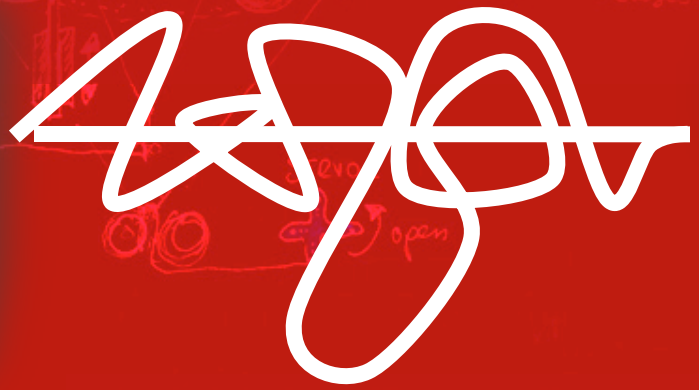
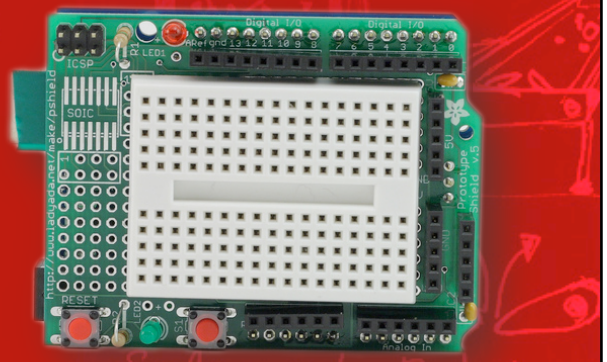
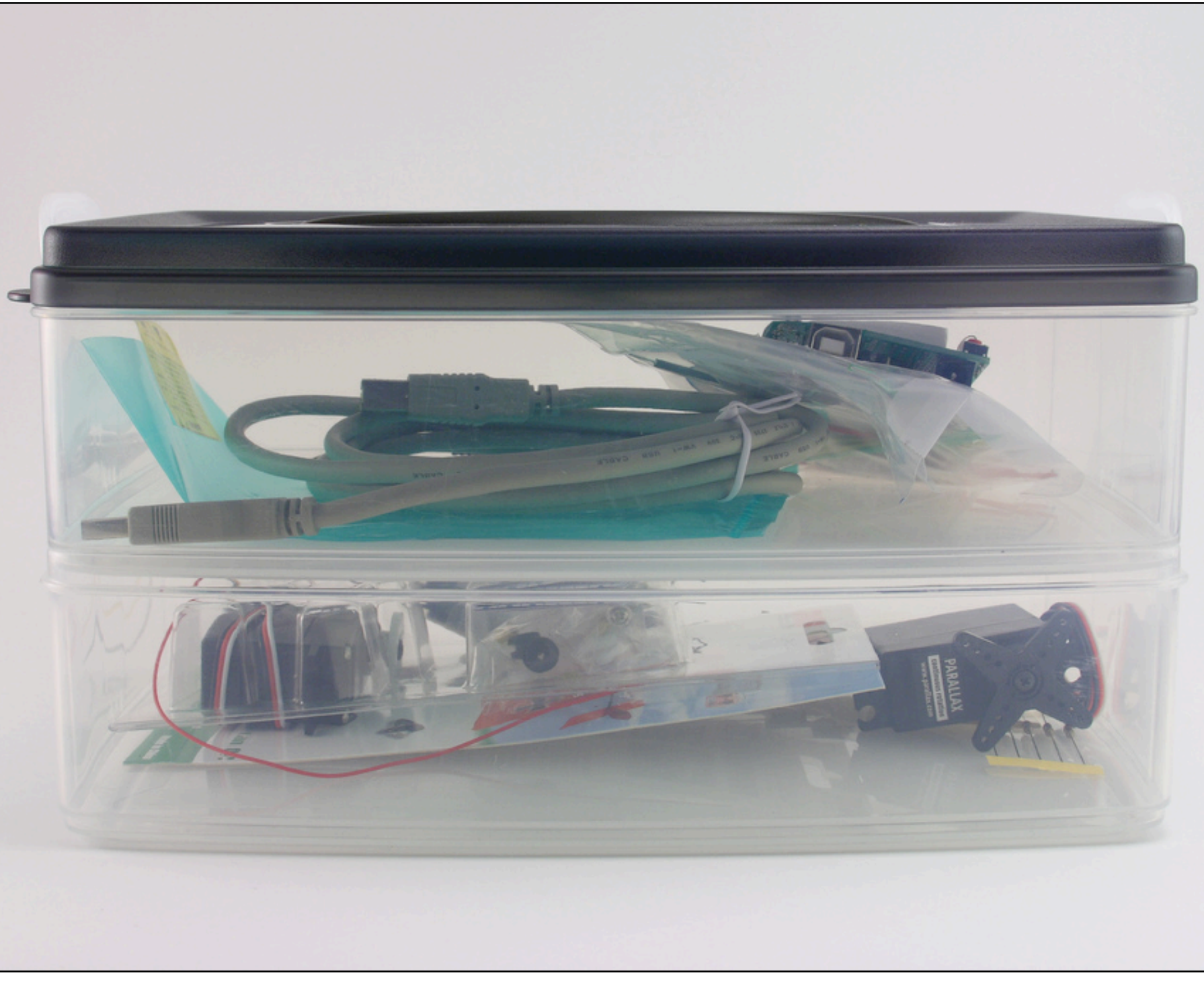
Redrawing the line between CPU and MCU

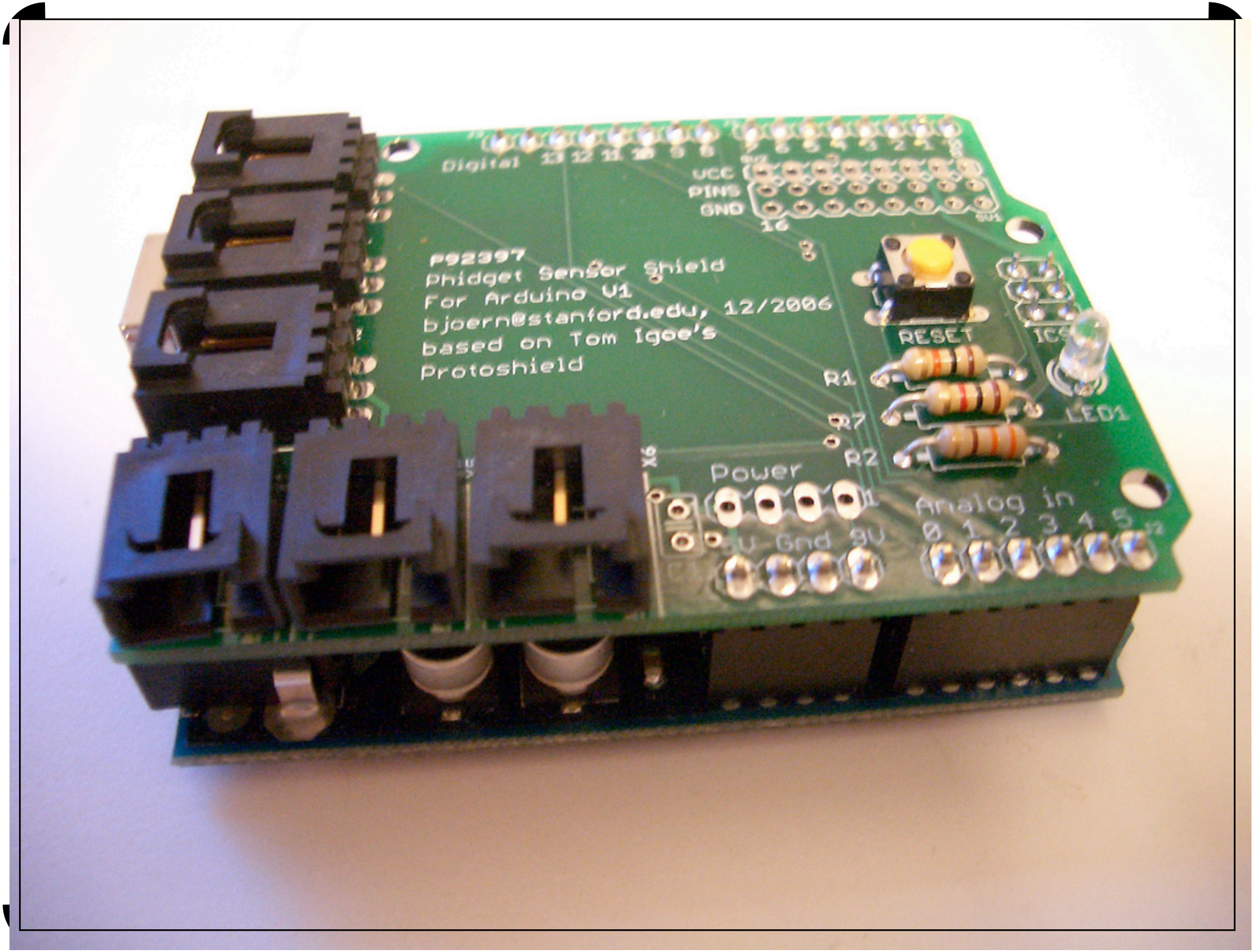




Björn Hartmann
Stanford University HCI Group


Teaching


PARALLAX


Digital
VCC
PINS
GND
P92397
Phidget Sensor Shield
For Arduino V1
bjoern@stanford.edu, 12/2006
based on Tom Igoe's
Protoshield
RESET
R1
R7
R2
LED1
Power
Gnd
Analog in
0 1 2 3 4 5

Installations


SAN JOSE MUSEUM OF ART


I WISH

Ambidextrous

- **Quarterly journal of design practice**

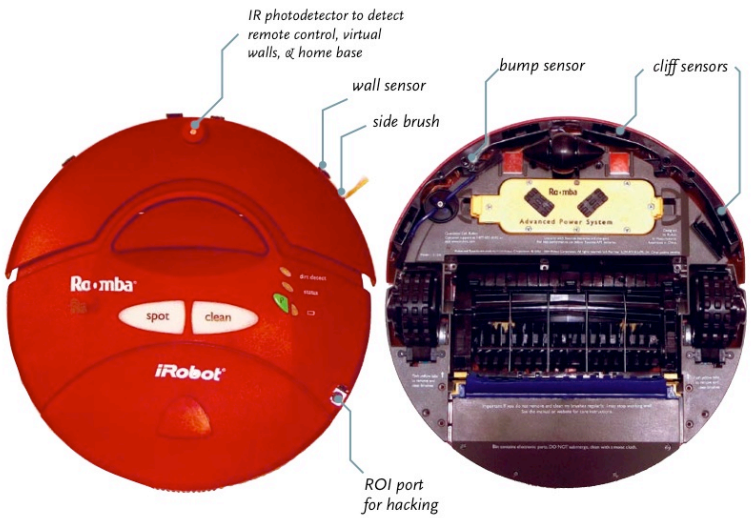
- **Student-run, professional readership**

- **Past contributors:**

  Mike K, Todd Kurt, Tom Igoe, Don Norman, Bill Moggridge, David Kelley, Genevieve Bell, Steve Portigal, Paul Dourish, Eli Blevis, Ryan Freitas, Nathan Shedroff, Bill Verplank, Larry Leifer, **...**

Flipping

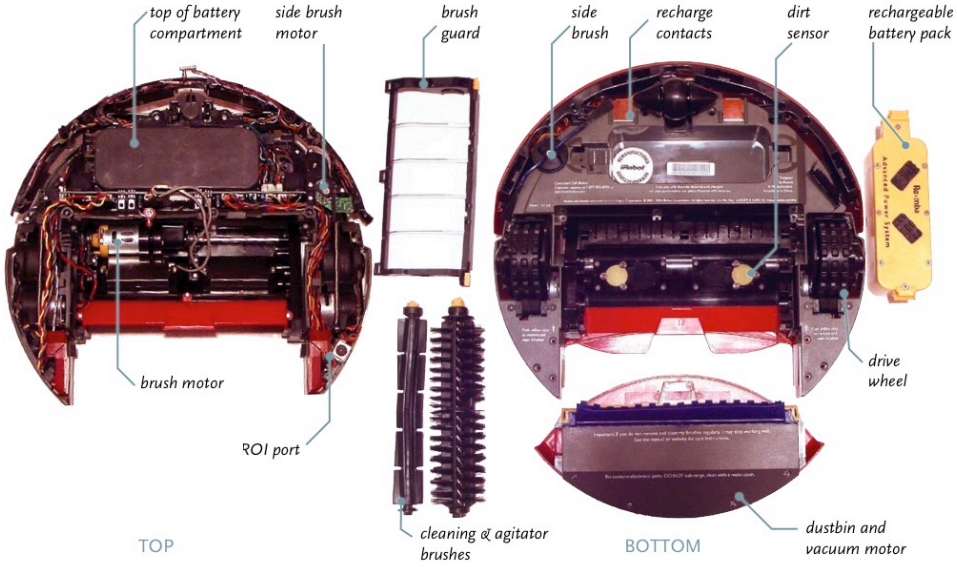
Peeking underneath at Roomba's private parts, we can see what makes the Roomba go—and how it decides to stop. The Roomba uses two drive wheels with sturdy treads, and a third caster wheel for stability, allowing it to zip around and maneuver curvy paths and tight corners. Infrared cliff sensors and bump contact sensors give Roomba some awareness. They warn Roomba if it is approaching a step or if it has bumped into something so it can avoid serious falls and collisions. (See "The Brawn" and "The Brains" for more details about the wheelsand sensors.)



TOP BOTTOM

Unclipping

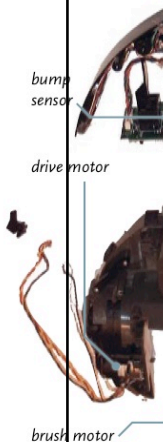
Roomba's modular designmake part replacement and cleaning easy. Its battery, brushes, dustbin and vacuum motor are all removable. Behind the dustbin, we can see Roomba's two dirt sensors above the agitator brush; these are piezo- acoustic sensors that measure the number of dirt particles being picked up, enabling the Roomba to increase the intensity of the vacuuming where it is needed, and to determine when the room is clean enough to stop.



