LET'S CODE BLACKSBURG!

STEM Educator's “Teach the Teacher” Arduino Workshop

For: STEM Educators of SW VA
By: Thomas “Tweeks” Weeks
2016-08-1 & 3

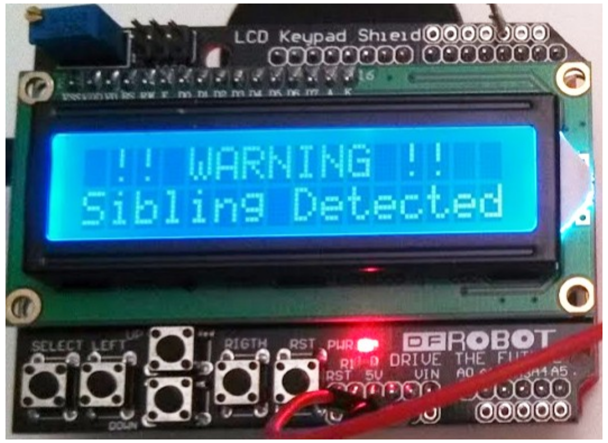
www.letscodeblacksburg.org

Day-1 Overview

- About Let's Code Blacksburg!
- About Teaching STEM Using The Arduino
- Jumping In – Arduino Overview & Running “Blink” Recipe
- Teaching Programming Theory (C++)
- Inputs vs Outputs
- Meal → “Output: Blink” + “INPUT: Button”
- Meal - Alarm On/Off → Combine Button + Sound Generation
- Meal - Proximity Alarm → Add Ultrasonic Range Sensor
- Arduino Resources

Day-2 Overview

- More C++ Programming Theory: Functions & Parameters
- Build – “Sibling Detector Alarm” using three recipes:
    - OUTPUT – Sound Buzzer
    - INPUT – People Motion Sensor
    - INPUT/OUTPUT – LCD Keypad Shield



- Teaching Robotics Based Workshops
- Arduino Resources
- Let's Code Blacksburg Resources

Day-1 Overview

- About Let's Code Blacksburg!
- About Teaching STEM Using The Arduino
- Jumping In – Arduino Overview & Running “Blink” Recipe
- Teaching Programming Theory (C++)
- Inputs vs Outputs
- Meal → “Output: Blink” + “INPUT: Button”
- Meal - Alarm On/Off → Combine Button + Sound Generation
- Meal - Proximity Alarm → Add Ultrasonic Range Sensor
- Arduino Resources

STEM Educator's Arduino Workshop

About Me

Name: Thomas “Tweeks” Weeks

Position: Rackspace Technical Community Evangelist

Started: Sept 1999 (as employee#12, current #2)

Contact Info:

tweeks@rackspace.com GPG - 1024D/750152F
https://fb.com/ThomasW.Weeks OTR - tomweeks@gmail.com
@tweeks_tx

STEM Educator's Arduino Workshop

About Me

Name: Thomas “Tweeks” Weeks

Position: Rackspace Technical Community Evangelist

Started: Sept 1999 (as employee#12, current #2)

Various Jobs Over 16yrs:

-Sys-Admin / Engineer, Corp. Technical Trainer
-Product Engineer, Systems Architect
-Customer Advocate Engineer, Author / Tech Writer
-Data Center RCA Team Lead, Cloud RCA Lead
-Tech Community Relations, Training & STEM Outreach

STEM Educator's Arduino Workshop

About Me

Name: Thomas “Tweeks” Weeks

Position: Rackspace Technical Community Evangelist

Started: Sept 1999 (as employee#12, current #2)

My Passions

STEM Educator's Arduino Workshop

About Me

Name: Thomas “Tweeks” Weeks

Position: Rackspace Technical Community Evangelist

Started: Sept 1999 (as employee#12, current #2)

My Passions

redhat.
CERTIFIED
ENGINEER

STEM Educator's Arduino Workshop

About Me

Name: Thomas “Tweeks” Weeks

Position: Rackspace Technical Community Evangelist

Started: Sept 1999 (as employee#12, current #2)

My Passions

redhat.
CERTIFIED
ENGINEER

STEM Educator's Arduino Workshop

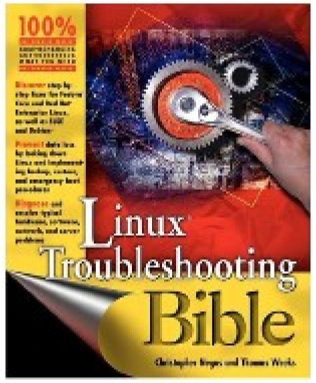
About Me

Name: Thomas “Tweeks” Weeks

Position: Rackspace Technical Community Evangelist

Started: Sept 1999 (as employee#12, current #2)

My Passions



STEM Educator's Arduino Workshop

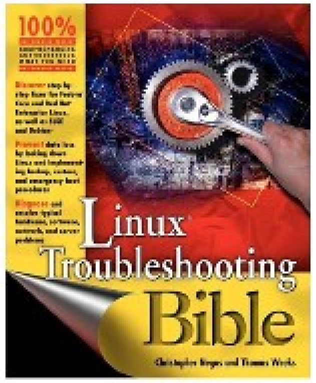
About Me

Name: Thomas “Tweeks” Weeks

Position: Rackspace Technical Community Evangelist

Started: Sept 1999 (as employee#12, current #2)

My Passions



100%

Linux Troubleshooting Bible

STEM Educator's Arduino Workshop

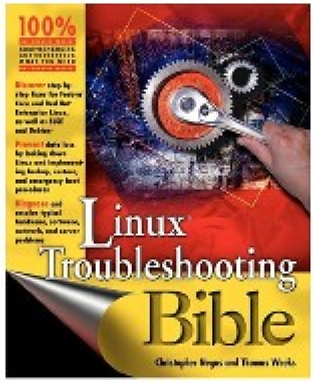
About Me

Name: Thomas “Tweeks” Weeks

Position: Rackspace Technical Community Evangelist

Started: Sept 1999 (as employee#12, current #2)

My Passions



STEM Educator's Arduino Workshop

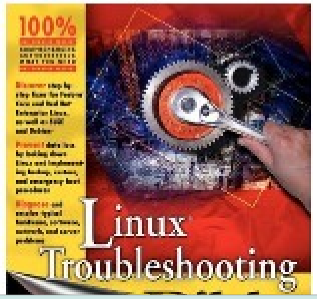
About Me

Name: Thomas “Tweeks” Weeks

Position: Rackspace Technical Community Evangelist

Started: Sept 1999 (as employee#12, current #2)

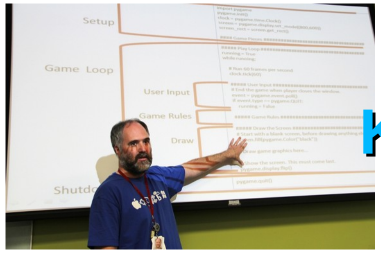
My Passions



HP Rocketry – www.nrvr.org

Kids & STEM – www.letscodeblacksburg.org

WHO are Let's Code Blacksburg's Trainers

LCBB Trainer	Teaches Workshops On
 Kevin Richey	pygame
Eddie Sheffied homas Weeks Monta Elkins	ARDUINO™ RaspberryPi
Nathan Liles	MINECRAFT

WHY Let's Code Teaches STEM

Our Vision

“Fostering the growth of our technology community by providing not-for-profit, volunteer based, STEM workshops.”

WHAT and HOW We Do It

WHAT:

HOW:

WHAT and HOW We Do It

WHAT: Share and Teach,

HOW:

WHAT and HOW We Do It

WHAT: Share and Teach,

HOW: Connect and Grow (the community)

WHAT and HOW We Do It

WHAT: Share and Teach,

by creating opportunities for locals (from kids,
to tech-learners and IT professionals) to learn new technologies and programming languages.

HOW: Connect and Grow

WHAT and HOW We Do It

WHAT: Share and Teach,

by creating opportunities for locals (from kids, to tech-learners and IT professionals) to learn new technologies and programming languages.

HOW: Connect and Grow

Connect local IT experts from our technical community to community-learners to facilitate the sharing of their knowledge into community – and by doing so grow the technical creative capacity of our community.

WHO Supports Us In The Community

rackspace®
the #1 managed cloud company

TEKsystems®
Our people make IT possible.

RB
TC

APEX
Systems

VirginiaTech
VT Engage
The Community Learning Collaborative
action · reflection · knowledge

WHAT Let's Code Teaches

Classes on

pygame

RaspberryPi

ARDUINO

MINECRAFT

Teach Languages

Python

C++

Javascript

WHO Let's Code Teaches & Reaches

Who We Target

What We Call Workshops

WHO Let's Code Teaches & Reaches

Who We Target	What We Call Workshops
Kids	Kids Series

WHO Let's Code Teaches & Reaches

Who We Target	**What We Call Workshops**
Kids	Kids Series
Technical Learners	General / Learner Series

WHO Let's Code Teaches & Reaches

Who We Target	**What We Call Workshops**
Kids	Kids Series
Technical Learners	General / Learner Series
IT/Dev Professionals	Pro Series

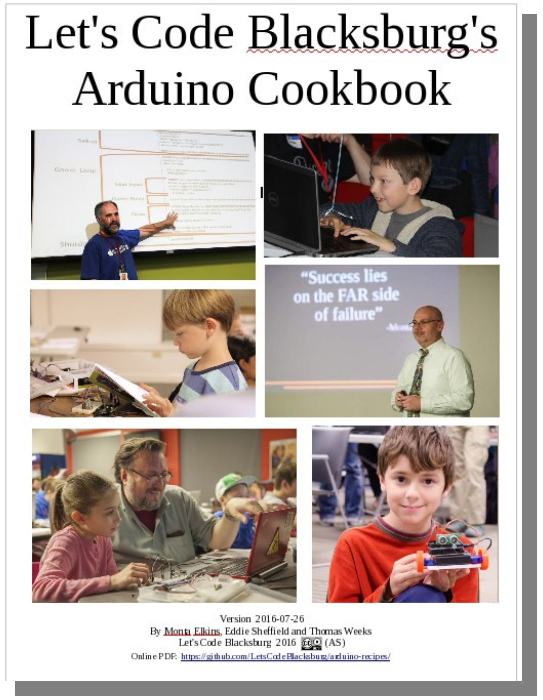
{oo} Teaching STEM w/Arduino

- About Let's Code Blacksburg!
- → About Teaching STEM Using The Arduino
- Jumping In – Arduino Overview & Running “Blink” Recipe
- Teaching Programming Theory (C++)
- Inputs vs Outputs
- Meal → “Output: Blink” + “INPUT: Button”
- Meal - Alarm On/Off → Combine Button + Sound Generation
- Meal - Proximity Alarm → Add Ultrasonic Range Sensor
- Arduino Resources

Teaching STEM w/Arduino

Teaching with the LCBB "Arduino Cookbook" System

- The "cook book" structure is comprised of mini "recipes" (see pg 2)



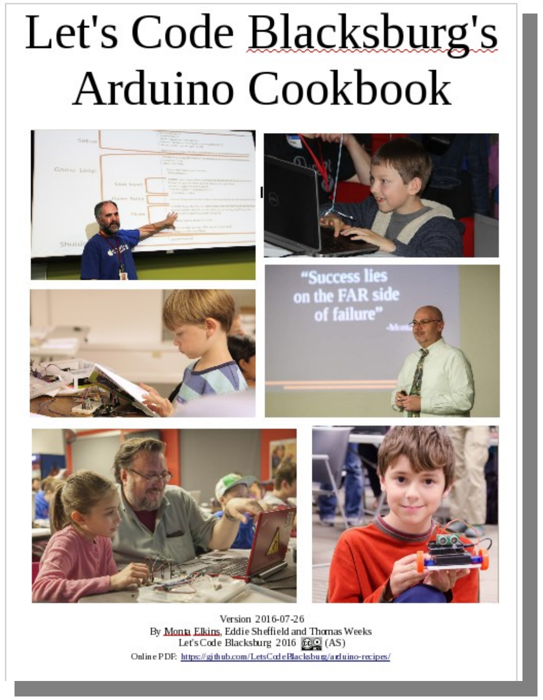
Let's Code Blacksburg's
Arduino Cookbook

Version 2016-07-26
By Monta Elkins, Eddie Sheffield and Thomas Weeks
Let's Code Blacksburg 2016 (AS)
Online PDF: https://github.com/LetsCodeBlacksburg/arduino-recipes/

Teaching STEM w/Arduino

Teaching with the LCBB “Arduino Cookbook” System

- The “cook book” structure is comprised of mini “recipes” (see pg 2)
  - 1 recipe = How 1 device is to be connected and programmed



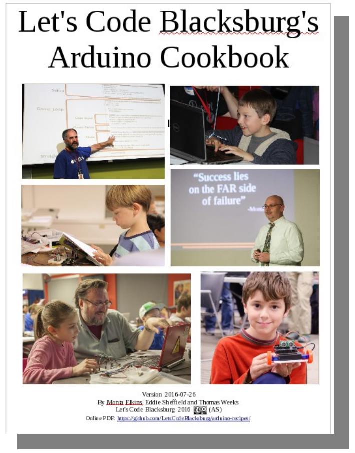
Let's Code Blacksburg's
Arduino Cookbook

Version 2016-07-26
By Monta Elkins, Eddie Sheffield and Thomas Weeks
Let's Code Blacksburg 2016 (AS)
Online PDF: https://github.com/LetsCodeBlacksburg/arduino-recipes/

Teaching STEM w/Arduino

Teaching with the LCBB “Arduino Cookbook” System

- The “cook book” structure is comprised of mini “recipes” (see pg 2)
  - 1 recipe = How 1 device is to be connected and programmed
  - Inventing something = “Cooking a Meal” = recipe1 + recipe2 + recipe3


Let's Code Blacksburg's
Arduino Cookbook

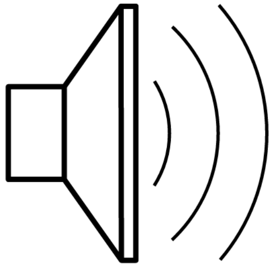
Version 2016-07-26
By Monta Elkins, Eddie Sheffield and Thomas Weeks
Let's Code Blacksburg 2016 🅭 (AS)
Online PDF: https://github.com/LetsCodeBlacksburg/arduino-recipes/

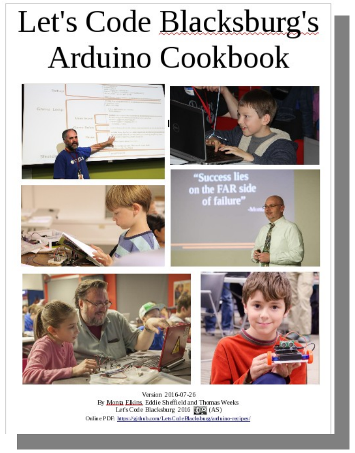
Teaching STEM w/Arduino

Teaching with the LCBB "Arduino Cookbook" System

- The "cook book" structure is comprised of mini "recipes" (see pg 2)
  - 1 recipe = How 1 device is to be <u>connected</u> and <u>programmed</u>
  - Inventing something = "Cooking a Meal" = recipe1 + recipe2 + recipe3

Speaker/sound
Recipe





Let's Code Blacksburg's
Arduino Cookbook

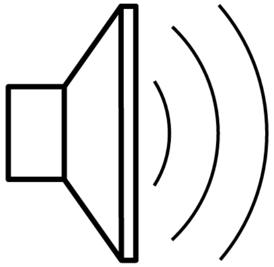
Version 2016-07-26
By Monta Elkins, Eddie Sheffield and Thomas Weeks
Let's Code Blacksburg 2016 (AS)
Online PDF: https://github.com/LetsCodeBlacksburg/arduino-recipes/

