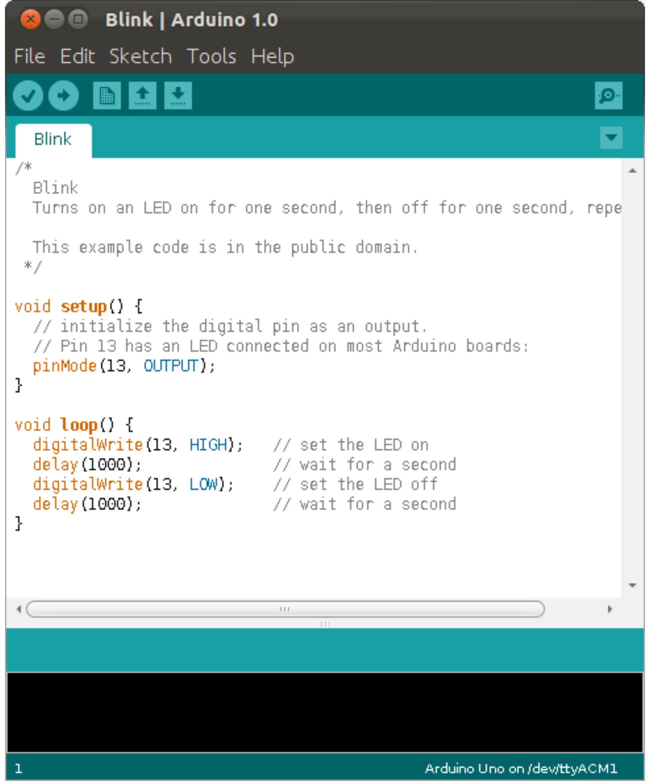
Internet of Things with Intel
A2 – IoT boards

