

2009 Seeed Studio Works

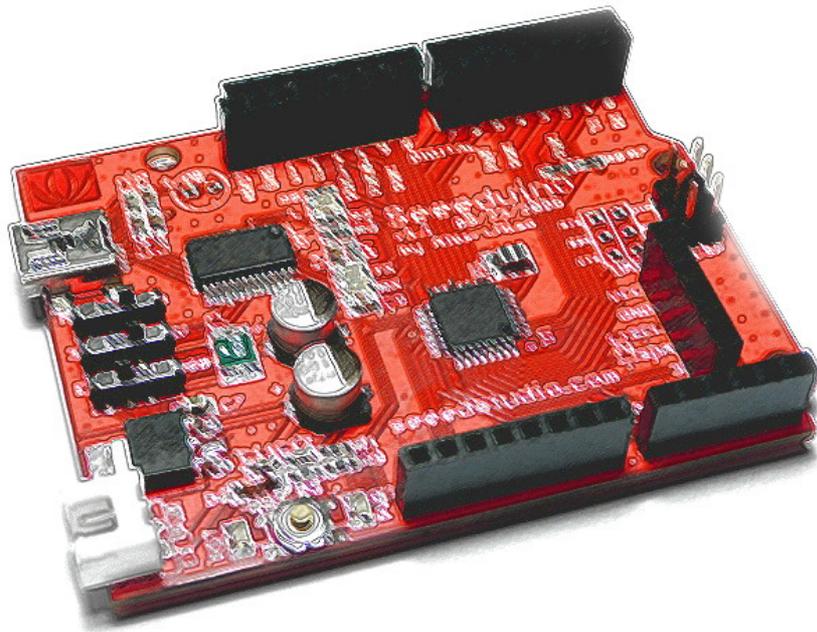
Text: Eric Pan

Photograph: X. Fan

Technical: Albert. Miao

Hello! Seeeduino

V0.9b



Hello! Seeduino



Overview

Seeduino is an Arduino compatible board. It's designed is basing on Diecimila scheme, 100% compatible to its existing program, shield and IDE. On the hardware part, remarkable changes are taken to improve the flexibility and user experience.

Words:

"I ordered the V1.0b beta version of this board and received it in 6 business days from Hong Kong to Tennessee (USA). I plugged it in, opened Arduino, changed the blink example file to 100 ms, verified, loaded, and the boards LED is flashing away as i write. Couldn't be simpler. This is an excellent implementation of the Arduino with several improvements the buyer should find very useful. I plan on ordering more stuff from Seedstudio."

Joe Pardue (aka Smiley on AVRfreaks)

Author of C Programming for Microcontrollers, and Smiley's Workshop in Nuts&Volts Magazine.

Schematic & Reference Design

EAGLE Schematic file: >>

<http://www.seeedstudio.com/depot/datasheet/seeeduino.sch>

EAGLE PCB file: >>

<http://www.seeedstudio.com/depot/datasheet/seeeduino.brd>

Summary

Microcontroller	ATmega168
Operating Voltage	5V/3.3V
Input Voltage (recommended)	7-12 V
Input Voltage (limits)	6-20 V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	8
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	16 KB (of which 2 KB used by bootloader)
SRAM	1 KB
EEPROM	512 bytes
Clock Speed	16 MHz

Photos

Available at Flickr: >> <http://www.flickr.com/photos/30537910@N07>

Features

- Inherits all of Arduino Diecimila's features.
- Compatible to Diecimila's pin layout, screw hole and dimensions.
- Evolved with SMD components
- Improved on extensibility and convenience.

How to Get Seeduino Running on Windows

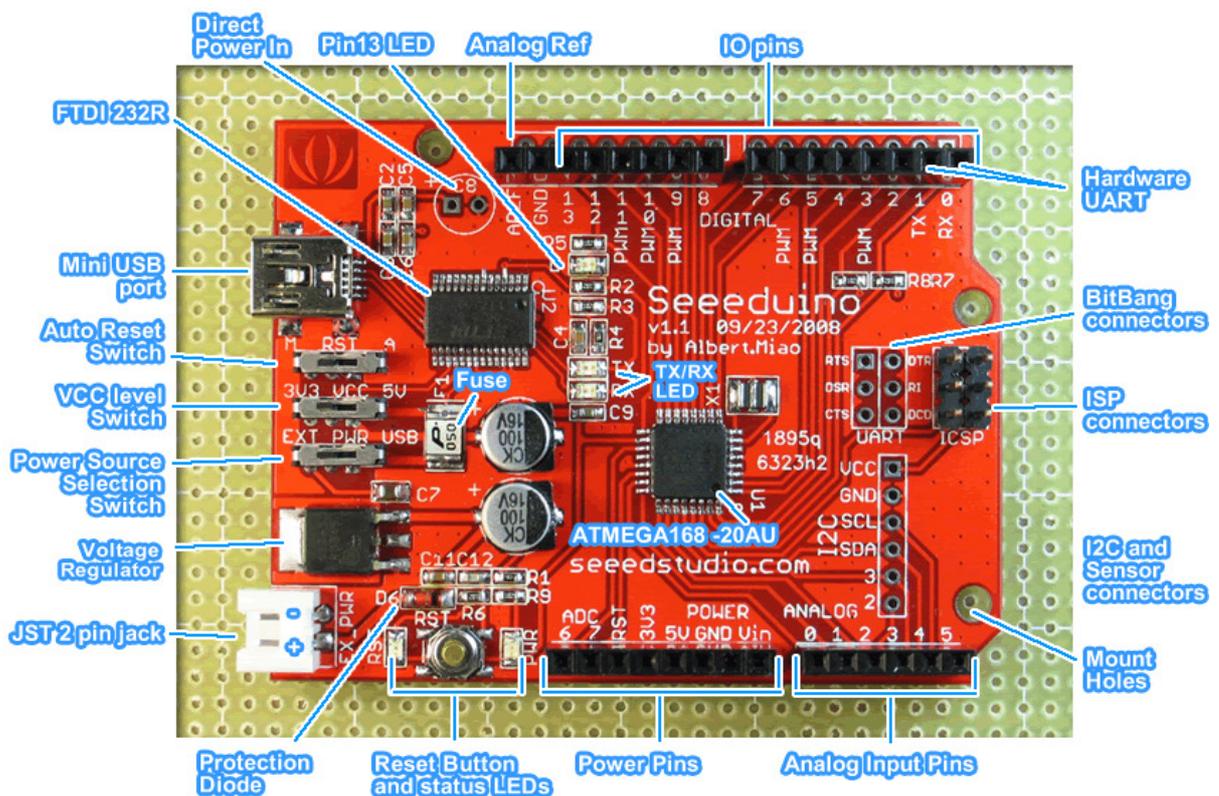
This part explains how to connect your Seeduino board to the computer and upload your first sketch; most of the contents are inherited from <http://arduino.cc/en/Guide/Windows> and under Creative Commons Attribution-ShareAlike 3.0 License.

