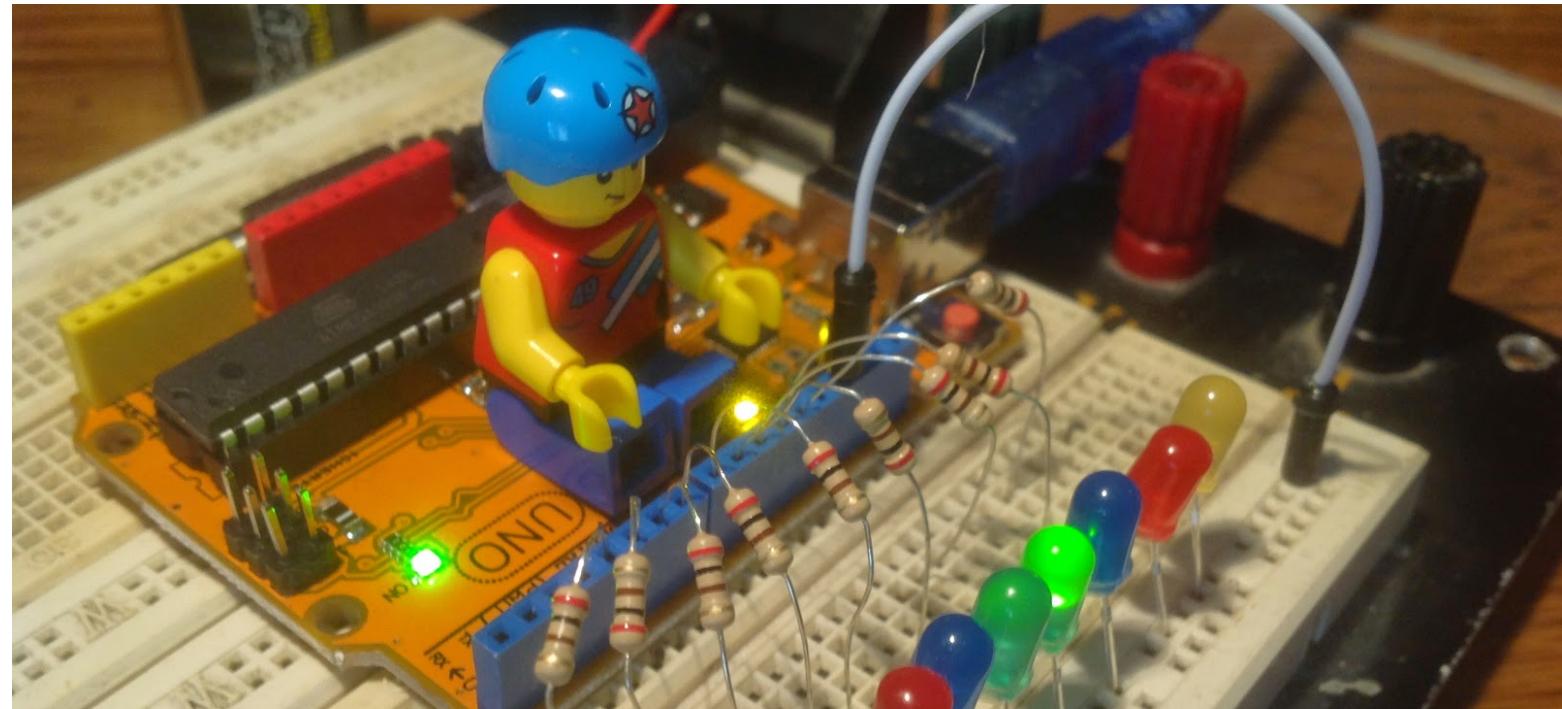


Introduction to Arduino & Microcontroller Programming

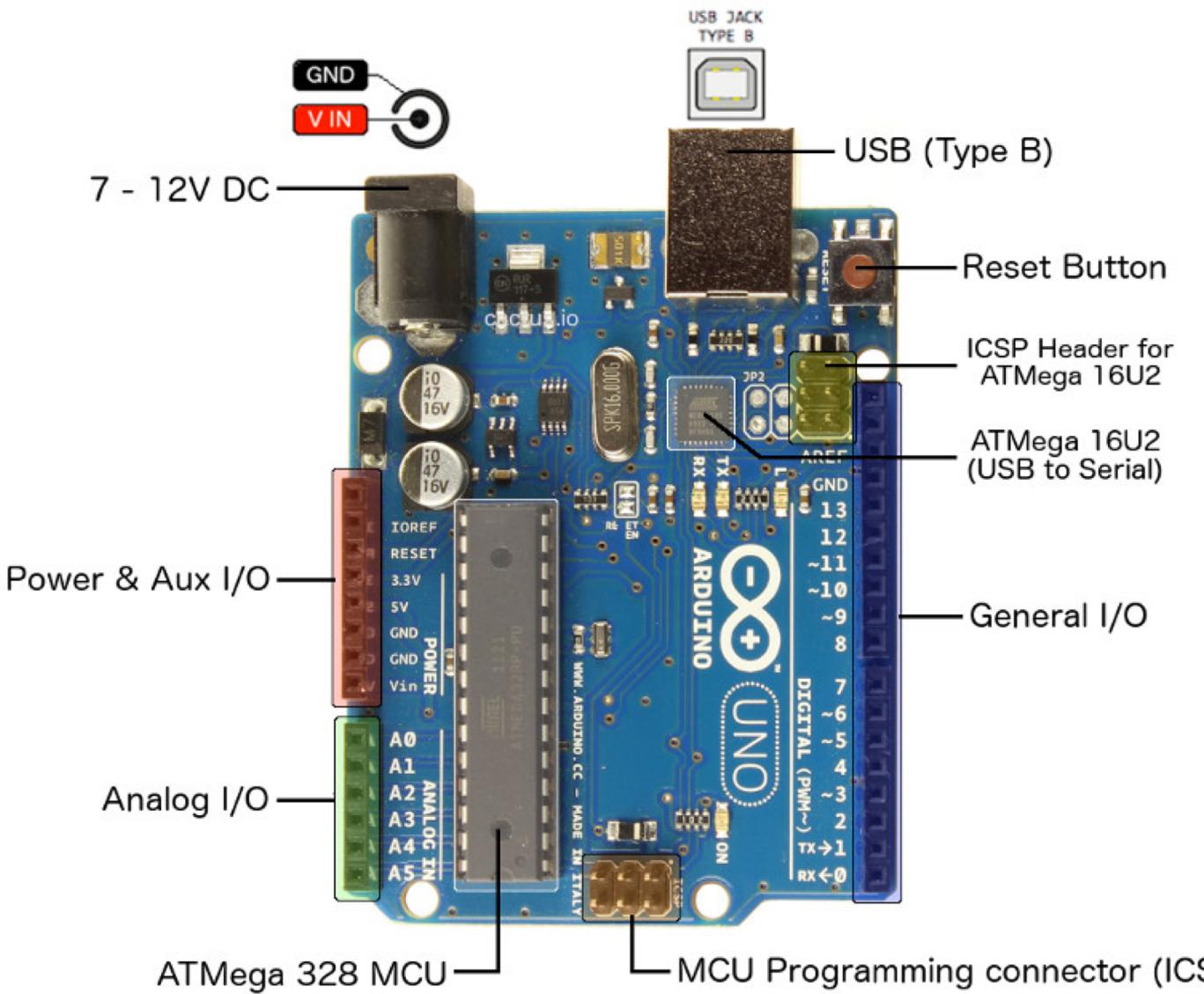
CODING BOOTCAMP @ BELKIN
APR-2019

What is Arduino?

