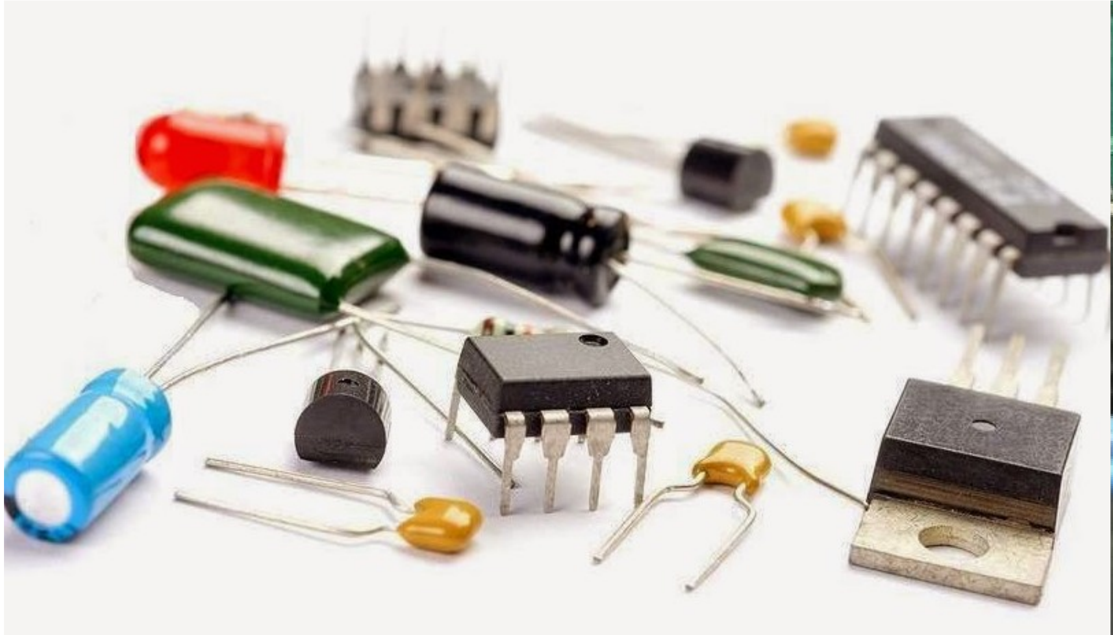
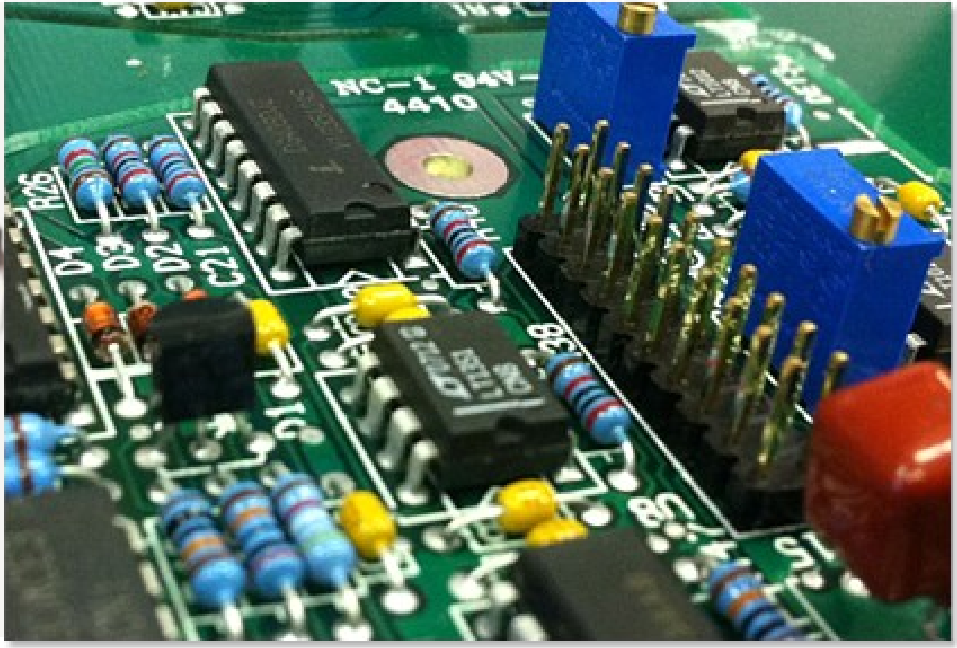
SP26 Electronic Projects for Artists II:

Programming for Interactivity

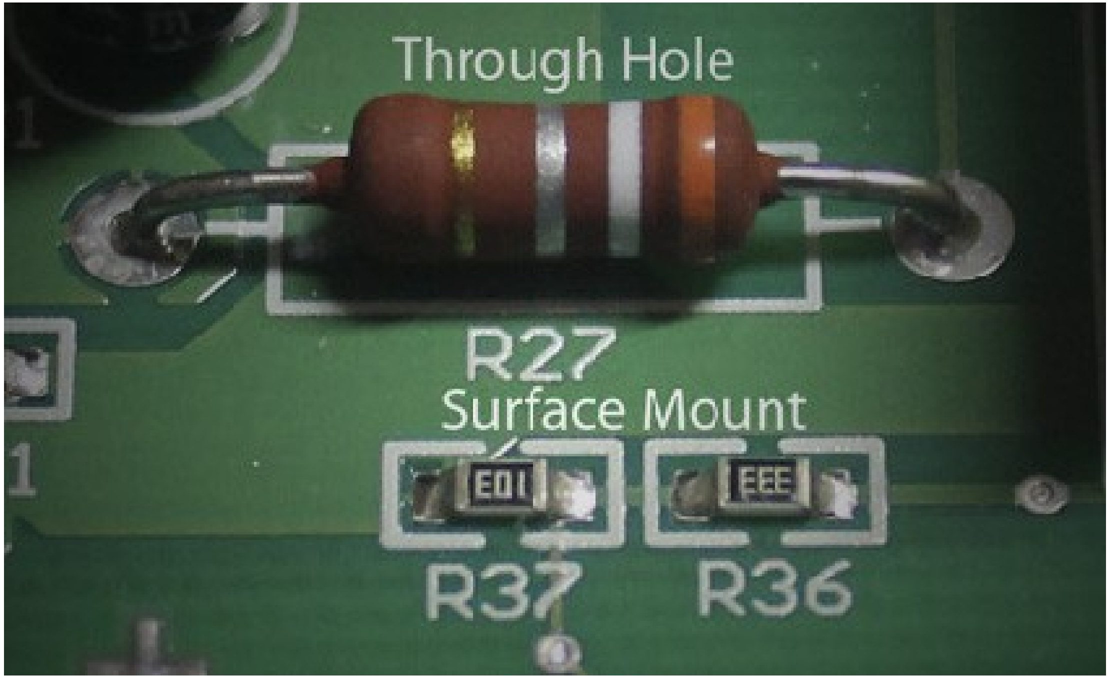
Arduino, Processing (p5js), Max, PureData and some JavaScript

Historically, electronic circuit design used "through-hole" components.

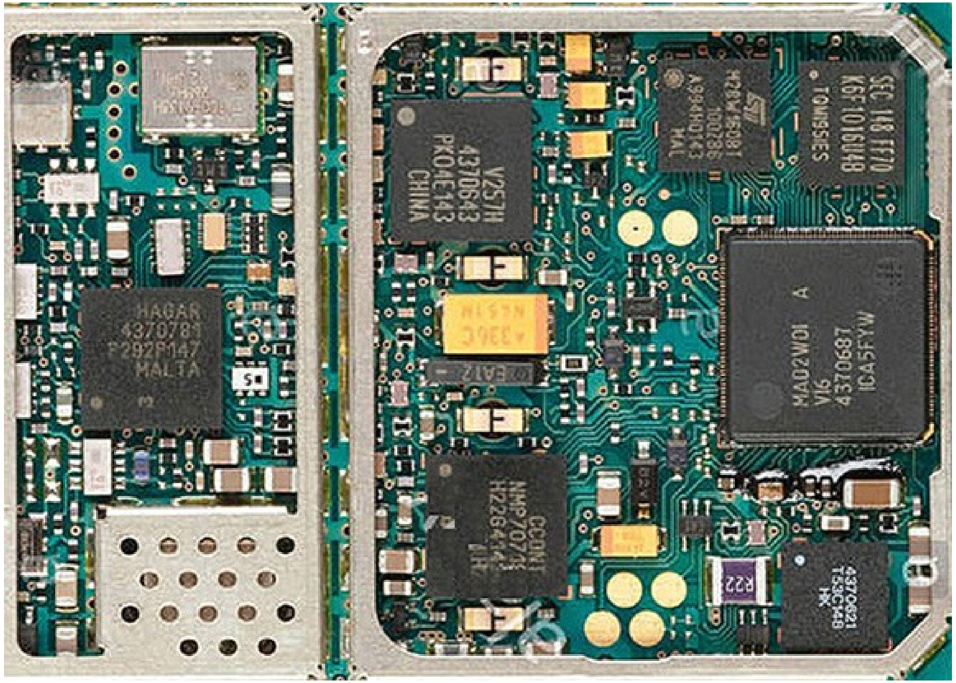


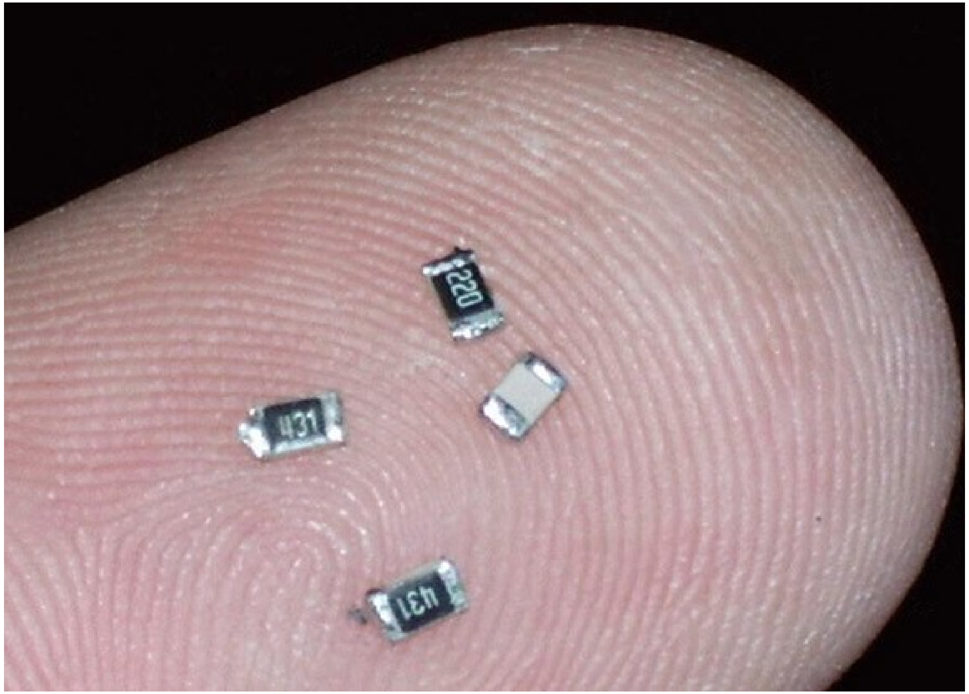


Still in use, also hybrid.

