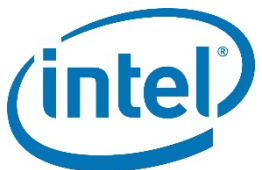


Internet das Coisas com Intel

A3 – Métodos de programação

paul.guermonprez@intel.com



Opções

1. **IDE Arduino** para plataformas IoT da Intel
2. Intel **XDK** para IoT
3. Intel IoT **SDK** com Eclipse
4. Desenvolvimento com **Linux na própria placa**



1. IDE Arduino para plataformas IoT da Intel

Emulando Arduino no Intel



IDE Arduino para plataformas IoT da Intel

Instalação :

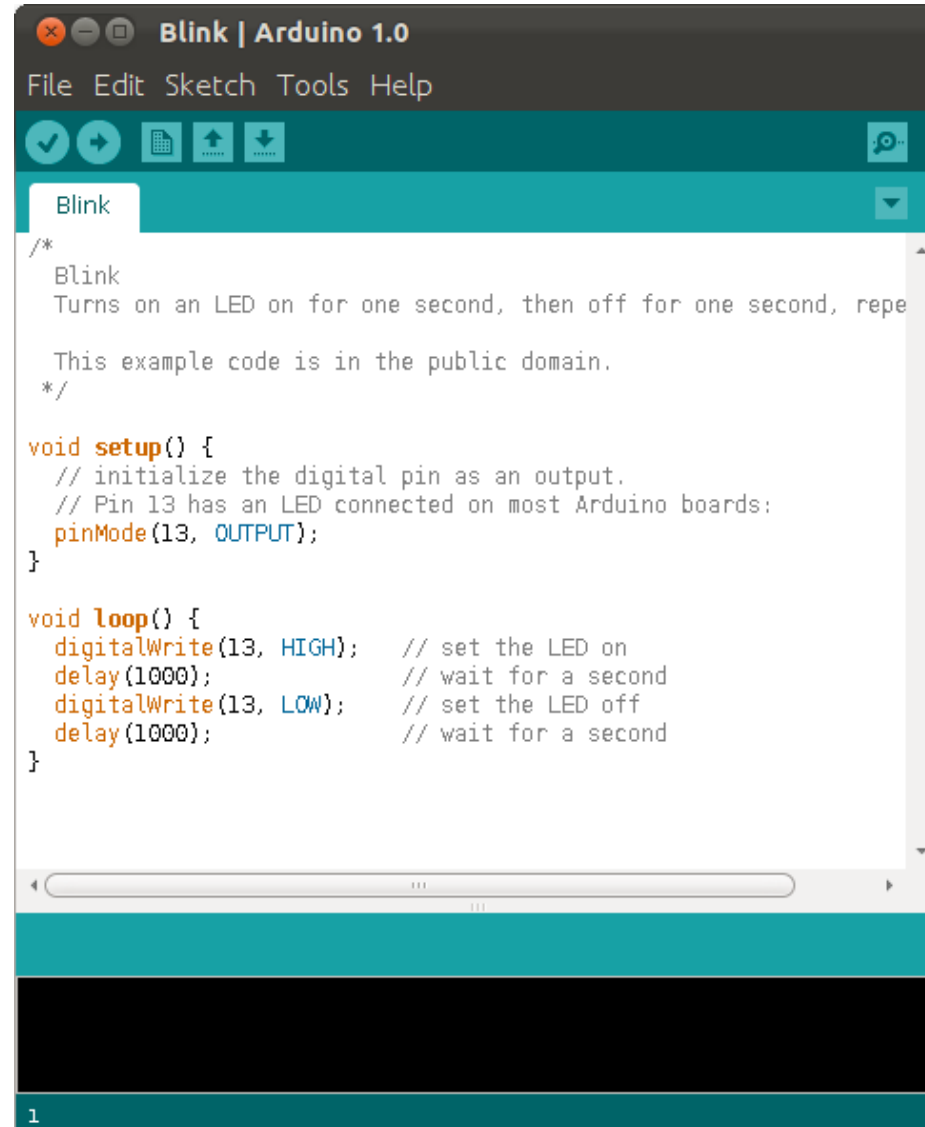
- No PC: O IDE Arduino é fácil de instalar e configurar.
Nota: É precisa a versão da Intel do IDE Arduino.
- Nas placas: o emulador de Arduino vem pré-instalado nas placas Galileo e Edison, só atualizar o firmware. Fácil.

O IDE Arduino foi desenhado para projetos simples, não para projetos complexos. Excelente documentação.

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



IDE Arduino para plataformas IoT da Intel

A screenshot of the Arduino IDE interface. The window title is "Blink | Arduino 1.0". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for checkmark, undo, redo, upload, and download. The main editor area shows a sketch named "Blink" with the following code:

```
/*
 * 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
}
```



Problema #1 : Potencial

O Arduino original é baseado num **microcontrolador**. Existe um **emulador software** do microcontrolador para Galileo e Edison.

