



PYTHON AND ARDUINO: A TALE OF SNAKES AND KINGS

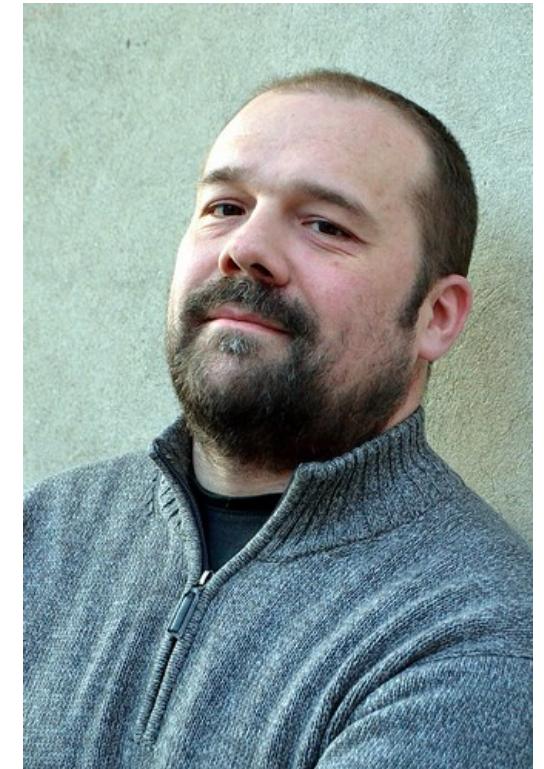
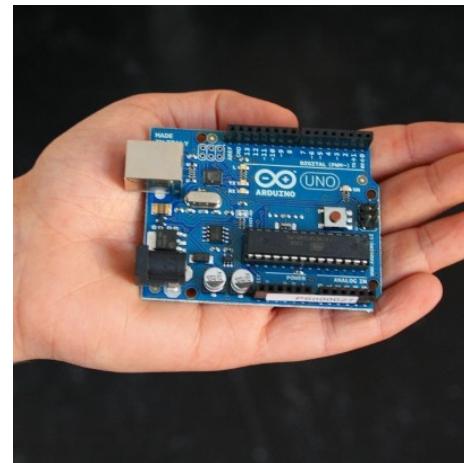
EuroPython 2012 - Florence

Alessandro Pasotti
a.pasotti@itopen.it
@elpaso

Davide Corio
davide.corio@domsense.com
@davidecorio

Introduction

who
why
what



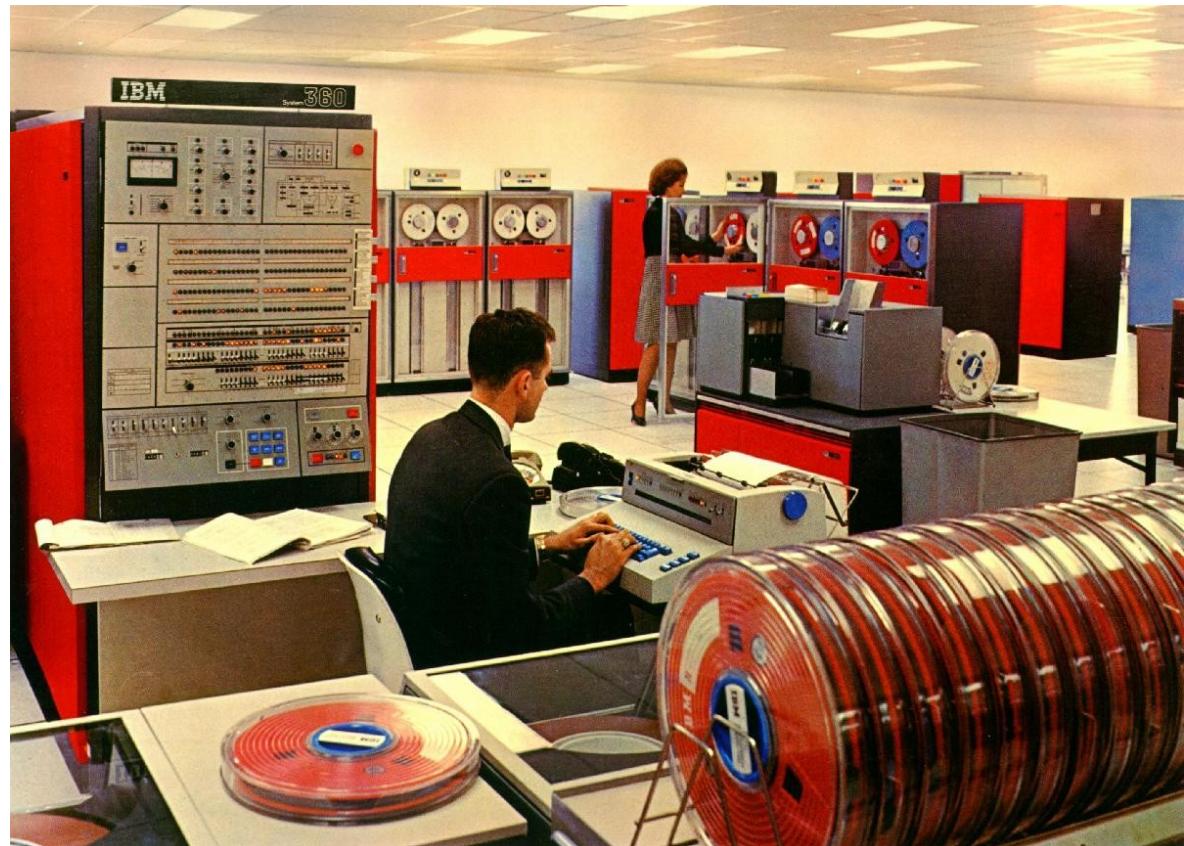
In the beginning

In the beginning was the abacus...
Technology started on real things
you could touch



