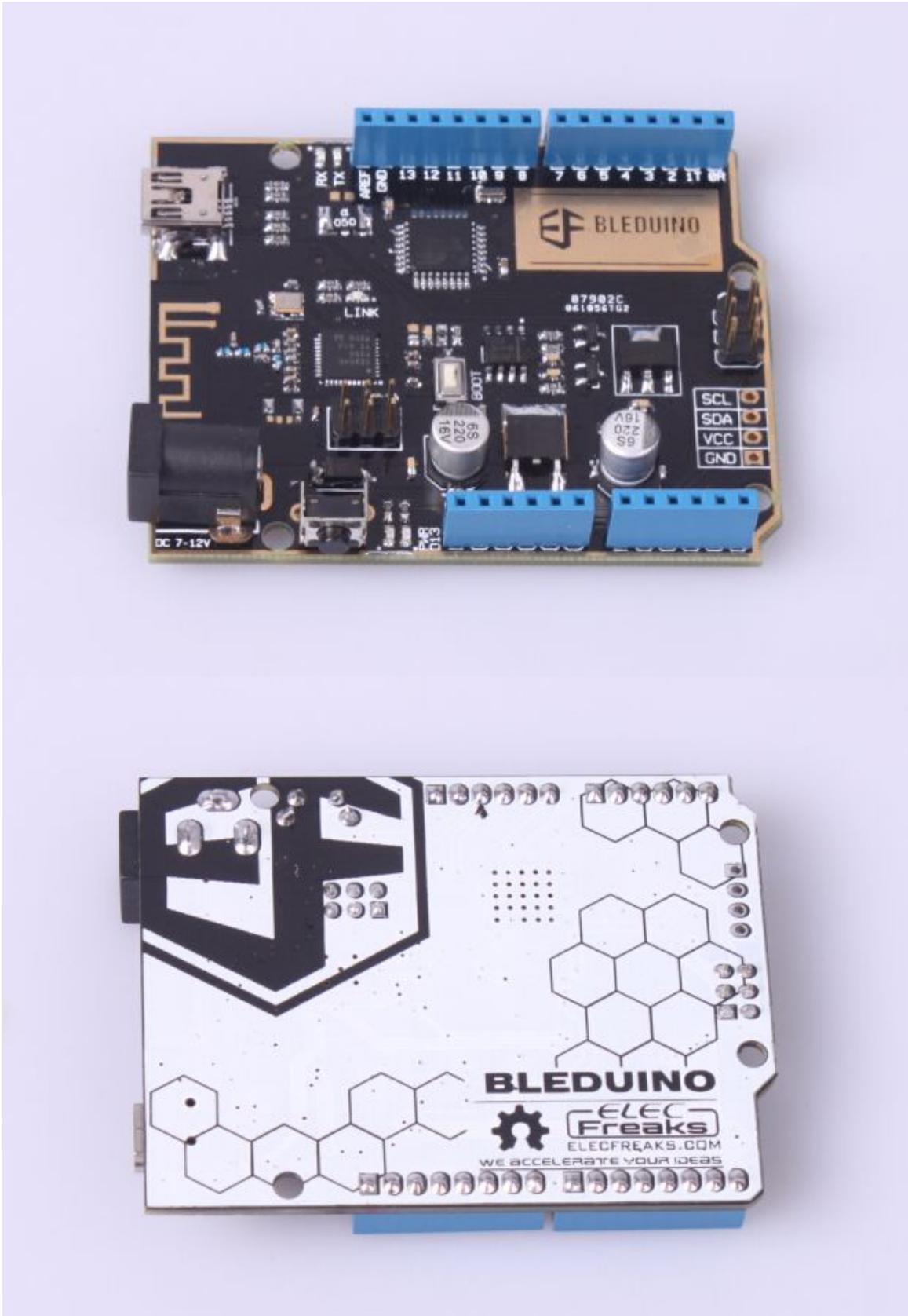


1. General description

It's time to get Bluetooth 4.0 into your project, together with your phone! For aficionados of smart devices and wearables, now you can go further than hacking things bought in the market to building your own prototype out of garage. The BLEduino board is first of its kind in integrating BT 4.0(BLE) module into Arduino Uno, making it an ideal prototyping platform for both software and hardware developers to go wireless. You will be able to develop your own smart bracelet, smart pedometer and more. Through the low- power Bluetooth 4.0 technology, real-time low energy communication can be made really easy. The BLEduino integrates with a TI CC2540 BT 4.0 chip with the Arduino UNO development board. It allows wireless programming via BLE, supports Bluetooth HID, supports AT command to configure the BLE, and you can upgrade BLE firmware easily. The BLEduino is also compatible with all Arduino Uno pins which means any project made with Uno can directly go wireless! What is more, we also developed the App for the BLEduino (both Android and IOS), and they are completely open source, so that you can modify and develop your own BLE-hardware platform. Below is a quick demo video covering some of the major features of BLEduino with the help of an Accessory Shield for BLEduino, which will also be available very soon. In short, you can use BLEduino with any Bluetooth 4.0 compatible devices and enjoy features such as wireless transmission, host and slave settings, wireless burning, and even establishing a Bluetooth HID connection with the PC

Note: For the demo application and Arduino code, we integrated [electfreaks wireless libraries](#) for the beginners. The idea is supplying a simple way for you to use wireless modules without learning the wireless communication protocol. However, for the developer, recommend to custom or choose the protocol according to the product features or the application.



