

Internet of Things with Intel

A3 – Programming Methods

paul.guermontprez@intel.com



Options

1. [Arduino IDE](#) for Intel IoT Platforms
2. Intel [XDK](#) for IoT
3. Intel IoT [SDK](#) with eclipse
4. [Linux](#) development [on the board](#)
5. Conclusion



1. Arduino IDE for Intel IoT Platforms

Emulating Arduino on Intel



Arduino IDE for Intel IoT platforms

Installation :

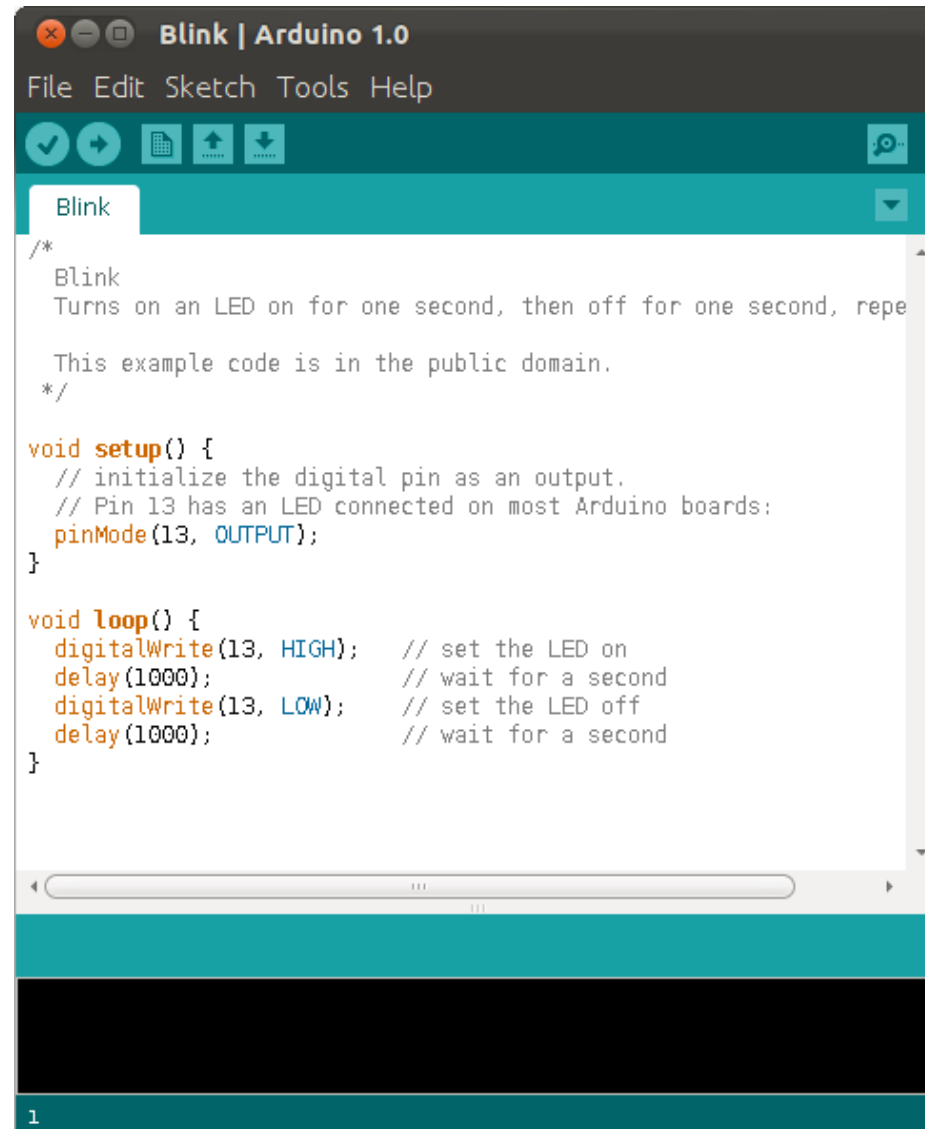
- On your PC : The Arduino IDE is easy to install and start.
Note : You'll need the Intel version of the Arduino IDE.
- Boards : Arduino emulator is already on Galileo and Edison, just update the firmware. Easy.

Arduino IDE is designed for very simple projects only, but won't go very far. Excellent docs and community.

