

Tindduino Documentation

For Tindduino Build v2.0 or above
Support Tindduino ver 3.2+



Tindduino

Small yet Powerful

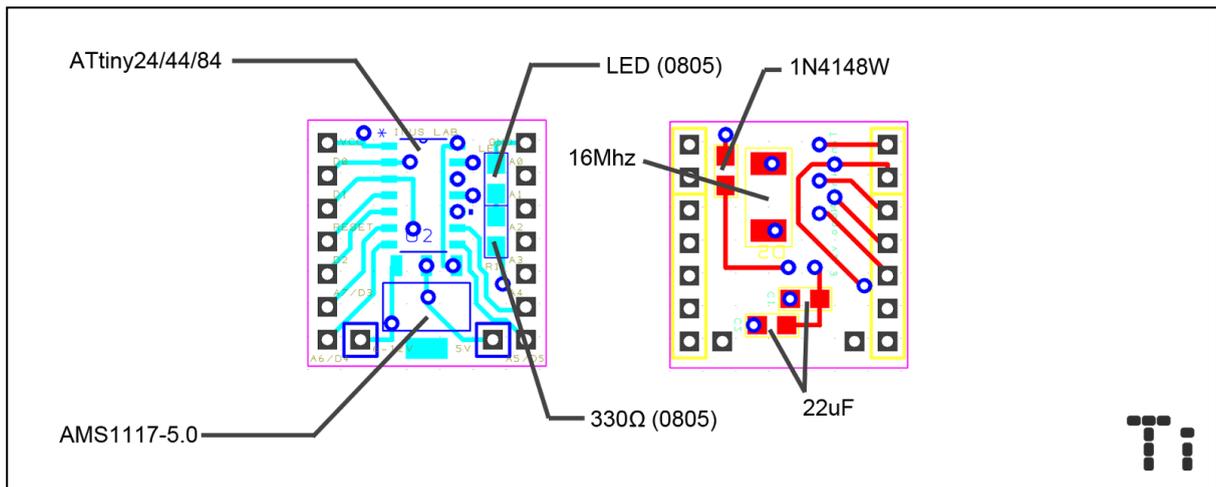
Copyright © 2017 by IMUS Laboratory

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical reviews and certain other noncommercial uses permitted by copyright law.

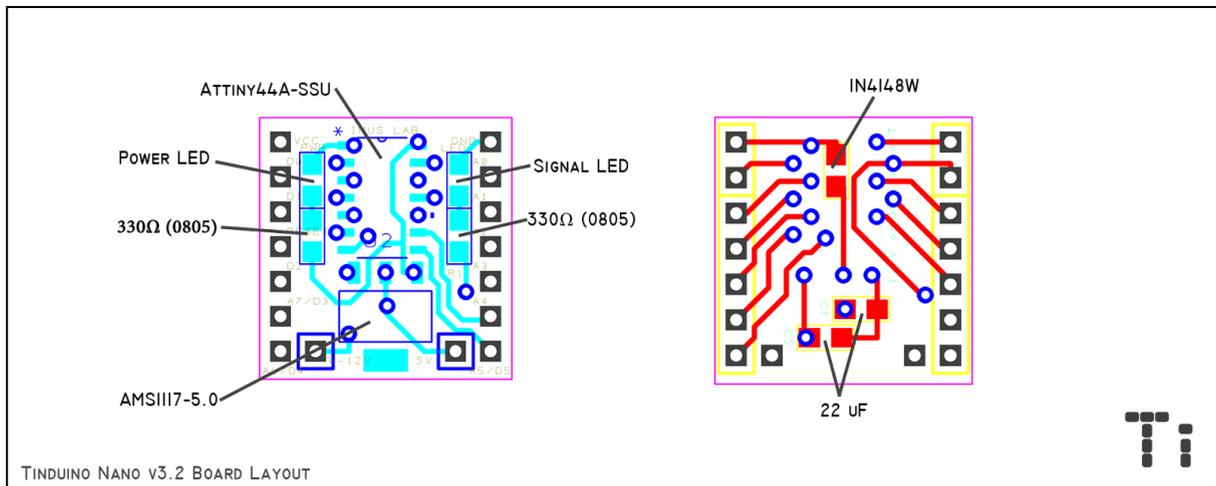
Hardware

The Tinduno is built in compatible with all ATtiny x4 series. Include but not limited to ATtiny24, ATtiny44 and ATtiny84. The hardware component for Tinduno has the following specification.

