

Can you please pass the data?
IoT communication in
Micropython

Sev Leonard
PyCascades 2018
[@gizm0_0.tenforward.social](https://twitter.com/gizm0_0)

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)

2

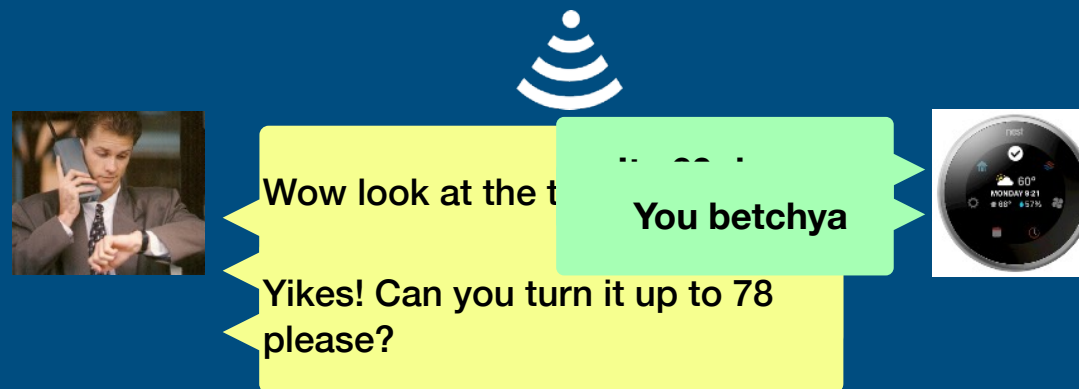
Portlandian where I live with my partner and our cat

Trans guy- not a lot of us in tech

Sadly no (or very little) snake language at my current job

Prior micropython talks include...

IoT Communication



3

The focus of this talk is how devices communicate with each other, so let's have an imaginary IoT conversation

So, the internet of things.

An internet

A thing

A humanoid with fancy internet connected devices

We're going to explore how devices like the Nest can pass messages using a very simple protocol (not email or HTTP based)



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

4

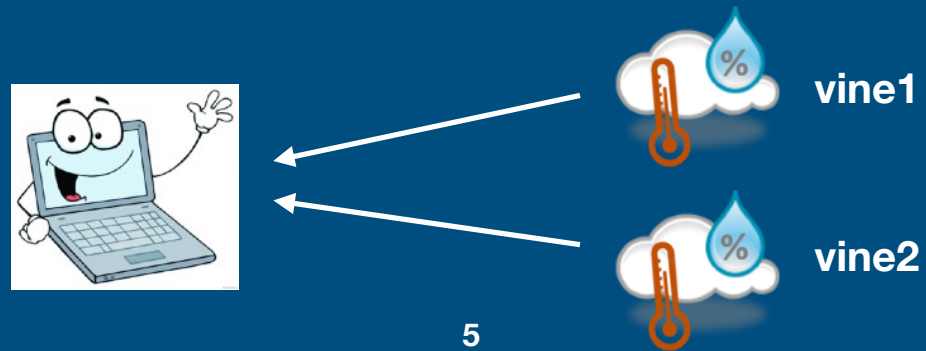
The previous example showed communication with a single device, but IoT is really about communicating with many devices. So, let's imagine we own a vineyard. Being conscientious grape farmers we want to know what the micro climates are like in different areas of the vineyard. So we deploy sensors at each vine to track this.

I'm going to be demoing some temp and humidity sensors that we can imagine are part of a vineyard IoT network

If it's too cold, we may need to turn on some heaters out in the vineyard

If it's too hot for an extended period, we may need to rethink next year's wine offerings or look into purchasing grapes

Our Internet of Things



We will use one ESP 8266 to provide the wifi for our sensors

We have 2 sensors, on vine 1 and vine2

We have the laptop that will monitor the conditions in the vineyard

And then the sensors will send the temp and humidity over the wireless to the laptop

