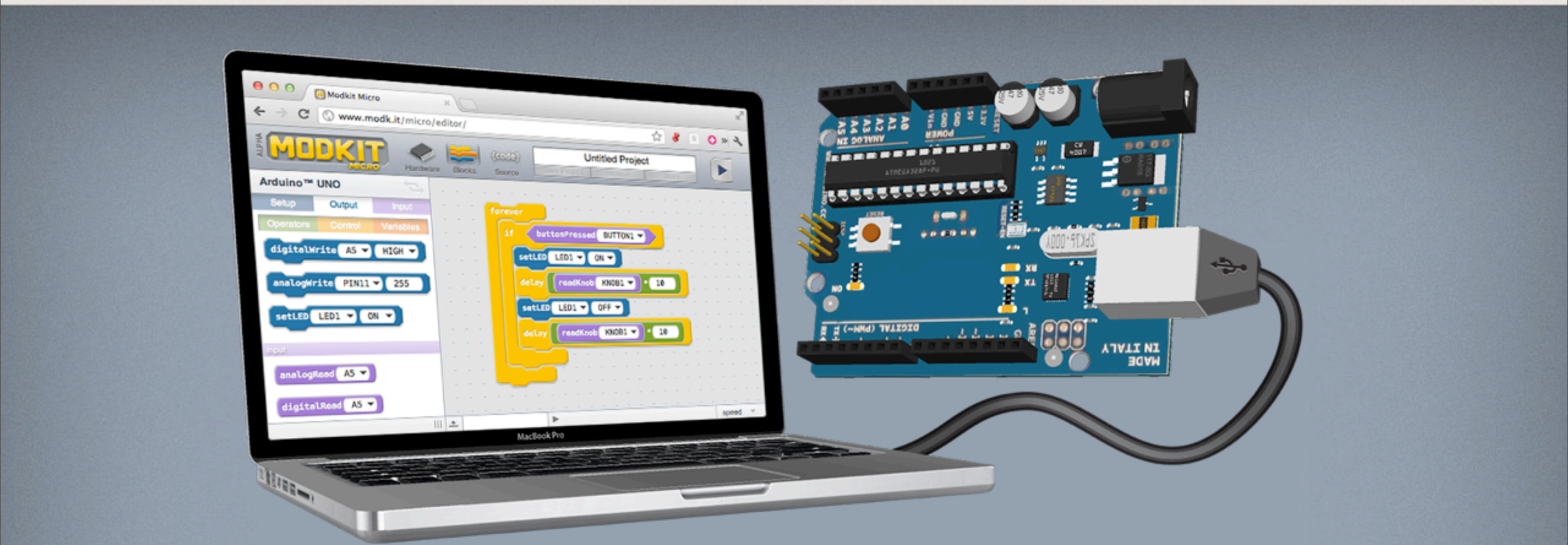
Wiring++™

The future of microcontroller programming

ed@modk.it

About Modkit



Programming & engineering *should* be:
Accessible, Affordable, Engaging, and Fun!

the beginning...

The Wiring Project

the beginning...

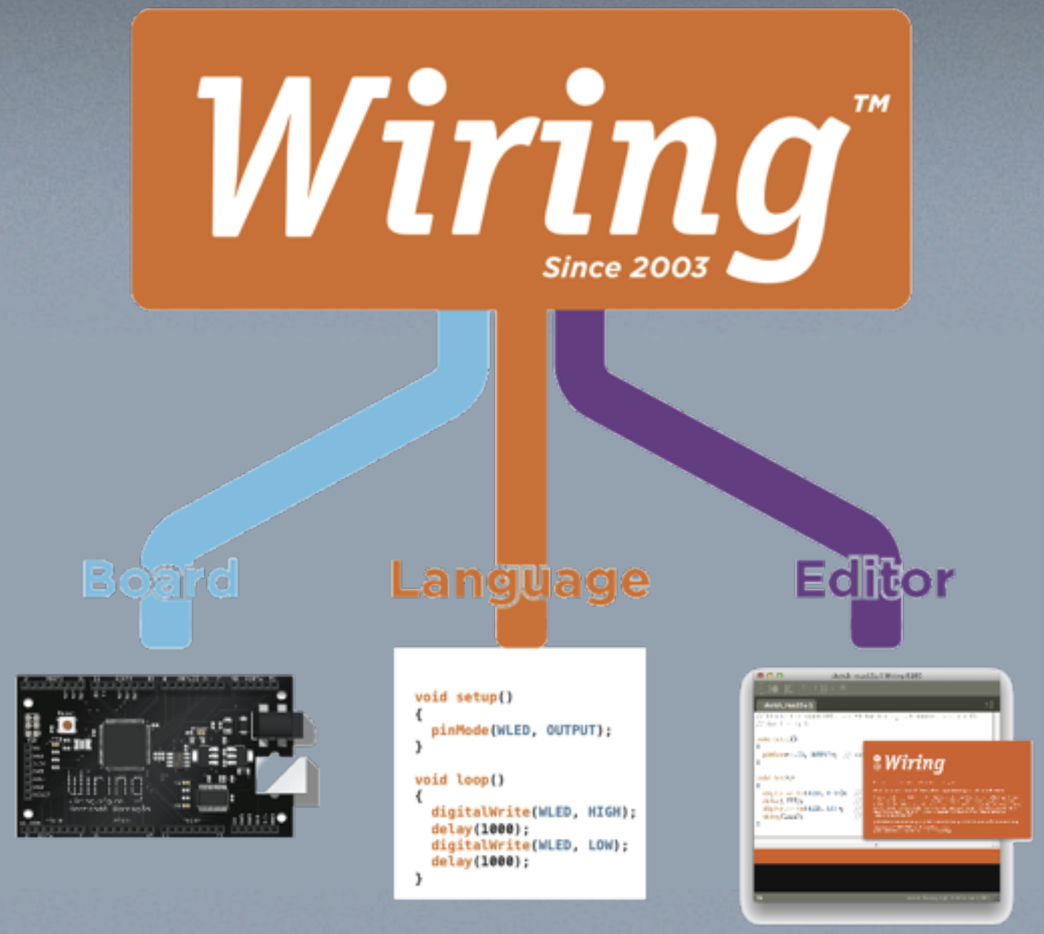
The Wiring Project:

who? Hernando Barragan
where? Interaction Design Institute of Ivrea (Italy)
when? Summer 2003

the beginning...

