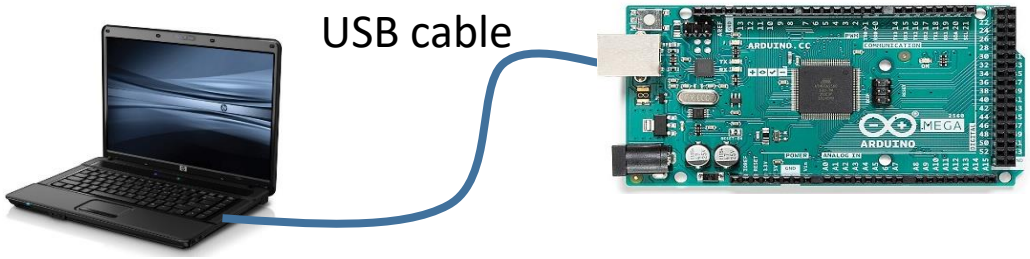
More on Serial Communication

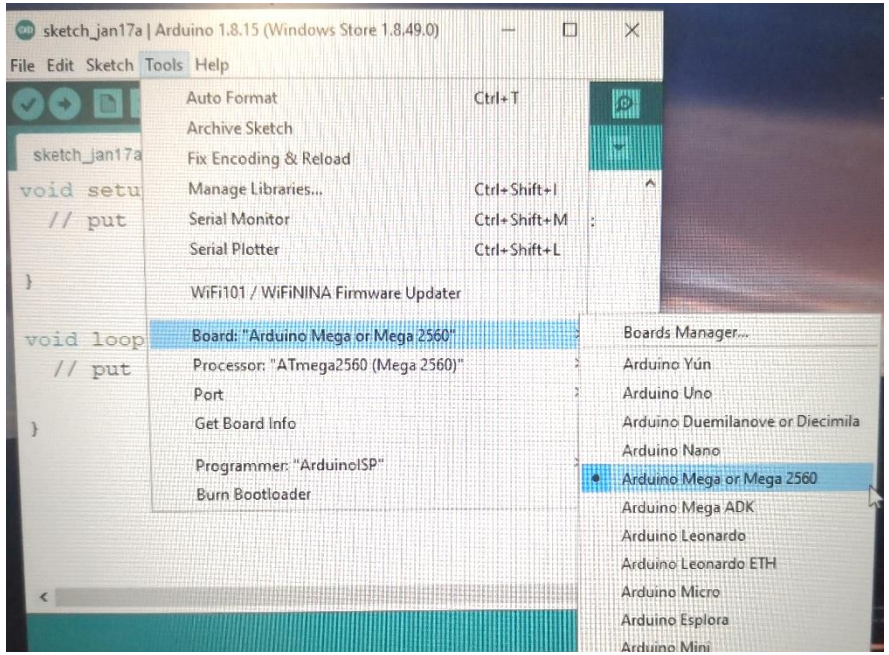
Serial Communication between Arduino and PC (via USB)

