



CHIANG MAI

**MAKERCLUB**

[github.com/androidthings](https://github.com/androidthings)

developer.android.com/things/hardware/index.html

cmmakerclub/androidthings-ws2017 | Android Things | Android Things | Developer Kits | Android Things | cmmakerclub/androidthings-ws2... | Chiang Mai Maker Club | Thailan... | +

≡  Things HARDWARE SDK Search

← Hardware

[Overview](#)

[Hardware 101](#)

[Developer Kits](#)

[Partner Toolkit](#)

## Turnkey Hardware Solutions

We work with SoC partners to build Android Things certified development boards that contain SoMs (System-on-Chips) to get you developing fast.

### Integrated Parts

SoMs integrate the SoC, RAM, Flash Storage, WiFi, Bluetooth and other components onto a single board and come with all of the necessary FCC certifications. You can optimize your board design by flattening existing modules onto a PCB to save costs and space.

### Differentiated hardware

Our partners provide development boards with different SoMs and form factors to suit your needs, giving you choice and flexibility.

See [Developer Kits](#) for a list of supported development boards and starter kits.

## Low Barrier to Entry

Take advantage of Google's platform and support to scale from prototypes to complete products quickly.

### Get started quickly

Building devices is now within reach for everyone, even if you've never designed embedded systems. We provide a turnkey hardware

### Managed costs

Android Things reduces the large upfront development costs and risks inherent in getting your idea off the ground. When you're ready



CHIANG MAI

MAKERCLUB

developer.android.com/things/hardware

cmmakerclub/androidthi... Developer Kits | Android... Developer Kits | Androi... cmmakerclub/android... Chiang Mai Maker Clu... +

Things HARDWARE SDK Search

## Intel® Edison



The Intel® Edison compute module includes a dual-core x86 CPU running at 500 MHz, a 100Mhz MCU, 1GB of memory, 4GB of eMMC storage, dual-band Wi-Fi and Bluetooth 4.0.

[Learn More](#)

[Get Started with Edison](#)

## NXP Pico i.MX6UL

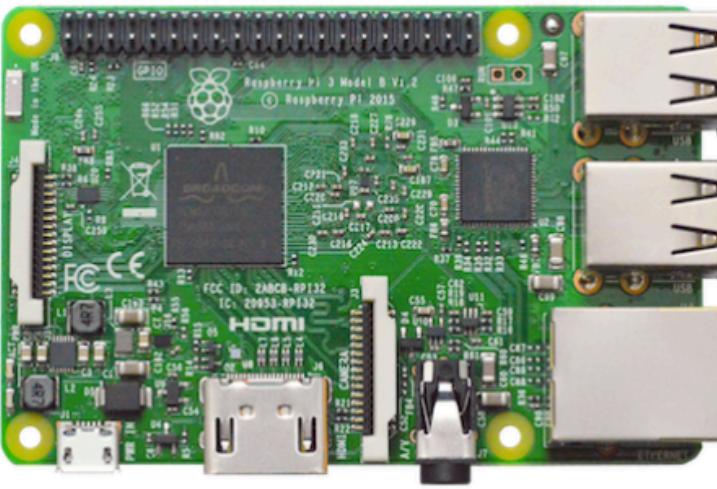


The Pico SoM provides an ARM® Cortex-A7 Core, 512MB of memory, 4GB of eMMC storage, Ethernet, wireless networking and Bluetooth 4.1.

[Learn More](#)

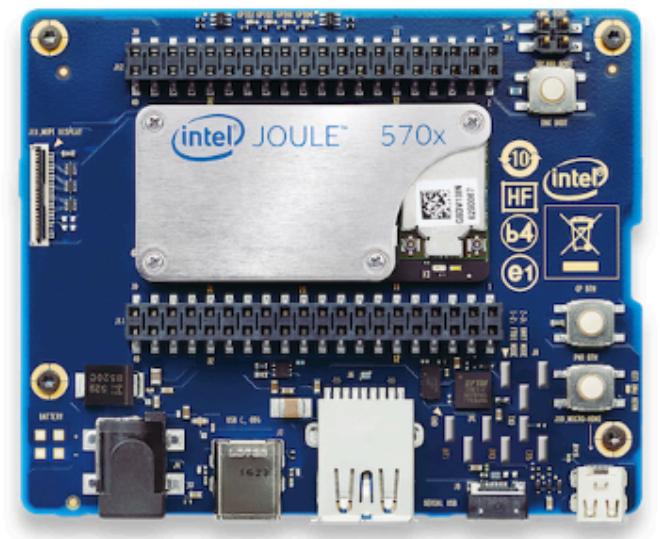
[Get Started with Pico](#)