2. Hardware and Software Preparation



Hardware:

1. Android phones with system 4.3 and Bluetooth 4.0
2. USB Cable
3. BLE Adapter
4. BLEduino Mainboard

Software:

1. Arduino IDE
2. APP: Arduino Controller
3. Serial port debugging software (sscom32E)

3. Application

Part 1 BLEduino USB Programming

Step 1 Hardware preparation

Connect the BLEduino to the PC via USB cable

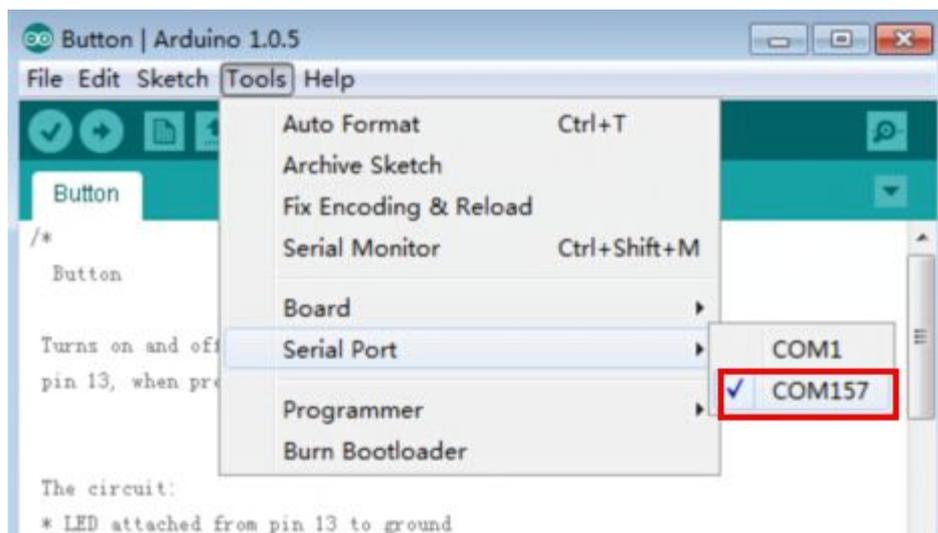
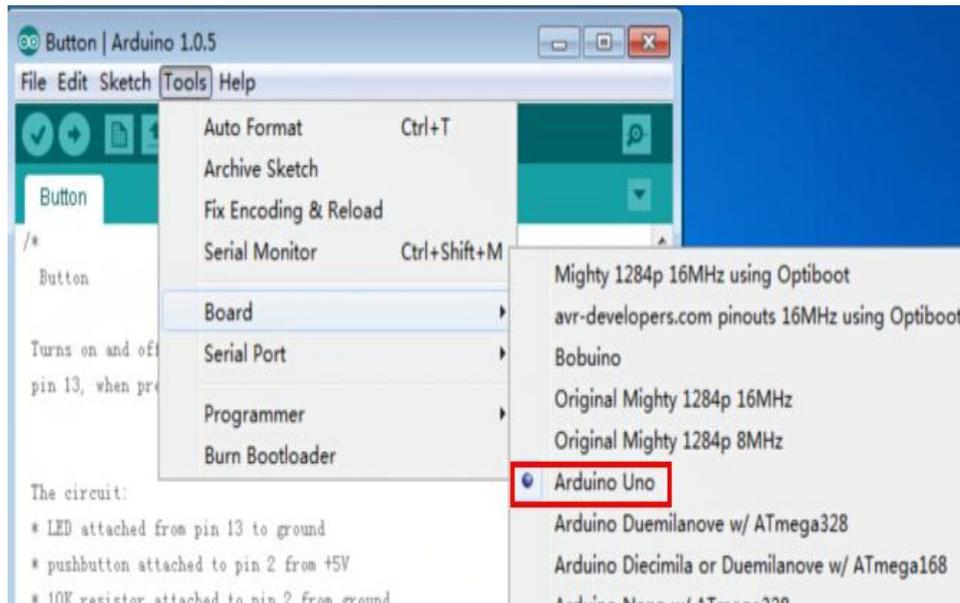
Step 2 Install the driver

When connected to the computer, the computer will prompt to install driver, follow the prompts to install it(the way to install the driver is the same with Arduino UNO and the Driver Name is TI CC2540 Drivers).



