Agenda

- Problem Statement
- Solution
- Demo
- How does it Work?
- JavaScript*
- Web USB
- Web Application

*Other names and brands may be claimed as the property of others.
Problem Statement
IoT is Hard!
Starting New IoT Project is Hard

- Development Environment Setup
- Cables
- Sensors
- Documentation
- Sample code
- BIOS, firmware, OS updates
Solution
What if...

...this is all you need?
Demo
How Does It Work?
How Does It Work?

1. Supported URLs
2. Tell URLs
3. Show USB Dialog
4. Get URL
5. Load Application
6. Connect

JS Runtime
JerryScript
Zephyr
MCU
ashell
WebUSB
Web Site
USB Cable
Development Flow

Native
- Edit
- Compile
- Reboot
- Flash
- Run

JavaScript*
- Edit
- JS Runtime for Zephyr
- Run
- Copy
- WebIDE
- WebUSB

*Other names and brands may be claimed as the property of others.
JavaScript

*Other names and brands may be claimed as the property of others.
Enable **JavaScript** application development on Zephyr OS

Address **large** JavaScript developer **community**

**Fast** development cycle - No flashing, just copy .js files

Based on open source JerryScript JS engine and **API layer**

**Well known** JavaScript **APIs** (Node.js* like)

Application **portability** between MCU and MPU platforms

Support now for **Arduino101** board and **FRDM-K64F**, all Zephyr OS supported boards in the future

*Other names and brands may be claimed as the property of others.
Architecture

- **JavaScript* App**
  - Business logic by the app developer

- **JS Runtime for Zephyr**
  - API bindings
  - Build tools
  - Sample and demo apps
  - API docs
  - Open source (Apache 2.0)

- **JS Engine**
  - Micro JS engine - JerryScript
  - Open source (Apache 2.0)

*Other names and brands may be claimed as the property of others.
Two Modes

- **Runtime Mode**
  - JavaScript* source is converted into C string and embedded into zephyr.bin image
  - Only the embedded JavaScript application is executed

- **Developer Mode**
  - JavaScript application is executed from Zephyr OS filesystem
  - JavaScript application replaceable via USB or BLE using browser IDE or CLI tool

*Other names and brands may be claimed as the property of others.
$ git clone git@github.com/01org/zephyr.js
$ cd zephyr.js
$ make DEV=ashell

Building...

zephyr.bin

$ make dfu

*Other names and brands may be claimed as the property of others.
Browser IDE

- Only in DEV mode
- Copy-n-Run
- 3rd Party IDEs
- CLI Tools
- Web USB

*Other names and brands may be claimed as the property of others.
### JavaScript* APIs

<table>
<thead>
<tr>
<th>API</th>
<th>Zephyr OS</th>
<th>Node.js</th>
</tr>
</thead>
<tbody>
<tr>
<td>Events, Timers, Console</td>
<td>Now</td>
<td>Core API</td>
</tr>
<tr>
<td>Buffer</td>
<td>Now</td>
<td>Core API</td>
</tr>
<tr>
<td>BLE</td>
<td>Now</td>
<td>Bleno NPM</td>
</tr>
<tr>
<td>GPIO, I2C, AIO, PWM, UART</td>
<td>Now</td>
<td>Johnny-Five like</td>
</tr>
<tr>
<td>OCF</td>
<td>Now</td>
<td>IoTivity-node NPM</td>
</tr>
<tr>
<td>CoAP</td>
<td>Planned</td>
<td>CoAP NPM</td>
</tr>
<tr>
<td>MQTT</td>
<td>Planned</td>
<td>MQTT NPM</td>
</tr>
<tr>
<td>W3C Sensors, Performance</td>
<td>Now</td>
<td>TBD</td>
</tr>
<tr>
<td>UDP</td>
<td>Now</td>
<td>Core API</td>
</tr>
<tr>
<td>TCP, HTTP</td>
<td>Coming</td>
<td>Core API</td>
</tr>
<tr>
<td>File System</td>
<td>Coming</td>
<td>Core API</td>
</tr>
</tbody>
</table>

