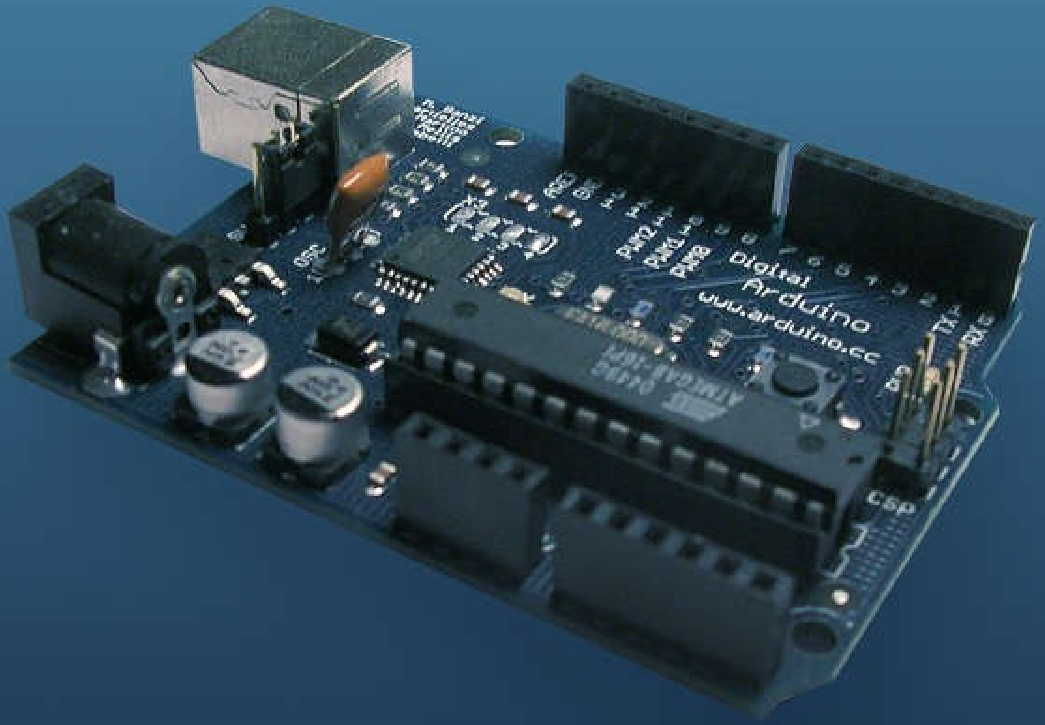
Arduino

Physical Computing I/O board



Avik Dhupar

Overview

What is Arduino?

- What is it used for?
- How to get started
- Demonstration

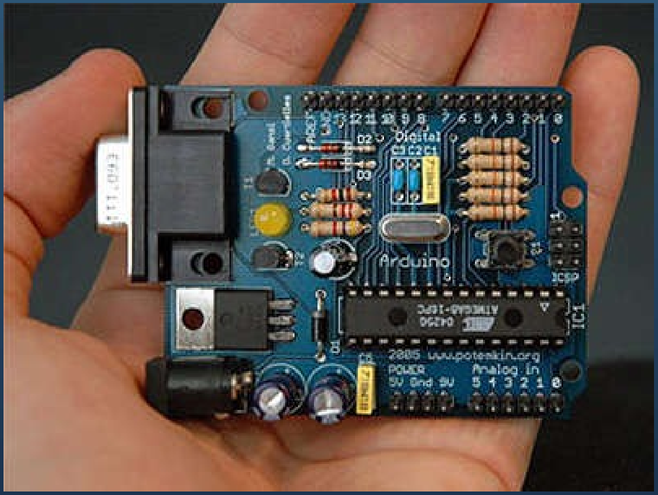
- Questions are welcome at any time.

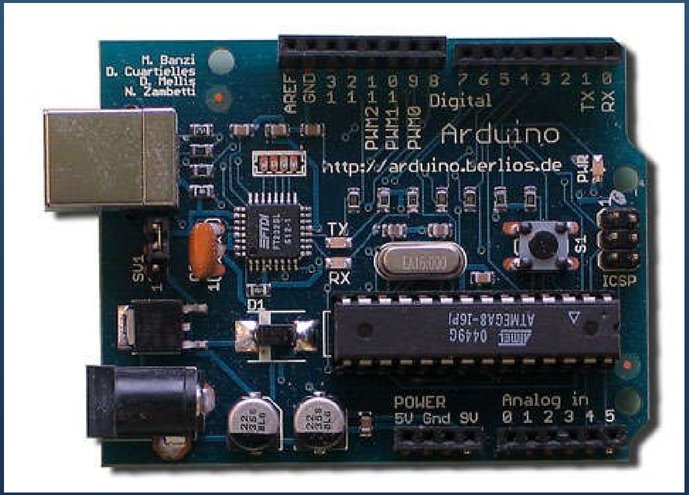
What is Arduino?

- *“Arduino is an open-source physical computing platform based on a simple i/o board and a development environment that implements the Processing / Wiring language. Arduino can be used to develop stand-alone interactive objects or can be connected to software on your computer.“* (www.arduino.cc, 2006)

Arduino is a platform

- A physical Input / Output board (I/O) with a programmable Integrated Circuit (IC).

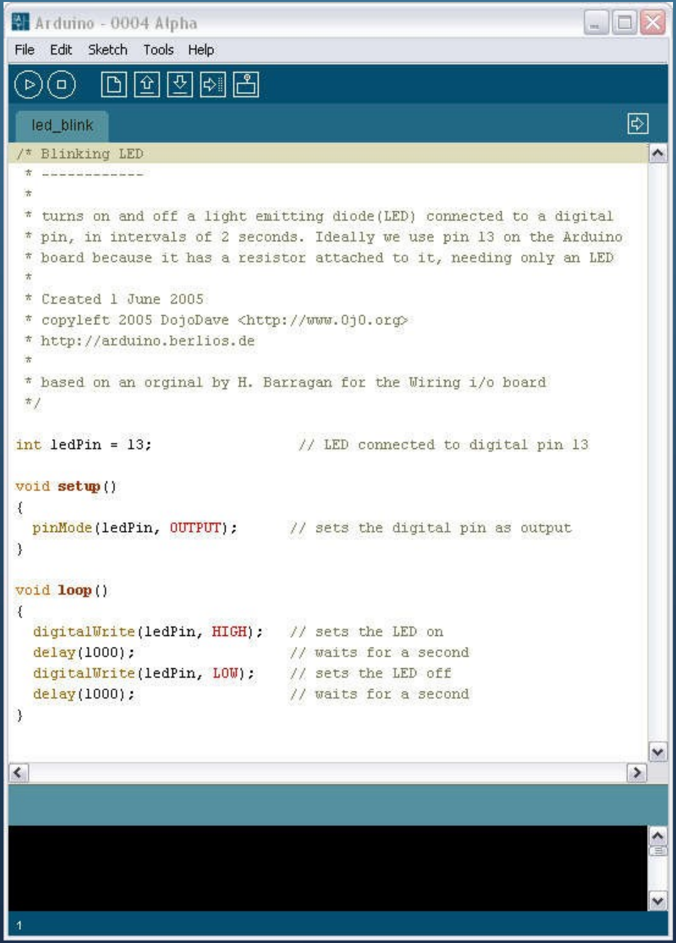




Arduino is a platform

- Also including an Integrated Development Environment (IDE) for programming.

- The language itself is based in C but is largely modeled upon the www.processing.org language.



```
/* Blinking LED
 * ------------
 *
 * turns on and off a light emitting diode(LED) connected to a digital
 * pin, in intervals of 2 seconds. Ideally we use pin 13 on the Arduino
 * board because it has a resistor attached to it, needing only an LED
 *
 * Created 1 June 2005
 * copyleft 2005 DojoDave <http://www.0j0.org>
 * http://arduino.berlios.de
 *
 * based on an orginal by H. Barragan for the Wiring i/o board
 */

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

void setup()
{
  pinMode(ledPin, OUTPUT);      // sets the digital pin as output
}

void loop()
{
  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
}
```

What is it used for?