<https://software.intel.com/en-us/iot/downloads>



Arduino IDE for Intel IoT platforms



The screenshot shows the Arduino IDE window titled "Blink | Arduino 1.0". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for checking, undo, redo, and uploading. The main editor area displays the "Blink" sketch, which is a standard Arduino program for blinking an LED. The code is as follows:

```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * 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
}
```

The IDE interface includes a status bar at the bottom showing the line number "1".



Problem #1 : Potential

The original Arduino is based on a **microcontroller**.

There's an **software emulator** simulating this microcontroller on Galileo and Edison.

But Intel IoT platforms are **processor** based.

That's why, with this method, you're only using a minuscule fraction of the processing and networking potential of the platform.

You can make IoT with this method,
but **basic IoT, not smart IoT**.



Problem #2 : Compatibility

This **software emulator** can't fully emulate the microcontroller, specifically the real time aspects.

First because the OS is **not real time** itself, but also because the **granularity** would not be the same.

Result : **some core Arduino IDE libraries are not available**,
some sensors like depth sensor or some LED strips
won't work.



Conclusion

Public for the Arduino IDE :

- Electronics with very limited software skills appreciate the Arduino IDE
- Makers with Arduino experience !
- With an minimal interest in software and Linux
- With no interest in moving to production later



2. Intel XDK IoT Edition

Sandbox development for web developers



Intel XDK IoT Edition

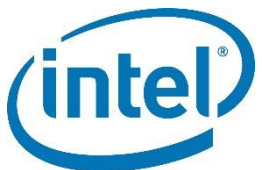
Familiar for web developers familiar with JavaScript.
But if you like NodeJS on linux, check the option 4 : linux.

Limited functionalities and libraries
but validated kits of **Plug'n'Play sensors**.

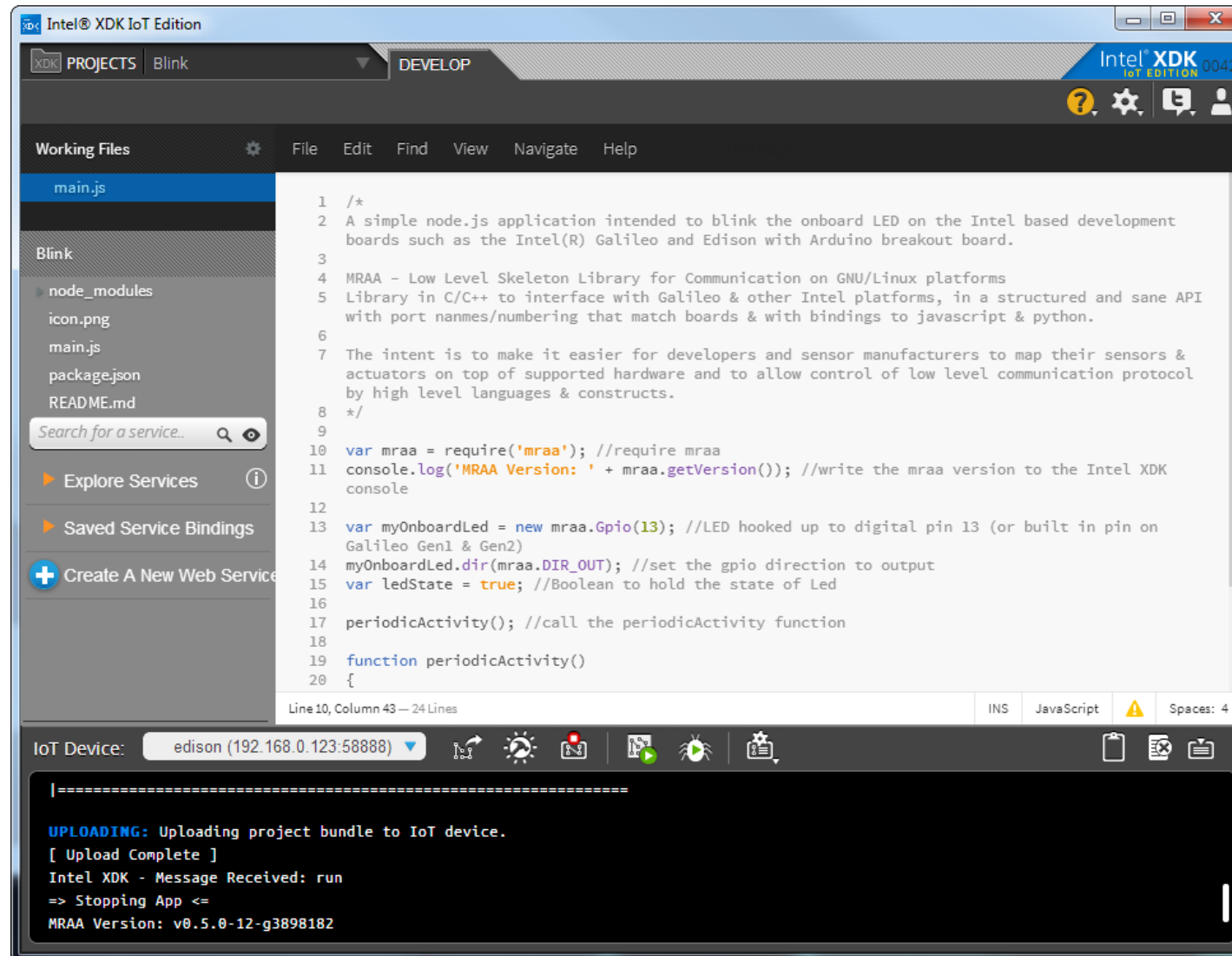
Easy to install and start the IDE,
but flashing the board is not yet easy.

