PULSAR ESP32-H2

NANO FACTOR

SKU: UE0071

Product Reference Manual (V1.1)

Description

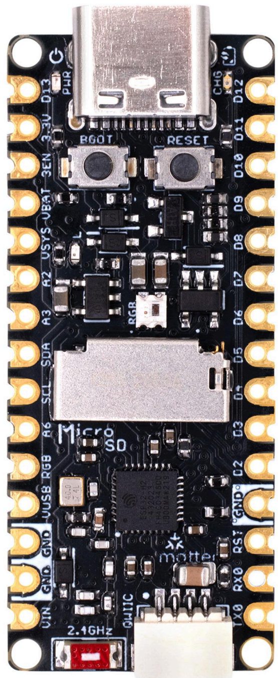
The **PULSAR ESP32 H2** by UNIT Electronics is a compact and powerful development board designed for modern IoT applications. Built around the **ESP32-H2** microcontroller, it supports **Bluetooth 5, Zigbee, Thread, and Matter**, making it an excellent choice for projects requiring efficient wireless 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 H2** 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 **19 accessible GPIOs** with support for **I2C, SPI, UART, and CAN**. Powered by the **ESP32-H2's RISC-V processor running at 96 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 H2** 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-H2 microcontroller**.

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

Advanced Wireless Communication Protocols

The **PULSAR H2** is equipped with cutting-edge **wireless communication capabilities**, making it a powerful choice for IoT, automation, and embedded applications. The **ESP32-H2** supports multiple wireless protocols, enabling seamless connectivity and high-performance data exchange between devices.

Bluetooth 5.3 & BLE: Dual-Mode Connectivity

The **PULSAR H2** features **Bluetooth 5.3 (Certified)**, offering robust and flexible communication for a wide range of applications, from low-power IoT devices to high-speed real-time systems. Key features include:

- **Bluetooth Low Energy (BLE 5.3)** – Optimized for low-power devices such as sensors, wearables, and smart home applications.
- **Extended Range and Long-Range Support** – BLE 5.0 and later with **Coded PHY** enables long-range communication at **125 Kbps** and **500 Kbps**, covering greater distances while minimizing energy consumption.
- **High-Speed Communication** – Supports **2 Mbps PHY mode**, doubling throughput for faster data transfer in real-time applications.
- **Bluetooth Mesh** – Enables large-scale device networks with reliable many-to-many communication.
- **Advertising Extensions & Multiple Advertising Sets** – Allows richer data broadcasting and more flexible device discovery.
- **Simultaneous Roles** – Operates concurrently as **Broadcaster, Observer, Central, and Peripheral**, enabling versatile network configurations.
- **Multiple Connections** – Supports connecting with several devices at once, ideal for complex IoT ecosystems.
- **LE Power Control** – Dynamically adjusts transmission power to optimize energy efficiency and maintain stable connections.

This combination of **long-range capability, high-speed transfer, multi-role operation, and mesh networking** makes the PULSAR H2 a versatile solution for advanced Bluetooth applications.

802.15.4 & Thread: Secure Mesh Networking

The **ESP32-H2** supports **IEEE 802.15.4-2015**, the foundational standard for low-power mesh networking technologies such as **Thread** and **Zigbee 3.0**. This enables reliable, scalable, and energy-efficient communication for smart home, industrial IoT, and sensor network applications.

- **Reliable, Low-Power Communication** – Supports **250 Kbps data rate** in the **2.4 GHz band** using **OQPSK PHY**, optimized for long-lasting battery-powered devices.

- **Mesh Networking Support** – Enables **Thread** and **Zigbee 3.0**, allowing multiple nodes to communicate seamlessly without reliance on a central router.

- **Matter Protocol Compatibility** – Fully supports **Matter**, ensuring interoperability with major smart home ecosystems, including:

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

- **Other Application-Layer Protocols** – Supports additional protocols like **HomeKit, MQTT**, and others, providing flexibility for diverse IoT implementations.

- **Scalability and Security** – Thread-based networks provide secure, fast, and energy-efficient communication across multiple devices, enabling robust and interoperable smart home deployments.

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

Optimized Wireless Coexistence

The **ESP32-H2** features advanced **wireless coexistence mechanisms**, allowing multiple communication protocols to operate simultaneously without interference. This ensures stable and reliable performance in complex IoT environments where **Bluetooth LE**, **Thread**, **Zigbee**, and other protocols may run concurrently.

- **RF Module** – Includes **antenna switches, RF balun, power amplifier,** and **low-noise receive amplifier** for optimized signal quality.

- **High Sensitivity Receivers** – Supports up to **-106.5 dBm sensitivity** for **Bluetooth LE receiver (125 Kbps)** and up to **-102.5 dBm sensitivity** for **802.15.4 receiver (250 Kbps)**, ensuring robust communication even in challenging RF environments.

- **Reliable Multi-Protocol Operation** – Enables simultaneous operation of Bluetooth LE and 802.15.4-based protocols without mutual interference, ideal for smart home, industrial, and sensor network applications.

I2C Connector:

The **PULSAR ESP32 H2** 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 H2** 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 (Red or Green, 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 H2** includes a **Micro SD card slot (47309-2651)** for expanded storage capabilities. This **interface** is directly connected to the **ESP32-H2** 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 H2** 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-H2 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 H2** includes native **CAN bus support**, enabling robust **real-time communication** for industrial, automotive, and automation applications. The **ESP32-H2 integrates a built-in TWAI controller, 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 H2** 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-H2** 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-H2

The **PULSAR ESP32-H2** is a versatile development board designed for a wide range of applications, bridging the gap between beginner-friendly development and professional-grade embedded solutions. With **low power consumption**, **flexible connectivity**, and **robust wireless capabilities**, it enables developers to create innovative and scalable IoT products with ease.

Key Application Areas

- **Education & Learning** – Ideal for students, hobbyists, and makers exploring wireless communication, IoT, and microcontroller programming.
- **IoT & Smart Devices** – Built-in **Bluetooth 5.3**, **Thread (802.15.4)**, and **Zigbee 3.0** enable seamless integration for **cloud-based applications**, **smart home automation**, and **remote monitoring**.
- **Prototyping & Embedded Systems** – Its compact form factor and rich peripheral support make it perfect for **rapid prototyping** and **product development**.
- **Robotics & Automation** – Integrated **CAN**, **I²C**, **SPI**, and **UART** interfaces simplify sensor and actuator integration for **robotic control systems**.
- **Smart Home** – Optimized for **Matter** and **Thread**, enabling interoperability across multiple smart home ecosystems.
- **Industrial Automation** – Suitable for low-latency, reliable control and data collection in industrial environments.
- **Health Care** – Supports wearable and monitoring devices requiring secure, low-power wireless communication.
- **Consumer Electronics** – Ideal for next-generation connected devices and appliances.
- **Smart Agriculture** – Enables efficient environmental monitoring and automation for precision farming.
- **Service Robots** – Provides robust wireless connectivity and low-power operation for autonomous systems.
- **Low-Power IoT Sensor Hubs & Data Loggers** – Excellent choice for distributed sensor networks that demand minimal power consumption and long-term operation.

