

A decorative graphic on the left side of the slide, consisting of a network of white and light blue lines and circles, resembling a circuit board or a data network. The lines are vertical and horizontal, with some diagonal connections, and the circles are of varying sizes, some acting as nodes or junctions.

ARDUINO WORKSHOP

Thomas Trickel

A decorative graphic on the left side of the slide, consisting of a network of light blue lines and circles that resemble a circuit board or a neural network. The lines are of varying thickness and connect to small circles at various points, creating a complex, branching structure.

ARDUINO WORKSHOP

Welcome

&

Introductions

Thomas Trickel

A decorative graphic on the left side of the slide, consisting of a network of white and light blue lines and circles, resembling a circuit board or a neural network. The lines are vertical and horizontal, with some diagonal connections, and the circles are of varying sizes, some acting as nodes or junctions.

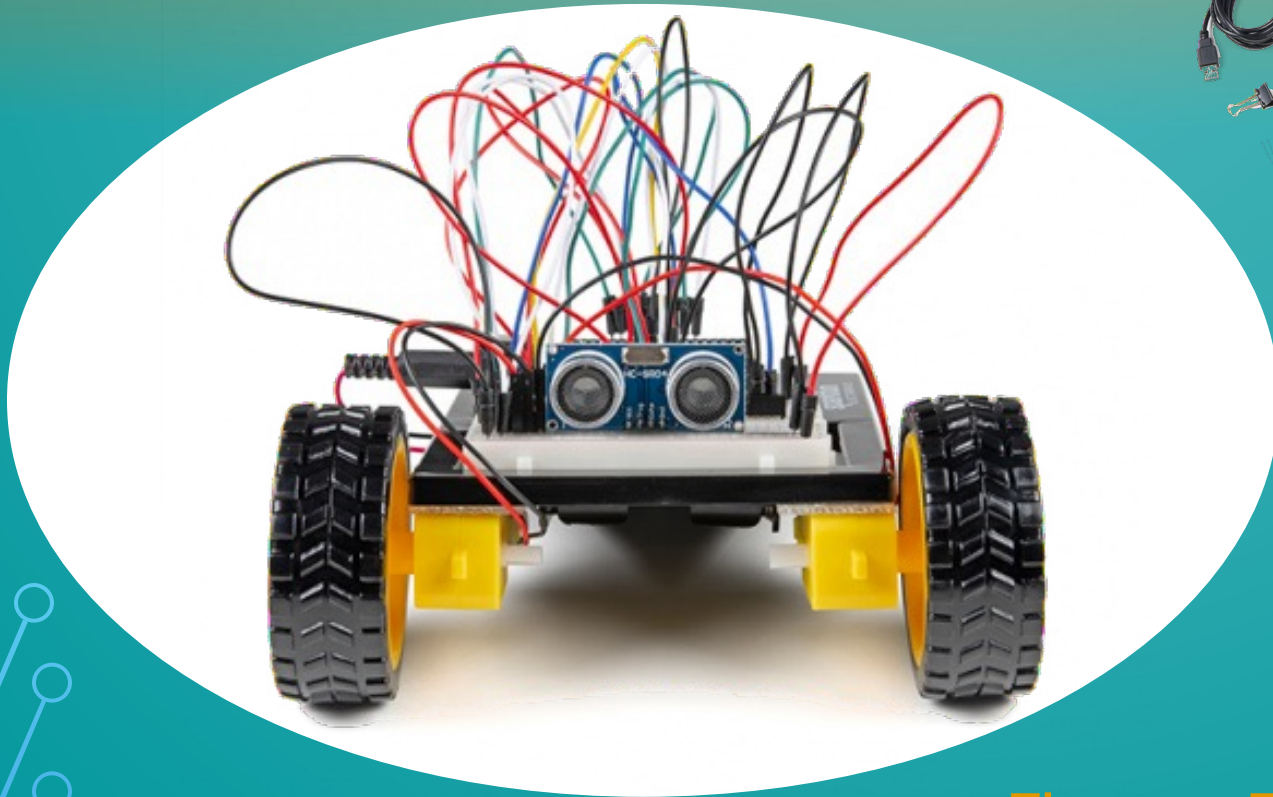
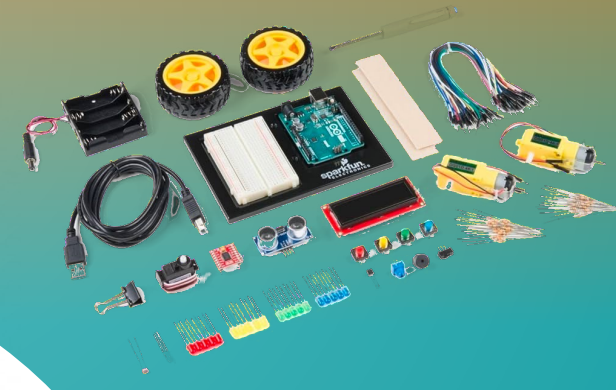
ARDUINO WORKSHOP

What Are We Going to Do?

Thomas Trickel

ARDUINO WORKSHOP

SparkFun Inventor's Kit

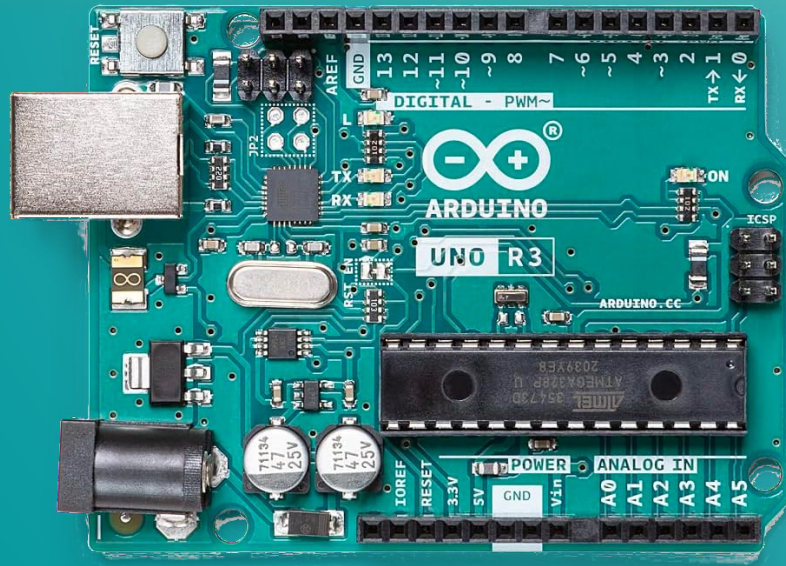


Thomas Trickel

The background features a gradient from orange at the top to teal at the bottom. White circuit-like lines with circular nodes are positioned in the corners: top-left, top-right, bottom-left, and bottom-right.

WHAT IS AN ARDUINO?

WHAT IS AN ARDUINO?



```
Blink | Arduino 1.8.5

This example code is in the public domain.

http://www.arduino.cc/en/Tutorial/Blink
*/

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

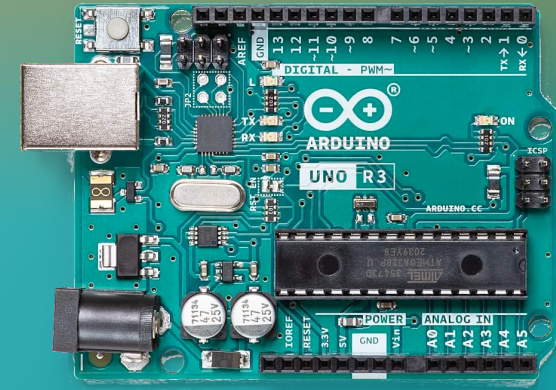
// the loop function runs over and over again forever
void loop() {$
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second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

32 Arduino/Genuino Uno on COM1
```



ARDUINO IS TWO PIECES?

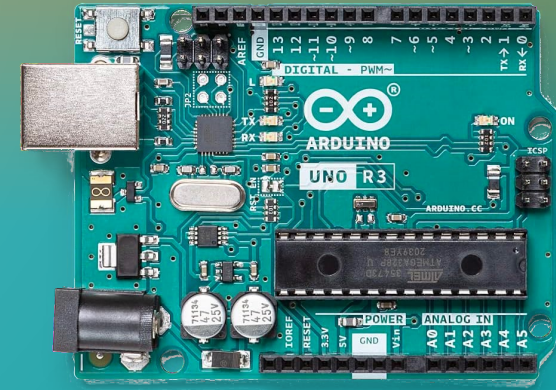
- Something you can touch and feel – the Hardware





ARDUINO IS TWO PIECES?

- Something you can touch and feel – the Hardware
- A program that allows us to
 - write programs to control the Hardware
 - copy those programs to the Hardware



```
Blink | Arduino 1.8.5
Blink $
This example code is in the public domain.
http://www.arduino.cc/en/Tutorial/Blink
*/
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}
// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
32 Arduino/Genuino Uno on COM1
```

WHAT IS AN ARDUINO?

- A simple, customizable computer
- Computers take in inputs, perform operations, and generate outputs based on how they're programmed
- Examples
 - Inputs: pressing buttons, reading a sensor, code from a programmer
 - Operations: doing math, moving and keeping track of information
 - Outputs: blinking light, sound, driving a motor

LET'S GET STARTED



LET'S GET STARTED

wait

The background features a vertical gradient from orange at the top to teal at the bottom. In each of the four corners, there are white, stylized circuit board traces that branch out and terminate in small circles, resembling nodes or components on a PCB.

A FEW THINGS

A FEW THINGS

- Computers are very complex
 - Don't expect success on the first try

- Ask for help



A FEW THINGS

- Get programs from SparkFun Inventor's Kit Examples Folder

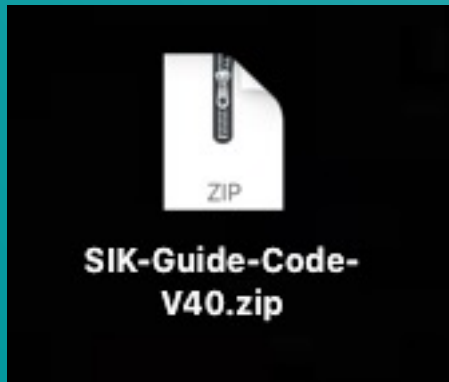
trickel.org/thomas/skc

- Get programs from SparkFun Inventor's Kit Examples Folder

trickel.org/thomas/skc

Match Your Book

Right click and save to Desktop
Go to Desktop
Double click on File



2024 Techcamp

- [2024 Tech Camp Presentation St. Ignatius](#)
- [2024 Tech Camp Presentation Ronan](#)
- [SparkFun Inventor's Kit Arduino Code V4.0](#)
- [SparkFun Inventor's Kit Arduino Code V4.1](#)
- [SparkFun Inventor's Kit Guide Book V4.1](#)
- [SparkFun Inventor's Kit LCD Errata](#)

2023 Spring Technology in the Classroom Showcase

- [littleBits](#)
- [Raspberry Pi](#)

Other Resources

- [Eagle Design Libraries](#)
- [ENGG Code](#)

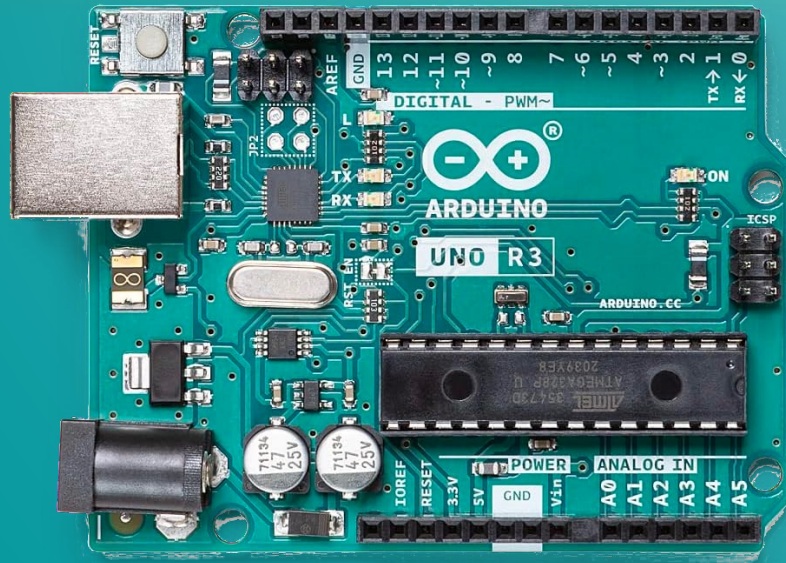
LET'S GET STARTED

- What do we need?





