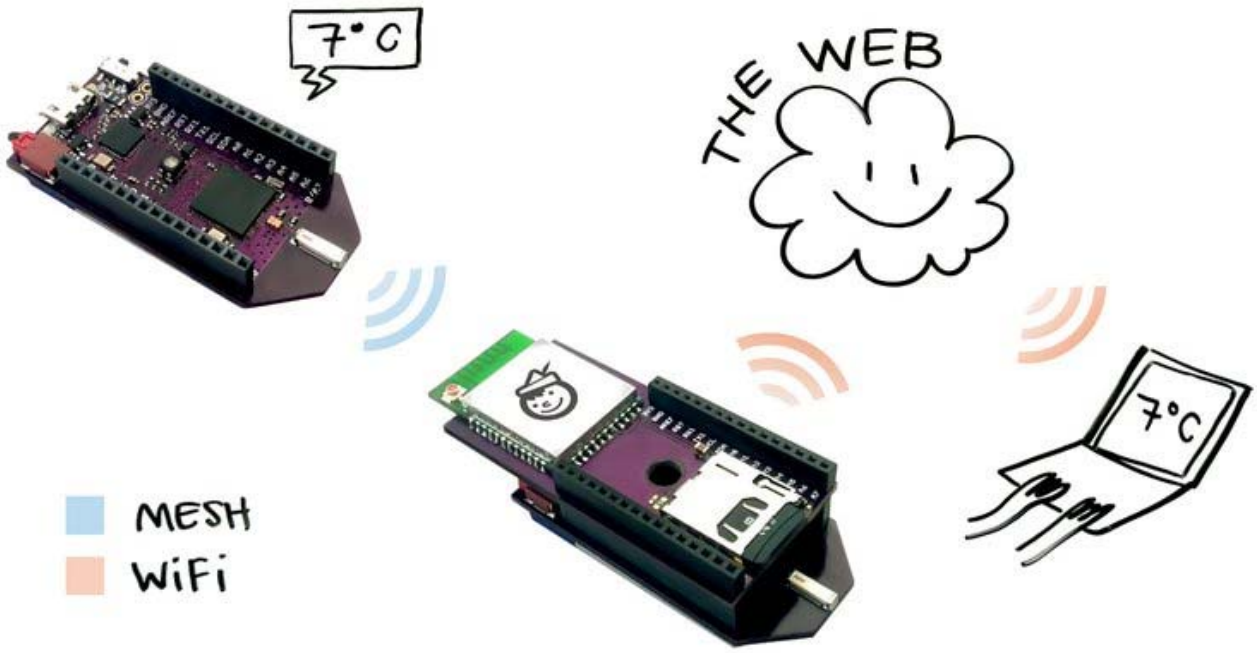
CROWD SUPPLY

Pinoccio

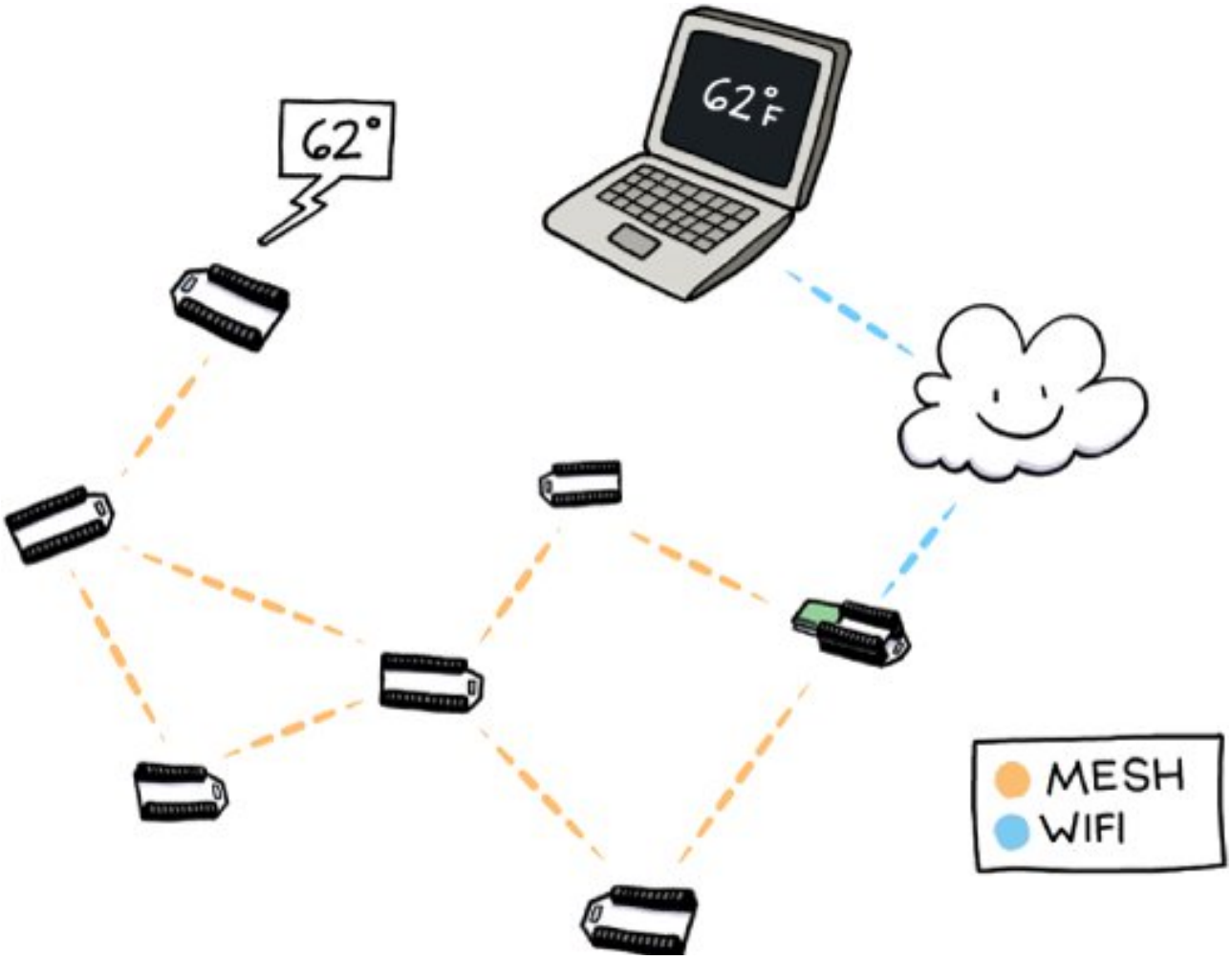
A wireless Arduino-compatible microcontroller with Wi-Fi, mesh networking, and rechargeable battery.



Mesh networked, web-connected

Field Scouts talk to each other using a mesh network (called a **Troop**), using an extremely low-power radio. This makes them *14 times more power efficient* than standard WiFi devices. Slap a WiFi backpack on a Scout to make it the **Lead Scout**, and connect your entire Troop to the web.

Sending messages between Scouts (or out to the web) is super straightforward, and it all just works.



Battery included, recharge circuit built-in

Every Scout comes with its own (removable) rechargeable battery, making it *truly* wireless, straight out of the box. We've also gone ahead and built one of those pesky charging circuits right into the board, so you can monitor your power levels, and recharge your Scout using the USB connector, or a power source of your own (bust out those solar panels).

