PHYSICAL COMPUTING WITH PYTHON

ANDY LENHART

I’m still learning!

Feel free to contribute!

An Apology in Advance…

Some of my code is still
Python 2.7

SEE MY GITHUB REPO FOR CODE

https://github.com/Andy1213/Python-and-Physical-Computing

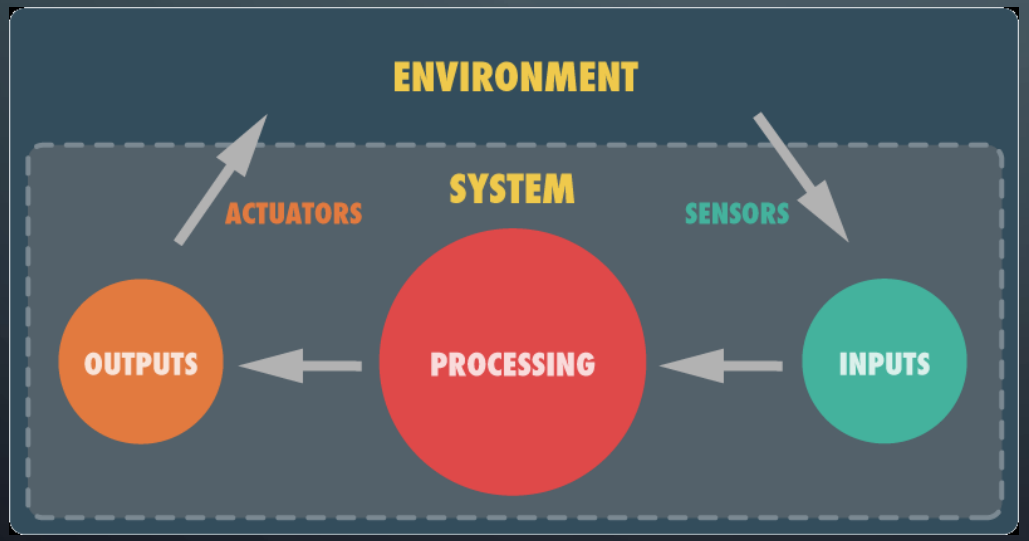
Any python file reference or URL that says

Try It!

can be run from a laptop and does not require Raspberry Pi GPIO

PHYSICAL COMPUTING

"Building interactive physical systems by the use of software and hardware that can sense and respond to the analog world"