- A microcontroller board
(*a small single chip computer*)
- A USB connection for programming and basic communication
- Has a number of connections that can be wired to external electronics for gathering data or control (GPIO).
- User friendly opensource hardware, software, and tools designed for learning programming and electronics



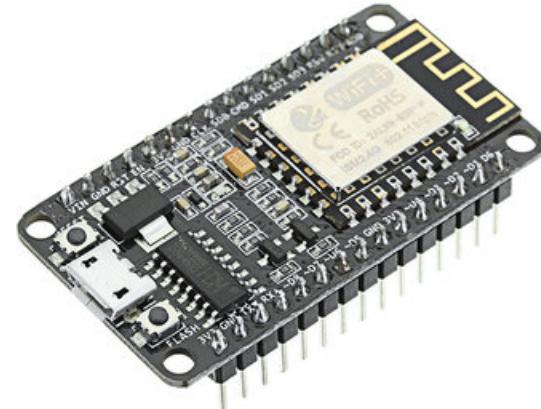
Arduino Uno Rev 3

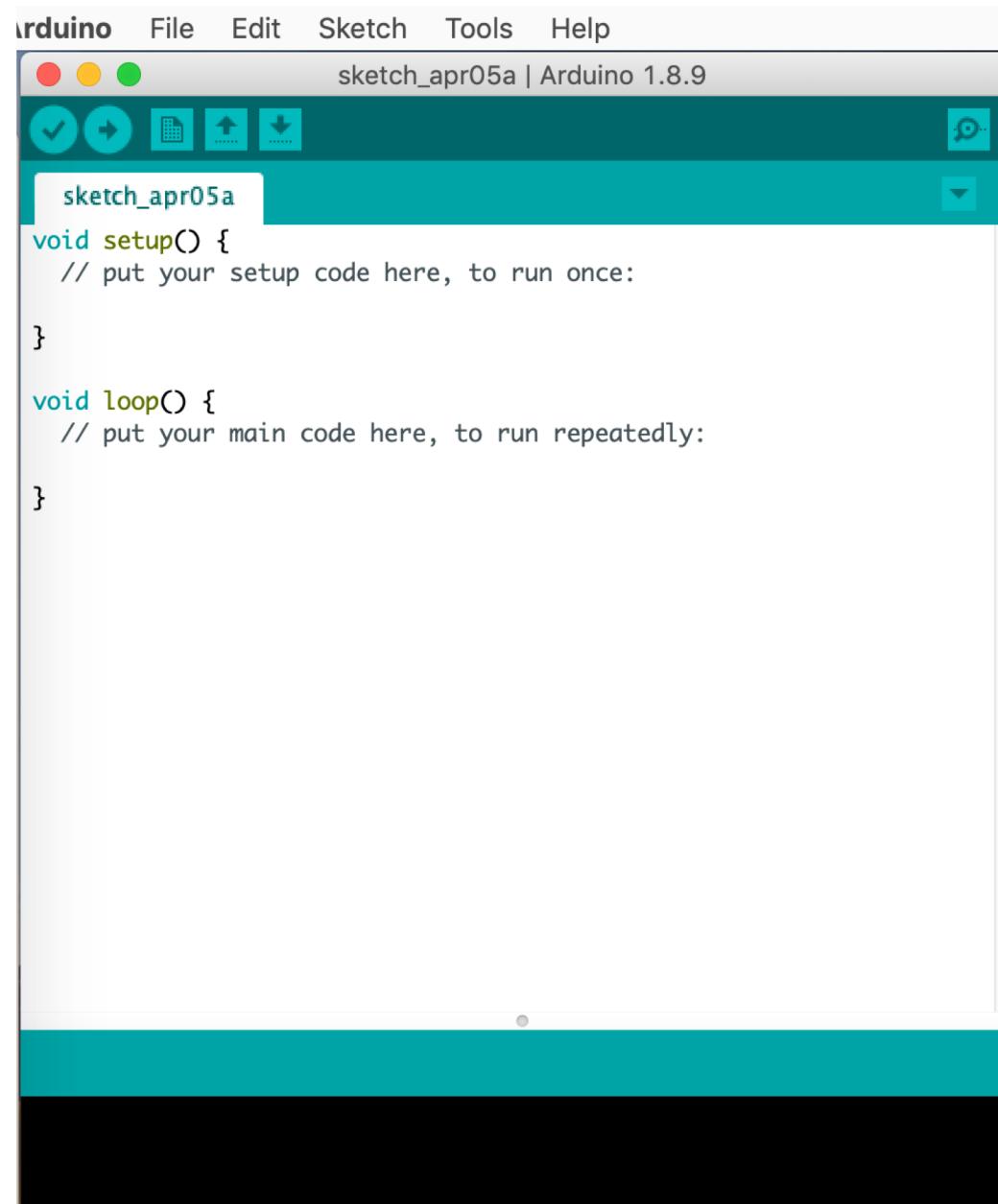


Meet Arduino Uno

NodeMCU

- Arduino compatible development board with built in 2.4Ghz Wifi
- C/C++ or Lua programming environment
- "Bread board friendly"
- An open source IOT platform
- Low cost





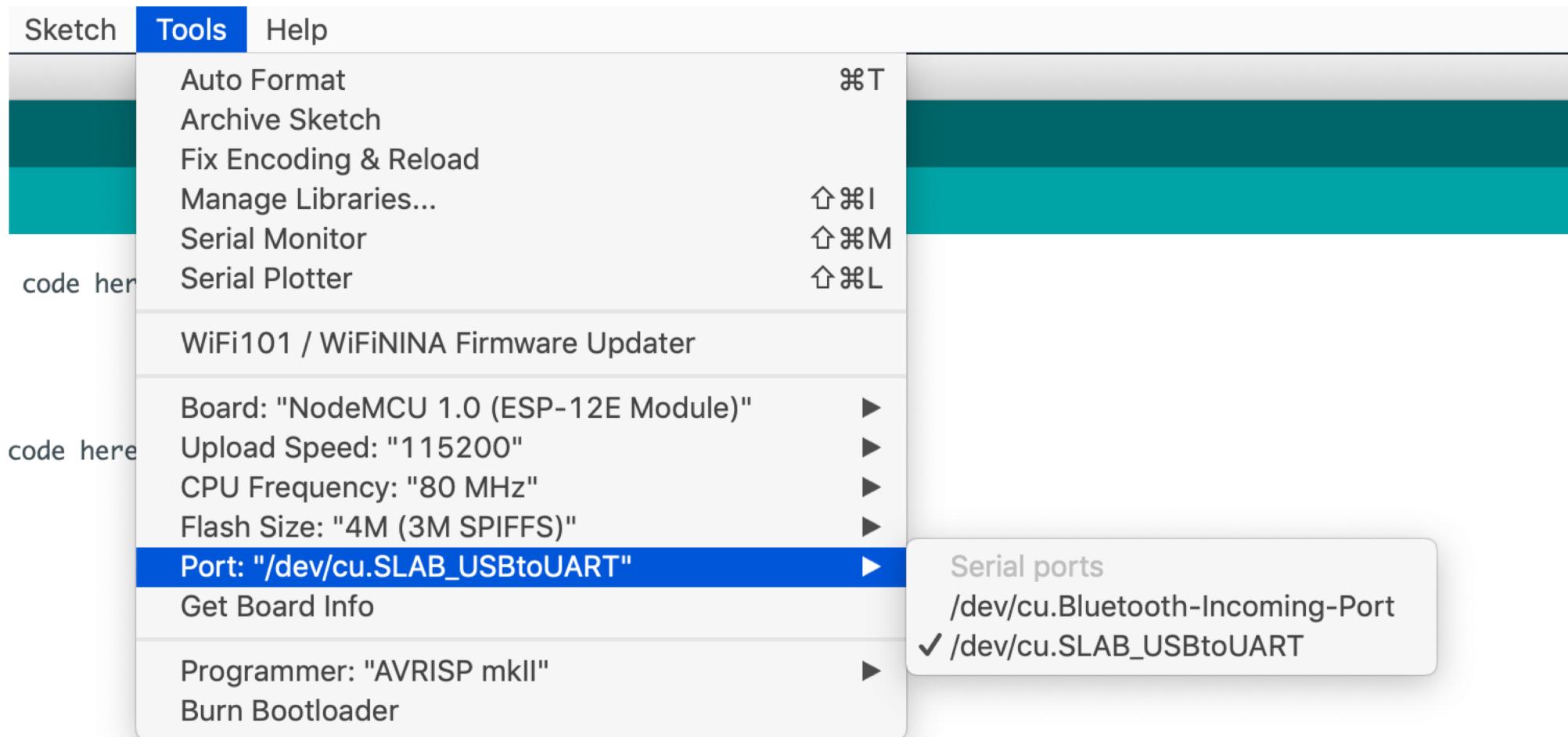
The image shows a screenshot of the Arduino IDE version 1.8.9. The window title is "sketch_apr05a | Arduino 1.8.9". The menu bar includes "Arduino", "File", "Edit", "Sketch", "Tools", and "Help". Below the menu is a toolbar with icons for file operations like Open, Save, and Print. The main area displays the code for "sketch_apr05a". The code consists of two functions: "void setup() { ... }" and "void loop() { ... }". The "setup()" function contains a comment: "// put your setup code here, to run once:". The "loop()" function also contains a comment: "// put your main code here, to run repeatedly:". The code is written in a syntax highlighted text editor.