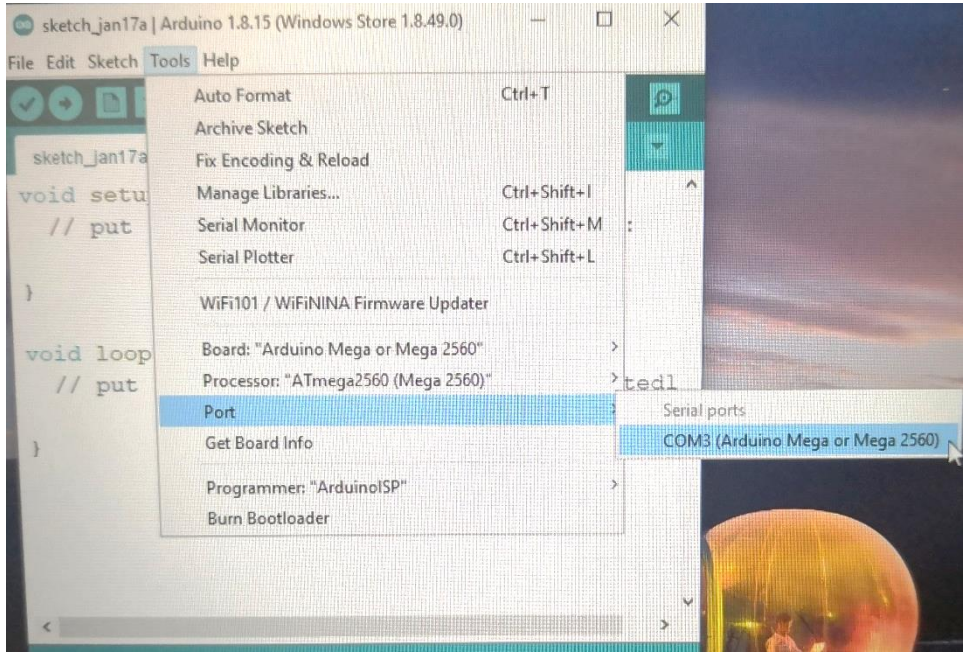
Department of Electrical Engineering
香港城市大學
City University of Hong Kong

- Serial communication can be established by connecting the Arduino to your PC.



- You'll need to select the correct Arduino type and the Port in the IDE
- Eg. The one shown is a MEGA 2560 and COM3 (this port is automatically set by PC)

