Soletta Technical Introduction

Otavio Pontes

OTC - Intel
IoT Questions and Problems

- Explosion of libraries
  - Subsets: I/O, comms
  - Specific targets
  - Too big for small devices
    - Hard to reuse knowledge
- Nothing is integrated
- Lack of documentation
What is Soletta?

- IoT Framework
- Open Source
- Easy access:
  - Sensors
  - Actuators
  - Communication
- Portable code
- Different platforms, including small OSs
## Architecture

### Soletta

<table>
<thead>
<tr>
<th>Machine Learning</th>
<th>Flow</th>
<th>OIC</th>
<th>MQTT</th>
<th>HTTP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Services</td>
<td>Network</td>
<td>Update</td>
<td>Crypto</td>
<td>Event dispatching</td>
</tr>
<tr>
<td>GPIO</td>
<td>SPI</td>
<td>UART</td>
<td>I2C</td>
<td>PWM</td>
</tr>
</tbody>
</table>

Hardware and Operating System Abstraction Layer

<table>
<thead>
<tr>
<th>System Libs</th>
<th>Comms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kernel</td>
<td></td>
</tr>
<tr>
<td>Hardware</td>
<td></td>
</tr>
</tbody>
</table>
Flow layer

- Domain-specific language (DSL)
- Interface
- Easy - target audience
- Code safety
- Visual
- OOBE
How to remotely toggle a light bulb?

Simple and canonical example
How to remotely toggle a light bulb?

Server

- light network resource
- relay switch
- VCC

Client

- sensor
- light network resource
- feedback led
- network protocol
- VCC
How to remotely toggle a light bulb?

Server

light network resource

relay switch (gpio/writer)

network protocol

Client

Sensor (gpio/reader)

light network resource

feedback Led (gpio/writer)
How to remotely toggle a light bulb?

#Server fbp file
light(oic/server-light)
led(LED)

light STATE -> IN led

#Client fbp file
button(Button)
#Update device_id with server device id
light(oic/client-light:device_id="")
feedback_led(LED)

button OUT -> STATE light
light STATE -> IN feedback_led
How to remotely toggle a light bulb?

## Server config file
```
{
    "name": "LED",
    "options": {
        "pin": "7"
    },
    "type": "gpio/writer"
}
```

## Client config file
```
{
    "name": "LED",
    "options": {
        "pin": "3"
    },
    "type": "gpio/writer"
},
{
    "name": "Button",
    "options": {
        "pin": "5"
    },
    "type": "gpio/reader"
}
```
How to remotely toggle a light bulb?

Running on the target board, supports multiple users and allows fetching git repositories.
How to remotely toggle a light bulb?

```plaintext
# gpioreader
GPIO reader

- **INPUT PORTS**
  - OUT | boolean
    - GPIO reader, port out

- **OPTIONS**
  - pin | string
    - Label of the desired pin on the board. If raw is set to true, this should be the pin number as recognized by the platform.
  - raw | boolean
    - Change 'pin' meaning to be the system parameters needed to address the desired pin. Use it if you know what you are doing.
  - poll_timeout | int
    - Polling time
  - active_low | boolean
    - Is active low
  - edge_rising | boolean
    - Is edge rising
  - edge_falling | boolean
    - Is edge falling
  - pull | string
    - Up for pull up, down for pull down, none for no pull

# gpiowriter
GPIO writer

- **INPUT PORTS**
  - IN | boolean
    - GPIO writer, port in

- **OUTPUT PORTS**
  - OUT | boolean

- **OPTIONS**
  - pin | string
    - Label of the desired pin on the board. If raw is set to true, this should be the pin number as recognized by the platform.
  - raw | boolean
    - Change 'pin' meaning to be the system parameters needed to address the desired pin. Use it if you know what you are doing.
  - active_low | boolean
    - Is active low
```
How to remotely toggle a light bulb?

This is a regular application, that can be installed alongside other applications and frameworks.

Image generator packs the application and the operating system into a single image to be flashed on the IoT device.

Users can run their FBP without the need to compile. Soletta Development Application uses this.
How to remotely toggle a light bulb?

#Client config file to be used in a desktop
{
  "name": "LED",
  "type": "gtk/led"
},
{
  "name": "Button",
  "type": "gtk/button"
}
Another example: Custom node
Soletta Support
Operating Systems support

- Linux
- Zephyr
- RIOT
- Contiki
Boards tested so far

- Intel Edison (Linux)
- Intel Galileo Gen 2 (Linux)
- Intel Minnowboard Max (Linux)
- Quark SE Dev Board (Zephyr)
- Atmel SAMR21 Xplained Pro (RIOT)
- Raspberry Pi (Linux)
Small OSes measurements

- Zephyr on Quark SE Dev Board
- Image size: 107k
- Peak used RAM: around 32k
How to get involved
**Roadmap**

**Go Stable (Embedded World)**
- Initial Public announcement
- Stable API (1.0)
- 100% API documentation
- OS: Linux, Zephyr (initial)...
- Languages: C/C++, FBP, JS
- Comms: OIC, CoAP, MQTT, HTTP
- Real-world usage samples

**v2.0**
- **Zephyr**: full I/O, persistence and 6loWPAN
- **FBP**: community packages (similar to npm)
- **Node.js**: full bindings

**v3.0**
- **Zephyr**: BLE, HTTP, MQTT, OTA updates
- **Comms**: BLE, Bluetooth, Management (ConnMan)

**v4.0**
- **Zephyr**: Small JS on 80Kb and FBP on 8Kb
How to get involved

Cool stuff we want to do

- Language bindings
- Visual Editor
- Plugins for other IDEs
- Communication protocols
How to get involved

Community

- GSoC
- Workshops / Talks
- Partnerships
- #soletta @ freenode
- Mail lists - https://lists.solettaproject.org/
- Site - https://solettaproject.org/
- Repos - https://github.com/solettaproject
Q&A

Thanks

Otavio Pontes - otavio.pontes@intel.com