Being digital

From Big-Blue to Micro-Soft



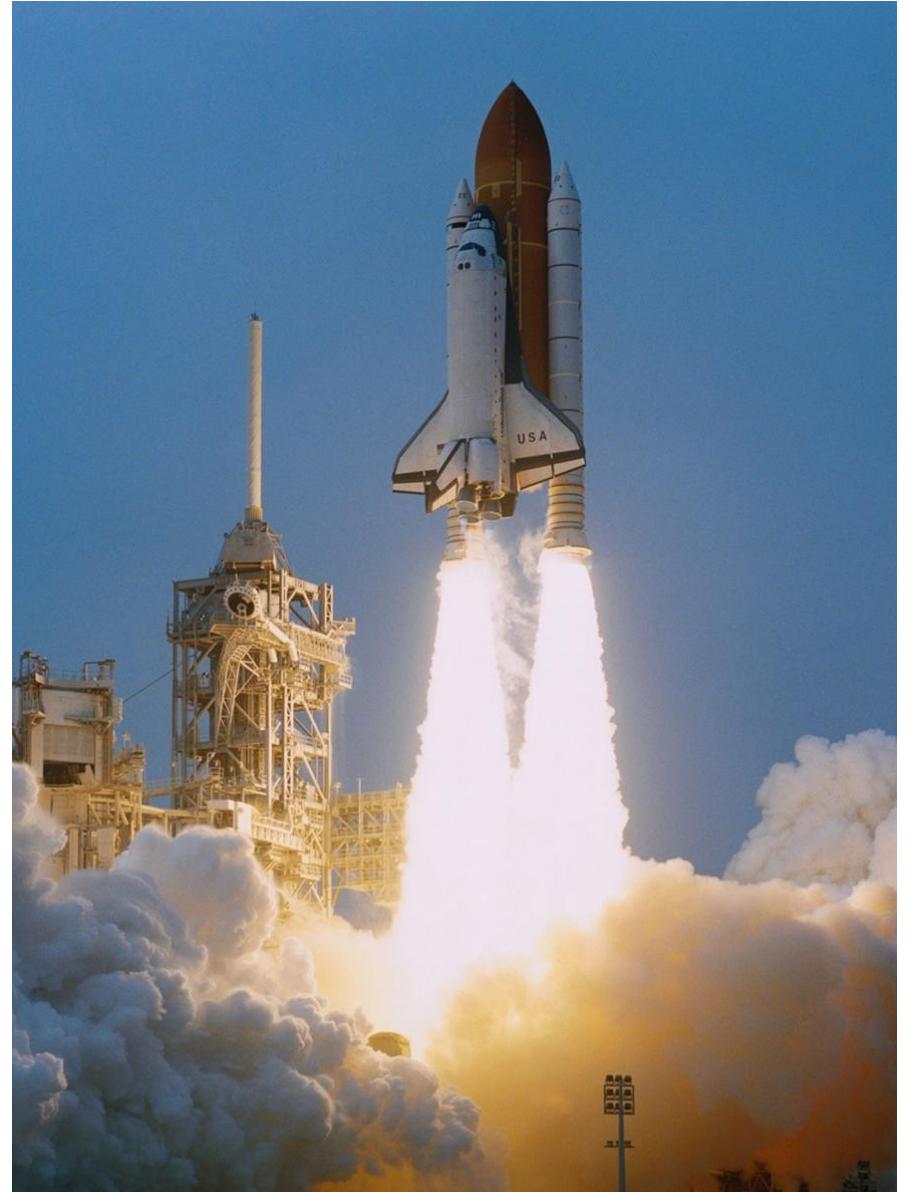
Touching the Sky

Alice in Metaphorsland:
desktops, trash-bins and wizards



Leaving Earth

Abstraction can be
addictive



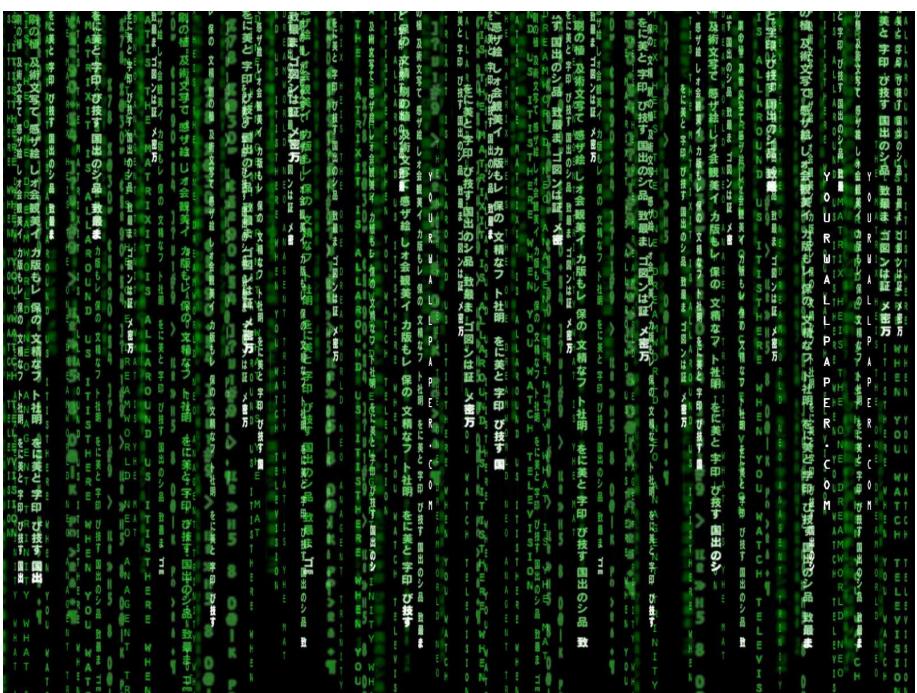
Unusable interfaces



It's all about Interactivity

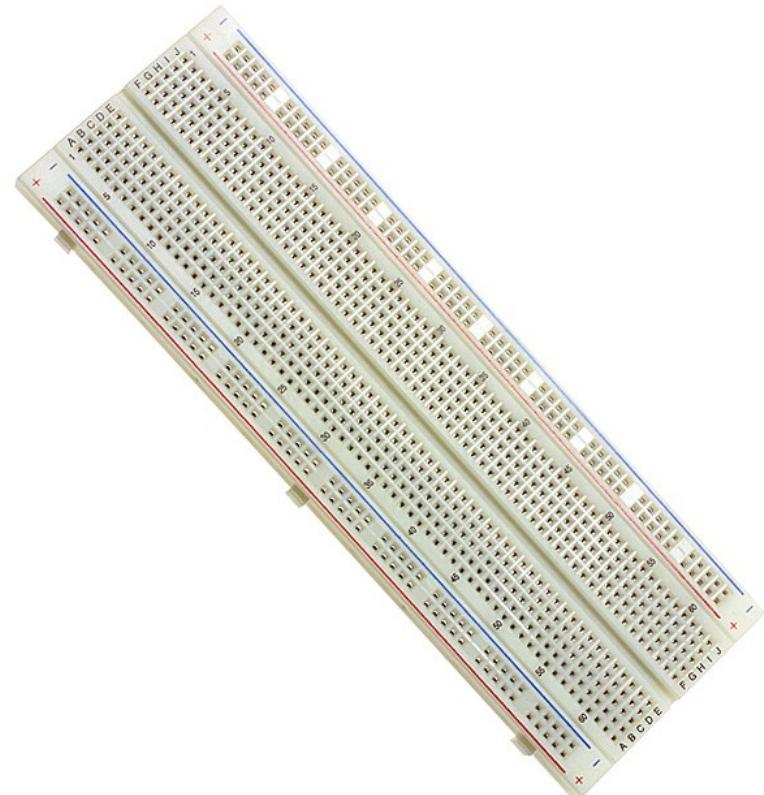