With **low power consumption**, **flexible connectivity options**, and **extensive software support**, the **PULSAR**

ESP32-H2 empowers developers to design **innovative, real-world IoT solutions** that combine efficiency, scalability, and reliability.

Features

CPU:

- Espressif ESP32-H2FH4S
- Single-core 32-bit RISC-V processor
- Up to 96 MHz operating frequency
- Four-stage pipeline

Internal Memory:

- 320 KB of internal SRAM
- 128 KB of ROM (for boot and system functions)
- 4 MB of integrated SPI Flash (in the ESP32-H2FH4 module)
- 4KB LP Memory
- 16 KB cache

Wireless Connectivity:

- 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:

- 19 programmable GPIOs (including GPIO8, GPIO9, and GPIO25 as strapping pins)
- 12-bit SAR ADC (up to 5 channels)
- Temperature sensor
- SPI, UART, I²C, I²S, PWM, RMT
- USB 2.0 Full-Speed (with integrated Serial/JTAG controller and PHY)
- General-purpose SPI, UART (×2), and I²C (×2) interfaces
- RMT with up to 2 transmit channels and 2 receive channels
- LED PWM controller (up to 6 channels)
- Motor Control PWM (MCPWM)
- Pulse Count Controller
- General DMA controller (3 TX channels, 3 RX channels)
- Parallel I/O (PARLIO) controller
- SoC Event Task Matrix (ETM)
- Two TWAI® Controllers (compatible with ISO 11898-1 / CAN 2.0)
- Two 54-bit general-purpose timers
- 52-bit system timer
- Three watchdog timers
- SDIO, JTAG, GPIO

Built-in Security:

- Secure Boot – Ensures firmware integrity during startup
- Flash Encryption – Provides secure memory encryption and decryption
- 4096-bit OTP (One-Time Programmable memory), with up to 1792 bits available for user data

- Cryptographic Hardware Acceleration:
  AES-128/256 (FIPS PUB 197) – Supports ECB, CBC, CFB, OFB, and CTR modes (FIPS PUB 800-38A)
  SHA Accelerator (FIPS PUB 180-4)
  RSA Accelerator
  ECC Accelerator
  ECDSA (Elliptic Curve Digital Signature Algorithm)
  HMAC (Hash-based Message Authentication Code)
  Digital Signature Engine
- Access Permission Management (APM)
- Random Number Generator (RNG)
- Power Glitch Detector

Power Management and Operating Voltage

- I/O Operating Voltage: **3.3 V**
- Ultra-Low Power Consumption – Designed for energy-efficient applications requiring extended battery life
- Fine-resolution power control through adjustable **clock frequency**, **duty cycle**, **RF operating modes**, and **individual power control** of internal components
- Four Power Modes optimized for different operation scenarios:
  Active, **Modem-sleep**, **Light-sleep**, and **Deep-sleep**
- Power Consumption in Deep-sleep Mode: **7 µA**
- Independent **RTC** (Real-Time Clock) for data and event retention during Deep-sleep mode
- LP (Low-Power) memory remains powered on in Deep-sleep mode

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 (0), GPIO (4), GPIO (5), and GPIO (11)**

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 GPIO12 and GPIO22 for Low-Power I2C
- **1 × microSD Card Holder**
- **1 × Auxiliary Battery Connector (optional):** 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:

- **Green or Red PWR LED (0603)** – Power indication
- **Orange CHG LED (0603)** – LiPo charging status
- **Pink BLINK LED (0603, GPIO (4))** – User-programmable
- **WS2812 RGB LED (2020)** – Fully addressable for status or visual feedback connected to GPIO (8)

Software Support:

- **Arduino IDE** (official Uelectronics-ESP32 Arduino 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◆	-	-	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 VUSB pin (**VUSB**)◆◆	-	-	1000	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-H2)	High-level input voltage	2.475	-	3.6	V
VIL (ESP32-H2)	Low-level input voltage	-0.3	-	0.825	V
IIH (ESP32-H2)	High-level input current	-	-	50 nA	A
IIL (ESP32-H2)	Low-level input current	-	-	50 nA	A
VOH (ESP32-H2)	High - level output voltage*	2.64	-	3.3	V
VOL (ESP32-H2)	Low - level output voltage*	-	-	0.33	V
IOH Max (ESP32-H2)	High-level source current at VDD=3.3V, VOH >=2.64V, output set high	-	40	-	mA
IOL Max (ESP32-H2)	Low-level sink current at VDD = 3.3 V, VOL =0.495 V, output set low	-	28	-	mA
I_{output}	Cumulative IO output current**	-	-	1300	mA

RPU (ESP32-H2)	Internal weak pull-up resistor	-	45	-	kΩ
RPD (ESP32-H2)	Internal weak pull-down resistor	-	45	-	kΩ
TOP (ESP32-H2)	Operating temperature	-40	-	105	°C

* VOH and VOL are measured using a high-impedance load.
** Sum of all current sourced by the GPIO pins. The product proved to be fully functional after all its I/O pins were pulled high while connected to ground for 24 consecutive hours at an ambient temperature of 25 °C.

◆ The **VIN** pin is internally connected to the 5 V power bus through a **NSR0320MW2T1G diode**.
Unlike the PULSAR C6, this pin functions **only as a power input**, not as an output.

The **VUSB** pin is directly connected to the **USB Type-C power line**, allowing the user to connect a load directly to the main USB power supply.

Due to the diodes connected between **VIN** and **VUSB**, both input voltages are **isolated**, preventing current flow between the two power sources.

The **VSYS** pin is connected to the main system power rail, meaning the voltage present here is either the **VBAT** voltage or approximately **5 V**, depending on the power source.

The **VBAT** pin is directly connected to the battery, allowing the user to power external loads that require higher current directly from the battery.

◆◆ When using the **VBUS (VUSB)** pin as a power input, note that the system may draw up to **200 mA** for battery charging and approximately **300 mA** for the **ESP32-H2** 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 available current from the USB input.

If your application requires sourcing current from the **VUSB pin** as an **output** (for example, to power external devices), ensure that the total current drawn does **not exceed 1000 mA**.
This recommendation is based on standard USB Type-C power supplies rated at **1 A**, where approximately **500 mA** may be used by the internal system and **500 mA** allocated to external loads.

◆◆◆ When operating near the **600 mA limit**, the **AP2112K regulator** requires adequate heat dissipation or thermal management to ensure safe operation.

- If the component temperature reaches **110 °C**, thermal protection will activate, stopping operation to prevent damage.
- To maintain optimal performance, it is recommended to **add a heatsink or improve airflow** around the component.

3 Functional Overview

The **PULSAR ESP32-H2** 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-H2 microcontroller, this board offers powerful connectivity with support for **Bluetooth 5.3**, **Zigbee**, **Thread**, and **Matter**, making it a premier solution for modern IoT, smart home, and industrial automation projects.

