Custom ESP32-C6 Development Board

Rupak Poddar

Index

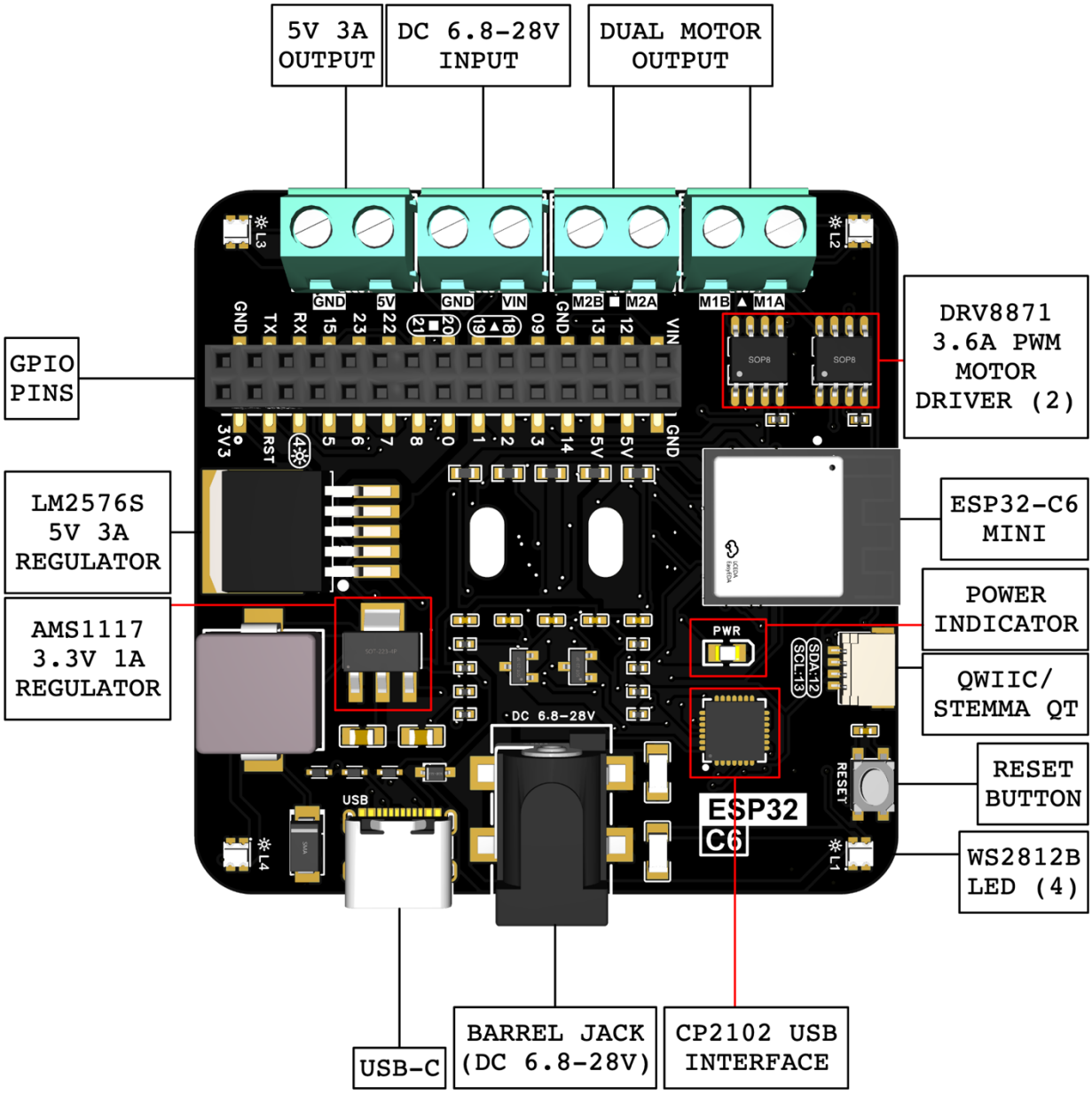
Title	Page(s)
Overview	2-3
Arduino IDE	4-6
Motor Drivers	7-8
QWIIC	9
NeoPixel	10-11

Overview

Discover the potential of the ESP32-C6 based development board, engineered to simplify your prototyping needs with integrated components that streamline connections and enhance functionality.