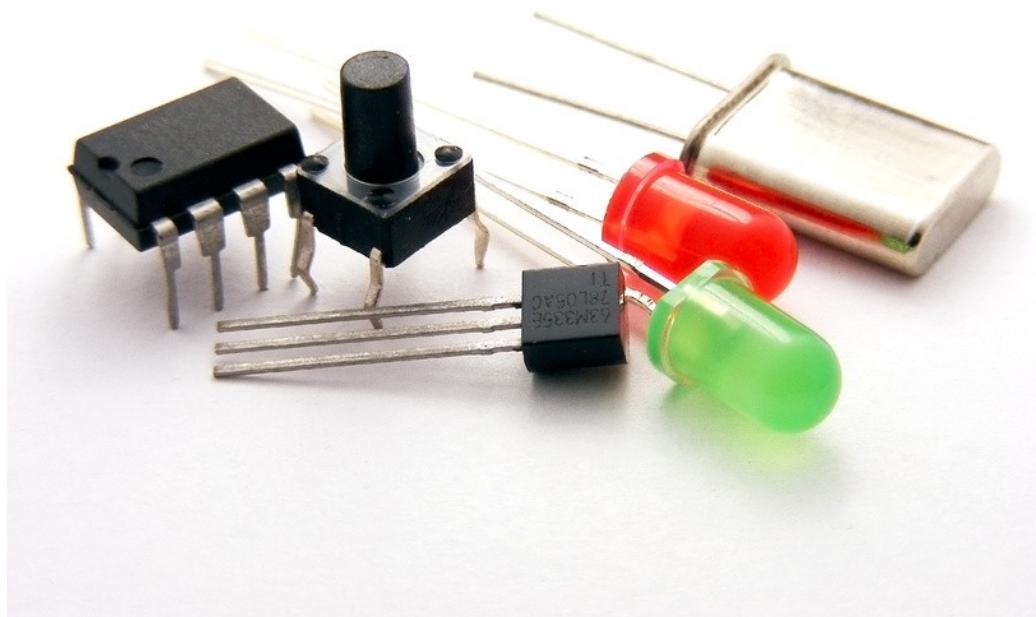


Sniffing the stacktrace



Being real

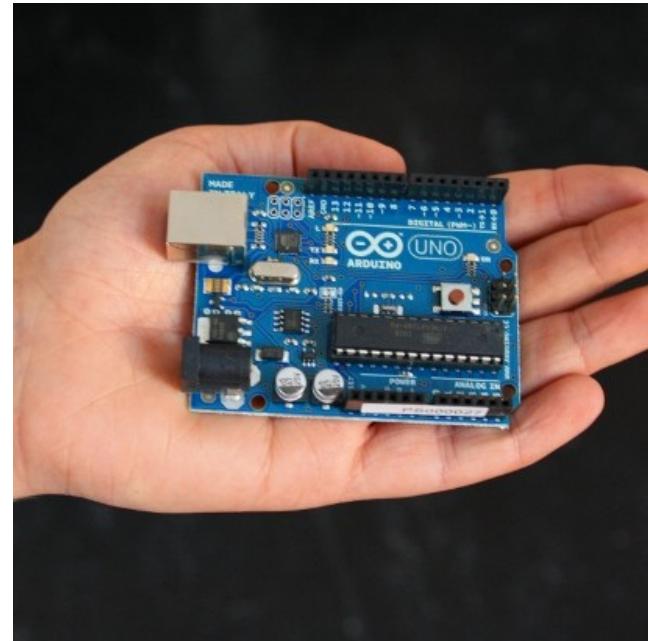
Physical computing: low cost MCU's and electronic components



Here comes the King

Arduino's revolution

minimal
easy to use
cheap
from artists
Made in Italy
open hardware



What's inside?

