two main design requirements.

4.1 PWM Output

The PWM output for each color had to have 256 different values to give 256 different brightnesses. This is the number of different values a byte or char holds. To achieve this result, timers were used to count through all the different values and turn on or off the LED.

For hardware PWM, the CCRs in the MSP430 were used to automatically set and reset the state of the output. This did not require configuring interrupts as the processor has internal circuitry to run the output. This PWM ran at 1 kHz.

For the software PWM, the CCPRs in the PIC18F were used alongside interrupts to calculate the new states of the output LEDs. In each interrupt trigger, the processor recalculated if the LED should be on following the duty cycle of the LED. This PWM ran at 500 Hz due to reaching the limits of the processor to handle 3 PWM channels. If more channels are added, the frequency would most likely have to be lowered.

4.2 UART Communication

UART communication ran at 9600 bps with one stop bit and no parity bit. The processors received the data one byte at a time, triggering an interrupt each time it got a complete byte. The communication protocol specified that:

- The first byte defines the total size of the packet received (inclusive of this first byte and the last byte)
- The next three bytes defined the red, green, and blue colors of this node
- The remaining bytes were passed along to the next nodes
- The last byte is a carriage return (0x0D)

The code kept track of the number of bytes remaining in the packet, and used that to determine what action to perform. If a wrong packet size was sent to the processor, calculations will be off and unexpected results will occur. If the packet only specified the data for one node, then nothing will be transmitted to the next node. A check is performed to make sure that the packet size is a valid size before sending to the next node. It will still set the LED values of the current node if it can, but prevents a problem with the packet from cascading down the chain.

On the PIC18F, if an overflow or framing error occurred, the byte would be ignored and the UART would be reinitialized to clear itself out.

5 Getting Started/How to use the device

The three primary connections to the boards are power, UART, and PWM. If any of these components are missing, the processor will perform unexpected functions. Follow the connections in figure 3 to properly connect the processor. It should be noted that the RGB LED is common anode, so the common pin should be connected to 3.3 volts.

6 Getting Started Software/Firmware

Users should refer to section 4.2 on the UART communication for communicating with the software that is running on the processor. Provided the correct packets are sent to the board over the UART line, the brightness and color of the RGB LED should change. Any remaining bytes in the packet are sent to the next node.

7 Test Setup

7.1 MSP430G2553

To set up the MSP430G2553, the LED must connected as shown in Figure 3. The anode pin of the RGB LED gets connected to 3.3V. The red pin of the LED gets connected to a $150\,\Omega$ resistor and then to P1.6 of the MSP430. The green pin of the LED gets connected to a $100\,\Omega$ resistor and then to P2.1 of the MSP430. The blue pin of the LED gets connected to a $80\,\Omega$ resistor and then to P2.4 of the MSP430.

In order to connect an MSP430 to another MSP430 to test adding more LEDs to a chain, the TX pin (P1.2) of the first MSP430 must connect to the RX pin (P1.1) of the second MSP430. This will allow for both MSP430s to communicate with one another.

7.2 PIC18F2520

To set up the PIC18F2520, the LED must connected as shown in Figure 3. The anode pin of the RGB LED gets connected to 5V. The red pin of the LED gets connected to a $150\,\Omega$ resistor and then to B0 of the PIC. The green pin of the LED gets connected to a $100\,\Omega$ resistor and then to B1 of the PIC. The blue pin of the LED gets connected to a $80\,\Omega$ resistor and then to B2 of the PIC.

In order to connect a PIC to another PIC to test adding more LEDs to a chain, the TX pin (C6) of the first PIC must connect to the RX pin (C7) of the second PIC. This will allow for both PICs to communicate with one another.

7.3 Test Data

Figure 4 shows a test of serial communication between multiple devices.

8 Design Files

8.1 Schematics

Figure 5 shows the pin connections to the MSP430G2553.

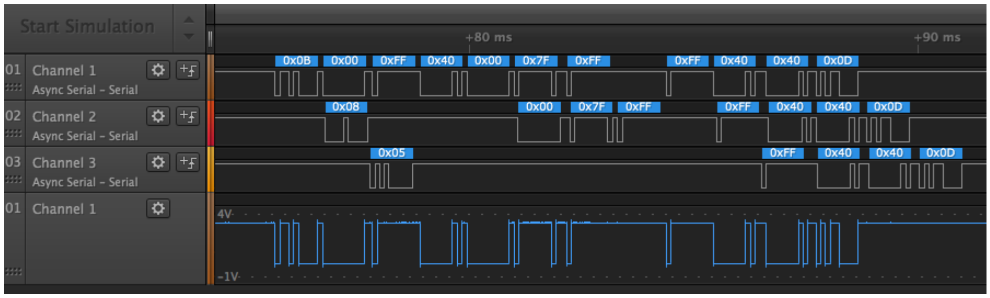

Figure 4: An MSP430G2553 and PIC18F2520 connected together showing the data being passed along in a chain. Channel 1 is the input data, channel 2 is the data between the two devices, and channel 3 is the output from the second device.

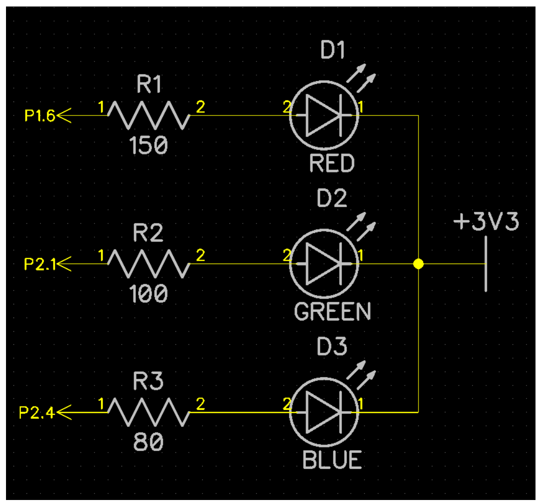

Figure 5: MSP430G2553 RGB LED Schematic

8.2 Bill of Materials

- MSP430G2553 or PIC18F2520
- Common anode RGB LED
- $150\,\Omega$ resistor
- $100\,\Omega$ resistor
- $80\,\Omega$ resistor