Wiring is an open-source programming framework for microcontrollers enabling anyone to program all kinds of interactive objects, spaces and physical experiences.

the beginning...



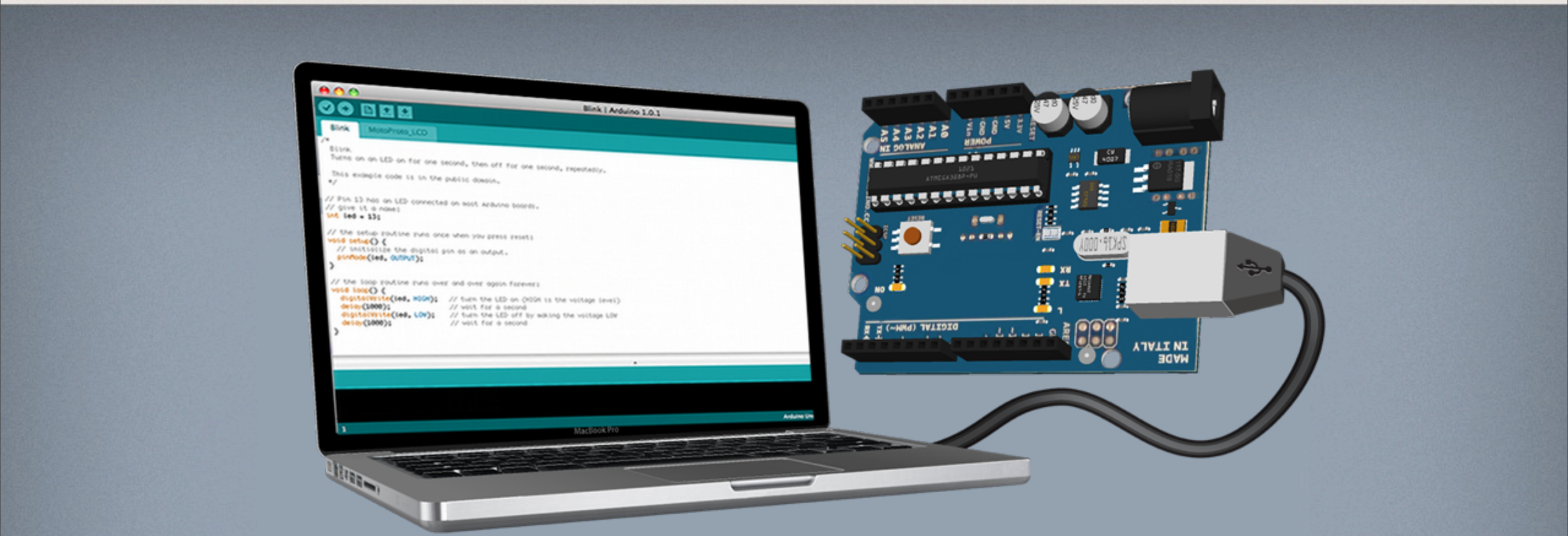
Board + Language + Editor

the beginning...

the beginning...

But wait.. Isn't that just like Arduino?

the beginning...



Arduino also consists of a set of boards, a language, and an editor.

the beginning...

Arduino was built around the Wiring project of Hernando Barragan... I don't think Arduino would exist without Wiring and I don't think Wiring would exist without Processing.

-Casey Raes (Processing co-creator)

the beginning...

