

IA:

Dev Tools and Resources
(Platform section)
Documentation and Training
(other item)
This tutorial
(other item)

Platform Documentation and Training page update:

Training

- ...
- (other tutorial)
Other tutorial blurb
- This tutorial
This tutorial blurb
- (other tutorial)
Other tutorial blurb
- ...

Side bar menu:

Introduction	2
Before you start	2
Setup your development environment.....	2
MediaTek Cloud Sandbox	2
Python modules.....	2
Building the LinkIt Smart 7688 MCS Hardware	3
What you need.....	3
Putting the components together	3
Setup LinkIt Smart 7688 in MCS	4
Step1: Registration	4
Step 2: Setup	4
Create a Python program to connect to MCS	13
Step1: Establish command pipe to MCS.....	13
Step 2: Send heart beat to TCP command.....	14
Step 3: Parse the command	14
Step 3: Set LED Value.....	14
Run your application.....	14
Conclusion	15

Tutorial content:

Title: Connecting LinkIt Smart 7688 to MediaTek Cloud Sandbox with Node.js

Introduction

MediaTek Cloud Sandbox is an IoT device prototyping service. In this guide you'll learn the steps to create a simple remote switch that allows you to turn on and off the on-board Wi-Fi LED from the web console of MCS using Node.js.

Before you start

If you haven't built a LinkIt Smart 7688 project before using MediaTek Cloud Sandbox, this section describes the steps you need to follow before commencing this project.

Setup your development environment

Full details on setting up the necessary LinkIt Smart 7688 development environment can be found in Get Started. Complete this before you continue, if you haven't already set up your development environment.

MediaTek Cloud Sandbox

This tutorial used the MediaTek Cloud Sandbox (MCS) to control the Wi-Fi LED of the LinkIt Smart 7688 from the web console. To use MCS [register](#) for a Labs account, if you haven't done so already, and [activate](#) your MCS account. You will then be able to define prototypes for your own devices and applications. By registering on Labs you also gain access to the hardware reference designs, the ability to submit and respond to items in the [forums](#), and more.

MCSjs modules

This tutorial uses MCSjs module. It is an easy-to-use HTTP request library. You can install it using npm. The steps are:

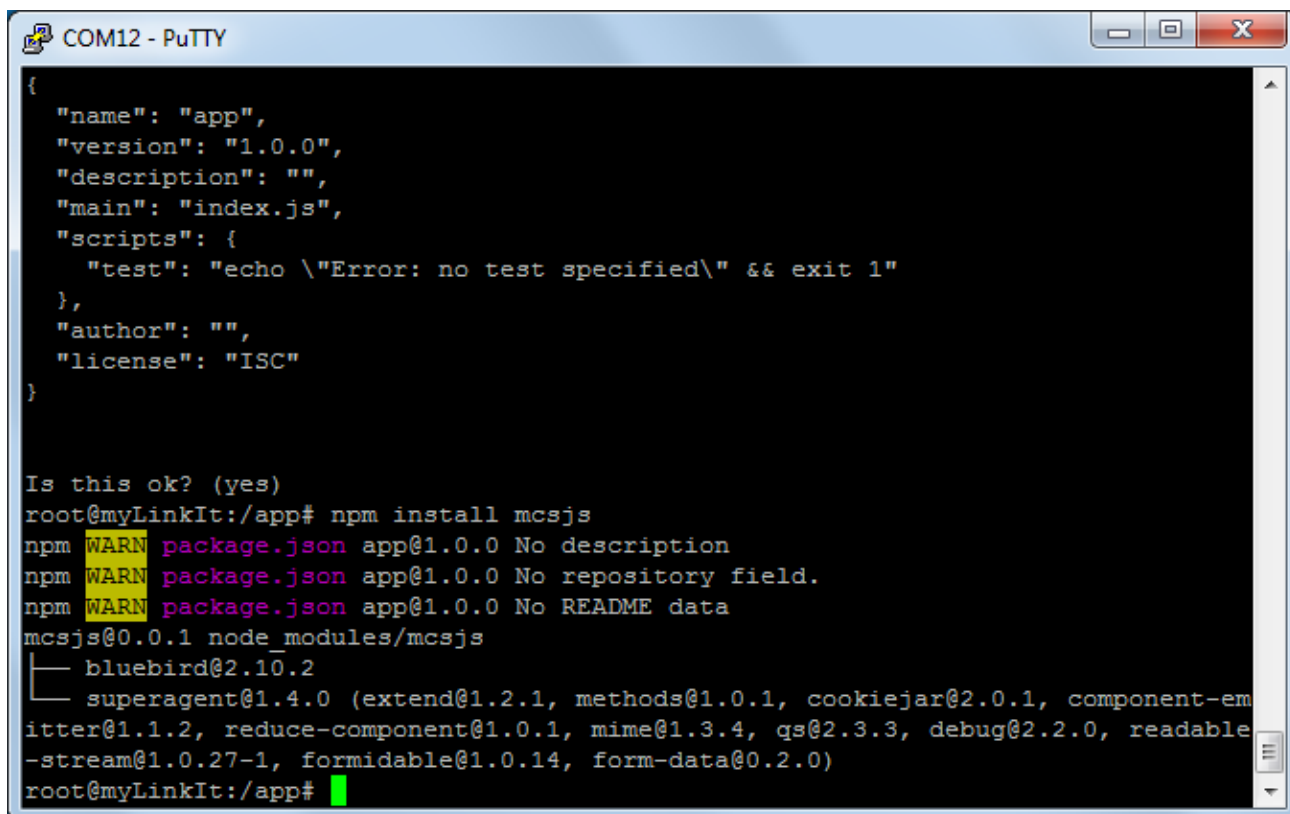
- 1) Make sure your LinkIt Smart 7688 is connected to the Internet (station mode).
- 2) Open a system console using SSH.
- 3) Create a folder app and initialize it.

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

- 4) Press enter to all the prompts when you see several questions, this is the application set up process.
- 5) Use npm to install module, for example:

```
npm install mcsjs
```

You should see a screen similar to below after mcsjs module installation.



```
COM12 - PuTTY

{
  "name": "app",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}

Is this ok? (yes)
root@myLinkIt:/app# npm install mcsjs
npm WARN package.json app@1.0.0 No description
npm WARN package.json app@1.0.0 No repository field.
npm WARN package.json app@1.0.0 No README data
mcsjs@0.0.1 node_modules/mcsjs
├─ bluebird@2.10.2
├─ superagent@1.4.0 (extend@1.2.1, methods@1.0.1, cookiejar@2.0.1, component-emitter@1.1.2, reduce-component@1.0.1, mime@1.3.4, qs@2.3.3, debug@2.2.0, readable-stream@1.0.27-1, formidable@1.0.14, form-data@0.2.0)
root@myLinkIt:/app#
```

Building the LinkIt Smart 7688 MCS Hardware

This section describes the hardware needed to build this tutorial provides details on how to put them together.

