# Can you please pass the data? IoT communication in Micropython

**Sev Leonard**
**PyCascades 2018**
**@gizm0_0.tenforward.social**

# Hello!

# Hello!

- Portland, OR

# Hello!

- Portland, OR
- Trans guy, trans masc slack

# Hello!

- Portland, OR
- Trans guy, trans masc slack
- Oregon Health & Science University

# Hello!

- Portland, OR
- Trans guy, trans masc slack
- Oregon Health & Science University
- He/Him

# Hello!

- Portland, OR
- Trans guy, trans masc slack
- Oregon Health & Science University
- He/Him
- Internet of Cats (PyDX 2016), IoT with Micropython and Friends (PyCon 2017)

# IoT Communication

# IoT Communication

# IoT Communication

# IoT Communication

# IoT Communication

Wow look at the time! My elderly mom from Florida will be here soon. I wonder what the temperature is at the house.

# IoT Communication

# IoT Communication

Hey Nest, whats the temp?

# IoT Communication

# IoT Communication

Its 60 degrees

# IoT Communication

# IoT Communication

Yikes! Can you turn it up to 78 please?

# IoT Communication

# IoT Communication

You betchya

# IoT Communication

Photo: Willamette Valley Vineyards http://www.wvv.com

# Our Internet of Things

# Our Internet of Things

vineyard_network

# Our Internet of Things

vineyard_network

vine1

vine2

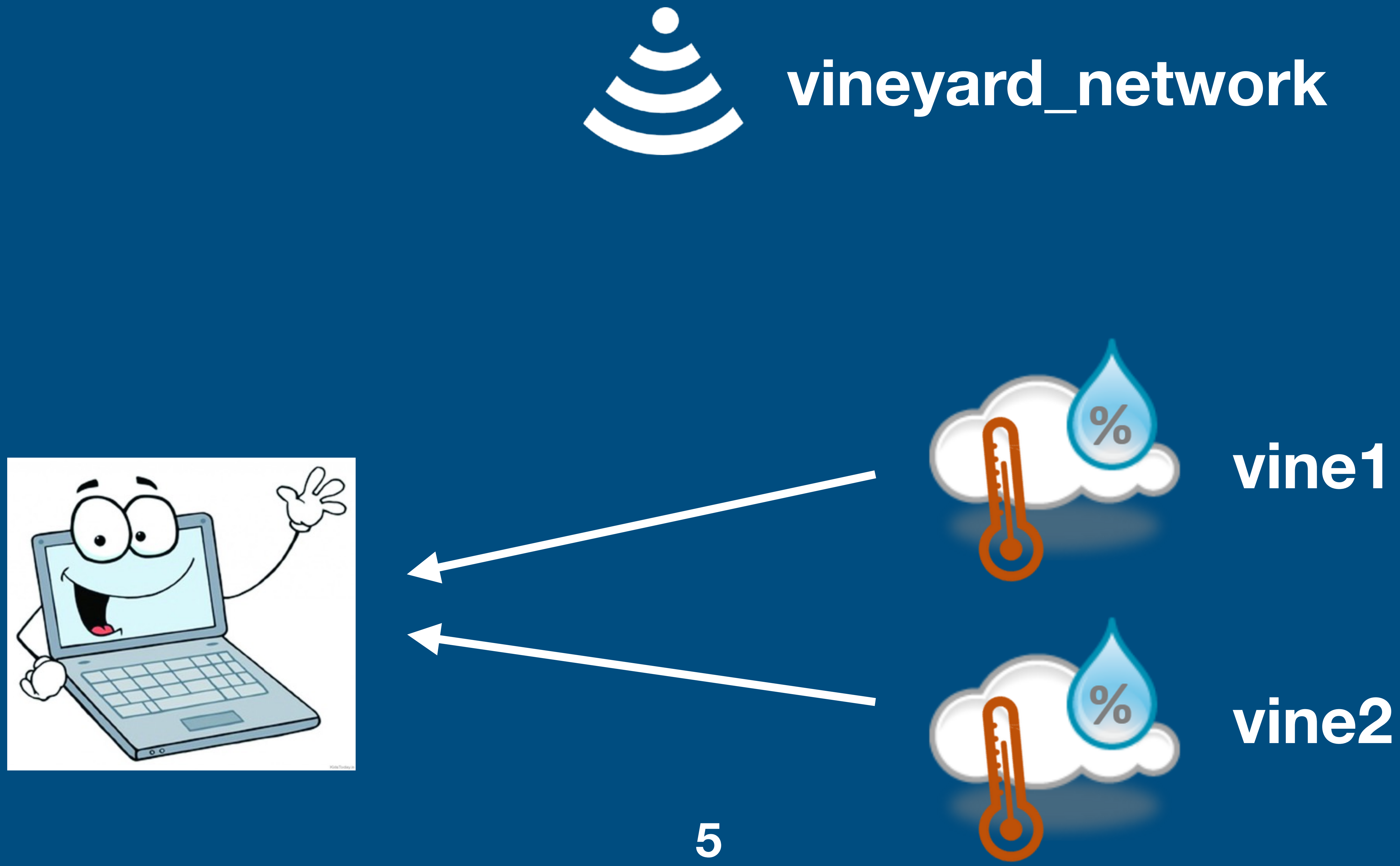
# Our Internet of Things

vineyard_network

vine1

vine2

# Our Internet of Things

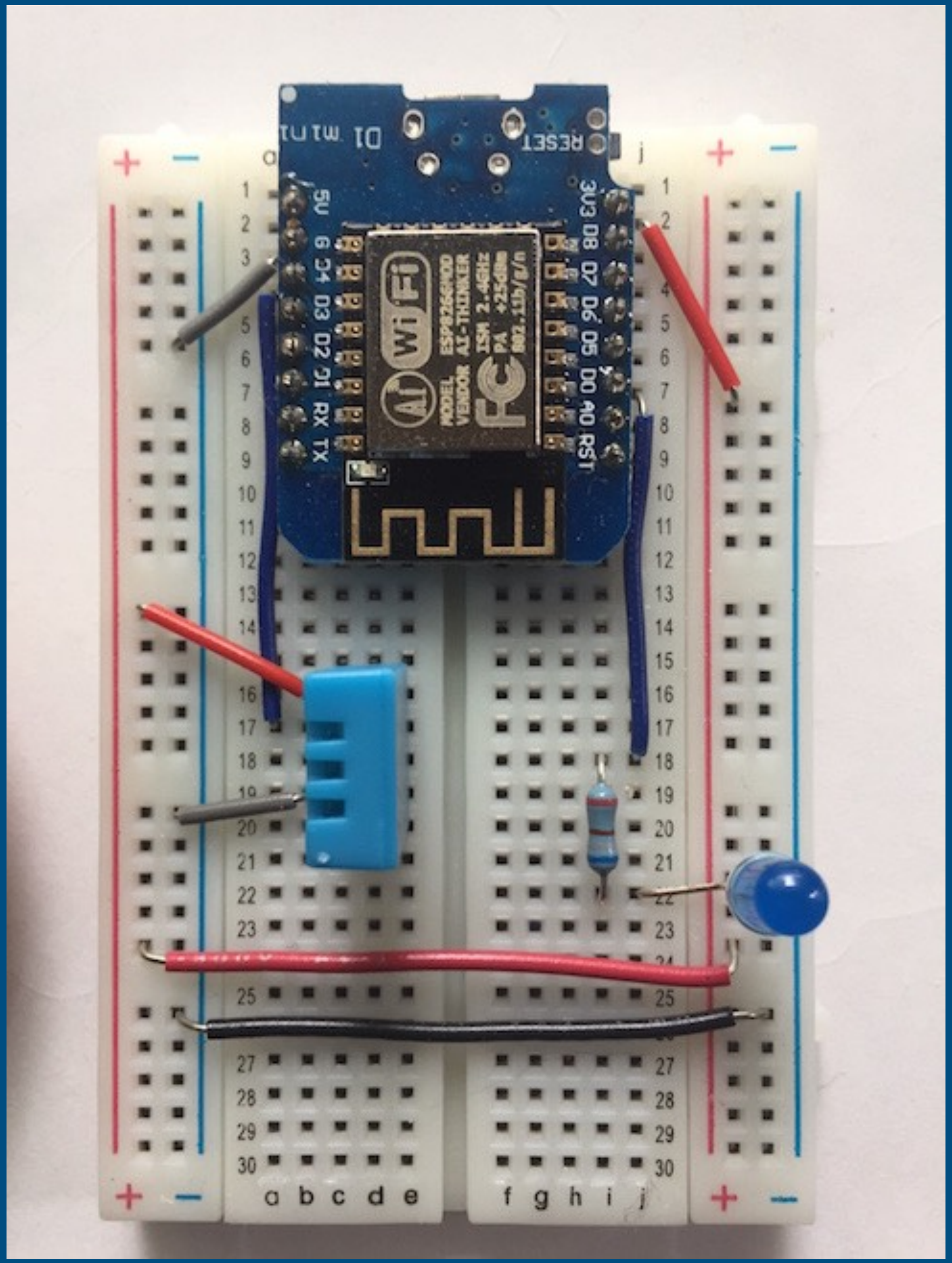


vineyard_network

vine1

vine2

# Vine Sensor



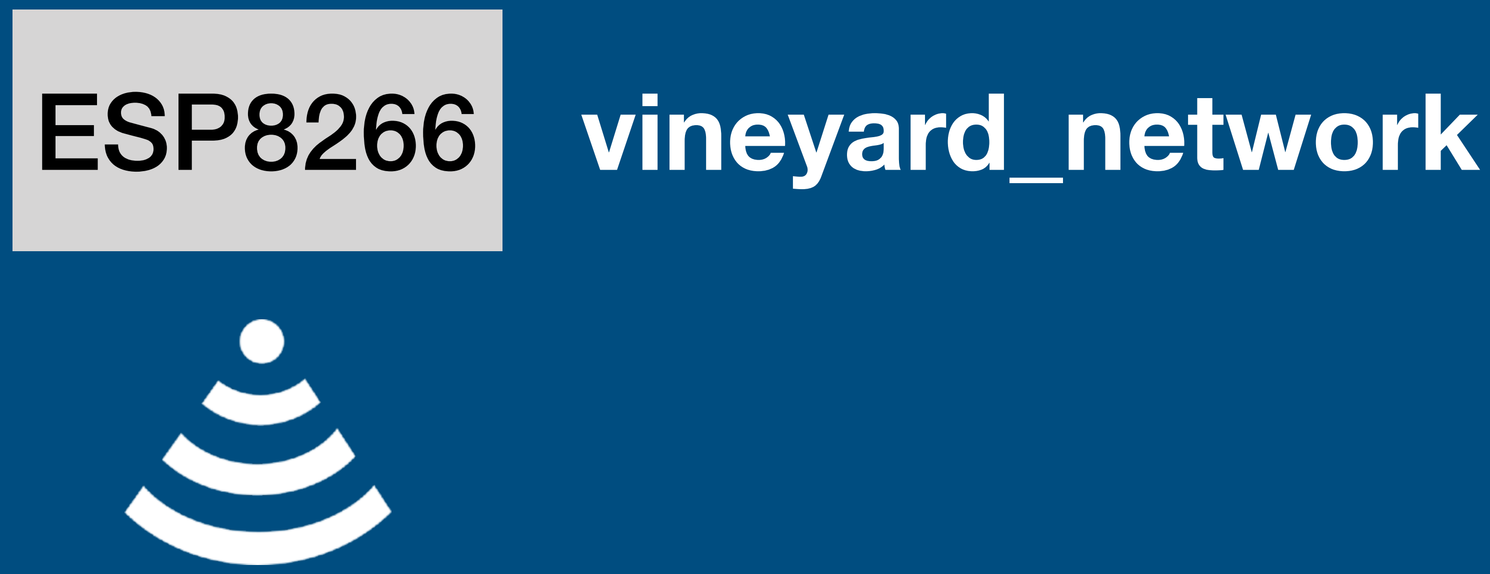
ESP8266

DHT11

LED

# Vineyard Network

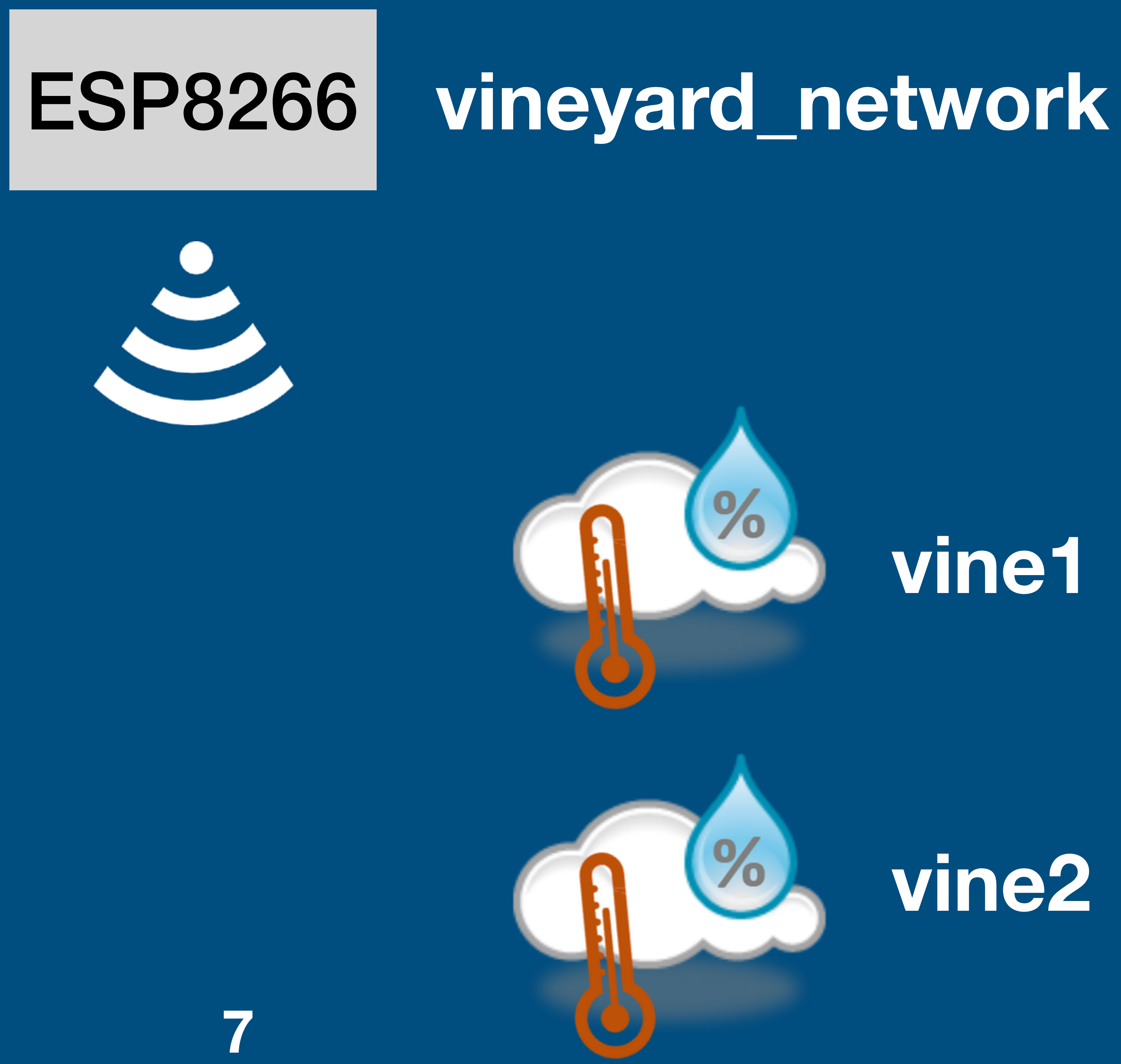
# Vineyard Network



ESP8266

vineyard_network

# Vineyard Network



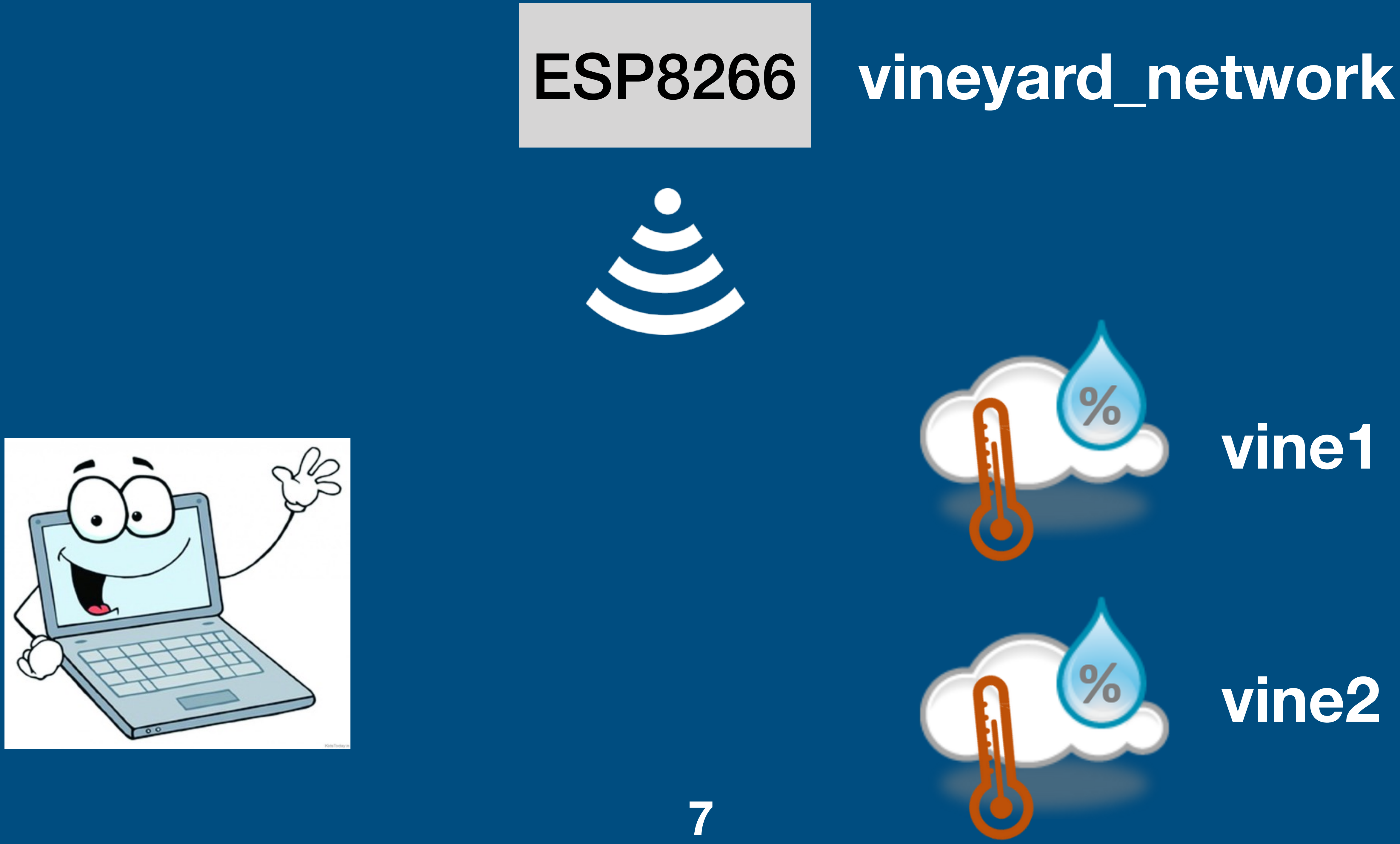
ESP8266

vineyard_network

vine1

vine2

# Vineyard Network



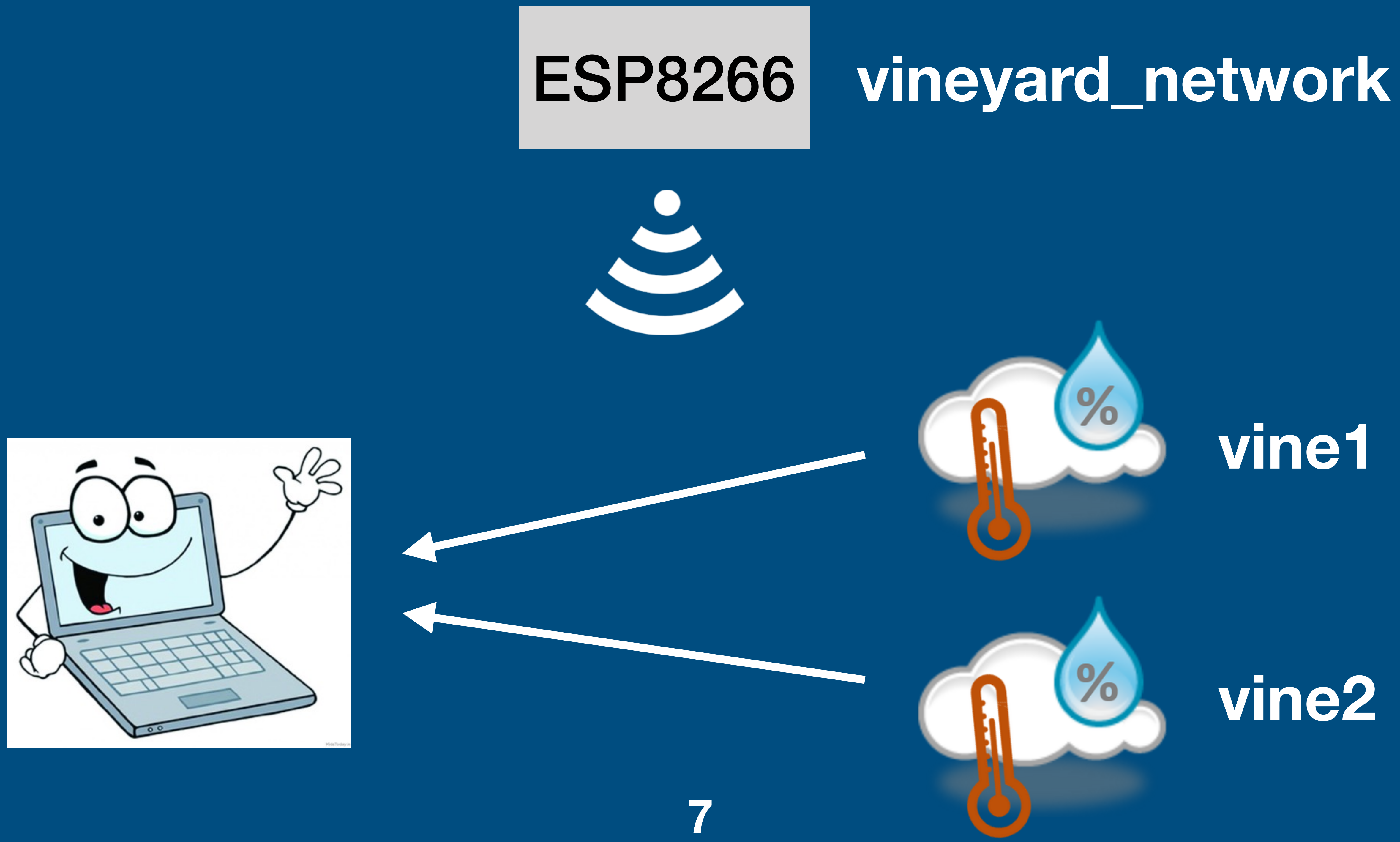
ESP8266

vineyard_network

vine1

vine2

# Vineyard Network



ESP8266

vineyard_network

vine1

vine2

# Setup WiFi

```
import network
ap_if = network.WLAN(network.AP_IF)
ap_if.config(essid=’vineyard_network’)
ap_if.active(True)
```

# Setup WiFi

```
import network
ap_if = network.WLAN(network.AP_IF)
ap_if.config(essid=’vineyard_network’)
ap_if.active(True)
```

