



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of "Quality Parts,Customers Priority,Honest Operation,and Considerate Service",our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

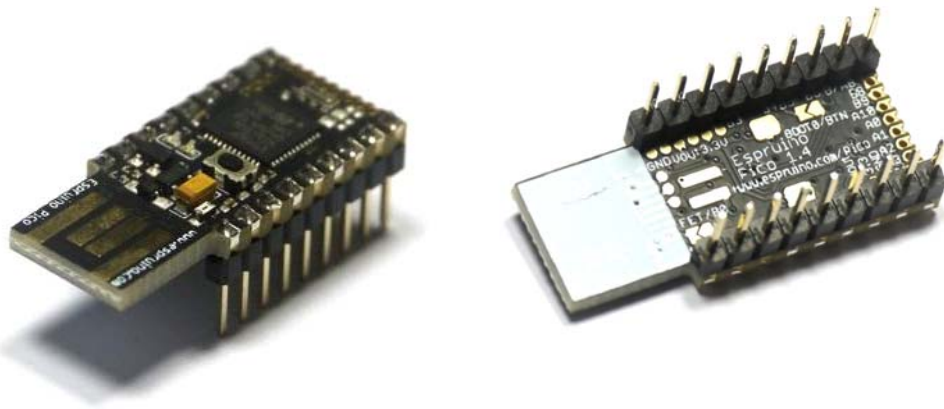
Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China



Espruino Pico

ESP001



A tiny USB stick that runs JavaScript - control things in the real world in just seconds!

The Espruino Pico is a tiny USB stick that runs JavaScript code (it doesn't need to be plugged into a computer). You can program it with nothing but a serial terminal program, but there's also an IDE that runs in the Chrome Web browser. It's got a syntax highlighted editor as well as a graphical programming language.

The IDE is so quick and easy to install that the Pico has possibly the shortest time-to-blink that we've ever seen.

Once you've uploaded code, you can inspect and change variables (including functions!) while your program is running.

There's loads of documentation, tutorials and support for a huge range of different hardware too.

See how to get started here, or if you have any questions ask away on our forums!

Not only that, but your source code is on the board itself. If you make something with an Espruino board and need to change it a year later, your original code is still there waiting for you!

Specifications

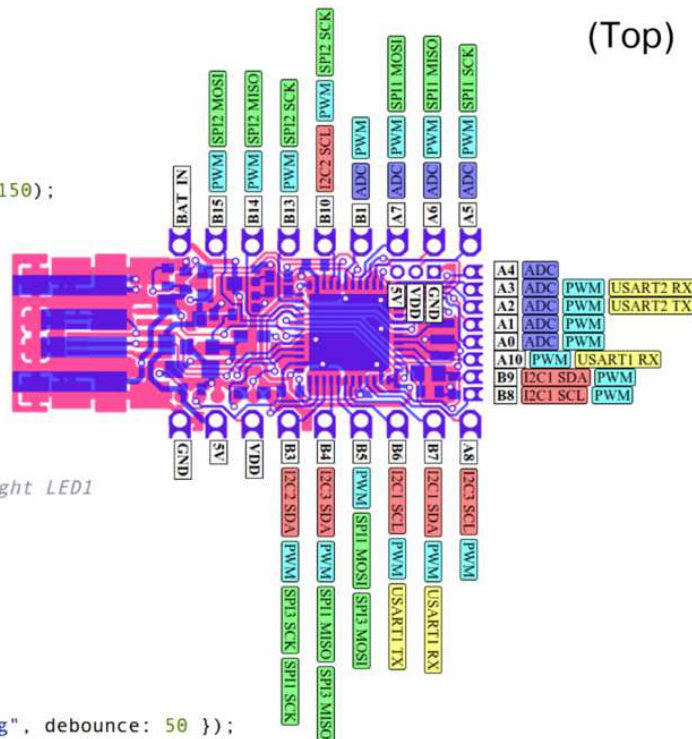
- 33mm x 15mm (1.3 x 0.6 inch)
- 22 GPIO pins : 9 Analogs inputs, 21 PWM, 2 Serial, 3 SPI, 3 I2C
- All GPIO is 5 volt tolerant (Arduino compatible)
- 2 rows of 9 0.1" pins, with a third 0.05" row of 8 pins on the end
- On-board USB Type A connector
- Two on-board LEDs and one button.
- STM32F401CDU6 CPU - ARM Cortex M4, 384kb flash, 96kb RAM
- On-board 3.3v 250mA voltage regulator, accepts voltages from 3.5v to 16v
- Current draw in sleep: < 0.05mA - over 2.5 years on a 2500mAh battery
- On-board FET can be used to drive high-current outputs

Pinned or Unpinned?

This is the pinned version of the board, which fits perfectly into breadboard

Quick Reference

```
// Light LED1
digitalWrite(LED1, 1);
// Blink LED2 for 150ms
digitalPulse(LED2, 1 /*polarity */, 150);
// Turn LED1 off after 1 sec
setTimeout(function() {
  digitalWrite(LED1, 0);
}, 1000 /* millisecs */);
// 40% duty cycle, 300Hz square wave
analogWrite(A8, 0.4, {freq:300});
// Internal pullup, read value
pinMode(B15, "input_pullup");
console.log(digitalRead(B15));
// Read analog value every 100ms, light LED1
setInterval(function() {
  var a = analogRead(A5);
  digitalWrite(LED1, a>0.5);
}, 100);
// When button is pressed
setWatch(function(e) {
  console.log("Press at "+e.time);
}, BTN, { repeat: true, edge: "rising", debounce: 50 });
```



Quick Reference

```
// Turn LED1 on
digitalWrite(LED1, 1);
// Blink LED1 for 100ms
digitalPulse(LED2, 1, 100, 150);
// Turn LED1 off after 1 sec
setTimeout(function() {
  digitalWrite(LED1, 0);
}, 1000, /* milliseconds */);
// 50% duty cycle, 300Hz square wave
analogWrite(A8, 0.5, {freq:300});
// Internal pullup, read value
pinMode(B15, "input_pullup");
console.log(digitalRead(B15));
// Read analog value every 100ms, light LED1
setInterval(function() {
  var a = analogRead(A5);
  digitalWrite(LED1, a>0.5);
}, 100);
// Button is pressed
setWatch(function() {
  console.log('Press at '+e.time);
}, BTN, {repeat: true, edge: "rising", debounce: 50 });
```