hw: device: electronic prototyping board

sw: bootloader

sw: libraries

sw: IDE

community

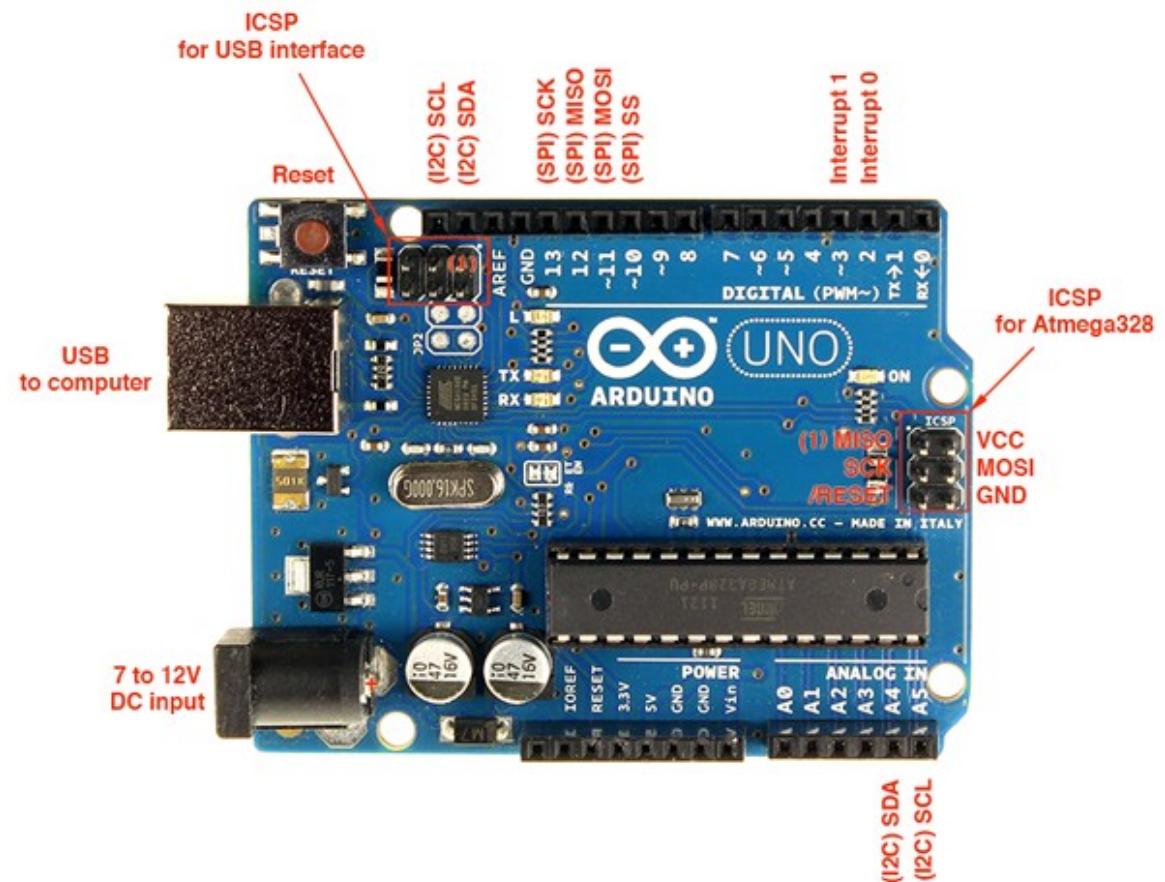
lot of manuals

lot of examples



The board

Arduino UNO



MCU

Atmega 328

RAM 2 KB

ROM 32 KB

EEPROM 1 KB

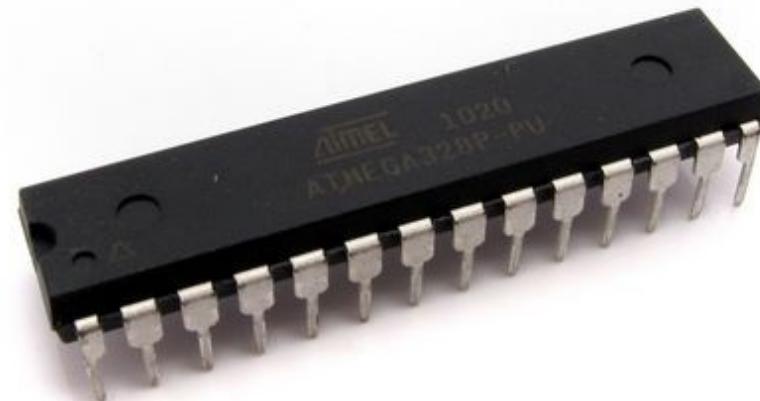
14 digital pins

6 PWM capable

6 analog inputs

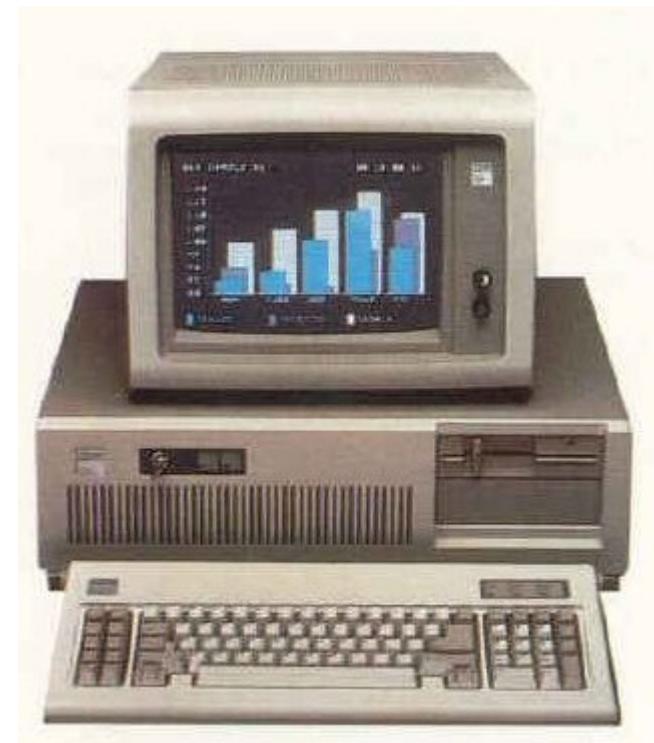
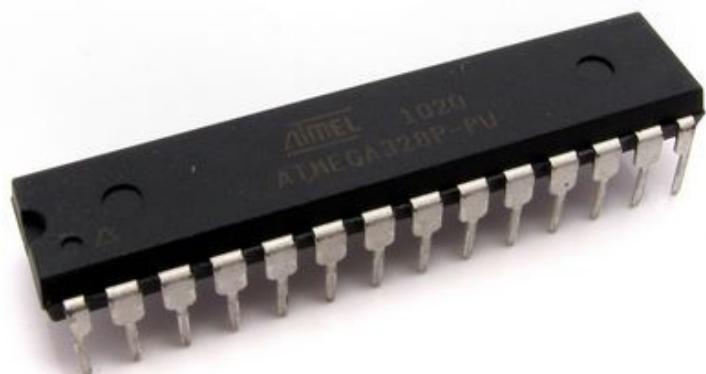
Up to 16 MHz

avr-gcc



Power to the people!

