

TinyPICO – ESP32 Development Board

PRODUCT ID: 4335

DESCRIPTION

The smallest, most feature-rich ESP32 dev board is here! TinyPICO has incredible processing power for a board that's shorter than an alkaline battery.

Thanks to creator-maker Seon Rozenblum, you can tinker with a tiny, mighty, uncompromising board with 4MB extra RAM, on-board RGB LED, and a 700 mAh 3.3V regulator. MicroPython firmware is pre-loaded, and with support for Arduino IDE and Espressif IDF, you get the flexibility to code your way.

Comes with unassembled male headers, stacking headers, JST PH battery connector, and JST MicroBlade battery connector.

[Getting Started Guide](#)

Why We Made TinyPICO

There are quite a few ESP32 boards on the market, but they all require you to compromise on one or more features. Some don't have on-board battery management, while some do but they don't have low deep sleep current. Others have great low-power modes, but are large and not breadboard-friendly, and none of them have extra RAM unless you go for a more expensive and larger WROVER-powered board.

We just weren't happy with the status quo – we wanted to have our cake and eat it too!

So we designed the smallest un-compromising ESP32 development board in the world, and then went a step further and gave it 4 MB of extra RAM, an on-board RGB LED, and more juice with a 700 mA 3.3 V regulator.

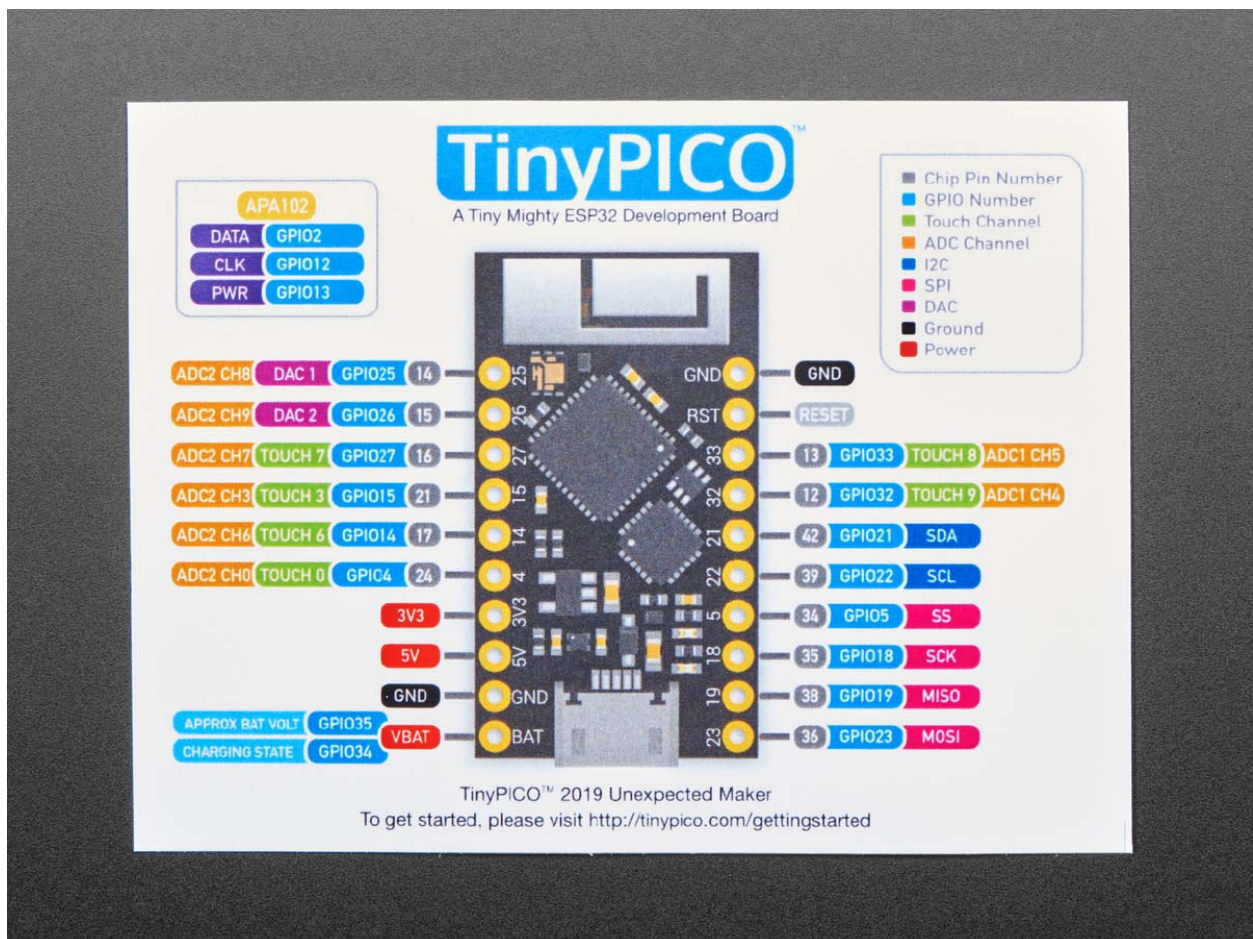
TECHNICAL DETAILS

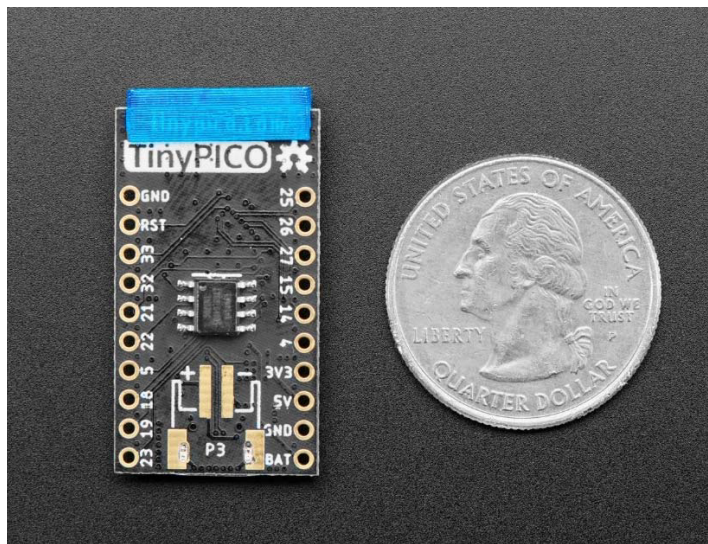
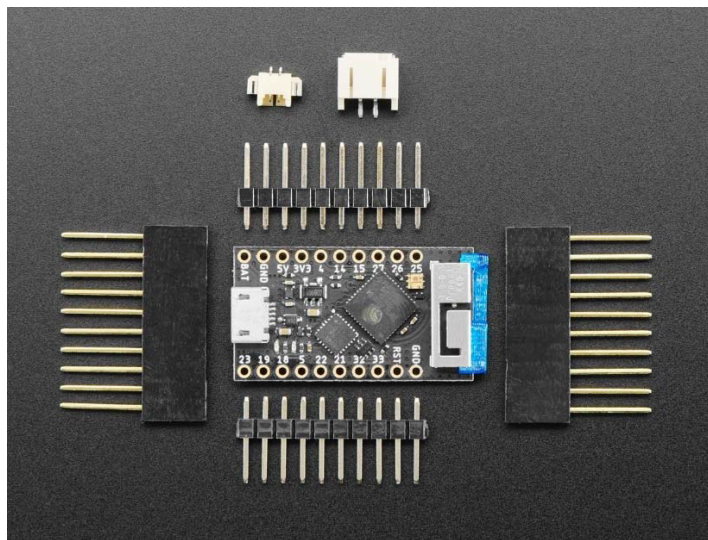
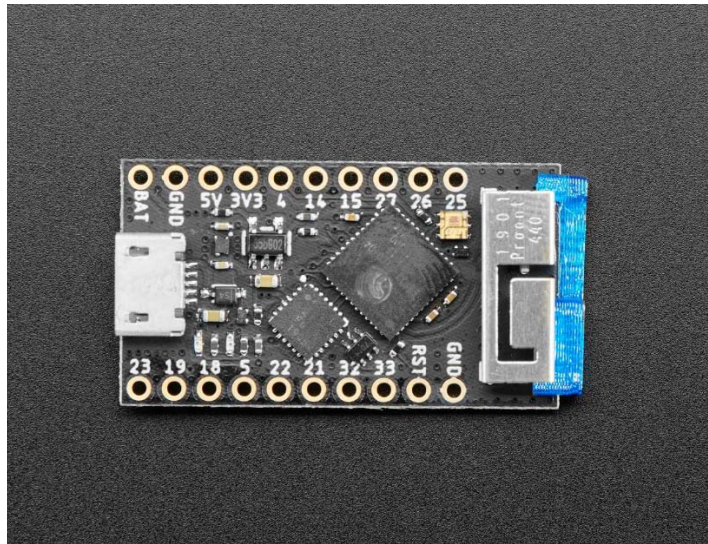
Specs:

- Processing:
 - 32-bit dual-core processor operating at 240 MHz
 - 4 MB SPI flash
 - 4 MB extra PSRAM
- Wireless:
 - 2.4 GHz Wi-Fi – 802.11 b/g/n
 - Bluetooth BLE 4.2
 - 3D antenna
- Power:
 - 700 mAh 3.3 V LDO regulator
 - Optimized power path for low-power battery usage
 - LiPo battery management
 - Battery pads on the bottom support both JST PH & MicroBlade connectors (included unassembled with every TinyPICO)
 - Power (red) & charge (orange) LEDs
- Form-factor:
 - Breadboard-friendly

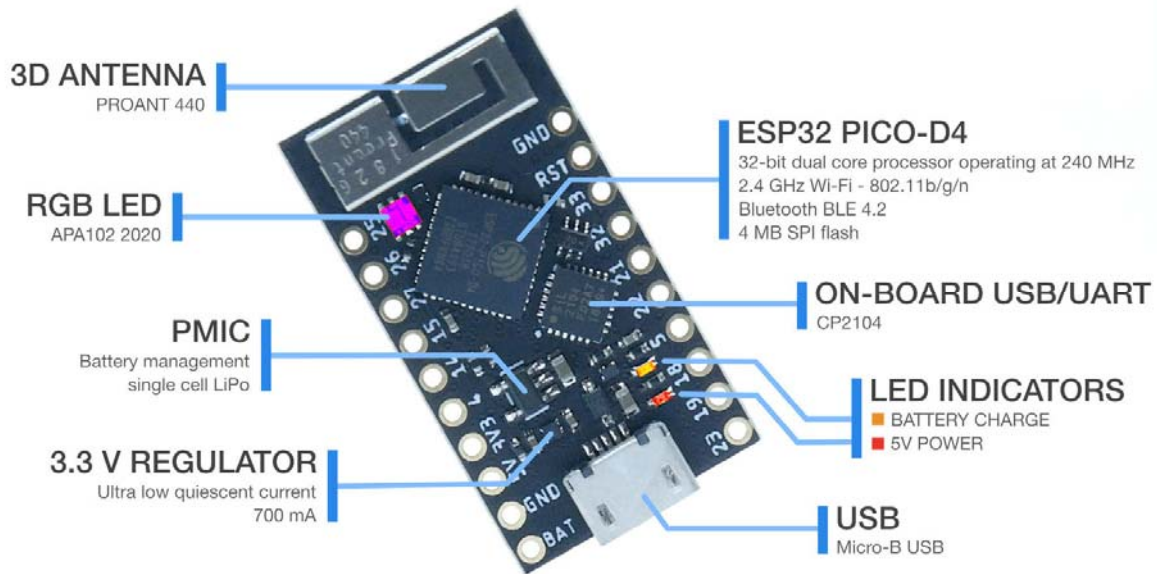
- Oh, it's tiny! Just 18 mm x 32 mm
- Connectivity:
 - 14x GPIO broken out to standard pitch headers
 - USB + serial/UART for programming
 - APA102 RGB LED
- Control:
 - MicroPython pre-installed
 - Supports Arduino IDE
 - Supports Espressif IDF
 - Lots of example code and libraries

Product Weight: 2.0g / 0.1oz





Getting Started



CONGRATULATIONS!

