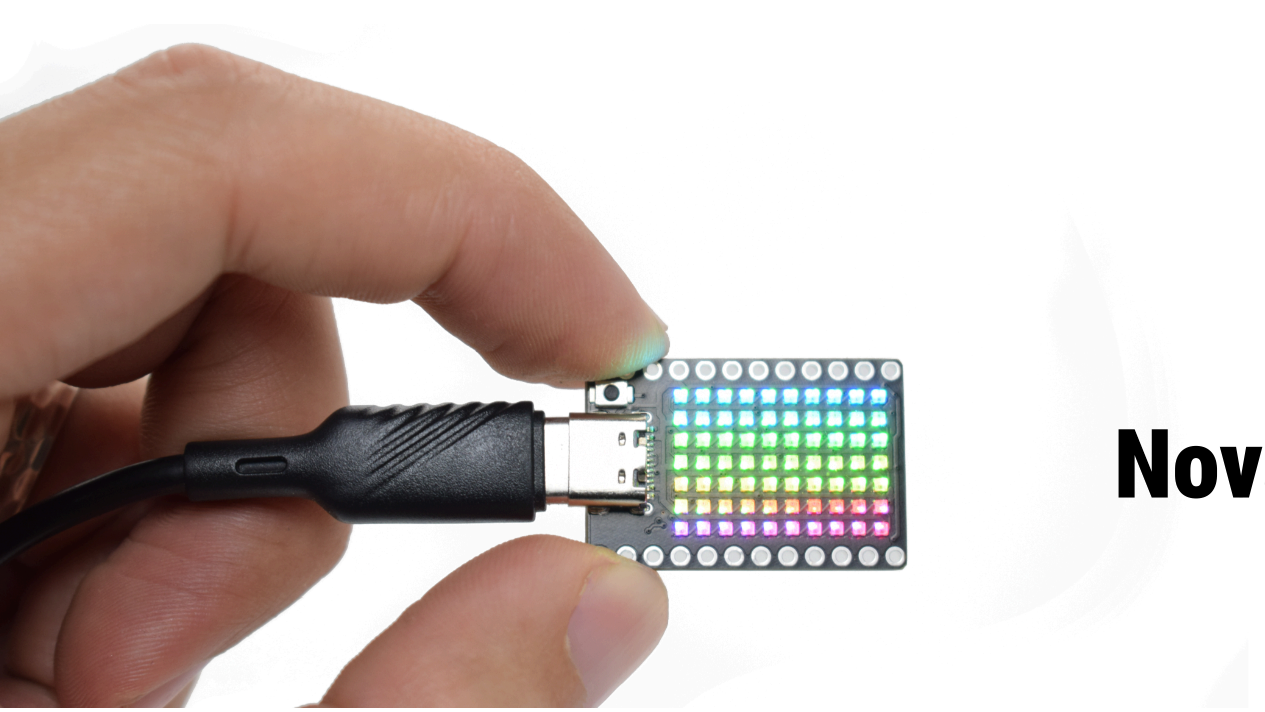


Nova

1. Introduction

The **Nova board** is an RP2040-based development board designed for engineers, developers, and embedded enthusiasts. With its powerful dual-core ARM Cortex M0+ processor and integrated 7x10 LED matrix, Nova offers a compact and versatile platform for various applications.

The board is **open-source hardware**, making it easy to customize and extend for your specific needs.

2. Features

- **Microcontroller**: Raspberry Pi RP2040 with dual ARM Cortex M0+ cores running up to 133 MHz.
- **Memory**: 264KB of SRAM, with support for external flash 16M-bit (2M-byte).
- **7x10 LED Matrix**: Integrated with 70 RGB LEDs (addressable WS2812).
- **GPIO**: Accessible for various I/O purposes.
- **Power Input**: Via USB Type-C or external power.
- **Connectivity**: USB and various standard GPIO pins for communication protocols (I2C, SPI, UART).
- **Open-source**: Fully customizable hardware with publicly available schematics and examples.