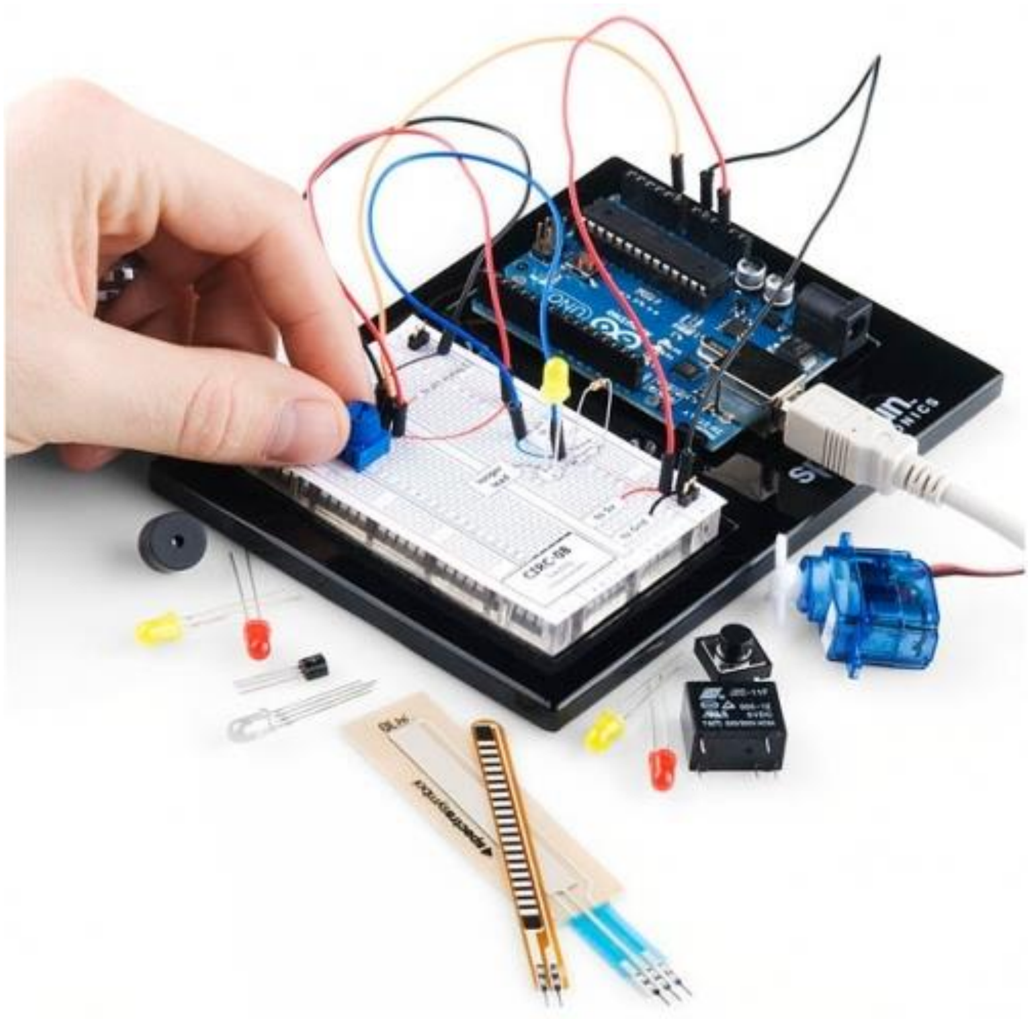
paul.guermonprez@intel.com

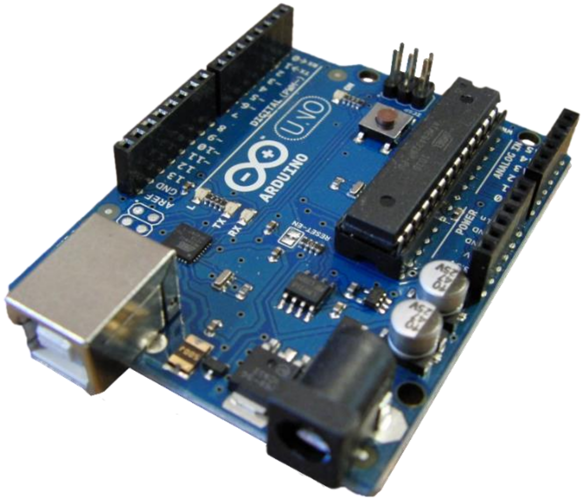
History

Arduino

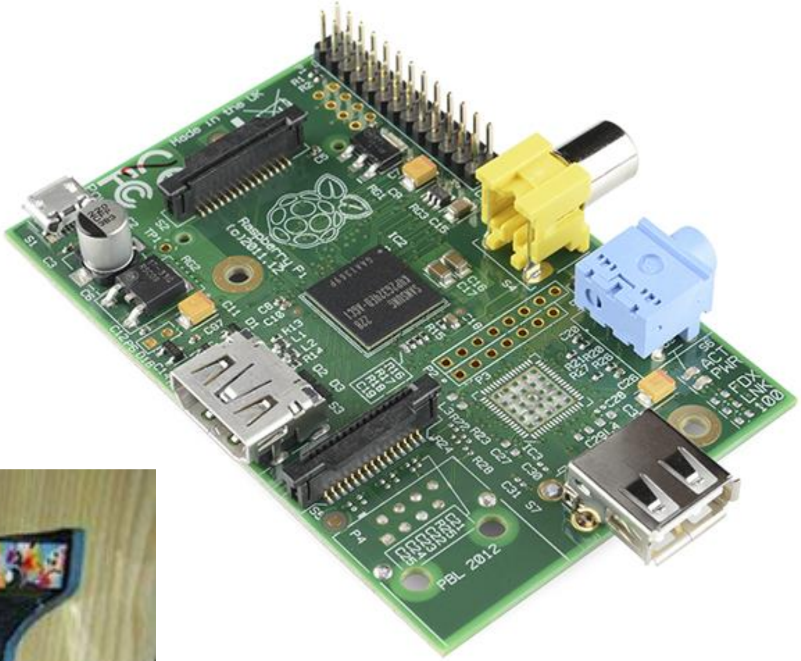


```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repe

  This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  delay(1000);              // wait for a second
}
```





Raspberry Pi



Evaluation of IoT boards

Microcontroller vs Processor

Arduino is using a 8-bit microcontroller. It’s simple and predictable for people new to software development.

But it’s impossible to install a full system like linux, so it’s limited by nature.

