

RoboLights

Project: Controllable LED lights for FRC robot

Summary:

This project will use a Trinket microcontroller to accept a 0-5v analog signal from the RoboRIO to select between LED output modes on the 5050 addressable LED strip. Since we are using a 3 amp converter to drive the LED strip, no more than 60 LEDs should be on the strip. If you want more LED's on the strip, use a DC-DC converter with a higher amp rating. Fifteen different colors/animation modes are available plus off mode.

Parts List:

LED strip, 5050 RGB addressable, 5v, 60 LED/meter, 2 meters



Trinket microcontroller, 5v, Arduino compatible, by Adafruit



12v to 5v DC-DC converter (UBEC), 3 amp rating or higher, typically used on drones and hobby aircraft



Capacitor, 1000 uF

Resistor, 220 - 470 ohms, 1/4 watt

Electrical tape, hot glue, shrink wrap

16 AWG wire, 2 conductor

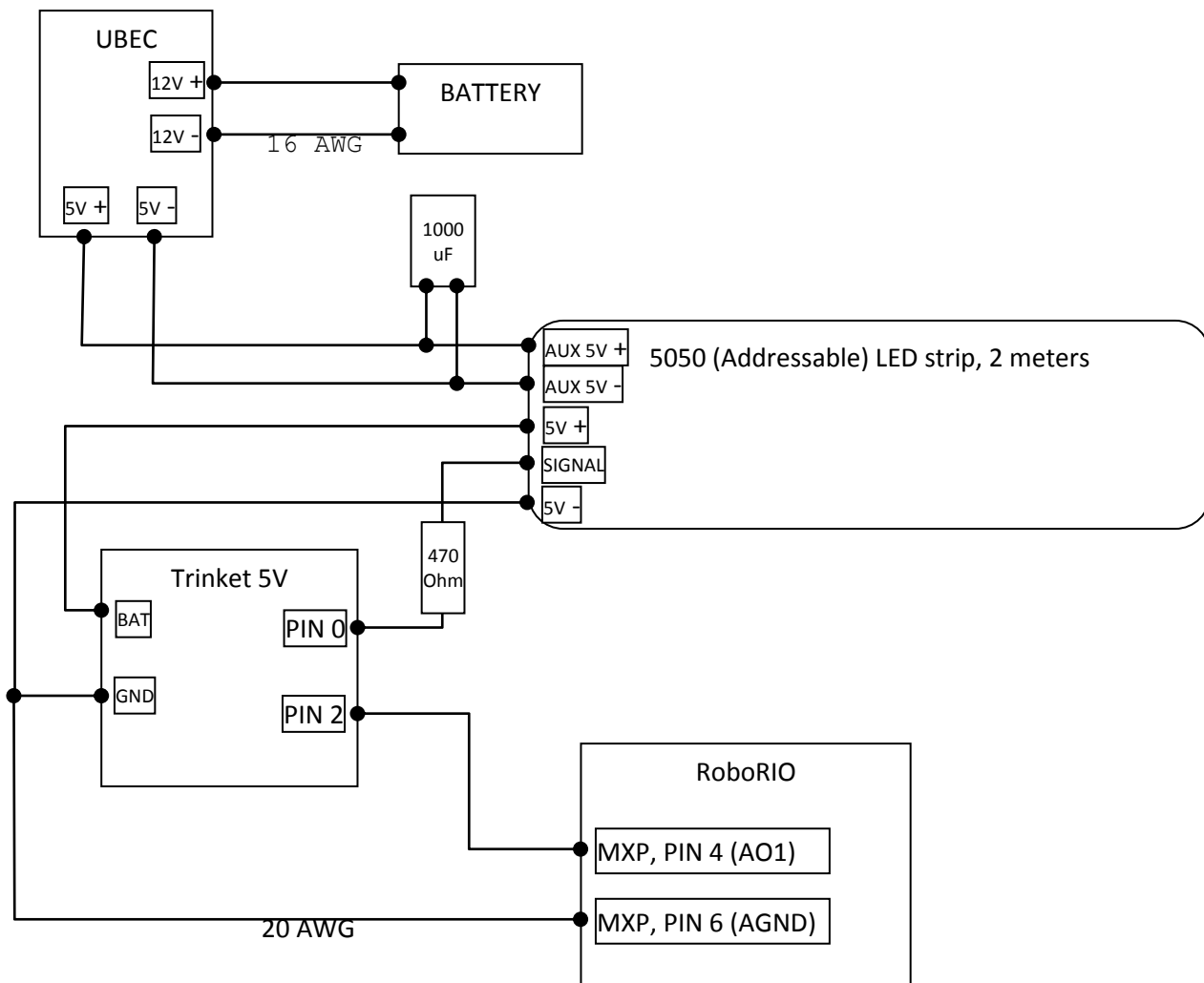
2 red/black anderson powerpole connectors