For Tinduno v2.0 to 2.3, the board layout is as follow.



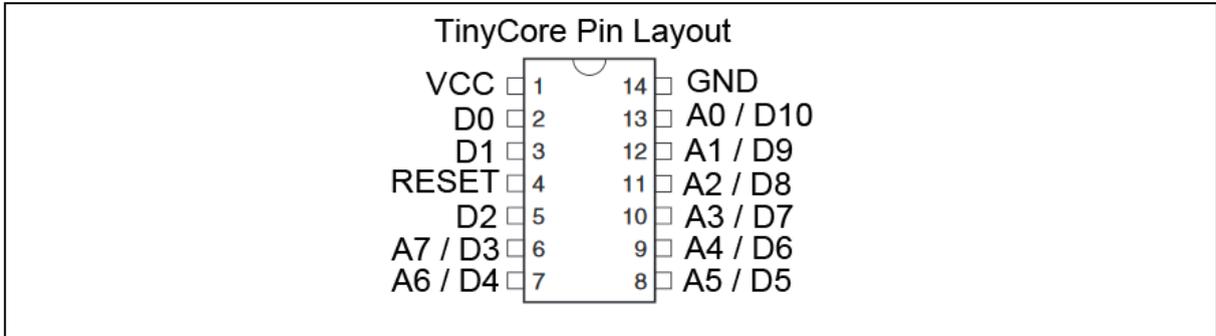
For Tinduno v3.0 to 3.2, the board layout is as follow.



Pin configuration (Provided by Atmel)

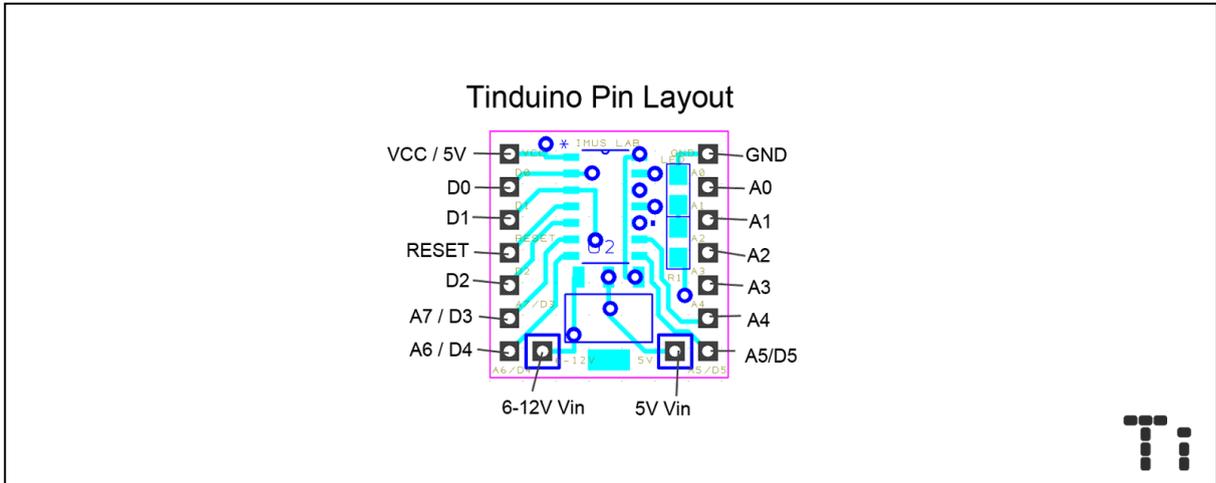
PDIP/SOIC			
VCC	1	14	GND
(PCINT8/XTAL1/CLKI) PB0	2	13	PA0 (ADC0/AREF/PCINT0)
(PCINT9/XTAL2) PB1	3	12	PA1 (ADC1/AIN0/PCINT1)
(PCINT11/RESET/dW) PB3	4	11	PA2 (ADC2/AIN1/PCINT2)
(PCINT10/INT0/OC0A/CKOUT) PB2	5	10	PA3 (ADC3/T0/PCINT3)
(PCINT7/ICP/OC0B/ADC7) PA7	6	9	PA4 (ADC4/USCK/SCL/T1/PCINT4)
(PCINT6/OC1A/SDA/MOSI/DI/ADC6) PA6	7	8	PA5 (ADC5/DO/MISO/OC1B/PCINT5)

ATtiny Pin Config (Modified base on TinyCore)



Tindduino Pin Layout

Tindduino firmware is modified from TinyCore. The pin layout of the Tindduino Nano is similar to the TinyCore with the following pin configuration.



