WHITE PAPER

Developer's perspective: Why choose Android Things*?

Android Things* and the Internet of Things market

The Internet of Things (IoT) market is poised to expand by 30 times between now and 2020 (Gartner). Projections suggest that by 2020 there will be 200 billion connected objects, which is about 26 smart devices for every human (IDC, Intel, United Nations). To put that in perspective, IoT devices will easily outnumber phones, tablets, and laptops combined. Despite the promise of this astonishing growth, the current ecosystem of embedded operating systems and protocols is still fragmented.

Bruce Beare
Principal Software Architect
Intel Corporation

December 2016

"The out-of-box experience is typically the first impression a product creates, such as the ease with which a buyer can begin using the product. For hardware products, a positive OOBE can be created with logical easy-to-follow instructions and good quality of manufacturing.

For software, this often means easy installation and "Welcome" or "Initial Configuration" wizard screens that simplify elaborate set-up. The OOBE can also include the complete lack of such wizards."

Wikipedia

There are many challenges that limit adoption of IoT platforms, including complex technologies that require a diverse range of technical expertise, various security vulnerabilities, the lack of open standards for interoperability, and fragmented services. Combined, these can make for a poor developer experience, particularly when quickly scaling designs from prototype to production.

The Android Things* operating system was created with these challenges in mind. This paper discusses how Android Things brings simplicity to IoT software and hardware development by providing a simple, secure, and updated development and deployment model. This allows developers to easily and confidently build prototypes and to expedite their product's time-to-market.

Android Things* out-of-box experience (OOBE)

A successful first experience is critical to capturing the hearts and minds of software and hardware developers who might consider using Intel® architecture system on chips (SoCs) for Android Things devices. The measure of success for an independent software vendor or independent hardware vendor will vary depending their level of experience with particular development boards or with Android Things.

Success for experienced developers will be based on whether there is a common experience with boards that they have used previously, the availability of open source software or hardware that they can leverage for their device, and how quickly they can bring their development board design to market. Success for new developers will be based on the availability of support and additional help in the form of clear documentation, step-by-step tutorials, code labs, and active community forums.

Many existing IoT development boards (such as Raspberry Pi) are great for the prototyping stage. However, these boards often fall short in terms of providing an open hardware design and a maintainable software deployment model that is critical to scale prototypes to large deployments.

On the other hand, Android Things is about moving from a developer or prototype board to a finished product quickly and without requiring the expertise of an operating systems engineer.

The Android Things out-of-box experience is optimized for those who are experienced in using the Android Studio IDE. When developers load an Android Things compliant development board with a binary distribution of the Android Things OS provided by Google*, the board is immediately ready for software development using Google's Android Studio* Integrated Development Environment (IDE). Hardware prototyping is as easy as connecting wires to breadboards for GPIOs and required buses; and developers can develop in C, C++, JavaScript or Java* software languages.

Table of contents

"Android Things is about moving from a prototype board to a finished and scalable product quickly without requiring the expertise of an operating systems engineer."

Google provides code labs for their core development boards that introduce the development environment and walk new users through utilizing the board. New key APIs for Android Things (such as the Peripheral Manager) are highlighted in the code examples.

When you use **Intel® development boards with Android Things**, like the **Intel® Edison board** and the **Intel® Joule development board**, Intel® provides a Getting Started guide and code examples in addition to fulfilling all of the Android Things developer's OOBE requirements outlined in the table below. This empowers both new and experienced developers to quickly get up to speed and to develop IoT applications and hardware.

Familiar programming model and development environment

An Android Things developer board is likely to be used by developers who want to hook up devices using buses and GPIOs on the board and then control the connected devices using the Google Weave* (see Cloud Communication Ready section) protocol on Google Home, on a browser or a phone.

For these developers, a preloaded Android Things image is very helpful because they can add the software used to control the attached devices and register or control the device from the cloud. Since the operating system as distributed for the system on a module (SOM) by Google provides access to the system buses and GPIOs on the SOM, developers don't need to be operating systems experts to use Android Things with boards such as the Intel Edison or Intel Joule developer boards.

Android Things also provides an application API to customize the available GPIOs, UARTs and buses (I2S, I2C, SPI) for each board and make them available to the IoT application. Access to the device GPIOs and various buses is via the new Peripheral Manager API and is programmed from user level code in Java, C, or C++. The Peripheral Manager daemon is accessed over the binder via new Android APIs from client applications while the daemon provides mutually excluded access to the hardware buses via the kernel's /sys interface. Writing applications that utilize devices on system buses is simplified by using the libupm and libpmraa libraries with C, C++ and Java bindings.

Developer's OOBE requirements

EXPERIENCED ANDROID THINGS DEVELOPERS	NEW ANDROID THINGS DEVELOPERS
• Documents providing specific procedures and information for their particular development board. • Easy access to all required board support packages and tools. • Trouble-free building and loading of the OS image of choice, as well as access to all available tools for the OS. • A stable, sturdy board with easy access to buses and GPIO pins. • The board must not fry components or get irretrievably bricked. • Clear instructions on how and what to order so developers have everything that they need when they get started.	• Clear step-by-step instructions explaining how to load the latest Android Things image on their board. • Clear step-by-step instructions explaining how to find and run the Android Things code labs. • Clear step-by-step instructions explaining how to use the Peripheral Manager provided access to GPIOs and busses. • Availability of active support forums.