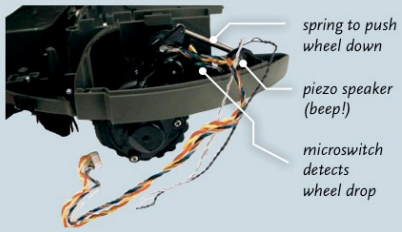
TOP BOTTOM

Stripping

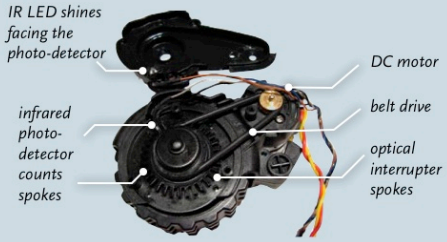
As we dig deeper, we chassis includes four side brush that helps interesting to note th mirrored on the left. clockwise circles, and



The Brawn



The Roomba's drive system is more sophisticated than the simple direct current (DC) motors you find in remote controlled cars. DC motors are part of the powertrain, but each motor unit also houses an optical encoder which measuers how fast and how far each wheel is moving. In the picture above, the red and yellow wire strands provide power to the motos, while the other wire bundle feeds the encoder output to the odometer.



The DC motor is connected to the wheel via a belt drive, which. offers a little safety in case the Roomba gets stuck: it will wear out before destroying whatever it's stuck on. The spoked wheel is used as an optical interrupter that alternately blocks and allows IR light from an LED to reach a photodetectorv. Each pulse of light between the spokes is another few millimeters of movement. Optical interrupters like this are also used in ball-type computer mice.

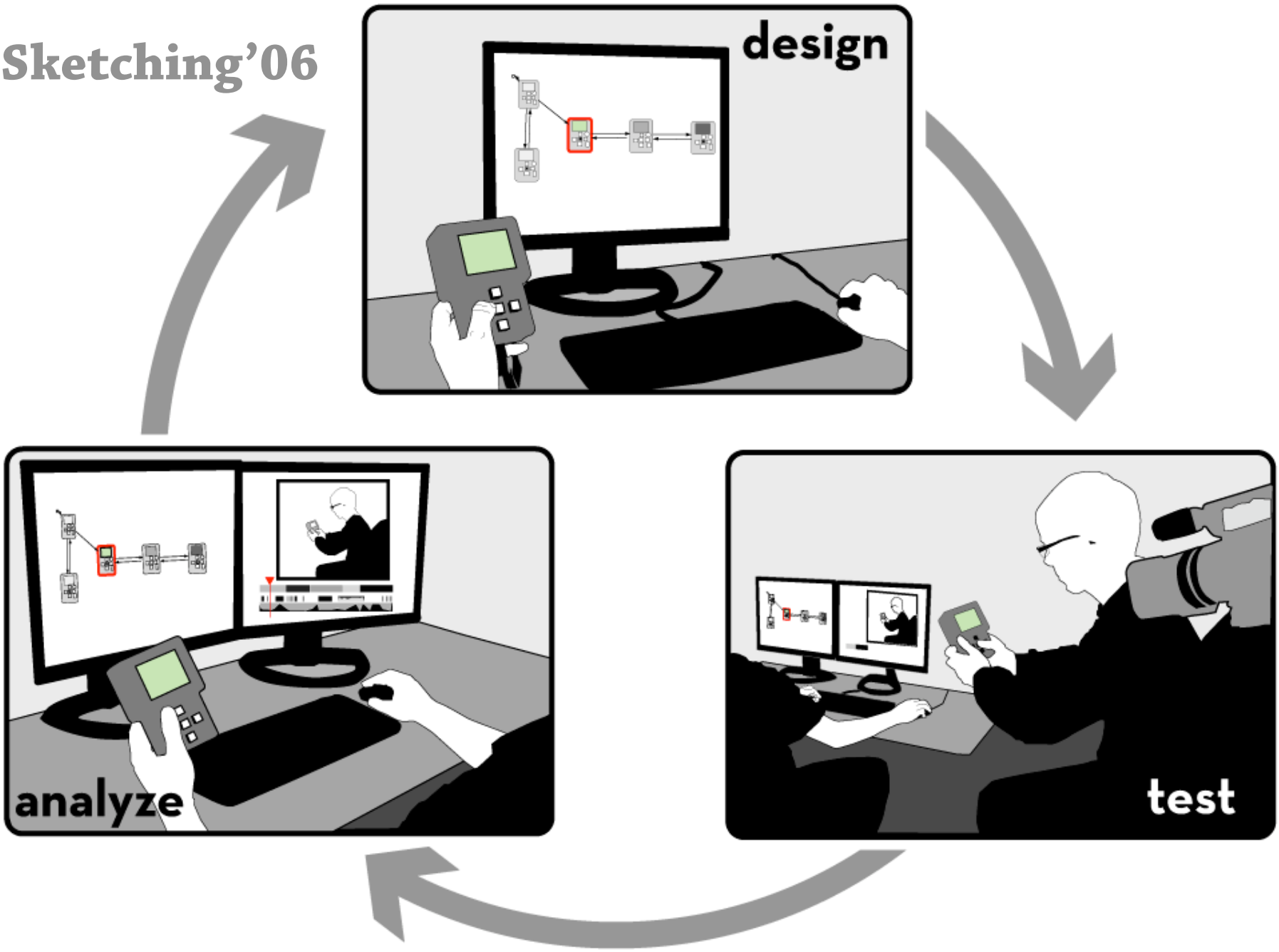
Inside each Roomba wheel is a sealed planetary gear system that converts the high speed, low torque DC motor output into high torque, low speed movement. Torque output is sufficient to carry about two gallons of house paint on a Roomba without problems.

The Brains

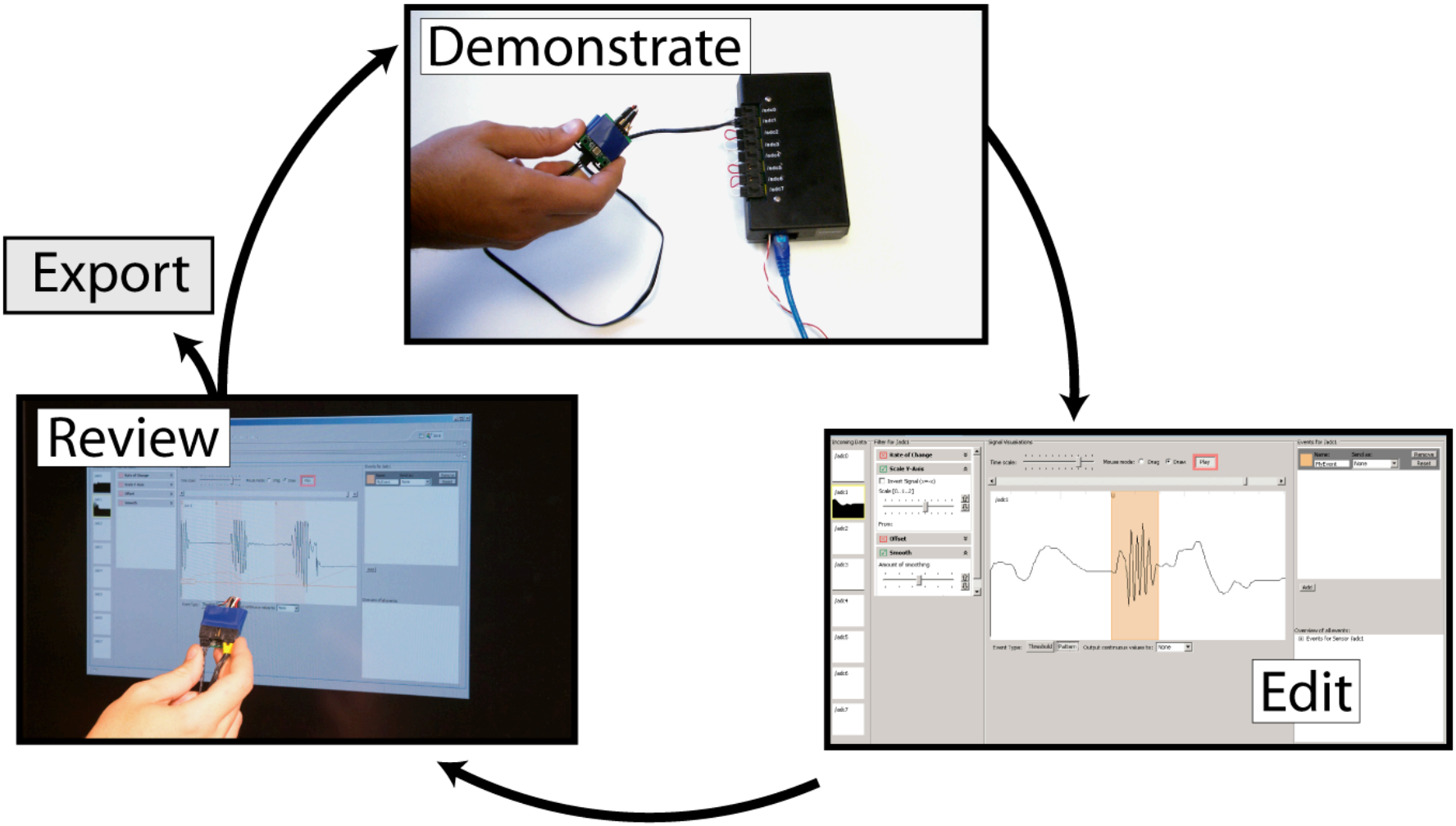
Connectors converge on the main processor's cuircuit board.

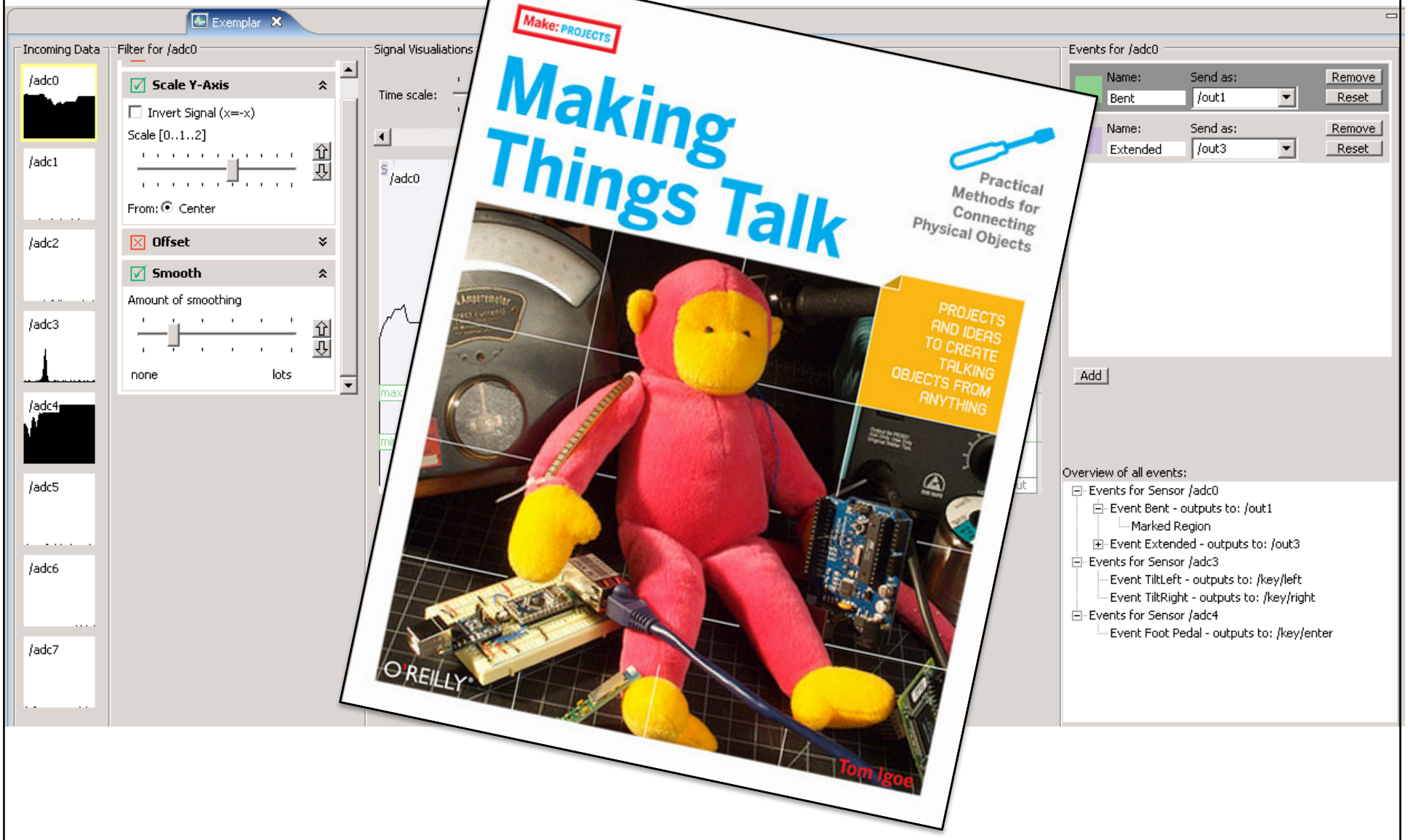
All sensor and actuator cables of the Roo circuit board. The board is easily servicea of soldering wires directly to the board, a of adhesives (except for some hot-glue to on a 6MHz, 16bit Motorola microcontoll reads sensor data 67 times per second to rithms are based on MIT professor Rodn

