Physical Interaction Design - Vision for a visual programming and simulation environment

Sebastian Gregor

INF-M3 - Ringvorlesung (WS 08/09)
Department Informatik
HAW Hamburg
15. Dezember 2008

Outline

- Motivation
  - Interaction Design Projects
  - Arduino
  - LilyPad
  - Fritzing
- Master Thesis
  - Vision
  - visual programming
  - Simulation
- Risks
- Perspective

Motivation

- Pentiment Summer Course 2008
  - Wearable Computing / E-Textiles (Eyal Sheffer)

- Cooperation Design Department
  - Master Project „Emotional Tent“

Motivation

Facts:

- electronic and (wearable) computer are widely available (and cheap)
- new (physical) interaction techniques
- fashion designer put the esthetic point to electronic and computing
- different kind of inputs / sensors
- physical computing:
  - human body as input source
  - use sensors
  - MCU process input
  - MCU controls actors (electro-mechanical devices): motors, servos, lightning ….

Motivation

Challenges:

- need to assemble different kind of electronics
- sewing is a problem
- a minority can program micro controller
- disappointing difficulties with installation
- can not take components apart and modify them
- debugging is difficult

Motivation

- Pentiment:
  - Gruppen mit 3 bis 4 Studenten
  - Erarbeitung eines Konzeptes
  - Auswahl von Hardware (was gerade da war)
  - Versuch der Umsetzung des Konzeptes in zwei Wochen
- Probleme:
  - Interaktive Komponenten erfordern den Einsatz von Mikrokontrollern
  - Schwierigkeiten der Abstraktion von Technik bei Designer
  - Nicht genügend „Techniker“ vor Ort
  - Einbau aller Komponenten auf einmal ➔ Problem wo liegt der Fehler
  - Keine Zeit für Debuggen vorhanden
  - Keine Möglichkeit Elektrische Komponenten oder elektrische Schaltkreise zu verändern ➔ alles fest vernäht
  - Schwierigkeiten beim Debuggen (nicht eingeplant)

Motivation

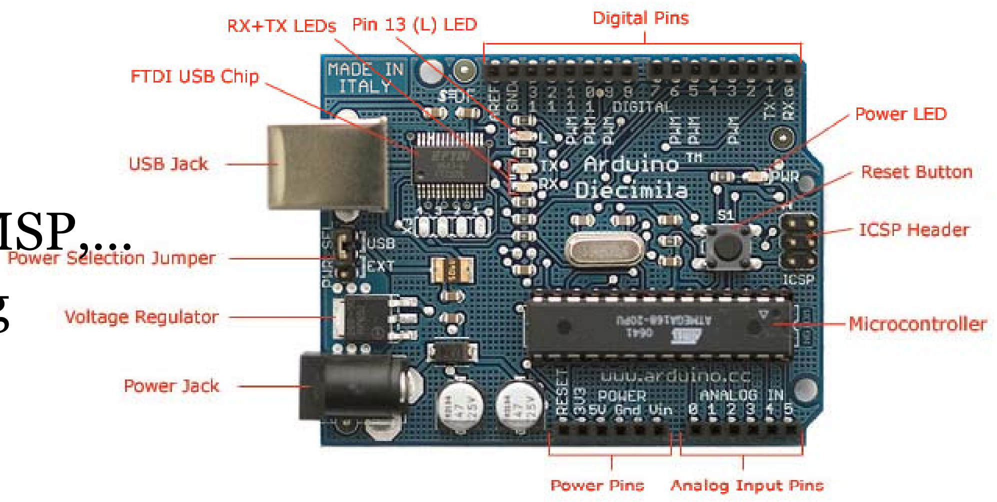
- physical computing:
  - human body as input source
  - use sensors
  - MCU process input
  - MCU controls actors (electro-mechanical devices):
    - motors
    - servos
    - lightning
    - ….

Motivation

- Platforms for physical Computing
  - Handyboard (http://handyboard.com)
  - LogoChip (http://www.wellesley.edu/Physics/Rberg/logochip/distribution)
  - Phidgets (http://grouplab.cpsc.ucalgary.ca/phidget)
  - d.tools (http://hci.stanford.edu/dtools/)
  - Gainer (http://gainer.cc)
  - MakingThings (http://www.makingthings.com)
  - Wiring (http://wiring.org.co/)
  - Arduino (http://www.arduino.cc)

Arduino

- Hernando Barragán (Interaction Design Institute Ivrea) developed Wiring in 2003
- small IO – Board based on Atmel MCU
- based on Wiring the international Arduino Projekt was launched
- IO – Board complete open-source
- can communicate with Flash, Processing, Max/MSP,…
- stand alone programming environment based on Processing



