

LinkIt smart 7688 e-book for Node.js community.

blue chen

Published
with GitBook

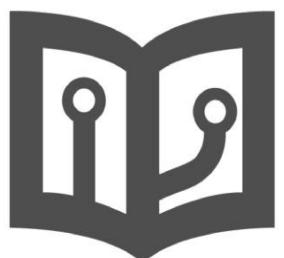


Table of Contents

Introduction	0
Introduction	1
LinkIt smart 7688 Spec	1.1
Initial Setup	1.2
How to setup WiFi mode?	1.3
How to ssh into LinkIt smart 7688?	1.4
(Optional)How to reset linkIt smart 7688	1.5
(Optional)How to update firmware ?	1.6
Basics	2
Running your first 'Hello World' program on LinkIt Smart 7688	2.1
Installation Node.js suite on LinkIt smart 7688	2.2
Use mraa to control GPIO	2.3
Open files on microSD	2.4
How to use video streaming	2.5
Controlling LED with firmda	2.6
Controlling LED with cylon.js	2.7
How to make your Node.js app start automatically on LinkIt Smart 7688 booting?	
Cloud Service (Mediatek Cloud Sandbox)	3 2.8
Let MCS control LinkIt smart 7688 by using MCSjs	3.1
Use MCS + firmata to control LED	3.2
Uploading data point to MCS	3.3
Use MCS FOTA to upgrade Arduino firmware	3.4
Make a remote tank with LinkIt smart 7688 Duo and MCS gamepad channel	3.5
Emulator	4
Education	5
Web of things	6
Classic Examples	7
Use touch sensor to trigger LED	7.1

LinkIt smart 7688 x Nodejs

前言

本電子書適合給想在 LinkIt smart 7688 版子上面開發豐富的 Nodejs application 的玩家們閱讀。

同時也因應世界潮流，本電子書會未來會循序漸進至少提供2種語言(中文. 英文)

Preface

This e-book is suitable for the players who want to develop various kinds of Node.js applications in LinkIt smart 7688 board to read. Besides, in response to the global trend, this e-book will be gradually provided at least in two languages (Traditional Chinese. English.)

注意事項

- 若您對本書細節有任何疑問想修改，歡迎至此 github 提交您的 PR：
<https://github.com/iamblue/linkit-smart-nodejs>。
- 本電子書也向大眾徵稿，如果您有更好的 idea 或者是文章想貢獻歡迎至 github 提交您的 PR (請記得檔案/文章/內文要符合規範) 審核通過後我會把它納入本書章節之中。

Precautions

- If you have any questions about the details of the book you want to modify, welcome to go to this github and submit your pull request.
- This e-book is also called for papers to the public, if you have a better idea or would like to contribute articles, welcome to go to the github and submit your pull request.
(Remember all archives / articles / contents should be meet the specifications.) After auditing, I'll include them in the corresponding chapter in this book.

Introduction to LinkIt smart 7688

The following series of chapters introduce what LinkIt smart 7688 is.

Preface

Mediatek Inc. released two LinkIt smart series boards on 2015/12/1 - 7688 and 7688 Duo, respectively.

Framework Diagram

LinkIt Smart 7688



LinkIt Smart 7688 Duo



This simple framework diagram shows the 7688 and 7688 Duo, as we can see, both these two boards use MT7688 chip. 7688 Duo has one more Arduino chip (32U4) than 7688. Simply speaking, users with 7688 Duo can have more options to play with Arduino sensors.

Development Board Specification

During defining the specifications for these two 7688 development boards, the memory and flash suitable for high level language players have been surveyed. Under this specification, the problems encountered in the npm install could be nearly avoided, for example, ran out of memory, or storage space shortage (even can not install Node.js). Here is the specification:

During defining the specifications for these two 7688 development boards, the memory and flash suitable for high level language players have been surveyed. Under this specification, the problems encountered in the npm install could be nearly avoided, for example, ran out of memory, or storage space shortage (even can not install Node.js). Here is the specification:

LinkIt Smart 7688



- MT7688
- **128MB DDR2**
- **32MB Flash**
- Chip Antenna (default)
- IPEX Ext Antenna
- microSD
- USB Host
- JTAG

LinkIt Smart 7688 Duo



- MT7688 + **ATmega32U4**
- 128MB DDR2
- 32MB Flash
- Chip Antenna (default)
- IPEX Ext Antenna
- microSD
- USB Host
- JTAG

About MT7688

Below is the specification of MT7688, we can see that it is kind of AP router chip product, and it is belong to MPU grade chip.

MT7688AN SOC Specifications	
CPU	MIPS24KEc (580 MHz)
Interface	Count
Total DMIPs	580 x 1.6 DMIPs
I-Cache, D-Cache	64 KB, 32 KB
L2 Cache	N/A
Memory	DDR1/DDR2 16 bits Max. 2 Gb, 193 MHz
SPI Flash	3B addr mode (max 128 Mbit) 4B addr mode (max 512 Mbit)
SD	SD-XC (class 10)
RF	1T1R 802.11n 2.4GHz
Package	DR-QFN156-12 mm x 12 mm
PCIe	1
USB 2.0	1
Fast Ethernet Switch	5
I2S	1
PCM	1
PWM	4
SPI	1
I2C	1
UARTLite	3
JTAG	1

Table 1 MT7688AN SOC Specification

Initial Setup

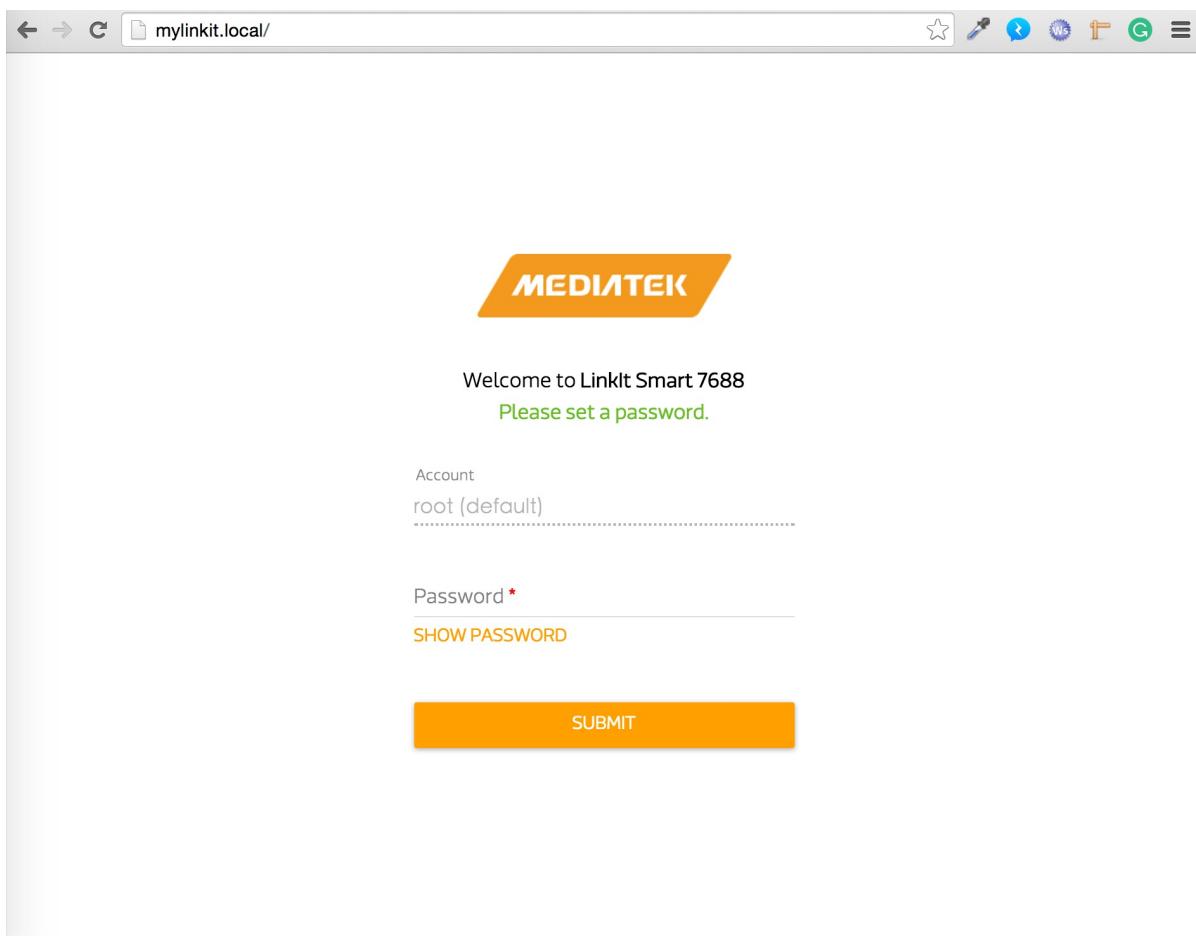
Preparation

- One microUSB cable. Please attach the USB connector to the PC or electric outlet, and the microUSB connector to the microUSB connector on the beveled side of LinkIt smart 7688 board (i.e. the USB Power shown in the following figure).

