– ECE 09.342: Milestone 1 –

Application Note Template

Dylan Dancel and David Russo
Rowan University

October 22, 2018

1 Design Overview

The design constructed is to control a common anode RGB LED with the MSP430F5529 processor from Texas Instruments. This accomplished by using a technique of 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 a Range of 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

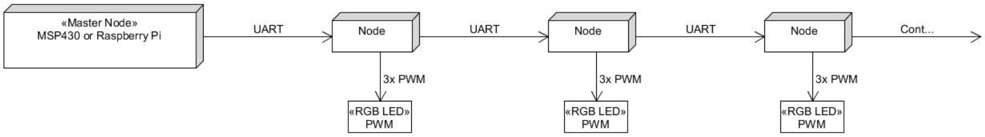


Figure 1: Node Flowing for LED Control

1.5 Board Image

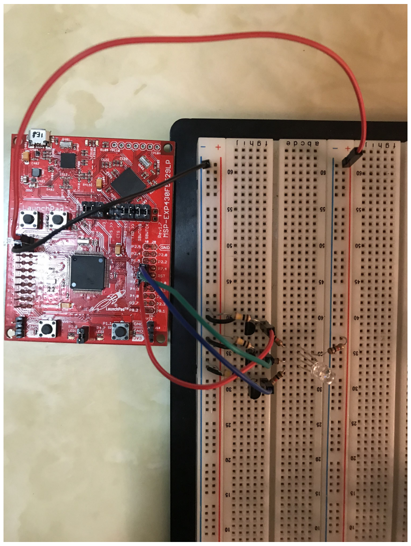


Figure 2: MSP430F5529 with Low Side switches

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

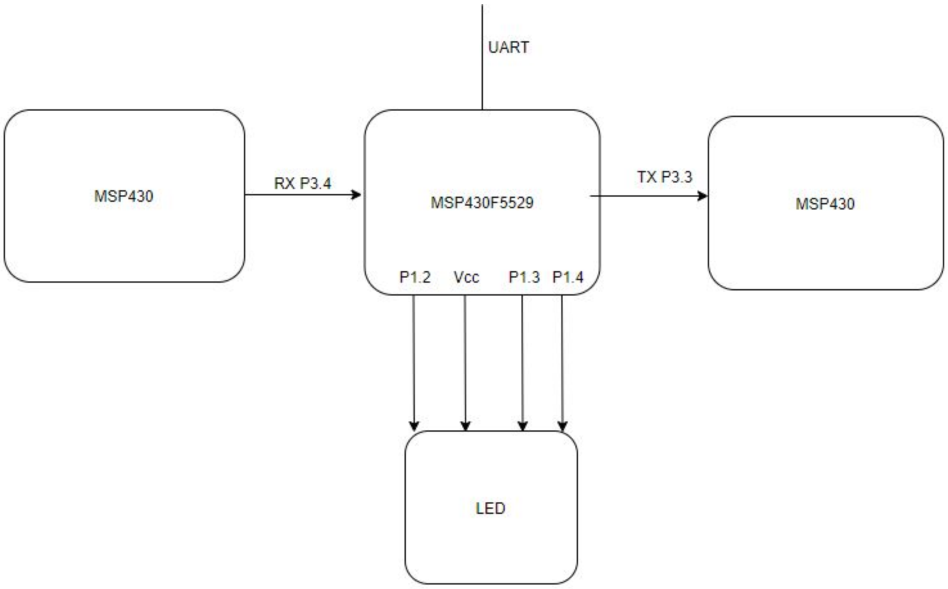


Figure 3: Block diagram of the behavior of the processor to control the LEDS and transmit and receive data

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 LED's 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 be are able to 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 setup to take in data from UART by first detecting how many packets of data is it going to receive. The following data after this initial piece will control the values stored into the CCR1, CCR2, and CCR3 registers. The data taken in exited by returning a value of 0x00. This is to tell the computer that it is the last byte of data sent into the processor.

5 Getting Started/How to use the device

The connections that must be made are mostly from pins of the MSP430F5529 and USB to micro-USB cable to power the development board itself. Stated earlier, the RGB LED is a common anode therefore it should be connected to V_{cc} from there. Figure 5 shows the detailed connections for the off-board circuit needed to operate. This is where V_cc is approximately 3.3V.

6 Getting Started Software/Firmware

The packets sent from RealTerm must be noted when sending them into the processor. One LED has only a maximum of 3 bytes of data it can see, this means any package length higher than this will transmit any data that wasn't used to the next node in the chain.

6.1 Device Specific Information

This is to be noted that when sending data from pin to pin to another board that the direction of P3.3 should be set to as an output. This allows for data to be transmit through this pin otherwise no data can be sent from it.

7 Test Setup

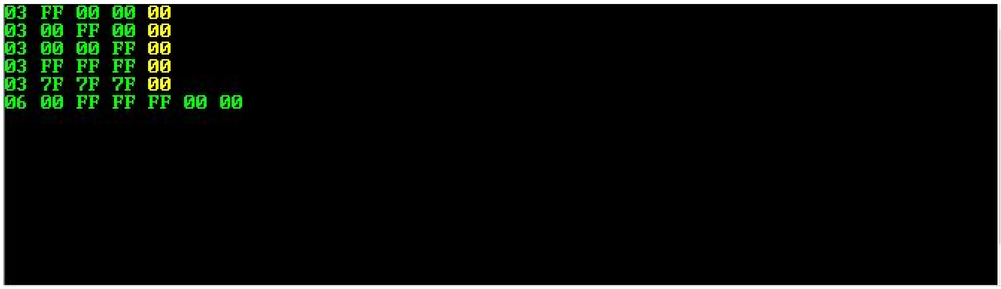
The test setup is started by executing RealTerm and adjusting the display settings which to the Hex[space] and Half Duplex setting. Configuring the RealTerm in this way lets for the analysis of input data and output data. Next the COM port must be identified and adjusted accordingly in the port settings while also changing the Baud Rate to 9600. The setup for the tests are now complete and time to put in data for the LED.

7.1 Test Data

The data that is sent in is represented by the green hex values represented in the test data in Figure 4. The first packet sent is the 0x03 0xFF 0x00 0x00 this produces a red color for the LED and a return byte of 0x00. The process is then repeated for green, blue, white, half-white, and a 2 packet length. Specifically, the two packet length it is to make the first LED teal and the next LED red. Since the first LED can only take 3 of the bytes as to determines its color from the 6 bytes the remaining bytes are sent over to the next LED producing red as it receives 0xFF. 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 real term.

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

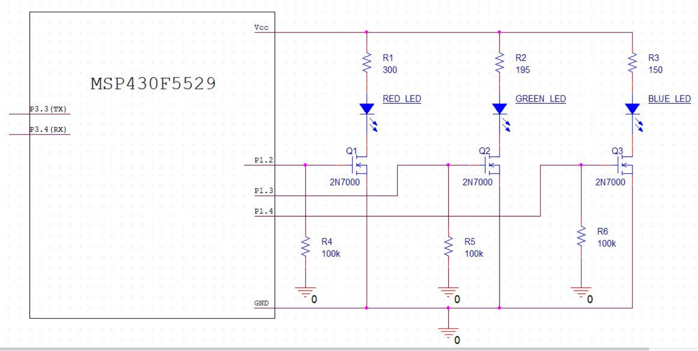


Figure 5: MSP430F5529 with low side switches