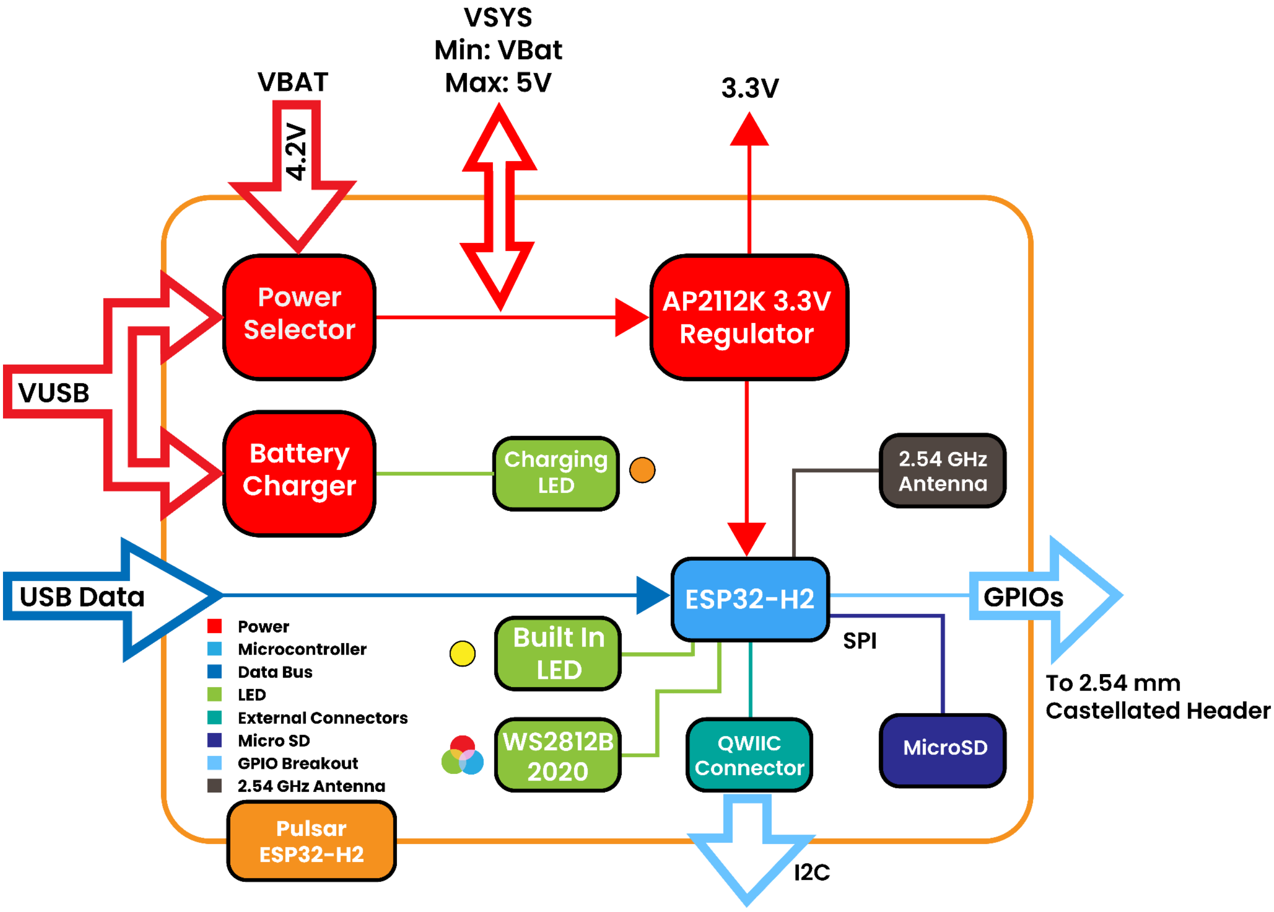
Key features include **native CAN bus support** for robust communication in automotive and industrial environments, **QWIIC-compatible I2C connector** for sensor expansion, and **USB Type-C with LiPo battery charging** for flexible and reliable power management. The board integrates visual feedback via **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 H2 offers **19 accessible GPIOs** with support for I2C, SPI, UART, PWM, ADC, and CAN interfaces, providing exceptional versatility for embedded and connected applications. Powered by a 96 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 H2 is future-ready and interoperable across major smart home ecosystems such as **Apple HomeKit**, **Google Home**, **Amazon Alexa**, and **Samsung SmartThings**.

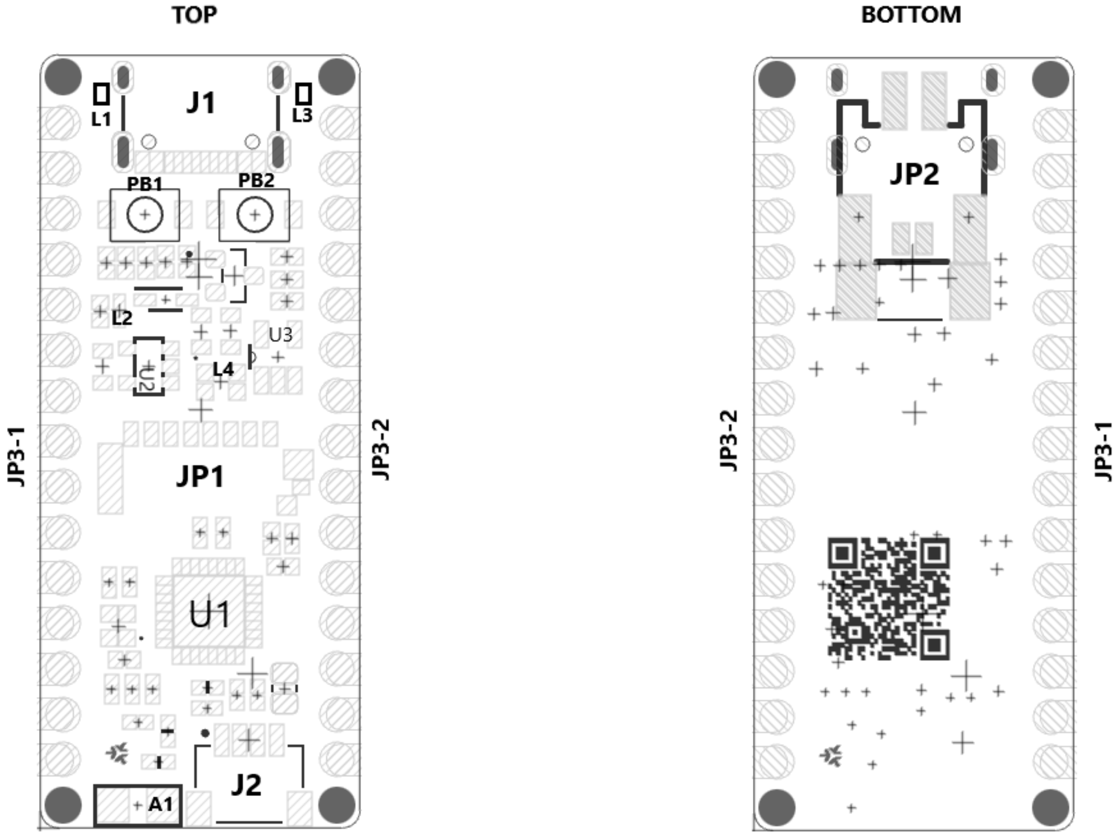
Thanks to its compact form factor, advanced connectivity, and extensive development support, the **PULSAR ESP32 H2** is the ideal platform for wireless prototyping, automation, education, and IoT innovation — serving both beginner and professional developers.