API - free, real-time, streaming, RESTful, and optional

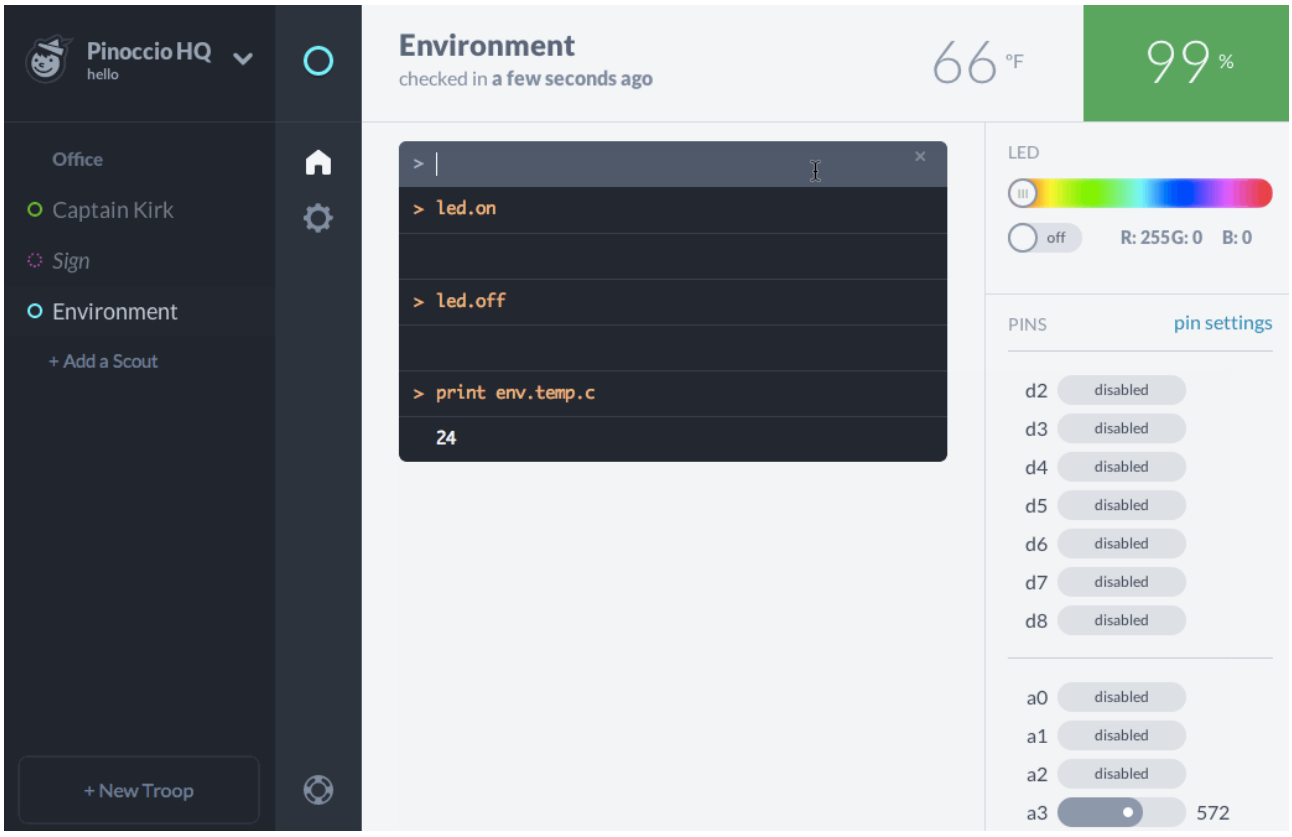
Use our free API, or roll your own, it's up to you. Our real-time REST API lets you stream data out of each and every Scout. That means you'll be notified *instantly* when something changes (pin values, LED, temperature, etc). You can also stream your own reports through the API, and effortlessly query historical data. Check out the API docs.

ScoutScript - instant feedback, no waiting to compile

Compile, test, tweak, repeat. Who has the time? ScoutScript gives you shell-like access to your hardware. Write and edit functions, run commands, read sensors, or send messages — all over-the-air, and instantaneously — no more waiting to compile! We even put a ScoutScript console in HQ, and an easy hq.print function to send it messages - it's the remote serial console you've always wanted! Take a look through the ScoutScript commands.

HQ gives your hardware some GUI goodness

In addition to being able to use the Arduino IDE, Pinoccio HQ gives you a fully-synced, real-time GUI for all your Scouts and Troops. See who's online/offline, check on battery levels, set pin modes, see (or make) pin changes instantly, mess with the LED, write some ScoutScript… the list really does go on. The best part is - **we built HQ with the Pinoccio API**. That's right, everything you see in HQ is driven by the very same API you have access to as an owner of Pinoccio.



It's Open Source, and Arduino compatible

Pinoccio is built on open source hardware and software. We give you all the files and data sheets you need to build your own Pinoccio, or integrate it into your own project.

Pinoccios are also Arduino compatible, so you can program them using the Arduino IDE, and utilize the huge collection of Arduino libraries.

Community supported

The Pinoccio Community is a place where you can share ideas, projects, code, help each other out, and collaborate on projects. Want to join the discussion? The community forums are open!