Step3 Programming

1. Open the program we need to upload with Arduino IDE, and then choose the Board of Arduino UNO, in addition, the corresponding Serial Port.



2. Compiling sketch until Done compiling appears
3. Uploading sketch until Done uploading appears (when uploading, TX, RX LED will be blinking)

Note: When USB programming, BLEduino had to disconnect with other Bluetooth, and the LINK LED will be twinkling when the Bluetooth is disconnected.

```

Done uploading.
### | 100% 0.33s

avrdude: verifying ...
avrdude: 1084 bytes of flash verified
avrdude: Send: Q [51] [20]
avrdude: Recv: . [14]
avrdude: Recv: . [10]

avrdude done. Thank you.
    
```

1 Arduino Uno on COM161

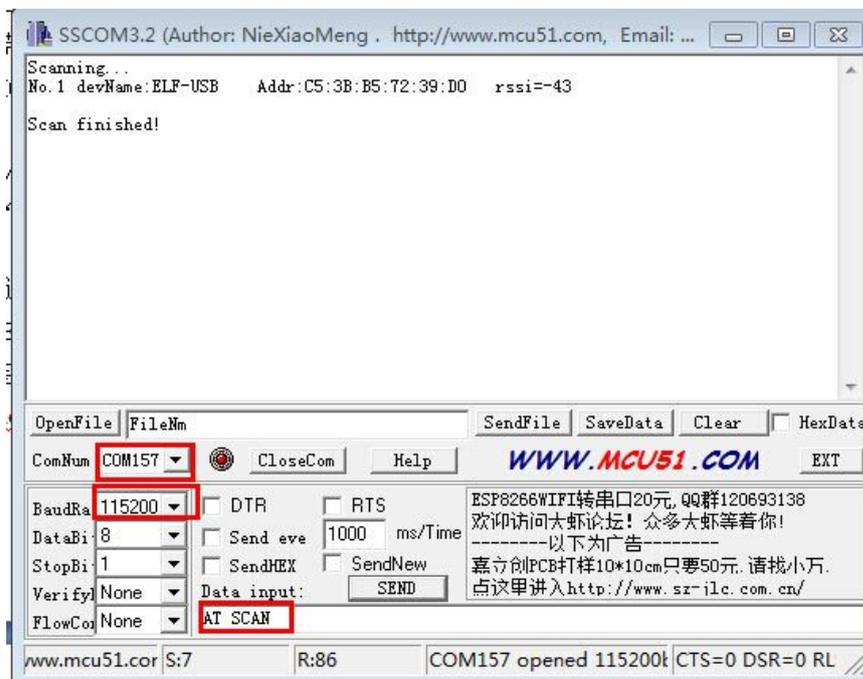
Part 2 BLEduino Wireless Programming

Step 1 Hardware preparation

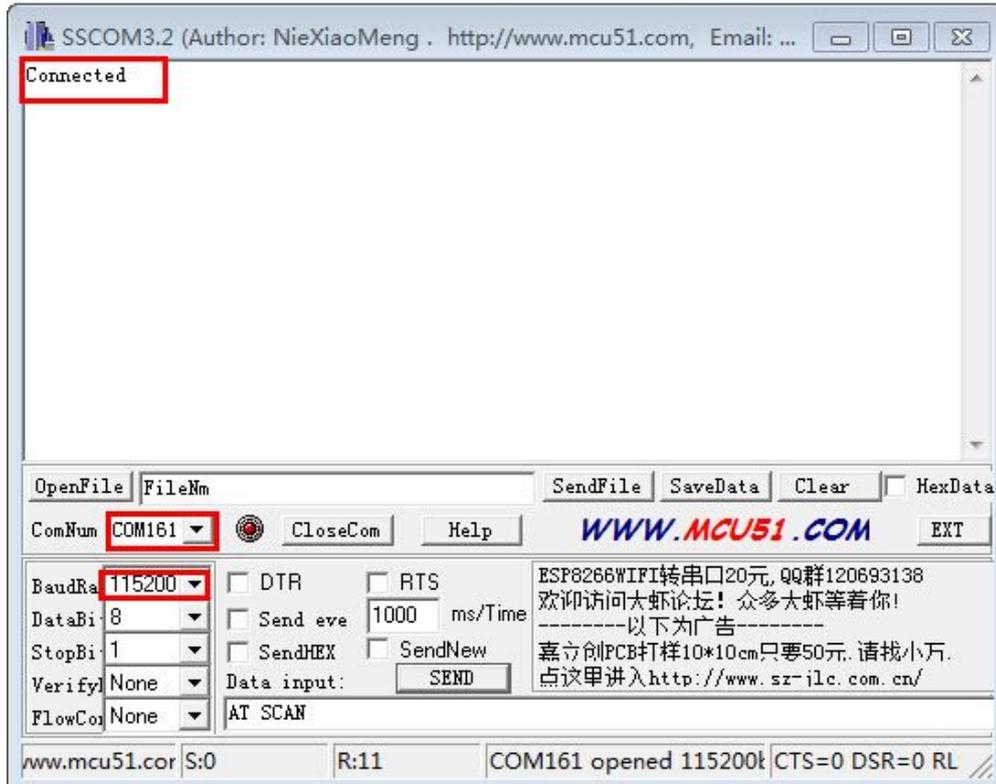
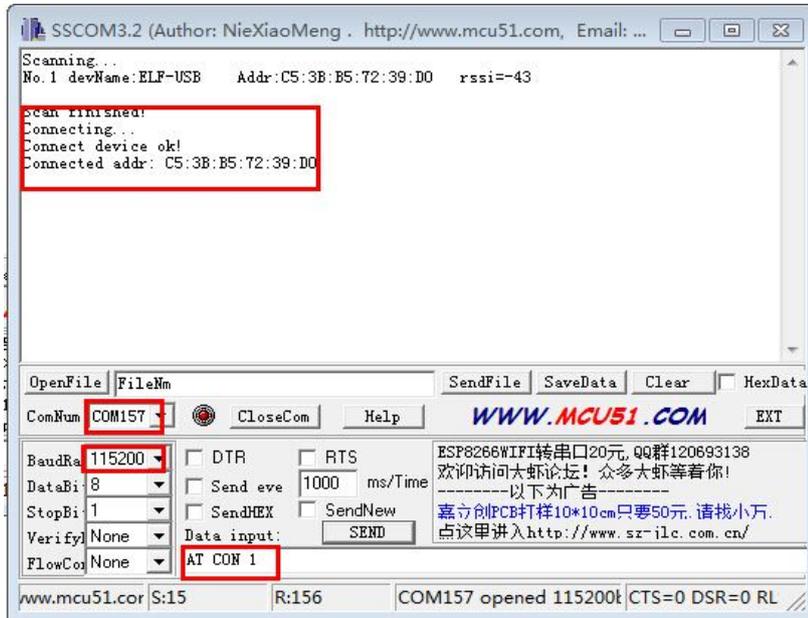
Prepared BLEduino, BLE Adapter and USB Cable

Step 2 Host searched slave and then connected to Slave

Connect the BLE Adapter (Defaulted as host), BLEduino (Defaulted as Slave) to the computer. Via serial port debugging software, open the serial port of host and Slave, and then host send AT SCAN+ Enter to search Slave, just as the following screenshot shows



After the host searched the Slave, the host send AT CON 1 + Enter to connect the Slave, and the Slave can get the host Return Value of connected if connection is successful,at the same time, BLEduino link LED will stop blinking and be always lit.



Step 3 BLE Adapter(host) Programming for BLEduino (Slave)

After the host connected to Slave, open the program need to be uploaded via Arduino IDE, choose the corresponding Board of UNO, choose the BLE Adapter (host) Serial Port, and then the specific operating steps are the same with USB programming, which has been introduced in detail previously.

