

Arduino notes 2: Serial communication

This is not a tutorial, but a collection of personal notes to remember the essentials of Arduino programming. The program fragments are snippets that represent the essential pieces of code, in an example.
jean-claude.feltes@education.lu

Arduino web site:

<http://arduino.cc>

Serial:

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

Good examples and summary on the commands:

<http://forum.arduino.cc/index.php?topic=396450.0>

Receive a character via Serial

This lets the Arduino react to serial input.

The following example switches a LED on or off with the characters '1' and '0':

```
int led = 13;

void setup() {
  pinMode(led, OUTPUT);
  Serial.begin(115200);
}

void loop() {
  if (Serial.available()){
    char ch = Serial.read();
    if (ch == '1')
      digitalWrite(led, 1);

    if (ch == '0')
      digitalWrite(led,0);
  }
}
```

Receive an integer number via Serial, explicit programming

To communicate with the controller, a number is sent, terminated by a newline character '\n'.

To test this program with the serial monitor, activate the newline option:



The example program waits for a number input, multiplies it by 2 (to do something with it) and sends it back:

```
void setup() {
  Serial.begin(115200);
}

unsigned int mynumber=0; // Max value is 65535, use long for bigger numbers
```

```

char incomingByte;

void loop() {
  if (Serial.available() > 0) {
    mynumber = 0;          // delete previous number

    // read until '\n':
    while(1) {
      incomingByte = Serial.read();
      if (incomingByte == '\n') break;
      if (incomingByte == -1) continue;
      mynumber *= 10;
      mynumber = ((incomingByte - 48) + mynumber);
    }
    // Number received, now do something with it
    Serial.println(mynumber * 2);
  }
}

```

When a character is read, the previous number is first set to zero.

Then, as long as there is no newline character, bytes are received, converted from ASCII to integer (by subtracting 48), and added to mynumber that is shifted one decimal place to the left (by multiplying with 10) before the addition.

For a better overview we can define a function readNumber():

This waits for a number to be input (ending with newline)

<pre> Serialread_number_02 \$ readNumber 1 long x=0; 2 3 4 void setup() { 5 Serial.begin(115200); 6 } 7 8 void loop() { 9 x = readNumber(); 10 Serial.println(x * 2); 11 } 12 </pre>	<pre> Serialread_number_02 \$ readNumber 1 long readNumber(){ 2 long mynumber=0; 3 char incomingByte; 4 5 6 // read until '\n': 7 while(1) { 8 incomingByte = Serial.read(); 9 if (incomingByte == '\n') break; 10 if (incomingByte == -1) continue; 11 mynumber *= 10; 12 mynumber = ((incomingByte - 48) + mynumber); 13 } 14 15 return mynumber; 16 } </pre>
--	--

The disadvantage is that the program execution is blocked while it waits for input.

Receive a number via Serial, with parseInt / parseFloat

These functions provide a simple method of receiving:

```

long x=0;

void setup() {
  Serial.begin(115200);
}

void loop() {
  if (Serial.available()){

```

```
        x=Serial.parseInt();  
        // do something with x  
        Serial.println(x * 2);  
    }  
}
```

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

To receive a float number, use `parseFloat()`

<http://forum.arduino.cc/index.php?topic=236162.0>

<http://forum.arduino.cc/index.php?topic=396450.0>

<http://forum.arduino.cc/index.php?topic=225329.msg1810764#msg1810764>

Literature:

[1] Julien Bayle: C Programming for Arduino, Packt.
This is an excellent and very practical introduction