

Bring JavaScript to the Internet of Things - From Embedded Device to Smart Gateway

Ziran Sun (ziran.sun@samsung.com)
Philippe Coval (p.coval@samsung.com)
Samsung Research, UK
Feb, 2019
FOSDEM, Brussels

- 1. Why JavaScript in IoT?**
- 2. JavaScript on Embedded Device**
 - a. Ultra-Light JS Engine - JerryScript
 - b. IoT Platforms on JerryScript
- 3. JavaScript on Smart Gateway**
- 4. Put It Together - A Simple Demo**
- 5. WebThings API**
- 6. There Are MORE You Can Do...**

1. Why JavaScript in IoT?

JavaScript is a good fit for IoT:

- Popularity
- Maturity
- Productivity

"Any application that can be written in JavaScript will eventually be written in JavaScript." - Jeff Atwood

2. JavaScript on Embedded Device

IoT HardWare [1]

IoT Hardware	Memory spec	IoT Hardware	Memory spec
 PanStamps	-RAM : 2KB -ROM : 33KB (Flash:32KB, EEPROM:1KB)	 Nanode	-RAM : 32KB -ROM
 TinyDuino	-RAM : 2KB -ROM : 33KB (Flash:32KB, EEPROM:1KB)	 Arduino Yun	-RAM : 2.5KB -ROM : 33KB (Flash:32KB, EEPROM:1KB)
 Arduino Uno	-RAM : 2KB -ROM : 33KB (Flash:32KB, EEPROM:1KB)	 mbed – LPC1768	-RAM : 32KB -ROM : 512KB
 RFduino	-RAM -ROM	 Wi-Go Module	-RAM -ROM : 2MB
 XinoRF	-RAM : 2KB -ROM : 33KB (Flash:32KB, EEPROM:1KB)	 pcDuino	-RAM : 1GB -ROM : 2GB
 OpenKontrol Gateway	-RAM : 32KB -ROM	 OpenPicus Flyport WiFi	-RAM -ROM : 16MB
 Pinoccio	-RAM : 32KB -ROM :	 Hackberry	-RAM :512MB/1GB -ROM : 4GB
 Raspberry Pi	-RAM : 512MB -ROM	 UDOO	-RAM : 1GB -ROM
 BeagleBone Black	-RAM : 512MB -ROM	 Libelium Wasmote	-RAM -ROM
 CubieBoard	-RAM: 512MB/1GB -ROM	 The Rascal	-RAM :64MB -ROM

(Table credit to Tilmann Scheller)

2a. Ultra-light JS Engine -JerryScript by Samsung



JerryScript

- An ultra lightweight JavaScript engine (<64KB RAM, <200 KB ROM, 160KB binary size compiled for ARM Thumb-2)
- Originally developed from scratch by Samsung
- Transferred to JS Foundation in 2016 (<https://js.foundation/>)
- Self-contained and extremely portable
- Mature C API, easy to embed in applications

JerryScript

- Optimization and performance have been top priorities
- Adding new features:
 - Jerry-debugger
 - ES6 related features
 - More hardware and OS supports
- Fast growing community

Hardware and OS Supports (JerryScript Github)



ESP8266



ARM MBED



STM32F4 discovery
Board with Nuttx, RIOT



TizenRT on
ARTIK O5X



OpenWrt



Particle



Particle firmware on
Photon board



Zephyr™



Zephyr OS on
Arduino 101



Raspberry Pi 2

JerryScript in Smartwatch [2]

