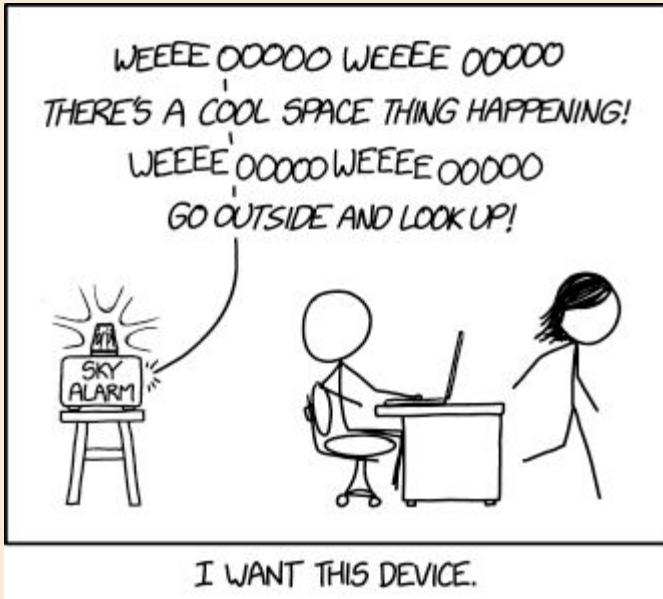


PyConUS 2025

Pittsburgh





You can and Should Build XKCD's "Sky Alarm" for Cool Space Things

Keith Murray (@KeithTheEE)

“Hello, PyCon!”

Keith Murray

KeithTheEE

Keith The EE (Electrical Engineer)

Kei Th Th e EE

Keithy



<https://github.com/KeithTheEE/PyConUS2025>

Slides, Notes, Links, Code can be found at:

<https://github.com/KeithTheEE/PyConUS2025>

<https://github.com/KeithTheEE/PyConUS2025>

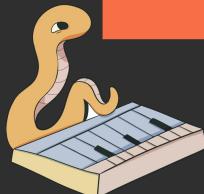


<https://github.com/KeithTheEE/PyConUS2025>

*PyConUS
2025*

Involved Online in the Python World

- Python Discord: Director/Outreach Lead
- PyOhio: Communication Chair
- PSF Education and Outreach Work Group
- Formerly a mod of the Python Subreddit (I love the project showcases)



<https://github.com/KeithTheEE/PyConUS2025>

PyCon US
2025

Involved Online in the Python World

- Python Discord: Director/**Outreach** Lead
- PyOhio: **Communication** Chair
- PSF Education and **Outreach** Work Group
- Formerly a mod of the Python Subreddit (I love the project **showcases**)



<https://github.com/KeithTheEE/PyConUS2025>

PyCon US
2025

I want YOU to build a Sky Alarm!

From the webcomic, XKCD

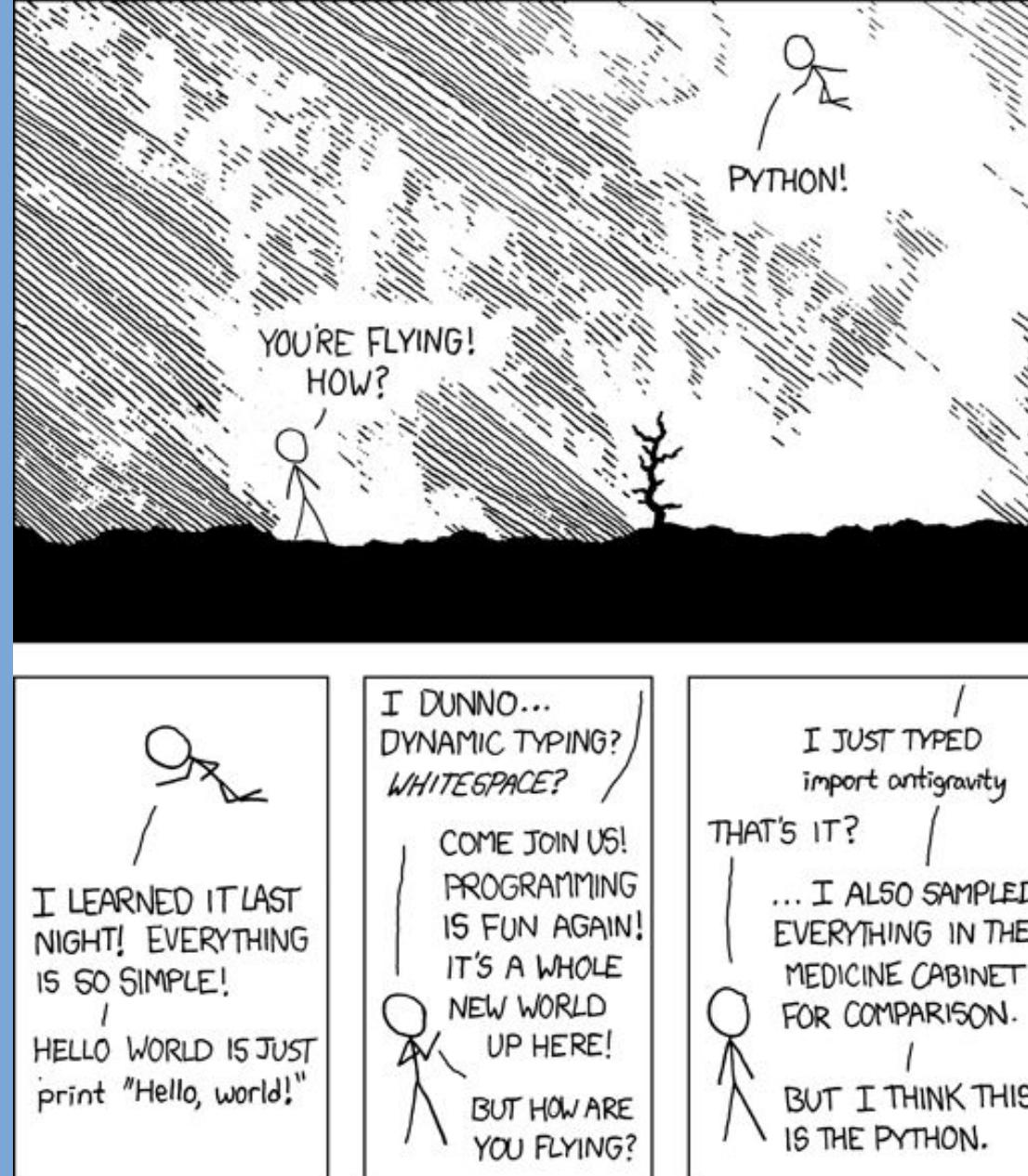
A webcomic of romance, sarcasm, math, and language.

import antigravity



<https://github.com/KeithTheEE/PyConUS2025>

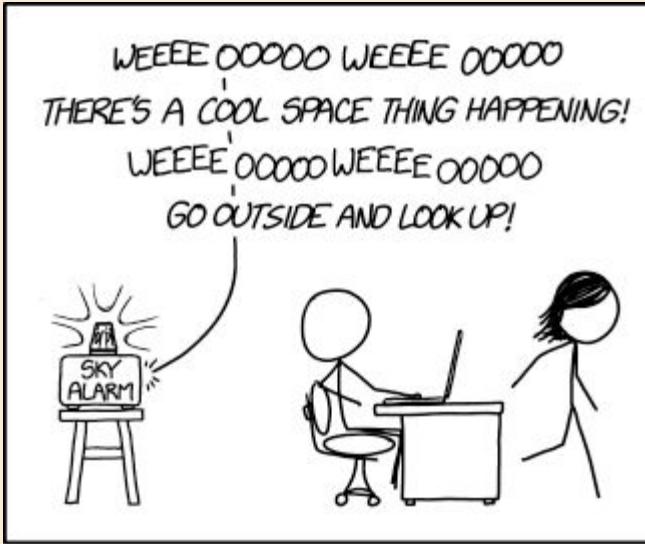
PyCon US
2025



<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

One such webcomic is titled, “Sky Alarm”



During the day it also activates for neat clouds and pretty sunsets



<https://github.com/KeithTheEE/PyConUS2025>

PyCon US
2025

I want you to build it!

- Introduction
- Basic Requirements
- The Sky Alarm
- Bells
- Whistles

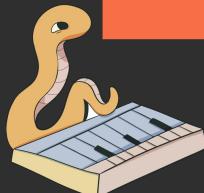
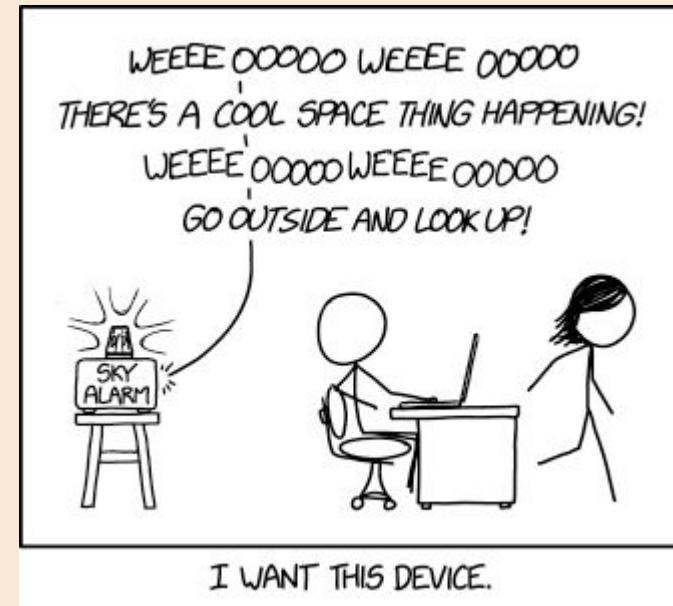


<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

What is needed?

- Box with the label “Sky Alarm”
- Something to make noise
- Something to make blinking lights
- Some way to tell if a cool space thing is happening



<https://github.com/KeithTheEE/PyConUS2025>

PyCon US
2025

Perfect use case for Microcontrollers

And Sensors!

And since this is a Python Conference:

Python!

(by way of Circuit Python)



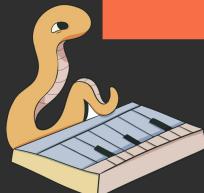
<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

Something to make blinking lights (LED)

```
# Blinking LED
## Setting Up the Pin
pixels = neopixel.NeoPixel(board.NEOPIXEL, 1, brightness=0.3)
pixels.direction = digitalio.Direction.OUTPUT

# Turning it on and Off
pixels[0] = (180, 0, 255)
pixels.show()
time.sleep(.1)
pixels[0] = (0, 0, 0)
pixels.show()
```



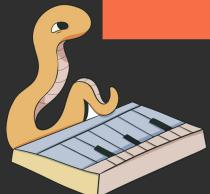
<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

Something to make noise (Buzzer)

```
# Buzzing a Buzzer
## Setting Up the Pin
buzzer = DigitalInOut(board.A0)
buzzer.direction = digitalio.Direction.OUTPUT

# Turning it on and Off
buzzer.value = True
time.sleep(.1)
buzzer.value = False
```



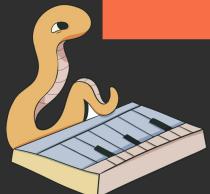
<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

Putting it all together into a full program:

```
# Imports
import board
import time

import digitalio
from digitalio import DigitalInOut
import neopixel
```



<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

Setting up the Pins

```
# Setup
# Blinking LED
## Setting Up the Pin
pixels = neopixel.NeoPixel(board.NEOPIXEL, 1, brightness=0.3)
pixels.direction = digitalio.Direction.OUTPUT
# Buzzing a Buzzer
## Setting Up the Pin
buzzer = DigitalInOut(board.A0)
buzzer.direction = digitalio.Direction.OUTPUT
```

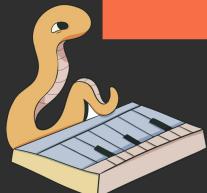


<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

Make the Lights and Sounds into a Function

```
def sky_alarm(pixels, buzzer):
    # Turning it on and Off
    pixels[0] = (180, 0, 255)
    pixels.show()
    buzzer.value = True
    time.sleep(0.1)
    buzzer.value = False
    pixels[0] = (0, 0, 0)
    pixels.show()
    time.sleep(0.1)
    return
```

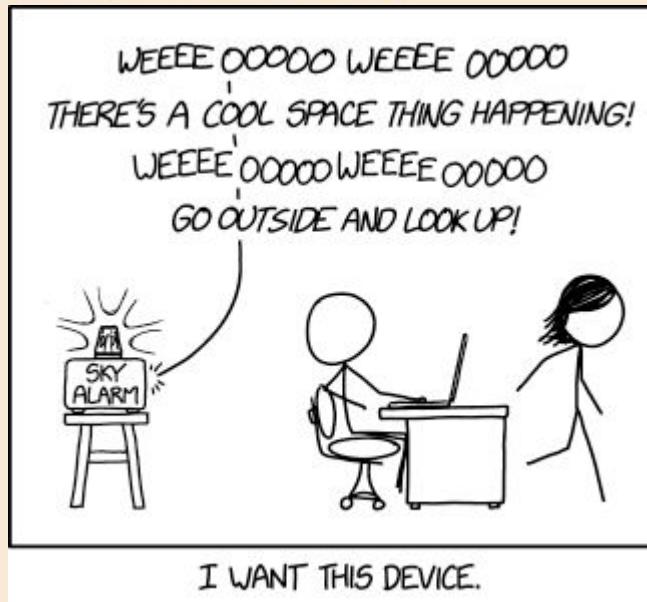


<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

Call it when a Cool Space Thing happens!

```
# Main Loop
while True:
    if cool_space_thing:
        sky_alarm(pixels, buzzer)
```