These are the steps that we'll go through:

1. Get an Seeduino board and cable
2. Download the Arduino environment
3. Install the USB drivers
4. Connect the board
5. Run the Arduino environment
6. Upload a program
7. Look for the blinking LED
8. Start to use Seeduino

1. Get an Seeduino board and cable

In this tutorial, we assume you're using Seeduino. Inherited from Arduino, Seeduino is a simple board that contains everything you need to start working with electronics and microcontroller programming. This diagram illustrates the major components of the board.



You also need a mini USB cable (type-A to mini type-B): the kind you would connect to a USB mobile hard disk, for example.

2. Download the Arduino environment

To program the Seeeduino board you need the Arduino environment.

Download: the latest version from the download page: <http://arduino.cc/en/Main/Software>

When the download finishes, unzip the downloaded file. Make sure to preserve the folder structure. Double-click the folder to open it. There should be a few files and sub-folders inside.

3. Locate the USB drivers

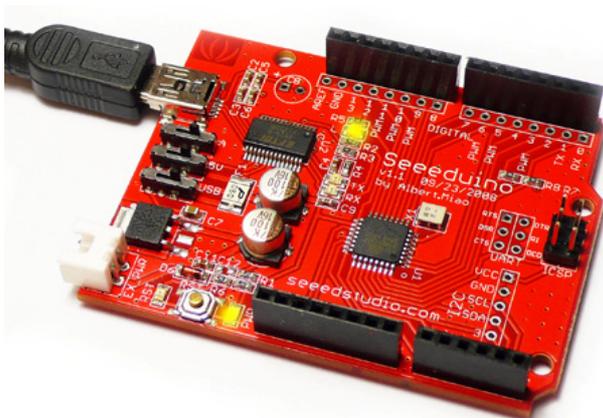
You will need to install the drivers for the FTDI chip on the board. These can be found in the drivers/FTDI USB Drivers directory of the Arduino distribution. In the next step ("Connect the board"), you will point Window's Add New Hardware wizard to these drivers.

The latest version of the drivers can be found on the FTDI website:
<http://www.ftdichip.com/Drivers/VCP.htm>

4. Connect the board

On the Seeeduino, the power source is selected by the PWR switch between the USB and power plugs. To power the board from the USB port (good for controlling low power devices like LEDs), toggle the switch towards the chip side. To power the board from an external power supply (6-12V), toggle the switch towards outside. Other two switches would be toggled towards chip for this startup tryout.

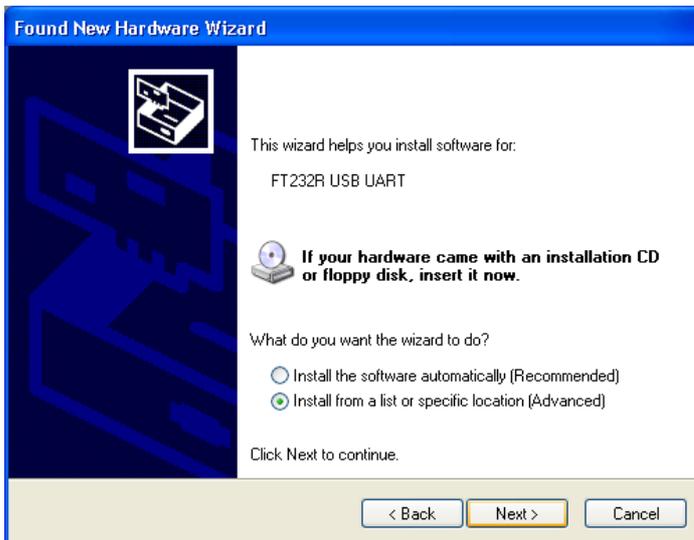
The power LED should be on.



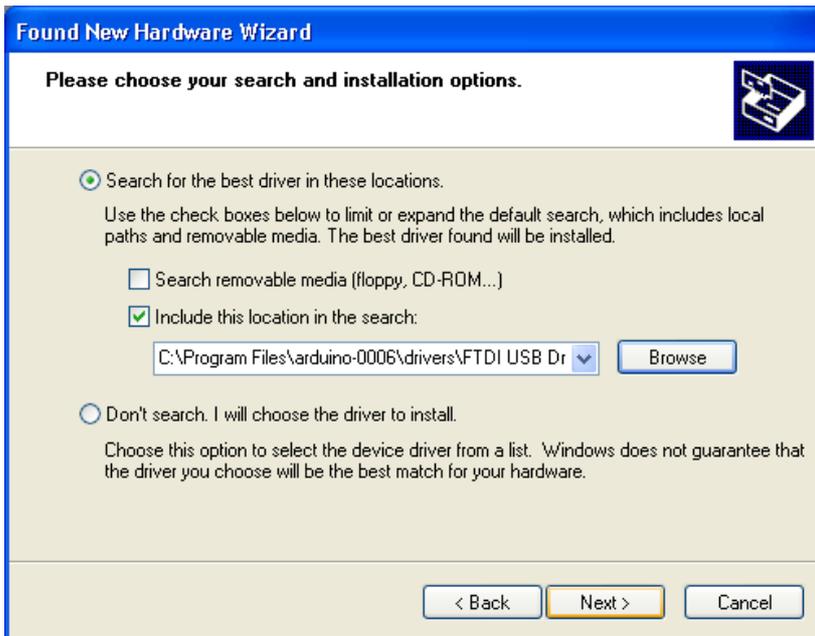
The Add New Hardware wizard will open. Tell it not to connect to Windows update and click next.



Then select "Install from a list or specified location (Advanced)" and click next.



Make sure that "Search for the best driver in these locations is checked"; uncheck "Search removable media"; check "Include this location in the search" and browse to the location you unzipped the USB drivers to in the previous step. Click next.



The wizard will search for the driver and then tell you that a "USB Serial Converter" was found. Click finish.



The new hardware wizard will appear again. Go through the same steps. This time, a "USB Serial Port" will be found.

