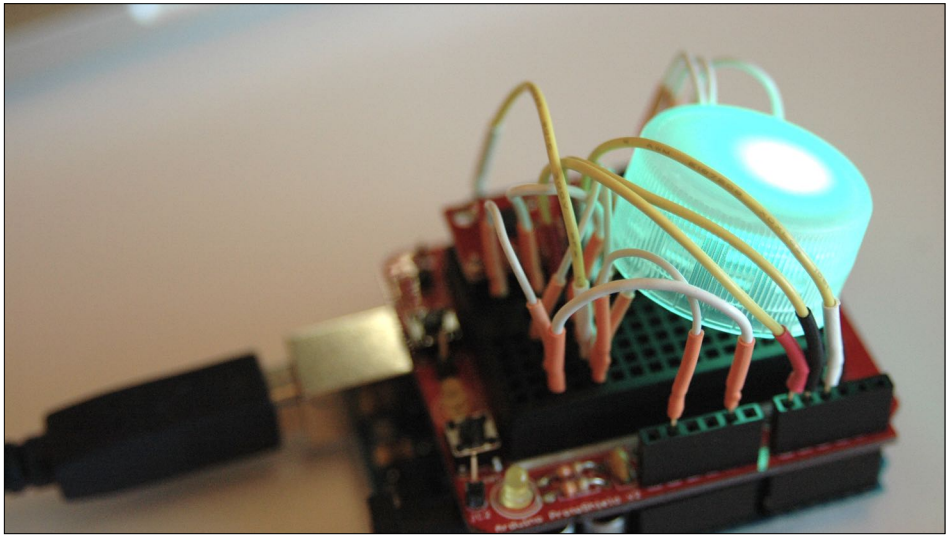
Drawing is putting a line round an idea

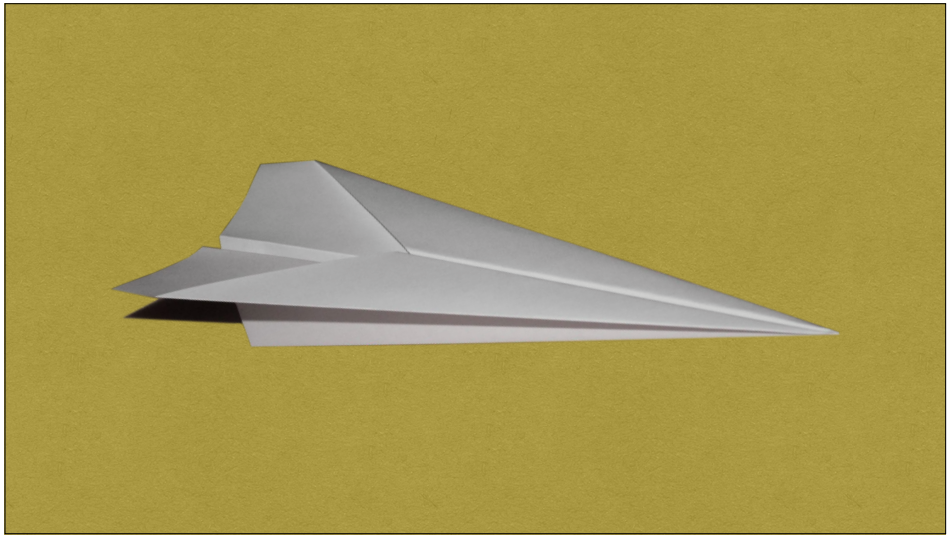
~Henri Matisse

1

2



3



4

WHY?

5

Hack to Learn

6

Hack to Communicate

7

Hack to Inspire

8

9

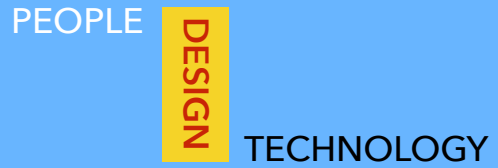
oscon

Hacking Smart Electronics

10

I'm Robert Gallup

11



12

Quick Survey: Coders?
Designers?
Strategists?
Heard of Arduino?
Hacked Hardware?

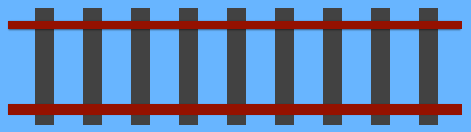
13

Agenda

Mad Cap Hacks
Concepts
LEDs, Buttons, Knobs, Speakers
Wrap-up

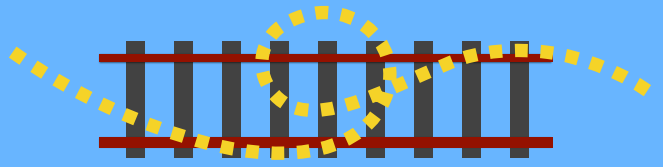
14

Tracks



15

Off The Rails



16

#oscon
#hackingsmartelectronics
@robertgallup

17

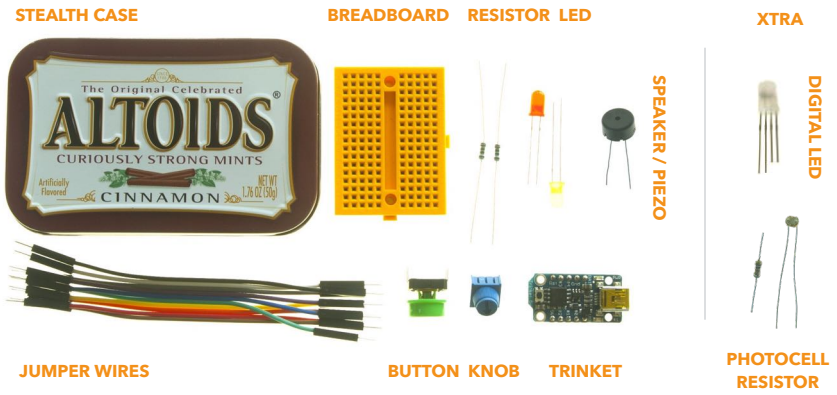
?

18

Let's Get Started!

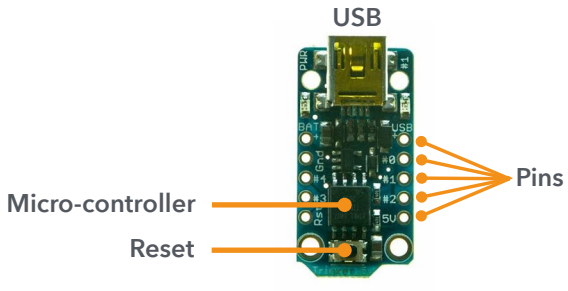
19

Your Hacker's Kit



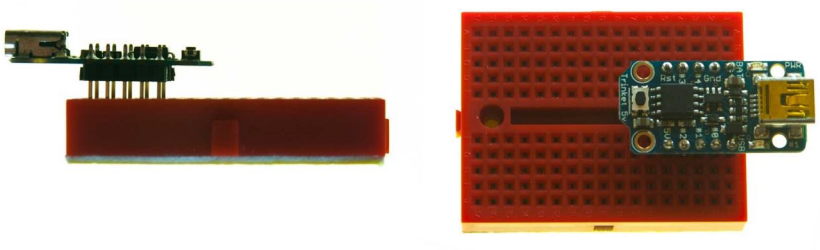
20

Trinket (Adafruit)



