ECE 09342 2: Milestone 1: Communicating with Will Byers

Application Note

Matthew Rodriguez, Austin Huang, and Seamus Plunkett
Rowan University

October 16, 2016

1 Abstract

The Stranger Things christmas light wall assignment allows for further exploration of the capabilities of the MSP430. The procedure involves learning background information on UART to create the colors on the RGB LED in order to complete the assignment. The code will be done in CodeComposer and run on the MSP439FR5994 to test the results and implement the Stranger Lights functionality. The code will be run and tested so that it can generate specific light patterns based on the information it receives to communicate the message.

2 Introduction

The goal of this milestone was to create an addressable RGB LED using the MSP430. Many leds would be strung together in order to replicate the Christmas Light Wall from the Netflix Original “Stranger Things.” In order to address and change the RGB LED, UART or Universal Asynchronous Receiver Transmitter was used. UART has many applications in complex peripheral devices such as GPS modules and LCD displays and should work well in this milestone. This milestone will apply and test all the knowledge gained from the past 4 labs.

3 Background

A UART device works by taking bytes of data, transmitting the bytes bit by bit, and a second UART device receiving these bits and building them back into bytes. Through the use of a shift registers, each UART is able to to convert between the byte and bit forms of the data. This is useful because parallel transmission of data through multiple wires is more expensive than bitwise transmission of data through a single wire.

The idle state of the UART device is a logical high. Each transmitted character is represented as a logical low start bit, the data bits, maybe a parity bit, and one or more stop bits. The parity bit is a way of checking that the number of bits in the transmission is always add by adding a bit if the length is odd or leaving it alone if the length is even,

UART devices are typically controlled by a clock signal running at 8 times the bit rate. If a signal transmitted to the bit rate lasts at least half as long as the bitrate, or equal to or more than 4 clock cycles, it is considered valid and transmitted. If not, it is considered a spurious pulse and ignored. After the standard length of the character has passed, (This can be 5 to 8 bits depending on the format) the bits received are made available to the receiving system in the form of a shift register. The UART will set a flag signalling that new data is available for transfer, and may also generate an interrupt to the processor to receive this data.

3.1 Figures

```
else if ((byteCount > 0) && (byteCount < 4))
{
    switch (byteCount)
    {
    case 1:
        redNum = UCA0RXBUF;
        break;
    case 2:
        greenNum = UCA0RXBUF;
        break;
    case 3:
        blueNum = UCA0RXBUF;
        break;

    }
    byteCount++;
}
```

Figure 1: Least significant bytes being used to configure the duty cycle of RGB LED

```
else if ((byteCount > 3) && (byteCount < numOfBytes))
{
    Message[byteCount + 1] = UCA0RXBUF;
    byteCount++;
}
else if (byteCount >= numOfBytes)
{
    byteCount = 0;
    UCA1IE &= ~UCRXIE;                    // Disable USCI_A1 RX interrupt
    UCA1IE |= UCTXIE;                      // Enable USCI_A1 TX interrupt
    UCA1TXBUF = Message[i];
    P1OUT &= ~flag
    ;
}
```

