Beginner Arduino Course

Session 1: What is Arduino?

Grades 5+

CONTEXT

A client requested two courses to introduce children to foundational skills in engineering and programming (such as syntax and logic, plus circuitry basics). Each course consisted of 10 lessons, 2-hours each. Each course would be both an independent introduction for total beginners and also novel for students who wished to take both courses. The following sample lesson plan and slide deck is for Session 1, providing a first project using a widely available Arduino kit.

Session 1: What is Arduino?

Summary

1. We will learn what Arduino is and safely create our first programs.
2. Learning Outcomes:
    a. Learn the basics of electrical safety.
    b. Learn what Arduino is.
    c. Create our first programs.
3. Time Allotment: 2 hours

Materials & Resources

Notes:

a. Materials & Quantities:

i. Elegoo Uno Super Starter Kit with Uno R3 (https://www.amazon.com/ELEGOO-Project-Tutorial-Controller-Projects/dp/B01D8KOZF4/) (ONE PER STUDENT + 1 for teacher)

ii. Laptop (with working USB port and either the Arduino Create Plugin installed or administrator privileges for the student to install the Arduino Create Plugin) (ONE PER STUDENT + 1 for teacher)

iii. 10 plastic snack bags per student (Use these for the resistors).

iv. Projector

v. A damaged resistor as a prop and/or any other visibly damaged components you may have from your own Arduino projects. If it's your first time teaching this class, there are lots of resistors in your kit. Take any one of them and bend it into ridiculous angles. It's still perfectly usable but can demonstrate the importance of care to students.

vi. Slide deck (Session 1 - Google Drive)

b. Resources/Links:

i. In-class example: Arduino Banana Keys: https://vimeo.com/146193319

ii. Additional examples if YouTube is accessible:

1. LED Cube https://youtu.be/T5Aq7cRc-mU (first 17 seconds)

2. Coffee Cup Spy Cam https://youtu.be/bWyFnC-EWnk (first 36 seconds)

3. Mario the Maker Magician https://www.youtube.com/watch?v=6cFprwKjCdU (26 seconds)

iii. If students ask questions about the exact nature of the programming language we'll use, this is a good resource: https://arduino.stackexchange.com/questions/816/c-vs-the-arduino-language Additionally, this line from Wikipedia (https://en.wikipedia.org/wiki/Arduino): "The microcontrollers can be programmed using the C and C++ programming languages, using a standard API which is also known as the 'Arduino language'."

c. General troubleshooting tips:

1. Check resistors. A student may grab a 10k resistor instead of a 200-ohm resistor.
2. Remind students frequently of how breadboard rails work. Check wiring.
3. Check polarized components, such as LEDs.
4. It is possible that one wire or LED could be faulty, especially if reusing kits. Try swapping components one by one. When removing a component, carefully put it away from all other materials so it can be discarded if nonfunctional.
5. When checking the code, ensure the pinMode has been established correctly.
6. Check that the pin wired is the pin referenced in the code.
7. Have students help each other troubleshoot. Being able to find mistakes quickly is an important skill for an engineer.

Session Procedure

1. Greetings (5-10 minutes)
    a. Students need to have clean hands. Excessively dirty hands could potentially damage the components. Ask students to wash their hands if necessary. If students are using their own laptops, ask if the Arduino Create plugin has been installed as required. If not, get the family member with computer administrator privileges to help now.
    b. Do your standard introduction game to get to know the students.
2. 1-sentence Arduino Introduction (5 minutes)
    a. Present the Uno and briefly introduce it. Ask if anyone has used one before and if so, what they did with it. Ask if any students have taken Arduino 2: Spy Camp with [Company Name]. Note how many students are experienced/inexperienced, as that will guide the rest of the day.
    b. Show the following examples.
        i. Example: Arduino Banana Keys: https://vimeo.com/146193319
    c. Additional examples if YouTube is accessible:
        i. LED Cube https://youtu.be/T5Aq7cRc-mU (first 17 seconds)
        ii. Coffee Cup Spy Cam https://youtu.be/bWyFnC-EWnk (first 36 seconds) (Same as in Arduino 2: Spy Camp)
        iii. Mario the Maker Magician https://www.youtube.com/watch?v=6cFprwKjCdU (26 seconds)

d. Encourage discussion about the example. What do they think? How does it work?

3. What is Arduino? (A little more depth) [5-10 minutes]
    a. Present the slide deck ("What is Arduino?"). Finish at the "Questions" slide.
    b. Review the key ideas with the students after the presentation.
4. What's in our kit? (10-15 minutes)
    a. Discuss the different components in the kit.
        i. Show your broken/damaged resistor and encourage gentle handling of delicate parts.
        ii. Note the long pin on LEDs is the positive pin (anode) and the short pin is negative (cathode).
        iii. Note the connections on the breadboard. (Show the Breadboard slide in the slide deck.) If you have an extra breadboard, you can peel away the sticker on the back to reveal the connections. Some students only understand once they can see this face-to-face.
        iv. Note that we will use neither the battery nor the battery to DC connector, as mistakes here could be dangerous.
    b. Materials: Pass out the plastic snack bags to organize resistors. There should be 10 different types of resistors. 1 type per bag.
    c. Quiz! Tell students to cover their kit parts lists (The CD and thank you card are perfect for this). Show a few parts and ask students to name them. Then ask them to show you various parts. Ask which pin is positive on the LED.

5. Electricity and circuits [5-10 minutes]

    a. Preview: What do students know about electricity, circuits, and electrical safety?

    b. Present the slide deck beginning with the title slide “Electricity and Circuits,” play the video about electricity, and discuss the questions that follow.

6. Safety Discussion (5 minutes)

    a. Teacher's note: To be 100% clear, our projects are completely safe for kids. However, if kids make the wrong kind of mistakes in class, they can damage or break components. If they go home and watch YouTube videos of projects that plug into the wall or use batteries and then make a mistake, they could potentially start a fire or cause a battery to explode.

    b. Discuss electrical safety.

        i. Key points:

            1. Dangers of electricity

            2. If students follow instructions, everything in class is safe.

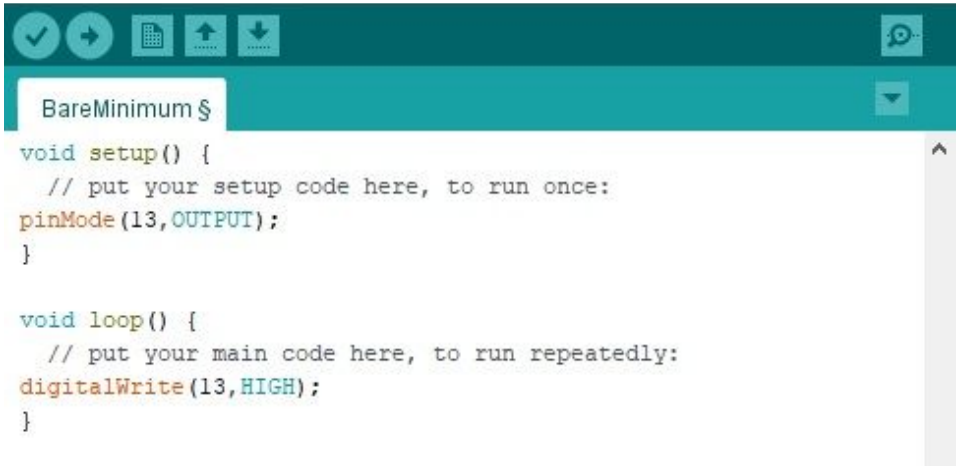
            3. Do not try projects with the battery or try to plug your kit into the wall without asking an experienced adult first. There is a risk of fire, electrocution, death, or even breaking your board.