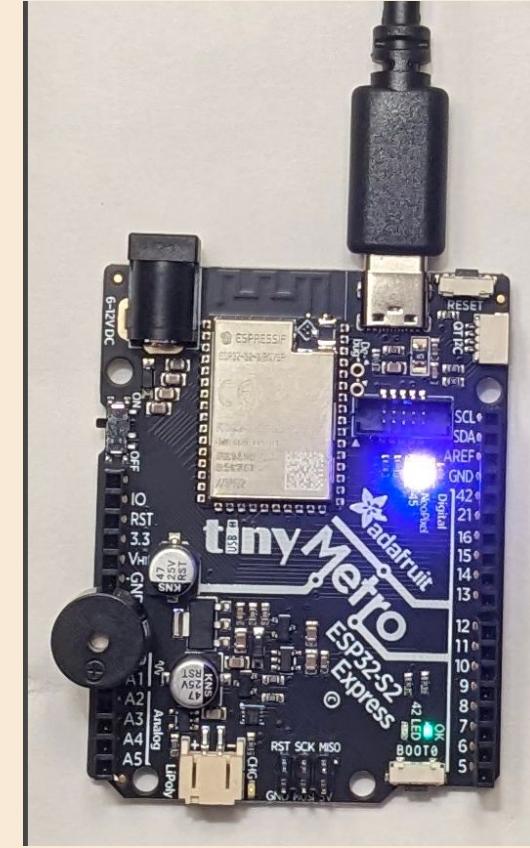
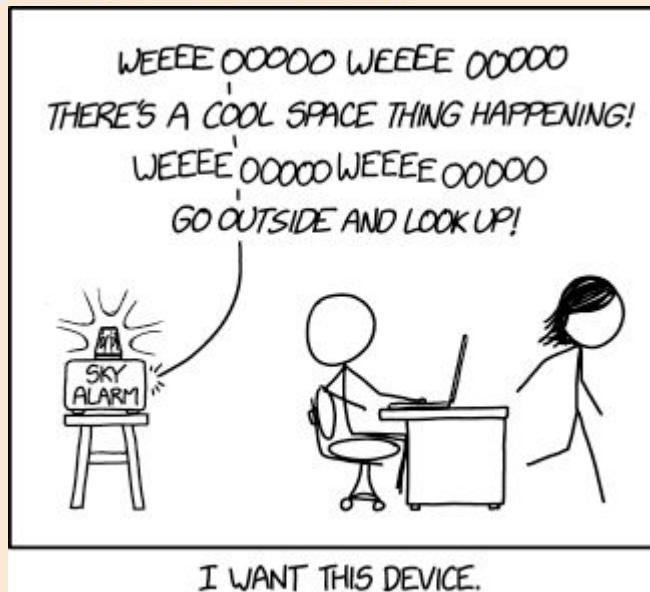


<https://github.com/KeithTheEE/PyConUS2025>

PyCon US
2025

Call it when a Cool Space Thing happens!

```
# Main Loop
while True:
    if cool_space_thing:
        sky_alarm(pixels, buzzer)
```

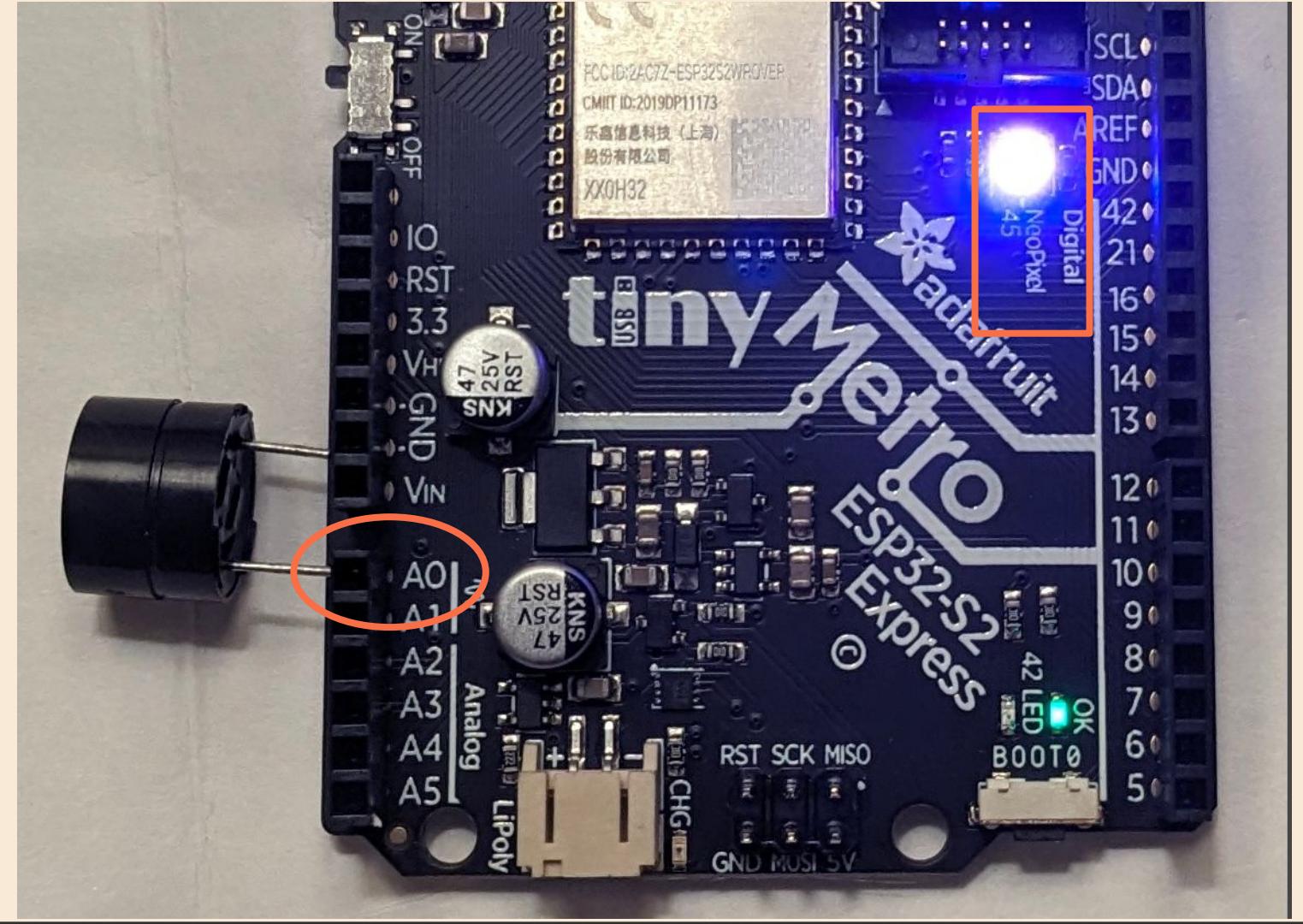


<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

Buzzer sits on
pin A0

LED is
“NeoPixel”



<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

Microcontrollers, Circuit Python, MicroPython

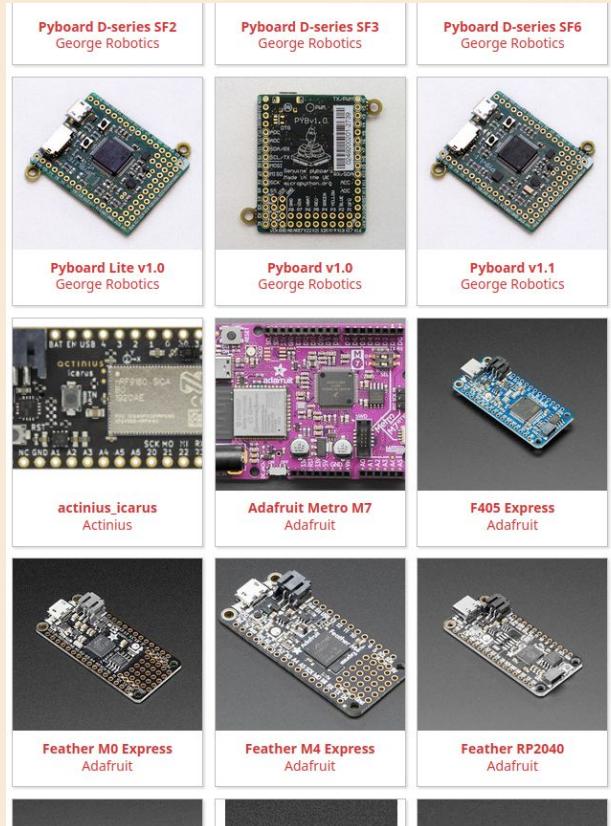
- are small integrated circuits, small tasks
- Lower power, slower speed, less memory, cheap
- MicroPython is a subset of the Python Language
- Circuit Python is a fork of MicroPython
- Python Libraries don't all work on Micro/Circuit Python
- Not every Microcontroller can run Micro/Circuit Python
- Board has a microcontroller on it (plus more)
- This Explanation is confusing, but they're tiny computers



<https://github.com/KeithTheEE/PyConUS2025>

*PyCon US
2025*

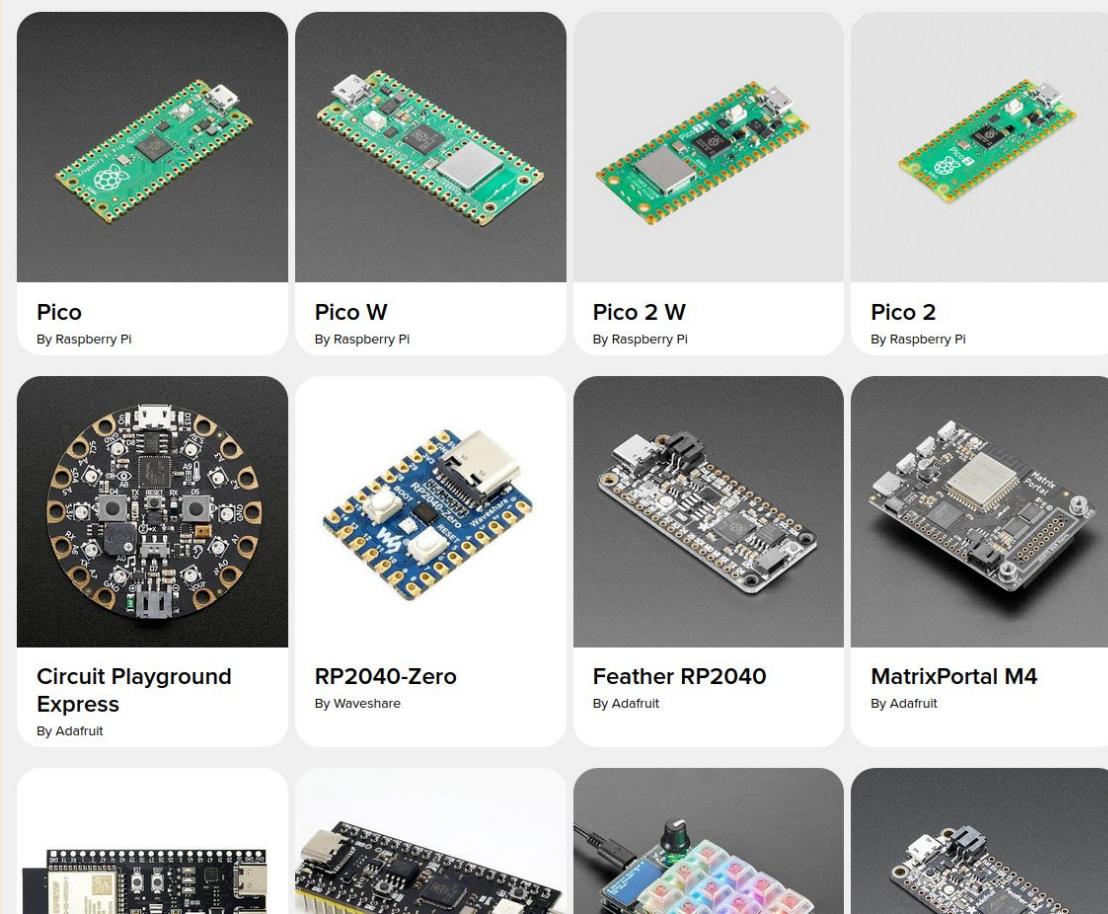
<https://micropython.org/download/>



<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

<https://circuitpython.org/downloads>



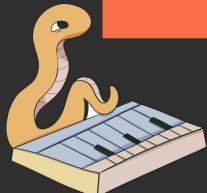
<https://github.com/KeithTheEE/PyConUS2025>

