UNIT ELECTRONICS

# PULSAR ESP32-C6

**NANO FACTOR**

**SKU: UE0061**

**Product Reference Manual (V1.1)**

## Description

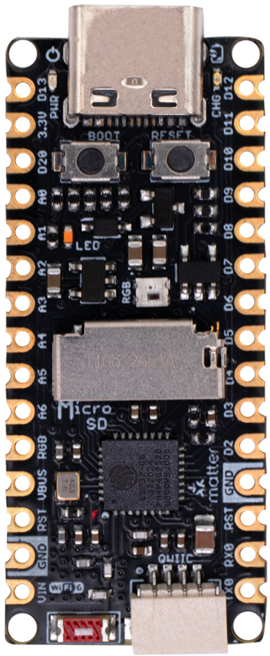
The **PULSAR ESP32 C6** by UNIT Electronics is a compact and powerful development board designed for modern IoT applications. Built around the **ESP32-C6** microcontroller, it supports **Wi-Fi 6, Bluetooth 5, Zigbee, Thread, and Matter**, making it an excellent choice for projects requiring efficient wireless communication.

With its **Arduino Nano-compatible form factor**, this board offers seamless integration with a wide range of existing shields and accessories. It features robust power management, including **USB-C power delivery and LiPo battery support with an integrated charging circuit**. Additionally, its **CAN bus capability** enables reliable communication in industrial and automotive environments.

The **PULSAR ESP32 C6** is optimized for flexibility, supporting development in **C/C++, MicroPython, and ESP-IDF**, making it ideal for both beginners and advanced developers working on **IoT, smart home, sensor networks, and embedded systems**.

### Arduino NANO format compatibility.

This **ultra-compact** board follows the **Arduino Nano** form factor, offering **22 accessible GPIOs** with support for **I2C, SPI, UART, and CAN**. Powered by the **ESP32-C6's RISC-V processor running at 160 MHz**, it ensures **low power consumption and high efficiency** in a



**compact 18mm x 43mm PCB**, making it a **versatile solution** for wireless and embedded applications.

The **PULSAR ESP32 C6** is fully supported in the Arduino development environment, ensuring seamless integration with existing **Arduino projects and libraries**. Developers can easily program the board using the **official Arduino package**, which provides full compatibility with the **ESP32-C6 microcontroller**.

For details on installing the Arduino package and configuring the board, refer to **Section 5.1**.

### Advanced Wireless Communication Protocols

The **PULSAR C6** is equipped with cutting-edge **wireless communication capabilities**, making it a powerful choice for IoT, automation, and embedded applications. The **ESP32-C6** supports multiple wireless protocols, enabling seamless connectivity and high-performance data exchange between devices.

### Wi-Fi 6: High-Efficiency Wireless Networking

The ESP32-C6 integrates **Wi-Fi 6 (802.11ax)**, offering improved performance compared to previous Wi-Fi standards. This next-generation wireless protocol brings key advantages:

- **Higher Network Efficiency** – Supports **OFDMA (Orthogonal Frequency Division Multiple Access)**, allowing multiple devices to share the same channel efficiently, reducing network congestion.

- **Lower Power Consumption** – Implements **Target Wake Time (TWT)**, enabling energy-saving scheduling of transmissions, which is crucial for battery-powered IoT applications.

- **Improved Coverage and Speed** – Operates in the **2.4 GHz band**, ensuring better penetration through walls and objects while maintaining strong data transfer rates.

### Bluetooth 5.0 & BLE: Dual-Mode Connectivity

The **PULSAR C6** includes **Bluetooth 5.0**, providing robust and flexible communication for low-power and high-speed applications. This module supports:

- **Bluetooth Low Energy (BLE)** – Ideal for low-power IoT devices such as sensors, wearables, and smart home applications.

- **Extended Range and Speed** – BLE 5.0 enhances range by up to 4× compared to BLE 4.2, allowing communication over **longer distances** with lower energy consumption.

- **Improved Throughput** – Enables **2 Mbps PHY mode**, doubling data transfer speeds for real-time applications.

### 802.15.4 & Thread: Secure Mesh Networking

The **ESP32-C6** supports **IEEE 802.15.4**, a foundational protocol for mesh networking technologies like **Thread** and **Zigbee**. This capability allows for:

- **Reliable, Low-Power Communication** – Designed for **smart home automation, industrial IoT, and sensor networks**.

- **Scalability** – Enables seamless communication between multiple nodes in a mesh network without reliance on a central router.

**Matter Protocol: Interoperability Across Ecosystems**

The **PULSAR ESP32 C6** is fully compatible with the **Matter protocol**, an industry-unifying standard for **smart home devices**. With Matter, the board can seamlessly integrate with major ecosystems like:

- **Apple HomeKit**
- **Google Home**
- **Amazon Alexa**
- **Samsung SmartThings**

Matter runs over **Wi-Fi 6 and Thread**, enabling **secure, fast, and energy-efficient communication** between smart home devices, ensuring **reliable interoperability across different manufacturers**.

For implementation details, refer to [Espressif's SDK for Matter]

**Optimized Wireless Coexistence**

The **ESP32-C6** features advanced **wireless coexistence mechanisms**, allowing multiple communication protocols to function simultaneously without interference. This enables:

- **Concurrent Wi-Fi & Bluetooth Operations** – Supports seamless operation of **Wi-Fi 6 and BLE 5.0** in parallel, making it ideal for IoT applications that require dual connectivity.
- **Adaptive Frequency Hopping (AFH)** – Minimizes interference between Bluetooth and Wi-Fi, ensuring **stable and reliable communication** in crowded environments.

The **PULSAR ESP32 C6** is a **versatile communication hub**, offering advanced **Wi-Fi 6, Bluetooth 5.0, 802.15.4, and Matter** support. These wireless capabilities enable **scalable, secure, and power-efficient connectivity**, making it ideal for IoT, smart home automation, and industrial applications.

**I2C Connector:**

The **PULSAR ESP32 C6** features a **QWIIC-compatible JST-SH I2C connector** with a **1 mm pitch and 4 pins** (**GND, +V, SDA, and SCL**), allowing easy integration with a wide range of I2C sensors and modules. This standardized **QWIIC interface** simplifies peripheral connections, enabling seamless expansion of IoT and embedded applications without the need for complex wiring.

For detailed pin mappings and configurations, refer to **Table 4.3.1 - Pin Mapping and Connections for QWIIC Connector.**

**Dynamic Visual Feedback:**

To enhance usability and real-time monitoring, the **PULSAR ESP32 C6** includes multiple onboard **LED indicators**, providing clear and immediate feedback for system status and interactions.

These LEDs serve various functions, including:

- **Status Indications** – Power, connectivity, and processing activity.
- **Visual Alerts** – Error notifications and event signaling.
- **User Interaction** – Customizable visual feedback for engaging applications.

The onboard LEDs include:

- **PWR LED (Blue, 0603)** – Indicates that the board is powered on.
- **CHG LED (Orange, 0603)** – Shows battery charging status: **On while charging, blinks when no battery is detected, and turns off when fully charged**.
- **BLINK-IN LED (Pink, 0603, D13)** – Can be used for general debugging or status feedback.
- **WS2812-2020 RGB LED** – Fully addressable for dynamic color indications, ideal for **status monitoring and interactive feedback**.

For more details, refer to:

- **Section 3.2 - Board Topology**
- **Section 3.7 - LED indicators**

**Micro SD:**

The **PULSAR ESP32 C6** includes a **Micro SD card slot (47309-2651)** for expanded storage capabilities. This **interface** is directly connected to the **ESP32-C6** via **SPI**, enabling **faster data access**.

This feature is particularly useful for applications that require:

- **Data Logging** – Storing sensor readings for IoT and industrial applications.
- **Multimedia Handling** – Managing images, audio, or configuration files.
- **Firmware Updates** – Local storage of update files for embedded systems.

For detailed pin mappings and installation guidelines, refer to **Table 4.4 - MicroSD Pin Assignments and SPI Mapping**.

**Power Supply:**

The **PULSAR ESP32 C6** is designed with an **efficient power management system** to ensure stable operation for various applications. It features a AP2112K **LDO regulator**, allowing it to accept a max 6V of input voltages while providing a **stable 3.3V output** for the ESP32-C6 microcontroller and connected peripherals. This power system enables **low power consumption**, making it ideal for IoT, embedded systems, and battery-powered projects.

For more details on power distribution and consumption, refer to **3.8 AP2112K and MCP73831 Power Management System** and **4.5 Battery connections.**

**CAN Bus Connectivity:**

The **PULSAR ESP32 C6** includes native **CAN bus support**, enabling robust **real-time communication** for industrial, automotive, and automation applications. The **ESP32-C6 integrates two built-in TWAI controllers, compatible with ISO 11898-1 (CAN Specifications 2.0)**, ensuring compatibility with industry-standard CAN transceivers for reliable data transmission over long distances.

