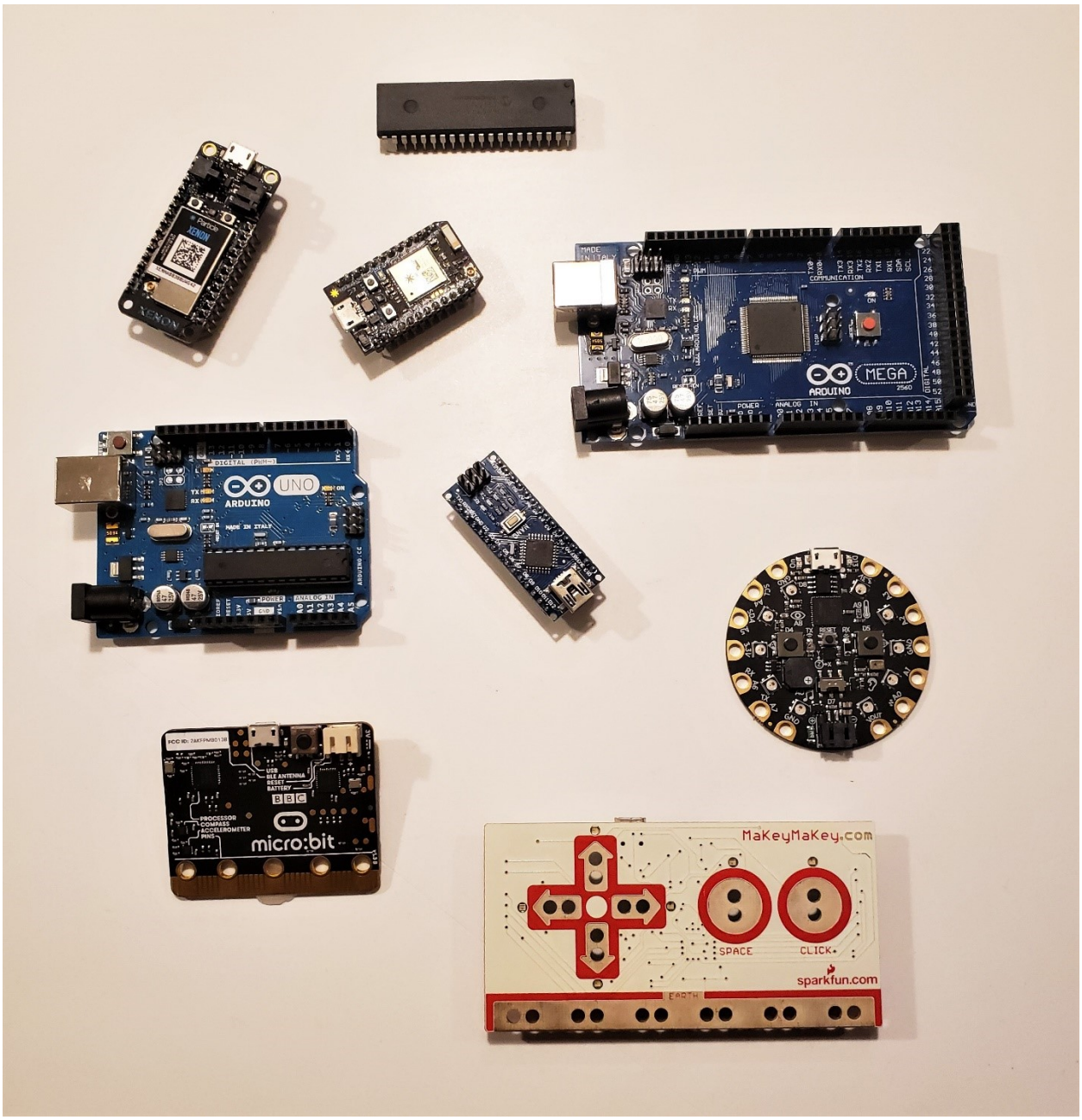
FOXBOT
INDUSTRIES

A Beginner's Guide to Microcontrollers

By Jen Fox (jenfoxbot)



FOXBOT
INDUSTRIES

Contents

Welcome to the wonderful world of microcontrollers!

What do video game controllers, WiFi routers, and radios all have in common? Microcontrollers! Microcontrollers are what power many of our electronic devices. They are particularly useful because they are low power and reliable.

These days, affordable and beginner-friendly microcontrollers can be coded with a regular ol' computer, a USB cable, and some (free) open-source software. Plus, it's a great way to learn or teach coding!

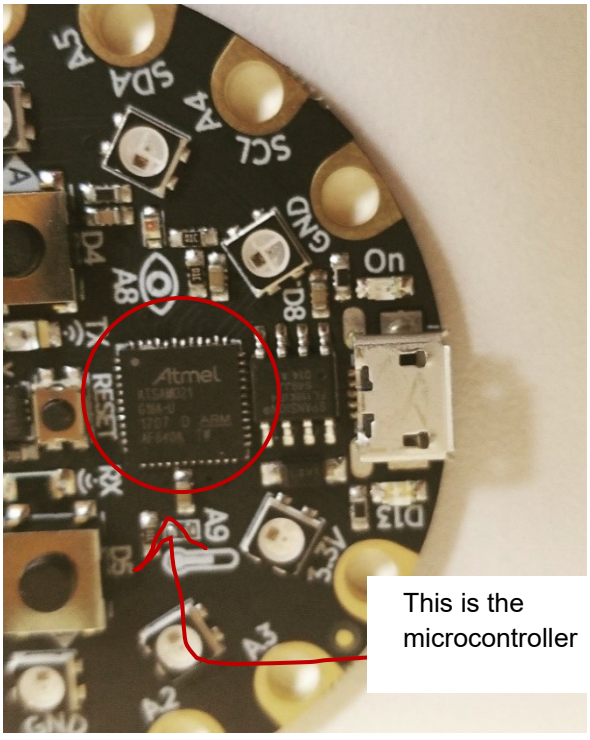
Since microcontrollers can solve so many problems, lots of companies have developed specialized boards. Which is fantastic! But it has also led to an overwhelming amount of choice: there are what feels like 432,000[1] different microcontrollers and it can be daunting to get started, especially if you're just getting into electronics. Where the heck to start?!

Right here! This guide will help you figure out what microcontroller is right for your needs, goals, and budgets. Let's get started!

