Resume

Hans Buchmann FHNW/ISE

7. Januar 2020

Um was geht es? GNU/Linux Teil 1

- ► Toolchain bare
- kernel
- libc
- ► Toolchain full
 - ▶ inkl. **C**++
- busybox
- ssh
 - ► Client/Server

Remark: basierend auf 17-build

Um was geht es? GNU/Linux Teil 2

- Network
 - per USB
 - per WiFi
- CrossDevelopment
 - ► Entwicklung auf dem *Host*
 - Executables ouf dem BBW
- ► Weitere *packages*
 - lighttpd

Host Verzeichnisstruktur

17-build
SIGNATURE invisible uniqe number
config
toolsscripts
config tools scripts component.sh
build
component
tctoolchain
target-root on the <i>Host</i>
mountroot on BBW per sshfs

		args
toolchain bare	binutils.sh	
	gcc-bare.sh	
kernel	kernel.sh	[bb.org_defconfig]
		zImage
		dtbs
		headers_install
libc	glibc.sh	

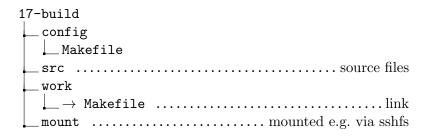
Build Fortsetzung

		args
toolchain full	gcc-target.sh	
UNIX	busybox.sh	[menuconfig]
		busybox
		install
	zlib.sh	
	openssl.sh	
	openssh.sh	
toolchain-host	gcc-host.sh	
Distro	target-root.sh	
	tc.sh	

BeagleBoneWireless

Verzeichnisstruktur

Development



Was ist noch zu tun?

- ▶ Wo befinden sich die Source Files
 - nicht immer suchen
- user-root Separation
 - Host user (fast immer)
 - **BBW root**: für den Moment mindestens
- ► Workflow für schrittweise Entwicklung
 - Saubere Anwendung von rsync
 - Ev. anderes Tool: (wie steht es mit git) ?
- Kernel Konfiguration
 - omap_i2c 4819c000.i2c: timeout waiting for bus ready
- Device Tree