Milestone 1

Michael Johns, Ben Zalewski
Rowan University

November 1, 2019

1 Design Overview

We were tasked with creating a device, utilizing the MSP430, to receive data from an outside source and use it to light up an RGB LED to a specified color. Then we needed to be able to only take the specific bytes of data we needed and pass on the rest through the data transfer pin on to the next board. In order to do this we needed to utilize certain on board features of the MSP430 as well as build the off-board circuit to ensure we wouldn't fry the board or the LED.

1.1 Design Features

These are the design features:

- MSP430G2553 Processor
- Adressable RGB LED
- LED generated patterns
- Used timers to set PWM
- Used UART code to communicate data with processor

1.2 Featured Applications

Applications used to complete Milestone 1:

- Code Composer Studio
- Realterm
- Git Bash

1.3 Design Resources

https://github.com/Intro-To-Embedded-Systems-RU09342/milestone-1-johnsm66/blob/master/Milestone_StrangerThings/main.c

1.4 Block Diagram

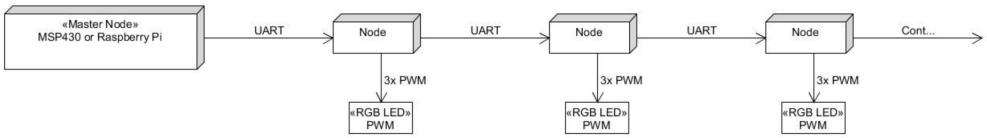

Figure 1: Simple block diagram

1.5 Board Image

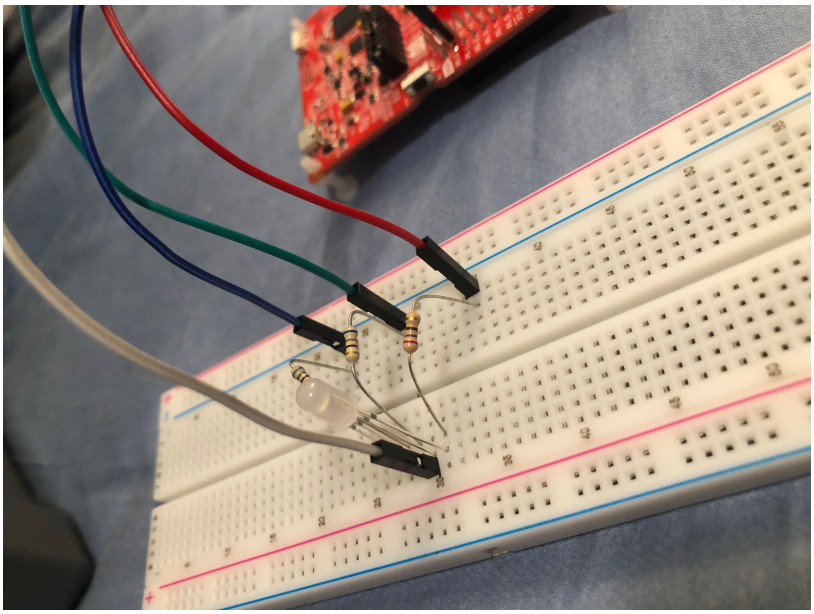

Figure 2: Breadboard connections for RGB LED

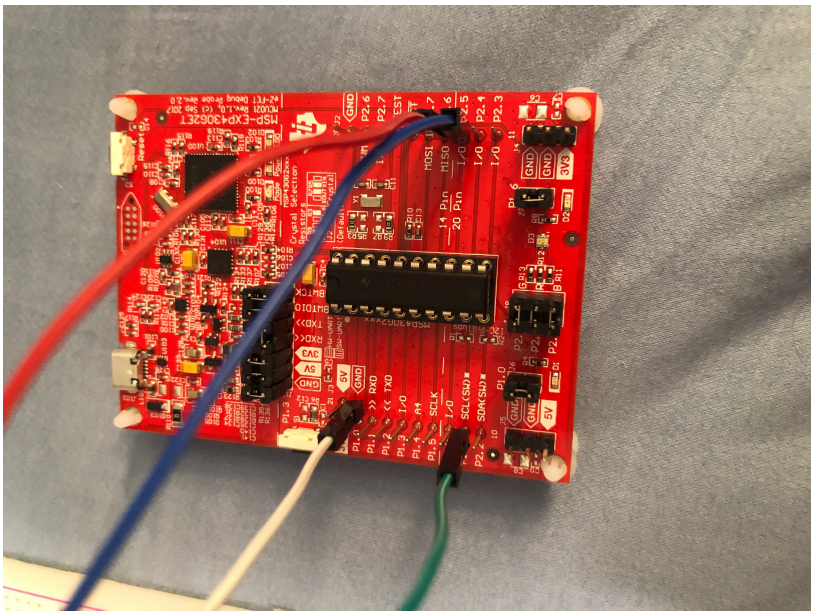

Figure 3: Pin out connections on MSP430G2553

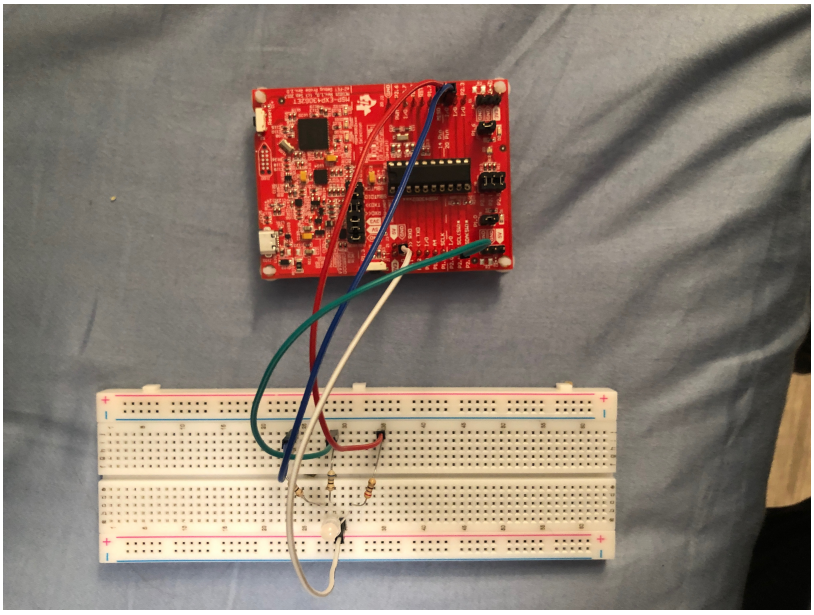

Figure 4: Processor connected to breadboard

2 Key System Specifications

