

TEAM REWIND 4419

LED onboard robot using WPI lib

WPI Lib Help Doc

<https://docs.wpilib.org/en/stable/docs/software/hardware-apis/misc/addressable-leds.html>

Sample LED code

<https://github.com/wpilibsuite/allwpilib/blob/main/wpilibcExamples/src/main/cpp/examples/AddressableLED/cpp/Robot.cpp>



ELECTRICAL COMPONENTS

1. W2812B LED lights DC 5V (60 led > 30 led)
 - a. RGB not RGBW
 - b. IP65 = good (waterproof and protected)
2. DC Buck Voltage Step Down Converter 12v > 5v (~5amp)
 - a. Any voltage step down from 12V to 5V with min 5amp will work
 - b. Some come with adjustable voltage to compensate for LED power



DC Buck Module, DROK Adjustable Buck Converter Step Down Voltage Regulator 6V-32V 30V 24V 12V to 1.5-32V 5V 5A LCD Power Supply Volt Reducer Transformer Module Board with USB Port Protective Case

★★★★☆ ~ 1,456

\$12.99 **prime** Overnight 4 AM - 8 AM

Price dropped 7% (was \$13.99 when added to List)

Size : Buck 6-32V to 1.5-32V

2 Used & New from \$12.99



WS2812B ECO LED Strip, BTF-LIGHTING Chasing Effects 5050SMD Individually Addressable 3.3FT 60Pixels/m Flexible Black FPCB Dream Color IP65 Waterproof for Bedroom DIY Projectsetc DC5V

★★★★☆ ~ 1,557

We don't know when or if this item will be back in stock.

Go to the product detail page.

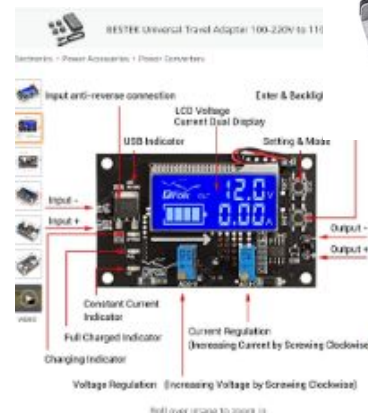


DROK DC-DC Buck Converter 12V/24V to 5V 5A/25W Step-Down Voltage Transformer Volt Regulator Power Supply Inverter Module Voltage Drop Transfer Board SR Waterproof for Car Auto Vehicle Motor etc

★★★★☆ ~ 131

\$10.25 **prime**

Price dropped 6% (was \$10.87 when added to List)



Down Voltage Regulator 6V-32V 30V 24V 12V to 1.5-32V 5V 5A LCD Power Supply Volt Reducer Transformer Module Board with USB Port Protective Case

Visit the DROK Store

★★★★☆ ~ 1,456 ratings

104 answered questions

1 Price Change

-35% \$12.99

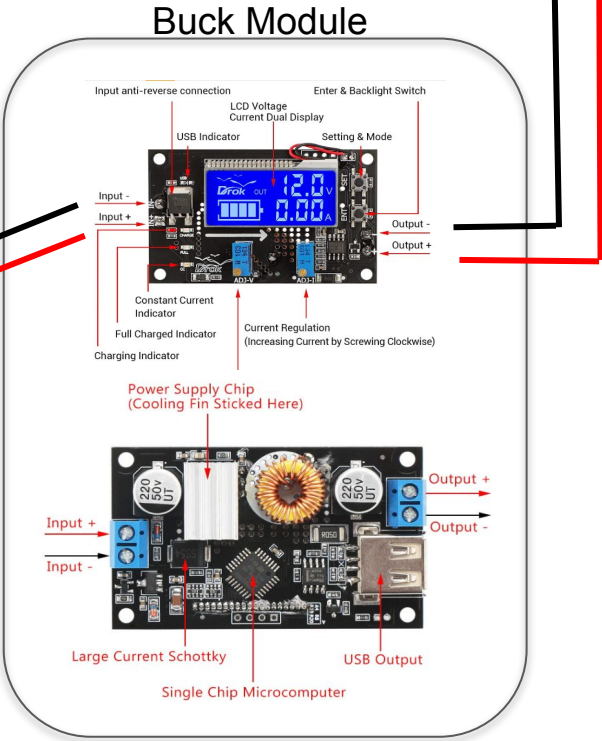
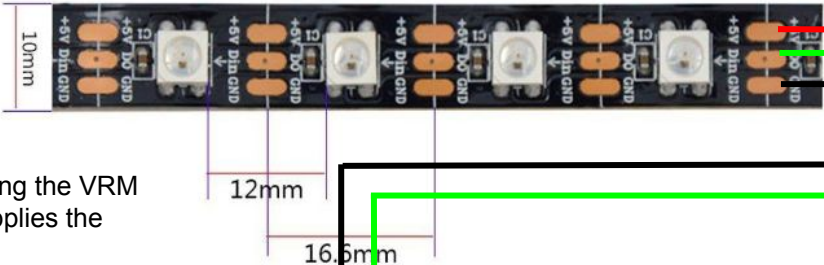
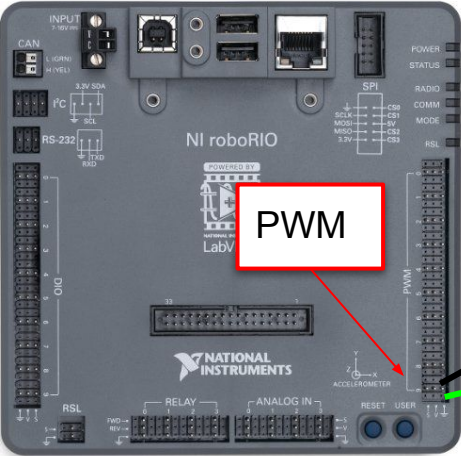
Unit Price \$4.99 @

prime Overnight

Electrical Circuit

WS2812B 60LED

Circuit avoids using the VRM 12v 2A which supplies the radio.



Sample Code C++

```
// Copyright (c) FIRST and other WPILib contributors.  
// Open Source Software; you can modify and/or share it under the terms of  
// the WPILib BSD license file in the root directory of this project.  
  
#include <array>  
#include <frc/AddressableLED.h>  
#include <frc/TimedRobot.h>  
#include <frc/smartdashboard/SmartDashboard.h>  
  
class Robot : public frc::TimedRobot {  
    static constexpr int kLength = 60;  
  
    // PWM port 9  
    // Must be a PWM header, not MXP or DIO  
    frc::AddressableLED m_led{9};  
    std::array<frc::AddressableLED::LEDData, kLength>  
        m_ledBuffer; // Reuse the buffer  
    // Store what the last hue of the first pixel is  
    int firstPixelHue = 0;  
  
public:  
    void Rainbow() {  
        // For every pixel  
        for (int i = 0; i < kLength; i++) {  
            // Calculate the hue - hue is easier for rainbows because the color  
            // of the pixel is dependent on the hue
```