Tech Specs:

Microcontroller

- Atmel ATmega256RFR2
- Built-in radio for mesh networking (unobstructed range ~ 10m)
- 16MHz MCU
- 256k Flash
- 32k SRAM
- 8k EEPROM
- 1.8 - 3.3 volt power

Battery

- Rechargeable LiPo battery (550mAh)
- Charges via micro USB
- Ability to check battery voltage from the microcontroller via a low-power fuel gauge IC.
- You can log into your account to check battery levels.

Connections

- 17 digital I/O pins, including four with PWM
- 8 analog input pins
- 2 hardware UART serial ports
- Hardware SPI port
- Dedicated I2C port
- Micro USB port for charging and programming

Sensors etc.

- On-board temperature sensor
- On-board RGB LED

- External switch support for custom enclosures
- ICSP support for low-level programming
- Open-source hardware

Radio

- 2.4GHz using 802.15.4
- Wireless, over-the-air programming

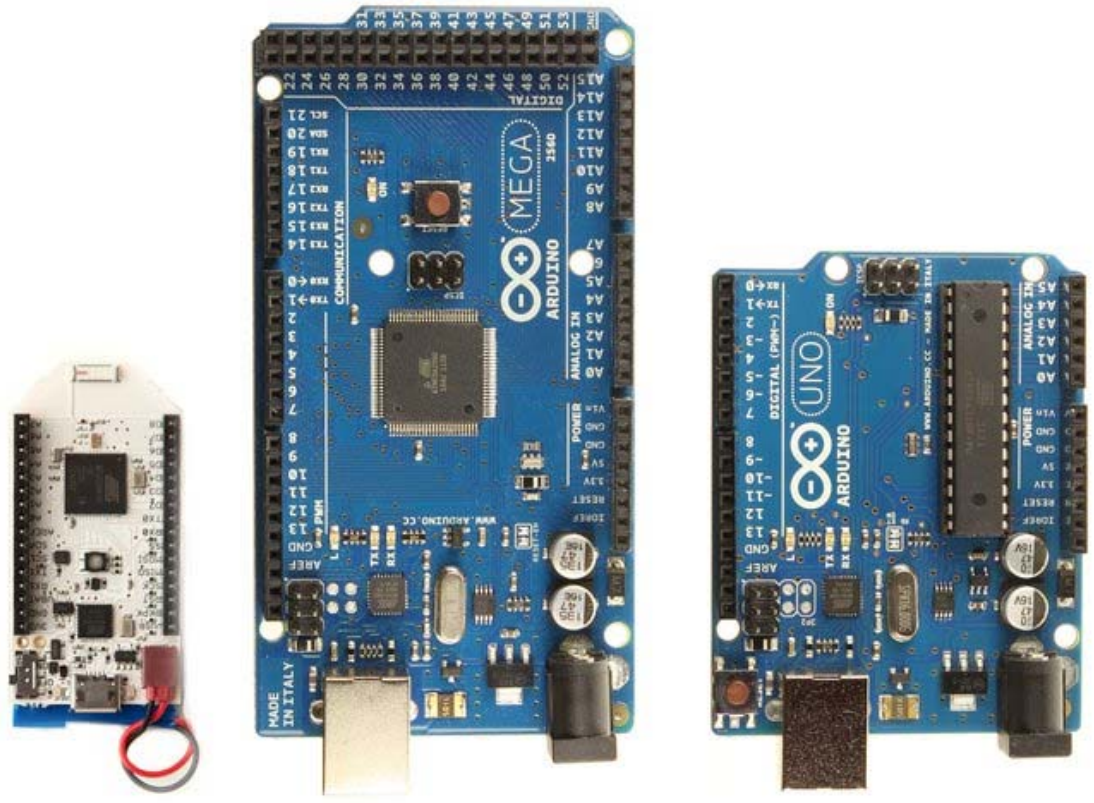
WiFi

- Web connectivity via the WiFi Backpack - *Lead Scout only*

Dimensions

	Field Scout	Lead Scout
Inches	~ 2" x 1" x .5"	~ 2.75" x 1" x 1"
Millimeters	~53 x 25 x 12mm	~ 70 x 25 x 23mm