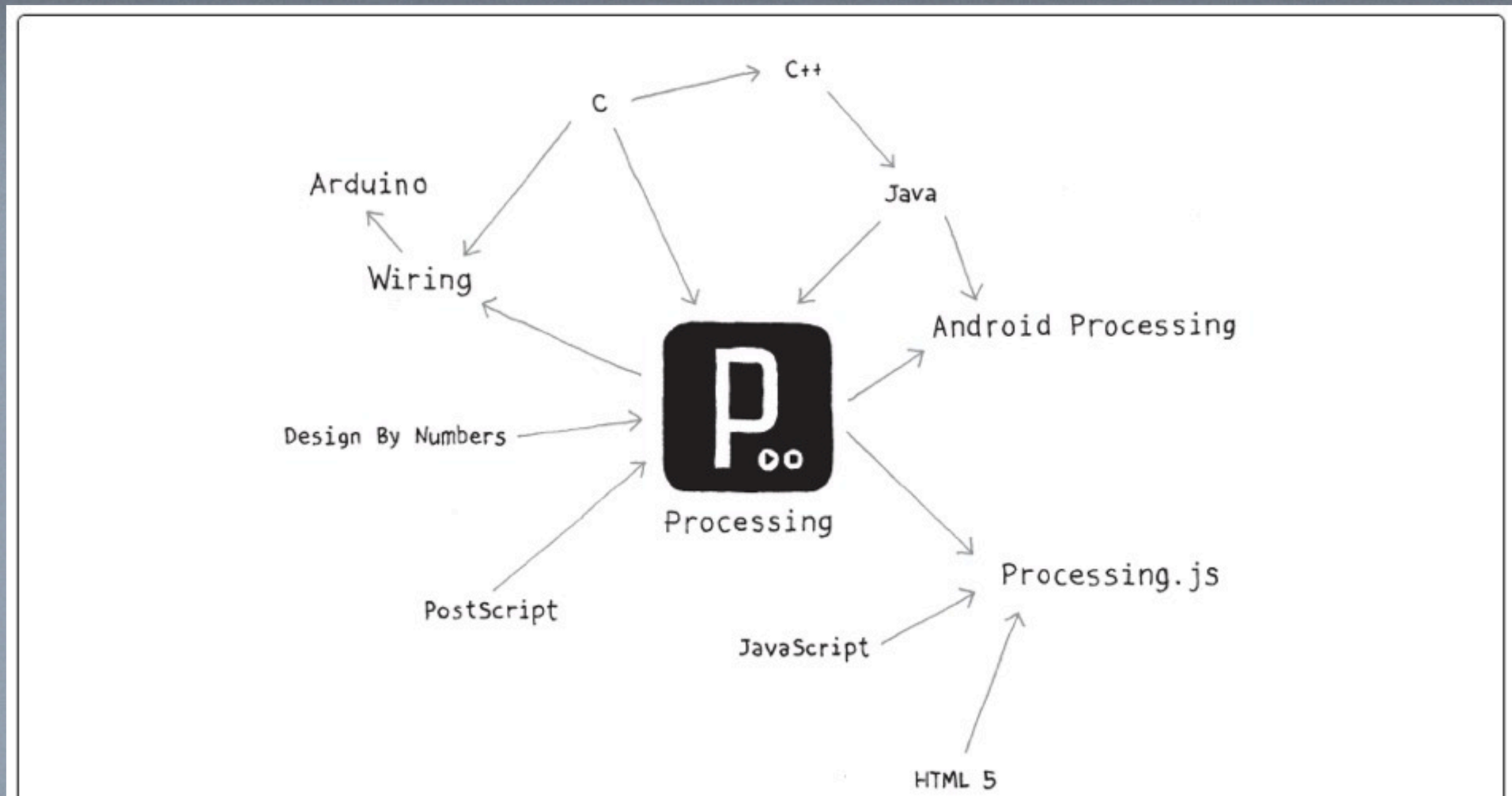


Figure 1-4. Processing has a large family of related languages and programming environments.

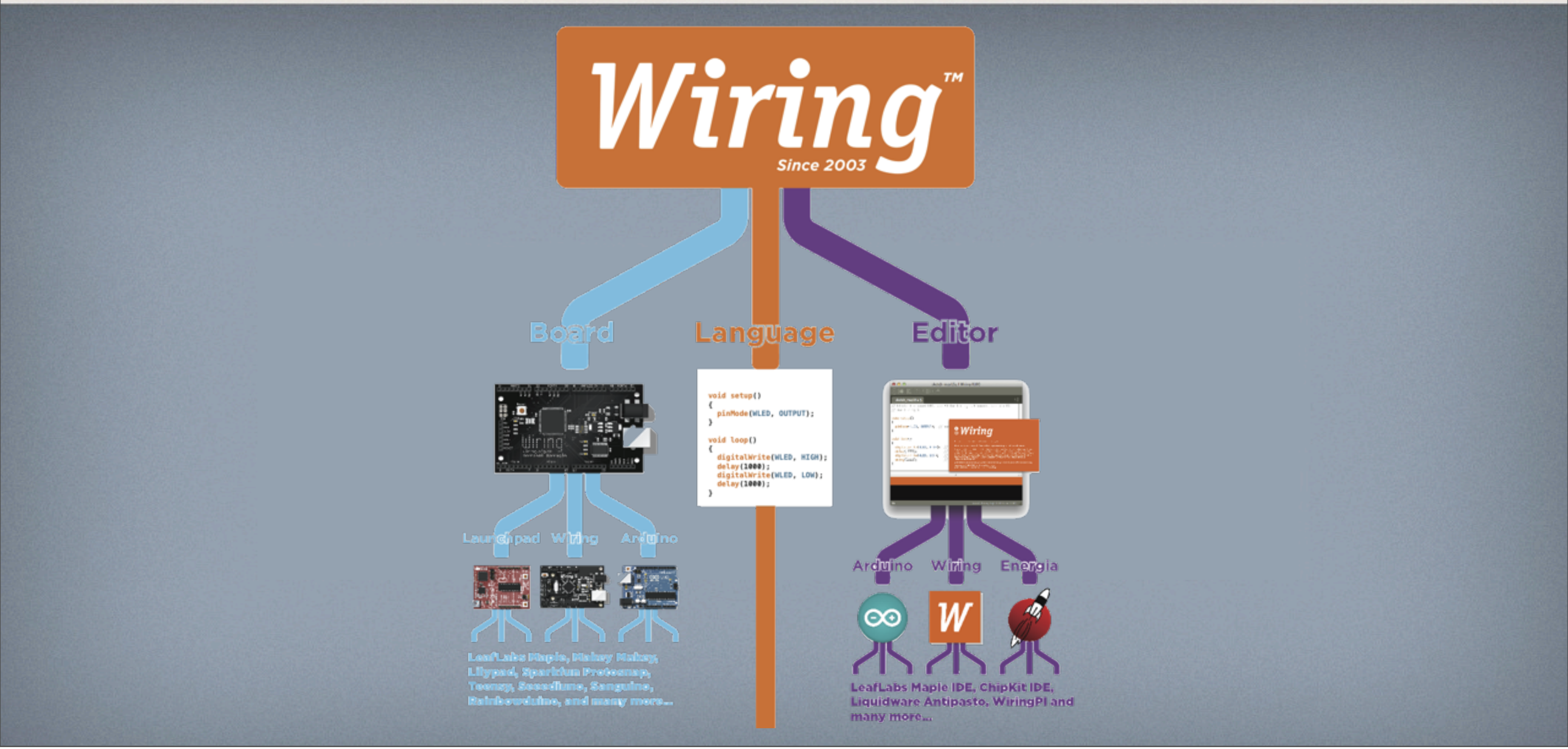
10 years later...