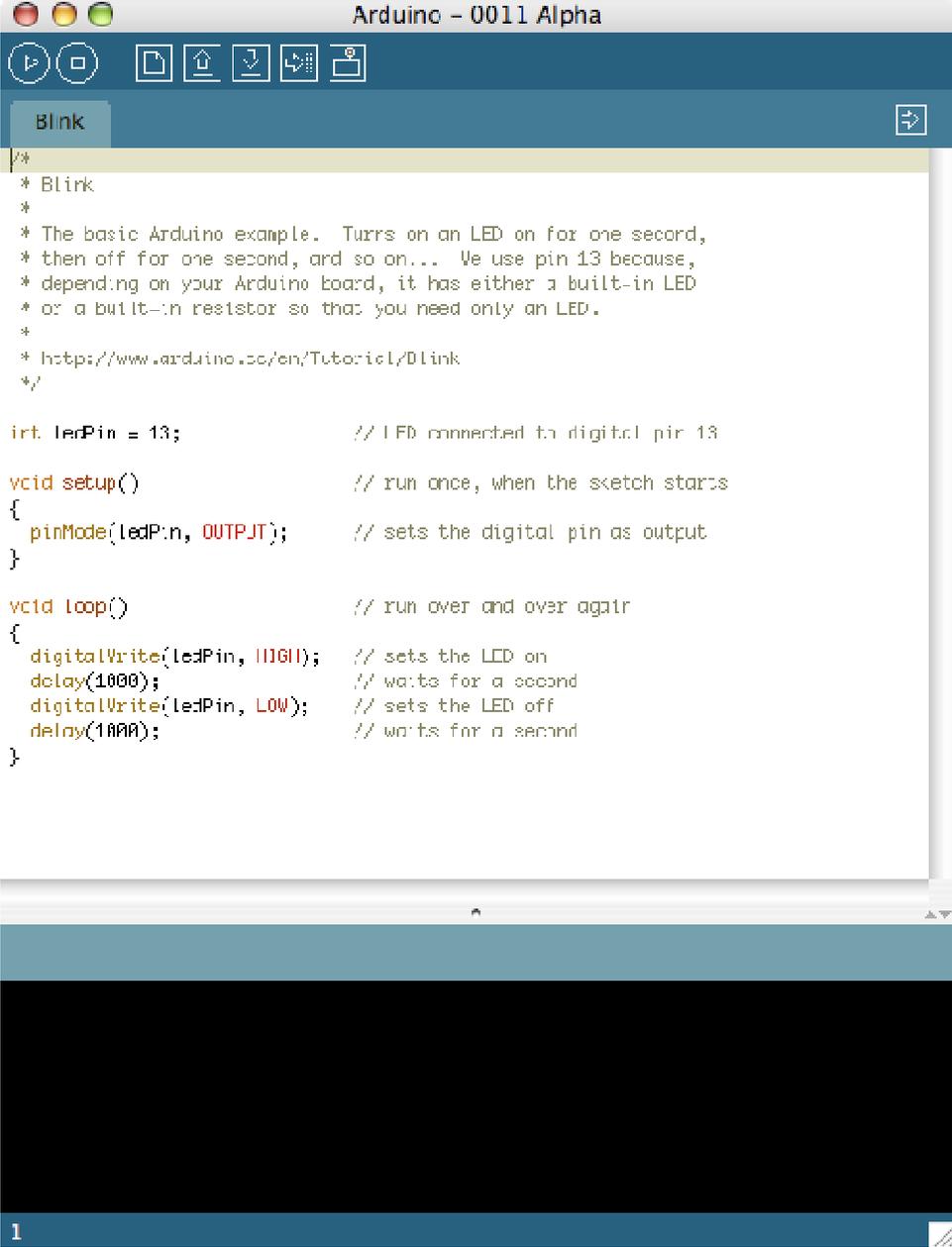
5. Run the Arduino environment

Open the Arduino folder and double-click the Arduino application.

6. Upload a program

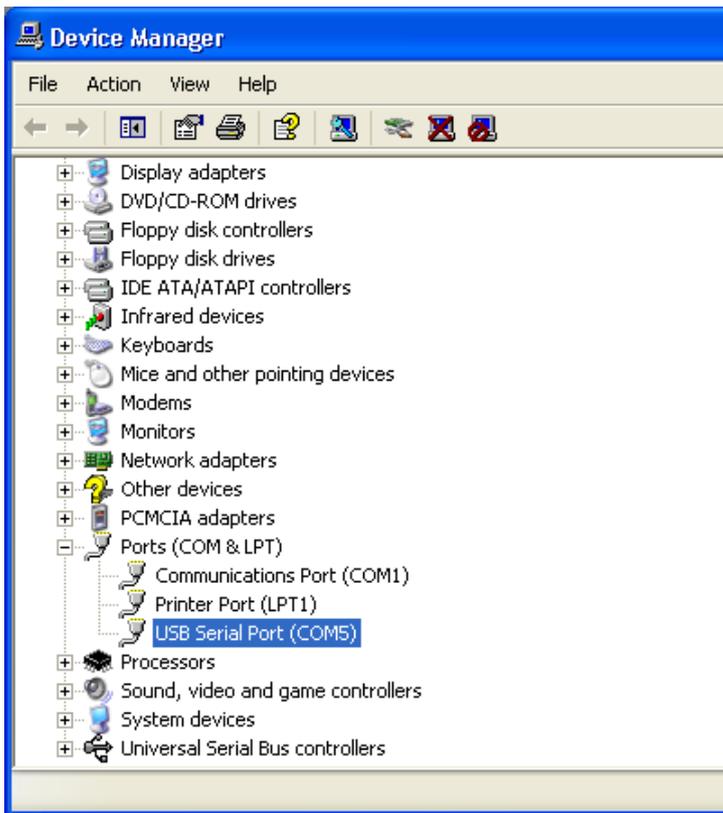
Open the LED blink example sketch: **File > Sketchbook > Examples > Digital > Blink.**

Here's what the code for the LED blink example looks like.

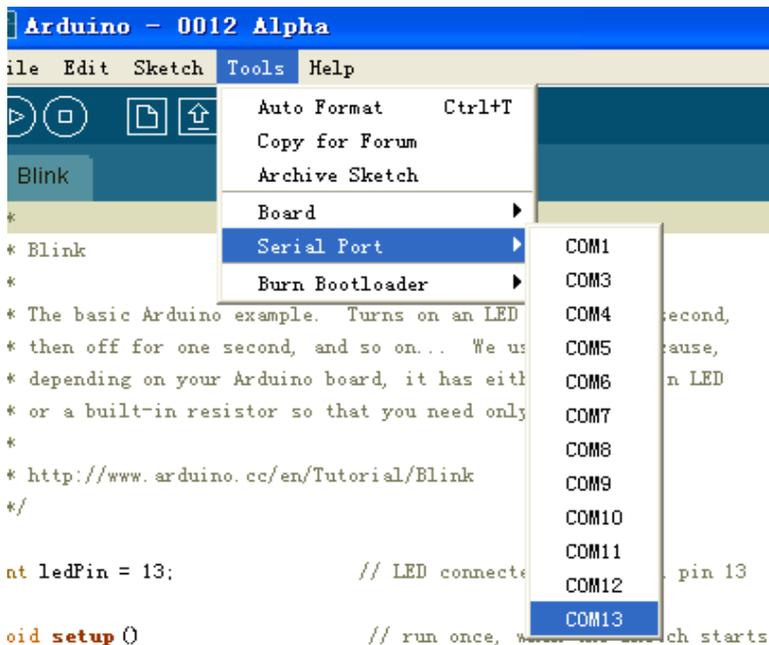
A screenshot of the Arduino IDE interface. The window title is "Arduino - 0011 Alpha". The menu bar includes "File", "Edit", "Tools", "Sketch", "Help", and "Window". The toolbar contains icons for running, stopping, saving, opening, and uploading. The main editor area shows the code for the "Blink" sketch. The code is as follows:

```
/*  
 * Blink  
 *  
 * The basic Arduino example. Turns on an LED on for one second,  
 * then off for one second, and so on... We use pin 13 because,  
 * depending on your Arduino board, it has either a built-in LED  
 * or a built-in resistor so that you need only an LED.  
 *  
 * http://www.arduino.cc/en/Tutorial/Blink  
 */  
  
int ledPin = 13;           // LED connected to digital pin 13  
  
void setup()              // run once, when the sketch starts  
{  
  pinMode(ledPin, OUTPUT); // sets the digital pin as output  
}  
  
void loop()               // run over and over again  
{  
  digitalWrite(ledPin, HIGH); // sets the LED on  
  delay(1000);                // waits for a second  
  digitalWrite(ledPin, LOW);  // sets the LED off  
  delay(1000);                // waits for a second  
}
```