PyCon US
2025

<https://circuitpython.org/downloads>

Search for CircuitPython boards  Filters   

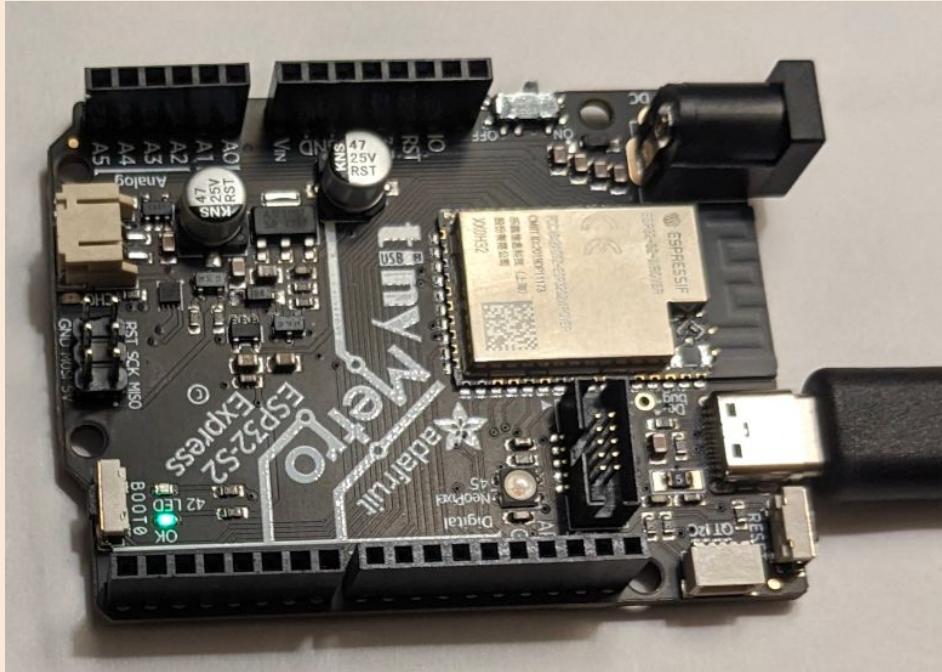
Manufacturers	Features	Sort By
<input type="checkbox"/> 01space	<input type="checkbox"/> Arduino Shield Compatible	<input checked="" type="radio"/> Downloads
<input type="checkbox"/> 0xCB	<input type="checkbox"/> Battery Charging	<input type="radio"/> Board Name (A To Z)
<input type="checkbox"/> 2231puppy	<input type="checkbox"/> Bluetooth/BTLE	<input type="radio"/> Board Name (Z To A)
<input type="checkbox"/> 42. Keebs	<input type="checkbox"/> Breadboard-Friendly	<input type="radio"/> Date Added (Newest First)
<input type="checkbox"/> 8086 Consultancy	<input type="checkbox"/> Camera	<input type="radio"/> Date Added (Oldest First)
<input type="checkbox"/> Adafruit	<input type="checkbox"/> Castellated Pads	
<input type="checkbox"/> Ai-Thinker	<input type="checkbox"/> Display	
<input type="checkbox"/> Alethea Flowers	<input type="checkbox"/> External Display	
	<input type="checkbox"/> Feather-Compatible	
	<input type="checkbox"/> GPS	
	<input type="checkbox"/> LoRa/Radio	
	<input type="checkbox"/> Raspberry Pi Pico Form Factor	
	<input type="checkbox"/> Robotics	
	<input type="checkbox"/> Solder-Free Alligator Clip	
	<input type="checkbox"/> Speaker	
	<input checked="" type="checkbox"/> STEMMA QT/QWIIC	
	<input type="checkbox"/> USB Host	
	<input type="checkbox"/> USB-C	
	<input checked="" type="checkbox"/> Wi-Fi	
	<input type="checkbox"/> Xiao / QTPy Form Factor	



<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

Metro ESP32-S2 Express



<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

File System

code.py

settings.toml

lib/



<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

So We've got an Alarm!

Wait, we need the sky part. The: “cool_space_thing”

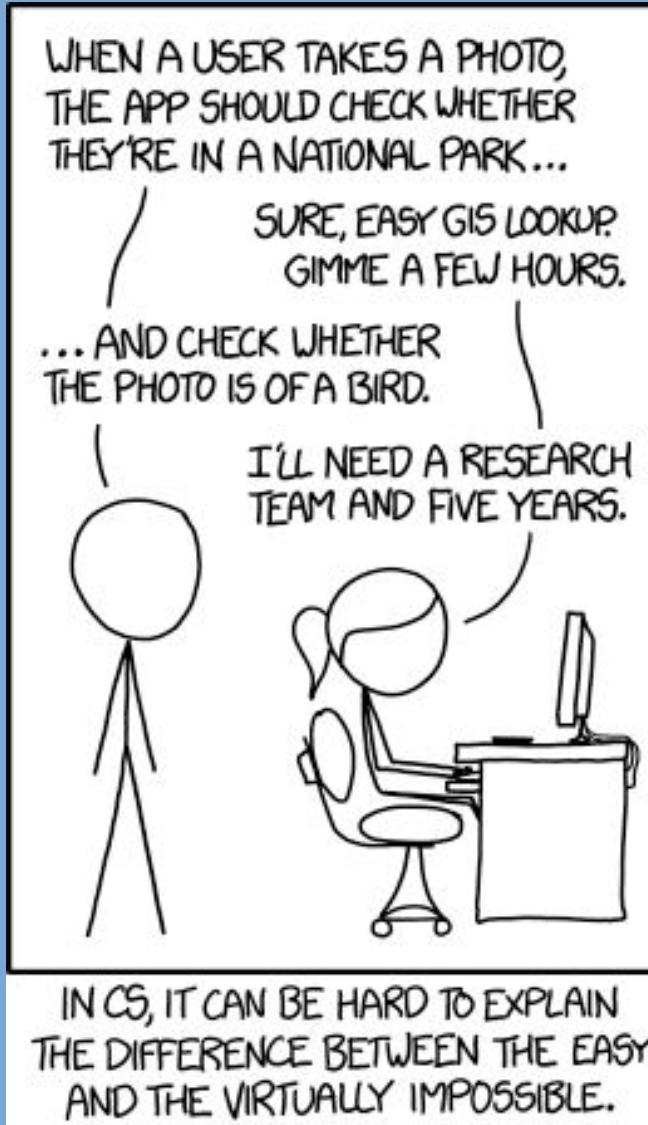
```
# Main Loop
while True:
    if cool_space_thing:
        sky_alarm(pixels, buzzer)
```



<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

Let's take inspiration from another comic:



<https://github.com/KeithTheEE/PyConUS2025>

PyCon US
2025

Northern Lights!

NOAA has a dashboard for forecasts of Aurora:

<https://www.swpc.noaa.gov/communities/aurora-dashboard-experimental>

And specific forecasts:

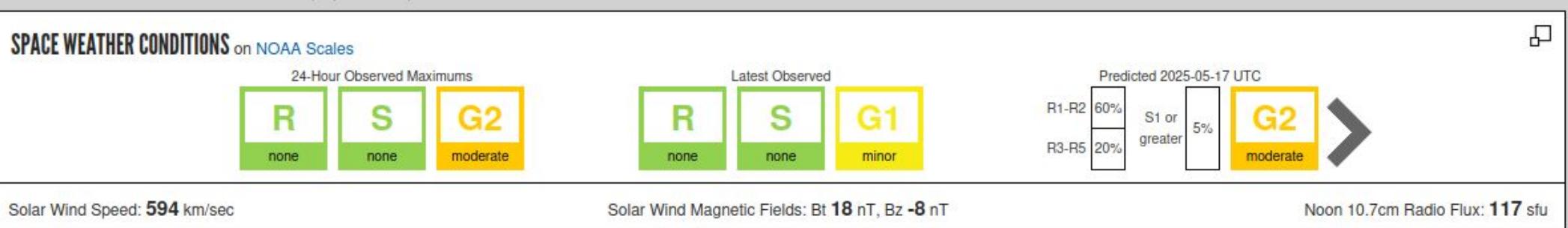
<https://www.swpc.noaa.gov/products/3-day-forecast>

<https://services.swpc.noaa.gov/text/3-day-forecast.txt>



<https://github.com/KeithTheEE/PyConUS2025>

PyCon US
2025



AURORA DASHBOARD (EXPERIMENTAL)

AURORA DASHBOARD INFORMATION



When and where can you see the northern and southern lights also known as the aurora? This page provides a prediction of the aurora's visibility tonight and tomorrow night in the charts below. The animations further down show what the aurora's been up to over the last 24 hours and estimates what the next 30 minutes will be like. The aurora's colorful green, red, and purple light shifts gently and often changes shape like softly blowing curtains.

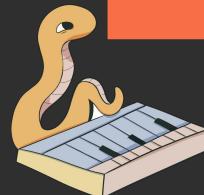
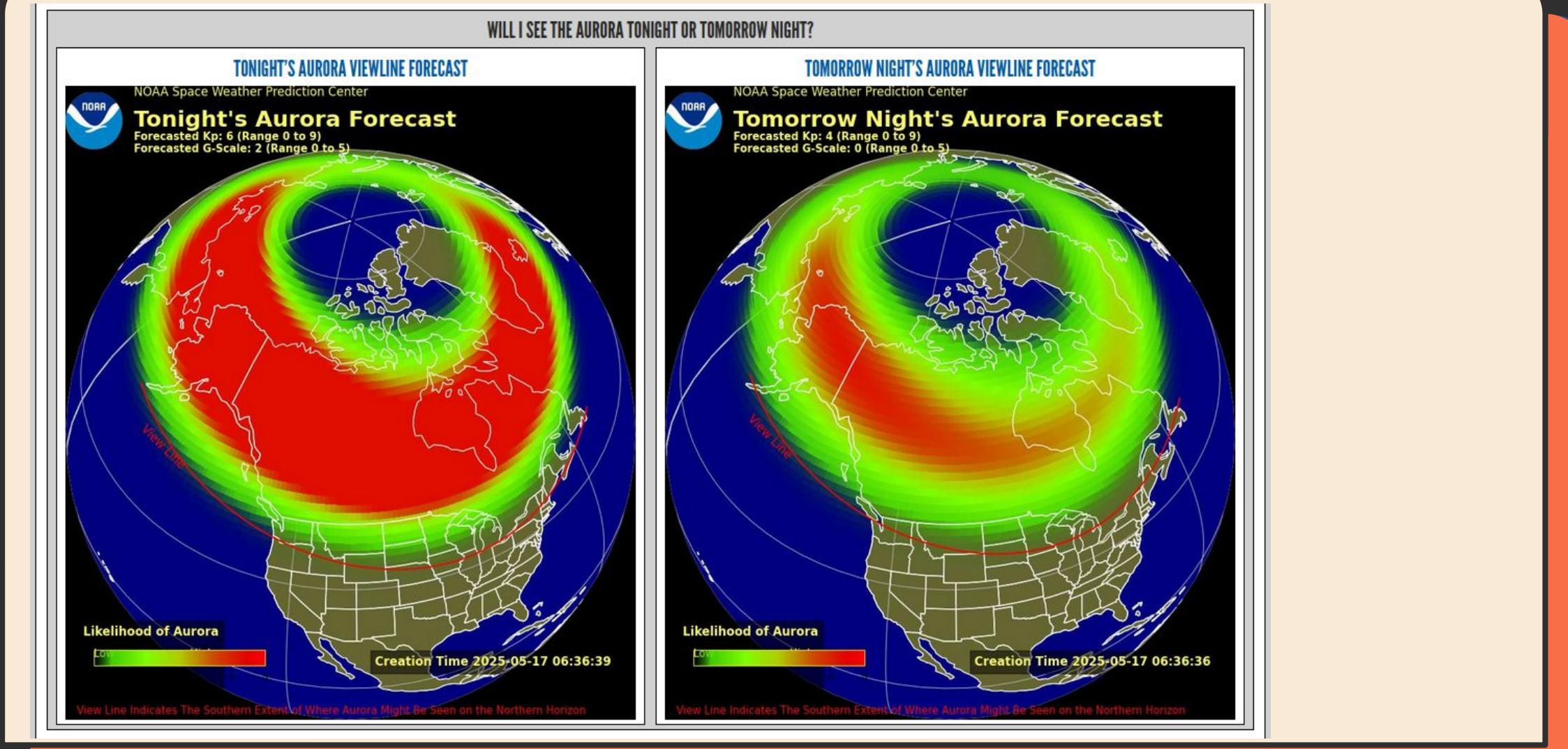
Helpful links:

- [Dive Deeper Into the Science of the Aurora](#)
- [Tips on Viewing the Aurora](#)
- [Learn More About Space Weather](#)