Wiring turns 10 this summer!

the now...

What's happened with the Wiring language?

the now...


Wiring™
Since 2003
Board
Language
Editor
Launchpad
Wiring
Arduino
LeafLabs Maple, Mikey Mikey, Lilypad, Sparkfun Proteomyx, Teensy, Seeedlume, Sanguino, Rainbowduino, and many more...
void setup()
{
pinMode(WLED, OUTPUT);
}
void loop()
{
digitalWrite(WLED, HIGH);
delay(1000);
digitalWrite(WLED, LOW);
delay(1000);
}
Wiring
Arduino
Wiring
Energia
LeafLabs Maple IDE, ChipKit IDE, Liquidware Antipasto, WiringPI and many more...

the now...

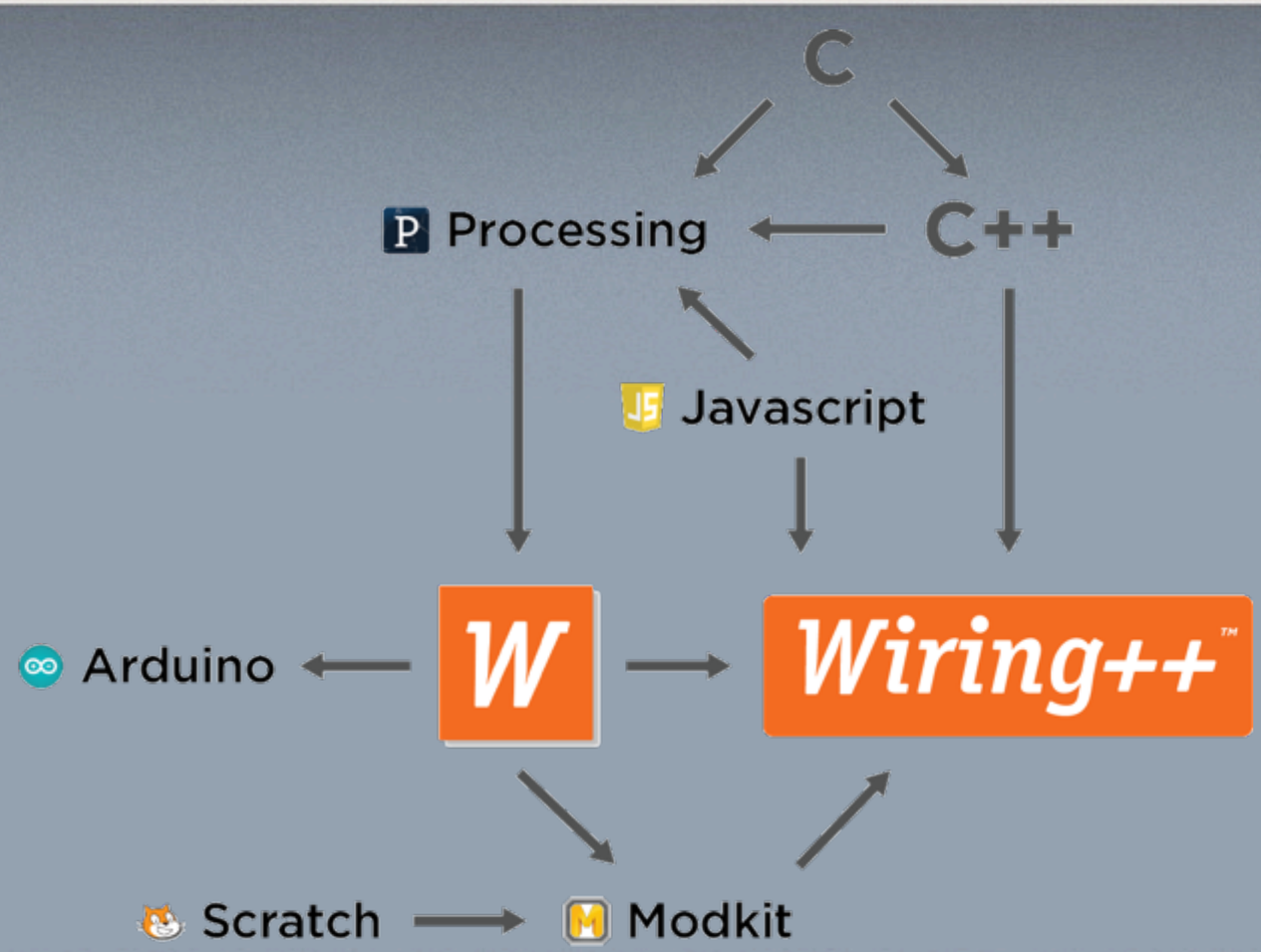
Not much has happened with the Wiring language!

