**阿米巴的第一步**

**Ameba Arduino: Getting Started**

Required Environment

Currently, Ameba currently supports Windows XP/7/8 32 and 64 bits and MAC OS operating systems. In this example, please use Arduino IDE with version 1.6.5 or later.

Introduction to Ameba

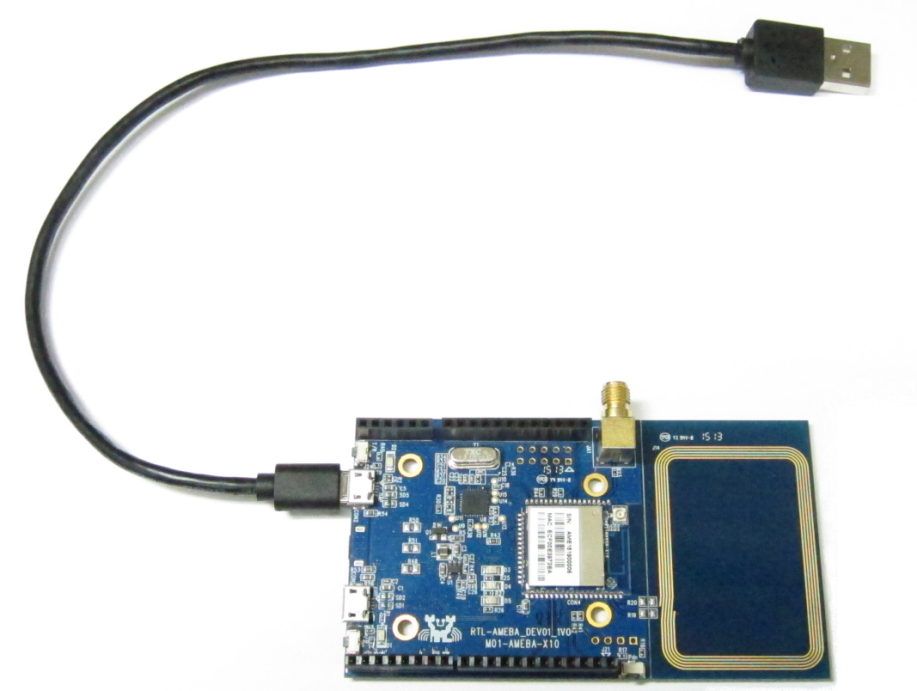
Ameba is a programmable platform for developing all kind of IoT applications. Ameba is equiped with various peripheral interfaces, including Wifi, GPIO, NFC, I2C, UART, SPI, PWM, ADC. Through these interfaces, Ameba can connect with electronic components such as LED, switches, manometer, hygrometer, PM2.5 dust sensors, …etc.  
The collected data can be uploaded via WiFi, and be utilized by applications on smart devices to realize IoT implementation.

  
  
Ameba and Arduino Uno have similar size, as shown in the above figure (Ameba is the upper one), and the pins on Ameba are compatible with Arduino Uno. Besides the pins, Ameba boards are equipped with an NFC antenna (the right part of Ameba board.) If the NFC feature is not required, the NFC antenna can be cut down from Ameba board.  
Ameba uses Micro USB to supply power, which is common in many smart devices.

Setting up Development Environment

Step 1. Installing Driver

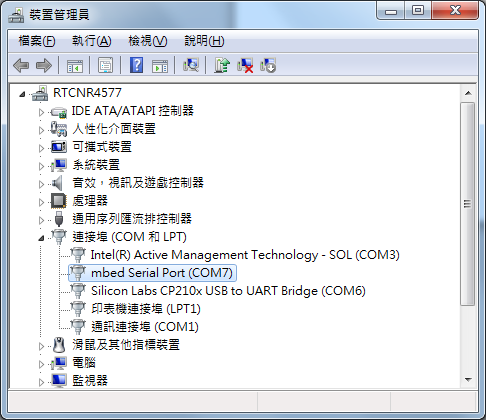
First, connect Ameba to the computer via Micro USB:



If this is the first time you connect Ameba to your computer, you have to install the USB driver for Ameba. Ameba uses the standard ARM MBED CMSIS DAP driver, you can get the installation file and related information in the following website:

