

Electric Imp: Overview

Table of Contents

1 Overview	1
1.1 Event-driven programming	1
1.2 Interactive imp	1
2 Objects	1
2.1 Overview	1
2.2 The hardware object	1

1 Overview

1.1 Event-driven programming

- **Device code:** Device (code on Device) reacts to Agent (code on cloud)

```
// on reception of notification headed
// "text_to_display" from the agent,
// call "led_matrix_print"
agent.on("text_to_display", led_matrix_print);

function led_matrix_print(message_to_display) {
    display_line(message_to_display);
    agent.send("message_displayed", true);
}
```

- **Agent code:** Agent (code on cloud) reacts to Device (code on Device)

```
// HTTP request handler registration
http.onrequest(request_handler);

function request_handler(request, response) {
    try {
        if ("message" in request.query)
            device.send("text_to_display",
                        request.query.message);
    }
    catch (ex) {
        device.send("error_to_display",
                    "Internal Server Error: "
                    + ex);
    }
}
```

- **Timer event**

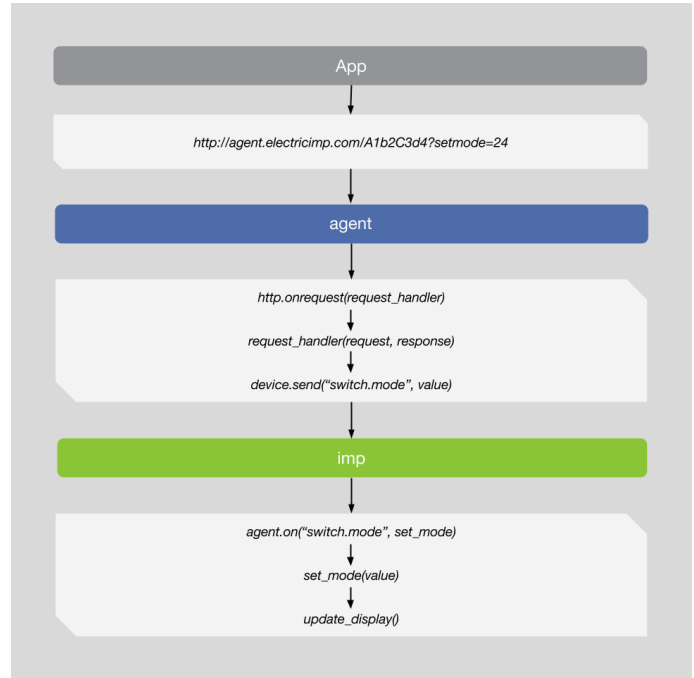
```
function loop() {
    // Gets integer from imp's light sensor,
    // converts it to string and relays via UART
    local current_light = hardware.lightlevel();
    uart57.write("Current light level is: " +
                current_light);

    // Set the imp to check again in one second

```

```
    imp.wakeup(1.0, loop);
}
```

1.2 Interactive imp



2 Objects

2.1 Overview

- **object:** software construct that combines data (called “properties”) and functions (Called “methods”)

2.2 The hardware object

The **hardware** object represents an imp’s IO peripherals and is used to control or monitor the hardware in the imp. The device operated by either an imp card (imp001) or a solder-down imp module (imp002, imp003/Murata LBWA1ZV1CD).

The hardware object has the following member methods:

- **hardware.getdeviceid():** returns the device’s unique ID as a string
- **hardware.lightlevel():** reads the imp’s light-level sensor
- **hardware.micros():** returns current value of a free-running microsecond timer
- **hardware.millis():** returns current value of a free-running millisecond timer
- **hardware.voltage():** returns the imp power supply voltage
- **hardware.wakereason():** returns the reason the imp woke up

The hardware object also has the following deprecated member methods, which should not be used in new code, but are documented here as an aid to understanding and migrating old code:

- `hardware.configure(const,...)`: deprecated and inflexible way to configure the hardware
- `hardware.getimpeedid()`: returns the device's unique ID as a string