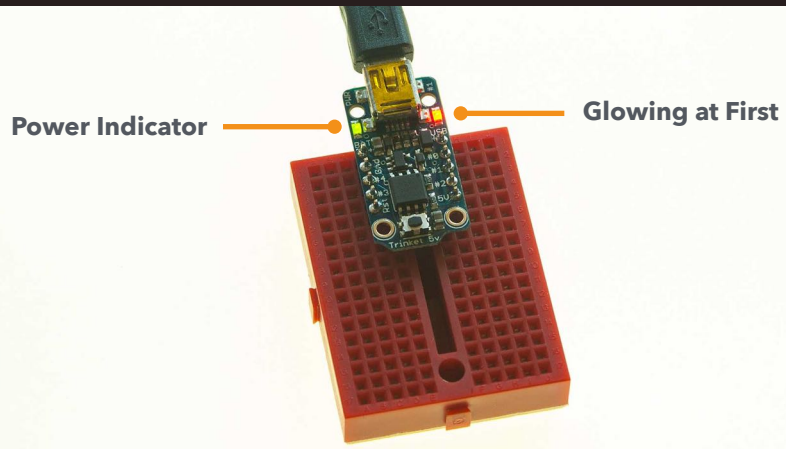
21

Put the Trinket in the Breadboard



22

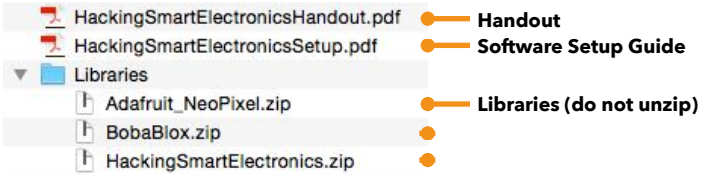
Connect the Trinket to your Computer (USB)



23

Download Courseware (and unzip)

robertgallup.github.io/get/OSCONCourseware.zip



24

Start Arduino

25

Mac OS X: Security Settings

1

Open Preferences

2

Unlock, and change "Allow apps ..." to Anywhere

3

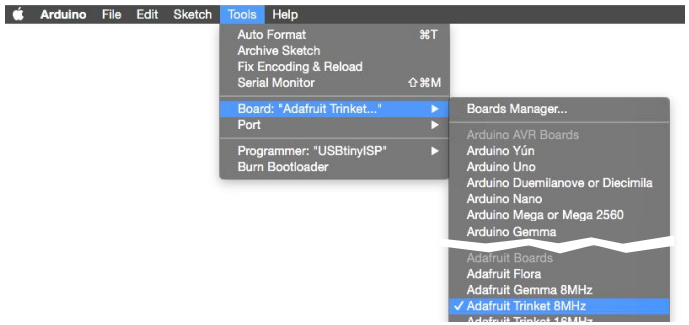
Run

4

IMPORTANT: Reset security settings

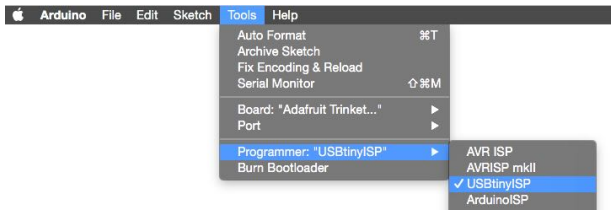
26

Configure the Board



27

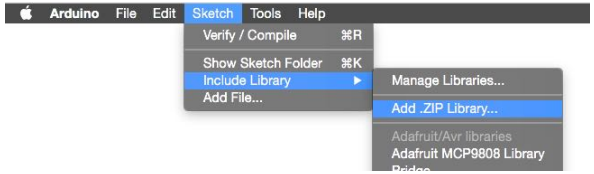
Configure the Programmer



28

Add the Workshop Examples

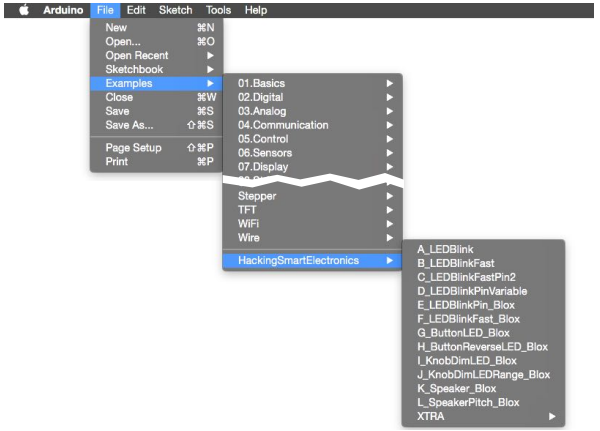
1



2 **Select HackingSmartElectronics.zip in the Libraries folder from the courseware download**

29

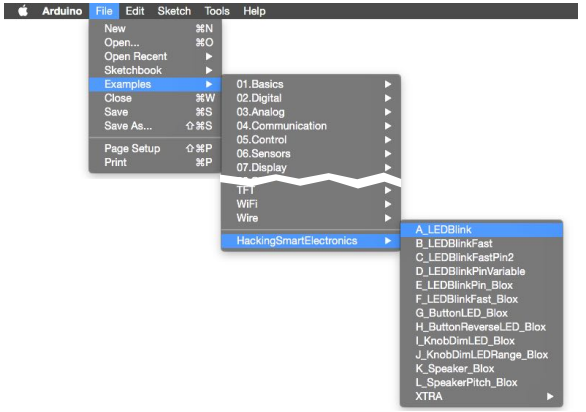
If it works, …



30

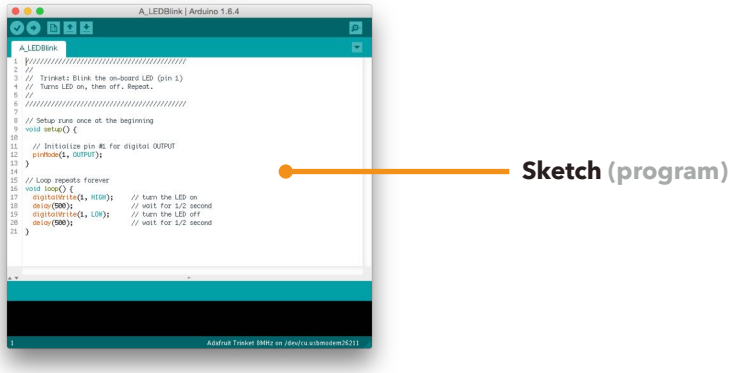
31

Load the LEDBlink Example

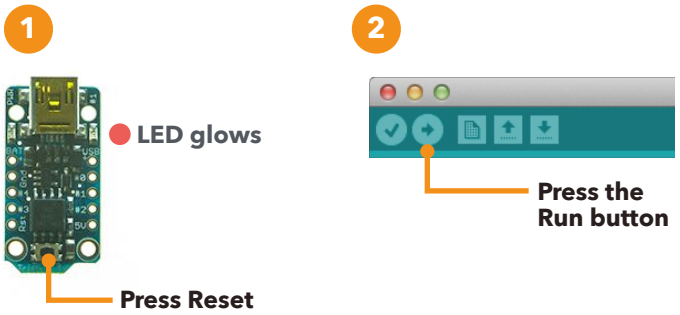


32

The Sketch Window

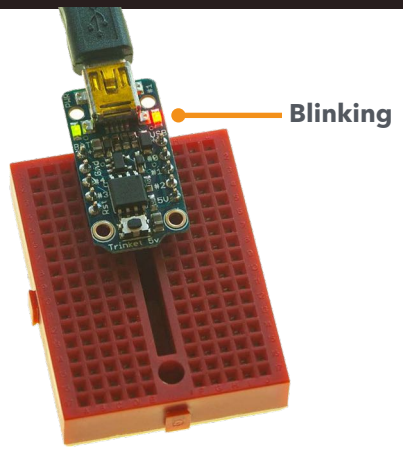


Run the Sketch on your Trinket



Wait. What?!