3. Getting Started

3.1 Setting Up Your Development Environment

To start developing with the Nova board, follow these steps:

1. **Install the Arduino IDE** (or VSCode with the PlatformIO extension).
    - Download the latest version from Arduino's website.
    - For PlatformIO, follow the guide here.
2. **Install the RP2040 Board Package**:
    - In the Arduino IDE, go to **File > Preferences**.
    - Under **Additional Boards Manager URLs**, paste this URL:

      ```
      https://github.com/earlephilhower/arduino-pico/releases/download/global/package_rp2040_index.json
      ```

    - Open the **Boards Manager** and search for "RP2040." Install the package.
3. **Install the FastLED Library**:
    - In the Arduino IDE, go to **Sketch > Include Library > Manage Libraries**.
    - Search for "FastLED" and install it, (any other suitable library for the WS2812 LEDs can also be used such as Neopixel).
4. **Connect the Nova Board**:
    - Plug in the Nova board via the USB Type-C connector.
    - Select the Nova board from the **Tools > Board** menu in the Arduino IDE.
    - Choose the correct port under **Tools > Port**.

3.2 Hello World: Blink Example

Let's begin with a simple LED blink example:

```
#include <FastLED.h>  // Include the FastLED library to control the LED matrix

#define DATA_PIN 22    // Define the pin where the LED matrix is connected (GPIO22)
#define NUM_LEDS 70    // Define the number of LEDs in the matrix (7x10 = 70)

CRGB leds[NUM_LEDS];   // Create an array to store the color data for each LED

void setup() {
  // Initialize the LED matrix using the NEOPIXEL type and link it to the data pin
  // FastLED.addLeds<CHIPSET, DATA_PIN>(led array, number of LEDs);
  FastLED.addLeds<NEOPIXEL, DATA_PIN>(leds, NUM_LEDS);
}

void loop() {
  leds[0] = CRGB::Red;   // Set LED 0 (top-right corner) to red
  FastLED.show();        // Send the data to the LED matrix to display the change
  delay(500);            // Wait for 500 milliseconds (half a second)

  leds[0] = CRGB::Black; // Turn off LED 0 (set it to black)
```

```
  FastLED.show();         // Update the LED matrix to show the change
  delay(500);             // Wait for 500 milliseconds before repeating the
loop
}
```

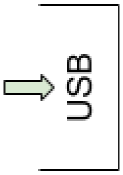
Upload the code to your Nova board, and you should see the first LED (top right corner) blinking!

4. LED Matrix Layout

The Nova board comes with a 7x10 LED matrix. Here's how the LEDs are arranged:

- **Rows**: The matrix is divided into rows, each containing 10 LEDs.
- **Addressing**: The LEDs are numbered from 0 to 69. The first row (LED0 to LED9) runs from right to left, while the second row (LED10 to LED10) runs from left to right, and so on.

Here's a quick reference:

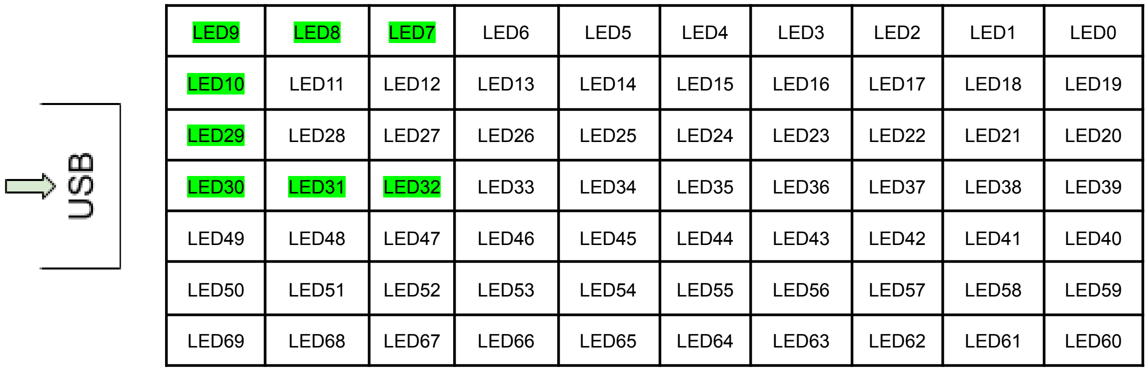


LED9	LED8	LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
LED10	LED11	LED12	LED13	LED14	LED15	LED16	LED17	LED18	LED19
LED29	LED28	LED27	LED26	LED25	LED24	LED23	LED22	LED21	LED20
LED30	LED31	LED32	LED33	LED34	LED35	LED36	LED37	LED38	LED39
LED49	LED48	LED47	LED46	LED45	LED44	LED43	LED42	LED41	LED40
LED50	LED51	LED52	LED53	LED54	LED55	LED56	LED57	LED58	LED59
LED69	LED68	LED67	LED66	LED65	LED64	LED63	LED62	LED61	LED60