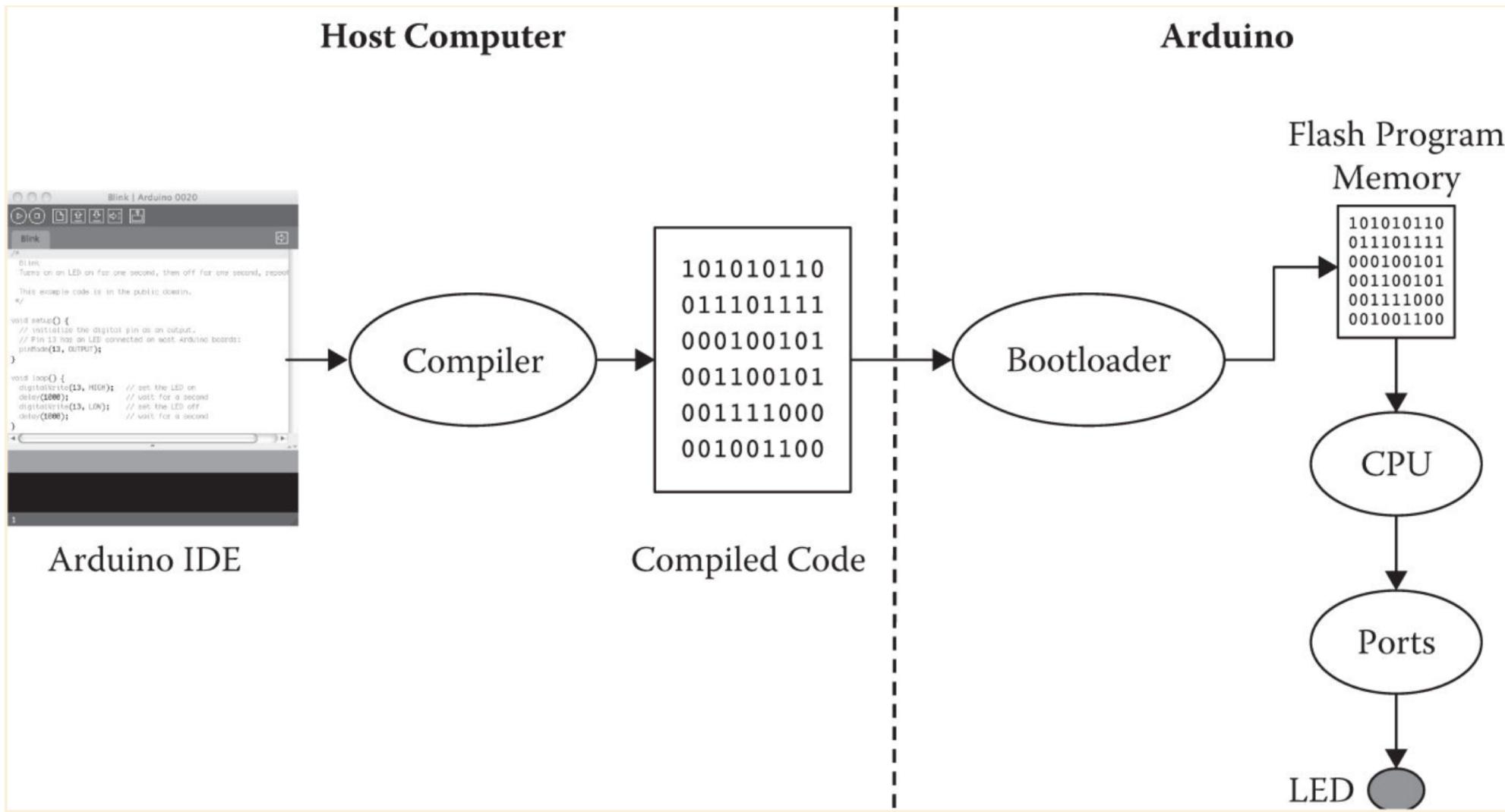
```
void setup() {
  // put your setup code here, to run once:
}

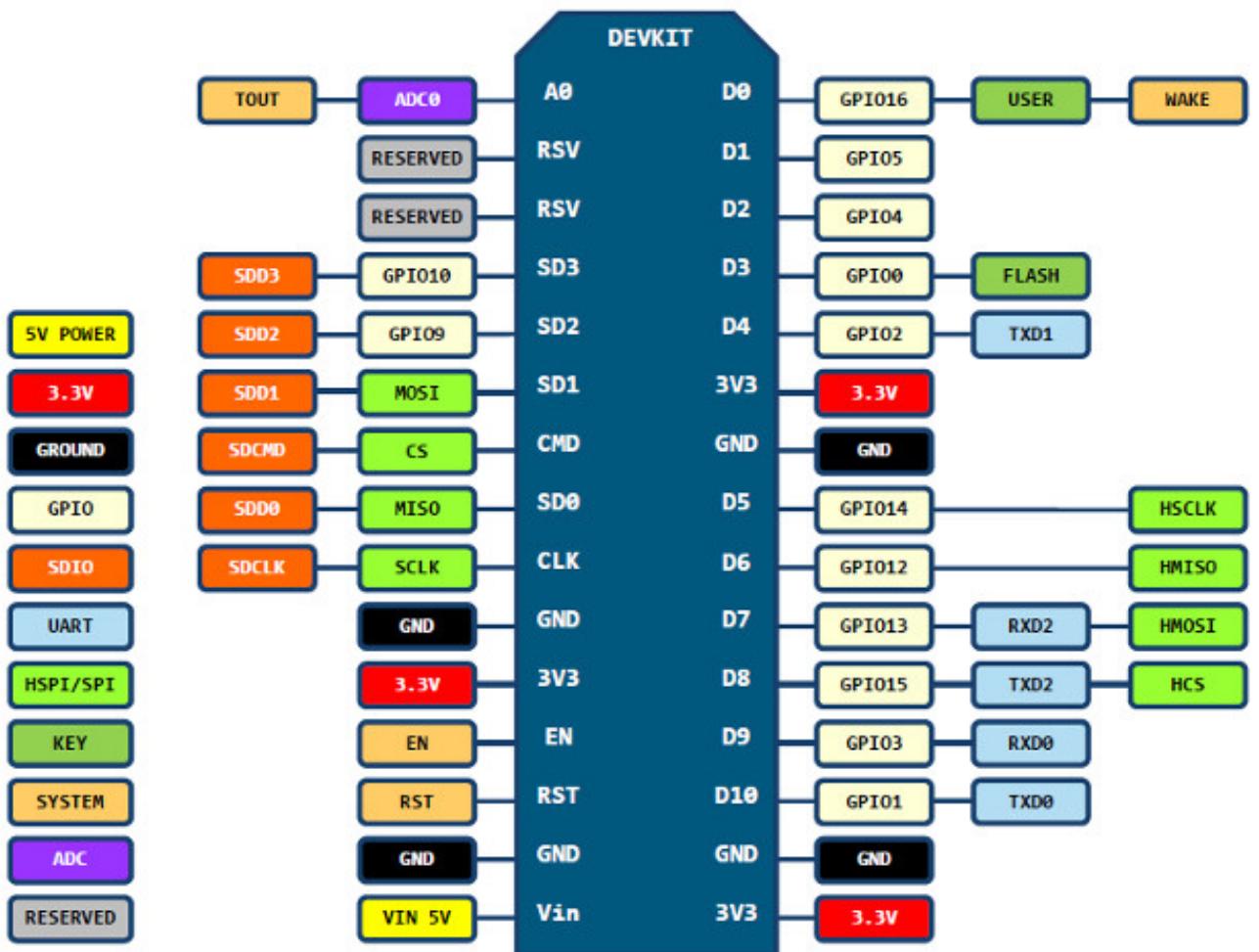
void loop() {
  // put your main code here, to run repeatedly:
}
```

Arduino development environment

Select Serial Port and Board







D0(GPIO16) can only be used as gpio read/write, no interrupt supported, no pwm/i2c/ow supported.