3.1 Block Diagram



Block Diagram of Pulsar ESP32-H2

3.2 Board Topology



Views of Pulsar ESP32-H2 Topology

Table 3.2.1 - Components Overview

Ref.	Description
U1	Espressif ESP32-H2FH4S
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 4 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-H2** board is powered by the **Espressif ESP32-H2FH4S** microcontroller (U1), a highly efficient and secure SoC designed for **low-power wireless and embedded applications**. Fabricated using a 40 nm process, the ESP32-H2 belongs to Espressif's next-generation lineup and integrates **Bluetooth® 5.3 LE** and **IEEE 802.15.4 (Thread/Zigbee)** connectivity, providing a robust platform for IoT and Matter-based devices.

Key features include:

CPU:

- 32-bit **RISC-V single-core processor** operating at up to **96 MHz**, optimized for low-power consumption and reliable real-time performance.

Memory:

- **320 KB of on-chip SRAM** for program and data storage.
- **4 MB of embedded Flash memory** (FH4 variant) within the same package.

Security:

- Integrated hardware cryptographic accelerators supporting **AES**, **SHA**, **RSA**, and **HMAC** algorithms.

- **Secure Boot**, **Flash Encryption**, and **Digital Signature (DS)** support for enhanced firmware protection.

Connectivity:

- **IEEE 802.15.4** radio supporting **Thread** and **Zigbee** mesh protocols.
- **Bluetooth® 5.3 Low Energy**, compliant with BLE Mesh and featuring long-range mode and multiple advertising extensions.

Low Power:

- Multiple power domains and advanced **sleep modes** (light-sleep and deep-sleep) for ultra-low power operation.
- Integrated PMU with fine-grained clock and voltage control for energy optimization in battery-powered applications.

USB Interface:

- Native **USB 2.0 Full-Speed device** interface (up to 12 Mbit/s) available through the **USB Type-C (J1)** connector.
  (Note: The controller does not support 480 Mbit/s high-speed mode.)

GPIO and I/O:

- Multiplexed GPIOs supporting **digital I/O**, **PWM**, **ADC**, **UART**, **SPI**, **I²C**, and **TWAI (CAN bus)** interface.
- ADC with 12-bit resolution and multiple channels for analog signal acquisition.

The **ESP32-H2FH4S** microcontroller combines energy efficiency, secure connectivity, and advanced wireless capabilities, making it ideal for **IoT networks, smart home applications, Matter-compatible devices, and other low-power wireless systems**.

3.4 Wireless Connectivity: Bluetooth LE 5.3 and IEEE 802.15.4

The **ESP32-H2** provides a comprehensive set of wireless communication capabilities, making it an excellent choice for **low-power IoT and embedded applications**. It integrates **Bluetooth® 5.3 Low Energy (LE)** and **IEEE 802.15.4**, enabling compatibility with modern wireless standards and protocols used in **smart home, industrial, and Matter-based devices**.

The **Bluetooth LE 5.3** subsystem includes a **hardware link controller**, RF/modem, and a full-featured software stack. Key features include:

- **1 Mbps and 2 Mbps PHY** for higher throughput.
- **Coded PHY (125 Kbps / 500 Kbps)** for extended range.
- **Hardware Listen Before Talk (LBT)** for coexistence in crowded RF environments.
- **LE Advertising Extensions** and **multiple advertisement sets**.
- **Simultaneous advertising and scanning**.
- **Multiple simultaneous connections** in both central and peripheral roles.
- **LE Data Length Extension** and **LE Power Control**.
- **LE Privacy 1.2**, **LE Ping**, and **Link Layer Encryption**.

These capabilities allow the ESP32-H2 to support **advanced BLE topologies**, including **mesh networks**, device beacons, and secure multi-node systems.

In addition, the **IEEE 802.15.4** radio provides the foundation for **Thread**, **Zigbee**, and **Matter** protocols, as well as other low-power mesh networking standards. Its PHY and MAC features include:

PHY Layer:

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

MAC Layer:

- **CSMA/CA** for fair channel access.
- **Active scan and energy detection**.
- **Hardware frame filtering** and **automatic acknowledgment**.
- **Auto frame pending** and **Coordinated Sampled Listening (CSL)** for power-efficient mesh communication.

By combining **Bluetooth LE 5.3** and **IEEE 802.15.4** in a single chip, the ESP32-H2 provides **secure, low-power, and reliable wireless connectivity** for next-generation IoT devices and constrained network environments, making it a versatile platform for developers building **connected and mesh-enabled applications**.

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

The **PULSAR ESP32-H2** features a **TWAI (Two-Wire Automotive Interface)** controller, enabling communication over the **CAN bus** for automotive, industrial, and distributed embedded systems. The TWAI controller is fully compatible with the **ISO 11898-1 (CAN 2.0)** protocol and supports data rates up to **1 Mbps**, providing robust and deterministic communication in noisy environments.

This controller can operate in multiple modes, including:

- **Normal Mode** – for standard communication.
- **Listen-Only Mode** – for passive network monitoring.
- **Self-Test Mode** – for internal diagnostic and validation purposes.

Advanced features include **error detection and handling**, **acceptance filters**, **automatic retransmission**, and special transmission options such as **Single-Shot** and **Self-Reception**.

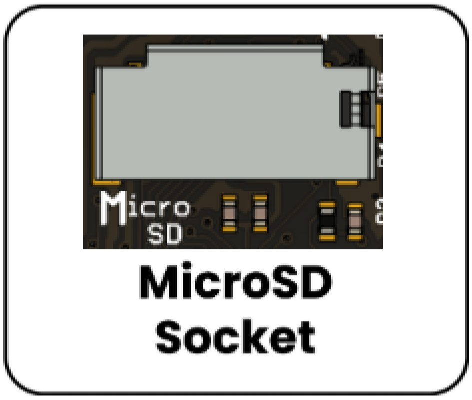
A key design feature of the ESP32-H2 implementation is its ability to interface with an **external CAN transceiver**, such as the **TCAN1051HVD** or **SN65HVD230**, to drive the physical CAN bus lines. The transceiver can be connected to **GPIO12 (TX)** and **GPIO22 (RX)** through the **dedicated JST-SH 1 mm QWIIC connector (J2)** used for low-power I²C (refer to **Section 4.3 – QWIIC Connector**).

Alternatively, the **2.54 mm castellated headers (JP3-1 and JP3-2)** provide another means of accessing the TWAI interface, allowing greater flexibility for **bus-based communication systems** and **external CAN hardware integration**.

For detailed information on **pin assignments** and **TWAI configuration**, refer to the [ESP32-H2 Technical Reference Manual](), specifically the **Two-Wire Automotive Interface** chapter.