4.1 Displaying Custom Patterns

You can use the FastLED library to control the matrix. Here's an example to display the letter 'C' on the left side of the matrix:

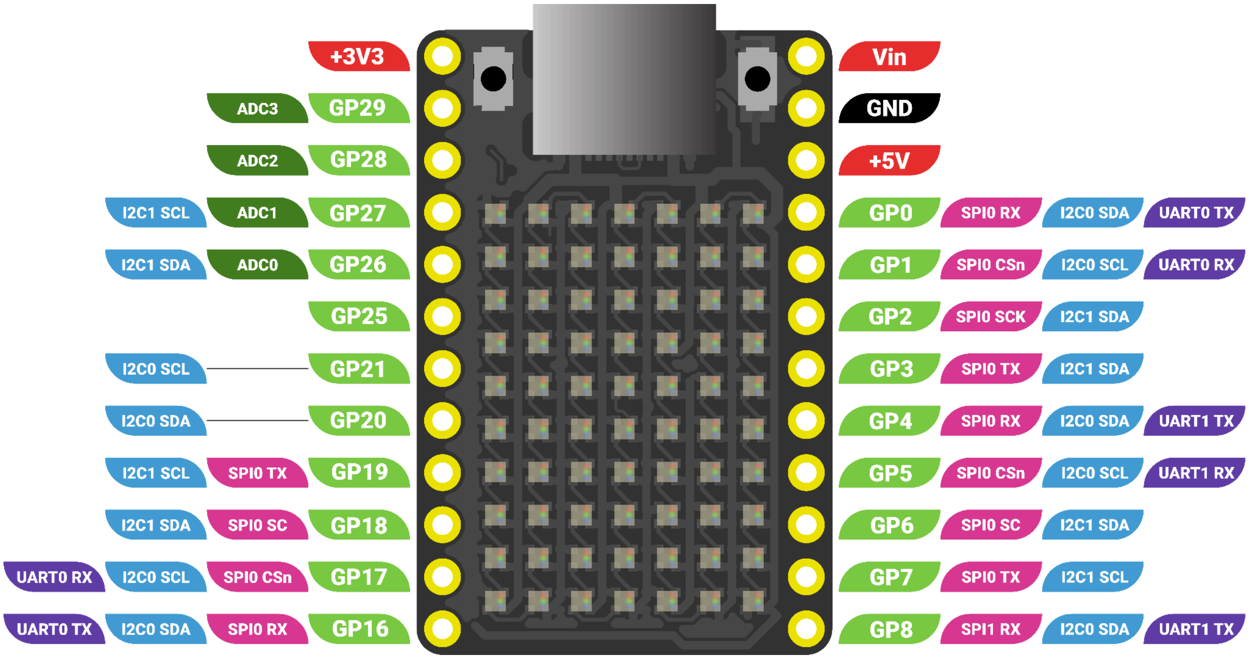
```
#include <FastLED.h>

#define DATA_PIN 22
#define NUM_LEDS 70
CRGB leds[NUM_LEDS];

void setup() {
  FastLED.addLeds<NEOPIXEL, DATA_PIN>(leds, NUM_LEDS);
  FastLED.clear();  // Clear the LED matrix
}

void loop() {
  displayLetterC();  // Display the smaller letter 'C' on the matrix
  FastLED.show();    // Update the LED matrix
  delay(5000);       // Hold the letter 'C' for 5 seconds
}

void displayLetterC() {
  // Small 'C' that fits in 4 rows and 3 columns

  // Left vertical column of 'C'
  leds[9] = CRGB::Green;   // Row 0
  leds[10] = CRGB::Green;  // Row 1
  leds[29] = CRGB::Green;  // Row 2
  leds[30] = CRGB::Green;  // Row 3

  // Top part of 'C' (Row 0)
  leds[8] = CRGB::Green;
  leds[7] = CRGB::Green;

  // Bottom part of 'C' (Row 3)
  leds[31] = CRGB::Green;
  leds[32] = CRGB::Green;
}
```



LED9	LED8	LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0
LED10	LED11	LED12	LED13	LED14	LED15	LED16	LED17	LED18	LED19
LED29	LED28	LED27	LED26	LED25	LED24	LED23	LED22	LED21	LED20
LED30	LED31	LED32	LED33	LED34	LED35	LED36	LED37	LED38	LED39
LED49	LED48	LED47	LED46	LED45	LED44	LED43	LED42	LED41	LED40
LED50	LED51	LED52	LED53	LED54	LED55	LED56	LED57	LED58	LED59
LED69	LED68	LED67	LED66	LED65	LED64	LED63	LED62	LED61	LED60

