Milestone 1

Dylan Dancel and David Russo
Rowan University

October 30, 2018

1 Design Overview

The design constructed is to control a common anode RGB LED with the MSP430F5529 processor from Texas Instruments. This is accomplished by using a technique called software PWM (Pulse-Width Modulation). The LED is then supposed to take packets of data which control the intensity of the color of the LED. These packets are sent through the RealTerm executable through the GPIO pins which are on the development board of the MSP430F5529.

1.1 Design Features

- Ability to communicate to Multiple Nodes
- USB Controllable
- Addressable
- Can Produce over 16 million unique colors
- Flashable

1.2 Featured Applications

- Holiday Lights
- Children Toys
- Decoration

1.3 Design Resources

Access to GitHub as well as TI resources accessed.

- GitHub
- MSP430 Family User Guide
- MSP430F5529 Datasheet

1.4 Block Diagram

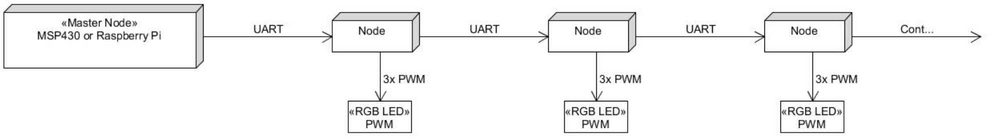


Figure 1: Node Flowing for LED Control

1.5 Board Image

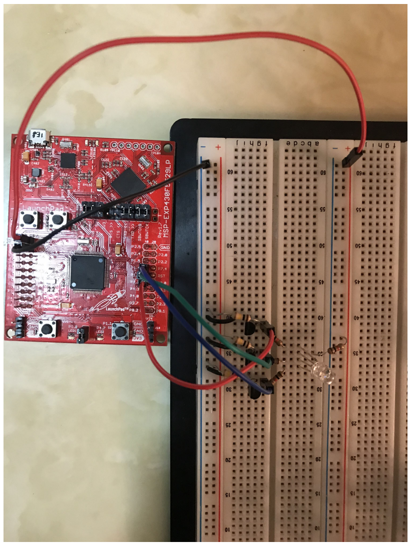


Figure 2: MSP430F5529 with Low Side switches

2 Key System Specifications

PARAMETER	SPECIFICATIONS	DETAILS
Baud Rate	9600	Baud Rate used to communicate with the MSP430F5529
Timer A0	16 bit	16 bit timer that has 5 capture compare registers
SMCLK	1MHz	1MHz clock that is active in low power mode 0 and 1
LED	RGB	Produce colors from range of 0-255 for each node

3 System Description

The desired outcome is to use and control the duty cycle of the LEDS to get different colors by sending packets through the UART interface.

3.1 Detailed Block Diagram

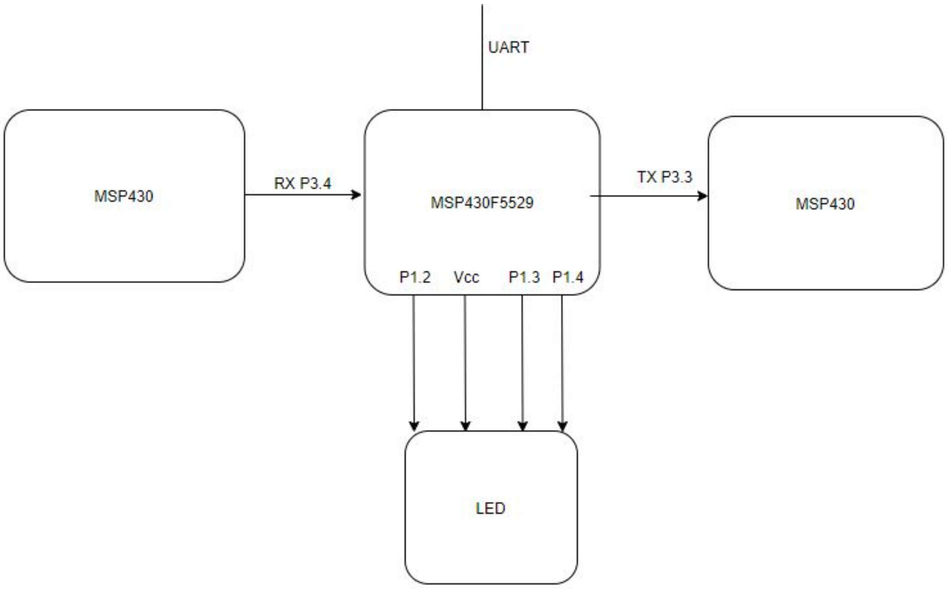


Figure 3: Block diagram of the behavior of the processor to control the LEDS and transmit and receive data

I

3.2 Highlighted Devices

- MSP430F5529

3.3 MSP430F5529

The MSP430F5529 is a Texas Instrument processor with a TimerA0 that contains 5 capture compare registers. Out of the 5 registers that could be used only four of the registers were needed. These capture compare registers were used to hold data coming in through the UART interface as packets were sent into the processor.

4 SYSTEM DESIGN THEORY

4.1 Design Requirement 1

The first design requirement is to control the RGB LEDs through software PWM from the pins of the MSP430F5529. This allows for the controlling for how the LED is lit up based on the duty cycle given to the specific color. The PWM can range from 0 to 255 where 255 is the color illuminating at its max intensity. TimerA0 CCR0 is set to 255 so that whenever the clock counts up to 255 it gets reset. The way these colors are controlled is by the value that is stored into the capture compare register of CCR1,CCR2, and CCR3.