## Raspberry Pi 3



CHIANG MAI  
**MAKERCLUB**

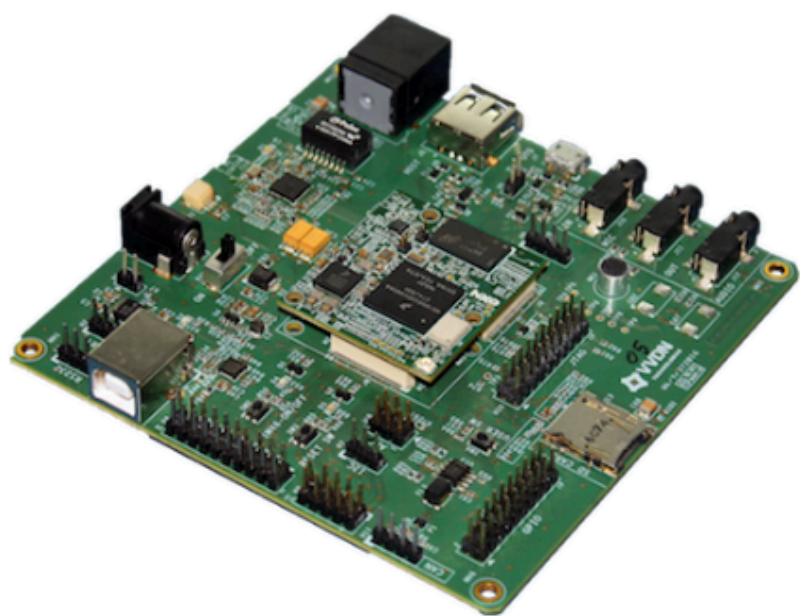
## Intel® Joule™ 570x



Intel® Joule™ provides a 64-bit quad-core Intel® Atom™ processor clocked at 1.5 GHz, 4GB of memory, between 8GB and 16GB on storage, 450 MHz GPU, dual band wireless networking and Bluetooth 4.2.

**Coming Soon!**

## NXP Argon i.MX6UL



The Argon SoM provides a ARM® Cortex®-A7 Core, 512MB of memory, 4GB of storage, wireless networking and Bluetooth 4.1.

**Coming Soon!**

developer.android.com/things/preview/download.html

cmmakerclub/androidthings-ws2017 System Image Downloads | Android Things Developer Kits | Android Things cmmakerclub/androidthings-ws2... Chiang Mai Maker Club | Thailand +

≡  Things HARDWARE SDK Search

← Hardware

Overview

Hardware 101

Developer Kits

System Image Downloads

Intel Edison

NXP Pico i.MX6UL

Raspberry Pi 3

Partner Toolkit

# System Image Downloads

This page contains binary image files that allow you to update or restore Android Things on a supported developer board.

Select a platform

| Platform     | Things package   | Size                        | SHA-1 checksum                            |
|--------------|--|-----------------------------|---|
| Intel        | <a href="#">androidthings_edison_devpreview_1.zip</a>      | 283 MB<br>(297119836 bytes) | 7827dafe8a4b7917a5fe536a5aaaf492091b54ed4 |
| NXP          | <a href="#">androidthings_imx6ul_pico_devpreview_1.zip</a> | 269 MB<br>(282087479 bytes) | d7ce6387be0a2ebb61c55a2e8a20dd205c1fa272  |
| Raspberry Pi | <a href="#">androidthings_rpi3_devpreview_1.zip</a>        | 268 MB<br>(282016162 bytes) | 911227a3c658bf74695118df9156cb20714f79d3  |

For more details, see the [preview release notes](#).



Need Help?



SELECT IMAGE

1



SELECT DRIVE

2



FLASH IMAGE

3

Select image

Select drive

Flash!