6. Pinout

- **Data Pin for LED Matrix**: GPIO22
- Other key GPIO pins are exposed for various I/O tasks. Please refer to the pinout diagram for detailed pin functions:



7. Powering the Board and Power Consumption

Power Regulation:

- Nova has two regulators: a **buck converter** and an **LDO (linear regulator)**.
- The **buck converter** handles input voltages from **7VDC to 18VDC** via the Vin pin and provides **5VDC** for the circuit, which is needed for the addressable LEDs.
- The **LDO** takes the **5VDC** from either the buck converter or USB and regulates it down to **3.3VDC** for the RP2040 and other 3.3V components.

Powering the Board:

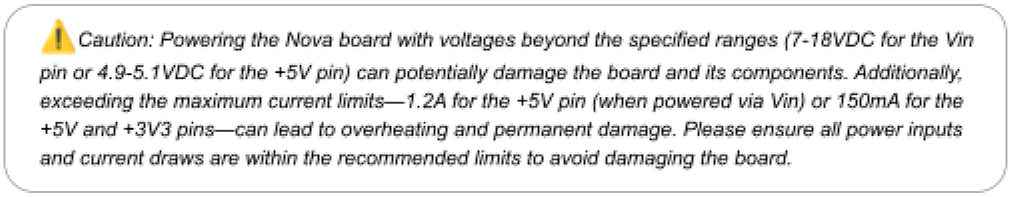
- The board can be powered via:
  - **Vin pin**: Accepts **7-18VDC**.
  - **+5V pin**: Accepts **4.8-5.2VDC**.
- **+3V3 pin** is not meant to power the whole board, only the MCU (without LEDs).

Power Output from +5V Pin:

- When powered by the **Vin pin**, the **+5V pin** can provide up to **1.2A**.
- When powered by **USB**, the **+5V pin** can provide up to **150mA**.

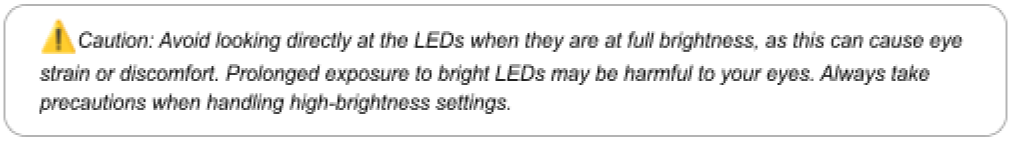
+3V3 Pin:

- The recommended maximum current draw from the **+3V3 pin** is **150mA** in all cases.



⚠ *Caution: Powering the Nova board with voltages beyond the specified ranges (7-18VDC for the Vin pin or 4.9-5.1VDC for the +5V pin) can potentially damage the board and its components. Additionally, exceeding the maximum current limits—1.2A for the +5V pin (when powered via Vin) or 150mA for the +5V and +3V3 pins—can lead to overheating and permanent damage. Please ensure all power inputs and current draws are within the recommended limits to avoid damaging the board.*

Power Consumption
The Nova board consumes approximately 9 watts when all LEDs are at full brightness in white color. It's recommended to avoid running all LEDs at maximum brightness for extended periods, as this can cause the board to overheat, potentially leading to damage to the board or its components.



⚠ *Caution: Avoid looking directly at the LEDs when they are at full brightness, as this can cause eye strain or discomfort. Prolonged exposure to bright LEDs may be harmful to your eyes. Always take precautions when handling high-brightness settings.*

8. Schematics and Hardware

The Nova board is open-source hardware. You can find the schematics, PCB design files, and BOM (Bill of Materials) on our GitHub repository.

9. Troubleshooting

Common Issues

1. **Board not recognized**:
   - Ensure that the USB cable is working and data-capable.
   - Check if the correct port and board are selected in the Arduino IDE.
2. **LEDs not lighting up**:
   - Verify that the data pin is identified in the code properly (GPIO22).
   - Make sure the FastLED library is installed and included in your sketch (any other suitable library for the WS2812 LEDs can also be used such as Neopixel).
   - You can enter bootloader mode using the 'Boot' button. To do this, hold down the 'Boot' button (located to the right of the USB port) while connecting the board to your PC via USB. This will put the board into bootloader mode and open a mass storage window on your computer. This is useful if the board is not being detected by the PC.

10. Contact & Support

For further questions or support, you can reach us at support@vcclabs.com. Join our community on GitHub and check out the forums for more resources and user projects.