

ESP32-H2-DevKit-Lipo

User Manual

Document revision 2.1, October 2024

www.olimex.com

Table of Contents

What is ESP32-H2-DevKit-Lipo.....	3
ESP32-H2-DevKit-Lipo features.....	4
Order codes for ESP32-H2-DevKit-Lipo and accessories:.....	5
HARDWARE.....	6
ESP32-H2-DevKit-Lipo layout:.....	6
ESP32-H2-DevKit-Lipo GPIOs:.....	7
ESP32-H2-DevKit-Lipo schematics:.....	8
ESP32-H2-DevKit-Lipo power supply:.....	8
UEXT connector:.....	8
External power sense and battery measurement:.....	10
SOFTWARE.....	12
Document Revision History.....	13

What is ESP32-H2-DevKit-Lipo

ESP32-H2 combines IEEE 802.15.4 connectivity with Bluetooth 5 (LE).

The SoC is powered by a single-core, 32-bit RISC-V microcontroller that can be clocked up to 96 MHz.

The ESP32-H2 has been designed to ensure low power consumption and security for connected devices.

ESP32-H2 has 320 KB of SRAM with 16 KB of Cache, 128 KB of ROM, 4 KB LP of memory, and a built-in 2 MB or 4 MB SiP flash.

It has 19 programmable GPIOs with support for ADC, SPI, UART, I2C, I2S, RMT, GDMA and LED PWM.

Bluetooth supports: Bluetooth Low Energy (Bluetooth 5.3 certified), Bluetooth mesh, Bluetooth Low Energy long range (Coded PHY, 125 Kbps and 500 Kbps), Bluetooth Low Energy high speed (2 Mbps)

IEEE Standard 802.15.4-2015 compliant

Supports Thread 1.3

Supports Zigbee 3.0

Supports Matter

Supports HomeKit, MQTT

ESP32-H2-DevKit-Lipo features

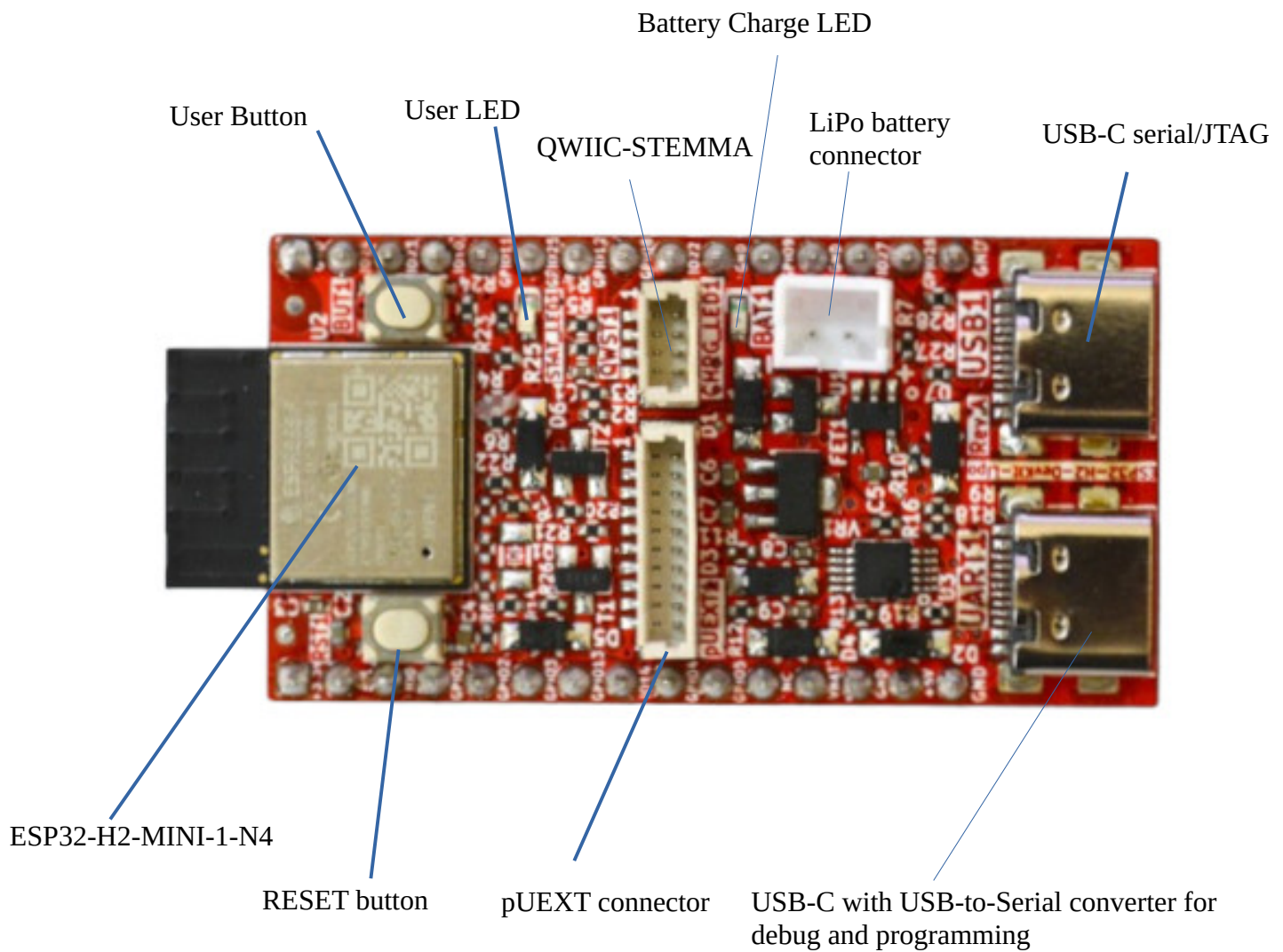
- ESP32-H2 embedded, RISC-V single-core 32-bit LX7 microprocessor, up to 96 MHz, 128 KB ROM, 320 KB SRAM, 4 KB LP Memory, 4 MB flash
- USB-C with USB to Serial converter for programming / debugging
- USB-C connected directly to ESP32-H2
- LiPo battery charger and connector
- pUEXT [UEXT connector](#)
- Qwiic/Stemma connector
- Reset button
- User button
- User LED
- All GPIOs available on two headers at 25.4 mm space
- Dimensions: 48 x 25 mm
- Operating temperature: -40+85C