Sketching'06



Sketching'07



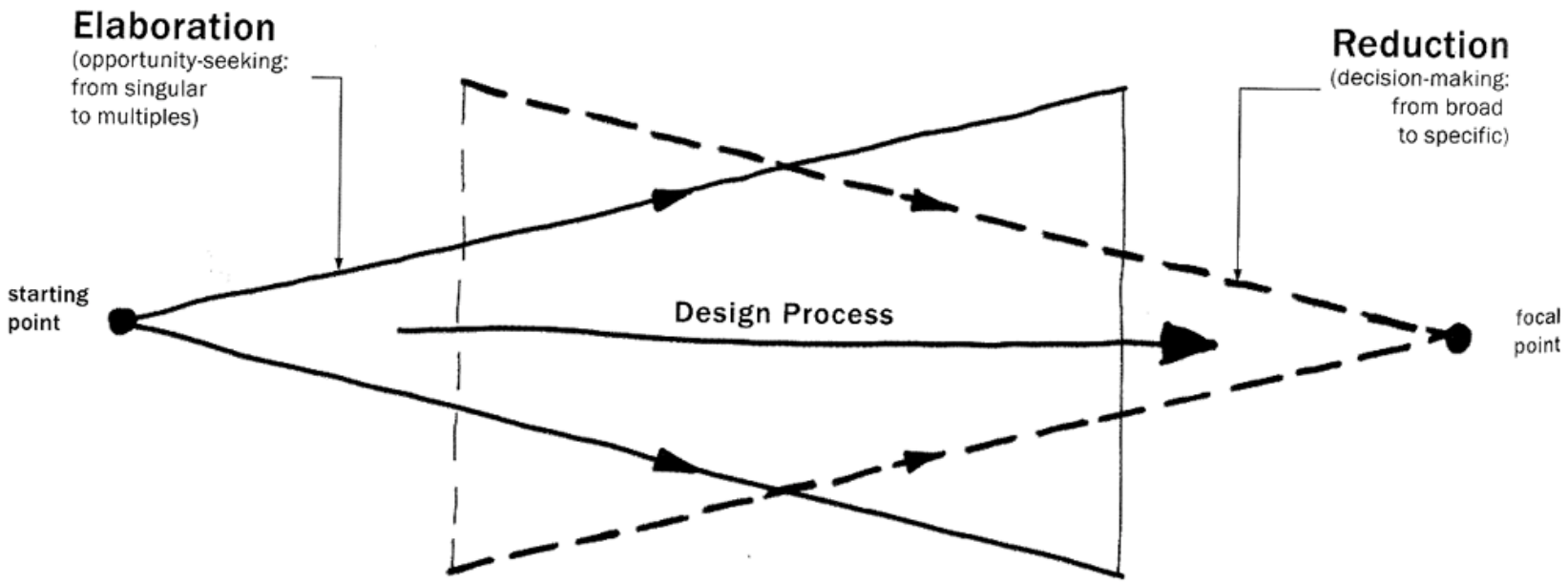

Exemplar
Incoming Data
/adc0
/adc1
/adc2
/adc3
/adc4
/adc5
/adc6
/adc7
Filter for /adc0
Scale Y-Axis
Invert Signal (x=-x)
Scale [0..1..2]
From: Center
Offset
Smooth
Amount of smoothing
none
lots
Signal Visualiations
Time scale:
/adc0
max
Events for /adc0
Name:
Send as:
Remove
Bent
/out1
Reset
Name:
Send as:
Remove
Extended
/out3
Reset
Add
Overview of all events:
Events for Sensor /adc0
Event Bent - outputs to: /out1
Marked Region
Event Extended - outputs to: /out3
Events for Sensor /adc3
Event TiltLeft - outputs to: /key/left
Event TiltRight - outputs to: /key/right
Events for Sensor /adc4
Event Foot Pedal - outputs to: /key/enter
Make: PROJECTS
Making Things Talk
Practical Methods for Connecting Physical Objects
PROJECTS AND IDEAS TO CREATE TALKING OBJECTS FROM ANYTHING
O'REILLY
Tom Igoe

Mini Talk #1 (research):

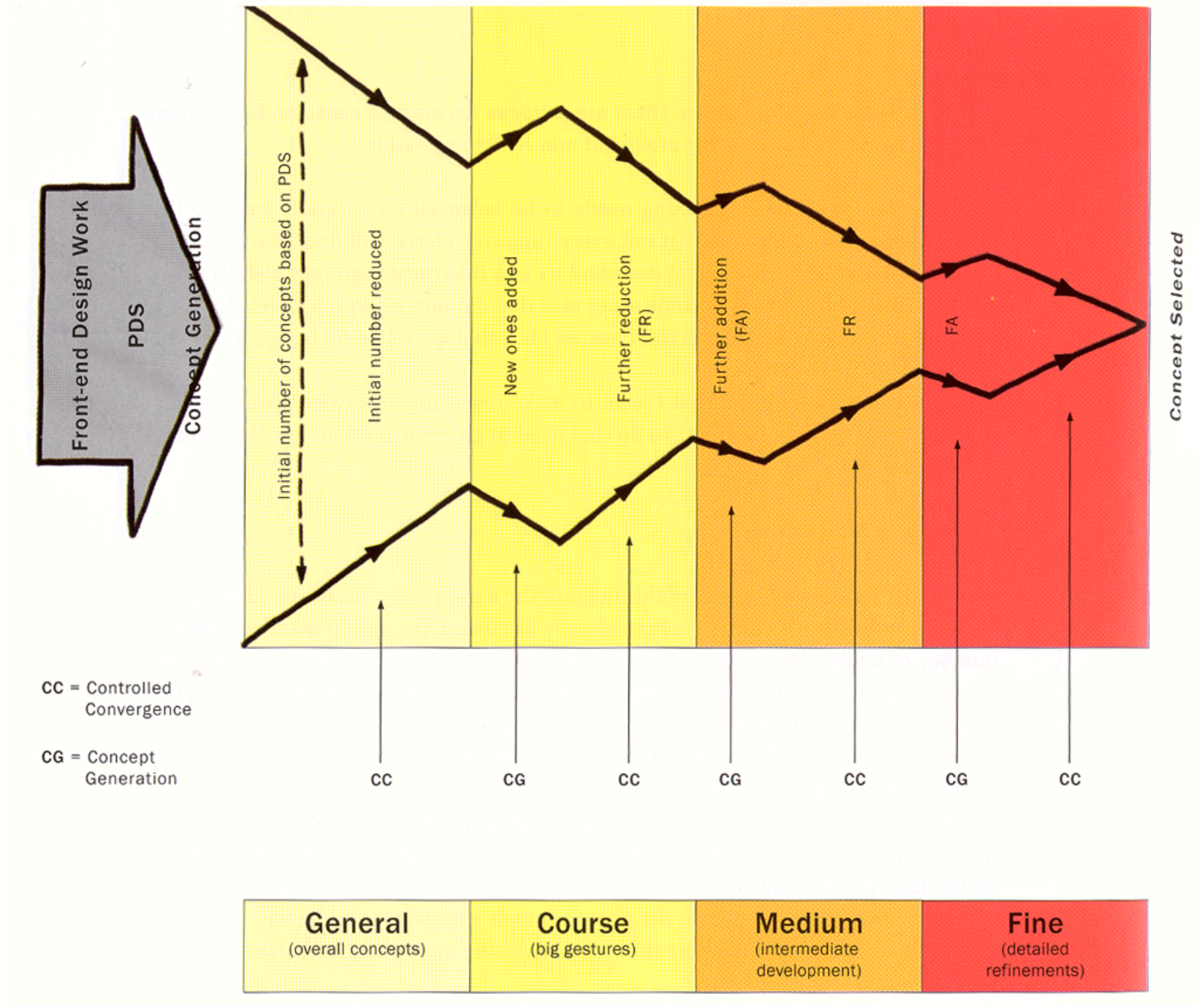
It's all about exploration!

"The best way to have a good idea is
to have lots of ideas."

-Linus Pauling

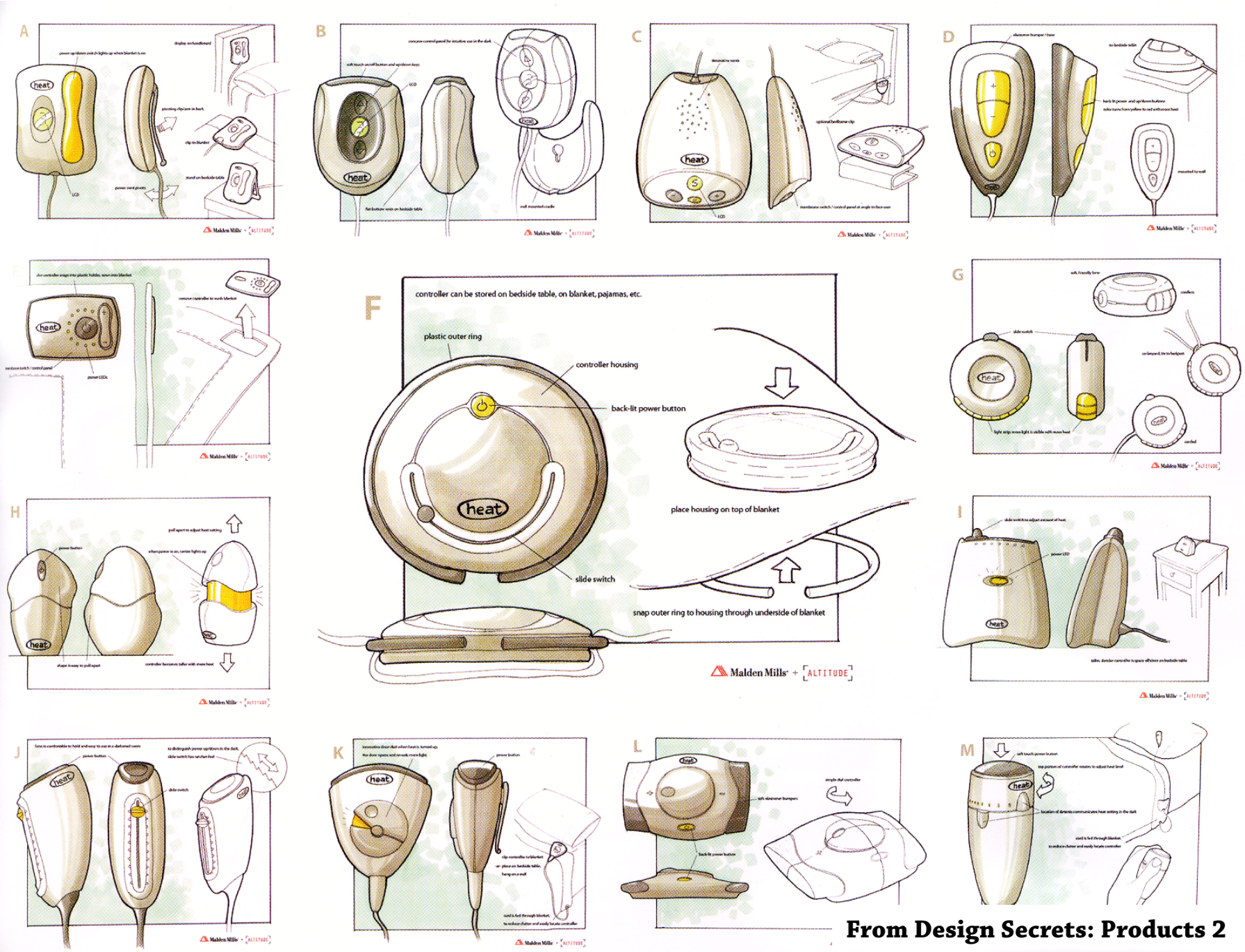


[Buxton, Sketching User Experiences]



[Buxton, Sketching User Experiences]

Prototypes for the Microsoft mouse From Moggridge, Designing Interactions, Chapter 2



