

Intro to Internet of Things

with ESP/Arduino



Ben Cheng

RISD ID

April 15, 2024

Outline I

1 Internet

2 Internet of Things

3 Node: ESP

4 Implementation

ESP8266

Servo

Ultrasonic Sensor

DHT22

HTTP

1 Internet

2 Internet of Things

3 Node: ESP

4 Implementation

ESP8266

Servo

Ultrasonic Sensor

DHT22

HTTP

What is the internet?

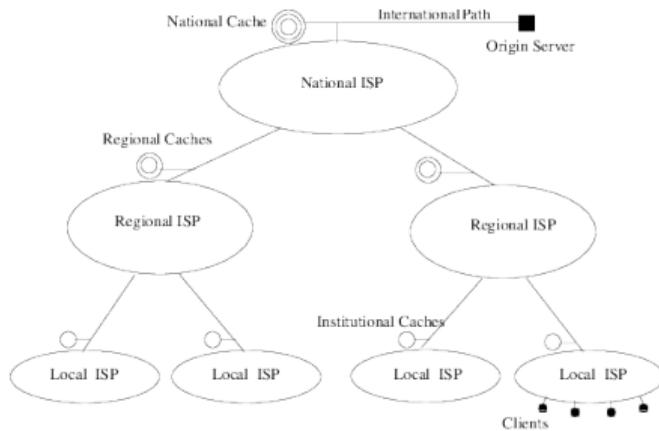
A network connecting an enormous number of computing devices.

How to operate a network like this?

- Wire them all together?
 - Who connects to whom?
 - How many steps to send a message?

Hierarchy+Protocol

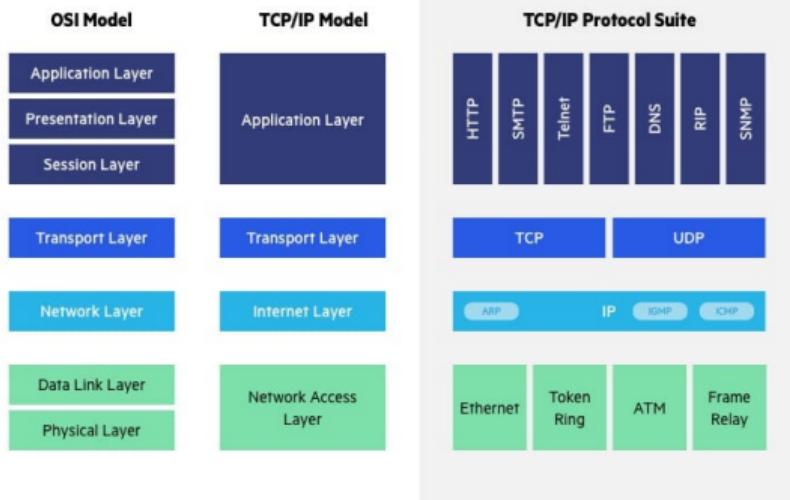
- Devices are connected by hierarchy.
 - Different devices are connected via different protocols.
 - Data is coded according to the layers of the internet model.



(Hu, Rodriguez, and Biersack 2000)

Layers of the internet

- 1 HTTP request
 - 2 TCP port
 - 3 IP address
 - 4 MAC address
 - 5 Wireless LAN

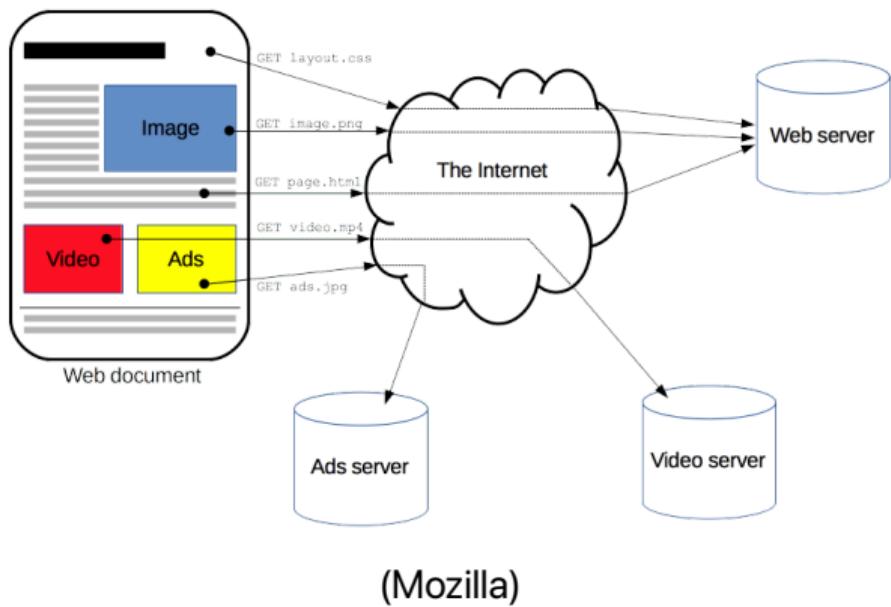


(Imperva)