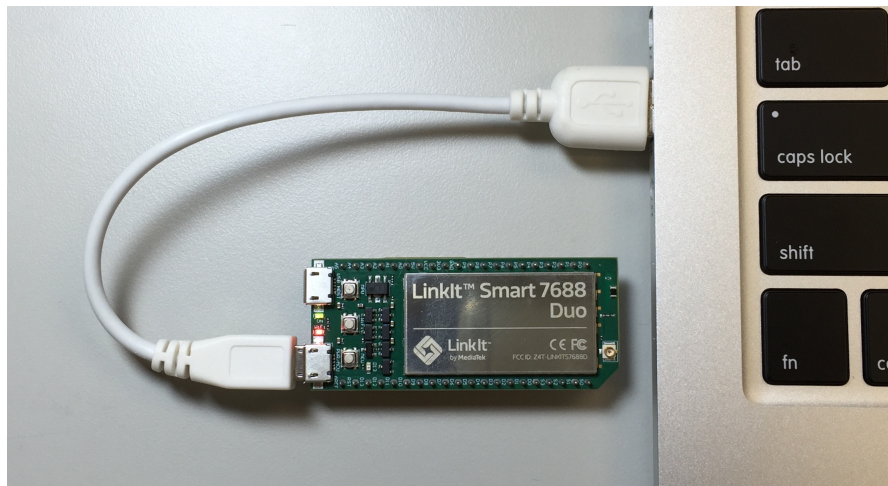
What you need

To build the (tutorial) hardware, in addition to a LinkIt (version) development board, you need the following components:

- LinkIt Smart 7688 development board
- Micro USB cable
- Host computer

Putting the components together

This section provides step-by-step instructions on putting the (tutorial) hardware together.



Hardware components

- 1) Connect the micro USB cable to the power connector of LinkIt Smart 7688 and a host computer.
- 2) Change LinkIt Smart 7688 to Station mode and connect to Internet. Please check LinkIt Smart 7688 Get Started Guide on using the web-based configuration tool to change to Station mode.

Setup LinkIt Smart 7688 in MCS

In this section you'll create a prototype device in MCS and connect it to LinkIt Smart 7688.

Step 1 : Registration

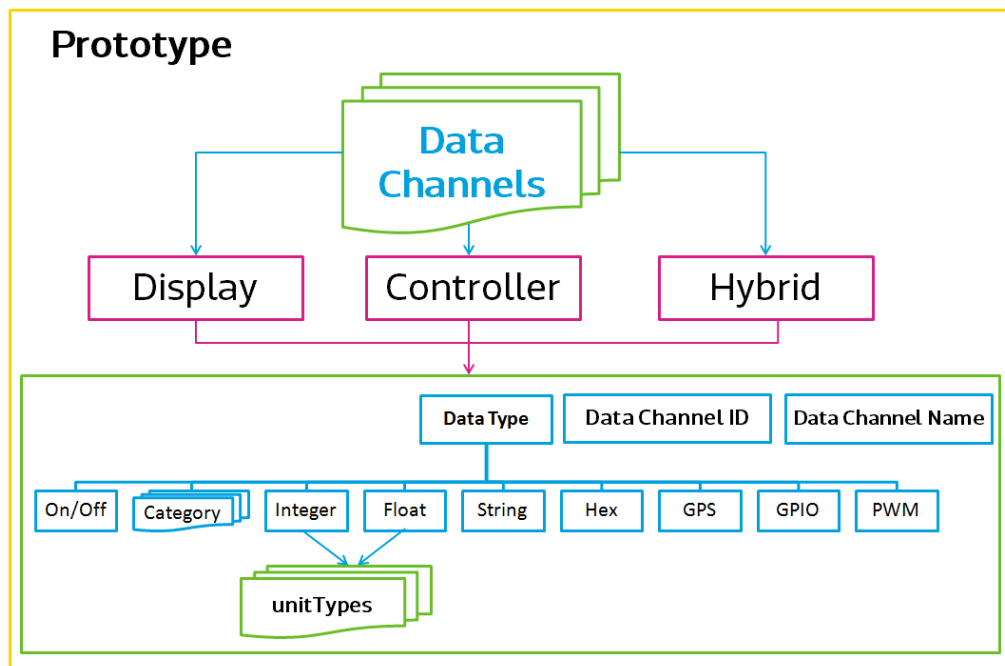
Click here to [register](#) on MCS. It's free.

Step 2: Setup

[Activate](#) a MCS account to prototype your own devices and applications. The connecting tutorial follows the general steps of application development on the MCS.

Step 3: Creating a new prototype for LinkIt Smart 7688

A prototype serves as a blueprint for the actual hardware setup. The prototype consists of one or more data channels of type display, controller and hybrid. The data channels are defined with a Data channel name, Data channel id and data type as required parameters. The variety of data types and overall structure of the prototype in general is shown in the figure below.