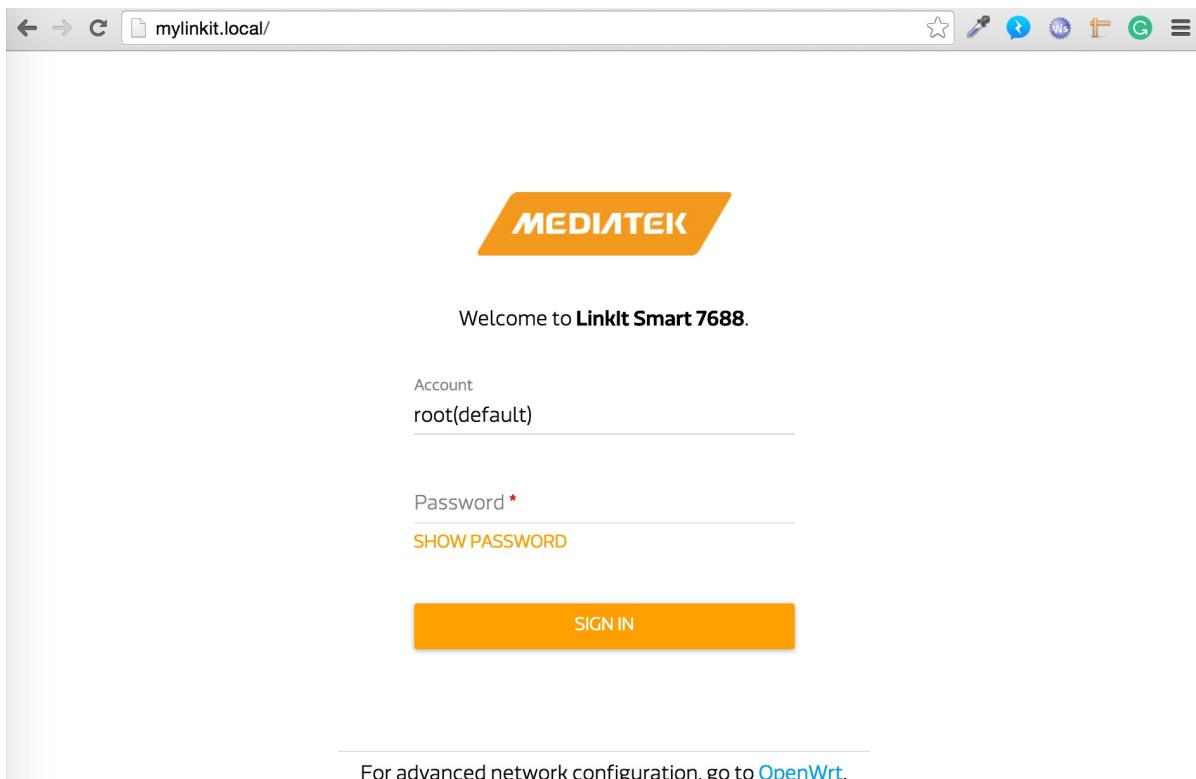


Steps

- Open your PC and scan WiFi AP.
- Choose `Linkit_Smart_7688_xxxxxx`, after your PC connected to the LinkIt smart 7688 board, make sure to see the orange light starting to flash on the LinkIt smart 7688 board.
- Open your browser, type <http://mylinkit.local/> in your URL command line.
- Now you will see the screen like this:



- The initial password needs to be set in the first time usage for 7688. Please enter the initial password you want to setup in this screen.
- After finishing the password setup, it will transfer to the login page automatically.



- Enter the password you just typed, if it is correct, it will transfer to the page like this:

The screenshot shows the LinkIt Smart 7688 dashboard. The top navigation bar includes the MediaTek logo, language selection ('English'), and a 'Sign out' link. The main content area has a header 'Welcome to MediaTek LinkIt Smart 7688' and a note 'For advanced network configuration, go to OpenWrt.'. Below this, there are two tabs: 'System information' (which is active) and 'Network'. The 'System information' tab displays 'Platform information' with 'Device name' set to 'mylinkit' and 'Current IP address' set to '192.168.100.1'. It also shows 'Account information' with the same 'Account' and 'Password *' fields as the login page. A large orange 'CONFIGURE' button is located at the bottom of this section. A 'Software information' tab is visible at the bottom.

- Link successfully!

It is strongly suggested to read the following chapters for players.

How to ssh into LinkIt smart 7688?

[Click to read](#)

How to burn Arduino code into LinkIt smart 7688 Duo?

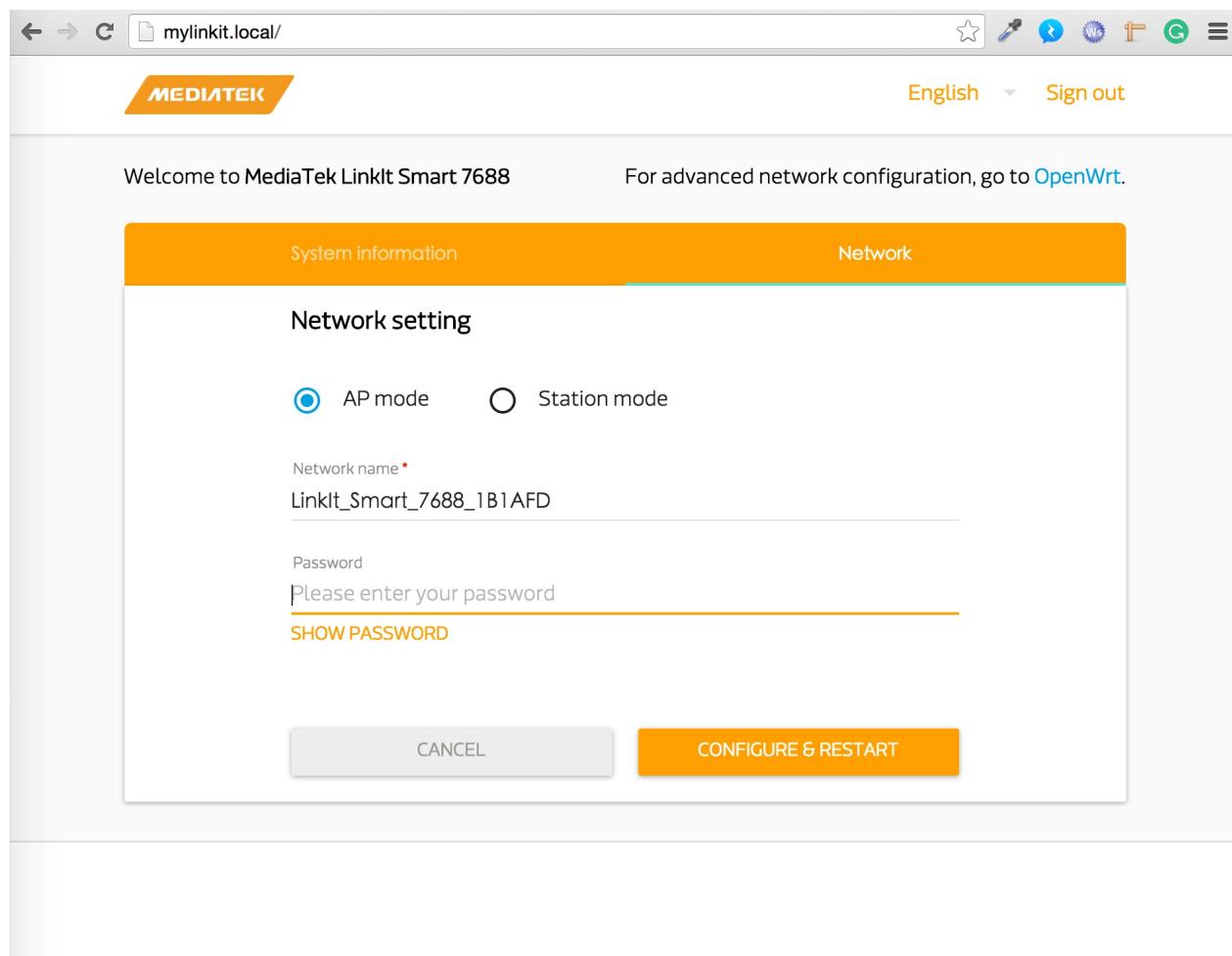
[Click to read](#)

How to setup WiFi mode (AP mode or Station mode)?

[Click to read](#)

How to setup WiFi mode?

- Enter the WebUI of your board.
- Choose Network panel as below:



- There are two modes in LinkIt smart 7688, you can choose either one based on your preferences:
 - AP mode: Use LinkIt smart 7688 as a hot spot (as the cellular phone hot spot), and let other devices (for example, your PC, cell phone, ...etc) can be connected to your LinkIt smart 7688 through WiFi.
 - Station mode: Let LinkIt smart 7688 connected to other hot spot. If you need your LinkIt smart 7688 with networking capability, please use this mode to connect to other hot spots with internet connection.
- Press configure & restart, and should be ready to use after waiting for a few minutes.

How to ssh into LinkIt smart 7688?

