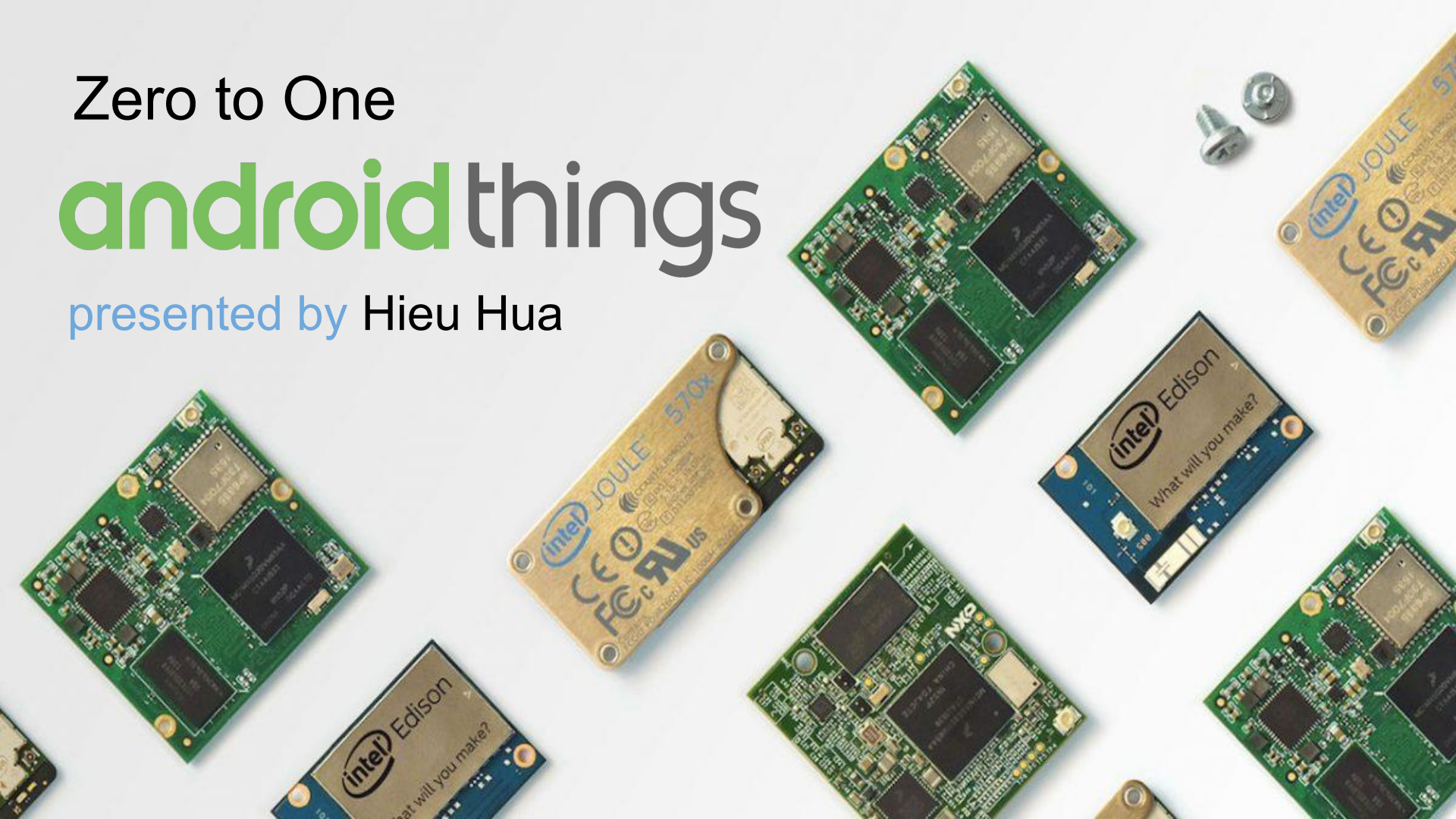
Zero to One

android things

presented by Hieu Hua



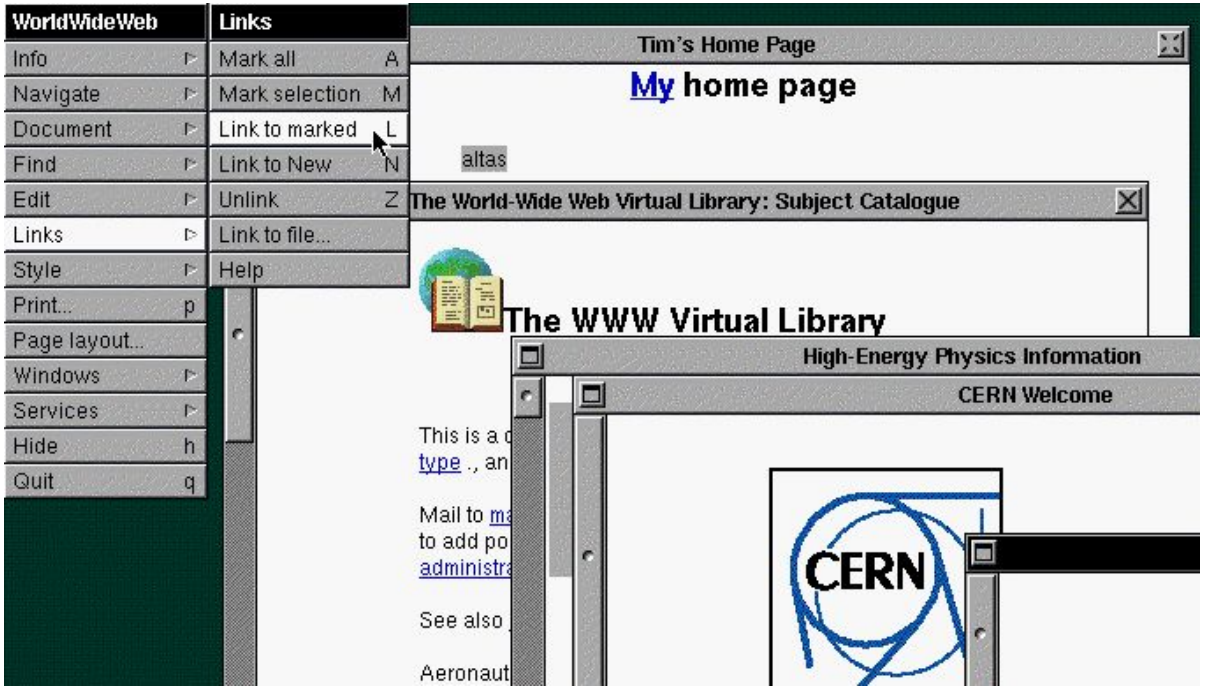
GDE Android

Zalo

https://github.com/rockerhieu

rockerhieu@gmail.com

The Internet is young



On June 8th 1991, the World Wide Web project was announced to the world


NEWS

90% of world's existing data has been created in the last 2 years

1 Billion - pieces of content shared on Facebook on a daily basis

2.5 Quintillions - the amount of data generated by people everyday

6 Billions - hours of video watched on YouTube every month

271 Millions - monthly active user on Twitter

8 Zetabytes - amount of data in the digital universe

www.quora.com (Dec 2016) | How big is a zetabyte?