SAMSUNG
Open Source Group



 fitbit

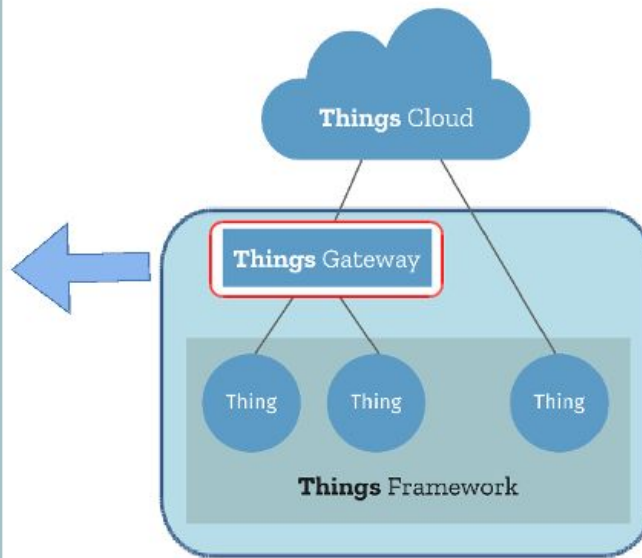
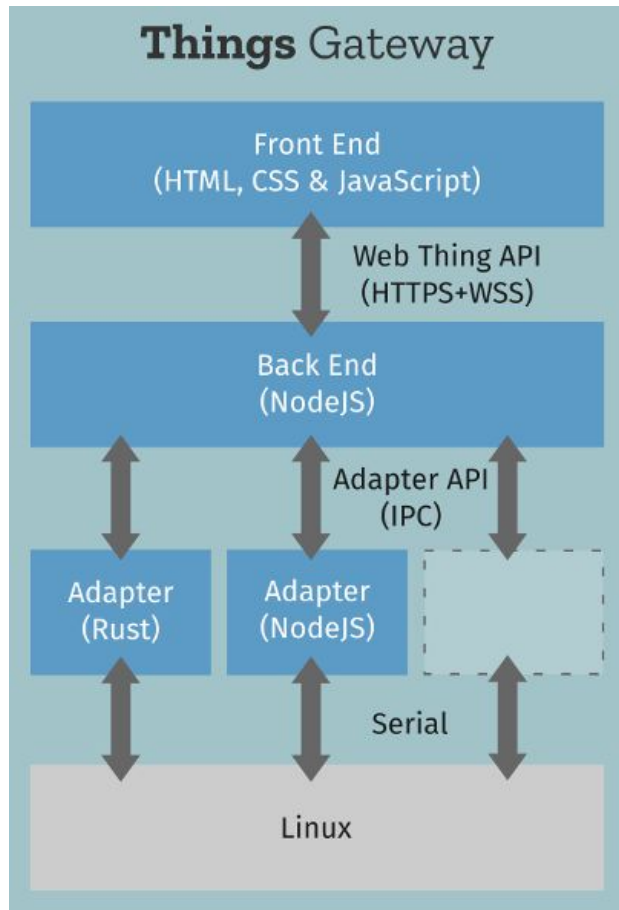
2b. IoT platforms based on JerryScript

IoT JavaScript Platforms based on JerryScript

- **IoT.js by Samsung**
 - a. Lightweight version of node.js for constrained devices
 - b. Hardware and OS supports are well inline with JerryScript.
- **Zephyr.js by Intel**
 - a. JavaScript* Runtime for Zephyr OS
 - b. Node.js* like APIs

3. JavaScript on Smart Gateway

Things Gateway By Mozilla [4]



Project Things

(Diagram credit to Ben Francis)

- **HTTPS via mozilla-iot.org tunnelling service**
 - a. Allow setting up a secure subdomains with LetsEncrypt TLS certificates
 - b. TCP tunnel uses PageKite from Mozilla cloud server to Gateway
- **OAuth to authorise third party apps & services**
- **JSON Web Tokens (JWT) used for authentication**