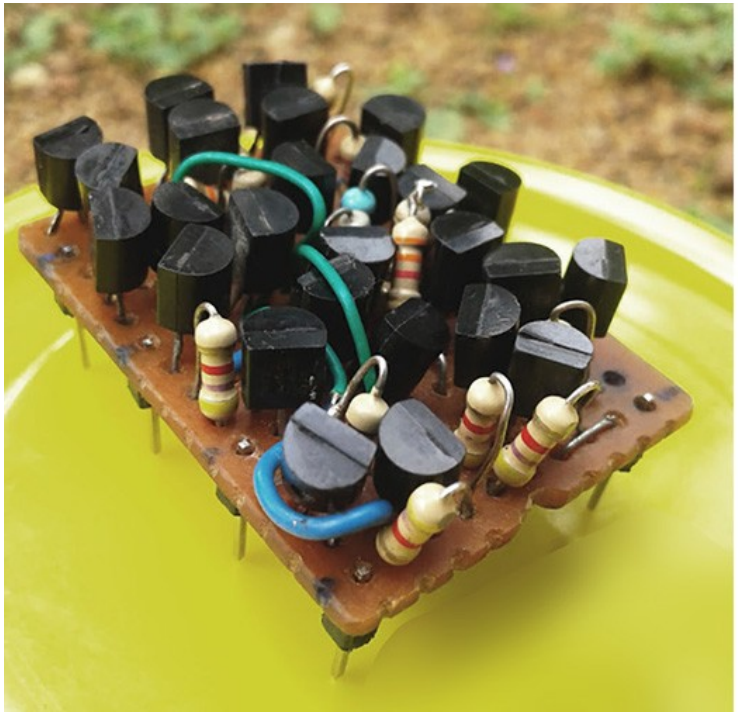


But most electronics today is tiny (often virtually impossible to repair) circuits made by robots.

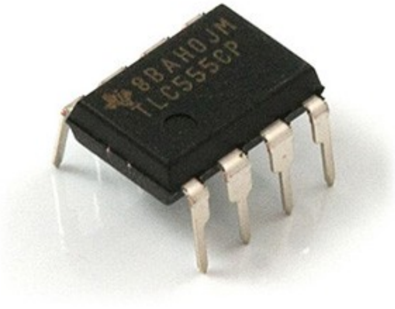




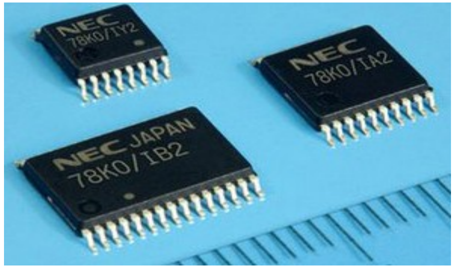
The process of making “integrated circuits” (a.k.a. “IC”s or just “chips”) has gotten easier and cheaper. These are now replacing circuits made of through-hole discrete components.

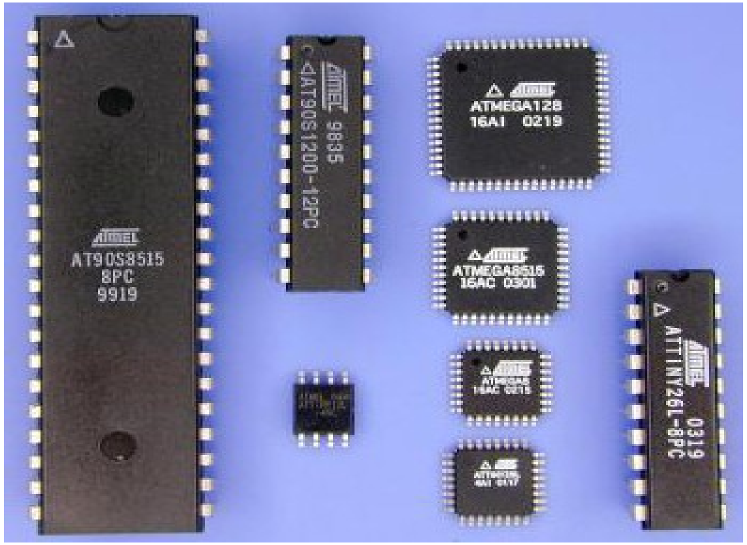


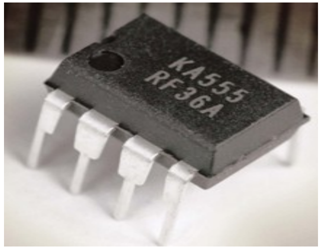
=

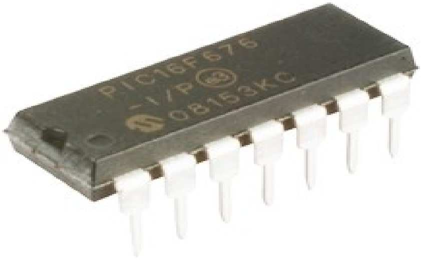


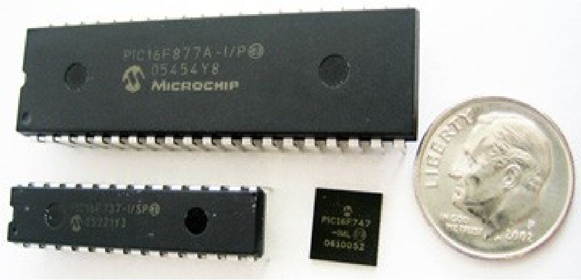
There are many different *microcontroller* ICs that are programmable.



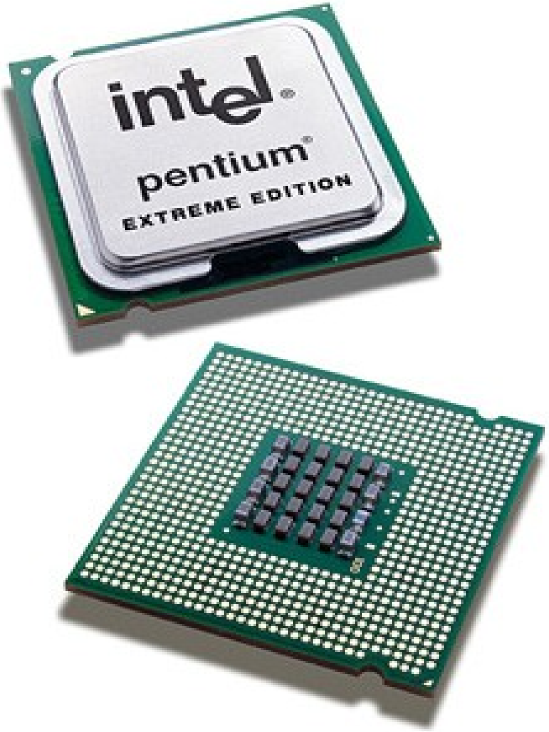


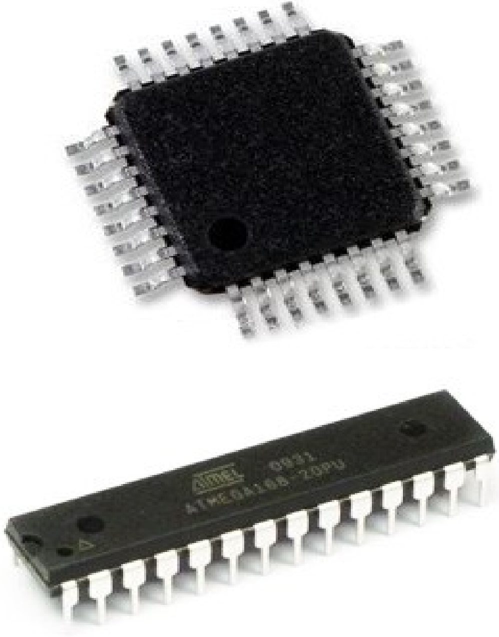






What's the difference between a microprocessor and a microcontroller?





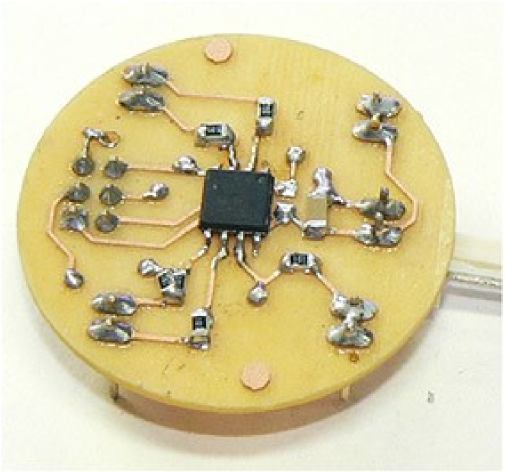
A *microprocessor:*

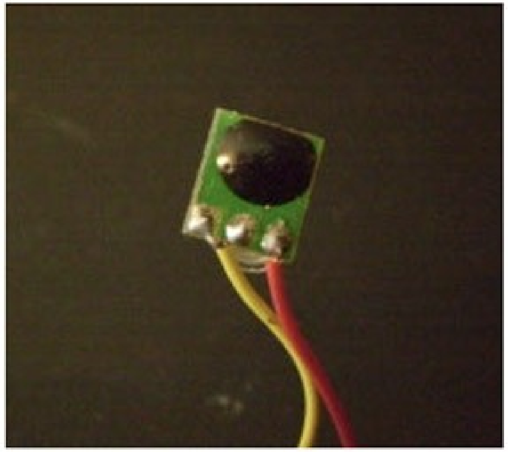
- an IC with only the Central Processing Unit (CPU)
- No RAM, ROM, or peripheral I/O on the chip.
  (System designers must add these externally to make these function in Desktop PC's, Laptops, notepads, tablets, etc.
- (Manufacturers include:
  Intel's Pentium, core 2 duo, i5, i7, ARM, PowerPC, AMD, etc.)

A *microcontroller:*

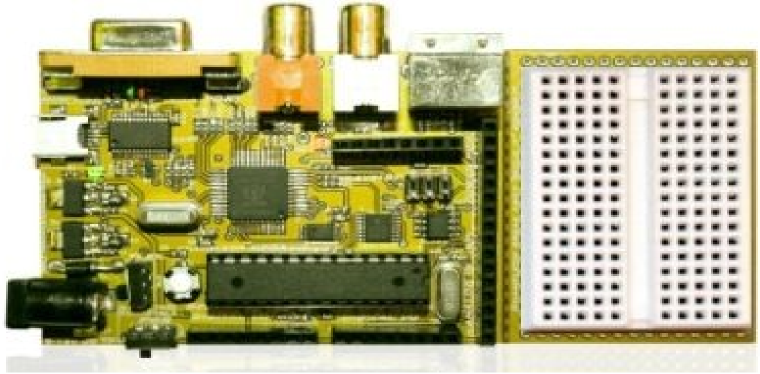
- an IC with CPU, I/O pins, a fixed amount of RAM, ROM, all embedded on a single, 'all in one' chip.
- (Manufacturers include:
  Microchip, ATMEL, TI, Freescale, Philips, Motorola)

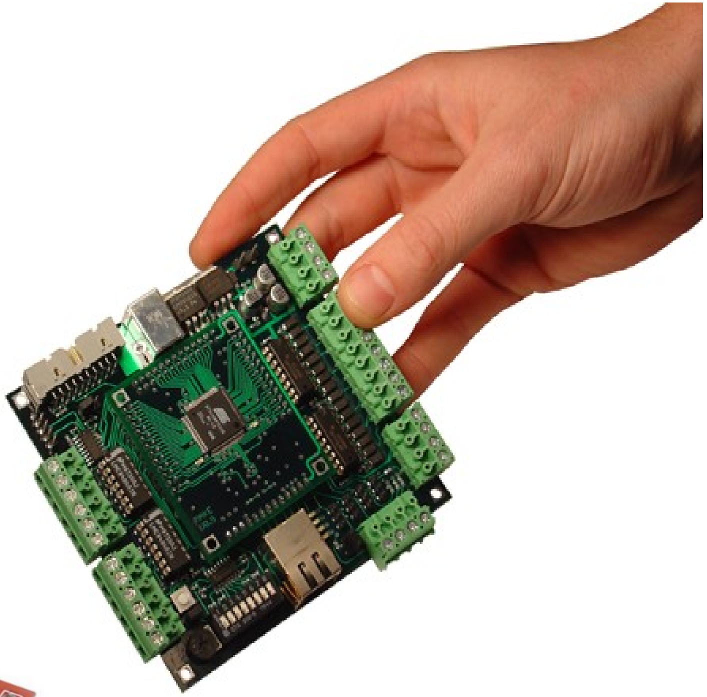
LED tea candles are running “flicker code” running on very small microcontroller chips.
(smaller than the ones pictured, actually *inside* the LED package.)