Wait...What is a microcontroller??

Maybe you've seen this word and were like "uhhh..?" but didn't feel comfy enough to ask.[2] That's OK! You found the right place.

A microcontroller is a "simple computer" that runs one program in a loop. They are designed to perform a single, specific task.



This is the microcontroller

In this guide, we'll be focusing on microcontrollers that have **breakout boards**, or a board that makes it easier to connect to and program the microcontroller.

On a breakout board, the tiny microcontroller pins are soldered, or connected, to a printed circuit board ("PCB"). Then headers or other connectors are added to the PCB, and some basic firmware[3] is loaded to prep the microcontroller to receive signals.

[1] Ok, ok, maybe not *that* many, but definitely a few dozen!
[2] Questions are always good even if they are "dumb", just find a safe space -- like FoxBot Industries!
[3] Firmware is permanent software.

What's the difference between the Raspberry Pi and a microcontroller?

The Raspberry Pi is not only small and adorable, it is also a full-fledged computer!

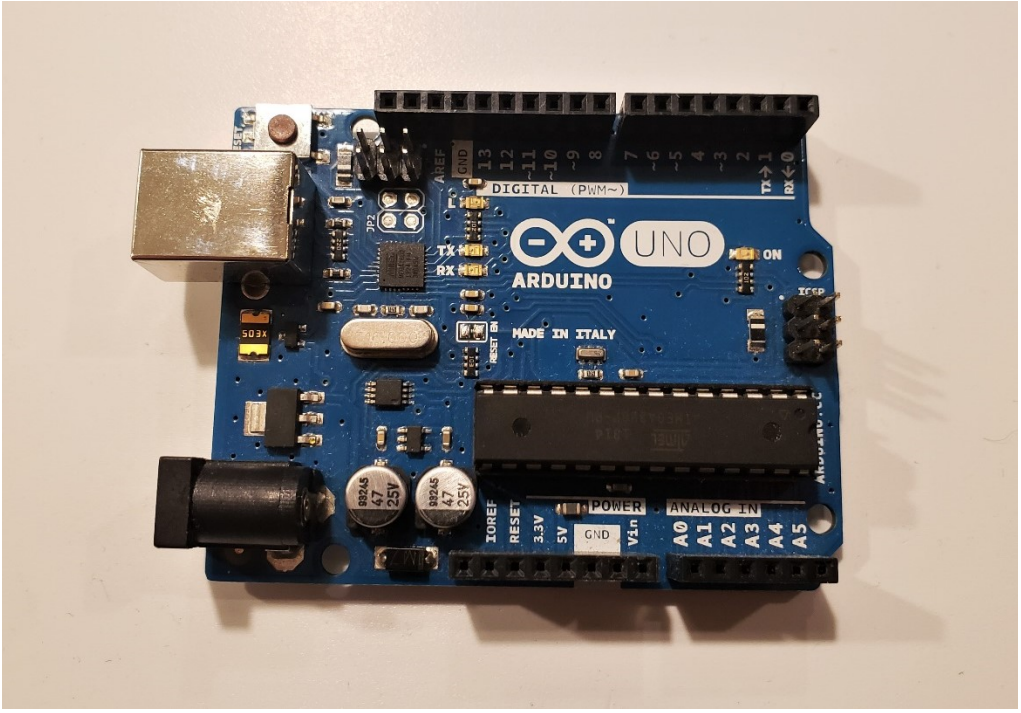
Computers have microprocessors AND microcontrollers that work together to perform many tasks at once.

The microprocessor is what does the "heavy lifting" in a computer. It performs the instructions and calculations that make the computer work. Microprocessors are much faster than microcontrollers, but they need external resources like RAM, Input/Output ports, etc., whereas a microcontroller is typically self-contained.

Computers (which are microprocessors) can run multiple programs at a time: you can surf the Internet, reminisce with old photos, write a paper, and have 1000 tabs open all at the same time! Microcontrollers... not so much. You can use a microcontroller for a specific task, but not multiple tasks all at once.

To learn more about the Raspberry Pi, check out the last section of this tutorial!

Arduino (Uno)



A robust, powerful, and open-source microcontroller and programming environment designed for beginners, **Arduino** is a popular choice for engineers, hobbyists, and folks new to coding to create interactive projects on a budget.

There are many different types of Arduino boards. Pictured above is the Arduino Uno, a great fit for folks just getting started in electronics. Being familiar with Arduino boards and its programming language maps well to careers in computer science, engineering, and design.

Recommended Ages:
12+ with algebra experience

Difficulty:
Intermediate to Expert

Coding language:
Wiring[4]

Average Cost:
$22

Required Accessories:

- ☐ Laptop or desktop
- ☐ Arduino IDE software (free download)
- ☐ USB type A to B cable or barrel jack (for power)

Recommended:

- ☐ Solderless breadboard
- ☐ Jumper cables (male/male, 150mm #758)
- ☐ A parts kit (like the SparkFun Beginner Parts kit or SparkFun Tinker Kit)

[4] The Wiring programming language is a combination of the C++ and Processing languages.

Sample Program: Blink an LED

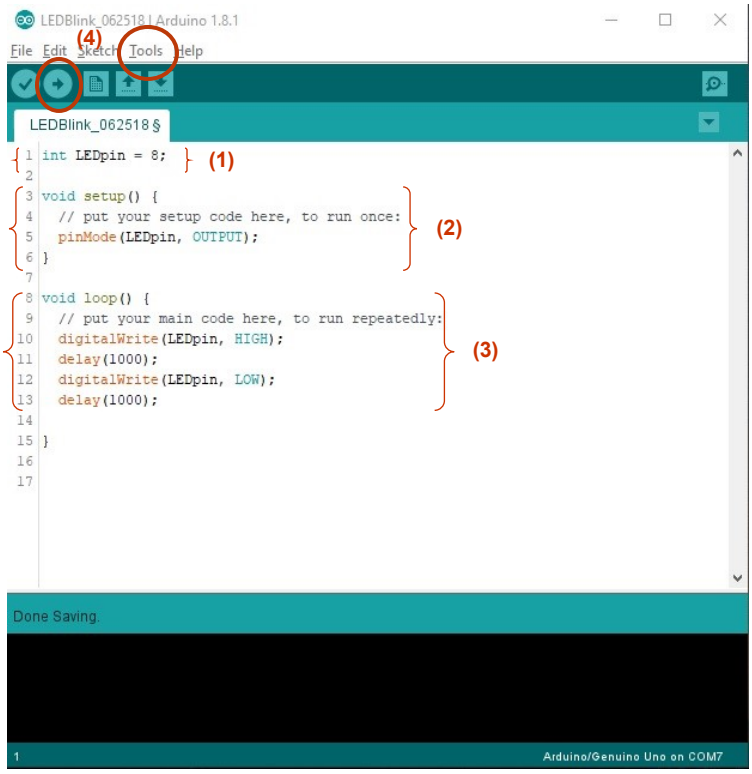
In addition to your Arduino Uno, **you will also need**: a breadboard, 1 LED (any color), 1 resistor (e.g. 100 Ω), and 2 jumper wires.

