Milestone 1 Application Note

K.Limbaga, S.Prendergast,
D.Tocarchick
Rowan University

November 1, 2019

1 Design Overview

For this milestone project, UART (Universal Asynchronous Receiver/Transmitter) communication would be utilized and integrated with an MSP430G2553 micro-controller to transmit PWM (Pulse Width Modulation) signals to an RGB (Red Green Blue) LED. The ultimate goal consisted of having multiple constructed and addressable LEDs capable of communicating with each other via UART, while having several micro-controllers each connected to their own designated RGB LED, simultaneously, creating a node-based system of RGB LEDs. The micro-controller will take 3 bytes of data (3 bytes as the RGB LED could alternate between red, green and blue depending on the PWM value for the three colors) and send the remaining bytes over to the following micro-controllers being interfaced.

1.1 Design Features

The MSP430G2553 micro controller was selected for the milestone project.

These are the design features of the MSP430G2553:

- UART communication
- Hardware PWM
- Brightness control for RGB LEDs

1.2 Featured Applications

- RGB LED lights for display
- Color coded signals

1.3 Design Resources

Kyle's GitHub repository link:
https://github.com/Intro-To-Embedded-Systems-RU09342/milestone-1-KaiPhilippines
Sean's GitHub repository link:
https://github.com/Intro-To-Embedded-Systems-RU09342/milestone-1-Prendergast1
Dawson's GitHub repository link:
https://github.com/Intro-To-Embedded-Systems-RU09342/milestone-1-dawsontorch

1.4 Block Diagram

This is a block diagram of the UART system. First, an 8 Byte signal is sent over UART to the first micro-controller, taking the first 3 RGB data bytes(4,5,6). Followed by sending the remaining data bytes over to the next micro-controller, receiving the final 3 data bytes(1,2,3).

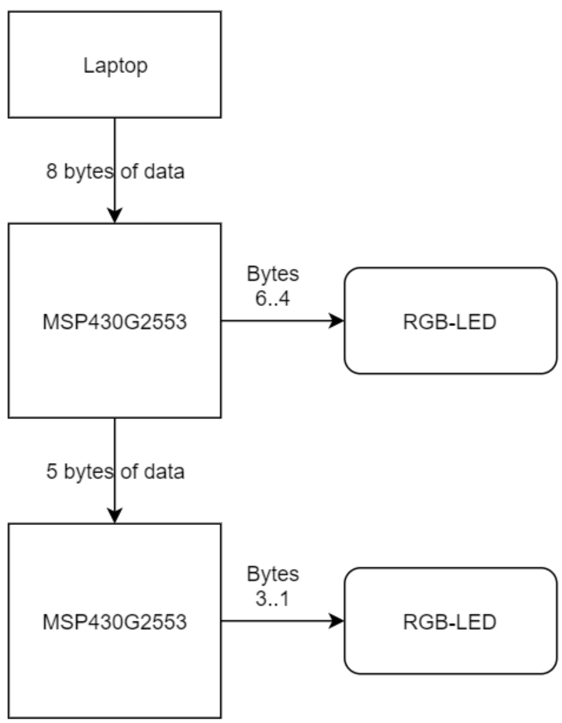

Figure 1: UART system sending and receiving data

1.5 Board Image

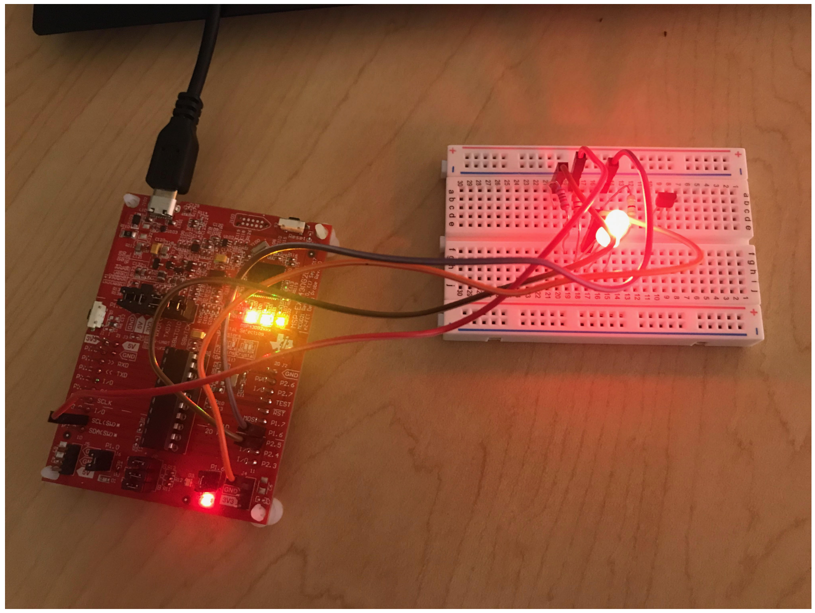

Figure 2: Image of breadboard and micro-controller setup

2 Key System Specifications

PARAMETER	SPECIFICATIONS	DETAILS
RGB-LED	ctrl Color and brightness	Controls RGB color and brightness, by setting each pin of an LED to a specified PWM signal allocated within a UART transmission
UART transmission	Send and Receive transmission	x amount of bytes sent and received between two or more microcontrollers.