ESP8266

# Connect vine sensors

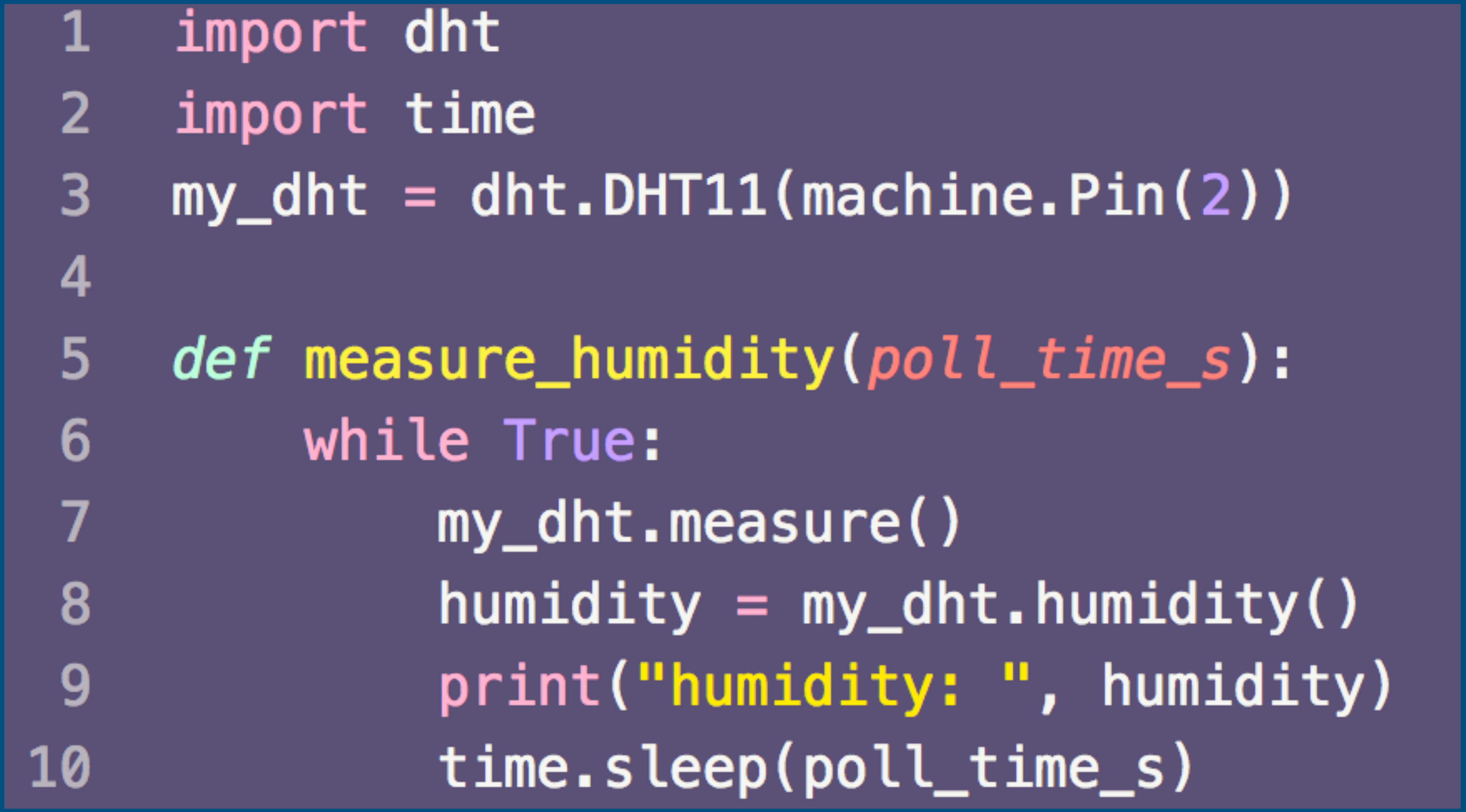
```
import network

sta_if = network.WLAN(network.STA_IF)

sta_if.active(True)

sta_if.connect(‘vineyard_network’,password)
```

# Connect vine sensors

```
import network

sta_if = network.WLAN(network.STA_IF)

sta_if.active(True)

sta_if.connect(‘vineyard_network’,password)
```

vine1

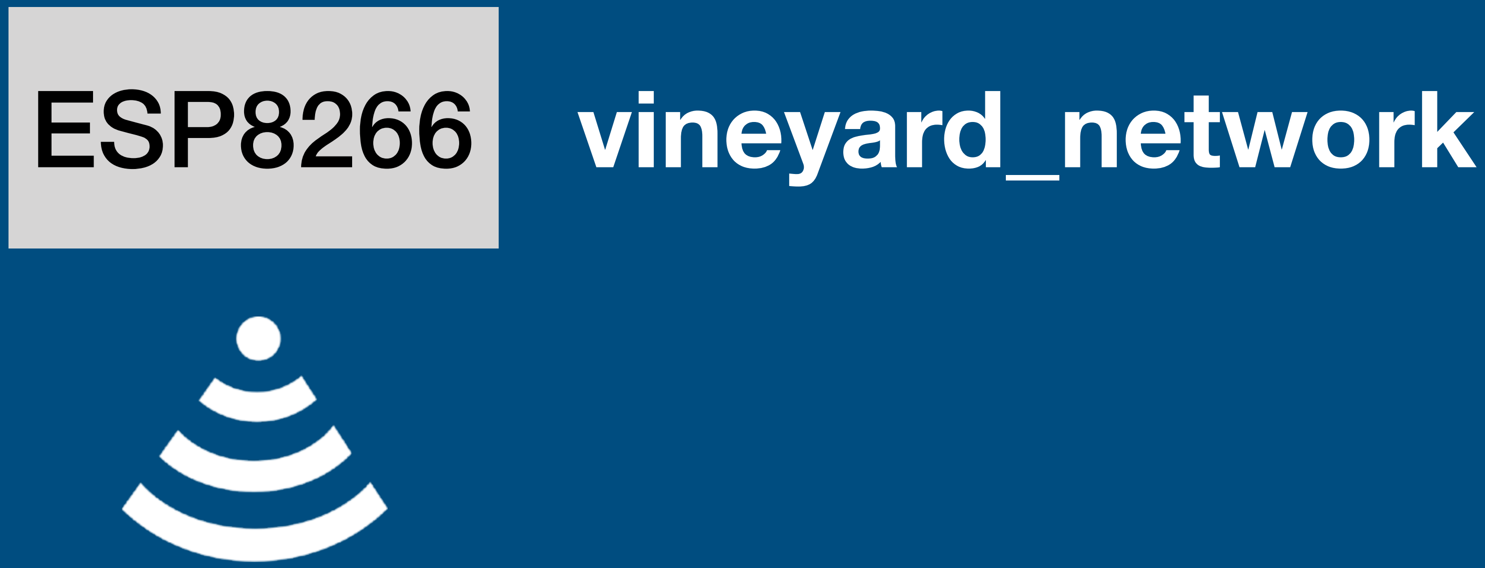
# Measuring humidity



```
import dht
import time
my_dht = dht.DHT11(machine.Pin(2))

def measure_humidity(poll_time_s):
    while True:
        my_dht.measure()
        humidity = my_dht.humidity()
        print("humidity: ", humidity)
        time.sleep(poll_time_s)
```