- Physical Computing projects / research
- Interactive Installations
- Rapid prototyping

- When you wish to move beyond the traditional Mouse, Keyboard and Monitor to develop novel and custom interactions in your project work.

What can it do?

- Sensors (to sense stuff)
  - Push buttons, touch pads, tilt switches.
  - Variable resistors (eg. volume knob / sliders)
  - Photoresistors (sensing light levels)
  - Thermistors (temperature)
  - Ultrasound (proximity range finder)
- Actuators (to do stuff)
  - Lights, LED’s
  - Motors
  - Speakers
  - Displays (LCD)

Why Arduino?

- It is Open Source, both in terms of Hardware and Software.
- It is cheap(1300रु), the hardware can be built from components or a prefab board can be purchased for approx 900रु.
- It can communicate with a computer via serial connection over USB.
- It can be powered from USB or standalone DC power.

Why Arduino?

- It can run standalone from a computer (chip is programmable) and it has memory (a small amount).
- It can work with both Digital and Analog electronic signals. Sensors and Actuators.
- You can make cool stuff! Some people are even making simple robots, and we all know robots are just cool. ☺

How to get started

- You’ll need a board of course, along with the USB cable and DC power supplies.
- Read about, understand what you are working with and download the IDE: http://www.arduino.cc
- Mac, Windows and Penguin friendly versions available
- Then you are ready to plug it in!

Not so fast!

- It’s important to note at this stage that Arduino’s are electronic devices.
- This means you MUST consider electrical safety and understand the basics before diving straight in.
- The board itself doesn’t operate at what would normally be considered dangerous Voltages or Current, but if in doubt at any stage of use you should seek more expert advice.

Basic Electrical knowledge

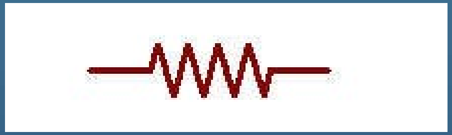
- At the moment we don’t have any electrical engineers so we need to do some ground work ourselves.
- A fantastic guide to electronics in theory, practice and of course safety is available as a PDF at:

  http://www.ibiblio.org/obp/electricCircuits/

  What you want is Volume 1, DC circuits. This will help you greatly in understanding how to wire circuits when using sensors and actuators.

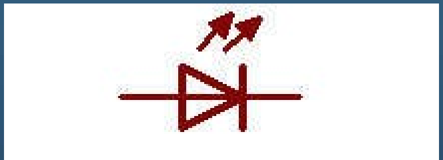
Basic Electrical knowledge

Each electronic component has a schematic symbol, which is a simplified drawing of the part. For resistors the symbol looks like this:

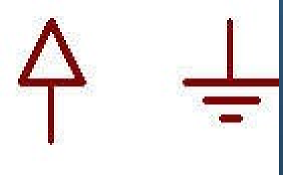


Resistor symbol

And the symbol for LED's look like this:



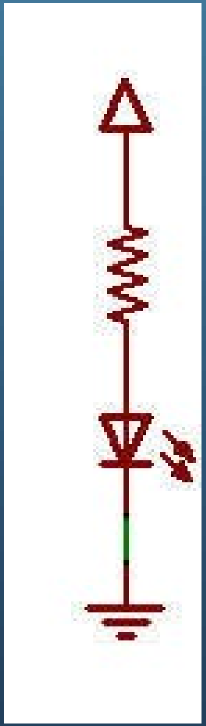
LED symbol, positive pin on the left, negative pin on the right



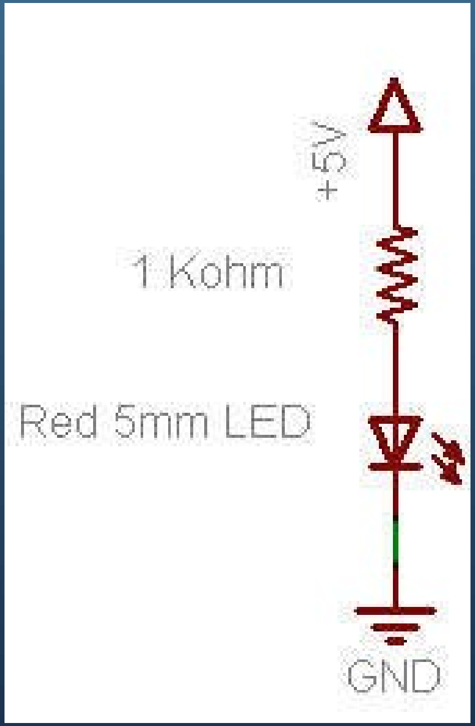
Power and Ground symbols

You can see that the resistor symbol is symmetric, just like resistors themselves. The LED symbol, however, has an arrow thing going on. This is the direction in which current flows. The little arrows that are coming out of the symbol indicate that this is a diode that emits light.

Basic Electrical knowledge



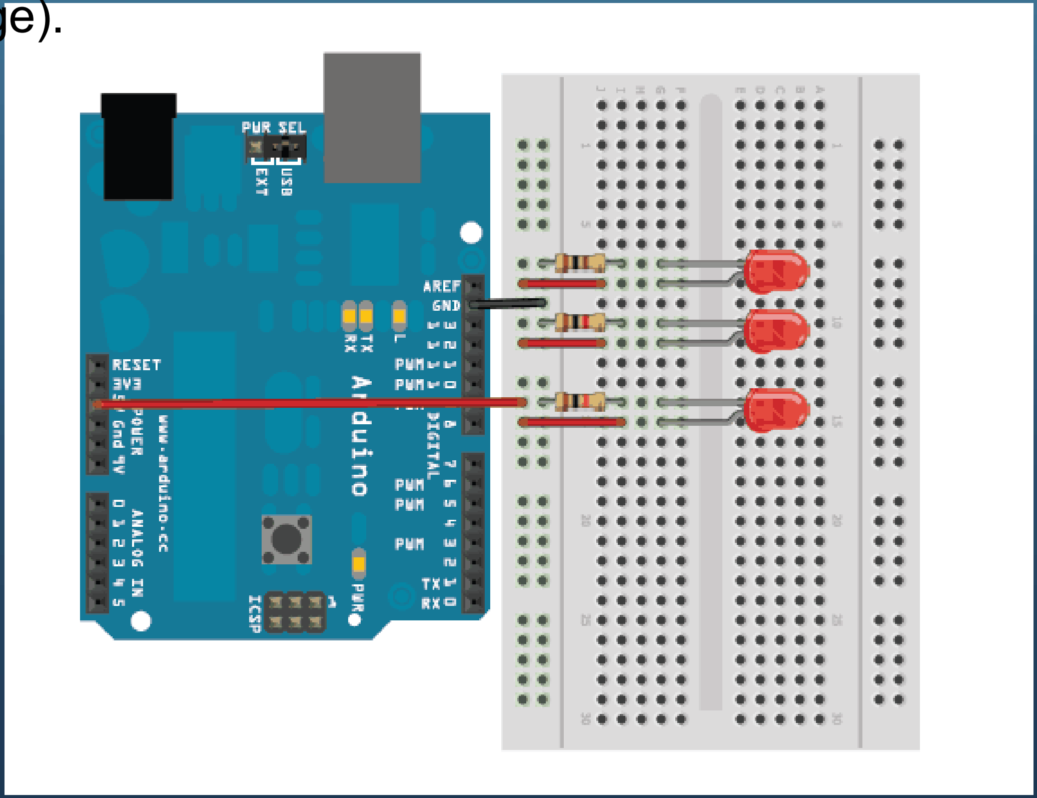
The only thing we need to do now is indicate how the LED and resistor are hooked up and show the 5V and ground connections.



Next to symbols, we often write important information like what the resistor value is, what color and size the LED should be, and the voltage associated with the power supply.

Quiz!

LED #1 has a 100 ohm resistor (Brown Black Brown)
LED #2 has a 1.0K (Brown Black Red)
LED #3 has a 10K (Brown Black Orange).