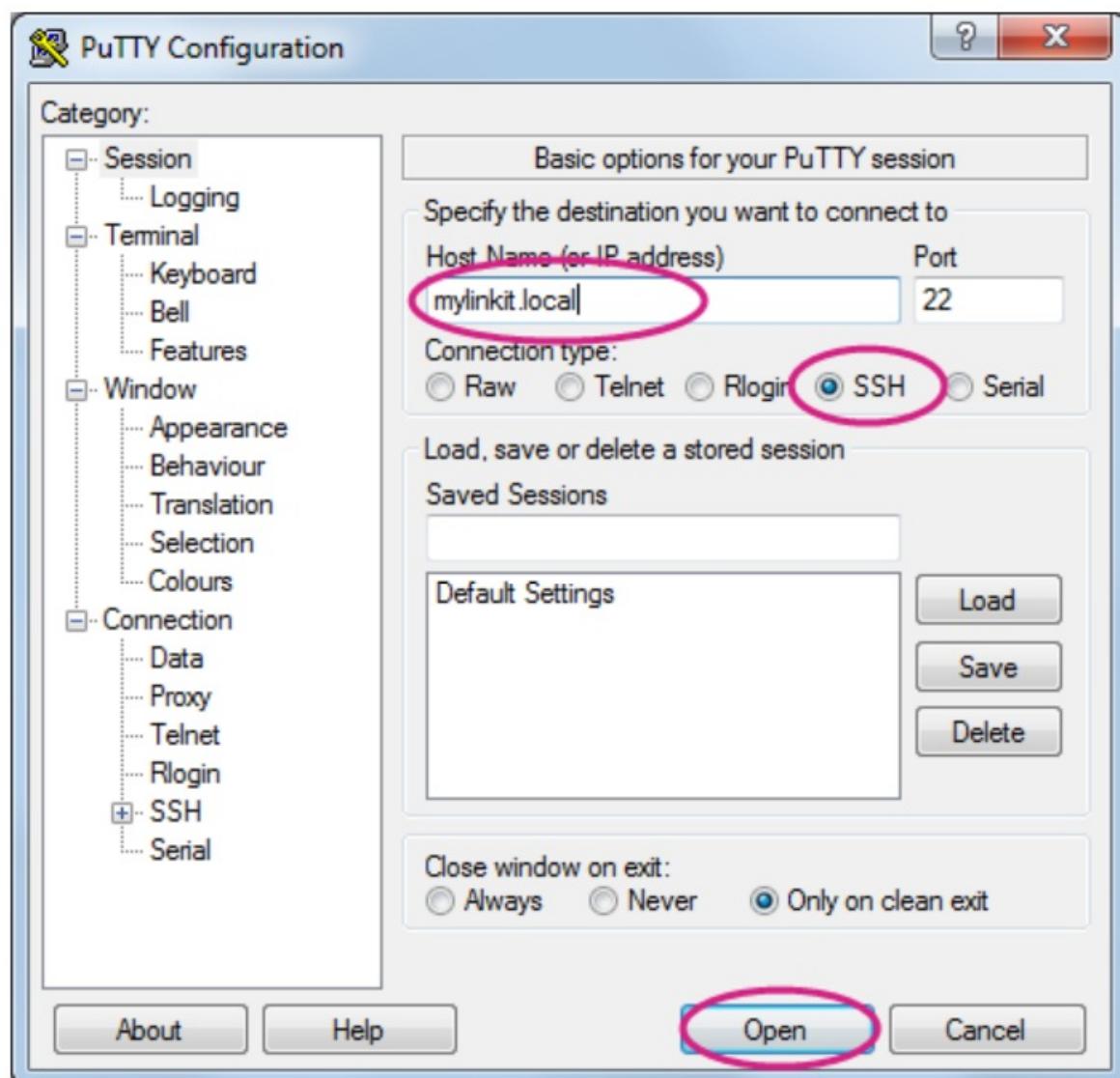
Mac & Linux

- Make sure that your PC is in the same network or listens to the same WiFi AP with LinkIt smart 7688.
- Open your command line and type:

```
ssh root@mylinkit.local
```

Windows

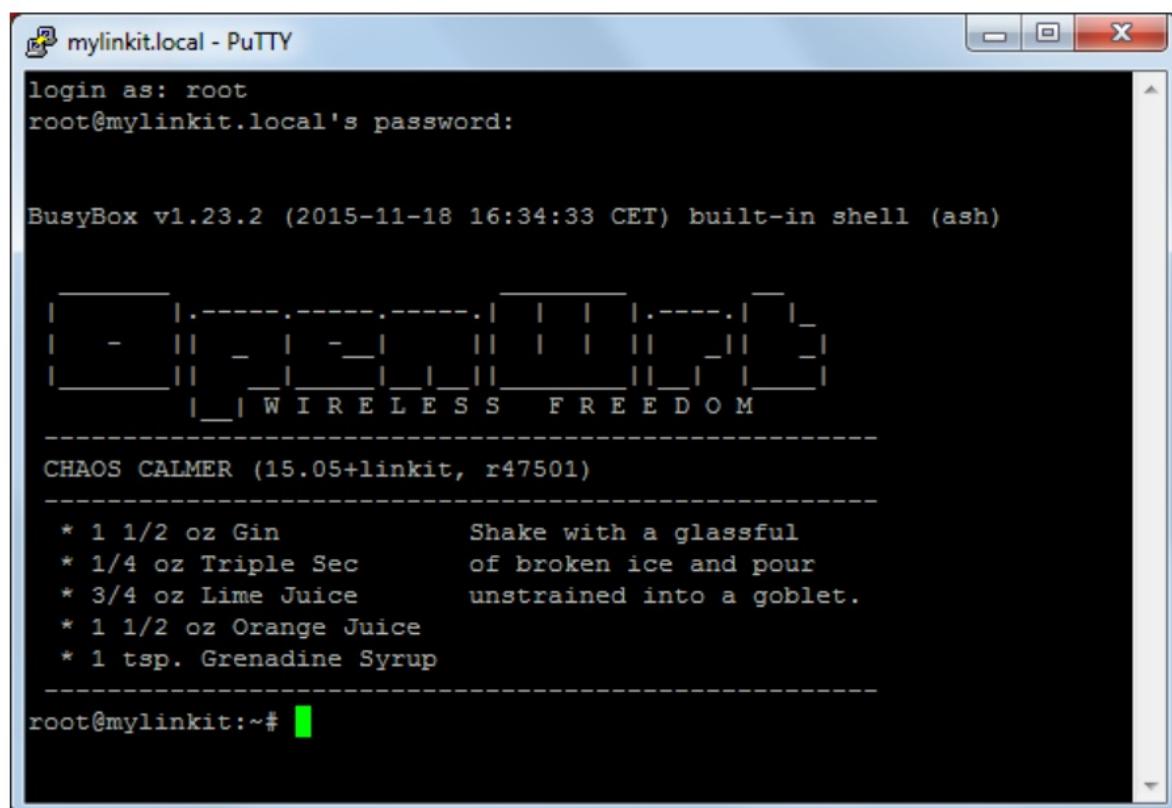
- Open your PuttY, an setup as follows:



- If you see this picture, please choose “Yes”.



- After that, you can enter to the terminal:



```
mylinkit.local - PuTTY
login as: root
root@mylinkit.local's password:

BusyBox v1.23.2 (2015-11-18 16:34:33 CET) built-in shell (ash)

 _ _ _ | .----.----.----.| _ _ _ | .----.| _ |
 | - || - | - | | | | | | | | | | | | | |
 | _ _ _ || | _ _ _ | _ _ _ | | | | | | | | | |
 | _ | W I R E L E S S   F R E E D O M
-----
CHAOS CALMER (15.05+linkit, r47501)
-----
* 1 1/2 oz Gin           Shake with a glassful
* 1/4 oz Triple Sec      of broken ice and pour
* 3/4 oz Lime Juice       unstrained into a goblet.
* 1 1/2 oz Orange Juice
* 1 tsp. Grenadine Syrup
-----
root@mylinkit:~#
```

Note:

1. (Not necessary to read) If you need to change your device name on the WebUI some time in the future (for example, to change to apple.local), please type the following instruction in your command line:

```
ssh root@apple.local
```

How to reset linkIt smart 7688

Via the button on the board

- When the board is plugged in, please wait for a period of time to let it finish the bootup process. (This step must be done.)
- For LinkIt smart 7688 and LinkIt smart 7688 Duo, press the WiFi button (in the middle) for 20 seconds.
- Release the button, then you will see the orange light starts to flash, which represents the board is in the reset process.
- After waiting for a period of time, use your PC to monitor if there is a new `LinkIt_smart_7688_xxxx`. If yes, it stands for completion.

[Please refer to this video](#)