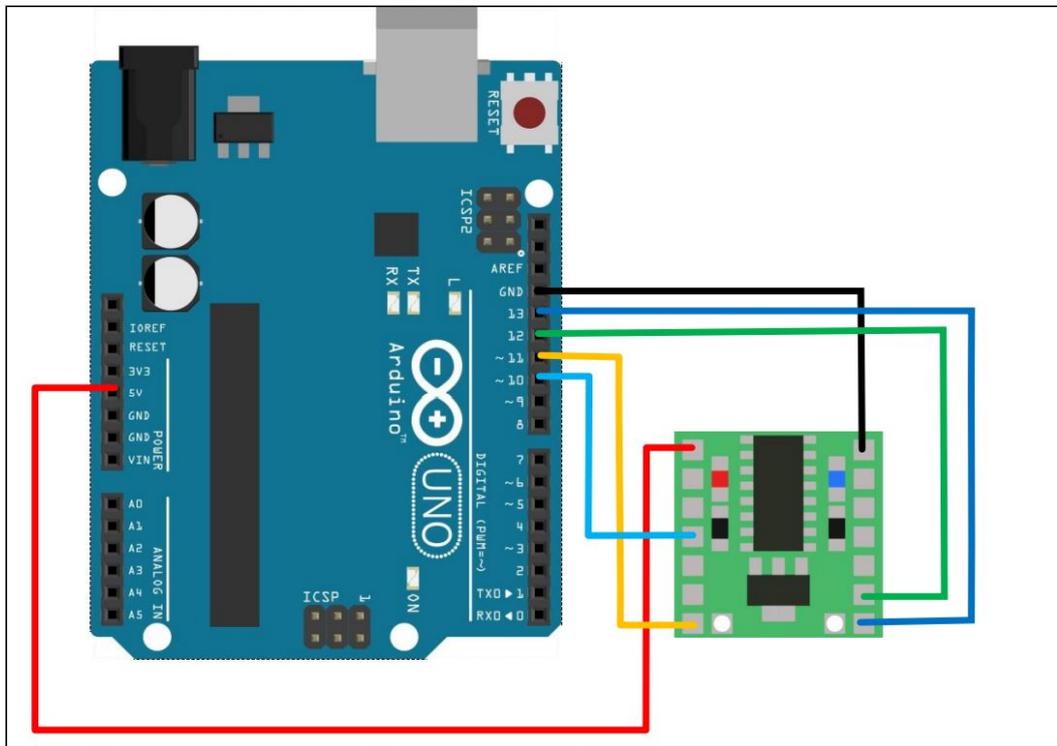
Specification

-  AVR ATTINY44A-SSU @ 8MHZ (INTERNAL CRYSTAL)
 -  4 KBYTES OF IN-SYSTEM, SELF-PROGRAMMABLE FLASH PROGRAM MEMORY
 -  256 BYTES OF IN-SYSTEM PROGRAMMABLE EEPROM
 -  256 BYTES OF INTERNAL SRAM
 -  5V PIN INPUT: 3.3 - 5.5V
 -  12V PIN INPUT: 6V - 12.2V
- 

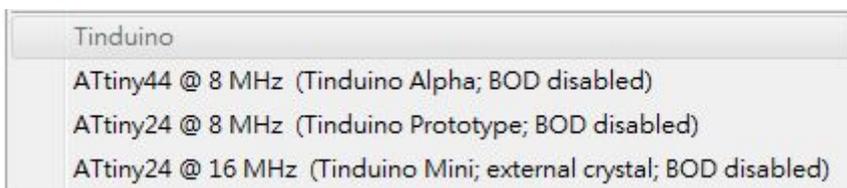
Manual Programming

Programing via Arduino ISP

1. Upload Arduino ISP to Arduino Uno or compatible Arduino
2. Connect the Tinduino to Arduino following the diagram below.



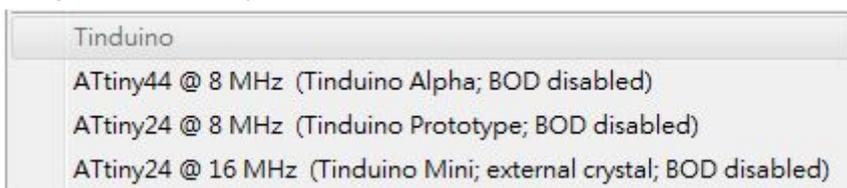
3. Select "Tindduino" from boards and select the suitable version of Tindduino you are using.



