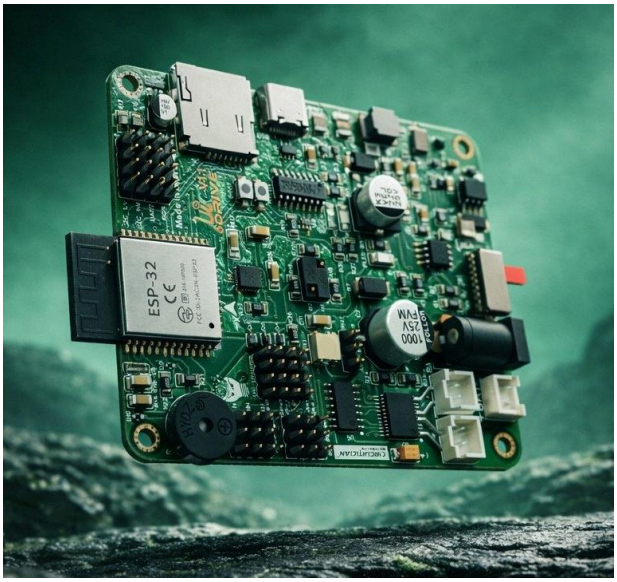
ESP32-Based Dual Motor Driver & Sensor Platform



µDrive V1.1

Power • Protection • Precision

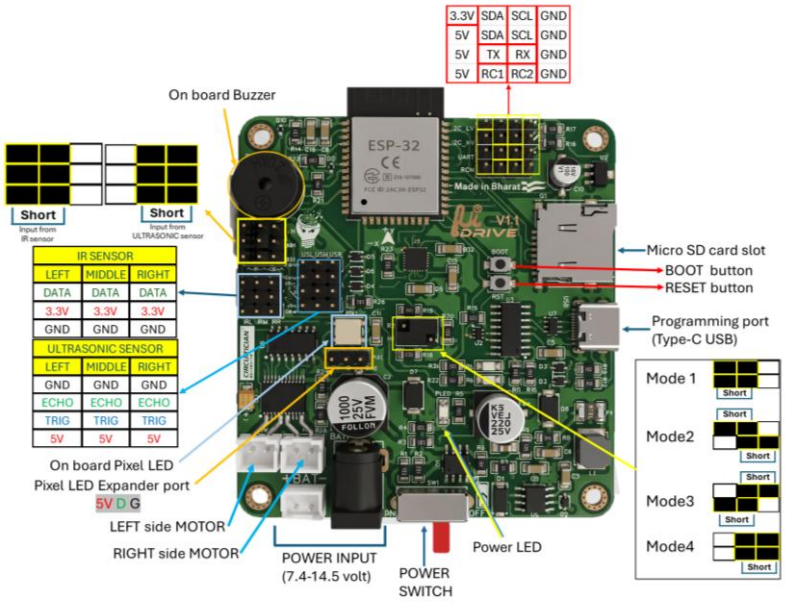
Key Features

- ESP32 with built-in Wi-Fi & Bluetooth
- Dual DC motor Driver
- 3× IR sensors port (left / middle / right)
- 3× Ultrasonic sensor ports (left / middle / right)
- On-board Neo Pixel RGB LED
- On-board piezo buzzer
- Micro SD card slot for data logging
- Battery voltage monitor (7.4 – 14.5 V input)
- I2C, UART, RC input headers (3.3 V & 5 V)
- 4 selectable hardware modes via jumper shorts
- Type-C USB programming port
- Compact form factor with all peripherals on-board
- USB VBUS surge & transient protection for safe programming
- On-board short-circuit protection on 5 V & 3.3 V power rails
- Sensor data collection support for machine learning datasets
- Edge AI ready – run lightweight ML models directly on the ESP32

Applications

- Line-following robots
- Maze-solving robots
- Obstacle-avoidance robots
- RC-controlled ground robots
- Sensor data logging
- Educational robotics & STEM competitions
- Rapid prototyping of ESP32-based robots

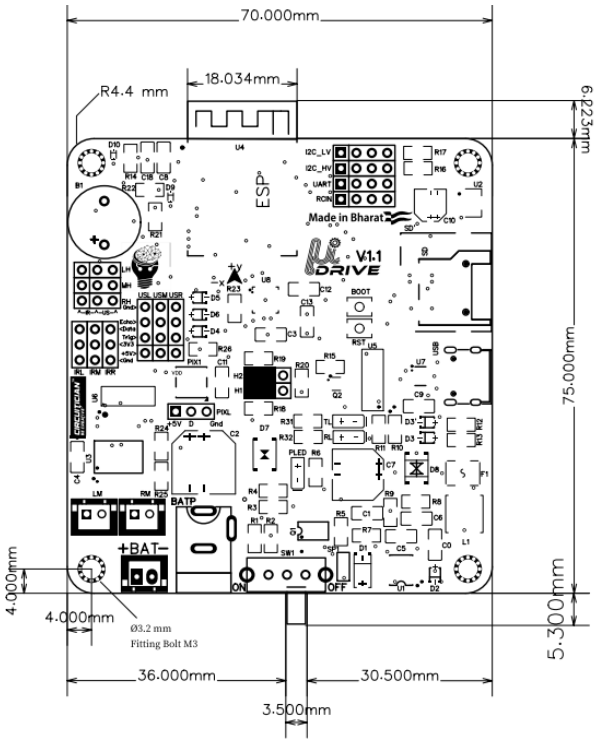
PINOUT DIAGRAM



µDrive V1.1 Pinout

CIRCUITICIAN™ BE CREATIVE

MECHANICAL DIMENSIONS



Parameter	Value
Board Length	75 mm
Board Width	70 mm
PCB Thickness	1.6 mm
Mounting Holes	4 × M3
Corner Radius	4.4 mm

CIRCUITICIAN™ BE CREATIVE

ABSOLUTE MAXIMUM RATINGS

Stresses beyond these values may cause permanent damage to the device.

