GREATER PHILADELPHIA

aws

— USER GROUP —

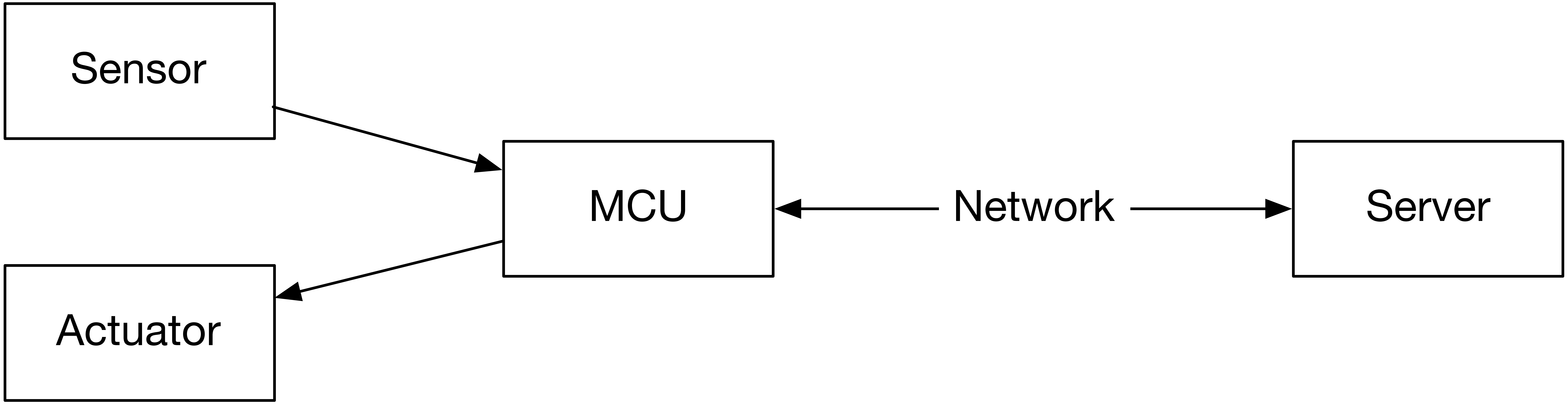
# AWS IoT Core

Connecting Devices to the Cloud

Don Coleman
don@chariotsolutions.com

CHARIOT
SOLUTIONS

# What is IoT?



Sensor

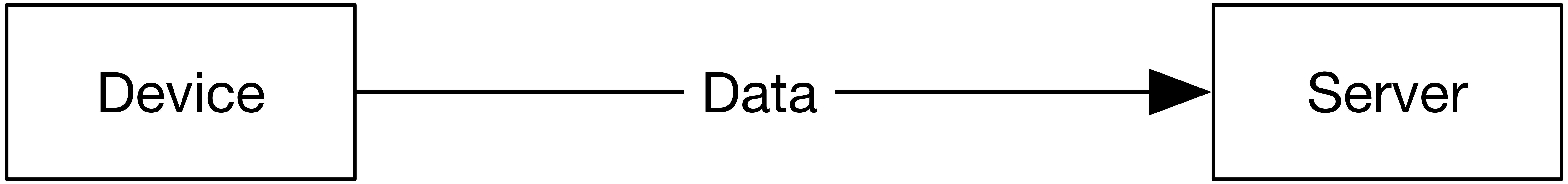
Actuator

MCU

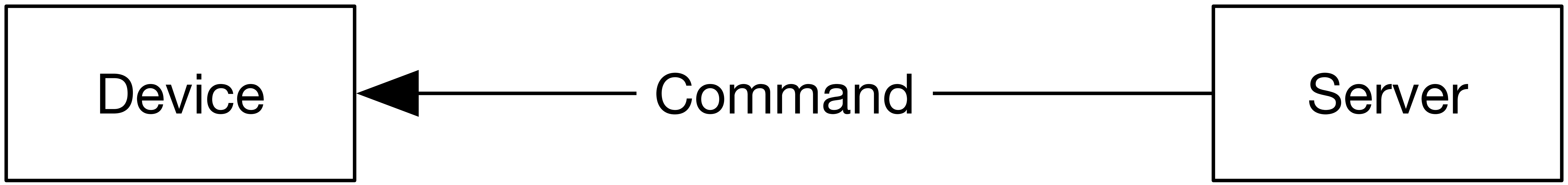
Network

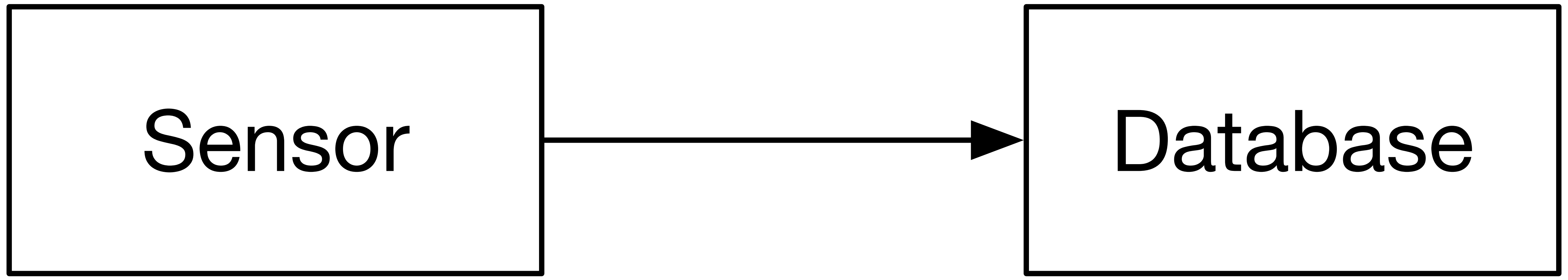
Server

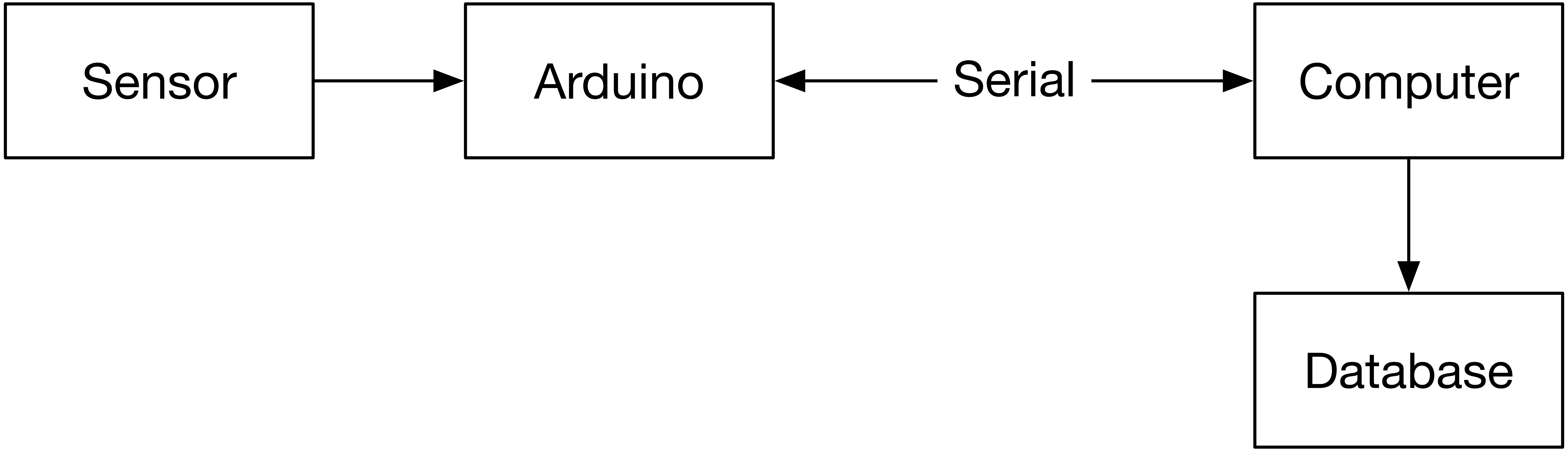
# Send sensor data to server



# Send commands to a device




Sensor
Database



Sensor

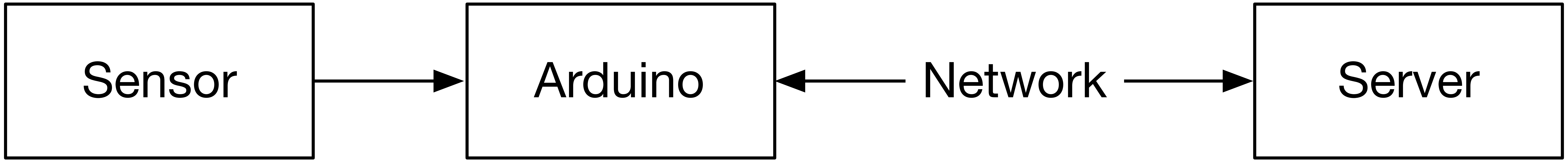
Arduino

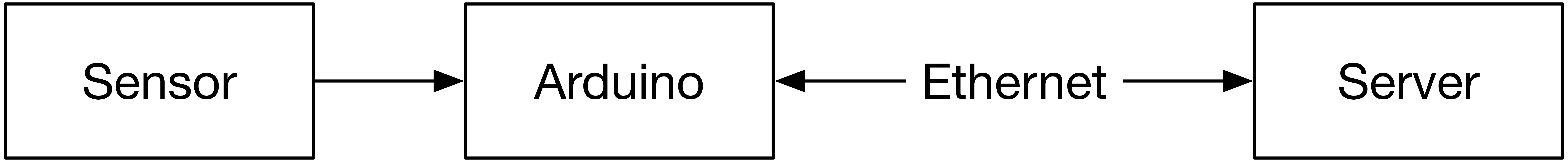
Serial

Computer

Database

# Transports


Sensor
Arduino
Network
Server

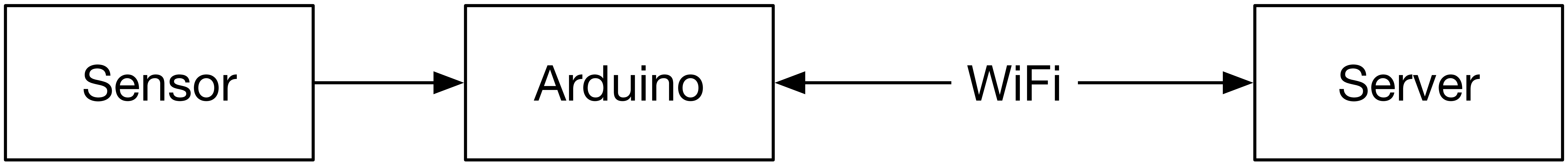


Sensor

Arduino

Ethernet

Server

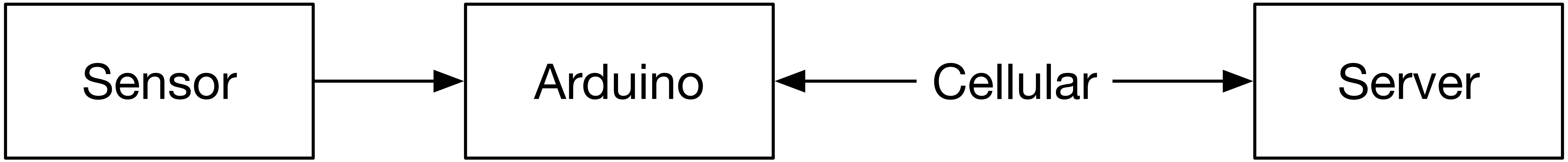


Sensor

Arduino

WiFi

Server

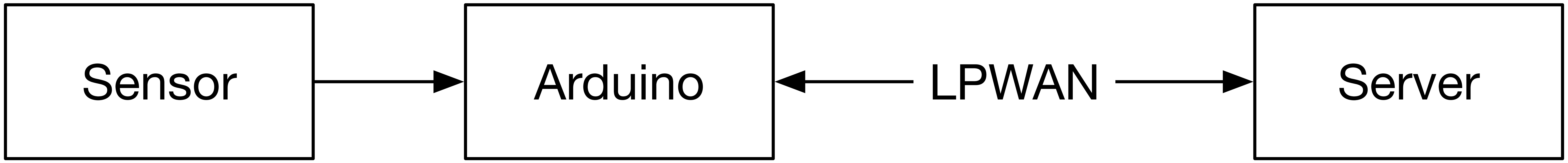


Sensor

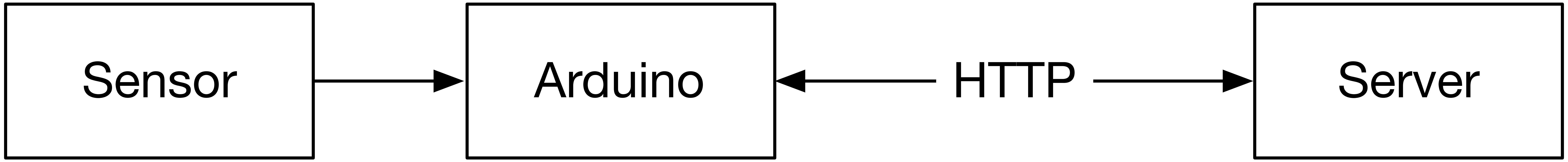
Arduino

Cellular

Server


Sensor
Arduino
LPWAN
Server

# Protocols

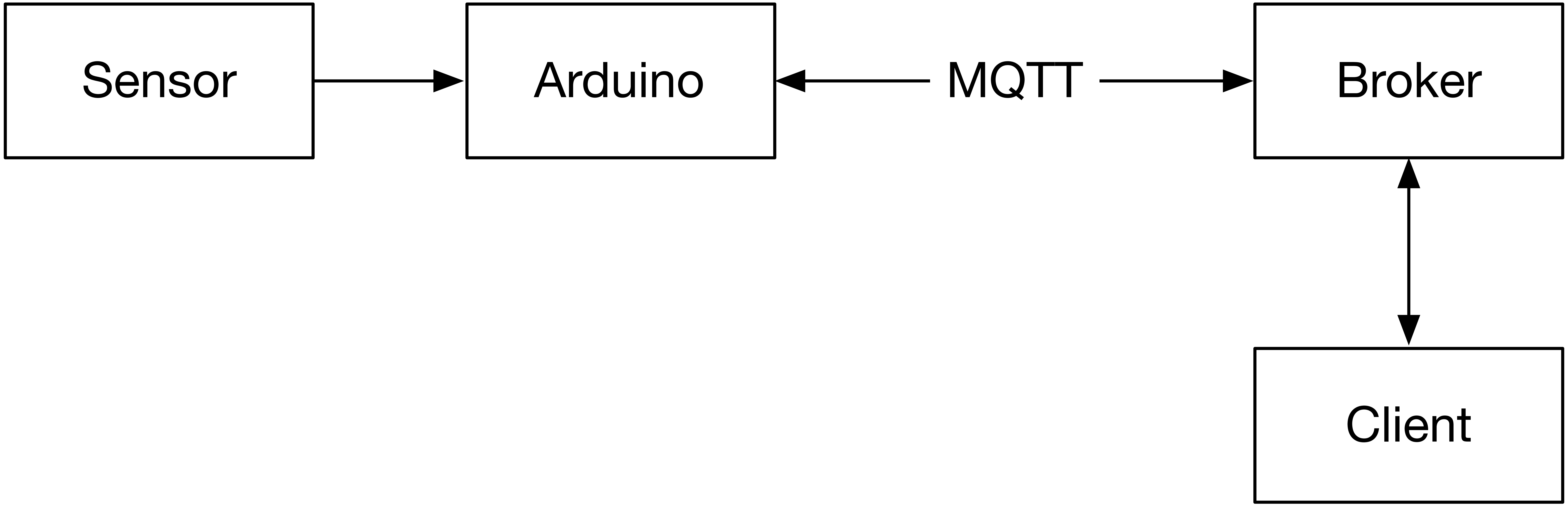