the now: setup/loop...

```
void setup(){
    setupSomething();
}
```

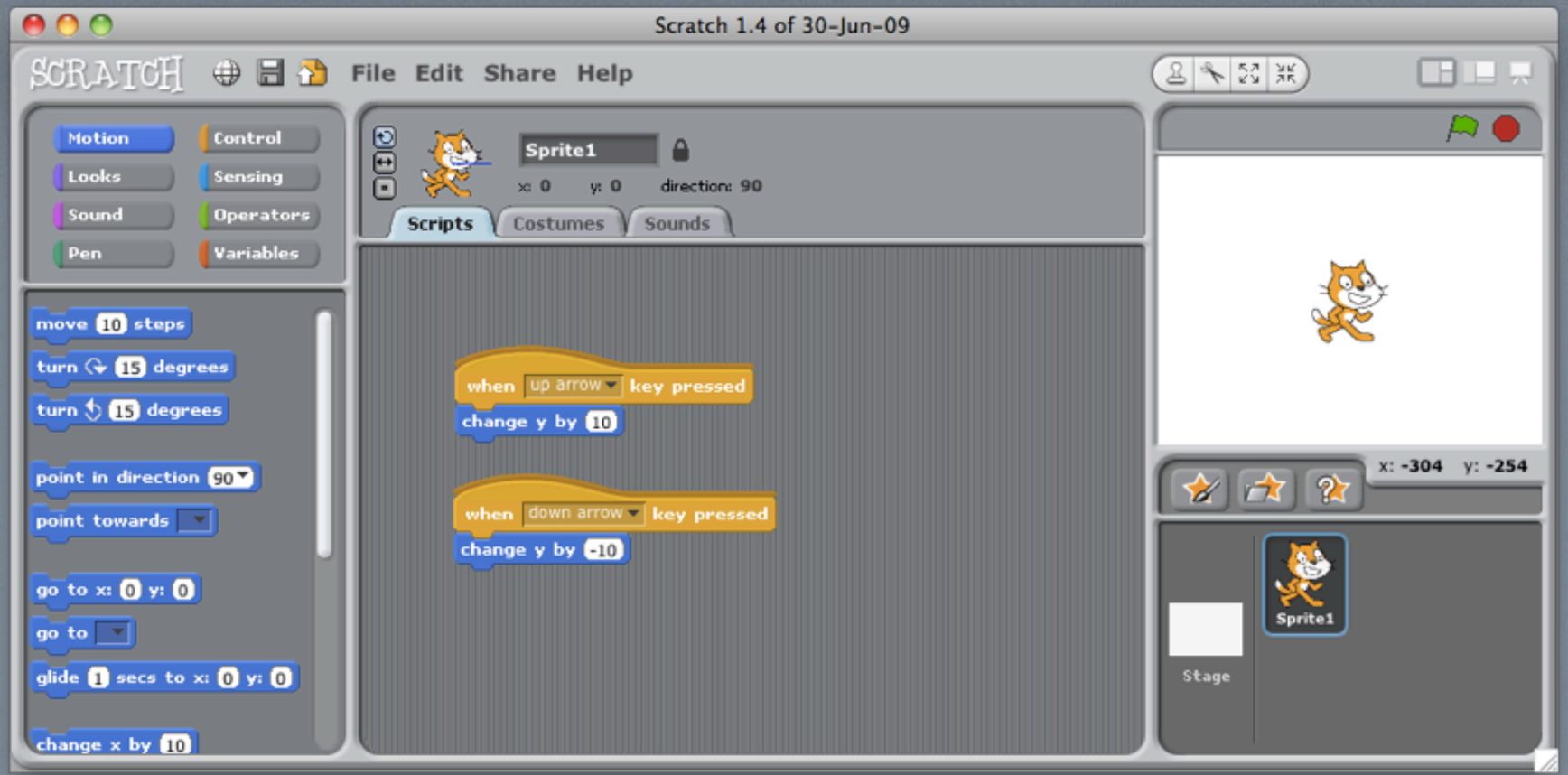
```
void loop(){
    if(checkSomething()){
        doSomething();
    }
}
```

the future...



Extending the family...

the future...



Scratch-like threads and Events?

the future: events...

```
GlobalEvents{EVENT1,EVENT2};

void events(){
   when(EVENT1){
     doSomething();
   }
      when(TWOSEC){
     doSomethingElse();
   }
}
void setup(){
     broadcast(EVENT1);
}
```

the future: threads...

```
void threads() {
    thread(THREAD_NAME) {
     while (1) {
        checkAndUpdate()
        delay(100);
      }
   }
}

void setup(){
    THREAD_NAME.start();
}
```

the future: built-in events...

Named events can be user generated or built-in events such as START which fires on startup after setup().

```
void events(){
    when(START){
      doSomething();
    }
}
```

the future: threading and event implementation...

Threading and event model needed to be very lightweight to fit on extremely constrained devices (MSP430-512b/ram)

~12 bytes ram/thread

~80 bytes flash/thread

the future: Dual C/C++ APIs...

Wiring++ API (C++ syntax)

```
Serial1.println(“Wiring++”);
PIN13.pinMode(OUTPUT);
PIN13.digitalWrite(HIGH);
```

Wiring Classic API (C syntax)

```
println(Serial1,“Wiring++”);
pinMode(PIN13,OUTPUT);
digitalWrite(PIN13,HIGH);
```

the future: pins as objects...

The Wiring++ API is centered around Pins-as-objects so rather than only being able to pass a pin to a function, Pins themselves have built-in functionality.

```
DigitalIoPin relayPin = PIN13;
relayPin.pinMode(OUTPUT);
relayPin.digitalWrite(HIGH);
```

the future: pins as objects - method chaining...

By introducing pins as objects we also gain the ability to add method chaining to all pin methods that would otherwise have no return value: e.g.

```
PIN13.pinMode(OUTPUT).digitalWrite(HIGH);
```

the future: pins as objects - interfaces...

Component authors can specify what functionality a valid pin parameter will have by naming a given Pin Interface -- e.g. PWM (compiler errors vs undefined runtime behavior):

```
void component::method(PwmPin pin, uint8_t volume){
        pin.analogWrite(volume*25);
 }
```

the future: component events...

Component events allow users to write code that respond to component-based-events that trigger automatically.