WHAT IS AN ARDUINO?



```
Blink | Arduino 1.8.5

This example code is in the public domain.

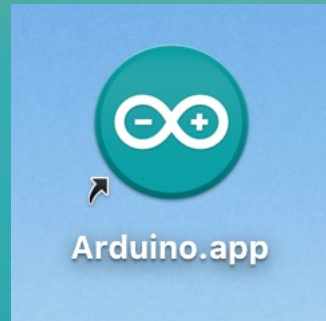
http://www.arduino.cc/en/Tutorial/Blink
*/

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

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

32 Arduino/Genuino Uno on COM1
```

ARDUINO IDE



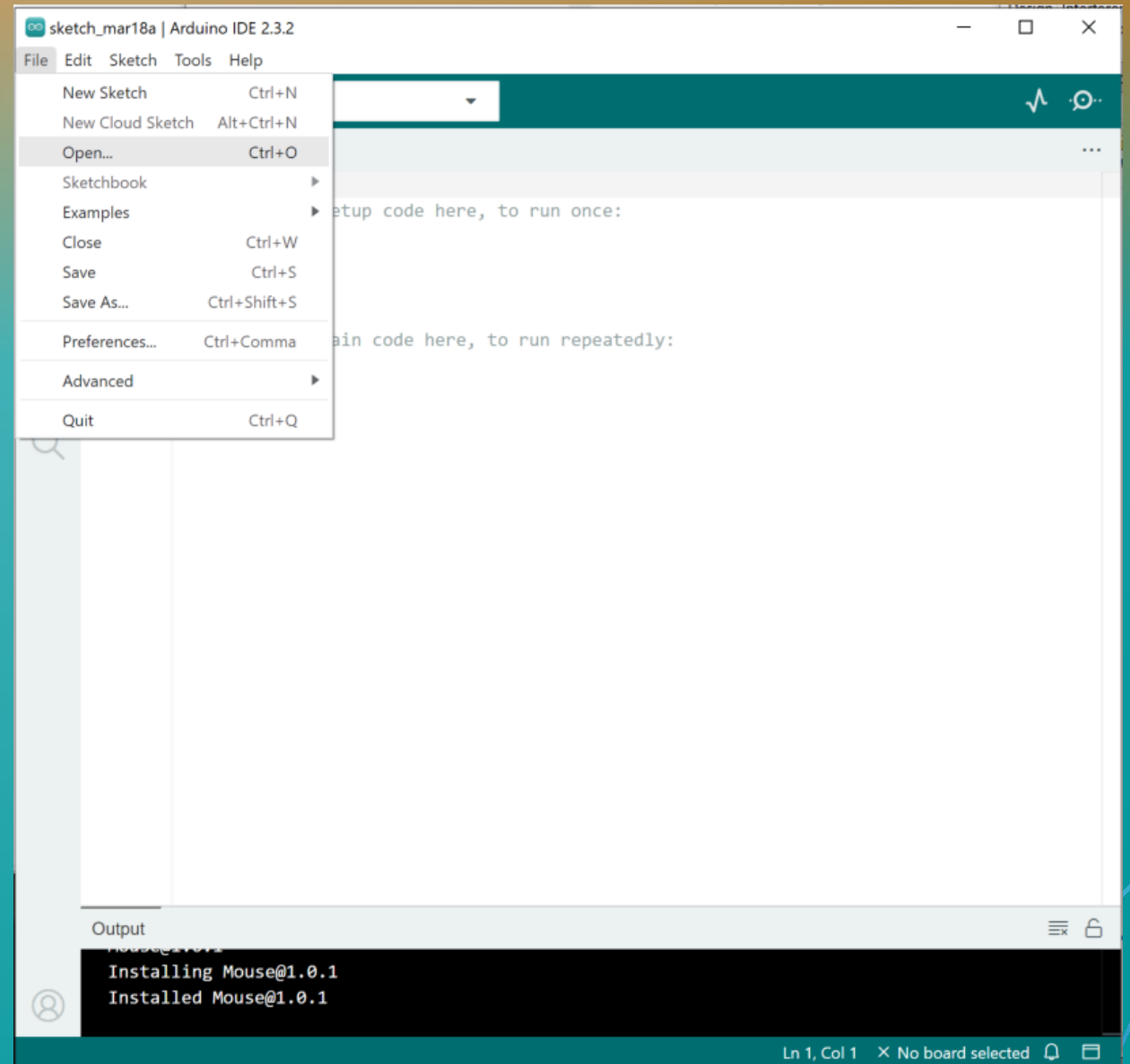
ARDUINO IDE

- File -> Open

SIK-Guide-Code-V40

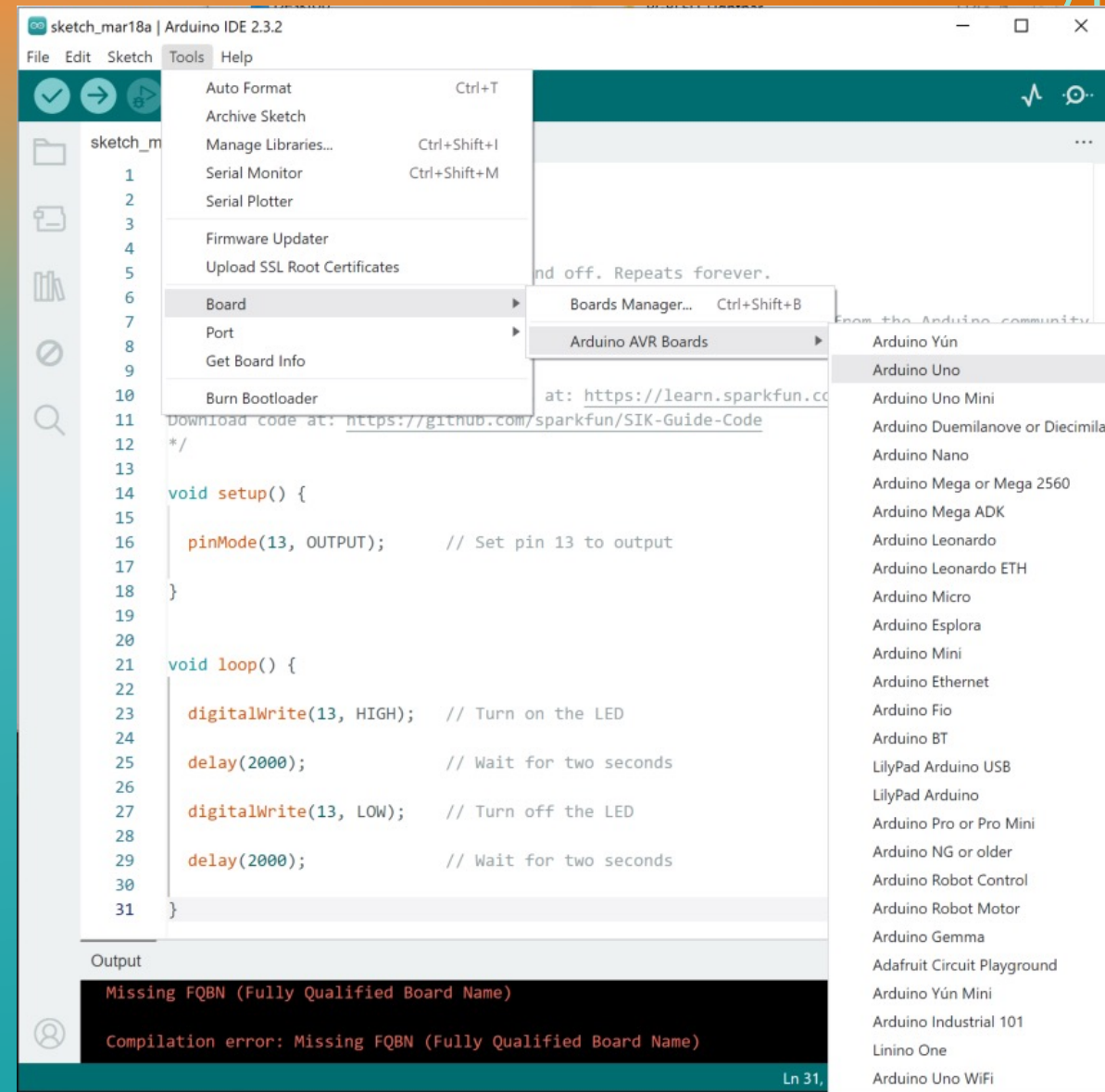
SIK_Circuit_1 A-Blink

SIK_Circuit_1 A-Blink.ino



ARDUINO IDE

- Tools -> Board -> Arduino AVR Boards
Arduino Uno



ARDUINO IDE

- Verify Program

- Output should have no errors

```
sketch_mar18a | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Uno
sketch_mar18a.ino
1  /*
2  SparkFun Inventor's Kit
3  Circuit 1A-Blink
4
5  Turns an LED connected to pin 13 on and off. Repeats forever.
6
7  This sketch was written by SparkFun Electronics, with lots of help from the Arduino community.
8  This code is completely free for any use.
9
10 View circuit diagram and instructions at: https://learn.sparkfun.com/tutorials/sparkfun-inventor
11 Download code at: https://github.com/sparkfun/SIK-Guide-Code
12  */
13
14 void setup() {
15
16     pinMode(13, OUTPUT);    // Set pin 13 to output
17
18 }
19
20
21 void loop() {
22
23     digitalWrite(13, HIGH); // Turn on the LED
24
25     delay(2000);            // Wait for two seconds
26
27     digitalWrite(13, LOW); // Turn off the LED
28
29     delay(2000);            // Wait for two seconds
30
31 }
```

Output

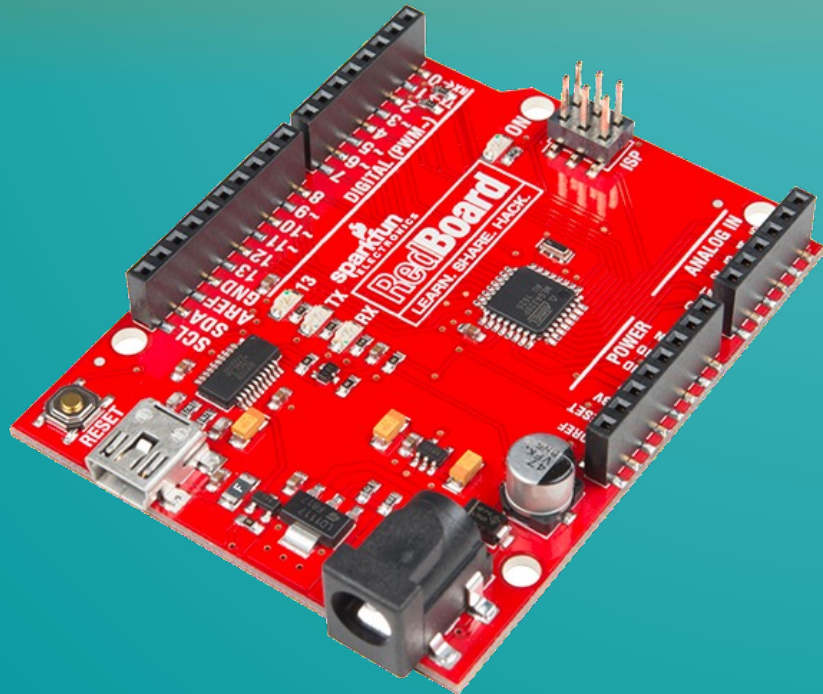
Sketch uses 924 bytes (2%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum

Ln 31, Col 2 Arduino Uno [not connected]

EXCELLENT!