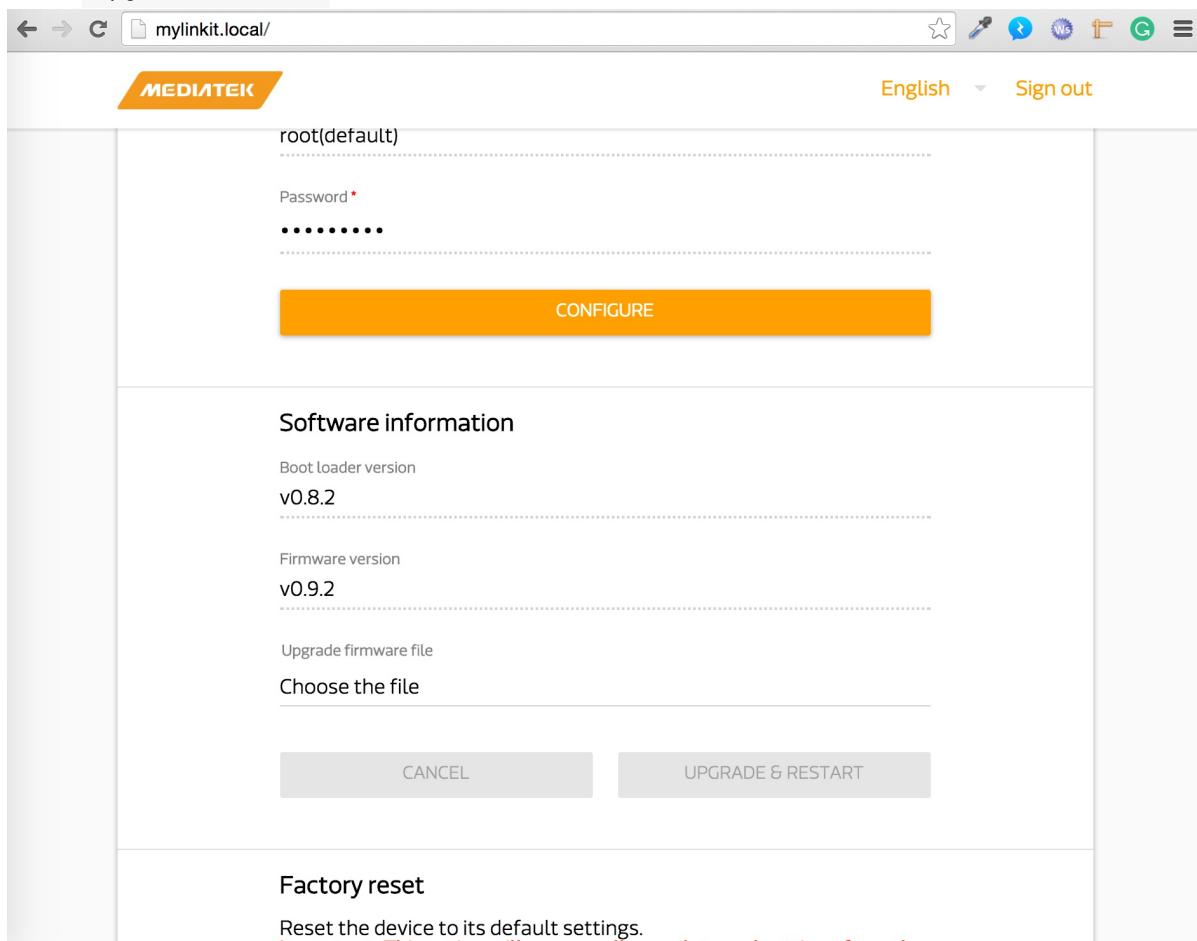
Via WebUI

- Enter WebUI and scroll down to the last page.
- Press Reset and wait for 2 to 3 minutes.

How to update firmware

Via WebUI

- Visit your WebUI, roll to the bottom:
- Click `Upgrade firmware`:



- `choose the file`, select the new firmware file, and upload.
- After the file upload is complete, wait for about 2 ~ 3 min.
- After that, you will find in the wifi ID `LinkIt_Smart_7688_xxxxx`. This indicates that firmware update was successful.

Please refer to this video:



[Video link](#)

Through microUSB + flash drive

Please refer to this video:



[Video link](#)

Basics

This chapter will introduce how to run Node.js applications on LinkIt Smart 7688 board.

Running your first 'Hello World' program on LinkIt Smart 7688

Aim

In this section, we are creating the first `Hello world` program through Node.js.

Steps

- Generate a folder named app

```
> mkdir app
```

- Edit `app.js`

```
> vim app.js
```

- In `app.js`, press `i` and enter the following codes:

```
console.log('Hello world');
```

- Press `Esc`, then enter `wq!` to save and leave.
- Enter `node app.js` and the result can be seen as below:

```
> Hello world
```

- Done!

Installation Node.js suite on LinkIt smart 7688

Aim

This section explains how to install the Node.js suite you need on LinkIt smart 7688.

If you encounter some problems on the installing Node.js suite, you can also find your answer in the following `Troubleshooting` list.

Steps

- Make sure the board is connected with your computer correctly.
- ssh into your board.
- Go into your Node.js project folder.
- For example, I want to install express package, enter:

```
npm install express
```

- If you see the following screen, it is all done. Cheers!

PS. If your package is to be installed on the Node.js global environment, enter:

```
npm install express -g
```

Troubleshooting

Q. `npm install` waits too long or shows connection timed out

A. For connection time out issue, please check your network first. For `npm install` time too long, please check if sort of `node-gyp` ... or similar strings shows up during the npm install process. If there is, that is because cross compile problem. MPU on LinkIt Smart 7688 need longer running time since it is not as powerful as your desktop or notebook computer.

It is recommended if such situation encountered, we need to use openWRT toolchain to build the ipk file, and then install the ipk with `opkg install` on Smart 7688. Related `Makefile` for building `node-gyp` package, please refer to(<https://github.com/openwrt/packages/tree/master/lang/node-serialport>)

Use mraa kit to control GPIO

Aim

In this section, we are going to use `mraa` to control GPIO.

Note: The LED lights to be controlled in this example is the same as wifi one on the board.

Controlling LED (outputting GPIO messages)

Steps

- ssh into Linkit Smart 7688.
- Generate a project.

```
> mkdir app
```

- Use npm to initialize your project.

```
> npm init
```

- Generate an app.'s file.

```
> vim app.js
```

- Enter the content in `app.js`.

```
var m = require('mraa');
var ledState = true;
var myLed = new m.Gpio(44);

myLed.dir(m.DIR_OUT);

function periodicActivity() {
    myLed.write(ledState ? 1 : 0);
    ledState = !ledState;
    setTimeout(periodicActivity, 1000);
}
periodicActivity();
```

- Then, we can see the results!



Open files on microSD

Aim

In this section, we are going to read files in the microSD within LinkIt Smart 7688.

Steps

- Insert a microSD into LinkIt Smart 7688.
- ssh into your LinkIt smart 7688.
- Generate a folder named 'app'.

```
> mkdir app
```

- Generate `app.js` .

```
> vim app.js
```

- Press `i` and edit `app.js` :

```
var fs = require('fs');
// **** Remarks ****
// 1. The following SD-P1 stands for read first partitions in SD card.
// 2. On Linkit Smart, all files will be found in /Media folder no matter whether you pic
fs.readdir('/Media/SD-P1', function (err, data) {
  if (err) throw err;
  console.log(data);
});
```

- 
- Press `esc` , then enter `wq!` to save and leave.
 - Enter `node app.js` , and the results can be seen!

How to use video streaming

Aim

In this section, we are going to create video streaming through mjpg-streamer.

Prerequisites

- webcam: uvc specification is recommended. In this example, Logitech C310 camera is taken.

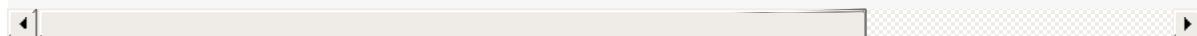
Installation

- Make sure mjpg-streamer is installed on LinkIt Smart 7688.

Work Principle

- Basically, this would be working if `mjpg-streamer` is installed.
- Type the following command in the command line:

```
> mjpg_streamer -i "input_uvc.so -d /dev/video0 -r 640x480 -f 25" -o "output_http.so"
```



- Open `http://mylinkit.local:8080` in your browser. If you see the video, it works properly.

Controlling LED with firmdata

Aim

Caution: Methods described in this section only applies to LinkIt smart 7688 Duo. LinkIt smart 7688 is not allowed to use.

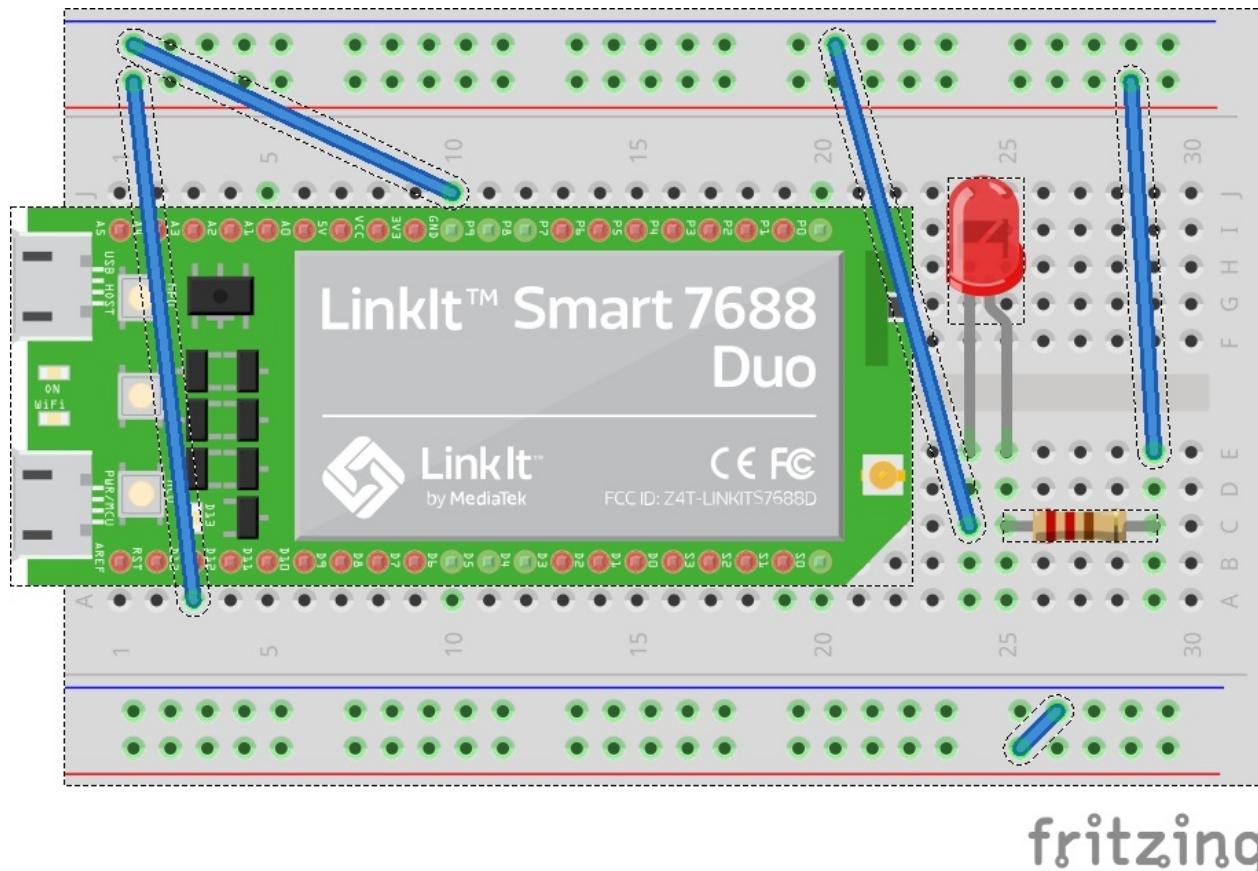
Preface: LinkIt Smart 7688 has two chips, one is the MPU (7688) capable to run linux; the other one is the Arduino MCU. For Node.js developers, it is desired to run Node.js app on the MPU. Arduino MCU is directly controlled by this app. As a result, we are going to use Node.js firmdata package to communicate between MPU and Arduino MCU.

