Towards Embedded Systems

Motivational Role of Free Software

Mojtaba Barzegari

mbarzegary@msn.com

Software Freedom Day
Sharif University of Technology
29th September 2016

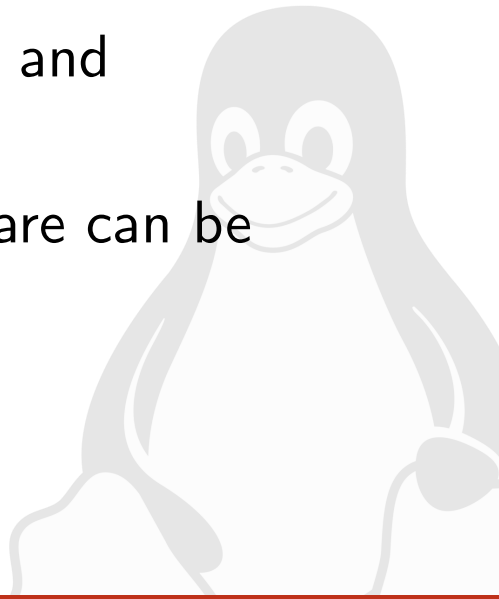
Outline

Free Software: a Prison Break

- Free Software: Four Essential Freedoms
- A bit about History
- GNU/Linux

Free Software: Four Essential Freedoms

- A program is considered free when its license offers to all its users the following four freedoms
  - Freedom to **run** the software for **any purpose**
  - Freedom to **study** the software and to **change** it
  - Freedom to **redistribute** copies
  - Freedom to **distribute** copies of **modified versions**
- These freedoms are granted for both commercial and non-commercial use
- They imply the availability of source code, software can be modified and distributed to customers
- **Good match for embedded systems!**



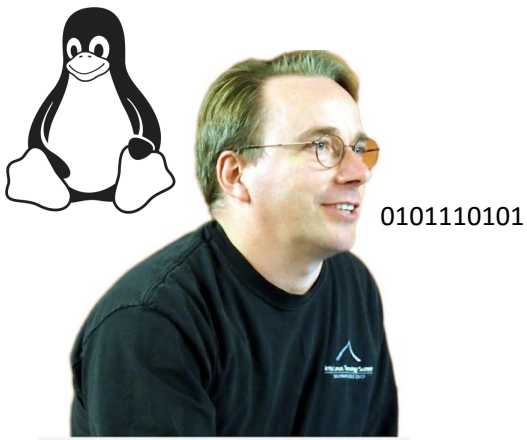
A bit about History

- **1983**, Richard Stallman, **GNU** project and the free software concept. Beginning of the development of gcc, gdb, glibc and other important tools
- **1991**, Linus Torvalds, **Linux kernel** project, a Unix-like operating system kernel. Together with GNU software and many other open-source components: a completely free operating system, **GNU/Linux**
- **1995**, Linux is more and more popular on server systems
- **2000**, Linux is more and more popular on embedded systems
- **2008**, Linux is more and more popular on mobile devices
- **2010**, Linux is more and more popular on phones

GNU/Linux

Richard Stallman

What you guys are referring to as Linux, is in fact, GNU/Linux, or as I've recently taken to calling it, GNU plus Linux. Linux is not an operating system unto itself, but rather another free component of a fully functioning GNU system made useful by the GNU corelibs, shell utilities and vital system components comprising a full OS as defined by POSIX.



0101110101

Play Your Notes: Embedded Systems

- What is an Embedded System?
- What makes Embedded Systems different?
- Embedded Systems Complexity
- To Be or Not to Be, That is the Question

What is an Embedded System?

An embedded system is a computer system with a **dedicated function** within a larger mechanical or electrical system, often with real-time computing constraints. It is **embedded** as part of a complete device often including hardware and mechanical parts. Embedded systems control many devices in common use today

What makes Embedded Systems different?