HTV7
NOKIA
NOKIA


NEWS

Image from ServerCentrix

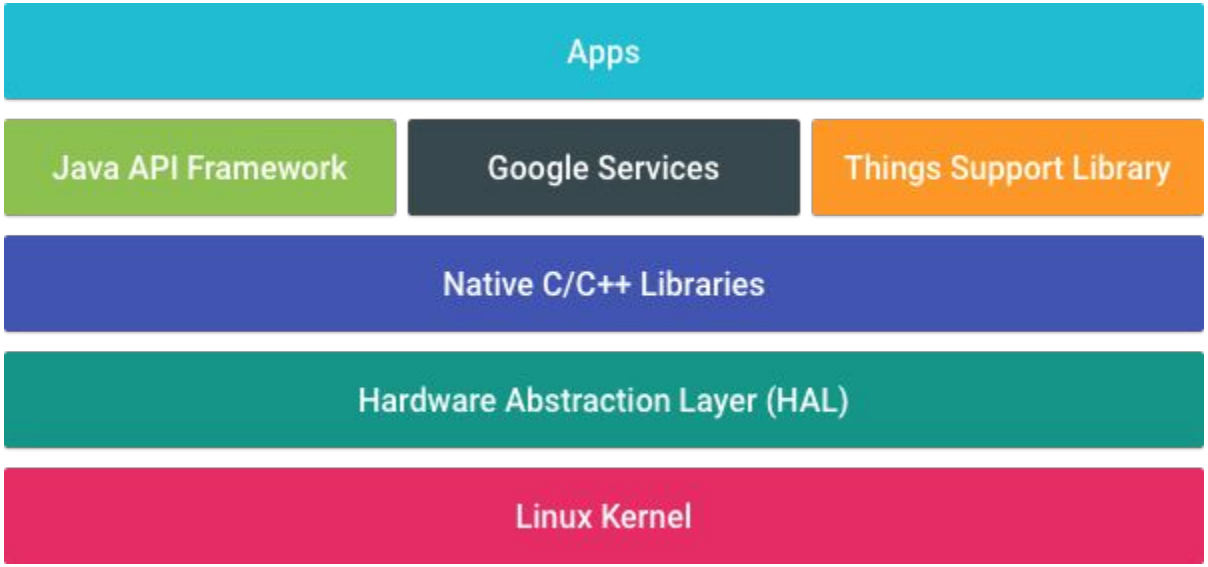
android things

- Build connected devices using familiar development tools, i.e Android SDK, Android Studio
- Google Play Services & Google Cloud
- Flashable image + SDK (developer preview)

Android Things

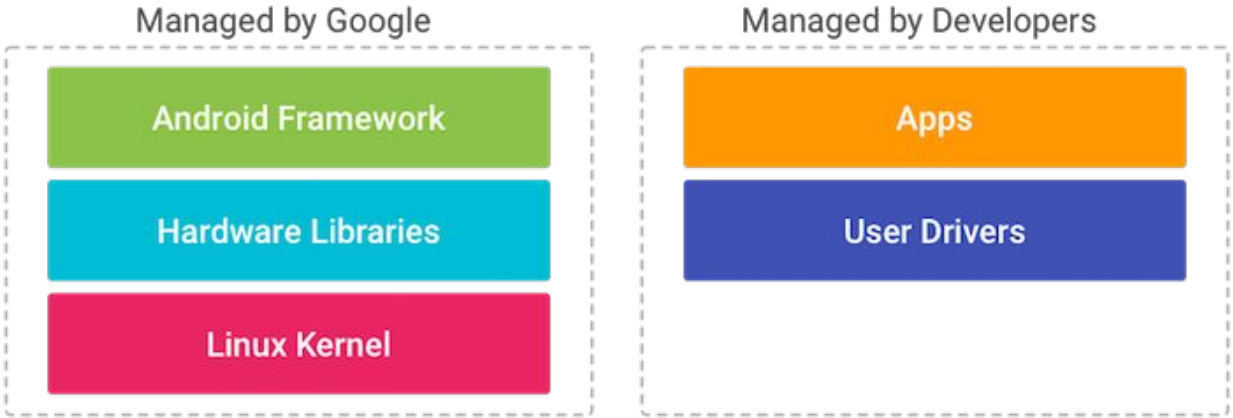
- Build connected devices using familiar development tools, i.e Android SDK, Android Studio
- Google Play Services & Google Cloud
- Flashable image + SDK (developer preview)