Highlights:

- **ESP32-C6 Mini** chip offers **WiFi 6** & **BLE 5.0**.
- Dual **DRV8871 6.8-28V 3.6A** PWM High Power Motor Drivers.
- **Qwiic/Stemma QT** support for easy sensor and accessory board connections.
- Built-in **WS2812B LEDs** (4) for status lights or animations.
- **USB-C** for power and programming with **CP2102** USB interface.
- **DC barrel jack** and screw terminal power input accepts 6.8-28V.
- **AMS1117 3.3V 1A** voltage regulator for stable MCU power.
- **LM2576S 5V 3A** high power voltage regulator to power external modules.
- GPIO headers for full ESP32-C6 chip capability.
- Compact form factor with no bottom components for easy integration.

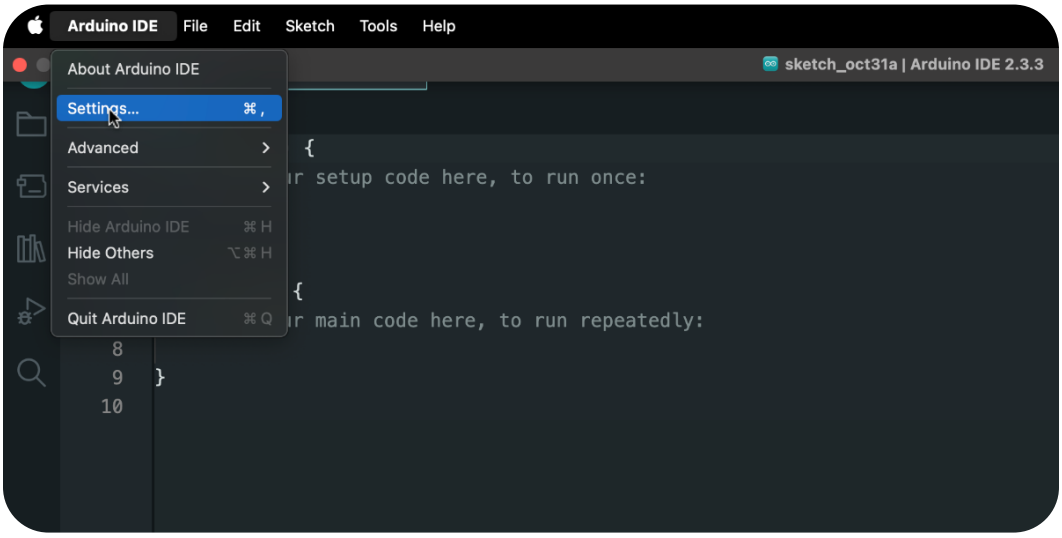


ESP32-C6 Mini Datasheet:
https://www.espressif.com/sites/default/files/documentation/esp32-c6-mini-1_mini-1u_datasheet_en.pdf

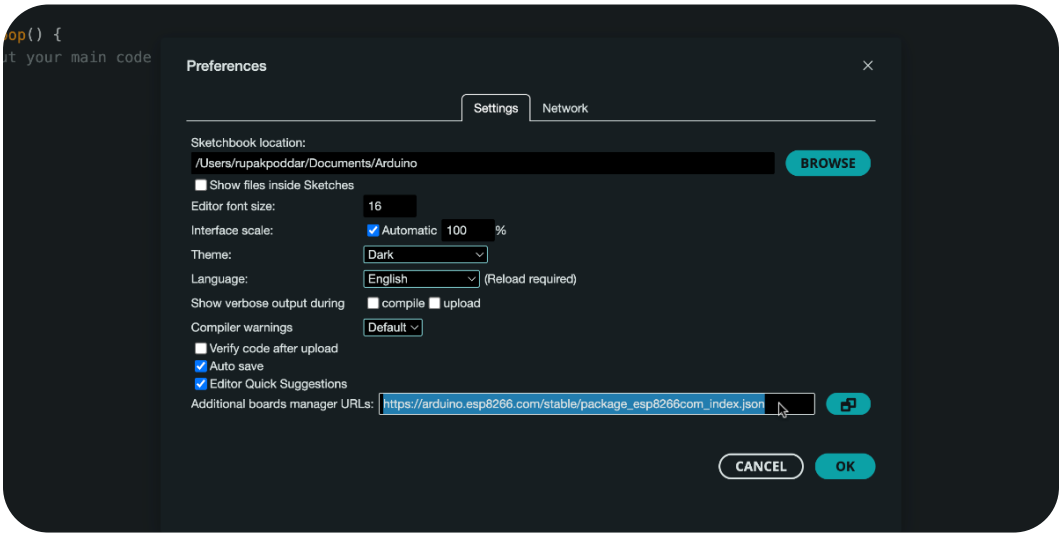
Arduino IDE

This is a one-time setup.

