

Arduino <--> Node Server

Adam Driggers
2025-26 Resident
NYU IMA Low Res

Check-In

Which cat are you vibing with?



Goal

Use an Arduino to send and receive data to a Node web Server.

Agenda

- Background 20 min
- Setup 10 min
- Getting Data 10 min
- Pushing Data 10 min
- Q/A 10+ min

**What are Arduinos good
for?**

**What are Web servers
good for?**

**What can we do when we
combine the two?**

Applications

- Collect Sensor Data
- Remote control of Lights/motors
- Communication between multiple Arduinos
- Much more...

Today

Two simple examples!

1. Turn an LED on/off based on a boolean value in a database.
2. Send a light sensor reading to a web server.

Prediction

NO Googling or LLMing

Make a prediction of what we need to do to make the Arduino communicate with a webserver.

Create a diagram, flow chart, list, or pseudo code.

Break the problem down into smaller and smaller steps.

< 10 min

Share Out

≈ 10 min

Client

Server

Client <--> Server

Types of Requests ("HTTP Verbs")

- GET – ask for some data
- POST – send some data
- PUT – replace some existing data
- DELETE – delete some data
- Some others...

So we need...

- Server with routes
- Arduino connected to the internet
- A way for the Arduino to make requests of the server
- Server processes these requests
- Server responds to the Arduino
- Arduino does something with the response.

So we need...

- Server with routes
- Arduino connected to the internet
- A way for the Arduino to make requests of the server
- Server processes these requests
- Server responds to the Arduino
- Arduino does something with the response.

Send/receive JSON Data!

Server

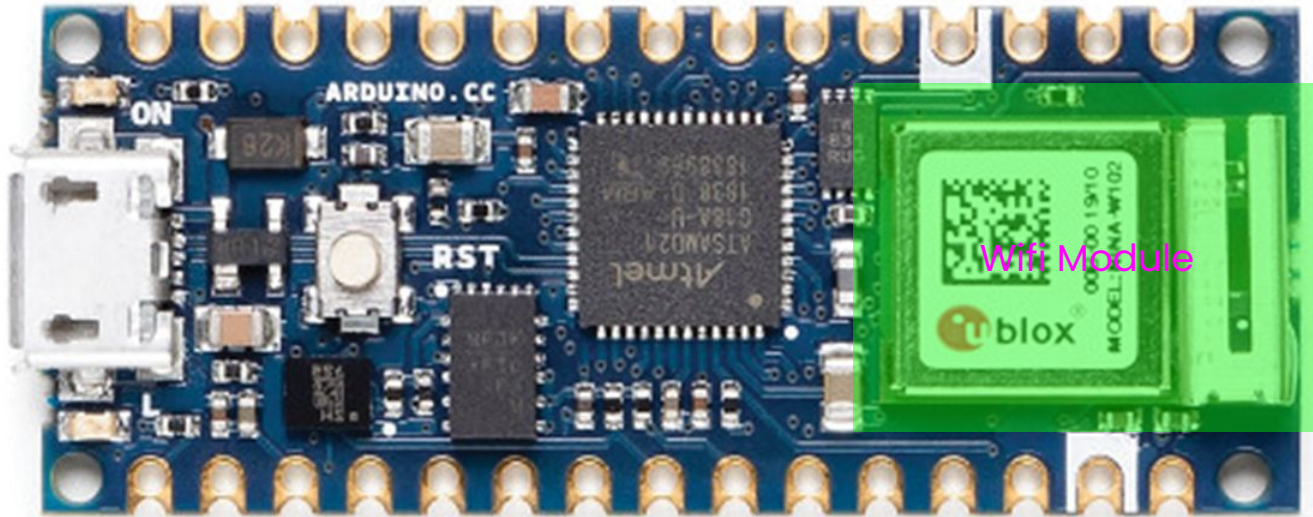
Read through `server.js` file. What do you see?

5 min

So we need...

- Server with routes
- Arduino connected to the internet
- A way for the Arduino to make requests of the server
- Server processes these requests
- Server responds to the Arduino
- Arduino does something with the response.