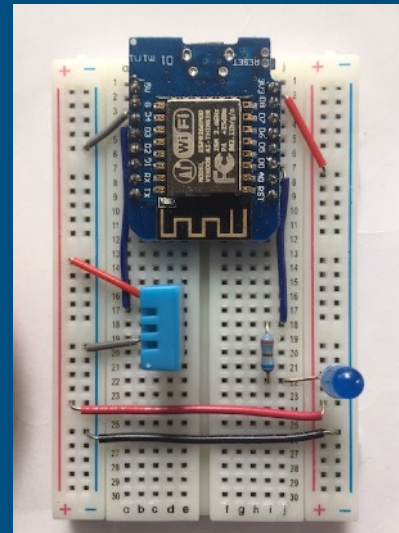
Vine Sensor



ESP8266

DHT11

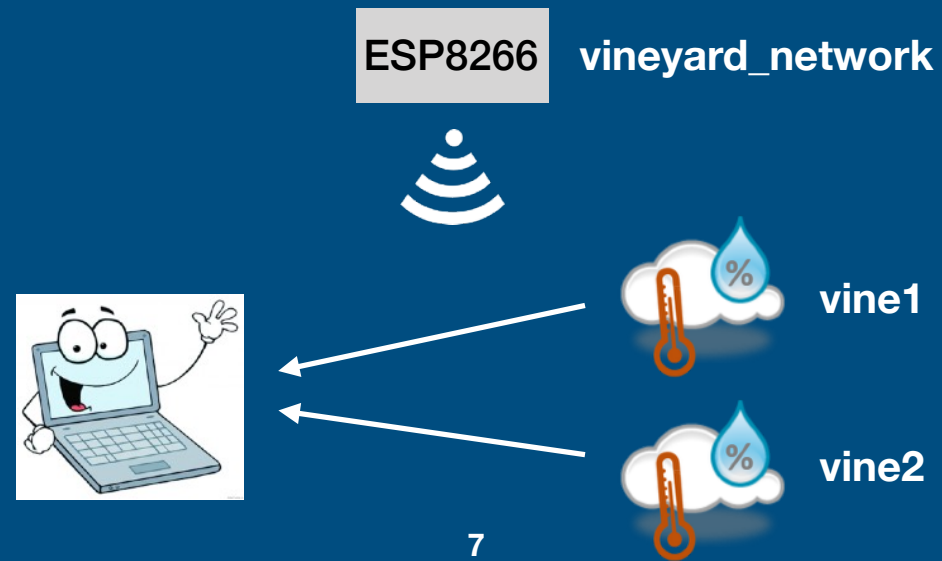
LED



6

ESP 8266 - WiFi enabled microcontroller. Can both act as an access point, broadcasting a network to connect to , and as a station - an element that connects to and communicates over a wireless network

Vineyard Network



We will use one ESP 8266 to provide the wifi for our sensors

We have 2 vine sensors - the ESP with the DHT, on vine 1 and vine2

We have the laptop that will monitor the conditions in the vineyard

And then the sensors will send the temp and humidity over the wireless to the laptop

Setup WiFi

```
1 import network
2 ap_if = network.WLAN(network.AP_IF)
3 ap_if.config(essid='vineyard_network')
4 ap_if.active(True)
```

ESP8266



8

We're going to setup the vineyard_network, so we will program and ESP to run as a wireless access point
Using the network module, we can create a wireless access point interface

Connect vine sensors

```
1 import network
2 sta_if = network.WLAN(network.STA_IF)
3 sta_if.active(True)
4 sta_if.connect('vineyard_network',password)
```



The vine sensors will connect to the vineyard_network as wireless stations, so we will use the network module to create station interfaces like this
First, active the station interface
Then, connect the station to the network

Measuring humidity

```
1 import dht
2 import time
3 my_dht = dht.DHT11(machine.Pin(2))
4
5 def measure_humidity(poll_time_s):
6     while True:
7         my_dht.measure()
8         humidity = my_dht.humidity()
9         print("humidity: ", humidity)
10        time.sleep(poll_time_s)
```

10

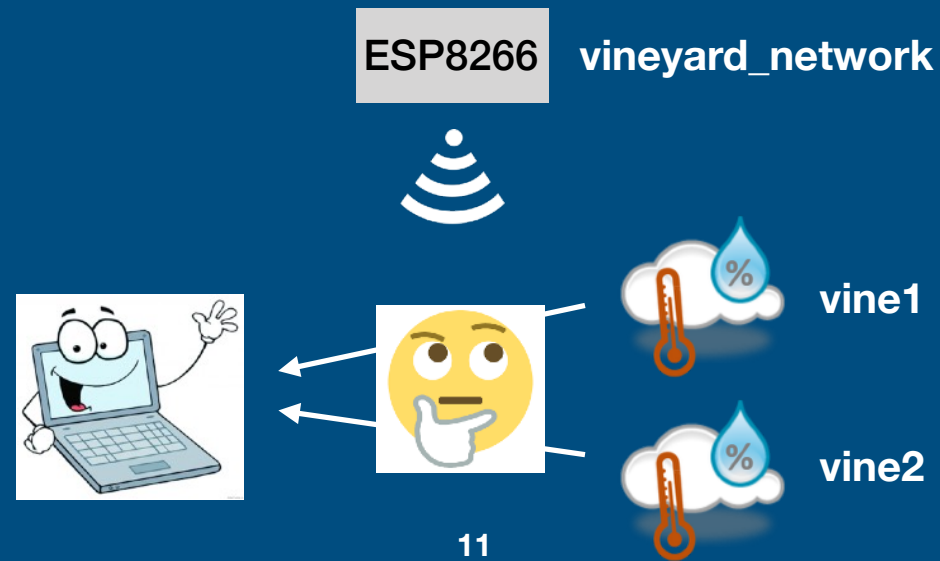
Based on Python3

Load files to be executed

Some standard python like things (time)