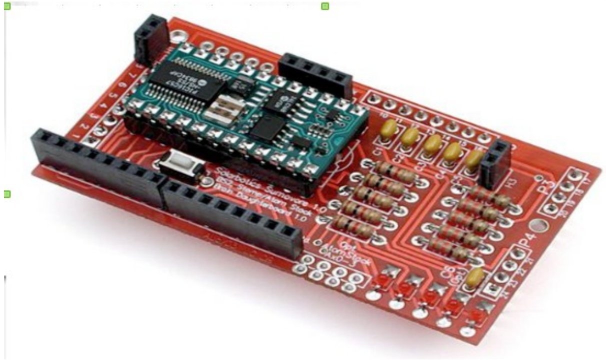




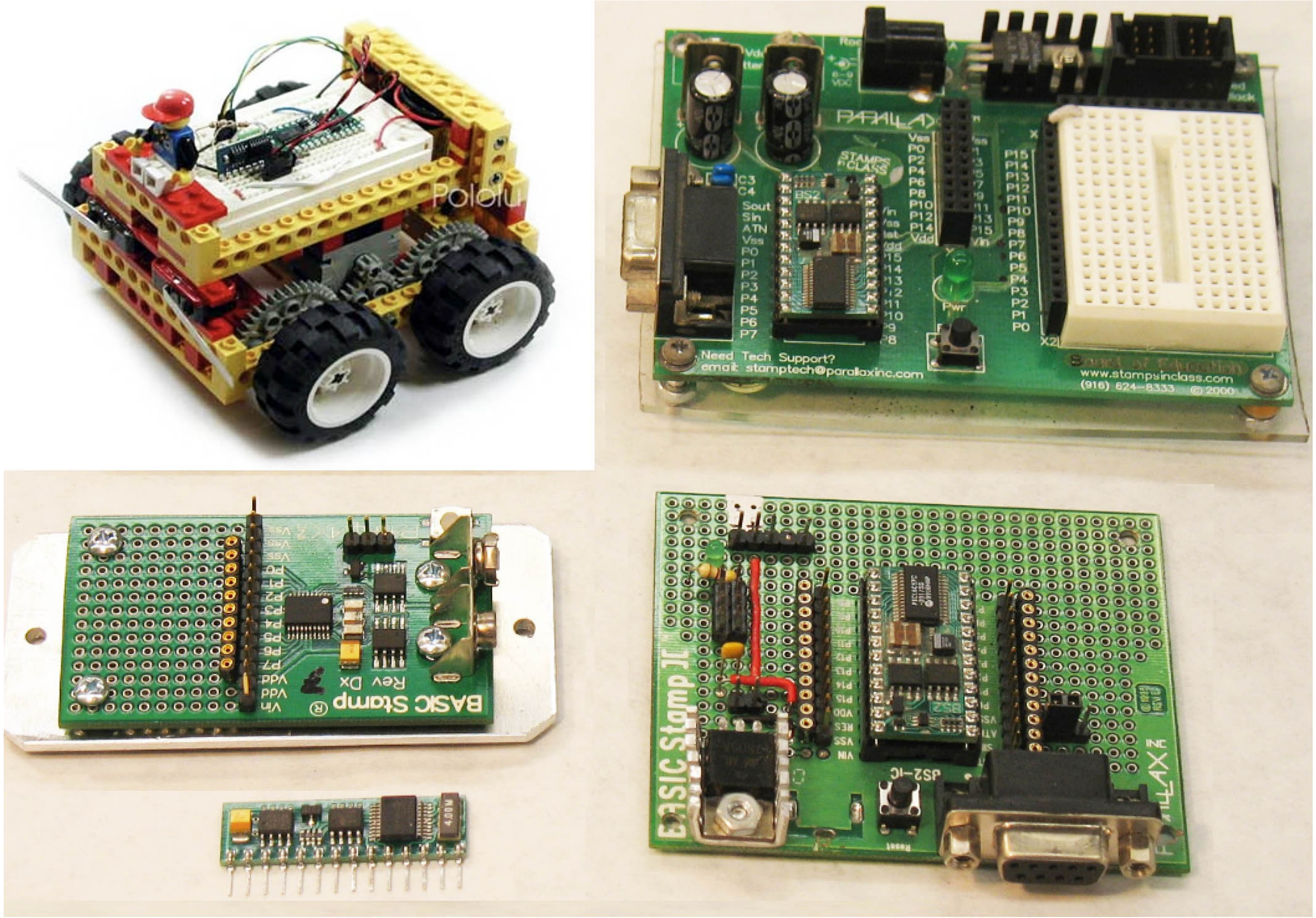
Many microcontrollers are available on handy “development boards”.



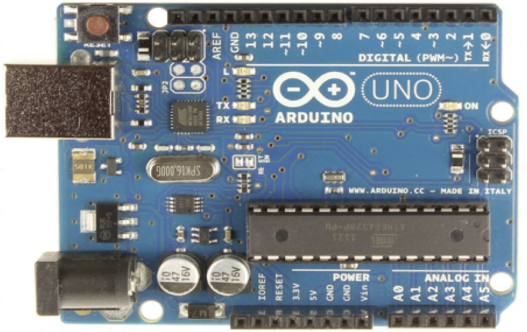


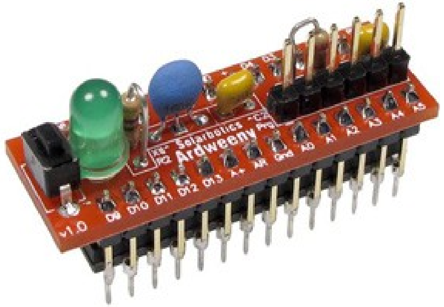


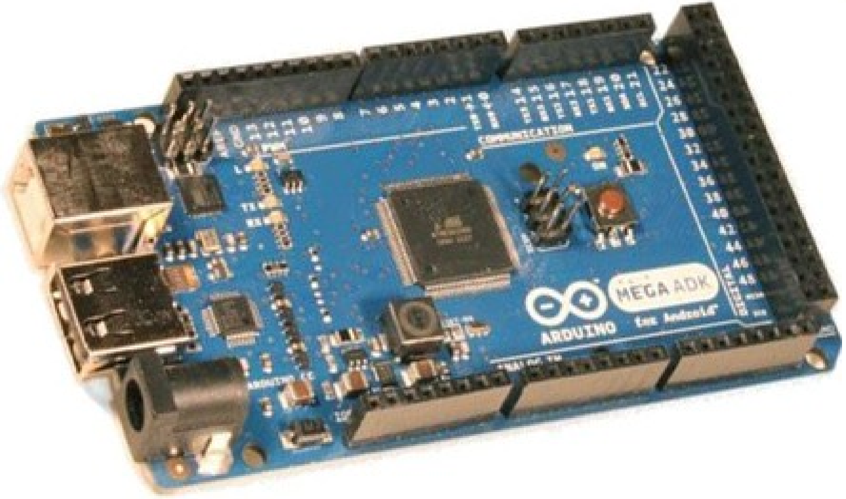
Parallax “Basic Stamp” (Lego Mindstorms) :

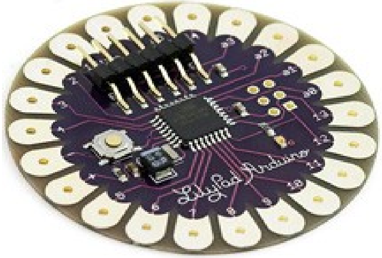


Arduino boards...



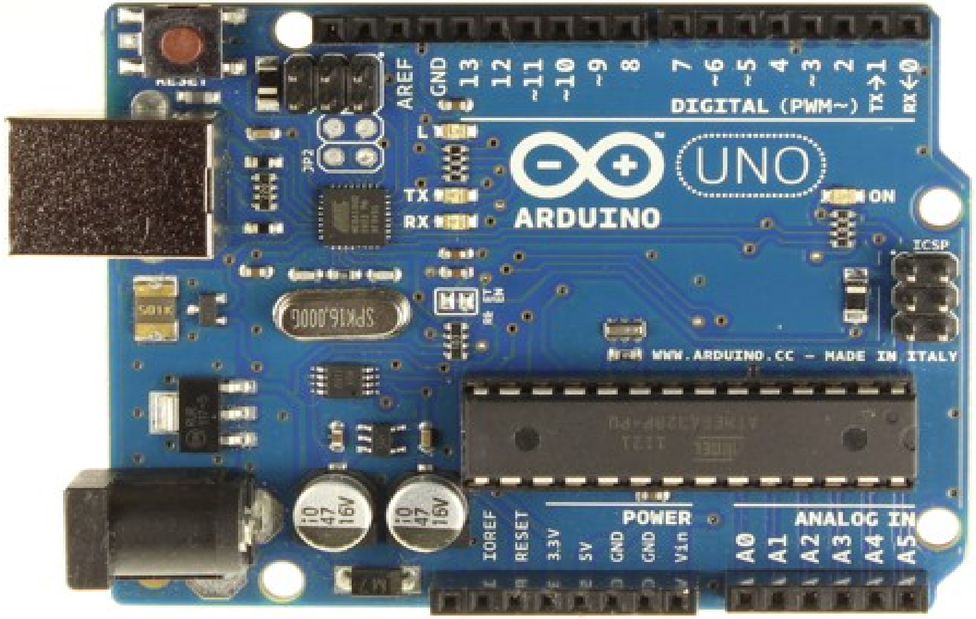






Leah Beuchley

The Arduino



The Arduino was released in 2005, by 2007 10,000 were sold. By 2021 that had risen to 10 million.

Began as: “Wiring”, a Graduate thesis project by Colombian engineering student, Hernando Barragán to make electronic programming development easy and open source.