Arduino Nano IOT



WifiNINA Library

Utility for connecting to wifi

Connect to Wifi

Run code in `wifi-tester`

So we need...

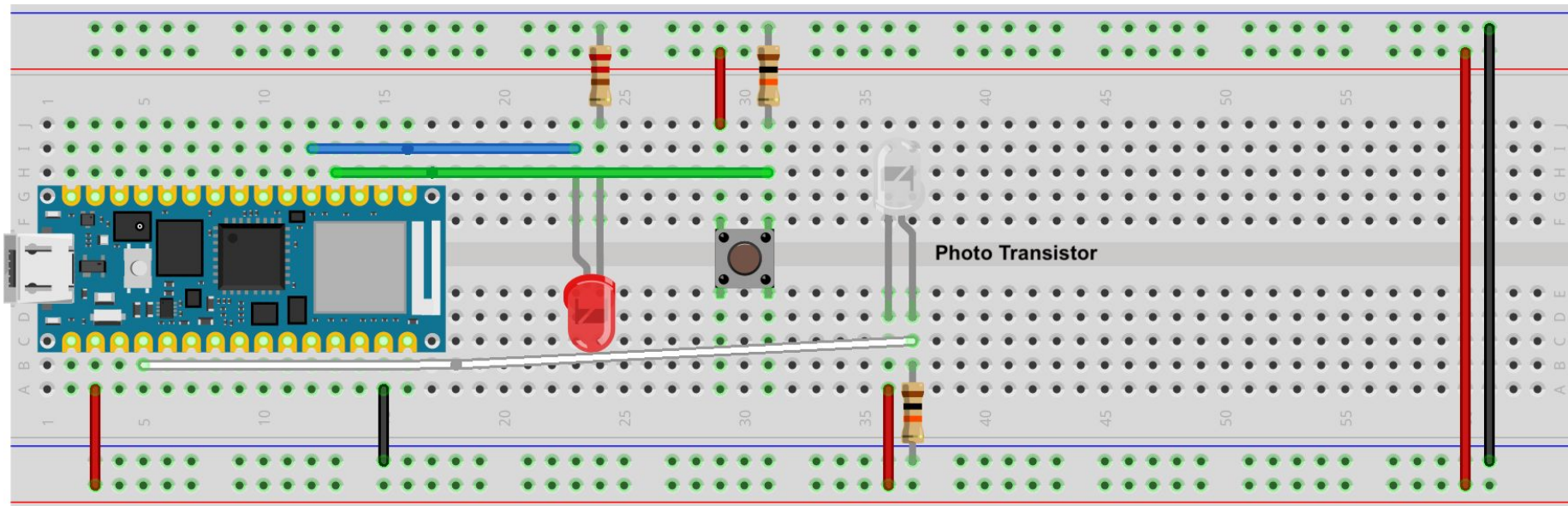
- Server with routes
- Arduino connected to the internet
- A way for the Arduino to make requests of the server
- Server processes these requests
- Server responds to the Arduino
- Arduino does something with the response.