<https://developer.mbed.org/handbook/Windows-serial-configuration>

In “Download latest driver” download and install “mbedWinSerial\_16466.exe”. Afterwards, you will see “mbed Serial Port” in Device Manager of your computer:

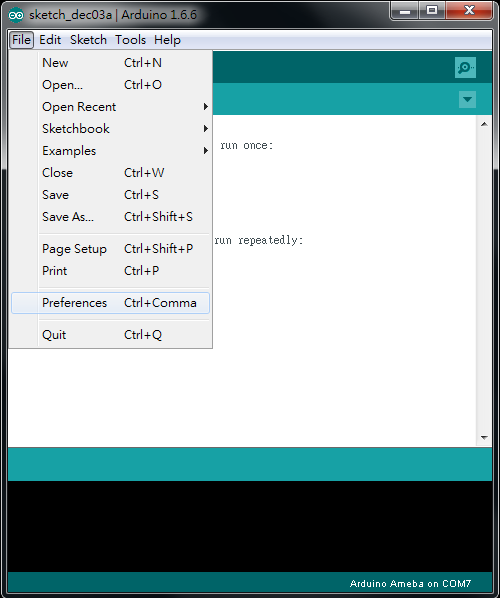


Step 2. Set up Arduino IDE

From version 1.6.5, Arduino IDE supports third-party hardware. Therefore, we can use Arduino IDE to develop applications on Ameba, and the examples of Arduino can run on Ameba too. Arduino IDE can be downloaded in the Arduino website:

<https://www.arduino.cc/en/Main/Software>

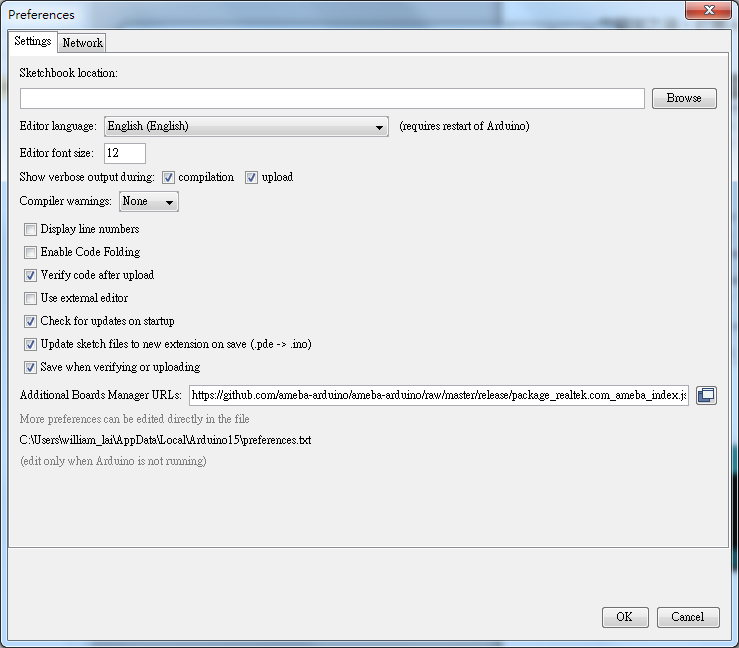
When the installation is finished, open up Arduino IDE. To set up Ameba correctly in Arduino IDE, go to “File” -> “Preferences”



And paste the following URL into “Additional Boards Manager URLs” field:

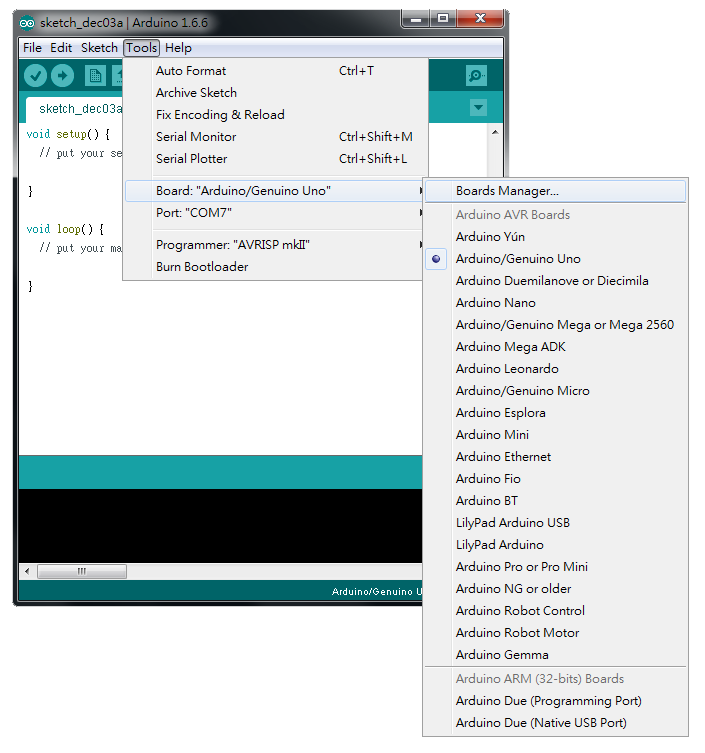
https://github.com/Ameba8195/Arduino/raw/master/release/package\_realtek.com\_ameba\_index.json

Moreover, if you use IDE version earlier than 1.6.7, please make sure the IDE language is set to English to avoid problem.  
In version 1.6.7, Arduino has resolved the language problem.

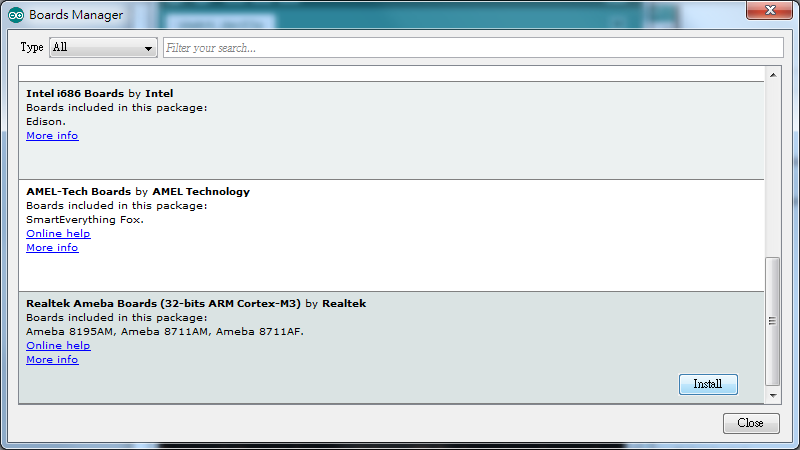


If you have changed the language setting, please restart the Arduino IDE.

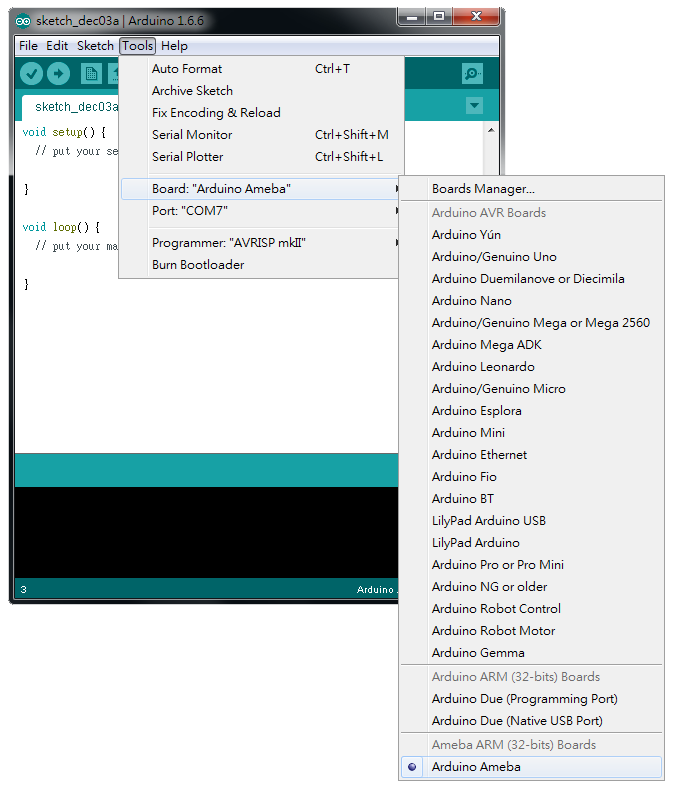
Next, go to “Tools” -> “Board” -> “Boards Manager”:



The “Boards Manager” requires about 10~20 seconds to refresh all hardware files (if the network is in bad condition, it may take longer). Every time the new hardware is connected, we need to reopen the Board Manager. So, we close the Boards Manager, and then open it again. Find “Realtek Ameba Boards” in the list, click “Install”, then the Arduino IDE starts to download required files for Ameba.