- Real-time operation
- Size
- Cost
- Time
- Reliability
- Safety
- Energy
- Security

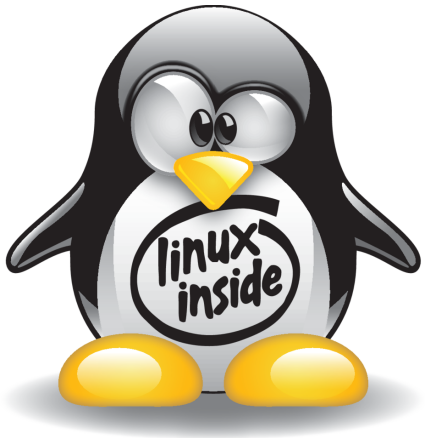
Embedded Systems Complexity

Moore's Law (Second edition)

The observation made in 1975 by Gordon Moore, co-founder of Intel, that the number of transistors per square inch on integrated circuits had doubled every double year since the integrated circuit was invented.

To Be or Not to Be, That is the Question

So, why does your TV run Linux? At frst glance, the function of a TV is simple: it has to display a stream of video on a screen. Why is a complex Unix-like operating system like Linux necessary?

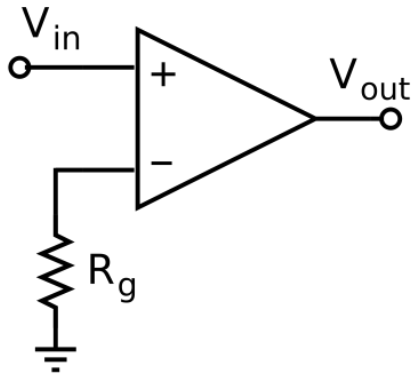


Digital Electronics and Operating Systems

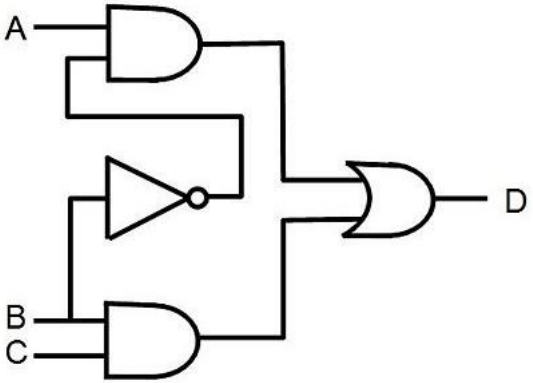
- Electronics: Analog and Digital
- Enter the Sandman: Microcontroller
- Productivity Nightmares: Primary Solutions
- Open Source OSes
- Linux and Free Software

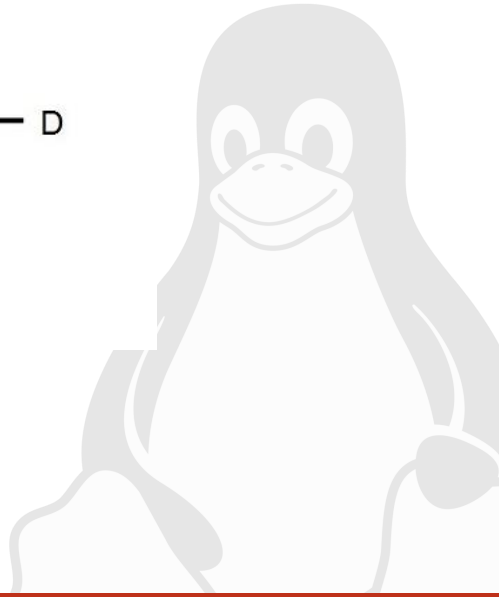
Electronics: Analog and Digital

Analog



Digital





Enter the Sandman: Microcontroller

- Gate + Clock ⇒ Microprocessor
- Microprocessor + Peripherals ⇒ Microcontroller
- Microcontroller + CoProcessors ⇒ Embedded Processor

Productivity Nightmares: Primary Solutions

Serious Challenges:

- Managing Peripherals
- Moore's Law
- Programming APIs
- Hardware Dependency

Recommended Solutions:

- Open Hardware
  - Arduino/Genuino
- Open Source Hardware Programming Platforms
  - Mbed
- Operating System

Open Source OSes

- **FreeRTOS**
- **RIOT**
- Contiki
- TinyOS
- OpenWSN
- Zephyr
- **GNU/Linux**

Linux and Free Software

Advantages of Linux and Free Software for embedded systems:

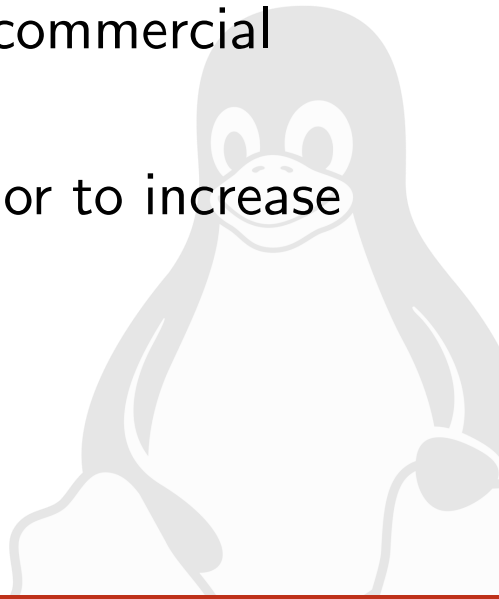
- Re-using components
- Low cost
- Full control
- Quality
- Easy testing of new features
- Community support

Re-using components

- The key advantage of Linux and open-source in embedded systems is the **ability to re-use components**
- The open-source ecosystem **already provides** many components for standard features, from hardware support to network protocols, going through multimedia, graphic, cryptographic libraries, etc.
- As soon as a hardware device, or a protocol, or a feature is wide-spread enough, high chance of having open-source components that support it.
- Allows to **quickly design and develop** complicated products, **based on existing components**.
- No-one should re-develop yet another operating system kernel, TCP/IP stack, USB stack or another graphical toolkit library.
- Allows to **focus on the added value of your product**.

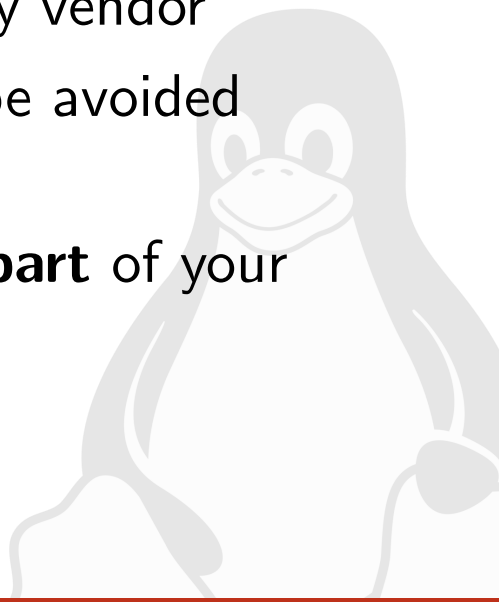
Low cost

- Free software **can be duplicated** on as many devices as you want, **free of charge**.
- If your embedded system uses only free software, you can reduce the **cost of software licenses to zero**. Even the development tools are free, unless you choose a commercial embedded Linux edition.
- Allows to have a higher budget for the hardware or to increase the companys skills and knowledge



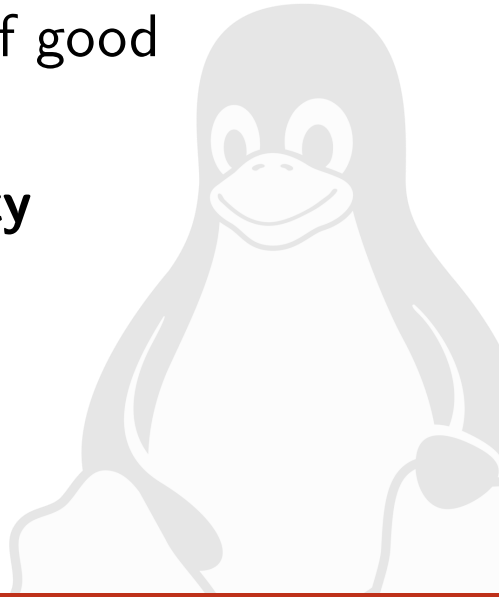
Full control

- With open-source, you have the source code for all components in your system
- Allows **unlimited modifications, changes, tuning, debugging, optimization**, for an unlimited period of time
- Without lock-in or dependency from a third-party vendor
- To be true, non open-source components must be avoided when the system is designed and developed
- Allows to have **full control over the software part** of your system



Quality

- Many open-source components are **widely used**, on millions of systems
- Usually **higher quality** than what an in-house development can produce, or even proprietary vendors
- Of course, not all open-source components are of good quality, but most of the widely-used ones are.
- Allows to **design your system with high-quality components** at the foundations