Mas as plataformas IoT da Intel possuem um **processador**. É por isso que, seguindo esse método, só uma minúscula fração do potencial de processamento e de rede da plataforma esta sendo usado.

É possível desenvolver soluções IoT com este método, mas **básicas, não soluções IoT inteligentes**.



Problema #2 : Compatibilidade

O **emulador software** não consegue emular plenamente o microcontrolador, particularmente as características de tempo real.

Primeiramente porque o SO **não é em tempo real**, mas também porque a **granularidade** é diferente.

Resultado: **algumas bibliotecas do IDE Arduino não estão disponíveis**, alguns sensores (sensor de profundidade e alguns LED) não funcionam.



Conclusão

Publico alvo do IDE Arduino:

- Pessoas com pouco conhecimento de desenvolvimento de *software*
- Desenvolvedores com experiência previa em Arduino!
- Sem um grande interesse em *software* e Linux
- Sem interesse em mudar para um ambiente de produção



2. Intel XDK para IoT

Desenvolvimento *sandbox* para programadores *web*



Intel XDK para IoT

Fácil para desenvolvedores *web* familiarizados com *JavaScript*.

Mas se preferir *NodeJS* no Linux, escolha a opção 4: Linux.

Funcionalidades e bibliotecas limitadas
mas kits de **sensores Plug'n'Play (PnP) validados**.

IDE fácil de instalar e de configurar,
mas “flashear” a placa não tão simples.

Necessário um **cartão microSD**.

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



Intel XDK para IoT

The screenshot displays the Intel XDK IoT Edition development environment. The main editor window shows a JavaScript file named `main.js` with the following code:

```
1 /*
2 A simple node.js application intended to blink the onboard LED on the Intel based development
3 boards such as the Intel(R) Galileo and Edison with Arduino breakout board.
4
5 MRAA - Low Level Skeleton Library for Communication on GNU/Linux platforms
6 Library in C/C++ to interface with Galileo & other Intel platforms, in a structured and sane API
7 with port names/numbering that match boards & with bindings to javascript & python.
8
9 The intent is to make it easier for developers and sensor manufacturers to map their sensors &
10 actuators on top of supported hardware and to allow control of low level communication protocol
11 by high level languages & constructs.
12 */
13
14 var mraa = require('mraa'); //require mraa
15 console.log('MRAA Version: ' + mraa.getVersion()); //write the mraa version to the Intel XDK
16 console
17
18 var myOnboardLed = new mraa.Gpio(13); //LED hooked up to digital pin 13 (or built in pin on
19 Galileo Gen1 & Gen2)
20 myOnboardLed.dir(mraa.DIR_OUT); //set the gpio direction to output
21 var ledState = true; //Boolean to hold the state of led
22
23 periodicActivity(); //call the periodicActivity function
24
25 function periodicActivity()
26 {
```

The interface includes a file explorer on the left showing the project structure for `Blink`, including `node_modules`, `icon.png`, `main.js`, `package.json`, and `README.md`. The terminal window at the bottom shows the following output:

```
=====
UPLOADING: Uploading project bundle to IoT device.
[ Upload Complete ]
Intel XDK - Message Received: run
=> Stopping App <=
MRAA Version: v0.5.0-12-g3898182
```



Intel XDK para IoT

Publico alvo:

- Desenvolvedores *web* ou iniciantes no desenvolvimento de *software*
- Se o projeto não vai usar sensores além dos PnP já suportados
- Sem interesse em usar Linux

Comparado com Arduino, Intel XDK é mais versátil, e ainda é possível mudar para linux-NodeJS se for necessário.