Architecture



Things Support Library

- Peripheral I/O API
  - GPIO, PWM, I2C, SPI, UART
- User Driver API
  - Allow apps to inject hardware events into the framework



Behaviour changes

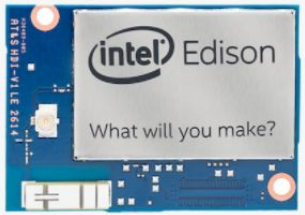
- Missing core applications and content providers
  - Avoiding using common intents and system content providers
- Displays are optional
- Support a subset of Google Play Services
- No runtime permissions
  - Dangerous permissions are granted on the next device reboot
- No notifications

Hardware

- Certified development boards
- SoMs (System-on-Module)
  - SoC, RAM, Flash Storage, WiFi, Bluetooth, USB, ...
- Board Support Package (for boot loader, boot manager) managed by Google

Hardware

Intel® Edison



Intel® Joule

NXP Pico i.MX7D

NXP Pico i.MX6UL

NXP Argon i.MX6UL

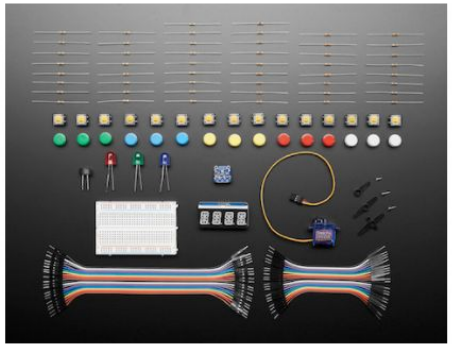
Raspberry Pi 3

NXP Argon i.MX6UL

Raspberry Pi 3

https://developer.android.com/things/hardware/developer-kits.html

Tinkering



Adafruit Project Kit

Pimoroni Rainbow HAT

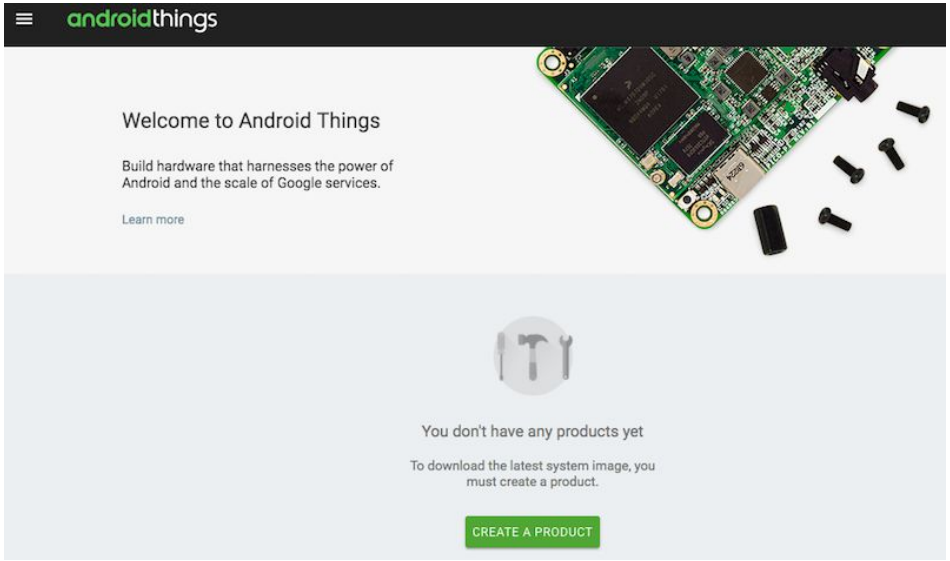
Sparkfun Project Kit

https://developer.android.com/things/hardware/developer-kits.html

Android Things Console

Getting IoT devices to enter the market faster and more securely

- Product creation and settings
- Factory images
- OTA updates
- Analytics

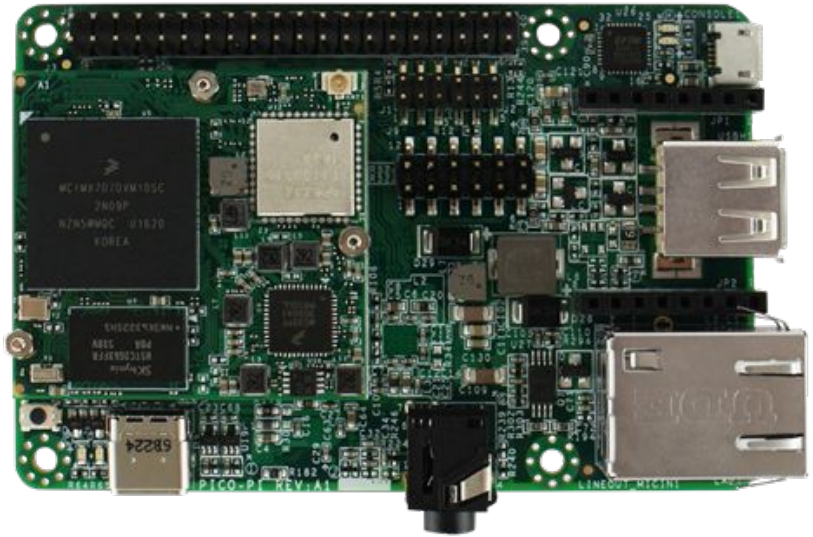


Rainbow HAT

- Seven APA102 multicolor LEDs
- Four 14-segment alphanumeric displays (green LEDs)
- Three capacitive touch buttons
- Green, red, blue LEDs
- BMP280 temperature and pressure sensor
- Piezo buzzer
- Breakout pins for servo, I2C, SPI, UART (all 3v3)