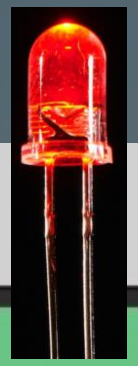
WIKIPEDIA
The Free Encyclopedia

INTERNET OF THINGS

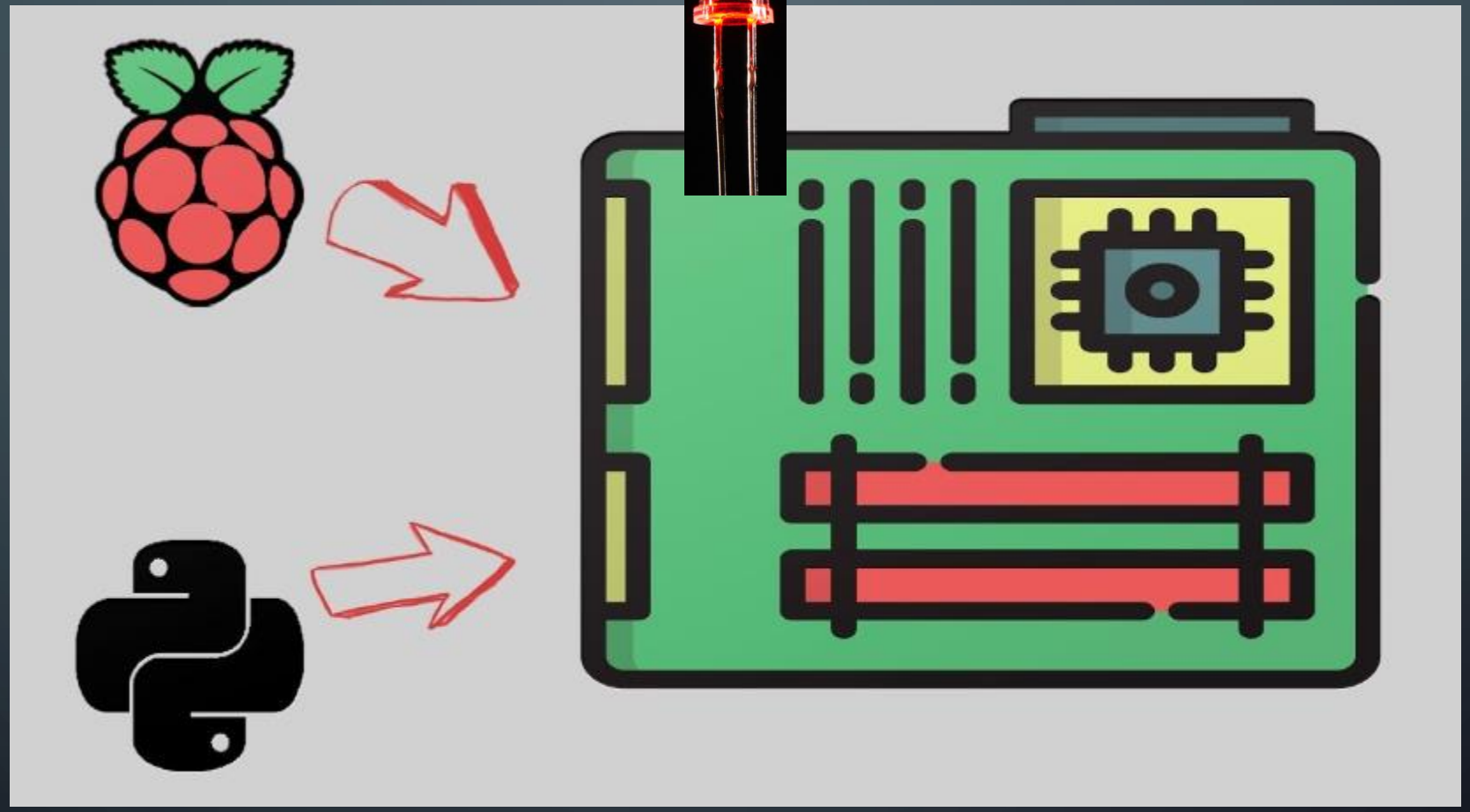
"The extension of Internet connectivity into physical devices and everyday objects"