If you are visiting this page, it's likely you are a proud owner of a TinyPICO - congratulations!

We've put together this getting started guide to help you get familiar with your new TinyPICO, and showcase some of the main features of the board - including an FAQ covering some of the questions we were asked during the crowd funding campaign.

MAIN FEATURES

Your TinyPICO is pretty spec'd up! We've taken a lot of time and care to make it small, performant and as powerful and useful as possible, given our self imposed size constraints ;)

Although the TinyPICO uses a PICO-D4 chip from Espressif, it's still the same powerful dual core ESP32 that you already might know and love. Just way smaller than you might be used to using!

That means you can do anything on the TinyPICO that you can do on any other ESP32 development board, within the constraints of the 14 GPIO exposed.

There are already hundreds of libraries and tutorials and videos online that cover everything from using an ESP32 to play music, serve web pages, output vision on a VGA screen, connect to RGB LED matrix panels and so much more. It's a really well supported platform, that is growing even better every day.

HOOKIN IT UP

All you need to do to get started is to plug your TinyPICO into your computer using a Micro-B USB cable.

Your TinyPICO ships with MicroPython pre-installed and has some hello world code on it that runs when you power it up.

Depending on when your TinyPICO was packed during the campaign, you will see one of 2 visual displays on the onboard RGB LED

A CYCLING RAINBOW

or

3 FLASHES OF GREEN, THEN A CYCLING RAINBOW

The 3 flashes in green are a RAM test result, indicating the extra PSRAM is working correctly. This extra visual indicator was added later in production to improve my testing speed on boards as they were being packed.

If you open a serial terminal with the TinyPICO plugged in, you will see some hello world info displayed after it boots. Here is an example of the output running on a Mac using `screen` in a terminal window:

The output will continue to display the PICO-D4 internal temperature every few seconds just to show you continued output so you know the TinyPICO is present.

MICROPYTHON

As mentioned above, your TinyPICO has MicroPython installed on it already, and the version it's been flashed with is now v1.11 from the mainline repository + a bunch of fixes and improvements that have been implemented since v1.11 was released.

If you would like to update the version of MicroPython on your TinyPICO, you can follow the steps in the RE-INSTALLING MICROPYTHON on our [coding examples page](#).

ARDUINO IDE

If you plan on using the Arduino IDE to program your TinyPICO, all you need to do is plug it in and flash it. MicroPython will be removed in the process and your TinyPICO will act just like any other development board you have used before.

If your TinyPICO is the first ESP32 development board you've used, remember to follow the [getting started guide](#) to make sure you have installed everything you need before hand.

Version 1.0.3 of the ESP32 Arduino Core is now available in the Arduino IDE boards manager - please make sure you are on the latest version.

If everything has been installed correctly, you'll need to select TinyPICO from the ESP32 boards list, and then you are set to go!

WHAT DO I DO NOW?

Now all you need need to do is check out our [coding examples](#) to find out where to get your TinyPICO helper library and get started coding!!!!

POWER FROM BATTERY

Your TinyPICO comes with 2 types of JST connectors and the board has pads on the back where you can solder either of the connectors.

You can also connect a battery to the BAT & GND pins on the headers, if you don't want to solder a JST connector on the board.

You can use any size, single cell Li-Po battery to power your TinyPICO. Obviously, the larger the *mAh* rating of the battery, the longer it will last.

To get optimal battery life, remember to shut down the RGB LED if you are not using it, or whenever it's not lit up.

Please use LiPo-Batteries with charge protection included on them. Most hobby RC batteries do not have over discharge protection, and if a battery discharges too much, it can damage it and it might never be able to hold a proper charge again.

NOTE: When connecting a battery to your TinyPICO, please check the polarity of the battery, as not every battery is wired the same way. The IC we use for charging does have in-built reverse voltage protection, but it's better not to tempt fate!

