



使用 yocto，您只需一天便可构建支持 Tizen-micro 的智能设备

Biao Lu
Austin Zhang



议程

- 当前状态和现有问题
- **Tizen-micro** 特性
- 快速、灵活构建设备
- 案例研究
- 概要
- 问答

当前状态



图片来源:

<http://www.rudebaguette.com/2014/02/13/mobile-smart-devices-set-drive-turnaround-frances-tech-device-market/>

<http://gadgets.in.com/smarthings-make-your-home-smarter.htm>

<http://www.cnn.com/2014/03/28/tech/innovation/smart-device-communication/>

现有问题

- 设置开发环境极为复杂。
- 需要掌握全面的堆栈知识
- 各种系统组件的稳定性难以验证。
- 难以满足占用空间方面的要求。



Tizen-micro 可如何帮助
解决这些问题?

Tizen-micro 101

它是什么



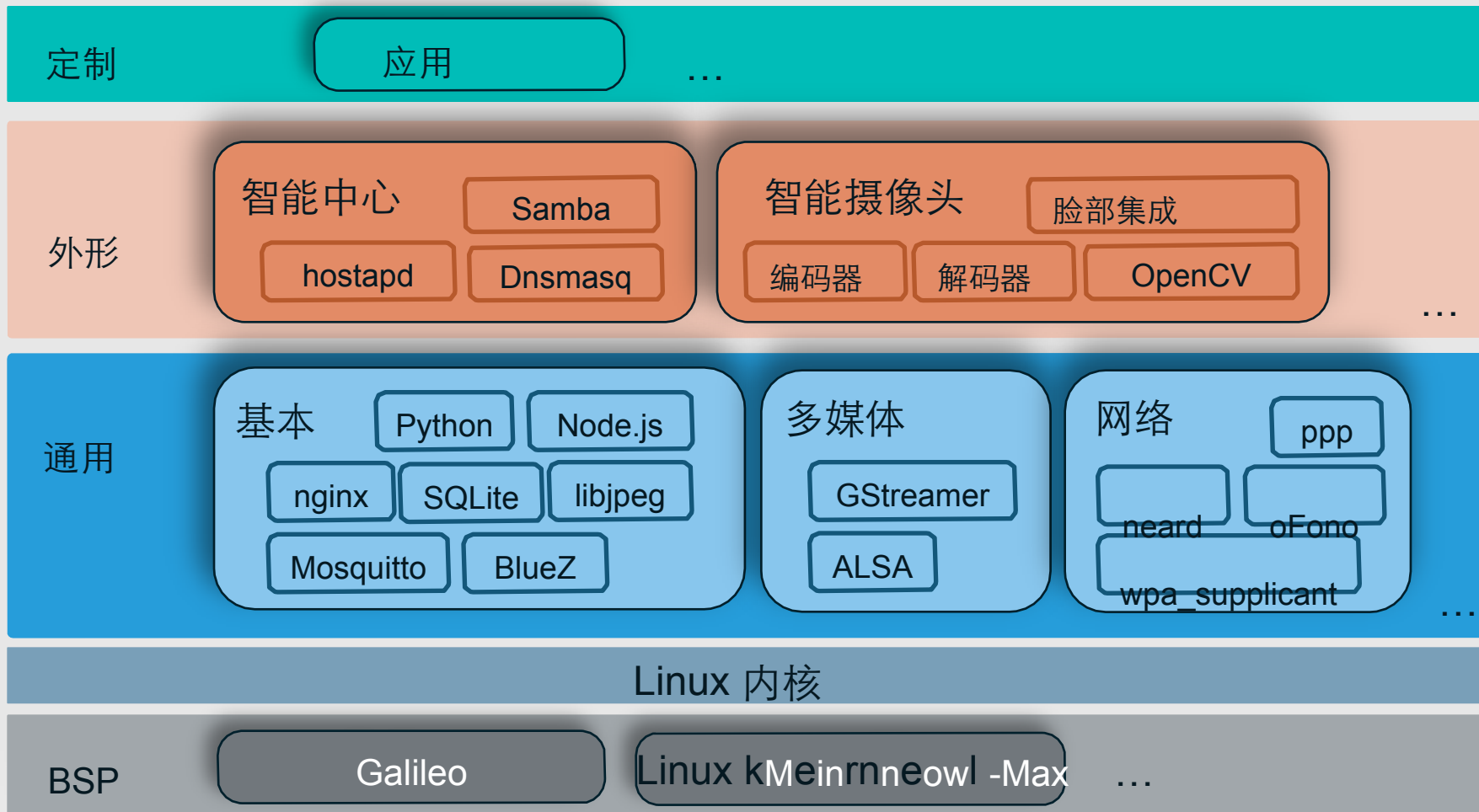
图片来源:

<http://www.xconomy.com/wordpress/wp-content/images/2011/11/question-mark-stockimage.jpg>

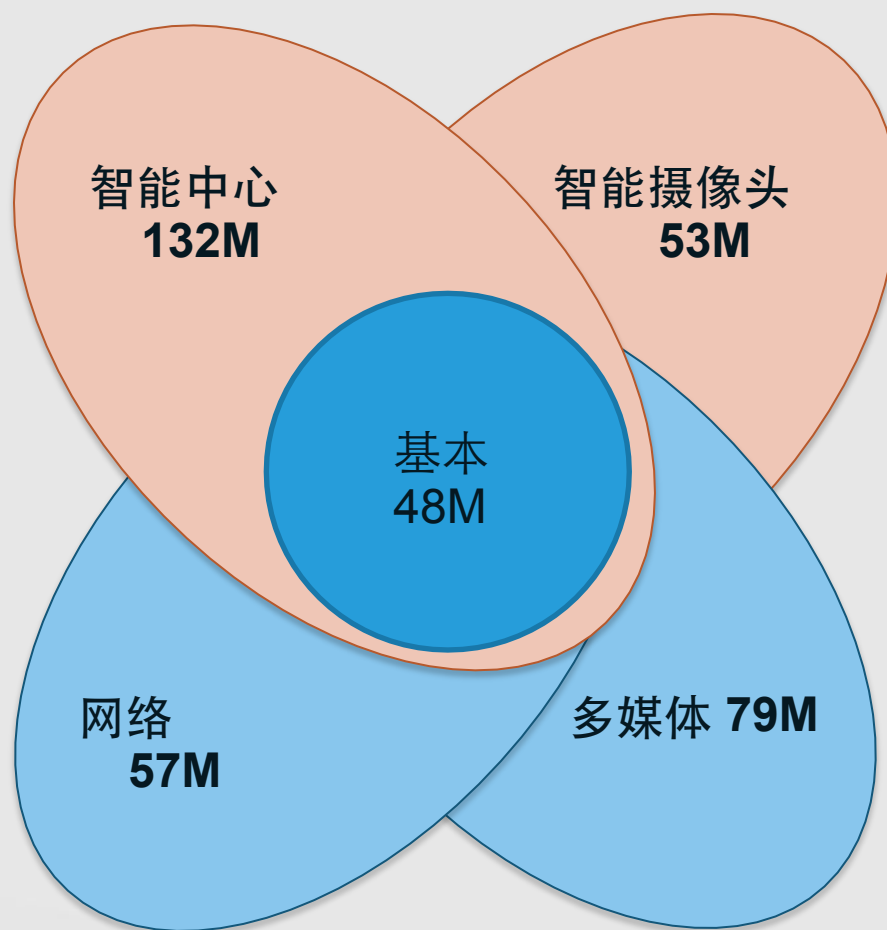
Tizen-micro

- 一种具有下列特性的 **Tizen** 系统：
 - 轻量（无头）、较小占用空间和多任务处理
 - 高度可配置和灵活性
 - 稳定性和验证
 - 清除和快速构建

Tizen-micro 架构



Tizen-micro 占用空间





如何将 Tizen-micro 用于 构建设备

快速、灵活构建设备

- 获取 **Tizen-micro**
 - 获得所有这些代码/层/pkg 组定义/脚本等，构建一个基于 Tizen-micro 的系统
- 树结构概览

```
├── tizen-micro/  
├── meta-micro-common/  
├── meta-smarthub/  
├── meta-intel/  
│   ├── meta-galileo/  
│   └── meta-minnowmax/  
├── .....  
├── setup/  
└── setup.sh
```