The "Blink"



Sketch Structure

Comment

Statements

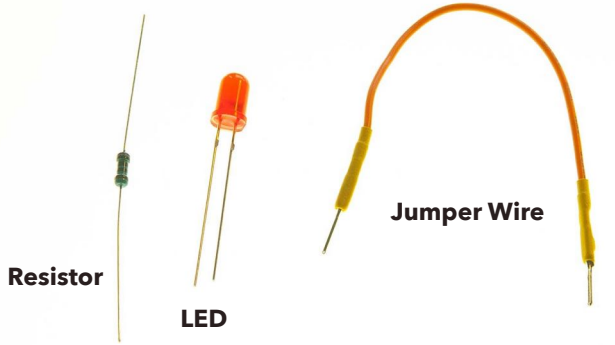
```
// Setup runs once
void setup() {
 pinMode(1, OUTPUT);       // Pin #1 is output
}

// Loop repeats forever
void loop() {
  digitalWrite(1, HIGH);  // turn the LED on
  delay(500);             // wait for 1/2 second
  digitalWrite(1, LOW);   // turn the LED off
  delay(500);             // wait for 1/2 second
}
```


Hack: Change the Blinking Speed

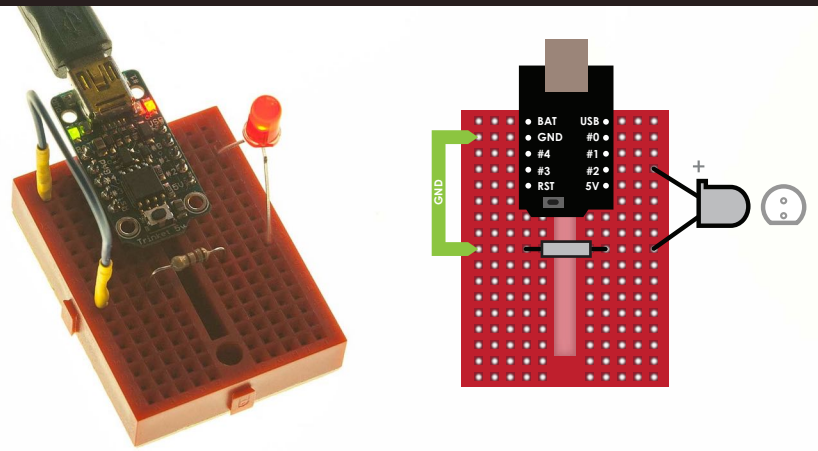
1. **Change the blink speed**
   Hint: edit the blink sketch to change the blink delay from 1 second to 1/4 second (note: 1 second = 1000 milliseconds). Then run the sketch again.
2. **Use the File > Save command to save your sketch with a new name.**
3. **STRETCH: Make a short-on, long-off blink**
   Hint: the delay times don't have to be the same.

Breath.

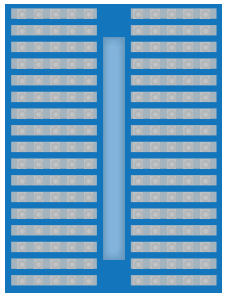
Your First Circuit



Wire Up an LED



A Note about Breadboards



All of the contact points in the same row are connected

The "rift" separates the two sides

Hack: An LED on Another Pin

1. **Rewire your hack to move the LED to Trinket pin #2.**
2. **In the "blink" sketch, change the LED pin to match.**
   Hint: You may have to change more than one line

Solution: An LED on Another Pin

```
// Setup runs once
void setup() {
 pinMode(2, OUTPUT);       // Pin #2 is output
}

// Loop repeats forever
void loop() {
  digitalWrite(2, HIGH);  // turn the LED on
  delay(1000);            // wait for a second
  digitalWrite(2, LOW);   // turn the LED off
  delay(1000);            // wait for a second
}
```

Solution: Another Way, "Variable"

Declare Variable

pinLED = 2

```
// Declarations
int pinLED = 2;

// Setup runs once at the beginning
void setup() {
  pinMode(pinLED, OUTPUT);       // Pin is output
}

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


Variable Declaration

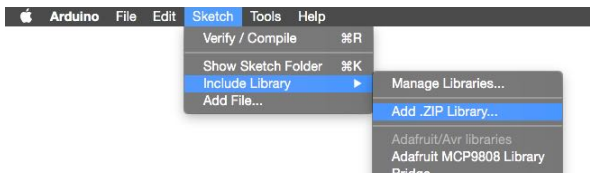
```
int pinLED = 2;
```

TYPE NAME VALUE

Libraries Make It Easier

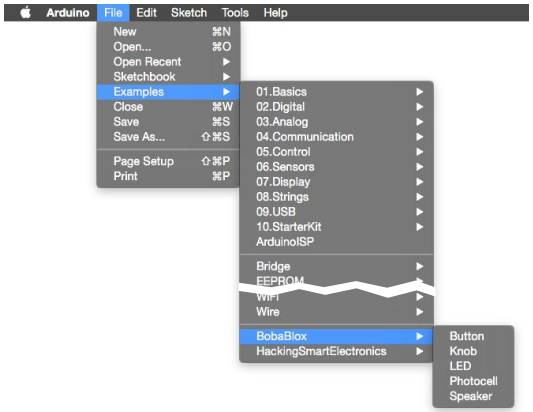
Add the BobaBlox Library

1

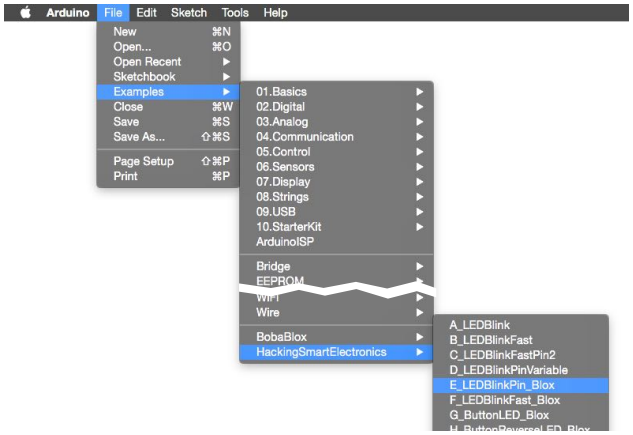


2 **Select BobaBlox.zip in the Libraries folder from the courseware download**

Verify Installation



Load and Run the LEDBlinkPin_Blox Example



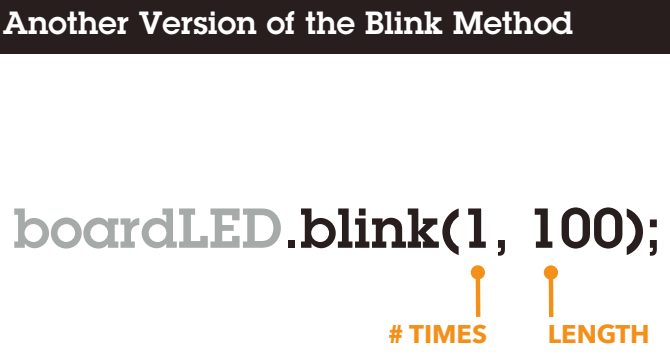
What's Different About This Sketch?