Parameters	Specifications	Details
TA0CCR1	Output pin 1.6	Timer A0 port for Red LED
TA1CCR1	Output pin 2.1	Timer A1 port for Green LED
TA1CCR2	Output pin 2.5	Timer A1 port for Blue LED

3 System Description

The problem that we were tasked with was to create a device, using the MSP430G2553, that can receive data from another board take in only what it needs and then send the rest of the date to the next one. The system we built utilized aspects of the board, as well as an off board circuit built on a bread board in order to connect an external LED. In order to do this successfully, and without anything breaking or frying, we needed to figure out resistor values for each wavelength, R G and B.

3.1 Detailed Block Diagram

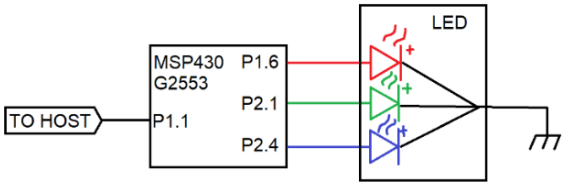

Figure 5: Detailed block diagram

3.2 Highlighted Devices

This just needs to be a bulleted list of what parts you used (not including passive components) with just a quick blurb of what it is doing in your system.

- MSP430G2553: This is the central processing unit of the system.
- Breadboard: This is where the circuit is built, it helps keep things connected and organized.
- RGB LED: This is the basis of the whole system, is this wasn't here we wouldn't know if it worked or not. Its a very necessary part of the system.
- Resistors of differing values: this keeps the LED and the MSP430 from frying and helps keep the current in a manageable place.