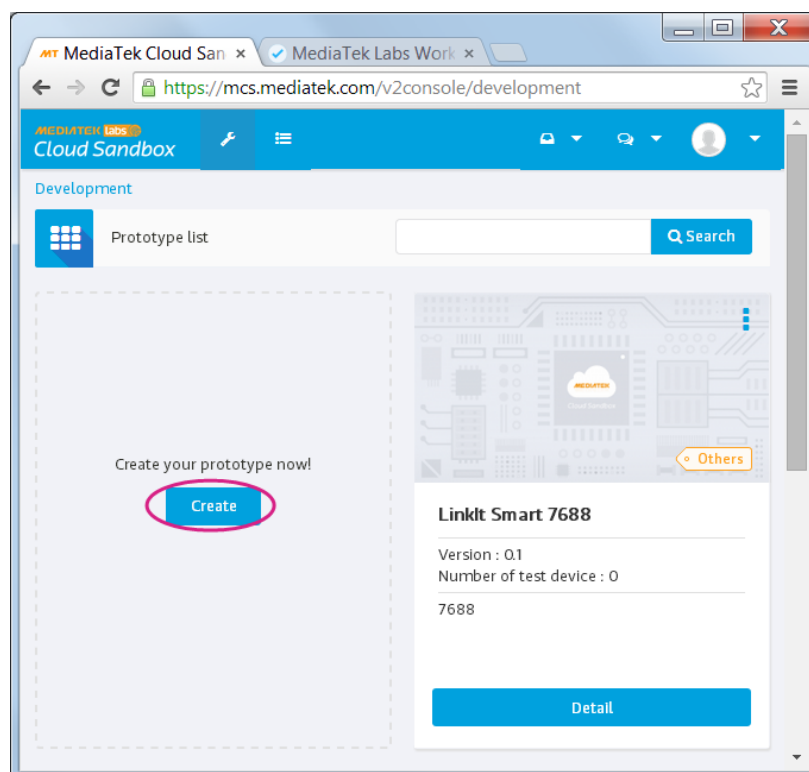


General prototype content in MCS

The LinkIt Smart 7688 prototype in this tutorial has one data channel to control Wi-Fi LED.

This section describes the details on how to create and configure a LinkIt Smart 7688 with corresponding data channel.

- 1) Click **Development** from the navigation toolbar, and then under Prototype list click **Create** to create a new prototype, as shown below.



Create a new prototype

2) In **Create prototype** define a basic profile of this prototype.

Create prototype

Prototype name *

LinkIt Smart 7688

Prototype version *

1.0

Hardware platform *

Others

▼

Description

LinkIt Smart 7688 platform

Industry *

Others


▼

Application *

Others

▼

Image upload



Drop an image here to upload!
Recommended 300 x 160 pixels.