Sensor

Arduino

HTTP

Server



Sensor

Arduino

MQTT

Broker

Client

location/device/sensor

demo/device_01/temperature

demo/device_01/humidity

# Write to a topic

72.3

```
{
    temperature : 72.3
    humidity : 46
}
```

# Subscribe to a topic

demo/device_01/temperature

# Wildcards

#

+

# #

The # wildcard matches anything

# +

The plus wildcard can substitute one part of a path demo/device_01/+

demo/+/temperature

# AWS IoT Core

# Devices Need Certificates

# Policies

AWS IoT > Security > Policies > AllowAllPolicy
Info
Edit active version
Delete
Versions
Targets
Noncompliance
Tags
Info
Builder
JSON

# AllowAllPolicy

## Details

| Policy ARN | Active version | Created | Last updated |
|---|---|---|---|
| arn:aws:iot:us-east-1:176627085118:policy/AllowAllPolicy | 1 | January 28, 2025, 16:22:20 (UTC-05:00) | January 28, 2025, 16:22:20 (UTC-05:00) |

## Active version: 1

| Policy effect | Policy action | Policy resource |
|---|---|---|
| Allow | iot:* | * |

AWS IoT > Security > Policies > ThingPolicy

# ThingPolicy Info

Edit active version
Delete

## Details

Policy ARN
arn:aws:iot:us-east-1:176627085118:policy/ThingPolicy

Active version
1

Created
January 28, 2025, 16:22:20 (UTC-05:00)

Last updated
January 28, 2025, 16:22:20 (UTC-05:00)

Versions
Targets
Noncompliance
Tags

## Active version: 1 Info

Builder
JSON

| Policy effect | Policy action | Policy resource |
|---|---|---|
| Allow | iot:Connect | arn:aws:iot:us-east-1:176627085118:client/${iot:Certificate.Subject.CommonName} |
| Allow | iot:Publish | arn:aws:iot:us-east-1:176627085118:topic/things/${iot:ClientId}/* |
| Allow | iot:Receive | arn:aws:iot:us-east-1:176627085118:topic/things/${iot:ClientId}/* |
| Allow | iot:Subscribe | arn:aws:iot:us-east-1:176627085118:topicfilter/things/${iot:ClientId}/* |

# Rules

> Create rule

# Configure SQL statement Info

Add a simplified SQL syntax to filter messages received on an MQTT topic and push the data elsewhere.