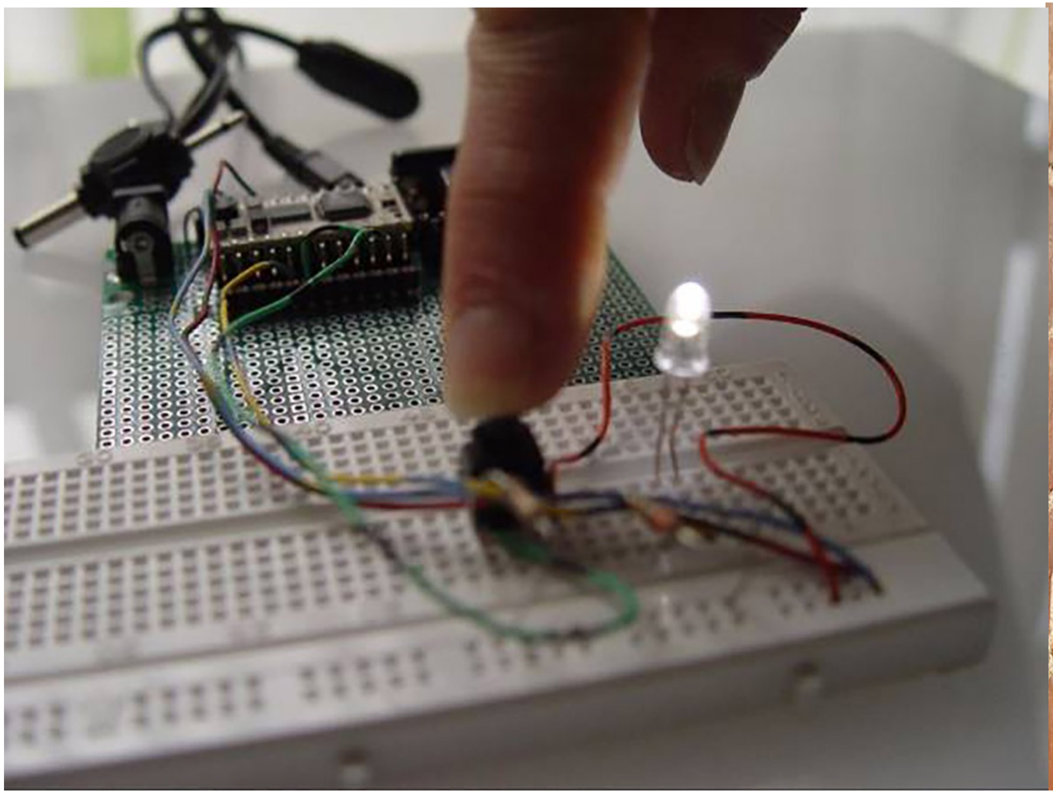


Photo: Randi Silberman Klett

Five of the core Arduino developer team:
David Cuartielles, Gianluca Martino, Tom Igoe, David Mellis and Massimo Banzi. (Barragán was excluded for some reason.)

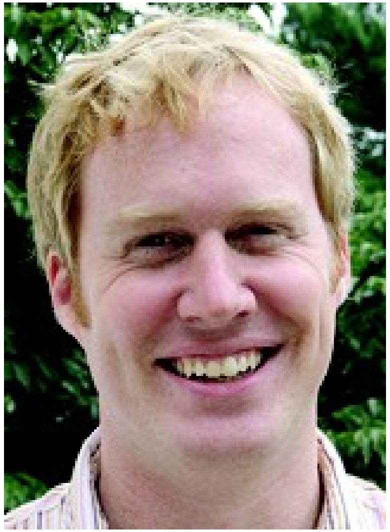
Arduino's kind of revolutionary potential:

“ If we take a piece of technology considered, rightly or wrongly, complex, and we make it simple by working not on the technology itself but on the experience that comes from using it, then everything gets simpler and we discover new possibilities that were not there before.

“If I were a chicken farmer, I could think of an idea to increase the efficiency or the safety of my farm. With Arduino, even those -who know about chickens but not about electronics and programming- are able to achieve, by themselves and in a very short time, the kludge they need to solve their practical problem.
All this is about moving technology towards what we might call a domain expert. In the end, the one who raises chickens is certainly much more expert about chickens than an engineer.”

- Massimo Banzi

processing.org

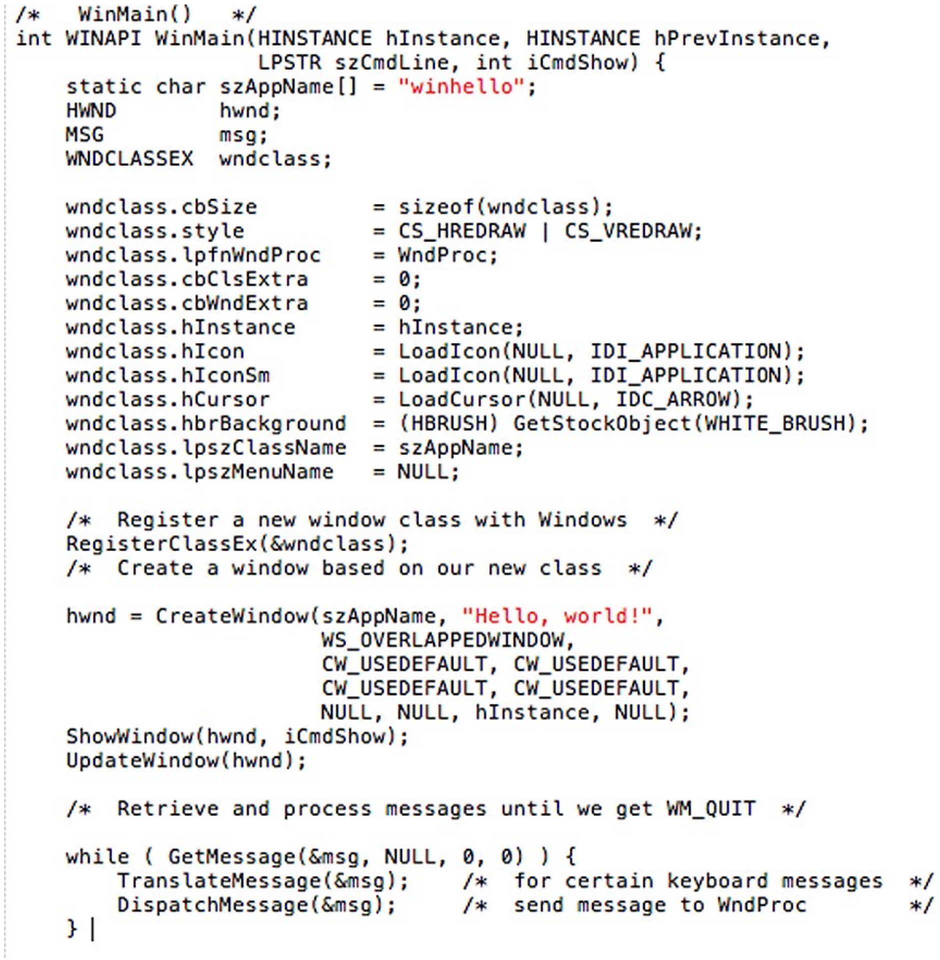


Project of Ben Fry and Casey Reas

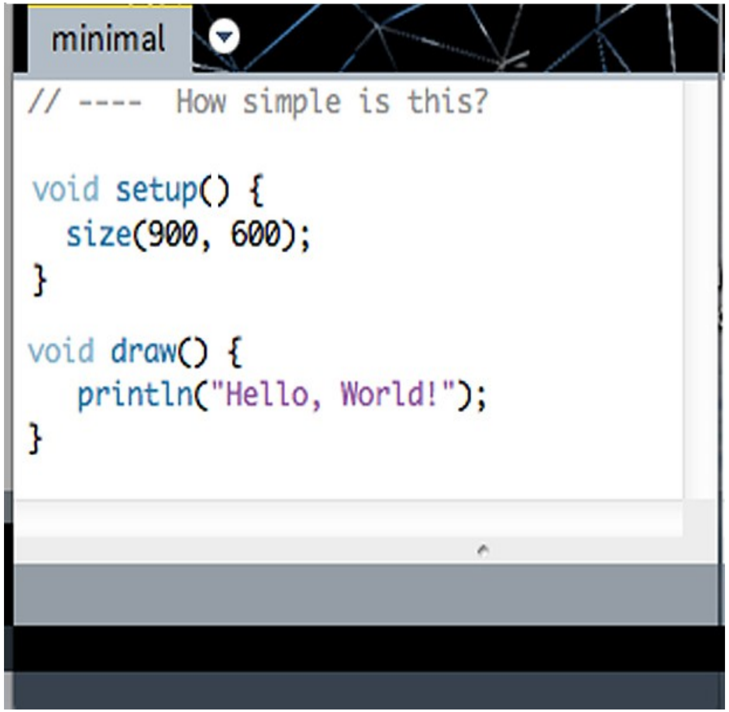
We stand by our mission statement:

“ Processing seeks to ruin the careers of talented designers by tempting them away from their usual tools and into the world of programming and computation.

Similarly, the project is designed to turn engineers and computer scientists to less gainful employment as artists and designers.”

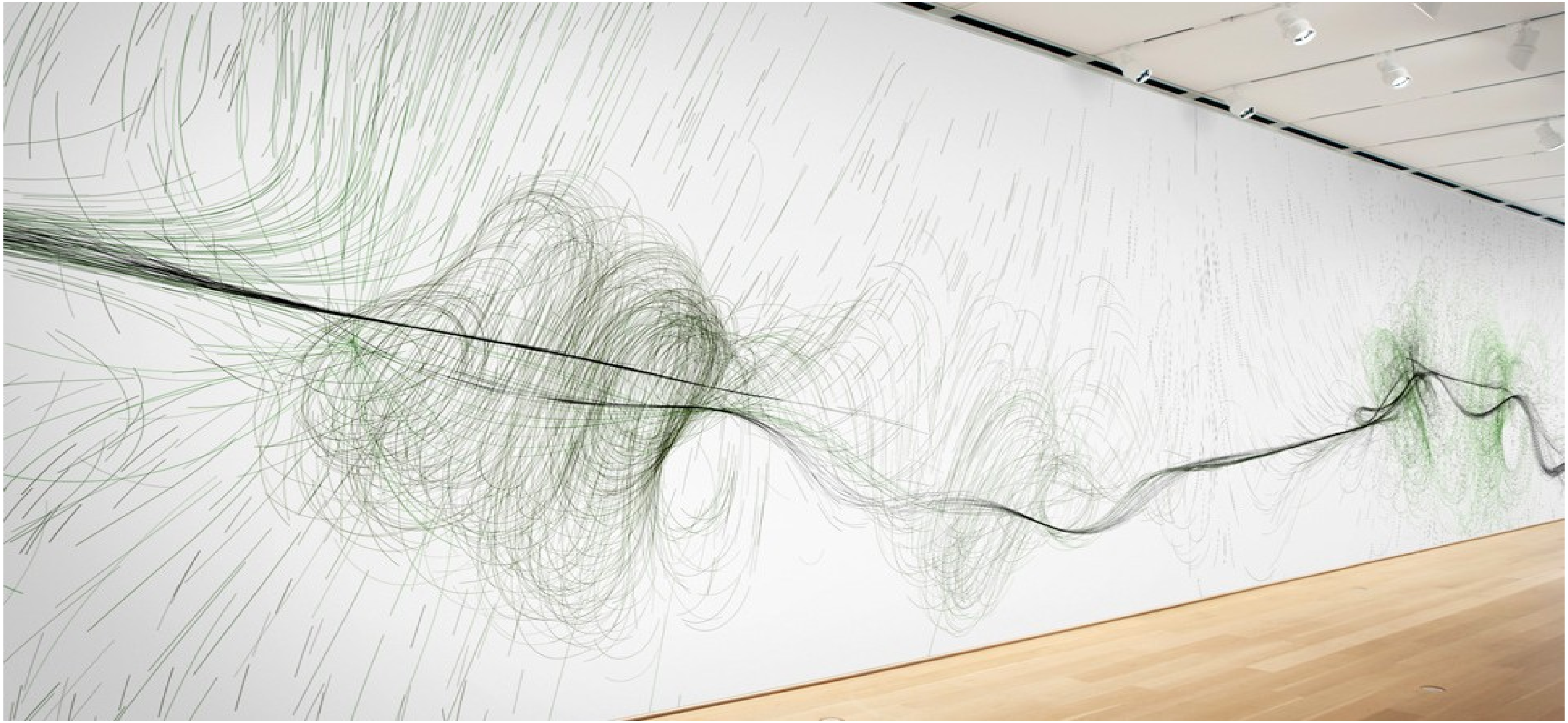


```
/*   WinMain()   */
int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance,
                   LPSTR szCmdLine, int iCmdShow) {
    static char szAppName[] = "winhello";
    HWND        hwnd;
    MSG         msg;
    WNDCLASSEX  wndclass;

    wndclass.cbSize         = sizeof(wndclass);
    wndclass.style          = CS_HREDRAW | CS_VREDRAW;
    wndclass.lpfnWndProc    = WndProc;
    wndclass.cbClsExtra     = 0;
    wndclass.cbWndExtra     = 0;
    wndclass.hInstance      = hInstance;
    wndclass.hIcon          = LoadIcon(NULL, IDI_APPLICATION);
    wndclass.hIconSm        = LoadIcon(NULL, IDI_APPLICATION);
    wndclass.hCursor        = LoadCursor(NULL, IDC_ARROW);
    wndclass.hbrBackground  = (HBRUSH) GetStockObject(WHITE_BRUSH);
    wndclass.lpszClassName  = szAppName;
    wndclass.lpszMenuName   = NULL;

    /*  Register a new window class with Windows  */
    RegisterClassEx(&wndclass);
    /*  Create a window based on our new class  */

    hwnd = CreateWindow(szAppName, "Hello, world!",
                        WS_OVERLAPPEDWINDOW,
                        CW_USEDEFAULT, CW_USEDEFAULT,
                        CW_USEDEFAULT, CW_USEDEFAULT,
                        NULL, NULL, hInstance, NULL);
    ShowWindow(hwnd, iCmdShow);
    UpdateWindow(hwnd);

    /*  Retrieve and process messages until we get WM_QUIT  */

    while ( GetMessage(&msg, NULL, 0, 0) ) {
        TranslateMessage(&msg);    /*  for certain keyboard messages  */
        DispatchMessage(&msg);     /*  send message to WndProc        */
    } |
```



