

Name:

Date:

Class:

## Activity Part 3 Worksheet: Microcontroller to Android Serial Communication


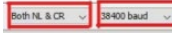
### Equipment Needed:

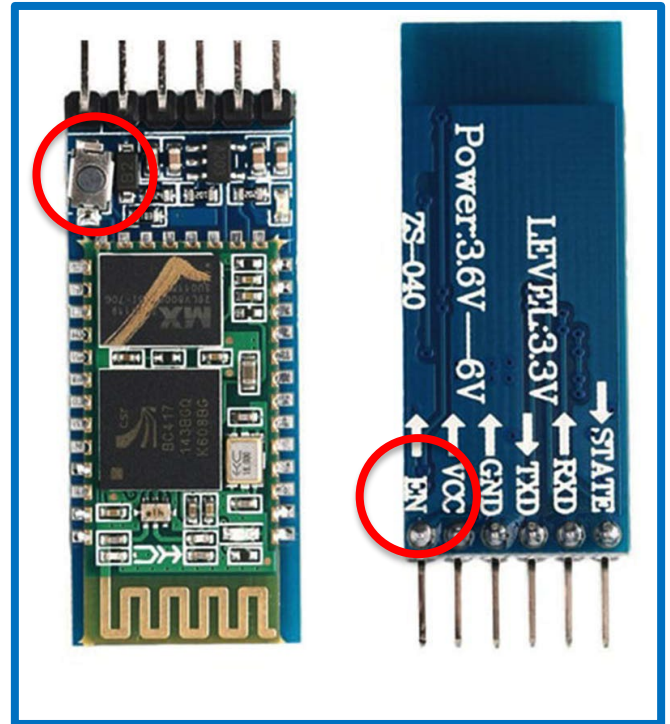
- All materials from Activity 1.
- All materials from Activity 2.
- One 9-volt battery
- One 9-volt battery snap and barrel connector
- One jumper wire
- One HC-05 standard Bluetooth module
- Five male-to-female jumper wires
- One Arduino microcontroller and corresponding USB cable
- One Android smartphone or tablet

### Procedure:

1. Open a new Arduino sketch. You don't need to name it as you will not be modifying nor saving it.
2. Connect the Arduino via USB to the computer. Verify the correct COM port is selected and upload the blank sketch.
3. DISCONNECT the Arduino USB cable from the computer.
4. Using the five female-to-male jumper wires, make the following connections between the HC-05 Bluetooth module and the Arduino Board:

HC-05 Module	Arduino Microcontroller
RX	RX
TX	TX
GND	GND
Vcc	3.3 V
EN	5 V

5. While pressing and holding the button on the HC-05, connect the USB cable to the computer. Release the button when you see the LED doing a slow blink.
6. Open a serial window in Arduino. Look for  the icon in the top right corner.
7. On the bottom, select the options BOTH NL & CR and 38400 baud .
8. In the bar at the top, type AT and then press ENTER or click SEND. Wait for a return message of OK.
9. Next, type AT+NAME=<yourname> and press enter or click SEND. For example, to name mine GIRLBOSS I would type AT+NAME = GIRLBOSS. Wait for a return message of OK.
10. Type AT+RESET and hit return.