How Pinoccio compares



	Pinoccio	Arduino Mega	Arduino Uno
Open Source Hardware	Yes	Yes	Yes
Microcontroller	ATmega256RFR2	ATmega2560	ATmega328
Clock Speed	16 MHz	16 MHz	15 MHz
SRAM	32 kB	8 kB	2 kB
EEPROM	8 kB	4 kB	1 kB
Flash Memory	256 kB	256 kB	32 kB
Programmable LED's	1 RGB	1 green	1 green
Built-in Radio	Yes	No	No
LiPo Battery	Yes	No	No
On-board Sensors	Yes	No	No
Serial Lines (UART)	2	4	1
Analog Input Pins	8	16	6
Digital I/O Pins	17 - 4 with PWM	54 - 15 with PWM	14 - 6 with PWM
Dimensions	2" x 1"	4" x 2"	2.7" x 2"

FAQ

Why this price point?

We've worked hard to make Pinoccio truly affordable for what it is. It's a matter of comparison: When looking at a Starter Kit, many people expect the price point of an Arduino Uno or two – but what you actually get is like two Arduino Megas, plus:

- Built-in, rechargeable batteries
- Baked-in wireless mesh networking, without needing the Internet
- Ready-to-go WiFi connectivity
- Charging circuit and fuel gauge
- Pinoccio was created because we didn't want you (or us) to have to buy 2 Arduino Megas, 2 XBee modules (or similar radios), 2 LiPo batteries, one or two battery chargers, and a Wifi shield (plus writing the firmware and API that makes it all work together) - just to get started hacking on mesh networks. You're already ahead of the curve!

That's why we can afford to be open-source. Anyone can build their own, fully-functional Pinoccio Scouts from our schematics and layout files. (If you want a custom board, we encourage it!) However, this is the cheapest and easiest way to go about it.

How long does the rechargeable battery last?

You can use the board non-stop at full power for around 27 hours. However, it's more likely that you'll put the board to sleep, and have it wake up when certain conditions are met. Using it this way, you could have a board run on one charge for years. This can vary a lot depending on how you're using Pinoccio, but we've examined every component going into Pinoccio boards to ensure they have very low leakage and can sustain very long battery life. More on this…

Do you support shields for custom functionality?

Absolutely. We have 32 header sockets on each Pinoccio board that can accept any number of shields, or "backpacks". We will be offering some of our own backpacks (including motion sensors, environment sensors, and Bluetooth 4.0), as well as assisting third parties in developing their own shields that work with Pinoccio boards. We're also building in ways to manage the power a backpack uses, so you can program your Pinoccio board to turn off power to the backpack if it's running low. And multiple backpacks can be stacked on top of one another, for all types of crazy projects!

How do Pinoccio boards talk to one another?

Pinoccio is designed for creating wireless mesh networks. Each board has an 802.15.4 radio that communicates with any other board on its PAN ID, in a mesh configuration. Several network configurations are possible, including the traditional coordinator/router/end-node, as well as a completely decentralized peer-to-peer mesh network with routing. More on this…

How do Pinoccio boards talk to the Web?

Our starter kit includes a Wi-Fi shield that fits on top of one of the Pinoccio boards. That board and shield combo acts as a bridge between the Pinoccio mesh network and the web. More on this…

What does it mean when HQ shows that my Scout is "offline"?

When a Scout goes offline, that means it has lost its WiFi connection. The Lead Scout in this troop might be out of range of the router, or this Scout (or the Lead Scout) might be out of battery power. Scouts will stay "awake" as long as they have battery power. You can increase the Scout's range by setting its datarate low, and/or increase its battery life by reducing the signal power.

Does Pinoccio still work without WiFi access?

Yes! Pinoccio boards can act as a standalone network — with no Wi-Fi around. We spoke with several environmental scientists about how they'd use Pinoccios, and they often have sensor networks in the middle of the desert or jungle, and laying coaxial cable at a site can

cost thousands of dollars and many man-hours of digging trenches for conduit. So we wanted to have the ability to do standalone networking, and 802.15.4 mesh networking with routing fit the ticket perfectly. More on this…

Why didn't you just use WiFi for networking?