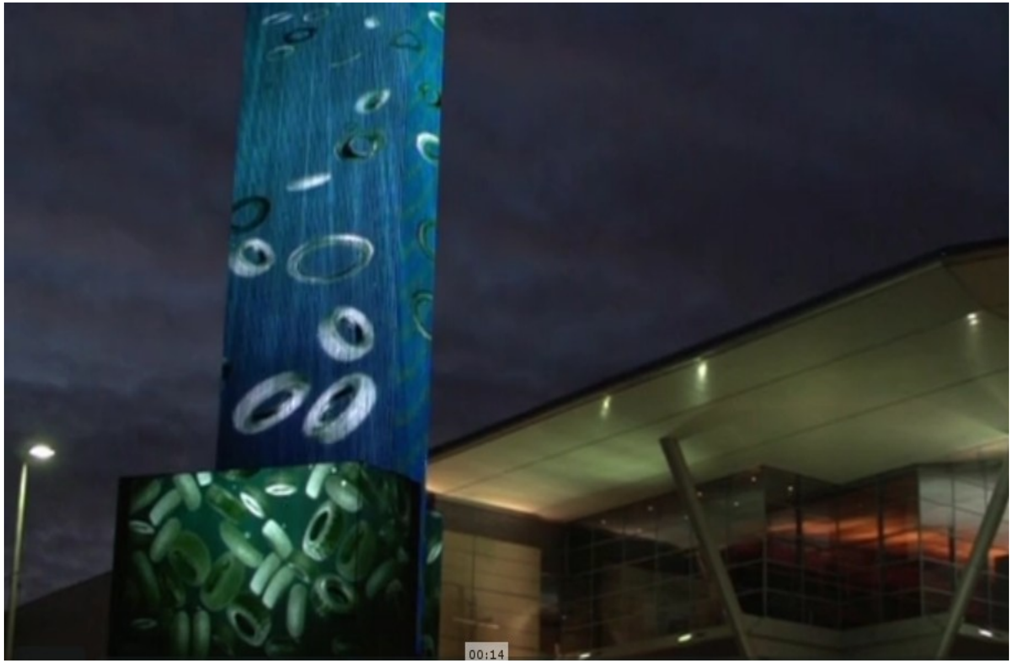
```
// ----  How simple is this?

void setup() {
  size(900, 600);
}

void draw() {
   println("Hello, World!");
}
```

By changing the relationship with code and coding...



Casey Reas: *Installation at the Art Institute of Chicago*

House of Cards - *Radiohead*
directed by Aaron Koblin



“Dumping” -Lina Maria Giraldo

Some Heroes in the Creative Free Hard/Software movement:

Limor Fried *Photo: Inc com*

– Limor Fried (Lady Ada) AdaFruit Industries

– Hernando Barragán: “Wiring” Arduino Libraries

– Karsten Schmidt “Toxi” Processing libs

– Daniel Shiffman -”Physical Computing” with Tom Igoe @ NYU's ITP

– Leah Beuchley -designed “Lilypad” Arduino

– Dr. Rafael Hernandez pd tutorials “cheetomoskeeto” YouTube channel

What is Open Source Hardware?

- License
- Schematics
- PCB layout data
- Bills of Sale (component distributors...)

Arduino is **open source hardware**: the Arduino hardware reference designs are distributed under a Creative Commons Attribution Share-Alike 2.5 license, available on the Arduino Web site.

Layout and production files for some versions are also available.

The source code for the IDE is available and released under the GNU General Public License, version 2.

The *Arduino* is a development platform using microcontrollers from Atmel's ATmega series.

The UNO for example, uses the ATmega 328:

- 28 pin, 8 bit IC
- Runs at 16 MHz (but chip can run at 20 MHz).
- 32K Bytes of flash memory, 2K SRAM
- 20 I/O pins: 14 digital, 6 analog (10 bit ADC)
- regulated at 5 and 3.3 volts

Requirements for the microcontroller:

1. Power

- It's an electrical component, so of course you have to give it power. But like many ICs, the voltage used to operate it needs to be controlled relatively precisely.

2. I/O

- Input and output, some way to communicate with the chip. This is generally done through some kind of connection to the chip's pins. Breadboards are handy.

3. Programming Interface

- Some way to write programs and download them to the chip and run them.

Requirements for the microcontroller:

1. **Power**:

There are four commonly used ways for controlling the voltage supplied to microcontrollers:

I. A voltage regulator

II. A regulated power supply

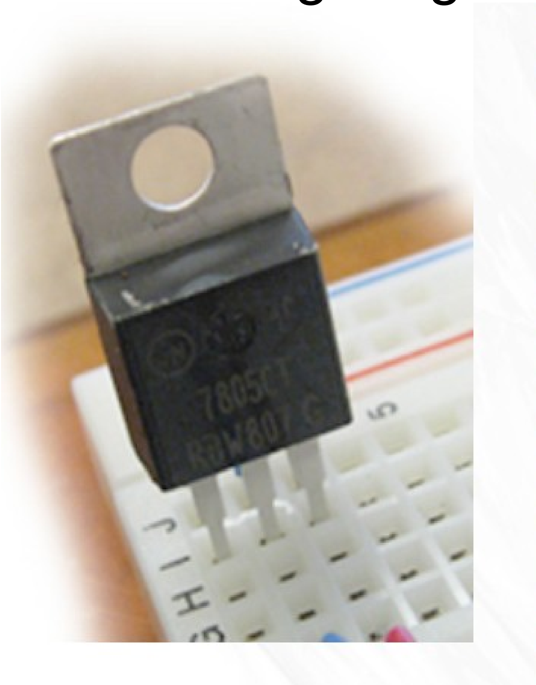
III. Battery power

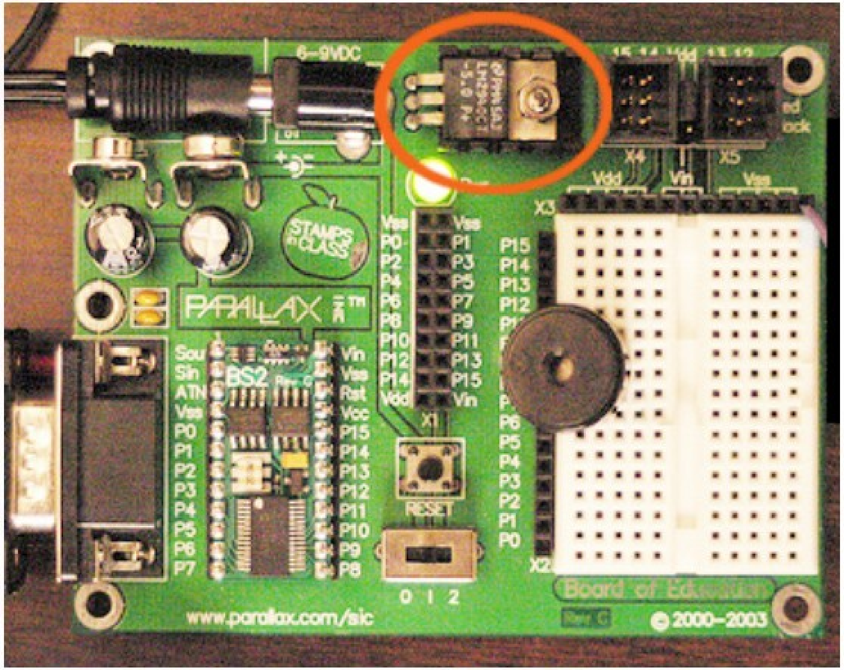
IV. from USB

Requirements for the microcontroller:

5 Volts (exactly):

Voltage regulator- external or on-board





Requirements for the microcontroller:

5 Volts (exactly):

Regulated power supply

Requirements for the microcontroller:

5 Volts (exactly):

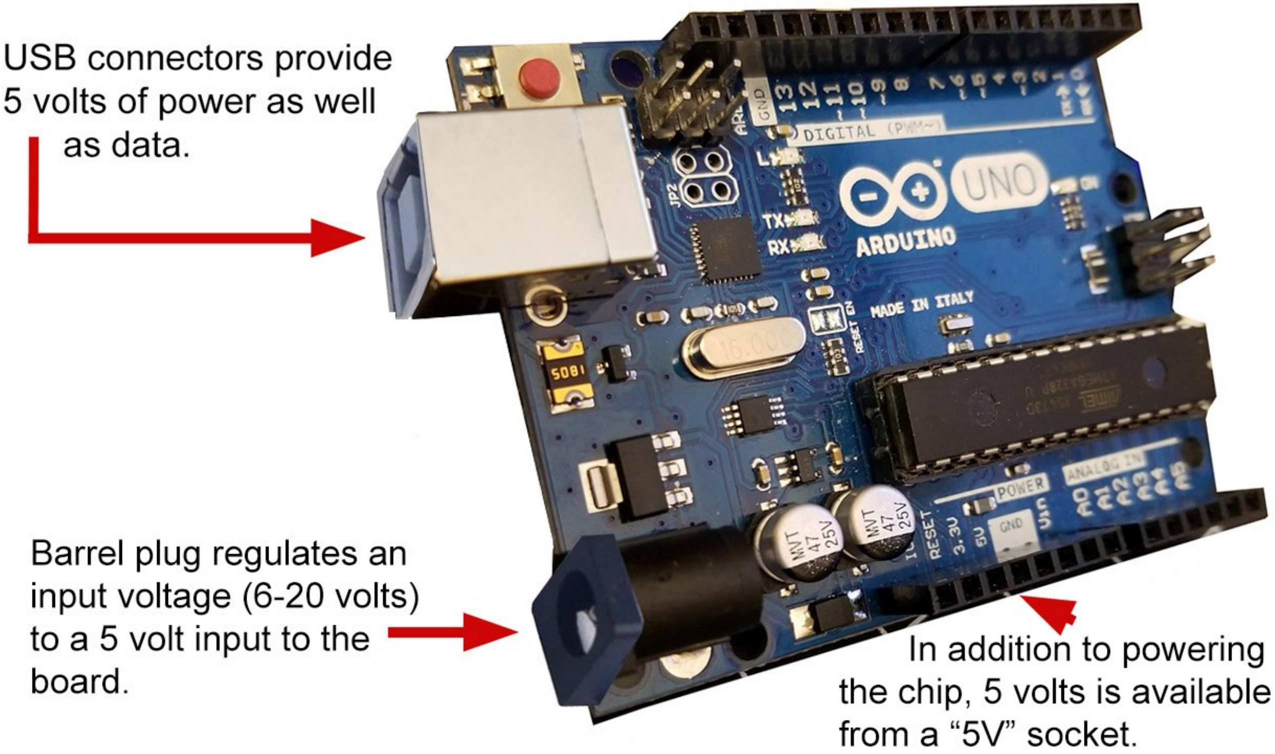
Battery pack

(Various adapters can convert from 6V or boost 3V up to 5 volts.)