(Almost) Everything is a HTTP request

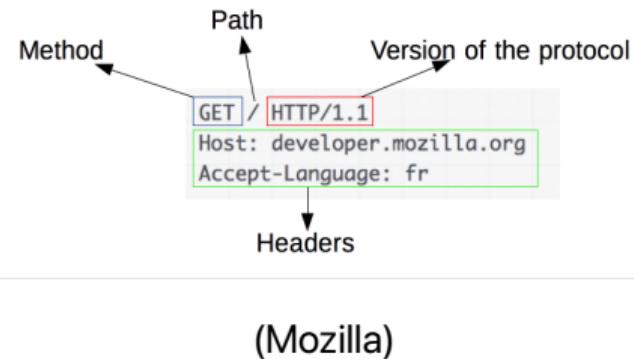
HTTP follows a client-server model.

- Client request
 - Serve respond



HTTP request

- Method: GET, POST
 - Path
 - Header
 - Body



1 Internet

2 Internet of Things

3 Node: ESP

4 Implementation

ESP8266

Servo

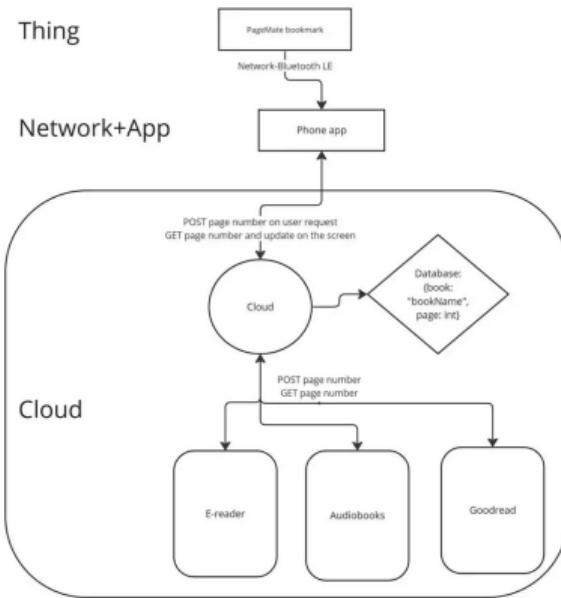
Ultrasonic Sensor

DHT22

HTTP

Components

- 1 Node
 - 2 Gateway
 - 3 Cloud
 - 4 Database



Example: Power Cable Monitoring

- Long distance between towers
 - Connection hard/dangerous to install



What's covered today?

- Node
 - Gateway: your phone
 - Cloud: a basic Express app
 - Database: MongoDB, mySQL

1 Internet

2 Internet of Things

3 Node: ESP

4 Implementation

ESP8266

Servo

Ultrasonic Sensor

DHT22

HTTP

What is a computer?

1 Processing Unit: ALU

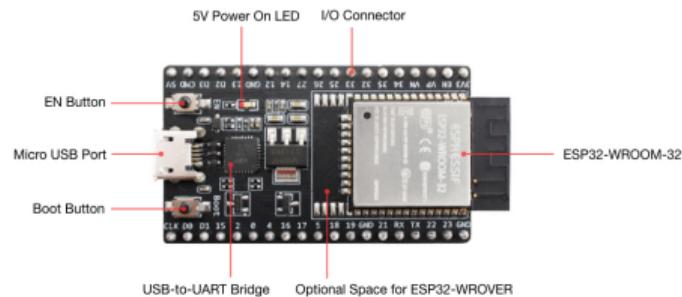
- Arithmetic operation
 - Signal processing
 - Conditional decision

