JerryScript and IoT.js
Plans, Status and Issues

Sakari Poussa

Intel® Open Source Technology Center

Plans

- Provide Node.js like JavaScript runtime and APIs to Zephyr
- Port JerryScript and IoT.js to Zephyr
- Initial hardware target is Arduino 101
  - Intel Curie and ARC MCUs
  - 192 K flash
  - 80 K RAM
- JavaScript APIs
  - All of IoT.js APIs
  - OIC/OCF
  - BLE and NFC
- Use Soletta or libtuv event loop or create our own
- Investigate if and how IoT.js can use duktape as alternate JS engine

Status

JerryScript

- Built minimal profile and runs on Zephyr
- Uses newlib instead of jerry-libc
- JS test cases: 801/940 pass on Zephyr
- Simple GPIO LED blinking works
- Using about 140 K of 192 K available flash
    - 152 K with Soletta

IoT.js and libtuv

- Attempted to build GPIO, Buffer, and Process for Zephyr modules, but ran into issues
- Built libtuv for Zephyr w/o TCP support

Issues

JerryScript

- Public API is leaking internals to the embedder. Makes it difficult to be replaced (wrt iot.js and duktape)

IoT.js and libtuv

- IoT.js
  - Contains C++. Zephyr is lacking new/delete support at the moment
  - Contains JavaScript which makes it unnecessarily big
  - No modular builds (to build only APIs the app uses)
- libtuv
  - No UDP support
  - Not following the libuv upstream

Next Steps

JerryScript

- Build additional support beyond minimal (like array) if needed

IoT.js and libtuv

- IoT.js
  - Drop
- Libtuv
  - Drop
- Use Soletta mainloop and/or write a new one from scratch

Demo: Zephyr, Arduino 101 and blink.js

```
function GPIO() {
}

GPIO.pin_configure = function(pin) {
    // call native c gpio_pin_configure api
    gpio_pin_configure(pin);
}

GPIO.pin_write = function (pin, value) {
    // call native c gpio_pin_write api
    gpio_pin_write(pin, value);
}

print("Start of GPIO sample run...");

gpio_pin = 8
GPIO.pin_configure(gpio_pin);

toggle = 0;
var blink = function () {
    toggle = 1 - toggle
    GPIO.pin_write(gpio_pin, toggle);
}

// setup to call blink every 1000ms
setInterval(blink, 1000);
```