Requirements for the microcontroller:

5 Volts (exactly):

from USB



USB connectors provide 5 volts of power as well as data.

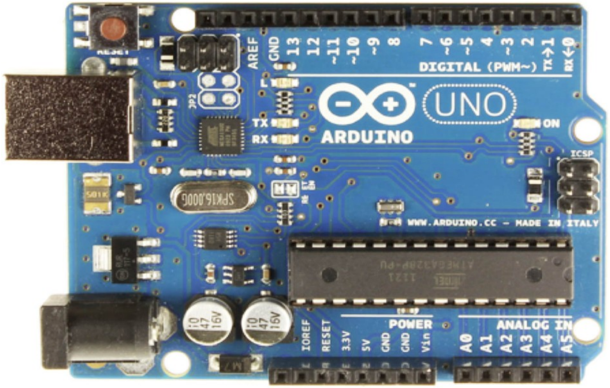
Barrel plug regulates an input voltage (6-20 volts) to a 5 volt input to the board.

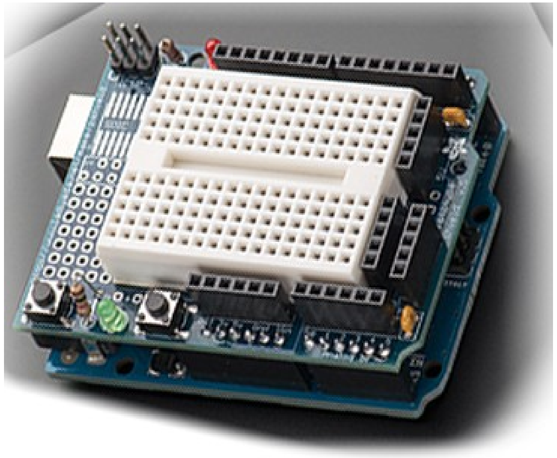
In addition to powering the chip, 5 volts is available from a “5V” socket.

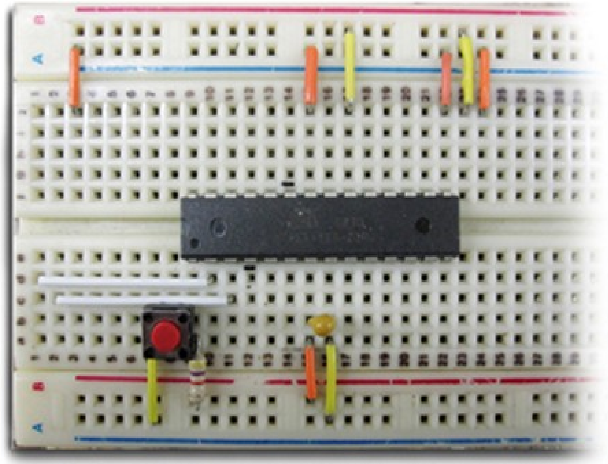
Requirements for the microcontroller:

2: I/O (Input and Output):

I. Development Board
II. “Shields”
III. Roll your own

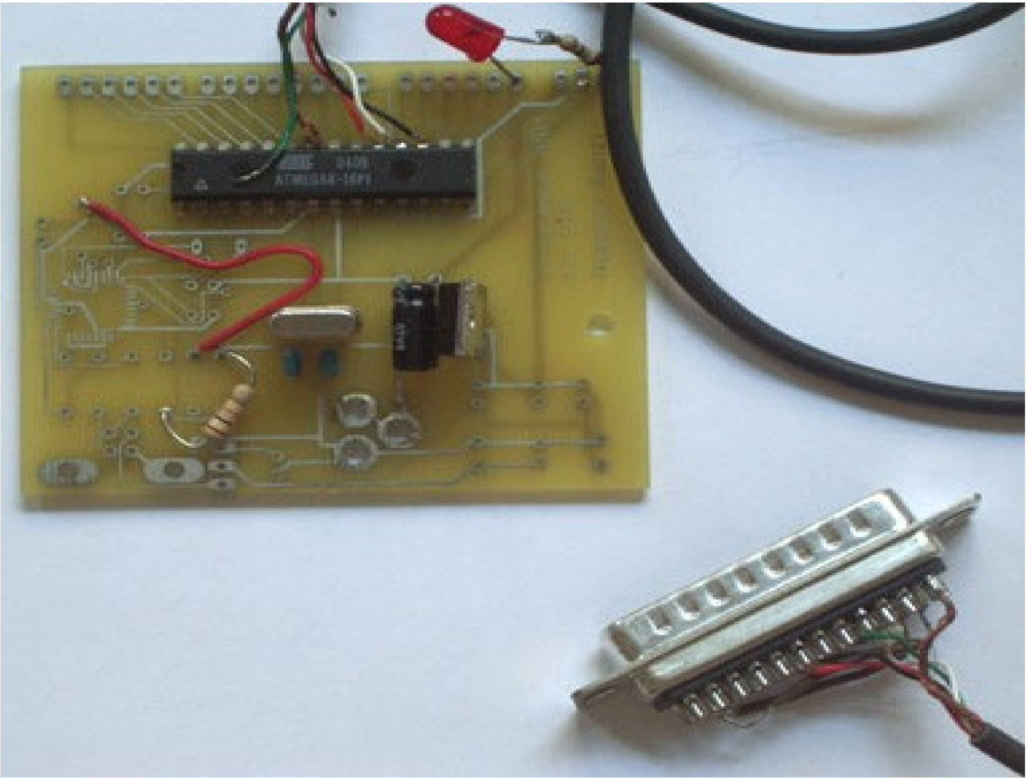


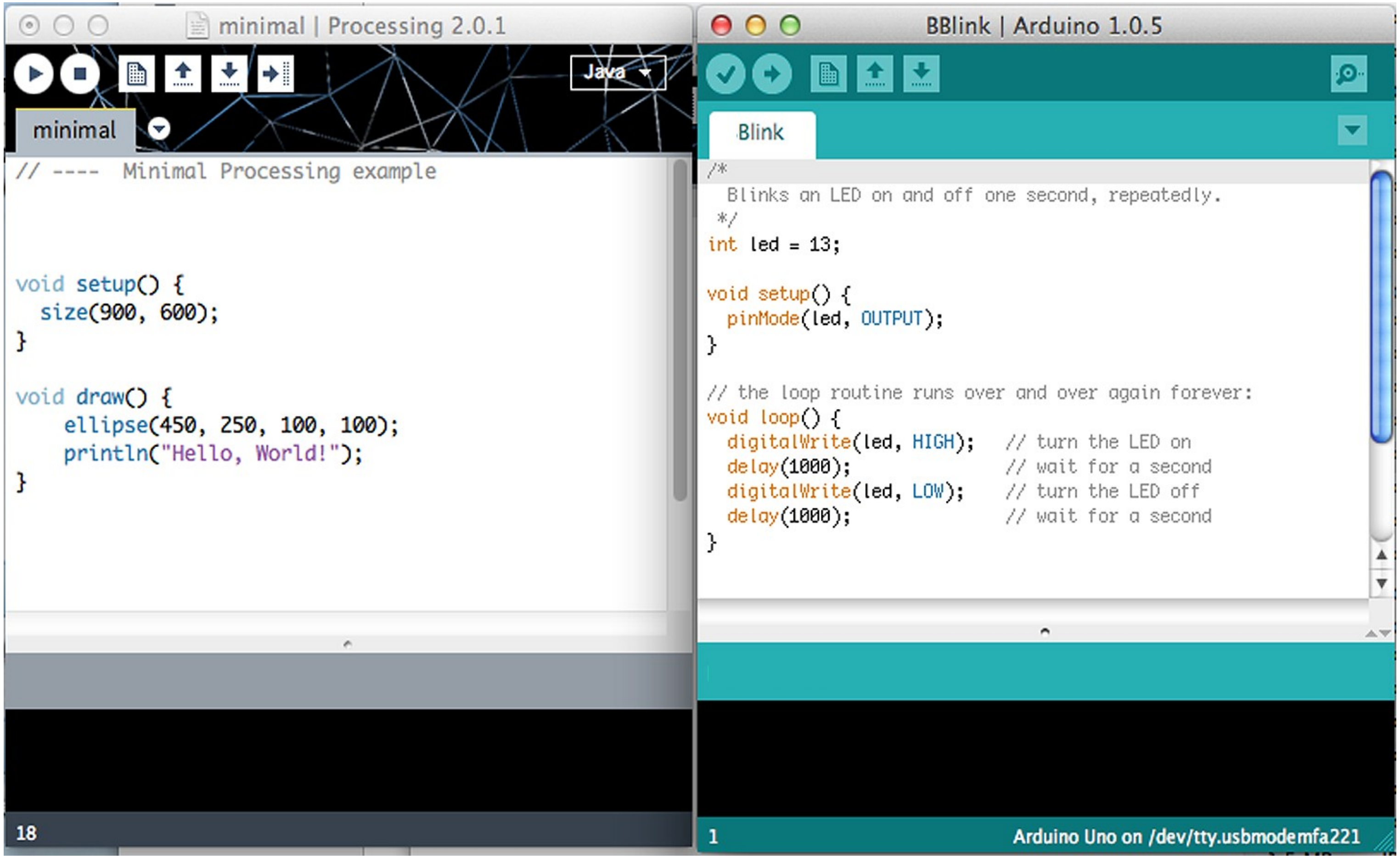




Requirements for the microcontroller:

3. Programming Interface:





minimal | Processing 2.0.1
Java
minimal
18
BBlink | Arduino 1.0.5
Blink
1
Arduino Uno on /dev/tty.usbmodemfa221

```
// ----  Minimal Processing example



void setup() {
  size(900, 600);
}

void draw() {
    ellipse(450, 250, 100, 100);
    println("Hello, World!");
}
```

```
/*
  Blinks an LED on and off one second, repeatedly.
 */
int led = 13;

void setup() {
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);   // turn the LED on
  delay(1000);               // wait for a second
  digitalWrite(led, LOW);    // turn the LED off
  delay(1000);               // wait for a second
}
```

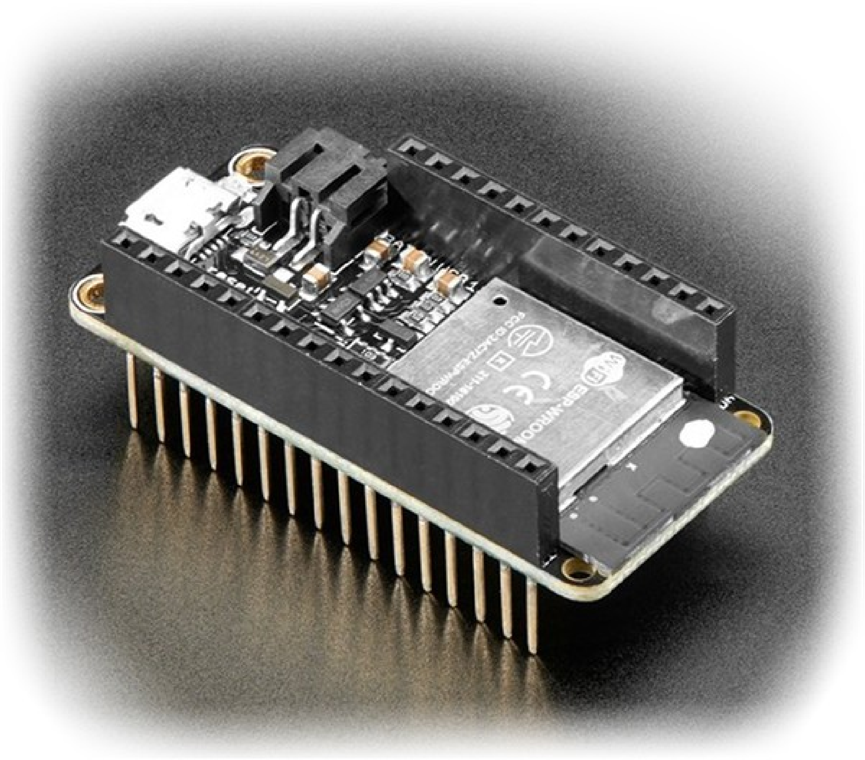
The similarity is not accidental.