Key features of the **CAN bus interface**:

- **Supports high-speed data rates** up to **1 Mbps**.
- **3.3V logic level compatibility** for seamless integration with modern embedded systems.
- **Reliable and noise-resistant** communication, essential for automotive and industrial applications.

For detailed, refer to **Section 3.5 - CAN Bus (Two TWAI Controllers Accessible via GPIO Multiplexing).**

**Development & IDE Support:**

The **PULSAR ESP32 C6** supports multiple development environments, offering flexibility for both **beginner and advanced users**:

- **Arduino IDE** – Fully compatible, allowing users to program the board using familiar Arduino libraries. For additional details, refer to **5.1 Getting Started with arduino IDE.**
- **Espressif IDF (ESP-IDF)** – Provides full access to advanced **ESP32-C6** features for optimized performance.
- **MicroPython & CircuitPython** – Enables rapid development for IoT and scripting-based applications using Python.
- **PlatformIO & VS Code** – Ideal for **professional development**, offering advanced debugging and cross-platform support.

This broad **IDE compatibility** ensures that users can **easily transition from Arduino-based development** to more advanced C/C++ and Python programming environments.

**Applications of the PULSAR ESP32 C6**

The **PULSAR ESP32 C6** is a **versatile development board** suitable for a wide range of applications, bridging the gap between beginner-friendly development and professional-grade embedded solutions.

**Key Application Areas**

- **Education & Learning** – Perfect for students, hobbyists, and makers exploring **wireless communication and microcontroller programming**.
- **IoT & Smart Devices** – Built-in **Wi-Fi 6, Thread (802.15.4) and BLE 5.0** enable seamless connectivity for **cloud-based applications, smart home automation, and remote monitoring**.
- **Prototyping & Embedded Systems** – Compact form factor and **extensive peripheral support** make it ideal for **rapid prototyping** and product development.
- **Robotics & Automation** – Integrated **CAN**, **I2C, SPI, and UART** interfaces provide easy **sensor and actuator integration** for robotic control systems.

With **low power consumption, flexible connectivity options, and extensive software support**, the **PULSAR ESP32 C6** empowers developers to create **innovative, real-world solutions** with ease.

## Features

### CPU:

- Espressif ESP32-C6FH4
- Single-core 32-bit RISC-V processor
- Up to 160 MHz operating frequency

### Internal Memory:

- 512 KB of internal SRAM
- 320 KB of ROM (for boot and system functions)
- 4 MB of integrated SPI Flash (in the ESP32-C6FH4 module)
- 1 Kbit EFUSE: stores MAC addresses, module configuration, flash encryption, chip ID

### Wireless Connectivity:

- Wi-Fi 6 (802.11 b/g/n/ax) in the 2.4 GHz band, up to 150 Mbps
- Bluetooth® 5.0 LE, supports LE 2M, LE Coded PHY, Extended Advertising, and Advertising Extensions
- IEEE 802.15.4 for Zigbee and Thread, with support for Matter over Thread and WiFi

### Peripheral Interfaces:

- 12-bit ADC
- SPI, UART, I2C, I2S, PWM
- USB 2.0 Full-Speed (with integrated PHY)
- SDIO, JTAG, GPIO
- Two TWAI® Controllers (compatible with CAN 2.0)

### Built-in Security:

- Hardware Flash Encryption
- Secure Boot
- RNG (Random Number Generator)
- Hardware Cryptography: AES, SHA, RSA, ECC, HMAC

### Operating Voltage:

- Core at 1.1 V (with internal LDO)
- I/O at 3.3 V

### Ultra-Low Power Consumption:

- Low power modes: **Modem-sleep**, **Light-sleep**, and **Deep-sleep**
- Independent RTC for data and event retention during **Deep-sleep**

**Antenna:**

- Integrated PCB antenna (no external antenna required)

**Storage**

- Integrated **microSD card slot via SPI** for data logging, multimedia storage, and firmware updates. Connected to **GPIO (2), GPIO (6), GPIO (7), and GPIO (19)**

**I/O**

- 22 digital pins and 7 analog pins (compatible with the Arduino NANO pinout). Supporting **ADC, PWM, I2C, SPI, UART, CAN**
- **2 × standard UARTs** (UART0 and UART1)
- **1 × LP UART** (Low-Power UART)
- 1 x SPI
- 1 x standard I2C
- 1 x LP I2C (which have a JST-1.0 mm pitch on-board connectors, compatible with STEMMA QT and QWIIC devices)
- **LED PWM Controller** and **Motor Control PWM**, and most GPIO pins can be assigned to generate PWM signals.
- **CAN bus** via integrated **TWAI®** controller **(ISO11898-1 compliant)**

**Power Management:**

- **Vin:** Up to 6V via pin header
- **USB-C powered (5V input)** **VUSB Output:** Available
- **3.3V AP2112K 3.3V LDO Regulator (max input 6V):** 350 mA nominal current, up to 600 mA peak with thermal protection
- **Supports LiPo battery charging** with an onboard power management circuit**. Charging current: 200 mA.**

**Interfaces and Connectors**

- **1 × I2C JST-SH (1.0 mm pitch):** Qwiic-compatible connector wired to GPIO6 and GPIO7 for Low-Power I2C
- **1 × microSD Card Holder**
- **1 × Auxiliary Battery Connector:** Supports both 2.0 mm and 1.25 mm pitch options
- **1 × USB Type-C Connector**
- **2 × 15-pin Header Connectors:** With castellated holes for easy surface mounting

**Communication and Connectivity**

- USB-C connector for programming and power
- Reset button and Flash/Boot button for manually entering flash mode

**LED Indicators:**

- **Blue PWR LED (0603)** – Power indication
- **Orange CHG LED (0603)** – LiPo charging status
- **Pink BLINK LED (0603, GPIO (6))** – User-programmable
- **WS2812 RGB LED (2020)** – Fully addressable for status or visual feedback connected to GPIO (8)

**Software Support:**

- **Arduino IDE** (official [ESP32-C6 board package])
- **ESP-IDF** for advanced native development
- **MicroPython** and **CircuitPython** support
- **PlatformIO / VS Code** for professional development

**Applications:**

- **Smart Home** (Matter, Thread, Wi-Fi 6)
- **Home and Industrial Automation** (including CAN Bus and low-power systems)
- **IoT Prototyping** and **Embedded Development**
- **Multi-radio Devices** and **Mesh Communication**
- **Robotics** and **Sensor Networks**

CONTENTS

## 1 The Board

1.1 Accessories

2x15-pin 2.54mm female headers

## 2 Ratings

### 2.1 Recommended Operating Conditions

| Symbol | Description | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| **VIN** | Output/Input System Voltage (Vsys rail)◆ | VBAT | VBUS -0.3 | 6V | V |
| $I_{sys(IN)}$ | System input current from **VIN** pin (without battery)◆ | - | - | 500 | mA |
| $I_{sys(OUT)}$ | System output current through **VIN** pin◆ | - | - | 500 | mA |
| **VBUS** | USB supply voltage (**VUSB**) | 4.5V | VUSB | 5.5V | V |
| **IBUS** | USB input/output current from VBUS pin (**VUSB**)◆◆ | - | - | 500 | mA |
| **V3v3** | 3.3V output to user application | 3.25 | 3.3 | 3.35 | V |
| **I3v3** | 3v3 output current (AP2112K)◆◆◆ | 300 | 350 | 600 | mA |
| **VIH (ESP32-C6)** | High-level input voltage | 2.475 | - | 3.6 | V |
| **VIL (ESP32-C6)** | Low-level input voltage | -0.3 | - | 0.825 | V |
| **IIH (ESP32-C6)** | High-level input current | - | - | 50 nA | A |
| **IIL (ESP32-C6)** | Low-level input current | - | - | 50 nA | A |
| **VOH (ESP32-C6)** | High - level output voltage* | 2.64 | - | 3.3 | V |
| **VOL (ESP32-C6)** | Low - level output voltage* | - | - | 0.33 | V |
| **IOH Max (ESP32-C6)** | High-level source current at VDD=3.3V, VOH >=2.64V, output set high** | - | 20 | 40 | mA |
| **IOL Max (ESP32-C6)** | Low-level sink current at VDD = 3.3 V, VOL =0.495 V, output set low | - | 28 | - | mA |

| | | | | | |
|---|---|---|---|---|---|
| $I_{output}$ | Cumulative IO output current*** | 500 | - | 1000 | mA |
| **RPU (ESP32-C6)** | Internal weak pull-up resistor | - | 45 | - | kΩ |
| **RPD (ESP32-C6)** | Internal weak pull-down resistor | - | 45 | - | kΩ |
| **TOP (ESP32-C6)** | Operating temperature | -40 | - | 105 | °C |

* VOH and VOL are measured using high-impedance load.
** Except for GPIO12 and GPIO13 whose default drive strength is 40 mA, the default drive strength for all the other pins is 20 mA.
*** Sum of all current being sourced by GPIO. The product proved to be fully functional after all its IO pins were pulled high while being connected to ground for 24 consecutive hours at ambient temperature of 25 °C.