4. Upload the sketch to the Arduino that connected to Tindduino with Arduino as ISP.

Programming via AVR Programmer (USBASP)

1. Select "Tindduino" from boards and select the suitable version of Tindduino you are using and connect your Tindduino to USBASP.

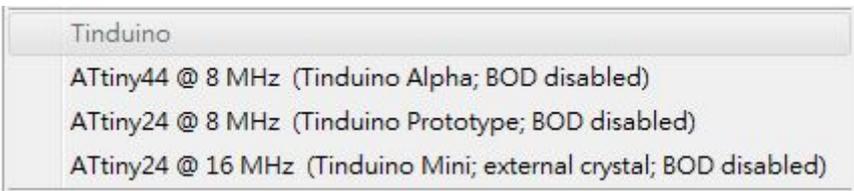


2. Compile the sketch and browse to the tmp folder which your Arduino IDE store your current sketch.
3. Copy the file with the extension “hex” to a folder which you store the “ino” sketch file as backup. (This step is not mandatory)
4. Upload the sketch to your Tindduino using USPASP option in Arduino IDE

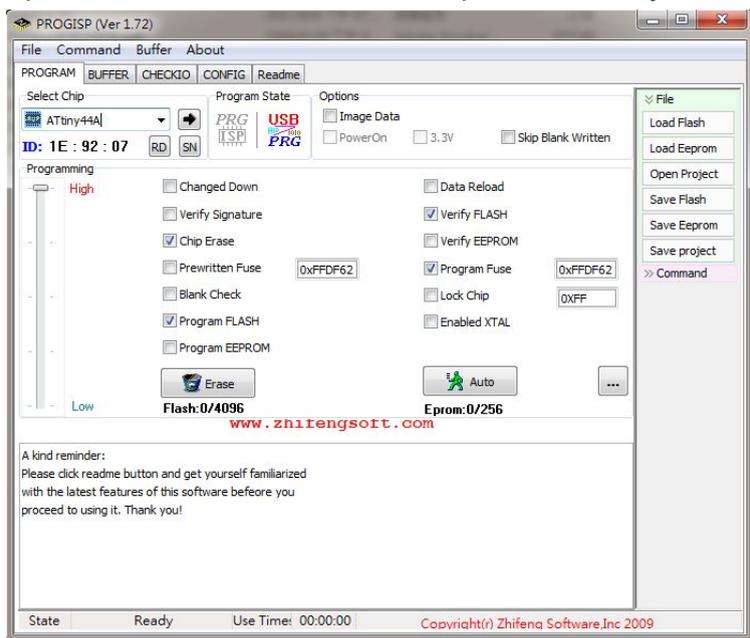
*For drivers of USBASP, please visit <http://www.fischl.de/usbasp/>

Programming via AVR Programmer (USBISP)

1. Select “Tindduino” from boards and select the suitable version of Tindduino you are using and connect your Tindduino to the USBISP.



2. Compile the sketch and browse to the tmp folder which your Arduino IDE store your current sketch.
3. Copy the file with the extension “hex” to a folder which you store the “ino” sketch file.
4. Open PROGISP.exe, select the chips number on your Tindduino. (e.g. ATtin44A)