1. Define the pin number(s) at the top of the program.

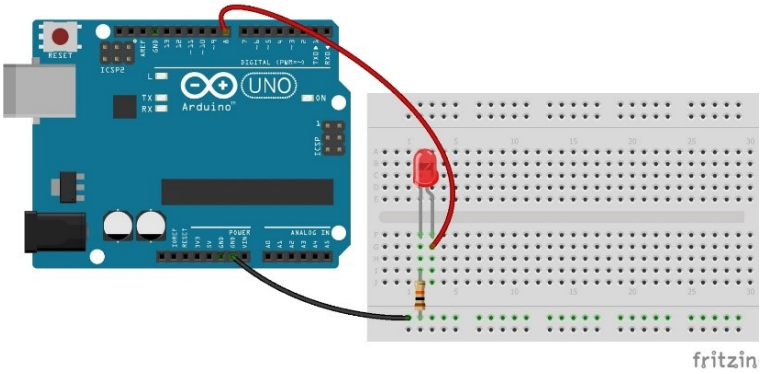
2. In *setup* loop, initialize the input and output pins, serial monitor, etc.

3. In the *loop* loop, write the program you want the microcontroller to perform.
   a. The microcontroller will read the instructions from top to bottom, then repeat until you unplug it.
   b. When you plug in the microcontroller, the existing code will run the *setup* loop once, then the *loop* loop until it is unplugged again or reprogrammed.

4. Choose the board and port (under tools), then upload the code.



Wire your Arduino like this:



Hardware Features

Input and Output ("I/O") Pins	14 Digital 6 Analog 2 Power Out (3.3V, 5V) 3 GND
Power Supply	5 – 12 V, USB or barrel jack
Additional Features	ICSP header: for connecting add-on boards called shields, like a WiFi shield

Learn More: Arduino Website (www.Arduino.cc)

micro:bit

The micro:bit is a great educational tool for teaching others how to code and work with hardware, particularly elementary school students.

Created through a collaboration between Microsoft and the BBC, the micro:bit was designed to bring coding to classrooms all over the world. Students use a free, in-browser coding platform, MakeCode, to write code for the micro:bit and control lights, motors, speakers, and more!

MakeCode is primarily a block-based coding platform, but may also be used in Javascript. With some extra setup, the micro:bit can also be programmed in micropython, or python for microcontrollers. This makes the micro:bit a good transition for kids "done with Scratch" who are ready for more powerful coding languages.

Recommended Ages:
8+

Difficulty:
Beginner

Coding language:
Block-Based
(also Javascript and MicroPython)

Average Cost:
$15

Required Accessories:
- ☐ Laptop or desktop with Internet access
- ☐ microUSB to USB cable (for power)

Recommended:
- ☐ Alligator clips
- ☐ Servo motor(s)
- ☐ Headphones (with standard 3mm audio connector)
- ☐ micro:bit edge connector (for more connections)

Sample Program: Blink an LED

Plug your micro:bit into the computer using the microUSB cable, then navigate to the MakeCode website (www.MakeCode.org), select the micro:bit option, and create a new file.

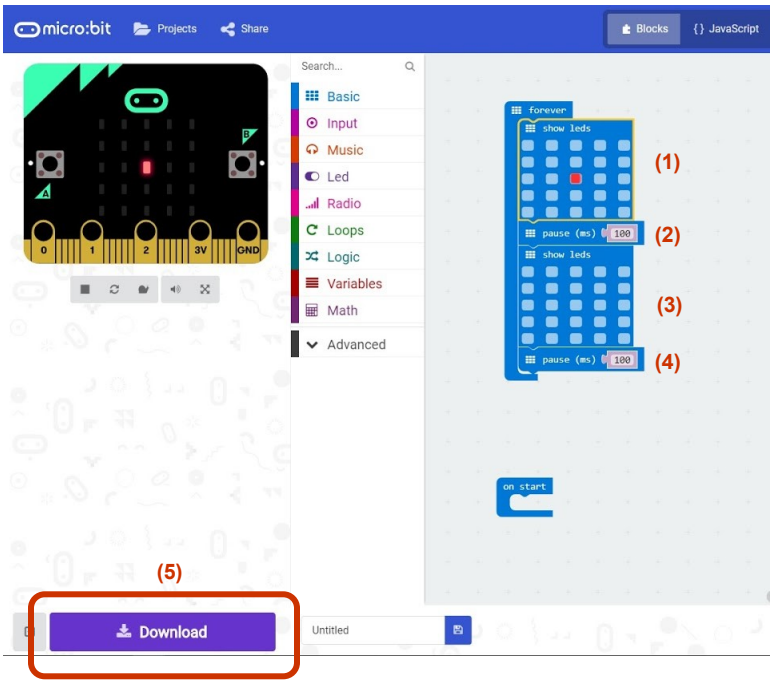
1. Under Basic, drag the "show LEDs" block into the "forever" loop.
    a. Click on one or more squares to turn the LED on.

2. Under Basic, drag the "pause" block into the forever loop.

3. Drag another "show LEDs" block, but leave this one blank.

4. Drag another "pause" block to the bottom.

5. Check the simulation on the left side of the screen, and then upload!
    a. Upload the code by saving it directly to the micro:bit drive[5], or saving it to your computer and then dragging the file onto the micro:bit drive.