## SQL statement Info

SQL version

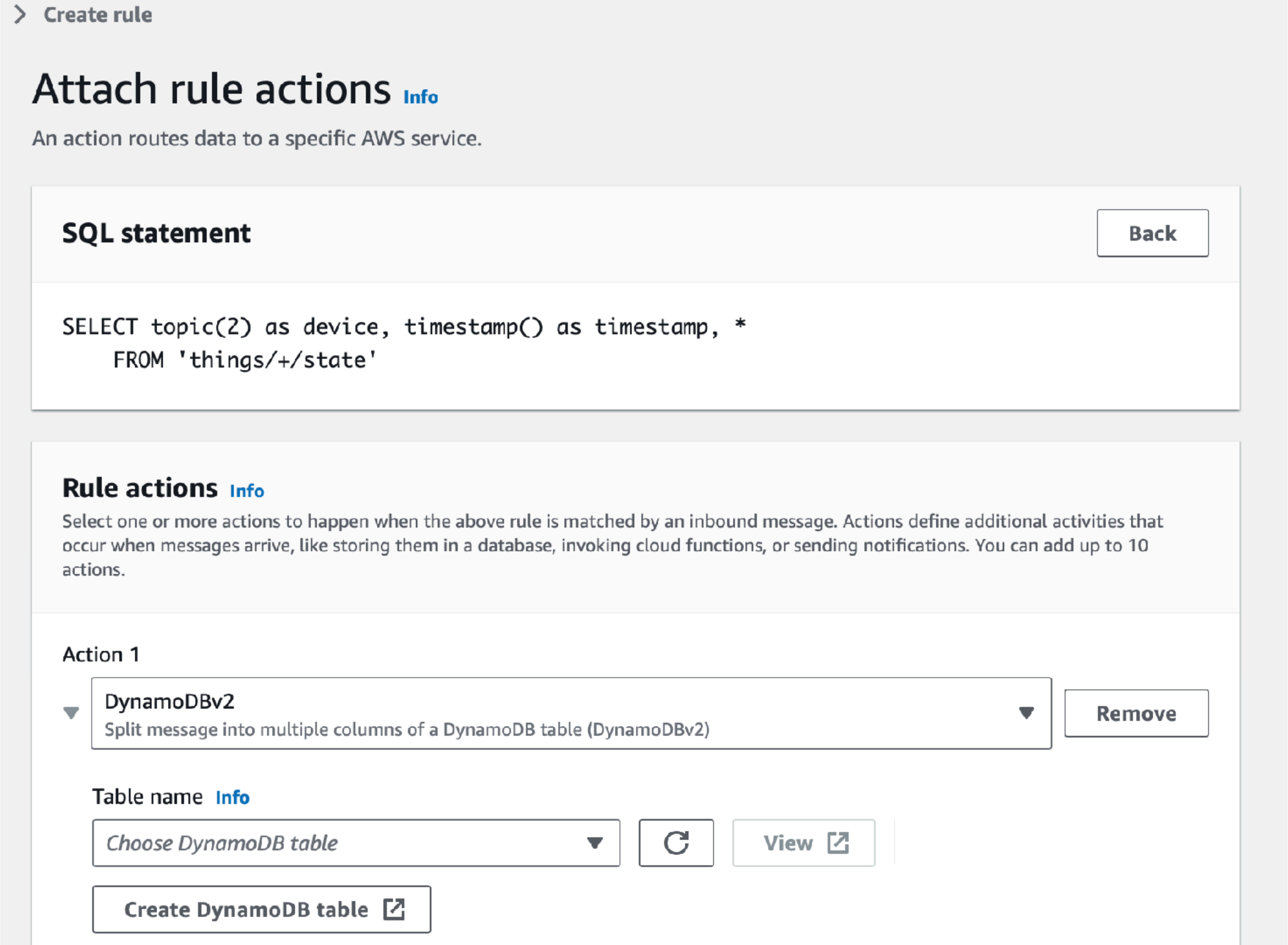
The version of the SQL rules engine to use when evaluating the rule.

2016-03-23 ▼

SQL statement

Enter a SQL statement using the following: SELECT <Attribute> FROM <Topic Filter> WHERE <Condition>. For example: SELECT temperature FROM 'iot/topic' WHERE temperature > 50. To learn more, see AWS IoT SQL Reference.

```
SELECT topic(2) as device, timestamp() as timestamp, *
    FROM 'things/+/state'
```



## Attach rule actions Info

An action routes data to a specific AWS service.

### SQL statement

```
SELECT topic(2) as device, timestamp() as timestamp, *
    FROM 'things/+/state'
```

### Rule actions Info

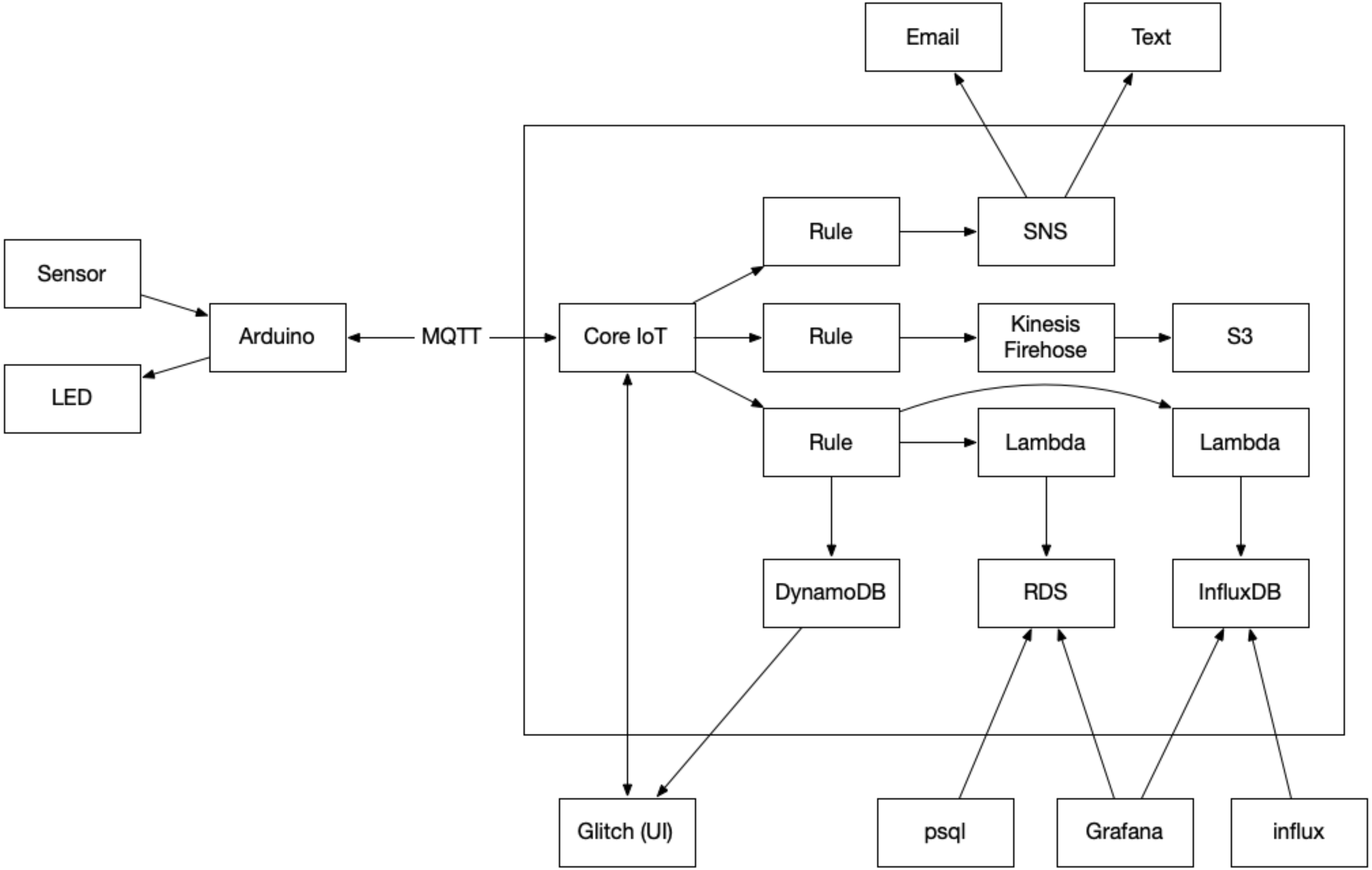
Select one or more actions to happen when the above rule is matched by an inbound message. Actions define additional activities that occur when messages arrive, like storing them in a database, invoking cloud functions, or sending notifications. You can add up to 10 actions.