4. Put it Together - A Simple Demo

Turn on your light by clapping hands



Blink! light



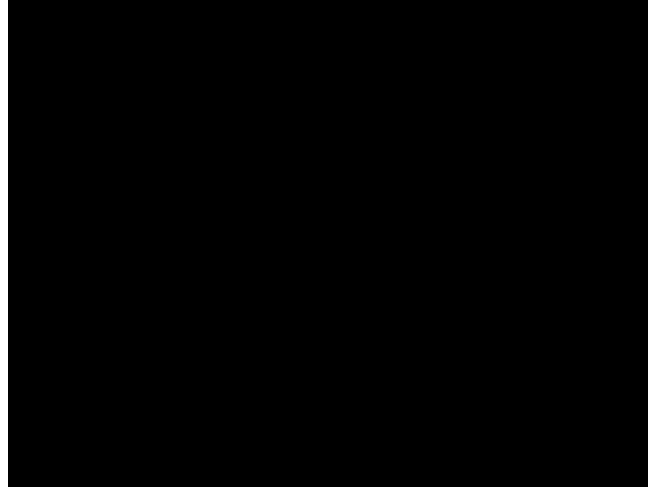
Raspberry Pi Zero



Things Gateway



Demo video



<https://vimeo.com/user90242752/review/293325534/5d519c4bc7>

Software on Device

Blinkt! Light WebThing

WebThings-IoT.js

IoT.js (with Built-in GPIO module)

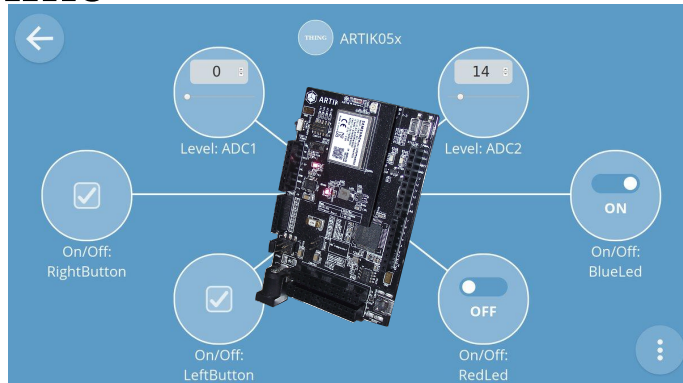
JerryScript

Raspbian

5. WebThings API

- Thing: standalone HTTP server (+Websockets)
 - REST Endpoints (for gateway's interaction):
 - Things / Properties / Values
 - Plus Actions & Events
 - Models are described using schemas (JSON)
- Several Technologies:
 - JavaScript: using Node.js or IoT.js
 - Native (C++), Python (& uPy), RUST...

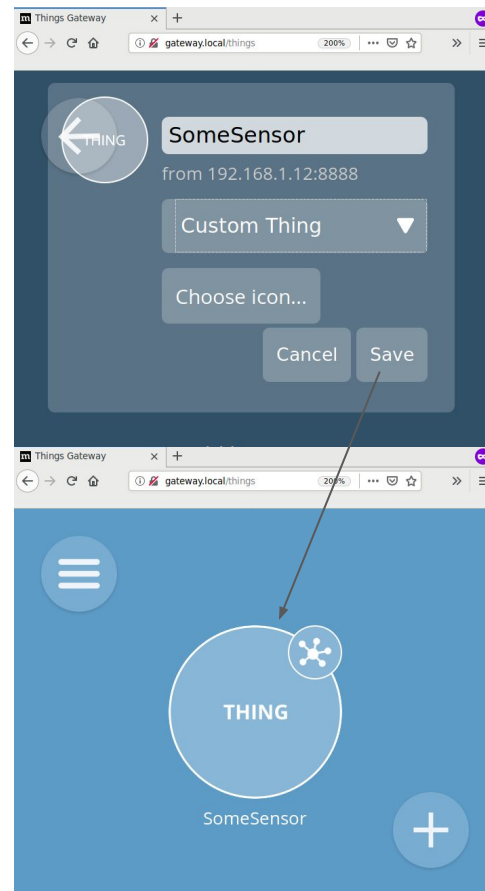
- Fork of webthing-node for IoT.js runtime
 - Port to older ECMAScript 5.1
 - Backward compatible for Node.JS
- Support: ADC, GPIO (+ some NPMs)
- Some features not yet implemented:
 - Websockets for real time (events & actions)
 - mDNS for discovery
- <https://www.npmjs.com/package/webthing-iotjs>