You'll need to find and **flash a microSD card**.

<https://software.intel.com/en-us/iot/downloads>



Intel XDK IoT Edition



Intel XDK IoT Edition

Public :

- Good for web devs or beginners in software development
- If your project plans to stay within the set of validated PnP sensors
- Not interested in using linux at all

Compared to Arduino, you can add a lot more code with the Intel XDK. **You can evolve** into linux-NodeJS development if you want to go further.



3. Intel IoT Developer Kit

With Eclipse IDE development



Intel IoT SDK

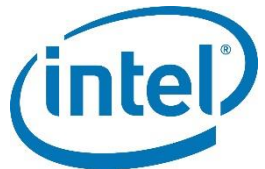
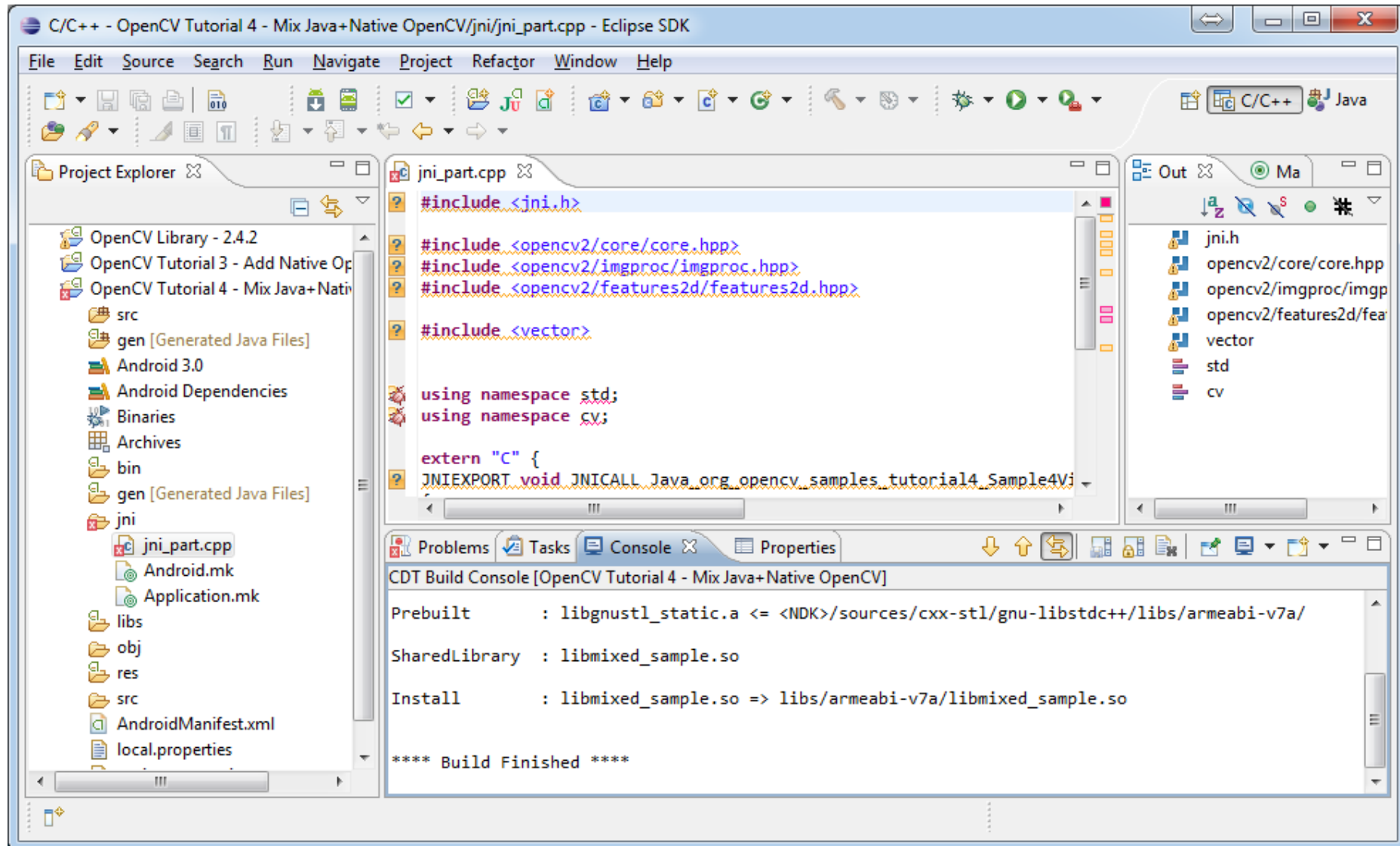
Eclipse on your workstation,
communicating with the IoT board.