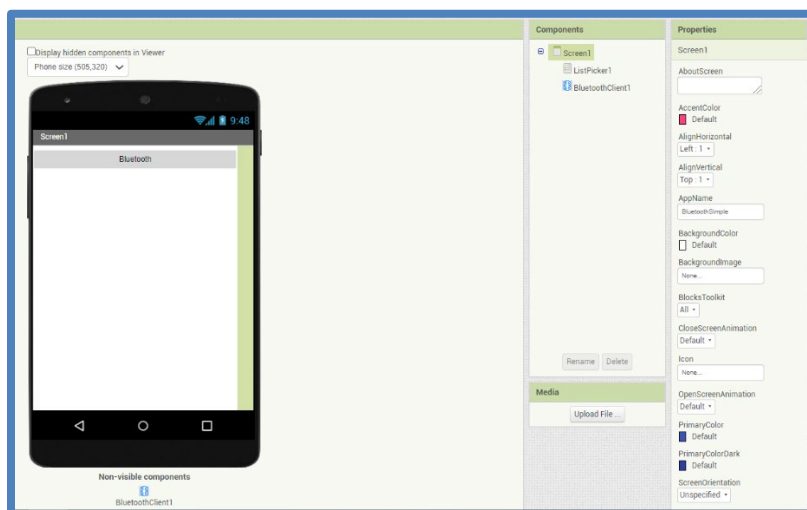
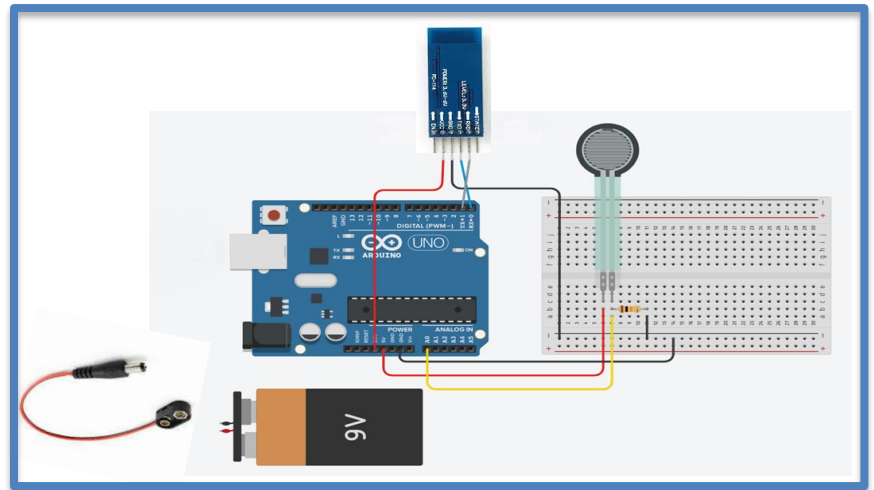


Name:

Date:

Class:

11. Disconnect the EN pin from 5V and remove that jumper wire.
  12. Disconnect the USB cable from the computer.
  13. On the FSR breadboard circuit, reposition the jumper wire that connects the ground to the 10k $\Omega$  resistor to connect to the breadboard ground rail; then add a new jumper between the resistor and the ground rail.
  14. Reconnect the Vcc ground (GND) pin of the HC-05 to the ground rail on the breadboard (shown in black) rather than directly connected to the Arduino.
  15. Swap the TX and RX pins on the HC-05 end. In other words, the TX pin on the HC-05 connects to the RX pin on the Arduino (shown in blue); likewise, the RX pin on the HC-05 connects to the TX pin on the Arduino (shown in grey).
- IMPORTANT: Notice that the transmit and receive connect to the opposite of each other on the two devices. This is different than when you were renaming the HC-05 module.
- IMPORTANT: Whenever you need to upload a new Arduino sketch, you will need to disconnect the TX and RX jumper wires from the Arduino and reconnect following the completion of the upload.
16. Connect the snap end on the 9-volt battery but hold off on connecting the barrel connector.
  17. Use the computer to create a new MIT App Inventor 2 project. On the Designer tab you will need a list picker labeled BLUETOOTH along with a Bluetooth Client non-visible component.

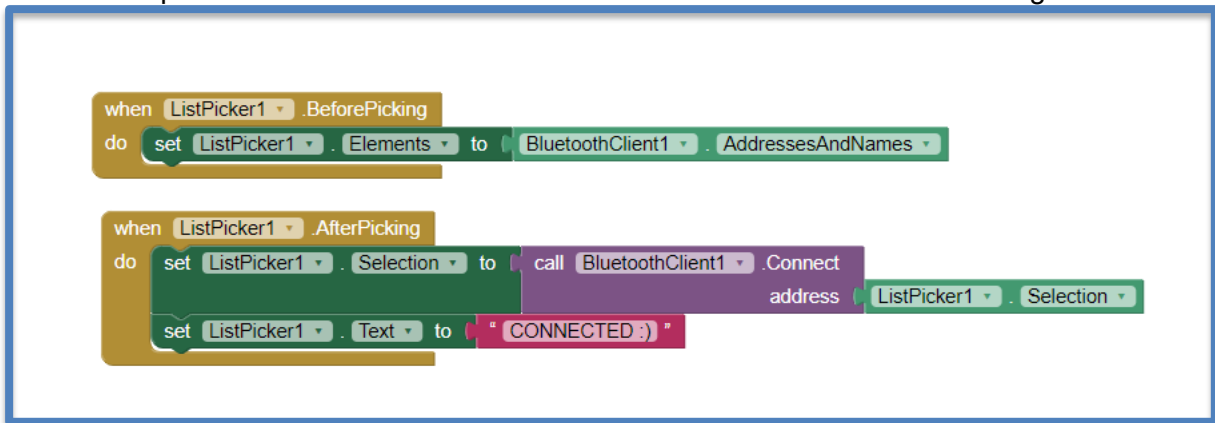


Name:

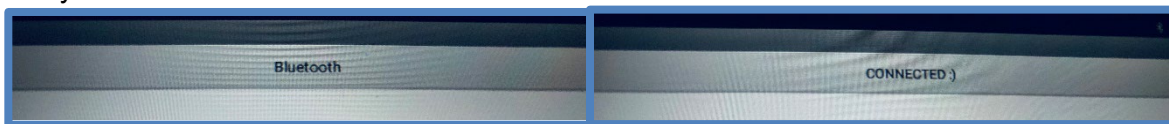
Date:

Class:

18. For the Blocks tab you will need one set of blocks to control selecting the Bluetooth device from the listpicker and then one set to connect to the selected device and change the label.



19. Choose Build App (provide QR code for .apk). If the tablet or phone is unable to read the QR code, simply type the given URL into Google Chrome. Download the app and open it.
20. Disconnect the USB cable from the Arduino board and connect the barrel connector to the barrel input of the Arduino microcontroller.
21. On the tablet or smartphone, make sure Bluetooth is enabled, select the HC-05 (by whatever name you programmed), and pair the devices.
22. On the tablet or smartphone, tap the listpicker button and wait for the HC-05 to appear.
23. Select the HC-05 and verify that the listpicker text changes to CONNECTED or whatever text you coded.



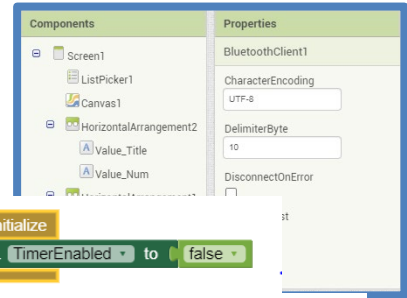
24. Open the Arduino code from the previous activity and change the Arduino code to eliminate any strings such as "The voltage is : " and only keep one serial monitor output such as: `Serial.println(FSRval);`
25. Disconnect the TX and RX jumpers from the Arduino board and upload the new code.
26. When the upload is complete, re-connect the TX and RX jumpers to the Arduino board.
27. In MIT App Inventor, add a horizontal arrangement with two labels. One label should display text such as "VALUE = " while the other should be set to 0 at the start.
28. Next, add a START button somewhere on the screen.

Name:

Date:

Class:

29. Next, add a clock to the non-visible components section. For the clock properties, set the timer to be enabled, the clock to always fire, and the interval to 250 milliseconds.
30. For the Bluetooth Client properties, set the delimiter character to 10.
31. In the Blocks section, you need to begin with the clock disabled using Screen1.Initialize.
32. Next, add code to enable the clock if the START button is clicked. The button text alternates between START and PAUSE and toggles the clock between enabled and disabled.
33. Create and initialize a global variable to hold the digital sensor values.



```
initialize global FSRvalue to 0
```

34. Finally, you will need to add blocks to control the clock and to display the sensor values.

```
when Screen1.Initialize
do
  set Clock1.TimerEnabled to false

when Start.Click
do
  if compare texts Start.Text == PAUSE
  then
    set Clock1.TimerEnabled to false
    set Start.Text to START
  else
    set Clock1.TimerEnabled to true
    set Start.Text to PAUSE
    set BluetoothClient1.DelimiterByte to 10
    set Value_Title.Text to VALUE =
```

```
when Clock1.Timer
do
  if call BluetoothClient1.BytesAvailableToReceive >= 0
  then
    set global FSRvalue to call BluetoothClient1.ReceiveText numberOfBytes -1
    set Value_Num.Text to get global FSRvalue
```

35. You will now have to repeat the process of building the .apk file and downloading it to the smart device.
36. When you press on the FSR, it may take a few seconds before you see values appear on the app.