Teaching STEM w/Arduino

Teaching with the LCBB “Arduino Cookbook” System

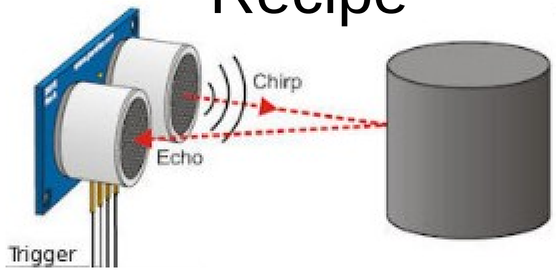
- The “cook book” structure is comprised of mini “recipes” (see pg 2)
  - 1 recipe = How 1 device is to be connected and programmed
  - Inventing something = “Cooking a Meal” = recipe1 + recipe2 + recipe3

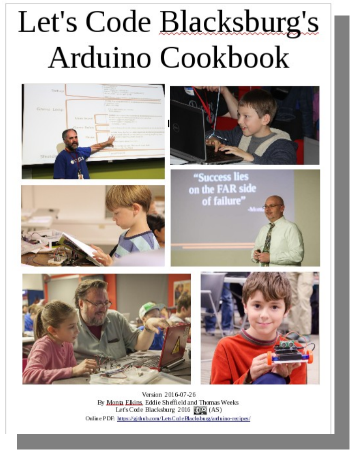
Speaker/sound Recipe



+

Range Sensor Recipe





Let's Code Blacksburg's
Arduino Cookbook

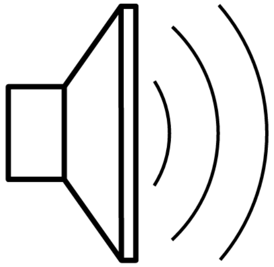
Version 2016-07-26
By Monta Elkins, Eddie Sheffield and Thomas Weeks
Let's Code Blacksburg 2016 (AS)
Online PDF: https://github.com/LetsCodeBlacksburg/arduino-recipes/

Teaching STEM w/Arduino

Teaching with the LCBB “Arduino Cookbook” System

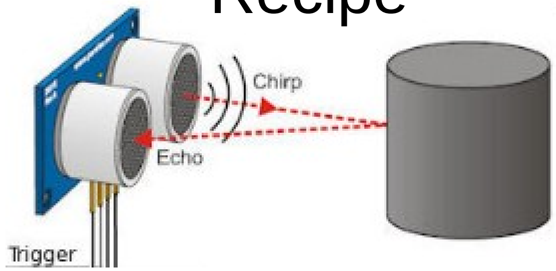
- The “cook book” structure is comprised of mini “recipes” (see pg 2)
  - 1 recipe = How 1 device is to be connected and programmed
  - Inventing something = “Cooking a Meal” = recipe1 + recipe2 + recipe3

Speaker/sound
Recipe



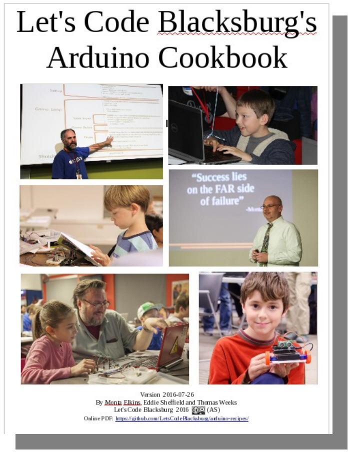
+

Range Sensor
Recipe



=

Musical
Instrument



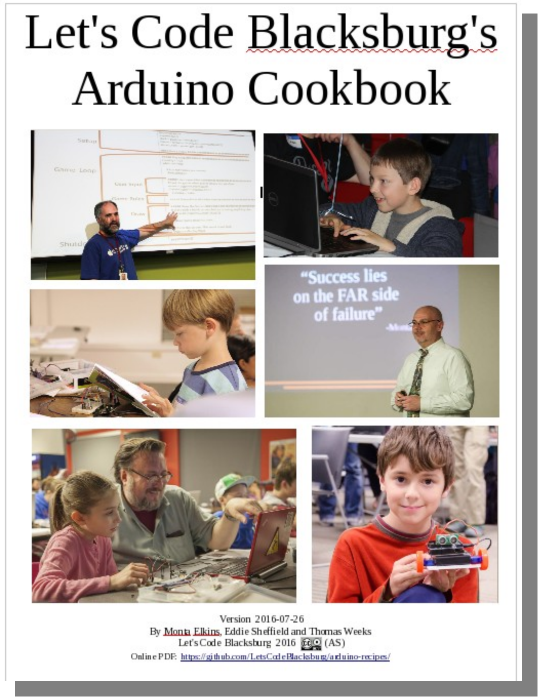
Let's Code Blacksburg's
Arduino Cookbook

Version 2016-07-26
By Monta Elkins, Eddie Sheffield and Thomas Weeks
Let's Code Blacksburg 2016 (AS)
Online PDF: https://github.com/LetsCodeBlacksburg/arduino-recipes/

Teaching STEM w/Arduino

Teaching w/the “Cookbook” System

If a child wants to dig deeper, we also include reference material (*scan over* pages 4-5)...

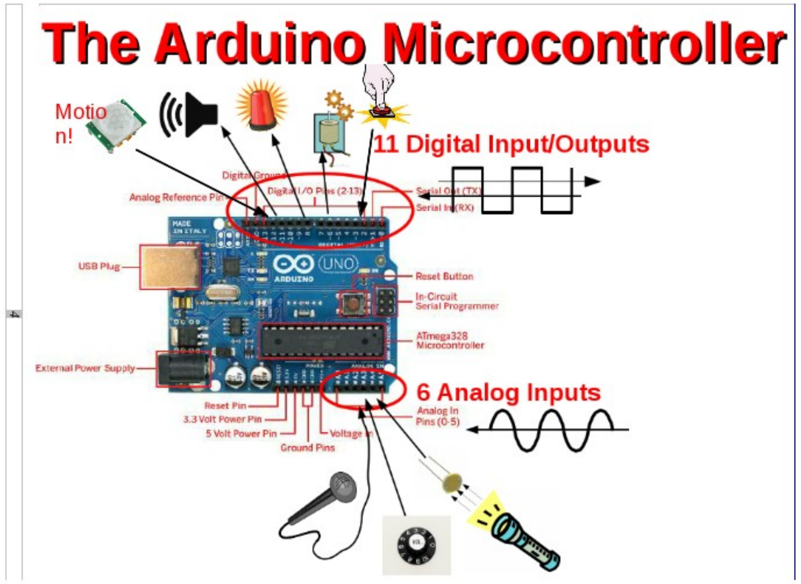


Let's Code Blacksburg's Arduino Cookbook

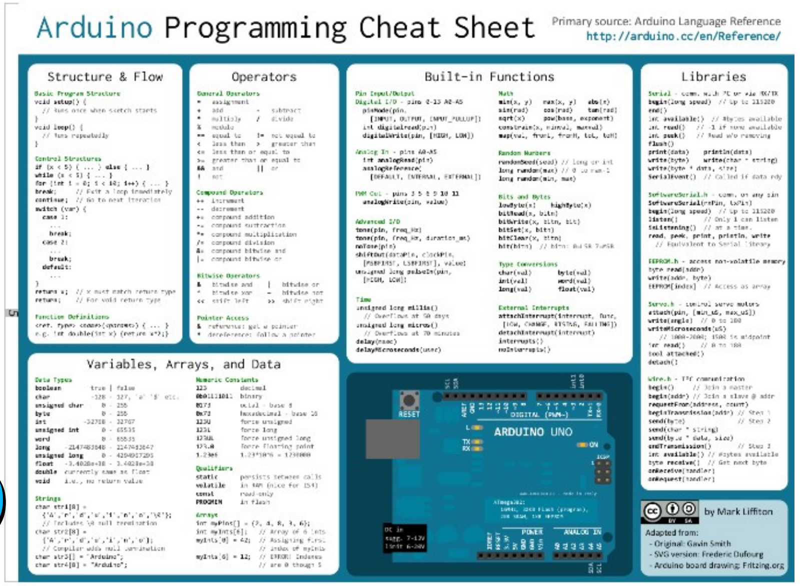
“Success lies on the FAR side of failure” -Morn

Version 2016-07-26
By Monta Elkins, Eddie Sheffield and Thomas Weeks
Let's Code Blacksburg 2016 (AS)
Online PDF: https://github.com/LetsCodeBlacksburg/arduino-recipes/

(pull this sheet out for use later)



The Arduino Microcontroller



Arduino Programming Cheat Sheet

Primary source: Arduino Language Reference
http://arduino.cc/en/Reference/

Structure & Flow

Basic Program Structure

```
void setup() {
  // Runs once when sketch starts
}
void loop() {
  // Runs repeatedly
}
```

Control Structures

```
if (x < 5) { ... } else { ... }
while (x < 5) { ... }
for (int i = 0; i < 10; i++) { ... }
break;     // Exit a loop immediately
continue;  // Go to next iteration
switch (var) {
  case 1:
    ...
    break;
  case 2:
    ...
    break;
  default:
    ...
}
```

Operators

General Operators

```
=   assignment
+   add
*   multiply
%   modulo
==  equal to
<   less than
<=  less than or equal to
>=  greater than or equal to
&&  and
!   not
```

Compound Operators

```
++  increment
--  decrement
+=  compound addition
-=  compound subtraction
*=  compound multiplication
/=  compound division
&=  compound bitwise and
|=  compound bitwise or
```

Bitwise Operators

Built-in Functions

Math

```
min(x, y)   max(x, y)   abs(x)
sin(rad)    cos(rad)    tan(rad)
sqrt(x)     pow(base, exponent)
constrain(x, minval, maxval)
```

Libraries

Serial - comm. with PC or via RX/TX

```
begin(long speed)  // Up to 115200
end()
int available()  // #bytes available
int read()   // -1 if none available
```

```
send(byte)                // Step 2
send(char * string)
send(byte * data, size)
endTransmission()         // Step 3
int available() // #bytes available
byte receive()  // Get next byte
onReceive(handler)
onRequest(handler)
```

Variables, Arrays, and Data

by Mark Liffiton

Adapted from:
- Original: Gavin Smith
- SVG version: Frederic Dufourg
- Arduino board drawing: Fritzing.org

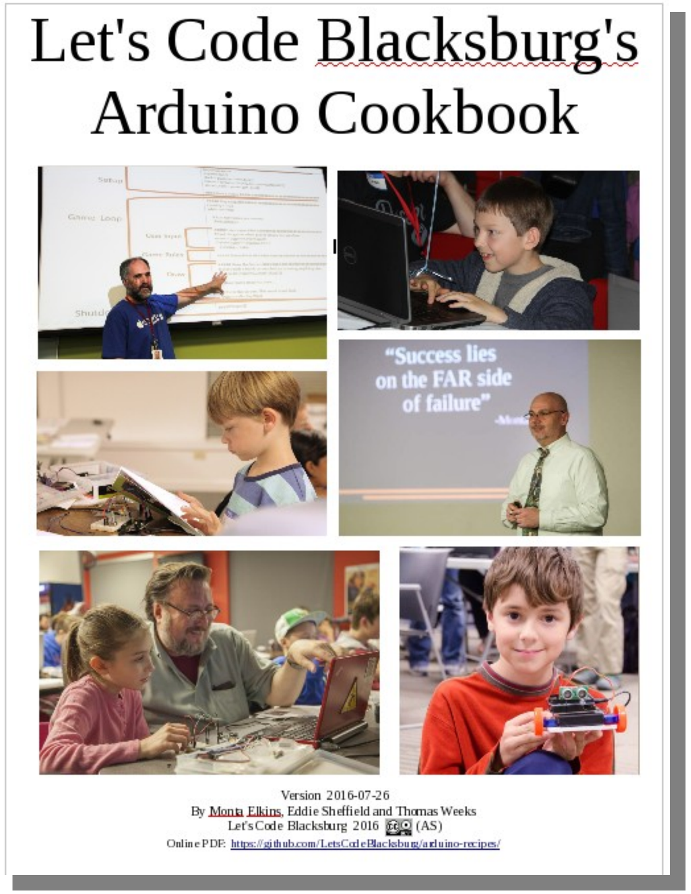
Teaching STEM w/Arduino

Teaching with the LCBB “Arduino Cookbook” System

Go ahead and look a the recipes on pages 6-8 of the cookbook..

<u>Note that each “recipe” has a:</u>

- **<u>What</u>** – Theory/concept intro
- **<u>How</u>** – to connect and program it
- **<u>Fail</u>** – to explore possible causes of it not working correctly.



Let's Code Blacksburg's
Arduino Cookbook

Version 2016-07-26
By Monta Elkins, Eddie Sheffield and Thomas Weeks
Let's Code Blacksburg 2016 (AS)
Online PDF: https://github.com/LetsCodeBlacksburg/arduino-recipes/

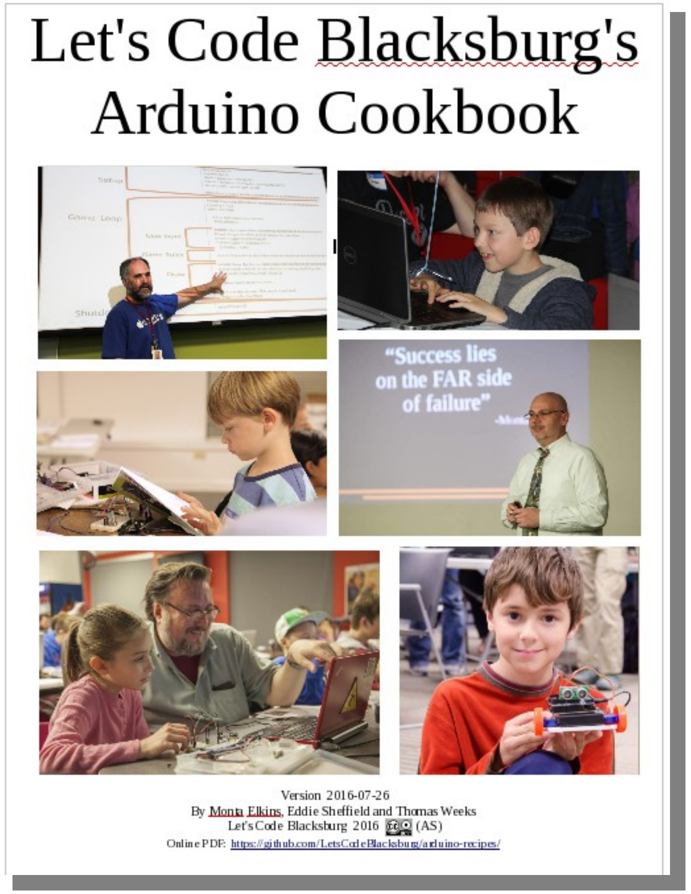
{oo} Teaching STEM w/Arduino

Teaching with the LCBB “Arduino Cookbook” System

Go ahead and look a the recipes on pages 6-8 of the cookbook..

Note that each “recipe” has a:

- **What** – Theory/concept intro
- **How** – to connect and program it
- **Fail** – to explore possible causes of it not working correctly.



Let's Code Blacksburg's
Arduino Cookbook

“Success lies on the FAR side of failure” -Monta

Version 2016-07-26
By Monta Elkins, Eddie Sheffield and Thomas Weeks
Let's Code Blacksburg 2016 (cc) (AS)
Online PDF: https://github.com/LetsCodeBlacksburg/arduino-recipes/

Teaching STEM w/Arduino

We REALLY Like To Stress
The *Importance* of...