```
// Include the library
#include <BobaBlox.h>

// Declarations
LED boardLED(1);      // Declare an LED on pin #1

// Setup runs once
void setup() {
}

// Use the "blink" method to blink boardLED. Repeat forever.
void loop() {
 boardLED.blink();
}
```


LED Variable Declaration

Using a Method

Another Version of the Blink Method



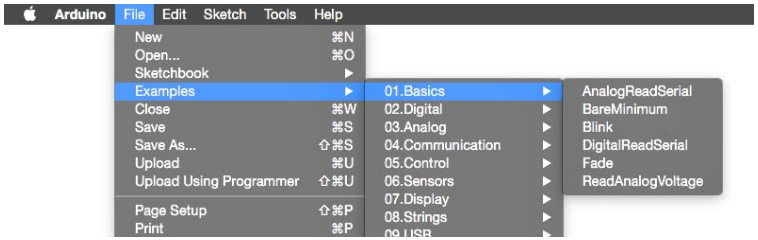
Hack: An LED on Any Other Another Pin …

1. **Change the library Blink sketch to use an LED on pin #3**
   Hint: You can add a 2nd LED, or move the LED you connected in the previous hack.

Hack: Fast Blink

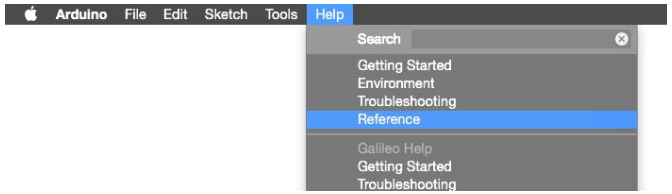
1. **Change the library sketch to blink faster**
   Hint: look at the example of the LED blink method that uses two parameters, times and delay.

Off The Rails: Examples



57

Off The Rails: Reference

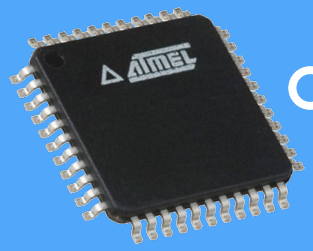


58

Breath, again.

59

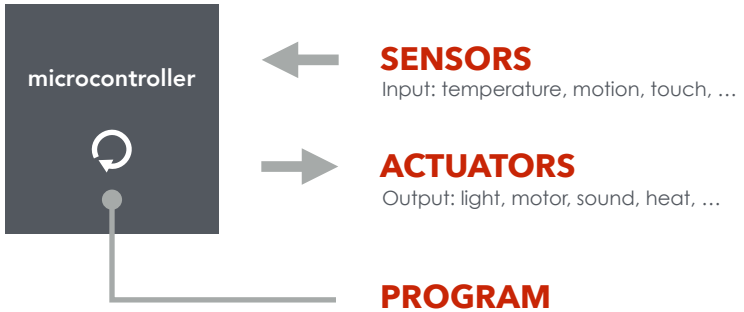
60



Of Micro-controllers and Electronics

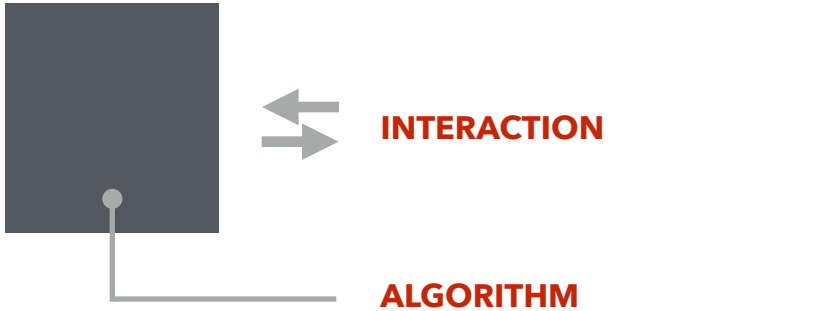
61

A Microcontroller



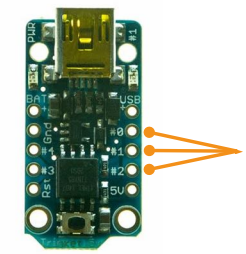
62

Experience = Interaction + Algorithm



63

Interaction



Micro-controllers sense and control voltages on their pins.

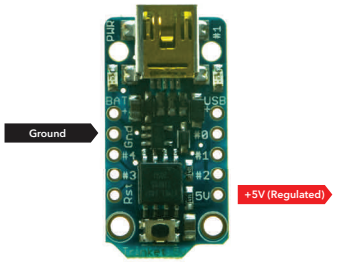
64

Most of us are familiar with the idea of voltage.

65

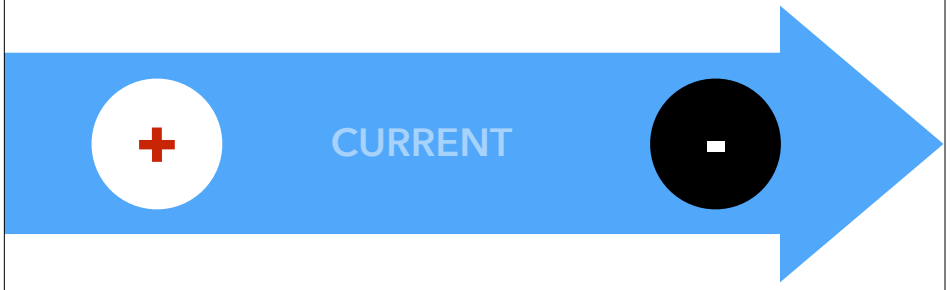
Trinket uses 5V power (USB)

Ground is zero volts. Pins can be HIGH (positive) or LOW (ground).



66

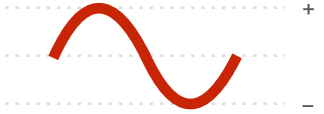
Voltage drives current (Positive to zero)



67