20 AWG wire, 2 conductor

2 pin Dupont connector

Assembly:

Cut your LED strip to 60 LED's or less. If you use less, change the `PIXEL_COUNT` variable in the program. Make sure to observe the direction arrows on the strip, solder your connections to the end the arrow is pointing away from. Solder all the connections per the diagram below. Use hot glue to insulate all connections and the exposed pins on the microcontroller. **Take great care to ensure no exposed connections will touch.** Use the powerpole connector for the 16 AWG wire going to the battery and the Dupont connector for the 20 AWG wire going to the RoboRio. After the hot glue cools down, wrap with electrical tape for more insulation and strength. Make sure everything is compact enough to be covered by the shrink wrap to make a nice neat package. Leave the mini-USB port exposed on the Trinket microcontroller for future program changes.



Trinket Microcontroller

Use the following instructions to program the Trinket:

<https://learn.adafruit.com/introducing-trinket/introduction>

<https://learn.adafruit.com/introducing-trinket/setting-up-with-arduino-ide>

```
// program modified from NeoPixel library buttoncycler example program
// Chester Marshall June 2017
//Uses analog input coming from roboRIO to set 16 different lighting modes of LED strip

#include <Adafruit_NeoPixel.h>

#define PIXEL_PIN 0 //trinket pin #0 = PWM out to LED strip
#define PIXEL_COUNT 60

// Parameter 1 = number of pixels in strip
// Parameter 2 = pin number (most are valid)
// Parameter 3 = pixel type flags, add together as needed:
// NEO_RGB Pixels are wired for RGB bitstream
// NEO_GRB Pixels are wired for GRB bitstream, correct for neopixel stick
// NEO_KHZ400 400 KHz bitstream (e.g. FLORA pixels)
// NEO_KHZ800 800 KHz bitstream (e.g. High Density LED strip), correct for neopixel stick
Adafruit_NeoPixel strip = Adafruit_NeoPixel(PIXEL_COUNT, PIXEL_PIN, NEO_GRB + NEO_KHZ800);

int rawRead = 0;
int curMode = 0;
int oldMode = 0;
int colorRed = 0;
int colorGreen = 0;
int colorBlue = 0;
long nowTime = 0;
long lastModeTime = 0;
long lastShowTime = 0;
int chasePos = 0;
int chaseRed = 0;
int chaseGreen = 0;
int chaseBlue = 0;
int pulseVal = 0;
bool pulseUp = false;

void setup() {
  pinMode(PIXEL_PIN, OUTPUT);
  pinMode(1, OUTPUT); //red LED on trinket pin #1
  strip.begin();
  strip.show(); // Initialize all pixels to 'off'
}
```

```

void loop()
{
    nowTime = millis();

    if(nowTime - lastModeTime > 1000)
    {
        lastModeTime = nowTime;
        rawRead = analogRead(1);
        curMode = getMode(rawRead);
        if (curMode != oldMode)
        {
            oldMode = curMode;
            setColor(curMode);
            digitalWrite(1, HIGH); //show we got a mode change
        }
        else digitalWrite(1, LOW);
    }

    if(nowTime - lastShowTime > 25)
    {
        lastShowTime = nowTime;
        runShow(curMode);
    }
}

int getMode(int nowRead)
{
    if(nowRead <= 63) return 0;           //0.1568
    else if(nowRead >= 64 && nowRead <= 127) return 1; //0.4704
    else if(nowRead >= 128 && nowRead <= 191) return 2; //0.784
    else if(nowRead >= 192 && nowRead <= 255) return 3; //1.0976
    else if(nowRead >= 256 && nowRead <= 319) return 4; //1.4112
    else if(nowRead >= 320 && nowRead <= 383) return 5; //1.7248
    else if(nowRead >= 384 && nowRead <= 447) return 6; //2.0384
    else if(nowRead >= 448 && nowRead <= 511) return 7; //2.352
    else if(nowRead >= 512 && nowRead <= 575) return 8; //2.6656
    else if(nowRead >= 576 && nowRead <= 639) return 9; //2.9792
    else if(nowRead >= 640 && nowRead <= 703) return 10; //3.2928
    else if(nowRead >= 704 && nowRead <= 767) return 11; //3.6064
    else if(nowRead >= 768 && nowRead <= 831) return 12; //3.92
    else if(nowRead >= 832 && nowRead <= 895) return 13; //4.2336
    else if(nowRead >= 896 && nowRead <= 959) return 14; //4.5472
    else if(nowRead >= 960 && nowRead <= 1023) return 15; //4.8608
    else return 0;
}

void setColor(int i)