To have a full OS with network, the programming language of your choice and a lot of potential, a processor is required.

Digital / Analog IO

If you develop on your laptop, you can only plug devices with high level IO like USB. But most IoT sensors are a lot simpler than that, using low level analog and digital IO.

To be really useful as an IoT platform,
you must have digital and/or analog IO.

Note : Some IoT platforms only have digital IO,
others like Intel Galileo and Intel Edison
have both digital and analog IO.

Graphics

Many wearable projects like watches have a touch screen, or a projector for glasses. They look like a lot like small mobile devices, and require graphics.

We have such platforms at Intel.

But for all the other IoT projects,
you don’t need and don’t want graphics.

Remember : display = interaction
= brain attention share = not scalable.

Networking

Network connectivity is important in IoT.

You may take a platform without wireless internet and add a dongle for prototyping, but :

- It's harder to switch to production
- Power optimizations are limited
- Dongle often propose limited features : Bluetooth instead of Bluetooth Low Energy.
- Integration with software-OS is not always trivial.

Advice : take platforms with great networking inside for prototyping then production.

Form factor

It’s convenient to have large boards with lots of IO for prototyping. But you’ll need to design a new board for production. It takes a lot of money, time and skills.

Or you take a board with a modular design : your compute module has all the complex parts you won’t redesign, and the simper connectivity board can be replaced or designed easily.

Power features

A big difference between IoT prototypes and production is total power consumption.

You would not plug a 3G dongle on a desktop and call it a mobile phone, right ? Same for IoT.

You need :

- a very efficient processor,
  with advanced sleep/hibernation features
- power optimized wireless
- great integration of all parts
- lots of software, driver and OS optimization

OS

We all like to prototype with our desktop OS.

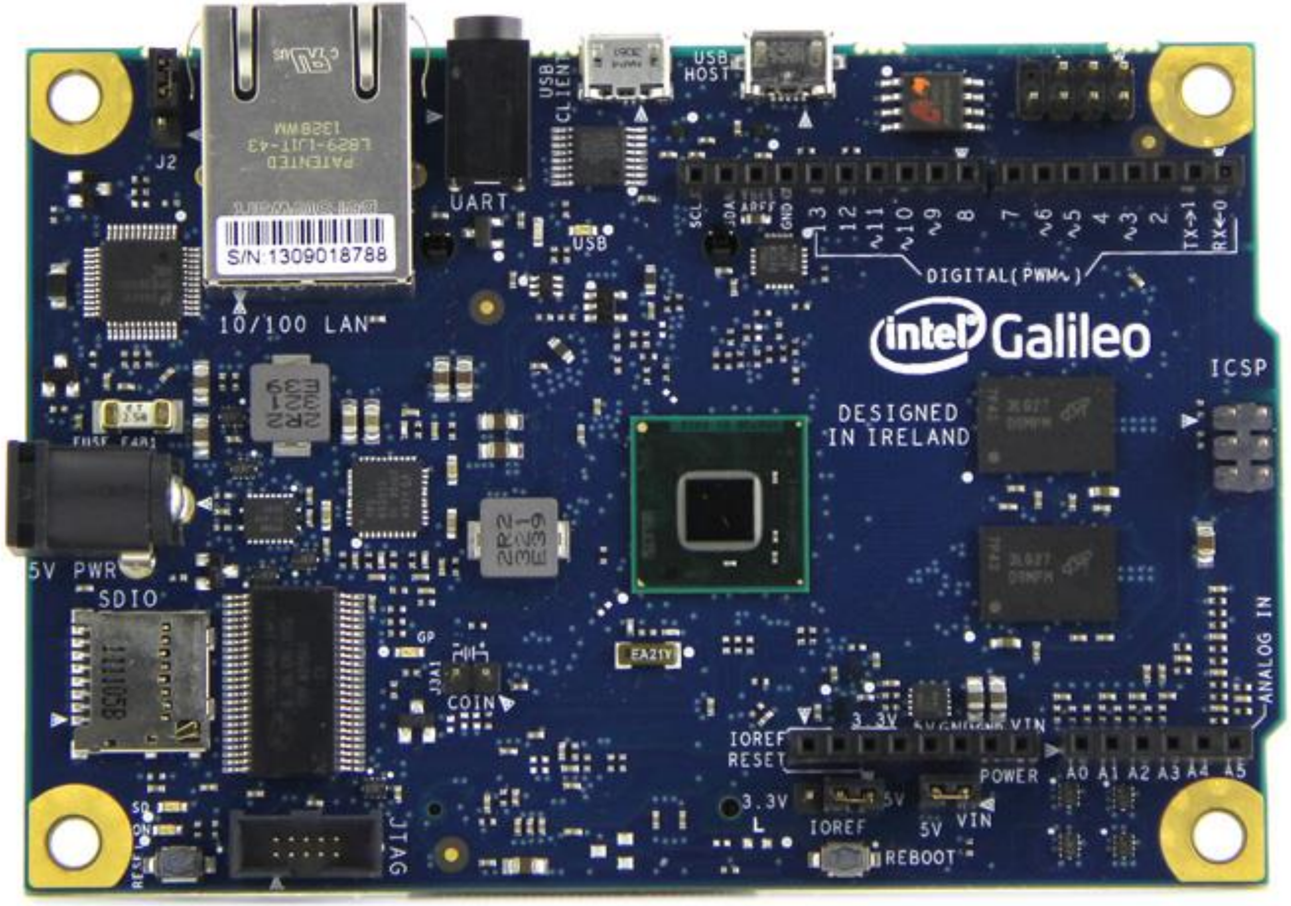
It can be a big linux distro like Ubuntu, Windows 10, OSX … it’s easy, all the packages are readily available.