From Design Secrets: Products 2

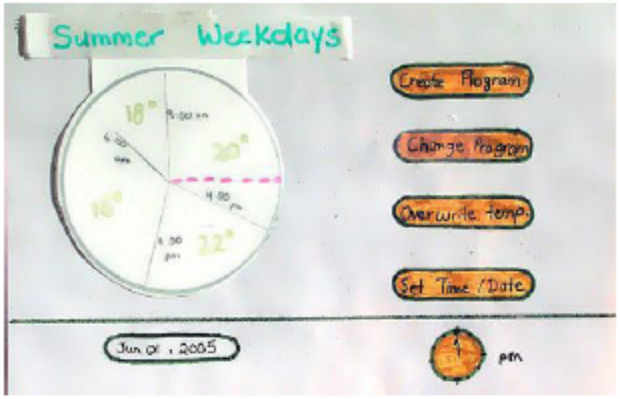


Figure 1. The "Circular" paper prototype

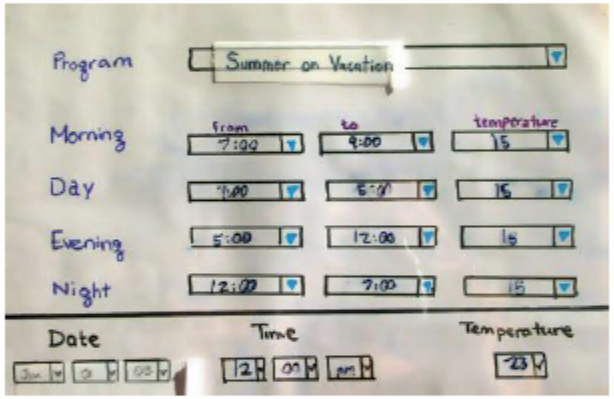


Figure 2. The "Tabular" paper prototype

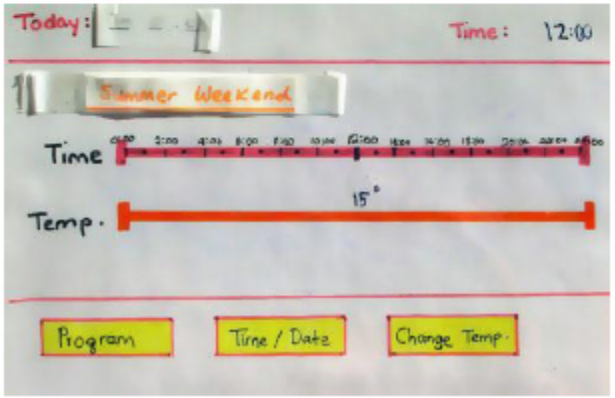
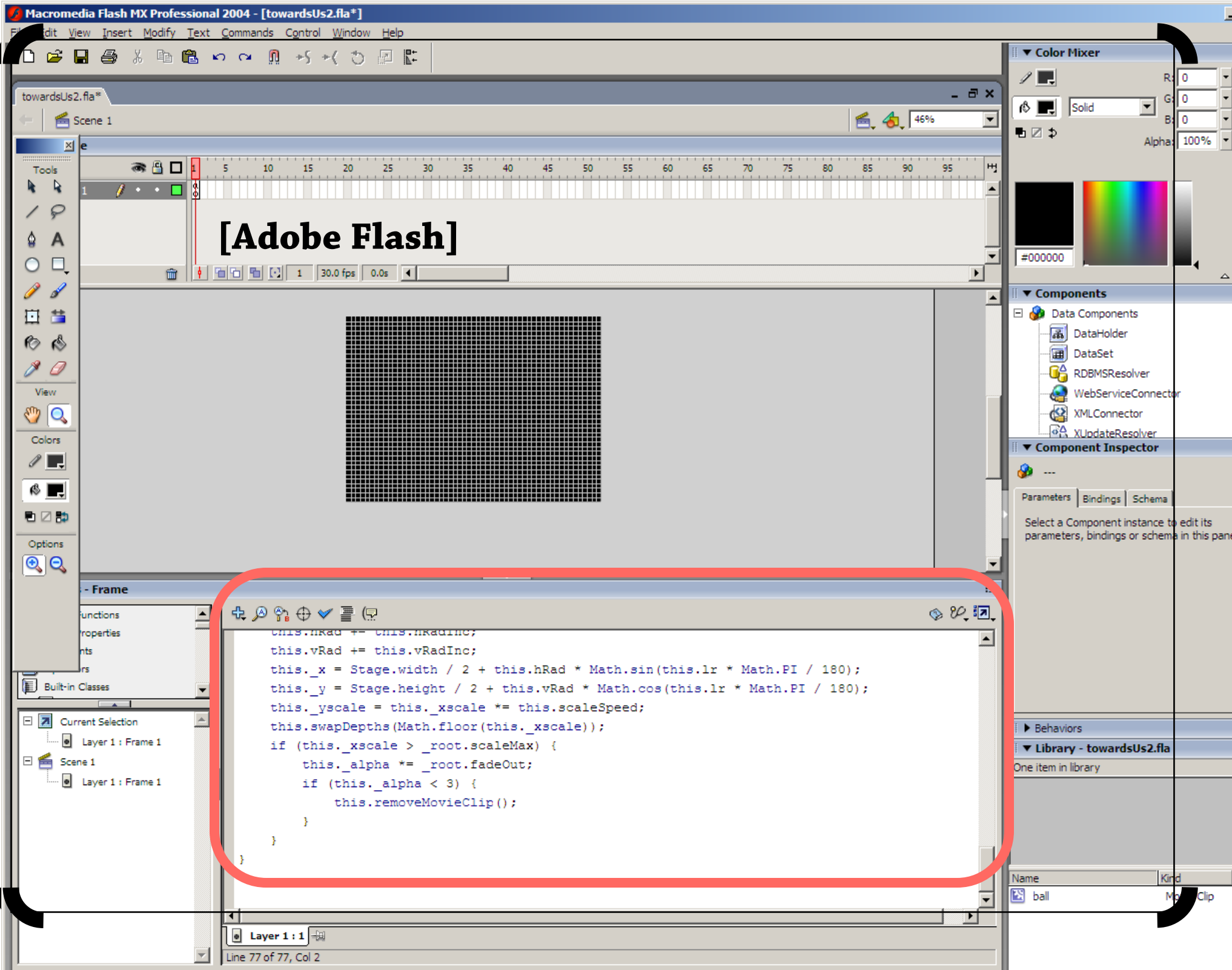
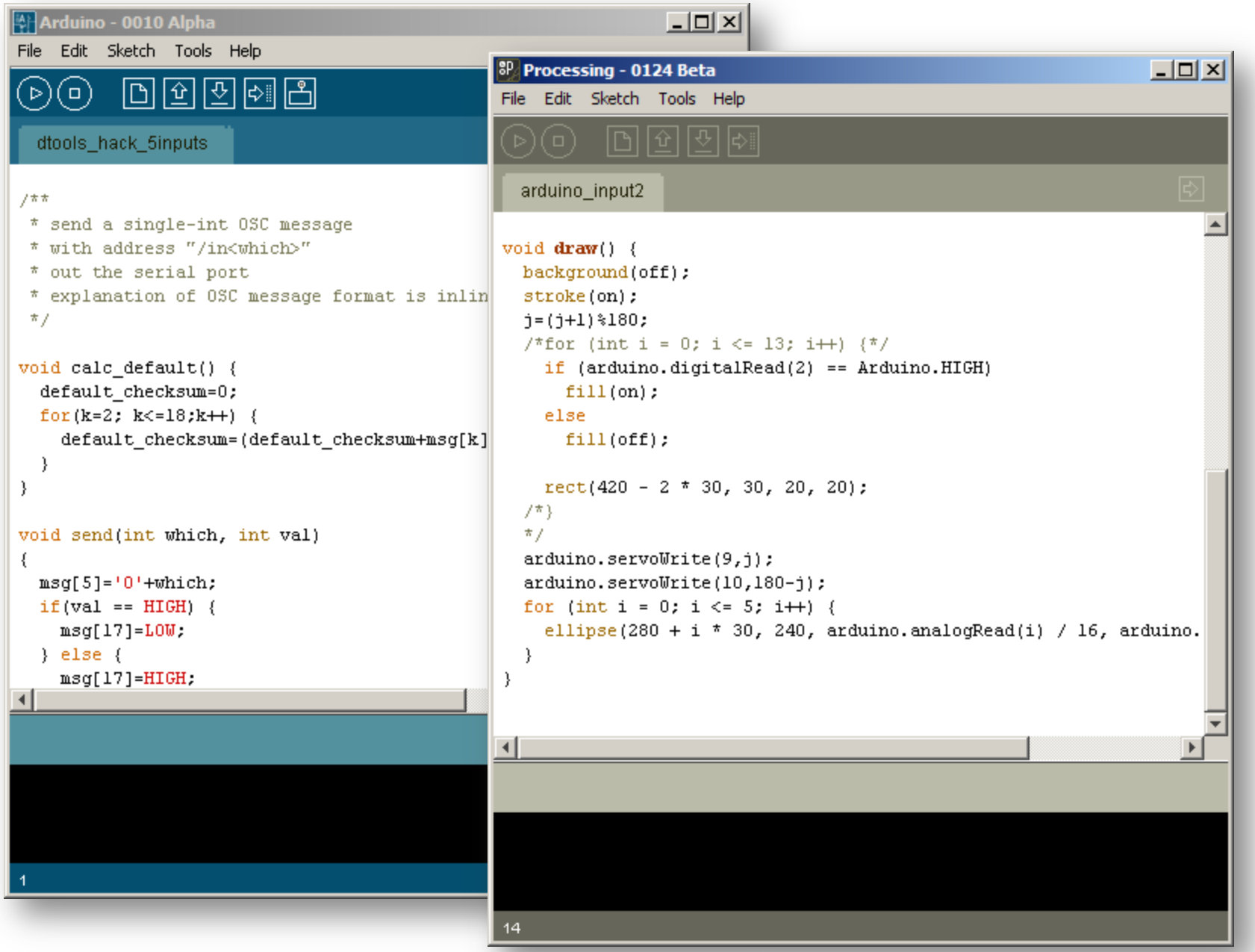


Figure 3. The "Linear" paper prototype

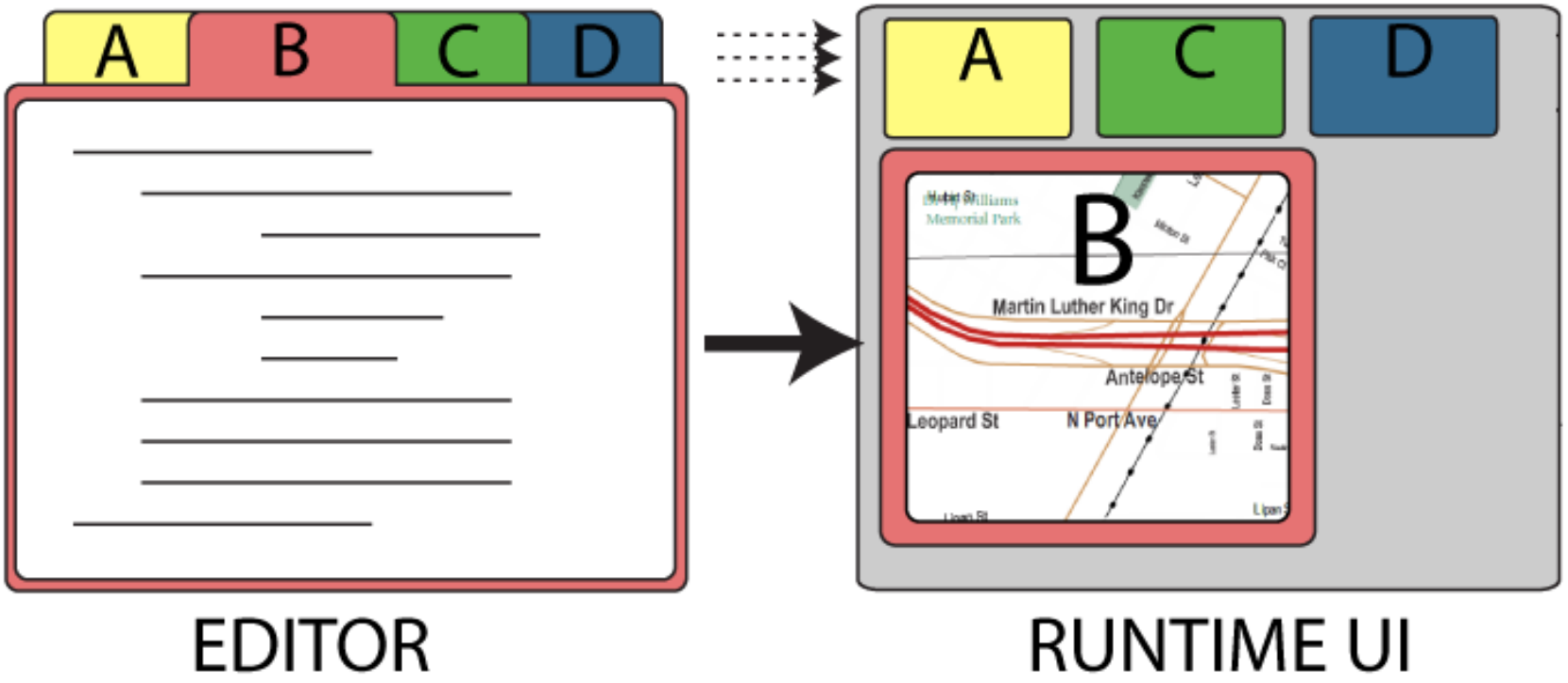
Tohidi et al, CHI 2006


Macromedia Flash MX Professional 2004 - [towardsUs2.fla*]
File Edit View Insert Modify Text Commands Control Window Help
towardsUs2.fla*
Scene 1
46%
[Adobe Flash]
Tools
View
Colors
Options
Color Mixer
Solid
R: 0
G: 0
B: 0
Alpha: 100%
#000000
Components
Data Components
DataHolder
DataSet
RDBMSResolver
WebServiceConnector
XMLConnector
XUpdateResolver
Component Inspector
Parameters
Bindings
Schema
Select a Component instance to edit its parameters, bindings or schema in this pane
Behaviors
Library - towardsUs2.fla
One item in library
Name
Kind
ball
Movie Clip
- Frame
Built-in Classes
Current Selection
Layer 1 : Frame 1
Scene 1
Layer 1 : Frame 1
this.vRad += this.vRadInc;
this._x = Stage.width / 2 + this.hRad * Math.sin(this.lr * Math.PI / 180);
this._y = Stage.height / 2 + this.vRad * Math.cos(this.lr * Math.PI / 180);
this._yscale = this._xscale *= this.scaleSpeed;
this.swapDepths(Math.floor(this._xscale));
if (this._xscale > _root.scaleMax) {
this._alpha *= _root.fadeOut;
if (this._alpha < 3) {
this.removeMovieClip();
}
}
}
Layer 1 : 1
Line 77 of 77, Col 2

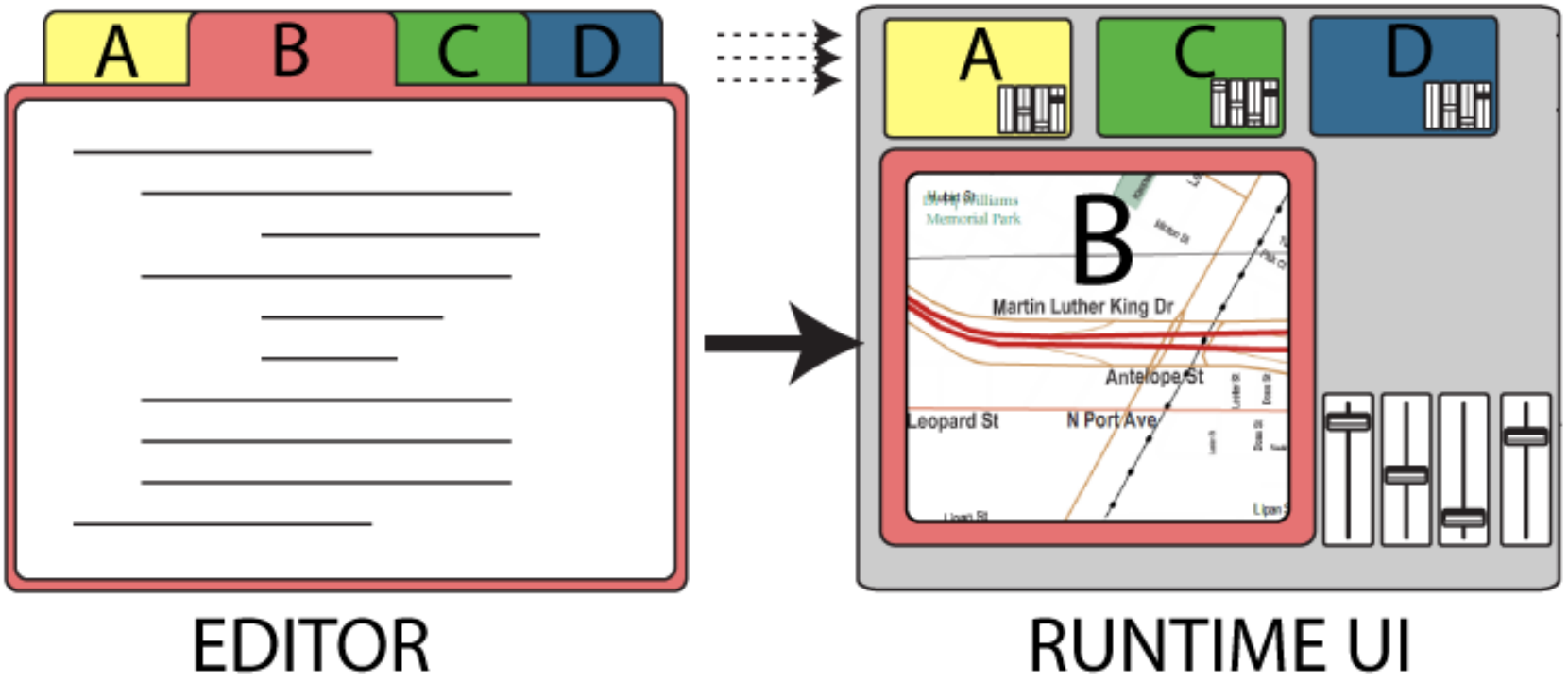

Arduino - 0010 Alpha
File Edit Sketch Tools Help
dtools_hack_5inputs
/**
* send a single-int OSC message
* with address "/in<which>"
* out the serial port
* explanation of OSC message format is inlin
*/
void calc_default() {
default_checksum=0;
for(k=2; k<=18;k++) {
default_checksum=(default_checksum+msg[k]
}
}
void send(int which, int val)
{
msg[5]='0'+which;
if(val == HIGH) {
msg[17]=LOW;
} else {
msg[17]=HIGH;
1
Processing - 0124 Beta
File Edit Sketch Tools Help
arduino_input2
void draw() {
background(off);
stroke(on);
j=(j+1)%180;
/*for (int i = 0; i <= 13; i++) {*/
if (arduino.digitalRead(2) == Arduino.HIGH)
fill(on);
else
fill(off);
rect(420 - 2 * 30, 30, 20, 20);
/*}
*/
arduino.servoWrite(9,j);
arduino.servoWrite(10,180-j);
for (int i = 0; i <= 5; i++) {
ellipse(280 + i * 30, 240, arduino.analogRead(i) / 16, arduino.
}
}
14