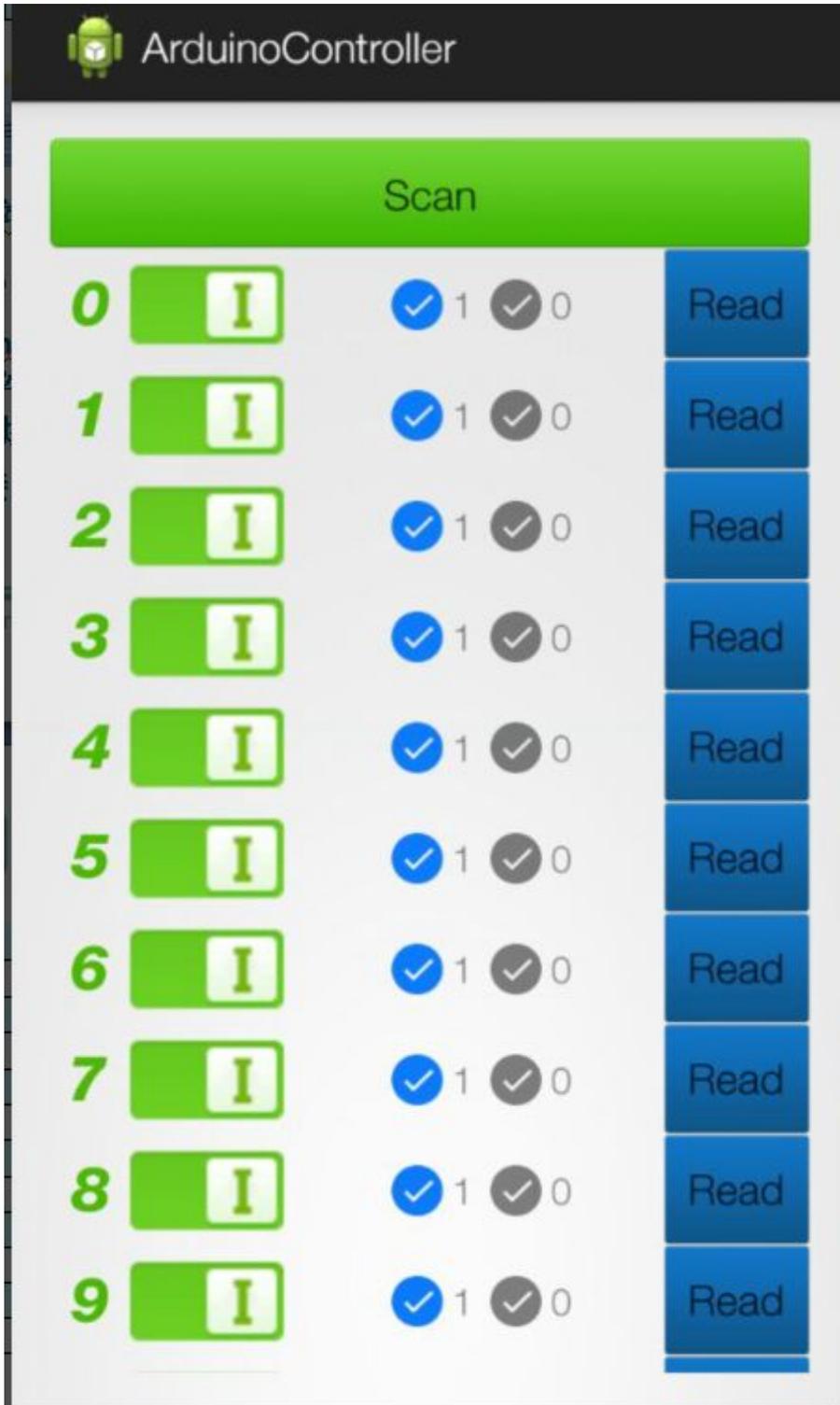
Part 3 Mobile APP Control BLEduino MCU IO Mode

Step 1 Firmware Programming

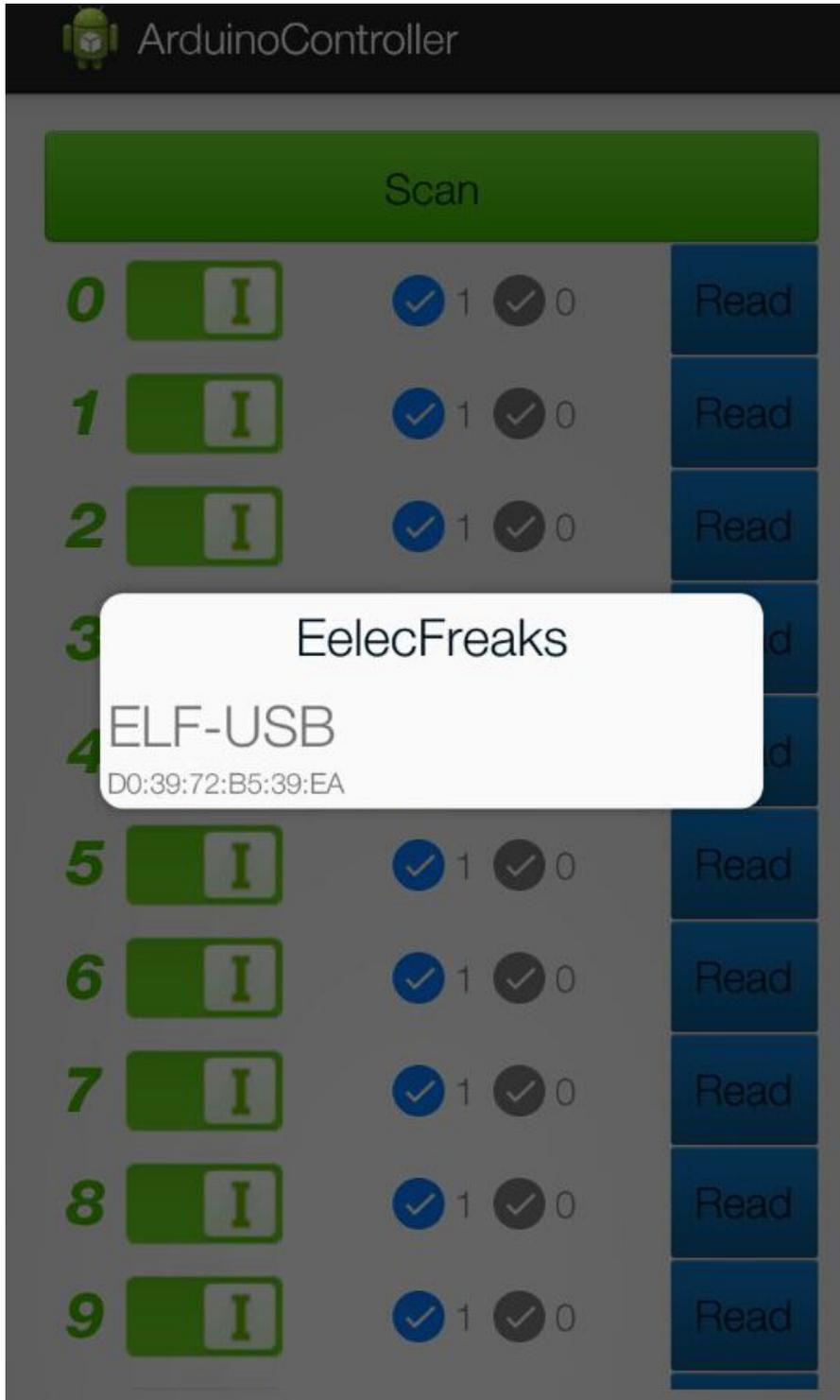
Download Firmware (AndroidIOControl) and then open it Via Arduino IDE, program the firmware to BLEduino MCU with USB, and the specific operation steps are the same with previous USB programming.