Implementing a Webthing server:

```
var webthing = require('webthing-iotjs');  
var thing = new webthing.Thing('SomeSensor');  
var server = new webthing.WebThingServer  
    (new webthing.SingleThing(thing), 8888);  
server.start();
```

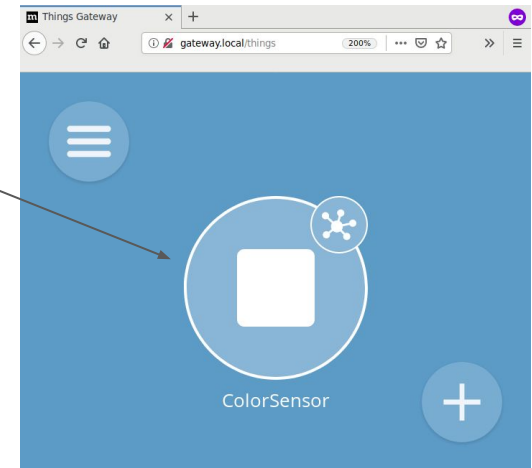
```
$ curl http://localhost:8888  
{ "name": "SomeSensor", "href": "/",  
  "@context": "https://iot.mozilla.org/schemas",  
  "@type": [null],  
  "properties": {},  
  "links": [{"rel": "properties", "href": "/properties"}]}
```



Color as webthing property:

```
function ColorProperty(thing) {  
  webthing.Property.call(this, thing, 'Color',  
    new webthing.Value('#000000'),  
    { '@type': 'ColorProperty', type: 'string',  
      readOnly: true });  
}  
//...  
var thing = new webthing.Thing('ColorSensor',  
  ['ColorControl']);  
thing.addProperty(new ColorProperty(thing));  
//...  
server.start();
```

```
$ curl http://localhost:8888  
{ ...  
  "properties":{"Color": ...  
    "links": [  
      {"rel": "property",  
        "href": "/properties/Color"  
    }  
  ]  
}
```



Sensor updating webthing

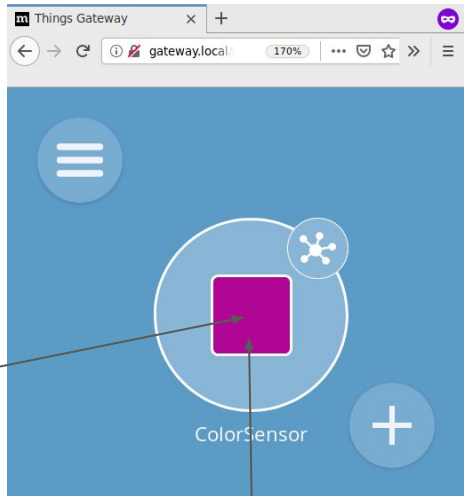
```
var ColorSensor = require('color-sensor-js/example'); // H.A.L.
```

```
function ColorProperty(thing) {  
  var self = this;  
  Webthing.Property.call(..., 'Color', new webthing.Value...  
  this.sensor = new ColorSensor();  
  this.sensor.onreading = function() {  
    self.value.notifyOfExternalUpdate(self.sensor.color);  
  };  
  this.sensor.start();  
}
```

```
$ curl http://localhost:8888/properties/Color
```

