

Building a Connected Plant

An IoT613 Technical Workshop

About Us

Daniel Lanthier: Development Manager @ macadamian

Joel Lord: Development Manager @ macadamian

Plant: Office Shrub @ macadamian. Also @office_shrub on twitter.

About today's workshop

We will connect lots of plants to the internet!

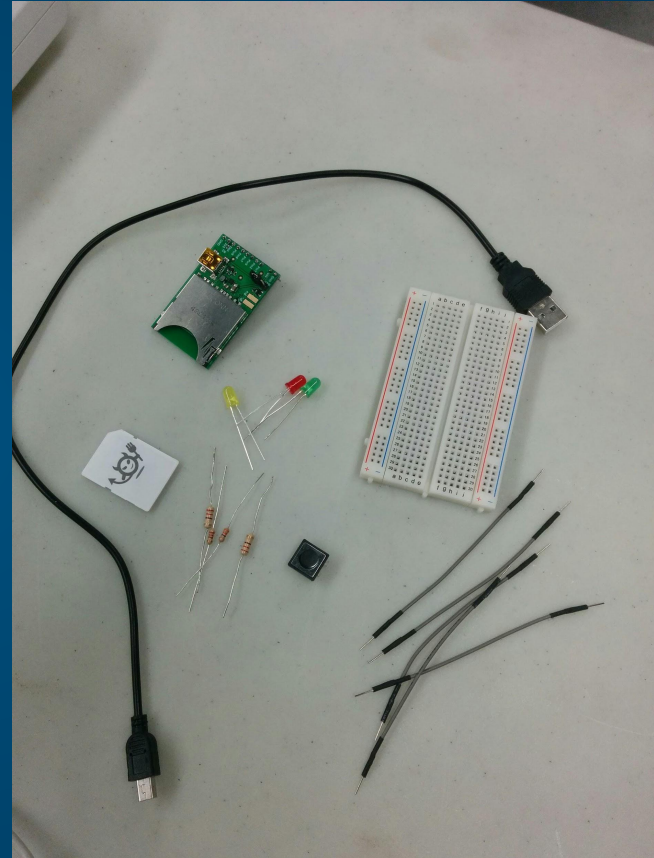
What you need:

- The sensor kit included.
- A laptop with a web browser and a free USB port.
- Information Sheet.
- Access to our GitHub: <http://tiny.cc/loT613-GH>
- Your code-fu.

Sensor kit - quick check

What you need:

- A breadboard
- A circuit board
- An Electric Imp
- 4 resistors (220 ohms)
- 3 LEDs (red, yellow, green)
- A photoresistor
- A button
- 3 small cables



The “Problem” Statement

- Plants are pretty dumb.
- I don’t want to have to walk up to it to know how it’s doing.

Solution:

- A connected plant, that provides:
 - Light Level
 - Water Level
 - A button to make it tweet

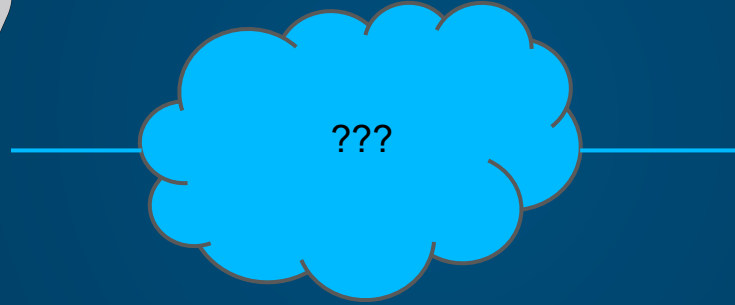


Why would you possibly want a tweeting plant?