You don't really access the board itself
or the linux OS running on it.

Mainly C development.



Intel IoT SDK



Intel IoT SDK

Public :

- Fan of C development with Eclipse
- Don't need linux access
or to add linux packages for your project
- Not interested in using linux at all
- Want to code high performance C code

Compared to the XDK, C is less fun than JavaScript.

You can evolve into linux-C development
if you want to go further.



Linux development on the board
Low level, no limits, linux style

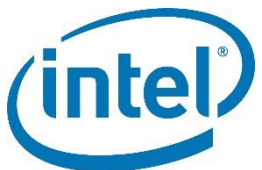


Linux development on the board

You ssh to the board from network or usb,
install all the packages you need,
and code with any language on linux.

- Nothing to install on your PC : Great for workshop setups.
- Edit with nano, emacs or vi.
- Compile with gcc on the board.
- Or use VMs like Python, NodeJS, ...
- Play with linux services like bluez for advanced Bluetooth features.
- Interact with the Yocto professional embedded distribution,
or even rebuilt your distro from scratch.

Highly recommended on linux for IoT : [NodeJS with Cylon](#).



Linux development on the board

Public :

- good for linux, perfect if you like Raspberry Pi
- connect any sensor on earth
- you control everything at low level,
required if you want to move to production later
- you don't want or have no install
to install something on your PC



Conclusion

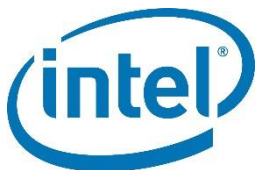


Conclusion

Good news : there's a method for everyone.

Evolution paths :

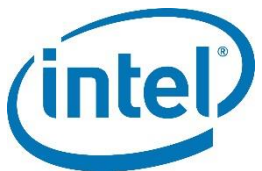
- Arduino : no easy evolution
- XDK : you can evolve towards linux or SDK
- SDK : you can evolve towards linux
- Linux : you can evolve towards production and professional IoT embedded development



Conclusion

Because this course is designed for universities, specifically software students, we'll use the **Linux method**, with a **strong focus on NodeJS/Cylon** for rapid prototyping.

It is the best method to understand **how things work inside** and learn.



License Creative Commons – By 3.0

You are free:

- **to Share** — to copy, distribute and transmit the work
- **to Remix** — to adapt the work
- to make commercial use of the work

Under the following conditions:

- **Attribution** — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

With the understanding that:

- **Waiver** — Any of the above conditions can be waived if you get permission from the copyright holder.
- **Public Domain** — Where the work or any of its elements is in the public domain under applicable law, that status is in no way affected by the license.
- **Other Rights** — In no way are any of the following rights affected by the license:
 - Your fair dealing or fair use rights, or other applicable copyright exceptions and limitations;
 - The author's moral rights;
 - Rights other persons may have either in the work itself or in how the work is used, such as publicity or privacy rights.
- **Notice** — For any reuse or distribution, you must make clear to others the license terms of this work. The best way to do this is with a link to this web page.

<http://creativecommons.org/licenses/by/3.0/>