            4. Rules to learn good electrical safety:

                1. Always disconnect your project from its power source before working on it.

                2. Be organized. Give each component lots of space. Don't allow components to touch accidentally.

7. Break [10-15 minutes, beginning around the 45-55 minute mark]
8. Arduino Create account setup [5-15 minutes]
    a. Direct students to create.arduino.cc to create or log into accounts. Students should use Google to register.
        i. Note: If they create an account without Google or Apple, it will ask for their age and their parent's email. Their parent will need to approve the request before the student can login. If you have any students without a Google account, they can create a new account and ask their parent to approve it when they get home. For today's session, here is a shared login:
        ii. Username: [Username]
        iii. Password: [Password]
    b. Direct students to the editor.
    c. If students took Arduino 2: Spy Camp with [Company Name] or if they say they are experienced with Arduino, encourage them to use the rest of class as a free build to demonstrate what they know and remember about Arduino. Consider calling on these students to help less experienced students as needed.
9. Our first program [20-30 minutes]
    a. Project: Turn on the LED by pin 13.
    b. Materials: Laptop with Arduino Create plugin installed, Uno, USB cable



c. Key points:

i. Check that all computers have the plugin installed and recognize the boards.

ii. Discuss the difference between void setup and void loop. Arduino likes to be organized. Here is where we organize our code.

iii. Discuss how pins can be input/output and where we designate that in our code.

iv. Some students may be new to programming. Students must type what you write exactly, character by character.

v. Capitalization is key. When commands are properly capitalized, the text color changes in the IDE. Encourage students to purposefully use incorrect capitalization to see what the IDE does.

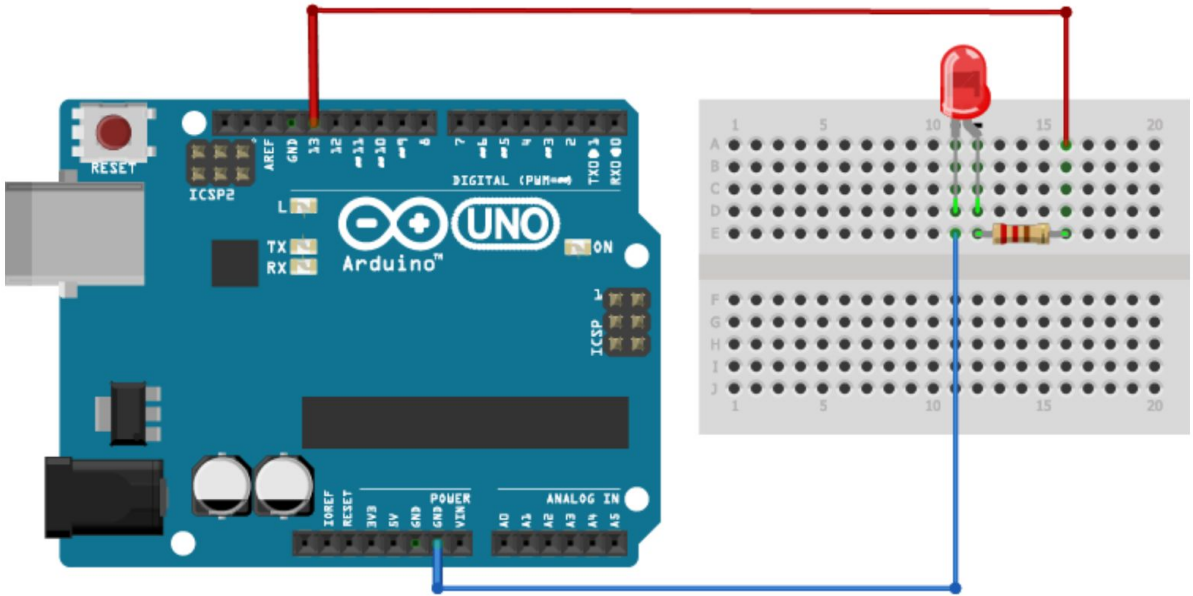
vi. End each line with a semicolon.

vii. Name your file something descriptive, such as Session1FirstCircuit or Session1Activity2, and save your file.

10. First Circuit [20-30 minutes]

a. Project: Connect an LED to the Arduino using the breadboard.

b. Materials: Laptop, USB cable, Uno, breadboard, 220 Ohm resistor, two wires, LED.



c. Key Points:

i. Create a tangible circuit with multiple components.

ii. Review how the breadboard works.

iii. Long leg of the LED is positive. Electricity flows into it.

iv. Resistors protect our components from too much electricity. Always use a resistor with an LED. Once you let the smoke out, you can't put it back in. (i.e. It is easy to permanently burn out components if you're not careful.)

v. Disconnect the breadboard from the wire providing electricity from the Arduino before modifying your circuit.

vi. Name your file something descriptive, such as Session1FirstCircuit or Session1Activity2, and save your file.

d. Encourage students who finish quickly to experiment with different pins. What if you use pin 12? Or experiment with resistors **larger than 220**. (Stay organized with those bags!) How does the light differ from 220 Ohm to 1000 Ohm? What about 220 Ohm compared to 1,000,000 Ohm?

12. Extension [5 min +]

c. If time remains, introduce how to make the LED blink:

```
digitalWrite(13,HIGH);
delay(200); //Note: This is in milliseconds.
digitalWrite(13,LOW);
delay(200);
```

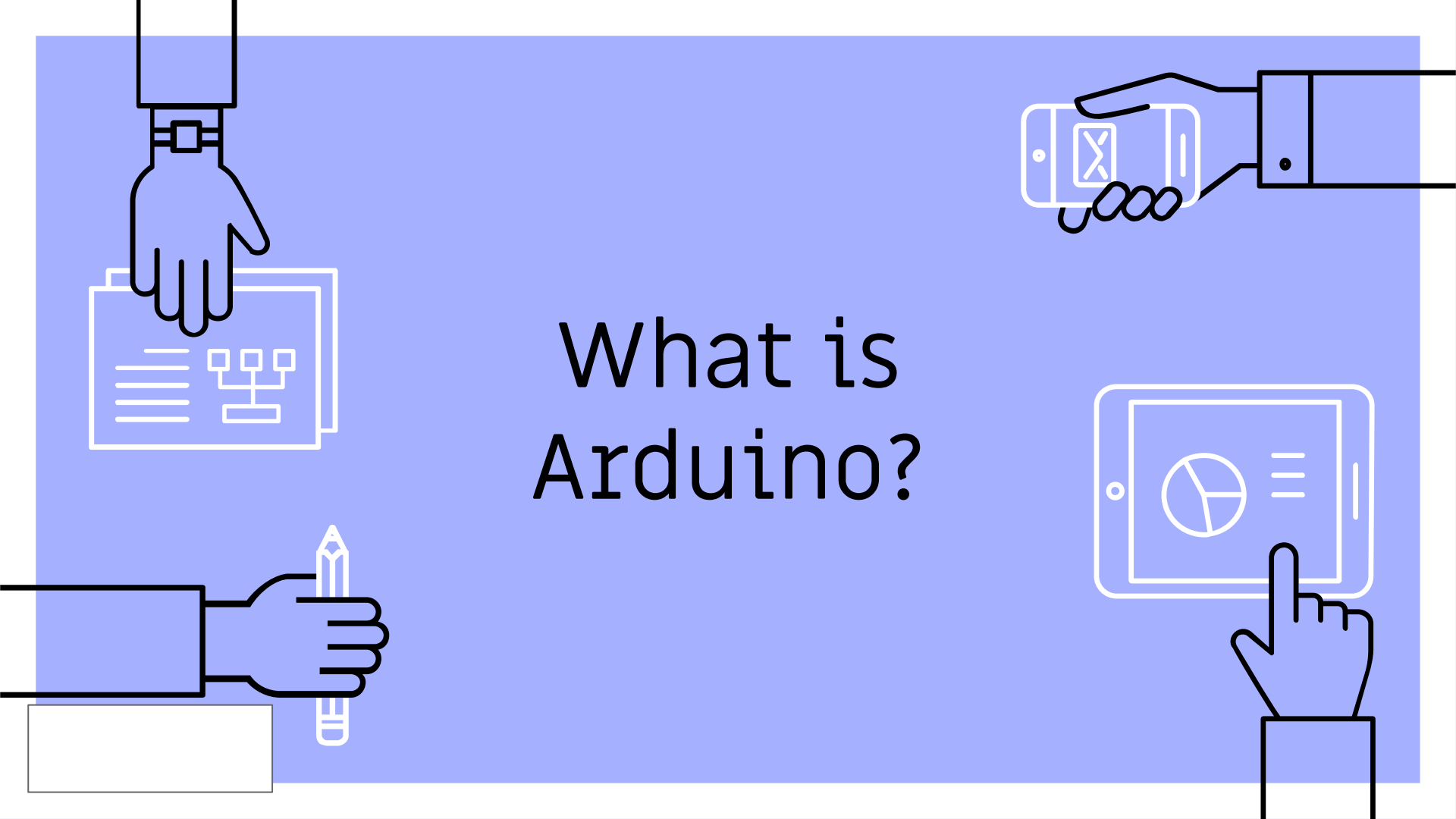
d. How fast can you make it blink and still see it's blinking? Can you make it stay on for a long time, then off for a short time, then back on? Can you change your blinking pattern, starting slow and getting faster, then repeating when it's as fast as possible?