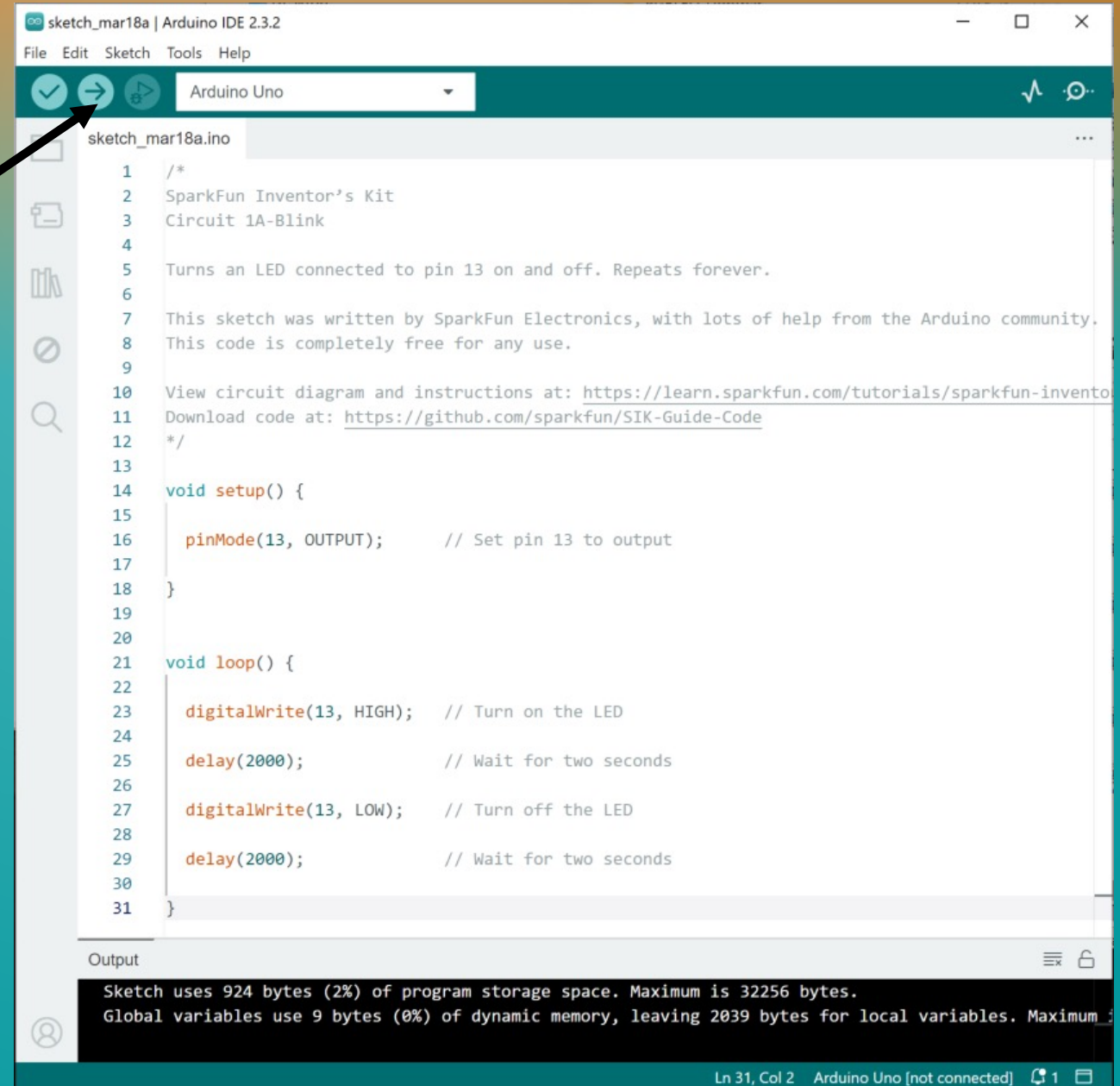


RUN IT ON THE HARDWARE



RUN IT ON THE HARDWARE

- Upload & Run Program



The screenshot shows the Arduino IDE 2.3.2 interface. The top menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu bar is a toolbar with icons for checking, uploading, and running. The 'Run' button (a play icon) is highlighted with a black arrow. The main editor area displays the code for 'sketch_mar18a.ino'. The code includes a comment describing the circuit and a loop that blinks an LED on pin 13. The output window at the bottom shows the compilation results.

```
sketch_mar18a.ino
1  /*
2  SparkFun Inventor's Kit
3  Circuit 1A-Blink
4
5  Turns an LED connected to pin 13 on and off. Repeats forever.
6
7  This sketch was written by SparkFun Electronics, with lots of help from the Arduino community.
8  This code is completely free for any use.
9
10 View circuit diagram and instructions at: https://learn.sparkfun.com/tutorials/sparkfun-inventor
11 Download code at: https://github.com/sparkfun/SIK-Guide-Code
12 */
13
14 void setup() {
15
16   pinMode(13, OUTPUT);    // Set pin 13 to output
17
18 }
19
20
21 void loop() {
22
23   digitalWrite(13, HIGH); // Turn on the LED
24
25   delay(2000);            // Wait for two seconds
26
27   digitalWrite(13, LOW);  // Turn off the LED
28
29   delay(2000);            // Wait for two seconds
30
31 }
```

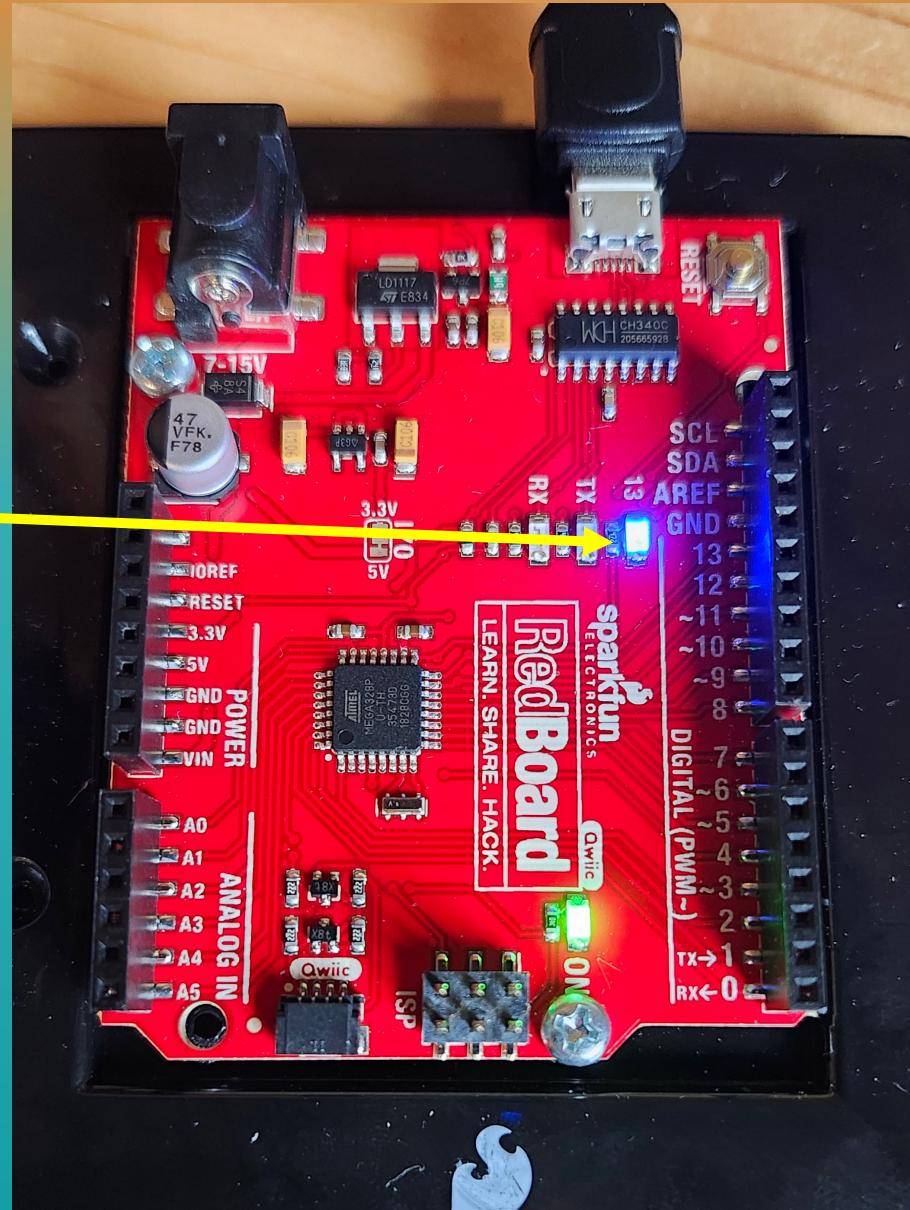
Output

```
Sketch uses 924 bytes (2%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum
```

Ln 31, Col 2 Arduino Uno [not connected] 1

WHAT YOU SHOULD SEE

- Blue LED should blink



EXCELLENT!





