GeeekNET ESP32 Development Board SKU: EZ-0062

From 52Pi Wiki

Contents

GeeekNET ESP32 Development Board

Description

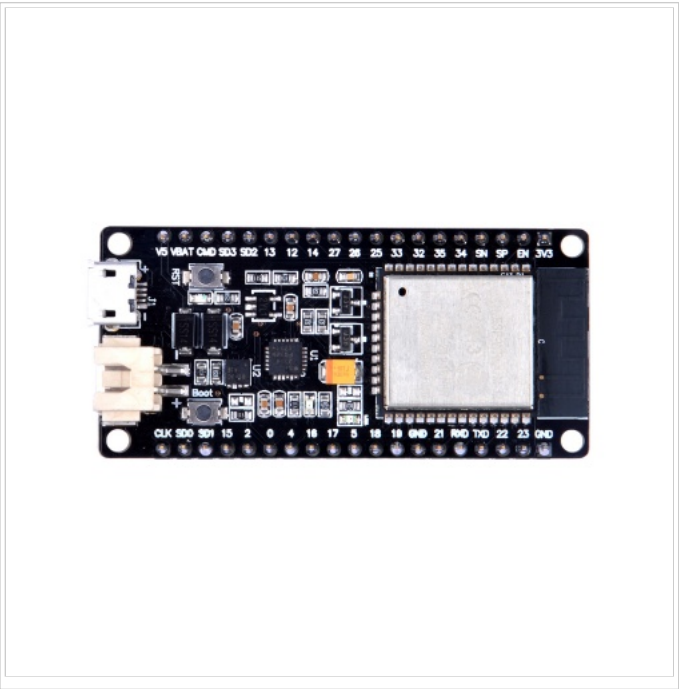
GeeekNET ESP32 Development Board is made with the official WROOM32 module.
There is built in USB-to-Serial converter, automatic bootloader reset, Lithium Ion/Polymer charger.
And just about all of the GPIOs brought out so you can use it with any sensor.

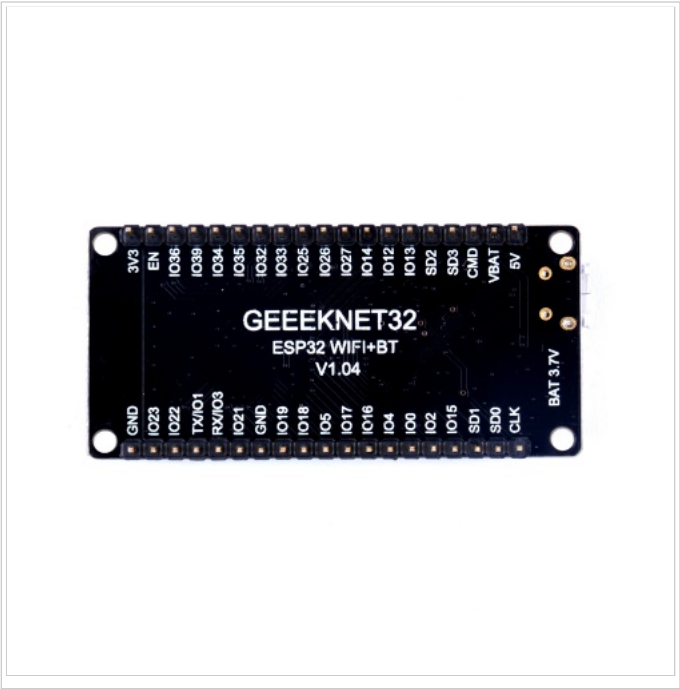
That module contains a dual-core ESP32 chip, 4 MB of SPI Flash, tuned antenna.
And all the passives you need to take advantage of this powerful new processor.
The ESP32 has both WiFi and Bluetooth Classic/LE support.
That means it's perfect for just about any wireless or Internet-connected project.

The ESP32 is a perfect upgrade from the ESP8266 that has been so popular.
In comparison, the ESP32 has way more GPIO, plenty of analog inputs,
two analog outputs, multiple extra peripherals (like a spare UART),
two cores so you don't have to yield to the WiFi manager, much higher-speed processor, etc.

Please note: The ESP32 is still targeted to developers.
Not all of the peripherals are fully documented with example code, and there are some bugs still being found and fixed.
We got it working under Arduino IDE, so you can expect things like I2C and SPI and analog reads to work.
And also we got it working under MicroPython, so you can test it with the command line console, it will be nice to programmer.
But other elements are still under development.
For that reason, we recommend this module for makers who have some experience with microcontroller programming, and not as a first dev board.