Order codes for ESP32-H2-DevKit-Lipo and accessories:

<u>ESP32-H2-DevKit-Lipo</u>	commercial grade -40+85C board with internal antenna
<u>USB-CABLE-A-TO-C-1M</u>	USB-C cable
<u>BATTERY-LIPO1400mAh</u>	- Lipo battery 3.7V 1400mAh – note these batteries can be shipped only by ground so we can deliver only to EU destinations.
<u>UEXT modules</u>	- different sensors, relays, LCDs, RTC, GSM, GPS etc accessories which can be connected to UEXT connector

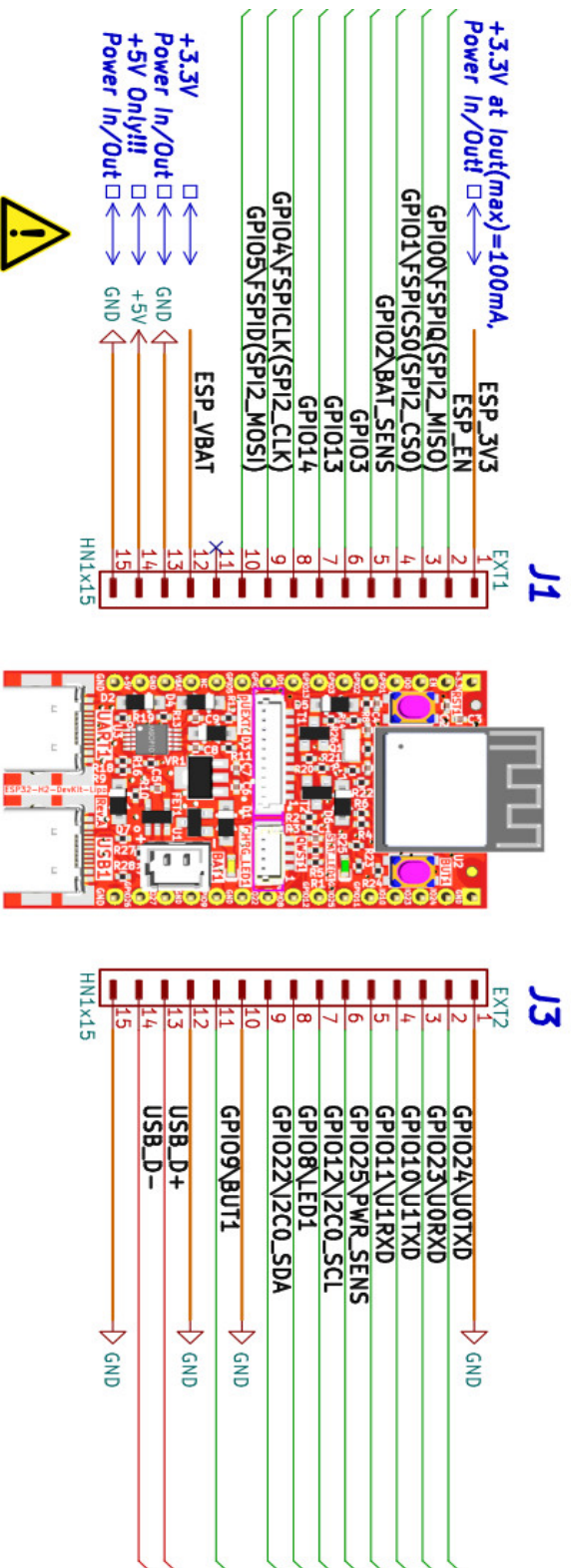
HARDWARE

ESP32-H2-DevKit-Lipo layout:



ESP32-H2-DevKit-Lipo GPIOs:

Extensions



Extension's pinout is fully compatible to the Espressif's ESP32-H2-DevKitM-1.

ESP32-H2-DevKit-Lipo schematics:

[ESP32-H2-DevKit-Lipo](#) latest schematic is on [GitHub](#)

ESP32-H2-DevKit-Lipo power supply:

[ESP32-H2-DevKit-Lipo](#) can be powered by either:

- USB-C connectors USB1 or UART1
- EXT1 pin 14 (+5V)
- LiPo battery

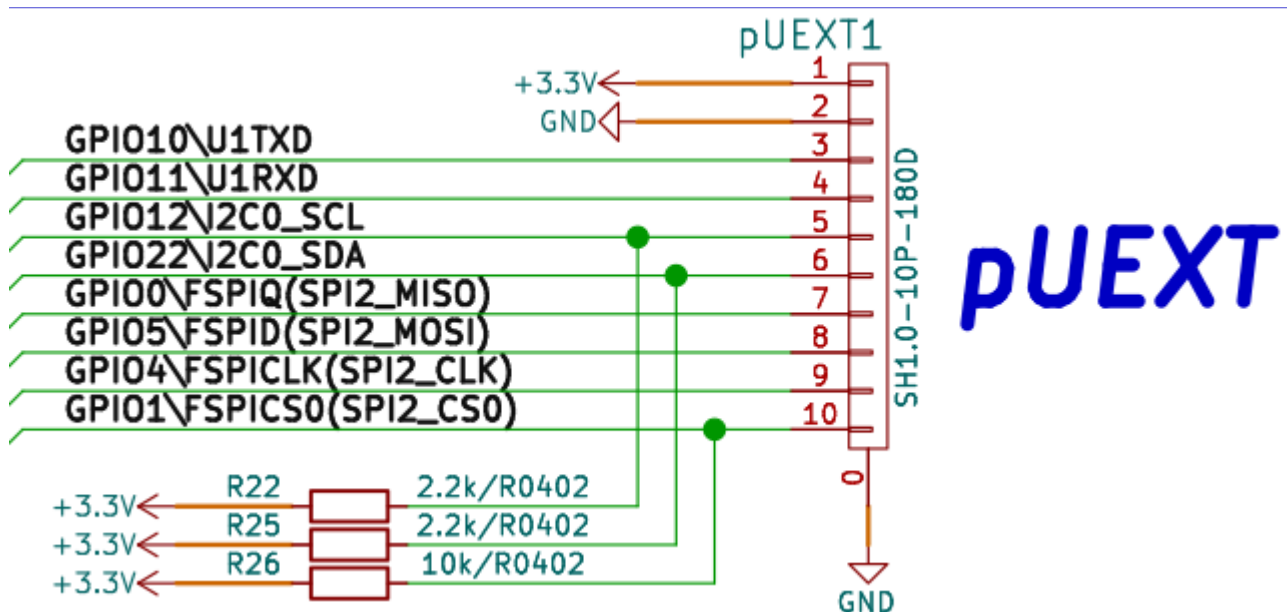
The switch between these is automatically without voltage drop.

Power consumption of ESP32-H2-DevKit-Lipo is less 10uA in deep sleep mode.