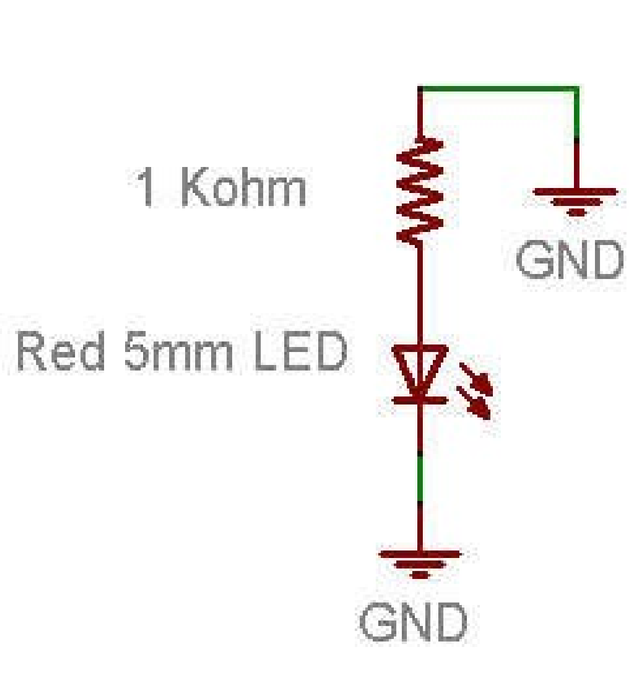
Which LED is brightest?

Which LED is dimmest ?

If we had an LED with a resistor that was 5K ohms, which LED would it be brighter than? Which LED would it be dimmer than?

A Quick Rewiring...

We're going to make a very small modification to our wired up circuit

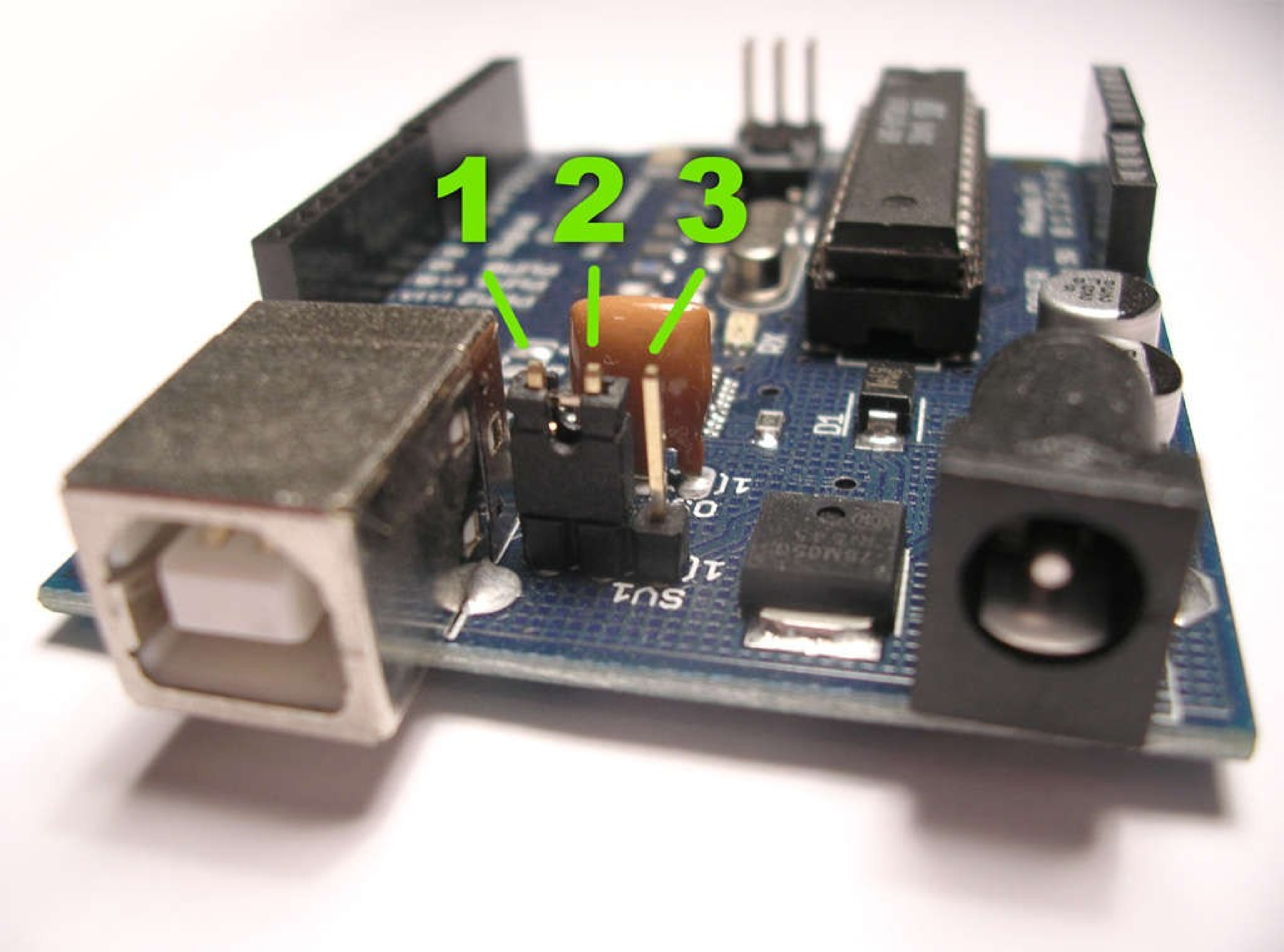


Result?

Getting up and running

- The power mode must be selected before you plug the board into anything.

- When powering from the USB cable (5 volts) the jumper should be closest to the USB input, for DC supply the jumper should be closest to the DC input.



1
2
3

Getting up and running

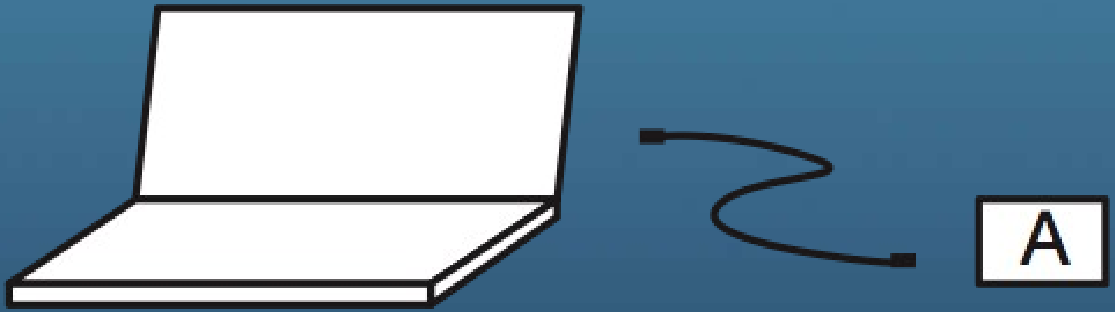
- Plug it into the USB port and install the USB drivers that come with the IDE.
- Open up the Arduino IDE and select the COM port, usually COM1 or COM2 on a Windows machine.
- Within the IDE, select the BAUD rate (communication speed for serial connections)
- Set BAUD rate to 9600 on Windows, (?) for Mac?