Galley





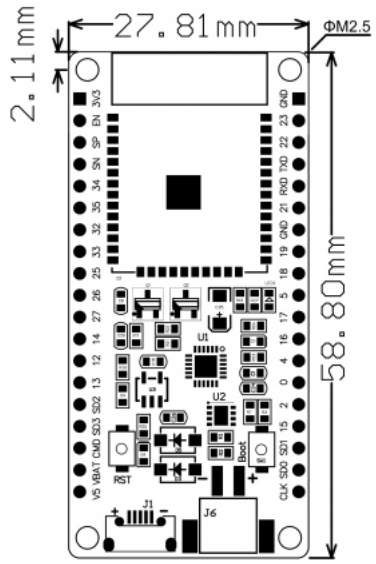
Product Details

- Voltage: 5V
- Current: 80mA
- Battery: 3.7V
- Dimension: 59.76mmx28.05mmx12.60mm
- Weight: 9.75g ±0.5

ESP32 module Specifications

WiFi	Bluetooth	CPU and Memory	Clocks and Timers	Advanced Peripheral Interfaces	Security	Cryptographic hardware acceleration	D
▪ 802.11 b/g/n/e/i ▪ 802.11 n (2.4 GHz), up to 150 Mbps ▪ 802.11 e: QoS for wireless multimedia technology ▪ WMM-PS, UAPSD ▪ A-MPDU and A-MSDU aggregation ▪ Block ACK ▪ Fragmentation and defragmentation ▪ Automatic Beacon monitoring/scanning ▪ 802.11 i security features: pre-authentication and TSN ▪ Wi-Fi Protected Access (WPA)/WPA2/WPA2-Enterprise/Wi-Fi Protected Setup (WPS) ▪ Infrastructure BSS Station mode/SoftAP mode ▪ Wi-Fi Direct (P2P), P2P Discovery, P2P Group Owner mode and P2P Power Management ▪ UMA compliant and certified ▪ Antenna diversity and selection	▪ Compliant with Bluetooth v4.2 BR/EDR and BLE specification ▪ Class-1, class-2 and class-3 transmitter without external power amplifier ▪ Enhanced power control ▪ +12 dBm transmitting power ▪ NZIF receiver with -97 dBm sensitivity ▪ Adaptive Frequency Hopping (AFH) ▪ Standard HCI based on SDIO/SPI/UART ▪ High speed UART HCI, up to 4 Mbps ▪ BT 4.2 controller and host stack ▪ Service Discover Protocol (SDP) ▪ General Access Profile (GAP) ▪ Security Manage Protocol (SMP) ▪ Bluetooth Low Energy (BLE) ▪ ATT/GATT ▪ HID ▪ All GATT-based profile supported ▪ SPP-Like GATT-based profile ▪ BLE Beacon ▪ A2DP/AVRCP/SPP, HSP/HFP, RFCOMM ▪ CVSD and SBC for audio codec ▪ Bluetooth Piconet and Scatternet	▪ Xtensa® Single-/Dual-core 32-bit LX6 microprocessor(s), up to 600 DMIPS ▪ 448 KB ROM ▪ 520 KB SRAM ▪ 16 KB SRAM in RTC ▪ QSPI flash/SRAM, up to 4 x 16 MB • Power supply: 2.3V to 3.6V	▪ Internal 8 MHz oscillator with calibration ▪ Internal RC oscillator with calibration ▪ External 2 MHz to 60 MHz crystal oscillator (40 MHz only for Wi-Fi/BT functionality) ▪ External 32 kHz crystal oscillator for RTC with calibration ▪ Two timer groups, including 2 x 64-bit timers and 1 x main watchdog in each group ▪ RTC timer with sub-second accuracy ▪ RTC watchdog	▪ 12-bit SAR ADC up to 18 channels ▪ 2 × 8-bit D/A converters ▪ 10 × touch sensors ▪ Temperature sensor ▪ 4 × SPI ▪ 2 × I2S ▪ 2 × I2C ▪ 3×UART ▪ 1 host (SD/eMMC/SDIO) • 1 slave (SDIO/SPI) ▪ Ethernet MAC interface with dedicated DMA and IEEE 1588 support • CAN 2.0 ▪ IR (TX/RX) ▪ Motor PWM ▪ LED PWM up to 16 channels ▪ Hall sensor ▪ Ultra-low-noise analog pre-amplifier	▪ IEEE 802.11 standard security features all supported, including WFA, WPA/WPA2 and WAPI ▪ Secure boot ▪ Flash encryption ▪ 1024-bit OTP, up to 768-bit for customers	▪ AES ▪ HASH (SHA-2) library ▪ RSA ▪ ECC ▪ Random Number Generator (RNG)	▪ ▪

