Sensing and acting on the environment
using micro-controllers

Xavier Serpaggi
<serpaggi@emse.fr>

Master Web Intelligence
Master Machine Learning and Data Mining

Novembre 2013

Micro-controllers

- A computer on chip
- Little computing power (10-20 MIPS)
- Few memory (some kB)
- Little power consumption
- A lot of ports to plug various things

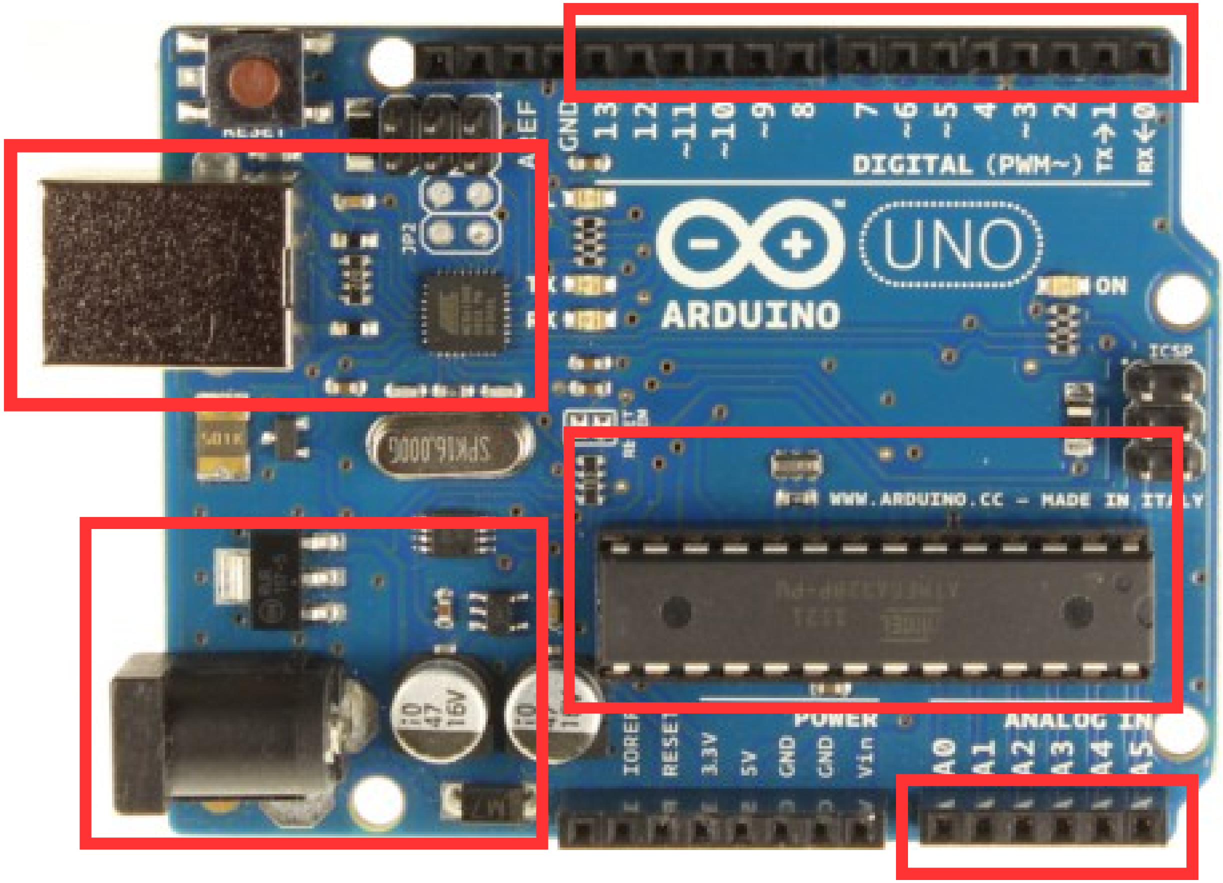
They are good friends for ambient computing projects

Arduino

- An open-source project (both hardware and software)
- AVR ATMega micro-controller
  - 8 bits, 16 MHz
  - 32 kB flash program memory, 2 kB SRAM data memory, 1 kB EEPROM
  - Digital I/O, ADC, PWM
  - Hardware and software interrupts
  - Timers
  - ...
- Cheap and robust
- Easy to use IDE
  - Look and feel taken from Processing
  - API based on Wiring

