Make Things Talk With Arduino

Didier Donsez

Université Joseph Fourier

PolyTech'Grenoble LIG/ADELE

Didier.Donsez@imag.fr

Didier.Donsez@ieee.org

Motivations

- Physical computing
  - Create interactive objects or environments.
    - for artists, designers, hobbyists
- Fast prototyping
  - Mockup, proof of concept, DIY
- Education
  - Teach/learn electronic and programming for dummies
    - teachers, high school

- Free and open-source software and hardware
  - Board designs, Tools chain, Programs …
    - but silicium and PCB can not be downloaded like FOSS

Community (TBD)

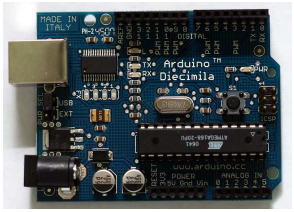
- Blog, Playground, Forum
- How-to
- Multi-langages
- Popularity
- Contributes !
  - Register and Q&A, translate, explain … !

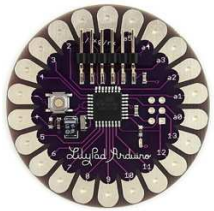
Hardware

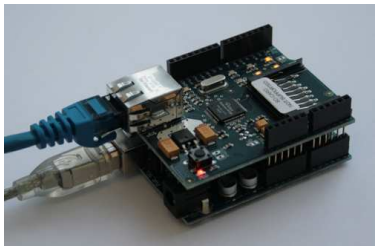
- Board
  - Microcontroller Atmel AVR (8 bits, 8-16Mhz)
    - ATmega8, ATmega168, ATmega328, and ATmega1280.
    - 16 to 128 KB flash
    - 1KB to 8KB SRAM,
    - 0.5 to 4 KB EEPROM)
  - Analogic Inputs
    - 6 pour 328, ?? for Mega2560
  - Digital IO and PWM
    - 14 (6 are PWM) for 328, ?? for Mega2560
  - Extension Catalog
    - Communications: RS-232, USB, Ethernet, BT, XBee
    - Sensors
    - Actuators (servo moteurs, …)
  - Several form factors : nano, …
  - and Shields
- Low cost
  - Start from 20€ (even 5€ if recycling)
  - Several manufacturers and dealers
  - Licence LGPL
  - But « **Arduino** » is only allow for official products

Form factors

- Boards
  - Official
    - USB
    - Mega
    - Pro
    - Mini & Nano
    - LilyPad (wearable)
  - Clones
    - Freeduino, Seeeduino Stalker,
    - Funnel IO, BlackWidow …
  - Erzats (Cortex M3, …)
    - FEZ Panda, Netduino, Leaf Maple
- *Piggy-backed* Shields
  - Ethernet
  - Bluetooth
  - XBee
  - Pilot
  - solderless breadboards …







Programming Languages

- Wiring
  - based on Processing *http://processing.org/*
  - C/C++ like syntax
- C/C++
  - GNU chain
- AMForth (ATMega Forth)
  - http://amforth.sourceforge.net
- AVR Assembly
  - WinAVR, AVRDude
- Graphical box tool
  - for artists

Syntax (i)

- Program structure
  - `void setup() { … }  void loop() { … }`
- Statements
  - `; {}`
  - `// /* */`
  - `#define #include`
- Control structures
  - `if, if...else, for, switch case, while, do... while`
  - `break, continue, return, goto`
- Control structures
  - `type func(type param, ...)`

Syntax (ii)

- Data Types
  - `void, boolean, char, unsigned char, byte,`
  - `int, unsigned int, word, long, unsigned long`
  - `float, double,` string (`char[]`)
- Contructor
  - []
- Variables
  - *local*, *global*, `static` *local*, `volatile`, `const`
  - `sizeof()`
- Operators
  - Arithmetic = `+ - * / %`
  - Comparison == `!= < > <= >=`
  - Boolean && `||` `!`
  - Bitwise & `|` `^` `~` `<< >>`
  - Pointer Access * &
  - Compound == `--` `+= -= *= /=&= |=`