Prerequisites

Control LED requires

- LED x 1
- Resistor x 1
- A few DuPont wires

Please install circuit first:



Steps

MCU side

- Open your Arduino IDE.
- Copy contents in this URL: <https://gist.github.com/edgarsilva/e73c15a019396d6aaef2>
- Burn into Arduino

MPU side

- ssh into your `LinkIt smart 7688 Duo`.
- Generate a folder named `app`.

```
> mkdir app && cd app
```

- Install `firmata`

Notice! Because there are some compilation procedures during npm install, which will cause too long running time on LinkIt smart 7688 Duo, it is not recommended to use `npm install` method for firmata package.

So we will first generate a testfirmata folder on host side (your computer).

```
mkdir testfirmata && cd testfirmata
```

Then execute:

```
npm init
```

Install firmata :

```
npm install firmata --save
```

Because of a dependent package inside `firmata`, it will generate the compilation spec file according on your computer's spec during installation on your computer. However, the MIPS compilation spec file has already been installed on LinkIt Smart 7688 Duo, so we must erase this package manually:

```
rm -rf ./node_modules/firmata/node_modules/serialport/
```

Package `firmata` folder:

```
tar -cvf ./firmata.tar ./node_modules/firmata
```

Transfer the tar file into your `LinkIt smart 7688 Duo` board.

```
scp ./firmata.tar root@mylinkit.local:/root/app/node_modules/
```

(If something like `can not find node_modules folder` appears, please return to `/app` folder on your board and make a folder named `node_modules` : `mkdir node_modules`)

- Go back to your board's terminal.
- Back to `/testfirmata` :
- Enter `node_modules` folder: `cd node_modules`
- Unpack the tar file: `tar -xvf ./firmata`
- Back to `/testfirmata` : `cd ..`
- generate a `app.js` file:

```
vim app.js
```

- Paste the following content:

```

console.log('WWW blink start ...');

var ledPin = 13;
var firmata = require('firmata');

var board = new firmata.Board("/dev/ttyS0", function(err) {
  if (err) {
    console.log(err);
    board.reset();
    return;
  }

  console.log('connected...');
  console.log('board.firmware: ', board.firmware);

  board.pinMode(ledPin, board.MODES.OUTPUT);

  var url = require('url');
  var http = require('http');

  http.createServer(function(request, response) {
    var params = url.parse(request.url, true).query;
    try {
      if (params.value.toLowerCase() == 'high') {
        board.digitalWrite(ledPin, board.HIGH);
      } else if (params.value.toLowerCase() == 'low'){
        board.digitalWrite(ledPin, board.LOW);
      }
    } catch(e) {

    }
    response.writeHead(200);
    response.write("The value written was: " + params.value);
    response.end();
  }).bind(this)).listen(8080);

  console.log('Listening on port 8080 ...');
});

```

- Execute `app.js`.

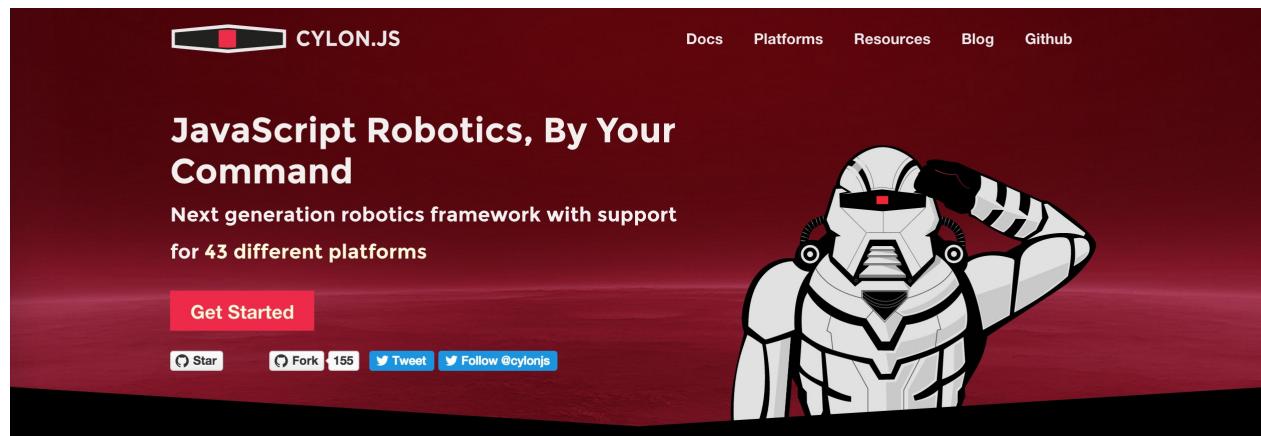
```
node app
```

- Press `Esc`, then enter `wq!` to save and leave.
- Open your browser:

- `http://mylinkit.local?value=high` (led-on)

- `http://mylinkit.local?value=low` (led-off)

Controlling LED with Cylon.js



Aim

(Notice: Before studying this section, be sure to finish firmata section)

In this section, we are going to use the famous open source project `cylon.js` on Nodejs to control LED.

Prerequisites

Control LED needs

(as the firmata section)

- LED x 1
- Resistor x 1
- A few DuPont wires

Circuit diagram

(as the firmata section)

Steps

MCU side

- Open your Arduino IDE.
- Copy the contents in this URL:

<https://gist.github.com/edgarsilva/e73c15a019396d6aaef2>

- Burn into Arduino.

MPU side

- ssh into LinkIt Smart 7688.
- Generate a folder named 'app':

```
> mkdir app && cd app
```

- Generate `app.js`:

```
> vim app.js
```

- Install `cylon` , `cylon-firmata` , `cylon-gpio` , `cylon-i2c` package:
 - Due to the same reason as firmda, node-gyp actions will cause too long running time on LinkIt Smart 7688 Duo. So we recommend using the same way as the firmata to install these four packages.
 - For detailed procedures, please refer to firmata section.
- After confirming above four kits is installed at `/node_modules` folder, add `app.js` in the root directory (`/app`) :

```
vim app.js
```

- After pressing the `i` key, enter:

```
var Cylon = require('cylon');

Cylon.robot({
  connections: {
    arduino: { adaptor: 'firmata', port: '/dev/ttyS0' }
  },
  devices: {
    led: { driver: 'led', pin: 13 }
  },
  work: function(my) {
    every((1).second(), my.led.toggle);
  }
}).start();
```

- Press `Esc`, then enter `wq!` to save and leave.
- Enter `node app.js` to see see lights on the board flashing.
- Done!

How to make your Node.js app start automatically on LinkIt Smart 7688 booting?

Aim

In this section, we are going to use some basic linux commands to make developer's Node.js app start automatically on LinkIt Smart 7688.

Prerequisites

- Your awesome Node.js App

Process

- Make sure your board is connected properly.
- ssh into it.
- Edit `/etc/rc.local` :

```
vim /etc/rc.local
```

- Refer to the content below and make `/etc/rc.local` into your app's path:

```
#!/bin/sh -e

node /root/app/app.js

exit 0
```

- After file saved, setup `/etc/rc.local` permissions:

```
chmod 777 /etc/rc.local
```

- Reboot and you will see the results! Bravo!

Easy Integration with Cloud Service (MCS)

This chapter is about the integration of LinkIt smart 7688 and MCS.

Let MCS control LinkIt smart 7688 by using MCSjs

Prerequisite

Please go through [this tutorial](#). It will guide you to create a test device in MCS. The device contains one on/off control type and one data channel with data channel ID named LED_control.

Note: After the test device is created in MCS, we need the deivceld and deviceKey on the right upper corner of device detail page. The deivceld and deviceKey will be used in the following steps.