1982 – IBM 80286



Splitting the byte

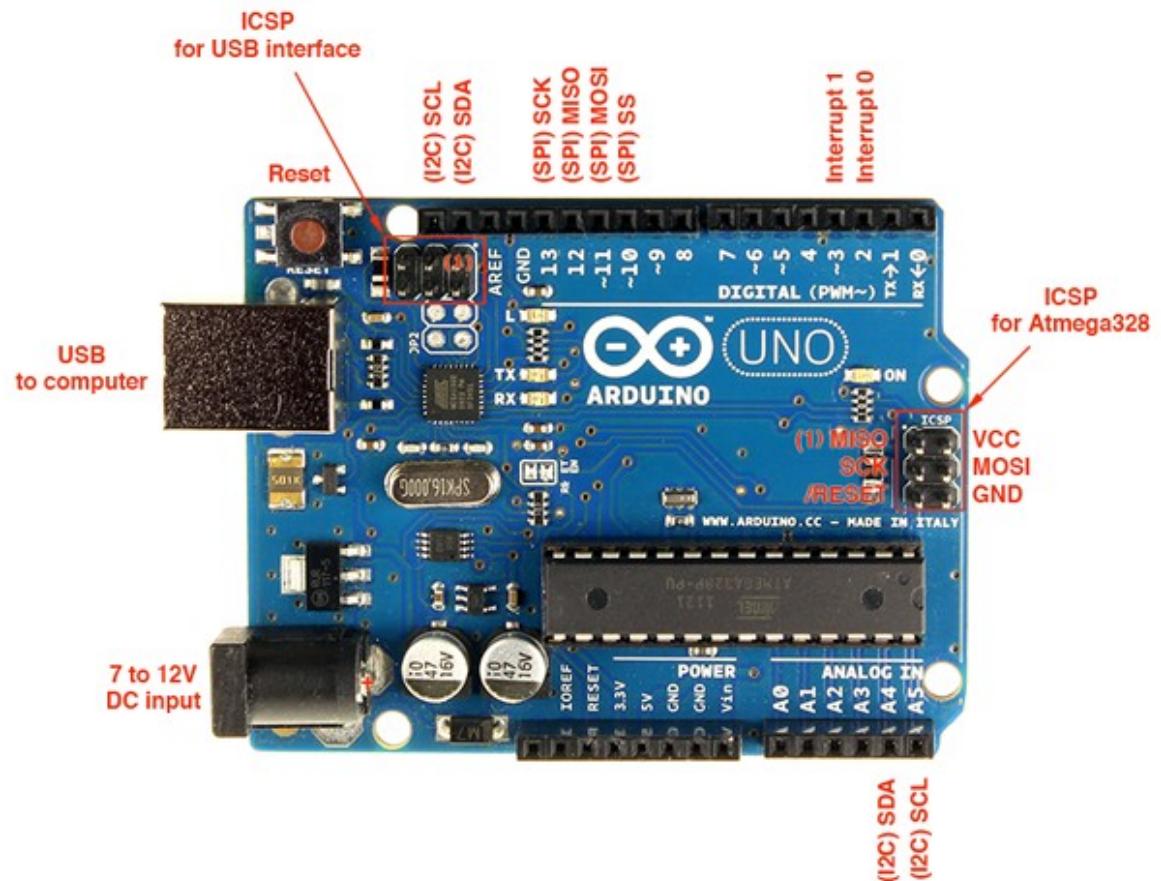
Atmega 328 Registers and ports

| | | | |
|--------------------------|----|----|------------------------|
| (PCINT14/RESET) PC6 | 1 | 28 | PC5 (ADC5/SCL/PCINT13) |
| (PCINT16/RXD) PD0 | 2 | 27 | PC4 (ADC4/SDA/PCINT12) |
| (PCINT17/TXD) PD1 | 3 | 26 | PC3 (ADC3/PCINT11) |
| (PCINT18/INT0) PD2 | 4 | 25 | PC2 (ADC2/PCINT10) |
| (PCINT19/OC2B/INT1) PD3 | 5 | 24 | PC1 (ADC1/PCINT9) |
| (PCINT20/XCK/T0) PD4 | 6 | 23 | PC0 (ADC0/PCINT8) |
| VCC | 7 | 22 | GND |
| GND | 8 | 21 | AREF |
| (PCINT6/XTAL1/TOSC1) PB6 | 9 | 20 | AVCC |
| (PCINT7/XTAL2/TOSC2) PB7 | 10 | 19 | PB5 (SCK/PCINT5) |
| (PCINT21/OC0B/T1) PD5 | 11 | 18 | PB4 (MISO/PCINT4) |
| (PCINT22/OC0A/AIN0) PD6 | 12 | 17 | PB3 (MOSI/OC2A/PCINT3) |
| (PCINT23/AIN1) PD7 | 13 | 16 | PB2 (SS/OC1B/PCINT2) |
| (PCINT0/CLKO/ICP1) PB0 | 14 | 15 | PB1 (OC1A/PCINT1) |

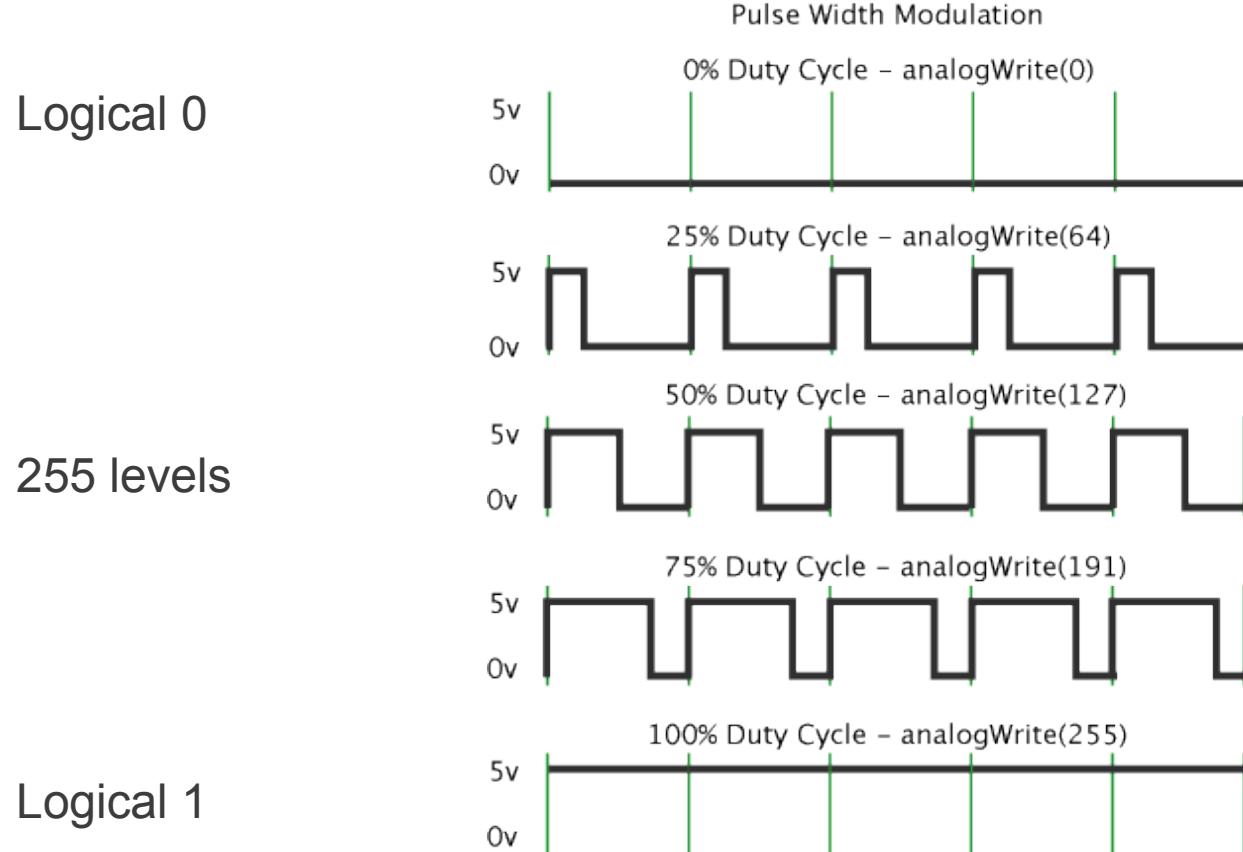
Pins

Digital IO
PWM (digital)