Interuptions

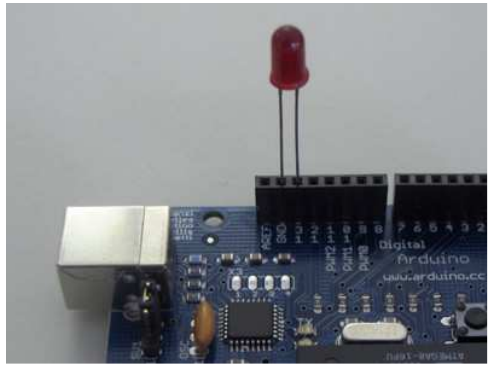
- Motivation : avoid polling (with complex timing calibration)
- External interuptions
  - Digital pin 2 and 3 on Arduino
  - + digital pin 21, 20,19,18 on Mega
  - `attachInterrupt`(*interrupt, funct, mode*), `detachInterrupt`(*funct*)
  - *mode* = *`LOW, CHANGE, RISING, FALLING`*
- Critical section
  - `noInterrupts(); ... interrupts();`
- Example

```
#define LED 13;
volatile int state = LOW;
void setup() {
  pinMode(LED, OUTPUT);
  attachInterrupt(0, blink, CHANGE);
}
void loop() {  digitalWrite(LED, state); }
void blink() {  state = !state; }
```

Code snippet
Blink

```
// blink.pde

#define LED_PIN 13
```

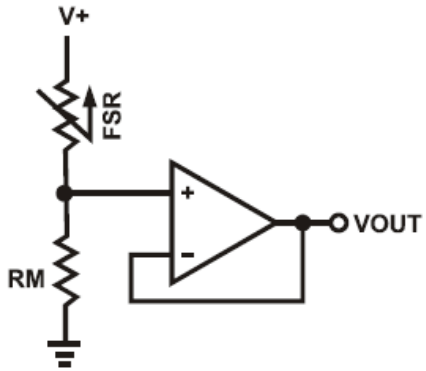

```
// run once at the start of a program which can be used for initializing settings
void setup () {
    pinMode (LED_PIN, OUTPUT);     // enable pin 13 for digital output
}

// called repeatedly until the board is powered off
void loop () {
    digitalWrite (LED_PIN, HIGH);  // turn on the LED
    delay (1000);                  // wait one second (1000 milliseconds)
    digitalWrite (LED_PIN, LOW);   // turn off the LED
    delay (1000);                  // wait one second
}
```

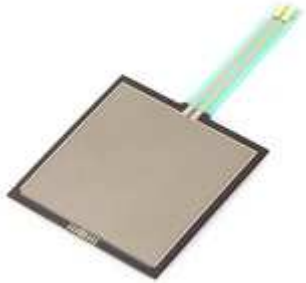


Code snippet for input Weight

- Application : wiifit, i-shoe, i-sofa …

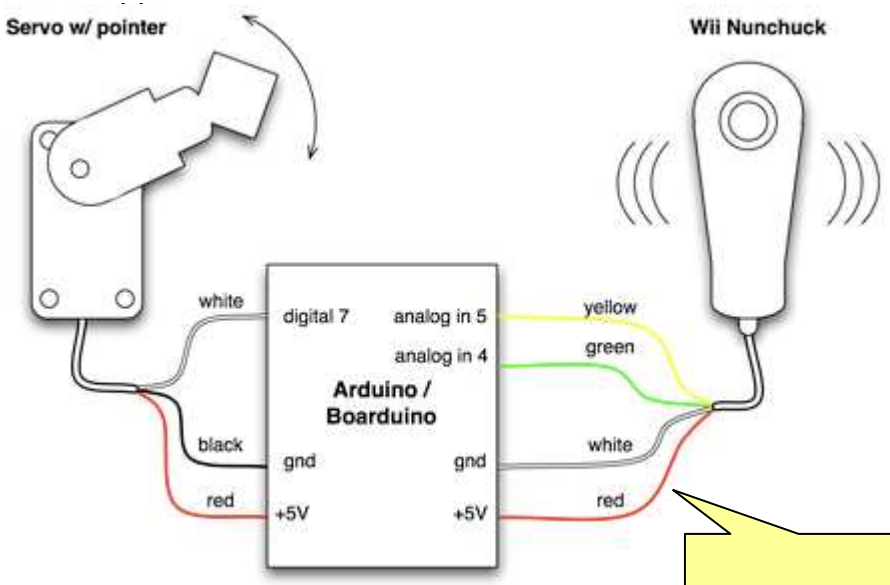
```
void setup() { Serial.begin(9600); }
void loop() {
  // read the analog input into a variable:
  int analogValue = analogRead(0);
  Serial.println(analogValue);
  delay(10);
}
```