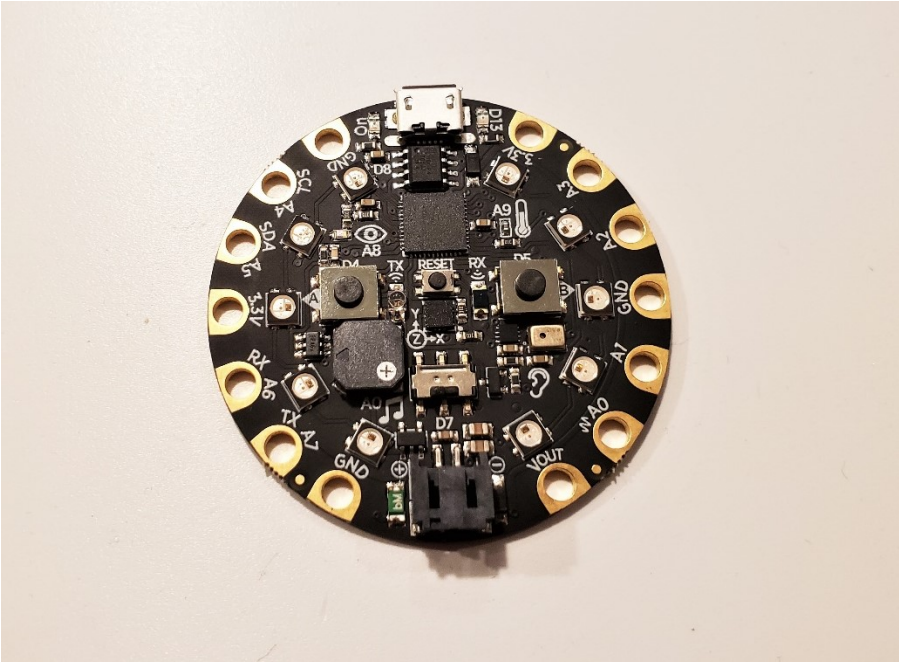
Hardware Features

Input and Output ("I/O") Pins	3 Digital and Analog rings *(also touch sensitive)* 1 Power Out (3.3V) 1 GND *More I/O pins available with an edge connector*
Power Supply	3 - 5 V (microUSB or battery pack)
Additional Features	5x5 LED matrix Two (2) pushbuttons Radio transmitter and receiver Accelerometer Compass Light and Temperature Sensors

Learn More: Micro:Bit Website (www.microbit.co.uk)

[5] When the micro:bit is connected to a computer, it will show up as a drive folder, much like a USB stick.

Circuit Playground Express



Created by Adafruit Industries, the Circuit Playground Express, or CPX, is a versatile, feature-loaded microcontroller great for kids and folks just getting started with coding and hardware.

It is a helpful tool for learning how to code, teaching others how to code, and making quick prototypes for beginners to experts.

*Note: There is also the **Circuit Playground Classic** -- the hardware is nearly identical, but this version can be programmed in the Arduino IDE.*

Recommended Ages:
8+

Difficulty:
Beginner

Coding language: Block-Based
(also Javascript, C++, MicroPython, and Wiring)

Average Cost:
$25

Required Accessories:

- ☐ Laptop or desktop with Internet access
- ☐ microUSB to USB cable (for power)

Recommended:

- ☐ Alligator clips
- ☐ Servo motor(s)
- ☐ Conductive Tape

Sample Program: Blink an LED

Plug your CPX into the computer using the microUSB cable, then navigate to the MakeCode website (www.MakeCode.org), select the CPX option, and create a new file.

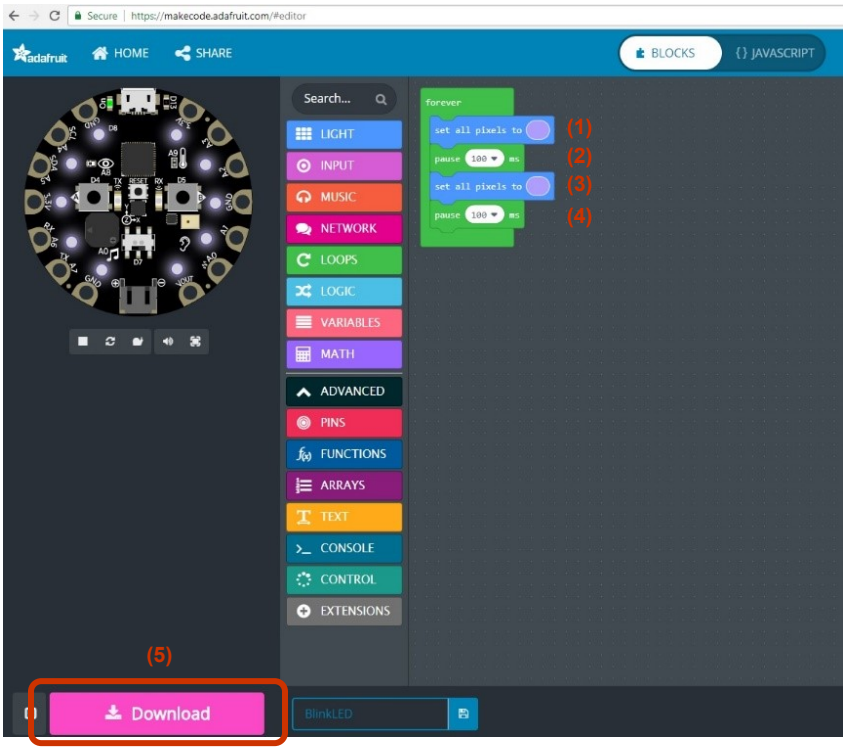
1. Under Light, drag the "set all pixels" block into the *forever* loop.
    a. Click on the color to change it.

2. Under Loops, drag the "pause" block into the *forever* loop under the first block.

3. Drag another "set all pixels" block into the *forever* loop under the pause block.

4. Drag another "pause" block to the bottom of the *forever* loop.

5. Check the simulation, and then upload!
    a. To upload, save the code file onto your computer. Next, hold down the CPX reset button until all of the neopixels turn **green**. Finally, drag the CPX code file onto the CPX drive[6].