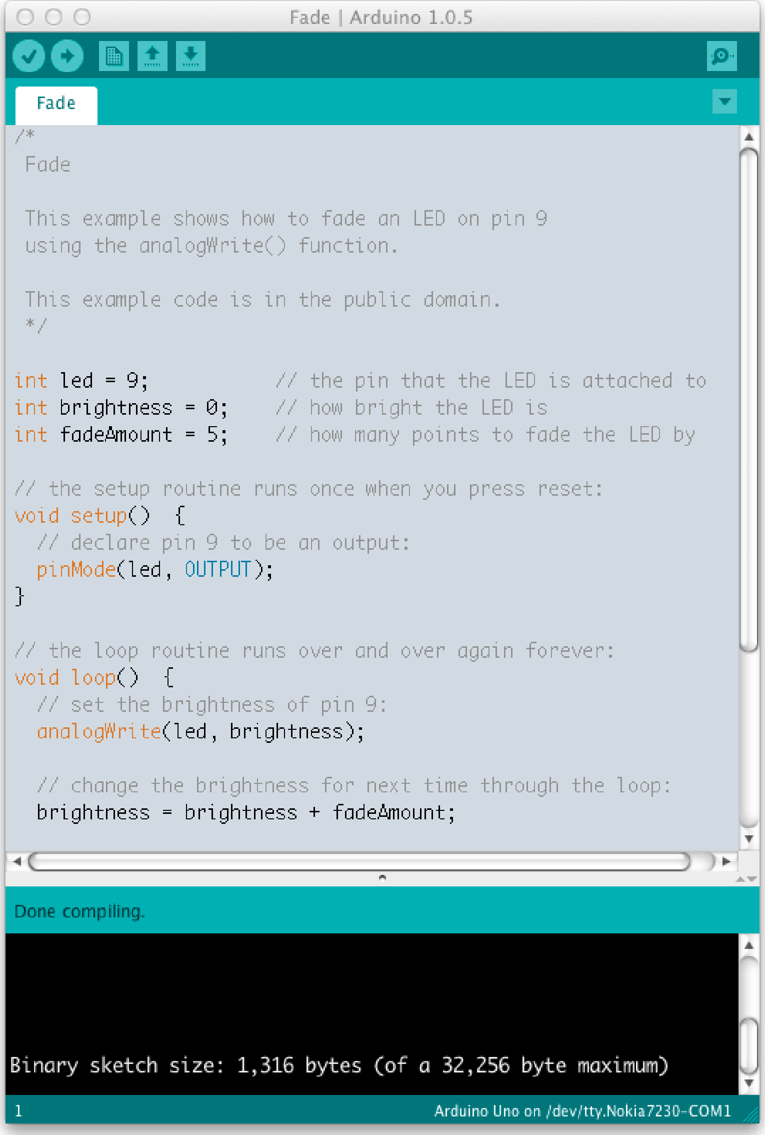
Brings computer back to real world

The Arduino board



The Arduino IDE

- Small and simple IDE
- C/C++/Arduino
- Based on Processing and GCC
- Creates sketches
- Compiles and Uploads to the Arduino board
- Provides the programmer with a serial monitor



```
/*
 Fade

 This example shows how to fade an LED on pin 9
 using the analogWrite() function.

 This example code is in the public domain.
 */

int led = 9;           // the pin that the LED is attached to
int brightness = 0;    // how bright the LED is
int fadeAmount = 5;    // how many points to fade the LED by

// the setup routine runs once when you press reset:
void setup()  {
  // declare pin 9 to be an output:
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop()  {
  // set the brightness of pin 9:
  analogWrite(led, brightness);

  // change the brightness for next time through the loop:
  brightness = brightness + fadeAmount;
```

Done compiling.

```
Binary sketch size: 1,316 bytes (of a 32,256 byte maximum)
```

Datatypes

- char, byte
- int, unsigned int
- long, unsigned long
- float
- bool

- uint8_t
- int16_t, uint16_t
- int32_t, uint32_t
- float
- boolean

Programming language is C++ with an Arduino touch

Can be programmed using assembly language

Creating a sketch

- The C++ `main()` function already exists « inside » the IDE, no need to write it
- Two functions must be present:
  - `void setup()` to initialise things (executed once)
  - `void loop()` to actually do things (executed forever)
- You can write as many functions as needed …

  … as long as the code fits into the 32 kB of memory!

Reading values from ports

- 13 digital ports (read 0V or 5V)
  - `pinMode(pinNb, INPUT) ;`
  - `int val = digitalRead(pinNb) ;`

- 6 analog inputs (ADC 10 bits)
  - `unsigned int val = analogRead(An) ;`
  - May be used as digital INPUT if needed

Analog ports may be used as digital INPUTS if needed

Writing values to ports

- 13 digital ports (write 0V or 5V)
  - `pinMode(pinNb, OUTPUT) ;`
  - `digitalWrite(pinNb, HIGH|LOW) ;`

- Some digital ports are PWM enabled
  - `analogWrite(pinNb, val) ; // val >=0 and val <=255`

Analog ports may be used as digital OUTPUTS if needed

Example

```
uint8_t d ;
float a ;

void setup() {
  Serial.begin(9600) ;
  pinMode(9, INPUT) ;
  pinMode(7, OUTPUT) ;
  digitalWrite(7, LOW) ;
  Serial.println(“Hello!”) ;
}

void loop() {
  d = digitalRead(9) ;
  a = analogRead(A0) ;
  if ( a < 300 && d == HIGH ) {
    digitalWrite(7, LOW) ;
    Serial.println(“OFF”) ;
  } else {
    digitalWrite(7, HIGH) ;
    Serial.println(“ON”) ;
  }
  delay(500) ;
}
```