Juxtapose

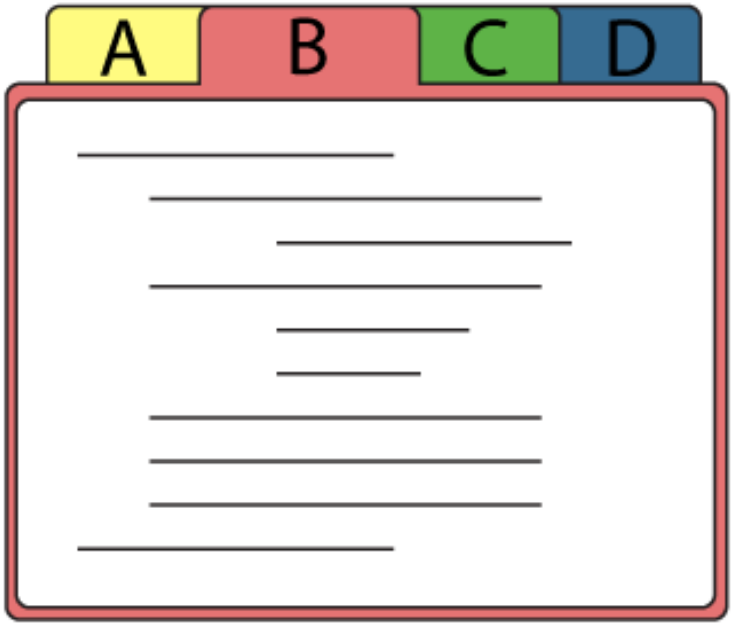
Juxtapose

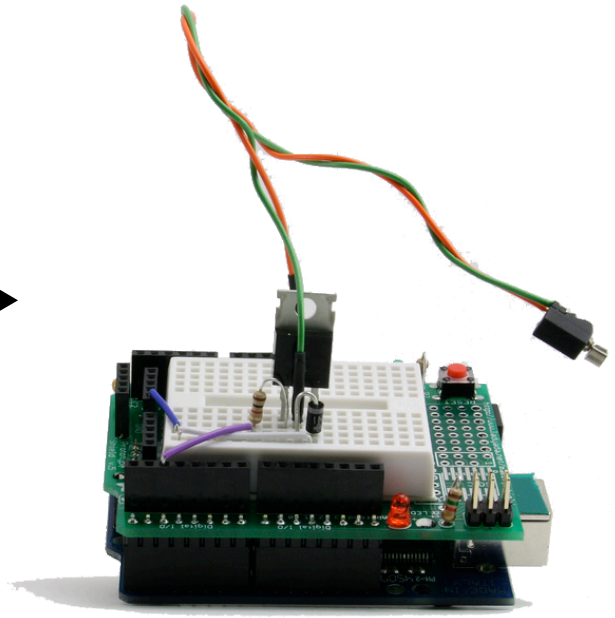

A
B
C
D
A
C
D
B
Martin Luther King Dr
Antelope St
Leopard St
N Port Ave
EDITOR
RUNTIME UI

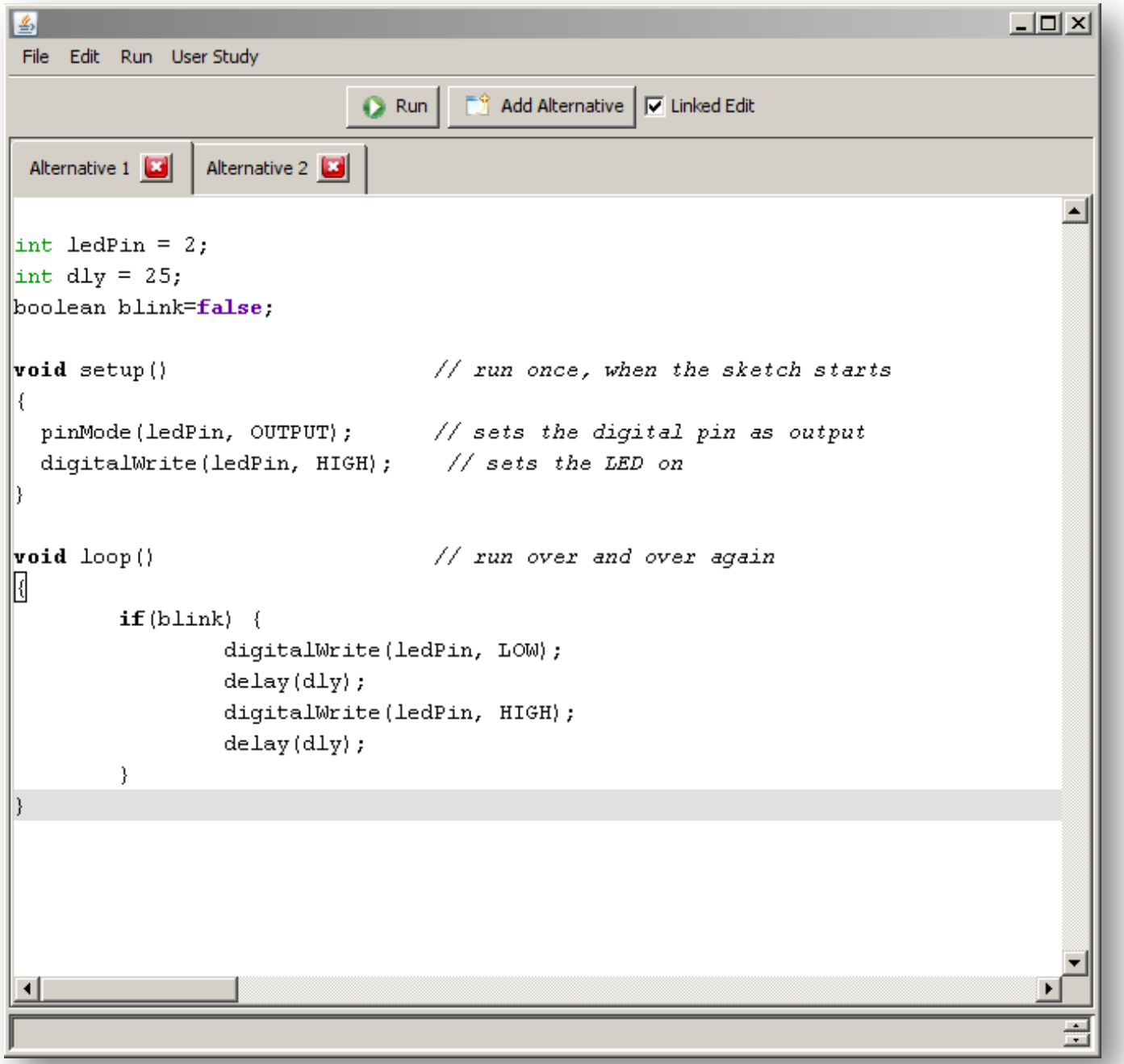
Juxtapose

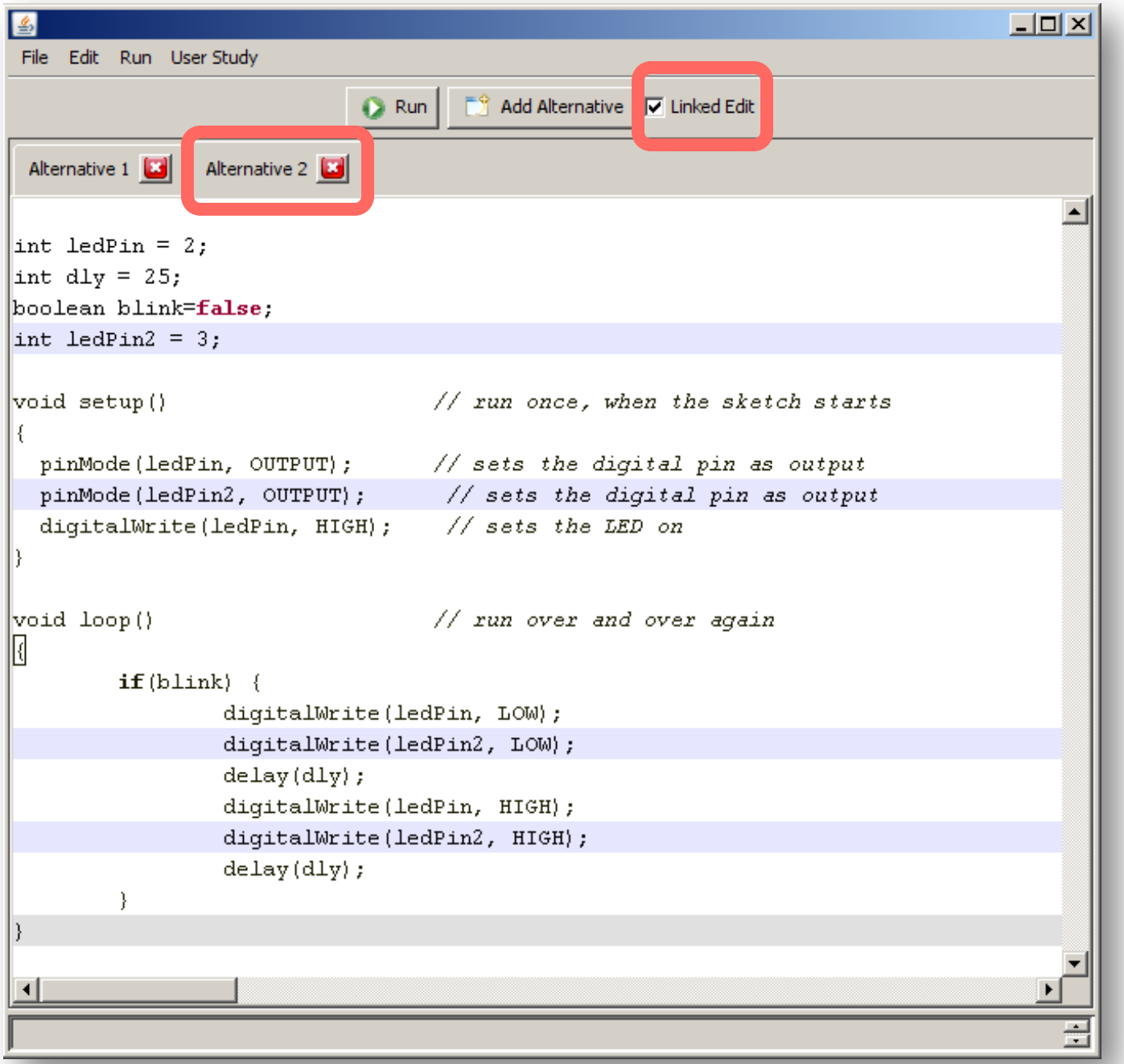

A
B
C
D
A
C
D
B
Martin Luther King Dr
Antelope St
Leopard St
N Port Ave
EDITOR
RUNTIME UI