Upload

Cancel

Save

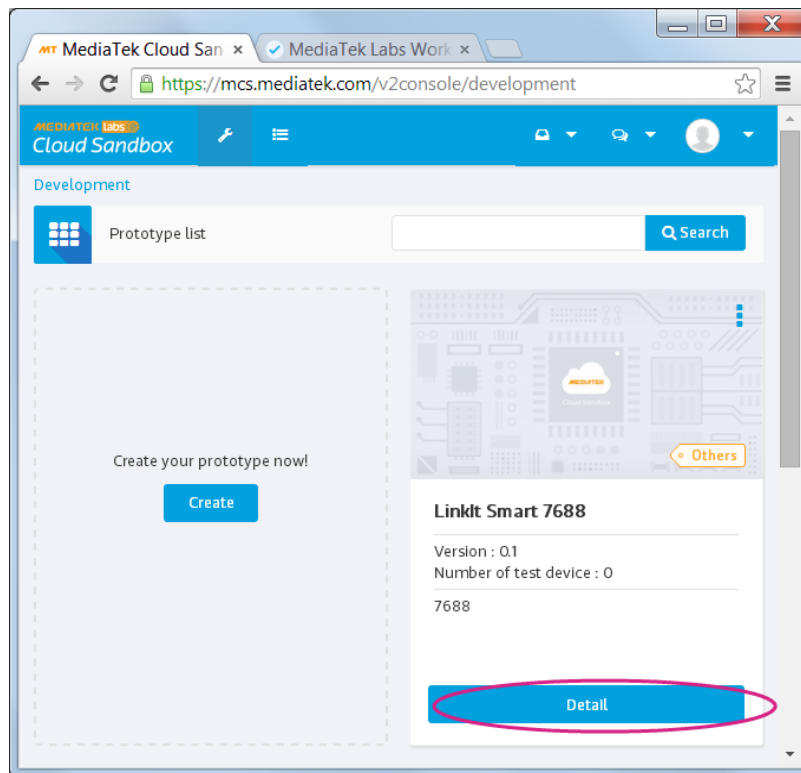
Create a prototype profile

Save the information and proceed to the next step.



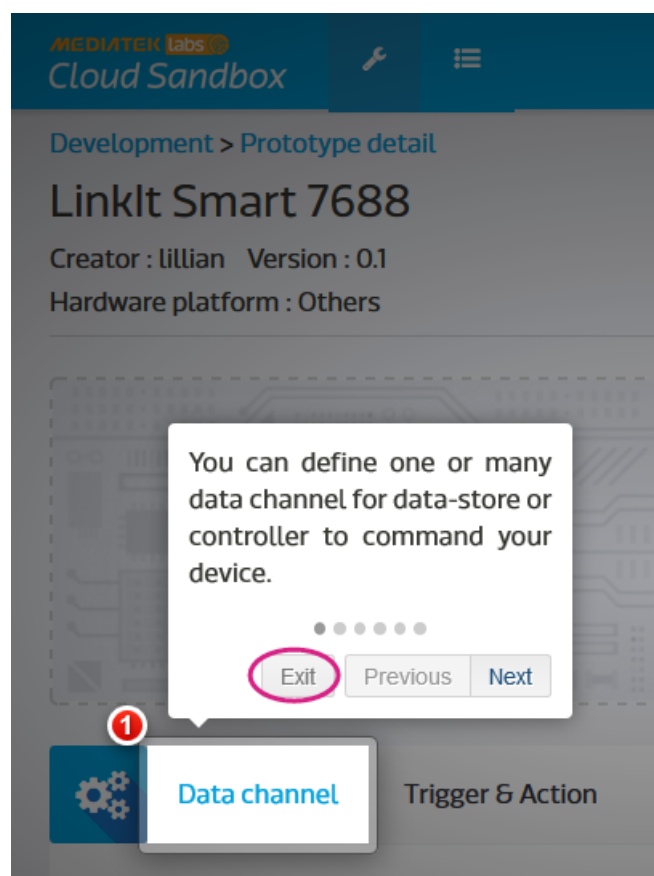
Note: Fields marked by a red asterisk (*****) are required fields.

3) Click **Detail** to view the prototype information, as shown below.