FAQ

- MY RED POWER LED DOESN'T LIGHT UP WHEN RUNNING OFF BATTERY
- MY ORANGE CHARGE LED BLINKS WHEN NO BATTERY IS CONNECTED
- HOW FAST WILL MY BATTERY CHARGE?
- SOMETIMES MY BATTERY DOESN'T START CHARGING WHEN I PLUG IT IN
- I'VE SET MY RGB LED TO BLACK/CLEAR, BUT IT'S STILL DRAWING A LOT OF CURRENT
- WHAT WIFI RANGE CAN I EXPECT?
- I'VE PLUGGED MY TINYPICO IN BUT I CAN'T SEE IT IN THE ARDUINO IDE OR IN /DEV/
- WHY IS THERE NO RESET BUTTON?
- WHAT'S THE BLUE PLASTIC THING ATTACHED TO THE ANTENNA?

MY RED POWER LED DOESN'T LIGHT UP WHEN RUNNING OFF BATTERY

That is by design! The power and charge LEDs and corresponding 2K resistors drain the battery, so they have been isolated to the 5V power domain, so they will only light up and draw power when the board is either being powered by the USB plug, or by a 5V power source connected to the 5V header pin.

MY ORANGE CHARGE LED BLINKS WHEN NO BATTERY IS CONNECTED

That is the correct behaviour! The charge IC can't tell the difference between a full battery and no battery, so it cycles between trying to charge and not charge.

HOW FAST WILL MY BATTERY CHARGE?

The charge rate is set to approximately 300mA based on the 3K3 charge resistor we used. This is far from the fastest you can charge a battery, but we'd rather err on the side of safety than be too aggressive.

SOMETIMES MY BATTERY DOESN'T START CHARGING WHEN I PLUG IT IN

If you have 5V power connected to your TinyPICO before you plug your battery in, sometimes it might not start charging. Just unplug and re-plug in the 5V source and it should start charging correctly. This is a common issue with PMICs (Power Management ICs).

I'VE SET MY RGB LED TO BLACK/CLEAR, BUT IT'S STILL DRAWING A LOT OF CURRENT

The APA102 has a quiescent current of between 800uA and 1mA - that's a lot! To counter this, we added a PNP high side switch to the power of the APA, connected to GPIO2. So you not only need to set the APA to be black/clear, but you also need to pull GPIO2 high to cut power to the APA.

To make this easy for you, we added a simple helper function to both our MicroPython and Arduino TinyPICO libraries :-)

WHAT WIFI RANGE CAN I EXPECT?

This is really a tough question to answer as there are so many external factors that can affect WiFi performance, including construction of the building you are in, your network layout, power or RF noise in your area and where you are placing your TinyPICO.

All we can say is we spent a great deal of time tuning the matching network and the antenna implementation, to ensure it's the best it can be. The rest is up to you and your environment.

I'VE PLUGGED MY TINY PICO IN BUT I CAN'T SEE IT IN THE ARDUINO IDE OR IN /DEV/

The TinyPICO uses a SiLabs CP2104 to connect the USB to the PICO-D4, so you need to make sure you have the driver installed. You can grab the [latest drivers from from the SiLabs website](#).

Once the driver is installed (you may need a reboot your computer) the TinyPICO should appear in the Arduino IDE and as a device on your computer in the following formats:

macOS: **/dev/tty.SLAB_USBtoUART**

Linux: **/dev/ttyUSBx** where **x** is the index of the USB device

Windows: **COMn** where **n** is the port number assigned by Windows

WHY IS THERE NO RESET BUTTON?

Honestly? Space, and though it can come in handy *sometimes*, We personally rarely need to use it. I have put the RESET and GND pins next to each other so they can be easily shorted to reset the board if you do need to, and depending on your project, you can always wire up a button to those pins. No pull-ups are needed as the reset line is already pulled up, so a button taking it to ground is all that is needed for a hardware reset.

Handy tip: In MicroPython, pressing Ctrl+D will trigger a software reset

WHAT'S THE BLUE PLASTIC THING NEXT TO THE ANTENNA?

The blue 3D printed plastic thing is a shipping protector, that's there to prevent the delicate antenna from getting squished or bent during shipping.

You can actually leave it on if you want. It will impact the antenna characteristics *a little bit*, but honestly, in unmeasurable amounts - and it looks cool!!!

If you do want to remove it, please do it carefully, by rotating it off sideways.