◆ The **VIN** pin is internally connected to the system power bus (**VSYS rail**), which means it can function as both an input and an output power pin. Please consider the following guidelines:

- **When using VIN as a power input** (e.g., when connected from an external shield or power source), **do not exceed 6 V**, or you may damage the **AP2112K** voltage regulator.

- **Do not connect a battery** when supplying power via VIN, as this may stress or damage the battery charging circuit. In such cases, we recommend powering the board through the **VBUS** pin instead, which is designed to handle power input alongside battery charging.

- **When using VIN as a power output**, the voltage available on the pin will correspond to either **VBAT** (from the battery) or **VUSB** (from USB power), depending on the active source. The maximum output current from the VIN pin is **500 mA**.

◆◆ When using the VBUS pin (VUSB) as a power input, keep in mind that the system may draw up to **200 mA** for battery charging and approximately **300 mA** for the ESP32-C6 and other integrated components through the AP2112K voltage regulator. The AP2112K supports a maximum output of **600 mA** at 3.3 V, which limits the total current available from the USB input.

If your application requires sourcing current through the VBUS pin as an output (e.g., to power external devices), ensure that the total current drawn does not exceed **500 mA**. This recommendation considers standard USB Type-C power supplies rated at **1 A**, where **500 mA** may be used for the internal system and **500 mA** for external loads.

◆◆◆ When operating at **600mA**, the AP2112K regulator requires the implementation of **heat dissipation** or **proper cooling** to ensure safe operation.
- If the component temperature reaches **110°C**, thermal protection will activate, halting operation to prevent damage.
- To maintain optimal performance, it is recommended to add a heatsink or enhance airflow.

## 3 Functional Overview

The **PULSAR ESP32-C6** is an advanced development board engineered by UNIT Electronics, delivering next-generation wireless capabilities in an ultra-compact Arduino Nano-compatible form factor. Built around the ESP32-C6 microcontroller, this board offers powerful connectivity with support for **Wi-Fi 6**, **Bluetooth 5**, **Zigbee**, **Thread**, and **Matter**, making it a premier solution for modern IoT, smart home, and industrial automation projects.

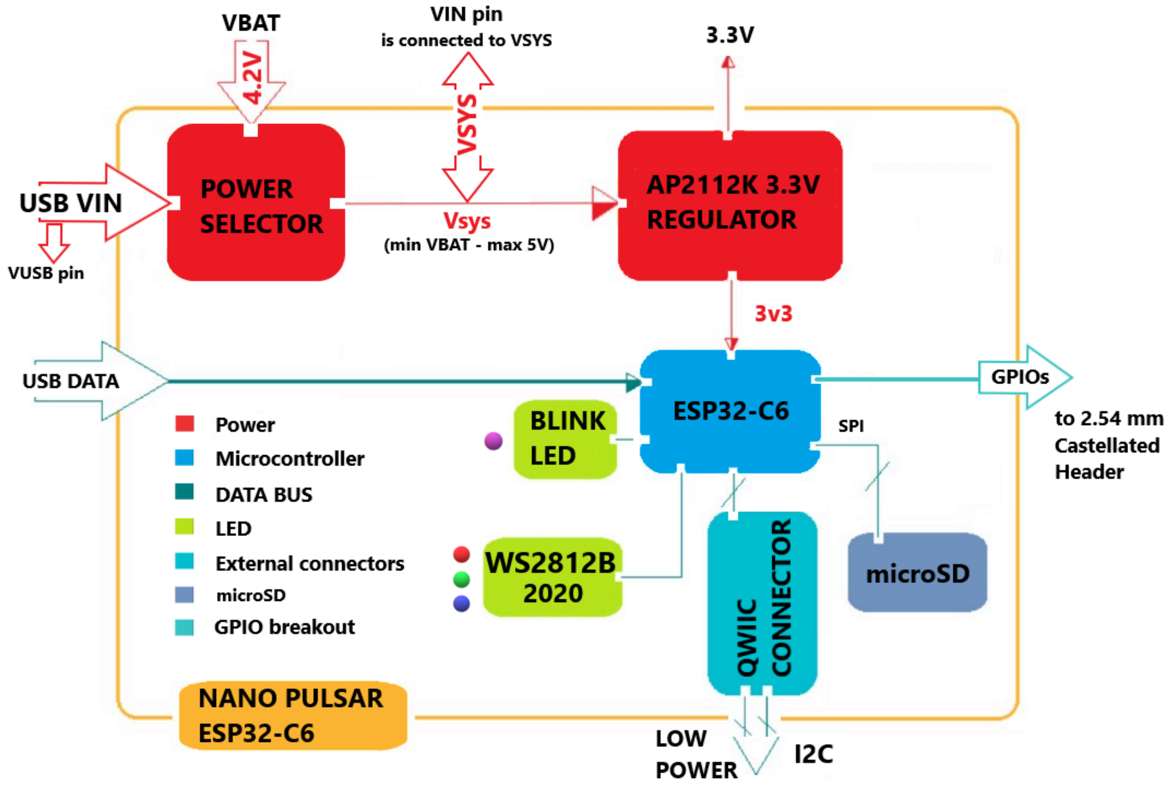
Key features include **native CAN bus support** for robust communication in automotive and industrial environments, **QWIIC-compatible I2C connector** for sensor expansion, and **USB Type-C with LiPo battery charging** for flexible and reliable power management. The board integrates visual feedback via **multiple LED indicators** including a WS2812 RGB LED, enhancing usability and system interaction. Additionally, a **Micro SD slot** supports extended storage for data logging and multimedia handling.

Despite its small 18mm x 43mm footprint, the PULSAR C6 offers **22 accessible GPIOs** with support for I2C, SPI, UART, PWM, ADC, and CAN interfaces, providing exceptional versatility for embedded and connected applications. Powered by a 160 MHz RISC-V core, it combines energy efficiency with high performance, ideal for edge computing and battery-powered systems.

The board is fully compatible with a wide range of programming environments including the **Arduino IDE**, **ESP-IDF**, **MicroPython**, **CircuitPython**, and **PlatformIO**, offering developers maximum flexibility and ease of integration. With full support for Matter and mesh networking technologies like Thread, the PULSAR C6 is future-ready and interoperable across major smart home ecosystems such as **Apple HomeKit**, **Google Home**, **Amazon Alexa**, and **Samsung SmartThings**.

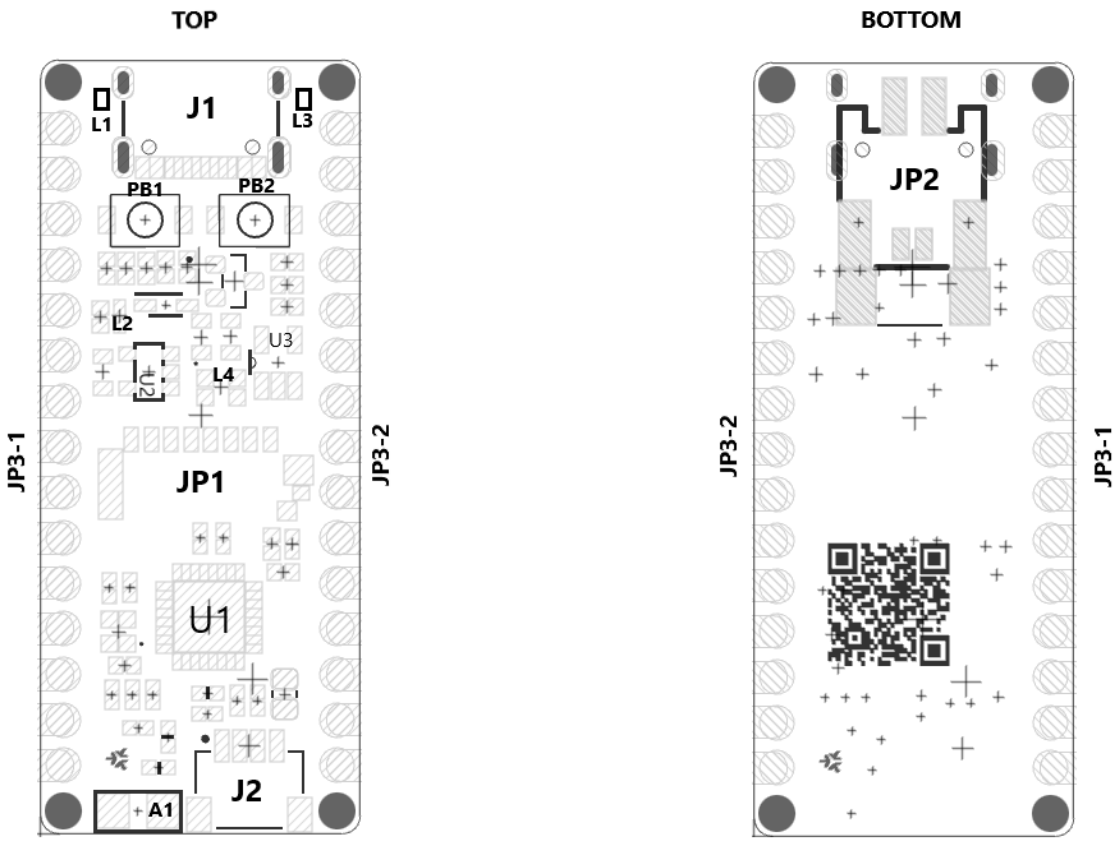
Thanks to its compact form factor, advanced connectivity, and extensive development support, the **PULSAR ESP32 C6** is the ideal platform for wireless prototyping, automation, education, and IoT innovation — serving both beginner and professional developers.