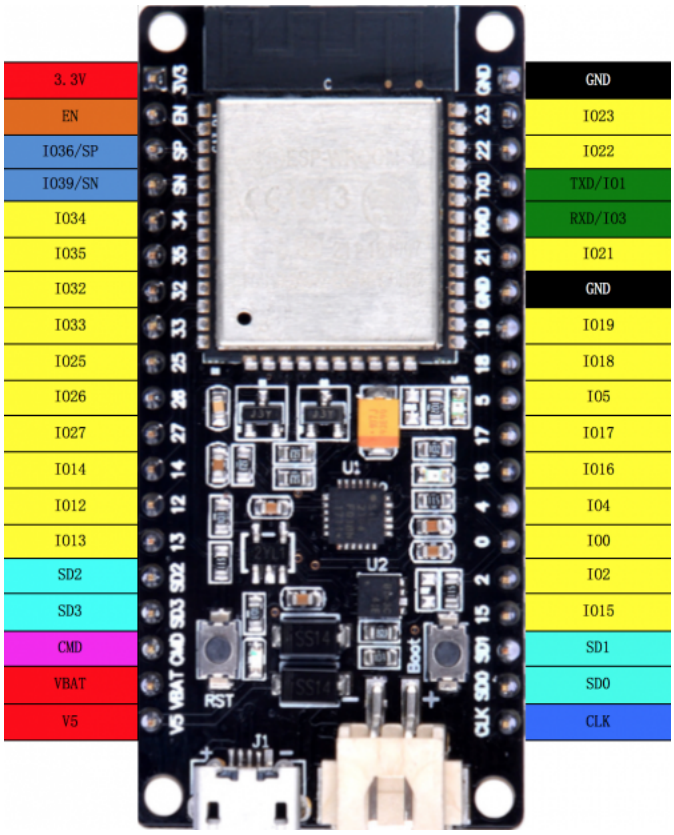
Mechanical Drawing



- Download PDF file:

File:GEEEKNET MEC.pdf

GeeeKNET ESP32 Development Board Pin Layout

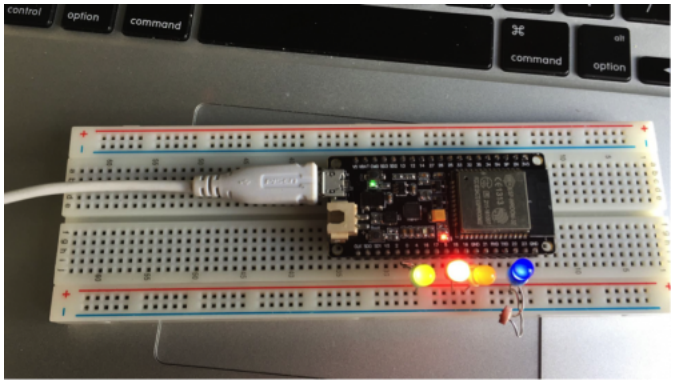


Package Include

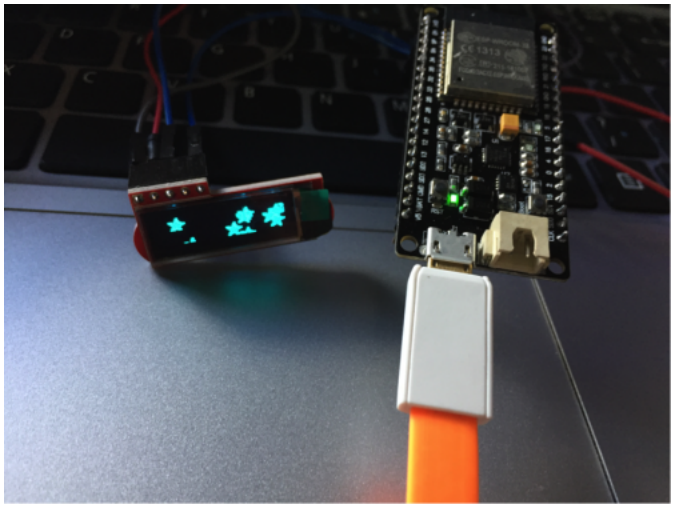
1 x GeeekNET ESP32 Development Board

Applications

Most people lights up LEDs at the very beginning when they got this module.



And you can also drive 0.91inch OLED by adafruit driver in Arduino IDE:



Working on Arduino IDE

- Installation instructions for Windows

[Installation Guide (https://github.com/geeekpi/arduino-esp32/blob/master/docs/arduino-ide/windows.md)]

- Installation instructions for Debian / Ubuntu OS

1. Install latest Arduino IDE from [arduino.cc (https://www.arduino.cc/en/Main/Software)]
Open Terminal and execute the following command (copy->paste and hit enter):

```
sudo usermod -a -G dialout $USER && \
sudo apt-get install git && \
wget https://bootstrap.pypa.io/get-pip.py && \
sudo python get-pip.py && \
sudo pip install pyserial && \
mkdir -p ~/Arduino/hardware/espressif && \
cd ~/Arduino/hardware/espressif && \
git clone https://github.com/espressif/arduino-esp32.git esp32 && \
cd esp32/tools/ && \
python get.py
```