Basic Process

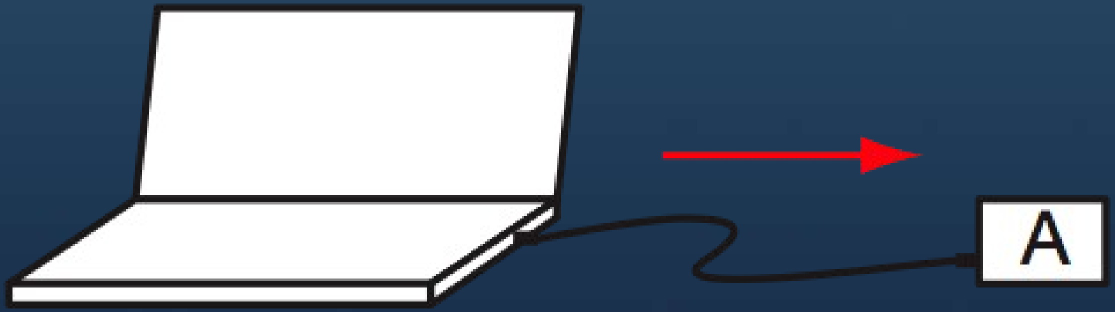
- Design the circuit:
  - What are electrical requirements of the sensors or actuators?
  - Identify inputs (analog inputs)
  - Identify digital outputs
- Write the code
  - Build incrementally
    - Get the simplest piece to work first
    - Add complexity and test at each stage
    - Save and Backup frequently
  - Use variables, not constants
  - Comment liberally

Writing and Downloading Code

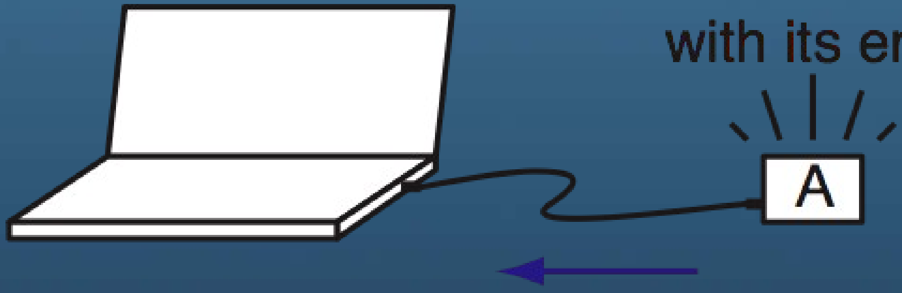
Write sketch on PC



Download sketch to Arduino



Running Code While Tethered

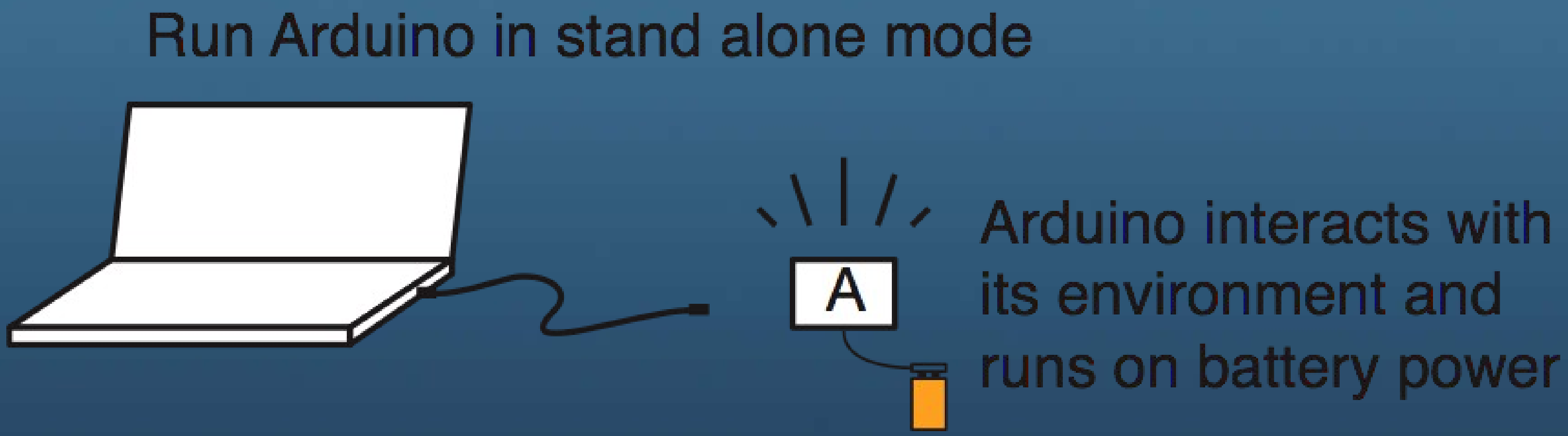


Run sketch on Arduino
and send data back to PC

Arduino interacts
with its environment

Serial communication
back to host

Running Code Stand-Alone



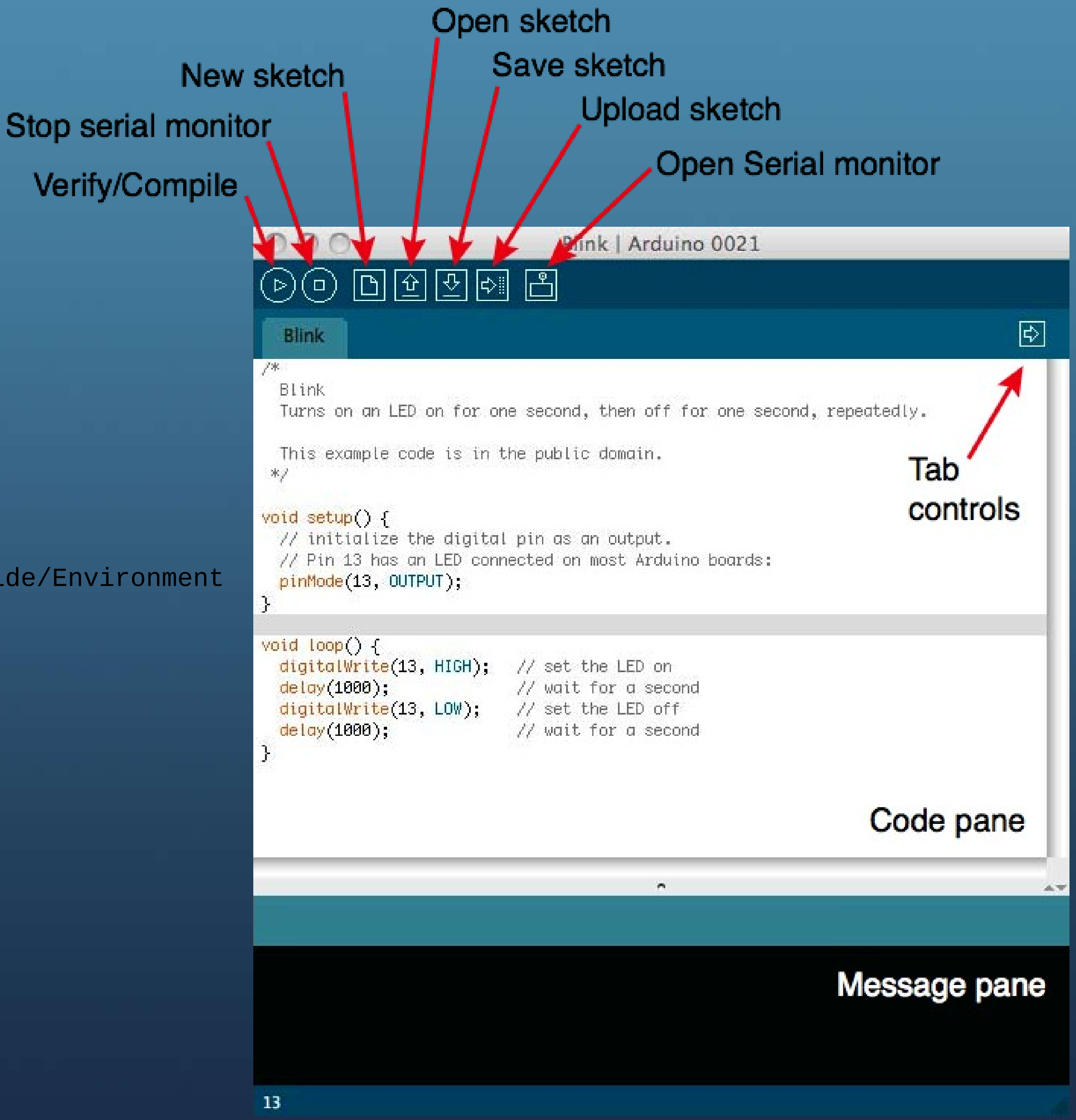
Run Arduino in stand alone mode

Arduino interacts with its environment and runs on battery power

Arduino IDE

IDE = Integrated Development Environment

http://www.arduino.cc/en/Guide/Environment



```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  delay(1000);              // wait for a second
}
```