NodeMCU Pinout

Projects Overview

#1. Blink LED

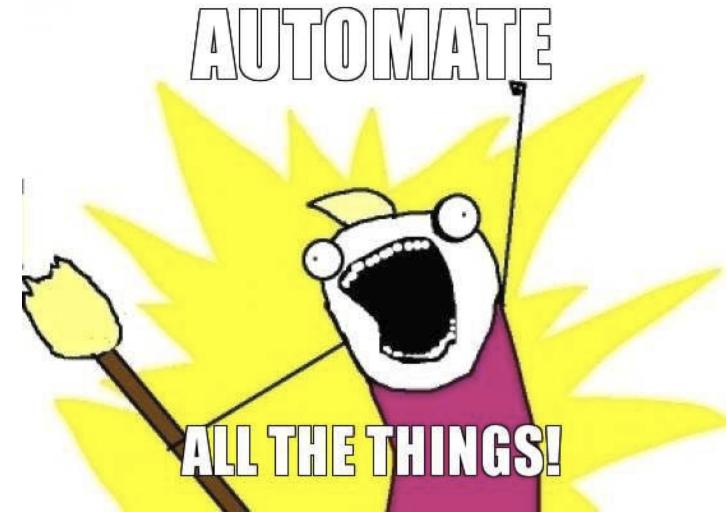
#2. Serial Communication

#3. Setting a simple HTTP web server

#4. Creating a WiFi / Web Controlled LED

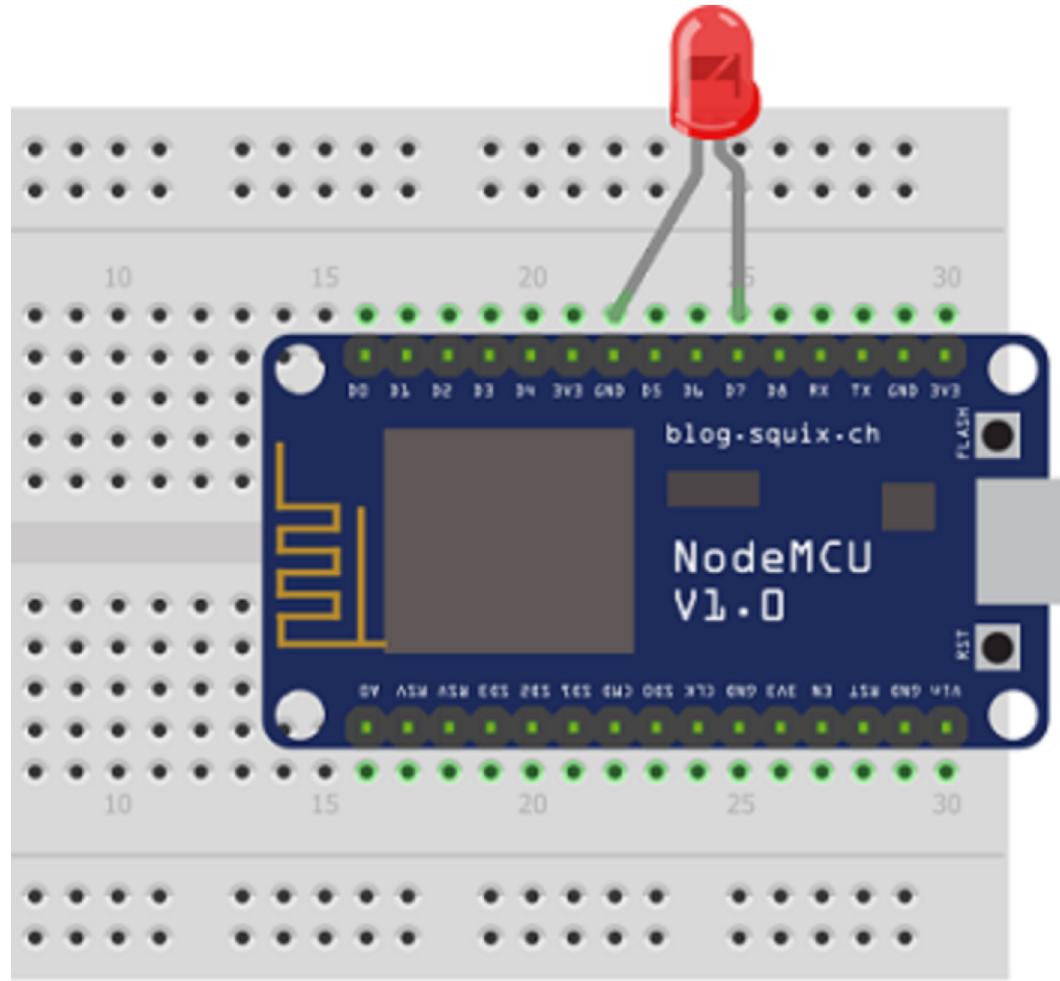
#5. Prototyping IoT with ESP8266 and IFTTT

```
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  
  delay(1000);                      // wait for a second  
  digitalWrite(LED_BUILTIN, LOW);     // turn the LED off  
  delay(1000);                      // wait for a second  
}
```



Our First Program

<https://github.com/SophiaBelkin/BelkinCodingBootcamp/blob/master/examples/blinkLED/BlinkLED.ino>



Circuit

Basic C/C++ Embedded programming terms

Variable: Variables are the parts of the program used to hold different values. Similar to X, Y, and other letters In algebra problems

Loop: Loops are repeating sections of code, that may stop if certain criteria are met

VCC/+ , GND/-: In electronics these are used to indicate the positive, and negative voltage and will show how current should flow through the electronics

Input / Output: The way that a computer receives or sends bits of information

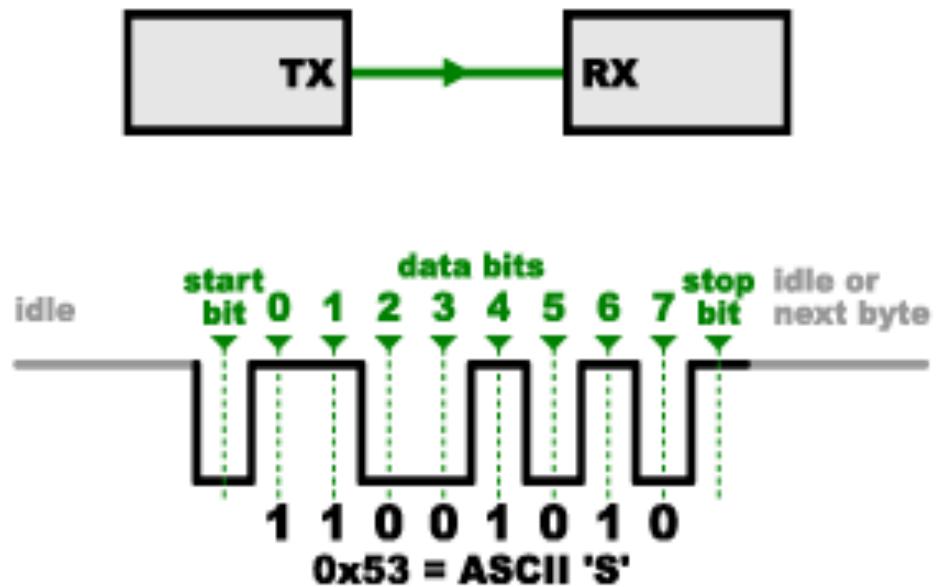
GPIO/IO pins: General purpose input output pins can be used as input or output devices, and can be turned on or off, or can detect if a connected signal is turned on or off