3.6 MicroSD Card Integration Overview

The **PULSAR ESP32-H2** 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-H2**, 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-H2 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 or green, 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** (pink, 0603, connected to GPIO 4) 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-H2. 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-H2** supports **three power input sources**: it can be powered via the **USB-C connector** (VBUS, 5 V), the **VIN pin** (up to 6 V), or the **VBAT connector**, which typically receives power from a **single-cell LiPo battery** (nominal 3.7 V – 4.2 V).

Power source selection is automatically managed by the **FDN338P P-channel MOSFET**, which seamlessly switches between the **5 V** and **VBAT** sources. When USB or VIN power is present, it is given priority; otherwise, the system draws power from the battery.

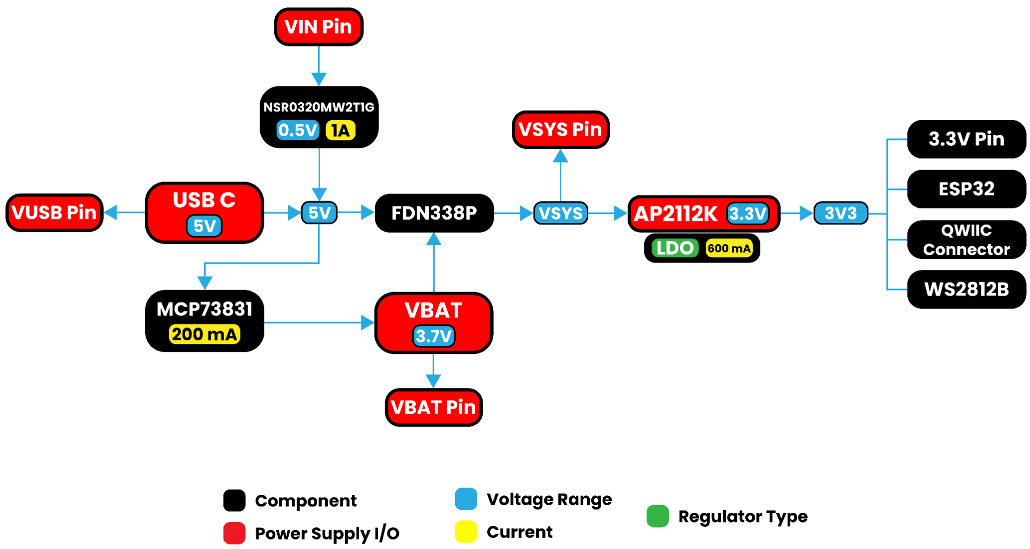
Battery charging is handled by the **MCP73831** linear Li-Ion/LiPo charger, supporting **charging currents up to 200 mA** for safe and efficient battery charging whenever USB power is available.

The **VUSB** and **VIN** inputs are protected and isolated by an **NSR0320MW2T1G Schottky diode**, which provides a low forward voltage drop (typically 0.5 V) and supports up to **1 A** of continuous current.

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

The regulated **3.3 V rail** powers the **ESP32-H2 microcontroller**, as well as all onboard peripherals including the **Qwiic connector**, **WS2812B RGB LED**, and **microSD card slot**, ensuring stable and reliable operation across the entire board.

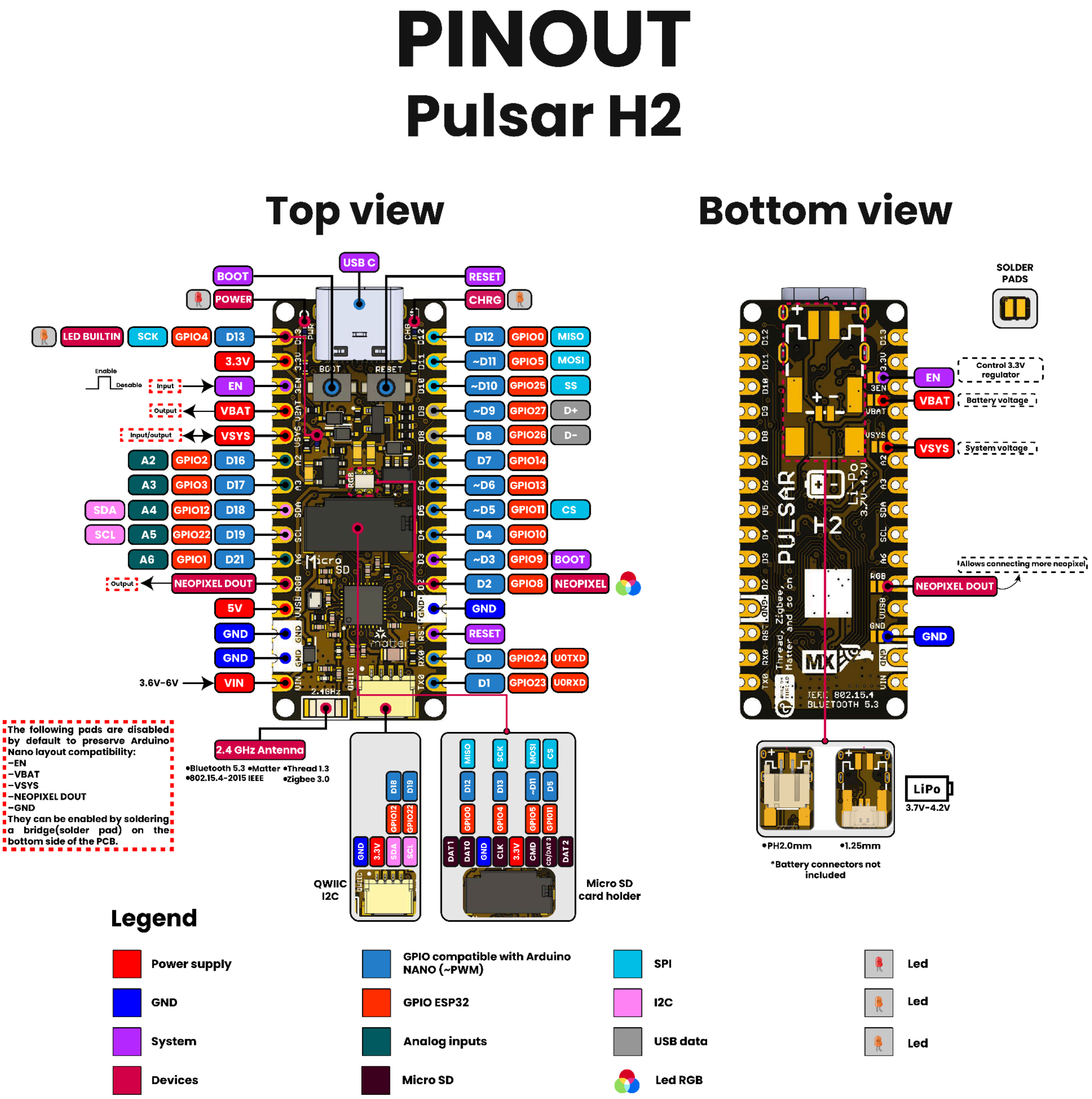
This **power architecture** provides **flexible, efficient, and safe power management**, making the board suitable for both **portable** and **USB-powered** applications.