Overview of The C Programming Language

Example 1

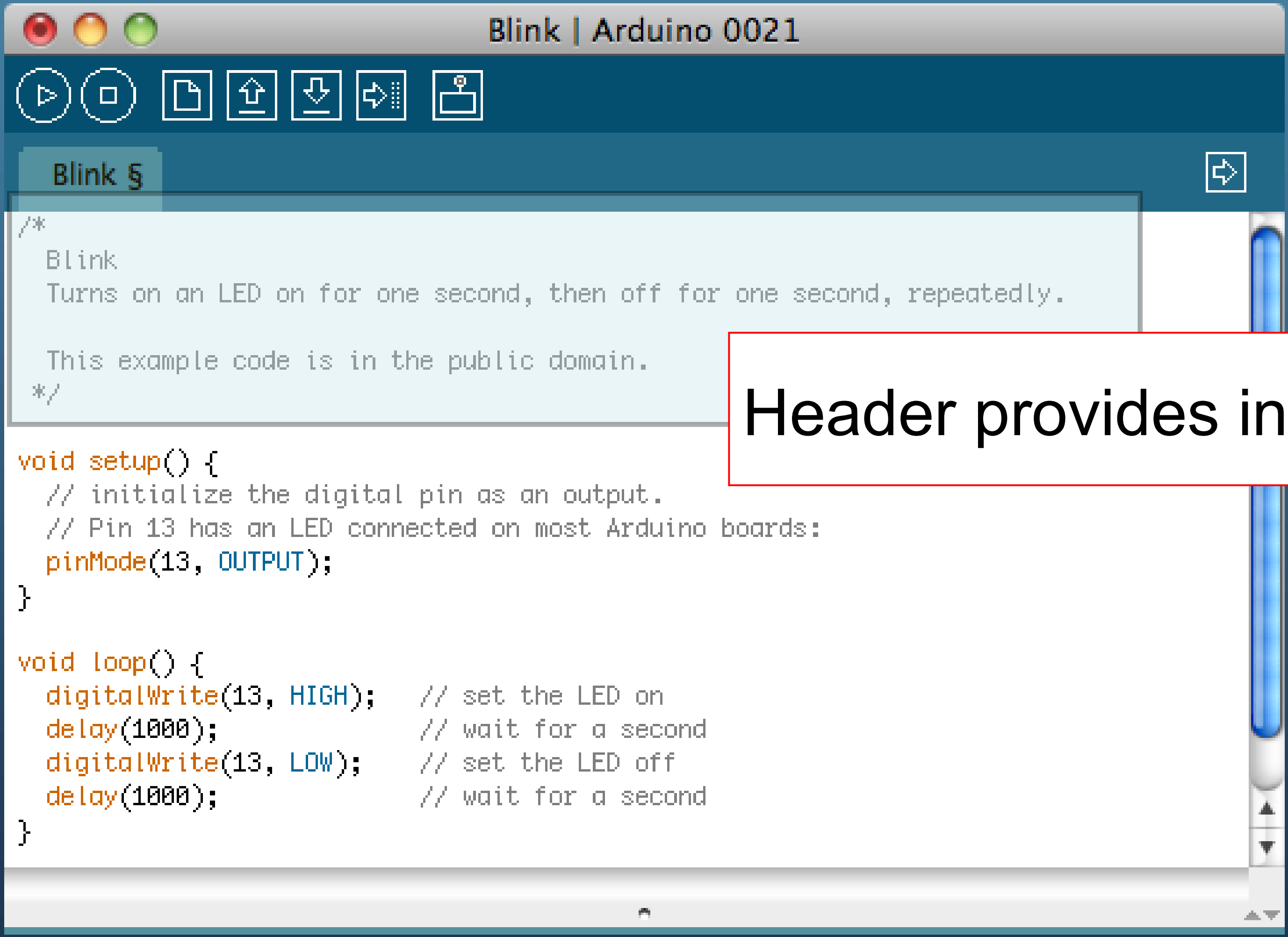
Demonstration

Start up the Arduino software and open up the **Blink** sketch.

For the most basic kind of program you'll need a simple actuator, an LED with the long leg (+) pushed into pin 13 and the short leg (-) in the adjacent ground pin (GND). Pin 13 is special, in the sense that it has a built in resistor to correctly control the voltage going into a testing LED just like this.

Example 1

Code Structure: Header

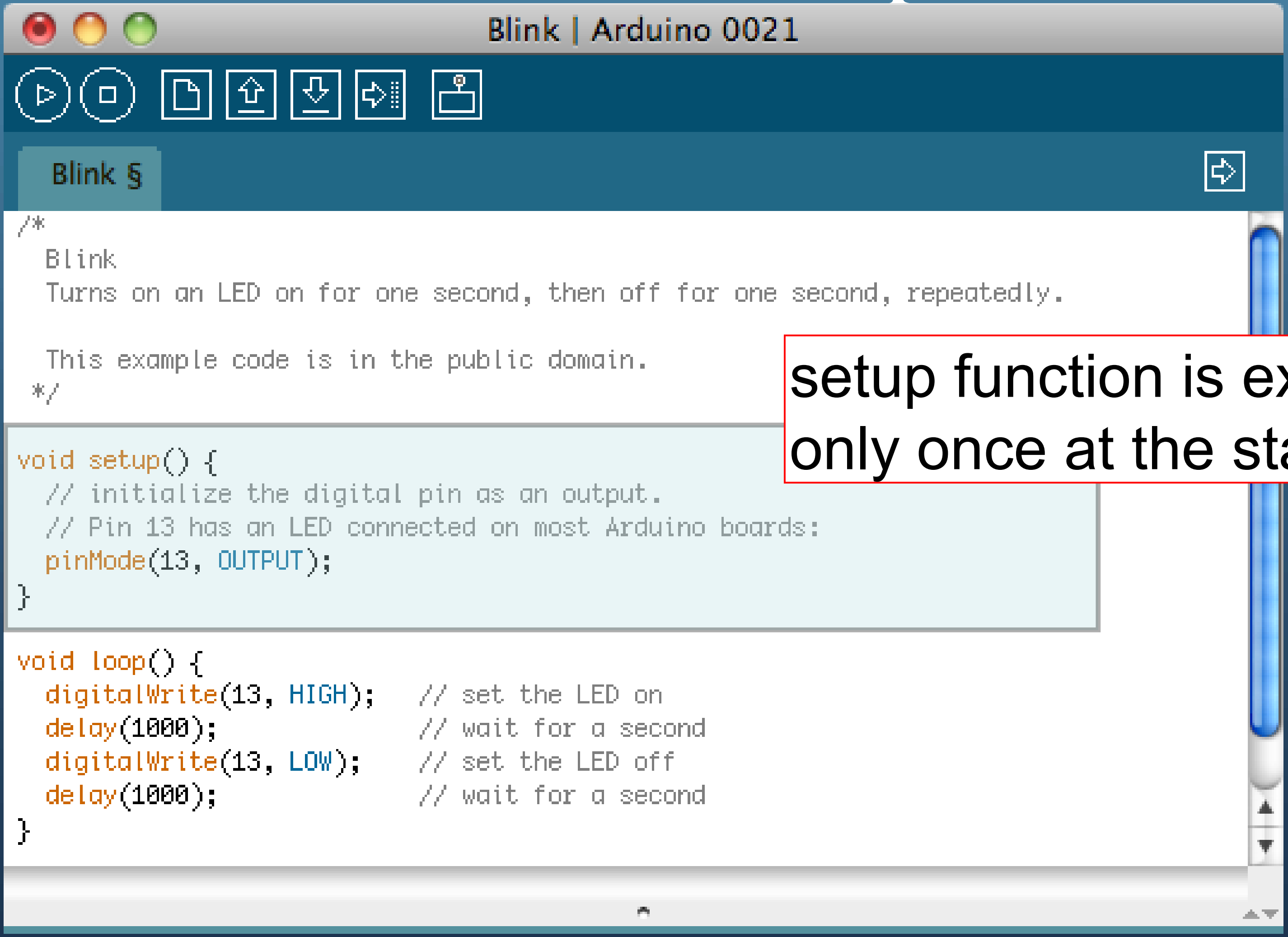


```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  delay(1000);              // wait for a second
}
```