FAiLURE

Teaching STEM w/Arduino

We REALLY Like To Stress
The *Importance* of...

FAiLURE

Teaching kids how to accept & learn from failure...
is CRITICAL for Critical Thinking!!

Teaching STEM w/Arduino

We REALLY Like To Stress
The *Importance* of...

FAiLURE

Teaching kids how to accept & learn from failure...
is CRITICAL for Critical Thinking!!

(you can't *do* science if you can't get past your own ego)

Teaching STEM w/Arduino

We REALLY Like To Stress
The Importance of...

FAiLURE

Teaching STEM w/Arduino

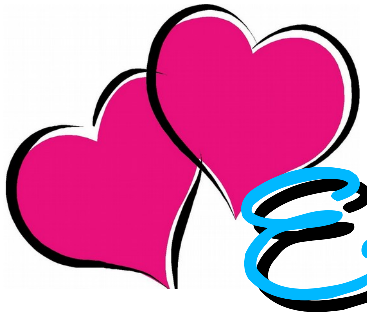
We REALLY Like To Stress The Importance of...

~~FAILURE~~

Teaching STEM w/Arduino

We REALLY Like To Stress The Importance of...

~~FAiLURE~~



Experiments!

{oo} Teaching STEM w/Arduino

We REALLY Like To Stress The Importance of...

~~FAILURE~~

Experiments!

Teaching STEM w/Arduino

We REALLY Like To Stress
The Importance of...

FAILURE

The secret to inventing
(or prototyping) is to..

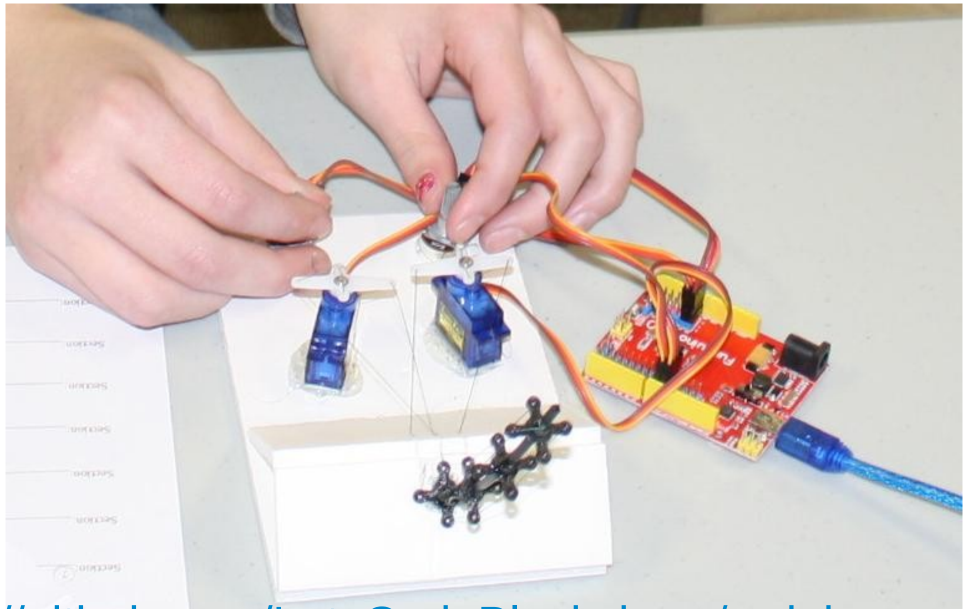
"Fail fast, fail cheap!"
—Monta Elkins

Teaching STEM w/Arduino

FAILURE STORY

"Fail fast, fail cheap!"
—Monta Elkins

The R.A.T. Workshop
(Robotic Alien Tentacle)



https://github.com/LetsCodeBlacksburg/arduino-recipes

Teaching STEM w/Arduino

The Translation of

"Fail fast, fail cheap!"
—Monta Elkins

Teach Kids To Only
Build Or Change..

One...
Small...
Thing...
At...
A...
Time...

Teaching STEM w/Arduino

Takeaways of Teaching Failure (today)

1) To do science, let go of your ego and assumptions.

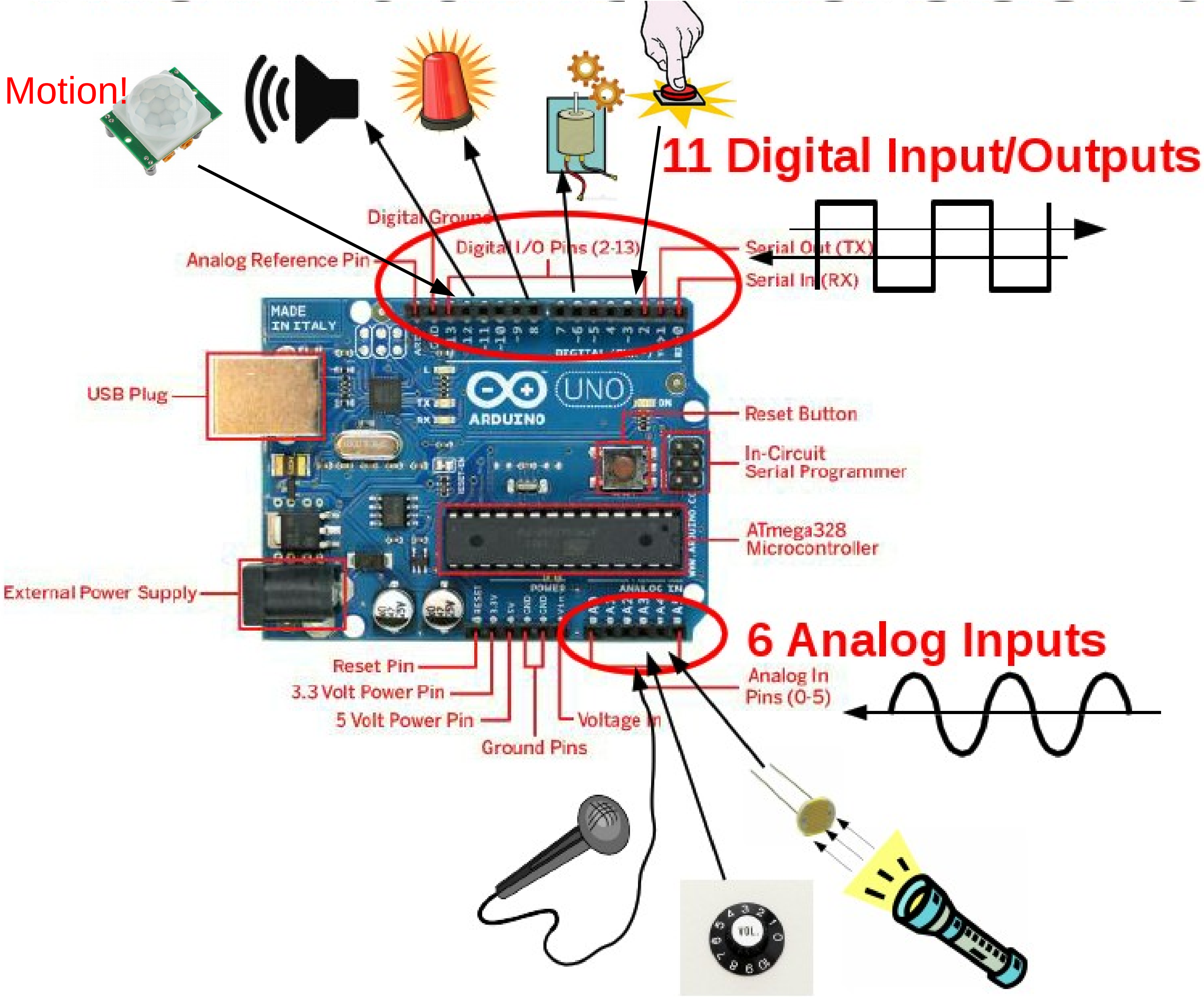
2) People Learn, Experiments Fail.

3) “Fail Fast, Fail Cheap!” =
Only change one small thing at a time.

Jumping In

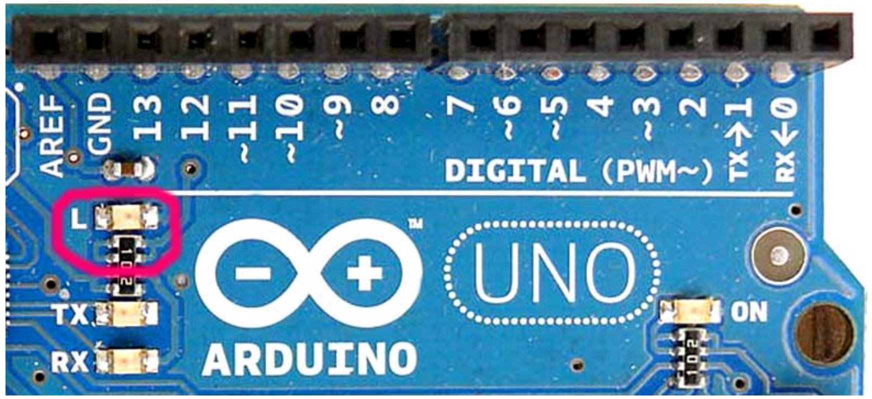
- About Let's Code Blacksburg!
- About Teaching STEM Using The Arduino
- Jumping In – Arduino Overview & Running “Blink” Recipe
- Teaching Programming Theory (C++)
- Inputs vs Outputs
- Meal → “Output: Blink” + “INPUT: Button”
- Meal - Alarm On/Off → Combine Button + Sound Generation
- Meal - Proximity Alarm → Add Ultrasonic Range Sensor
- Arduino Resources

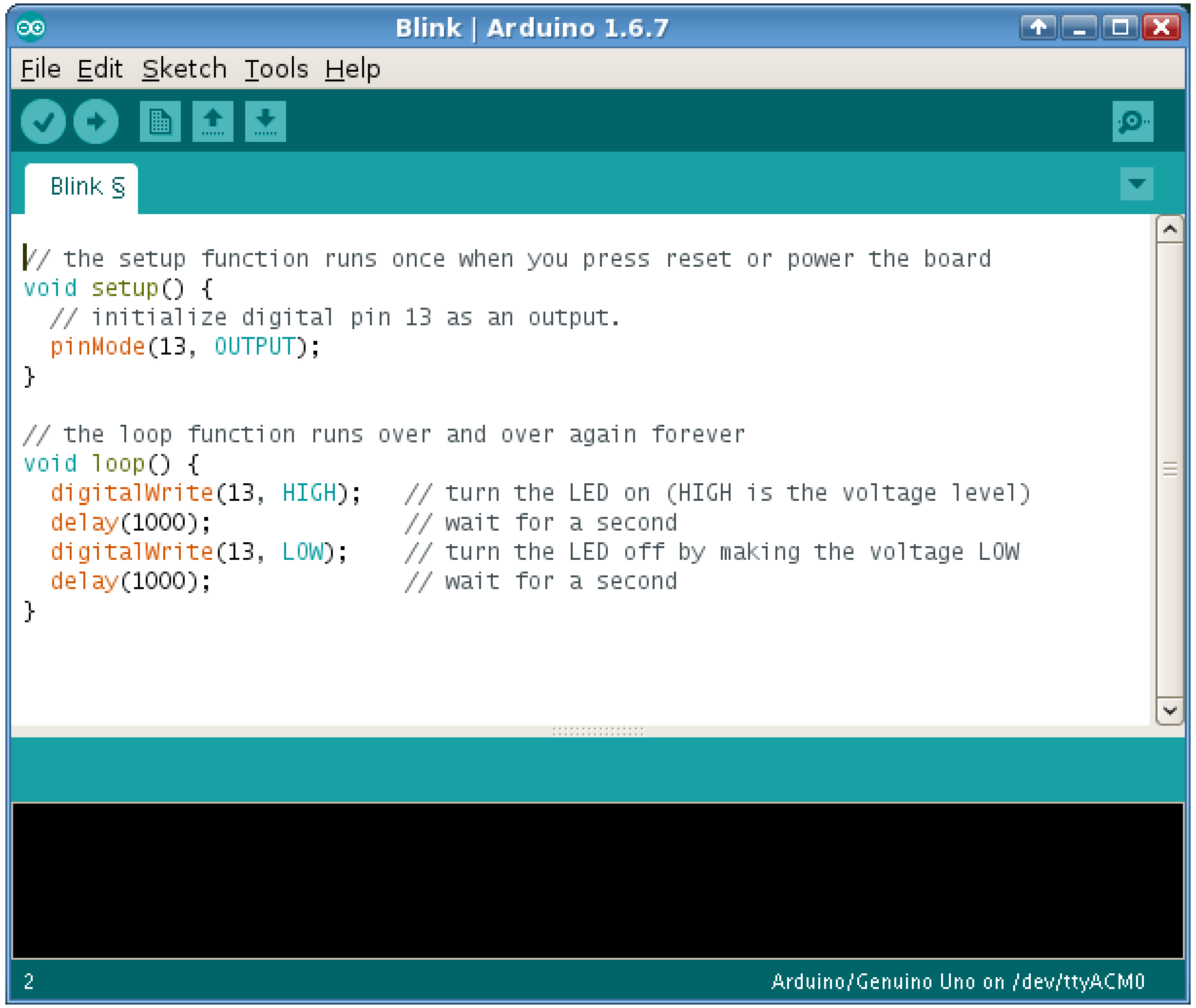
Arduino Overview (pg4)



Every Class' First Program, "Blink" pg7

Blink is more than just a "Hello World" program..