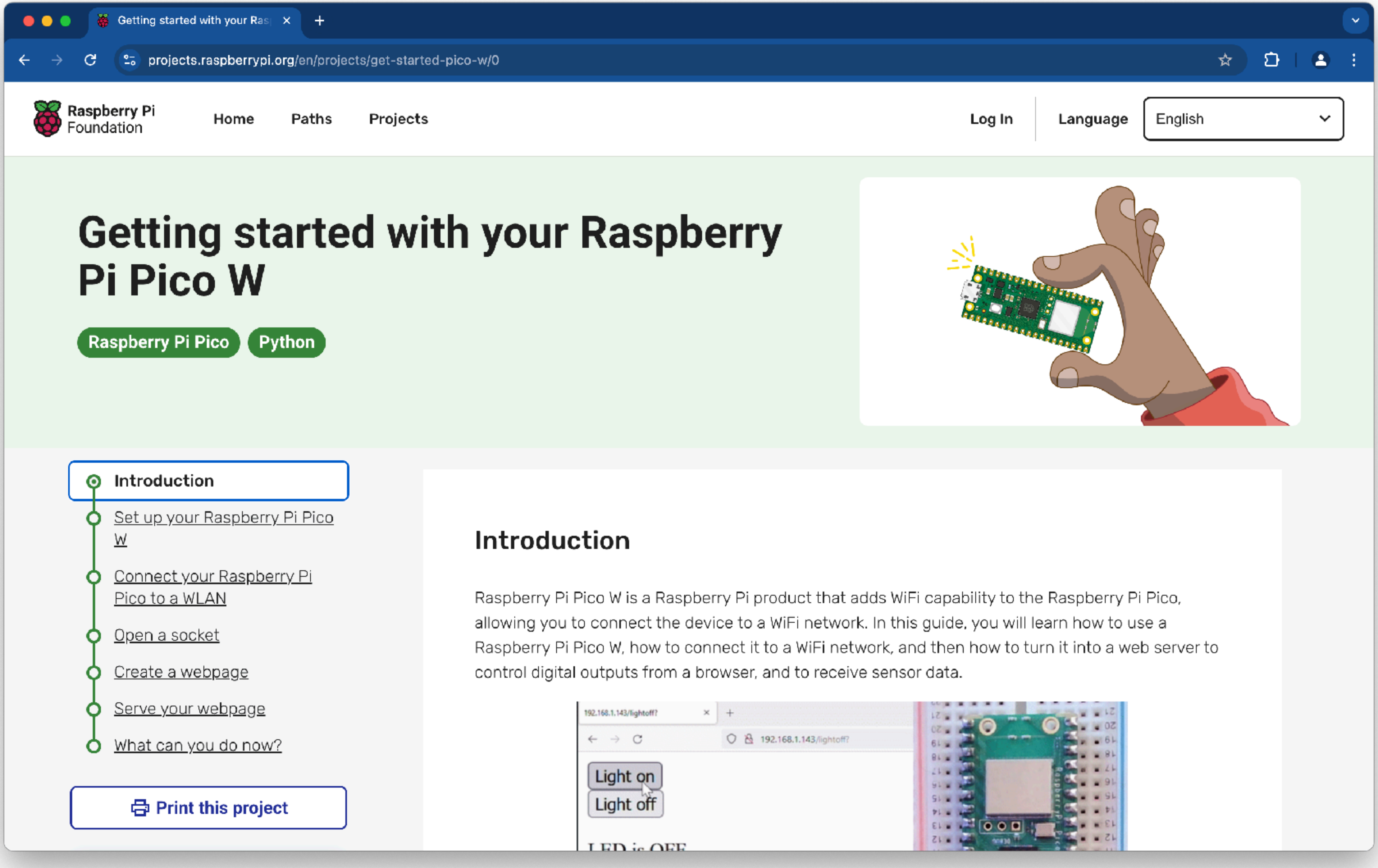
Action 1

DynamoDBv2
Split message into multiple columns of a DynamoDB table (DynamoDBv2)

Table name Info

*Choose DynamoDB table*


Email
Text
Rule
SNS
Sensor
Arduino
MQTT
Core IoT
Rule
Kinesis Firehose
S3
LED
Rule
Lambda
Lambda
DynamoDB
RDS
InfluxDB
Glitch (UI)
psql
Grafana
influx



Getting started with your Ras
projects.raspberrypi.org/en/projects/get-started-pico-w/0
Raspberry Pi Foundation
Home
Paths
Projects
Log In
Language
English