FIRST CIRCUIT

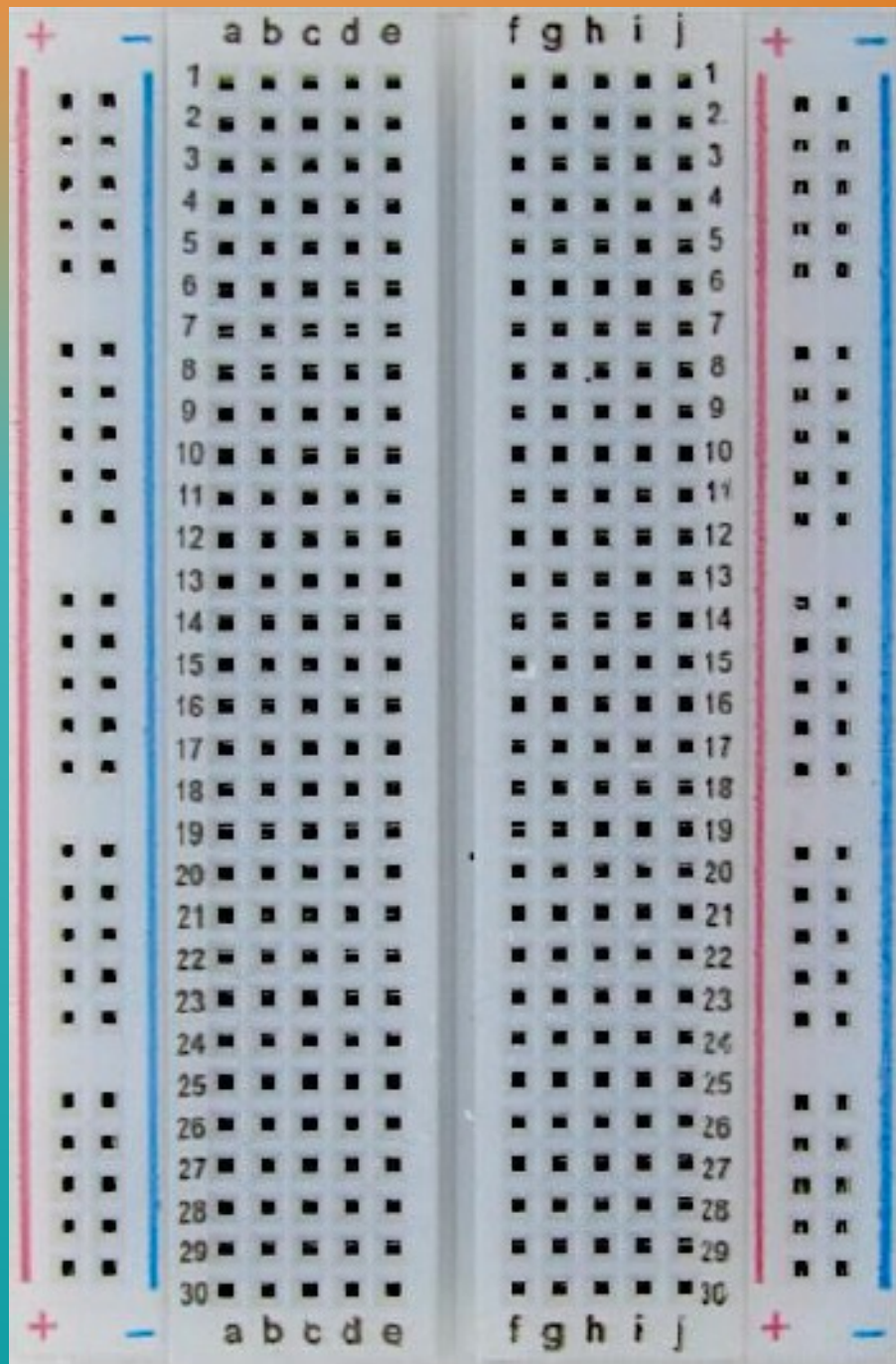


BEFORE WE BEGIN



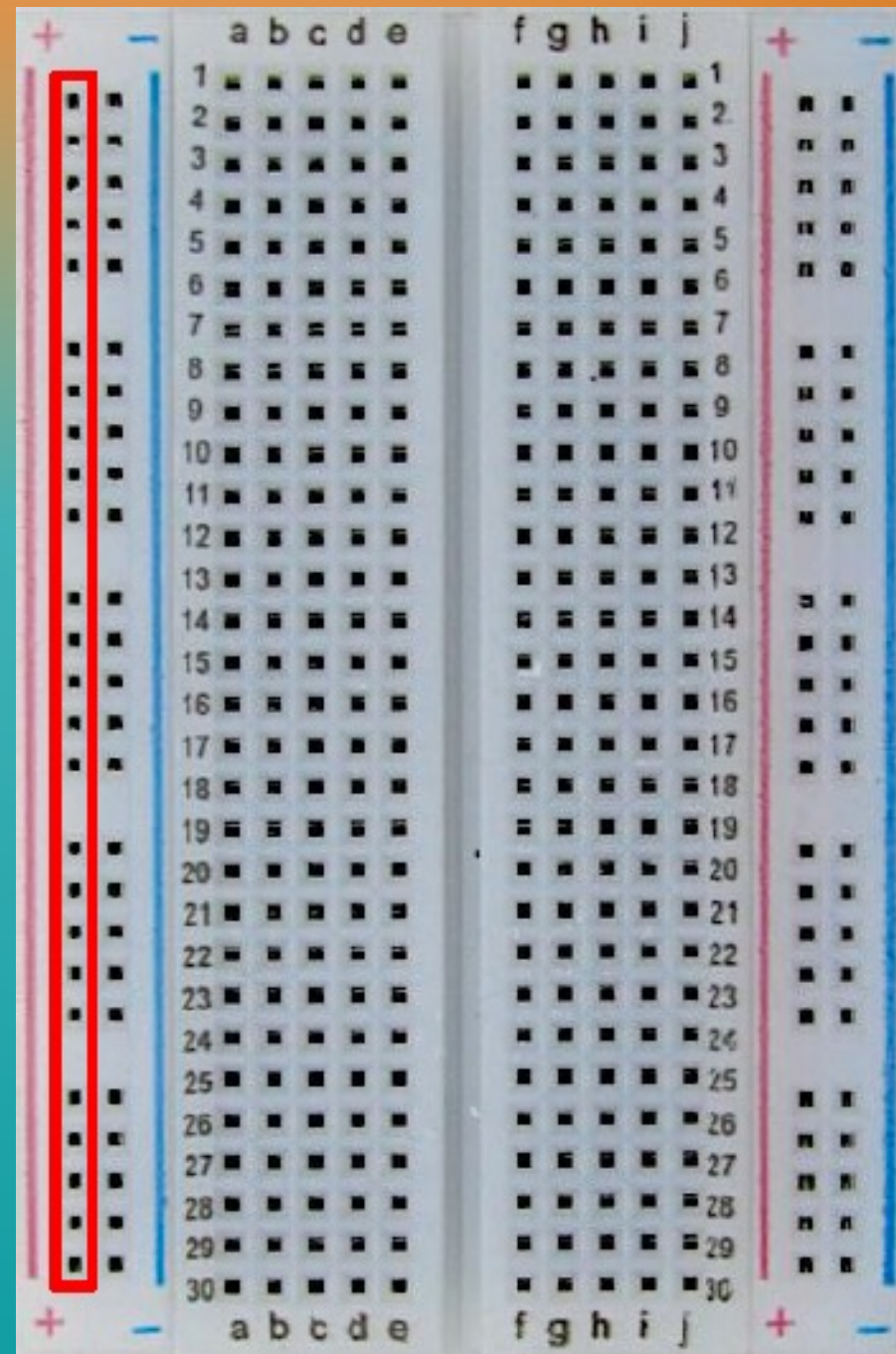
FIRST CIRCUIT

- Breadboard



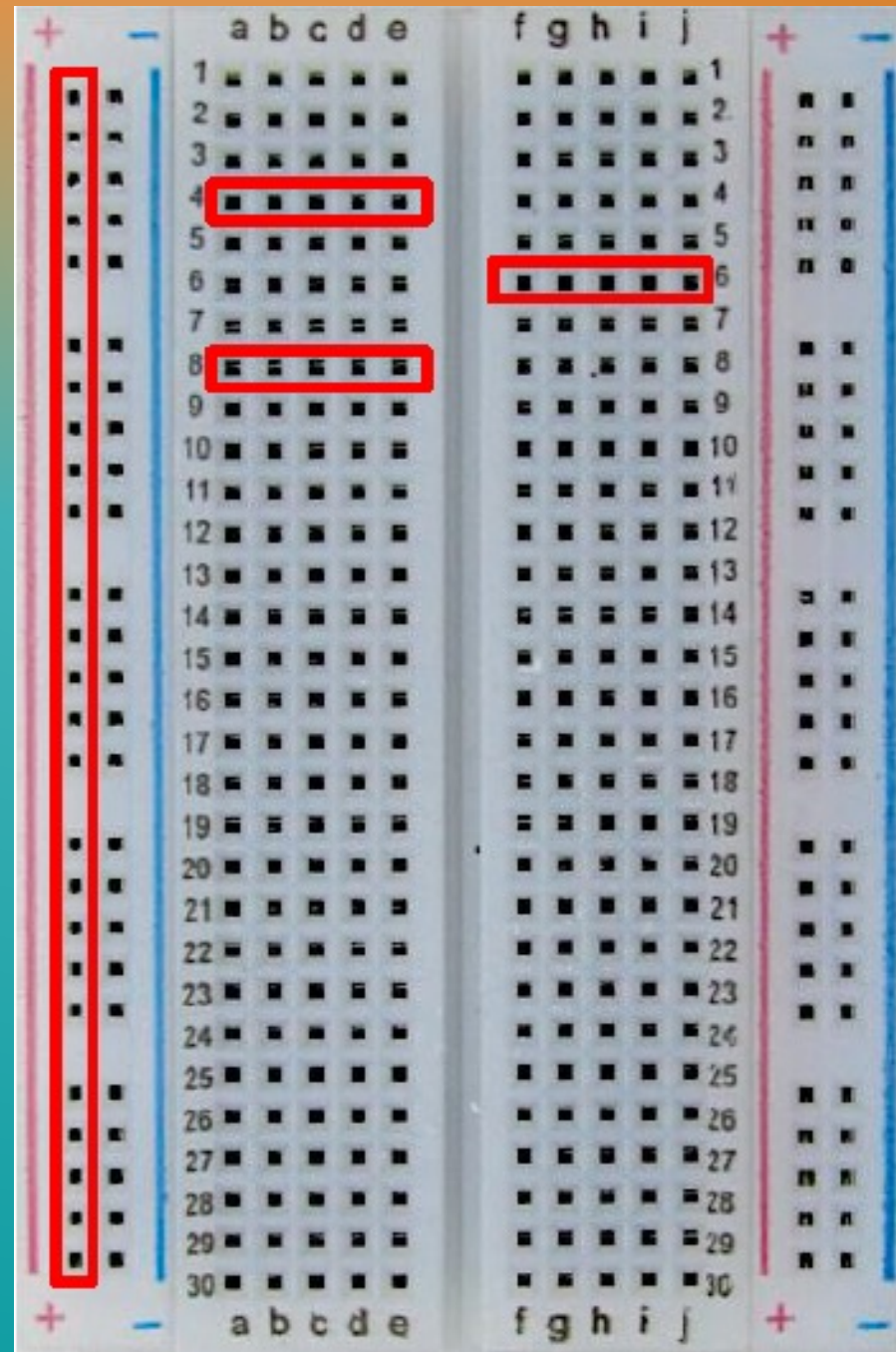
FIRST CIRCUIT

- Breadboard



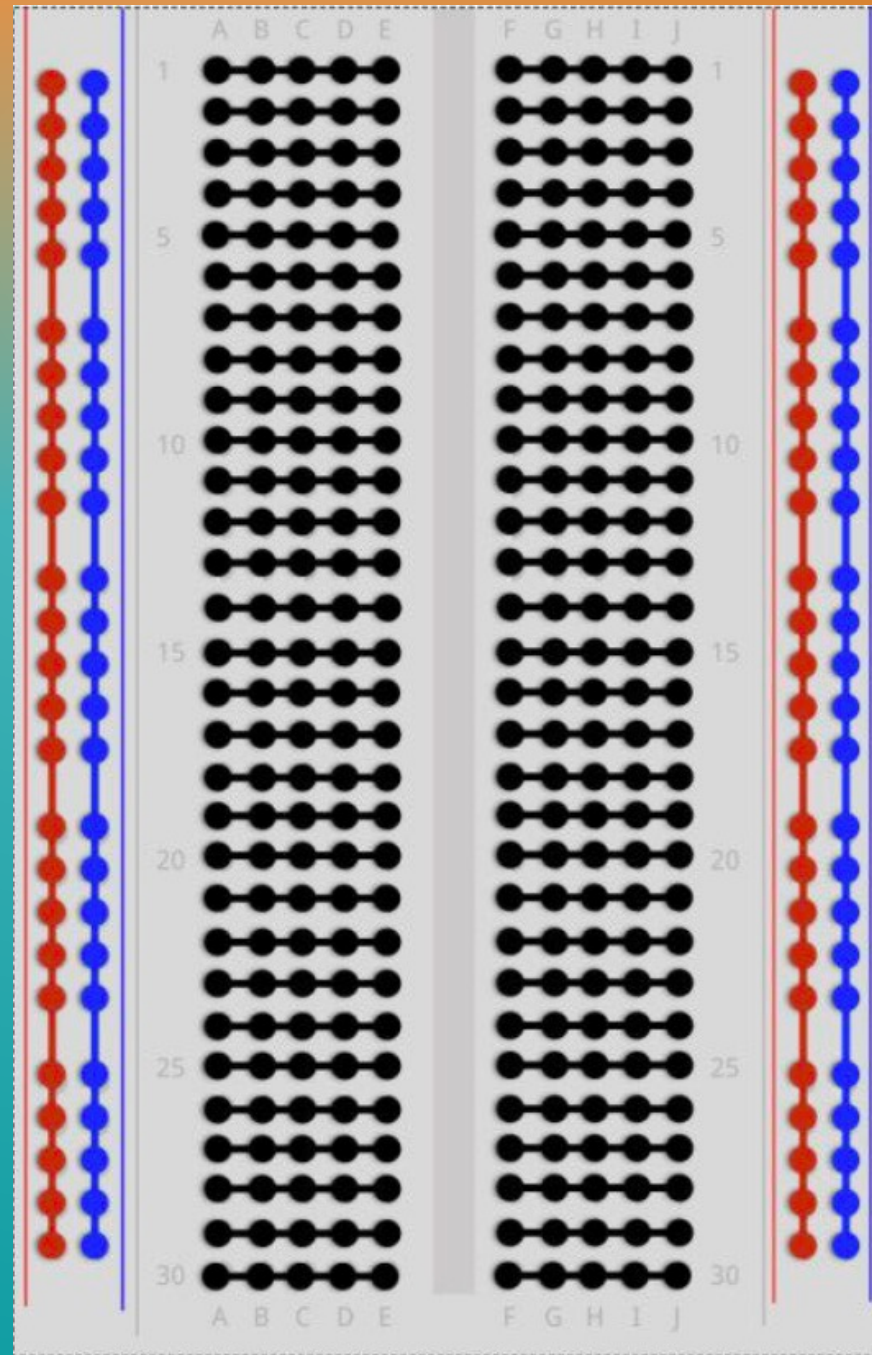
FIRST CIRCUIT

- Breadboard



FIRST CIRCUIT

- Breadboard



THE BOOK

Contents

INTRODUCTION 2

- 2** The RedBoard Platform
- 3** Baseplate Assembly
- 4** RedBoard Anatomy
- 5** Breadboard Anatomy
- 6** The Arduino IDE
- 10** Inventory of Parts

PROJECT 1: LIGHT 12

- 13** Circuit 1A: Blinking an LED
- 20** Circuit 1B: Potentiometer
- 26** Circuit 1C: Photoresistor
- 31** Circuit 1D: RGB Night-Light

PROJECT 2: SOUND 36

- 37** Circuit 2A: Buzzer
- 42** Circuit 2B: Digital Trumpet
- 47** Circuit 2C: "Simon Says" Game

PROJECT 3: MOTION 53

- 54** Circuit 3A: Servo Motors
- 60** Circuit 3B: Distance Sensor
- 65** Circuit 3C: Motion Alarm

PROJECT 4: DISPLAY 71

- 72** Circuit 4A: LCD "Hello, World!"
- 77** Circuit 4B: Temperature Sensor
- 82** Circuit 4C: "DIY Who Am I?" Game

PROJECT 5: ROBOT 88

- 89** Circuit 5A: Motor Basics
- 96** Circuit 5B: Remote-Controlled Robot
- 102** Circuit 5C: Autonomous Robot

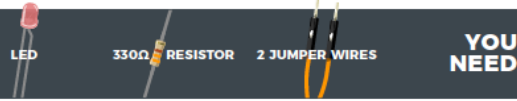
GOING FURTHER 106

THE BOOK

INTRODUCTORY SECTION

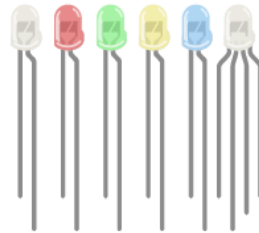
Circuit 1A: Blinking an LED

You can find LEDs in just about any source of light, from the bulbs lighting your home to the tiny status lights flashing on your home electronics. Blinking an LED is the classic starting point for learning how to program embedded electronics. It's the "Hello, World!" of microcontrollers. In this circuit, you'll write code that makes an LED blink on and off.



NEW COMPONENTS

LIGHT-EMITTING DIODES (LEDs) are small lights made from a silicon diode. They come in different colors, brightnesses and sizes. LEDs (pronounced el-ee-dees) have a positive (+) leg and a negative (-) leg, and they will only let electricity flow through them in one direction. LEDs can also burn out if too much electricity flows through them, so you should always use a resistor to limit the current when you wire an LED into a circuit.



RESISTORS resist the flow of electricity. You can use them to protect sensitive components like LEDs. The strength of a resistor (measured in ohms) is marked on the body of the resistor using small colored bands. Each color stands for a number, which you can look up using a resistor chart. One can be found at the back of this book.

NEW CONCEPTS

POLARITY: Many electronics components have polarity, meaning electricity can (and should) flow through them in only one direction. Polarized components, like an LED, have a positive and a negative leg and only work when electricity flows through them in one direction. Some components, like resistors, do not have polarity; electricity can flow through them in either direction.



OHM'S LAW describes the relationship between the three fundamental elements of electricity: **voltage**, **resistance** and **current**. This relationship can be represented by this equation:

$$V=I \cdot R$$

V = Voltage in volts

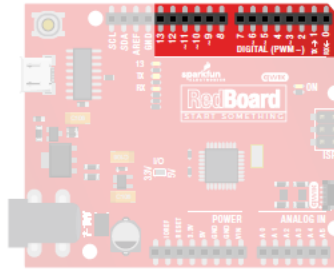
I = Current in amps

R = Resistance in ohms (Ω)

This equation is used to calculate what resistor values are suitable to sufficiently limit the current flowing to the LED so that it does not get too hot and burn out.

THE BOOK

INTRODUCTORY SECTION



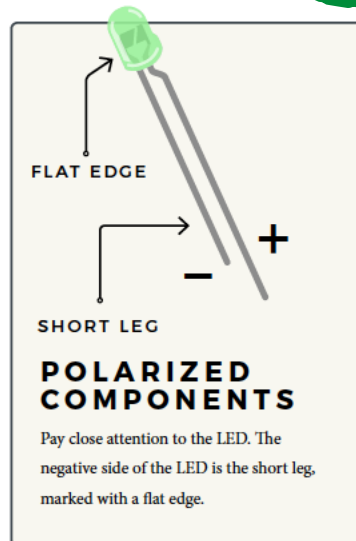
DIGITAL OUTPUT: When working with microcontrollers such as the RedBoard, there are a variety of pins to which you can connect electronic components. Knowing which pins perform which functions is important when building your circuit. In this circuit, we will be using what is known as a digital output. There are 14 of these pins found on the RedBoard. A digital output only has two states: **ON** or **OFF**. These two states can also be thought of