"Different programming models can coexist on an Intel Architecture Android Things device, providing flexibility to utilize the programming language that is best suited to the needs and experiences of developers."

Example code using libupm and the Peripheral Manager for various Intel architecture boards is available on GitHub. Binaries for these libraries are available on jcentral and can be easily integrated into your Android Studio application build and debug environment. The source code for these examples and libraries is available on GitHub, and application developers are free to contribute to or to rebuild the libraries and add additional device support as required.

In addition, Android Studio IDE provides developers with a highly popular Android application development experience. Android application engineers will feel right at home when developing code for Android Things since all of the Android APIs are available. The new APIs for IoT (such as Weave and the Peripheral Manager) are available with Java bindings and are integrated into the **online Android documentation**. IoT applications are structured exactly as other Android applications are and are tested using Android Studio via the Android Debug Bridge (ADB). Android Studio can use ADB to install and debug applications on custom devices as well.

Open Source and availability of source code

The Board Support Packages (BSP) for Intel Edison and Intel Joule modules are prebuilt by Google and distributed in the Google image for the SOM. Once Android Things reaches general availability, the source for the Intel BSP for Android Things will be available either via the Android Open Source Project (AOSP) tree or GitHub. Developers can rebuild Android Things for Intel architecture images almost entirely from open source.

The only licensed components are the Google libraries and firmware binaries. This openness allows developers to debug any issues that they encounter when working with their custom hardware, review their code for correctness, and address any security concerns. Additionally, developers can leverage the active open source Android communities for further help with their concerns.

Choice of programming languages

Android Things utilizes the programming paradigms and tools of the Android ecosystem. Chief among these tools is **Android Studio**. The Android Studio IDE runs on most host systems (Linux, OS X, Windows) and provides a highly optimized environment for application developers. Android things applications may be developed using Android studio in Java, as well as in C and C++. The IDE includes many application editing, debugging, organizing, packaging and optimizing tools.

Intel supports Apache Cordova* HTML5 and JavaScript* programming starting with the Intel Joule module. An **Apache Cordova** program is a great alternative for programming the device user interface and pulling in information, pictures, or media from the web without having to develop a custom Android application in Java. Check the Apache Cordova website for the latest status of Cordova for Android Things.

All of these programming models can coexist on an Intel Architecture Android Things device, providing flexibility to utilize the programming language best suited to the needs and experiences of developers.

Cloud communication ready

Android Things is designed for IoT using Google's Weave* platform. Weave is an open communications platform for IoT devices to connect to Google services, such as the Google Assistant. Weave will be built into Android Things to make it easy for developers to connect to the Weave cloud service on Google's Cloud* via Wi-Fi. The Weave cloud service allows secure registration of devices, propagation of commands, and storage of device states. The Weave platform supports popular device schemas and traits, and will broaden support to a wider range of devices over time.

With Weave, developers get the on-device communication infrastructure and the Weave cloud server ready to be deployed.

The Google IoT Developer Console is used to register and control IoT devices so developers don't need to set up their own cloud servers.

Note:
Weave will be added to Android Things in an update to the developer preview.

"Android Things is a unique development model in that hardware vendors do not need to customize an operating system or deploy a cloud."

Secure by design

The Android Things security model is based on Android and includes the latest security features and fixes. Android Things devices are fully locked down from power on so that each software loading stage verifies the integrity of the next stage before executing it. Security fixes are managed by Google and are automatically uploaded to the Android Things developer console on a regular basis. Developers can then test and approve these Google OS updates and security patches, as well as provide their own device updates using the same OTA infrastructure that Google uses for their own devices.

Communications with the Cloud are verified with a crypto algorithm to ensure communications are not altered via a man-in-the-middle attack. You can find more information on the **Android security model**.

Availability of developer/ prototype boards

There are a number of developer preview boards available from various manufacturers that are Android Things ready and include a prebuilt binary provided by Google. Intel provides the **Intel® Edison board** and the **Intel® Joule development board** for early prototyping and software development using Google's Developer Preview release of Android Things.

Both of these SOM boards are equipped with key system components, including CPU, PMIC, memory, storage, Wi-Fi, and BIOS. The SOM design greatly speeds up the hardware and software development cycle for hardware vendors by allowing hardware designers to concentrate on the peripherals that make their device unique. Hardware vendors should not need to be operating systems programmers or have expertise in Android board support development when working with a SOM design.

Final remarks

There are many choices available for developing an IoT connected device. Android Things offers a unique model because hardware vendors do not need to customize an operating system or spend extra effort to deploy a cloud. The security of IoT devices has become a large concern for the health of the internet and for the safety and privacy of device owners. With Android Things and software updates driven by Google, IoT devices have a far better chance of staying current, secured, and useful for many years.

"With Android Things and Google driven software updates, IoT devices have a far better chance of staying current, secured, and useful for many years."