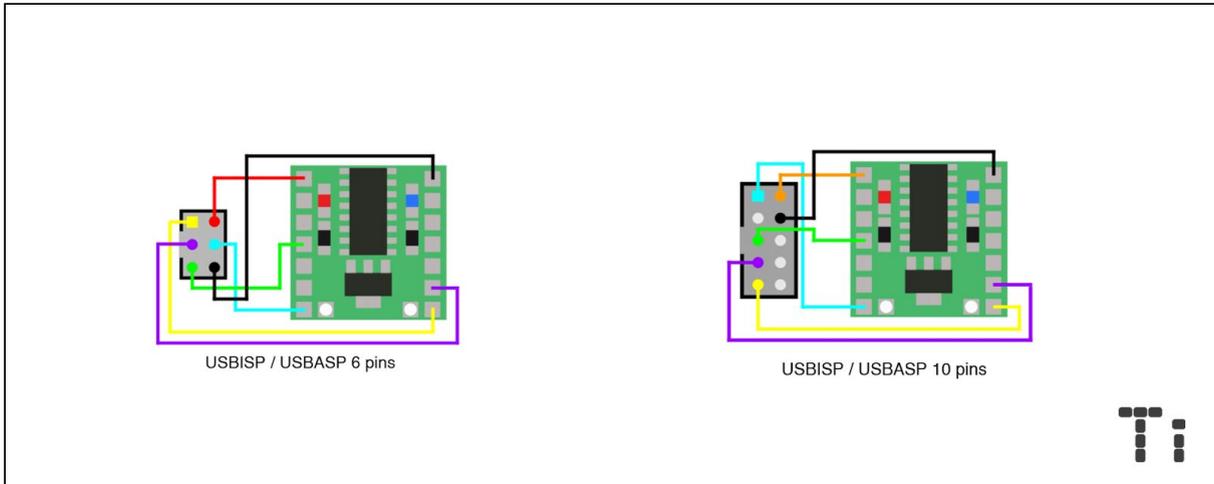


5. Click “Load Flash” from the right bar and select your “.hex” file.
6. Click “AUTO” and wait for the write process to be finished.

*PROGISP can be downloaded at <http://www.electrodragon.com/w/ProgISP>

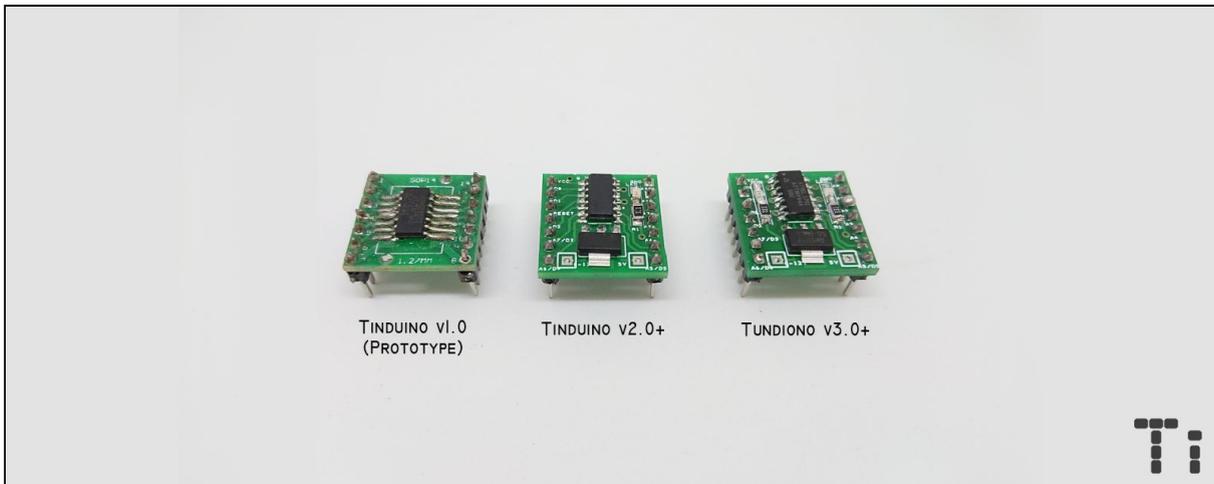
Wiring for Programming

The following wiring diagram is written for both USBISP and USBASP.



Version Identification

You can identify the Tindduino your own by observing the version label printed on the PCB on the bottom side of the Tindunos. You can also observe the main version number of your Tindduino by the board layouts listed in the graph below.



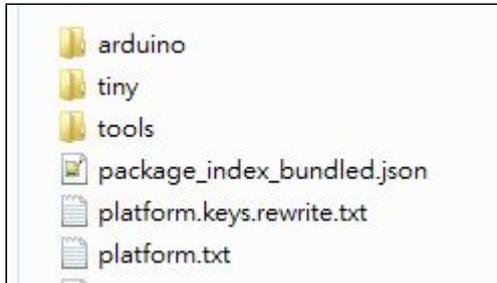
Online Compiler

*The online compiler is still in experimental build. Please use with your own risk.