\*supported files: .img, .iso, .zip

ETCHER

IS AN OPEN SOURCE PROJECT BY



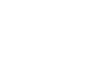
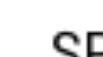
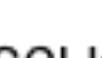
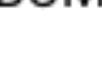
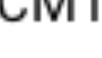
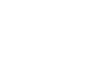
CHIANG MAI  
**MAKERCLUB**

```
$ adb connect 192.168.2.xx  
connected to 192.168.2.xx:5555
```

```
$ adb shell am startservice \  
-n com.google.wifisetup/.WifiSetupService \  
-a WifiSetupService.Connect \  
-e ssid <Network_SSID> \  
-e passphrase <Network_Passcode>
```

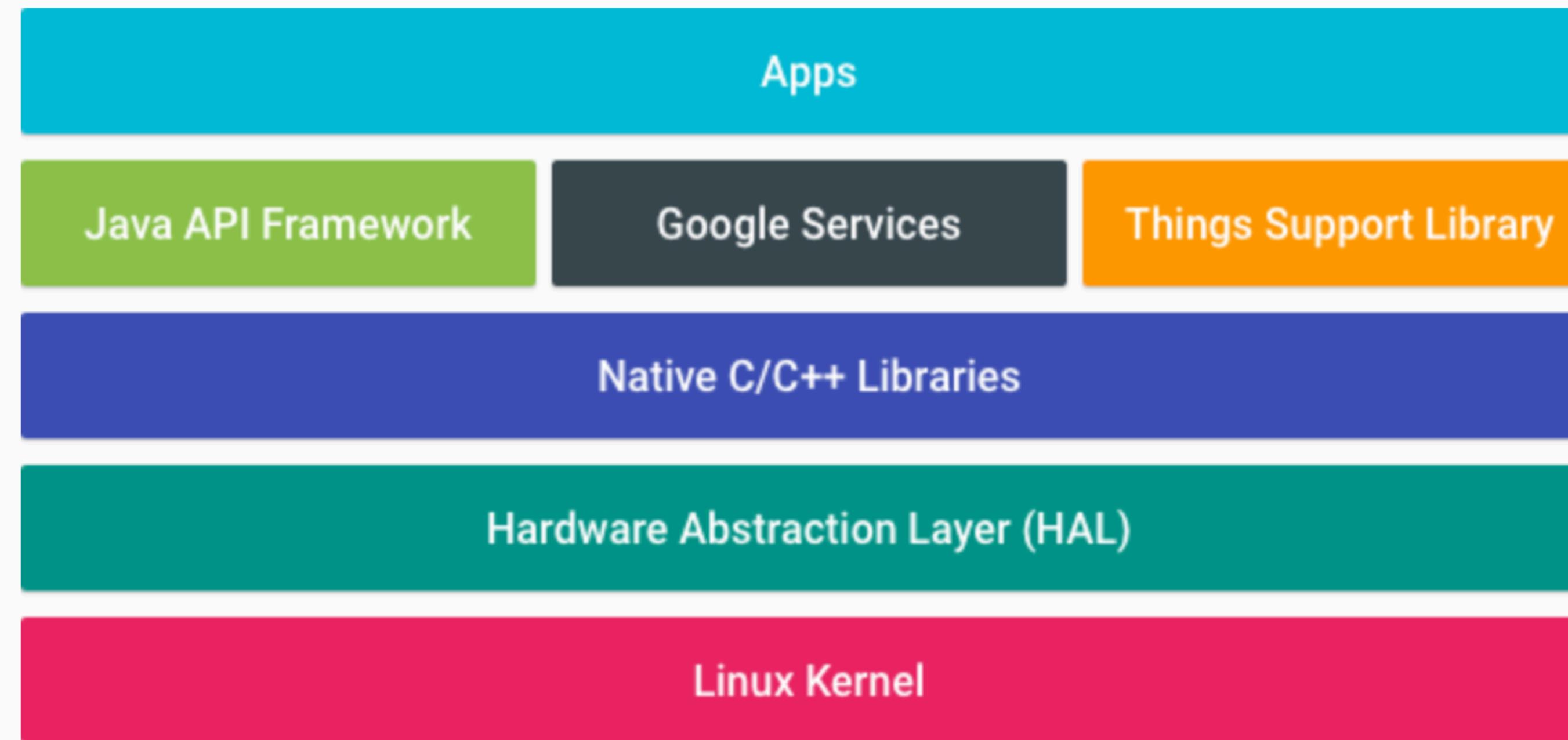
```
$ adb logcat -d | grep Wifi
```

|  |  |   |  |
|--|--|---|--|
|  = 5V   |  = Ground |  = PWM |  = SPI  |
|  = 3.3V |  = GPIO   |  = I2C |  = UART |