Figure 2: Message(string of characters) being outputted to through the TX Line

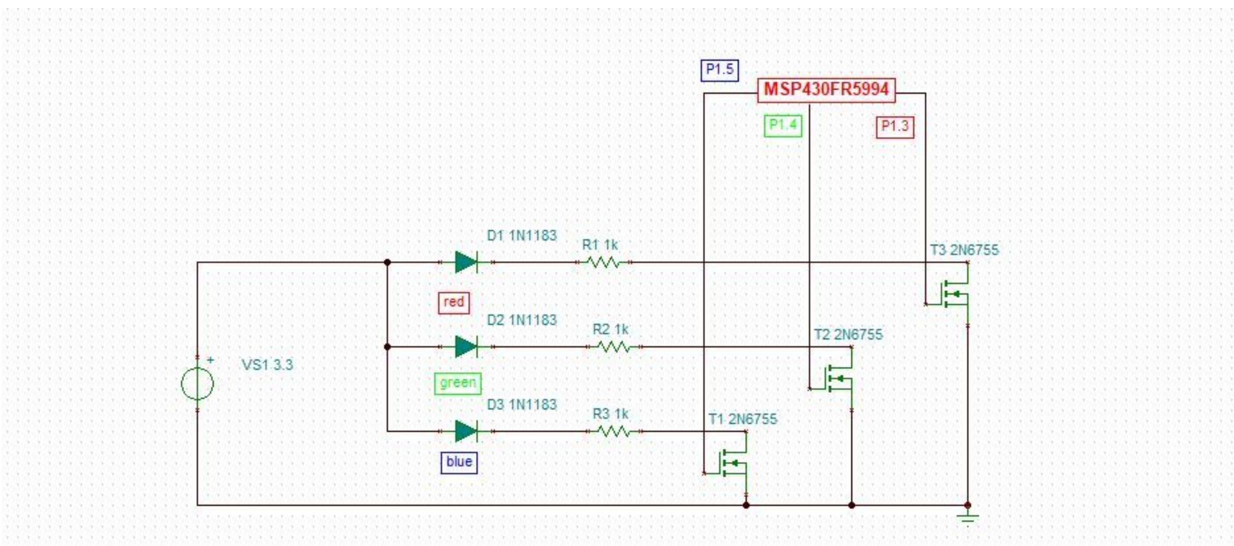

Figure 3: Tina-Ti schematic of RGB LED

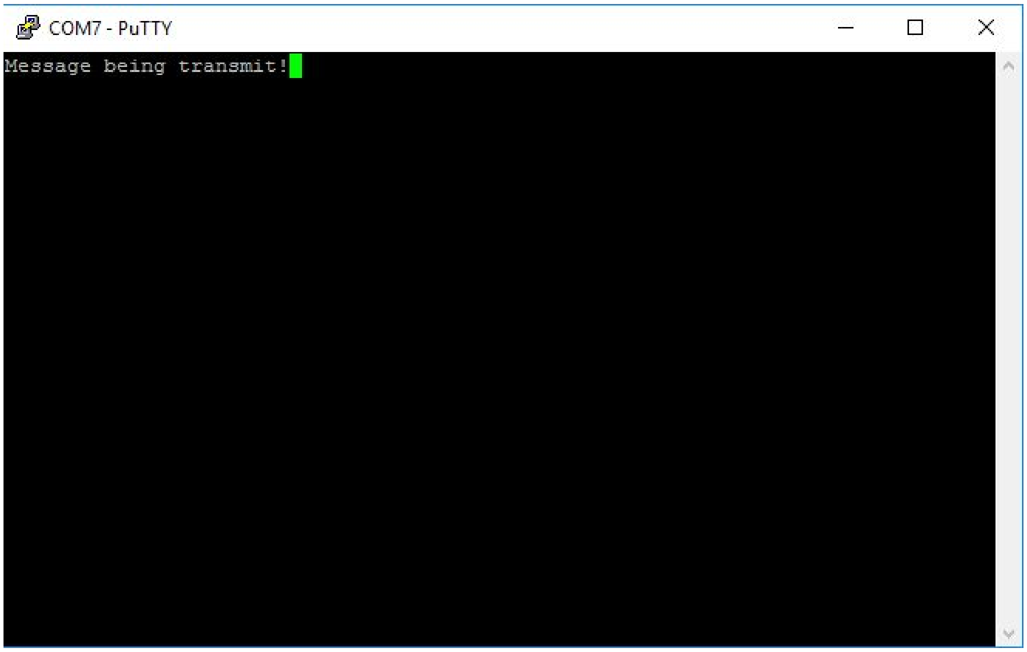

Figure 4: Putty message being Transmitted.