```

```
{
switch(i)
{
case 0: colorRed = 0; //Off
        colorGreen = 0;
        colorBlue = 0;
        break;
case 1: colorRed = 255; //Red static
        colorGreen = 0;
        colorBlue = 0;
        break;
case 2: colorRed = 0; //Green static
        colorGreen = 255;
        colorBlue = 0;
        break;
case 3: colorRed = 0; //Blue static
        colorGreen = 0;
        colorBlue = 255;
        break;
case 4: colorRed = 255; //Orange static
        colorGreen = 140;
        colorBlue = 0;
        break;
case 5: colorRed = 0; //Black with orange chaser
        colorGreen = 0;
        colorBlue = 0;
        chaseRed = 255;
        chaseGreen = 140;
        chaseBlue = 0;
        chasePos = 0;
        break;
case 6: colorRed = 0; //Blue with orange chaser
        colorGreen = 0;
        colorBlue = 0;
        chaseRed = 255;
        chaseGreen = 140;
        chaseBlue = 0;
        chasePos = 0;
        break;
case 7: colorRed = 0; //orange wipe
        colorGreen = 0;
        colorBlue = 0;
        chaseRed = 255;
        chaseGreen = 140;
        chaseBlue = 0;
        chasePos = 0;
        break;
case 8: colorRed = 0; //blue wipe
```

```

    colorGreen = 0;
    colorBlue = 0;
    chaseRed = 0;
    chaseGreen = 0;
    chaseBlue = 255;
    chasePos = 0;
    break;
case 9: //rainbow static
    for(int z=0;z<256;z++)
    {
        for(int i=0;i<PIXEL_COUNT;i++)
        {
            strip.setPixelColor(i, Wheel((i+z) & 255));
        }
    }
    strip.show();
    break;
case 10: //rainbow shuffle
    for(int z=0;z<256;z++)
    {
        for(int i=0;i<PIXEL_COUNT;i++)
        {
            strip.setPixelColor(i, Wheel((i+z) & 255));
        }
    }
    strip.show();
    break;
case 11: colorRed = 0; //Off
    colorGreen = 0;
    colorBlue = 0;
    break;
case 12: colorRed = 0; //Off
    colorGreen = 0;
    colorBlue = 0;
    break;
case 13: colorRed = 0; //Off
    colorGreen = 0;
    colorBlue = 0;
    break;
case 14: colorRed = 0; //Off
    colorGreen = 0;
    colorBlue = 0;
    break;
case 15: colorRed = 0; //Off
    colorGreen = 0;
    colorBlue = 0;

    break;