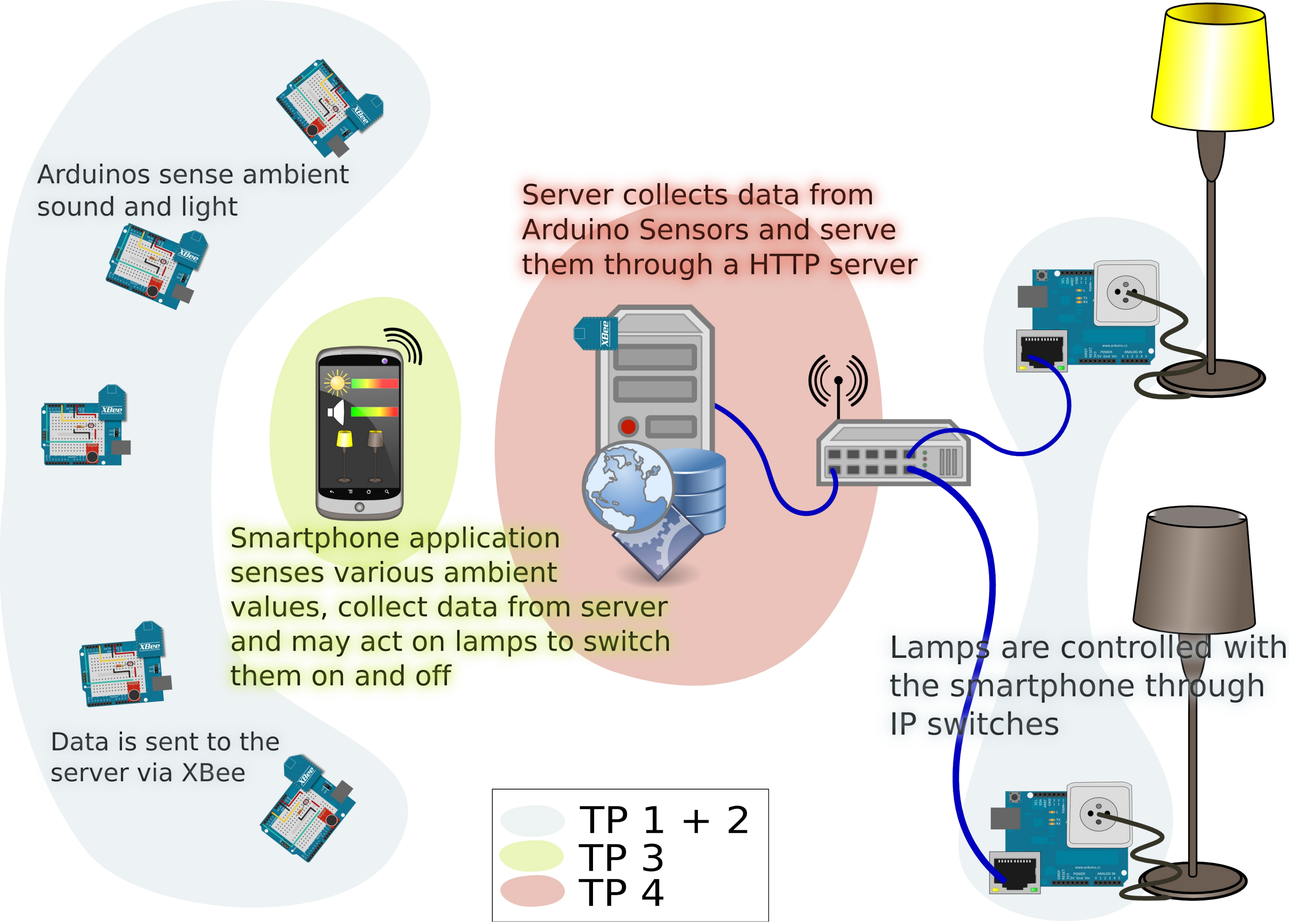
- Pointless example
- Use `pinMode()` to set the ports to the right direction
- Use a serial communication to print data
- Switch something on and off depending on values read

Other Arduinos and Shields

- Arduino
  - Duemilanove, Uno, Mega, Nano, Mini, ...
  - Micro-controller changes depending on Arduino model (ATMega 168, ATMega 328, ATMega 2560, ...)

- Shields
  - Daughter boards you can plug on top of an Arduino module
  - Extend capabilities (Ethernet, Wi-Fi, motors, GPS, displays, ...)
  - External libraries (often needed)

Use Case



Arduinos sense ambient sound and light

Data is sent to the server via XBee

Smartphone application senses various ambient values, collect data from server and may act on lamps to switch them on and off

Server collects data from Arduino Sensors and serve them through a HTTP server

Lamps are controlled with the smartphone through IP switches

⬜ TP 1 + 2
🟩 TP 3
🟥 TP 4

References

- http://www.arduino.cc/
- http://www.processing.org/
- http://www.fritzing.org/
- Arduino Programming Notebook
- Comprendre les Microcontrôleurs (french MOOC on http://class.coursera.org/)