2. Restart Arduino IDE

- Installation instructions for Mac OS

1. Install latest Arduino IDE from [arduino.cc (https://www.arduino.cc/en/Main/Software)]
Open Terminal and execute the following command (copy->paste and hit enter):

```
mkdir -p ~/Documents/Arduino/hardware/espressif && \
cd ~/Documents/Arduino/hardware/espressif && \
git clone https://github.com/espressif/arduino-esp32.git esp32 && \
cd esp32/tools/ && \
python get.py
```

- Restart Arduino IDE

Working on MicroPython

- You can download the latest firmware for ESP32 boards:

Download Firmware for ESP32 boards (https://micropython.org/download#esp32)

- The following files are daily firmware for ESP32-based boards. (Unzip it before using it)

File:Esp32-20170822-v1.9.1-438-g392aaffc.bin.zip

Program your board using the esptool.py program, and put the firmware starting at address 0x1000
If you are putting MicroPython on for the first time then you should first erase the entire flash.
PS: /dev/ttyUSB1 may different on your PC, please make sure it correctly.

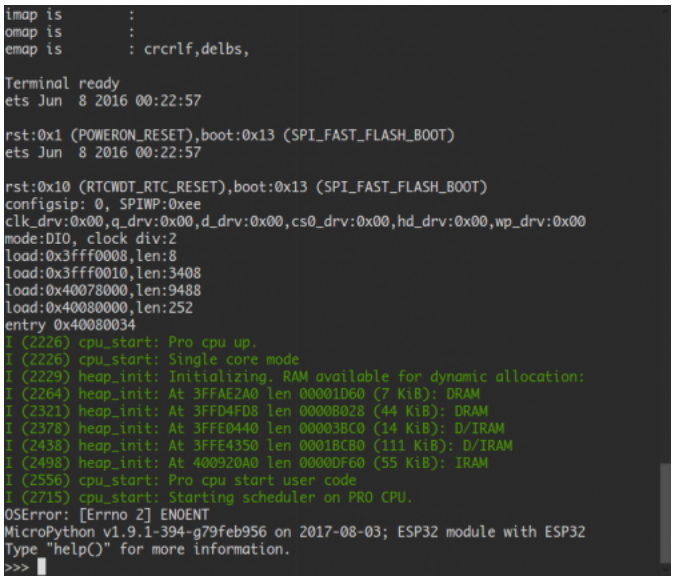
```
sudo esptool.py --chip esp32 --port /dev/ttyUSB1 erase_flash
sudo esptool.py --chip esp32 --port /dev/ttyUSB1 write_flash -z 0x1000 firmware.bin)
```

After that, you need install esptool.py and a software called "picocom" in your Linux system(debain, ubuntu, or raspbian):
There are a lot of terminal softwares such as "minicom" or "screen". You can select one of them by yourself.

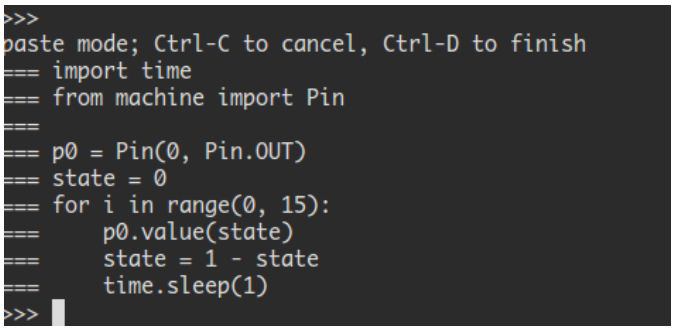
```
sudo pip install esptool.py
sudo apt-get -y install picocom
sudo picocom -b 115200 /dev/ttyUSB1
```

You will enter an IDE environment so you can coding by python language.
eg. you can import machine module as:

```
import machine
```



- Or just test range function it in your IDE:



- Control LED Pin

```
from machine import Pin
p0 = Pin(0, Pin.OUT) # Setting GPIO0's direction to output mode
p0.value(1) # Setting values to 1 means "HIGH" level
p0.value(0) # Setting values to 0 means "LOW" level
```

Machine module

You can using the Machine module to read CPU frequency.

```
import machine
machine.freq() # get current CPU frequency
machine.freq(160000000) # set current CPU frequency
```

ESP-IDF SDK

- Obtaining v2.1

The source files attached to this release will not work due to our use of git submodules. Use one of the following methods instead:

- Using git

To get this release, use the following commands:

```
git clone https://github.com/espressif/esp-idf.git esp-idf-v2.1
cd esp-idf-v2.1/
git checkout v2.1
git submodule update --init --recursive
```

This is the recommended way of obtaining v2.1 of ESP-IDF.