| J8           |    |  |              |
|--------------|----|--|--------------|
| 3.3V         | 1  |  2  | 5V           |
| I2C1 (SDA)   | 3  |  4  | 5V           |
| I2C1 (SCL)   | 5  |   6      | Ground       |
| BCM4         | 7  |   8      | UART0 (TXD)  |
| Ground       | 9  |   10     | UART0 (RXD)  |
| BCM17        | 11 |   12 | PWM0 / BCM18 |
| BCM27        | 13 |   14 | Ground       |
| BCM22        | 15 |   16 | BCM23        |
| 3.3V         | 17 |   18 | BCM24        |
| SPI0 (MOSI)  | 19 |   20 | Ground       |
| SPI0 (MISO)  | 21 |   22 | BCM25        |
| SPI0 (SCLK)  | 23 |   24 | SPI0 (SS0)   |
| Ground       | 25 |   26 | SPI0 (SS1)   |
|              | 27 |   28 |              |
| BCM5         | 29 |   30 | Ground       |
| BCM6         | 31 |   32 | BCM12        |
| PWM1 / BCM13 | 33 |   34 | Ground       |
| BCM19        | 35 |   36 | BCM16        |
| BCM26        | 37 |   38 | BCM20        |
| Ground       | 39 |   40 | BCM21        |

# Overview

Android Things makes developing connected embedded devices easy by providing the same Android development tools, best-in-class Android framework, and Google APIs that make developers successful on mobile.



Apps for embedded devices bring developers closer to hardware peripherals and drivers than phones and tablets. In addition, embedded devices typically present a single app experience to users. This document goes over the major additions, omissions, and differences between core Android development and Android Things.

Android Things extends the core Android framework with additional APIs provided by the Things Support Library. These APIs allow apps to integrate with new types of hardware not found on mobile devices.

The Android Things platform is also streamlined for single application use. System apps are not present, and your app is launched automatically on startup to immerse your users in the app experience.

## Things Support Library

---

### Peripheral I/O API

The Peripheral I/O APIs let your apps communicate with sensors and actuators using industry standard protocols and interfaces. The following interfaces are supported: GPIO, PWM, I<sub>2</sub>C, SPI, UART.

See the [Peripheral I/O API Guides](#) for more information on how to use the APIs.

### User Driver API

User drivers extend existing Android framework services and allow apps to inject hardware events into the framework that other apps can access using the standard Android APIs.

See the [User Driver API Guides](#) for more information on how to use the APIs.

## Displays are optional

Android Things supports graphical user interfaces using the same [UI toolkit](#) available to traditional Android applications. In graphical mode, the application window occupies the full real estate of the display. Android Things does not include the system status bar or navigation buttons, giving applications full control over the visual user experience.

However, [Android Things does not require](#) a display. On devices where a graphical display is not present, activities are still a primary component of your Android Things app. This is because the framework delivers all [input events](#) to the foreground activity, which has focus. Your app cannot receive key events or motion events through any other application component, such as a [service](#).

## Home activity support

Android Things expects one application to expose a "home activity" in its manifest as the main entry point for the system to automatically launch on boot. This activity must contain an intent filter that includes both [CATEGORY\\_DEFAULT](#) and [IOT\\_LAUNCHER](#).

## Support for Google services

Android Things supports a subset of the [Google APIs for Android](#). As a general rule, APIs that require user input or authentication credentials aren't available to apps.

The following table breaks down API support in Android Things:

| Supported APIs                                 | Unavailable APIs                        |
|--|---|
| <a href="#">Cast</a>                           |   |
| <a href="#">Drive</a>                          | <a href="#">AdMob</a>                   |
| <a href="#">Firebase Analytics</a>             | <a href="#">Android Pay</a>             |
| <a href="#">Firebase Cloud Messaging (FCM)</a> | <a href="#">Firebase App Indexing</a>   |
| <a href="#">Firebase Crash Reporting</a>       | <a href="#">Firebase Authentication</a> |
| <a href="#">Firebase Realtime Database</a>     | <a href="#">Firebase Dynamic Links</a>  |
| <a href="#">Firebase Remote Config</a>         | <a href="#">Firebase Invites</a>        |
| <a href="#">Firebase Storage</a>               | <a href="#">Firebase Notifications</a>  |
| <a href="#">Fit</a>                            | <a href="#">Maps</a>                    |
| <a href="#">Instance ID</a>                    | <a href="#">Play Games</a>              |
| <a href="#">Location</a>                       | <a href="#">Search</a>                  |
| <a href="#">Nearby</a>                         | <a href="#">Sign-In</a>                 |
| <a href="#">Places</a>                         |   |
| <a href="#">Mobile Vision</a>                  |   |

## Permissions

[Requesting Permissions at Runtime](#) is not supported because embedded devices aren't guaranteed to have a UI to accept the runtime dialog. [Declare permissions](#) that you need in your app's manifest file. All normal and dangerous permissions declared in your app's manifest are granted at install time.