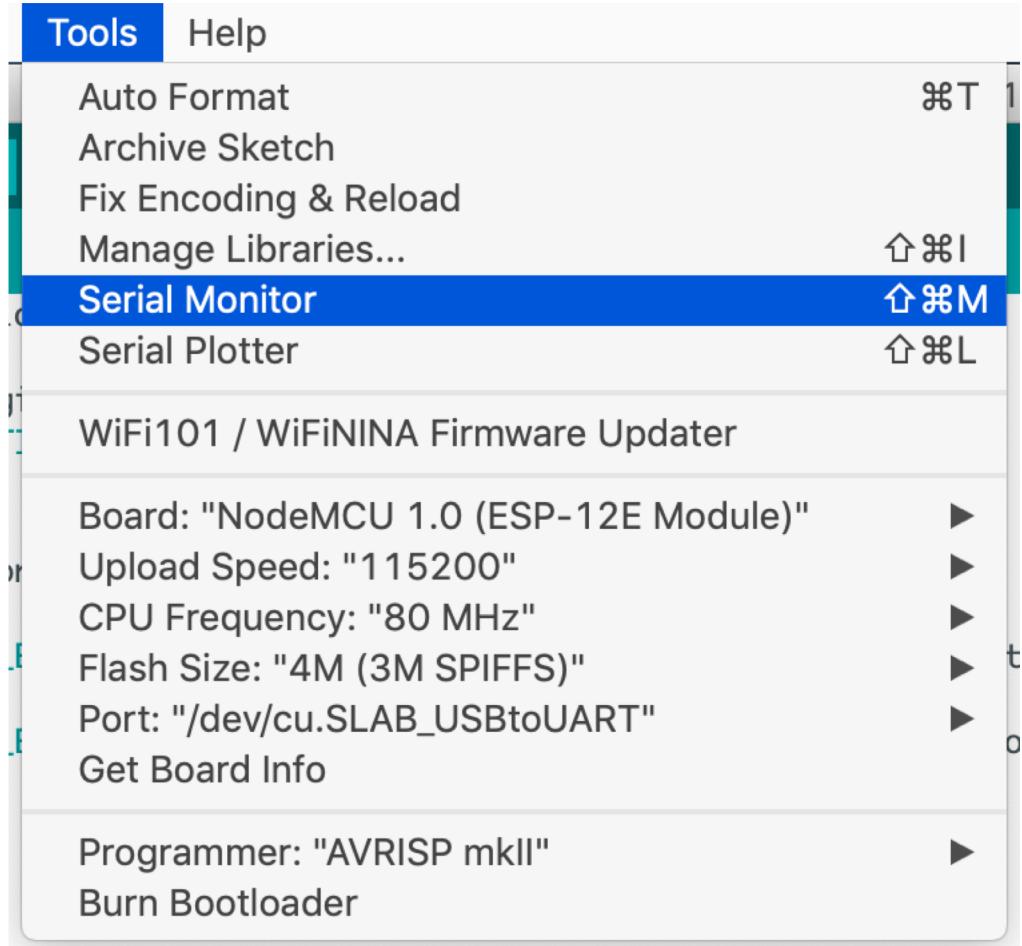
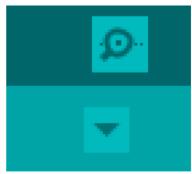
Function: A function is a defined set of instructions that can be reused in a program without the programmer repeating the same instructions

Library: A collection of functions typically designed to specialize at some specific task