快速、灵活构建设备（续）

- 它使用 **Yocto** 作为构建工具和一些组件库
- 最简单的映像创建方式

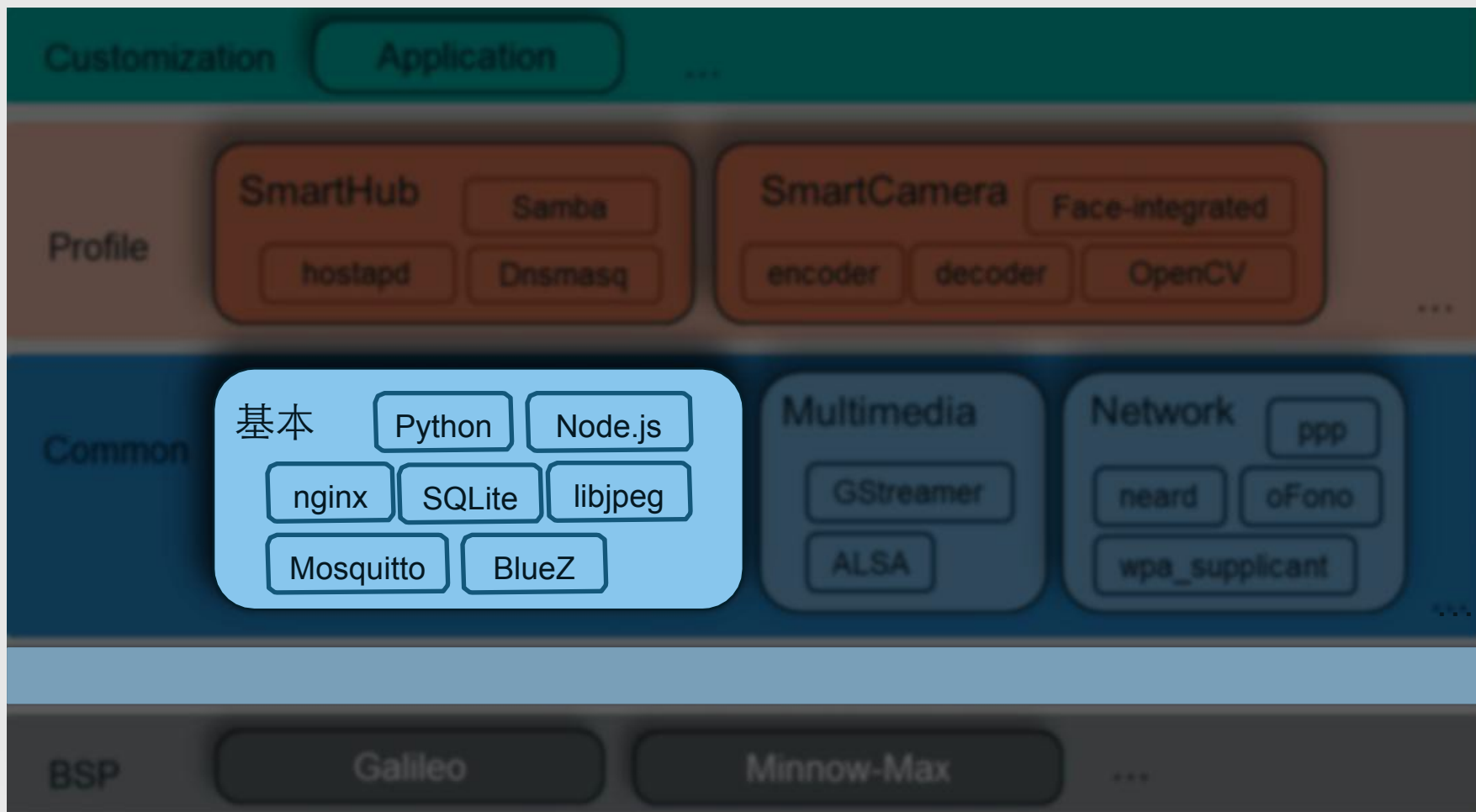
```
$ ./setup.sh  
$ bitbake micro-image
```