## Notifications

Since there is no system-wide status bar and window shade in Android Things, [notifications](#) are not supported. Avoid calling the [NotificationManager](#) APIs in your apps.

The screenshot shows a web browser window displaying the 'Release Notes' page for the Android Things developer preview. The URL in the address bar is [developer.android.com/things/preview/releases.html](http://developer.android.com/things/preview/releases.html). The page has a blue header with the Android Things logo, 'HARDWARE' and 'SDK' tabs, and a search bar. A sidebar on the left is titled 'SDK' and contains sections for 'Overview', 'SDK Preview' (which is selected), 'Release Notes' (highlighted in blue), 'Peripheral I/O', 'User-space Drivers', and 'Samples'. The main content area features a large heading 'Release Notes' and a paragraph explaining the purpose of the document. It also includes instructions for filing bugs, checking for known issues, and running the latest system image. Below this, there's information about the Developer Preview 1, including its date, build number, and preview APIs. A section on general guidelines for the preview release follows, with two bullet points detailing stability issues and supported hardware.

# Release Notes

This document outlines issues and fixes related to each release of the Android Things developer preview.

Please file bugs using the [Android Things template](#) for issues discovered in the system, hardware support, and documentation. Before filing a bug, please check this page for known issues and ensure you are running the [latest version](#) of the preview system image.

To ask questions and discuss ideas with other developers working on Android Things, join the [IoT Developers Google+ community](#). You can submit new feature requests using [this template](#).

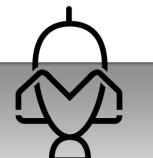
## Developer Preview 1

*Date: December 2016*  
*Build Number: NIF73/NIF74*

Preview APIs [Javadoc reference](#).

This preview release is for **developers and early adopters** to use for development and compatibility testing on supported hardware platforms. Please note the following general guidelines about the preview:

- This release may have various **stability issues** on supported hardware. Please file bugs to report issues that you discover.
- Developer Preview 1 is available on the Intel Edison, NXP Pico, and Raspberry Pi 3 development boards.