## 3.1 Block Diagram




Block Diagram of PULSAR ESP32-C6

## 3.2 Board Topology




Views of Pulsar ESP32-C6 Topology

Table 3.2.1 - Components Overview

| Ref. | Description |
|---|---|
| U1 | Espressif ESP32-C6FH4 |
| U2 | AP2112K 3.3V LDO Voltage Regulator |
| U3 | MCP73831 Battery Charge Management IC |
| A1 | SMD Antenna |
| L1 | Power On LED |
| L2 | Built-in LED (GPIO 6 or D13) |
| L3 | Charge On LED |

| | |
|---|---|
| L4 | WS2812B-2020 LED |
| J1 | Male USB Type-C Connector |
| J2 | Low-Power I2C-QWIIC JST Connector |
| PB1 | ESP32 Flash Button |
| PB2 | ESP32 Reset Button |
| JP1 | Micro SD Card Socket |
| JP2 | 2.0 mm or 1.25 mm Pitch Battery Connector |
| JP3 | JP3-1, JP3-2: Female Castelled Headers 2.54 mm, compatible with Arduino Nano Pinout |

### 3.3 Processor

The PULSAR ESP32-C6 board is powered by the **Espressif ESP32-C6FH4** microcontroller (U1), a powerful and modern SoC designed for wireless and secure embedded applications. Built on a 40 nm process node, the ESP32-C6 is part of Espressif's next-generation lineup, bringing together Wi-Fi 6, Bluetooth® 5, and robust processing capabilities in a single chip.

**Key features include:**

- **CPU**: 32-bit **RISC-V single-core processor** operating at up to **160 MHz**, optimized for low power consumption and efficient embedded processing.
- **Memory**:
  - **512 KB** of on-chip SRAM.
  - **4 MB Flash memory** embedded in the same package (FH4 variant).
- **Security**:
  - Integrated **hardware crypto accelerators** (AES, SHA, RSA, HMAC).
  - **Secure boot** and **Flash encryption** support.

- **Connectivity**:
  - Integrated **Wi-Fi 6 (802.11ax)** supporting 20 MHz bandwidth in 2.4 GHz.
  - Integrated **Bluetooth® 5.0 LE**, compliant with BLE Mesh.
- **Low Power**: Multiple power domains with advanced sleep modes for battery-powered applications.
- **USB Interface**: Native **USB 2.0 Full-Speed device** interface available through the **USB Type-C (J1)** connector. Capable of up to 12 Mbit/s transfer speed (Note that this controller does not support the faster 480 Mbit/s high-speed transfer mode.
- **GPIO and I/O**: Multiplexed GPIOs capable of digital I/O, PWM, ADC, UART, TWAI, I2C, and SPI.

The ESP32-C6FH4 microcontroller combines performance, wireless connectivity, and integrated security features, making it ideal for IoT applications, secure edge devices, and advanced wireless projects.

3.4 Wireless Connectivity: Wi-Fi 6, Bluetooth LE 5, and IEEE 802.15.4

The **ESP32-C6** provides an **extensive suite of wireless communication capabilities**, making it a powerful and flexible choice for a wide range of **IoT and embedded applications**. It supports **Wi-Fi 6 (802.11ax)** in **2.4 GHz**, **Bluetooth® Low Energy (LE)**, and **IEEE 802.15.4**, enabling compatibility with modern wireless standards and protocols used in **smart home**, **industrial**, and **wearable devices**.

The **Wi-Fi subsystem** includes full support for **TCP/IP networking**, **ESP-WIFI-MESH**, and other protocols. It enables **secure connections using TLS 1.0, 1.1, and 1.2**, and supports both **Station** and **SoftAP** modes. This allows devices to operate as **clients, access points, or mesh nodes** in complex network topologies. It supports up to **150 Mbps (802.11n)**, **MU-MIMO**, **OFDMA**, **beamforming**, **antenna diversity with external RF switch**, and enhanced MAC features like **multiple virtual interfaces**, **QoS (WMM)**, **A-MPDU/A-MSDU**, and **advanced security protocols** such as **WPA3** and **WAPI**. These features ensure **robust, efficient, and secure wireless communication** in **high-density environments**.

The **Bluetooth LE** capabilities of the ESP32-C6 are equally robust, integrating a **hardware link controller**, **RF/modem**, and a **feature-rich software stack**. It supports core features of **Bluetooth 5**, including:

• **1 Mbps and 2 Mbps PHY** for higher throughput
• **Coded PHY (125 Kbps / 500 Kbps)** for extended range
• **Hardware Listen Before Talk (LBT)**
• **LE Advertising Extensions** and **Multiple Advertisement Sets**

• **Simultaneous advertising and scanning**
• **Multiple simultaneous connections** in **central and peripheral roles**
• **LE Data Length Extension** and **LE Power Control**
• **LE Privacy 1.2**, **LE Ping**, and **Link Layer Encryption**

These features enable the ESP32-C6 to support **sophisticated BLE topologies**, such as **device beacons**, **mesh networking**, and **secure multi-node systems**.

In addition to Wi-Fi and BLE, the **ESP32-C6** supports the **IEEE 802.15.4 protocol**, which underpins connectivity for **Thread**, **Zigbee**, **Matter**, **MQTT**, and other **low-power mesh networking standards**. Its **PHY layer** supports:

• **O-QPSK modulation** in the 2.4 GHz band
• **250 Kbps data rate**
• **RSSI and LQI** for signal quality measurement

And the **MAC layer** includes features such as:
• **CSMA/CA** for channel access
• **Active scan** and **energy detection**
• **Hardware frame filtering** and **auto acknowledgment**
• **Auto frame pending** and **Coordinated Sampled Listening (CSL)**

These capabilities make the **ESP32-C6** an ideal candidate for **secure, low-power, and reliable communication** in **constrained and dense wireless environments**. It brings together **three major wireless technologies** into a **single chip—Wi-Fi**, **BLE**, and **802.15.4**—offering **unparalleled flexibility** for developers working on **next-generation connected devices**.

3.5 CAN Bus (Two TWAI Controllers Accessible via GPIO Multiplexing)

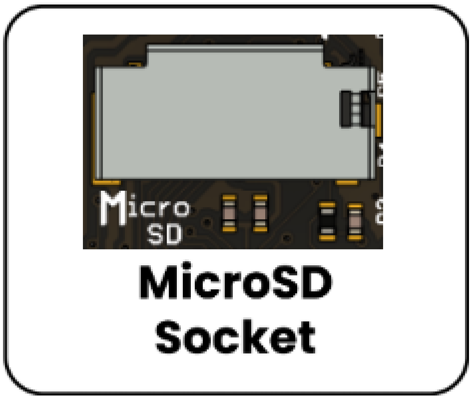
The **PULSAR ESP32-C6** features two **TWAI** (Two-wire Automotive Interface) controllers, enabling the use of the **CAN protocol** in automotive and industrial applications. These controllers can operate in multiple modes, including **Normal**, **Listen-only**, and **Self-test**, and are compatible with the **ISO 11898-1 protocol (CAN 2.0)**. They support data rates up to **1 Mbps** and offer advanced features such as error detection, acceptance filters, and special message transmissions like **Single-shot** and **Self Reception**.