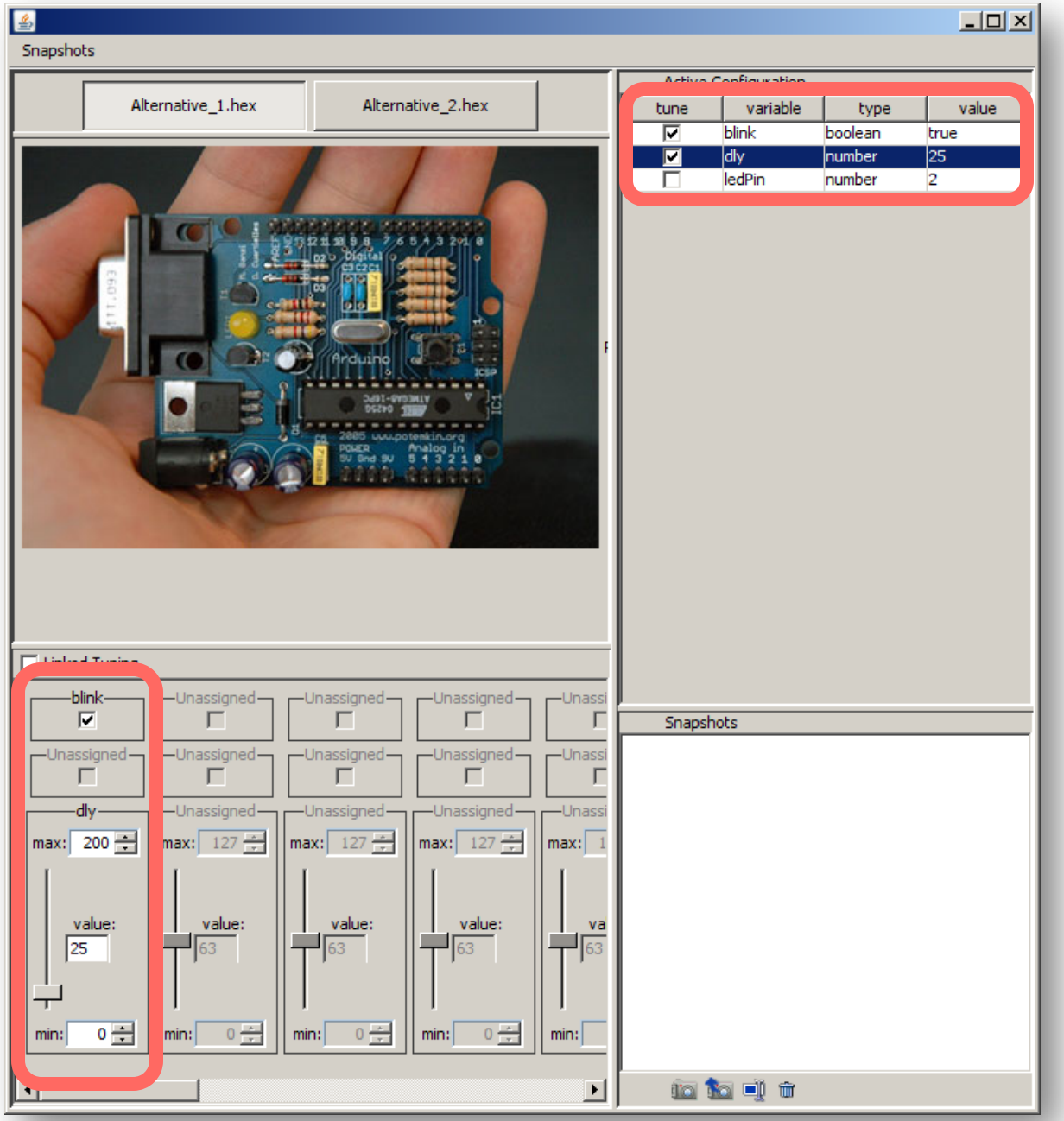
Juxtapose for Arduino

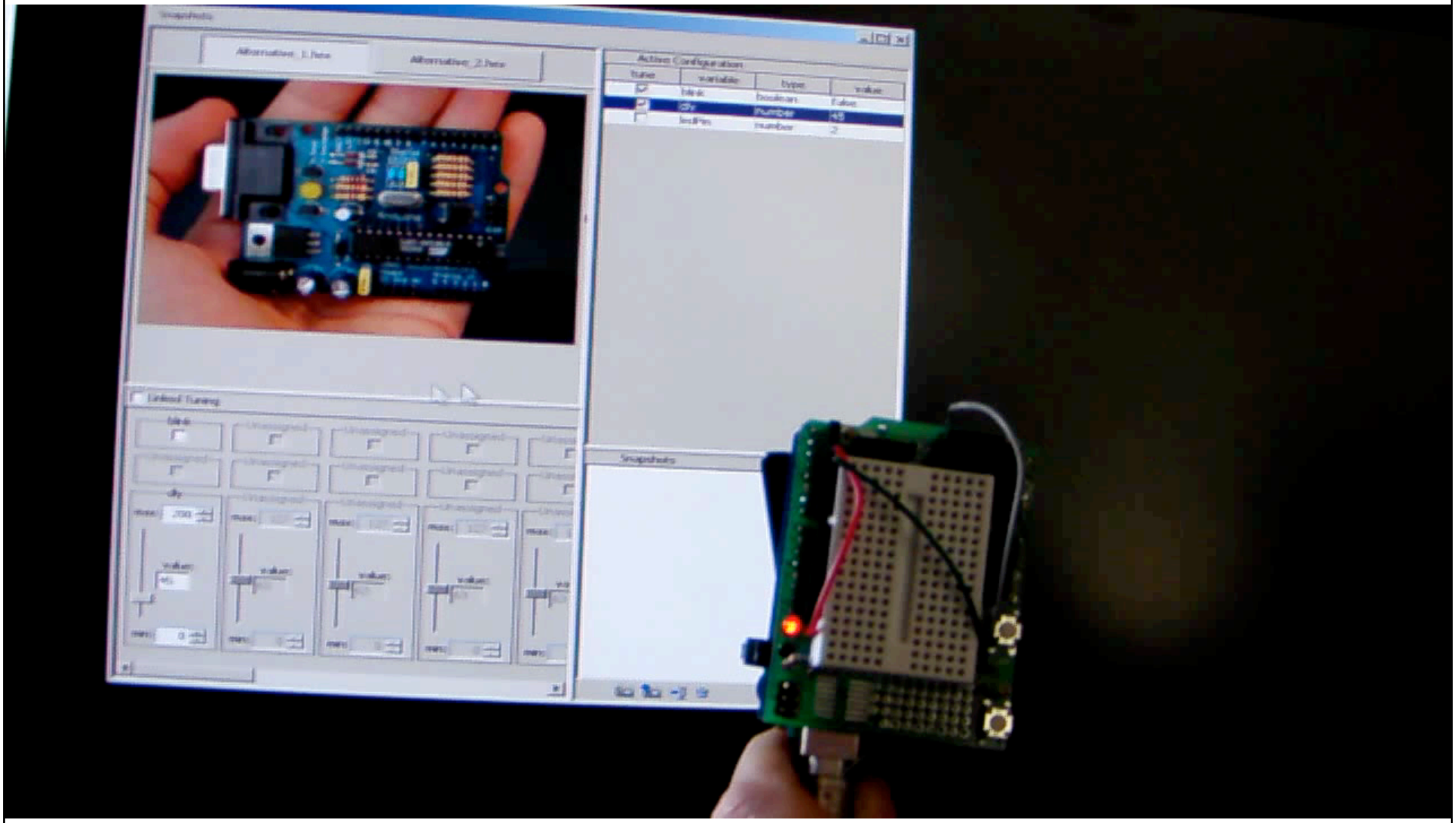


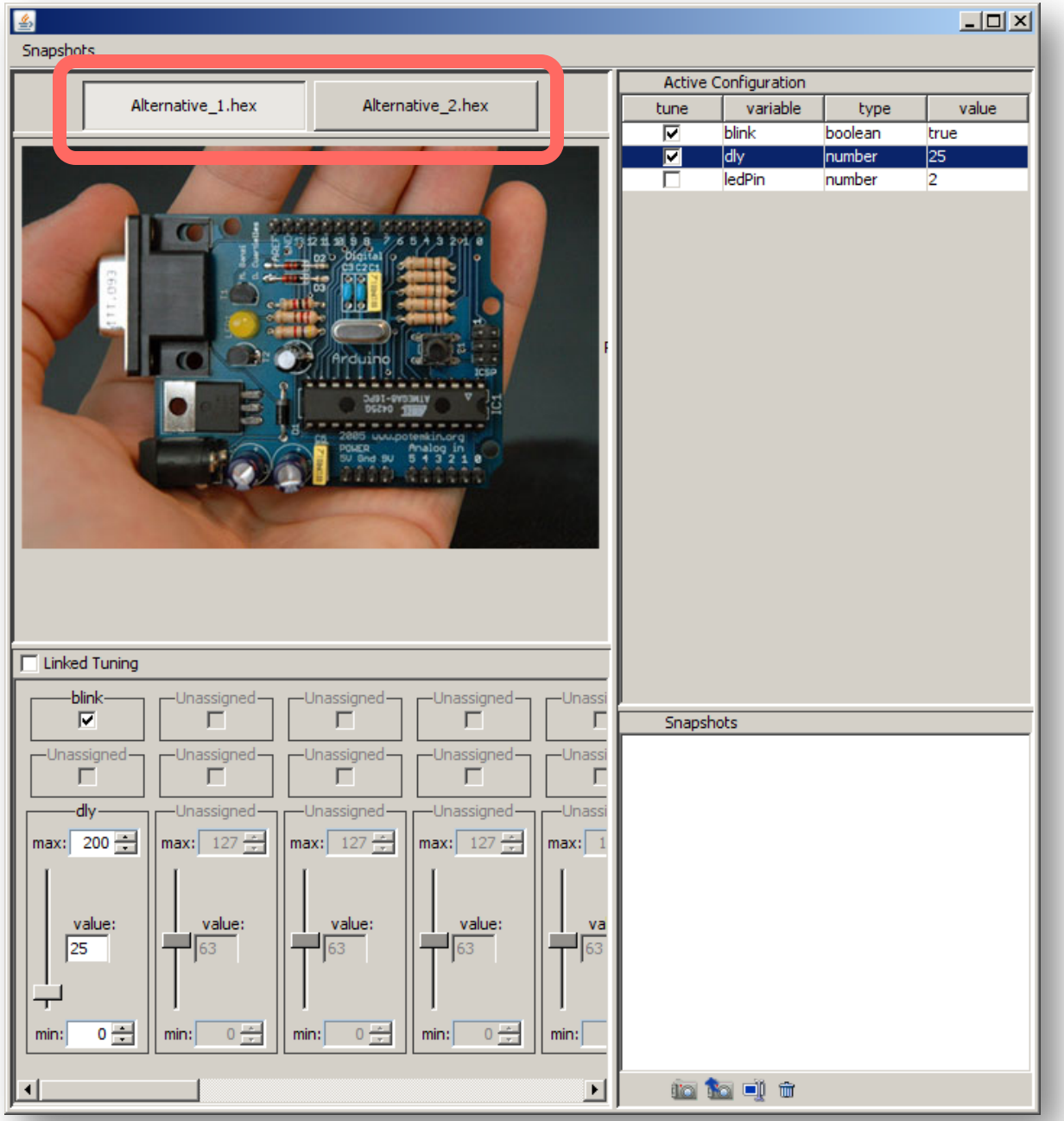


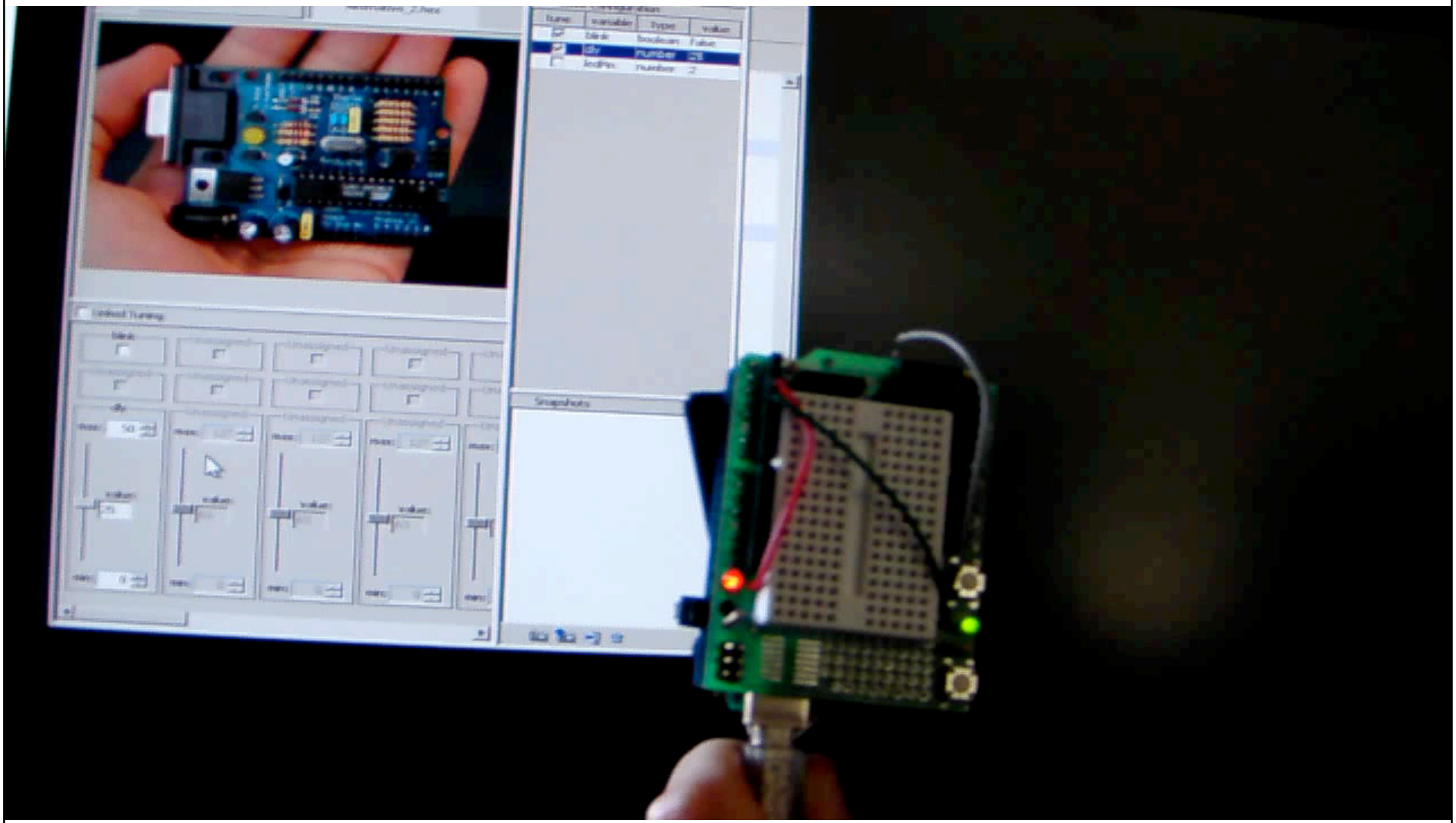

File Edit Run User Study
Run
Add Alternative
Linked Edit
Alternative 1
Alternative 2
int ledPin = 2;
int dly = 25;
boolean blink=false;
void setup()   // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT);   // sets the digital pin as output
  digitalWrite(ledPin, HIGH);   // sets the LED on
}
void loop()   // run over and over again
{
        if(blink) {
                digitalWrite(ledPin, LOW);
                delay(dly);
                digitalWrite(ledPin, HIGH);
                delay(dly);
        }
}


File Edit Run User Study
Run
Add Alternative
Linked Edit
Alternative 1
Alternative 2
int ledPin = 2;
int dly = 25;
boolean blink=false;
int ledPin2 = 3;
void setup() // run once, when the sketch starts
{
pinMode(ledPin, OUTPUT); // sets the digital pin as output
pinMode(ledPin2, OUTPUT); // sets the digital pin as output
digitalWrite(ledPin, HIGH); // sets the LED on
}
void loop() // run over and over again
{
if(blink) {
digitalWrite(ledPin, LOW);
digitalWrite(ledPin2, LOW);
delay(dly);
digitalWrite(ledPin, HIGH);
digitalWrite(ledPin2, HIGH);
delay(dly);
}
}