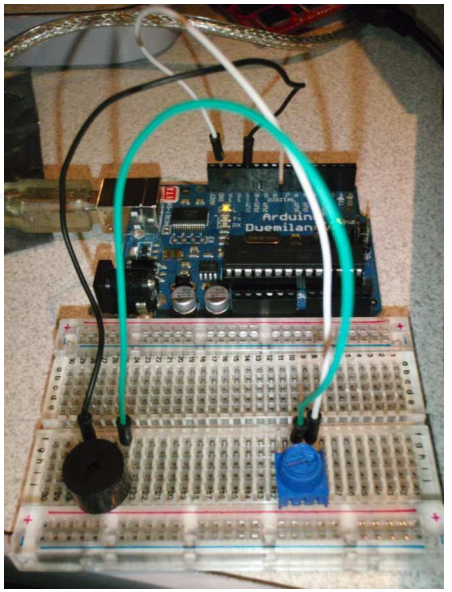
Code snippet for I2C Input and Control Nunchuck → Servo motor

```
#include <Wire.h>
#include "nunchuck_funcs.h"
#include <Servo.h>

Servo myservo;
byte joyx,prevjoyx, zbut,cbut;
prevjoyx=0; // global

void setup() {
   myservo.attach(9);  // attaches the servo on pin 9 to th
   nunchuck_setpowerpins();
   nunchuck_init(); // send the initilization handshake
}

void loop() {
       nunchuck_get_data();
       joyx  = nunchuck_joyx();
       zbut = nunchuck_zbutton();
       cbut = nunchuck_cbutton();

       if(zbut==1) {
         myservo.write(0); delay(500);
       } else if(zbut==1) {
         myservo.write(180); delay(500);
       } else if((prevjoyx+5<joy) || (joy<prevjoyx-5)) {
         prevjoyx=joyx;
         myservo.write(joyx+90); delay(100);
       } else { delay(100); }
}
```



I2C

Code snippet for PWM
Music maestro !



```
// from Tom Igoe
 #include "pitches.h"
#define PIN 8
int melody[] = { NOTE_C4 /* 262 Hz */, NOTE_G3,
  NOTE_G3, NOTE_A3, NOTE_G3,0, NOTE_B3, NOTE_C4};
int noteDurations[] = { 4,8,8,4,4,4,4,4 };

void setup() { }

void playMelody() {
  for (int thisNote = 0; thisNote < 8; thisNote++) {
    int noteDuration = 1000/noteDurations[thisNote];
    tone(PIN, melody[thisNote],noteDuration);
    int pauseBetweenNotes = noteDuration * 1.30;
    delay(pauseBetweenNotes);
  }
}

void playMelody() {
    play(); delay(2000);
}
```

Communications

- Talking to the Cloud, Internet of Things

- Serial: RS232, I2C, 1-Wire
- Ethernet
- WiFi
- Bluetooth
- ZigBee (XBee) 900 MHz, 2.4GHz
- RF 433MHz
- IrDA