# Getting started with your Raspberry Pi Pico W

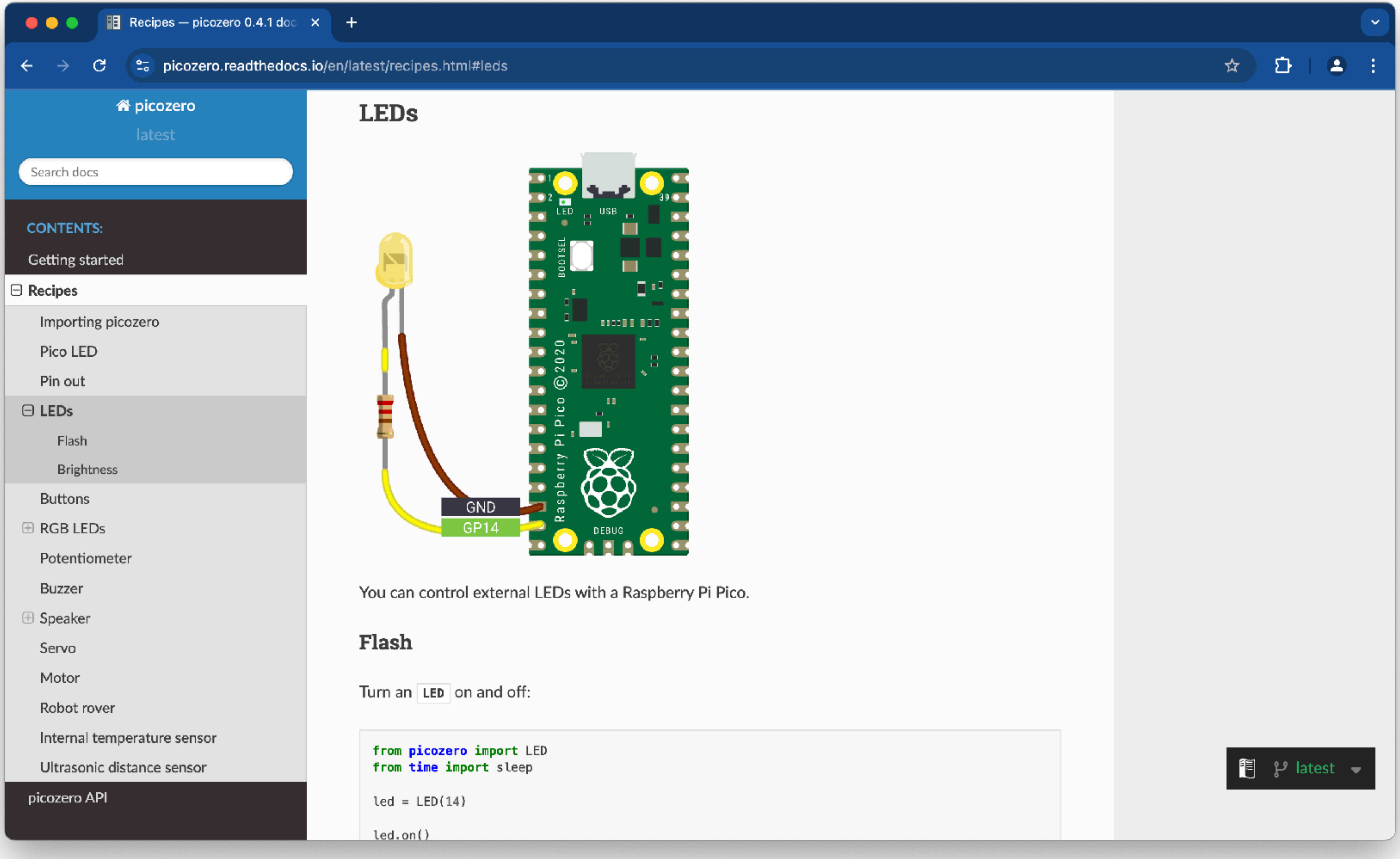
Raspberry Pi Pico  Python

- Introduction
- Set up your Raspberry Pi Pico W
- Connect your Raspberry Pi Pico to a WLAN
- Open a socket
- Create a webpage
- Serve your webpage
- What can you do now?

Print this project

## Introduction

Raspberry Pi Pico W is a Raspberry Pi product that adds WiFi capability to the Raspberry Pi Pico, allowing you to connect the device to a WiFi network. In this guide, you will learn how to use a Raspberry Pi Pico W, how to connect it to a WiFi network, and then how to turn it into a web server to control digital outputs from a browser, and to receive sensor data.



## LEDs

You can control external LEDs with a Raspberry Pi Pico.

### Flash

Turn an `LED` on and off:

```
from picozero import LED
from time import sleep

led = LED(14)

led.on()
```

