



**FTF 2016**  
TECHNOLOGY FORUM

# CREATING AN ADVANCED EMBEDDED SETUP WITH YOCTO

**FTF-AUT-N1787**

HEINZ WROBEL  
AUTOMOTIVE FAE MANAGER  
FTF-AUT-N1787  
MAY 19, 2016

PUBLIC USE



# AGENDA

- Goals and Ease Of Use, Demo
- Setting up Meta-bluebox, Demo
- Machine and Image Configuration, Demo
- Using Classes to Add Generic Features, Demo
- Kernel and U-Boot Tweaks, Demo
- XFCE on an Embedded System, Demo
- Miscellaneous Topics

**BEWARE!**  
**WE WILL USE VERY**  
**FEW POWERPOINT**  
**SLIDES!**

# GOALS AND EASE OF USE

# Goals

- Show you a real implementation of a complex SDK modification, **meta-bluebox**
- Illustrate possible options for Yocto setup and recipe changes
- Discuss (bad | good | better) options with you!
- Point to compromises made
  
- We want to be more nimble with **meta-bluebox** releases than with the “big” SDK
  - **meta-bluebox** is pretty rapid Work In Progress!
  - Quality level should still be high
  
- You learn about what I do for BlueBox!
- I learn about what you think of it!

# Ease Of Use

- **meta-bluebox** layer should be easy to integrate into existing SDKs
- BlueBox images should be easy to build, and deploy
  - We look at that for images and deploy scripts
- The images should be full featured for Ease Of Use
  - Image definitions

# SETTING UP META-BLUEBOX



# The Setup Support

- Single README.FIRST
- Single setup script
- Names matched to the SDK structure
  
- **Be obvious**
- **Be reproducible**
- **Be straightforward**

Left	File	Command	Op
<	~/meta-bluebox		
'n	Name		Size
/..		UP - -DIR	
/classes			4096
/conf			4096
/images			4096
/packagegroups			4096
/recipes-bsp			4096
/recipes-connectivity			4096
/recipes-core			4096
/recipes-devtools			4096
/recipes-dpaa2			4096
/recipes-extended			4096
/recipes-graphics			4096
/recipes-kernel			4096
/recipes-support			4096
/recipes-virtualization			4096
/sdkpatches			4096
README.FIRST			14455
*setup-meta-bluebox			5979



**DEMO TIME!**



# MACHINE AND IMAGE CONFIGURATION

# Distro and Machine are Two Separate Topics!

- layer.conf always needed
- distro defines overall behavior
- machine defines target specifics

```
< ~/meta-bluebox/conf
```

<b>n</b>	<b>Name</b>	<b>Size</b>
	<b>/..</b>	<b>UP--DIR</b>
	<b>/distro</b>	<b>4096</b>
	<b>/machine</b>	<b>4096</b>
	<b>layer.conf</b>	<b>1168</b>

# Image Definitions

- Incremental image definition for BlueBox
- SDK specific baseline references
- Use case specific images
  - NOR Flash image
  - Standard image
  - Graphical Desktop image

```
< ~/meta-bluebox/images
```

'n	Name	Size
	/..	UP--DIR
	crosslayer-image-core.inc	97
	crosslayer-image-full.inc	97
	fsl-image-blueboxadditions.inc	2665
	fsl-image-blueboxadditionsdt.inc	1014
	fsl-image-blueboxadditionsvirt.inc	481
	fsl-image-blueboxls2.bb	104
	fsl-image-blueboxls2dt.bb	74
	fsl-image-blueboxls2nor.bb	2181
	fsl-image-blueboxs32v2xx.bb	104
	fsl-image-blueboxs32v2xxdt.bb	78
	fsl-image-blueboxt4.bb	194
	fsl-image-blueboxt4dt.bb	122
	fsl-image-blueboxt4nor.bb	2449
	fsl-image-core.bbappend	53
	fsl-image-full-s32v2xx.bb	898
	fsl-image-s32v2xx.bb	807
	fsl-image-s32v2xxadditions.inc	171

**DEMO TIME!**



# USING CLASSES TO ADD GENERIC FEATURES

# Custom Classes

- Classes add features to recipes
  - Automatic NOR binary image
  - .itb file generation for ARMv8
  - Bootable SD-Card for S32V234
  