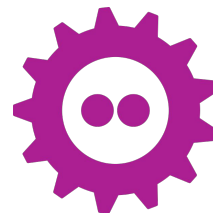
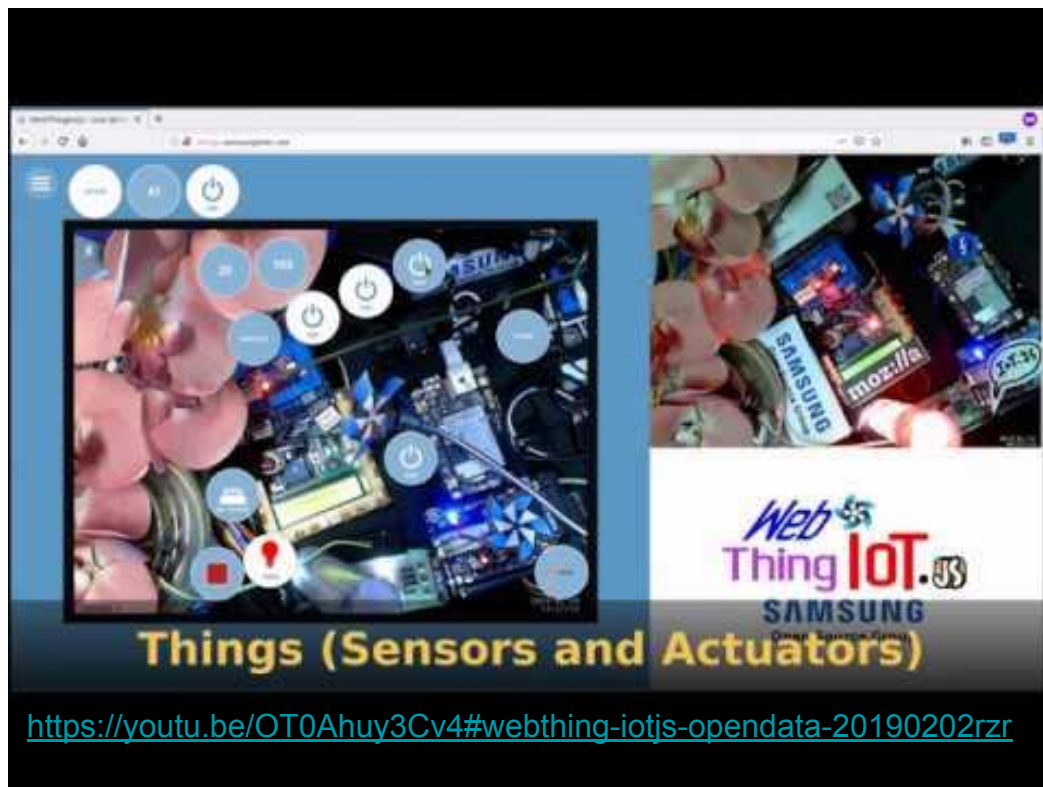
```
{"Color": "#af0695"}
```

```
// More hints: https://github.com/rzr/webthing-iotjs/wiki/Sensor
```



6. There are more you can do...

Live demo !



moz://a



Extra demos

- Tutorial & Demo recipes:
 - <https://github.com/rzr/webthing-iotjs/wiki>
- Alternative browser using TAU framework
 - For Tizen or as Progressive Web App:
 - supported by SamsungInternet (Android)
- From IoT to Social Web:
 - using mastodon-lite, IoT.js/Node.js module
- MCU: TizenRT and hints (STM, ESP, Arduino)