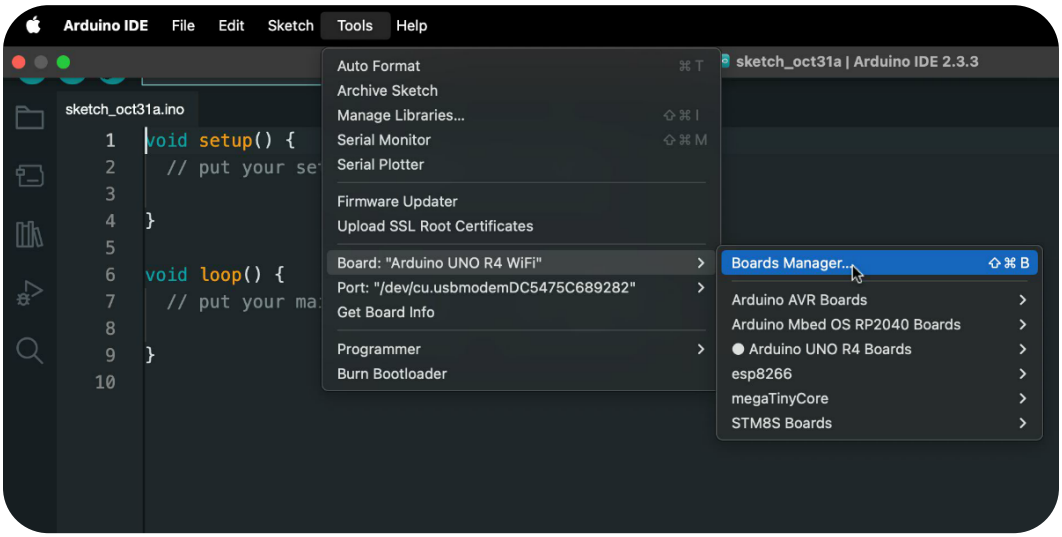
Step 1: Open Arduino IDE and go to Settings/Preferences.



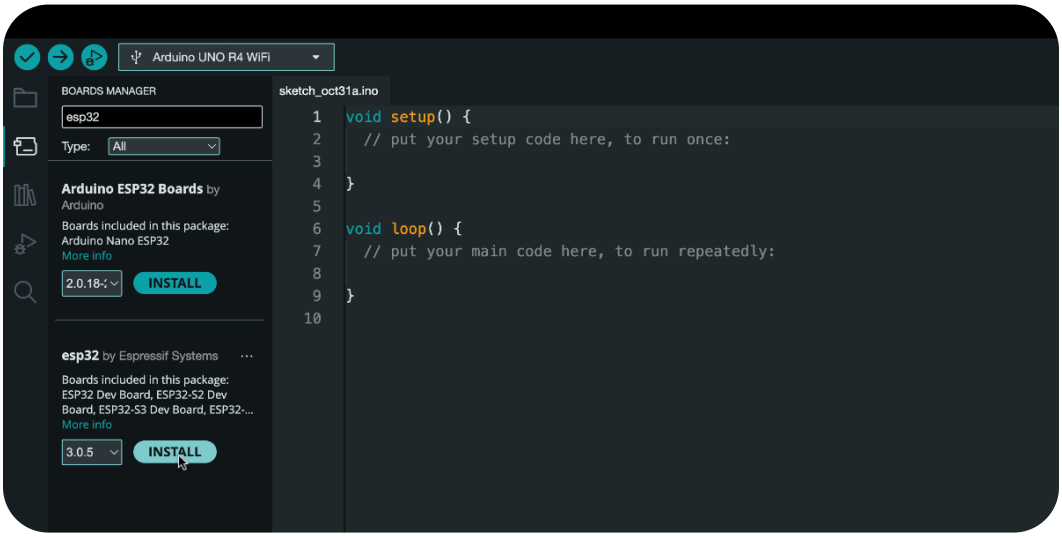
Step 2: In Additional Board Manager URLs, paste: https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json