Some hardware things (DHT - digital humidity temp sensor) and PIN - turn the LEDs on and off

Vineyard Network



Now, we have the vineyard network access point running

We have the vine sensors connected to the network

AND we have the vine sensors recording the humidity (and temp)

Our computer is still excitedly waiting for the data from the sensors

How will we get the data there?

MQTT

Message Queuing Telemetry Transport!

- Light weight, pub/sub model w/ broker



12

MQTT - originally designed in 1999 to monitor oil pipelines, it is low power and low bandwidth

Publish / Subscribe model

Easy to scale - just set topics accordingly when new devices are added to the network

Mosquitto broker - open source MQTT broker with a nice python API

Now, our laptop can tell the broker 'hey i want to subscribe to vineyard temp and humidity'

The sensors say 'hey broker, i have some temp and humidity data to publish'

Then the broker says 'hey laptop - heres some data for you'

MQTT - Topics

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

13

Maybe you want to subscribe to all vineyard MQTT topics.

... maybe just temp

... maybe just temp for a subset of vines

the + acts like a wildcard within a topic, so vineyard/+/vine1 would subscribe to all topics related to vine1

MQTT Pub in Micropython

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

14

create an MQTT client pointed at the IP address of the broker, first param is client id

MQTT Sub in Micropython

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

15

Setup mqttt client as with pub

To subscribe, first we need a callback function to handle the message

creating an MQTT client sub using the same procedure as the preceeding slide

set the call back

subscribe to the topic (# is a wildcard to listen to any sub topics)

wait for messages

MQTT Sub in Micropython

```
1 s.set_callback(cb)
2 s.subscribe("vineyard/temp/#")
3 while 1:
4     s.wait_msg()
```

16

Setup mqttt client as with pub

To subscribe, first we need a callback function to handle the message

creating an MQTT client sub using the same procedure as the preceeding slide

set the call back

subscribe to the topic (# is a wildcard to listen to any sub topics)

wait for messages


```
1 from umqtt.simple import MQTTClient
2 import dht
3 import time
4 my_dht = dht.DHT11(machine.Pin(2))
5
6 def measure_mqtt(poll_time_s, broker_ip, client_id, topic):
7     p = MQTTClient(client_id, broker_ip)
8     p.connect()
9
10    while True:
11        my_dht.measure()
12        humidity = my_dht.humidity()
13        humid_topic = topic + '/humidity/' + client_id
14        p.publish(humid_topic, str(humidity))
15        time.sleep(poll_time_s)
```

17

Lets revisit our humidity measuring function, now with MQTT
The real code has a lot of try/except and garbage collection

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

- 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@thedata scout.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

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

25

MQTT - a protocol for IoT communication

Agricultural Technology



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

26

Tech crunch headlines from 2017

Prospera: Tel-Aviv based startup using sensors, computer vision, and AI to improve farm yields

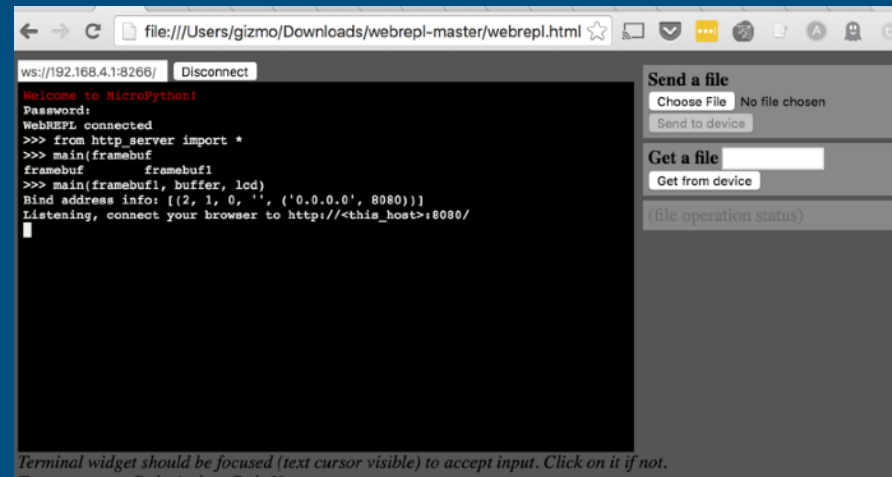
Ceres: Oakland based startup using sensors & cameras to pinpoint crop stress