Accessing the LinkIt Smart 7688 prototype details

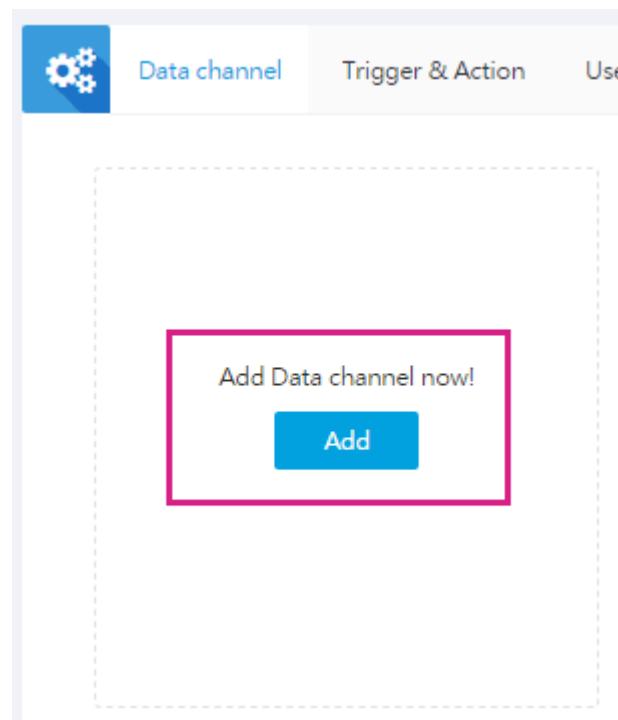
- 4) Click **Exit** to bypass initiation instructions, as shown below.



Bypassing the data channel initiation instructions

Next, add Data Channels for Data Control.


- 5) Click **Add** under **Data channel** toolbar, as shown below to provide data channels for the LinkIt Smart 7688 prototype.



Add a data channel to the LinkIt Smart 7688 prototype

- 6) Create a **Controller** data channel for Red color LED control on MCS by clicking **Add**.

Add data channel




Controller

The controller data channels allow you to control the status of your devices. eg, ON/OFF for a switch

Add


OFF ON



Display

The display data channels allow you to get the data from your devices.

Add



Oct 2

25 °C

3:21:25 PM

+

20 °C

-

hybrid

The Hybrid data channel is a two ways data channel contains bothe controller and display.

Coming soon.

Can't find your template?

Tell us what kind of template you want to add!

Send us a message!

Add a data channel type to the LinkIt Smart 7688 interface

- 7) Enter the information for the controller as shown and click **Save**. This defines the data channel and will be used in the Python program to communicate the data from the boards to MCS.

Add data channel


Data channel name *

Data channel Id *

Description

Data type *

Template preview: Select a template that suits your data channel.



Data channel details

- 8) Create a test device based on the LS7688 prototype as shown below. Each prototype on MCS is defined on a specific hardware platform. The hardware platform assigned in this tutorial is the LinkIt Smart 7688 development board. The LS7688 prototype is mapped to the actual device. Click **Create test device** as shown below.

Development > Prototype detail

LinkIt Smart 7688

Creator : lillian Version : 0.1
Hardware platform : Others

Industry : Others
Application : Others
Description : 7688

Data channel Trigger & Action User privileges Firmware **Test device**

Devices (0)

Name	DeviceId	DeviceKey	Registered date
------	----------	-----------	-----------------

+ Add test device

Creating test device for LinkIt Smart 7688 prototype

- 9) Provide Device name, as an example **device name** shown in the figure below, click **OK** to continue.

Create test device

Device name *

Device description

Test device configuration

- 10) Click **Go to detail**, as shown in the figure below to view the device information.

Success!



A test device is created successfully!
You can see its detail in "My devices".

No, thanks

Go to detail

1 Confirmation on creating a device for 7688 prototype

- 11) It's essential to store the **DeviceID** and **DeviceKey** values in a file because you'll need to replace them in the Python program in the next step to enable API calls and device connectivity.

The screenshot shows the 'My devices > Device detail' page in the Mediatek Cloud Sandbox. The device name is 'MyLS7688', created by 'lillian', version '0.1', on a hardware platform of 'Others'. A 'Back to prototype' button is visible. Below this, a circuit diagram of the device is shown on the left, and fields for 'Industry : Others', 'Application : Others', and 'Description :' are on the right. A warning message states: 'You will need the deviceId and deviceKey when calling our API to access this device'. Below this, the 'Deviceld:' is 'UKMIMISA77' and the 'DeviceKey:' is 'UbZSH53LKUHUXMZZ', both with 'Copy' buttons. At the bottom, there are tabs for 'Data channel', 'Trigger & Action', 'User privileges', and 'API hint'. The 'Data channel' tab is active, showing a toggle switch and the text 'ON/OFF control' with 'Data channel Id: LED_control'.