# Vineyard Network

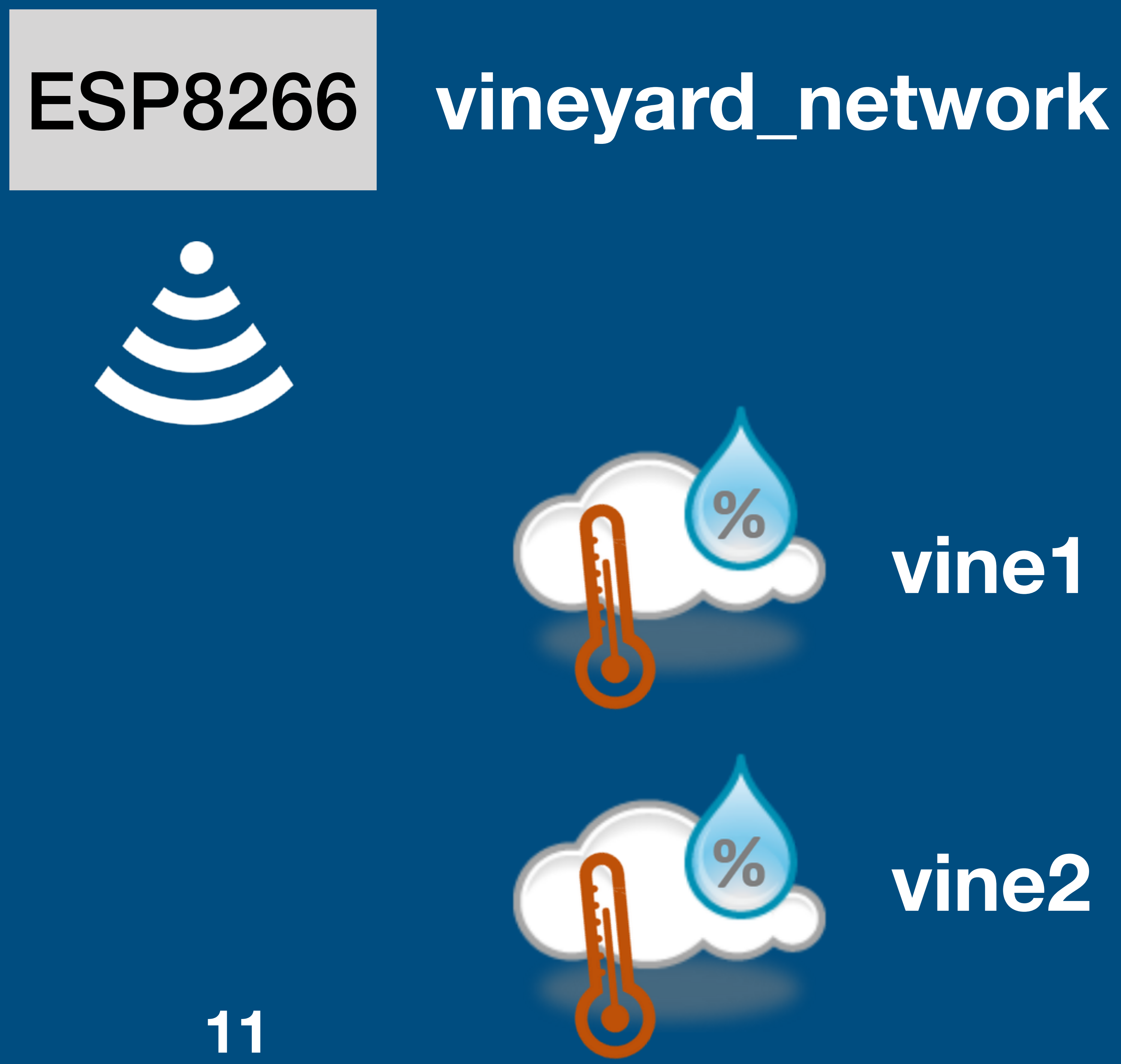
# Vineyard Network



ESP8266

vineyard_network

# Vineyard Network



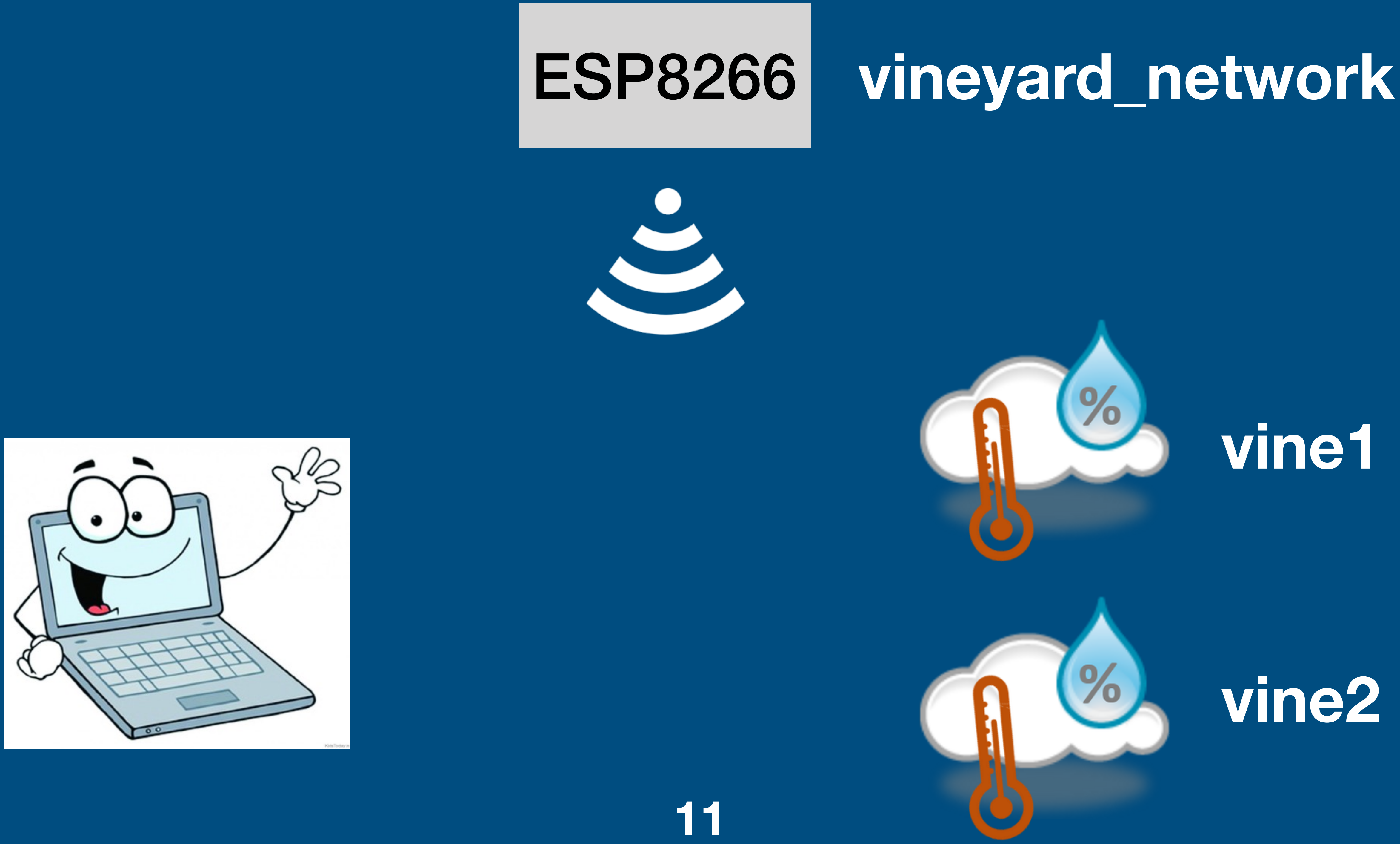
ESP8266

vineyard_network

vine1

vine2

# Vineyard Network



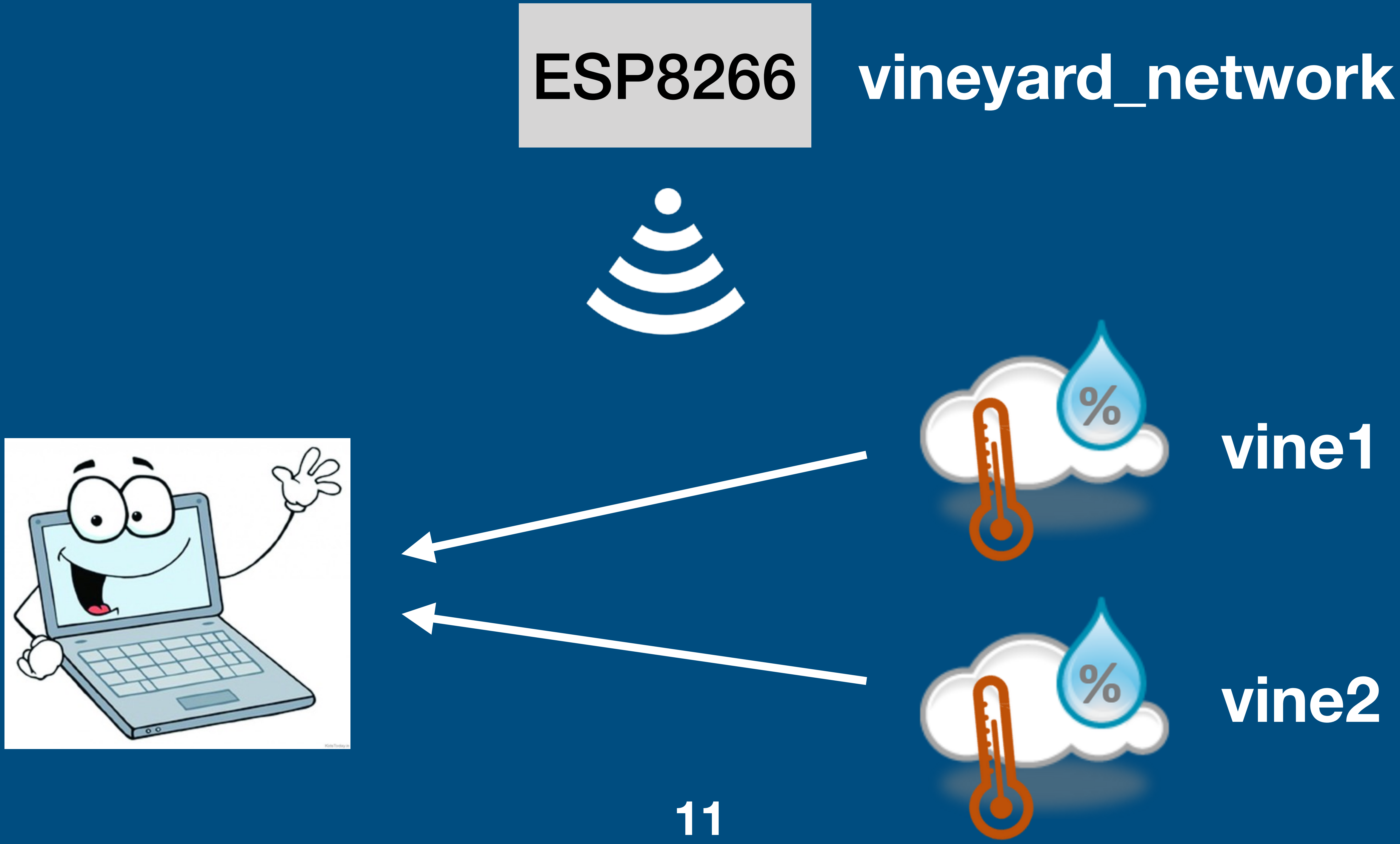
ESP8266

vineyard_network

vine1

vine2

# Vineyard Network



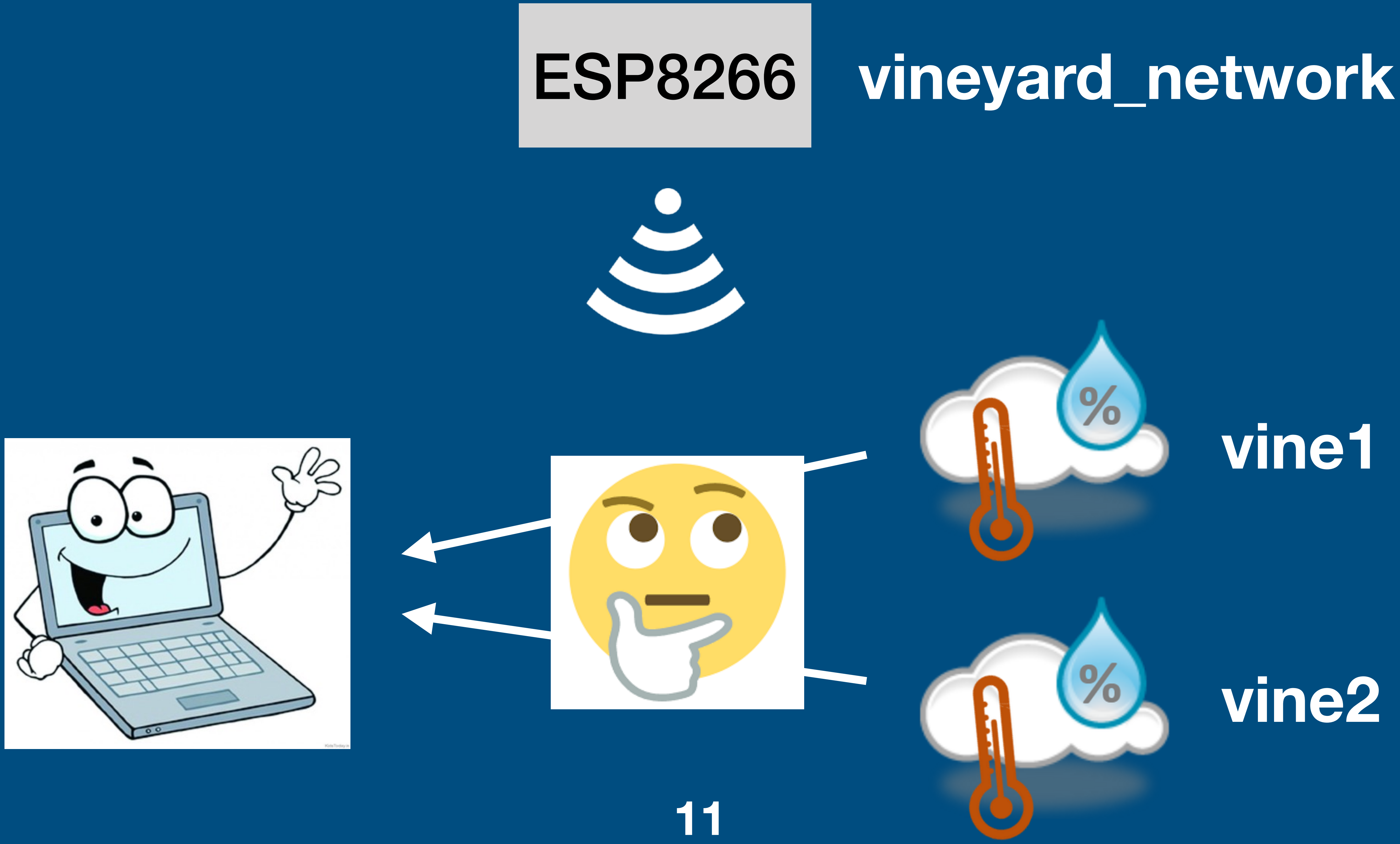
ESP8266

vineyard_network

vine1

vine2

# Vineyard Network



ESP8266

vineyard_network

vine1

vine2

# MQTT

**Message Queuing Telemetry Transport!**

- **Light weight, pub/sub model w/ broker**

# MQTT

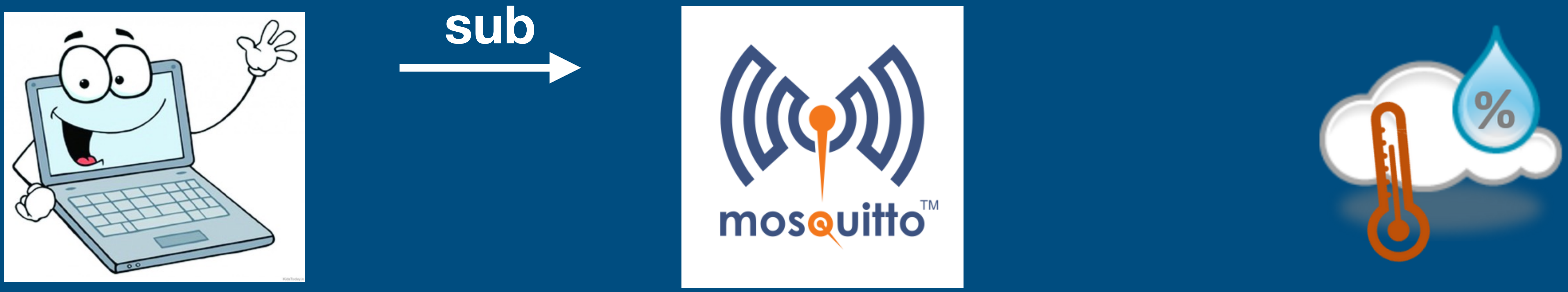
**Message Queuing Telemetry Transport!**

- **Light weight, pub/sub model w/ broker**

# MQTT

**Message Queuing Telemetry Transport!**

- Light weight, pub/sub model w/ broker

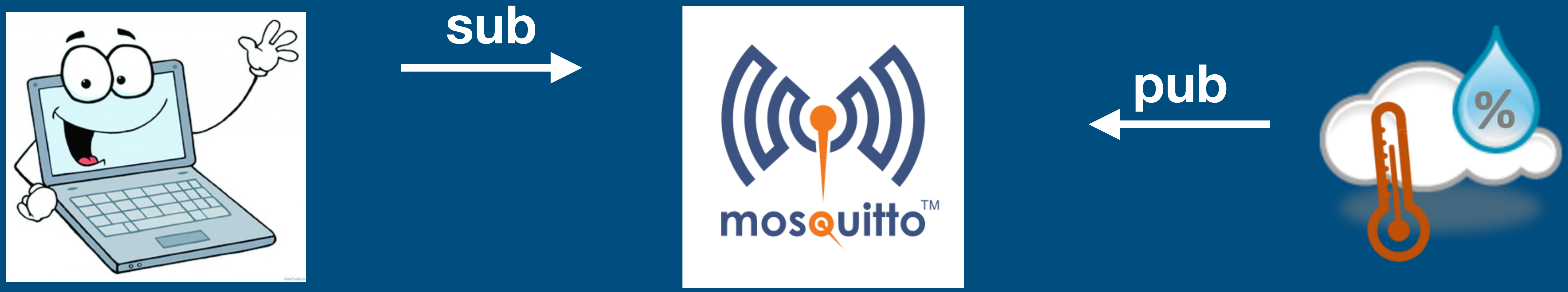


sub

# MQTT

## Message Queuing Telemetry Transport!

- Light weight, pub/sub model w/ broker



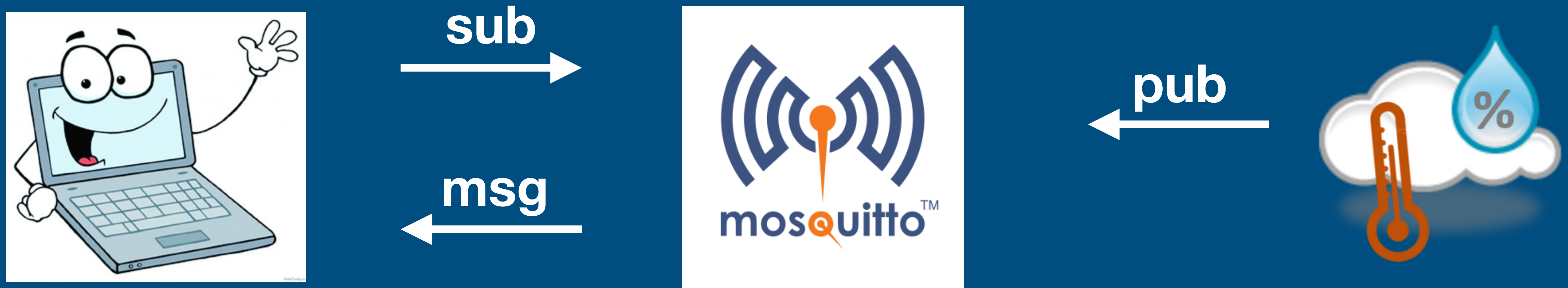
sub

pub

# MQTT

**Message Queuing Telemetry Transport!**

- **Light weight, pub/sub model w/ broker**



sub

msg

pub

# MQTT - Topics

# MQTT - Topics

- vineyard/

# MQTT - Topics

- vineyard/
    - vineyard/humidity

# MQTT - Topics

- vineyard/
  - vineyard/humidity
    - vineyard/humidity/vine1

# MQTT - Topics

- vineyard/
    - vineyard/humidity
        - vineyard/humidity/vine1
    - vineyard/temp/#