Host

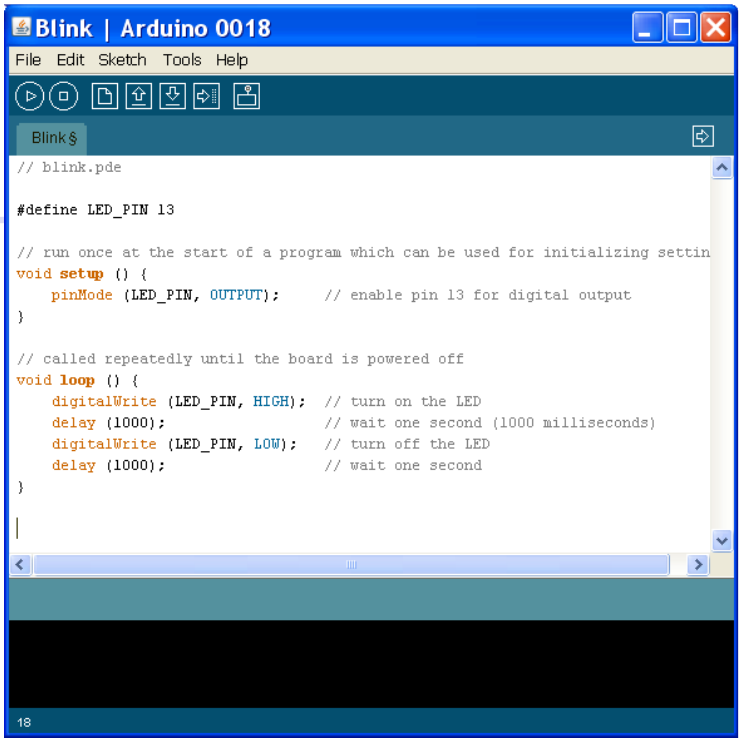
- RXTX.org
- processing.serial

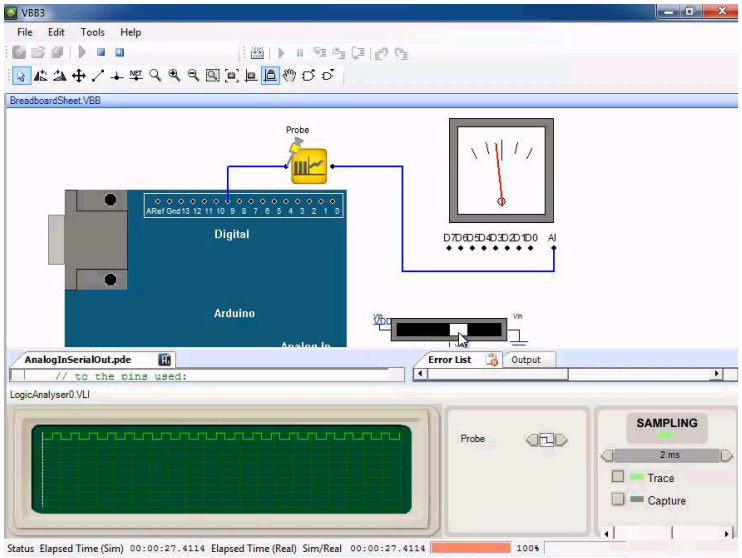
```
import processing.serial.*;
import cc.arduino.*;
Arduino arduino;
int ledPin = 13;
void setup() {
  //println(Arduino.list());
  arduino = new Arduino(this, Arduino.list()[0], 57600);
  arduino.pinMode(ledPin, Arduino.OUTPUT);
}
void draw() {
  arduino.digitalWrite(ledPin, Arduino.HIGH);
  delay(1000);
  arduino.digitalWrite(ledPin, Arduino.LOW);
  delay(1000);
}
```