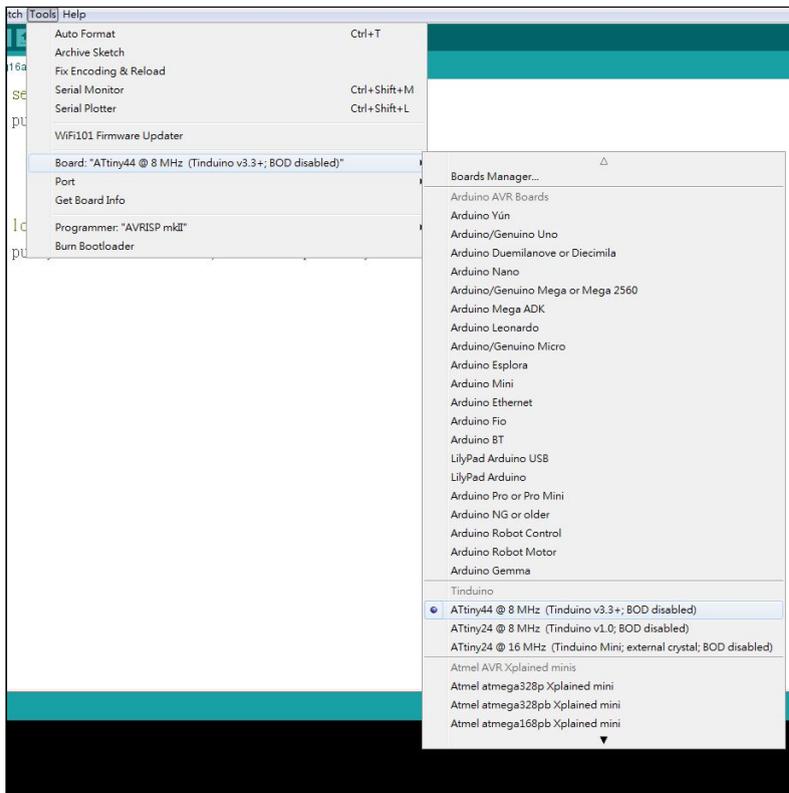
<http://tindduino.com/compiler>

Setting up Arduino IDE for Tindduino

1. Download the board info file from <http://download.hkwtc.org/tiny.zip>
2. Unzip it into “hardware” under your Arduino IDE’s root folder.
(Usually “C:\Program Files (x86)\Arduino\hardware” if you are running x64 version of windows OR “C:\Program Files\Arduino\hardware” if you are running x86 version of windows)
3. After unzipping, you will see at least 3 folders and some other files generated by the Arduino IDE as follow.



4. Launch the Arduino IDE, look for the “Tindduino” section in the Tools/board/ list.



5. If the “Tindduino” section exists, it means that you have installed Tindduino into Arduino IDE successfully.

Advance Board Information Editing

During the process of installing Tindduino into the Arduino IDE, if your Arduino IDE already exists a folder named “tiny”, please do not overwrite the folder or files and follow the instruction below to prevent any bugs created by incompatibility of alternative Arduino compatible boards.

The reason for the existence of “tiny” folder is due to the fact that Tindduino driver and bootloader was modified from the open source “TinyCore” bootloader. Hence, overwriting the existing “tiny” folder will lead to problems if the original TinyCore in your PC have different version of bootloaders.

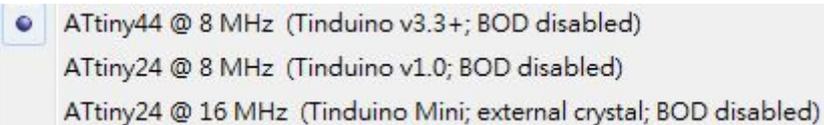
Method 1

1. Unzip the “tiny” folder somewhere else and rename it as “Tindduino”
2. Move the “Tindduino” folder into the “hardware” directory under the root of the Arduino IDE.
3. Check if the installation was successfully.

Method 2

1. Open “boards.txt” under the “tiny” folder downloaded from the Tindduino site.
2. Copy ALL lines in the Tindduino’s “boards.txt” that contains no “#”
3. Open “boards.txt” from your original “tiny” folder.
4. Paste all the text from your clipboard into the bottom part of your original “boards.txt”
5. Save and Exit.
6. Check if the installation was successful

For this installation method, the section name will not show as “Tindduino”. However, Tindduino boards setting are still being added into the board list as shown in the picture below.



```
ATtiny44 @ 8 MHz (Tindduino v3.3+; BOD disabled)
ATtiny24 @ 8 MHz (Tindduino v1.0; BOD disabled)
ATtiny24 @ 16 MHz (Tindduino Mini; external crystal; BOD disabled)
```