# MQTT - Topics

- vineyard/
  - vineyard/humidity
    - vineyard/humidity/vine1
  - vineyard/temp/#
  - vineyard/+/vine1

# MQTT Pub in Micropython

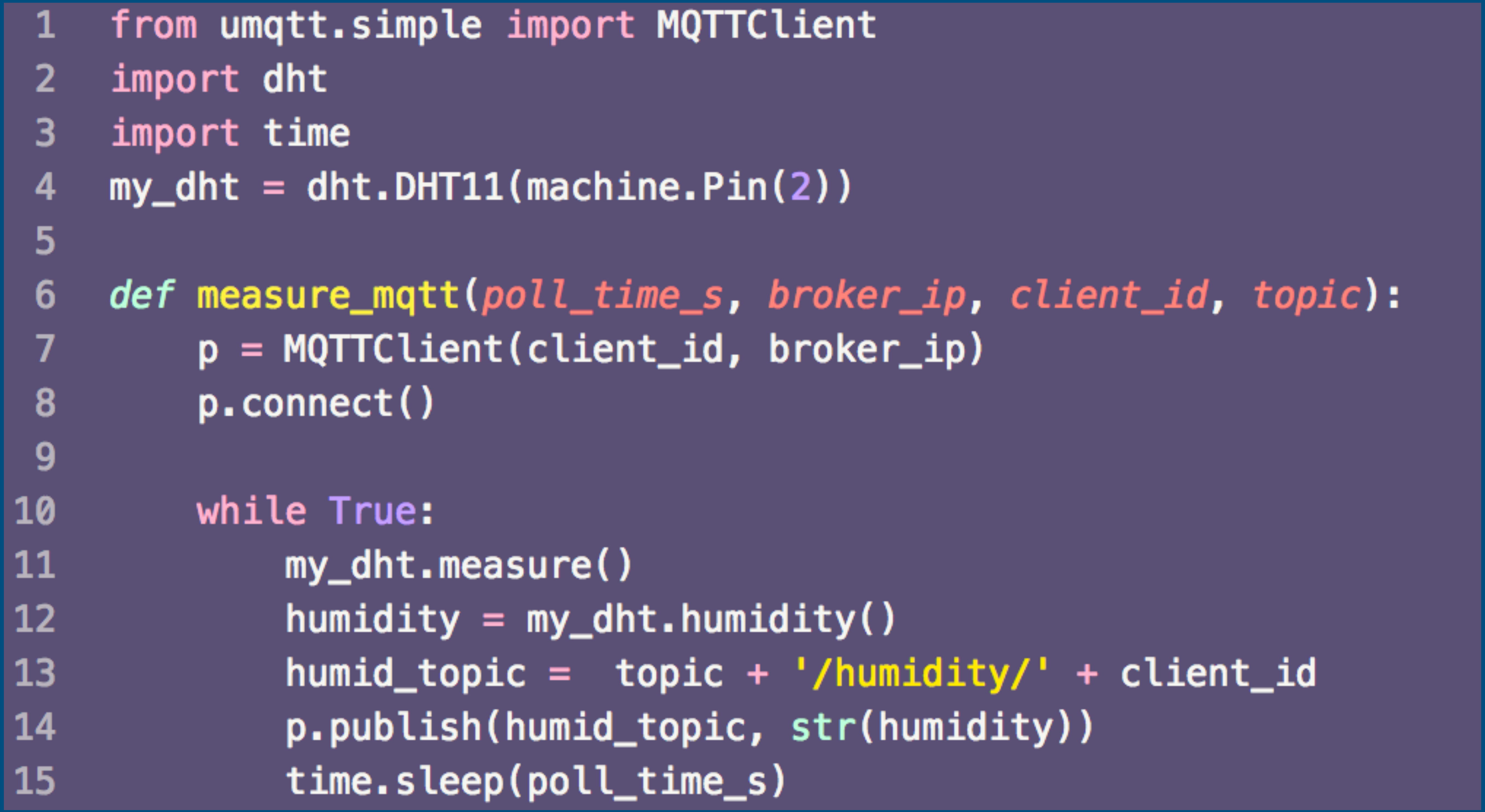
```
from umqtt.simple import MQTTClient
p = MQTTClient( client_id, broker_ip )
p.connect()
p.publish( vineyard/temp/vine1, 15 )
```

# MQTT Sub in Micropython

```
s = MQTTClient( client_id, broker_ip )
s.connect()
def cb(topic, message):
        print(topic + ": " + message)
```

# MQTT Sub in Micropython

```
s.set_callback(cb)
s.subscribe(”vineyard/temp/#”)
while 1:
    s.wait_msg()
```



```python
from umqtt.simple import MQTTClient
import dht
import time
my_dht = dht.DHT11(machine.Pin(2))

def measure_mqtt(poll_time_s, broker_ip, client_id, topic):
    p = MQTTClient(client_id, broker_ip)
    p.connect()

    while True:
        my_dht.measure()
        humidity = my_dht.humidity()
        humid_topic =  topic + '/humidity/' + client_id
        p.publish(humid_topic, str(humidity))
        time.sleep(poll_time_s)
```

# Demo!

```
from mqtt_measure import *
measure_mqtt(4, "192.168.4.2", "vine1", "vineyard")

from mqtt_measure import *
measure_mqtt(4, "192.168.4.2", "vine2", "vineyard")
```

# Summary

# Summary

- **Create WLAN access point**

# Summary

- Create WLAN access point
- Connect sensors to WLAN

# Summary

- Create WLAN access point
- Connect sensors to WLAN
- Measure humidity and temperature

# Summary

- Create WLAN access point
- Connect sensors to WLAN
- Measure humidity and temperature
- MQTT for message relay

# Thanks!

- References in backup
- gizm0_0@tenforward.social
- github.com/gizm00/pycascades_2018
- sev@thedatascout.com

# Backup

# Demo setup

- connect laptop to vineyard_network
- ipconfig | grep inet
- pico /usr/local/Cellar/mosquitto/1.4.11_2/etc/mosquitto/mosquitto.conf
- brew services restart mosquitto
- mosquitto_sub -h 192.168.4.2 -t vineyard/#
- nav webrepl to 192.168.4.3 and 4.4 to get to vine sensors
- from mqtt_measure import *
- measure_mqtt(4, "192.168.4.2", "vine1", "vineyard")
- from mqtt_measure import *
- measure_mqtt(4, "192.168.4.2", "vine2", "vineyard")

# MQTT Spy Setup

- Goto Connections -> Manage Connections
- Update Server URI if needed
- Click “Close and reopen existing connections”
- Under subscriptions & received messages click New tab
- Add vineyard/temp/#
- Add vineyard/humidity/#
- To show graphs:
  - Right click received messages and go to Charts
  - Show all browsed topics

# References

- https://en.wikipedia.org/wiki/MQTT
- https://mosquitto.org
- https://micropython.org
- https://www.hivemq.com/blog/how-to-get-started-with-mqtt

# Agenda

# Agenda

- IoT IRL

# Agenda

- IoT IRL
- ESP8266

# Agenda

- IoT IRL
- ESP8266
- Micropython

# Agenda

- IoT IRL
- ESP8266
- Micropython
- MQTT

# Agenda

- IoT IRL
- ESP8266
- Micropython
- MQTT
- Sending MQTT messages

# Agenda

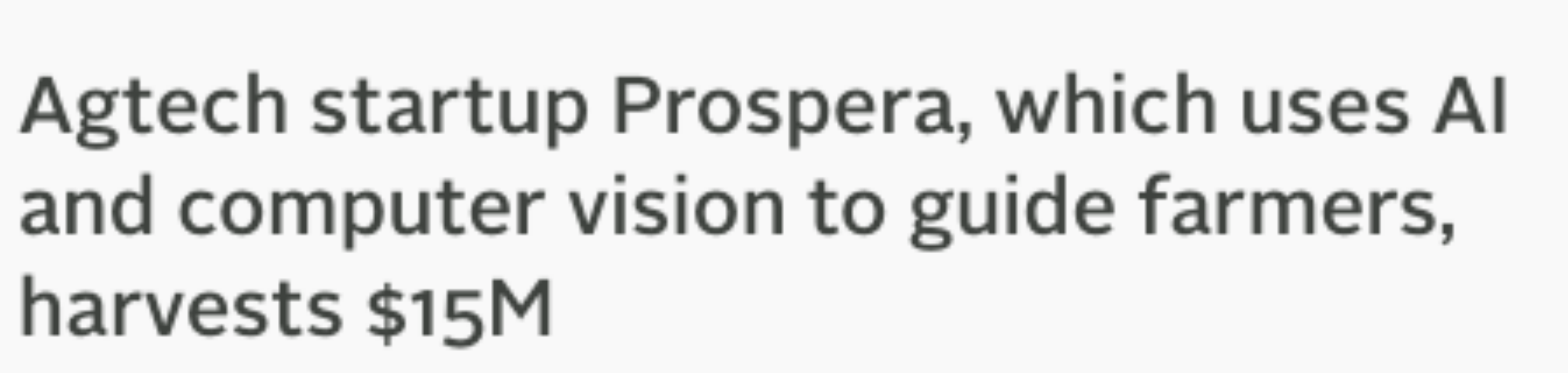
- IoT IRL
- ESP8266
- Micropython
- MQTT
- Sending MQTT messages
- Demo!

# Agricultural Technology

TechCrunch

# Agricultural Technology

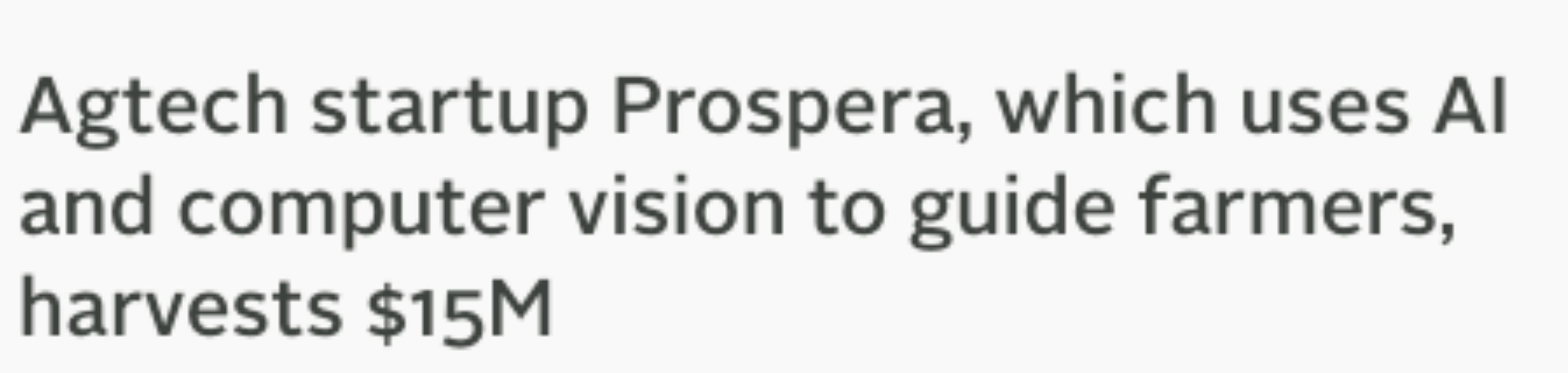
TechCrunch

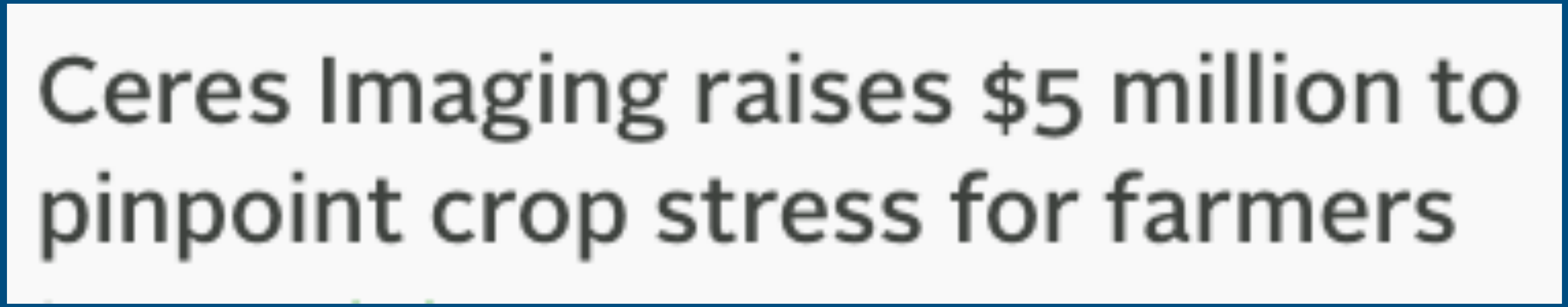


Agtech startup Prospera, which uses AI and computer vision to guide farmers, harvests $15M