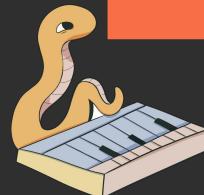
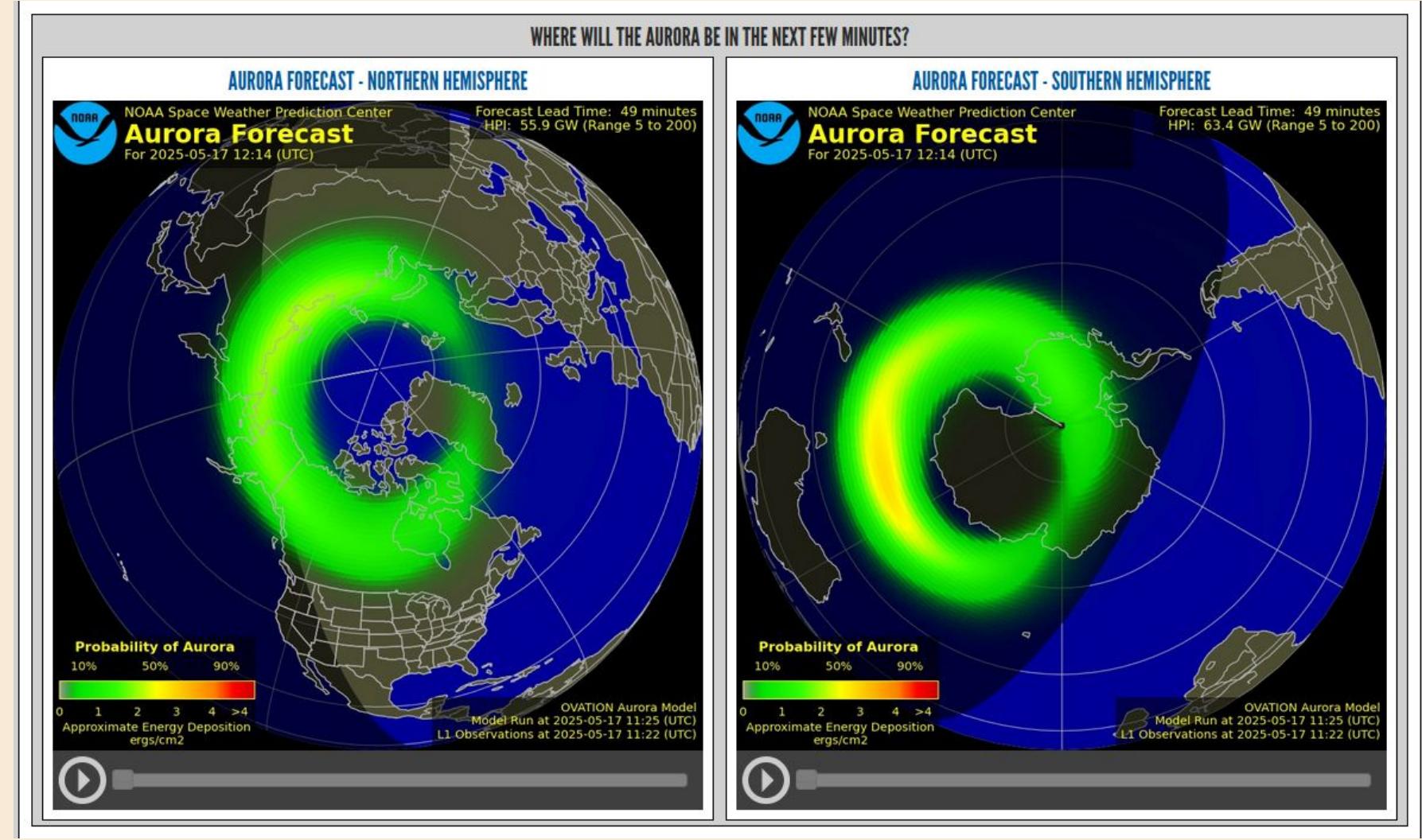
<https://github.com/KeithTheEE/PyConUS2025>

PyCon US
2025



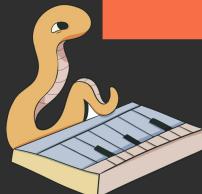
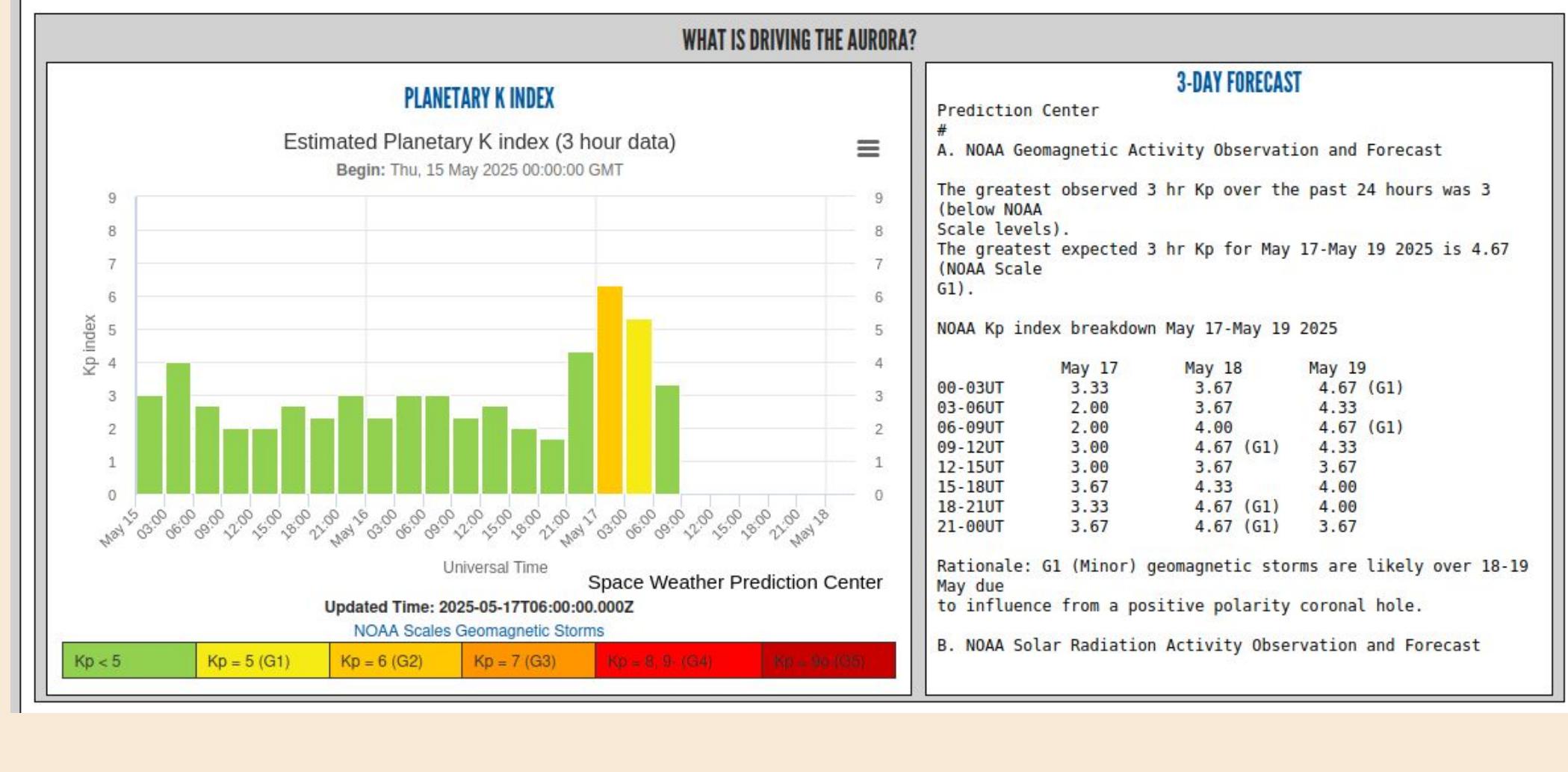
<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025



<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025



<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

:Product: 3-Day Forecast
:Issued: 2025 May 17 0030 UTC
Prepared by the U.S. Dept. of Commerce, NOAA, Space Weather Prediction Center

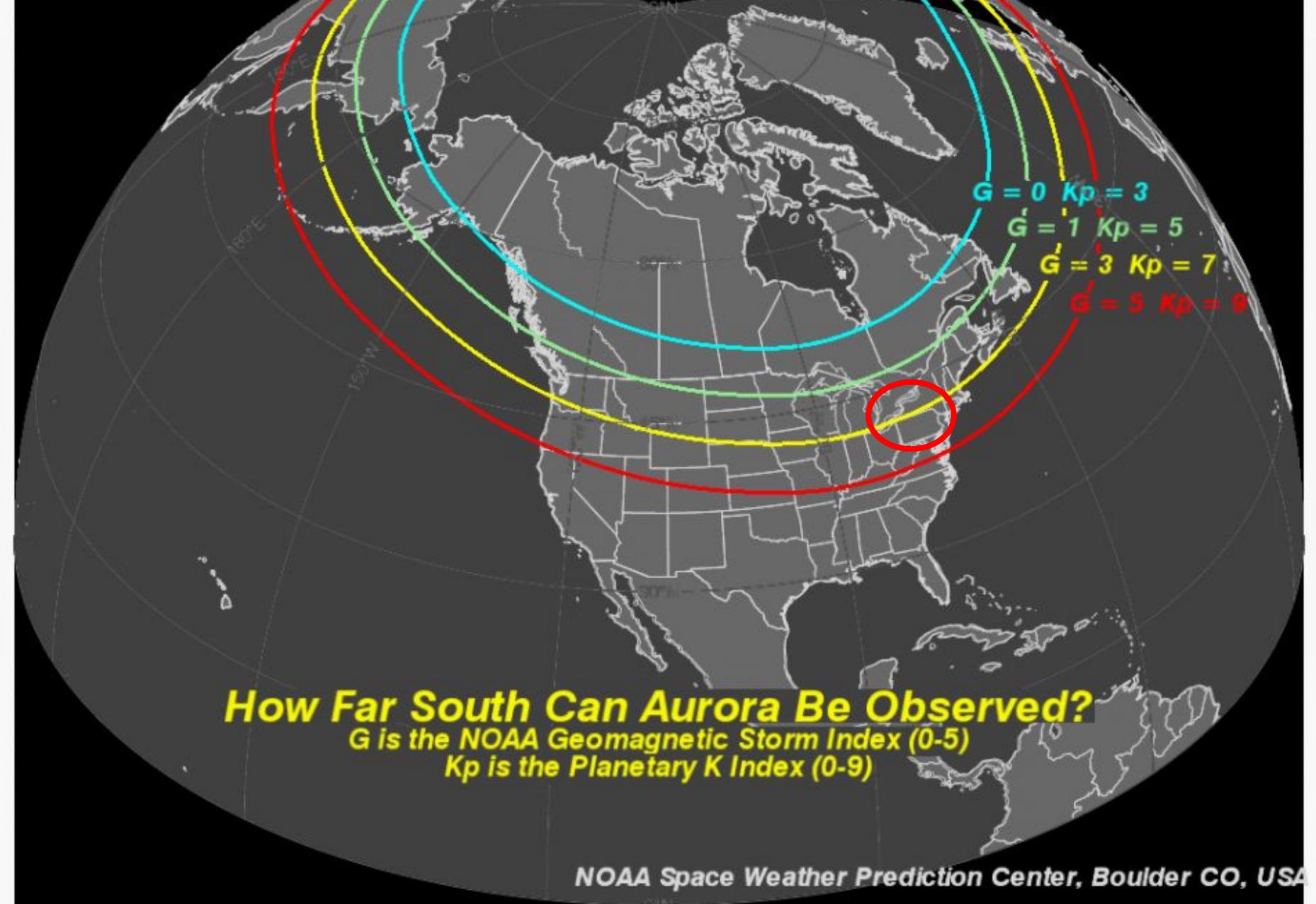
A. NOAA Geomagnetic Activity Observation and Forecast

The greatest observed 3 hr Kp over the past 24 hours was 3 (below NOAA Scale levels).

The greatest expected 3 hr Kp for May 17-May 19 2025 is 4.67 (NOAA Scale G1).

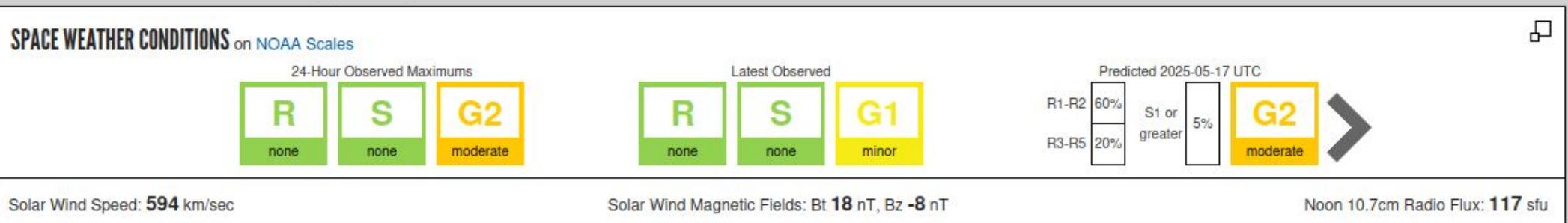
NOAA Kp index breakdown May 17-May 19 2025

	May 17	May 18	May 19
00-03UT	3.33	3.67	4.67 (G1)
03-06UT	2.00	3.67	4.33
06-09UT	2.00	4.00	4.67 (G1)
09-12UT	3.00	4.67 (G1)	4.33
12-15UT	3.00	3.67	3.67
15-18UT	3.67	4.33	4.00
18-21UT	3.33	4.67 (G1)	4.00
21-00UT	3.67	4.67 (G1)	3.67



<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025



AURORA DASHBOARD (EXPERIMENTAL)



AURORA DASHBOARD INFORMATION

When and where can you see the northern and southern lights also known as the aurora? This page provides a prediction of the aurora's visibility tonight and tomorrow night in the charts below. The animations further down show what the aurora's been up to over the last 24 hours and estimates what the next 30 minutes will be like. The aurora's colorful green, red, and purple light shifts gently and often changes shape like softly blowing curtains.