12. Wrap-up [10-15 minutes]
    a. Group reflection prompts:
        i. What did you learn?
        ii. What are some key things to remember when writing an Arduino program?
        iii. What is Arduino?
        iv. How do we stay safe when using Arduino?
        v. If time permits, return to the code on the "Software" slide of the slide deck and ask students what the program would do.
    b. Clean-up Instructions: Carefully deconstruct the circuit (remove electricity first) and put away your components. Put away kits and laptops if school-owned. Before students go, try to do a walk through the room looking for dropped/forgotten components.

Post-session Instructions

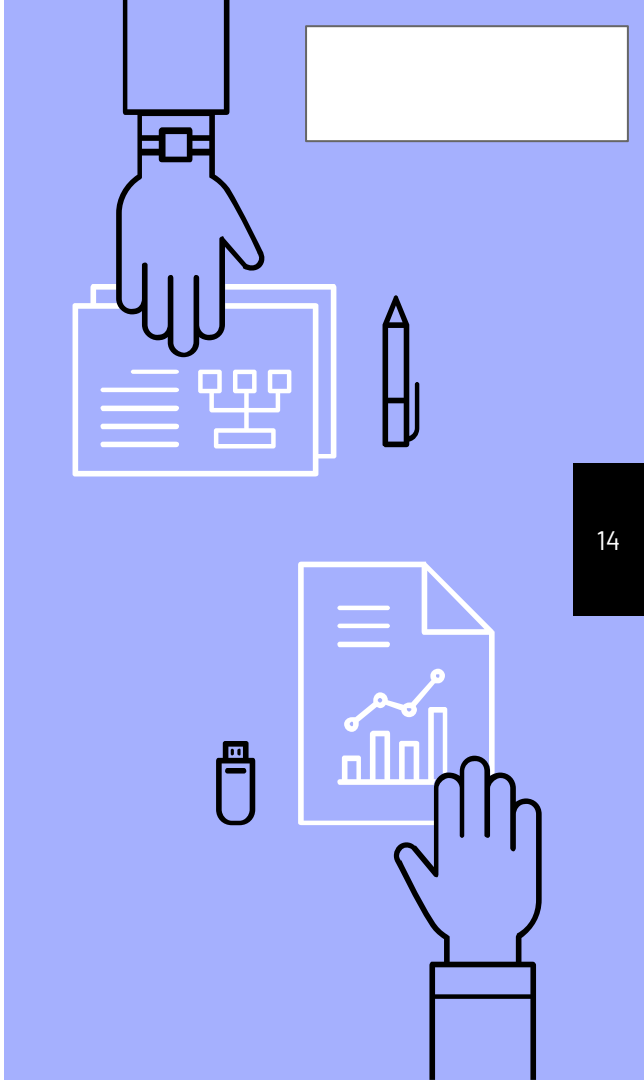
a. Check out your kids from the roster in your app.
b. Report any issues or questions to our staff.
c. Make any notes to yourself about what to review or follow up with for the next session.
d. In Session 7, we will be using the IRremote library. This library, at the time of this writing (April 2021) was in the process of being updated (to version 3.2.0). Verify before class that the code in the lesson plan runs. Troubleshoot any issues with the library ReadMe: https://github.com/Arduino-IRremote/Arduino-IRremote/blob/master/README.md
    i. Additionally, there is a button battery in the remote that may need to be replaced in older kits. If using school kits that have been used in previous classes, now is a good time to test that all remotes are working properly and/or purchase CR2025 batteries in case any batteries die.

What is Arduino?



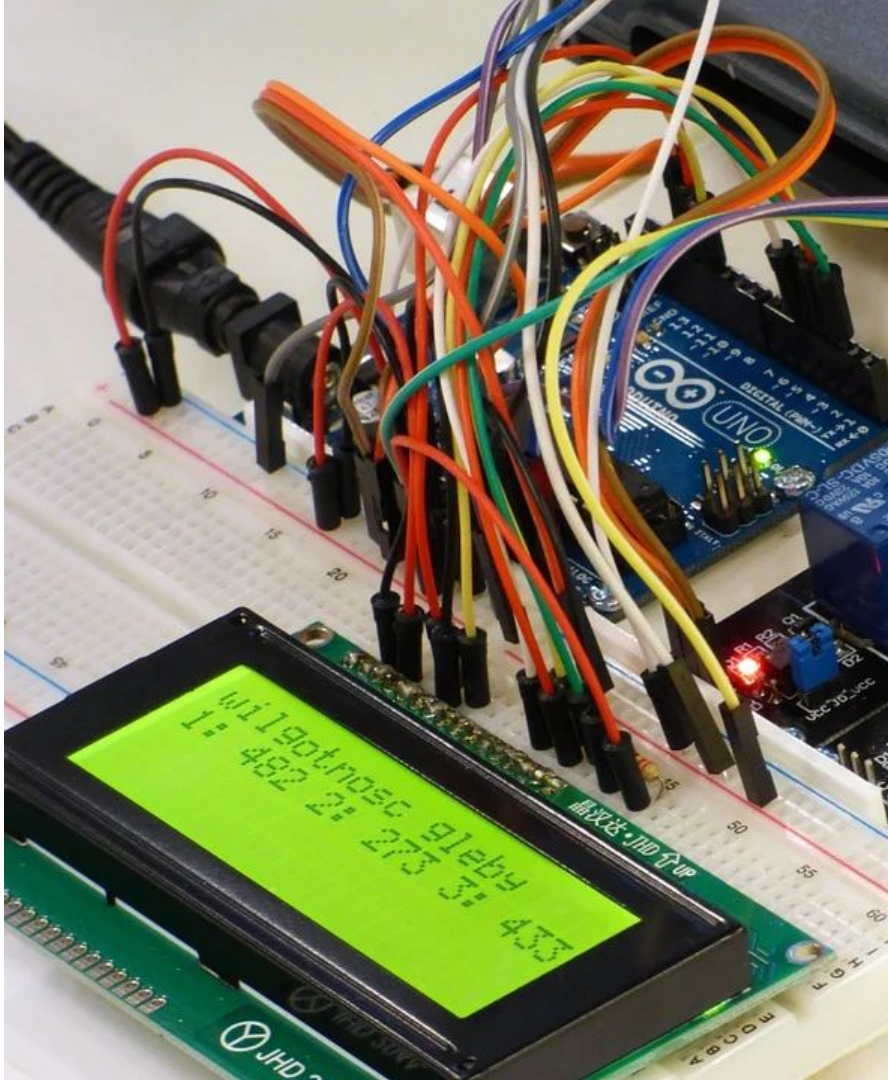
Outline

- What is Arduino?
  - Hardware
  - Software
- Who created Arduino?



