

Project 8: Bluetooth Connection

With the tutorial moves on, you may come up with ideas based on wireless data exchange, such as a mobile weather station, or a portable heart rate monitor. Thanks to the on-board Bluetooth 4.1 module, Genuino 101 can work together with Bluetooth devices, including PC, cellphone or other Bluetooth supported gadgets. In this chapter, we will set our Genuino 101 communicating with the computer through a virtual serial port via Bluetooth dongle. By the end of the chapter, you will be able to send and receive data between 101 and a USB Bluetooth dongle using Arduino IDE serial monitor.

COMPONENT LIST:

1x USB BLE-link



HARDWARE CONNECTIONS

Insert the USB BLE-link into PC's USB port.

Before getting started, we need to know what a **USB BLE-link** is and how it works.

USB BLE-link is a wireless USB dongle based on Bluetooth 4.0. When paired up, it receives data from 101 through Bluetooth and send data to computer via COM port. Need to notice that the firmware loaded in USB BLE-link included in the kit only supports communication with 101 exclusively.

To make it easier to use, USB BLE-link is designed to pair with 101 by approaching. We will walk through a detailed instruction in the following sessions.

BLUETOOTH CONNECTION

Setup Bluetooth Connection

To learn how to use the Bluetooth function, we will first learn how to configure the Bluetooth and identify the connection status.

First, upload the following code to Genuino 101.

The data communication function through serial port is supported by the library "**BLESerial.h**", you will need to download this library from DFRobot website.

Sample Code

```
#include "BLESerial.h"
void setup() {
  pinMode(13, OUTPUT);
  Serial.begin(115200);    // initialize serial communication
  BLESerial.setName("Bluno101");
  BLESerial.begin();
  //Serial.println("Bluetooth device active, waiting for connections...");
  while(!BLESerial);
}

void loop() {
  while (BLESerial.operator bool())
  {
    digitalWrite(13, HIGH);
  }
  digitalWrite(13, LOW);
}
```

The code above initializes 101's Bluetooth function and sets it to be searching for nearby devices (the USB BLE-link in our case). When connected, the LED on pin D13 lights up, otherwise it stays off.

Pair-by-approach Instruction

Once the code is successfully uploaded, turn off all other Bluetooth devices within range and insert the dongle into PC's USB port. To pair up, place the **Bluetooth antenna** of 101 close to the dongle (less than 20mm) and wait till the LED indicators on both devices light up (the blue LED indicator of the dongle can be found inside the pinhole). When both LEDs light up at the same time and stay on, the Bluetooth is connected. After paired up, you can move the 101 away from the dongle and the Bluetooth will stay connected.

Code Analysis

The Bluetooth data exchange is based on the library BLESerial. To initialize Bluetooth connection, we need to use following functions.

`BLESerial.begin()`

Initializes BLESerial function for incoming Bluetooth connection.

`BLESerial`

Checks if BLESerial function has been initialized. Returns 1 when initialization is done.

`BLESerial.operator bool()`

Checks if 101 is paired with the dongle. When paired up, returns 1, otherwise returns 0.

We also added "digitalWrite(13, HIGH)" inside the while loop so the LED stays on when connected

Send and receive data via Bluetooth

In this step, we will add more code so that it can send data back and forth between 101 and USB dongle. By the end, the serial monitor of Genuino 101 will display whatever you sent through the serial monitor of the dongle, and vice versa.

CODE INPUT

```
#include "BLESerial.h"

void setup() {
  pinMode(13, OUTPUT);
  Serial.begin(115200);    // initialize serial communication
  BLESerial.setName("Bluno101");
  BLESerial.begin();
  //Serial.println("Bluetooth device active, waiting for connections...");
  while(!BLESerial);
}

void loop() {
  while (BLESerial.available()) {
    digitalWrite(13, HIGH);
    while(Serial.available()){
      BLESerial.write(Serial.read());
    }
    while(BLESerial.available()){
      Serial.write(BLESerial.read());
    }
  }
  digitalWrite(13, LOW);
}
```

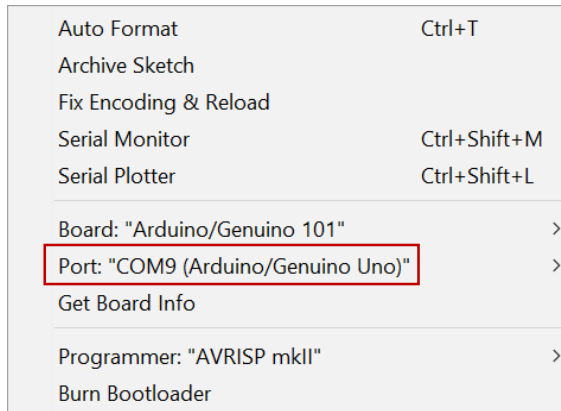
Setup serial communication

When uploaded and paired up (Bluetooth indicator on both devices stay on), we need to use two serial monitors for different COM ports to test the function. To open up two serial monitors, open two Arduino IDE windows by running "Arduino.exe" twice (the file that starts Arduino IDE software). Notice that opening up a new window by creating a new scripts from the menu bar will NOT allow you to use serial monitor for two different COM ports at the same time.