Step 2 Mobile phone connected to BLEduino(had to be set to Slave)

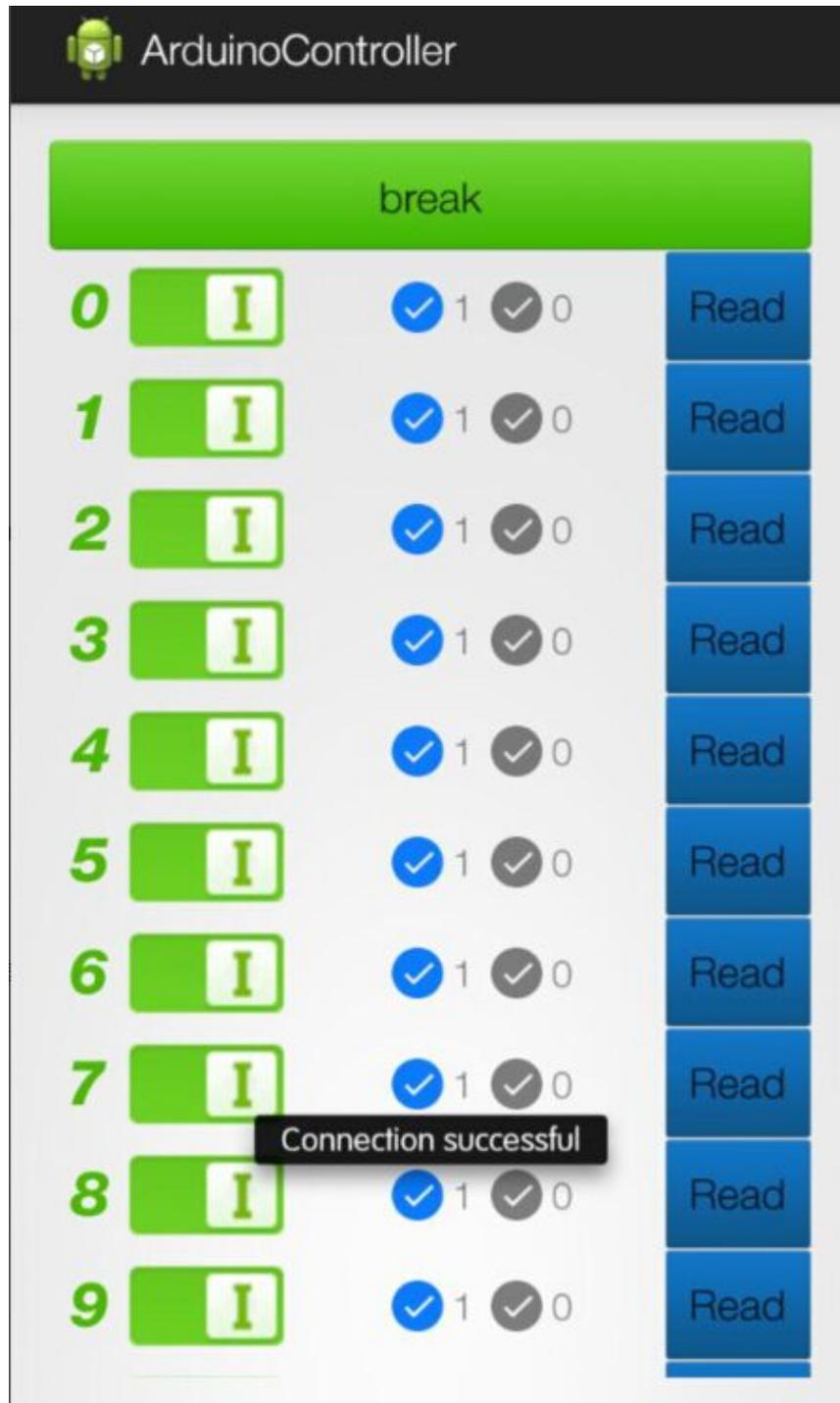
1. Open the Phone APP operating interface