2 Memory: hierarchy

3 | /O

- ADC
 - USB
 - Wireless

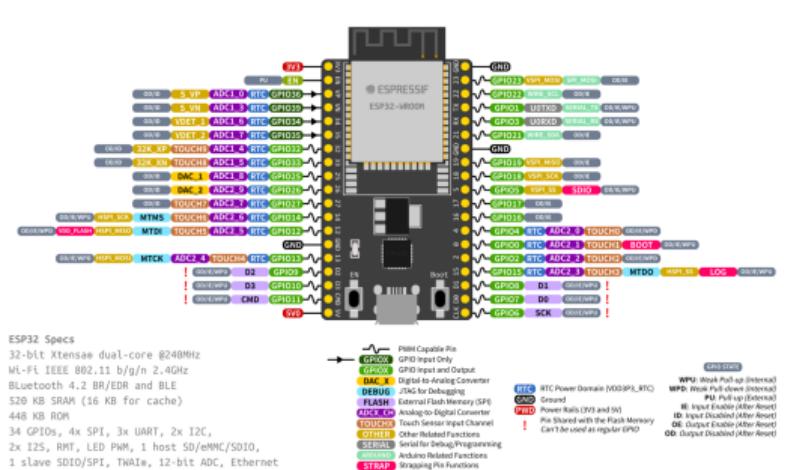
ESP32 is a system on chip (SoC).



(Espressif)

Embedded Computer

- Does not have operating system.
 - Application is embedded into the firmware.



(Espressif)

1 Internet

2 Internet of Things

3 Node: ESP

4 Implementation

ESP8266

Servo

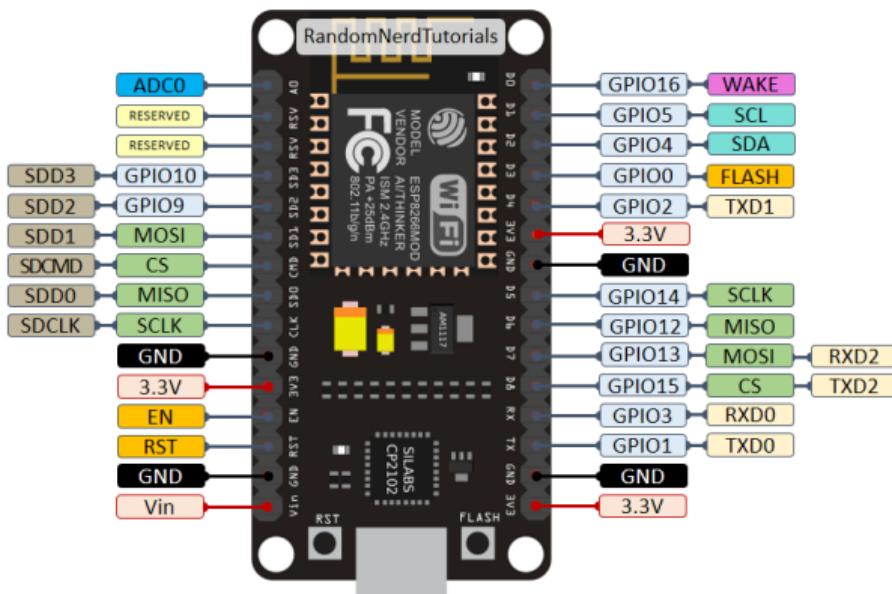
Ultrasonic Sensor

DHT22

HTTP

ESP8266

ESP8266



(Random Nerd Tutorials)

Checking and Basic Code Structure

- 1 Connect ESP8266 to your computer and open Arduino IDE
 - 2 Select board NodeMCU 1.0 (ESP12E) and choose the right port in Tools
 - 3 Open File>Example>ESP8266>Blink and upload

```
1 void setup() {
2     pinMode(LED_BUILTIN, OUTPUT);
3 }
4
5 // the loop function runs over and over again forever
6 void loop() {
7     digitalWrite(LED_BUILTIN, LOW);
8     delay(1000);
9     digitalWrite(LED_BUILTIN, HIGH);
10    delay(2000);
11 }
```

Servo

Servo Motor

- Small
 - Powered by internal 3.3V
 - Easy to control
 - Only rotate 180 degree



(ElectronicWings)

Sweep

- Orange to GPIO14=D4, red to 3V3, brown to GND

```
1 #include <Servo.h>
2 Servo myservo;
3 void setup() {
4     myservo.attach(14);
5 }
6 void loop() {
7     int pos = 0;
8     for (pos = 0; pos <= 180; pos += 1) {
9         myservo.write(pos);
10        delay(30);
11    }
12    for (pos = 180; pos >= 0; pos -= 1) {
13        myservo.write(pos);
14        delay(30);
15    }
16 }
```

Ultrasonic Sensor

Ultrasonic Sensor

- Includes an emitter and a receiver.
 - Calculate distance by time taken to travel from emitter to receiver.
 - Ultrasonic is a mechanical wave

```
1 #include <Servo.h>
2 Servo myservo;
3 const int trigPin = 0; //connect trig pin to GPIO0=D3
4 const int echoPin = 4; //connect echo pin to GPIO4=D2
5
6 float duration, distance;
```