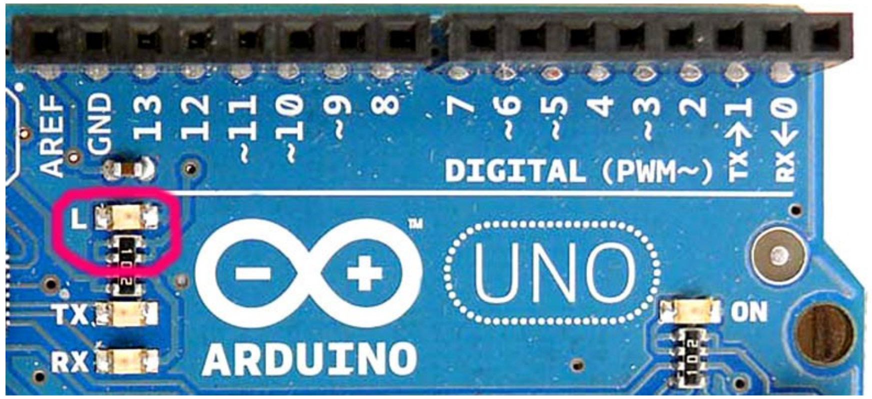


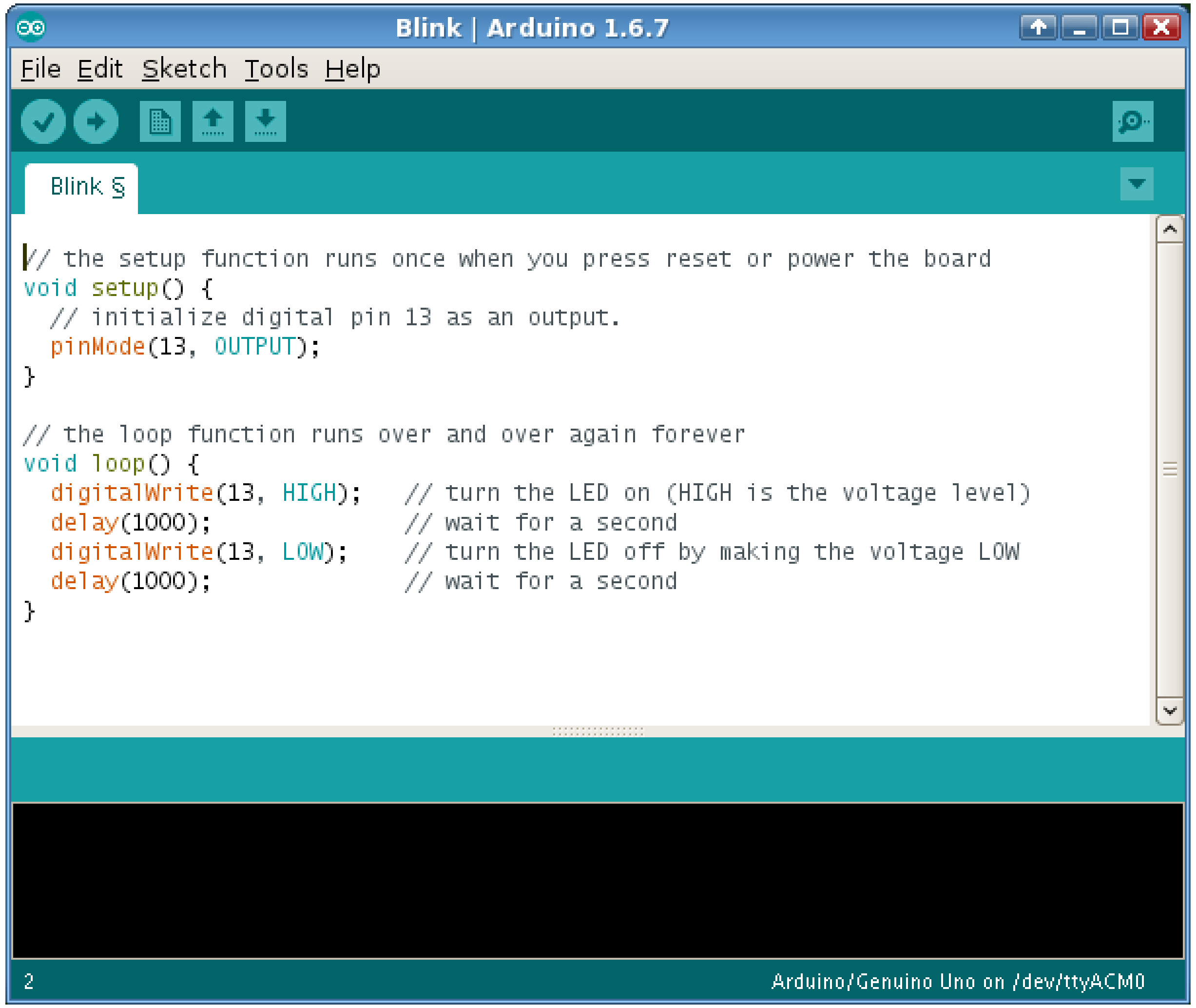
```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);              // wait for a second
}
```

Every Class' First Program, “Blink” pg7

Blink is more than just
a “Hello World” program..
It accomplishes:

- ✔ Tests hardware & drivers
- ✔ Tests USB/serial link settings
- ✔ Verifies everything works



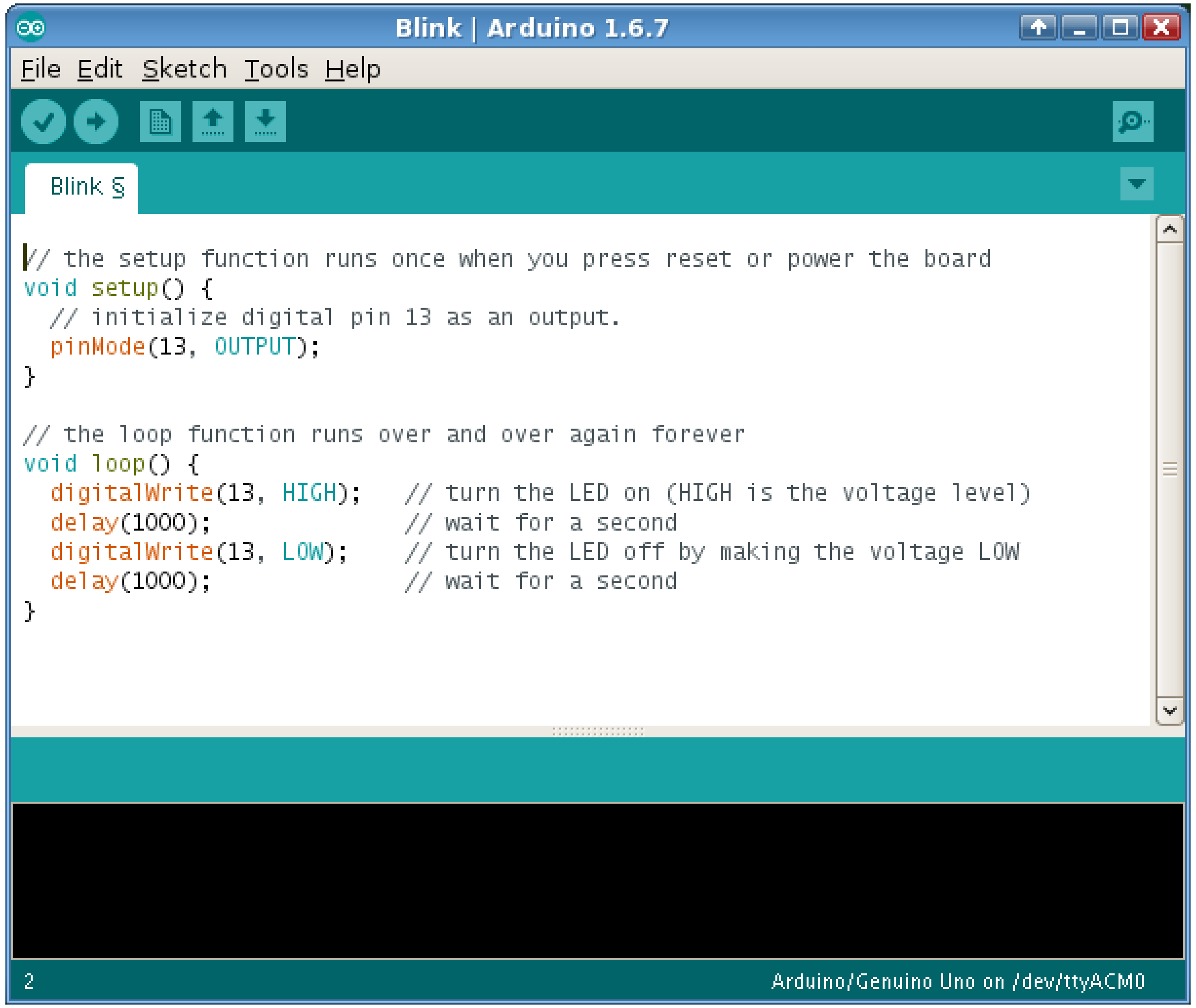

Blink | Arduino 1.6.7
File Edit Sketch Tools Help
Blink §
2
Arduino/Genuino Uno on /dev/ttyACM0

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);              // wait for a second
}
```

Every Class' First Program, "Blink" pg7

Let's Dive In!

- Turn to pg 7
- Start Arduino Program
- Follow "How" Section
- If (success), then
  - Add variable for ledPin and replace 13 with ledPin and run


Blink | Arduino 1.6.7
File Edit Sketch Tools Help
Blink §
2
Arduino/Genuino Uno on /dev/ttyACM0

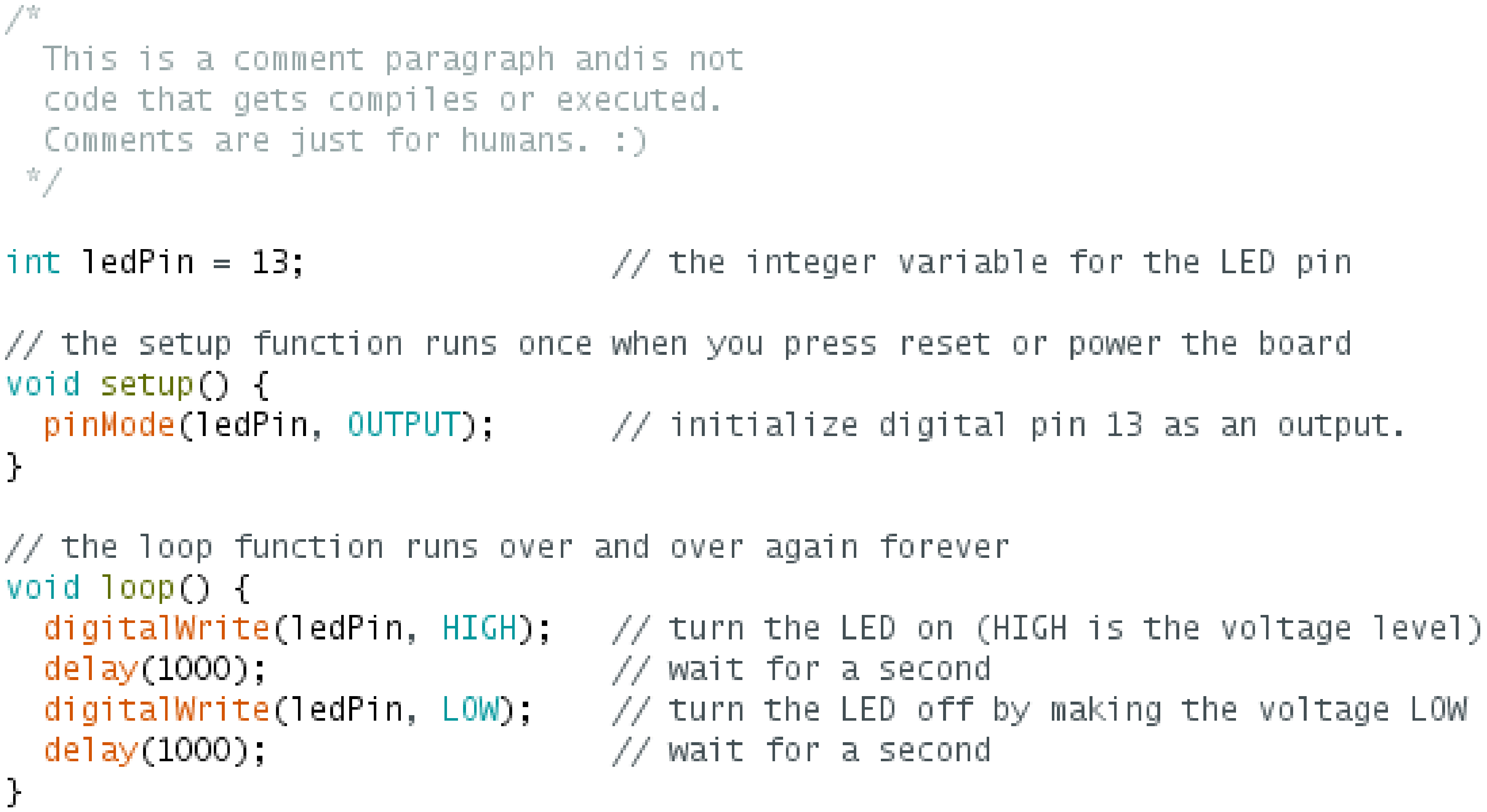
```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);              // wait for a second
}
```

Programming Theory

- About Let's Code Blacksburg!
- About Teaching STEM Using The Arduino
- Jumping In – Arduino Overview & Running “Blink” Recipe
- → Teaching Programming Theory (C++)
- Inputs vs Outputs
- Meal → “Output: Blink” + “INPUT: Button”
- Meal - Alarm On/Off → Combine Button + Sound Generation
- Meal - Proximity Alarm → Add Ultrasonic Range Sensor
- Arduino Resources

Programming Theory

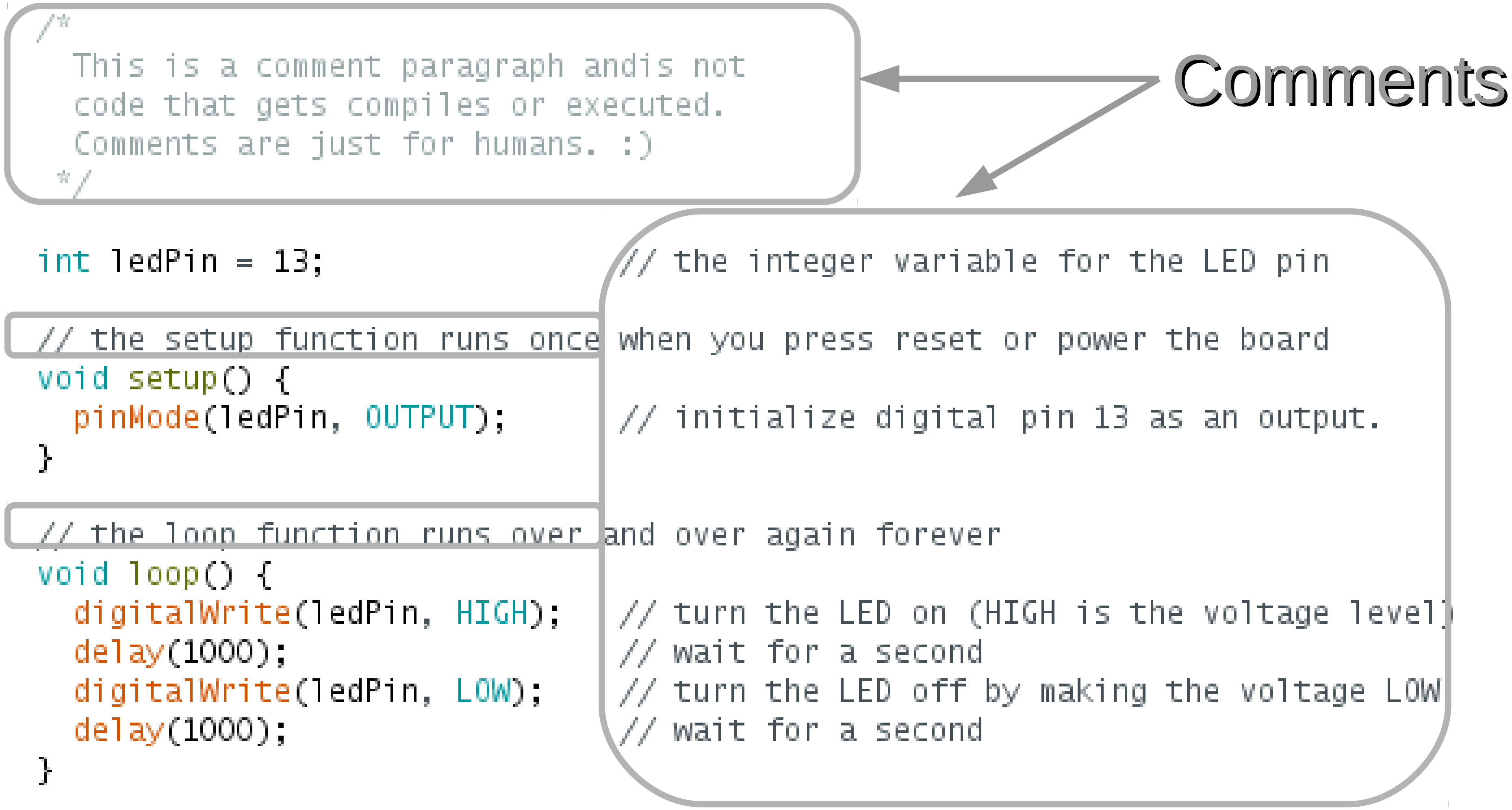
Demonstrating C++/Wiring structure using “Blink”