2. Click Scan to search Slave (BLEduino) as below



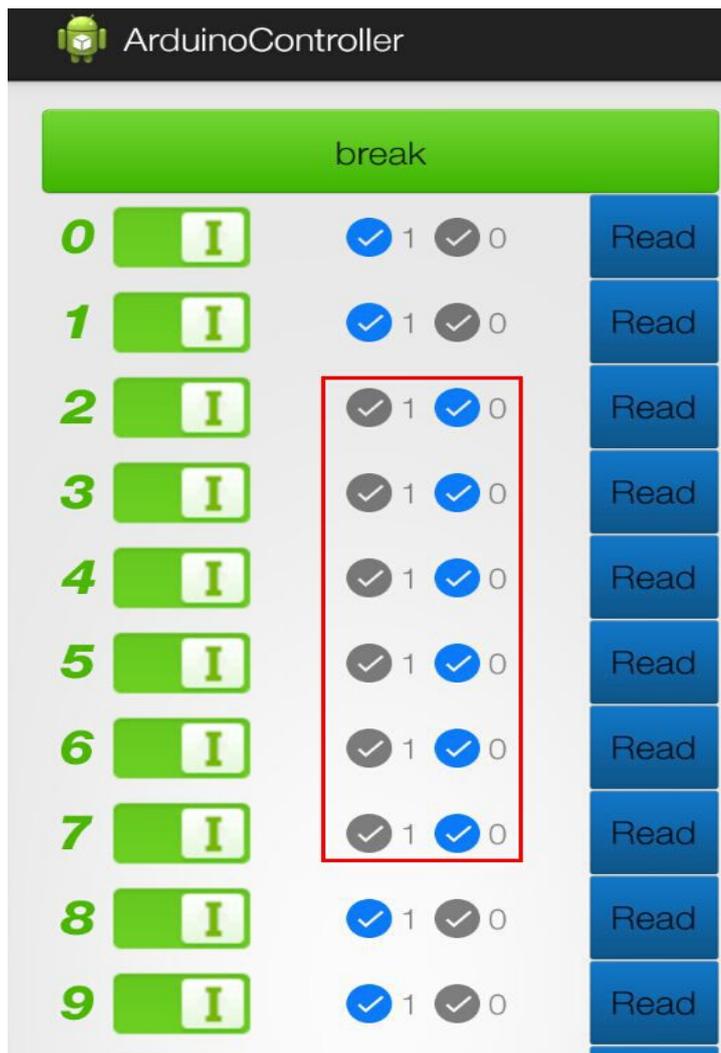
3. After searching the Slave (ELF-USB) , click ELF-USB to connect it, when it succeeded, Link LED will be always lit and phone App will prompt“connection successful”