#2 Serial Communication

- `Serial.begin(9600)`
- `Serial.print()` or `Serial.println()`



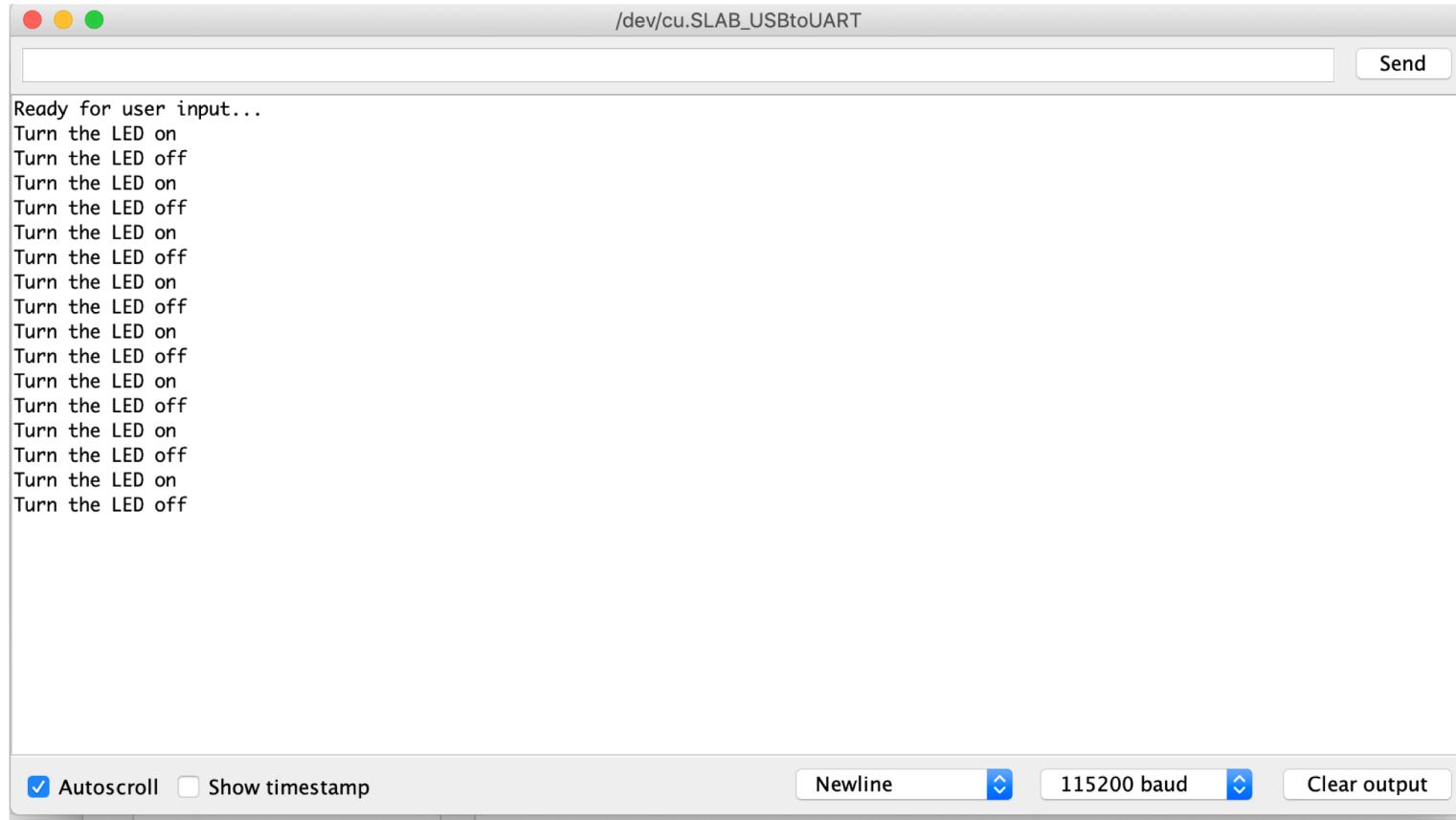


Serial Monitor

```
// 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);
    // start the serial port at 115200 Baud ( bps )
    Serial.begin(115200);
    Serial.println("Ready for user input...");
}

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

Serial Monitor

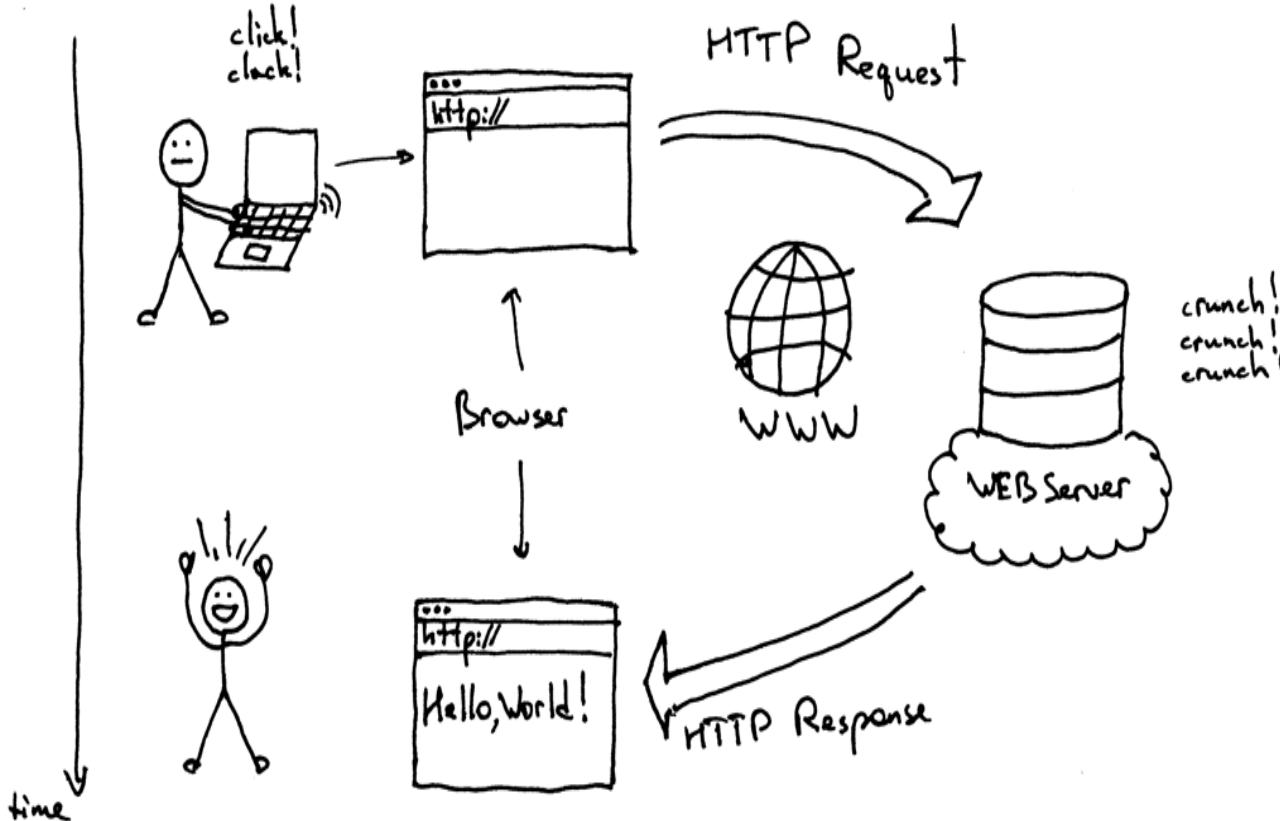


A screenshot of a Mac OS X-style terminal window titled "/dev/cu.SLAB_USBtoUART". The window contains the following text:

```
Ready for user input...
Turn the LED on
Turn the LED off
```

At the bottom of the window, there are several controls: "Autoscroll" (checked), "Show timestamp" (unchecked), "Newline" (dropdown menu), "115200 baud" (dropdown menu), and "Clear output".

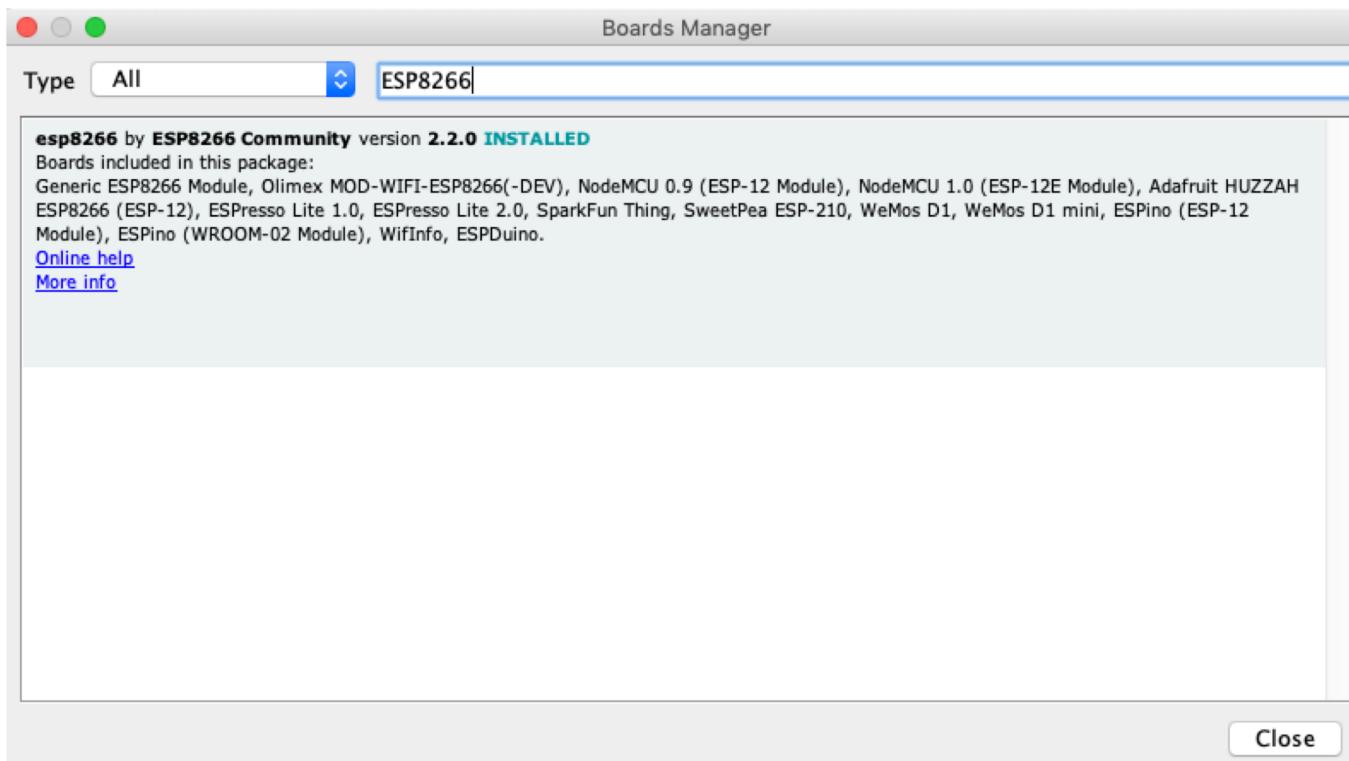
<https://github.com/SophiaBelkin/BelkinCodingBootcamp/blob/master/examples/CommunicationSerial/CommunicationSerial.ino>



How does the Internet work (Clients and Servers)

- **TCP/IP** – A set of protocols for sending and receiving data using packets and addresses
- **Client** – A program that ‘connects’ to servers to start communication
web browsers are a client program
- **Server** – A program that ‘accepts’ connections from client programs
web servers handle connections from a web browser

#3 Setting a simple HTTP web server



Step 1: Configure the Arduino IDE to support the ESP8266

<https://github.com/SophiaBelkin/BelkinCodingBootcamp/blob/master/examples/ESP8266Server/ESP8266Server.ino>

Step 2: Include ESP8266WiFi and ESP8266WebServer libraries

Step 3: Declare a global object variable from ESP8266WebServer class

```
//Include ESP8266 libraries
#include "ESP8266WiFi.h"
#include "ESP8266WebServer.h"

//Declare a global object variable from ESP8266WebServer class
ESP8266WebServer server(80);
```

Step 4: Connect to the WiFi network with ESP8266

Step 5 : Get local IP address

```
const char* ssid = "ssid";
const char* password = "password";

void setup() {

    Serial.begin(115200);
    WiFi.begin(ssid, password); //Connect to the WiFi network

    while (WiFi.status() != WL_CONNECTED) { //Wait for connection
        delay(500);
        Serial.println("Waiting to connect...");
    }