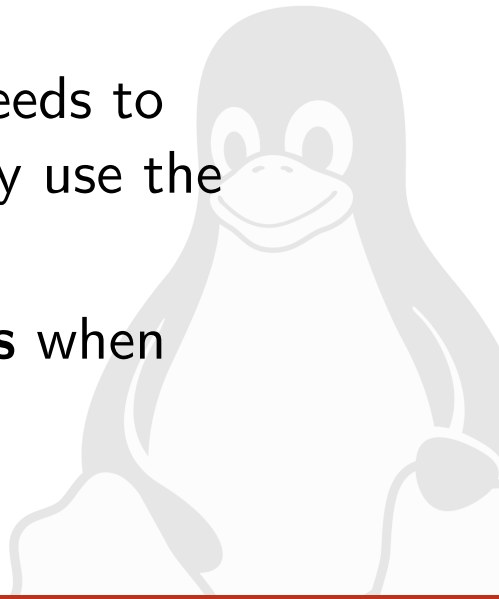


Easy testing of new features

- Open-source being freely available, it is **easy to get a piece of software and evaluate it**
- Allows to **easily study several options** while making a choice
- Much **easier than purchasing** and demonstration procedures needed with most proprietary products
- Allows to **easily explore new possibilities** and solutions

Community support

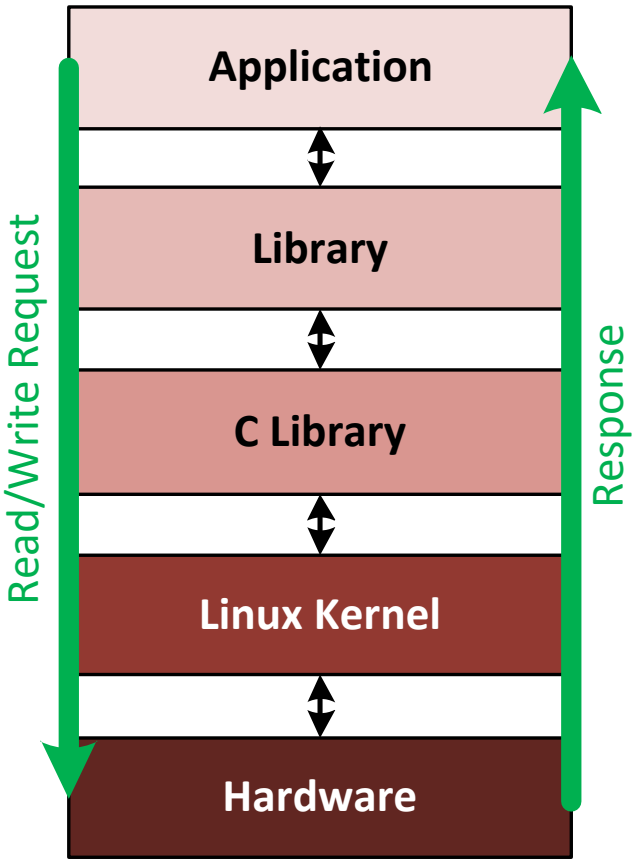
- Open-source software components are developed by communities of developers and users
- This community can provide a **high-quality support**: you can directly contact the main developers of the component you are using. The likelyhood of getting an answer doesn't depend what company you work for.
- Often better than traditional support, but one needs to understand how the community works to properly use the community support possibilities
- Allows to **speed up the resolution of problems** when developing your system



Linux Disadvantages!!!

- Linux certainly is a robust and developer-friendly OS
- Linux will have many uses in embedded devices, particularly ones that provide graphically rich user interfaces.
- Linux has a disadvantage when compared to other embedded operating systems:
  - Memory footprint
  - It simply will not run on 8 or 16-bit MCUs
  - Real-time operations problem

Real-Time OS: Dream or Reality??



Real-Time OS: Dream or Reality??

Typical Problems:

- Controlling Servo Motors
- Generating High Frequency Signals
- Generating PWM and Duty Cycles
- Sampling a Signal (e.g. Audio Signal)
- Controlling a Sensitive Robot
- Developing Flight Controllers

Real-Time OS: Dream or Reality??