Rainbow HAT + NXP i.MX6UL

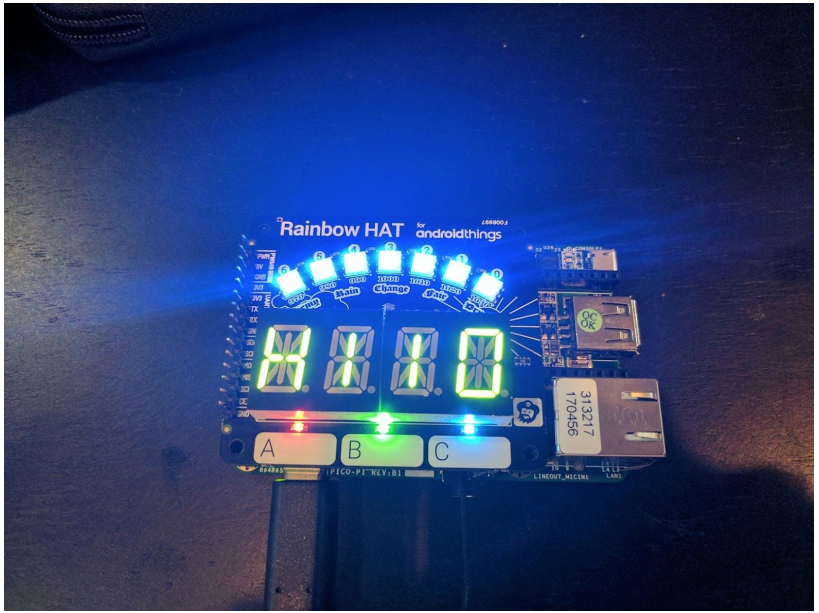
Getting started



Demo

https://github.com/rockerhieu/bitano

duet-debug.apk