# Agricultural Technology

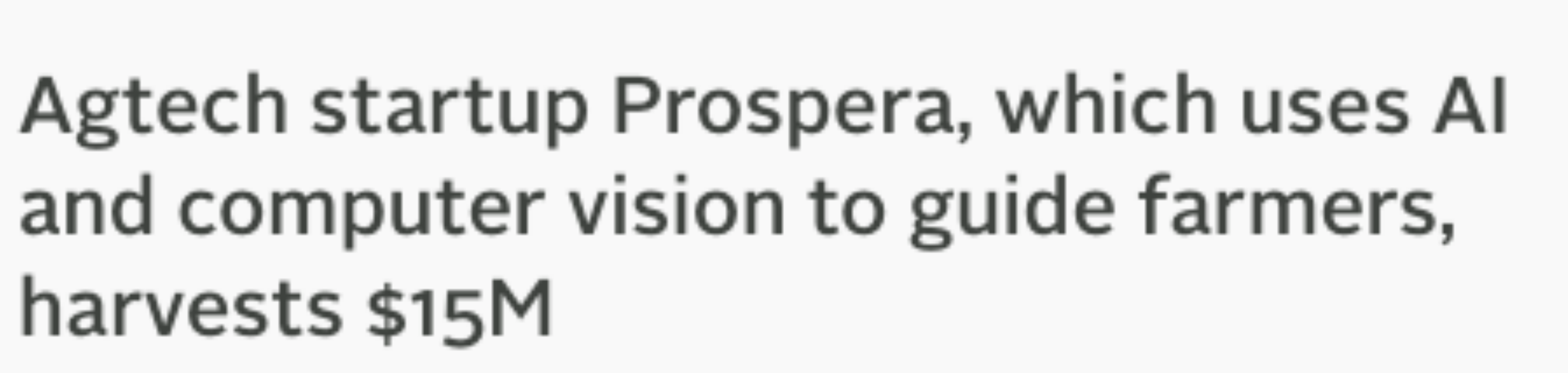
TechCrunch


Agtech startup Prospera, which uses AI and computer vision to guide farmers, harvests $15M

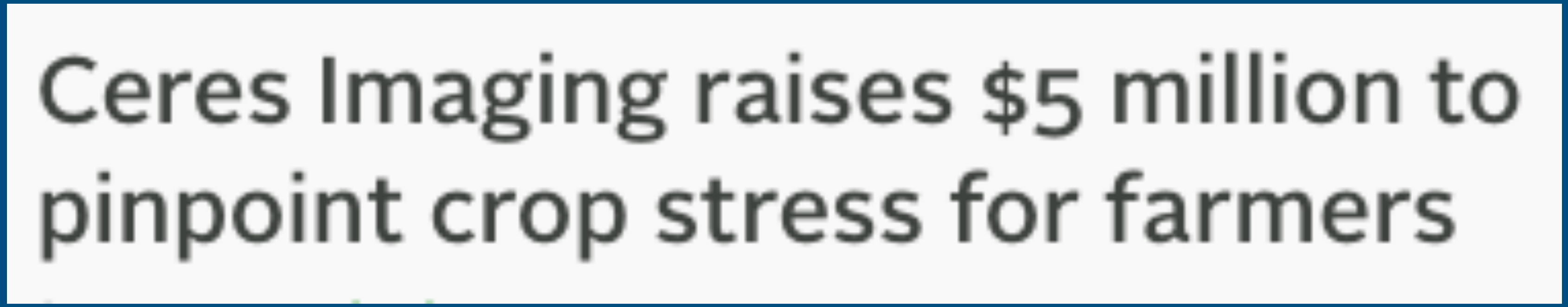

Ceres Imaging raises $5 million to pinpoint crop stress for farmers

# Agricultural Technology

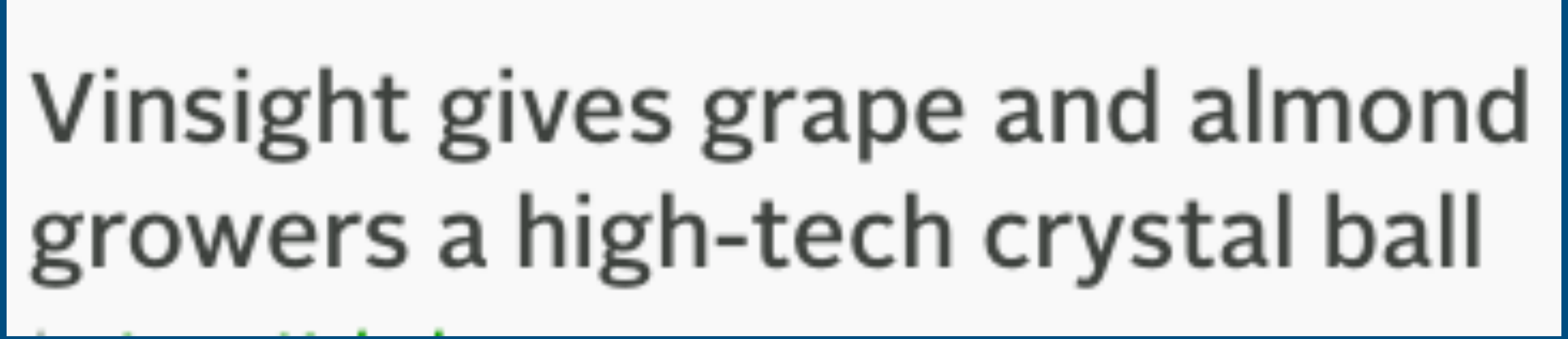
TechCrunch



Agtech startup Prospera, which uses AI and computer vision to guide farmers, harvests $15M

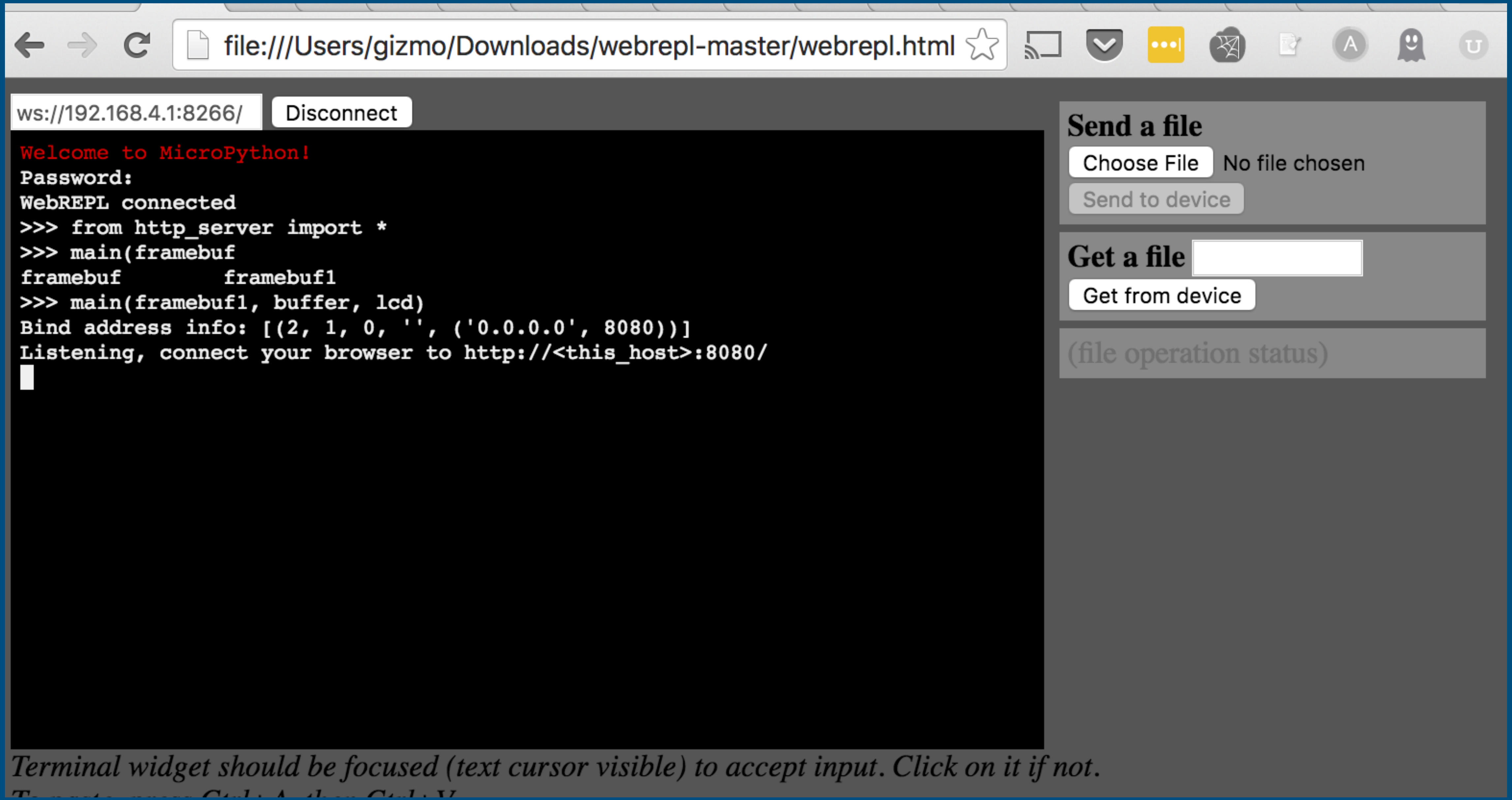


Ceres Imaging raises $5 million to pinpoint crop stress for farmers



Vinsight gives grape and almond growers a high-tech crystal ball

# WebREPL



```
Welcome to MicroPython!
Password:
WebREPL connected
>>> from http_server import *
>>> main(framebuf
framebuf        framebuf1
>>> main(framebuf1, buffer, lcd)
Bind address info: [(2, 1, 0, '', ('0.0.0.0', 8080))]
Listening, connect your browser to http://<this_host>:8080/
```

*Terminal widget should be focused (text cursor visible) to accept input. Click on it if not.*

*To paste, press Ctrl+A, then Ctrl+V*

# Micropython

A. Like regular Python, but small and hard to read X

B. A version of Python optimized for use on microcontrollers
Funded via Kickstarter

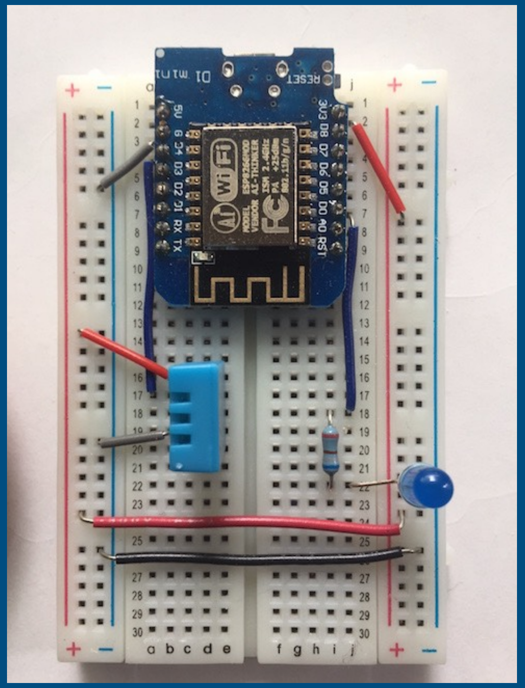
C. All of the above

D. Some of the above

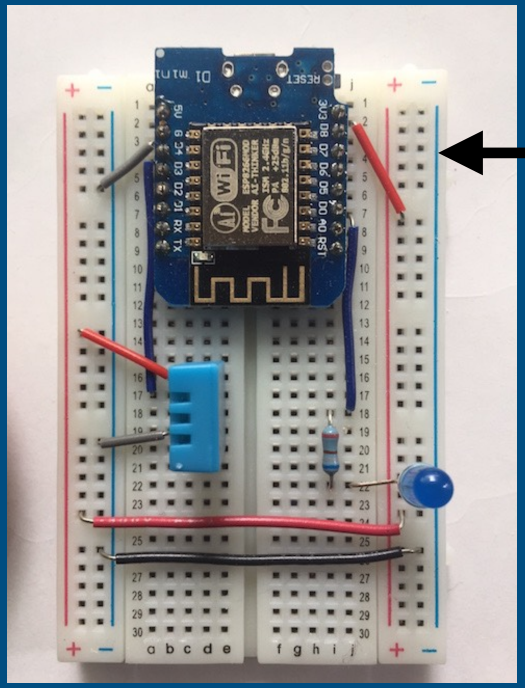
E. Some of the  the above!