- Eclipse Terminal plugin
- Putty

Tools

- Arduino IDE
  - http://arduino.cc/en/Main/Software
  - basic but quick startup
  - Portable (Java app)
  - based on Processing
- Eclipse IDE CDE (C/C++)
  - + AVR plugin + RXTX plugin
    - http://www.arduino.cc/playground/Code/Eclipse
  - More complex
- Emulators
  - Virtual Breadboard
- Remote management ??
- Unit testing
- Command lines
  - http://www.arduino.cc/playground/Code/WindowsCommandLine
- Builder (cmake, ant, maven)
  - http://www.arduino.cc/playground/Main/DevelopmentTools





Arduino + Eclipse

```
#include <HardwareSerial.h>
#include <WProgram.h>
#include <wiring.h>
#include <WConstants.h>
#include <binary.h>
#include <pins_arduino.h>
#include <wiring_private.h>

int main(void) {

  /* Must call init for arduino to work properly */
  init();

  /*****************************/
  /* Add your setup code here */
  /*****************************/

  for (;;) {

    /*****************************/
    /*** write main loop here ***/
    /*****************************/

  } // end for

// you-MUST-NEVER-return-from-main

} // end main
```

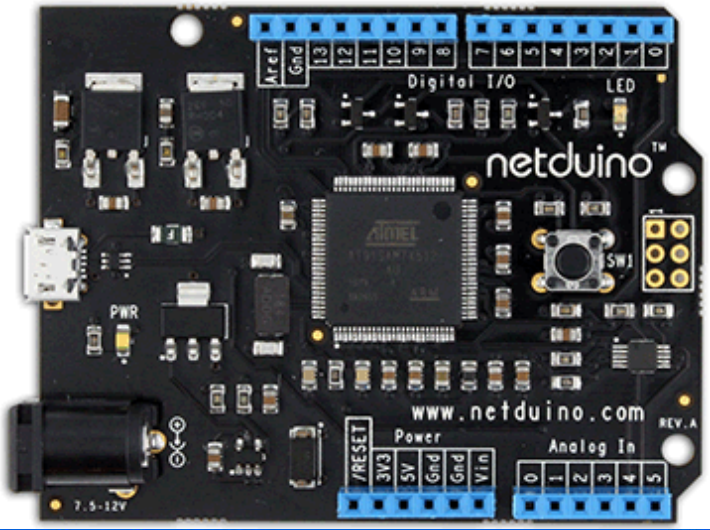
- Wiring
  - http://wiring.org.co
  - http://www.processing.org

Erzats

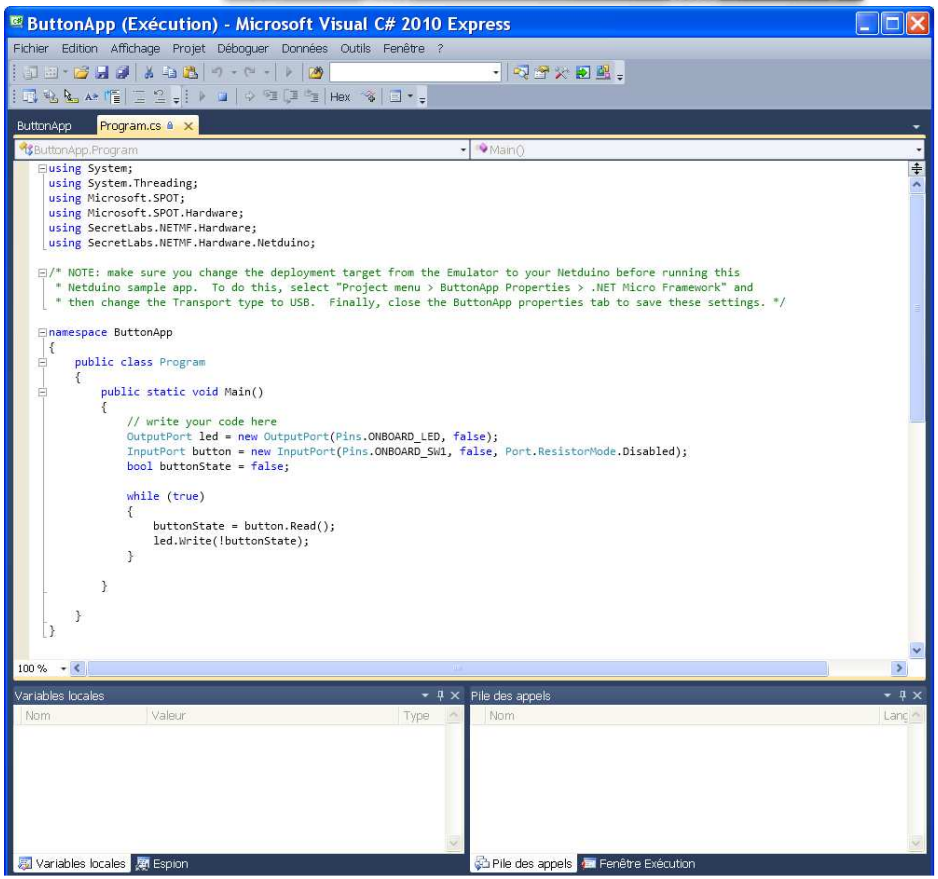
- Arduino pin compatible → Shield compatibility
- But powerful processors
  - Netduino (http://netduino.com)
  - Leaf Maple http://leaflabs.com/docs/maple/
  - FEZ (http://www.tinyclr.com)