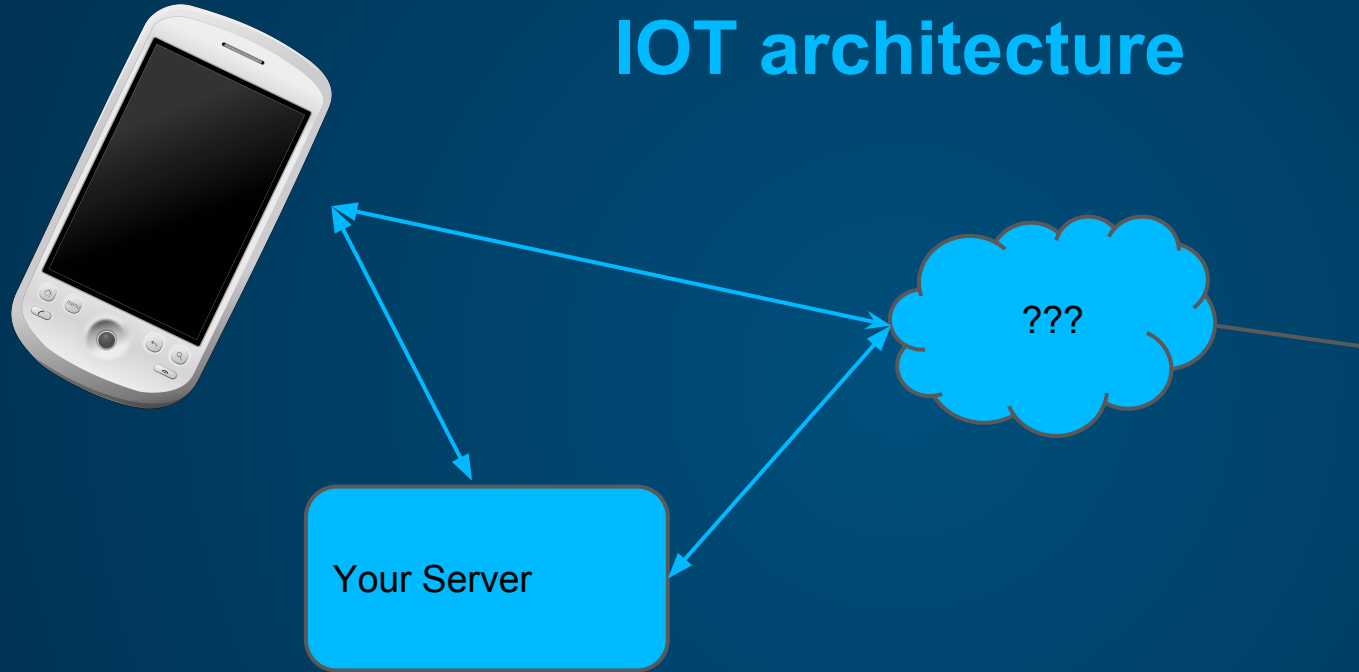
Adds features to any boring neighborhood shrub:

- The plant tweets when it is watered!
- The plant tweets in the morning when it wakes up about how his previous day went.
- The plant tells us how it feels on a website, LEDs, and tweets when you press a button.

DYI project IOT architecture



Real world IoT architecture



The Electric Imp Platform

- Cloud-based computing (agents)
- Device-side computing (device)
- Uses JavaScript-like language called Squirrel.
- Pre-certified by the FCC



Demo: @office_shrub



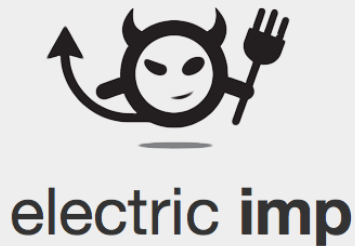
“You pressed my button! I feel dark
and wet”

-- @office_shrub, on twitter.

Okay, now it's your turn

Create an Electric Imp Account

Create an account at ide.electricimp.com



Sign In

[Forgot Password?](#)

Sign Up

By clicking Sign Up, you agree to our [Terms](#).

Create a model for your code

Don't close ide.electricimp.com!