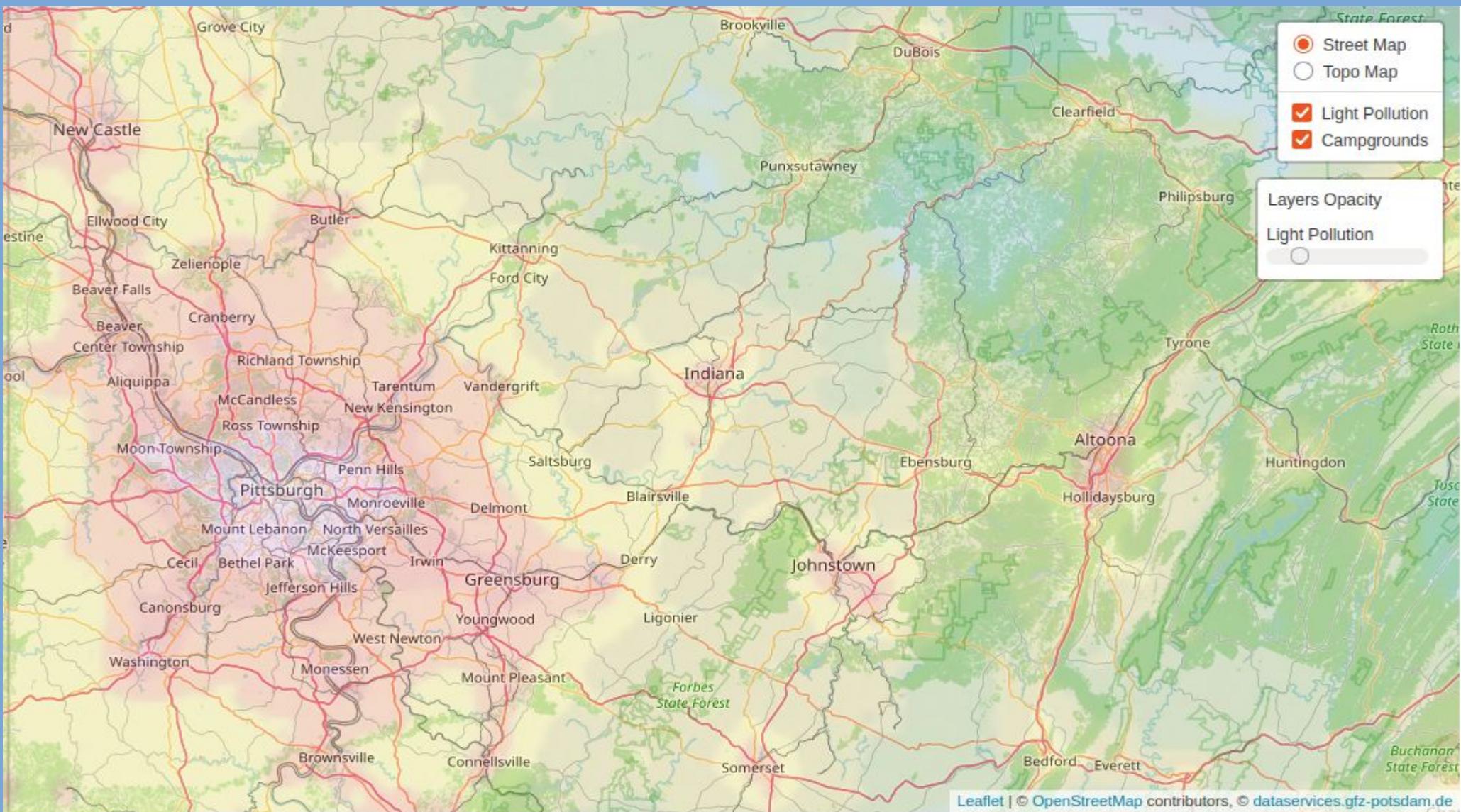
Helpful links:

- [Dive Deeper Into the Science of the Aurora](#)
- [Tips on Viewing the Aurora](#)
- [Learn More About Space Weather](#)



<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025



<https://rwtadventures.com/stargazing-campground-map/>

<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

White: Level 9 – Inner-city Sky

Magenta: Level 8 – City Sky

Red: Level 7 – Suburban/Urban Transition

Orange: Level 6 – Bright Suburban Sky

Yellow: Level 5 – Suburban Sky

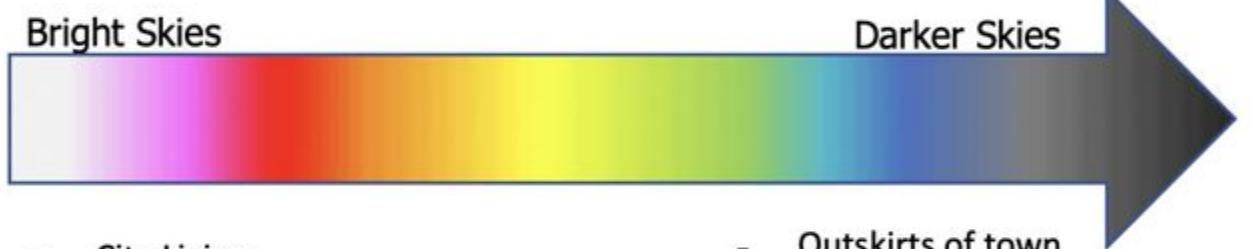
Green: Level 4 – Rural/Suburban Transition

Blue: Level 3 – Rural Sky

Gray: Level 2 – Typical Truly Dark Sky

Black: Level 1 – Excellent Dark Sky Site

Light Pollution Color Scale

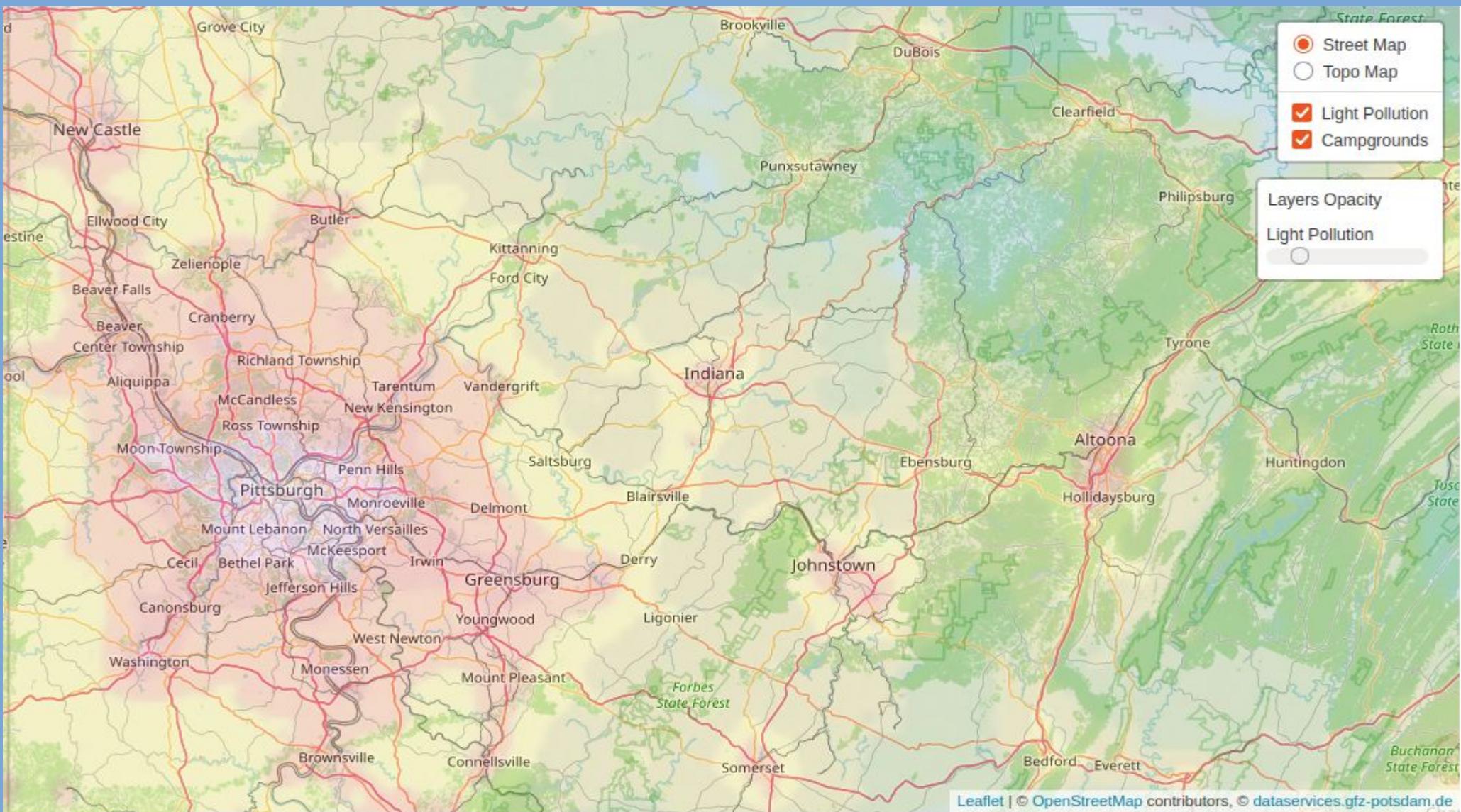


- City Living
- Suburban Neighborhoods
- Outskirts of town
- Rural Land
- Uninhabited Areas

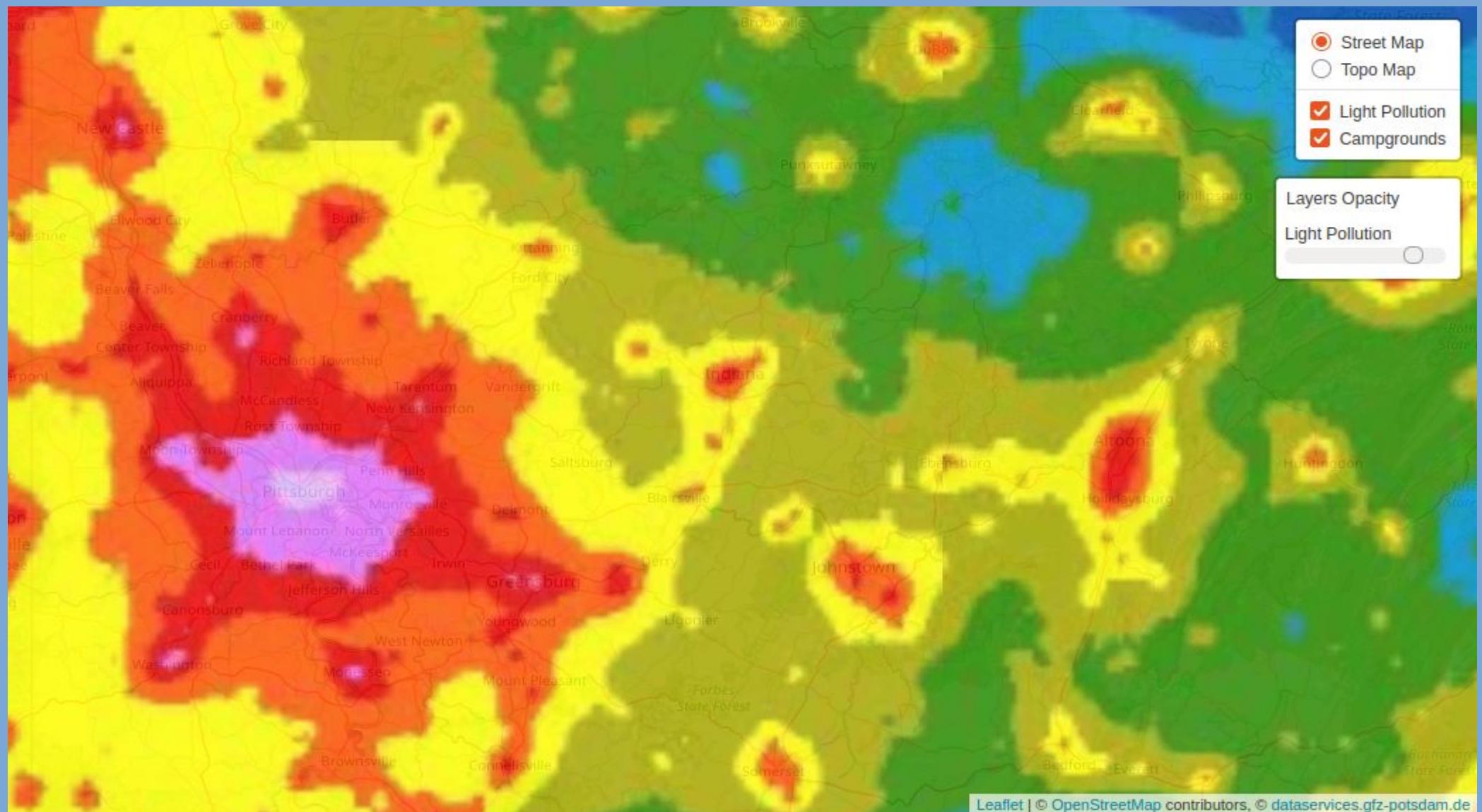
The Bortle scale simplified | © RWT Adventures

<https://rwtadventures.com/stargazing-campground-map/>

<https://github.com/KeithTheEE/PyConUS2025>



<https://rwtadventures.com/stargazing-campground-map/>
<https://github.com/KeithTheEE/PyConUS2025>