Netduino

http://netduino.com



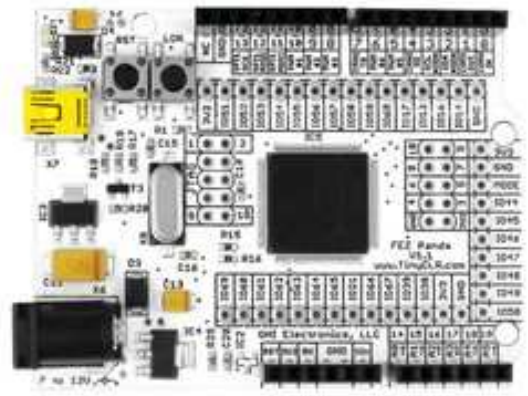
- Atmel ARM7 48MHz
  - Code Storage: 128 KB
  - RAM: 60 KB
  - 20 GPIOs with SPI, I2C
  - 2 UARTs (1 RTS/CTS)
  - 4 PWM and 6 ADC channels
  - Compliant with Arduino shields
- Netduino Plus
  - + Micro SD + Ethernet
- Development
  - .NET Micro Framework C#
  - Visual Studio Express
  - SharpDevelop ???



FEZ

http://www.tinyclr.com

- FEZ Panda
  - 72Mhz NXP ARM processor, 62KB RAM, 148KB FlashRAM
  - 60 PIN (6PWM, 4TTL UART, SPI, I2C, CAN, OneWire)
  - JTAG exposed, USB Client
- FEZ Domino
  - 72Mhz NXP ARM processor, 62KB RAM, 148KB FlashRAM
  - 30 PIN (6PWM, 3TTL UART, SPI, I2C, CAN, OneWire)
  - USB Host & Client, RTC, micro SD connector
- Dev tools
  - C# .NET MF
  - VisualStudio