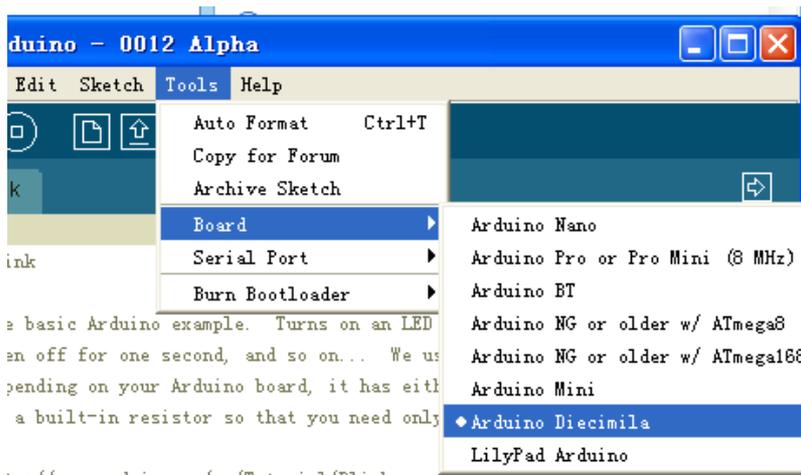
The status bar at the bottom left shows the line number "1".



Select the serial device of the Seeeduino board from the Tools | Serial Port menu. On Windows, this should be COM3, COM4, or bigger # for a USB board. To find out, open the Windows Device Manager (in the Hardware tab of System control panel). Look for a "USB Serial Port" in the Ports section; that's the Seeeduino board.



Make sure that "Arduino Diecimila" is selected in the **Tools > Board** menu.



Now, simply click the "Upload" button in the environment. Wait a few seconds - you should see the RX and TX leds on the board flashing. If the upload is successful, the message "Done uploading." will appear in the status bar.



7. Look for the blinking LED

A few seconds after the upload finishes, you should see the amber (yellow) LED on the board start to blink. If it does, congratulations! You've gotten Arduino up-and-running.

If you have problems, please see the [troubleshooting suggestions](#).

8. Learn to use Seeeduino/Arduino

- <http://arduino.cc/en/Tutorial/HomePage> : try these example programs.
- <http://arduino.cc/en/Reference/HomePage> : read the reference for the Arduino language.

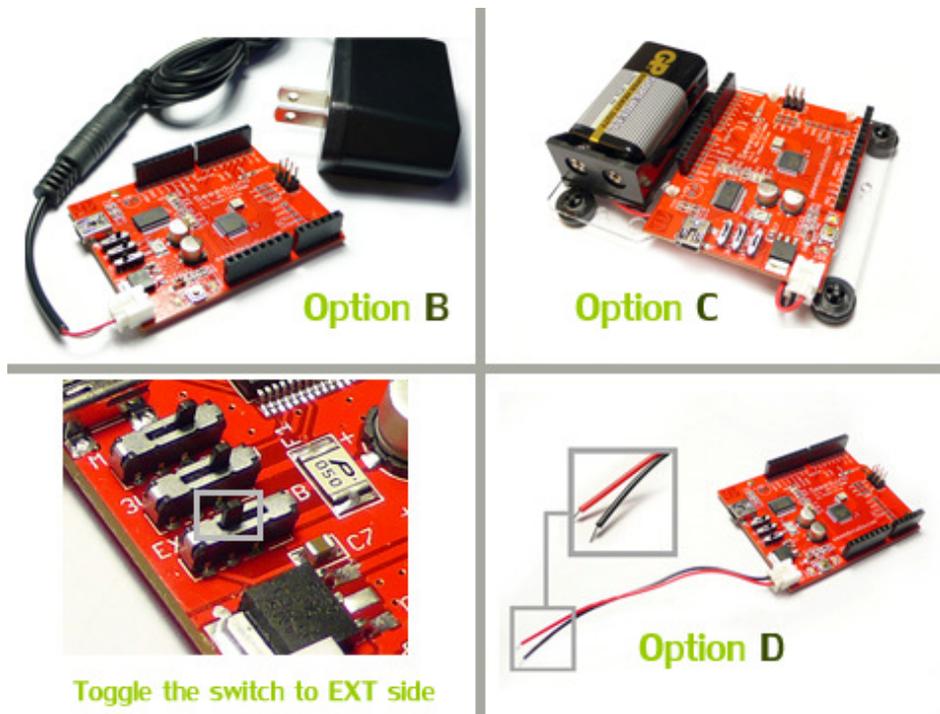
Seeeduino features tour

1. How would it be powered up?

Seeeduino has two basic ways of power input: **USB** and **external power jack**. You may use a mini USB cable for downloading and debugging (Option A), and our JST 2pin power jack would provide you more options when going field. (Option B-D)