Suggested Solutions:

1. Using High-Speed Interfaces
   - PCI
   - Ethernet
   - USB
   - RS232
2. Developing System Drivers
3. Designing Auxiliary Hardware and Co-Processor

Internet of Things: Domination of Free Software

- What is Internet of Things (IoT)?
- Things Connected Through Internet
- IoT Applications
- Top Free and Open Source IoT Projects
- Rise of Linux: IoT-ready Hardwares
- Beyond the IoT: The Computation Thunderbolt

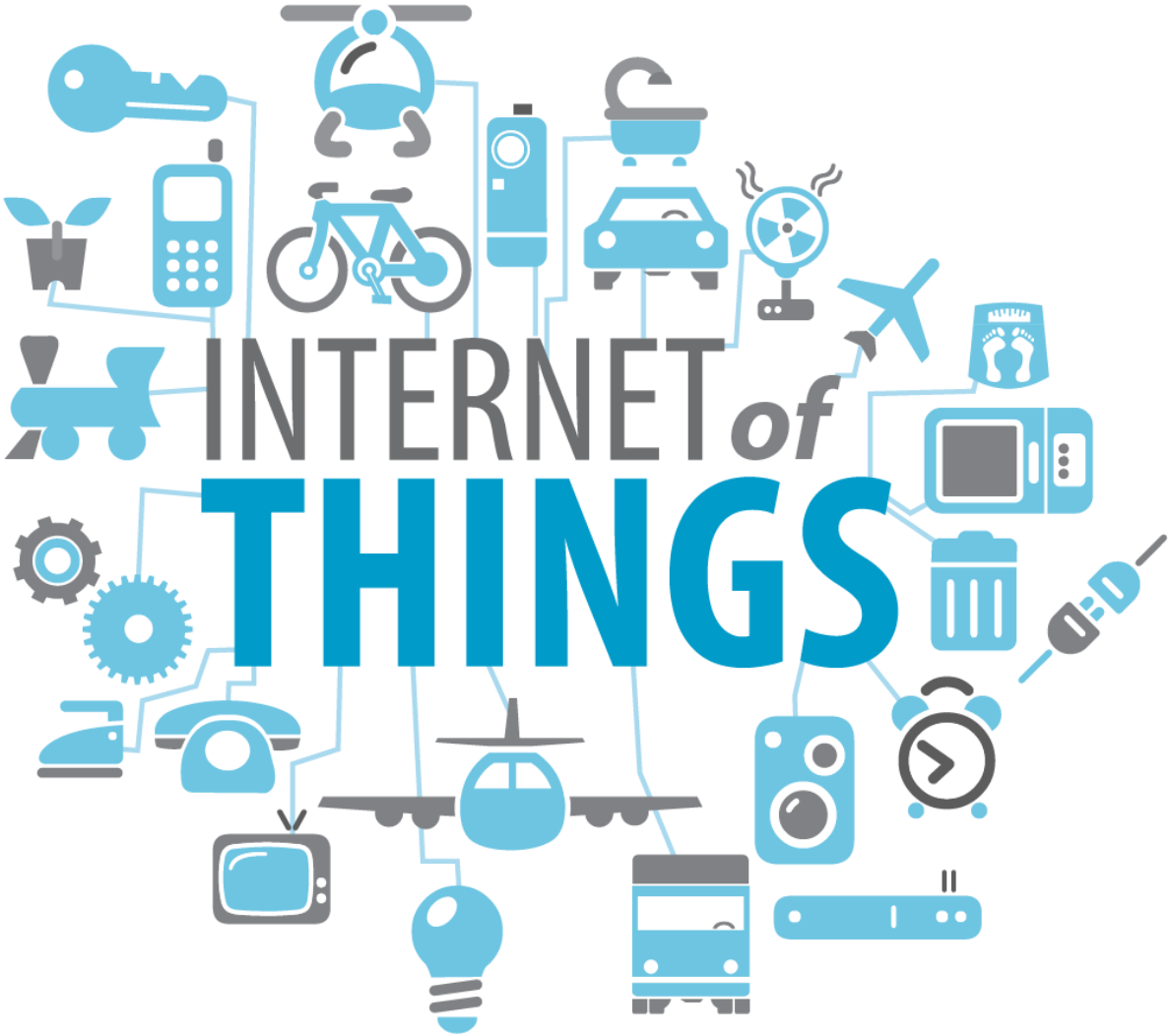
What is Internet of Things (IoT)?

What is IoT?

[**Wikipedia**]: The network of physical objects or things embedded with electronics, software, sensors, and connectivity to enable objects to exchange data with the manufacturer, operator and/or other connected devices.