Header provides information

Example 1

Code Structure: setup function

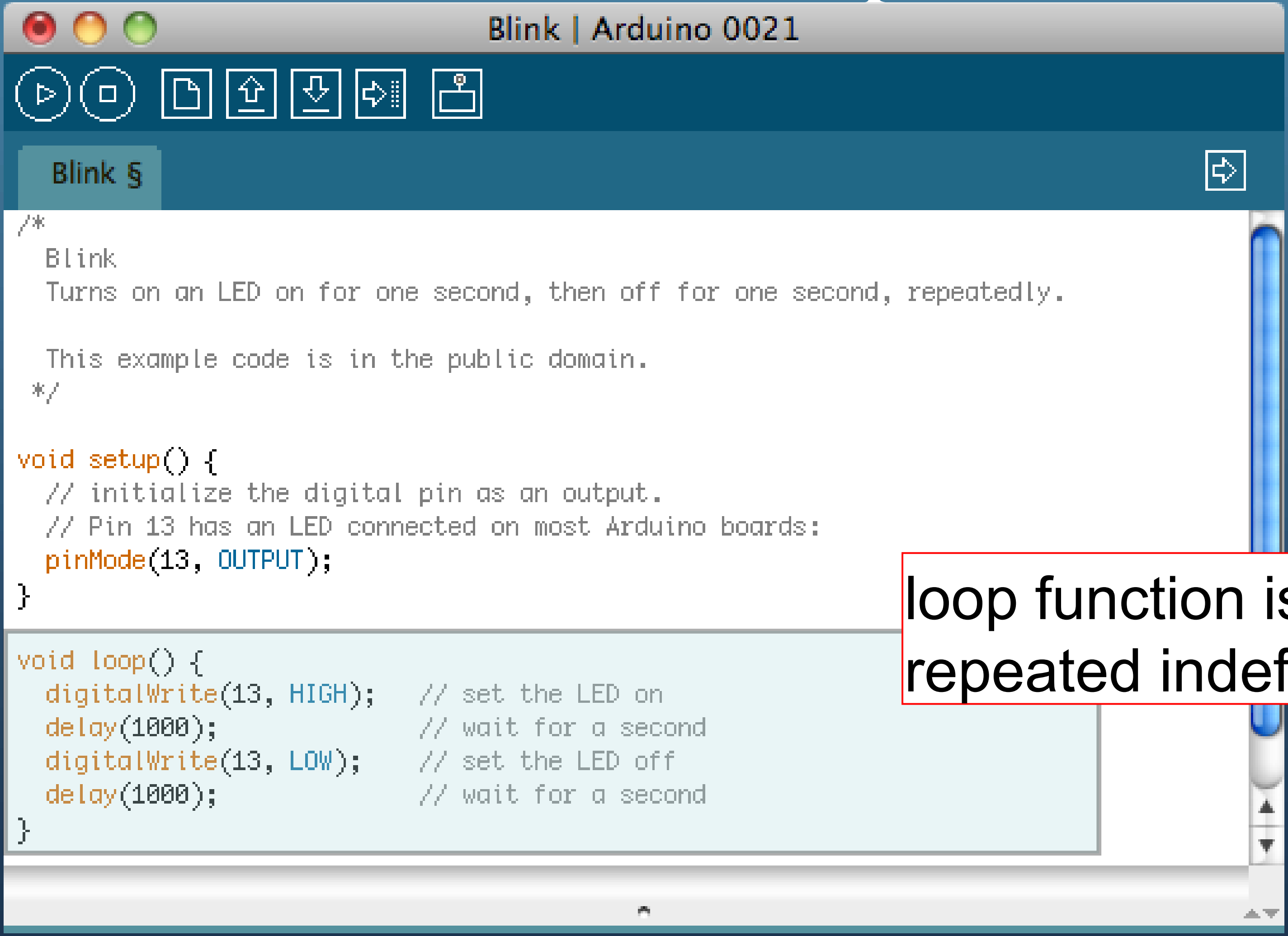


```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  delay(1000);              // wait for a second
}
```

setup function is executed only once at the start

Example 1

Code Structure: loop function

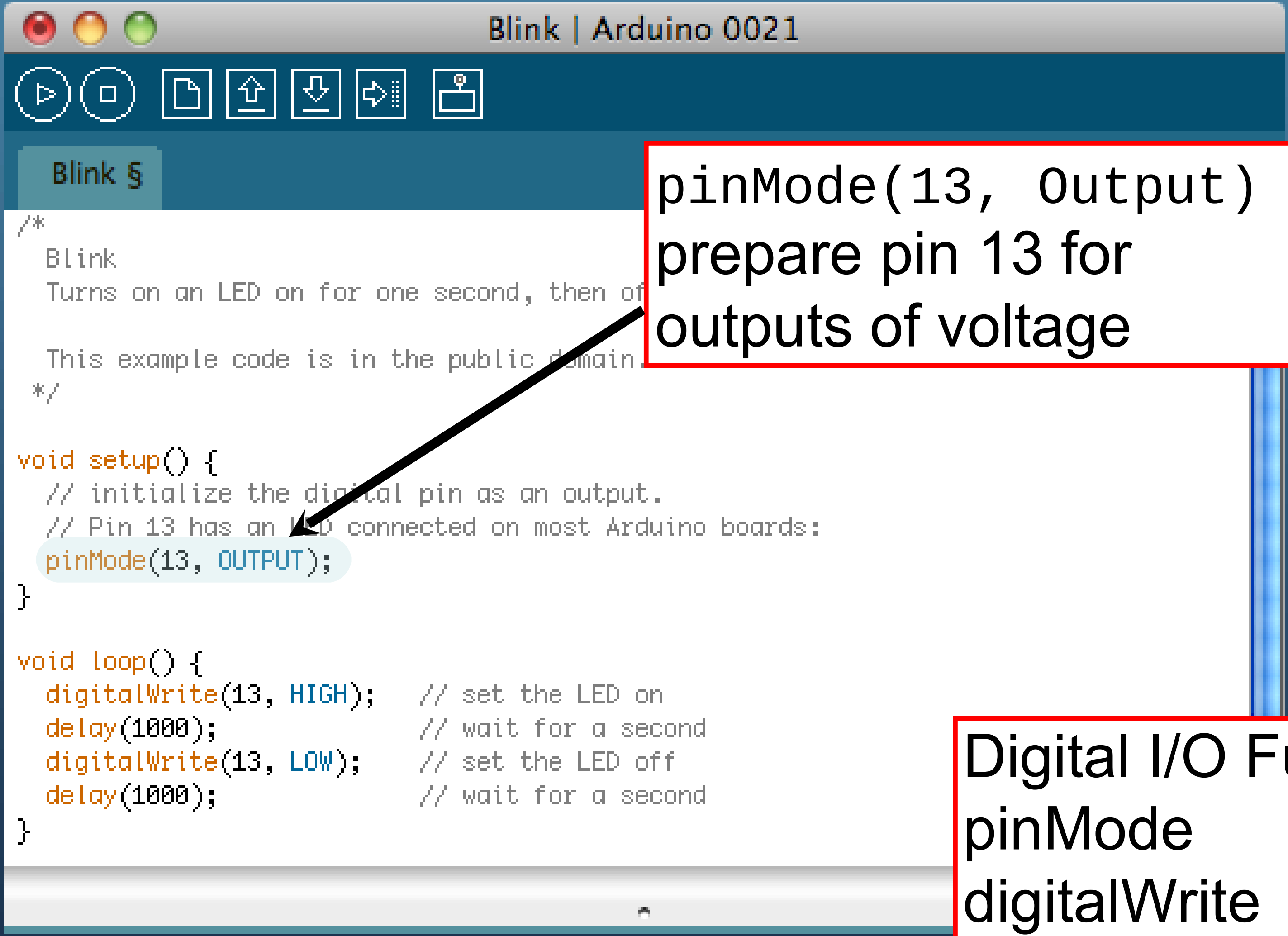


```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  delay(1000);              // wait for a second
}
```