as **HIGH** or **LOW**, **TRUE** or **FALSE**. When an LED is connected to one of these pins, the pin can only perform two jobs: turning on the LED and turning off the LED. We'll explore the other pins and their functions in later circuits.

NEW IDEAS

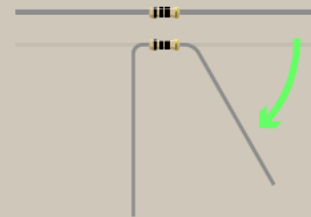
ELECTRICAL SAFETY: Never work on your circuits while the board is connected to a power source. The SparkFun RedBoard operates at 5 volts, which, while not enough to injure you, is enough to damage the components in your circuit.

COMPONENT ORIENTATION & POLARITY: Instructions on how to orient each of the new components will be given before each circuit diagram. Many components have polarity and have only one correct orientation, while others are nonpolarized.



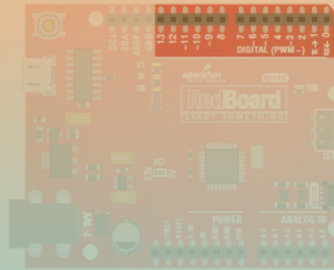
RESISTOR LEADS

Components like resistors need to have their legs bent into 90° angles in order to correctly fit in the breadboard sockets.



THE BOOK

INTRODUCTORY SECTION

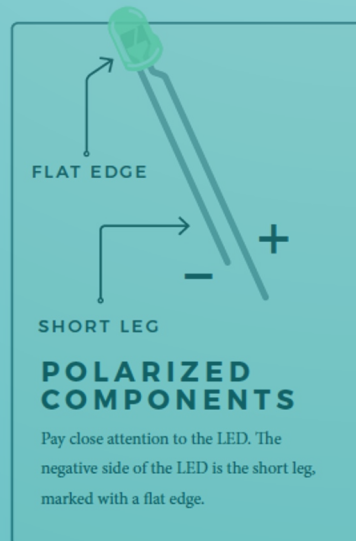


DIGITAL OUTPUT: When working with microcontrollers such as the RedBoard, there are a variety of pins to which you can connect electronic components. Knowing which pins perform which functions is important when building your circuit. In this circuit, we will be using what is known as a digital output. There are 14 of these pins found on the RedBoard. A digital output only has two states: ON or OFF. These two states can also be thought of

as HIGH or LOW, TRUE or FALSE. When an LED is connected to one of these pins, the pin can only perform two jobs: turning on the LED and turning off the LED. We'll explore the other pins and their functions in later circuits.

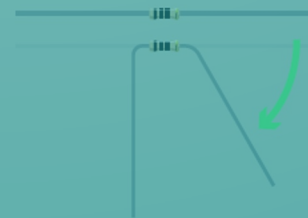
NEW IDEAS

ELECTRICAL SAFETY: Never work on your circuits while the board is connected to a power source. The SparkFun RedBoard operates at 5 volts, which, while not enough to injure you, is enough to damage the components in your circuit.



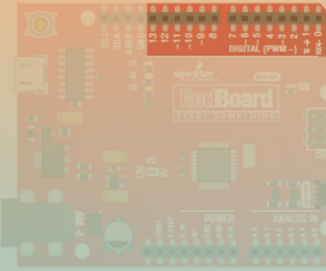
RESISTOR LEADS

Components like resistors need to have their legs bent into 90° angles in order to correctly fit in the breadboard sockets.



THE BOOK

INTRODUCTORY SECTION



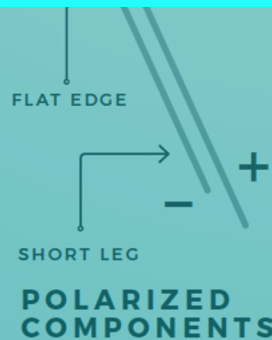
DIGITAL OUTPUT: When working with microcontrollers such as the RedBoard, there are a variety of pins to which you can connect electronic components. Knowing which pins perform which functions is important when building your circuit. In this circuit, we will be using what is known as a digital output. There are 14 of these pins found on the RedBoard. A digital output only has two states: ON or OFF. These two states can also be thought of

as HIGH or LOW, TRUE or FALSE. When an LED is connected to one of these pins, the pin can only perform two jobs: turning on the LED and turning off the LED. We'll explore the other pins and their functions in later circuits.

NEW IDEAS

ELECTRICAL SAFETY: Never work on your circuits while the board is connected to a power source. The SparkFun RedBoard operates at 5 volts, which, while not enough to injure you, is enough to damage the components in your circuit.

- DISCONNECT USB Cable before wiring circuits

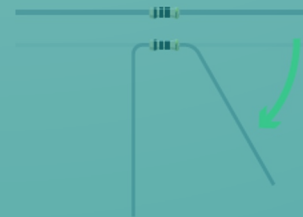


POLARIZED COMPONENTS

Pay close attention to the LED. The negative side of the LED is the short leg, marked with a flat edge.

LEADS

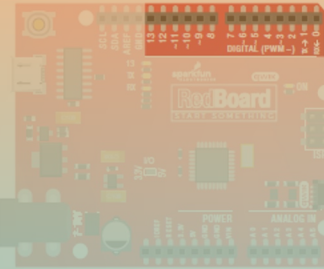
Components like resistors need to have their legs bent into 90° angles in order to correctly fit in the breadboard sockets.