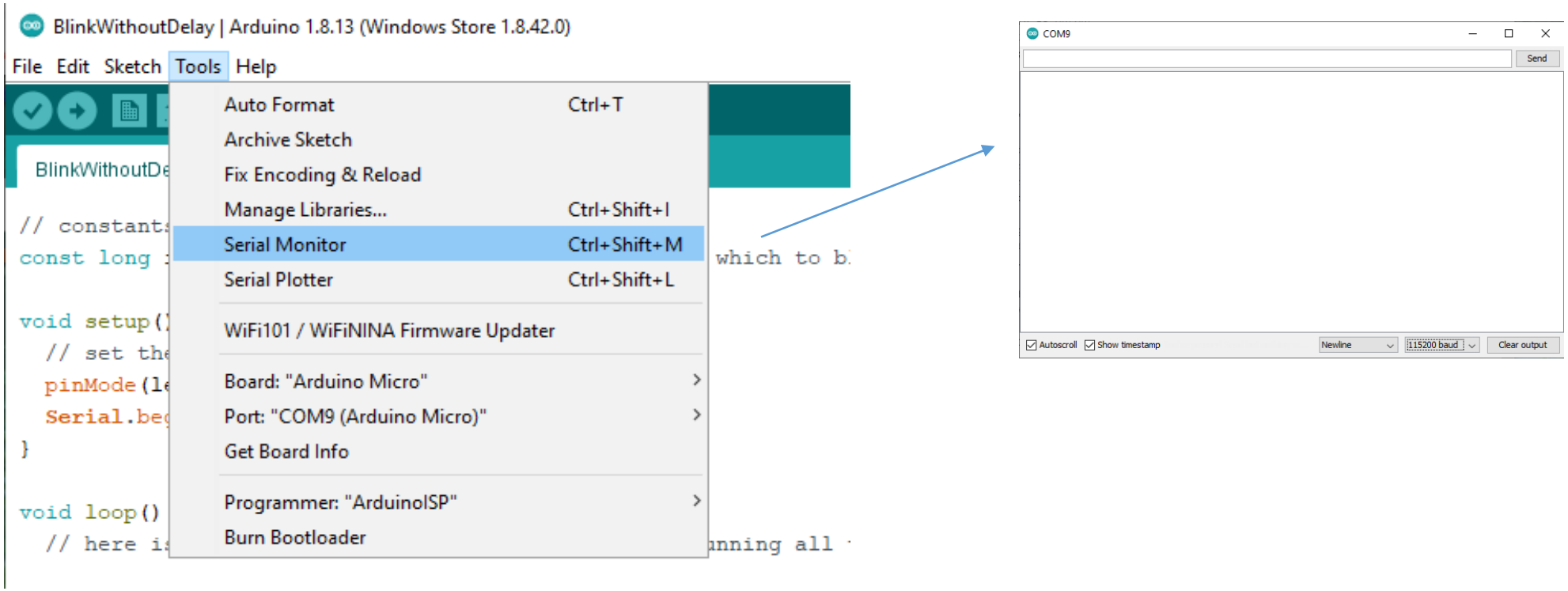




Serial Communication between Arduino and PC (via USB)

Department of Electrical Engineering
香港城市大學
City University of Hong Kong

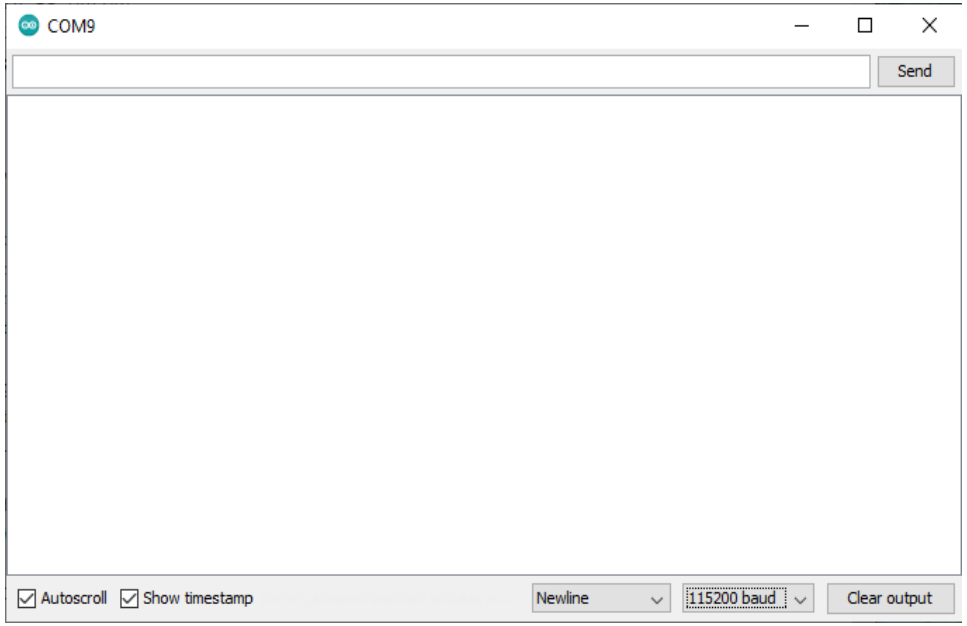
- To receive / send data at PC side, you'll need to open the Serial Monitor (work as a terminal)



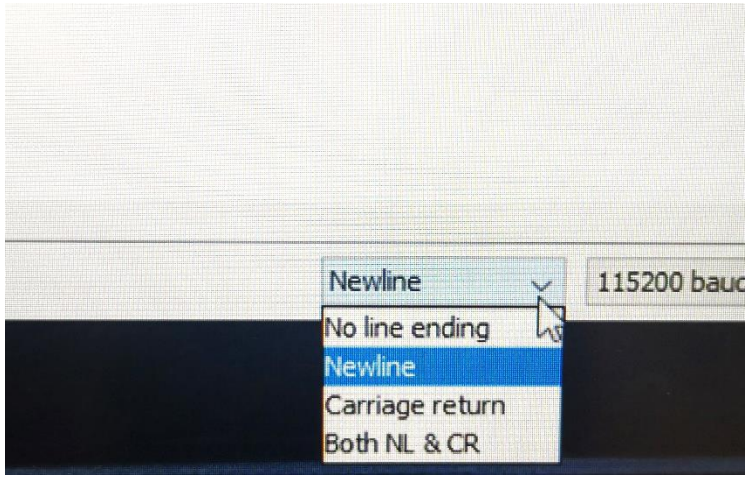
Department of Electrical Engineering
香港城市大學
City University of Hong Kong