The screenshot shows the electric imp IDE interface. The background displays the 'Code' tab with a file named 'iot613-0-0' and a code editor showing JavaScript code. The left sidebar contains a search bar with the text 'device name or Id', a 'Create New Model' button, and a list of development models including 'Blob', 'Connected-Plant-step1', 'flash-green-fast', and 'iot613-final'. The 'iot613-final' model is selected, showing its details. Overlaid on the interface is a 'Create New Model' dialog box. The dialog has a title bar, a 'Name' field with the placeholder text 'Enter a new model name...', and a table of 'Assigned Devices'. The table lists five devices with their IDs and names. At the bottom of the dialog are 'Cancel' and 'Create Model' buttons.

Create New Model

Name:

Assigned Devices:

<input type="checkbox"/>	231c76aeaf952dee	iot613-final
<input type="checkbox"/>	Highstead Z Board	Blob
<input type="checkbox"/>	232a6dafaf952dee	flash-green-fast
<input type="checkbox"/>	233025b030728cee	Connected-Plant-step1
<input type="checkbox"/>	233646068fb7bdee	flash-green-fast

Electric Imp Editor

Agent (Cloud) on the left side.
Device Code on the right side.
Logging at the bottom.

The screenshot displays the Electric Imp Editor interface. On the left sidebar, there is a search bar with the text "device name or id" and a "Create New Model" button. Below this, a list of development models is shown, including "Connected-Plant-step1" which is selected and highlighted. The main area is titled "Connected-Plant-step1" and shows the device ID "233025b030728cee". It includes buttons for "Build 6", "Check", and "Build and Run". The interface is split into two panels: the left panel shows the Agent code (cloud-side) and the right panel shows the Device code (hardware-side). The Agent code is in JavaScript and includes a function to set registration and a function to set alert mode. The Device code is in C and includes a function to blink the LED. The status of the agent is "not running".

device name or id

Create New Model

Unassigned Devices 0

Development Models

Blob

Connected-Plant-step1

233025b030728cee

flash-green-fast

Connected-Plant-step1

233025b030728cee / device offline / agent not running / agent link

Build 6 ✓ Check ▶ Build and Run

Agent - <https://agent.electricimp.com/nPSD1DkY9-o-> not running

```
1 |
2 local serverURL = "http://iot613-officeshrub.azurewebsites.net";
3 local agentId = split(http.agenturl(), "/")[2];
4
5 server.log("Request sensor data: " + http.agenturl() + "?all=read");
6 server.log("Set Registration: " + http.agenturl() + "register=true");
7 server.log("Set Alert Mode: " + http.agenturl() + "?alert&light=on");
8
9 function setRegistration(request, response) {
10     device.send("setRegistration", registrationValue);
11     device.on("onSetRegistration", function(value) {
12         response.send(200, "Device Registered");
```

Device - 233025b030728cee

```
1 onlineLed <- hardware.pin9; // registration state,
2 onlineLed.configure(DIGITAL_OUT); // DIGITAL out means a
3
4 // reset our values
5 local isRegistered = false;
6
7 function blink() {
8     if(!isRegistered) {
9         return;
10     }
11
12     onlineLed.write(1 - onlineLed.read());
```

Electric Imp Application + BlinkUp

Install the mobile application, and BlinkUp your device.
Once BlinkUp, configure your device with the model that you created.

WiFi Credentials: MSNGuest / M3mb3r5!

Success!

The SD Card's is blinking green.

Uhh?

SD Card is blinking Red, or Orange. Restart the imp, and reboot right away.

Create a Twitter Account

- Check your information sheet to enable an application.
- Browse to <https://apps.twitter.com/>
 - Please notify your followers that you might be spammy for the next 60 minutes.
- The important part is under Keys and Access Tokens.

Keep these values handy:

Consumer Key (API Key)

Consumer Secret (API Secret)

Access Token

Access Token Secret

We are ready!

Building the project

Visit our Github: <http://bit.ly/1Mq5qMh>.

Visit <http://bit.ly/1KvaNJq> for our dashboard.

Folder for each step:

Step #1: Hello World!

Step #2: Tweeting Button

Step #3: Water Sensor

Step #4: Light sensor

Let's Hack!



Contact Us



dlanthier@macadamian.com @DC_LAN

jlord@macadamian.com @joel__lord

Thank you.