UEXT connector:

UEXT connector stands for Universal EXTension connector and contains +3.3V, GND, I2C, SPI, UART signals.

pUEXT is 1mm step boxed plastic connector. All signals are with 3.3V levels.



Olimex has developed number of [MODULES](#) with this connector. There are temperature, humidity, pressure, magnetic field, light sensors. Modules with LCDs, LED matrix, Relays, Bluetooth, Zigbee, WiFi, GSM, GPS, RFID, RTC, EKG, sensors and etc.

External power sense and battery measurement:

Battery measurements are disabled by default to provide more free GPIOs but they can be enabled by closing a couple of jumpers. In order to measure the battery you first need to close SMT jumper BAT_SENS_E (solder its pads together). This will allow you to measure the battery voltage via ESP32-H2 pin GPIO2. Notice that the board has voltage divider formed by R1 and R5(1M and 4.7M).

The board also has external power sense – aka whether you have external power or board is powered via battery (it is good idea to know if you have external power so you can put the board in some power saving mode or perform some saves). This can be enabled if you close jumper PWR_SENSE_E1 (again solder its pads together). This will allow you to determine if external power is present via pin GPIO25.

After the jumper or jumpers are closed you need some simple code to read voltage and detect external power presence. Code in Arduino (with Espressif ESP32 package installed) should be similar to this:

```
#define POWER_SENSE 25
#define BATTERY 2

void setup()
{
    Serial.begin (115200);
    pinMode (POWER_SENSE, INPUT);
    pinMode (BATTERY, INPUT);
}

void loop()
{
    Serial.print ("External power sense: ");
    Serial.println (digitalRead (POWER_SENSE));

    Serial.print ("Battery measurement: ");
    Serial.print (analogReadMilliVolts (BATTERY)*5.7);
    Serial.println (" mV");
    Serial.println ();
    delay (1000);
}
```

This would give good enough reading, but value will fluctuate a bit (this is normal for this ADC). If the fluctuations bother you try filling an array and giving average.

I also recommend you reading this bit about Arduino ADC libraries for ESP32 (it also mentions S2 specifics), it explains the functions, the options, and gives some more complex examples at the end:

<https://espressif-docs.readthedocs-hosted.com/projects/arduino-esp32/en/latest/api/adc.html>