    Serial.print("IP address: ");
    Serial.println(WiFi.localIP()); //Print the local IP
}
```

<https://github.com/SophiaBelkin/BelkinCodingBootcamp/blob/master/examples/ESP8266Server/ESP8266Server.ino>

```
server.on("/", []() { //Define the handling function for the path
    server.send(200, "text/plain", "Hello world");
});

server.on("/other", handleRootPath);

server.begin(); //Start the server
Serial.println("Server listening");
}

void handleRootPath() { //Handler for the rooth path
    server.send(200, "text/plain", "another url");
}

void loop() {
    server.handleClient(); //Handling of incoming requests
}
```

Step 6: Specify the code to be execute when an HTTP request is performed on each path

Step 7: Handle the actual incoming of HTTP requests

#4 Creating a WiFi / Web Controlled LED

```
#include "ESP8266WiFi.h"
#include "ESP8266WebServer.h"

// Replace with your network credentials
String ssid = "coding-bootcamp";
String password = "belkin123";

ESP8266WebServer server(80); //instantiate server at port 80 (http port)

String page = ""; //Creates the HTML Page
int LEDPin = D8;
void setup(void){
    //the HTML of the web page
    page = "<h1>Simple NodeMCU Web Server</h1>
    "<p><a href=\"LEDOn\"><button>ON</button></a>\n"
    "<a href=\"LEDOff\"><button>OFF</button></a></p>";
```

Assign LED pin number and
create the HTML page

What is HTML?

- HTML stands for Hyper Text Markup Language
- It describes the structure of a webpage.
- HTML element are represented by tags

Simple NodeMCU Web Server

ON OFF

```
<!DOCTYPE html>
<html>
<body>
    <h1>Simple NodeMCU Web Server</h1>
    <p>
        <a href="LEDOn"><button>ON</button></a>
        <a href="LEDOff"><button>OFF</button></a>
    </p>
</body>
</html>
```

```
//make the LED pin output and initially turned off
pinMode(LEDPin, OUTPUT);
digitalWrite(LEDPin, LOW);

Serial.begin(115200);
WiFi.begin(ssid, password); //begin WiFi access point

// Wait for connection
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.println("Waiting to connect...");
}
```

```
Serial.println("IP address: ");
Serial.println(WiFi.localIP());

server.on("/", [](){
    server.send(200, "text/html", page);
});

server.on("/LEDOff", [](){
    server.send(200, "text/html", page);
    digitalWrite(LEDPin, LOW);

});

server.on("/LEDOn", [](){
    server.send(200, "text/html", page);
    digitalWrite(LEDPin, HIGH);

});

});
```

Specify which code to execute when an HTTP request is performed to each path.

```
server.begin();
Serial.println("Web server started!");
}

void loop(void){
    server.handleClient();
}
```

Start our server and handle
the incoming http request

#5 Prototyping IoT with ESP8266 and IFTTT

IFTTT

- Free platform that lets you create connection between different web-service and applications
- Stands for ‘If this, then that’
- A combination of a TRIGGER and an ACTION
- Examples:
 - If the sensor is triggered, send me a message
 - If the garage door is opened, turn the living room lights on

Step 1 Create an account on <https://ifttt.com/>

Step 2 Create an IFTTT Webhook

Triggers

Receive a web request

This trigger fires every time the Maker service receives a web request to notify it of an event. For information on triggering events, go to your Maker service settings and then the listed URL (web) or tap your username (mobile)

Trigger Fields

Event Name



If Maker Event
"pushbutton",
then Send me an
SMS at phone
number

by sophia79lx

On

- Created on Apr 09 2019
- Never run

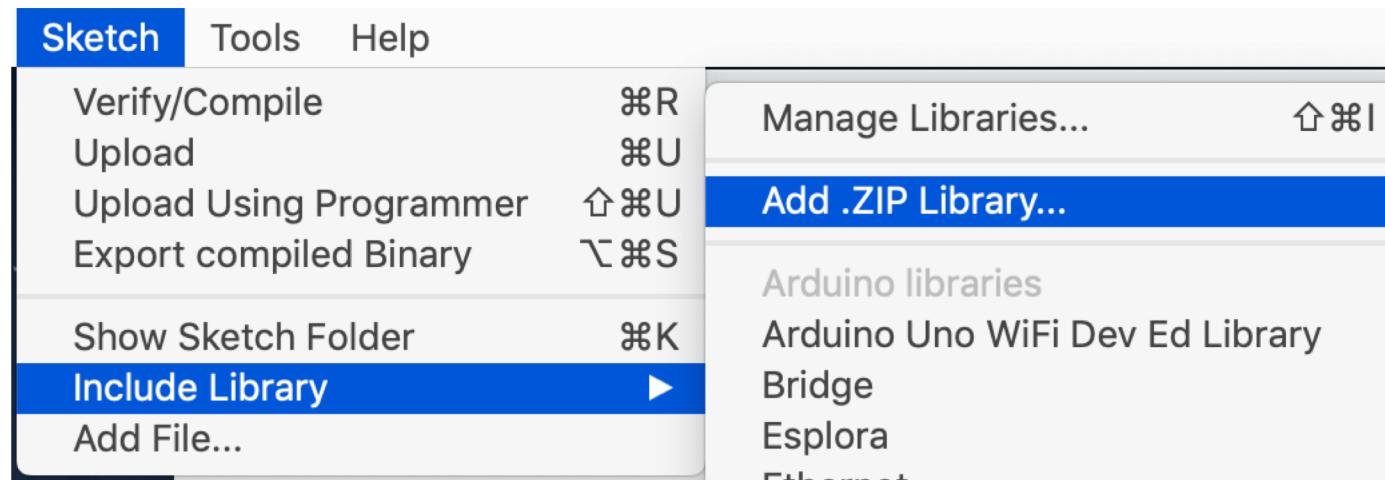
This Applet usually
runs within a few
seconds

Check now

works with

Step 4: Add **ESP8266IFTTT** to your Arduino libraries

Download the **ESP8266IFTTT** library from <https://github.com/SophiaBelkin/ESP8266IFTTT>



```
#include "ESP8266IFTTT.h"
#include "ESP8266WiFi.h"

#define ssid "coding-bootcamp"
#define password "belkin123"
#define IFTTT_API_KEY "czXVUEU4zqK"
#define IFTTT_EVENT_NAME "button_pushed"

int ledPin = D8;
void setup() {

    pinMode(ledPin, OUTPUT); // initialize the LED pin as an output:
    Serial.begin(115200);
    // Wait for connection
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.println("Waiting to connect...");
    }
}
```

Step 5 Connect to Wi-Fi

Step 6 Set up IFTTT integration

```
Serial.println("IP address: ");
Serial.println(WiFi.localIP());  
  
IFTTT.trigger(IFTTT_EVENT_NAME, IFTTT_API_KEY);
digitalWrite(ledPin, HIGH);
delay(200);
digitalWrite(ledPin, LOW);
}  
  