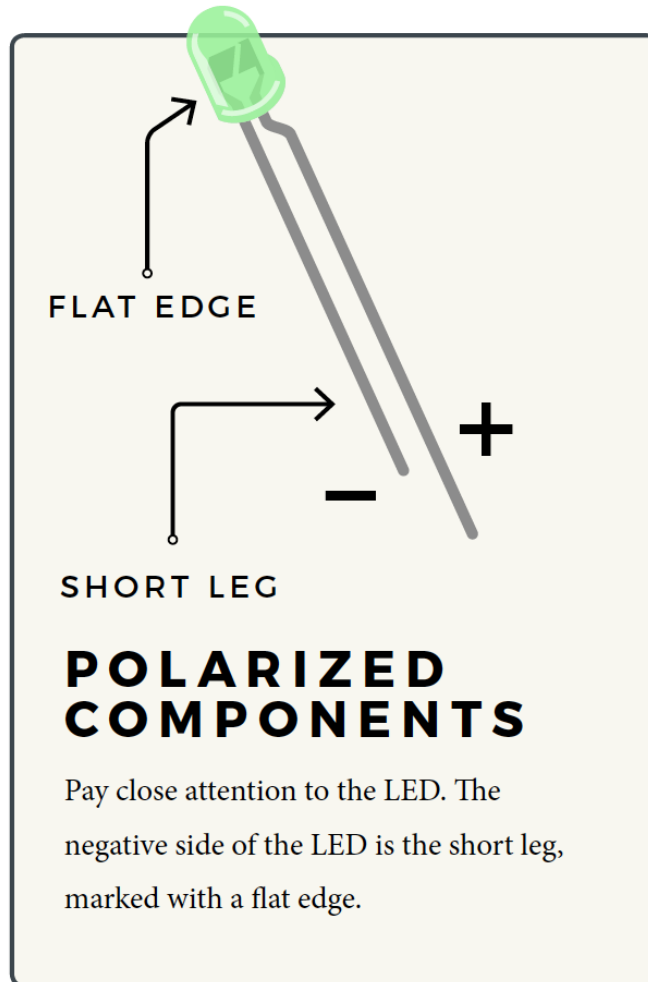
THE BOOK

INTRODUCTORY SECTION

- LED
- Polarized

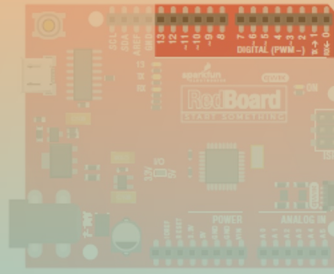


DIGITAL OUTPUT: When working with microcontrollers such as the RedBoard, there are a variety of pins to which you can connect electronic components. Knowing which pins perform which functions is important when building your circuit. In this circuit, we will be using what is known as a digital output. There are 14 of these pins found on the RedBoard. A digital output only has two states: ON or OFF. These two states can also be thought of



FIRST CIRCUIT

- Resistor
 - not polarized



DIGITAL OUTPUT: When working with microcontrollers such as the RedBoard, there are a variety of pins to which you can connect electronic components. Knowing which pins perform which functions is important when building your circuit. In this circuit, we will be using what is known as a digital output. There are 14 of these pins found on the RedBoard. A digital output only has two states: ON or OFF. These two states can also be thought of

as HIGH or LOW. These pins can only perform one function: other pins can perform multiple functions.

NEW

ELECTRICITY can be dangerous. A power source can injure you.

COMPONENTS have polarity. The new polarity is marked with a plus and minus sign.

FLAT E

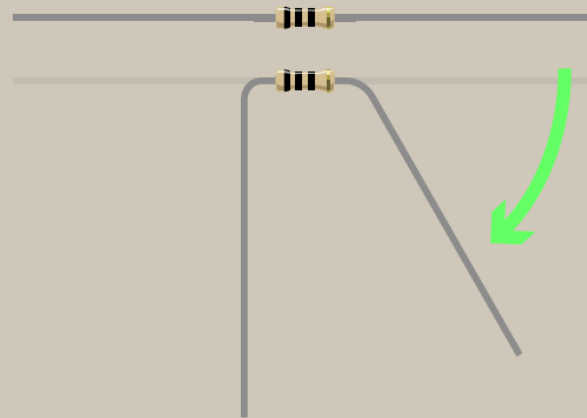
SHOR

POLARIZED COMPONENTS

Pay close attention to the negative and positive polarity marked with a plus and minus sign.

RESISTOR LEADS

Components like resistors need to have their legs bent into 90° angles in order to correctly fit in the breadboard sockets.



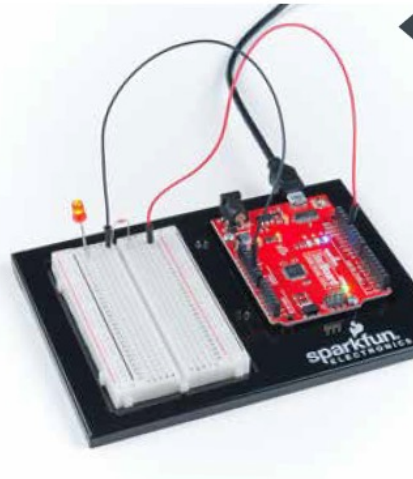
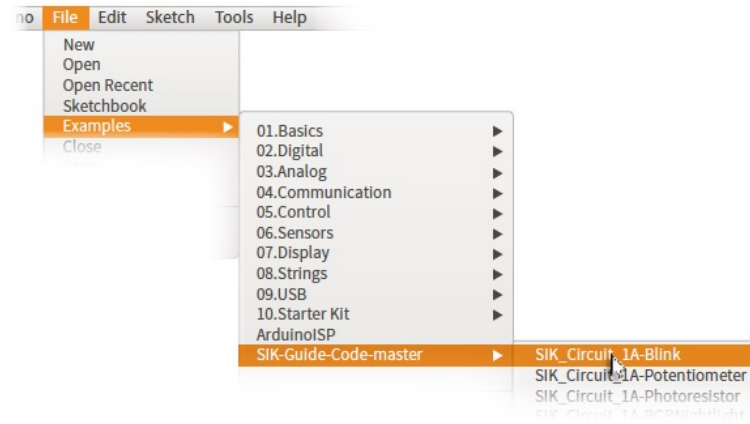
FIRST CIRCUIT

Open the Arduino IDE

Connect the RedBoard to a USB port on your computer.

➦ Open the Sketch: File > Examples > SIK-Guide-Code-master > **CIRCUIT_1A-BLINK**

➦ Select **UPLOAD** to program the sketch on the RedBoard.



WHAT YOU SHOULD SEE

The LED will flash on for two seconds, then off for two seconds. If it doesn't, make sure you have assembled the circuit correctly and verified and uploaded the code to your board. See the Troubleshooting section at the end of this circuit if that doesn't work. One of the best ways to understand the code you uploaded is to change something and see how it affects the behavior of your circuit. What happens when you change the number in one or both of the `delay(2000);` lines of code (try 100 or 5000)?

FIRST CIRCUIT

Open the Arduino IDE

Connect the RedBoard to a USB port on your computer.

➤ Open the Sketch: File > Examples > SIK-Guide-Code-master > **CIRCUIT_1A-BLINK**

➔ Select **UPLOAD** to program the sketch on the RedBoard.

no File Edit Sketch Tools Help

New
Open
Open Recent
Sketchbook

WHAT YOU SHOULD SEE

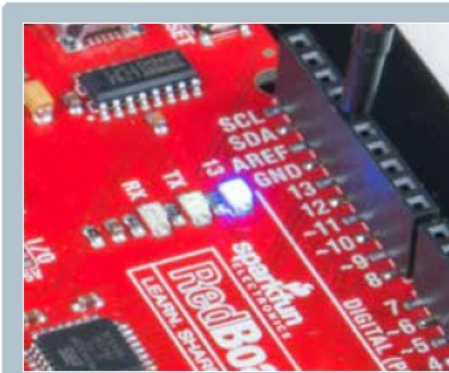
The LED will flash on for two seconds, then off for two seconds. If it doesn't, make sure you have assembled the circuit correctly and verified and uploaded the code to your board. See the Troubleshooting section at the end of this circuit if that doesn't work. One of the best ways to understand the code you uploaded is to change something and see how it affects the behavior of your circuit.

What happens when you change the number in one or both of the `delay(2000);` lines of code (try 100 or 5000)?

FIRST CIRCUIT

PROGRAM OVERVIEW

- 1 Turn the LED on by sending power (5V) to digital pin 13.
- 2 Wait 2 seconds (2000 milliseconds).
- 3 Turn the LED off by cutting power (0V) to digital pin 13.
- 4 Wait 2 seconds (2000 milliseconds).
- 5 Repeat.



ONBOARD LED PIN 13:

You may have noticed a second, smaller LED blinking in unison with the LED in your breadboard circuit. This is known as the onboard LED, and you can find one on almost any Arduino or Arduino-compatible board. In most cases, this LED is connected to **digital pin 13 (D13)**, the same pin used in this circuit.

NEW IDEAS

CODE TO NOTE: The sketches that accompany each circuit introduce new programming techniques and concepts as you progress through the guide. The Code to Note section highlights specific lines of code from the sketch and explains them in greater detail.

CODE TO NOTE

SETUP AND LOOP:

```
void setup() {} &  
void loop() {}
```

Every Arduino program needs these two functions. Code that goes in between the curly brackets {} of `setup()` runs once. The code in between the `loop()` curly brackets {} runs over and over until the RedBoard is reset or powered off.

INPUT OR OUTPUT?:

```
pinMode(13, OUTPUT);
```

Before you can use one of the digital pins, you need to tell the RedBoard whether it is an **INPUT** or **OUTPUT**. We use a built-in “function” called `pinMode()` to make pin 13 a digital output. You’ll learn more about digital inputs in Project 2.

FIRST CIRCUIT

CODE TO NOTE

DIGITAL OUTPUT:

```
digitalWrite(D13, HIGH);
```

When you're using a pin as an **OUTPUT**, you can command it to be **HIGH** (output 5 volts) or **LOW** (output 0 volts).

DELAY:

```
delay(2000);
```

Causes the program to wait on this line of code for the amount of time in between the brackets, represented in milliseconds (2000ms = 2s). After the time has passed, the program will continue to the next line of code.

COMMENTS:

```
//This is a comment  
  
/* So is this */
```

Comments are a great way to leave notes in your code explaining why you wrote it the way you did. Single line comments use two forward slashes //, while multi-line comments start with a /* and end with a */.

NEW IDEAS

CODING CHALLENGES: The Coding Challenges section is where you will find suggestions for changes to the circuit or code that will make the circuit more challenging. If you feel overwhelmed by the tasks in each circuit, visit the Coding Challenges section to push yourself to the next level.

CODING CHALLENGES

PERSISTENCE OF VISION: Computer screens, movies and the lights in your house all flicker so quickly that they appear to be on all of the time but are actually blinking faster than the human eye can detect. See how much you can decrease the delay time in your program before the light appears to be on all the time but is still blinking.

MORSE CODE: Try adding and changing the `delay()` values and adding more `digitalWrite()` commands to make your program blink a message in Morse code.

TROUBLESHOOTING

I get an error when uploading my code

The most likely cause is that you have the wrong board selected in the Arduino IDE. Make sure you have selected **Tools > Board > Arduino/Genuino Uno**.