Analog inputs

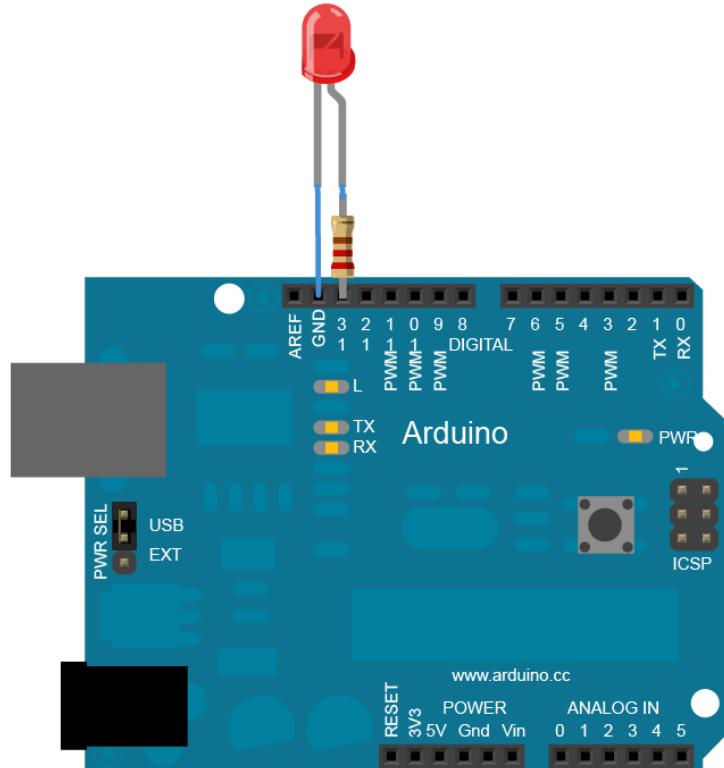


PWM (pulse width modulation)



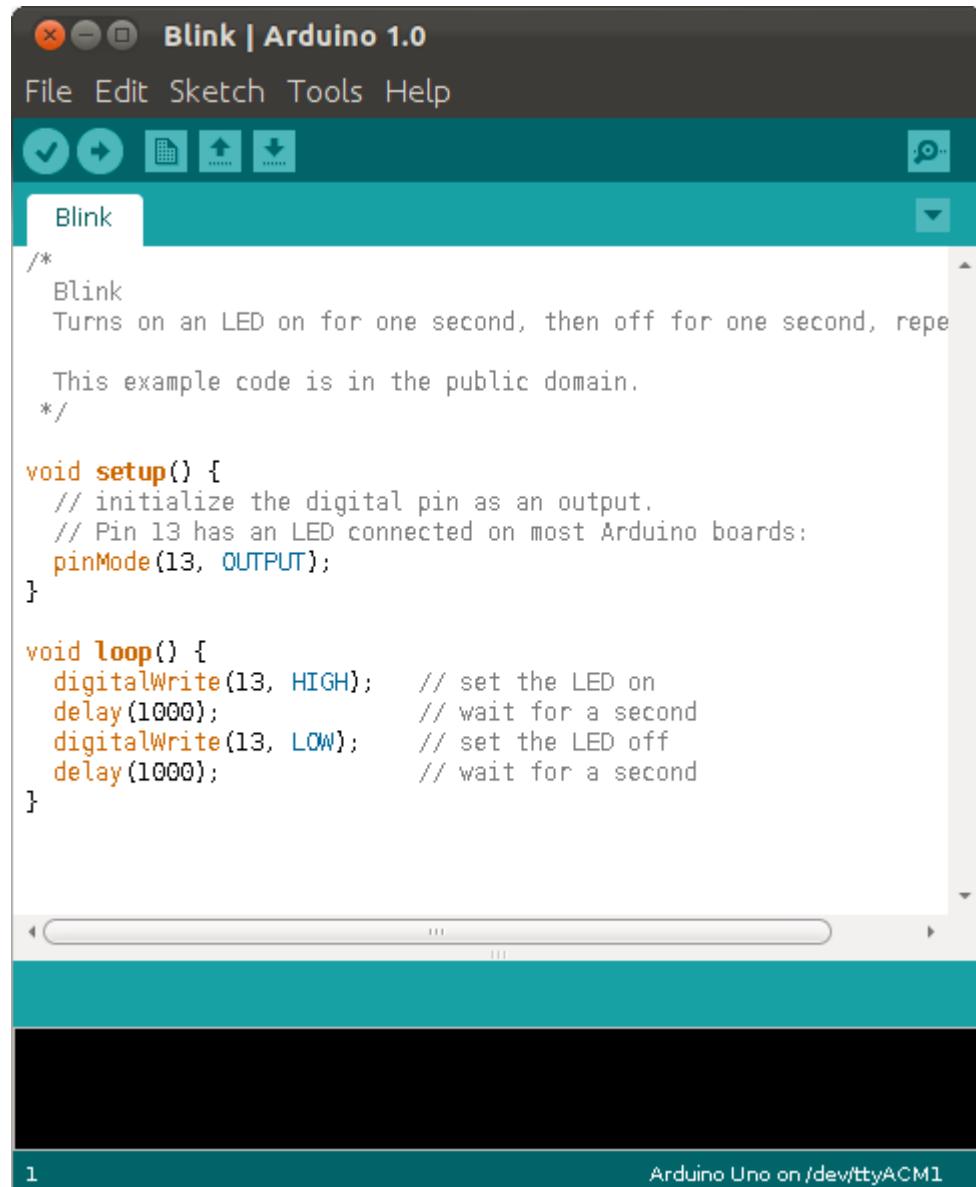
C programming

```
1 #define F_CPU 10000000UL
2 #include <avr/io.h>
3 #include <avr/delay.h>
4
5 void delayms(uint16_t millis) {
6     uint16_t loop;
7     while ( millis ) {
8         _delay_ms(1);
9         millis--;
10    }
11 }
12
13 int main(void) {
14     /* set PB5 (digital pin 13)
15      to output */
16     DDRB |= 1<<PB5;
17     while(1) {
18         PORTB &= ~(1<<PB5); /* pin off */
19         delayms(100);
20         PORTB |= 1<<PB5; /* pin on */
21         delayms(900);
22     }
23     return 0;
24 }
```



Programming

setup
loop
blink!