Alternating vs Direct Current

AC



DC

68

Some Interesting Facts

Wall		110-240 volts, 15-20 amps
Battery		1.5 volts, ~2+ amps
Static		~20,000 volts, ~1 milliamp

69

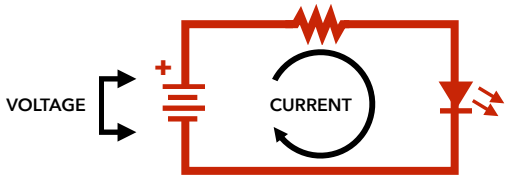
Volts Jolt! Current Kills!

70

But, Not Without a Circuit

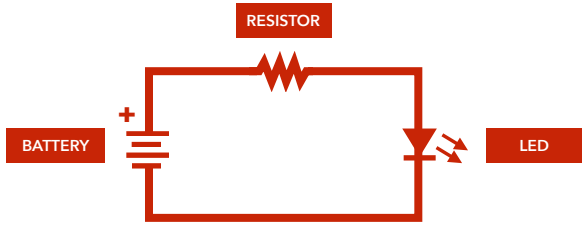
71

72



A Circuit allows current to flow from positive to ground

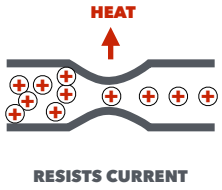
73



Circuits use Components

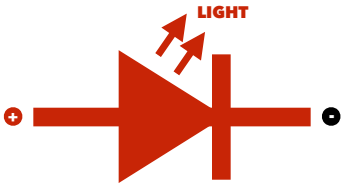
74

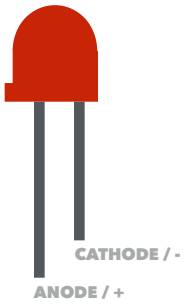
Component: Resistor (Ohms Ω)



75

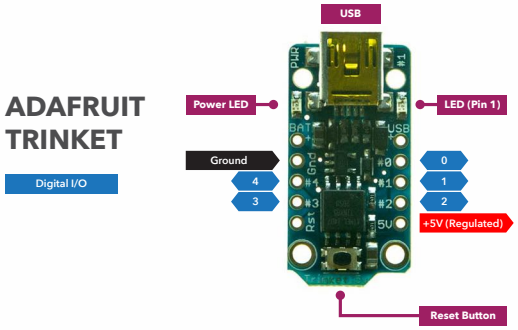
Component: LED (Light Emitting Diode)





76

The Trinket Basic Connections (pinout)



77

78

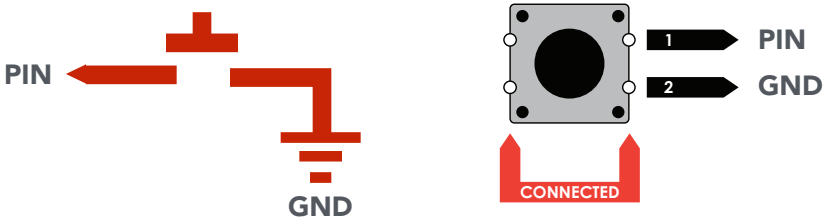
More Hacks

79

The Button (Digital Input)

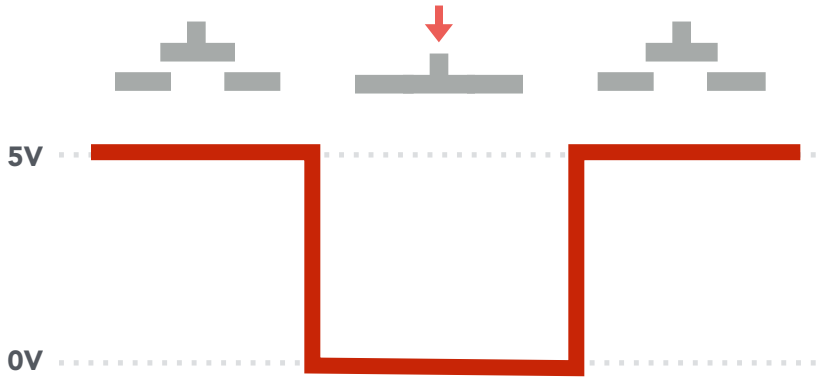
80

Component: Button



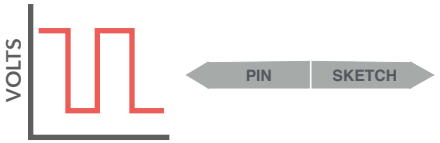
81

Button Voltage at Pin



82

Digital Input

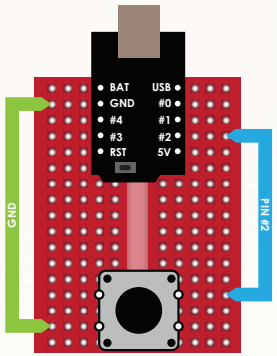


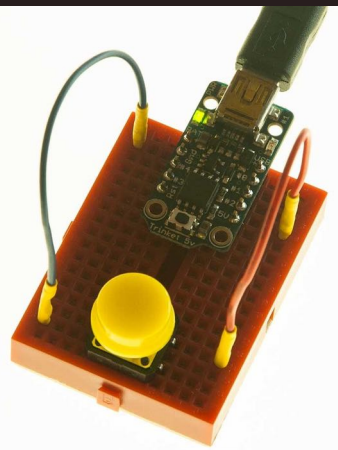
HIGH, LOW, HIGH ...

A digital input converts voltages above a threshold to HIGH.
Below the threshold is LOW.

83

Wiring Up a Single Button





84

Hack: Logic / Reverse Logic

1. **Load and run the ButtonLED_Blox sketch. How do you think it works?**
2. **Reverse the logic in the button sketch (i.e. make the LED go OFF when the button is pressed).**
   Hint: there are two ways to do this
3. **STRETCH: make the LED blink when the button is pressed.** Hint: what other LED functions do you have to work with?

ButtonLED Walkthrough