WebThing-IoT.js on TizenRT

SAMSUNG
Open Source Group



TIZEN™ 

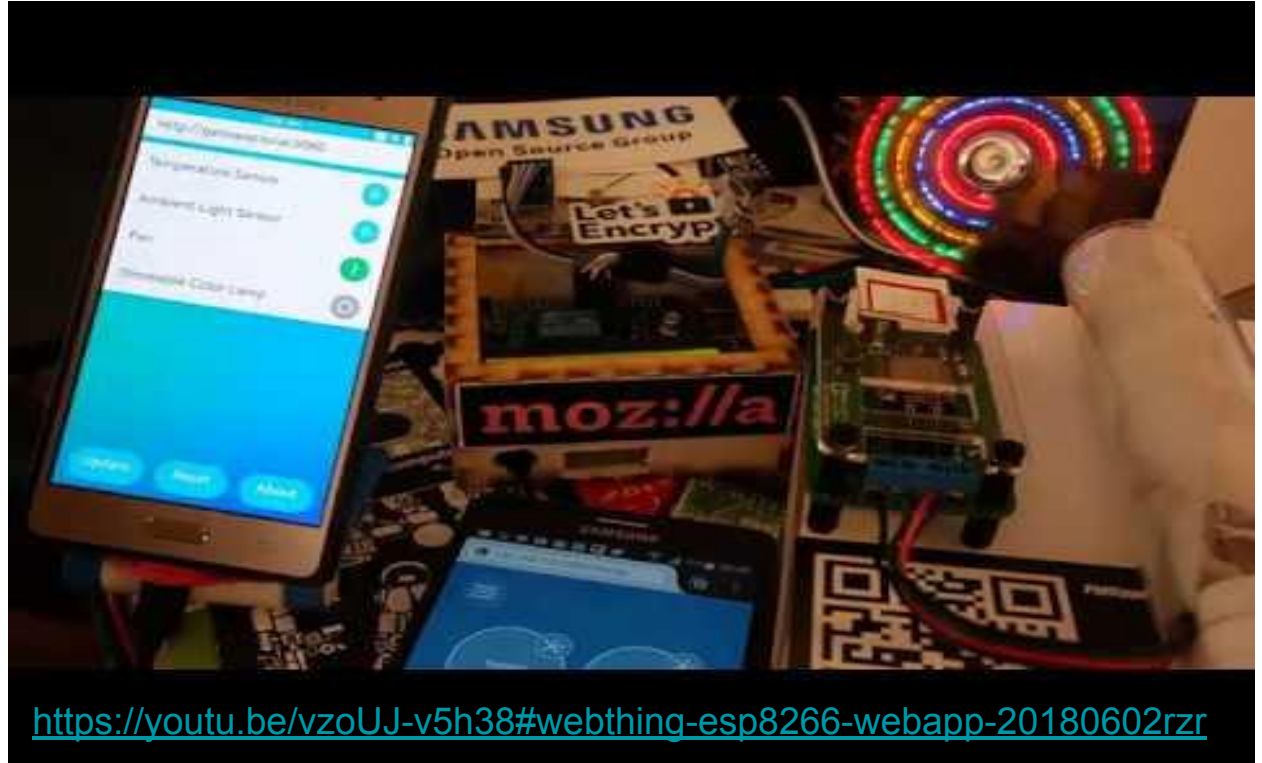


WebThing-IoT.js on TizenRT

SAMSUNG
Open Source Group



TIZEN™ 



<https://youtu.be/vzoUJ-v5h38#webthing-esp8266-webapp-20180602rzt>

Bridge Web to any framework ?

SAMSUNG
Open Source Group

moz://a

IoTivity

TIZEN™



Websockets for real time feedback to WebApp



Web IoT speedometer demo
Using W3C Javascript Websockets
on OSVehicle Tabby Evo
<https://fr.flossmanuals.net/camposv/>
Hackathon Rennes France 2018
<https://blogs.s-osg.org/author/pcoval/>
CC BY SA 4.0

<https://youtu.be/xl6eweCZLe8/#web-iot-automotive-20180315zr>

Thank You!

References:

1. *"Introduction to IoT.js", Tillman Scheller, OpenIoT & ELC 2016, San Diego, CA, Apr 2016..*
2. *"Fitbit loves JerryScript: JavaScript Breaks into the Internet of Things", Michelle Gienow,*
<https://thenewstack.io/fitbit-%E2%9D%A4%EF%B8%8Fs-jerryscript-javascript-breaks-internet-things/>
3. *"IoT.js and JerryScript: a JS platform for the embedded world", Zoltan Herczeg, Hosung Ki, Node Summit 2017*
4. *"Building the Web of Things", Ben Francis,*
<https://hacks.mozilla.org/2017/06/building-the-web-of-things/>