A notable feature of the ESP32-C6 is its ability to connect to a **TWAI bus** via an external **transceiver**, such as the **TCAN1051HVD** or **SN65HVD230**, which can be connected to **GPIO6** and **GPIO7** through the dedicated JST-SH 1 mm QWIIC connector for **low-power I2C (J2)** and multiplexing one of the two TWAI controllers (refer to **Section 4.3 -** QWIIC Connector). Alternatively, the **2.54mm castellated headers (JP3-1 and JP3-2)** can be used to connect the second TWAI controller, expanding communication capabilities for bus-based applications.

For more details on pin assignments and using TWAI, refer to the [ESP32-C6 Technical Reference Manual](), specifically **Chapter Two-wire Automotive Interface** and **Section** [2.3.5 Peripheral Pin Assignment]().

## 3.6 MicroSD Card Integration Overview

The **PULSAR ESP32-C6** is equipped with a **47309-2651 MicroSD card socket**, allowing seamless integration of external storage via **SPI communication**. This is essential for systems where internal flash is insufficient, offering **reliable and scalable storage** for complex data structures or frequent read/write operations. The interface supports **microSD cards up to 64 GB** (validated in field tests), with optimal performance when formatted in **FAT32**. Typical applications include **persistent data logging**, **storage of multimedia content**, and **offline firmware management**. The MicroSD socket is directly connected to the **SPI bus of the ESP32-C6**, ensuring **low-latency access** and broad compatibility with ESP-IDF and Arduino libraries. For hardware-level details, refer to **Table 4.4 - MicroSD Pin Assignments and SPI Mapping**.




## 3.7 LED Indicators

The **WS2812-2020 RGB LED** on the PULSAR ESP32-C6 is a fully addressable **NeoPixel** capable of generating millions of color combinations. It is ideal for **real-time visual feedback**, such as system states, error conditions, or interactive effects. Connected via **GPIO8**, this compact LED can be controlled through popular libraries like **Adafruit NeoPixel** or **FastLED**, making it highly adaptable for **custom UI indicators** in embedded projects.

**Power, Charge, and Built-in LEDs:**

The board integrates three discrete status LEDs for essential system feedback. The **PWR LED** (red, 0603) lights up when the board is powered. The **CHG LED** (orange, 0603) provides **battery charging status**, turning **on while charging**, **blinking** when no battery is detected, and **off when fully charged**. The **BLINK-IN LED** (blue, 0603, connected to GPIO 6) serves as a **built-in user LED**, useful for **debugging or custom activity indicators**. These LEDs ensure quick visual diagnostics without the need for external equipment.

## 3.8 AP2112K and MCP73831 Power Management System

The **AP2112K** low-dropout (LDO) regulator and **MCP73831** battery management IC provide a **stable and efficient power delivery system** for the PULSAR ESP32-C6. The board accepts input voltages of **up to 6V via Vin or USB-C**, and delivers a **regulated 3.3V output** with a nominal current of **350 mA** and peak support up to **600 mA**, protected by internal thermal shutdown.

The **MCP73831** enables safe and reliable **LiPo battery charging** at **200 mA**, making the board ideal for **portable, battery-powered IoT applications**. With onboard voltage regulation, USB power management, and battery integration, the system ensures consistent and **low-noise power** for all core and peripheral operations.

3.9 Power Tree

The **PULSAR ESP32-C6** supports dual power input sources: it can be powered via the **USB-C connector (VBUS, 5V)** or through the **VBAT connetor**, which typically receives power from a **single-cell LiPo battery (3.7V - 4.2V nominal)**.

Power source selection is managed by the **FDN338P P-MOSFET**, which seamlessly switches between VBUS and VBAT. When USB power is present, it is prioritized; otherwise, the system draws power from the battery.

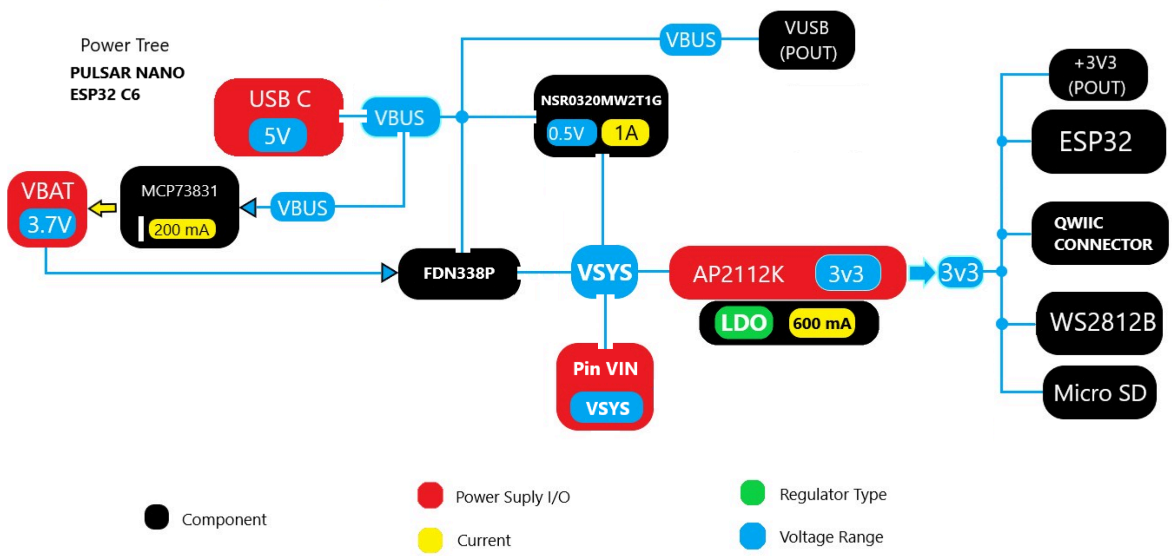
Battery charging is handled by the **MCP73831** linear charger, which supports charging currents up to **200 mA**, enabling safe LiPo charging when USB is connected.

VBUS is further protected and regulated by the **NSR0320MW2T1G Schottky diode**, ensuring low forward voltage drop (0.5V) and supporting up to **1A of current**.

The selected supply voltage (VBUS or VBAT) is routed to the **VSYS rail**, which serves as the main intermediate supply line. This VSYS voltage is then regulated down to **3.3V** by the **AP2112K Low Dropout (LDO) Regulator**, capable of supplying up to **600 mA** of current.

The regulated **3.3V rail** powers the **ESP32-C6 microcontroller**, along with all onboard peripherals including the **Qwiic connector**, **WS2812B LED**, and **Micro SD slot**, ensuring stable operation throughout the board.

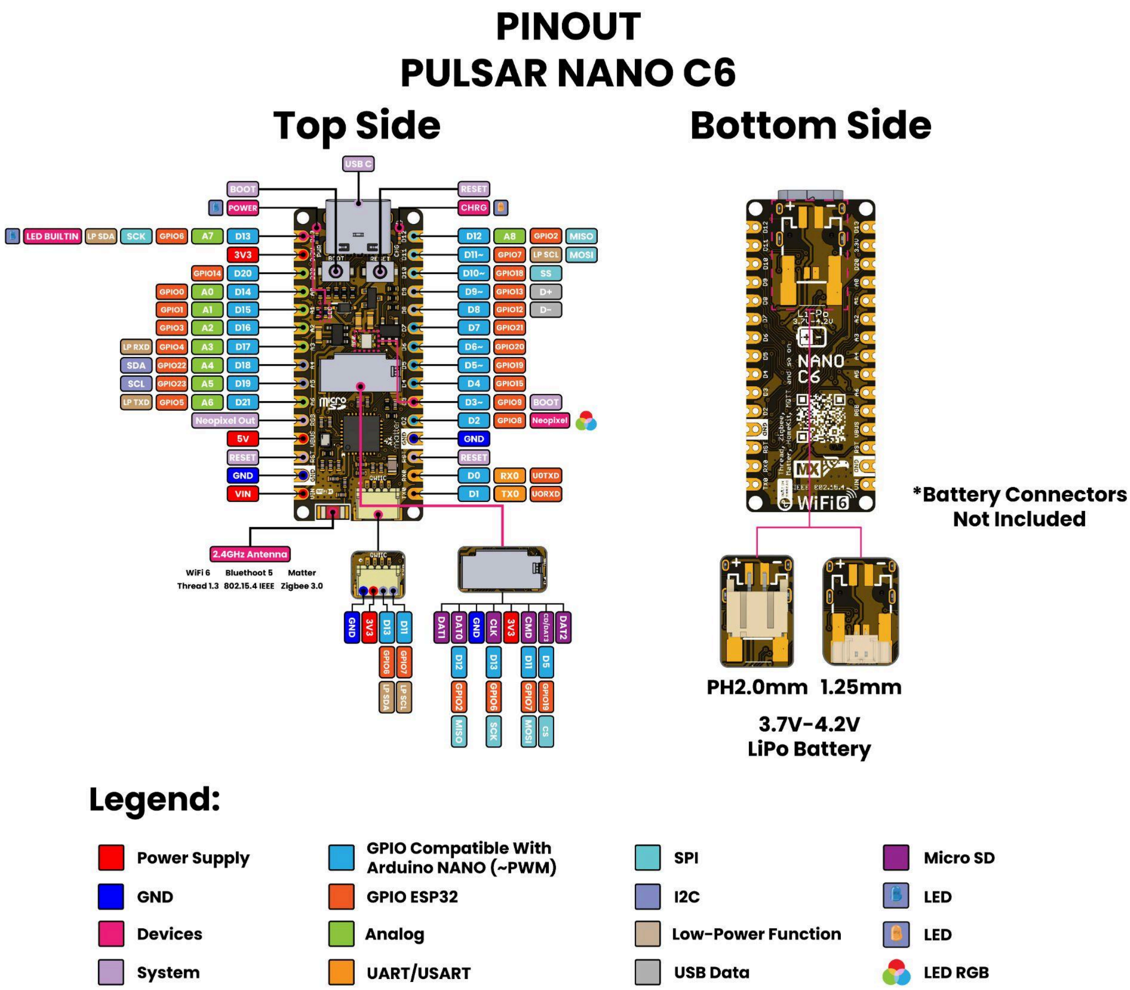
This power architecture offers flexible, efficient, and safe power management for portable and USB-powered applications.




