02.03 Serial input: introduction

Until now, we have always sent serial data from the controller to the PC. The purpose of this command is to send commands from the PC to the controller via the UART. We will do this by means of a mini protocol (=agreement on how to communicate). This command also has the additional purpose of interpreting and executing the commands.

Basic	
Command	**Description**
AT	Test AT startup
AT+RST	Restart module
AT+GMR	View version info
AT+GSLP	Enter deep-sleep mode
ATE	AT commands echo or not
AT+RESTORE	Factory Reset
AT+UART	UART configuration, [@deprecated]
AT+UART_CUR	UART current configuration
AT+UART_DEF	UART default configuration, save to flash
AT+SLEEP	Sleep mode
AT+RFPOWER	Set maximum value of RF TX Power
AT+RFVDD	Set RF TX Power according to VDD33

02.03 Serial input: command

Create an Arduino Sketch that:

- sets the UART to a baud rate of 9600 bits/second
- Configures 3 pins as output to drive an RGB LED.
  (Please refer to 01.02 blinky rgb)
- The following protocol implements

The protocol consists of 4 bytes each (all bytes are printable ASCII characters):

- The first byte can be 'r', 'g' or 'b'. This byte specifies which colour is to be controlled (red, green or blue).
- The second byte is always '='.
- The third byte can be '0' or '1'. It determines whether the LED must be on or off.
  (To know which LED: see first byte)
- The fourth byte is always the ASCII character for a new line: '\n'.

Example of the protocol ('omitted' for clarity see below):

```
r=1
```

Will turn on the red LED.

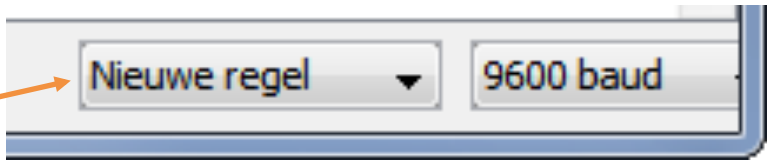
```
r=0
```

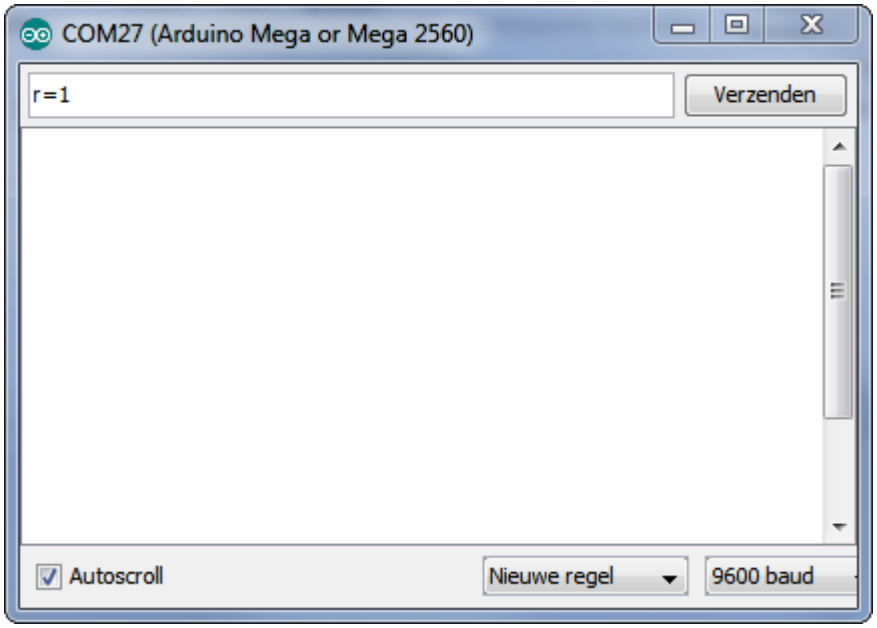
Will turn off the red LED.

```
b=1
```

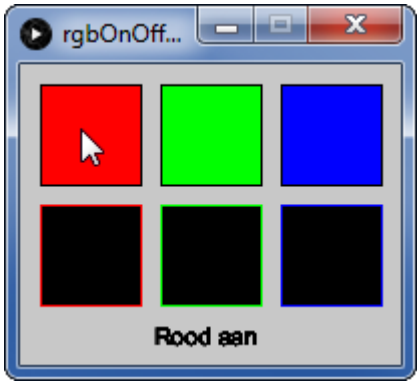
Will turn on the blue LED.

You can test this via the serial monitor of the Arduino IDE. **But make sure that the correct line end option is selected**:





A Processing application rgbOnOffOverSerial is also available which can control the commands:



Tip: check out the Serial. readBytesUntil function in the Arduino.cc reference.

Challenge: adjust the processing code (rgbOnOffOverSerial) so that you have 8 buttons: red, yellow, green, cyan, blue, magenta, white, black
The buttons set the colour without having to change the protocol.