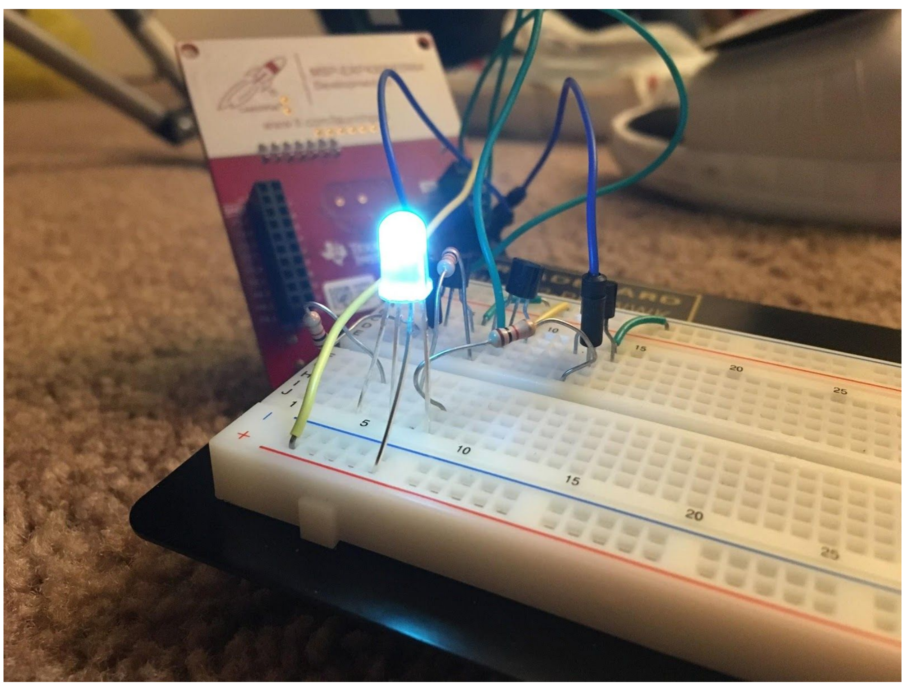

Figure 5. Picture of RGB LED when when message from PuTTY is received

4 Design Considerations

Of all the boards the MSP430F5994 was chosen to complete this lab. The main reason the FR5994 was chosen over other boards was that it provided consistent reliable UART communication. The shortcomings of the other boards would have been a wider variety of factors to consider that were not necessary for the given application. For example, the FR2311 has only Timer B which is a nuisance that limits its PWM capabilities and the FR6989 does not have reliable UART capabilities. The FR5994 has the tools to accomplish the goal of the milestone.

3.3V is supplied by the MSP430F5994 board and powers the three LEDs. Some design considerations were determining small enough resistor values to properly power the LEDs that were also large enough to keep them from burning out.

5 Evaluation and Results

The code takes a string of characters(ASCI) and converts the three least significant bytes to binary and uses the values of the three least significant bytes to configure the duty cycles of the three LEDs inside the RBG LED. After configuring duty cycle, the rest of the string of characters is saved onto a character array called “Message” with a fixed size of 80. After, it copies the characters onto that array it sends each value in that array through the UART TX line allowing for a message to be sent to the next node.

The code uses case statements to configure the duty cycles of the red, green, or blue LEDs in the RGB LED. This can be seen in Figure 1. The code demonstrates the three least significant bytes being separated and put into variables called redNum, greenNum, and blueNum which will determine the duty cycle of each respective LED.

The code in Figure 2 uses if statements to check the current byte being handled.. If the byteCount variable (represents the current byte of the string of characters) of the code is less than the numOfBytes variable (represents the length of string of characters), then it will add a character to the Message array of characters for each iteration of the loop while the condition is true. Once byteCount equals the numOfBytes, it then resets the byteCount and proceeds to output the Message array through the TX line.

Furthermore, Putty was used to test the code. Putty was used to transmit the string of characters or message to the MSP430 which would then handle the message to set its own RGB values and then send the next message if necessary. Furthermore, Putty was also used to test if the board was able to successfully transmit the remainder of the message out. This would be done by checking if Putty received a message back. Figure 4 shows an example of a message being transmitted by the MSP430FR5994 and received by putty. The message is printed on putty.

6 Conclusion

The behavior of the program was driven using Putty serving two functions as a transmitter and receiver as seen in Figure 4. The board was able to successfully handle a message by using the least significant three bytes to determine its RGB values. Figure 5 demonstrates that the board successfully configured the RGB values as seen by the LED glowing an aqua color. The RGB values changed accordingly to the different messages inputted by the user. The transmission of the rest of the message was tested with Putty.

One of the main design challenges was creating a program that allows the board to be placed at any point in the light string. Each board needed to use the three least significant bytes of the message to configure its LEDs without transmitting those three in addition to the message. This was solved using coding logic to add the correct characters to an array of characters in order to spell out the message to be passed on. After each node, the message moves through and becomes shorter until it reaches the last node. This process allows each node to run the same code only using the three least significant bits to alter the node's RGB state.

The design choice to use software PWM as opposed to hardware PWM was made for a number of reasons. Most importantly, hardware PWM could cause some issues when using UART because common pins would be used. In addition, the software PWM written for Lab 4 could easily be inserted in our program.

The end result of combining 26 teams' milestones should be a fully functioning addressable LED string using UART and PWM. The success of the milestone can be judged once the demonstration has occurred.