loop function is repeated indefinitely

Example 1

Code



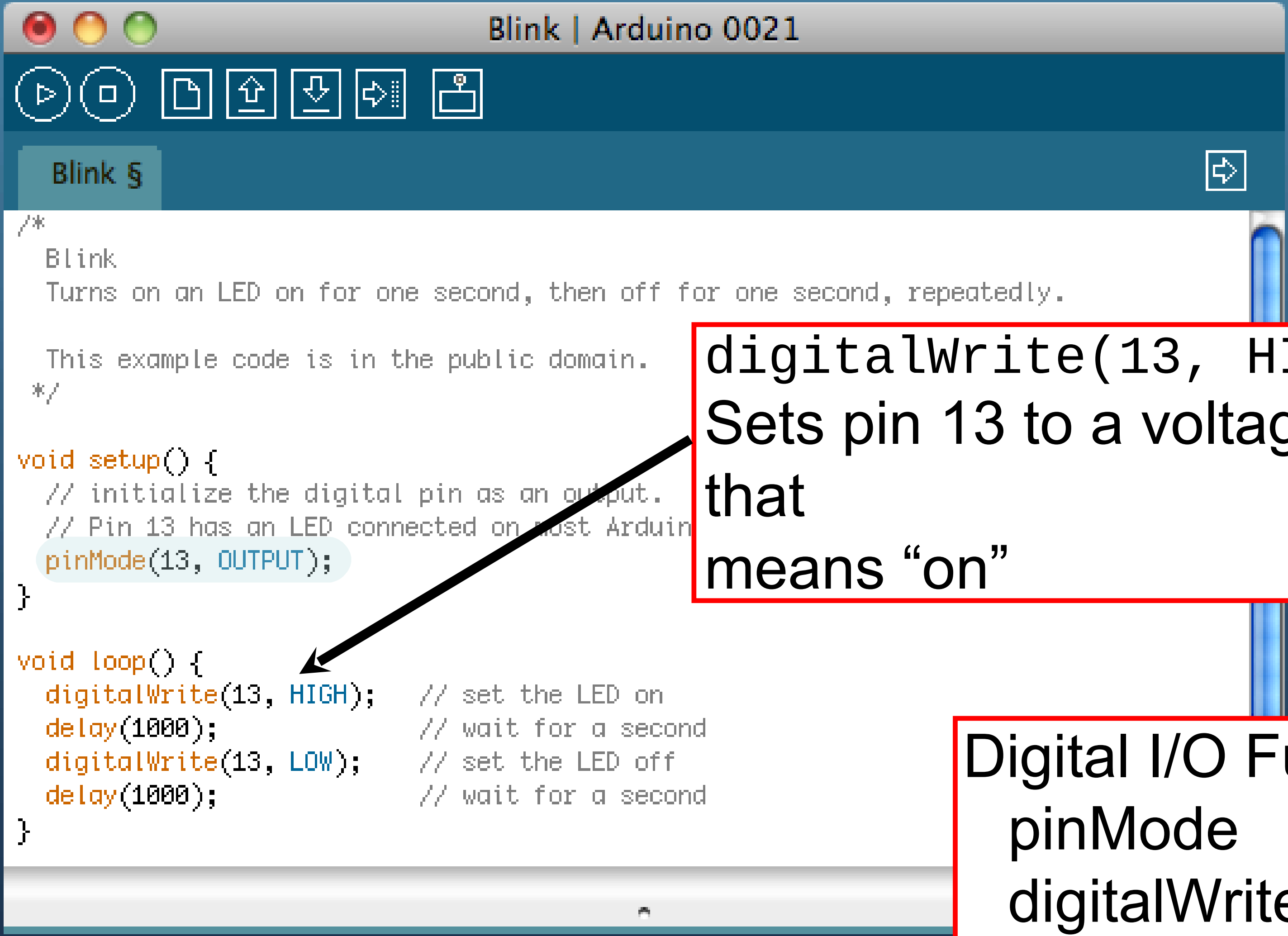
```
/*
  Blink
  Turns on an LED on for one second, then of

  This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  delay(1000);              // wait for a second
}
```

`pinMode(13, Output)`
prepare pin 13 for outputs of voltage

Digital I/O Functions:
pinMode
digitalWrite
digitalRead

Example 1

Code



```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduin
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  delay(1000);              // wait for a second
}
```

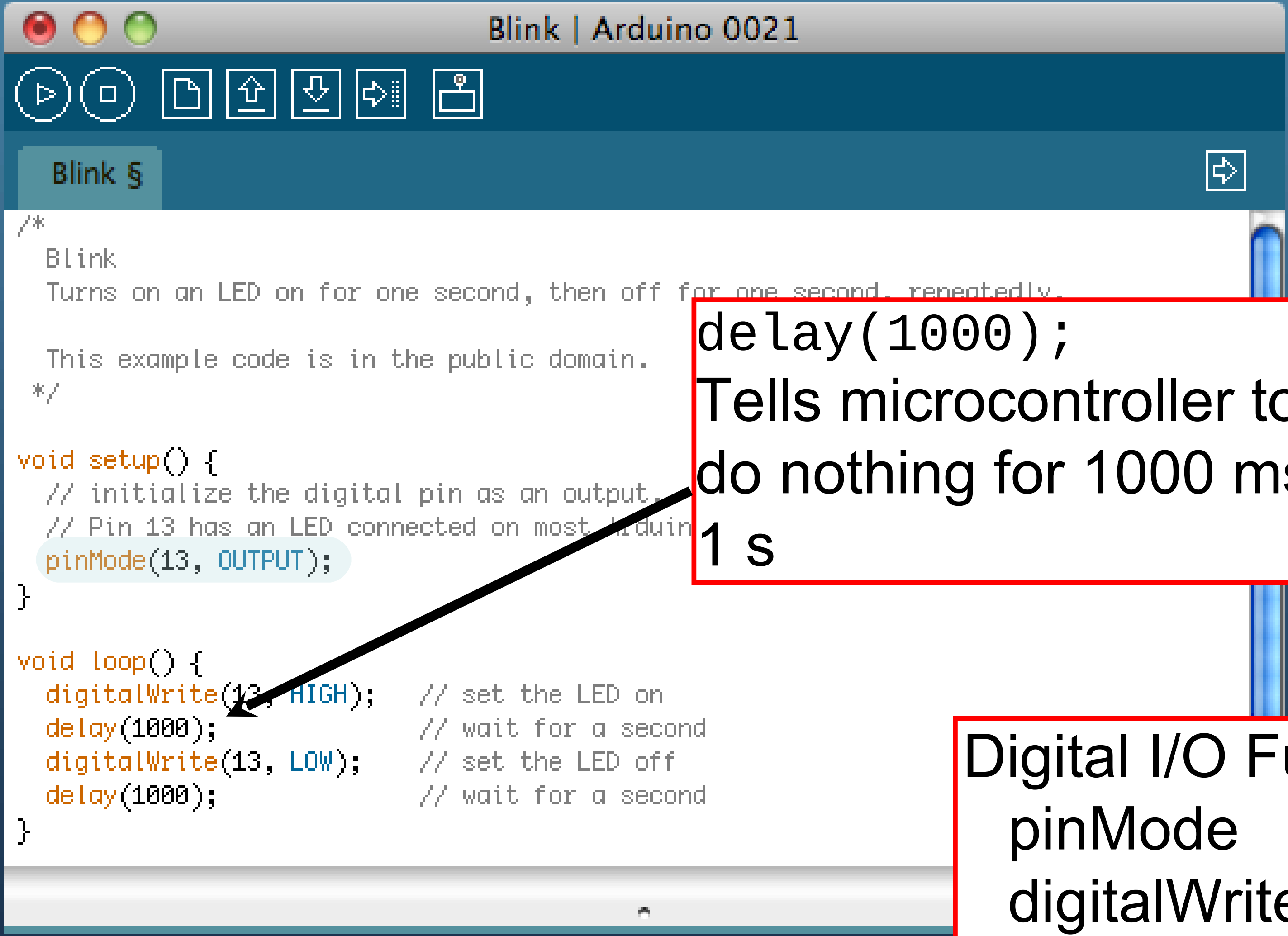
`digitalWrite(13, HIGH)`
Sets pin 13 to a voltage
that
means “on”