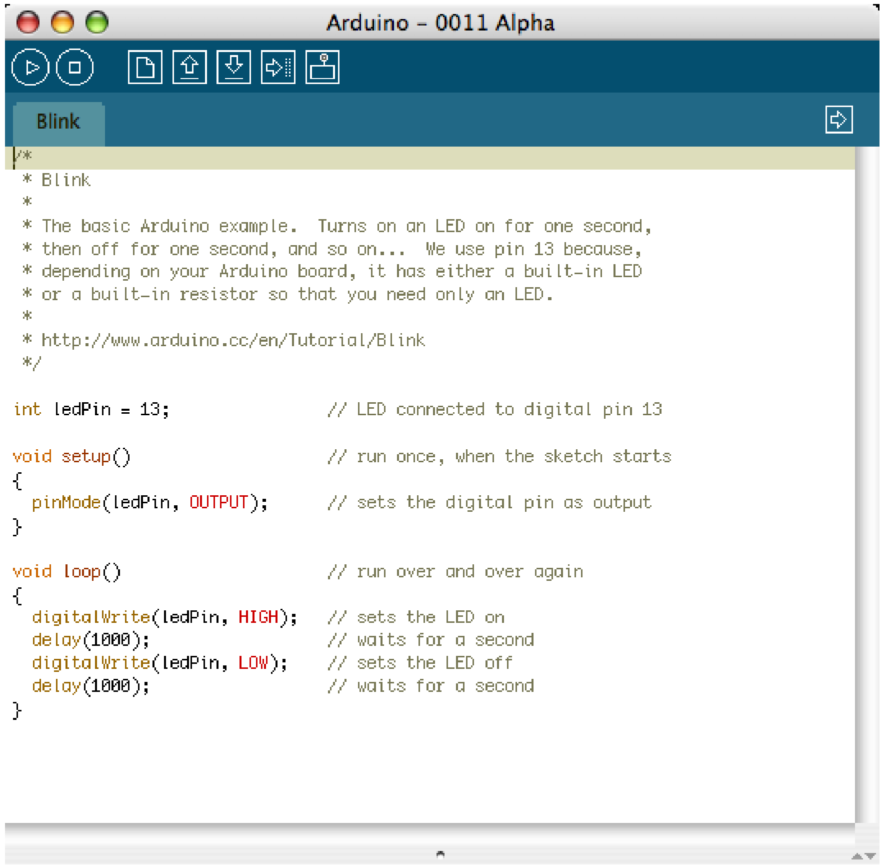
Ref 7

Photograph by SparkFun Electronics. Used under the Creative Commons Attribution Share-Alike 3.0 license.

[http://www.arduino.cc]

Arduino

- development environment runs on Windows, OS X and Linux
- integrated compiler and communication tools
- C like language (based on Wiring)
- uploading to IO – Board by clicking on the upload button
- bootloader on Atmel starts Sketch
- communication through USB – Serial converter
- environment extendable



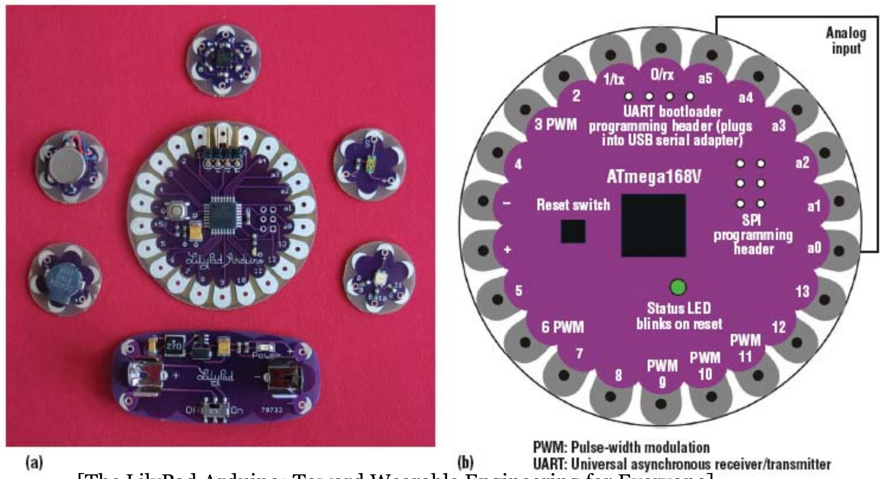
```
/*
 * Blink
 *
 * The basic Arduino example.  Turns on an LED on for one second,
 * then off for one second, and so on...  We use pin 13 because,
 * depending on your Arduino board, it has either a built-in LED
 * or a built-in resistor so that you need only an LED.
 *
 * http://www.arduino.cc/en/Tutorial/Blink
 */

int ledPin = 13;                // LED connected to digital pin 13

void setup()                    // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT);      // sets the digital pin as output
}

void loop()                     // run over and over again
{
  digitalWrite(ledPin, HIGH);   // sets the LED on
  delay(1000);                  // waits for a second
  digitalWrite(ledPin, LOW);    // sets the LED off
  delay(1000);                  // waits for a second
}
```