Serial Monitor

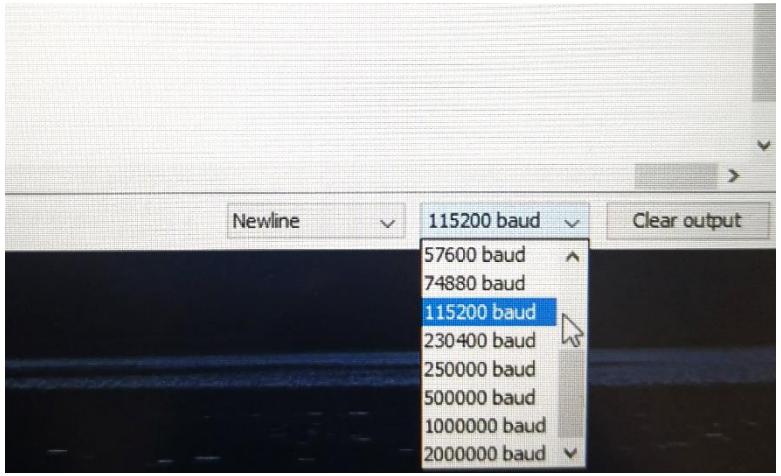
- Menu 1: Determine how to end the sending string; whether special character NL(newline) or CR (Carriage return) are appended at the end or not.
- Menu 2: Set the baud rate (data transfer speed)



Menu 1



Menu 2



Program with Serial Communications

- Serial is a class (containing many functions)

https://www.arduino.cc/reference/en/language/functions/communication/serial/

- Serial.begin(val)
  - One parameter
    - val: baud rate, eg. 9600, 57600 …
  - It is used to set the baud rate for the serial communications. Need to match with the one set in Serial Monitor

```
//----- SETUP SERIAL PORT -----
Serial.begin(9600);
while(!Serial)
  ;           //Leonardo/Micro should wait for serial init
```

- Serial is true if the port is ready.
- Serial.end (): Disable serial communication

Department of Electrical Engineering
香港城市大學
City University of Hong Kong

Some functions for Sending to Serial Monitor (Arduino→PC)

- Sending text or string of texts

```
Serial.print("Hello");         //Write string no new line
Serial.println(my_variable);  //Write a value with line break at end
```

- Sending Variable as ASCII

Serial.print(78, BIN) gives "1001110"

Serial.print(78, OCT) gives "116"

Serial.print(78, DEC) gives "78"

Serial.print(78, HEX) gives "4E"

Serial.println(1.23456, 0) gives "1"

Serial.println(1.23456, 2) gives "1.23"

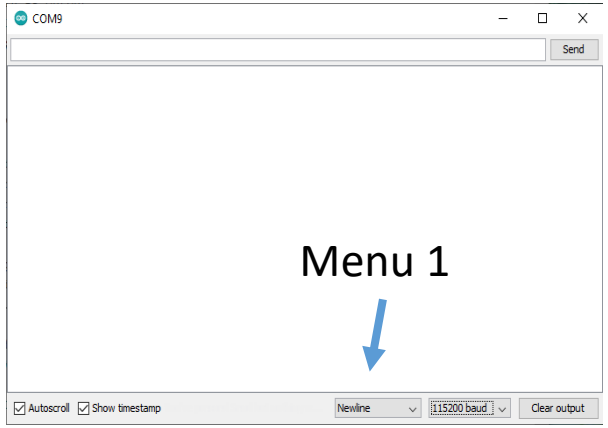
Serial.println(1.23456, 4) gives "1.2346"