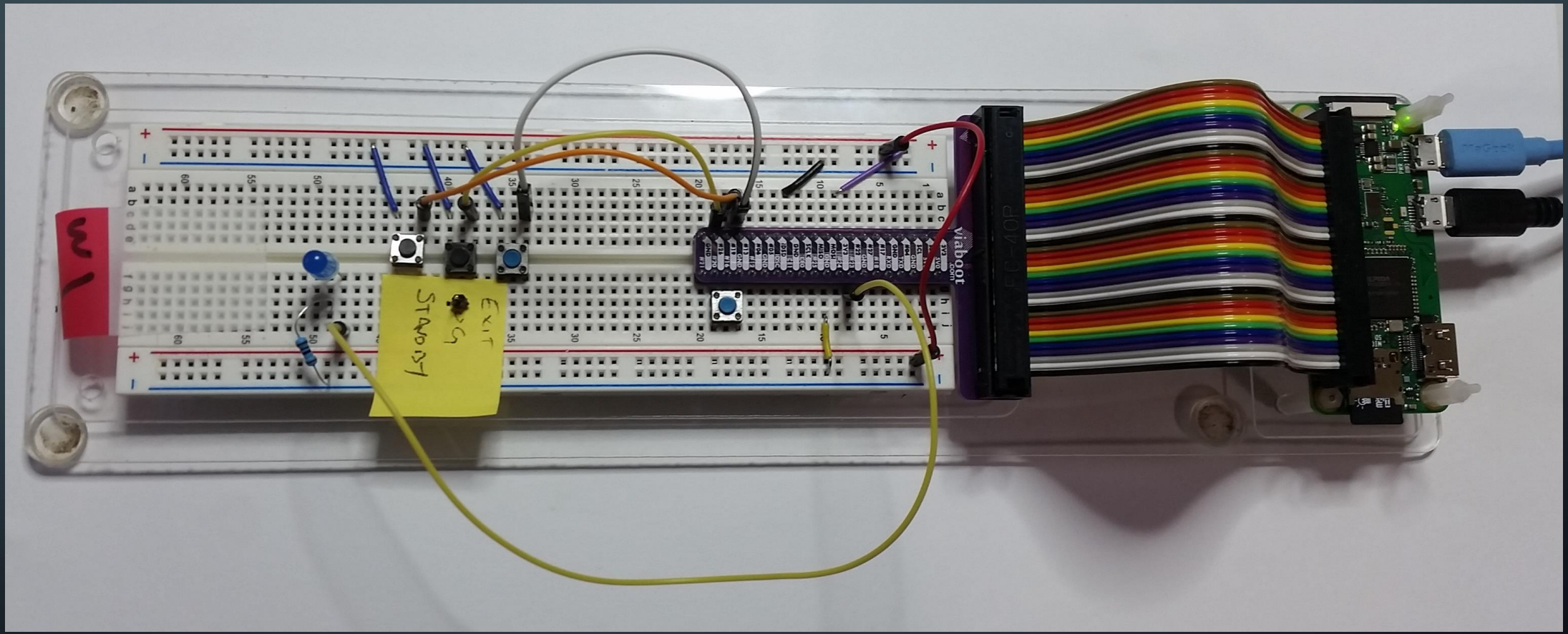
WIKIPEDIA
The Free Encyclopedia

Python + Raspberry Pi + LED



Think of the LED as a light you want to control




viaboot
EXIT
STARTY

BLINKING AN LED

- The physical computing equivalent of “Hello World”
- See Simple_Blink.py

BLINKING AN LED BASED ON AN EVENT

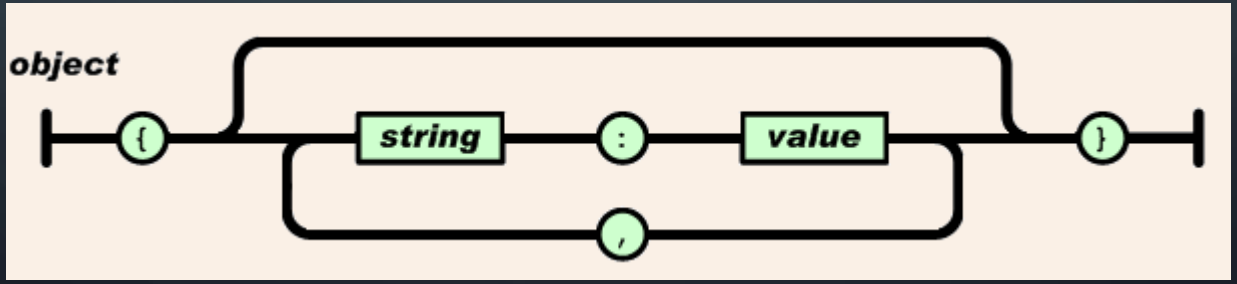
- Push a button and the LED comes on
- Release and the LED goes off
- See Button_Blink.py

BLINKING AN LED BASED ON SUNRISE AND SUNSET

- Our goal is automation and not replacing a physical switch
- Let’s turn the light on at sunset and turn it off at sunrise
- First we need to learn about JSON!

JAVASCRIPT OBJECT NOTATION - JSON

- A standard for data transmission
- Uses label/value pairs
- Common way REST API's return information
- See http://www.json.org/



JAVASCRIPT OBJECT NOTATION - EXAMPLES

{“Name”:”Andy”}

{“FirstName”:”Andy”, “LastName”:”Lenhart”}

{“Organization”:”EMPUG”,“MeetingDates”:[“Jan-01”,”May-01”,”Jul-15”]}

https://www.empug.org/data

https://forecast.weather.gov/MapClick.php?lat=42.41&lon=-83.01&FcstType=json

JAVASCRIPT OBJECT NOTATION - PARSING

- Chrome JSON Viewer extension
- http://www.jsonparseronline.com/
- *Try It!* myHomeRainForecastNOAA.py

BLINKING AN LED BASED ON SUNRISE AND SUNSET

Now that we can use JSON, back to blinking an LED…

- *Try It!* https://api.sunrise-sunset.org/json?lat=42.4606&lng=-83.1346
- *Try It!* http://www.l3nhart.io/cgi-bin/SunRiseSunSetJSONwDST.py
- See Sunset_Blink.py

BLINKING AN LED BASED ON A MESSAGE

- Ideally we would want to control many lights centrally as in home automation systems
- We can then use commands like “Turn on the kitchen lights” or “Turn on all the outside lights” or “Dim the TV room lights”
- Let’s learn about MQTT!