```
/*
  This is a comment paragraph andis not
  code that gets compiles or executed.
  Comments are just for humans. :)
 */

int ledPin = 13;                // the integer variable for the LED pin

// the setup function runs once when you press reset or power the board
void setup() {
  pinMode(ledPin, OUTPUT);      // initialize digital pin 13 as an output.
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(ledPin, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                  // wait for a second
  digitalWrite(ledPin, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                  // wait for a second
}
```

Programming Theory

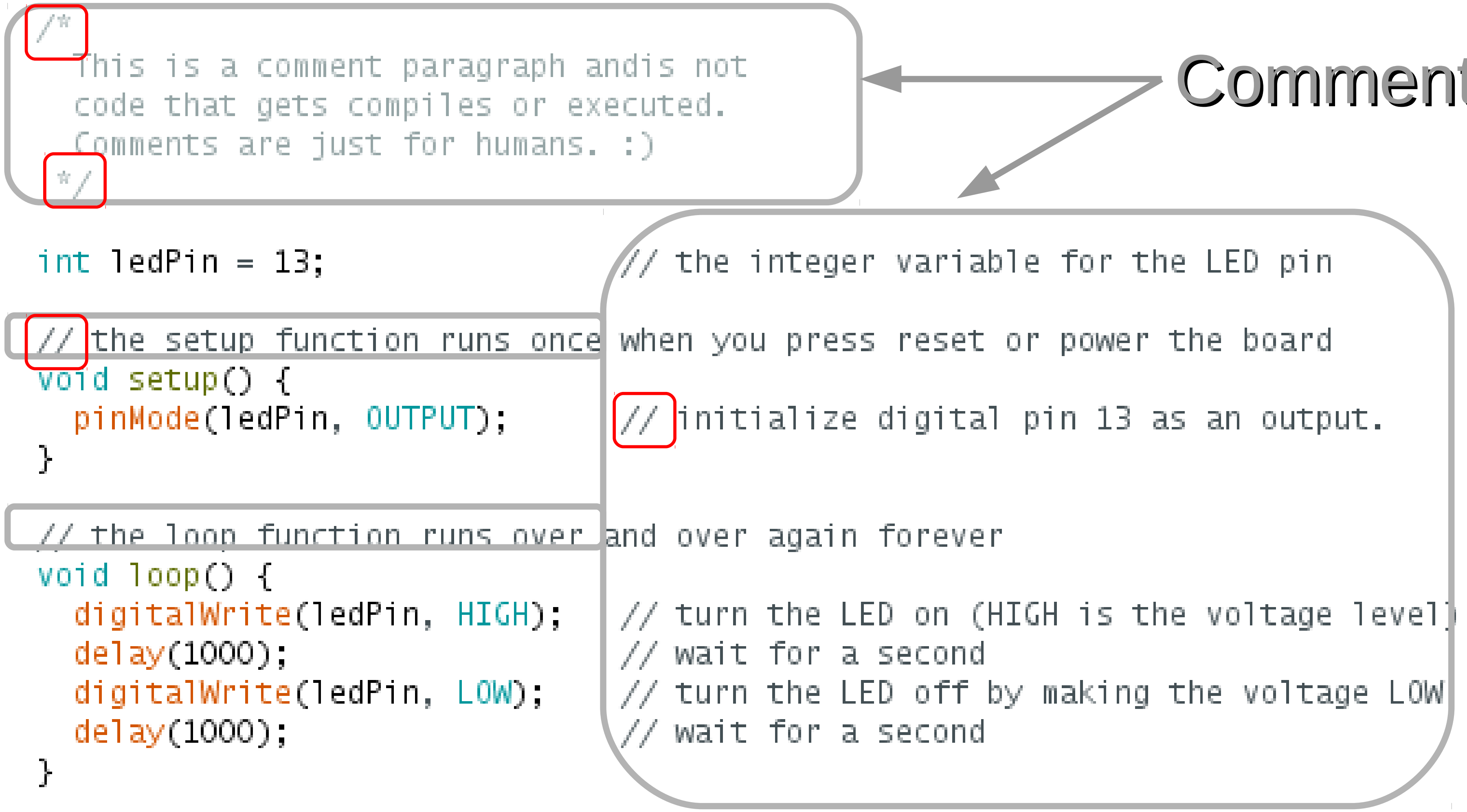
Demonstrating C++/Wiring structure using “Blink”



```
/*
  This is a comment paragraph andis not
  code that gets compiles or executed.
  Comments are just for humans. :)
 */

int ledPin = 13;               // the integer variable for the LED pin

// the setup function runs once when you press reset or power the board
void setup() {
  pinMode(ledPin, OUTPUT);     // initialize digital pin 13 as an output.
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(ledPin, HIGH);  // turn the LED on (HIGH is the voltage level)
  delay(1000);                 // wait for a second
  digitalWrite(ledPin, LOW);   // turn the LED off by making the voltage LOW
  delay(1000);                 // wait for a second
}
```

Programming Theory

Demonstrating C++/Wiring structure using “Blink”

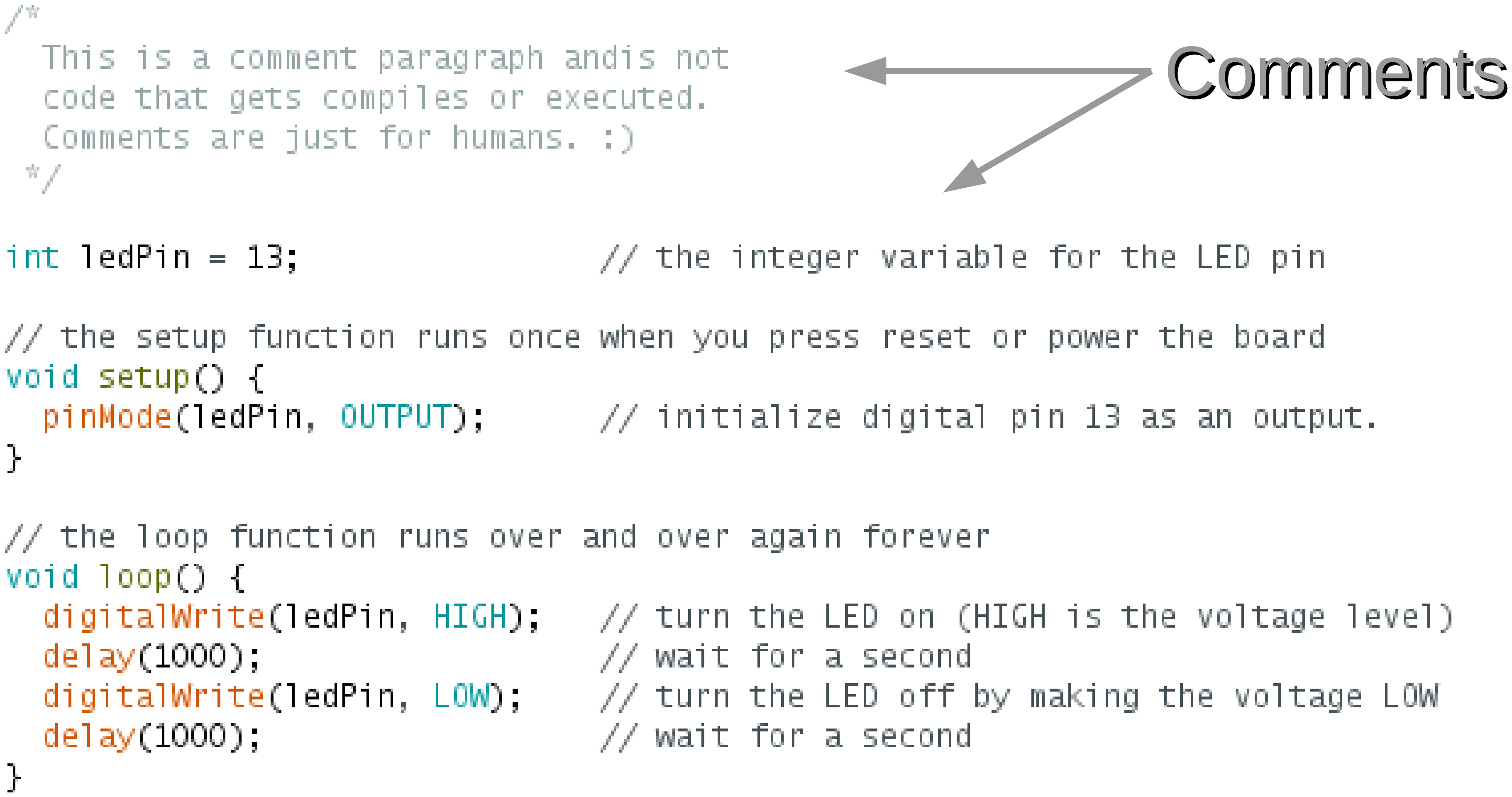


```
/*
  This is a comment paragraph andis not
  code that gets compiles or executed.
  Comments are just for humans. :)
 */

int ledPin = 13;                // the integer variable for the LED pin

// the setup function runs once when you press reset or power the board
void setup() {
  pinMode(ledPin, OUTPUT);      // initialize digital pin 13 as an output.
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(ledPin, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                  // wait for a second
  digitalWrite(ledPin, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                  // wait for a second
}
```

Comments

Programming Theory

Demonstrating C++/Wiring structure using “Blink”

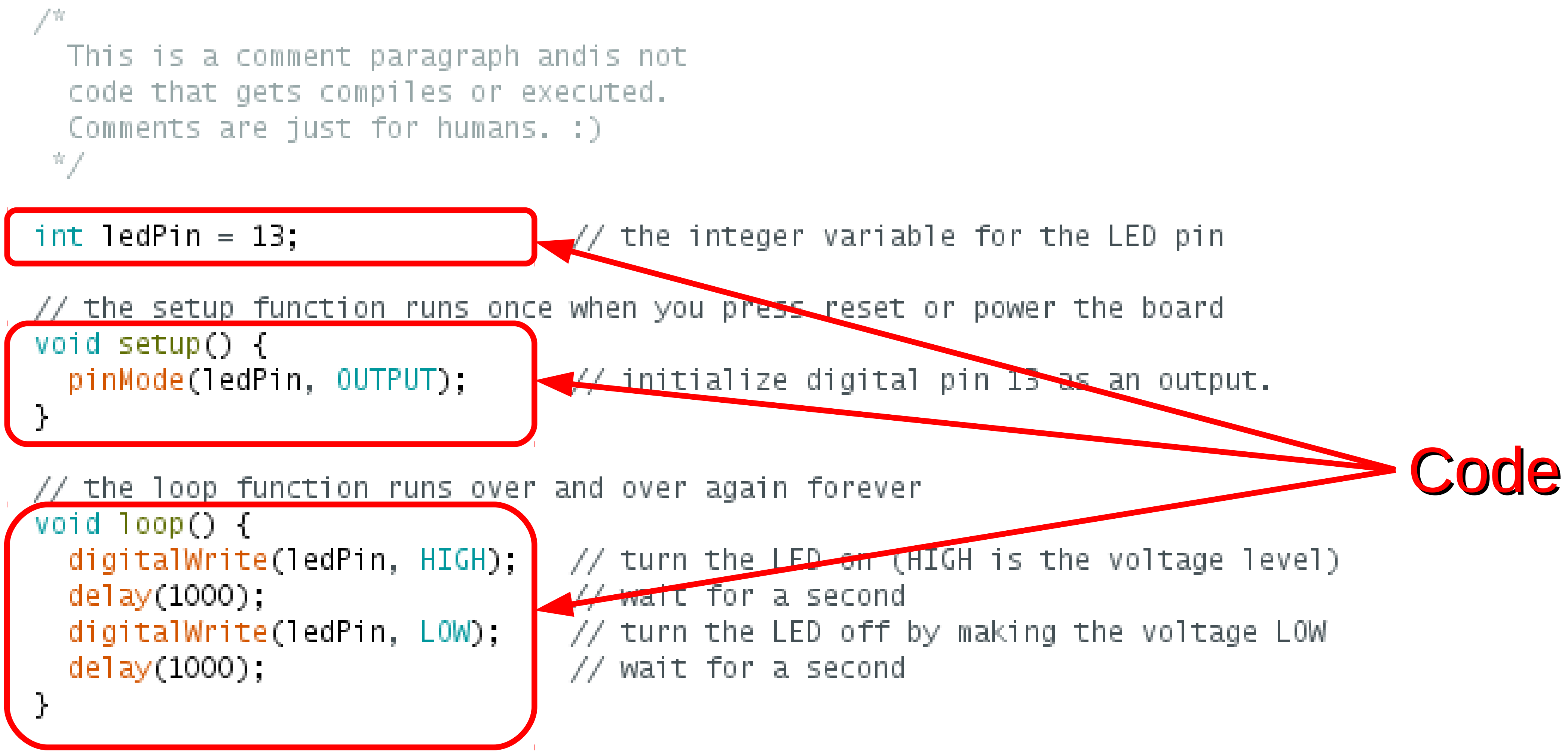


```
/*
  This is a comment paragraph andis not
  code that gets compiles or executed.
  Comments are just for humans. :)
 */

int ledPin = 13;                // the integer variable for the LED pin

// the setup function runs once when you press reset or power the board
void setup() {
  pinMode(ledPin, OUTPUT);      // initialize digital pin 13 as an output.
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(ledPin, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                  // wait for a second
  digitalWrite(ledPin, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                  // wait for a second
}
```

Comments

Programming Theory

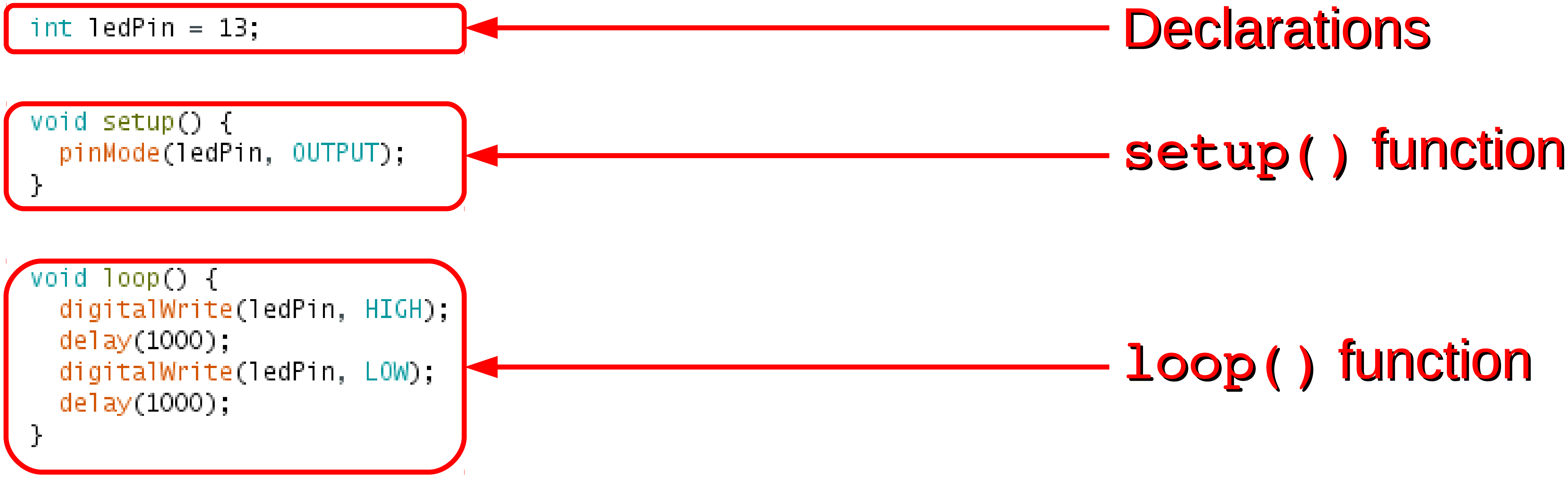
Demonstrating C++/Wiring structure using “Blink”