4.2 Design Requirement 2

The second design requirement is that the RGB LEDs are able to be controlled through UART. This means that sending in packets of data also allows for the LEDs to change color as data is received. The code is set up to take in data from UART by first detecting how many packets of data it is going to receive. The following data after this initial piece will control the values stored into the CCR1, CCR2, and CCR3 registers. If there is no more data to process, then the MSP430F5529 outputs a value of 0x00 through UART. This is to tell the computer that it is the last byte of data sent into the processor. Otherwise, the microprocessor will continue to pass along data to the next node in the chain.

5 Getting Started/How to use the device

To power the device, a USB-to-micro-USB must connect the MSP430F5529 development board to a computer with the ”Realterm” application. All other connections use MSP430F5529 pins. Because the RGB LED is a common-anode, the anode should connect to V_{cc}. Figure 5 shows the detailed connections for the off-board circuit needed to operate, where V_{cc} is 5V.

6 Getting Started Software/Firmware

The user must be aware of the information they send into the MSP430F5529 through RealTerm. The first byte represents the number of bytes in the package. The next three bytes represent the color of the RGB LED, and all other bytes are transferred through to the next node. If a user inputs a package length that is not equal to the actual package length, the MSP430F5529 may not function correctly.

6.1 Device Specific Information

When sending data from the pin of the MSP430F5529 to the pin of another board, the direction of P3.3 should be set as an output. This allows data to be transmitted through this pin. When receiving data from the pin of another board to the pin of the MSP430F5529, the jumper on the RX pin must be removed, and pin 3.4 should be set as an input.

7 Test Setup

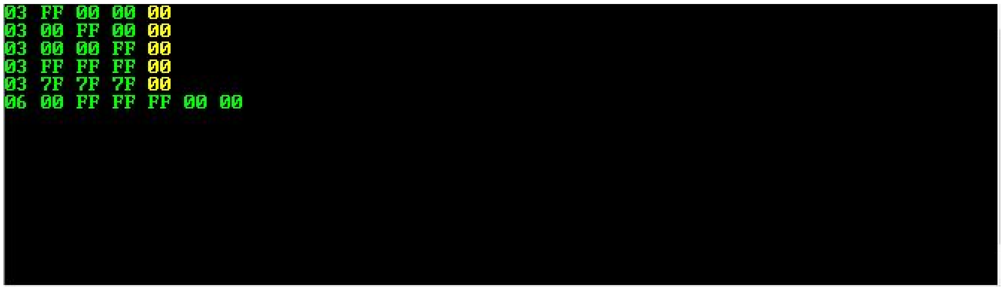
The test setup is started by executing RealTerm and configuring the display settings to the ”Hex[space]” and ”Half Duplex” setting. Configuring RealTerm in this way allows the user to see the transmitted and received bytes on the screen. Next the COM port must be identified and adjusted accordingly in the port settings. The COM port is unique to each device and computer. The Windows Device Manager shows the particular COM port of the MSP430 on the user’s computer. Navigate to ports (COM LPT), then to MSP Application UART1 (COMn). Finally, the Baud Rate must be set to 9600.

7.1 Test Data

The data that is sent in is represented by the green hex values shown in Figure 4. The first packet sent is 0x03 0xFF 0x00 0x00. this illuminates the LED red and returns 0x00. The process is then repeated for green, blue, white, half-white, and a 2 packet length. The two packet length makes the LED teal and transmits the remaining data to the next node. The next node should interpret this as data and light up its LED red. Since the first LED can only take three of the six bytes to determines its color, the remaining bytes are sent over to the next LED producing red. Since this data has been sent via transmit data the return signal does not come back to the computer as the device is not connected in such a way to return it back to RealTerm.

8 Design Files

8.1 Schematics



```
03 FF 00 00 00
03 00 FF 00 00
03 00 00 FF 00
03 FF FF FF 00
03 7F 7F 7F 00
06 00 FF FF FF 00 00
```

Figure 4: Data Sent into the MSP430F5529

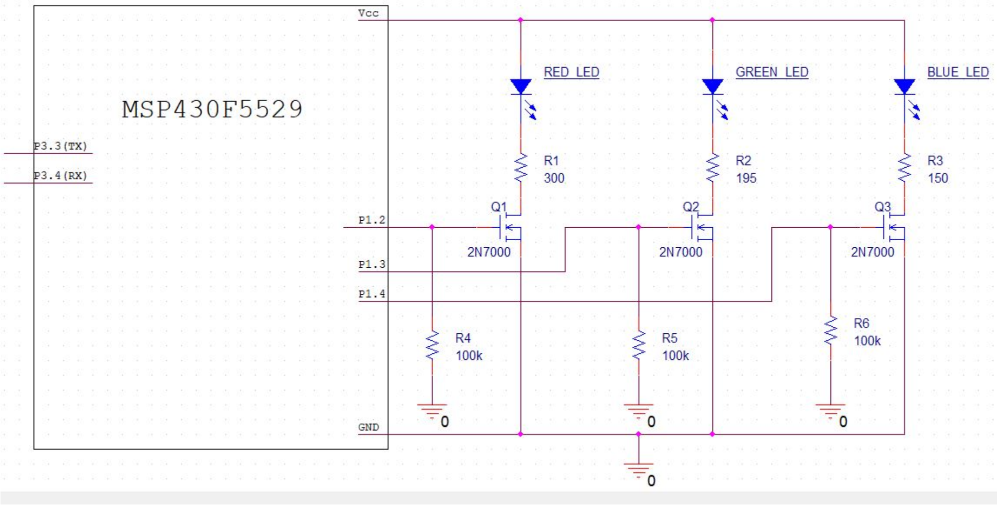


Figure 5: MSP430F5529 with low side switches