micropython.org

# ESP 8266 Microcontroller

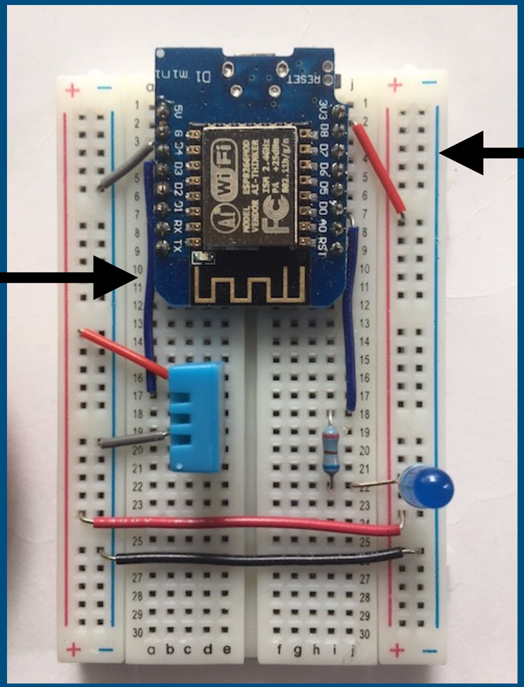


# ESP 8266 Microcontroller



ESP8266
(WeMos D1)

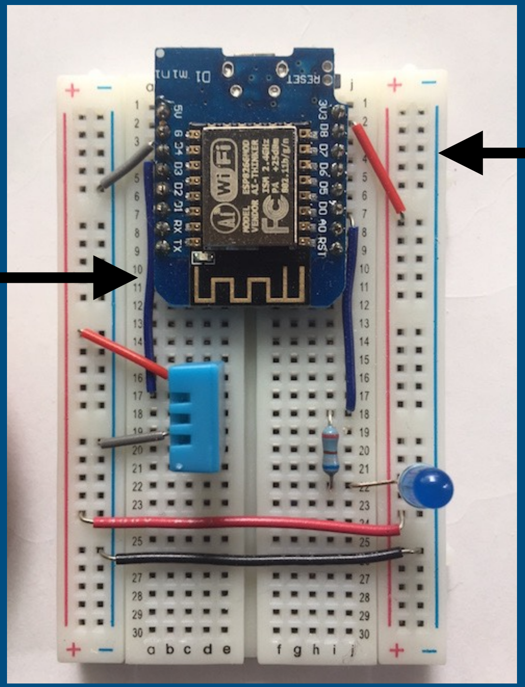
# ESP 8266 Microcontroller



Antenna

ESP8266
(WeMos D1)

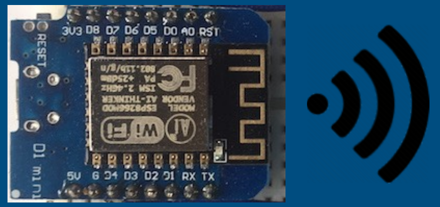
# ESP 8266 Microcontroller



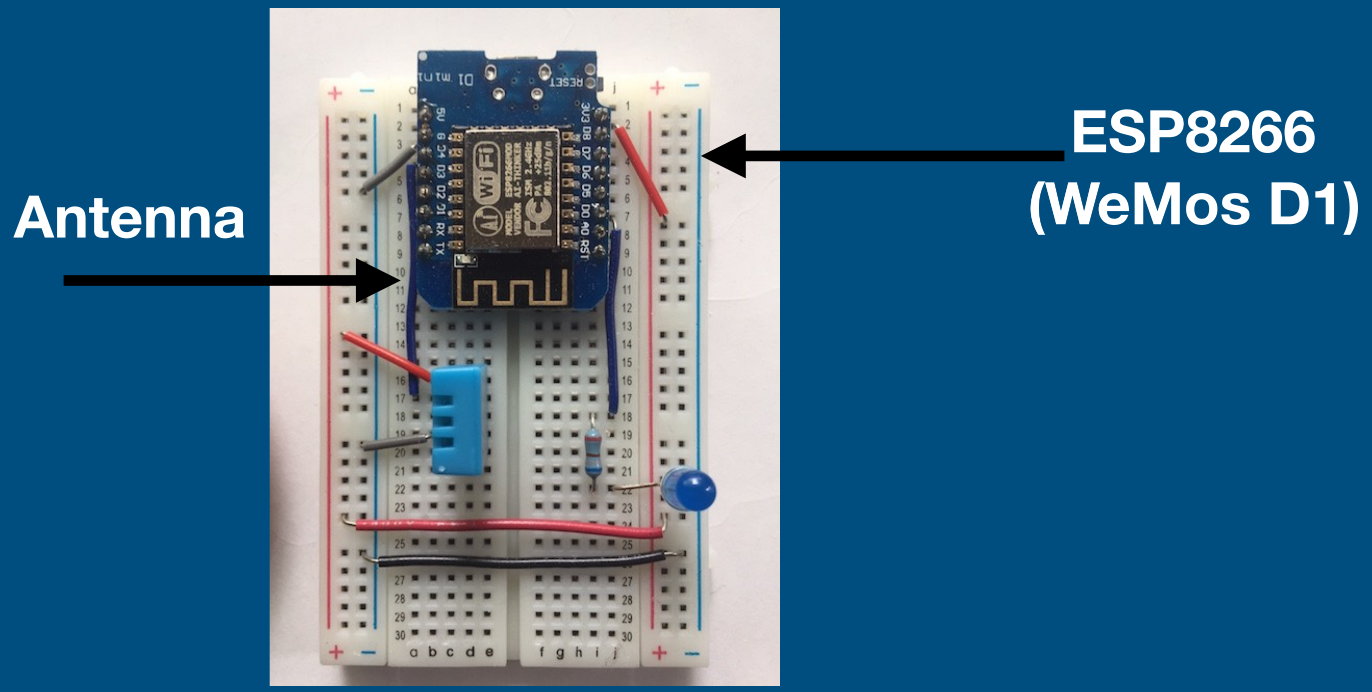
Antenna

ESP8266
(WeMos D1)

Access Point

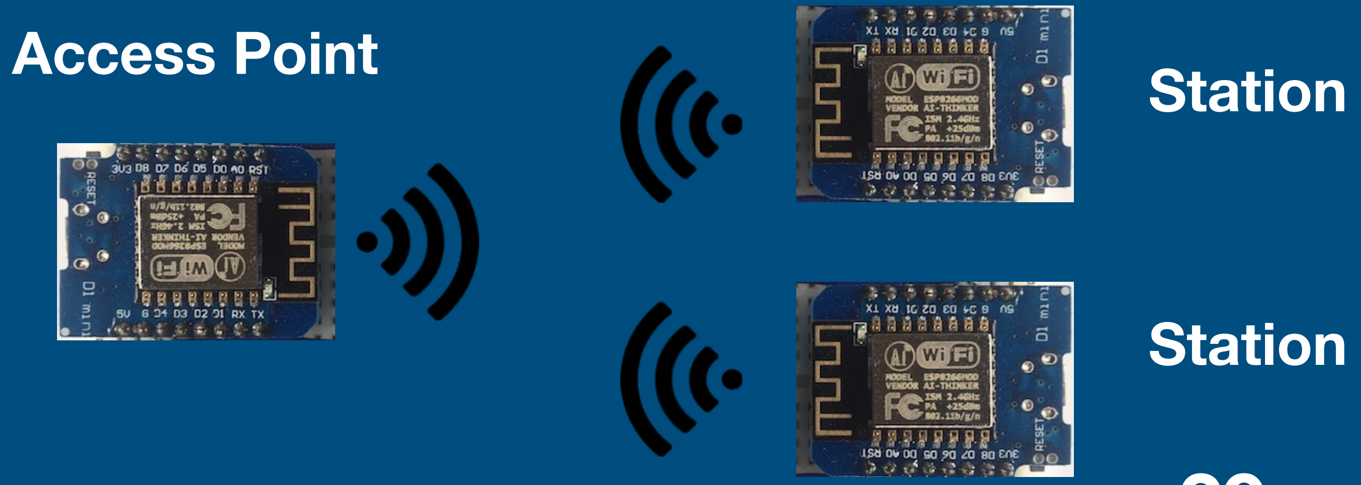


# ESP 8266 Microcontroller



Antenna

ESP8266
(WeMos D1)

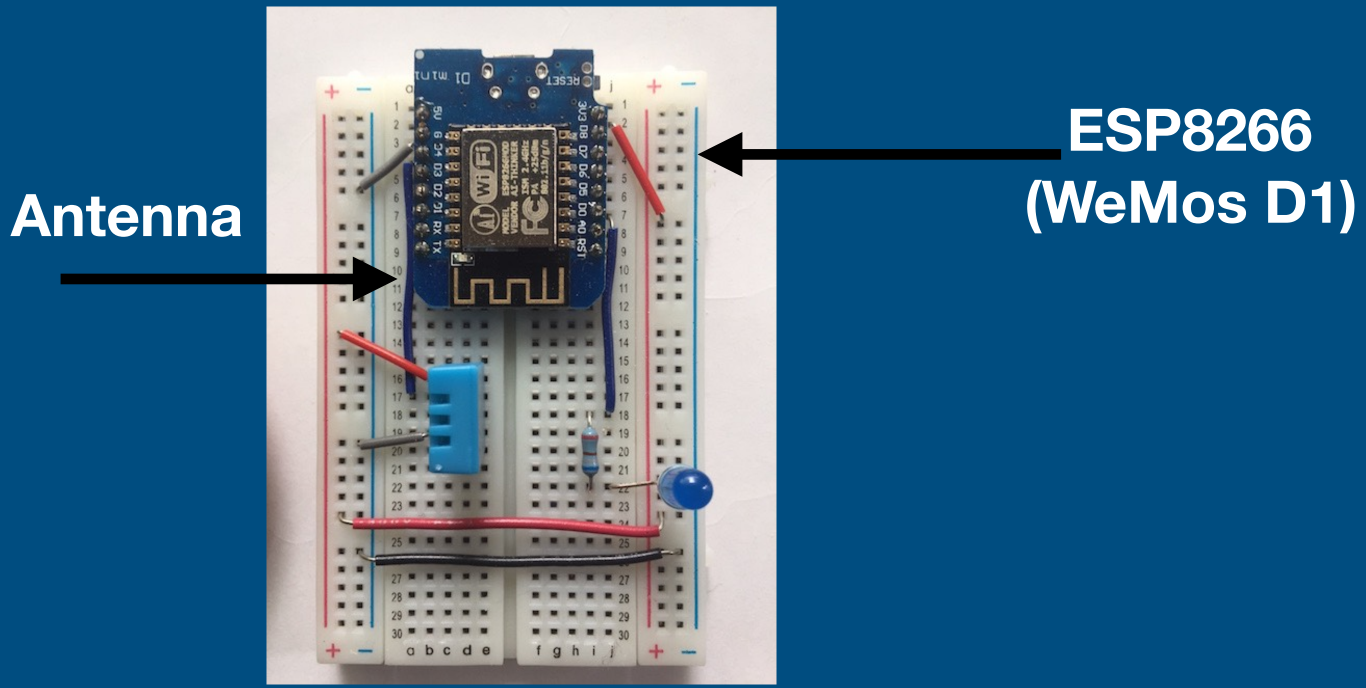


Access Point

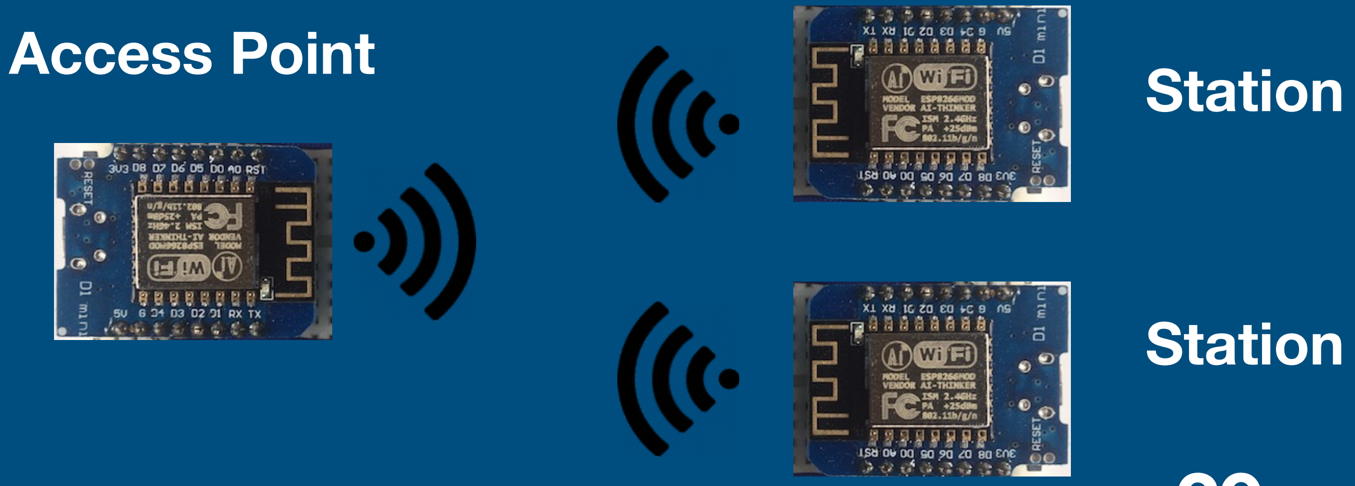
Station

Station

# ESP 8266 Microcontroller

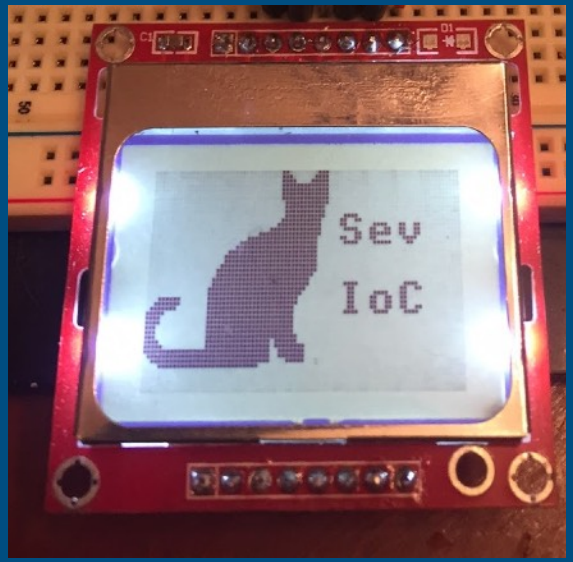

Antenna

ESP8266
(WeMos D1)