Raspberry Pi Pico W
2022
BOOTSEL
DEBUG
+
-
60
55
50
45
40
35
30
25
20
15

# things/device_01/button

sends **pressed** or **released**

**things/device_01/temperature**

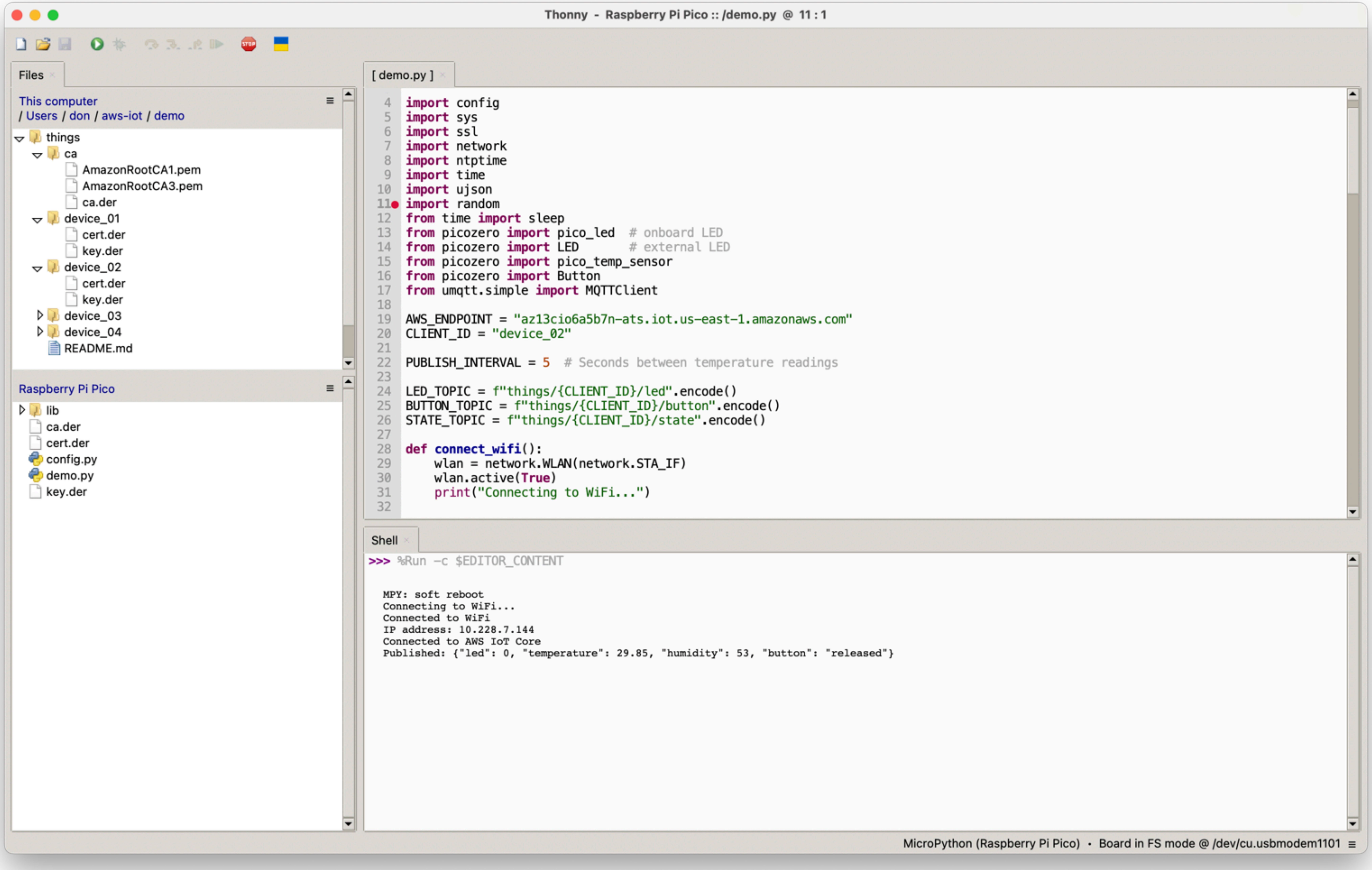
72

# things/device_01/state

```
{
    temperature: 72,
    humidity: 52,
    led: 0,
    button: released,
}
```

# things/device_01/led

0 for off, 1 for on



```python
import config
import sys
import ssl
import network
import ntptime
import time
import ujson
import random
from time import sleep
from picozero import pico_led  # onboard LED
from picozero import LED       # external LED
from picozero import pico_temp_sensor
from picozero import Button
from umqtt.simple import MQTTClient

AWS_ENDPOINT = "az13cio6a5b7n-ats.iot.us-east-1.amazonaws.com"
CLIENT_ID = "device_02"

PUBLISH_INTERVAL = 5  # Seconds between temperature readings

LED_TOPIC = f"things/{CLIENT_ID}/led".encode()
BUTTON_TOPIC = f"things/{CLIENT_ID}/button".encode()
STATE_TOPIC = f"things/{CLIENT_ID}/state".encode()

def connect_wifi():
    wlan = network.WLAN(network.STA_IF)
    wlan.active(True)
    print("Connecting to WiFi...")

```

```
>>> %Run -c $EDITOR_CONTENT

  MPY: soft reboot
  Connecting to WiFi...
  Connected to WiFi
  IP address: 10.228.7.144
  Connected to AWS IoT Core
  Published: {"led": 0, "temperature": 29.85, "humidity": 53, "button": "released"}
```

Manage packages for Raspberry Pi Pico @ /dev/cu.usbmodem1101
Search micropython-lib and PyPI
<INSTALL>
picozero
umqtt.simple
Close

**Install from PyPI**
If you don't know where to get the package from, then most likely you'll want to search the Python Package Index. Start by entering the name of the package in the search box above and pressing ENTER.

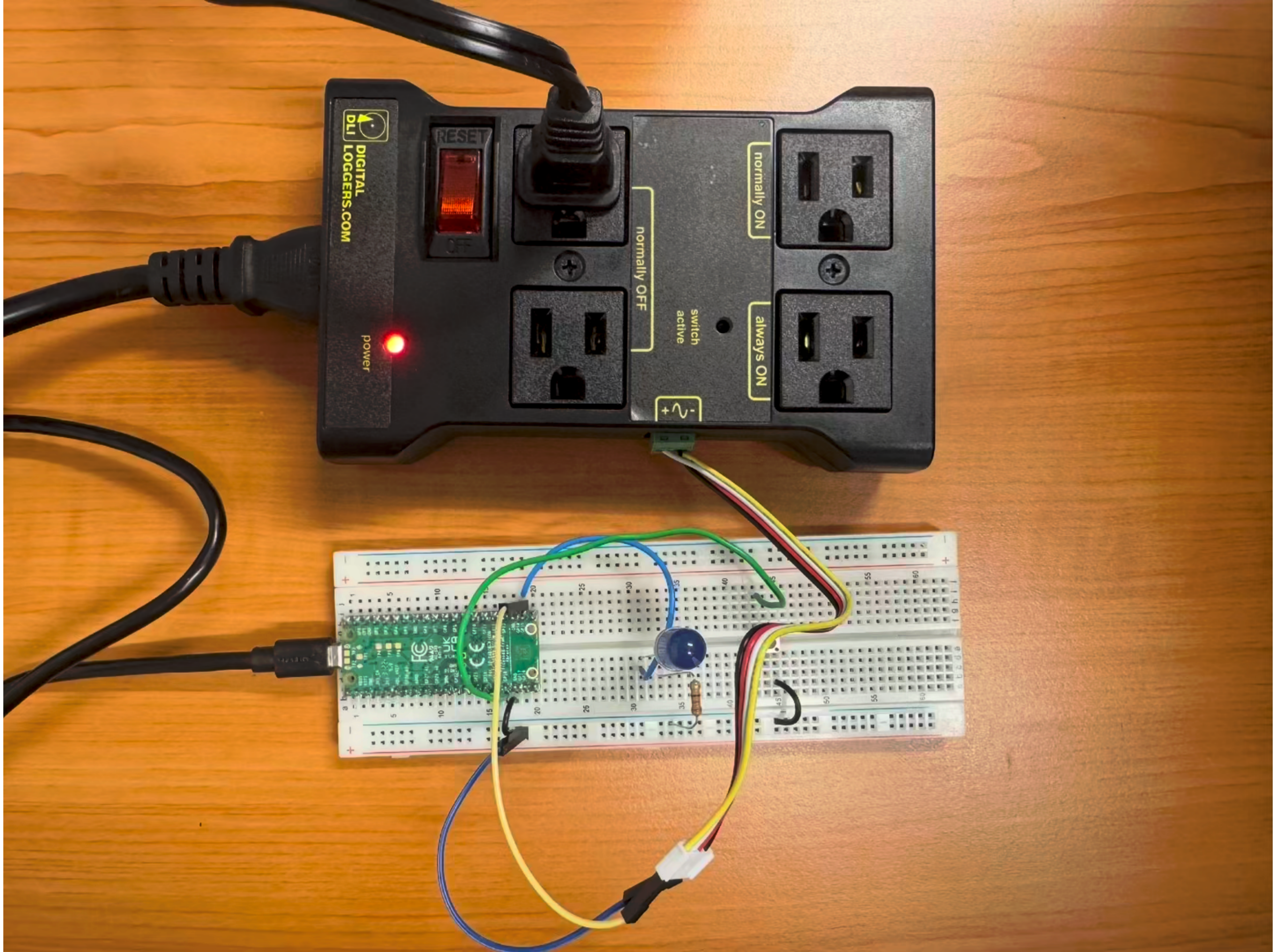
**Install from requirements file**
Click here to locate requirements.txt file and install the packages specified in it.