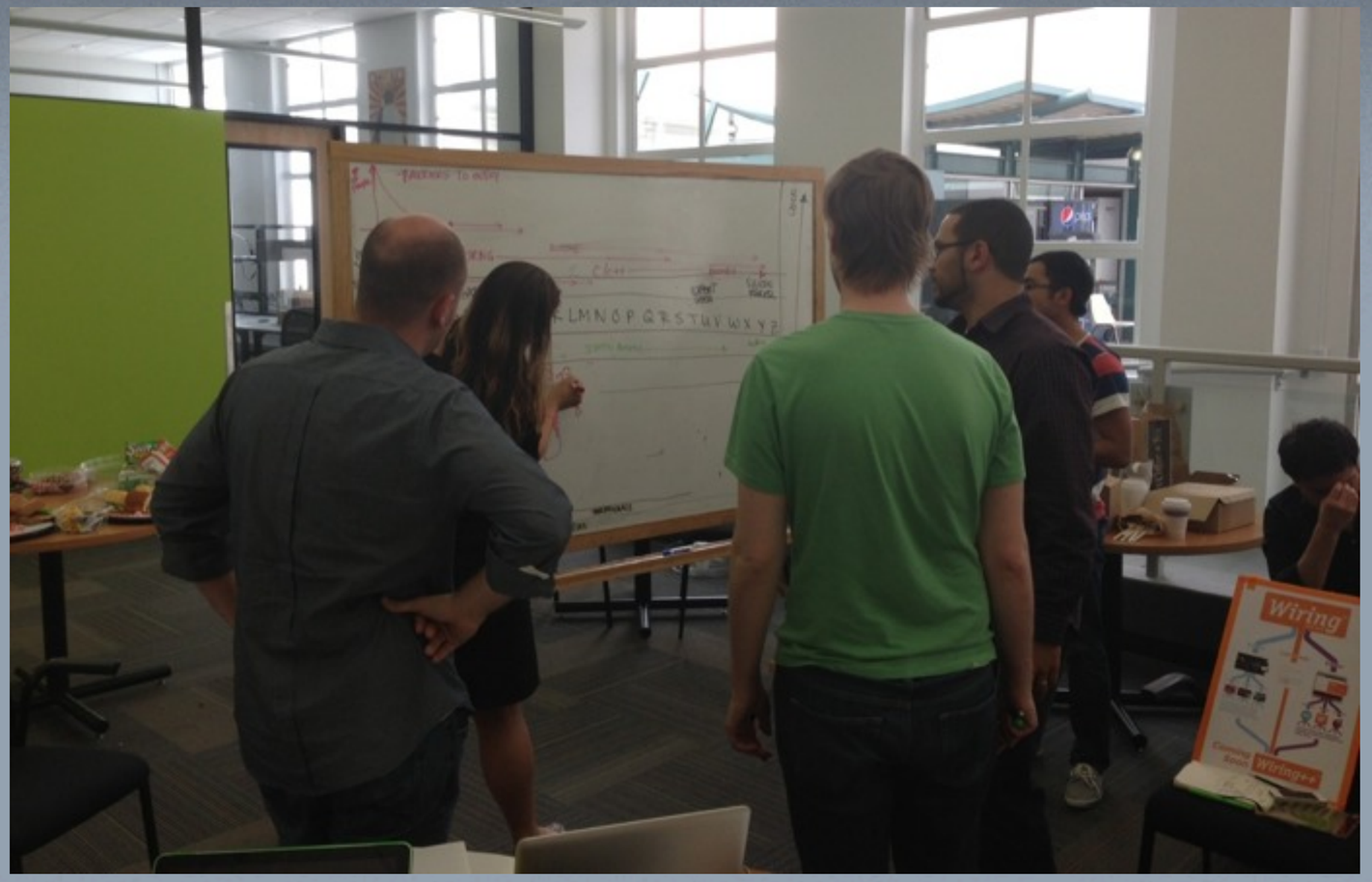
```
#include “Button.h”
Button button = Button(12);

void events() {
    when(button.PRESSED) {
        doSomething();
    }
}
```

the future: Wiring++ licensing...

Wiring has traditionally had an LGPL (Lesser GNU Public License aka Library GNU Public License)

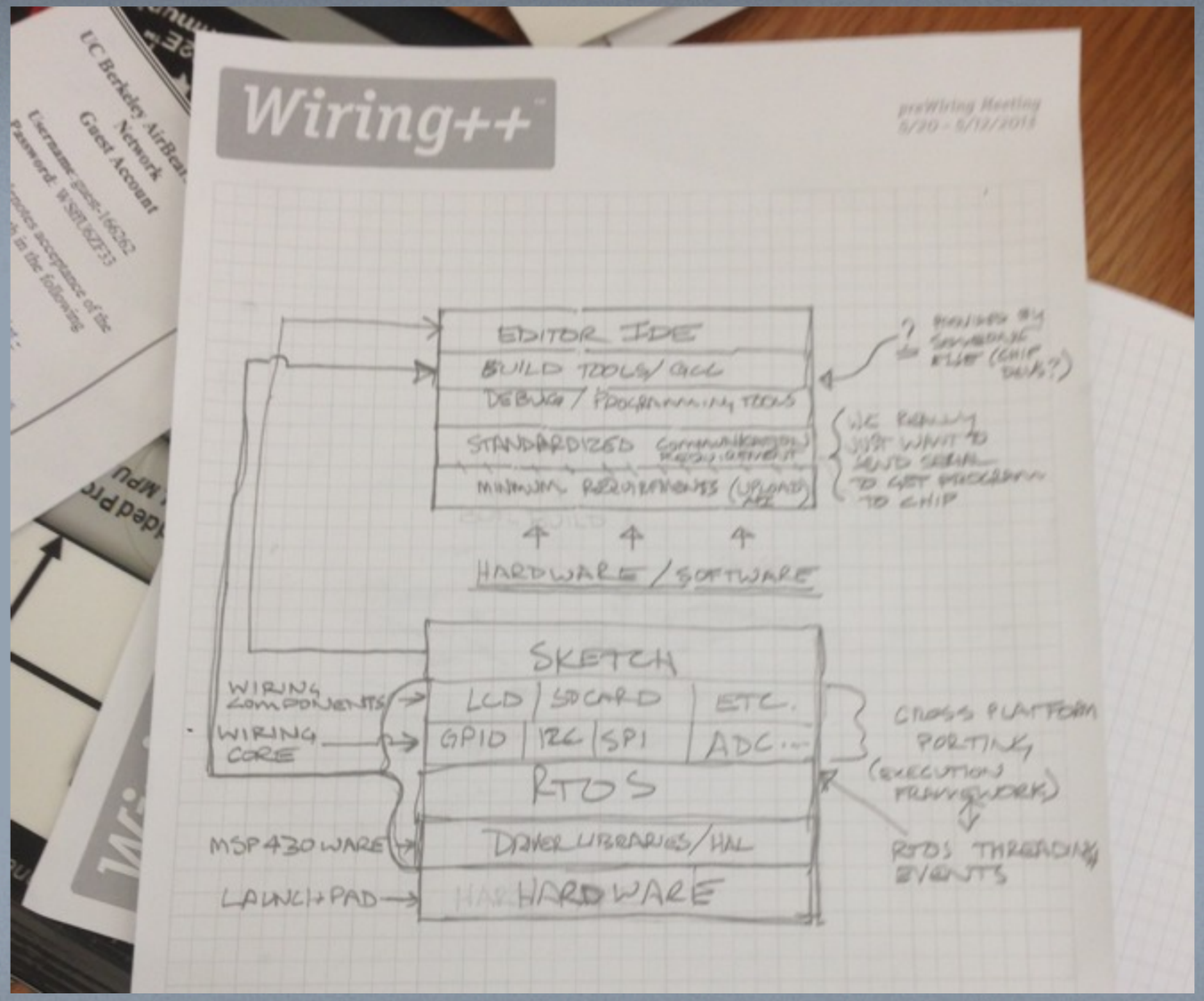
But many businesses and end-user products are being developed around the Wiring APIs. To better support these uses, we're moving to an MIT/BSD license.