What is IoT?

[**WhatIs**]: A scenario in which objects, animals or people are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.



Things Connected Through Internet

IoT Applications

- Smart home
- Wearables
- Smart City
- Smart grids
- Industrial Internet
- Connected Car
- Connected Health
- Smart Retail
- Smart Supply Chain
- Smart Farming

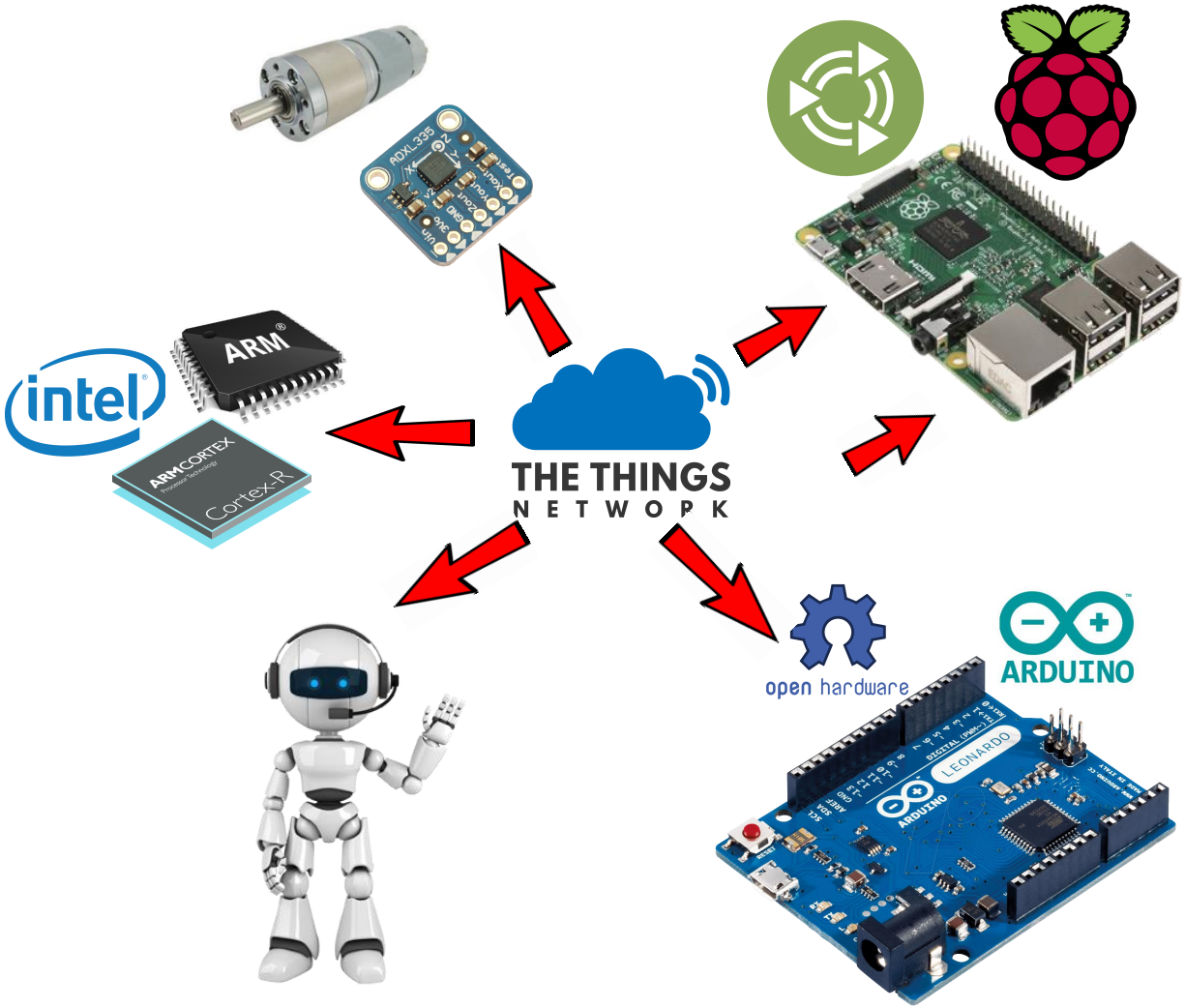
Top Free and Open Source IoT Projects

- AllSeen Alliance (Interoperability framework)
- DeviceHive (Big Data analytics)
- DSA (Distributed Services Architecture)
- Eclipse IoT (Java-based rules-based routing engine)
- Kaa (scalable end-to-end IoT framework)
- Macchina.io (Web-enabled, modular and extensible JavaScript and C++ runtime)
- Open Connectivity Foundation (IoTivity)
- OpenIoT (Java-based connectivity framework)
- PlatformIO (Python-based suite for remote data access)