3 System Description

The objective of this milestone was to recreate a scene from the popular Netflix TV show, Stranger Things, where different colored Christmas lights were shown to light up at different times as a method of communication between the characters. This scene would be replicated such that a different color would light up on the connected RGB LED based on the given UART signal transmitted to a MSP430G2553 micro-controller. The problem to solve was determining how to send a signal via UART to power and light up an LED based on that signal received. In addition to transmitting a new signal to another interfaced micro-controller to light up another RGB LED based on set parameters in the original signal.

3.1 Detailed Block Diagram

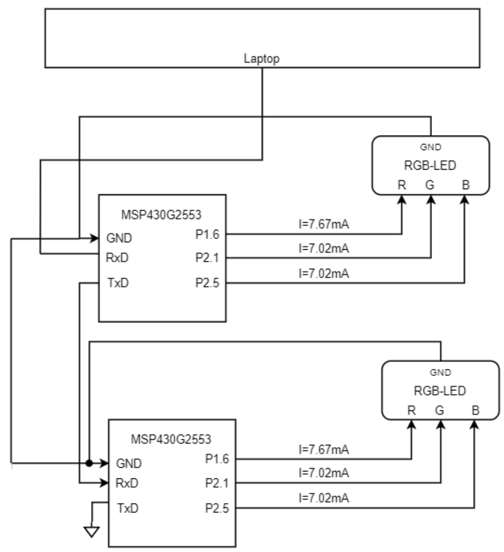

Figure 3: Detailed Block Diagram of UART system

3.2 Highlighted Devices

- **MSP430G2553** - Microcontroller
- **RGB LED** - Common Node

3.3 Device/IC 1

The MSP430 micro controller utilized for this milestone was the MSP430G2553 microcontroller. Manufactured by Texas Instruments, its built-in Timers modules, GPIO (General-Purpose Input/Output) pins and built in universal serial communication interface were utilized for this milestone project. The timer modules in the board were incorporated to set and reset GPIO pins based on values in its Capture/Compare Registers (CCR). When the timer count reaches the desired CCR value, an action is performed. The UART mode allows two or more interfaced micro-controllers to receive and transmit data to each other via RXD (Receive Data) and TXD (Transmit Data) pins found on the MSP430G2553 itself.

3.4 Device/IC 2

The RGB LED is ultimately comprised of 3 LEDs combined into one package. These 3 LEDs correlate with the colors red, green and blue respectively such that if power is driven to any of these three individual terminals on the diode, the LED will turn on to that desired color. Any combination of these color-assigned terminals being on could produce a variety of different colors outputting on the LED, depending on the signal being transmitted. The RGB LED has four total terminals, three of these terminals are utilized for powering on red, green or blue while the fourth terminal is used strictly for grounding purposes. The resistors applied for the green and blue terminals were 470 Ohm resistors while a 430 Ohm resistor would be used for the red terminal. The reason behind utilizing different resistor values for different terminals of the RGB LED is simply because of the differing turn on voltages for the red, green and blue terminals.

4 SYSTEM DESIGN THEORY

The design of this project looks straightforward to begin with, although getting more in-depth on this project it became a little more complicated than it let on first sight. The main required components for this project design were a UART transmission signal along with a timer, with PWM signal capability. First, the UART transmission signal will contain several bytes of data required to send to a micro-controller to read and input specific values into a timer based off of that UART transmission. The micro-controller will modify that signal and use the first 3 bytes of RGB data for inputs of a timer for use in PWM, then send the rest of the bytes of data to another interfaced micro-controller for the same process. The final process of this design is to output the signal to PWM, from the timer, to the RGB LED.

4.1 Design Requirement 1

The UART transmission signal is an important part of this system. That being said, without this signal being correct, the LED would never receive the correct values for colour and brightness. Therefore, a standard was set so that each micro-controller connected is based off of the same order. So in this case, our standardized UART signal begins with a length byte. This byte is used to determine how many bytes are being received, and after going through the micro-controller, how many bytes are being transmitted to the next board. Each byte after that (excluding the stop byte) will be in order of the colour we want to light up. So, the order of each concurring byte would be in repeating order of 1 red byte of data for LED1, 1 green byte of data for LED1 and 1 blue byte of data for LED1. This pattern repeats for each concurring 3 bytes of data (for each LED), 1 red byte of data for LED2, 1 green byte of data for LED2 and 1 blue byte of data for LED2. Lastly, a stop byte is used to tell the system to stop reading data, and execute.