```
// Declarations
LED boardLED(1);
Button pushButton(2);

void setup() {
}

void loop() {


 if (pushButton.isDown()
   boardLED.on();
  } else {
   boardLED.off();
  }

 }
```


Pushbutton Variable Declaration

```
Button pushButton(1);
```

TYPE — NAME — PIN

Pushbutton isDown Method

```
pushButton.isDown();
```

NAME — METHOD NAME

Other Pushbutton Methods (true | false)

```
pushButton.isDown();
          .isUp();
          .wasPressed();
          .wasReleased();
```


Branching (One Thing or Another)

```
if (pushButton.isDown()) {
  boardLED.on();
} else {
  boardLED.off();
}
```

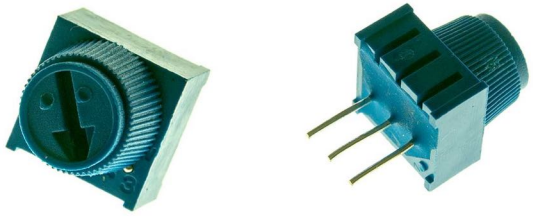

The If statement (if/else)

```
if (condition) {
    ●●●   ——— if condition is true
} else {
    ●●●   ——— if condition is false
}
```

CONDITIONS:

```
a
!a
a == b
a <  b
a >  b
a != b
```


The Knob (Analog Input)

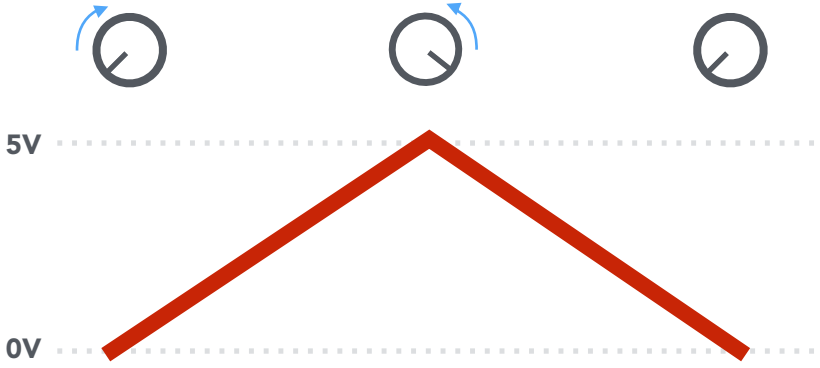


93

Component: Knob (Potentiometer)

94

Knob Voltage at Pin



95

Analog to Digital Converter (ADC)

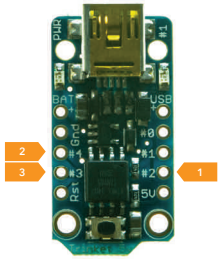
An Analog to Digital Converter, translates a voltage into a number the sketch can use.

96

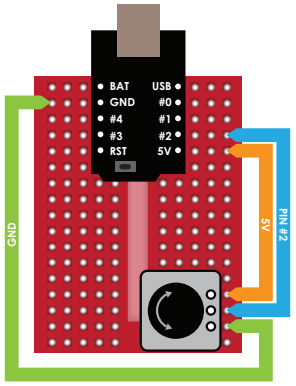
Trinket: The Analog Inputs

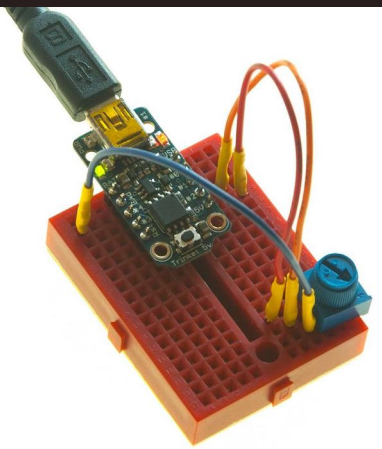
ADAFRUIT TRINKET

Analog Input



Wiring Up a Knob





Hack: Run the knobDimLED Example

1. **What happens when you turn the knob?**
2. **STRETCH: there's one anomaly with this sketch's behavior (experience). What is it?**
   Hint: look at the range of the analogRead() and analogWrite() commands in the Arduino reference. Those commands are the basis of the knob.value() and LED.setBrightness() methods.

KnobDimLED Walkthrough

```
LED boardLED(1);
Knob bluKnob(1);

int knobValue;

void setup() {
}

void loop() {

  // Get the knob value and use it to set the LED brightness
  knobValue = bluKnob.value();
  boardLED.setBrightness(knobBrightness

}
```

.setBrightness() method

Knob Variable Declaration

```
Knob bluKnob(1);
```

TYPE — NAME — PIN

101

Knob Value Method

```
bluKnob.value();
```

NAME — METHOD NAME

102

Knob Value Method with Range

```
bluKnob.value(0, 255);
```

MIN — MAX

103

LED Method to Set Brightness

NOTE: brightness only works for LEDs connected to a pin supporting analog output (see Trinket pinout).

```
boardLED.setBrightness(v);
```

LED VARIABLE — METHOD — VALUE 0-255

104

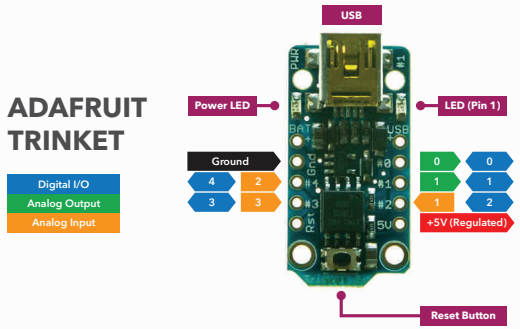
Hack: Change Knob Direction

1. **Change the direction of the knob (i.e. so the brightness goes down when the knob it turned up.)**

2. **STRETCH: Change the blink speed in the standard blink sketch based on the knob position.**

105

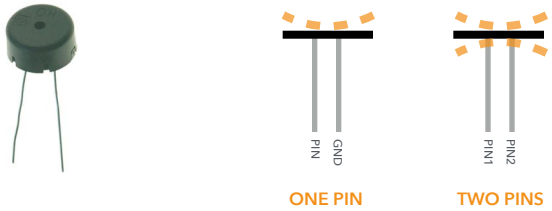
The Trinket: A More Complete Picture



106

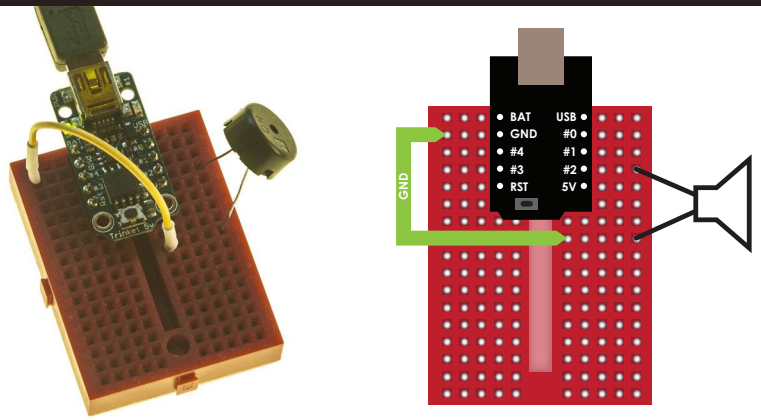
107

Speaker (Sound)



108

Wiring Up a Speaker



Hack: Run the Speaker Example

1. After wiring the Speaker up, load and run the Speaker sketch.
2. Change the pitch of the speaker
3. Make two beeps of different pitches each time through the loop

Speaker Sketch Walkthrough

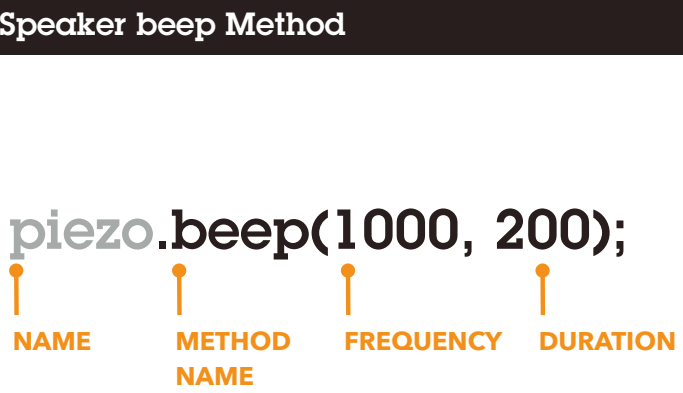
```
Speaker piezo(2);

void setup() {
}

void loop() {

  piezo.beep(1000, 200);
  delay (100);