who?

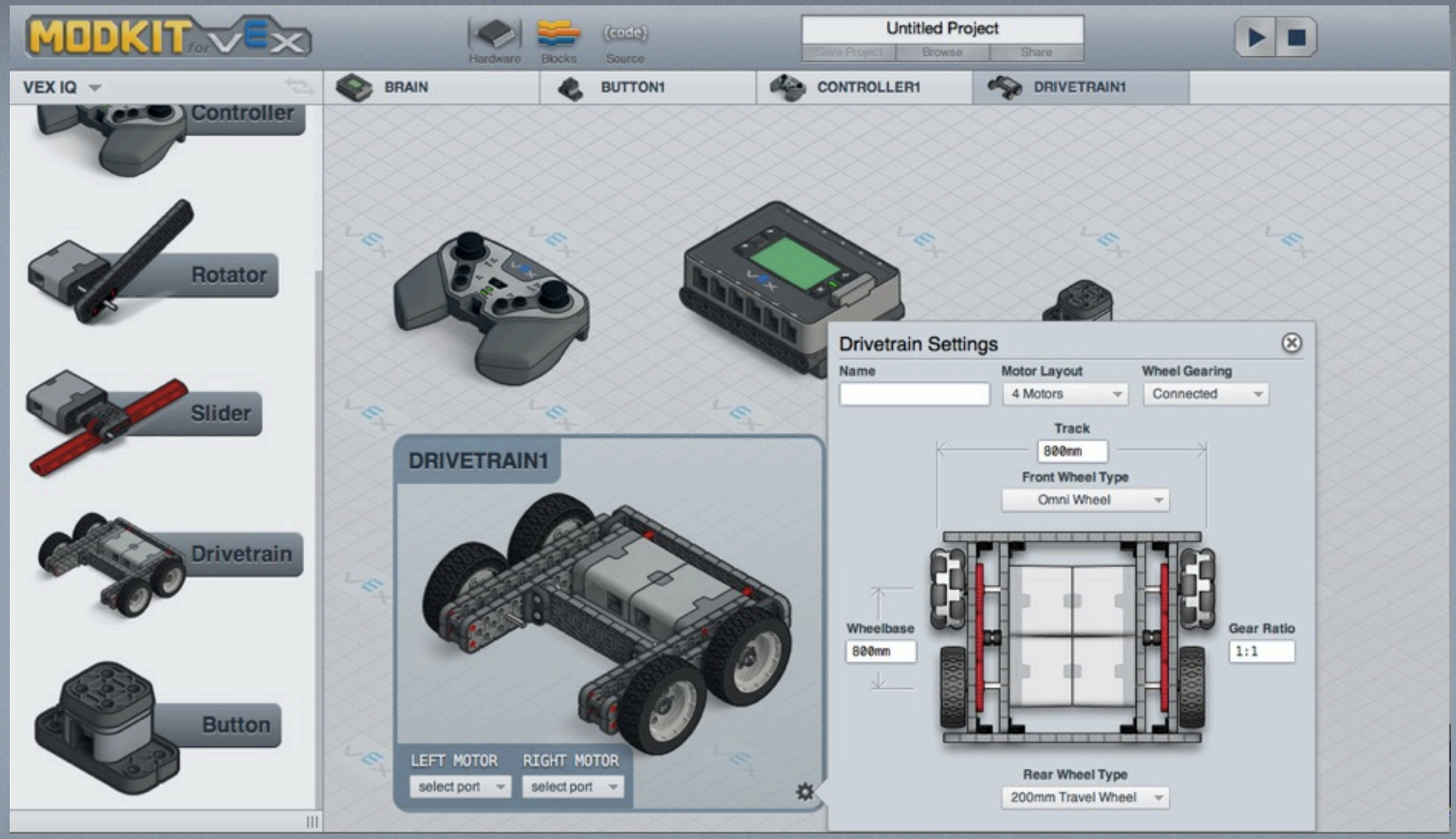


Wiring + Modkit + Texas Instruments

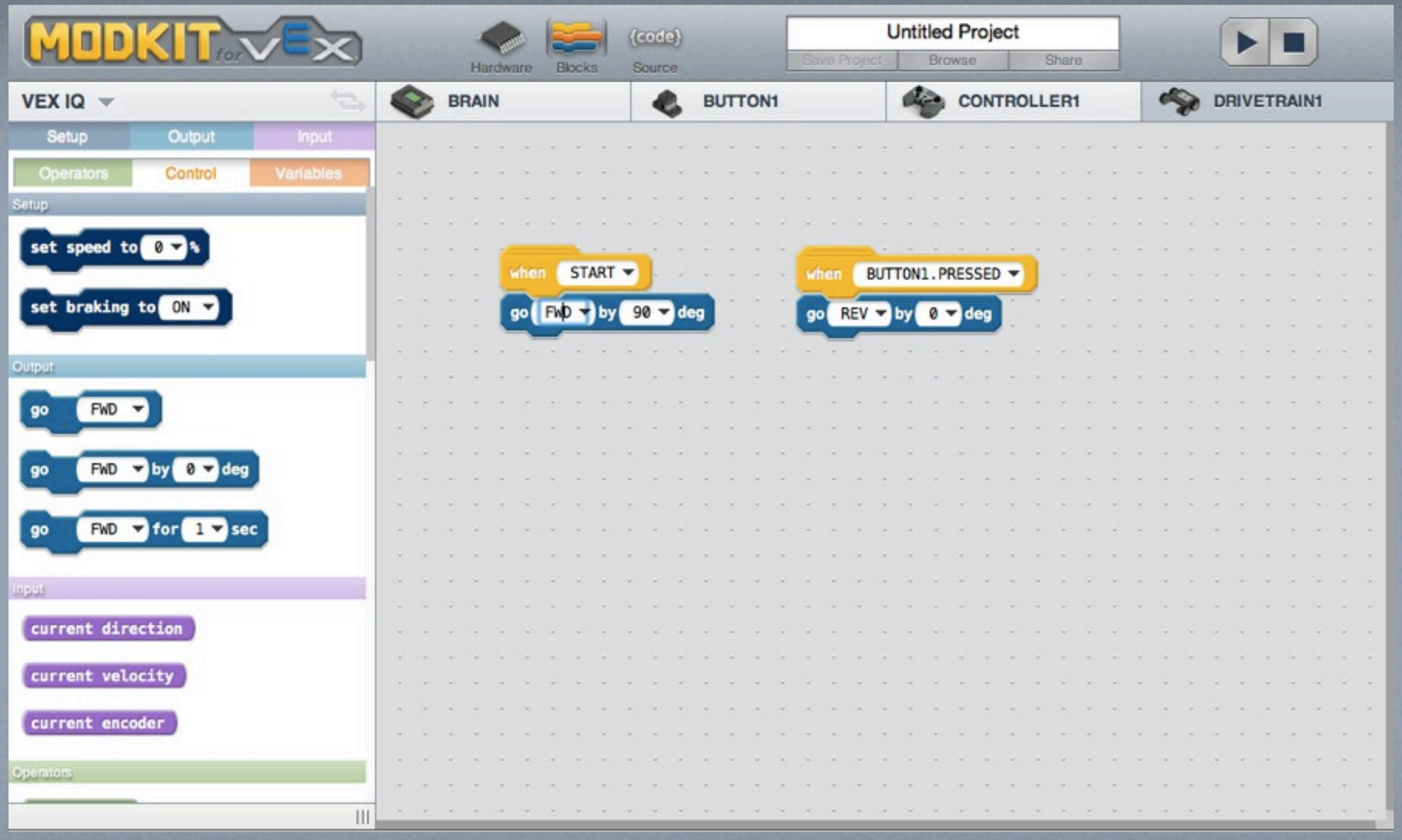
who?


Wiring++
preWiring Meeting
5/20 - 5/12/2013
EDITOR IDE
BUILD TOOLS/ GCC
DEBUGGER / PROGRAMMING TOOLS
STANDARDIZED COMMUNICATION PROTOCOL
MINIMUM REQUIREMENTS (UPLOAD ETC)
HARDWARE / SOFTWARE
SKETCH
LCD
SD CARD
ETC.
GPIO
I2C
SPI
ADC...
RTOS
DRIVER LIBRARIES / HAL
HARDWARE
WIRING COMPONENTS
WIRING CORE
MSP430 WARE
LAUNCHPAD
CROSS PLATFORM PORTING
(EXECUTION FRAMEWORK)
RTOS THREADING EVENTS
WE REALLY JUST WANT TO SEND SERIAL TO GET PROGRAM TO CHIP

why?


MODKIT for VEX
Hardware
Blocks
{code}
Source
Untitled Project
Save Project
Browse
Share
VEX IQ
BRAIN
BUTTON1
CONTROLLER1
DRIVETRAIN1
Controller
Rotator
Slider
Drivetrain
Button
DRIVETRAIN1
LEFT MOTOR
select port
RIGHT MOTOR
select port
Drivetrain Settings
Name
Motor Layout
4 Motors
Wheel Gearing
Connected
Track
800mm
Front Wheel Type
Omni Wheel
Wheelbase
800mm
Gear Ratio
1:1
Rear Wheel Type
200mm Travel Wheel

why?


MODKIT for VEX
Hardware
Blocks
{code}
Source
Untitled Project
Save Project
Browse
Share
VEX IQ
BRAIN
BUTTON1
CONTROLLER1
DRIVETRAIN1
Setup
Output
Input
Operators
Control
Variables
Setup
set speed to 0 %
set braking to ON
Output
go FWD
go FWD by 0 deg
go FWD for 1 sec
Input
current direction
current velocity
current encoder
Operators
when START
go FWD by 90 deg
when BUTTON1.PRESSED
go REV by 0 deg

thanks...

Wiring team: Hernando Barragan, Brett Hagman, Alex Brevig

TI team: Adrian Fernandez, Robert Wessels