Option A – USB cable power



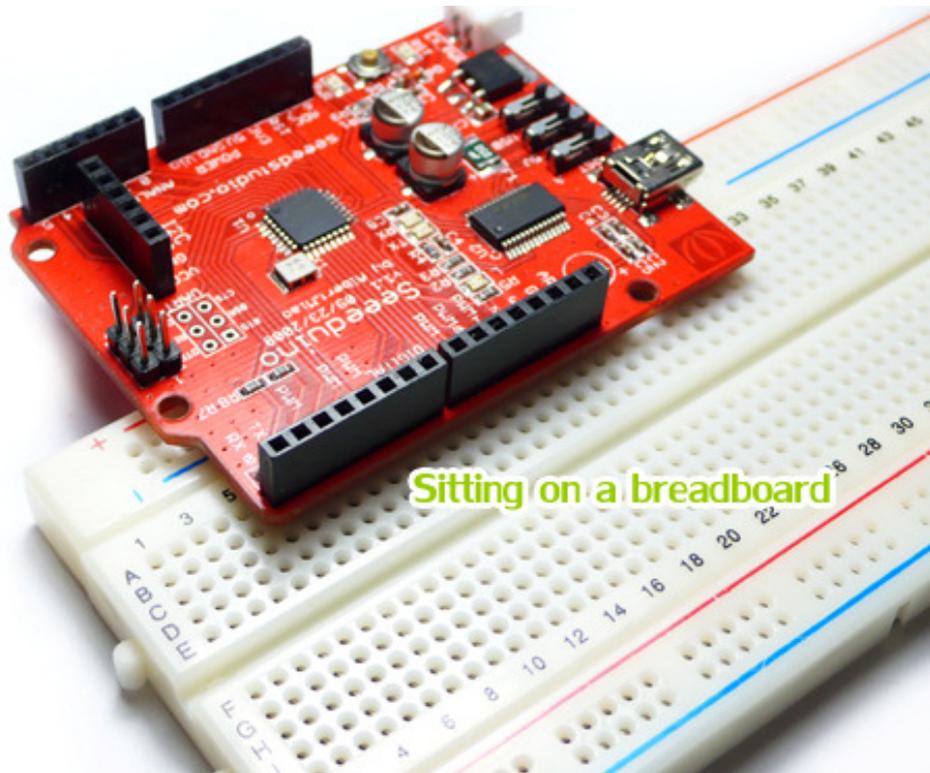
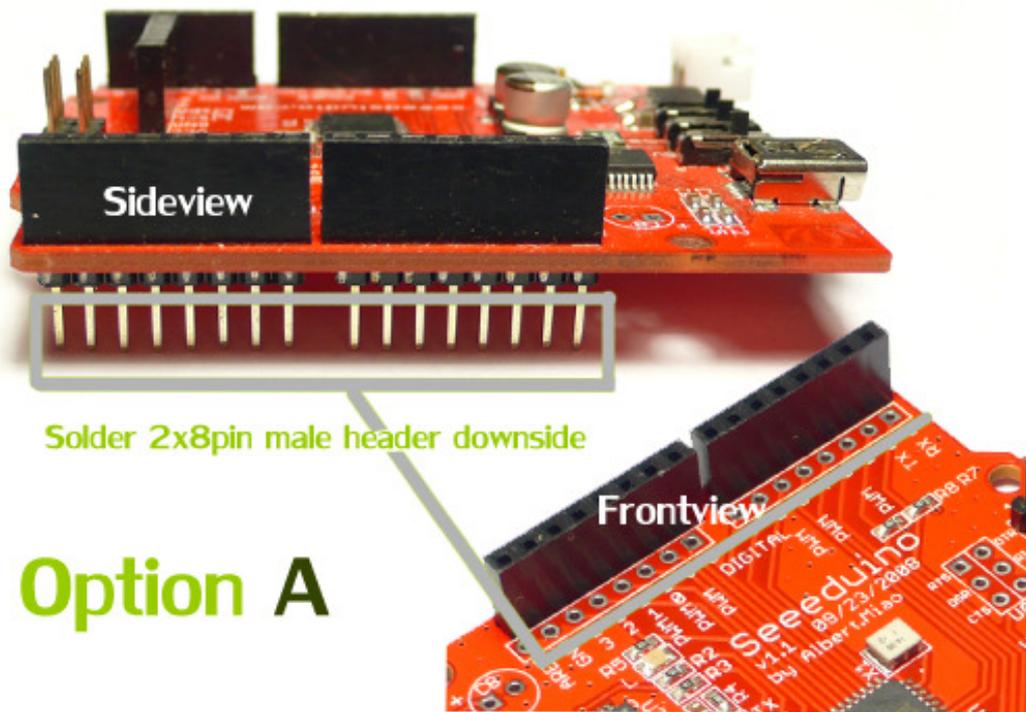
Option B – 2pin JST to 2.5 mm plug Converter for power adapters.

Option C – Harness for all-in-one mobile usage.

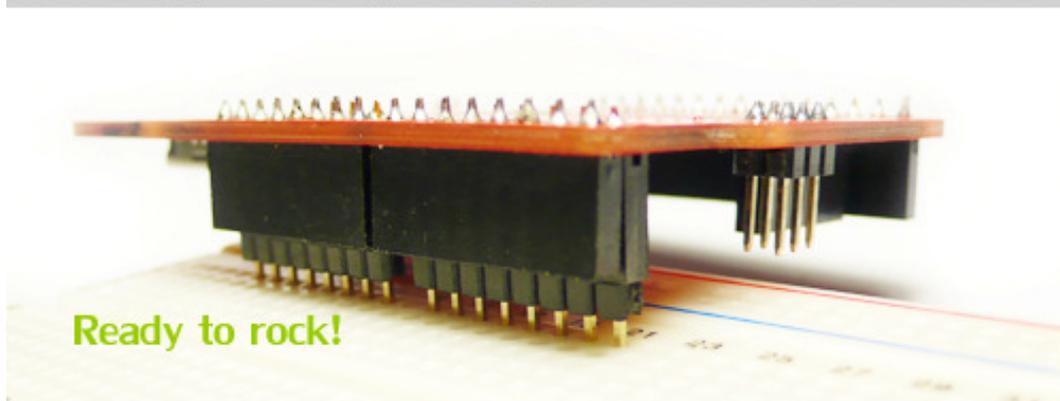
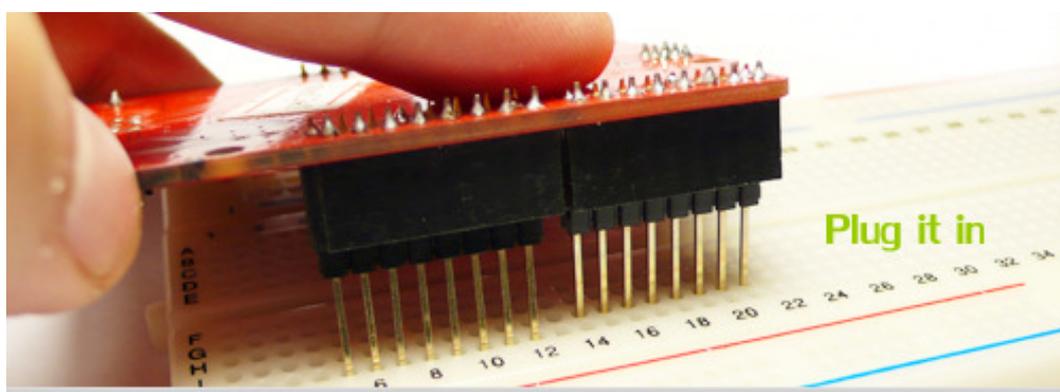
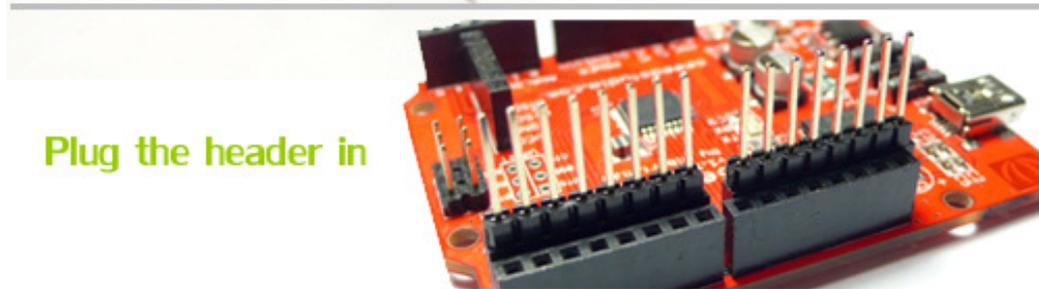
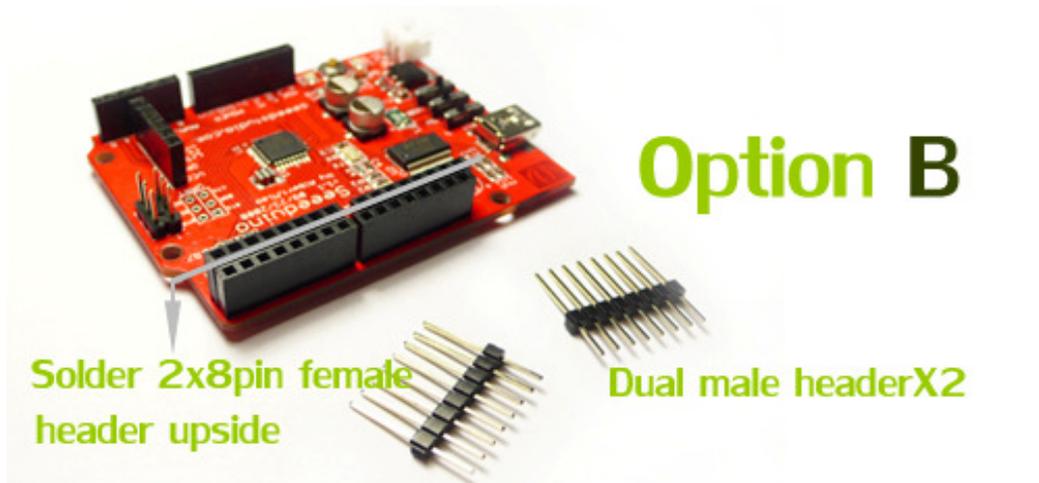
Option D – 2pin JST connector for other alternatives.