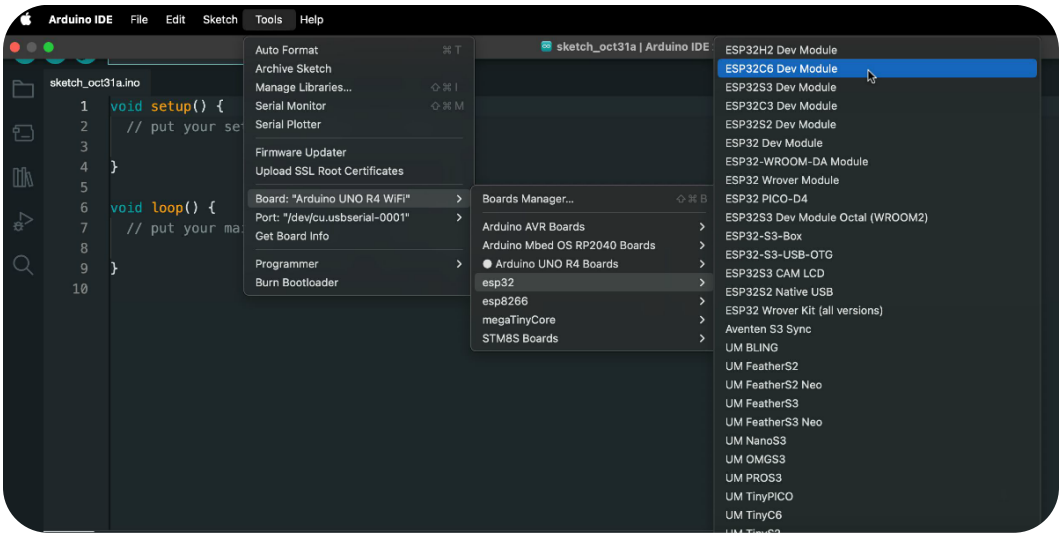
Step 3: Go to **Tools → Board → Board Manager**.



Step 4: In the search box, type **esp32** and install the package by **Espressif Systems**.



Step 5: Connect your board to the PC via **USB-C**, select **ESP32C6 Dev Module**, and choose the **port**.



You can now proceed to flash a sketch to the ESP32 board using the Arduino IDE.

Motor Drivers

The development board includes two DRV8871 motor drivers (6.8-28V, 3.6A PWM) capable of operating two DC motors or functioning as four high-power output pins. These outputs can directly control high-power devices such as relays and solenoids, eliminating the need for external components/modules.

The motor drivers are configured to draw power directly from the board's primary power supply. Motor Driver 1 is accessible via GPIO pins 18 and 19, while Motor Driver 2 is available through GPIO pins 20 and 21 of the ESP32-C6. Pulse Width Modulation (PWM) control facilitates precise speed regulation for connected devices.

A code template is provided on the following page to utilize the motor drivers.

```
/*
* Motor drivers are connected to GPIO pins 18-21 of the ESP32-C6.
* Motor drivers use the main power source of the board.
* PWM (Pulse Width Modulation) signals control motor speed or
    output voltage by varying the duty cycle.
*/

// Define motor control pins.
#define MOTOR_1A 18 // Motor 1, Input A
#define MOTOR_1B 19 // Motor 1, Input B
#define MOTOR_2A 20 // Motor 2, Input A
#define MOTOR_2B 21 // Motor 2, Input B

void setup() {
    // Set motor control pins as outputs
    pinMode(MOTOR_1A, OUTPUT);
    pinMode(MOTOR_1B, OUTPUT);
    pinMode(MOTOR_2A, OUTPUT);
    pinMode(MOTOR_2B, OUTPUT);
}

void loop() {
    // Move Motor 1 forward
    // A value of 200 (out of 255) corresponds to 80% power.
    analogWrite(MOTOR_1A, 200); // MOTOR_1A set to 200 (forward motion)
    analogWrite(MOTOR_1B, 0);   // MOTOR_1B set to 0 (no reverse motion)
    delay(2000);                // Motor runs for 2 seconds

    // Stop Motor 1 by setting both MOTOR_1A and MOTOR_1B to 0
    analogWrite(MOTOR_1A, 0);
    analogWrite(MOTOR_1B, 0);
    delay(1000);                // Pause for 1 second

    // Move Motor 2 in reverse by applying a PWM signal to MOTOR_2B
    // A value of 150 indicates 58% power for reverse motion.
    analogWrite(MOTOR_2A, 0);   // MOTOR_2A set to 0 (no forward motion)
    analogWrite(MOTOR_2B, 150); // MOTOR_2B set to 150 (reverse motion)
    delay(2000);                // Motor runs in reverse for 2 seconds

    // Stop Motor 2 by setting both MOTOR_2A and MOTOR_2B to 0
    analogWrite(MOTOR_2A, 0);
    analogWrite(MOTOR_2B, 0);
    delay(1000);                // Pause for 1 second
}
```