Pulsar ESP32 H2 Power Tree

4 Connectors & Pinouts

4.1 General Pinout



PULSAR ESP32-H2 General Pinout

4.2 PULSAR H2: 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 H2 (Function)	ESP32 H2 GPIO	GPIO Function
1	D13 (SCK/LED)	D13/SCK/	GPIO4	ADC1_CH3/FSPICLK/MTCK
2	3.3V	3.3V	3.3V	3.3V
3	AREF	EN_3V3	NC	Enable 3V3 Regulator
4	A0 (Analog)/D14	VBAT	NC	Battery IVoltage
5	A1 (Analog)/D15	VSYS	-	-
6	A2 (Analog)/D16	A2/D16	GPIO2	ADC1_CH1/FSPIWP/MTMS
7	A3 (Analog)/D17	A3/D17	GPIO3	ADC1_CH2/FSPIHD/MTMS
8	A4/(SDA)	(SDA)/D18	GPIO12	GPIO12
9	A5/(SCL)	(SCL)/D19	GPIO22	GPIO22
10	A6 (Analog)	A6	GPIO1	ADC1_CH0/FSPICS0
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	GPIO23	U0RXD/FSPICS1

17	D1 (TX)	D1/TX	GPIO24	U0TXD/FSPICS2
18	RESET	RST	RST	RESET
19	GND	GND	GND	GND
20	D2	D2/NEOP	GPIO8	LOG
21	D3 (PWM)	D3	GPIO9	BOOT
22	D4	D4	GPIO10	ZCD0
23	D5 (PWM)	D5	GPIO11	ZDC1
24	D6 (PWM)	D6	GPIO13	XTAL_32K_P
25	D7	D7	GPIO14	XTAL_32K_N
26	D8	D8	GPIO26	USB_D-/FSPICS4
27	D9 (PWM)	D9	GPIO27	USB_D+/FSPICS5
28	D10 (PWM/SS)	D10/SS	GPIO25	FSPICS3
29	D11 (PWM/MOSI)	D11/MOSI/A4	GPIO5	ADC1_CH4/FSPID/MTDI
30	D12 (MISO)	D12/MISO	GPIO0	FSPIQ

Notes on This Pin Mapping

I²C: The pins used for I²C can be freely selected from any GPIO through the internal GPIO matrix, allowing flexible configuration according to application needs.

SPI: Dedicated pins have been assigned for Fast SPI operation to ensure optimal performance and avoid pin conflicts with other peripherals.

GPIO: Digital pins (GPIO0 ~ GPIO5, GPIO22 ~ GPIO27): These pins cannot operate in Deep-sleep mode but may function in Light-sleep mode, provided that the power domain controlled by XPD_TOP remains active.

Low-Power (LP) pins (GPIO8 ~ GPIO14): These pins can operate in any power mode of the chip, including Deep-sleep and Light-sleep.

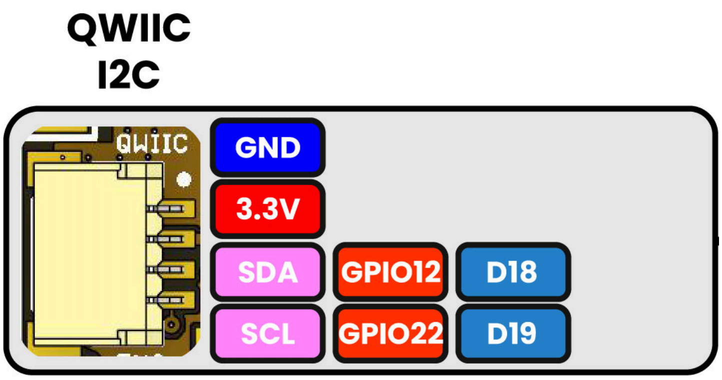
Neopixel: The NEOP_DO dedicated output is reserved to maintain compatibility with WS2812B addressable LEDs across all PULSAR development boards.

ADC: The same analog channels available on the ESP32-H2 (ADC1_CH0 – ADC1_CH4) have been prioritized to maintain compatibility and measurement consistency.

4.3 QWIIC Connector

Table 4.3.1 - Pin Mapping and Connections for QWIIC Connector

JST Pin	JST Function	Header Connection	ESP32 H2 GPIO	GPIO Function
1	GND	GND	GND	GND
2	3.3V	3.3V	3.3V	3.3V
3	LP_SDA/ MUX IO	D18	GPIO12	GPIO12/MUX IO
4	LP_SCL/ MUX IO	D19	GPIO22	GPIO22/MUX IO

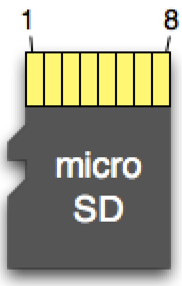


GPIO12 (D18) and **GPIO22 (D19)** on the ESP32-H2, accessible via the **QWIIC connector (J2)**, support **functional multiplexing**. While commonly used as **SDA / SCL** forr I²C, these pins can also be repurposed for other functions such as **PWM, UART**, or to enable the onboard **TWAI (CAN Bus) controllers**. 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.

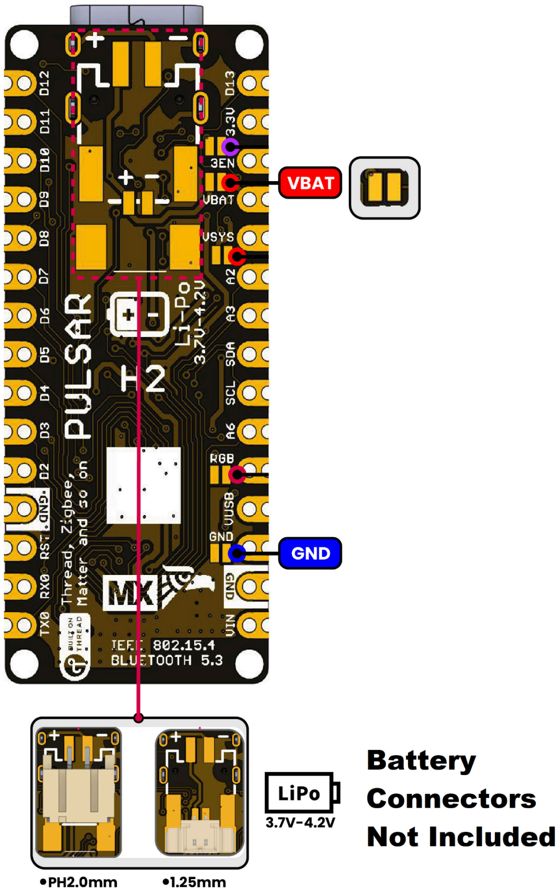
microSD Connector Pin	**microSD Pin Name**	**SPI Function**	**ESP32 H2 GPIO**	Arduino NANO (Function)
1	**DAT2**	Not used in SPI	-	-
2	**CD/DAT3**	**CS** (Chip Select)	GPIO11	D5
3	**CMD**	**MOSI** (Master Out Slave In)	GPIO5	D11/MOSI
4	**VDD**	**VDD**	3.3V	-
5	**CLK**	**SCLK** (Serial Clock)	GPIO4	D13(SCK/LED)
6	**VSS**	**GND**	GND	**GND**
7	**DAT0**	**MISO** (Master In Slave Out)	GPIO0	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 H2**, users can solder either a **PH 2.0 mm** or **JST 1.25 mm** connector to add a **3.7 V–4.2 V LiPo battery**. These connectors are not included with the board but can be purchased separately and soldered according to the user's preference.