But a professional grade embedded project requires to start from scratch, perhaps use a substitute of glic, control each piece of code added to the system and integrate with a large team of software developers.

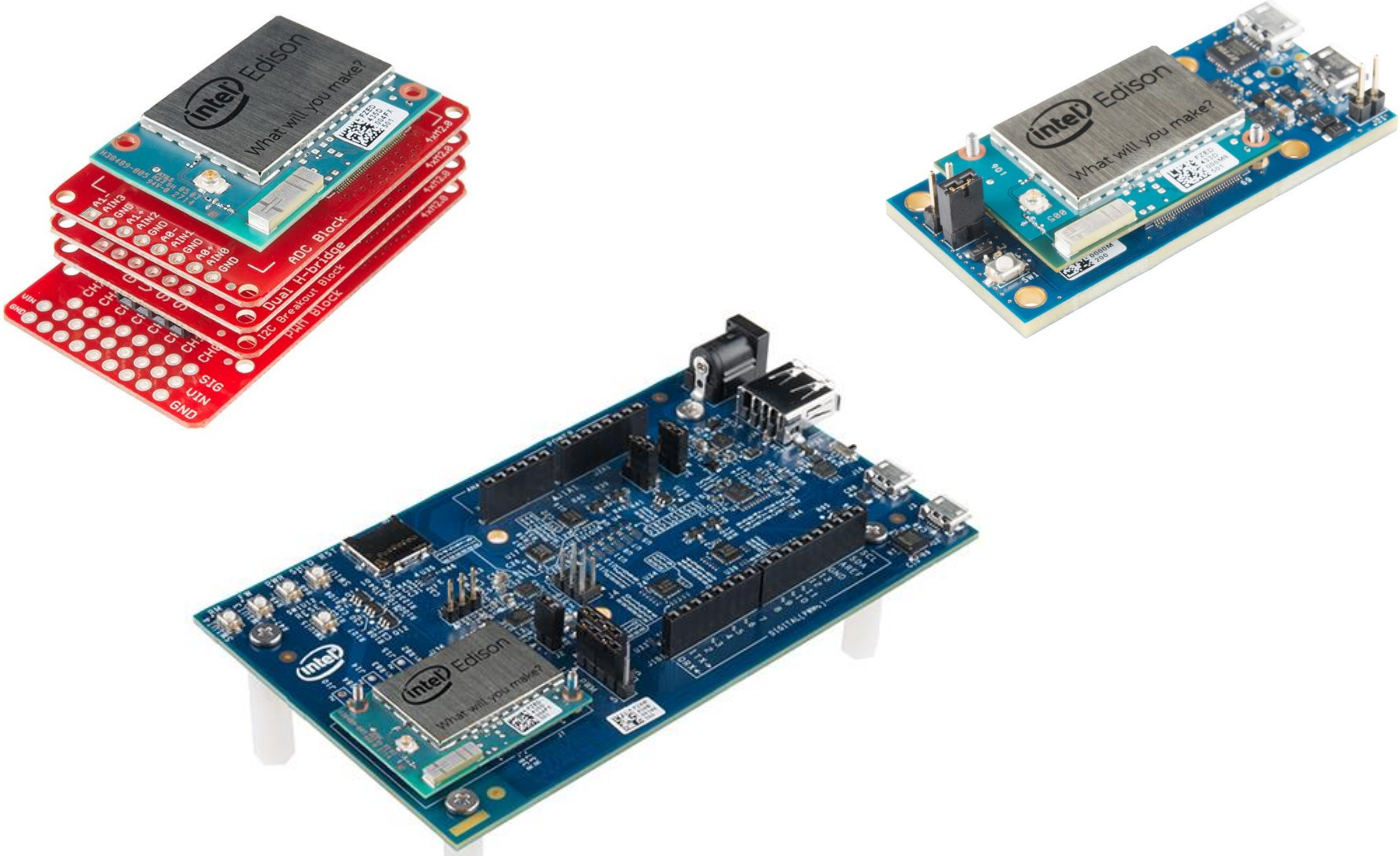
A typical open source OS for professional projects is Yocto.