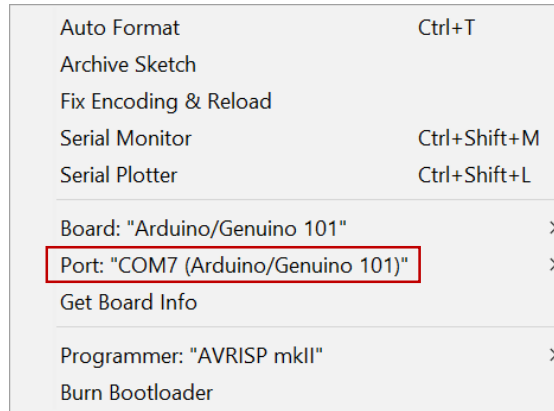


Two Arduino IDE tasks running at the same time

After that, select the **COM port** in Arduino IDE. In our case, 101 is on **COM7**, USB dongle is on **COM9**.

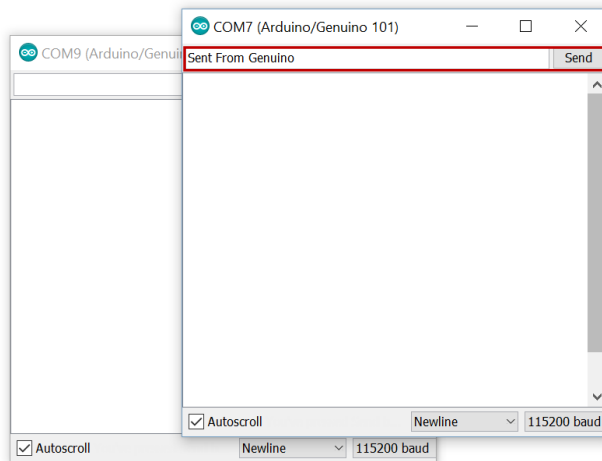


USB dongle on COM9

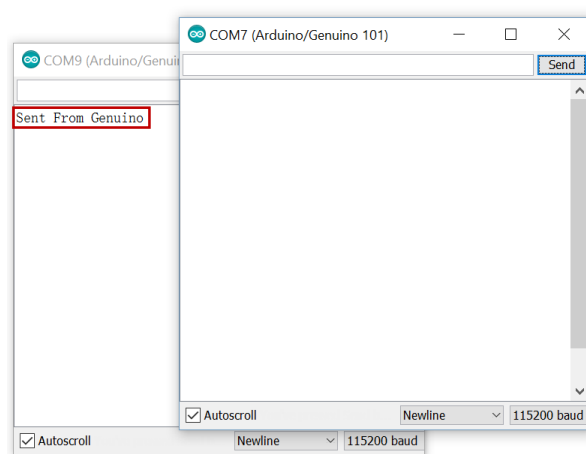


Genuino 101 on COM7

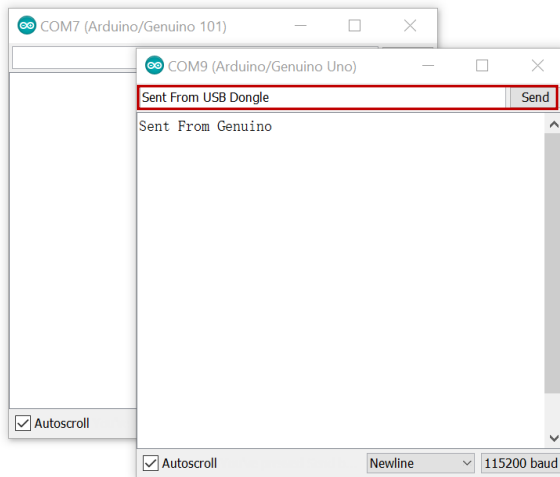
Now you can open the serial monitors and start typing from the input bar in one of the monitor. When done, press **"Send"** and the words will show up in the other monitor.



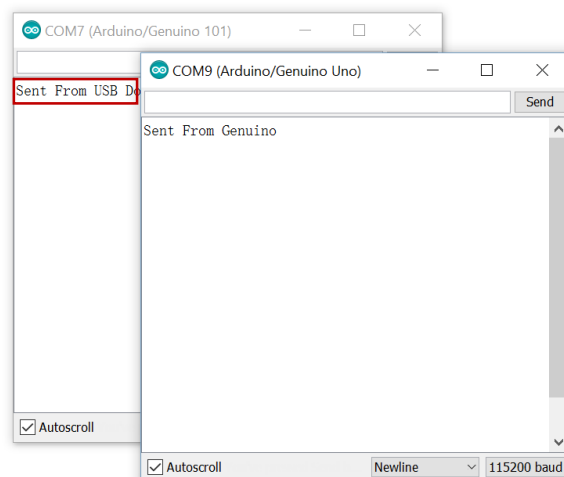
Send string from Genuino 101 (COM7)



Receive string from USB dongle (COM9)



Send string from USB dongle (COM9)



Receive string from Genuino 101 (COM7)

Code Analysis

The mechanism of data exchange function can be interpreted as following. When Bluetooth connection is detected, 101 starts monitoring data fed into its serial port. If there is any, it will be sent to the dongle and printed from dongle' s serial monitor. Meanwhile, it also monitors data fed from the Bluetooth virtual serial port, print it out in its own serial monitor.

```
BLESerial.available()
```

Checks if there is data sent from the virtual Bluetooth serial port. Returns 1 if there is any, otherwise returns 0.

```
BLESerial.write()
```

Sends data to the USB dongle via Bluetooth serial port.

```
BLESerial.read()
```

Reads data to the USB dongle via Bluetooth serial port, returns the data.