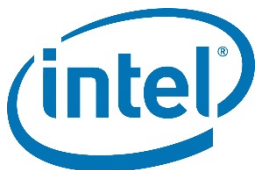
3. Intel IoT Developer Kit

Desenvolvimento com o IDE Eclipse

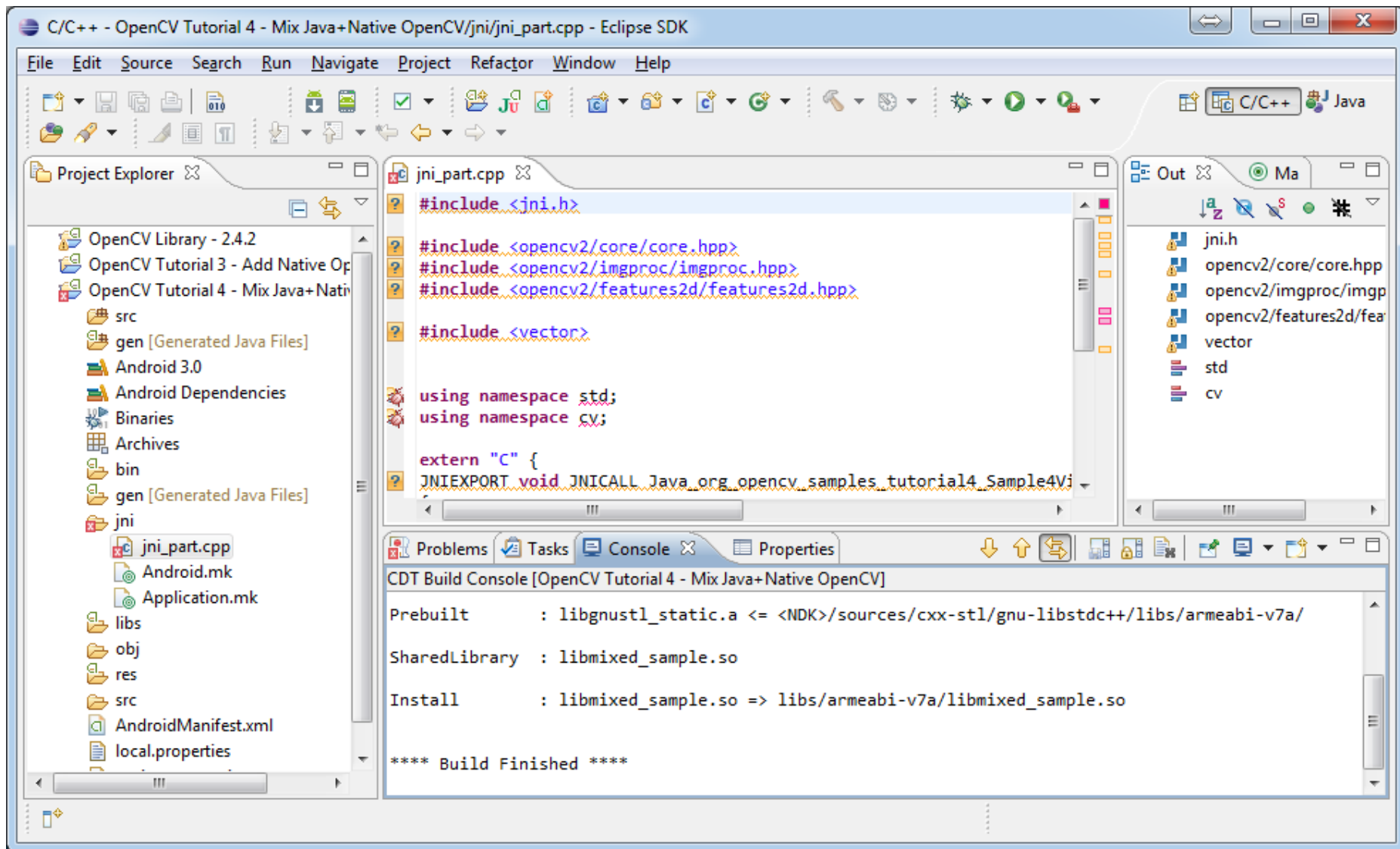


Intel IoT SDK com Eclipse

Eclipse no PC,
se comunicando com a placa IoT.
Sem acesso direto a placa,
ou a distribuição Linux instalada nela.
Principalmente desenvolvimento em C.



Intel IoT SDK com Eclipse



Intel IoT SDK com Eclipse

Publico alvo :

- Aficionado a desenvolver em C com Eclipse
- Sem necessidade de acesso ao Linux ou de instalar pacotes Linux para o projeto
- Sem interesse em usar Linux
- Possibilidade de programar código C de alto rendimento

Comparado ao XDK, C é menos “divertido” que JavaScript.

Possibilidade de mudar a linux-C em uma fase posterior do projeto.



Desenvolvimento com Linux na própria placa
Baixo nível, sem limites, estilo Linux



Desenvolvimento com Linux na placa

Acesso por *ssh* na placa desde a rede ou USB, instala os pacotes necessários, e programa com qualquer linguagem no Linux.

- Sem necessidade de instalar nada no PC
- Edição de código com *nano*, *emacs* ou *vi*.
- Compilação com *gcc* na própria placa.
- Possibilidade de usar VMs como *Python*, *NodeJS*, ...
- Possibilidade de interagir com serviços Linux (como *bluez* para características avançadas de Bluetooth).
- Interação com Yocto
- Possibilidade de construir uma distribuição a medida.

Recomendação: [NodeJS com Cylon](#).



Desenvolvimento com Linux na placa

Publico alvo :

- pessoas com experiência em Linux, ou em Raspberry Pi
- possibilidade de conectar qualquer sensor
- controle absoluto em baixo nível,
necessário se for mudar para um ambiente de produção
- sem necessidade de instalar nada no PC



Conclusão



Conclusão

Boas notícias: tem um método para todo o mundo.

Para migrar o código:

- Arduino : não é fácil migrar o código para outro entorno
- XDK : possível migração a Linux ou SDK
- SDK : possível migração a Linux
- Linux : possível migração a um entorno de produção e a um desenvolvimento profissional de sistemas embutidos / IoT