Digital I/O Functions:

- pinMode
- digitalWrite
- digitalRead

Example 1

Code



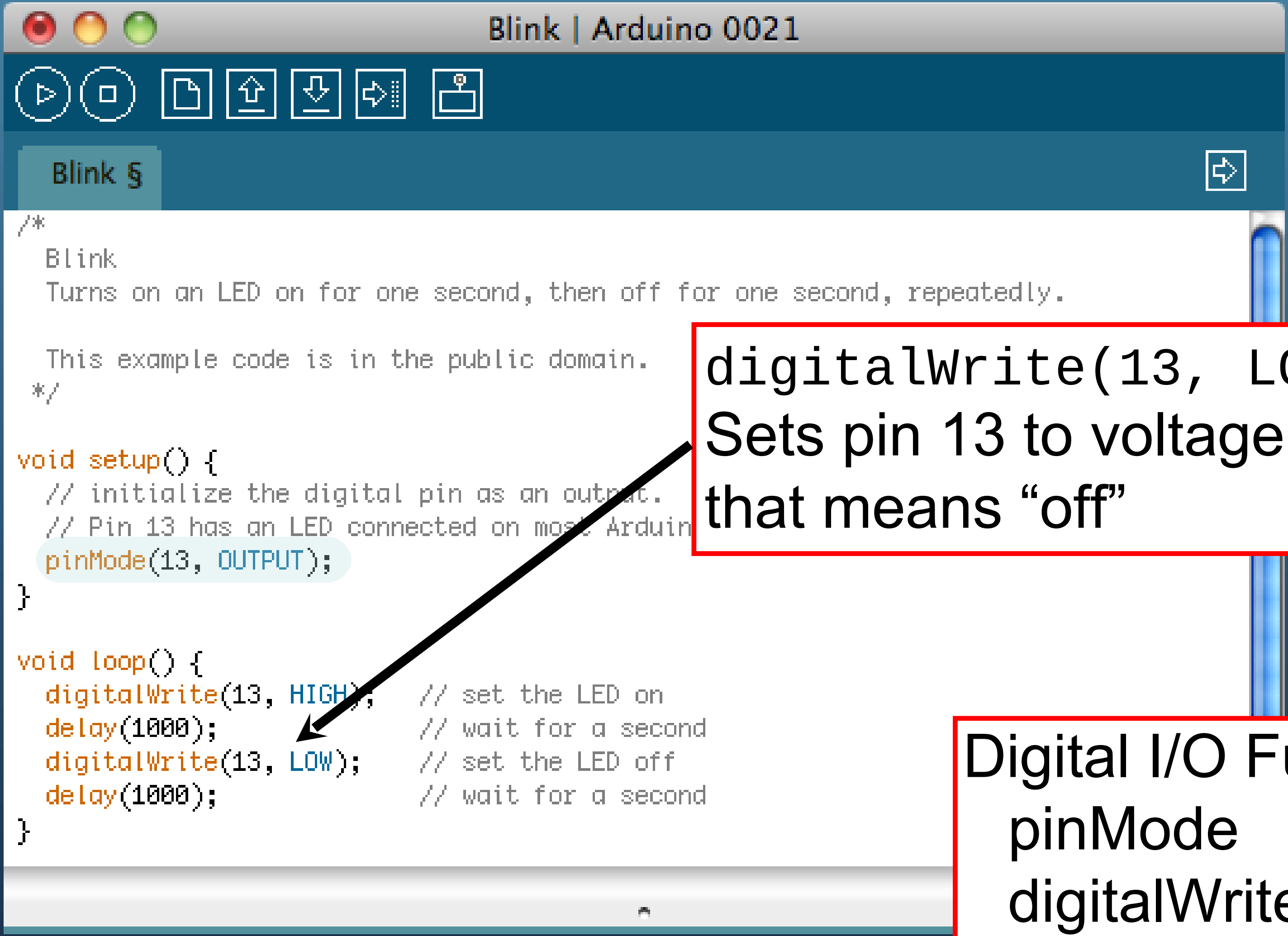
```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduin
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  delay(1000);              // wait for a second
}
```

`delay(1000);`
Tells microcontroller to do nothing for 1000 ms = 1 s

Digital I/O Functions:
- pinMode
- digitalWrite
- digitalRead

Example 1

Code



```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduin
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  delay(1000);              // wait for a second
}
```

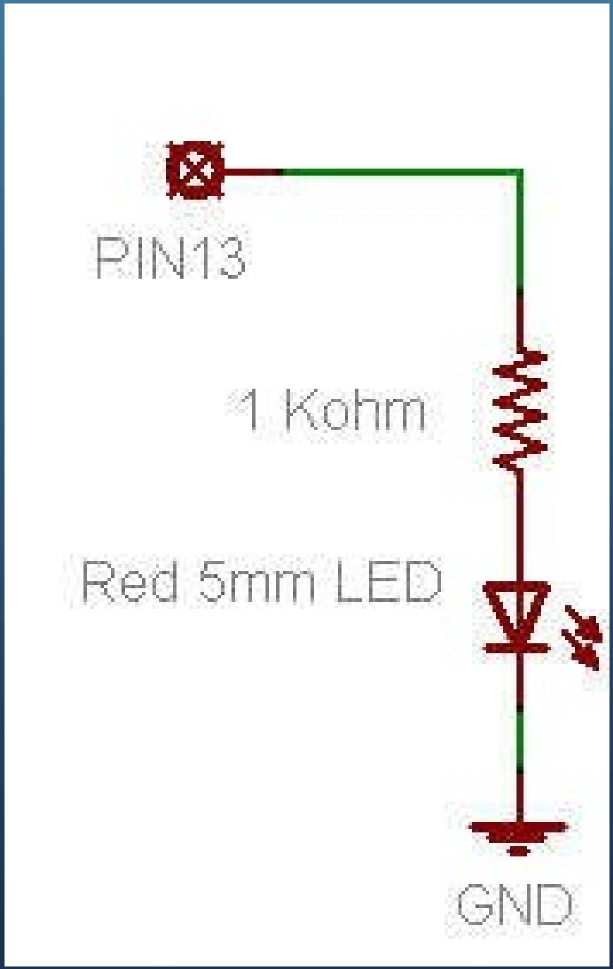
`digitalWrite(13, LOW)`
Sets pin 13 to voltage that means "off"

Digital I/O Functions:

- pinMode
- digitalWrite
- digitalRead

Example 1

The connections



Upload a program

- At this stage we just programmed the LED to blink on and off at a set time interval.

- Press the reset button on the board and then click ‘Upload to I/O board’ in the IDE. If all goes well lights should flicker on the board and the IDE will confirm success.

MORE
FUN
AHEAD

Example 2

Demonstration

Robot controlled by a TV remote

Example 3

Demonstration

RFID based wireless login system

Conclusion

- There’s heaps more to explore, I’ve barely scratched the surface of what you can really do with this technology.

- Make it happen. If you want to use an Arduino in your project you’ll have to be proactive about it and do the research yourself. It really isn’t hard, just get stuck in.

¿ Questions ?

- Use the website and associated forum for tutorials, code samples and general help:

www.arduino.cc

- Find my PPTs at

www.slideshare.net/avikdhupar

- Find me at

www.facebook.com/avikd

- Or drop an email

avik.tk@gmail.com

- Find my videos at

http://vimeo.com/avikd/videos

Thank you

☺