- 在虚拟机中运行

```
# ./setup/vm-run.sh
```

Yocto 项目: <https://www.yoctoproject.org/>

仅限基本系统



快速、灵活构建设备（续）

- 添加上图要求的组件
 - 预定义中间件组

```
$ vi meta-micro-common/recipes-core/images/micro-image.bb IMAGE_INSTALL +=  
"packagegroup-multimedia"
```

- 您需要的中间件组件/组

```
IMAGE_INSTALL += "your-middleware"
```

- 您的应用

```
$ vi your-app_0.1.bb
```

添加其他的中间件和应用



快速、灵活构建设备（续）

- 选择上图要求的 **BSP** 和外形
 - BSP

```
$ ./setup.sh -b boardname  
$ bitbake micro-image
```

- 如何集成您的 BSP 代码？

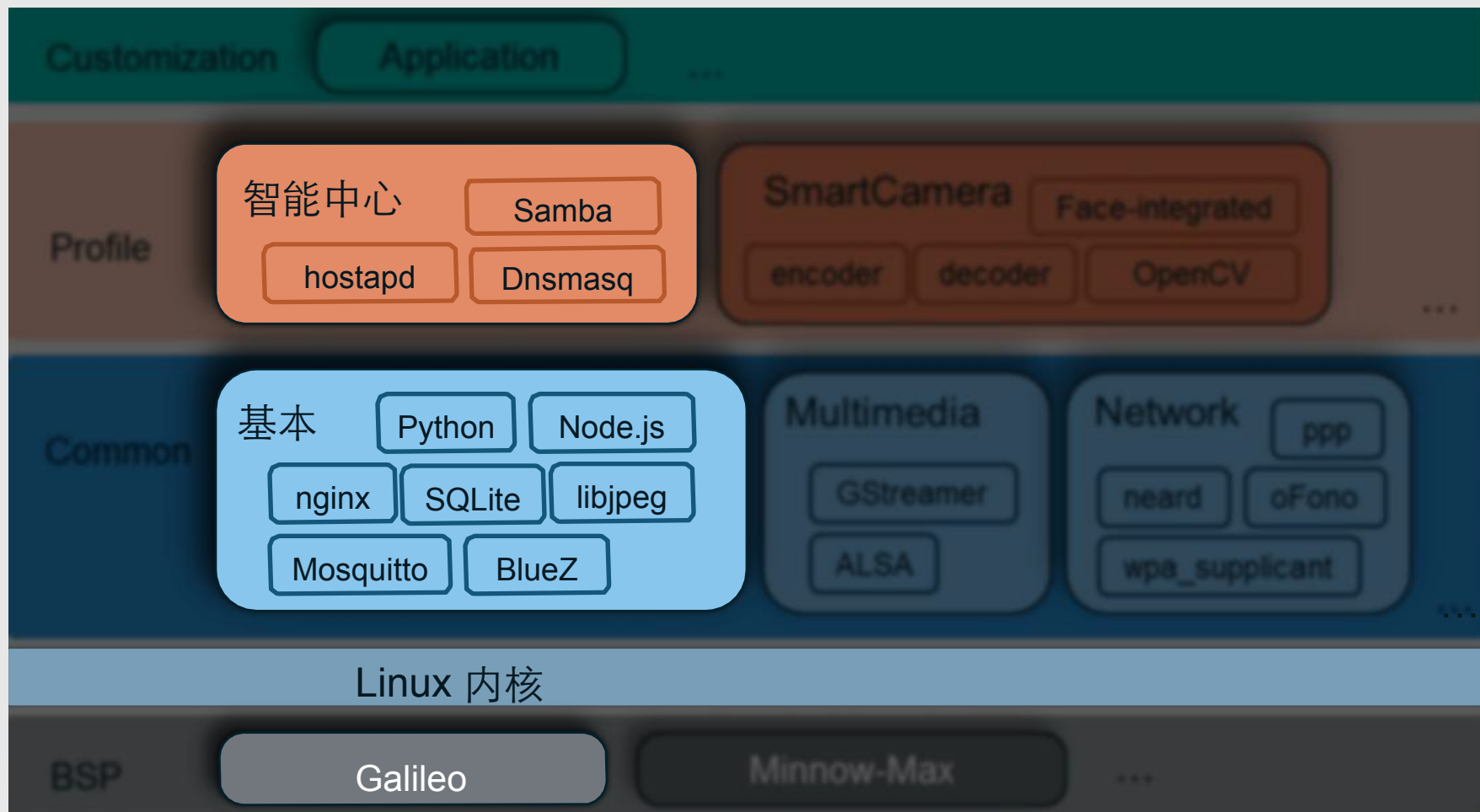
Put BSP under directory “meta-intel”, e.g.