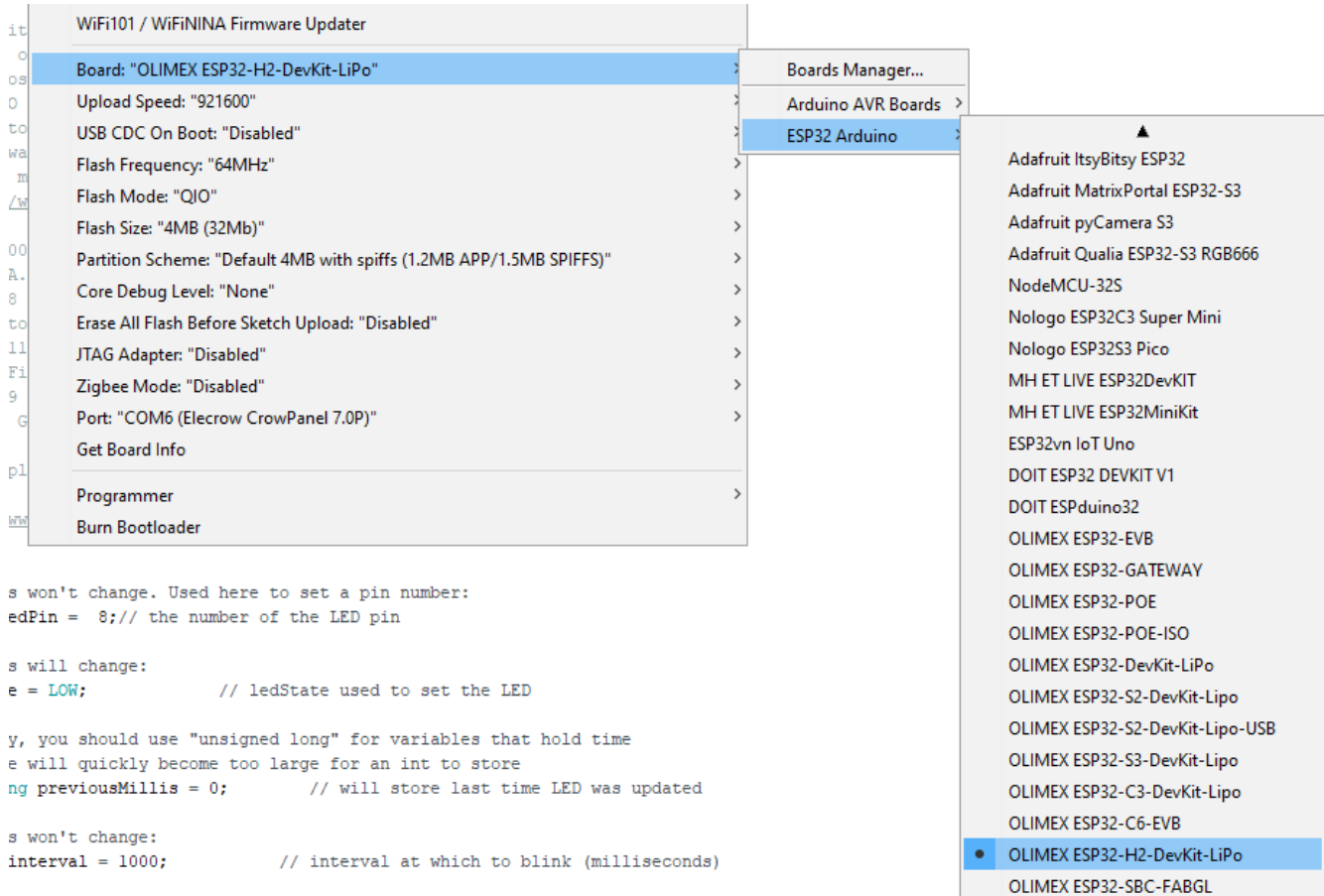
SOFTWARE

For regular programming via USB type C cable, use USB connector named "UART1".

ESP32-H2-DevKit-Lipo is supported by

- [Espressif ESP-IDF](#)
- [Arduino IDE](#)

The board has own config in the ESP32 package for Arduino IDE:



WiFi101 / WiFININA Firmware Updater

Board: "OLIMEX ESP32-H2-DevKit-LiPo"

Upload Speed: "921600"

USB CDC On Boot: "Disabled"

Flash Frequency: "64MHz"

Flash Mode: "QIO"

Flash Size: "4MB (32Mb)"

Partition Scheme: "Default 4MB with spiiffs (1.2MB APP/1.5MB SPIFFS)"

Core Debug Level: "None"

Erase All Flash Before Sketch Upload: "Disabled"

JTAG Adapter: "Disabled"

Zigbee Mode: "Disabled"

Port: "COM6 (Elecrow CrowPanel 7.0P)"

Get Board Info

Programmer

Burn Bootloader

Boards Manager...

Arduino AVR Boards >

ESP32 Arduino >

Adafruit ItsyBitsy ESP32

Adafruit MatrixPortal ESP32-S3

Adafruit pyCamera S3

Adafruit Qualia ESP32-S3 RGB666

NodeMCU-32S

Nologo ESP32C3 Super Mini

Nologo ESP32S3 Pico

MH ET LIVE ESP32DevKIT

MH ET LIVE ESP32MiniKit

ESP32vn IoT Uno

DOIT ESP32 DEVKIT V1

DOIT ESPduino32

OLIMEX ESP32-EVB

OLIMEX ESP32-GATEWAY

OLIMEX ESP32-POE

OLIMEX ESP32-POE-ISO

OLIMEX ESP32-DevKit-LiPo

OLIMEX ESP32-S2-DevKit-Lipo

OLIMEX ESP32-S2-DevKit-Lipo-USB

OLIMEX ESP32-S3-DevKit-Lipo

OLIMEX ESP32-C3-DevKit-Lipo

OLIMEX ESP32-C6-EVB

• OLIMEX ESP32-H2-DevKit-LiPo

OLIMEX ESP32-SBC-FABGL

```
s won't change. Used here to set a pin number:
ledPin = 8; // the number of the LED pin

s will change:
e = LOW; // ledState used to set the LED

y, you should use "unsigned long" for variables that hold time
e will quickly become too large for an int to store
ng previousMillis = 0; // will store last time LED was updated

s won't change:
interval = 1000; // interval at which to blink (milliseconds)
```

Document Revision History

Revision 2.1 October 2024

- Fixed wrong voltage divider multiplier
- Improved information about ADC and battery measurements

Revision 2.0 August 2024

- added info about the battery measurement

Revision 1.0 January 2024