[http://www.arduino.cc]

LilyPad

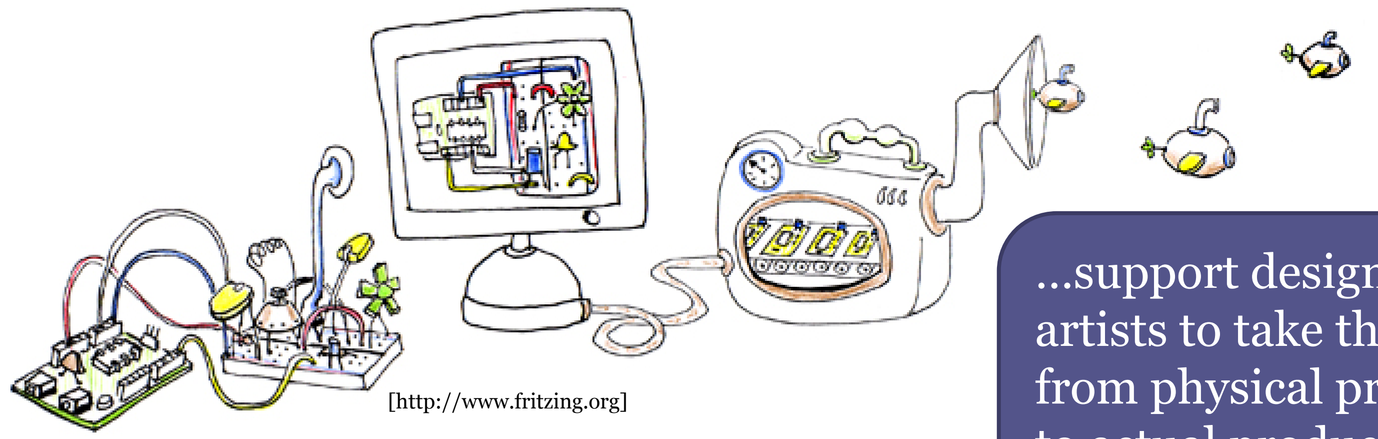
- microcontroller board designed for wearables
- developed by Leah Buechley Univerity of Colorado 2007
- can be sewn to fabric
- available as of October 2007 from Spark Fun
- fully Arduino compatible
- a lot of different sensors and actors are available

[http://www.cs.colorado.edu/~buechley/]


Analog input
1/tx
0/rx
a5
a4
2
UART bootloader programming header (plugs into USB serial adapter)
3 PWM
a3
4
a2
ATmega168V
Reset switch
a1
SPI programming header
a0
5
13
Status LED blinks on reset
6 PWM
12
7
PWM 11
8
PWM 9
PWM 10
(a)
(b)
PWM: Pulse-width modulation
UART: Universal asynchronous receiver/transmitter

[The LilyPad Arduino: Toward Wearable Engineering for Everyone]