PULSAR ESP32 C6 Power Tree

## 4 Connectors & Pinouts

### 4.1 General Pinout




*PULSAR ESP32-C6 General Pinout*

## 4.2 PULSAR C6: Arduino NANO Pinout Compatibility

Table 4.2.1 - Pin Mapping and Connections for the Arduino NANO-Compatible Pinout

| Arduino Nano Pin | Arduino NANO (Function) | PULSAR C6 (Function) | ESP32 C6 GPIO | GPIO Function |
|---|---|---|---|---|
| 1 | **D13 (SCK/LED)** | D13/SCK/A7 | GPIO6/A6/(SCK) | ADC1_CH6/LP_I2C_SDA/LP_GPIO6/MTCK/FSPICLK |
| 2 | **3.3V** | 3.3V | 3.3V | 3.3V |
| 3 | **AREF** | - | GPIO14 | GPIO14 |
| 4 | **A0 (Analog)/D14** | A0 /D14 | GPIO0 | ADC1_CH0/LP_UART_DTRN/LP_GPIO0/XTAL_32K_P |
| 5 | **A1 (Analog)/D15** | A1/D15 | GPIO1 | ADC1_CH1/LP_UART_DSRN/LP_GPIO1/XTAL_32K_N |
| 6 | **A2 (Analog)/D16** | A2/D16 | GPIO3 | ADC1_CH3/LP_UART_CTSN/LP_GPIO3 |
| 7 | **A3 (Analog)/D17** | A3/D17 | GPIO4 | ADC1_CH4/LP_UART_RXD/LP_GPIO4/MTMS/FSPIHD/SDIO |
| 8 | **A4 (SDA)/D18** | (SDA)/D18 | GPIO22 | SDIO_DATA2 |
| 9 | **A5 (SCL)/D19** | (SCL)/D19 | GPIO23 | SDIO_DATA3 |
| 10 | **A6 (Analog)** | A6/D21 | GPIO5 | ADC1_CH5/LP_UART_TXD/LP_GPIO5/MTDI/FSPIWP/SDIO |
| 11 | **A7 (Analog)** | NEOP_DO | NC | WS2812B-2020 OUT (DO) |
| 12 | **5V** | 5V | 5V | 5V |
| 13 | **RESET** | RST | RST | RESET |

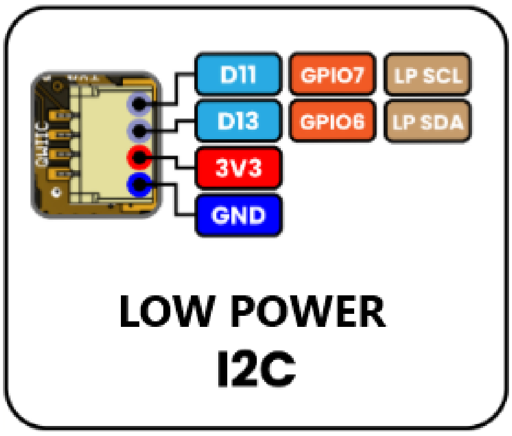
| | | | | |
|---|---|---|---|---|
| 14 | **GND** | GND | GND | GND |
| 15 | **VIN** | VIN | VIN | VIN |
| 16 | **D0 (RX)** | D0/RX | GPIO17 | U0RXD/FSPICS1 |
| 17 | **D1 (TX)** | D1/TX | GPIO16 | U0TXD/FSPICS0 |
| 18 | **RESET** | RST | RST | RESET |
| 19 | **GND** | GND | GND | GND |
| 20 | **D2** | D2/NEOP | GPIO8 | GPIO8/BOOT/ROM |
| 21 | **D3 (PWM)** | D3 | GPIO9 | GPIO9/BOOT |
| 22 | **D4** | D4 | GPIO15 | GPIO15/JTAG |
| 23 | **D5 (PWM)** | D5 | GPIO19 | SDIO_CLK/FSPICS3 |
| 24 | **D6 (PWM)** | D6 | GPIO20 | SDIO_DATA0/FSPICS4 |
| 25 | **D7** | D7 | GPIO21 | SDIO_DATA1/FSPICS5 |
| 26 | **D8** | D8 | GPIO12 | USB_D- |
| 27 | **D9 (PWM)** | D9 | GPIO13 | USB_D+ |
| 28 | **D10 (PWM/SS)** | D10/SS | GPIO18 | SDIO_CMD/FSPICS2 |

| | | | | |
|---|---|---|---|---|
| 29 | **D11 (PWM/MOSI)** | D11/MOSI | GPIO7/(MOSI) | LP_I2C_SCL/LP_GPIO7/MTDO/FSPID |
| 30 | **D12 (MISO)** | D12/MISO/A8 | GPIO2/A2/(MISO) | ADC1_CH2/LP_UART_RTSN/LP_GPIO2/FSPIQ |

## 4.3 QWIIC Connector

Table 4.3.1 - Pin Mapping and Connections for QWIIC Connector

| JST Pin | JST Function | Header Connection | ESP32 C6 GPIO | GPIO Function |
|---|---|---|---|---|
| 1 | GND | GND | GND | GND |
| 2 | 3.3V | 3.3V | 3.3V | 3.3V |
| 3 | LP_SDA/ MUX IO | D13 | GPIO6 | LP_SDA/ADC1_CH6/MTCK/PWM |
| 4 | LP_SCL/ MUX IO | D11 | GPIO7 | LP_SCL/MTDO/PWM |

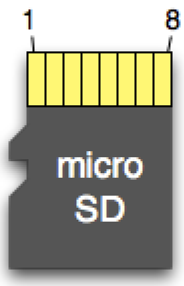



**GPIO6 (D13)** and **GPIO7 (D11)** on the ESP32-C6, accessible via the **QWIIC connector (J2)**, support **functional multiplexing**. While commonly used as **LP_SDA / LP_SCL** for low-power I²C, these pins can also be repurposed for other functions such as **PWM, UART, ADC**, or to enable one of the two onboard **TWAI (CAN Bus) controllers**. This flexibility allows developers to tailor the board's I/O capabilities to fit a wide range of application needs, from serial communication to industrial CAN-based networks.

## 4.4 MicroSD Connector

Table 4.4 - MicroSD Pin Assignments and SPI Mapping.