Vinsight: Redwood City based startup collecting data from sensors, satellites and weather stations to correlate with crop health

Not just startups - Independence Oregon working to become an ag-tech innovation hub

- City provided gigabit broadband for every home and business
- Farm and vineyard partners to be test sites
- Focused on drones and imaging, also sensor based data

WebREPL



27

REPL - read evaluate print loop

Basically any shell you've used is a REPL

Allows hardware to be programmed over wireless, can bypass driver installation

Micropython



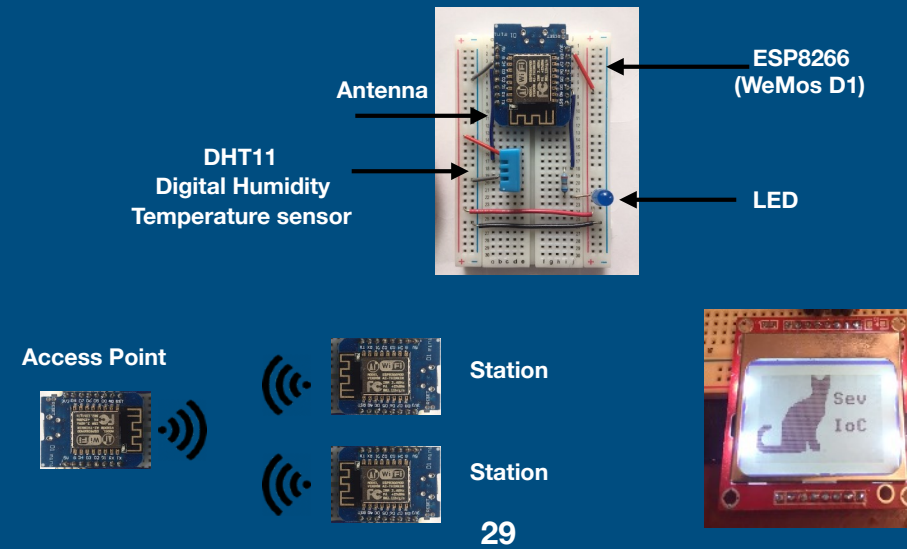
- A. Like regular Python, but small and hard to read X
- B. A version of Python optimized for use on microcontrollers
Funded via Kickstarter
- C. All of the above
- D. Some of the above
- E. Some of the above!

micropython.org

28

Micropython is a version of Python3 optimized to run on microcontrollers
Lots of great docs on micropython.org

ESP 8266 Microcontroller



Prototype board with the ESP and some additional friends.

Microcontroller - specialized system that includes a microprocessor, memory & I/O targeted to a specific application.

ESP is a microcontroller with WiFi, and is capable of both providing network access as an Access Point, and connecting to an existing network as a station **IMPORTANT FOR OUR TALK**

You can run a web server ...

This is a development board, WeMos, for the ESP8266

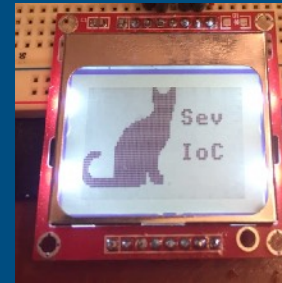
- provides nice things like Reset, onboard LED, microUSB connection

In addition to the internet, the ESP can drive displays using the GPIO pins

And if you combine the wifi functionality and the display you can create the Internet of Cats - HTTP request a cat pic and have it delivered to you at the speed of pixel rendering

Vineyard Sensors

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



30

Vineyard sensor - sense temp and humidity and relay that information

Microcontroller - specialized system that includes a microprocessor, memory & I/O targeted to a specific application.

ESP is a microcontroller with WiFi, and is capable of both providing network access as an Access Point, and connecting to an existing network as a station IMPORTANT FOR OUR TALK

You can run a web server ...

This is a development board, WeMos, for the ESP8266

- provides nice things like Reset, onboard LED, microUSB connection

In addition to the internet, the ESP can drive displays using the GPIO pins

And if you combine the wifi functionality and the display you can create the Internet of Cats - HTTP request a cat pic and have it delivered to you at the speed of pixel rendering