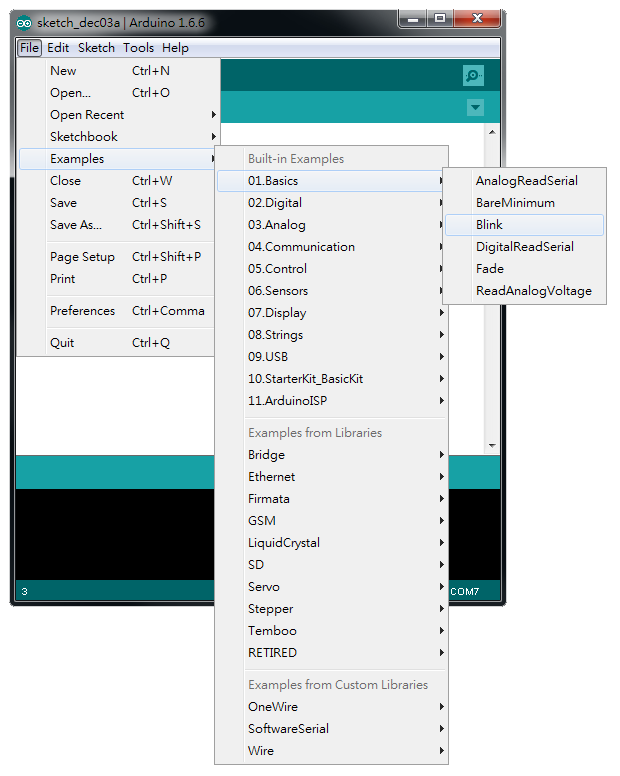
Finally, we select Ameba as current connected board in “tools” -> “Board” -> “Arduino Ameba”：



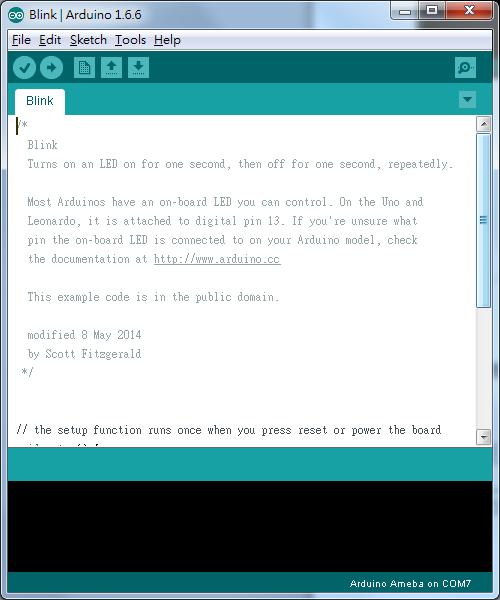
Try the First Example

Step 1. Compile & Upload

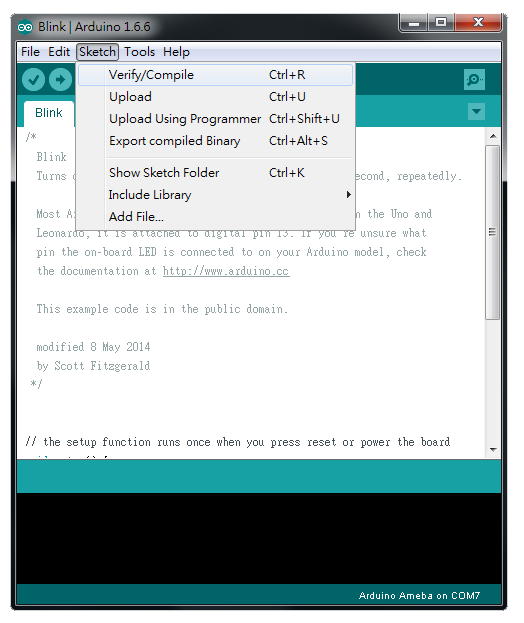
Arduino IDE provides many built-in examples, which can be compiled, uploaded and run directly on the boards. Here, we take the “Blink” example as the first try.  
Open “File” -> “Examples” -> “01.Basics” -> “Blink”:



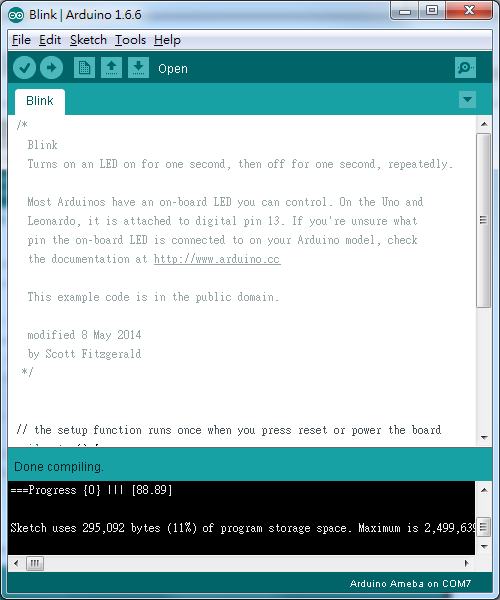
Arduino IDE opens a new window with the complete sample code.



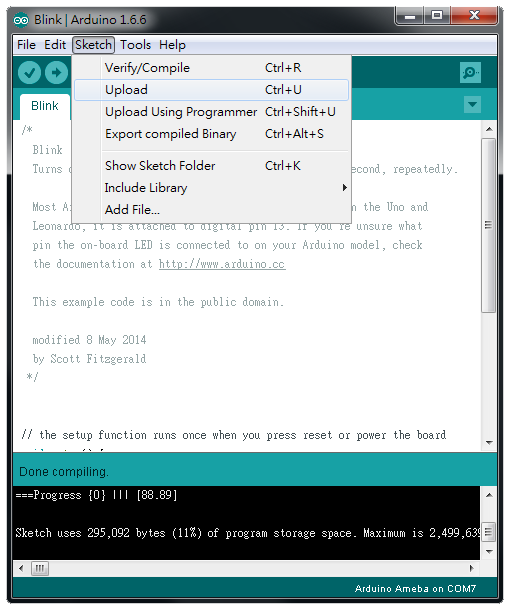
Next, we compile the sample code directly, click “Sketch” -> “Verify/Compile”



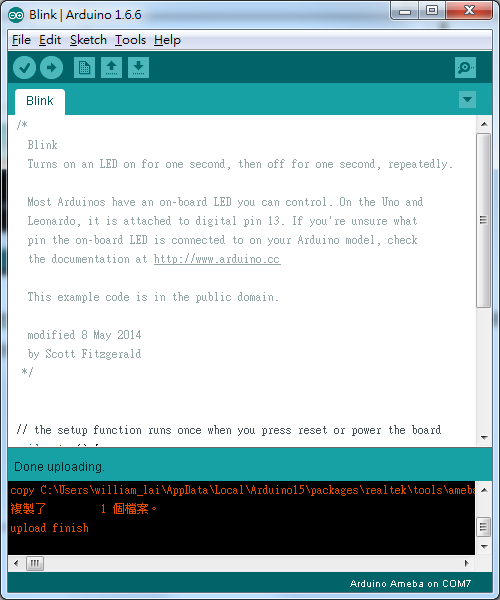
Arduino IDE prints the compiling messages in the bottom area of the IDE window. When the compilation is finished, you will get the message similar to the following figure:



Afterwards, we will upload the compiled code to Ameba. Please make sure Ameba is connected to your computer, then click “Sketch” -> “Upload”



Again, during the uploading procedure the IDE prints messages. Uploading procedure requires respectively longer time (about 30 seconds to 1 minute). When upload completed, the “upload finish” message is printed.