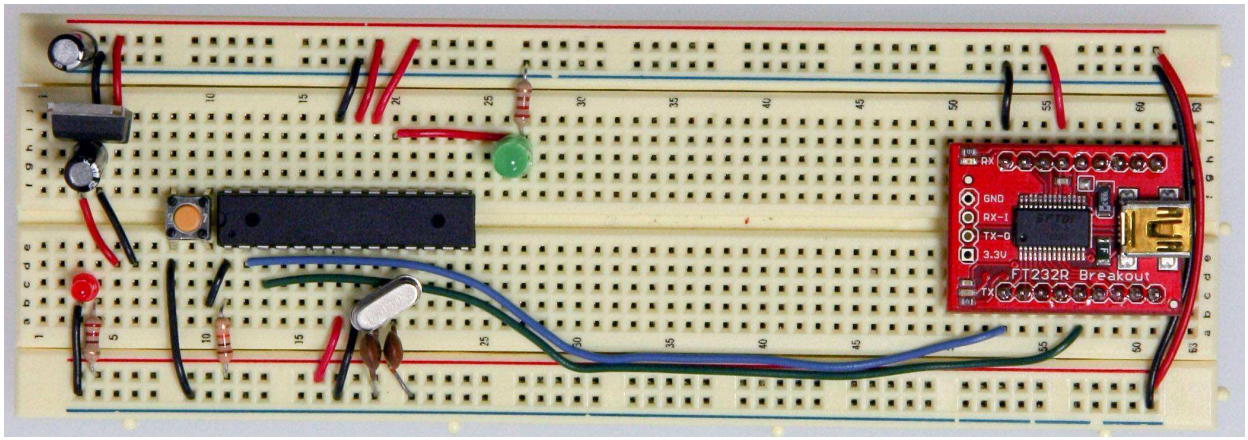
WaspMote

http://www.libelium.com/products/waspmote/hardware

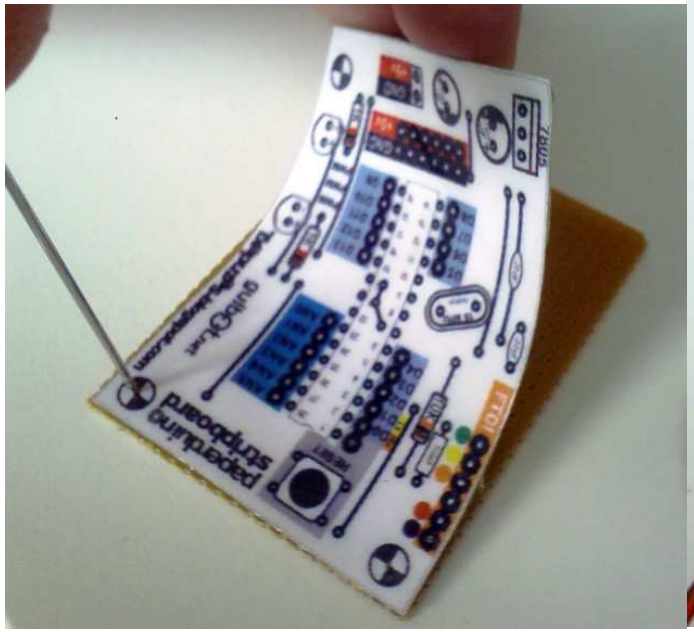
- Hardware
  - ATmega1281 8MHz
  - 8KB SRAM 4KB EEPROM 128KB FLASH
  - SD Card, XBee Socket
  - RTC (32KHz)
  - 7 Analog, 8 Digital (I / O), 1 PWM, 2 UARTs, 1 I2C, 1USB
  - On board sensors: Temperature, Accelerometer: ±2g / ±6g
  - Battery and Solar panel slots. 3V CR2032 auxilary battery
  - **Not compatible with Arduino shields**
- Software
  - **Wiring**
  - FOSS API