```
/*
  This is a comment paragraph andis not
  code that gets compiles or executed.
  Comments are just for humans. :)
 */

int ledPin = 13;                // the integer variable for the LED pin

// the setup function runs once when you press reset or power the board
void setup() {
  pinMode(ledPin, OUTPUT);      // initialize digital pin 13 as an output.
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(ledPin, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                  // wait for a second
  digitalWrite(ledPin, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                  // wait for a second
}
```

Programming Theory

Demonstrating C++/Wiring structure using “Blink”



```
int ledPin = 13;
```

Declarations

```
void setup() {
  pinMode(ledPin, OUTPUT);
}
```

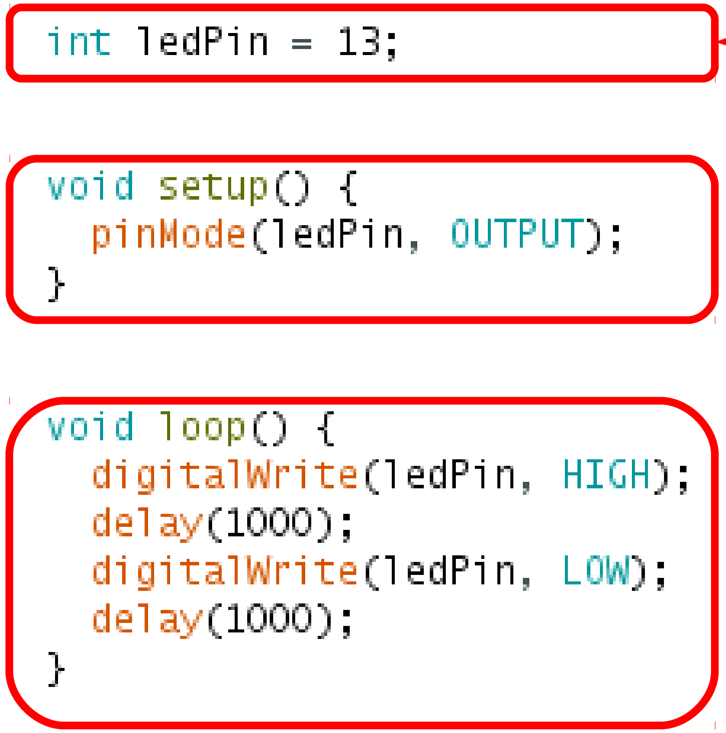
`setup()` function

```
void loop() {
  digitalWrite(ledPin, HIGH);
  delay(1000);
  digitalWrite(ledPin, LOW);
  delay(1000);
}
```

`loop()` function

Programming Theory

Demonstrating C++/Wiring structure using “Blink”

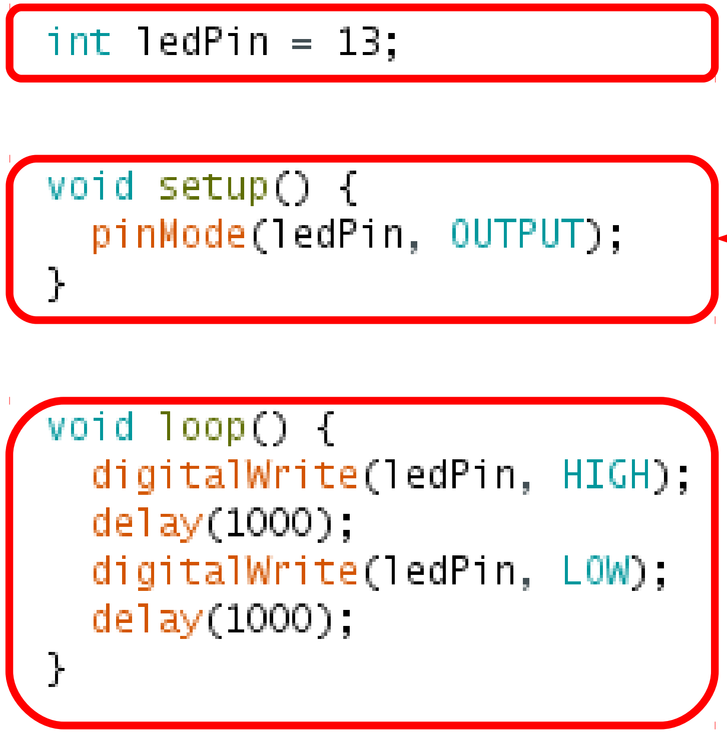


```
int ledPin = 13;

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

void loop() {
  digitalWrite(ledPin, HIGH);
  delay(1000);
  digitalWrite(ledPin, LOW);
  delay(1000);
}
```

Declarations

Programming Theory

Demonstrating C++/Wiring structure using “Blink”



```
int ledPin = 13;
```

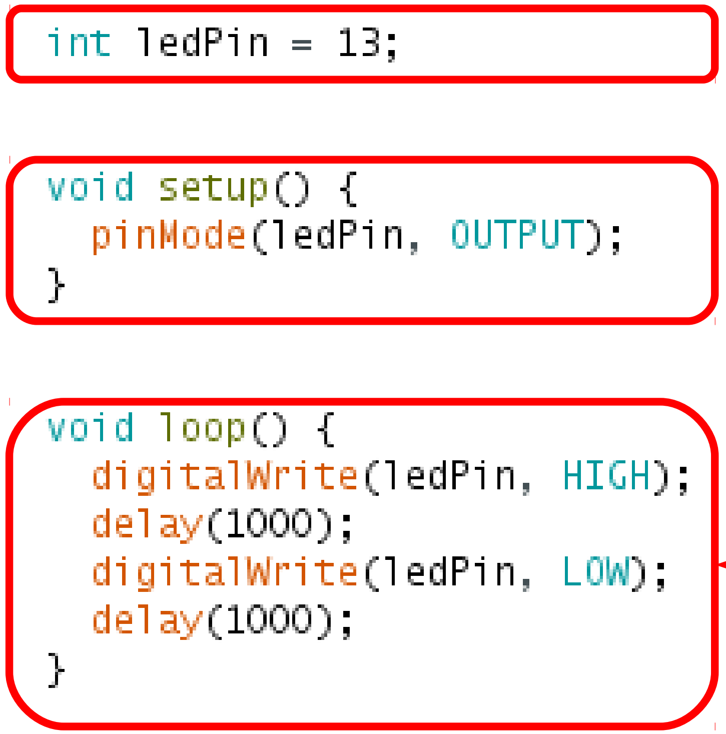
```
void setup() {
  pinMode(ledPin, OUTPUT);
}
```

```
void loop() {
  digitalWrite(ledPin, HIGH);
  delay(1000);
  digitalWrite(ledPin, LOW);
  delay(1000);
}
```

setup( ) function

Programming Theory

Demonstrating C++/Wiring structure using “Blink”

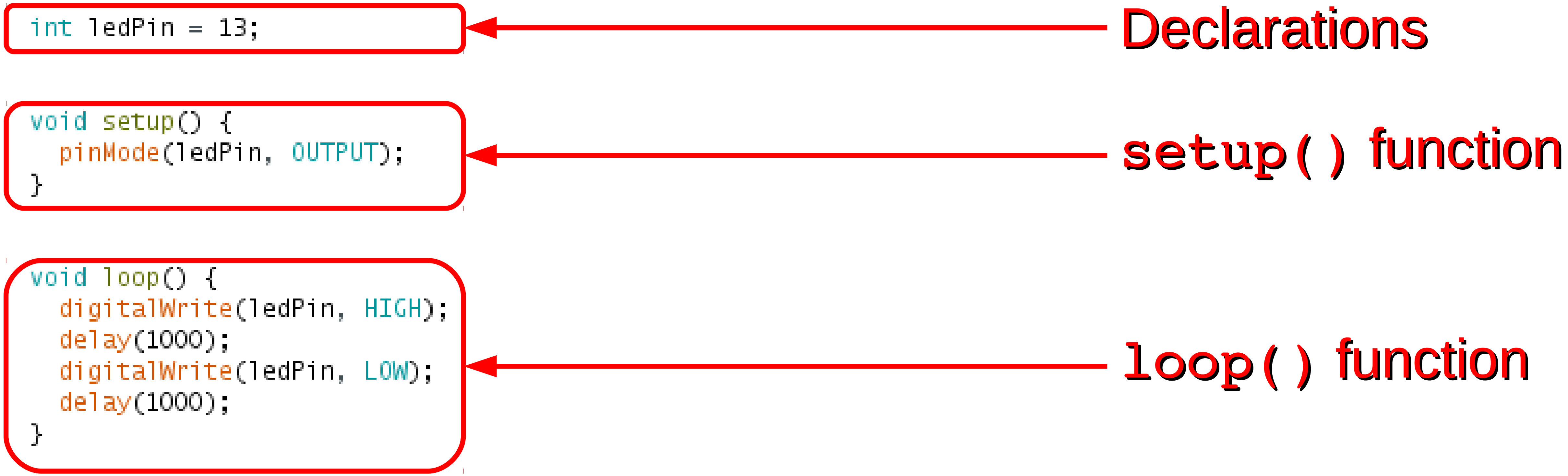


```
int ledPin = 13;

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

void loop() {
  digitalWrite(ledPin, HIGH);
  delay(1000);
  digitalWrite(ledPin, LOW);
  delay(1000);
}
```

`loop()` function

Programming Theory

Demonstrating C++/Wiring structure using “Blink”



```
int ledPin = 13;
```

Declarations

```
void setup() {
  pinMode(ledPin, OUTPUT);
}
```

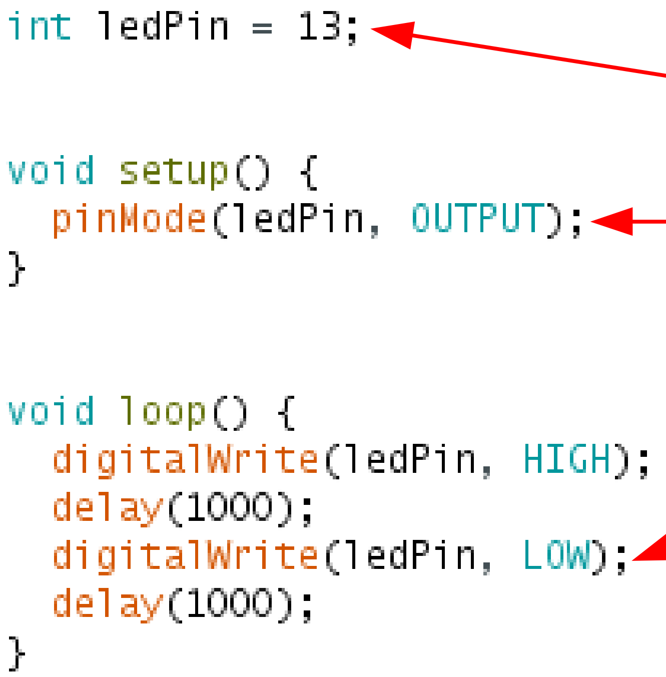
`setup( )` function

```
void loop() {
  digitalWrite(ledPin, HIGH);
  delay(1000);
  digitalWrite(ledPin, LOW);
  delay(1000);
}
```

`loop( )` function

Programming Theory

Demonstrating C++/Wiring structure using “Blink”

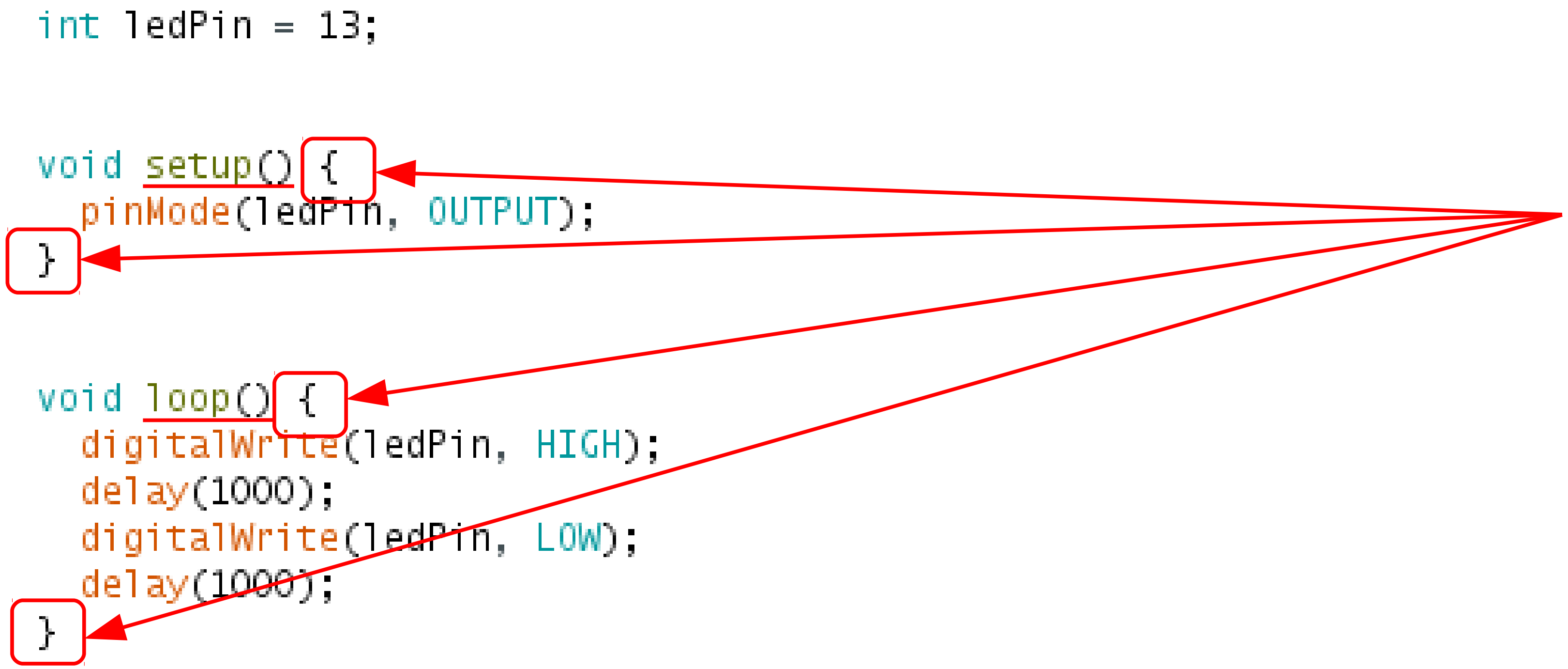


```
int ledPin = 13;