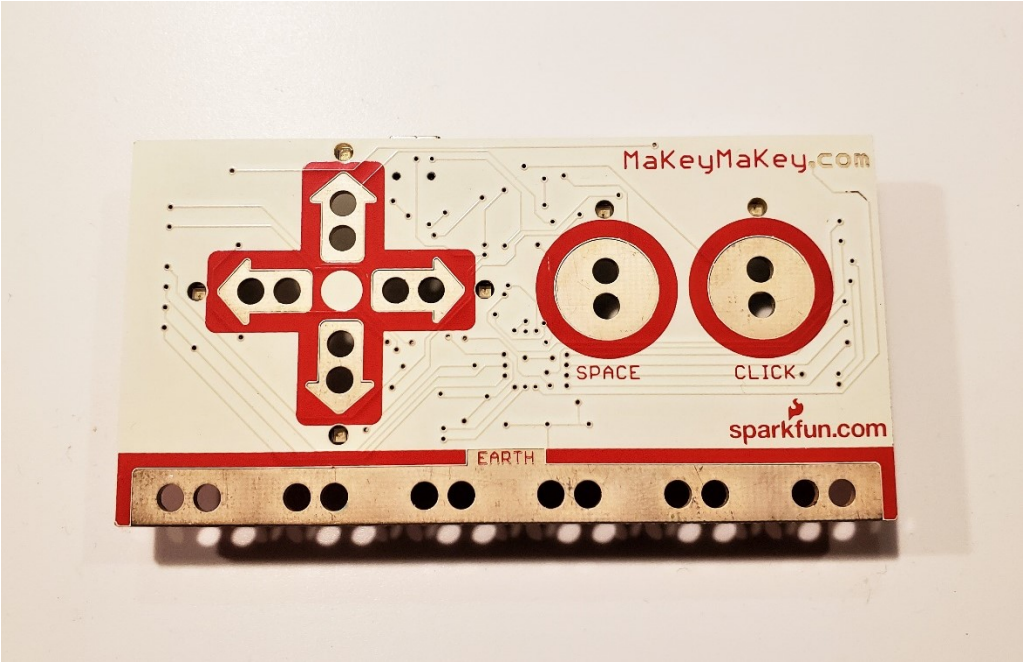
Hardware Features

Input and Output ("I/O") Pins	7 Digital and Analog rings *(also touch sensitive)* 2 Power Out (3.3V) 3 GND
Power Supply	3 - 5 V (microUSB or battery pack)
Additional Features	10 neopixels (can be many colors!) 2 pushbuttons 1 slide switch Infrared transmitter and receiver Accelerometer Sound sensor and mini speaker Light and Temperature Sensors

Learn More: Adafruit Industries (www.Adafruit.com)

[6] When the Circuit Playground Express is connected to a computer, it will show up as a drive folder, much like a USB stick.

Makey Makey



An interactive introductory microcontroller great for young kids and folks new to electronics and coding, especially for those who want to play with technology without having to build circuits and code.

The Makey Makey is a great first step into electronics and technology - no programming required! Connect alligator clips between the pads and any material that is somewhat conductive, like hands, fruit, or metal objects, to trigger keyboard and mouse keys.

This is an Arduino-compatible board, meaning that you can also reprogram it using the Arduino Integrated Development Environment ("IDE").

Recommended Ages:
5+

Difficulty:
Beginner

Coding language: None required
(can be coded in Wiring if desired)

Average Cost:
$50

Required Accessories:

- ☐ Laptop or desktop (Internet connectivity recommended)
- ☐ microUSB to USB cable (for power)

Recommended:

- ☐ Alligator clips
- ☐ Fruit or other conductive objects
- ☐ Aluminum Foil

Sample Project: Banana Piano

You'll also need: 7 alligator clips and 6 bananas or other conductive objects!

1. Connect your first alligator clip to the ground pad on the bottom.

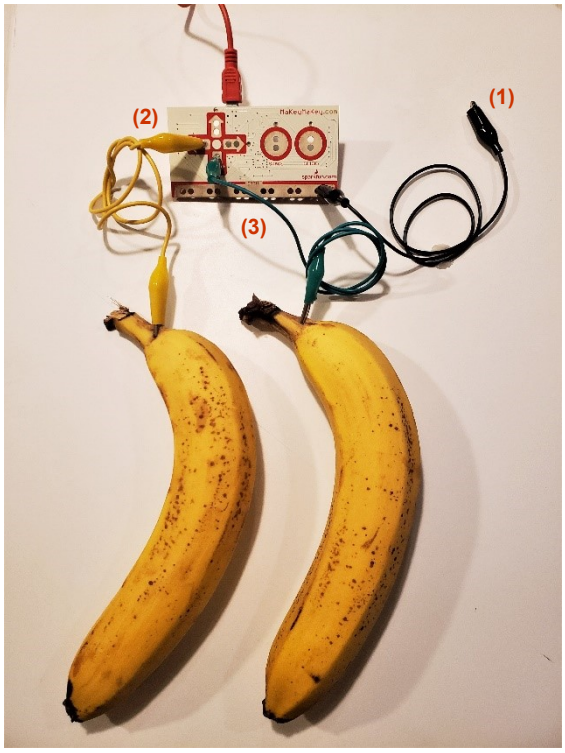
2. Connect your next alligator clip between the first banana and one of the Makey Makey arrow keys. Repeat for the remaining arrow keys.

3. With your next alligator clip, connect it between the Makey Makey space pad and a banana.

4. Finally, connect your last alligator clip between the Makey Makey click pad and the last banana.

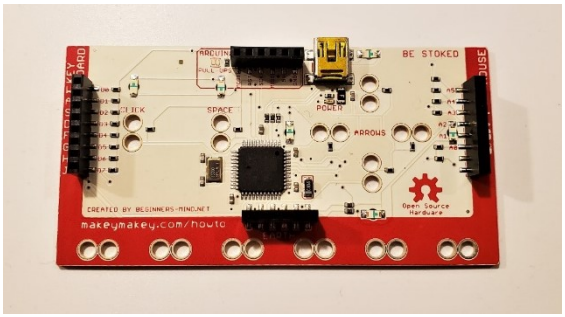
5. Plug in the Makey Makey to your computer, then navigate to the piano app here (https://apps.makeymakey.com/piano/)

6. Hold the metal end of the ground pad alligator clip from step 1, then touch any of the bananas to trigger!



Hardware Features

Touch Inputs (front)	6 Touch Inputs: • 4 to control keyboard arrow keys • 1 to control keyboard spacebar • 1 to control left mouse click
Power Supply	3 - 5 V (microUSB)
Additional Features (on back)	3 General I/O pins 1 5V Power Pin 1 GND Pin 6 pins to control keyboard letters 4 pins to control keyboard arrow keys 2 pins to control right and left mouse click 1 pin to control spacebar