The screenshot shows the Arduino IDE interface with the title bar "Blink | Arduino 1.0". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for save, upload, and other functions. The main window displays the "Blink" sketch. The code is as follows:

```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeating
 * This example code is in the public domain.
 */

void setup() {
    // initialize the digital pin as an output.
    // Pin 13 has an LED connected on most Arduino boards:
    pinMode(13, OUTPUT);
}

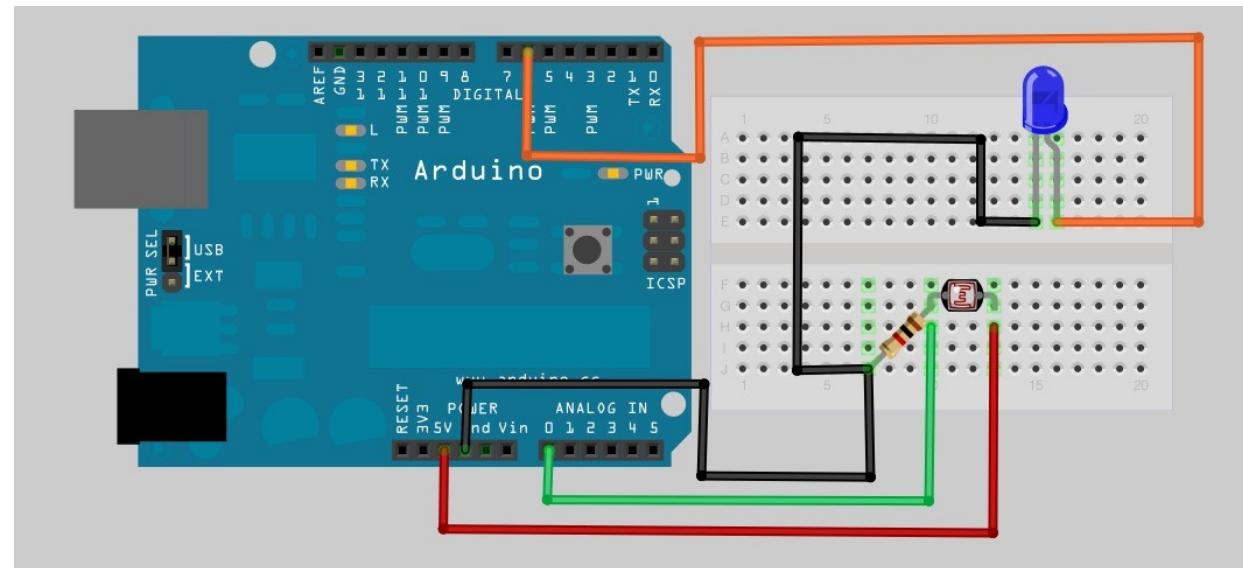
void loop() {
    digitalWrite(13, HIGH);      // set the LED on
    delay(1000);                // wait for a second
    digitalWrite(13, LOW);       // set the LED off
    delay(1000);                // wait for a second
}
```

The status bar at the bottom indicates "Arduino Uno on /dev/ttyACM1".

King's audience

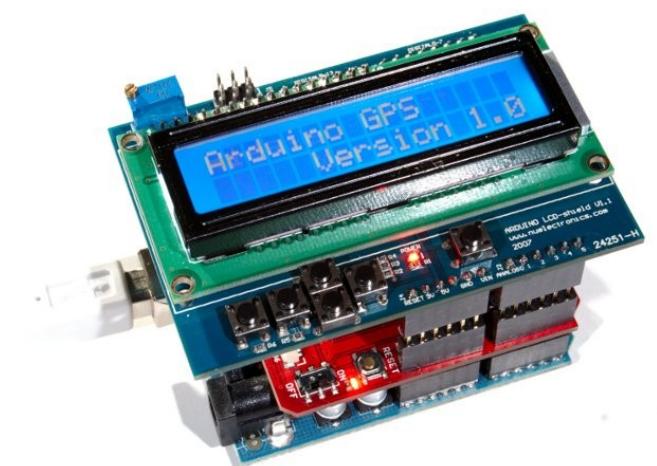
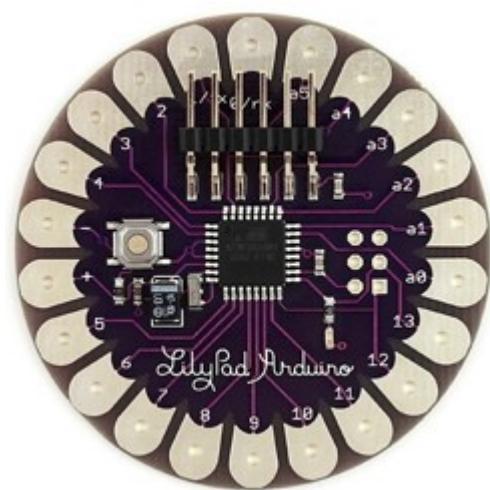
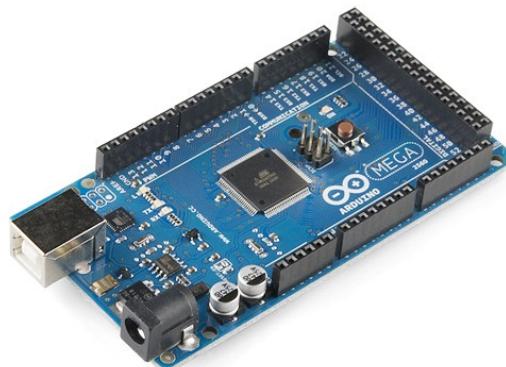
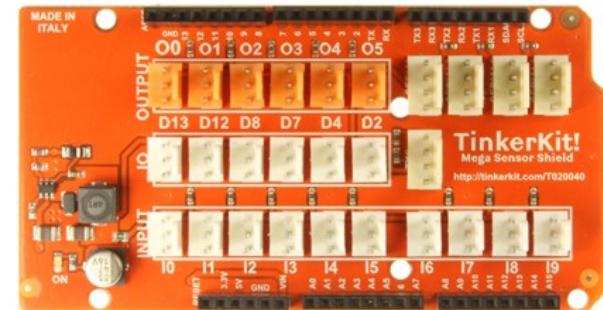
Arduino talks and listens