FIRST CIRCUIT

CODE TO NOTE

DIGITAL OUTPUT: When you're using a pin as an OUTPUT, you can command it to be HIGH (output 5 volts) or LOW (output 0 volts).
`digitalWrite(D13, HIGH);`

DELAY: Causes the program to wait on this line of code for the amount of time in between the brackets, represented in milliseconds (2000ms = 2s). After the time has passed, the program will continue to the next line of code.
`delay(2000);`

COMMENTS: Comments are a great way to leave notes in your code explaining why you wrote it the way you did. Single line comments use two forward slashes //, while multi-line comments start with a /* and end with a */.
`//This is a comment`
`/* So is this */`

CODING CHALLENGES

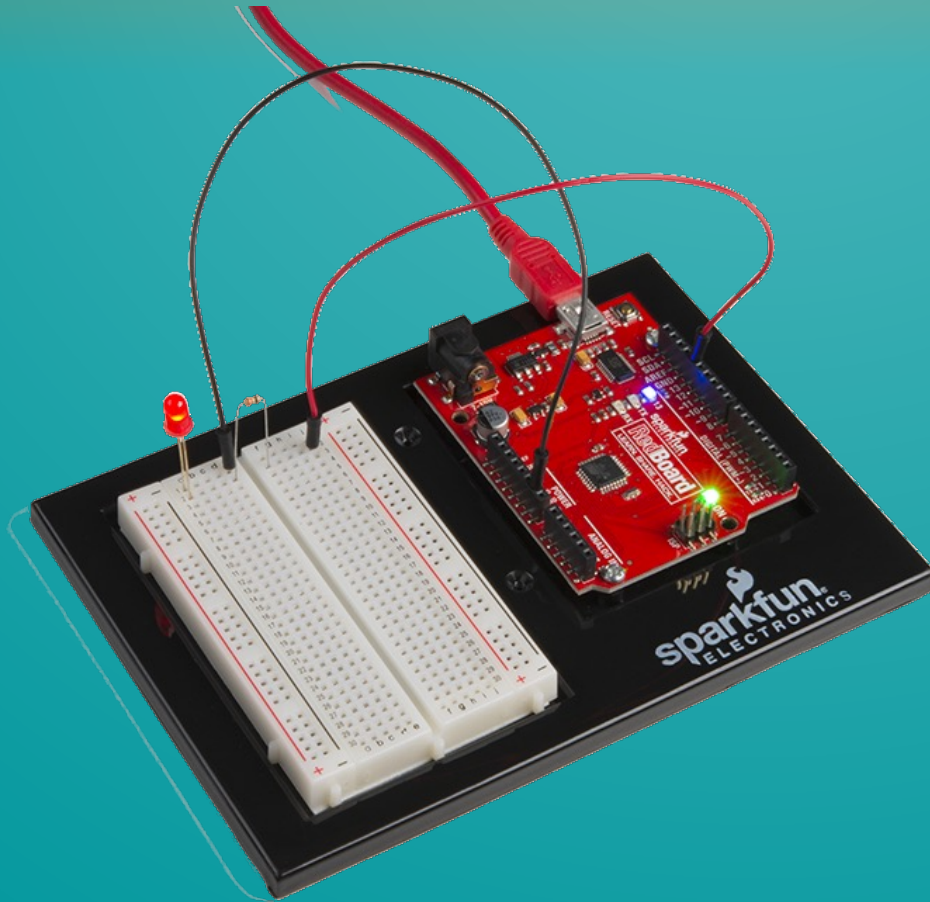
PERSISTENCE OF VISION: Computer screens, movies and the lights in your house all flicker so quickly that they appear to be on all of the time but are actually blinking faster than the human eye can detect. See how much you can decrease the delay time in your program before the light appears to be on all the time but is still blinking.

MORSE CODE: Try adding and changing the `delay()` values and adding more `digitalWrite()` commands to make your program blink a message in Morse code.

I get an error when uploading my code The most likely cause is that you have the wrong board selected in the Arduino IDE. Make sure you have selected **Tools > Board > Arduino/Genuino Uno**.

FIRST CIRCUIT

- Circuit 1A: Blinking an LED -pg 13

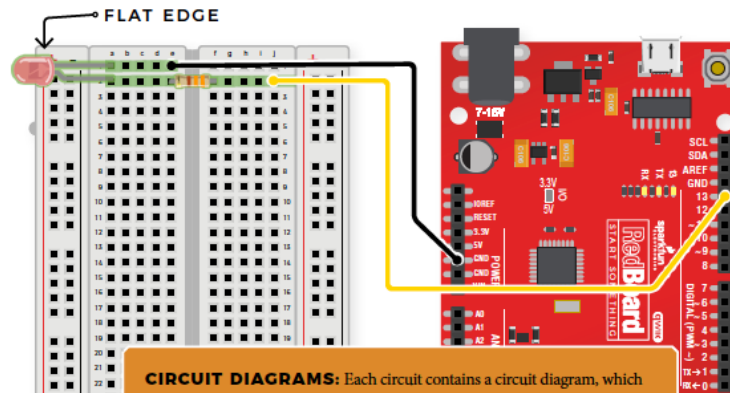


FIRST CIRCUIT

- Circuit 1A:
Blinking an LED
-pg 13

HOOKUP GUIDE

READY TO START HOOKING EVERYTHING UP? Check out the circuit diagram and hookup table below to see how everything is connected.



CIRCUIT DIAGRAMS: Each circuit contains a circuit diagram, which acts as a visual aid designed to make it easier for you to see how your circuit should be built. Each colored line represents a jumper wire connection in the circuit. All wires should have two connection points, which you also see in the hookup table below.

COLORS: Please note that while traditionally red is used for power and black is used for ground, all wires, no matter their color, function the same.

HOOKUP TABLES: Many electronics beginners find it helpful to have a coordinate system when building their circuits. For each circuit, you'll find a hookup table that lists the coordinates of each component or wire and where it connects to the RedBoard, the breadboard, or both. The breadboard has a letter/number coordinate system, just like the game Battleship.

◆ **D13 to J2**

...means one end of a component connects to digital pin 13 on your RedBoard and the other connects to J2 on the breadboard

CONNECTION TYPES ◆ REDBOARD CONNECTION ■ BREADBOARD CONNECTION

JUMPER WIRES ◆ **D13 to J2** ◆ **GND to E1**

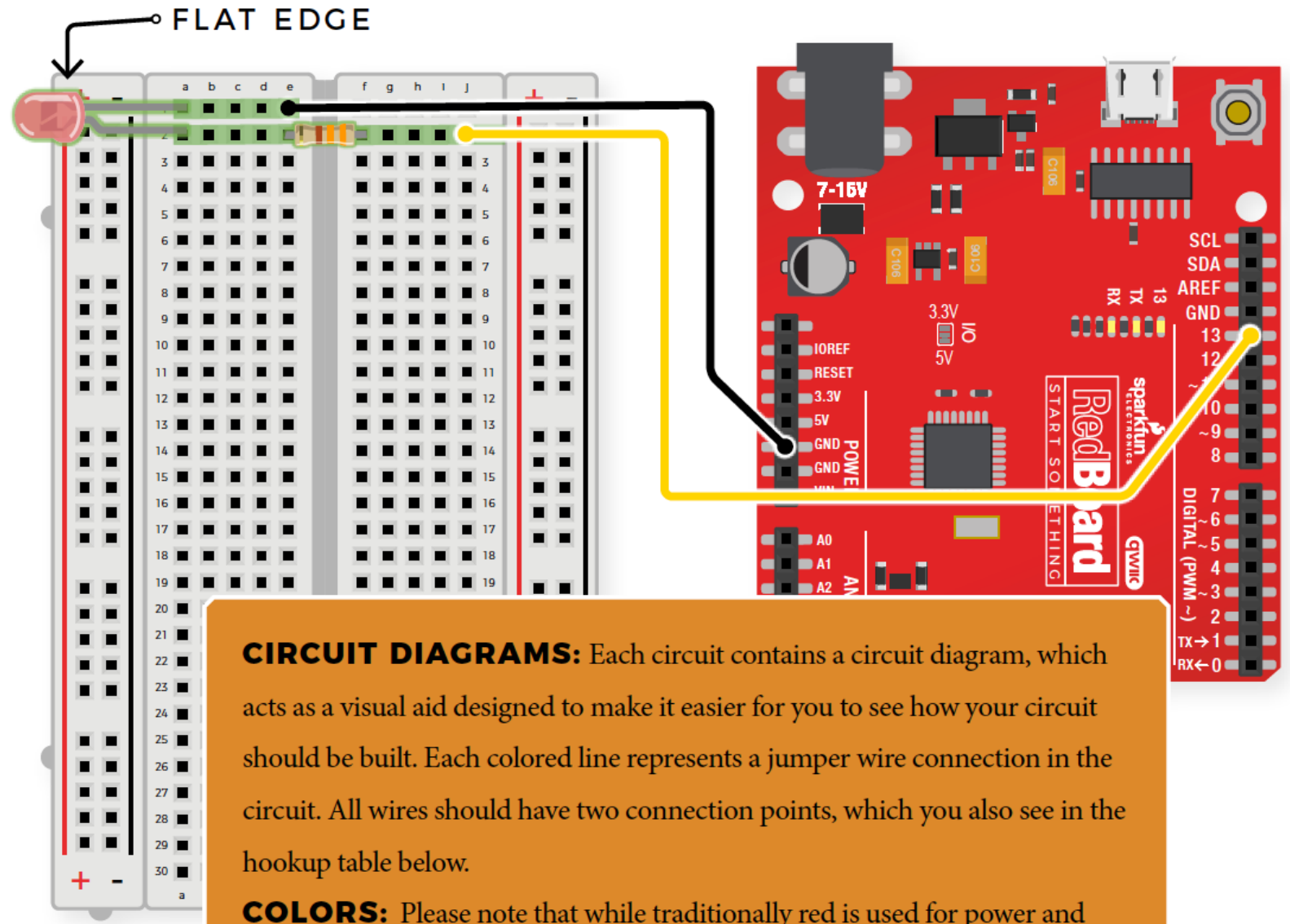
LED ■ **A1(-) to A2(+)**

330Ω RESISTOR (ORANGE, ORANGE, BROWN) ■ **E2 to F2**

In this table, a yellow highlight indicates that a component has polarity and will only function if properly oriented.

FIRST CIRCUIT

- Circuit 1A:
Blinking an LED
-pg 13



CIRCUIT DIAGRAMS: Each circuit contains a circuit diagram, which acts as a visual aid designed to make it easier for you to see how your circuit should be built. Each colored line represents a jumper wire connection in the circuit. All wires should have two connection points, which you also see in the hookup table below.

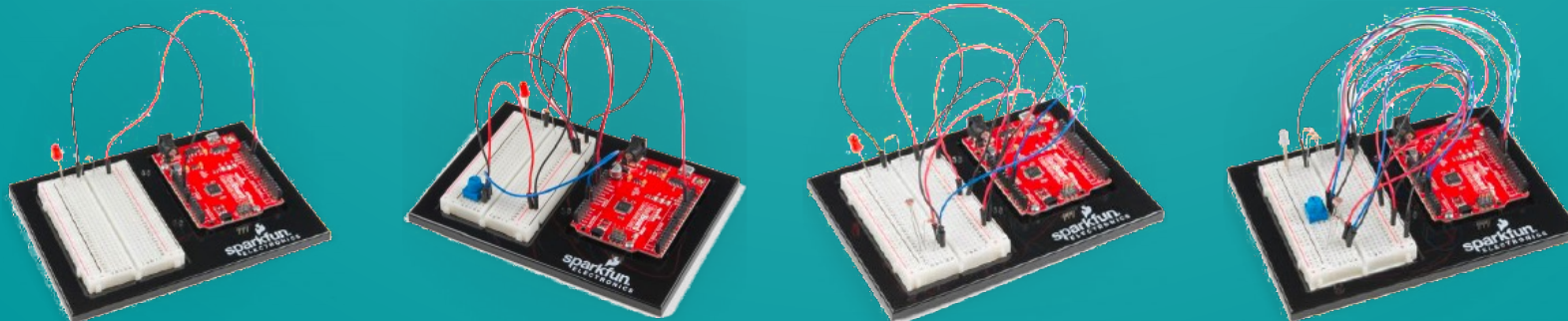
COLORS: Please note that while traditionally red is used for power and black is used for ground, all wires, no matter their color, function the same.

EXCELLENT!