meta-intel/meta-your-bsp

- 外形

```
$ ./setup.sh -b boardname -p smarthub  
$ bitbake smarthub-image
```


添加 BSP / 外形支持

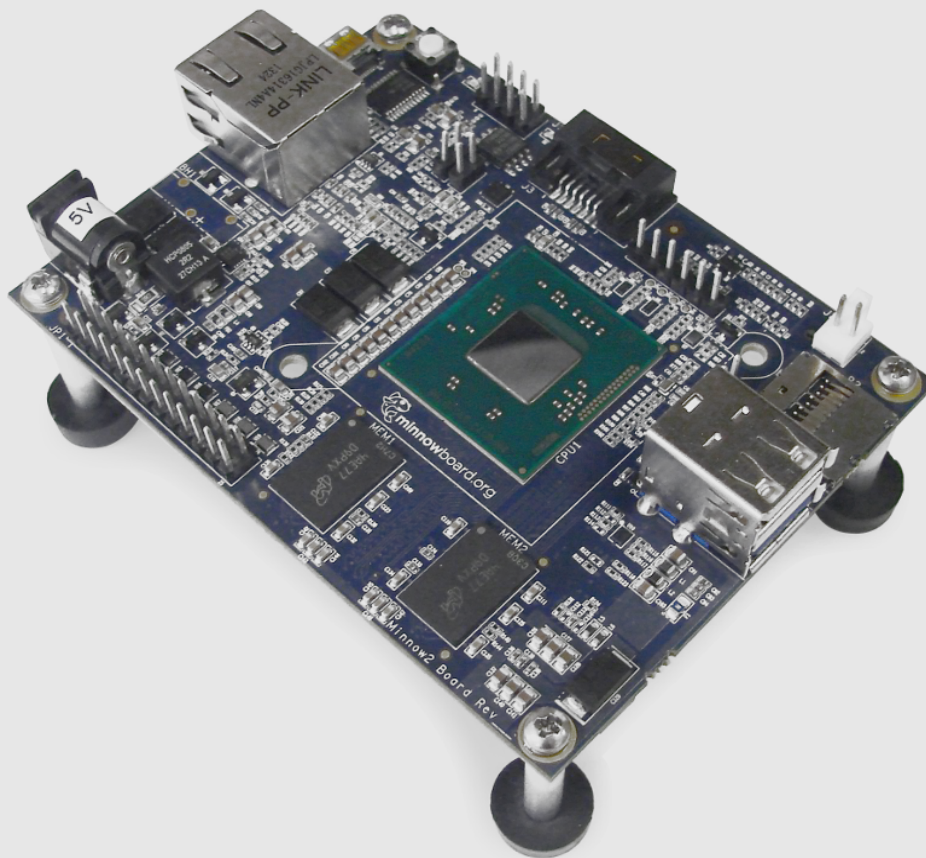




从头构建一台智能设备：

现在，我们能够快速、高效地构建智能设备！

案例研究 – Minnow-Max 智能中心



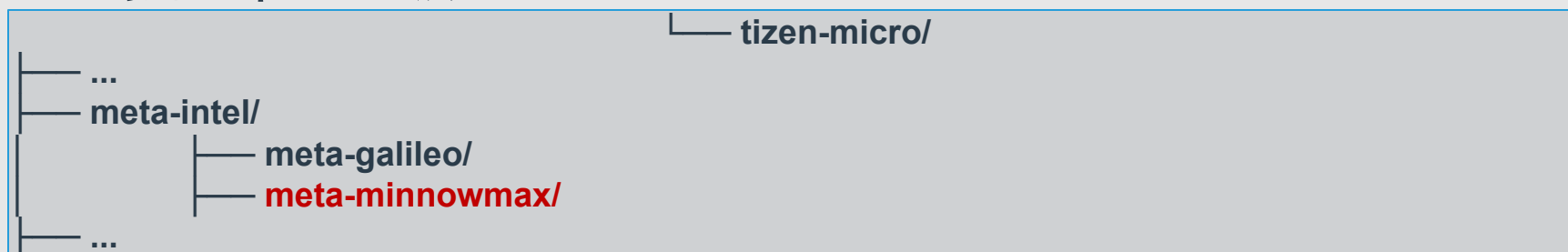
图片来自: <http://www.minnowboard.org/meet-minnowboard-max/>

案例研究 – Minnow-Max 智能中心

- 一个稳定、经验证的基本系统：**Tizen-micro** 通用基本
- 其他的产品外形：智能中心
- 定制 **UI/UX**
- 一种出色的硬件平台：**Minnow-Max**

案例研究 – Minnow-Max 智能中心（续）

- 创建一个 **BSP** 层



- 新层配置文件 (layer.conf)

```
BBPATH .= ":${LAYERDIR}"
```

```
BBFILES += "${LAYERDIR}/recipes-*//*.bb \
${LAYERDIR}/recipes-*//*.bbappend"
```

```
BBFILE_COLLECTIONS += "minnowmax" BBFILE_PATTERN_minnowmax := "^${LAYERDIR}/"
BBFILE_PRIORITY_minnowmax = "5"
```

案例研究 – Minnow-Max 智能中心（续）

- 新机器配置文件 (machine/minnowmax.conf)

```
DEFAULTTUNE ?= "corei7-64"
```

```
require conf/machine/include/tune-corei7.inc require conf/machine/include/x86-base.inc
```

```
PREFERRED_PROVIDER_virtual/kernel = "linux-minowmax"
```

```
PREFERRED_VERSION_linux-minowmax = "3.14%"