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

void loop() {
  digitalWrite(ledPin, HIGH);
  delay(1000);
  digitalWrite(ledPin, LOW);
  delay(1000);
}
```

; = end of command

Programming Theory

Demonstrating C++/Wiring structure using “Blink”

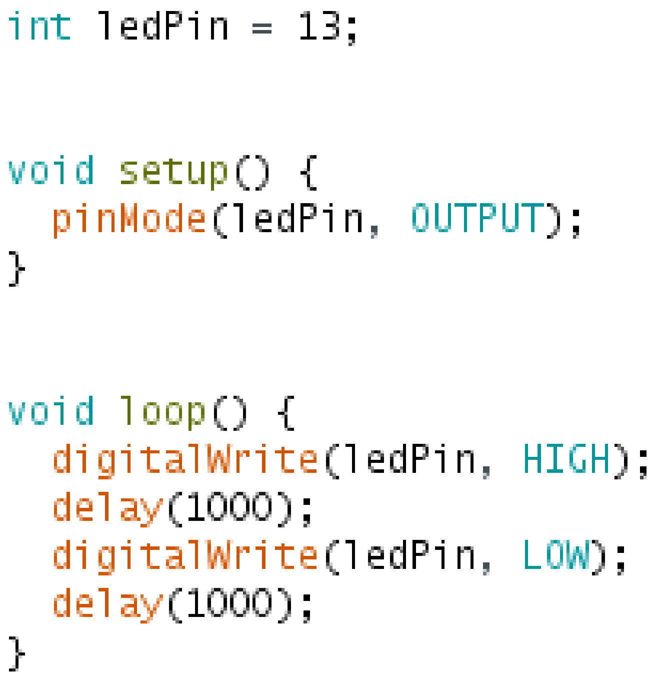


```
int ledPin = 13;

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

void loop() {
  digitalWrite(ledPin, HIGH);
  delay(1000);
  digitalWrite(ledPin, LOW);
  delay(1000);
}
```

{ } = start and end of a function

Programming Theory

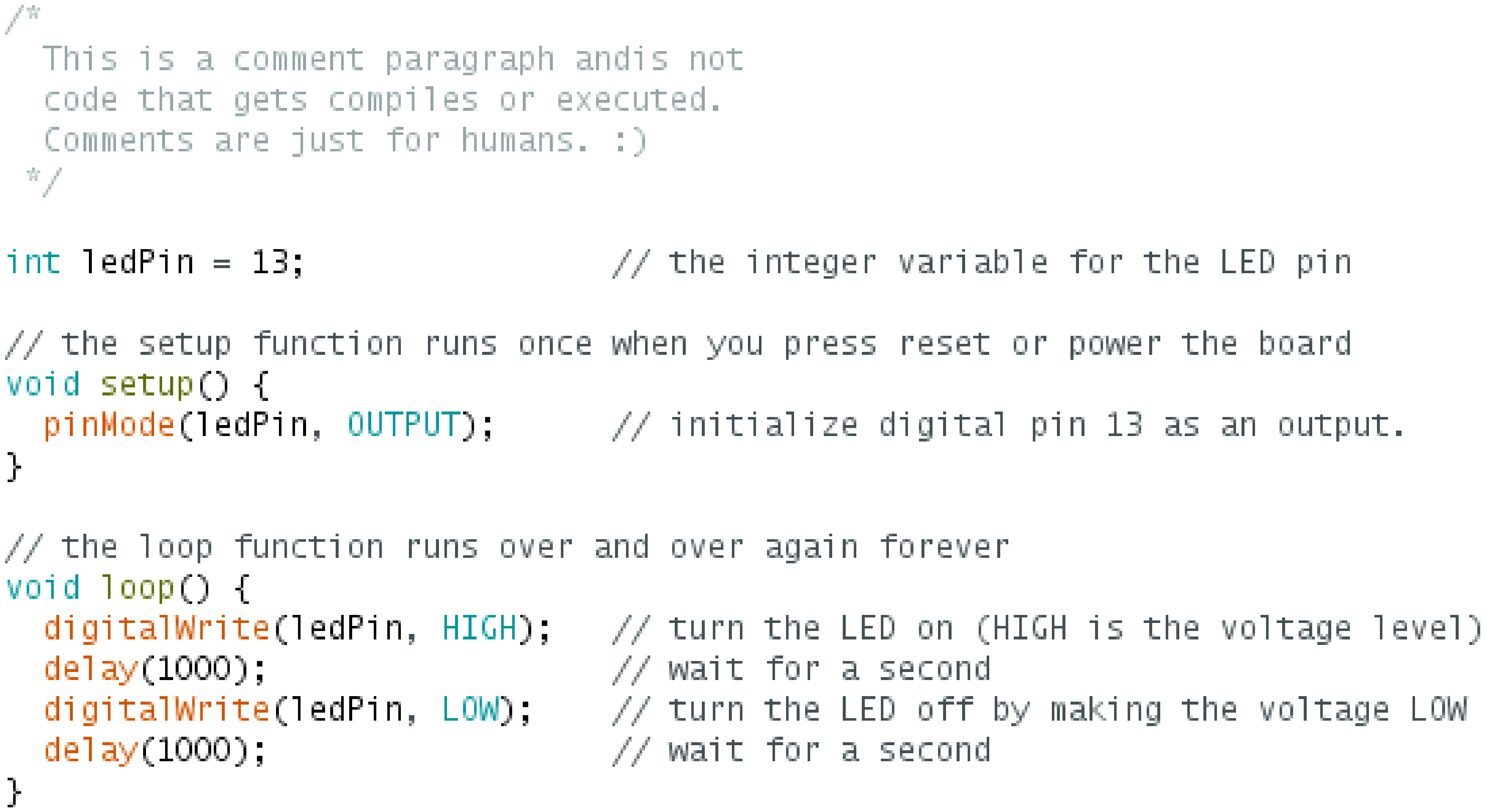
Demonstrating C++/Wiring structure using “Blink”



```
int ledPin = 13;


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


void loop() {
  digitalWrite(ledPin, HIGH);
  delay(1000);
  digitalWrite(ledPin, LOW);
  delay(1000);
}
```

Programming Theory

Demonstrating C++/Wiring structure using “Blink”



```
/*
  This is a comment paragraph andis not
  code that gets compiles or executed.
  Comments are just for humans. :)
 */

int ledPin = 13;                // the integer variable for the LED pin

// the setup function runs once when you press reset or power the board
void setup() {
  pinMode(ledPin, OUTPUT);      // initialize digital pin 13 as an output.
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(ledPin, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                  // wait for a second
  digitalWrite(ledPin, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                  // wait for a second
}
```

Programming Theory

Demonstrating simple programming logic using...

Programming Theory

Demonstrating simple programming logic using...

```
if()      // to test for conditions and run code once if true
```

Programming Theory

Demonstrating simple programming logic using...

```
if()      // to test for conditions and run code once if true

while() // to run code as long as a condition is true
```

Programming Theory

Demonstrating simple programming logic using...

```
if()      // to test for conditions and run code once if true

while() // to run code as long as a condition is true

for()   // to run code for a determined number of times
```

Programming Theory

Demonstrating simple programming logic using...

```
if(condition==true) {
   doThisOnce();      // if true, run code once
}


while(condition==true) {
   doThis();          // run code until condition is not true
}


for(int y=0 ; y<10 ; Y++)   {
   doThis();          // run ten times
}
```

Programming Theory

Demonstrating simple programming logic using...