```

```

}
if(i != 9 && i != 10)
{
    for(int i=0;i<PIXEL_COUNT;i++) strip.setPixelColor(i, strip.Color(colorRed,colorGreen,colorBlue));
    strip.show();
}
}

void runShow(int i) {
    if(i==0 || i==1 || i==2 || i==3 || i==4 || i==9) //static colors
    {
    }
    else if(i==5 || i==6) //chasers
    {
        strip.setPixelColor(chasePos,strip.Color(colorRed,colorGreen,colorBlue));
        strip.setPixelColor(chasePos+1,strip.Color(colorRed,colorGreen,colorBlue));
        strip.setPixelColor(chasePos+2,strip.Color(colorRed,colorGreen,colorBlue));
        chasePos++;
        if(chasePos > PIXEL_COUNT - 4) chasePos = 0;
        strip.setPixelColor(chasePos,strip.Color(chaseRed,chaseGreen,chaseBlue));
        strip.setPixelColor(chasePos+1,strip.Color(chaseRed,chaseGreen,chaseBlue));
        strip.setPixelColor(chasePos+2,strip.Color(chaseRed,chaseGreen,chaseBlue));
        strip.show();
    }
    else if(i==7 || i==8) //wipers
    {
        strip.setPixelColor(chasePos, strip.Color(chaseRed,chaseGreen,chaseBlue));
        strip.show();
        chasePos++;
        if(chasePos>PIXEL_COUNT)
        {
            chasePos = 0;
            for(int i=0;i<PIXEL_COUNT;i++) strip.setPixelColor(i, strip.Color(colorRed,colorGreen,colorBlue));
            strip.show();
        }
    }
    else if(i==10) //shuffle
    {
        strip.setPixelColor(PIXEL_COUNT-1, strip.getPixelColor(0));
        for(int i=0;i<PIXEL_COUNT-1;i++)
        {
            strip.setPixelColor(i, strip.getPixelColor(i+1));
        }
        strip.show();
    }
    else if(i==11 || i==12 || i==13 || i==14 || i==15) //not implemented yet
    {
    }
}

```

```
}
```

```
// Input a value 0 to 255 to get a color value.
```

```
// The colours are a transition r - g - b - back to r.
```

```
uint32_t Wheel(byte WheelPos) {
```

```
    WheelPos = 255 - WheelPos;
```

```
    if(WheelPos < 85) {
```

```
        return strip.Color(255 - WheelPos * 3, 0, WheelPos * 3);
```

```
    }
```

```
    if(WheelPos < 170) {
```

```
        WheelPos -= 85;
```

```
        return strip.Color(0, WheelPos * 3, 255 - WheelPos * 3);
```

```
    }
```

```
    WheelPos -= 170;
```

```
    return strip.Color(WheelPos * 3, 255 - WheelPos * 3, 0);
```


RoboRio controller

Example in C++

```
int LEDMode = 0;
double LEDModeOut = 0;

void TeleopPeriodic()
{
    //cycle the LED modes when button 3 pushed
    if (Xbox.GetRawButton(3))
    {
        LEDMode++;
        if (LEDMode > 9) LEDMode = 0;
        LEDModeOut = (LEDMode * 0.3136) + 0.1568;
        LEDout.SetVoltage(LEDModeOut);
        SmartDashboard::PutNumber("LED", LEDModeOut);
    }
    if (Xbox.GetRawButton(4))
    {
        LEDMode = 0;
        LEDModeOut = (LEDMode * 0.3136) + 0.1568;
        LEDout.SetVoltage(LEDModeOut);
        SmartDashboard::PutNumber("LED", LEDModeOut);
    }
}
```