Steps

- Check the connection of linkit smart 7688.
- ssh into it.
- Create a folder and go into that folder, and do npm initialization:

```
mkdir app && cd app && npm init
```

- Install MCSjs modules:

```
npm install mcsjs
```

- Edit app.js:

```
vim app.js
```

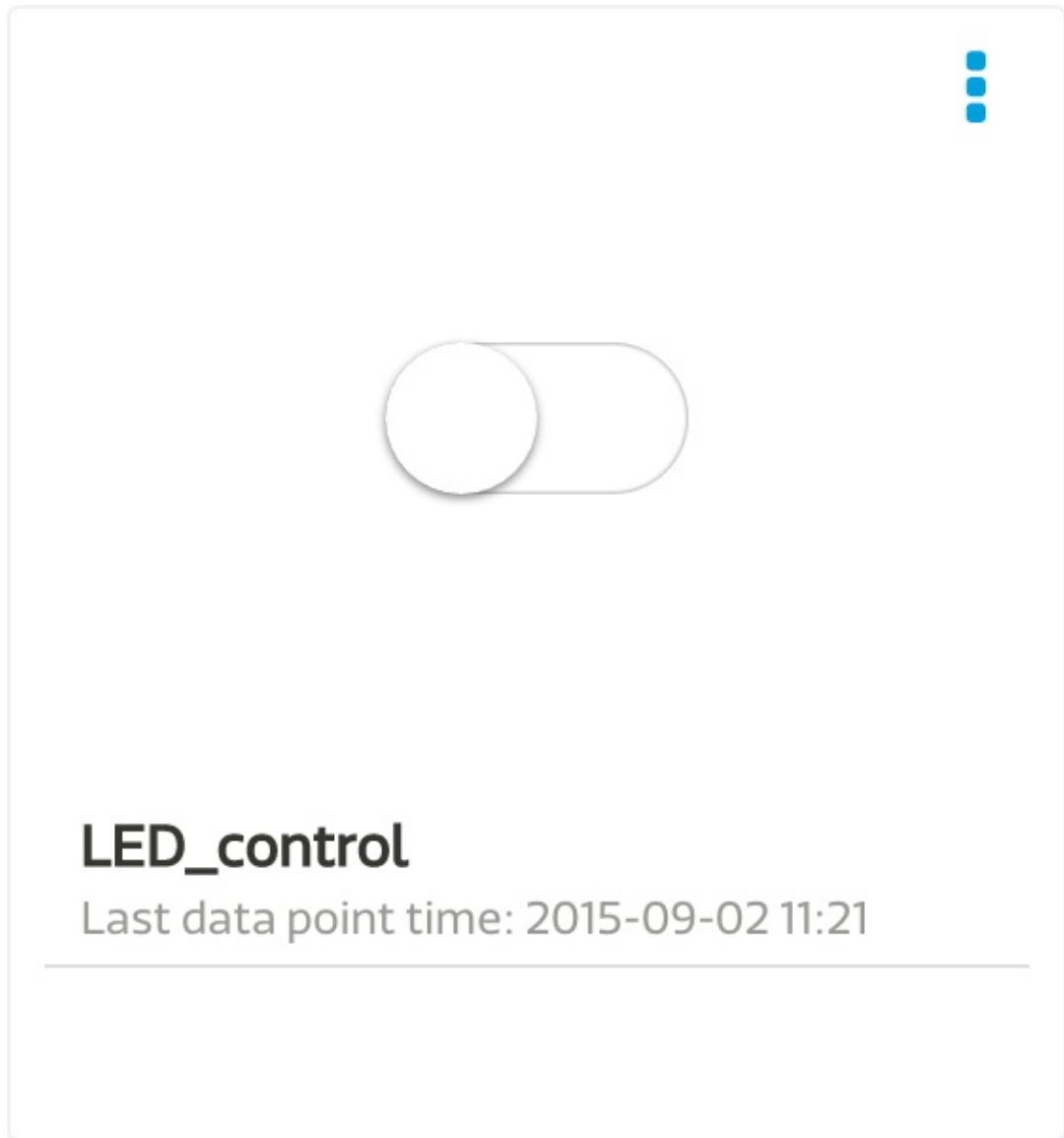
- input the content:

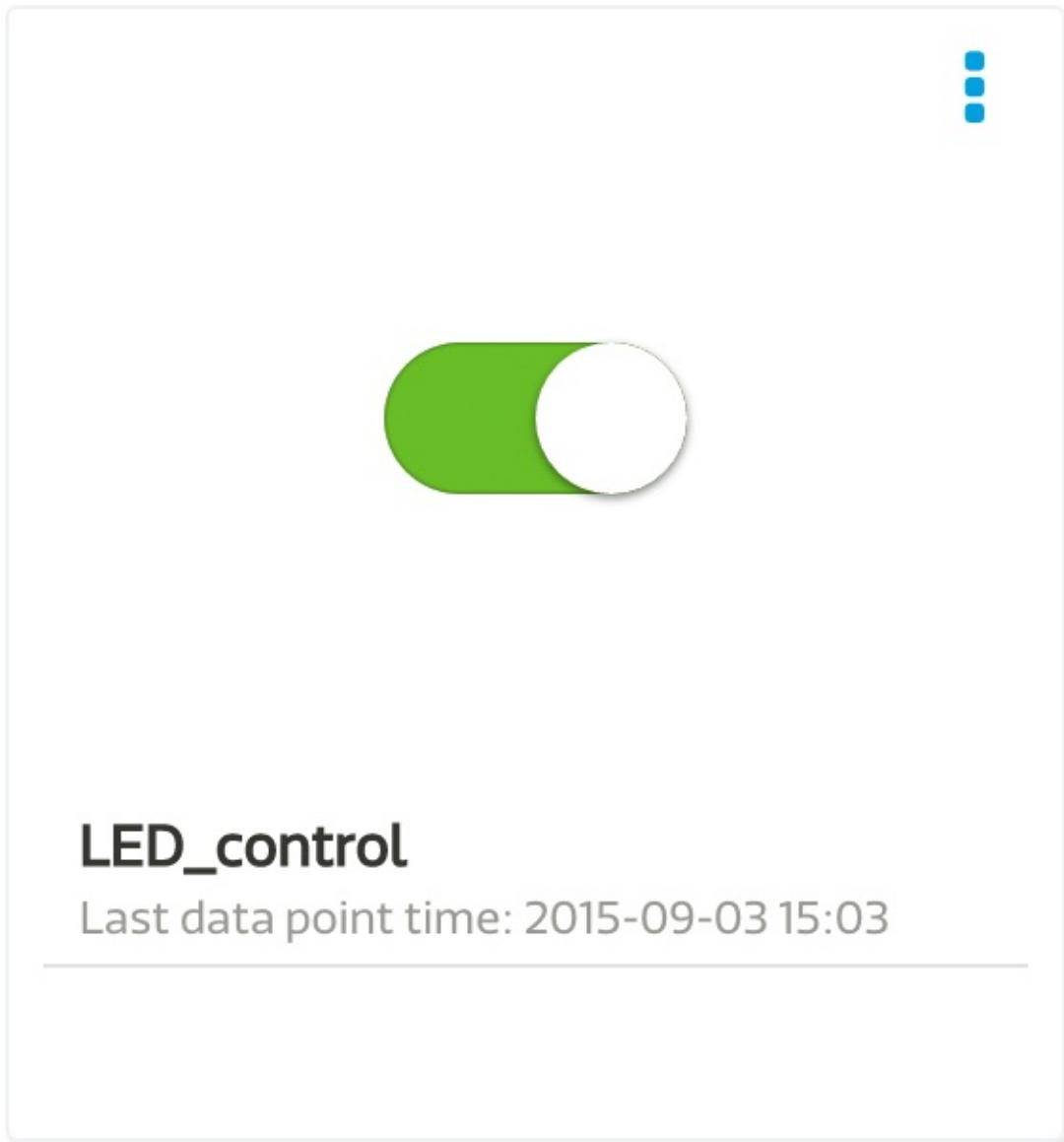
```
var mcs = require('mcsjs');

var myApp = mcs.register({
  deviceId: 'Input your deviceId',
  deviceKey: 'Input your deviceKey',
});
// Input the deviceId and deviceKey mentioned above.

myApp.on('LED_control', function(data, time) {
  if(Number(data) === 1){
    console.log('blink');
  } else {
    console.log('off');
  }
});
```

- Save and run node app
- Go back to MCS screen, and press the switch button on the data channel below.





- Switch to command line, and you will see:

```
blink!
```

- The LinkIt smart 7688 and MCS are successfully connected if the above step works fine.

We will control real LED light though the connection between MCS <-> LinkIt smart 7688 <-> LED (by firmata package).

Use MCS + firmata to control LED

Prerequisite

Please finish [previous chapters]/content/en/cloud/MCSjs.md).

Circuit Diagram

Please connect the hot wire of LED to D13.

Steps

MCU Side

- Open Arduino IDE
- copy the code from this site <https://gist.github.com/edgarsilva/e73c15a019396d6aaef2>
- Burn the code into Arduino

MPU Side

- Confirm the connection to LinkIt smart 7688.
- ssh into it.
- Replace the previous content of app.js with the following:

```
var ledPin = 13;
var firmata = require('firmata');
var mcs = require('mcsjs');
var board = new firmata.Board("/dev/ttyS0", function(err) {
  if (err) {
    console.log(err);
    board.reset();
    return;
  }
  console.log('connected...');
  console.log('board.firmware: ', board.firmware);
  board.pinMode(ledPin, board.MODES.OUTPUT);
  var myApp = mcs.register({
    deviceId: 'Input your deviceId',
    deviceKey: 'Input your deviceKey',
  });
  myApp.on('LED_control', function(data, time) {
    console.log('blink');
    console.log(data);
    if(Number(data) === 1){
      board.digitalWrite(ledPin, board.HIGH);
    } else {
      board.digitalWrite(ledPin, board.LOW);
    }
  });
});
});
```