MESSAGE QUEUING TELEMETRY TRANSPORT MQTT

- A light weight protocol based on a publish/subscribe model
- Intended for Machine to Machine (M2M) communications
- Uses a central Broker to route messages between clients
- See http://mqtt.org/

MESSAGE QUEUING TELEMETRY TRANSPORT MQTT

- Your MQTT client must connect to a broker
- When connecting, you provide
  - The broker name or IP address and port to connect to
  - A unique client ID (username)
  - Optionally
    - Last will and testament topic and message
    - Quality of Service (QoS)
    - Keep alive interval
- Topics you want subscribe to

MESSAGE QUEUING TELEMETRY TRANSPORT

- Topics follow a hierarchy much like a directory tree

  /lights/

  /lights/outside/

  /lights/outside/front_yard/

  /lights/outside/front_yard/porch

- *Try It!* MQTT_Subscriber_i3.py (Make sure to change the client ID first!)

MESSAGE QUEUING TELEMETRY TRANSPORT

- A publish at minimum consists of a topic and a message

  Topic: “/lights/inside/kitchen”

  Message: “on”

- When publishing a message you can also specify QoS and retention
- *Try It!* MQTT_Publish.py

MESSAGE QUEUING TELEMETRY TRANSPORT

- Let's control our LED

  Topic: "/empug/lightcontrol/"

  Message: "on" or "off"

- We can also publish the current state of the light

  Topic: "/empug/lightstate/"

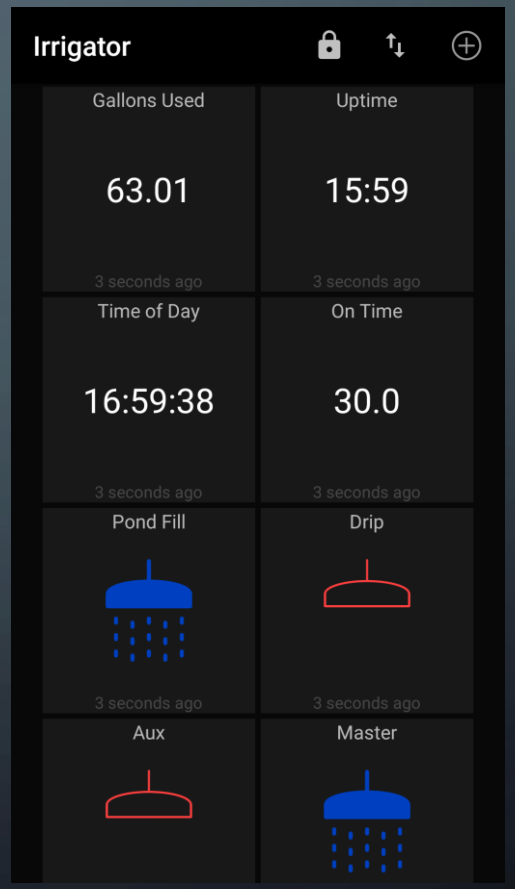
- See MQTT_Blink.py
- *Try It!* MQTT_OnOff.py [0,1] (include a command line argument for on and off)

MESSAGE QUEUING TELEMETRY TRANSPORT

- Free public brokers are available (we just used test.mosquitto.org)
- USE AT YOUR OWN RISK!
- See https://github.com/mqtt/mqtt.github.io/wiki/public_brokers

MESSAGE QUEUING TELEMETRY TRANSPORT

- There is a free android app "MQTT Dash" that you can use to subscribe and publish



MORE PYTHON AND PHYSICAL COMPUTING

- MicroPython
- CircuitPython (maintained by Adafruit)
- Python compatible microcomputers and microcontrollers
  - Raspberry Pi
  - Circuit Playground
  - BBC MicroBIT
  - PyBoard

INTERNET SERVICES

- *Machine to machine pull system:* https://dweet.io/
- *Plotting and analyzing measurements:* https://thingspeak.com/
- *Connecting services:* https://ifttt.com/discover