Fritzing



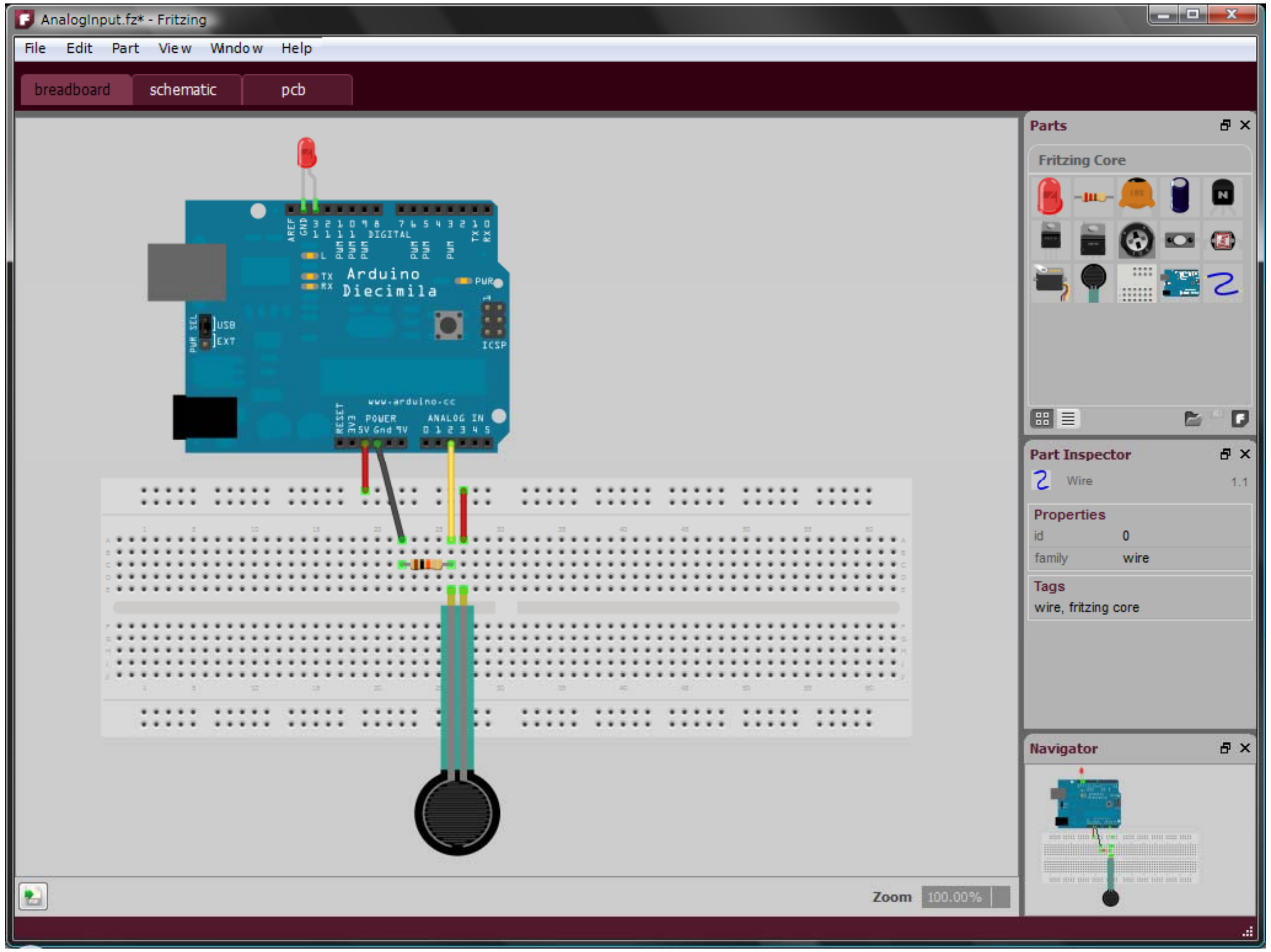
[http://www.fritzing.org]

...support designers and artists to take the step from physical prototyping to actual product.

[http://www.fritzing.org/]

- Fritzing:
  - open-source initiative
  - startet October 2007 University of Applied Science Potsdam
  - Elektronic Design Automation Software
  - goal: allows the designer to create a finished PCB of an individual circuit

Fritzing



[http://www.fritzing.org]

Motivation - Summary

- there are a lot of different platforms
- some platforms with development environment
- Arduino with huge community
  - a lot of, projects and examples
  - different tutorial and development tools

BUT:

- need some practice to program MCU
- still to complex for non-programmers
- difficult to debug
- no simulation environment

Vision

A graphical programming and simulation environment:

- allowing non – programmers to easily explore physical computing
- for Arduino / LilyPad
- with integrated graphical simulation tool
- possibility to enhance the visually generate program with ‘handwritten’ code
- stable and simple to use
- using visual programming techniques

Vision

Why visual programming?

- lowering the barriers to programming
- drag & drop commonly used
- easier to take in a lot of information's
- use symbolic of a domain
- easier to change the program
- fixed instruction set

Possible problems:

- multidimensional ➔ can be confusing
- require more space
- less documentation

Vision

- Why is simulation so important?
  - hardware and software design
  - you never know where the problem is:
    - hardware correctly assembled?
    - software fully functional?
    - both together work as expected?
    - …
  - often impossible to take components apart and change the design
  - easier for artists to imagine what the final piece looks like

Risks / Perspective

- Risks:
  - oblique approach to the topic
  - Too extensive for one master thesis?
  - visual programming inapplicable

- Perspective:
  - a lot more research necessary
  - precise the subject
  - project Svenja Keune & Martin Tischmann in summer ➔ first prototype / possible tester

Questions?