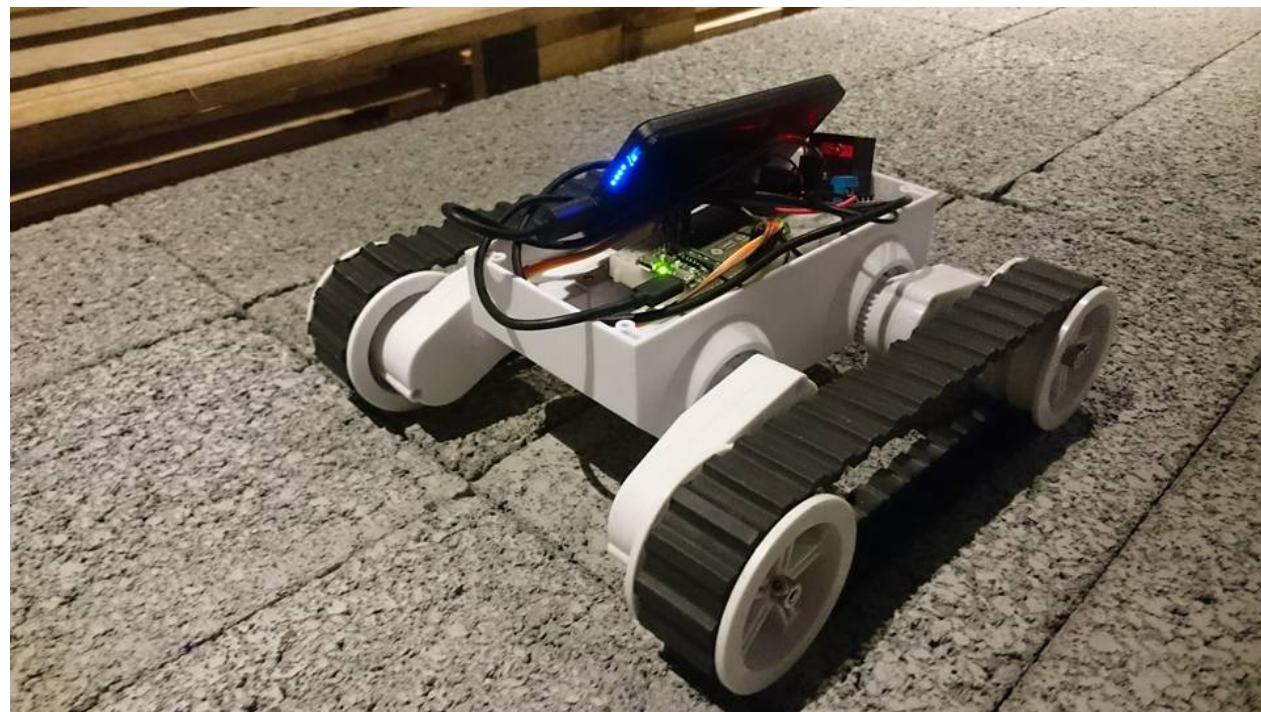
- Save file, and run node app.
- Press the switch in MCS, and the LED will change with it.
- We have done here!

上傳資料到 MCS (以指律心跳傳感器為例)

FOTA

Make a remote tank with LinkIt smart 7688 Duo and MCS gamepad channel

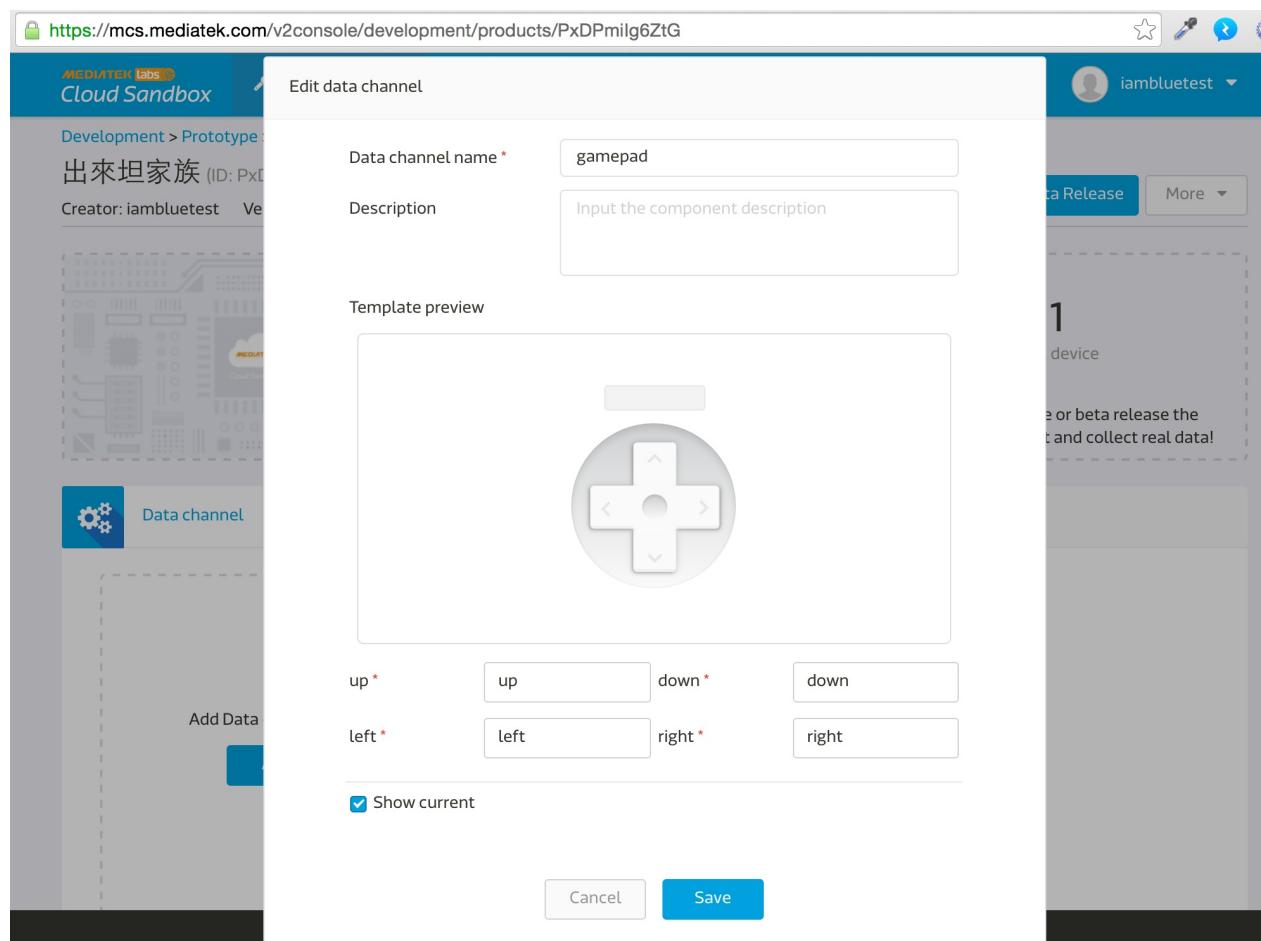
The screenshot shows the MediaTek Cloud Sandbox interface. At the top, there are navigation links: Development, Management, Resources, Help, and a user profile for 'iambluetest'. Below the header, the page title is 'Development > Prototype > 出來坦家族' (ID: PxDPmilg6ZtG). It displays the device's status as 'Under Development', industry as 'Government', application as 'Security', and description as 'for 7688'. A photo of a white tracked robot is shown on the left. On the right, there is a section for 'Test device' with a count of '1'. A button at the bottom right says 'Create test device or beta release the prototype to try out and collect real data!'. Below the main content, there are tabs for Data channel, Trigger & Action, User privileges, Firmware, and Test device.



Prerequisite

Please read [this tutorial](#) to creating a test device in MCS. It will contain only one gamepad (control type) and one data channel with data channel ID named 'gamepad.'

Please refer to the following image for details on creating data channel:



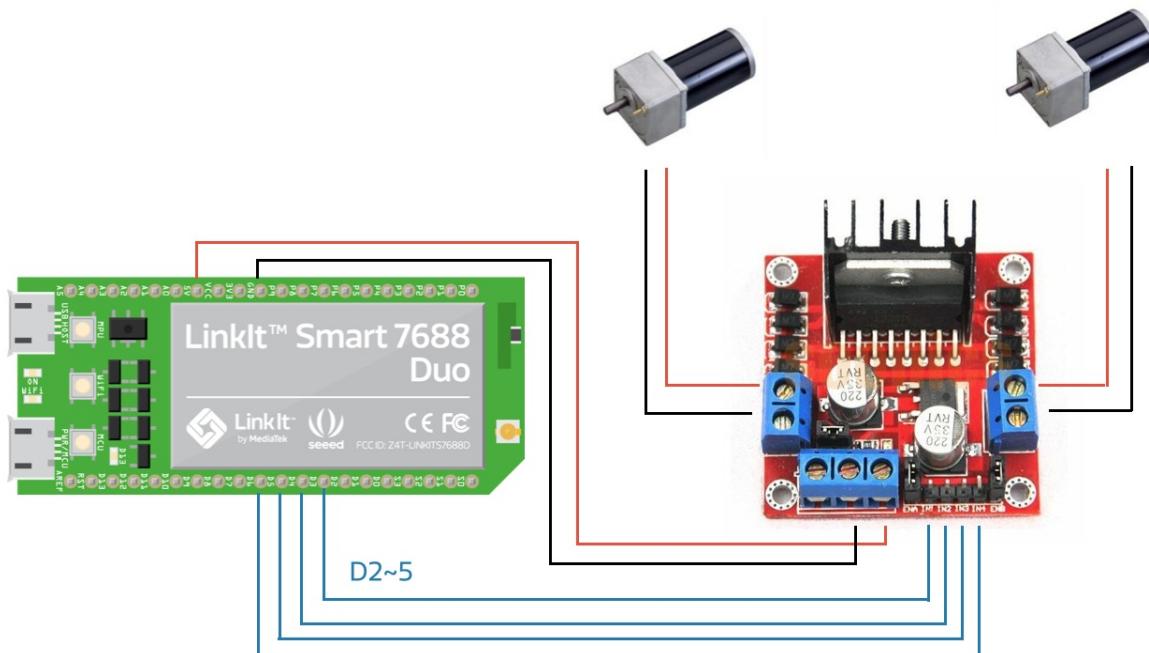
Note: After the test device is created in MCS, the deivceld and deviceKey can be found at the top right corner of device detail page. The deivceld and deviceKey are required in the following steps.