Learn More: Makey Makey website (www.makeymakey.com)

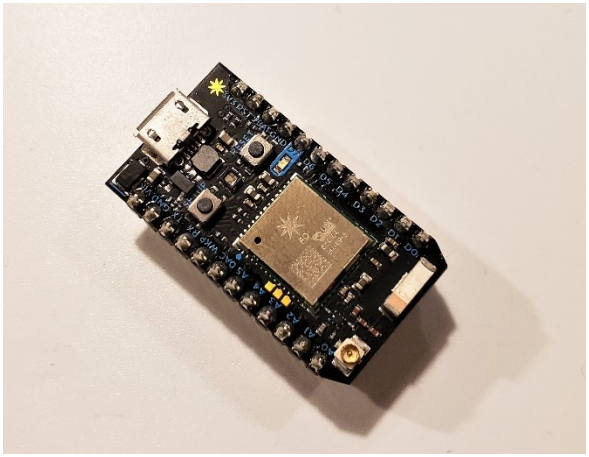
Other Popular Boards

One guide cannot do justice to the plethora of readily available microcontrollers. If you have a super specific specialty need, there is probably a microcontroller designed just for that (like smartphone apps!). To get a feel for other types of microcontrollers, peruse the inventories of SparkFun Electronics and Adafruit Industries or ask folks in the field!

Here are a few of my other favs:

Particle Photon

The Photon is a WiFi-connected microcontroller that can be programmed wirelessly. The easiest setup uses a (free) smartphone app to control its inputs and outputs. If the app is insufficient for your needs, connect the Photon to your computer with a microUSB cable, navigate to the Particle web-based coding platform, and write more complex programs!



Recommended Ages:
12+ with algebra experience

Difficulty:
Intermediate

Coding language: Wiring[7]
(also C/C++ or ARM assembly)

Average Cost:
$20

Required Accessories:

- ☐ Smartphone
- ☐ microUSB to USB cable (for power and/or coding)

Recommended:

- ☐ Laptop or Desktop with Internet Connection
- ☐ LEDs
- ☐ Assorted sensors

Learn More: Particle online store (www.store.particle.io/collections/photon)

[7] Wiring is the code framework, so most Arduino code will work without modifications. Can also write in C/C++ or ARM assembly

TinyPICO - ESP32 Development Board

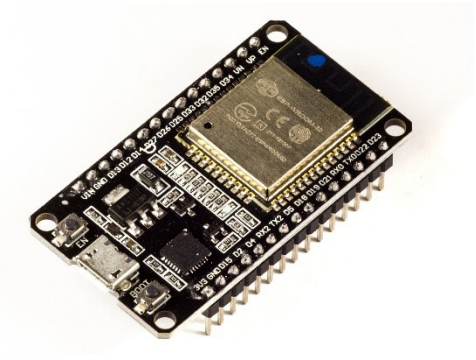
A super small, super powerful WiFi-enabled microcontroller that is popular in the IoT[8] community. This is a newer, "super-charged" version of the beloved ESP8266 IoT microcontroller.[9]

Recommended Ages:
14+ with some coding and hardware experience

Difficulty:
Intermediate++

Coding language:
MicroPython

Average Cost:
$20



Source: Wikimedia Commons

Learn more: Adafruit TinyPICO webpage (https://www.adafruit.com/product/4335)

Note: SparkFun Electronics has a similar board: the "ESP32 Thing" for $22 (https://www.sparkfun.com/products/13907).

Adafruit Trinket M0

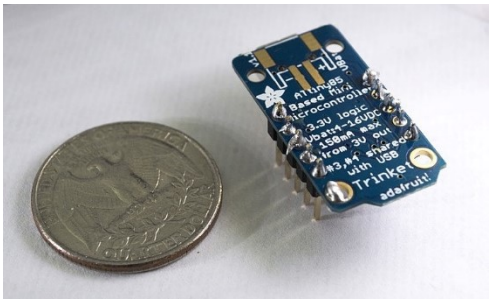
A teeny tiny yet powerful microcontroller that blurs the lines between computer and microcontroller as it has microprocessor onboard. There are lots of other M0 boards, so search around if this doesn't quite fit your needs or your fancy!

Recommended Ages:
14+ with some coding and hardware experience

Difficulty:
Intermediate++

Coding language:
CircuitPython[10]

Average Cost:
$9

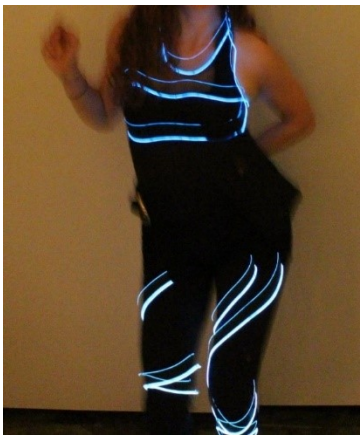


Source: Wikimedia Commons

Learn more: Adafruit Tinket M0 webpage (https://www.adafruit.com/product/3500)

[8] IoT stands for "Internet of Things", which is the term that refers to connecting various hardware devices, like sensors and household electronics, to the Internet.
[9] Please note that both the ESP32 and ESP8266 have potential security vulnerabilities. It is recommended to only use them for hobbyist applications.
[10] Just like MicroPython, CircuitPython is a flavor of Python for microcontrollers.

Wearable Microcontrollers

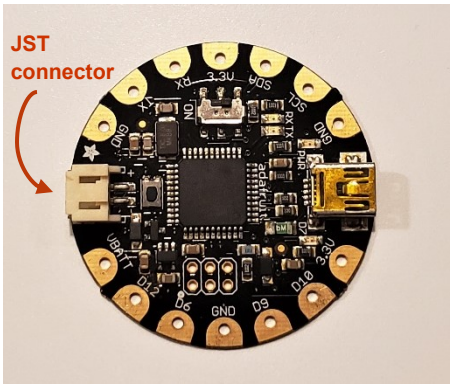


For you fashion savvy folks, there are also a handful of microcontrollers designed for wearable projects! These boards are special because they can be washed, so you don't have to remove them from clothing (but **do remove the battery**).

Wearable microcontrollers also have custom I/O pins that make it easier to sew into clothing and stitch circuits with conductive thread. Here are a few of my favs:

Adafruit FLORA

A circular, sewable microcontroller with 14 inputs and outputs. You can program it with a standard microUSB cable and then power it via battery pack with a JST connector!