Fortunately, popularity invites copying and competition...

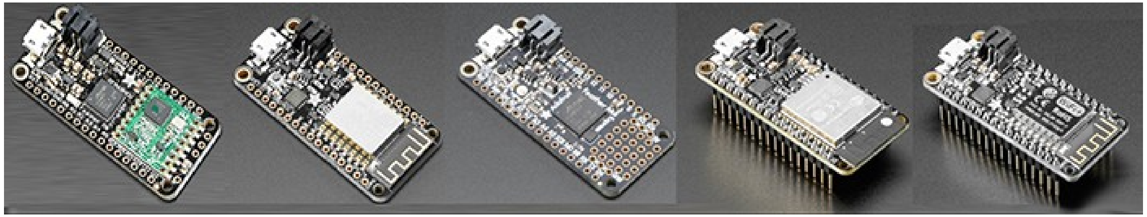
...on the smaller end:
Adafruit Feather

(https://adafruit.com):

~ $20. various



- specialized function/types:
- WiFi
- Bluetooth
- I/O for motor control
- Python-programmable



...on the higher end:
Odroid-X

$70 – $180.00

(or Sheevaplug, PandaBoard...)

- Quad core ARM processor
- 1 Gig DDR2 RAM
- 50 pin IO pin I/O
- SD card storage
- Runs Android OS

...on the higher end: Mac mini

$700.00

- 8 core processor
- No I/O, only USB

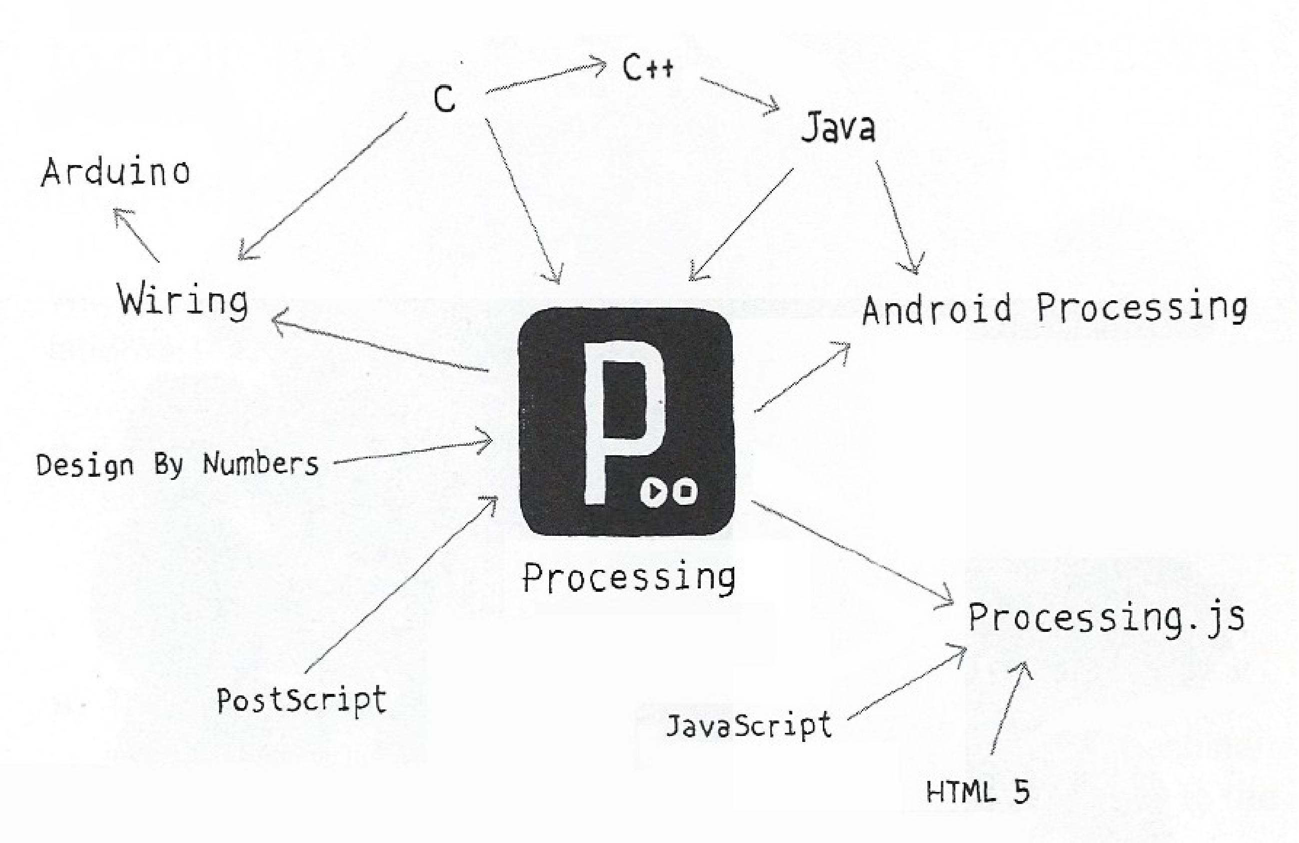
The Raspberry Pi:

$10 - 35

- 64-bit quad core 1.5GHz ARM processor
- 1-2 Gig DDR2 RAM
- 40 pin I/O header

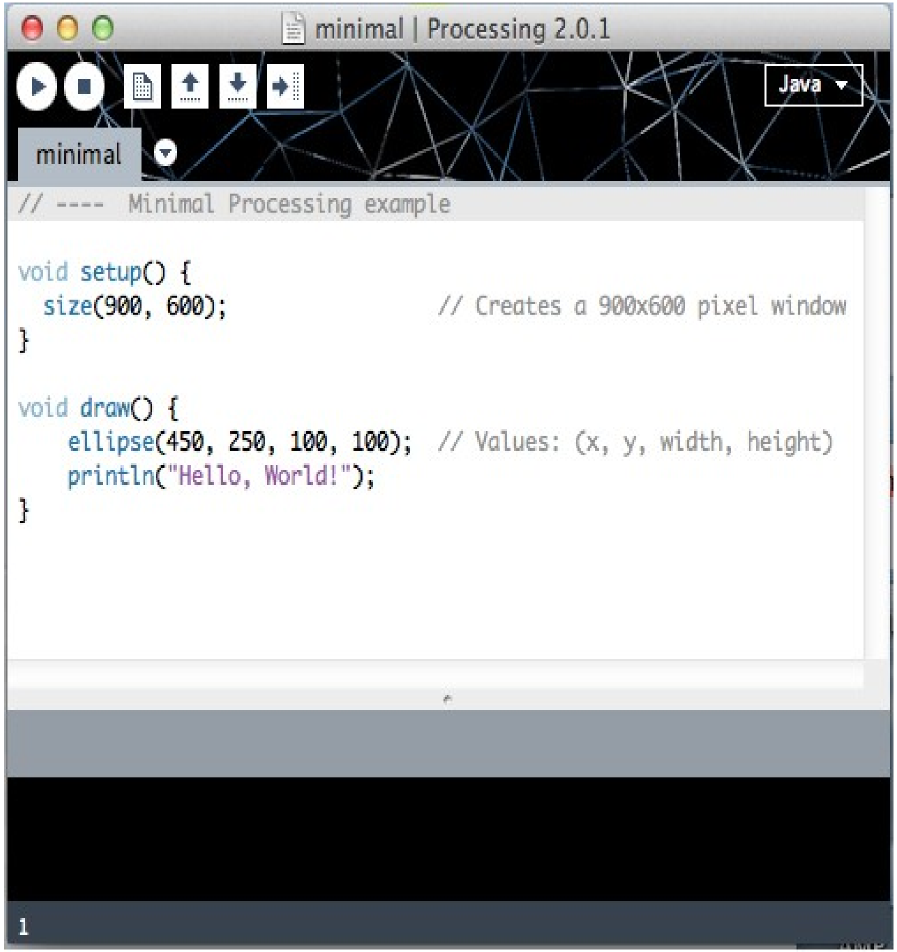
Flickr.com/Poptech/CC BY SA 2.0

Eben Upton, Executive Director, Paspberry Pi Foundation

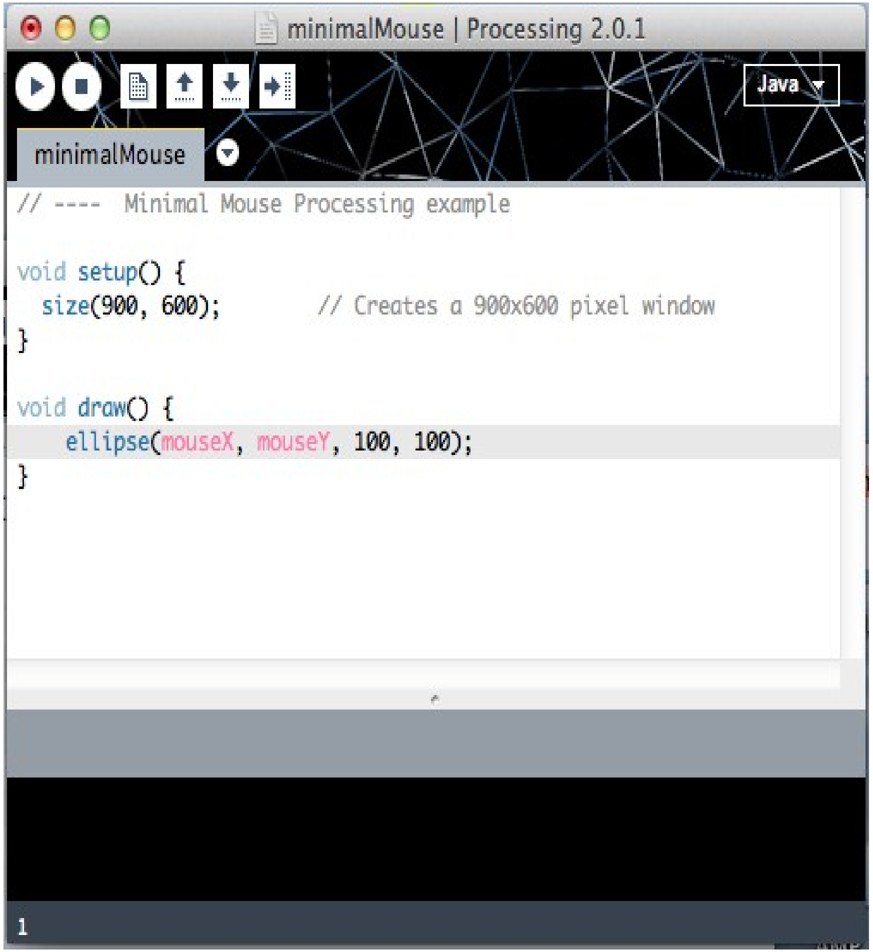


Processing comes from a large family of related languages.