*Other names and brands may be claimed as the property of others.*
Web USB
Web USB - Why It Matters

- USB is a de-facto standard for connecting devices over short distances
- It is fast, reliable and inexpensive
- It allows for powering devices while communicating
- Advances over Bluetooth® and other wireless techs
  - Faster and more reliable - just works
  - Works with laptop, desktop and phone
  - Allows for powering the device
  - Can be used in places where signals are disallowed (planes, hospitals)
  - Access to a device can be guarded physically
Introducing Web USB

- A new W3C standard allowing web sites and apps access to USB devices
- New USB headers creates further security and allows for a popup when plugging in the peripheral
- Headers will become optional, but are recommended and required for popup
- Works in Chrome* today on Linux*, Macintosh*, and Windows*

*Other names and brands may be claimed as the property of others.
Issues Web USB

- **Linux**
  - modemmanager hijacks USB CDC devices unless their VID/PID are blacklisted

- **Windows**
  - Does not autoload Web USB driver but requires MS OS descriptors (or INF file)
  - Earlier versions (< 10) of Windows, requires a signed "driver", which is basically a signed INF text file, binding the VID/PID to usbser.sys (the default USB serial driver)

- **Macintosh**
  - No issues!

*Other names and brands may be claimed as the property of others.
Web Application
```
// Copyright (c) 2016, Intel Corporation.

console.log("Starting Button example...\n
text = require("text")

var pinMode = require("pinmode");
var buttons = require("buttons2");
var led = pinMode.open;

var led = pins.open()
plg = pins.leds;
direction = "out"

var button = pins.open()
plg = pins.leds;
direction = "in"
edge = "any"

button.onchange = function(event) {
led.write\n
---

CONSOLE
```

`help` This help
eval Evaluate JavaScript in realtime
clear Clear the terminal screen
load (FILE) Saves the input text into a file
run (FILE) Runs the JavaScript program in the file
parse (FILE) Check if the JavaScript syntax is correct
stop Stops current JavaScript execution
ls (FILE) List directory contents or file stat
cat (FILE) Print the file contents of a file
du (FILE) Estimate file space usage
rm (FILE) Remove file or directory
mv (SOURCE) (DEST) Move a file to destination
rmself (TODD)
reboot (TODD)
test Runs your current test
error Prints an error using JerryScript
ging Prints (POWER) to check that we are alive
at OK used by the driver when initializing
set (INPUT MODE) For -load"Accept Data
transfer (raw)
transfer (json)
get Get states on the shell
reboot Reboots the device
```
JavaScript* Web IDE for Zephyr OS

- Browser only Web Application - no server component
- JavaScript code editor (Monaco from Microsoft*)
- Console (Google* console module)
- Board Viewer (new)
- Web USB for device communication
- Multiple tabs and devices
- Git Hub Integration
- Angular2

*Other names and brands may be claimed as the property of others.
JavaScript Web IDE for Zephyr OS (#2)

- Open Source
  - https://github.com/01org/zephyrjs-ide
- Live site
  - https://01org.github.io/zephyrjs-ide/
- Apache* 2.0

*Other names and brands may be claimed as the property of others.
Next Steps and Summary
Next Steps

- More boards to board viewer
- Communication between panels
- Proper API documentation to Monaco
- Plugins to other IDEs
- Flashing with Web USB
- Access to local file system
- WebUSB changes
- Arduino 101* board ROM size
- Testing with other boards

*Other names and brands may be claimed as the property of others.

Join the Open Source Project!

We are looking for contributions

https://github.com/01org/zephyr.js
https://github.com/01org/zephyrjs-ide
Summary

- Easy to use development environment
- Lower the entry barrier to start IoT Projects
- Good fit for class rooms, hackathons and demos

Join the Open Source Projects!

- We are looking for contributions
- [https://github.com/01org/zephyr.js](https://github.com/01org/zephyr.js)
- [https://github.com/01org/zephyrjs-ide](https://github.com/01org/zephyrjs-ide)
Questions ?
Try it Out @Hackathon
Wed 1-4 pm
Forum Suite
Space is limited, registration required

http://sched.co/9XjO