- Color Coded Female to Male Jumper Wires: These create the connections between the other parts of the system.
- Computer running Realterm and the Code: This is another necessary part of the system, this is how we test the code through the board to see if we are getting what we want.

3.3 Device/IC 1: MSP430G2553

In our system, the MSP430 is an integral factor, it is the central processing unit that drives the code, processes the inputted data and transfers it to the necessary location. Using the RealTerm software on our computer, we were able to send a certain amount of bytes of data to the board which, utilizing PWM and on board timers, was able to output a certain frequency on an RGB LED which resulted in a specific color being produced. Then whatever left over data that the LED did not use is sent out via the transmit pin on the MSP430. This was what enabled the class to be able to hook up as many processors as we could together and send one string of data through to all, this was done by connecting the transmit pin from the preceding device to the receive pin on the next one, thus making the most expensive LED strip we have ever seen.

3.4 Device/IC 2: RGB LED

The RGB LED was a critical part of the system as well, it was the main source of output where we can actually see whats going on. Each byte of data, except the stop byte and size byte, was mapped to either the Red, Green, or Blue wavelength inside the RGB LED. Since human beings only have 3 kinds of cones, Red, Green and Blue, these three frequencies, and the mixing of them, make up the whole visible light spectrum. Using RealTerm, we were able to output whatever amount of each of the red, green and blue we wanted to yield any specific color. On RGB LEDs there are 4 pins, Red, Green, Blue and ground. Each pin was connected to the board via jumper cables and a breadboard. Resistors were connected to regulate the voltage from the board only to the red, green and blue pins on the LEDs, because they were the only pins getting any power.

4 SYSTEM DESIGN THEORY

In the system there are two major components which are the processor code and the breadboard circuitry. Amongst the code there are important sections which code for UART, Pin Setup, and Pulse Width Modulation (PMW).

4.1 UART

UART stands for universal asynchronous receiver transmitter. The MSP430G2553 supports communication via UART which allows data to be sent to the board and

transferred amongst other boards or back to a computer. In the system, the UART code receives data in Hex and the beginning bytes have a specific meaning. The first byte represents the total number of bytes in the data sent. The next three bytes represent the Red, Green, and Blue LED respectively. The bytes that follow, are not needed and sent on to the following board and the process is repeated. The last byte, ”0x0D” represents the end of the message. Fig. 6 and Fig 7. below can be referenced to help explain UART.

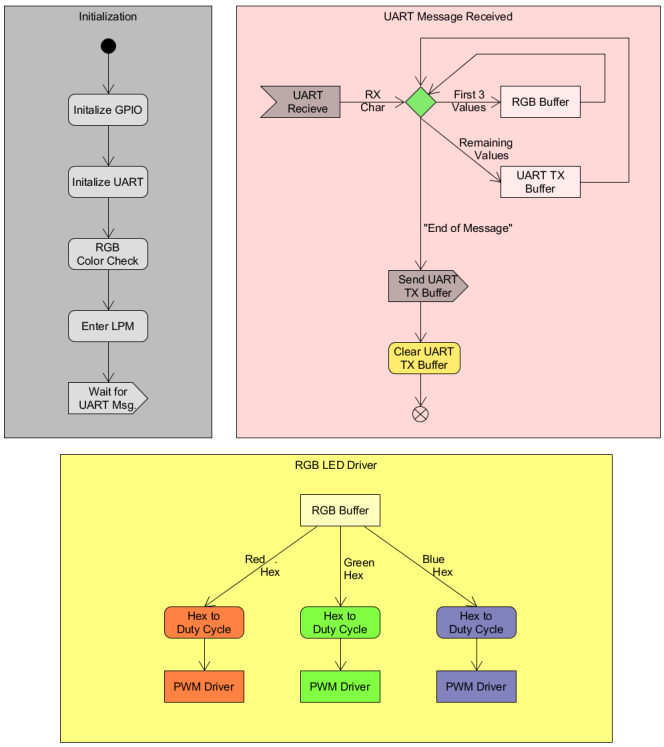

Figure 6: UART paths

Byte Number	Contents	Example
Byte 0	Number of bytes (N) including this byte	0x50 (80 bytes)
Bytes 1-(N-2)	RGB colors for each node	0xFF (red) 0x00 (green) 0x88 (blue) ...
Byte N-1	End of Message Character	0x0D (carriage return)