(Graphic from *Getting Started with Processing* by Casey Reas and Ben Fry.)

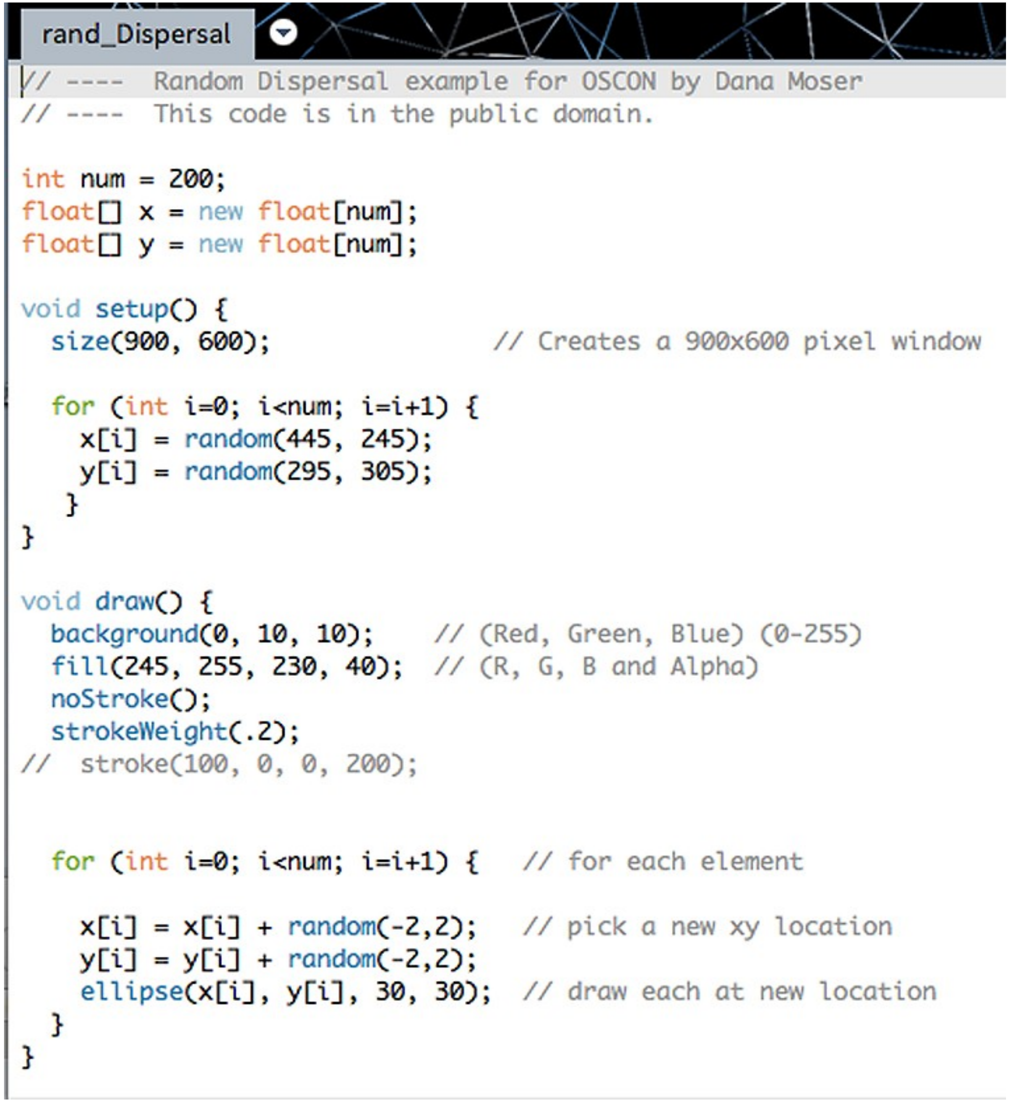


```
// ----  Minimal Processing example

void setup() {
  size(900, 600);                  // Creates a 900x600 pixel window
}

void draw() {
    ellipse(450, 250, 100, 100);  // Values: (x, y, width, height)
    println("Hello, World!");
}
```



```
// ----  Minimal Mouse Processing example

void setup() {
  size(900, 600);        // Creates a 900x600 pixel window
}

void draw() {
    ellipse(mouseX, mouseY, 100, 100);
}
```

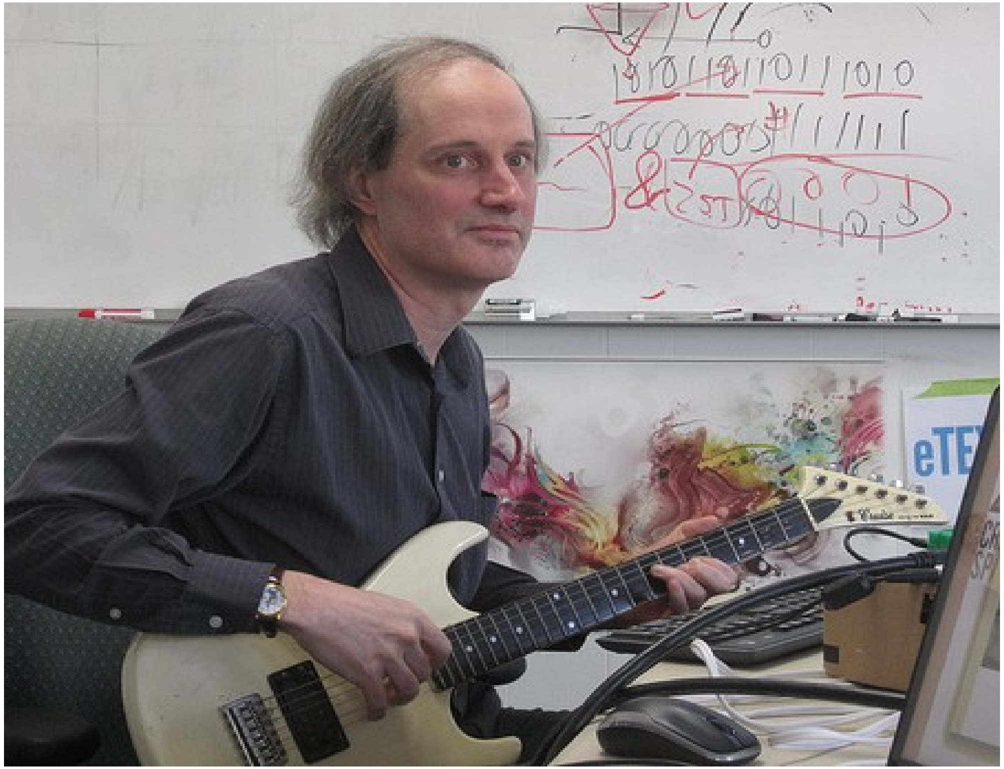
And go a little further...



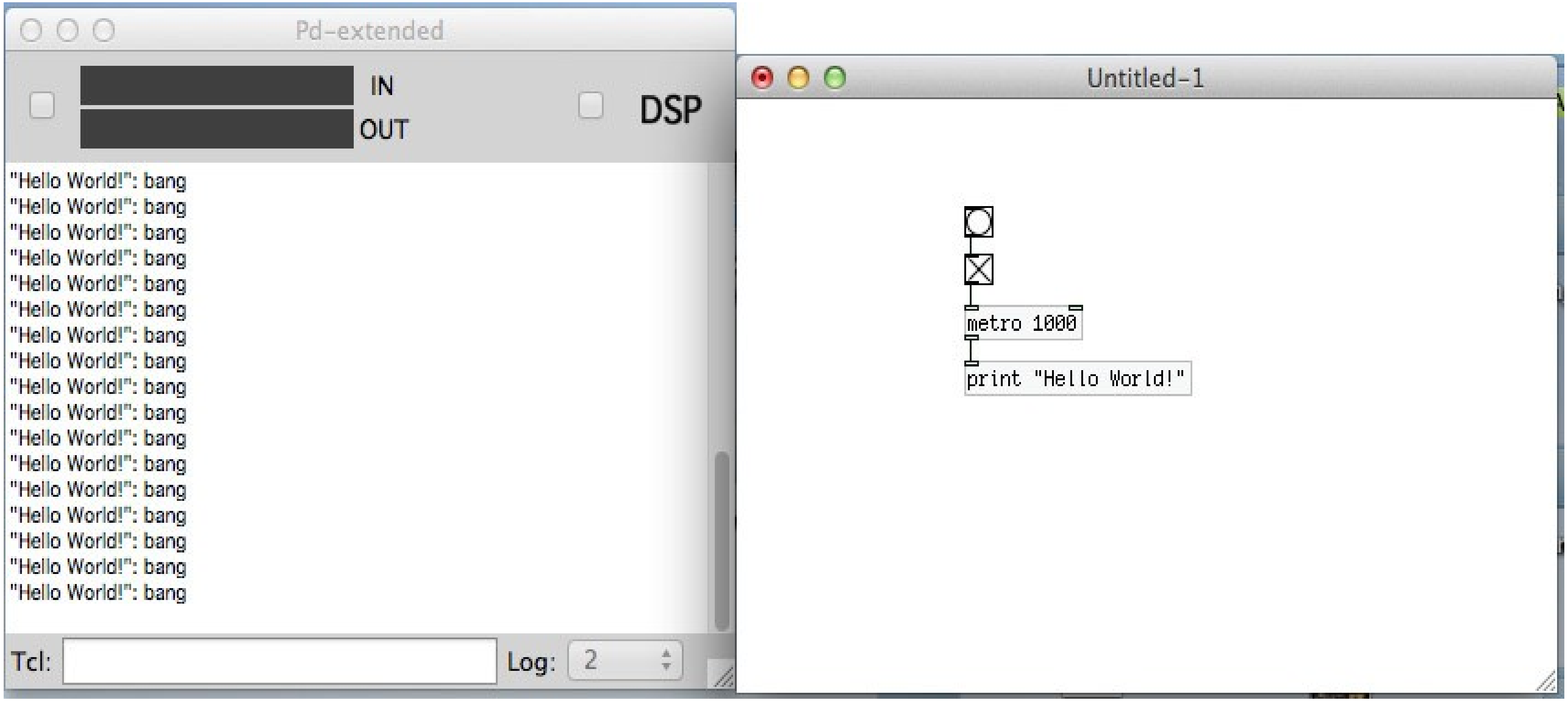
```
// ----  Random Dispersal example for OSCON by Dana Moser
// ----  This code is in the public domain.

int num = 200;
float[] x = new float[num];
float[] y = new float[num];

void setup() {
  size(900, 600);               // Creates a 900x600 pixel window

  for (int i=0; i<num; i=i+1) {
    x[i] = random(445, 245);
    y[i] = random(295, 305);
   }
}

void draw() {
  background(0, 10, 10);    // (Red, Green, Blue) (0-255)
  fill(245, 255, 230, 40);  // (R, G, B and Alpha)
  noStroke();
  strokeWeight(.2);
//  stroke(100, 0, 0, 200);


  for (int i=0; i<num; i=i+1) {   // for each element

    x[i] = x[i] + random(-2,2);   // pick a new xy location
    y[i] = y[i] + random(-2,2);
    ellipse(x[i], y[i], 30, 30);  // draw each at new location
  }
}
```

Other Free Software projects:

PureData (pd)



Miller Puckette



```
"Hello World!": bang
"Hello World!": bang
"Hello World!": bang
"Hello World!": bang
"Hello World!": bang
"Hello World!": bang
"Hello World!": bang
"Hello World!": bang
"Hello World!": bang
"Hello World!": bang
"Hello World!": bang
"Hello World!": bang
"Hello World!": bang
"Hello World!": bang
"Hello World!": bang
"Hello World!": bang
"Hello World!": bang
```

Log: 2

```
metro 1000
print "Hello World!"
```

pd's “flow control” programming interface.