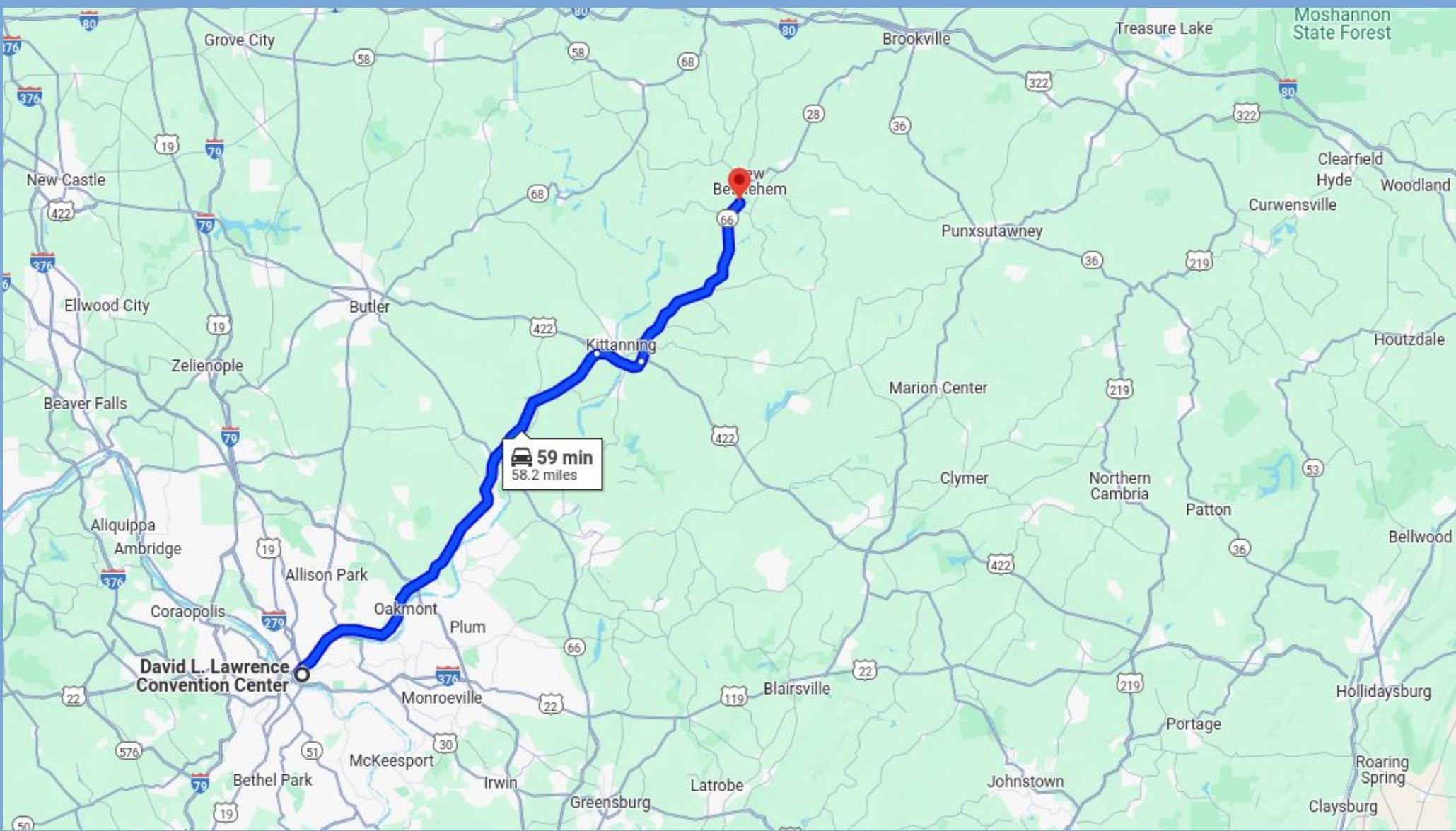


Leaflet | © OpenStreetMap contributors, © dataservices.gfz-potsdam.de

<https://rwtadventures.com/stargazing-campground-map/>

<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025



<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

:Product: 3-Day Forecast
:Issued: 2025 May 17 0030 UTC
Prepared by the U.S. Dept. of Commerce, NOAA, Space Weather Prediction Center

A. NOAA Geomagnetic Activity Observation and Forecast

The greatest observed 3 hr Kp over the past 24 hours was 3 (below NOAA Scale levels).

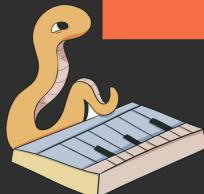
The greatest expected 3 hr Kp for May 17-May 19 2025 is 4.67 (NOAA Scale G1).

NOAA Kp index breakdown May 17-May 19 2025

	May 17	May 18	May 19
00-03UT	3.33	3.67	4.67 (G1)
03-06UT	2.00	3.67	4.33
06-09UT	2.00	4.00	4.67 (G1)
09-12UT	3.00	4.67 (G1)	4.33
12-15UT	3.00	3.67	3.67
15-18UT	3.67	4.33	4.00
18-21UT	3.33	4.67 (G1)	4.00
21-00UT	3.67	4.67 (G1)	3.67

Get data if a cool space thing is happening

- All on a micro controller*
- Import libraries
- Turn on Wifi Radio
- Sign onto the network
- Make a get request to the server
- Parse the data



<https://github.com/KeithTheEE/PyConUS2025>

PyCon US
2025

Import Libraries, get the Radio Ready

```
import os
import adafruit_connection_manager
import wifi
import adafruit_requests

# Initialize Wifi, Socket Pool, Request Session
pool = adafruit_connection_manager.get_radio_socketpool(wifi.radio)
ssl_context = adafruit_connection_manager.get_radio_ssl_context(wifi.radio)
requests = adafruit_requests.Session(pool, ssl_context)
```



<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

Sign on to Network

```
# Get WiFi details, ensure these are setup in settings.toml
ssid = os.getenv("WIFI_SSID")
password = os.getenv("WIFI_PASSWORD")
wifi.radio.connect(ssid, password)
```



<https://github.com/KeithTheEE/PyConUS2025>

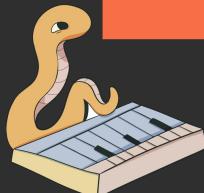
PyConUS
2025

Get information in requests like fashion

```
requests = adafruit_requests.Session(pool, ssl_context)

target_url = "https://services.swpc.noaa.gov/text/3-day-forecast.txt"
TEXT_URL = target_url

with requests.get(TEXT_URL) as response:
    print(f" | GET Response: {response.text}")
```

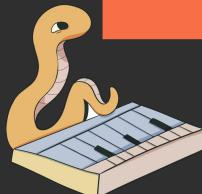


<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

Parse the returned text

```
i = 0
in_table = 0
kp_of_interest = []
text = response.text.split('\n')
for line in text:
    if "NOAA Kp index breakdown" in line:
        in_table = True
    if in_table:
        if i < 6:
            if "UT" in line:
                kp_of_interest.append ( float(line.split()[1]))
    i += 1
```



<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

Parse the returned Text

```
i = 0
in_table = 0
kp_of_interest = []
text = response.text.split('\n')
for line in text:
    if "NOAA Kp index breakdown" in line:
        in_table = True
    if in_table:
        if i < 6:
            if "UT" in line:
                kp_of_interest.append ( float(line.split()[1]))
        i += 1
```



<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

Parse the returned text

```
kp_peak = max(kp_of_interest)
if kp_peak >= 5:
    cool_space_thing = True
```



<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

And that's a Sky Alarm!

Thank you!



<https://github.com/KeithTheEE/PyConUS2025>

PyCon US
2025

But, we can do more..

Add more Lights

Play audio, not just a buzzer



<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

LED Strip

```
# Blinking LED
## Setting Up the Pin
pixels = neopixel.NeoPixel(board.NEOPIXEL, 1, brightness=0.3)
pixels.direction = digitalio.Direction.OUTPUT
```



<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

LED Strip

```
# Blinking LED
## Setting Up the Pin
pixels = neopixel.NeoPixel(board.NEOPIXEL, 1, brightness=0.3)
pixels.direction = digitalio.Direction.OUTPUT

led_count = 30
pixels = neopixel.NeoPixel(board.IO12, led_count, brightness=0.3)
pixels.direction = digitalio.Direction.OUTPUT
```



<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

LED Strip

Add animation

```
import ulab
raw_color = ulab.numpy.array([180,0,255])
def fade(basecolor, w):
    adj = ulab.numpy.floor(((ulab.numpy.cos(w)+1)/2)*basecolor)
    return tuple(adj)
```



<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

LED Strip

Add animation

```
w = 0
while True:
    for i in range(led_count):
        pixels[i] = fade(raw_color, (i+1)*w/120)
    w+= 1
    print(pixels[0])
    pixels.show()
```



<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

Playing Sound

```
from audiocore import WaveFile
from audioio import AudioOut

wave_file = open("SkyAlarm.wav", "rb")
wave = WaveFile(wave_file)
audio = AudioOut(board.A1)
```



<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

Playing Sound

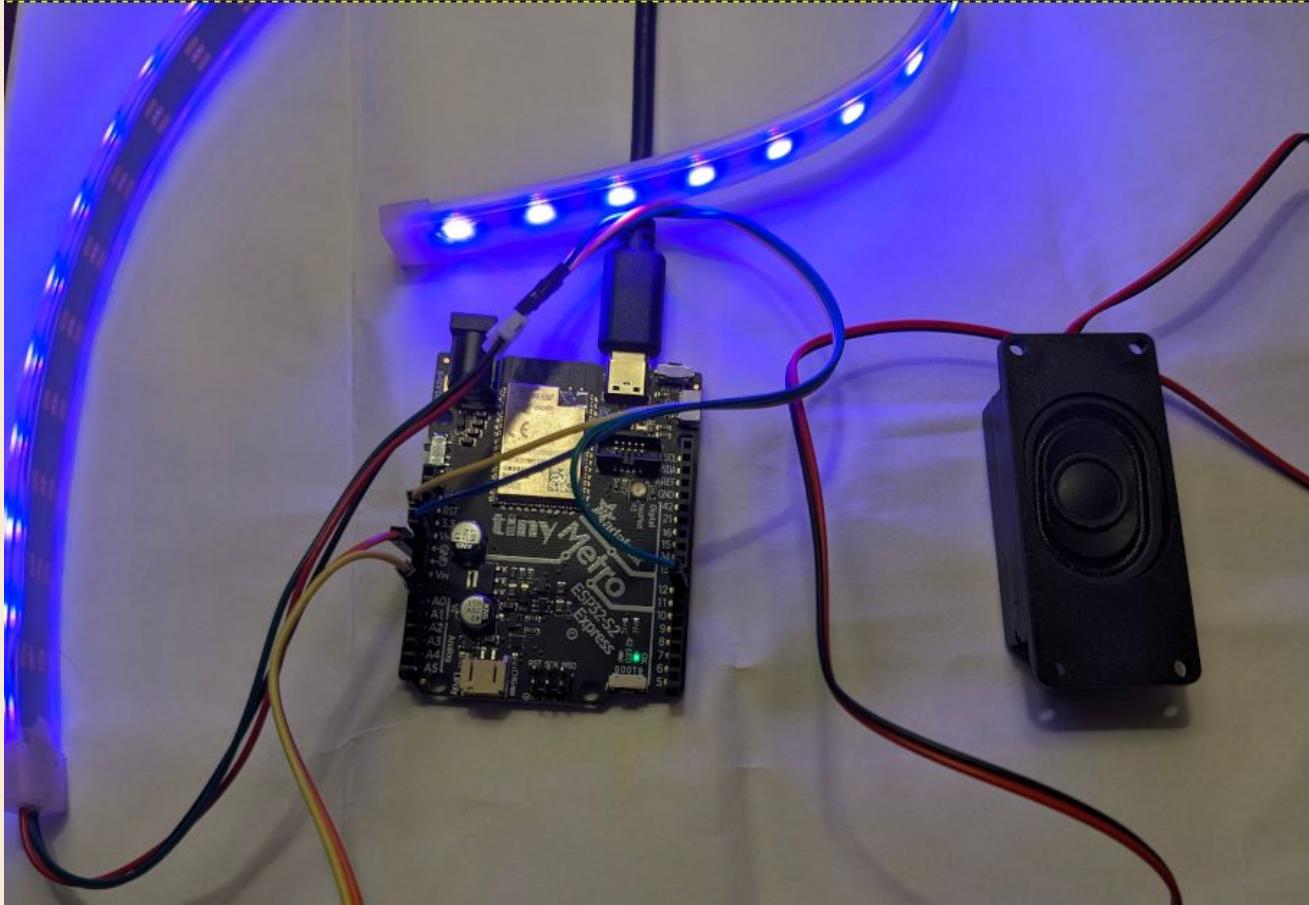
```
while True:  
    audio.play(wave)  
  
    while audio.playing:  
        for i in range(led_count):  
            pixels[i] = fade(raw_color, (i+1)*w/120)  
        w+= 1  
        print(pixels[0])  
        pixels.show()  
    pass
```



<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

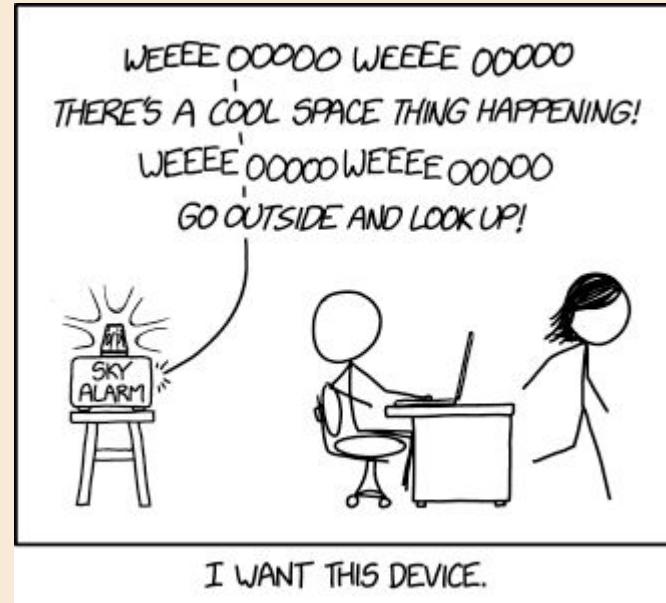
Sky ALARM



<https://github.com/KeithTheEE/PyConUS2025>

PyCon US
2025

Sky ALARM



<https://github.com/KeithTheEE/PyConUS2025>

*PyConUS
2025*

But, we can Still do more..