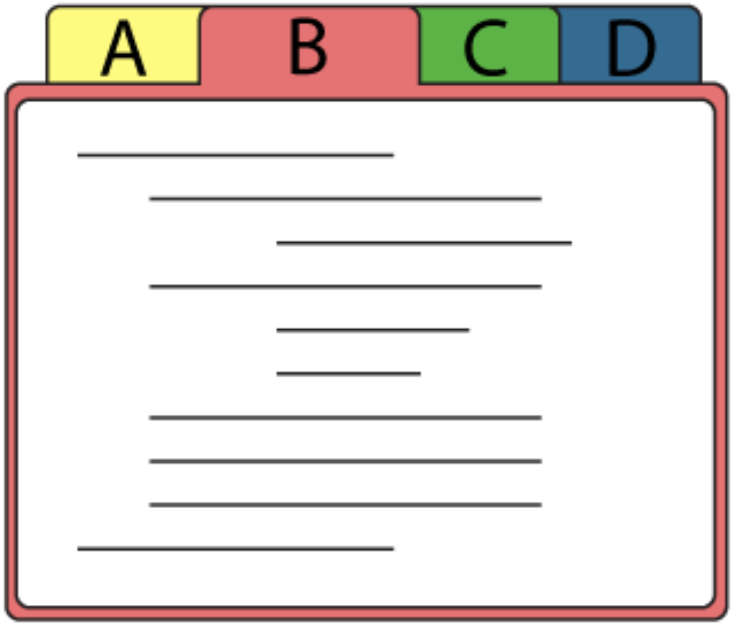

Snapshots
Alternative_1.hex
Alternative_2.hex
tune
variable
type
value
blink
boolean
true
dly
number
25
ledPin
number
2
Snapshots
blink
Unassigned
dly
max:
200
value:
25
min:
0
127
63

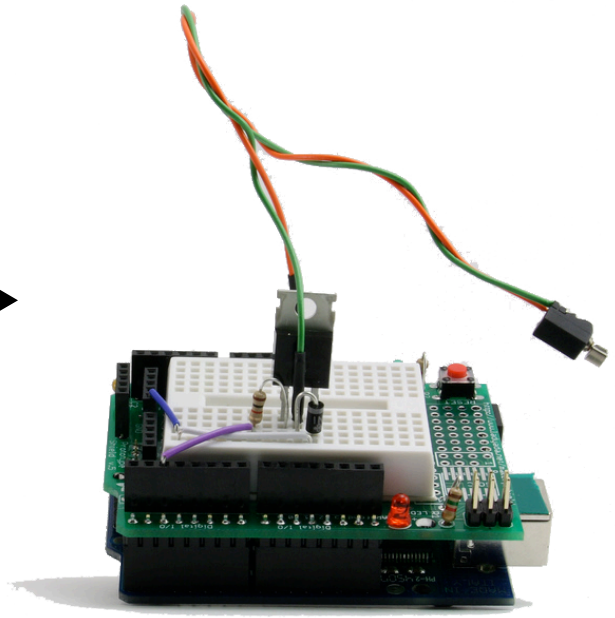

Active Configuration
Snapshots

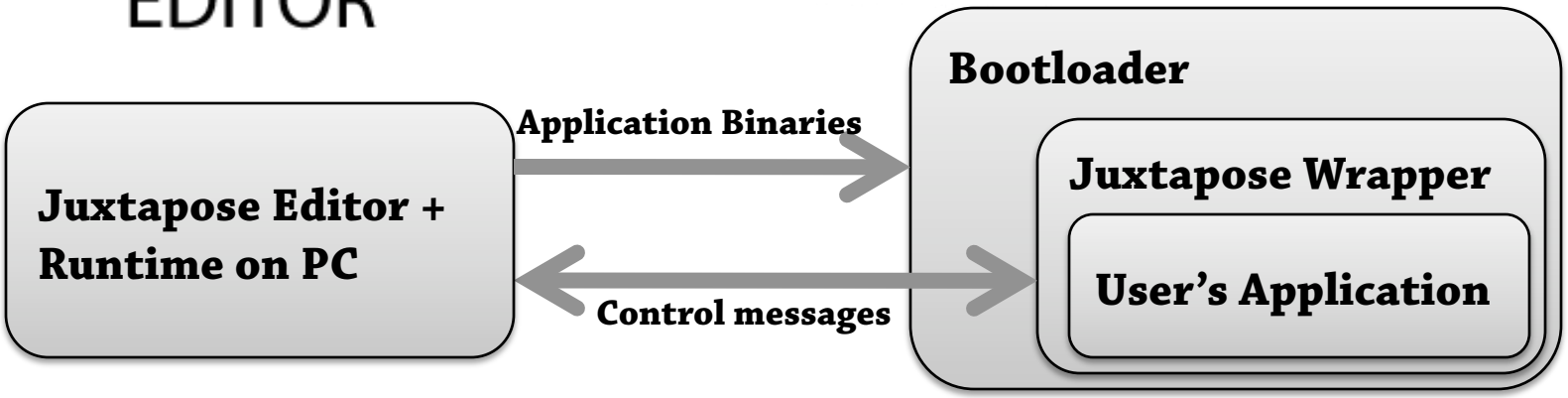

Snapshots
Alternative_1.hex
Alternative_2.hex
Active Configuration
tune
variable
type
value
blink
boolean
true
dly
number
25
ledPin
number
2
Linked Tuning
blink
Unassigned
dly
max:
200
127
value:
25
63
min:
0
Snapshots



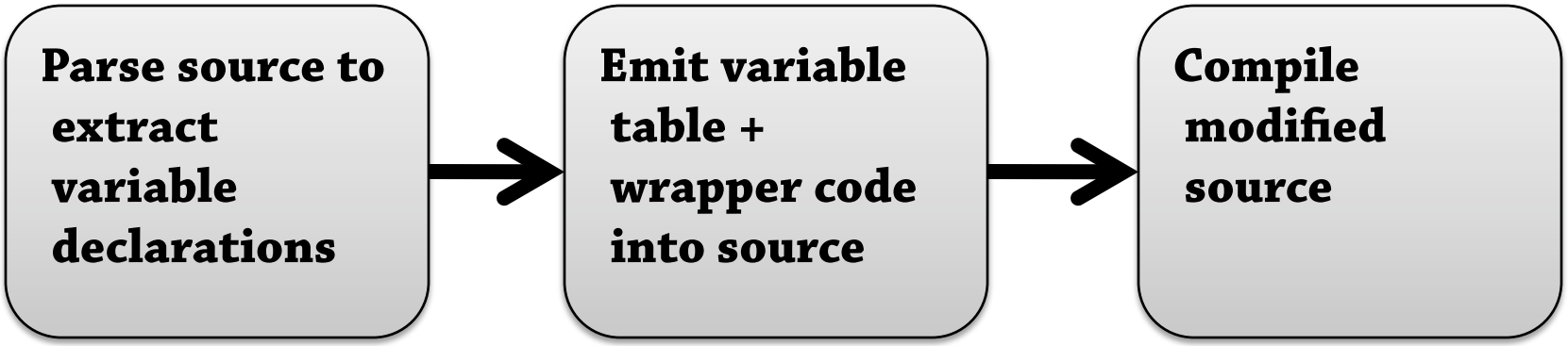
Juxtapose for Arduino







Compilation



“Hello World” Example


```
int dly = 25;
boolean blink=false;
int ledPin =2; //@IGNORE

void setup() {...}

void loop() {
   if(blink) {
    digitalWrite(ledPin, LOW);
     delay(dly);
     digitalWrite(ledPin, HIGH);
     delay(dly);
} }
```

Parse code and build table

```
int dly = 25;
boolean blink=false;
```

Var. Name	Type	Pointer
“dly”	int	&dly
“blink”	boolean	&blink

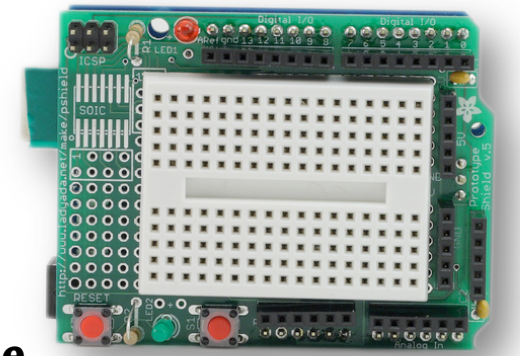
Parse code and build table

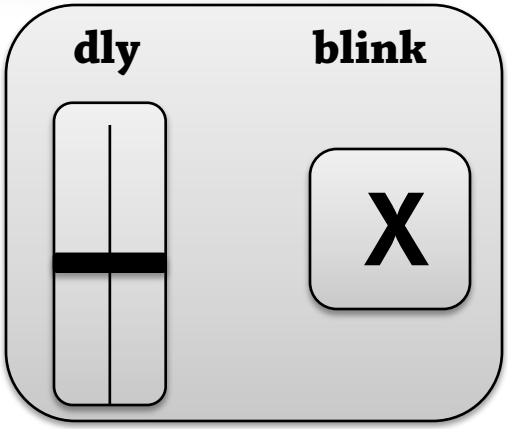

```
void initVarTable(void) {
  varTable[1].varName = PSTR("dly");
  varTable[1].varType = VAR_TYPE_INT;
  varTable[1].varPtr = &dly;

  varTable[2].varName = PSTR("blink");
  varTable[2].varType = VAR_TYPE_BOOLEAN;
  varTable[2].varPtr = &blink;
}

// plus a bunch of communication code...
```

At Runtime: startup

Serial message:

my variables are
“dly”, int, value 50
“blink”, boolean, value false

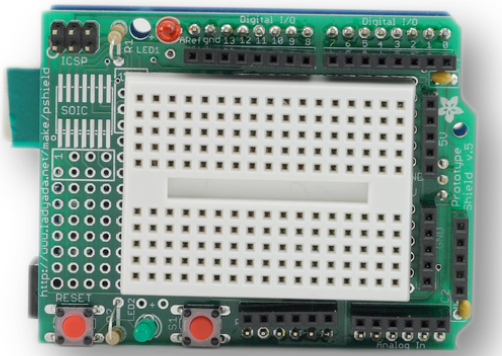


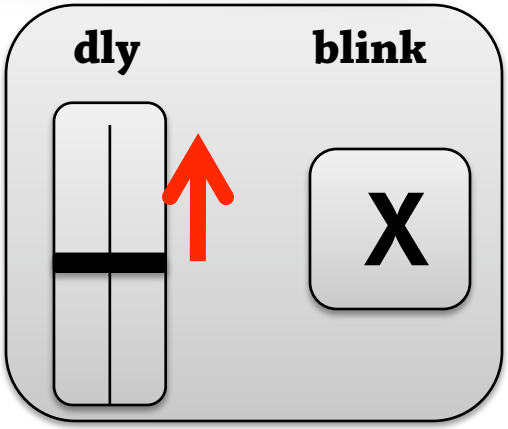


At Runtime: startup

Serial message:

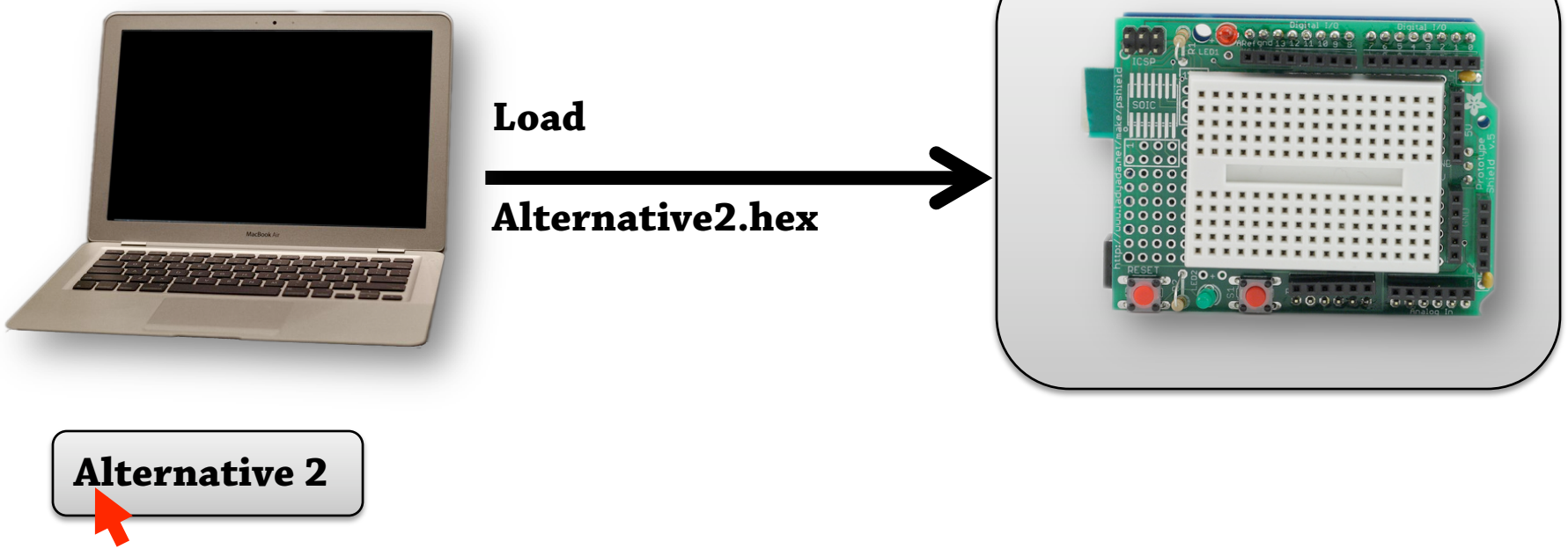
tune “dly” value 100






```
*(varTable[“dly”].varPtr)
    = 100;
```

Switching alternatives



Tradeoffs

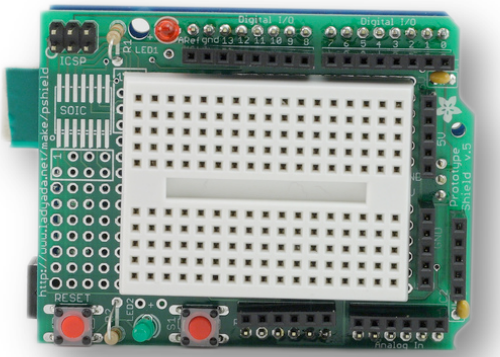
- What you gain:
  - Tuning “on the fly”
  - Relatively little overhead
- What you lose:
  - Use of a hardware serial port

Mini Talk #2 (not research):

No more zombie prototypes!