Rise of Linux: IoT-ready Hardwares

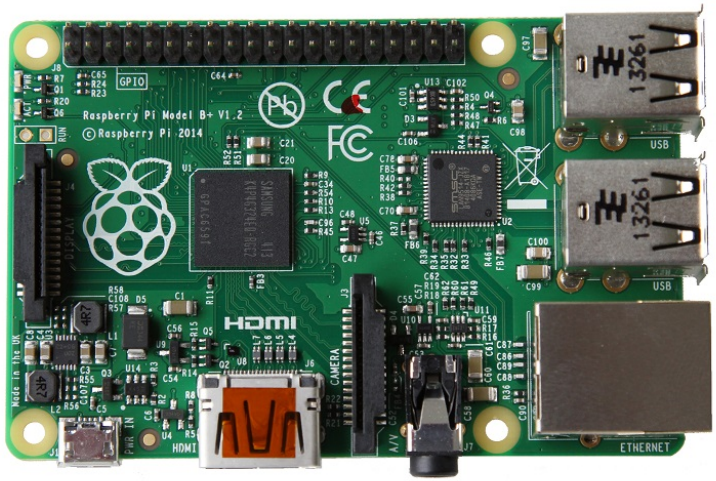
- Yocto Project!!!
- Raspberry Pi
- Intel Edison
- Intel Joule
- BeagleBone Black
- Qualcomm DragonBoard
- MinnowBoard MAX
- Orange Pi

IoT and Linux



Raspberry Pi

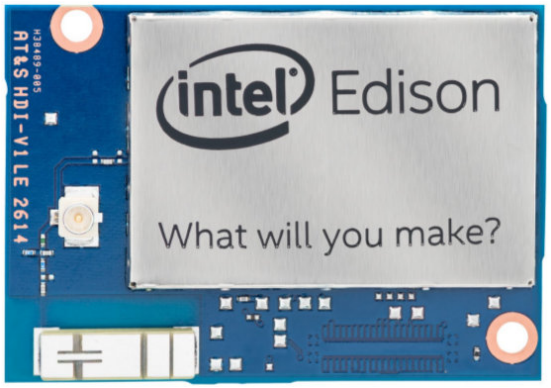
- **SoC**: Broadcom BCM2837
- **CPU**: 4 ARM Cortex-A53, 1.2GHz
- **GPU**: Broadcom VideoCore IV
- **RAM**: 1GB LPDDR2 (900 MHz)
- **Networking**: Ethernet, 2.4GHz 802.11n wireless
- **Bluetooth**: Bluetooth 4.1 Low Energy
- **Storage**: microSD
- **GPIO**: 40-pin header



Intel Edison

- **CPU**: Dual-core Intel Atom @500 MHz
- **RAM**: 1 GB LPDDR3
- **Storage**: 4 GB eMMC + micro SD card connector
- **Connectivity**: Dual band Wi-Fi and Bluetooth 4.0
- **USB**: 1x micro USB connector

- **I/Os**:
  - 2x UART
  - 2x I2C
  - 1x SPI with 2 chip selects
  - 1x I2S
  - 12x GPIO including 4 capable of PWM output



Intel Joule

- **CPU**: Intel Atom T5700 64-bit quad-core @1.7GHz/2.4GHz
- **RAM**: 4GB LPDDR4 RAM
- **GPU**: Intel HD Graphics with 4K video capture and display
- **Storage**: 16GB eMMC memory
- **Connectivity**: 802.11ac Wi-Fi with MIMO and Bluetooth 4.1

Beyond the IoT: The Computation Thunderbolt

These devices can be utilized in standalone computations:

- Speech Recognition
- Machine Learning
- Image Processing
- Grid Computing
- Video Encoding/Decoding

Conclusion

Why Reinvent The Wheel When You Don't Have To???

Thanks for Your Attention

Question & Answer