2. Easy breadboard-ing

By adding an extra row of headers, Seeeduino I/O pins are now aligned with 2.54mm (100mil) grid. You may mount it to breadboard and prototyping board easily. Here are some demonstrations about sitting Seeeduino onto a breadboard.

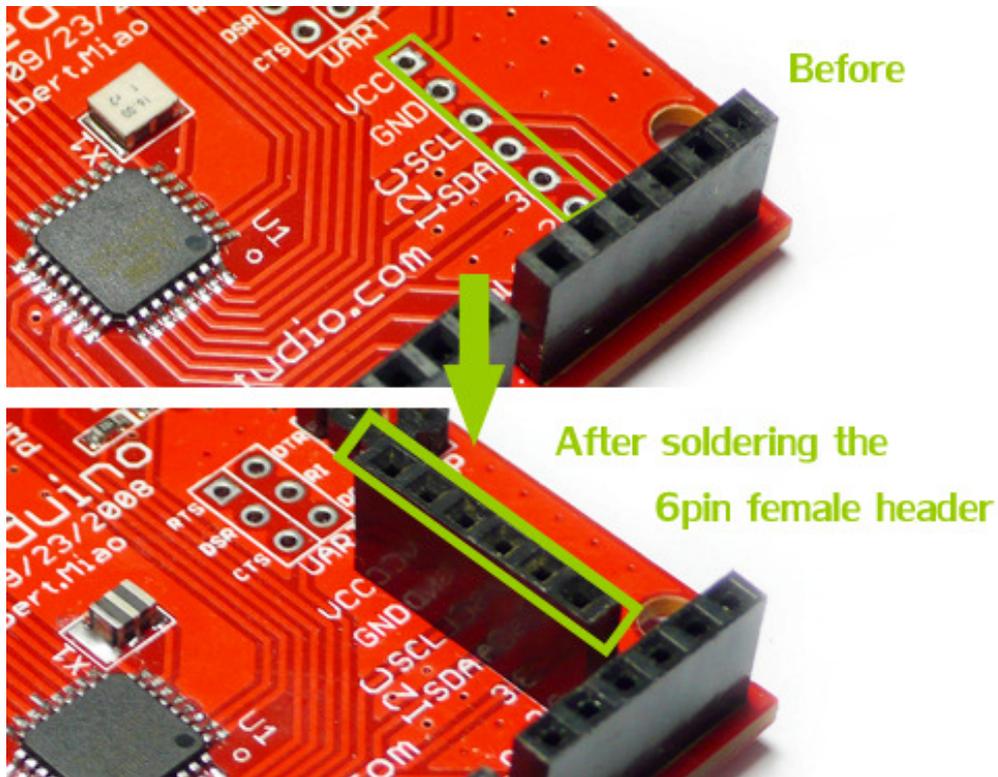


If you find soldering an extra row of header on the bottom is not making things comfortable, you may do the other way round:

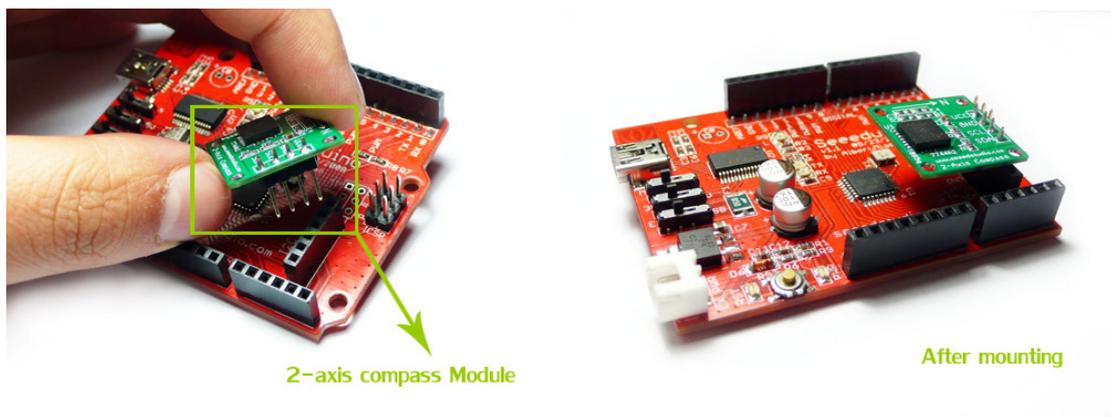


3. I2C and sensor connection

Seeeduino has prepared an extra row of I2C connections, which was reserved to reduce complexity when shipped out. You may solder it and get ready to use in minutes!