serial
sensors
RF
ethernet



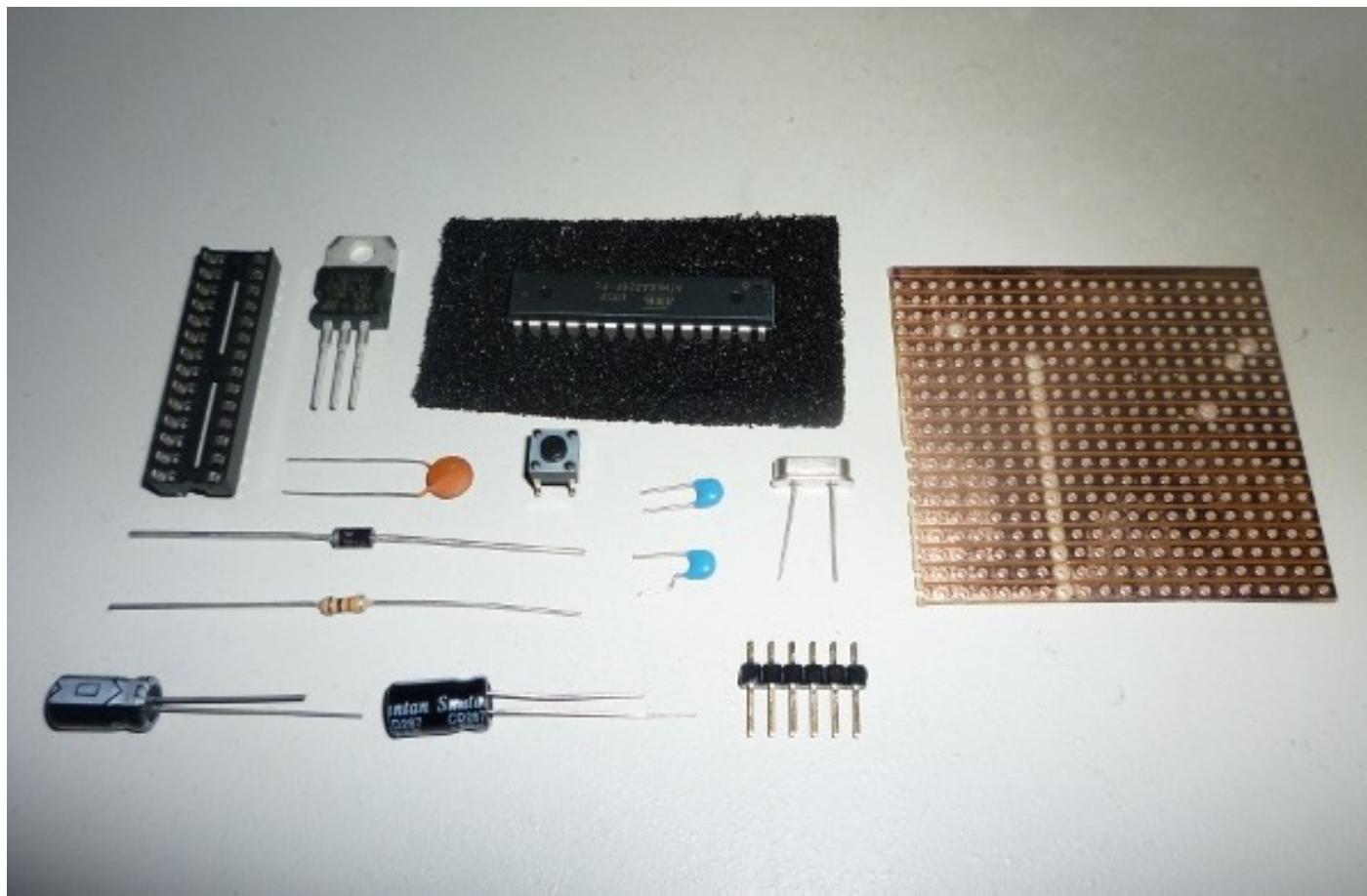
At the king's court

shields
clones



DIY Arduino

< 10 €



Snake's attack

ASCII messages
compact protocols

```
>>> import pyfirmata  
>>> print "yeah, it works!! :-D"
```

ASCII snake

```
>>> import serial
>>> ser = serial.Serial('/dev/ttyACM0', 9600)
>>> while 1:
...     ser.readline()
'1 Hello world!\r\n'
'2 Hello world!\r\n'
'3 Hello world!\r\n'
```

```
>>> import serial
>>> ser = serial.Serial('/dev/ttyACM0', 9600)
>>> ser.write('5')
```

The firmata protocol

http://firmata.org/wiki/Main_Page

pyfirmata to the rescue

<https://bitbucket.org/tino/pyfirmata/src>

PyFirmata example

```
1 import pyfirmata  
2  
3 PIN = 13 # Pin 13 is used  
4 DELAY = 2 # A 2 seconds delay  
5  
6 PORT = '/dev/ttyACM0'  
7  
8 # Creates a new board  
9 board = pyfirmata.Arduino(PORT)  
10  
11 # Loop for blinking the led  
12 while True:  
13     # Set the LED pin to 1 (HIGH)  
14     board.digital[PIN].write(1)  
15     board.pass_time(DELAY)  
16     # Set the LED pin to 0 (LOW)  
17     board.digital[PIN].write(0)  
18     board.pass_time(DELAY)  
19
```

Embedded snakes

Python on a chip



<http://code.google.com/p/python-on-a-chip/>

Other projects

Python Arduino Prototyping API

<https://github.com/HashNuke/Python-Arduino-Prototyping-API>

Python 2B (ASCII protocol)

<http://arduino.cc/playground/Code/2B>

OpenERP arduino controller

Demo



<https://github.com/elpaso/arduinocontroller>

http://www.youtube.com/watch?feature=player_embedded&v=MkxK9bHPZxQ



THANKS!

Alessandro Pasotti
a.pasotti@itopen.it
@elpaso

Davide Corio
davide.corio@domsense.com
@davidecorio