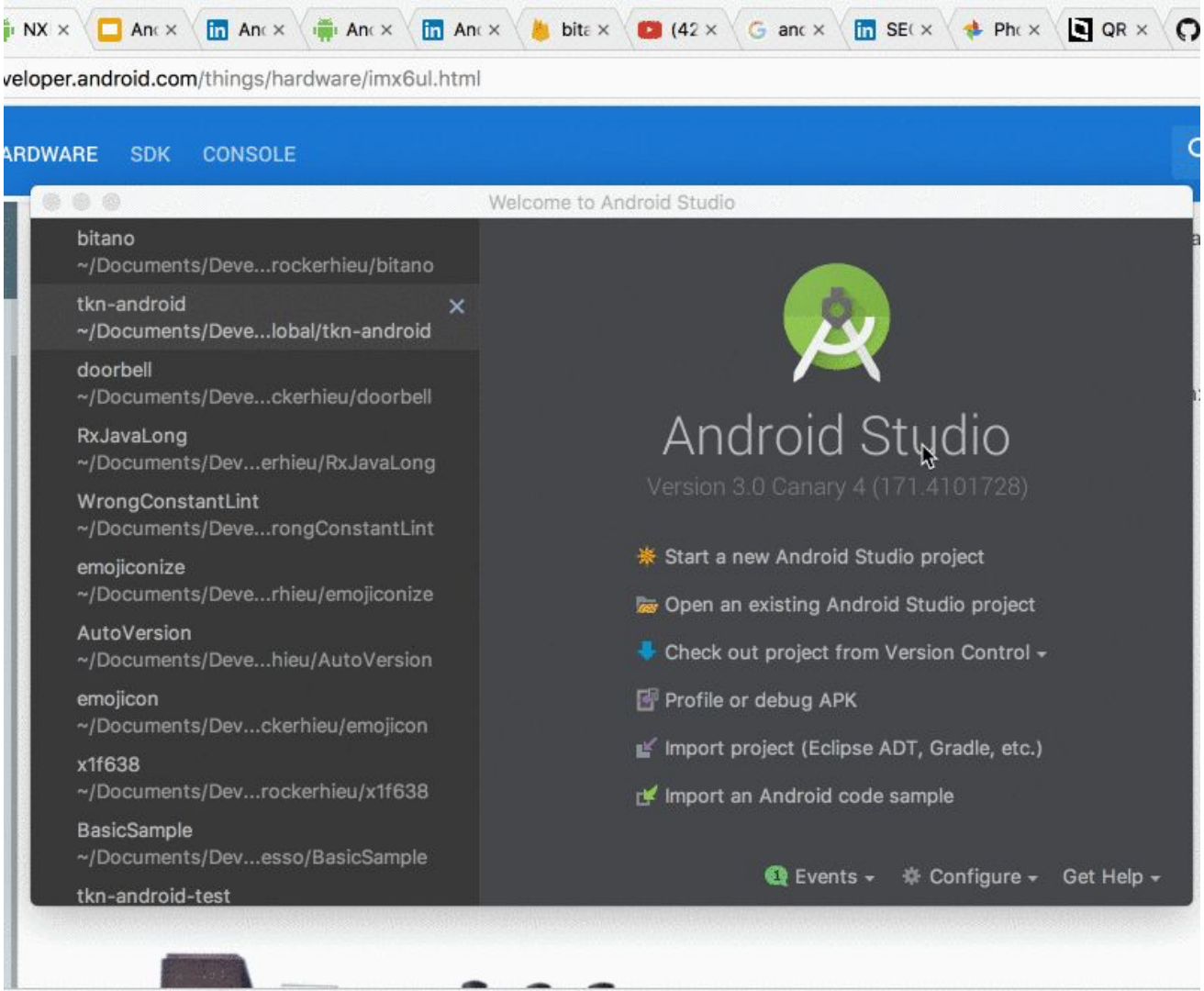


adb tool

```
$ adb devices

List of devices attached

000002564f839b02 device


$ adb shell
```