QWIIC

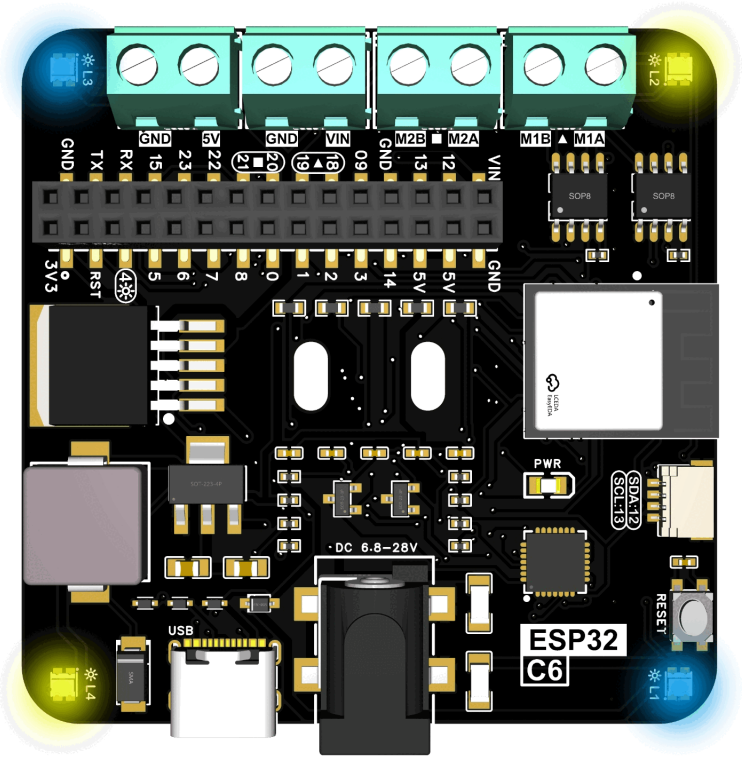
The onboard Qwiic connector supports I2C modules from SparkFun and Adafruit, making it easy to expand with additional sensors and peripherals. The SDA line is mapped to GPIO 12, and the SCL line to GPIO 13, facilitating smooth communication between devices.

```
#include <Wire.h>
#include "libraryName.h"

void setup() {
    Wire.begin(12, 13);
    libraryName.begin(Wire);
}
```

NeoPixel

Built-in WS2812B addressable LEDs offer customizable lighting options, perfect for adding visual feedback or aesthetic appeal to your projects. These LEDs can be easily operated using popular off-the-shelf libraries such as Adafruit NeoPixel, FastLED, and others. The LEDs are connected to GPIO 04 of the ESP32, making integration simple and flexible for a variety of applications.



Usage with Adafruit NeoPixel library:

```
#include <Adafruit_NeoPixel.h>

#define NUM_LEDS 4
#define DATA_PIN 4

Adafruit_NeoPixel leds(NUM_LEDS, DATA_PIN, NEO_GRB + NEO_KHZ800);

void setup() {
    leds.begin();
    //
    // ...
    //
    leds.show();
}
```

Usage with FastLED library:

```
#include <FastLED.h>

#define NUM_LEDS 4
#define DATA_PIN 4

CRGB leds[NUM_LEDS];

void setup() {
    FastLED.addLeds<NEOPIXEL, DATA_PIN>(leds, NUM_LEDS);
    //
    // ...
    //
    FastLED.show();
}
```