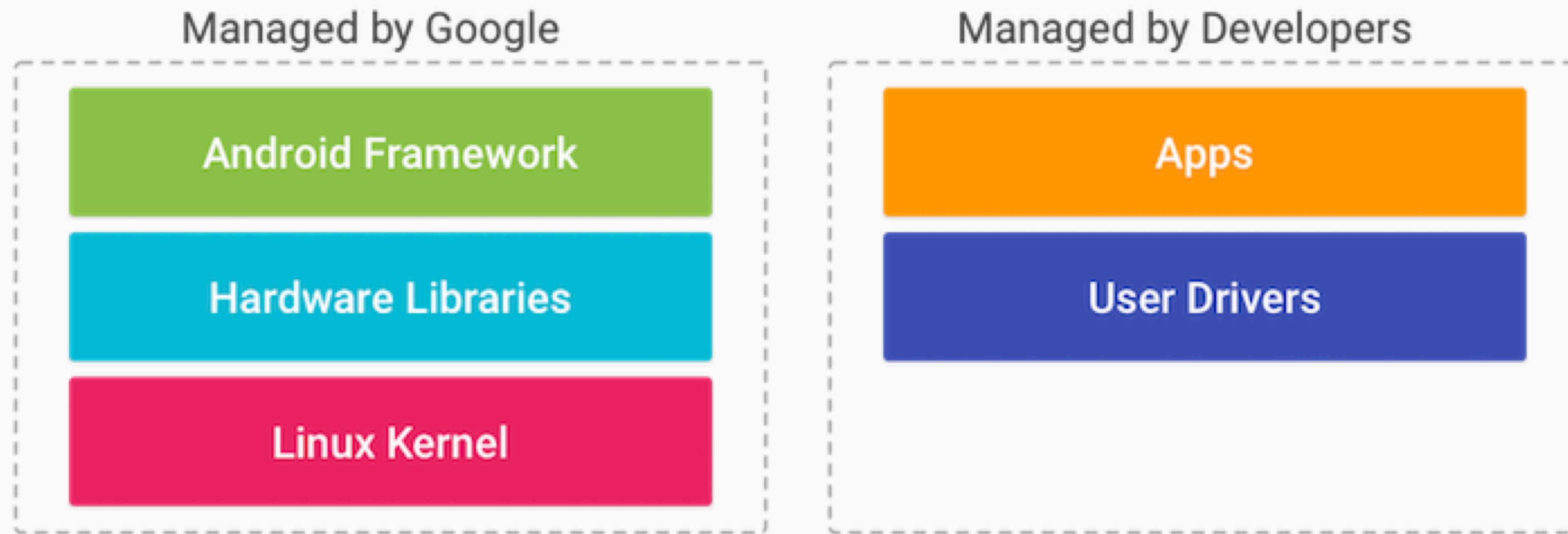


CHIANG MAI

MAKERCLUB

# User-Space Drivers

To allow app developers to register new device drivers with the framework, Android Things introduces the concept of a **user driver**. User drivers are components registered from within apps that extend existing Android framework services. They allow any application to inject hardware events into the framework that other apps can process using the standard Android APIs.



**Note:** You cannot customize the behavior of device drivers in the [Linux kernel](#) or [Hardware Abstraction Layer \(HAL\)](#) to add new functionality to a device.

## Benefits

---

In many apps, using [Peripheral I/O](#) to communicate directly with external hardware devices is sufficient. However, there are some benefits to integrating your hardware with the rest of the Android framework:

- **Portability:** Application code that purely targets the Android framework can run on a variety of different boards and configurations without additional abstractions for the device driver implementation.
- **Reuse:** You can pull existing Android code snippets and libraries into your application without the need to modify or fork them to handle your specific hardware implementation.
- **Integration:** Android often combines data from various services together to enhance the information reported to apps or create new virtual data sets. User drivers can contribute to this process.

## Device Driver Types

---

**GPS** - GPS provides high accuracy physical location information to apps. Integrating the location results from GPS devices with a user driver allows the framework to connect that data with other location sources, such as WiFi, and Google's [Fused Location Provider](#).

**HID** - Human Interface Devices (HID) provide user input to apps. Touch pads, keyboards, and game controllers are all examples of devices that provide this type of input. Input user drivers let devices interact with the enhanced input framework APIs, such as [Gesture Support](#) or [Drag and Drop](#).

**Sensors** - Sensors measure and report the conditions of the physical environment. The Android sensor framework implements ***sensor fusion*** to combine the raw data from multiple physical sensors into a single virtual sensor. This is particularly common with motion sensors, such as accelerometers and gyroscopes. Connecting your sensor to the framework with a user driver allows the data it produces to be included in sensor fusion.

[https://github.com/cmmakerclub/  
androidthings-ws2017](https://github.com/cmmakerclub/androidthings-ws2017)

blinky

mqtt

build.gradle (project)

```
allprojects {  
    repositories {  
        jcenter()  
        maven {  
            url "https://repo.eclipse.org/  
content/repositories/paho-releases/  
        }  
    }  
}
```

build.gradle (app)

```
compile 'com.android.support:appcompat-v7:25.1.0'  
compile 'org.eclipse.paho:org.eclipse.paho.client.mqttv3:1.1.0'  
compile 'org.eclipse.paho:org.eclipse.paho.android.service:1.1.0'
```

```
private void setupMqttClient() {
    mqttAndroidClient = new MqttAndroidClient(getApplicationContext(), serverUri, clientId);
    mqttAndroidClient.setCallback(new MqttCallbackExtended() {
        @Override
        public void connectComplete(boolean reconnect, String serverURI) {
            Log.d(TAG, "connectComplete: ");
            subscribe("test");
            publish("test", "hello world");
        }

        @Override
        public void connectionLost(Throwable cause) { Log.d(TAG, "connectionLost: "); }

        @Override
        public void messageArrived(String topic, MqttMessage message) throws Exception {
            Log.d(TAG, "messageArrived: " + message.toString());
        }

        @Override
        public void deliveryComplete(IMqttDeliveryToken token) {
            Log.d(TAG, "deliveryComplete: ");
        }
    });

    MqttConnectOptions mqttConnectOptions = new MqttConnectOptions();
    mqttConnectOptions.setAutomaticReconnect(true);
    mqttConnectOptions.setCleanSession(true);

    try {
        mqttAndroidClient.connect(mqttConnectOptions, null);
    } catch (MqttException ex) {
        ex.printStackTrace();
    }
}
```

setupMqttClient: **<http://bit.ly/2j9Pci1>**

# AndroidManifest.xml

```
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission
    android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
```

<!-- Mqtt Service -->  
<service  
    android:name="org.eclipse.paho.android.service.MqttService" />

# MainActivity

```
private MqttAndroidClient mqttAndroidClient;
```