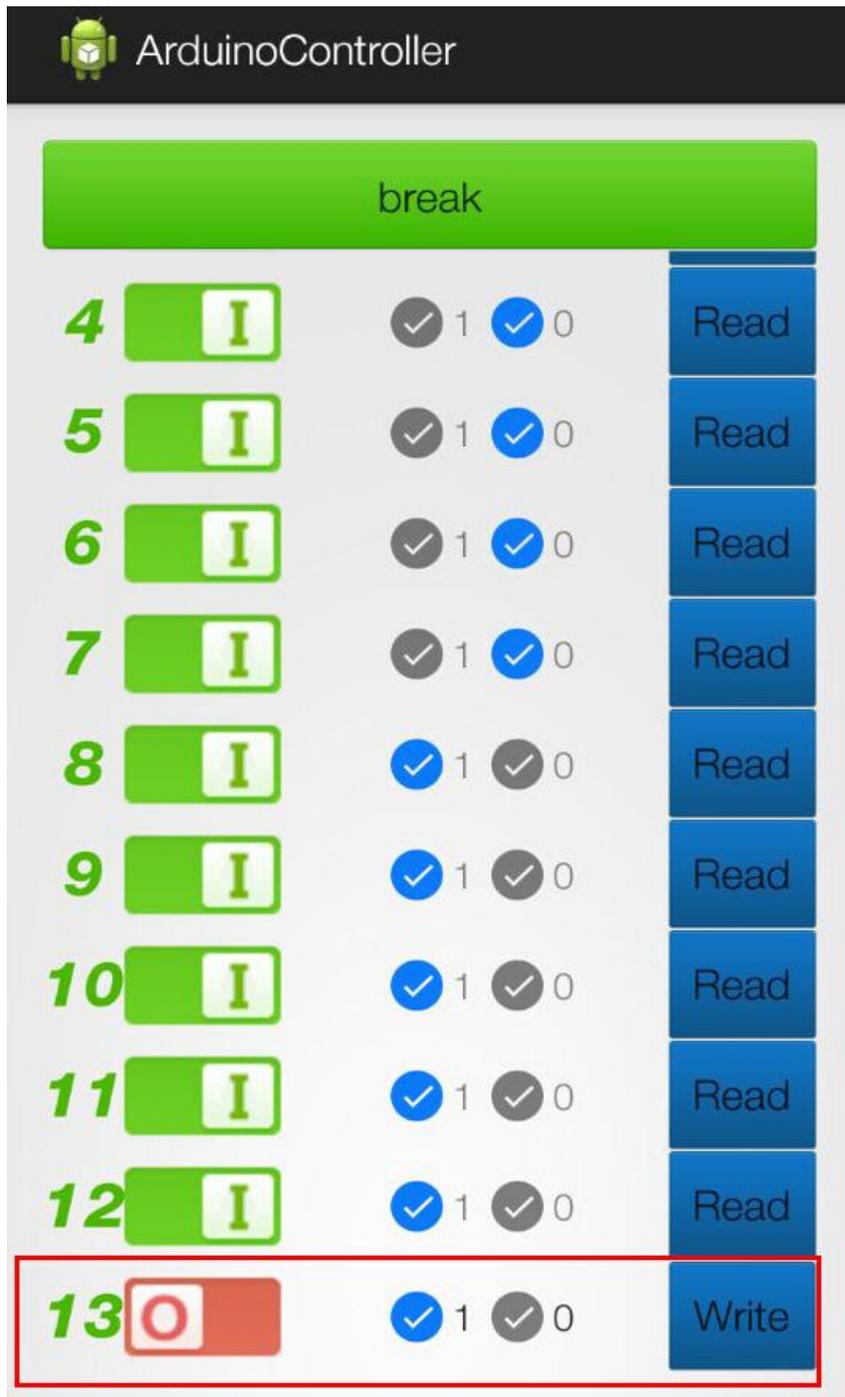
Step 3 Mobile Control BLEduino MCU IO

Note: After the connection between BLEduino and mobile, if you want to perform the MCU read and write operations, MCU has to be reset firstly (pressing the reset button is ok)

1. Firstly is Read Control to BLEduino MCU IO. On the left side of the App UI, modify the IO switches which mean to control to I mode, then press the Read button on the right, you can read the current MCU IO status, such as the following examples. (Operation IO: 2, 3, 4, 5 6 7)
2. When press the Read, the IO value is low (0). This is the complete simple Read operation.



- Below is the Write control for BLEduino MCU IO. On the left side of the App UI, modify the IO switches which mean to control to O mode, such as the following examples. (Operation IO: 13)
- When press the Write button, Write out IO13 value as high (1), and the D13 LED of the BLEduino mainboard would be lit, but if Write out IO13 value as low (0), the D13 LED would be off.



Part 4 AT Command description

1. AT CON // Connecting to slave, and the following parameters are the Slave Number from the searching list. AT CON ? can view help information.
2. AT BAUD // View, Set the baud rate, following no parameter means viewing baud rate, following with parameter means setting baud rate(parameters are baud rates). AT BAUD ? can view help information.
3. AT NAME // View, Set the device name, following no parameter means viewing device name, following with parameter means setting device name (parameters are device names). AT NAME ? can view help information.
4. AT RENEW // Restore to factory settings, no parameters, AT RENEW ? Incapable use, factory setting baud rate as 115200.
5. AT ROLE // View, Set the current role, following no parameter means viewing current role , following with parameter means setting current role(parameters are Role Code, 0 represents Slave, 1 represents host). AT ROLE ? can view help information.
6. AT MYMAC // Obtain MAC Address, no parameters, AT MYMAC ? Incapable use.
7. AT TRANSMODE // View, Set the transfer mode, following no parameter means viewing current transfer mode, following with parameter means setting transfer mode,(parameters are transfer odes)AT TRANSMODE ? can view transfer code detail.
8. AT SCAN // Scan Settings, following no parameters, AT SCAN ? Incapable use
9. AT DISCON // Disconnect the current connection, following no parameters, AT DISCON ? incapable use

4. Revision history

REVISION	DESCRIPTION	RELEASE DATE
V1.2	Initial version	9/29/2014

5. Contact information

If you need more information, please refer to: <http://www.electfreaks.com>