Recommended Ages:
12+ with some algebra experience

Difficulty:
Intermediate

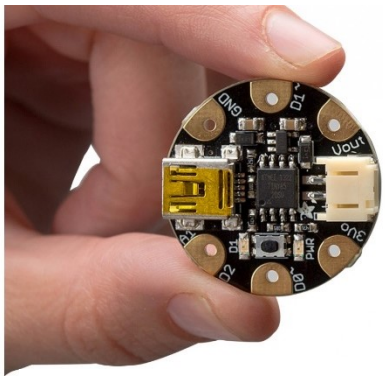
Coding language:
Wiring (Arduino IDE)

Average Cost:
$15

Learn more: Adafruit FLORA webpage (https://www.adafruit.com/product/659)

Arduino Gemma

A tiny lil' sewable microcontroller with just a few inputs and outputs. Perfect for hiding, connecting to small objects, and creating jewelry.



Source: Wikimedia Commons

Recommended Ages:
12+ with some algebra experience

Difficulty:
Intermediate

Coding language:
Wiring (Arduino IDE)

Average Cost:
$5

Learn more: Arduino Gemma webpage (https://store.arduino.cc/arduino-gemma)

Arduino Lilypad

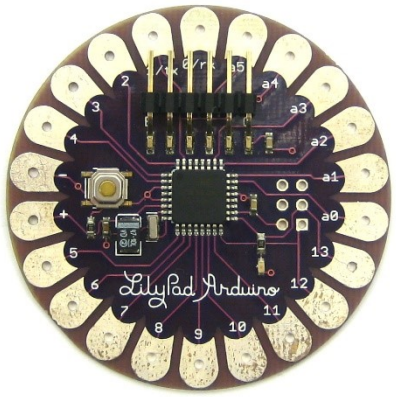
A circular sewable microcontroller with 14 available inputs and outputs. Requires an FTDI[11] cable for power and programming.

Recommended Ages:
12+ with some algebra experience

Difficulty:
Intermediate

Coding language:
Wiring (Arduino IDE)

Average Cost:
$25



Source: Wikimedia Commons

Learn more: SparkFun Lilypad webpage (https://www.sparkfun.com/products/14631)

By the way, what's the difference between digital and analog?

Digital signals are binary signals, meaning they can only have two states: on or off, like a light switch.

Analog signals are continuous, which means they can be any value between two extremes, like temperature or speed.

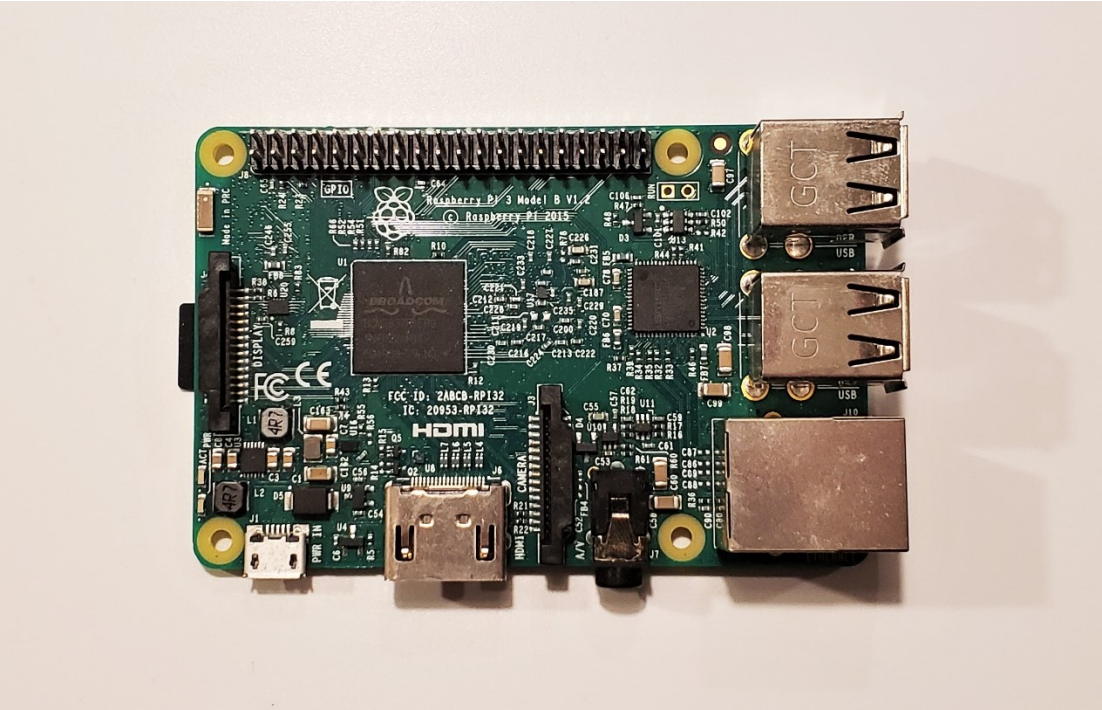
Most microcontrollers with breakout boards support both analog and digital signals, so you can read in and output both types.

However, there are many different types of digital signals. Some are easier to handle than others.

Before purchasing a sensor or other electronic component, check the part description to see what kind of signal it needs. Do a little research to be sure that you are familiar and comfortable programming with that type of signal.

[11] An FTDI cable is a special type of USB to Serial converter commonly used for microcontrollers.

Raspberry Pi



The Raspberry Pi, or Pi for short, is a credit-card sized computer[12] that runs a special version of Linux and can be programmed to control hardware. Due to its low cost, it is a great first computer, especially for learning to code. The Pi is also widely used by hardware experts to build all sorts of electronic projects, from robots to 3D printers to home automation systems!

There are a few different versions of the Pi, the most recent is the Raspberry Pi 4 and the Pi Zero, a miniature version for just $10.