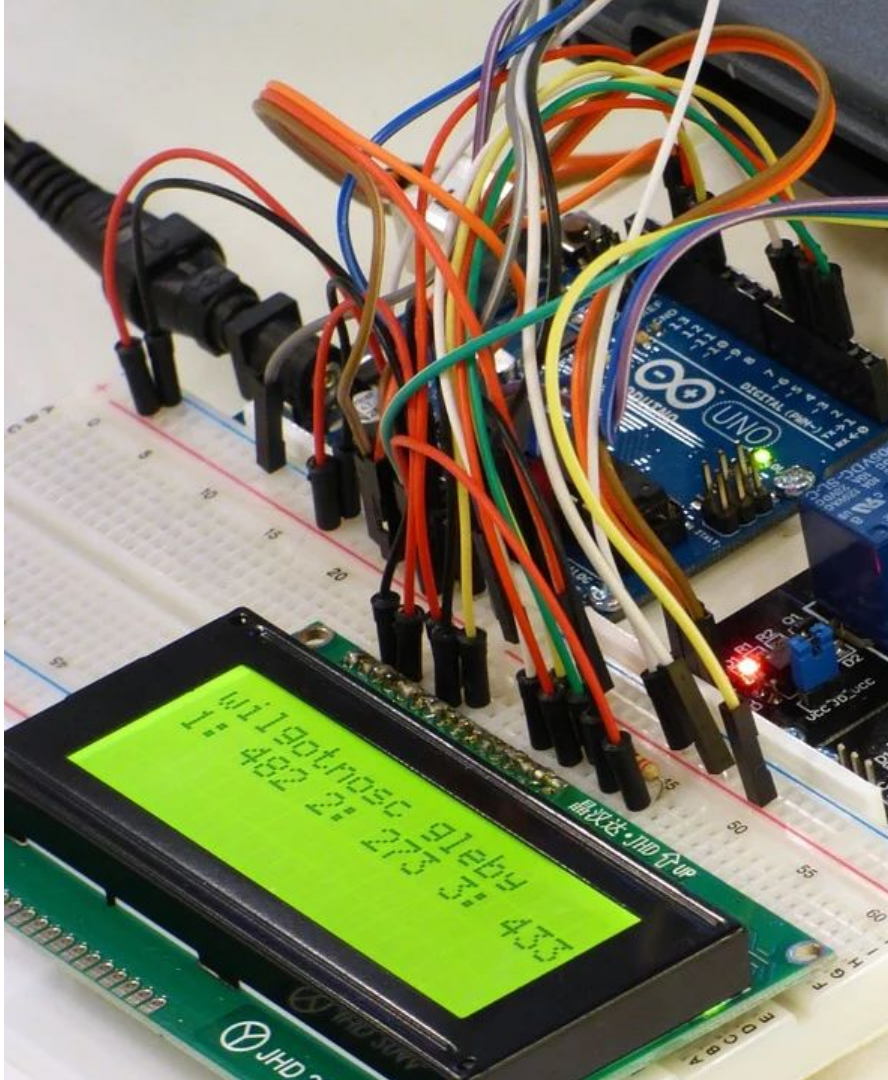
What is Arduino?

- Hardware and software system to make fun digital projects
- Take physical input and/or
- Send digital output



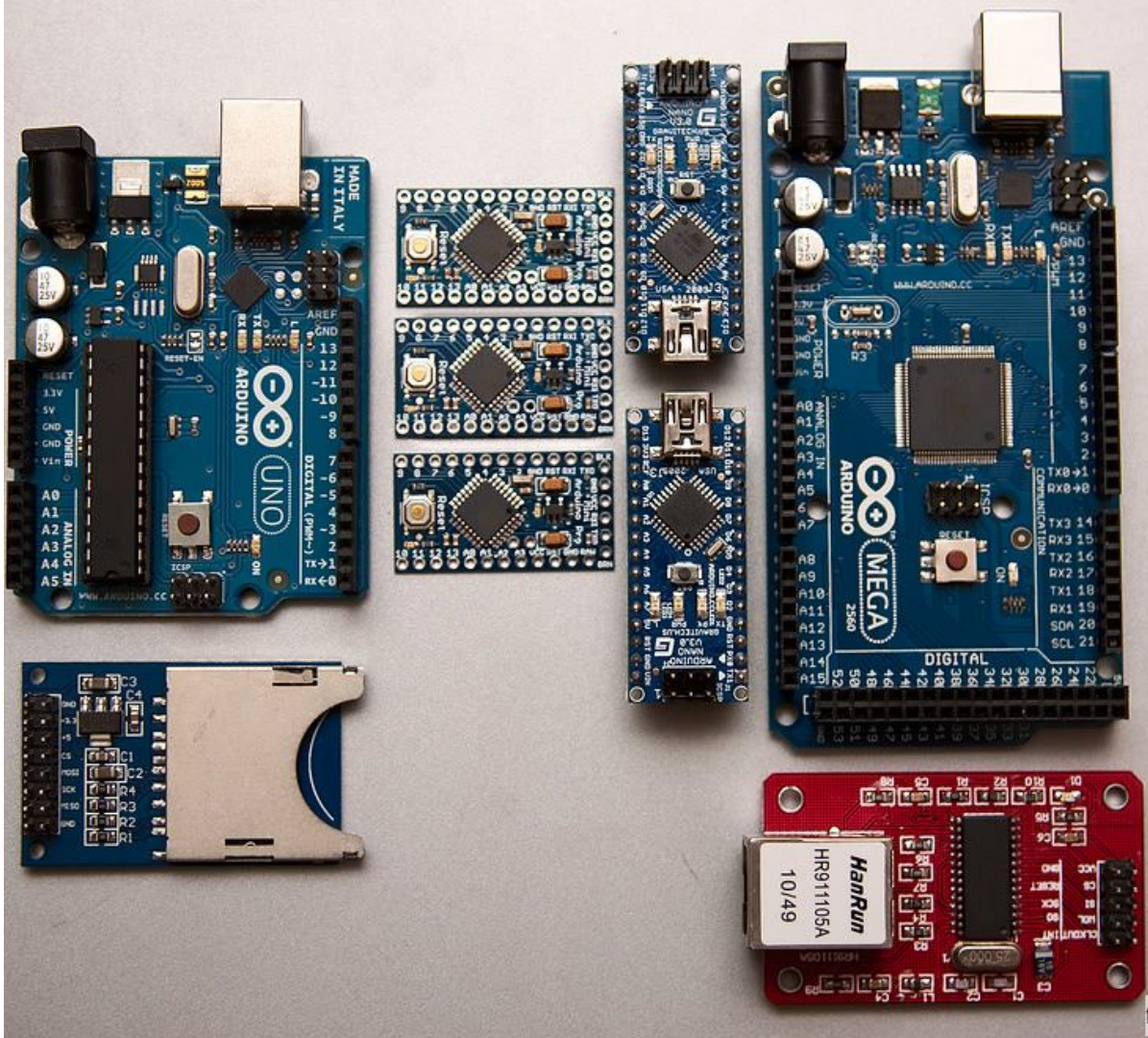
Hardware

- What's the difference between hardware and software?
- Printed circuit board (PCB)
- Microcontroller
  - manages inputs & outputs
- What components on your Uno do you recognize?



Hardware

- Different boards for different projects
  - Nano
  - Mega
  - Why "Uno?"