Preparation

- L289N motor-driven version
- A tank kit like this : <http://goods.ruten.com.tw/item/show?21438261771967>

Architecture

LinkIt Smart 7688 Duo



The four pins of L298N are connected to D2, D3, D4, D5 of Linkit smart 7688 Duo.

Steps on MCU (ATMega32U4) side

- Open Arduino IDE.
- In Tools -> board, choose Linkit smart 7688.
- Port -> select the one with Linkit smart 7688 port.
- Burn code from this site: <https://gist.github.com/iamblue/c71727f6cba953fcfd70>

Steps on MPU (MT7688) side

- Conform the connection with linkit smart 7688.
- ssh into it.
- Create a folder, go into it, and init npm:

```
mkdir app && cd app && npm init
```

- Install MCSjs modules:

```
npm install mcsjs
```

- Edit app.js:

```
vim app.js
```

- Input the following content: (Check this for details:

<https://gist.github.com/iamblue/27d79a58827093f4442c>)

```
var mcs = require('mcsjs');

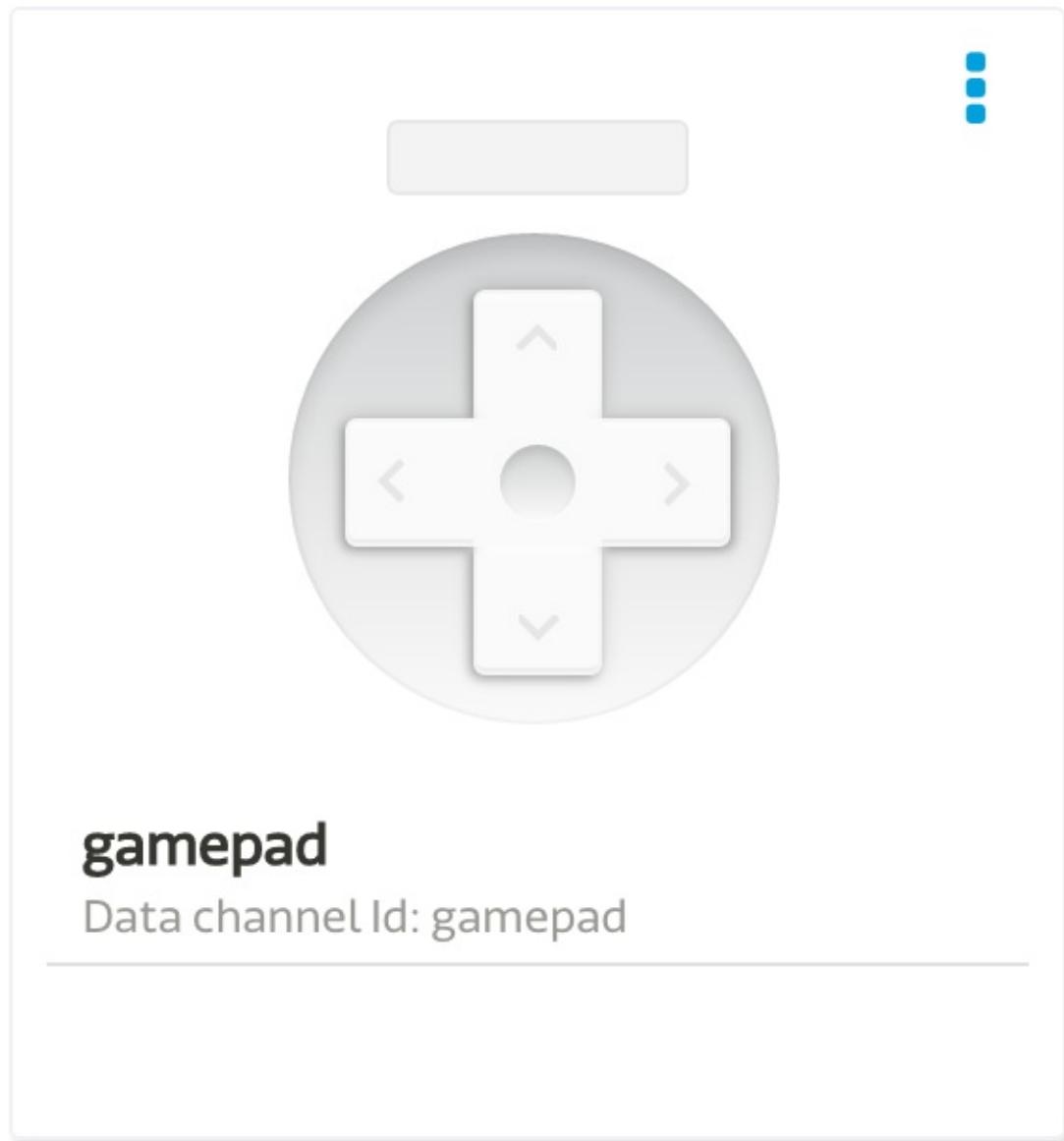
// regist your device to mcs.
var myApp = mcs.register({
    deviceId: 'Input your deviceId', // Input your deviceId.
    deviceKey: 'Input your deviceKey', // Input your deviceKey.
});

var SerialPort = require("serialport").SerialPort;
var serialPort = new SerialPort("/dev/ttyS0", {
    baudrate: 57600
});

// communicate with Arduino chip (32U4).
serialPort.on("open", function () {
    // listen the mcs command.
    myApp.on('gamepad', function(data, time) { // gamepad is your datachannel.
        serialPort.write(data); // send message to Arduino chip.
    });
});
```

- Save and run node app (Don't close command line window)

- Switch back to MCS screen, press up, down, left, or right arrows on this channel. You can use w, s, a, d key on keyboard instead.



- The tank will move now!

LinkIt smart 7688 模擬器

前言

LinkIt smart 7688 本身只有 32mb flash 以及 128 mb Memory, 對於想在版子上面做一些較耗資源的開發過程需要的東西，例如 cross compile 是非常痛苦的（因為這時候需要較多的 Memory 空間）因此在社群上有大神 (Fred) 針對 LinkIt smart 7688 的 Image 出一套 qemu 上的模擬器，我們可以在這個模擬器上面模擬更多記憶體和硬碟空間，去幫助我們克服這些事情。

相關連結

<https://github.com/cfsghost/makerboard>

如何使用

- 首先，先起一台 Ubuntu server (after 14.04 version)
- 在這台 Server 安裝好 Node.js 4 環境
- 安裝 Ubuntu 對應套件：

```
sudo apt-get install qemu-user-static squashfs-tools
```

- 安裝 makerboard: `npm install makerboard -g`
- 創建 emulator: `makerboard create my7688`
- 跑出模擬器: `makerboard run my7688`

注意：若您發現有這個字樣 `/bin/sh: Invalid ELF image for this architecture` 因為目前的 Ubuntu qemu core 的版本有小 bug，建議這時做一件事情：

- 下載 qemu-core-static for ubuntu 14.04

```
wget https://s3-ap-southeast-1.amazonaws.com/mtk.linkit/qemu-core-static
```

- 複製進剛剛 makerboard 創建好的 my7688/usr/bin:

```
cp qemu-core-static ./my7688/usr/bin/
```

最後再 run 一次模擬器：

```
makerboard run my7688
```

看到這樣的畫面即代表進去 LinkIt smart 7688 模擬器囉！

```
BusyBox v1.23.2 (2015-12-15 14:05:57 CST) built-in shell (ash)

/ # ls
IoT      dev      init.sh  mnt      proc      root      sys      usr      www
bin      etc      lib       overlay  rom       sbin      tmp      var
/ #
```

Educational programs

We observed that Maker in education can be divided into two groups: one capable of using of Linux and the other one not.

Elementary or high school students do not have basic Linux knowledge. Besides, Arduino communities already have a variety of open source software with drag and drop GUI interface (such as Scratch), so this group would majorly use the Arduino as the main development environment. In education, we recommend:

- Elementary and junior high schools: LinkIt smart 7688 Duo
- Senior high schools and universities: LinkIt smart 7688 or LinkIt smart 7688 Duo

This section provide to those who are to provide "elementary and junior high schools" educational resources and textbooks.

Open the Yunbridge on LinkIt smart 7688 Duo

Most Internet of Things projects needs wifi, and MT7688 Duo may provide wifi to the Arduino chip (32U4) which does not have wifi. However, default operation on Duo takes MPU (MT7688) and MCU (Arduino) as backup. If it need to turn to a MCU (Arduino) major, MPU (MT7688) minor system, it need to do something to open up the Yunbridge:

- Enter advanced WebUI, set the Duo to station mode to connect to the external network.
- ssh into LinkIt smart 7688 Duo:

```
ssh root@mylinkit.local
```

- Turn yunbridge on:

```
uci set yunbridge.config.disabled=0
```

- Save the above commit:

```
uci commit
```

- Reboot:

```
reboot
```

In Arduino side, use HttpClient to make sure bridge has been open.

- Open your Arduino IDE.
- File -> example -> Bridge -> HttpClient
- Burn this example into board.
- Open Arduino monitor window (magnifying glass icon). If this screen is seen, it succeeds.



After preparing the above materials, you can let students to concentrate on the development of the Internet of Things project on Arduino IDE.

Web of things

本章節待補中，敬請期待。

經典範例

本章節待補中，敬請期待。

使用 Touch sensor 觸發 LED

本章節待補中，敬請期待。