Give the Sky Alarm a sense of “When”

Log when the alarm has gone off

Add a local server

Using Flask



<https://github.com/KeithTheEE/PyConUS2025>

PyCon US
2025

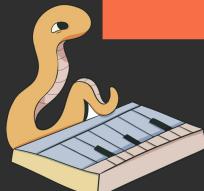
Making a Flask Server

```
from flask import Flask
from flask import Flask, render_template, request, \
    redirect, url_for, json, send_from_directory

app = Flask(__name__)

@app.route("/")
def hello_world():
    return "<p>Hello, World!</p>"
```

```
$ python -m flask --app simple_server run --host=0.0.0.0
```



<https://github.com/KeithTheEE/PyConUS2025>

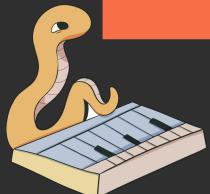
PyConUS
2025

Add functions to get UTC and Local Time on Flask Server

```
import datetime
import json

def get_datetime_utc():
    return datetime.datetime.now(datetime.timezone.utc)

def get_datetime_local():
    return get_datetime_utc().astimezone()
```

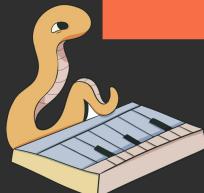


<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

Making a web page display the time

```
@app.route("/api/current_time_since_epoch", methods=['GET'])
def return_current_time():
    utc_now=datetime.datetime.now(datetime.timezone.utc)
    my_now = utc_now.astimezone()
    data = {"utc_time_tuple":tuple(utc_now.timetuple()),
            "local_time_tuple":tuple(my_now.timetuple()),
            "time_since_epoch_seconds":int(utc_now.timestamp()),
            "time_since_epoch_microseconds":utc_now.microsecond}
    return json.dumps(data)
```



<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

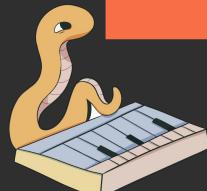
Getting and Setting time on the Microcontroller

Last internet fetch:

```
requests = adafruit_requests.Session(pool, ssl_context)

target_url = "https://services.swpc.noaa.gov/text/3-day-forecast.txt"
TEXT_URL = target_url

with requests.get(TEXT_URL) as response:
    print(f" | GET Response: {response.text}")
```



<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

Getting and Setting time on the Microcontroller

```
import json
import rtc

local_network = "http://192.168.00.123:5000/"
fetch_time_target = local_network + "/api/current_time_since_epoch"

with requests.get(fetch_time_target) as site_time:
    time_options = site_time.json()
```



<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

Getting and Setting time on the Microcontroller

```
import json
import rtc

local_network = "http://192.168.00.123:5000/"
fetch_time_target = local_network + "/api/current_time_since_epoch"

with requests.get(fetch_time_target) as site_time:
    time_options = site_time.json()
```



<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

Getting and Setting time on the Microcontroller

```
import json
import rtc

local_network = "http://192.168.00.123:5000/"
fetch_time_target = local_network + "/api/current_time_since_epoch"

with requests.get(fetch_time_target) as site_time:
    time_options = site_time.json()
```



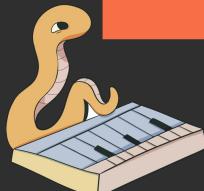
<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

Getting and Setting time on the Microcontroller

```
r = rtc.RTC()  
r.datetime = time.struct_time(time_options['utc_time_tuple'])  
  
t=r.datetime  
print(f"The date is {t.tm_mday}/{t.tm_mon}/{t.tm_year}")  
print(f"The time is {t.tm_hour}:{t.tm_min:02}:{t.tm_sec:02}")
```

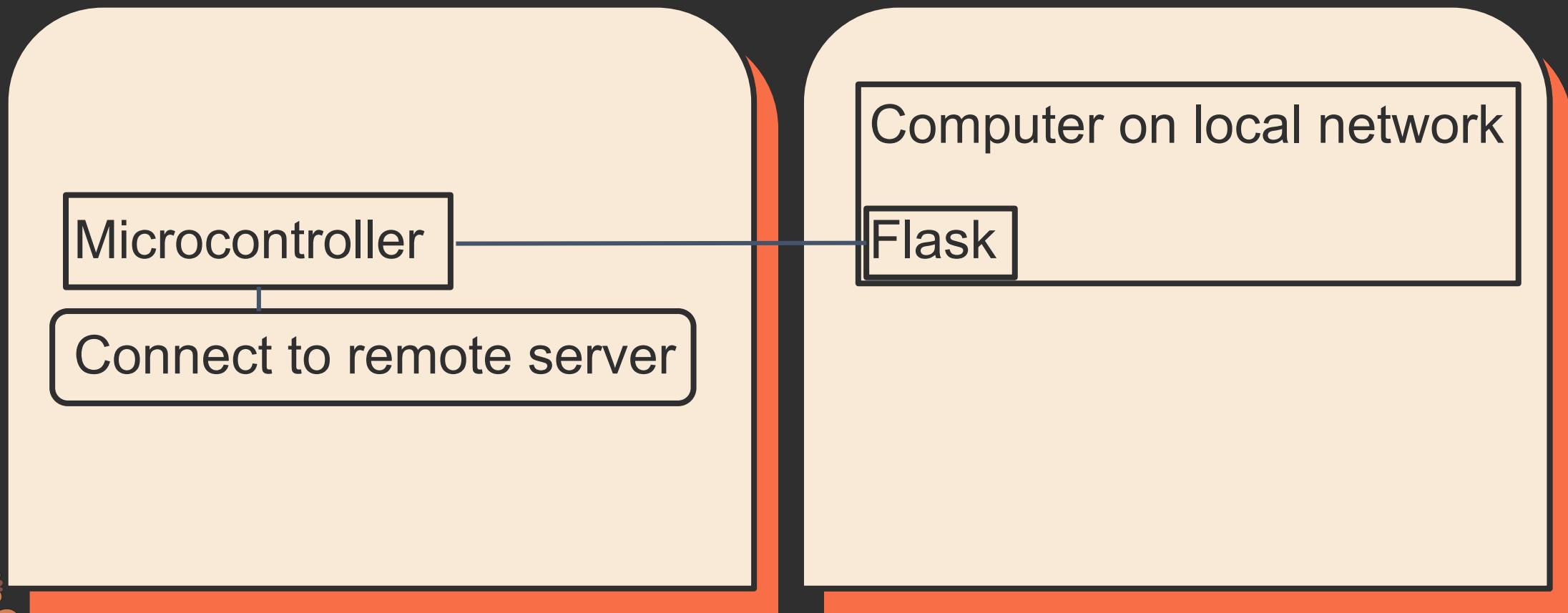
The date is 17/5/2025
The time is 15:27:44



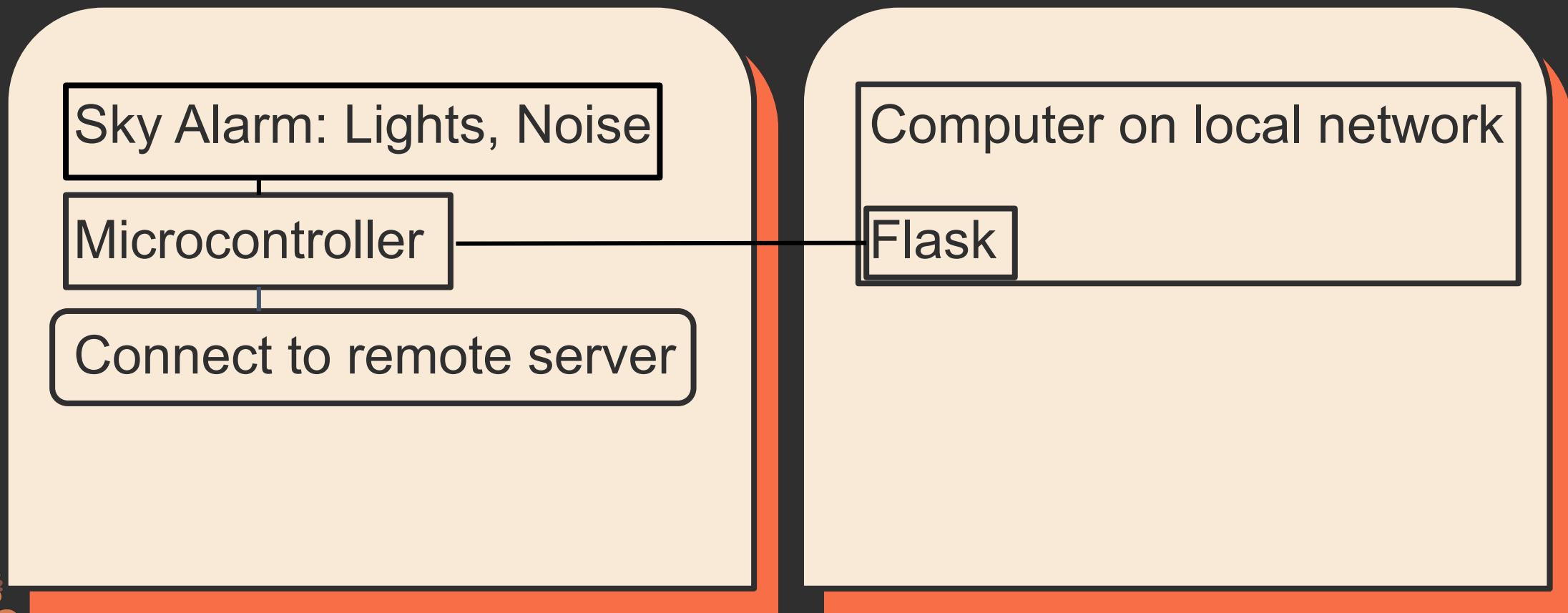
<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

Current Setup



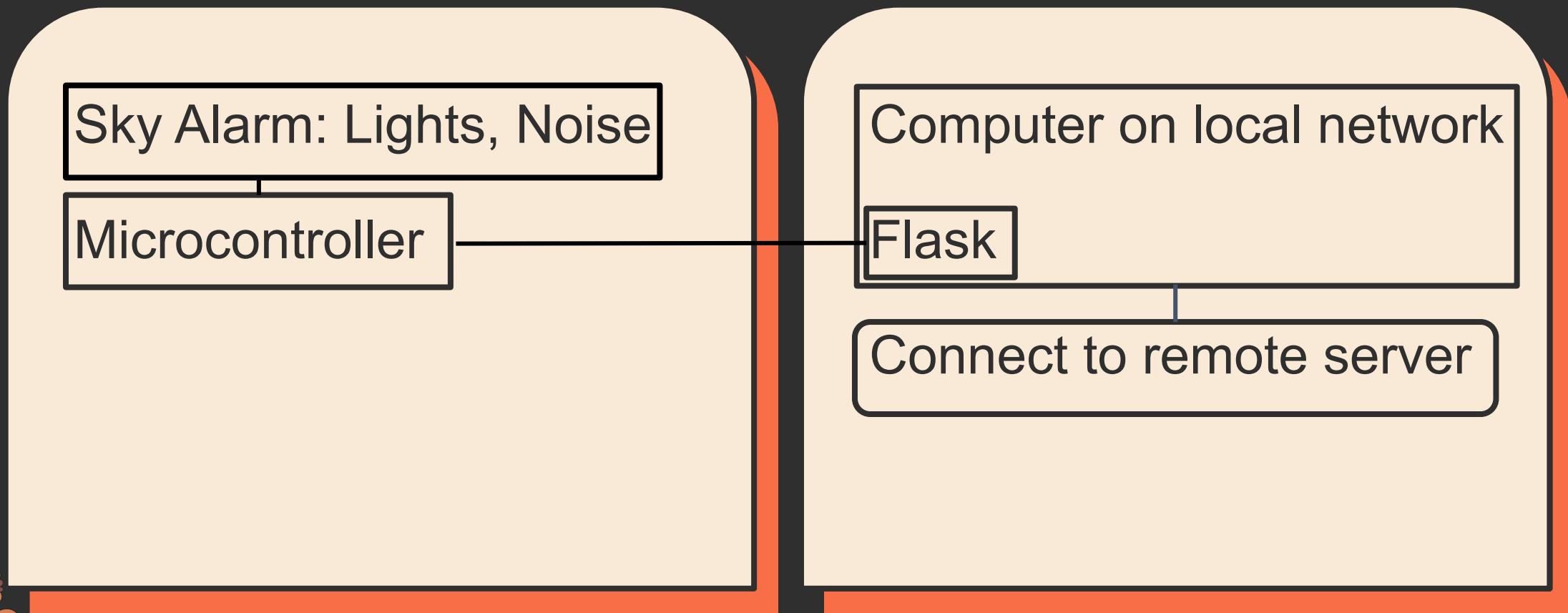
Current Setup



<https://github.com/KeithTheEE/PyConUS2025>

PyCon US
2025

Current Setup



Moving the get code to a python file

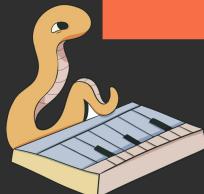
```
import requests

target_url = "https://services.swpc.noaa.gov/text/3-day-forecast.txt"
TEXT_URL = target_url

with requests.get(TEXT_URL) as response:
    print(f" | GET Response: {response.text}")

i = 0
in_table = 0
kp_of_interest = []
text = response.text.split('\n')
for line in text:
    if "NOAA Kp index breakdown" in line:
        in_table = True
    if in_table:
        if i < 6:
            if "UT" in line:
                kp_of_interest.append ( float(line.split()[1]))
        i += 1

kp_peak = max(kp_of_interest)
if kp_peak >= 5:
    cool_space_thing = True
```