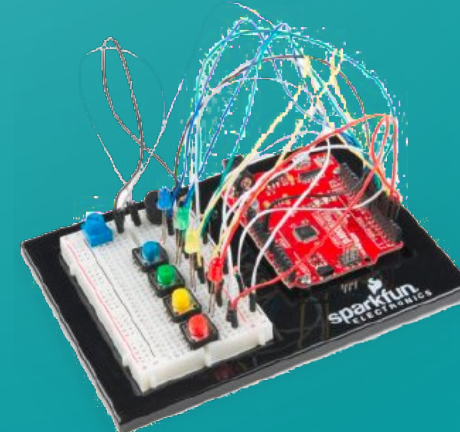
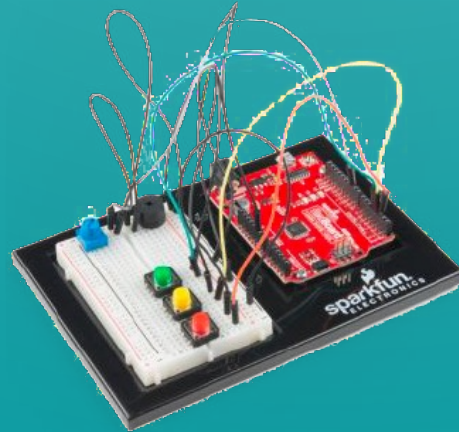
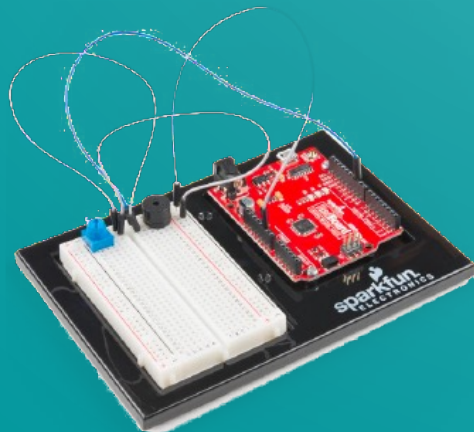
PROJECT 1: LIGHT

- Circuit 1A: Blinking an LED -pg 13
- Circuit 1B: Potentiometer -pg 20
- Circuit 1C: Photoresistor -pg 26
- Circuit 1D: RGB Night-Light -pg 31



PROJECT 2: SOUND

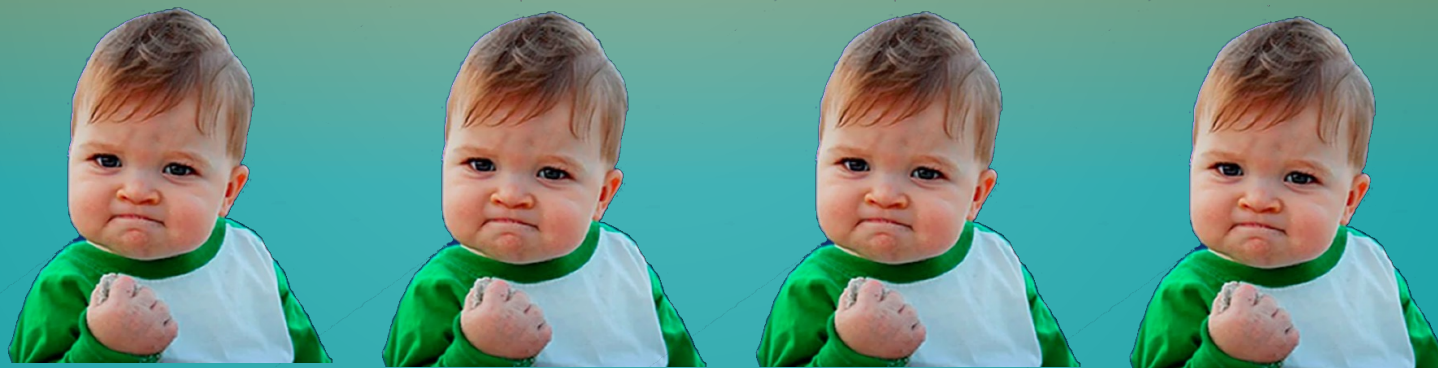
- Circuit 2A: Buzzer -pg 37
- Circuit 2B: Digital Trumpet -pg 42
- Circuit 2C: “Simon Says” Game -pg 47



The background features a vertical gradient from orange at the top to teal at the bottom. In each of the four corners, there are white, stylized circuit board traces that branch out and terminate in small circles, resembling nodes or components on a PCB.

WHAT DID WE LEARN?

EXCELLENT!

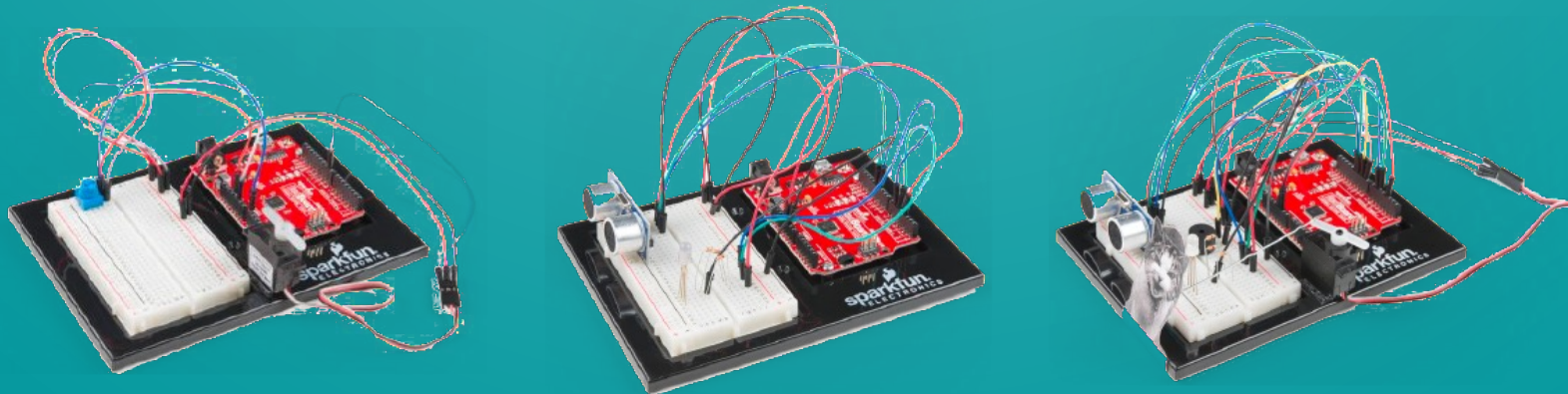


The image features a background with a vertical gradient from orange at the top to teal at the bottom. In the four corners, there are decorative white line-art patterns resembling circuit traces or fiber optic paths, with small circles at the end of the lines. The text "SEE YOU TOMORROW!" is centered in the upper half of the image.

SEE YOU TOMORROW!

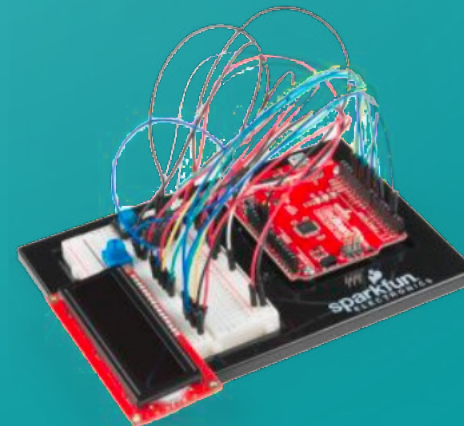
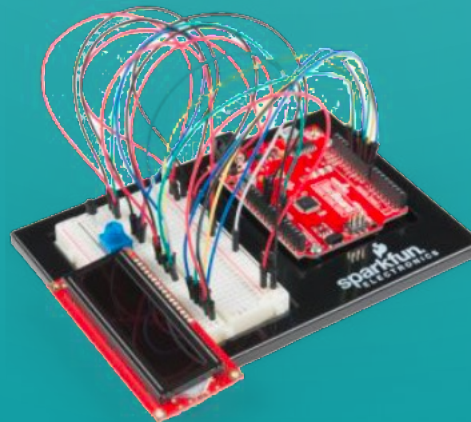
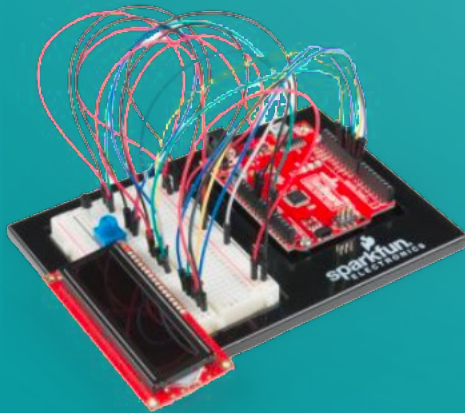
PROJECT 3: MOTION

- Circuit 3A: Servo Motors -pg 54
- Circuit 3B: Distance Sensor -pg 60
- Circuit 3C: Motion Alarm -pg 65



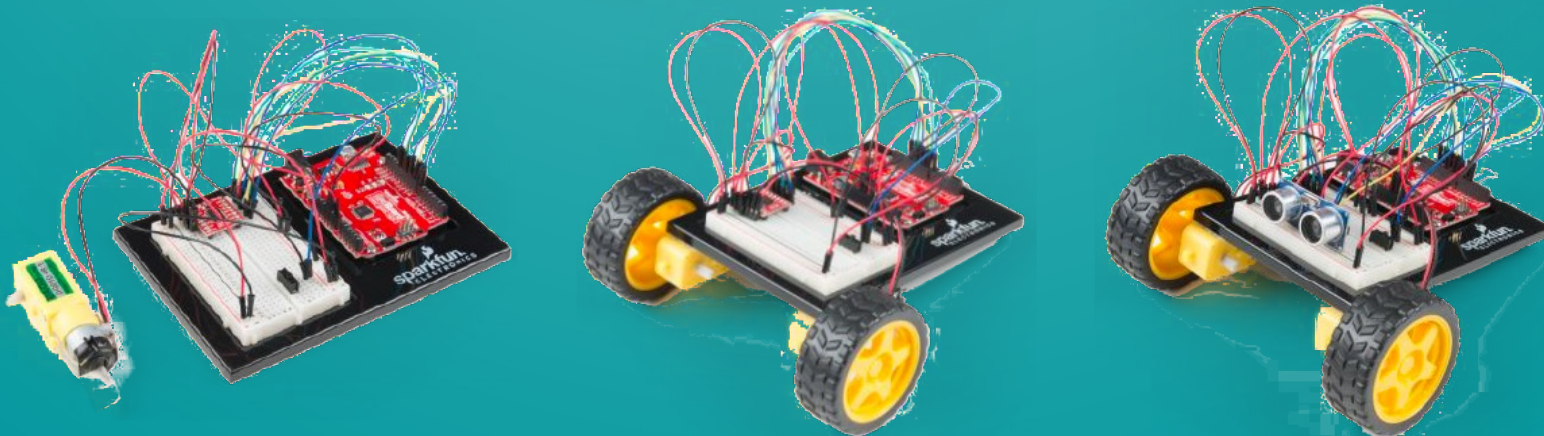
PROJECT 4: DISPLAY

- Circuit 4A: LCD “Hello World!” -pg 72
- Circuit 4B: Temperature Sensor -pg 77
- Circuit 4C: DIY “Who Am I?” Game -pg 82



PROJECT 5: ROBOT

- Circuit 5A: Motor Basics -pg 89
- Circuit 5B: Remote-Controlled Robot -pg 96
- Circuit 5C: Autonomous Robot -pg 102



The background features a vertical gradient from orange at the top to teal at the bottom. White circuit-like lines with circular nodes are positioned in the corners: top-left, top-right, bottom-left, and bottom-right.

HOPE YOU HAD FUN!
&
LEARNED SOMETHING

EXCELLENT!