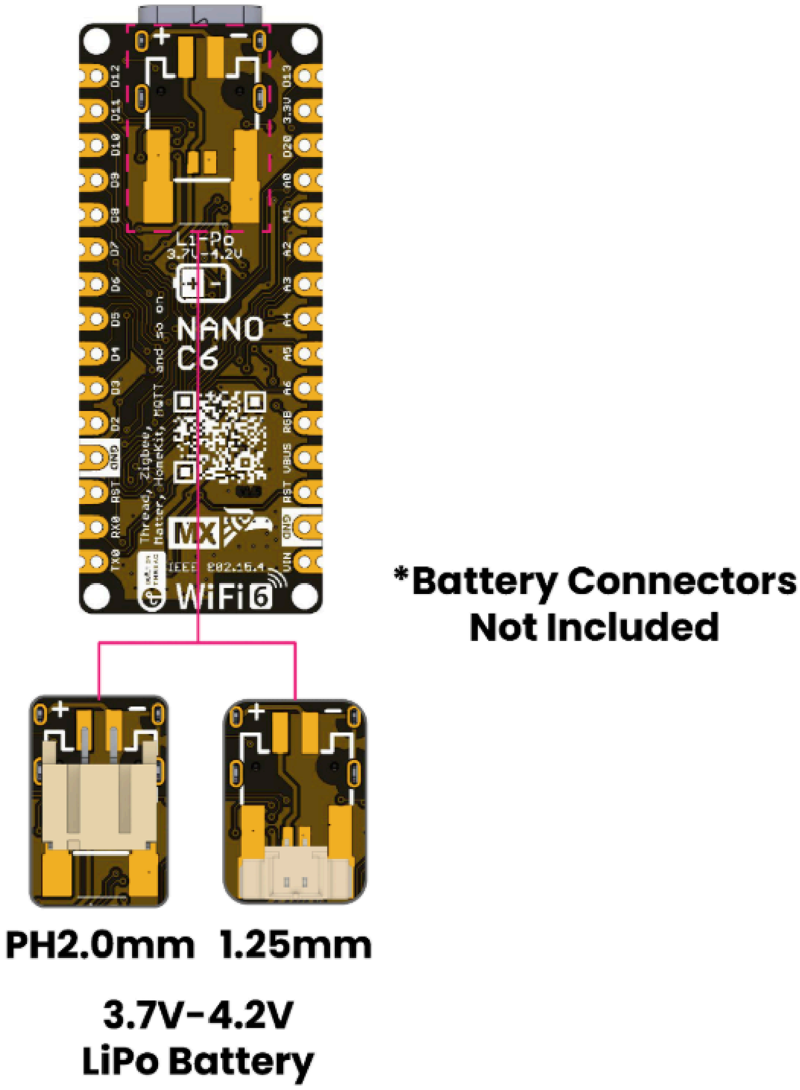
| **microSD Connector Pin** | **microSD Pin Name** | **SPI Function** | **ESP32 C6 GPIO** | Arduino NANO (Function) |
|---|---|---|---|---|
| **1** | **DAT2** | Not used in SPI | - | - |
| **2** | **CD/DAT3** | **CS** (Chip Select) | GPIO19 | D5 |
| **3** | **CMD** | **MOSI** (Master Out Slave In) | GPIO7 | D11/MOSI |
| **4** | **VDD** | **VDD** | 3.3V | - |
| **5** | **CLK** | **SCLK** (Serial Clock) | GPIO6 | D13(SCK/LED) |
| **6** | **VSS** | **GND** | GND | **GND** |
| **7** | **DAT0** | **MISO** (Master In Slave Out) | GPIO2 | D12/MISO |
| **8** | **DAT1** | Not used in SPI | - | - |




| Pin | SD | SPI |
|---|---|---|
| 1 | DAT2 | X |
| 2 | CD/DAT3 | CS |
| 3 | CMD | DI |
| 4 | VDD | VDD |
| 5 | CLK | SCLK |
| 6 | VSS | VSS |
| 7 | DAT0 | DO |
| 8 | DAT1 | X |

***MicroSD Pinout***

4.5 Battery connections




On the back of the PULSAR C6, users can solder either a **PH 2.0 mm** or **JST 1.25 mm** connector to add a **3.7V–4.2V LiPo battery**. These connectors are **not included** with the board but can be purchased separately and soldered based on the user's preference.

The battery charging is managed by the **MCP73831** IC with a **charging current of 200 mA**. Alternatively, the battery can also be soldered directly to the designated pads (+ and –) if no connector is required, providing flexibility for both prototyping and final designs.

> ⚠️ **Important:** Always **double-check the battery connector polarity** before soldering. Incorrect polarity can **damage the onboard Battery Management System (BMS)** and render the board inoperable.

## 5 Board Operation

### 5.1 Getting Started with arduino IDE

To configure the **PULSAR C6** in the Arduino IDE (version 2.0 or higher is recommended ), follow these steps:

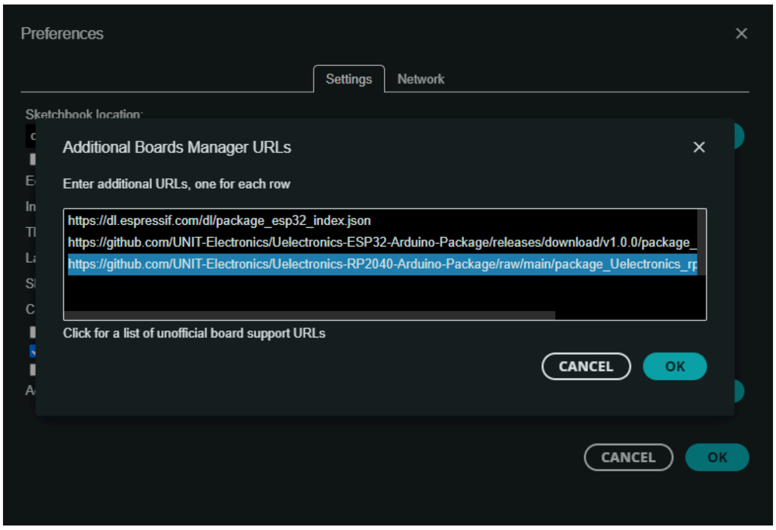
1. **Adding the UNIT Electronics Boards Package:**
   - **Open Preferences**, go to the **File** menu and select **Preferences** (or **Settings** on some systems).

2. **Add Board Manager URLs**:
   - In the **Additional Board Manager URLs** field, add the following links:
     - **ESP32 JSON URL**:

     `https://raw.githubusercontent.com/UNIT-Electronics/Uelectronics-ESP32-Arduino-Package/main/package_Uelectronics_esp32_index.json`

   - Click **OK** to save.

3. **Tip**: If there are multiple URLs to add, separate them with commas.



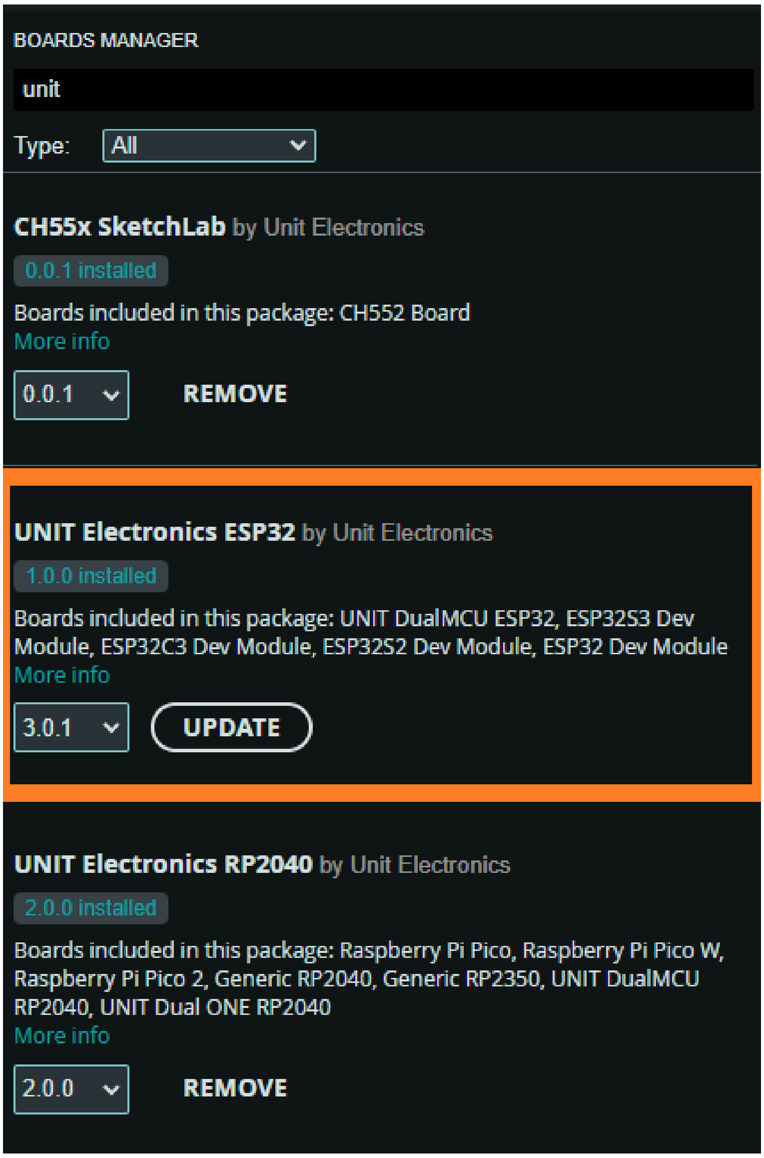
4. **Installing the Board Packages**

- **Open Board Manager:**
  Go to **Tools > Board > Board Manager**.