Intel IoT boards

Intel Galileo



Intel Edison



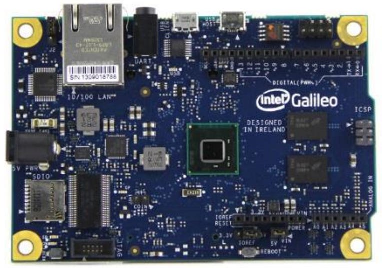
Intel Curie

But also : HDMI Stick, NUC

Intel IoT boards

Intel Galileo Specifications

Intel Galileo Specs

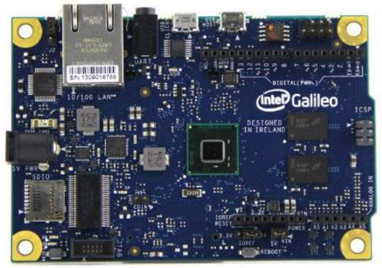


Intel Quark System-on-Chip (SoC) x1000
with a 32bit core running at 400MHz.

Connections : mini-PCI Express (for Wifi-Bluetooth module), 100Mb Ethernet port, Micro-SD slot, RS-232 serial port, USB Host, USB Client.

Form factor : Arduino compatible pins.

Intel Galileo Specs



The board can run :

- Linux Yocto by default from Intel.
  (Yocto is an open source linux used by embedded professionals)
- Debian variant by the community.
- Arduino style code by an emulator.
- Windows by Microsoft.
- In some cases, WindRiver solutions.

Intel IoT boards

Intel Edison Specifications

Intel Edison Specs

22 nm Intel SoC that includes :

- a dual-core, dual-threaded Intel Atom CPU at 500 MHz, running linux
- a 32-bit Intel Quark microcontroller at 100 MHz, running RTOS

Memory : 1 GB LPDDR3

Storage : 4 GB eMMC

Wifi : a/b/g/n

Bluetooth : 4.0 Low Energy

Intel Edison Specs

40 GPIO :

- SD card : 1 interface
- UART : 2 controllers (1 full flow, 1 Rx/Tx)
- I2C : 2 controllers
- SPI : 1 controller with 2 chip selects
- I2S : 1 controller
- GPIO : Additional 12 (with 4 capable of PWM)
- USB 2.0 : 1 OTG controller

Form factor : modular. The connectivity boards are available from Intel, Sparkfun, …

Intel Edison Specs

The board can run :

- Linux Yocto by default from Intel.
  (Yocto is an open source linux used by embedded professionals)
- Debian variant Ubilinux by the community.
- Arduino style code by an emulator.

License Creative Commons – By 3.0