Write code here

Run code here



Store source here

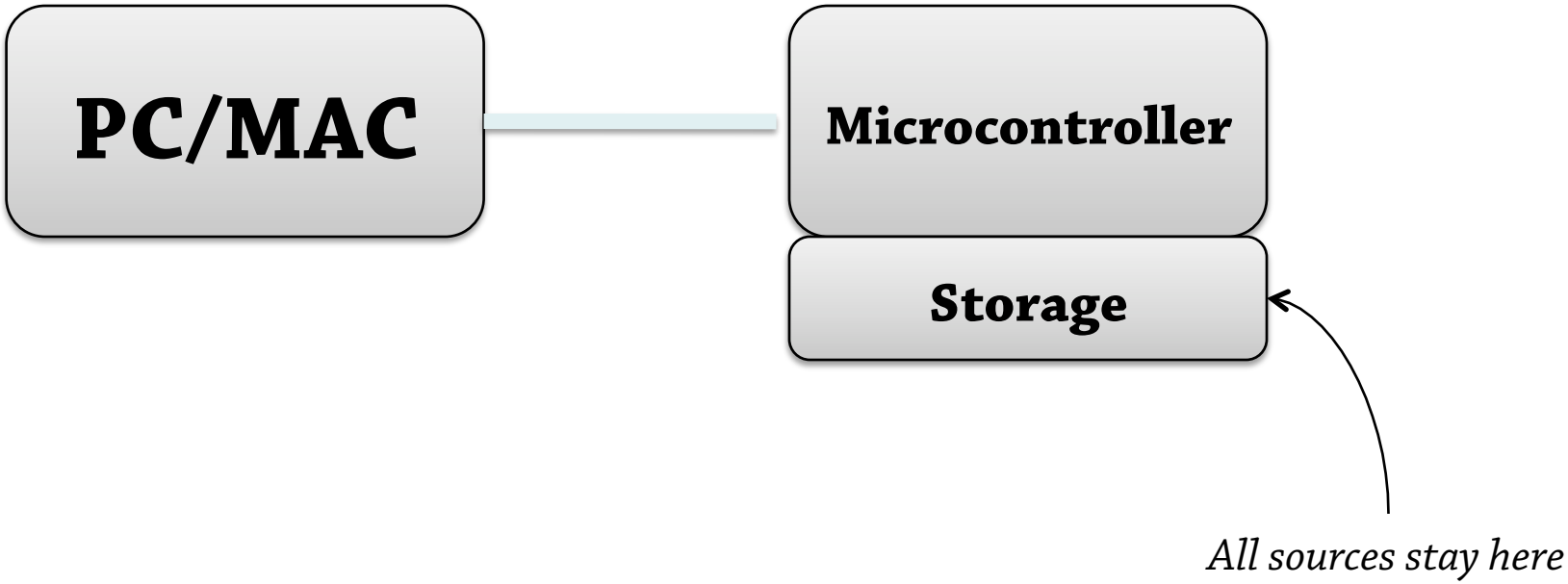
Store binaries here

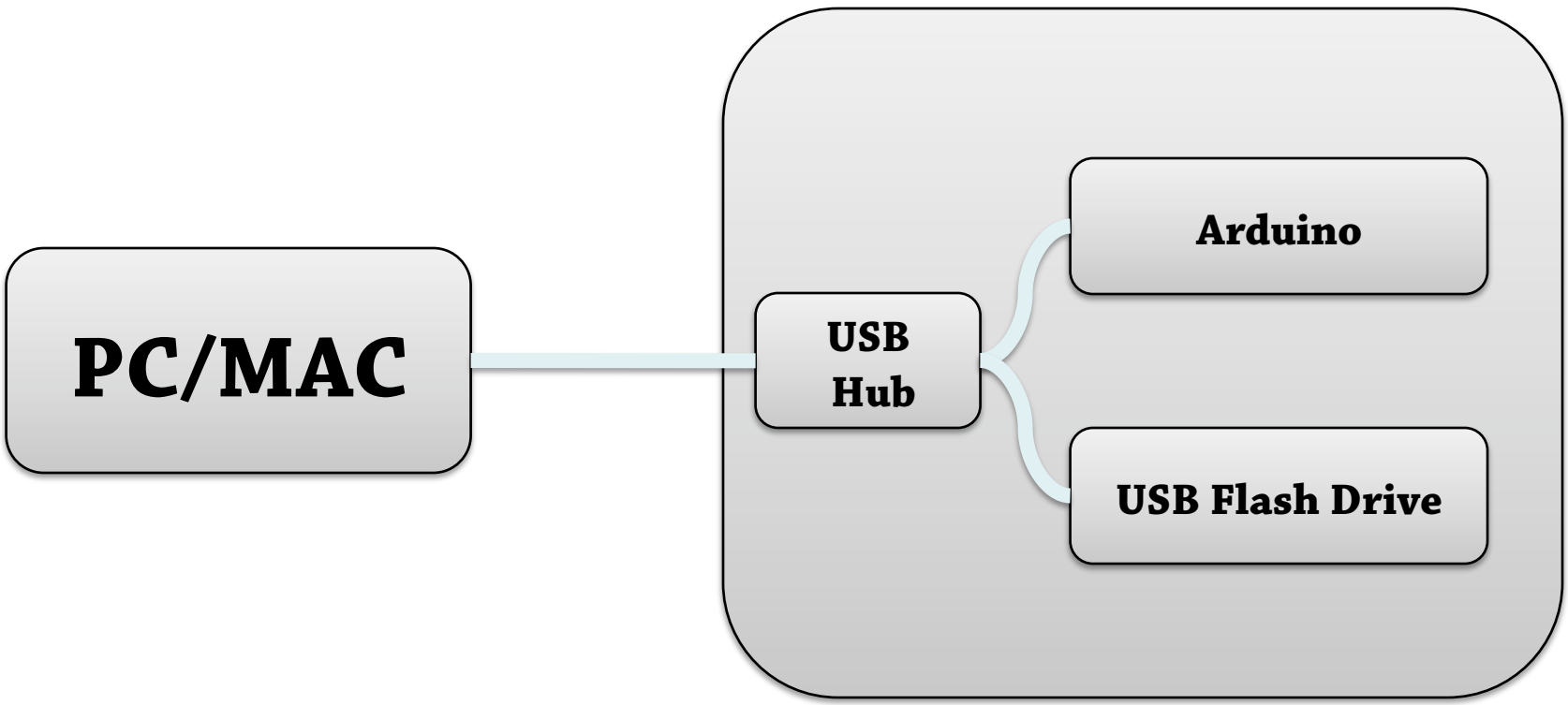
What could possibly go wrong?

- My dog ate the laptop
- My files are a total mess
- Uhm, that was 3 years ago…
- Oops, forgot that at _____

How $8 can save the day…

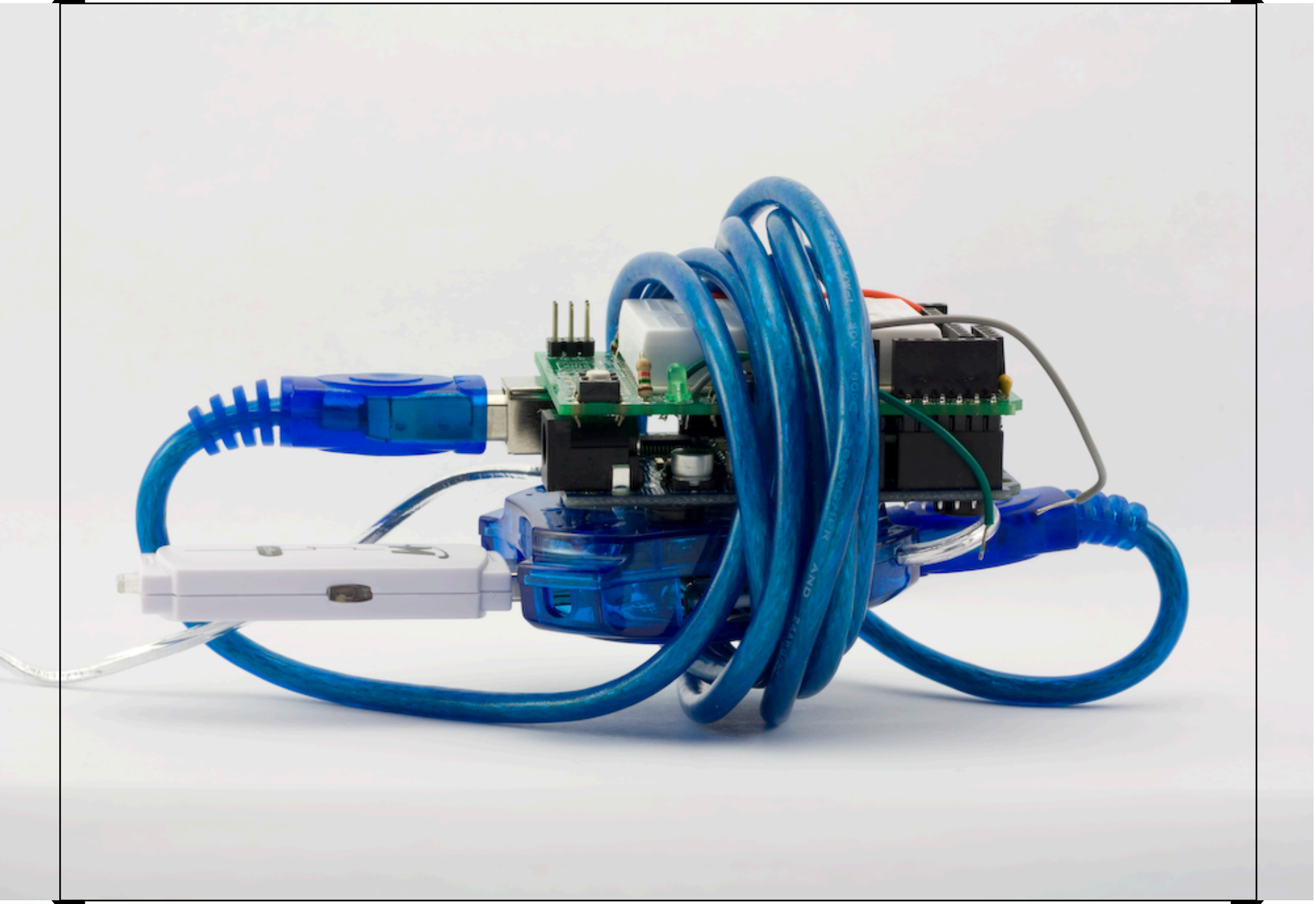

PC/MAC
Microcontroller

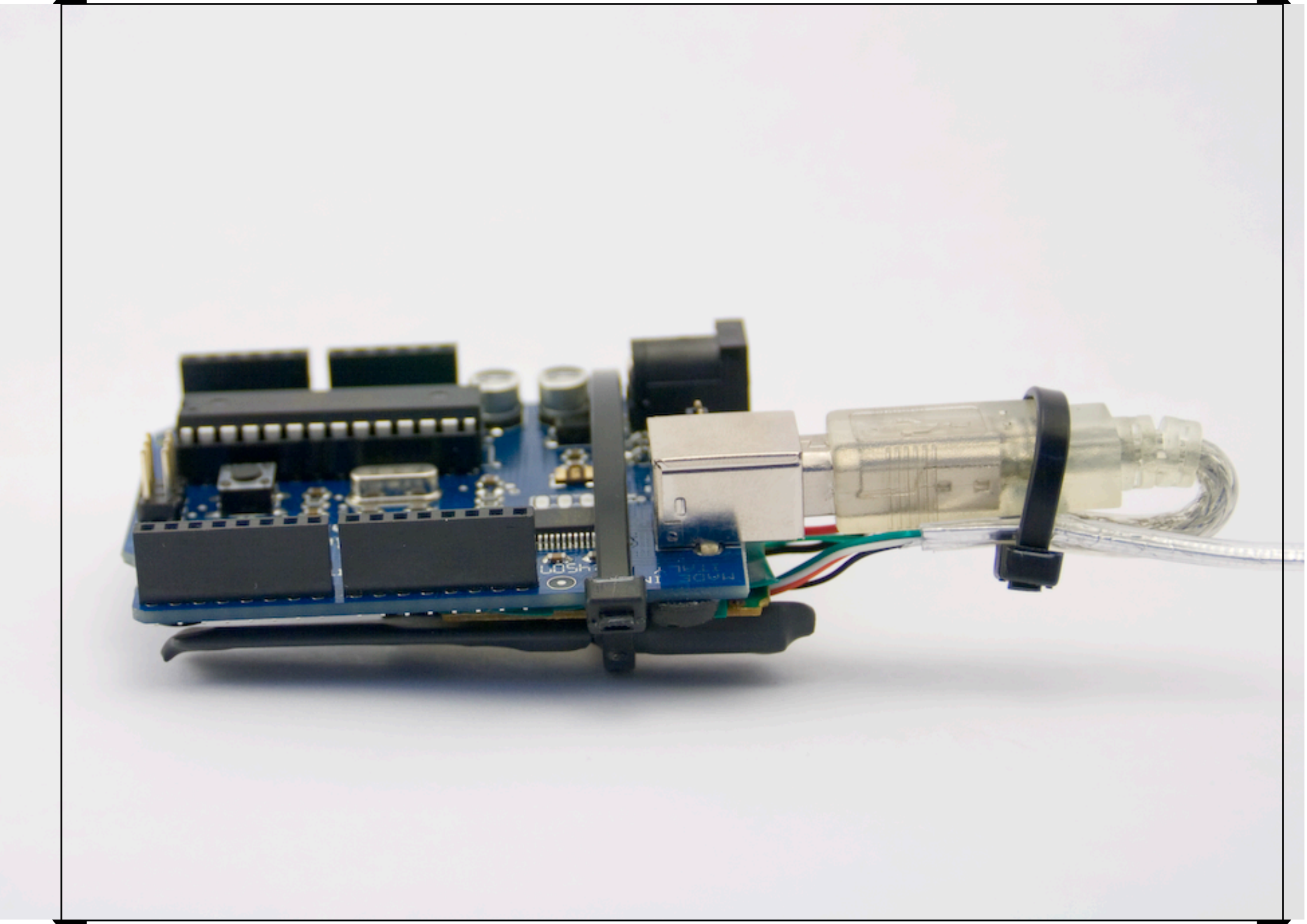

PC/MAC
Microcontroller
Storage
All sources stay here


PC/MAC
USB
Hub
Arduino
USB Flash Drive


USB HUB
4 Port Mini
USB
HB-018
HIRO USB 1.1 $ 4.99
HIRO USB 1.1 HUB
USB 4 PORT HUB
2;359
H22029 5348628


Memorex
128 MB
USB
Flash Drive
Portable Storage Device
Store, carry and share files
Great for photos or MP3 files
LED light indicates drive's status
Works with Windows, Mac and Linux
2.99
DEPT 2
7:7
332:332
5535430





Observations

- This can obviously be miniaturized.
- Upgrade the hardware (256MB,USB2.0) and run entire development environment directly off the stick
- Where should storage ideally be located? On a shield? Or on the board itself?

Final requests:

- Good design process examples (products & end-user modifications). The prototypes and discussion that went into them, alternatives tried, etc.
- Know of any open academic jobs?

http://**bjoern.org**
http://**hci.stanford.edu**
http://**ambidextrousmag.org**