Connect to WiFi

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


veloper.android.com/things/hardware/imx6ul.html
ARDWARE
SDK
CONSOLE
Welcome to Android Studio
bitano
~/Documents/Deve...rockerhieu/bitano
tkn-android
~/Documents/Deve...lobal/tkn-android
doorbell
~/Documents/Deve...ckerhieu/doorbell
RxJavaLong
~/Documents/Dev...erhieu/RxJavaLong
WrongConstantLint
~/Documents/Deve...rongConstantLint
emojiconize
~/Documents/Deve...rhieu/emojiconize
AutoVersion
~/Documents/Deve...hieu/AutoVersion
emojicon
~/Documents/Dev...ckerhieu/emojicon
x1f638
~/Documents/Dev...rockerhieu/x1f638
BasicSample
~/Documents/Dev...esso/BasicSample
tkn-android-test
Android Studio
Version 3.0 Canary 4 (171.4101728)
Start a new Android Studio project
Open an existing Android Studio project
Check out project from Version Control
Profile or debug APK
Import project (Eclipse ADT, Gradle, etc.)
Import an Android code sample
Events
Configure
Get Help

AndroidManifest.xml

```
<application>
    <uses-library android:name="com.google.android.things" />
    <!-- ... -->
</application>
```

Activity

```
class MainActivity : Activity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
    }
}
```

Declare Activity

```
<activity android:name=".MainActivity">
  // launch from Android Studio
  <intent-filter>
     <action android:name="android.intent.action.MAIN" />
     <category android:name="android.intent.category.LAUNCHER" />
  </intent-filter>
</activity>
```

Declare Activity

```
<activity android:name=".MainActivity">
  // automatically launch on reboot
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.IOT_LAUNCHER" />
    <category android:name="android.intent.category.DEFAULT" />
  </intent-filter>
</activity>
```

Dependencies

```
dependencies {

    // ...

   compileOnly 'com.google.android.things:androidthings:+'

   implementation 'com.google.android.things.contrib:driver-rainbowhat:+'

}
```

Rainbow HAT driver

```
// Play a note on the buzzer.
Speaker buzzer = RainbowHat.openPiezo();
buzzer.play(440);
// Stop the buzzer.
buzzer.stop();
// Close the device when done.
buzzer.close();
```

Another Demo

https://github.com/rockerhieu/bitano

duet-debug.apk

Image recognition

- Use Cloud Vision to annotate image
- Build a “serverless” endpoint (like Parse Cloud Code)
  - Use Serverless framework
  - Deploy to Google Cloud Functions

Setup Google Cloud Functions

- Sign up for a Google Cloud Billing Account
- Create a new Google Cloud project, associate with the previous billing account
- Go to API Dashboard and enable these APIs and Services:
  - Google Cloud Functions
  - Google Cloud Deployment Manager
  - Google Cloud Vision
- Go to Credentials page
  - Create credentials -> create a Service account key in json format
  - Download and store the key at ~/.gcloud/keyfile.json

Setup serverless

```
$ npm install -g serverless

$ serverless create --template google-nodejs -p myproject

$ cd myproject

$ npm install

$ npm install --save @google-cloud/vision
```

Edit serverless.yml

```
service: myproject

provider:
 name: google
 runtime: nodejs
 project: <PROJECT_ID>
 credentials: ~/.gcloud/keyfile.json

plugins:
 - serverless-google-cloudfunctions

functions:
 first:
   handler: http
   events:
     - http: http
```

Edit index.js

```
'use strict';

const vision = require('@google-cloud/vision');

exports.http = (request, response) => {
 console.log(request.query);
 const uri = request.query.uri;
 const client = new vision.ImageAnnotatorClient();

 client.labelDetection(uri)
   .then((results) => { returnResult(results) })
   .catch((err) => { returnError(err) });
};
```

Deploy & test

```
$ serverless deploy

$ serverless log -f <function_name>
```

Another Demo

https://github.com/rockerhieu/bitano

duet-debug.apk

References

- https://developer.android.com/things/index.html
- https://serverless.com/blog/google-cloud-functions-application/
- https://thingspeak.com/
- https://www.blynk.cc/
- https://thingsboard.io/
- https://ifttt.com
- https://cloud.google.com/iot-core/
- https://aws.amazon.com/iot/
- https://azure.microsoft.com/en-us/suites/iot-suite/

Q & A

rockerhieu@gmail.com