Device detail for the LinkIt Smart 7688 prototype

You have now created a new device in MCS matching the LinkIt Smart 7688 prototype with a data channel type Control (ON/OFF) as a switch to control the LED. Next section describes the software implementation.

Create a Node.js program to connect to MCS

This section provides the Node.js example code that listens for commands from MCS web console.

- 1) Create a file app.js using an editor, vi is used in this example:

```
vim app.js
```

2) Type and Copy/paste the following code in the editor.

In this example:

- Device ID: ABC123
- Device Key: XYZ123

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

var myApp = mcs.register({
  deviceId: 'ABC123',
  deviceKey: 'XYZ123',
});
// Replace the device ID and device Key obtained from your test device
// created in MCS.

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

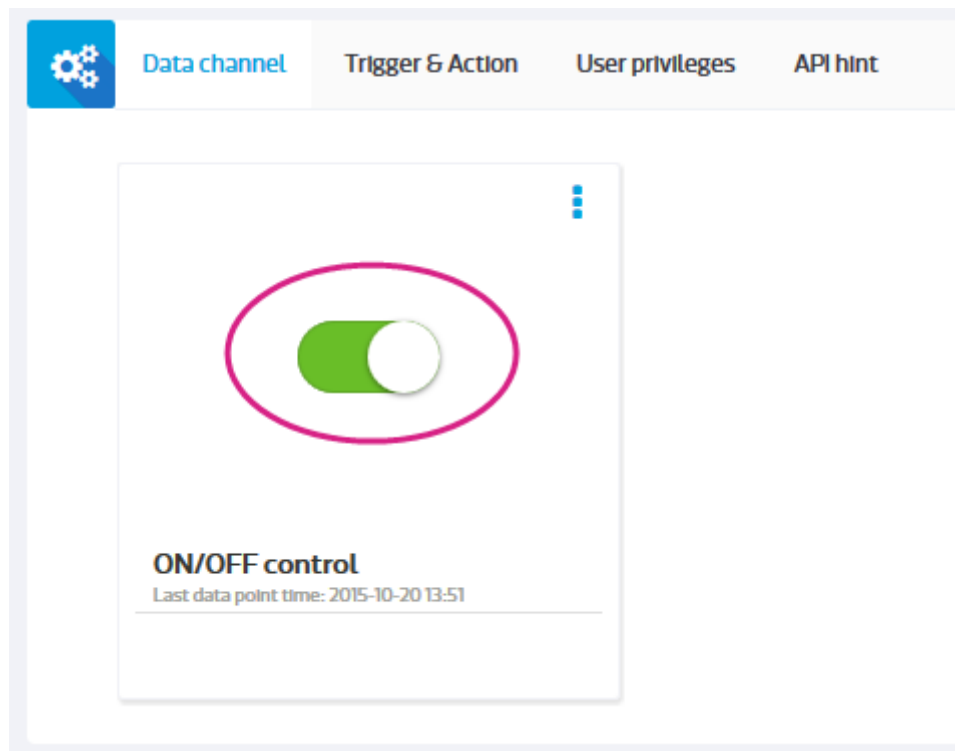
Next, run the Node.js example program.

Run your application

You are now ready to execute the Python program. In the system console, type the following command: # is command prompt and is not part of command.

```
# node app
```

Go to MediaTek Cloud Sandbox and use the controller panel to flip the button on and off and watch the Wi-Fi LED on LinkIt Smart 7688



Using MCS control switch to control LED

Conclusion

In this tutorial you've implemented a remote controlled LED switch application using LinkIt Smart 7688 development board, MediaTek Cloud Sandbox and Node.js programming language.

For more information on MediaTek LinkIt Smart 7688 development and prototyping board and cloud services refer to LinkIt Smart 7688 Developer's Guide and MediaTek Cloud Sandbox.