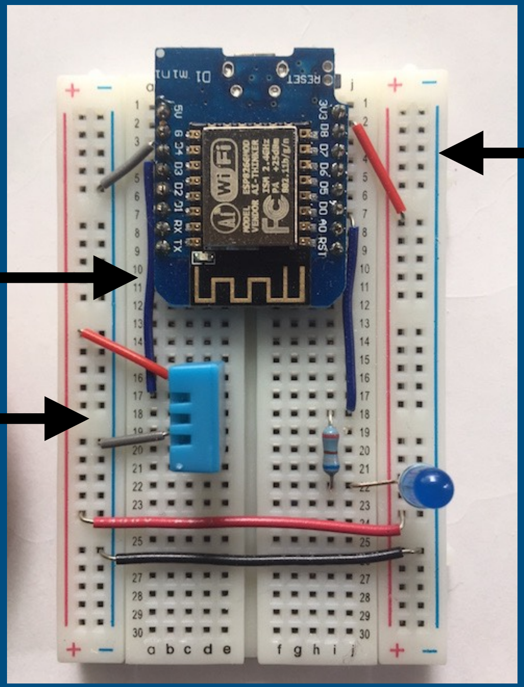

Access Point

Station

Station



# ESP 8266 Microcontroller

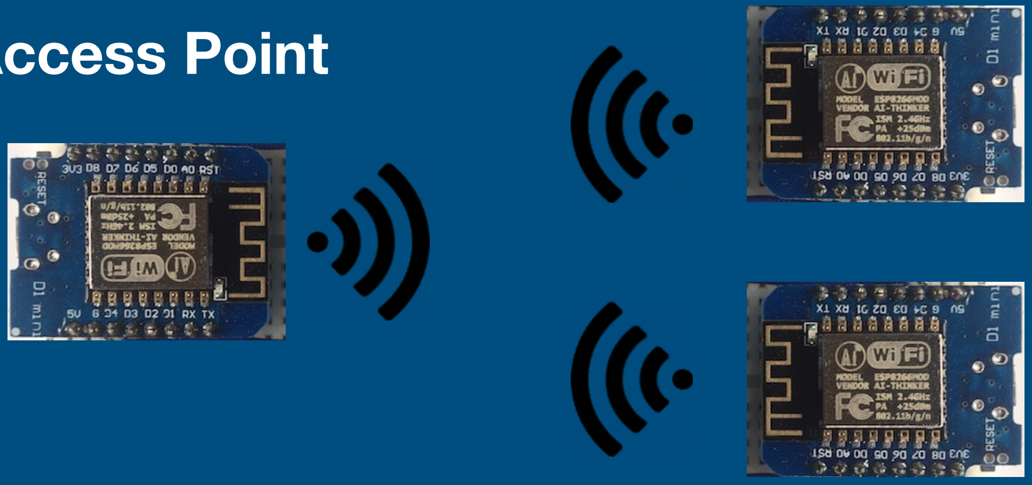


Antenna

DHT11
Digital Humidity
Temperature sensor

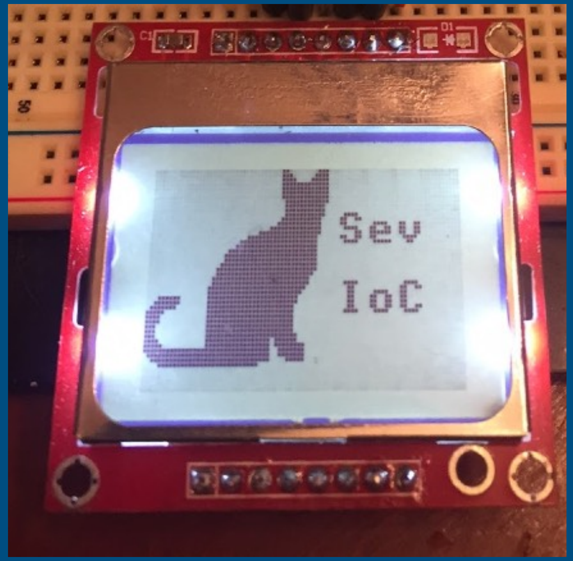
ESP8266
(WeMos D1)

Access Point

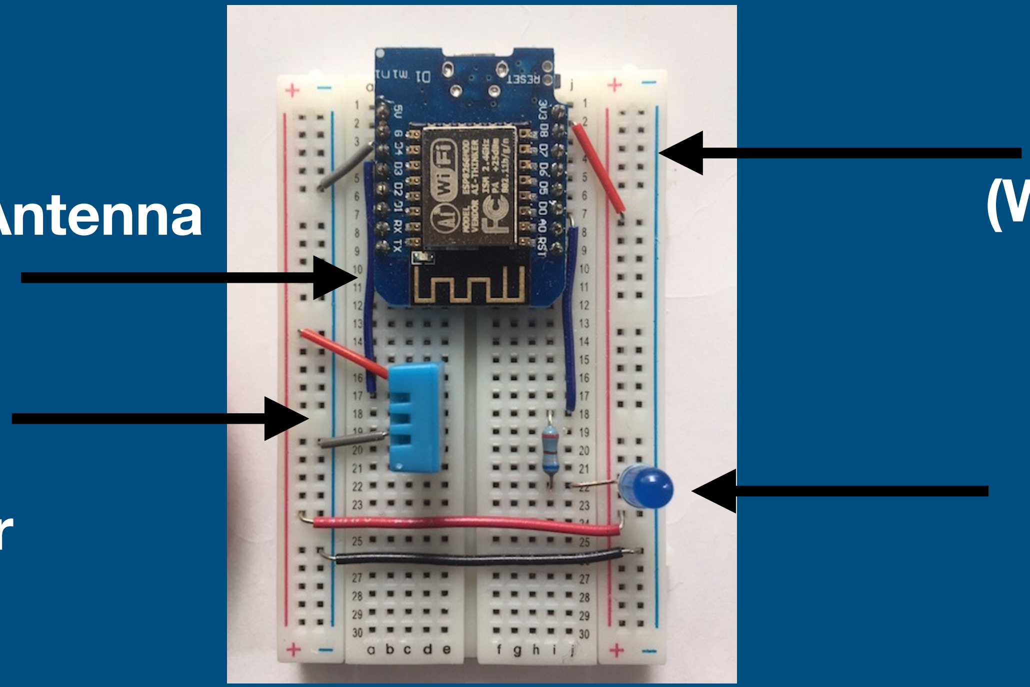


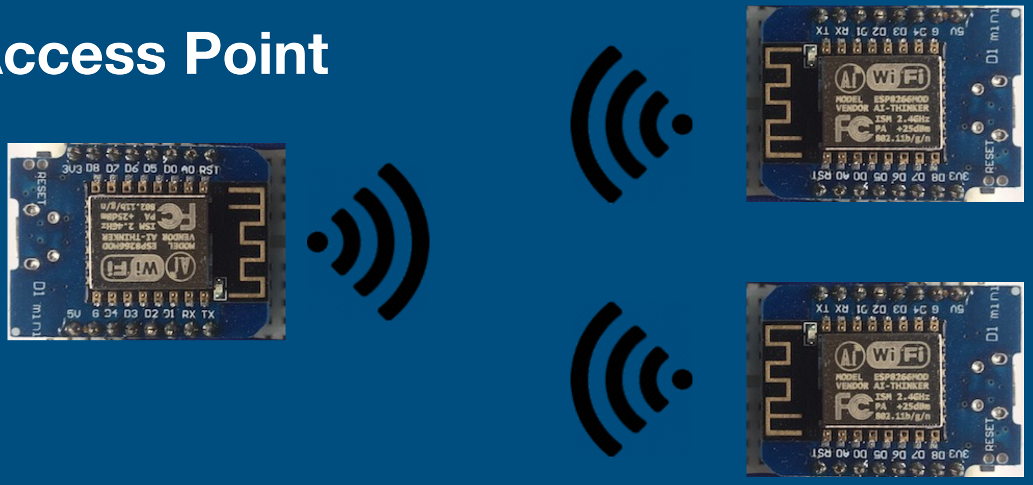
Station

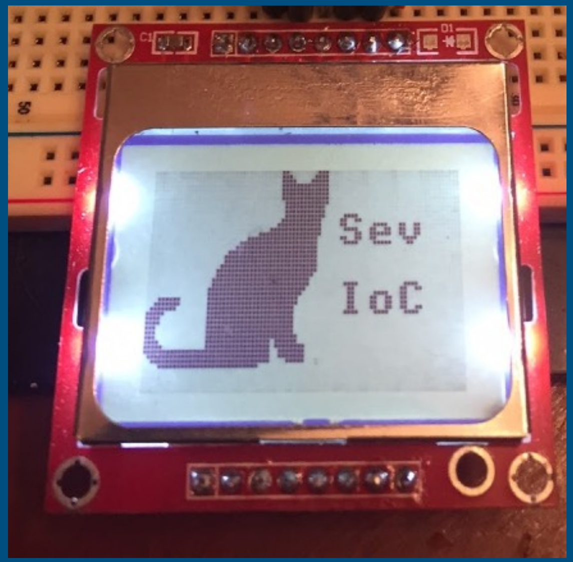
Station



# ESP 8266 Microcontroller



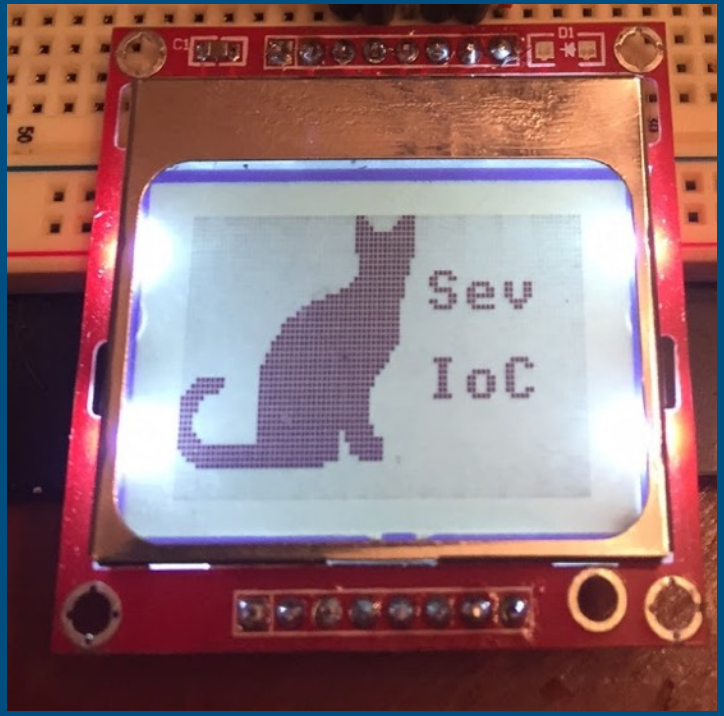




# Vineyard Sensors

- ESP 8266 Microcontroller
    - WiFi enabled
    - 3.3V
- Digital Humdity and Temperature Sensor DHT 11
- 5 USD

# Vineyard Sensors



- ESP 8266 Microcontroller
  - WiFi enabled
  - 3.3V
- Digital Humdity and Temperature Sensor DHT 11
- 5 USD