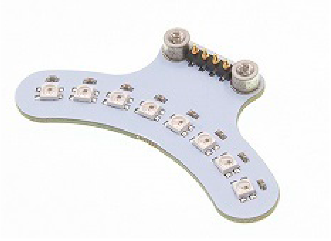
# Expansion：Rainbow LED Ligth Bar Module

## Contents

## 7.1. Introduction

Ring:bit car V2 light-bar module is the dedicated module designed for Ring:bit Car V2. It brings you mutiple color with simple assemblling methods.

It comes with 8 Rainbow full color LEDs. You can update it to a car with auto-headlights and rainbow lights.



## 7.2. Features

Input voltage 3V~5V，driving directly by micro:bit.

Standard 3 line GVS port, occupying only one I/O port.

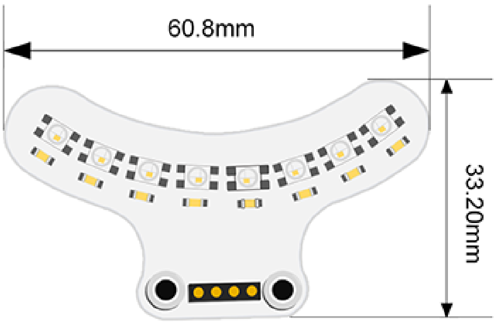
8 small LEDs, power saving and energy saving.

Each led can be programmed seperately with RGB color.

## 7.3. Parameter

| Items | Parameter | Note |
|---|---|---|
| Name | Ring:bit Car v2 ligth bar | |
| SKU | EF03425 | |
| Working Voltage | DC 3-5V | |
| Connections | Ring:bit Car dedicated pin ports | Fixed with scews |
| Output Signal | Simulation | |
| LEDs | 8 units | |
| Size | 60.8×33.20mm | |
| NW | 5.7g | |

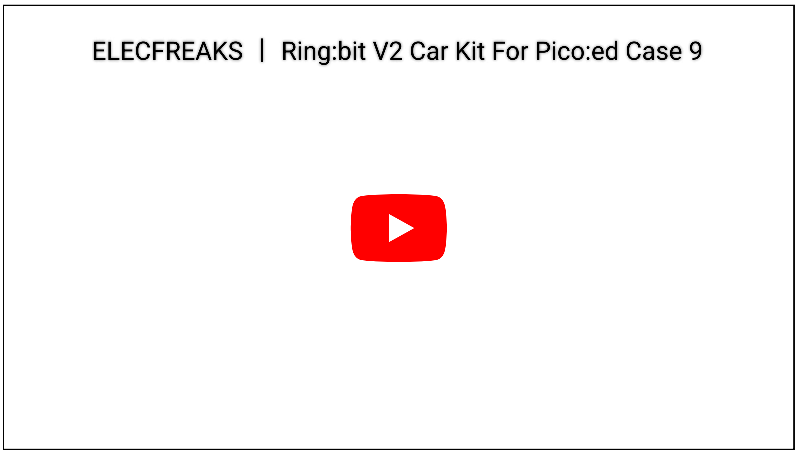
## 7.4. Outlook and Dimensions



## 7.5. Quick to Start

### Hardware Connections

First, insert this module to the baseboard of the Ring:bit car V2,Fix with the screws,Completed.


ELECFREAKS | Ring:bit V2 Car Kit For Pico:ed Case 9

## 7.6. Programming

Preparation for programming: Introduction

## Sample Code

```
# Import the modules we need:
import board
from ringbit import *
from digitalio import *
from neopixel_write import *
from time import *
from random import *

# Set the pins and directions of the wheels and the light bar
ringbit = Ringbit(board.P1, board.P2)
pin = DigitalInOut(board.P0)
pin.direction = Direction.OUTPUT

# Set the speed of the car
ringbit.set_speed(100, 100)

# While true, change the color of the light bar
while True:
    rings = []
    for i in range(30):
        rings.append(randint(0, 255))
    neopixel_write(pin, bytearray(rings))
    sleep(0.1)
```

## Details of the Program:

1.Import the modules that we need. `board` is the common container, and you can connect the pins you'd like to use through it; `ringbit` module contains classes and functions for Ring:bit smart car operations, the `time` module contains functions for time operations, the `digitalio` module contains classes for basic digital IO access, the `neopixel_write` module contains functions for strip operations, and the `random` module contains functions that can return random numbers

```
import board
from ringbit import *
from digitalio import *
from neopixel_write import *
from time import *
from random import *
```

2.Set the pins of the servos and the pins and directions of the breakout board.

```
ringbit = Ringbit(board.P1, board.P2)
pin = DigitalInOut(board.P0)
pin.direction = Direction.OUTPUT
```

3.Set the speed of the car

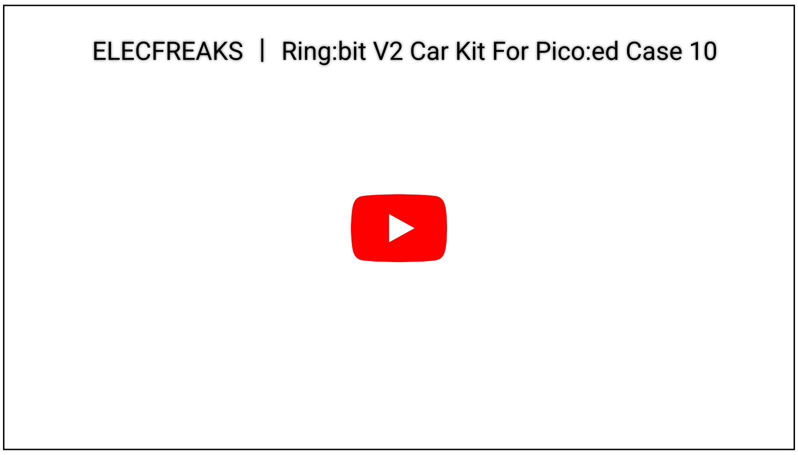
```
ringbit.set_speed(100, 100)
```

4.While true, change the RGB value of each LED in the light bar to detect if the A\B button is pressed to set the speed of the left and right wheels of the Ring:bit smart car

```
while True:
    rings = []
    for i in range(30):
        rings.append(randint(0, 255))
    neopixel_write(pin, bytearray(rings))
    sleep(0.1)
```

# 7.7. Result

The two lights on the bottom of the Ring:bit car as well as the light bar change colour randomly to give a colourful effect; press A to move the car forward at full speed, press B to stop the car.



## 7.8. FAQ

## 7.9. Relevant File

By ELECFREAKS Team