```

```
SERIAL_CONSOLES = "115200;ttyS0"
```

案例研究 – Minnow-Max 智能中心（续）

- 内核配置文件 (linux-minnowmax_3.14.bb)

```
require recipes-kernel/linux/linux-yocto.inc
```

```
SRC_URI = "git://git://git.kernel.org...git;branch=linux-3.14.y;" SRC_URI += "file://  
minnowmax.patch"
```

```
SRC_URI += "file://minnowmax.cfg"
```

```
LINUX_VERSION ?= "3.14.4" LINUX_VERSION_EXTENSION ?= "-minnowmax"
```

```
SRCREV_machine = "62f236c734996f240d91daee2cb6a05669c7326c"
```

```
PR = "r0"
```

```
PV = "${LINUX_VERSION}"
```

```
COMPATIBLE_MACHINE = "minnowmax"
```

案例研究 – Minnow-Max 智能中心（续）

- 构建映像

```
$ ./setup.sh -b minnowmax -p smarthub  
$ bitbake smarthub-image
```




最后且同样重要的 ϑ 概要

概要

- 随着智能设备的创新不断深化，开发人员需要便捷地支持基本系统，以便更快速地将想法转化为现实。
- **Tizen-micro** 的许多特性可帮助您快速构建智能设备。
 - 完整的构建环境
 - 稳定和经验证的中间件
 - 较小的占用空间
- 开发人员能够通过简单步骤，快速、灵活地构建系统。
- 将 **Minnow-Max** 作为示例，展示如何使用 **Tizen-micro** 构建系统。



问答



28



TIZENTM

DEVELOPER SUMMIT 2014



SHANGHAI



TIZEN开发者峰会（上海）

TIZENTM DEVELOPER SUMMIT
2014 SHANGHAI
TIZEN开发者峰会（上海）