We could have simply used WiFi radios for networking, but WiFi modules tend to use a lot of battery when sending or receiving data. They typically draw up to 150mA when transmitting, and we're at just 18mA with the 802.15.4 radio. Plus, the WiFi modules are still relatively expensive, and we wanted the Field Scout to stay closer to the cost of a standalone Arduino board, not the cost of an Arduino board and a Wi-Fi shield–which is usually over $100.

What if one board in my mesh network is out of range of WiFi?

No problem. We support routing between boards, so if board A and board C are out of range of each other, but they can both reach a board B, then B will route packets for A and C to reach each other. More on this (and a helpful diagram)…

Can I use Pinoccio as an access point to provide WiFi internet to an area?

While the GainSpan WiFi module is capable of acting as an access point, we've designed Pinoccio to support its role as a sensor and control mesh network that can be connected to the Internet. This means prioritizing components and capabilities, to keep the board compact and low-cost for what it does. As such, you'll want to look into other solutions for providing WiFi Internet access to computers and mobile devices.

What is this Pinoccio API you mention?

Ooh, we're excited about this! Getting physical devices to communicate with the Web only solves part of the problem. Often, you still have to build your own support for polling or sending information with specific APIs, storing the data that your boards generate, and you need to decide on ideal protocols and architectures to maximize the efficiency of your project. This is hard, and most people don't find it fun!

We've built a full-featured REST API to do all this hard work for you. You can find more details on the Pinoccio API page.

Am I locked in to using Pinoccio's API?

No way! One idea that's really important to us is that you control your own data, not us. You don't have to use Pinoccio's API server. Program your board to connect to your own server and go nuts! We do want to make sure it's dead-easy to get your board talking to the web, and the Pinoccio API is simply the fastest and easiest way to do it.

Why did you choose this particular architecture?

The Atmel 802.15.4 radio in the ATmega256RFR2 chip we're using has a ridiculously low power draw for the range you get. We can blast the radio non-stop at full power, and a 550mAh battery will power that (and the MCU) for something like 27 hours. It only draws around 18mA at full radio transmitting power, and the MCU in the highest power state. Putting a Scout to sleep will give you much longer battery life. We wanted this sort of endurance while still remaining networkable. Thus this was the right package.

Do you plan on supporting IPv6 or 6LoWPAN?

We want the ability to easily get a Pinoccio network onto the web today, and to be able to reach it — and web/mobile apps to reach a board. So we settled on a shield/bridge setup, where it's easy to get Pinoccios online if you want to. But we see everything moving towards an IPv6 future, and we're ready for it. We have the ability to upgrade the 802.15.4 network library to support 6LoWPAN, which lets IPv6 route across an 802.15.4 network. It's still a ways out, but our grand vision is to have each Pinoccio board pingable over IPv6 from the web.

Does it work with my Raspberry Pi?

It sure does. Many people think they need to use a Raspberry Pi as the gateway between a Pinoccio mesh network and the Internet, but no need for that if you have a Pinoccio and the WiFi shield–this will do the bridging for you. However, if you want a full Linux system for your particular project, and still want the mesh networking of Pinoccio, you can attach a Pinoccio to the pins or USB port on the Raspberry Pi and get your Pi talking to the mesh. Raspberry Pi has also discussed its support for Arduino boards.

Why the name "Pinoccio"?

The name is an homage to Pinocchio, the little dude from Italian folklore who wanted to cut his strings so that he could become a real boy and explore the world. He just wanted to be wireless!

Where's the "h"?

It's a bit of a sensitive topic around here, but we'll try to give you the highlights and lowlights. After a promising career as a child actor, h blew his remaining wealth on go-karts and paddle boats. H then went on to community college, paying his own way by working part-time as a gas station clerk. Unable to ward off his demons, h struggled with an addiction to candy bars and soda. H was last seen in the Mojave Desert, hopped up on Dum-Dums and Whoppers, shooting pop bottles with a BB gun while babbling incoherently. H, if you're out there, call us, we'll pay for the bus ticket home.

Where can I go for technical support?

Always good to start with the support pages! The Pinoccio forums are also an excellent

place to look for guidance, and maybe even a little inspiration.

For industrial applications, see Filament. https://filament.com/