Ultrasonic Sensor

Setup and Serial Communication

- Serial sends text to your computer
 - Tools>Serial Monitor>Set correct baud rate

```
7 void setup() {  
8     myservo.attach(14);  
9     pinMode(trigPin, OUTPUT);  
10    pinMode(echoPin, INPUT);  
11  
12    Serial.begin(115200);  
13    Serial.println("Hello World!");  
14 }
```

Ultrasonic Sensor

Calculate distance and send via serial

```
15 void loop() {  
16     digitalWrite(trigPin, LOW);  
17     delayMicroseconds(2);  
18     digitalWrite(trigPin, HIGH);  
19     delayMicroseconds(10);  
20     digitalWrite(trigPin, LOW);  
21  
22     duration = pulseIn(echoPin, HIGH);  
23     distance = (duration*.0343)/2;  
24     Serial.print("Distance: ");  
25     Serial.println(distance);  
26 }
```

DHT22

DHT22

- An integrated temperature and humidity sensor
- Easy to use with complete library

```
1 #include <Servo.h>
2 #include "DHT.h"
3
4 #define DHTPIN 4 //GPIO4=D2
5 #define DHTTYPE DHT22
6
7 Servo myservo;
8 DHT dht(DHTPIN, DHTTYPE);
```

Setup

```
9 void setup() {  
10     myservo.attach(14);  
11     dht.begin();  
12  
13     Serial.begin(115200);  
14     Serial.println("Hello World!");  
15 }
```

Loop

```
16 void loop() {
17     float t = dht.readTemperature(true);
18     if (isnan(t)) {
19         Serial.println("DHT failed");
20     }
21     else {
22         Serial.println("Temp: " + String(t));
23     }
24 }
```

Send HTTP request

Files>Examples>ESP8266HTTPClient>BasicHttpsClient

Modify the following lines:

```
37 WiFiMulti.addAP("RISD-MiscDevices", "T3chn0l0gy");  
  
46 // client->setFingerprint(fingerprint);  
47 // Or, if you happy to ignore the SSL certificate, then use the  
// following line instead:  
48 client->setInsecure();  
  
53 if (https.begin(*client, "https://idsb-iot.onrender.com")) { //  
// HTTPS
```

You should get a response of Hello World!.

GET request

```
55     Serial.print("[HTTPS] GET...\n");
56     // start connection and send HTTP header
57     int httpCode = https.GET();
58
59     // httpCode will be negative on error
60     if (httpCode > 0) {
61         // HTTP header has been send and Server response header has been handled
62         Serial.printf("[HTTPS] GET... code: %d\n", httpCode);
63
64         // file found at server
65         if (httpCode == HTTP_CODE_OK || httpCode == HTTP_CODE_MOVED_PERMANENTLY) {
66             String payload = https.getString();
67             Serial.println(payload);
68         }
69     } else {
70         Serial.printf("[HTTPS] GET... failed, error: %s\n", https.errorToString(httpCode
71                         ).c_str());
72     }
73
74     https.end();
```

GET vs POST request

Definition (GET)

The GET method requests a representation of the specified resource. Requests using GET should only retrieve data. (Mozilla)

GET /index.html

Definition (POST)

The POST method submits an entity to the specified resource, often causing a change in state or side effects on the server. (Mozilla)

```
POST /test HTTP/1.1
Host: foo.example
Content-Type: application/x-www-
    form-urlencoded
Content-Length: 27
```

field1=value1&field2=value2

POST a JSON object

```
{  
    "name": "ben",  
    "num": 40  
}
```

```
https.addHeader("Content-Type", "application/json");  
Serial.print("[HTTPS] POST...\n");  
// start connection and send HTTP header  
int httpCode = https.POST("{\"name\":\"ben\", \"num\":\"" + String(t)  
+ "\"});
```

What is `String(t)`?

Try to figure it out!

Listing: Pseudo code

```
void loop() {  
    if wifi connected  
        read sensor (DHT or ultrasonic)  
        float t = value read from sensor  
        post a json object to the server //this is done  
        retrieve response from server //also done  
        if response = true  
            sweep motor once  
}
```

What does this device do?

- Understand environment (sensor)
- Take actions to change environment (actuator)
- Make decision (MCU)
- Two-way communication with the server
 - Data collection
 - Remote control

Thank you!

A recap

- HTTP request
- 4 components of IoT
- `setup(){} , loop(){}`
- Serial communication
- JSON