4.2 Design Requirement 2

The PWM signal is also an integral part of this system. That being said, the RGB LED is based off of these PWM values, directly from the UART transmission signal. Where the most significant RGB data bytes are read and inputted into TIMER A CCRx registers. Therefore, the output mode of the timer is set to OUTMOD7 in which the G2553 is set for outputting the PWM signals to an RGB LED.

5 Getting Started/How to use the device

- Connect MSP430G2553 to Computer
- Connect P1.6 to Red lead of LED
- Connect P2.1 to Green lead of LED
- Connect P2.5 to Blue lead of LED
- Ground the LED to the MSP430G2553 GND pin.

If using two or more boards

- Connect Grounds together on all boards.
- Connect the TXD pin of the MSP430G2553 connected to PC,
  to the RXD pin of the second MSP430G2553.
- Connect each additional MSP430G2553 RXD pin to the last MSP430G2553 TXD pin.

6 Getting Started Software/Firmware

6.1 Hierarchy Chart

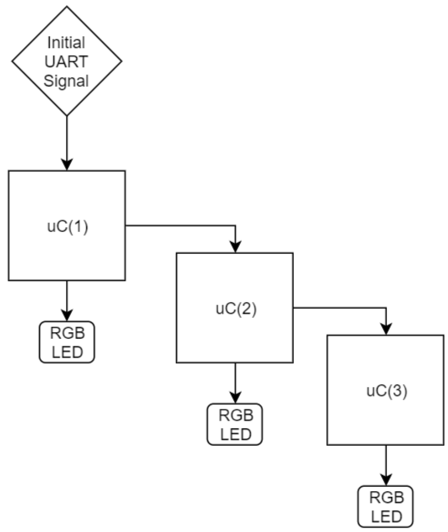

Figure 4: Hierarchy chart showing flow of communication between devices

6.2 Communicating with the Device

Communication with the first device is accomplished by using a UART software program on your PC, Realterm. Setting up Realterm for use on the MSP430G2553 is fairly straight forward. First, open up Realterm, then under the Display tab, select ”Hex[space]”, check the ”Half Duplex” Box. Next, click the Port tab and select 9600 baud rate as well as choosing the correct device port connected from the PC to the MSP430G2553 in order for Realterm to be compatible and function accordingly with it. Lastly, click the Send tab to send bytes of data over to the micro-controller based on the colour you would like to have displayed on your device. (Example: Show Red on the first LED, Green on the second LED, and Blue on the Third LED. **Data:** 0x0B 0xFF 0x00 0x00 0x00 0xFF 0x00 0x00 0x00 0xFF 0x0D). Where 0x0B is the length byte being 11 in decimal, or 0B in hexadecimal, meaning 11 bytes are being sent including the length byte. The next 3 bytes, 0xFF 0x00 0x00, meaning output red on the first LED at a peak brightness level. The 3 bytes after that is green because the order of bytes matters for red, green and blue output. The same theory applies to the next 3 bytes after green. The very last byte, 0x0D, is a stop byte, telling the system to stop reading data.

Data In	LED Colour
0x05 0xFF 0x00 0x00 0x0D	RED
0x05 0x00 0xFF 0x00 0x0D	GREEN
0x05 0x00 0x00 0xFF 0x0D	BLUE
0x05 0xFF 0xFF 0xFF 0x0D	WHITE

6.3 Device Specific Information

Some of the devices tested in this lab were faulty when it came to baud rate detection. The micro-controller would work and read data, although not at the correct baud rate, therefore giving faulty data that is not usable for our communication purposes between boards. To correct this problem, some boards may have to adjust their software and use a process called baud rate trimming.

7 Test Setup

The first step to testing the device is to make sure the code is flashed to the MSP430 board. Once that is ensured, the next step is to make sure the setup on the breadboard with the RGB LED and resistors are correct. This means connecting the Red, Green, and Blue leads of the RGB LED to resistors then to the correct pins. Make sure the wires are connected to the right pins and everything is grounded correctly. Once everything is checked, the next step is to run the implemented code.

The RGB LED could be operated and turn on at different brightness levels and different colors based on the values being transmitted through Realterm. To monitor the color outputting from the LED and the brightness of it, specific input values expressed in hexadecimal will need to be set in Realterm to achieve this task. On Realterm, specific settings compatible with the MSP430G2553 would need to be applied to have UART function accordingly with the microcontroller and RGB LED. It is extremely vital for the baud rate to be set at 9600, as well as selecting the appropriate device port connected to the micro-controller from the laptop being used, in order for every interfaced component to communicate efficiently.

