# Bluetooth and Serial

On a coding level, bluetooth works the same as standard serial communication, with the board connected to a bluetooth module that wirelessly transfers data. Establishing the physical connection is done through the bluetooth module and the connected device. The most common bluetooth modules are HC-05 and HC-06.

Sadly most Arduino boards only have one hardware serial port, which in most cases is used by the USB connection. E.g on Arduino Uno, the serial port on pins 0(RX) and 1(TX) is the same as the hardware USB port, so they cannot be in use at the same time. Luckily there are workarounds to this problem, primarily the libraries SoftwareSerial.h and AltSoftSerial.h.

## Using more than one serial

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(RX_pin, TX_pin);
```

After initialising a SoftwareSerial object, you can use the same methods as normal serial with it, e.g `mySerial.print(data)`, `mySerial.read()` and `mySerial.available();`
Any pins (except pin 13) can be used to create a SoftwareSerial connection, but only one software serial port can receive data at the same time. SoftwareSerial does not have any negative impact on hardware serial ports, like the USB connection.

## Simultaneous serial transfer

AltSoftSerial.h must be manually installed via the Arduino IDE library manager.

```
#include <AltSoftSerial.h>
AltSoftSerial altSerial; //Pins are assigned automatically
```

Always uses pins 9(TX) and 8(RX), also disables PWM and timer functionality of pin 10 on an Arduino Uno.

**Good to know:**

- Both libraries can be sensitive to interrupt usage by other libraries. E.g if a servo is attached using `Servo.h`, receiving serial data may cause unexpected movement in the servo.
- A higher baudrate minimizes the impact SoftwareSerial has on other interrupt-based libraries.
- Long interrupt sequences should not be used if any of these libraries are in use, since data might get lost.
- Both libraries and the hardware serial port can all be used at the same time, for effectively 3 simultaneous connections, but the transfer baud rates have to be chosen wisely. Scroll down to the bottom of this page for instructions.

**Examples:**

```
#include <SoftwareSerial.h>

SoftwareSerial mySerial(9,10);

void setup() {
  Serial.begin(19200);
  mySerial.begin(19200);
}

void loop() {
  char c;

  //If anything is input into the serial monitor, send it to the
  //device connected via SoftwareSerial
  if (Serial.available()) {
    c = Serial.read();
    mySerial.print(c);
  }

  //If anything is received from the connected device, print it to
  //the serial monitor
  if (mySerial.available()) {
    c = mySerial.read();
    Serial.print(c);
  }
}
```