WifiNINA Library

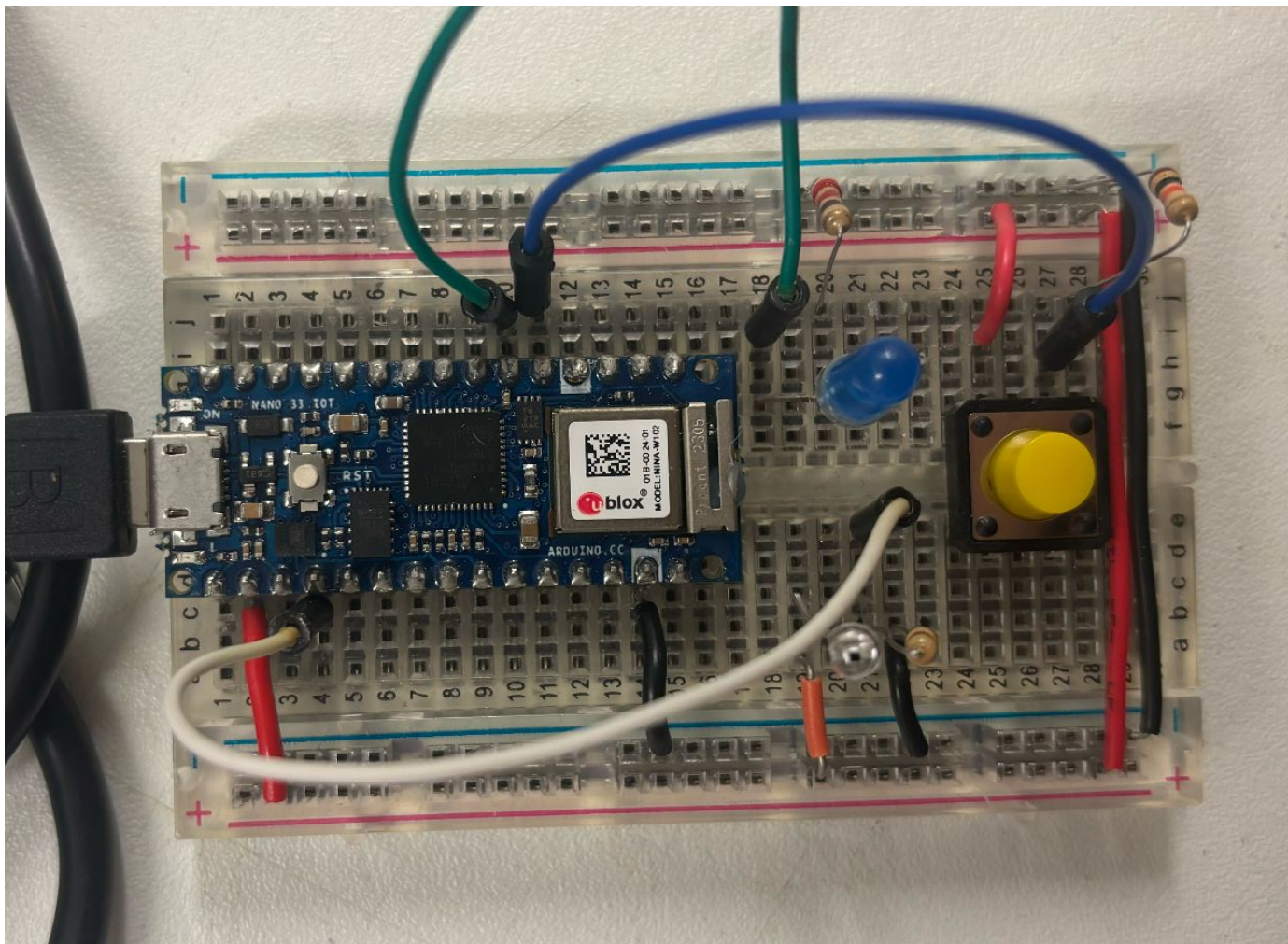
Also includes a client library for making HTTP requests

Sidequest

Before we continue, let's make sure
our circuit is working



fritzing



Upload and Test

Upload the code to the Arduino and open the serial monitor

Press the button

- Led turns on
- Light level is printed to the monitor

Cover up the transistor or shine a bright light on it and press the button

- Light level reading should change. Expect numbers from 0 to 4000. 0 is absolutely no light, 4000 super intense light. If you are getting values in a different range (i.e. between 0 - 300) try flipping the phototransistor.

WifiNINA Library

Also includes a client library for making HTTP requests

So we need...

- Server with routes
- Arduino connected to the internet
- A way for the Arduino to make requests of the server
- Server processes these requests
- Server responds to the Arduino
- Arduino does something with the response.

GET Request

Arduino gets data from the server

Turn on a light!

≈ 10 min

POST Request

Arduino posts data to the server

Send light sensor data

≈ 10 min

Note:

Both these examples use "polling," every once in a while the client is asking the server "any updates for me?"

This is inefficient but really simple and great for low-traffic scenarios

A better solution would be to use **websockets**, which is a little more difficult to set up since Arduino can't use Sockets.io

Questions?