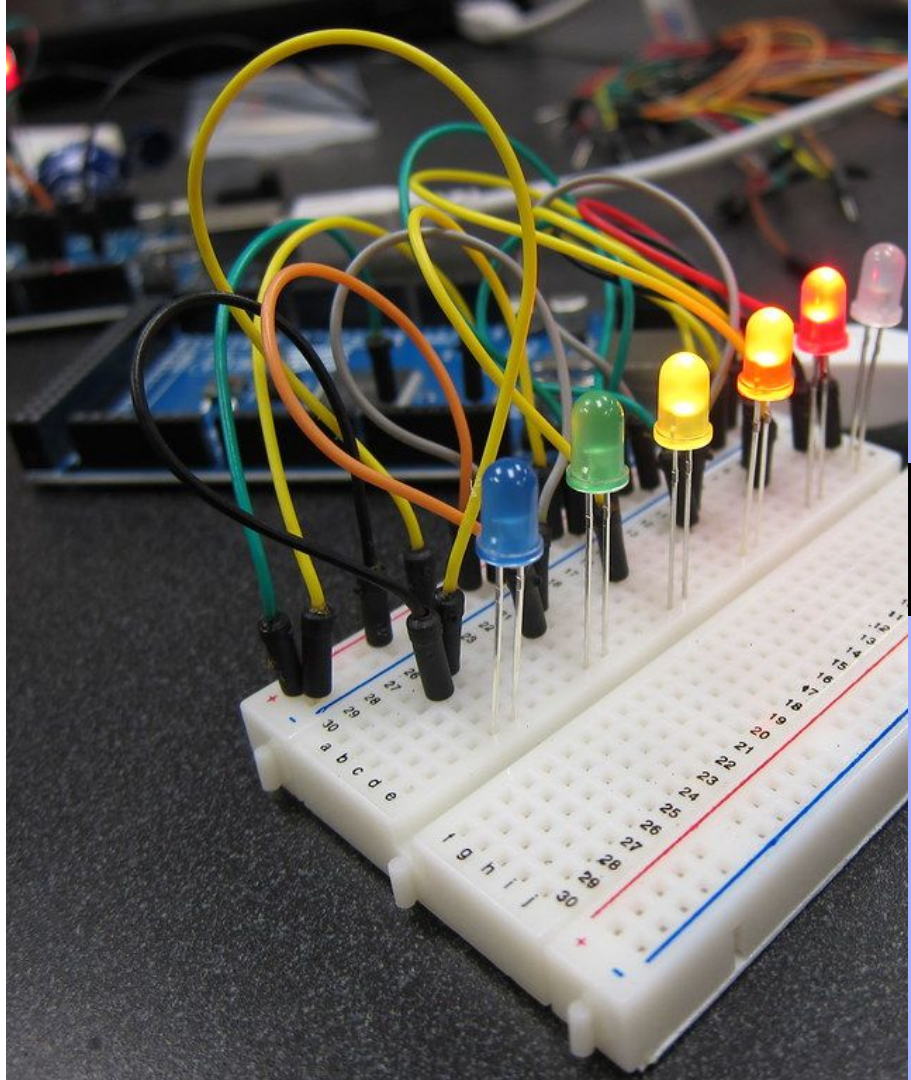


Software

- Arduino IDE
- Arduino language
  - basically C++
- You'll know what this code does by the end of the day!

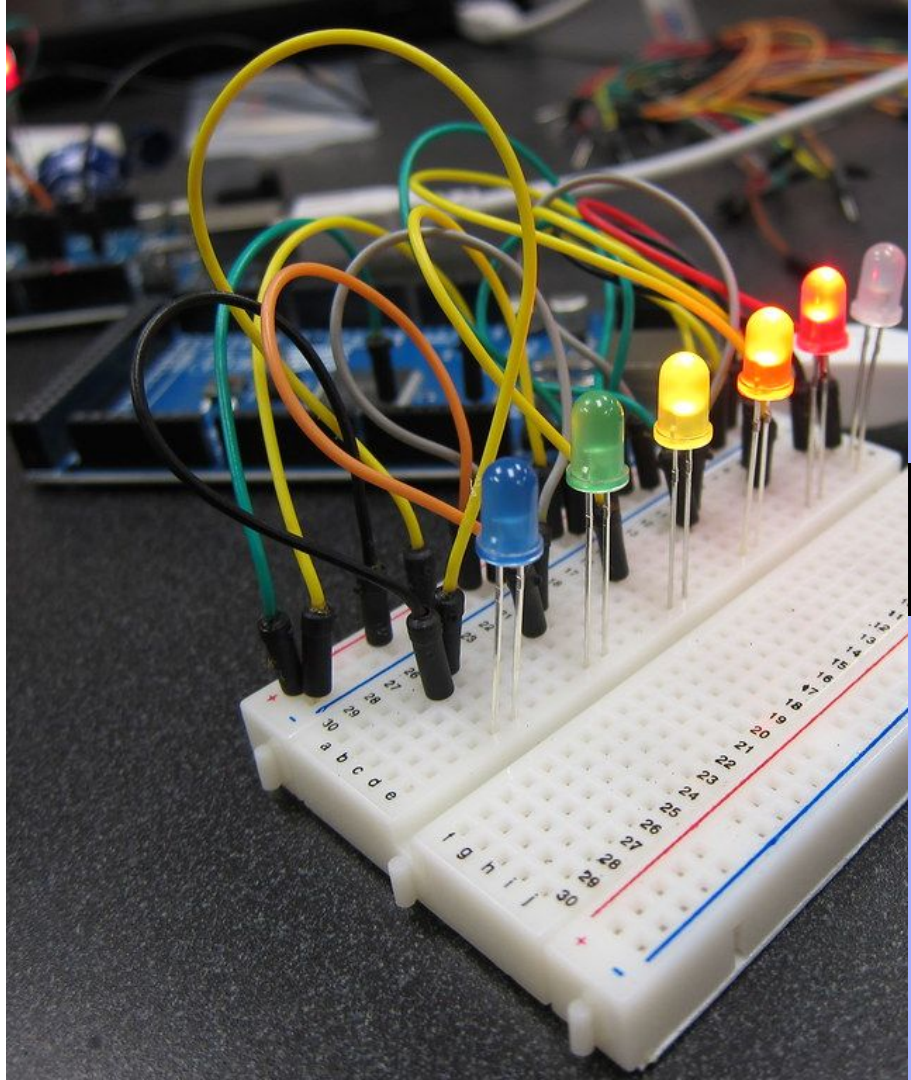
Who created Arduino?

- Italian college students in 2005
- Named after their favorite place to meet
- Low-cost, easy electronics projects



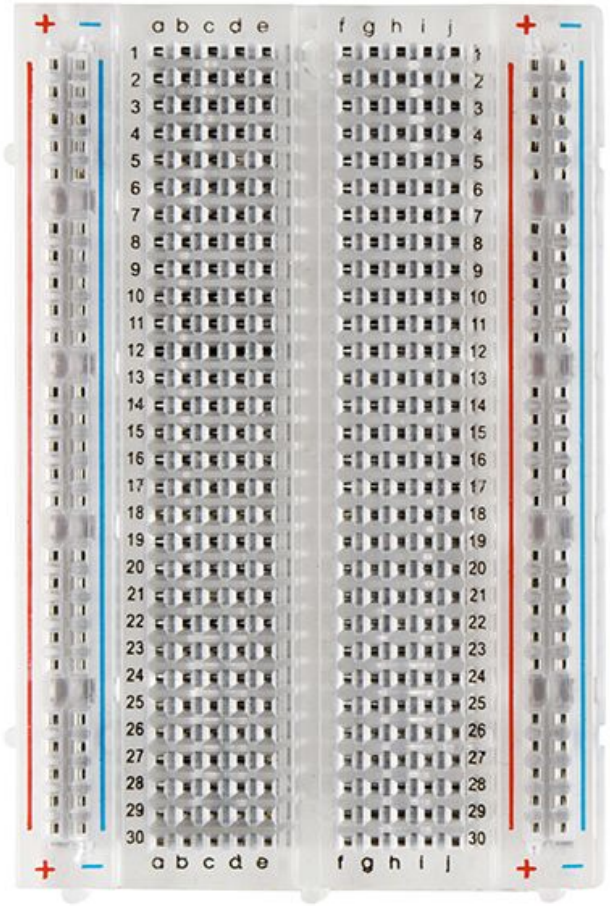
Who created Arduino?