Vcc/GND/SCL/SDA for I2C, 3 and 2 are wired to Analog input pin 3/2.



Example of mounting an i2C interface 2-axis compass module, no extra wiring needed.

4. Bit bang – Burn Boot loader via USB

FT232RL is an USB-Serial bridge on an Arduino Diecimila/NG/Duemilanove PCB. It has the function to manipulate each signal pin directly. It's called [BitBang Mode](#).

"**avrdude-serjtag**" is AVR-Writer software (avrdude + BitBang-patch) developed by [Mr.Suz](#). It include the function to control FT232RL BitBang Mode.

If we use "avrdude-serjtag" we can burn the bootloader by Diecimila itself.

Mr. Kimio Kosaka has a wonderful guide describe below the method:

http://www.geocities.jp/arduino_diecimila/bootloader/index_en.html

Modify the Seeeduino

Prepare Seeeduino for BitBang mode:

Seeeduino's bit bang pins are labeled according to config file of AVRDUDE. If you are starting from Mr. Kimio Kosaka's tutorial, the mapping is like this:

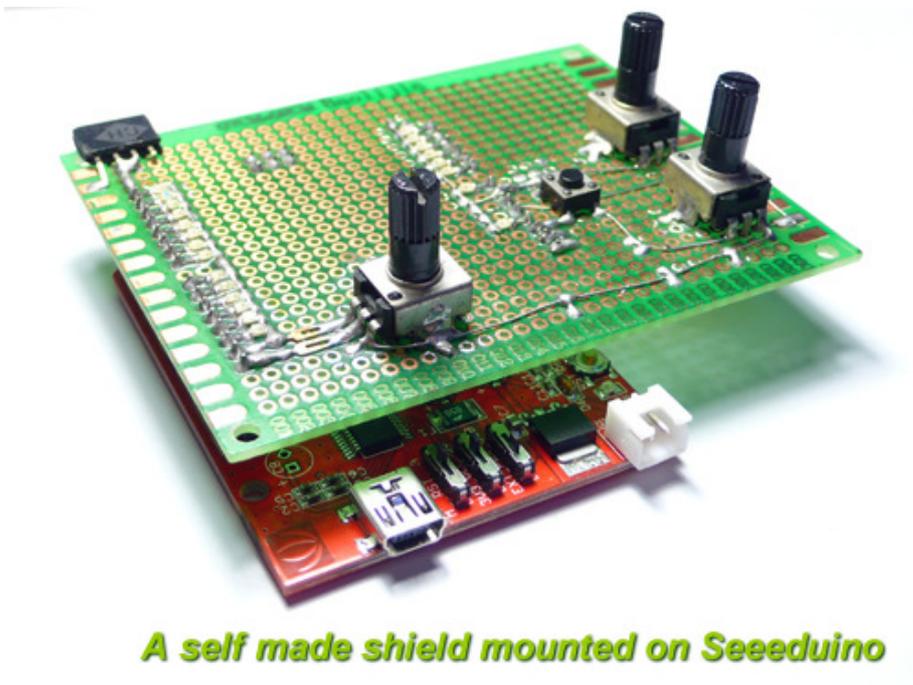
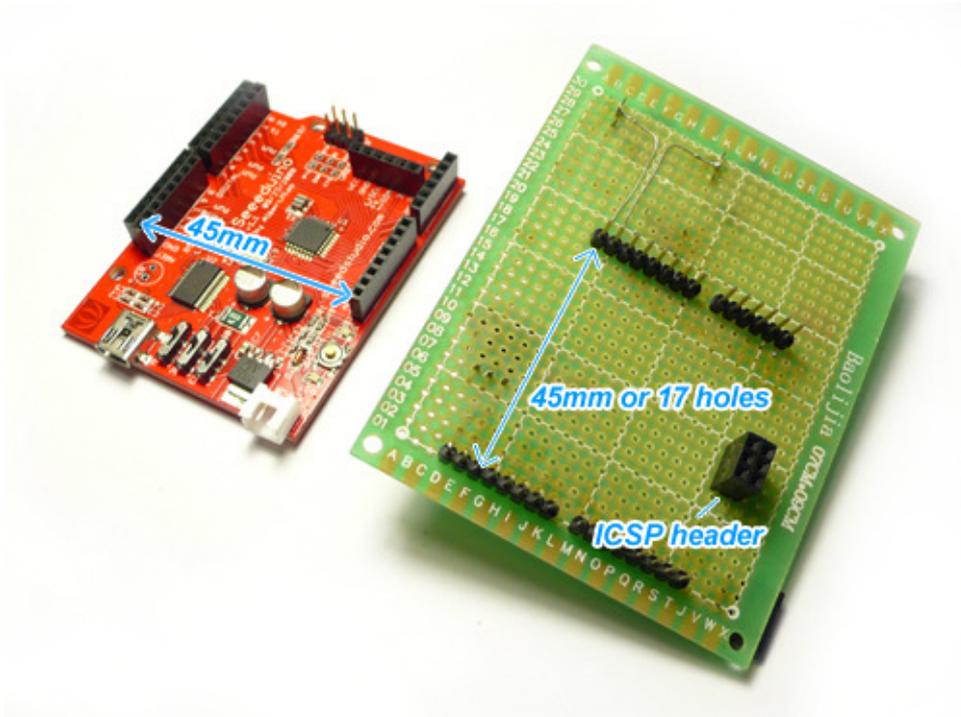
Seeeduino UART		Arduino x3
CTS	->	1
DSR	->	2
DCD	->	3
RI	->	4



Application: You may upgrade or restore Seeeduino's bootloader by this method, please be caution about the possible risk in mistaken fusing. Strictly follow the guide or READ related documents before you try this out.

5. Make your own prototyping shield

You may solder your prototyping shield easily now, the only requirement is making sure the two male headers on the proto-Shield are 45mm or 17 holes away. Also ICSP is easy to be adapted.



A self made shield mounted on Seeeduino

6. Pals - Deluxe Starter Kit, shields, and many upcoming

Please visit our depot for the latest update:

<http://www.seeedstudio.com/depot/arduino-related-c-27.html>

All accessories mentioned above are listed as routine stock.