7.1 Test Data

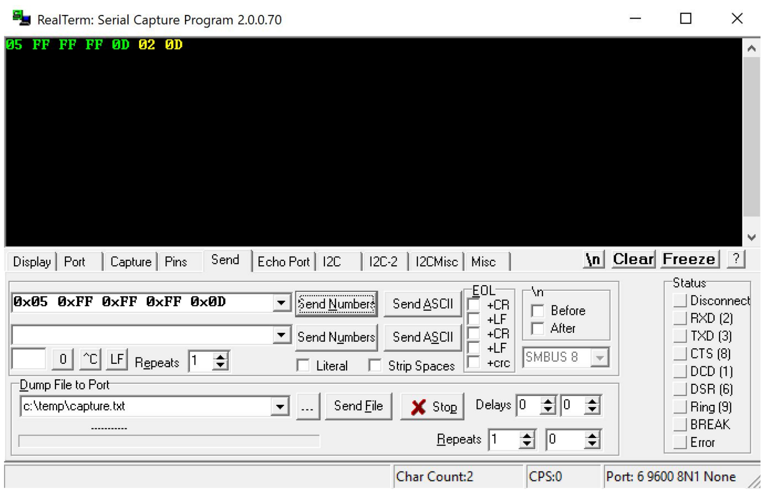

Figure 5: Realterm sending UART Signal for Figure 6

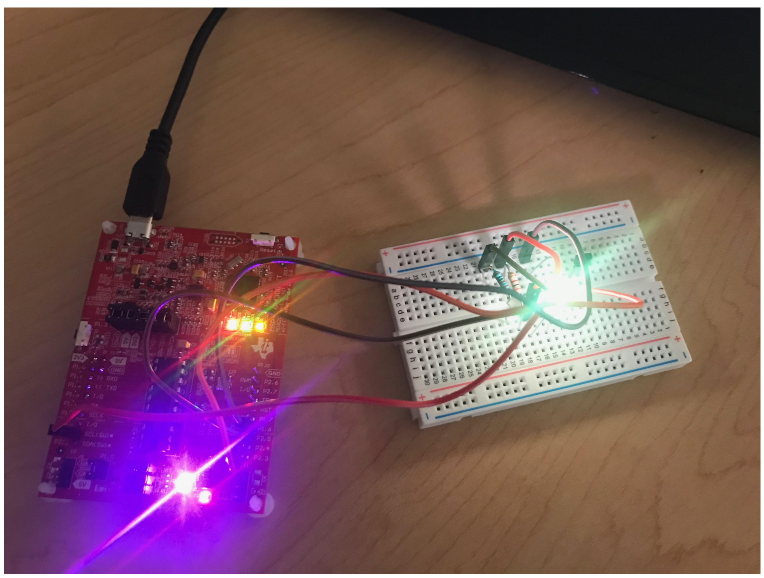

Figure 6: Board Data reading FF FF FF (White)

8 Design Files

8.1 Schematics

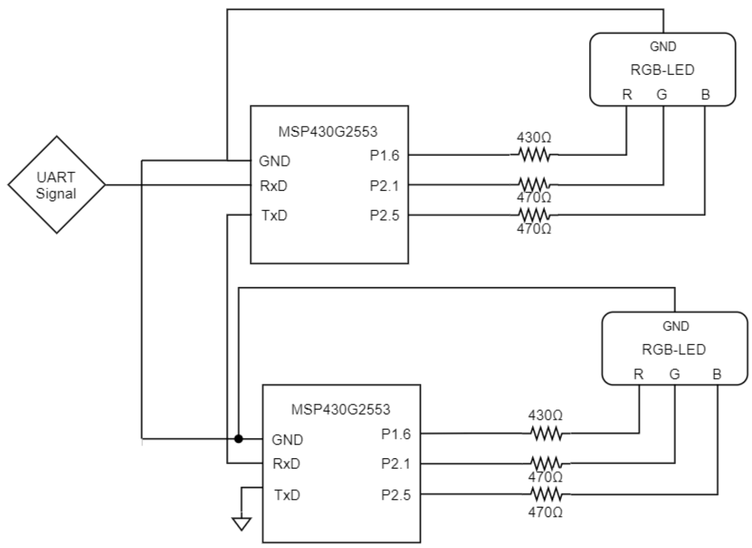

Figure 7: System Schematic

8.2 Bill of Materials

Item #	Description	Quantity
1	Microcontroller	1
2	RGB LED	1
3	430 ohm, 1 watt, 5% tolerance resistor	1
4	470 ohm, 1/2 watt, 5% tolerance resistor	2