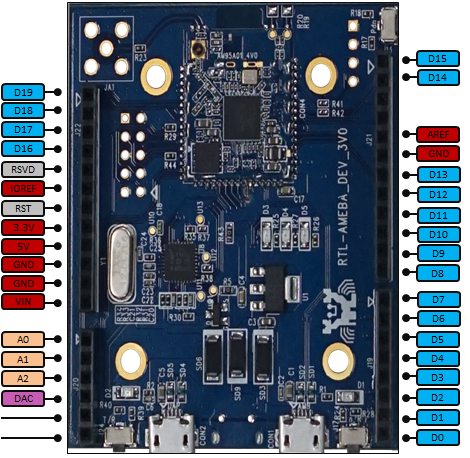


Step 2. Run the Blink example

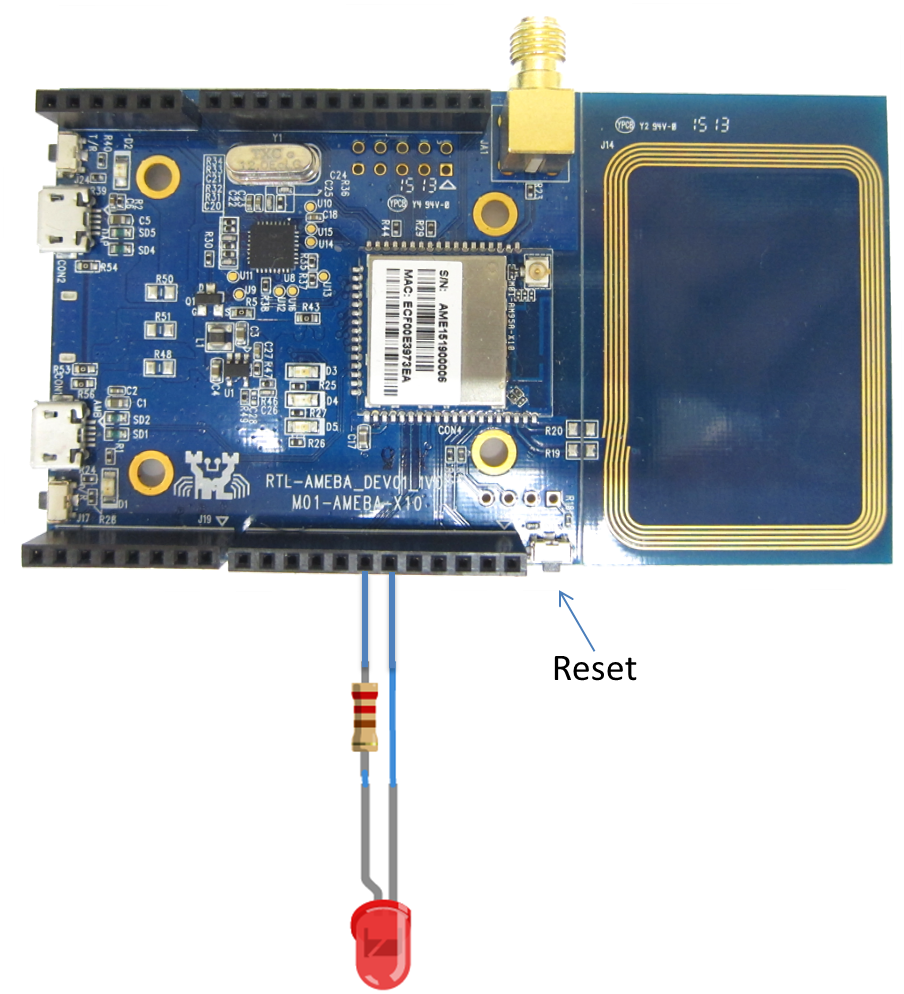
In each example, Arduino not only provides sample code, but also detailed documentation, including wiring diagram, sample code explanation, technical details, …etc. These examples can be directly used on Ameba.  
So, we find the detailed information of the Blink example:

<https://www.arduino.cc/en/Tutorial/Blink>

In short, this example make LED blinks, and it uses GPIO pin 13.  
According to the pin diagram of Ameba, we find pin D13:



Then we connect the LED and resistance as the following figure:  
（NOTE: In an LED, the longer pin is the positive pole, and shorter pin is the negative pole. So we connect the longer pin to D13, and connect the shorter pin to GND.）



Finally, press the reset button, and you can see the LED blinking.  
If you encounter any problem, please refer to [Trouble-shooting](http://www.amebaiot.com/ameba-arduino-trouble-shooting/)