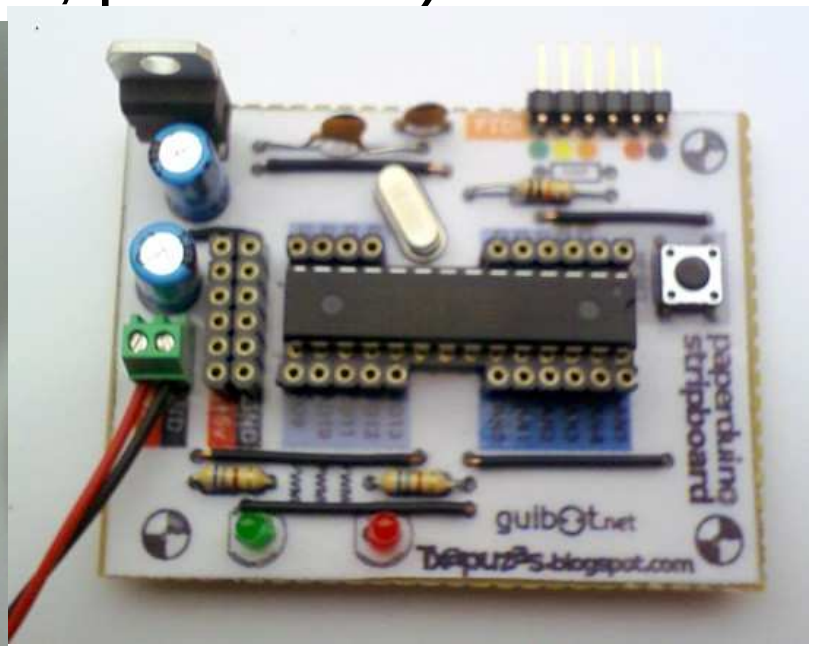
Homebrew Arduino

- BreadDuino



- Paperduino (stripboard, perfboard)





Books

- Banzi, Massimo (March 24, 2009). Getting Started with Arduino (1st ed.). Make Books. pp. 128. ISBN 0596155514. http://www.makershed.com/ProductDetails.asp?ProductCode=9780596155513.
- Tom Igoe, Making Things Talk: Practical Methods for Connecting Physical Objects, Make Books , 2007, ISBN-10: 0596510519
- Oxer, Jonathan; Blemings, Hugh (December 28, 2009). Practical Arduino: Cool Projects for Open Source Hardware (1st ed.). Apress. pp. 500. ISBN 1430224770. http://www.apress.com/book/view/9781430224778.
- Noble, Joshua (July 15, 2009). Programming Interactivity: A Designer's Guide to Processing, Arduino, and openFramework (1st ed.). O'Reilly Media. pp. 768. ISBN 0596154143. http://oreilly.com/catalog/9780596800581/.
- Schmidt, Maik (November 20, 2010). Arduino: A Quick-Start Guide (1st ed.). The Pragmatic Bookshelf. pp. 275. ISBN 978-1-93435-666-1. http://pragprog.com/titles/msard/arduino.
- Sparkfun Inventor'Guide
  - http://www.sparkfun.com/tutorial/AIK/CIRC00-sheet-SPAR.pdf
  - http://www.sparkfun.com/tutorial/AIK/ARDX-EG-SPAR-PRINT-85.pdf

Shops

- Lectronix.fr
- Farnell.fr
- Radiospare.fr

- Sparkfun.com
- Seeedstudio.com
- CoolComponent.co.uk

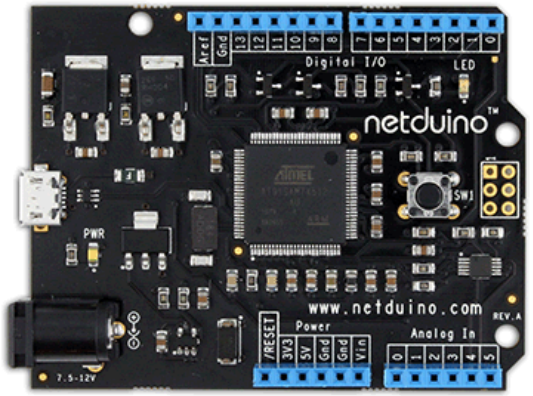
- eBay

And Now !
The Exercices

- Choose one in the Sparkfun Inventor'Guide
  - http://www.sparkfun.com/tutorial/AIK/CIRC00-sheet-SPAR.pdf
  - http://www.sparkfun.com/tutorial/AIK/ARDX-EG-SPAR-PRINT-85.pdf

Bonus track

Personal Collection



Video Game Shield

Personal DIY projects

- Wattmeter
- Plant
- Arcade joystick

Polytech'Grenoble projects