- An image is a recipe!

```
< ~/meta-bluebox/classes
```

'n	Name	Size
/..		UP--DIR
	bash-completion.bbclass	180
	image_types_fsl_flashimage.bbclass	6511
	image_types_fsl_itb.bbclass	4986
	image_types_fsl_sdcard.bbclass	18870
	upstream-version-is-even.bbclass	299



**DEMO TIME!**





# KERNEL AND U-BOOT TWEAKS

# Linux Kernel Customization

- How to override the kernel config
- Dealing with target specific patches
  - Patches
  - File additions
  - Kernel config fragments

```
< ~/meta-bluebox/recipes-kernel/linux
```

'n	Name	Size
	../	UP--DIR
	/linux-ls2-sdk	4096
	/linux-qoriq	4096
	/linux-s32v2xx	4096
	linux-ls2-sdk_4.1.bbappend	519
	linux-qoriq_3.12.bbappend	371
	linux-s32v2xx.inc	2972
	linux-s32v2xx_4.1.17.bb	1147

# U-Boot Customization

- U-Boots are very SoC/Board specific
- U-Boot default environments required!
  - Added simple way for custom env
  - Added missing tool build for env

```
< ~/meta-bluebox/recipes-bsp/u-boot
```

'n	Name	Size
	/. .	UP- -DIR
	/u-boot-environment	4096
	/u-boot-ls2	4096
	/u-boot-qoriq	4096
	/u-boot-s32v2xx	4096
	u-boot-environment_1.0.bb	1549
	u-boot-ls2_2015.10.bbappend	800
	u-boot-mkenvimage_2015.10.bb	1233
	u-boot-qoriq_2015.01.bbappend	552
	u-boot-s32v2xx_2016.01.bb	4328

**DEMO TIME!**



# XFCE ON AN EMBEDDED SYSTEM

# Adding XFCE Into the Picture

- Few additions actually, but ...
- **Complex** to get something usable
  - lxdm
  - x11vnc/xvfb
  - consolekit
  - Packagegroups for xfce/x11
  - Config file tweaks ...
  - Dependencies, dependencies ...

```
< ~/meta-bluebox/recipes-core
'n      Name      Size
/..     UP--DIR
/base-files      4096
/busybox
/dbus
/glib
/initramfs
/jpeglib
/libc
~/meta-bluebox/recipes-graphics
'n      Name      Size
/..     UP--DIR
/gphoto2      4096
/gpu-viv-bin-s32v2xx 4096
/kernel-module-g
/lxdm
/packagegroups
/x11vnc
/xorg-xserver
~/meta-bluebox/recipes-xfce
'n      Name
/..
/ca-certificates
/consolekit
/debianutils
/lm_sensors
/udisks
/vim
File: fsl-imag-sdt.inc Line 1 Col 0
#
# This file contains the additions specified in the
# ease of use for the Blue Box by providing a
# desktop to use. Hence the "dt" suffix.
#
# The various gfx related packages are
# some of the selections are quite expensive
# Ensure lxdm is our display manager and
IMAGE_INSTALL += " \
    lxdm \
    lxdm-init \
"
# Ensure X is in this installation
IMAGE_INSTALL += " \
    packagegroup-core-x11 \
    xserver-xorg-xvfb \
"
# Support for remote display usage
IMAGE_INSTALL += " \
    x11vnc \
    x11vnc-init \
"
```



**DEMO TIME!**



# MISCELLANEOUS TOPICS



# There is More...

- Examples for additional package setups
  - lmsensors
  - yum
  - Bug fixes through .bbappend
  - Various other things
- What else would you like to discuss?



SECURE CONNECTIONS  
FOR A SMARTER WORLD

## ATTRIBUTION STATEMENT

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, CoolFlux, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE Classic, MIFARE DESFire, MIFARE Plus, MIFARE Flex, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TrenchMOS, UCODE, Freescale, the Freescale logo, AltiVec, C 5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. All other product or service names are the property of their respective owners. ARM, AMBA, ARM Powered, Artisan, Cortex, Jazelle, Keil, SecurCore, Thumb, TrustZone, and  $\mu$ Vision are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. ARM7, ARM9, ARM11, big.LITTLE, CoreLink, CoreSight, DesignStart, Mali, mbed, NEON, POP, Sensinode, Socrates, ULINK and Versatile are trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org. © 2015–2016 NXP B.V.