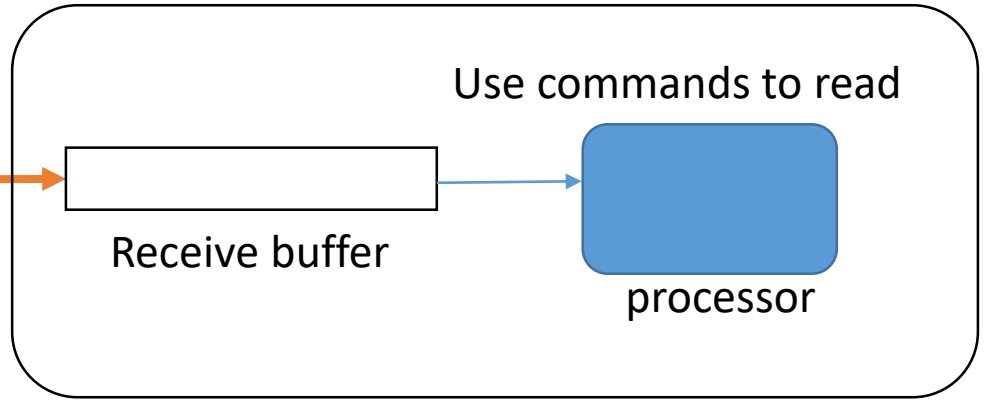
Default is two decimals.

Department of Electrical Engineering
香港城市大學
City University of Hong Kong

Receiving from Serial Monitor (PC →Arduino)

- Basic Concept about a communication channel





(i) ASCII codes are sent, which are interpreted as text characters

(ii) Menu 1 specifies how you end the sending texts.

(i) At Arduino side, the texts are stored in a receive buffer

Example:

If Newline is chosen in Menu 1, and “abc” is sent. At the receive buffer, there will be 4 characters, “abc” plus “the new line character”.

If No line ending is chosen in Menu 1, and “abc” is sent. At the receive buffer, there will be 3 characters, “abc”

Some functions

- Serial.available()
  - Get the number of bytes (characters) arrived and stored in the serial receive buffer (which holds 64 bytes).

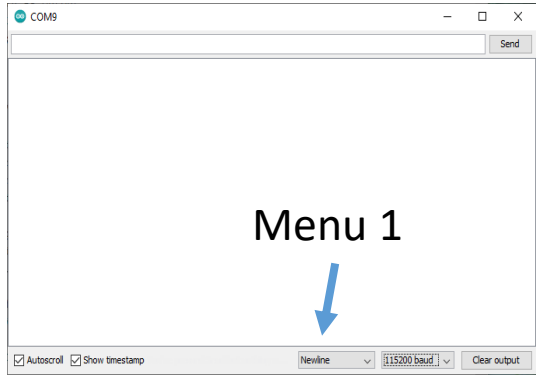
- Serial.parseInt()
  - Get the first valid integer number from serial buffer.
    - Parsing stops when no characters have been read for a configurable time-out value, or a non-digit is read;
    - If no valid digits were read when the time-out (see Serial.setTimeout()) occurs, 0 is returned;

- Serial.readString()
  - Read characters from serial buffer and output the string (a string of characters)

Program with Serial Communications

Department of Electrical Engineering
香港城市大學
City University of Hong Kong

Examples



(1) If Newline is chosen in Menu 1, and send “abc”

(2) If both NL &CR is chosen in Menu 1, and send “abc”

(3) baud rate does not match with the program

(4) If Newline is chosen in Menu 1, and send “123”

(5) No line ending is chosen in Menu 1, and send “123”


```
while (Serial.available() > 0)
{
    str1 = Serial.readString();
}
```

(1) Str1 becomes a string of 4 characters: “abc”+ NL

(2) Str1 becomes a string of 5 characters: “abc” + NL + CR

(3) Strange characters

```
while (Serial.available() > 0)
{
    a = Serial.parseInt()
}
```

(4) a = 123 and then a=0

(5) a = 123

Note: There are also other commands available.