- **Search for UNIT Electronics Boards:**
  In the search bar, type **UNIT Electronics ESP32**.

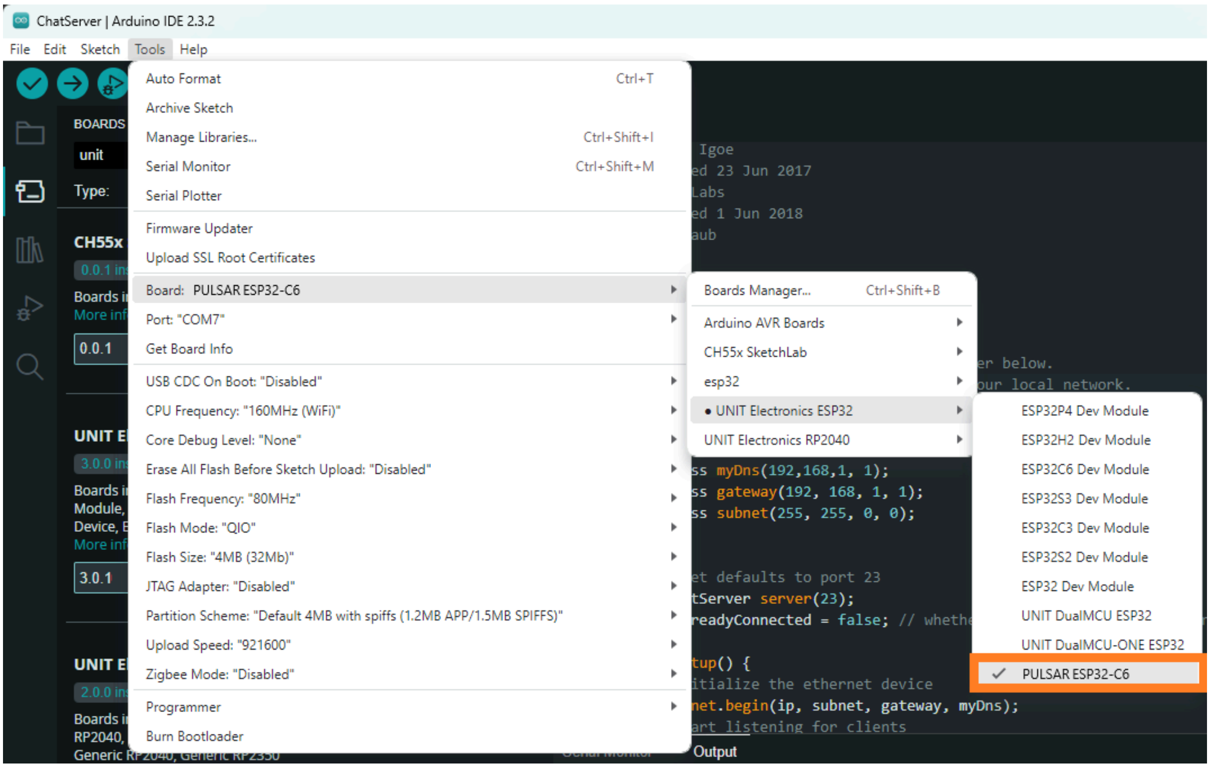
- **Install the Latest Version:**
  Select the latest available version and click **Install**.
  *(The PULSAR C6 board is supported starting from version 3.0.1.)*

**Note:** Make sure you have an active internet connection during installation.



5. **Selecting the Board**

   - **Set Your Board**: Go to **Tools > Board**, and select the **UNIT Electronics ESP32** category. Once inside the selected category, choose the **PULSAR ESP32-C6** board.




6. **Connect the Board**:

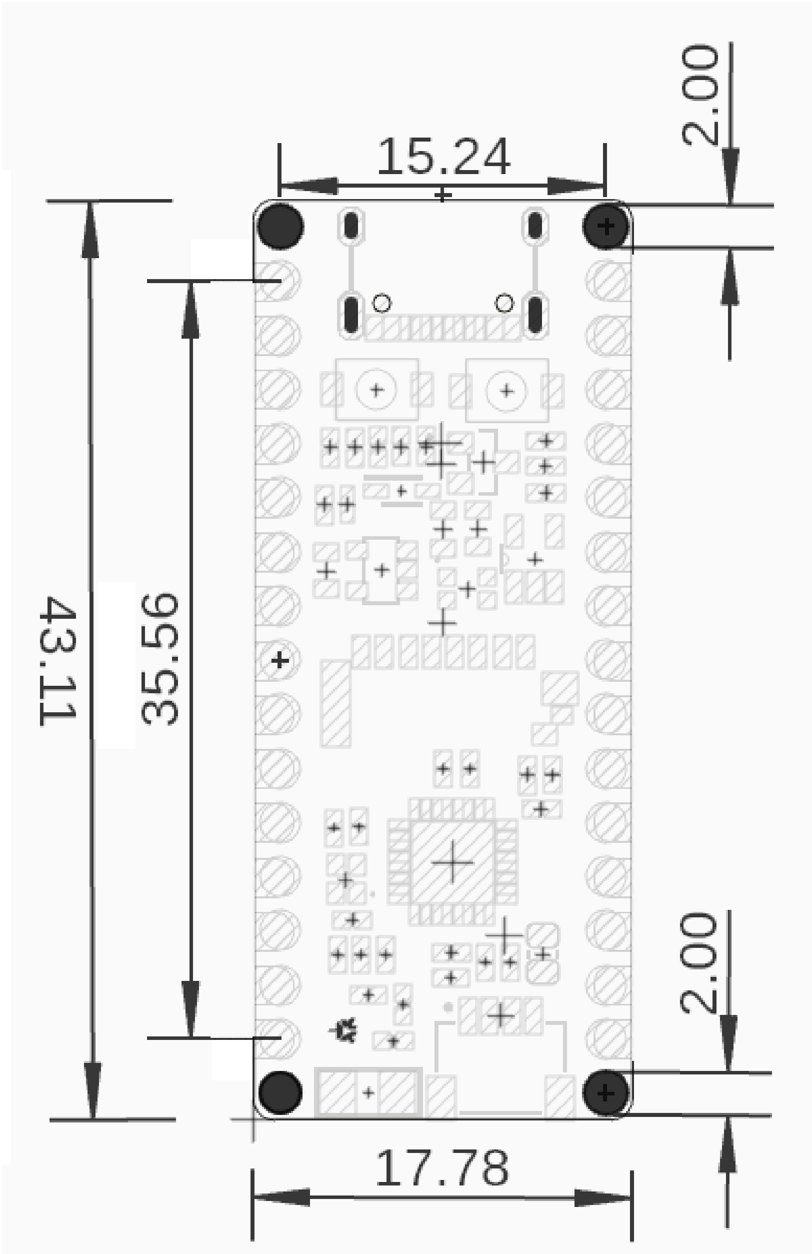
   - Use a USB Type-C cable to connect the PULSAR C6 to your computer. Ensure the correct port is selected under **Tools > Port**.

7. **Getting Started with Examples:**

   - To start programming your **PULSAR C6** using the **Arduino IDE**, go to **File > Examples**. You'll find a wide variety of example sketches tailored for the board—perfect for exploring its capabilities and getting started with your first projects.

*"By completing these steps, you'll be ready to start programming your PULSAR C6 using Arduino IDE".*

## 6 Mechanical Information




Mechanical dimensions in millimeters of PULSAR ESP32-C6

## 7 Company Information

| | |
|---|---|
| **Company name** | UNIT Electronics |
| **Company website** | [https://uelectronics.com/](https://uelectronics.com/) |
| **Company Address** | Salvador 19, Cuauhtémoc, 06000 Mexico City, CDMX |

### 8 Reference Documentation

| Ref | Link |
|---|---|
| PULSAR ESP32-C6 Documentation | https://github.com/UNIT-Electronics-MX/unit_pulsar_esp32_c6 |
| Uelectronics-ESP32 Arduino Package | https://github.com/UNIT-Electronics/Uelectronics-ESP32-Arduino-Package |
| PULSAR C6 Getting Started Guide | https://unit-electronics-mx.github.io/unit_pulsar_esp32_c6/ |
| Thonny IDE | https://thonny.org/ |
| Arduino IDE | https://www.arduino.cc/en/software |
| Visual Studio Code | https://code.visualstudio.com/download |
| ESP32-C6FH4 | https://www.espressif.com/sites/default/files/documentation/esp32-c6_datasheet_en.pdf |

**9 Appendix**

9.1 Schematic (*https://github.com/UNIT-Electronics-MX/unit_nano_esp32_c6/blob/main/hardware/Schematics_UNIT%20NANO_C6_V1_6.pdf*)

# PULSAR ESP32-C6

## NANO FACTOR

**SKU: UE0061**