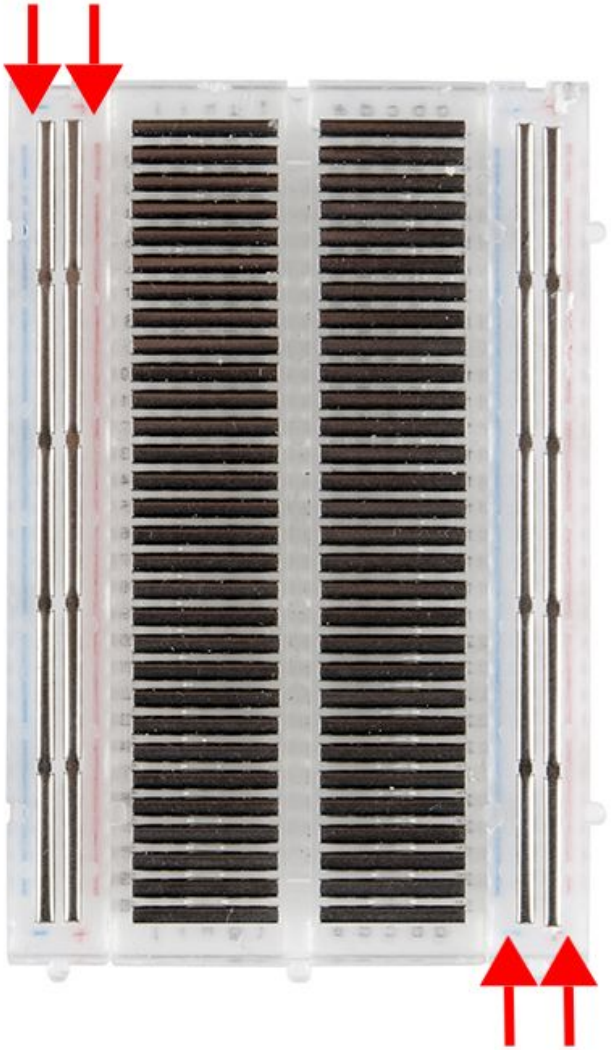
- Open-Source
- *Technically...*
  - Boards are only called "Arduino" boards if made by the creators in Italy
  - What does your board say?



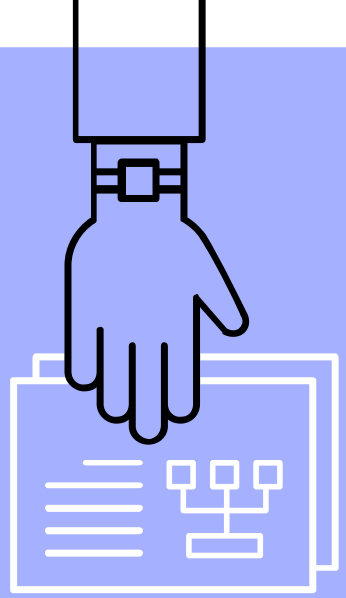
“

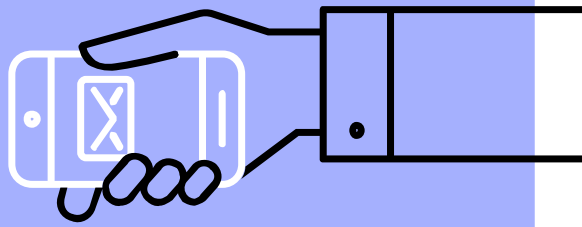
Any questions?

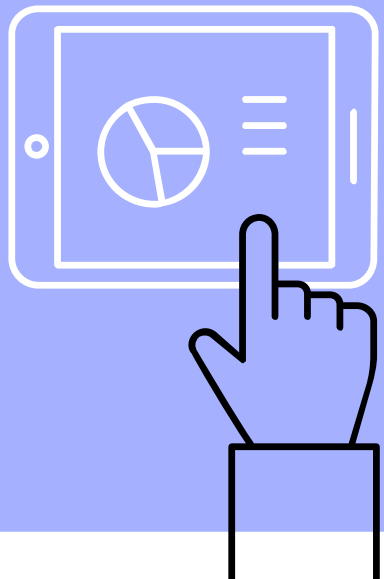

a b c d e
f g h i j
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

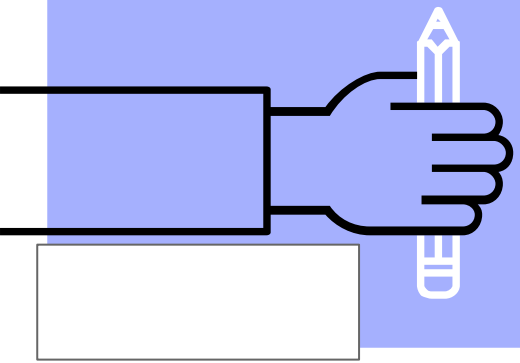


Electricity and Circuits



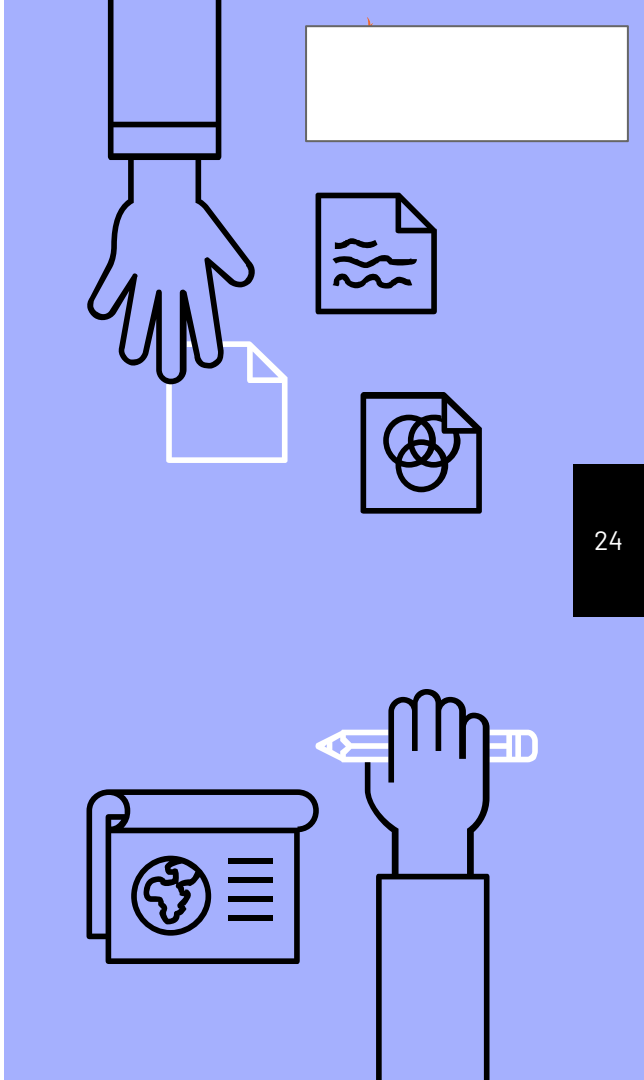






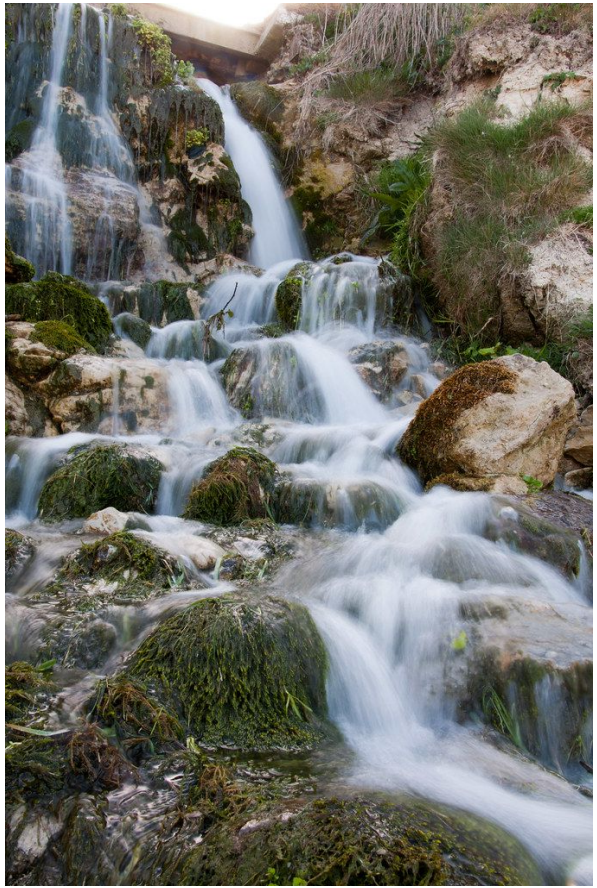
Quick intro to electricity and circuits:

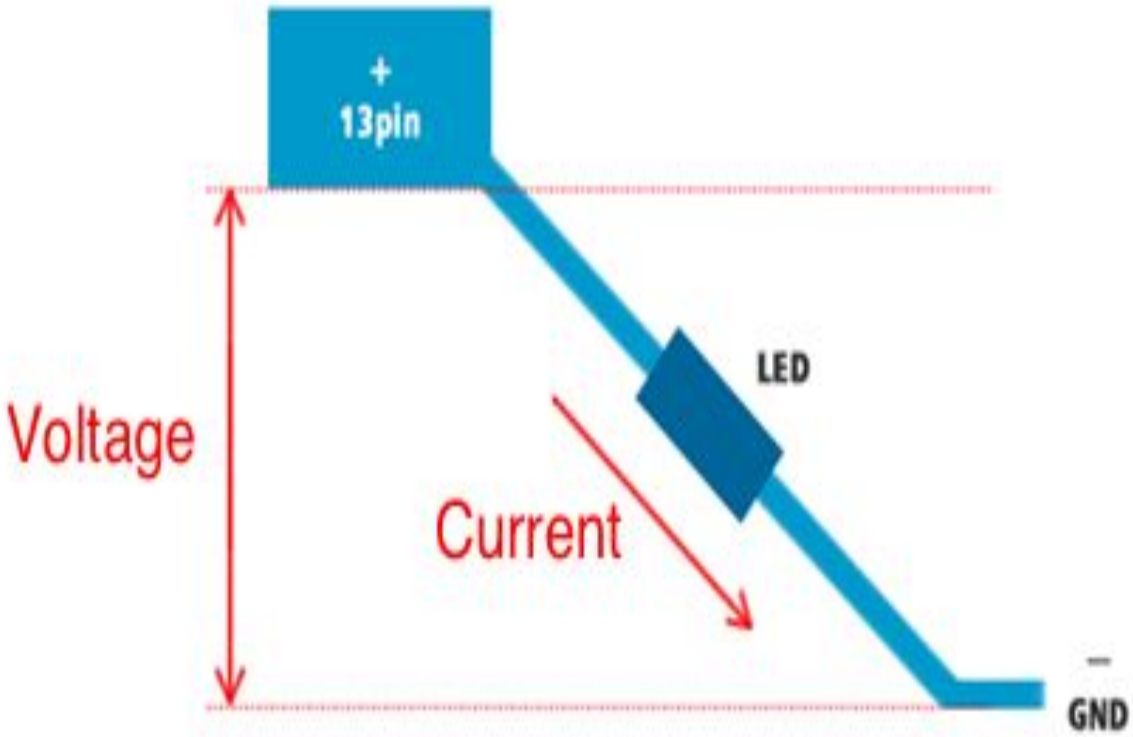
https://vimeo.com/429083521

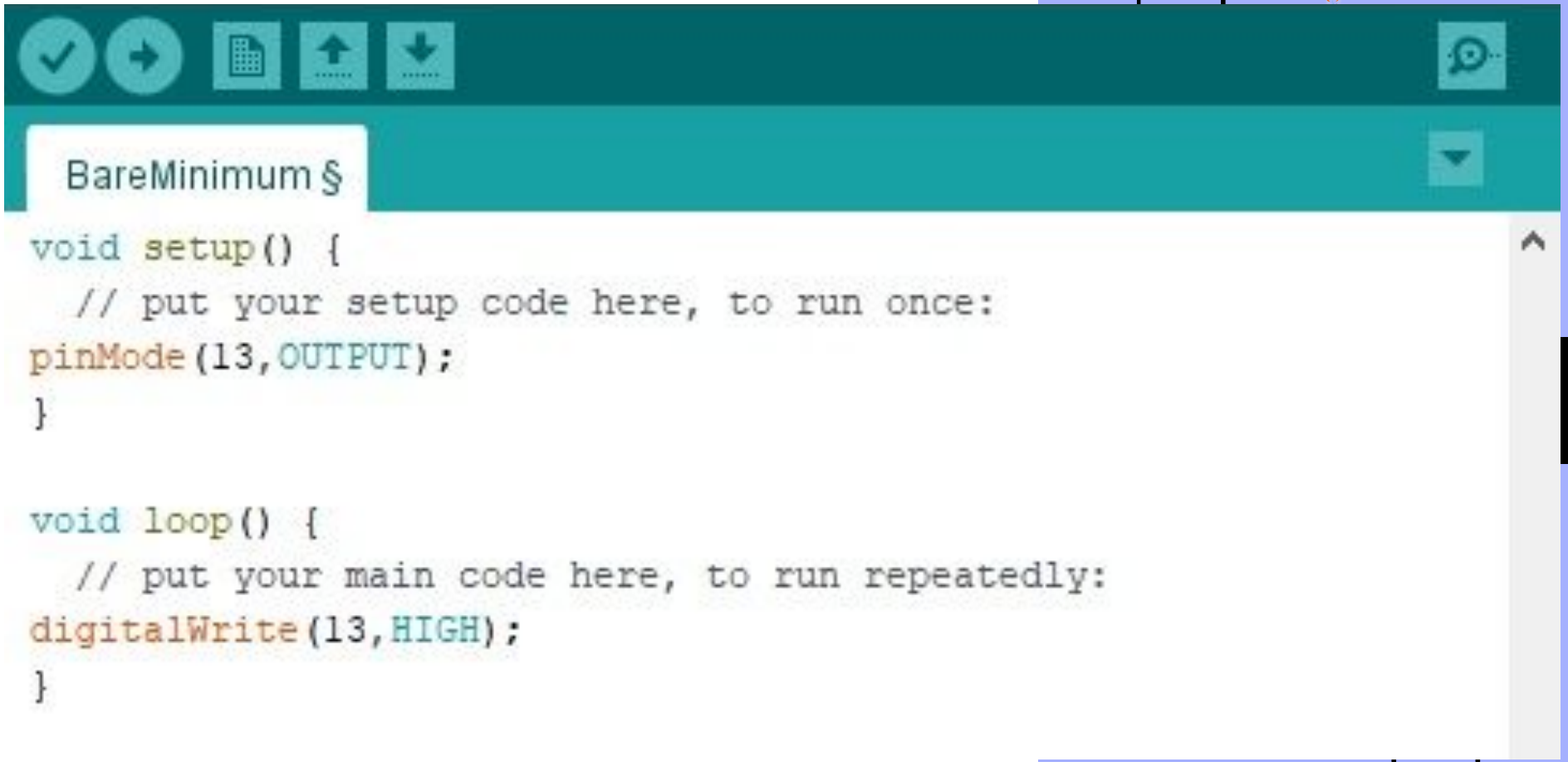


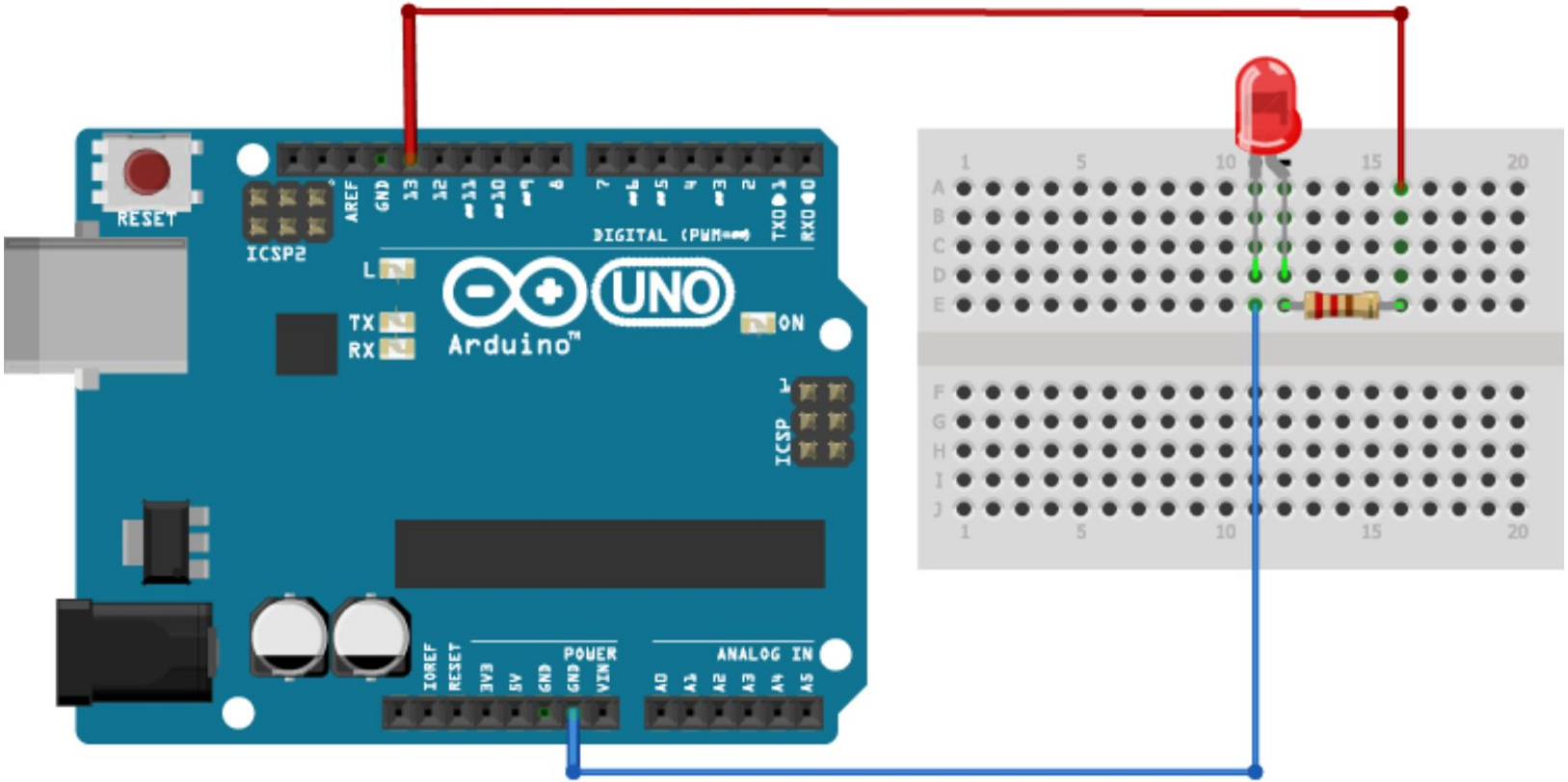
Quick Review

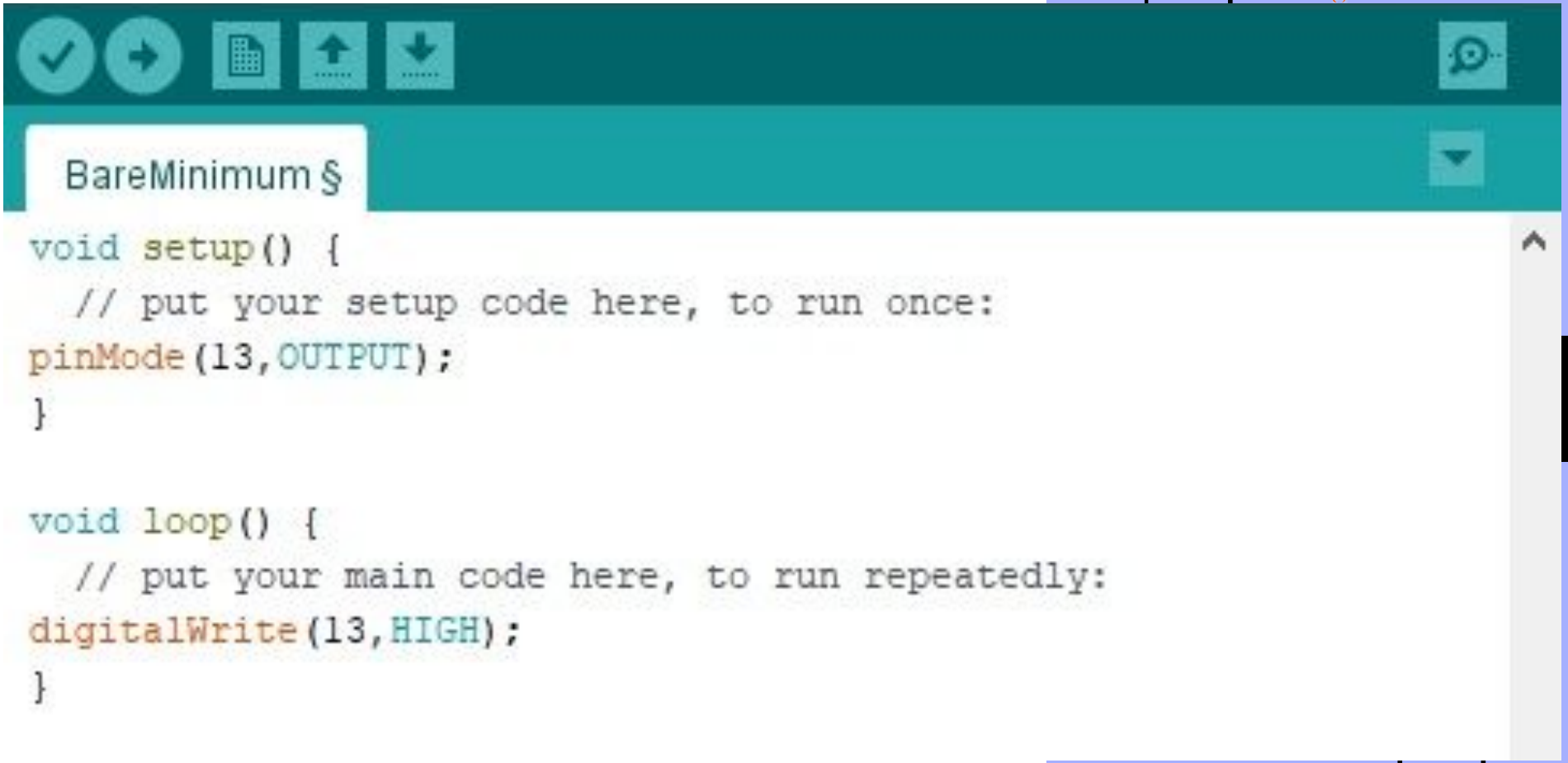
- What's a circuit?
- What's a conductor?
- What's an insulator?




+
13pin
LED
Voltage
Current
–
GND


BareMinimum §
void setup() {
// put your setup code here, to run once:
pinMode(13,OUTPUT);
}
void loop() {
// put your main code here, to run repeatedly:
digitalWrite(13,HIGH);
}


RESET
ICSP2
AREF
GND
13
12
11
10
9
8
7
6
5
4
3
2
TXD
RXD
DIGITAL (PWM~)
L
TX
RX
UNO
Arduino™
ON
ICSP
POWER
IOREF
RESET
3V3
5V
GND
GND
VIN
ANALOG IN
A0
A1
A2
A3
A4
A5
1
5
10
15
20
A
B
C
D
E
F
G
H
I
J


BareMinimum §
void setup() {
// put your setup code here, to run once:
pinMode(13,OUTPUT);
}
void loop() {
// put your main code here, to run repeatedly:
digitalWrite(13,HIGH);
}

Sources:

"Integrated Circuit Computer" by Tomasz_Mikolajczyk on Pixabay

"Arduino microcontroller / Arduino Uno microcontroller ATmega328 / Arduino Mega 2560 / Arduino Nano ATmega328 / Arduino Pro Mini ver.3.3V ATmega328 / Photo by Arkadiusz Sikorski http://www.arq.pl/" by Arkadiusz Sikorski vel ArakuS is licensed under CC BY 2.0

"Arduino-Code" by lespounder is licensed under CC BY-SA 2.0

"intro to arduino" by sobczak.paul is licensed under CC BY 2.0

"Clip ends" by oskay is licensed under CC BY 2.0

"Waterfall" by Thomas Strosse is licensed under CC BY-SA 2.0

https://www.deviceplus.com/arduino/entry_003/