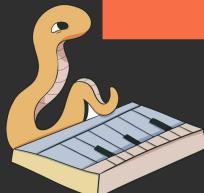


<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

Let Server accept POST requests

```
@app.route('/api/sky_alarm', methods=['GET', 'POST'])
def sky_alarm():
    try:
        if request.method == "POST":
            posted_data = request.json
            with open(timestamp_filename(), 'w') as f:
                json.dump(posted_data, f)
            with open("api/data_log/latest.json") as f:
                json.dump(posted_data, f)
```



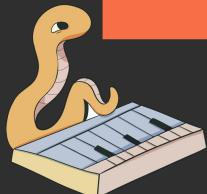
<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

Let Server accept POST requests

```
    elif request.method == 'GET':
        with open("api/data_log/latest.json") as json_fl:
            data = json.load(json_fl)
        return json.dumps(data)

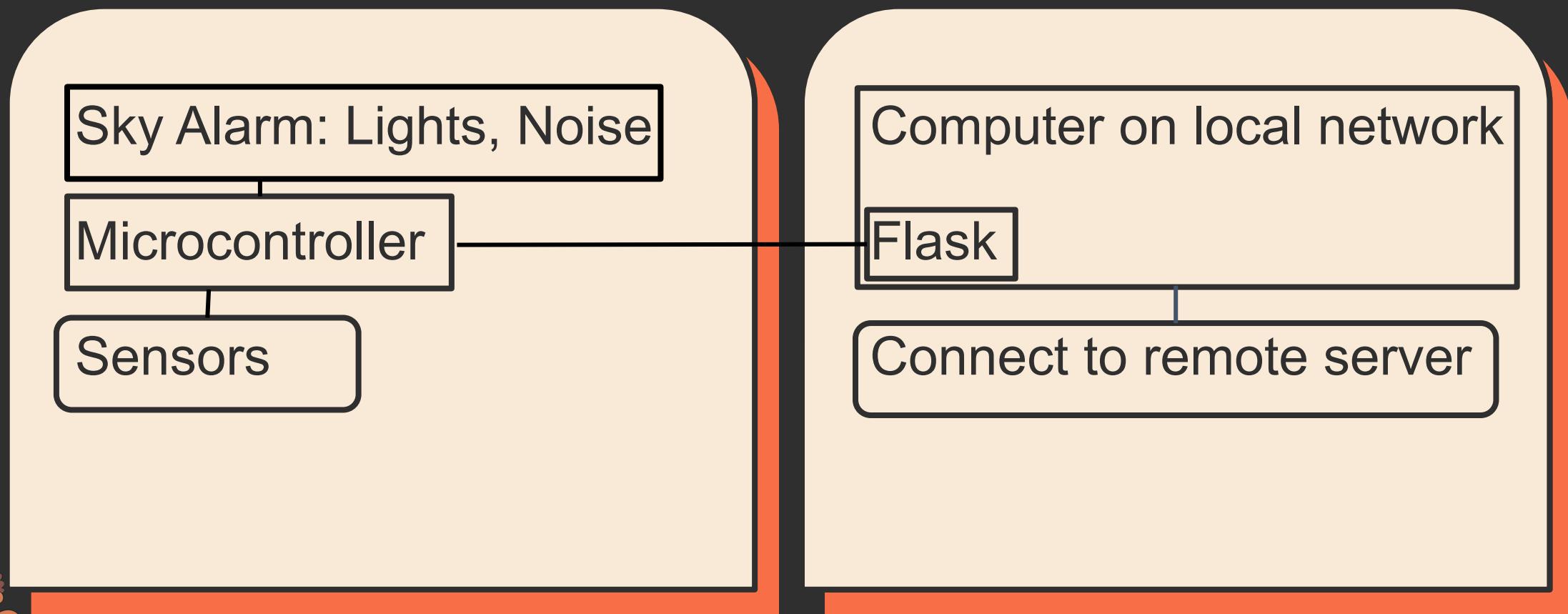
    else:
        pass
```



<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

But, we can STILL do more..



<https://github.com/KeithTheEE/PyConUS2025>

PyCon US
2025

But, we can STILL do more..

i2c Sensors with a STEMMA QT/QWIIC connector

JUMP TO SENSORS SUBCATEGORY:

BIOMETRICS (37) CAPACITIVE (5) CURRENT (8) DISTANCE (42) ENVIRONMENTAL SENSORS (100) FLEX / FORCE (35) IMAGING (60) INFRARED (44) MISC. SENSORS (10)

MOVEMENT (29) NFC (4) RADIATION (0) RFID (25) SENSOR KITS (14) SOUND (6)

SPARKFUN ECOSYSTEM + Sort By: Position ↑ Items per page: 24

QWIIC CONNECTOR? +

BRANDS ⓘ +

PRICE -

\$0.00 - \$400.00

Product	Price	Status
SparkFun Environmental Combo Breakout - ENS160/BME280 (Qwiic)	\$39.95	BACKORDERED
SparkFun Indoor Air Quality Combo Sensor - SCD41, SEN55 (Qwiic)	\$186.48	IN STOCK
SparkFun Tristimulus Color Sensor - OPT4048DTSR (Qwiic)	\$13.50	IN STOCK
SparkFun Optical Tracking Odometry Sensor - PAA5160E1 (Qwiic)	\$84.95	IN STOCK



<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

But, we can STILL do more..

i2c Sensors with a STEMMA QT/QWIIC connector

Weight (7) Accel, Gyro, and Magnetometers (124) Barometric Pressure (21) Biometric (14) Cameras (52) Coin (2) Gas / TVOC / Air Quality (16)

Humidity (36) Light / Color / Photo (35) Liquid / Flow (13) Location / GPS (20) Motion (12) Proximity (50) Radiation / Geiger (2)

Magstripe / Barcode (5) Microphone / Sound (11) Temperature (96) Touch (62)

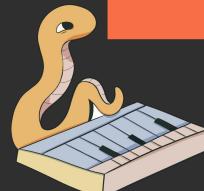


[Adafruit 9-DOF Orientation IMU Fusion Breakout - BNO085 \(BNO080\) - STEMMA QT / Qwiic](#)

PRODUCT ID: 4754

Here it is, the motion sensor you were looking for: the one that just gives you the directly usable information without requiring you to first consult with a PhD to learn the arcane arts of Sensor Fusion. The BNO085 takes the life's work of multiple people who have spent their entire career focused on how to get useful information from direct motion sensor...

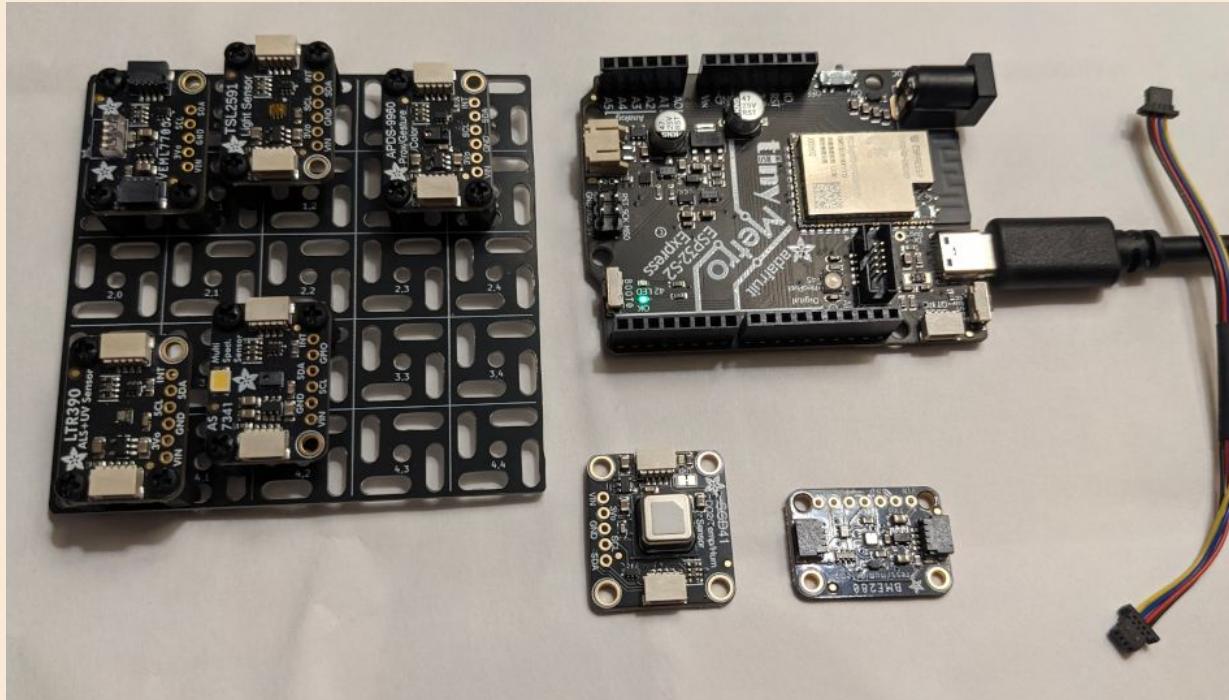
[Add to Cart](#) **\$24.95**
In stock



<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

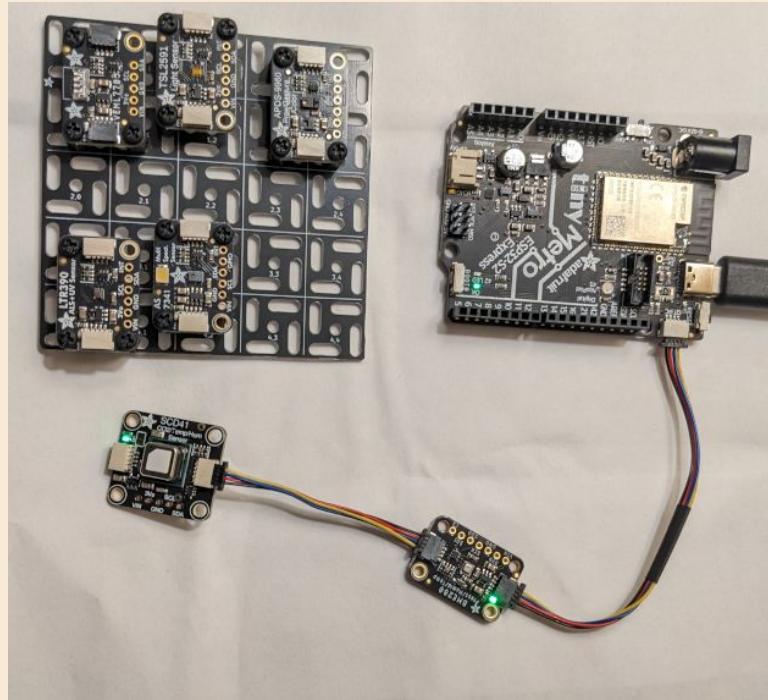
But, we can STILL do more..



<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

But, we can STILL do more..



<https://github.com/KeithTheEE/PyConUS2025>

PyConUS
2025

Adding i2c Sensors

```
from adafruit_bme280 import basic as adafruit_bme280  
  
i2c = board.STEMMA_I2C()  
bme280 = adafruit_bme280.Adafruit_BME280_I2C(i2c)
```



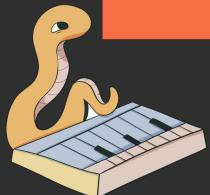
Adding i2c Sensors

```
packet = {}
packet['temp']=bme280.temperature
packet['humidity']=bme280.relative_humidity
packet['pressure']=bme280.pressure

alarm_target = local_network+"/api/sky_alarm"
packet = json.dumps(packet).encode("utf-8")
headers = {'Content-type': 'application/json', 'Accept': 'text/plain'}
r = requests.post(alarm_target, data=packet, headers=headers)
```



You can and Should Build a Sky Alarm





You can and Should Build XKCD's "Sky Alarm" for Cool Space Things and Much More

Keith Murray (@KeithTheEE)