Distributors and education organizations

Distributors and education organization policies could be found here:

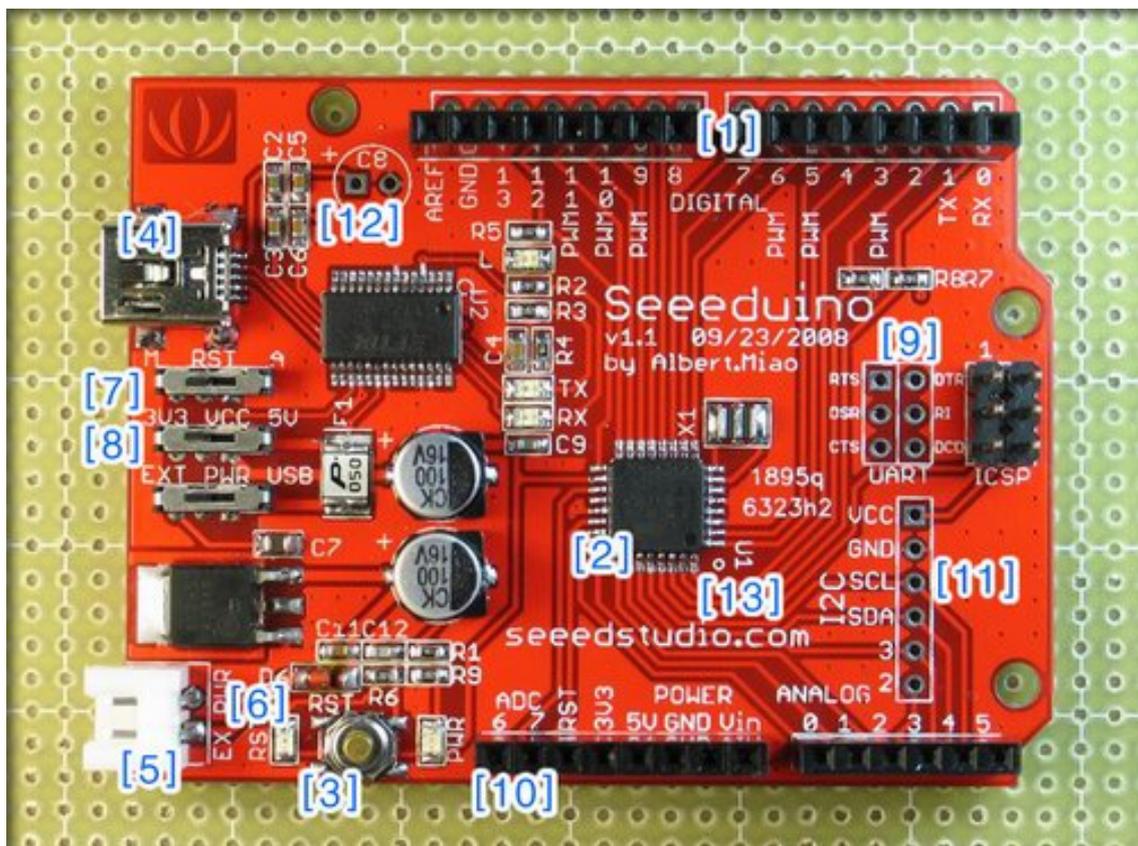
http://www.seeedstudio.com/blog/?page_id=262

Links and References

- Seeed Studio Site: www.seeedstudio.com
- Arduino Site: www.arduino.cc
- Official guide for trouble shooting references: <http://www.arduino.cc/en/Guide/Troubleshooting>
- Getting started with Arduino: <http://arduino.cc/en/Guide/Windows>
- Kimio Kosaka Guide on Arduino bitbang uage:
http://www.geocities.jp/arduino_diecimila/bootloader/index_en.html

What are the changes from Arduino to Seeeduino?

- Duplicate digital IO to 100mil grid for prototype board compatibility.
- Shrink components height below female headers.
- Easier access to buttons and switches.
- Change Type-B USB port to Mini USB.
- Replace 3.5mm DC power Jack to battery 2 Pin plug.
- Reset and power indicator near RST button.
- Auto-reset selection.
- 3.3V Operating Voltage selection.
- Pin out UART for FTDI232 bit-bang operation.
- Pin out 2 extra ADC.
- Optional capacitor for improving 3.3V output performance.
- Re-route PCB
- Pin out for I2C and sensors



Seeeduino Front View

1. Duplicate digital IO to 100mil grid for prototype board compatibility.

Adding an second row of header align to standard 100mil grid, so it will be compatible both to existing shield and to a cheap 100mil grid prototype board. By default, the new IO is soldered instead or traditional headers, and you may add the traditional header yourself.

2. Shrink components height below female headers.

Seeeduino is made as flat as possible, even the crystals are SMD version. So we may put your soldered prototype board onto it with less worry.

3. Easier access to buttons and switches.

All buttons or switches are moved nearer to the edges, so we may reach them easier under the shield or from the wire forests.

4. Change Type-B USB port to Mini USB.

Consideration for dimensions.

5. Replace 3.5mm DC power Jack to battery 2 Pin plug.

Soldering the 3.5mm DC power jack is usually the most difficult part, and we think it's too giant. Maybe a battery pack is more convenient than power adapters? Don't worry about abandoning your 12V adapter, we have a simple converter.

6. Reset and power indicator near RST button.

We use one LED for power indicator and the other for reset (it will also lid when downloading programs).

7. Auto-reset selection.

Auto-reset is a great feature loved by many. But sometimes we would also like to disable it in production-like mode.

8. 3.3V Operating Voltage selection.

The Vcc could be selected by a switch, for making life easier dealing 3.3v devices.

9. Pin out UART for FTDI232 bit-bang operation.

At least you can burn bootloader without an ISP cable now (Plz refer to Kimio's detail tutor: http://www.geocities.jp/arduino_diecimila/bootloader/index_en.html)

10. Pin out 2 extra ADC.

Why not? There are spaces :-)

11. Pin out for I2C and sensors.

Easier connection to I2C and sensors by a bended female header.

12. Optional capacitor for improving 3.3V output performance.

This is just another trial, we found in the FTDI manual about the capacitor used for improve 3.3v output performance. Although many Arduino, Freeduino, and other -duinos have proven it's just optional, we decide to keep an option to check the difference.

13. Reroute PCB

SMD version Atmega168 give us more space to route, so we carefully rearranged them for us to see the flow more clearly.

Please visit the supporting thread for suggestions on next revision.

<http://www.seeedstudio.com/forum/viewtopic.php?f=4&t=11>

New for v1.1

- Fixed bug - unnecessary via on Pin4 wire.
- Fixed bug - Label mistakenly covered by switches.
- Added I2C and sensor port.
- Added an 100uF capacitor for vcc switch.
- Grouped new 100mil grid pins.
- Replaced reset buttons with metal ones.
- Rearranged texts and labels for clearer indications.
- ROHS compliant

License & Credit

This is the first open source board from our studio, we are proud to name it **Seeeduino**. However, it is never going to be built without the great community. Thanks for everybody interested, offered suggestions, guidance or lift their concerns. Original Link:

<http://www.arduino.cc/cgi-bin/yabb2/YaBB.pl?num=1220573022/0> .



Seeeduino is licensed under a [Creative Commons Attribution 3.0 Unported License](http://creativecommons.org/licenses/by/3.0/). Based on a work at www.arduino.cc.

