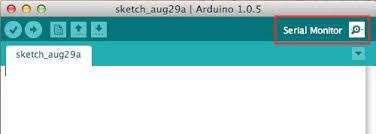
Arduino Romeo v1.0 BLE User’s Tutorial

This tutorial focuses on the Arduino Romeo v1.0’s BLE capability and how to utilize all the features of the board, and Arduino IDE associated with BLE, including terminology and vocab, along with step by step instructions on how to set up the Arduino for BLE use of connecting two Arduino Romeo’s to each other and sending information back and forth. All below is assuming you know the basics of the Arduino Romeo board, and basic programing skills.

What is Bluetooth Low Energy?

Bluetooth Low Energy or BLE may also be called Bluetooth Smart or Bluetooth 4.0, and is a personal area network used for applications, for devices to connect to each other. In the case of the Arduino, the Arduino has built in BLE, and can be connect to other BLE devices, or another Arduino BLE.

The Arduino IDE has a handy tool called a Serial Monitor, and it can be written to and read from whilst executing the Arduino sketch, and when opened the information can displayed to the user. It is the link between the computer and your Arduino, and is useful for debugging and controlling you Arduino with ease via keyboard. To open the monitor you can either do CTRL + SHIFT + M, or go to tools, and select the monitor, or you can click on the icon in the top right hand corner, each way opens the Serial Monitor if the Arduino board is connected.

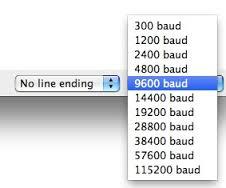


There are several ways to interact with the Serial Monitor while running your Arduino sketch, but before you can send information and receive from the monitor, first you have to use the commands to set up the Serial Monitor for use.

Using the command Serial.begin(baud);will initialize the serial monitor for use at a selected baud rate.

What is a baud rate?

The baud rate is the rate at which information is transferred in a communication channel. In the serial port context, "9600 baud" means that the serial port is capable of transferring a maximum of 9600 bits per second. (MathWorks.com) For this tutorial we will be using 9600 baud because that is all we need for the Arduinos to communicate however any matching baud rate on both Arduinos will work such as both Arduinos are set to 115200, or 300, ect. You set the baud rate in your code, however when opening the monitor you should make sure the monitor is set to the same baud rate and on “no line ending”.



To print to the Serial Monitor for the basics, and for the use of this tutorial, you can use Serial.println (), and to read from the monitor, and store it into a variable you can use Serial.read(), or var = Serial.read() this will store whatever is printed to the monitor into the variable var to be used later.

Example:

Serial.println(127); //will also display this to the monitor

Int var = Serial.read() will return var = 127

Programming the BLE on the Arduino Romeo using AT Commands

AT Commands are instruction commands for controlling a modem. AT is an abbreviation for attention, it essentially tells that part of the board that these bits of information you are sending to it are for it to pay attention to.

To enter AT Mode:

1. Open up the Serial Monitor, and set the baud rate to 115200, and “no line ending”
2. In the text bar at the top type “+++”
3. This will return “Enter AT Mode”
4. After receiving this, set the monitor from “no line ending” to “Both NL & CR”, and keep the baud rate

A list of commands is available at Dfrobot.com, and on the datasheet for the Arduino Romeo, however below are the commands we are using to bind the two Arduino boards, and this is where the boards come in.

1. First we want to get the mac address of the first board we will call this the controller, which will send over information to board two, the receiver, which will receive information.
2. In the serial monitor for the first board after following the steps above type “AT+MAC=?” which will return the mac address of the board currently plugged into to your computer, write it down you will need it later.
3. Now plug in the second board, and get into AT Mode, and input AT+MAC=? To return the second board’s mac address, write it down. While here you can do AT+BIND = (Mac address of the first board), and AT+ROLE=ROLE\_PERIPHERAL which will set the board into the default mode, receiver or slave mode.
4. Plug the first board back in, and AT+BIND = (Mac address of second board), and AT+ROLE = ROLE\_CENTRAL to set the board into controller mode or master mode.
5. All successful commands should return “OK” but feel free to check them with AT+(Whatever) = ? to display the mac address or role inputted.
6. Now the two boards should be connected, and if power is set to both of them a green light labeled link on the controller should light up on both boards.

At this point to communicate the two boards, on the master side Serial.print will be mostly used to send information, and on the receiver side Serial.read will be mostly used to get the information. It helps to think of the two boards as now sharing a common Serial Monitor.

DO NOT UPLOAD CODE WHILE THE BOARDS ARE LINKED, IT DOES NOT WORK

Trouble Shooting:

1. Make sure the baud rates are set the same in both Arduino codes Serial.begin(baud rate);, and in the serial monitors as well, and they are both on “no line ending”
2. For some more complex project such as sending over strings, it helps in the setup routine, to Serial.begin like always, and then Serial.print(“something”) which will set up the monitor, and print to it before the Arduinos are linked, and this should be done on both sides if done at all.
3. Make sure the Arduino boards are not linked before attempting to upload code, and while messing with just one, try not to have the other one on, even though it shouldn’t effect it, it might if you are printing to the serial monitor at any point.