}
```


Speaker Variable Declaration

```
Speaker piezo(2);
```

TYPE NAME PIN

Speaker beep Method



Hack: Run the SpeakerPitch Sketch

1. Load and run the SpeakerPitch sketch.
2. The For{} statement creates a loop within a loop. Look up the command in the Arduino reference.
3. Change pitch steps to 100Hz each, rather than 50.
4. STRETCH: Change the pitch step to be 10% each time rather than a fixed number.

Hack: More Speaker Hacks

1. BIG STRETCH: See if you can figure out how to control the pitch with a knob.
2. GIANT STRETCH: Wire up a button and use it to create a sound each time it is released. Create a sound with personality.

Wax On. Wax Off.

117

LED

118

Button

119

Knob

120

Speaker

121

Wrap Up

122

Many Other Platforms

Arduino

RPi

BeagleBone

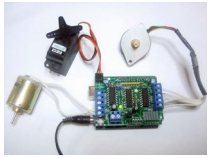
spark.io

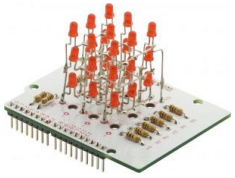
pinocc.io

teensy

123

Add-On Boards (Shields)





124

Next Steps

Bonus Material (End of these slides)

XTRA

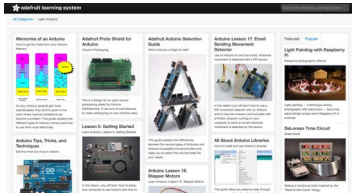
Digital LED
Photocell
Trinket @ 16MHz
Resistor Values

Online Tutorials

http://arduino.cc/en/Tutorial
http://playground.arduino.cc
https://learn.sparkfun.com

https://learn.adafruit.com
https://learn.adafruit.com/introducing-trinket

google "arduino tutorials"



Online Resources

ARDUINO: www.arduino.cc, reference, tutorials
SPARKFUN: www.sparkfun.com, components, devices, tutorials
ADAFRUIT: www.adafruit.com, components, devices, tutorials
MAKE MAGAZINE: www.makezine.com, magazine, books, stuff
MAKER SHED: www.makershed.com, components, kits, stuff
INSTRUCTABLES: www.instructables.com, tutorials
AMAZON: www.amazon.com, books, components, devices, tutorials
O'REILLY: www.oreilly.com, books, tutorials, etc.

ELECTRONIC GOLDMINE: http://www.goldmine-elec-products.com, components
JAMECO: www.jameco.com, components
MOUSER: www.mouser.com, components
DIGIKEY: www.digikey.com, components

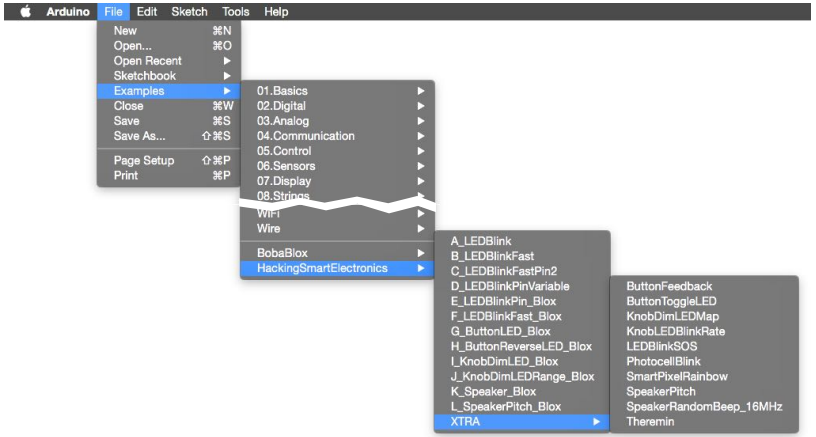
Thank You!

Robert Gallup bob@robertgallup.com @robertgallup

XTRA

Digital LED
Photocell
Trinket @ 16MHz
Resistor Values

The XTRA Sketches



DIGITAL LED

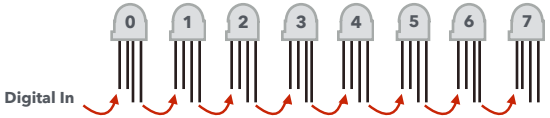
133

The Smart LED (Digital LED)

Unlike a standard LED, the Smart LED has a small processor inside that controls the light and communicates digitally with your microprocessor. They are also called NeoPixels. Generically, they are sometimes referred to by their "controller", the WS2812 or WS2811.

134

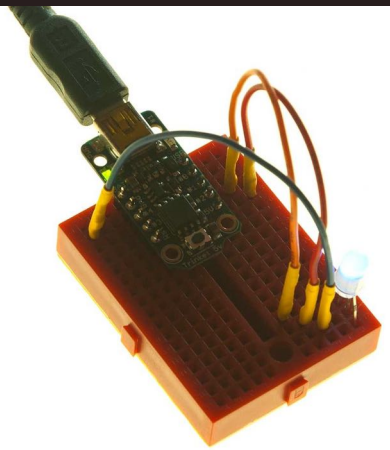
Component: Smart LED (NeoPixel)

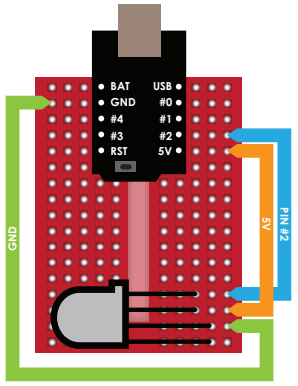


Multiple Smart LEDs can be "cascaded" from a singe pin on your microprocessor (but, you may need additional power).

135

Wiring Up a Smart LED (NeoPixel)





136

Hack: Install and Verify the Library

1. **Install the Adafruit_NeoPixel.zip library using the menu Sketch > Include Library > Add .ZIP Library...**
2. **Verify that Adafruit_NeoPixel appears under Examples in the File menu.**

Hack: The SmartPixel Example

1. **After wiring the SmartPixel up and installing the library, load and run the SmartPixel sketch in the XTRA examples.**
2. **Change the rate the colors change in the sketch**
3. **The sketch currently uses red, green, and blue as it's colors. Change to three different colors**
   Hint: look at the comments to figure out the parameters for the setPixelColor() method.