Figure 7: Byte correlation

4.2 Pin Setup

The pins were connected as follows:

- Pin 1.6 on the MSP430 was connected to the Red pin on the LED.
- Pin 2.1 on the MSP430 was connected to the Green pin on the LED.
- Pin 2.5 on the MSP430 was connected to the Blue pin on the LED.
- Ground on the MSP430 was connected to the Ground pin on the LED.

4.3 PMW

PWM stands for Pulse Width Modulation, this allows for each color inside the RGB to be controlled down to the frequency it is trying to output. It modifies the duty cycle of each RGB color, respectively, to a specific percentage, the lower percentage yields a lower brightness, or less of that color if you are mixing them to make one other than red green or blue. It is implemented in our code by utilizing the timer modules and capture and compare registers. The timer is set to up mode which means it will count up to the capture and compare registers and then reset and do it again repeatedly. After this, the register is checked via an if statement and if the value doesn't match up with what was sent through the UART it is changed accordingly.

4.4 Circuitry

The circuitry that was built on the bread board is just a simple resistor and LED circuit to regulate the amount of voltage going to the LED, this makes sure we won't fry the LED and to protect the board as well. The resistors were only connected to the red, green and blue pins because these were getting power, a grounding pin doesnt need a resistor in this case. The resistor values were the same for green and blue pins because they have a similar duty cycle, reds had to be a little higher because it has a higher duty cycle.

5 Getting Started/How to use the device

In order to send data with the device, the main lead processor must be connected to a computer. With all the board connected properly, the data is sent to the lead board and the data should be passed throughout each board if the code and setup is correct.

6 Getting Started Software/Firmware

6.1 Hierarchy Chart

The lead board is connected to a computer and recieves data via Realterm. After the first board uses the bytes needed from the data it sends the left over bytes through the pin 1.2 which is Txd. The next processor would then receive that data through pin 1.1 which is Rxd. That processor would then use the bytes needed and send the rest of the data the same way. This connection between boards can be repeated to send data between numerous boards.

6.2 Communicating with the Device

Pins 1.1 and 1.2 are the two pins used to communicate between processors. Pin 1.1 receives data and pin 1.2 transfers the left over data. The boards in the system would be connected so pin 1.2 from board is connected to pin 1.1 of the next board. The first board in the system does not have a connection at pin 1.2 and the last board in the system does have a connection at pin 1.2.

7 Test Setup

To test device, the processor must first be connected to the computer via USB. The code can then be debugged and flashed onto the board. Once the board is flashed it can simply be connected to power and the code will run. The next step in the setup is to make sure the RGB LED is properly set on the breadboard with the correct resistor values and connections to power and ground. Next, the pin outs on the processor can be wired to the correct locations on the breadboard. Once the wiring is complete, the processor can be connected to the computer. The software, Realterm, is then used to send bytes to the processor. Realtearm has setting that must be changed in order to send the bytes to the board. Under ”Display” tab, ”Hex[space]” and ”Half Duplex” must be selected. Under ”port” tab, the baud must be set to ”9600” and the port must be selected to the correct port the processor is. This can be checked using the device manager on the computer. Then, under the ”Send” tab, hex values can be sent to the board to test if the processor is functioning correctly.

7.1 Test Data

To test the RGB LED, the hex value ”0x05 0xFF 0x00 0x00 0x0D” can be sent using Realterm. If the system is functioning correctly, the RED led should light up and the ”0x0D” should be seen sent back to Realterm. After ”0x05” which represents the number of bytes, the following three bytes represent the Red, Green, Blue LED respectively. The hex value can changed to test each RGB and different combinations as well. The bytes after the three RGB bytes, are not used and sent on to the next board so those bytes should be seen sent back to Realterm.

8 Design Files

8.1 Bill of Materials

-RGB LED
-MSP430G2553
-Three resistors