Parameter	Min	Typ	Max	Unit
Supply Voltage (VBAT)	7.4	11.1	14.5	V
Motor Supply Current (each)	—	—	1.2	A
Total Board Current	—	—	3	A
Logic Supply (ESP32 I/O)	—	3.3	—	V
Storage Temperature	-40	25	85	°C

RECOMMENDED OPERATING CONDITIONS

Parameter	Min	Typ	Max	Unit
Battery Input Voltage	7.4	11.1	14.5	V
Motor PWM Frequency	—	20	—	kHz
PWM Resolution	—	8	—	bit
Buzzer Tone Frequency	500	2000	5000	Hz
IR Sensor Supply	—	3.3	—	V
Ultrasonic Sensor Supply	—	5	—	cm
ADC Reference (Battery pin)	—	3.3	—	V
Battery Voltage Divider Ratio	—	3.127	—	—

PIN ASSIGNMENTS (ESP32)

GPIO	Signal	Description
13	LIN1	Left motor – PWM channel 0 (LEDC CH_L1)
12	LIN2	Left motor – direction control (digital)
27	RIN1	Right motor – PWM channel 1 (LEDC CH_R1)
14	RIN2	Right motor – direction control (digital)
32	TRIG_L	Left IR input / Ultrasonic TRIG
33	TRIG_M	Middle IR input / Ultrasonic TRIG
25	TRIG_R	Right IR input / Ultrasonic TRIG
34	ECHO	Ultrasonic echo input (input-only pin)
26	BUZZER	Piezo active buzzer – LEDC channel 2
2	PIXEL	Neo Pixel data out (WS2812B)
35	BATTERY	Battery voltage sense (ADC input-only)
5	SD_CS	Micro SD card – SPI chip select

CIRCUITICIAN™ BE CREATIVE

µ-DRIVE LIBRARY – FUNCTION REFERENCE

Include the library and call uD.begin() once inside setup(). All functions are called on the global uD object.

Initialization

Function	Description
`uD.begin()`	Initializes all pins, LEDC PWM channels, and the Neo Pixel. Must be called once in setup().

Motor Control

Function	Description
`uD.drive(direction, speed)`	Drives both motors in a specified direction. direction: FWD, BWD, LFT, RIT, or STP. speed: 0–255.
`uD.driveLR(leftSpeed, rightSpeed)`	Sets each motor independently. Positive values = forward, negative = reverse. Range: -255 to 255.
`uD.stop()`	Stops both motors immediately (sets speed to 0).

`Direction macros: FWD=1 BWD=2 LFT=3 RIT=4 STP=5`

Sensors

Function	Description
`uD.enableIRMode()`	Configures the shared sensor pins as digital inputs for IR sensor reading.
`uD.getIR(position)`	Returns 0 if IR sensor detects a surface (active LOW logic). position: LEFT, MIDDLE, or RIGHT.
`uD.enableUltrasonicMode()`	Configures the shared sensor pins as outputs for ultrasonic TRIG and sets ECHO as input.
`uD.getDistance(position)`	Triggers an ultrasonic pulse and returns the measured distance in centimetres (float). position: LEFT, MIDDLE, or RIGHT. Returns 0 on timeout (>30 ms).

`Position macros: LEFT=0 MIDDLE=1 RIGHT=2`

On-Board NeoPixel LED

Function	Description
`uD.colourRGB(colour, brightness)`	Sets the Neo Pixel to a predefined colour at a given brightness (0–255). Uses built-in colour macros.

`Colour macros: RED GREEN BLUE YELLOW PINK CYAN WHITE BLACK`

CIRCUITICIAN™ BE CREATIVE

Buzzer

Function	Description
`uD.buz(count, delayMs)`	Beeps the buzzer count times. Each beep is ON for delayMs milliseconds, then OFF for delayMs milliseconds. Frequency is fixed at 2000 Hz.

Data Logger (SD Card)

Function	Description
`uD.dataLoggerBegin()`	Initializes the SD card over SPI (CS=GPIO5). Returns true on success, false if the card is not found.
`uD.createCSV(filename)`	Creates an empty CSV file with the given filename. Prints status to Serial.
`uD.createCSVWithHeader(filename, col1…col6)`	Creates a CSV file with a 6-column header row. Skips creation if the file already exists. Returns true on success.
`uD.appendCSV(filename, timeStamp, sensorDataL, sensorDataM, sensorDataR, motorL, motorR)`	Appends one row of data to the CSV: timestamp, three sensor readings, and two motor values.

Power Management

Function	Description
`uD.getBatteryVoltage()`	Reads the raw ADC value from GPIO35, converts it using a 3.127× voltage divider ratio, and returns the battery voltage as a float (V).

QUICK-START EXAMPLE

```
#include "uD.h"

void setup() {
  uD.begin();
  uD.buz(2, 100);              // startup beep
  uD.colourRGB(GREEN, 128);    // LED green
  uD.enableIRMode();
}

void loop() {
  int L = uD.getIR(LEFT);
  int M = uD.getIR(MIDDLE);
  int R = uD.getIR(RIGHT);

  if (M && !L && !R) {
    uD.drive(FWD, 180);        // go forward
  } else if (!M && L) {
    uD.drive(LFT, 150);        // turn left
  } else if (!M && R) {
    uD.drive(RIT, 150);        // turn right
  } else {
    uD.stop();
  }
}
```

For more details go to our Github Repository
Get your µ-Drive today