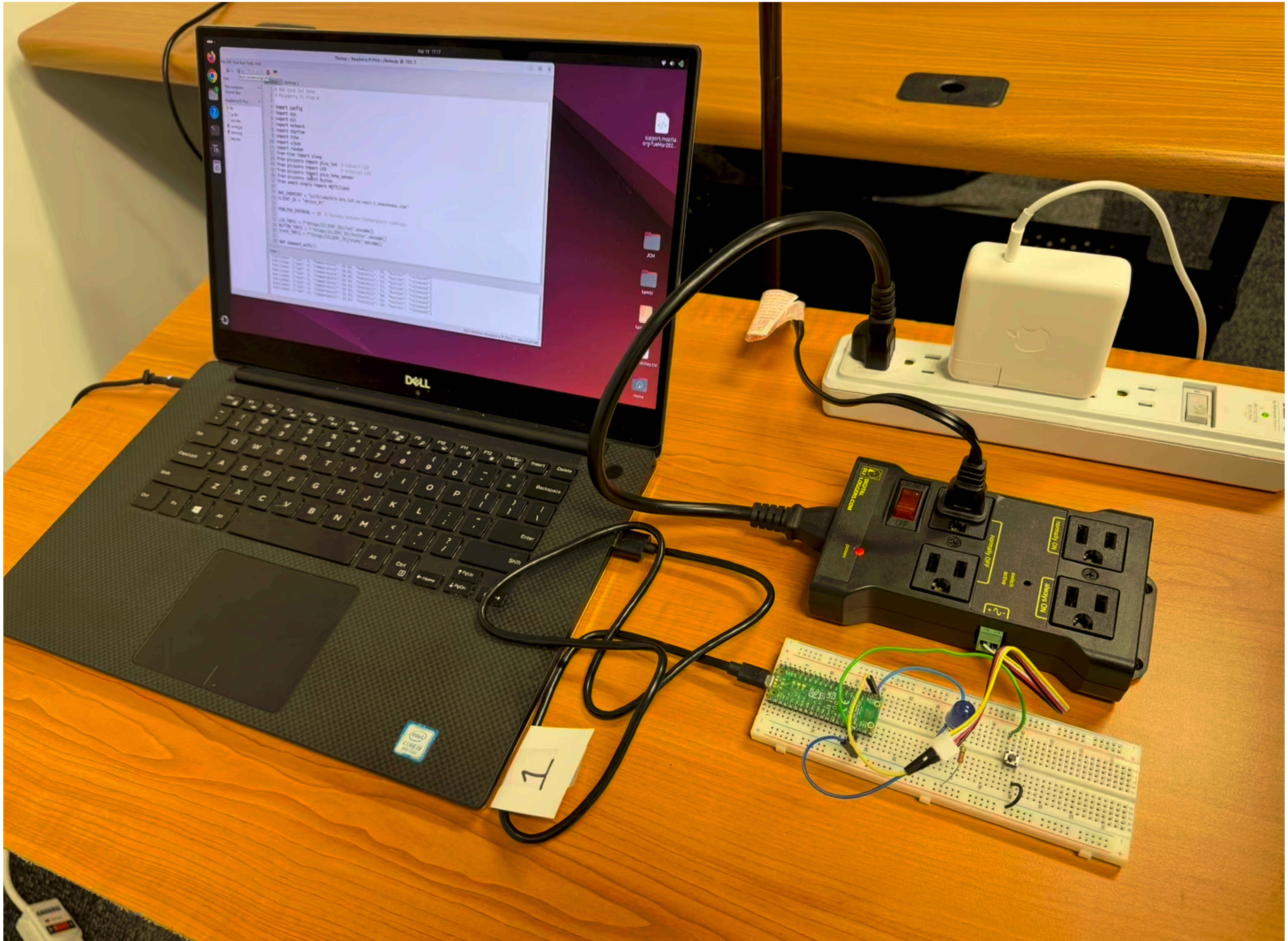
**Install from local file**
Click here to locate and install the package file (usually with .whl, .tar.gz or .zip extension).

**Upgrade or uninstall**
Start by selecting the package from the left.

**Target**
/lib

**Under the hood**
This dialog uses `pipkin`, a new command line tool for managing MicroPython and CircuitPython packages.
See https://pypi.org/project/pipkin/ for more info.


DLI
DIGITAL LOGGERS.COM
RESET
OFF
power
normally OFF
switch active
normally ON
always ON
+∿-


DELL
DIGITAL LOGGERS.COM
OFF
power
normally OFF
normally ON
switch active
always ON
1

# Demos