Script Reference

1.0 Unity Scripting

after making sure that your Bluetooth module isworking correctly, you can start using the following methods :

BtConnection.connect()

search your paired devices for a device named "HC-05" or a specified mac address _ you can change the name using BtConnection.moduleName("Name of your Bluetooth module") or you can instead use the MAC address using BtConnection.moduleMac("MAC")_ then if it finds the device it will create a connection.

Returns Integer

Integer > 0 when everything work , Integer < 0 when it failed.

Integer	
1	the connection process started.
-1	Bluetooth isn't enabled.
-2	the device doesn't support Bluetooth

BtConnection.close()

close the connection and free all used data.

BtConnection.isConnected()

returns true if there's a connection otherwise false, it's a good practice to check the connection before trying to do any communication.

BtConnection.available()

returns true if there is a data in the buffer, else false.

BtConnection.read()

Returns the last line sent by your Bluetooth Module. A line is represented by zero or more characters followed by `'\n'`, `'\r'`, `"\r\n"` or the end of the reader. The returnedstring does not include the newline sequence.

- the plugin will rewrite over any data in the buffer every time you call it, so you should hold it in a temporary variable :

  string temp = BtConnection.read();

  then you can use it to check if there's a new data to process.

  if(temp.length > 0) // process data

BtConnection.readBuffer(int Length)

returns Bytes array (Byte []), it will return the last bunch of bytes sent by your Microcontroller asbytes array of max length equal to the **Length** parameter. , so it will terminate if the determined length has been read.

- the plugin will rewrite over any data in the buffer every time you call it, so you should hold it in a temporary variable :

  byte [] temp = BtConnection.read();

  then you can use it to check if there's a new data to process.

  if(temp.length > 0) // process data

BtConnection.readBuffer(int Length, byteterminator)

same as BtConnection.readBuffer(int Length), butThe function terminates if the Terminator byte is detected, so it guarantees you get what you want, instead of reading any available bunch of bytes

BtConnection.stopListen()

stop Listening, use it when you don't need to read data, it will enhance performance. the plugin will start listen again whenever you call read() or readBuffer().

BtConnection.sendString("your message")

sends a string to your Bluetooth Module. it will add '\n' to the last of your string, which means new line.

BtConnection.sendChar('H')

sendsonechar to your Bluetooth Module, uses ASCII table chars only, don't use it to send bytes.

BtConnection.sendBytes(byte [] Buffer)

sends a buffer of bytes.

BtConnection.isSending()

returns true if the plugin still sending data otherwise false.

BtConnection.moduleName("Module Name")

the plugin uses the name "HC-05" as the default name for your module, you can change that using this method, it's important to make sure that you provide the exact same name, otherwise it will not find your module.

BtConnection.moduleName(String MacAddress)

the plugin uses the name "HC-05" as the default name for your module, you can change that by providing your Device's MAC address.

- the parameter MacAddress uses a hexadecimal 6 numbers separated by colons "MM:MM:MM:SS:SS:SS"
- for example : "00:A0:C9:14:C8:29".

BtConnection.controlData()

returns Integer that represent the status of the whole process.

returns Integer

Integer > 0 when everything work , Integer < 0 when it failed

Integer	Meaning
1	Connected
2	Disconnected after calling BtConnetion.cose()
-1	found your Bluetooth Module but unable to connect to it.
-2	Bluetooth module with the name or the MAC you provided can't be found
-3	Connection Failed, usually because your Bluetooth module is off
-4	error while closing
-5	error while writing
-6	error while reading

BtConnection.readControlData()

returns the string in the Meaning section of the previous table, instead of an integer.

1.1 Checking and Enabling Bluetooth Adabter

before starting any communication request, you need to make sure Bluetooth adapter is on, and if not to ask the user to turn it on. BtConnection has two methods for that :

BtConnection.isBluetoothEnabled()

returns true if Bluetooth is enabled otherwise false.

BtConnection.askEnableBluetooth()

A dialog will appear requesting user permission to enable Bluetooth, as shown in Figure 1. If the user responds "Yes," the system will begin to enable Bluetooth and focus will return to your game once the process completes (or fails).

BtConnection.enableBluetooth()

Turn on the local Bluetooth adapter, returns true to indicate adapter startup has begun, or false on immediate error.

- **requesting user permission to enable Bluetooth after clicking a button :**

```
//C#
if( GUILayout.Button ("Connect")){
            if (!BtConnection.isBluetoothEnabled ()){
                        BtConnection.askEnableBluetooth();
            } else BtConnection.connect();
      }
```

- **if you want to turn Bluetooth on directly when clicking connect, without asking the user :**

```
//C#
bool startConnection = false;
void OnGUI() {
      if( GUILayout.Button ("Connect")){
            startConnection = true;
      }
      if ( startConnection ){
            if (!BtConnection.isBluetoothEnabled ()){
                  BtConnection.enableBluetooth();
            } else {
                  BtConnection.connect();
                  startConnection = false;
            }
      }
}
```

2.0 Arduino Scripting

you can use your Bluetooth module with most of the available microcontrollers, but I'm going to talk about Arduino, because it is so popular. the followings are the most used methods for serialcommunication :

Serial.begin()

opens serial port and sets data rate. with Bluetooth modules HC-05 and HC-06 use9600 bps _Serial.begin(9600)_.

Serial.available()

Get the number of bytes (characters) available for reading from the serial port.

Serial.println()

sends line of string, so you can read it using BtConnection.read().

Serial.read()

use it when you send strings or chars.

Serial.readBytes()

use it with BtConnection.sendBytes().

Example (Reading data from unity)

```
byte incoming;

void setup() {
      Serial.begin(9600);
}

void loop() {
      if (Serial.available() > 0){
            incoming = Serial.read();
            // do whatever you want with (incoming)
      }
}
```