Hack: The SmartPixelRainbow Sketch

1. **Load and run the SmartPixelRainbow sketch.**
2. **Change the rate the colors change in the sketch.**
   Hint: You can guess what to do (perfectly respectable), or look at the more complicated code in the NeoPixelFunctions tab for a clue (your sketch can have multiple tabs, BTW).
3. **BIG STRETCH: See if you can figure out how to control the rainbow color with a knob.**
   Hint: look at the setPixelColor() and wheel() methods.

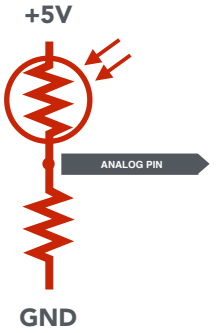
PHOTOCELL

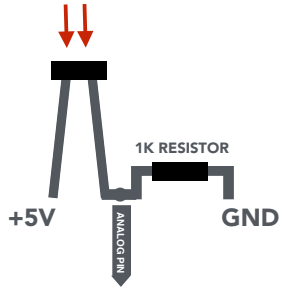
Photocell (Light Dependent Resistor)

A photocell changes its resistance depending on the amount of light that falls on it. As the light increases, the resistance decreases. Combined with another fixed resistor, it acts very much like a knob.

141

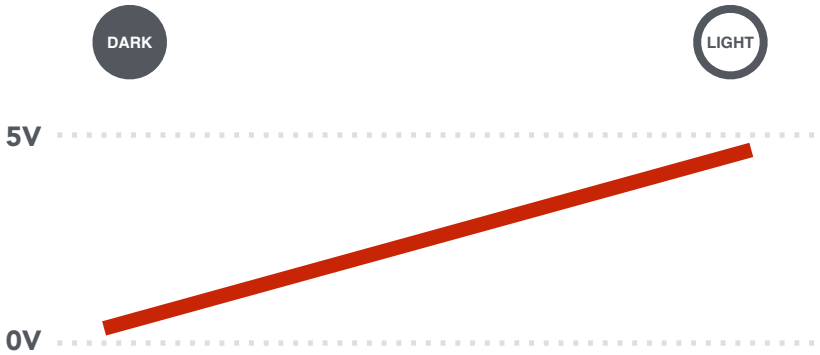
Component: Photocell





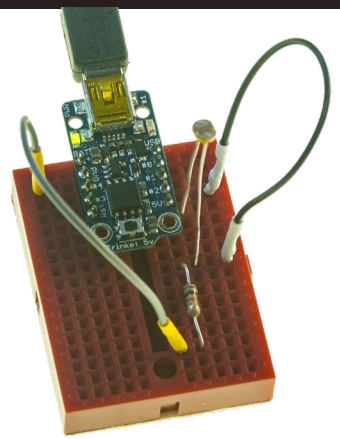
142

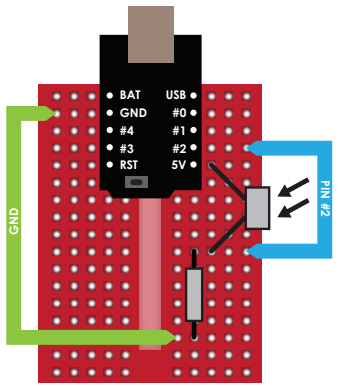
Photocell Voltage at Pin



143

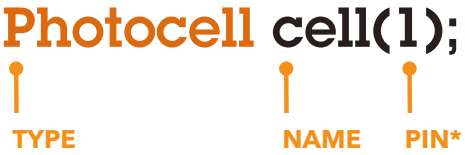
Wiring Up a Photocell





144

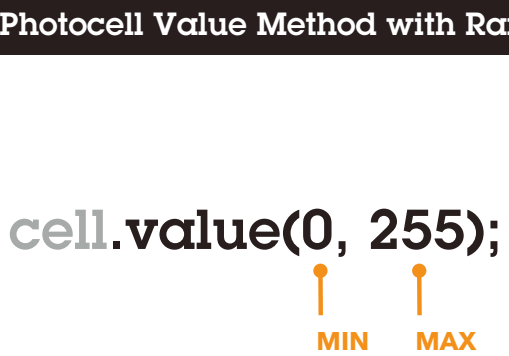
Photocell Declaration



* The photocell is an analog device. So, the pin is an analog pin. Analog pin #1 is *physical* pin #2.

Photocell Value Method

Photocell Value Method with Range



Hack: Run the PhotocellBlink sketch

1. **Build the photocell circuit and run the PhotocellBlink sketch from the XTRA workshop examples. What happens when you vary the light on the photocell?**
2. **Reverse the logic of the photocell (i.e. so the LED blink rate goes up when the light on the cell goes up.)**
3. **BIG STRETCH: write a speaker sketch that changes tone with the light on the photocell.**

TRINKET @ 16 MHz

149

The Trinket Can Go Faster (16 MHz)

By default, the Trinket has a clock that runs at 8 MHz (megahertz). But, the 5V Trinket is capable of running twice as fast at 16 MHz. Follow the steps below to run your Trinket at 16 MHz. (More information can be found at https://learn.adafruit.com/introducing-trinket/16mhz-vs-8mhz-clock.)

1. **At the beginning of your sketch, include:**
   ```
   #include <avr/power.h>
   ```
2. **In the Setup function, include:**
   ```
   if (F_CPU == 16000000) clock_prescale_set(clock_div_1);
   ```
3. **Choose Tools > Board > Adafruit Trinket 16 MHz**
4. **The XTRA example, SpeakerRandomBeep_16MHZ demonstrates this technique.**

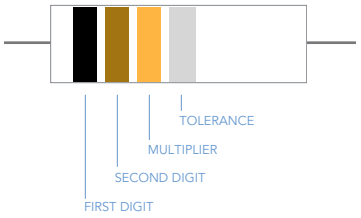
150

RESISTOR VALUES

151

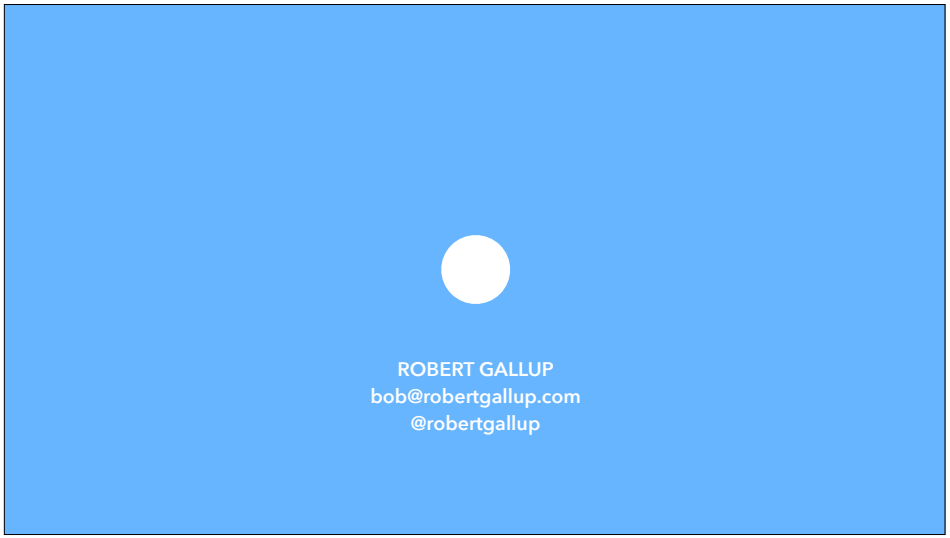
Identifying Resistor Values

COLOR	DIGIT 1	DIGIT 2	MULTIPLIER	TOLERANCE
BLACK	0	0	$x10^0$	0
BROWN	1	1	$x10^1$	±1%
RED	2	2	$x10^2$	±2%
ORANGE	3	3	$x10^3$	
YELLOW	4	4	$x10^4$	
GREEN	5	5	$x10^5$	±0.5%
BLUE	6	6	$x10^6$	±0.25%
VIOLET	7	7	$x10^7$	±0.1%
GRAY	8	8	$x10^8$	±0.05%
WHITE	9	9	$x10^9$	
GOLD			x0.1	±5%
SILVER			x0.01	±10%
-NONE-		9		±20%



Resistor values (ohms) are identified by colored bands painted around their body. The colors represent two digits followed by a multiplier. So, for example, the illustration above has a first digit=0 (black), the second digit=1 (brown), and the multiplier is x1000 (orange, 10^3). So the value is 01 x 1000 = 1000 ohms. The tolerance is indicated by the fourth band. It will be gold (±5%) or silver (±10%) for typical applications.

152


ROBERT GALLUP
bob@robertgallup.com
@robertgallup