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

Additionally, the **VBAT pin** is accessible through the **castellated headers**, allowing direct connection to the battery voltage. This feature enables users to power or interface **external modules or loads** that operate within the same voltage range as the LiPo battery (3.7 V–4.2 V).

⚠️ **Important:** Always verify the **battery connector polarity** before soldering. Incorrect polarity can permanently 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 H2** in the Arduino IDE (version 2.0 or higher is recommended), follow these steps:

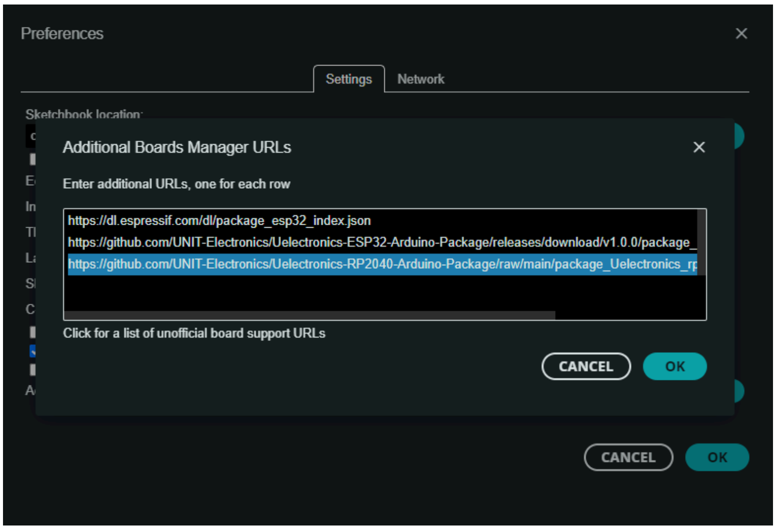
1. **Adding the UNIT Electronics Boards Package:**
    - **Open Preferences**, go to the **File** menu and select **Preferences** (or **Settings** on some systems).

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.

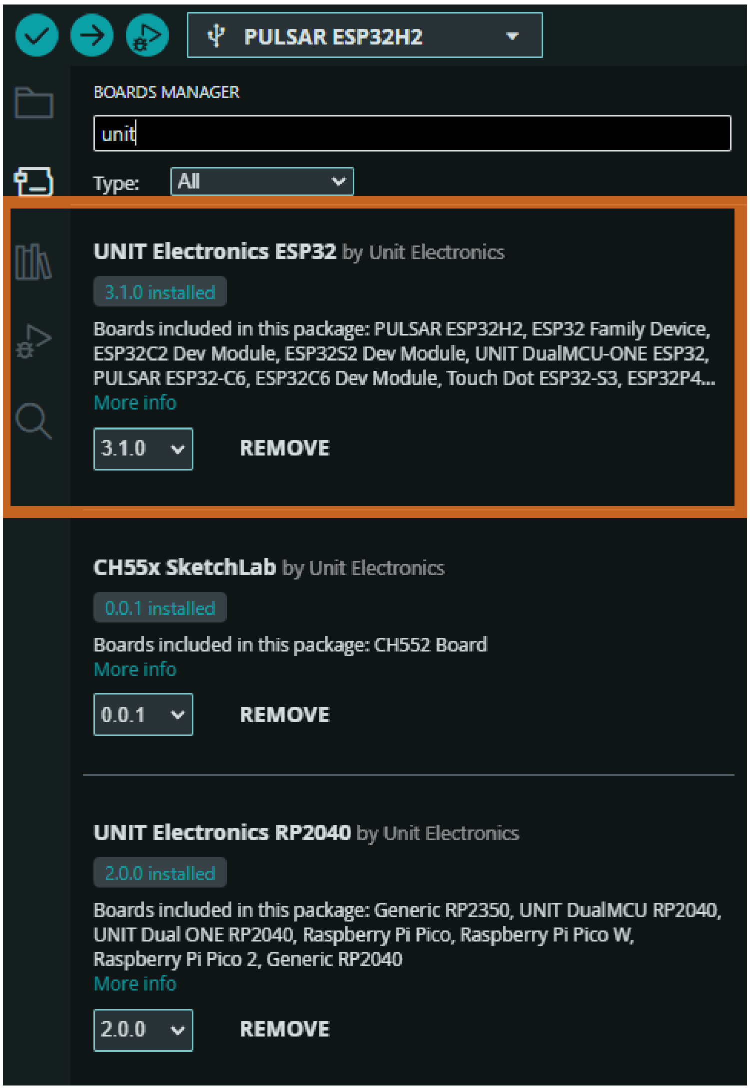
3. **Tip**: If there are multiple URLs to add, separate them with commas.