Recommended Ages:
12+ with algebra experience

Difficulty:
Intermediate

Coding language: Python
(can also use as a typical computer to practice other coding languages)

Average Cost:
$35

Required Accessories:

- ☐ SD Card (4GB or larger)
- ☐ microUSB power supply (5V)
- ☐ Monitor (HDMI compatible)
- ☐ HDMI Cable
- ☐ Keyboard
- ☐ Mouse

Recommended:

- ☐ GPIO Connector
- ☐ Assorted sensors
- ☐ Speakers or headphones (3mm audio jack)

[12] The Pi can be used similar to a standard microcontroller AND can also control microcontrollers! Basically, the Pi is super awesome and I *have* to include it even though it is technically a computer :)

Sample Program: Blink an LED

First, we need to install a Python library to use the GPIO pins. Do this by opening the command prompt and typing the following:

```
$ sudo apt-get install python-rpi.gpio python3-rpi.gpio
```

Press enter to execute the command and start the installation process. Next, open your favorite Python editor! If you are unsure where to start, use the pre-installed IDLE3 program. Inside the editor, write the following program:

```
1   #Blink LED Program
2   import RPi.GPIO as GPIO # Import Pi GPIO library
3   from time import sleep # Import sleep function from  time library
4
5   GPIO.setmode(GPIO.BOARD) # Use physical pin numbering
6   GPIO.setup(8, GPIO.OUT, initial=GPIO.LOW) # Set pin 8 as output
7
8   while True: # Run forever
9     GPIO.output(8, GPIO.HIGH) # Turn on pin 8
10    sleep(1) # Sleep for 1 second
11    GPIO.output(8, GPIO.LOW) # Turn off pin 8
12    sleep(1) # Sleep for 1 second
```

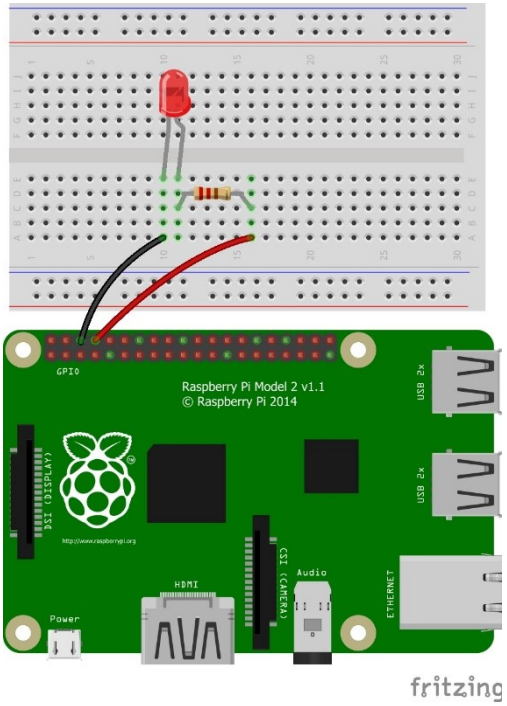
Save the file as "ledBlink.py" somewhere easily accessible, like the Desktop. Then, open the command console and navigate to where you saved the file. Use the command `cd`, like so:

```
$ cd home/pi/Desktop
```

Run the python program with the following command:

```
$ sudo python ledBlink.py
```

Finally, wire up an LED and resistor to your Pi as shown in the diagram to the right. And behold, a blinking light!



Hardware Features

Board Features	4 USB ports 1 Ethernet port 1 HDMI port 1 Audio jack (3mm) 1 Camera module port
General Purpose Input and Output ("GPIO") Pins	40 pins in total: • 26 Digital I/O (no Analog) • 4 Power out (two 3.3V and two 5V) • 8 GND • 2 Specialty (advanced use only)
Power Supply	5V (microUSB)

Learn More: Raspberry Pi Foundation (www.RaspberryPi.org)

Final Thoughts and Recommendations

If you are just getting started and want to build lots of different projects, I'd recommend the Circuit Playground Express. It's super easy to get up and running and has a ton of onboard gadgets to play with and explore.

If you are super interested in computer networking, AI and machine learning, or connecting things to the Internet (e.g. making a "Smart Home"), I'd suggest the Raspberry Pi.

If you want a sturdy, stable, and reliable board to build a wide variety of projects, go with an Arduino.

If you still have no idea where to start and are totally intimidated, go with the Micro:Bit – at $15 it is the most affordable, beginner-friendly option and has plenty of fancy features to keep you busy. Plus, if you get one for your friend, you can send messages via Bluetooth!

In general, to learn electronics, microcontrollers, and coding, the best advice I can give you is to **find a project you are passionate about and build it**! There are tons of tutorials online so search around for someone who has built the same or similar project. Build off of their findings and adjust as you please. Have fun, try new things, and before you know it, you'll have all sorts of new skills and knowledge!

Happy hacking!

Need more help?

Contact us for personalized assistance, project ideas, and more! We will provide a **free 30-minute consultation** to get you started.

If you need more guidance or want to learn specific skills, we offer customized training programs to help you achieve your learning and career goals. Contact us for a quote!

Website: www.FoxBotIndustries.com
E-mail: jen@foxbotindustries.com
Phone: (206) 395- 5195

About the Author and FoxBot Industries

About the Author and Founder of FoxBot Industries, Jen Fox (aka jenfoxbot)

Jen Fox is an engineer, a maker, and an educator. After dabbling in dark matter, Fox settled into engineering and inventing to solve problems related to climate change and social justice. Combining her varied interests and passion for learning, Fox founded FoxBot Industries in mid-2015 to provide an arts-based approach to STEM education.

Fox's research contributions in physics are published in multiple journals, including Astroparticle Physics, the Journal of Instrumentation, and Nuclear Instruments and Methods in Physics Research. Fox's engineering work has been featured by the Raspberry Pi Foundation, Make: Media, SparkFun Electronics, Adafruit Industries, the Seattle Times, and many more. She has taught STEM subjects to thousands of students in a variety of settings, from week-long maker camps for kids to university physics courses at UCLA.

About FoxBot Industries

Our mission is to empower and educate technical and non-technical folks alike to ignite their curiosity, unlock their potential, and become an agent of positive change.

To do this, we provide an arts-based approach to STEM (Science-Technology-Engineering-Mathematics) **education.**

We teach our students how to ask questions, think critically, and use STEM subjects as a tool to explore the world, solve problems, and build better systems.

We believe that education is the key to a successful, equitable, and progressive society. The more we know, the more problems we can solve and the better we can solve them!

We provide resources and activities for students of all ages and for formal educators in STEM subjects that integrate with the Next Generation Science Standards (NGSS).

We promote and advocate for: kindness and curiosity; equal rights and education for all persons; environmental conservation; and reduction of consumption and waste.