```
if(condition==true) {
   doThisOnce();
}
```

```
while(condition==true) {
   doThis();
}
```

```
for(int y=0 ; y<10 ; Y++)   {
   doThis();
}
```

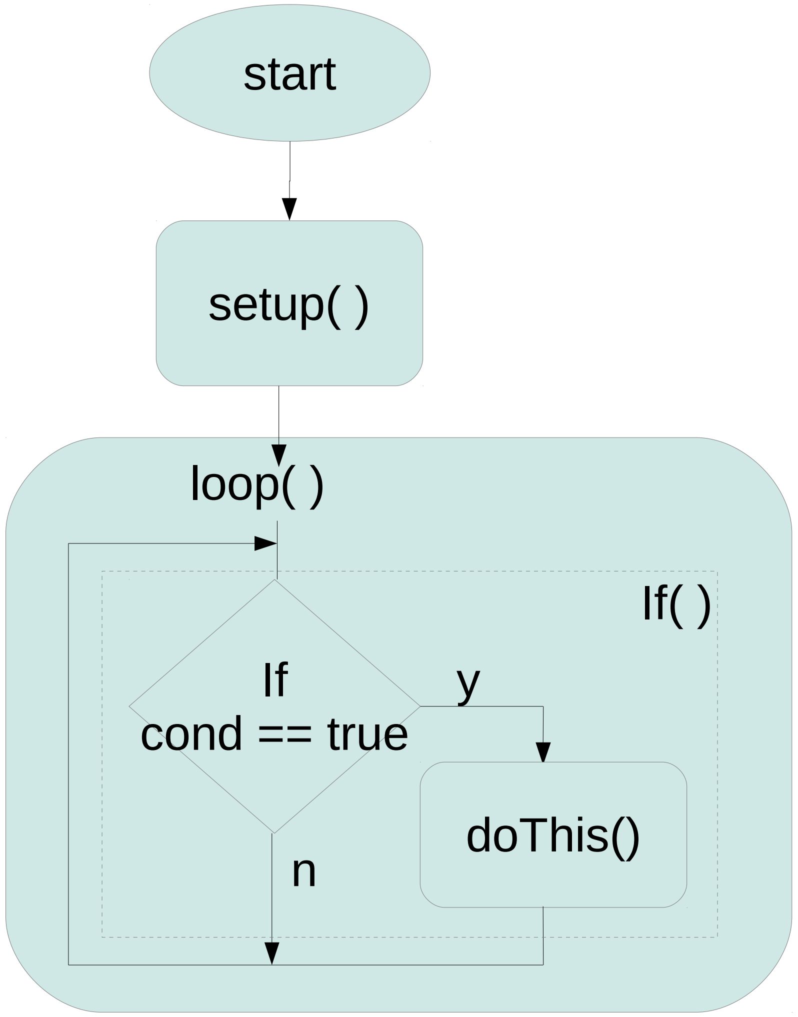
Let's implement these in your Blink

Programming Theory

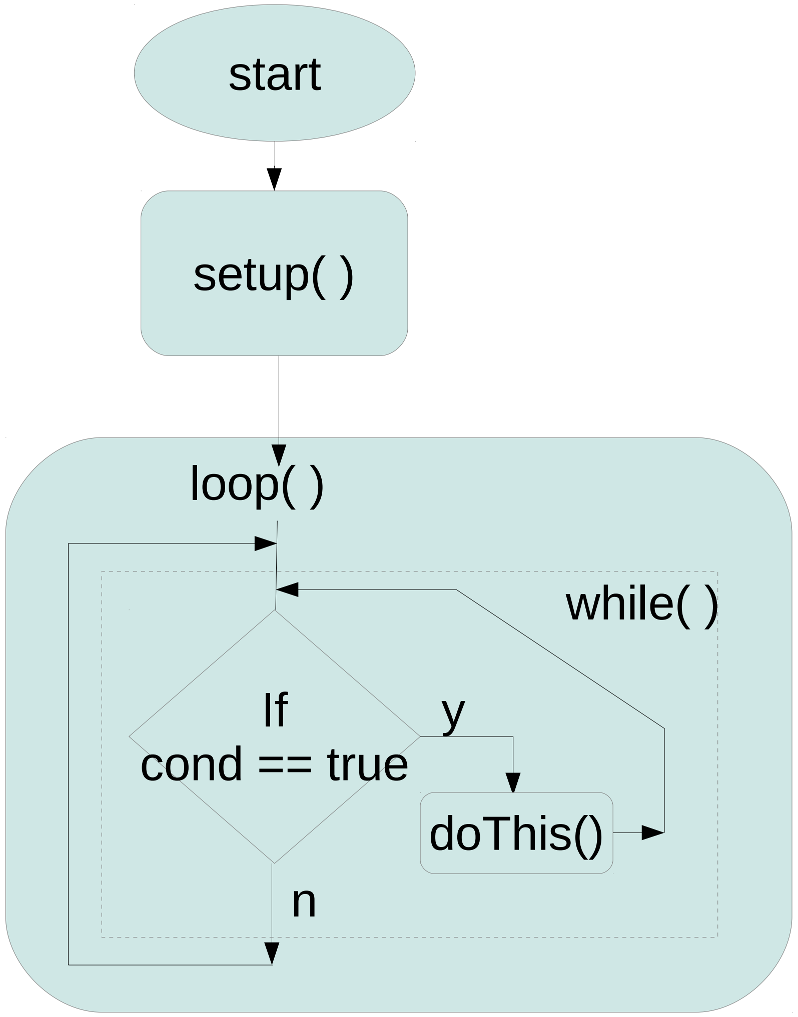
Have Kids Do Simple Paper Flowcharts Before Coding...

Programming Theory

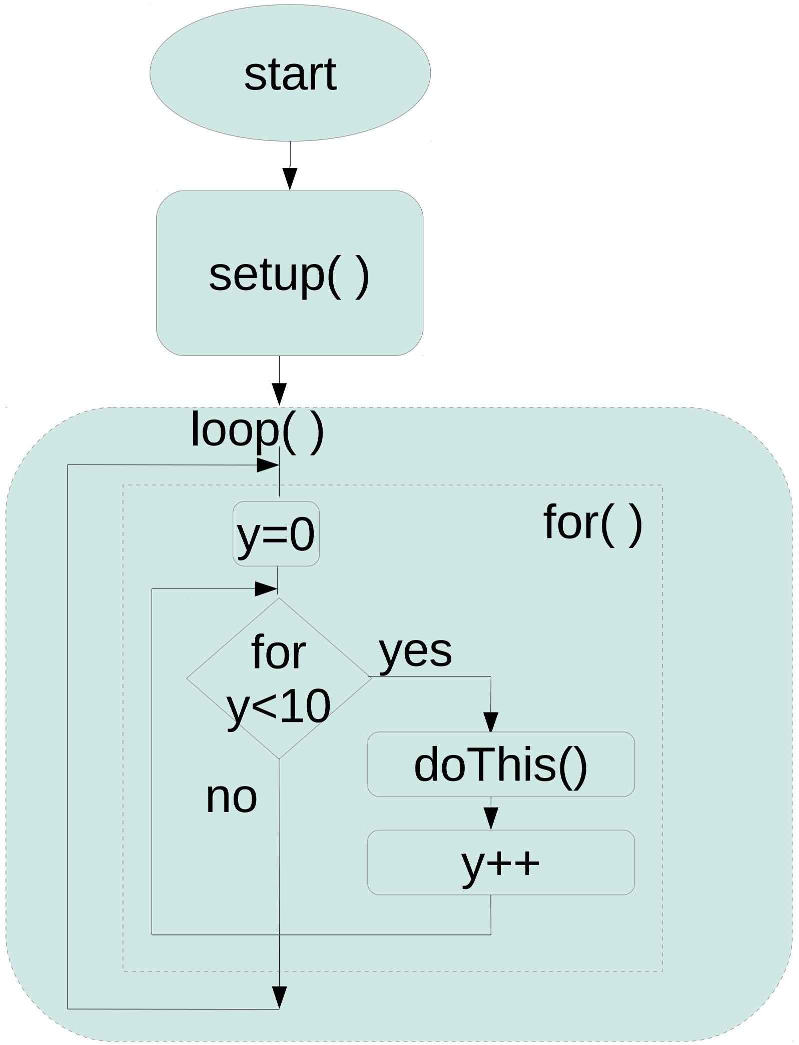
If()



while()



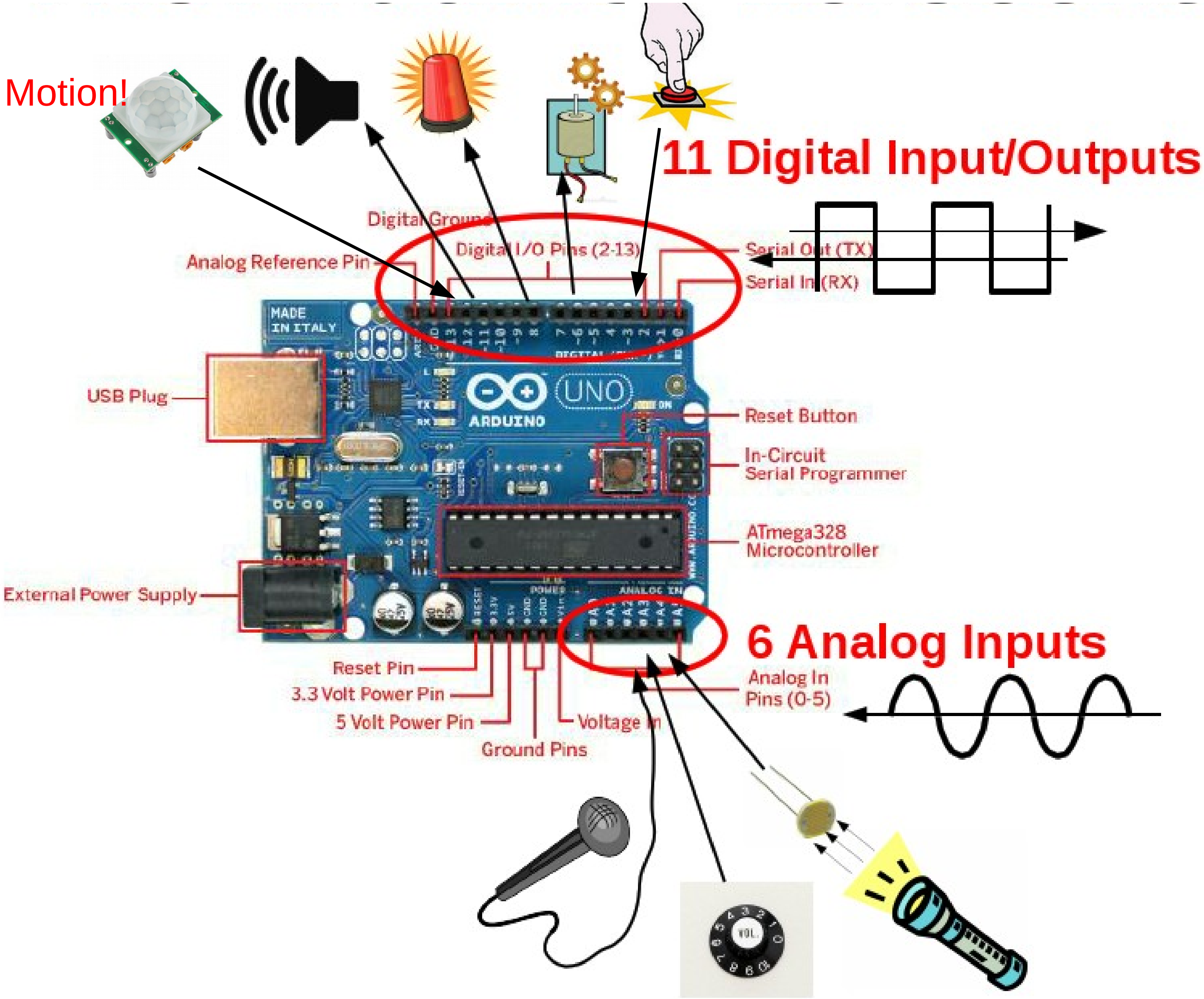
for()



Inputs vs Outputs

- About Let's Code Blacksburg!
- About Teaching STEM Using The Arduino
- Jumping In – Arduino Overview & Running “Blink” Recipe
- Teaching Programming Theory (C++)
- ➡ Inputs vs Outputs
- Meal → “Output: Blink” + “INPUT: Button”
- Meal - Alarm On/Off → Combine Button + Sound Generation
- Meal - Proximity Alarm → Add Ultrasonic Range Sensor
- Arduino Resources

Inputs vs Outputs



Motion!

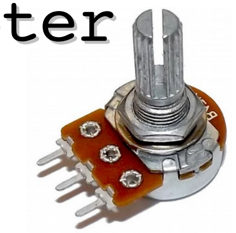
11 Digital Input/Outputs

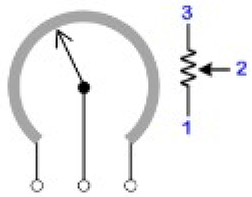
6 Analog Inputs

{oo} Inputs vs Outputs

Inputs/Sensors

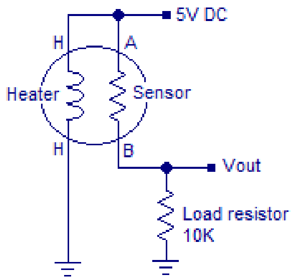
Potentiometer (knob)



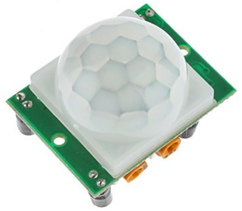


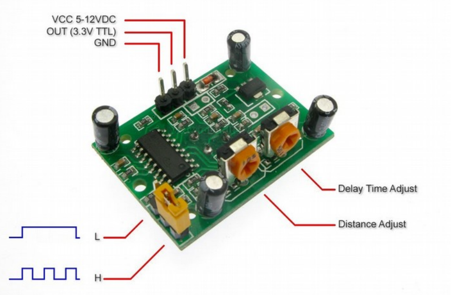
MQ2 Gas Sensor

1 = Output
2 = Vcc (positive voltage)
3 = Gnd

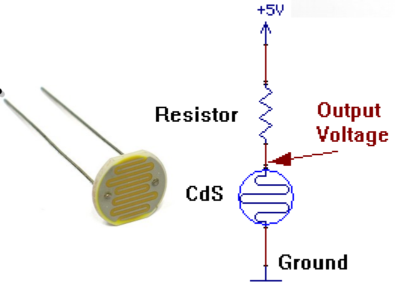


PIR Motion Sensor



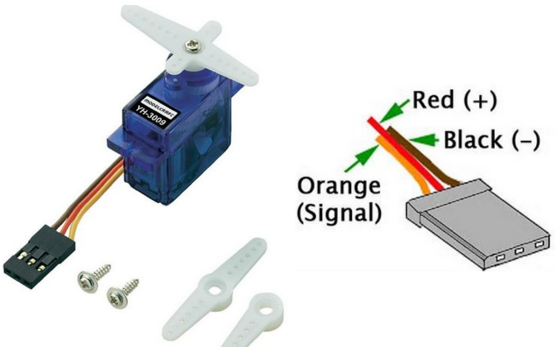


Light Sensor

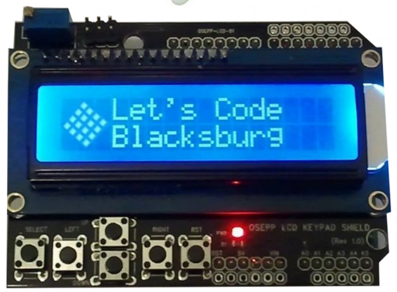


Outputs

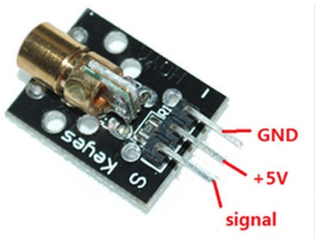
Servo Motor



LCD Screen



Laser Module



IR Transmitter

Programming Theory

- About Let's Code Blacksburg!
- About Teaching STEM Using The Arduino
- Jumping In – Arduino Overview & Running “Blink” Recipe
- Teaching Programming Theory (C++)
- Inputs vs Outputs
- ➜ Meal → “Output: Blink” + “INPUT: Button”
- Meal - Alarm On/Off → Combine Button + Sound Generation
- Meal - Proximity Alarm → Add Ultrasonic Range Sensor
- Arduino Resources

{oo} Programming Theory

- About Let's Code Blacksburg!
- About Teaching STEM Using The Arduino
- Jumping In – Arduino Overview & Running “Blink” Recipe
- Teaching Programming Theory (C++)
- Inputs vs Outputs
- Meal → “Output: Blink” + “INPUT: Button”
- ➔ Meal - Alarm On/Off → Combine Button + Sound Generation
- Meal - Proximity Alarm → Add Ultrasonic Range Sensor
- Arduino Resources

Programming Theory

- About Let's Code Blacksburg!
- About Teaching STEM Using The Arduino
- Jumping In – Arduino Overview & Running “Blink” Recipe
- Teaching Programming Theory (C++)
- Inputs vs Outputs
- Meal → “Output: Blink” + “INPUT: Button”
- Meal - Alarm On/Off → Combine Button + Sound Generation
- ➡ Meal - Proximity Alarm → Add Ultrasonic Range Sensor
- Arduino Resources

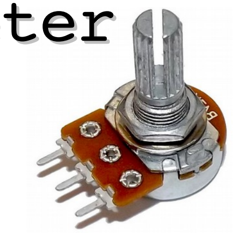
{OO} Programming Theory

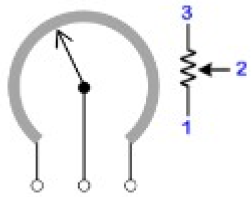
- About Let's Code Blacksburg!
- About Teaching STEM Using The Arduino
- Jumping In – Arduino Overview & Running “Blink” Recipe
- Teaching Programming Theory (C++)
- Inputs vs Outputs
- Meal → “Output: Blink” + “INPUT: Button”
- Meal - Alarm On/Off → Combine Button + Sound Generation
- Meal - Proximity Alarm → Add Ultrasonic Range Sensor
- Arduino Resources

Inputs vs Outputs

Inputs/Sensors

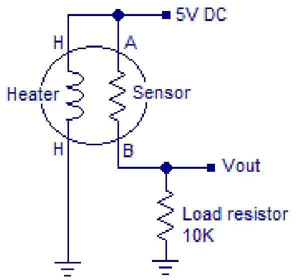
Potentiometer (knob)



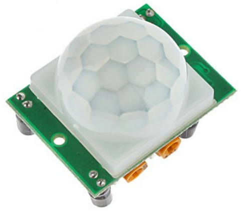


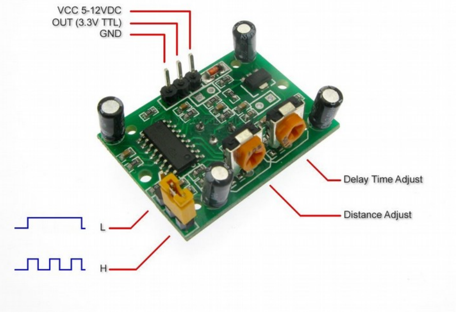
MQ2 Gas Sensor

1 = Output
2 = Vcc (positive voltage)
3 = Gnd

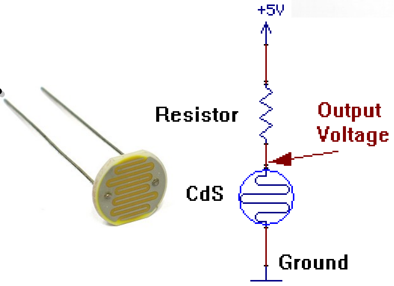


PIR Motion Sensor



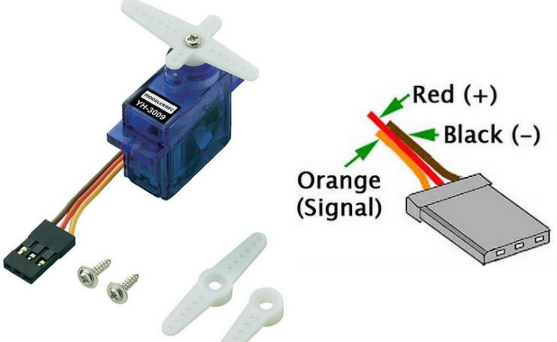


Light Sensor

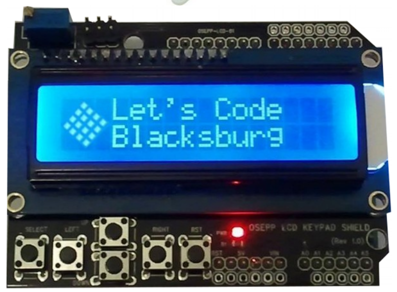


Outputs

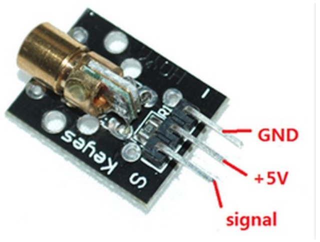
Servo Motor



LCD Screen



Laser Module



IR Transmitter

Arduino Resources

- **Code/Activities, LCBB Github:** https://github.com/LetsCodeBlacksburg/
- **Questions, drivers, learning:** http://playground.arduino.cc/
- **Arduino projects/lessons:** learn.adafruit.com, toptechboy.com/arduino-lessons
- **Arduino Parts:** amazon.com, mpja.com, sparkfun.com, aliexpress.com
- **Robotics:** robotshop.com, aliexpress.com

Day-2 Overview

- More C++ Programming Theory: Functions & Parameters
- Build – “Sibling Detector Alarm” using three recipes:
    - OUTPUT – Sound Buzzer
    - INPUT – People Motion Sensor
    - INPUT/OUTPUT – LCD Keypad Shield

- Teaching Robotics Based Workshops
- Arduino Resources
- Let's Code Blacksburg Resources

C++, Functions & Parameters

- More C++ Programming Theory: Functions & Parameters
- Build – “Sibling Detector Alarm” using three recipes:
  - OUTPUT – Sound Buzzer
  - INPUT – People Motion Sensor
  - INPUT/OUTPUT – LCD Keypad Shield
- Teaching Robotics Based Workshops
- Arduino Resources
- Let's Code Blacksburg Resources

Build “Sibling Detector”

- More C++ Programming Theory: Functions & Parameters
- Build – “Sibling Detector Alarm” using three recipes:
  - OUTPUT – Sound Buzzer
  - INPUT – People Motion Sensor
  - INPUT/OUTPUT – LCD Keypad Shield
- Teaching Robotics Based Workshops
- Arduino Resources
- Let's Code Blacksburg Resources

Robotics Workshops

- More C++ Programming Theory: Functions & Parameters
- Build – “Sibling Detector Alarm” using three recipes:
  - OUTPUT – Sound Buzzer
  - INPUT – People Motion Sensor
  - INPUT/OUTPUT – LCD Keypad Shield
- → Teaching Robotics Based Workshops
- Arduino Resources
- Let's Code Blacksburg Resources

Arduino Resources

- More C++ Programming Theory: Functions & Parameters
- Build – “Sibling Detector Alarm” using three recipes:
  - OUTPUT – Sound Buzzer
  - INPUT – People Motion Sensor
  - INPUT/OUTPUT – LCD Keypad Shield
- Teaching Robotics Based Workshops
- Arduino Resources
- Let's Code Blacksburg Resources

Arduino Resources

- **Code/Activities, LCBB Github:** https://github.com/LetsCodeBlacksburg/
- **Questions, drivers, learning:** http://playground.arduino.cc/
- **Arduino projects/lessons:** learn.adafruit.com, toptechboy.com/arduino-lessons
- **Arduino Parts:** amazon.com, mpja.com, sparkfun.com, aliexpress.com
- **Robotics:** robotshop.com, aliexpress.com

LCBB Resources

- More C++ Programming Theory: Functions & Parameters
- Build – “Sibling Detector Alarm” using three recipes:
    - OUTPUT – Sound Buzzer
    - INPUT – People Motion Sensor
    - INPUT/OUTPUT – LCD Keypad Shield
- Teaching Robotics Based Workshops
- Arduino Resources
- Let's Code Blacksburg Resources