void loop() {
}
```

Glossary

- Analog: An input or output that can send or receive a range of values
- Digital: An input our output that can send or receive a value of 0 or 1 (LOW, and HIGH)
- Function: A defined section of code that executes a specific task.
- Ground: The point of a circuit where there is 0 potential electrical energy. Without a ground, electricity will not have a place to flow in a circuit.
- Serial monitor : A tool used to communicate with the Arduino board via the USB Serial port.

Links

<https://www.arduino.cc/en/Main/Software> - The official Arduino IDE

<https://www.makerspaces.com/arduino-uno-tutorial-beginners/> - Arduino Beginner tutorial(s)

<https://www.makeuseof.com/tag/getting-started-with-arduino-a-beginners-guide/>

<https://www.arduino.cc/en/tutorial/links> - More Arduino programming links

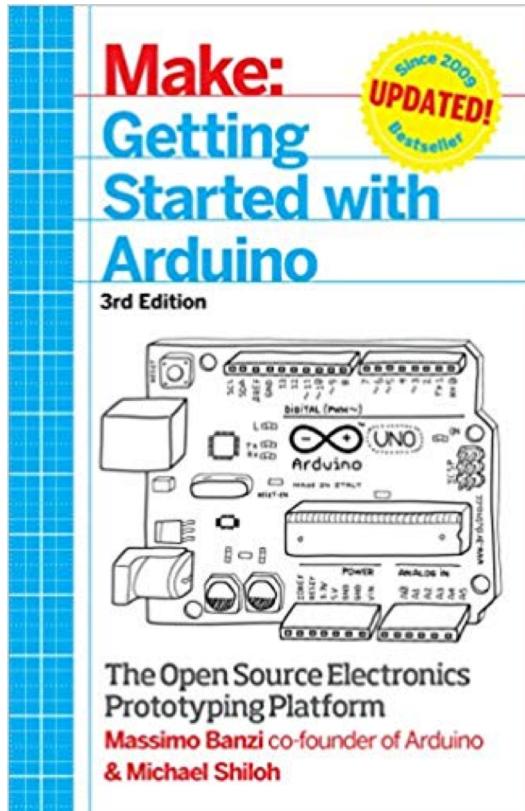
<http://www.oddwires.com/arduino-project-ideas/> - Arduino Project Ideas

<https://www.hackster.io/Aritro/getting-started-with-esp-nodemcu-using-arduinoide-aa7267>

<https://github.com/SophiaBelkin/BelkinCodingBootcamp>

Further Reading

For Absolute Beginners



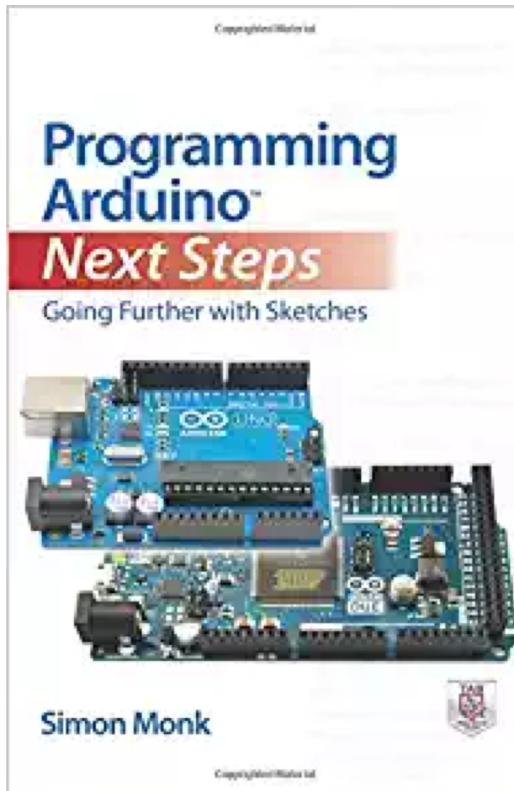
Getting Started with Arduino: The Open Source Electronics Prototyping Platform

by [Massimo Banzi](#) , [Michael Shiloh](#)

This is a beginner friendly book written based on the platforms's cofounder Massimo Banzi's expensive experience in teaching, using and building Arduinio

Further Reading

Take your skill to the next level



Programming Arduino Next Steps: Going Further with Sketches

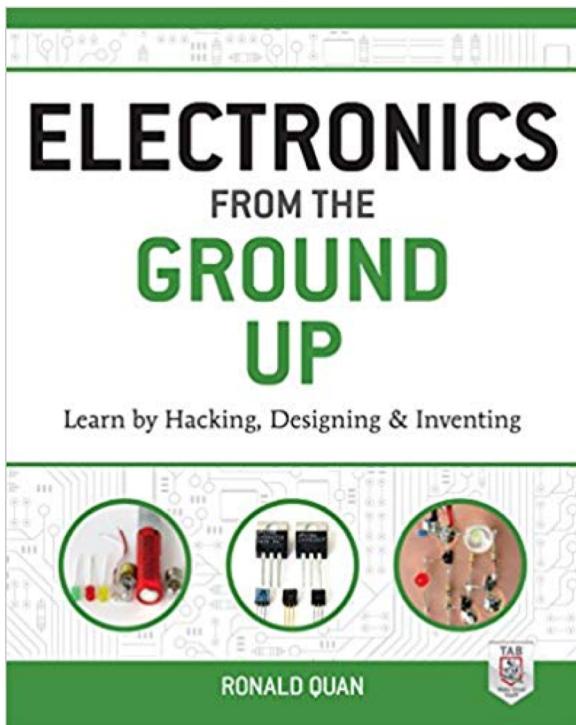
by [Simon Monk](#)

Learn advanced Arduino programming techniques, including how to:

- Program Arduino for the Internet
- Maximizing serial communications
- Managing memory
- Performing digital signal processing
- Create and release your own code library

Further Reading

Take your skill to the next level



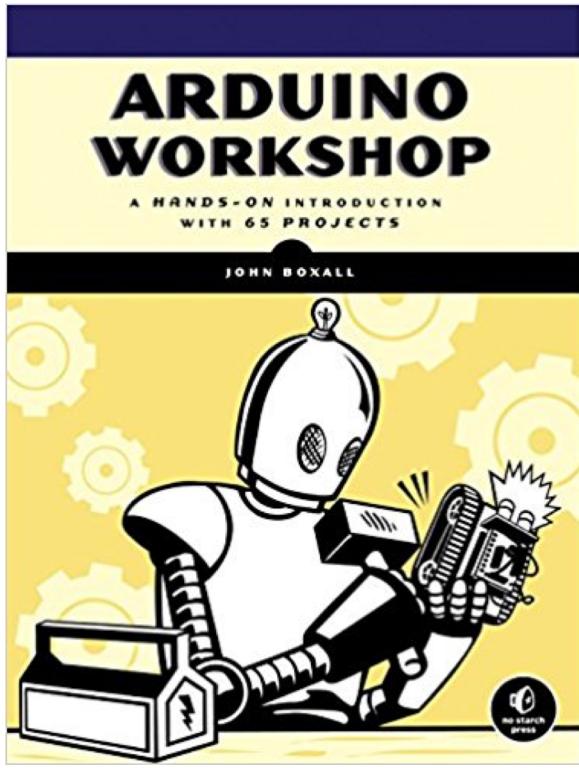
Electronics from the Ground Up: Learn by Hacking, Designing, and Inventing

by [Ronald Quan](#)

Guides you through step-by-step experiments that reveal how electronic circuits function.

Further Reading

Fun project ideas



Arduino Workshop: A Hands-On Introduction with 65 Projects

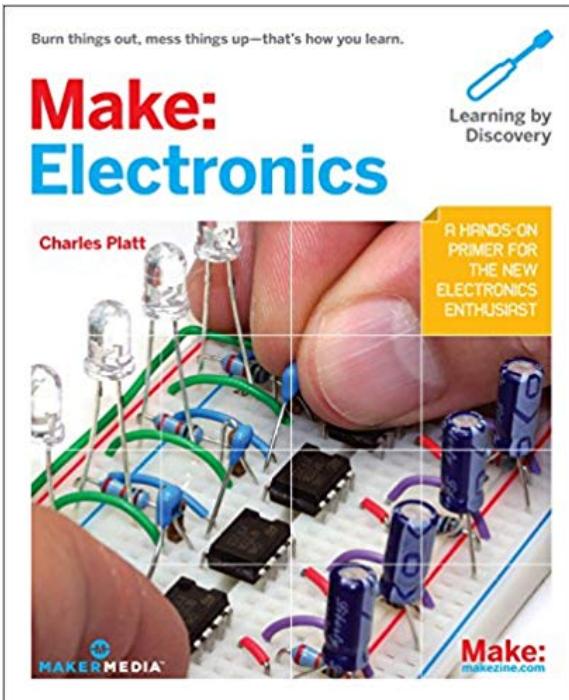
by [John Boxall](#)

Among the book's 65 projects are useful devices like:

- A digital thermometer that charts temperature changes on an LCD
- A GPS logger that records data from your travels, which can be displayed on Google Maps
- A handy tester that lets you check the voltage of any single-cell battery
- A keypad-controlled lock that requires a secret code to open

Further Reading

All about circuits and electrical components



Make: Electronics

by **Charles Platt**

"Burn things out, mess things up—that's how you learn".

- Learn about common electronic components
- Circuit design, and creation
- How to use a multimeter
- Learn how to use the most popular IC's in projects



THANK YOU