4. Installing the Board Packages

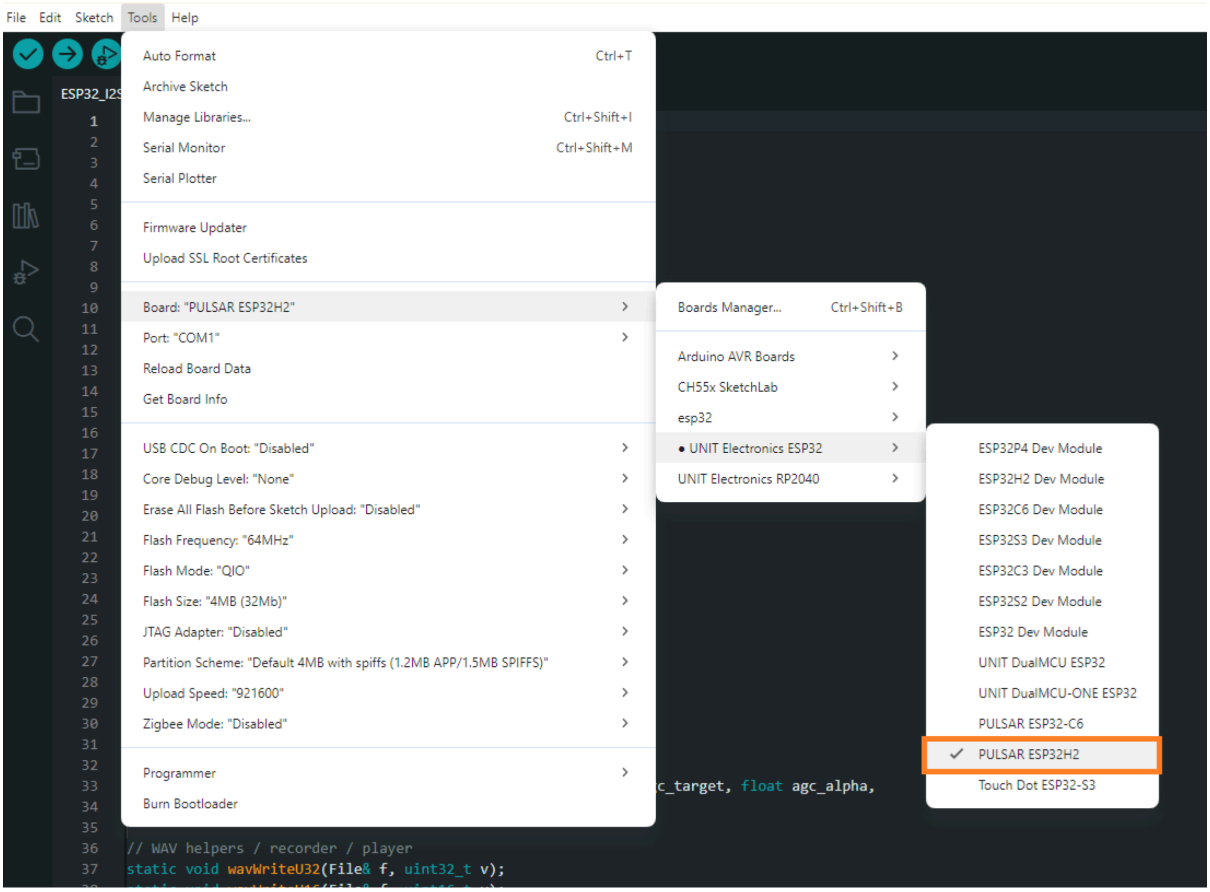
- **Open Board Manager:**
  Go to **Tools > Board > Board Manager**.
- **Search for UNIT Electronics Boards:**
  In the search bar, type **UNIT Electronics ESP32**.
- **Install the Latest Version:**
  Select the latest available version and click **Install**.
  (The PULSAR H2 board is supported starting from version 3.1.0.)

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-H2** board.



6. **Connect the Board**:

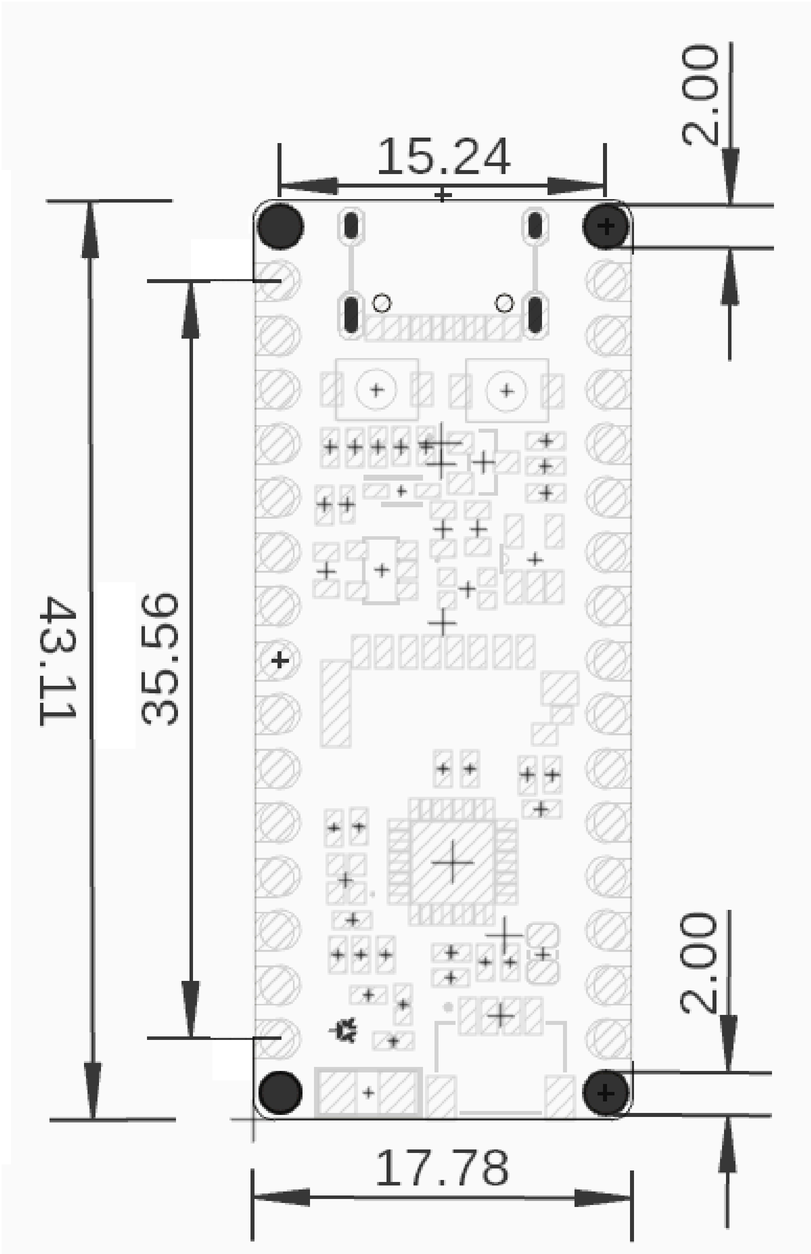
   - Use a USB Type-C cable to connect the PULSAR H2 to your computer. Ensure the correct port is selected under **Tools > Port**.

7. **Getting Started with Examples:**

   - To start programming your **PULSAR H2** 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 H2 using Arduino IDE".

6 Mechanical Information



Mechanical dimensions in millimeters of PULSAR ESP32-H2

7 Company Information

Company name	UNIT Electronics
Company website	https://uelectronics.com/
Company Address	Salvador 19, Cuauhtémoc, 06000 Mexico City, CDMX

8 Reference Documentation

Ref	Link
PULSAR ESP32-H2 Documentation	https://github.com/UNIT-Electronics-MX/unit_pulsar_esp32_h2
Uelectronics-ESP32 Arduino Package	https://github.com/UNIT-Electronics/Uelectronics-ESP32-Arduino-Package
PULSAR H2 Getting Started Guide	https://unit-electronics-mx.github.io/unit_pulsar_esp32_h2
Thonny IDE	https://thonny.org/
Arduino IDE	https://www.arduino.cc/en/software
Visual Studio Code	https://code.visualstudio.com/download
ESP32-H2FH4S Datasheet	https://www.espressif.com/sites/default/files/documentation/esp32-h2_datasheet_en.pdf
ESP32-H2 Technical Reference Manual	https://www.espressif.com/sites/default/files/documentation/esp32-h2_technical_reference_manual_en.pdf#twai

9 Appendix

9.1 Schematic (https://github.com/UNIT-Electronics-MX/unit_pulsar_esp32_h2/blob/main/hardware/SCHEMATICS_PULSAR_H2_V1.pdf)

PULSAR ESP32-H2

NANO FACTOR

SKU: UE0071